

**Západočeská univerzita v Plzni**

**Fakulta aplikovaných věd**

**Katedra kybernetiky**

**BAKALÁŘSKÁ PRÁCE**

**Plzeň, 2014**

**Jaroslav Růžička**

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Akademický rok: 2013/2014

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jaroslav RŮŽIČKA**  
Osobní číslo: **A11B0674P**  
Studijní program: **B3902 Inženýrská informatika**  
Studijní obor: **Počítačové řízení strojů a procesů**  
Název tématu: **Návrh, modelování a řízení manipulátoru se šesti stupni volnosti**  
Zadávací katedra: **Katedra kybernetiky**

### Z á s a d y p r o v y p r a c o v á n í :

1. Navrhněte konstrukci manipulátoru se šesti stupni volnosti.
2. Navrhněte HW řídicího systému manipulátoru.
3. Vytvořte matematický model manipulátoru.
4. Navrhněte řídicí algoritmus manipulátoru.

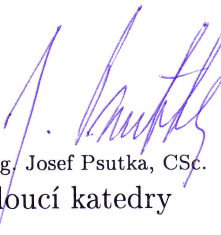
Rozsah grafických prací: **dle potřeby**  
Rozsah pracovní zprávy: **30-40 stránek A4**  
Forma zpracování bakalářské práce: **tištěná**  
Seznam odborné literatury:  
**Dodá vedoucí bakalářské práce.**

Vedoucí bakalářské práce: **Ing. Miroslav Flídr, Ph.D.**  
Katedra kybernetiky

Datum zadání bakalářské práce: **1. listopadu 2013**  
Termín odevzdání bakalářské práce: **16. května 2014**

  
Doc. Ing. František Vávra, CSc.  
děkan



  
Prof. Ing. Josef Psutka, CSc.  
vedoucí katedry

V Plzni dne 1. listopadu 2013

## PROHLÁŠENÍ

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 16. května 2015

.....  
podpis

## PODĚKOVÁNÍ

Chtěl bych poděkovat panu Ing. Miroslavu Flídřovi, Ph.D., za odborné vedení, za pomoc a rady při zpracování této práce.

# ANOTACE

## **Anotace:**

Cílem této práce je na základě poznatků sestavit paralelní manipulátor se šesti stupni volnosti. V této práci bude uvedeno vhodné konstrukční řešení a jeho realizace. Dále se autor bude věnovat oblasti použitelného hardwaru pro tento manipulátor, bude popsán použitelný pohon, měřící jednotka polohy a řídicí systém. Dále bude odvozen matematický model, který bude popisovat pohyb vrchní desky manipulátoru. Poslední část se bude zabývat návrhem algoritmu, který bude řídit manipulátor.

## **Abstract:**

The aim of this bachelor thesis is to build a parallel manipulator with six degrees of freedom based on the acquired knowledge. In this bachelor thesis appropriate structural solution and its realization will be described. Furthermore, the author will deal with the issue of applicable hardware for this manipulator, will describe the applicable actuator, measuring unit of position and control system. In the next section, mathematical model which will describe the movement of top manipulator plate will be derived. The last section will deal with the design of the algorithm that will control the manipulator.

## **Klíčová slova:**

paralelní robot, návrh paralelního robotu, model paralelního robotu, 6DoF, Stellaris Launchpad, IMU, krokový motor, ovladač krokového motoru

## **Key words:**

parallel robot, design of parallel robot, model of parallel robot, 6DoF, Stellaris Launchpad, IMU, stepper motor, driver stepper motor

# Obsah

<b>Obsah</b>	<b>7</b>
<b>1 Úvod</b>	<b>1</b>
<b>2 Konstrukce manipulátoru</b>	<b>2</b>
2.1 Manipulátor . . . . .	2
2.1.1 Základní pojmy . . . . .	2
2.2 Sériový manipulátor . . . . .	4
2.3 Paralelní manipulátor . . . . .	5
2.4 Stewartova plošina . . . . .	5
2.4.1 Popis Stewartovy plošiny . . . . .	6
2.4.2 Použité řešení . . . . .	7
2.4.3 Sestavení modelu . . . . .	8
<b>3 Matematický model</b>	<b>12</b>
3.1 Popis matematického modelu . . . . .	12
3.1.1 Matematické vztahy pro lineární pohon . . . . .	13
3.1.2 Převedení lineárního ramene na rotační rameno . . . . .	17
3.2 Matematický model v Simulinku . . . . .	18
<b>4 HW řídicího systému manipulátoru</b>	<b>23</b>
4.1 Krokový motor . . . . .	23
4.1.1 Obecný popis . . . . .	23
4.1.2 Ovládání krokového motoru . . . . .	24
4.1.3 Použitý krokový motor . . . . .	24
4.2 Řadič motoru . . . . .	26
4.3 Inerciální měřicí jednotka (IMU) . . . . .	27
4.3.1 Obecný popis . . . . .	28
4.3.2 Vybraná inerciální měřicí jednotka . . . . .	28
4.4 Řídicí systém . . . . .	29
4.4.1 Mikrokontrolér LM4F120H5QR . . . . .	30

4.4.2	Vstupy a výstupy vývojové desky . . . . .	31
4.5	Použité vstupy a výstupy . . . . .	32
4.5.1	SPI . . . . .	32
4.5.2	Serial UART komunikace . . . . .	33
4.5.3	I2C komunikace . . . . .	34
4.6	Návrh řídicího systému . . . . .	36
4.6.1	Řídicí hardware . . . . .	36
4.6.2	Řídicí software . . . . .	37
4.6.3	Vývojové prostředí pro Stellaris Launchpad . . . . .	38
4.6.4	algoritmus plošiny . . . . .	39
4.6.5	Algoritmus ovládání krokového motoru . . . . .	41
<b>5</b>	<b>Závěr</b>	<b>42</b>
	<b>Literatura</b>	<b>44</b>



# Kapitola 1

## Úvod

Tato bakalářská práce bude pojednávat o návrhu, modelování a řízení manipulátoru se šesti stupni volnosti těže označovaný jako Stewartova plošina po anglickém vědci D. Stewartovi, který ji jako první představil v roce 1954. V dnešní době ji lze též najít jako Gough/Stewartova plošina, jelikož současně se Stewartem ji popsal i americký vědec E. Gough.

Tento typ manipulátoru se též nazývá paralelní robot se šesti stupni volnosti. Jedná se o dvě desky, které jsou spolu specificky spojené pomocí šesti ramen. Základna má 6 bodů, ve kterých jsou napojena ramena, zatímco vrchní část má pouze 3 body, kde jsou ramena spojena. Jedná se o jednu z nejtěžších úloh v inverzní kinematice. Ramena tohoto robotu zpravidla tvoří lineární aktuátory, které pohybují s vrchní částí. Jelikož se lineární aktuátory nevyrábějí v laboratorním provedení pro malé plošiny, budou pohyb horní desky obstarávat krokové motory a jejich rotační pohyb bude převáděn na lineární posun jednotlivých ramen robotu.

Prvním úkolem bakalářské práce je vytvořit návrh takového paralelního robotu, který bude odpovídat definici. Dalším úkolem bude podle návrhu sestavit reálný model. Model musí být popsán vhodnými matematickými vztahy, které budou popisovat pohyb jednotlivých ramen a hlavně horní desky robotu.

Poté bude matematický model ověřen na simulačním modelu vytvořeném v programu Simulink. Tento simulační model bude odpovídat skutečnému modelu, který bude navrhnout a sestaven.

Poslední část bakalářské práce se bude věnovat nalezení vhodného řízení navrhovaného robota, tak aby bylo možné orientovat plošinu dle zadání od uživatele.

# Kapitola 2

## Konstrukce manipulátoru

V této části bude popsána konstrukce manipulátoru. Nejprve bude popsáno co to je manipulátor, jaké jsou možnosti manipulátoru. Dále se tato část bude věnovat zvolenému manipulátoru a jeho realizaci.

### 2.1 Manipulátor

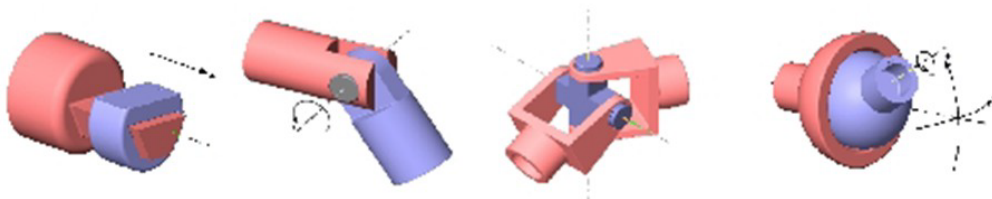
Manipulátor je jeden z mnoha druhů robotů, který usnadňuje práci člověku. Nejdříve vznikaly manipulátory mechanické např. pákové a kladkové mechanismy, později začaly vznikat tzv. elektromechanické manipulátory např. bagry a nakladače. V dnešní době se s rozvojem mechatroniky rozšířily manipulátory do mnoha odvětví, kde usnadňují práci člověku, nebo jej přímo nahrazují. Manipulátory můžeme najít v průmyslu, kde se jedná o skladovací manipulátory, manipulátory pracující na výrobní lince, svařovací manipulátory. Dále je můžeme využít pro průzkum oblastí, které jsou nebezpečné pro člověka (chemicky zamořené území, zaminované území), nebo jsou těžko přístupné či dokonce nedostupné pro člověka. V poslední době se manipulátory též uplatňují v medicíně, kde díky své přesnosti slouží u operací jako nástroje chirurga.

Manipulátory lze rozdělit do dvou skupin a to sériový manipulátor a paralelní manipulátor. Oba typy popíšu v následujících podkapitolách.

#### 2.1.1 Základní pojmy

V této části budou vysvětleny některé základní pojmy, které se budou objevovat v textu bakalářské práce. Tyto informace byly převzaty ze skript (Goubej et al.; 2012)

1. DoF - počet stupňů volnosti manipulátoru (DoF - Degree of Freedom). Tato vlastnost označuje minimální počet parametrů, které popisují polohu manipulátoru v rovině/prostoru. Pokud uvažuje manipulátor v rovině tak hovoříme o 3-DoF, jestliže v prostoru poté je to 6-DoF.  
Obecná poloha manipulátoru je určena jeho translací a rotací. Translace je pohyb, ve kterém jsou popsány souřadnice libovolného bodu manipulátoru v prostoru (souřadnice  $x$ ,  $y$  a  $z$ ). Rotace označuje natočení manipulátoru, lze ji vyjádřit pomocí rotační matice nebo Eulerovými úhly a jinými způsoby.
2. Základna manipulátoru - nepohyblivá část manipulátoru, na které je definován souřadný systém celého manipulátoru.
3. Koncový efektor - označuje buď poslední část manipulátoru, nebo poslední rameno, na kterém je umístěn pracovní nástroj manipulátoru, a je určen zobecněnými souřadnicemi manipulátoru. Toto bývá zpravidla úkolem kinematiky, aby se koncový efektor nacházel v požadované poloze.
4. Kinematický řetězec - je dán vazbou dvou ramen. Ramena jsou spolu spojena pomocí kloubů, které vymezují vzájemný pohyb ramen. Zpětná kinematická úloha je inverzní úloha, kdy máme zadány zobecněné souřadnice koncového efektoru a hledáme kloubové souřadnice jednotlivých kloubů.
5. Klouby - ty tvoří důležitou součást manipulátorů. Umožňují totiž jeho pohyb, který vychází z úlohy. Klouby mohou být aktivní nebo pasivní. Jako aktivním kloub označujeme takový, který má v sobě aktuátor. Pasivní kloub je bez aktuátoru. Nejčastěji se můžeme setkat se čtyřmi základními typy kloubů a to jsou: prizmatický, rotační, univerzální (kardanův), nebo sférický.



Obrázek 2.1: Klouby: 1) Prizmatický 2) Rotační 3) Univerzální 4) Sférický

6. Aktuátor - je pohony ramen. V dnešní době jsou dva hlavní typy, které se používají. První typ se používá častěji a je to tzv. rotační pohon např. krokové motory, serva. Druhý typ je tzv. lineární pohon např. elektrohydraulické válce, nebo přímočaré motory.

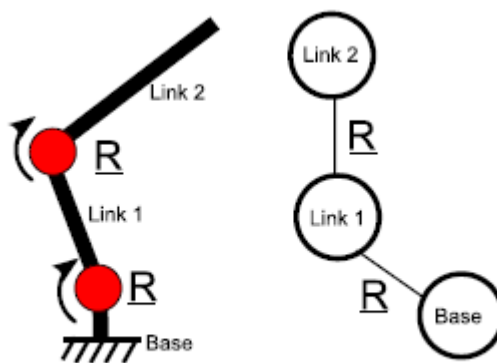
7. Přesnost - je určena rozdílem mezi požadovanou polohou a skutečnou polohou manipulátoru, nebo koncového efektoru. Nejčastěji se měří buď pomocí referenčního nastavení nebo kalibračním senzorem.
8. Opakovatelnost - je vlastnost manipulátoru. Okamžik, kdy je manipulátor schopný dostat se do požadované polohy, která je stejná pro jeho různé počáteční polohy.

## 2.2 Sériový manipulátor

Sériový manipulátor je popsán otevřeným kinematickým řetězcem. To znamená, že každé rameno je spojeno klouby s dalšími dvěma rameny vyjma základny a koncového efektoru. Jedná se o nejrozšířenější roboty v praxi, kde jsou uplatňováni v průmyslu např. svařovací roboty v automobilovém průmyslu.

Výhody sériového manipulátoru jsou zejména v jednoduchosti návrhu, kdy lze využít snadnější mechaniku pro popis pohybu a zároveň celkový popis je snazší než u paralelního manipulátoru. Další výhodou je pracovní prostor, který je velký podle velikosti a počtu ramen manipulátoru.

Nevýhody manipulátoru jsou, že manipulátor musí být velmi robustní, jelikož každé rameno musí unést váhu nákladu na koncovém efektoru, ale i celou váhu robotu. Proto musí být manipulátor robustní a tím ztrácí na dynamice. Další nevýhodou je nepřesnost a neschopnost přesné opakovatelnosti, jelikož chyby vzniklé v jednom rameni se nasčítávají do dalších ramen a tím klesá jeho přesnost a opakovatelnost. Další nevýhodou je umístění aktuátorů, které musí být v každém kloubu, a tím se také zvyšuje zatížení a omezuje dynamiku robotu.



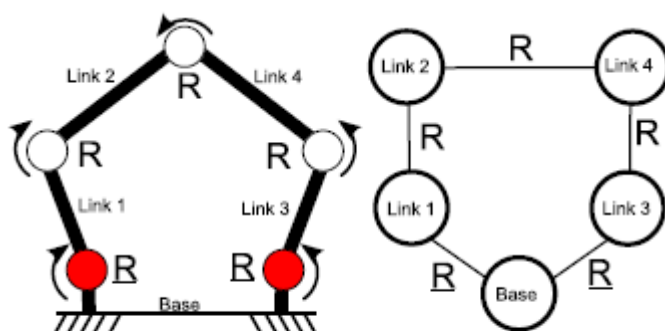
Obrázek 2.2: Sériový manipulátor

## 2.3 Paralelní manipulátor

Paralelní manipulátor je popsán uzavřeným kinematickým řetězcem. Platí, že vždy minimálně dvě ramena jsou spojena v jednom bodě manipulátoru. Dříve byl spíše jako akademická záležitost, ale v dnešní době se též aplikují v průmyslu např. aktivní tlumení vibračních desek.

Výhody paralelního manipulátoru spočívají v tom, že má větší užitnou hodnotu, kdy se váha rozloží rovnoměrně po celé konstrukci, a není nutné, aby každé rameno neslo celou váhu manipulátoru a nákladu. Dále se zpravidla aktuátory umísťují na základu, kde jsou s ní pevně spojeny a pohybují ramena, která už jsou pouze mechanické. Chyby vzniklé v návrhu jsou průměrovány, a proto je lepší opakovatelnost a přesnost manipulátoru.

Nevýhody jsou, že je manipulátor popsán složitější mechanickou vazbou, kdy lze řešit pouze inverzní geometrickou úlohu a nelze řešit přímou geometrickou úlohu. Dále je razantně omezen pracovní prostor manipulátoru.



Obrázek 2.3: Paralelní manipulátor

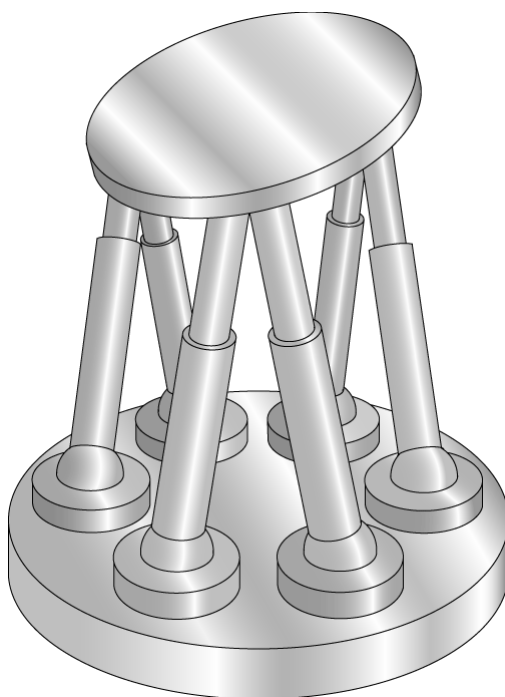
## 2.4 Stewartova plošina

Jak už bylo zmíněno v úvodu, Stewartova plošina je jeden z mnoha typů paralelního manipulátoru. Jedná se o paralelní manipulátor, který má 6 - DoF a tudíž toto řešení vyhovuje zadání bakalářské práce a bude realizované.

### 2.4.1 Popis Stewartovy plošiny

Jedná se o nejnámější variantu paralelního manipulátoru. Lze se setkat i s označením Stewartova/Gough plošina, ačkoliv Gough tento manipulátor vymyslel už v roce 1954 pro testování pneumatik a Stewart představil svůj paralelní manipulátor až v roce 1964, jako letecký simulátor, tak se Stewart označuje častěji, jako autor tohoto řešení.

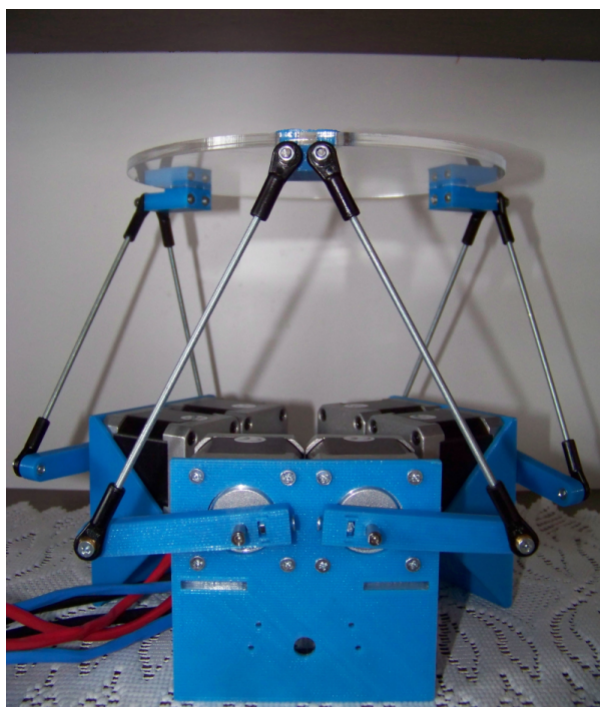
Koncept Stewartovi plošiny lze rozdělit na tři části. První část je základna, která je nepohyblivá a jsou na ní umístěny aktuátory. Aktuátory jsou na základně rozmístěny po dvojicích a rozděleny pod úhlem  $120^\circ$ . Druhá část jsou samotné aktuátory, které jsou běžně tvořeny lineárními aktuátory. Tich je celkem šest a jsou umístěny, tak aby bylo dosaženo všech požadovaných DoF. Poslední část tvoří vrchní deska, které se pohybuje, dle zadání od uživatele, je kruhová a jsou na ni opět připevněny aktuátory po dvojicích v úhlu  $120^\circ$ , ale v jiném páru, než jsou na základně.



Obrázek 2.4: Stewartova plošina

## 2.4.2 Použité řešení

Při návrhu řešení Stewartovi plošiny musel jako první být zvolen vhodný typ aktuátoru (viz část 2.1.1). Pro tento typ by bylo vhodné použít lineární aktuátory. Bohužel se tyto aktuátory v dnešní době nedají sehnat za přijatelnou finanční částku, proto musel být zvolen alternativní aktuátory. Na webových stránkách Marginally Clever (Marginally Clever; 2014) je řešení této úlohy pomocí krokových motorů. Toto řešení se zdálo jako vhodné nahrazení lineárních aktuátorů. Změna aktuátorů, respektive náhrada lineárních aktuátorů za rotační aktuátory znamená, že se musí najít vhodné vztahy, které převedou pohyb přímočarý na pohyb rotační. Dále z těchto stránek bylo čerpáno při vlastním návrhu modelu, který je velmi podobný. Jelikož autor poskytuje svobodně ke stažení výkres celého modelu. Tento model byl nakonec modifikován, byly zvoleny odlišné krokové motory, které by nešly osadit do tohoto modelu, proto došlo ke změně dílů. Celá základna modelu včetně ramen a držáků horního dílu byla vytištěna na 3D tiskárně. Vrchní díl byl vyhotoven v externí firmě, jelikož 3D tiskárna by nedokázala vytisknout celý díl v jednom kuse.

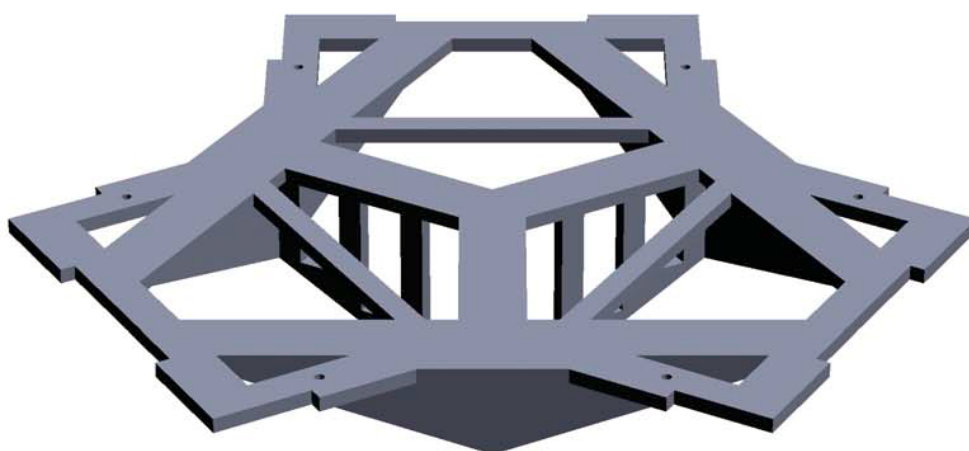


Obrázek 2.5: Kompletně sestavený manipulátor.

### 2.4.3 Sestavení modelu

V této části budou popsány jednotlivé díly, ze kterých je složen celkový model manipulátoru. Jak již bylo zmíněno výše téměř celý model byl vytištěn na 3D tiskárně. Model je složen z několika základních dílů, které jsou spolu spojeny a tvoří tak celek.

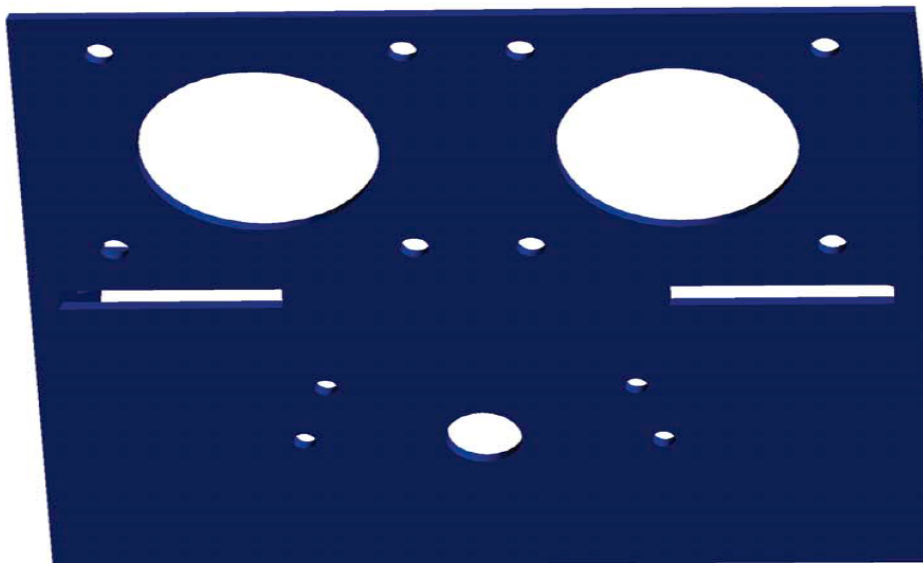
Jak už bylo zmíněno v předchozí kapitole, model není úplně totožný s tím, který udělal autor na stránkách Marginally Clever, ale je upravený tak, aby se do něj vešly motory, které jsou použité v této práci.



Obrázek 2.6: Středový díl

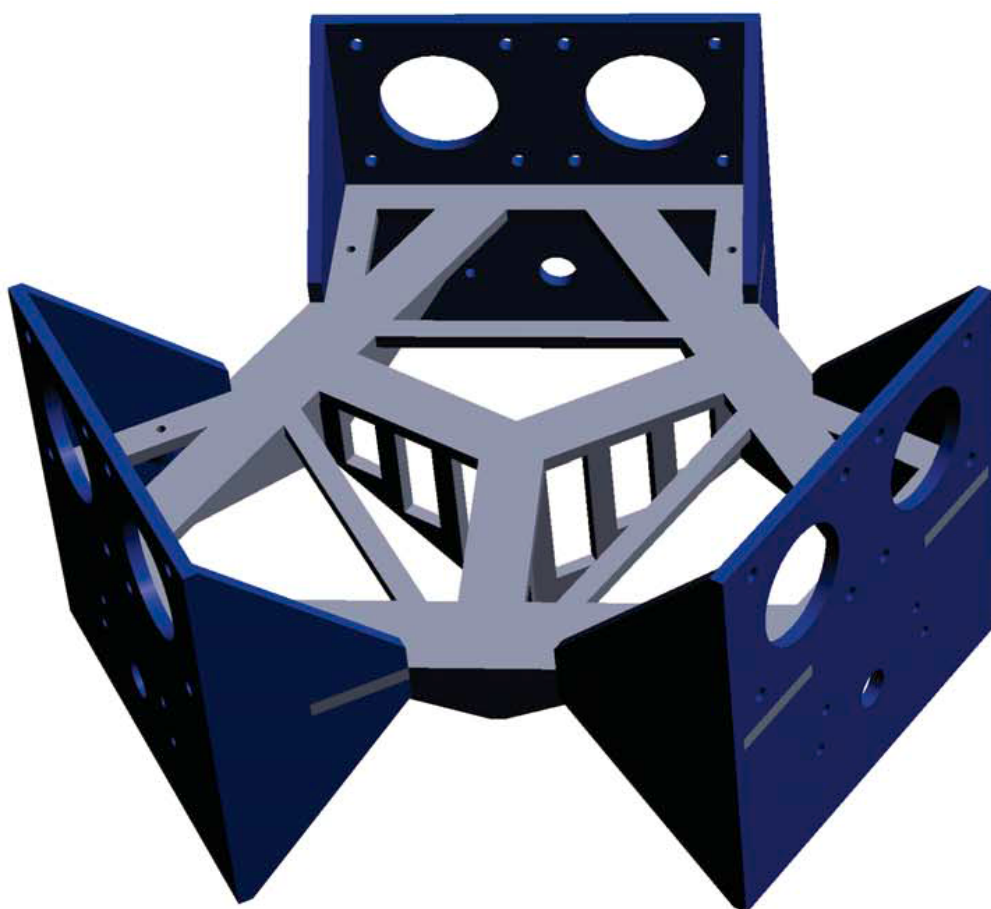
Na obrázku (obrázek 2.6) je vidět středový díl základny, který je kostrou celé základny. K tomuto dílu jsou vruty připevněny tři boční části. Dále tento díl slouží jako držák motorů. Ve středu tohoto dílu je střed vztažné soustavy pro určování polohy manipulátoru.





Obrázek 2.7: Bočnice

Na obrázku (obrázek 2.7) je vidět jedna bočnice. V bočnici jsou připravené otvory, kterými jsou vyvedeny rotory motorů. V každé bočnici jsou připevněny dva motory. Motory jsou k bočnicím připevněny pomocí šroubů. V dolní části bočnic jsou připraveny úchyty pro koncová tlačítka, které zatím nejsou realizované, ale pokud budou nutná, budou dodělaná v navazující práci.



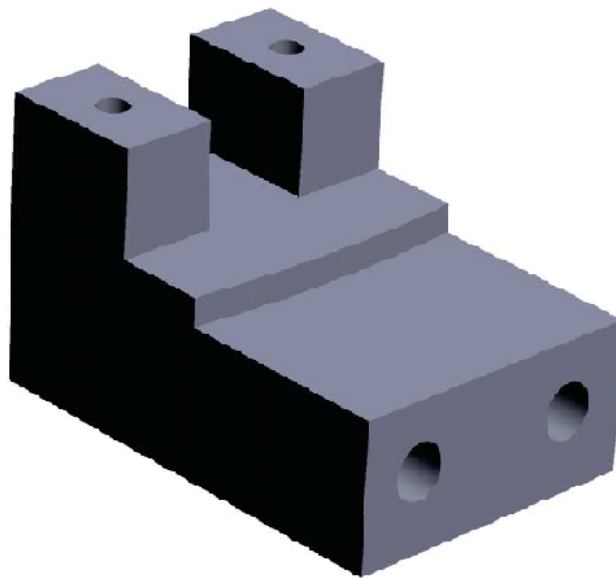
Obrázek 2.8: Kompletní základna

Na tomto obrázku (obrázek 2.8) je vidět vizualizace kompletní základny, které se skládá ze čtyř dílů, kostry a třech bočnic, které slouží pro uchycení krokových motorů.



Obrázek 2.9: Rameno rotoru

Na tomto obrázku (obrázek 2.9) je vidět rameno, které se otáčí podle natočení rotoru krokového motoru a tím hýbe plošinou. Na koncích ramen jsou umístěny kulové čepy.



Obrázek 2.10: Držák vrchní desky

Na obrázku (obrázek 2.10) je vidět uchycení k horní desce, toto uchycení je k vrchní desce přišroubované. Zároveň držák slouží k připojení ramen základny. Toto spojení je realizováno pomocí kulových čepů. Dále k sestavení plošiny byly použity modelářské kulové čepy, které slouží ke spojení ramene krokového motoru a vrchní desky. Tyto čepy jsou spojeny závitovou tyčí, jak je vidět na obrázku (obrázek 2.5)

## Kapitola 3

# Matematický model

Tato část bakalářské práce se bude věnovat odvození vhodného matematického modelu, který bude popisovat pohyb vrchní desky manipulátoru. Tento matematický model bude sestaven též v simulačním programu Simulink. Toto simulační schéma bude vypočítávat, jak se mají otočit jednotlivé krokové motory, tak aby plošina byla v požadované poloze.

### 3.1 Popis matematického modelu

V této části budou popsány matematické vztahy, které popisují pohyb plošiny. Matematické vztahy lze v obecném případě odvozovat dvěma způsoby: přímou geometrickou úlohou, kdy se podle kloubových souřadnic vypočítává pozice koncového efektoru, nebo pomocí inverzní geometrické úlohy, kdy je známa poloha koncového efektoru a dopočítávají se kloubové souřadnice. Tuto úlohu lze řešit pomocí inverzní geometrické úlohy. Pomocí přímé geometrické úlohy ji nelze řešit, jelikož je to tak komplikovaný systém, tak že nelze najít analytické řešení.

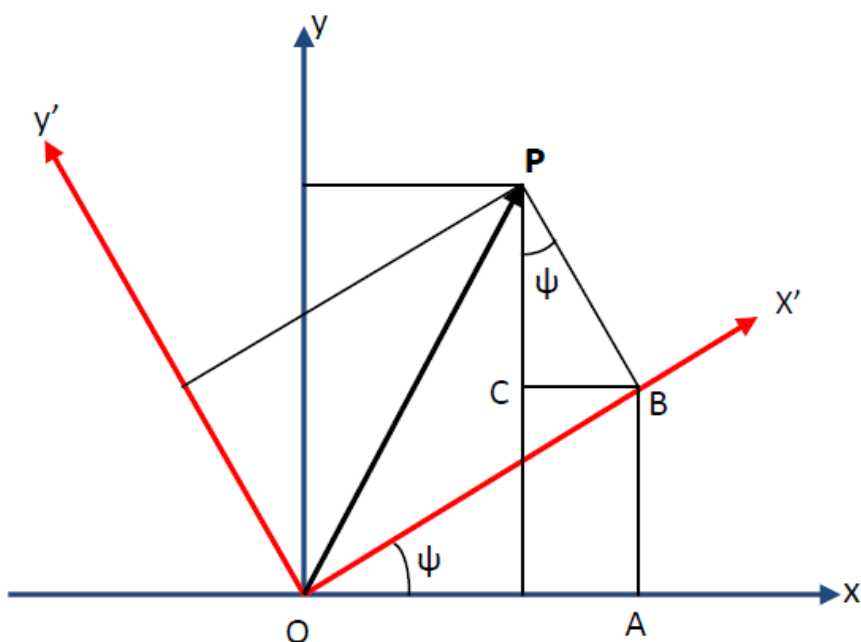
Tato část bude rozdělena na dvě části. V první části budou odvozeny vztahy pro lineární aktuátory Stewartovy plošiny. Druhá část se bude zabývat převodem lineárního pohonu na rotační.

### 3.1.1 Matematické vztahy pro lineární pohon

Pro výpočty jsou zavedeny následující předpoklady:

- plošina je složena ze dvou desek, které jsou spolu spojeny celkem 6 rameny
- základna je popsána referenční soustavou souřadnic  $x, y, z$ , která je ve středu plošiny.
- vrchní desce je přiřazena vlastní soustava souřadnic  $x', y'$  a  $z'$ , která je uprostřed vrchní desky
- původní souřadnice základny lze pomocí 3 transformačních matic vyjádřit v souřadnicích vrchní desky
- rotace okolo třech os vrchní desky vůči základně lze vyjádřit následujícími třemi Eulerovskými úhly:
  - rotace o úhel  $\Psi$  okolo osy  $z$  se nazývá vybočení
  - rotace o úhel  $\Theta$  okolo osy  $y$  se nazývá stoupání
  - rotace o úhel  $\Phi$  okolo osy  $x$  se nazývá vytočení

Nyní budou popsány rotace, které transformují souřadnice bodu z jednoho souřadného systému do druhého soustavného systému.



Obrázek 3.1: Znázornění transformace úhlů z jedné soustavy souřadnic z do soustavy souřadnic  $z'$ .

Na obrázku jsou vidět dvě soustavy souřadnic, které jsou vůči sobě posunuté o úhel  $\Psi$ . Na obrázku jsou vidět body, které slouží pro výpočet transformace z jedné souřadné soustavy do druhé. Bod O je střed soustav, bod P je bod, který má své souřadnice jak v soustavě  $x, y$  tak v soustavě  $x', y'$ , body C, B, P jsou body trojúhelníku, které popisují bod P v soustavě  $x', y'$ , a bod A slouží k určení velikosti natočení soustavy souřadnic  $x', y'$  od soustavy  $x, y$ . Nyní musíme najít vhodné vztahy pro popis souřadnic bodu P v souřadném systému  $X', Y'$ .

$$P = i'x' + j'y' + k'z' = ix + jy + kz \quad (3.1)$$

Tato rovnice říká, že bod P lze vyjádřit v soustavě  $x, y$  a  $x', y'$  pomocí jejich vektorových složek  $i, j, k$  a  $i', j', k'$ . Nyní musíme najít vztahy pro  $x, y$  a  $z$ .

Dále lze vyjádřit  $x, y$  a  $z$ :

$$x = OA - BC = x' \cos \Psi - y' \sin \Psi \quad (3.2)$$

$$y = AB + PC = x' \sin \Psi + y' \cos \Psi \quad (3.3)$$

$$z = z' \quad (3.4)$$

Nyní pro přehlednost bude zavedena matice rotace  $R_z(\Psi)$ , používá se toto ozna-

čení, které říká, že jde o rotaci kolem osy z o úhel  $\Psi$ :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_z(\Psi) \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad (3.5)$$

kde

$$R_z(\Psi) = \begin{pmatrix} \cos \Psi & -\sin \Psi & 0 \\ \sin \Psi & \cos \Psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.6)$$

Stejným výpočtem lze dojít k matici rotace kolem osy y o úhel  $\Theta$ , která se označuje  $R_y(\Theta)$ :

$$R_y(\Theta) = \begin{pmatrix} \cos \Theta & 0 & \sin \Theta \\ 0 & 1 & 0 \\ -\sin \Theta & 0 & \cos \Theta \end{pmatrix} \quad (3.7)$$

Stejným výpočtem lze dojít k matici rotace kolem osy x o úhel  $\Phi$ , která se označuje  $R_x(\Phi)$ :

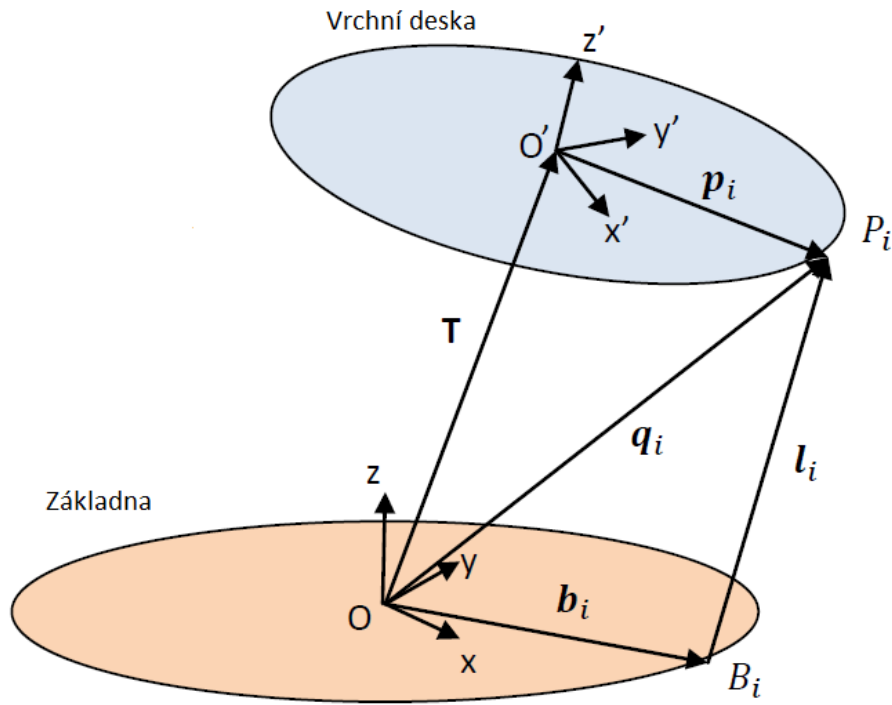
$$R_x(\Phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \Phi & -\sin \Phi \\ 0 & \sin \Phi & \cos \Phi \end{pmatrix} \quad (3.8)$$

Nyní pomocí odvozených matic rotace pro každou osu (3.6) – (3.8) lze určit matici  ${}^P R_B$ , která popisuje kompletní transformaci z jedné soustavy souřadnic do druhé soustavy souřadnic.

$$\begin{aligned} {}^P \mathbf{R}_B &= \mathbf{R}_z(\Psi) * \mathbf{R}_y(\Theta) * \mathbf{R}_x(\Phi) = & (3.9) \\ & \begin{pmatrix} \cos \Psi & -\sin \Psi & 0 \\ \sin \Psi & \cos \Psi & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} \cos \Theta & 0 & \sin \Theta \\ 0 & 1 & 0 \\ -\sin \Theta & 0 & \cos \Theta \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \Phi & -\sin \Phi \\ 0 & \sin \Phi & \cos \Phi \end{pmatrix} \\ &= \begin{pmatrix} \cos \Psi \cos \Theta & -\sin \Psi \cos \Theta & \sin \Theta \\ \sin \Psi \cos \Theta & \cos \Psi \cos \Theta & \sin \Psi \sin \Theta \\ -\sin \Theta & 0 & \cos \Theta \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \Phi & -\sin \Phi \\ 0 & \sin \Phi & \cos \Phi \end{pmatrix} \\ &= \begin{pmatrix} \cos \Psi \cos \Theta & -\sin \Psi \cos \Phi + \cos \Psi \sin \Theta \sin \Phi & \sin \Psi \sin \Phi + \cos \Psi \sin \Theta \cos \Phi \\ \sin \Psi \cos \Theta & \cos \Psi \cos \Phi + \sin \Psi \sin \Theta \sin \Phi & -\cos \Psi \sin \Phi + \sin \Psi \sin \Theta \cos \Phi \\ -\sin \Theta & \cos \Theta \sin \Phi & \cos \Theta \cos \Phi \end{pmatrix} \end{aligned}$$

Nyní, když jsou definovány matice rotace v každé ose z jednoho souřadného

systemu do druhého souřadného systému, bude představen popis Stewartovy plošiny.



Obrázek 3.2: Popis i-tého ramene Stewartovi plošiny

Na tomto obrázku (obrázek 3.2) je vidět, jak se vypočítá délka ramene  $l_i$ . Na obrázku jsou vidět významné body, které slouží k výpočtu:  $b_i$  je vektor vzdálenosti mezi středem soustavy souřadnic základny a přípojného bodu  $B_i$ ,  $B_i$  je vektor přípojného bodu ramene na základně,  $T$  je matice posunutí vrchní desky od základny,  $P_i$  je přípojný bod ramene na vrchní desce, vektor  $q_i$  určuje vzdálenost přípojného bodu  $P_i$  od středy soustavy souřadnic základny, vektor  $p_i$  je vektor, který určuje vzdálenost mezi přípojným bodem  $P_i$  a soustavou souřadnic označenou  $O'$ . Vektor  $q_i$  je vzdálenost bodu  $P_i$  od počátku vztahné soustavy  $O$ . Vektor  $q_i$  vyjadřuje následující rovnice:

$$q_i = T + {}^P R_B \cdot p_i \quad (3.10)$$

Kde  $T$  označuje vektor translace, který udává poziční posunutí vrchní desky vůči základně. Vektor  $p_i$  definuje souřadnice od středu soustavy souřadnic vrchní



desky a přípojného bodu  $P_i$ .  
Podobně lze určit vztah pro  $l_i$  rameno:

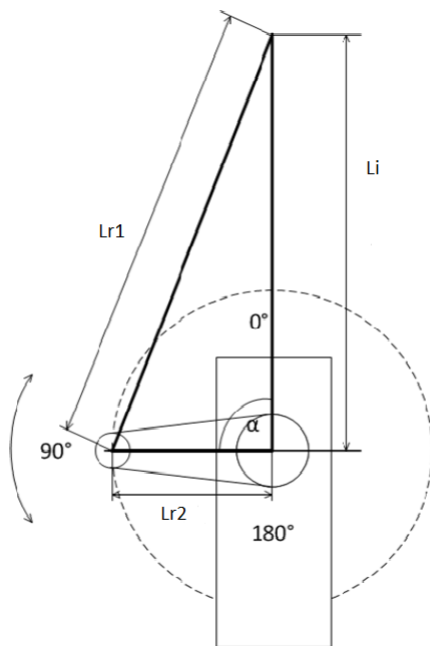
$$l_i = T + {}^P R_B \cdot p_i - b_i \quad (3.11)$$

Kde  $b_i$  je vektor, který definuje dolní přípojný bod  $B_i$ . Těchto 6 rovnic dohromady dávají délky všech 6 ramen pro požadovanou pozici a natočení vrchní desky. Tyto rovnice byly převzaty ze zdroje: (Wokingham U3A; 2011)

### 3.1.2 Převod lineárního ramene na rotační rameno

Tato část bude věnována odvození rovnic pro jednotlivá ramena Stewartovi plošiny, kdy je jako aktuátor použit rotační pohyb.

Pokud jsou délky ramen měněny pomocí rotačního pohonu, probíhá převod z lineárního posunu na rotační za použití kosinové věty. Na následujícím obrázku je uveden postup.



Obrázek 3.3: Převod lineárního pohybu na rotační.

Pohon je tvořen krokovým motorem, které se může otáčet o  $360^\circ$ , ale v této aplikaci není využit celý jeho rozsah, jelikož rozsah závisí na možné otáčení ramena. Rameno  $L_{r2}$  se může otáčet o úhel  $\alpha$  v rozmezí od  $0^\circ$  do  $180^\circ$  a má svoji pevně danou velikost. Délka spojovacího ramene  $L_{r1}$  je též pevně dána,

délka ramene  $L_i$  je známa z inverzní kinematické úlohy. Zbývá určit úhel natočení  $\alpha$  rotačního pohonu, který se určí analyticky pomocí Kosinovy věty, podle následující rovnice:

$$L_{r1}^2 = L_{r2}^2 + L_i^2 - 2L_iL_{r2} \cdot \cos \alpha \quad (3.12)$$

úhel  $\alpha$

$$\alpha = \arccos \frac{-L_{r1}^2 + L_{r2}^2 + L_i^2}{2L_iL_{r2}} \quad (3.13)$$

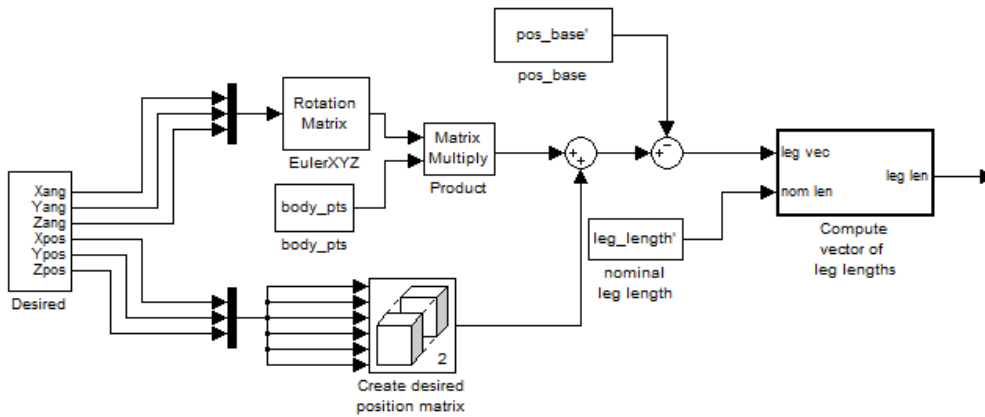
Tento výpočet je aplikován na všech 6 lineárních pohybech a pomocí něj jsou tyto pohyby převedeny na rotační pohyb, kde potřebuje znát úhel natočení. Tento výpočet lze použít jak pro servo pohony, tak i pro krokové motory.

## 3.2 Matematický model v Simulinku

V této části bude popsána implementace výpočtu natočení krokových motorů dle zadání uživatele pomocí simulačního modelu v programu Simulink. Simulink je nástavba programu MATLAB od společnosti MathWorks (The MathWorks, Inc.; 1994-2014), ve kterém se díky sestavování blokových schémat modelují a simulují různé dynamické systémy, a který při výpočtech využívá algoritmy MATLABu pro lineární i nelineární výpočty diferenciálních rovnic.

Při tvorbě simulačního modelu bylo využito již hotového schématu pro výpočet Stewartovy plošiny (Wendlandt; 2002), kde jsou použity lineární motory a bylo doplněné o bloky, které slouží pro přepočítání na pohyb pomocí krokového motoru. Výhodou bylo, že stačilo upravit nastavovací skript v MATLABu, kde se změnilo pouze rozměry plošiny, tak aby odpovídali modelu, který byl vytvořen v této práci.

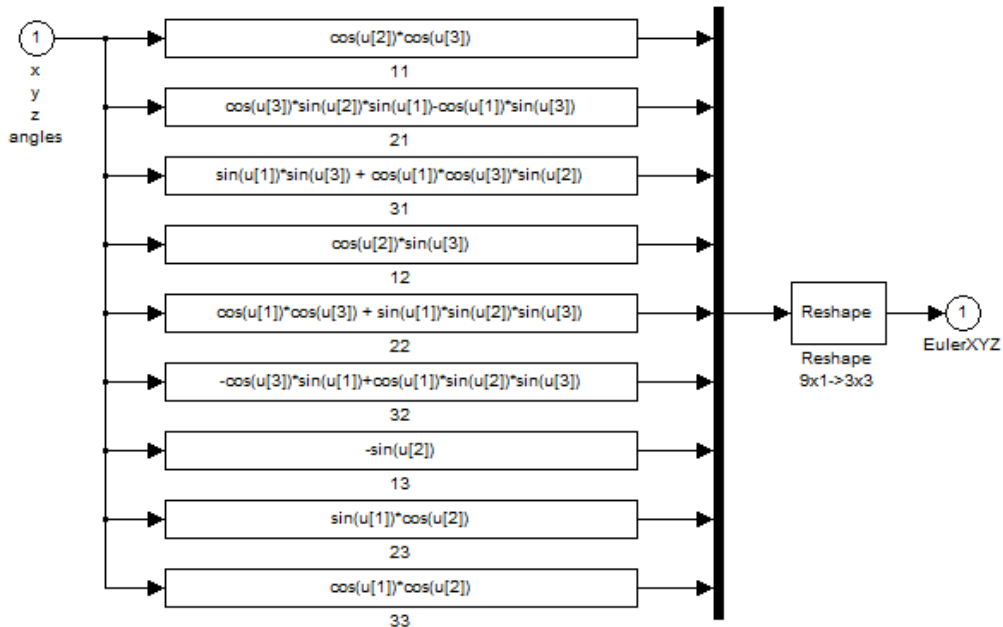
Nyní budou podrobně popsány jednotlivé bloky simulačního modelu.



Obrázek 3.4: Simulační schéma výpočtu lineárního pohybu

Na tomto obrázku je vidět schéma, ve kterém probíhá výpočet lineárního pohybu ramen plošiny, který vychází z odvozených vztahů. Nejprve se nastaví požadovaná poloha horní desky. Jednotlivé úhly vstupují do bloku EULERXYZ, v tomto bloku probíhá výpočet matice rotace, dle odvozených vztahů (3.9). Tato vypočtená matice je následně vynásobena v bloku Matrix Multiply s vektorem bodů plošiny. V bloku Create desired position matrix se vytváří matice translace horní desky. Oba tyto signály se sečtou a odečte se od nich pomocí bloku pos\_base vektor základny a tím získáme vektory  $q$  (3.10). Poté následuje výpočet délky jednotlivých lineárních ramen (3.11).

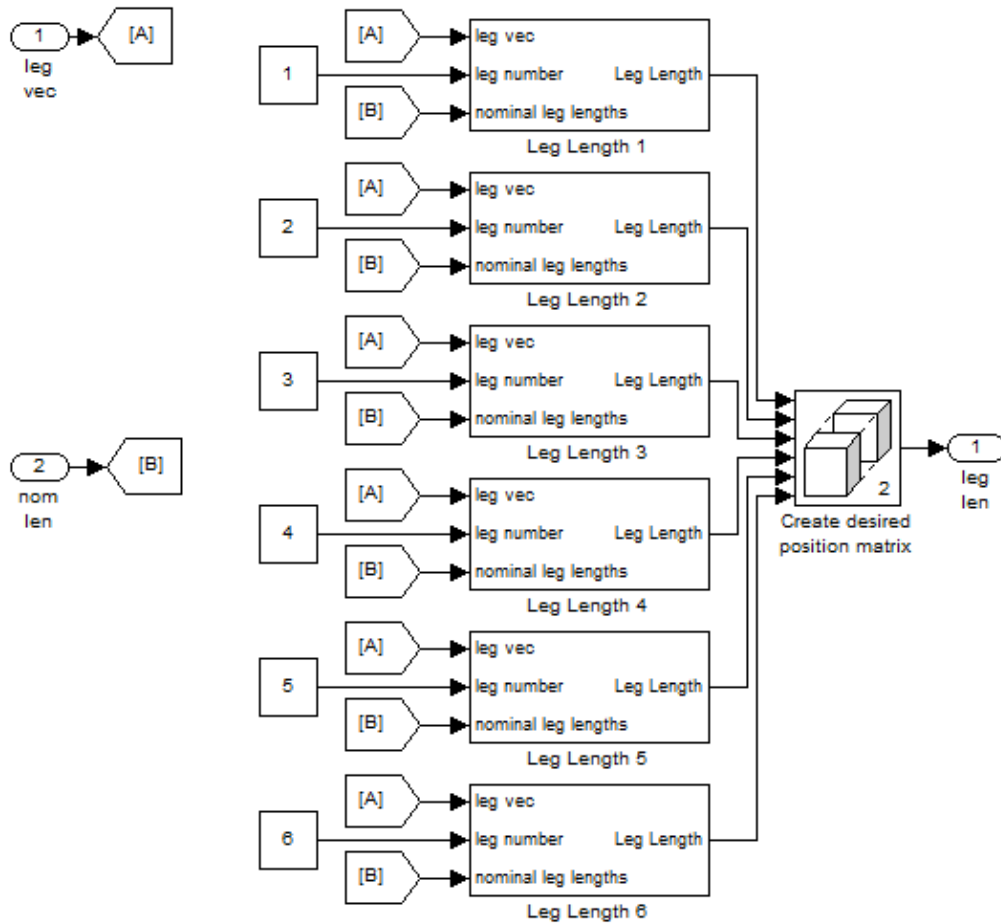
Nyní bude popsán subsystém EULERXYZ, který je vidět na následujícím obrázku.



Obrázek 3.5: Schéma výpočtu matice rotace

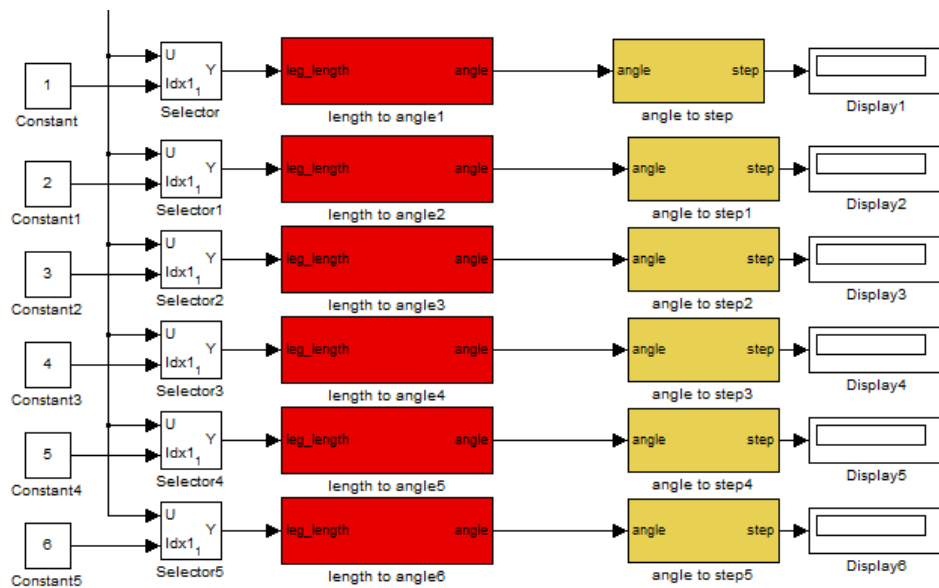
V tomto subsystému jsou výpočty prováděny pomocí bloků Fcn, které dokáží pomocí funkcí pracovat se vstupními signály. Každý řádek odpovídá jednomu prvku v matici rotace (3.9), která byla odvozena v předchozí části. Pro přehlednost bylo těchto bloků použito 9 a každý blok představuje jeden prvek v matici. Výsledky se poté pomocí bloku MUX spojí do jedné matice, která má rozměr  $9 \times 1$ . Tento rozměr neodpovídá odvozené matici, proto je použit blok Reshape, který z této matice udělá matici  $3 \times 3$ , kterou požadujeme dle odvození.

Poté následuje výpočet délky každého lineárního ramene v bloku Compute vector of leg lengths podle vztahů (3.11). Tyto výpočty nelze provést najednou a každé rameno se musí vypočítat zvlášť a poté jsou převedeny na matici  $1 \times 6$ .



Obrázek 3.6: Schéma výpočtu délek ramen

Poté co jsou vypočteny lineární délky ramen dochází k přepočtu na počet kroků každého krokového motoru zvlášť. Tyto přepočty opět probíhají pomocí odvozených vztahů (3.12) a (3.13). Nejprve se vypočítá v subsystému length to angle úhel natočení ramene. Poté tento výsledek přechází do subsystému angle to step, kde se vypočítaný úhel vydělí úhlem jednoho kroku. Díky tomuto bloku je získán počet kroků, o které se musí krokový motor otočit, aby byl v požadované poloze.



Obrázek 3.7: Schéma výpočtů kroků jednotlivých krokových motorů

Tento matematický model se v simulačním programu Simulink nechová vždy přesně tak, jak autor očekává. Možná příčina neuspokojivého chování je v tom, že je chybně nastavený model ve výchozí pozici. Z toho důvodu bude nutné v navazující práci lépe zadat vstupní parametry nebo upravit matematické vztahy, které popisují pohyb vrchní desky plošiny.

## Kapitola 4

# HW řídicího systému manipulátoru

Tato kapitola se bude věnovat popisu realizovaného modelu Stewartovy plošiny. Budou popsány krokové motory, řadiče krokových motorů, inerciální řídicí jednotka a jako poslední bude popsáno zařízení pro ovládání řadičů krokových motorů.

### 4.1 Krokový motor

Tato část se bude věnovat krokovému motoru, který je použit jako aktuátor pro jednotlivá ramena Stewartovy plošiny. Stejně tak se bude věnovat i obecnému popisu a ovládání krokového motoru, poté bude podrobně popsán krokový motor, který byl zvolen při řešení.

#### 4.1.1 Obecný popis

Krokový motor je synchronní točivý stroj, který je většinou napájen impulsy stejnosměrného proudu. Magnetické pole, které otáčí rotorem motoru, je generováno postupně, tak jak jsou napájené jednotlivé pólové dvojice. Rotor krokového motoru se pohybuje vždy v určitém úhlu mezi stabilními polohami tzv. pohyb v krocích. Počet kroků je dán počtem pólových dvojic, také jej ovlivňuje způsob ovládání. Krokové motory se dají řídit jen za použití řídicí elektroniky. Výhodou krokových motorů je, že k mechanickému kontaktu dochází pouze v ložiscích, proto mají krokové motory při správném zacházení velkou životnost a jsou téměř bezúdržbové. Krokový motor má v podstatě jen jedinou nevýhodu, která lze velmi snadno eliminovat vhodným zvolením krokového motoru a ovladače krokového motoru. Tato negativní vlastnost je označována jako ztráta kroku, která nastává v situaci, kdy je překročeno mezní zatížení udávané výrobcem, nebo kdy uživatel zadá příliš vysoké otáčky, které krokový motor nezvládá. Poté

dochází k tomu, že některé kroky motoru jsou vynechány a krokový motor začne mechanicky zakmitávat. Tento stav vede k poškození až zničení krokového motoru. (Daewoo; 2009)

### **4.1.2 Ovládání krokového motoru**

Jak již bylo zmíněno v části 4.1.1 pro pohyb krokového motoru je třeba použít speciální elektronické zařízení, které dokáže generovat impulsy v určitém sledu a délce. Tyto impulsy následně přes výkonovou část budí v přesném pořadí jednotlivé vinutí rotoru. Frekvence, pořadí a délka impulsů z ovládacího obvodu řídí počet, smysl otáček rotoru (otáčení hřídele buď podle směru hodinových ručiček, nebo proti směru otáčení ručičkových hodinek) a rovněž točivý moment stroje. Řadič se dá řídit buď zadáním kroku, nebo úhlu kroku. V současnosti lze ovládací obvod sestavit nebo koupit již hotové od mnoha výrobců.

### **4.1.3 Použitý krokový motor**

Z velkého množství výrobců krokových motorů byla vybrána firma Pololu, která má prodejní zastoupení i v České republice, proto není problém objednat jejich výrobky. Jejich krokové motory jsou velmi oblíbené zejména díky ceně a vlastnostem, proto se s nimi lze setkat v dnešní době velmi oblíbených 3D tiskárnách, dále se používají i v CNC strojích, ale i k dalším použitím.

Pro realizaci bakalářské práce byl zvolen krokový motor s označením Pololu 1200 (Pololu Corporation; 2001-2014b). Tento krokový motor se vyznačuje tím, že lze ovládat buď jako unipolární, nebo jako bipolární. Unipolární motor znamená, že v daném okamžiku prochází proud pouze jednou cívkou, což znamená, že má menší odběr, ale též menší kroutící moment. Bipolární motor znamená, že v daný okamžik proud prochází dvěma cívkami, které jsou protilehlé. Firma Pololu doporučuje tento motor používat jako bipolární, protože je to jednodušší při ovládání, jelikož pokud by se používal jako unipolární mohlo by dojít k poškození motoru.





Obrázek 4.1: Krokový motor Pololu1200

Další výhodou tohoto motoru je jeho cena, které je velmi příznivá v porovnání s ostatními výrobci. Další nespornou výhodou je kompaktní rozměr, kterého ne každý výrobce dosahuje. Tento motor dokáže pracovat s velkou zátěží na hřídeli, aniž by se poškodil, toto bylo zohledněno při výběru. Další výhodou je, že má 200 kroků na otočení, což je poměrně dost. Jeden krok je otočení o  $1.8^\circ$ , a to je ideální pro použití ve Stewartově plošině.

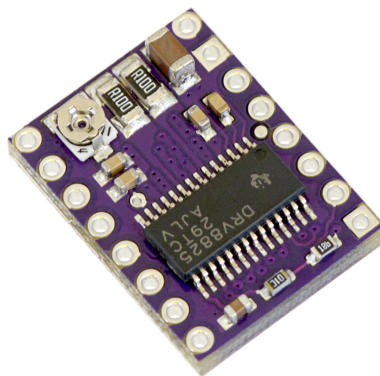
Specifikace:

Velikost:	42.3 × 48 mm
Váha:	350 g
Počet kroků na otočení:	200
Ampérová zátěž:	1.2 A na cívice
Jmenovité napětí:	4.0 V
Odpor:	3.3 Ohm na cívice
Zátěž:	3.2 kg na cm
Pracovní délka ramene:	30 cm

## 4.2 Řadič motoru

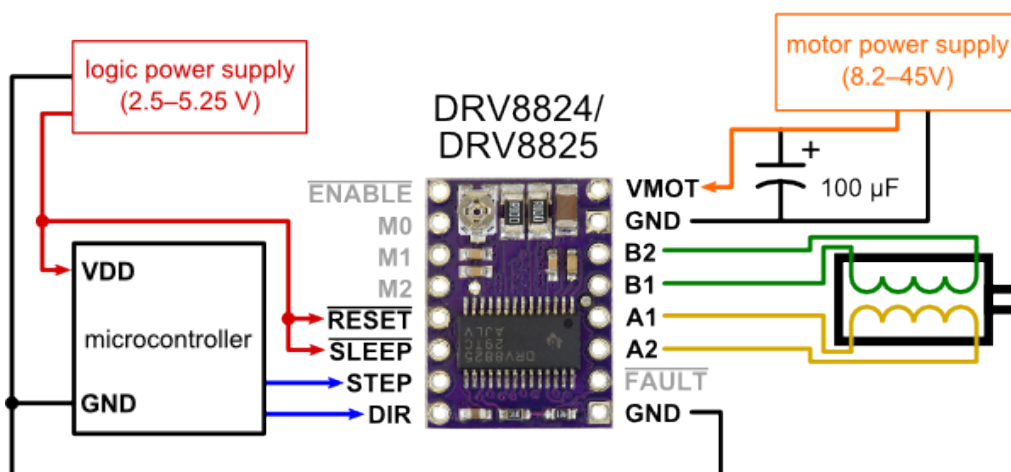
Jak už bylo popsáno v části 4.1.2, pro ovládání krokového motoru lze využít jak vlastní, tak už sestavené a hotové obvody. V této bakalářské práci bude využit již hotový ovládací obvod od firmy Pololu, jelikož sestavení vlastního obvodu je časově náročné a bylo by to velmi komplikované, zároveň by to přesahovalo rámec znalostí autora.

Ke zvolenému krokovému motoru byl vybrán řadič DRV8825, který slouží pro ovládání bipolárních motorů a výrobce čipu je společnost Texas Instruments (Texas Instruments; 2012a). Řadič DRV8825 disponuje 6 mikro kroky, které umožňují zpřesnit krok rotoru, pokud to aplikace vyžaduje. Řadič je schopný pracovat s výstupním napětím od 8.2V do 45 V a s proudem až 1.5 A bez chlazení a nebo s chlazením až do 2.2A. Další výhodou je, že pracuje na logické úrovni od 2.5V do 5.25V tj. lze ho použít i při řízení pomocí 3.3V.



Obrázek 4.2: Řadič DRV8825

Nyní bude popsáno použité zapojení, které je vidět na obrázku níže. Nejprve bude popsána pravá část obrázku. Napájení je přivedeno na piny VMOT a GND přes kondenzátor o hodnotě 100  $\mu\text{F}$ . Tento kondenzátor slouží k tomu, aby při prvním spuštění nedošlo k tzv. špičkovému výkyvu napájení, které by tento řadič nevydržel. Dále jsou použity piny B2, B1, A1 a A2, které ovládají připojený krokový motor. Poslední použitý pin má označení GND a je na něj přivedena zem z řídicího systému. V levé části obrázku jsou vidět piny, které slouží k samotnému ovládání krokového motoru. V levé části jsou piny, které v této práci nejsou využity a jsou na obrázku označeny šedou barvou. Tyto piny se používají v případě, pokud aplikace vyžaduje použít místo plného kroku mikrokrok. Piny *RESET* a *SLEEP* pracují v negované logice. V tomto zapojení to znamená, že je stále přivedena hodnota označená jako HIGH, které je použita, proto aby se řadič neuspával a neresetoval během ovládání krokových motorů. Piny STEP a DIR slouží k samotnému ovládání krokového motoru podle signálu z řídicího systému. Zapojení bylo převzato ze stránky výrobce řadiče (Pololu Corporation; 2001-2014a).



Obrázek 4.3: Zapojení řadiče

### 4.3 Inerciální měřicí jednotka (IMU)

Tato část se bude věnovat IMU, která bude použita pro určení polohy horní desky plošiny. Popis IMU je uveden v této práci pro úplnost popisu vytvořeného modelu Stewartovy plošiny. Nicméně její využití je naplánované v další práci.

### 4.3.1 Obecný popis

Inerciální měřicí jednotka, též známá pod zkratkou IMU je elektronické zařízení, které je možné použít k vypočítávání rychlosti, orientace a orientace magnetického pole. IMU v sobě používá tři základní prvky: akcelerometr, gyroskop a magnetometr. IMU se typicky používá v letectví k určení polohy, využívá se pro řízení bezpilotního letadla. V dnešní době lze najít IMU i v GPS navigacích, kdy slouží k odhadu polohy zařízení v místech, kde není signál GPS např. tunely. IMU se využívá i při navádění řízených střel a bylo ho použito např. v projektu Apollo vesmírné společnosti NASA (Mor; 2005).

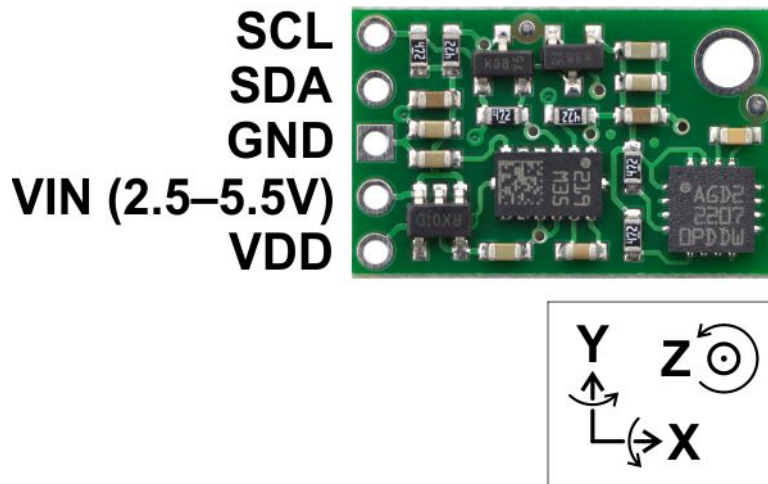
### 4.3.2 Vybraná inerciální měřicí jednotka

Pro řešení byla zvolena inerciální měřicí jednotka od společnosti Pololu s označením MinIMU-9 v2<sup>1</sup>, která obsahuje gyroskop, akcelerometr a magnetometr. Jedná se o jednotku, která má 9-DoF tj. dokáže změřit polohu, rychlost a působení magnetické síly ve všech třech osách. Tento modul byl zvolen, jelikož je schopný pracovat s logickým napětím o hodnotě 3.3V, které bude používat i řídicí systém. Moduly, které pracují s vyšším napětím by nepracovaly korektně, nebo by nepracovaly vůbec. Pro komunikaci se využívá sériová sběrnice I<sup>2</sup>C.

Specifikace:

Napájení:	2.5 - 5.0 V
Proud:	10 mA
Výstupní formát I2C:	
Gyro:	16 bitů pro každou osu v každém čtení
Akcelerometr:	12 bitů pro každou osu v každém čtení
Magnetometr:	12 bitů pro každou osu v každém čtení
Rozsahy citlivosti (nastavitelné):	
Gyro:	$\pm 250, \pm 500$ , nebo $\pm 2000^\circ/s$
Akcelerometr:	$\pm 2, \pm 4, \pm 8$ , nebo $\pm 16$ g
Magnetometr:	$\pm 1.3, \pm 1.9, \pm 2.5, \pm 4.0, \pm 4.7, \pm 5.6$ , nebo $\pm 8.1$ gausse

<sup>1</sup>Pololu - MinIMU-9 v2 Gyro, Accelerometer, and Compass (L3GD20 and LSM303DLHC Carrier)[online].c2013 [cit. 26. dubna 2014]. Dostupné na WWW: <<http://www.pololu.com/product/1268>>

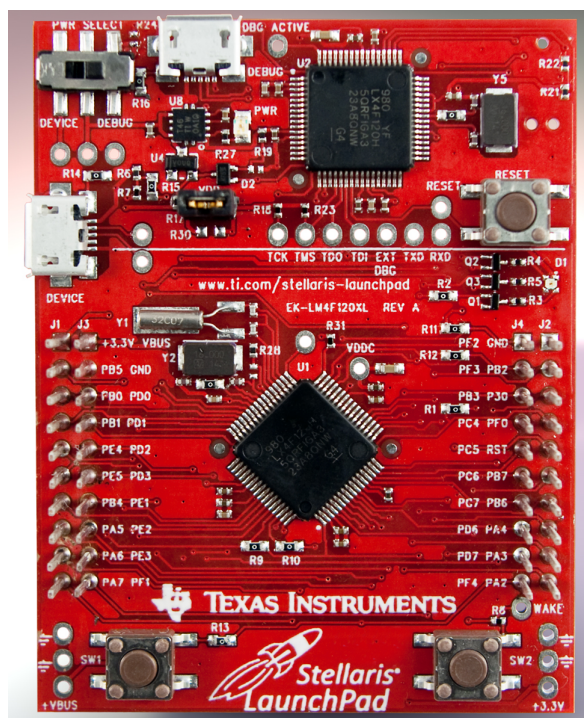


Obrázek 4.4: minIMU

## 4.4 Řídící systém

Tato část se bude věnovat řídicímu systému od firmy Texas Instruments, který slouží pro ovládání všech ovladačů krokových motorů a bude zpracovávat signál z IMU.

Jako řídicí systém je využita vývojová platforma na bázi mikrořadiče s jádrem ARM Cortex - M4, jedná se konkrétně o produkt Stellaris Launchpad (Texas Instruments; 2012b). Tato vývojová deska má spoustu výhod, proto byla použita v této bakalářské práci. Hlavní výhodou je velké množství pinů, které lze použít buď jako vstupy, nebo výstupy, které mohou být použité jako: analogové, digitální, nebo pro I<sup>2</sup>C. Další výhodou je integrace dvou tlačítek přímo na vývojové desce. Níže bude popsán mikrokontrolér vývojové desky a dále budou popsány použité a použitelné vstupy/výstupy vývojové desky.



Obrázek 4.5: Vývojová deska

#### 4.4.1 Mikrokontrolér LM4F120H5QR

Mikrokontrolér LM4F120H5QR je srdcem vývojové desky. Jedná se o čip, který má 64 pinů. Mikroprocesor má takt 80Mhz je to 32-bitový procesor a umí provádět aritmetické výpočty s plovoucí řádkou. Celkem má 256KB Flash / 32KB SRAM / 2Kb EEPROM paměti. Dále obsahuje duální 12-bitový ADC převodník, 6x64 bitové a 6x32 bitové časovače. Nyní budou popsány některé klíčové vlastnosti mikrokontroléru.

Mikrokontrolér podporuje synchronní a asynchronní sériovou komunikaci pomocí následujících sběrnic:

- CAN 2.0 A/B kontrolér
- USB 2.0 zařízení
- 8 UART portů s IrDA, 9. bitové a ISO7816 podporou
- 4 I<sup>2</sup>C moduly se čtyřmi přenosovými rychlostmi včetně vysokorychlostního přenosu

- 4 synchronní sériové zařízení moduly

## 4.4.2 Vstupy a výstupy vývojové desky

Jak už bylo zmíněno vývojová deska má velké množství V/V pinů, které lze řídit v různých režimech. Celkem se jedná o čtyři řady pinů, kde každá řada má celkem deset použitelných pinů. Vývojová deska jich má tolik, protože společnost Texas Instruments nabízí širokou škálu nadstavbových desek pro tuto vývojovou desku označené jako Boost Packs. Velké množství pinů je důležité a je to velká výhoda např. oproti vývojovým deskám Arduino, které mají méně použitelných pinů. V této bakalářské práci je nutno použít velké množství pinů, neboť samotné ovladače krokových motorů vyžadují celkem 12 pinů, dále je potřeba použít 2 piny na přijímání dat z IMU a další 2 piny na komunikaci s výpočetní zařízení, které se bude řešit v navazující práci.

**Stellaris LaunchPad with LM4F120H5QR**  
Revision 1

**Legend:**

- Hardware: Pin number, Other pin number
- PC: Serial UART, SPI
- analogRead()
- digitalRead(), digitalWrite() and analogWrite()

**Header J1/J3 Pin Configuration:**

Header	Pin	Function
CS (2)	A11	PB.5
RX (1)	PB.0	3
RX (2)	SDA (0)	A0
RX (3)	SDA (1)	A3
SCK (2)	A10	PA.5
SCK (1)	SDA (1)	PA.7
TX (1)	PB.1	4
TX (2)	SDA (0)	A3
TX (3)	SDA (1)	A3
MISO (1)	SDA (1)	PA.5
MISO (2)	SDA (1)	PA.7
MISO (3)	SDA (1)	PA.5
MISO (4)	SDA (1)	PA.7
TX (7)	SDA (1)	PA.5
TX (8)	SDA (1)	PA.7
TX (9)	SDA (1)	PA.5
TX (10)	SDA (1)	PA.7
TX (11)	SDA (1)	PA.5
TX (12)	SDA (1)	PA.7
TX (13)	SDA (1)	PA.5
TX (14)	SDA (1)	PA.7
TX (15)	SDA (1)	PA.5
TX (16)	SDA (1)	PA.7
TX (17)	SDA (1)	PA.5
TX (18)	SDA (1)	PA.7
TX (19)	SDA (1)	PA.5
TX (20)	SDA (1)	PA.7
TX (21)	SDA (1)	PA.5
TX (22)	SDA (1)	PA.7
TX (23)	SDA (1)	PA.5
TX (24)	SDA (1)	PA.7
TX (25)	SDA (1)	PA.5
TX (26)	SDA (1)	PA.7
TX (27)	SDA (1)	PA.5
TX (28)	SDA (1)	PA.7
TX (29)	SDA (1)	PA.5
TX (30)	SDA (1)	PA.7
TX (31)	SDA (1)	PA.5
TX (32)	SDA (1)	PA.7
TX (33)	SDA (1)	PA.5
TX (34)	SDA (1)	PA.7
TX (35)	SDA (1)	PA.5
TX (36)	SDA (1)	PA.7
TX (37)	SDA (1)	PA.5
TX (38)	SDA (1)	PA.7
TX (39)	SDA (1)	PA.5
TX (40)	SDA (1)	PA.7
TX (41)	SDA (1)	PA.5
TX (42)	SDA (1)	PA.7
TX (43)	SDA (1)	PA.5
TX (44)	SDA (1)	PA.7
TX (45)	SDA (1)	PA.5
TX (46)	SDA (1)	PA.7
TX (47)	SDA (1)	PA.5
TX (48)	SDA (1)	PA.7
TX (49)	SDA (1)	PA.5
TX (50)	SDA (1)	PA.7
TX (51)	SDA (1)	PA.5
TX (52)	SDA (1)	PA.7
TX (53)	SDA (1)	PA.5
TX (54)	SDA (1)	PA.7
TX (55)	SDA (1)	PA.5
TX (56)	SDA (1)	PA.7
TX (57)	SDA (1)	PA.5
TX (58)	SDA (1)	PA.7
TX (59)	SDA (1)	PA.5
TX (60)	SDA (1)	PA.7
TX (61)	SDA (1)	PA.5
TX (62)	SDA (1)	PA.7
TX (63)	SDA (1)	PA.5
TX (64)	SDA (1)	PA.7
TX (65)	SDA (1)	PA.5
TX (66)	SDA (1)	PA.7
TX (67)	SDA (1)	PA.5
TX (68)	SDA (1)	PA.7
TX (69)	SDA (1)	PA.5
TX (70)	SDA (1)	PA.7
TX (71)	SDA (1)	PA.5
TX (72)	SDA (1)	PA.7
TX (73)	SDA (1)	PA.5
TX (74)	SDA (1)	PA.7
TX (75)	SDA (1)	PA.5
TX (76)	SDA (1)	PA.7
TX (77)	SDA (1)	PA.5
TX (78)	SDA (1)	PA.7
TX (79)	SDA (1)	PA.5
TX (80)	SDA (1)	PA.7
TX (81)	SDA (1)	PA.5
TX (82)	SDA (1)	PA.7
TX (83)	SDA (1)	PA.5
TX (84)	SDA (1)	PA.7
TX (85)	SDA (1)	PA.5
TX (86)	SDA (1)	PA.7
TX (87)	SDA (1)	PA.5
TX (88)	SDA (1)	PA.7
TX (89)	SDA (1)	PA.5
TX (90)	SDA (1)	PA.7
TX (91)	SDA (1)	PA.5
TX (92)	SDA (1)	PA.7
TX (93)	SDA (1)	PA.5
TX (94)	SDA (1)	PA.7
TX (95)	SDA (1)	PA.5
TX (96)	SDA (1)	PA.7
TX (97)	SDA (1)	PA.5
TX (98)	SDA (1)	PA.7
TX (99)	SDA (1)	PA.5
TX (100)	SDA (1)	PA.7

**Header J4/J2 Pin Configuration:**

Header	Pin	Function
CS (1)	BLUE_LED	PF.2
CS (2)	GREEN_LED	PF.3
RX (1)	SDA (0)	PB.3
RX (2)	SDA (1)	PB.4
RX (3)	SDA (2)	PB.5
RX (4)	SDA (3)	PB.6
RX (5)	SDA (4)	PB.7
RX (6)	SDA (5)	PB.8
RX (7)	SDA (6)	PB.9
RX (8)	SDA (7)	PB.10
RX (9)	SDA (8)	PB.11
RX (10)	SDA (9)	PB.12
RX (11)	SDA (10)	PB.13
RX (12)	SDA (11)	PB.14
RX (13)	SDA (12)	PB.15
RX (14)	SDA (13)	PB.16
RX (15)	SDA (14)	PB.17
RX (16)	SDA (15)	PB.18
RX (17)	SDA (16)	PB.19
RX (18)	SDA (17)	PB.20
RX (19)	SDA (18)	PB.21
RX (20)	SDA (19)	PB.22
RX (21)	SDA (20)	PB.23
RX (22)	SDA (21)	PB.24
RX (23)	SDA (22)	PB.25
RX (24)	SDA (23)	PB.26
RX (25)	SDA (24)	PB.27
RX (26)	SDA (25)	PB.28
RX (27)	SDA (26)	PB.29
RX (28)	SDA (27)	PB.30
RX (29)	SDA (28)	PB.31
RX (30)	SDA (29)	PB.32
RX (31)	SDA (30)	PB.33
RX (32)	SDA (31)	PB.34
RX (33)	SDA (32)	PB.35
RX (34)	SDA (33)	PB.36
RX (35)	SDA (34)	PB.37
RX (36)	SDA (35)	PB.38
RX (37)	SDA (36)	PB.39
RX (38)	SDA (37)	PB.40
RX (39)	SDA (38)	PB.41
RX (40)	SDA (39)	PB.42
RX (41)	SDA (40)	PB.43
RX (42)	SDA (41)	PB.44
RX (43)	SDA (42)	PB.45
RX (44)	SDA (43)	PB.46
RX (45)	SDA (44)	PB.47
RX (46)	SDA (45)	PB.48
RX (47)	SDA (46)	PB.49
RX (48)	SDA (47)	PB.50
RX (49)	SDA (48)	PB.51
RX (50)	SDA (49)	PB.52
RX (51)	SDA (50)	PB.53
RX (52)	SDA (51)	PB.54
RX (53)	SDA (52)	PB.55
RX (54)	SDA (53)	PB.56
RX (55)	SDA (54)	PB.57
RX (56)	SDA (55)	PB.58
RX (57)	SDA (56)	PB.59
RX (58)	SDA (57)	PB.60
RX (59)	SDA (58)	PB.61
RX (60)	SDA (59)	PB.62
RX (61)	SDA (60)	PB.63
RX (62)	SDA (61)	PB.64
RX (63)	SDA (62)	PB.65
RX (64)	SDA (63)	PB.66
RX (65)	SDA (64)	PB.67
RX (66)	SDA (65)	PB.68
RX (67)	SDA (66)	PB.69
RX (68)	SDA (67)	PB.70
RX (69)	SDA (68)	PB.71
RX (70)	SDA (69)	PB.72
RX (71)	SDA (70)	PB.73
RX (72)	SDA (71)	PB.74
RX (73)	SDA (72)	PB.75
RX (74)	SDA (73)	PB.76
RX (75)	SDA (74)	PB.77
RX (76)	SDA (75)	PB.78
RX (77)	SDA (76)	PB.79
RX (78)	SDA (77)	PB.80
RX (79)	SDA (78)	PB.81
RX (80)	SDA (79)	PB.82
RX (81)	SDA (80)	PB.83
RX (82)	SDA (81)	PB.84
RX (83)	SDA (82)	PB.85
RX (84)	SDA (83)	PB.86
RX (85)	SDA (84)	PB.87
RX (86)	SDA (85)	PB.88
RX (87)	SDA (86)	PB.89
RX (88)	SDA (87)	PB.90
RX (89)	SDA (88)	PB.91
RX (90)	SDA (89)	PB.92
RX (91)	SDA (90)	PB.93
RX (92)	SDA (91)	PB.94
RX (93)	SDA (92)	PB.95
RX (94)	SDA (93)	PB.96
RX (95)	SDA (94)	PB.97
RX (96)	SDA (95)	PB.98
RX (97)	SDA (96)	PB.99
RX (98)	SDA (97)	PB.100
RX (99)	SDA (98)	PB.101
RX (100)	SDA (99)	PB.102

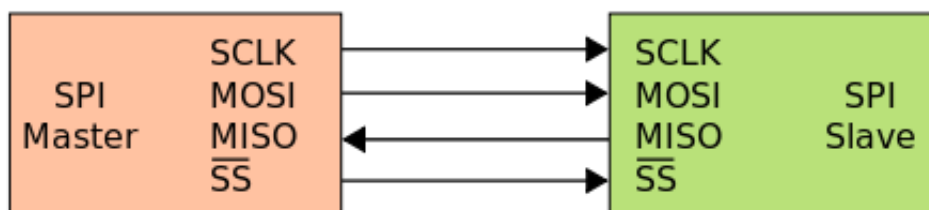
Obrázek 4.6: Rozpis pinů a jejich funkce vývojové desky

## 4.5 Použité vstupy a výstupy

V této části budou popsány jednotlivé V/V, které jsou použity v této práci. Dále se tato část bude věnovat popisu sériové komunikace, která bude též využita při komunikaci s periferními zařízeními a s nadřazeným řídicím systémem.

### 4.5.1 SPI

SPI je sériové periferní rozhraní. Používá se pro komunikaci mezi mikroprocesorem a ostatními integrovanými periferiemi např. EEPROM, DA převodníky, display a pod. Komunikace je realizována pomocí společné sběrnice. Adresace probíhá pomocí zvláštních vodičů, které při logické nule zaktivují příjem a vysílání zvoleného zařízení, které je přivedeno na signál  $\overline{SS}$  (Suicide Scientist; 2007).



Obrázek 4.7: Komunikace mezi Master a Slave

V této komunikaci jsou zařízení rozdělena do dvou skupin na Master a Slave. Master řídí komunikaci podle hodinového signálu dále se určuje, se kterým zařízením na sériové lince bude komunikovat pomocí  $\overline{SS}$  (Slave Select). Slave vysílá data, pokud je aktivován pomocí  $\overline{SS}$  signálu.

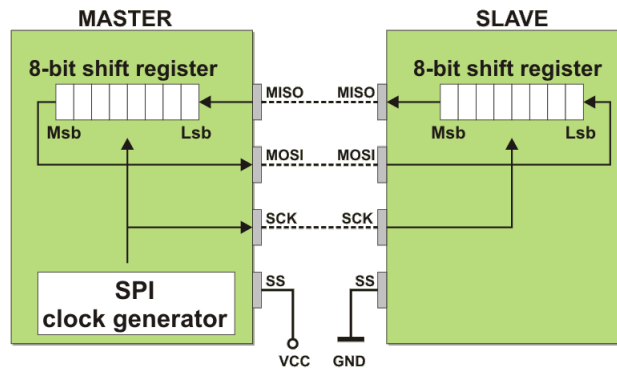
Dále se používá signál SCLK (Serial Clock), který je generátor hodinového signálu. Signál MOSI slouží k vlastní komunikaci, kde MO označuje Master Out a SI Slave In, tj. odesílaná data z mikroprocesoru a přijímaná data ze Slave zařízení. Port MISO je obrácený tj. vysílá data ze Slavu a přijímá data na Masteru.

Délka vysílaných dat je buď osmi bitová, nebo šestnácti bitová.

Výhodou této komunikace je, že jeden Master může ovládat více Slave zařízení a není tedy problém při využití jedné sběrnice komunikovat s více zařízeními. Způsob komunikace je následující. Master zařízení iniciuje přenos aktivováním  $\overline{SS}$  a generováním hodinového signálu. Při každém cyklu hodin dojde k přenesení jednoho bitu od Master zařízení ke Slave zařízení a naopak. Po přenesení



jednoho rámce tj. 8, nebo 16 bitů dojde k deaktivaci signálu  $\overline{SS}$  kvůli synchronizaci. Data Master a Slave zařízení jsou uložena v posuvném registru.



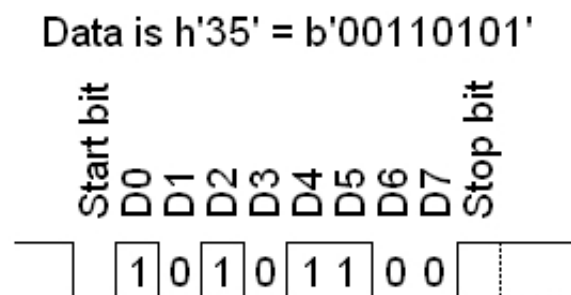
Obrázek 4.8: Schéma komunikace SPI mezi Master a Slave

#### 4.5.2 Serial UART komunikace

Serial UART komunikace probíhá buď synchronně, nebo asynchronně dle požadovaného nastavení. Jedná se o zařízení pro sériovou komunikaci. Ke komunikaci využívá dvou pinů RX a TX. Pin TX slouží k vysílání dat do druhého zařízení a pin RX k přijímání dat. Klidová úroveň je logická jednička, pokud je zahájen přenos, tak je Start bit nastaven na logickou nulu, poté dojde k odeslání informace v podobě 8 bitů zprávy, následně je odeslán Stop bit, který má opět hodnotu logické jedničky. Po odeslání tohoto bitu může opět začít znova odesílání zprávy, pokud je znovu vyžadována a je tedy Start bit na logické nule (obrázek 4.9) (Newcastle; 2008).

Komunikace pomocí USART může být nakonfigurována ve dvou režimech. První režim je asynchronní (full duplex), používá se často ke komunikaci s perifériemi zařízení jako jsou PC, CRT terminály apod. Druhý režim je synchronní (half duplex), které se využívá při komunikaci s perifériemi jako jsou EEPROM DA převodníky a pod. Tento synchronní přenos můžeme nastavit jako Master nebo Slave.

Tato sériová komunikace bude využita v navazující práci, kdy pomocí ní bude probíhat komunikace mezi Stellaris Launchpadem a nadřazeným řídicím systémem.

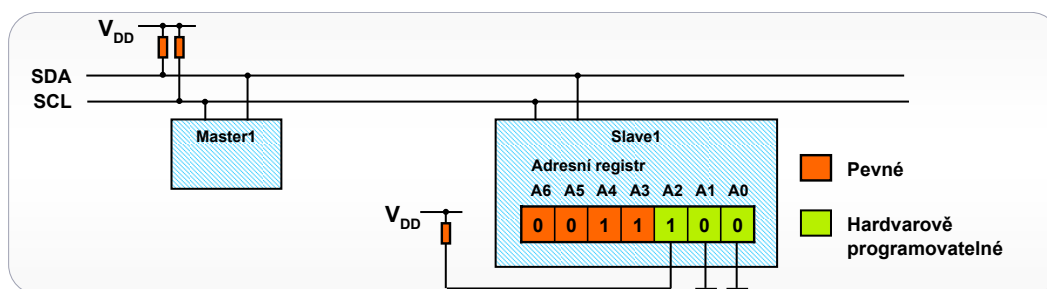


Obrázek 4.9: Asynchronní 8.bitový přenos

### 4.5.3 I2C komunikace

Sběrnice I<sup>2</sup>C je interní datová sběrnice, která slouží pro komunikaci a přenos dat mezi jednotlivými integrovanými obvody v jednom zařízení, vyvinuta firmou Philips. (HW server; 2000) V dnešní době se využívá především o mikrokontroléry, sériové paměti, inteligentní LCD a další zařízení. Hlavní výhodou je, že veškerá komunikace a přenos dat probíhá pouze na dvou vodičích, které jsou označeny jako SDA- datová a SCL což je hodinový signál. V této komunikaci je opět jeden řídí mikrokontrolér označen jako Master a ostatní obvody jako Slave. Sběrnice umožňuje současně řídit 128, nebo 1024 zařízení podle verze adresování zařízení, zda se jedná o sedmi bitové, nebo v rozšířené verzi o deseti bitové zařízení.

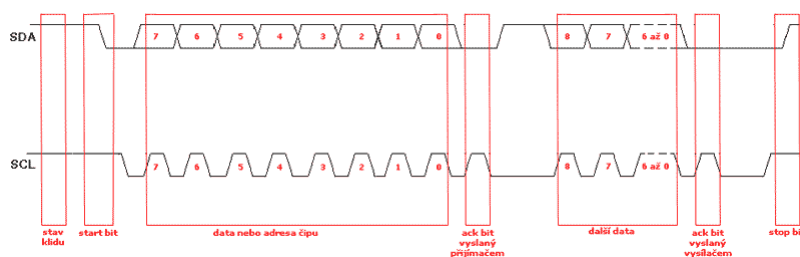
Tato komunikace může probíhat třemi rychlostmi: standard (100 kbps), fast (400 kbps) a high speed (3.4 Mbps). Již je zabudována detekce kolizí, podporuje více Master mikrokontrolérů, lze využít data broadcast (obecné vysílání). Každé zařízení má jedinečnou adresu, která je pevně stanovená. Každé zařízení má HW programovatelné piny a je SW programovatelné.



Obrázek 4.10: Adresace zařízení

Přenos probíhá kombinací následujících celků:

- **stav klidu** Je zajištěn logickými jedničkami na obou vodičích, master tedy negeneruje hodinový signál a neprobíhá žádný přenos. Logické jedničky jsou na obou vodičích zajištěny pull-up rezistory (rezistory mezi vodičem a napájecím napětím), takže klidový stav nastane i pokud jsou výstupy obvodu master ve stavu vysoké impedance (tedy odpojeny).
- **podmínka start** Zahajuje přenos nebo jeho další část. Je vygenerován tak, že se změní úroveň SDA z 1 na 0 zatímco je SCL v logické 1.
- **podmínka stop** Ukončuje přenos. Je vygenerován podobně jako start bit. Logická úroveň SDA se změní z 0 na 1 zatímco je SCL v logické 1. Stop bit může být generován pouze po "nepotvrzení přenosu", tedy pouze po přijmutí Ack v logické 1.
- **přenos dat** Data jsou přenášena po 1Byte tedy 8 po sobě jdoucích bitů od nejvyššího po nejnižší. Při přenosu dat se může logická úroveň na SDA měnit pouze pokud je SCL v logické 0. Při každém pulzu na SCL je přenesen jeden bit.
- **potvrzující bit ACK** Tento bit slouží k potvrzení správného přijmutí dat. Ack bit se odesílá stejným způsobem jako by se odesílal devátý bit dat, ale s tím rozdílem, že ho generuje čip, který přijímal (příjímač) a nikoliv ten, který data odesílal. Pokud přenos proběhl v pořádku, tak odešle logickou 0. Logická 0 potvrzujícího bitu znamená rovněž to, že je příjímač připraven na příjem dalšího byte, který následuje okamžitě po něm při dalším pulzu na SCL. Pokud přenos selhal odešle logickou 1. Nebo pokud má dojít k ukončení přenosu, tak "neodešle nic". Pull-up rezistor pak zajistí, že bude na SDA logická 1 a Ack bit (v logické 0) odešle vysílač.



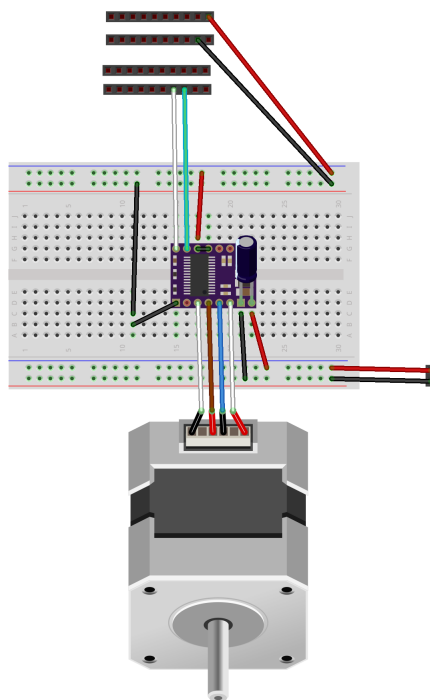
Obrázek 4.11: Průběh komunikace I2C

## 4.6 Návrh řídicího systému

V této části bude popsáno realizované zapojení hardwaru a návrh řízení celého manipulátoru.

### 4.6.1 Řídicí hardware

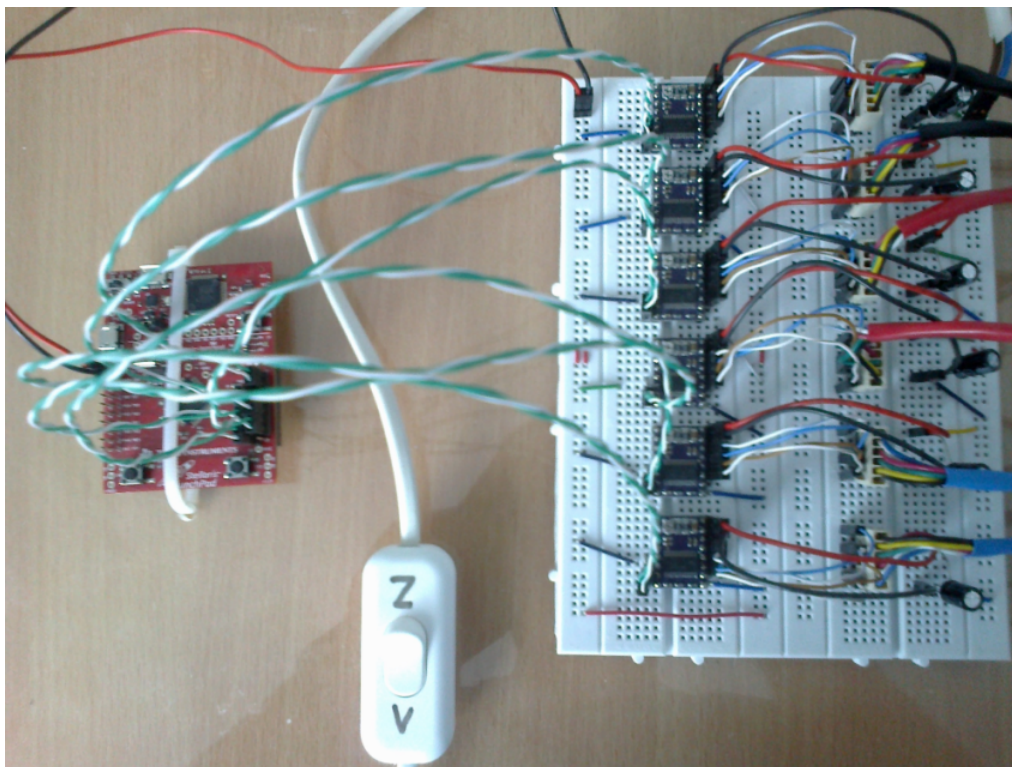
V této části bude uvedeno použité zapojení řídicího systému s ovládací krokovými motorů. Při zapojení bylo využito kontaktního nepájivého pole, na kterém je umístěno všech šest řadičů DRV8825. Na nepájivé pole jsou dále přivedeny napájecí kabely z řídicího systému Stellaris Launchpad a řídicí signály pro každý řadič. Každý řadič má v dvojici přivedeno datové kabely STEP a DIR na příslušné piny. Dále jsou na nepájivém poli přivedeny krokové motory, které jsou propojeny s řadiči. Nechybí ani napájení motorů, které obstarává zdroj napájení, a na každý řadič je přivedeno přes kondenzátor o kapacitě  $100 \mu\text{F}$ . Na řídicím systému Stellaris Launchpad bylo pro řízení krokových motorů vyhrazeny piny na lištách J4 a J2 (viz obrázek 4.6). Celkové zapojení je vidět na následujícím obrázku.



Obrázek 4.12: Zapojení řadiče

Na obrázku (obrázek 4.12) je vidět schéma realizovaného zapojení. Toto za-

pojení je pro všech šest řadičů stejné. Na obrázku jsou vidět vodiče, které mají červenou barvu to je kladné napětí, černé vodiče jsou záporné napětí. Barevné vodiče, které vedou od motoru, odpovídají skutečným vodičům. Vstupní napájení pro motory prochází přes kondenzátor, tak jak je uvedeno v manuálu řadiče. Zelený vodič je přiveden na Step a je to výstup z řídicího systému a bílý vodič je na Dir, což je to též výstup z řídicího systému.



Obrázek 4.13: Realizované kompletní zapojení šesti řadičů

Na obrázku (obrázek 4.13) je vidět, jak ve skutečnosti vypadá kompletní zapojení na kontaktním nepájivém poli. Celý řídicí systém je uchycen na desce z plexiskla a je též zapojeno tlačítko, které zapíná a vypíná napětí na motory. Toto je uděláno, protože se nejprve musí zapnout řadiče a až poté se může pustit napájení na motory, pokud by se to udělalo obráceně, tak hrozí zničení řadičů.

#### 4.6.2 Řídicí software

Tato část se bude věnovat ovládání jednotlivých krokových motorů pomocí řídicího systému Stellaris Launchpad. Nejprve bude popsáno zvolené vývojové prostředí, dále bude uveden a popsán algoritmus, který ovládá motory.

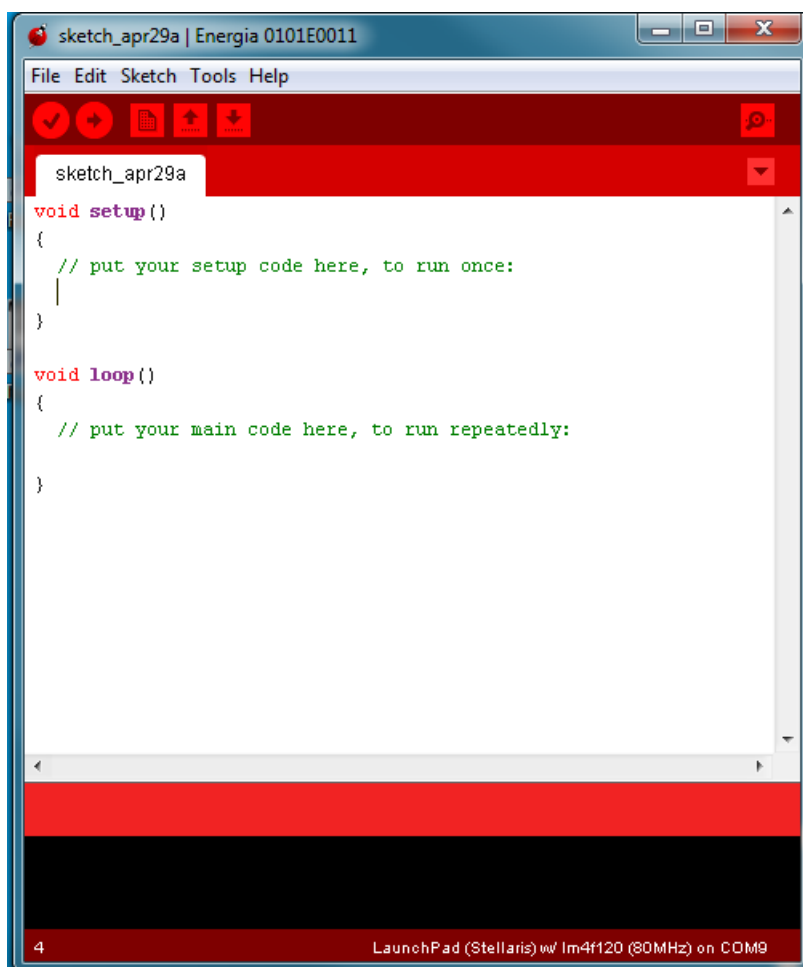
### 4.6.3 Vývojové prostředí pro Stellaris Launchpad

Pro zvolený řídicí systém existují dvě vývojové prostředí. První vývojové prostředí je placené od firmy TI a nese název Code Composer Studio<sup>2</sup>. Toto prostředí je relativně komplikované a nepřehledné, proto bylo použito druhé možné vývojové prostředí.

Toto vývojové prostředí se nazývá Energia (Wessels; 2012), jedná se o open source vývojové prostředí, které vychází z Arduina IDE, které je také open source, ale vývojáři Energie jej modifikovaly tak, aby v něm šlo programovat vývojové desky Stellaris Launchpad. Toto vývojové prostředí bylo zvoleno zejména proto, že je zdarma a také proto, že autor má již zkušenosti s programováním mikrořadičů na bázi Arduino a bylo tak jednodušší a rychlejší programování zvoleného řídicího systému. Vývojové prostředí umožňuje kontrolu napsaného kódu, dále je v něm již zakomponováno přeložení kódu do interního kódu mikrokontroléru a následné nahrání do jeho paměti a spuštění požadovaného kódu. Kód stejně jako v Arduinu obsahuje dvě základní metody. První metoda se nazývá setup() zde probíhá inicializace V/V pinů a je zde možno vykonat kód, který se má vykonat pouze jednou. Druhá důležitá metoda je loop(), ve kterém napsaný kód běží v nekonečné smyčce, to je proto, že mikrokontrolér běží stále a musí vykonávat nějakou činnost. Dále je možné psát funkce, které se volají v nekonečné smyčce loop(). Další výhodou je uživatelská podpora. Na stránkách jsou uvedené jednotlivé piny vývojové desky a jaké jsou možnosti jejich používání. Dále jsou na stránkách uvedeny všechny dostupné funkce, které lze využít během programování, a ve vývojovém prostředí nechybí ani ukázky kódů pro různé periférie.

---

<sup>2</sup>Code Composer Studio (CCS) Integrated Development Environment (IDE) - CCSTUDIO - TI Tool Folder[online].c2013 [cit. 26. dubna 2014]. Dostupné na WWW: <<http://www.ti.com/tool/ccstudio>>



Obrázek 4.14: Ukázka vývojového prostředí Energia

#### 4.6.4 algoritmus plošiny

Tato část se bude věnovat celkovému algoritmu řízení celé plošiny. Algoritmus bude implementován v navazující práci a je počítáno s tím, že se bude jednat o zpětnovazební řízení. V rámci této práce je aplikováno přímovazební řízení, které se skládá ze dvou částí. První část je získání informace o natočení krokových motorů. Tyto údaje jsou vypočítávané podle vztahů, které byly odvozeny v kapitole 3.1. Tyto hodnoty jsou poté zadány pomocí sériové linky do řídicího systému krokových motorů, který posune motory do požadovaného umístění. V navazující práci bude tento výpočet probíhat ve vyšší vrstvě řídicího systému, ve kterém se vytvoří virtuální model Stewartovy plošiny. Poté se bude čekat až uživatel zadá polohu vrchní desky.

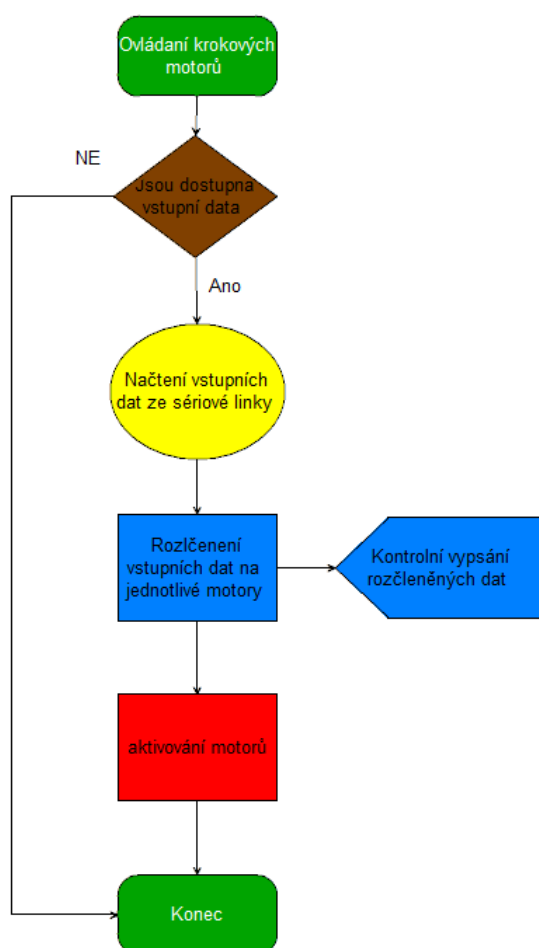
Řídicí systém bude muset nejprve provést určení polohy manipulátoru a nastavení manipulátoru do domovské pozice. Pro toto určení polohy bude vhodné použít nějaké čidlo na snímání polohy ramene každého motoru. Jako čidlo by mohlo stačit tlačítko, které po dotyku s ramenem vyšle signál řídicímu systému, ten bude vědět, že je rameno v nějaké poloze (např. minimální) a poté už bude snadné se dostat do domovské polohy plošiny.

Řídicí systém vypočítá délky ramen a odešle informace po sériové lince do podřazeného řídicího systému, který je Stellaris Launchpad. Ten vykoná zaslané informace, poté přečte informace z IMU a ty předá nadřazenému řídicímu systému, který vyhodnotí, zda se plošina nachází v požadované poloze. Pokud ano tak bude čekat na další zadání hodnoty od uživatele, pokud ne tak proběhne korekce natočení motorů, tak aby se plošina nacházela v požadovaném umístění. Tato implementace již bude zpětnovazební díky informacím z IMU, která bude sloužit jako senzor polohy.



#### 4.6.5 Algoritmus ovládání krokového motoru

V této části bude popsán vývojový diagram pro ovládání krokových motorů. Nejprve se čeká na vstupní data, která jsou přijata po sériové lince v poli hodnot o celkové velikosti 12 položek, jelikož dochází k řízení všech motorů současně a každému motoru náleží počet kroků a směr otáčení. Poté následuje rozdělení dat pro jednotlivé motory do dvou polí. V prvním poli jsou počty kroků a ve druhém směry. Tyto údaje jsou pro kontrolu vypsány pomocí sériové linky na monitor sériové linky. Poslední část je zavolání funkce, které ovládá řadiče. Ty realizují pohyb každého krokového motoru, dle zadaných parametrů.



Obrázek 4.15: Vývojový diagram řízení motorů

# Kapitola 5

## Závěr

Cílem této práce bylo navrhnout, modelovat a řídit manipulátor se šesti stupni volnosti. Jednalo se o velmi rozsáhlou práci a nyní budou zhodnoceny výsledky v jednotlivých pod úkolech této práce.

První úkolem bylo navrhnout konstrukci manipulátoru se šesti stupni volnosti. Poté, co si autor nastudoval problematiku manipulátorů, se rozhodl pro splnění úkolu zvolit paralelní manipulátor se šesti stupni volnosti, konkrétně manipulátor, který se nazývá Stewartova plošina. Její konstrukční prvky byly vytištěny na 3D tiskárně.

Druhý úkol byl navrhnout HW řídicího systému manipulátoru. Zde autor musel nejprve zvolit vhodnou náhradu lineárních pohonů. Zvoleny byly krokové motory, které se jeví jako dobrá náhrada nedostupných lineárních pohonů. Následně byly vybrány elektronické součástky, které byly použity v této práci nebo budou využity v práci následující. Jednalo se o ovladače krokových motorů DRV8825, dále byla vybrána vhodná IMU a nejdůležitější řídicí hardware Stellaris Launchpad, který byl nejlepší možnou volbou pro tuto i následující práci.

Třetí úkol byl vytvořit matematický model manipulátoru tento model byl odvozen pomocí dostupných informací a znalostí za použití zpětné kinematické úlohy, která se jako jediná hodí na popsání tohoto manipulátoru.

Poslední úkol byl navrhnout řídicí algoritmus manipulátoru. Řídicí algoritmus byl navržen a v této práci byl implementován jako dopředné řízení. Byl ukázán i návrh zpětnovazebního řízení, které bude aplikováno v následující práci.

Tato práce splnila všechny zadané cíle. Autor se seznámil s problematikou navrhování paralelního manipulátoru se šesti stupni volnosti. Dále se seznámil s možnostmi použitelného hardwaru pro ovládání manipulátoru. Sestavil funkční model, na kterém aplikoval dopředné řízení.

Tato práce je možná rozšířit o zpětnovazební řízení, které bude potřebovat odečítat polohu z IMU, dále je možné rozšíření o vyšší řídicí systém, který nahradí výpočty polohy z počítače. Další zajímavým rozšířením by mohla být úloha stabilizace nějakého objektu v určitém místě vrchní desky plošiny.

# Literatura

Daewoo (2009). Krokový motor - wikipedie.

**URL:** [http://cs.wikipedia.org/wiki/Krokový\\_motor](http://cs.wikipedia.org/wiki/Krokový_motor)

Gouběj, M., Švejda, M. and Schlegel, M. (2012). *Úvod do mechatroniky, robotiky a systémů řízení pohybu*, Katedra kybernetiky, ZČU.

HW server (2000). Stručný popis sběrnice i2c a její praktické využití k připojení externí eeprom 24lc256 k mikrokontroléru pic16f877.

**URL:** <http://www.hw.cz/navrh-obvodu/strucny-popis-sbernice-i2c-a-jeji-prakticke-vyuziti-k-pripojeni-externi-eeeprom-24lc256>

Marginally Clever (2014). Rotary stewart platform v2.

**URL:** <https://www.marginallyclever.com/shop/stewart-platforms/rotary-stewart-platform-v2?cPath=1>

Mor (2005). Inertial measurement unit - wikipedia, the free encyclopedia.

**URL:** [http://en.wikipedia.org/wiki/Inertial\\_measurement\\_unit](http://en.wikipedia.org/wiki/Inertial_measurement_unit)

Newcastle (2008). Usart – wikipedie.

**URL:** <http://cs.wikipedia.org/wiki/USART>

Pololu Corporation (2001-2014a). Pololu - drv8825 stepper motor driver carrier, high curren.

**URL:** <http://www.pololu.com/product/2133>

Pololu Corporation (2001-2014b). Pololu - stepper motor.

**URL:** <http://www.pololu.com/product/1200/>

Suicide Scientist (2007). Serial peripheral interface – wikipedie.

**URL:** <http://cs.wikipedia.org/wiki/SPI>

Texas Instruments (2012a). Analog, embedded processing, semiconductor company, texas instruments - ti.com.

**URL:** <http://www.ti.com/>

Texas Instruments (2012b). Stellaris® lm4f120 launchpad evaluation kit - ek-lm4f120x1 - ti tool folder.

**URL:** *<http://www.ti.com/>*

The MathWorks, Inc. (1994-2014). Mathworks - matlab and simulink for technical computing - b.

**URL:** *<http://www.mathworks.com/>*

Wendlandt, J. (2002). Stewart platform mechanical system - file exchange - matlab central.

**URL:** *<http://www.mathworks.com/matlabcentral/fileexchange/2334-stewart-platform-mechanical-system>*

Wessels, R. (2012). Energia reference – getting started.

**URL:** *[http://energia.nu/Guide\\_index.html](http://energia.nu/Guide_index.html)*

Wokingham U3A (2011). Maths.

**URL:** *<http://www.wokinghamu3a.org.uk/Maths.html>*