



Diplomová práce

Pokročilý algoritmus řízení balancujícího robota

Studijní program:

N0714A270010 Mechatronika

Autor práce:

Bc. Radek Kulhavý

Vedoucí práce:

Ing. Lukáš Hubka, Ph.D.

Ústav mechatroniky a technické informatiky

Liberec 2024



Zadání diplomové práce

Pokročilý algoritmus řízení balancujícího robota

Jméno a příjmení:

Bc. Radek Kulhavý

Osobní číslo:

M21000179

Studijní program:

N0714A270010 Mechatronika

Zadávací katedra:

Ústav mechatroniky a technické informatiky

Akademický rok:

2022/2023

Zásady pro vypracování:

1. Seznamte se s aktuální konstrukcí, HW, SW a řízením balancujícího robota.
2. Sestavte vhodný simulační model robota, parametrizujte jej podle reálného zařízení a validujte jej.
3. Vytvořte řídicí algoritmus pro stabilizaci i kontrolovanou jízdu robota včetně zatáčení. Algoritmus založte na prediktivním řízení nebo robustním loop-shaping přístupu.
4. Po simulačním ověření implementujte a ověřte řízení na reálném stroji.

Rozsah grafických prací: dle potřeby dokumentace
Rozsah pracovní zprávy: 40-50 stran
Forma zpracování práce: tištěná/elektronická
Jazyk práce: čeština

Seznam odborné literatury:

- [1] MAMULA, Petr. Stabilizace a navádění balancujícího robota. Liberec, 2021. Diplomová práce. TUL.
- [2] BIN He, Liu Wen ZHEN, Hai FENG. The Kinematics Model of a Two-Wheeled Self-Balancing Autonomous Mobile Robot and Its Simulation. 2010 Second International Conference on Computer Engineering and Applications. IEEE, 2010, s. 64-68. ISBN 978-1-4244-6079-3. Dostupné z: doi:10.1109/ICCEA.2010.169.
- [3] XU, Changkai, Ming LI a Fangyu PAN. The system design and LQR control of a two-wheels self-balancing mobile robot. In: 2011 International Conference on Electrical and Control Engineering, IEEE, 2011, s. 2786-2789. ISBN 978-1-4244-8162-0. Dostupné z: doi: 10.1109/ICECENG.2011.6057680.
- [4] Hu, J., Tsai, M. Control of an auto-balancing two-wheeled cart. Proceedings of the SICE Annual Conference. 2005.
- [5] WANG, Fu-Cheng a Zih-Jia WANG. The Development of a Multi-Loop Control Structure for a Two-Wheeled Inverted Pendulum Robot. In: 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT) [online]. IEEE, 2019, 2019, s. 551-556 [cit. 2022-10-11]. ISBN 978-1-7281-0521-5. Dostupné z: doi:10.1109/CoDIT.2019.8820509.

Vedoucí práce: Ing. Lukáš Hubka, Ph.D.
Ústav mechatroniky a technické informatiky

Datum zadání práce: 12. října 2022
Předpokládaný termín odevzdání: 15. května 2024

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

doc. Dr. Ing. Jaroslav Hlava
garant studijního programu

V Liberci dne 12. října 2022

Prohlášení

Prohlašuji, že svou diplomovou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Jsem si vědom toho, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má diplomová práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

Pokročilý algoritmus řízení balancujícího robota

Abstrakt

V diplomové práci je popsán aktuálně použitý hardware a software balancujícího robota. Byl sestaven simulační model balancujícího robota, který byl parametrizován podle reálného zařízení. Následně byl vytvořen řídicí algoritmus pro stabilizaci a jízdu robota včetně zatočení. Řízení bylo testováno v programu Matlab Simulink a následně implementováno na reálné zařízení. Při implementaci na reálné zařízení se objevily problémy s nepřesným umístěním a výpadky snímače náklonu robota.

Klíčová slova: prediktivní řízení, balancující robot, inverzní kyvadlo, Matlab

Abstract

The diploma thesis describes the currently used hardware and software of the balancing robot. A simulation model of the balancing robot was created, which was parameterized according to the real device. Subsequently, a control algorithm was created for stabilizing and driving the robot, including turning. The control was tested in the Matlab Simulink program and then implemented on a real device. When implemented on a real device, there were problems with inaccurate placement and crashes of robot tilt sensor.

Keywords: predictive control, balancing robot, inverted pendulum, Matlab

Poděkování

Tímto bych rád poděkoval Ing. Lukáši Hubkovi, Ph.D. za vedení mé diplomové práce a poskytnutí přístupu k balancujícímu robotu.

Obsah

Úvod	12
1 Aktuální konstrukce HW a SW balancujícího robota	13
1.1 Hardware balancujícího robota	14
1.1.1 Raspberry Pi	14
1.1.2 Wemos D1	15
1.1.3 MPU-6050	15
1.1.4 DC motor	16
1.2 Současný a použitý software	16
1.2.1 Matlab	16
1.2.2 Raspberry Pi	17
1.2.3 Komunikace robota s okolím	18
2 Řízení za pomoci prediktivního regulátoru	19
2.0.1 Výhody MPC	19
2.1 Princip MPC	19
2.1.1 Parametry MPC	20
2.1.2 Účelová funkce	20
2.1.3 Omezení v MPC	21
2.2 MPC pro nelineární systém	22
2.2.1 Linearizace	22
2.2.2 Přepínání modelů	22
2.2.3 Hybridní přístup	23
2.3 Problematika robustnosti v MPC	23
2.3.1 Robustní MPC	23
2.3.2 Worst case přístup	24

2.3.3	TubeBased approach	24
3	Návrh modelu pro balancujícího robota	25
3.1	Popis dynamiky robota	25
3.1.1	Kola	26
3.1.2	Tělo robota	27
3.1.3	Vyjádření obecných souřadnic	28
3.2	Linearizace a tvorba stavového popisu	29
3.2.1	Decoupling	29
3.2.2	Stavový popis	30
3.3	Identifikace parametrů robota	31
3.4	Tvorba modelu v Matlabu	32
3.5	Návrh prediktivního regulátoru	32
4	Řízení	34
4.1	Kalibrace a měření MPU-6050	34
4.2	Řízení balančního robota	36
4.2.1	Subsystem Esp8266	36
4.2.2	Inverzní Decoupling matice	37
4.2.3	Ovladač Robota	37
5	Závěr	39
	Použitá literatura	40

Seznam obrázků

1.1	Balancující robot - pohled zepředu	13
1.2	Balancující robot - pohled z boku	14
1.3	Wemos D1 mini (zdroj: [4])	15
1.4	Kompaktní sada motoru JGA25-370 a kola robota	17
1.5	Příkaz pro připojení k Raspberry Pi 3B+	17
1.6	Příkaz pro zjištění IPv4	18
1.7	Příkaz pro změnu uživatelského hesla	18
2.1	Princip MPC regulátoru	20
3.1	Orientace balancujícího robota	26
3.2	Naměřené a použité parametry v Matlabu	31
3.3	Simulovaný průběh regulačního obvodu	32
3.4	Schéma pro testování nastavení MPC	32
4.1	Měření sklonu pomocí MPU-6050	35
4.2	Výstup ze sběrnice UART	36
4.3	Systém sloužící k logice zatáčení	37
4.4	Ovládání pojezdu a funkcí robota	38

Seznam tabulek

1.1	Parametry motoru JGA25-370 [Výrobce]	16
3.1	Označení použitých veličin	26

Seznam zkratek

DC Motor	Stejnoseměrný motor (Direct Current Motor)
MPC	Prediktivní řízení (Model predictive control)
PID	Proporcionálně integrační derivační regulátor
USB	Univerzální sériová sběrnice (Universal Serial Bus)

Úvod

Balancující roboty mají široké teoretické i praktické využití. Nové technologie v oblasti řízení jsou často testovány a používány právě na balancujících robotech. Speciálně balancující zařízení ve tvaru inverzního kyvadla jsou častou úlohou pro testování různých metod řízení a jejich výzev. Důvodem toho je několik vlastností které jsou obsaženy v této struktuře. Tyto vlastnosti umožňují velký průnik různých technologií, které je na jejich řízení možno aplikovat a jsou ideální k testování nových technologií, například problematika dvounohé chůze. Inverzní kyvadlo však nemá pouze jen teoretické využití, mnoho dopravních systémů využívá technologii založenou na tomto principu, která se navíc dočkala i komerční uplatnění v podobě známého vozítka od firmy Segway.

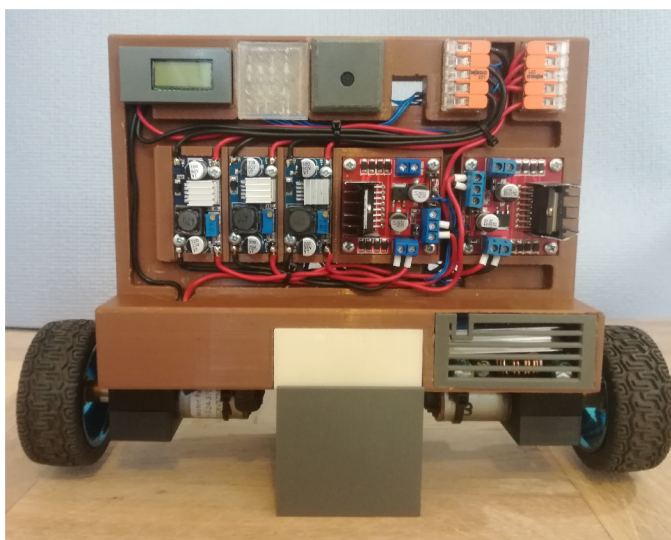
Prediktivní řízení je oproti tomu technologie, která je na vzestupu od jejich počátků okolo roku 1970. Jedná se o řízení založené na využití modelu k predikování chování systému v závislosti na sekvenci budoucích řídicích akcí. To sebou nese několik klíčových výhod, ale i nevýhod oproti jiným technologiím.

Cílem této práce je sestavení simulačního modelu existujícího balancujícího robota ve tvaru invertovaného kyvadla a aplikování regulátoru na principu prediktivního regulátoru k jeho stabilizaci a řízení. Řízení by poté mělo být implementováno a otestováno za využití reálného zařízení. V teoretické části jsou uvedeny významné prvky aktuální konstrukce reálného zařízení a dále je nastíněno prediktivní řízení. Praktická část se zabývá tvorbou modelu v prostředí Matlab Simulink a výzvami implementace řízení na reálném zařízení řízeném skrze RaspberryPi 3B+.

1 Aktuální konstrukce HW a SW balancujícího robota

Balancující robot je tvořen konstrukcí ve tvaru kvádru s dvěma koly. Konstrukce tohoto typu je často označována jako inverzní kyvadlo a je často využívána v oblastech robotiky a teorie řízení. V těchto oblastech má jak teoretické, tak i praktické využití. Balancující robot využit pro účely této práce byl postaven v [1]. Na obr 1 a 1 lze vidět reálnou konstrukci robota.

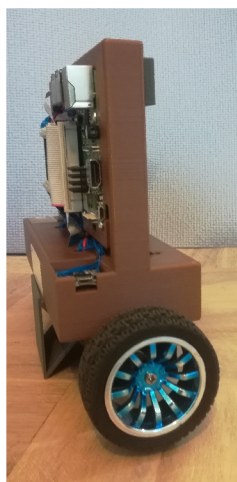
Teoretická hodnota tohoto typu konstrukce v oblasti řízení spočívá ve vlastnosti inverzního kyvadla jako nestabilní nelineární úlohy, kterou lze v hlavní oblasti linearizovat. To přináší výhodu, kdy je pro stejný systém možné použít a porovnávat velké množství různých strategií řízení. Zároveň se jedná o typ úlohy, jejíž řízení má uplatnění v praxi hlavně v oblasti balančních systémů při řešení úloh moderní přepravy a robotiky. Mimo řízené systémy nachází využití při konstrukci budov a pomůckám pro pohybově postižené. Komerční využití lze vidět například u dronové dopravy a u vozítka Segway [2].



Obrázek 1.1: Balancující robot - pohled zepředu

Jak lze vidět viz obr 1, tělo samotného robota není dokonalý kvádr. Podstava je širší než horní část a zároveň se v ní nachází velká část rozložení hmotnosti. Těžiště robota se nachází nad přelomem dvou částí. Lze tedy očekávat nepřesnosti v určení momentu setrvačnosti robota.

Na obrázku 1 lze v pravém horním rohu robota vidět ukazatel stavu baterie. Robot je dále vybaven kamerou, jejíž rozpoznávací schopnosti jsou ale dle [1] pro účely praktického použití mimo optimální podmínky nedostatečné. Kamera není z tohoto důvodu v práci použita.



Obrázek 1.2: Balancující robot - pohled z boku

1.1 Hardware balancujícího robota

Z hlediska hardwarových komponent jsou uvedeny ty, které jsou významné pro řízení balancujícího robota. Za významné jsou považovány především komponenty plnící funkce aktuátorů, komunikace s okolím a senzorů pro zpětnou vazbu.

1.1.1 Raspberry Pi

Primární řídicí jednotkou celého systému je jednodeskový počítač typu Raspberry Pi 3B+. Ten umožňuje komunikaci s okolím pomocí bezdrátového připojení skze

Wifi a Bluetooth nebo pevné připojení skrze 4 porty USB 2.0 a jeden 10/100 Mbps Ethernet port. Sít je zajištěna routerem N750 od firmy TP-Link.[3]

1.1.2 Wemos D1

Komunikace s ovládáním je zajištěna za pomoci vývojové desky Wemos D1 mini. Deska obsahuje modul ESP8266MOD, který slouží k propojení robota se sítí pomocí standardu 802.11. ESP8266MOD je dále připojen na enkodéry a slouží také k získávání dat o natočení kol. Deska i modul k programování využívají technologie Arduino. [esp, 1]



Obrázek 1.3: Wemos D1 mini (zdroj: [4])

1.1.3 MPU-6050

Zpětnou vazbu pro řízení v uzavřené smyčce zajišťuje modul MPU-6050 od firmy InvenSense. Jedná se o integrovaný senzor polohy fungující na principu spojení 3-osého gyroskopu s akcelerometrem na stejném čipu. Kombinace na této úrovni eliminuje nepřesnost způsobenou nesouosostí mezi gyroskopem a akcelerometrem. Modul má 4 možné nastavení rozsahů a citlivostí, která lze nastavit. Šum gyroskopu má tabulkovou hodnotu 0.005 mdps/rtHz. [MPU6050] Tuto hodnotu je nutné se jako nepřesnost pokusit potlačit vzhledem k očekávané nepřesnosti integrace hodnoty gyroskopu a skutečným nepřesnostem snímače.

1.1.4 DC motor

[htbp!] Jako pohon slouží dvojice kompaktních stejnosměrných motorů JGA25-370. Jedná se o kartáčový motor s vestavěným převodovým soukolím. Použit byl konkrétně motor s napětím 6V a převodovým poměrem $n=21,3$. Parametry jsou uvedeny níže (viz Tabulka 1.1) a jsou dostupné na stránkách výrobce [Výrobce]. Motor a kolo robota, které jsou součástí jedné sady je zobrazen na obr 1.4.

Jmenovité napětí U	6 V
Hmotnost m_w	92 g
Převodový poměr n	1:21.3
Volnoběžné otáčky	280 RPM
Volnoběžný proud	100 mA
Otáčky se zátěží	220 RPM
Moment se zátěží	49.02 mN m
Proud se zátěží	450 mA
Moment při zastavení	166,7 mN m
Proud při zastavení	1.3 A
Účinnost	0.65

Tabulka 1.1: Parametry motoru JGA25-370 [Výrobce]

1.2 Současný a použitý software

1.2.1 Matlab

Velká část práce je tvořena v programovém prostředí Matlab a jeho grafickém prostředí Matlab Simulink. To je zvoleno jak z několika praktických důvodů, tak z důvodu osobní zkušenosti s tímto prostředím. Základní prostředí Matlab je v práci využito ke tvorbě MPC regulátoru a k přehlednému uložení naměřených parametrů pro model robota a linearizovaný stavový popis.



Obrázek 1.4: Kompaktní sada motoru JGA25-370 a kola robota

Matlab Simulink je prostředí umožňující snadnou tvorbu modelů a simulaci regulačních systémů. V tomto prostředí byl vytvořen lineární model balancujícího robota. Dále zde byla provedena simulace a nastavení MPC regulátoru na nelineárním modelu. Jeho další velkou výhodou je v tomto případě nadstavba Matlabu Simulink umožňující propojení s platformami typu Raspberry Pi, kterou balancující robot obsahuje. Nadstavba má název „Simulink Support Package for Raspberry Pi Hardware” a je k nalezení jako oficiální add-on skrze interní databázi rozšíření programu Matlab. Připojení lze ověřit skrze příkaz 1.5.

```
raspi('IPv4', 'username', 'password')|
```

Obrázek 1.5: Příkaz pro připojení k Raspberry Pi 3B+

1.2.2 Raspberry Pi

Raspberry Pi 3B+ pracuje na operačním systému typu Linux, konkrétně jde o oficiální verzi Raspberry Pi OS (dříve Raspbian). Pro komunikaci Matlabu s Raspberry Pi 3B+ je třeba zajistit přístup přes protokol SecureShell (SSH), k čemuž je nutné zjistit IPv4, uživatelské jméno a heslo. Při přístupu počítače přes Ethernet lze využít

vlastnosti a nízkého zabezpečení operačního systému Linux. Z příkazové konzole lze zjistit uživatelské jméno. Pro IPv4 v konzoli existuje příkaz (Obrázek 1.6). Heslo pro SSH lze z konzole defaultně přenastavit pomocí příkazu níže (Obrázek 1.7).

```
<hostname> -I
```

Obrázek 1.6: Příkaz pro zjištění IPv4

```
sudo passwd <username>
```

Obrázek 1.7: Příkaz pro změnu uživatelského hesla

1.2.3 Komunikace robota s okolím

Jak je uvedeno v [1], robot primárně s okolím komunikuje pomocí Wifi sítě. Komunikace přes Bluetooth není vzhledem ke konstrukci umožněna a komunikace přes Raspberry Pi 3b+ ethernetovým kabelem je možná pouze pokud není robot aktivní. Komunikace přes Wifi síť probíhá dvěma způsoby. Tím prvním je připojení přímo k řídicímu programu v Raspberry Pi 3b+, například pomocí nadstavby Matlab či vzdáleného přístupu. Druhou možností je připojení k modulu ESP8266MOD, který je s Raspberry Pi 3B+ propojeno skrze sběrnici UART.

2 Řízení za pomoci prediktivního regulátoru

Technika řízení využívající dynamického modelu systému. Regulátor využívá model k vytvoření predikce chování systému při použití sekvence inputů a pomocí optimalizačního algoritmu vybere nejlepší sekvenci dle indexu výkonosti. V praxi se využívá zjednodušený dynamický model, který vede k predikování pravděpodobné predikce a výběru dostatečně dobré řídicí akce. Kvalita akčního zásahu je závislá na přesnosti dynamického modelu.

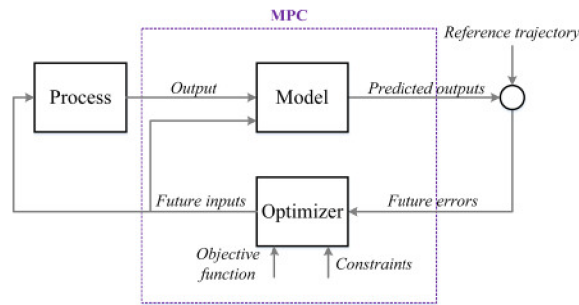
2.0.1 Výhody MPC

Při porovnání řízení pomocí MPC s další populární strategií, jako je například PID, má MPC několik užitečných výhod. Jednou z hlavních předností je zahrnutí omezení a účelové funkce do výpočtu řídicí akce. MPC je také při znalosti budoucího průběhu reference schopnost reagovat na změny v hodnotě reference s předstihem.

2.1 Princip MPC

Základní princip MPC regulátoru lze vidět znázorněn viz Obrázek (2.1). Regulátor pomocí modelu systému vytváří predikci hodnoty výstupní veličiny $y(t_n)$ a porovnává ji s předpokládanou žádanou hodnotou. Tím vznikají předpokládané budoucí odchylky $e(t_n)$ od žádané hodnoty regulované veličiny. Ze znalosti odchylek je poté za přítomnosti omezení pomocí účelové funkce nalezena budoucí hodnota akčního zásahu $u(t_{n+1})$, která je poté opět dosazena do modelu pro výpočet y_{n+1} . Tento proces je opakován, dokud není dle horizontu řízení nalezena sekvence o délce N_u optimálních vstupů, kde N_u označuje horizont řízení. [tandt]

Z takto výpočítané sekvence vstupů je poté použit pouze vstup $u(t_{n+1})$ a zbytek sekvence je nevyužit. Pro další krok znovu je vypočítána nová sekvence o délce N_u , a tím se horizont posouvá. To je označováno pojmem ustupující horizont (receding



Obrázek 2.1: Princip MPC regulátoru

horizon).

2.1.1 Parametry MPC

Základní parametry, které je nutné definovat při použití MPC, jsou horizont predikce a horizont řízení. Pro výpočet akčního zásahu je také nutné definovat účelovou funkci a různá omezení veličin a stavů. Jednotlivé parametry jsou přiblíženy níže v této kapitole.

- Horizont predikce označuje počet budoucích kroků, pro které je predikováno chování systému.
- Horizont řízení označuje počet budoucích kontrolních akcí, které má regulátor při optimalizaci k dispozici.

2.1.2 Účelová funkce

Také nazývána jako kriteriální funkce, slouží k výpočtu optimálních budoucích přírůstků akční veličiny. Funkce je minimalizována v každém kroku na základě aktuálních hodnot a predikovaných budoucích hodnot. Tvar účelové funkce se liší v závislosti na účelu MPC a zahrnutí měkkých omezení a přítomnosti nejistot. Základní tvar funkce pro sledování reference lze vidět v (2.1). Optimalizovanou funkci je možné vyjádřit jako problém kvadratického programování, který je po-

té řešen. [5][MPCD TandT]

$$\sum_{k=0}^{N-1} (y_k - r(t)) Q_y (y_k - r(t)) + \Delta u'_k k R \Delta u_k \quad (2.1)$$

2.1.3 Omezení v MPC

Omezení je několik typů a mohou být tvrdá nebo měkká. Tvrdá omezení neumožňují porušení daných hranic, což může při přespříliš konzervativním nastavení vést k nesplnění cílů řízení. Měkká omezení překročení hranic pouze penalizují při výpočtu účelové funkce. To vede ke kompromisu mezi překračováním omezení a performance. Nejběžnější omezení jsou akčního zásahu, výstupní veličiny, stavů systému a překmitu.

- Omezení akčního zásahu To lze aplikovat jak na maximální a minimální hodnotu akční veličiny, tak i na rychlost její změny. Nebude tedy docházet k nerealistickým či nežádoucím požadavkům na akční zásah. Jelikož se jedná především o technologická, fyzikální či bezpečnostní omezení, jedná se zpravidla o tvrdá omezení.
- Omezení výstupní veličiny Zabraňuje dosažení nežádoucích či nebezpečných hodnot výstupní veličiny. Žádaná hodnota výstupní veličiny se v průmyslu pro dosažení nejvyšší efektivity často nachází v blízkosti hraničních oblastí. Při regulování vícero na sebe vázaných výstupních veličin by se optimální řešení také mohlo nacházet v nežádoucí oblasti některé z veličin. Příkladem je dosažení nežádoucích hodnot zrychlení při řízení rychlosti. Omezit lze také monotónnost průběhu výstupní veličiny.
- Omezení stavů Omezení jednotlivých stavů systému do určených rozsahů. Podobně jako u omezení výstupní veličiny zabraňuje dosažení nežádoucích operačních oblastí. Příkladem využití je koncentrace určitých látek u chemických procesů.
- Omezení překmitu Omezuje umožněnou velikost překmitu přes žádanou hodnotu. Překmit přes žádanou hodnotu obzvláště při velké regulační odchylce

(např po změně žádané hodnoty) může ve výsledku vést k rychlejšímu snížení regulační odchylky z důvodu agresivnější regulace. Nutnost tohoto omezení závisí na typu úlohy.

2.2 MPC pro nelineární systém

Řešení pomocí MPC pro lineární modely vede na optimalizaci řešící konvexní úlohu. Řešení konvexní úlohy je matematicky dobře popsáno analyticky i numericky. Z principu konvexní funkce lze garantovat dosažení blízkosti globálního minima. Numerické řešení za použití současné úrovně výpočetní techniky je dostatečně rychlé pro využití v průmyslu.[6]

U nelineárních modelů však řešení může vést na nekonvexní úlohu. Nalezení minima u nekonvexní úlohy je výpočetně mnohem náročnější a méně náročné metody nemusí garantovat, že nalezené minimum je skutečně minimem globálním. V praxi je tedy často výhodné vyhnout se použití nelineárního modelu pro výpočet optimálního akčního zásahu za pomoci různých metod linearizace.

2.2.1 Linearizace

Standardní linearizace je metoda často využívaná při řešení nelineárních úloh v MPC. Jejím principem je aproximace dynamiky systému za pomoci lineárního modelu v okolí operačního bodu. Linearizace zjednodušuje úlohu MPC, ale její použití je limitováno na okolí linearizovaného bodu. To vede k nepřestnostem v závislosti na nelinearitách a odchylce od operačního bodu.

2.2.2 Přepínání modelů

Jedním z možných přístupů k regulaci nelineárního systému je použití více-modelového prediktivního regulátoru. Pokud je možné aproximovat nelineární systém na sadu více lineárních modelů, je možné získat přesnější aproximaci, která neztrácí přesnost modelu při vzdalování se od základního linearizovaného bodu. Oproti základní linearizaci je tento přístup výhodný u systémů, jejichž operační rozsah

zasahuje do oblastí s rozdílnými vlastnostmi či dochází během pracovního cyklu ke známým změnám v dynamice systému.

Přechod mezi modely je možné provést jako tvrdý nebo měkký. U tvrdého přechodu dojde při dosažení daného bodu přechodu k prostému přepnutí na daný model. U měkkého přepínání se v blízkosti přechodu pracuje s váženým součtem generovaných akčních veličin. To je výhodné u úloh, kde jsou výrazně omezeny či penalizovány prudké změny akční veličiny.

2.2.3 Hybridní přístup

Další významnou možností je kombinování MPC přístupu s dalšími přístupy k nelineárnímu řízení. Obvyklé metody ke kombinaci s MPC jsou linearizace za pomoci zpětné vazby, adaptivní řízení a řízení pomocí klouzavého režimu.

2.3 Problematika robustnosti v MPC

Robustnost je schopnost zajištění zároveň stability a požadovaných cílů řízení kontrolního systému navzdory určité míře nejistot a nepřesností v matematickém modelu, změnám v parametrech systému a působení vlivů prostředí při praktickém použití. Vyšší míra robustnosti obvykle vede k zhoršení kvality regule systému za běžných pracovních podmínek. V robustním řízení se zpravidla počítá s nejhorším možným případem, který může v běžném provozu nastat. U vysoce rizikových systémů (např. letecký průmysl) je navrhován regulační systém tak, aby byla jeho robustnost zachována i v případě změn zasahujících mimo rámec běžného provozu (např. selhání či poškození částí systému).

2.3.1 Robustní MPC

Při návrhu prediktivního řízení je možné počítat s nejistotami modelu a zajistit jeho robustnost přímo vnitřně použitím různých metod robustního prediktivního řízení (RMPC). Nevýhodou mnoha metod je však vysoká výpočetní náročnost v kombinaci s výslednou konzervativní regulací vedoucí ke snížení nebo nedosažení požadované

performance. Zároveň je nutná znalost maximálního ohraničení nejistot. Běžné přístupy k řešení tohoto problému jsou dokumentovány v rámci Worst-case přístupu. Alternativou jsou stochastické přístupy, jako momentové nebo pravděpodobnostní metody.

2.3.2 Worst case přístup

Také známý jako min-max přístup, je deterministický přístup, který je možné popsat v několika bodech. Žádné uskutečnění ze sady nejistot nesmí destabilizovat systém v uzavřené smyčce nebo porušit omezení. Řízení zahrnuje do rozhodovacího procesu pouze danou sadu nepřesností, která může být úplná (tedy worst case) nebo pouze částečná (budgeted worst case). Akční zásah je vypočítáván s využitím aktuálních informací o nepřesnostech systému za předpokladu nejhoršího scénáře nebo s využitím konzervativnějšího odhadu nepřesností, ale s možností úpravy řídicí akce v souvislosti se získáním nových informací. Tato myšlenka vede k vysoké míře pesimismu v akčním zásahu. To je způsobeno zahrnutím i vysoce nepravděpodobných předpovídaných trajektorií.[6]

2.3.3 TubeBased approach

Princip tohoto přístupu spočívá v oddělení problému robustnosti a řízení MPC. Toho se dosáhne vytvořením nominálního stavu (značeno x_n). Tento stav není na rozdíl od reálného stavu ovlivňován nejistotami a MPC tedy vypočítává nominální trajektorii řízení, která zanedbává vliv nejistot. Vliv nejistot na skutečný stav je do regulace zahrnut v rámci pomocného řízení, které potlačuje odchylky mezi skutečným a nominálním stavem. Možné hodnoty rozdílu mezi skutečným stavem x a nominálním stavem x_n skrze horizont predikce poté vytváří obálku. Celá tato obálka poté musí splňovat omezení, čímž se zaručí, že i skutečný stav bude ležet uvnitř hranic omezení.[6, 7]

3 Návrh modelu pro balancujícího robota

Pro efektivní řízení robota je nejprve nutné vytvoření matematického modelu. Tím se odstraňuje nutnost tvorby celého řízení převážně za online připojení k robotu a zároveň je nezbytný pro použití metody MPC, které je na existenci modelu založena. Pro

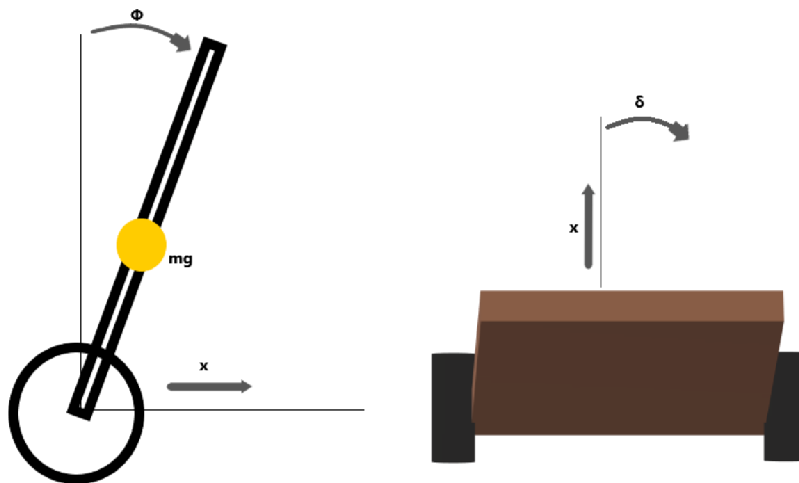
Tvorba simulačního modelu spočívá nejprve ve vytvoření teoretických pohybových rovnic dle tvaru robota. Na nelineární rovnice je poté provedena linearizace a z lineárních rovnic je vytvořen stavový popis. Pro dokončený popis je nutné provést identifikaci parametrů robota a validaci modelu. Model je poté vytvořen v prostředí Matlab simulink. Pro teoretický popis dynamiky robota je použita Lagrangeova rovnice druhého druhu viz 3.1.

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} = Q_i - \frac{\partial D}{\partial \dot{q}_i} \quad (3.1)$$

Vzhledem ke konstrukci balancujícího robota je možné a výhodné nejprve sestavovat rovnice kinetických energií pro několik samostatných částí. Celý robot je z hlediska výpočtu rozdělen na tři samostatná tuhá tělesa, dvě pro jednotlivá kola a další pro hlavní tělo robota.

3.1 Popis dynamiky robota

Pro znázornění zjednodušené varianty robota jako tuhého tělesa použitého pro tvorbu rovnic dynamiky a orientace jeho os byla vytvořena ukázka (Obrázek 3.1). Označení parametrů a veličin použitých v teoretickém výpočtu dynamického modelu je uvedeno níže (Tabulka 3.1).



Obrázek 3.1: Orientace balancujícího robota

Úhly	
Sklon	ϕ
Vybočení	δ
Vzdálenosti	
Osa rotace-těžiště	l
Kolo-kolo	D
Poloměr kola	r
Hmotnosti	
Kolo	m_k
Tělo	m_b
Momenty setrvačnosti	
Kolo	J_k
Tělo	J_b

Tabulka 3.1: Označení použitých veličin

3.1.1 Kola

Obě kola jsou geometricky považována za identická, jsou pro ně použity stejné parametry a rovnice pro pravé a levé kolo budou tedy obdobné. Kinetická energie kola

je uvedena v rovnici (3.2).[8]

$$Ek_1 = \frac{1}{2}m_k V_k^2 + \frac{1}{2}J_k \phi_k^2 \quad (3.2)$$

3.1.2 Tělo robota

Hlavní část robota je možné při tomto rozdělení pokládat za tuhé těleso tvaru kvádrů, což značně zjednodušuje tvorbu rovnic. Kinetická energie zahrnuje složku translace a dvě složky rotace. Rotace okolo osy x (změny nerovnosti povrchu) není uvažována, jelikož rotaci v tomto směru nelze řízením ovlivnit.

Rozbor rychlostí pro jednotlivé osy těla robota byl převzat z [9]. Celková rychlost byla pomocí Matlabu vypočtena dle vzorce (3.3). Výsledná rychlost je uvedena v rovnici (3.4). V rovnici výsledné rychlosti těla robota lze pozorovat člen závislý na druhých mocninách změny úhlu $\dot{\delta}$ a $\sin \phi$, který nemá význačný vliv na stabilizaci robota a jeho vliv je dále potlačen při linearizaci. Je tedy možné provést úpravu (3.5).

$$V_b = \sqrt{V_x^2 + V_y^2 + V_z^2} \quad (3.3)$$

$$V_b^2 = (\dot{x} + L\dot{\phi}\cos\phi)^2 + L^2(\dot{\phi}^2 + \dot{\delta}^2)\sin^2\phi \quad (3.4)$$

$$L^2\dot{\delta}^2\sin^2\phi = 0 \quad (3.5)$$

Translační složka kinetické energie těla robota je uvedena v (3.6), a rovnice pro jeho rotace jsou uvedeny v (3.7) pro náklon (rotace okolo osy z) a (3.8) pro zatáčení (rotace okolo osy x).

$$Ek_2 = \frac{1}{2}m_b \left((\phi_b l \cos \phi_b + \dot{x}_b)^2 + (\phi_b l \sin \phi)^2 \right) \quad (3.6)$$

$$Ek_3 = +\frac{1}{2}J_b \phi_b^2 \quad (3.7)$$

$$Ek_4 = +\frac{1}{2}J_b\delta^2 \quad (3.8)$$

Součtem těchto jednotlivých rovnic je získána celková kinetická energie robota. Potenciální energie robota považovaného za tuhé těleso lze získat vztahem 3.9. Při znalosti předchozích rovnic lze poté určit lagranžián a obecné koordináty.

$$Ep_b = m_g l \cos \phi \quad (3.9)$$

3.1.3 Vyjádření obecných souřadnic

Jako obecné souřadnice se nabízí použít polohu a sklon. Další možností je použít natočení kola a sklon. Výhodou použití natočení kol je separování akčního zásahu na jednotlivá kola už v tomto kroku tvorby modelu. Nevýhodou je poté nutnost výpočtu polohy z natočení. V [8] je jako obecná souřadnice určen úhel natočení kola, pro potřeby práce s modelem se však jako výhodnější jeví pracovat s polohou. Jako obecné souřadnice byly tedy zvoleny posun x a sklon ϕ . Výpočet polohy x je pak proveden separátně. Správnost výpočtu je pak ověřena z původních rovnic pro moment kol (převzatých z [8]) jejich spojením a úpravou dle (3.10) a využitím vztahu (3.11).

$$Q_x = \frac{(\tau_{k1} + \tau_{k2})}{r} \quad (3.10)$$

$$x = r\phi \quad (3.11)$$

Výsledné rovnice s obecnými souřadnicemi pro polohu a sklon popisující dynamické chování robota jsou uvedeny v (3.14) a (3.15). Jelikož jsou části rovnic vycházející z dynamiky kol obdobné, jsou pro přehlednost označeny indexovaným symbolem K , který je definován v (3.12) a (3.13).

$$K_{xi} = (m_{ki} + \frac{J_{ki}}{r^2} + \frac{1}{2}m_b)\ddot{x}_{ki} \quad (3.12)$$

$$K_{\phi_i} = \frac{1}{2}m_b r l \cos\phi_{ki} \ddot{\phi}_{ki} \quad (3.13)$$

$$x : K_{x1} + K_{x2} + m_b l \cos\phi \ddot{\phi} - m_b l \sin\phi \dot{\phi}^2 = \frac{(\tau_{k1} + \tau_{k2})}{r} - \frac{1}{2r^2} b_k \dot{x}^2 \quad (3.14)$$

$$\phi : K_{\phi1} + K_{\phi2} + (J_b + m_b l^2) \ddot{\phi} = m_b g l \sin\phi - (\tau_R + \tau_L) \quad (3.15)$$

3.2 Linearizace a tvorba stavového popisu

Výsledný popis obsahuje nelineární prvky. Nelinearizovaný popis je použit k tvorbě modelu, použitého v simulaci skutečného zařízení. Pro vnitřní model MPC, který slouží k výpočtu akčního zásahu, je v tomto případě pro zjednodušení komplexnosti a náročnosti výpočtu řízení výhodné provést linearizaci nelineárních prvků. Jak lze vidět v [8] a [1], je možné provést linearizaci okolo pracovní polohy, kdy je sklon relativně blízký nule. Použitá linearizace níže v (3.16) je provedena dle sady rovnic, která je vhodná pro tento typ úlohy.

$$\sin\phi \approx \phi, \cos\phi \approx 1, \dot{\phi}^2 \approx 0 \quad (3.16)$$

3.2.1 Decoupling

Decoupling je anglický výraz ozačující rozpojení dvou (či více) komponent v systému, což umožňuje jejich vzájemně nezávislou funkčnost. V tomto případě se bude jednat o rozpojení systému zatačení a systému ovlivňujícím náklon a rychlost. Tato možnost je vidět v [8] a [9]. Rozpojovací matice je uvedena níže viz (3.17).

$$\begin{pmatrix} \tau_L \\ \tau_R \end{pmatrix} = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{pmatrix} \begin{pmatrix} \tau_\phi \\ \tau_\delta \end{pmatrix} \quad (3.17)$$

3.2.2 Stavový popis

Z linearizovaného modelu je poté vytvořen stavový popis. To je provedeno dle základní rovnice (3.18), kde je tvar vektoru \mathbf{x} uveden v (3.19). Na matice byla použita Výsledná stavová matice s jejími členy je uvedena v rovnicích (3.20) až (3.28).

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \quad (3.18)$$

$$\mathbf{x} = \left(x \quad V \quad \phi \quad \omega \quad \delta \quad \dot{\delta} \right)^T \quad (3.19)$$

$$\dot{\mathbf{x}} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & 0 & 1 \\ 0 & a_{42} & a_{43} & a_{44} \end{pmatrix} \mathbf{x} + \begin{pmatrix} 0 \\ b_2 \\ 0 \\ b_4 \end{pmatrix} \tau_\phi \quad (3.20)$$

$$a_{22} = \frac{-2K_m K_e b_2}{Rd} \quad (3.21)$$

$$a_{23} = \frac{-m_b^2 L^2 g}{m_B J_d + K_1}; \quad (3.22)$$

$$a_{24} = 2K_m K_e b_2; \quad (3.23)$$

$$a_{42} = \frac{-2K_m K_e b_4}{Rd} \quad (3.24)$$

$$a_{43} = \frac{m_b^2 g L + 2m_b g L \frac{m_w + J_w}{R^2}}{m_b J_b + K_1} \quad (3.25)$$

$$a_{44} = 2K_m K_e b_4 \quad (3.26)$$

$$b_2 = K_m \frac{\frac{J_b + m_b L^2}{R + m_b L}}{J_b m_b + K_1} \quad (3.27)$$

```

mB = 1.098 %hmotnost robota
mw = 0.050; % hmotnost kolecek
R = 0.034; % polomer kolecka
a = 0.15; b = 0.21; c = 0.06; % rozmery robota,
L = 0.051; % vzdalenost rotacni osa-teziste
g = 9.81;
Ke = 0.0175;
Rd = 2.3;
Km = 0.127;
%konstanty modelu
Jw = 0.5*mw*R*R; %2.89*e^-5
D = 0.25; %rozpětí kol
Ckx1 = mw+0.5*mB+(Jw/(R*R)); %0.6240
Ckx2 = mw+0.5*mB+(Jw/(R*R));
Ckf1 = 0.5*mB*R*L;
Ck = 3.51
Jb = (1/12)*mB*(a^2+b^2)+mB*0.25*a^2; %0.0123
omega = (2*(Jb+mB*L^2)*(mw+Jw/R^2)); %0.023
Jw = 0.5*mw*R^2;
Jd = (1/12)*mB*(a^2+c^2);

```

Obrázek 3.2: Naměřené a použité parametry v Matlabu

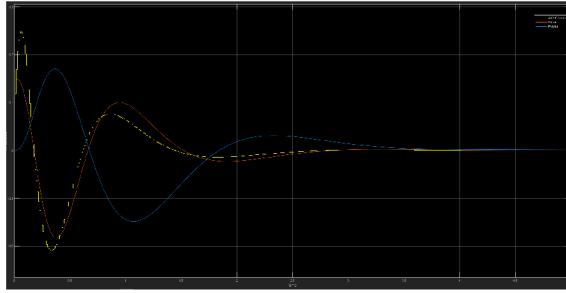
$$b_4 = \frac{K_m \frac{-m_b(R+L)}{R} - 2K_m \left(m_w + \frac{J_w}{R^2}\right)}{K_1} \quad (3.28)$$

Do výpočtu stavového popisu byl zahrnut i vliv motoru JGA25-370 s vestavěnou převodovou skříní. Vliv převodové skříně je uvažován jako konstanta uvedená v 3.29. Hodnoty pro motor jsou uvedeny v tabulce 1.1

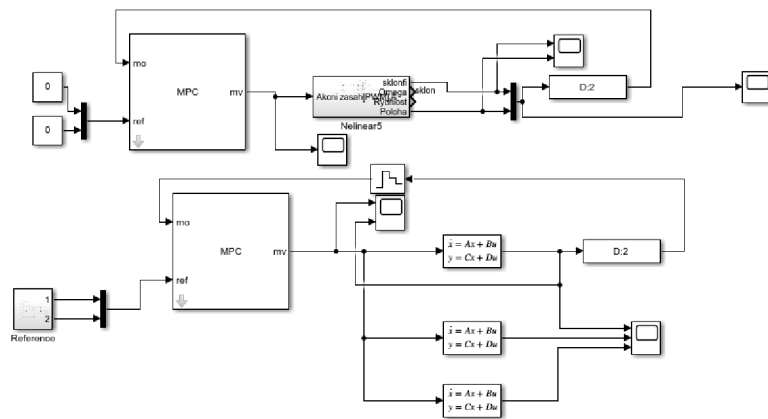
$$K_2 = \frac{\eta}{G} = 3,51 \quad (3.29)$$

3.3 Identifikace parametrů robota

Řadu parametrů jako jsou jednotlivé rozměry a hmotnost bylo od robota možné měřit přímo. Některé, jako jsou například momenty setrvačnosti, však bylo nutné zjistit výpočtem. Vzorce pro tyto výpočty jsou vzhledem k složitému tvaru a nerovnoměrně rozložené hmotnosti pouze přibližné a vnášejí do modelu další nejistoty. U některých obtížně zjistitelných hodnot bylo nutné se spolehnout pouze na tabulkové hodnoty.



Obrázek 3.3: Simulovaný průběh regulačního obvodu



Obrázek 3.4: Schéma pro testování nastavení MPC

3.4 Tvorba modelu v Matlabu

Model robota byl tvořen v několika krocích. Nejprve bylo vytvořeno simulační schéma pomocí linearizovaného stavového popisu viz (3.4). To sloužilo k prvotnímu návrhu parametrů MPC. MPC bylo také testováno na nelineárním modelu vytvořeném z původních rovnic před jejich linearizací. V pozdější fázi na něm byl testován i model vytvořený skrze inputa/output data.

3.5 Návrh prediktivního regulátoru

MPC bylo navrhováno pro kontrolu náklonu a polohy robota. Regulátor má jednu vstupní veličinu, kterou je napětí motoru. Referencí je požadavek na nulovou

odchylku sklonu a nulovou či měnící se polohu.

Omezení Manipulované veličiny jsou koncipována jako tvrdá a akční zásah je škálován pro potřeby PWM. Výstupní veličiny mají měkká omezení, jelikož porušení omezení polohy není kritické. U natočení úhlu byla pro měkká omezení (s tvrdým postihem) zjištěna lepší odezva regulátoru při výskytu chyb a špiček, které se na měření při vyšší úhlové rychlosti vyskytovaly. U polohy bylo nastaveno omezení $0.3m$ a pro úhel se jednalo o $\frac{\pi}{10}$. V nastavení vah bylo výzvou odbouření předčasného snížení akčního zásahu, které mělo za následek setrvání v nakloněné pozici, které často vedlo k destabilizaci robota. Důvodem rozdílu mezi očekáváním regulátoru a reálným stavem může být u vyšších hodnot náklonu převážení vlivu nelineárních členů a nepravidelnosti v rozložení hmotnosti a tvaru robota, způsobující nepřesnost v momentu setrvačnosti a změny v pozici těžiště.

4 Řízení

Pro propojení robota s počítačem bylo využito HDMI propojení a po nastavení Raspberry Pi přístup pomocí SSH.

Základní komunikační rozhraní mezi senzory a RaspberryPi bylo postaveno v [1]. K novému přiřazení pinů nebyl žádný důvod, přiřazení pinů tedy bylo u robota ponecháno. Pro čtení a odesílání dat do RaspberryPi byly použity bloky z knihovny Matlabu zmíněné v části 1.2.1.

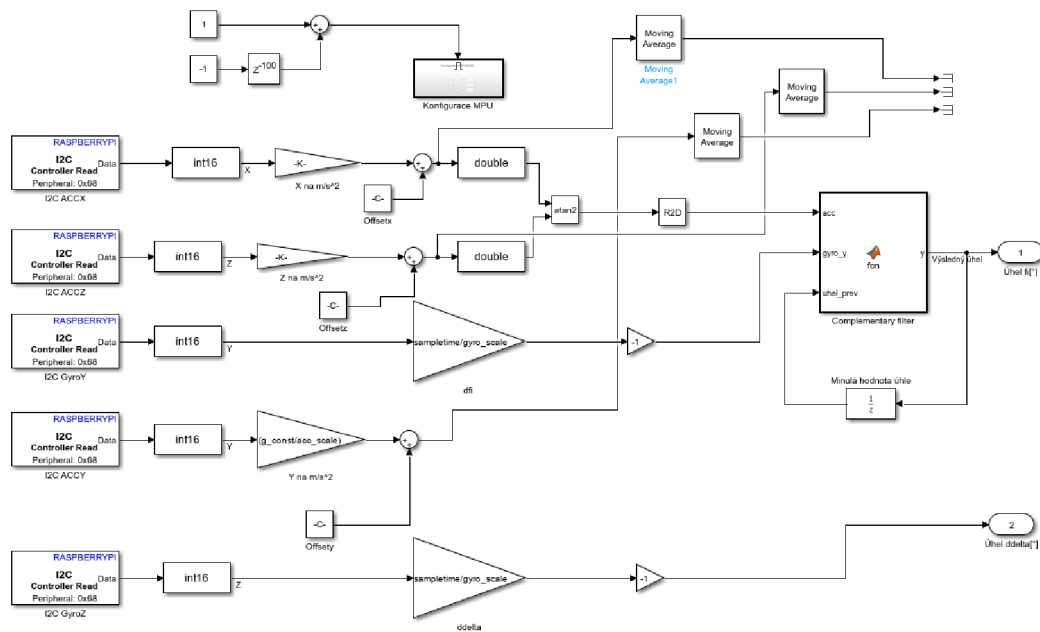
4.1 Kalibrace a měření MPU-6050

Po připojení k robotu a provedení prvotního spuštění bylo nutné provést kalibraci akcelerometrů v klidové stavu. Kalibrace byla provedena manuálně pro klidový stav za působení pouze gravitačního zrychlení. Pomocí měřicího řetězce zobrazeného viz obr 4.1 byl zjištěn offset pro jednotlivé osy. Výchylka jednotlivých os z důvodu nepřesnosti v orientaci čipu při instalaci byla u osy Y naměřena zanedbatelná (okolo 0.5%) a pro nepřesnost zbývajících os je model doplněn o kompenzační konstantu. Ta je nastavena tak, aby co nejvíce eliminovala nepřesnost v měření úhlu ϕ (sklon), který je klíčový pro funkci regulátoru a dosažení vzpřímené polohy robota.

Měření změny sklonu vychází z dat gyroskopu, je tedy nutné, jak bylo již však zmíněno v sekci 1.1.3, potlačit vliv šumu, aby nedocházelo k postupnému růstu odchylky od skutečné hodnoty. Potlačení je obvykle provedeno komplementárním či Kálmánovým filtrem. Komplementární filtr kompenzuje vznikající nepřesnost za pomocí dat z akcelerátorů, která jsou stabilní pro stacionární a nízkofrekvenční měření. Největší výhodou komplementárního filtru je jednoduchost jeho implementace, nízké zpoždění a nižší náročnost na výpočetní výkon.

Alternativou k komplementárnímu filtru by mohl být Kálmánův filtr. Kálmánův filtr je náročnější na výpočet i implementaci (nutnost dostatečně rychlé periody

a přesnosti nastavení), avšak by měl být teoreticky účinnější. V praxi však často nepřesnosti v jeho nastavení zhoršují jeho účinnost a rozdíl mezi nepřesnostmi obou filtrů jsou v nižších rychlostech nevýznamné viz [Filter], a teoretické výhody Khalmánova filtru pro měření pomocí MPU-6050 nepřeváží zvýšenou náročnost implementace.



Obrázek 4.1: Měření sklonu pomocí MPU-6050

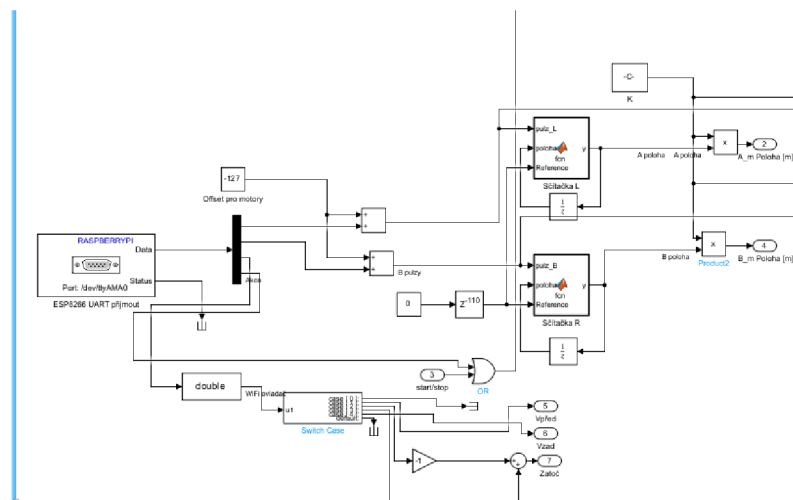
V robotu je tedy implementován měřící řetězec s komplementárním filtrem. V měřícím řetězci byly upraveny převody za pomoci bloků pro drobné snížení výpočetní chyby. Dále byly upraveny váhy pro zvýšení podílu akcelerometru v komplementárním filtru pro zlepšení dlouhodobé chyby z důvodu předpokladu nižších rychlostí změny úhlu. Řetězec je vidět na obr 4.1. Úhel δ je možné pomocí MPU-6050 měřit pouze pomocí dat z gyroskopu. Důvodem toho je nemožnost použití akcelerometru pro osu kolmou ke gravitačnímu poli. Přesné měření však není vyžadováno pro systém ovládání, v kterém je zatáčení voleno uživatelem manuálně.

4.2 Řízení balančního robota

V úloze bylo voleno zatačení pomocí manuálního inputu uživatele. Důvodem je problematické získávání hodnoty úhlu zatočení δ . Pro získávání přesné hodnoty zatočení by bylo možné robota doplnit o magnetometr, což by ale vyžadovalo změnu konstrukce. Řídicí schéma je rozděleno do několika subsystémů, které budou uvedeny v následujících částech. Konkrétně se jedná o subsystémy Modulu MPU-6050, Esp8266, Decoupling matici a její inverzi, spouštění motorů a ovladač robota.

4.2.1 Subsystém Esp8266

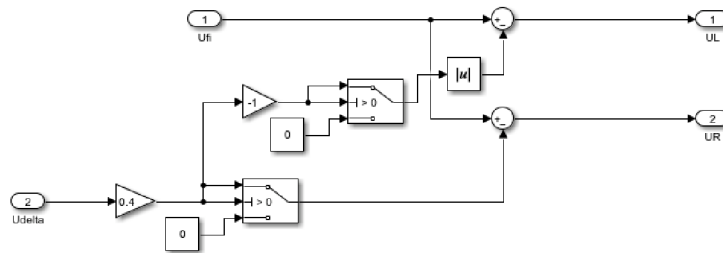
Tento subsystém zahrnuje komunikaci přes sběrnici UART skrze sériový port. Z tohoto portu jsou získávána data o rychlosti rotace kol z enkodérů ve formátu int8. Pro získání polohy ze složky rychlosti není vhodné použít sumu. Ta by způsobila obdobný problém jako byl popsán u měření veličiny δ gyroskopem viz 4.1. O něco lepší se jeví metoda odměření uražené vzdálenosti pro otáčku kola a přenásobení hodnoty rychlosti tímto poměrem. Zbylé porty nejsou pro aplikaci řízení potřebné, avšak lze je využít k zpětné kompatibilitě s původní aplikací uvedenou v [1] pro starší verze Android. Této kompatibilitě je dosaženo pomocí logických bloků OR a Switch Case viz.4.2



Obrázek 4.2: Výstup ze sběrnice UART

4.2.2 Inverzní Decoupling matice

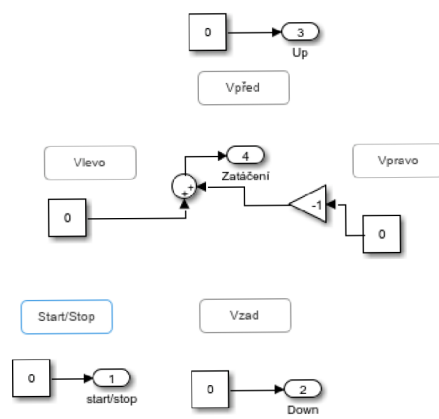
Pro potřeby regulátoru byla v kapitole 3.2.1 použita matice transformující momenty $\tau_L \tau_R$ na $\tau_\phi \tau_\delta$. Pro potřeby řízení motorů je však třeba opět získat akční zásahy pro levý a pravý motor. K tomuto účelu slouží subsystém 4.3.



Obrázek 4.3: Systém sloužící k logice zatáčení

4.2.3 Ovladač Robota

V subsystému zobrazeném na obr 4.4 lze vidět ovladač, který slouží k testování funkcí řízení. Ovladač je vybaven příkazy jízdy dopředu, dozadu, vlevo a vpravo. Dále je zde umístěna funkce zapínání a vypínání motorů.



Obrázek 4.4: Ovládání pojezdu a funkcí robota

5 Závěr

Diplomová práce se nejprve zabývá přehledem aktuální konstrukce, popisem významných částí hardwaru balancujícího robota, použitého softwaru a principem prediktivního řízení. Dále obsahuje postup navrhování modelu balancujícího robota jako inverzního kyvadla.

V teoretické části je nejprve přiblížena aktuální architektura hardwaru a softwaru balancujícího robota a prostředí Matlab, ve kterém je vytvořen model a řízení robota. Další část práce se zabývá shnutím parametrů prediktivního regulátoru a jeho možných přístupů při řízení nelineárního systému.

V praktické části byl nejprve sestaven simulační model balancujícího robota. Simulační model byl vytvořen pomocí matematického modelu, který byl linearizován okolo pracovní polohy. Tento model byl parametrizován podle reálného zařízení. Poté byl vytvořen řídicí algoritmus pro stabilizaci a jízdu robota včetně zatočení. Algoritmus byl založen na principu MPC regulátoru. Prvotní nastavení regulátoru proběhlo v programu Matlab Simulink kde bylo jeho chování testováno simulací. Skrze nadstavbu Matlabu bylo řízení implementováno na reálné zařízení řízeném RaspberryPi 3B+.

Narozdíl od simulace se implementace na reálném zařízení potýkala s problémy v podobě nepřesného umístění snímače náklonu robota a jeho častých výpadků. Pro přesnější identifikaci modelu, která by mohla kompenzovat nepřesnosti snímače, by bylo vhodné použití jiného typu řízení, který by identifikoval model v uzavřené smyčce.

Úloha by mohla být zlepšena nahrazením snímače náklonu robota za přesnější typ a doplněním o magnetometr. Implementaci by usnadnilo doplnění o opěrná kola určená pro ochranu robota během testování.

Použitá literatura

- [1] PETR, Mamula. *Stabilizace a navádění balancujícího robota*. 2021.
- [2] FAHMIZAL, MUHAMMAD ARROFIQ. Robot Inverted Pendulum Beroda Dua (IPBD) dengan Kendali Linear Quadratic Regulator (LQR). *ELKOMIKA*. 2019.
- [3] *Raspberry Pi 3 Model B+* [online]. 2023. [cit. 2022-11-05]. Dostupné z: https://www.tinyosshop.com/index.php?route=product/product&product_id=908.
- [4] [B.r.]. Dostupné také z: <https://www.laskakit.cz/wemos-d1-mini-esp8266-wifi-modul/>.
- [5] MARTIN, Moštěk. *Prediktivní řízení s filtrací akčních a řízených veličin*. 2023.
- [6] SALTİK, M. Bahadır et al. An outlook on robust model predictive control algorithms: Reflections on performance and computational aspects. *Journal of Process Control*. 2018.
- [7] SINGH, Sumeet, Marco PAVONE a Jean-Jacques E. SLOTTINE. Tube-Based MPC : a Contraction Theory Approach. In: 2016. Dostupné také z: <https://api.semanticscholar.org/CorpusID:30825043>.
- [8] XU, Changkai, Ming LI a Fangyu PAN. The System Design and LQR control of a Two-wheels Self-balancing Mobile Robot. *Shanghai University*. 2024.
- [9] BIN, He, Liu Wen ZHEN a Lv Hai FENG. The Kinematics Model of a Two-wheeled Self-balancing Autonomous Mobile Robot and Its Simulation. *Second International Conference on Computer Engineering and Applications*. 2010.
- [10] *ESP8266MOD WiFi Module* [online]. 2024. [cit. 2022-11-05]. Dostupné z: https://www.tinyosshop.com/index.php?route=product/product&product_id=908.