



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

ŘÍZENÍ POHYBU KULIČKY NA 2D PLOŠINĚ

CONTROL OF BALL AND PLATE SYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Ladislav Bednář

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Libor Veselý, Ph.D.

BRNO 2018



Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Ladislav Bednář

ID: 186028

Ročník: 3

Akademický rok: 2017/18

NÁZEV TÉMATU:

Řízení pohybu kuličky na 2D plošině

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je návrh algoritmu pro řízení pohybu kuličky na 2D plošině se zpětnou polohovou vazbou pomocí kamery.

1. Popište jednotlivé části zařízení, proveďte rozbor metod řízení kuličky na 2D plošině.
2. Sestavte zjednodušený matematický model přípravku.
3. Navrhněte algoritmus pro řízení pohybu kuličky na 2D plošině.
4. Realizujte zpracování obrazu z webkamery a polohování plošiny, verifikujte navržené algoritmy na reálné soustavě.
5. Sestavte laboratorní úlohy pro vybrané předměty bakalářského a magisterského studia.

DOPORUČENÁ LITERATURA:

Soderstrom, T., Stocia, P.: System identification. Prentice Hall International Series in Systems and Control Engineering. UK. July 1989. 528s. ISBN 978-0138812362

Termín zadání: 5.2.2018

Termín odevzdání: 21.5.2018

Vedoucí práce: Ing. Libor Veselý, Ph.D.

Konzultant:

doc. Ing. Václav Jirsík, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské 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

Cieľom predkladanej bakalárskej práce je navrhnutie riadiaceho algoritmu pre riadenie polohy guľičky na 2D plošine so spätnou väzbou realizovanou pomocou kamery.

Venuje sa návrhu ovládača krokových motorov, ktorý je ovládaný analógovými vstupmi. V prvej časti, sa ovládač odladil na platforme CompactRIO, neskoršie sa platforma CompactRIO nahradí mikroprocesorom od firmy Freescale. Pre mikroprocesor je navrhnutá doska plošných spojov. Ďalej sa venuje popisu regulovaného systému matematickým modelom, na ktorý je navrhnutý riadiaci algoritmus. Navrhnutý je lineárny aj nelineárny matematický model, ktorý rozoberá nelinearitu typu valivé trenie. Spracovanie obrazu, tak ako regulačná slučka prebieha v prostredí Matlab-Simulink. Na záver je navrhnutá laboratórna úloha ku prípravku.

Kľúčové slová

guľička na plošine, krokové motory, CompactRIO, mikroprocesor, riadenie

Abstract

The goal of this work is to design a control algorithm for position controlling of the ball on 2D plate with feedback implemented by camera.

Frist part is devoted to the design of stepper motor controller, that is controlled by analog inputs. Controller was tuned on CompactRIO platform later replaced with microprocessor by Freescale. PCB design for microprocessor board is included. Next we described controlled system by mathematical model, which we use for PID controller design. Mathematical model is designed in linear form and nonlinear form, which analyze nonlinearity of rolling friction. Image processing with control loop is contained in Matlab-Simulink workspace. At last, laboratory task is designed.

Keywords

ball on plate, stepper motor, CompactRIO, microprocessor, control

Bibliografická citace:

BEDNÁŘ, L. *Řízení pohybu kuličky na 2D plošině*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 59 s. Vedoucí bakalářské práce Ing. Libor Veselý, Ph.D..

Prohlášení

„Prohlašuji, že svou diplomovou (bakalářskou) práci na téma Řízení pohybu kuličky na 2D plošině jsem vypracoval samostatně pod vedením vedoucí/ho diplomové (bakalářské) 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 uvedené diplomové (bakalářské) práce dále prohlašuji, že v souvislosti s vytvořením této diplomové (bakalářské) práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si 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.

V Brně dne:

.....
podpis autora

Pod'akovanie

Ďakujem vedúcemu bakalárskej práce Ing. Liborovi Veselému, Ph.D za účinnú metodickú, pedagogickú a odbornú pomoc a ďalšie cenné rady pri spracovaní mojej bakalárskej práce.

V Brne dňa:

.....
podpis autora

Obsah

1	Úvod.....	9
1.1	Stav prípravku	9
2	Technické prevedenie úlohy	10
2.1	Krokové motory	10
2.2	Driver krokového motora	12
2.2.1	Driver SLO-SYN® 230-T	13
2.2.2	Driver Leadshine® M542	13
2.3	Zapojenie krokových motorov	13
3	Dynamika motorov	14
3.1	Riadenie motorov	14
3.1.1	Riadenie motorov pomocou variabilnej rýchlosti	14
3.1.2	Riadenie motorov konštantnou rýchlosťou.....	15
4	Ovládače motorov	16
4.1	Platforma CompactRIO	16
4.1.1	NI 9401.....	16
4.1.2	NI 9215.....	16
4.2	Program pre CompactRIO	17
4.3	Procesor Freescale MC56F82743	19
4.4	Návrh analógového vstupu.....	19
4.5	Návrh plošného spoja	22
4.6	Program pre procesor Freescale MC56F82743	22
5	Spätná väzba	25
5.1	Kamera.....	25
5.1.1	Uloženie kamery	25
5.2	Algoritmy detekcie polohy.....	25
6	Prenos riadiaceho signálu	28
6.1	Riadiaca karta MF 642	28
6.1.1	D/A prevodník.....	28
6.2	Riadiaci signál.....	28
7	Matematický model.....	32
7.1	Podsystem gulička.....	32

7.2	Podsystem plošina	33
7.3	Dosadenie do Euler-Lagrangovej metódy.....	33
7.4	Zjednodušenie matematického modelu	35
7.5	Simulácia pohybu motorov	37
7.6	Simulácia valivého trenia	37
7.7	Simulovanie matematického modelu v Matlabe.....	38
8	Riadiaci algoritmus.....	40
8.1	Návrh regulátoru pre matematický model	42
8.2	Spojité regulátor	43
8.3	Diskrétny regulátor	44
8.4	Odozva systému na regulátor	45
9	Laboratórna úloha	46
	Záver	48
	Literatúra	49
	Zoznam symbolov, veličín a skratiek	50
	Zoznam cudzích slov	51
	Zoznam príloh	52

Zoznam obrázkov

Obrázok č. 1 Schematický model prípravku [1]	10
Obrázok č. 2 Hybridný krokový motor [3]	11
Obrázok č. 3 Princíp otáčania hybridného krokového motora [2]	12
Obrázok č. 4 Zapojenie krokových motorov [2]	13
Obrázok č. 5 Spätná väzba na riadenie motorov.....	15
Obrázok č. 6 CompactRio	17
Obrázok č. 7 Grafické užívateľské prostredie CompactRIO.....	17
Obrázok č. 8 Čítač v programe Labview	18
Obrázok č. 9 Operačný zosilňovač v invertujúcom režime	19
Obrázok č. 10 Operačný zosilňovač v invertujúcom režime so zdrojom referenčného napätia.....	20
Obrázok č. 11 Operačný zosilňovač v neinvertujúcom režime	21
Obrázok č. 12 Napäťový delič.....	21
Obrázok č. 13 Kompletná schéma pre operačný zosilňovač	22
Obrázok č. 14 Algoritmus kalibrácie.....	23
Obrázok č. 15 Prevedenie kalibrácie v jazyku C.....	23
Obrázok č. 16 Prevedenie prerušenia v jazyku C.....	24
Obrázok č. 17 Spracovanie obrazu v prostredí Simulink.....	27
Obrázok č. 18 Spôsob prenosu riadiaceho signálu.....	29
Obrázok č. 19 Prepočet riadiaceho signálu Matlab Simulink.....	30
Obrázok č. 20 Prepočet riadiaceho signálu v prostredí Labview	30
Obrázok č. 21 Prepočet riadiaceho signálu v jazyku C.....	31
Obrázok č. 22 Chyba snímania	35
Obrázok č. 23 Znázornenie valivého trenia [12].....	37
Obrázok č. 24 Skoková charakteristika krokových motorov	38
Obrázok č. 25 Simulácia matematického modelu.....	39
Obrázok č. 26 Simulácia nelineárneho matematického modelu.....	39
Obrázok č. 27 Plošná regulácia	40
Obrázok č. 28 Schematické prevedenie PID regulátoru [13]	41
Obrázok č. 29 Koreňový hodograf regulovanej sústavy.....	42
Obrázok č. 30 Frekvenčná charakteristika.....	43
Obrázok č. 31 Odozva systému na skokovú zmenu	43
Obrázok č. 32 Prevedenie spojitého PD regulátora	44
Obrázok č. 33 Prevedenie diskrétného PD regulátora	44
Obrázok č. 34 Odozva systému na skokovú zmenu	45
Obrázok č. 35 Odozva systému na sledovanie trajektórie	45

Zoznam tabuliek

Tabuľka 1 Popis dôležitých indikátorov:	18
Tabuľka 2 Hodnoty použitých súčastok:	22

1 ÚVOD

Riadenie pohybu guľičky po náklonnej plošine znie z počiатku ako jednoduchá úloha. Treba si však uvedomiť, že nebudeme riadiť len samotný systém guľičky na plošine, ale aj motory, ktoré ju nakláňajú. Riadenie bez merania nie je možné a kvalita spätnej väzby je dôležitým faktorom. Samotná spätná väzba môže byť riešená rôznymi spôsobmi. Jedným z najlepších riešení je rezistívny panel, s ktorým by sme získali možnosť vzorkovať polohu guľičky vysokou frekvenciou. My budeme využívať kameru so vzorkovacou periódou 120 snímok za sekundu. Týmto sa plynulo dostávame k faktu, že budeme systém riadiť v diskretnom čase.

Na realizáciu riadiaceho algoritmu využijeme počítač, ktorý bude spracovávať obraz a vyhodnocovať polohu guľičky na náklonnej plošine. Na túto úlohu využijeme program Matlab s prostredím Simulink, v ktorom vytvoríme regulačnú schému a samotný riadiaci algoritmus. Výstupom z počítača bude vďaka riadiacej karte Humusoft analógový signál, ktorého hodnota bude predstavovať požadovaný náklon plošiny. Analógový signál bude spracovávať a prepočítavať medzičlen, ktorý budeme volať ovládač krokových motorov. Algoritmus pre ovládač sa pokúsime odladiť na platforme CompactRIO a neskoršie preniesť na mikroprocesor firmy Freescale. Ovládač bude spracovávať analógový signál a vypočíta koľko pulzov bude potrebné odoslať na výkonové ovládače krokových motorov.

V teoretickej časti je popísaný systém pomocou matematického modelu. Predstavíme si hlavné nelinearity ovplyvňujúce systém a pokúsime sa navrhnuť regulátor na riadenie systému. Poslednou časťou tejto práce bude návrh laboratórnej úlohy.

1.1 Stav prípravku

Prípravok sa nachádzal v nefunkčnom stave. Nachádzali sa na ňom krokové motory, pôvodné drivre krokových motorov a kamera na detekciu polohy guľičky so vzorkovacou periódou 30 snímok za sekundu.

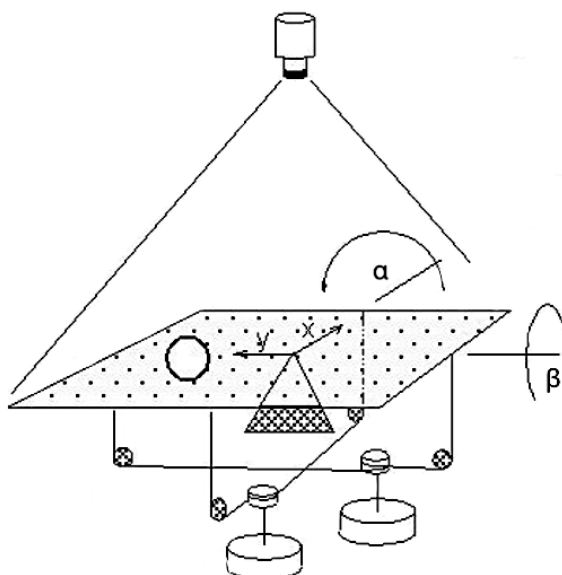
2 TECHNICKÉ PREVEDENIE ÚLOHY

Na prípravku sa nachádzajú dva krokové motory, ktoré ovládajú náklon na plošine. Plošina samotná je umiestnená na kĺbe, ktorý umožňuje ohyb v oboch osiach prípravku. Na hriadeli motora je umiestnený navijak, okolo ktorého je niekoľkokrát omotané oceľové lanko. Obidve strany lanka sú pevne spojené s plošinou.

Jedinou viditeľnou chybou je vôľa na kĺbe, ktorá ale nemá v konečnom dôsledku až taký veľký vplyv na nakláňanie plošiny.

Ku hlavnému rámu je pripevnený statív, na ktorom je umiestnená web kamera. Statív má tri stupne voľnosti, vďaka ktorým sme schopní kameru nastaviť presne na snímanie plošiny. Rozmery plošiny sú 400x400 mm s možnosťou náklonu približne o 11° na každú stranu. Schematické prevedenie prípravku:

Obrázok č. 1 Schematický model prípravku [1]



2.1 Krokové motory

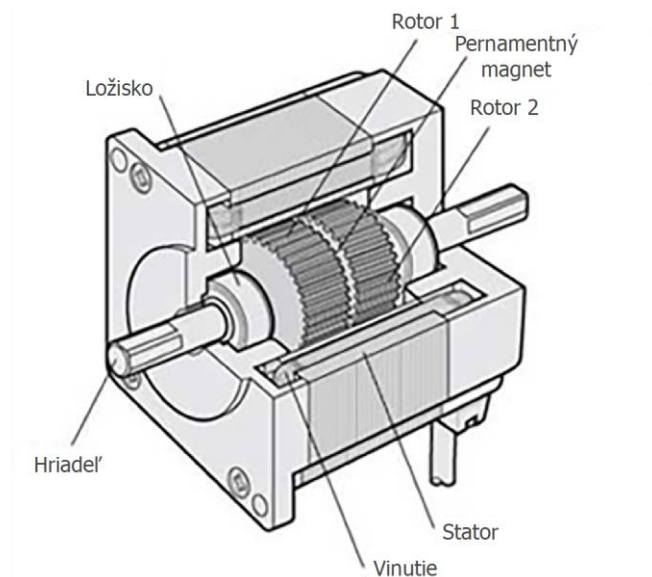
Krokové motory sú synchronne impulzne budené elektrické stroje, ktorých pohyb nie je plynulý, ale pozostáva z jednotlivých krokov. Medzi hlavné výhody patrí nepotrebnosť zložitejších regulačných obvodov na ich ovládanie. Pokiaľ motory nie sú preťažované, sme schopní obísť sa aj bez spätnej väzby. Polohovanie alebo konštantná rýchlosť otáčania sa realizuje počítaním krokov, ten je daný počtom pólových dvojíc, teda konštrukčným prevedením motora. Počet krokov je možné zväčšiť spôsobom ovládania. Ku mechanickému kontaktu dochádza jedine v ložiskách motora, z toho plynú ďalšie výhody, ako je životnosť či minimálna údržba. [2]

Krokové motory môžeme rozdeliť na rotačné a lineárne, ktoré sa ďalej rozdeľujú na niekoľko základných druhov, vzhľadom na ich konštrukciu a princípy činnosti. Základné kategórie krokových motorov sú:

- motory s premenlivou reluktanciou,
- motory s permanentnými magnetmi,
- hybridné motory.

Z nášho pohľadu sú však zaujímavé najmä hybridné krokové motory, ktoré sú v skutočnosti najčastejšie používané. Kombinujú princíp krokového motora s permanentným magnetom a krokového motora s premenlivou reluktanciou. Vďaka jeho konštrukcii umožňuje veľmi malý uhol kroku a vysoký hmotný výkon. Dôležitým znakom hybridného krokového motora je jeho rotor, ktorý je zobrazený na *Obrázok č. 2*. Ten sa skladá z valcového permanentného magnetu, ktorý je magnetizovaný pozdĺžne, to znamená v rovine rovnobežnej s hriadelom a vytvára unipolárne pole. Na oboch koncoch magnetu sa nachádzajú rotorové polové nástavce, ktoré majú po obvode zuby. Tieto dva polové nástavce sú totožné a sú voči sebe natočené o jednu polovicu zubového rozstupu rotora. Pólové nástavce rotora, na obrázku označené rotor 1 a rotor 2, sú väčšinou zložené z kremíkových plechov.

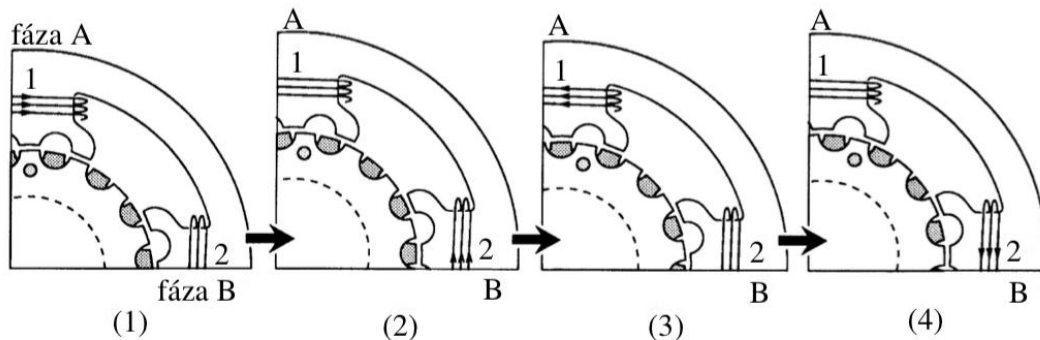
Obrázok č. 2 Hybridný krokový motor [3]



Princíp otáčania hybridného krokového motora je znázornený na *Obrázok č. 3*. V stave (1) je budená fáza A a pól 1 pritiahne rotorové zuby severného pólu, zatiaľ čo statorový pól 3 pritiahne zuby rotora južného pólu. V stave (2) je zopnutá fáza B (smer prúdu je znázornený šípkami). Rotor sa pootočí o štvrtinu zubového rozstupu a jeho južný pól (vyšrafovaný) sa zoradí so statorovým pólom 2, pričom jeho severný pól sa zoradí so statorovým pólom 4. V ďalšom kroku je opäť napájaná fáza A, ale opačnou polaritou prúdu ako v stave (1). Rotor sa pootočí o ďalšiu štvrtinu zubového rozstupu a zaujme pozíciu (3). Zuby sa zoradia v opačnej

magnetickej polarite ako v stave (1), to znamená, že teraz je južný pól rotora zoradený so statorovým pólom 1 a severný pól rotora so statorovým pólom 3. Ak ďalej zopneme prúd opačnej polarity vo fáze B, vytvorí sa rovnovážny stav (4), ako výsledok následného pootočenia rotora o ďalšiu štvrtinu zubového rozostupu. [2]

Obrázok č. 3 Princíp otáčania hybridného krokového motora [2]



Krokový motor nemôže byť rozbehnutý okamžite na plnú rýchlosť a taktiež ho nemôžeme okamžite zastaviť a to z dôvodu straty krokov. Motory teda musíme rozbiehať pomocou nábehovej rampy. Z mechanického pohľadu sa jedná o postupné zväčšovanie počtu krokov za daný čas.

Rýchlosť motorov je ovládaná frekvenciou pulzov TTL signálu. V našom prípade sa jedná o postupné zvyšovanie frekvencie signálu, až sa dostaneme na maximálne otáčky motora. Postup musíme zopakovať aj pre spomaľovanie, kde naopak frekvenciu znižujeme. Moment, pri ktorom motor nestratí kroky počas rozbiehania alebo zastavovania, nazývame pull-out točivý moment. Tento moment určuje strmosť rampy na rozbiehanie/zastavovanie, je ovplyvnený hlavne výkonom motora, ale aj veľkosťou záťaže naňho pripojenou. Keďže na ovládanie motorov sú použité nové drivre, jediným spôsobom akým sme schopný určiť strmosť rampy, je experimentálne testovanie pre rôzne rýchlosti.

2.2 Driver krokového motora

Na riadenie polohy krokových motorov je potrebné použitie takzvaných driverov. Sú to elektronické zariadenia, ktoré na základe počtu pulzov vstupného signálu pootočia motor do požadovanej polohy. Jeden pulz na vstupe driveru znamená jeden krok motora.

Nainštalované drivre *SLO-SYN® 230-T* nemajú plynulý chod v spolupráci s motormi. Motory sú hlučné a nedosahujú presnosť v polohovaní na nami požadovanej úrovni, z toho dôvodu boli zakúpené novšie drivre *Leadshine® M542*, ktoré sú schopné ovládať motory s vyššou presnosťou. Presnosť pre nás

predstavuje jemnejšie krokovanie, ktoré môžeme dosiahnuť použitím mikrokrokovania.

Mikrokrokovanie umožňuje ovládanie krokových motorov na väčší než daný počet krokov motoru. Jeden krok môžeme v skutočnosti rozdeliť až na 128 menších krokov.

2.2.1 Driver SLO-SYN® 230-T

Driver nemá možnosť mikrostepingu. Je navrhnutý špeciálne na riadenie nainštalovaného motoru. Ponúka možnosť plného alebo polovičného krokovania s maximálnou rýchlosťou 10 000 krokov za sekundu. [4]

2.2.2 Driver Leadshine® M542

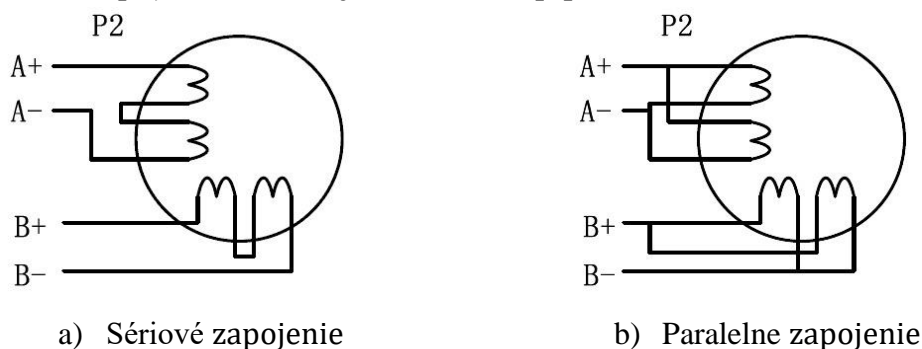
Jedná sa o vysoko výkonný mikro-krokový driver. Je určený na riadenie dvojfázových a štvorfázových hybridných motorov. Ponúka možnosť rozdelenia jednej otáčky krokového motora až na 25 600 krokov pre motory s $1,8^\circ$ na jeden krok. Maximálna rýchlosť je 300 000 krokov za sekundu. [5]

2.3 Zapojenie krokových motorov

Zapojenie výrazne ovplyvňuje chovanie krokových motorov. V našom prípade je veľmi dôležitá rýchlosť. Motory použité na prípravku sú typu *SLO-SYN® M061-LE08*, jedná sa o krokové motory hybridného typu s ôsmimi vinutiami.

Motory môžu byť zapojené dvomi spôsobmi a to s vinutiami zapojenými sériovo (*Obrázok č. 4a*) alebo paralelne (*Obrázok č. 4b*). Pri sériovom zapojení získame veľký točivý moment pri nižších rýchlostiach a naopak paralelne zapojenie ponúka veľký točivý moment aj pri vyšších rýchlostiach.

Obrázok č. 4 Zapojenie krokových motorov [2]



3 DYNAMIKA MOTOROV

Jedná sa o jednu z najväčších prekážok celého projektu. Na požadovanú polohu sa musíme vedieť dostať s čo najväčšou rýchlosťou a presnosťou. Z pohľadu riadenia by bolo najlepšie nachádzať sa na požadovanej polohe okamžite. V skutočnosti sa však na požadovanú polohu dostaneme až za určitý čas a ten je určený len rýchlosťou náklonu plošiny. Náklon plošiny ovládame pomocou motorov a teda jeho rýchlosť závisí na otáčkach motorov. Linearizácii pohybu motorov sa práca venuje v kapitole 7.7.

3.1 Riadenie motorov

Keďže riadenie prebieha v reálnom čase, realizovanie rozbehovej rampy môže byť problematické. Riešením môžu byť dve varianty, **prvou variantou** bude vzorkovanie riadiaceho signálu. Medzi vzorkami budeme motor rozbiehať a zastavovať. **Druhou variantou** bude konštantná rýchlosť otáčok na motoroch.

3.1.1 Riadenie motorov pomocou variabilnej rýchlosti

Riadiaci signál bude vzorkovaný nami určeným časovým intervalom. Motor sa začne od začiatku rozbiehať a následne spomaľovať, ako sa bude blížiť ku koncu časového úseku.

V tomto prípade by sme mohli za výstup, teda riadiaci signál, posielat' požadovanú rýchlosť. Poloha by bola počítaná v PC ako integrácia rýchlosti. Druhou možnosťou je na riadiaci signál stále posielat' požadovanú polohu a vďaka stálej dĺžke časového úseku, stanovovať rampu pre rozbeh alebo spomaľovanie krokových motorov.

Vzniká tu však obmedzenie maximálnej zmeny vstupného signálu. Môže vznikat' neplynulý chod motorov, kvôli zastavovaniu a rozbiehaniu, aj keď motory sa budú chovať ako jednoduchý filter. Ďalším obmedzením je samotná vzorkovacia frekvencia, pokiaľ by jej hodnoty boli príliš veľké, dochádzalo by k nepresnému kopírovaniu riadiaceho signálu. Vzorkovaciu frekvenciu by sme museli voliť na základe parametrov strmosti rampy. Aby malo zmysel rozbiehať a zastavovať krokové motory, vzorkovacia frekvencia by musela byť dostatočne dlhá na rozbehnutie motora, aspoň na polovicu maximálnej rýchlosti otáčok.

Výhoda tohto riešenia však spočíva v minimálnej strate pulzov, ktoré riadia krokové motory.

3.1.2 Riadenie motorov konštantnou rýchlosťou

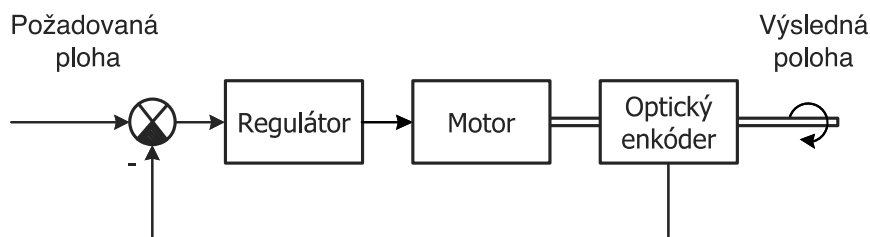
Riadiaci signál bude sledovaný konštantnou rýchlosťou otáčok motorov. Vstupný, teda riadiaci signál bude vzorkovaný a rovno prepočítavaný na počet pulzov, o aký sa má krokový motor posunúť.

Jednou z nevýhod tohto riešenia je však strata riadiacich pulzov pri vyšších rýchlostiach. Na rozbeh motorov môžeme strácať neznámy počet pulzov. Tento problém vyúsťuje k strataniu počiatkovej nulovej polohy, čo by v konečnom dôsledku mohlo spôsobiť nepresnosti v regulácii požadovaného deja. Najjednoduchším spôsobom potlačenia tohoto problému je zníženie rýchlosti motorov, ale len do rýchlosti, v ktorej sme schopní sledovať riadiaci signál.

Výhodou však je jednoduchá implementácia do programu, keďže riadiaci signál len násobíme konštantou, ktorá zodpovedá počtu pulzov, ktoré sú potrebné na dosiahnutie požadovanej polohy/náklonu.

Ak by sme pulzy naďalej strácali, mohli by sme motory zapojiť do jednoduchého regulačného obvodu, kde by spätnú väzbu realizoval optický enkodér, ako je zobrazené na *Obrázok č. 5*.

Obrázok č. 5 Spätná väzba na riadenie motorov



4 OVLÁDAČE MOTOROV

Na ovládanie diverov krokových motorov používame pulzujúci signál, ktorý riadiaca karta v PC nie je schopná vygenerovať v požadovanej rýchlosti. Z tohoto dôvodu je potrebné dodať medzičlen, ktorý bude zdrojom takéhoto signálu. Vstupom do ovládača budú analógové signáli a výstupom budú digitálne signáli.

Na odladenie algoritmu použijeme platformu CompactRIO. Ďalej sa budeme venovať nahradeniu CompactRia pomocou procesoru a návrhu plošného spoju.

4.1 Platforma CompactRIO

CompactRIO je platforma obsahujúca rad vstavaných riadiacich jednotiek, ktorých účelom je spracovanie signálov v reálnom čase. Obsahuje realtime procesor, užívateľom programovateľné FPGA na implementovanie vysoko rýchlostného riadenia a vlastného nastavenia časovania či triggrovania priamo v hardvéri.

Skladá sa z takzvaných šasi, do ktorých môžeme pripojiť nami zvolené moduly, podľa potreby a aplikácie. V našom prípade sme použili dva vstupno-výstupné digitálne moduly a jeden vstupný analógový modul.

Použitý je šasi typu cRio-9076 (*Obrázok č. 6a*) a jeho použitie je schematicky zobrazené na (Príloha č. 1). Na schéme je vyobrazené zapojenie koncových snímačov, driverov motorov a riadiacej karty ku konkrétnym vývodom na moduloch CompactRIO (z dôvodu prehľadnosti v schéme nie je zobrazené spoločné uzemnenie).

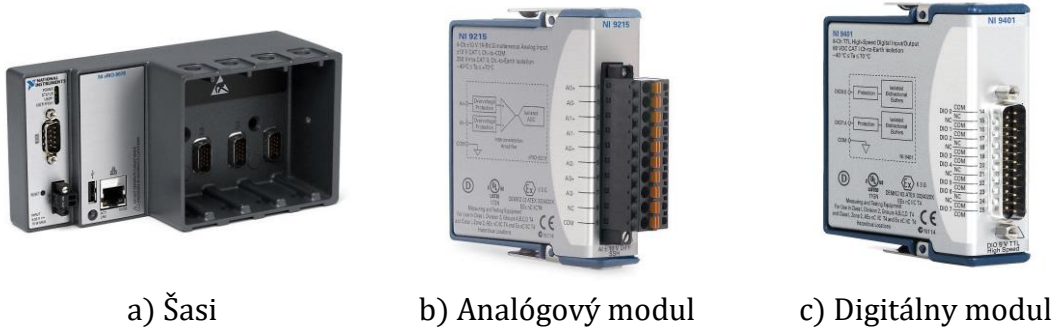
4.1.1 NI 9401

Je to obojsmerný digitálny modul. Osem DIO kanálov je zoskupených do dvoch portov, ktoré je možné nakonfigurovať nezávisle od vstupu alebo výstupu. Modul je možné využiť ako čítač, časovač alebo ako generátor pulzov či iné. (*Obrázok č. 6c*) [6]

4.1.2 NI 9215

Jedná sa o vstupný analógový modul, ktorý obsahuje štyri simultánne vzorkované vstupné analógové kanále s 16 bitovým analógovo-digitálnym prevodníkom a postupne aproximáčnym registrom (SAR). (*Obrázok č. 6b*) [7]

Obrázok č. 6 CompactRio



a) Šasi

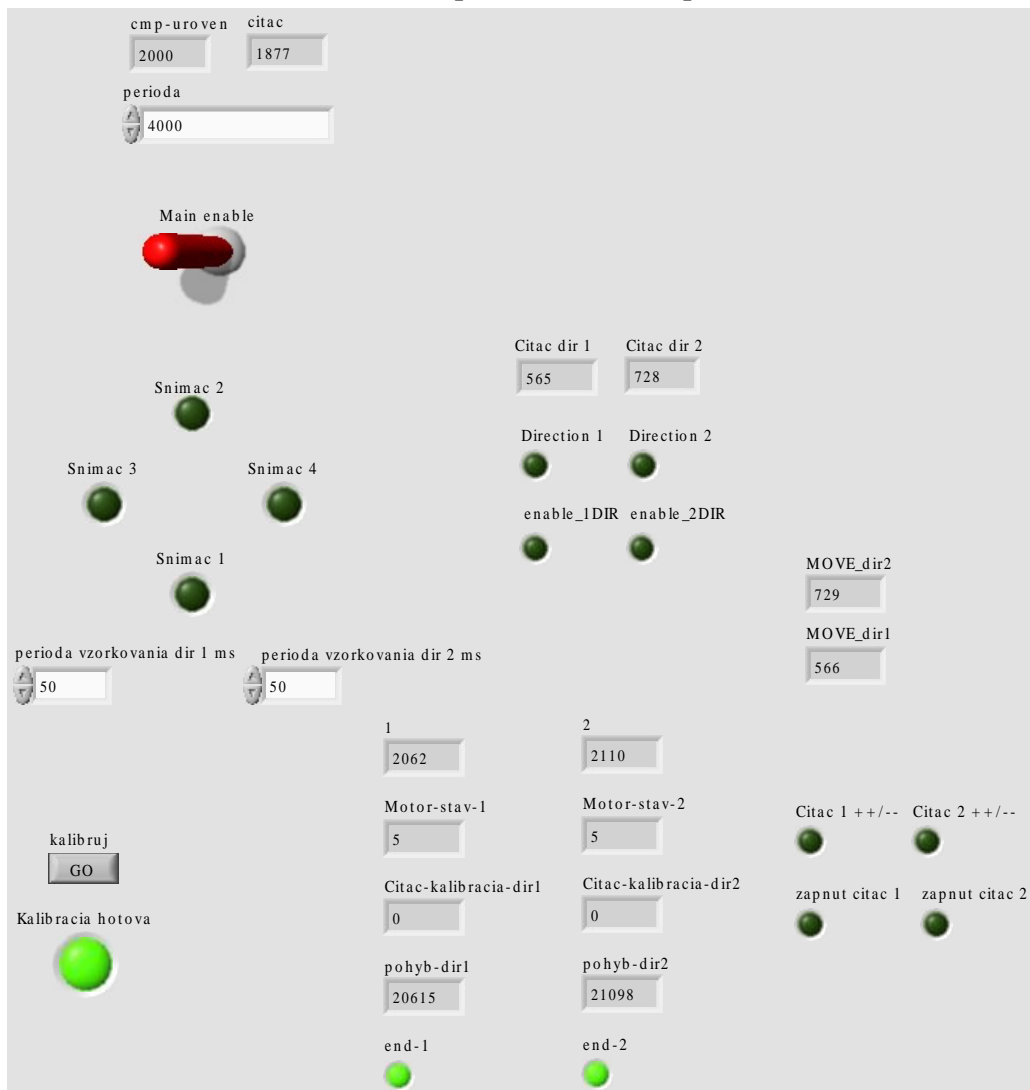
b) Analógový modul

c) Digitálny modul

4.2 Program pre CompactRIO

Skladá sa zo samotného kódu, ale aj s užívateľského prostredia, z ktorého je možné ovládať rýchlosť aj kalibráciu. Užívateľské prostredie (Obrázok č. 7). Popis najdôležitejších indikátorov nájdeme v *Tabuľka 1*

Obrázok č. 7 Grafické užívateľské prostredie CompactRIO

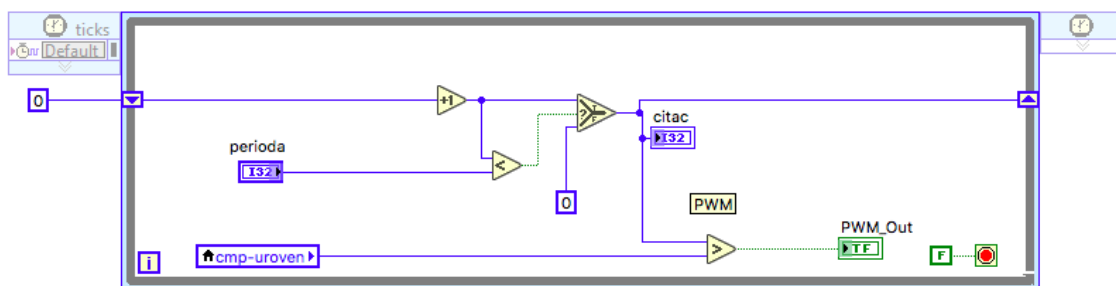


Tabuľka 1 Popis dôležitých indikátorov:

Indikátor:	Popis:
Main enable	Zastavuje/spúšťa motory
Periódá	Možnosť nastavenia rýchlosti výstupného PWM signálu
Cmp - úroveň	Hodnota striedy pulzujúceho signálu (automaticky polovica periódy)
Periódá vzorkovania dir 1 ms a dir 2 ms	Možnosť nastavenia rýchlosti vzorkovania A/D prevodníku
Kalibruj	Možnosť spustenia kalibrácie
1 a 2	Počet krokov k jednotkovej zmene A/D prevodníku
Motor-stav-1 a Motor-stav-2	Hodnota poukazujúca na stav stavového automatu
Čítač-kalibracia-dir1 a dir2	Hodnota čítača pre kalibráciu, vyčítaním tejto hodnoty sme schopný určiť konštanty 1 a 2
Pohyb-dir1 a Pohyb-dir2	Hodnota vyjadruje počet krokov z krajnej polohy na koncovom snímači do polohy zkalibrovanej
End-1 a End-2	Kontrolka vyjadruje dokončenie kalibrácie v jednotlivých smeroch
Kalibrácia hotová	Kontrolka vyjadruje ukončenie kalibrácie v oboch smeroch
Snímač 1,2,3,4	Kontroly koncových snímačov
MOVE_dir1 a MOVE_dir2	Hodnota vyjadruje aktuálny počet korek na dosiahnutie požadovanej polohy
Čítač dir 1 a Čítač dir 2	Hodnota vyjadruje aktuálny počet krokov
Direction 1 a Direction 2	Kontrolka signalizuje smer, ktorým sa motory hýbu
Enable dir1 a Enable dir2	Kontrolka signalizuje stav, kedy je povolený pohyb krokových motorov

Najdôležitejšou súčasťou samotného programu sú čítače. Pomocou čítačov sme schopní vytvoriť pulzujúci signál na riadenie motorov, ako aj počítanie pulzov potrebných na dosiahnutie polohy. Príklad čítača na vytvorenie pulzujúceho signálu je na (Obrázok č. 8).

Obrázok č. 8 Čítač v programe Labview



4.3 Procesor Freescale MC56F82743

Procesor z rodiny 56800EX. Jedná sa o digitálny signálový kontrolér (DSC), ktorý je založený na 32-bitovom jadre s využitím duálnej Harvardskej architektúry. Je vytvorený spojením digitálnych signálových procesorov a mikroprocesorov. Využíva sa v aplikáciách ako priemyselné riadenie, v napájacích rozvodoch, fotovoltaiických systémoch, bezdrôtovom nabíjaní, riadení motorov a mnohých ďalších. Pracuje na frekvencii do 100MHz v rýchlom režime alebo 50MHz v normálnom režime. Veľkosť flash pamäte je 64KB a RAM 8KB. Obsahuje taktiež vstupno-výstupné periférie, z ktorých je potrebné spomenúť dva 12-bitové analógovo digitálne prevodníky s ôsmimi kanálmi, GPIO piny a PWM moduly. Procesor má napájanie 3,0V až 3,6V a je schopný spracovávať 5V I/O signály. Budeme používať 32-nožičkové prevedenie.

4.4 Návrh analógového vstupu

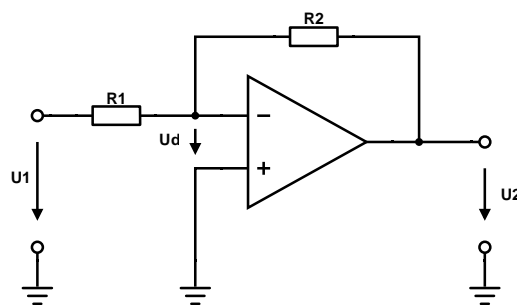
Na analógový vstup je potrebné pripojiť filtráciu kvôli šumu, ktorý môže vytvárať plošný spoj. Filtráciu vyriešime pomocou dolnej prepusti, teda RC článkom. Jeho zapojenie do obvodu môžeme vidieť na schéme (Obrázok č. 13). Na výpočet medznej frekvencie (frekvencia pri ktorej zisk poklesne o -3dB) platí vzťah:

$$f_0 = \frac{1}{2\pi R_{filt} C1} \quad (1)$$

Potrebujeme odfiltrovať frekvencie vyššie ako desiatky kilo-Hertz, takže môžeme navrhnúť filter s hodnotami $R_{filt} = 100\Omega$ a $C1 = 2,2nF$, pri ktorých je medzná frekvencia rovná $f_0 = 72,343KHz$, čo zhruba vyhovuje našim požiadavkam.

Na úpravu vstupného signálu z rozsahu $\pm 10V$ na 0 až 3,3V použijeme zapojenie operačného zosilňovača v invertujúcom režime.

Obrázok č. 9 Operační zosilňovač v invertujúcom režime

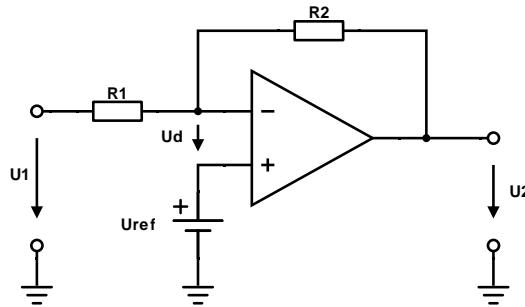


Pre operačný zosilňovač v invertujúcom zapojení platí za podmienok $U_d = 0$ a prúdy, ktoré tečú cez odpory $R1$ a $R2$ sú si rovné, nasledujúci vzťah [8]:

$$U2 = -U1 \frac{R2}{R1} \quad (2)$$

V tomto prípade však použijeme modifikované zapojenie, v ktorom pripojíme U_{ref} na kladný vstup operačného zosilňovača.

Obrázok č. 10 Operačný zosilňovač v invertujúcom režime so zdrojom referenčného napätia



Na výpočet hodnôt odporov a referenčného napätia U_{ref} môžeme s výhodou použiť princíp superpozície, keďže operačný zosilňovač považujeme za lineárnu súčiastku. Skratujeme zdroj napätia U_{ref} a pokračujeme s výpočtom pre invertujúci zosilňovač. Požadovaný vstupný rozsah je $\pm 10V \Rightarrow 20V$ a výstupný rozsah 0 až $3,3V \Rightarrow -3,3V$. Hodnoty dosadíme do vzťahu (1).

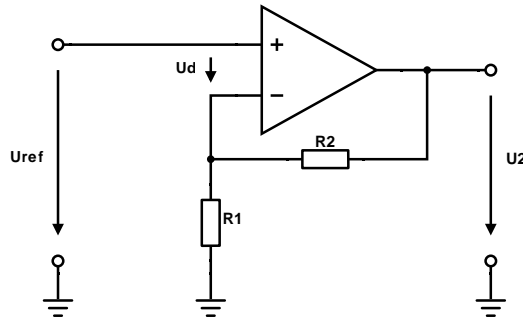
$$-3,3 = -20 \frac{R2}{R1} \quad (3)$$

Následne je potrebné zvoliť veľkosť odporu $R1$. Do tohto odporu však musíme započítať všetky odpory zapojené do série s ním. V našom prípade sa jedná o filter, ktorý sa nachádza na vstupe. Teda $R1 = R_{filt} + R_{zosil}$. Hodnotu $R1$ volíme s ohľadom na zdroj analógového signálu. Zdrojom je riadiaca karta MF624. Pri napätiach v hodnotách $\pm 10V$ bude vhodné zvoliť veľkosť odporu $R1$ v hodnotách okolo $10K\Omega$, aby prúd z riadiacej karty bol čo najmenší. Veľkosť prúdu však nemôže byť príliš malá a to kvôli šumu, ktorý by sme mohli spôsobiť. Pokračovať teda budeme s hodnotou $R1 = 100 + 12000 = 12100\Omega$

$$-3,3 = -20 \frac{R2}{12100} \Rightarrow R2 = 1996,5\Omega \doteq 2000\Omega \quad (4)$$

Pre výpočet hodnoty referenčného napätia U_{ref} skratujeme zdroj vstupného napätia $U1$ a schému (Obrázok č 10.) si môžeme predstaviť ako neinvertujúci zosilňovač.

Obrázok č. 11 Operačný zosilňovač v neinvertujúcom režime



Pre operačný zosilňovač v neinvertujúcom režime za podmienok $U_d = 0$ a prúdy, ktoré tečú cez odpory R_1 a R_2 sú si rovné, nasledujúci vzťah [8]:

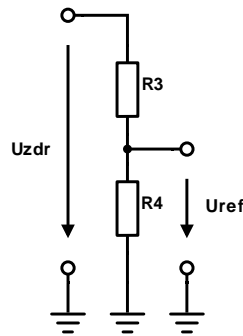
$$U_2 = U_{ref} \left(1 + \frac{R_2}{R_1} \right) \quad (5)$$

Keďže uvažujeme že vstupné napätie $U_1 = 0$ na výstupe U_2 požadujeme polovicu rozsahu teda $U_2 = 3,3/2 = 1,65V$. Po dosadení za hodnotu odporov a hodnotu U_2 dostávame:

$$1,65 = U_{ref} \left(1 + \frac{2000}{12100} \right) \Rightarrow U_{ref} = 1,416V \quad (6)$$

Hodnotu U_{ref} nastavíme pomocou napäťového deliča (Obrázok č. 12) zo zdroja napätia 3,3V.

Obrázok č. 12 Napäťový delič



Vzťah pre výpočet deliča napätia:

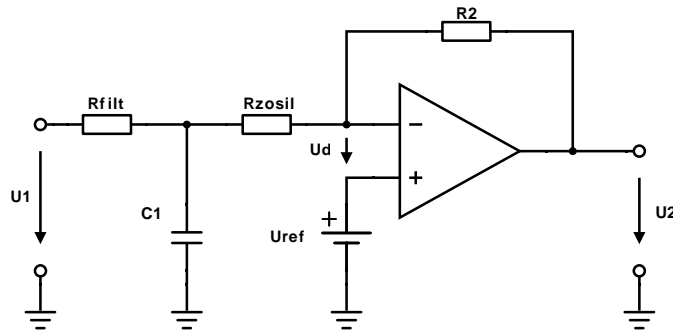
$$U_{ref} = U_{zdr} \left(\frac{R_4}{R_3 + R_4} \right) \quad (7)$$

Hodnotu odporov volíme opäť vhodne ku zdroju napätia, zvolíme teda $R_4 = 2000\Omega$. Po dosadení hodnôt:

$$1,416 = 3,3 \left(\frac{2000}{R_3 + 2000} \right) \Rightarrow R_3 = 2661\Omega \doteq 2700\Omega \quad (8)$$

Kompletnú schému zapojenia môžeme vidieť na obrázku (Obrázok č. 13) a platný vzťah v rovnici č (9).

Obrázok č. 13 Kompletná schéma pre operačný zosilňovač



$$U_2 = U_{ref} \left(1 + \frac{R_2}{R_{filt} + R_{zosil}} \right) - U_{ref} \left(\frac{R_2}{R_{filt} + R_{zosil}} \right) \quad (9)$$

Tabuľka 2 Hodnoty použitých súčiastok:

Názov súčiastky:	Hodnota:
R_{filt}	100Ω
R_{zosil}	12000Ω
R_2	2000Ω
R_3	2000Ω
R_4	2700Ω
C_1	2,2nF

Kompletný zoznam súčiastok je v prílohe (Príloha č. 5).

4.5 Návrh plošného spoja

Dôležitou súčasťou je spoločné napájanie, ktoré je zabezpečené spínaným zdrojom LM2594MX. Vstupné napätie je v rozsahu 5V až 40V. Výstupné napätie je 5V pri výstupnom prúde 500mA. Tento zdroj je využitý na napájanie operačných zosilňovačov rail-to-rail a má vyvedený výstup na pripojenie ďalšieho spotrebiča, pôvodne plánovaný na možnosť pripojenia prísvetlenia pre kameru.

Schému zapojenia môžeme vidieť v prílohe (Príloha č. 3). Návrh samotnej DPS sa nachádza v prílohe (Príloha č. 2). Fotodokumentácia kompletne osadenej DPS je v prílohe (Príloha č. 4).

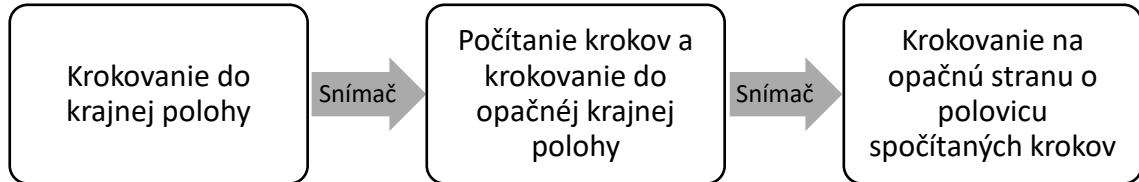
4.6 Program pre procesor Freescale MC56F82743

Na vytvorenie programu použijeme prostredie CodeWarrior Eclipse 11.0 s balíčkom DSP a na programovanie procesora USB Tab od Freescale.

Program sa opäť skladá z dvoch častí, ako prvú rozoberieme kalibráciu. Ak by sme chceli kalibráciu vykonávať na oboch osiach súčasne, museli by sme

program vykonávať na dvoch vláknach procesoru. Z tohto dôvodu budeme kalibrovat' každú os samostatne. Kalibrácia je spustená automaticky, hneď po spustení procesoru, ale môže byť vyvolaná aj prerušením z riadiaceho pinu. Spustenie kalibrácie je teda možné aj priamo z Matlabu.

Obrázok č. 14 Algoritmus kalibrácie



Počítanie krokov je nutné a to z dôvodu získania konštanty, ktorá je využitá v normálnom režime. Táto konštanta je použitá na násobenie hodnôt s A/D prevodníku pre výpočet počtu krokov, ktorý zodpovedá požadovanej polohe.

Obrázok č. 15 Prevedenie kalibrácie v jazyku C

```

PPG1_Disable(MyPWM1Ptr); /*Pulse signal off*/
Direction2=TRUE; /*Determine direction of
movement*/
PPG2_Enable(MyPWM2Ptr); /*Pulse signal on*/
EInt1_Enable(); /*Enable possibility of interrupt
of position sensor*/
while(Interupt_snimac1==0){}; /*Waiting for interrupt on
sensor*/
Interupt_snimac1=0; /*Cleaning variable for next use*/
EInt1_Disable(); /*Disabling interrupt of sensor*/
Direction2=FALSE; /*Changing direction of movement*/
CountPWM2=0 /*Variable for count of steps is
being cleaned*/
EInt2_Enable(); /*Enable possibility of interrupt
of position sensor*/
while(Interupt_snimac2==0){}; /*Waiting for interrupt on
sensor*/
Interupt_snimac2=0; /*Cleaning variable for next use*/
EInt2_Disable(); /*Disabling interrupt of sensor*/
Direction2=TRUE; /*Changing direction of movement
for final calibration*/
Const1=(unsigned int)CountPWM2>>1; /*Determines constant size by
dividing count of steps by 2*/
CountPWM2=0; /*Cleaning variable for next use*/
while(Const1>=CountPWM2){} /*Waits till number of steps is
equal to half of required steps*/
PPG2_Disable(MyPWM1Ptr); /*Disable pulsing signal for first
direction calibration is
finished*/
Const1=Const1>>11 /*Determining final value dividing
by 2048, which will be used as
multiplication constant in normal
mode*/
  
```

Samotný algoritmus normálneho režimu sa odohráva v prerušeníach, ktoré sú vyvolané dokončením periódy pulzujúceho signálu. Pulzujúci signál sa v normálnom režime nachádza na výstupe procesora stále. Najdôležitejšia je však zmena smeru pohybu, ktorá pri dosiahnutí požadovanej polohy motorov začne meniť smer na frekvencii pulzujúceho signálu. Týmto spôsobom zdanlivo zastavíme chod motorov. Zmenu smeru pohybu nastavujeme podľa požadovanej polohy náklonu plošiny. Ukážka prerušenia na konci periódy pulzujúceho signálu je na nasledujúcom obrázku. (*Obrázok č. 16*)

Obrázok č. 16 Prevedenie prerušenia v jazyku C

```
#pragma interrupt called
void PPG1_OnEnd(LDD_TUserData *UserDataPtr)
{
    if(Move1>=CountPWM1 & MODE==TRUE) {
        Direction1=TRUE;
        ++CountPWM1;
    }
    else if(MODE==TRUE) {
        Direction1=FALSE;
        --CountPWM1;
    }
    else
        ++CountPWM1;

    Dir1Pin_PutVal(Direction1);
}
```

/*If the number of required step is bigger then we increase count of steps*/
/*Setting direction*/
/*Increasing count of steps*/
/*In other case we decrease number of steps and set opposite direction*/
/*Setting direction*/
/*Decreasing number of steps*/
/*In case of calibration we only increase count of steps*/
/*Increasing count of steps*/
/*Puts value of direction to output pin*/

5 SPÄTNÁ VÄZBA

Spätná väzba je realizovaná pomocou WEB kamery. Kamera je pripojená do PC, kde spracovávame dáta v prostredí Simulink. Rýchlosť, s akou je kamera schopná snímať polohu guľičky, ovplyvňuje chovanie celého systému. Prakticky bude platiť, čím bude väčšia rýchlosť snímania kamery, tým bude celý systém bežať plynulejšie.

Riadenie je realizované s diskretným regulátorom, ktorý má periódu vzorkovania rovnakú ako má kamera. Kvôli tomuto faktoru nemá zmysel, aby bolo vzorkovanie riadiaceho signálu na riadení motorov rýchlejšie.

Ďalším dôležitým faktorom je presnosť, s akou určujeme polohu guľičky na plošine. To dokážeme ovplyvniť jedine rozlíšením kamery, ktoré však vplýva aj na rýchlosť akou polohu snímame.

Najlepším prevedením spätnej väzby by však bolo použitie odporovej platničky, ktorá by polohu určovala veľmi presne a rýchlejšie ako kamera.

5.1 Kamera

Kamera, ktorú používame má 120 snímok za sekundu s nastaveným rozlíšením 480x620. Pomocou novej kamery sme schopní zamerať polohu presnejšie a rýchlejšie, oproti staršej kamere s rýchlosťou 30 snímok za sekundu .

5.1.1 Uloženie kamery

Kamera je uložená v puzdre, ktoré bolo navrhnuté v programe SolidWorks®. Vytlačené bolo na 3D tlačiarňi. 3D model puzdra sa nachádza v prílohe (Príloha č.11).

5.2 Algoritmy detekcie polohy

Polohu môžeme určiť vďaka veľkému rozdielu vo farbách guľičky a plošiny. Guľička samotná má bielu farbu, ktorú reprezentujeme číslom 256 a čiernu farbu reprezentujeme číslom 1. Tieto čísla reprezentujú jas v čiernobielym zobrazení obrazu. Na základe týchto znalostí je celkom jednoduché rozlíšenie guľičky od plošiny. V ideálnom prípade by guľička mala priradenú hodnotu 256, čo sa nám však v skutočnosti nepodarí, ovplyvňovať to bude hlavne osvetlenie či samotná farba povrchu guľičky, či plošiny. Kamera sníma obraz vo farebnom režime, zmena na čiernobiely obraz je možná viacerými spôsobmi.

Na Obrázok č. 17 vidíme vstup z kamery. Obraz je reprezentovaný v troch dvojrozmerných maticiach, každá matica reprezentuje jednu farbu vo výslednom obraze. Taktiež môžeme povedať, že hodnoty v maticiach budú od 1 do 256, čo bude

vyjadrovať, koľko dotyčnej farby sa vo výslednom obraze nachádza. Bielu farbu bude vytvárať zmes farieb, červená (R), zelená (G) a modrá (B), s maximálnou hodnotou každého prvku (256). Naopak čiernu farbu bude predstavovať minimálna hodnota prvkov týchto matic (0). Najjednoduchším vytvorením čierneho-bieleho obrazu teda bude sčítanie hodnôt matic. Výsledná matica bude obsahovať príliš veľké hodnoty ($256+256+256$) a preto ich podelíme tromi, aby bol rozsah hodnôt správny.

S takto vytvoreným čierno-bielym obrazom môžeme pristúpiť k detekcii polohy guľičky. Keďže guľička má bielu farbu, musíme na obraze nájsť objekt s hodnotou jas, ktorá reprezentuje bielu farbu. Maticu obrazu budeme skenovať a hľadať objekt s nami požadovanou hodnotou jas. Pokiaľ nájdeme objekt s požadovanými parametrami, musíme ešte určiť jeho stred, od ktorého budeme počítat súradnice X a Y. Na detekciu polohy je použitý algoritmus dodávaný s pôvodnou verziou prípravku. Algoritmus je napísaný pomocou programovacieho jazyka C. Aby ho bolo možné použiť, potrebujeme takzvaný MEX file, ktorý slúži práve na volanie vlastných funkcií vytvorených v jazykoch C alebo C++.

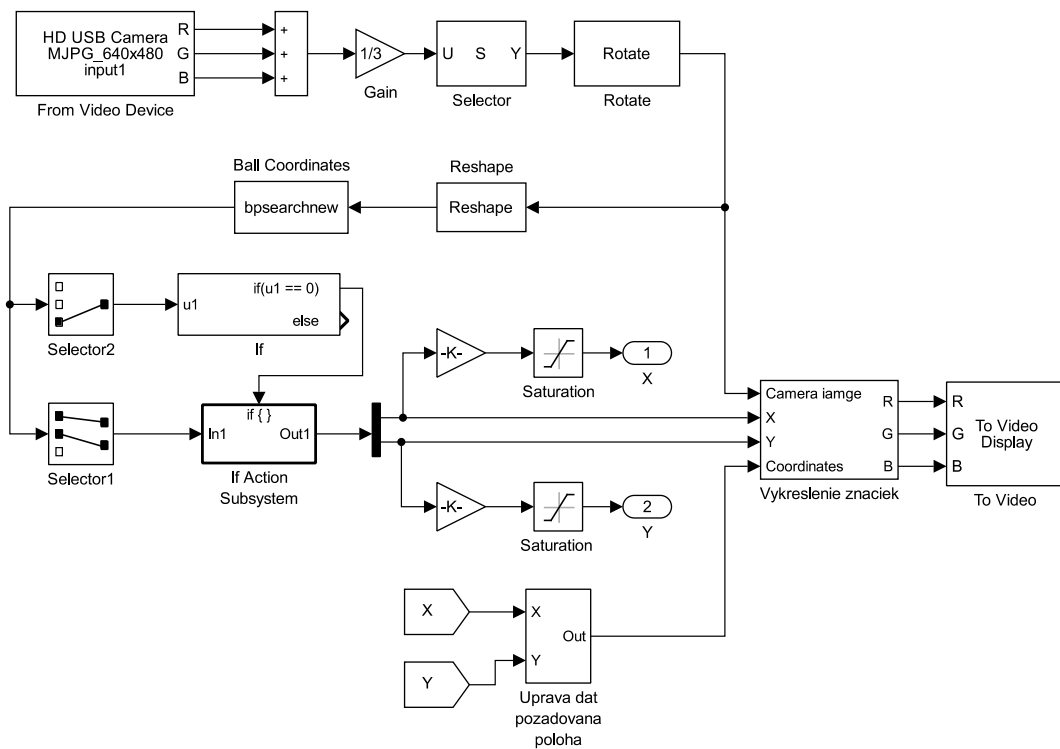
Volanie MEX file je možné aj v Simulinku, na *Obrázok č. 17*, bloček s názvom bpsearch. Vstupom do funkcie je jednorozmerná matica a preto je nutné transformovať pôvodnú maticu, na tento účel sa v schéme nachádza bloček Reshape. Výstupom z funkcie sú koordináty X a Y a servisný signál. Servisný signál môže nadobúdať tri hodnoty a to :

- 0 Guličku sa podarilo nájsť
- 1 Gulička nebola nájdená
- 2 Gulička je príliš veľká

Servisný signál využijeme práve v prípade straty polohy guľičky, ak sa guľička nenájde alebo bude príliš veľká na výstup nebudeme posielat žiadnu polohu. Na tento prípad slúži podmienkový bloček *if*. Bloček Selektor slúži na úpravu rozlíšenia kamery, na rozmer 480x480 pixlov.

Program je obohatený o zobrazovanie aktuálne nájdenej polohy na reálnom obraze z kamery a taktiež zobrazovanie požadovanej polohy, na ktorú sa má guľička vyregulovať. O vykresľovanie značiek aktuálnej a požadovanej polohy sa stará bloček *Vykreslenie značiek*, kód tohto bločku bude možné nájsť v prílohe (Príloha č.6).

Obrázok č. 17 Spracovanie obrazu v prostredí Simulink



6 PRENOS RIADIACEHO SIGNÁLU

6.1 Riadiaca karta MF 642

MF 642 je multifunkčná I/O karta, ktorá je navrhnutá pre pripojenie počítačov k signálom vyskytujúcim sa v reálnom svete. Na karte sa nachádza 8 kanálov s rýchlym 14 bitovým A/D prevodníkom, so simultánne vzorkovacím obvodom, 8 nezávislých 14 bitových D/A prevodníkov, 8 digitálnych vstupov, 8 digitálnych výstupov, 4 vstupy pre optické enkodéry s diferenciálnym alebo single ended zapojením a piatimi vstupmi pre časovače alebo čítače. Karta je navrhnutá pre štandardné zhromažďovanie dát a riadenie aplikácií. Optimalizovaná je na použitie s Real Time Toolbox pre Matlab® Simulink®. Karta plne podporuje 32 bitovú architektúru pre rýchlu odozvu. [9]

6.1.1 D/A prevodník

Jedná sa o súčiastku, vďaka ktorej sme schopní previesť digitálny signál na analógový. Prevodník má presnosť 14 bitov s rozsahom $\pm 10V$. V skutočnosti to znamená, že výstupná hodnota z D/A prevodníku má presnosť na tisícinu voltu.

6.2 Riadiaci signál

Jednoduché pripojenie riadiacej karty ku výstupu regulátoru bolo nedostačujúce. Keďže signál vystupujúci z riadiacej karty MF 624 nemá dostatočnú presnosť, je nutné použiť metódu, ktorou by bolo možné spresniť hodnoty na výstupe. Využijeme teda ďalšie digitálno-analógové prevodníky riadiacej karty.

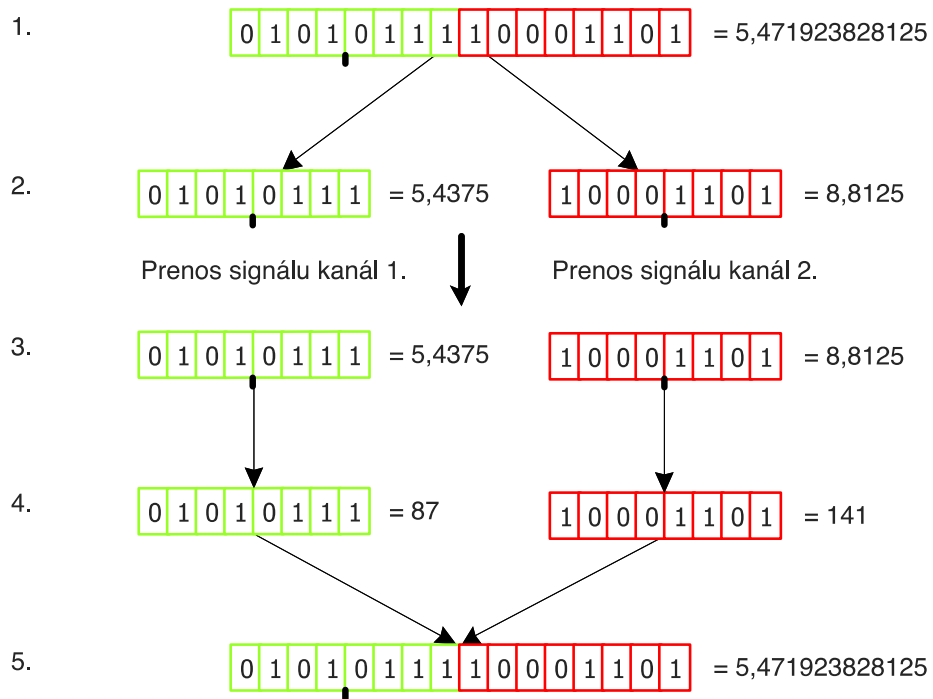
Signál v prostredí Matlab je väčšinou prenášaný pomocou dátového typu double. Tento typ je možné zmeniť na nami požadovaný. Zvolili sme dátový typ FixedPoint s dĺžkou slova 16 bitov. Následne delíme slovo na hornú polovicu 8 bitov a spodnú polovicu 8 bitov. Každéj časti je ešte potrebné priradiť správnu desatinnú čiarku, kvôli rozsahu a presnosti riadiacej karty. Pri 8 bitoch potrebujeme maximálne hodnotu 15. Desatinná čiarka sa teda bude nachádzať v polovici slova. Pri každom type je nutné ponechať znamienko, to znamená signed FixedPoint.

Riadiaca karta dokáže spracovať len dátový typ Double. Takže poslednou úpravou, ktorú musíme vykonať, bude prekonvertovanie dátového typu FixedPoint na typ Double. (Obrázok č. 19)

Na strane prijímania riadiaceho signálu (CompactRIO) musíme úkon zopakovať presne opačným postupom. Hodnotu z analógovo-digitálnych vstupov musíme prekonvertovať na typ FixedPoint o dĺžke slova 8 bitov, s desatinou čiarkou nepôsobiacou na hodnotu bajtu. Spojenie týchto častí musíme vykonať rešpektujúc

hornú a dolnú polovicu slova. Opätovným posunom desatinnej čiarky na správnu pozíciu, dostávame presnú hodnotu riadiaceho signálu. Čiarku posunieme za štvrtý bit slova od najdôležitejšej hodnoty (*Obrázok č. 20*). Grafické znázornenie spôsobu prenosu riadiaceho signálu je na *Obrázok č. 18*.

Obrázok č. 18 Spôsob prenosu riadiaceho signálu



1. Hodnota prekonvertovaná z typu Double na FixedPiont.
2. Rozdelenie na hornú polovicu bitov v zelenom a dolnú polovicu bitov v červenom. Odoslanie hodnoty pomocou digitálno-analógového prevodníku.
3. Načítanie signálu na analógovo digitálnom prevodníku.
4. Posun desatinnej čiarky za najmenej dôležitý bit.
5. Spojenie hornej a dolnej polovice do 16 bitového slova typu FixedPoint. Opätovné posunutie desatinnej čiarky na správnu pozíciu

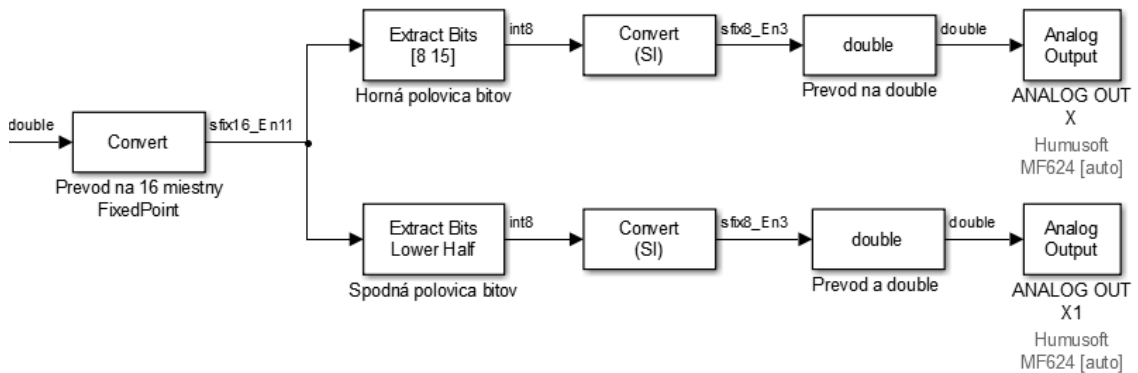
Vďaka tejto metóde prenosu sa zbavíme nepríjemného šumenia riadiaceho signálu a nebudeme pozorovať vibrovanie plošiny pri dosiahnutí požadovanej hodnoty.

Pre procesor upravujeme vstupnú hodnotu pomocou bitového posunu (*shiftovanie*), teda delením mocninami čísla 2 a pomocou masiek (*Obrázok č. 21*). Stranu, na ktorú sa má pohnúť krokový motor, určujeme na základe pozitívnej alebo negatívnej hodnoty od 0. Musíme teda odčítať polovicu plného 12 bitového dátového typu integer (2048) od hodnoty z A/D prevodníku (riadok 6 a 7).

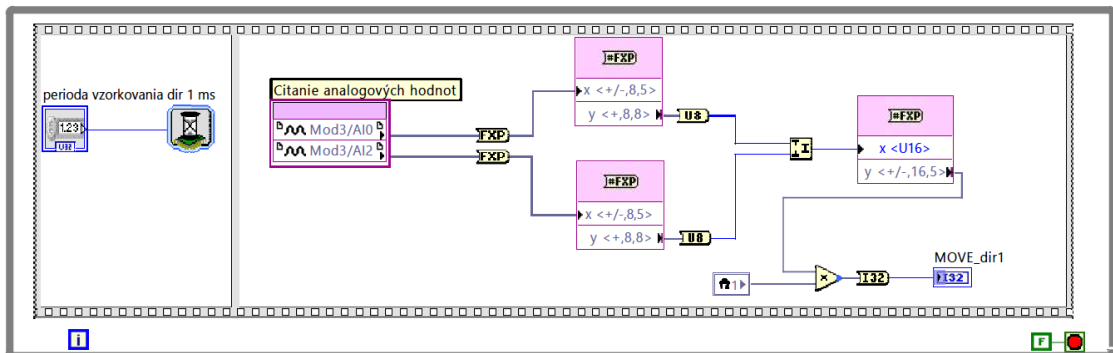
Následne pomocou preddefinovaných masiek odstránime dolnú polovicu 6 bitov (riadok 6) a hornú polovicu bitov (riadok 7). Po spojení týchto dvoch hodnôt získame požadovaný uhol náklonu. Posledným dôležitým úkonom je vynásobenie získanej hodnoty konštantou počtu krokov na jednotkovú zmenu hodnoty s A/D prevodníkov (riadok 8) a získame počet pulzov (krokov), ktorý docieli požadovaný náklon plošiny. Na riadkoch 9 až 11 sa rieši rovnaký proces, ale pre druhý smer. Načítanie hodnôt z A/D prevodníku je na riadkoch 2 až 5.

Kompletný projekt pre mikroprocesor je v prílohe (Príloha č.7)

Obrázok č. 19 Prepočet riadiaceho signálu Matlab Simulink



Obrázok č. 20 Prepočet riadiaceho signálu v prostredí Labview



Obrázok č. 21 Prepočet riadiaceho signálu v jazyku C

```
1  while (!Control1_GetVal()) {    /*Checking if there is requirement
                                   of calibration*/
2      Error = AD1_SelectSampleGroup(MyADCPtr, 0U);    /*Set
                                   created sample group*/
3      Error = AD1_StartSingleMeasurement(MyADCPtr);    /*Starts
                                   single measurement*/
4      while (!AD1_GetMeasurementCompleteStatus(MyADCPtr)) {}; /*
                                   Wait for conversion completeness */
5      Error = AD1_GetMeasuredValues(MyADCPtr, (LDD_TData *)
                                   MeasuredValues);    /*Read measured values */

                                   /*PROCESSING OF VALUES*/
6      RealValues[0]=((int)MeasuredValues[0]-2048) & MASK;
                                   /*Real Values of first direction*/
7      RealValues[0]=(((int)MeasuredValues[1]-2048) >> 6) & MASK2)
+ RealValues[0];
8      Move1=(RealValues[0]) * (-Const1);

9      RealValues[1]=((int)MeasuredValues[2]-2048) & MASK;
                                   /*Real Values of second direction*/
10     RealValues[1]=(((int)MeasuredValues[3]-2048) >> 6) & MASK2)
+ RealValues[1];
11     Move2=(RealValues[1]) * (-Const2);

12     ControlValues=MeasuredValues[4];    /*Control values of fifth
                                   ADC channel*/
13 }
```

7 MATEMATICKÝ MODEL

Na riešenie použijeme Euler-Lagrangovú metódu, ktorej zápis môžeme vidieť v nasledujúcej rovnici:

$$\frac{d}{dt} \left[\frac{\partial K}{\partial \dot{g}_j} \right] - \frac{\partial K}{\partial g_j} + \frac{\partial V}{\partial g_j} = \Psi_j \quad (10)$$

kde: K kinetická energia
 V potenciálna energia
 g_j popisuje i -ty smerový koordinát
 Ψ_j popisuje i -tu zloženú silu

Systém má dva stupne voľnosti. Prvé dva môžeme pripísať nakláňaniu plošiny, ďalšie zase pohybu guľičky na plošine.

Budeme teda uvažovať s koordinátami guľičky na plošine x_b, y_b a náklonom plošiny v uhloch α, β . Poloha koordinátou $x_b = y_b = 0$ sa nachádza v strede plošiny a uhol $\alpha = \beta = 0$ sa nachádza v rovine.

Na použitie Euler-Lagrangovej metódy si musíme najskôr vyjadriť kinetickú a potenciálnu energiu. Systém sa skladá z dvoch podsystemov a to guľičky a plošiny. Ako prvý si vyjadríme podsystem guľička.

7.1 Podsystem guľička

V tomto prípade sa kinetická energia bude skladať z rotačnej a translačnej energie guľičky:

$$K_b = K_{Transl} + K_{Rot} \quad (11)$$

$$K_{Transl} = \frac{1}{2} m v^2 = \frac{1}{2} m \dot{x}_b^2 + \frac{1}{2} m \dot{y}_b^2 = \frac{1}{2} m (\dot{x}_b^2 + \dot{y}_b^2) \quad (12)$$

$$K_{Rot} = \frac{1}{2} J_b \omega^2 = \frac{1}{2} m \omega_x^2 + \frac{1}{2} m \omega_y^2 = \frac{1}{2} J_b (\omega_x^2 + \omega_y^2) \quad (13)$$

$$K_b = \frac{1}{2} m (\dot{x}_b^2 + \dot{y}_b^2) + \frac{1}{2} J_b (\omega_x^2 + \omega_y^2) \quad (14)$$

kde: m hmotnosť guľičky
 J_b zotrvačný moment guľičky
 ω_x uhlová rýchlosť v osi x
 ω_y uhlová rýchlosť v osi y
 v rýchlosť

Keďže sa jedná o guľičku pohybujúcu sa po plošine a uvažujeme o stálom kontakte povrchu guľičky s plošinou, môžeme tvrdiť, že jej rýchlosť závisí od nasledujúcich vzťahov:

$$\dot{x}_b = r_b \omega_x \quad (15)$$

$$\dot{y}_b = r_b \omega_y \quad (16)$$

kde r_b polomer guľičky

Po dosadení vyššie uvedených vzťahov do rovnice kinetickej energie guľičky dostávame:

$$K_b = \frac{1}{2} \left[m(\dot{x}_b^2 + \dot{y}_b^2) + \frac{J_b}{r_b^2} (\dot{x}_b^2 + \dot{y}_b^2) \right] = \frac{1}{2} \left(m + \frac{J_b}{r_b^2} \right) (\dot{x}_b^2 + \dot{y}_b^2) \quad (17)$$

Potenciálna energia guľičky ku stredú náklonnej plošiny je vyjadrená nasledujúcim vzťahom:

$$V = mgh = mg(x_b \sin \alpha + y_b \sin \beta) \quad (18)$$

kde g gravitačné zrýchlenie

h výška od stredú náklonnej plošiny

7.2 Podsystem plošina

Pre kinematickú energiu plošiny uvažujeme o guľičke ako o bode a hmotnosti umiestnenej na koordinátoch x_b, y_b .

$$V_p = \frac{1}{2} (J_p + J_b) (\dot{\alpha}^2 + \dot{\beta}^2) + \frac{1}{2} m (x_b \dot{\alpha} + y_b \dot{\beta})^2 \quad (19)$$

7.3 Dosadenie do Euler-Lagrangovej metódy

Sčítame kineticke energie guľičky a plošiny:

$$K = K_b + K_p \quad (20)$$

$$K = \frac{1}{2} \left[\left(m + \frac{J_b}{r_b^2} \right) (\dot{x}_b^2 + \dot{y}_b^2) + (J_p + J_b) (\dot{\alpha}^2 + \dot{\beta}^2) + m (x_b \dot{\alpha} + y_b \dot{\beta})^2 \right] \quad (21)$$

Jednotlivé derivácie pre kineticke energiu:

$$\frac{\partial K}{\partial \dot{\alpha}} = (J_p + J_b) \dot{\alpha} + m x_b (x_b \dot{\alpha} + y_b \dot{\beta}) \quad (22)$$

$$\frac{\partial K}{\partial \dot{\beta}} = (J_p + J_b) \dot{\beta} + m y_b (x_b \dot{\alpha} + y_b \dot{\beta}) \quad (23)$$

$$\frac{\partial K}{\partial x_b} = \left(m + \frac{J_b}{r_b^2} \right) \dot{x}_b \quad (24)$$

$$\frac{\partial K}{\partial y_b} = \left(m + \frac{J_b}{r_b^2} \right) \dot{y}_b \quad (25)$$

$$\frac{d}{dt} \frac{\partial K}{\partial \dot{\alpha}} = (J_b + J_p + mx_b^2) \ddot{\alpha} + m(\ddot{\beta} x_b y_b + \dot{\beta}(\dot{x}_b y_b + x_b \dot{y}_b) + 2\dot{\alpha} \dot{x}_b y_b) \quad (26)$$

$$\frac{d}{dt} \frac{\partial K}{\partial \dot{\beta}} = (J_b + J_p + mx_b^2) \ddot{\beta} + m(\ddot{\alpha} x_b y_b + \dot{\alpha}(\dot{x}_b y_b + x_b \dot{y}_b) + 2\dot{\beta} \dot{x}_b y_b) \quad (27)$$

$$\frac{d}{dt} \frac{\partial K}{\partial \dot{x}_b} = \left(m + \frac{J_b}{r_b^2} \right) \dot{x}_b \quad (28)$$

$$\frac{d}{dt} \frac{\partial K}{\partial \dot{y}_b} = \left(m + \frac{J_b}{r_b^2} \right) \dot{y}_b \quad (29)$$

$$\frac{\partial K}{\partial \alpha} = 0 \quad (30)$$

$$\frac{\partial K}{\partial \beta} = 0 \quad (31)$$

$$\frac{\partial T}{\partial x_b} = m(\dot{\alpha} \dot{\beta} y_b + \dot{\alpha}^2 x_b) \quad (32)$$

$$\frac{\partial T}{\partial y_b} = m(\dot{\alpha} \dot{\beta} x_b + \dot{\alpha}^2 y_b) \quad (33)$$

Jednotlivé derivácie pre potenciálnu energiu:

$$\frac{\partial V}{\partial \alpha} = mgx_b \cos \alpha \quad (34)$$

$$\frac{\partial V}{\partial \beta} = mgy_b \cos \beta \quad (35)$$

$$\frac{\partial V}{\partial x_b} = mg \sin \alpha \quad (36)$$

$$\frac{\partial V}{\partial y_b} = mg \sin \beta \quad (37)$$

Je potreba si uvedomiť, že na guľičku nepôsobia žiadne ďalšie sily okrem gravitačnej. To však platí len v zjednodušenom modeli, v realite na guľičku pôsobia odporové sily, ako napríklad valivý odpor. Tejto problematike sa budeme venovať v kapitole 7.6. Na zmenu náklonu pôsobí krútiaci moment $(\tau_\alpha, \tau_\beta)$, ktorý musíme zahrnúť do rovníc.

S pomocou vzťahov (22-37) dosadených do rovnice (10), dostávame diferenciálne rovnice opisujúce systém:

$$x: \quad \left(m + \frac{J_b}{r_b^2}\right) \ddot{x}_b - m(\dot{\alpha}\dot{\beta}y_b + \dot{\alpha}^2x_b) + mg \sin \alpha = 0 \quad (38)$$

$$y: \quad \left(m + \frac{J_b}{r_b^2}\right) \ddot{y}_b - m(\dot{\alpha}\dot{\beta}x_b + \dot{\beta}^2y_b) + mg \sin \beta = 0 \quad (39)$$

$$\alpha: \quad (J_b + J_p + mx_b^2)\ddot{\alpha} + m(\ddot{\beta}x_by_b + \dot{\beta}(\dot{x}_by_b + x_b\dot{y}_b) + 2\dot{\alpha}\dot{x}_by_b) + mgx_b \cos \alpha = \tau_\alpha \quad (40)$$

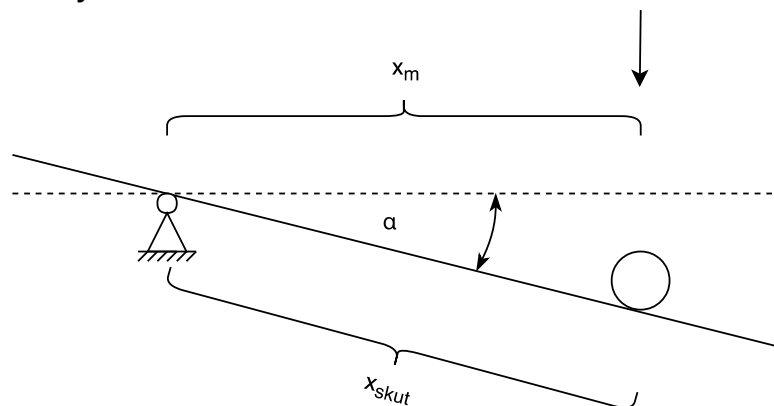
$$\beta: \quad (J_b + J_p + mx_b^2)\ddot{\beta} + m(\ddot{\alpha}x_by_b + \dot{\alpha}(\dot{x}_by_b + x_b\dot{y}_b) + 2\dot{\beta}\dot{x}_by_b) + mgy_b \cos \beta = \tau_\beta \quad (41)$$

7.4 Zjednodušenie matematického modelu

Rovnice (38 až 41) sú príliš komplikované na navrhovanie riadiaceho algoritmu, preto podstúpime ich zjednodušenie a linearizáciu.

Netreba zabúdať, že súradnicový systém sa vzťahuje k súradniciam priamo na plošine, po ktorej sa guľička pohybuje. Meranie jej polohy však bude prevedené priamo z vrchu. Túto chybu môžeme vyjadriť ako: $x_{skut} = x_m / \cos \alpha$. Zreteľná je z nasledujúceho obrázku:

Obrázok č. 22 Chyba snímania



Pri simulácii však s touto chybou nemusíme počítať, môžeme používať priamo koordináty zo systému. Na reálnom systéme túto chybu môžeme zanedbať, v konečnom dôsledku zmení len bod, ku ktorému bude guľička regulovaná. Podobná chyba nastáva aj pri realizácii spätnej väzby, kde skresľuje obraz kamery šošovka.

Uhly α a β sa na prípravku pohybujú maximálne vo veľkosti -11° až 11° z čoho môžeme písať $|\alpha| \ll 1$ a $|\beta| \ll 1$ v radiánoch. Funkcie $\sin \alpha$ a $\sin \beta$ je teda možné linearizovať ako $\sin \alpha \approx \alpha$ a $\sin \beta \approx \beta$. Derivácia uhlov α a β popisuje uhlovú rýchlosť a s ohľadom na maximálnu rýchlosť motorov môžeme písať $|\dot{\alpha}| \ll 1 \Rightarrow \dot{\alpha} \approx 0$ a $|\dot{\beta}| \ll 1 \Rightarrow \dot{\beta} \approx 0$. Prostredný člen diferenciálnej rovnice (38 a 39) $m(\dot{\alpha}\dot{\beta}y_b + \dot{\alpha}^2x_b)$ teda môžeme zanedbať.

Dosadením za zotrvačný moment guľičky (J_b), pre plnú guľičku platí:

$$J_b = \frac{2}{5}mr_b^2 \quad (42)$$

alebo pre obal guľičky:

$$J_b = \frac{2}{3}mr_b^2 \quad (43)$$

Zotrvačný moment plošiny (J_p) môžeme zanedbať kvôli jeho veľkosti. Rovnicu na jeho výpočet môžeme vidieť nižšie:

$$J_p = \frac{1}{6}ma^2 \quad (44)$$

kde a rozmer plošiny v [m]
 m hmotnosť plošiny

Po linearizácii a zjednodušení môžeme model popísať nasledovne:

$$\ddot{x} = -\frac{3}{5}g \quad (45)$$

$$\ddot{y} = -\frac{3}{5}g \quad (46)$$

Prenos systému bude mať tvar:

$$F_x(p) = -\frac{K_s}{p^2} \quad (47)$$

$$F_y(p) = -\frac{K_s}{p^2} \quad (48)$$

kde K je parameter určujúci typ guľičky, pre dutú guľičku $K_s = \frac{3}{5}g$ a plnú $K_s = \frac{5}{7}g$

7.5 Simulácia pohybu motorov

Motory budeme simulovať zjednodušene a to pomocou prenosovej funkcie prvého alebo druhého rádu. Prechodové funkcie majú tvar [10]:

$$F_M(p) = \frac{K_M}{Tp + 1} \quad \text{alebo} \quad F_M(p) = \frac{K_M}{(T_1p + 1)(T_2p + 1)} \quad (49)$$

7.6 Simulácia valivého trenia

Valivé trenie sme schopní vypočítať ako [11]:

$$F_t = \xi \frac{F_N}{R} \quad (50)$$

kde: F_N kolmá tlaková sila medzi telesami, pre nás: $F_N = mg$
 ξ súčiniteľ valivého trenia
 R polomer guličky

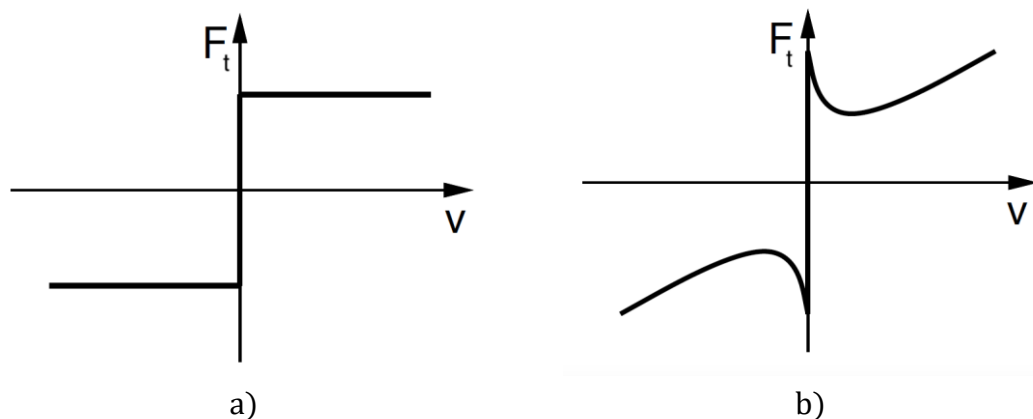
Do matematického modelu by ho bolo možné zakomponovať dosadením do rovníc (38 a 39) za zloženú silu Ψ_j . Keďže valivé trenie vždy pôsobí proti smeru pohybu telesa, vzťah na jeho výpočet bude vyzeráť nasledovne:

$$F_t = \xi \frac{mg}{R} \operatorname{sgn}(v) \quad (51)$$

kde: v rýchlosť valiaceho sa telesa

Tento výpočet má tvar Cuombovho trenia (*Obrázok č. 23 a*). Existujú však aj ďalšie aproximácie trenia ako (*Obrázok č. 23 b*) [12]. V našom prípade však kvôli jednoduchosti použijeme najjednoduchší tvar trenia. Valivé trenie je spomenuté najmä kvôli skutočnosti, že na reálnom modeli vytvorí nelinearitu, ktorá spôsobuje nevyregulovanie guličky na požadovanú polohu. Odmeranie nelinearity pre každú guličku je veľmi komplikované.

Obrázok č. 23 Znáznornenie valivého trenia [12]



Máme dve možnosti ako nelinaritu nasimulovať. A to použitím funkcie *Deadzone*, pred vstupom do matematického modelu alebo použitím rovnice (51) naviazanej na rýchlosť v matematickom modeli (Obrázok č. 26).

7.7 Simulovanie matematického modelu v Matlabe

Ako bolo spomínané dynamiku servomotorov budeme simulovať prechodovou funkciou prvého alebo druhého rádu. Časovú konštantu musíme odvodiť zo zvolenej rýchlosti krokových motorov. Perióda pulzujúceho signálu má hodnotu $125\mu\text{s}$ a predstavuje jeden krok motora. Tento krok má samozrejme vlastnú prechodovú charakteristiku. Jej priebeh však s ohľadom na jej dĺžku trvania zanedbáme. Presun zo skalibrovanej do krajnej polohy je rozdelený na 2048 polôh. Hodnota konštanty, ktorou násobíme hodnotu s A/D prevodníku je 10. S týmito hodnotami sme schopní vypočítať čas, za aký krokový motor dosiahne požadovanú hodnotu. A teda:

$$t_m = P_{poloha} K_{prev} T_{krok} = 2948 \cdot 10 \cdot 125 \cdot 10^{-6} = 2,56\text{s} \quad (52)$$

Časovú konštantu prechodovej funkcie prvého rádu teda vypočítame ako:

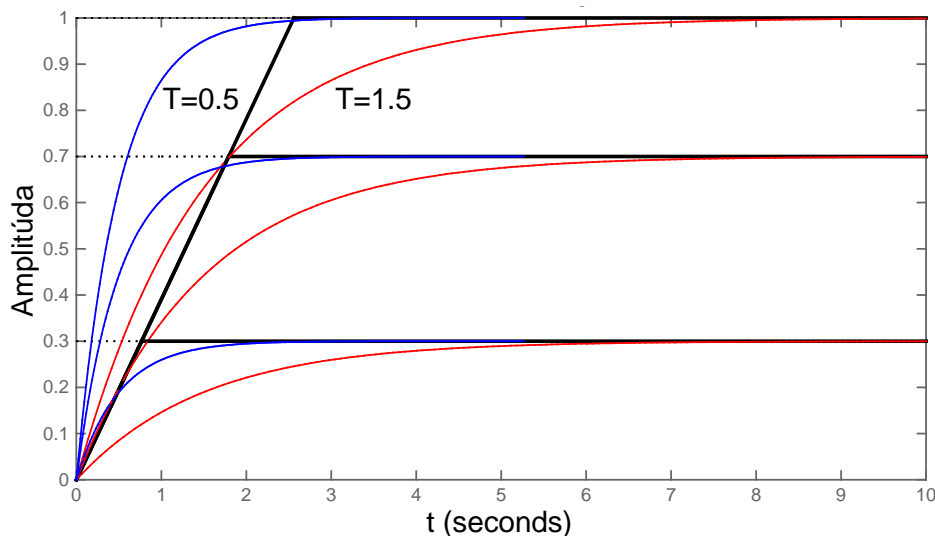
$$Y(p) = \frac{1}{p(Tp + 1)} \Rightarrow y(t) = 1 - e^{-\frac{1}{T}t} \quad (53)$$

Bežne sa časová konštanta určuje v 68% prechodu, hodnotu $y(t)$ teda zvolíme 0.68. Matematickými úpravami vyjadríme premennú T a dostávame:

$$T = \frac{1}{\ln(0,32)} = 1,53\text{s} \quad (54)$$

Na ďalšom obrázku môžeme vidieť prechodovú charakteristiku krokových motorov s aproximáciou pomocou prechodovej funkcie prvého rádu a skutočné približné chovanie krokových motorov (vyznačené čiernou čiarou). Do matematického modelu dosadíme agresívnejšiu verziu s časovou konštantou $T=0,5\text{s}$.

Obrázok č. 24 Skoková charakteristika krokových motorov

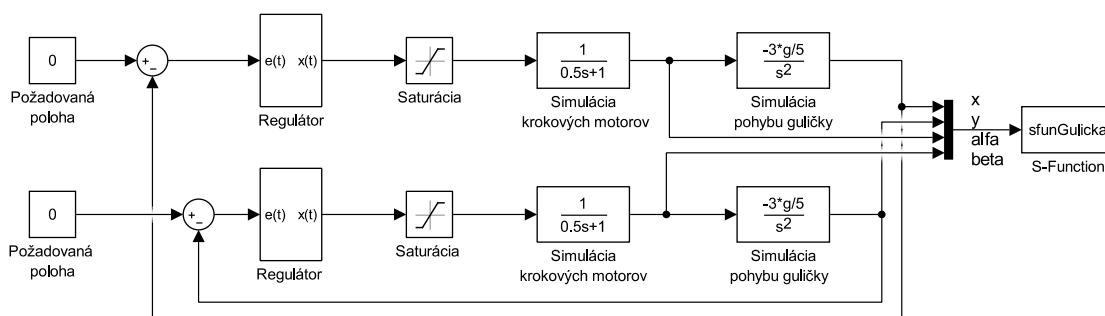


Na simuláciu môžeme použiť Simulink, v ktorom popíšeme matematický model vložením *Transfer function* alebo *priamym/paralelným/sériovým* typom programovania.

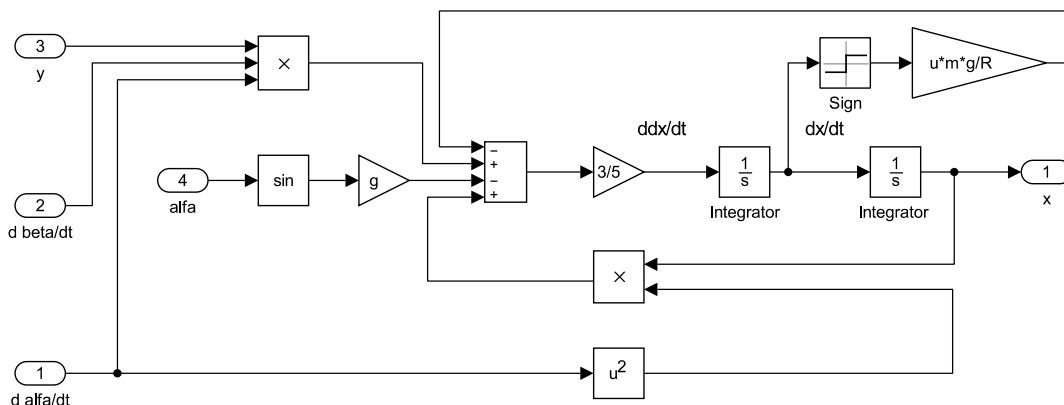
Pre grafické vykreslenie použijeme *S-function*, ktorá je modifikovaná tak aby vykresľovala guľičku v x, y, z súradniciach. Kód tejto funkcie je priložený ako príloha (Príloha č.8). Vstupom do funkcie sú uhly natočenia v osi a súradnica, na ktorej sa nachádza guľička. Funkcia je ošetrená proti priveľkým hodnotám súradníc či uhlu a to zablokovaním do krajnej polohy, pokiaľ sa hodnota nevráti do povolených hraníc.

Regulačná schéma lineárneho matematického modelu je na obrázku (Obrázok č. 25) Nelineárny model budeme simulovať podobne, s rozdielom že za *Transfer function* predstavujúcu pohyb guľičky, dosadíme schému z obrázka (Obrázok č. 26).

Obrázok č. 25 Simulácia matematického modelu



Obrázok č. 26 Simulácia nelineárneho matematického modelu



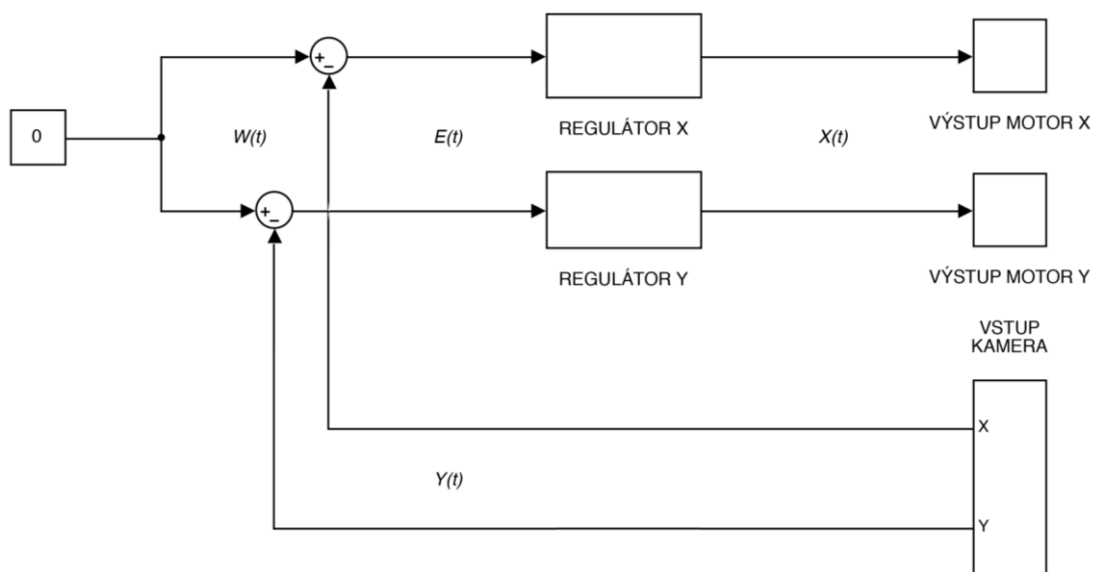
Je dôležité si uvedomiť, že na reálnom systéme sa budú nelinearity prejavovať. Na ich vykompenzovanie bude vhodné použiť funkciu „*Lookup table*“ s vhodne zvolenými parametrami.

Simulácia lineárneho a nelineárneho modelu sa nachádza v prílohe (Príloha č.9).

8 RIADIACI ALGORITMUS

V tejto kapitole sa budeme venovať návrhu riadenia systému. Keďže sa snažíme riadiť guľičku na plošine musíme si uvedomiť, že riadime dynamický systém a riadenie sa týka dvoch smerov. Nemôžeme teda použiť len jeden regulátor, riadiť budeme obidva smery, ktorými sa guľička môže hýbať. Úpravou základnej spätno-väzobnej regulačnej schémy, dostávame schému uvedenú na *Obrázok č. 27*.

Obrázok č. 27 Plošná regulácia



Na regulovanie systému sme zvolili PID regulátor. Jeho rovnica v časovej oblasti je [13]:

$$x(t) = K \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad (55)$$

kde: K zosilnenie regulátoru
 $e(t)$ regulačná odchýlka
 $x(t)$ akčný zásah
 T_i integračná zložka
 T_d derivačná zložka

Použitím Laplaceovej transformácie, je možné vyjadriť tvar PID regulátoru takto:

$$F_R(p) = \frac{X(p)}{E(p)} = K \left(1 + \frac{1}{T_i p} + T_d p \right) \quad (56)$$

Keďže takýto regulátor nie je fyzikálne realizovateľný, je nutné pridať zotrvačný člen s časovou konštantou ε . Reálny PID regulátor ma tvar:

$$F_R(p) = K \left(1 + \frac{1}{T_i p} + \frac{T_d p}{\varepsilon p + 1} \right) = \frac{K_R (T_1 p + 1)(T_2 p + 1)}{p(\varepsilon p + 1)} \quad (57)$$

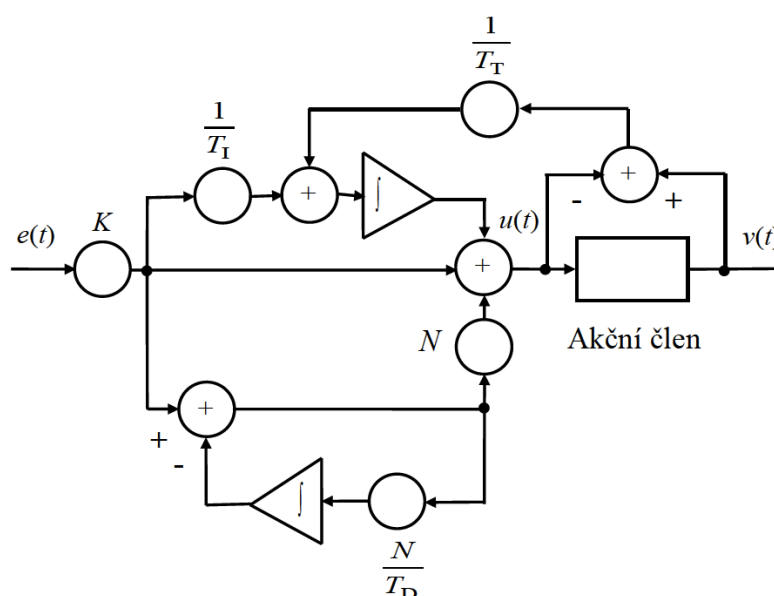
Vyššie uvedená rovnica býva často modifikovaná a to hlavne v derivačnej časti. Zmena časových konštánt T_d a ε má totižto vplyv aj na amplitúdu derivačnej zložky regulátoru. Relatívne veľký vplyv derivačnej zložky môže rozkmytávať za prítomnosti šumu regulačný obvod, malá hodnota nastavenia zase môže zväčšovať kmitanie v regulačnom obvode z dôvodu zmenšenia fázovej bezpečnosti. Z tohto dôvodu sa používa väčšia časová konštanta ε , ktorá tu pôsobí ako filter, zvolenie príliš veľkej hodnoty konštanty môže spôsobovať nestabilitu. V praxi používa každý výrobca regulátorov svoju vlastnú upravenú formu regulátora. Najčastejšie sa môžeme stretnúť s takýmto zápisom [13]:

$$F_R(p) = K \left(1 + \frac{1}{T_i p} + \frac{N p}{p + \frac{N}{T_d}} \right) \quad (58)$$

Pričom nahradzame časovú konštantu ε výrazom T_d/N a až po úprave dostávame vyššie uvedenú rovnicu. Konštanta N sa volí v rozmedzí 3 až 20. Fyzikálny význam tejto konštanty spočíva v tom, že N/T_d je frekvenciou zlomu, v ktorom sa amplitúdová charakteristika láme z $+20$ dB/dekádu na 0 dB/dekádu a vyššie frekvencie za týmto zlomom už nie sú zosilňované [13].

Pre výpočet pomocou počítača je nutné previesť zápis regulátoru do schematickeho tvaru:

Obrázok č. 28 Schematické prevedenie PID regulátoru [13]



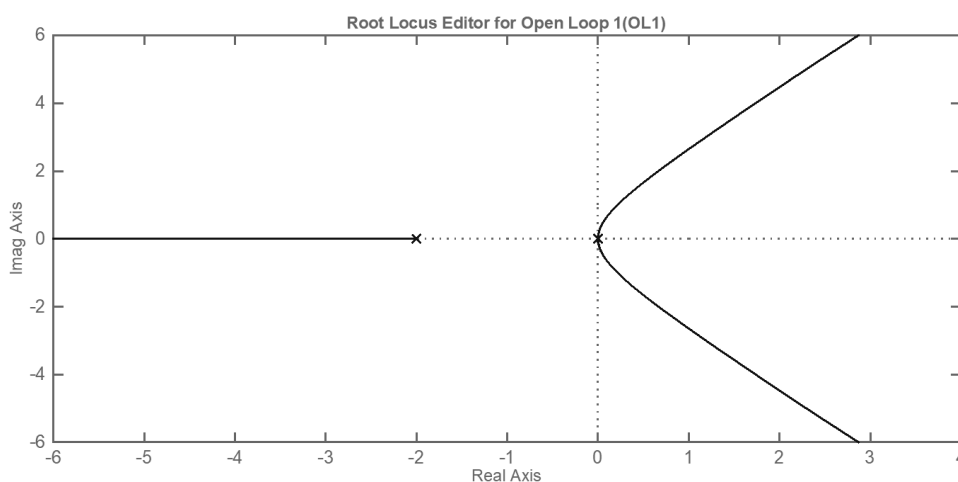
Problém nastáva ak hodnota na výstupe integračnej zložky prekročí hodnotu, ktorú je akčný člen schopný spracovať. Hodnota na výstupe regulátora pritom môže ďalej

rásť. Klesať začne až keď sa obráti znamienko odchýlky $e(t)$. To spôsobí časové oneskorenie, ktoré pôsobí kým sa prebytočná hodnota neodintegruje. Tento jav sa označuje ako prebudenie (*windup*). Regulátor na schéme je opatrený ochranou proti prebudeniu (*antiwindup*), ktorá je riešená privedením zápornej diferencie medzi akčným zásahom a skutočnou hodnotou akčného členu cez časovú konštantu T_T na vstup integračného členu regulátoru. Derivačná zložka je opatrená proti zosilňovaniu šumu vo vyšších frekvenciách vďaka filtračnej konštante N . [13]

8.1 Návrh regulátoru pre matematický model

Z koreňového hodografu regulovanej sústavy (Obrázok č. 29) je zrejmé, že neexistuje žiadne zosilnenie $K_R > 0$, ktoré by dostalo všetky vetvy hodografu na ľavú stranu imaginárnej polroviny. Jediným riešením je vloženie nuly, tú obsahuje regulátor typu PD alebo PID. Regulátor typu PID obsahuje integračnú zložku, ktorá by pridala astatizmus do sústavy, čo je nežiaduce. Navrhovať teda budeme regulátor typu PD.

Obrázok č. 29 Koreňový hodograf regulovanej sústavy



Na návrh regulátora použijeme metódu štandardného tvaru frekvenčnej charakteristiky. Ako prvé si vyjadríme amplitúdovú charakteristiku v dB a fázu.

$$F_0(p) = F_R(p)F_S(p) = \frac{K_R K_S (T_d p + 1)}{p^2 (0,5p + 1) \left(\frac{T_d}{N} p + 1\right)} \quad (59)$$

$$\varphi_0(\omega) = \arctan(T_d \omega) - \pi - \arctan(0,5\omega) - \arctan\left(\frac{T_d \omega}{N}\right) \quad (60)$$

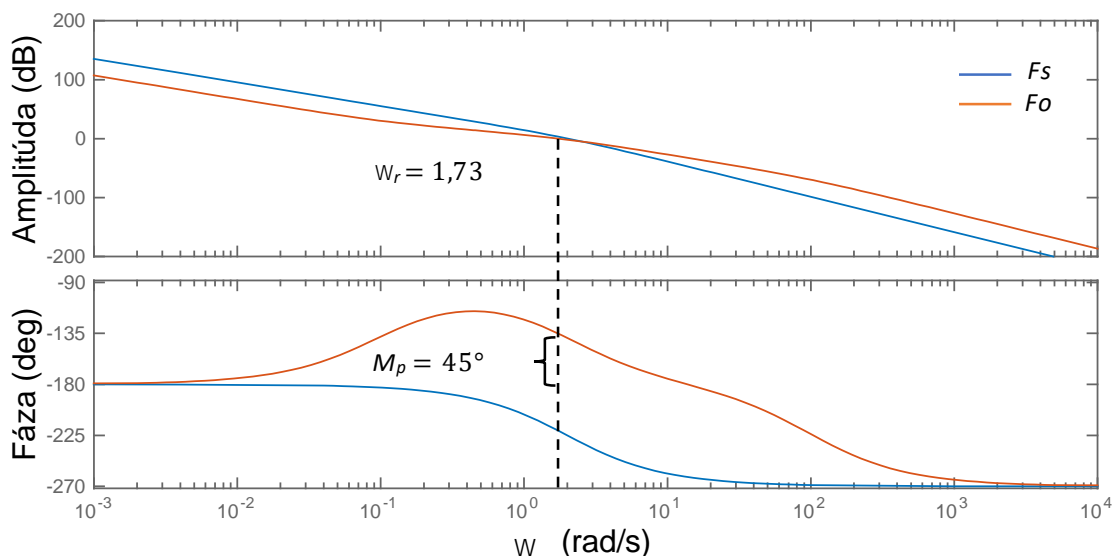
Nakreslíme frekvenčnú charakteristiku (Obrázok č. 30). Aby bol systém stabilný, je nutné aby amplitúdová charakteristika pretínala os 0dB v sklone -20dB/dek, to sme schopní docíliť jedine vložení deriváčnej zložky. Požiadavkou na regulátor je čo

najvyššia hodnota frekvencie rezu ω_r – bod, v ktorom amplitúdová charakteristika pretne os 0dB a čo najväčšia hodnota fázovej bezpečnosti M_p – rozdiel: $M_p = -\pi + \varphi_0(\omega_r)$ [10]. Zvolíme kompromis, znížime hodnotu fázovej bezpečnosti na $M_p = 45^\circ$, dovolíme teda regulátoru prekmit , aby sme dosiahli čo najkratšiu dobu regulačného deja.

Navrhnutý regulátor má hodnoty:

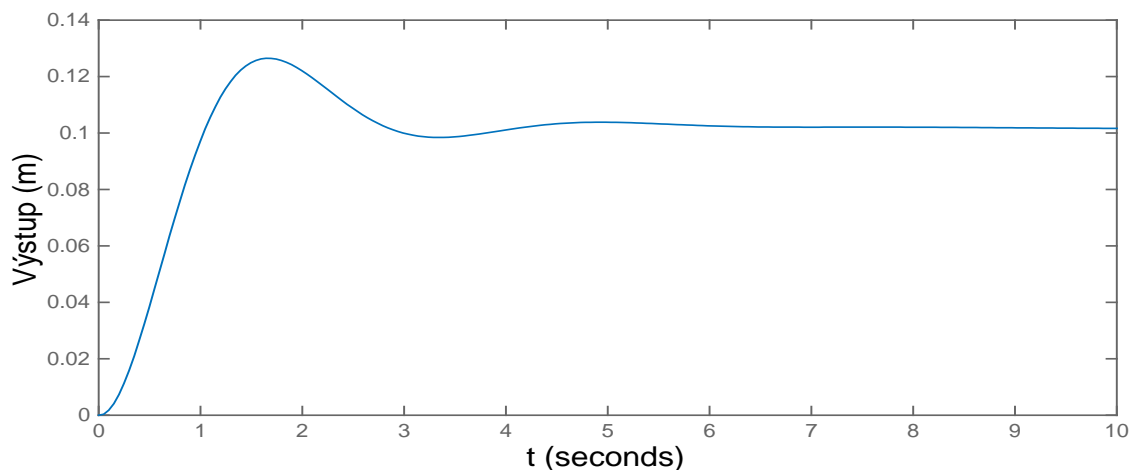
$$F_R = \frac{0,0394(10p + 1)}{(0.01p + 1)} \quad (61)$$

Obrázok č. 30 Frekvenčná charakteristika



S takto navrhnutým regulátorom má systém nasledovnú odozvu na skokovú zmenu:

Obrázok č. 31 Odozva systému na skokovú zmenu

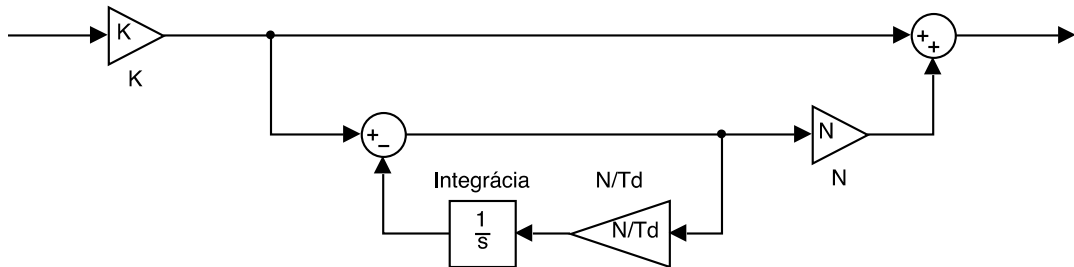


8.2 Spojitý regulátor

Rovnica (62) pre reálny regulátor PD a jeho schematická realizácia pomocou spätnej väzby s filtračnou zložkou (Obrázok č. 32).

$$F_R(p) = K \left(1 + \frac{T_d p}{\varepsilon p + 1} \right) = \frac{K_R (T_d p + 1)}{\left(\frac{T_d}{N} p + 1 \right)} \quad (62)$$

Obrázok č. 32 Prevedenie spojitého PD regulátora



8.3 Diskrétny regulátor

Výstup zo spätnej väzby je diskrétny a preto je regulátor potrebné prerobiť do diskrétneho tvaru. Diskrétny ekvivalent PID regulátoru môžeme vytvoriť náhradou, obdĺžnikmi z prava alebo z ľava či použiť lichobežníkovú náhradu. Pre náhradu obdĺžnikmi z ľava má regulátor tvar [13]:

$$x(k) = K \left[e(k) + \frac{T_v}{T_i} \sum_{i=0}^k e(k) + \frac{T_d}{T_v} (e(k) - e(k-1)) \right] \quad (63)$$

Kde T_d je vzorkovacia frekvencia. Pomocou Z transformácie dostaneme tvar:

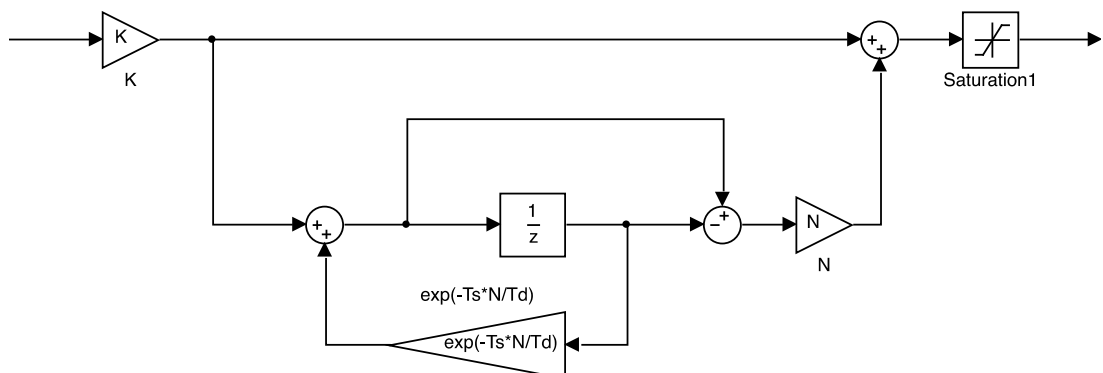
$$x(k) = \frac{X(z)}{E(z)} = K \left(1 + \frac{T_z}{T_i} \frac{1}{z-1} + \frac{T_d}{T_v} \frac{z-1}{z} \right) \quad (64)$$

Po dosadení za konštantu ε a úprave:

$$x(k) = \frac{X(z)}{E(z)} = K \left(1 + \frac{T_z}{T_i} \frac{1}{z-1} + N \frac{z-1}{z - e^{-\frac{T_v N}{T_d}}} \right) \quad (65)$$

Schéma použitého diskrétneho PD regulátora je na Obrázok č. 33.

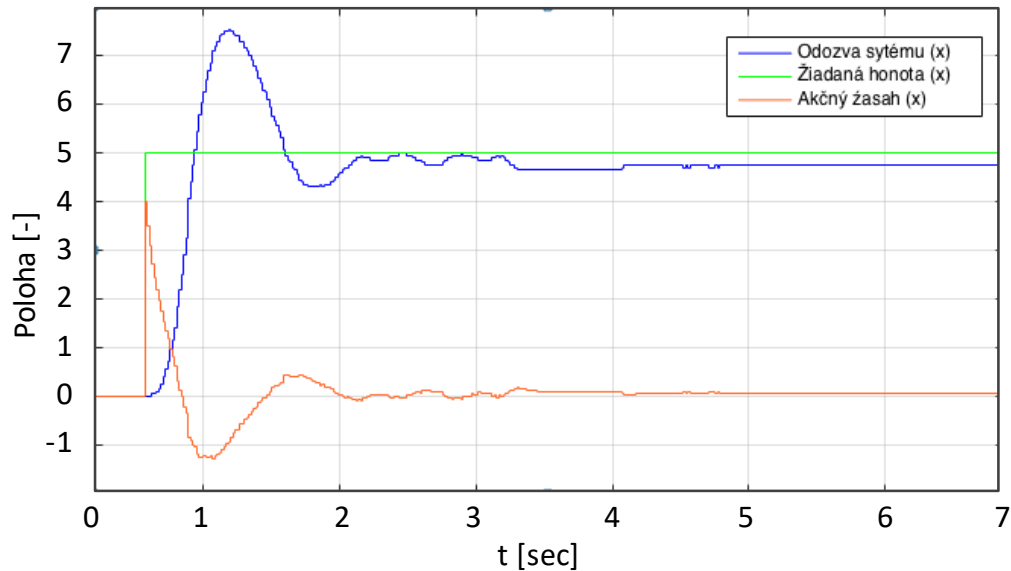
Obrázok č. 33 Prevedenie diskrétneho PD regulátoru



8.4 Odozva systému na regulátor

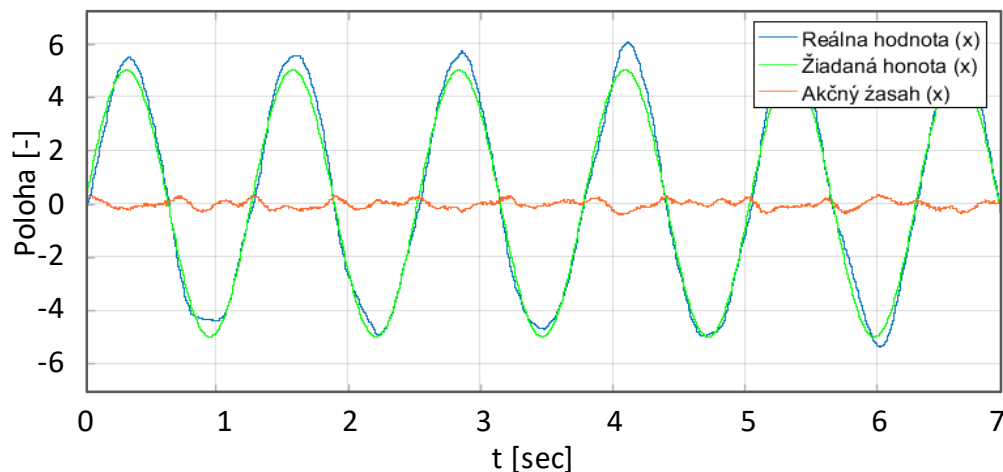
Regulátor použitý na reálnej sústave zo schémy na Obrázok č. 33, má parametre $T_d = 0,08, N = 3, K_R = 0,25$ a $T_s = 1/120$.

Obrázok č. 34 Odozva systému na skokovú zmenu



Odozva systému na sledovanie trajektórie v tomto prípade kružnice je na obrázku nižšie.

Obrázok č. 35 Odozva systému na sledovanie trajektórie



Ako môžeme vidieť na Obrázok č. 34, regulátor nedosiahne úplne presne požadovanú polohu, to je dané nelinearitou spôsobenou stykom guličky a plošiny (gulička nie je dokonale guľatá).

Kompletný algoritmus použitý na riadenie je v prílohe (Príloha č.10).

9 LABORATÓRNA ÚLOHA

Zadanie:

1. Simulujte matematický model v prostredí Matlab-Simulink a pokúste sa naňho navrhnuť regulátor pomocou metódy štandardného tvaru frekvenčnej charakteristiky.
2. Zoznámte sa so zapojením prípravku, pokúste sa rozpohybovať krokové motory a spustiť detekciu polohy guličky.
3. Využite znalosti o systéme pri nastavovaní regulátoru na prípravku.

Popis sústavy:

Sústava je tvorená náklonnou rovinou ovládanou krokovými motormi v dvoch osiach. Na rovine sa nachádza gulička kontrastnej farby detekovaná kamerou.



Matematický popis systému:

Systém v jednej osi je možné popísať nasledujúcou diferenciálnou rovnicou:

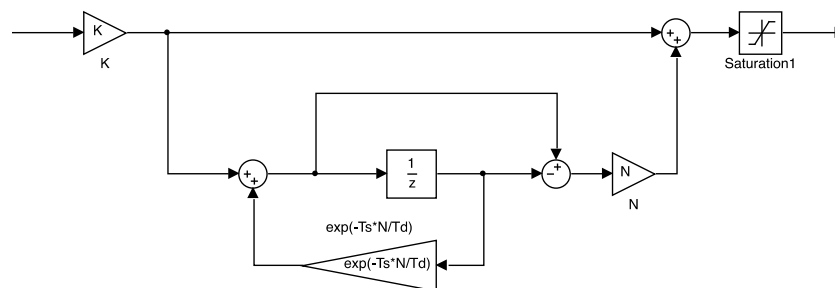
$$\left(m + \frac{J_b}{r_b^2}\right) \ddot{x}_b - m(\dot{\alpha}\dot{\beta}y_b + \dot{\alpha}^2 x_b) + mg \sin \alpha = 0$$

kde:	m	hmotnosť guličky
	J_b	moment zotrvačnosti guličky
	r_b	polomer guličky
	α	uhol náklonu v osi x
	β	uhol náklonu v osi y
	x_b	súradnica v osi x
	y_b	súradnica v osi y

Postup:

Príklad 1)

- Linearizujte rovnicu popisujúcu systém s nasledujúcimi predpokladmi v [rad]:
 - $|\alpha| \ll 1; |\beta| \ll 1 \Rightarrow \sin \alpha \doteq \alpha; \sin \beta \doteq \beta$
 - $|\dot{\alpha}| \ll 1; |\dot{\beta}| \ll 1 \Rightarrow \dot{\alpha}\dot{\beta} \doteq 0; \alpha^2 \doteq 0; \dot{\beta}^2 \doteq 0$
- Nasimulujte linearizovaný model v prostredí Matlab-Simulink. Na simuláciu krokových motorov použite prechodovú funkciu prvého rádu s časovou konštantou $T = 0.5s$.
- Vytvorte regulačný obvod s pripojením grafického rozhrania „*sfunGulicka.m*“. Vstupom do *S-function* je 4 rozmerný signál v rade za sebou: x, y, α, β . Regulátor použite v nasledujúcom formáte:



- Pomocou metódy frekvenčných charakteristík navrhnete hodnoty parametrov regulátora.

Príklad 2)

- Pomocou priloženého súboru „*regulacia_gulicka.m*“ sa pokúste spustiť krokové motory a detekciu polohy guľičky.
- Vytvorte regulačnú schému, v ktorej bude kamera spätnou väzbou a vstup do systému bude blok ovládania motorov.

Príklad 3)

- Zo znalostí z (*Príkladu 1*) sa pokúste nastaviť parametre regulátora na reálnom systéme. Vytvorte ľubovoľnú trajektóriu a otestujte regulátor.
- Pokúste sa vymenovať hlavné nelinearity systému, poprípade navrhnuť krivku pomocou „*Lookup table*“, ktorá ich bude kompenzovať.

ZÁVER

Práca je venovaná problematike regulovania guličky na náklonnej rovine. V prvej časti sa nám podarilo odladiť algoritmus pre ovládače krokových motorov, ktorý spracováva analógový signál a vysiela digitálny pulzujúci signál na riadenie krokových motorov. Následne sa nám podarilo algoritmus aplikovať do mikroprocesoru od firmy Freescale a nahradiť tak platformu CompactRIO. Pre procesor bola vytvorená doska plošných spojov s úpravou analógového vstupu. Navrhli sme prenos riadiaceho signálu pomocou analógového signálu a kvôli presnosti sme využili dva kanály pre každý smer. Ďalšou úlohou bolo vytvorenie spätnej väzby pomocou spracovania video obrazu. Nahradili sme staršiu kameru novou a dosiahli tak zrýchlenie celého systému. Upravili sme algoritmus detekcie polohy guličky na väčšie rozlíšenie a ošetrili stratu guličky z obrazu. Pre kameru bolo navrhnuté nové puzdro vytlačené na 3D tlačiarňi. Upravená bola taktiež kabeláž celého prípravku, kde na vstupoch zo snímačov krajnej polohy dochádzalo k presluchom.

Predstavili sme matematický model systému, ktorý sme linearizovali a odsimulovali pomocou programu Matlab. Pre matematický model sme pomocou metódy frekvenčných charakteristík navrhli regulátor. Na základe znalostí z návrhu sme odladili riadiaci algoritmus na reálnom systéme. Podarilo sa nám sledovať trajektóriu a dosahovať požadovanú polohu v relatívne krátkom čase, s čo najnižšou ustálenou odchýlkou. V prílohe (Príloha č.6) sa nachádza skript v programe Matlab-Simulink, ktorý obsahuje algoritmus prenosu signálu a taktiež algoritmus detekcie polohy guličky spolu s ovládaním kalibrácie či vypnutím/zapnutím (*enable*) krokových motorov.

Bola navrhnutá laboratórna úloha na prípravok, v ktorej by si študenti zopakovali návrh regulátora pomocou metódy štandardného tvaru frekvenčnej charakteristiky a neskôršie si svoj návrh skúsili otestovať a odladiť na prípravku.

Pri riadení pohybu guličky predstavuje najväčší problém nelinearita vytvorená v bode styku guličky a plošiny, kde aj najmenšia čiastočka prachu spôsobí zmenu smeru pohybu guličky (pozorovateľné hlavne na ľahkej dutej guličke). Túto nelinearitu sa môžeme pokúsiť popísať valivým trením, v skutočnosti sú to však nerovnosti na plošine a guľatosti guličky. Nelinearitu dokážeme v určitej miere kompenzovať, ale nepodarí sa nám ju odstrániť.

Existuje veľa spôsobov ako tento prípravok vylepšiť, za spomenutie stojí ovládanie rýchlosti krokových motorov priamo z počítača a prisvetlenie pre kameru, navrhnutá doska plošných spojov je na to pripravená. Pri práci na tejto úlohe som si prehĺbil znalosti v oblasti riadenia a regulácie, naučil sa používať software ako LabView, CodeWarrior, Solidworks, Eagle či Realtime-Toolbox v Simulinku.

LITERATÚRA

1. **Humusoft.** <http://www.humusoft.cz>. [Online] [Dátum: 01. 01 2018.] <http://www.humusoft.cz/models/ce151/>.
2. **Hrabovcová, V., Janoušek, L., Rafajdus, P., Ličko, M.** *Moderné elektrické stroje*. Žilina : EDIS, 2001. 80-7100-809-5.
3. **Collins, D.** [motioncontroltips.com](http://www.motioncontroltips.com). *What stepper motor type is best for high torque?* [Online] 06. 05 2017. [Dátum: 20. 12 2017.] <http://www.motioncontroltips.com/what-stepper-motor-type-is-best-for-high-torque/>.
4. **Superior electric .** *INSTRUCTION MANUAL*. s.l. : Danaher Motion, 2004.
5. **Leadshine Technology Company Limited.** *User's Manual For M542 V2.0 High Performance Microstepping Driver*. Shenzhen : Leadshine Technology Company Limited, 2008.
6. **National Instruments .** *User's manual for NI 9401*. USA : National Instruments, 2015. 374068A-02.
7. **National Instruments.** *User's manual for NI 9215*. USA : National Instruments, 2016.
8. **Patočka, M.** *Řídicí elektronika - aktivní obvody*. Brno : Vysoké Učení Technické v Brně, 2004.
9. **Humusoft.** *USER'S MANUAL MF 624*. s.l. : HUMUSOFT, 2014.
10. **Blaha, P.** *Řízení a regulace I*. Brno : Vysoké Učení Technické v Brně, 2017.
11. **Hockicko, J., Kúdelčík, P.** *Základy fyziky*. Žilina : Žilinská univerzita v Žiline, 2011. ISBN 978-80-554-0341-0.
12. **Olsson, H.** *Control Systems with Friction*. Lund Sweden : Lund Institute of Technology, 1996. ISSN 0280–5316.
13. **Pivoňka, P.** *Číslicová řídicí technika*. Brno : Vysoké Učení Technické v Brně, 2012.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

- TTL* - signál s parametrami popisujúcimi logické stavy, od 0 do 0,8V logická 0, od 2 do 5 V logická 1 (*transistor-transistor logic*)
- FPGA* - pole logických členov programovateľné užívateľom (*Field programmable gate array*)
- DIO* - digitálne vstupno-výstupné (*Digital input output*)
- I/O* - vstup/výstup
- D/A* - digitálno-analógové
- A/D* - analógovo-digitálne
- SAR* - postupne aproximačný register (*Successive approximation register*)
- DPS* - doska plošných spojov
- DSP* - Digitálne signálové procesory
- GPIO* - digitálny vstupno-výstupný port

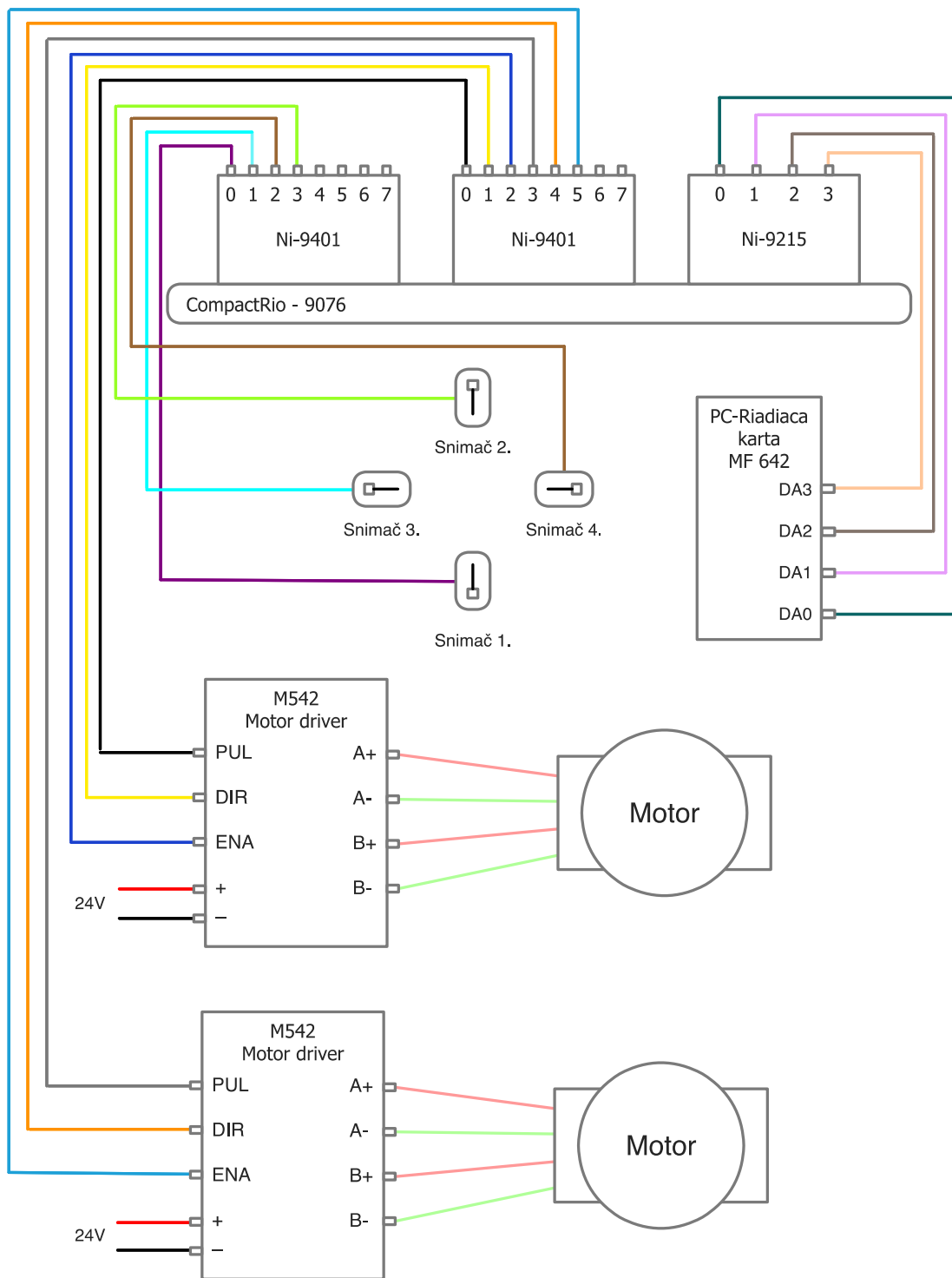
Zoznam cudzích slov

<i>Driver</i>	-	Ovládač pre krokové motory
<i>Microstepping</i>	-	Mikro-krokovanie
<i>Pull-out</i>	-	Názov parametru, doslovný preklad vytiahnuť
<i>Realtime</i>	-	V reálnom čase
<i>Triggering</i>	-	Spúšťanie
<i>Reshape</i>	-	Zmeniť/pretvoriť
<i>Signed</i>	-	So znamienkom
<i>Single-ended</i>	-	Názov pre prepojenie pomocou spoločného uzemnenia
<i>Convert</i>	-	Konvertovať

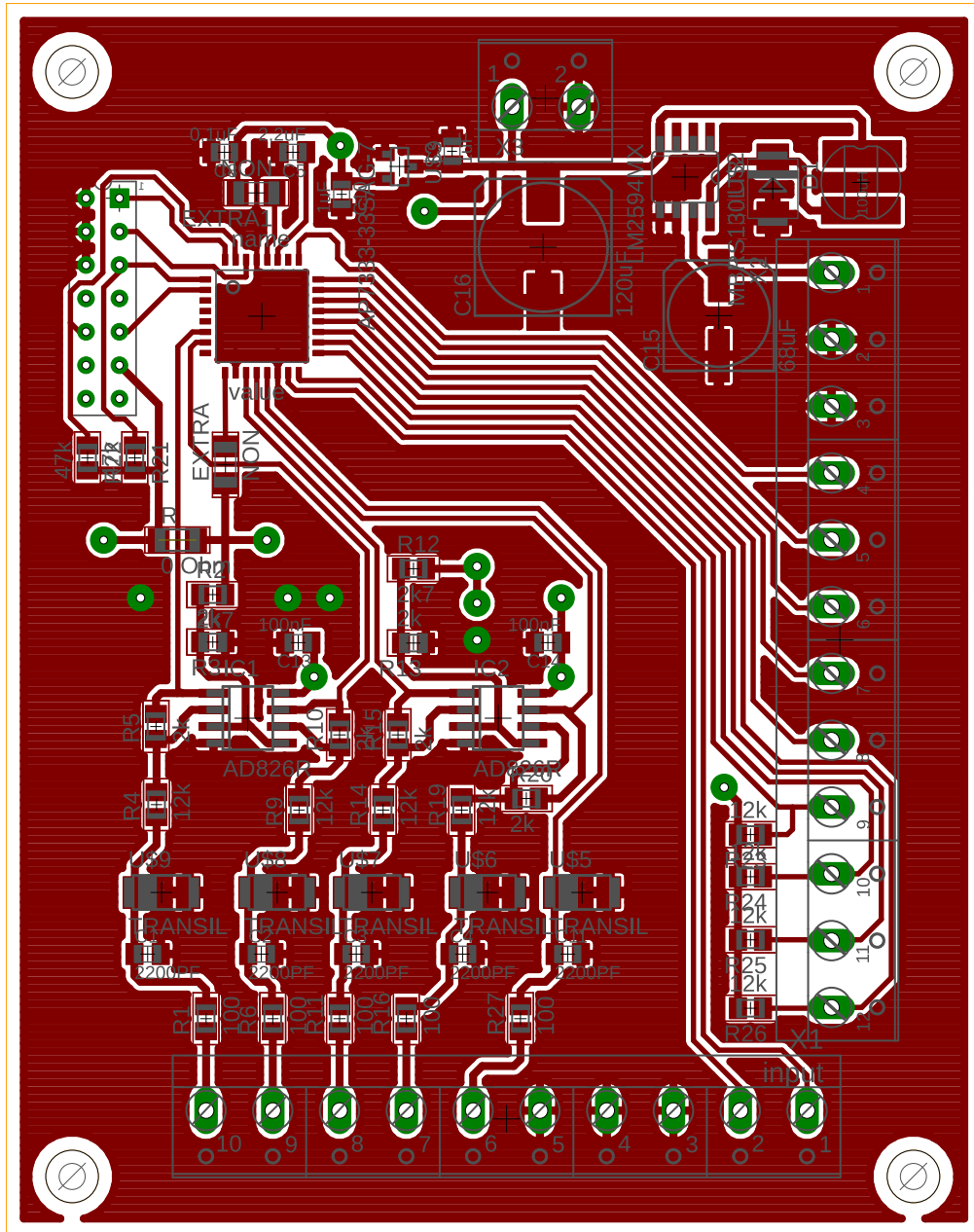
Zoznam príloh

Príloha č. 1.....	53
Príloha č. 2.....	54
Príloha č. 3.....	55
Príloha č. 4.....	58
Príloha č. 5.....	59
Príloha č.6, Príloha č.7, Príloha č.8, Príloha č.9, Príloha č.10, Príloha č.11v elektronickej podobe, priložené na DVD	

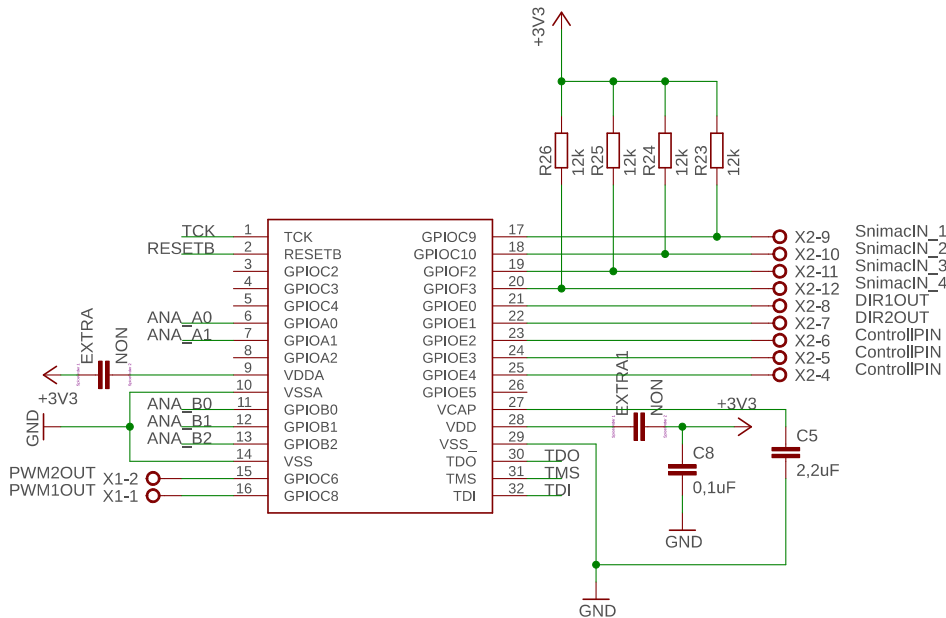
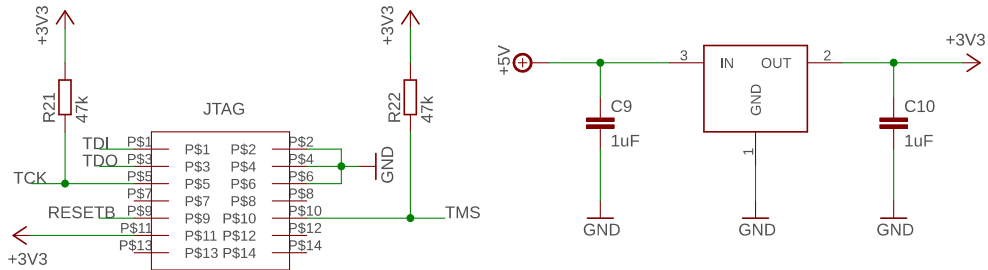
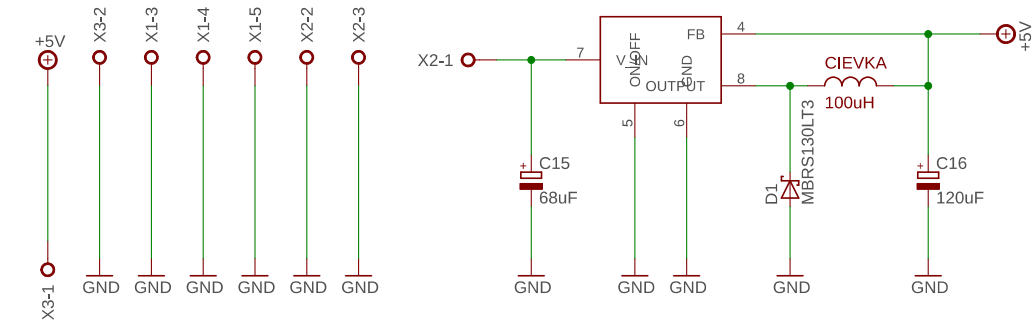
Príloha č. 1



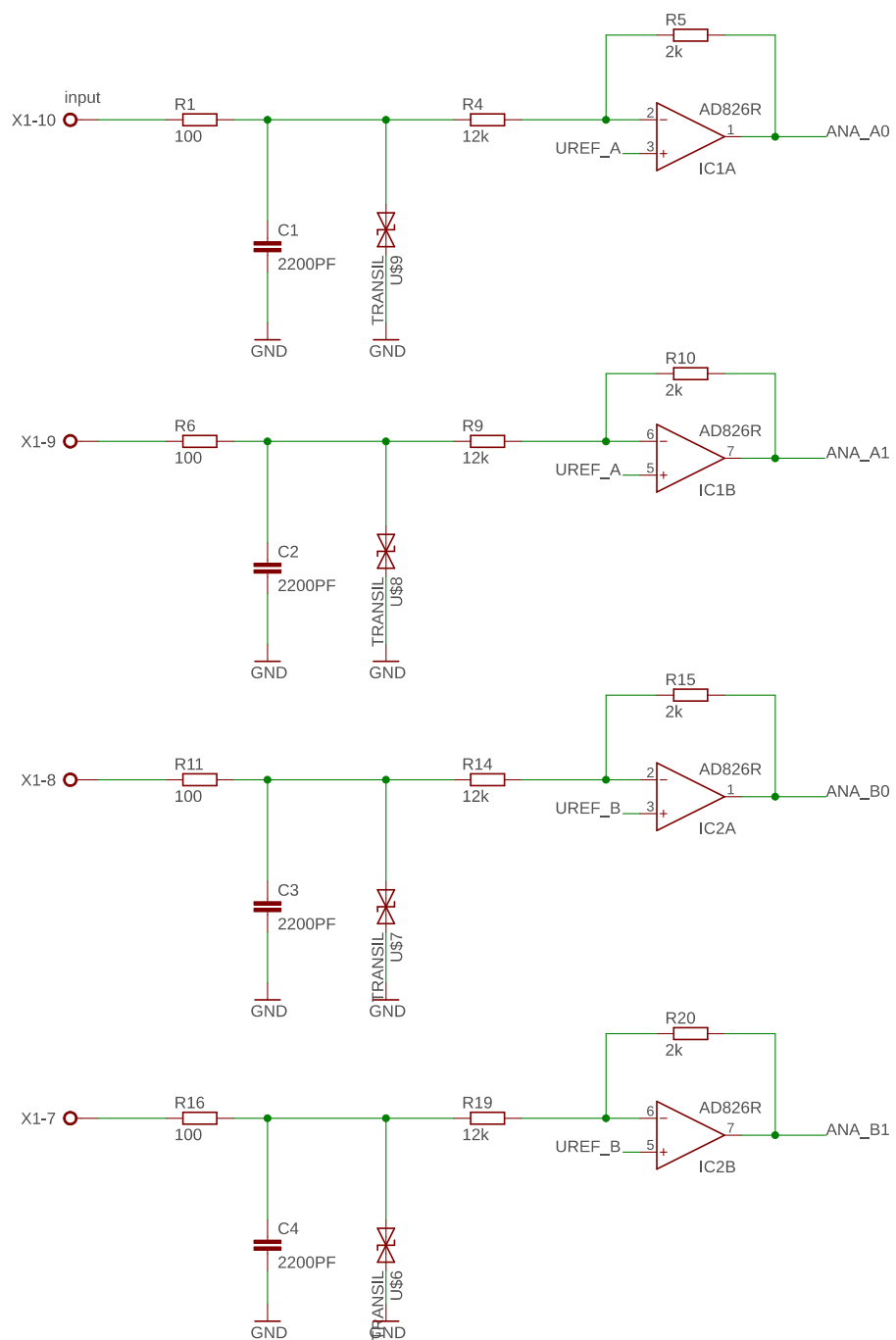
Príloha č. 2



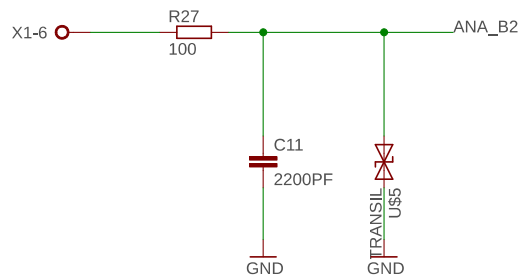
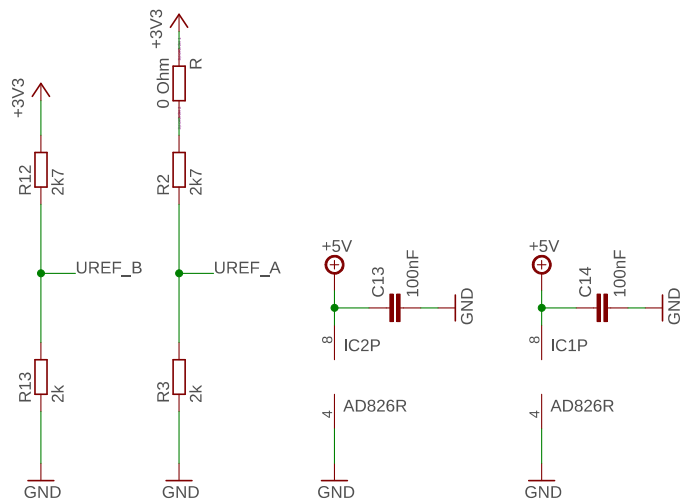
Príloha č. 3



Názov: Ovládač krokových motorov	Vypracoval: Ladislav Bednár
Číslo dokumentu: ZAPOJENIE_PROCESORA_01	
Dátum: 28.4.2018	Strana: 1/3

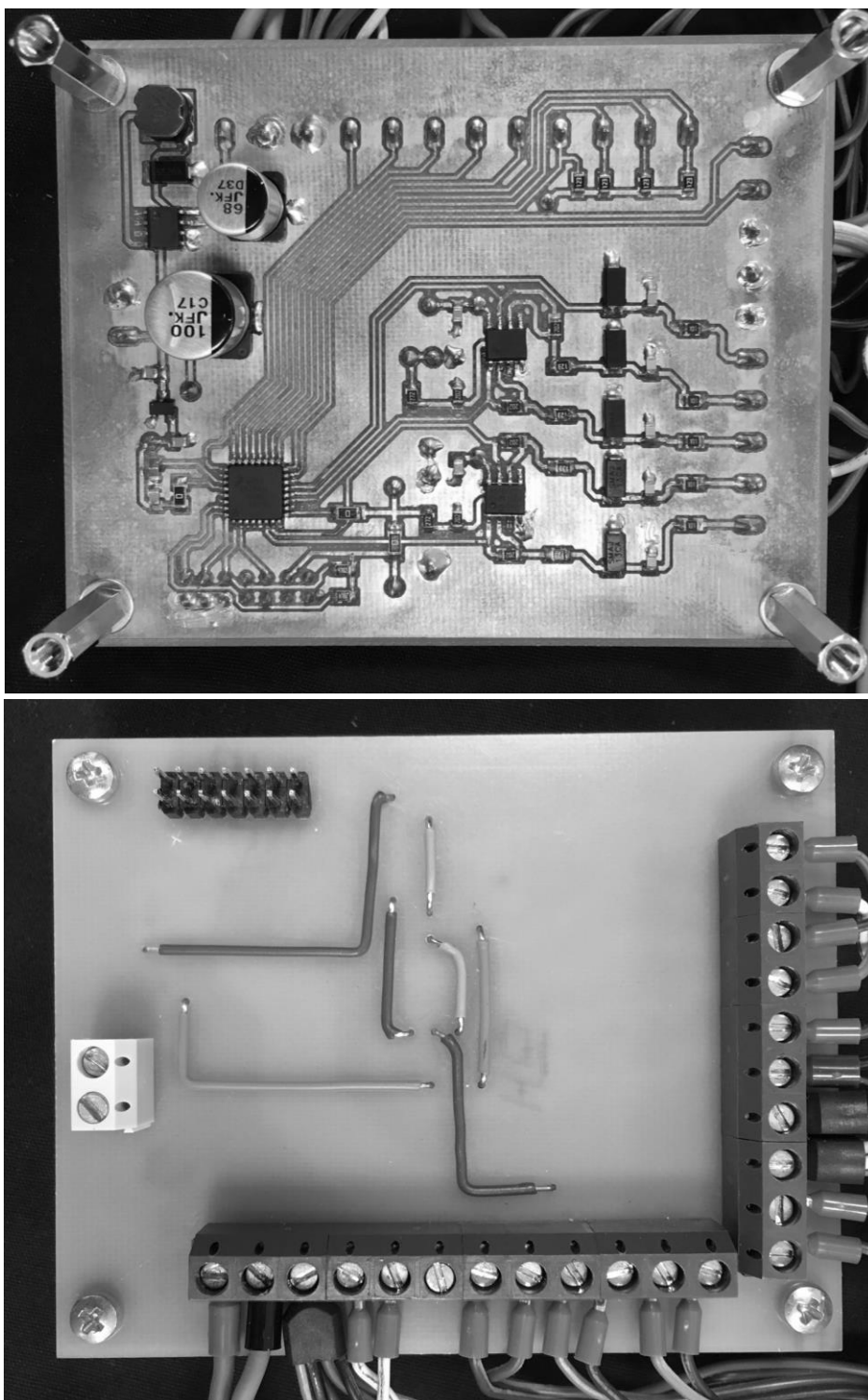


Názov: Ovládač krokových motorov	Vypracoval: Ladislav Bednář
Číslo dokumentu: ZAPOJENIE_OPERACNYCH_ZOSIL_02	
Dátum: 28.4.2018	Strana: 2/3



Názov: Ovládač krokových motorov	Vypracoval: Ladislav Bednář
Číslo dokumentu: NAPAJANIE_OPERACNYCH_ZOSIL_03	
Dátum: 28.4.2018	Strana: 3/3

Príloha č. 4



Príloha č. 5

Zoznam súčiastok:		
	SMAJ13CA+ - TVS Dioda, SMAJ Series, Obojsmerná, 13 V, 21.5 V, DO-214AC	SMAJ13CA+
	SMD Hliníkový Elektrolytický Kondenzátor, Radiální Plechovka - SMD, 68 µF, 63 V, Série FK	EEEFK1J680UP
	SMD Hliníkový Elektrolytický Kondenzátor, Radiální Plechovka - SMD, 100 µF, 63 V, Série FK	EEEFK1J680UP
	TVS Dioda, SMAJ Series, Obojsmerná, 5 V, 9.2 V, DO-214AC	SMAJ5.0CA+
	Výkonový Induktor (SMD), 100 µH, 700 mA, Stíněný, 800 mA, Série SRN6045, 6mm x 6mm x 4,5mm	SRN6045-101M
	Schottkyho Usměrňovač, 60 V, 2 A, Samostatná, DO-214AC, 2 Pinů, 550 mV	SS26A+
	Buck (Step Down) Spínaný Regulátor, Pevný, 4,5 V až 40 V Vstup, 5 V/500 mA Výstup, 150 kHz, SOIC-8	LM2594MX-5.0/NOPB
	Pevný LDO Napěťový Regulátor, 4,5 V až 12 V, 450 mV Pokles, 3,3Vout, 300mAout, SOT-23-3	RT9161-33GV
	Operační Zesilovač, Dva, AEC-Q100, 2 Zesilovače, 10 MHz, 7 V/µs, 2.5V až 5.5V, SOIC, 8 Pinů	MCP6021-2/SN
0805	Rezistor [Ω]	100, 2K, 2K7, 12K, 47K
0805	Kapacitor [F]	2200p, 100n, 1u, 2.2u
32pin	Procesor Freescale MC56F82743	