Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra kybernetiky

DIPLOMOVÁ PRÁCE

Plzeň 2016

Adam Zíka

PROHLÁŠENÍ

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

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

V Plzni dne 06/05/16 .....

Poděkování

Děkuji vedoucímu diplomové práce Ing. Martinu Švejdovi za trpělivost a výborné vedení při jejím zpracování.

Anotace

Cílem diplomové práce je nalezení vhodné metody kalibrace sériových manipulátorů. Nejprve jde o prostudování teoretické možnosti kalibrace sériových manipulátorů a poté o navržení algoritmu pro reálného robota. Práce se zabývá jednotlivými možnostmi kalibrace včetně počtu senzorů a jejich polohy. Používá se též prostředí *simmechanics* pro jeho využití při kalibraci. Následuje aplikace dříve získaných poznatků na reálném robotovi typu RRR pomocí senzorů měření vzdálenosti. Všechny naměřené a spočtené hodnoty byly analyzovány a vyhodnoceny. Zjistilo se, že vybraná metoda je použitelná na daného robota, avšak pro její obecnější použití by bylo vhodné ji do budoucna vylepšit o přidání dalšího senzoru.

Klíčová slova

kalibrace, sériový manipulátor, kinematika, robot, *simmechanics*, rotační kloub, senzor měření vzdálenosti, trigonometrie, geometrie v prostoru, řídicí systém reálného času *REX*, Denavitova-Hartenbergova notace

Annotation

The main topic of the diploma thesis is to find a suitable method for calibration of the serial manipulator. First part is about theoretical possibilities of calibration. Next part is focused on an algorithm for the real robot. The thesis discusses the number of sensors and their position. Program *simmechanics* is used, because of its help with calibration. It is followed by applying the algorithm on real robot which is RRR type. From several methods the best one has been chosen. The resultant data are then evaluated. It has been found that the chosen method is useful on this type of robot, but for its general use the adding of the third sensor would be appropriate.

Key words

calibration, serial manipulator, kinematics, robot, *simmechanics*, rotational joint, sensors for distance measurement, trigonometry, geometry in three dimensional space, *REX* system, Denavit-Hartenberg notation

Obsah

Úvod 1

Kapitola 1: Kinematika robotů

1.1 Typy kloubů 2

1.2 Rozdělení robotů 2

Kapitola 2: Kalibrace obecného robota RRR

2.1 Možnosti kalibrace 3

2.2 Kalibrace s jedním měřicím zařízením 4

2.3 Kalibrace se dvěma měřicími zařízeními 11

2.4 Vzájemná poloha měřicích zařízení 12

2.5 Měřicí zařízení umístěná přímo na robotovi 15

Kapitola 3: Model v *simmechanics*

3.1 Prostředí *simmechanics* 17

3.2 Postup metody 18

Kapitola 4: Aplikace na konkrétním robotovi

4.1 Reálný robot 22

4.2 Senzor měření vzdálenosti 23

4.3 Umístění robota 25

4.4 Ovládací prostředí 27

Kapitola 5: Kalibrace reálného robota

5.1 Natočení první osy 30

5.2 Denavitova-Hartenbergova notace 37

5.3 Transformační matice 37

5.4 Výpočet druhé a třetí osy 39

Kapitola 6: Výsledky

6.1 Naměřené hodnoty 42

6.2 Přepočet výsledků na reálného robota 43

6.3 Vyhodnocení 44

Závěr 45

Úvod

Tato diplomová práce je zaměřena na nalezení vhodné metody kalibrace sériových manipulátorů. Praktická část práce byla vypracována na robotovi vlastní konstrukce v budově NTIS Západočeské univerzity v Plzni. Mým úkolem bylo nejprve prostudovat teoretické možnosti kalibrace obecných sériových manipulátorů a poté navrhnout algoritmus pro reálného robota.

Tematicky je práce rozdělena do šesti kapitol. První kapitola popisuje obecně kinematiku robotů a vysvětluje pojmy. V druhé kapitole se uvažují jednotlivé možnosti kalibrace včetně počtu senzorů a jejich polohy. Třetí kapitola se zabývá prostředím *simmechanics* a jeho využitím pro kalibraci. Čtvrtá kapitola obsahuje informace o reálném robotovi, senzorech a řídicím systému. V páté kapitole se věnuji kalibraci reálného robota a popisu algoritmu. Poslední šestá kapitola obsahuje naměřené a spočtené hodnoty následované přepočtem výsledků a vyhodnocením.

V celé této diplomové práci je mnohokrát použito slovo robot. Dle pravidel českého pravopisu lze používat jak tvar neživotný tak i životný. [6] Mně je mnohem bližší tvar životný, kterého se budu držet.

1) Kinematika robotů

1.1 Typy kloubů

V kinematické struktuře se obvykle setkáváme s dvěma nejčastěji používanými typy kloubů, které nám zajišťují pohyb robotů: rotační a posuvný. V našem případě je dle zadání použita trojice kloubů rotačních. Co se týče výskytu jednotlivých typů kloubů v průmyslové automatizaci, je drtivá většina z nich rotačních, což přibližuje naši úlohu blíže reálnému světu.

1.2 Rozdělení robotů

Pro průmyslové roboty existuje spousta kritérií, dle kterých je lze rozřadit do různých kategorií. Nás zajímá to, které celou skupinu dělí na sériové a paralelní. Naším úkolem bylo kalibrovat robota z kategorie sériových, jejichž kinematický řetězec je tzv. otevřený a lze ho popsat acyklickým grafem (bez smyček). Hrany představují jednotlivé klouby a uzly jsou ramena robota.[1] Sériový robot je vidět na obrázku 1.

Obrázek 1: Sériový manipulátor [2]

Opakem jsou roboti paralelní, které se v průmyslu používají především v oblasti paletování. Příklad paralelního robota lze vidět na obrázku 2.

Obrázek 2: Paralelní manipulátor [3]

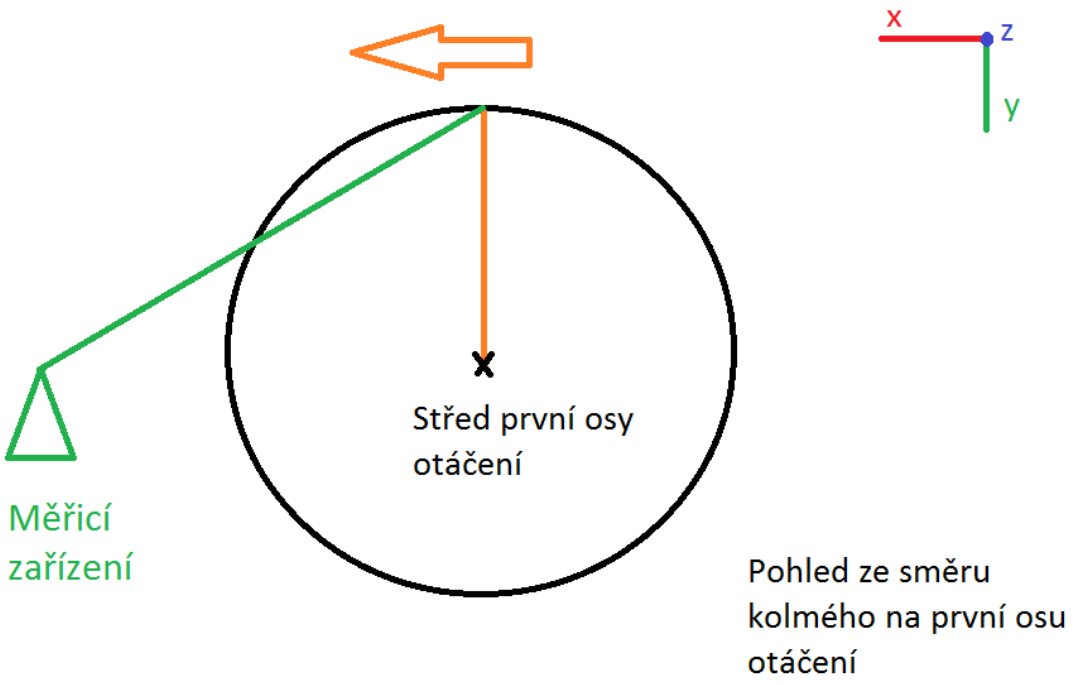
2) Kalibrace obecného robota RRR

2.1 Možnosti kalibrace

Dle zadání si lze objekt kalibrace představit jako první tři klouby klasického průmyslového robota. Využijeme informace, že osa *z* ze souřadného systému robota je shodná s osou otáčení prvního kloubu. Dále vycházíme z předpokladu, že máme k dispozici dvě měřicí zařízení. Jedno z nich umístíme tak, abychom se k němu dostali od robota ve směru osy x. Již s tímto jedním zařízením jsme schopni kalibrovat robota – jeho kloubové souřadnice. Vycházíme z předpokladu, že robot umí otočit jakoukoli svoji osou o určený úhel, přestože nezná absolutní postavení svých kloubů. Nejprve jsme si pomocí teoretických algoritmů připravili půdu pro praktické použití na reálném robotovi.

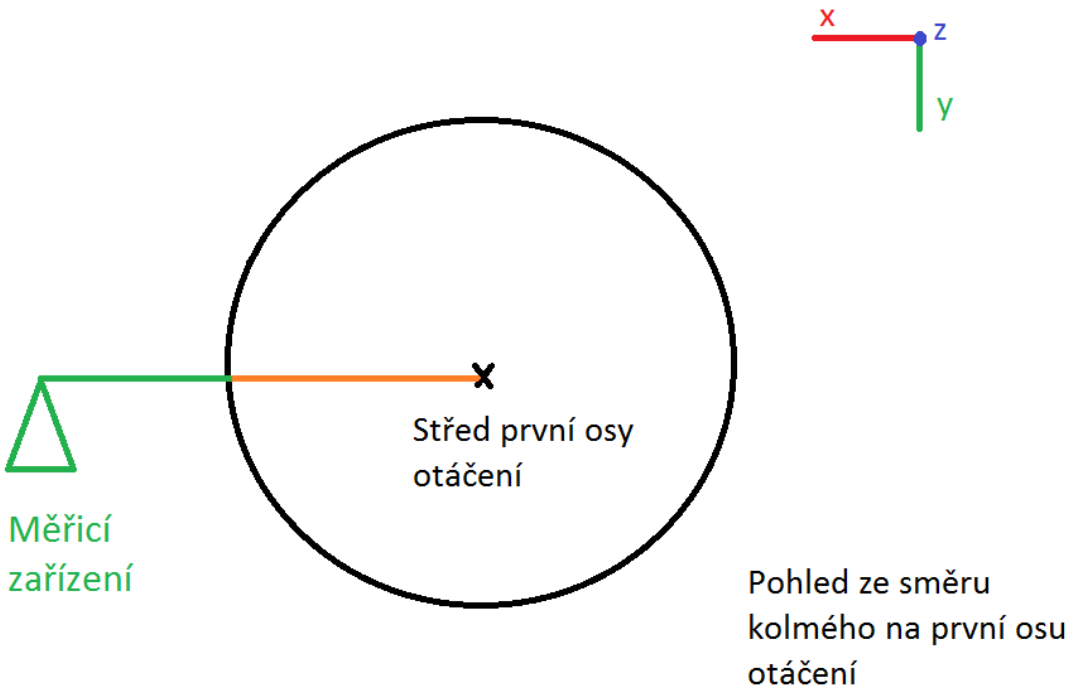
2.2 Kalibrace s jedním měřicím zařízením

Začněme nejprve obecným případem. Speciálním zvláštním polohám, mechanickým omezením a výjimkám se budeme věnovat později. Nejprve otáčíme robotem tak, aby se měřená vzdálenost TCP (*Tool Center Point* – koncový bod robota) zmenšovala – TCP se přibližuje k měřicímu zařízení. K tomu slouží otáčení pouze první osy. Tímto zajistíme, že se TCP pohybuje po kružnici, což je patrné z obrázku 3.



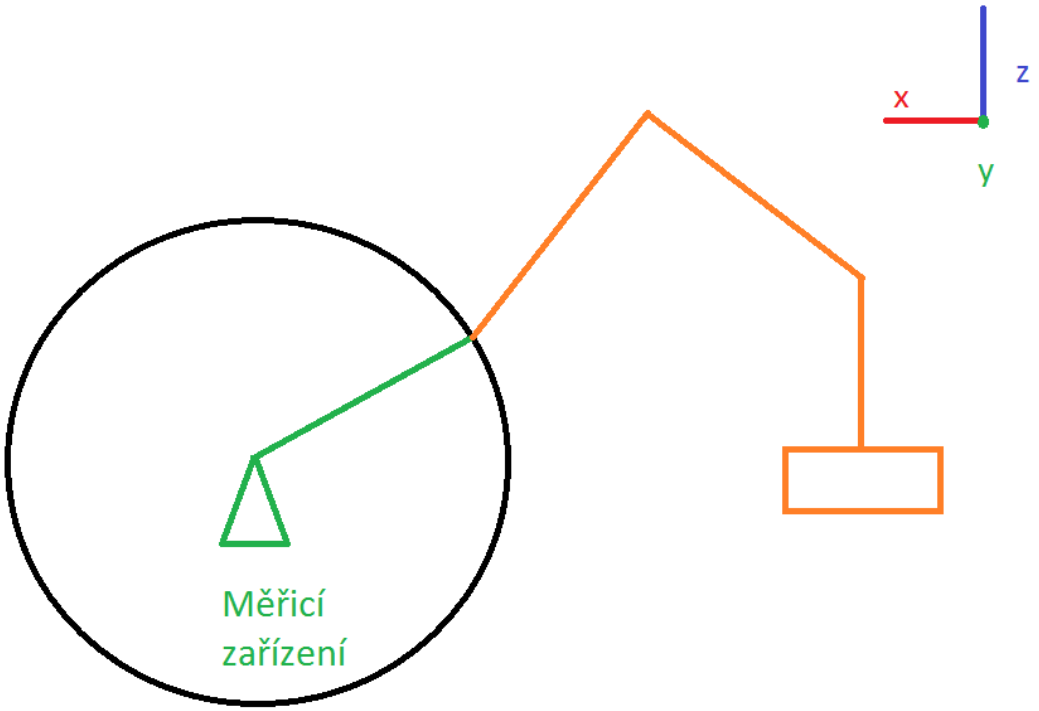
Obrázek 3: Otáčení první osy

Jakmile zjistíme, že se vzdálenost zvětšuje (minuli jsme minimum), vrátíme se zpět do minimální hodnoty. Teď víme, že právě zde v minimu se první osa nachází ve své nulové hodnotě. Pozici v minimu ukazuje obrázek 4.



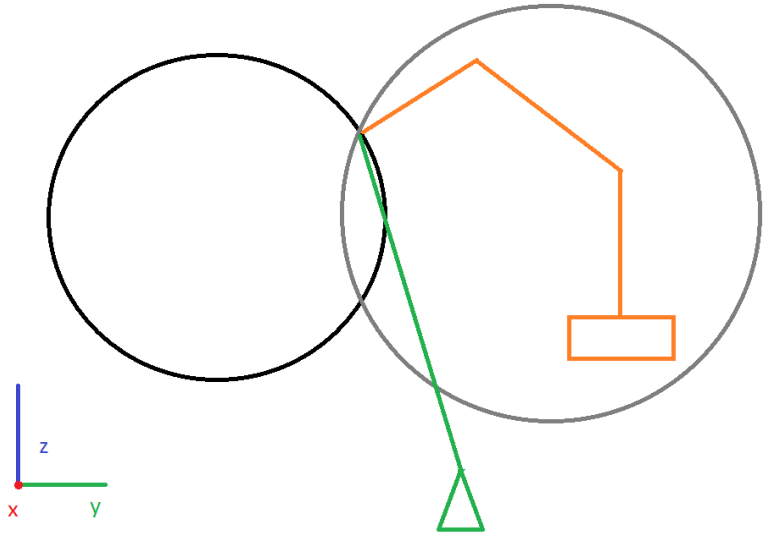
Obrázek 4: Pozice první osy v minimu

V tuto chvíli by se nám hodilo zjistit polohu TCP. Jediné, co o ní víme, je, že se nachází v průsečíku koule se středem v měřicím zařízení a roviny *xz*, což je znázorněno na obrázku 5.



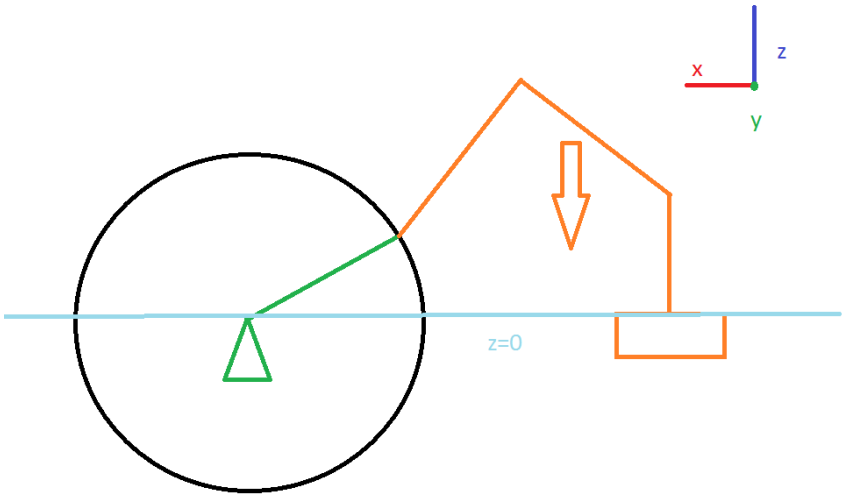
Obrázek 5: Pohled z boku

Tento průsečík je kružnice, která nám přesnou polohu TCP ještě sdělit nedokáže. Proto otočíme první osou robota o čtvrt otočky (π/2). Nyní máme opět rovinu (*yz*), opět původní kouli, kterou jsme otočili spolu s TCP okolo osy *z*, a novou kouli, která má střed v měřicím zařízení. Průnikem každé jednotlivé koule s rovinou vzniká kružnice. Z toho vyplývá, že zde máme průsečík dvou kružnic ležíc v jedné rovině. Máme tedy dva body, z nichž jeden udává polohu TCP. Vše nám ukazuje obrázek 6.



Obrázek 6: Průnik dvou kružnic

První má hodnotu v ose *z* kladnou, druhý zápornou. Otočíme zpět první osou o π/2 a poté otočíme o předem danou hodnotu druhou osou směrem k základně robota, což je znázorněno na obrázku 7.

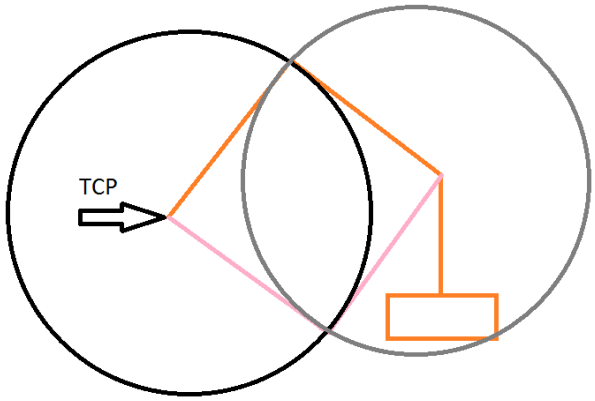


Obrázek 7: Pohyb ke zjištění správného bodu

Pokud se měřená vzdálenost zmenší, výsledný bod je ten s kladnou hodnotou na ose z. Pokud se vzdálenost zvětší, výsledný bod je ten se zápornou hodnotou na ose z. Následně otočíme druhou osou zpět o výše zmíněnou předem danou hodnotu.

Jakou velikost má tato hodnota by bylo řešeno až na místě na reálném robotovi s ohledem na vhodnou velikost pohybu. Co se týče směru k základně robota, záleželo by na směru otáčení druhé osy, což by se vyřešilo opět až na místě s reálným robotem. U bodu, který máme spočtený, prohodíme hodnotu u x a y, protože výpočet probíhal v pozici otočené o π/2 vůči aktuální. V tento moment známe polohu TCP a zbývá nám určit hodnota natočení druhé a třetí osy.

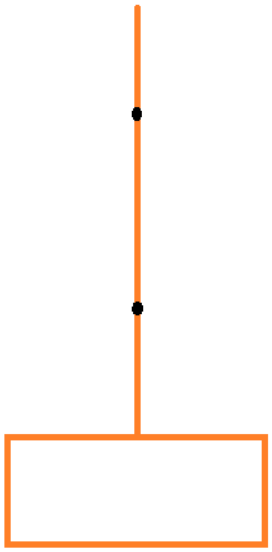
Vytvoříme kružnice se středy ve druhé ose a v TCP o poloměrech rovnajících se délkám druhého a třetího ramene. V průsečíku těchto kružnic se nachází třetí osa. K určení toho, který ze dvou bodů je náš hledaný, nám slouží jednoduchá, okem postřehnutelná informace o pozici robota. Zda odpovídá oranžové či růžové na obrázku 8, dokáže určit i osoba robotů zcela neznalá.



Obrázek 8: Dopočtení posledního bodu

Teď již známe všechny polohy jednotlivých bodů, které nám představují pozice kloubů robota v prostoru. Tím je myšlena pozice vůči základně, nikoliv hodnoty natočení. Z těchto tří bodů již můžeme pomocí trigonometrických vlastností zjistit natočení druhé a třetí osy.

Během kalibrace bychom se mohli potýkat s některými speciálními případy. Pokud se při otočení první osou vzdálenost TCP nemění, víme, že robot je rovně směrem vzhůru, což nám ukazuje obrázek 9.

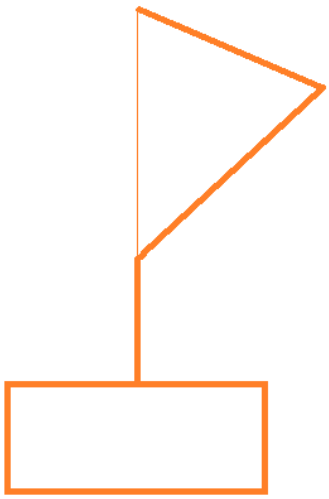


Obrázek 9: Speciální pozice robota

Tudíž můžeme usoudit, že druhá a třetí osa jsou v nulové poloze. Polohu první osy určíme tak, že otočíme druhou osou o libovolnou hodnotu (např. π/6) a poté budeme otáčet první osou tak, aby se měřená vzdálenost TCP zmenšovala, což znamená, že se TCP přibližuje k měřicímu zařízení. Také to znamená, že se TCP pohybuje po kružnici se středem v ose otáčení prvního kloubu. Jakmile zjistíme, že vzdálenost začíná stoupat, vrátíme se zpět do minima, kde se nachází hledaná nulová pozice

první osy. V tuto chvíli již známe všechny tři hledané hodnoty kloubových souřadnic. To, že je robot v přímce směrem vzhůru, nám pomůže zjistit též vzdálenost TCP od senzoru, kterou můžeme dopředu spočítat.

Může se nám též stát, že TCP bude v ose prvního kloubu a přesto nebude robot v přímce, což můžeme spatřit na obrázku 10.



Obrázek 10: Zvláštní poloha robota

V tomto případě bude vzdálenost TCP menší než v předchozím případě. I v této situaci se ale nebude měnit se změnou pohybu první osy. Pohneme tedy jinou osou o libovolnou hodnotu a poté postupujeme podle původního algoritmu.

Tímto jsou popsány všechny možné startovní konfigurace pro kalibraci robota s jedním měřícím zařízením. Algoritmus neřeší přesnost senzoru. Při malé přesnosti se zde může vyskytnout problém vyhodnocování polohy

první osy. V okolí minima jsou rozdíly mezi naměřenými vzdálenostmi velmi malé, tudíž s jeho určením můžeme mít problém. Řešením může být použití dalšího senzoru, o čemž však pojednává kapitola 2.3.

2.3 Kalibrace se dvěma měřicími zařízeními

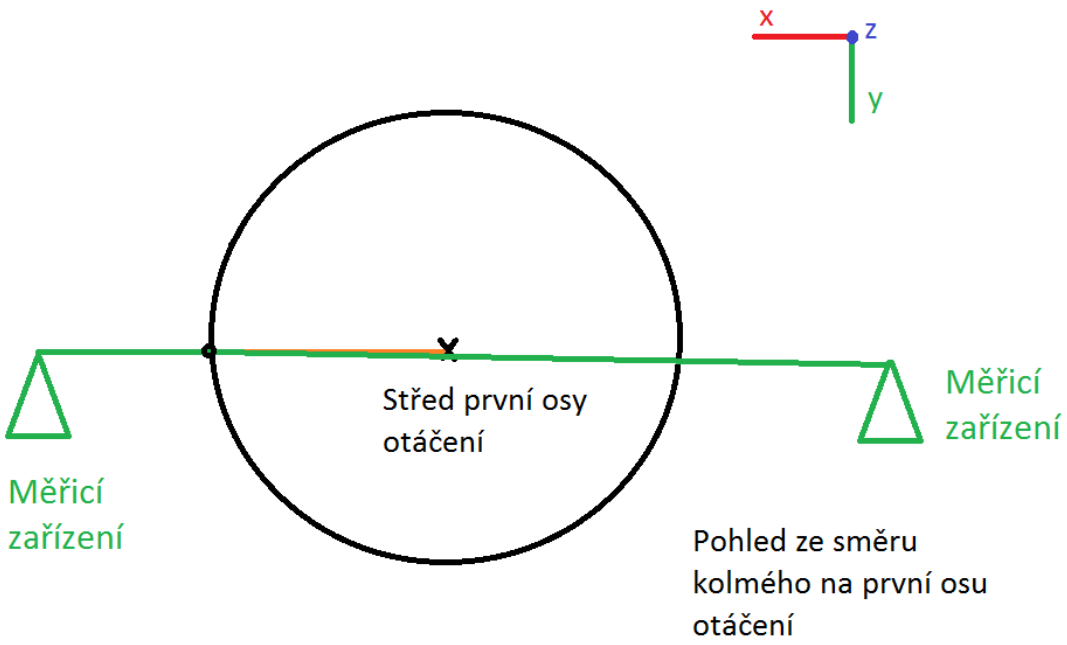
Největší výhodou kalibrace se dvěma měřicími zařízeními je přesnější informace o poloze koncového bodu. Jestliže prvotní informací u metody s jedním měřicím zařízením byla koule, na které se TCP nachází, při použití dvou zařízení je prvotní informace průnik dvou koulí (kružnice). Ještě před samotným započtením algoritmu a vymýšlením optimálního pohybu pro určení přesných kloubových souřadnic tedy víme mnohem více, než při metodě s jediným zařízením.

V této části se nelze vyhnout debatě, jak by to vypadalo v případě více měřicích zařízení? Samozřejmě zvyšující počet měřicích zařízení nám ulehčuje práci ve smyslu rychlosti nalezení kloubových souřadnic, ale naopak ztěžuje hledání optimálního algoritmu, jelikož počet možností, co lze dělat se všemi zařízeními, výrazně narůstá. Jelikož ale máme měřicí zařízení dvě, je tato otázka veskrze teoretická. Pokud bychom měli k dispozici tři měřicí zařízení, nepotřebovali bychom s robotem vůbec hýbat, abychom zjistili souřadnice koncového bodu. Pouze bychom si museli dát pozor, abychom se vyhnuli špatným umístěním měřicích zařízení. Tím je myšleno, aby zařízení nebyla na jedné přímce, případně aby robot nezasahoval svým tělem do lanek z čidel vycházejících. V praxi bychom byli omezeni dosahem senzoru a místy, kam ho můžeme umístit.

2.4 Vzájemná poloha měřicích zařízení

Při použití dvou měřicích zařízení vyvstává otázka jejich vzájemné polohy. Jde nejen o to najít takovou polohu, která algoritmus co nejvíce urychlí, ale také i o použitelnost takového algoritmu v praxi, čímž je myšleno, aby druhé měřicí zařízení robotu nepřekáželo a aby se robot svým pohybem nedotýkal ostatními částmi nataženého lanka měřícího vzdálenost TCP.

První zkoušenou polohou jsou měřicí zařízení umístěná naproti sobě v jedné ze souřadnicových os (např. x), což nám ukazuje obrázek 11.

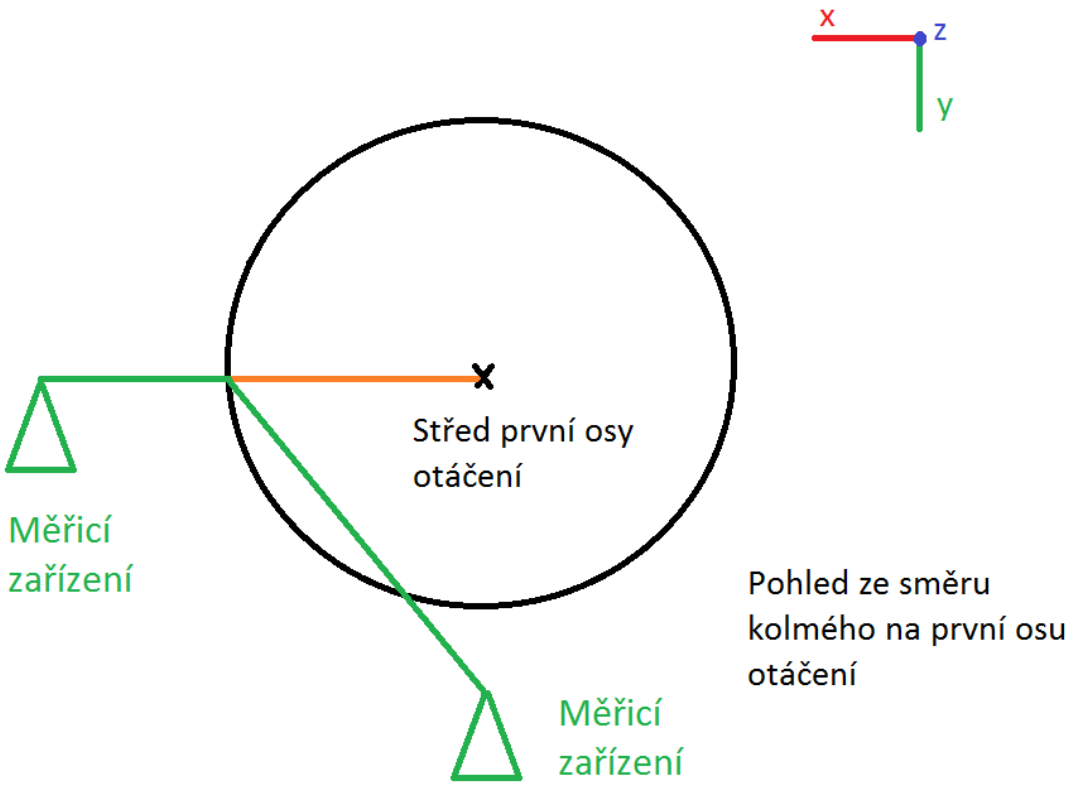


Obrázek 11: Dvě měřicí zařízení v ose x

Je vidět, že tato varianta příliš praktická není, protože robot svým tělem při měření vždy jednomu ze senzorů překáží. Dalším důvodem vyřazení

této možnosti je, že obě zařízení měří stejné hodnoty pouze v opačném směru osy. Takovéto rozložení je tedy špatné.

Další zkoušenou polohou jsou měřicí zařízení umístěná o čtvrt otáčky od sebe, což je vidět na obrázku 12.



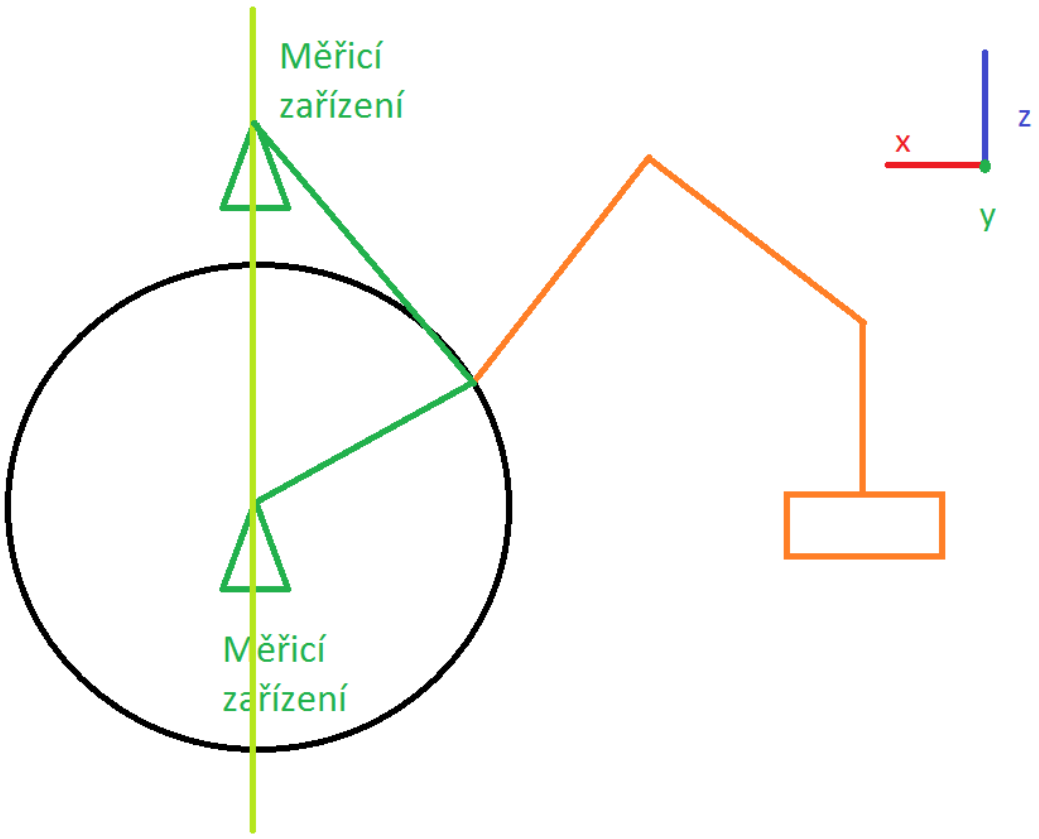
Obrázek 12: Měřicí zařízení umístěná čtvrt otáčky od sebe

Tato pozice je již použitelná v praxi, protože se dokážeme vyhnout místům, kde by robot omezoval svým tělem přesnost měření. Při hledání kloubových souřadnic dojde ke zjednodušení algoritmu avšak nikterak výraznému. Nejvíce nám zařízení pomůže, pokud nebude připojeno na TCP, nýbrž bude měřit vzdálenost třetího kloubu. Měření této vzdálenosti nemusí být nijak přesné, a pokud by se náhodou během svého pohybu robot lanka dotkl, našemu algoritmu to neuškodí. Senzor by sloužil pouze k rozeznání, která ze dvou možností (viz nevyřešený

problém v případě použití jednoho měřicího zařízení) je aktuální reálná situace.

Pokud si uvědomíme hlavní použití druhého zařízení, vyjde nám, že jeho polohu můžeme optimalizovat na místě na základě co nejjednodušší obsluhy a instalace.

Nejvýhodnější pozicí se nakonec ukázala dvojice měřicích zařízení umístěná nad sebou v ose z, což nám ukazuje obrázek 13. Alternativně lze také použít dvě zařízení umístěná vedle sebe v ose y. Což si lze také představit, jako bychom otočili souřadný systém o π/2 okolo osy x.

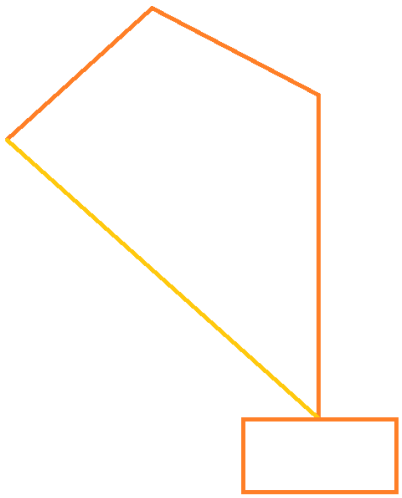


Obrázek 13: Optimální umístění senzorů

Při určování první kloubové souřadnice postupujeme podle původního algoritmu stejně, jako by druhé měřicí zařízení vůbec neexistovalo. Pro určení polohy TCP již nemusíme složitě otáčet robotem, ale můžeme ho spočítat přímo v aktuální pozici. Z předchozího algoritmu víme, že TCP leží v průniku koule se středem v původním měřicím zařízení a roviny *xz*. Druhé měřicí zařízení (také spojené s TCP) nám říká, že také musí ležet na kouli se středem v novém měřicím zařízení. Z toho vyplývá, že průnik všech těchto geometrických útvarů jsou dva body. Ačkoliv je redukce značná, rádi bychom dosáhli ideálně bodu jednoho. To lze dosáhnout vhodným umístěním obou měřicích zařízení relativně dále od robota, abychom zamezili robotovi dosáhnout bodu, který leží za přímkou spojující obě měřicí zařízení. Pokud za tuto osu robot nedosáhne, můžeme vzdálenější bod vyřadit a známe souřadnice TCP. Poté pokračujeme podle původního algoritmu s vědomím, že jsme jej asi o polovinu zkrátili.

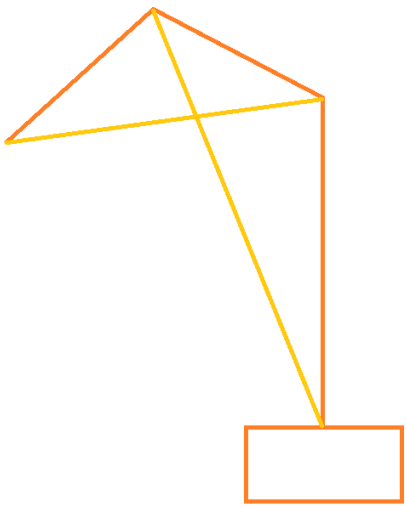
2.5 Měřicí zařízení umístěná přímo na robotovi

Absolutně nejlepší metodou, jak zjistit poslední dvě kloubové souřadnice, je umístění měřicích zařízení přímo na robota. Tímto odpadá problém, aby robot nezavadil svým tělem o měřicí lanko. Dále pak není nutné s robotem vůbec pohybovat, jelikož všechny hodnoty lze spočítat pouze z aktuální polohy. Tuto metodu lze však použít pouze se dvěma měřicími zařízeními a neumí vypočítat polohu první osy. Pokud bychom měli pouze jedno měřicí zařízení, bylo by naším úkolem najít čtyřúhelník se zadanými stranami, což je názorně ukázáno na obrázku 14.



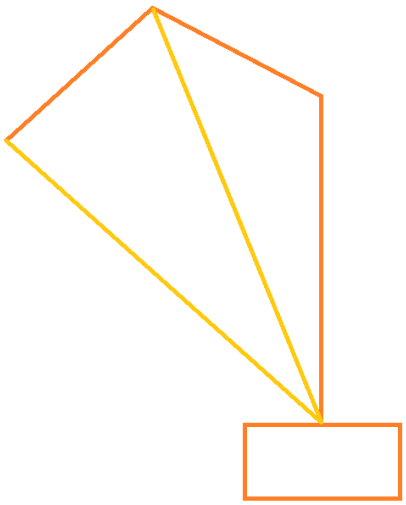
Obrázek 14: Jedno měřicí zařízení na robotovi

Taková úloha má nekonečně mnoho řešení. Pokud tedy vyjdeme z předpokladu, že máme dvě měřicí zařízení, umístíme první na počátek robota (na první kloub) a jeho konec na třetí kloub. Druhé zařízení připevníme k druhému kloubu a jeho konec na TCP, což můžeme vidět na obrázku 15. Tímto dostaneme dva trojúhelníky, u kterých známe jejich strany, čímž jsou i úhly jasně dané a úloha je vyřešena.



Obrázek 15: Rozdělení robota na dva trojúhelníky

Alternativně můžeme umístit druhé měřicí zařízení také na první kloub a jeho konec na TCP, což nám názorně ukazuje obrázek 16. Tato možnost vyjde matematicky také správně, ale je náročnější na výpočet.



Obrázek 16: Jiná možnost rozdělení robota

Zde vyvstává otázka rozměrů. Pokud je robot výrazně větší než měřicí zařízení, můžeme tuto metodu pohodlně aplikovat. Pokud se však velikost robota relativně k velikosti měřicího zařízení zmenšuje, vyvstává problém, jak zařízení vhodně umístit. To lze však řešit až na místě s reálným robotem i měřicím zařízením.

3) Model v *simmechanics*

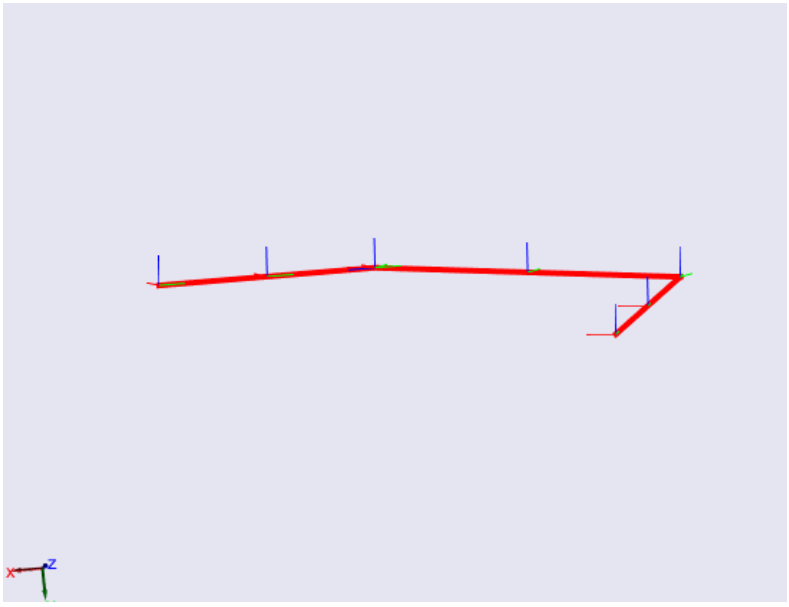
3.1 Prostředí *simmechanics*

Pro simulaci pohybu robota při průchodu algoritmů použitých v bodech 2.2 a 2.3 jsme použili prostředí *simmechanics*, které je součástí softwaru *matlab.* Pro tuto simulaci jsme vycházeli z algoritmu navrženého

pro obecného robota. Algoritmus použitý na našeho reálného robota jsme si na rozdíl od obecného mohli vyzkoušet v praxi. Simulace v prostředí *simmechanics* se dá uložit také jako video, což však pro prezentaci algoritmu zde ve statické podobě postrádá využití. Můžeme však virtuálním robotem hýbat všemi směry, což nám usnadňuje vytváření náhledů. Nejprve jsme si museli poradit s rozdílným ovládáním oproti různým typům softwaru určeným k průmyslovému programování robotů (*Robotstudio, Famos*), abychom se naučili virtuálního robota ovládat. Poté jsme již mohli přistoupit k vykreslení všech požadovaných poloh. Výhodou prostředí *simmechanics* je snadná úprava jednotlivých fyzikálních vlastností robota (hmotnost, délka,…), které však v této části pro vytváření obrázků nebudeme potřebovat. Kromě vizualizace lze v prostředí *simmechanics* vytvářet simulační modely a tím se nám otevírá možnost ověřovat algoritmus kalibrace.

3.2 Postup metody

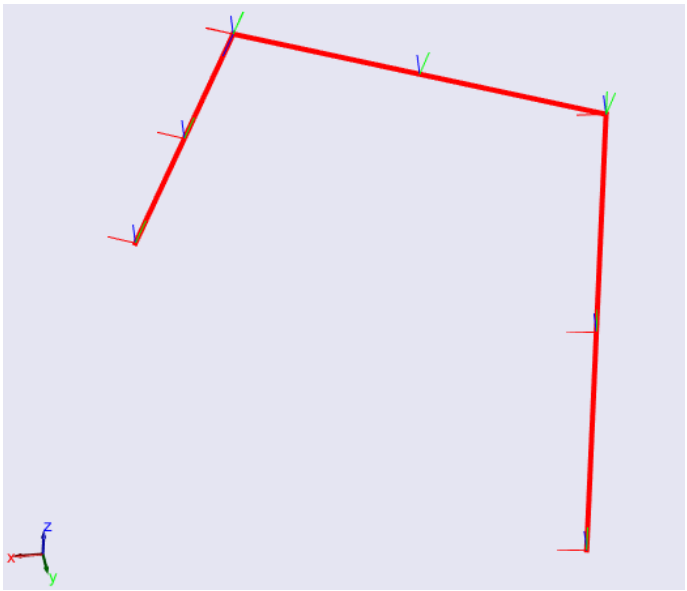
Postupovali jsme ve stejném pořadí jako v bodech 2.2 a 2.3. Začali jsme tedy pohledem shora, což nám ukazuje obrázek 17.



Obrázek 17: Robot v prostředí simmechanics

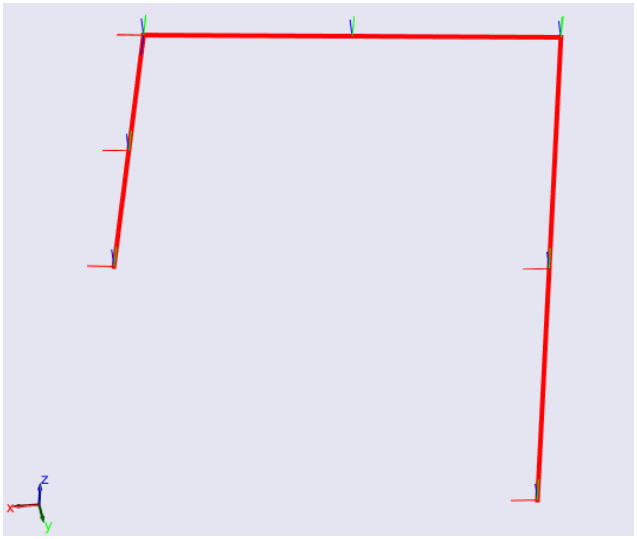
Můžeme si všimnout, že každý kloub a každé rameno má zde vykresleny směry os *x,y,z.* Tuto funkci lze v *simmechanics* také samozřejmě vypnout. Další věcí, kterou lze na obrázku zobrazit, jsou jednotlivá těžiště. Tuto možnost jsme však do náhledu nezahrnuli, jelikož snižovala jeho přehlednost.

Na obrázku 18 je robot při pohledu z boku, stejně jako v bodě 2.2.



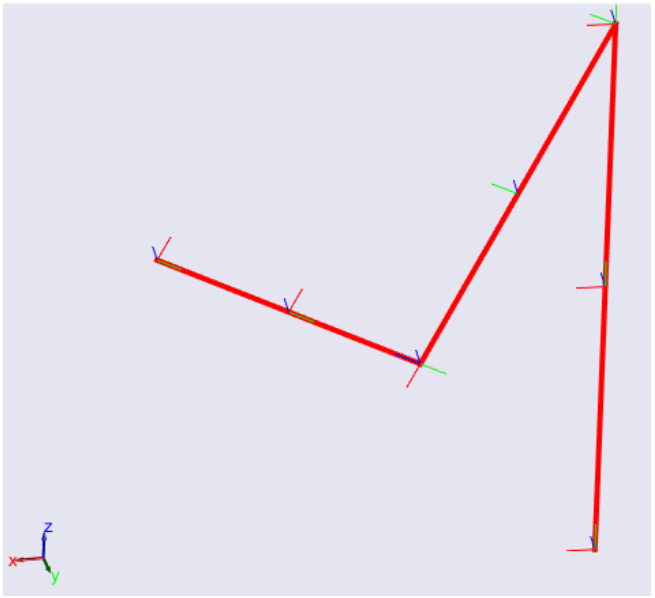
Obrázek 18: Robot při pohledu z boku

Posunutí druhé osy směrem k základně ve výše zmíněném algoritmu má být o předem danou hodnotu. Pro grafickou představu je výhodnější větší hodnota, aby byl vidět mezi jednotlivými náhledy rozdíl, což dokazuje obrázek 19.



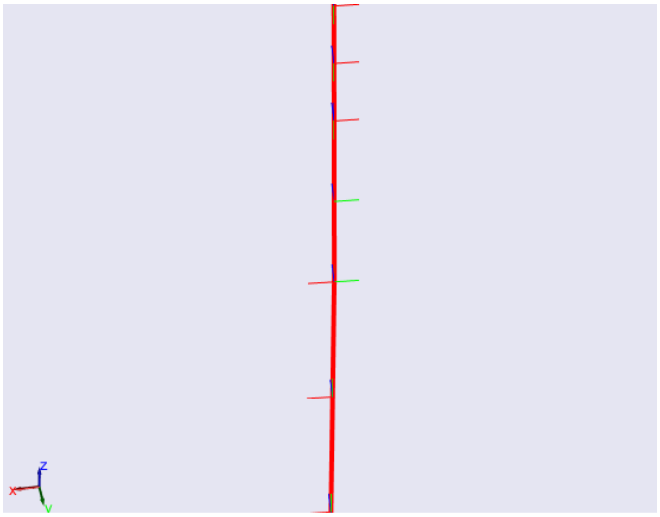
Obrázek 19: Robot po otočení druhé osy

Problém dvou možností, jak dosáhnout cílového bodu je vidět, pokud porovnáme předchozí obrázek 19 s následujícím obrázkem 20. I přestože známe cílový bod a natočení první osy. Stále nevíme, jak přesně bude konfigurace robota vypadat.



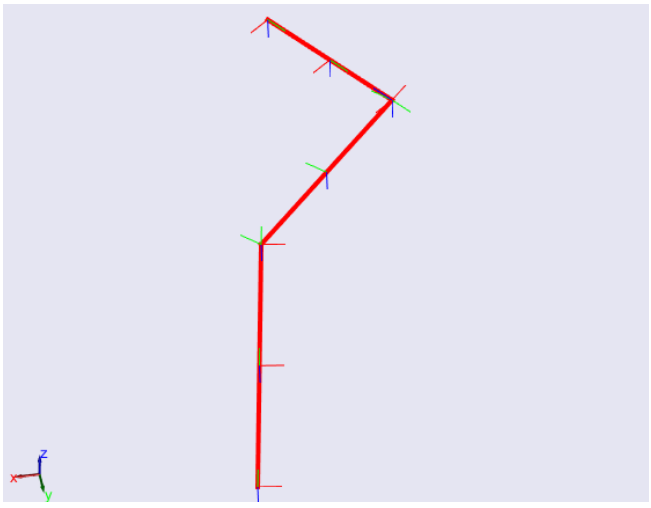
Obrázek 20: Stejné souřadnice koncového bodu

Ze zajímavých poloh, které zde prezentujeme obrazově, jsme ještě nevykreslili tu, kde jsou všechna tři ramena v jedné přímce, což napravujeme obrázkem 21.



Obrázek 21: Všechna ramena robota v přímce

Poslední polohou, kterou jsme získali v prostředí *simmechanics*, je ta, kde koncový bod leží na přímce dané prvním ramenem, což nám ilustruje obrázek 22.



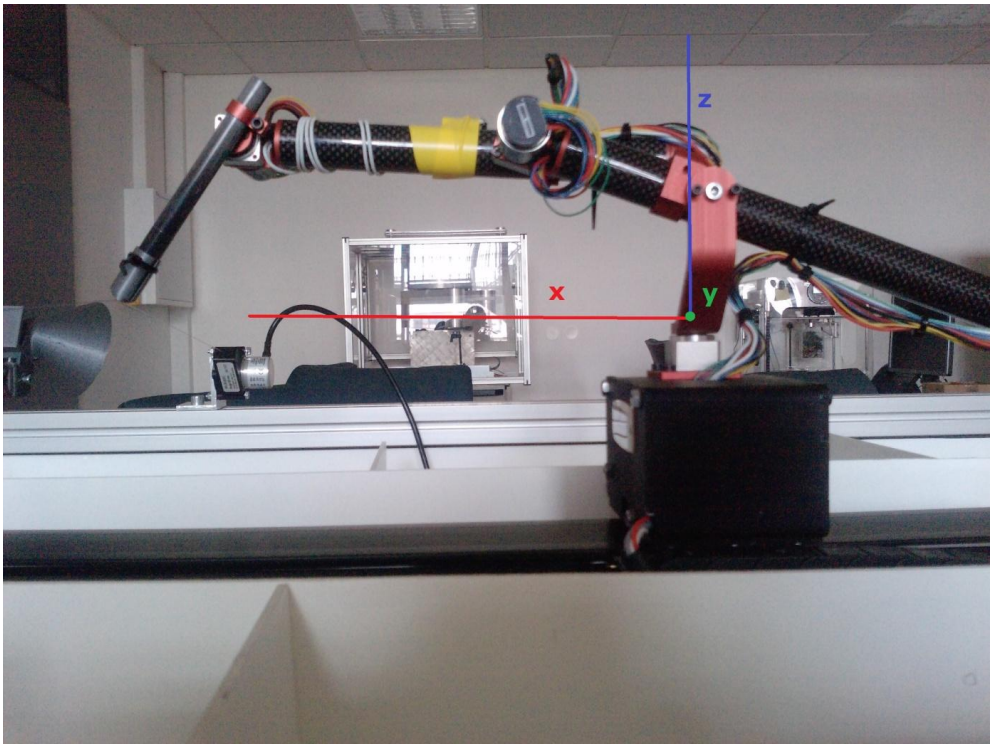
Obrázek 22: Koncový bod v přímce dané prvním ramenem

Tímto jsme nasimulovali polohy teoreticky zpracované v předchozích bodech. Mohli jsme si je zobrazit ze všech různých pohledů, což zlepšilo naši představu o jednotlivých algoritmech. Pro případ praktického využití je také možnost nahrát vše jako video velkou výhodou.

4) Aplikace na konkrétním robotovi

4.1 Reálný robot

Náš reálný robot se nachází v učebně UN406 budovy NTIS Západočeské univerzity v Plzni, kde původně sloužil jinému účelu (házení míčkem), ale pro naši práci vyhovuje perfektně. Pracoviště s robotem včetně souřadného systému je vidět na obrázku 23.



Obrázek 23: Robot se souřadným systémem

Robot je ve skutečnosti typu PRRR, ale abychom se drželi našeho zadání, budeme považovat hodnotu posuvného kloubu za známou. Také případné zjištění její hodnoty lze provést například metrem poté, co si dopředu zjistíme, v jaké poloze má nulovou polohu. Robot má překvapivě velké možnosti pohybu. Omezen je pouze zapojením pohonů jednotlivých kloubů. Rozměry robota jsou přehledně zobrazeny v tabulce 1.

Rameno	Délka [mm]
1.	106
2.	126
3.	90

Tabulka 1: Rozměry robota

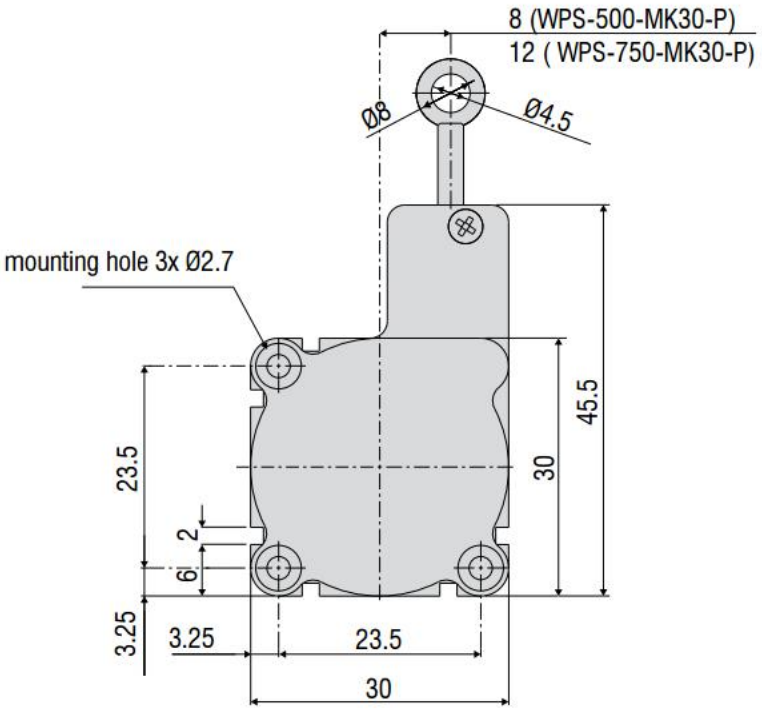
4.2 Senzor měření vzdálenosti

Pro měření vzdálenosti jsme měli k dispozici lankový senzor *MicroEpsilon WPS-500-MK30*, jehož vlastnosti jsou přiloženy v tabulce 2.

Vlastnost	Vzdálenost [mm]
rozlišení	0,1
rozsah	500
linearita	<0,25

Tabulka 2: Vlastnosti senzoru [4]

Senzor má kompaktní rozměry, které jsou vidět na obrázku 24.



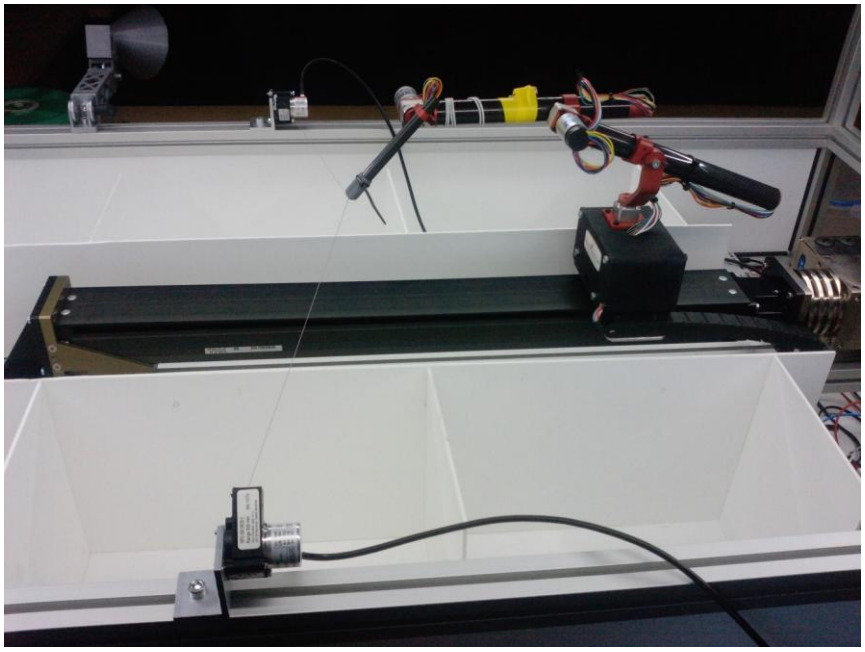
Obrázek 24: Rozměry senzoru [4]

Rozpětí pracovní teploty je od -20 do 80 °C [4], tudíž při naší pokojové teplotě nenastává v tomto směru žádný problém. Výstupem senzoru je číslo udávané v pulzech, které je nutno, pro obdržení vzdálenosti v milimetrech, vydělit koeficientem 10 [4]. Ze senzoru vychází ocelové lanko, které má dostatečnou pevnost, aby se nám nepřetrhlo. Vzhled senzoru lze hodnotit na obrázku 25.

Obrázek 25: Senzor měření vzdálenosti [5]

4.3 Umístění robota

Měření probíhalo v boxu, ve kterém se robot pohyboval a na jehož okrajích byly senzory umístěny. Vše znázorňuje obrázek 26.



Obrázek 26: Prostředí pro měření

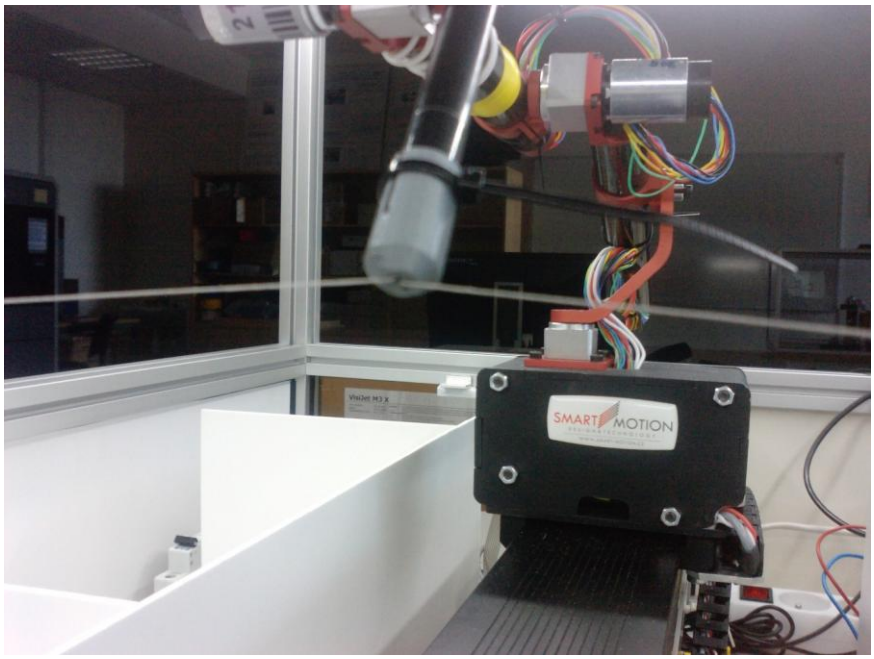
Zde je vidět pozice posuvného počátku robota, která byla pro tuto úlohu zvolena jako nulová a jejíž vzdálenost od okraje je pevně dána. Pro všechny výpočty v naší úloze budeme počítat v souřadném systému základny. Z toho vyplývá, že počátek leží v první ose otáčení a ve výšce, kde končí část robota přimontovaná na posuvnou osu. K tomuto bodu vztáhneme umístění senzorů, jejichž souřadnice jsou vidět v následující tabulce 3. Kladná osa *x* vede z tohoto bodu ve směru posuvné osy k senzorům. Kladná osa *z* leží v ose otáčení prvního kloubu a kladný směr je vzhůru. Osa *y* je pak dána pravorukým souřadným systémem.

souřadnice	Senzor 1	Senzor 2
x [mm]	273	273
y [mm]	331	-291
z [mm]	-12	-12

Tabulka 3: Poloha senzorů

Toto umístění senzorů bylo zvoleno záměrně, aby byly výpočty maximálně zjednodušeny. Nejprve bylo zkoušené rozmístění s druhým senzorem umístěným na horním rámu celé klece, ale vyskytl se problém s pohybem robota v blízkosti singularity, která se nachází na přímce mezi dvěma senzory. Tomuto se zabránilo díky umístění obou senzorů do záporných hodnot osy *z*, kde robotovi v dosažení singularity brání fyzická omezení.

Připojení senzorů na konec robota je vidět na obrázku 27.

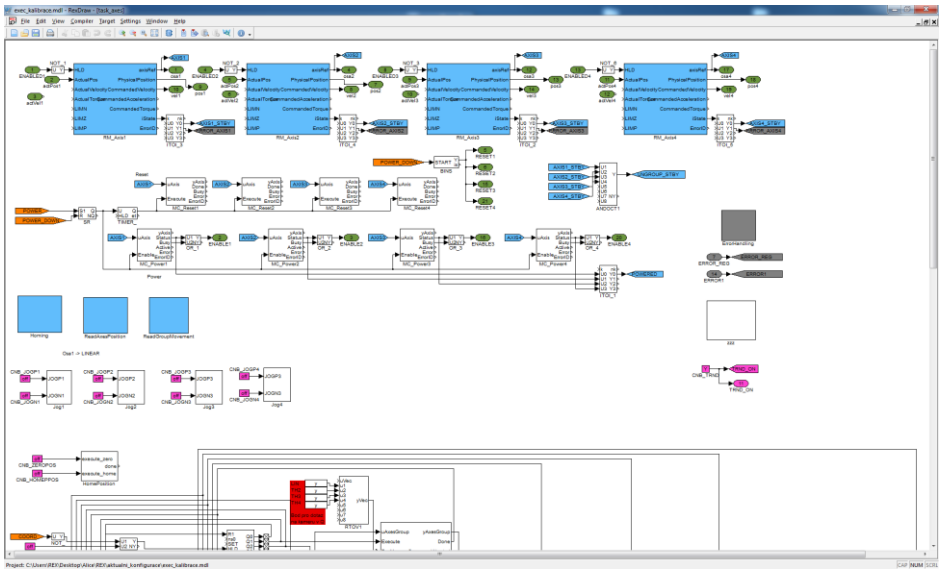


Obrázek 27: Upevnění senzorů k robotovi

Kvůli tomuto fyzickému omezení se zmenšil pracovní prostor robota, neboť z důvodu komplikovaněji odnímatelné příruby je robot omezen dosahem obou senzorů.

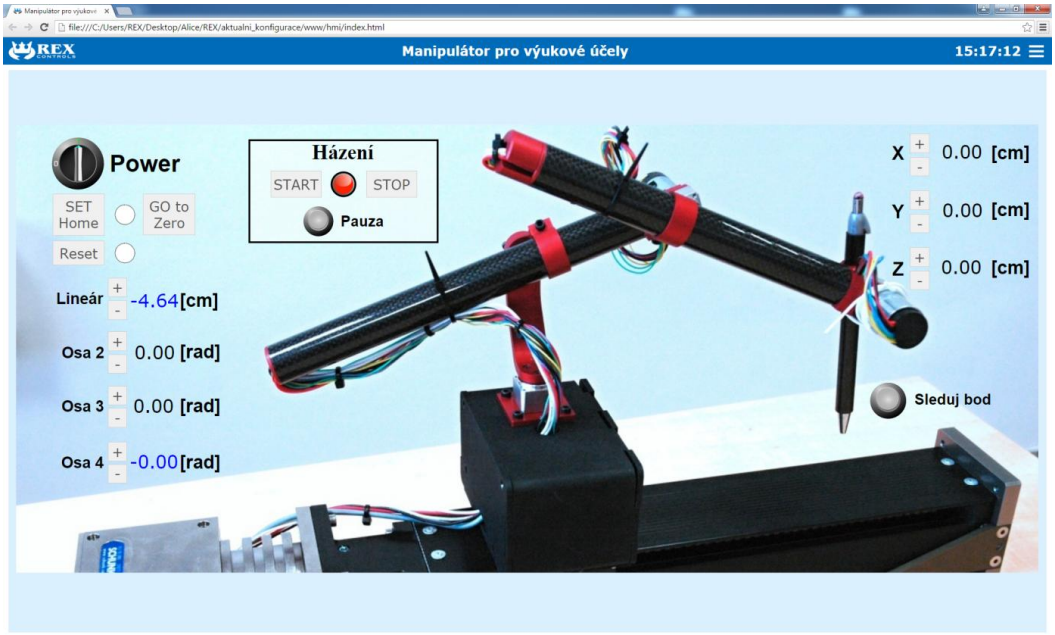
4.4 Ovládací prostředí

Realizace zapojení a ovládání byla vypracována v řídicím systému reálného času *REX*, který je vidět na obrázku 28.



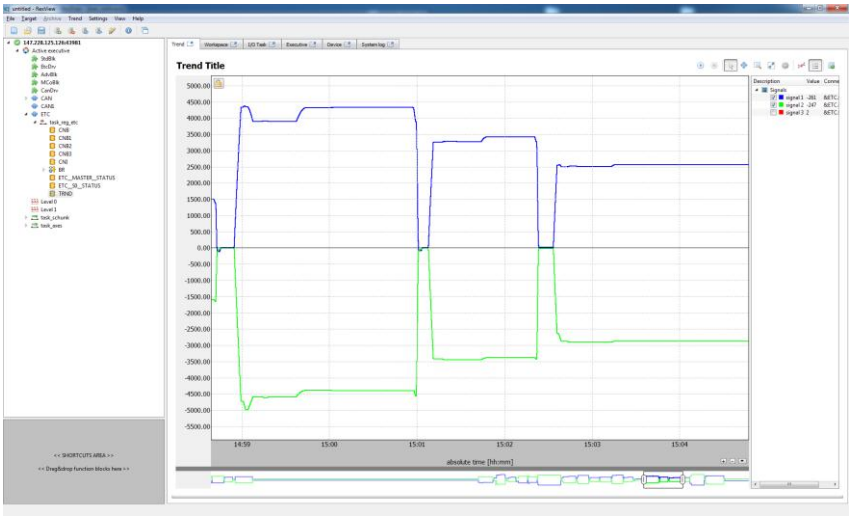
Obrázek 28: Řídicí systém reálného času REX

Celé zapojení jsem převzal z projektu házení míčkem, které bylo již dříve realizováno na stejném robotovi. Samotné ovládání probíhá přes aplikaci spouštěnou v prohlížeči, kterou lze sledovat na obrázku 29.



Obrázek 29: Ovládání robota

Zde lze jednoduše pohybovat jednotlivými osami a případně také robota poslat do jeho domovské pozice. Data ze senzoru byla zpracována v prostředí *RexView*, které nám ukazuje obrázek 30.



Obrázek 30: Zobrazení dat v prostředí Rex

Odtud byla data exportována pomocí souboru *.csv* do *Matlabu*, ve kterém probíhal výpočet celého algoritmu. Nevýhodou byla absence *Matlabu* v učebně UN406, což znamenalo využívání učebny UC432, kde je tento program dostupný.

Problémem, na který jsme narazili až v této fázi práce, bylo měření senzorů pouze v relativním slova smyslu. Znamená to, že nevíme, v jaké vzdálenosti od jednotlivých senzorů se koncový bod robota v daný okamžik nachází, ale pouze v jakém vztahu je tato vzdálenost k vzdálenosti původní, jež je brána jako nulová. Jedním z našich problémů je tedy relativní vzdálenost přeměnit na absolutní, abychom mohli určovat polohu koncového bodu robota.

Algoritmus jsme museli upravit na reálný projekt. Byli jsme například omezeni možnostmi umístění senzorů a také místo pro uchycení měřicího lanka na robotu bylo dopředu dáno. Z těchto hledisek nám vyšel následující algoritmus.

5) Kalibrace reálného robota

5.1 Natočení první osy

Našim prvním cílem je určit souřadnice koncového bodu robota. S ohledem na zadání jsme se snažili, aby člověk nemusel pohybovat s robotem v příliš mnoha úkonech. Tím je myšleno, aby robot prováděl co nejméně pohybu.

Určení souřadnice *x* je díky vhodnému umístění senzorů velmi prosté. Pohybujeme robotem pomocí posuvné osy, dokud nám koncový bod neprotne rovinu danou přímkou mezi oběma senzory a osou *z.* Po jejím protnutí pokračujeme ještě centimetr, abychom měli jistotu, že jsme již za touto rovinou. Poté provedeme opačný pohyb zpět a to pomocí pokynu *Go home*, který je v nabídce. Výhodou této zvolené metody je její jednoduchost. Pro další výpočty již není zapotřebí žádných dalších pohybů robota. Jsme schopni pouze z tohoto jednoduchého pohybu vyčíst všechny potřebné informace k určení kalibrovaných natočení rotačních kloubů. Tím je míněn fakt, že není nutné s robotem již více pohybovat.

Vraťme se však k určení souřadnice *x*. Jelikož jsme celým robotem pohybovali pouze v ose *x*, ostatní souřadnice během celého pohybu zůstaly neměnné. Díky vhodnému umístění senzorů víme, že nejblíže k oběma senzorům byl koncový bod v již výše zmíněné rovině dané senzory a osou *z.* Ze senzorových dat musíme tedy zjistit, v jakém čase naměřily oba senzory minimum.

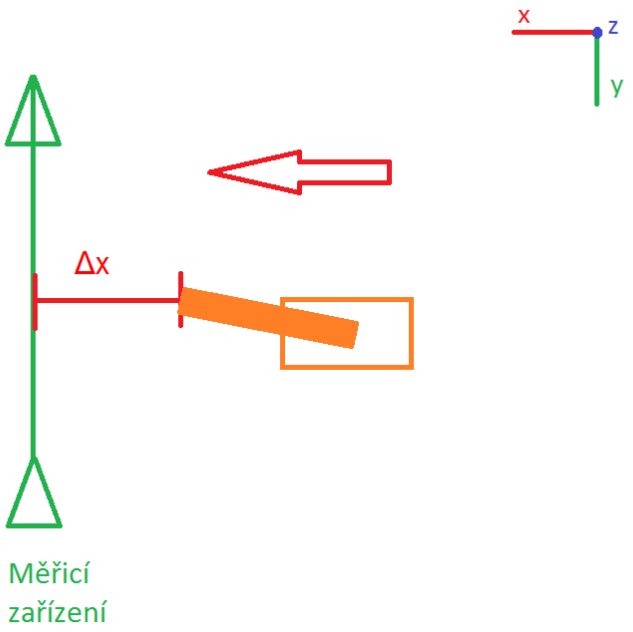
Zde jsme se setkali s tím, že časové údaje pro senzory a robota se zdály mírně posunuté. Usoudili jsme, že chyba bude v nestejném měření času pro jednotlivé senzory a pro osu. To jsme si ověřili tím, že jsme se podívali na počátek měření, kde bylo vidět, jak se osa robota začala

pohybovat a až po nějakém čase se začaly měnit údaje na obou senzorech.

S tímto problémem jsme se vypořádali pomocí posunutí naměřených dat do stejných časových okamžiků, což sice nevyřešilo chybu absolutně, ale pro jednotlivá měření nám to umožnilo ji zanedbat. Pro každé měření je však chyba jiná, proto se tento postup nedá aplikovat obecně a v případě nových dat by bylo nutné opět přijít na to, jak data posunout, aby absolutní časy byly všude stejné. Nakonec bylo zjištěno, že příčina tkví v záměrně zavedeném zpoždění v konfiguraci řídicího systému za účelem synchronizace dvou rozdílných typů pohonů (lineárního a rotačního). Její absolutní řešení je však mimo téma této práce a vyžádalo by si delší čas na analýzu.

Po synchronizaci dat, vycházivších ze senzoru, jsme konečně mohli přistoupit k nalezení minima. Senzor se v minimu nachází ve více časových okamžicích, jelikož jeho perioda vzorkování je výrazně menší než u robota. Tím je myšlen fakt, že na určitý časový interval připadá více vzorků ze senzoru než na stejný časový interval v případě robota. Z tohoto důvodu jsme pro každý senzor zvlášť zvolili čas a z obou poté udělali aritmetický průměr.

V tuto chvíli víme, kdy robot proťal naši výše zmíněnou rovinu. Podíváme se do dat z robota a zjistíme, jakou vzdálenost v onen inkriminovaný čas urazil a tím zjistíme rozdíl v ose *x* mezi bodem, který leží v této rovině a původním bodem, kde se konec robota nacházel na počátku, což ukazuje obrázek 31.

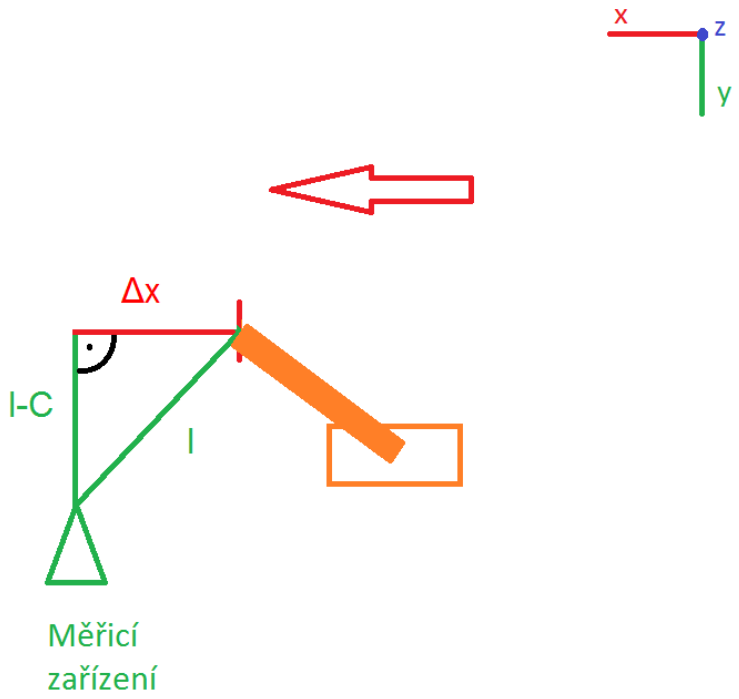


Obrázek 31: Souřadnice x koncového bodu

Jelikož je hodnota souřadnice *x* v naší rovině konstantní (273), stačí odečíst vzdálenost uraženou robotem Δx a máme první souřadnici koncového bodu.

$$x = 273 - \Delta x$$

Dalším problémem, který je nutné vyřešit je relativní měření senzorů. Díky znalosti rozdílu vzdáleností mezi počátečním bodem a bodem v naší rovině a díky již zjištěné vzdálenosti v ose *x* lze převést relativní měření senzoru na absolutní, což ukazuje obrázek 32.



Obrázek 32: Přepočet na absolutní měření

Trojúhelník je umístěný v prostoru a má pravý úhel. Díky Pythagorově větě víme, že:

$$l^2 = (l - C)^2 + \Delta x^2$$

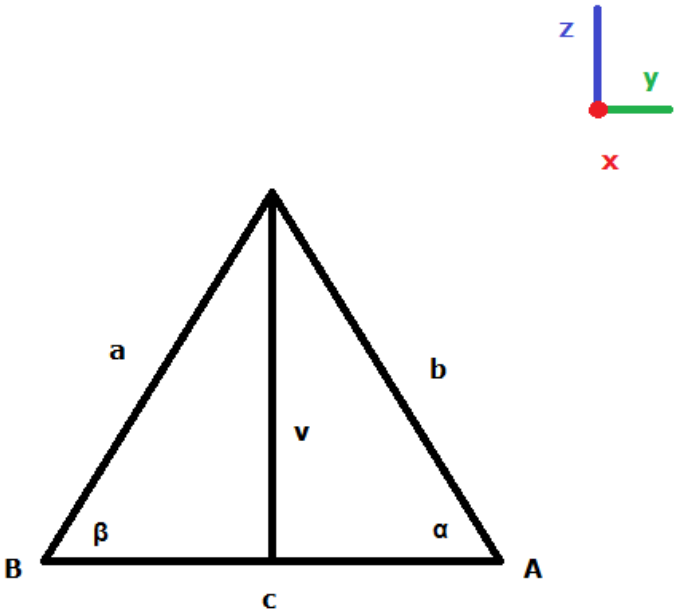
Jelikož známe C (změna vzdálenosti oproti *l* – relativní měření) a Δ*x*, můžeme *l* spočítat a tím dostáváme počáteční hodnotu senzoru. To samé provedeme také pro druhý senzor a máme měření absolutní.

Zjistivše absolutní hodnoty senzoru, můžeme přistoupit ke zjištění zbylých dvou souřadnic *y* a *z.* Víme, že jsou stejné jak v původním bodě, tak i v bodě, kde robot protnul naši rovinu, jelikož během pohybu se měnila pouze souřadnice *x.*

Měření každého jednotlivého senzoru nám dává kouli v prostoru, kde se může hledaný bod nalézat. Průnikem dvou koulí, se středy v senzorech, je kružnice. Průnikem této kružnice a naší roviny jsou dva body. Ze znalosti

reálného prostředí víme, že bod pod úrovní senzorů je nerealizovatelný, tudíž nám zbývá poslední bod, jehož souřadnice potřebujeme zjistit.

Na obrázku 33 je vidět trojúhelník, který leží v naší rovině dané dvěma senzory a osou z.



Obrázek 33: Výpočet souřadnic koncového bodu

Strany *a,b* jsou hodnoty ze senzorů zjištěné výše. Strana *c* je vzdálenost mezi oběma senzory, která je nám rovněž známa. Naší snahou je spočítat výšku v tomto trojúhelníku. Ze sinové věty víme, že poměr stran je roven poměru úhlů naproti nim. Tudíž:

$$\frac{v}{a} = \frac{\sin\beta}{\sin\frac{\pi}{2}}$$

Musíme tedy zjistit úhel β, který spočteme tentokráte z věty kosinové:

$$b^2 = c^2 + a^2 - 2 * c * a * \cos\beta$$

Po vyjádření a dosazení dostáváme:

$$v = a * \sin\left(\operatorname{acos}\left(\frac{b^2 - c^2 - a^2}{-2 * a * c}\right)\right)$$

Jelikož víme, že úhel β leží v prvním kvadrantu, můžeme tuto rovnici bez problémů použít. Při znalosti, že pata kolmice leží na přímce mezi oběma senzory, můžeme určit souřadnici *z* u vrcholu trojúhelníka:

$$z = v + z_{senzor} = v - 12$$

Stejnou souřadnici má i původní koncový bod robota.

Poslední souřadnici budeme opět počítat z výše zobrazeného trojúhelníka. Potřebujeme spočítat vzdálenost mezi bodem B a patou výšky na stranu *c*, což učiníme z následující rovnice a označíme si ji jako *d*:

$$d = a * \cos\beta$$

Což si opět při znalosti kosinové věty můžeme přepsat jako:

$$d = a * \left(\frac{b^2 - c^2 - a^2}{-2 * a * c}\right)$$

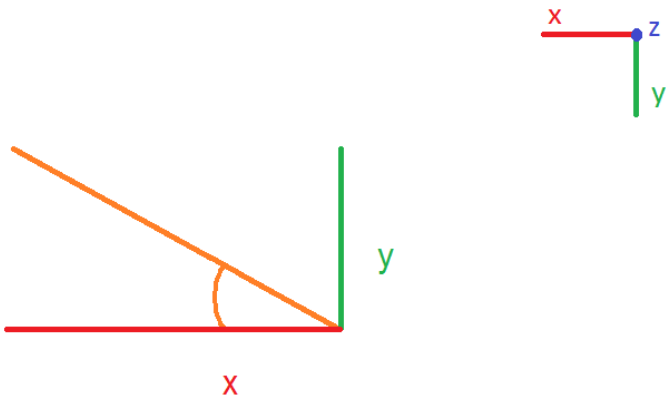
Při znalosti souřadnic bodu B v souřadném systému základny tedy výslednou souřadnici *y* vrcholu trojúhelníka můžeme dopočítat jako:

$$y = y_{senzor} + d = -291 + d$$

Stejnou souřadnici má i původní koncový bod robota.

V tuto chvíli již známe všechny souřadnice koncového bodu robota a zbývá nám jen určit jednotlivé úhly v jeho kloubech.

Jelikož známe fyzická omezení robota, můžeme dopředu říci, v jakých intervalech budou ležet výsledné hodnoty. Na obrázku 34 je náčrt robota při pohledu shora.

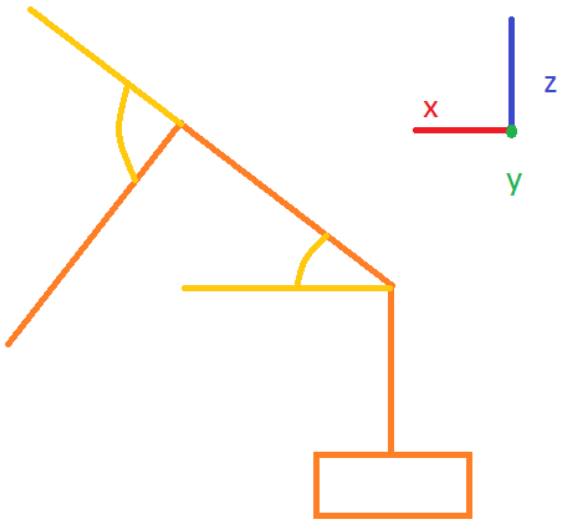


Obrázek 34: Výpočet prvního úhlu

Z tohoto obrázku je patrné, že úhel pro první kloub (θ_1) lze spočítat:

$$\theta_1 = atan\left(\frac{y}{x}\right)$$

Ani v tomto případě nám definiční obor nebude dělat žádné problémy. Na obrázku 35 je pak robot při pohledu ze strany včetně naznačených hledaných úhlů.



Obrázek 35: Druhý a třetí úhel

5.2 Denavitova-Hartenbergova notace

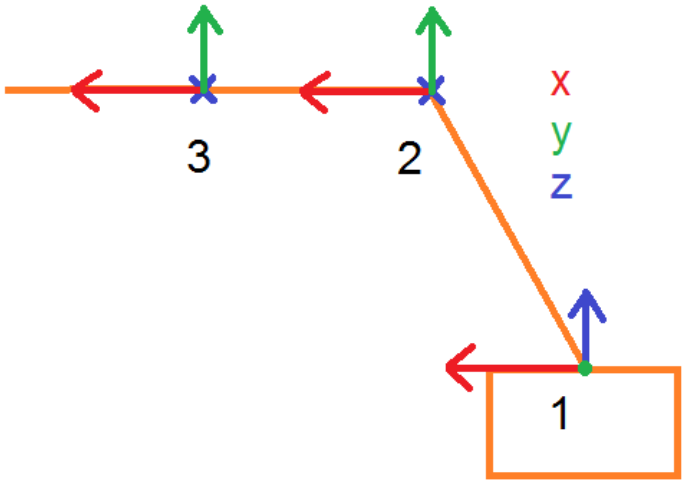
Denavitova-Hartenbergova notace je postup, jehož výsledkem jsou zvolené souřadnicové systémy a transformační matice, popisující vztahy mezi nimi. Matice jsou popsány čtyřmi parametry: *d, θ, r, a*. [7]

5.3 Transformační matice

V tomto bodě se snažíme o vyjádření koncového bodu v souřadnicích bodu počátečního. K tomu nám slouží transformační matice *G*. Ta vznikne jako součin jednotlivých transformačních matic pro každou jednotlivou osu robota. Označme si tyto matice *A* s indexy vyjadřujícími spojení počátků jednotlivých souřadných systémů. Tudíž např. matice A_{23}nám udává matici přechodu mezi souřadnicovými systémy dva a tři. Jinými slovy řečeno, je to příspěvek třetí osy. Každá transformační matice se vytváří podle Denavitovy-Hartenbergovy notace. Pomocí algoritmu vytvoříme pro každou jednotlivou dvojici kloubů matici *A*, která má tvar:

$$A = \begin{bmatrix} cos\,\theta & -sin\,\theta\,cos\,\alpha & sin\,\theta\,sin\,\alpha & r\,cos\,\theta \\ sin\,\theta & cos\,\theta\,cos\,\alpha & -cos\,\theta\,sin\,\alpha & r\,sin\,\theta \\ 0 & sin\,\alpha & cos\,\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Pro zjišťování jednotlivých parametrů je vhodné nakreslit si náčrt se všemi osami jednotlivých kloubů, který je vidět na obrázku 36.



Obrázek 36: Náčrt os jednotlivých kloubů

Pro jednotlivé dvojice kloubů si teď můžeme vyjádřit všechny potřebné parametry. Jsou zapsány do tabulky 4.

Dvojice kloubů	d [mm]	θ	r [mm]	α [rad]
1-2	77	θ_1	73	$\frac{\pi}{2}$
2-3	0	θ_2	126	0
3-4	0	θ_3	90	0

Tabulka 4: Parametry Denavitovy-Hartenbergovy notace

Nyní můžeme parametry dosadit a dostáváme jednotlivé matice *A*:

$$A_{12} = \begin{bmatrix} cos\ \theta_1 & 0 & sin\ \theta_1 & 73\ cos\ \theta_1 \\ sin\ \theta_1 & 0 & -cos\ \theta_1 & 73\ sin\ \theta_1 \\ 0 & 1 & 0 & 77 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{23} = \begin{bmatrix} cos\ \theta_2 & -sin\ \theta_2 & 0 & 126\ cos\ \theta_2 \\ sin\ \theta_2 & cos\ \theta_2 & 0 & 126\ sin\ \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{34} = \begin{bmatrix} cos\ \theta_3 & -sin\ \theta_3 & 0 & 90\ cos\ \theta_3 \\ sin\ \theta_3 & cos\ \theta_3 & 0 & 90\ sin\ \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Hodnoty *x,y,z* jsou v maticích na místech jedna až tři v posledním sloupci. Jelikož známe θ_1, můžeme určit, kde se nachází druhá osa:

$$x_2 = 73\ cos\ \theta_1$$

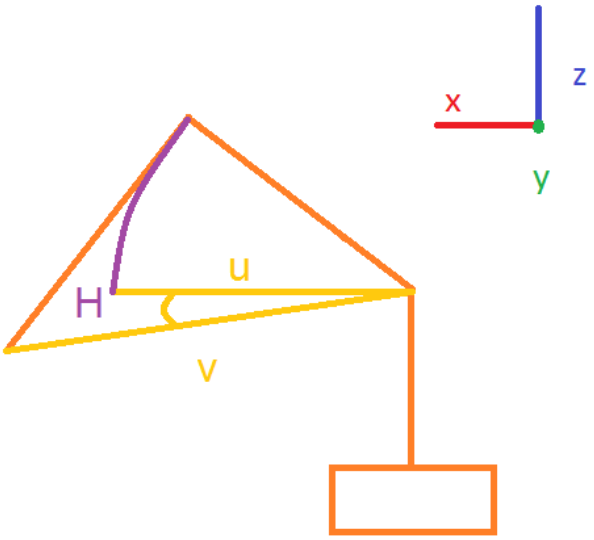
$$y_2 = 73\ sin\ \theta_1$$

$$z_2 = 77$$

Tímto dostáváme souřadnice důležitého bodu pro zbytek výpočtů. Jelikož známe také koncový bod a délky jednotlivých ramen robota, zbývá nám zjistit, kde se nachází třetí osa.

5.4 Výpočet druhé a třetí osy

Nejprve se podíváme, o kolik by se otočila druhá osa, kdyby směřovala přímo do koncového bodu, což je patrné na obrázku 37.



Obrázek 37: Hledání hodnoty druhé osy

Nejprve si spočtěme souřadnice bodu H:

$$x_H = x_2 + 126\ cos\ \theta_1$$

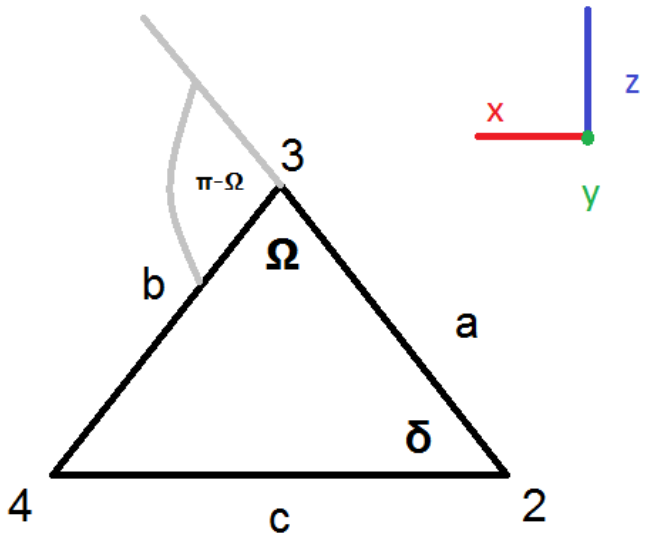
$$y_H = y_2 + 126 \sin \theta_1$$

$$z_H = z_2$$

Definujme si vektor $\vec{u}$ jako rozdíl souřadnic bodu H a druhé osy. Dále definujme vektor $\vec{v}$ jako rozdíl souřadnic koncového bodu a druhé osy. Úhel mezi těmito dvěma vektory je poté dán:

$$\varphi = \text{acos}\left(\frac{\vec{u}\,.\,\vec{v}}{|\vec{u}||\vec{v}|}\right)$$

Kde „tečka" značí skalární součin vektorů. Pokud koncový bod má souřadnici *z* menší než 77 (souřadnice *z* druhé osy), tak hodnota úhlu φ je záporná. Samotná velikost nám říká o kolik je otočen trojúhelník na obrázku 38 oproti nulové poloze. V tomto trojúhelníku nás zajímá úhel δ a Ω.



Obrázek 38: Výpočet hodnot druhé a třetí osy

Jelikož známe souřadnice bodů 2 a 4 i délky všech stran, můžeme dopočítat všechny pro nás tak důležité úhly. Souřadnice bodu 3 nás v tuto chvíli nezajímají. Opět použijeme kosinovou větu, abychom spočetli:

$$\delta = \operatorname{acos}\left(\frac{b^2 - c^2 - a^2}{-2 * a * c}\right)$$

$$\Omega = \operatorname{acos}\left(\frac{c^2 - a^2 - b^2}{-2 * a * b}\right)$$

V posledním kroku výpočtů nám zbývá jen vypočítat natočení druhé a třetí osy:

$$\theta_2 = \delta + \varphi$$

$$\theta_3 = \pi - \Omega$$

Pro kontrolu dosadíme tyto tři úhly do Denavitovy-Hartenbergovy notace a ve výsledné matici se nám musí příslušné hodnoty pro *x,y,z* shodovat se souřadnicemi koncového bodu.

6) Výsledky

6.1 Naměřené hodnoty

Algoritmus byl proveden pro dvacet rozdílných bodů v pracovním prostoru robota. Pro každý bod pak proběhlo také ruční měření polohy, pro možnost ověření výsledků a jejich porovnání. Toto měření bylo prováděno pomocí měřicího pásma a posuvného měřidla, schopného měřit s přesností na desetinu milimetru. Pro naměřené hodnoty pak byla vypočítána inverzní kinematická úloha. Všechny hodnoty, zaokrouhlené pro lepší přehlednost na tři desetinná místa, jsou zapsány v tabulce 5.

Číslo měření	Spočítaná hodnota θ_1 [rad]	Spočítaná hodnota θ_2 [rad]	Spočítaná hodnota θ_3 [rad]	Skutečná hodnota θ_1 [rad]	Skutečná hodnota θ_2 [rad]	Skutečná hodnota θ_3 [rad]
1	0,035	0,267	-1,128	0,035	0,260	-1,114
2	-0,261	0,284	-1,115	-0,261	0,279	-1,111
3	0,043	0,286	-1,011	0,043	0,280	-1,009
4	0,040	0,501	-1,202	0,040	0,508	-1,203
5	-0,427	0,232	-0,649	-0,427	0,230	-0,646
6	-0,217	0,348	-1,103	-0,217	0,346	-1,095
7	0,376	0,369	-1,055	0,376	0,362	-1,045
8	0,426	0,293	-0,686	0,426	0,282	-0,686
9	-0,170	0,435	-1,316	-0,170	0,426	-1,302
10	0,366	0,436	-1,191	0,366	0,421	-1,180
11	-0,183	0,397	-1,238	-0,183	0,391	-1,227

12	-0,391	0,274	-0,740	-0,391	0,272	-0,738
13	0,458	0,405	-1,005	0,458	0,400	-1,000
14	0,569	0,356	-0,814	0,569	0,350	-0,813
15	0,341	0,440	-1,229	0,340	0,434	-1,217
16	0,379	0,390	-1,014	0,378	0,391	-1,010
17	0,367	0,437	-1,173	0,367	0,437	-1,164
18	0,308	0,315	-0,926	0,308	0,301	-0,922
19	-0,424	0,328	-0,944	-0,424	0,320	-0,937
20	0,555	0,307	-0,515	0,555	0,298	-0,532

Tabulka 5: Výsledky kalibrace

6.2 Přepočet výsledků na reálného robota

Jelikož jsme si zvolili souřadný systém našeho robota tak, aby se nám snáze počítalo, je nutné výsledné úhly pro reálné použití ještě přepočítat. Nulovou polohu má robot ve skutečnosti o čtvrt otáčky první osy posunutou v záporném směru otáčení. Z toho vyplývá, že reálná hodnota první osy je o $\frac{\pi}{2}$ větší. Druhá a třetí osa mají ve skutečnosti přesně opačný směr, což znamená, že u jejich hodnot musíme změnit znaménka:

$$\theta_{1R} = \theta_1 + \frac{\pi}{2}$$

$$\theta_{2R} = -\theta_2$$

$$\theta_{3R} = -\theta_3$$

6.3 Vyhodnocení

Z tabulky můžeme vyčíst, že maximální chybu se nám podařilo stlačit pod jednu šedesátinu radiánu. Co se týče prvního kloubu, zde se chyby pohybovaly maximálně do patnácti stotisícin. Hůře na tom jsou klouby dva a tři, u kterých se chyby pohybují do již zmíněné jedné šedesátiny radiánu. Je to dáno tím, že poslední dvě osy otáčení jsou rovnoběžné a tudíž se navzájem více ovlivňují. Navíc v případě špatného určení první osy se tato chyba přenáší i na osy následné. Minimální chyba byla prakticky u všech kloubů stejná a to kolem jedné desetitisíciny. Co se týče případného vylepšení, určitě by stály za zmínku senzory, které by dávaly absolutní hodnoty, ne jen relativní, jako tomu bylo v našem případě. Dále by přicházela v úvahu kalibrace relativních senzorů před vlastním měřením na testovacím bodu vzdáleném o známou hodnotu. Tím by se výrazně ulehčil výpočet a především zvýšila přesnost, jelikož by odpadlo dopočítávání absolutních naměřených hodnot. Dále by bylo zajímavé použití třetího senzoru zejména ohledně nacházení jeho optimálního umístění vzhledem k ostatním senzorům. Aplikace výsledného algoritmu je omezena testovaným typem robota. V případě průmyslových šestiosých robotů by bylo zapotřebí algoritmus upravit (absence možnosti posuvu) a především nepoužívat měření v relativních hodnotách. Přínosem této práce je jednoduché nalezení aktuální pozice při ztrátě okamžitých hodnot robota v případě výpadku napájení.

Závěr

Cílem diplomové práce bylo nalezení vhodné metody kalibrace sériových manipulátorů. Nejprve jsem se seznámil s obecnou kinematikou robotů a pojmy s ní souvisejícími. Dále jsem se zaměřil na jednotlivé možnosti kalibrace včetně počtu senzorů a jejich polohy. Snažil jsem se o co nejjednodušší algoritmy, které by byly v praxi použitelné. Zajímavý byl nápad ohledně umístění senzorů přímo na robota, který se nabízí pro velké průmyslové roboty. Jeho nevýhodou je nemožnost kalibrace první osy, vyniká naopak jednoduchostí výpočtu. Po seznámení se s reálným robotem a řídicím systémem *REX* jsem vhodně umístil senzory měření vzdálenosti do dříve vymyšlené pozice. Dále jsem zjistil omezení dané reálnými parametry senzorů a robotem. Poté jsem již mohl přistoupit k samotnému algoritmu kalibrace jednotlivých os, který byl přizpůsoben na míru robotovi. Při výpočtech jednotlivých hodnot bylo využito inverzní kinematiky ve spolupráci s Denavitovou-Hartenbergovou notací a pomohl jsem si také geometrií. Největším problémem byl časový rozdíl mezi pohybem robota a příjmem signálu ze senzorů. Po jeho vyřešení posunutím časových údajů v senzoru, jsem již mohl naměřit potřebná data.

Vybral jsem si v pracovním prostoru robota dvacet bodů a pro každý jsem nejprve nasbíral data pomocí senzorů a poté, abych měl k vypočteným datům také nějakou kontrolu, jsem pomocí měřicího pásma a posuvného měřidla zjistil polohu také ručně. Odchylku výsledných a vypočtených hodnot se mně podařilo stlačit pod jednu šedesátinu radiánu. Nejlépe na tom byla první osa, s chybou do patnácti stotisícin radiánu, jelikož její hodnota není ovlivněna jinými osami.

Pro vylepšení algoritmu do budoucna bych použil třetí senzor měření vzdálenosti a případně bych se také zaměřil na získávání reálných vzdáleností ze senzorů.

Použitá literatura a zdroje:

[1] Martin Švejda [online] *Kinematika robotických architektur* [citováno 12.4.2016]. Dostupný z WWW: http://home.zcu.cz/~msvejda/URM/materialy/KinematikaRobotArchitektur.pdf

[2] ABB [online] *IRB 2600* [citováno 12.4.2016]. Dostupný z http://new.abb.com/products/robotics/industrial-robots/irb-2600

[3] ABB [online] *IRB 360 FlexPicker* [citováno 12.4.2016]. Dostupný z http://new.abb.com/products/robotics/industrial-robots/irb-360

[4] MicroEpsilon [online] *WPS MK30* [citováno 17.4.2016]. Dostupný z http://www.microepsilon.com/download/products/_displacement/dax--wireSENSOR-MK30--en-us.html

[5] MicroEpsilon [online] *WPS 500 MK30* [citováno 17.4.2016]. Dostupný z https://www.google.cz/search?q=micro+epsilon+wps-500-mk30-e&rlz=1C1FDUM_enCZ472CZ472&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiKss7HhJbMAhUEIpoKHVaGAPEQ_AUIBygB&biw=784&bih=703#imgrc=p4EpwjrCiDI9sM%3A

[6] Kolektiv autorů, *Pravidla českého pravopisu*, 5. doplněné vydání, Praha: Fin Publishing, 2009, 464 stran, ISBN 80-86002-89-7

[7] Haruhiko Asada & Jean-Jacques E. Slotine, *Robot Analysis and Control,* John Wiley and Son, New York, Spojené státy, 1986, 286 stran, ISBN 9780471830290