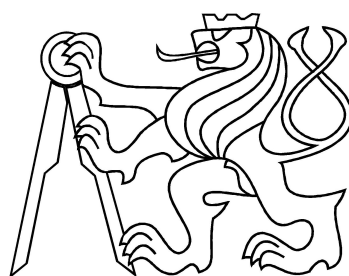


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ

Katedra řídicí techniky



BAKALÁŘSKÁ PRÁCE

Řízení modelu lokomotivy

Vedoucí: Ing. František Vaněk

Praha, 2007

Autor: Dan Pavlišta

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb. , o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne _____

_____ podpis

Poděkování

Děkuji vedoucímu Ing. Františku Vaňkovi za odborný dohled a cenné rady při řešení této práce. Neméně tak děkuji rodičům a blízkým za plnou podporu při studiu.

Abstrakt

Cílem této práce je navrhnout a realizovat software pro řízení modelové lokomotivy, jako součást projektu dopravního sálu fakulty dopravní ČVUT. Realizace bude provedena pomocí procesoru Atmel ATMega8 a dodaného hardware. Software musí obsahovat regulátor rychlosti a udržovat spojení s nadřazeným systémem.

Abstract

Objective of this bachelor thesis is realizing control software of model locomotive for department of transportation sciences of CTU. Solution is performed for microprocessor Atmel ATMega8 and existing hardware. Control software implements speed controller and communicates with shift system.

originální zadání umístit sem !!!!

Obsah

Obsah	viii
Seznam obrázků	ix
Seznam tabulek	xi
1 Úvod	1
1.1 Popis projektu dopravního sálu a lokomotivy	1
1.2 Rozbor zadání práce	2
1.3 Návrh řešení s mikroprocesorem Atmel ATMega8	5
1.3.1 Atmel ATMega8	5
1.3.2 Vývojový kit STK500	5
1.3.3 Sériové rozhraní UART	6
1.3.4 Porty mikrokontroléru	6
1.3.5 Rozhraní TWI - Two Wire Interface (I^2C)	6
2 Řešení	7
2.1 Identifikace modelu lokomotivy	7
2.2 Výpočet převodních konstant	11
2.2.1 Konstanta rychlosti na jednotku šířky PWM	13
2.2.2 Konstanta počtu přerušení na jednotku šířky PWM	13
2.2.3 Konstanta převodu rychlosti na počet přerušení	13
2.3 Návrh regulátoru	13
2.3.1 Volba vzorkovací frekvence	14
2.3.2 Návrh regulátoru metodou geometrického místa kořenů.	14
2.3.3 Diskretizace regulátoru	15
2.3.4 Windup	16
2.4 Realizace regulátoru	18

2.4.1	Generování PWM (akční veličina)	18
2.4.2	Implementace softwarového regulátoru	18
2.4.3	Pásmo necitlivosti	19
2.4.4	Optický IR snímač (zdroj informace o výstupu soustavy)	19
2.5	Popis implementovaného SW	20
2.5.1	Hlavní smyčka	20
2.5.2	Seriová komunikace	21
2.5.3	Infrabalýza	21
2.5.4	TWI	22
2.5.5	Zabezpečení dat komunikace BCC	23
2.5.6	Timer	23
2.5.7	Odometr	23
2.5.8	PWM	23
3	Ověření funkce a vyhodnocení parametrů	25
3.1	Monitorovací SW	25
3.1.1	Komunikační protokol	26
3.2	Experimenty	27
3.2.1	Regulace bez jevu windup	27
3.2.2	Regulace na pomalou rychlost	27
3.2.3	Experiment s pomalou jízdou ve stoupání.	28
4	Závěr	31
	Literatura	33
A	Seznam použitého software	I
B	Obsah příloženého CD	III

Seznam obrázků

1.1	Blokové schema dopravního sálu	2
1.2	Blokové schema vnitřního zapojení lokomotivy	4
2.1	Identifikace modelu. (Nevodivá spojka na značce 190cm.)	8
2.2	Přechodová charakteristika, střída PWM 100%(maximální rychlost)	9
2.3	Přechodová charakteristika, střída PWM 59%	10
2.4	Převodní charakteristika (naměřené otáčky motoru v závislosti na šířce PWM).	12
2.5	Schema pro ověření návrhu regulátoru v simulinku	15
2.6	Simulovaná odezva vlaku s PI regulátorem	16
2.7	Naměřená odezva lokomotivy při skoku na $r(k) = 100int/T_{vz} \equiv (294mm.s^{-1})$	17
2.8	Blokové schema řídicího software	20
2.9	Paket odesílaný z PC do lokomotivy	21
2.10	Paket odesílaný lokomotivou do monitorovacího SW	21
2.11	Paket odesílaný ze serveru přes Bluetooth do lokomotivy	22
2.12	Paket odesílaný z lokomotivy zpět serveru	22
3.1	Screenshot aplikace;	26
3.2	Ukázka regulace bez jevu windup.	27
3.3	Ukázka regulace na $41mm.s^{-1}$	28
3.4	Ukázka regulace do stoupání při rychlosti $59mm.s^{-1}$	29

Seznam tabulek

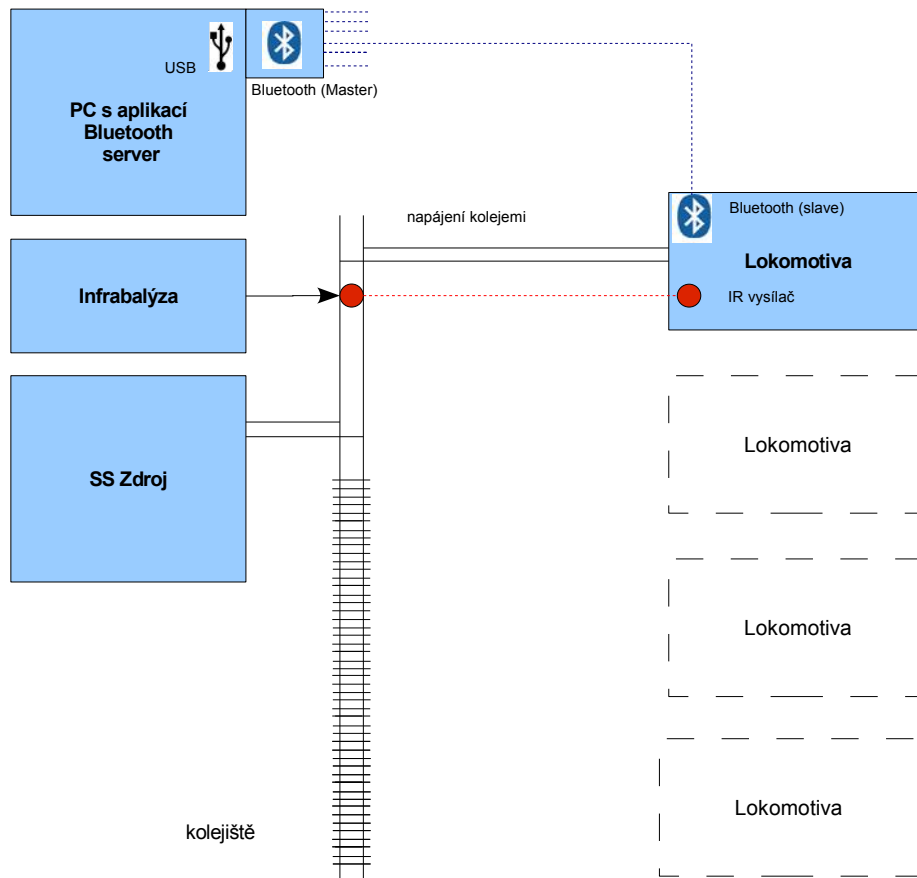
2.1	Přechodová charakteristika pro PWM se střídou 100%	8
2.2	Přechodová charakteristika pro PWM se střídou 59%	10

Kapitola 1

Úvod

1.1 Popis projektu dopravního sálu a lokomotivy

Pro podporu výuky zabezpečovací techniky provozu na železničních tratích staví fakulta dopravní (FD) sál s modelem několika stanic propojených kolejištěm. Měřítko je standardní modelářské 1:87, označované jako H0. Počítá se provoz maximálně 20-ti vlakových souprav na kolejišti. Vlakové soupravy budou vybaveny komunikačním zařízením pracujícím na technologii Bluetooth prostřednictvím něhož bude probíhat komunikace mezi modelem lokomotivy a centrálním řídicím serverem. Server, jehož úkolem je monitorování celkové provozní situace bude na základě okamžitého stavu vyhodnocovat polohu a rychlost jednotlivých vlakových souprav a odesílat povely lokomotivám. Úkolem lokomotivy je udržovat spojení se serverem a podle dodávaných dat dodržovat příslušnou rychlost. Komunikace mezi lokomotivou a serverem je oboustranná. Lokomotiva má na podvozku umístěno IR čidlo, které komunikuje s IR zařízeními pevně umístěnými v kolejišti. Čte infraznačky a odesílá informace o své poloze zpět serveru, který upravuje podle těchto hodnot své výpočty. Veškeré tyto úkoly v lokomotivě obstarává mikroprocesor Atmel ATmega8. Napájení je zprostředkováno kolejištěm, které usměrněno napájí SS motorky v lokomotivách a z něj získané stabilizované napětí +5V napájí procesor v lokomotivě. Pro snímání počtu otáček je v lokomotivě na setrvačnicku umístěn optický infračervený snímač poskytující 5 pulsů na 1 otáčku setrvačnicku lokomotivy. Prokluzu koleček je zabráněno gumovou bandáží jedné z čtyř náprav.

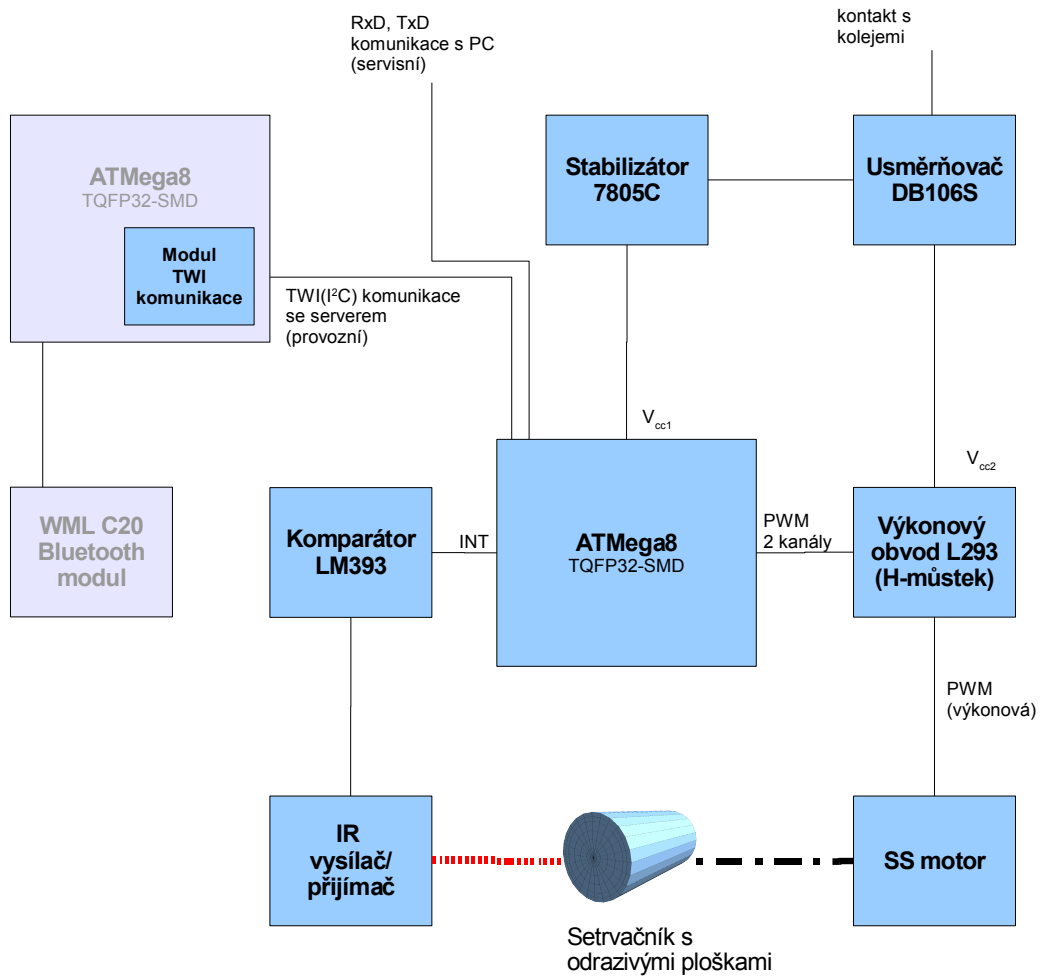


Obrázek 1.1: Blokové schéma dopravního sálu

1.2 Rozbor zadání práce

Nejprve je nutno se seznámit s funkcemi a programováním mikroprocesorů Atmel AT-Mega8 a zejména pak obsluze implementovaných periferií. Modul Bluetooth komunikace (zařízení WMLC20 a AT-Mega8 jsou spolu osazeny na jedné desce) byl převzat s obslužným SW do kterého je třeba implementovat komunikaci s hlavním řídicím procesorem. Nelze však využít seriové linky procesoru který řídí Bluetooth, protože je využita právě ke komunikaci s tímto zařízením. Pro tuto komunikaci je vhodné využít protokolu TWI,

který je v procesoru hardwareově implementován. Dále pak navrhnout realizovat software lokomotivy a to tak aby obsahoval regulátor rychlosti kterému bude výstupem četnost impulsů z optického snímače a vstupem zasláná hodnota velikosti rychlosti. Přitom je vhodné využít vestavěné PWM kanály, IR snímač otáček a obvod L293 jako výkonovou část ovládání motorku. Konkrétní zapojení bylo navrženo hlavním konzultantem práce. Na následujícím schématu jsou vedeny pro přehlednost jen hlavní signálové cesty.



Obrázek 1.2: Blokové schéma vnitřního zapojení lokomotivy

1.3 Návrh řešení s mikroprocesorem Atmel ATMega8

1.3.1 Atmel ATMega8

Jedná se o osmibitový nízkopříkonový mikrokontroler s 8kB programovatelné FLASH paměti. Je založen na pokročilé AVR RISC architektuře. Tyto mikrokontrolery mají Harwardskou architekturu, tedy mají oddělené paměti programu a dat. Paměť programu je typu FLASH, paměť dat je typu EEPROM. Mikroprocesor implementuje množství IO periférií, například 3 čítače/časovače, které lze využít jako nezávislé PWM kanály. Osmikanálový AD převodník s přesností 8 až 10 bitů. Implementované rozhraní TWI, prakticky odpovídá protokolu I^2C jehož patent je však vlastnictvím společnosti Philips a jde tedy o jeho variantu. Ke komunikaci dále může být využito sériové rozhraní USART. Pracovní frekvence procesoru je maximálně 16MHz při napájecím napětí 5V. Pouzdro procesoru pro lokomotivu je v provedení TQFP (montáž SMD). K programování jsem využíval zapůjčené vývojové prostředí STK500 společnosti Atmel společně se software AVR Studio 4. Pro odladění aplikace a seznámení se s procesory AVR jsem měl k dispozici kontroler Atmel ATMega16 v pouzdře PDIP (standardní velké pouzdro). Tento způsob vývoje byl výhodný, protože ATMega16 oproti své menší verzi ATMega8 podporuje JTAG, což je programátor umožňující debugování aplikace přímo v procesoru. K ATMega8 nelze připojit JTAG programátor a programuje se rozhraním SPI (Serial Peripheral Interface), které aplikaci debugovat přímo v procesoru neumožňuje. Navíc je díky malým rozměrům pouzdra TQFP poměrně obtížné připojování vodičů na piny a testování funkce aplikace.

1.3.2 Vývojový kit STK500

Jedná se o produkt společnosti Atmel určený k práci s AVR mikrokontroléry Atmelu. Je k němu dodáván příslušný software AVR Studio 4, které složí jako překladač zdrojového kódu do .hex souboru pro AVR procesory. Je v něm možno simulovat funkci aplikace bez připojeného hardware, a u některých typů procesorů debugovat výslednou aplikaci přímo v cílovém procesoru. Implicitním jazykem pro AVR Studio je assembler, avšak pro vyšší přehlednost kódu v celém projektu byl zvolen jazyk C. Vývojový kit nabízí několik typů programování z nichž jsem využíval ISP (serial In-System Programming), které má vyveden port i pro připojení externě umístěného zařízení (v mém případě procesor

umístěný na plošném spoji uvnitř lokomotivy). K vývoji lze s výhodou použít periferií kitu jako jsou LED diody, tlačítka, sériový port nebo snadno přístupné vyvedení portů na piny.

1.3.3 Sériové rozhraní UART

Pro komunikaci s PC při vývoji potřebného software jsem využíval právě sériovou linku. Na portu D, jsou vyvedeny signály RxD a TxD. Tyto signály (Received data, Transferred data) však nemohou být připojeny přímo k portu RS232 osobního počítače a vyžadují přizpůsobení TTL logice. To lze řešit pomocí integrovaného obvodu (např. Max232 firmy Maxim apod.). Využíval jsem proto vývojového kitu STK-500, kde je tento převodník k dispozici.

1.3.4 Porty mikrokontroléru

Mikrokontroler ATmega8 má celkem 3 porty z nichž jsou dva osmibitové a jeden sedmibitový. Porty jsou obousměrné a téměř každému pinu je přiřazena alternativní funkce. Pro komunikaci sériovým rozhraním jsou to pro signály RxD a TxD piny PD0 a PD1. Na PD2 je vstup od externího přerušování INT0, které je využito pro zachytávání sestupných hran signálu optického snímače otáček motoru.

1.3.5 Rozhraní TWI - Two Wire Interface (I^2C)

Rozhraní používané pro komunikaci mezi malými zařízeními. Jeden vodič je datový a druhý nese hodinový signál. Celá sběrnice je připojena přes pull-up rezistory na kladný potenciál a zařízení tedy sběrnici uvádí do úrovně logické nuly. Při použití rozhraní bylo nutno vyřešit problém při komunikaci mezi zařízeními s různým napájecím napětím. Řídicí mikropočítač lokomotivy využívá napájecí napětí 5V aby mohl pracovat na nejvyšší možné frekvenci 16MHz. Bluetooth technologie je napájena zásadně napětím 3,3V. Podle specifikace TWI [4] je $V_{IHmin} = 0,7 \cdot V_{cc}$ a $V_{IHmax} = V_{cc} + 0,5V$. Napětí na sběrnici je určeno pull-up rezistory připojenými na úroveň H, která musí vyhovovat oboum podmínkám. Po dosažení napájecích napětí do podmínek vyjde interval $< 3,5; 3,8 > V$. Napětí je v lokomotivě realizováno odporovým děličem.

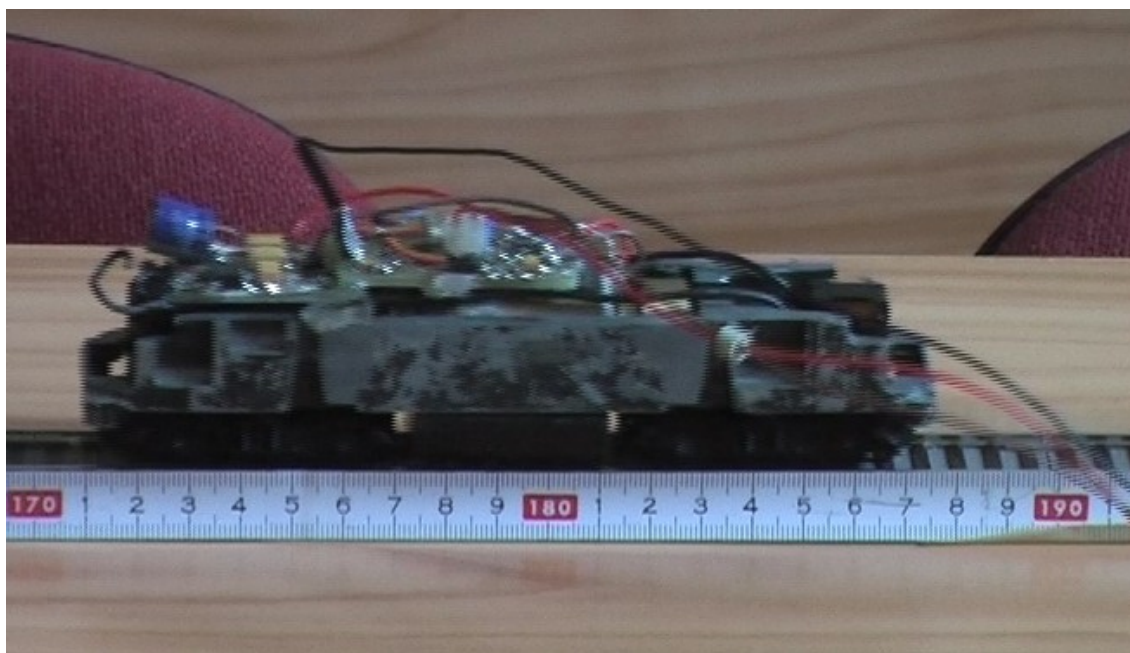
Kapitola 2

Řešení

Jako alternativní postup k prozkoumání byla doporučena možnost měřit ADpřevodníkem napětí na motoru v neaktivní části periody PWM a považovat jej za výstup soustavy. Řízení tímto způsobem nebylo implementováno z důvodů popsaných v závěru.

2.1 Identifikace modelu lokomotivy

Systém byl identifikován experimentálně pomocí přechodových charakteristik. Vstupem do systému nechť je číslo 0-255 představující šířku kladné hodnoty průběhu PWM uvnitř periody s délkou 255. Jde tedy o hodnotu tzv. Output compare registru (OCR) mikropočítače, jehož hodnota se neustále porovnává s čítačem a při rovnosti načítané hodnoty a OCR shodí/nastaví výstup. Lokomotiva byla připojena k PC seriovou linkou. K ovládání pod Windows byla použita aplikace popsaná v sekci 3.1 vytvořená přímo pro tento účel pomocí Microsoft Visual Studia. Aplikace periodicky odesílá žádanou hodnotu OCR do lokomotivy a přijímá zpět informace o okamžitých otáčkách. Díky zpoždění při odeslání a zpracování informace nelze přechodovou charakteristiku správně a jednoduše získat odesláním nulové žádané hodnoty (při charakteristice na dojezd). Měření tedy probíhalo na přechodu mezi napájenými a izolovanými koleji spojenými nevodivou propojkou. Propojovací izolace však byla zanedbatelně krátká (1mm) aby se neprojevila změna povrchu. Místo spoje bylo sledováno kamerou s frekvencí 25 snímků za vteřinu a na koleje viditelně přilepené měřítko s krokem 1mm. Výsledné nekomprimované video ve formátu avi bylo rozloženo na jednotlivé snímky a z nich poté získána funkce ujeté dráhy v čase pomocí diferencí. Naměřené přechodové charakteristiky byly proloženy systémy prvního



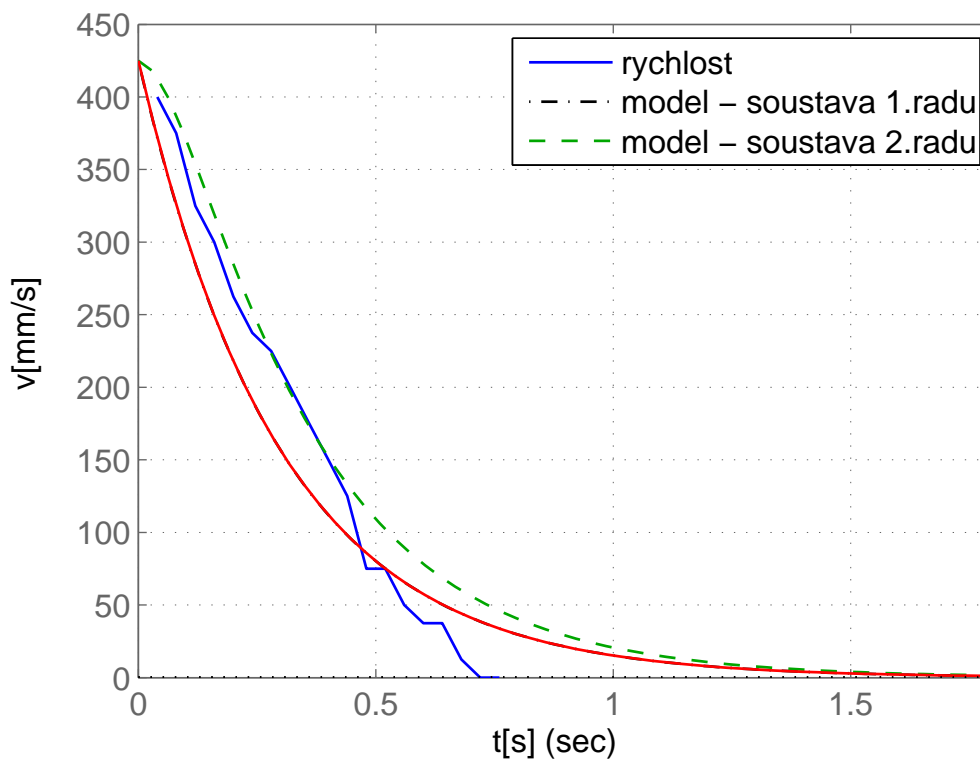
Obrázek 2.1: Identifikace modelu. (Nevodivá spojka na značce 190cm.)

nebo přesnějšími systémy druhého řádu. Výsledné časové konstanty byly vypočteny aritmetickým průměrem ze tří charakteristik pro různé střídy PWM.

Tabulka 2.1: Přechodová charakteristika pro PWM se střídou 100%

t[ms]	0	40	80	120	160	200	240	280	320
l[mm]	-3	13	28	41	53	63	73	82	90
v[mm/s]	-	400	375	325	300	262	237	225	200

t[ms]	360	400	440	480	520	560	600	640	680
l[mm]	97	103	108	111	114	116	117	119	119,5
v[mm/s]	175	150	125	75	75	50	37,5	37,5	12,5



Obrázek 2.2: Přechodová charakteristika, střída PWM 100% (maximální rychlost)

Přenos prvního řádu : ¹

$$F(s) = \frac{1416}{s + 3,3} \quad (2.1)$$

Přenos druhého řádu :

$$F(s) = \frac{17708}{(s + 3,3)(s + 12,5)} \quad (2.2)$$

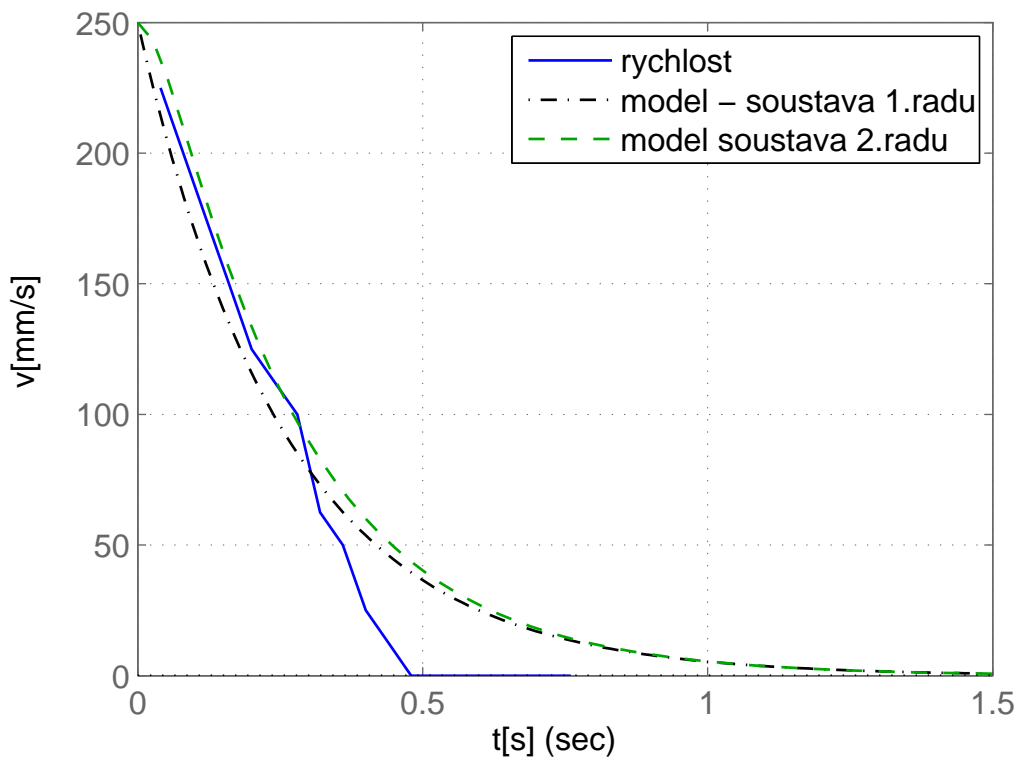
Měření dojezdu pro další nastavení střídavy :

¹Konstanty přenosů byly zaokrouhleny, přesné hodnoty viz. m-file.

Tabulka 2.2: Přejchodová chrakteristika pro PWM se střídou 59%

t[ms]	0	40	80	120	160	200	240	280
l[mm]	-3	6	14	21	27	32	36,5	40,5
v[mm/s]	-	225	200	175	150	125	112	100

t[ms]	320	360	400	440	480	520	560	600
l[mm]	43	45	46	46,5	46,5	46,5	46,5	46,5
v[mm/s]	62	50	25	12	0	0	0	0



Obrázek 2.3: Přejchodová chrakteristika, střída PWM 59%

Přenos prvního řádu : ²

$$F(s) = \frac{964}{s + 3,8} \quad (2.3)$$

²Konstanty přenosů byly zaokrouhleny, přesné hodnoty viz. m-file.

Přenos druhého řádu :

$$F(s) = \frac{25000}{(s+4)(s+25)} \quad (2.4)$$

Obdobně jsem získal časové konstanty z třetí charakteristiky a použil zprůměrovanou hodnotu. Výsledný přenos systému použitý pro regulaci tedy je :

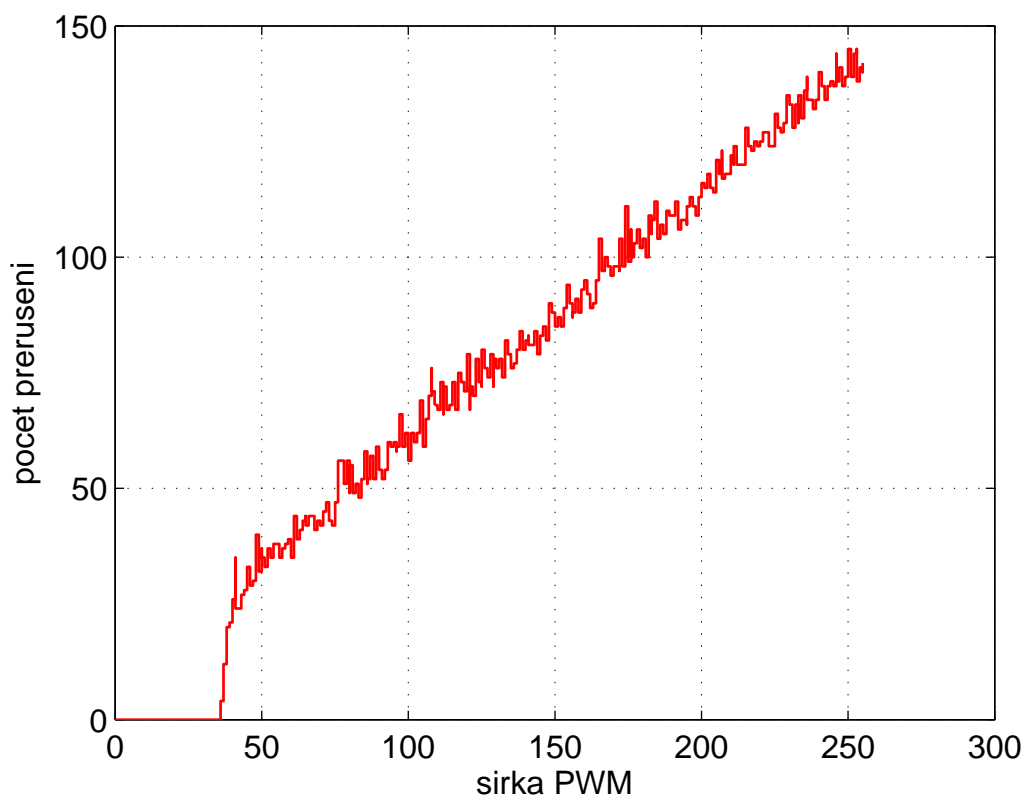
$$F(s) = \frac{33915}{(s+3,8)(s+21)} \quad (2.5)$$

2.2 Výpočet převodních konstant

Konkrétní požadavky na přesnost regulace nebyly stanoveny, ale lze je usuzovat z modelového měřítka. Vezměme-li skutečnou rychlost $10km \cdot h^{-1}$ a přepočteme v měřítku 1:87 vyjde nám rychlost $0,11km \cdot h^{-1}$ což je $0,03m \cdot s^{-1}$. Průměr kolečka je 11mm, obvod 34,5mm. Tato hodnota přibližně odpovídá požadavku na ujetou dráhu za vteřinu. Na jednu otáčku kolečka se setrvačnick otčí 10x a získá se tak 100 logických změn, tedy přerušení pro mikropočítač za vteřinu. Vydělíme-li tuto hodnotu 10 což je počet odměřů během vteřiny ($T_{vz} = 100ms$) získáme 10 přerušení což je pro řízení nízká hodnota pokud uvážíme pro tuto T_{vz} horní hranici 140 přerušení. Při nízkých rychlostech se navíc mohou nezanedbatelně projevit nelinearity typu suché tření, které řízení více komplikují než za vyšších rychlostí. Pro návrh regulátoru a ověření jeho funkce je zapotřebí převodních konstant mezi řídicími a výstupními veličinami.

Lokomotiva v měřítku 1:87 je poháněna stejnosměrným motorem s $U_{jm} = 12V$, jmenovité otáčky motoru nebylo možno přesně zjistit, převod na nápravu je 1:10. Měřením počtu přerušení byly jmenovité otáčky motoru stanoveny na $8000ot.min^{-1}$ tuto hodnotu je nutno brát pouze jako orientační.

Obsluha přerušení spočívá v inkrementaci proměnné, která je jednou za T_{vz} vynulována a použita jako informace o stavu výstupu soustavy. Experimenty s lokomotivou jsme stanovili přibližné požadavky tak, že pro potřeby řízení je nutné aby se proměnná za dobu T_{vz} načítala do hodnot řádu alespoň několika desítek.



Obrázek 2.4: Převodní charakteristika (naměřené otáčky motoru v závislosti na šířce PWM).

2.2.1 Konstanta rychlosti na jednotku šířky PWM

Tuto konstantu lze odešíst s naměřených přechodových charakteristik. Konstanta je zprůměrována ze všech tří charakteristik. Její hodnota je

$$K_{v1} = 1,68 \left[\frac{mm}{sec \cdot PWM} \right] \quad (2.6)$$

2.2.2 Konstanta počtu přerušení na jednotku šířky PWM

Konstanta byla změřena pomocí vnitřního SW lokomotivy a pomocí vlastní komunikační aplikace popsané dále. Měření probíhalo nastavením několika různých šíří PWM a zprůměrováním hodnot počtu přerušení v lokomotivě. Pro každou šířku PWM byla vypočtena konstanta a výsledná konstanta určena jako průměr.

$$K_{v2} = 4,9 \left[\frac{int}{sec \cdot PWM} \right] \quad (2.7)$$

Jednotka takto zapsána může vypadat chaoticky, popsána slovy je: 4,9 přerušení za vteřinu při šířce PWM 1/255. Výstupní veličina počet přerušení je v lokomotivě zjišťována periodicky s periodou T_{vz} jejíž volba je popsána dále 2.3.1.

2.2.3 Konstanta převodu rychlosti na počet přerušení

Kombinací dvou předešlých konstant lze přepočíst počet přerušení na rychlost a zpět.

$$K_{v3} = \frac{1,68}{4,9} = 0,34 \left[\frac{mm}{int} \right] \quad (2.8)$$

2.3 Návrh regulátoru

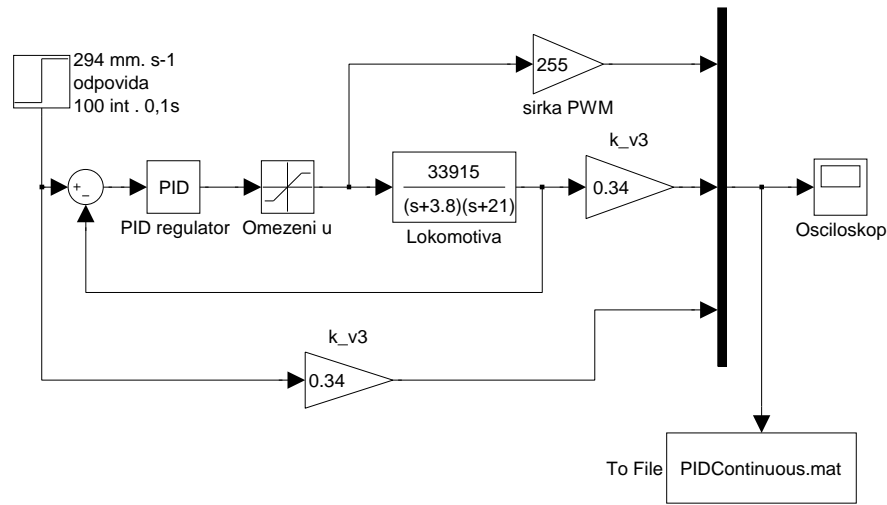
V zásadě se při návrhu regulátorů pro diskretní systém můžeme vydat dvěma cestami : Ze spojitého modelu soustavy nalézt diskretní a provést diskretní návrh, nebo pro spojitý model soustavy navrhnout regulátor metodami pro návrh spojitých regulátorů a ten posléze diskretizovat. Diskretní ekvivalent soustavy najdeme nejnadhěji pomocí Matlabu některou z metod diskretizace [2]. Jako spojitý model byl vzat model zprůměrovaný z identifikovaných modelů 2.5. Značení veličin v průbězích odpovídá [2], tedy $r(t)$ je žádaná hodnota, $y(t)$ výstup soustavy a $u(t)$ akční zásah.

2.3.1 Volba vzorkovací frekvence

Frekvence odběru vzorků musí splňovat základní zákon o vzorkování tedy Shannon - Kotělnikovův theorem. Tento zákon říká, že vzorkovací frekvence musí být alespoň dvakrát vyšší než frekvence nejvyšší složky která má být zpětně rekonstruována. My zde samozřejmě rekonstruovat signál nebudeme a pro nás je tato nejvyšší frekvenční složka informace o výstupu soustavy o kterou nesmíme přijít. Určuje ji časová konstanta soustavy, nalezená při identifikaci. Tuto konstantu můžeme na první pohled zkusmo určit z modelového průběhu prvního řádu nalezením tečny k exponenciále. To lze provést na průběhu 2.2 a na 2.3. V místě kde tečna protne časovou osu odečteme časovou konstantu systému. Časová konstanta zprůměrovaná ze všech charakteristik vyjde 355ms. T_{vz} musí být zvolena nejvýše polovina této hodnoty. Pro snadnější převody naměřených hodnot byla zvolena $T_{vz} = 0,1s$ tedy $f_{vz} = 10Hz$. Shannon - Kotělnikovův theorem je tedy splněn a vzorkované hodnoty správně zachytí dynamiku systému.

2.3.2 Návrh regulátoru metodou geometrického místa kořenů.

Tento druh návrhu spočívá ve vhodné volbě polohy pólů, nul a zesílení regulátoru. Hledání této polohy je výpočetně zdoluhavá úloha a tedy je více než vhodné použít utilitu Matlabu Sisotool. Sisotool na základě znalosti pólů a nul v otevřené smyčce pro navržený regulátor určuje polohu pólů a nul uzavřené smyčky. Jak už z názvu (SISO z angl. single input and single output) vyplývá utilita slouží k návrhu regulátoru a pozorování průběhů stěžejních veličin pro systémy s jedním vstupem a jedním výstupem. S touto podporou je návrh relativně snadný a řídí se pravidly popsanými v [2]. Je žádoucí aby ustálená odchylka výstupní veličiny byla nulová, proto regulátor bude obsahovat integrační složku. Jelikož informace o výstupu soustavy je zatížena šumem a kolísá, případné zařazení derivační složky musí být provedeno tak aby tento šum nebyl příčinou nestability. Při provádění experimentů s regulátory derivační složka regulaci zásadně neovlivnila a spíše byla příčinou problémů se stabilitou systému, proto nebyla použita. Regulátor bude hledán spojité a posléze diskretizován. Dále je nutné uvažovat omezený akční zásah daný číselně jako největší šířka kladného pulsu modulace 255. Tato vlastnost se projeví zpomalením regulace oproti simulaci v Sisotool, kde omezení akční veličiny není uvažováno. Regulátor byl již navrhován s vědomím nutnosti použití celočíselné aritmetiky a proto jsou konstanty vhodně zaokrouhleny.



Obrázek 2.5: Schema pro ověření návrhu regulátoru v simulinku

Přenos PI regulátoru je

$$C(s) = k_P + \frac{k_I}{s} = \frac{k_P}{s} \left(s + \frac{k_I}{k_P} \right) \quad (2.9)$$

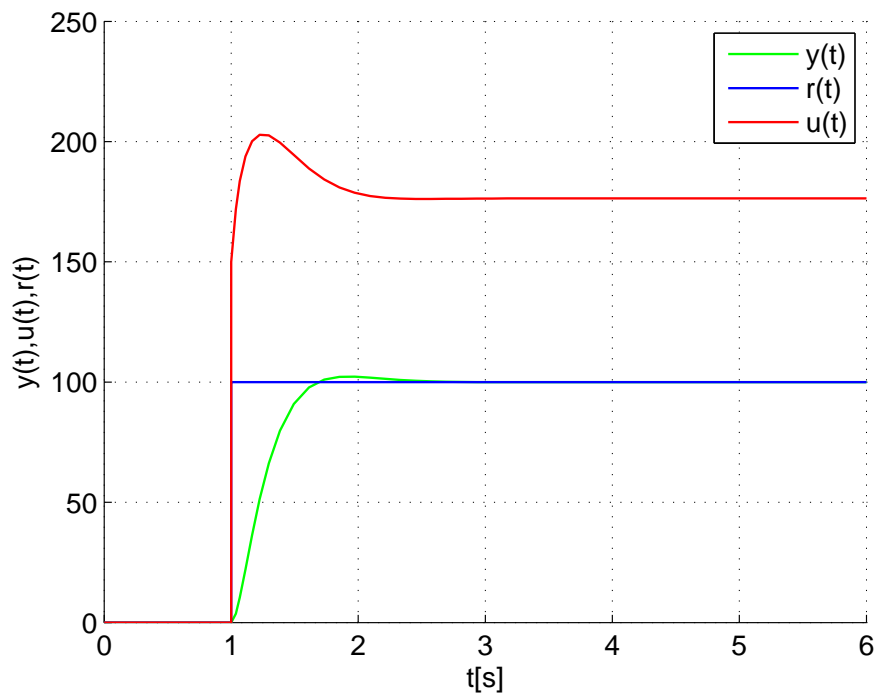
s konkrétními čísly

$$C(s) = 0,002 + \frac{0,001}{s} = \frac{0,002 \cdot (s + 0,5)}{s} \quad (2.10)$$

2.3.3 Diskretizace regulátoru

Navržený regulátor je nutno diskretizovat aby jej bylo možno implementovat. Diskretizace byla provedena pomocí metodou ZOH. Postup získání diskrétního přenosu je převod do spojité oblasti zpětnou Laplasovou transformací a poté provedení Z-transformace. Konstanty jsou pak přepočteny na základě linearit transformace spojitého přenosu do diskrétní oblasti. PID regulátor je tvořen třemi složkami, které se vzájemně sčítají. V případech lineárních transformací platí princip superpozice a tedy lze diskretizovat každou dílčí složku regulátoru zvlášť a poté sečíst.

$$C(s) = k_P + k_D s + \frac{k_I}{s} \quad (2.11)$$

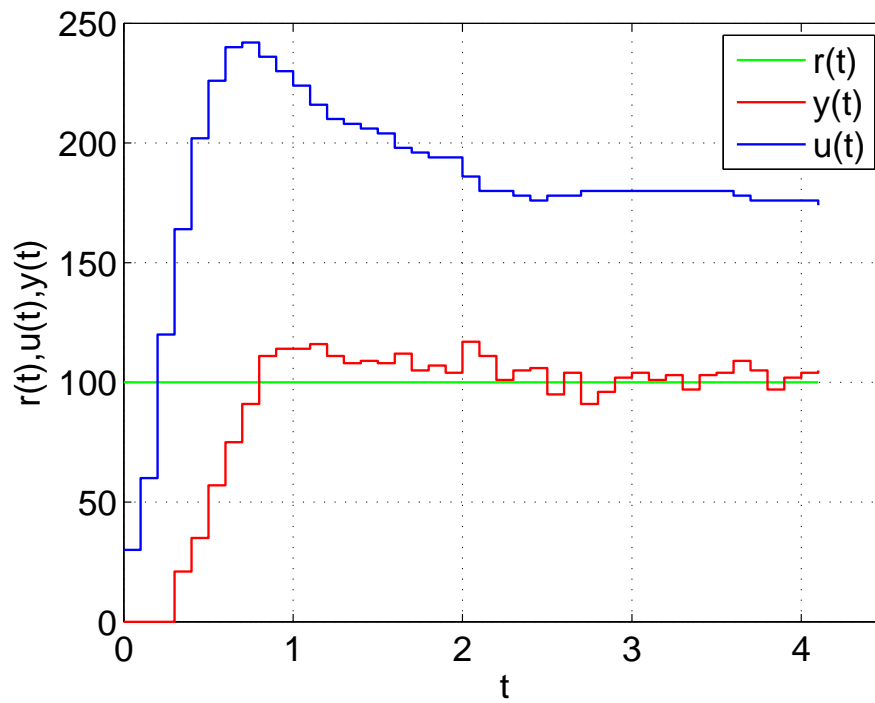


Obrázek 2.6: Simulovaná odezva vlaku s PI regulátorem

Proporcionální konstanta je shodná ve spojitě i v diskrétní oblasti. Přenos D složky s filtrací $C_D(s) = k_D \cdot \frac{s}{s+p_c}$ je diskretizován s $T_{vz} = 0,1s$ jako $D_D(z) = k_D \cdot \frac{z-1}{z-p_d}$ a tedy konstanta D složky zůstane nezměněna. Přenos integrátoru $C_I(s) = \frac{k_I}{s}$ je diskretizován jako $D_I(z) = k_I \cdot \frac{0,1}{z-1}$ a konstanta integrátoru je násobena 0,1. Konstanty pro diskrétní regulátor s $T_{vz} = 0,1s$ tedy jsou $k_P = 0,002$, $k_I = 0,001$.

2.3.4 Windup

Jev navíjení, jak zní český ekvivalent je způsoben konečnou hodnotou akčního zásahu a přítomností integrační složky při regulaci. V případě, že akční veličina zasahuje do systému svou maximální hodnotou a navzdory tomu je stále kladná regulační odchylka (tedy žádaná veličina je vyšší než maximální dosažitelná) nastává Windup. Dochází totiž k integrování regulační odchylky a následná reakce na pokles žádané hodnoty je opožděna, protože naintegrovaná hodnota způsobuje stále maximální akční zásah. Problém lze řešit softwarově v regulátoru omezením integrační složky při maximálním akčním zásahu. Ukázka regulace kde se tento jev podařilo eliminovat je v sekci 3.2.1.



Obrázek 2.7: Naměřená odezva lokomotivy při skoku na $r(k) = 100 \text{int}/T_{vz} \equiv (294 \text{mm} \cdot \text{s}^{-1})$

2.4 Realizace regulátoru

2.4.1 Generování PWM (akční veličina)

PWM (Pulse Width Modulation) česky pulsně šířková modulace je oblíbený nástroj na realizaci přesnějšího řízení otáček stejnosměrných motorů. Princip spočívá v napájení motoru impulsy proměnné šířky, jejichž střída určuje výsledné otáčky motoru. Výhoda této metody je zejména v tom, že moment motoru je kvadraticky závislý na napětí kotvy. Pokud tedy budeme otáčky řídit velikostí napětí bude s druhou mocninou klesat moment což je nežádoucí. K realizaci je zapotřebí výkonové části která bude zpracovávat tuto modulaci. Tímto řešením je integrovaný obvod L293. ATmega8 nabízí způsob generování těchto impulsů, pomocí Čítače/časovače č.2. Základní funkce jak již z názvu vyplývá je funkce standardního 8bitového čítače, avšak tento je doplněn jednotkou porovnání (compare match unit) a jí příslušejícím registrem OCR (output compare register). Nastavením hodnoty registru OCR definujeme okamžik spádové hrany výstupu uvnitř periody čítání. Registr TCCR (Timer/Counter control register) slouží jednak k nastavení volby dělicího poměru základního kmitočtu oscilátoru, ale především k volbě módu výstupu (WGM - waveform generation mode). Na výběr je Fast PWM, což je pulsně šířková modulace na vysoké frekvenci, která je vhodná především pro řízení napájecích zdrojů. Phase Correct PWM, která pracuje na nízkých frekvencích a mnohem lépe se hodí pro řízení aplikací stejnosměrných motorů.

2.4.2 Implementace softwarového regulátoru

Mikroprocesory AVR mají podporu aritmetiky s plovoucí řádovou čárkou, ale z hlediska ušetření výpočetního výkonu a strojového času byla pro regulátor použita celočíselná aritmetika. Díky tomu bylo nutno zavést několik zjednodušení a vypočtené konstanty zaokrouhlit. Jelikož regulační konstanty vychází zpravidla jako necelá čísla, bylo zavedeno namísto násobení těmito konstantami celočíselné dělení jejich převrácenou hodnotou.

Přímý návod jak implementovat regulátor jako program je zřetelný z diferenčních rovnic diskrétních regulátorů. Veličiny jsou značeny jako v [2], tedy $u(k)$ akční zásah, $e(k)$ regulační odchylka.

$$u(k) = K \cdot e(k) \quad (2.12)$$

$$u(k) = u(k - 1) + \frac{K}{T_I} \cdot e(k) \quad (2.13)$$

$$u(k) = K \cdot T_D \cdot [e(k) - e(k - 1)] \quad (2.14)$$

2.4.3 Pásmo necitlivosti

Na počátku převodní charakteristiky (rychlost jako funkce šířky PWM) je patrné pásmo, kdy akční veličina vlivem statického tření nedokáže odtrhnout lokomotivu. viz 2.4 Toto pásmo se po použití regulátoru zúží, ale stále bude existovat interval žádaných hodnot kterých nelze dosáhnout. Je to způsobeno příliš nízkou hodnotou rychlosti, pro kterou senzor poskytne nízký počet přerušení. Šum, který je přítomen nízkou hodnotu mnohem více zkreslí. Při nízké rychlosti lokomotivy se nezanedbatelně a nepředvídatelně projevuje suché tření a nerovnosti kolejí, které je za vyšších rychlostí zanedbatelné.

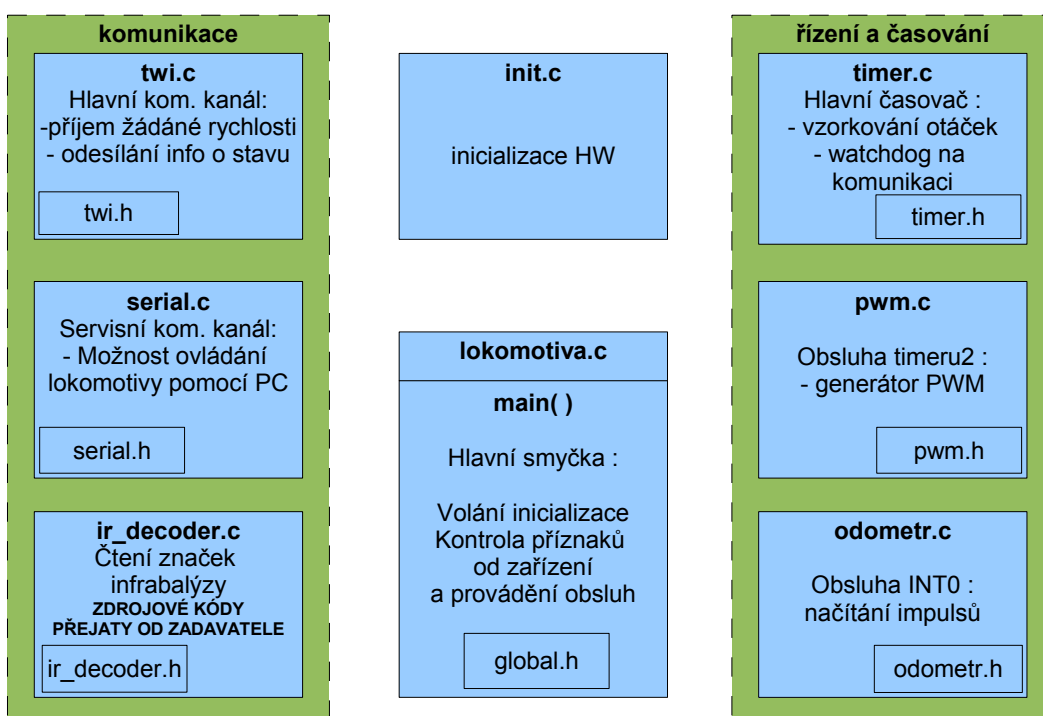
2.4.4 Optický IR snímač (zdroj informace o výstupu soustavy)

Lokomotiva byla převzata se snímačem, který poskytoval 1 změnu logické úrovně během otáčky setrvačnicku. Na stěnu motorku je připevněn fototranzistor a IR dioda, namířeny na podstavu válce setrvačnicku, kde je polovina obarvena pohltivým lakem. Jelikož požadavek na provoz lokomotivy směřuje spíše k provozu v malých rychlostech, bylo třeba zvýšit počet změn logické úrovně na jednu otáčku. Na setrvačnicku byla obarvena modelářskou barvou odrazivá plocha a vytvořeno 5 odrazivých a 5 neodrazivých oblastí. Limitující omezení počtu odrazivých ploch, které bylo nutno vzít v úvahu jsou dána jednak počtem impulsů na jednu otáčku setrvačnicku za jednotku času a to vzhledem k frekvenci procesoru, který tento signál zpracovává. Nesmí nastat situace, že zpracování obsluhy přerušení bude zabírat nezanedbatelnou část strojového času vůči obsluze ostatních periférií a řídicích smyček softwaru. Další omezení je velikostí použitého datového typu INTEGER, který je zde 2^{16} a tedy maximální hodnota je 65535. Toto omezení se v aplikaci ukázalo nepodstatné.

2.5 Popis implementovaného SW

Úloha programování se rozpadla na dvě propojené části, jednak obsáhlejší software lokomotivy a za druhé implementace komunikace protokolem TWI do převzatého programu na obsluhu Bluetooth komunikace.

Řídicí software lokomotivy je rozdělen na samostatné funkční bloky volané podle potřeby v hlavní smyčce nebo za pomoci přerušovacího mechanismu.



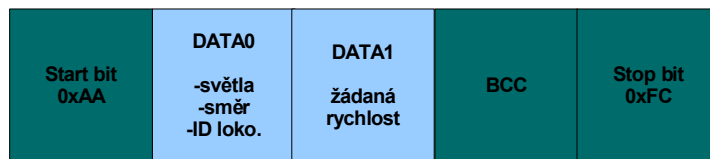
Obrázek 2.8: Blokové schéma řídicího software

2.5.1 Hlavní smyčka

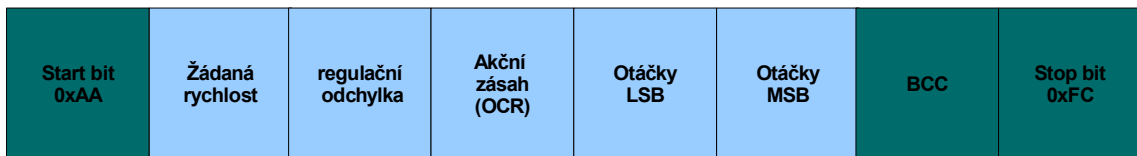
V hlavní smyčce programu se kontrolují příznaky událostí jako jsou přijatá data po seriové lince, přijatá data TWI, uplynutí doby pro vzorkování od časovače, připravená data pro odeslání, přijatá data od infrabalízy. Při kladném příznaku je událost obsloužena příslušnou procedurou.

2.5.2 Seriová komunikace

Seriová linka v lokomotivě slouží pouze jako servisní komunikační kanál. Lokomotivu lze plně ovládat, ale nevýhodou je nutnost připojení na kabel. Této části programu bylo používáno především při ladění regulátoru a experimentů s lokomotivou. Následuje schematický náčrt paketů používaných v seriové komunikaci. Každé pole znamená 1 byte.



Obrázek 2.9: Paket odesílaný z PC do lokomotivy



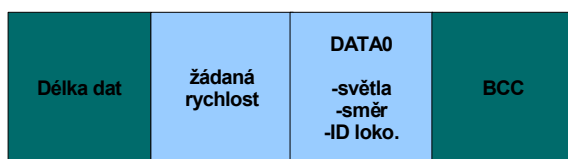
Obrázek 2.10: Paket odesílaný lokomotivou do monitorovacího SW

2.5.3 Infrabalýza

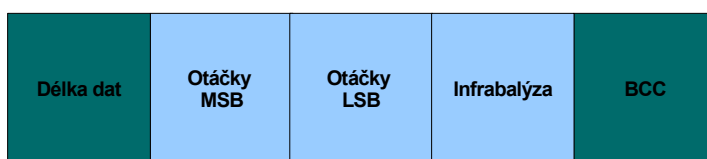
Zdrojové kódy od této části programu byly již vytvořené a byly přejaty. Význam infrabalýzy spočívá v předání identifikačního kódu infravysílače řídicímu serveru, který na základě znalosti identifikačních kódů vysílačů a jejich poloze v kolejišti může upravovat své výpočty.

2.5.4 TWI

Hlavní řídicí komunikace lokomotivy. Zařízení na sběrnici mohou pracovat v zásadě jako master nebo slave a přijímač nebo vysílač. Master je zařízení, které iniciuje komunikaci vysláním start sekvence a po celou dobu komunikace udržuje hodinový signál. V našem případě byl zvolen master modul s bluetooth vysílačem a slave lokomotiva. Jelikož je zapotřebí data přenášet obousměrně, master pracuje nejdříve jako vysílač a poté jako přijímač. Spojení se tedy navazuje pouze jednou ze strany mastera. Řešení je provedeno formou stavového automatu s využitím přerušení. Pokud na hardware dokončí běžnou obsluhu sběrnice nastaví příslušnou hodnotu do status registru (TWSR) a vyvolá přerušení. Toto přerušení je obslouženo a na základě hodnoty status registru, která určuje stav automatu je provedena reakce na nastalou situaci.



Obrázek 2.11: Paket odesílaný ze serveru přes Bluetooth do lokomotivy



Obrázek 2.12: Paket odesílaný z lokomotivy zpět serveru

TWI komunikace je řešena s ohledem na bezpečnost, kterou by mohlo způsobit uváznutí programu. Během této obsluhy je plně využíváno přerušení a uvnitř obslužné rutiny není žádná čekací smyčka, kde by program mohl při chybě v komunikaci uváznout.

2.5.5 Zabezpečení dat komunikace BCC

Standardní mechanismus kontroly platnosti dat, který umožňuje identifikovat neplatná data. Na vysílací straně se vypočte kontrolní XOR přes celou datovou část a uloží se do paketu na následující pozici. Na přijímací straně je opět vypočítán kontrolní XOR a porovnán se záznamem uloženým v paketu. Pokud nesouhlasí data jsou neplatná a při komunikaci došlo k chybě a celý paket se zahazuje.

2.5.6 Timer

Časovač v mikropočítači označen jako timer0 generuje přerušení každou 1 ms. V obsluze přerušení je prováděn odměr 100ms pro odečtení počtu přerušení a odměr 1s pro případ ztráty komunikace. Přerušení načítá hodnotu watchdog proměnné, které je nulována příchodem platných dat. Pokud by komunikace uvázla nebo se přerušila dojde po 1s k přetečení této proměnné a zavolání obsluhy pro zastavení vlaku. Volání obsluhy zastavení vlaku se provádí uvnitř rutiny přerušení, neboť jde zásadní a důležitou operaci.

2.5.7 Odometr

Realizuje obsluhu přerušení INT0, které je spouštěno změnou logického stavu na optickém snímači otáček setrvačnicku. Obsluha tohoto přerušní spočívá v pouhé inkrementaci proměnné, která je jednou za vzorkovací periodu vynulována.

2.5.8 PWM

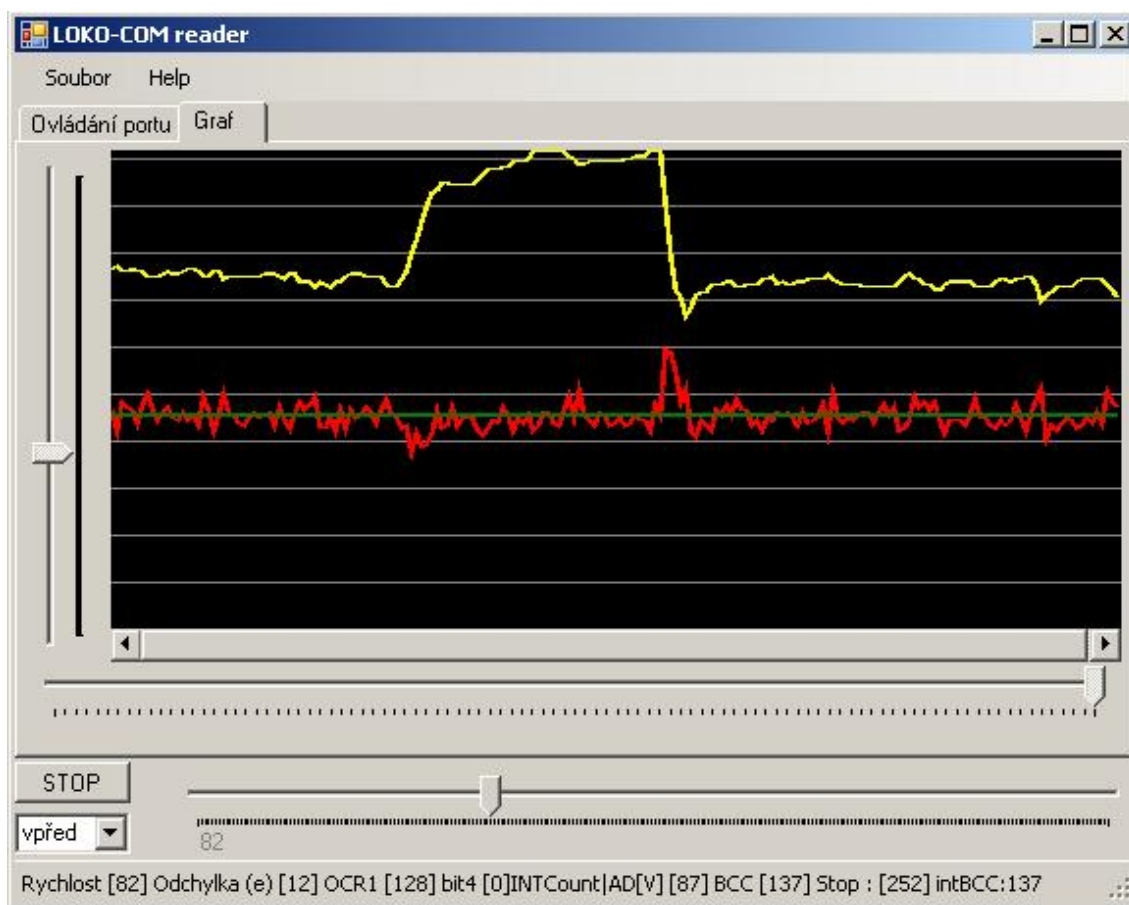
Zde probíhá výpočet nového akčního zásahu na základě regulačních konstant a vypočtené regulační odchylky. Řízení motoru bylo navrženo jednokvadrantové neboť je jednoduché a k řízení malého modelu, kde není třeba řídit brzdění či rekupraci naprosto dostačuje. Pro generování pulsně šířkové modulace je použit timer1, který má k dispozici dva kanály pro PWM. Každý kanál slouží pro jeden směr otáček motoru. Metoda výpočtu je popsána v sekci 2.4.2.

Kapitola 3

Ověření funkce a vyhodnocení parametrů

3.1 Monitorovací SW

Jelikož je nutné pozorovat výsledek návrhu regulátoru a vyhodnotit parametry a chování musel být do SW v lokomotivě zaimplementován způsob jak předávat data do PC. Hlavními sledovanými daty jsou otáčky setrvačnicku (výstupní veličina), šířka PWM (akční veličina) a žádaná hodnota otáček. Pro potřeby testování regulátoru jsem využil rozhraní RS232 a vytvořil si aplikaci zachytávající data, která procesor pravidelně odesílá. Za vývojové prostředí bylo zvoleno Microsoft Visual Studio a jazyk C#. V aplikaci je i možnost celou lokomotivu ovládat odesíláním řídicích dat. Monitorovací SW byl vytvořen pro potřeby návrhu regulátoru a tedy je tomuto účelu přizpůsoben. Zaznamenané průběhy lze ukládat do souboru a uložené průběhy později otevřít. Největší předností grafického vykreslení zaznamenaných dat je i možnost sledovat průběh v reálném čase a tak přímo pozorovat regulační proces. Z rastru mřížky lze hrubě odečítat hodnoty měřených veličin. Pro podrobnější rozbor dat (ne však již v reálném čase) aplikace vygeneruje M-file pro Matlab. Na následující obrázku je vidět experiment s regulátorem. Lokomotiva byla podložena a kola se nedotýkala podložky. Spuštěná regulace sledovala žádanou hodnotu a přibližně uprostřed průběhu byl setrvačnick přibrzděn prstem. To je doprovázeno zřetelným zvýšením akčního zásahu a mírným poklesem otáček. Při náhlém odlehčení setrvačnicku opět dochází k vyrovnání otáček. Tento servisní program však nebyl přímo předmětem této práce, ale pouze ji pomohl realizovat. Nezabýváme se zde proto jeho strukturou a řešením.



Obrázek 3.1: Screenshot aplikace;

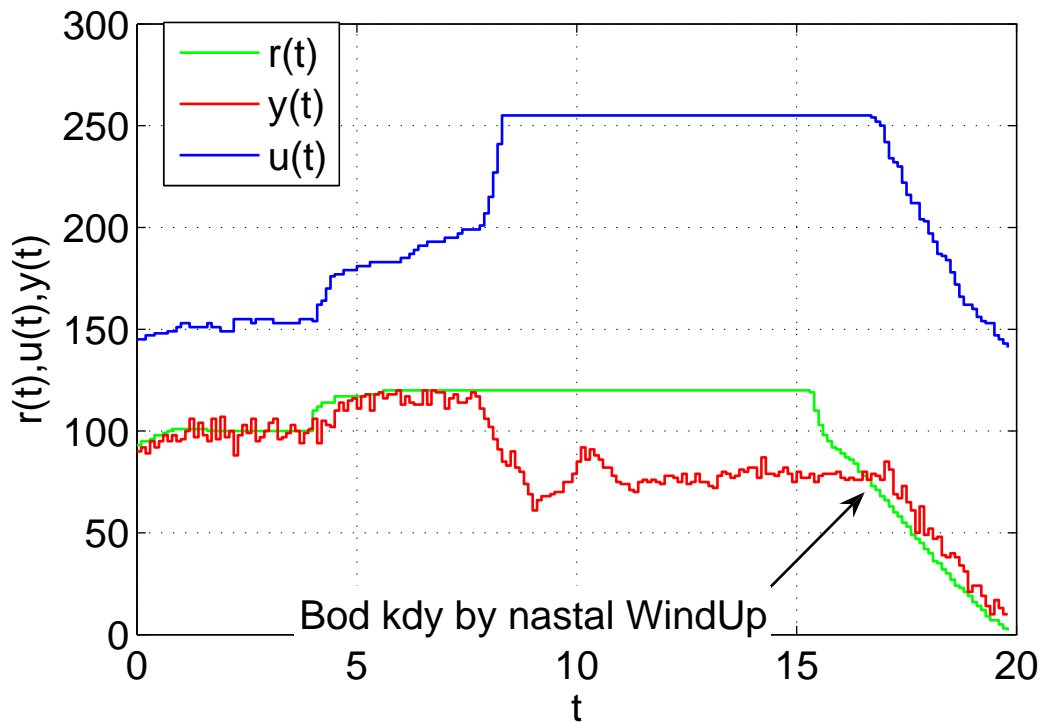
3.1.1 Komunikační protokol

Struktura dat odesílaných z PC a z lokomotivy se nemění, popsána již byla v kapitole 2.5.2. Aplikace řídicí data odesílá pouze pokud chceme zároveň lokomotivu ovládat v opačném případě data pouze zachytává. Této vlastnosti bylo využito při odladění TWI komunikace kdy tato aplikace sloužila pouze jako pozorovací nástroj.

3.2 Experimenty

3.2.1 Regulace bez jevu windup

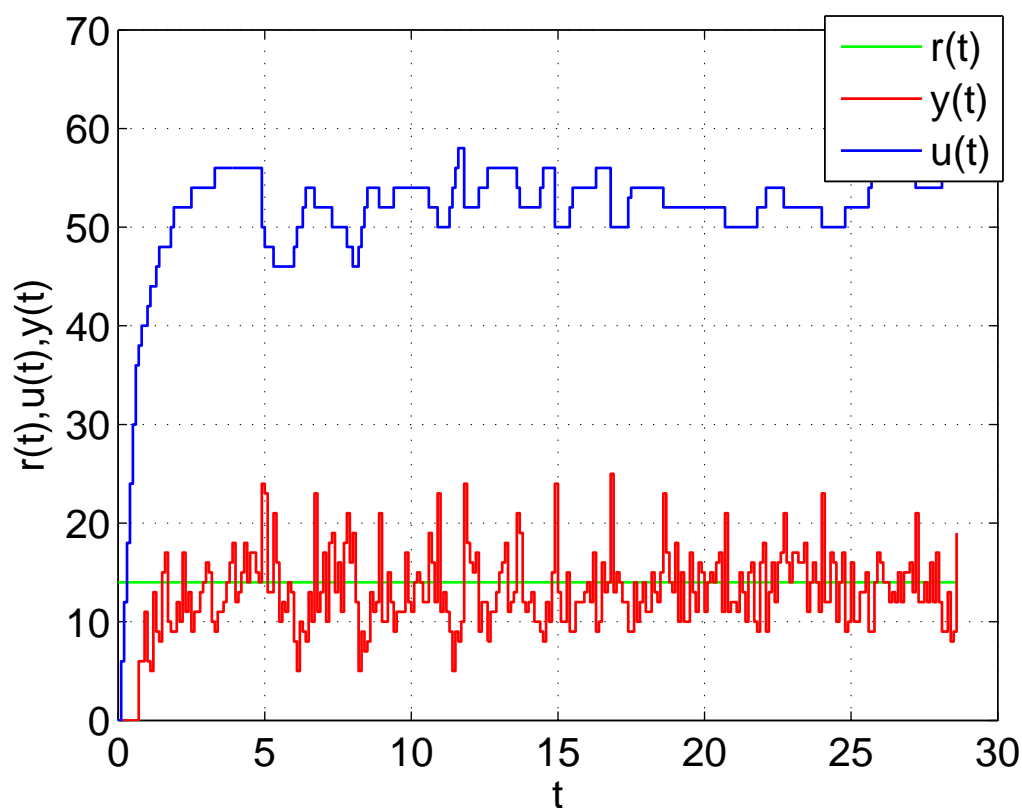
Jak již bylo zmíněno v kapitole 2.3.4, windup je jev, který může způsobovat nepředvídatelné chování modelu. Na následujícím grafu je pozorovatelné, jak došlo dosažení maxima akčního zásahu (modrý průběh), ale výstup (červený průběh) nedosáhl žádané hodnoty (zelený průběh). V okamžiku snižování žádané hodnoty, kdy se žádaná hodnota s průběhem výstupu rovnají, musí regulátor být shopen zareagovat ihned. Z průběhů je vidět, že regulátor reaguje ihned a windup nenastává.



Obrázek 3.2: Ukázka regulace bez jevu windup.

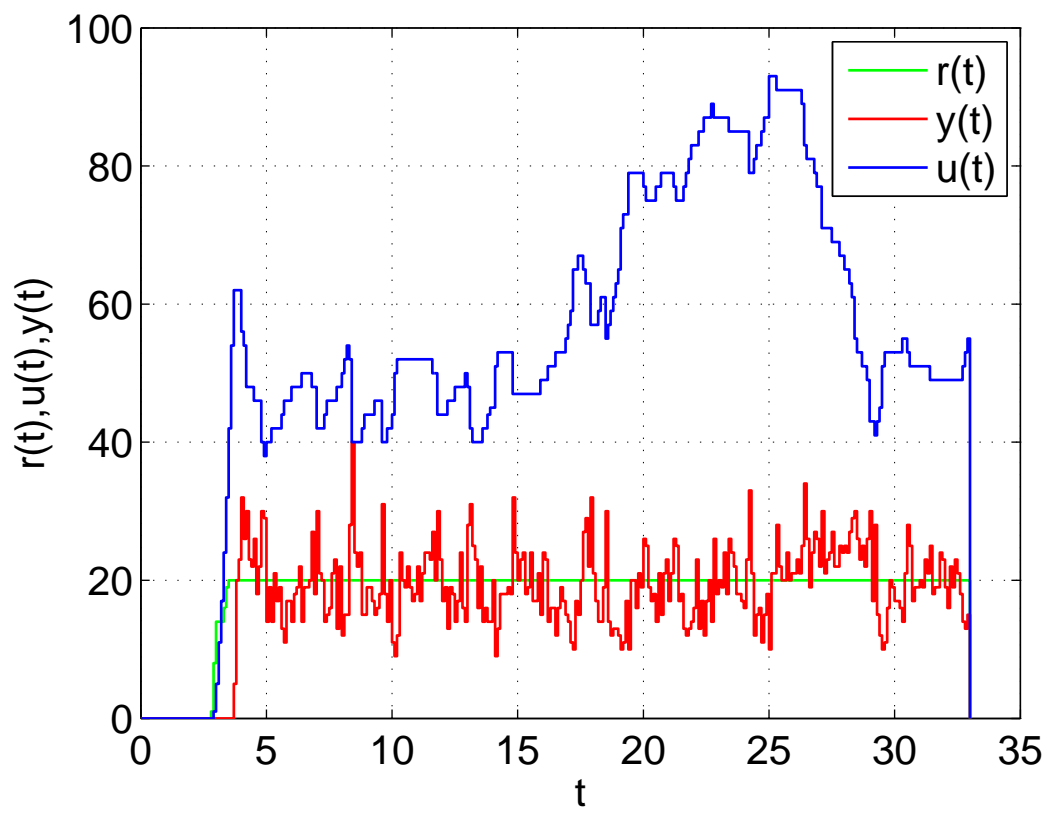
3.2.2 Regulace na pomalou rychlost

Experiment byl proveden na rovném úseku kolejí bez převýšení. Žádaná hodnota byla nastavena na 14 přerušení za $T_{vz} = 100ms$ což odpovídá $41mm.s^{-1}$. Tato hodnota je dosažena experimentálně za modelově nejjednosušších podmínek, a proto ji nelze považovat za spolehlivou.

Obrázek 3.3: Ukázka regulace na $41\text{mm}\cdot\text{s}^{-1}$.

3.2.3 Experiment s pomalou jízdou ve stoupání.

Experiment byl proveden rovném úseku kolejí s převýšením 60mm na dráze 500mm. Požadovaná rychlost byla 20 přerušení $T_{vz} = 100\text{ms}$ což odpovídá $59\text{mm}\cdot\text{s}^{-1}$. Na průběhu akčního zásahu je zřetelně vidět nutnost jeho zvýšení ve stoupání a opětovný pokles na rovině.



Obrázek 3.4: Ukázka regulace do stoupání při rychlosti $59\text{mm}\cdot\text{s}^{-1}$.

Kapitola 4

Závěr

Během práce jsem se seznámil s prací s mikrokontrolery Atmel ATMega8(16). Do převzatého software pro obsluhu Bluetooth zařízení byla doimplementována část pro TWI komunikaci s ohledem na riziko uváznutí programu. Jako nedostatek lze označit nízkou prioritu přerušení od TWI zařízení. V těchto mikroprocesorech však není priorita přerušení programovatelná a proto tento nedostatek nebyl nikterak ošetřen. Při experimentech ke ztrátě komunikace nedocházelo. V případě výpadku komunikace lokomotiva za 1s začne brzdit a zastaví.

Lokomotiva jako soustava pro řízení byla identifikována pomocí přechodových charakteristik při brzdění a to pro tři různé rychlosti. Výsledný model byl vypočten průměrováním časových konstant. Identifikace proběhla pomocí měření nezávislého na hardware lokomotivy. Systém se dle očekávání ukázal jako stabilní a byl namodelován jako soustava druhého řádu.

Navržený regulátor je schopen regulovat na žádanou hodnotu od $60\text{mm}\cdot\text{s}^{-1}$ bez zásadního vlivu na charakter trati. Byla prakticky ověřena funkčnost ošetření proti jevu windup. Pásmo necitlivosti bylo softwarově ošetřeno. Pro rychlosti pod prahem dosažitelných hodnot je žádaná hodnota nejnižší možná dosažitelná. Zařazení D složky do regulátoru nikterak nepřineslo zásadní zlepšení regulace a proto v zájmu co nejkratších výpočtů nebyla D složka použita pro řízení. Pro generování PWM bylo použito vestavěných PWM kanálů jednoho z časovačů.

Pro pozorování systému a experimenty při regulaci byl vytvořen program pro Windows. Umožňuje plné řízení lokomotivy oběma směry. Velkou předností je možnost pozorovat průběhy žádané hodnoty, akčního zásahu a výstupu soustavy ihned po předání jednoho vzorku tedy téměř v reálném čase. Další zjednodušení práce s naměřenými průběhy přináší možnost generování m-file pro Matlab. Naměřené hodnoty lze ukládat do souboru

na pevný disk a programem prohlížet i dříve pořízené.

Možnost řízení pomocí měření napětí v neaktivní části PWM byla prověřena a rutina měřící napětí v neaktivní části periody PWM implementována. Naměřené hodnoty však neodpovídaly předpokladům a následně bylo zjištěno že obvod L293 neumožní měření napětí na motoru. Nelze jednoduše dosáhnout stavu vysoké impedace pouze na jednom výstupu a stavu L na druhém proti kterému by se měřilo.

Literatura

- [1] *Roubal, J., Hušek, P. a kol. Základy Regulační techniky*
(pdf) <http://dce.felk.cvut.cz/roubal>

- [2] *Franklin Gene F. ,Powell David J. ,Naeini Abbas Emami. Feedback control of dynamic systems*

- [3] *Šebek, M. Přednášky k předmětu SRI 2006/07*

- [4] *Atmel Katalogový list procesoru ATMega8*

- [5] **Bluetooth 11 specification book**

- [6] *Šusta, R. Materiály k přednáškám a cvičením předmětu X35PJR*

Příloha A

Seznam použitého software

- AVR Studio 4 - software dodávaný k vývojovému kitu STK500.
- Microsoft Visual Studio 8
- Image Craft - kompilátor jazyka C
- Mathworks Matlab R2006b (7.3.0)
- MV2 player - k uložení plain .avi po snímcích do formátu .jpeg
- TeXnicCenter - správa projektu v L^AT_EXu
- MikTeX 2.6 - česká lokalizace L^AT_EXu
- GhostScript, GhostView - práce s obrázky ve formátu .eps, konverze do .pdf pro pdfLaTeX
- OpenOffice.org Drawing - tvorba blokových schemat a jejich konverze do .pdf

Příloha B

Obsah přiloženého CD

K této práci je přiloženo CD, na kterém jsou uloženy zdrojové kódy.

- Složka se zdrojovými kódy lokomotivy
- Složka s pomocnou aplikací k řízení lokomotivy
- Složka se záznamy porízeními při identifikaci
- Složka s některými záznamy při experimentech