

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
DEPARTMENT OF CONTROL AND INFORMATICS

OVLÁDACÍ SOFTWARE MANIPULÁTORU MINI-SWING

IMPLEMENTATION OF A OPERATING SOFTWARE FOR MANIPULATOR MINI-SWING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ONDŘEJ PRAŽÁK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PAVEL HOUŠKA, Ph.D

BRNO 2010

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky
Akademický rok: 2009/2010

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): Ondrej Pražák

který/která studuje v **bakalářském studijním programu**

obor: **Aplikovaná informatika a řízení (3902R001)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Ovládací software manipulátoru Mini-Swing

v anglickém jazyce:

Implementation of a operating software for manipulator Mini-Swing

Stručná charakteristika problematiky úkolu:

Cílem práce je realizovat software pro řízení robotického manipulátoru se třemi řízenými osami.

Cíle bakalářské práce:

1. Prostudujte problematiku řízení robotických manipulátorů;
2. Navrhněte strukturu ovládacího software;
3. Software realizujte v prostředí NI LabVIEW;
4. Realizovaný software ověřte.

Seznam odborné literatury:

- [1] Ďaďo S., Kreidl M.: Senzory a měřicí obvody, Skriptum ČVUT, 1999
- [2] Noskievič, P.: Modelování a identifikace systémů, Montanex a.s., Ostrava 1999, ISBN 80-7225-030-2
- [3] Skalický J., Elektrické servopohony, Nakladatelství VUT, Brno
- [4] National Instruments: LabVIEW Control Design User Manual[online], June 2009, 371057G-01, dostupné z URL:<<http://www.ni.com>>
- [5] National Instruments: LabVIEW Fundamentals, August 2005, 374029C-01, dostupné z URL:<<http://www.ni.com>>

Vedoucí bakalářské práce: Ing. Pavel Houška, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2009/2010.

V Brně, dne 18.11.2009

L.S.

Ing. Jan Roupec, Ph.D.
Ředitel ústavu

prof. RNDr. Miroslav Doupovec, CSc.
Děkan fakulty

Abstrakt

Bakalářská práce se zabývá řízením robotického manipulátoru Mini-swing se třemi řízenými osami. Pro řízení byla navržena a realizována řídicí aplikace pro řízení pohybu celého manipulátoru s možností zadávání pracovních bodů. Dále byla navržena a realizována real-time řídicí aplikace pro řízení pohybu samostatných motorů. Obě aplikace jsou realizovány v prostředí NI LabVIEW. Aplikace mezi sebou navzájem komunikují pomocí sdílených síťových proměnných.

Abstract

The bacheleor thesis deal with operation of the robotic manipulator Mini-swing with three controlled axes. An application was designed to control the movement of the whole manipulator with the option of entering different working points. Furthermore, a real-time controlling application was designed in order to control the individual motors of the manipulator. Both applications are implemented in NI LabVIEW environment and they are able to communicate with each other with the help of shared network variables.

Klíčová slova

Robot, manipulátor, motor, efektor, automat, simulátor

Keywords

Robot, manipulator, motor, effector, automat, simulator

Bibliografická citace mé práce:

PRAŽÁK, O. *Ovládací software manipulátoru Mini-Swing*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2010. 45 s. Vedoucí bakalářské práce Ing. Pavel Houška, Ph.D.

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci na téma: „Návrh konstrukce a řízení manipulátoru pro laboratorní měření teplotního pole“, vypracoval samostatně pod vedením Ing. Pavla Houšky, Ph.D., na základě dostupné literatury a dostupných informačních zdrojů, které jsem všechny odcitoval v seznamu použité literatury.

V Brně dne

podpis.....

Poděkování

Velký dík patří vedoucímu mé bakalářské práce Ing. Pavlu Houškovi Ph.D.. Rád bych mu tímto poděkoval za cenné rady, podněty a odborné vedení, při vypracování této bakalářské práce.

Obsah

1	Úvod.....	15
2	Manipulátory.....	17
2.1	Rozdělení manipulátorů (robotů).....	17
2.1.1	Sériové roboty	17
2.1.2	Paralelní roboty	17
2.1.3	Kombinované roboty	18
2.2	Robotika sériových i paralelních manipulátorů.....	19
3	Popis manipulátoru Mini-swing	21
3.1	Pohony.....	21
3.2	Snímače	23
3.2.1	Princip rotačního inkrementálního snímače.....	23
3.2.2	Parametry enkodérů Maxon MR	24
3.3	Kinematický model manipulátoru.....	25
3.3.1	Pracovní prostor	25
3.3.2	Přímá úloha pro polohu	26
3.3.3	Nepřímá úloha pro polohu	28
3.3.4	Dosažení okrajové polohy	30
4	Návrh řízení manipulátoru	31
4.1	Snímání polohy a přesnost manipulátoru	31
4.2	Návrh způsobu řízení	31
4.3	Realizace řízením manipulátoru	32
4.3.1	Řídicí aplikace pro řízení pohybu manipulátoru.....	32
4.3.2	Ovládací program na cRIO	35
5	Závěr.....	41
6	Použitá literatura	43

1 Úvod

Robotické manipulátory se v současnosti používají v mnoha oborech. Dělí se na tři základní skupiny. Jsou to manipulátory (roboty) sériové, paralelní a kombinované. U sériových robotů se každá jeho část pohybuje nezávisle na sobě. Proto při přesunu do požadované polohy se může hýbat jedna část, anebo všechny zároveň. Paralelní roboty ke svému pohybu využívají složený pohyb všech jeho částí, kombinované využívají kombinace pohybů z obou předchozích typů. Tato oblast je však mnohem členitější a složitější. Patří sem například robotické ruce.

Jako pohony je možné využít 3 typy pohonů. Elektrické, pneumatické a hydraulické. Elektrické motory jsou levnější na pořízení, údržbu i provoz. U přeměny elektrické energie na práci nedochází k takovým ztrátám jako u zbylých dvou typů motorů. Ty jsou zase až 5x menší při zachování stejného výkonu a dají se mnohem více přetěžovat. Potřebují však zařízení pro výrobu stlačeného vzduchu nebo olejová čerpadla. Nehledě na to že olej je potřeba udržovat v určité provozní teplotě a vzniká zde problém s recyklací při jeho výměně.

Pro snímání polohy je možné použít inkrementální nebo absolutní snímače. Absolutní si pamatují svou polohu i při vypnutí a opětovném zapnutí přístroje na rozdíl od inkrementálních, které jsou schopny pouze snímat otáčky a rychlost, a po vypnutí si nepamatují svou předešlou polohu.

V práci je řešeno řízení jednoduchého manipulátoru Mini-swing, který má dvě řízené osy a koncový efektor s jednou řízenou osou. Pro tento manipulátor je navržen ovládací program který nejprve zjišťuje polohy žádané a skutečné. Pak se provedou veškeré výpočty, jako například přepočítávání souřadnic na úhly, vybírání kratší dráhy pro přejezd a podobně. Z těchto spočítaných úhlů jsem schopen odesílat natočení, o které je potřeba motor pootočit. Spolu s tím reguluji motory tak aby se zastavili v dané poloze a byly regulovány tak, aby tuto polohu drželi.

2 Manipulátory

V této kapitole se budu zabývat rozdělením manipulátorů, jejich využitím v praxi a teoretickými problémy.

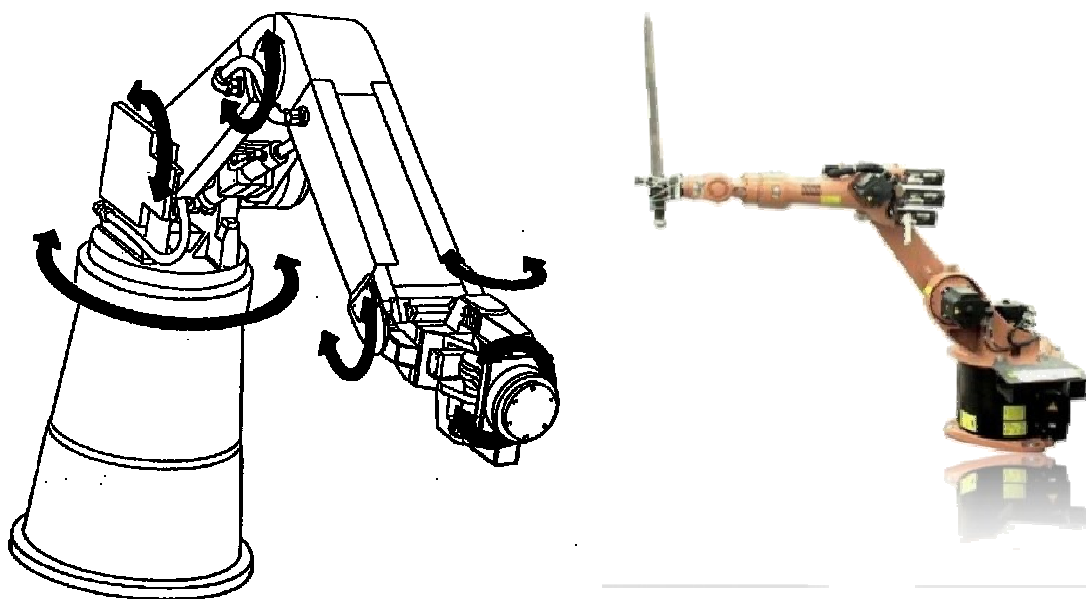
Manipulátory nebo spíše průmysloví roboty se ve větší míře používají od poloviny 20-tého století. Tehdy spíše jako mechanické manipulátory, dnes například svařovací automaty, letové simulátory, ale také ve zdravotnictví, kosmonautice, strojírenství a mnoho jiných. Dnešní manipulátory, například hexaply (obr.2.2), jsou rychlejší, přesnější, mají lepší dynamické vlastnosti, a proto nacházejí mnohem větší uplatnění v praxi.

2.1 Rozdělení manipulátorů (robotů)

Manipulátory lze dělit takto [2][3]:

2.1.1 Sériové roboty

U robotů sériových, se každá část pohybuje nezávisle na sobě a výsledný pohyb je složen z na sebe navazujících pohybů. Tyto pohyby jsou nejčastěji elementární. Na Obr. 2.1 vidíte schéma šestičlenného robota.



Obr. 2.1. Schéma šestičlenného robota [2] [4]

2.1.2 Paralelní roboty

Pohyb paralelních robotů vzniká součinností všech jeho částí. Tyto části jsou řazeny vedle sebe. Chceme-li změnit polohu je většinou zapotřebí pohybu všech částí, proto

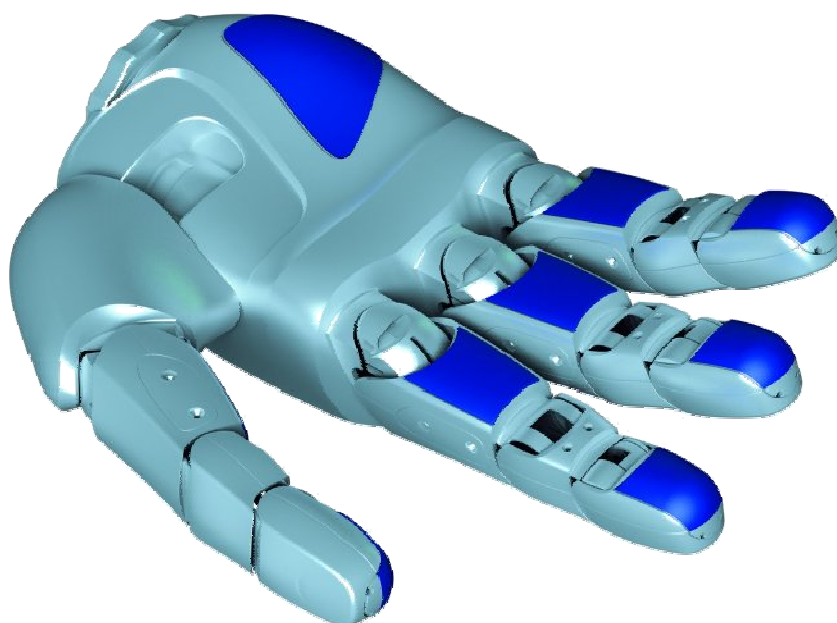
pohyb jedné části ovlivňuje pohyb ostatních částí. Například letecký simulátor, jehož pohybová část je na Obr. 2.2.



Obr. 2.2. Paralelní manipulátor (hexapod) [5]

2.1.3 Kombinované roboty

Kombinované roboty vznikají kombinací pohybů ze sériových a paralelních robotů. Například mechanické ruky Obr. 2.3, chodící stroje a jiné.



Obr. 2.3. Mechanická ruka [6]

2.2 Robotika sériových i paralelních manipulátorů

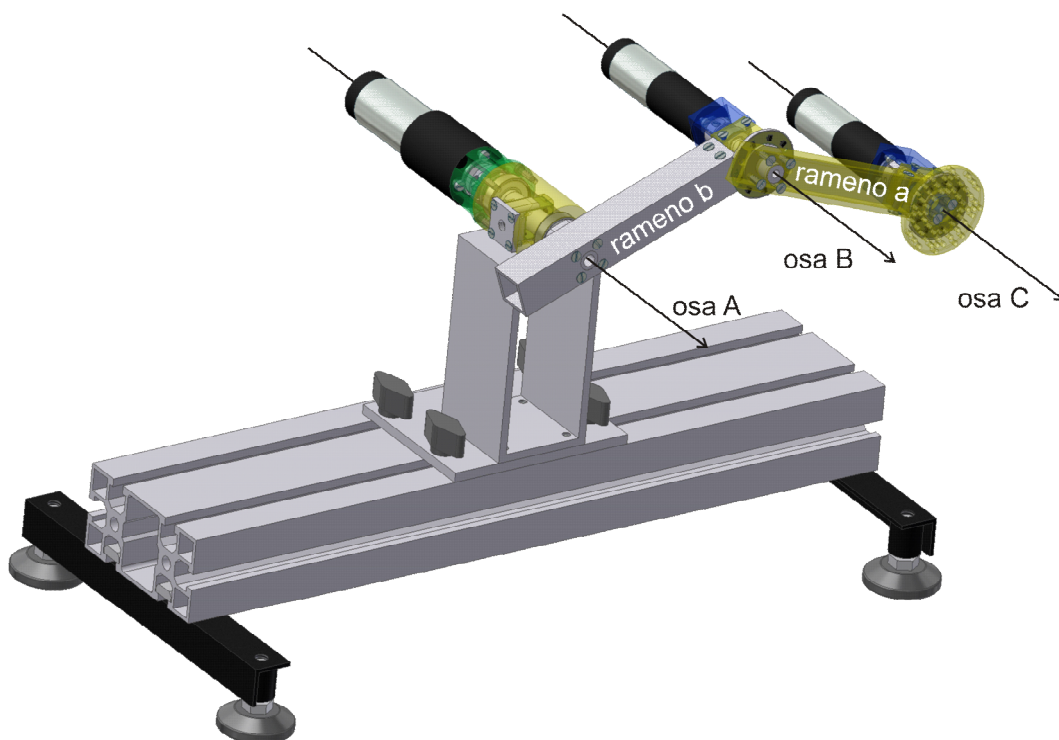
Tato kapitola vychází z [6].

Budeme-li na robota pohlížet z kinematického hlediska, dá se rozdělit na dvě mechanické soustavy, a to pevné a hybné. Pevná soustava neboli báze je pevně uložena v pracovním prostoru stroje. Hybná soustava je pevně spojena s nástrojem případně s členem, pomocí kterého robot vykonává výsledný pohyb (end-effector). Pohyb end-effektoru, který je zprostředkován robotem, je popsán aplikací kinematiky a geometrie v robotice. Je tedy obrovský rozdíl mezi kinematickým popisem paralelního manipulátoru a sériového robota. Proto se jimi budu zabývat zvlášť.

- **Přímá úloha pro polohy** – ze známé vzájemné polohy členů robota najít polohu nástroje nebo efektoru. V případě sériového robota je vzájemná poloha členů robota určena vzdálenostmi a úhly jejich os, v případě paralelního manipulátoru známe délky jednotlivých teleskopických ramen.
- **Obrácená úloha pro polohy** – známe předpokládanou polohu nástroje nebo efektoru a hledáme odpovídající „nastavení“ členů robota.
- **Přímá úloha pro rychlosti** – známe relativní rychlosti pohybu jednotlivých členů robota a chceme najít operátor rychlosti v prostoru nástroje nebo efektoru.
- **Obrácená úloha pro rychlosti** – známe operátor rychlosti v prostoru nástroje nebo efektoru a chceme určit rychlosti pohybu jednotlivých členů robota nebo manipulátoru.

3 Popis manipulátoru Mini-swing

Manipulátor Mini-swing na obr. 3.1 je dvouramenný manipulátor se dvěma řízenými osami a s koncovým efektem s řízeným jedním stupněm volnosti. Dva motory nastavují ramena tak, aby koncový bod dosáhl požadované polohy a tento koncový bod je schopný se díky třetímu motoru natáčet. Může tedy například natáčet případným, například měřicím zařízením. Délka obou ramen je 100 mm. Tento manipulátor není schopen přejíždět přes podstavu ani přes svá vlastní ramena díky vyčnívajícím motorům.

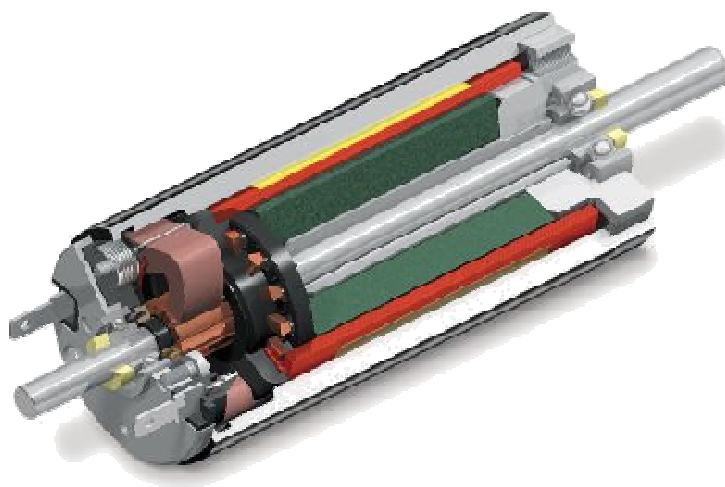


Obr. 3.1. Manipulátor Mini-swing

3.1 Pohony

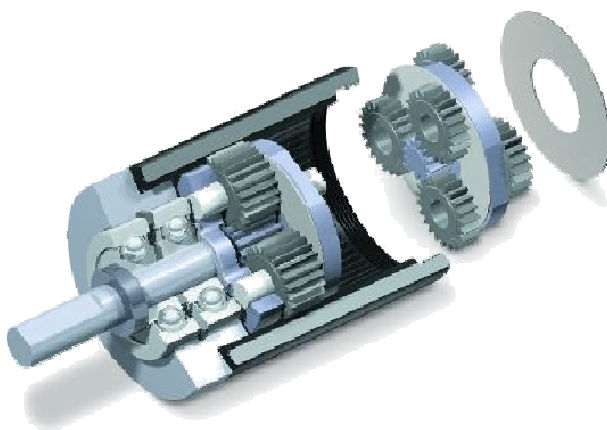
Motory se skládají z rotoru a statoru. Stator je zpravidla vyráběn z magneticky tvrdých materiálů. Rotor (kotva) se skládá z několika cívek, ze kterých jsou jednotlivé konce cívek připojeny ke komutátoru. Na tento komutátor je přiváděn proud pomocí uhlíkových kartáčů. Toto zařízení se nazývá sběrací ústrojí.

Na tomto manipulátoru jsou použity motory od firmy MAXON [7]. Viz. Obr. 3.2. Tyto motory vynikají výbornými dynamickými vlastnostmi díky výkonným magnetům na bázi vzácné zeminy FeNdB a samonosnému vinutí bez ocelového rámu. Díky tomu jsou zároveň odolnější proti magnetickému rušení. Jsou odolnější i vyšším teplotám. Maximální přípustnou teplotu uvádí výrobce až 125°C. Životnost je v příznivých podmínkách uváděna na více než 20 000 hodin.



Obr. 3.2. MAXON DC motor [7]

Použité planetové převodovky od firmy MAXON (Obr. 3.3) vynikají možností přenášet vysoké krouticí momenty, malými rozměry a soustředně uložené vstupní i výstupní hřídele. V Tab. 1 jsou jednotlivé převodové poměry spolu s parametry motorů a enkodérů.



Obr. 3.3. Planetová převodovka MAXON [9]

Největší motor RE-max 29 je pohon kloubu A a hýbe ramenem „b“, respektive celým manipulátorem, viz. Obr. 3.8. Zbylé dva motory jsou stejné, liší se pouze převodovkami. Převodovka s větším převodem je použita jako pohon kloubu B a hýbe ramenem „a“. Třetí motor pouze otáčí upínací maticí v bodě C.

Osa	A	B	C-efektor
MOTOR	RE-max 29	RE-max 22	RE-max 22
Jmenovité napětí [V]	24	24	24
Průměr [mm]	29	22	22
Jmenovité otáčky [ot/min]	7680	7430	7430
Jm. krouticí moment [mNm]	26,9	6,79	6,79
Jmenovitý proud [A]	1,08	0,35	0,35
Účinnost [%]	87	73	73
PŘEVODOVKA	GP 32 C	GP 22 C	GP 22 C
Průměr [mm]	32	22	22
Převodový poměr	132:1	370:1	84:1
Absolutní redukce	3312/25	10556001/28561	185193/2197
ENKODÉR MR	Typ ML	Typ M	Typ M
dílky na otáčku	1000	512	512
počet kanálů	3	3	3
max. operační frekvence [kHz]	200	320	320
rozlišitelnost [°]	0,0007	0,0005	0,0021

Tab. 1. Tabulka parametrů motorů, převodovek a enkodérů [7]

3.2 Snímače

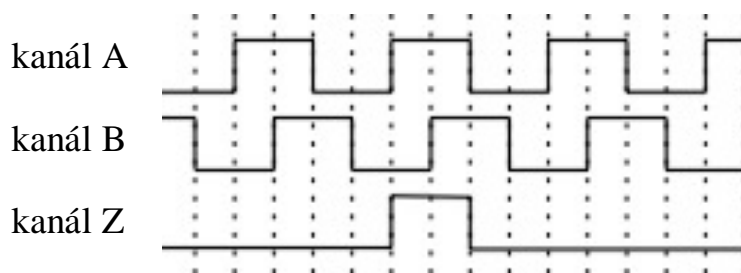
Použité pohony jsou osazeny přírůstkovými rotačními snímači (IRC) pro snímání polohy natočení hřídele. Dále je konstrukce manipulátoru připravena na osazení absolutními snímači natočení na principu sledování polohy permanentního magnetu pomocí Hall snímačů. Protože absolutní snímače polohy nebyly během řešení této práce k dispozici, je nutné výchozí polohu manipulátoru definovat externě.

3.2.1 Princip rotačního inkrementálního snímače

Rotační inkrementální (přírůstkové) snímače [1] pracují jako elektromechanické převodníky, které převádí otáčivý pohyb na elektrické signály. Tyto signály jsou

generovány například průchodem světla šěrbinami v kotouči, odrazem světla, elektromagnetickou indukcí, detekcí změny magnetické pole magnetu a podobně.

Snímač tvoří dvě části, rotující kotouč se značkami a detektor, který značky detekuje a generuje odpovídající elektrický signál. Detektor představuje tzv. kanál, v technické praxi se používají snímače s jedním, dvěma nebo třemi kanály (detektory). Z tříkanalového snímače dostáváme signál A a B, které jsou díky posunutým snímačům také posunuty o 90 elektrických stupňů. Třetí signál Z generuje pouze jeden „referenční“ signál na otáčku. Viz obr. 3.4.

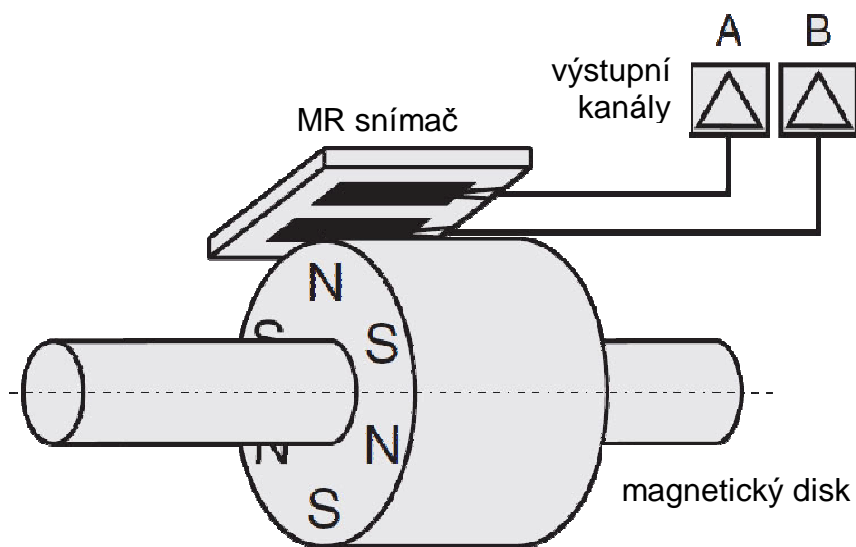


Obr. 3.4. Výstupní signály snímače [8]

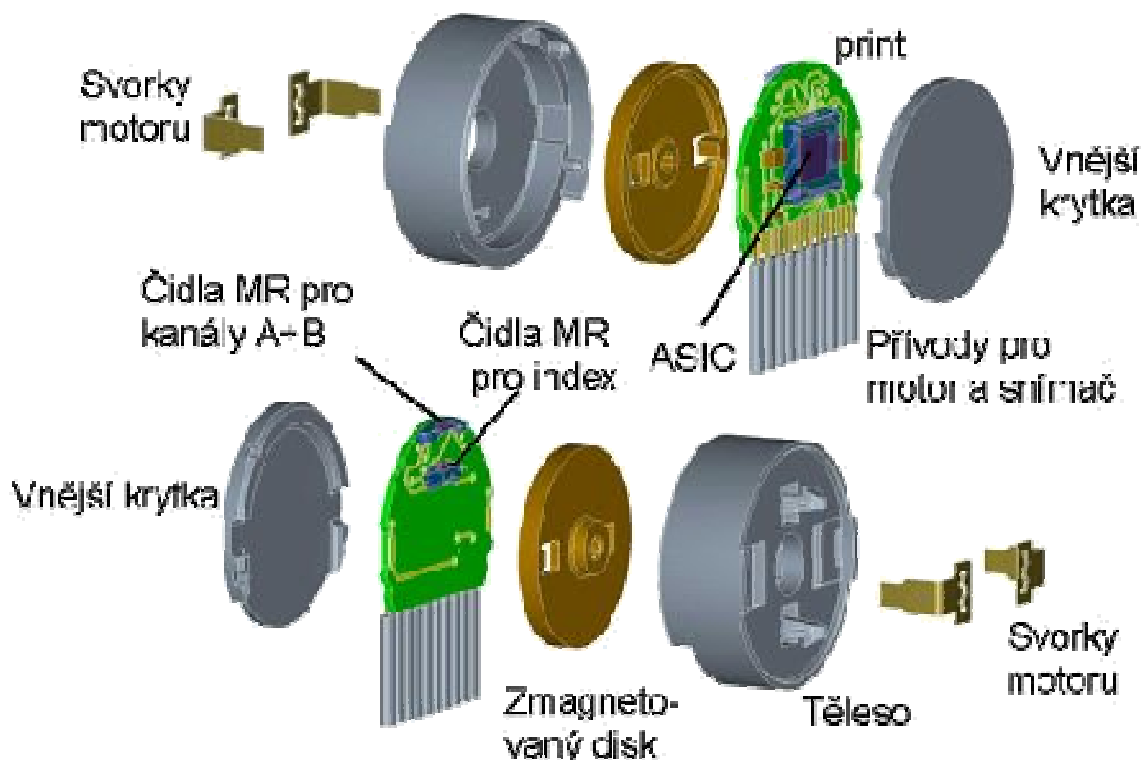
3.2.2 Parametry enkodérů Maxon MR

Použité pohony jsou osazeny magnetickými enkodéry Maxon MR [7], vlastnosti enkodérů na jednotlivých pohonech jsou uvedeny v tab. 1.

Enkodéry Maxo MR pracují na magneto-rezistentním (MR) principu (Obr. 3.5), který dovoluje realizaci rozměrově velmi malých snímačů. Signál z MR snímače je interpolován, aby se dosáhlo vysokého rozlišení enkodéru. MR enkodéry se vyrábí dvou nebo tří kanálové obr. 3.4. V MR-enkodéru, vytváří magnetický disk namontovaný na hřídeli sinusové vlny, které se mění na napětí v MR senzoru. Signál z MR-enkodéru je tvořen interpolací a následným zesílením elektrického signálu.



Obr. 3.5. Schéma Magneto-rezistentního enkodéru [7]



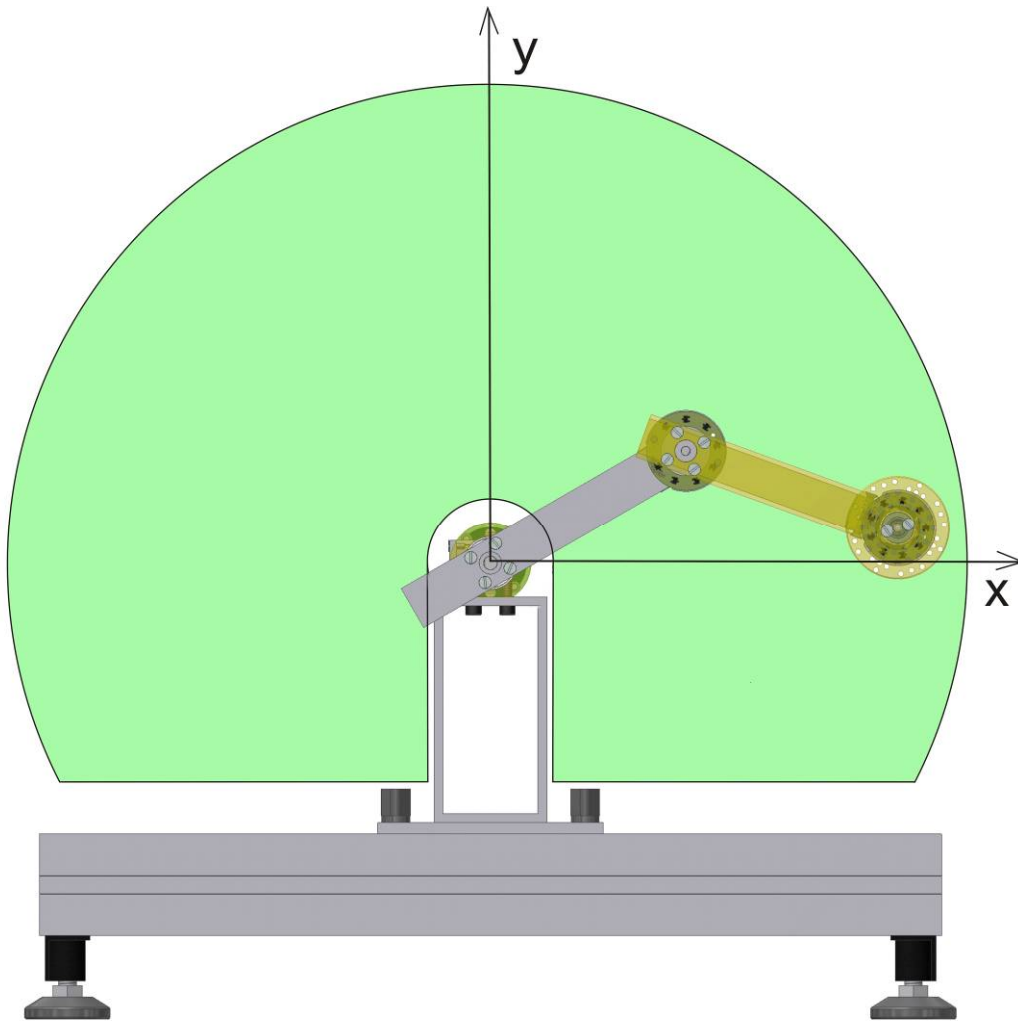
Obr. 3.6. Použitý MR snímač spolu s popisem [7]

3.3 Kinematický model manipulátoru

3.3.1 Pracovní prostor

Mini-swing je dvouramenný manipulátor, proto je jeho pracovní prostor rozdělen na dvě pomyslné kružnice. Vnější kružnice vznikne napřímením obou ramen do jedné přímky a následnou rotací kolem osy. V této kružnici je tedy pracovní prostor, kde mohou ramena manipulátoru pracovat. Velikost vnitřní kružnice je dána velikostí jednotlivých motorů. Zezdola je prostor omezen podstavou a podstavcem na kterém je uchycen motor A. Na je zeleně tento pracovní prostor vyznačen.

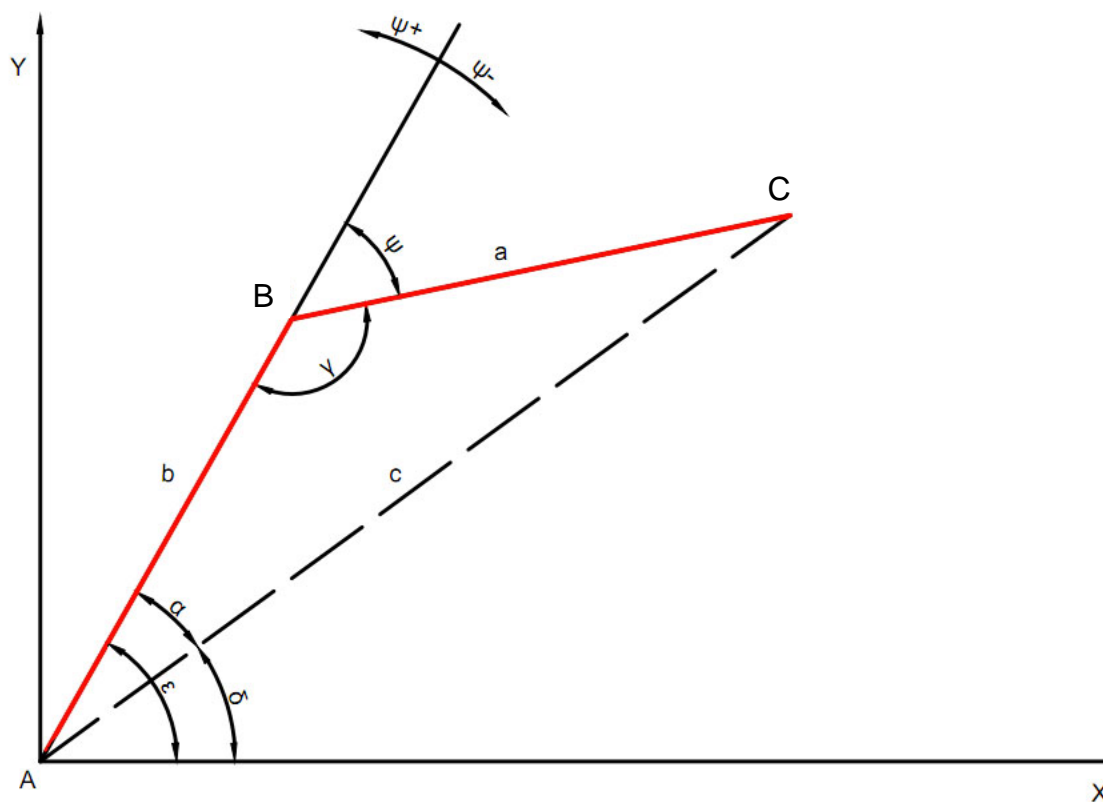
Obě ramena mají délku 100 mm, pokud tato ramena napřímíme, dostáváme velikost vnější kružnice. Ta je však omezena podstavou, která je vzdálena od osy x 100 mm. Vnitřní kružnice, která omezuje pracovní prostor, vzniká tloušťkou vnitřního a koncového motoru. Tuto kružnici jsem však zvětšil podle šířky podstavce, na kterém je připevněn motor A. Tato kružnice má tedy 50 mm stejně jako podstavec. Na obr. 3.7 je vidět že by se tento pracovní prostor dal zvětšit, ale pro mé potřeby stačilo toto jednoduché omezení.



Obr. 3.7. Pracovní prostor manipulátoru

3.3.2 Přímá úloha pro polohu

Vzájemná poloha jednotlivých členů manipulátoru je dána délkami ramen a úhly které tyto ramena svírají. Z těchto vzájemných poloh hledáme konečnou polohu. Zjednodušený model manipulátoru (Obr. 3.8) se skládá ze dvou ramen. Rameno „b“ je spojeno v počátku rotační vazbou, na svém druhém konci je spojeno s ramenem rameno „a“ taktéž rotační vazbou.



Obr. 3.8 Zjednodušený model manipulátoru

Spojíme-li počáteční a koncový bod A a C vznikne nám pomyslný trojúhelník se stranami a, b, c a úhly α , β , γ . Počáteční bod $A[x_A, y_A]$ má souřadnice $A[0,0]$. Koncový bod $C[x_C, y_C]$ a bod $B[x_B, y_B]$ který spojuje obě ramena rotační vazby. Velikost přepony „c“, neboli vzdálenost koncového bodu C od počátku, se vypočítá pomocí Kosinovy věty:

$$c^2 = a^2 + b^2 - 2ab \cdot \cos(\gamma) \quad (3.1)$$

$$c = \sqrt{a^2 + b^2 - 2ab \cdot \cos(\gamma)}$$

Jelikož známe všechny 3 strany, a úhly γ , ϵ musíme dopočítat velikost úhlu δ a α a to takto:

$$\alpha = \arccos \left[\frac{b^2 + c^2 - a^2}{2 \cdot b \cdot c} \right] \quad (3.2)$$

Úhel δ svírá osa „x“ a přepona „c“. Tento úhel vypočítáme odečtením vypočítaného úhlu α od známého úhlu ε :

$$\delta = \varepsilon - \alpha \quad (3.3)$$

Pro zjednodušení si zavedeme úhel ψ který je pouze doplněk úhlu γ do 180° . Tento úhel bude kladný nebo záporný podle toho jak bude natočený k rameni „b“.

$$\psi = \gamma - 180^\circ \quad (3.4)$$

Tímto máme veškeré potřebné úhly a strany pro výpočet konečných souřadnic bodu C a bodu B:

$$B_x = b \cdot \cos(\varepsilon) \quad (3.5)$$

$$B_y = b \cdot \sin(\varepsilon)$$

$$C_x = a \cdot \sin(\psi) \quad (3.6)$$

$$C_y = a \cdot \cos(\psi)$$

Souřadnice konečného bodu B dostaneme sečtením x-ových a y-ových složek obou souřadných systémů:

$$B_x = C_x + B_x \quad (3.7)$$

$$B_y = C_y + B_y$$

3.3.3 Nepřímá úloha pro polohu

Zde známe polohu konečného bodu a snažíme se najít optimální nastavení ramen manipulátoru. Nepřímá úloha může mít více řešení a proto je komplikovanější než úloha přímá. Musíme s tímto faktem počítat hned na začátku výpočtů. Přesto je řešení pro obě úlohy nejprve stejné.

Známe-li tedy požadovaný bod $C[x_C, y_C]$ a délky ramen, jsme schopni vypočítat nastavení polohy manipulátoru. Použijeme trojúhelník z Obr. 3.8, kde fiktivní strana „c“ má délku:

$$c^2 = a^2 + b^2 - 2ab \cdot \cos(\gamma) \quad (3.8)$$

$$c = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

Úhel δ , který svírá strana „c“ a osa „x“, se vypočítá pomocí rovnice:

$$\delta = \arccos \left[\frac{x_B}{c} \right] \quad (3.5)$$

Jelikož známe všechny 3 strany, můžeme pro výpočet velikostí úhlů α a γ použít Kosinovu větu:

$$\alpha = \arccos \left[\frac{b^2 + c^2 - a^2}{2 \cdot b \cdot c} \right] \quad (3.6)$$

$$\gamma = \arccos \left[\frac{a^2 + b^2 - c^2}{2 \cdot a \cdot b} \right]$$

Tímto máme veškeré potřebné úhly pro výpočet výsledného úhlu ε . Je to naklonění ramene „b“ od osy „x“. Zde nám ale vznikají dvě řešení. Přičtením a odečtením úhlu alfa:

$$\varepsilon_1 = \alpha + \delta \quad (3.7)$$

$$\varepsilon_2 = \alpha - \delta$$

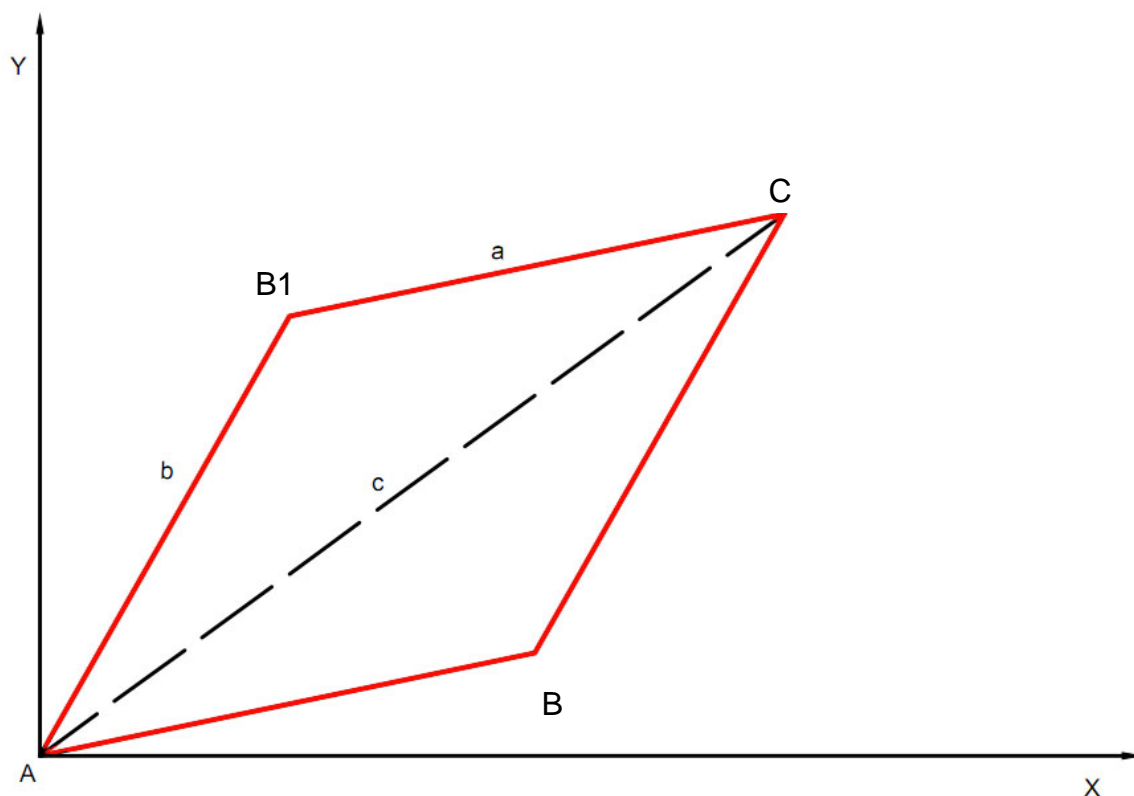
Na Obr. 3.9 jsou tato dvě řešení patrná. Tímto dostaneme bod $B_1[x, y]$ a bod $B_2[x, y]$. Mezi nimi je čerchovaně vyznačená osa symetrie. Tyto body dostaneme takto:

$$B_1(x) = b \cdot \cos(\varepsilon_1) \quad (3.8)$$

$$B_1(y) = b \cdot \sin(\varepsilon_1)$$

$$B_2(x) = b \cdot \cos(\varepsilon_2) \quad (3.9)$$

$$B_2(y) = b \cdot \sin(\varepsilon_2)$$



Obr. 3.9 Dvě možná řešení nastavení manipulátoru

3.3.4 Dosažení okrajové polohy

Manipulátor je v této poloze maximálně napřímen a dosahuje na samotný okraj pracovního prostoru. Pokud bychom potřebovali znát polohu bodu $C[x, y]$, můžeme ji vypočítat takto:

$$C_x = a + b \cdot \cos(\varepsilon) \quad (3.10)$$

$$C_y = a + b \cdot \sin(\varepsilon)$$

4 Návrh řízení manipulátoru

Řízení manipulátoru jsem se rozhodl realizovat pomocí grafického vývojového prostředí LabVIEW 2009 (Laboratory Virtual Instruments Engineering Workbench). Tento program od firmy National Instruments je schopen vytvářet aplikace pro nejrůznější oblasti, jak o například měření, sběr, analýzy a prezentace naměřených dat. Uživatel může pomocí tzv. Grafického jazyku plnohodnotně vytvářet aplikace se všemi programovými a datovými strukturami.

Mnou vytvářený ovládací program musí umožňovat zadávat požadovanou polohu uživatele, zobrazovat aktuální polohu koncového bodu a zároveň musí zvládat běžné provozní stavy a pohybovat se ve svém pracovním prostoru.

4.1 Snímání polohy a přesnost manipulátoru

Parametry enkodérů jsou uvedeny v tab.1. Přesnost celého manipulátoru je značně omezena konstrukcí. Velký vliv na přesnost mají tuhosti jednotlivých ramen a vůle v převodovkách. Zatímco se motor otáčí tak v prvním okamžiku dojde k vymezení vůlí, a teprve po vymezení vůlí v převodovkách se začínají pohybovat ramena. Dostane-li se rameno na opačnou horizontální polohu, vůle v převodovkách se díky gravitačnímu zrychlení vymezí opačným směrem a rameno se nepatrně „propadne“.

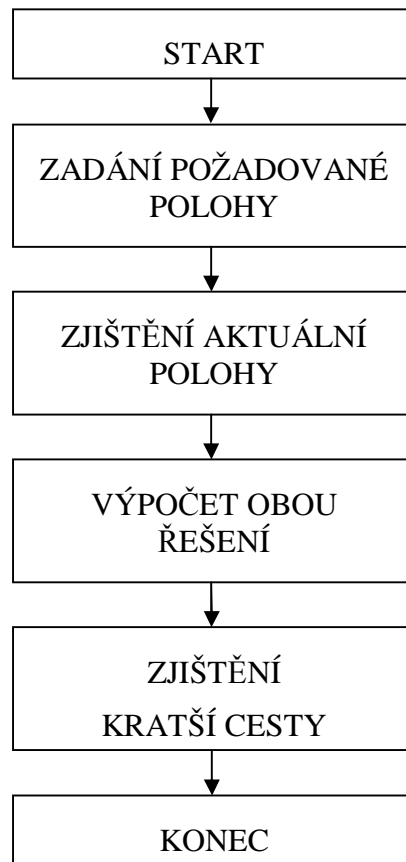
Použil jsem dva typy snímačů. Pohony B a C mají snímače typu M. Viz. Obr. 3.6. Ty mají 512 dílků na otáčku a 3 kanály. Z každého enkodéru pohonu B a C je možno detekovat 2048 impulzů na otáčku. Počítám-li vzestupnou a sestupnou hranu a dva navzájem o 90° posunuté snímače. Rozlišitelnost ve stupních zjistím tak že, jednu otáčku tedy 360° vydělím 2048 impulzy. Rozlišení snímače je tedy 0,18°. Enkodér pohonu C je schopen rozlišit změnu úhlu $0,18^\circ/370$ což činí asi 0,0005° a motor B je schopen rozlišit změnu úhlu $0,18^\circ/84$, to je asi 0,0021°. Pohon A má tříkrálový snímač typu ML s 1000 dílky na otáčku a převodovkou o převodovém poměru 132:1, ten je schopen rozlišit $0,09^\circ/132$, což je asi 0,0007°. Tyto hodnoty jsou také uvedeny v tab. 1.

4.2 Návrh způsobu řízení

Nejprve bylo zapotřebí navrhnout základní algoritmus řízení. Viz obr 4.1.

Jako první program načte aktuální polohu a požadovanou polohu.

- Pak provede výpočet úhlů natočení jednotlivých ramen podle zadané polohy. Provede kontrolu dostupnosti bodu, tj. zda leží v pracovním prostoru. A podle předchozí polohy vybere kratší dráhu pro nastavení ramen. Viz. kapitola 5. Zjišťuje se zde také, zda-li po sobě jdoucí body nejsou stejné, pokud ano, manipulátor stojí.
- Mám veškeré úhly pro natočení jednotlivých motorů a zbývá přiřazení rychlostí pro jednotlivé motory tak, aby ramena došla do požadované polohy ve stejný čas. Nejdelsí dráha je přiřazena maximální rychlost daného motoru a z poměru velikostí obou drah je stanovena rychlost pro zbylé motory.
- V okamžiku kdy je manipulátor v poslední zadané poloze, program se ukončí.

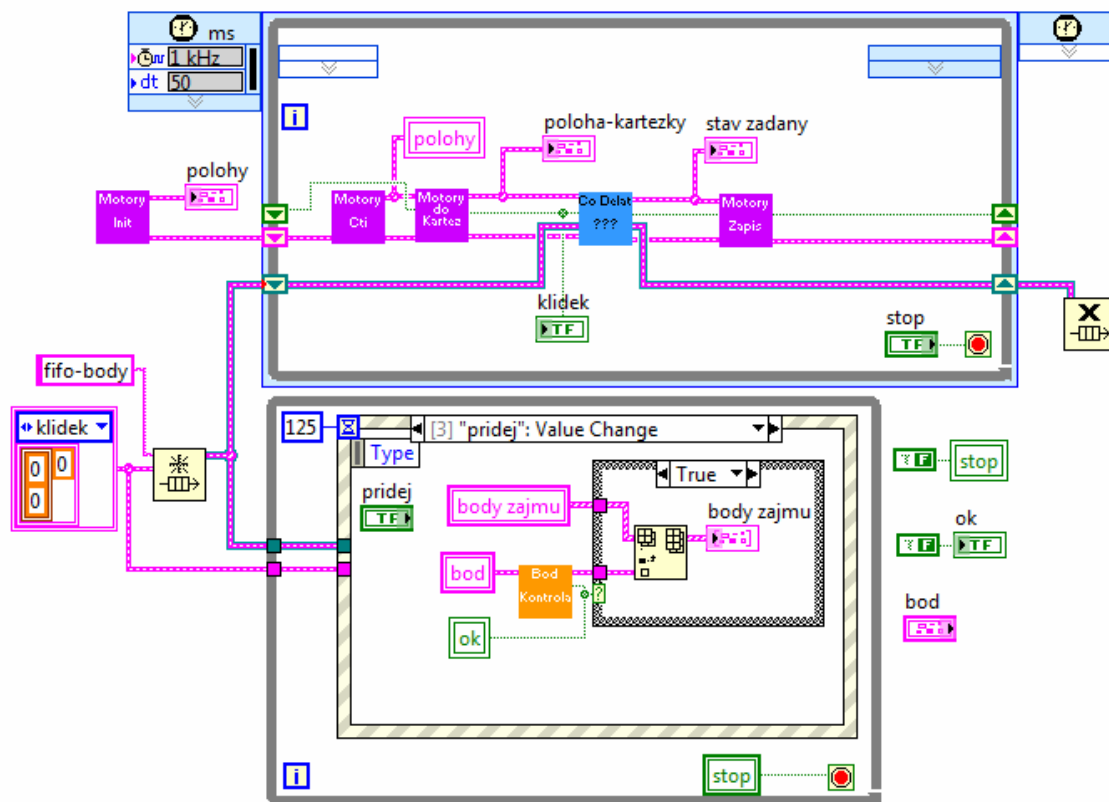


Obr. 4.1. Algoritmus řízení

4.3 Realizace řízením manipulátoru

4.3.1 Řídicí aplikace pro řízení pohybu manipulátoru

Řídicí aplikace pro řízení pohybu manipulátoru je navržena pro běh na PC. Aplikace umožňuje operátorovi ovládat manipulátor a řídit přejezdy do požadovaných bodů zájmu a tvořit jednoduché posloupnosti bodů zájmu. Aplikace každého zadaného bodu zájmu zkontroluje jeho dosažitelnost, tj. leží-li tento bod v pracovním prostoru. Je-li bod dosažitelný, přepočítá jeho souřadnice na úhly v jednotlivých osách a předá je řídicí aplikaci pohonů, která zajistí dosažení předepsaných poloh. Stav jednotlivých pohonů získává v pravidelných intervalech zpět.



Obr. 4.2. Blokový diagram řídicí aplikace pro řízení pohybu manipulátoru

Aplikace je realizována jako jednoduchý stavový automat (obr. 4.2), je použito schéma „producer-consumer“. V horní části blokového diagramu je řídicí časová smyčka, která provádí řízení a chová se jako „consumer“. Ve spodní části blokového diagramu je smyčka zajišťující obsluhu událostí od uživatele a zobrazování aktuálních informací. Tato smyčka se chová jako „producer“. Jednotlivé události jsou zpracovány a předány jako příkazy řídicí smyčce pomocí fronty FIFO.

VI je dílčí část programu, které se dále dělí na Front panel a Blokový diagram. V tomto Front panelu může uživatel zadávat požadované body, může je hned spouštět nebo přidávat do fronty a teprve potom spouštět. Zobrazuje se zde také požadované natočení jednotlivých motorů, aktuální natočení motorů, aktuální poloha v kartézských souřadnicích a také graf, ve kterém se zobrazují jednotlivé polohy ramen. Na Obr. 4.2. jsou vidět dvě smyčky. Ve spodní While smyčce, která pouze rozděluje uživatelské rozhraní od běhu vlastního programu, se nachází Event struktura, která provádí události z uživatelského rozhraní, jako například přidej bod, proved' zadaný bod, kontrola zdali je zadaný bod dosažitelný a také vykresluje aktuální polohu. Vykreslování polohy se provádí každých 125 ms. Do VI „zobraz“ vstupují aktuální souřadnice jednotlivých ramen a ty se vykreslují do grafu jako ramena manipulátoru.

4.3.1.1 Zpracovávané události

V obsluze události „bod“ se provádí kontrola pomocí VI „bod kontrola“ po každém zadání jedné ze souřadnic. V tomto VI je pomocí podmínek v matematickém skriptu vymezen pracovní prostor manipulátoru. Leží-li bod mimo pracovní prostor, na výstupu dostáváme false a kontrolka nesvítí. Ta upozorňuje obsluhu o správnosti zadaného bodu, ale také rozhoduje, zda je bod možné přidat nebo provést.

Přidání bodu se provede po kontrole v události „přidej“. Kontrola se provádí na základě výstupu z „bod kontrola“, a to true nebo false, které jsou přivedeny do Case struktury. Leží-li bod v pracovním prostoru manipulátoru, Case struktura přidá souřadnici mezi seznam bodů, které se po spuštění provedou.

Velmi podobnou funkci má událost „proved“. Ta však nepřidává bod do seznamu bodů, ale příkaz po stisknutí tlačítka přidá do fronty „fifo-body“ ze které se ve VI „Co dělat“ přečtou body, příkazy a provedou se.

Pro nastavení výchozí polohy slouží událost „výchozí poloha“. Tady provádím příkaz „reset“ a zároveň zapisuji aktuální polohu jako výchozí polohu.

Událost „spust“ vybere body zájmu, tedy body, do kterých se chce uživatel postupně dostat. Tyto body přidá do již zmíněné fronty „fifo-body“ spolu s příkazem pro motory „pracuj“.

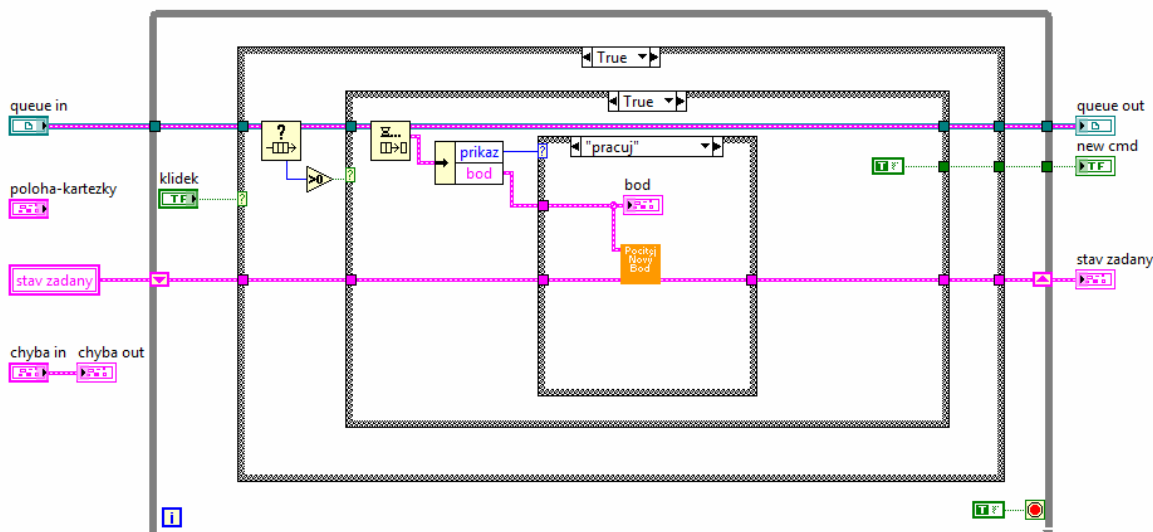
4.3.1.2 Řídící časová smyčka

Ta je zvolena nejen kvůli rozdělení chodu programu od uživatelského rozhraní, ale také kvůli časovému omezení, které je nastaveno na 50 ms. To vyplynulo z následného ladění programu, původně byly obě smyčky typu While.

Při prvním spuštění celého ovládacího programu, se z VI „motory init“ zapíše do VI „motory zápis“ počáteční nulové hodnoty a stav „reset“. Tím se dostaneme na nulovou počáteční polohu. Ve VI „motory čti“ se pouze kontroluje, zda nenastává nějaká chyba, pokud ne, skutečné stavy motorů se zapíší do VI „motory do katrez“. Tady úhlové natočení jednotlivých motorů (které je již přepočítáváno z počtu impulzů enkodéru na stupně) přepočítávám na body v kartézských souřadnicích. A zároveň, pokud jsou stavy všech tří motorů současně ve stavu „klídek“, výstupem je také stav „klídek“. Tento stav říká motorům, aby se nepohybovali, a pouze drželi polohu.

Ten používám jako vstup do VI „co dělat“, viz. Obr. 4.3.. Je-li „klídek“ ve stavu false „stav žádaný“ pouze prochází tímto VI a zapisuje se do motorů. Je-li však „klídek“ ve stavu true, provádí kontrola fronty „fifo-body“. Pokud ve frontě není žádný bod, tak do „stav žádaný“ zapisuji „klídek“. Jakmile je ve frontě bod se stavem „pracuj“, posílám tento bod do VI „počítej nový bod“ ze kterého nakonec dostávám konečný stav „stav žádaný“.

VI „počítej nový bod“ počítá rozdíl natočení skutečného a požadovaného. Tímto rozdílem dostávám natočení, o které se mají motory potočit, abych se dostal do žádané polohy, a také se kontrolují pracovní stavy pro motory. Kdyby chtěl totiž uživatel přejet do polohy, ve které se manipulátor již nachází, stav pro motory by byl „klídek“, a nic by se neprovádělo. Zároveň se nejdelšímu natočení přiřadí maximální rychlost a poměrem dráhy druhého ramene s nejdelší dráhou se vypočítá rychlost pro druhý motor tak, aby do požadované polohy dojezly obě ramena současně.



Obr. 4.3. Blokový diagram: „co dělat“

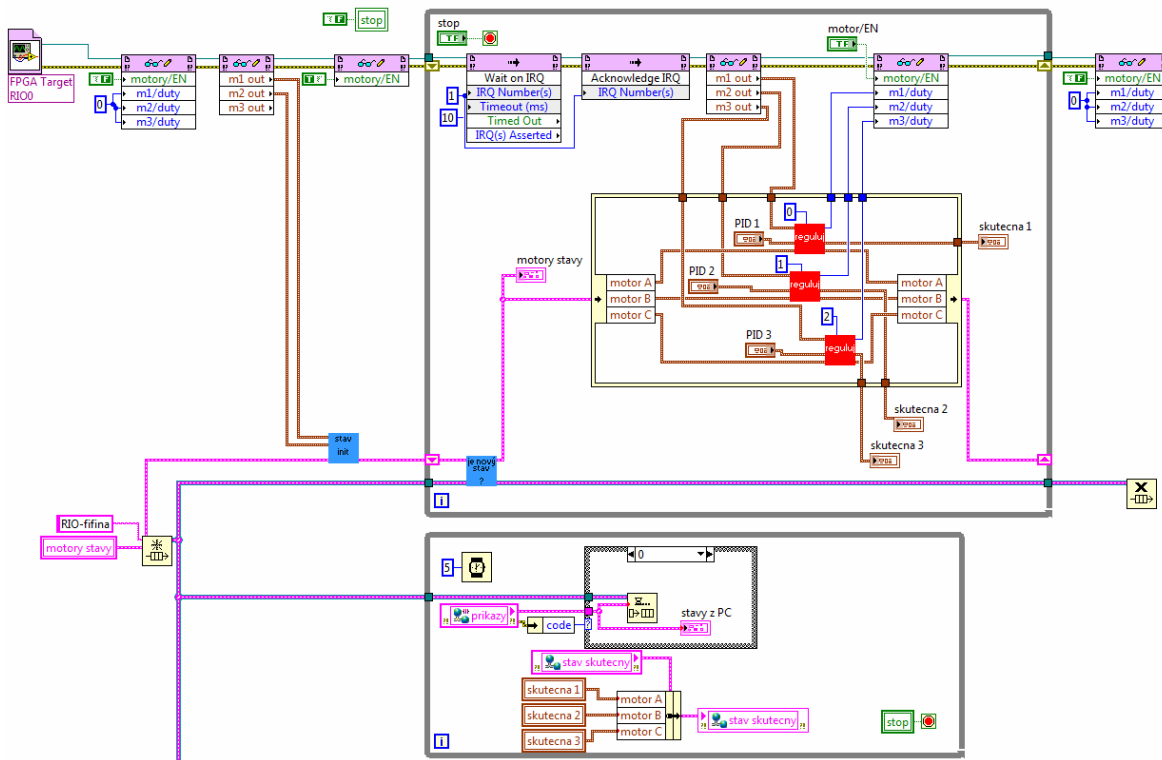
Požadované natočení je potřeba přepočítat z kartézských souřadnic zadávaných uživatelem do úhlového natočení pro motory. Tento přepočet se provádí ve VI „kartéz do motor“ zároveň s výhodnějším nastavením ramen podle předchozí polohy. Protože jak jsem již zmínil v kapitole 5, manipulátor má pro dosažení jednoho bodu dvě odlišná nastavení ramen. Je zde také ošetřeno přejíždění jednotlivých ramen přes konstrukcí omezená místa. Jako například přejíždění motorů přes podstavec. Výhodnější poloha se zjišťuje podle předešlého nastavení ramen. Porovnává se tedy předešlé natočení s úhlem δ , viz. Obr. 3.8. Je-li toto natočení větší než úhel δ , ramenu „b“ se přiřadí větší úhel ze dvojice ε_1 a ε_2 . Ramenu „a“ se přiřadí úhel ψ kladný pokud je úhel δ větší než předešlém kroku použité ε . Je-li úhel δ menší než ε , rameno „a“ se natočí o záporný úhel ψ . Úhel δ a ε díky funkci arkus kosinus vždy záporný, proto je potřeba pro kvadrant III. přičítat k úhlu δ a ε 180° a ve IV. kvadrantu měnit znaménko. Díky tomu nejsou ramena schopna jezdit přes podstavec a přes sebe sama. Natočení motoru B zadává uživatel a toto natočení se vůči hlavnímu souřadnému systému nemění. Tohoto je dosaženo jednoduchým sčítáním obou úhlů, o které se mají jednotlivá ramena natáčet a přičtením požadovaného natočení.

4.3.2 Ovládací program na CompactRIO

Motory jsou ovládány přes modul CompactRIO. Do hlavního programu VI „ovládání motoru“ běžícího na tohoto modulu posíláme přes síť potřebná data z PC pro ovládání motoru. Aby se nemohla odsílaná data (zadané body) po přenosu do modulu CompactRIO ztratit vytvořil jsem si frontu fifo „RIO-fifina“. Do ní se zapisují požadované hodnoty z PC. Dále se v této smyčce zapisují skutečné hodnoty do síťové proměnné „stav skutečný“ které jsou každých 5 ms odesílány do PC.

VI „je nový stav“ kontroluje, jestli jsou ve frontě „RIO-fifina“ nějaké zadané body. Pokud ne, zapisuje se na výstup inicializovaný stav, ve kterém se přepočítávají rychlosti a natočení podle použitých převodovek. Popis tohoto přepočtu spolu s popisem VI „reguluj“ je uveden níže. Jsou-li ve frontě VI „je nový stav“ nějaké zadané body, jsou zapsány na vstup VI „reguluj“. Spolu s nimi do tohoto VI vstupují i aktuální polohy motorů, a index

pro rozlišení jednotlivých motorů. Jedná-li se o body pro motor A zapisují se do VI „reguluj“ s indexem 0, jedná-li se o body pro motor B zapisují se do VI „reguluj“ s indexem 1. Index 2 je pro motor C.



Obr. 4.4. Blokový diagram: „ovládání motoru“

Do VI „reguluj“ které je možno vidět na Obr. 4.5. vstupuje index motoru a hodnoty z motoru. Index rozhoduje pouze mezi parametry motoru, jestli se jedná o motor A s převodovým poměrem 132:1 nebo motor B s převodovým poměrem 84:1. Zároveň je ke každému motoru přiřazen podle enkodéru počet dílků na otáčku (cpr). Z těchto informací se počítá natočení motoru φ ve stupních, natočení převodovky φ ve stupních, rychlost motoru ω ve stupních za sekundu a rychlost převodovky ω ve stupních za sekundu pomocí těchto rovnic:

$$\text{motor } \varphi = \text{natočení} \div (\text{cpr} \cdot 4 \div 360^\circ) \tag{4.1}$$

$$\text{motor } \omega = 1000 \cdot \text{rychlost} \div (\text{cpr} \cdot 16 \div 360^\circ) \tag{4.2}$$

$$\text{převodovka } \varphi = (\text{motor } \varphi \cdot \text{tp}) \div \text{ik} \tag{4.3}$$

$$\text{převodovka } \omega = (\text{motor } \omega \cdot \text{tp}) \div \text{ik} \tag{4.4}$$

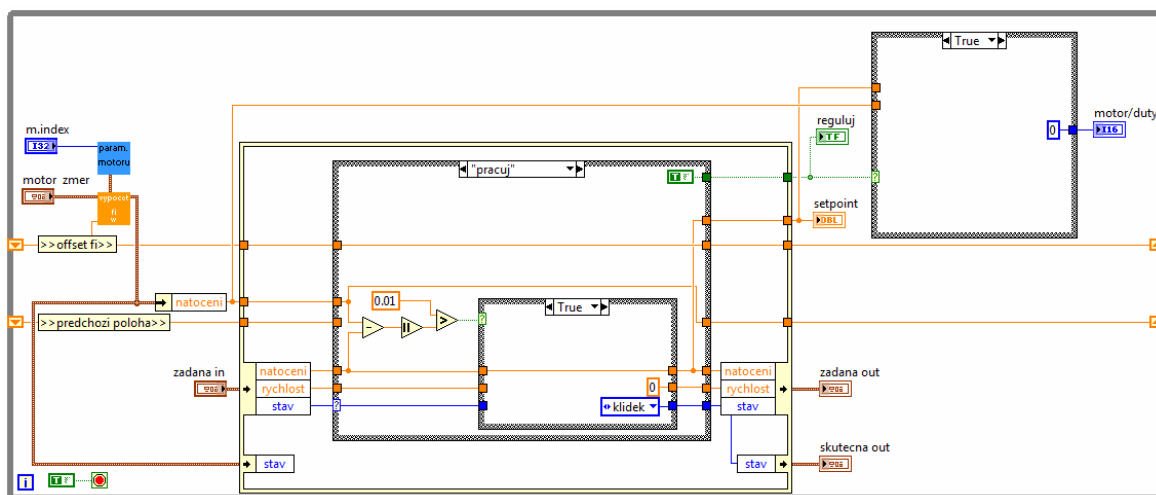
V event struktuře, v události „pracuj“ je patrné na Obr. 4.5., porovnávám absolutní hodnotu rozdílu natočení požadovaného a skutečného s konstantou 0,01. Díky tomu se regulátor nebude snažit regulovat, když se poloha bude lišit o méně než jednu setinu stupně. Tímto krokem částečně předcházím problémům vzniklým například rušením, nebo nedokonalým nastavením regulátoru. Pokud je tedy porovnáváná hodnota menší než zadaná konstanta, do motorů se odesílá stav „klídek“ a motory tak budou pouze držet požadovanou polohu. V opačném případě budou motory polohovat ramena do požadované polohy.

Při stlačení tlačítka „PAUZA“ si událost „pauza“ zapamatuje svou polohu, motory se v této poloze zastaví a regulátor udržuje tuto polohu. Toto zapamatování si bodu je provedeno přes funkci shift registr, ta si pamatuje svou hodnotu do doby, než ne přepíše novou, která již proběhla smyčkou. Tím, že touto událostí propojíme počáteční hodnotu s koncovou, bude v shift registru při každém průchodu smyčkou stejná hodnota. Stejným způsobem si pamatuje polohu bodu, do kterého měl polohovat. Spustíme-li program, manipulátor pokračuje v polohování.

Událost „klídek“ zastaví motory, dojedou-li do požadované polohy. Tato událost funguje velice podobně jako předchozí, ale hodnoty v paměti jsou totožné s hodnotami požadovanými uživatelem a rychlost je vždy nulová. Regulátor však udržuje tuto polohu. Stejně funguje i událost „zastav“, jediným rozdílem je, že se regulátor vypne a tím se vypnou i motory.

Událost „reset“ ruší veškeré požadované hodnoty. Tím musíme do regulátoru rozdíl zbylého natočení od natočení požadovaného. Tento rozdíl se odečte v regulátoru tak aby si myslel, že je již v dané poloze a regulační odchylka tak byla nulová. Nepřicházíme však o počáteční (inicializační) hodnotu. Dál pak udržuje polohu obou ramen v resetované poloze. Body, které byly ve frontě, jsou tímto krokem smazány.

Pro inicializaci slouží příkaz „init“. Ten se provede vždy při zapnutí manipulátoru. Tato událost zapíše do setpoint v PSD regulátoru nulu. Tím nastavíme polohu, ve které se manipulátor nachází, jako nulovou (počáteční) hodnotu. Nejčastěji se tento krok provádí při zapnutí manipulátoru.



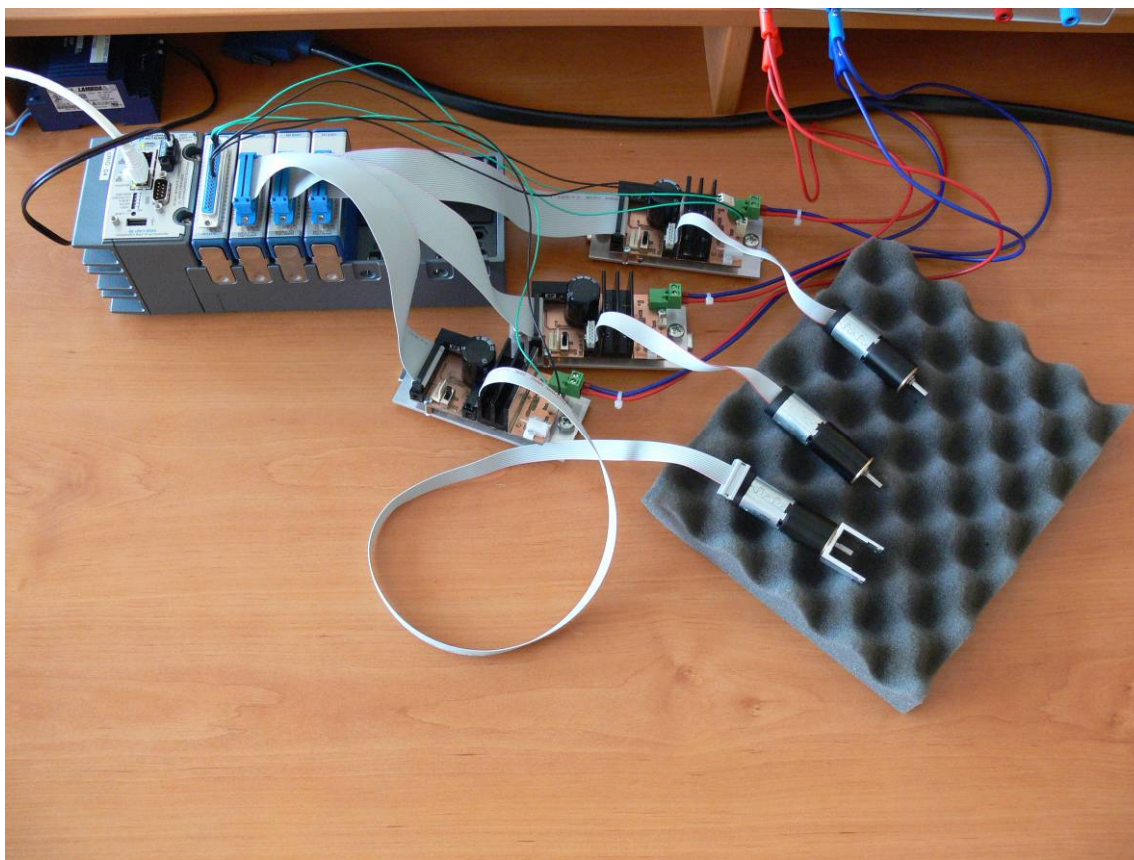
Obr. 4.5. Blokovaný diagram: „reguluj“

Pro regulaci motoru jsem použil kaskádové zapojení PSD regulátoru polohy a rychlosti. Tímto zapojením dosáhnou regulace polohy a zároveň mohu omezovat maximální rychlost pro jednotlivé motory. Jednodušší způsob pro omezení rychlosti je rozdělit jednotlivé dráhy na určitý počet dílů a mezi ně vložit časovou prodlevu. Pohyb by však byl velice trhaný, proto jsem volil polohový a rychlostní regulátor zapojený do kaskády od spolužáka Miroslava Rozkošného. Tím, že omezím maximální a minimální akční zásah, mohu u jednotlivých motorů nastavovat maximální rychlost pohybu.

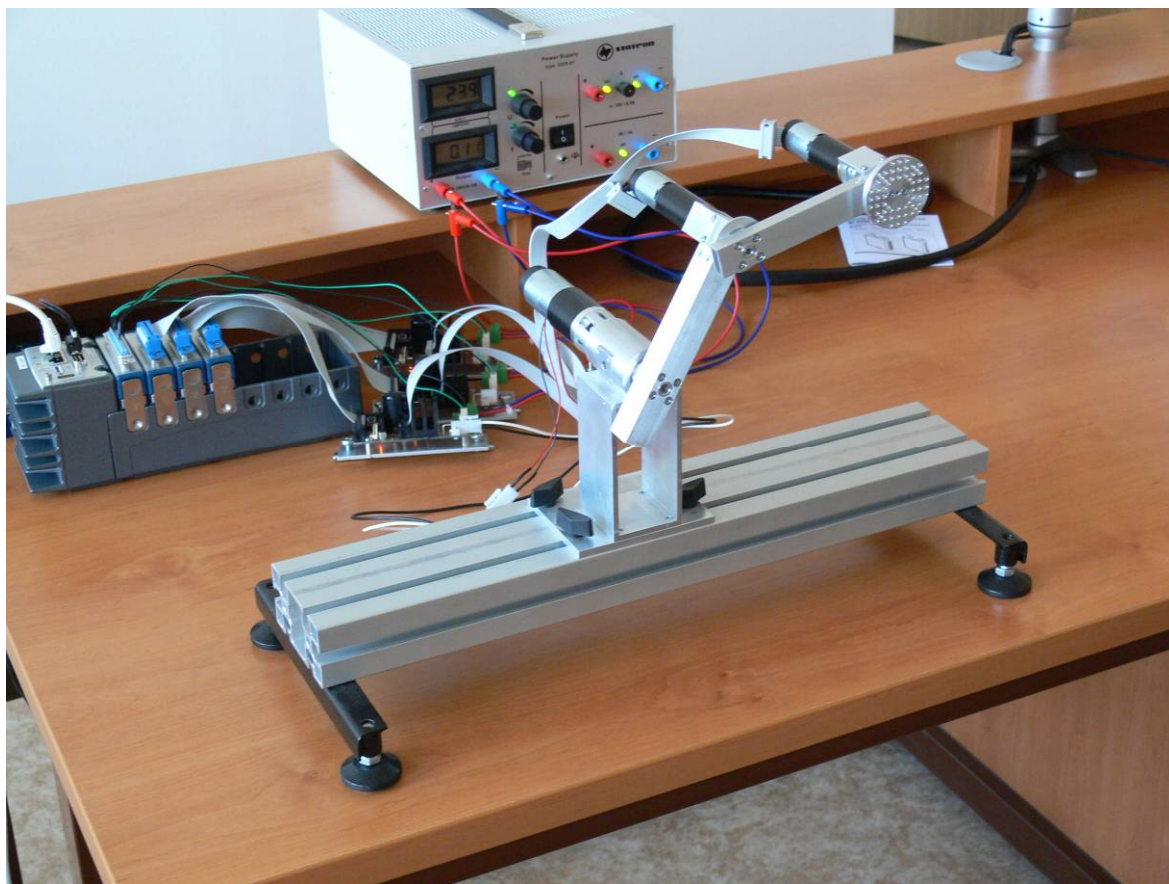
4.3.3 Postup vývoje aplikací

Vývoje aplikací byl realizován tak, aby v raných fázích vývoje nedošlo k poškození manipulátoru. Proto jsem nejprve prováděl simulaci pohybu v části programu, který běžel v PC. Tam jsem také ověřil správnost počítaných bodů, správné definice stavů pro řízení motorů a přestavování ramen tak aby nezasahovala mimo pracovní prostor.

Potom jsem vytvořil ovládací program pro ovládání motorů v CompactRIO. Motory jsem vymontoval z manipulátoru, připojil k CompactRIO a zkoušel je ovládat pouze v programu běžícím na tomto modulu (obr.4.6). Když jsem ověřil funkčnost, spojil jsem hlavní ovládací program v PC s ovládacím programem motorů v CompactRIO. Nakonec jsem motory namontoval zpět na manipulátor a testoval funkčnost vytvořeného programu (obr. 4.7).



Obr. 4.6. Ověřování funkčnosti ovládání přes CompactRIO pouze s motory.



Obr. 4.7. Ověřování funkčnosti ovládání s celým manipulátorem.

5 Závěr

Podařilo se mi vytvořit ovládací software pro manipulátor Mini-swing. Část tohoto programu běží v PC a část na modulu CompactRIO. V PC uživatel zadává souřadnice požadovaného bodu a ty pak může nechat manipulátor hned provést nebo přidat do fronty potenciálních bodů, do kterých bude po odstartování postupně přejíždět. Uživatel také může pohyb manipulátoru kdykoli zastavit a znovu spustit. Je zde také možnost resetování celého manipulátoru. Program přepočítává kartézské souřadnice, zadané uživatelem, na natočení, které porovnává se skutečným natočením motorů. Nejprve však vybere podle předešlého nastavení ramen výhodnější polohu. Protože pro stejný bod jsme schopni nastavit ramena dvěma různými způsoby. Rozdíl těchto dvou natočení je odeslán do motoru jako hodnota, o kterou se motory natočí.

Ovládací software, který vytvářím, běží z části na PC a z části na modulu CompactRIO od firmy National Instruments. V PC uživatel zadává požadované body, do kterých se má manipulátor nastavit a také povely jako například „pauza“, „přidej“ a „stop“ pro ukončení programu. Zároveň se zde přepočítávají kartézské souřadnice, které zadá uživatel, na natočení jednotlivých motorů. Podle předchozí polohy se vybírá výhodnější přestavení ramen a výsledné natočení, o které se mají motory natočit. Tyto hodnoty se sítí odesílají do modulu CompactRIO, kde běží ovládací program. Ten pomocí PSD regulátorů řídí motory a zároveň reguluje tak, aby se ramena dostala do požadované polohy a v této poloze zůstala stát.

V rámci práce se podařilo realizovat funkční software. Podařilo se mi nastavit PSD regulátory jednotlivých pohonů tak, aby manipulátor měl, podle mého názoru, dostatečnou dynamiku a přesnost. Do software bude potřeba doplnit moduly pro práci s absolutními snímači natočení jednotlivých os a zvýšit komfort uživatelského rozhraní programu.

6 Použitá literatura

- [1] STANISLAV ĎADO, MARCEL KREIDL, SENZORY a měřicí obvody, vydavatelství, Praha 1999
- [2] KARGER A, KARGEROVÁ M. *Základy robotiky a prostorové kinematiky*. 1. vyd. ČVUT, 2000. 265 s. ISBN 80-01-02183-1
- [3] SKALICKÝ, J. Elektrické servopohony. 2. vyd. Brno, 2001. 86 s. ISBN 80-214-1978-4
- [4] HACK A WII, 26.5.2010: <<http://hackawii.com/wiibot-sword-wielding-tennis-playing-wiimote-controlled-friendly-robot/>>
- [5] ParalleMIC, the Parallel Mechanisms Information Center, 26.5.2010: <<http://www.parallemic.org/Material//PI-M-850-hexapod.jpg>>
- [6] MM Průmyslové centrum, Strojírenský měsíčník, 26.5.2010: <http://www.mmspektrum.com/multimedia/image/32/3273_big.jpg>
- [7] Katalog MAXTON, Maxton Program 08/09,
- [8] AUTOMATED MOTION SYSTEMS, 29.3.2010: <<http://www.automotsys.com.au/incpulses.jpg>>
- [9] MAXON MOTOR, Planetové převodovky - maxon motor ag - UZIMEX: Špičkové technologie do automatizace a robotizace, cit. 26.5.2010: <<http://www.uzimex.cz/Vyrobce/maxon-motor-ag/Motory-DC/Rada-A.html>>

Seznam obrázků

Obr. 2.1. Schéma šestičlenného robota.....	17
Obr. 2.2. Paralelní manipulátor (hexapod).....	18
Obr. 2.3. Mechanická ruka.....	19
Obr. 3.1. Manipulátor Mini-sving	21
Obr. 3.2. MAXON DC motor	22
Obr. 3.3. Planetová převodovka MAXON.....	22
Obr. 3.4. Výstupní signály snímače.....	24
Obr. 3.5. Schéma Magneto-rezistentního enkodéru	24
Obr. 3.6. Použitý MR snímač spolu s popisem	25
Obr. 3.7. Pracovní prostor manipulátoru	1
Obr. 3.8 Zjednodušený model manipulátoru.....	27
Obr. 3.9 Dvě možná řešení nastavení manipulátoru.....	30
Obr. 4.1. Algoritmus řízení	32
Obr. 4.2. Blokový diagram řídicí aplikace pro řízení pohybu manipulátoru	33
Obr. 4.3. Blokový diagram: „co dělat“	35
Obr. 4.4. Blokový diagram: „ovládání motoru“	36
Obr. 4.5. Blokový diagram: „reguluj“	38
Obr. 4.6. Ověřování funkčnosti ovládání přes cRIO pouze s motory.....	38
Obr. 4.7. Ověřování funkčnosti ovládání s celým manipulátorem.....	39