



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV VÝROBNÍCH STROJŮ, SYSTÉMŮ A ROBOTIKY

INSTITUTE OF PRODUCTION MACHINES, SYSTEMS AND ROBOTICS

NÁVRH SW PRO ŘÍZENÍ DELTA ROBOTU

DESIGN OF SW FOR CONTROL OF THE DELTA ROBOT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Kristýna Šimková

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jan Vetiška, Ph.D.

BRNO 2019

Zadání diplomové práce

Ústav:	Ústav výrobních strojů, systémů a robotiky
Studentka:	Bc. Kristýna Šimková
Studijní program:	Strojní inženýrství
Studijní obor:	Výrobní stroje, systémy a roboty
Vedoucí práce:	Ing. Jan Vetiška, Ph.D.
Akademický rok:	2018/19

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

Návrh SW pro řízení delta robotu

Stručná charakteristika problematiky úkolu:

Programátorská práce, která je zaměřena na vytvoření řídicího SW delta robotu na platformě PLC. V průběhu práce má student/ka získat teoretické znalosti z řídicích systémů průmyslových manipulátorů a nabyté znalosti aplikovat při vytváření SW delta robotu.

Cíle diplomové práce:

Rešerše dané problematiky.
Popis HW konfigurace.
Návrh řídicího SW.
Návrh HMI.

Seznam doporučené literatury:

MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. Řízení programovatelnými automaty. Vyd. 2. V Praze: Vydavatelství ČVUT, 2004. ISBN 80-010-2925-5.

SICILIANO, Bruno a Oussama. KHATIB. Springer handbook of robotics. Berlin: Springer, c2008. ISBN 978-3-540-23957-4.

KOLÍBAL, Z. a kol.: Roboty a robotizované výrobní technologie. VUTIUM Brno, 2016, ISBN 978-8-214-4828-5.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2018/19

V Brně, dne

L. S.

doc. Ing. Petr Blecha, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt

Tato práce se zabývá tvorbou softwaru pro delta robot v systému TwinCAT 3. V první části práce je obecně popsán delta robot, v další části je popsán hardware a programování PLC v TwinCAT 3. V závěru práce je popsána tvorba vlastního programu a HMI.

Abstract

This thesis deals with software creation for delt robot in TwinCAT 3 program. First part describes the general characteristics of a delta robot. Next part deals with hardware and PLC coding in TwinCAT 3 and the final part discusses the creation of an application.

Klíčová slova

Delta robot, sériová kinematika, kinematické transformace, PLC, TwinCAT, Beckhoff

Keywords

Delta robot, serial kinematics, kinematic transformation, PLC, TwinCAT, Beckhoff

Bibliografická citace

ŠIMKOVÁ, Kristýna. *Návrh SW pro řízení delta robotu*. Brno, 2019, 57 s. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/113098>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav výrobních strojů, systémů a robotiky. Vedoucí práce Ing. Jan Vetiška, PhD.

Poděkování

Zde bych ráda poděkovala vedoucímu mé diplomové práce panu Ing. Janu Vetiškovi, PhD. za odborné vedení diplomové práce a za čas a trpělivost, kterou mi věnoval.

Čestné prohlášení

Prohlašuji, že tato práce je mým původním dílem, zpracovala jsem ji samostatně pod vedením pana Ing. Jana Vetišky, PhD. a s použitím literatury uvedené v seznamu.

V Brně dne 24. 5. 2019

.....
Bc. Šimková Kristýna

Obsah

1	Úvod.....	15
2	Motivace	16
3	Teoretická část.....	17
3.1	Definice pojmu robot	17
3.2	Kinematické struktury robotů	18
3.3	Srovnání sériových a paralelních struktur	19
3.4	Vlastnosti delta robotu	19
3.5	Konstrukce delta robotu.....	20
3.6	Pracovní prostor delta robotu.....	21
3.7	Kinematika.....	23
3.7.1	Přímá kinematika	23
3.7.2	Inverzní kinematika	25
3.7.3	Souřadnicový systém a kinematická transformace.....	26
3.7.4	Skupiny os	27
3.7.5	Řízení pohybu.....	30
4	Praktická část	35
4.1	Hardware.....	35
4.2	Nastavení hardwaru	37
4.3	Tvorba softwaru	41
4.3.1	PLC a NC.....	41
4.3.2	TwinCAT 3	41
4.3.3	Struktura PLC projektu v TwinCAT 3	43
4.3.4	Výsledný PLC projekt	45
4.3.5	HMI	48
5	Diskuze	51
6	Závěr	53
7	Seznam použitých zdrojů.....	55
8	Seznam obrázků	57

1 Úvod

Současná doba si žádá stále častější využití robotů v mnoha různých odvětvích. Vzhledem k vysokým nárokům, především na rychlost a přesnost, jsou vyvíjeny manipulátory s novými kinematickými řetězci. Velmi často to jsou manipulátory s paralelní kinematickou strukturou, mezi které patří i delta robot, který je předmětem této diplomové práce. Tento typ manipulátorů dosahuje větší rychlosti, přesnosti a životnosti, protože díky paralelnímu systému a druhu namáhání je možné při jejich výrobě používat lehčí materiály a méně výkonné pohony. Konstrukce delta robota je typická trojúhelníkovou základnou a pohybující se platformou s pohony umístěnými na základně. Tento typ robotů se využívá především ve strojním, potravinářském a chemickém průmyslu, ale nachází uplatnění také v oblasti medicíny při provádění složitých operací.

2 Motivace

Hlavní motivací k vypracování této diplomové práce byla možnost seznámení se s PLC programováním a možnost reálné aplikace vytvořeného softwaru.

3 Teoretická část

3.1 Definice pojmu robot

Definice pojmu robot existuje v dnešní době mnoho, ale jednoznačná na sjednocení tohoto pojmu mezi odborníky dosud neexistuje. Je to způsobeno tím, že se jedná o velmi složitý technický systém, na kterém spolupracují odborníci z mnoha vědeckých odvětví. Tito odborníci poté prosazují svůj pohled na tento problém a také svou vlastní definici pojmu robot. Nicméně zde je uvedeno alespoň pár těchto definic.

Podle Websterova slovníku je robot „antropomorfní mechanická bytost postavená k rutinní manuální práci pro lidské bytosti“. Další definici uvádí Robotics Institute of America, který říká, že „robot je reprogramovatelný multifunkční manipulátor navržený pro přenášení materiálu, součástí, nástrojů, nebo specializovaných zařízení, pomocí variabilně programovaných pohybů k provádění různých úkolů“. Poslední definice, která zde bude uvedena, je podle Australian Robotics and Automation Association a ta pojem robot definuje pomocí jeho tří charakteristik:

- umožňuje nějakou formu mobility
- může být naprogramován k velmi variabilním úkolům
- po naprogramování již pracuje v automatickém režimu.

Rozdělení robotů podle různých charakteristik [1]:

Podle počtu stupňů volnosti robotu

- Univerzální robot – 6 stupňů volnosti
- Redundantní robot – více než 6 stupňů volnosti
- Deficitní robot – méně než 6 stupňů volnosti

Podle kinematické struktury

- Sériové roboty – otevřený kinematický řetězec
- Paralelní roboty – uzavřený kinematický řetězec
- Hybridní roboty – kombinace otevřeného a uzavřeného kinematického řetězce

Podle druhu pohonů

- Elektrický pohon – dnes nejčastější
- Hydraulický pohon – pro vysoké nosnosti
- Pneumatický pohon – pro vysoké rychlosti

Podle činnosti

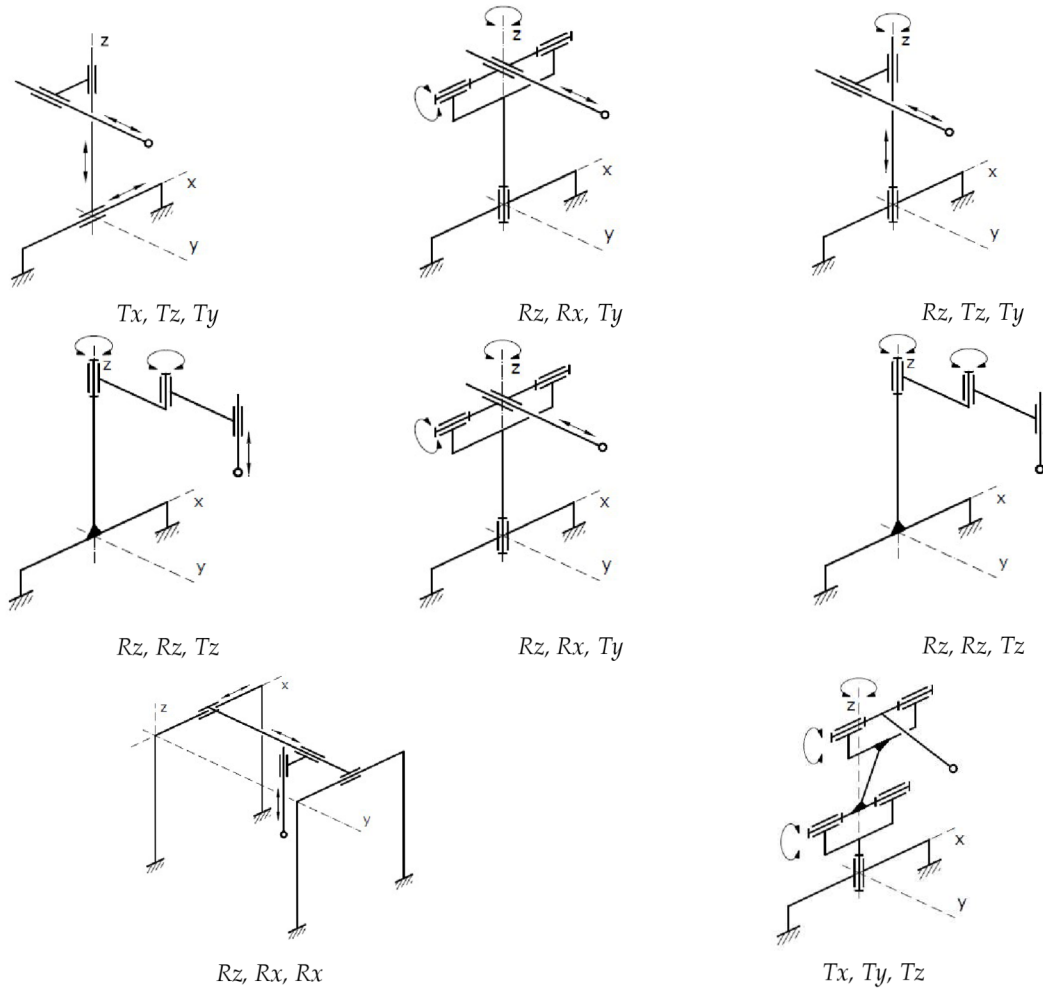
- Průmyslové roboty – ve výrobě
- Servisní roboty – ve zdravotnictví, v průmyslu, ve službách

Podle geometrie pracovního prostoru

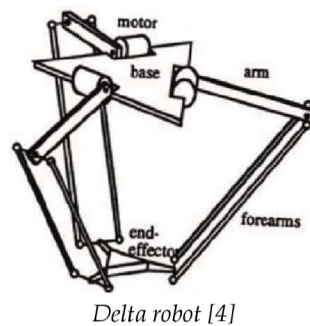
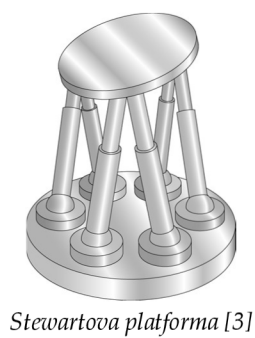
- Kartézské
- Cylindrické
- Sférické
- Angulární
- Scara

3.2 Kinematické struktury robotů

Kinematické řetězce průmyslových robotů jsou složeny pouze ze dvou druhů kinematických dvojic (vazeb), a to z vazby rotační a posuvné. Na následujících obrázcích jsou příklady kinematických řetězců složených z rotačních a z posuvných kinematických dvojic.



Obr. 3-1 – Sériové kinematické struktury [2]



Obr. 3-2 – Paralelní kinematické struktury

Úlohou kinematiky je popis chování těles a soustav těles bez ohledu na síly a momenty, které vyvolávají jakýkoliv pohyb, resp. se jedná o popis polohy částí manipulátoru a případně derivacemi je možné určit jejich rychlost a zrychlení. V případě mechanismů s paralelní kinematickou strukturou je proces popisu kinematiky poněkud obtížný oproti mechanismům, které využívají sériovou kinematickou strukturu. Modelování kinematiky paralelních mechanismů se dá řešit buď přímou, nebo inverzní metodou. [2]

3.3 Srovnání sériových a paralelních struktur

Sériový manipulátor – kinematický řetězec, který je na začátku řetězce vázaný k základně. Mezi výhody patří velký dosah a velký pracovní prostor v porovnání s celkovou velikostí stroje. Nevýhodou je pak nízká tuhost a sumace chyb jednotlivých kinematických dvojic na koncovém členu robota.

Paralelní manipulátor – kinematický řetězec, který je k základně vázaný na začátku i na konci řetězce. Je složený z báze, platformy a nejméně dvou nezávislých vodících řetězců. Vodící řetězce jsou vůči bázi a platformě řazeny paralelně. Mezi výhody paralelních kinematických struktur patří především nízká hmotnost pohybujících se částí, vysoká pracovní rychlost a zrychlení, vyšší únosnost a vysoká přesnost. Naopak mezi nevýhody řadíme vysoké nároky na řízení, náročné uložení prutů a náročný systém měření polohy pro zajištění tuhosti a přesnosti a nepříznivý poměr velikosti pracovního prostoru k celkovému objemu stroje. Cílem tohoto mechanismu je dosažení co nejvyšší tuhosti, přesnosti a rozsahu pohybu. [5]

Tato práce se zabývá delta robotem, proto se dále bude věnovat jen tomuto typu robota a jeho vlastnostem. Dále bude rozebírán jen paralelní kinematický řetězec.

3.4 Vlastnosti delta robotu

Konstrukce delta robotu byla vyvinuta profesorem Reymondem Clavelem ze švýcarské univerzity École Polytechnique Federale de Lausanne. Na počátku 80. let 20. století Prof. Clavel navrhl paralelní robot za použití rovnoběžníků, který měl tři translační a jeden rotační stupeň. Robot byl navržen pro velmi rychlou manipulaci s lehkými, malými předměty. V devadesátých letech začaly tento typ robota nabízet firmy jako např. ABB nebo Systems Sigpack, nicméně původní koncepce nebyla překonaná a delta roboty dnes patří mezi nejúspěšnější stroje s paralelní kinematickou strukturou. Mezi modifikace patří například dvojité delta robot, STAR modifikace, RPS modifikace atd.

Delta robot je paralelní robot se třemi stupni volnosti. Umožňuje tedy přímočarý pohyb koncového efektoru ve směru os x , y a z . Kinematický řetězec delta robotu je složený v pořadí R2S2S – rotační vazba, dvakrát paralelně sférický kloub, dvakrát paralelně sférický kloub. Tento typ robota je vhodný na rychlou manipulaci s malými a lehkými předměty. Skládá se ze dvou platform, třech ramen, šesti vzpěr a efektoru. Pohony jsou montovány na základnu, což má za následek velmi vysoké zrychlení koncového efektoru. Pohony jsou umístěny na ramenech, která jsou vůči sobě pootočena o 120° . Ramena jsou na konci propojena se vzpěrami pomocí sférických kloubů. Klouby jsou uspořádány paralelně a zároveň uchycují pohybující se platformu. Takové uspořádání vzpěr umožňuje přímočarý pohyb koncového efektoru. [5]

Na následujících obrázcích jsou uvedeny příklady komerčně vyráběných delta robotů.



3-osý delta robot M-3iA/12H od firmy Fanuc [6]



4-osý delta robot M-3iA/6S od firmy Fanuc [6]



6-osý delta robot M-3iA/6A od firmy Fanuc [6]



4-osý delta robot IRB 360 FlexPicker od firmy ABB [7]

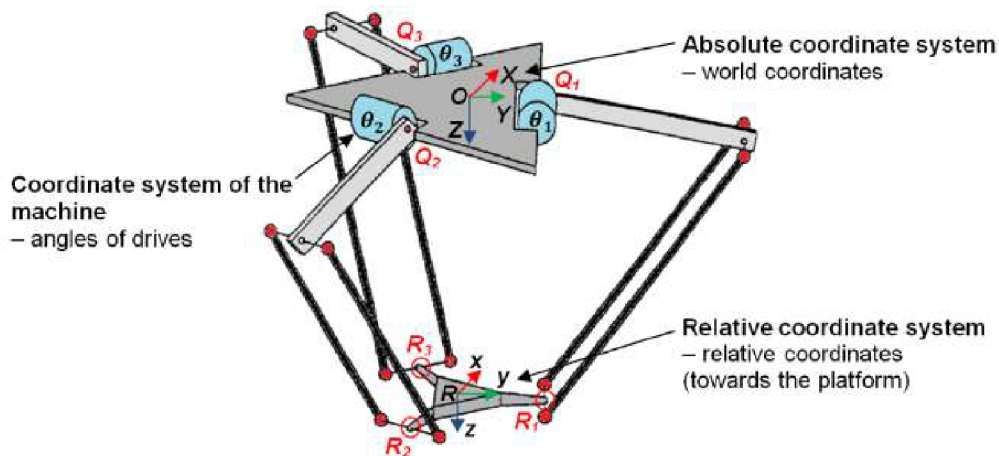


4-osý delta robot X-Delta 3+1 od firmy Omron [8]

Obr. 3-3 – Komerčně vyráběné delta roboty

3.5 Konstrukce delta robotu

Vlastnosti a možnosti manipulátoru jsou závislé na typu a vhodné volbě stavebních prvků. Výběr materiálů a prvků pro sestavení je v dnešní době široký. [10]



Obr. 3-4 - Konstrukce delta robotu a souřadnicové systémy [9]

Základna tvoří statickou část robotu a nemá přesně stanovený tvar. Od základny se odvíjí další části mechanismu. Její velikost nepřímo ovlivňuje velikost pracovního prostoru. Obvykle bývají na základnu připevněné pohonné a řídicí prvky. U základny je vyžadována vysoká tuhost, protože při nedostatečné tuhosti může dojít k rozkmitání celého mechanismu.

Ramena společně s klouby tvoří základní část vodících řetězců. Jejich úkolem je přenos pohybu ze základny směrem k platformě. Vhodným uspořádáním ramen spojených pomocí kloubů můžeme dosáhnout vysoké tuhosti, která ovlivňuje přesnost a možnost dosažení vysokých rychlostí manipulátoru. Ramena jsou buď s proměnlivou, nebo stabilní délkou.

Klouby mají přenášet síly a pohyb mezi základnou a menší platformou. Přesnost manipulátoru je závislá na kvalitě použitých kloubů. Klouby jsou namáhané a podléhají opotřebení v průběhu času. Velký vliv na přesnost má hmotnost tělesa, se kterým se manipuluje. Příliš velká hmotnost způsobuje velké deformace a tím dochází k nepřesnému pohybu. U delta robotu jsou používány kulové klouby, jejichž nevýhodou je možné vyklouznutí kloubu z kulové jamky. Pro zabránění vyklouznutí z jamky se používají přitlačné pružiny.

Úkolem platformy je upevnění koncového efektoru. Platforma je poslední pohyblivá část kinematického řetězce. Platforma se u paralelních mechanismů může nacházet nad základnou případně pod základnou. Platformy by měly být vyrobené z lehkých a pevných materiálů, čímž se dosáhne snížení setrvačných sil a momentů.

Efektor slouží k přenášení předmětů případně k přidržení předmětů během obrábění. Efektor může předměty uchopovat mechanicky, magneticky nebo podtlakem.

3.6 Pracovní prostor delta robotu

Pracovní prostor je prostor, ve kterém se robot pohybuje. U delta robotu dochází ke dvěma typům singularity. Okrajová singularita se vyskytuje v krajních polohách, pokud je manipulátor vytažený nebo zatažený. Tuhle singularitu je možné vyloučit tak, že se robot neřídí na hranici dosažitelného pracovního prostoru. Druhým typem jsou singularities vnitřní, které se vyskytují uvnitř pracovního prostoru. Bývají běžně způsobeny vyrovnaním dvou nebo více os pohybu robotu. Vzhledem k tomu, že tyto singularities se mohou vyskytnout kdekoli v pracovním prostoru, představují velké omezení a je třeba je řešit. [11]

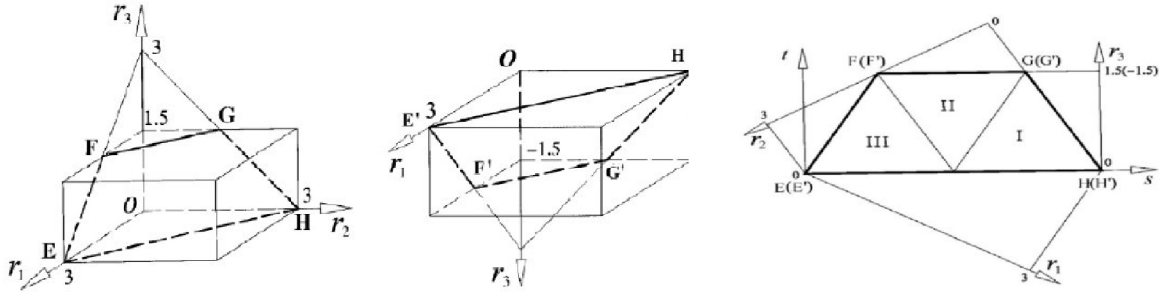
Pracovní prostor je možné spočítat dvěma způsoby, a to za pomoci přímé nebo inverzní kinematiky. Délky ramen L_1 a L_2 a parametry platformy R a r vymezují pracovní prostor. Při výpočtu pomocí přímé kinematiky je nutné znát minimum a maximum úhlu θ (např. $\theta = 0^\circ - 90^\circ$). Při výpočtu pracovního prostoru metodou inverzní kinematiky je nutné znát krajní polohy pohybující se platformy.

Kde R je skutečná vzdálenost středu základny k bodu A , r je skutečná vzdálenost bodu C od středu pohybující platformy a L_1 a L_2 jsou délky ramen. Bod A je uchycení ramene robotu k základně, bod B je kloub mezi rameny robotu a bod C je uchycení ramene robotu k pohybující se platformě (Obr. 3-7).

Vypočítáme posun bodů B_i' , A_i' a C_i' . Polohy bodů C_i' a P se nyní shodují. Je důležité, aby prostorový nástroj zahrnoval všechny parametry delta robotu, protože parametry L_1 , L_2 a $|R_1|$ se mohou měnit mezi hodnotami 0 a ∞ . Hledání optimálních robotických parametrů je založeno na následujících výpočtech:

$$D = \frac{L_1 + L_2 + |R_1|}{3} \quad (1)$$

Délky částí ramen L_1 , L_2 a $|R_1|$ ku D jsou definovány jako tři nerozměrové parametry, které jsou:



Obr. 3-5 – Řešení prostoru delta robotu [11]

$$r_1 = \frac{L_1}{D} ; r_2 = \frac{L_2}{D} ; r_3 = \frac{|R_1|}{D} \quad (2)$$

z toho plyne:

$$r_1 + r_2 + |r_3| = 3 \quad (3)$$

Tento postup snižuje trojrozměrné parametry prostoru na dvourozměrné. Z obrázku 10 vyplývá, že neprostorový delta robot není možné zkonstruovat, pokud $|r_3| > 1,5$. Proto platí následující podmínky:

$$0 < |r_3| < 1,5; \quad 0 < r_1; \quad r_2 < 3 \quad (4)$$

Na Obr. 3-5 je znázorněno řešení na fyzikálním modelu prostoru delta robotu. Model se sestává se dvou lichoběžníků $EFGH$ a $E'F'G'H'$, kde případné kombinace délek ramen jsou označeny hodnotami r_1 , r_2 a $|r_3|$. Řešení prostoru umožňuje snížení trojrozměrného problému na dvourozměrný. Jsou zde dvě ortogonální polohy s a t použité v rovinné soustavě souřadnic $O-st$ vyjádřené pomocí r_1 , r_2 a $|r_3|$.

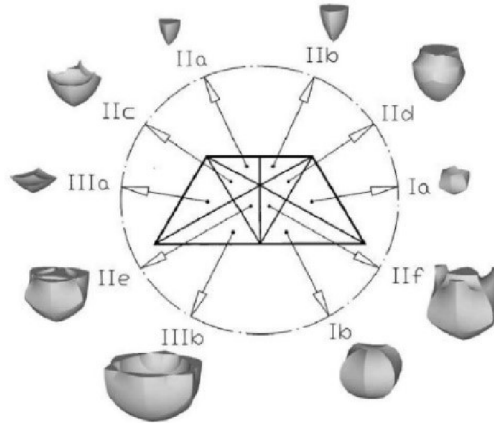
$$s = \frac{2}{\sqrt{3}} * r_1 + \frac{1}{\sqrt{3}} * |r_3| \quad (5)$$

$$t = |r_3| \quad (6)$$

$$r_1 = \frac{\sqrt{3}s - t}{2} \quad (7)$$

$$|r_3| = t \quad (8)$$

Poloha s a t může být přeměněna na r_1 , r_2 a $|r_3|$. Tyto rovnice se využívají pro tvorbu výkonnostních grafů. Tvary pracovního prostoru robotu lze rozdělit do tří základních částí, které se dělí do dalších deseti tvarů, jak je znázorněno na Obr. 3-6.



Obr. 3-6 – Tvary pracovního prostoru delta robotu [11]

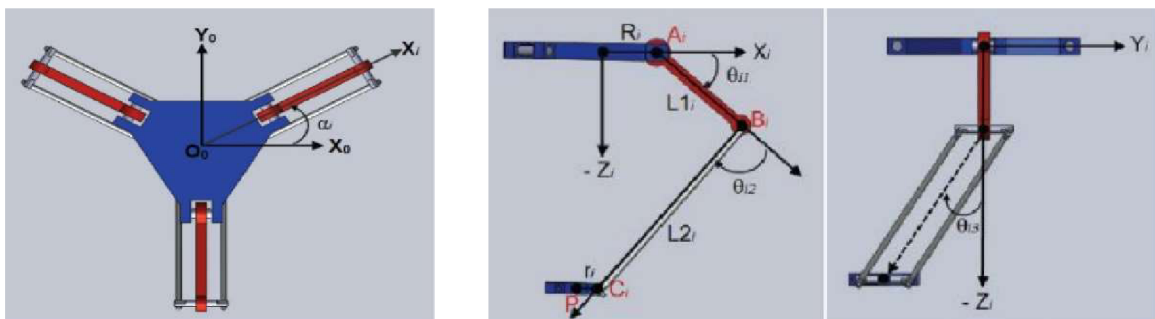
Délka a počet ramen ovlivňují velikost pracovního prostoru robotu. Při změně počtu a délek ramen, musí být vždy splněna podmínka fyzické realizovatelnosti. Prodloužení ramen vede k delšímu dosahu manipulátoru, což však nemusí znamenat i zvětšení pracovního prostoru.

3.7 Kinematika

Existují dvě základní kinematické úlohy – přímá a inverzní. Při řešení přímé kinematické úlohy jsou známé úhly natočení ramen robotu a řešením jsou souřadnice koncového bodu. Při řešení inverzní kinematické úlohy jsou známy souřadnice koncového bodu a řešením jsou natočení jednotlivých ramen robotu.

3.7.1 Přímá kinematika

Přímá kinematika delta robotu je definována rovnicí (9) obsahující neznámé polohy bodu $P = [X_p \ Y_p \ Z_p]$ pro daný soubor úhlů θ_i , pro $i = 1, 2, 3$. Řešením soustavy rovnic může být bod, který je průsečíkem tří kulových ploch. Obecně platí, že existují dvě možná řešení této soustavy rovnic, z čehož plyne, že pro skupinu ovládacích úhlů má mobilní platforma dvě konfigurace s ohledem na polohu základní podstavy. Z toho vyplývá, že pohybující platforma může být v ose +Z nebo -Z. Kde X_p , Y_p a Z_p jsou souřadnice pohybující se platformy. [12]



Obr. 3-7 – Řešení úlohy přímé kinematiky [12]

$$(x_p - x_i)^2 + (y_p - y_i)^2 + (z_p - z_i)^2 = L_2^2 \quad (9)$$

kde X_p, Y_p a Z_p jsou souřadnice vektoru OP a:

$$\begin{aligned}x_i &= (R + L_1 \cos \theta_{i1} - r) * \cos \alpha_i \\y_i &= (R + L_1 \cos \theta_{i1} - r) * \sin \alpha_i \\z_i &= -L * \sin \theta_{i1}\end{aligned}\quad (10)$$

Rovnice (9) odpovídá rovnic pro kouli o poloměru L_2 pro každou paži. Pomocí rovnice (11) vypočteme bod $B1', B2', B3'$:

$$B'_i = \cos(\alpha_i) * (L_{1i} * \cos(\theta_i) + R_1) - (R_1 + L_{1i} * \cos(\theta_i)) * \sin(\alpha_i) - L_{1i} * \sin(\theta_i)\quad (11)$$

kde $R_1 = R - r_i$ a α označuje natočení osy X_i od skutečné osy X_0 . Zároveň $\alpha_1, \alpha_2, \alpha_3 = [0^\circ, 120^\circ, 240^\circ]$. Nyní lze spočítat průsečík tří kružnic se středem v bodech $B1', B2', B3'$.

$$\begin{aligned}(x_p - x_1)^2 + (y_p - y_1)^2 + (z_p - z_1)^2 &= L_2^2 \\(x_p - x_2)^2 + (y_p - y_2)^2 + (z_p - z_2)^2 &= L_2^2 \\(x_p - x_3)^2 + (y_p - y_3)^2 + (z_p - z_3)^2 &= L_2^2\end{aligned}\quad (12)$$

Rovnice můžeme rozepsat:

$$\begin{aligned}x_p^2 + y_p^2 + z_p^2 - 2x_p x_1 - 2y_p y_1 - 2z_p z_1 &= -x_1^2 - y_1^2 - z_1^2 - L_2^2 \\x_p^2 + y_p^2 + z_p^2 - 2x_p x_2 - 2y_p y_2 - 2z_p z_2 &= -x_2^2 - y_2^2 - z_2^2 - L_2^2 \\x_p^2 + y_p^2 + z_p^2 - 0 - 2y_p y_3 - 2z_p z_3 &= -x_3^2 - y_3^2 - z_3^2 - L_2^2\end{aligned}\quad (13)$$

Pro zjednodušení si pravou stranu rovnic přepíšeme do tvaru:

$$\begin{aligned}w_1 &= -x_1^2 - y_1^2 - z_1^2 - L_2^2 \\w_2 &= -x_2^2 - y_2^2 - z_2^2 - L_2^2 \\w_3 &= -y_3^2 - z_3^2 - L_2^2\end{aligned}\quad (14)$$

Dostaneme tvar rovnic:

$$\begin{aligned}x_p^2 + y_p^2 + z_p^2 - 2x_p x_1 - 2y_p y_1 - 2z_p z_1 &= w_1 \\x_p^2 + y_p^2 + z_p^2 - 2x_p x_2 - 2y_p y_2 - 2z_p z_2 &= w_2 \\x_p^2 + y_p^2 + z_p^2 - 0 - 2y_p y_3 - 2z_p z_3 &= w_3\end{aligned}\quad (15)$$

Rovnice od sebe odečteme:

$$x_p(x_1 - x_2) - y_p(y_1 - y_2) - z_p(z_1 - z_2) = \frac{w_1 - w_2}{2}\quad (16)$$

$$x_p x_1 - y_p(y_1 - y_3) - z_p(z_1 - z_3) = \frac{w_1 - w_3}{2}\quad (17)$$

$$x_p x_2 - y_p(y_2 - y_3) - z_p(z_2 - z_3) = \frac{w_2 - w_3}{2}\quad (18)$$

Nyní od sebe odečteme rovnice (17) a (18), vyjádříme x_p a y_p a dosadíme do rovnice:

$$x_p^2 + y_p^2 + z_p^2 - 0 - 2y_p y_3 - 2z_p z_3 = -x_3^2 - y_3^2 - z_3^2 - L_2^2\quad (19)$$

3.7.2 Inverzní kinematika

Inverzní kinematika je definována řešením rovnice (9) pro neznámé θ_i vzhledem k poloze pohybující platformy $P = [X_p \ Y_p \ Z_p]$. Řešení nalezneme také pomocí průsečíků jednotlivých kružnic k_1 a k_2 . [12]



Obr. 3-8 – Řešení úlohy inverzní kinematiky [13, 14]

Spočítáme průsečík kružnic k_1 a k_2 , tedy souřadnice bodu B_i :

$$\begin{aligned} (B_x - A_x)^2 + (B_y - A_y)^2 + (B_z - A_z)^2 &= L_2^2 \\ (B_x - C'_x)^2 + (B_y - C'_y)^2 + (B_z - C'_z)^2 &= L_2^2 \end{aligned} \quad (20)$$

Výpočet si usnadníme promítnutím do roviny YZ:

$$\begin{aligned} (B_y - A_y)^2 + (B_z - A_z)^2 &= L_2^2 \\ (B_y - C'_y)^2 + (B_z - C'_z)^2 &= L_2^2 - X_p^2 \end{aligned} \quad (21)$$

Vzhledem k pohybu ramene L_1 pouze v rovině YZ má rovnice tvar:

$$(B_y - A_y)^2 + (B_z - A_z)^2 = L_2^2 \quad (22)$$

Pohyblivá platforma se však pohybuje po osách XYZ, proto najdeme bod C' , který je posunut o vzdálenost X_p a tím ji promítneme do roviny YZ. Rovnice má poté tvar:

$$(B_y - Y_p + r)^2 + (B_z - C'_z)^2 = L_2^2 - X_p^2 \quad (23)$$

Pro správné vyjádření bodu B má soustava rovnic tvar:

$$\begin{aligned} (B_y - R)^2 + B_z^2 &= L_2^2 \\ (B_y - Y_p + r)^2 + (B_z - C'_z)^2 &= L_2^2 - X_p^2 \end{aligned} \quad (24)$$

Úhel θ_i spočítáme pomocí nalezeného průsečíku v bodě B, proto můžeme počítat:

$$\arctan \theta = \frac{B_z}{\sqrt{A_y - B_y}} \quad (25)$$

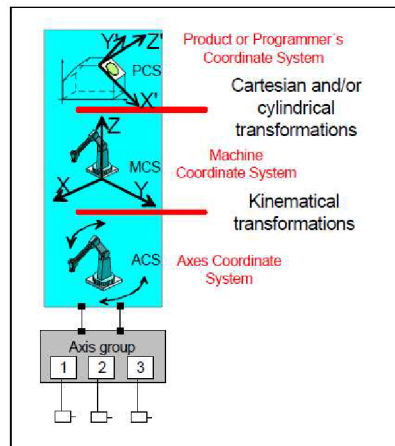
Průsečíky jsou počítány pro každé rameno, proto musíme taky počítat s pootočením bodu P o $+120^\circ$ a -120° :

$$\begin{aligned} X_p &= X_p * \cos 120 + Y_p * \sin 120 \\ Y_p &= -X_p * \sin 120 + Y_p * \cos 120 \end{aligned} \quad (26)$$

3.7.3 Souřadnicový systém a kinematická transformace

V této kapitole bude popsána souvislost mezi souřadnicovými systémy a kinematickou transformací.

Podstatou trajektorie je koordinovaný pohyb dvou nebo více os od počátečního bodu k cílovému bodu přes definovanou cestu se stanovenou rychlostí cesty. Jako cestu lze považovat přímku, kruhový pohyb nebo pohyb po křivce definovaný funkcí. Definice informací o cestě nebo jakékoli pozici v prostoru vyžaduje souřadnicový systém. V rámci této specifikace jsou definovány tři souřadnicové systémy:



Obr. 3-9 – Souřadnicové systémy [16]

ACS (Axis Coordinate System = Souřadnicový systém osy) – aktuální poloha fyzické osy (po najetí do referenční polohy).

MCS (Machine Coordinate System = Souřadnicový systém stroje) - kartézský souřadnicový systém s referenční polohou je pevně dán vzhledem ke stroji. (Někdy je nazvaný jako "Světový souřadnicový systém" nebo "Základní souřadnicový systém"). Souřadnicový systém ACS složený z více fyzických os je propojen se souřadnicovým systémem MCS prostřednictvím kinematické transformace.

PCS (Product or Programmer's Coordinate System = Souřadnicový systém produktu nebo programový souřadnicový systém) - skutečný obrobek může být pootočen nebo posunut vzhledem k souřadnicovému systému MCS nebo dokonce může být vzhledem k tomuto souřadnicovému systému pohyblivý. Ve většině případů je potřeba popisovat jeho trajektorii nezávisle na situaci stroje. Pro převod mezi těmito dvěma souřadnicovými systémy (MCS na PCS a naopak) se běžně provádí kartézská nebo cylindrická transformace.

Pro určení bodu nebo orientace v prostoru musí být pozice vždy spojena se souřadnicovým systémem. Pomocí transformací lze tuto pozici transformovat na jiné souřadnicové systémy. V rámci této specifikace jsou pro tyto transformace definovány funkční bloky, které skryjí složitost těchto transformací programátorovi při jeho každodenním použití. Všechny příkazy pro pohyb více os se vztahují vždy pouze k jednomu ze souřadnicových systémů.

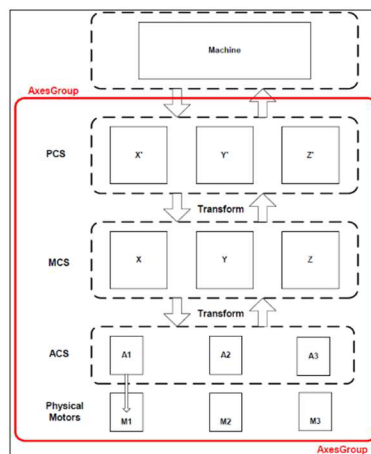
Fyzické osy jsou propojeny prostřednictvím mechanických vazeb, které zajišťují pohyb koncového efektoru (TCP = Tool Center Point) v prostoru. Fyzická sestava os, a tudíž i poloha TCP v souřadnicovém systému MCS, je popsána tzv. kinematickou transformací. Kinematická

transformace spojuje souřadnicové systémy ACS a MCS (přímá transformace). Použitím kinematické transformace pro pozici související s ACS lze tuto pozici transformovat do pozice v MCS. Na druhou stranu, při použití inverzní kinematické transformace, pozice související s MCS může být transformována na pozici v ACS (inverzní transformace).

Pokud chceme, aby efektor stroje sledoval pohyblivý cíl (např. obrobek na pohyblivém pásu), můžeme toho snadno dosáhnout pomocí dynamické transformace. Pokud je dynamická transformace aktivní, může osa, jejíž poloha je definovaná v MCS, sledovat tuto dynamickou transformaci a pohybovat se zároveň s pohyblivým cílem definovaným v PCS, nebo může zůstat v klidu v MCS nebo ACS.

3.7.4 Skupiny os

Související osy seskupujeme do skupin pro jejich přehlednější a efektivnější řízení. Příklad vztahu mezi různými úrovněmi os a skupin je zobrazen na obrázku níže.



Obr. 3-10 – Skupiny os [16]

Skupina os zobrazena červeně poskytuje rozhraní pro uživatele. Lze přistupovat buď zvlášť ke každému souřadnicovému systému (ACS, MCS, PCS) nebo k celé skupině jako k jednomu celku.

V následujících odstavcích jsou popsány jednotlivé funkční bloky používané v souvislosti se skupinou os.

MC_AddAxisToGroup – Tento funkční blok přidá jednu osu do skupiny os. Jedná se o administrativní funkční blok, protože není generován žádný pohyb.

MC_RemoveAxisFromGroup – Tento funkční blok odstraní jednu osu ze skupiny os. Jedná se o administrativní funkční blok, protože není generován žádný pohyb. Pokud ve skupině os nezůstane žádná osa, změní se stav skupiny na hodnotu `GroupDisabled`.

MC_UngroupAllAxes – Tento funkční blok odstraňuje všechny osy ze skupiny os. Jedná se o administrativní funkční blok, protože není generován žádný pohyb. Po dokončení je stav skupiny změněn na `GroupDisabled`.

MC_GroupEnable – Tento funkční blok změní stav skupiny z `GroupDisabled` na `GroupStandby`. Jedná se o administrativní funkční blok, protože není generován žádný pohyb.

MC_GroupDisable – Tento funkční blok změní stav skupiny na `GroupDisabled`. Jedná se o administrativní funkční blok, protože není generován žádný pohyb.

MC_SetKinTransform – Tento funkční blok nastavuje kinematickou transformaci mezi ACS a MCS na základě předem definovaného kinematického modelu. Kinematická transformace je reprezentací konstrukce stroje. U kinematicky jednoduchých strojních konstrukcí, jako u kartežského robota se třemi osami nemusí být kinematická transformace nutná. Vstup KinTransform se týká kinematického modelu včetně parametrů. Tento funkční blok vždy pracuje na předem definované skupině os. Vzhledem k tomu, že kinematická transformace musí vždy odpovídat příslušné skupině os, bude volání tohoto funkčního bloku vést k chybě, pokud nebude definována příslušná skupina os.

MC_GroupSetPosition – Tento funkční blok nastavuje polohu všech os ve skupině os. Nové souřadnice jsou zapsány v poli. Se zadáním souřadnicového systému je zvolen příslušný souřadný systém. MC_GroupSetPosition ovlivňuje souřadnicové systémy vyšší úrovně (takže pokud je zvolen ACS, ovlivní se MCS a PCS).

MC_GroupReadActualPosition – Tento funkční blok vrátí aktuální polohu skupiny os ve zvoleném souřadnicovém systému. Jedná se o administrativní funkční blok, protože není generován žádný pohyb.

MC_GroupReadActualVelocity – Tento funkční blok vrátí aktuální rychlost skupiny os ve vybraném souřadnicovém systému. Jedná se o administrativní funkční blok, protože není generován žádný pohyb.

MC_GroupReadActualAcceleration – Tento funkční blok vrátí aktuální zrychlení skupiny os ve vybraném souřadnicovém systému. Jedná se o administrativní funkční blok, protože není generován žádný pohyb.

MC_GroupStop – Tento funkční blok provede řízené zastavení skupiny os a převede ji do stavu „GroupStopping“. Přeruší veškeré probíhající aktivity tohoto funkčního bloku. Pokud je skupina os ve stavu „GroupStopping“, žádný jiný funkční blok nemůže provádět pohyb ve stejné skupině os. Po dosažení nulové rychlosti skupiny os je výstup Done okamžitě nastaven na hodnotu TRUE. Skupina os, zůstává ve stavu "GroupStopping", pokud je Execute stále TRUE nebo pokud není dosud dosaženo nulové rychlosti. Jakmile je "Done" ve stavu TRUE a "Execute" je FALSE, skupina os se dostane do stavu "GroupStandBy". Příkaz lze přerušit pouze příkazem MC_GroupDisable.

MC_GroupHalt – Tento funkční blok způsobí řízené zastavení pohybu. Přeruší jakoukoli aktivitu jiných funkčních bloků. Skupina os je ve stavu "GroupMoving", dokud není její rychlost nula. Po dosažení nulové rychlosti se výstup DONE, dostane do stavu TRUE a skupina os do stavu GroupStandby. MC_GroupHalt se používá pro zastavení skupiny os v běžných provozních podmínkách. V režimu bez vyrovnávací paměti lze při zpomalení skupiny os nastavit další pohybový příkaz, který zruší MC_GroupHalt a bude okamžitě proveden. Při provádění funkčního bloku MC_GroupHalt zůstávají příslušné osy na stejné cestě, která by byla provedena bez vydání MC_GroupHalt.

MC_GroupInterrupt – Tento funkční blok přeruší probíhající pohyb a zastaví pohyb skupiny os. Tento funkční blok je spojen s funkčním blokem MC_GroupContinue. Aktivací MC_GroupContinue se program přenesou zpět do situace při aktivaci MC_GroupInterrupt.

MC_GroupContinue – Tento funkční blok přenesou program zpět do situace při vydání příkazu MC_GroupInterrupt. Na původně přerušeném funkčním bloku nastaví výstup ACTIVE do stavu TRUE. Tento funkční blok lze také použít k pokračování po chybovém stavu v případě, že je uložena potřebná sada dat při výskytu chyby.

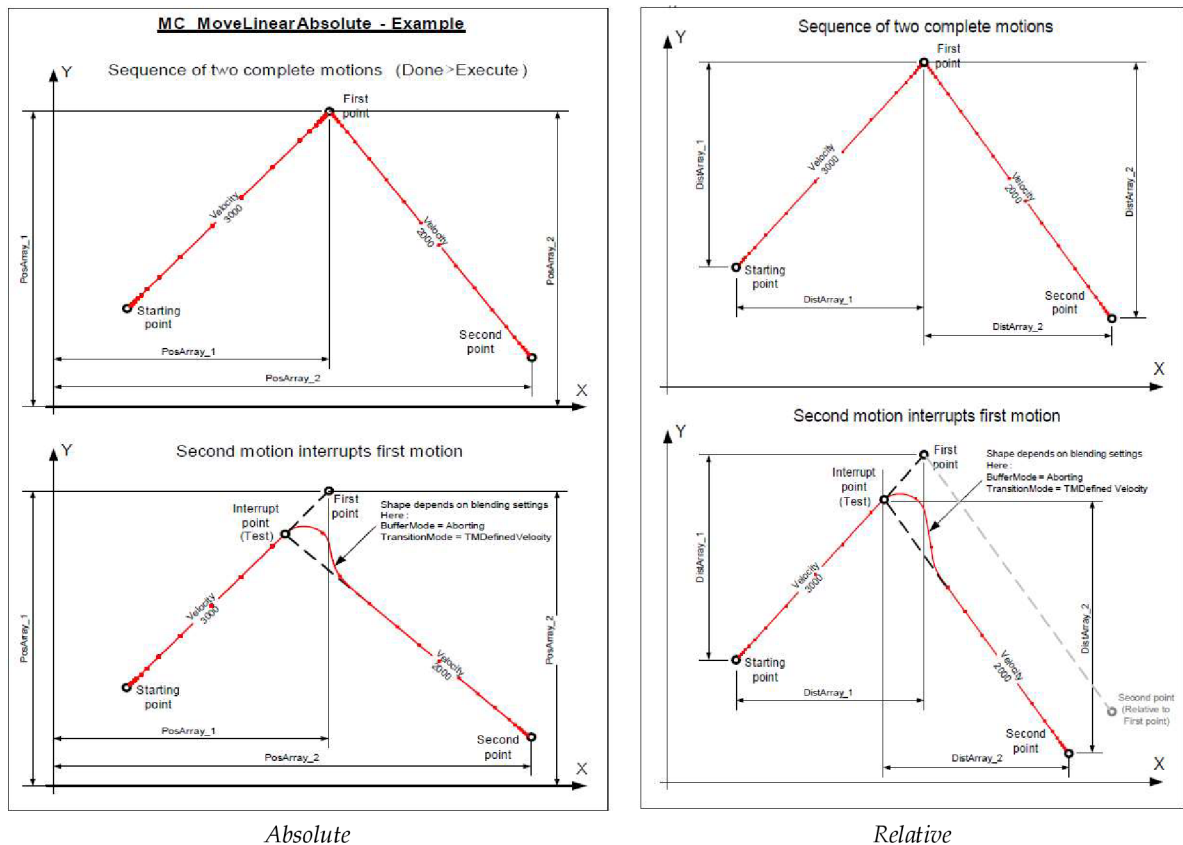
MC_GroupReadStatus – Výstupem tohoto funkčního bloku je stav, ve kterém se nachází skupina os. Jedná se o administrativní funkční blok, protože není generován žádný pohyb.

MC_GroupReadError – Tento funkční blok popisuje obecné chyby skupiny os, které se netýkají funkčních bloků. Jedná se o administrativní funkční blok, protože není generován žádný pohyb.

MC_GroupReset – Tento funkční blok provede přechod ze stavu GroupErrorStop na GroupStandby resetováním všech chyb souvisejících se skupinou os. Tento funkční blok také resetuje všechny osy v této skupině stejně jako příkaz MC_Reset.

MC_MoveLinearAbsolute – Tento funkční blok provede interpolovaný lineární pohyb skupiny os z aktuální polohy TCP do absolutní polohy zadané v konkrétním souřadnicovém systému. Chování tohoto funkčního bloku je znázorněno na obrázku níže.

MC_MoveLinearRelative – Tento funkční blok provede interpolovaný lineární pohyb skupiny os z aktuální polohy TCP do relativní polohy zadané v konkrétním souřadnicovém systému vzhledem k počátečnímu bodu. Chování tohoto funkčního bloku je znázorněno na obrázku níže.



Obr. 3-11 – Move Linear [16]

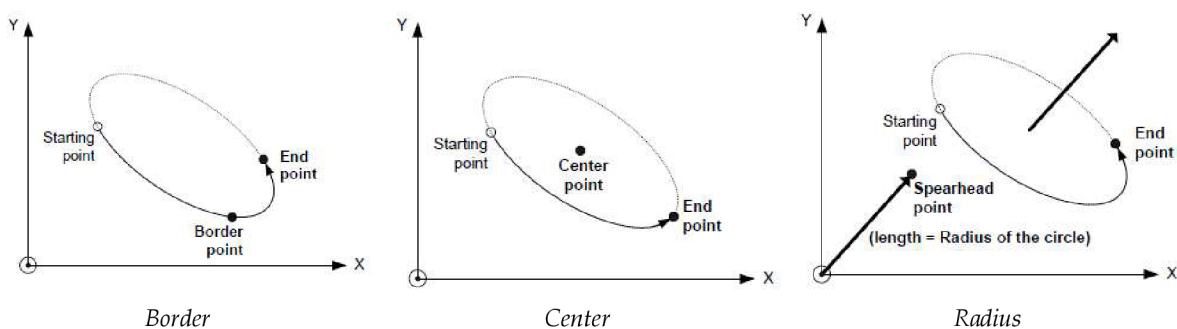
MC_MoveCircularAbsolute – Tento funkční blok provede interpolovaný kruhový pohyb skupiny os z aktuální polohy TCP do absolutní koncové polohy zadané v konkrétním souřadnicovém systému. Je třeba také zadat pomocný bod podle toho, v jakém módu je pohyb prováděn. Existují následující tři módy:

„Border“ – V tomto módu uživatel zadá pomocný bod, který leží na části kružnice opisované strojem, a stroj musí tímto bodem projít.

„Center“ – V tomto režimu uživatel zadá pomocný bod, který je středem kružnice opisované strojem, a stroj tímto bodem neprochází. V tomto režimu je definováno vstupem „PathChoice“, zda má být provedena kratší nebo delší část kružnice.

„Radius“ – V tomto módu uživatel definuje vektor kolmý k rovině kruhu podle pravidla pravé ruky. Délka tohoto vektoru odpovídá poloměru kružnice opisované strojem.

MC_MoveCircularRelative – Tento funkční blok provede interpolovaný kruhový pohyb skupiny os z aktuální polohy TCP do relativní koncové polohy zadané vzhledem k počátečnímu bodu v konkrétním souřadnicovém systému. Je třeba také zadat pomocný bod vzhledem k počátečnímu bodu podle toho, v jakém módu je pohyb prováděn. Existují tři módy pohybu, které jsou stejné jako u funkčního bloku MC_MoveCircularAbsolute.



Obr. 3-12 – Move Cirkular [16]

MC_MoveDirectAbsolute – Tento funkční blok provede pohyb skupiny os do zadané absolutní polohy v konkrétním souřadnicovém systému, aniž by se staral o to jak (po jaké trajektorii) je cílové polohy dosaženo.

MC_MoveDirectRelative – Tento funkční blok provede pohyb skupiny os do zadané relativní polohy vzhledem k počáteční poloze v konkrétním souřadnicovém systému, aniž by se staral o to jak (po jaké trajektorii) je cílové polohy dosaženo. Počáteční poloha je aktuální pozice TCP.

MC_PathSelect – Tento funkční blok připravuje příslušná data o trajektorii a pomocí výstupu „PathData“ je zpřístupní systému. Jedná se o administrativní funkční blok.

MC_MovePath – Tento funkční blok provede pohyb skupiny os po trajektorii zadané v „PathData“.

3.7.5 Řízení pohybu

Typ použitého funkčního bloku určuje cestu, po které dojde k dosažení nové cílové pozice. Souřadnicový systém, ve kterém je nová pozice definována, nemá vliv na tuto cestu. Existují dva typy pohybů, které je třeba rozlišovat:

Prvním typem jsou pohyby PTP (Point to Point). Podstatou tohoto typu pohybu je dosažení cílové pozice co nejrychleji. Toho lze dosáhnout pohybem každé osy po co nejkratší cestě z výchozí polohy do cílové polohy. Obvykle je tento druh pohybu nejrychlejší cestou k dosažení nové cílové pozice, protože v každém okamžiku se alespoň jedna osa pohybuje na

svém dynamickém limitu. Cesta a rychlost cesty TCP nejsou důležité. Jsou určeny polohou os a kinematickou transformací stroje. Proto je tento druh pohybu použitelný pro manipulace a v případě, kdy není cesta TCP rozhodující. Doporučuje se, aby všechny osy dorazily na příkazovou pozici ve stejném časovém okamžiku (synchronizovaně).

Druhým typem jsou pohyby CP (Cartesian Path movements). Pohyby CP způsobují, že se TCP pohybuje po definované cestě v kartézském prostoru. Cesta může být přímka, kruhový pohyb nebo funkce křivky. Cesta, přes kterou je dosaženo nové cílové pozice, je důležitá. Například v případě zpracování obrobku. Dále lze přímo řídit rychlost cesty TCP. Na rozdíl od PTP pohybu je proces polohování každé osy určován požadovanou cestou a inverzní kinematickou transformací.

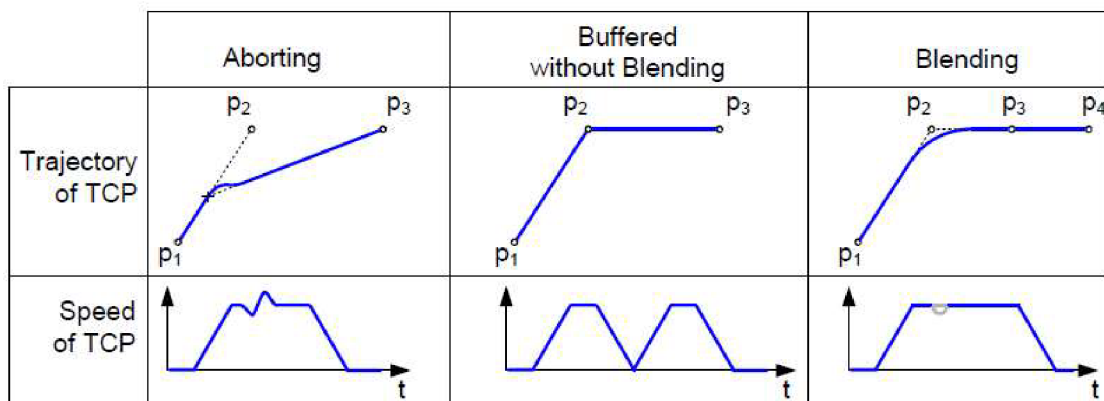
Navazování různých trajektorií

Základním prvkem interpolačního řízení pohybu je plynulé navazování po sobě jdoucích příkazů. Bez plynulého navazování pohybu se osy pohybují směrem k cílové pozici, zpomalí a přeruší pohyb přesně na cílové pozici. Následující příkaz s následující cílovou pozicí se zatím nezobrazuje. Osy poté musí zrychlit znovu. V mnoha aplikacích je požadováno jiné chování, chceme pohyby navázat bez zastavení. Důvody jsou např. snížení doby cyklu, vytvoření hladšího pohybu a snížení mechanického namáhání, dosažení konstantní rychlosti, která je některými aplikacemi požadována (nanášení lepidla, svařování atd.). [23]

Toho lze dosáhnout různými druhy navazování pohybu. Všechny tyto druhy navazování pohybu mají společnou jednu charakteristiku, a to modifikaci původní cesty, výsledkem je hladká trajektorie bez rohů.

Řízení pohybu má na starosti plánovač trajektorie, který je součástí interpolátoru a řídí a plánuje pohyb a jeho navazování podle nastavených módů, které budou popsány v kapitolách níže.

Sloučení pohybových příkazů v interpolačním řízení pohybu se liší od sloučení pohybových příkazů v rámci jednotlivých os. Při řízení jednotlivých os je vždy dosaženo cílové pozice. Rychlost v okamžiku, kdy je dosaženo cílové pozice, může být změněna podle vstupního parametru. Při interpolačním řízení pohybu lze v závislosti na aplikaci a procesu uvažovat o několika typech sloučení pohybu.



Obr. 3-13 – Průběh trajektorie a rychlosti [16]

Definice pojmů

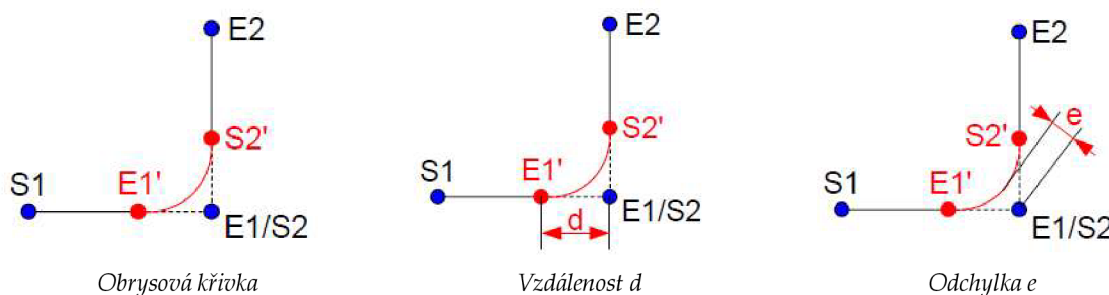
Obrysová křivka – křivka, která upravuje původní trajektorii ($E1' - S2'$).

Pre-block – Funkční blok pohybu před obrysovou křivkou ($S1 - E1$).

Post-block – Funkční blok pohybu za obrysovou křivkou ($S2 - E2$).

Vzdálenost d – Vzdálenost počátečního bodu obrysové křivky ($E1'$) k naprogramovanému cílovému bodu ($E1$)

Odchylka e – Nejkratší vzdálenost mezi naprogramovaným rohovým bodem ($E1 / S2$) a obrysovou křivkou.



Obr. 3-14 – Definice pojmů [16]

Vstupní parametry

Vstupní parametr "TransitionMode" v kombinaci se zadáním "TransitionParameter" definuje tvar a dynamiku vložené obrysové křivky pro připojení aktuálního funkčního bloku pohybu s následujícím funkčním blokem pohybu. Pro tento účel jsou naprogramované funkční bloky pohybu modifikovány. Tento parametr neurčuje chronologii provádění, to je dáno vstupem "BufferMode". Pro každý BufferMode musí být zadán konkrétní TransitionMode.

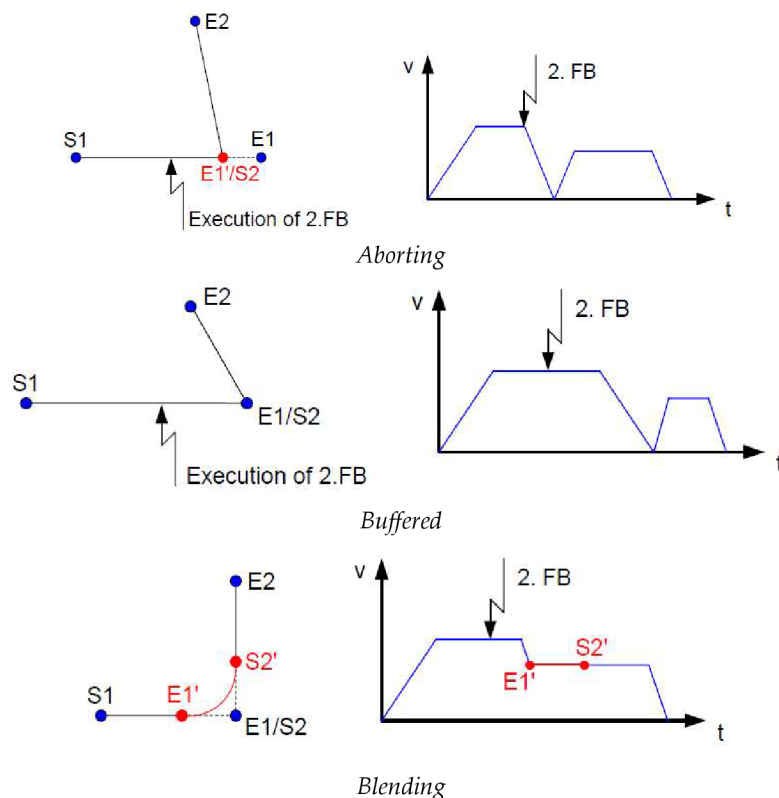
Buffer Mode

Nyní budou popsány možné vstupy proměnné „BufferMode“.

„Aborting“ – Funkční blok se vstupem "Aborting" přeruší probíhající pohyb a okamžitě zahájí nový pohyb. Následující obrázek ukazuje příklad, kdy se první pohyb zastaví a pak se spustí další pohyb.

„Buffered“ – Funkční blok se vstupem „Buffered“ ovlivní skupinu os, jakmile je předchozí krok "Done", tzn. nepřeruší probíhající pohyb.

„Blending“ – Aktuální a následující pohybové funkční bloky jsou zkombinovány, takže skupina os se nezastaví mezi pohyby a plynule naváže na následující funkční blok. Obrysová křivka je definován vstupním parametrem „TransitionMode“.



Obr. 3-15 – Buffer Mode [16]

Transition Mode

Nyní budou popsány možné vstupy proměnné „TransitionMode“.

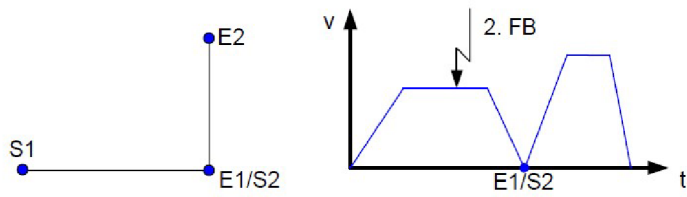
„TMNone“ – Tento vstup nevloží žádnou obrysovou křivku a funkční bloky se nezmění, jedná se o výchozí režim. Jde o jediný možný vstup pro režim „Buffered“. Pokud použijeme tento vstup při režimu „Blending“, výsledek bude stejný jako v režimu „Buffered“. Na obrázku níže je vidět obrysová křivka tohoto vstupu v režimu „Aborting“.

„TMStartVelocity“ – Jedná se o vstup, který má danou maximální počáteční rychlost pro obrysovou křivku. Tato počáteční rychlost se udává v procentech vzhledem k rychlosti, která je daná ve funkčním bloku předcházejícímu obrysové křivce. V momentě dosažení této počáteční rychlosti se začíná provádět obrysová křivka, ale rychlost během provádění obrysové křivky nemusí být konstantní. Na obrázku níže je příklad pohybu, pokud je vstup nastaven na 50 %.

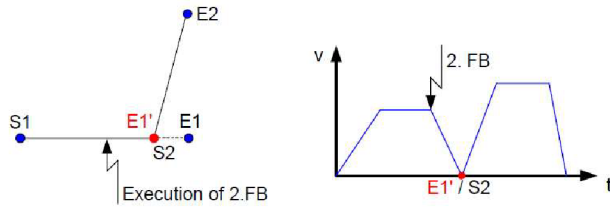
„TMConstantVelocity“ – Jedná se o vstup, který má danou konstantní rychlost pro obrysovou křivku. Rychlost se udává v procentech vzhledem k rychlosti dané v následujícím funkčním bloku po provedení obrysové křivky. Na obrázku níže je znázorněno provedení obrysové křivky v režimu Blending a v režimu Aborting.

„TMCornerDistance“ – Jedná se o vstup, který má danou vzdálenost d počátečního bodu obrysové křivky k naprogramovanému cílovému bodu.

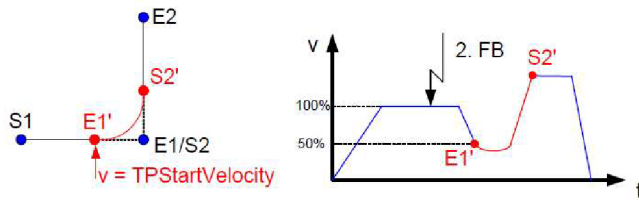
„TMMaxCornerDeviation“ – Jedná se o vstup, který má danou odchylku e , tedy nejkratší vzdálenost mezi naprogramovaným rohovým bodem a obrysovou křivkou.



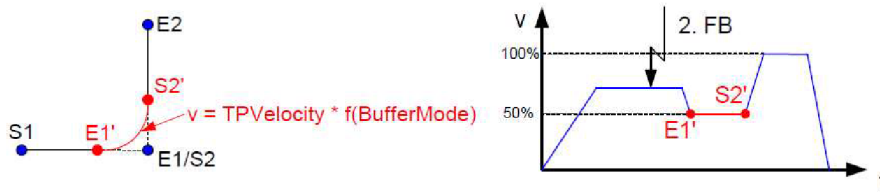
TMNone pro režim Buffered a Blending



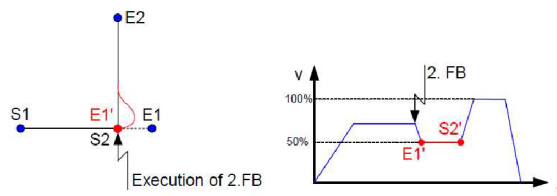
TMNone pro režim Abortig



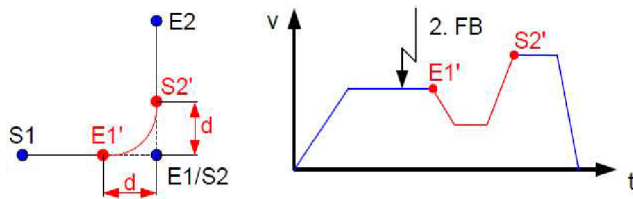
TMStartVelocity = 50%



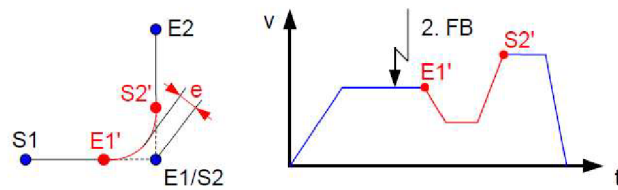
TMConstantVelocity pro režim Blending



TMConstantVelocity pro režim Abortig



TMCornerRadius



TMMaxCornerDeviation

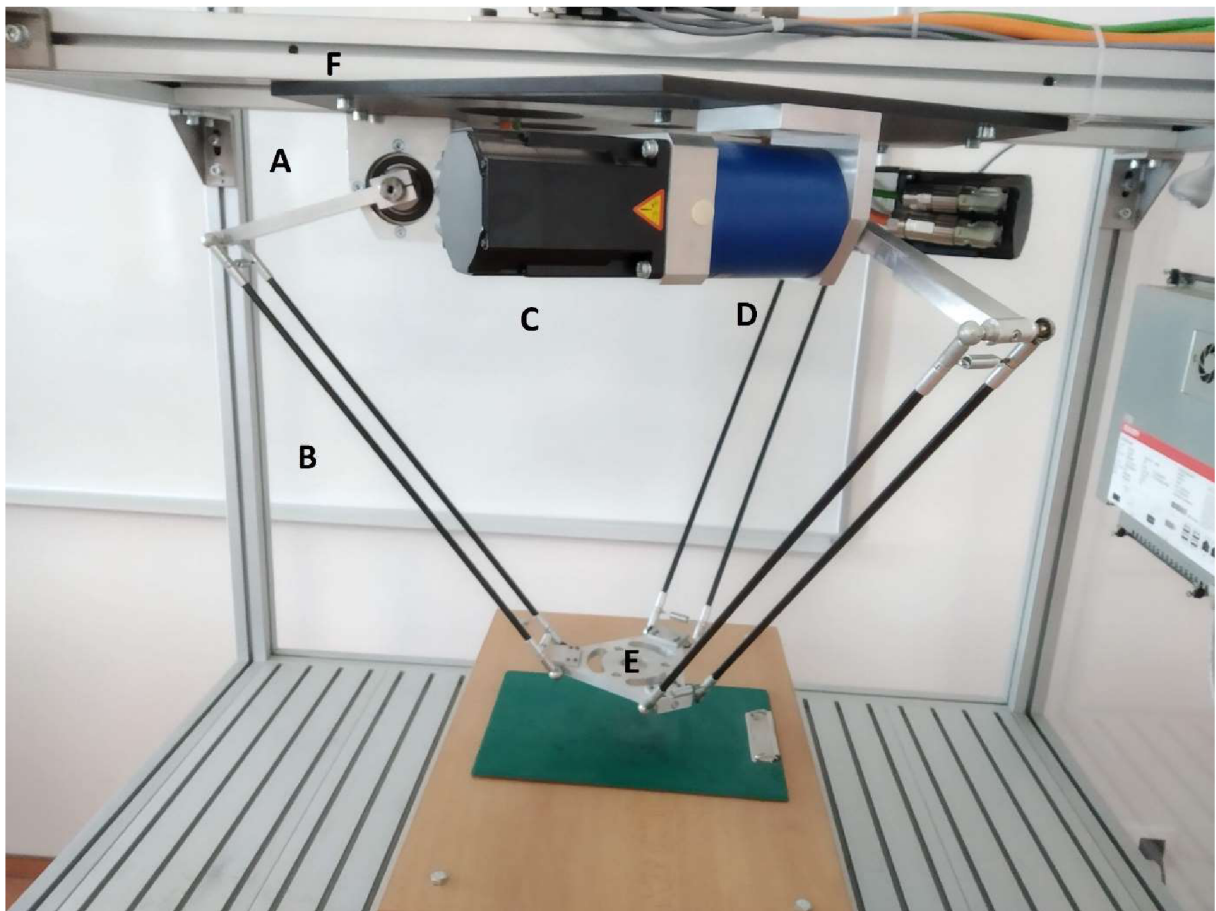
Obr. 3-16 – Transition Mode [16]

4 Praktická část

Cílem praktické části této diplomové práce je vytvořit řídicí software pro delta robot a HMI (Human Machine Interface) pro řízení tohoto robotu. Řídicí software bude vytvořen v programu TwinCAT 3 od firmy Beckhoff.

4.1 Hardware

Hardware je zobrazen na Obr. 4-1. Jedná se o tříosý delta robot, který se skládá z pevné základny (Obr. 4-1 F), tří ramen, kdy každé rameno má dvě části (Obr. 4-1 A a B) propojené mezi sebou rotační vazbou, pohyblivé platformy (Obr. 4-1 E), motoru (Obr. 4-1 C) a převodovky (Obr. 4-1 D).



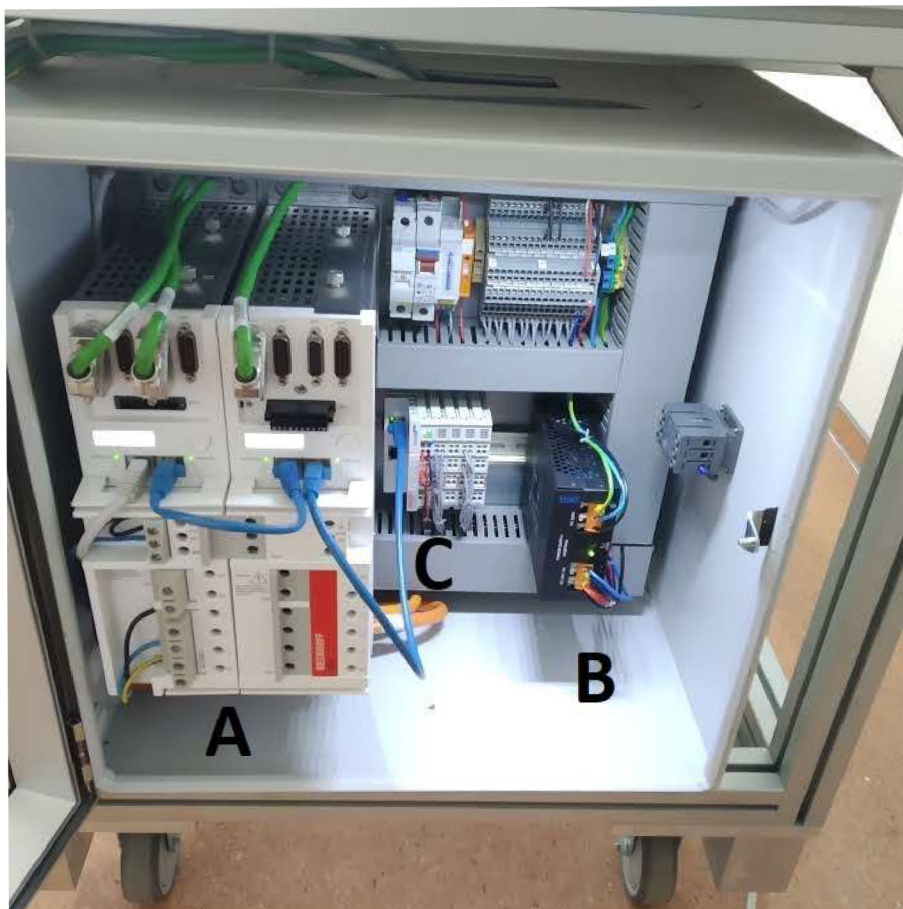
Obr. 4-1 – Delta robot

Rameno A měří 180 mm a rameno B měří 555 mm, displacement u základny je 260 mm a u platformy 90 mm. K pohyblivé platformě E lze připevnit koncový efektor. Každé rameno pohání vlastní motor (Obr. 4-1 C) propojený s planetární převodovkou (Obr. 4-1 D) typ Beckhoff AG2200 [18]. Motor je synchronní servomotor model Beckhoff AM3041 s následujícími parametry [17]:

Jmenovité napětí.....	400 V AC
Jmenovitý proud.....	2,7 A
Jmenovité otáčky.....	6000 min ⁻¹
Krouticí moment.....	1,92 Nm
Jmenovitý výkon.....	0,93 kW

Dále se systém skládá ze zdroje (Obr. 4-2 B) a dvou dvoukanalových měničů (Obr. 4-2 A) typu Beckhoff AX5203 s následujícími parametry [19]:

Vstupní jmenovité napětí (1 fáze)	100-240 V AC
Vstupní jmenovité napětí (3 fáze)	100-480 V AC
Výstupní jmenovitý proud.....	2 · 3 A
Vstupní frekvence.....	50/60 Hz
Výstupní frekvence.....	0-1000 Hz

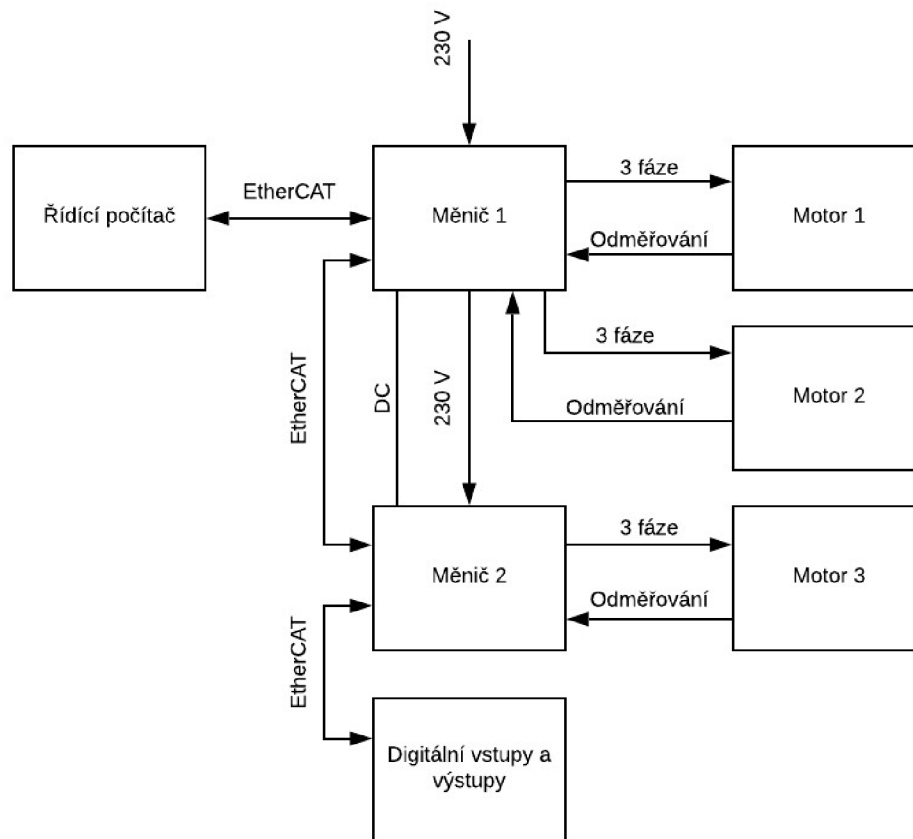


Obr. 4-2 – Hardware

Na Obr. 4-2 u písmene C jsou zobrazeny karty digitálních vstupů a výstupů. Karta pro digitální vstupy má označení EL1008 (8 vstupů) [21] a karta pro digitální výstupy EL2008 (8 výstupů) [22].

Vše je řízeno řídicím počítačem C6015-0010 od společnosti Beckhoff [20].

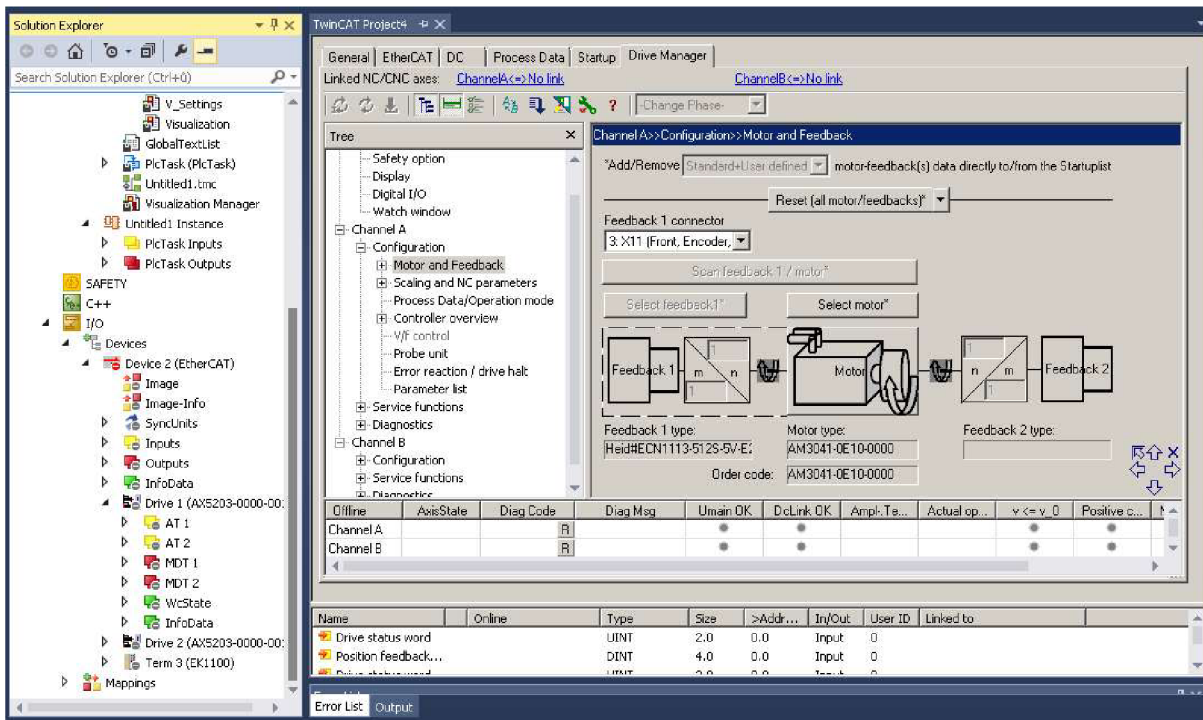
Na následujícím obrázku je schéma propojení hardwarových prvků.



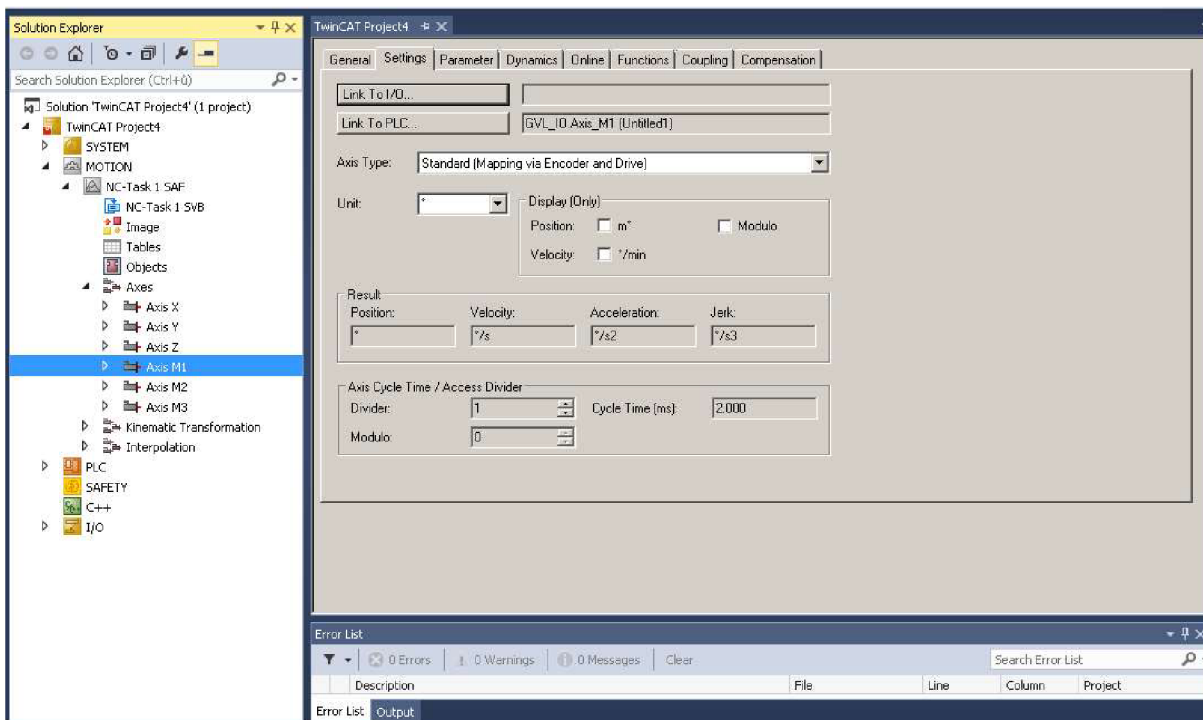
Obr. 4-3 - Hardware schéma

4.2 Nastavení hardwaru

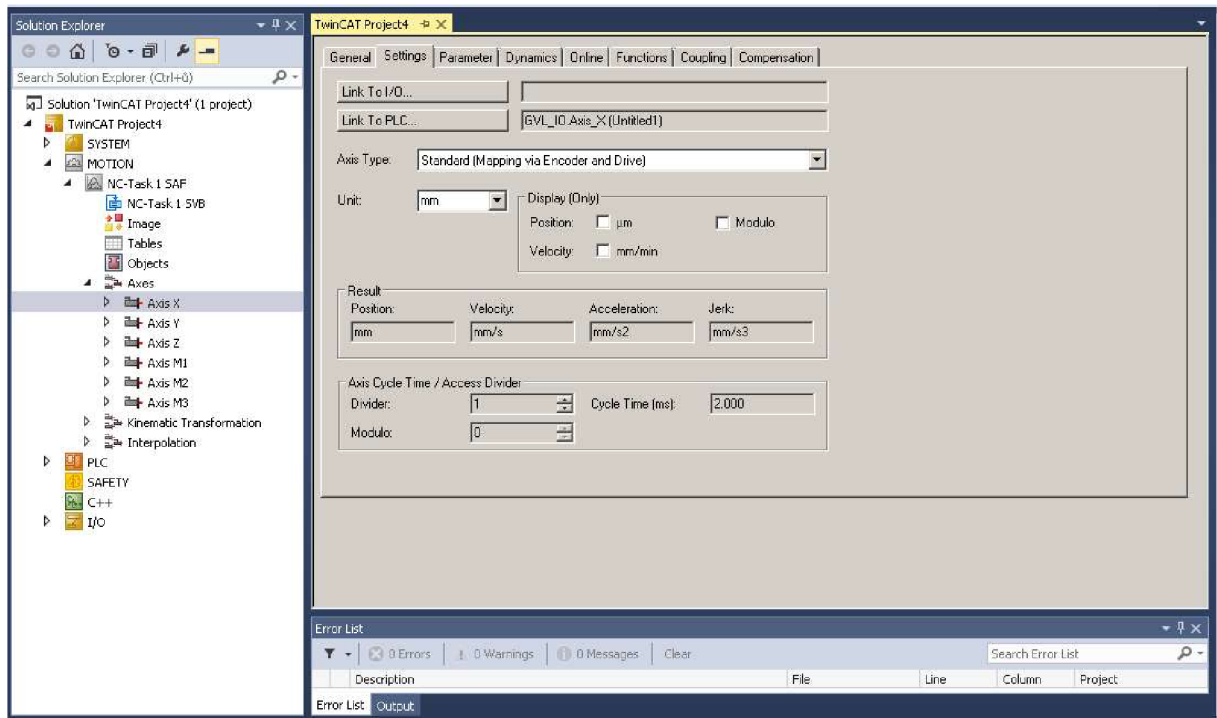
V programu TwinCAT 3 bylo nejprve nutné naskenovat hardware, díky tomu nebylo nutné zadávat ručně typ motoru a jeho parametry, program byl schopný je sám rozpoznat (**Chyba! Nenalezen zdroj odkazů.** Obr. 4-4). Poté bylo nutné vytvořit virtuální (X, Y, Z) a reálné osy (M1, M2, M3). Reálné osy (Obr. 4-5) představují motory na ramenech robotu, jejichž poloha je dána úhlem natočení ve stupních. Tyto reálné osy byly poté propojeny s motory, které se v systému objevily pomocí skenování hardwaru. Poloha virtuálních os je dána vzdáleností od počátku v milimetrech. Virtuální osy (Obr. 4-6) slouží k řízení robota v kartézském souřadném systému. Tyto virtuální osy byly sloučeny do skupiny os, aby mohly být řízeny naráz pomocí interpolace (Obr. 4-7). Nakonec byla vytvořena kinematická transformace, která převádí souřadnice virtuálních os na souřadnice reálných os a naopak (Obr. 4-8). Kinematická transformace je definována typem robota – „Delta Type 1“ (vybráno z nabídky programu TwinCAT 3) a jeho parametry (Obr. 4-9 a Obr. 4-10), které byly uvedeny již v kapitole 4.1.



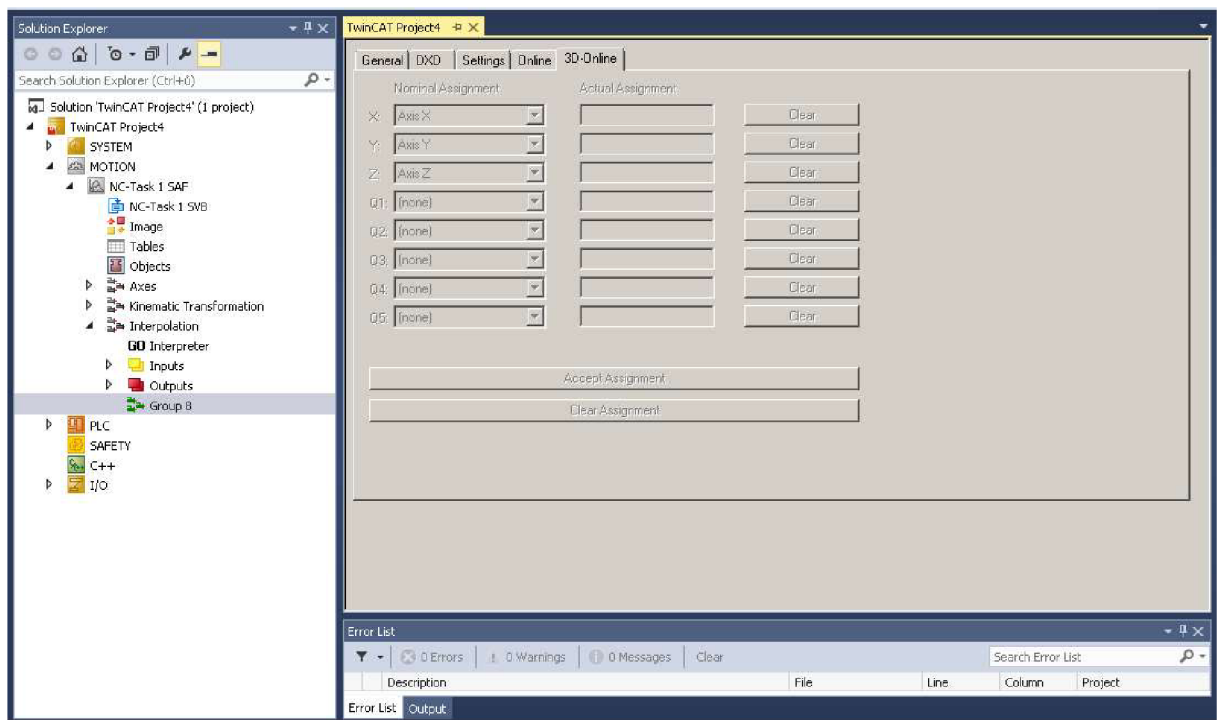
Obr. 4-4 - Skenování hardwaru - motory



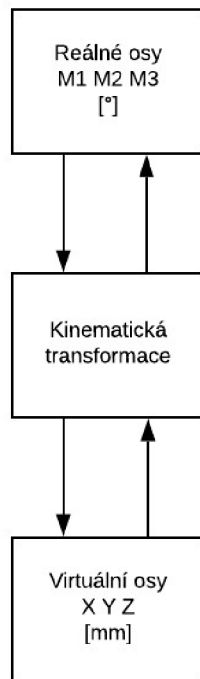
Obr. 4-5 - Reálné osy



Obr. 4-6 - Virtuální osy

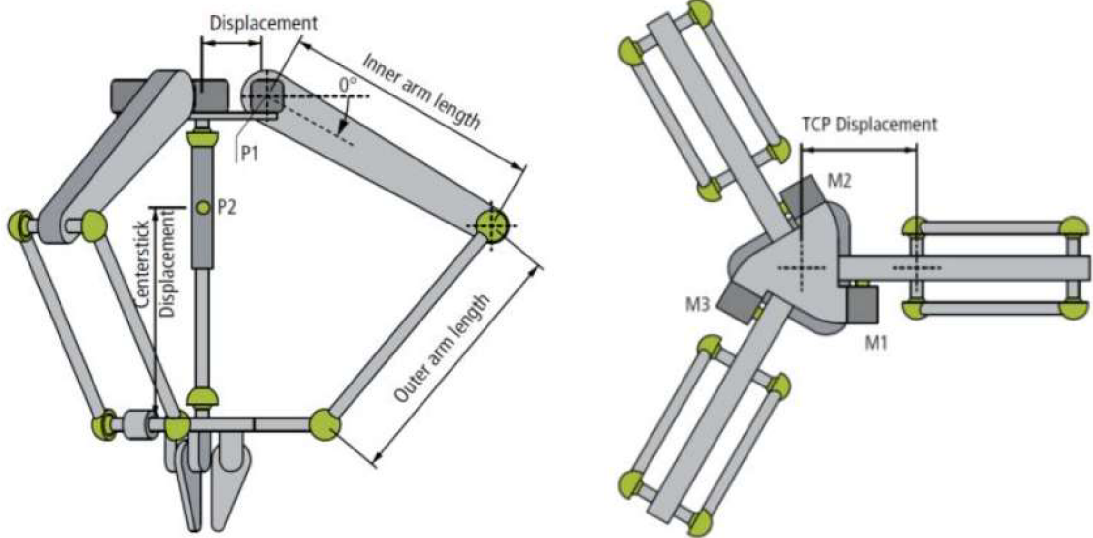


Obr. 4-7 – Interpolace

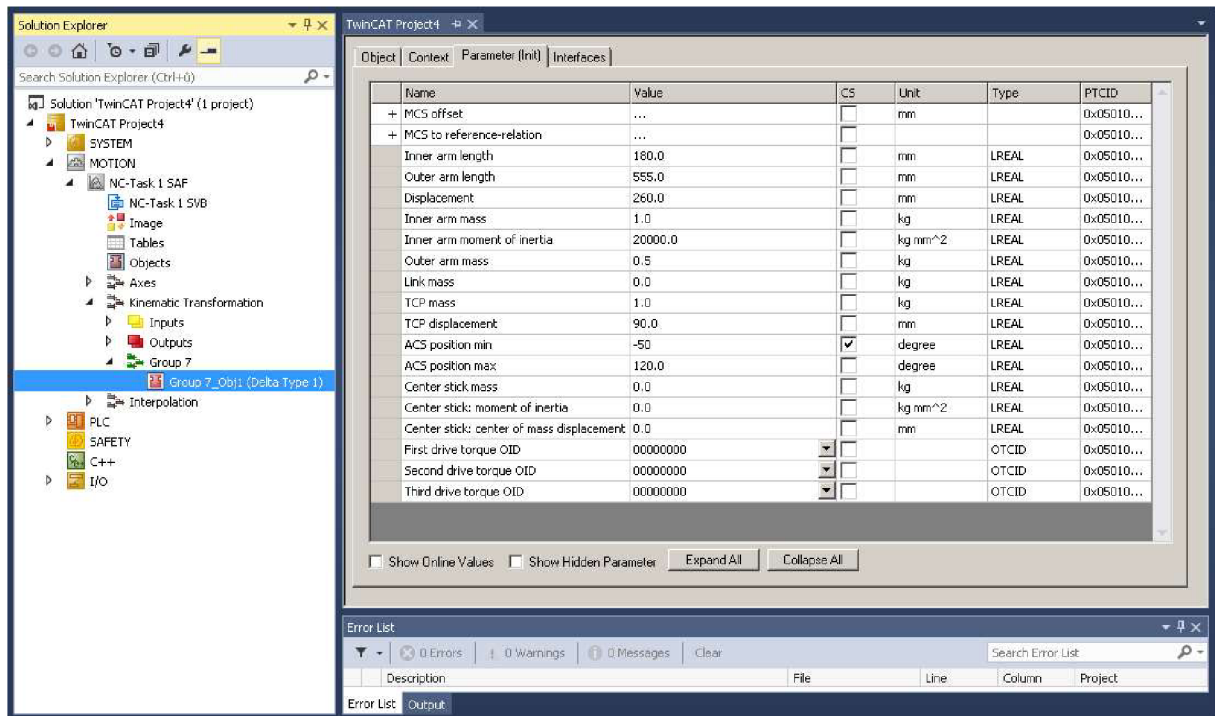


Obr. 4-8 - Kinematická transformace

Delta Type 1



Obr. 4-9 - Delta Type 1 [15]



Obr. 4-10 - Parametry kinematické transformace

4.3 Tvorba softwaru

V této kapitole bude popsána tvorba softwaru v programu TwinCAT 3 od společnosti Beckhoff.

4.3.1 PLC a NC

Všechny úlohy, které v systému probíhají, se dají rozdělit na PLC (programmable logic controller) a NC (numerical control) úlohy. PLC řídí vizualizaci, vstupy a výstupy a je nadřazené NC – dává příkazy k vykonání NC úloh. NC zajišťuje pohyb všech ramen robotu po požadované trajektorii, požadovanou rychlostí a požadovaným zrychlením. Součástí NC jsou i kinematické transformace nutné pro přepočítávání souřadnic a dosažení požadovaného pohybu delta robotu.

Klasické PLC funguje tak, že načte vstupy, provede daný program, zapíše výstupy a toto opakuje neustále dokola ve smyčce. V tomto případě bylo použito softwarové PLC, kdy PC vykonává úlohy pro vlastní operační systém i PLC úlohy a je zároveň i vývojovým prostředím pro vytvoření PLC programu. Výhody softwarového PLC jsou hlavně v téměř neomezené velikosti paměti, vysokém výpočetním výkonu a nízkému množství komponent. [15]

Součástí NC úloh je i NC I (NC interpolation), který v systému TwinCAT 3 zajišťuje interpolaci trajektorie pohybu a řízení daných os podle zvolené kinematiky – v tomto případě delta robotu. Vše je řízeno pomocí PLC jednotlivými funkčními bloky.

4.3.2 TwinCAT 3

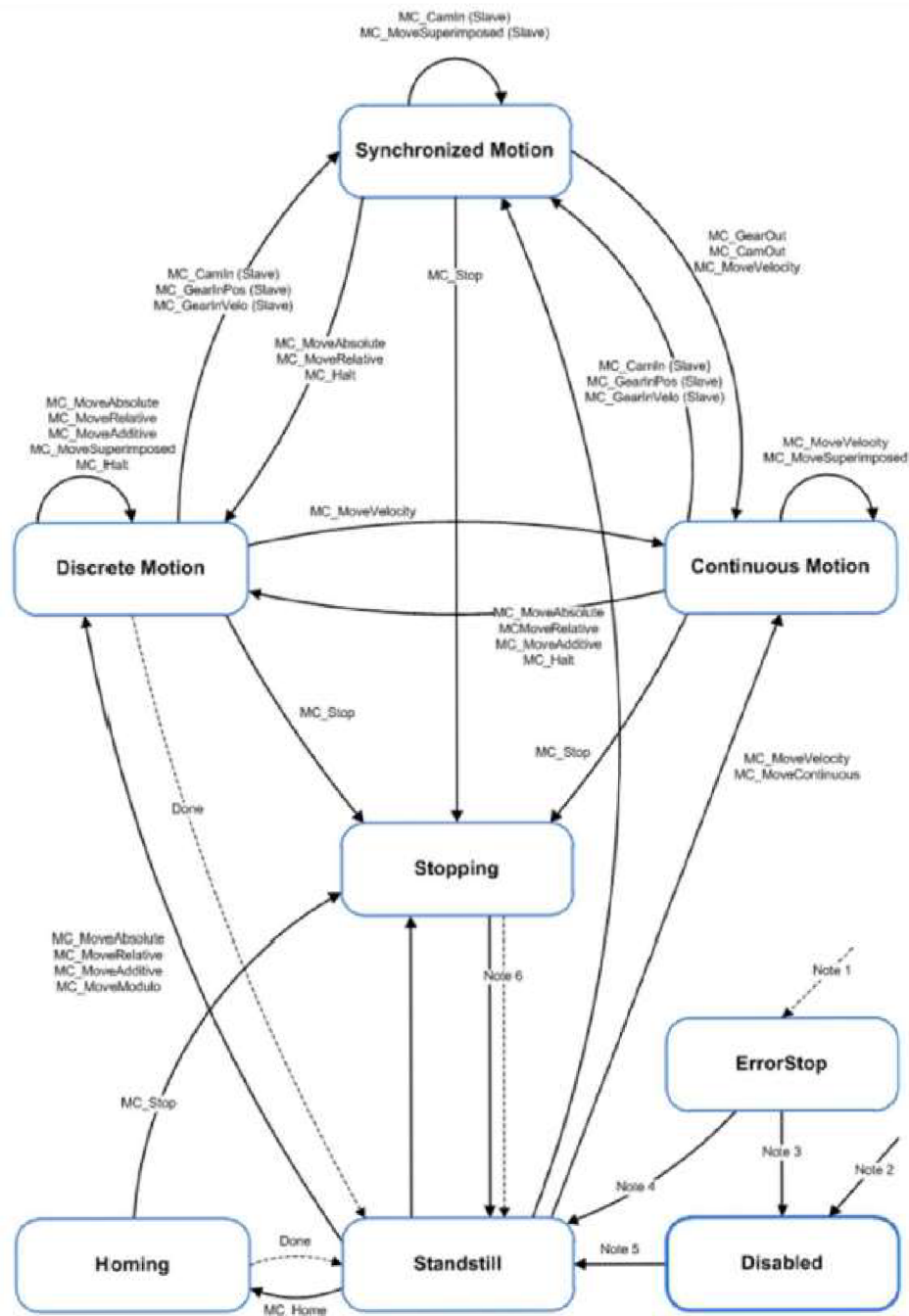
Software TwinCAT (The Windows Control and Automation Technology) od společnosti Beckhoff je založen na platformě Microsoft Visual Studio. Díky tomu je možné použít téměř kterýkoliv PC s procesorem Intel a systémem Windows k řízení PLC. TwinCAT se skládá

z několika modulů, na které se kupuje licence zvlášť, tudíž se vyhodnotí účel vytvářené systému a nakoupí se jen nutné licence. Seznam základních modulů: [15]

- TC3 Engineering (TE1000) – Modul umožňující realizaci vlastních projektů, vizualizace, online diagnostiku a měření.
- TC3 I/O (TC1100) – Modul k propojení fyzických vstupů a výstupů s proměnnými použitými v programu.
- TC3 PLC (TC1200) – Modul k vytvoření vlastních softwarových PLC systémů.
- TC3 NC PTP (TF5000) – Modul k NC řízení PTP pohybu (point to point).
- TC3 NC I (TF5100) – Modul umožňující interpolaci 3D trajektorie mezi jednotlivé posuvové osy. V kombinaci s modulem kinematických transformací umožňuje interpolaci pro sériovou kinematiku delta robota – použito v tomto případě.
- TC3 CNC (TF52xx) – Modul pro řízení obráběcích strojů.
- TC3 PLC HMI (TF1800) – Modul pro zobrazení vizualizace.

System TwinCAT nabízí také množství knihoven funkcí a funkčních bloků. Základní knihovna Tc2_standard je automaticky připojena ke každému nově vytvořenému projektu, další knihovny je možné do projektu přidat podle potřeby. Popis vybraných použitých knihoven: [15]

- Tc2_standard – tato knihovna obsahuje základní funkční bloky jako časovače, spouštěče, čítače atd.
- Tc2_Mc2 – tato knihovna obsahuje funkční bloky pro ovládání os jako spouštění, resetování, referování, relativní a absolutní pohyby, čtení polohy a rychlostí atd.



Obr. 4-11 - Funkční bloky knihovny Tc2_Mc2 [15]

4.3.3 Struktura PLC projektu v TwinCAT 3

PLC projekt v systému TwinCAT 3 je rozdělen do POU (Program Organisation Unit), kterou může být například program, funkce nebo funkční blok. POU mohou být psány jazyky ST (Structured Text), LD (Ladder Diagram), FBD (Function Block Diagram), IL (Instruction List), SFC (Sequential Function Chart), CFC (Continuous Function Chart) a skládají se z deklarační části a vlastního programového těla. [15]

Typy POU

Program je souhrnem instrukcí a nemá žádné vstupní ani návratové hodnoty. Program může být volán z jiného POU, pokud se nejedná o funkční blok.

Funkce poskytuje jednu návratovou hodnotu a počet vstupů není omezen. Funkce se volá pomocí svého názvu a hodnot vstupů v závorkách. Volání funkce se stejnými vstupy vrátí vždy stejné výstupy, protože funkce neponechává žádné stavové informace.

Funkční blok nemá omezen počet vstupů ani výstupů a na rozdíl od funkce uchovává stavové informace. Funkční blok se volá pomocí své instance – což je vlastně kopie funkčního bloku.

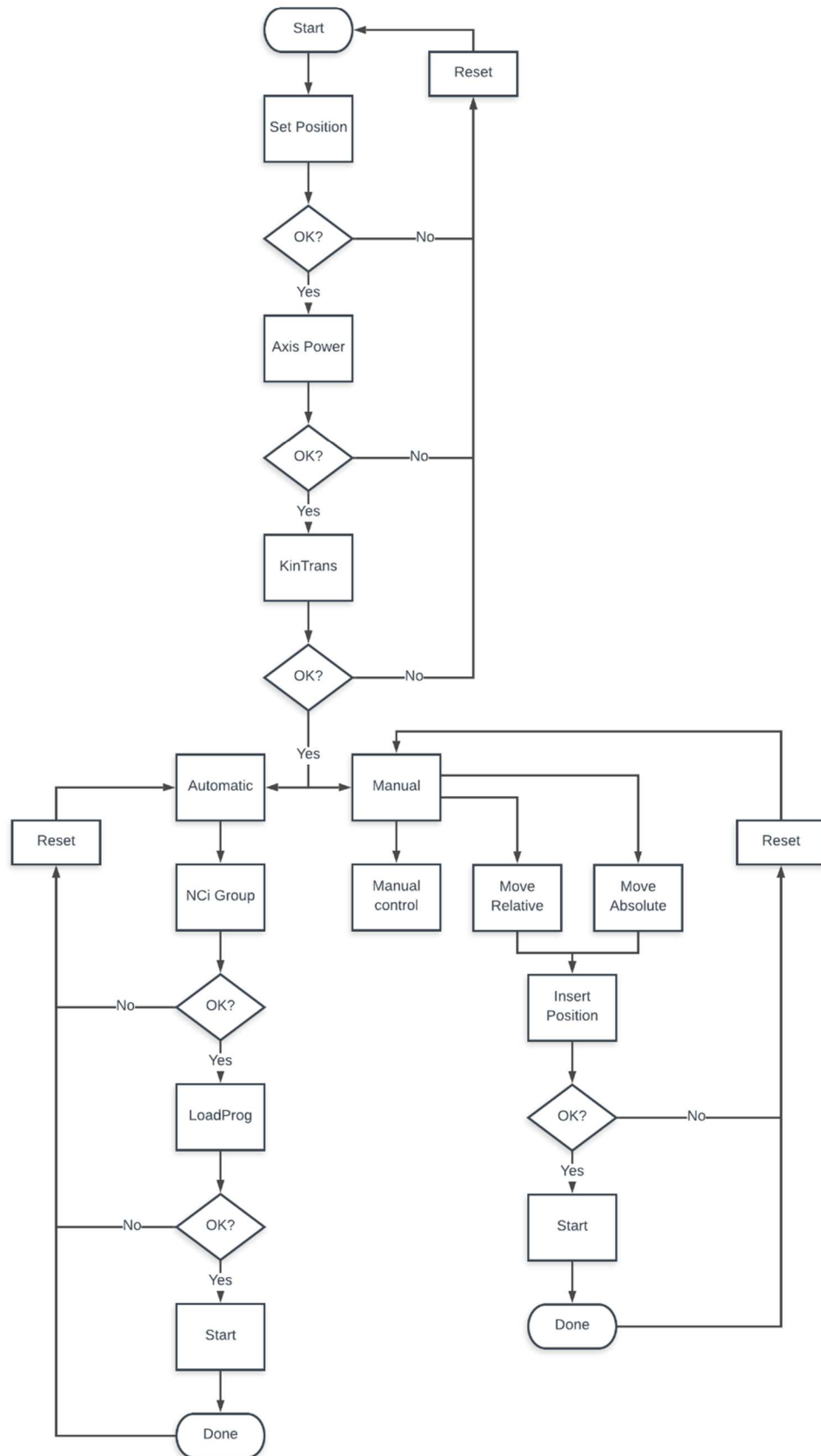
Global Variable List (GVL)

GVL je seznam a deklaráce globálních proměnných, které mohou používat všechny POU. V GVL musí být definovány také všechny proměnné, kterým se bude přiřazovat fyzický vstup nebo výstup. V GVL je deklaráce všech os pomocí datového typu `AXIS_REF`, které je zároveň rozhraním mezi PLC a NC a obsahuje všechny potřebné informace o dané ose jako je například stav, poloha a rychlost.

Visualization (VISU)

System TwinCAT umožňuje vytvořit grafické znázornění hodnot pomocí vkládání vizualizačních prvků z definovaných knihoven. Pomocí této vizualizace lze program také ovládat, zadávat vstupy, či měnit hodnoty proměnných.

4.3.4 Výsledný PLC projekt



Obr. 4-12 - Struktura softwaru

Všechny podprogramy a funkční bloky jsou volány z programové smyčky s názvem „MAIN“. Můj výsledný program pro řízení delta robotu obsahuje v této smyčce tyto podprogramy:

- P_SetPosition
- P_AxisPower
- P_KinTrans
- P_Jog
- P_MoveAbsolute
- P_MoveRelative
- P_NCiGroup
- P_LoadProg
- P_Correction
- P_Home

P_SetPosition

Program s názvem „P_SetPosition“ slouží k referování os. Obsahuje funkční blok „MC_SetPosition“. Vzhledem k tomu, že motory nemají absolutní odměřování a robot není vybaven ani referenčními spínači, je nutné toto referování provést ručně. Nejprve se tedy ručně nastaví ramena robotu do výchozí pozice a poté se stiskne na dotykové obrazovce v panelu „Settings“ tlačítko „Set Position“, kterým se osy softwarově vynulují.

P_AxisPower

Program s názvem „P_AxisPower“ slouží k zaktivování os, zapnutí pohonů a také k resetování všech os. Na dotykové obrazovce v panelu „Settings“ se stiskne tlačítko „Axis Power“, kterým se zapnou pohony. Toto obstarává funkční blok „MC_Power“. Pokud všechno proběhlo v pořádku, rozsvítí se zelené světýlko u každé osy, to značí, že osa je aktivní a je možné s ní jezdit. Pokud dojde k chybě a u některé osy nesvítí zelené světýlko, stiskne se tlačítko „Axis Reset“, které slouží k resetování všech os. Resetování os obstarává funkční blok „MC_Reset“.

P_KinTrans

Program s názvem „P_KinTrans“ slouží k zapnutí kinematické transformace, která přepočítává souřadnice mezi MCS a ACS. Tento program obsahuje funkční blok „FB_KinConfigGroup“, který slouží k zapnutí kinematické transformace a funkční blok „FB_KinResetGroup“, který slouží k resetování. Na dotykové obrazovce se kinematické transformace zapnou tlačítkem „KinTrans ON“ v panelu „Settings“, správné zapnutí kinematických transformací je indikováno zeleným světýlkem vedle tohoto tlačítka. V případě zapnuté kinematické transformace se dají ovládat pouze osy X, Y a Z a jejich aktuální poloha se přepočítává na aktuální polohu os M1, M2 a M3. Pokud je nutné kinematickou transformaci vypnout, stiskne se tlačítko „KinTrans OFF“ opět v panelu „Settings“, správné vypnutí indikuje zelené světýlko u tohoto tlačítka. V případě, že jsou kinematické transformace vypnuté, je možné ovládat pouze osy M1, M2 a M3, protože se aktuální polohy mezi těmito osami a osami X, Y a Z nepřepočítávají.

P_Jog

Program s názvem „P_Jog“ umožňuje ruční ovládání pohybu všech os. Obsahuje funkční blok „MC_Jog“. Tento funkční blok je v programu volán pro každou osu zvlášť a definuje její pohyb dopředu a dozadu. V tomto programu je také naprogramováno přepínání režimu „SLOW“ a „FAST“, které se na dotykové obrazovce přepíná tlačítkem „SLOW/FAST“ v panelu „Manual“. V tomto panelu je také ruční ovládání všech os pomocí tlačítek „+X“, „-X“, „+Y“,

„-Y“, „+Z“, „-Z“, „+M1“, „-M1“, „+M2“, „-M2“, „+M3“, „-M3“ a zobrazení aktuální polohy všech os.

P_MoveAbsolute

Program s názvem „P_MoveAbsolute“ slouží k najetí na konkrétní požadovanou polohu, kterou uživatel ručně zadá pomocí dotykové obrazovky. V programu je funkční blok „MC_MoveAbsolute“ pomocí kterého je tato funkce realizována. Na dotykové obrazovce se tato funkce aktivuje stisknutím tlačítka „MoveAbsolute“, poté uživatel zadá požadované souřadnice do polí „PositionX“, „PositionY“ a „PositionZ“ a stiskne tlačítko „Start“. Tím zahájí pohyb os na danou pozici.

P_MoveRelative

Program s názvem „P_Moverelative“ slouží k posunutí os o konkrétní požadovanou vzdálenost, kterou uživatel ručně zadá pomocí dotykové obrazovky. V programu je funkční blok „MC_MoveRelative“ pomocí kterého je tato funkce realizována. Na dotykové obrazovce se tato funkce aktivuje stisknutím tlačítka „MoveRelative“, poté uživatel zadá požadovanou vzdálenost, o kterou chce s osami popojet, do polí „PositionX“, „PositionY“ a „PositionZ“ a stiskne tlačítko „Start“. Tím zahájí pohyb os na danou vzdálenost.

P_NCiGroup

Program s názvem „P_NCiGroup“ je nezbytný pro automatický režim robotu. Tento program obsahuje funkční blok „CfgBuild3DGroup“ pro vytvoření 3D interpolační skupiny a funkční blok „CfgReconfigGroup“ pro zrušení 3D interpolační skupiny. Na dotykové obrazovce v panelu „Automatic“ vytvoříme tuto skupinu pomocí tlačítka „Automatic“, a tím zároveň zaktivníme automatický režim robotu. V tomto režimu není možné ruční ovládání a pohyb všech os, je možné pouze nahrání programu a jeho spuštění v panelu „Automatic“ (viz kap. 4.5.8.). Pokud je potřeba se vrátit do manuálního režimu, tak je možné 3D interpolační skupinu vypnout pomocí tlačítka „Manual“ v panelu „Manual“, a tím také vypnou automatický režim a znemožnit nahrání a spuštění programu v panelu „Automatic“.

P_LoadProg

Program s názvem „P_LoadProg“ slouží k nahrání již vytvořeného programu ve formátu .nc a jeho spuštění. K tomu slouží funkční blok „ItpLoadProg“, který načte daný program, funkční blok „ItpStartStop“, který program spustí a zastaví a funkční blok „ItpReset“, který program resetuje. Na dotykové obrazovce je ovládání v panelu „Automatic“, do okna vedle tlačítka „Load“ se napíše cesta k požadovanému programu, poté se stiskne tlačítko „Load“, čímž se program načte. Správné načtení programu je indikováno zeleným světýlkem vedle nápisu „LoadProg OK“, v případě chyby se rozsvítí červené světýlko vedle nápisu „LoadProg Error“. Pokud je vše v pořádku, program lze spustit pomocí tlačítka „Start“ a zastavit pomocí tlačítka „Stop“. V panelu „Automatic“ lze také nastavit požadovaný Override a jsou zde zobrazeny i aktuální pozice os X, Y a Z.

P_Correction

Program s názvem „P_Correction“ slouží k posunutí počátku souřadného systému. Počátek lze posunout do předem definované polohy pomocí funkcí G54, G55 a G56. Toto posunutí aktivujeme pomocí tlačítek „G54“, „G55“ a „G56“ v panelu „Manual“. Zde je také možné tyto funkce ručně předefinovat a zvolit každé z nich vlastní posunutí, a to pomocí tlačítek „XYZ“ pro vynulování všech os naráz, nebo vynulováním každé osy zvlášť pomocí tlačítek „X0“, „Y0“ a „Z0“.

P_Home

Program s názvem „P_Home“ byl nachystán pro případ, že se na robot namontují referenční snímače. Funkční blok „MC_Home“ zajistí pohyb os směrem k nulové pozici, jakmile osy najedou na referenční snímač, tento snímač pošle signál a funkční blok zastaví pohyb. Toto referování se spustí tlačítkem „Home M1, M2, M3“ v panelu „Settings“ a jeho správné dokončení je signalizováno zeleným světýlkem vedle toho tlačítka. Vzhledem k tomu, že v současné době nejsou na robotu referenční snímače, je jejich funkce pro testování nahrazena tlačítkem „Calibration Cam (test button)“, které po stlačení zastaví pohyb os na pozici „Home“.

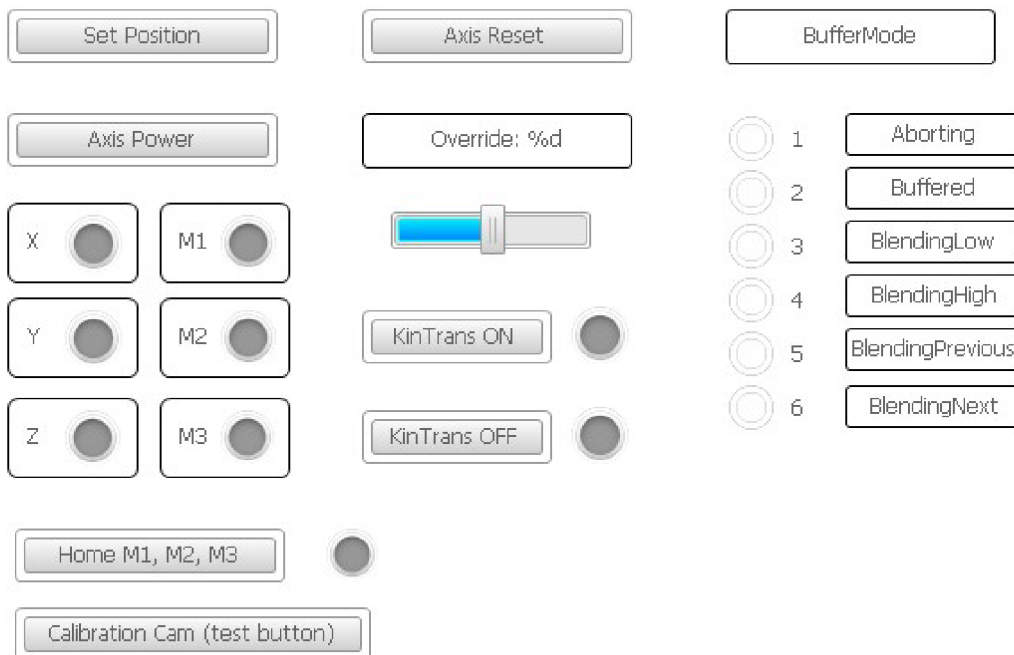
4.3.5 HMI

HMI (Human-Machine Interface) bylo realizováno pomocí vizualizace vytvořené v prostředí TwinCAT 3 pomocí prvků dostupných z knihovny. Každý prvek ve vizualizaci představuje konkrétní proměnnou. Některé prvky pouze zobrazují hodnotu dané proměnné a pomocí některých prvků může uživatel měnit hodnotu dané proměnné. Konečná vizualizace je složena ze 3 panelů:

- Settings
- Manual
- Automatic

Settings

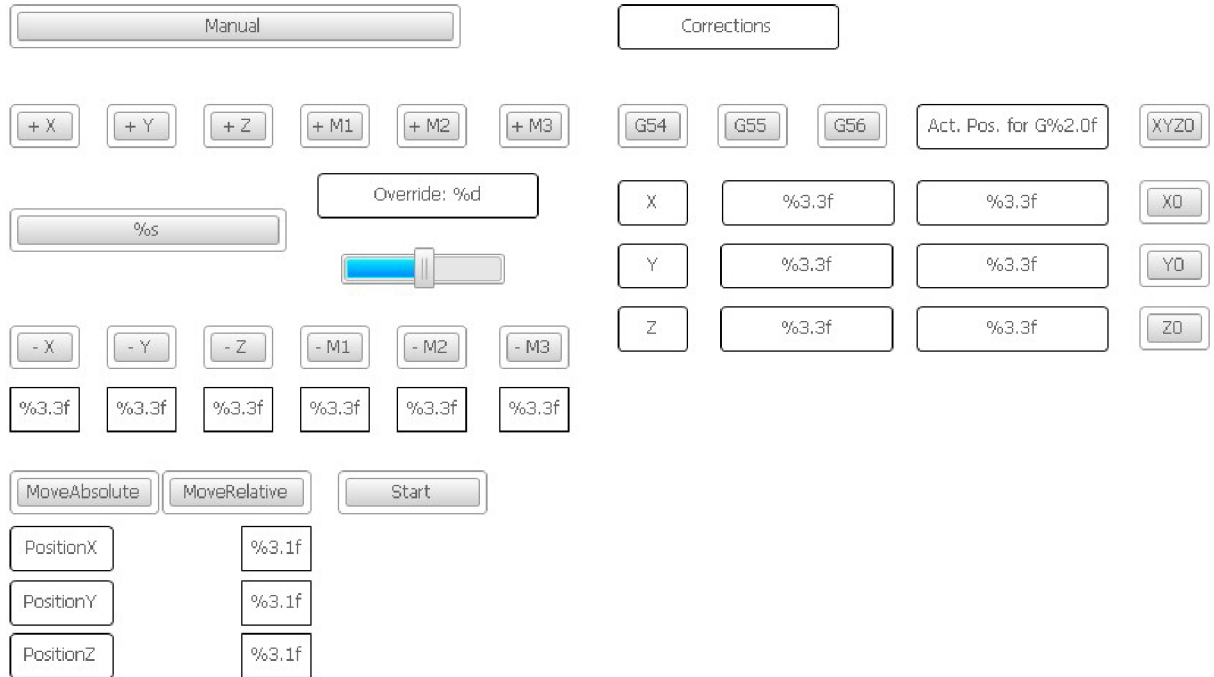
Na panelu „Settings“ se referují osy, zapínají pohony os, resetují osy, nastavuje override, zapíná a vypíná kinematickou transformaci, nastavuje „BufferMode“ a je zde také funkce „Home“. Všechny tyto funkce již byly popsány výše a jsou realizovány pomocí konkrétních programů a funkčních bloků, které jsou volány v těchto programech. Realizace panelu „Settings“ je znázorněna na následujícím obrázku.



Obr. 4-13 - Vizualizace Settings

Manual

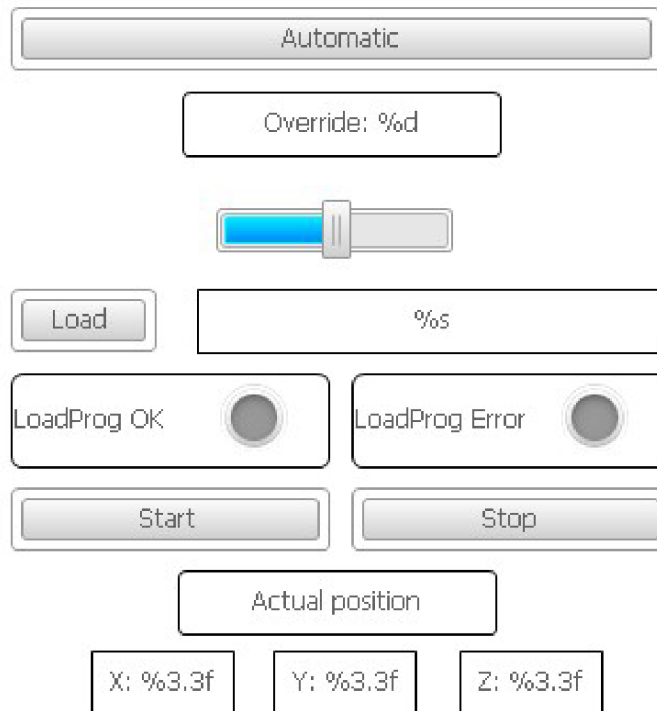
Na panelu „Manual“ se ručně ovládá pohyb os X, Y, Z, M1, M2 a M3, zobrazuje se zde jejich aktuální poloha, přepíná se mezi režimem FAST a SLOW, dá se zde také nastavit override, ovládá se zde MoveAbsolute a MoveRelative a také se zde nastavují korekce pomocí tlačítek G54, G55 a G56. Všechny tyto funkce již byly popsány výše a jsou realizovány pomocí konkrétních programů a funkčních bloků, které jsou volány v těchto programech. Realizace panelu „Manual“ je znázorněna na následujícím obrázku.



Obr. 4-14 - Vizualizace Manual

Automatic

Na panelu „Automatic“ se zapíná 3D interpolační skupina, nastavuje se override, načítá a spouští se program a zobrazuje se aktuální poloha os X, Y a Z. Všechny tyto funkce již byly popsány výše a jsou realizovány pomocí konkrétních programů a funkčních bloků, které jsou volány v těchto programech. Realizace panelu „Automatic“ je znázorněna na následujícím obrázku.



Obr. 4-15 - Vizualizace Automatic

5 Diskuze

Vytvořený software a HMI rozhraní jsou plně funkční a splňují cíle dané zadáním této diplomové práce. Všechny funkce byly softwarově otestovány a jsou připraveny i na testování na robotu. Výjimku tvoří pouze funkce „Homing“, kterou je zatím možné realizovat jen softwarově z důvodu chybějících referenčních spínačů.

Jako vhodné zlepšení celého systému by byla právě instalace referenčních spínačů, které by mohly být využity pro automatické referování os při spuštění celého systému a pro již zmíněnou funkci „Homing“. Tato funkce je již softwarově připravená a v případě instalace spínačů ji stačí poté pouze propojit s hardwarem.

Další prostor pro zlepšení je přidání čtvrtého motoru pro pohon čtvrté osy, která by otáčela koncovým efektem kolem vlastní osy. Hardwarová konfigurace tuto možnost umožňuje.

6 Závěr

Cílem této diplomové práce bylo vytvoření softwaru pro řízení delta robotu, popis hardwaru a návrh HMI.

Pro lepší pochopení dané problematiky se tato práce ze začátku věnuje řešerši v oblasti robotiky. Nejprve je definován pojem robot, rozdělení robotů a popis kinematických struktur robotů. Poté je srovnání sériových a paralelních struktur, do kterých spadá právě i delta robot a dále se práce věnuje už jen delta robotu, popisu jeho vlastností, konstrukci a kinematice.

V praktické části se práce věnuje popisu hardwaru, tedy delta robotu, pro který bude vytvářen software. Poté se věnuje základní teorii k PLC programování a tvorbě vlastního PLC programu. Na konci práce je popsáno vytvořené HMI rozhraní a návrh testování hotového softwaru na delta robotu.

7 Seznam použitých zdrojů

- [1] CHVÁLA, Břetislav, R. MATIČKA a J. TALÁCKO. *Průmyslové roboty a manipulátory*. Praha: SNTL, 1990, 275 s. ISBN 80-03-00361-X.
- [2] SKAŘUPA, Jiří. *Průmyslové roboty a manipulátory učební text*. Ostrava: VŠB, 2007, 260 s. ISBN 978-80-248-1522-0.
- [3] MathWorks. *Creating a Stewart Platform Model Using SimMechanics* [online]. 2002 [cit. 2018-07-18]. Dostupné z: <https://www.mathworks.com/company/newsletters/articles/creating-a-stewart-platform-model-using-simmechanics.html>
- [4] Clavel, R. Une nouvelle structure de manipulation parallèle pour la robotique légère. R.A.I.R.O. 1986, APII, Vol 23, N° 6.
- [5] KOLÍBAL, Zdeněk. *Roboty a robotizované výrobní technologie*. Brno: VUTIUM, 2016. ISBN 978-8-214-4828-5.
- [6] Fanuc. *Řada delta robotů Fanuc M-3* [online]. [cit. 2018-07-20]. Dostupné z: <https://www.fanuc.eu/cz/cs/roboty/str%C3%A1nka-filtru-robot%C5%AF/%C5%99ada-m3>
- [7] ABB. *IRB 360 - Průmyslové roboty* [online]. [cit. 2018-07-20]. Dostupné z: <https://new.abb.com/products/robotics/cs/prumyslove-roboty/irb-360>
- [8] Omron. *X-Delta 3+1* [online]. [cit. 2018-07-20]. Dostupné z: <https://industrial.omron.cz/cs/products/x-delta-3+1>
- [9] Neugebauer, R., Wittstock, V. and Drossel, W. “Werkzeugmaschinen-Mechatronik“, Arbeitsblätter. Chemnitz: TU-Chemnitz, 2010
- [10] Springer Handbook of Robotics. *Industrial Robot: An International Journal* [online]. Emerald Group Publishing Limited, 2009, **36**(4) [cit. 2018-08-03]. DOI: 10.1108/ir.2009.04936dae.001. ISSN 0143-991X.
- [11] LIU, X-j, J WANG a H ZHENG. Workspace atlases for the computer aided design of the Delta robot. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* [online]. London, England: SAGE Publications, 2003, **217**(8), 861-869 [cit. 2018-08-07]. DOI: 10.1243/095440603322310413. ISSN 0954-4062.
- [12] MAYA, Mauro, Eduardo CASTILLO, Alberto LOMELÍ, Emilio GONZÁLEZ-GALVÁN a Antonio CÁRDENAS. Workspace and Payload-Capacity of a New Reconfigurable Delta Parallel Robot. *International Journal of Advanced Robotic Systems* [online]. London, England: SAGE Publications, 2013, **10**(1) [cit. 2018-08-07]. DOI: 10.5772/54670.
- [13] KOZUBÍK, Jiří. *Experimentální robotizované pracoviště s delta-robotem*. 1. Brno: Vysoké učení technické v Brně. Fakulta strojního inženýrství, 2011.
- [14] TrossenRobotics. *Delta Robot Kinematics* [online]. [cit. 2018-08-07]. Dostupné z: <http://forums.trossenrobotics.com/tutorials/introduction-129/delta-robot-kinematics-3276/>

- [15] Beckhoff. *Beckhoff Information System* [online]. [cit. 2019-01-20]. Dostupné z: <https://infosys.beckhoff.com/>
- [16] PLCOpen. *Motion Control* [online]. [cit. 2019-01-19]. Dostupné z: <https://www.plcopen.org/technical-activities/motion-control>
- [17] Beckhoff. *AM3041* [online]. [cit. 2019-01-25]. Dostupné z: https://download.beckhoff.com/download/Document/Catalog/Main_Catalog/english/separate-pages/Drive_Technology/am3041.pdf
- [18] Beckhoff. *AG2200* [online]. [cit. 2019-01-25]. Dostupné z: https://download.beckhoff.com/download/document/motion/ag2200_ba_en.pdf
- [19] Beckhoff. *AX52xx* [online]. [cit. 2019-01-25]. Dostupné z: https://download.beckhoff.com/download/Document/Catalog/Main_Catalog/english/separate-pages/Drive_Technology/AX52xx.pdf
- [20] Beckhoff. *C6015-0010* [online]. [cit. 2019-01-25]. Dostupné z: <https://infosys.beckhoff.com/content/1033/ipcinfosys/PDF/C6015-0010.pdf>
- [21] Beckhoff. *EL1008* [online]. [cit. 2019-01-25]. Dostupné z: https://download.beckhoff.com/download/Document/Catalog/Main_Catalog/english/separate-pages/EtherCAT/EL1008.pdf
- [22] Beckhoff. *EL2008* [online]. [cit. 2019-01-25]. Dostupné z: https://download.beckhoff.com/download/Document/Catalog/Main_Catalog/english/separate-pages/EtherCAT/EL2008.pdf
- [23] GOUBEJ, Martin, Martin ŠVEJDA a Miloš SCHLEGEL. *Úvod do mechatroniky, robotiky a systémů řízení pohybu*. Plzeň, 2012. Skriptum. ZČU.

8 Seznam obrázků

Obr. 3-1 – Sériové kinematické struktury [2]	18
Obr. 3-2 – Paralelní kinematické struktury	18
Obr. 3-3 – Komerčně vyráběné delta roboty	20
Obr. 3-4 - Konstrukce delta robotu a souřadnicové systémy [9]	20
Obr. 3-5 – Řešení prostoru delta robotu [11]	22
Obr. 3-6 – Tvary pracovního prostoru delta robotu [11]	23
Obr. 3-7 – Řešení úlohy přímé kinematiky [12]	23
Obr. 3-8 – Řešení úlohy inverzní kinematiky [13, 14]	25
Obr. 3-9 – Souřadnicové systémy [16]	26
Obr. 3-10 – Skupiny os [16].....	27
Obr. 3-11 – Move Linear [16].....	29
Obr. 3-12 – Move Cirkular [16].....	30
Obr. 3-13 – Průběh trajektorie a rychlosti [16].....	31
Obr. 3-14 – Definice pojmů [16].....	32
Obr. 3-15 – Buffer Mode [16].....	33
Obr. 3-16 – Transition Mode [16].....	34
Obr. 4-1 – Delta robot	35
Obr. 4-2 – Hardware	36
Obr. 4-3 - Hardware schéma	37
Obr. 4-4 - Skenování hardwaru - motory	38
Obr. 4-5 - Reálné osy	38
Obr. 4-6 - Virtuální osy	39
Obr. 4-7 – Interpolace	39
Obr. 4-8 - Kinematická transformace.....	40
Obr. 4-9 - Delta Type 1 [15]	40
Obr. 4-10 - Parametry kinematické transformace	41
Obr. 4-11 - Funkční bloky knihovny Tc2_Mc2 [15]	43
Obr. 4-12 - Struktura softwaru	45
Obr. 4-13 - Vizualizace Settings	48
Obr. 4-14 - Vizualizace Manual	49
Obr. 4-15 - Vizualizace Automatic	50