

Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

Inverzní kinematika robota sestrojeného na bázi stavebnice Merkur

Bakalářská práce

Jaromír Čada

Školitel: Mgr. Jiří Pech

České Budějovice 2019

Čada, J., 2019: Inverzní kinematika robota sestrojeného na bázi stavebnice Merkur
[Inverse kinematics of the robot constructed on the basis of the Merkur kit. Bc. Thesis, in
Czech] – 40 p., Faculty of Science, University of South Bohemia, České Budějovice,
Czech Republic.

Anotace:

Tato bakalářská práce se zabývá problematikou inverzní kinematiky a její aplikací na robotickou ruku postavenou ze stavebnice Merkur Beta 6°. Obsahuje implementaci rovnic do programu napsaného v prostředí Matlab a vytvoření funkcí pro výpočet inverzní kinematiky metodou FABRIK. Dále ukazuje, jak propojit program s programovatelnou platformou Arduino a pomocí poslaných dat ovládat robotickou ruku. Výsledky práce jsou na konci ověřeny vzorovým programem simulující manipulátor.

Abstract:

This Bachelor thesis deals with problematics of inverse kinematics and its application to a robotic arm built from building kit - Merkur Beta 6°. It contains implementation of equations into a program written in Matlab workspace and creation of functions for computing inverse cinematics by FABRIK method. Further it shows, how to set up connection with an Arduino programmable platform and control the robotic arm with sent data. The results of this thesis are verified at the end by sample program, which simulates a manipulator.

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b č. 111/1998 Sb. V platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. Zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

České Budějovice, 9. 12. 2019

.....

Podpis

Poděkování

Chtěl bych poděkovat vedoucímu bakalářské práce Mgr. Jířímu Pechovi za ochotu, odborné vedení, cenné rady a veškeré informace, které mi velmi pomohli při zpracování bakalářské práce.

Obsah

Úvod	1
1 Základní rozdělení robotů.....	2
1.1 Dělení dle konstrukce	2
1.1.1 Sériový robot	2
1.1.2 Paralelní robot	2
1.1.3 Kombinovaný robot.....	3
1.2 Dělení dle druhu pohonu.....	3
1.3 Dělení dle funkčního hlediska	3
2 Výrobci průmyslových robotu.....	5
2.1 FANUC	5
2.2 YASKAWA	5
2.3 ABB	5
2.4 Kawasaki.....	6
2.5 Nachi-Fujikoshi Corp.	6
2.6 Denso	7
2.7 KUKA	8
2.8 MITSUBISHI.....	8
2.9 STÄUBLI.....	8
3 Arduino.....	9
3.1 Arduino shieldy.....	9
3.1.1 Ethernet Shield	9

3.1.2	Wifi Shield.....	9
3.1.3	Motor Shield.....	9
3.1.4	GPS Shield.....	10
3.2	Arduino Nano	10
4	Úvod do kinematiky	11
4.1	Mechanický pohyb.....	11
4.2	Trajektorie a dráha hmotného bodu	11
4.3	Rychlost hmotného bodu	11
5	Kinematický model	12
5.1	Přímá kinematika	12
5.1.1	Denavit-Hartenbergova notace	13
5.2	Inverzní kinematika	18
5.2.1	Goniometrická metoda	19
5.2.2	Metoda FABRIK	24
6	Matlab.....	28
6.1	Komunikace mezi programem a deskou Arduino Nano.....	28
6.2	Metoda FABRIK v programu Matlab.....	30
6.2.1	Metoda FABRIK pro rovinnou úlohu	32
6.3	Volání funkce z jiného souboru	35
6.4	Vzorový program.....	36
7	Závěr.....	39
	Literatura	40
	Seznam obrázků.....	43

Zdroje obrázků.....	43
---------------------	----

Úvod

Svět, jak ho dnes známe, bychom si bez robotů nedokázali představit. Ať už mluvíme o robotech v domácnosti, jako jsou kuchyňští roboti, nebo v průmyslu. Plno lidí by se neobešlo bez autonomního robotického vysavače nebo kávovaru. Jak jsme my závislí na těchto přístrojích, tak průmysl je čím dál více závislý na robotech a automatizaci. Z ekonomického hlediska toto představuje nedílnou součást moderní firmy. Využití těchto strojů je prakticky neomezené a záleží jen na lidské představivosti. Mezi základní použití patří například svařování nebo manipulace s materiálem. Mnoho lidí si neuvědomuje, že se s nimi, nebo aspoň jejich vlivem, setkáváme denně, když se podíváme na auto, které pravděpodobně vyjelo z automatizované linky nebo bylo nalakováno roboty. Ačkoliv si to mnohokrát nechceme přiznat, tyto přístroje jsou v mnoha ohledech lepší než lidská síla. Mezi jejich přednosti patří přesnost, rychlost a neúnavnost. Přeci jen člověk má v těchto oblastech jisté mezery. V této práci se budu zabývat problematikou inverzní kinematiky neboli výpočtu úhlů natočení jednotlivých kloubů pro nastavení robotické ruky Merkur Beta 6° a dosažení požadované pozice koncového bodu. Dále tento výpočet implementuji do programovacího jazyka prostředí Matlab. Ten následně propojím se řídicí deskou Arduino Nano. Následně získané výsledky z výpočtů otestuji na reálném robotu, pomocí ilustračního programu.

1 Základní rozdělení robotů

Existuje několik parametrů, podle kterých můžeme roboty rozdělit do několika kategorií.

1.1 Dělení dle konstrukce

Dle typu konstrukce můžeme roboty dělit na sériové, paralelní nebo kombinované.

1.1.1 *Sériový robot*

Tento typ robota se skládá z ramen vzájemně propojených rotačními či posuvnými vazbami. Výsledný pohyb se skládá z jednotlivých na sebe navazujících pohybů. Členy se mohou pohybovat nezávisle na sobě, což umožňuje velkou flexibilitu. Nevýhodou tohoto řešení bývá menší tuhost ve vazbách, a tudíž větší nepřesnost end-effectoru. Tento typ se často používá jako podavač, nebo svařovací robot. [1]



Obr. 1 - Sériový robot [2]

1.1.2 *Paralelní robot*

Jednotlivé části tohoto manipulátoru jsou spojeny paralelně, a tudíž výsledný pohyb vzniká pohybem všech jeho částí. Pohyb jednoho ramena ovlivňuje a zároveň je ovlivňován druhým. Z konstrukčního hlediska je toto řešení tužší, a tedy ho lze použít u aplikací, kde potřebujeme dosáhnout přesného polohování. Mezi velké nevýhody tohoto typu patří relativně malý pracovní prostor. S tímto typem se často setkáváme u různých trenažerů jako např. v letectví. [1]

Adept Quattro s650HS



Obr. 2 – Paralelní robot [3]

1.1.3 Kombinovaný robot

Tento typ vzniká spojením jednotlivých sériových a paralelních substruktur. Do této skupiny lze zařadit chodící stroje, mechanické ruky apod. [1]

1.2 Dělení dle druhu pohonu

Dle použití pohonů lze roboty rozdělit do tří skupin. Nejběžnějším způsobem pohonu v dnešní době jsou elektromotory. Dalšími, často používanými, jsou hydraulické, které se využívají tehdy, jestliže chceme dosáhnout velkých sil. Posledním druhem jsou pneumatické pohony pro velké rychlosti.

1.3 Dělení dle funkčního hlediska

Z funkčního hlediska roboty dělíme podle oblasti nasazení. Jedná se o roboty průmyslové (užívané ve výrobě) a roboty servisní (užívání při obslužných činnostech, např. domácí práce, zdravotnictví, údržba průzkum či armáda).



Obr. 3 – Automatizovaná linka [4]

2 Výrobci průmyslových robotů

Používání robotů v průmyslu zjednodušuje a urychluje práci. Jejich využití je možné vidět přes celou škálu různých odvětví. V dnešní době jsou roboti nedílnou součástí zemědělství, stavebnictví, strojírenství, těžarství i lékařství. V této kapitole si představíme několik nejznámějších výrobců.

2.1 FANUC

Jednou z největších společností je japonská firma FANUC s více než 450 000 nainstalovanými roboty v roce 2017. Její sídlo se nachází ve městě Fanuc v Japonsku na úpatí hory Fuji. Firma byla založena v roce 1956 Dr. Seiemonem Inabou. Zakladatel je spoluvůrcem číslicového řízení (NC). V roce 1973 se společnost FANUC objevuje na evropském trhu instalací prvního CNC stroje. Výroba inteligentních robotů začíná v roce 2003. O 10 let později se stává předním světovým výrobcem průmyslových automatizovaných zařízení. V dnešní době evropské pobočky působí pod jménem FANUC Europe Corporation. Katalog společnosti nabízí více než 100 modelů robotů a díky tomu má nejširší nabídku na světě. Jeden z modelů má užitečné zatížení až 2,3 tuny a dosah 4,7 metru. [5]

2.2 YASKAWA

Na druhé místo, s přibližně 300 000 nainstalovanými roboty (v roce 2017), se řadí společnost s bohatou historií, YASKAWA. Firma byla založena v roce 1915 Keiichirem Yasukawou. Z počátku svého působení se firma zabývala výrobou různých elektromotorů a první NC stroj byl představen až roku 1968. Právě zakladatel této společnosti stál za zrodem pojmu mechatronika. V roce 1973 firma předvádí svůj první CNC stroj používající mikroprocesor. Na evropský trh se značka YASKAWA dostává roku 1981 založením YASKAWA Electric Europe GmbH. Aktivity společnosti jsou rozděleny do 4 sektorů: kontrola pohybu, robotika, inženýring a informační technologie. V dnešní době YASKAWA má přibližně 14 500 zaměstnanců po celém světě. [6]

2.3 ABB

Společně na druhé příčce s YASKAWOU je společnost ABB. Zrod firmy začíná v roce 1883 založením Elektriska Aktiebolaget ve Stockholmu panem Ludvigem Fredholmem.

ABB vzniká roku 1988 sloučením Elektriska Aktiebolaget a BBC, která byla založena v roce 1891 Charlesem Brownem. Hlavní sídlo firmy se nachází v Curychu. ABB má dvě hlavní pole: energetiku a automatizaci. Celosvětově firma působí ve 100 zemích a má více než 124 000 zaměstnanců. Z toho v oblast robotiky zasahuje do 53 zemích. [7]

2.4 Kawasaki

Dalším z předních výrobců je japonská společnost Kawasaki se sídlem v Akashi a padesátiletou zkušeností v oblasti robotiky. V roce 2017 bylo instalováno přibližně 110 000 robotů. Kawasaki začalo s produkcí průmyslových robotů v Japonsku roku 1969. Výroba začala ve spolupráci s americkou firmou Unimate. Po necelých 20 letech však spolupráce skončila. Následně začali vyvíjet své vlastní roboty. Roku 1995 Kawasaki založilo dceřinou společnost v Německu a rok poté ve Velké Británii. V oblasti lékařství se společnost objevuje v roce 2013 založením Medicaroid Corporation pro vývoj robotů pro lékařské účely. Firma nabízí širokou škálu aplikací v různých odvětvích průmyslu. Robot Division firmy Kawasaki působí v mnoha částech světa od Spojených států amerických až po Indii. V Evropě má společnost sídla ve Velké Británii a Německu. [8]

2.5 Nachi-Fujikoshi Corp.

V roce 1928 založil Kohki Imura ve městě Toyama v Japonsku společnost FUJIKOSHI STEEL INDUSTRY CO. V roce 1963 se změnil název společnosti na NACHI-FUJIKOSHI CORP. Prvotním záměrem zakladatele bylo vyrábět řezné nástroje a obráběcí stroje pro domácí trh při modernizaci průmyslu. O rok později začala firma vyrábět listy pro rámové pily. Po necelých 20 letech byla výroba rozšířena o stroje pro potravinářský průmysl, jízdní kola, tříkolky, ocelové plechy, dieselové motory nebo motocykly.

V roce 1958 započala výroba hydraulických zařízení a v roce 1969 vyvinula společnost NACHI hydraulické průmyslové roboty. Patří proto mezi nejstarší výrobce průmyslových robotů v Japonsku. V roce 1979 vyvinuli jako první na světě motoricky poháněné roboty pro bodové svařování, čímž se stali průkopníky automatizace. První manipulační roboty uvedli na trh roku 1990.

V roce 2013 představila společnost NACHI nejrychlejší kompaktní robot na světě MZ07. Projevuje se jako jeden z celosvětových čelních výrobců high-tech mechatronických systémů vyrábějících všechny potřebné součásti ve vlastních výrobních zařízeních.

Počátkem prosince 2017 představila tato společnost na výstavě robotiky a automatizace IREX nový model robotu MZ12 z úspěšné řady kompaktních robotů MZ.

V současnosti se společnost zabývá výrobou průmyslových robotů (elektronických nebo hydraulických) pro automobilový průmysl, pro strojírenství, pro moderní výrobní linky, pro manipulaci s těžkými břemeny, pro bodové svařování nebo pro svařování elektrickým obloukem. [9] [10]

2.6 Denso

Historie společnosti Denso sahá do roku 1949, kdy byla založena společnost Nippondenso Co., Ltd., jako samostatný subjekt, vzniklý odštěpením od společnosti Toyota Motor Co., Ltd. v prefektuře Aichi v Japonsku. V roce 1996 se změnil název společnosti na DENSO Corporation.

Díky technické spolupráci s německou společností Robert Bosch GmbH., která byla zahájena v roce 1953, bylo umožněno rozšířit technické zkušenosti mimo hranice společnosti a dosáhnout tak světové úrovně v oblasti strojírenství.

Vývoj robota zahájila společnost už v roce 1967. O tři roky později vyrobila prvního skutečného robota pro tlakové lití hliníku a v roce 1991 byl uveden na trh komerční model robota první generace.

Součástí společnosti Denso Corporation je společnost Denso Robotics, která je jedním ze dvou největších dodavatelů automobilového průmyslu na světě. Po celém světě je přibližně 100 000 robotů, z nichž více než 20 000 je využíváno ve vlastní výrobě společnosti.

V roce 2001 byla založena dceřiná společnost Denso Wave Incorporated, která vyvíjí a vyrábí automatizační produkty pro automatizaci výrobních závodů (průmyslové roboty) a automatické identifikační produkty (čtečky čárových kódů a kódů QR).

Pozornost veřejnosti získal robot společnosti Denso v roce 2014, kdy hrál Japonské šachy proti profesionálním hráčům. [11] [12]

2.7 KUKA

Podnik KUKA byl založen v roce 1898 v německém Ausbugu Johannem Josefem Kellerem a Jakobem Knappichem. Původně se jednalo o acetylénový závod na výrobu osvětlení, jehož cílem byla výroba dostupného pouličního a domácího osvětlení.

Zanedlouho se firma přeorientovala na oblast svářečské techniky a v roce 1905 přišla s novým vynálezem: svařování autogenem. V roce 1936 firma zkonstruovala první elektrické bodové svařovací kleště v Německu a roku 1972 firma představila první svářečku s technologií Magnetarc. Dodnes se jedná o jednu z nejrychlejších a nejspolehlivějších metod při spojování konstrukčních dílů, které musí odolávat nejvyššímu zatížení.

Již v roce 1973 se společnost KUKA stala průkopníkem robotiky, když vyvinula celosvětově prvního průmyslového robota se šesti elektromechanicky poháněnými osami - FAMULA. V roce 1994 byla založena společnost KUKA Robotics Corporation se sídlem v Michiganu. Roku 2007 byl uveden na trh největší a nejsilnější šestiosý průmyslový robot světa, s nosností 1 000 kg a s dosahem 3 200 mm. Tento robot byl zapsán do Guinnessovy knihy rekordů. V roce 2013 představila společnost KUKA celosvětově prvního průmyslového robota lehké konstrukce s integrovanou sensorikou v každé ose. [13]

2.8 MITSUBISHI

Původně loďařskou společnost Mitsubishi založil v roce 1870 v Osace Yataro Iwasaki. V následujících letech se společnost rozrostla na nadnárodní úroveň a začala se věnovat i dalším odvětvím, jako např. těžební průmysl, loďařství, bankovníctví, obchodování a realitní činnost.

V současnosti firma Mitsubishi nabízí průmyslové roboty MELFA. Jedná se o roboty s kloubovými rameny se 6 osami a také roboty se 4 osami pro montážní a paletovací aplikace. [14]

2.9 STÄUBLI

Společnost Stäubli byla založena v roce 1892 ve švýcarském Horgenu dvojicí Rudolphem Schellingem a Hermannem Stäubli. Původně to byla jen malá dílna, která se rozrostla na mezinárodní úroveň se sídlem ve švýcarském Pfäffikonu.

Od svého založení expandovala firma Stäubli do tří různých oblastí výroby a služeb - textil, spojovací součástky a robotika. Divize zaměřena na automatizaci a robotiku byla založena v roce 1982 a v současnosti nabízí jak malé čtyřosé SCARA roboty, tak velké roboty pracující se zátěží až 250 kg. Rovněž nabízí lakovací roboty, frézovací roboty i roboty pro čisté průmyslové prostředí nebo do náročných provozů. [15]

3 Arduino

Arduino je otevřená programovatelná platforma založená na mikrokontrolerech ATmega od firmy Atmel. Jedním z hlavních kladů je, že má uživatelsky jednoduchý hardware a software. Dnes již existuje řada typů s různými mikrokontrolery. Využití najde především při výuce ve školách. Desky Arduino byly poprvé představeny v roce 2005. Základní vývojové prostředí se jmenuje Arduino IDE, ale platformu lze programovat i z prostředí Matlab po stažení *MATLAB Support Package for Arduino*. [16]

3.1 Arduino shieldy

Pole působnosti rozšiřují tzv. shieldy. Jsou to rozšiřující moduly, které je možné zapojit na standartní piny Arduina. Ty následně mohou společně komunikovat. Podle typu komunikují buď skrze SPI (sériové periferní rozhraní), I²C (multi-masterová počítačová sběrnice) nebo po sériové lince. [17]

3.1.1 Ethernet Shield

Tento rozšiřující modul umožňuje připojení Arduina k internetu. Na shieldu se nachází konektor RJ-45 pro připojení síťového kabelu a slot pro paměťové microSD karty. Komunikace probíhá přes SPI. [17]

3.1.2 Wifi Shield

Wifi shield je možné pomocí bezdrátové sítě Wi-Fi připojit k internetu. Na shieldu se nachází opět slot pro micro SD kartu a mini USB určený hlavně k aktualizaci firmwaru. Komunikace probíhá přes SPI. [17]

3.1.3 Motor Shield

Tento modul je určen k jednoduchému ovládní servomotorů, stejnosměrných a krokových motorů. Pro připojení motorů jsou zde čtyři piny (M1 až M4). Pro energetickou

náročnost k řízení motorů je zde konektor na externí napájení. Rozsah napájecího napětí je od 4,5 V do 25 V. Pro proud je maximální velikost 0,6 A ve špičkách až 1,2 A. [17]

3.1.4 *GPS Shield*

Při zapojení přidává na Arduino GPS modul pro získání aktuální polohy, nadmořskou výšku nebo rychlost pohybu. Pro získání dat GPS je stačí přečíst ze sériové linky. Shield obsahuje slot pro micro SD kartu. [17]

3.2 Arduino Nano

Arduino Nano je kompaktní elektronická platforma založená na mikrokontroleru ATmega328. Operační napětí je zde 5 V a velikost flash paměti je 32 kB, ze které jsou 2 kB využity pro bootování. Na desce se nachází 8 analogových pinů a 6 PWM výstupů. [18]

4 Úvod do kinematiky

Mechanika se zabývá nejjednodušším fyzikálním jevem, který každý pozoruje kolem sebe, a to mechanickým pohybem. Podle toho, o jaké vlastnosti pohybu se zajímáme, dělíme mechaniku na dvě kategorie:

- a) **kinematiku** – zajímá se o popis pohybu (trajektorie, dráha, rychlost, ...), v kinematice se tedy zajímáme jak se dané těleso či hmotný bod pohybuje;
- b) **dynamiku** – zajímá se o příčiny pohybu, (síly vyvolávající pohyb), neboli zkoumá, proč se těleso či hmotný bod pohybuje. [19]

4.1 Mechanický pohyb

Pohyb je relativní, proto jej vždy pozorujeme vzhledem k jinému tělesu (**vztažné soustavě**). Vztažnou soustavou mohou být skutečná tělesa, nebo smyšlená (soustava souřadnic).

Pro zjednodušení fyzikální situace zavádíme pojem hmotný bod, kdy neřešíme do detailu vnitřní složení nebo tvar, které pohyb neovlivní. Skutečnost nahradíme souřadnicemi ve vztažné soustavě a fyzikálními parametry (např. hmotnost). [19]

4.2 Trajektorie a dráha hmotného bodu

Trajektorii hmotného bodu nazýváme čáru, kterou opisuje hmotný bod při svém pohybu. Podle tvaru trajektorie dělíme pohyby na přímočaré a křivočaré. Délka trajektorie, kterou opíše hmotný bod za určitý čas, se nazývá dráha. [19]

4.3 Rychlost hmotného bodu

Velikost okamžité rychlosti v hmotného bodu v čase t definujeme jako podíl přírůstku dráhy Δs , k němuž dojde za čas Δt , a této doby $v = \frac{\Delta s}{\Delta t}$ (přitom Δt je velmi malé);
[v] = $m \cdot s^{-1}$. [19]

5 Kinematický model

Mechanismus robotické ruky je prostorové zařízení umožňující pohyb end effectoru. Robotická ruka Merkur se skládá z podstavy, rotačních motorů, tří ramen, na jejichž koncích se nacházejí motory umožňující pohyb, a end effectoru. Ramena jsou k sobě vázána pouze rotačními vazbami. Kombinací všech těchto rotací dostaneme pracovní prostor. Zařízení má 6 stupňů volnosti.

5.1 Přímá kinematika

Konstrukce robotické ruky je situována tak, že rotační vazby mezi jednotlivými rameny jsou v jedné rovině, tudíž lze úlohu nejprve zjednodušit na dvojrozměrnou úlohu a k řešení ve třech dimenzích se dostaneme později. K popsání kinematiky robotické ruky musíme znát několik konstrukčních parametrů. Prvním je délka jednotlivých ramen, která je definována jako vzdálenost mezi jednotlivými klouby neboli jointy. Body O, A a B jsou místa kde se nacházejí klouby a bod E koncový bod end effectoru. Délky ramen označíme jako l_1 , l_2 , l_3 , přičemž l_3 je délka od kloubu označeném jako B a koncovým bodem E. Předpokládejme, že pohon 1 pohybuje s prvním ramenem a první rameno je pevně spojeno se základnou, a první rameno je pevně spojeno se základnou θ_1 . Pohon 2 pohybuje s druhým ramenem, druhé rameno je spojeno s koncem prvního ramene. Druhé rameno svírá úhel θ_2 s prvním ramenem. [20]

Třetí rameno je poháněno pohonem 3 a je připojeno ke konci ramene 2 a s tímto ramenem svírá úhel θ_3 . S měnícími se úhly natočení jednotlivých kloubů sledujeme pohyb koncového bodu E. Pozice bodu E je popsána kartézskými souřadnicemi x, y a souřadnice bodu x_e a y_e . Počátek souřadného systému (bod O) je v kloubu 1, který je spojen se základnou. Pozice koncového bodu E zjistíme použitím teorie vektorových n-úhelníků, kdy součet jednotlivých vektorů uzavřeného n-úhelníku je roven 0. Po úpravě získáme soustavu rovnic o 2 neznámých:

$$x_e = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \quad (1)$$

$$y_e = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \quad (2)$$

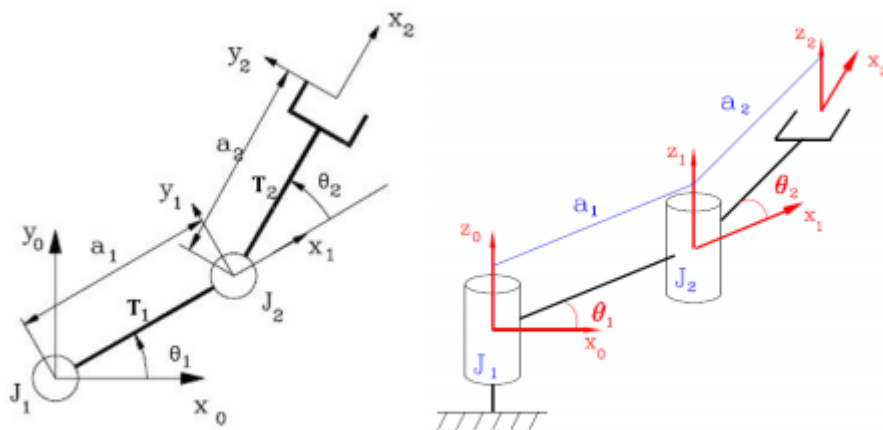
Z těchto rovnic je možné určit polohu koncového bodu E a zároveň úhel natočení posledního ramene ϕ_e vzhledem ke kladné části osy x. Úhel se vypočte jako:

$$\phi_e = \theta_1 + \theta_2 + \theta_3. \quad (3)$$

5.1.1 Denavit-Hartenbergova notace

Existuje několik způsobů, jak postupovat při výpočtu přímé úlohy. Ze znalosti geometrie robota můžeme sestavit sadu rovnic, které vyjadřují koncovou polohu end effektoru. U složitějších struktur může být sestavení rovnic obtížné, proto je často výhodné použít určitých sestavovacích algoritmů.

Nejpoužívanější metodou je Denavit-Hartenbergova notace, která vychází ze zápisu tak zvané homogenní notace T_H . Pro přiblížení postupu si sestavíme rovnice rovinného manipulátoru (obr. 4).



Obr. 4 – Názorné schéma kinematického modelu [24]

Denavit-Hartenbergova notace popisuje transformaci z jedné kinematické vazby do druhé pomocí čtyř dílčích transformací v určitém pořadí – rotace kolem osy z $R_z(\theta)$, posuv podél osy z $P_z(d)$, posuv podél osy x $P_x(a)$, a rotace kolem osy x $R_x(\vartheta)$. Výsledná transformační matice $T_i(\theta, d, a, \vartheta)$ je dána součinem dílčích transformací (4). Takto získáme transformaci pro jednu kinematickou vazbu. Pak výsledná homogenní matice transformace je dána součinem transformací mezi jednotlivými vazbami. [24]

$$T_i = R_z \cdot P_z \cdot P_x \cdot R_x \quad (4)$$

$$T_i = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \vartheta & -\sin \vartheta & 0 \\ 0 & \sin \vartheta & \cos \vartheta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Pro přehlednost si můžeme před dosazením sestavit tabulku Denavit – Hartenbergových parametrů.

Transformace	θ	d	a	ϑ
T_1	θ_1	0	a_1	0
T_2	θ_2	0	a_2	0

Tab. 1 – Tabulka Denavit-Hartenbergových parametrů

Výsledná homogenní matice, jak již bylo zmíněno, je dána součinem transformačních matic pro jednotlivé kinematické vazby. Tyto vztahy jsou vyjádřeny v rovnicích 6,7,8.

$$T_H = T_1 \cdot T_2 \quad (6)$$

$$T_H = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & a_1 \cos \theta_1 \\ \sin \theta_1 & \cos \theta_1 & 0 & a_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & a_2 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & a_2 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$$T_H = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & a_1 \cos \theta_1 + a_2 \cdot \cos(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & a_1 \sin \theta_1 + a_2 \cdot \sin(\theta_1 + \theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Z matice \mathbf{T}_H lze následně vyjádřit posuv a rotaci kartézského souřadného systému.

$$x = a_1 \cdot \cos \theta_1 + a_2 \cdot \cos(\theta_1 + \theta_2) \quad (9)$$

$$y = a_1 \cdot \sin \theta_1 + a_2 \cdot \sin(\theta_1 + \theta_2) \quad (10)$$

$$z = 0 \quad (11)$$

$$\alpha = \theta_1 + \theta_2 \quad (12)$$

$$\beta = 0 \quad (13)$$

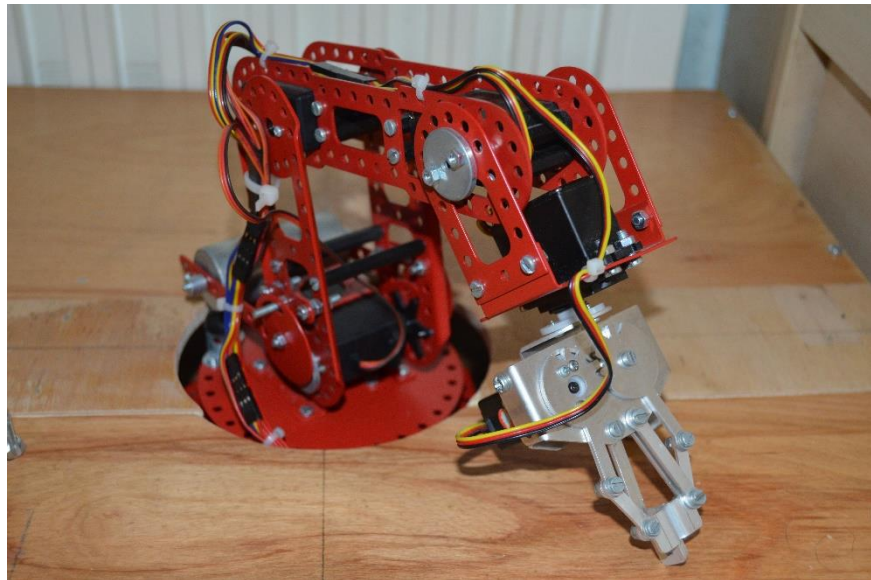
$$\gamma = 0 \quad (14)$$

Polohu koncového bodu dostaneme dosazením parametrů θ_1 a θ_2 do rovnic 9 a 10. [24]

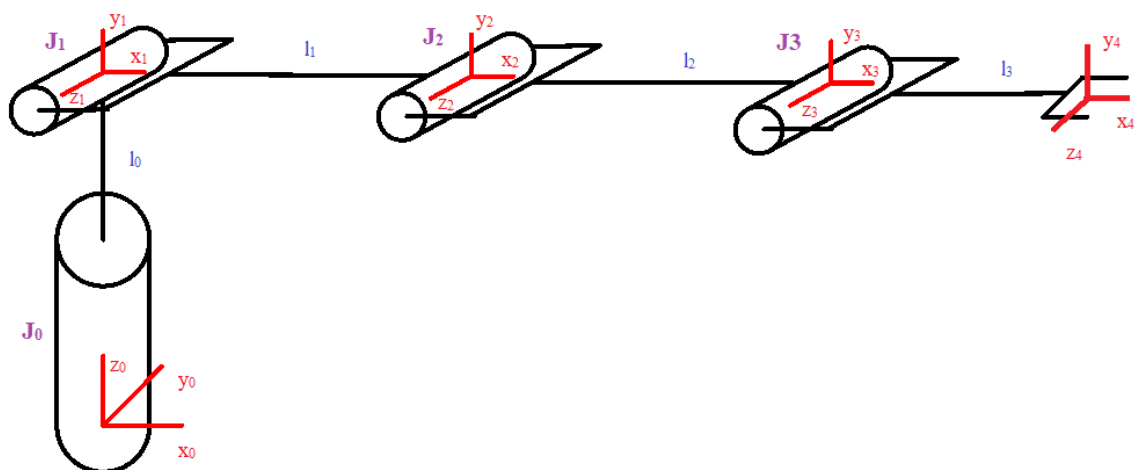
Denavit-Hartenbergova notace pro robota Merkur

Pro náš konkrétní případ bude notace vypadat obdobně jako v ukázkovém příkladu.

V našem případě se bude homogenní matice skládat ze součinu čtyř transformačních matic.



Obr. 5 – Fotografie robota Merkur



Obr. 6 – Schéma kinematického modelu skutečného robota Merkur (Obr.5)

Transformace	θ	d	a	ϑ
T_1	θ_1	l_0	0	$\pi/2$
T_2	θ_2	0	l_1	0
T_3	θ_3	0	l_2	0
T_4	θ_4	0	l_3	0

Tab. 2 – Denavit-Hartenbergova tabulka parametrů pro robota Merkur

Délka l_0 je nulová, neboť pracovní plocha, jak je vidět na obrázku, je ve stejné úrovni jako kloub J_1 . Kloub J_0 s rameno l_0 nikterak nenaklání tudíž má vliv pouze na souřadnici z , která by později musela být upravena kvůli úrovni pracovní plochy.

S touto tabulkou jsme připraveni sestavit transformační matice(15,16,17,18) a následně vyjádřit výsledné vztahy pro kinematický model robota Merkur.

$$T_1 = \begin{bmatrix} \cos \theta_0 & -\sin \theta_0 & 0 & 0 \\ \sin \theta_0 & \cos \theta_0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \frac{\pi}{2} & -\sin \frac{\pi}{2} & 0 \\ 0 & \sin \frac{\pi}{2} & \cos \frac{\pi}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_1 = \begin{bmatrix} \cos \theta_0 & 0 & \sin \theta_0 & 0 \\ \sin \theta_0 & 0 & -\cos \theta_0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

$$T_2 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & l_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & l_1 \cdot \cos \theta_1 \\ \sin \theta_1 & \cos \theta_1 & 0 & l_1 \cdot \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

$$T_3 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 0 \\ \sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & l_2 \cdot \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & l_2 \cdot \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (17)$$

$$T_4 = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & 0 \\ \sin \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4 = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & l_3 \cdot \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & l_3 \cdot \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (18)$$

Výsledná homogenní matice T_H je součinem těchto výsledných transformačních matic.

$$T_H = T_1 \cdot T_2 \cdot T_3 \cdot T_4 \quad (19)$$

$$T_H = \begin{bmatrix} \cos \theta_0 & 0 & \sin \theta_0 & 0 \\ \sin \theta_0 & 0 & -\cos \theta_0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & l_1 \cdot \cos \theta_1 \\ \sin \theta_1 & \cos \theta_1 & 0 & l_1 \cdot \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot$$

$$\begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & l_1 \cdot \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & l_1 \cdot \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & l_3 \cdot \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & l_3 \cdot \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (20)$$

$$T_H = \begin{bmatrix} \frac{\cos(\theta_0 + \theta_1 + \theta_2 + \theta_3) + \cos(\theta_0 - \theta_1 - \theta_2 - \theta_3)}{2} & \frac{-\sin(\theta_0 + \theta_1 + \theta_2 + \theta_3) + \sin(\theta_0 - \theta_1 - \theta_2 - \theta_3)}{2} & \sin \theta_0 & \frac{l_1 \cos(\theta_0 + \theta_1) + l_1 \cos(\theta_0 - \theta_1) + l_2 \cos(\theta_0 + \theta_1 + \theta_2) + l_2 \cos(\theta_0 - \theta_1 - \theta_2) + l_3 \cos(\theta_0 + \theta_1 + \theta_2 + \theta_3) + l_3 \cos(\theta_0 - \theta_1 - \theta_2 - \theta_3)}{2} \\ \frac{\sin(\theta_0 + \theta_1 + \theta_2 + \theta_3) + \sin(\theta_0 - \theta_1 - \theta_2 - \theta_3)}{2} & \frac{\cos(\theta_0 + \theta_1 + \theta_2 + \theta_3) - \cos(\theta_0 - \theta_1 - \theta_2 - \theta_3)}{2} & -\cos \theta_0 & \frac{l_1 \sin(\theta_0 + \theta_1) + l_1 \sin(\theta_0 - \theta_1) + l_2 \sin(\theta_0 + \theta_1 + \theta_2) + l_2 \sin(\theta_0 - \theta_1 - \theta_2) + l_3 \sin(\theta_0 + \theta_1 + \theta_2 + \theta_3) + l_3 \sin(\theta_0 - \theta_1 - \theta_2 - \theta_3)}{2} \\ \sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & 0 & l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (21)$$

$$x = \frac{l_1 \cos(\theta_0 + \theta_1) + l_1 \cos(\theta_0 - \theta_1) + l_2 \cos(\theta_0 + \theta_1 + \theta_2) + l_2 \cos(\theta_0 - \theta_1 - \theta_2) + l_3 \cos(\theta_0 + \theta_1 + \theta_2 + \theta_3) + l_3 \cos(\theta_0 - \theta_1 - \theta_2 - \theta_3)}{2} \quad (22)$$

$$y = \frac{l_1 \sin(\theta_0 + \theta_1) + l_1 \sin(\theta_0 - \theta_1) + l_2 \sin(\theta_0 + \theta_1 + \theta_2) + l_2 \sin(\theta_0 - \theta_1 - \theta_2) + l_3 \sin(\theta_0 + \theta_1 + \theta_2 + \theta_3) + l_3 \sin(\theta_0 - \theta_1 - \theta_2 - \theta_3)}{2} \quad (23)$$

$$z = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \quad (24)$$

l_1	100 mm
l_2	95 mm
l_3	128 mm

Tab. 3 – Tabulka s délkou ramen

5.2 Inverzní kinematika

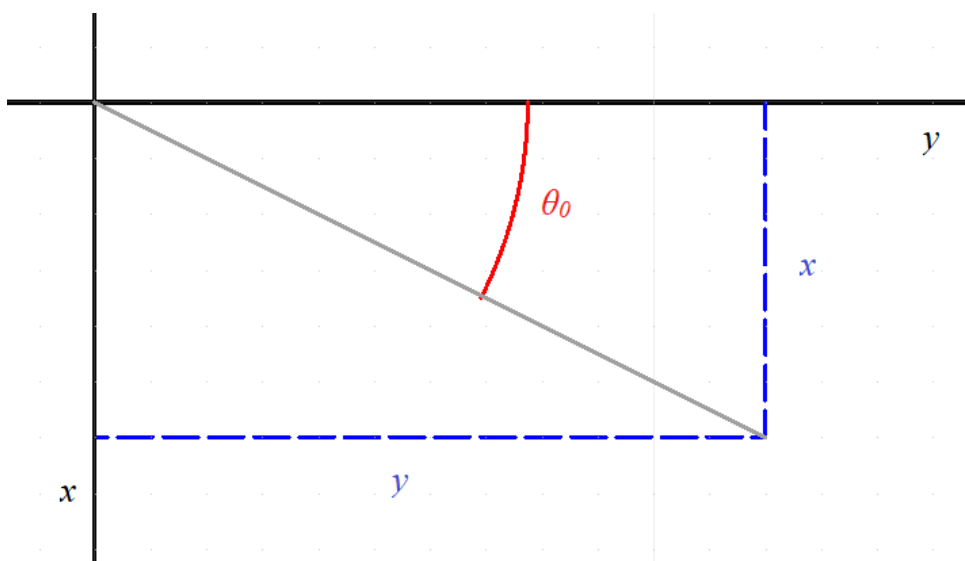
Inverzní úloha, jak již vyplývá z názvu, slouží k určení natočení jednotlivých ramen se znalostí polohy koncového bodu. Existuje několik metod, jak úlohy tohoto typu řešit. Například to lze řešit maticově, využitím goniometrických rovnic nebo jinými metodami jako je FABRIK (forward backward inverse kinematic), která se často využívá v herních enginech. Ze znalostí geometrických rozměrů, mezních úhlů natočení jednotlivých servomotorů a zadání koncového bodu lze spočítat úhly natočení jednotlivých ramen.

Řešení pomocí inverzní kinematiky je složitější než počítání přímé. Pro jednu pozici koncového bodu bývá více řešení (kromě krajních pozic). Těto skutečnosti lze využít, když vypočtená pozice je mimo konstrukční parametry ramene. Jiný výsledek může být v rámci rozsahů natočení kloubů. [20]

5.2.1 Goniometrická metoda

Často se v praxi stává, že potřebujeme se s manipulátorem dostat na požadované místo pod určitým úhlem end effectoru. Pro jednoduché sériové manipulátory se nabízí goniometrická metoda, kdy se souřadnicemi (x,y,z) koncového bodu a požadovaným natočením φ_e jsme schopni pomocí goniometrických funkcí spočítat natočení jednotlivých ramen.

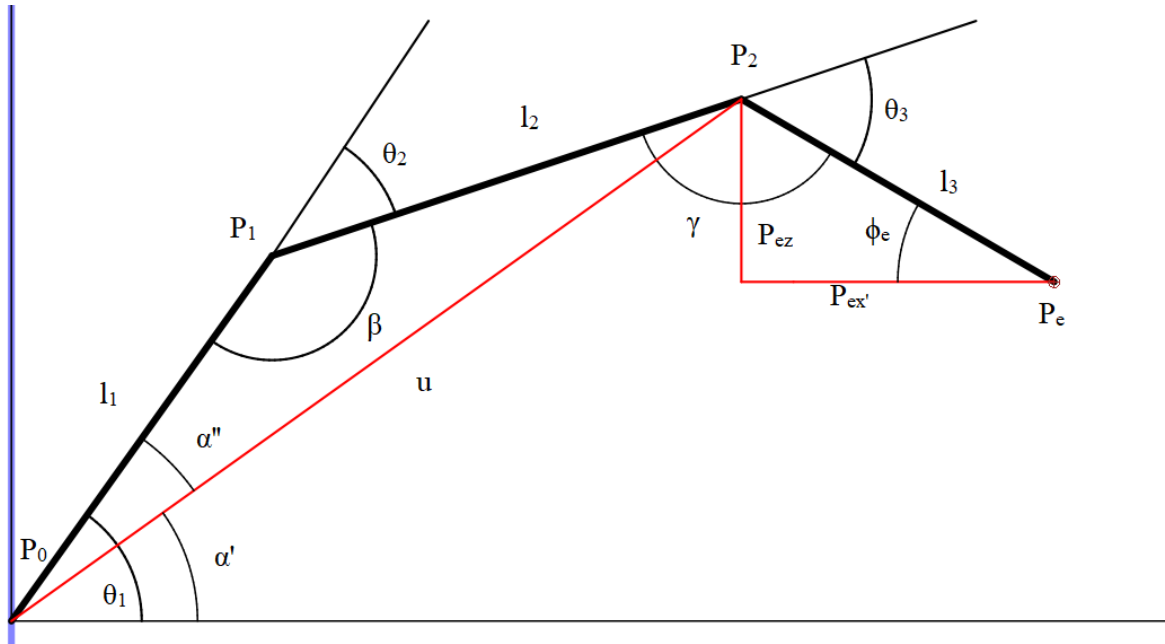
V prvním kroku si zjednodušíme úlohu do rovinného problému, Pomocí souřadnic x a y jsme schopni ihned zjistit požadovaný úhel θ_0 (25) a spočítat délku průmětu x' (26) do roviny xy .



Obr. 7 – Schéma robota shora

$$\theta_0 = \text{atan} \frac{x}{y} \quad (25)$$

$$x' = \sqrt{x^2 + y^2} \quad (26)$$



Obr. 8 – Struktura rovinného robotického ramene o třech stupních volnosti

Na Obr. 8 je znázorněn rovinný mechanismus, pro který budeme řešit úlohu inverzní kinematiky. Ze znalosti úhlu φ_e můžeme spočítat souřadnice bodu \mathbf{P}_2 .

$$l_{3x'} = l_3 \cdot \cos \varphi_e \quad (27)$$

$$l_{3z} = l_3 \cdot \sin \varphi_e \quad (28)$$

$$P_{2x'} = x' - l_{3x'} \quad (29)$$

$$P_{2z} = z - l_{3z} \quad (30)$$

Následně použitím kosinovi na trojúhelník $P_0P_1P_2$ spočítáme potřebné úhly β a α'' .

$$\beta = \arccos \frac{l_1^2 + l_2^2 + u^2}{2ul_1} \quad (31)$$

$$u = \sqrt{P_{2x'}^2 + P_{2z}^2} \quad (32)$$

$$\beta = \arccos \frac{l_1^2 + l_2^2 + P_{2x'}^2 + P_{2z}^2}{2\sqrt{P_{2x'}^2 + P_{2z}^2}l_1} \quad (33)$$

$$\alpha'' = \arccos \frac{u^2 + l_1^2 - l_2^2}{2ul_1} \quad (34)$$

$$\alpha'' = \arccos \frac{P_{2x'}^2 + P_{2z}^2 + l_1^2 - l_2^2}{2\sqrt{P_{2x'}^2 + P_{2z}^2}l_1} \quad (35)$$

Dále spočítáme úhel α' viz. Obr. 8. a následně součet α' a α'' je náš hledaný úhel α .

$$\alpha = \theta_0 = \arccos \frac{P_{2x'}^2 + P_{2z}^2 + l_1^2 - l_2^2}{2\sqrt{P_{2x'}^2 + P_{2z}^2}l_1} + \arccos \frac{u^2 + l_1^2 - l_2^2}{2ul_1} \quad (36)$$

Dále dopočítáme naznačené úhly θ_2 a θ_3 pro dopočítání posledního hledaného úhlu γ .

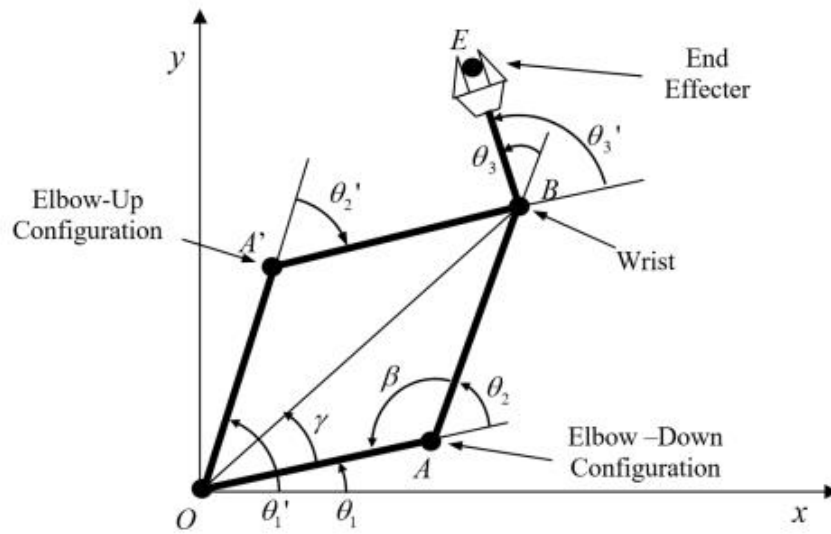
$$\theta_2 = \pi - \beta \quad (37)$$

$$\theta_3 = \varphi_e - \theta_2 - \theta_1 \quad (38)$$

$$\gamma = \pi - \theta_3 \quad (39)$$

Z rovnic (37), (38) a (39) získáme požadované úhly natočení jednotlivých kloubů, pro splnění vstupních parametrů.

Při řešení goniometrickou metodou u tohoto typu úloh z pravidla dostaneme dvě řešení (viz Obr. 9).



Obr. 9 - Obě řešení [20]

Druhé řešení lze vypočítat:

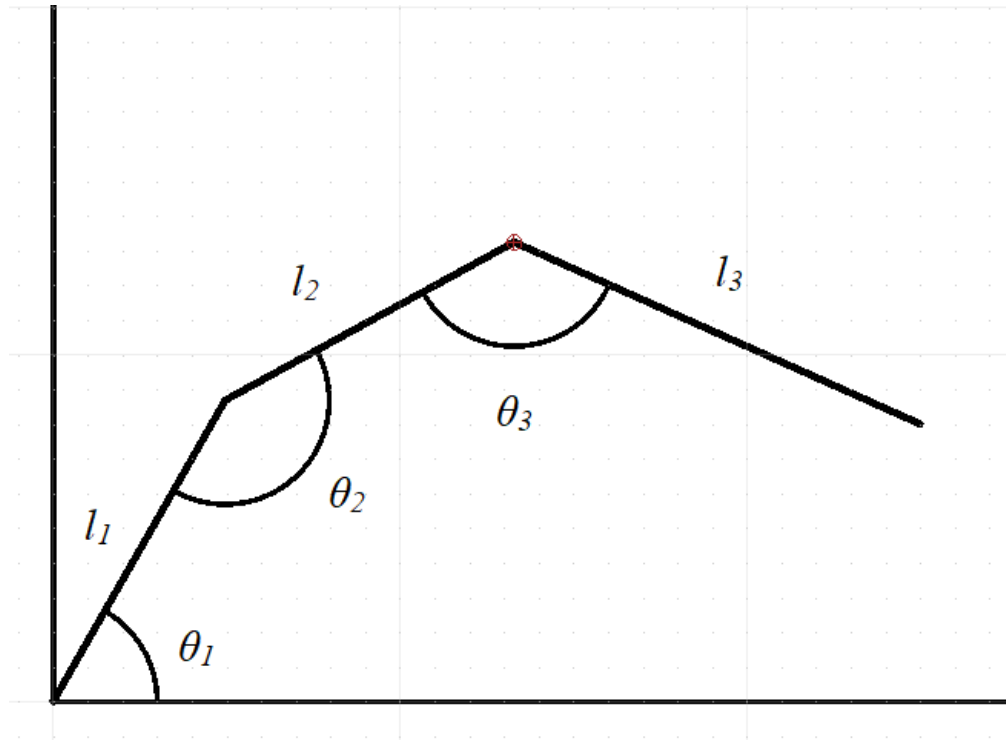
$$\theta_1' = \theta_1 + 2\gamma, \quad (40)$$

$$\theta_2' = -\theta_2, \quad (41)$$

$$\theta_3' = \phi_e - \theta_1' = \theta_3 + 2\theta_2 - 2\gamma. \quad (42)$$

Toto lze aplikovat na robota Merkur, ale nejprve musíme rozšířit úlohu do třetího rozměru.

Obrázek 9 znázorňuje rozměry reálného robota Merkur Beta 6°.



Obr. 10 - Schéma robotické ruky

Na obrázku 10 jsou vyznačeny délky jednotlivých ramen l_1 , l_2 a l_3 a úhly jednotlivých natočení kloubů. Délky podle technických specifik by měli být shodné, ale měřením bylo zjištěno, že jsou rozdílné: $l_1 = 100$ mm, $l_2 = 95$ mm, $l_3 = 128$ mm.

Rozsahy jednotlivých servomotorů, jak je vidět na obrázku, jsou zjištěné měřením, kde θ_0 je úhel rotace kolem osy z, θ_4 znázorňuje natočení end effectoru a θ_5 rozevření kleští s rozpětím od 0 mm do 45 mm:

$\theta_0 \in \langle 0^\circ, 180^\circ \rangle$, $\theta_1 \in \langle 0^\circ, 115^\circ \rangle$, $\theta_2 \in \langle 35^\circ, 215^\circ \rangle$, $\theta_3 \in \langle 63^\circ, 243^\circ \rangle$, $\theta_4 \in \langle 0^\circ, 180^\circ \rangle$, $\theta_5 \in \langle 65^\circ, 110^\circ \rangle$.

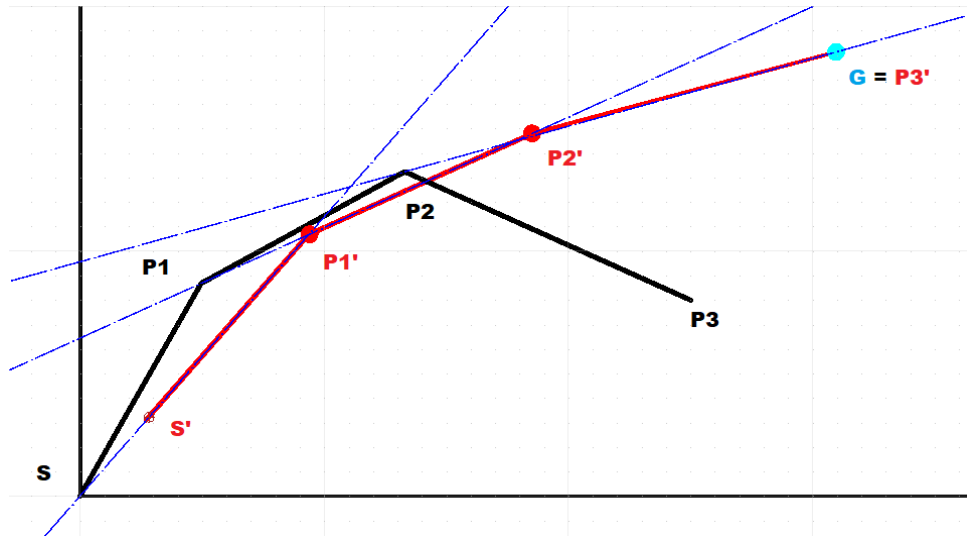
5.2.2 Metoda FABRIK

Metoda FABRIK byla poprvé publikována v roce 2011 Andreasem Aristidou a Joanem Lasenby z univerzity v Cambridge. Tato metoda je postavená na stejném principu jako metoda relaxace délky [25].

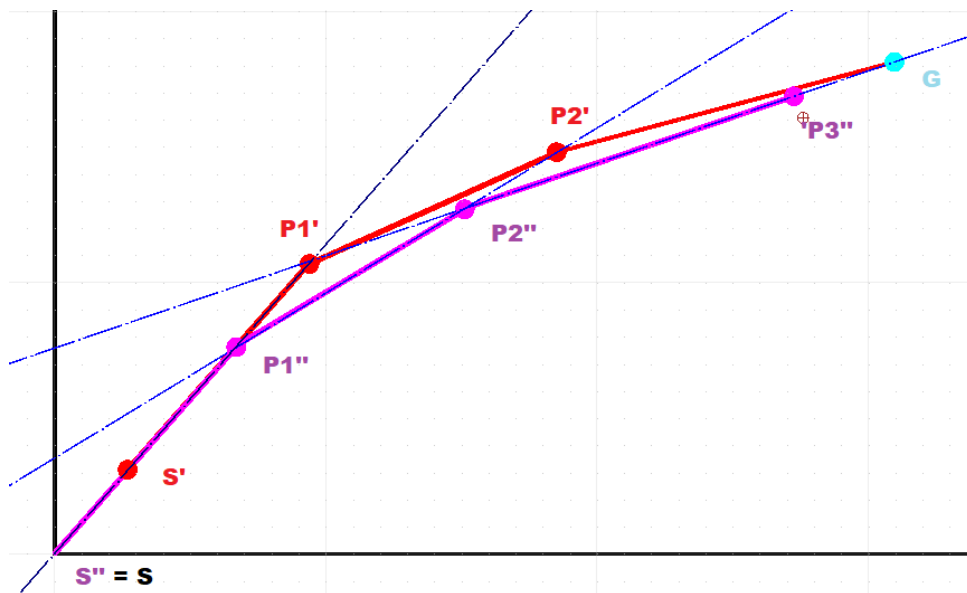
Tato metoda by se dala nazvat iterační, protože se zvyšujícím počtem opakování výpočtu se výsledek blíží skutečnosti. Vstupní parametry jsou cílová pozice, startovní pozice (úhly natočení) a délky ramen. Dále není nutné, ale z důvodu plynulosti pohybu velmi žádoucí, znát vždy předchozí pozici.

Pomocí vektorů vždy mezi požadovaným bodem kloubu nebo end effectoru a aktuální pozicí předchozího bodu (kloubu). Z tohoto získáme jednotkové vektory, které následně vynásobíme délkou ramene. Po odečtení získaného vektoru od požadované pozice získáme předchozí bod (kloub) a tímto způsobem postupujeme zprvu od end effectoru k patě robota. Zde nám vznikne pomyslná mezera mezi nově spočítaným bodem paty robota a aktuální reálnou (robot je pevně zafixovaný v jednom bodě). Dále pokračujeme stejným způsobem, ale opačným směrem, od paty k end effectoru. Proto se nazývá FABRIK, což znamená forward backward reaching inverse kinematics.

Již po jednom cyklu se nacházíme mnohem blíže k cílové pozici. Tyto početní operace opakujeme do doby, kdy odchylka mezi skutečným bodem koncovým bodem a požadovaným je dostatečně malá v závislosti na použití. Celý postup je znázorněn na Obr. 11 a Obr. 12 [22].



Obr. 11 - FABRIK od end effectoru k patě

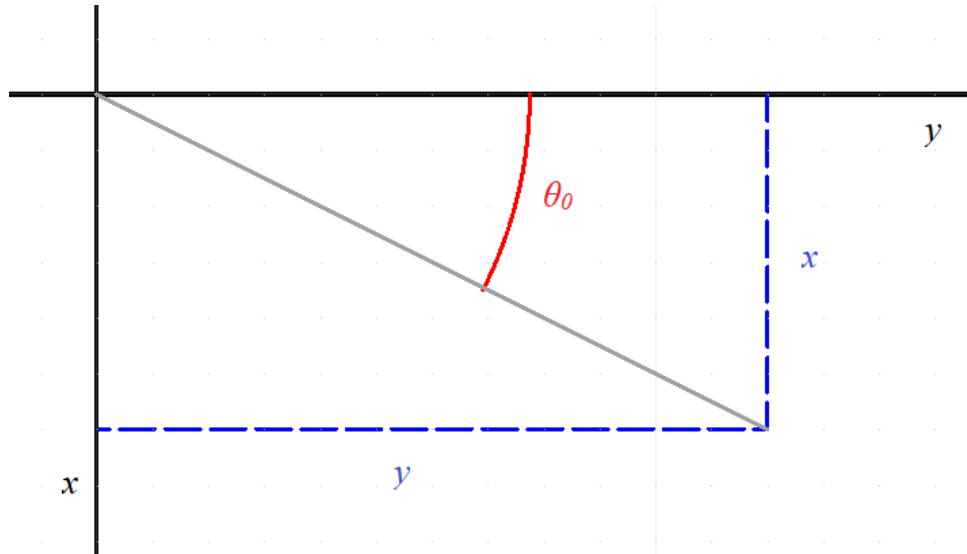


Obr. 12 - FABRIK od paty k end effectoru

Matematický popis robotické ruky Merkur pomocí metody FABRIK:

Během prvního kroku vypočteme úhel průmětu ruky do roviny xy s osou y (Obr. 13).

$$\theta_0 = \arctan \frac{x}{y} \quad (43)$$



Obr. 13 - Úhel natočení robotické ruky

Tímto výpočtem jsme úlohu zjednodušili do dvou rozměrů, proto musíme přepočítat původní souřadnice cílové pozice do nové souřadné soustavy (44), (45).

$$x_n = \sqrt{x^2 + y^2} \quad (44)$$

$$y_n = z \quad (45)$$

Teď můžeme přejít k samotné metodě FABRIK. V prvním kroku musíme získat vektor mezi body G a P₂ viz. Obr. 7. A následně vypočítat jeho směrový (jednotkový) vektor.

$$\overrightarrow{P_2G} = G - P_2 \quad (46)$$

$$\overrightarrow{P_2G_u} = \frac{\overrightarrow{P_2G}}{|\overrightarrow{P_2G}|} \quad (47)$$

Pro získání nového vektoru P₂'G (Obr. 11) získaný jednotkový vektor vynásobíme délkou příslušného ramene v tomto případě l₃.

$$\overrightarrow{P_2'G_N} = \overrightarrow{P_2G_u} \cdot l_3 \quad (48)$$

Toto je již skutečný vektor znázorňující natočení ramene 3. Následně dopočteme souřadnice bodu P₂' (Obr. 11).

$$P_2' = G - \overrightarrow{P_2'G_N} \quad (49)$$

Touto sérií výpočtů jsme zjistili pozici nového bodu P₂'. Tento postup použijeme i pro dvě zbylá ramena, až dostaneme nový bod S' (Obr. 11). Z logiky věci zde nemůže vzniknout

mezera, neboť bod S je pevně spjat se základnou. Proto následně opakujeme postup, ale opačným směrem (od bodu S k bodu G) (Obr.12).

První krok druhé části výpočtu je lehce zjednodušen, neboť vektor znázorňující rameno 1 známe z předchozího kroku. Tudiž pro získání nového bodu P1“ stačí k bodu S tento vektor přičíst (Obr.12).

$$P_1'' = S + \overrightarrow{S'P_1'} \quad (50)$$

V dalších krocích je ale nutné opět použít celou sérii výpočtů s tím rozdílem, že při získávání nového bodu budeme vektor představující příslušné rameno přičítat viz rovnice (50).

6 Matlab

Matlab je interaktivní programovací prostředí a skriptovací programovací jazyk. Tento program je podporován na operačních systémech Windows, Linux a MacOC. Autorem programu je společnost The MathWorks Inc. Název Matlab vznikl spojením anglických slov MATrix a LABoratory. [23] Původní záměr vzniku bylo usnadnění práce s knihovnamí LINEPACK a EISPACK pro práci s maticemi. [24] Tento systém je používán v mnoha oblastech, jako: vědecké a inženýrské výpočty, modelování, simulace, analýza dat včetně tvorby grafického uživatelského rozhraní. V programu existuje vlastní programovací jazyk MATLAB. Vytvořené soubory jsou ukládány jako *m-file*, nebo v předzpracované podobě *p-file*. [24]

Grafické rozhraní pracovního prostředí je rozděleno do čtyř hlavních částí: Current Folder, Editor, Command Window a Workspace. V Current Folder jsou zobrazeny soubory aktuálně načtené složky, dalo by se to přirovnat k Průzkumníku Windows. Command Window slouží k spouštění různých příkazů a zobrazení jejich výstupů. Workspace zobrazuje informace o všech uložených proměnných, či objektech. [23]

Matlab již v základní verzi obsahuje mnoho knihoven funkcí, které jsou nazývány Toolboxy. Jedním z nejužitečnějších Toolboxem je Simulink, který se prolíná i se samotným programem Matlab. Slouží k simulaci chování dynamického systému, pomocí schématického znázornění. [24]

6.1 Komunikace mezi programem a deskou Arduino Nano

Komunikace probíhá po sériovém portu. V programu Matlab lze stáhnout support package pro Arduino. V kartě HOME na vrchní liště je tlačítko Add-Ons, které po stisknutí otevře prohlížeč s podpůrnými programy pro Matlab. Zde vyhledáme *MATLAB Support Package for Arduino Hardware* a nainstalujeme.

Pro vytvoření komunikace deklarujeme proměnou *s* s příslušným portem. Číslo portu můžeme ověřit ve správci zařízení (device manager).

```
s = serial('COM45');
```

Dále nastavíme hodnotu modulační rychlosti a ujistíme se, zda je shodná s rychlostí používaného Arduina.

```
s.Baudrate=19200;
```

Další nastavení doporučuji nechat defaultní. Terminator je zde nastaven na prázdnou hodnotu, neboť by při posílání dat přidával další znak, který by znemožnil správné přečtení poslaných dat na desku Arduino. Terminátor je defaultně nastaven na LF (line feed).

```
s.StopBits=1;  
s.Terminator='';  
s.Parity='none';  
s.FlowControl='none';
```

Pro otevření datového toku se použije příkaz:

```
fopen(s);
```

Vždy musíme pamatovat na uzavření a vymazání sériové komunikace před uzavřením programu. Toto by mělo za následek zahlcení portu a nadále bychom ho nemohli použít. Může se stát, že se program přeruší v průběhu a tím pádem se neukončí komunikace a port se zahltí. Potom musíme postupovat následovně. Ve správci zařízení si nalezneme port, na kterém je připojeno Arduino. Otevřeme vlastnosti a na kartě Nastavení portu zmáčkne tlačítko Upřesnit a změním číslo portu na nové. Dále odpojíme Arduino a akci potvrdíme.

V programu ukončení a smazání komunikace provedeme příkazy:

```
fclose(s);  
delete(s);
```

Pro posílání dat použijeme funkci `fprintf(a,b)`; se dvěma vstupními parametry. Parametr *a* určuje sériovou linku po které bude data vysílat a druhý parametr *b* obsahuje samotná data, která chci odeslat.

V mém programu jsem napsal funkci `sendData`, která odesílá přepočtené úhly natočení jednotlivých servomotorů.

```
function sendData(d0,d1,d2,d3,d4,d5,s)  
for i =1:3
```

```

fprintf(s,181);
fprintf(s,d5);
fprintf(s,182);
fprintf(s,d4);
fprintf(s,183);
fprintf(s,d3);
fprintf(s,184);
fprintf(s,d2);
fprintf(s,185);
fprintf(s,d1);
fprintf(s,186);
fprintf(s,d0);
end
end

```

Vstupní parametry jsou zde co má odeslat (d0, d1,...d5) a po jaké lince (s). Cyklus se třemi opakování je zde proto, že robotovi chvíli trvá, než najde do cílové polohy a po celou tuto dobu potřebuje dostávat signál.

6.2 Metoda FABRIK v programu Matlab

Pro jednoduchost a větší přehlednost jsem vytvořil funkci `fabrikIK`, která pomocí metody FABRIK počítá úlohy inverzní kinematiky. Tato funkce má celkem jedenáct vstupních parametrů a deset výstupních.

```

function [error,d0,d1,d2,d3,d4,d5,P1N,P2N,Gn] =
fabrikIK(gx,gy,gz,rot,ef,P1N,P2N,Gn,L1,L2,L3)

```

Nejprve si všechny tyto parametry vysvětlíme. První tři vstupní parametry (g_x , g_y , g_z) jsou požadované souřadnice end effectoru. Proměnná rot určuje natočení kleští v rozsahu 0° až 180° . Pátý parametr ef udává jejich rozevření v milimetrech od 0 do 45. Další parametry $P1N$, $P2N$ a G_n jsou souřadnice bodů v prostoru, kde se nachází klouby a end effector. $L1$, $L2$ a $L3$ jsou délky jednotlivých ramen.

Po deklaraci proměnných na začátku funkce první příkazy přepočítávají na nové souřadnice v rovině natočení robotické ruky.

```
gnx = sqrt (gx^(2)+gy^(2));  
gny = gz;
```

Dále proběhne kontrola parametru natočení a rozevření end efektoru. Pro hodnoty mimo rozsah se nastaví nejbližší krajní pozice.

```
if (ef<0)  
    d5 = 0;  
elseif (ef>45)  
    d5 = 45;  
else  
    d5 = ef+65;  
end  
if (rot<0 || rot > 180)  
    d4 = 90;  
else  
    d4 = rot;  
end
```

Kontrola souřadnic, zda jsou v pracovním poli robota, je provedena podmínkou, kde ověřuji dosah ruky pomocí plochy kruhu, jehož poloměr byl zjištěn pokusy. Během nich se ověřovala nejen konstrukční dostupnost, ale i schopnost servomotorů ruku udržet.

```
if (gnx <=280 && (sqrt (gnx^(2)+gny^(2)))<=280)
```

Ověření spočitatelnosti úhlu natočení servomotoru rotujícího kolem osy z, kdy pro souřadnice x a y rovno nule požadovaného koncového existuje nekonečně mnoho rovin protínajících bod [0,0] a zároveň kolmých k rovině xy, proběhne vyřazením této možnosti a fixním nastavením úhlu θ_0 na 0. Pokud jsou hodnoty x a y různé od 0 pak proběhne výpočet úhlu θ_0 .

```
if (gx ==0 && gy == 0)  
    theta0 = 0;  
else  
    theta0 = atan(gx/gy);  
end
```

Samotné počítání inverzní kinematiky probíhá v cyklu *while*, kde na začátku ověřuji velikost odchylky skutečné koncové pozice a požadované. Přijatelná chyba výpočtu je nastavena na 1 mm.

6.2.1 Metoda FABRIK pro rovinnou úlohu

Prvním krokem v této metodě je spočítat nový vektor $\overrightarrow{P2G_N}$, ze kterého vypočteme jednotkový vektor, a ten nám bude udávat směr nového vektoru $\overrightarrow{P2G_n}$, který získáme vynásobením s délkou ramene l_3 . Pro získání nového bodu $P2_N$ odečteme tento vektor od cílového bodu G_N .

```
P2GN = GN - P2N;
%zavolání funkce na jednotkový vektor P2G
[P2Gu(1), P2Gu(2)] = unitVector(P2GN(1), P2GN(2));
%nový vektor P2'G
P2Gn = P2Gu * L3;
%nový bod P2'
P2N = GN - P2Gn;
```

Funkce `unitVector` spočítá jednotkový vektor ze zadaných hodnot.

```
function [xu, yu] = unitVector(x, y)
xu = x / (sqrt(x^2 + y^2));
yu = y / (sqrt(x^2 + y^2));
end
```

Tímto stylem pokračujeme až k bodu S, který je počátek souřadné soustavy.

```
%výpočet nové pozice ramene 2
P1P2 = P2N - P1N;
[P1P2u(1), P1P2u(2)] = unitVector(P1P2(1), P1P2(2));
P1P2n = P1P2u * L2;
%nový bod P1
P1N = P2N - P1P2n;

%výpočet nové pozice ramene 1
SP1 = P1N - S;
[SP1u(1), SP1u(2)] = unitVector(SP1(1), SP1(2));
SP1n = SP1u * L1;
%nový bod SN
SN = P1N - SP1n;
```

Následuje zpětná část úlohy, kde budu pokračovat od bodu S k cílové pozici GN.

```
P1N = S + SP1n;  
  
P1P2 = P2N - P1N;  
[P1P2u(1), P1P2u(2)] = unitVector(P1P2(1), P1P2(2));  
P1P2n = P1P2u * L2;  
P2N = P1N + P1P2n;  
  
P2GN = GN - P2N;  
[P2Gu(1), P2Gu(2)] = unitVector(P2GN(1), P2GN(2));  
P2Gn = P2Gu * L3;  
Gn = P2N + P2Gn;
```

Těmito výpočty jsme získali nové pozice kloubů a end efektoru, z nichž vypočteme úhly natočení jednotlivých ramen vůči sobě. Pro výpočty jsem zvolil funkci arkus kosinus. Obor hodnot této funkce je $\langle 0, \pi \rangle$, tudíž z prostého výpočtu nelze určit orientaci úhlů. Tuto skutečnost jsem ošetřil podmínkou, která zpětně ověřuje shodnost pozic kloubů s původními souřadnicemi. Pokud tuto podmínku vypočtený úhel nesplňuje, pak hledaný úhel je doplněk ke 2π .

```
theta1 = acos(P1N(1)/L1);  
%ověření správnosti orientace úhlu  
xi = abs(L1*sin(theta1) - P1N(2));  
if (xi>0.5 || theta1 <0.1)  
    theta1 = 2*pi - theta1;  
end  
  
theta2pom = acos((P2N(1)-P1N(1))/L2);  
theta2pompom = theta2pom;  
xi = abs(L2*sin(theta2pom) - (P2N(2)-P1N(2)));  
if (xi>0.5 || theta2pom <0.1)  
    theta2pom = 2*pi - theta2pom;  
end  
theta2 = theta2pom - theta1-pi;  
if (theta2 < 0)  
    theta2 = theta2pompom - theta1 + pi;  
end  
  
theta3pom = acos((Gn(1)-P2N(1))/L3);  
xi = abs(L3*sin(theta3pom) - (Gn(2)-P2N(2)));  
if (xi>0.5 || theta3pom <0.1)  
    theta3pom = 2*pi - theta3pom;  
end  
theta3 = theta3pom - theta2pompom-pi;
```


Tyto vypočtené hodnoty úhlů jsou teoretické a zatím nebyly ověřeny pro rozsahy servomotorů a konstrukčního provedení. Tuto skutečnost ověřuji následující soustavou podmínek. Teoreticky nekonečně mnoho řešení mi dává prostor k tomu, abych lehce upravil úhly natočení a pokusil se spočítat nové hodnoty.

```
%ověření rozsahů servomotorů a konstrukčních omezení
%upravení podmínek pro dosažení jiného řešení
if(theta1>2.0071)
    theta1 = 1.8;
elseif(theta2>3.7525)
    theta3 = 3.7525;
elseif(theta2<0.610865)
    theta3 = 0.610865;
elseif(theta3>4.2412)
    theta3 = 4.2412;
elseif(theta3<1.0996)
    theta3 = 1.0996;
    theta1 = theta1-0.349066;
end
```

Po úpravě podmínek musíme znovu přepočítat souřadnice pozic kloubů a end effectoru. A následně získat odchylku od požadované pozice. Tato hodnota je porovnávána na začátku cyklu se zvolenou přijatelnou chybou.

```
P1N = [L1*cos(theta1), L1* sin(theta1)];
P2N = [L1*cos(theta1) + L2*cos(theta2+theta1-pi), L1*
sin(theta1) + L2*sin(theta2+theta1-pi)];
p3xk = L1*cos(theta1) + L2*cos(theta2+theta1-
pi)+L3*cos(theta1+theta2+theta3);
p3yk = L1* sin(theta1) + L2*sin(theta2+theta1-
pi)+L3*sin(theta1+theta2+theta3);
P3K = [p3xk,p3yk];
P3KGN = GN - P3K;
Gn = P3K;
%výpočet chyby vzdálenosti mezi GN a Gn
e = sqrt(P3KGN(1)^(2)+P3KGN(2)^(2));
```

Proti nekonečnému cyklu je zde zavedeno opatření, které počítá počet opakování. Jestliže je překročena jistá mez, pak se cyklus přeruší i s úhly mimo rozsah.

```
if(n >20)
    break;
end
end
```

Jelikož servomotory přijímají signály od 0 do 180 musí se hodnoty úhlů nejprve převést z radiánů na stupně. Dále se vypočtené hodnoty převedou na signál, kterému bude ovládací prvek rozumět. V mém případě kolega ovzorkoval servo pohony hodnotami 0 až 180 včetně v oboru celých čísel, to znamená, že je řídí skokově s krokem o velikosti 1°. Tudíž jsem byl nucen hodnoty zaokrouhlit pomocí funkce `round` na celá čísla.

```
[a0,a1,a2,a3] =
toDegrees('radians',theta0,theta1,theta2,theta3);
%kalibrace
d1 = 115 - a1;
d2 = 215 - a2;
d3 = 243 - a3-2.7;%2,7 stupně korekce kvůli ectricitě end
effectorou
%zaokrouhlení kvůli přenosu dat
d0 = round(a0);
d1 = round(d1);
d2 = round(d2);
d3 = round(d3);
```

Dále proběhne poslední kontrola úhlů natočení skrze ověření rozmezí hodnot signálů. Při nesplnění podmínky se úhly nastaví do výchozí pozice a objeví se chybová hláška. Pro cyklické programy se proměnná *error* nastaví na 1.

6.3 Volání funkce z jiného souboru

Pro volání funkce z jiného .m souboru je nutné definovat třídu, která má shodný název s názvem souboru. Tudíž funkce vytvořené v této třídě se stanou metodami a volají se příkazem `nazevTridy.metoda(vstupní parametry)`. Dále budu mluvit o metodách jako o funkcích. Při volání funkcí s jedním či více výstupními parametry musíme tuto hodnotu či jiný objekt uložit do své nové proměnné, popřípadě matice.

Definice funkce:

```
classdef funtions
    methods (Static)

        function error
            f = msgbox('Error')
        end
    end
end
```

Zavolání funkce:

```
funtions.error;
```

Tato funkce je bez parametrů, jiné obsahující parametry jsou ukázány v samotném programu.

6.4 Vzorový program

Asi nejčastějším využitím robotické ruky v průmyslu je podavač. Vzorový program je simulace přenášení předmětu z pásu na pás.

Jako první nastavíme komunikaci přes sériovou linku, jak již bylo ukázáno v kapitole 6.1. Poté probíhá deklarace proměnných, ve kterých jsou uloženy parametry robota (délky ramen). Před samotným výpočtem musíme získat souřadnice kloubů a end efektoru základní pozice pro metodu FABRIK. Předpokládáme, že stroj je vždy natočen v rovině požadovaného bodu.

```
L1 = 100;
L2 = 95;
L3 = 128;
pi = 3.141592;
phi1 = pi/2;
phi2 = pi/2;
phi3 = pi/2;

p1x = L1* cos(phi1);
p1y = L1 * sin(phi1);

p2x = p1x + L2 *cos(phi2);
p2y = p1y + L2 *sin(phi2);

p3x = p2x + L3 *cos(phi2);
p3y = p2y + L3 * sin(phi2);
P1N = [p1x,p1y];
P2N = [p2x,p2y];
P3N = [p3x,p3y];
```

Dále začne probíhat cyklus s určeným počtem kroků, kolikrát má ruka přenést předmět z pásu na pás. Nejprve nastavíme pozici, ze které má daný předmět sebrat. Poté zavoláme funkci pro výpočet inverzní kinematiky metodou FABRIK (kapitola 6.2.1). Ta vezme jako vstupní parametry souřadnice x , y a z , dále do ní zadáme natočení end effectoru

a s rozevřenými kleštěmi. Dále funkce si uloží souřadnice kloubů a end efektoru základní pozice, ale to jen během prvního cyklu. V dalších opakování jsou zde souřadnice poslední známé pozice. Poslední tři parametry jsou délky ramen robota Merkur Beta 6°. Tato funkce vrátí několik dalších proměnných. První proměnná *error* má v sobě uloženou hodnotu 0/1, která v případě 1 cyklus ukončí. Dále vrátí úhly natočení ve formě signálu pro program, který natáčí servomotory. Poslední čtyři parametry jsou souřadnice umístění kloubů a end effektoru, které jsou využity pro další výpočty jako výchozí pozice.

Nyní máme vše potřebné k nastavení robota do cílové pozice. Pro odeslání dat použijí funkci `sendData` (kapitola 6.1). Dále proběhne ověření, zda bylo možné dosáhnout požadované pozice pomocí již zmíněné proměnné *error*. V této chvíli se robot nachází nad předmětem, který chceme přesunout. Zavoláním funkce `ocEF` ovládáme pouze rozevření či sevření kleští end efektoru.

```
function [d5] =ocEF(ef)
if (ef<0)
    d5 = 0;
elseif (ef>45)
    d5 = 45;
else
    d5 = ef+65;
end

end
```

Vyslaný signál sevře předmět a je připraven k přesunutí. Nastavíme novu cílovou pozici, která je použita jako nový parametr pro funkci `fabrikIK`. Funkce nám opět vrátí hodnoty signálu pro natočení servomotorů. Ty pošleme na Arduino, které má na starost řízení. Robot se natočí do nové pozice a pomocí funkce `ocEF` se rozevrou kleště a předmět pustí. Po každém výpočtu inverzní kinematiky musí proběhnout ověření, zda bylo dosaženo cílové pozice. Příkazy `pause(a)`; jsou zde zaimplementovány, aby mezi jednotlivými operacemi měl robot čas je vykonat.

```

for i = 1:5

x = 100;
y = 0;
z = 50;
%rozpětí kleští(endeffecoru) za zadává v mm a maximální
rozpětí je 45 mm
[error,d0,d1,d2,d3,d4,d5,P1N,P2N,P3N] =
FABRIK_J_CADA.fabrikIK(x,y,z,90,45,P1N,P2N,P3N,L1,L2,L3);
FABRIK_J_CADA.sendData(d0,d1,d2,d3,d4,d5,s);

if error == 1
    pause(5);
    break;
end
pause(2);
[d5] = FABRIK_J_CADA.ocEF(0);
FABRIK_J_CADA.sendData(d0,d1,d2,d3,d4,d5,s);

x = 200;
y = 0;
z = 50;

[error,d0,d1,d2,d3,d4,d5,P1N,P2N,P3N] =
FABRIK_J_CADA.fabrikIK(x,y,z,90,0,P1N,P2N,P3N,L1,L2,L3);
FABRIK_J_CADA.sendData(d0,d1,d2,d3,d4,d5,s);
pause(2);
[d5] = FABRIK_J_CADA.ocEF(45);
FABRIK_J_CADA.sendData(d0,d1,d2,d3,d4,d5,s);
pause(0.5);
if error == 1
    pause(5);
    break;
end

end

```

Po každé komunikaci je nutné uzavřít sériovou linku.

```

fclose(s);
delete(s);

```

7 Závěr

Cílem této bakalářské práce bylo popsat problematiku inverzní kinematiky pro stavebnici Merkur Robotická ruka Beta 6° volnosti. V textu zmiňuji dvě metody řešení. První metoda používá k výpočtu Goniometrické funkce a druhá s názvem FABRIK dosahuje výsledku pomocí vektorů a opakovaného propočítávání s vědomím drobné odchylky od přesného výsledku.

U goniometrické metody mimo souřadnic x , y a z vstupuje do rovnice parametr navíc (natočení koncového bodu), který odebírá jistý stupeň flexibility. Proto jsem se rozhodl pro další řešení použít metodu FABRIK. Sestavené rovnice jsem implementoval pomocí programovacího jazyka do prostředí Matlab. Zde proběhl samotný výpočet požadované pozice a přepočet na signály pro řídicí desku Arduino Nano. Po navázání spojení byly data odeslána a robotická ruka Merkur Beta 6° se nastavila do požadované pozice. Funkce byla nadále použita u vzorového programu simulující podavač.

Pomocí funkcí vytvořených v bakalářské práci lze sestavit program, pro širokou škálu použití robotické ruky Merkur Beta 6°. Robot ze softwarového hlediska je funkční a připraven na použití. Program lze pomocí několika úprav nastavit na jakoukoliv ruku se stejným konstrukčním provedením. Tato práce dále může být použita při výuce robotiky, či inverzní kinematiky. Výsledek práce považuji za funkční a velice uspokojivý.

Literatura

- [1] KARGER, Adolf a Marie KARGEROVÁ. Základní problémy robotiky. KARGER, Adolf a Marie KARGEROVÁ. Základy robotiky a prostorové kinematiky. Praha 6: Vydavatelství ČVUT, 2000, s. 211-213. ISBN 80-01-0283-01.
- [5] FANUC [online]. Fanuc [cit. 2019-04-17]. Dostupné z: <https://www.fanuc.eu/cz/cs>
- [6] YASKAWA [online]. Banbury, 2019 [cit. 2019-04-17]. Dostupné z: <https://www.yaskawa.co.uk/uk/>
- [7] ABB [online]. Curych, 2019 [cit. 2019-04-17]. Dostupné z: <https://new.abb.com/>
- [8] ABB [online]. Neuss, 2019 [cit. 2019-04-17]. Dostupné z: <https://robotics.kawasaki.com/en1/>
- [9] NACHI Europe GmbH [online]. 2011 [cit. 2019-04-17]. Dostupné z: <http://www.nachirobotics.eu/>
- [10] History & Milestones. NACHI-Fujikoshi [online]. Tokyo: NACHI-FUJIKOSHI, 2016 [cit. 2019-04-17]. Dostupné z: <http://www.nachi-fujikoshi.co.jp/eng/company/history.html>
- [11] Dědictví. DENSO CORPORATION [online]. 2016 [cit. 2019-04-17]. Dostupné z: <https://www.denso.com/cz/cs/about-us/heritage/1970s-1980s/>
- [12] HISTORY of the DENSO ROBOTICS. DENSO WAVE INCORPORATED [online]. DENSO WAVE INCORPORATED, 2016 [cit. 2019-04-17]. Dostupné z: <https://www.denso-wave.com/en/robot/solution/history.html>
- [13] Historie společnosti KUKA. KUKA AG [online]. Augsburg: 2019 [cit. 2019-04-17]. Dostupné z: <https://www.kuka.com/cs-cz/o-spole%C4%8Dnosti-kuka/historie>
- [14] Historie. Mitsubishi Electric [online]. Mitsubishi Electric Corporation [cit. 2019-04-17]. Dostupné z: <https://cz3a.mitsubishielectric.com/fa/cs/company/history>
- [15] A passion for innovation. Stäubli [online]. Stäubli International, 2019 [cit. 2019-04-17]. Dostupné z: <https://www.staubli.com/en/profile/company/>

- [16] Co je to Arduino. CzechDUINO.cz [online]. 2012 [cit. 2019-04-17]. Dostupné z: <http://czechduino.cz/?co-je-to-arduino,29>
- [17] Arduino. Wikipedie [online]. San Francisco: Wikimedia Foundation, 2019 [cit. 2019-04-17]. Dostupné z: <https://cs.wikipedia.org/wiki/Arduino>
- [18] ARDUINO NANO. Arduino [online]. 2019 [cit. 2019-04-17]. Dostupné z: <https://store.arduino.cc/arduino-nano>
- [19] Mechanika. Encyklopedie fyziky [online]. 2006 [cit. 2019-04-17]. Dostupné z: <http://fyzika.jreichl.com/index.php/main.article/view/149-mechanika>
- [20] Merkur perFEKT Challenge [online]. Brno, 2019 [cit. 2019-04-17]. Dostupné z: http://www.feec.vutbr.cz/merkur/podklady/2016_Kresba_ctyrlistku_pomoci_roboticke_rukypdf. Studijní materiály. VUT v Brně.
- [22] FABRIK (Inverse kinematics) In: Youtube [online]. 17.04.2019 [cit.2019-04-17]. Dostupné z: <https://www.youtube.com/watch?v=UNoX65PRehA&t=813s>. Kanál uživatele EgoMoose.
- [23] Jak pracovat s MATLABem [online]. Brno, 2014 [cit. 2019-04-17]. Dostupné z: https://www.math.muni.cz/~kolacek/vyuka/vypsyst/navod.pdf?fbclid=IwAR1-q0RAkRg7N77euyhv0irDu-plyOEMr3ylGf3cyUS5c9Q_ZCj7cX18x0g. Studijní materiály. Masarykova Univerzita.
- [24] Matlab, laboratoř nejen pro matematiky. Sdělovací technika [online]. [cit. 2019-04-17]. Dostupné z: http://cmp.felk.cvut.cz/~pisa/Public/ST_matlab.html?fbclid=IwAR24H-jL0hsasFmoxiLDnfZ2wzVV8n8a98Y_DxYi2OxCedy9JMAZABhlInk

[25] ČEJKA, Ing. Jan. Rychlé heuristické metody numerického řešení úlohy inverzní kinematiky [online]. Liberec, 2019 [cit. 2019-12-06]. Dostupné z: https://dspace.tul.cz/bitstream/handle/15240/153383/Disertace_Cejka.pdf?sequence=1&isAllowed=y. Dizertační. Technická Univerzita v Liberci.

Seznam obrázků

Obrázek 1 Sériový robot [2].....	2
Obrázek 2 Paralelní robot [3].....	3
Obrázek 3 Automatizovaná linka [4]	4
Obrázek 4 Názorné schéma kinematického modelu [24]	13
Obrázek 5 Fotografie robota Merkur	15
Obrázek 6 Schéma kinematického modelu robota merkur	16
Obrázek 7 Schéma robota shora.....	19
Obrázek 8 Struktura rovinného robotického ramene o třech stupních volnosti.....	20
Obrázek 9 Obě řešení [20]	22
Obrázek 10 Schéma robotické ruky	23
Obrázek 11 FABRIK od end effectoru k patě.....	25
Obrázek 12 FABRIK od paty k end effectoru.....	25
Obrázek 13 Úhel natočení robotické ruky	26

Zdroje obrázků

[2] MEDIAEFEKT. In: MEDIAEFEKT [online]. Brno, 2014 [cit. 2019-04-17]. Dostupné z: <http://www.mediaefekt.cz/cz/>

[3] Adept Quattro. In: MEDIAEFEKT [online]. Brno, 2014 [cit. 2019-04-17]. Dostupné z: https://automatizace.hw.cz//files/styles/full/public/story_automat/11732/adept-quattro-650h_2.jpg?itok=Z2dBzsjf

[4] Jak nejčastěji vypadá robotizované pracoviště. In: Factory automation [online]. FANUC Czech s.r.o, 2017 [cit. 2019-04-17]. Dostupné z: <https://factoryautomation.cz/jak-nejcasteji-vypada-robotizovane-pracoviste/>

[20] Merkur perFEKT Challenge [online]. Brno, 2019 [cit. 2019-04-17]. Dostupné z: http://www.feec.vutbr.cz/merkur/podklady/2016_Kresba_ctyrlistku_pomoci_roboticke_ruky.pdf. Studijní materiály. VUT v Brně.

[24] ČEJKA, Ing. Jan. Rychlé heuristické metody numerického řešení úlohy inverzní kinematiky [online]. Liberec, 2019 [cit. 2019-12-06]. Dostupné z:

https://dspace.tul.cz/bitstream/handle/15240/153383/Disertace_Cejka.pdf?sequence=1&isAllowed=y. Dizertační. Technická Univerzita v Liberci.