

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STROJNÍ

ÚSTAV MECHANIKY, BIOMECHANIKY A MECHATRONIKY

Odbor mechaniky a mechatroniky



BAKALÁŘSKÁ PRÁCE

**Návrh řízení pohybové osy pomocí předdefinovaných
kinematických trajektorií**

Praha, 2017

Jaroslav Běťák

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Běťák** Jméno: **Jaroslav** Osobní číslo: **438958**
Fakulta/ústav: **Fakulta strojní**
Zadávající katedra/ústav: **Ústav mechaniky, biomechaniky a mechatroniky**
Studijní program: **Teoretický základ strojního inženýrství**
Studijní obor: **bez oboru**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Návrh řízení lineární pohybové osy pomocí předdefinovaných kinematických trajektorií.

Název bakalářské práce anglicky:

Design of linear motion axis control using predefined kinematic trajectories.

Pokyny pro vypracování:

Cílem práce je vytvořit řídicí program, který umožňuje definovat žádané pohybové trajektorie pomocí různých typů interpolace a tyto trajektorie následně využít pro řízení krokového motoru.

1. Seznamte se detailněji s problematikou tvorby programů v Simulinku a Matlabu
2. Proveďte rešerši lineárních pohybových servo os a způsobů jejich řízení
3. Vytvořte Matlab/Simulink software pro řízení lineární pohybovou osy s krokovým motorem
4. Proveďte ověření funkčnosti navrženého programu

Seznam doporučené literatury:

- C. W. de Silva, Mechatronic Systems: Devices, Design, Control, Operation and Monitoring, CRC Press, 2007
- Valášek, M. a kol. Mechatronika, Praha : ČVUT 1995

Jméno a pracoviště vedoucí(ho) bakalářské práce:

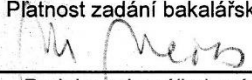
Ing. Martin Nečas MSc., Ph.D., ústav mechaniky, biomechaniky a mechatroniky FS

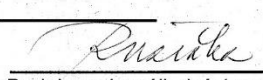
Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

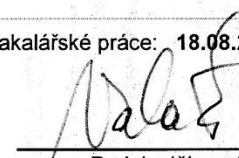
Datum zadání bakalářské práce: **19.04.2017**

Termín odevzdání bakalářské práce: **18.08.2017**

Přijatost zadání bakalářské práce: _____


Podpis vedoucí(ho) práce



Podpis vedoucí(ho) ústavu/katedry


Podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

4.5.2017
Datum převzetí zadání


Podpis studenta

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pouze za odborného vedení vedoucího práce. Veškeré zdroje informací, ze kterých jsem čerpal, jsou uvedeny v seznamu použité literatury.

V Praze, dne

Podpis:

Poděkování

Tímto bych rád poděkoval vedoucímu práce, panu Ing. Martin Nečas MSc. Ph.D. za odborné vedení práce, ochotu při poskytování rad i připomínek a vstřícnost během konzultací. Práce vznikla v rámci řešeného projektu pro Agenturu pro zdravotnický výzkum České republiky NV15-27941A.

Anotační list

Jméno autora:	Jaroslav Běťák
Název bakalářské práce:	Návrh řízení lineární pohybové osy pomocí předdefinovaných kinematických trajektorií
Anglický název:	Design of linear motion axis control using predefined kinematic trajectories.
Akademický rok:	2016/2017
Obor studia:	Teoretický základ strojního inženýrství
Ústav/odbor:	Ústav mechaniky, biomechaniky a mechatroniky Odbor mechaniky a mechatroniky
Vedoucí bakalářské práce:	Ing. Martin Nečas MSc. Ph.D.
Bibliografické údaje:	počet stran: 40 počet obrázků: 28 počet příloh: 1CD
Klíčová slova:	lineární pohybové osy, řízení, simulace, Matlab, Simulink
Keywords:	linear motion axes, control, simulation, Matlab, Simulink
Anotace:	Cílem práce je vytvoření řídicího programu, který umožňuje definování pohybových trajektorií a následné využití těchto trajektorií při řízení krokového motoru.
Abstract:	The main aim is to create a control program that allows a definition of the desired motion trajectory and used these trajectories to control a stepper motor.

Obsah

Prohlášení.....	3
Poděkování.....	4
Anotační list.....	5
Obsah	6
Úvod.....	8
Cíle práce	8
1 Lineární osy	9
1.1 Lineární motory.....	9
1.2 Krokové motory	11
1.2.1 Krokový motor s proměnnou reluktancí	12
1.2.2 Krokový motor s aktivním rotorem	13
1.2.3 Hybridní krokový motor	13
1.3 Servomotory	14
1.3.1 Stejnoseměrné servomotory	14
1.3.2 Komutovaný servomotor	15
1.3.3 Asynchronní servomotor.....	16
1.3.4 Synchronní servomotory.....	16
2 Převodní mechanismy.....	17
2.1 Kluzný pohybový šroub	17
2.2 Kuličkový šroub a matice.....	18
2.3 Pastorek a hřeben	19
3 Výpočtový program - Matlab	21
3.1 Simulační prostředí - Simulink	22
4 Řídicí software lineární osy	23
4.1 Použitá lineární osa	23
4.2 Vstupní data	25

4.3	Interpolátor	27
4.3.1	Interpolace	28
4.4	Stepper.....	30
5	Experiment s lineární osou	32
5.1	Sestavení modelu	32
5.2	Ovládání pohonu	33
5.2.1	Příprava dat	33
5.2.2	Spuštění HW	35
6	Závěr	38
7	Literatura.....	39
8	Seznam obrázků.....	40

Úvod

S dynamickým rozvojem nových průmyslových technologií klesá poptávka po manuální lidské práci, která je drahá a již mnohdy nedokáže vyhovět kvalitativním nárokům na finální produkt. Manuální práci v mnoha odvětvích nahrazují automatické výrobní linky, jež jsou schopny výrobu zefektivnit a eliminují riziko vzniku závad způsobených lidským faktorem. Tyto linky se sestávají z mnoha menších a samostatně pracujících celků. Ovládání celků bývá prováděno jednotlivými řídicími programy a je simulačně odladěno.

Cíle práce

Předmětem této práce je detailnější seznámení se s možnostmi realizace lineárních pohybových os a způsobem jejich řízení. Výsledkem práce bude simulační řídicí program, který na základě definovaných souřadnic bude vykreslovat interpolovanou křivku pohybu osy. Obdobný řídicí program bude následně použit přímo pro řízení krokového motoru skutečné osy.

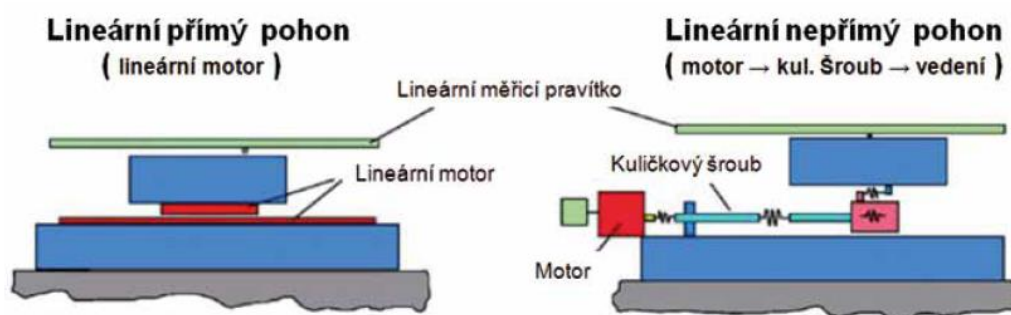
1 Lineární osy

Lineárními osami jsou nazývána zařízení pro realizaci přímočarého pohybu. V praxi jsou tyto osy hojně zastoupené, například jako polohovací zařízení nebo pohony ve výrobní technice.

K realizaci posuvného pohybu se nejčastěji používají servopohony, krokové motory nebo lineární motory.

1.1 Lineární motory

Jsou mnohopólové elektrické zařízení fungující na elektromagnetickém principu. Hlavní výhodou lineárních elektrických motorů je schopnost realizovat posuvný pohyb bez vložení dalších převodů. Toho je lineární motor schopen díky rotoru rozvinutému do roviny. Posuvové síly jsou vytvářeny přímo působením elektromagnetických sil mezi jezdcem primární části a částí sekundární. Primární část se u lineárních motorů nazývá stator a rozvinutá část, tvořící pojezd, je nazývána rotorem. Primární část je tvořena feromagnetickým svazkem tvořených z elektrotechnických plechů a trojfázového vinutí. [4]



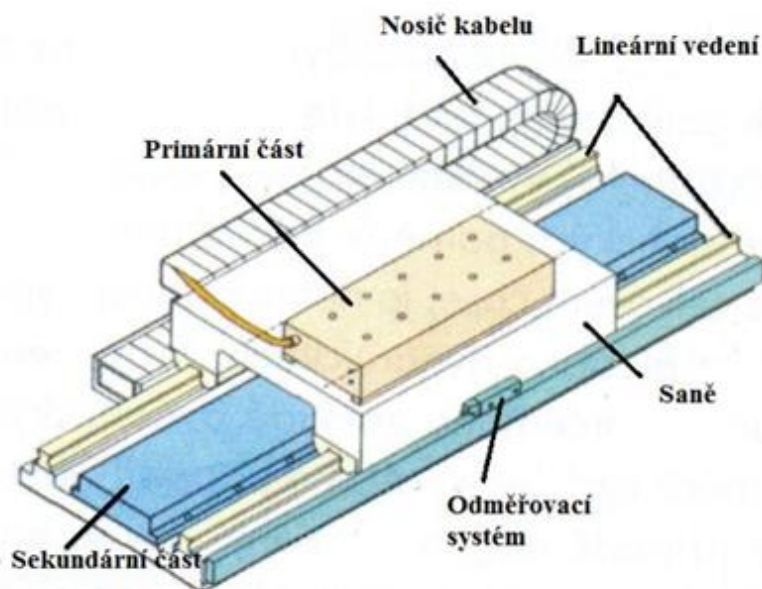
Obr. 1 - Srovnání lineární a klasický motor¹

Do primární části je přiváděn elektrický proud, jehož velikost je nastavená regulátorem a má vliv na rychlost pohybu motoru. Sekundární část je uložena proti primární a tvoří ji uspořádané permanentní magnety zalité v pryskyřici. Sekundární část bývá uspořádaná dle požadovaných specifikací a lze ji libovolně prodlužovat.

¹ Srovnání lineární a klasický motor. In: Hiwin s.r.o. - Polohovací systémy lineární osy [online]. [cit. 2017-03-18]. Dostupné z: www.hiwin.cz/media/files/05_Polohovaci_systemy_Linearni_osy.pdf

Lineární motory mají dobrou možnost regulace rychlosti, pomocí změn přívodního proudu, a jsou tak velice rychlé a dynamické. Bohužel elektromagnetické pole není schopno vybudit posuvové síly srovnatelné se silami získanými za pomoci ostatních typů pohonů. Proto jsou lineární motory používány v aplikacích, kde není zapotřebí vysokých sil, ale uplatní se jejich rychlost, např.: vysokorychlostní třískové obrábění, samoobslužné linky či polohovací mechanismy. V případě potřeby vyšších posuvových sil může být použito dvou motorů. Takto vzniklá dvojice bývá napájena z jednoho zdroje a řízená jedním odměřovacím systémem. [5]

Nevýhodou lineárních motorů je, že k pohyblivému se jezdcí musí být přiveden napájecí kabel, snímač polohy a případně chlazení. Veškerá kabeláž bývá uložena v chráněném nosiči, viz Obr. 2.



Obr. 2 - Části lineárního motoru²

Díky absenci veškerých vložených převodů dosahují lineární motory mimořádné přesnosti, zejména co se opakovatelnosti pohybu týče. Informace o poloze poskytuje zabudovaný odměřovací systém. Částečnou nevýhodou je vznik magnetického pole a jeho šíření mimo prostor samotného motoru. Aby se tomu zabránilo, používají se různé formy stínění.

² MAREK, Jiří a Petr BLECHA. Konstrukce CNC obráběcích strojů. Vyd. 2., přeprac. a rozš. Praha: MM publishing, 2010. MM speciál. ISBN 978-80-254-7980-3.

1.2 Krokové motory

Krokový motor je pojmenován podle toho, že jeho pohyb se skládá z jednotlivých kroků, jež ve výsledku tvoří točivý pohyb. Jde o typ synchronního motoru se značným počtem pólů. Princip fungování je takový, že proud procházející cívkou statoru vytvoří magnetické pole kolem cívky a natočí protipól rotoru. Vhodným přiváděním proudu na jednotlivé cívky je vytvořeno rotující magnetické pole, které následně unáší rotor. Mezi jednotlivými póly motoru je vykonáván nespojitý pohyb, jenž je odezvou na jeden puls ovládací cívky. [1]

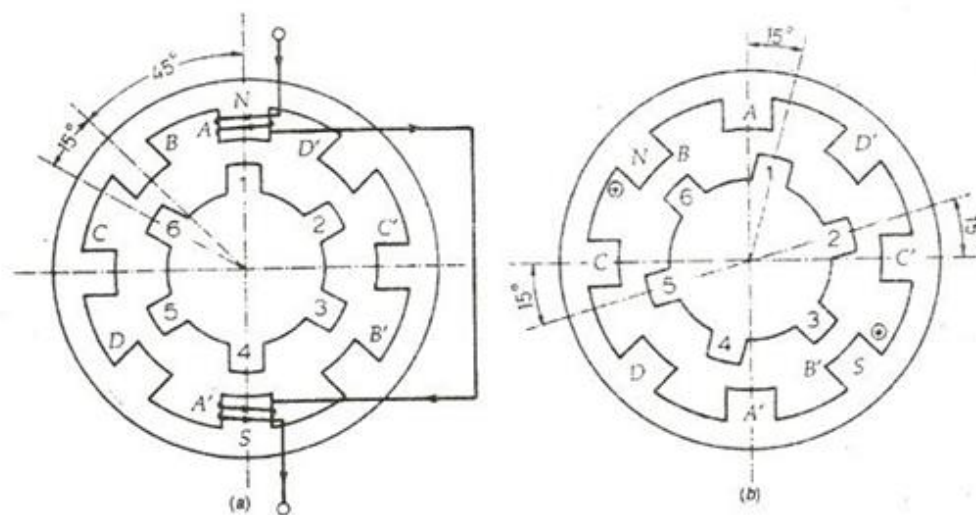
Samotný motor, po připojení do sítě, není schopen samostatného fungování. Správnou funkci zajistí až řídicí jednotka se systémem ovládající motor. Řídicí systém je založený na jednoduchých hnacích pulzech a ovládá jednotlivé fáze pohybu motoru. Pomocí hnacích pulzů lze přesně definovat polohu, do které se má motor natočit, a také lze libovolně měnit směr jeho otáčení. Otáčky motoru jsou řízeny změnou frekvence hnacích pulzů.

Výhodou (a současně možnou nevýhodou) krokových motorů je absence zpětného regulačního obvodu, který by udával informace o aktuální poloze. Počítá se s tím, že poloha je jasně definována hnacím pulsem z řídicí jednotky. Zároveň se motor při zastavení sám aretuje a není se třeba obávat změny polohy. [6]

Maximální rychlost otáčení je omezena magnetickými přechodovými jevy. Ty způsobují, že při určitých rychlostech a zatíženích může docházet k rezonancím vlivem vzájemného působení frekvencí hnacích pulzů a vlastní frekvence motoru. Kombinace těchto jevů může vést až ke ztrátě kroku motoru, tedy tomu, že se magnetické pole točí rychleji než samotný rotor. V takovém případě se ztrácí i informace o přesné poloze a motor musí být znovu inicializován. Další nevýhodou spojenou s nespojitým pohybem motoru je hlučnost jednotlivých kroků.

1.2.1 Krokový motor s proměnnou reluktancí

Tento typ krokového motoru využívá proměnného magnetického odporu, reluktance, v motoru. Tyto motory se používají jako tří a více fázové. Stator se skládá z izolovaných ocelových plechů, na nichž je navinut měděný drát, jenž tvoří vinutí jednotlivých cívek pólových nástavců. Protilehlé dvojice cívek bývají vždy vinutím spojené v jednu fázi. Rotor je vyroben z magneticky měkkého materiálu a nemá vlastní vinutí. Na povrchu rotoru jsou vystouplé zuby tvořící jeho póly. Hnací hřídel prochází rotorem a na stranách je podepřena kuličkovými ložisky. [1]



Obr. 3 - Reluktanční krokový motor³

Reluktance motoru je lineární funkcí překrytí zubů rotoru a pólových nástavců. Po připojení fáze, je indukčnost cívky největší, jsou-li osy zubů rotoru a statoru natočeny proti sobě. Naopak minimální indukčnost vzniká jejich vzájemným posunutím o polovinu zubové rozteče. Přivedením fáze na jeden z párů cívek je způsobeno to, že se motor natočí tak, aby reluktance obvodu, který je tvořen rotorem a připojenými póly statoru, byla minimální. V poloze s minimální reluktancí je motor ve stabilním stavu. Připojením dalšího páru cívek se motor opětovně posune do další stabilní polohy. Postupným spínáním cívek lze tedy plynule otáčet rotorem. Takto může být dosaženo vysokých rychlostí, ale s poměrně nízkým točivým momentem. Spínáním cívek v opačném pořadí se bude motor otáčet ve zpětném chodu. [6]

³ Variable-reluctance-motor-fig-2.jpg. In: Variable Reluctance Stepper Motor [online]. [cit. 2017-03-27]. Dostupné z: <http://circuitglobe.com/wp-content/uploads/2016/02/variable-reluctance-motor-fig-2.jpg>

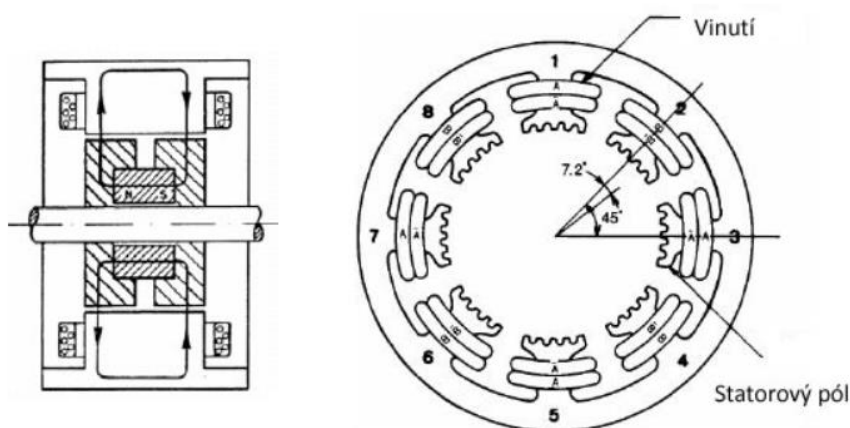
1.2.2 Krokový motor s aktivním rotorem

Na rozdíl od krokového motoru s variabilní reluktancí, už nemá na rotoru zuby, ale póly rotoru tvoří jednotlivé permanentní magnety. Magnety jsou uspořádány tak, aby se jejich magnetické póly na obvodu rotoru střídaly. Počet pólů rotoru je poloviční oproti počtu pólů statoru.

Použitím většího počtu pólů, se krokový úhel motoru zmenšuje, motor je lépe tlumen a roste dosažitelný točivý moment. Výhodou konstrukce krokového motoru s permanentními magnety je jeho jednoduchost a s tím související nízká výrobní cena. Díky tomu je hojně užíván ve spotřební a automobilové elektrotechnice. [1]

1.2.3 Hybridní krokový motor

V současné době jsou nejpoužívanějším typem hybridní krokový motory. To díky tomu, že kombinují užité vlastnosti předešlých dvou zmíněných typů. Jsou rychlé jako reluktanční motory a silné jako motory s permanentními magnety.



Rotor motoru je tvořen dvěma pólovými nástavci z lisovaných plechů. Ty mají zuby, jež jsou vzájemně pootočené o polovinu zubové rozteče. Počet zubů rotoru určuje velikost kroku motoru. Mezi nástavci je navíc vložen permanentní magnet, díky němuž mají nástavce opačnou polaritu. Stator má na jednotlivých pólech zuby stejného tvaru jako rotor. Počet zubů statoru a rotoru není shodný, rotor má více zubů než stator. [6]

⁴ Konstrukce hybridního motoru. In: Krokový motor [online]. [cit. 2017-04-01]. Dostupné z: <http://www.pohonnatechnika.cz/skola/motory/krokovy-motor>

1.3 Servomotory

Servomotory jsou elektrické regulační pohony s možností vysoce přesného řízení polohy a rychlosti. Skládají se z hnacího elektromotoru s napájecím výkonovým měničem a řídicích regulačních obvodů. Zatímco dříve zmíněné krokové motory byly zapojené v takzvané otevřené regulační smyčce, tj. bez zpětné vazby, tak servomotory musí být vždy zapojené v uzavřené regulační smyčce, tedy se zpětnou vazbou.

Při výběru servomotoru pro určitou aplikaci má největší význam právě přesnost a rychlost regulační smyčky. Zpětná vazba poskytuje informace o aktuální rychlosti motoru a o jeho poloze. Dle požadovaných regulačních a výkonových parametrů může být vybráno z různých konstrukčních provedení servomotoru. [7], [8]

1.3.1 Stejnoseměrné servomotory

Stejnoseměrně napájený servomotor je ideální zejména díky snadné regulaci pomocí změny napájecího napětí. Další výhodou je bezpečnost napájecího proudu a poměrně vysoká účinnost. Ve stejnoseměrném provedení se nejčastěji používá kartáčkového motoru buzeného pomocí permanentních magnetů ve statoru. Rotor je tvořen vinutím jednotlivých cívek, jež jsou připojeny na komutátor. Komutátor je otáčející se přepínač, který je v kontaktu s kartáčky a s jeho pomocí se mění polarita cívek rotoru.

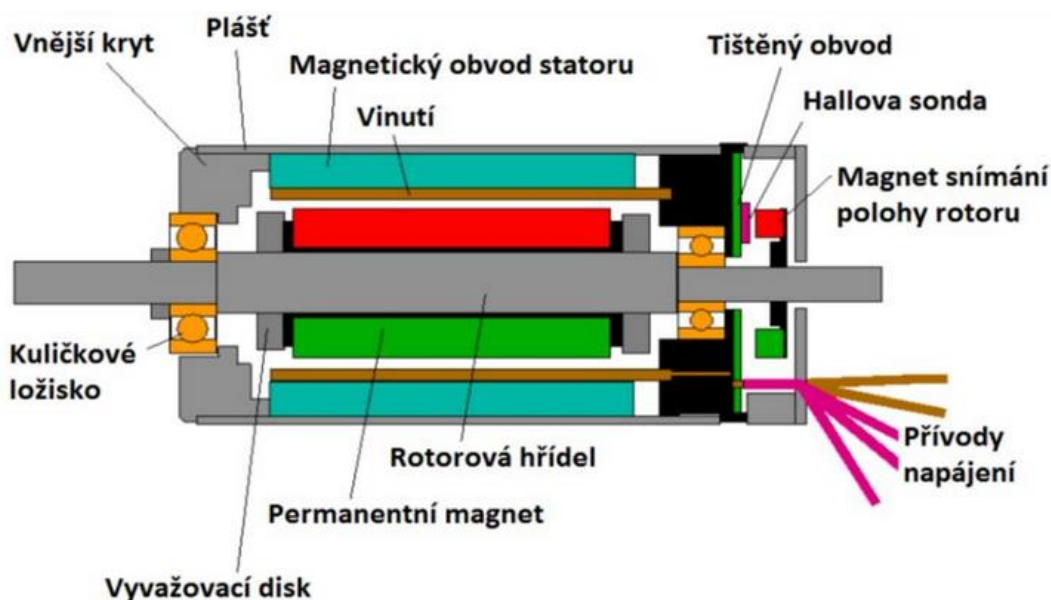
Pro dosažení vyšších výkonů lze použít motory s cizím buzením. Cizí buzení je zajištěno budícím vinutím v pólech statoru. Průchodem proudu tímto vinutím je vytvářeno zesilující magnetické pole, jež zvyšuje výkon motoru. Nevýhodou jsou vyšší ztráty ve formě Joulova tepla a s tím spojené vyšší nároky na chlazení motoru.

Dalším důležitým parametrem, kromě výkonu, je dodávaný točivý moment a jeho zvlnění. Ideální je, aby dodávaný moment během otáček motoru byl co nejrovnoměrnější a tudíž měl servomotor co největší tuhost chodu a vysokou přesnost regulace rychlosti a polohy. Rovnoměrnosti momentu se dosahuje zešikmením drážek rotoru ve vztahu k podélnému vedení jeho vinutí. [7], [8]

1.3.2 Komutovaný servomotor

Jedná se taktéž o stejnosměrný elektromotor, ale na rozdíl od předešlého jde o motor bez komutátoru. Tím odpadají problémy spojené s opotřebením kartáčků a možným jiskřením motoru. Další výhodou komutovaných servomotorů je jejich vysoká výkonost a dynamika.

Komutovaný servomotor má permanentní magnety umístěné na rotoru a vinutí na statoru. Stator má podobnou stavbu jako u třífázových strojů a v jeho drážkách je uloženo třífázové vinutí. Funkci zmiňovaného chybějícího komutátoru zde přejímá tranzistorový měnič, který podle natočení rotoru přepíná proud do jednotlivých statorových vinutí. Rotor může být uspořádán s magnety na povrchu nebo vestavěnými do tělesa rotoru. To podle požadavků na dosaženou koncentraci magnetického toku.



Obr. 5 - Komutovaný motor⁵

Pro samotnou funkci motoru a ovládání tranzistorového měniče musí být snímána poloha rotoru. Snímání je prováděno nejčastěji diskrétně, pomocí magnetických či fotoelektrických sond, viz Obr. 5. Zpracováním výstupních signálů sond, prostřednictvím řídicích obvodů, se generují parametry pro komutaci proudu na jednotlivá statorová vinutí. Otáčková zpětná vazba bývá zajištěna bezkontaktním komutovaným tachodynamem. [8]

⁵ Elektronicky komutovaný motor. In: Elektromotorky [online]. [cit. 2017-04-07]. Dostupné z: <http://player.slideplayer.cz/42/11246095/data/images/img10.png>

1.3.3 Asynchronní servomotor

Asynchronní motory jsou dnes nejpoužívanějším typem elektrických motorů. S rozvojem řídicí elektrotechniky, která umožňuje přesnou regulaci jejich otáček, se začaly hojně využívat i v servosystémech. Fungují na principu unášivého magnetického pole mezi rotorem a státorem. Napájeny jsou třífázovým napětím, jež má jednotlivé fáze od sebe posunuty o třetinu periody. Vynikají zejména svou robustností, spolehlivostí a také jsou poměrně levné.

Stator je tvořen třífázovým vinutím zapuštěným do drážek kostry statoru a může být zapojen do trojúhelníku nebo do hvězdy. Rotor bývá tvořen klecovým vinutím s hliníkových či měděných tyčí spojených nakrátko. Druhou možností je rotor kroužkový, kde jsou izolované sběrací kroužky zapuštěné v tělese rotoru. Připojováním vnějších odporů ke kroužkovému vinutí rotoru se můžou měnit charakteristiky motoru a lze tím dosáhnout lepších rozběhových parametrů a regulace skluzu. U motoru s klecovým vinutím rotoru tento princip nelze aplikovat, namísto toho může být použita tzv. vírová kotva, která využívá zvýšení odporu pomocí snížení efektivního průřezu vodiče. [8]

1.3.4 Synchronní servomotory

Synchronní motory jsou elektrické stroje, jejichž rotor se otáčí synchronně s unášivým magnetickým polem statoru. Výhodou synchronních motorů, oproti motorům asynchronním, je jejich menší váha a rozměry. Dalšími výhodami jsou menší tepelné ztráty a větší rovnoměrnost chodu.

Pro použití v servopohonech se používá synchronních motorů s permanentními magnety na rotoru, které se konstrukčně dosti podobají komutovaným motorům. Stator bývá napájen třífázovým sinusovým proudem z napěťového měniče kmitočtu. Magnety mohou být, stejně jako u komutovaného elektromotoru, na povrchu rotoru nebo zapuštěné dovnitř rotoru. Rozdílné je snímání polohy, zatímco u komutovaných motorů stačilo natočení rotoru snímat diskretně, tj. po určitých intervalech, asynchronních motory jsou snímány nepřetržitě, například pomocí resolveru. Navíc je používán termistor na statoru motoru, jež slouží jako ochrana proti tepelnému přetížení vinutí. Dále se motory dovybavují nouzovou brzdou, pro případ nenadálého výpadku proudu. [8]

2 Převodní mechanismy

Zatímco lineární motor je konstruován tak, že je schopen rovnou vykonávat lineární pohyb, tak při užití rotačního servomotoru nebo krokového motoru musí být zajištěn převod rotačního pohybu na pohyb posuvný. V této kapitole jsou vyjmenovány základní možnosti realizace tohoto převodu.

2.1 Kluzný pohybový šroub

Jedním ze základních způsobů převodu rotačního pohybu na pohyb posuvný je pomocí pohybového šroubu. Je používán zejména tam, kde jsou malé rychlosti přímočarého pohybu a zároveň vysoké převodní poměry.

Konstrukčně se pohybový šroub skládá z vlastního šroubu a matice. Samotný převodní mechanismus lze poté realizovat dvěma odlišnými způsoby. Prvním způsobem je otáčející se šroub a posouvající se matice. Druhým je stojící šroub a matice, která se otáčí a zároveň posouvá. V některých aplikacích lze využít i kombinaci předešlých dvou způsobů.

Při volbě profilu závitové dvojice šroubu je na výběr z více možností s ohledem na požadované vlastnosti. V současnosti je nejpoužívanější souměrný lichoběžníkový závit. Jeho předností je levná výroba a snadná možnost vymezování vůle v závitech. V případě vyšších požadavků na přesnost odměřované dráhy může být použit profil lichoběžníkový nesouměrný nebo plochý profil závitu. [3]

Šrouby se vyrábí z oceli a matice bývají zpravidla bronzové, výjimečně litinové. Pro zajištění správného chodu musí mít závit určitou vůli, která má ovšem vliv na přesnost polohy. Vliv celkové vůle na přesnost polohování se musí analyzovat a případně korigovat. Vůli lze vymezit stažením rozříznuté matice nebo použitím púlené matice. [2]

Nevýhodou tření v kluzných plochách šroubu je následný nárůst teplot styčných ploch. Proto je důležité zajištění čistoty těchto ploch a jejich správné promazání. Další nevýhodou kluzných šroubů je nízká účinnost, zhruba 30 až 40 %, a možnost trhavého pohybu. Z těchto důvodů jsou v náročnějších provozech kluzné šrouby na ústupu a nahrazují je šrouby kuličkové.

2.2 Kuličkový šroub a matice

Spojení pomocí kuličkového šroubu a matice, neboli KŠM, funguje na principu valivého tření. Výhodou valivého tření, oproti tření kluznému, je vznik menšího opotřebení a dosažení delší životnosti převodu. Takto realizované spojení se vyznačuje účinností pohybující se okolo 90% a minimálním oteplováním během provozu.

Šroub i matice jsou vyrobeny z oceli. Po celé délce šroubu je přesná kalená drážka. V této drážce mezi šroubem a maticí obíhají ocelové kuličky, jež se vždy na konci matice, pomocí vratného kanálku, dostanou zpátky na začátek matice. Jistou nevýhodou spojení KŠM je nesamosvornost, proto musí být zařazen brzdící prvek. Pro zvýšení tuhosti matice je potřeba vyvolat předpětí. Toho se dosáhne pomocí matice složené ze dvou částí nebo posunutím stoupání matice. [3]



Obr. 6 - Kuličkový šroub ⁶

Drážka mezi šroubem a maticí může být zhotovena ve dvou odlišných profilech. Buďto s klasickým kruhovým profilem nebo s profilem gotickým. Kruhový profil je výrobně snazší a také levnější než gotický. Má ovšem menší účinnost, a také se vlivem malých stykových ploch rychleji opotřebovává. Gotický profil je, oproti kruhovému, tvořen dvěma lomenými oblouky. Tím je zajištěn čtyřbodový styk, jenž prodlužuje životnost jak kuliček, tak i matice. [2]

⁶ *Ball-screw.jpg*. In: Продукция - Флуитек Системз: [online]. [cit. 2017-03-14]. Dostupné z: <http://fluitech.com.ua/assets/images/Ball-screw/ball-screw.jpg>

2.3 Pastorek a hřeben

V aplikacích s dlouhými posuvy se jako pohon používá pastorek s hřebenem. Výhodou, oproti převodu realizovaného šroubem a maticí, je vyšší účinnost a možnost převodu s nižšími otáčkami, který eliminuje kmitání.

Spojení pastorku s hřebenem je nesamosvorné, proto je třeba zajistit jeho havarijní zastavení pomocí nouzové brzdy, např. elektromagnetické. Přesnost chodu je menší než za použití pohybového šroubu, neboť konvenční výrobní technologie nedosahují stejné přesnosti jako výrobní technologie u kuličkových šroubů. Jestliže by mělo být dosaženo vyšší přesnosti, musely by se použít specializované výrobní stroje [2]



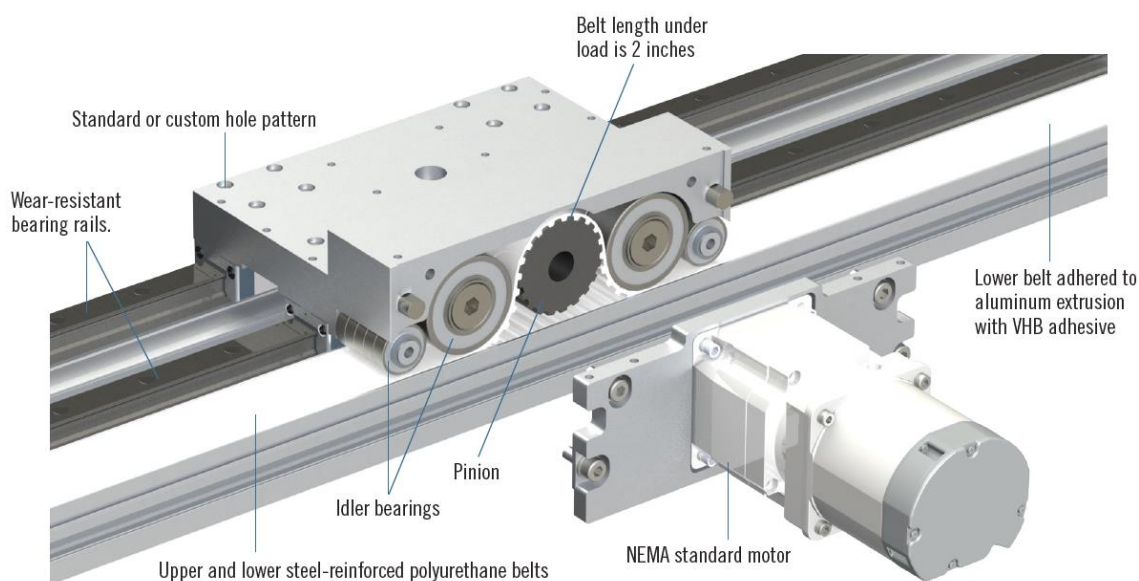
Obr. 7 - Pastorek a hřeben⁷

Pro použití v přesném polohování je nutné vymezení vůle mezi pastorkem a hřebenem. Vymezení je docíleno pomocí předepnutí. To se provádí mechanicky pružinou, děleným pastorkem nebo elektrickými servomotory. [3] Obvykle je použito dvou servomotorů, každý s jedním pastorkem, přičemž první motor „Master“ slouží k pohonu a druhý motor „Slave“ jenom přizpůsobuje a vymezuje vůli.

⁷ Hřebeny a pastorky od Atlanta. In: Hřebeny a pastorky [online]. [cit. 2017-06-15]. Dostupné z: <https://www.rem-technik.cz/foto/1000-700/58-734-hrebeny-a-pastorky.png>ServoBelt systém

Technicky zajímavým způsobem převodu rotačního pohybu na posuvný je systém ServoBelt od firmy Bell-Everman, Inc. Jedná se o speciální typ řemenového převodu, který funguje na podobném principu jako převod mezi pastorkem a hřebenem. Tento patentovaný systém elegantním způsobem řeší problém poddajnosti hnacích řemenů a to tím, že k vyvození síly dochází lokálně od protilehlého řemene, který je pevně spojen s rámem.

Lineární pohyb vykonává platforma, která se pohybuje po pevném hliníkovém loži. Hnací rotační pohyb je zajištěn třífázovým bezkartáčovým servomotorem. Motor pohání pastorek, jenž se pohybuje mezi dvěma ozubenými polyuretanovými řemeny. Každý z řemenů plní odlišnou roli. První, tzv. aktivní, je tažen přes napínací kladky, opásává pastorek a slouží k realizaci pohybu. Druhý řemen je připevněn k hliníkovému loži, kde funguje obdobně jako ozubnice, viz Obr. 8. Mimo záběr pastorku jsou oba řemeny spojeny a tím chráněny před zanášením nečistotami. [9]



Obr. 8 - Princip fungování ServoBelt systému⁸

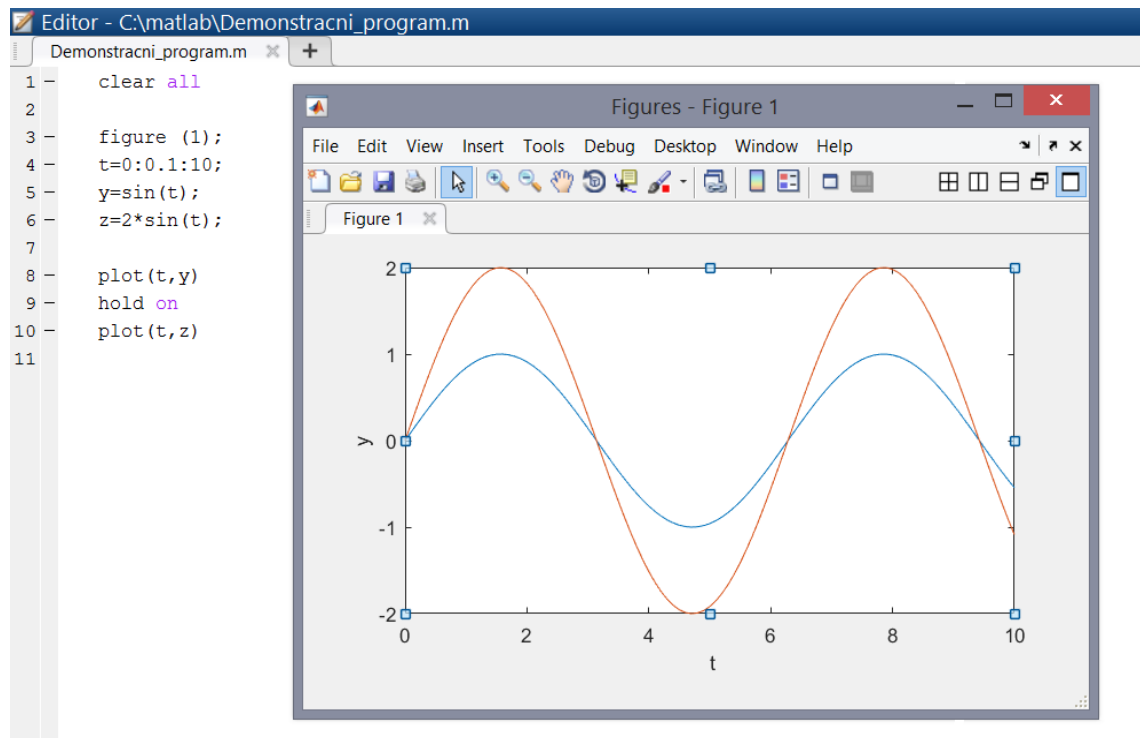
Výhodou řemenového pohonu, oproti například lineárním motorům, je nízká pořizovací cena se současně dobrými výkonovými parametry. Příznivá je taky přesnost a vysoká opakovatelnost pohybu v obou směrech. Nevýhodou jsou možné vibrace platformy, a tím i snížená životnost ložisek vedení.

⁸ BE_Data_SBL_r14.indd - Bell_Everman_SBL_H_Tech_Specs.pdf [online]. 2014 [cit. 2017-04-11]. Dostupné z: http://bell-everman.com/images/stories/Bell_Everman_SBL_H_Tech_Specs.pdf

3 Výpočtový program - Matlab

K řízení pohonu lineární osy budou použita data získána z výpočtového programu Matlab. Tento program je speciálně navržen pro vědecké a inženýrské výpočty. Ke svému fungování používá vlastní interpretovaný programovací jazyk s knihovnou předdefinovaných funkcí sloužících k zjednodušení programování. Další výhodou Matlabu je možnost okamžité vizualizace výsledků pomocí řady integrovaných vykreslovacích příkazů.

Základem programování v Matlabu, jak přímo plyne z jeho nezkráceného názvu Matrix Laboratory, jsou tzv. datová pole. V polích jsou jednotlivé hodnoty uloženy v řádcích a sloupcích ve formě matic. Přístup k těmto hodnotám je zprostředkovan pomocí zadání názvu pole a příslušného indexu. Pole mohou být jednodimenzionální nebo více dimenzionální. Každé pole zabírá určitou velikost v paměti počítače, tato část paměti obsahující dané pole se nazývá proměnnou.



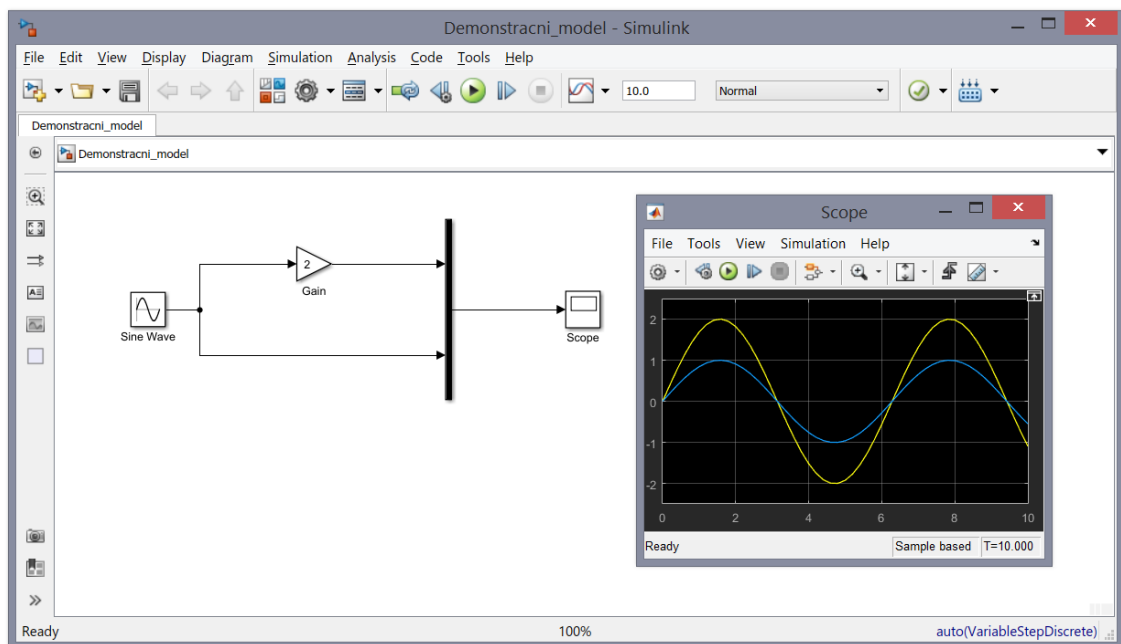
Obr. 9 - Demonstrační program v Matlabu

Definováním jednotlivých proměnných a následným zpracováním dat v nich se provádí veškeré operace v Matlabu, od základních matematických úkonů, jako sčítání a násobení, až po složitější operace definované v knihovně funkcí.

3.1 Simulační prostředí - Simulink

Simulink je integrované grafické prostředí Matlabu, jenž slouží pro simulování a analyzování modelů dynamických soustav. Soustavy jsou modelovány pomocí sestavených schémat skládajících se z funkčních bloků a výsledný model je tvořen propojením těchto bloků. Bloky jsou voleny z předpřipravené knihovny, kde jsou rozděleny do jednotlivých *toolboxů* podle svého účelu. V případě potřeby je možné si nadefinovat vlastní bloky.

Po sestavení modelu je třeba spustit simulaci systému. Během simulace jsou použity vstupní signály a zpracovány do požadovaných výsledků. Je třeba si uvědomit, že veškeré funkce v Simulinku jsou funkcí času. Tento čas je simulační nebo může být použita simulace založena na skutečném čase. Počáteční a koncový čas je nadefinován před spuštěním simulace. Rychlost simulace se odvíjí od výpočetního výkonu počítače.



Obr. 10 - Demonstracní model v Simulinku

Výsledky simulace je možné zobrazit na výstupních blocích. Následně lze tyto výstupy přímo analyzovat, případně změnit nastavené parametry a simulaci opakovat. Zároveň se mohou výsledky ukládat a později být použity při dalších výpočtech.

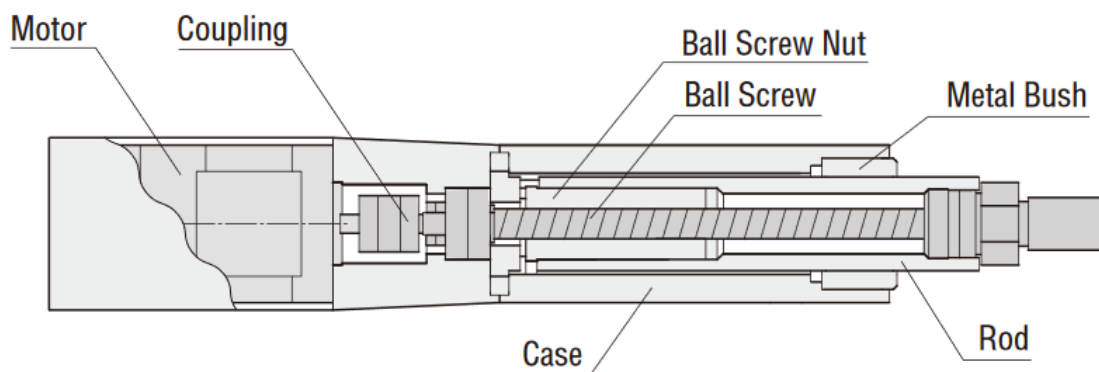
4 Řídicí software lineární osy

V této části je popsán proces tvorby řídicího programu pro pohon osy. Přiblížena je reálně použitá pohybová osa a způsob jakým bude řízení provedeno. Dále je uveden popis tvorby jednotlivých částí programu, postup při zpracování vstupních dat a tvorba vstupních signálů pro řízený pohon.

4.1 Použitá lineární osa

Před samotným začátkem tvorby řídicího programu musí být zjištěný typ lineární osy a způsob jakým je možno ji ovládat.

V případě této práce se jedná o systém lineární osy EZ Limo od firmy Oriental Motor, konkrétně model EZCM 6, viz Obr. 11. Tento systém slouží jako kompaktní řešení linearizace pohybu a je navrhnut tak, aby po dodání mohl být co nejrychleji uveden do provozu.

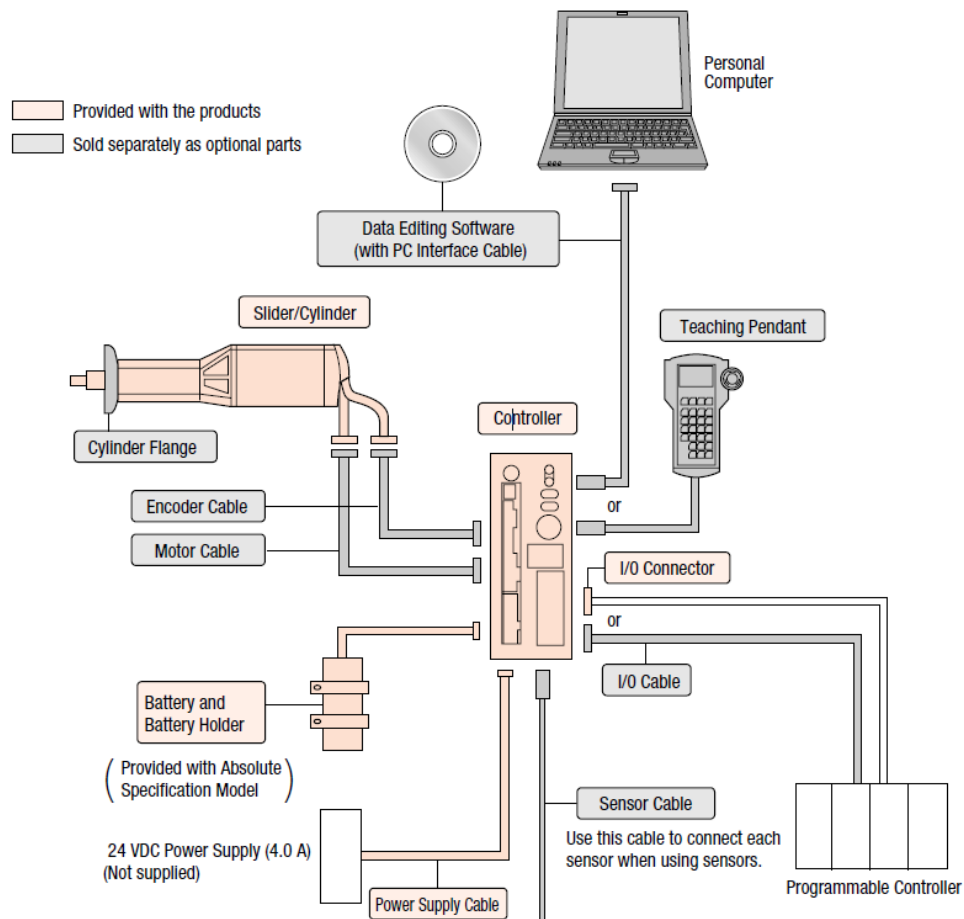


Obr. 11 - Lineární osa EZ Limo¹¹

Osa je poháněna krokovým motorem s vestavěným snímačem polohy, který zajišťuje zpětnou vazbu po uzavřené regulační smyčce. Touto smyčkou je garantována přesnost a stabilita provozu. Samotný rotační pohyb motoru je převeden na pohyb posuvný pomocí kuličkového šroubu. Pro případ ztráty kontroly nad motorem či překročení maximálního rozsahu je celý systém vybaven nouzovou elektromagnetickou brzdou. [10]

¹¹ EZ Limo. In: ORIENTAL MOTOR - Documentation: EZ limo [online]. [cit. 2017-04-09]. Dostupné z: <http://www.audin.fr/pdf/documentations/oriental-motor/mouvement-lineaires-et-actuateurs/EZC.pdf>

Samotná osa s pohonem a kuličkovým šroubem tvoří jen jednu část systému. Druhou částí je tzv. controller. Zatímco těleso osy slouží pouze ke zprostředkování samotného pohybu, tak controller zprostředkovává propojení mezi motorem osy a řídicím programem, schéma jeho zapojení je k vidění na Obr. 12. Controller tedy má na starosti napájení a chod motoru, kontrolu polohy a zpracovávání vstupního signálu z počítače.



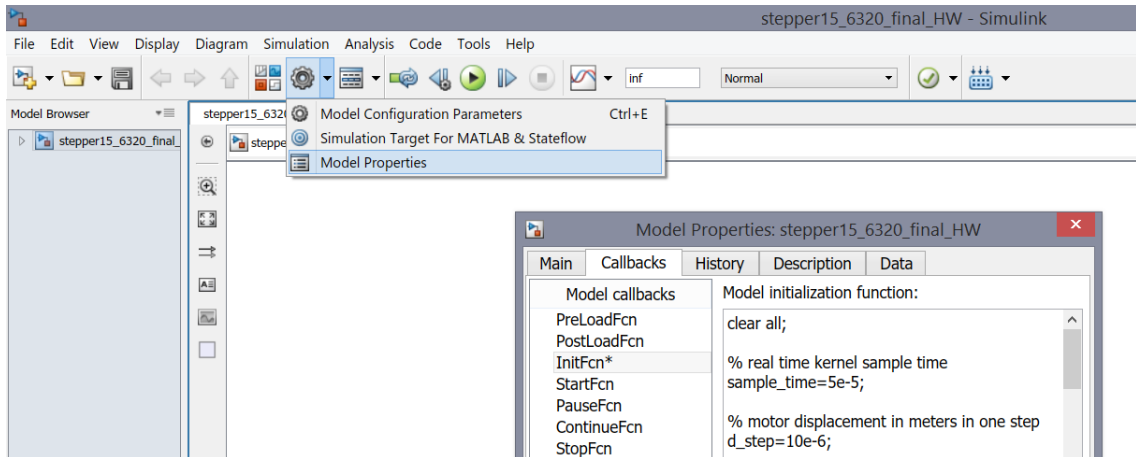
Obr. 12 - Zapojení řídicí jednotky¹²

Controller umožňuje výběr mezi dvěma možnými způsoby ovládání. První způsob je pomocí řídicího softwaru na počítači a druhý prostřednictvím doplňkového ovladače (Teaching pendant). Pro námi zvolený způsob řízení bude použit počítač s ovládacím softwarem v prostředí Simulink.

¹² EZC series. In: Ezlino_catalog.pdf [online]. 2004 [cit. 2017-04-09]. Dostupné z: http://www.dynaservo.com/products/documents/ezlino_catalog.pdf

4.2 Vstupní data

Vstupními daty jsou pole interpolovaných souřadnic, v nichž se má osa v zadaných časech nacházet. Interpolované souřadnice jsou vykreslovány ze zadaných parametrů trajektorie. Data pro výpočet trajektorii jsou zadávány do inicializační funkce. Tu lze otevřít z nástrojové lišty v Simulinku, *Model properties* → *Callbacks* → *InitFcn*.



Obr. 13 - Přístup k vstupním datům

Začátek inicializační funkce obsahuje vstupní parametry určené na základě technické dokumentace motoru. Jedná se zejména o velikost jednotkového časového kroku, který je uložen v proměnné *sample_time*. Velikost tohoto časového kroku vychází z toho, že motor je schopen přijímat řídicí pulzy s frekvencí až 20 kHz. Dále je uvedena velikost posunutí na jeden krok a maximální rychlost motoru. Maximální rychlost nesmí být překročena, protože poté dochází ke ztrátě kroku a motor přestane vykonávat pohyb po požadované trajektorii.

```
clear all;

% real time kernel sample time
sample_time=5e-5;

% motor displacement in meters in one step
d_step=10e-6;

% max permissible velocity with selected step and t_step
v_max=d_step/(2*sample_time)
```

Obr. 14 - Definování základních parametrů

Následující části kódu zabývající se definováním základních dat pro sestavení trajektorie. Nejprve je určen počet a velikost datových polí, do kterých budou vstupní data vkládána. Data v těchto polích se řetězí a obsahují veškeré potřebné informace k provedení interpolace, struktura je k vidění na Obr. 15.

```

%prealokace velikosti pole
n = 101;

% pocet datovych poli
m = 2;

global Data
Data = ones(m,5,n)*-1;

dataset = 1;
T = [0 1 5 10 15];
Y = [0 0.01 0.03 0.04 0];
Data(dataset,1,1:length(T)) = T; % T osa
Data(dataset,2,1:length(Y)) = Y; % Y osa
Data(dataset,3,1) = 3; % #repetice
Data(dataset,4,1) = 1; % typ interpolace 1-akima
Data(dataset,5,1) = length(T);

%prealokace prvnioho prvku
switch_times(1) = Data(1,1,Data(1,5,1)) * Data(1,3,1);

%vypocet switch times
for i = 2:size(Data,1)
    switch_times(i) = switch_times(i-1)+Data(i,1,Data(i,5,1)) * Data(i,3,1);
end

```

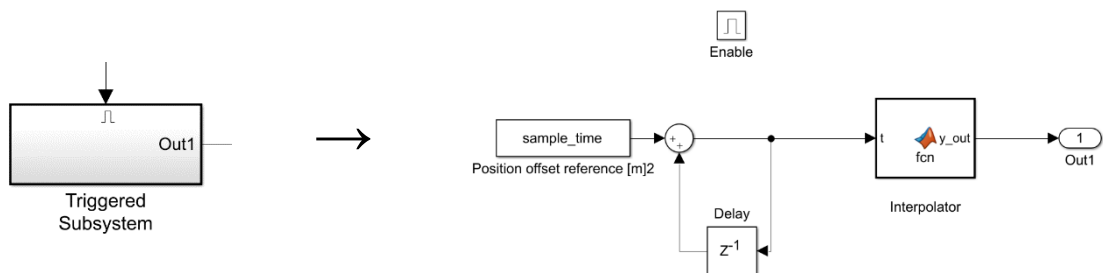
Obr. 15 - Ukázka strukturování vstupních dat

Vektor T je vektorem lokálního času a vždy musí začínat nulou. Y je vektorem příslušných poloh určených k interpolaci. Platí, že jednotlivé hodnoty vektoru Y jsou polohy v časech určených vektorem času T . Tudiž musí být počet prvků obou vektorů shodný. Dále je uveden požadovaný počet repetice, který určuje, kolikrát za sebou se mají interpolované hodnoty, pro jedno datové pole opakovat. Nakonec je vybrán způsob požadované interpolace.

Zároveň musí být určeny časy přechodů jednotlivých polí. K tomu slouží proměnná $switch_times$, jež na základě délky časového vektoru T , a počtu jeho opakování, vypočte počáteční časy pro další pole.

4.3 Interpolátor

Po nadefinování vstupních dat trajektorií do inicializační funkce je potřeba provést samotnou interpolaci. K tomu slouží skript *interpolator_simulink.m*. Tento skript načte zadané parametry trajektorie a jeho výstupem jsou interpolované souřadnice, které jsou funkcí reálného času. Celý skript je zakomponován do spínaného subsystému, viz Obr. 16, který slouží pro vstup dat do spuštěného řídicího programu.



Obr. 16 - Subsystém s interpolátorem

V úvodu si skript načte proměnné z datových polí *Data* a *switch_times*. Podle parametrů v nich obsažených je vybráno aktuální datové pole pro běžící reálný čas. Z tohoto pole jsou získány informace o celkovém trvání jedné jeho periody, počtu repetice a typ žádané interpolace.

```
Editor - C:\matlab\interpolator_simulink.m
interpolator_simulink.m x +
1 function y_out = interpolator_simulink(t)
2 %function y = interpolator_simulink(Data, switch_times,t)
3
4 global Data
5 global switch_times
6
7 % vektor aktualniho
8 data_set = find(switch_times > t,1);
9
10 % selekce aktualnich dat a jejich parametru
11 data_set_time_period = Data(data_set,1,Data(data_set,5,1));
12 data_set_repetition = Data(data_set,3,1);
13 interpolace = Data(data_set,4,1);
14
15 % vypocet aktualniho globalniho casoveho offsetu
16 if t < switch_times(1)
17     time_offset = 0;
18 else
19     time_offset = switch_times(data_set-1);
20 end
21
22 % modulace casu v ramci opakujich se sekvenci
23 t = mod(t-time_offset,data_set_time_period);
```

Obr. 17- Ukázka z *interpolator_simulink.m*

V další části kódu je stanoven časový offset. Ten se rovná nule, jestliže se pohybujeme v prvním interpolovaném datovém poli. V opačném případě se offset rovná době trvání předešlého datového pole. Časový offset umožňuje modulaci času pro provedení repetice.

Před volbou konkrétní interpolace je načtena proměnná *rozsah*. Ta poslouží jako pomocná proměnná při tvorbě vektorů interpolovaných hodnot. Výběr interpolace je proveden pomocí příkazu *switch*, viz Obr 18. Pro každou zadanou číselnou hodnotu proměnné *interpolace* je určen odlišný typ jejího provedení. Hodnota proměnné musí být od jedné do tří. V opačném případě je výstup interpolace z daného datového pole roven nule.

```

Editor - C:\matlab\interpolator_simulink.m
interpolator_simulink.m
25 global rozsah
26
27
28 % volba konkrétní intepolace
29 switch interpolace
30
31     case 1
32         % zpusob interpolace_1
33         rozsah = reshape(Data(data_set,5,1),1,1);
34         y_out = akima(squeeze(Data(data_set,1,1:rozsah)), squeeze(Data(data_set,2,1:rozsah)),t);
35
36     case 2
37         % zpusob interpolace_2
38         rozsah = reshape(Data(data_set,5,1),1,1);
39         y_out = interp1(squeeze(Data(data_set,1,1:rozsah)), squeeze(Data(data_set,2,1:rozsah)),t,'linear');
40
41     case 3
42         % zpusob interpolace_3
43         rozsah = reshape(Data(data_set,5,1),1,1);
44         y_out = interp1(squeeze(Data(data_set,1,1:rozsah)), squeeze(Data(data_set,2,1:rozsah)),t,'spline');
45     otherwise
46         y_out = 0;

```

Obr. 18 – Možné typy interpolace

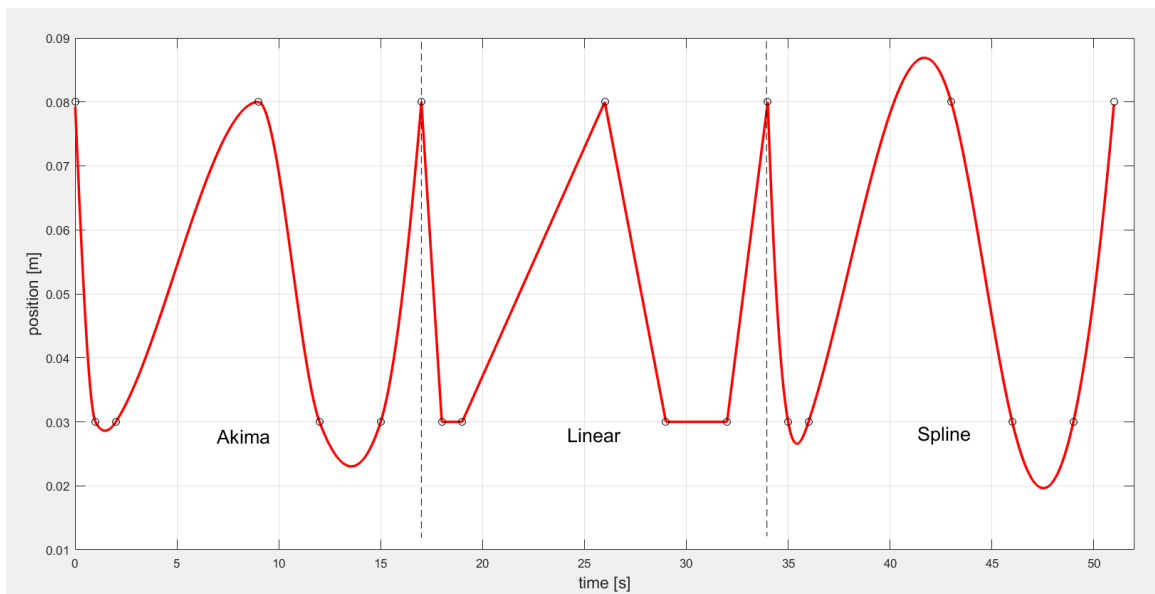
4.3.1 Interpolace

Protože zadávaná datová pole obsahují jen pár souřadnic příslušící hodnotám vektoru času T , tak pro dokreslení zbytku křivky je použita interpolace. Požadavkem je, aby výsledná kinematická trajektorie procházela zadanými body. Zbytek bodů křivky bude dopočítán pro hodnoty reálného času t .

Vstupní data před vložením do inicializační funkce si lze otestovat ve skriptu *Data_input_new.m*. Tento skript využívá stejné datové struktury jako datové pole v *InitFcn* a umožňuje vykreslení interpolovaných křivek do grafu. Poté co jsou křivky zkontrolovány, tak je stačí překopírovat do Simulinku.

Nezákladnějším a zároveň nejjednodušším provedením je interpolace lineární. Ta spojuje body pouze za pomoci polynomu 2. řádu. Výslednou trajektorii jsou tedy zadané body pospojované pomocí úseček a to pouze s C_0 spojitostí.

Další možností je interpolační spline. Ten vykresluje chybějící body křivky pomocí kubické interpolace. Pro sestavení splinu je potřeba nejméně 4 zadaných bodů a je výpočtově náročnější než lineární interpolace, ale lze tak dosáhnout přesnějších hodnot s C_2 spojitostí křivek.



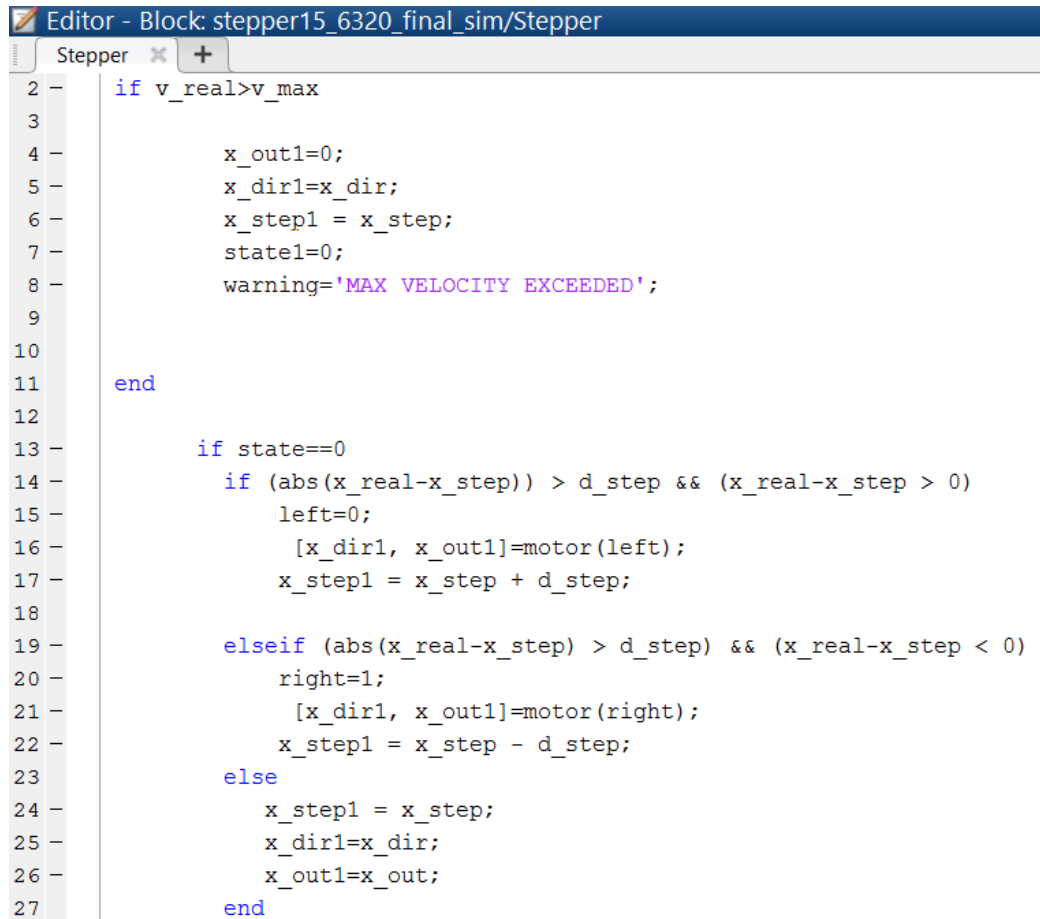
```
T = [0 1 2 9 12 15 17];
Y = [0.08 0.03 0.03 0.08 0.03 0.03 0.08];
```

Obr. 19 - Ukázka interpolovaných křivek

Třetím použitým typem interpolace je akima spline. Nejde o běžně nabízenou interpolaci z Matlabu, a tak musela být doplněna prostřednictvím skriptu *akima.m*. K dopočítání chybějících bodů křivky využívá také kubických polynomů, ale jeho výhodou je, že nemá takovou tendenci k vlnění mezi vzdálenějšími hodnotami jako klasický spline. Srovnání jednotlivých typů interpolace, pro stejně zadanou trajektorii, je k vidění na Obr. 19.

4.4 Stepper

Řízení samotného krokového motoru je zajištěno funkcí *Stepper*. Ta zpracovává souřadnice polohy získané z *interpolator_simulink.m* a následně z nich vytváří výstupní signál pro controller lineární osy.



```
Editor - Block: stepper15_6320_final_sim/Stepper
Stepper x +
2 -   if v_real>v_max
3
4 -       x_out1=0;
5 -       x_dir1=x_dir;
6 -       x_step1 = x_step;
7 -       state1=0;
8 -       warning='MAX VELOCITY EXCEEDED';
9
10
11      end
12
13      if state==0
14          if (abs(x_real-x_step)) > d_step && (x_real-x_step > 0)
15              left=0;
16              [x_dir1, x_out1]=motor(left);
17              x_step1 = x_step + d_step;
18
19          elseif (abs(x_real-x_step) > d_step) && (x_real-x_step < 0)
20              right=1;
21              [x_dir1, x_out1]=motor(right);
22              x_step1 = x_step - d_step;
23          else
24              x_step1 = x_step;
25              x_dir1=x_dir;
26              x_out1=x_out;
27          end
```

Obr. 20 - Funkce Stepper

V úvodu funkce je vložena podmínka pro kontrolu maximální povolené rychlosti chodu motoru. Při překročení této rychlosti dochází k tzv. ztrátě kroku motoru. Při překročení rychlosti dojde k dočasnému zastavení motoru, které trvá do doby, než se rychlost navrátí na povolenou hodnotu.

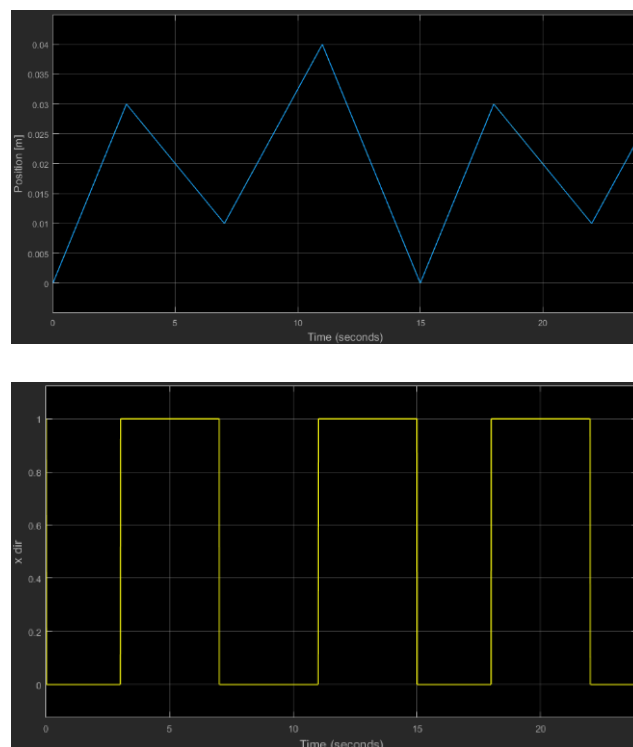
Následuje část kódu zabývající se ovládáním motoru. Je vyhodnocováno, zda je možné vykonat pohyb, případně ve kterém směru. Nejprve je určena absolutní hodnota rozdílu mezi současnou a požadovanou polohou. Tato hodnota musí být větší než velikost jednoho kroku uloženého v proměnné *d_step*. Pokud by tato podmínka nebyla splněna, tak pohyb není možný a motor zůstane stát na místě.

Při splnění podmínky proveditelnosti kroku je dále rozhodováno o směru pohybu. Rozdíl současné a požadované polohy se porovná s nulou a tím je určeno, zda se má osa pohybovat dopředu, případně se vrátit. Pro pohyb motoru to znamená směr otáčení doleva nebo doprava. Určený směr pohybu je uložen do funkce *motor(dir)*. Výstupy této funkce následně slouží jako řídicí signály pro controller motoru osy. Proměnná *x_dir* určuje směr pohybu motoru a *x_out* je spínací pulz pro vykonání jednoho kroku.

```
Editor - Block: stepper15_6320_final_sim/Stepper
Stepper x +
46 function [x_dir, x_out]=motor(dir)
47     if dir==0
48         x_dir=1;
49         x_out=1;
50     else
51         x_dir=0;
52         x_out=1;
53     end
```

Obr. 21 - Funkce *motor(dir)*

Controller vyžaduje, aby se vstupní signály řídily negativní logikou, proto jsou na výstupech proměnných *x_dir* a *x_out* umístěny bloky s negativní logikou, které převrací hodnoty proměnných.



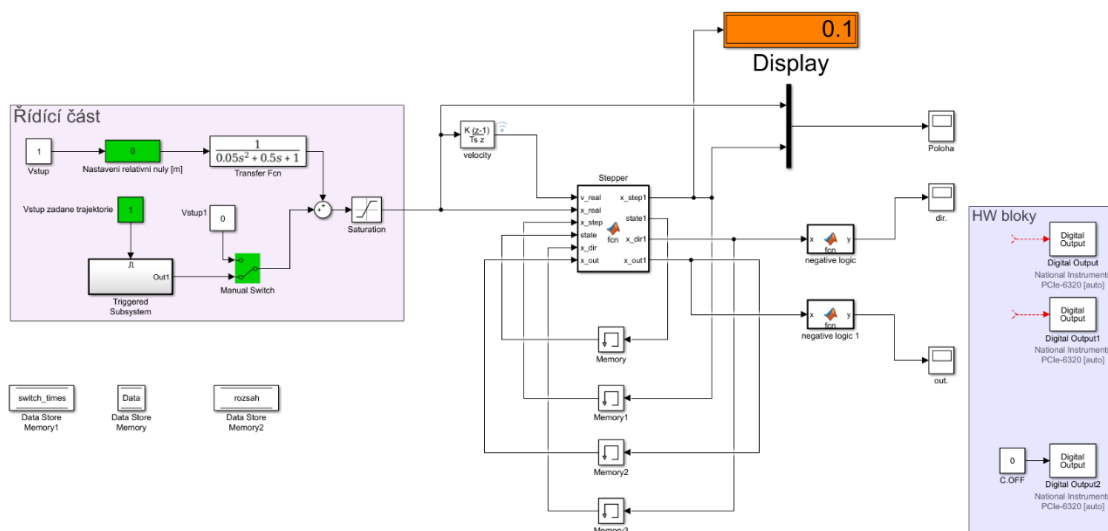
Obr. 22 - Ukázka změny směru motoru

5 Experiment s lineární osou

Tato kapitola je zaměřena na implementaci řídicího softwaru na reálnou osu. Popisuje způsob ovládání lineárního pohonu osy a obsahuje základní informace potřebné k obsluze.

5.1 Sestavení modelu

Sestavené funkční bloky a naprogramované skripty byly poskládány do výsledného modelu. Model umožňuje snadné zadání vstupních hodnot a nastavení výchozích parametrů. Po zadání vstupních dat a spuštění modelu, může uživatel pozorovat aktuální informace o poloze osy na zvýrazněném displeji, hodnoty jsou uvedeny v metrech. Případně lze také otevřít některý z grafů, jež vykreslují průběhy simulačních veličin v závislosti na reálném čase.



Obr. 23 - Řídicí model

Řídicí modely byly vytvořeny ve dvou variantách. První variantou je *stepper15_6320_final_sim.slx*, ten umožňuje snadné testování vstupních dat bez nutnosti mít k dispozici počítač propojený s řízenou osou.

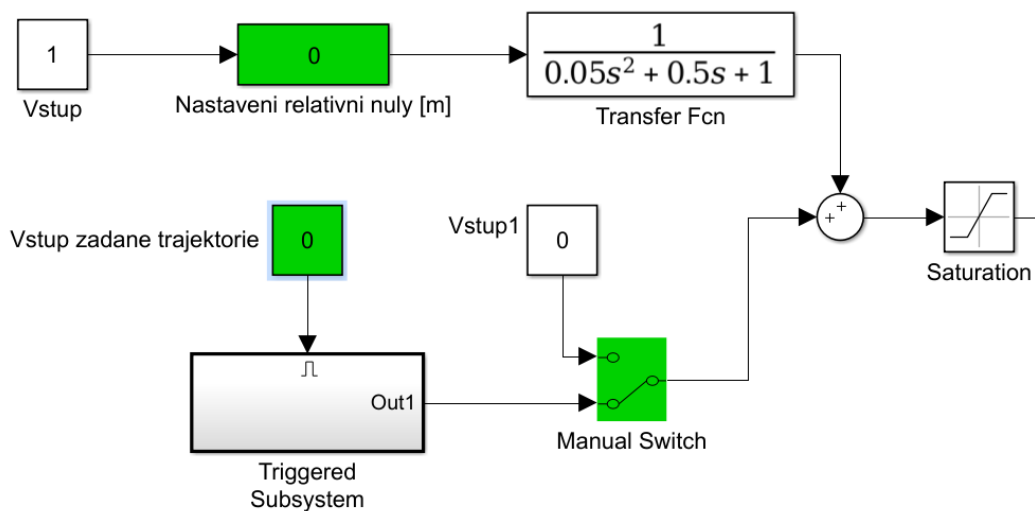
Druhou variantou je model *stepper15_6320_final_HW.slx*. Jediný rozdíl mezi oběma modely je, že druhý zmíněný obsahuje bloky *Digital output*, jenž zajišťují propojení mezi modelem v počítači a controllorem samotné osy. Tyto bloky jsou zobrazeny v oddělení HW bloky na Obr. 23.

5.2 Ovládání pohonu

Spouštění a ovládání pohonu osy lze rozdělit do dvou samostatných etap. První etapou je příprava a kontrola výchozích dat a druhou etapou je použití těchto dat v řídicím programu propojeném s pohonem osy.

5.2.1 Příprava dat

Příprava a kontrola správnosti vkládaných parametrů probíhá v simulačním programu *stepper15_6320_final_sim.slx*. Nejprve je nutné zadat vstupní parametry a definující data trajektorie do inicializační funkce *InitFcn*. Postup vkládání dat byl blíže popsán v kapitole 4.2 Vstupní data.



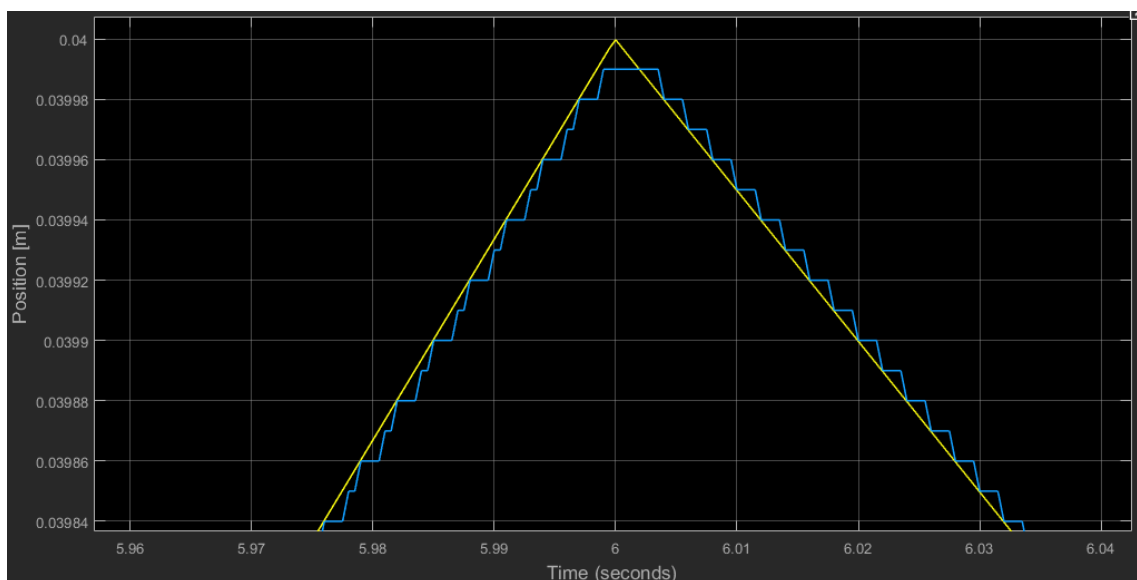
Obr. 24 - Řídící část modelu

Poté, co jsou definovány vstupní data, se nastaví parametry v řídicí části modelu, viz Obr. 24. Nejprve je nutné zkontrolovat, zda je hodnota relativní nuly skutečně nulová. Následně je dočasně vypnut vstup zadávání trajektorie pomocí nastavení taktéž nulové hodnoty. Zároveň se manuální přepínač přepne do dolní polohy, jinak by vstupem byla konstantní nulová hodnota.

Po provedení výchozího nastavení řídicí části modelu, může být přistoupeno k jeho samotnému spuštění. Model se spustí tlačítkem *Run*, umístěným v horní liště Simulinku. Úspěšnost dokončení kompilace programu a spuštění simulace lze poznat podle běžícího reálného času v dolní liště.

V případě, že se spuštění simulace povedlo, lze nyní změnit výchozí hodnoty v řídicí části modelu. Opět je nejprve provedena případná změna nastavení relativní nuly na požadovanou hodnotu, ke které má být trajektorie superponována. Po stisknutí Control+D, pro aktualizaci modelu, lze pozorovat tuto změnu na displeji. Nastavením hodnoty jedna, do políčka vstupu zadané trajektorie, je tento vstup povolen. Po otevření grafu „Poloha“ a opětovném stisknutí Control+D se tato změna projeví tím, že se začne daná trajektorie v grafu vykreslovat.

V této chvíli je možno přistoupit ke kontrole správného vynesení trajektorie. Především musí být zkontrolováno, zda se modrá křivka polohy motoru kryje se žlutou křivkou předepsané trajektorie, viz Obr. 25. Vzájemná vzdálenost obou křivek by neměla být větší než velikost jednoho kroku motoru, čili 0,01 mm.



$$T = [0 \ 6 \ 8 \ 13 \ 18];$$

$$Y = [0 \ 0.04 \ 0.03 \ 0.02 \ 0];$$

Obr. 25 - Kontrola trajektorie

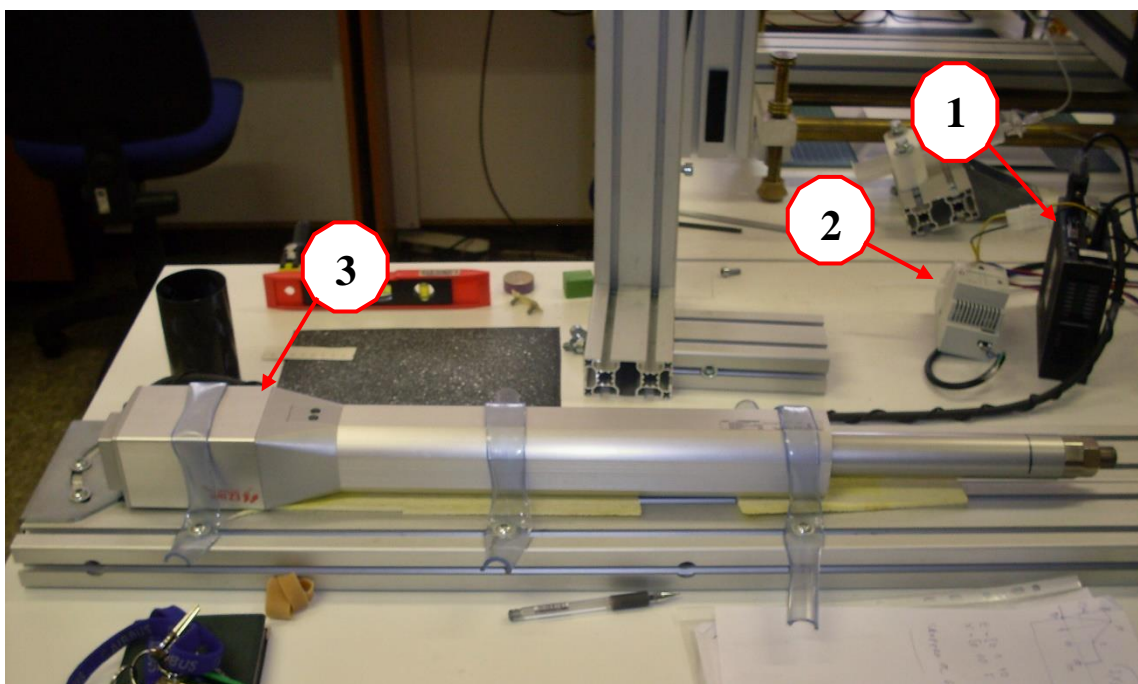
Jestliže by se obě křivky rozcházely o větší vzdálenost, než je velikost jednoho kroku, tak by docházelo k výše zmiňované ztrátě kroku motoru. V takovém případě je nutné v dané oblasti snížit rychlost.

Další možnou komplikací je překročení rozsahu osy. Tento problém se pozná tak, že se v grafu „Poloha“ vykreslovaná křivka ustálí na hodnotě 0,145 m, případně -0,145. To jsou limitní hodnoty osy, jejichž nepřekročení je zabezpečeno blokem *saturation*. Proto je potřeba změnit zadávané parametry trajektorie nebo nastavení relativní nuly.

5.2.2 Spuštění HW

Po úspěšné kontrole vstupních parametrů trajektorie lze simulaci ukončit a přistoupit k samotnému experimentu na ose. V této druhé etapě je již zapotřebí počítač propojený s controllorem pohonu lineární osy.

Nejprve je nutné zapnout měnič napětí, viz Obr. 26. Otevřít řídicí program *stepper15_6320_final_HW.slx* s modelem připojeným k HW portům. Poté vzít odzkoušená data z předchozí etapy a překopírovat je do inicializační funkce *InitFcn*. Důležité je po změně dat nezapomenout uložit soubor, uložení je indikováno zmizením hvězdičky u jména souboru.

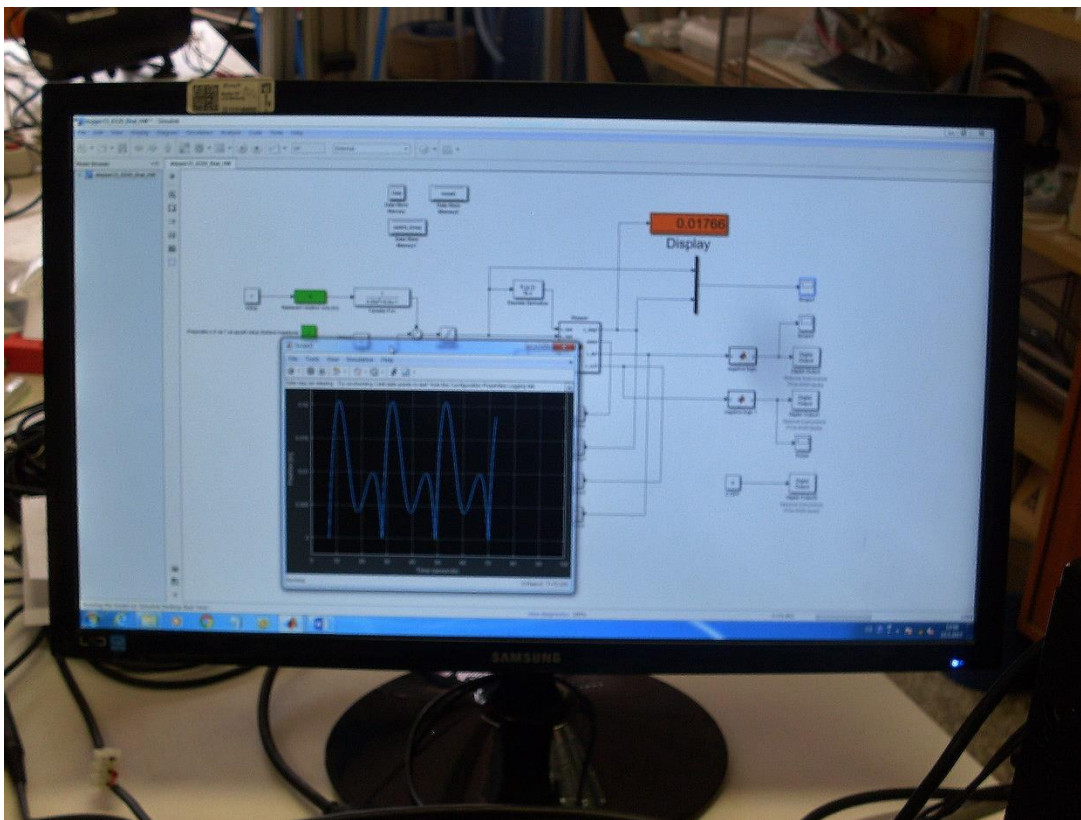


1. Controller; 2. Měnič AC/DC; 3. Osa EZ Limo

Obr. 26 - Sestavená osa

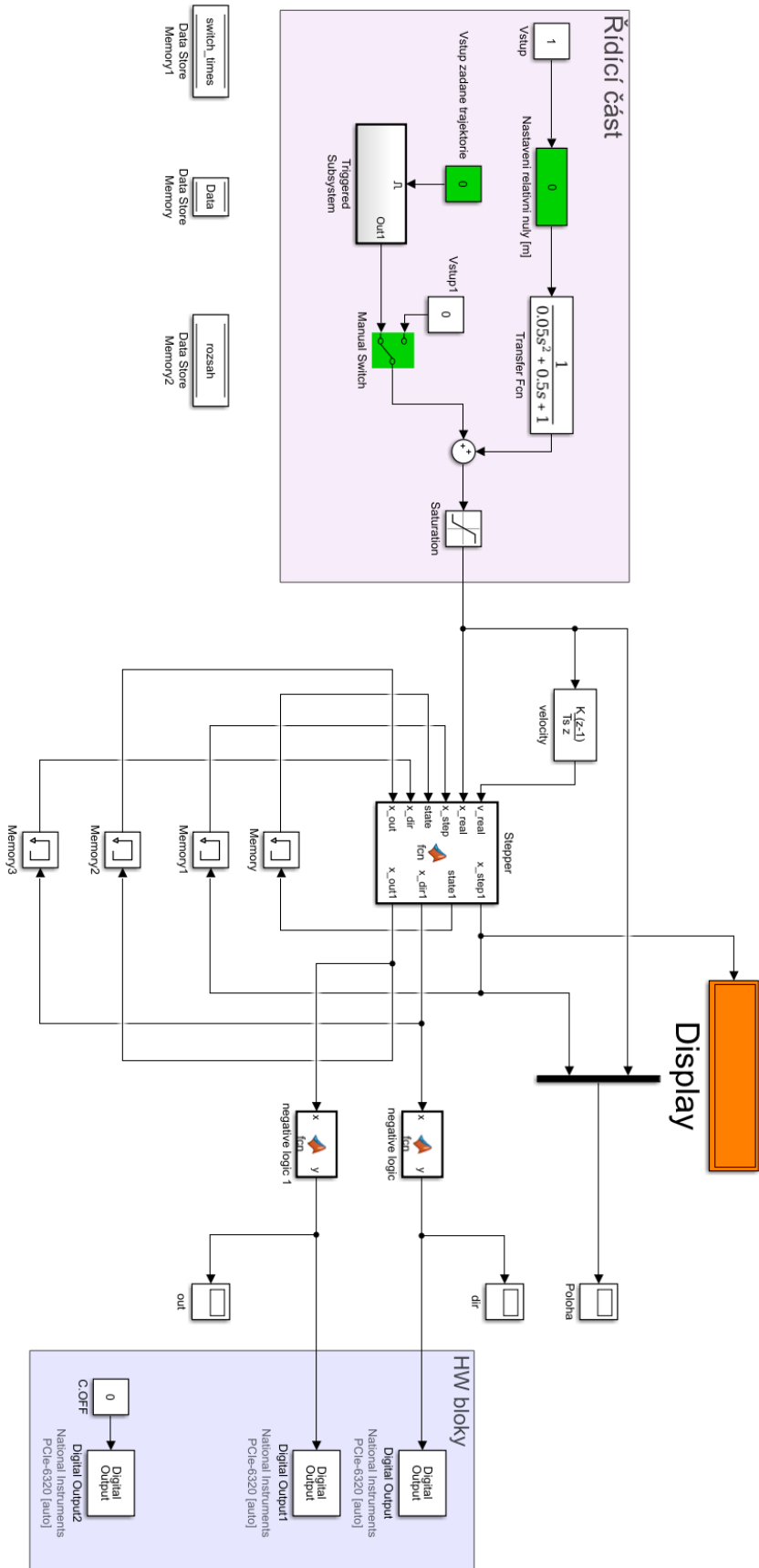
Po vložení vstupních dat je provedeno výchozí nastavení řídicí části modelu. Postup nastavování a spuštění simulace je totožný jako při přípravě dat. Po spuštění simulace lze opět změnit nastavení relativní nuly a stisknout Control+D, pro aktualizaci modelu. Změna se projeví na displeji v programu a je taktéž doprovázena pohybem osy do dané polohy. Následným povolením vstupu zadané trajektorie začne osa vykonávat pohyb po definované trajektorii.

Souřadnice aktuální polohy je zobrazována na displeji. Časový průběh trajektorie lze pozorovat v grafu „Poloha“, viz Obr 27. V případě potřeby je možno experiment pozastavit tlačítkem *Pause* v horní liště. Následně je možné opětovným stisknutím tlačítka *Run* experiment dokončit.



Obr. 27 - Průběh experimentu

Celý experiment i simulaci lze kdykoliv ukončit tlačítkem *Stop*. Ideální je provést ukončení v prostředku pohybového rozsahu, tedy na nule. To umožňuje snadné započetí nového experimentu.



Obr. 28 - Stepper15_6320_final_HW.slx

6 Závěr

V práci je popsán způsob vzniku řídicího programu pro pohon lineární osy na základě předdefinovaných trajektorií. V prvních kapitolách byly shromážděny informace o možných typech pohonů a o způsobu jejich řízení. Dále byly získány informace o mechanismech zajišťujících převod rotačního pohybu na pohyb posuvný. V dalších kapitolách následovalo seznámení se s prostředím, ve kterém bude řídicí program vznikat a postup sestavování tohoto programu.

V závěru práce bylo provedeno ověření funkčnosti navrženého programu pomocí experimentu s reálnou pohybovou osou. Během experimentu došlo k vykreslení požadované trajektorie a k reálnému pohybu řízené osy na základě této trajektorie. Tím byla funkčnost řídicího programu ověřena.

7 Literatura

- [1] UHLÍŘ, Ivan. *Elektrické stroje a pohony*. Vyd. 2. přeprac. Praha: ČVUT, 2007. ISBN 978-80-01-03730-0.
- [2] BORSKÝ, Václav. *Základy stavby obráběcích strojů*. Brno: Ediční středisko Vysokého učení technického, 1986.
- [3] LAŠOVÁ, Václava. *Základy stavby obráběcích strojů* [online]. Západočeská univerzita v Plzni, 2012 [cit. 2017-03-13]. ISBN 978-80-261-0126-0. Dostupné z: http://zcu.cz/export/sites/zcu/pracoviste/vyd/online/Zaklady_stavby.pdf
- [4] Polohovací systémy lineární osy. In: *Hiwin s.r.o. - Lineární technika, vedení, motory, pohony, kuličkové šrouby* [online]. [cit. 2017-03-18]. Dostupné z: www.hiwin.cz/media/files/05_Polohovaci_systemy_Linearni_osy.pdf
- [5] VUES Brno. *Lineární motory* [online]. [cit. 2014-8-22]. Dostupný na [www](http://www.vues.cz/file/424/CZ_LIN-OBECNE_020909.PDF): http://www.vues.cz/file/424/CZ_LIN-OBECNE_020909.PDF
- [6] Krokový motor. In: *PohonnaTechnika.cz* [online]. [cit. 2017-06-15]. Dostupné z: <http://www.pohonnatechnika.cz/skola/motory/krokovy-motor>
- [7] Servo motor. In: *PohonnaTechnika.cz* [online]. [cit. 2017-06-15]. Dostupné z: <http://www.pohonnatechnika.cz/skola/motory/servo-motor>
- [8] Elektrické servopohony. In: *Elektrické servopohony* [online]. [cit. 2017-06-15]. Dostupné z: https://www.vutbr.cz/www_base/priloha.php?dpid=33400
- [9] ServoBelt Linear Engineering Guide. In: *Bell-Everman, Motion Control Technologies* [online]. [cit. 2017-06-15]. Dostupné z: <http://bell-everman.com/products/linear-positioning/servobelt-linear-sbl>
- [10] Ezlimo_catalog.pdf. In: *EZ Limo - EZC Series* [online]. [cit. 2017-06-15]. Dostupné z: http://www.dynaservo.com/products/documents/ezlimo_catalog.pdf

8 Seznam obrázků

Obr. 1 - Srovnání lineární a klasický motor	9
Obr. 2 - Části lineárního motoru	10
Obr. 3 - Reluktanční krokový motor	12
Obr. 4 - Schéma hybridního krokového motoru	13
Obr. 5 - Komutovaný motor	15
Obr. 6 - Kuličkový šroub	18
Obr. 7 - Pastorek a hřeben	19
Obr. 8 - Princip fungování ServoBelt systému	20
Obr. 9 - Demonstrační program v Matlabu.....	21
Obr. 10 - Demonstrační model v Simulinku.....	22
Obr. 11 - Lineární osa EZ Limo	23
Obr. 12 - Zapojení řídicí jednotky	24
Obr. 13 - Přístup k vstupním datům.....	25
Obr. 14 - Definování základních parametrů	25
Obr. 15 - Ukázka strukturování vstupních dat.....	26
Obr. 16 - Subsystem s interpolátorem	27
Obr. 17- Ukázka z interpolator_simulink.m	27
Obr. 18 – Možné typy interpolace	28
Obr. 19 - Ukázka interpolovaných křivek	29
Obr. 20 - Funkce Stepper.....	30
Obr. 21 - Funkce motor(dir)	31
Obr. 22 - Ukázka změny směru motoru.....	31
Obr. 23 - Řídicí model	32
Obr. 24 - Řídicí část modelu.....	33
Obr. 25 - Kontrola trajektorie	34
Obr. 26 - Sestavená osa.....	35
Obr. 27 - Průběh experimentu.....	36
Obr. 28 - Stepper15_6320_final_HW.slx	37