

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

Katedra elektromechaniky a výkonové elektroniky

BAKALÁŘSKÁ PRÁCE

**Využití kinematických úloh pro řízení robotického
systému**

**vedoucí práce: Ing. Petr Beneš
autor: Lukáš Kopřiva**

2012

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta elektrotechnická
Akademický rok: 2011/2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Lukáš KOPŘIVA**
Osobní číslo: **E08B0075P**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Elektrotechnika a energetika**
Název tématu: **Využití kinematických úloh pro řízení robotického systému**
Zadávající katedra: **Katedra elektromechaniky a výkonové elektroniky**

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

1. Výpočet matic přímé a inverzní kinematiky pro řízený systém.
2. Vývoj vhodných řídicích algoritmů.
3. Implementace algoritmů do řídicí jednotky systému.



Rozsah grafických prací: **podle doporučení vedoucího**
Rozsah pracovní zprávy: **20 - 30 stran**
Forma zpracování bakalářské práce: **tištěná/elektronická**
Seznam odborné literatury:

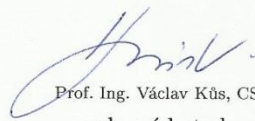
1. Upřesní vedoucí práce

Vedoucí bakalářské práce: **Ing. Petr Beneš**
Katedra elektromechaniky a výkonové elektroniky

Datum zadání bakalářské práce: **17. října 2011**
Termín odevzdání bakalářské práce: **3. června 2012**


Doc. Ing. Jiří Hammerbauer, Ph.D.
děkan




Prof. Ing. Václav Kůs, CSc.
vedoucí katedry

V Plzni dne 17. října 2011

Anotace

Předkládaná bakalářská práce je zaměřena na základní pohled pro oblast robotiky a jejího rozdělení v současnosti. Dále je tato práce zaměřena na přímou a inverzní kinematiku. Seznámení se s modelem *AL5A* od společnosti *Lynxmotion* a naprogramování jeho ovládání. Při programování řídicí jednotky bylo využito goniometrických funkcí pro výpočet úhlů pro dané servomechanismy. Dále jsou pro model vytvořené matice pro rotační a translační posuv. Závěrem je nastíněn návrh rozšíření o čidlo pro snímání poloh servomechanismů.

Klíčová slova

Robotika, kinematika, robotický systém, SSC-32, AL5A, snímač, AS5043, EAGLE, matice, MATLAB, Denavit-Hartenbergovy principy.

Abstract

Presented bachelor thesis focuses on the basic view for the area of robotics and its distribution in the present. Furthermore, this work focuses on direct and inverse kinematics. Introduction to model *AL5A* from *Lynxmotion* programming and its control. The trigonometric functions of angles for the servo were used during programming. Matrixes for rotational and translational shift were generated for model. Finally, design for extending sensor for sensing servo is outlined.

Key words

Robotics, kinematics, robotic system, SSC-32, AL5A, sensor, AS5043, EAGLE, matrix, MATLAB, Denavit-Hartenberg parametres.

Prohlášení

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

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této bakalářské práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

V Plzni dne 7.6.2012

.....

Jméno příjmení

Poděkování

Tímto bych rád poděkoval vedoucímu bakalářské práce Ing. Petru Benešovi za cenné profesionální rady, připomínky a metodické vedení práce.

V neposlední řadě mi je milou povinností poděkovat rodičům, kteří mě jak morálně, tak finančně během celého studia podporovali, a všem vyučujícím, kteří se na mém vzdělání na ZČU podíleli.

Obsah

ÚVOD.....	8
SEZNAM ZKRATEK	9
1 ROBOTIKA.....	10
1.1 Šest stupňů volnosti.....	11
1.1.1 Další rozdělení podle	12
1.2 Plánování dráhy robota.....	14
1.2.1 Přímé programování – učení.....	14
1.2.2 Nepřímé programování – offline	15
1.2.3 Přímé plánování – online	15
1.3 Pracovní režimy robota.....	15
2 APLIKACE NA MODEL.....	17
2.1 Robotické rameno.....	17
2.2 Řídící jednotka.....	17
2.3 Komunikace.....	18
2.3.1 Komunikace se servomechanismy.....	19
2.4 Přímá a inverzní kinematika	20
2.5 Goniometrické funkce na model AL5A	21
2.6 Matice pro model AL5A.....	24
2.6.1 Obecné tvary transformačních matic	25
2.6.2 Dosazení do matic.....	26
3 SNÍMAČ.....	28
3.1 Magnetický senzor AS5043	28
3.2 Návrh snímače	31
ZÁVĚR.....	33
POUŽITÁ LITERATURA	34
PŘÍLOHY.....	36

Úvod

Tato práce je rozdělena do tří částí.

První část se zabývá odvětvím robotiky, při čemž je zmíněna její historie a dále jsou popsány možnosti rozdělení dle počtu stupňů volnosti, kinematické struktury, druhů pohonů, vykonávání činností (druhu nasazení) a nesmíme opomenout ani geometrii pracovního prostoru. V robotice je také potřeba správně nastavit dráhu robota, kterou nadefinujeme přes metody ovládání (přímé programování – učení, nepřímé programování – offline, přímé plánování – online) a jeho pracovní režimy.

V druhé části je popsán model AL5A od společnosti Lynxmotion a výběr řídicí jednotky, pomocí které vedeme komunikaci s PC a servomechanismy. Dále je popsána přímá a inverzní kinematika. Použití goniometrických funkcí a matic pro rotační a translační pohyb.

Ve třetí části je návrh čidla v programu EAGLE (za pomoci magnetického senzoru AS5043), které v praxi nabízí kontrolu nad prostředím, ve kterém se robotické rameno nachází.

Seznam zkratek

UI - umělá inteligence

např. - například

tzv. - tak zvaný

tzn. - to znamená

resp. - respektive

1 Robotika

Robotika prodělala do dnešní doby obrovský vývoj a neustále se vyvíjí. Pod pojmem *robot* máme na mysli zařízení, které by nahradilo činnosti člověka v práci. Snažíme se vytvořit mechanismus, který sám získá informace o prostředí, ve kterém provádí svou definovanou činnost. Tyto informace pak musí umět sám vyhodnocovat a využívat pro svoje nasazení. [3]

Tomuto procesu se říká *automatizace* a je například nesdílou součástí automobilového průmyslu, kde se průmyslové roboty používají pro:

- svařování plamenem
- elektrický oblouk
- bodové svařování
- nanášení barev

Nasazujeme je z důvodu bezpečnosti pro člověka a jeho zdraví. [3]

V průběhu vývoje se např. po roce 1980 začaly vybavovat první průmyslové roboty prvky výzkumu UI (počítačové vidění, snímače hmatu aj.). V roce 1994 se objevuje první chirurgický robot AESOP (Automated Endoscopic System for Optimal Positioning) 100 a o dva roky poté byl na Marsu vysazen robot Sojourner. [3, 9]



Obr. 1.1 - Chirurgický robot Zeus [3]



Obr. 1.2 - robot Sojourner [3]

Spisovatel Issac Asimov v roce 1950 v knize Já robot (I, Robot) definoval zákony, které by měl ctít v tomto oboru každý vědecký pracovník.

1. Robot nesmí ublížit člověku nebo svou činností dopustit, aby člověku bylo ublíženo.
2. Robot musí uposlechnout příkazů člověka, kromě případů, kdy tato ochrana je v rozporu s prvním zákonem.
3. Robot musí chránit sám sebe před zničením, kromě případů, kdy tato ochrana je v rozporu s prvním nebo druhým zákonem. [3]

Ramena a zápěstí s efektozem nesou hlavní roli mechanické části manipulátoru. Ta je nezbytnou součástí pro průmyslové roboty. Právě manipulátor nám musí zajistit správnou polohu předmětu, který chceme pomocí průmyslového robotu přenášet v prostoru. Poloha a orientace tělesa v prostoru je definována pomocí 6 údajů. [3]

1.1 Šest stupňů volnosti

Jedná se o tři souřadnice $[x, y, z]$ referenčního bodu tělesa v základním kartézském souřadném systému a třemi úhly $[\alpha, \beta, \gamma]$ natočení referenčního systému, který je pevně spojený s tělesem vůči základnímu souřadnému systému. [3]

Šest stupňů volnosti dosáhneme za pomoci *os - kloubů*, které ovládáme díky pohonům. Robotický systém, který má 6 stupňů volnosti je považován za *univerzálního* robota. *Redundantní* robot má více než 6 stupňů volnosti, ty využívá k větší volnosti k obcházení překážek. Naopak *deficitní* robot jich má méně (3-4) pro provádění montáží prvků v rovině. [2, 3]

K dosažení jakéhokoliv bodu v dosahu pracovní oblasti robota stačí 3 osy tzv. *hlavní osy*. K nastavení úchopu (nebo naopak nástroje) do libovolné polohy nebo směru potřebujeme minimálně další 3 osy tzv. *manipulační osy*.

1.1.1 Další rozdělení podle

1) Kinematické struktury

Hlavní rozdělení kinematické struktury robotů je: *sériové, paralelní, hybridní*.

Sériové roboty mají otevřený kinematický řetězec manipulátoru (open-loop chain), zatímco *paralelní roboty* ho mají uzavřený (closed-loop chain). Oba typy řetězců kombinují *hybridní roboty*. [2]

2) Druhu pohonů

- Elektrické
- Hydraulické
- Pneumatické

[2]

V konstrukci průmyslových robotů a manipulátorů v současnosti převažují hlavně elektrické pohony. Hydraulické pohony se používají pro vysoké nosnosti a pneumatické pohony pro vysoké rychlosti. [2]

3) Vykonávaných činností a oblastí nasazení

- *Průmyslové roboty* zavádíme pro činnosti spojené s výrobou různých produktů.
- *Servisní roboty* zavádíme pro obslužné činnosti, humánních (např. zdravotnictví, domácí práce...) nebo v průmyslu či službách (stavebnictví, hlídání objektů, údržba).
- *Stacionární roboty* zavádíme pro výrobní automaty a manipulátory.
- *Mobilní roboty* zavádíme pro robotické konstrukce schopné pohybu vlastní silou. (na souši, ve vodě, ve vzduchu, ve vesmírném prostoru)

[2]

4) Geometrie pracovního prostoru

Kartézská kinematická struktura

Obsahuje 3 posuvné pohyby, při kterých nedochází ke změně orientace objektu. Pracovní prostor připomíná hranol. Struktura má nepříznivé vlastnosti pro posuvné mechanismy. Je nutné počítat s velkým prostorem pro umístění. Není tak často využívána, spíše pro podavače nebo pro obsluhu výrobních strojů. [2, 6]

Cylindrická kinematická struktura

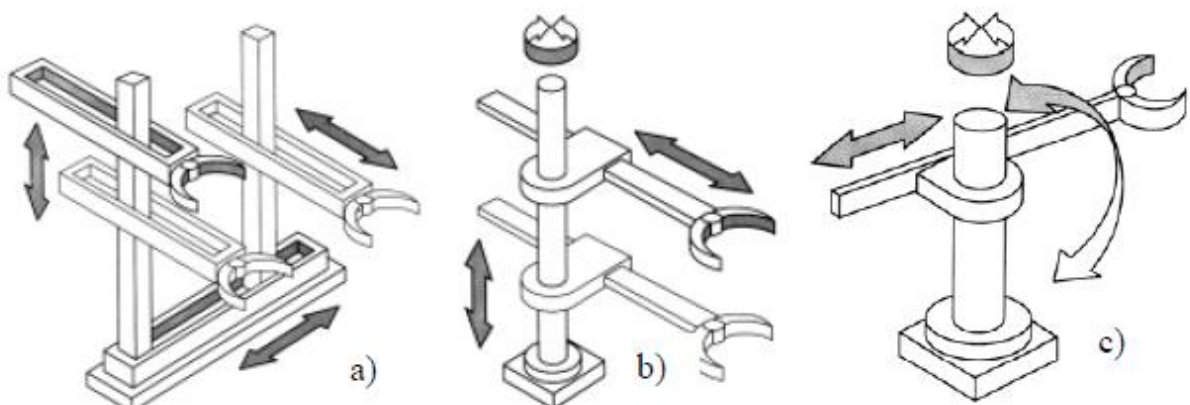
Obsahuje 1 rotační pohyb a 2 posuvné pohyby, při kterých dochází ke změně orientace objektu. Pracovní prostor připomíná válcový prstenec. Pro manévrování v tomto prostoru je potřeba velkého operačního prostoru. Využíváme ji pro roboty na obsluhu vstřikovacích strojů nebo strojů na tlakové lití. Její modifikací je montážní robot Scara. [2, 6]

Sférická kinematická struktura

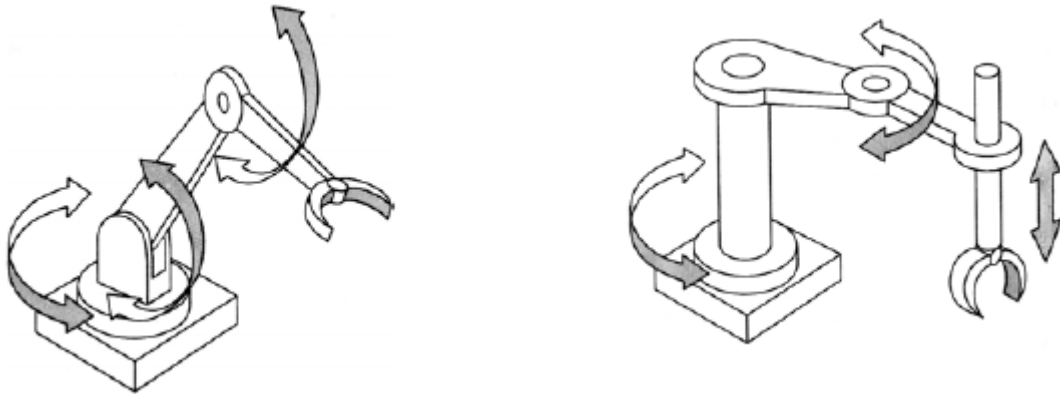
Obsahuje 2 rotační pohyby a 1 posuvný pohyb, při kterých dochází ke změně orientace objektu. Pracovní prostor připomíná kulový vrchlík. Polohu musíme definovat ve sférických souřadnicích. Největší uplatnění najde pro svařovací linky. [2, 6]

Angulární kinematická struktura

Obsahuje 3 rotační osy. Za výhody považujeme dobré dynamické vlastnosti (vysoké rychlosti manipulačních pohybů), nejméně zastavěného prostoru a jeho rozšíření (např. portál, pojezd, aj.) a technologické výhody při výrobě rotačních částí vzhledem k nižší ceně. Nevýhoda této struktury je menší manipulační prostor, kdy opakovatelná přesnost klesá s délkou rotačních ramen. Angulární systémy se v praxi používají nejčastěji. [2, 6]



Obr. 1.3 - Základní kinematické koncepce ramen průmyslových robotů
a) kartézská b) cylindrická c) sférická [3]



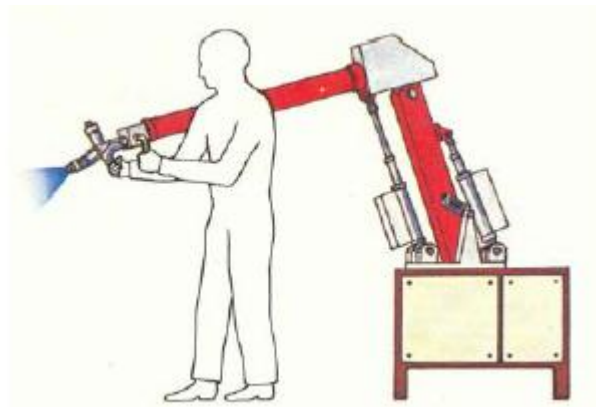
Obr.1.4 - Ukázka jiných možných kinematických koncepcí ramen průmyslových robotů [3]

1.2 Plánování dráhy robota

1.2.1 Přímé programování – učení

a) Playback

Obsluha (člověk) vede rameno a zápěstí robota po požadované dráze a žádanou rychlostí pro vykonání úkonu. Tyto úkony pak má robot sám opakovat. Takto se programují např. roboty určené pro nanášení barev. Pohyb je zapamatovaný ve formě tabulky údajů $\mathbf{q}_z(t)$. Nevýhoda je jednoznačně v přesnosti, kterou musí člověk dodržovat při provádění pohybu, neboť by robot všechny případné chyby, jako např. trhnutí ruky, zopakoval. [3]



Obr.1.5 - Přímé programování robota [3]

b) Teach in

Obsluha (člověk) pomocí tlačítek na programovacím panelu směřuje robota do požadovaných pozic v prostoru. Výhodou je, že ji může nastavit velmi přesně a po libovolně dlouhou dobu než ji nechá uložit do paměti robota. Tím se uloží poměrně málo počtu údajů $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n$. K nim se před

spuštěním robota a jeho programu musí přidat údaj o čase a případně o způsobu jak mají být propojeny body v prostoru. Nevýhodou pro obsluhu může být neznalost pohybu mezi těmito pozicemi. [3]

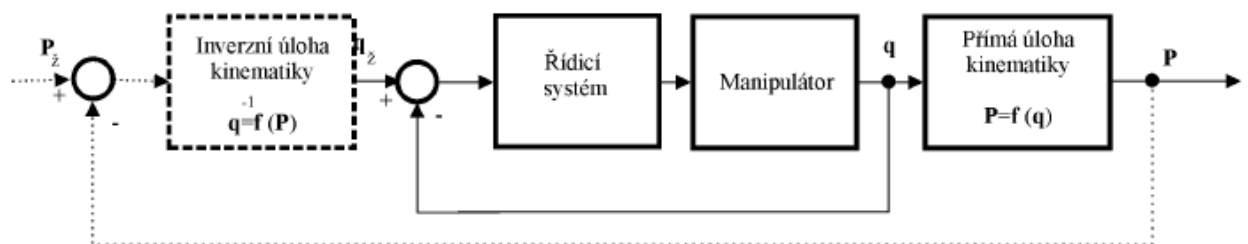
Pokud používáme přímé programování, řeší tak obsluha spolu s manipulátorem inverzní úlohu kinematiky velmi jednoduše a přirozeně. [3]

1.2.2 Nepřímé programování – offline

Při tomto způsobu je nutné dostat do programu trajektorii pohybu $\mathbf{P}_z(t)$, která je zadána formou křivek v prostoru (např. formou výkresů). Jejich parametrem je čas a ten nastavíme vzhledem k úkonu, který bude systém vykonávat. Offline řešení lze aplikovat i na inverzní kinematiku. [3]

1.2.3 Přímé plánování – online

Přímé plánování se podobá předchozí metodě, ale inverzní úloha kinematiky se řeší v reálném čase. Toho docílíme, pokud má robot snímače, které reagují na okolní prostředí a umí je využívat pro svůj pohyb. Např. robot má uchopit objekt, který se pohybuje a neznáme jeho trajektorii předem. Na *obr. 1.6* je znázorněna zpětná vazba od pozice \mathbf{P} . [3]



Obr. 1.6 - Blokové schéma řízení průmyslového robota [3]

1.3 Pracovní režimy robota

1. Ruční řízení

Provádí se pomocí přístroje pro ruční ovládání, kterým můžeme např. vyvolávat jednotlivé příkazy pro pohyb. Procesor řídicího systému tyto příkazy vyhodnocuje a zadává požadované hodnoty regulátorům pohonů. [6]

2. Zadávání programu

Do paměti řídicího systému se ukládají načtené uživatelské programy a hodnoty poloh, které jsou vyvolány přes ediční program. Všechny příkazy a funkce jsou napsány ve zvoleném programovacím jazyce. [6]

3. Automatický režim

V automatickém režimu jsou vyvolány uživatelské programy. Automatický průběh znamená, že se jeden řádek po druhém provádí bez přerušení. [6]

2 Aplikace na model

2.1 Robotické rameno

Model *AL5A* od společnosti *Lynxmotion* je robotický manipulátor skládající se z 6 stupňů volnosti: *otočná základna, rameno, loket, náklon zápěstí, rotaci zápěstí a chapadlo (efektor)*. Stavebnice obsahuje převážně hliníkové díly, některé díly jsou vyrobené z plastu. Pro ovládání analogových servomechanismů se stará elektronická deska *SSC-32* s výstupem COM port. [4, 7]

Použité servomechanismy:

Hitec HS-422 (otočná základna, loket, náklon zápěstí, rotace zápěstí, chapadlo)

Hitec HS-755HB (rameno)



Obr. 2.1 - Model AL5A [4]



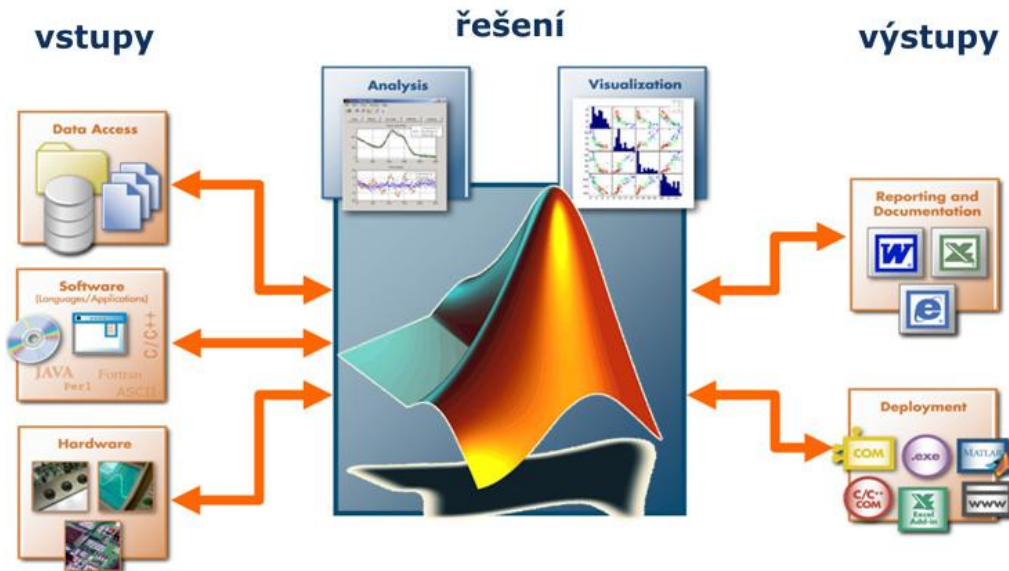
Obr. 2.2 - SSC-32 [7]

2.2 Řídící jednotka

Jako řídicí jednotka pro model *AL5A* bylo zvoleno PC. Pro ovládání a vývoj řídicích algoritmů s deskou *SSC-32* byl zvolen program *MATLAB*.

MATLAB (MATrix LABoratory) je vyvíjen pod společností *MathWorks* a nabízí plně interaktivní programové prostředí a skriptovací programovací jazyk (*čtvrtá generace*). Jedná se o multiplatformní software (operační systémy Linux (32-bit, 64-bit), Windows (32-bit, 64-bit), Mac OS X (64-bit)). *MATLAB* zvládá rychle řešit náročné matematické úlohy. [5, 10]

V našem bude zapotřebí výpočtu matic a základních goniometrických funkcí pro výpočet souřadnic umístění efektoru a úhlů (resp. hodnotu pulzů v ms pro jednotlivé servomechanismy). [5, 10]



Obr. 2.3 - MATLAB a jeho možnosti [5]

Právě díky jeho kombinaci programování a počítání s maticemi je vhodnějším nástrojem než programovací jazyk C#/C++, v němž bychom si museli vytvořit početní funkce, které *MATLAB* již obsahuje.

2.3 Komunikace

Elektronická deska *SSC-32* komunikuje prostřednictvím sériové komunikace (COM port) a převodem na USB konektor na straně PC. Na probíhající komunikaci nás upozorňuje zelená dioda, která začne svítit přerušovaně. V programu *MATLAB* musí být správně nastaveny parametry jako je rychlost komunikace, která se určí z nastavení propojek na desce a pomocí katalogového listu. Dále je potřeba správně nadefinovat zbytek parametrů k otevření COM portu:

```
%nastavení portu
port = serial('COMx');
set(port, 'Terminator', 'CR');
set(port, 'BaudRate',y,'DataBits',8,'Parity','none','StopBits',1);
port.FlowControl = 'none';
fopen(port)
```

zdrojový kód

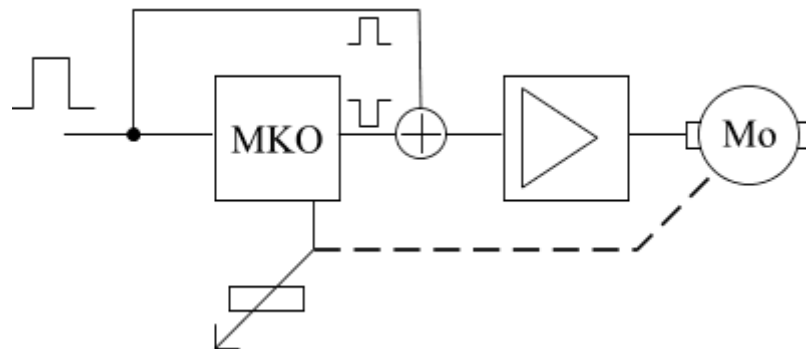
⋮

fclose(port)

kde x značí číslo pro zvolený COM port a y je rychlost komunikace (např. 115200)

2.3.1 Komunikace se servomechanismy

Uvnitř servomechanismu najdeme elektromotor s převodovkou a řídicí elektroniku. Na vstup je nutné poslat řídicí impuls, který spustí monostabilní klopný obvod. Ten vygeneruje impuls o délce odpovídající momentální poloze servomechanismu a má opačnou polaritu než je právě vstupní řídicí impuls. Porovnáním těchto dvou impulsů je rozdílový impuls, který po zesílení přes můstkový spínač způsobí roztočení elektromotoru po směru/proti směru hodinových ručiček. Elektromotor přes převodovku otáčí výstupní hřídeli a současně i potenciometrem. Ten je tu pro zpětnou vazbu do monostabilního klopného obvodu. [13]



Obr. 2.4 - Principiální schéma [13]

Pro pohyb přes SSC-32 je nutné dodržovat syntaxe:

<ch> P <pw> S <spd> ... # <ch> P <pw> S <spd> T <time> <cr>

<ch> - číslo zapojeného pinu (0-31)

<pw> - šířka impulsu v μs (500-2500)

<spd> - rychlost otáčení v $\mu\text{s}/\text{sekundu}$ pro vybraný pin

<time> - čas pro vykonání pohybu v ms (max. 65535)

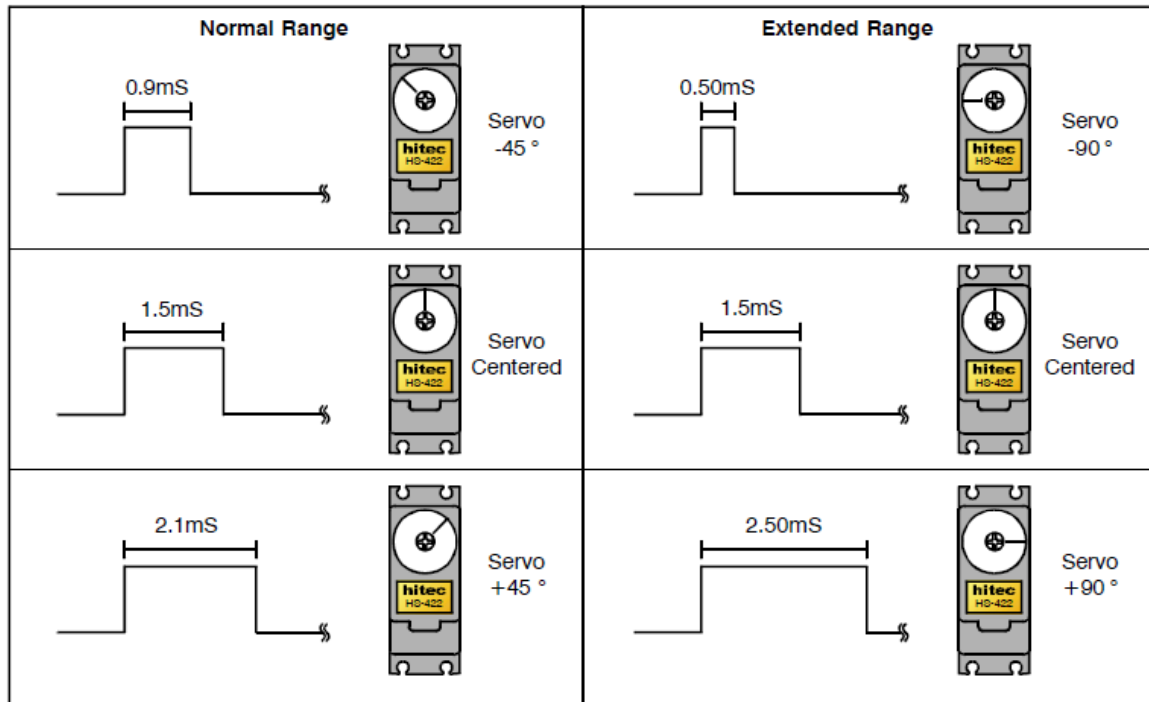
<cr> - ASCII 13, potvrzení pro vykonání pohybu

<esc> - zrušení probíhajícího příkazu (ASCII 27)

[8]

Pro příklad nastavení servomechanismu číslo 2 na 90° za 1s je nutná tato syntaxe:

#2 P1500 T1000 <cr>



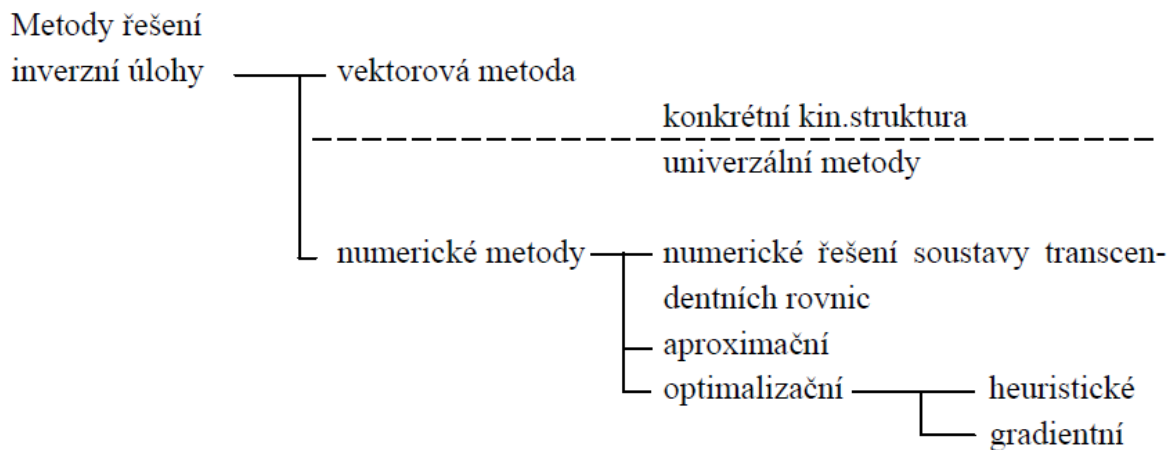
Obr. 2.5 - Princip délky impulsu a úhlu natočení [8]

2.4 Přímá a inverzní kinematika

Přímá úloha kinematiky je o stanovení pozice a orientace koncového bodu (efektoru), pokud známe jednotlivé kloubové proměnné (zobecněné souřadnice). Zatímco když hledáme jednotlivé zobecněné souřadnice a známe polohu a orientaci koncového bodu, jedná se o úlohu inverzní kinematiky. Přímou úlohu kinematiky můžeme řešit např. za pomoci goniometrických vztahů. Inverzní úloha kinematiky je obtížnější a s větším počtem stupňů volnosti se obtížnost zvyšuje. Výhoda inverzní kinematiky je v její univerzálnosti v použití (stejný algoritmus pro libovolnou kinematickou strukturu a počet stupňů volnosti). Nevýhoda je v nutnosti rychlého řešení inverzní úlohy, protože v ideálním případě probíhá v reálném čase. Způsoby metody řešení jsou nastíněny v obr. 2.7. Počet řešení pro jednotlivé struktury přímé a inverzní kinematiky je na obr. 2.6. [1]

kinematika	struktura	počet řešení	obtížnost
přímá	otevřený kin. řet.	1	snadná
	smíšený kin. řet.	$0, 1, N, \infty$	obtížná
	paralelní robot	$0, 1, N, \infty$	obtížná
inverzní	otevřený kin. řet.	$0, 1, N, \infty$	obtížná
	smíšený kin. řet.	$0, 1, N, \infty$	obtížná
	paralelní robot	1	snadná

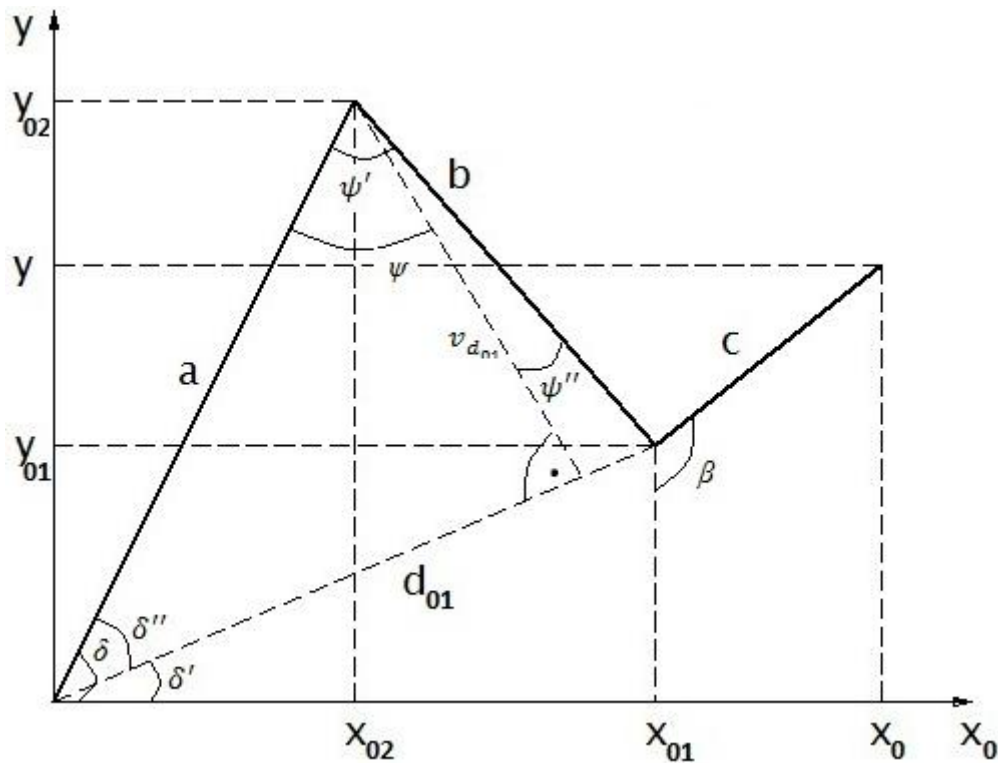
Obr. 2.6 - Počet řešení pro přímou a inverzní kinematiku [14]



Obr. 2.7 – Metody řešení pro inverzní úlohy [1]

2.5 Goniometrické funkce na model AL5A

V programu zadáme souřadnice bodu pro umístění efektoru, náklonu zápěstí, jeho natočení (x, y, z, β, γ) a pokud chceme chapadlo v otevřeném nebo zavřeném stavu. Program propočítá pomocí goniometrických funkcí a *Heronova* vzorce v *MATLABu* úhly (δ, ψ) pro servomechanismy, ty převede funkce v *MATLABu* na hodnoty v ms, které se poté pošlou přes *SSC-32*.

Obr. 2.8 - Znáornění modelu v osách x_0 a y

Délka ramen je v našem případě $a = 9,4$ cm; $b = 10,8$ cm; $c = 10$ cm.

Z obr. 3.5 se vypočte vzdálenost x_0 , která znamená délku seskládaného ramena a úhel α , který nám definuje úhel otočení základny.

$$x_0 = \sqrt{x^2 + z^2} \quad (2-1)$$

$$\alpha = \tan^{-1}(x/z) \quad (2-2)$$

Souřadnice x_{01} a y_{01} určují náklon zápěstí, z nich se dopočítá vzdálenost d_{01} , která bude dále potřeba pro výpočet úhlů.

$$y_{01} = y - c \cdot \sin(\beta - 90^\circ) \quad (2-3)$$

$$x_{01} = x_0 - c \cdot \cos(\beta - 90^\circ) \quad (2-4)$$

$$d_{01} = \sqrt{x_{01}^2 + y_{01}^2} \quad (2-5)$$

Heronův vzorec slouží k výpočtu obsahu trojúhelníku, ze kterého si dále vypočítáme výšku ($v_{d_{01}}$) na stranu d_{01} .

$$s = \frac{a+b+d_{01}}{2} \quad (2-6)$$

$$S = \sqrt{s \cdot (s - a) \cdot (s - b) \cdot (s - d_{01})} \quad (2-7)$$

$$v_{d_{01}} = \frac{2 \cdot S}{d_{01}} \quad (2-8)$$

Výpočet úhlů pro ohyb ramen:

$$\delta' = \sin^{-1}(y_{01}/d_{01}) \quad (2-9)$$

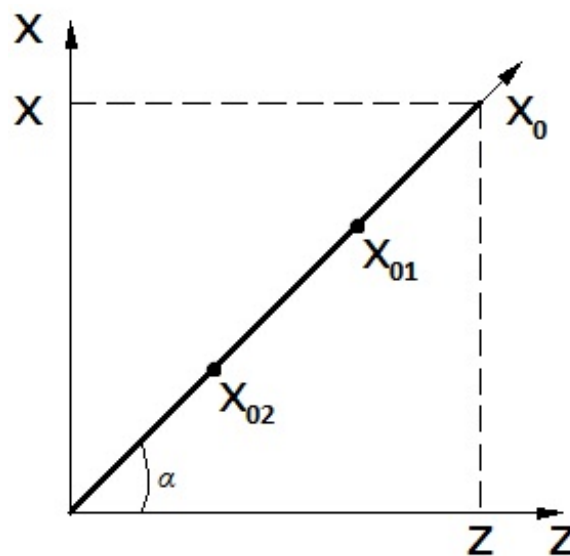
$$\delta'' = \sin^{-1}(v_{d_{01}}/a) \quad (2-10)$$

$$\delta = \delta' + \delta'' \quad (2-11)$$

$$\psi' = \cos^{-1}(v_{d_{01}}/a) \quad (2-12)$$

$$\psi'' = \cos^{-1}(v_{d_{01}}/b) \quad (2-13)$$

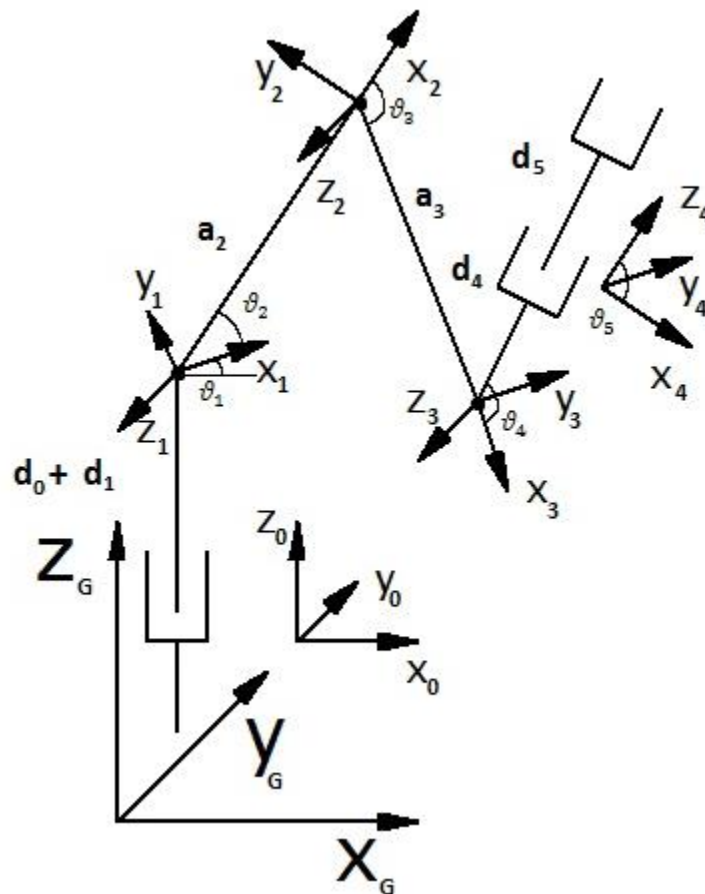
$$\psi = \psi' + \psi'' \quad (2-14)$$



Obr. 2.9 - Znázornění modelu v osách z a x

2.6 Matice pro model AL5A

K vytvoření Jakobiovy matice je potřeba za pomoci Denavit-Hartenbergova principu sestavit tabulku parametrů. Nejprve rotujeme souřadný systém podle osy z_0 (o úhel ν_1), poté podle rotace kolem osy x_1 (o úhel $\frac{\pi}{2}$) a vedeme translační posun o d_0+d_1 . Z bodu 1 do bodu 2 vedeme translační pohyb o velikosti ramene (a_2) a úhel mezi osami x_1 a x_2 (úhel ν_2). Z bodu 2 do bodu 3 vedeme translační pohyb o velikosti loktu (a_3) a úhel mezi osami x_2 a x_3 (úhel ν_3). Z bodu 3 do bodu 4 rotujeme souřadný systém podle osy z_3 (o úhel ν_4), poté podle osy x_4 (o úhel $-\frac{\pi}{2}$) a vedeme translační posun o d_4 . Nakonec rotujeme souřadný systém okolo osy z_4 (o úhel ν_5) a provedeme poslední translační pohyb o d_5 do středu chapadla. [1]



Obr. 2.10 - Schéma modelu

	\mathbf{a}_i	α_i	\mathbf{d}_i	\mathbf{v}_i
0,1	0	$\frac{\pi}{2}$	$d_0 + d_1$	v_1
1,2	a_2	0	0	v_2-90°
2,3	a_3	0	0	v_3-90°
3,4	0	$-\frac{\pi}{2}$	d_4	v_4
4,5	0	0	d_5	$-v_5$

Tab. 2.1 - Parametry

ϑ_i	úhel mezi osami x_{i-1} a x_i při otáčení kolem osy z_{i-1}
d_i	nejkratší vzdálenost (normála) mezi osami x_{i-1} a x_i , kladný směr ve směru z_{i-1}
a_i	nejkratší vzdálenost (normála) mezi osami z_{i-1} a z_i , kladný směr ve směru x_i
α_i	úhel mezi osami z_{i-1} a z_i při otáčení kolem osy x_i

Tab. 2.2 - Geometrický význam parametrů Denavit-Hartenbergova principu [1]

2.6.1 Obecné tvary transformačních matic

Obecné tvary a jejich vznik jsou vysvětleny ve zdroji [1] na straně 9.

matice pro rotaci kolem osy x o úhel α ve tvaru:

$$\mathbf{R}_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \quad (2-15)$$

matice pro rotaci kolem osy y o úhel φ ve tvaru:

$$\mathbf{R}_{y,\varphi} = \begin{bmatrix} \cos \varphi & 0 & \sin \varphi \\ 0 & 1 & 0 \\ -\sin \varphi & 0 & \cos \varphi \end{bmatrix} \quad (2-16)$$

matice pro rotaci kolem osy z o úhel ϑ ve tvaru:

$$\mathbf{R}_{z,\vartheta} = \begin{bmatrix} \cos \vartheta & -\sin \vartheta & 0 \\ \sin \vartheta & \cos \vartheta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2-17)$$

2.6.2 Dosazení do matic

Pokud dosadíme úhly z tabulky, dostaneme matice ve tvaru (kde $c_1 = \cos(\vartheta_1)$ a $s_1 = \sin(\vartheta_1)$):

$$\mathbf{R}_{x,\alpha_1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_1) & -\sin(\alpha_1) \\ 0 & \sin(\alpha_1) & \cos(\alpha_1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (2-18)$$

$$\mathbf{R}_{x,\alpha_4} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_4) & -\sin(\alpha_4) \\ 0 & \sin(\alpha_4) & \cos(\alpha_4) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \quad (2-19)$$

$$\mathbf{R}_{z,\vartheta_1} = \begin{bmatrix} c_1 & -s_1 & 0 \\ s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2-20)$$

$$\mathbf{R}_{z,\vartheta_4} = \begin{bmatrix} c_4 & -s_4 & 0 \\ s_4 & c_4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2-21)$$

Homogenní transformační matice:

$${}^0\mathbf{A}^1 = \begin{bmatrix} \mathbf{R}_{z,\vartheta_1} \cdot \mathbf{R}_{x,\alpha_1} & 0 & 0 \\ 0 & 0 & 0 & d_0 + d_1 \\ & & & 1 \end{bmatrix} = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & d_0 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-22)$$

$${}^1\mathbf{A}^2 = \begin{bmatrix} \mathbf{R}_{z,\vartheta_2} & a_2 \cdot \cos(\vartheta_2) \\ 0 & 0 & 0 & 1 \\ & & & 1 \end{bmatrix} = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 \cdot \cos(\vartheta_2) \\ s_2 & c_2 & 0 & a_2 \cdot \sin(\vartheta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-23)$$

$${}^2\mathbf{A}^3 = \begin{bmatrix} \mathbf{R}_{z,\vartheta_3} & a_3 \cdot \cos(\vartheta_3) \\ 0 & 0 & 0 & 1 \\ & & & 1 \end{bmatrix} = \begin{bmatrix} c_3 & -s_3 & 0 & a_3 \cdot \cos(\vartheta_3) \\ s_3 & c_3 & 0 & a_3 \cdot \sin(\vartheta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-24)$$

$${}^3\mathbf{A}^4 = \begin{bmatrix} \mathbf{R}_{z,\vartheta_4} \cdot \mathbf{R}_{x,\alpha_4} & 0 & 0 \\ 0 & 0 & 0 & d_4 \\ & & & 1 \end{bmatrix} = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-25)$$

$${}^4\mathbf{A}^5 = \begin{bmatrix} \mathbf{R}_{z,\vartheta_5} & 0 \\ 0 & 0 & 0 & d_5 \\ & & & 1 \end{bmatrix} = \begin{bmatrix} c_5 & s_5 & 0 & 0 \\ -s_5 & c_5 & 0 & 0 \\ 0 & 1 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-26)$$

Z homogenních transformačních matic vytvoříme vektory ze třetího sloupce (první tři řádky):

$$\mathbf{l}_{z_0} = \begin{bmatrix} s_1 \\ -c_1 \\ 0 \end{bmatrix} \quad (2-27)$$

$$\mathbf{l}_{z_1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (2-28)$$

$$\mathbf{l}_{z_2} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (2-29)$$

$$\mathbf{l}_{z_3} = \begin{bmatrix} -s_4 \\ c_4 \\ 0 \end{bmatrix} \quad (2-30)$$

$$\mathbf{l}_{z_4} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (2-31)$$

Celková homogenní transformační matice se získá postupným vynásobením:

$${}^0\mathbf{A}^5 = {}^0\mathbf{A}^1 \cdot {}^1\mathbf{A}^2 \cdot {}^2\mathbf{A}^3 \cdot {}^3\mathbf{A}^4 \cdot {}^4\mathbf{A}^5 \quad (2-32)$$

Z celkové homogenní transformační matice potřebujeme poslední sloupec, ze kterého si vyjádříme vektory \mathbf{p}_0 , \mathbf{p}_1 , \mathbf{p}_2 , \mathbf{p}_3 , \mathbf{p}_4 . (viz. příloha)

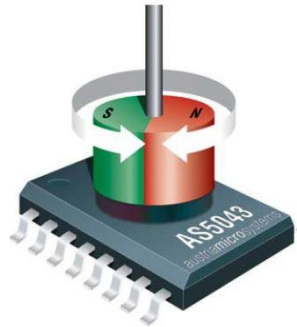
Výpočet *Jakobiovy* matice je znázorněn níže. Vzhledem k její rozsáhlosti není spočítána, protože pro konkrétní účel by se na její výsledek vytvořil program v *MATLABu*.

$$\mathbf{J} = \begin{bmatrix} \mathbf{l}_{z_0} \times (\mathbf{p} - \mathbf{p}_0) & \mathbf{l}_{z_1} \times (\mathbf{p} - \mathbf{p}_1) & \mathbf{l}_{z_2} \times (\mathbf{p} - \mathbf{p}_2) & \mathbf{l}_{z_3} \times (\mathbf{p} - \mathbf{p}_3) & \mathbf{l}_{z_4} \times (\mathbf{p} - \mathbf{p}_4) \\ \mathbf{l}_{z_0} & \mathbf{l}_{z_1} & \mathbf{l}_{z_2} & \mathbf{l}_{z_3} & \mathbf{l}_{z_4} \end{bmatrix} \quad (2-33)$$

3 Snímač

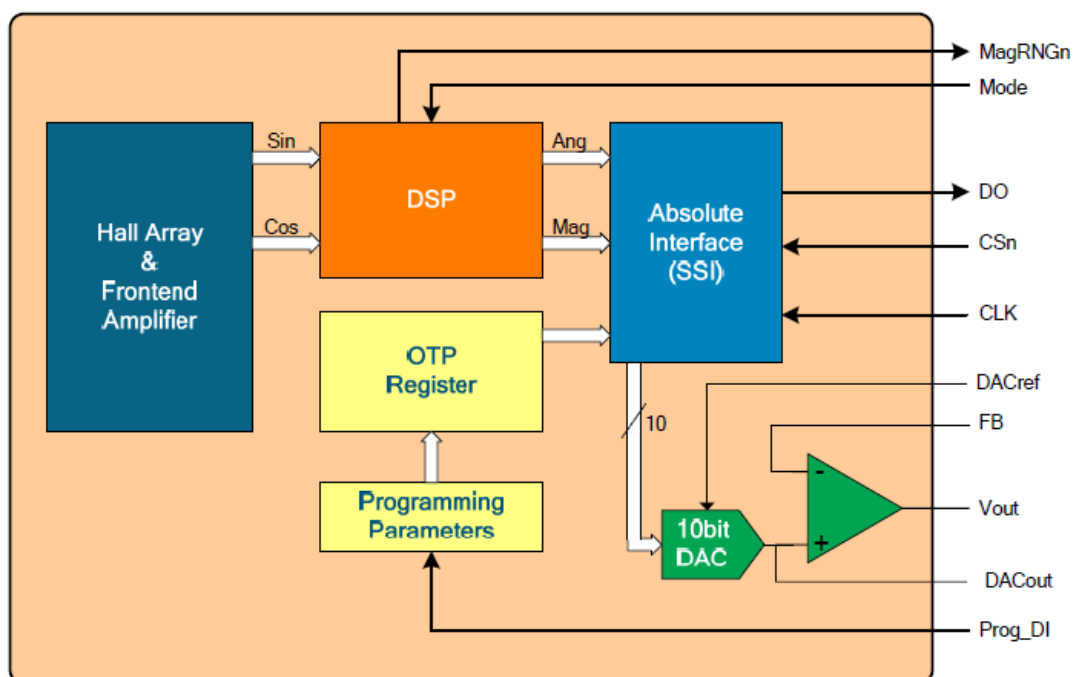
3.1 Magnetický senzor AS5043

Obvod *AS5043* je 10 bitový magnetický rotační enkodér od společnosti *austriamicrosystems*, který obsahuje bezkontaktní 10 bitový magnetický senzor. Rozsah zahrnuje 0 - 360° s *analogovým* a *digitálním* výstupem absolutní polohy. [11, 12]



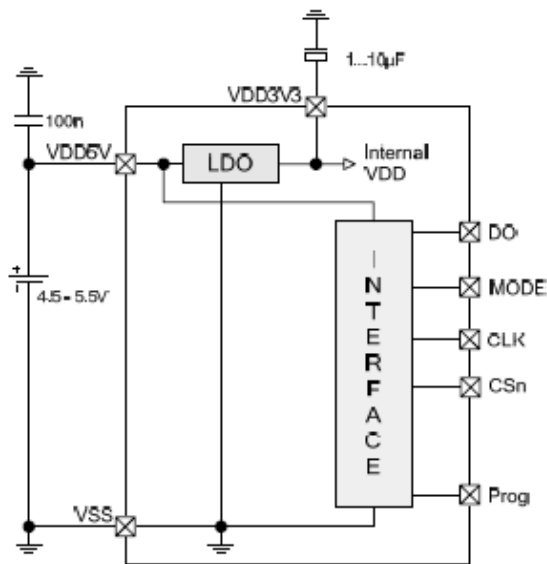
Obr. 3.1 - AS5043 [12]

AS5043 integruje Hallové sondy uspořádané do kruhu, Sigma-Delta AD převodník a signálový DSP procesor. Ten udává přesnou hodnotu natočení magnetu nad pouzdrem obvodu v rozsahu 0 – 360°. Dále získáváme pomocí synchronního sériového rozhraní *SSI* (Synchronous Serial Interface) informace o tom, zdali se magnet vzdaluje či přibližuje. Jako vstup použijeme dvupólový magnet (průměr 6 mm, výška 2,5 mm). [11, 12]

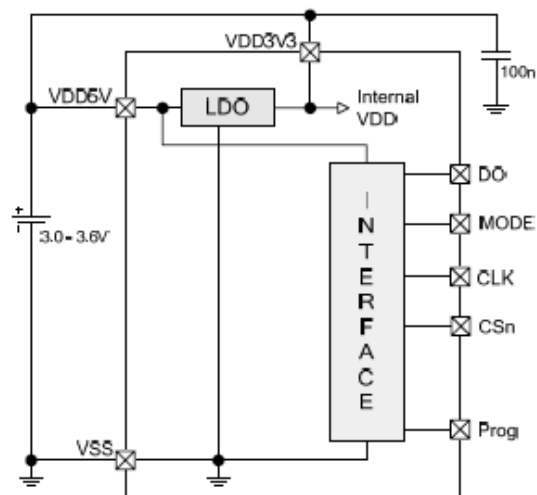


Obr. 3.2 - Blokové schéma obvodu AS5043 [12]

Napájení obvodu můžeme provozovat pro napětí 3,3V nebo pro 5V. Pro napájení 3,3V je potřeba propojit vstup (pin 16) a výstup (pin 15) na pouzdrě kvůli přemostění interního LDO stabilizátoru, který stabilizuje 3,3V pro jádro. Zbylou logiku napájíme před stabilizátorem. Pracovní rozsah teplot, pro které je zaručena funkčnost nastává od -40°C až $+125^{\circ}\text{C}$. Zatížitelnost analogového výstupu je max. $8\text{k}\Omega$ a má rozlišení 10 bitů. [11, 12]

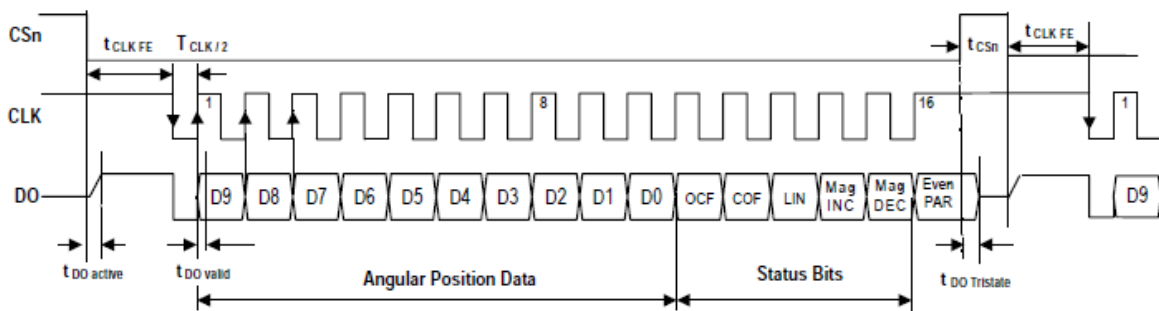


Obr. 3.3 - Napájení 5 V [12]



Obr. 3.4 - Napájení 3,3 V [12]

Všechny hodnoty můžeme vyčíst v jediném kroku za pomoci jednoduchého synchronního sériového interface (SSI), jelikož neobsahuje žádné registry. Datový výstup (DO) je v klidovém stavu ve třetím stavu, pokud aktivujeme pin CSn (přivedeme ho do L), nabyde pak datový výstup logický stav H, čímž je možné posupně odečíst všech 14 bitů na sestupnou hranu hodinového signálu, který má frekvenci max. 1MHz. [11, 12]

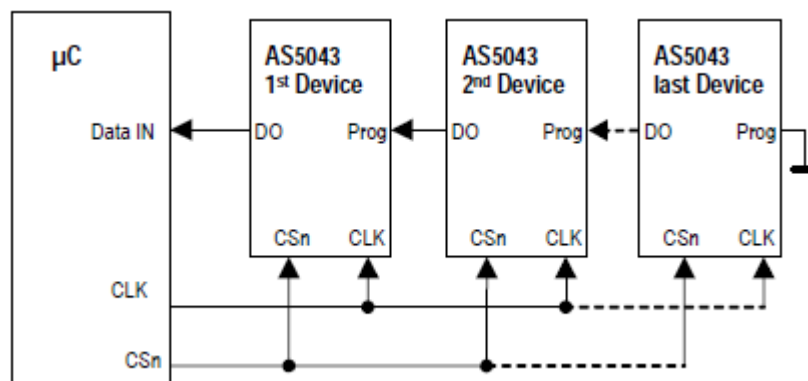


Obr. 3.5 - Komunikace s obvodem SSI protokolem [12]

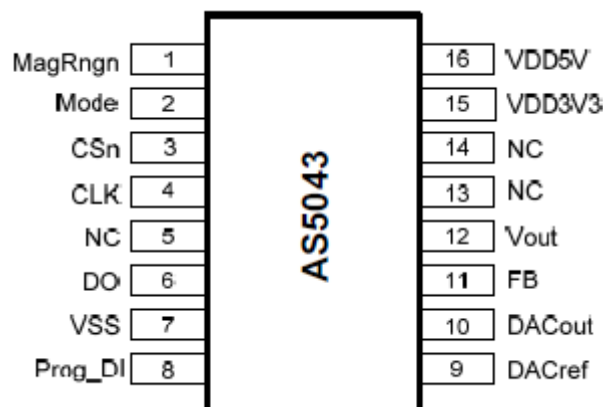
Bit	Popis
D9:D0	Hodnota absolutní pozice (D9=MSB)
OCF	Indikuje úspěšné ukončení funkce „Offset Compensation Algorithm“ a tedy platná na výstupu
COF	1 = magnet je mimo rozsah požadovaných vzdáleností
LIN	1 = výstupní hodnota není lineární
MagInc	1 = magnet se přibližuje k obvodu
MagDec	1 = magnet se vzdaluje od obvodu
Even Parity	Parita k předchozím bitům

Tab. 3.1 - Význam jednotlivých bitů [11]

Obvody můžeme dále řetězit (tzn. jejich násobné použití v zařízení). V našem případě by obvody snímaly otáčení servomechanismů a údaje poskytovaly řídicí jednotce (PC).



Obr. 3.6 - Řetězení obvodů [12]



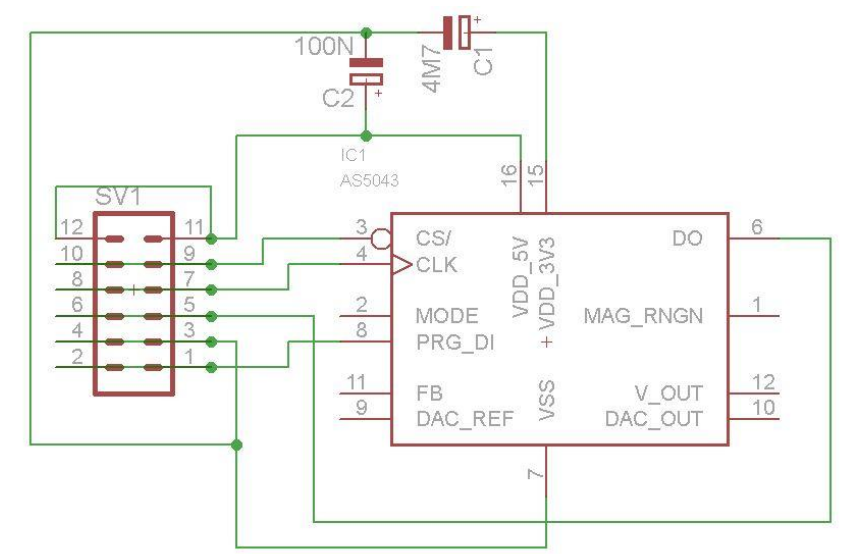
Obr. 3.7 - Pouzdro SSOP-16 [12]

Číslo pinu	Označení	Popis
1	MagRngn	Indikuje nepovolenou vzdálenost magnetu od pouzdra. Aktivní v nule
2	Mode	Volba mezi nízkým šumem (nezapojen, L) a vysokou rychlostí (H)
3	CSn	Chip Select pro SSI
4	CLK	Clock pro SSI
5	NC	nezapojen
6	DO	Data Output pro SSI
7	VSS	Zem (GND)
8	Prog_DI	OTP Programming Input
9	DACref	Vstup externí reference pro DAC
10	DACout	Výstup reference DAC (max 8kΩ)
11	FB	Feedback, negativní vstup OZ
12	Vout	Analogový výstup
13	NC	nezapojen
14	NC	nezapojen
15	VDD3V3	Výstup 3,3V regulátoru, případně pro napájení 3,3V spojit s pinem VDD5V
16	VDD5V	Vstup napájení 3,0 – 5,5V

Tab. 3.2 - Význam jednotlivých pinů [11]

3.2 Návrh snímače

Pro snímání polohy ramen pro řetězové spojení byl vytvořen návrh v programu EAGLE od společnosti CadSoft na obr. 3.8, při kterém se vycházelo z předchozích informací.

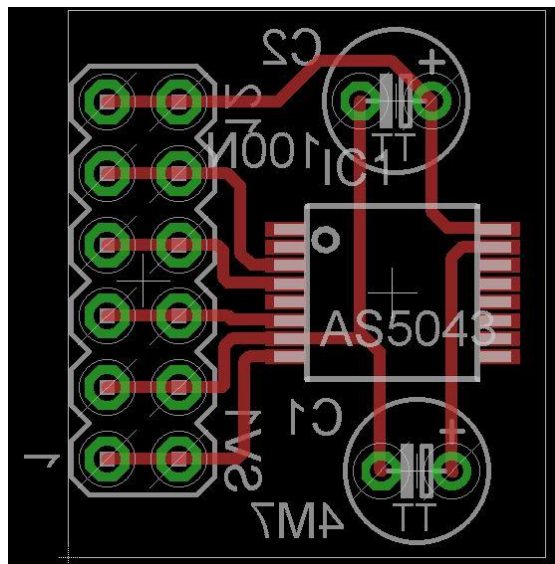


Obr. 3.8 - Schéma zapojení čidla

Napájení 5V přivádíme na piny 11-12, oproti tomu zem na piny 3-4. Hodinový signál (CLK) je přiveden na piny 7-8. Z pinů 5-6 jsou čtena data DO. Na piny 9-10 je připojen výběrový signál CSn. Programming Input je na pinech 1-2. (viz. Tab. 3.3)

PIN	Funkce
1-2	Prog_DI
3-4	VSS
5-6	DO
7-8	CLK
9-10	CSn
11-12	VDD5V

Tab. 3.3 - Propojení pinů



Obr. 3.9 - Návrh desky plošného spoje

C1	Tantalový kondenzátor	CT 4M7/10V	4M7
C2	Tantalový kondenzátor	CT 0.1M/35V	100N
IC1	Integrovaný obvod	AS5043	
SV1	Konektorové kolíky		6 pinů

Tab. 3.4 - Seznam součástek

Závěr

V této práci jsem se zabýval rozvíjející se vědou, která se nazývá *robotika* a je součástí našeho každodenního života. Rozvíjí se nejen v oblasti průmyslu, ale i v ostatních oblastech, kdy pomáhá zachraňovat a šetřit lidské životy (roboti v medicíně, ovládané moduly proti pyrotechnice, záchranné moduly, moduly pro vesmírný program aj.).

Hlavním důvodem pro nasazení automatické výroby je zefektivnění práce, ušetření nákladů, bezpečnost pro člověka a jeho zdraví. Důležité je rozdělení na manipulátory a průmyslové roboty, kdy právě druhé jmenované můžeme volně programovat.

Cílem této bakalářské práce bylo vytvoření vhodných řídicích algoritmů pro ovládání robotického systému. Jako řídicí jednotka bylo zvoleno PC. Pro komunikaci mezi PC a elektronickou deskou *SSC-32* jsem si zvolil *MATLAB*. Nabízí širokou škálu možností, od podpory programovacích jazyků, přes různé možnosti zápisu výsledků a v neposlední řadě silné počítačové (matematické) jádro. Nevýhodou je potřeba výkonnějšího PC pro plynulý běh oproti programu napsaného v jazyce *C#/C++*.

Robotické rameno (model *AL5A* od společnosti *Lynxmotion*) lze ovládat pomocí zadávání úhlu pro jednotlivé servomechanismy, kde se pomocí funkce přepočítá velikost impulsu pro vykonání pohybu. Vlastně se jedná o tzv. manuální ovládání. Dále je za pomoci funkce vyřešena možnost výpočtu úhlů pro jednotlivé servomechanismy po zadání souřadnic a úhlu polohy efektoru přes goniometrické funkce.

Kinematiku pohybu lze řešit i přes matice, kdy se využívá rotace podle os (nejčastěji podle x a z) a posunutí (translace). Důležité je si správně rozvrhnout souřadný systém pro nejsnazší zápis a minimum vykonaných pohybů. Ty se zapisují do tabulky podle Denavit-Hartenbergova principu. Tato metoda je pracnější, ale v praxi často využívána.

V poslední části je nastíněný návrh snímače pro zpracování poloh servomechanismů, které by se daly využít např. pro učení pohybu, kdy by uživatel naváděl chapadlo robota a ten si v časových intervalech zapisoval svou polohu a následně pohyb zopakoval.

Použitá literatura

- [1] MOSTÝN V., SKAŘUPA J.: *Teorie průmyslových robotů* [online]. Košice: Vienaľa Košice, 2000, [cit. 2012-01-05], ISBN 80-88922-35-6. Dostupné z WWW: <http://matescb.skvorskmal.cz/robotika_kybernetika/teorie_prumyslovych_robotu.pdf>
- [2] SKAŘUPA J.: *Průmyslové roboty a manipulátory* [online]. Ostrava: Vysoká škola báňská - Technická univerzita Ostrava, 2007, [cit. 2012-02-07], ISBN 978-80-248-1522-0. Dostupné z WWW: <http://www.elearn.vsb.cz/archivcd/FS/PRM/Text/Skripta_PRaM.pdf>
- [3] ŠOLC F., ŽALUD L.: *Robotika* [online]. Brno: Vysoké učení technické v Brně, 2002, [cit. 2012-01-07]. Dostupné z WWW: <http://matescb.skvorskmal.cz/robotika_kybernetika/VUT_Brno_Robotika.pdf>
- [4] AAHAZ: *Manipulátor AL5A a digitální serva* [online]. [cit. 2012-01-15]. Dostupné z WWW: <<http://robodoupe.cz/2012/manipulator-al5a/>>
- [5] HUMUSOFT: *MATLAB - Jazyk pro technické výpočty | Humusoft* [online]. [cit. 2012-05-04]. Dostupné z WWW: <<http://www.humusoft.cz/produkty/matlab/matlab/>>
- [6] KOHOUT L.: *Roboty a manipulátory* [online]. [cit. 2012-05-01]. Dostupné z WWW: <http://www.edumat.cz/texty/Roboty_manipulatory.pdf>
- [7] LYNXMOTION, Inc.: *Lynxmotion Robot Kits* [online]. [cit. 2012-01-08]. Dostupné z WWW: <<http://www.lynxmotion.com/default.aspx>>
- [8] LYNXMOTION, Inc.: *SSC-32 Ver 2.0* [online]. [cit. 2012-03-06]. Dostupné z WWW: <<http://www.lynxmotion.com/images/data/ssc-32.pdf>>
- [9] MPRESENT, s.r.o.: *Automa :: Roboty a automatizace v chirurgii* [online]. [cit. 2012-04-06] Dostupné z WWW: <http://www.odbornecasopisy.cz/index.php?id_document=28597>
- [10] WIKIPEDIA: THE FREE ENCYCLOPEDIA: *MATLAB - Wikipedie* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2012 [cit. 2012-04-01]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/MATLAB>>
- [11] *AS5043 - bezkontaktní 10 bitový magnetický senzor* [online]. 2008 [cit. 2011-11-25]. ISSN 1803-6007. Dostupné z WWW: <http://pandatron.cz/?504&as5043_-_bezkontaktni_10_bitovy_magneticky_senzor>
- [12] *AS5043 PROGRAMMABLE 360° MAGNETIC ANGLE ENCODER WITH ABSOLUTE SSI AND ANALOG OUTPUTS PRELIMINARY DATA SHEET* [online]. [cit. 2011-11-25]. Dostupné z WWW: <http://www.spezial.cz/pdf/AS5043_Datasheet.pdf>
- [13] *Bastlení - jak funguje modelářské servo* [online]. [cit. 2012-03-06]. Dostupné z WWW: <<http://vlastikd.webz.cz/bastl/serva.htm>>

Použité obrázky:

[14] SMUTNÝ Vladimír: *Přímá a inverzní kinematická úloha* [online]. [cit. 2012-04-05]. Dostupné z WWW: <<http://cmp.felk.cvut.cz/cmp/courses/ROB/roblec/serial-notecz.pdf>>

Přílohy

Příloha 1: Pomocné výpočty

Příloha 2: Program

Příloha 1: Pomocné výpočty

$${}^0\mathbf{A}^5 = \begin{bmatrix} \dots & \dots & \dots & a2 \cdot c2 \cdot c1 + d4 \cdot s1 - d5 \cdot (c4 \cdot (c1 \cdot c2 \cdot s3 + c1 \cdot c3 \cdot s2) + s4 \cdot (c1 \cdot c2 \cdot c3 - c1 \cdot s2 \cdot s3)) + a3 \cdot c3 \cdot c1 \cdot c2 - a3 \cdot s3 \cdot c1 \cdot s2 \\ \dots & \dots & \dots & a2 \cdot c2 \cdot s1 - c1 \cdot d4 - d5 \cdot (c4 \cdot (c2 \cdot s1 \cdot s3 + c3 \cdot s1 \cdot s2) + s4 \cdot (c2 \cdot c3 \cdot s1 - s1 \cdot s2 \cdot s3)) + a3 \cdot c3 \cdot c2 \cdot s1 - a3 \cdot s3 \cdot s1 \cdot s2 \\ \dots & \dots & \dots & a2 \cdot s2 + d0 + d1 + a3 \cdot s3 \cdot c2 + a3 \cdot c3 \cdot s2 + d5 \cdot (c4 \cdot (c2 \cdot c3 - s2 \cdot s3) - s4 \cdot (c2 \cdot s3 + c3 \cdot s2)) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{p} - \mathbf{p}_0 = \begin{bmatrix} a2 \cdot c2 \cdot c1 + d4 \cdot s1 - d5 \cdot (c4 \cdot (c1 \cdot c2 \cdot s3 + c1 \cdot c3 \cdot s2) + s4 \cdot (c1 \cdot c2 \cdot c3 - c1 \cdot s2 \cdot s3)) + a3 \cdot c3 \cdot c1 \cdot c2 - a3 \cdot s3 \cdot c1 \cdot s2 \\ a2 \cdot c2 \cdot s1 - c1 \cdot d4 - d5 \cdot (c4 \cdot (c2 \cdot s1 \cdot s3 + c3 \cdot s1 \cdot s2) + s4 \cdot (c2 \cdot c3 \cdot s1 - s1 \cdot s2 \cdot s3)) + a3 \cdot c3 \cdot c2 \cdot s1 - a3 \cdot s3 \cdot s1 \cdot s2 \\ a2 \cdot s2 + d0 + d1 + a3 \cdot s3 \cdot c2 + a3 \cdot c3 \cdot s2 + d5 \cdot (c4 \cdot (c2 \cdot c3 - s2 \cdot s3) - s4 \cdot (c2 \cdot s3 + c3 \cdot s2)) \end{bmatrix}$$

$$\mathbf{p} - \mathbf{p}_1 = \begin{bmatrix} a2 \cdot c2 \cdot c1 + d4 \cdot s1 - d5 \cdot (c4 \cdot (c1 \cdot c2 \cdot s3 + c1 \cdot c3 \cdot s2) + s4 \cdot (c1 \cdot c2 \cdot c3 - c1 \cdot s2 \cdot s3)) + a3 \cdot c3 \cdot c1 \cdot c2 \\ a2 \cdot c2 \cdot s1 - c1 \cdot d4 - d5 \cdot (c4 \cdot (c2 \cdot s1 \cdot s3 + c3 \cdot s1 \cdot s2) + s4 \cdot (c2 \cdot c3 \cdot s1 - s1 \cdot s2 \cdot s3)) + a3 \cdot c3 \cdot c2 \cdot s1 \\ a2 \cdot s2 + d0 + d1 + a3 \cdot s3 \cdot c2 + a3 \cdot c3 \cdot s2 + d5 \cdot (c4 \cdot (c2 \cdot c3 - s2 \cdot s3) - s4 \cdot (c2 \cdot s3 + c3 \cdot s2)) \end{bmatrix}$$

$$\mathbf{p} - \mathbf{p}_2 = \begin{bmatrix} a2 \cdot c2 \cdot c1 + d4 \cdot s1 - d5 \cdot (c4 \cdot (c1 \cdot c2 \cdot s3 + c1 \cdot c3 \cdot s2) + s4 \cdot (c1 \cdot c2 \cdot c3 - c1 \cdot s2 \cdot s3)) \\ a2 \cdot c2 \cdot s1 - c1 \cdot d4 - d5 \cdot (c4 \cdot (c2 \cdot s1 \cdot s3 + c3 \cdot s1 \cdot s2) + s4 \cdot (c2 \cdot c3 \cdot s1 - s1 \cdot s2 \cdot s3)) \\ a2 \cdot s2 + d0 + d1 + a3 \cdot s3 \cdot c2 + a3 \cdot c3 \cdot s2 + d5 \cdot (c4 \cdot (c2 \cdot c3 - s2 \cdot s3) - s4 \cdot (c2 \cdot s3 + c3 \cdot s2)) \end{bmatrix}$$

$$\mathbf{p} - \mathbf{p}_3 = \begin{bmatrix} a2 \cdot c2 \cdot c1 + d4 \cdot s1 \\ a2 \cdot c2 \cdot s1 - c1 \cdot d4 \\ a2 \cdot s2 + d0 + d1 + a3 \cdot s3 \cdot c2 + a3 \cdot c3 \cdot s2 + d5 \cdot (c4 \cdot (c2 \cdot c3 - s2 \cdot s3) - s4 \cdot (c2 \cdot s3 + c3 \cdot s2)) \end{bmatrix}$$

$$\mathbf{p} - \mathbf{p}_4 = \begin{bmatrix} a2 \cdot c2 \cdot c1 \\ a2 \cdot c2 \cdot s1 \\ a2 \cdot s2 + d0 + d1 + a3 \cdot s3 \cdot c2 + a3 \cdot c3 \cdot s2 + d5 \cdot (c4 \cdot (c2 \cdot c3 - s2 \cdot s3) - s4 \cdot (c2 \cdot s3 + c3 \cdot s2)) \end{bmatrix}$$

Příloha 2: Program

Funkce:

```
function data = parametry(strServo0,strServo1,strServo2,strServo3,strServo4,strServo5,strSpeed0,
strPohyb0,strPohyb1,strPohyb2,strPohyb3,strPohyb4,strPohyb5)
% nastaveni promennych pro serva
data.servo0=strServo0;
data.servo1=strServo1;
data.servo2=strServo2;
data.servo3=strServo3;
data.servo4=strServo4;
data.servo5=strServo5;
data.speed=strSpeed0;
data.pohyb0=strPohyb0;
data.pohyb1=strPohyb1;
data.pohyb2=strPohyb2;
data.pohyb3=strPohyb3;
data.pohyb4=strPohyb4;
data.pohyb5=strPohyb5;
end

function ms = PrevodUhlu(uhel)
%Prevod uhlu na hodnotu v ms pro servo.
ms=round(11.1111*uhel+500);
ms=int2str(ms);
end

function [alfa,delta,psi]=trojuhelnik(x,y,z,beta)
% vypocet uhlu pro posuv ramene

a=9.4; %delky ramen
b=10.8;
c=10;

x0=sqrt(x^2+z^2)
alfa=atand(x/z)
if(alfa<0)
    alfa=alfa+180
end
y01=y-c*sind(beta-90)
x01=x0-c*cosd(beta-90)
d01=sqrt((x01)^2+(y01)^2)
if (d01>a+b)
    d01=a+b;
end

% Heronuv vzorec
s=(a+b+d01)/2
S=sqrt(s*(s-a)*(s-b)*(s-d01)) % obsah trojuhelnika
vd01=2*S/d01 % vyska

delta1=asind(y01/d01)
delta2=asind(vd01/a)
delta=delta1+delta2
psi1=acosd(vd01/a)
psi2=acosd(vd01/b)
psi=psi1+psi2
end
```

Hlavní program:

```

%Skajnet v0.4
%ovladani pomoci klavesnice
while 1
clc;
clear all;

%nastaveni portu
port = serial('COM8');
set(port, 'Terminator', 'CR');
set(port, 'BaudRate',115200,'DataBits',8,'Parity','none','StopBits',1);
port.FlowControl = 'none';
fopen(port)

%parametry
data=parametry('#0P','#1P','#2P','#3P','#4P','#5P','S750',' ',' ',' ',' ');

%menu
vypis=sprintf('--- Skajnet ---\n 1. manual\n 2. prepocet pres gon. fce\n 3. KONEC');
disp(vypis);
fprintf(port,'#0P1500S750'); %posilani na SSC-32
pause(1);
fprintf(port,'#1P2120S750');
pause(1);
fprintf(port,'#2P1740S750');
pause(1);
fprintf(port,'#3P1335S750');
pause(1);
fprintf(port,'#4P1500S750');
pause(1);
fprintf(port,'#5P885S750');
menu=input("");

%manual
if(menu==1)
while 1
clc;

vypis=sprintf('--- Skajnet ---\n Manual - ovladani:\n 1. zakladna\n 2. rameno a\n 3. rameno b\n 4. rameno
c\n 5. natoceni\n 6. chnapka\n 7. GO\n 8. smazani promennych \n 9. KONEC');
disp(vypis);
rozhodnuti=input("");
switch rozhodnuti
case 1
%zakladna
uhel=input('Zadejte uhel: ');
ms0=PrevodUhlu(uhel);
data.pohyb0=strcat(data.servo0,ms0);

case 2
%rameno a
uhel=input('Zadejte uhel: ');
%Prevod uhlu na hodnotu v ms pro servo.
ms1=round(11.1111*uhel+910);
ms1=int2str(ms1);
data.pohyb1=strcat(data.servo1,ms1);

case 3
%rameno b

```



```
uhel=input('Zadejte uhel: ');
uhel=180-uhel;
ms2=PrevodUhlu(uhel);
data.pohyb2=strcat(data.servo2,ms2);

case 4
%rameno c
uhel=input('Zadejte uhel: ');
%Prevod uhlu na hodnotu v ms pro servo.
ms3=round(11.1111*(uhel)+710);
ms3=int2str(ms3);
data.pohyb3=strcat(data.servo3,ms3);

case 5
%otoceni hlavice
uhel=input('Zadejte uhel: ');
ms4=PrevodUhlu(uhel);
data.pohyb4=strcat(data.servo4,ms4);

case 6
%chnapka
uhel=input('Zadejte uhel: ');
ms5=PrevodUhlu(uhel);
data.pohyb5=strcat(data.servo5,ms5);

case 7
pohyb=strcat(data.pohyb0,data.speed) %slozeni pohybu a rychlosti
fprintf(port,pohyb) %posilani na SSC-32
pause(1)
pohyb=strcat(data.pohyb1,data.speed)
fprintf(port,pohyb)
pause(1)
pohyb=strcat(data.pohyb2,data.speed)
fprintf(port,pohyb)
pause(1)
pohyb=strcat(data.pohyb3,data.speed)
fprintf(port,pohyb)
pause(1)
pohyb=strcat(data.pohyb4,data.speed)
fprintf(port,pohyb)
pause(1)
pohyb=strcat(data.pohyb5,data.speed)
fprintf(port,pohyb)

case 8
%smazani obsahu promennych
data=parametry('#0P','#1P','#2P','#3P','#4P','#5P','S750',' ',' ',' ',' ');

case 9
%ukoncení volby manual
break;

otherwise
%chyba volby
disp('Zkus to jeste raz :');
end
end
end

%ovladani pomoci gon. fci
```

```

if (menu==2)
    while 1
        clc;
        %parametry
        data=parametry('#0P','#1P','#2P','#3P','#4P','#5P','S750',' ',' ',' ',' ');

        vypis=sprintf('--- Skajnet ---\n ');
        disp(vypis);
        x=input('Zadejte souradnici x:');
        y=input('Zadejte souradnici y:');
        z=input('Zadejte souradnici z:');
        beta=input('Zadejte uhel pro ohyb zapesti:');
        otoceni=input('Zadejte uhel otoceni zapesti:');
        volba_klepeta=input('1. Efektor otevreny| 2. Efektor stazeny: ');

        [alfa,delta,psi]=trojuhelnik(x,y,z,beta)

        % zakladna
        uhel=alfa;
        ms0=PrevodUhlu(uhel);
        data.pohyb0=strcat(data.servo0,ms0);

        % rameno a
        uhel=delta;
        ms1=round(11.1111*uhel+910);
        ms1=int2str(ms1);
        data.pohyb1=strcat(data.servo1,ms1);

        % rameno b
        uhel=180-psi;
        ms2=PrevodUhlu(uhel);
        data.pohyb2=strcat(data.servo2,ms2);

        % rameno c
        uhel=270-beta
        ms3=round(11.1111*uhel+710);
        ms3=int2str(ms3);
        data.pohyb3=strcat(data.servo3,ms3);

        % otoceni hlavice
        uhel=otoceni;
        ms4=PrevodUhlu(uhel);
        data.pohyb4=strcat(data.servo4,ms4);

        % chnapka
        if(volba_klepeta==1)
            fprintf(port,'#5 P885 S750')
            pause(1);
        end
        if(volba_klepeta==2)
            fprintf(port,'#5 P2250 S750')
            pause(1);
        end

        % pohyb
        pohyb=strcat(data.pohyb0,data.speed) % slozeni pohybu a rychlosti
        fprintf(port,pohyb) % posilani na SSC-32
        pause(1);

```

```
pohyb=strcat(data.pohyb1,data.speed)
fprintf(port,pohyb)
pause(1);

pohyb=strcat(data.pohyb2,data.speed)
fprintf(port,pohyb)
pause(1);

pohyb=strcat(data.pohyb3,data.speed)
fprintf(port,pohyb)
pause(1);

pohyb=strcat(data.pohyb4,data.speed)
fprintf(port,pohyb)
pause(1);

%pohyb=strcat(data.pohyb5,data.speed)
%fprintf(port,pohyb)

pokracovani=input('Zadat nove souradnice? (1. Ano | 2. Ne) ');
if (pokracovani==2)
    break;
end
end

%konec
if (menu==3)
    exit;
end

%zavreni portu
fclose(port)
delete(port)
clear port

end
```