



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

ŘÍZENÍ MODELU V REÁLNÉM ČASE

REAL-TIME CONTROL OF BALL ON WHEEL MODEL

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

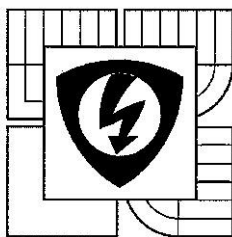
AUTOR PRÁCE
AUTHOR

Bc. ROMAN KREJČÍ

VEDOUcí PRÁCE
SUPERVISOR

Ing. ONDŘEJ HYNČICA

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Roman Krejčí
Ročník: 2

ID: 136551
Akademický rok: 2014/15

NÁZEV TÉMATU:

Řízení modelu v reálném čase

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je implementace řídicího algoritmu do vestavěného mikrokontroléru s operačním systémem reálného času. Řízeným systémem je laboratorní model balancování míčku na kole.

- 1) Sestavte laboratorní model systému, vytvořte matematický model a navrhnete způsob řízení.
- 2) Navrhnete regulátor a ověřte jeho funkčnost na modelu.
- 3) Implementujte programové vybavení pro ovládání servopohonu, snímač vzdálenosti a algoritmus regulátoru jako úlohy v operačním systému reálného času do řídicího mikrokontroléru.
- 4) Vytvořené řešení otestujte a navrhnete úpravy laboratorního modelu.

DOPORUČENÁ LITERATURA:

SLOSS, Andrew N, Dominic SYMES a Chris WRIGHT. ARM system developer's guide: designing and optimizing system software. Amsterdam: Elsevier, 2004, 689 s. ISBN 15-586-0874-5.

Termín zadání: 9. 2. 2015

Termín odevzdání: 18.5.2015

Vedoucí práce: Ing. Ondřej Hynčica

Konzultanti diplomové práce:

doc. Ing. Václav Jirsík, CSc.

předseda oborové rady



UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

ABSTRAKT

Tato diplomová práce se zabývá řízením systému Ball on the wheel v reálném čase. V první části práce je vytvořen matematický model systému, který je popsán stavovými rovnicemi. Dále je popsán vytvořený laboratorní model systému Ball on the wheel a jeho součástí. Laboratorní model se skládá ze servomotoru, servozesilovače, mikrokontroléru s operačním systémem reálného času a snímačů vzdálenosti. V praktické části práce je navrženo řízení systému pomocí lineárně kvadratického regulátoru a jeho implementace spolu s ovládáním periférií do firmwaru mikrokontroléru s operačním systémem reálného času. Systém Ball on the wheel je řízen mikrokontrolérem, který posílá příkazy servozesilovači po sběrnici CAN.

KLÍČOVÁ SLOVA

systém Ball on the wheel, servomotor, mbed, operační systém reálného času, mbed-rtos, sběrnice CAN, lineárně kvadratický regulátor

ABSTRACT

This diploma thesis deals with control the Ball on the wheel system in real time. In the first part there is made mathematical model of system, which is described by state equations. Then the laboratory model of Ball on the wheel system and its parts are described. The laboratory model consists servo drive, servo amplifier, microcontroller with real-time operating system and distance sensors. In the practical part of the thesis there is designed control of system by linear quadratic regulator and its implementation with control peripherals into microcontroller with real-time operating system. Ball on the wheel system is controlled by microcontroller, which sends a commands to servo amplifier by CAN bus.

KEYWORDS

Ball on the wheel system, servomotor, mbed, real-time operating system, mbed-rtos, CAN bus, linear quadratic regulator

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Řízení modelu v reálném čase“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Ondřeji Hynčicovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Také bych rád poděkoval mé rodině za trpělivost a podporu při celém studiu i psaní této diplomové práce.

Brno

.....

podpis autora(-ky)

OBSAH

Úvod	11
1 Systém Ball on the wheel	12
1.1 Modelování systému	13
1.2 Stavový popis systému	16
1.3 Linearizace systému	17
1.4 Modelovací schéma systému	18
1.5 Výpočet momentu setrvačnosti kola	19
1.6 Výsledné parametry modelu	20
2 Laboratorní model systému	21
2.1 Celkové blokové schéma systému	21
2.2 Vývojová deska mbed LPC1768	22
2.2.1 Vlastnosti vývojové desky	22
2.3 Digitální servozesilovač TGA-24-9/20	23
2.3.1 Operační režimy servozesilovače	23
2.3.2 Komunikace se servozesilovačem	24
2.4 Střídavý synchronní servomotor TGN2-0095	25
2.4.1 Parametry servomotoru	26
2.5 Použité senzory pro měření pozice míčku	26
2.5.1 Ultrazvukový snímač vzdálenosti US-100	26
2.5.2 Fotoelektrický snímač vzdálenosti FADK 14	28
3 Servomotor a servozesilovač	31
3.1 Regulace momentu v d-q souřadnicích	31
3.2 Nastavení proudových regulátorů	31
3.3 Komunikace na sběrnici CAN	33
3.3.1 Parametrizační datový objekt - SDO	34
3.3.2 Procesní datový objekt - PDO	35
3.3.3 Objekt SYNC	36
3.3.4 Příklad komunikace	36
4 Řízení systému Ball on the wheel	38
4.1 Stavová zpětná vazba	38
4.2 Měření stavů systému Ball on the wheel	39
4.2.1 Výpočet úhlu pozice míčku	39
4.2.2 Výpočet úhlové rychlosti míčku	39
4.2.3 Měření úhlového natočení kola	40

4.2.4	Zjištění úhlové rychlosti kola	40
4.3	Navržený model systému	40
4.4	Návrh řízení systému pomocí stavové zpětné vazby	42
4.5	Návrh linérně kvadratického řízení	43
5	Programové vybavení	47
5.1	Operační systém mded-rtos	47
5.1.1	Třída Thread	47
5.1.2	Třída Mutex	49
5.1.3	Signal Management	49
5.2	mbed SDK	49
5.3	Implementace programového vybavení	50
5.3.1	Vytvořené třídy	50
5.3.2	Vytvořená vlákna	52
5.3.3	Kritická oblast	58
5.3.4	Stavy řídicího programu	58
6	Výsledky a posouzení vytvořeného řešení	60
6.1	Naměřené průběhy stavů systému při řízení	60
6.2	Posouzení řešení laboratorního modelu	62
6.2.1	Servomotor	63
6.2.2	Výběr snímačů vzdálenosti	63
6.2.3	Výběr kola	64
6.2.4	Výběr míčku	64
6.3	Návržené úpravy laboratorního modelu	64
7	Závěr	66
	Literatura	67
	Seznam symbolů, veličin a zkratk	69
	Seznam příloh	71
A	Obsah přiloženého CD	72
B	Deska plošných spojů	73
B.1	Schéma zapojení DPS	73
B.2	Návrh DPS	74
C	Class diagramy vytvořených tříd	75

SEZNAM OBRÁZKŮ

1.1	Schéma systému Ball on the wheel	12
1.2	Systém Ball on the wheel	13
1.3	Schématické zobrazení systému Ball on the Wheel	14
1.4	Blokové modelovací schéma systému Ball on the wheel	18
1.5	Průběh rychlosti kola při rozběhu motoru	19
2.1	Laboratorní model systému Ball on wheel	21
2.2	Celkové blokové schéma systému	22
2.3	Používané rozhraní	23
2.4	Servozesilovač TGA-24-9/24	24
2.5	Zapojení servozesilovače TGA-24-9/24	24
2.6	Ukotvení servomotoru TGN2-0095	25
2.7	Ultrazvukový snímač vzdálenosti US-100	27
2.8	Závislost úhlové pozice míčku na šířce přijatého pulzu	27
2.9	Fotoelektrický snímač vzdálenosti FADK 14	28
2.10	Napěťový dělič s operačním zesilovačem	29
2.11	Závislost úhlové pozice míčku na normalizovaných hodnotách napětí	29
2.12	Uložení senzorů v nerezovém pouzdře	30
3.1	Regulace momentu v d-q souřadnicích	31
3.2	Ekvivalentní fázový model servomotoru v d-q souřadnicích	32
3.3	Odezva q složky proudu na skokovou změnu žádané hodnoty proudu	33
3.4	Schéma zapojení CAN budiče	33
3.5	Příklad komunikace po sběrnici CAN	37
4.1	Blokový diagram stavové zpětné vazby	39
4.2	Model systému Ball on the wheel	41
4.3	Schéma řízení systému pomocí stavové zpětné vazby	42
4.4	Schéma řízení systému Ball on the wheel	45
4.5	Průběh úhlové pozice a úhlové rychlosti míčku	45
4.6	Průběh úhlového natočení a úhlové rychlosti kola	46
4.7	Průběh momentotvorné složky proudu do motoru	46
5.1	Struktura programu s CMSIS-RTOS API [17]	47
5.2	Schématické zobrazení stavů vláken [15]	48
5.3	Zobrazení funkce mutexu [15]	49
5.4	Signal Management	49
5.5	Struktura zdrojového kódu	50
5.6	Jednotlivá vlákna programu	52
5.7	Vývojový diagram pro inicializační vlákno	53
5.8	Vývojový diagram vlákna pro snímače vzdálenosti	54

5.9	Vývojový diagram vlákna pro vysílání synchronizace	54
5.10	Vývojový diagram vlákna pro zobrazení stavu programu	55
5.11	Vývojový diagram vlákna pro řízení	56
5.12	Vývojový diagram vlákna pro sběr dat	57
5.13	Vývojový diagram pro výpis dat	58
5.14	Stavový diagram řídicího programu	59
6.1	Průběhy úhlové pozice a úhlové rychlosti míčku	60
6.2	Průběhy úhlového natočení a úhlové rychlosti kola	61
6.3	Průběh požadované momentotvorné složky proudu	61
6.4	Průběh úhlové pozice míčku měřené ultrazvukovým snímačem	62
6.5	Nepájivé kontaktní pole	63
6.6	Konstrukce laboratorního modelu	65
B.1	Schéma zapojení DPS	73
B.2	Návrh desky plošných spojů	74
C.1	Class diagram třídy Can	75
C.2	Class diagram třídy cPrint	76
C.3	Class diagram třídy cUltrasonic	76
C.4	Class diagram třídy cLedSensor	76
C.5	Class diagram třídy cControl	76

SEZNAM TABULEK

1.1	Parametry laboratorního modelu	20
1.2	Výsledné koeficienty laboratorního modelu	20
2.1	Parametry synchronního servomotoru TGN2-0095 [7]	26
3.1	Struktura zprávy SDO	34
3.2	Struktura zprávy požadavku na vyčtení	34
3.3	Struktura zprávy odpovědi na požadavek na vyčtení	34
3.4	Struktura zprávy požadavku na zápis	35
3.5	Struktura zprávy odpovědi na požadavek na zápis	35
3.6	Struktura RPDO v režimu souvislého momentového režimu	35
3.7	Příklad zprávy RPDO pro nastavení požadovaného proudu 3 A	35
3.8	Struktura TPDO v režimu souvislého momentového režimu	36
3.9	Struktura SYNC	36
4.1	Vlastní čísla matice A	41
4.2	Vlastní čísla nové matice A'	44

ÚVOD

Tato diplomová práce se bude zabývat problematikou řízení systému Ball on the wheel, u kterého je cílem dosáhnout stabilizace polohy míčku v horní pozici kola.

Nejdříve bude nutné vytvořit matematický model systému Ball on the wheel a také sestavit laboratorní model systému. To bude zahrnovat výběr řídicího mikrokontroléru a servozesilovače, který bude s tímto mikrokontrolérem komunikovat a řídit vybraný servomotor pohánějící kolo. Dalším prvkem laboratorního modelu bude výběr snímače vzdálenosti, který bude vyhodnocovat pozici míčku na kole. To celé zahrnuje splnění prvního bodu zadání a je popsáno v kapitolách 1 a 2 této práce.

Kapitola 3 se bude zabývat vybraným servomotorem a jeho nastavení. Důležitý bude také způsob komunikace mezi řídicím mikrokontrolérem a servozesilovačem, který bude také popsán v této kapitole.

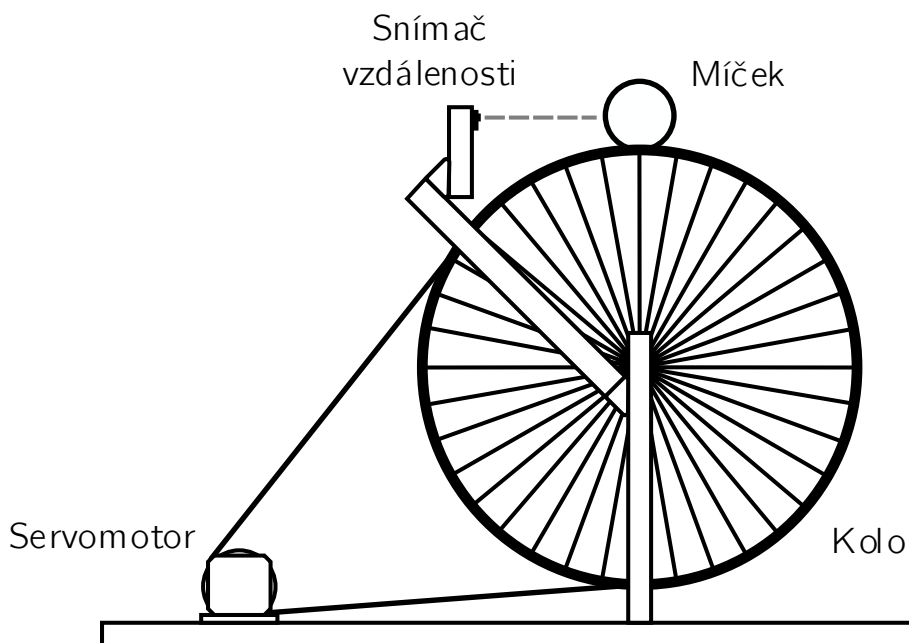
Dále bude nutné navrhnout řízení systému Ball on the wheel. To bude popsáno v kapitole 4 a jedná se o druhý bod zadání této práce. Tato kapitola bude zahrnovat teoretický rozbor způsobu řízení systému, poté návrh řízení systému Ball on the wheel a následně ověření funkčnosti řízení na modelu Ball on the wheel.

Poté bude do řídicího mikrokontroléru implementováno programové vybavení pro ovládání servomotoru, vybraných snímačů vzdálenosti, komunikace se servozesilovačem a řízení systému Ball on the wheel. Programové vybavení mikrokontroléru bude založeno na operačním systému reálného času. Tomuto bude věnována kapitola 5 a bude zahrnovat třetí bod zadání diplomové práce.

Celé vytvořené řešení bude následně otestováno a ověřena jeho funkčnost. To bude uvedeno v poslední kapitole této práce a jedná se tak o poslední bod zadání. Tato kapitola bude také obsahovat navržené úpravy laboratorního modelu, které by měli do budoucna přinést usnadnění práce s modelem či lepší výsledky při řízení.

1 SYSTÉM BALL ON THE WHEEL

V této kapitole je popsán systém Ball on the wheel, kde je cílem udržet míček v horní pozici pomocí natáčení kola. Na obrázku 1.1 se nachází schéma systému Ball on the wheel.



Obr. 1.1: Schéma systému Ball on the wheel

Kolo je poháněno střídavým synchronním servomotorem přes ozubenou řemenici. Servomotor je řízen servozesilovačem, který komunikuje s mikrokontrolérem mbed přes sběrnici CAN. Pozice míčku je snímána senzorem vzdálenosti. Pro snímání pozice míčku jsou použity dva druhy snímačů, které budou popsány v kapitole 2. Tyto snímače nepracují zároveň a při uvádění do provozu je nutné, aby uživatel zvolil, který snímač bude použit. Úhlové natočení kola a úhlová rychlost kola je měřena pomocí resolveru zabudovaného v servomotoru. Na obrázku 1.2 se nachází systém Ball on the wheel.

Systém Ball on the wheel je zejména využíván k výuce řízení nelineárních systémů. Jedná se o systém jehož hlavními vlastnostmi jsou nelinearita a nestabilita. Dalším příkladem může být inverzní kyvadlo, kulička na tyči či kulička na desce. [2]



Obr. 1.2: Systém Ball on the wheel

1.1 Modelování systému

K získání modelu systému se vycházelo z prací popsaných v [1], [2]. Pro vytvoření pohybových rovnic modelu byly použity Lagrangeovy rovnice druhého druhu, které zjednodušují práci při sestavování pohybových rovnic systémů [3]. Na obrázku 1.3 se nachází základní schéma systému. Podoba Euler-Lagrangeovy rovnice je

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}} \right] - \frac{\partial L}{\partial q} = Q, \quad (1.1)$$

kde

L je Lagrangeova funkce,

Q jsou zobecněné síly,

q jsou zobecněné souřadnice.

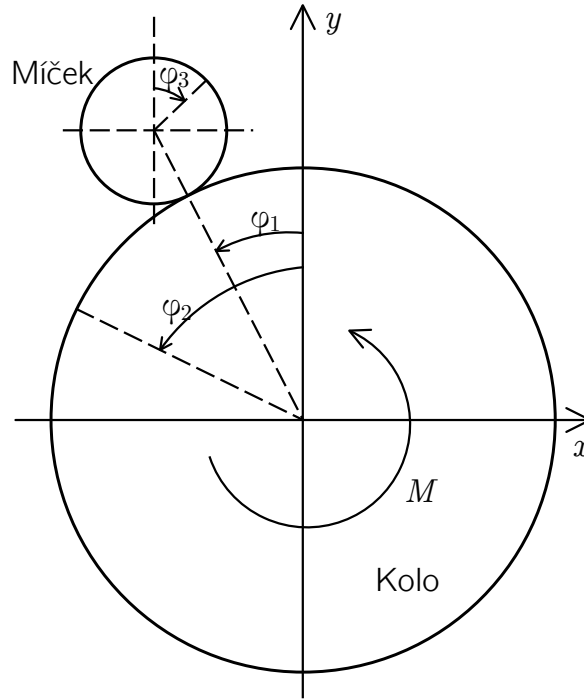
Lagrangeova funkce je definována podle rovnice 1.2

$$L = T - V, \quad (1.2)$$

kde

T je kinetická energie a

V je potenciální energie systému.



Obr. 1.3: Schématické zobrazení systému Ball on the Wheel

Pro zobecněné souřadnice platí

$$q = \begin{bmatrix} \varphi_1 \\ \varphi_2 \end{bmatrix}, \quad (1.3)$$

kde

φ_1 [rad] je úhel mezi osou y a spojnici středů kola a míčku,

φ_2 [rad] je úhlové natočení kola.

Pro zobecněné síly platí vztah

$$Q = \begin{bmatrix} 0 \\ M \end{bmatrix} = \begin{bmatrix} 0 \\ nk_m i_q \end{bmatrix}, \quad (1.4)$$

kde

M [N · m] je točivý moment působící na kolo,

n [–] je převodový poměr,

k_m [N · m/A] je momentová konstanta motoru a

$i_q [A]$ je momentotvorná složka proudu motorem.

Kinetická energie míčku je definována jako

$$T_b = \frac{1}{2} m_b (r_w + r_b)^2 \dot{\varphi}_1^2 + \frac{1}{2} J_b \dot{\varphi}_3^2, \quad (1.5)$$

kde

$m_b [kg]$ je hmotnost míčku,

$r_w [m]$ je poloměr kola,

$r_b [m]$ je poloměr míčku,

$J_b [kg \cdot m^2]$ je moment setrvačnosti míčku,

$\varphi_3 [rad]$ je úhel natočení míčku vztažený k ose y .

Moment setrvačnosti míčku J_b je získán jako moment setrvačnosti duté koule, jehož vztah je

$$J_b = \frac{2}{3} m_b r_b^2. \quad (1.6)$$

Kinetická energie kola je

$$T_w = \frac{1}{2} J_w \dot{\varphi}_2^2, \quad (1.7)$$

kde J_w je moment setrvačnosti kola, který je obtížný získat analyticky, proto bude jeho experimentální výpočet uveden v podkapitole 1.5.

Celková kinetická energie T je dána jako součet kinetických energií míčku a kola

$$T = T_w + T_b = \frac{1}{2} J_w \dot{\varphi}_2^2 + \frac{1}{2} m_b (r_w + r_b)^2 \dot{\varphi}_1^2 + \frac{1}{2} \left(\frac{2}{3} m_b r_b^2 \right) \dot{\varphi}_3^2. \quad (1.8)$$

Podmínka valení je dána jako

$$r_w \dot{\varphi}_2 - (r_w + r_b) \dot{\varphi}_1 = r_b \dot{\varphi}_3. \quad (1.9)$$

Ve vztahu 1.8 není $\dot{\varphi}_3$ přímo měřitelná, lze proto tuto veličinu získat ze vztahu 1.9.

Celková kinetická energie bude tedy

$$T = \frac{1}{2} J_w \dot{\varphi}_2^2 + \frac{1}{2} m_b (r_w + r_b)^2 \dot{\varphi}_1^2 + \frac{1}{3} m_b (r_w \dot{\varphi}_2 - r_w \dot{\varphi}_1 - r_b \dot{\varphi}_1)^2. \quad (1.10)$$

Celková potenciální energie je dána potenciální energií míčku podle vztahu 1.11

$$V = m_b g (r_w + r_b) \cos \varphi_1, \quad (1.11)$$

kde $g [m \cdot s^{-2}]$ je tíhové zrychlení.

Lagrangeova funkce poté bude mít tvar

$$L = T - V = \frac{1}{2} J_w \dot{\varphi}_2^2 + \frac{1}{2} m_b (r_w + r_b)^2 \dot{\varphi}_1^2 + \frac{1}{3} m_b (r_w \dot{\varphi}_2 - r_w \dot{\varphi}_1 - r_b \dot{\varphi}_1)^2 - m_b g (r_w + r_b) \cos \varphi_1. \quad (1.12)$$

Výpočet pohybových rovnic z Lagrangeovy funkce

$$\frac{\partial L}{\partial \varphi_1} = m_b g (r_w + r_b) \sin \varphi_1, \quad (1.13)$$

$$\frac{\partial L}{\partial \dot{\varphi}_1} = \left(\frac{5}{3} m_b r_w^2 + \frac{10}{3} m_b r_w r_b + \frac{5}{3} m_b r_b^2 \right) \dot{\varphi}_1 + \left(-\frac{2}{3} m_b r_w^2 - \frac{2}{3} m_b r_w r_b \right) \dot{\varphi}_2, \quad (1.14)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\varphi}_1} \right) = \left(\frac{5}{3} m_b r_w^2 + \frac{10}{3} m_b r_w r_b + \frac{5}{3} m_b r_b^2 \right) \ddot{\varphi}_1 + \left(-\frac{2}{3} m_b r_w^2 - \frac{2}{3} m_b r_w r_b \right) \ddot{\varphi}_2, \quad (1.15)$$

$$\frac{\partial L}{\partial \varphi_2} = 0, \quad (1.16)$$

$$\frac{\partial L}{\partial \dot{\varphi}_2} = \left(-\frac{2}{3} m_b r_w^2 - \frac{2}{3} m_b r_b r_w \right) \dot{\varphi}_1 + \left(J_w + \frac{2}{3} m_b r_w^2 \right) \dot{\varphi}_2, \quad (1.17)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\varphi}_2} \right) = \left(-\frac{2}{3} m_b r_w^2 - \frac{2}{3} m_b r_b r_w \right) \ddot{\varphi}_1 + \left(J_w + \frac{2}{3} m_b r_w^2 \right) \ddot{\varphi}_2. \quad (1.18)$$

Z 1.1 a 1.13 až 1.18 získáme pohybové rovnice

$$(5r_b + 5r_w) \ddot{\varphi}_1 - 2r_w \ddot{\varphi}_2 - 3g \sin \varphi_1 = 0, \quad (1.19)$$

$$\left(-\frac{2}{3} m_b r_w^2 - \frac{2}{3} m_b r_b r_w \right) \ddot{\varphi}_1 + \left(J_w + \frac{2}{3} m_b r_w^2 \right) \ddot{\varphi}_2 = M. \quad (1.20)$$

Při vyjádření $\ddot{\varphi}_1$ a $\ddot{\varphi}_2$ dostaneme

$$\ddot{\varphi}_1 = \frac{(2m_b r_w^2 + 3J_w) g}{(r_b + r_w) (2m_b r_w^2 + 5J_w)} \sin \varphi_1 + \frac{2r_w}{(r_b + r_w) (2m_b r_w^2 + 5J_w)} M, \quad (1.21)$$

$$\ddot{\varphi}_2 = \frac{2m_b r_w g}{2m_b r_w^2 + 5J_w} \sin \varphi_1 + \frac{5}{2m_b r_w^2 + 5J_w} M. \quad (1.22)$$

Tyto rovnice platí pouze v případě, že dostředivá síla je dostatečně velká pro udržení míčku na kole. Proto musí být splněna následující podmínka

$$g \cos \varphi_1 > (r_w + r_b) \dot{\varphi}_1^2. \quad (1.23)$$

Při maximální úhlové rychlosti míčku 4 rad/s vychází úhel $\varphi_1 = \pm 60^\circ$, proto je tato podmínka pro tento systém vždy splněna, protože se předpokládá, že se míček nebude příliš vzdalovat od nulové pozice.

1.2 Stavový popis systému

Ve stavovém vektoru se nachází úhlová pozice míčku a úhlová rychlost míčku po obvodu kola, dále úhlové natočení kola a úhlová rychlost kola.

Stavový vektor bude mít tedy tvar

$$\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4]^T = [\varphi_1 \ \dot{\varphi}_1 \ \varphi_2 \ \dot{\varphi}_2]^T. \quad (1.24)$$

Výstupem systému je úhlová pozice míčku

$$y = \varphi_1 \quad (1.25)$$

a vstupem do systému je momentotvorná složka proudu motorem

$$u = i_q. \quad (1.26)$$

Stavový popis systému je vyjádřen z rovnic 1.21 a 1.22 jako

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u) = \begin{bmatrix} x_2 \\ a \sin x_1 + bu \\ x_4 \\ c \sin x_1 + du \end{bmatrix} \quad (1.27)$$

$$y = \mathbf{g}(\mathbf{x}, u) = x_1 \quad (1.28)$$

Koeficienty a, b, c, d jsou definovány jako

$$a = \frac{2gm_b r_w^2 + 3J_w g}{(r_b + r_w)(2m_b r_w^2 + 5J_w)}, \quad (1.29)$$

$$b = \frac{2r_w k_m n}{(r_b + r_w)(2m_b r_w^2 + 5J_w)}, \quad (1.30)$$

$$c = \frac{2gm_b r_w}{2m_b r_w^2 + 5J_w}, \quad (1.31)$$

$$d = \frac{5k_m n}{2m_b r_w^2 + 5J_w}. \quad (1.32)$$

1.3 Linearizace systému

Rovnice 1.27 a 1.28 popisují nelineární dynamický systém. Aby bylo možné systém řídit metodami z oblasti lineárních systémů, je nutné tento systém linearizovat. Pro linearizaci systému byla použita lineární náhrada systému rozvojem do Taylorovy řady v okolí pracovního bodu, kdy je úhlová pozice míčku nulová. Proto musí být nulový i vstup do systému a také zbývající stavy. Je známo, že tato náhrada je platná pouze v malém okolí pracovního bodu, protože se ale předpokládá, že úhlová pozice kuličky se nebude od nulové hodnoty výrazně vzdalovat, bude tato náhrada dostačující. Rovnice linearizovaného systému lze tedy poté napsat jako

$$\begin{aligned} \Delta \dot{\mathbf{x}} &= \mathbf{A} \Delta \mathbf{x} + \mathbf{B} \Delta u, \\ \Delta y &= \mathbf{C} \Delta \mathbf{x} + \mathbf{D} \Delta u, \end{aligned} \quad (1.33)$$

$$\begin{aligned} \mathbf{A} &= \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right]_{\mathbf{x}_0, u_0} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ a & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ c & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \left[\frac{\partial \mathbf{f}}{\partial u} \right]_{\mathbf{x}_0, u_0} = \begin{bmatrix} 0 \\ b \\ 0 \\ d \end{bmatrix}, \\ \mathbf{C} &= \left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right]_{\mathbf{x}_0, u_0} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{D} = \left[\frac{\partial \mathbf{g}}{\partial u} \right]_{\mathbf{x}_0, u_0} = \begin{bmatrix} 0 \end{bmatrix}. \end{aligned} \quad (1.34)$$

1.4 Modelovací schéma systému

Stavové rovnice linearizovaného systému lze tedy napsat jako

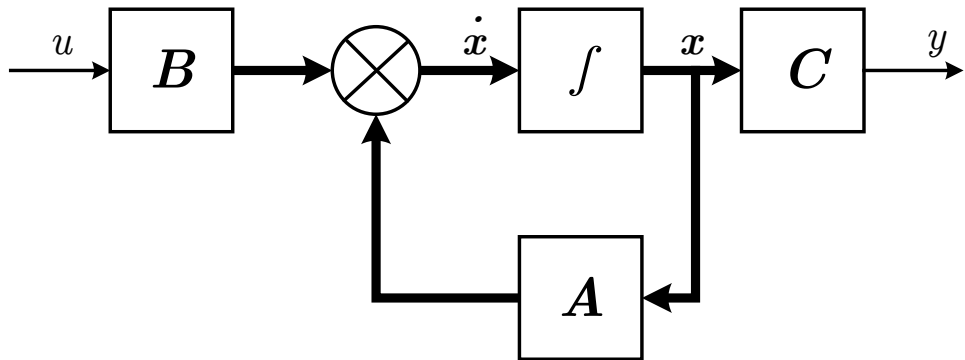
$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ a & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ c & 0 & 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ b \\ 0 \\ d \end{bmatrix} u(t), \quad (1.35)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}u(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \end{bmatrix} u(t),$$

kde

stavový vektor $\mathbf{x}(t)$ obsahuje všechny 4 stavy systému $[\varphi_1, \omega_1, \varphi_2, \omega_2]$, vstupem systému $u(t)$ je momentová složka proudu motorem a výstupem systému $y(t)$ je úhlová pozice kuličky.

Blokové modelovací schéma se nachází na obrázku 1.4. Jednotlivé matice jsou získány z údajů popsaných výše. Matice \mathbf{D} zde není vykreslena, jelikož má nulovou hodnotu, neexistuje tedy žádná vazba výstupu na vstup.



Obr. 1.4: Blokové modelovací schéma systému Ball on the wheel

1.5 Výpočet momentu setrvačnosti kola

Jedním z parametrů modelu je moment setrvačnosti kola. Protože však servomotor pomocí řemenu pohybuje kolem, bude moment setrvačnosti kola nahrazen momentem setrvačnosti celé soustavy obsahující moment setrvačnosti motoru, kola a řemenice. Pro výpočet celkového momentu setrvačnosti bylo využito rozběhu motoru konstantním momentem.

$$M = J \frac{1}{n} \frac{d\omega}{dt}, \quad (1.36)$$

kde

M [$N \cdot m$] je točivý moment motoru,

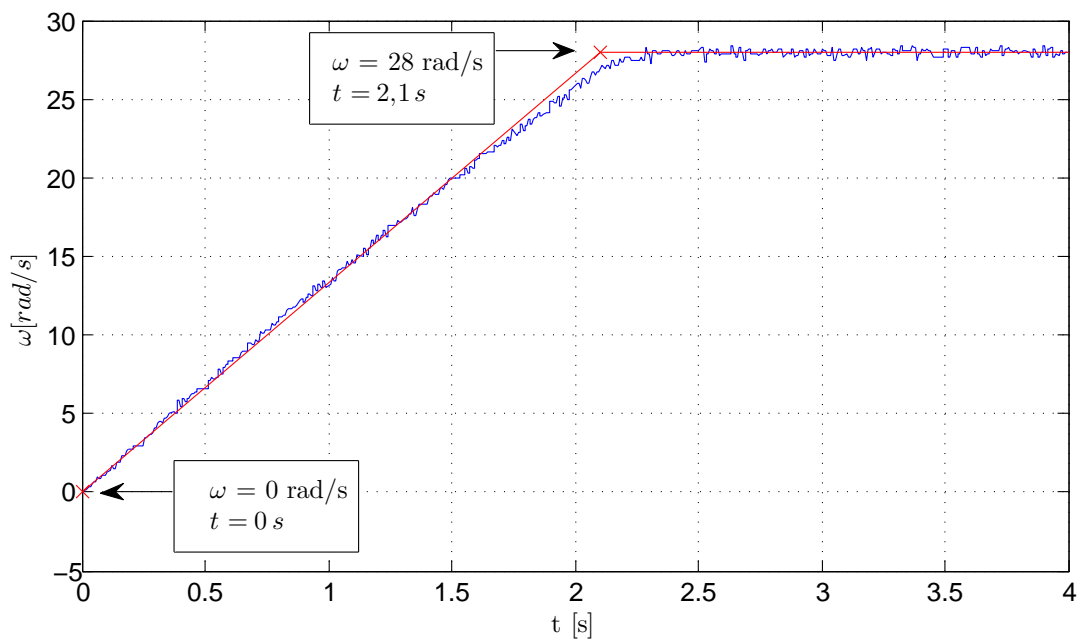
n $[-]$ je převodový poměr,

ω [rad/s] je úhlová rychlost kola,

J [$kg \cdot m^2$] je celkový moment setrvačnosti.

Pro výpočet celkového momentu setrvačnosti se vychází ze vztahu 1.36 a dosazení hodnot z grafu 1.5. Hodnoty $\Delta\omega$ a Δt byly odečteny z grafu a jsou vyznačeny červeným křížkem.

$$J = \frac{nM}{\frac{\Delta\omega}{\Delta t}} = \frac{7,7 \cdot 0,13}{\frac{28-0}{2,1-0}} = 0,075 \, kg \cdot m^2 \quad (1.37)$$



Obr. 1.5: Průběh rychlosti kola při rozběhu motoru

1.6 Výsledné parametry modelu

V tabulce 1.1 se nachází výsledné fyzikální parametry laboratorního modelu, ze kterých byly určeny koeficienty a, b, c a d . Tyto koeficienty byly dále vypočítány podle vztahů 1.29 až 1.32 a jsou uvedeny v tabulce 1.2.

Tab. 1.1: Parametry laboratorního modelu

Zkratka	Veličina	Hodnota
J_w	Moment setrvačnosti kola	$0,075 \text{ kg} \cdot \text{m}^2$
m_b	Hmotnost míčku	$0,058 \text{ kg}$
r_b	Poloměr míčku	$0,0425 \text{ m}$
r_w	Poloměr kola	$0,275 \text{ m}$
K_m	Momentová konstanta motoru	$0,13 \text{ N} \cdot \text{m/A}$
n	Převodový poměr	$7,7$
g	Tíhové zrychlení	$9,81 \text{ m} \cdot \text{s}^{-2}$

Tab. 1.2: Výsledné koeficienty laboratorního modelu

Koeficient	Hodnota
a	$18,8019$
b	$4,5183$
c	$0,8146$
d	$13,0416$

Výsledné stavové rovnice systému Ball on the wheel poté budou mít tvar

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 18,8019 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0,8146 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 4,5183 \\ 0 \\ 13,0416 \end{bmatrix} u(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}u(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \end{bmatrix} u(t). \end{aligned} \quad (1.38)$$

2 LABORATORNÍ MODEL SYSTÉMU

Konstrukce laboratorního modelu je sestavena z hliníkových profilů. V této konstrukci je uloženo hliníkové kolo použité z jízdního kola, které je přes ozubený řemen spojeno se střídavým synchronním servomotorem TGN2-0095. Ke snímání pozice míčku slouží ultrazvukový snímač vzdálenosti a fotoelektrický snímač vzdálenosti. Tyto dva snímače byly vybrány pro porovnání mezi levným řešením snímání pozice míčku a dražším řešením, jelikož pořizovací cena ultrazvukového snímače je v řádech desítek tisíc, naproti tomu cena fotoelektrického čítá řádově tisíce.

Na obrázku 2.1 se nachází vytvořený laboratorní model systému a dále jsou v této kapitole popsány jednotlivé části vytvořeného modelu.

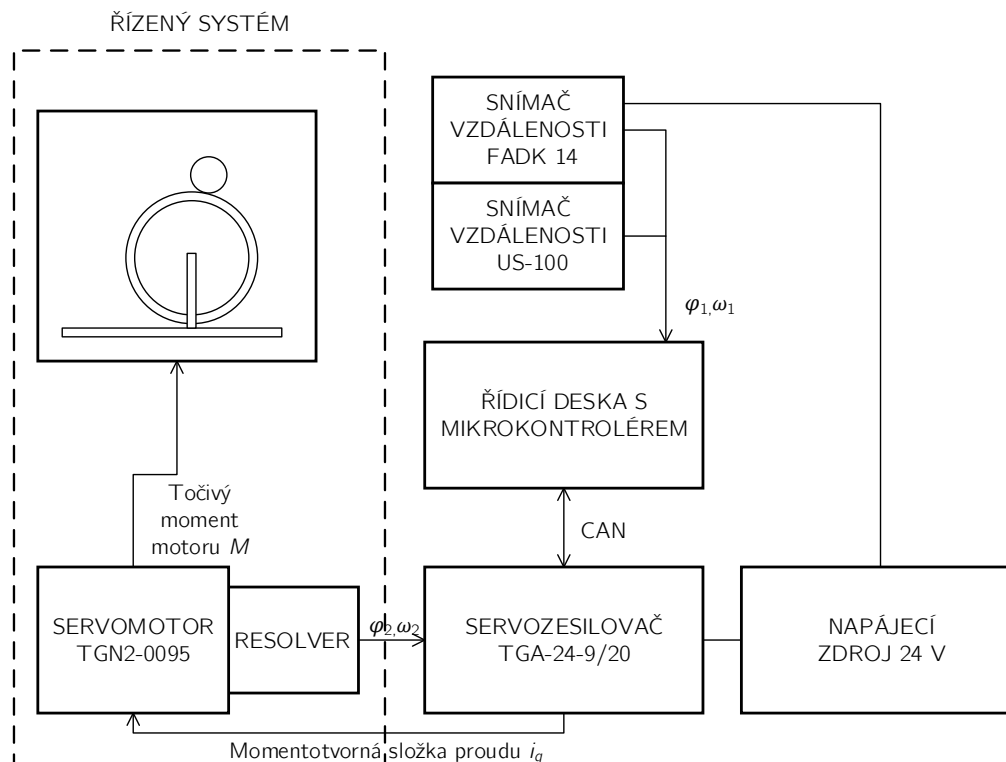


Obr. 2.1: Laboratorní model systému Ball on wheel

2.1 Celkové blokové schéma systému

Na obrázku 2.2 se nachází celkové blokové schéma systému. Napájecí zdroj 24 V slouží k napájení servozesilovače TGA-24-9/20 a fotoelektrického snímače vzdálenosti FADK 14. Servomotor je řízen servozesilovačem TGA-24-9/20, se kterým ko-

munikuje mikrokontrolér přes průmyslovou sběrnici CAN. Ke snímání pozice míčku slouží ultrazvukový snímač vzdálenosti US-100 a fotoelektrický snímač vzdálenosti FADK 14 od společnosti Baumer. Stav systému jsou zjišťovány pomocí snímačů vzdálenosti a vyčítány ze servozesilovače přes sběrnici CAN.



Obr. 2.2: Celkové blokové schéma systému

2.2 Vývojová deska mbed LPC1768

K řízení soustavy, snímání stavů soustavy a komunikaci se servozesilovačem slouží vývojová deska mbed NXP LPC1768, která byla navržena pro vytváření mnoha druhů aplikací. Deska je založena na výkonném procesoru NXP LPC1768 s 32-bitovým jádrem ARM Cortex-M3. [4]

2.2.1 Vlastnosti vývojové desky

- ARM ®Cortex™- M3
- 96 MHz, 32KB RAM, 512KB FLASH
- Ethernet, USB Host/Device
- 2×SPI, 2×I²C, 3×UART, CAN, 6×PWM, 6×ADC, GPIO

<div> <div> mbed LPC 1768 </div> <div> mbed-rtos </div> </div>	CAN
	AnalogIn
	InterruptIn
	DigitalIn
	DigitalOut

Obr. 2.3: Používané rozhraní

Vývojovou desku je možné napájet 5 V USB nebo 4,5 – 9 V. Je využíváno C/C++ SDK, kde probíhá programování na vyšší úrovni. K dispozici je také online kompilátor, publikované knihovny a projekty.

2.3 Digitální servozesilovač TGA-24-9/20

K řízení střídavého synchronního servomotoru byl použit digitální servozesilovač TGA-24-9/20 od společnosti TG drives [5]. Tento servozesilovač obsahuje výkonný procesor, který zajišťuje kvalitní řízení servomotorů a také obsahuje mnoho uživatelských funkcí, umožňující různá využití servozesilovače. Obsahuje digitální proudovou smyčku pracující s frekvencí 16 kHz, rychlostní regulační smyčku s frekvencí 4kHz a polohovou regulační smyčku s frekvencí 1kHz. Servozesilovač obsahuje 10 digitálních vstupů, 4 digitální výstupy, 1 analogový vstup a 2 konfigurovatelné vstupy. Tyto vstupy a výstupy však využity nebyly, protože pro komunikaci se servozesilovačem je využita sběrnice CAN.

2.3.1 Operační režimy servozesilovače

Servozesilovač TGA-24-9/20 je možné provozovat v následujících režimech:

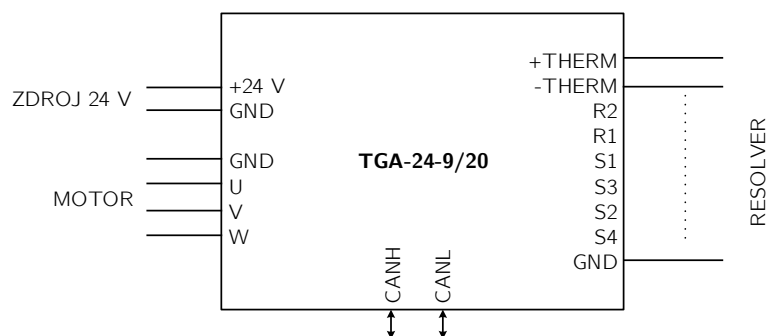
- momentové řízení,
- rychlostní řízení,
- absolutní polohování,
- relativní polohování,
- řízení signály *KROK* a *SMĚR*,
- elektronická převodovka,
- souvislé řízení po sběrnici CAN.



Obr. 2.4: Servozesilovač TGA-24-9/24

Momentové řízení se dělí na analogové momentové řízení a digitální momentové řízení. V této práci bylo využito digitálního momentového řízení, kdy je požadovaný moment zadáván přes sběrnici CAN.

Na obrázku 2.5 se nachází schéma zapojení servozesilovače. Podrobné schéma zapojení servozesilovače TGA-24-9/20 lze nalézt v [6].



Obr. 2.5: Zapojení servozesilovače TGA-24-9/24

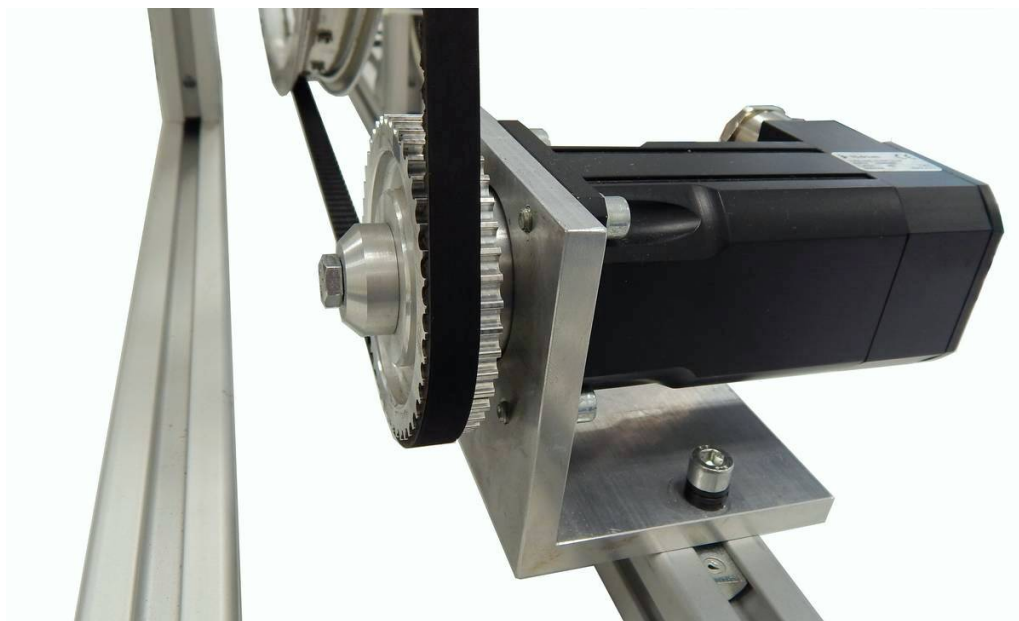
2.3.2 Komunikace se servozesilovačem

Se servozesilovačem je možné komunikovat po sériové lince RS 232 a po sběrnici CAN. Dále je také možné komunikovat i po sériové lince RS 422/485 s protokolem

MODBUS. V této práci probíhá komunikace se servozesilovačem po sběrnici CAN. Princip, jak tato komunikace probíhá, je podrobněji popsán v kapitole 3.

2.4 Střídavý synchronní servomotor TGN2-0095

K pohonu kola slouží střídavý synchronní servomotor TGN2-0095, stejně jako servozesilovač od společnosti TG drives. Jedná se o střídavý synchronní servomotor, který má ve statoru rozložené třífázové vinutí a na rotoru magnety ze vzácných zemin neodymželezobór. K přesnému odměřování polohy a úhlové rychlosti rotoru slouží 2-pólový resolver [7]. Servomotor je pevně upevněn na hliníkové konstrukci viz obrázek 2.6.



Obr. 2.6: Ukotvení servomotoru TGN2-0095

2.4.1 Parametry servomotoru

Tab. 2.1: Parametry synchronního servomotoru TGN2-0095 [7]

Veličina	Zkratka	Hodnota	Jednotka
Klidový moment	M_0	0,95	$N \cdot m$
Klidový proud	I_0	7,3	A
Jmenovitý moment	M_N	0,93	$N \cdot m$
Jmenovitý proud	I_N	7,5	A
Jmenovité otáčky	n_N	1500	min^{-1}
Maximální moment	M_{max}	3,1	$N \cdot m$
Maximální proud	I_{max}	25	A
Maximální otáčky	n_{MAX}	12 000	min^{-1}
Momentová konstanta	K_M	0,13	$N \cdot m/A$
Napěťová konstanta	K_E	7,8	$V/1000 min^{-1}$
Odpor dvě fáze	R_{2Ph}	0,82	Ω
Indukčnost dvě fáze	L_{2Ph}	1,18	mH
Počet pólů motoru	2_p	6	-
Vlastní moment setrvačnosti	J	0,11	$kg \cdot cm^2$

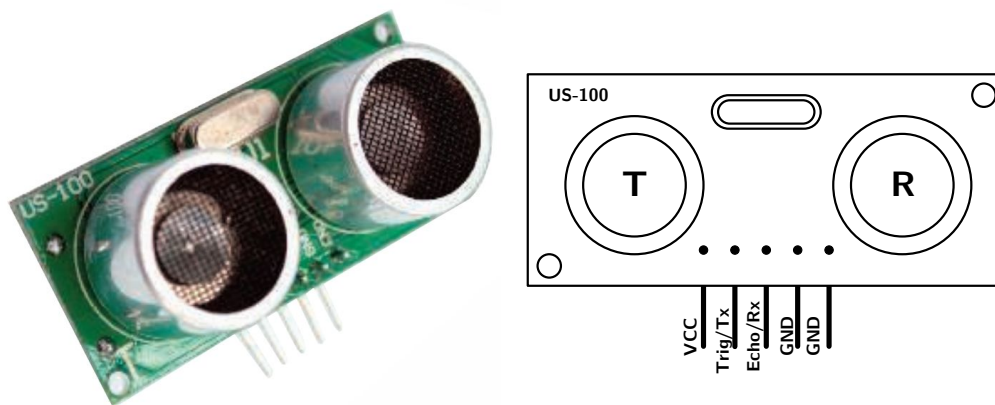
2.5 Použité senzory pro měření pozice míčku

2.5.1 Ultrazvukový snímač vzdálenosti US-100

Jako levnější varianta měření pozice míčku byl použit ultrazvukový snímač vzdálenosti US-100 [8], jehož pořizovací cena, jak již bylo výše zmíněno, je čítána v řádech desítek korun. Princip senzoru spočívá v přivedení logické 1 na pin *Trig* po dobu minimálně $10 \mu s$. Senzor poté vyšle krátké pulzy ultrazvukových vln a automaticky detekuje příchozí pulz. Dále na výstupním pinu *Echo* bude logická 1 po dobu odpovídající času mezi vysláním a přijetím pulzu.

Vlastnosti snímače

- **Napájecí napětí:** 5 V
- **Měřicí rozsah:** 2 cm – 450 cm
- **Klidový proud:** $< 2 mA$
- **Přesnost:** do 1 mm
- **Snímací úhel:** do 15°



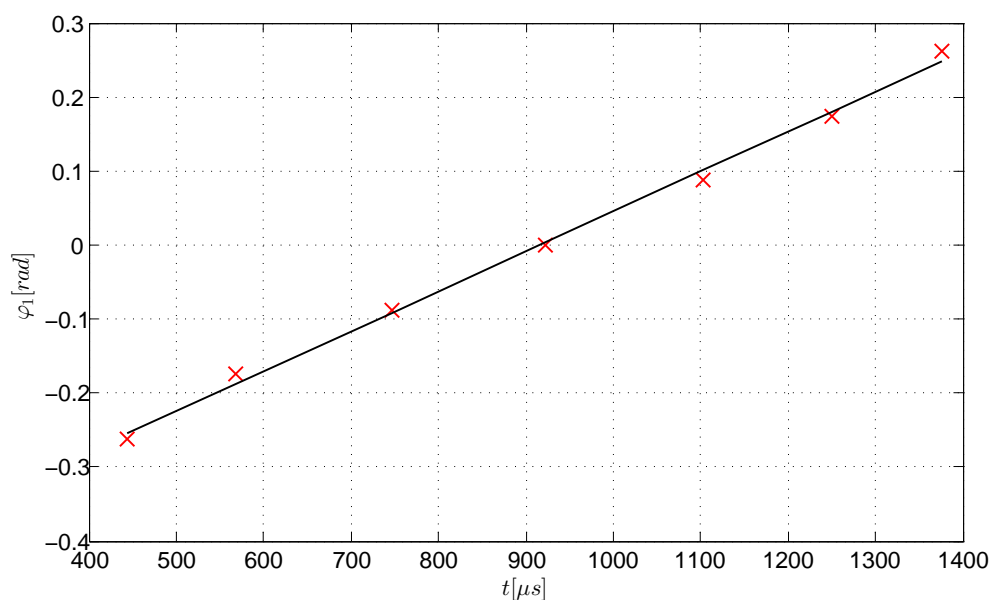
(a) Ultrazvukový snímač [8]

(b) Zobrazení pinů

Obr. 2.7: Ultrazvukový snímač vzdálenosti US-100

Měření pozice míčku pomocí ultrazvukového snímače

Při měření úhlové pozice míčku pomocí ultrazvukového snímače byl míček pevně umístěn na kolo, které bylo rotováno o požadovaný úhel. Mikrokontrolérem byla měřena šířka pulzu přijatá od snímače. Závislost úhlové pozice míčku na této šířce pulzu je znázorněna v grafu 2.8.



Obr. 2.8: Závislost úhlové pozice míčku na šířce přijatého pulzu

Jednotlivé naměřené body bylo možné proložit přímkou s rovnicí

$$\varphi_1 = 0,0004865t - 0,4912. \quad (2.1)$$

2.5.2 Fotoelektrický snímač vzdálenosti FADK 14



(a) Snímač vzdálenosti FADK 14

(b) Zobrazení pinů

Obr. 2.9: Fotoelektrický snímač vzdálenosti FADK 14

Dalším senzorem, který slouží pro měření pozice míčku, je fotoelektrický snímač vzdálenosti FADK 14U4470/S35A/IO od společnosti Baumer [9]. Senzor pracuje s červenou pin point LED diodou na triangulačním principu. Snímače z řady FADK 14 jsou dodávány s proudovým či napěťovým výstupem, pro tuto aplikaci byl vybrán snímač s výstupem napěťovým. Výstupní napětí snímače je podle polohy snímaného objektu v rozmezí 0 – 10 V, proto je nutné toto napětí předzpracovat, jelikož analogový vstup mikrokontroléru pracuje s napětím v rozmezí 0 – 3,3 V.

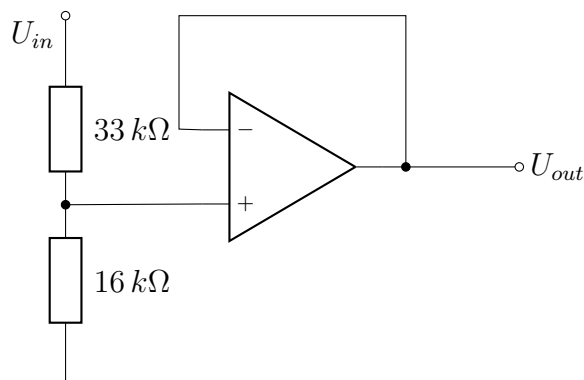
Vlastnosti snímače

- **Napájecí napětí:** 14 – 26 V
- **Měřicí rozsah:** 50 mm – 400 mm
- **Doba odezvy:** < 3 ms
- **Rozlišení:** 0,1 – 1 mm
- **Analogový výstup:** 0 – 10 V

Měření pozice míčku pomocí fotoelektrického snímače

Výstupní napětí fotoelektrického snímače polohy je podle polohy snímaného objektu v rozmezí 0 – 10 V. Napětí ze snímače je dále pomocí děliče napětí s operačním

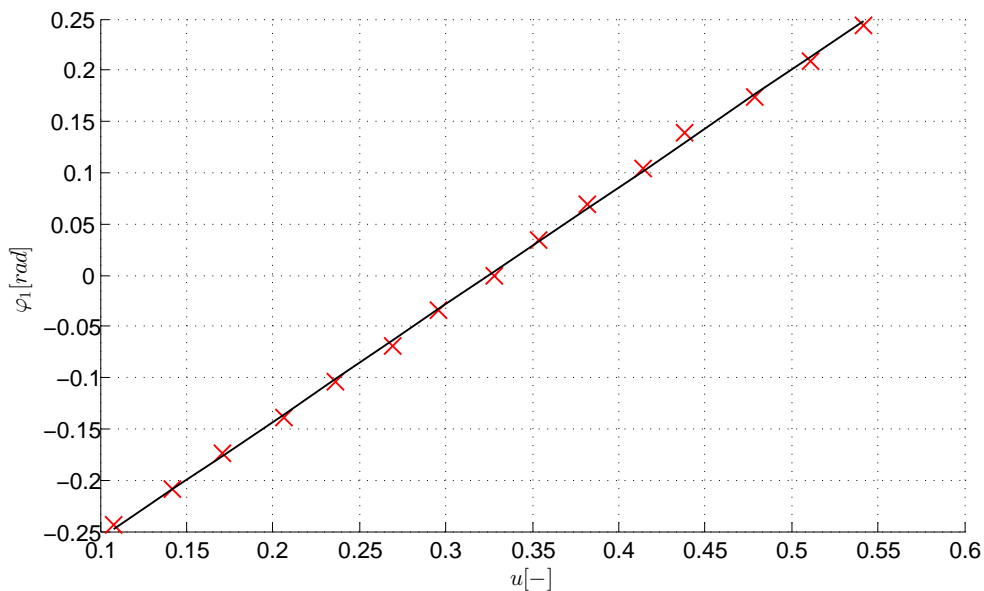
zesilovačem zmenšeno na hodnoty $0 - 3,3\text{ V}$, viz obrázek 2.10. Toto napětí dále



Obr. 2.10: Napěťový dělič s operačním zesilovačem

zpracovává analogový vstup mikrokontroléru, který ho normalizuje na rozsah čísel $0 - 1$.

Měření úhlové pozice míčku bylo prováděno obdobným způsobem jako u ultrazvukového snímače, pouze s tím rozdílem, že nebyla měřena šířka pulzu ale analogová hodnota napětí. Závislost úhlové pozice míčku na velikosti normalizovaného napětí je znázorněna v grafu 2.11.



Obr. 2.11: Závislost úhlové pozice míčku na normalizovaných hodnotách napětí

Jak je z grafu vidět, je tato závislost lineární a lze tedy napsat rovnici proložené přímkou jako

$$\varphi_1 = 1,1439u - 0,3718. \quad (2.2)$$

Oba senzory jsou pro lepší kompaktnost uloženy ve vytvořeném nerezovém pouzdře 2.12, u kterého lze nastavovat výšku upevnění pro právě používaný snímač.



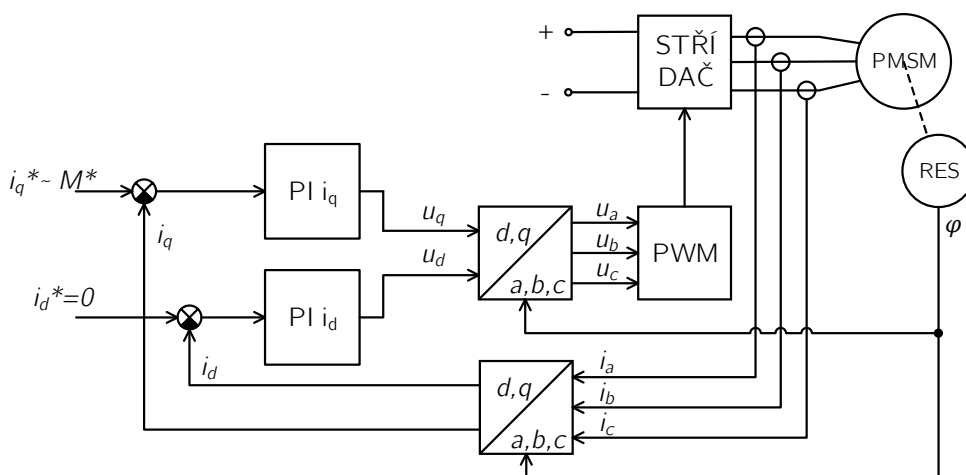
Obr. 2.12: Uložení senzorů v nerezovém pouzdře

3 SERVOMOTOR A SERVOZESILOVAČ

Pro správnou funkčnost servomotoru bylo nutné správně nastavit servozsilovač. To bylo prováděno pomocí ovládacího softwaru TGA-24-S.C.D. od společnosti TG drives. Ten komunikuje se servozilovačem pomocí sériové linky RS-232. Byly zde zadány parametry motoru a také konstanty regulátorů proudu, které jsou popsány níže. Dále již probíhala komunikace mezi deskou mbed a servozsilovačem po sběrnici CAN. Tato komunikace je popsána v podkapitole 3.3. Níže je popsáno jakým způsobem byly získány konstanty regulátorů proudu.

3.1 Regulace momentu v d-q souřadnicích

Na obrázku 3.1 se nachází blokové schéma regulace momentu v $d - q$ souřadnicích. Při požadavku na regulaci rychlosti motoru by dále přibyla nadřazená rychlostní smyčka a dále pak při požadavku na polohu také polohová smyčka. Tato regulační smyčka obsahuje dvě nezávislé proudové smyčky, také tedy dva regulátory proudu typu PI. Tyto dva proudy se dělí na momentotvornou složku proudu i_q a tokotvornou složku i_d . Referenční hodnota d složky proudu je nulová a požadovaná hodnota q složky proudu je úměrná požadovanému momentu.



Obr. 3.1: Regulace momentu v d-q souřadnicích

3.2 Nastavení proudových regulátorů

Při ukotveném rotoru motoru je možné fázový model servomotoru ve složkách d a q získat jako sériové spojení odporu R_s a indukčnosti L , viz obrázek 3.2.

Takto je možné získat perátorový přenos RL článku pro d a q souřadnice

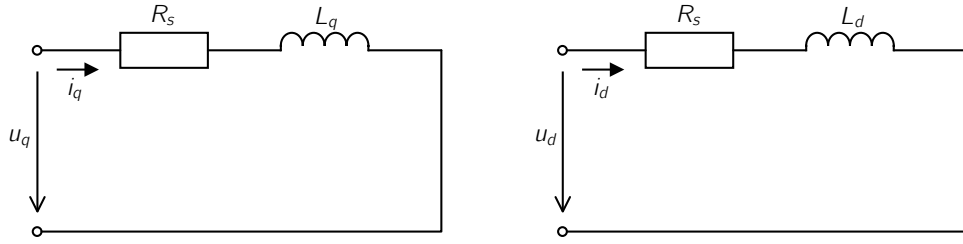
$$F_d(p) = \frac{\frac{1}{R_s}}{\frac{L_d}{R_s}p + 1} \quad F_q(p) = \frac{\frac{1}{R_s}}{\frac{L_q}{R_s}p + 1}, \quad (3.1)$$

kde

R_s $[\Omega]$ je odpor statorového vinutí,

L_d $[H]$ je indukčnost v ose d ,

L_q $[H]$ je indukčnost v ose q .



Obr. 3.2: Ekvivalentní fázový model servomotoru v d-q souřadnicích

Operátorový přenos PI regulátoru proudu bude mít tvar

$$F_{PI}(p) = K_R \frac{T_i p + 1}{p} = K_p + \frac{K_i}{p}. \quad (3.2)$$

Časová konstanta PI regulátoru proudu kompenzuje časovou konstantu RL článku, tedy

$$T_i = \frac{L_q}{R_s}. \quad (3.3)$$

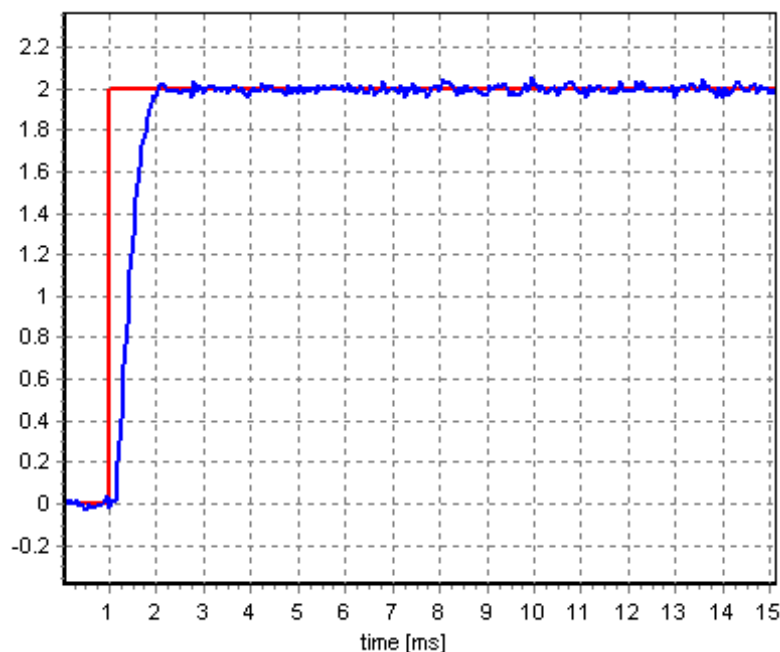
Operátorový přenos otevřené smyčky poté bude

$$F_0(p) = K_R \frac{\frac{1}{R_s}}{p}. \quad (3.4)$$

Jelikož však detailní struktura regulátorů proudu od společnosti TG drives nebyla získána a návrh regulátorů podle výše uvedeného principu nebyl úspěšný, protože konstanty regulátorů byly nerealizovatelné. Konstanty regulátorů byly tedy získány experimentálně s ohledem na nulovou ustálenou odchylku, minimální překmit a co nejrychlejší přechodový děj. Odezva složky proudu v q ose na žádanou skokovou změnu q složky proudu je znázorněna na obrázku 3.3, který byl získán pomocí softwaru TGA-24-S.C.D. Červený průběh znázorňuje požadovaný proud a modrým průběhem je aktuální proud motorem. Nastavené konstanty regulátorů proudu mají hodnoty

$$K_p = 80,$$

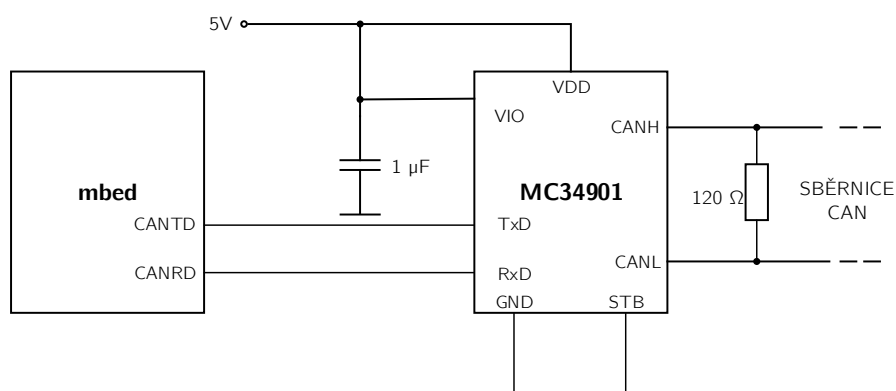
$$K_i = 85.$$



Obr. 3.3: Odezva q složky proudu na skokovou změnu žádané hodnoty proudu

3.3 Komunikace na sběrnici CAN

Jako fyzické rozhraní mezi mikrokontrolérem a sběrnici CAN byl použit High speed CAN budič MC34901 od společnosti Freescale semiconductor [10]. Schéma zapojení CAN budiče, které vychází z doporučeného zapojení výrobce, je znázorněno na obrázku 3.4. Přenosová rychlost sběrnice je zvolena 500kB/s.



Obr. 3.4: Schéma zapojení CAN budiče

Dále je popsán komunikační protokol na sběrnici CAN, který je využíván pro řízení servozesilovače [11]. Jako základ je využit protokol CANopen, který je zjednodušen,

avšak je s tímto protokolem kompatibilní. Protokol obsahuje dva typy zpráv - *Parametrizační datový objekt (SDO)* a *Procesní datový objekt (PDO)*, jejichž struktura je popsána níže.

3.3.1 Parametrizační datový objekt - SDO

Struktura parametrizačního datového objektu je uvedena v tabulce 3.1. Dále je tato struktura modifikována podle typu zprávy.

Tab. 3.1: Struktura zprávy SDO

ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x600+ID 0x580+ID	Příkaz	Adresa ⁽⁰⁾	Adresa ⁽¹⁾	Velikost dat	Data ⁽⁰⁾	Data ⁽¹⁾	Data ⁽²⁾	Data ⁽³⁾

ID je adresa servozesilovače v rozsahu 0x01-0x3F. V práci je adresa servozesilovače nastavena na hodnotu 0x01. Tento objekt je možné vyčítat, k čemuž slouží zpráva *Požadavek na vyčtení*, servozesilovač poté do 2 ms vyšle zprávu *Odpověď na požadavek na vyčtení*. Tento typ zprávy je v práci použit zejména pro vyčtení aktuální úhlové rychlosti kola. Dále je ho možné modifikovat pomocí zprávy *Požadavek na zápis*. Při řízení modelu je tento objekt používán pro nastavení módu serva, nulování resolveru a zapínání či vypínání serva.

Požadavek na vyčtení

Tab. 3.2: Struktura zprávy požadavku na vyčtení

ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x600+ID	0x40	Adresa ⁽⁰⁾	Adresa ⁽¹⁾	Velikost	0	0	0	0

Odpověď na požadavek na vyčtení

Tab. 3.3: Struktura zprávy odpovědi na požadavek na vyčtení

ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x580+ID	0x4B/0x43	Adresa ⁽⁰⁾	Adresa ⁽¹⁾	Velikost	Data ⁽⁰⁾	Data ⁽¹⁾	Data ⁽²⁾	Data ⁽³⁾

Požadavek na zápis

Tab. 3.4: Struktura zprávy požadavku na zápis

ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x600+ID	0x22	Adresa ⁽⁰⁾	Adresa ⁽¹⁾	Velikost	Data ⁽⁰⁾	Data ⁽¹⁾	Data ⁽²⁾	Data ⁽³⁾

Odpověď na požadavek na zápis

Tab. 3.5: Struktura zprávy odpovědi na požadavek na zápis

ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x580+ID	0x60	Adresa ⁽⁰⁾	Adresa ⁽¹⁾	Velikost	0	0	0	0

3.3.2 Procesní datový objekt - PDO

Procesní datový objekt slouží k řízení servomotoru. Existují dva typy objektů, kde prvním z nich je příjmový PDO, pomocí kterého lze v módu momentového řízení nastavovat požadovaný proud motorem a také nastavovat digitální výstupy. Druhým objektem je vysílací PDO, pomocí něhož je získávána poloha resolveru a skutečný proud motorem.

Příjmový procesní datový objekt pro servozesilovač - RPDO

Při režimu souvislého momentového řízení vypadá struktura RPDO podle tabulky 3.6. Žádaný proud se vysílá v rozsahu hodnot $\langle -14348; 14348 \rangle$, které odpovídají

Tab. 3.6: Struktura RPDO v režimu souvislého momentového režimu

ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x200+ID	Žádaný proud		Rezervováno		Digitální výstupy	

maximálními hodnotám proudu 10 A a -10 A. V tabulce 3.7 je uveden příklad zprávy pro nastavení žádaného proudu 3 A, který odpovídá hodnotě 4304 (0x10D0).

Tab. 3.7: Příklad zprávy RPDO pro nastavení požadovaného proudu 3 A

ID	Data							
0x181	D0	10	00	00	00	00	00	00

Vysílací procesní datový objekt od servozesilovače - TPDO

Tab. 3.8: Struktura TPDO v režimu souvislého momentového režimu

ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x180+ID	Stav	Aktuální poloha resolveru				Aktuální proud do motoru		Digitální vstupy

Byte 0 obsahuje informace o aktuálním stavu servopohonu:

- Bit 0 - výkonový most je aktivní,
- Bit 1 - servo je v žádané poloze,
- Bit 2 - souvislé řízení po sběrnici CAN je aktivní,
- Bit 3 - nulová rychlost,
- Bit 4 - koncový spínač 1,
- Bit 5 - koncový spínač 2,
- Bit 6 - porucha.

3.3.3 Objekt SYNC

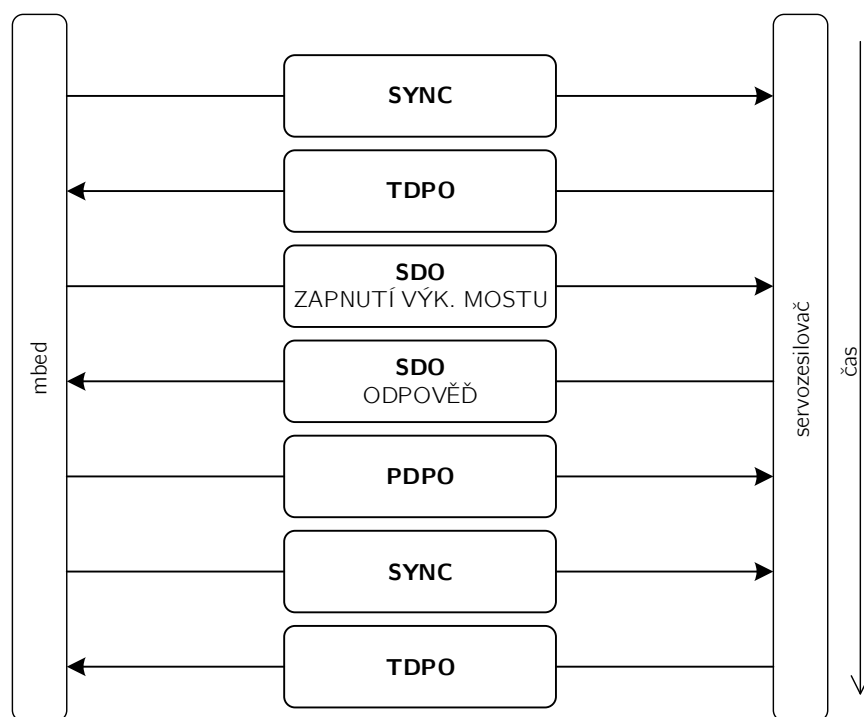
Jestliže servozesilovač přijme objekt SYNC, předá hodnoty přijaté přes RPDO do regulátorů a zpět odešle TPDO s aktuálními hodnotami. Objekt SYNC je v této práci posílán servozesilovači každé 2 ms.

Tab. 3.9: Struktura SYNC

ID
0x80

3.3.4 Příklad komunikace

Na obrázku 3.5 se nachází příklad komunikace na sběrnici CAN. Každé 2 ms je vysílán servozesilovači objekt SYNC, kdy poté servozesilovač odpoví objektem TPDO s aktuálními hodnotami. Je zde ukázka zapnutí výkonového mostu pomocí objektu SDO - požadavek na zápis. Servozesilovač vyšle SDO - odpověď na požadavek na zápis.



Obr. 3.5: Příklad komunikace po sběrnici CAN

4 ŘÍZENÍ SYSTÉMU BALL ON THE WHEEL

V kapitole 1 byl sestaven model systému Ball on the wheel včetně jeho stavového popisu a následné linearizace. Výsledný model systému byl popsán stavovými rovnicemi 1.38 a poté již bylo možné přistoupit k návrhu řízení systému. K tomu bylo využito programové prostředí MATLAB 2012 s nadstavbou Simulink.

4.1 Stavová zpětná vazba

Systém Ball on the wheel obsahuje celkem 4 stavy, které jsou všechny měřitelné. Je proto vhodné pro řízení systému použít stavovou zpětnou vazbu, kde lze za jistých předpokladů měnit libovolně dynamiku původního systému. Na obrázku 4.1 je blokové schéma stavové zpětné vazby a níže je popsán princip stavové zpětné vazby.

Je dán systém popsáný stavovými rovnicemi 4.1

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t),\end{aligned}\tag{4.1}$$

jehož dynamické vlastnosti jsou dány jeho póly, které jsou rovny vlastním číslům matice \mathbf{A} . Je-li zavedena zpětná vazba od vektoru stavů, jak je znázorněno na obrázku 4.1, bude novým vstupem do systému

$$\mathbf{u}(t) = \mathbf{w}(t) - \mathbf{v}(t) = \mathbf{w}(t) - \mathbf{K}\mathbf{x}(t).\tag{4.2}$$

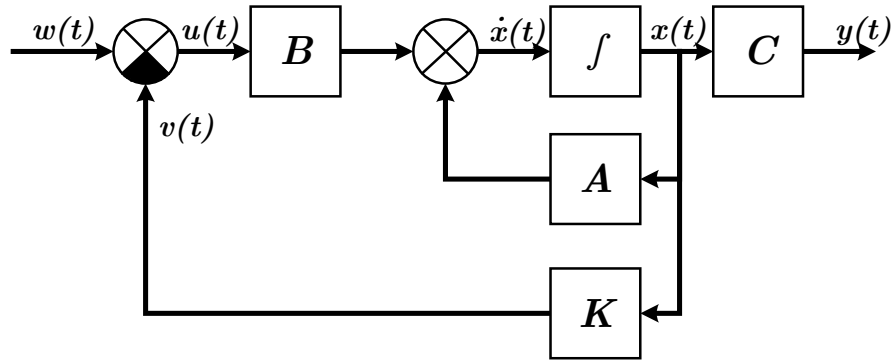
Stavové rovnice celého systému i se zpětnou vazbou budou mít tvar

$$\dot{\mathbf{x}}(t) = (\mathbf{A} - \mathbf{BK})\mathbf{x}(t) + \mathbf{B}\mathbf{w}(t).\tag{4.3}$$

Nyní je nová matice zpětných vazeb dána jako

$$\mathbf{A}' = \mathbf{A} - \mathbf{BK}\tag{4.4}$$

a dynamika celého systému se zpětnou vazbou je dána vlastními čísly matice \mathbf{A}' . Z toho vyplývá, že vhodnou volbou matice \mathbf{K} lze měnit vlastní čísla matice \mathbf{A}' a tím i dynamiku celého systému. [12], [13]



Obr. 4.1: Blokový diagram stavové zpětné vazby

4.2 Měření stavů systému Ball on the wheel

K řízení systému Ball on the wheel je nutné mít k dispozici všechny stavy systému. V následujících 4 podkapitolách jsou popsány metody měření či výpočtu jednotlivých stavů.

4.2.1 Výpočet úhlu pozice míčku

Prvním stavem systému a zároveň i výstupem je úhlová pozice míčku vztažená k ose y . Pro výpočet úhlové pozice míčku byly použity vztahy 2.8 a 2.11 na základě právě používaného senzoru.

4.2.2 Výpočet úhlové rychlosti míčku

Dalším stavem systému, který je nutný mít k dispozici, je úhlová rychlost míčku. Výpočet úhlové rychlosti míčku vychází z toho, že pozice míčku umístěného na kole je již známá. Úhlová rychlost míčku poté bude určena výpočtem difference

$$\omega_1 = \frac{\Delta\varphi_1}{\Delta t} = \frac{\varphi_1(k) - \varphi_1(k-1)}{T}, \quad (4.5)$$

kde T je perioda vzorkování 15 ms .

Tento jednoduchý výpočet difference s sebou nese nevýhodu z toho důvodu, že i malý šum snímače vydělený malou hodnotou již není zanedbatelný. Z toho důvodu byla úhlová rychlost míčku dále filtrována. Pro filtraci byl využit exponenciální klouzavý průměr

$$\omega_1(k) = \alpha x(k) + (1 - \alpha)\omega_1(k-1), \quad (4.6)$$

kde

$\omega_1(k)$ je filtrovaná hodnota úhlové rychlosti míčku v aktuálním kroku,
 $\omega_1(k-1)$ je předchozí hodnota filtrované úhlové rychlosti míčku,
 $x(k)$ je aktuální hodnota úhlové rychlosti míčku,
 α je vyhlazovací koeficient.

Vyhlažovací koeficient $\alpha = 0,2$ byl zvolen s ohledem na kvalitu filtrace a vneseného dopravního zpoždění.

4.2.3 Měření úhlového natočení kola

Třetím stavem systému je úhlové natočení kola. Servozesilovač s každým přijetím SYNC vyše procesní datový blok PDO obsahující aktuální polohu resolveru v inkrementech. Tuto hodnotu je dále nutné převést z polohy rotoru na polohu kola a také na výslednou hodnotu v radiánech.

$$\varphi_2 = \frac{0,01745}{1401,7422} inc, \quad (4.7)$$

kde konstanta 0,01745 je použita pro přepočtení stupňů na radiány. Dále 1 otáčka motoru, tedy 360° , odpovídá 65536 *inc*. Z této skutečnosti byla vynásobením převodovým poměrem určena konstanta 1401,7422.

4.2.4 Zjištění úhlové rychlosti kola

Posledním stavem, který je nutné pro stavovou zpětnou vazbu mít k dispozici, je úhlová rychlost kola ω_2 . Tu lze zjistit vysláním SDO požadavku servozesilovači na vyčtení okamžité rychlosti, servozesilovač poté do 2 *ms* vyše SDO odpověď s aktuální hodnotou rychlosti.

4.3 Navržený model systému

V podkapitole 1.6 byly získány stavové rovnice systému. Tyto rovnice mají tvar

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}u(t), \end{aligned}$$

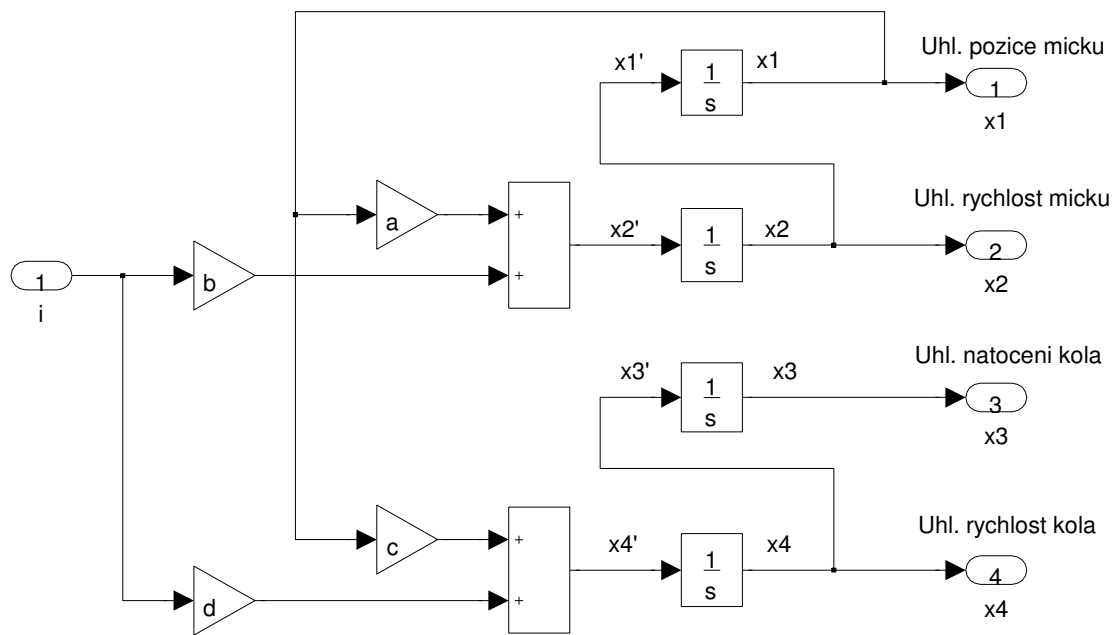
kde tvar jednotlivých matic je popsán níže.

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ a & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ c & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 18,8019 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0,8146 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ b \\ 0 \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 4,5183 \\ 0 \\ 13,0416 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} 0 \end{bmatrix}$$

Podle získaných stavových rovnic systému bylo sestaveno modelovací schéma systému Ball on the wheel, které se nachází na obrázku 4.2.



Obr. 4.2: Model systému Ball on the wheel

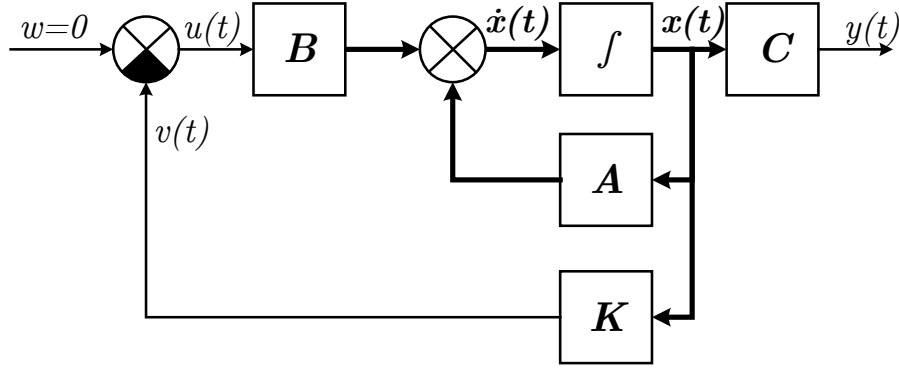
Příkazem `eig` lze získat vlastní čísla matice \mathbf{A} čili póly systému, které jsou uvedeny v tabulce 4.1. Jak je z tabulky 4.1 patrné, tak jeden z pólů systému je kladný, což bylo očekáváno, protože se jedná o nestabilní systém.

Tab. 4.1: Vlastní čísla matice \mathbf{A}

Pól	Hodnota
$p_{1,2}$	0
p_3	4,336
p_4	-4,336

4.4 Návrh řízení systému pomocí stavové zpětné vazby

Pro návrh zpětnovazebního řízení systému Ball on the wheel se vychází ze schématu na obrázku 4.3. Důležité při návrhu řízení systému je zjistit, zda je tento systém plně



Obr. 4.3: Schéma řízení systému pomocí stavové zpětné vazby

řiditelný, tedy zda jsou všechny stavy systému řiditelné. Aby byl systém řiditelný, musí mít matice řiditelnosti

$$\mathbf{H} = [\mathbf{B} \quad \mathbf{AB} \quad \mathbf{A}^2\mathbf{B} \quad \mathbf{A}^3\mathbf{B}] \quad (4.8)$$

hodnost shodnou s řádem soustavy, tedy tato matice musí mít hodnost 4. Výpočet matice řiditelnosti byl proveden pomocí příkazu $\mathbf{H} = \text{ctrb}(\mathbf{A}, \mathbf{B})$ a dále byla určena hodnost této matice příkazem $\text{rank}(\mathbf{H}) = 4$. Výsledkem byla hodnost matice \mathbf{H} shodná s řádem soustavy, z toho vyplývá, že systém je plně řiditelný. Poté již bylo možné přejít k návrhu řízení. Ze schématu 4.3 je patrné, že stavová rovnice bude mít tvar

$$\dot{\mathbf{x}}(t) = (\mathbf{A} - \mathbf{BK}) \mathbf{x}(t) + \mathbf{B}w(t), \quad (4.9)$$

jelikož vstup soustavy $w = 0$, přejde stavová rovnice do tvaru

$$\dot{\mathbf{x}}(t) = (\mathbf{A} - \mathbf{BK}) \mathbf{x}(t) = \mathbf{A}' \mathbf{x}(t), \quad (4.10)$$

kde $\mathbf{A}' = \mathbf{A} - \mathbf{BK}$ je nová matice zpětných vazeb. Dynamické vlastnosti soustavy jsou nyní dány vlastními čísly matice \mathbf{A}' , která jsou určena řešením algebraické rovnice

$$\det(p\mathbf{I} - \mathbf{A} + \mathbf{BK}) = 0. \quad (4.11)$$

K získání zpětné vazby je proto nutné vhodně zvolit poměrově velikosti jednotlivých pólů, které jsou určeny podle níže uvedených kritérií:

- Pól, který je příslušný ke stavu φ_1 , by měl mít poměrně vysokou hodnotu, aby byl regulátor schopný rychle reagovat na změny úhlové pozice míčku.
- Pól, který je příslušný ke stavu ω_1 , by měl mít také vyšší hodnotu, aby byl regulátor schopen rychleji reagovat na změny úhlové rychlosti míčku.
- Póly příslušné k φ_2 a ω_2 by měli mít nižší hodnoty, aby regulátor nereagoval rychle na změnu natočení kola a svým zásahem tak nezpůsobil pád míčku z kola.

Dále pro výpočet zesílení \mathbf{K} byla zvolena metoda, kde se minimalizuje daná váhová funkce. A přímá volba velikosti pólů přechází k volbě vah pro jednotlivé stavové proměnné.

4.5 Návrh lineárně kvadratického řízení

V předcházejících podkapitolách bylo vysvětleno, jak se měří všechny stavy systému. K dispozici jsou tedy všechny 4 stavy systému a je použita stavová zpětná vazba

$$u = -\mathbf{K}\mathbf{x}, \quad (4.12)$$

která se snaží řídit tyto stavy směrem k nule. Problémem řešení stavové zpětné vazby je nalézt vektor \mathbf{K} . K hledání zesílení zpětné vazby \mathbf{K} bylo využito lineárně kvadratického řízení, kde se hledá takové \mathbf{K} , které minimalizuje váhovou funkci 4.13.

$$J = \int_0^\infty (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \quad (4.13)$$

Výsledkem minimalizace váhové funkce J je pohyb stavových veličin k nule s co nejmenší odchylkou stavových veličin od referenční hodnoty a spotřebovanou energií. Pomocí váhových matic \mathbf{Q} a \mathbf{R} lze tak stanovit poměr mezi odchylkou stavových veličin od referenční hodnoty a spotřebovanou energií. V případě systému Ball on the wheel, který obsahuje celkem 4 stavy se volí váhová matice \mathbf{Q} rozměru 4×4 s prvky $q_i \geq 0$ na hlavní diagonále. Tyto prvky q_i určují váhy vzhledem k příslušným stavovým proměnným, ale také ke zvolené matici \mathbf{R} . Váhová matice \mathbf{R} je v tomto případě skalár $R > 0$. Návrhem u lineárně kvadratického řízení je tedy vhodné nalezení váhových matic \mathbf{Q} a R . Například v případě $q_i = 0$ není kladen žádný důraz na odchylky jednotlivých stavových proměnných, zatímco regulátor se snaží udržet stavové veličiny na nule. Naopak vysoké hodnoty q_i vzhledem k R znamenají, že je možné spotřebovat více energie na snížení stavové veličiny směrem na nulu. Pro nalezení vektoru \mathbf{K} , které minimalizuje váhovou funkci J zahrnuje výpočet Ricattiho rovnice. [14]

V prostředí MATLAB je k výpočtu \mathbf{K} možné využít příkazu `lqr` s parametry \mathbf{A} , \mathbf{B} , \mathbf{Q} , R . Matice R byla zvolena jako $R = 1$ a následně bylo nutné správně zvolit váhovou matici \mathbf{Q} , k tomu bylo využito výše uvedených pravidel pro velikost jednotlivých pólů. V návaznosti na tato pravidla byly vhodně zvoleny jednotlivé prvky q_i . Váhové matice R a \mathbf{Q} měli tedy hodnoty

$$R = [1], \quad \mathbf{Q} = \begin{bmatrix} 0,1 & 0 & 0 & 0 \\ 0 & 0,08 & 0 & 0 \\ 0 & 0 & 3 \cdot 10^{-3} & 0 \\ 0 & 0 & 0 & 3 \cdot 10^{-3} \end{bmatrix}.$$

Výsledkem příkazu `lqr` je vektor zesílení zpětné vazby \mathbf{K} , které má hodnoty

$$\mathbf{K} = [11,318 \quad 2,630 \quad -0,055 \quad -0,133].$$

Vlastní čísla nové matice zpětných vazeb $\mathbf{A}' = \mathbf{A} - \mathbf{B}\mathbf{K}$ jsou uvedena v tabulce 4.2

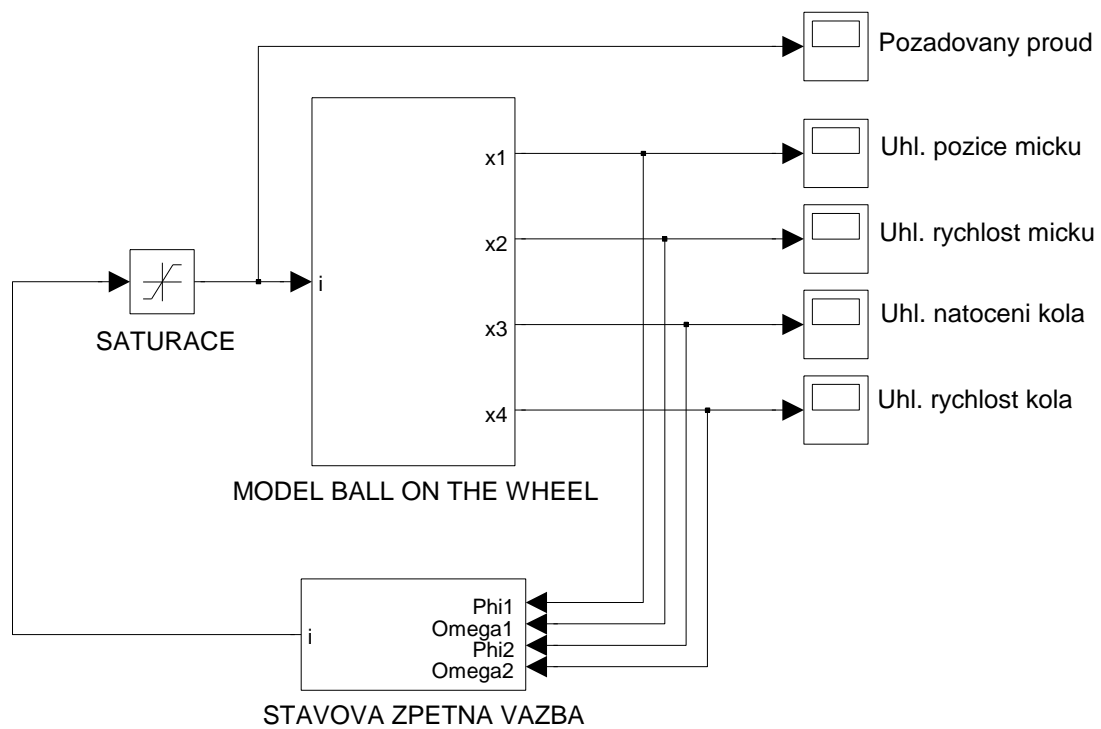
Tab. 4.2: Vlastní čísla nové matice \mathbf{A}'

Pól	Hodnota
p_1	$-5,011$
p_2	$-3,761$
p_3	$-0,688 + 0,478j$
p_4	$-0,688 - 0,478j$

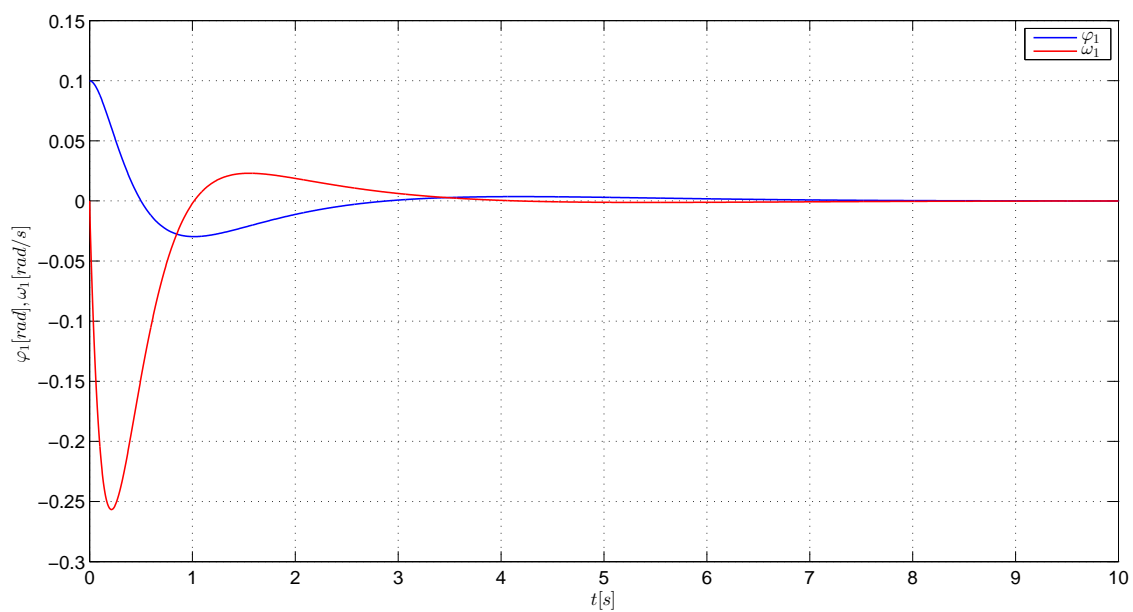
Na obrázku 4.4 je znázorněno schéma řízení systému *Ball on the wheel* vytvořené v prostředí MATLAB Simulink. Blok *MODEL BALL ON THE WHEEL* obsahuje model systému, který je znázorněn na obrázku 4.2. Výstupem subsystému *STAVOVA ZPETNA VAZBA* je požadovaná momentotvorná složka proudu do motoru, která je vypočítána podle následující rovnice

$$\dot{i}_q = -K_1 \cdot x_1 - K_2 \cdot x_2 - K_3 \cdot x_3 - K_4 \cdot x_4. \quad (4.14)$$

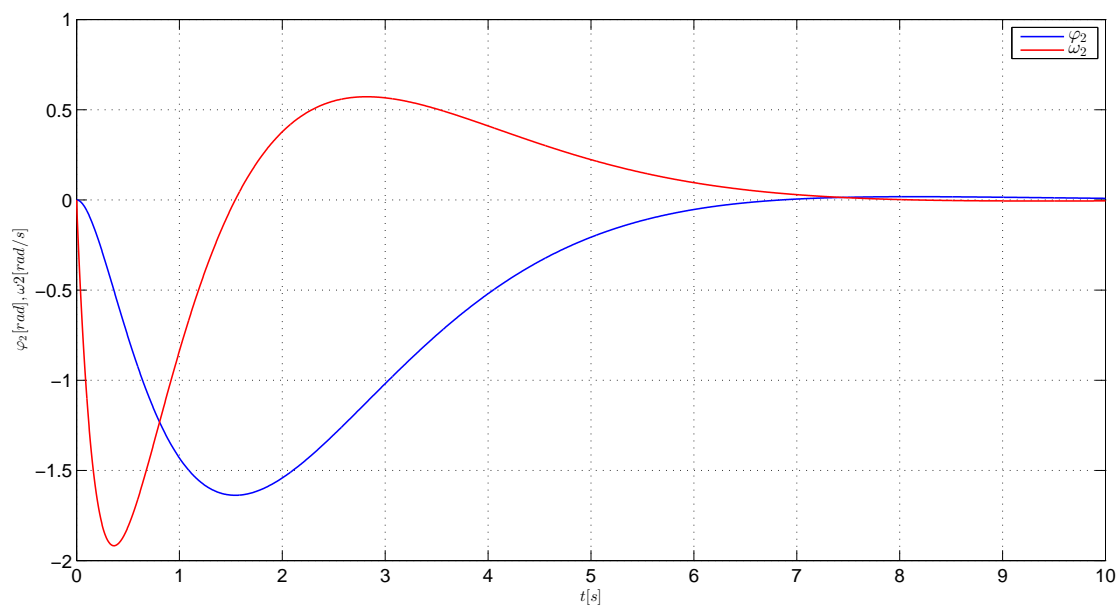
Dále je zde zařazen blok saturace, který omezuje maximální hodnotu proudu do motoru. Na obrázku 4.5 se nachází průběhy úhlové pozice míčku a úhlové rychlosti míčku při počáteční úhlové pozici míčku $0,1 \text{ rad}$. Dále jsou na obrázcích 4.6 a 4.7 uvedeny průběhy dalších dvou stavů systému a průběh momentotvorné složky proudu motorem.



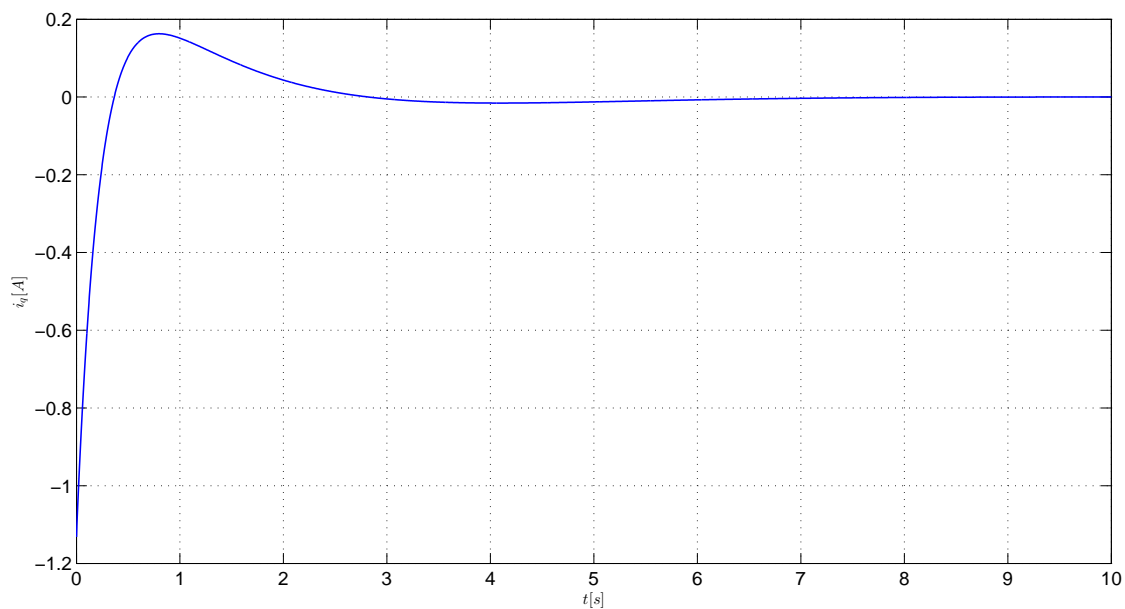
Obr. 4.4: Schéma řízení systému Ball on the wheel



Obr. 4.5: Průběh úhlové pozice a úhlové rychlosti míčku



Obr. 4.6: Průběh úhlového natočení a úhlové rychlosti kola

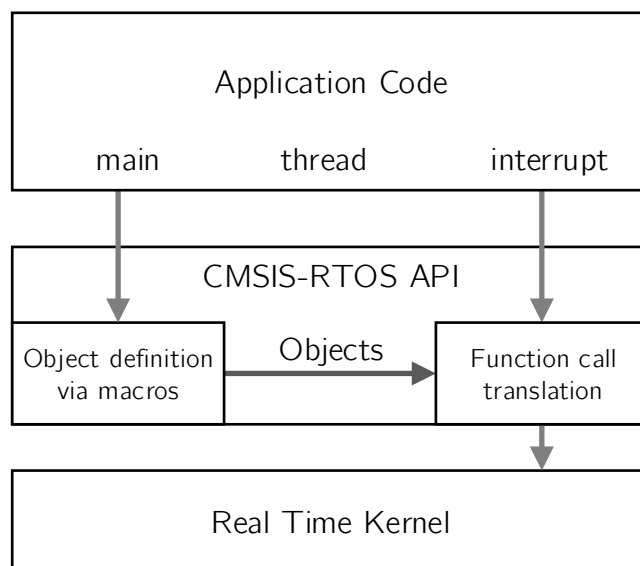


Obr. 4.7: Průběh momentotvorné složky proudu do motoru

5 PROGRAMOVÉ VYBAVENÍ

5.1 Operační systém mbed-rtos

Jako operační systém reálného času byl implementován operační systém *mbed-rtos*, který byl vyvinut komunitou mbed a je založen na implementaci RTX standardu CMSIS-RTOS API.



Obr. 5.1: Struktura programu s CMSIS-RTOS API [17]

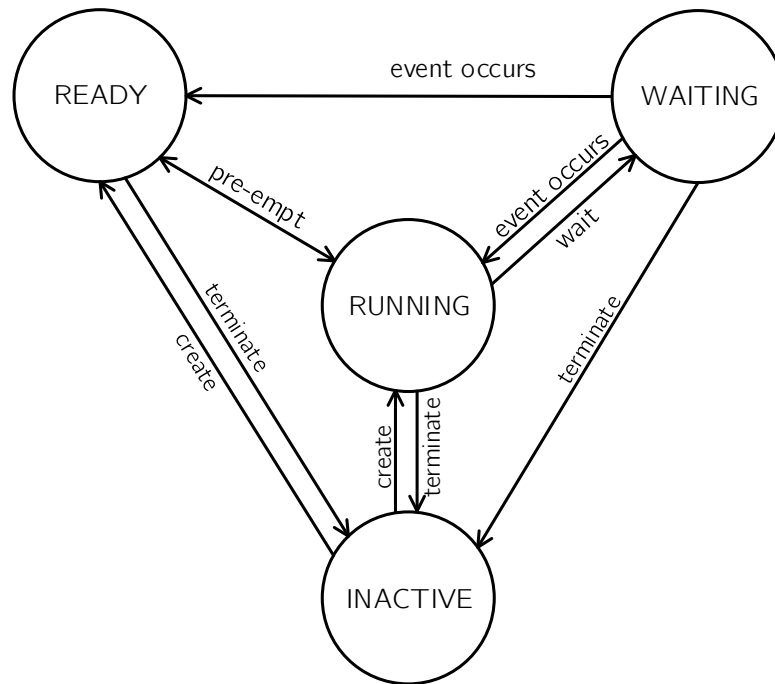
mbed-rtos je napsán v jazyce C++ a obsahuje třídy *Thread*, *Mutex*, *Semaphore*, *Queue*, *MemoryPool*, *Mail*, *RtosTimer*. [15] V implementaci byly využity třídy *Thread* a *Mutex*, jejichž popis je uveden níže v textu. Ostatní třídy využity nebyly, protože jejich implementace by byla zbytečná.

5.1.1 Třída Thread

Třída *Thread* umožňuje definovat, vytvářet a řídit funkce vláken. Vytvořené vlákno se může nacházet ve čtyřech stavech a přechody mezi těmito stavy jsou znázorněny na obrázku 5.2 a jednotlivé stavy jsou popsány níže.

Jednotlivé stavy vláken

- **RUNNING** - V tomto stavu je vlákno, které je právě vykonáváno. V jeden konkrétní okamžik může být ve stavu RUNNING právě jedno vlákno.



Obr. 5.2: Schématické zobrazení stavů vláken [15]

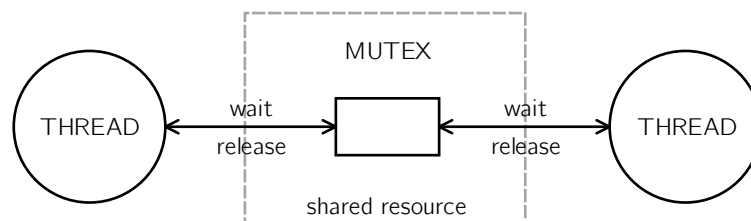
- **READY** - Vlákná, která jsou připravena k vykonání jsou ve stavu READY. Jakmile je vlákno, které je aktuálně ve stavu RUNNING, ukončeno nebo přejde do stavu WAITING, tak vlákno s nejvyšší prioritou ve stavu READY přejde do stavu RUNNING.
- **WAITING** - Ve stavu WAITING jsou vlákna, která čekají až nastane nějaká událost.
- **INACTIVE** - V tomto stavu se nacházejí vlákna, která nejsou vytvořena nebo byla ukončena.

Pro vytvoření nové instance třídy *Thread* se nastavují následující parametry:

- **Exekuční kód** - funkce, která bude vykonávána tímto vláknem.
- **Argument** - ukazatel, který je předán do funkce vlákna jako počáteční argument.
- **Priorita** - počáteční priorita vlákna. Podle priority vlákna je rozhodováno, které vlákno může přejít do stavu RUNNING. Existuje celkem 7 priorit, které jsou uloženy jako výčtový typ od nejnižší priority `osPriorityIdle` až po nejvyšší `osPriorityRealtime`.
- **Velikost zásobníku** - velikost zásobníku v bytech potřebnou pro danou funkci vlákna.
- **Ukazatel na zásobník** - ukazatel na zásobník pro použití daného vlákna.

5.1.2 Třída Mutex

Mutex je využíván k synchronizaci provádění vláken a je obecně používán v mnoha operačních systémech k řízení zdrojů. Mutexy slouží k zabezpečení přístupu ke sdíleným prostředkům. Mutex je nejdříve vytvořen a poté je předáván mezi vlákny, kde tato vlákna mohou mutex získat a poté mutex uvolnit. Třída *Mutex* obsahuje hlavní

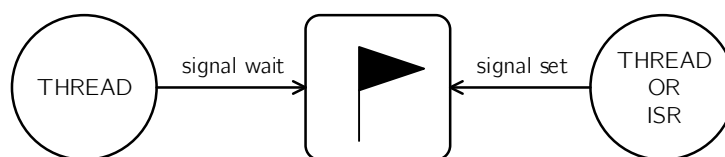


Obr. 5.3: Zobrazení funkce mutexu [15]

dvě metody, jedná se o metodu `lock`, kdy vlákno čeká až bude mutex k dispozici. Poté může vlákno manipulovat se sdílenými prostředky. Druhou metodou je metoda `unlock`, kdy vlákno mutex uvolní.

5.1.3 Signal Management

Jednoduchým prvkem sloužícím ke komunikaci mezi vlákny jsou signály. Každé vlákno může čekat na jeden či více signálů použitím metody `signal_wait` a také vygenerovat signál nebo více signálů pomocí metody `signal_set`.



Obr. 5.4: Signal Management

5.2 mbed SDK

mbed SDK je C/C++ softwarová platforma používaná více než desetitisíci vývojáři pro snazší vývoj jejich projektů a byla také používána pro tvorbu řídicího programu v rámci této diplomové práce. SDK se nachází pod licencí Apache 2.0, takže je možné jej používat jak pro komerční, tak i soukromé projekty. mbed SDK se používá s online

kompilátorem, který je kompatibilní s mnoha dalšími nástroji mikrokontrolérů ARM. Programování zde probíhá na vyšší úrovni, kdy jsou používány abstraktní objekty a volání vytvořených API. Při požadavku na nižší úroveň programování je možné přejít k CMSIS APIs. [16]

5.3 Implementace programového vybavení

Pro tvorbu řídicího programu systému Ball on the wheel bylo využito online vývojového prostředí komunity mbed. Program je napsán v jazyce C++ a jsou využity zdrojové kódy pro obsluhu jednotlivých periférií, které byly vytvořeny vývojáři komunity mbed. Řídicí program mikrokontroléru byl rozdělen do několika souborů viz obrázek 5.5. Rozdělení do více souborů má tedy výhodu v přehlednosti, ale také jednotlivé části mohou být překládány odděleně. V adresáři *mbed* se nacházejí knihovní funkce vytvořené komunitou mbed, které slouží k obsluze periférií. Dále adresář *mbed-rtos* obsahuje zdrojové kódy operačního systému reálného času, opět od komunity mbed. A poslední adresář *FastAnalogIn* obsahuje knihovní funkce pro AD konverzi, která je využita pro fotoelektrický snímač vzdálenosti. Ostatní soubory již byly vytvořeny v rámci diplomové práce. Pomocí operačního systému reálného času mbed-rtos bylo možné rozdělit program na jednotlivá vlákna, která budou popsána v podkapitole 5.3.2. K obsluze periférií bylo vytvořeno pět tříd, jejichž metody jsou volány z příslušných vláken. Níže je uveden základní popis jednotlivých tříd a vytvořených vláken řídicího programu, který je obsahem přílohy na CD.



Obr. 5.5: Struktura zdrojového kódu

5.3.1 Vytvořené třídy

V této podkapitole jsou popsány jednotlivé vytvořené třídy v rámci implementace programového vybavení. Tyto třídy se starají o obsluhu jednotlivých periférií připojených k mikrokontroléru. Class diagramy těchto vytvořených tříd jsou uvedeny v příloze C.

Třída *cUltrasonic*

Třída *cUltrasonic* slouží k obsluze ultrazvukového senzoru. Třída obsahuje metodu *setTrig*, která nastaví pin *Trig* ultrazvukového senzoru do logické 1 na dobu $10\ \mu\text{s}$. Metoda *setTrig* je volána z vlákna pro snímače polohy každých $15\ \text{ms}$. Dalšími metodami třídy jsou obslužné funkce, které jsou volány při nástupné či sestupné hraně příchozího pulzu od ultrazvukového snímače. Obslužná funkce *riseEdge* je volána při nástupné hraně příchozího pulzu a obsahuje spuštění časovače. Poté při sestupné hraně v obslužné funkci *fallEdge* je časovač zastaven a uložena jeho hodnota. Uložená hodnota tak odpovídá šířce příchozího pulzu od snímače. Dále je nastaven signál vláknu pro sběr dat, které si aktuální šířku pulzu vyčte a přepočítá na úhlovou pozici míčku.

Třída *cLedSensor*

Třídou sloužící pro obsluhu fotoelektrického senzoru je třída *cLedSensor*, jehož hlavní metodou je metoda *read*. Metoda *read* je volána vlákem pro snímače polohy každé $3\ \text{ms}$ a vyčítá hodnotu z AD převodníku, kterou zapisuje do pomocného pole. Výsledná vyčtená hodnota je poté získána jako aritmetický průměr z pěti dílčích hodnot uložených v pomocném poli. Když je výsledná analogová hodnota k dispozici, nastaví se signál vláknu pro sběr dat, které opět tuto hodnotu přepočítá na úhlovou pozici míčku. K vyčítání analogové hodnoty z AD převodníku byla využita třída *FastAnalogIn*, která byla vytvořena vývojáři v komunitě mbed. Její hlavní výhodou je provádění AD konverze na pozadí bez nutnosti zásahů jádra mikrokontroléru.

Třída *cCan*

Jednou z nejdůležitějších vytvořených tříd je třída *cCan*, která obsahuje metody určené pro komunikaci na sběrnici CAN. Tato třída zapouzdřuje třídu *CAN* vytvořenou komunitou mbed a využívá třídu *CANMessage*, která umožňuje vytvářet formát zprávy vysílané po sběrnici CAN. Třída *cCan* umožňuje například pomocí SDO požadavku na zápis zapínat či vypínat výkonový most nebo nulovat resolver. Další funkcí této třídy je vysílat servozesilovači PDO s žádanou hodnotou proudu. Jednou z metod je také obslužná funkce sběrnice CAN, která je volána při příjmu zprávy od servozesilovače. Jestliže například mikrokontrolér přijal PDO od servozesilovače, je zapsána aktuální hodnota resolveru a nastaven signál vláknu pro sběr dat. Toto vlákno si pak může vyčíst aktuální polohu resolveru a zapsat do sdílené proměnné. Stejný princip funguje i s příjmem objektu SDO, který byl přijat na základě požadavku na vyčtení rychlosti a obsahuje aktuální rychlost motoru.

Třída cControl

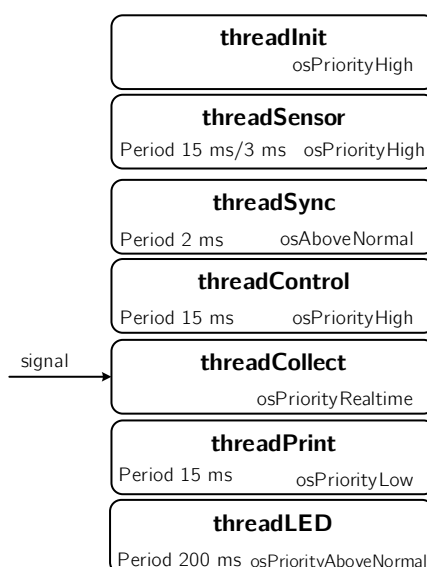
Jednoduchou třídou sloužící k výpočtu požadovaného proudu do motoru je třída *cControl*. Tato třída obsahuje metodu *computeCurrent*, která vypočítá z aktuálních stavů systému požadovanou momentotvornou složku proudu. Tu poté vlákno sloužící pro řízení systému vyšle pomocí PDO servozesilovači, který ji předá do regulátoru proudu.

Třída cPrint

Poslední vytvořenou třídou je třída *cPrint*. Třída obsahuje metody, které slouží pro posílání dat po sériové lince do PC. Tyto metody byly vytvořeny na základě testování řídicího programu. Komunikace probíhá rychlostí 57 600 baud, 8-mi datovými bity, žádnou paritou a jedním stop bitem. Další metodou je metoda *printStates*, která slouží pro zápis stavů systému do lokálního souborového systému. Než dojde pomocí této metody k zápisu do souborového systému, musí být pomocí mutexu zabezpečen přístup ke sdílené proměnné obsahující stavy systému.

5.3.2 Vytvořená vlákna

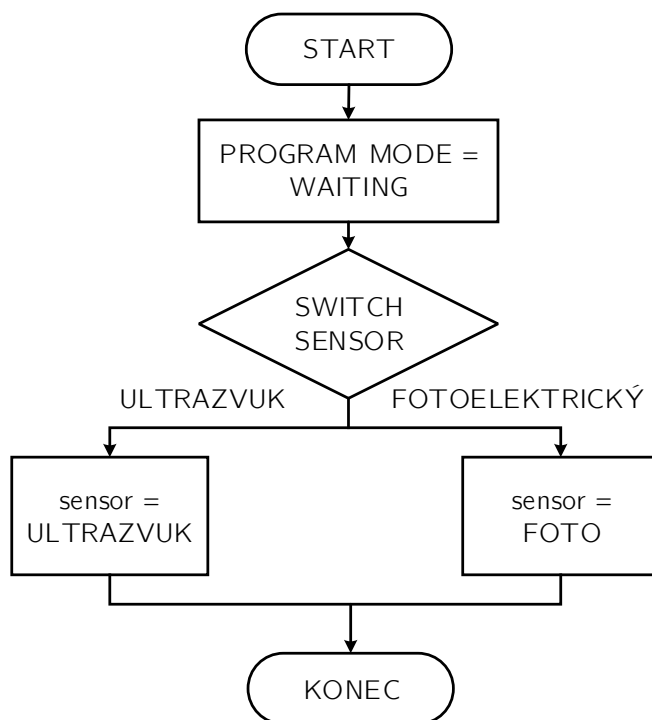
V této podkapitole jsou popsána jednotlivá vlákna sloužící ke správnému běhu programu. Tato vlákna využívají metody tříd, které jsou popsány výše. Jednotlivá vlákna a jejich spouštění jsou zobrazeny na obrázku 5.6.



Obr. 5.6: Jednotlivá vlákna programu

Vlákno pro inicializaci

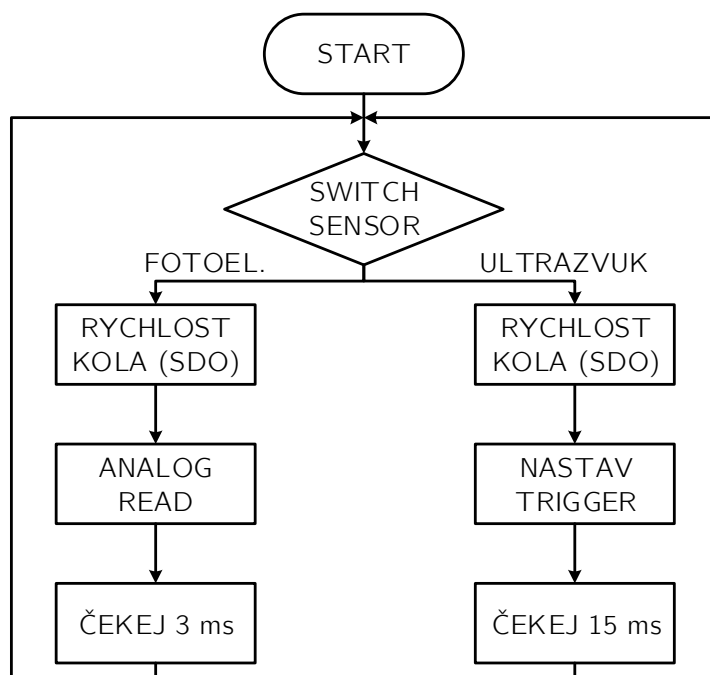
Prvním vytvořeným vláknem je inicializační vlákno, které nastaví mód programu do režimu WAITING. Dále je zjištěno, jaký typ senzoru bude v průběhu řízení využíván, to je v případě fotoelektrického snímače vzdálenosti docíleno přivedením logické 1 na pin *p7* mikrokontroléru, v opačném případě je využíván ultrazvukový snímač vzdálenosti. Po této inicializaci je vlákno ukončeno. Na obrázku 5.7 se nachází vývojový diagram inicializačního vlákna.



Obr. 5.7: Vývojový diagram pro inicializační vlákno

Vlákno pro snímače vzdálenosti

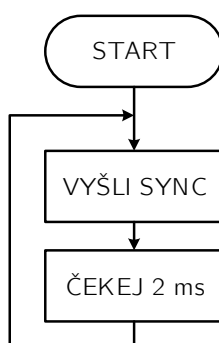
Vlákno pro snímače vzdálenosti je rozděleno podle použitého snímače. Pokud byl zvolen ultrazvukový snímač, tak toto vlákno volá metodu *setTrig* třídy *cUltrasonic*, která vyšle krátký pulz ultrazvukovému snímači a následně vlákno přejde na 15 ms do stavu WAITING. Při použití fotoelektrického snímače vzdálenosti je princip podobný, s tím rozdílem, že místo vyslání krátkého pulzu se vyčítá hodnota napětí přivedená na analogový vstup mikrokontroléru. To je prováděno každé 3 ms. Výsledná hodnota vyčtená z analogového vstupu je vždy po pěti vzorcích průměrována, k dispozici je tedy aktuální pozice míčku vždy po 15 ms. V tomto vlákně je také vyslán SDO požadavek servozesilovači na vyčtení aktuální rychlosti kola. Vývojový diagram vlákna pro snímače vzdálenosti je zobrazen na obrázku 5.8.



Obr. 5.8: Vývojový diagram vlákna pro snímače vzdálenosti

Vlákno pro vysílání synchronizace

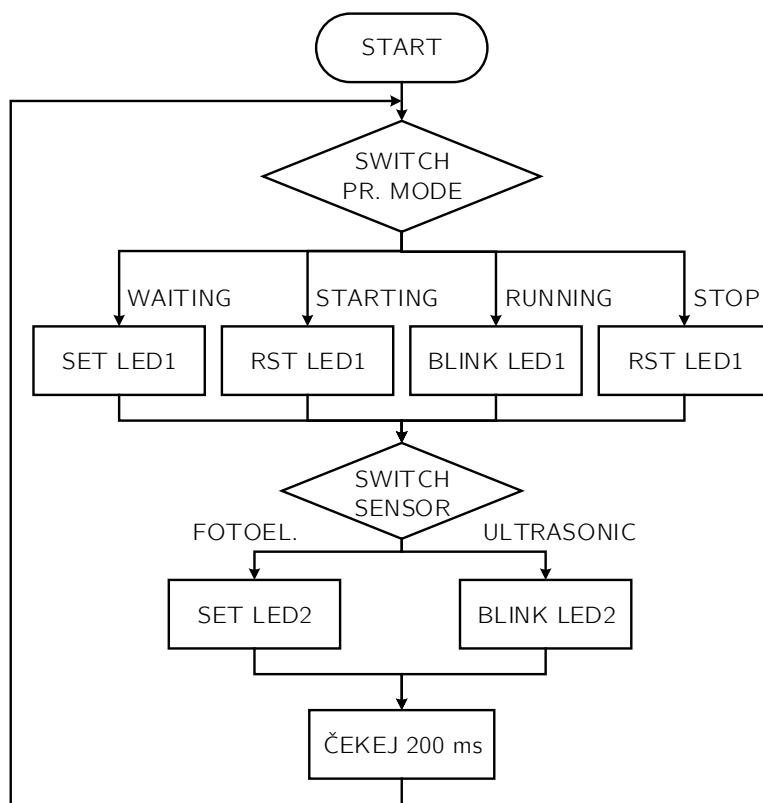
Dalším vytvořeným vlákem je vlákno pro vysílání synchronizace. Slouží k posílání objektu SYNC po sběrnici CAN a tedy k synchronizaci mezi mikrokontrolérem mbed a servozsilovačem. Toto vlákno je opět periodické s periodou opakování 2 ms a vývojový diagram vlákna se nachází na obrázku 5.9.



Obr. 5.9: Vývojový diagram vlákna pro vysílání synchronizace

Vlákno pro zobrazení stavu programu

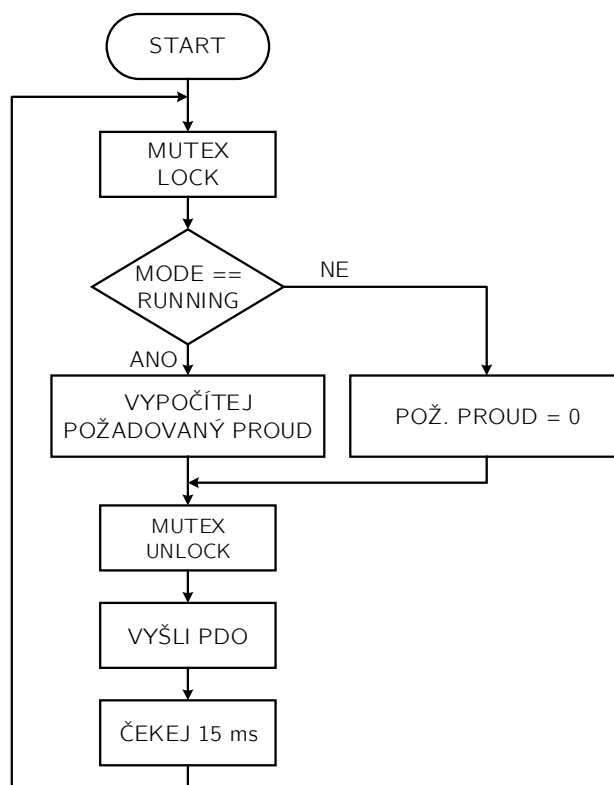
Vlákno pro zobrazení stavu programu slouží k zobrazení stavu programu pomocí LED diod umístěných na mikrokontroléru. Nachází se opět v nekonečné smyčce a má nejnižší prioritu. K zobrazení stavu programu slouží dvě LED diody, kde jedna slouží přímo pro zobrazování stavu programu a druhá pro zobrazení použitého snímače vzdálenosti. Funkce tohoto vlákna je zřejmá z obrázku 5.10.



Obr. 5.10: Vývojový diagram vlákna pro zobrazení stavu programu

Vlákno pro řízení

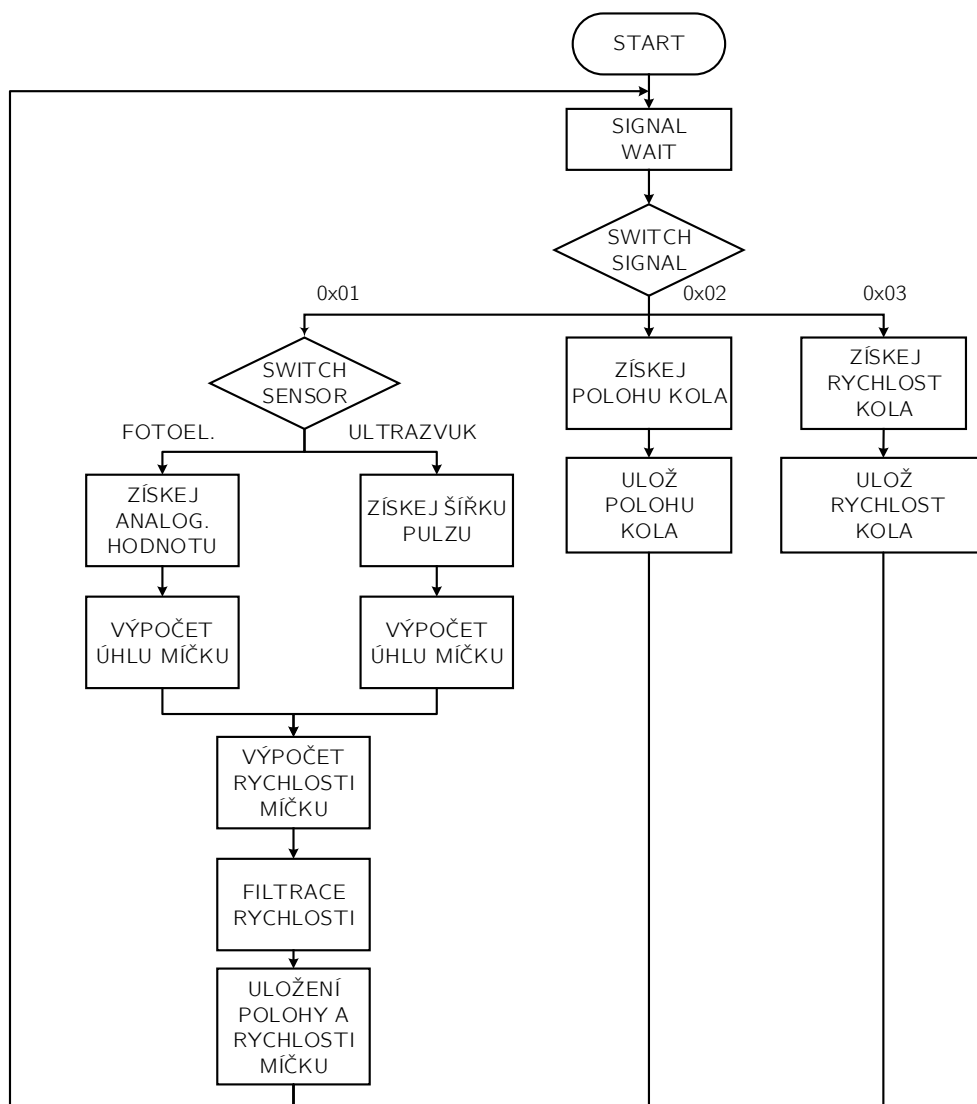
Toto vlákno každých 15 ms vypočítává požadovanou momentotvornou složku proudu na základě hodnot stavů systému. To je prováděno voláním metody *computeCurrent* třídy *cControl*. Jestliže se program nenachází ve stavu *RUNNING*, je momentotvorná složka nulová. Poté je požadovaná hodnota proudu poslána jako procesní datový objekt po sběrnici CAN servozesilovači, kde se tato hodnota po příjmu objektu *SYNC* zapíše do regulátoru proudu servozesilovače.



Obr. 5.11: Vývojový diagram vlákna pro řízení

Vlákno pro sběr dat

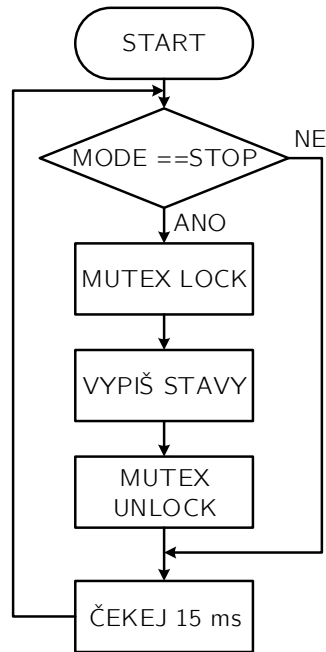
Jedním z nejdůležitějších vláken je vlákno pro sběr dat. Toto vlákno se stará o výpočet úhlové pozice míčku a také pro výpočet a filtraci úhlové rychlosti míčku. Dále získává od servozesilovače úhlové natočení kola a úhlovou rychlost kola. Tyto vypočítané a získané stavy systému ukládá do sdílené proměnné, která je dále k dispozici ostatním vláknům. Vlákno pro sběr dat čeká až bude nastaven příslušný signál a na základě toho poté provede příslušné operace, které jsou zřejmé z obrázku 5.12.



Obr. 5.12: Vývojový diagram vlákna pro sběr dat

Vlákno pro výpis dat

Vlákno pro výpis dat zapisuje stavy systému do lokálního souborového systému každých 15 *ms*. Jelikož stavy systému se nachází v globální proměnné, je k přístupu použit mutex. Ten je také vždy použit i v ostatních vláknech, které přistupují k této sdílené proměnné. Na obrázku 5.13 se nachází vývojový diagram pro výpis dat.



Obr. 5.13: Vývojový diagram pro výpis dat

5.3.3 Kritická oblast

Pro sdílený prostředek, ke kterým má přístup více vláken, byla použita struktura, ve které jsou uloženy stavy systému ($\varphi_1, \omega_1, \varphi_2, \omega_2$). K tomu, aby se zabezpečil přístup k této sdílené proměnné, byl využit mutex, jehož popis je uveden v podkapitole 5.1.2. Když tedy vlákno potřebuje zapsat nebo vyčíst nějaký stav systému, tak čeká až bude mutex uvolněn, následně mutex zamkne a může pracovat se sdílenou proměnnou. Poté je mutex opět uvolněn.

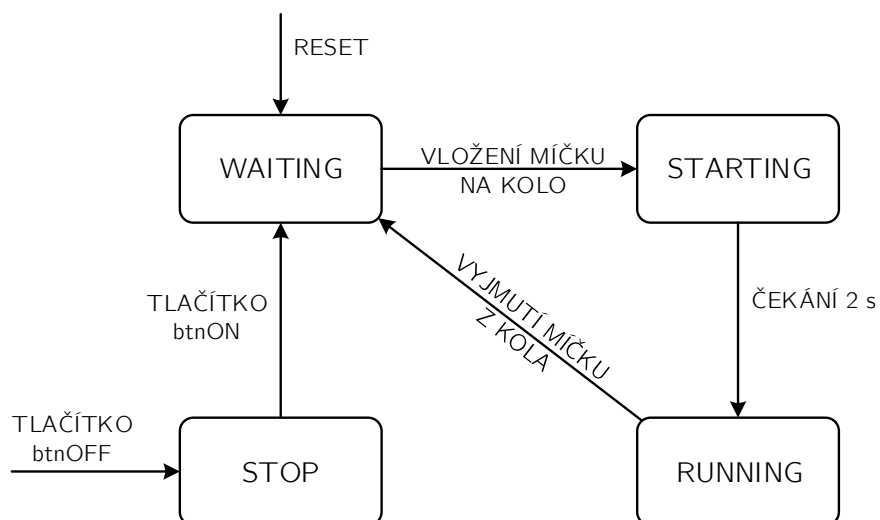
5.3.4 Stavy řídicího programu

Řídicí program se může nacházet celkem ve čtyřech stavech, kterými jsou stavy:

- **WAITING** - stav programu, kdy je čekáno na vložení míčku na kolo a v tomto stavu je zapnuto servo. Do stavu WAITING se přejde tlačítkem RESET, tlačítkem ON nebo vyjmutím míčku z kola.
- **STARTING** - je stav, kdy je míček umístěn na kolo a čeká se 2 s, než program přejde do stavu RUNNING. To je z důvodu, aby se předešlo špatným informacím ze snímače vzdálenosti při umísťování míčku na kolo.
- **RUNNING** - v tomto stavu je míček umístěn na kole a na základě měření stavů systémů je posílán požadovaný proud do motoru. Do tohoto stavu se přejde pouze ze stavu STARTING.

- **STOP** - do tohoto stavu se přejde tlačítkem OFF, kdy je vypnuto servo. Ze stavu STOP do stavu WAITING se přechází stiskem tlačítka ON.

Na obrázku 5.14 je uveden stavový diagram řídicího programu, ze kterého jsou patrné přechody z jednotlivých stavů popsaných výše.



Obr. 5.14: Stavový diagram řídicího programu

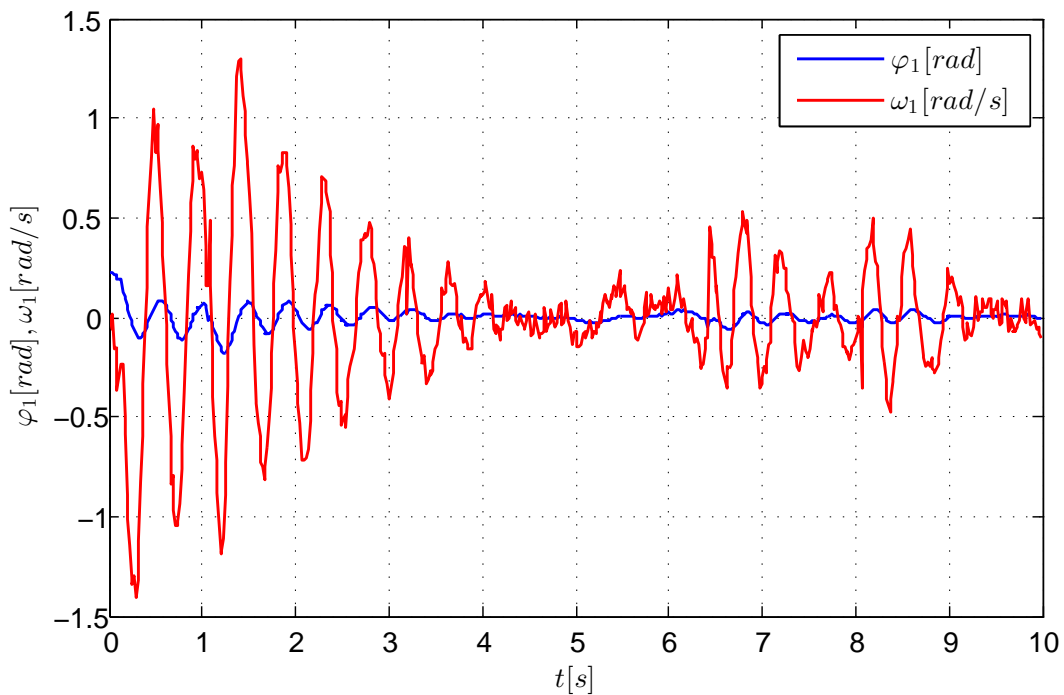
6 VÝSLEDKY A POSOUZENÍ VYTVOŘENÉHO ŘEŠENÍ

Po implementaci programového vybavení pro ovládání servomotoru, snímačů vzdálenosti a řízení systému Ball on the wheel se přešlo k testování a ladění regulátoru. První získané výsledky řízení pomocí stavové zpětné vazby popsané v podkapitole 4.5 byly neuspokojivé z důvodu dlouhé doby ustálení úhlové pozice míčku na nulové hodnotě. Kolo se tak roztáčelo na vyšší hodnoty rychlosti a docházelo k padání míčku z kola. To bylo vyřešeno snížením zesílení zpětné vazby od úhlové pozice míčku a zvýšením zesílení zpětné vazby od úhlové rychlosti kola a úhlového natočení kola. Zesílení zpětné vazby má po doladění následující hodnoty

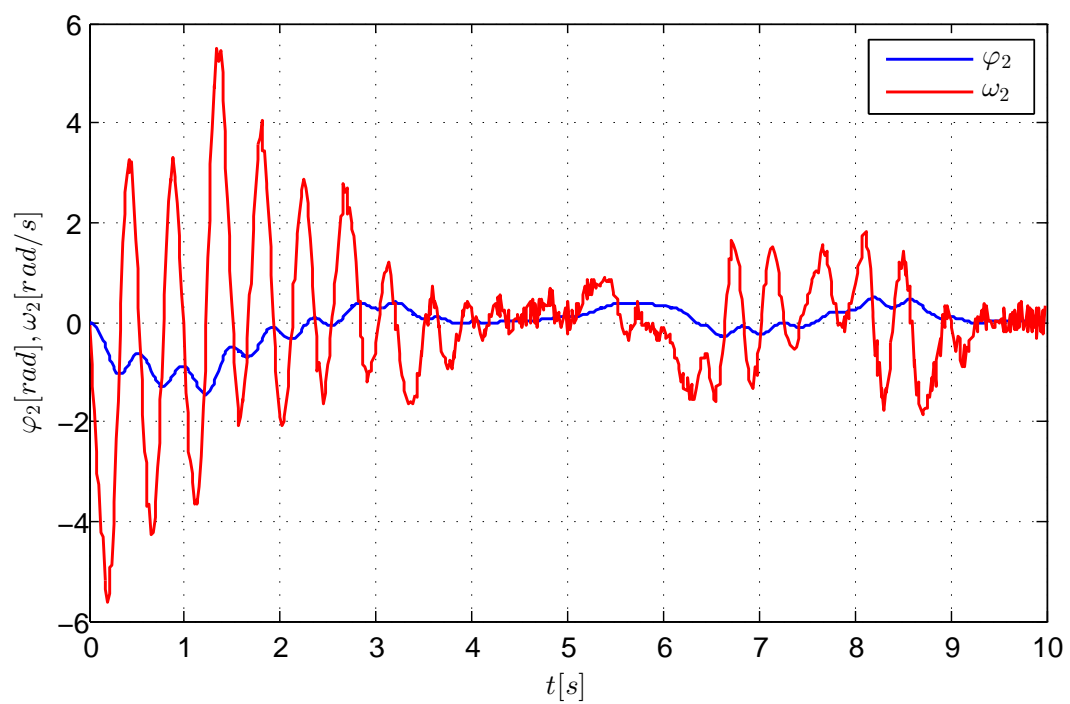
$$\mathbf{K} = [9,6 \quad 2,3 \quad -0,12 \quad -0,2].$$

6.1 Naměřené průběhy stavů systému při řízení

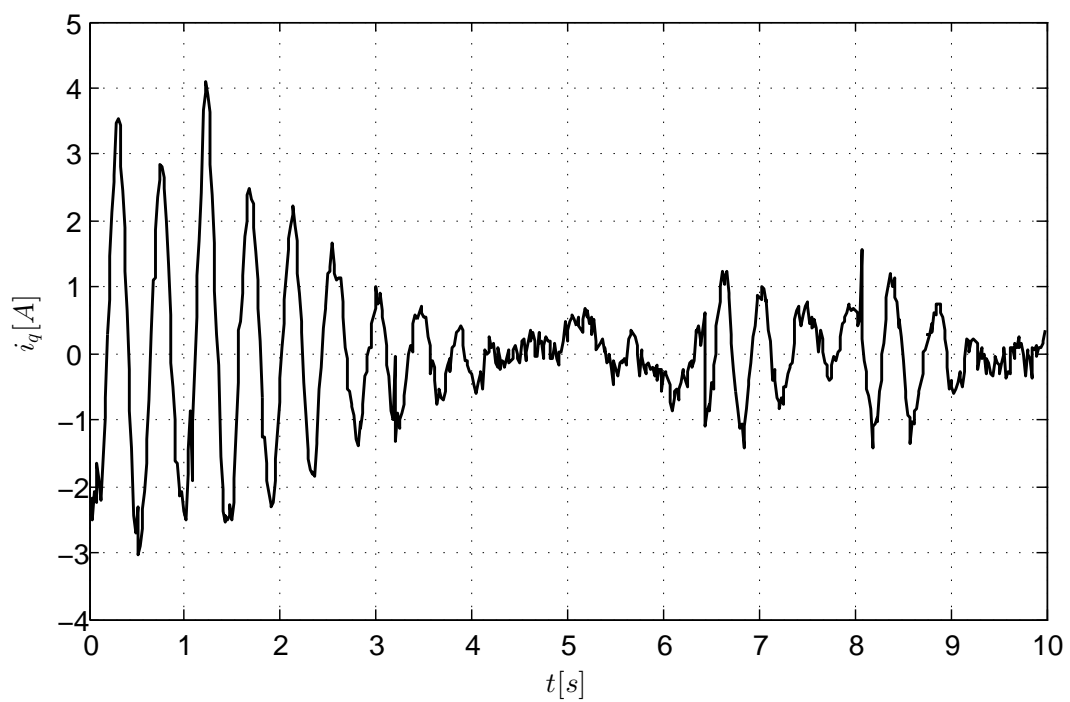
Na obrázku 6.1 se nachází průběhy prvních dvou stavů systému, tedy úhlové pozice míčku a úhlové rychlosti míčku. Na obrázku 6.2 se nachází průběhy úhlového natočení kola a úhlové rychlosti kola. Obrázek 6.3 znázorňuje požadovaný proud do motoru.



Obr. 6.1: Průběhy úhlové pozice a úhlové rychlosti míčku

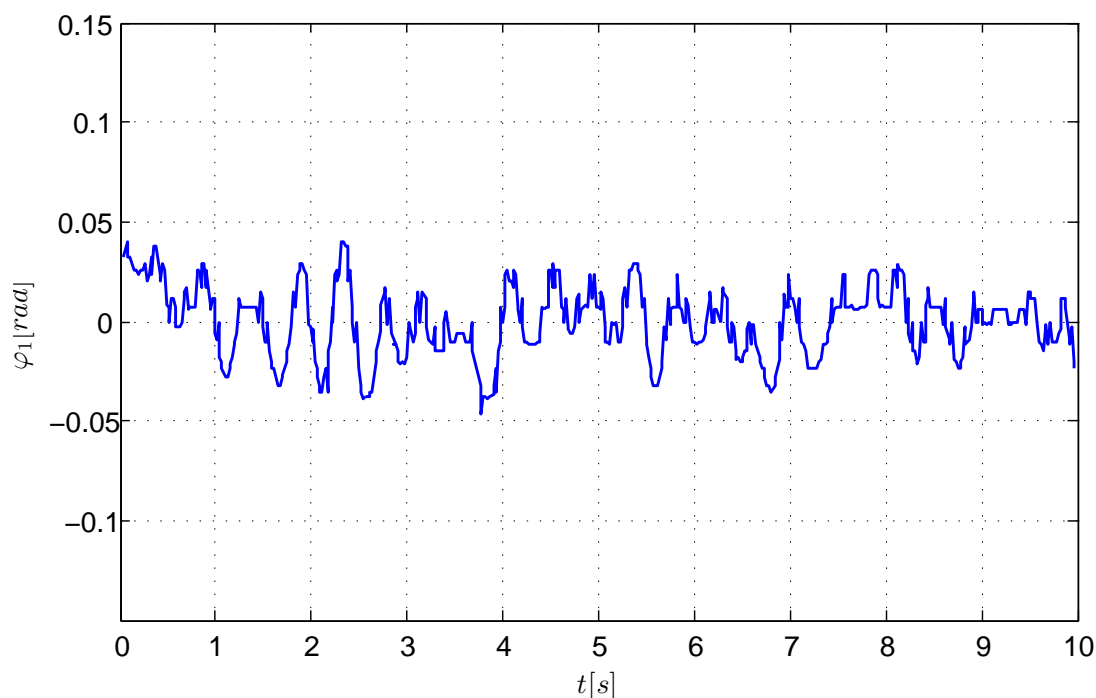


Obr. 6.2: Průběhy úhlového natočení a úhlové rychlosti kola



Obr. 6.3: Průběh požadované momentotvorné složky proudu

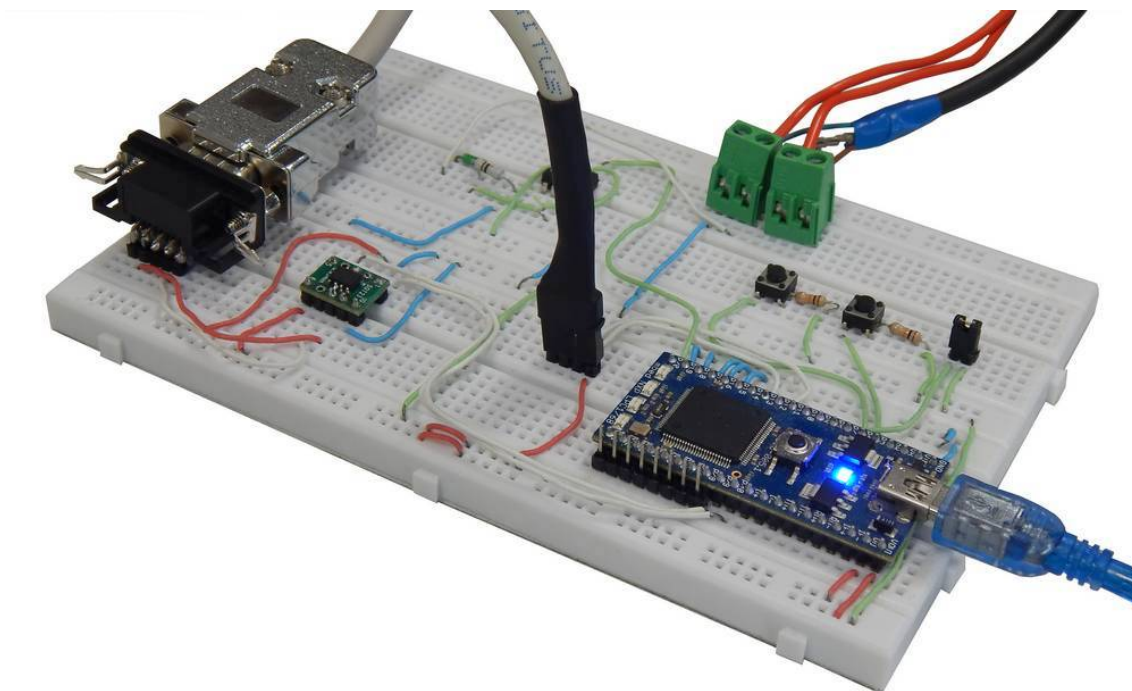
Tyto výše uvedené průběhy byly změřeny při použití fotoelektrického snímače vzdálenosti. Byla tedy ověřena funkčnost řízení systému pomocí stavové zpětné vazby a tedy stabilizace míčku v horní pozici kola. Stabilizace míčku byla úspěšná i s použitím ultrazvukového snímače, nicméně výsledné průběhy jsou úměrné kvalitě snímače. Průběh měřené úhlové pozice míčku pomocí ultrazvukového snímače je znázorněn na obrázku 6.4.



Obr. 6.4: Průběh úhlové pozice míčku měřené ultrazvukovým snímačem

6.2 Posouzení řešení laboratorního modelu

Laboratorní model systému Ball on the wheel byl vytvořen na základě zadání této diplomové práce. Výhodou tohoto modelu je jeho jednoduchost při sestavování, protože hliníkové profily umožňují snadnou montáž a případné úpravy. Nevýhodou je však jeho malá robustnost, protože při vyšších akčních zásazích dochází ke kmitání konstrukce modelu, zejména hliníkového profilu nesoucího snímače vzdálenosti. V rámci této diplomové práce byl také vytvořen návrh desky plošných spojů (DPS), která byla vytvořena na základě zapojení uvedené v příloze B.2 a otestování proběhlo na nepájivém kontaktním poli viz obrázek 6.5.



Obr. 6.5: Nepájivé kontaktní pole

6.2.1 Servomotor

Při sestavování laboratorního modelu se vyskytovaly obavy, zda bude točivý moment servomotoru dostatečný. Proto byl brán také zřetel na průměr použitého kola, aby se tak zvýšil převodový poměr a tím se snížil moment zatížení na hřídel motoru. Z naměřených průběhů 6.2 a 6.3 je patrné, že se tyto naměřené veličiny nenachází v maximálních hodnotách a nedochází tak k saturaci. Proto točivý moment použitého servomotoru je pro tuto aplikaci dostatečný.

6.2.2 Výběr snímačů vzdálenosti

Při sestavování modelu byly vybrány dva druhy snímačů. Prvním z nich byl ultrazvukový snímač vzdálenosti US-100 a druhým byl fotoelektrický snímač vzdálenosti SHARP GP2Y0A21YK, u něhož byla nevýhodou poměrně vyšší doba odezvy 40 ms a také byl citlivý na drift. U naměřené převodní charakteristiky snímače tak docházelo postupem času k jejímu posunutí. Pro měření úhlové pozice míčku byl tedy využit pouze ultrazvukový snímač vzdálenosti US-100 a dále také fotoelektrický snímač vzdálenosti FADK 14. U ultrazvukového snímače je důležitá jeho pozice umístění při snímání polohy míčku, protože tento senzor má snímací úhel okolo 15° . Z toho důvodu musí být umístěn výše než fotoelektrický snímač vzdálenosti, aby tak nedocházelo k odrazům ultrazvukových vln od kola, a tedy k chybnému vyhodno-

cení vzdálenosti míčku. Na oba dva snímače bylo vytvořeno nerezové pouzdro, které umožňuje snadnou montáž a posunutí podle právě využívaného snímače vzdálenosti.

6.2.3 Výběr kola

Při vytváření laboratorního modelu byl brán zřetel na točivý moment motoru, proto bylo vybráno kolo s poměrně větším průměrem, aby byl tak převodový poměr vyšší. Celkem byly otestovány 3 kola s průměry 32 *cm*, 42 *cm* a 55 *cm*. Nakonec bylo vybráno kolo s průměrem 55 *cm*, protože při zbývajících dvou kolech byl točivý moment motoru nedostačující. Nevýhodou použitého kola je malá šířka ráfku, která působí jako malá zábrana pro míček. Při použití míčku s větším průměrem tak dochází k padání míčku do stran.

6.2.4 Výběr míčku

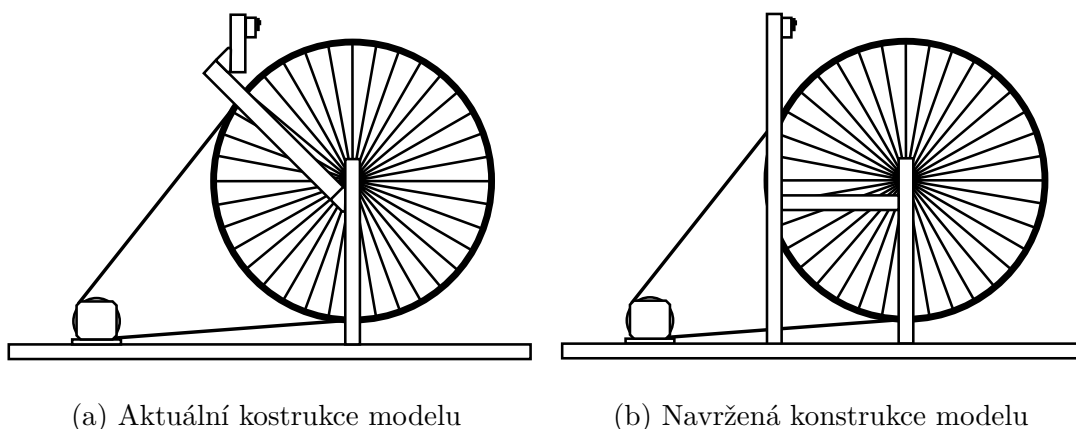
Dalším důležitým bodem byl správný výběr míčku. Ten musel mít přílnavý povrch, aby nedocházelo k prokluzu mezi míčkem a kolem. Dále bylo testováno více míčků pro oba dva snímače. Při použití fotoelektrického snímače vzdálenosti musel být povrch míčku matný, protože u lesklého povrchu docházelo k odrazům světelného paprsku a tedy k chybnému vyhodnocení snímače. U ultrazvukového snímače vzdálenosti musel být míček dostatečně velký, aby snímač dokázal tento míček zachytit. Celkem bylo otestováno 6 míčků s průměry od 5 *cm* do 11 *cm*, ze kterých bylo 5 míčků vyřazeno z důvodu jejich velikosti či povrchu míčku. Jako nejvhodnější byl vybrán matný bílý gumový míček s průměrem 8,5 *cm*, který je možné použít s oběma druhy snímačů.

6.3 Návržené úpravy laboratorního modelu

Níže jsou uvedeny navržené úpravy laboratorního modelu, které by měly usnadnit práci s modelem a také vylepšit získané výsledky při řízení.

- **Napájecí zdroj:** V práci byl pro napájení servozesilovače použit spínaný síťový zdroj MEAN WELL DRP-480-24, který obsahuje ochrany proti přepětí a přetížení. Proto při rychlých změnách otáčení kola docházelo k vypínání zdroje. Vhodné by bylo tedy použití tvrdého napájecího zdroje.
- **Konstrukce modelu:** Na obrázku 6.6a se nachází náčrt konstrukce laboratorního modelu, který byl vytvořen v rámci této diplomové práce. Jeho hlavní nevýhodou je kmitání hliníkového profilu, na kterém jsou umístěny snímače vzdálenosti. Na obrázku 6.6b je zobrazen náčrt navržené úpravy laboratorního

modelu, kde hliníkový profil nesoucí snímače vzdálenosti je svislý a příčně spojen s profily, na kterých je umístěno kolo. Toto řešení by mělo zajistit větší robustnost laboratorního modelu.



Obr. 6.6: Konstrukce laboratorního modelu

- **Řídicí mikrokontrolér:** Jako řídicí mikrokontrolér byl použit mbed LPC1768 obsahující mnoho různých periférií jako například Ethernet, USB OTG, SPI či I2C, které však využity nebyly, ale pro budoucí využití by přínosem byly. Tento mikrokontrolér neobsahuje FPU, ale pro tuto aplikaci byl výpočetní výkon mikrokontroléru i s absencí FPU dostačující.
- **Zobrazení stavu programu:** V řídicím programu je možné posílat data do PC pomocí sériové linky. Tato data by dále mohla být zpracována aplikací, pomocí které by se daly zobrazovat stavové proměnné a stavy programu. Dále by bylo možné například pomocí bluetooth modulu, připojeného k mikrokontroléru, posílat data bezdrátově do mobilu či tabletu.
- **Zachycení míčku:** Dalším vylepšením modelu by také mohlo být vytvoření postranních příčných zachycení na míček, na kterých by byly umístěny koncové spínače. Pomocí nich by pak míček mohl sám vyjet do horní pozice kola vlastní setrvačností pomocí otočení a zabrždění kola.
- **Bezpečnostní prvky:** Poslední úpravou by mohlo být rozšíření modelu o bezpečnostní prvky. Ke konstrukci by se tak mohla přidat světelná závora, která by zajistila zabrždění kola při přerušení světelného paprsku. Nouzové zabrždění kola by také mohlo být docíleno přidáním bezpečnostního tlačítka TOTAL STOP.

7 ZÁVĚR

Cílem této diplomové práce bylo implementovat řídicí algoritmus do mikrokontroléru s operačním systémem reálného času. Jako řízený systém byl vytvořen laboratorní model systému Ball on the wheel, u kterého je v horní pozici kola umístěn míček, který je v této nestabilní pozici stabilizován.

Nejdříve byl vytvořen matematický model systému Ball on the wheel. K získání pohybových rovnic modelu bylo využito Lagrangeových rovnic druhého druhu. Po získání stavového popisu nelineárního dynamického systému byla provedena linearizace, kde se provedla lineární náhrada systému rozvojem do Taylorovy řady. Poté již byly získány výsledné stavové rovnice systému Ball on the wheel.

Dále se přešlo k sestavení laboratorního modelu systému, jehož konstrukce byla sestavena z hliníkových profilů. V kapitole 2 je uveden popis vytvořeného laboratorního modelu. Jsou zde popsány jednotlivé prvky, ze kterých se model skládá. Těmi je řídicí mikrokontrolér mbed NXP LPC1768, snímače vzdálenosti pro měření úhlové pozice míčku, dále použitý servomotor sloužící k otáčení kola a digitální servozesilovač sloužící k řízení servomotoru.

K řízení servomotoru bylo využito digitální momentové řízení, u kterého bylo nutné správně nastavit regulátory proudu typu PI. Návrh nastavení těchto regulátorů je uvedeno v kapitole 3. K řízení servomotoru slouží servozesilovač, se kterým může řídicí systém komunikovat více způsoby. V této práci bylo využito komunikace po sběrnici CAN. V kapitole 3 je také popsán komunikační protokol na této sběrnici, který vychází z protokolu CANopen.

Dále bylo navrženo řízení systému Ball on the wheel pomocí stavové zpětné vazby. V kapitole 4 je popsán princip stavové zpětné vazby a způsob měření jednotlivých stavových veličin. Dále bylo pro nalezení zesílení stavové zpětné vazby použito lineárně kvadratického řízení, kde byla minimalizována hodnota váhové funkce. Tento způsob řízení byl následně otestován na vytvořeném modelu.

Poté se již přešlo k implementaci programového vybavení do řídicího mikrokontroléru mbed NXP LPC1768. Což zahrnovalo řídicí algoritmus regulátoru a ovládání jednotlivých komponent laboratorního modelu včetně komunikace po sběrnici CAN. Programové vybavení mikrokontroléru bylo postaveno na operačním systému reálného času, kterým byl mbed-rtos vyvinutý komunitou mbed.

Vytvořené řešení bylo poté otestováno a ověřena funkčnost stabilizace míčku v horní poloze kola. Dále byla posouzena kvalita vytvořeného laboratorního modelu a uvedeny navržené úpravy modelu do budoucna.

LITERATURA

- [1] VAN GILS, R. W. *Control of the Ball on the Wheel* [online]. Eindhoven, 2007 [cit. 2014-12-21]. Dostupné z: <http://www.mate.tue.nl/mate/pdfs/7848.pdf>. Trainership report. Technische Universiteit Eindhoven.
- [2] MING-TZU HO, YI-WEI TU a HAO-SHUAN LIN. 2009. Controlling a ball and wheel system using full-state-feedback linearization. *IEEE Control Systems Magazine* [online]. 29(5): 93-101 [cit. 2015-05-11]. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5256359>
- [3] ŠOLC, F. a P. VÁCLAVEK. *Modelování a simulace* [online]. 2015 [cit. 2015-04-21]. Dostupné z: <http://bmod.vaclavek.net/>
- [4] *mbed LPC1768*. [online]. [cit. 2014-12-19]. Dostupné z: <http://developer.mbed.org/platforms/mbed-LPC1768/>
- [5] *Digitální servozesilovače TGA-24-9/20*. [online]. [cit. 2014-12-19]. Dostupné z: http://www.tgdrives.cz/fileadmin/user_upload/download/TGA-24-letak.pdf
- [6] TG DRIVES. *Digitální servozesilovač TGA-24-9/20: Instrukční manuál* [online]. 2014 [cit. 2015-04-26]. Dostupné z: http://www.tgdrives.cz/fileadmin/user_upload/download/TGA-24-HW-man-01-2014.pdf
- [7] *Servomotory: Střídavé synchronní servomotory TGN* [online]. 2014 [cit. 2014-12-19]. Dostupné z: http://www.tgdrives.cz/fileadmin/user_upload/download/TG_Drives_2014_Katalog_servomotory_TGN_CZ.pdf
- [8] E-GIZMO MECHATRONIX CENTRAL. *Ultrasonic Sonar Module: Hardware Manual* [online]. 2011 [cit. 2015-04-23]. Dostupné z: <http://www.e-gizmo.com/KIT/images/ultrasonicsonar/ultrasonic%20sonar%20module%20r0.pdf>
- [9] *Distance sensors: FADK 14 (voltage output)* [online]. 2015 [cit. 2015-04-21]. Dostupné z: http://pfinder.baumer.com/pfinder_sensor/downloads/Produkte/PDF/Datenblatt/Optoelektronische_Sensoren/Distanzsensoren_FADK_14_Spannungsausgang_web_EN.pdf
- [10] FREESCALE SEMICONDUCTOR. *High-speed CAN Transceiver* [online]. 2015 [cit. 2015-04-21]. Dostupné z: http://cache.freescale.com/files/analog/doc/data_sheet/MC33901.pdf?pspll=1
- [11] PALUŠ, L. *Popis komunikačního protokolu na sběrnici CAN: Servozesilovač TGA 24-9/20*. [online]. 2010 [cit. 2014-12-21]. Dostupné z: http://www.tgdrives.cz/fileadmin/user_upload/download/TGA_CAN.pdf

- [12] HAVLENA, V. a J. ŠTECHA. 1999. *Moderní teorie řízení*. Praha: Ediční středisko ČVUT.
- [13] BLAHA, P. *Stavový regulátor - stavová zpětná vazba, rekonstruktor stavu* [online]. [cit. 2014-12-21]. Dostupné z: <https://sites.google.com/site/teoriedynamickyhsystemu/prednasky>
- [14] PASSINO, K. M. a N. QUIJANO. 2002. *Linear Quadratic Regulator and Observer Design for a Flexible Joint* [online]. [cit. 2015-05-09]. Dostupné z: <http://www2.ece.ohio-state.edu/~passino/lab4prelab.pdf>
- [15] RTOS. *ARM mbed* [online]. 2015 [cit. 2015-04-28]. Dostupné z: <http://developer.mbed.org/handbook/RTOS>
- [16] mbed SDK. *ARM mbed* [online]. [cit. 2015-04-28]. Dostupné z: <http://developer.mbed.org/handbook/mbed-SDK>
- [17] *CMSIS-RTOS API: Generic RTOS interface for Cortex-M processor-based devices*. [online]. [cit. 2015-05-14]. Dostupné z: <http://www.keil.com/pack/doc/CMSIS/RTOS/html/index.html>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

API	Application Programming Interface – Rozhraní pro programování aplikací
CAN	Controller Area Network – Sběrnice využívána pro komunikaci řídicích jednotek
CMSIS	Cortex Microcontroller Software Interface Standard – Standard softwarového rozhraní pro mikrokontroléry ARM Cortex
FPU	Floating Point Unit – Jednotka pro operace s čísly v pohyblivé řádové čárce
PDO	Process Data Object – Procesní datový objekt
PWM	Pulse Width Modulation - Pulsně šířková modulace
RTOS	Real-time Operating system – operační systém reálného času
SDO	Service Data Object – Parametrizační datový objekt
L	Lagrangeova funkce
Q	zobecněné síly
q	zobecněné souřadnice
V	potenciální energie
m_b	hmotnost míčku
r_w	poloměr kola
r_b	poloměr míčku
T_b	kinetická energie míčku
T_w	kinetická energie kola
J_b	moment setrvačnosti míčku
J_w	moment setrvačnosti kola
T	celková kinetická energie
g	tíhové zrychlení

M	točivý moment působící na kolo
φ_1	úhel mezi osou y a spojnicí středů kola a míčku
φ_2	úhlové natočení kola
φ_3	úhel natočení míčku vztažený k ose y
n	převodový poměr
k_m	momentová konstanta motoru
i_q	momentotvorná složka proudu motorem
R_s	odpor statorového vinutí motoru
L_d	indukčnost vinutí v ose d
L_q	indukčnost vinutí v ose q

SEZNAM PŘÍLOH

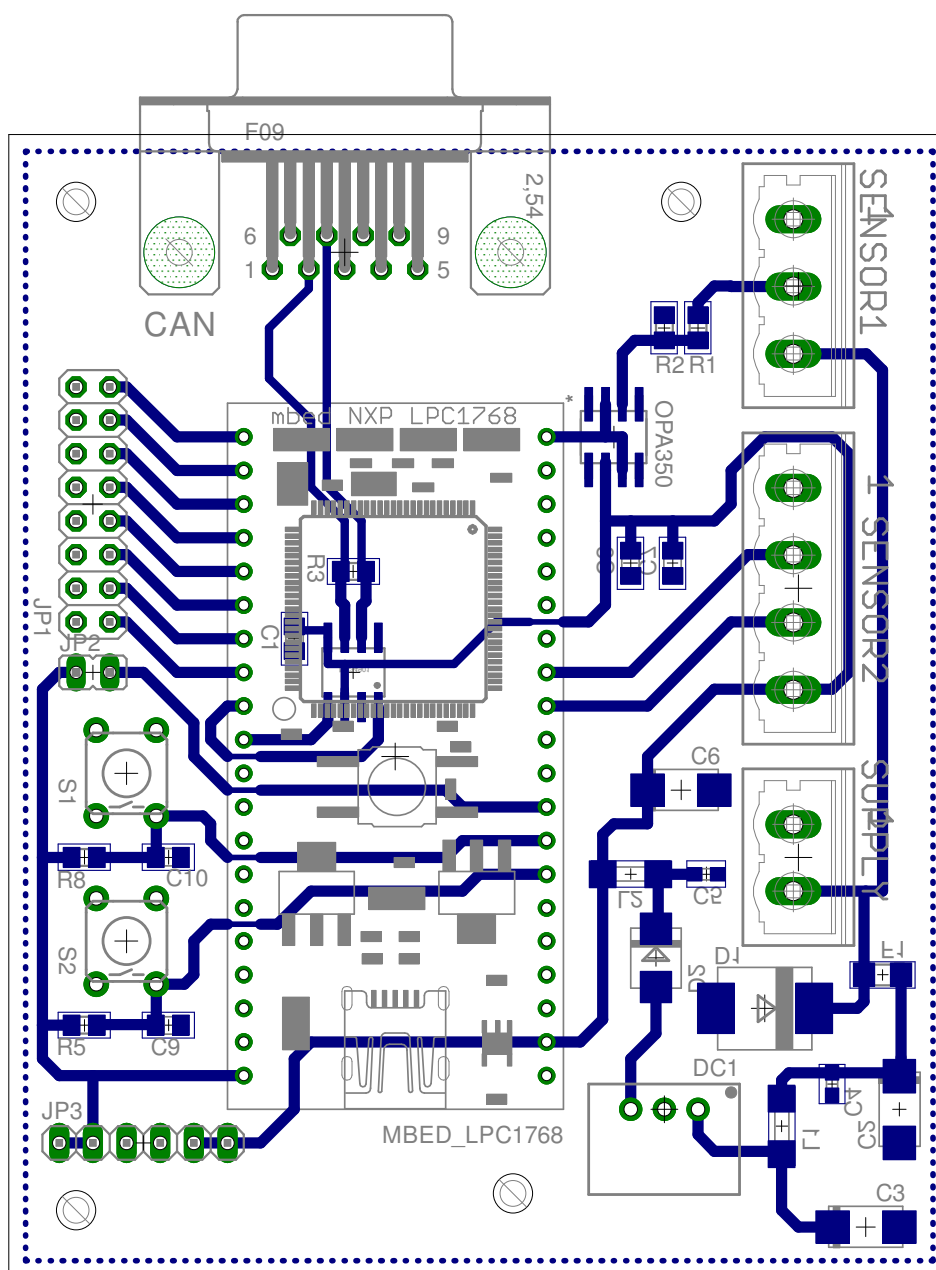
A	Obsah přiloženého CD	72
B	Deska plošných spojů	73
B.1	Schéma zapojení DPS	73
B.2	Návrh DPS	74
C	Class diagramy vytvořených tříd	75

A OBSAH PŘILOŽENÉHO CD

Na přiloženém médiu se nachází plný text diplomové práce ve formátu pdf. Dále vytvořené programové vybavení. Je zde také obsažen m-file s modelem vytvořeným v toolboxu Simulink, obsahující model systému a navržené řízení. Ten byl vytvořen v programu MATLAB 2012. Dále dvě videa ve formátu wmv, na kterých je zobrazena funkčnost stabilizace míčku pro oba dva snímače vzdálenosti.

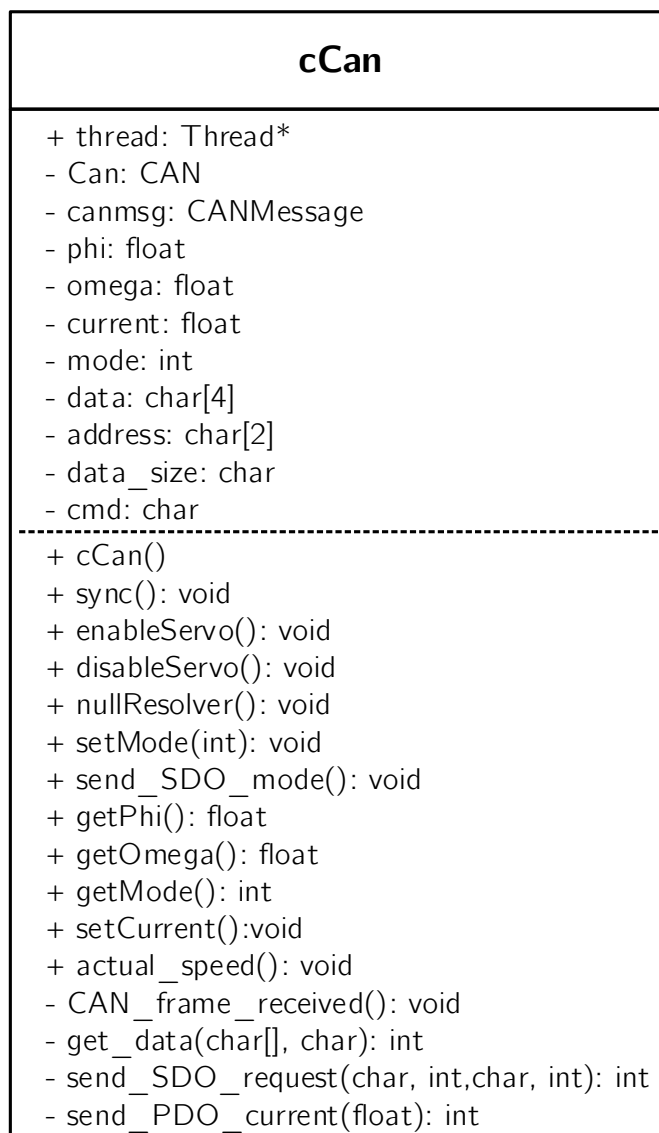
```
dp_text
├── krejci_dp_2015.pdf
└── firmware
    ├── FastAnalogIn
    ├── mbed-rtos
    ├── can.h
    ├── can.cpp
    ├── control.h
    ├── control.cpp
    ├── FastAnalogIn.lib
    ├── ledsensor.h
    ├── ledsensor.cpp
    ├── main.cpp
    ├── mbed.bld
    ├── mbed-rtos.lib
    ├── print.h
    ├── print.cpp
    ├── threads.h
    ├── threads.cpp
    ├── ultrasonic.h
    ├── ultrasonic.cpp
    ├── simulation
    │   ├── simulation.m
    │   └── simulation_model.mdl
    └── video
        ├── fotoelectric_sensor.wmv
        └── ultrasonic_sensor.wmv
```


B.2 Návrh DPS

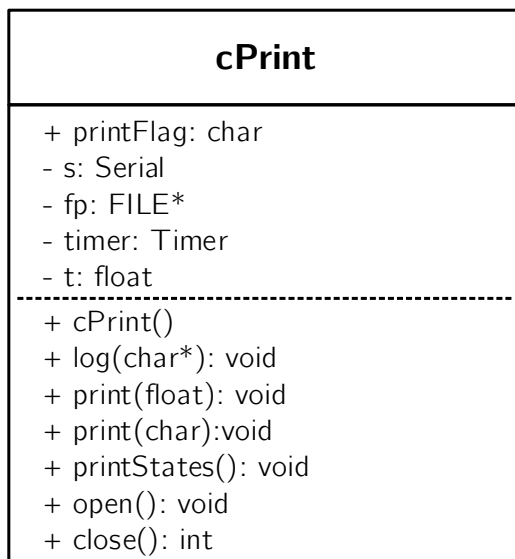


Obr. B.2: Návrh desky plošných spojů

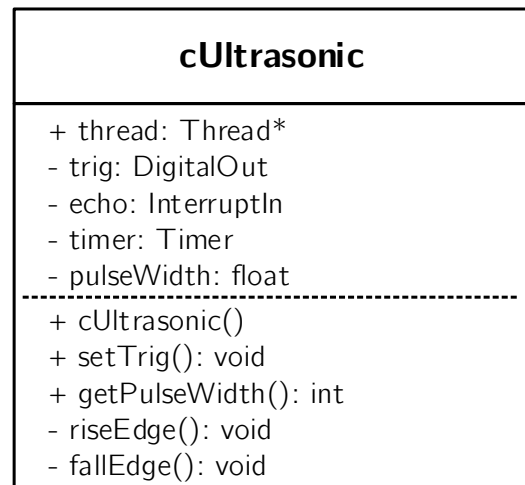
C CLASS DIAGRAMY VYTVOŘENÝCH TŘÍD



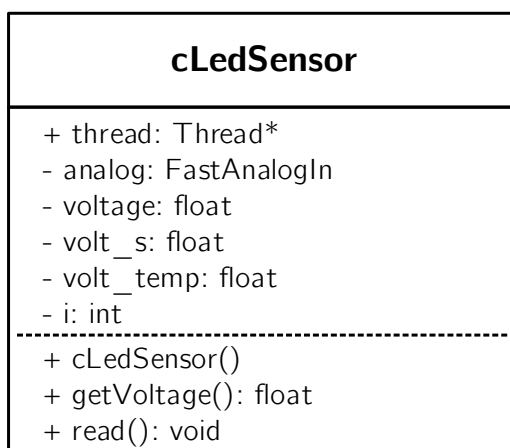
Obr. C.1: Class diagram třídy Can



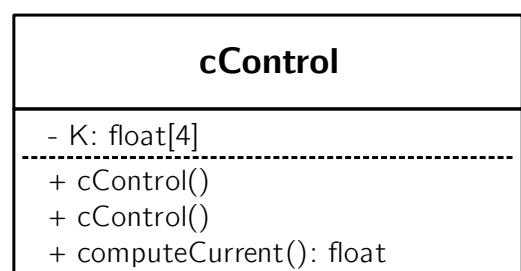
Obr. C.2: Class diagram třídy cPrint



Obr. C.3: Class diagram třídy cUltrasonic



Obr. C.4: Class diagram třídy cLedSensor



Obr. C.5: Class diagram třídy cControl