



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií ■

# ANALÝZA A PROGRAMOVÁNÍ BUŇKY SVAŘOVACÍHO ROBOTA

**Diplomová práce**

*Studijní program:* N2612 – Elektrotechnika a informatika

*Studijní obor:* 3906T001 – Mechatronika

*Autor práce:* **Bc. Marek Jelínek**

*Vedoucí práce:* Ing. Josef Černožorský, Ph.D.

---

Liberec 2015







TECHNICAL UNIVERSITY OF LIBEREC  
Faculty of Mechatronics, Informatics  
and Interdisciplinary Studies ■

# ANALYZING AND PROGRAMMING OF WELDING ROBOT

**Diploma thesis**

*Study programme:* N2612 – Electrical Engineering and Informatics

*Study branch:* 3906T001 – Mechatronics

*Author:* **Bc. Marek Jelínek**

*Supervisor:* Ing. Josef Černožský, Ph.D.





## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: Bc. Marek Jelínek  
Osobní číslo: M13000219  
Studijní program: N2612 Elektrotechnika a informatika  
Studijní obor: Mechatronika  
Název tématu: Analýza a programování buňky svařovacího robota  
Zadávací katedra: Ústav mechatroniky a technické informatiky

### Zásady pro vypracování:

1. Seznamte se s možnostmi průmyslového robota Fanuk a programovacími postupy.
2. Proveďte rešerši obvyklých datových formátů používaných pro výměnu dat mezi CAD programy a robotickými systémy.
3. Navrhněte vhodné postupy pro digitalizaci robotické buňky, pro tvorbu knihovny nástrojů a pracovních objektů.
4. Zhodnoťte možnosti přímého přechodu do online režimu a navrhněte postupy pro adjustaci modelu.

Rozsah grafických prací: dle potřeby dokumentace  
Rozsah pracovní zprávy: 40–50 stran  
Forma zpracování diplomové práce: tištěná/elektronická  
Seznam odborné literatury:

- [1] Šolc, F., Žalud, L.: Základy robotiky. VUT v Brně, elektronická skripta AMT010, 2002.
- [2] Skařupa, J.: Průmyslové roboty a manipulátory. Ediční středisko VŠB-TUO, 2007, ISBN 978-80-248-1522-0.

Vedoucí diplomové práce: **Ing. Josef Černožorský, Ph.D.**  
Ústav mechatroniky a technické informatiky  
Konzultant diplomové práce: **Ing. Pavel Zezula, Ph.D.**  
Škoda Auto a.s.  
Datum zadání diplomové práce: **10. října 2014**  
Termín odevzdání diplomové práce: **15. května 2015**

  
prof. Ing. Václav Kopecký, CSc.  
děkan



  
doc. Ing. Milan Kolář, CSc.  
vedoucí ústavu

V Liberci dne 10. října 2014

## Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 13.5.2015

Podpis: 

## **Anotace**

Tato práce se zabývá Analýzou a programováním buňky reálného svařovacího robota FANUC M-20iA. Je zde nastíněna základní rešerše v oblasti robotiky a CAD systému, která by měla uvést čtenáře do základů dané problematiky. V další části práce je vytvořen v prostředí ROBOGUIDE model reálného pracoviště, na kterém byl realizován program, který byl i prakticky implementován do reálného robota a následně model validován.

## **Annotation**

This project concerns the analysis and programming of real cells with FANUC M-20iA robots. Basic research into robotics and CAD systems is outlined herein, which should allow the reader to understand the basis of the associated problems. In the next section, a program is developed using a model of a real workstation is created in ROBOGUIDE. This program was also implemented in a real robot and subsequently proved the accuracy of the model with a real robotic cell.



## **Abstrakt**

V této práci je nejdříve představen stav robotiky u ŠKODA AUTO a.s. a v krátkosti též stav průmyslové robotiky celkově. Představení CAD produktů, výrobců robotů a třetích poskytovatelů je vedeno s ohledem na téma této práce, která se omezuje na nejrozšířenější kategorii průmyslových robotů.

Vzhledem k zadání a technickým možnostem padla volba na produkty firmy FANUC, jejíž produkty jsou pro tuto práci k dispozici.

Modelovat se bude reálná robotová instalace, kterou v tomto případě bude představovat školící cela s roboty FANUC v robotické laboratoři ŠKODA ACADEMY. Postup bude odpovídat tzv. integraci, kdy se již existující robotické pracoviště digitalizuje.

Takto vzniklý model se zkalibruje s reálným pracovištěm. Kalibrovat se bude hlavně pracovní plocha na oplocence, a ostatních detailech linky v našem případě až tak nezáleží. To vše bude sloužit k tomu, aby bylo možné v OFF-LINE prostředí ROBOGUIDE vytvořit výrobní program a tento následně bez problému přenést do skutečného robota výše zmiňovaného pracoviště.

Výrobní program se bude tvořit importem obrobku z CADu. Zde budou probrány možné formáty importovaných dat. Práce na obrobku se nejdříve odsimuluje z hlediska dosažitelnosti robotem v programu ROBOGUIDE. Provede se též odhad trvání pracovního cyklu tak, jak se tomu děje při návrhu reálné výrobní linky nejen pro automobilky.

Následně se se takto vzniklý program přeneseme jedním z možných způsobů do instalovaného robota a odzkouší.

Výsledky této práce se použijí pro demonstrační účely, výuku, a další studium laboratoře robotiky ŠKODA ACADEMY.

## **Klíčová slova**

Průmyslový robot, OLP, ROBOGUIDE, CAD

## **Abstract**

This study introduces the state of robotics of Skoda Auto a.s. and briefly describes the state of industrial robotics overall. The introduction of CAD products, robot manufacturers and other providers has been drawn up with the respect to this study assignment which focuses on the most widespread category of industrial robots.

Due to the assignment and technical possibilities, the FANUC robot was chosen for practical part of this project.

A real robot installation will be modelled, which will be a training cell with FANUC robots in the robotics laboratory of the Skoda Academy. This process will be integrated when existing robotic workstations are digitalised.

The resulting model is calibrated with a real workplace. Mainly the working area and boundaries will be calibrated, the other details of the line in our case do not matter. All of this will serve to allow manufacturing programs to be created in the offline Roboguide environment and to then be transferred to the aforementioned workplace without problems.

The manufacturing program will be formed from an imported CAD workpiece. Possible import data formats will be discussed. The first part of the simulation will be regarding the accessibility of the workpiece for the robot in the ROBOGUIDE program. An estimation of the cycle time will also be generated as is done for real production lines not only in the auto industry.

Subsequently, the program provides one possible way to install and test the robot.

The results of this project will be used for further education in Skoda Academy robotics lab and will be used for other case studies of FANUC M-20iA robot.

## **Keywords**

Industrial robot, FANUC, OLP, ROBOGUIDE, CAD

## **Poděkování**

Tímto bych rád poděkoval panu Ing. Josefovi Černoorskému, Ph.D. za jeho cenné rady ohledně tvorby této práce a rovněž panu Ing. Pavlovi Zedulovi, Ph.D., který mě ve firmě ŠKODA AUTO vždy správně nasměroval.

Nemalé poděkování patří rovněž i zaměstnancům robotických laboratoří a panu Baumrukovi za poskytnutí studijních materiálů, přístupu k robotům a za jejich trpělivost.

## Obsah

Anotace.....	4
Annotation .....	4
Abstrakt .....	5
Klíčová slova .....	5
Abstract.....	6
Keywords.....	6
Poděkování .....	7
Obsah .....	8
Seznam zkratk.....	11
Seznam obrázků.....	12
Seznam tabulek.....	13
Úvod .....	14
Obsahem této práce je .....	14
Obsahem práce není .....	15
1    Možnosti a programovací postupy robota FANUC.....	16
1.1    Datové formáty uložených poloh v průmyslových robotech FANUC .....	16
1.1.1    Základní (nativní) datový formát .....	16
1.1.2    Kartézské souřadnice .....	17
1.1.3    Druhy singularit jednotlivých os .....	18
1.2    Úvod do problematiky řízení robotů.....	21
1.2.1    Rotace a translace souřadných systémů za pomoci Eulerových úhlů .....	21
1.2.2    Denavit-Hartenbergova koncepce .....	22
1.3    Robot FANUC M-20iA .....	23
1.3.1    Mechanická jednotka FANUC M-20iA .....	23
1.3.2    Kontrolér .....	25
1.3.3    Uživatelský panel .....	27
1.4    Pohybové příkazy průmyslových robotů .....	28
1.4.1    Rychlopřejezd .....	29
1.4.2    Technologické pohyby .....	30
1.5    Logické příkazy (SPS).....	32
1.5.1    Vstupy/Výstupy.....	32
1.5.2    Vnitřní paměti .....	33
1.6    Řízení a větvení programu.....	33
1.6.1    Bezpodmínečné větvení programu .....	33
1.6.2    Podmíněné větvení programu.....	34
1.7    OFF-LINE programování (OLP).....	35
1.8    Generování trajektorie podle hran importovaných objektů .....	36
2    Rešerše CAD systémů .....	37
2.1    Historie .....	38
2.2    AUTOCAD.....	39
2.3    Catia.....	40

2.4	Creo Parametric .....	40
2.5	AUTODESK Inventor Professional.....	40
2.6	DraftSight .....	40
2.7	SOLIDWORKS .....	41
2.8	FreeCad.....	41
2.9	Google SketchUp.....	41
2.10	Další možnosti importu dat do prostředí ROBOGUIDE .....	42
2.10.1	3D Studio MAX .....	42
2.10.2	Process Simulate .....	42
2.11	CAD produkty .....	45
2.12	Formáty CAD dat .....	45
2.12.1	Formát STL .....	45
2.12.2	Formát IGES .....	46
2.12.3	Formát STEP.....	46
2.12.4	Formát DXF .....	46
2.12.5	Formát DWG.....	47
2.13	Výhody používání CAD/CAM .....	47
2.14	Porovnání nejpoužívanějších průmyslových CAD systémů.....	47
2.14.1	CATIA V5 R21 .....	47
2.14.2	AUTODESK Inventor Professional .....	47
2.14.3	SolidWorks 2009.....	48
3	Digitalizace robotické buňky pro tvorbu knihovny nástrojů a pracovních objektů .....	49
3.1	ROBOGUIDE a CAD.....	49
3.2	Zaměření nástroje – TOOL FRAME .....	52
3.3	Zaměření souřadného systému – USER FRAMES / EXTERN TOOL.....	53
3.4	Založení nového projektu .....	53
3.5	Vytvoření modelu vývojového pracoviště.....	54
3.6	Cellbrowser.....	55
3.7	Reálný dosah robota .....	57
3.8	Model reálného pracoviště.....	58
3.9	Modelování koncového nástroje TPC.....	60
4	Adjustace modelu .....	62
4.1	Adjustace výrobního stolu v software ROBOGUIDE.....	62
4.2	Kontrola kolizních stavů.....	62
4.3	Kontrola kolizí v softwaru ROBOGUIDE.....	62
4.4	Kalibrace pracovní buňky.....	63
4.4.1	Obecné informace .....	63
4.4.2	Postup pro kalibraci objektů.....	63
4.4.3	Postup pro kalibraci součástky (Part) umístěné v přípravku (Fixture).....	68
4.4.4	Postup pro kalibraci metodou přímého zadání Offsetu .....	69
4.5	Adjustace modelu .....	69
4.6	Vygenerování dráhy podle CAD objektu .....	74
4.7	Praktický výstup práce.....	75
5	Závěr.....	78
6	Literatura.....	79

Příloha A:	Podporované typy souborů v prostředí ROBOGUIDE .....	81
Příloha B:	Ovládání prostředí ROBOGUIDE .....	82
Příloha C:	Hlavní program FOLGE099 .....	83
Příloha D:	Podprogram UP6_3 „ŠKODA LOGO“ .....	85

## Seznam zkratek

<b>Zkratka</b>	<b>Celý název</b>	<b>Překlad</b>
TCP	Tool Center Point	Nositel souřadnice nástroje
POV	Program Override	Programová jízda
HOV	Hand Override	Ruční jízda
OLP	OFF-LINE Programming	OFF-LINE programování
IKÚ	Inverzní kinematická úloha	Inverzní kinematická úloha
PKÚ	Přímá kinematická úloha	Přímá kinematická úloha
MIG	Metal Inert Gas	Obloukové svařování v ochranné atmosféře
MAG	Metal Activ Gas	Obloukové svařování s aktivními plyny
CAD	Computer Aided Design	Počítačem podporované projektování
CAM	Computer Aided Manufacturing	Počítačová podpora obrábění
$J_n$	Vektor úhlů ramen	Vektor úhlů ramen
W, P, R	Orientace robota	Orientace robota
X, Y, Z	Nositel souřadnice robota	Nositel souřadnice robota

## Seznam obrázků

Obr. 1:	Kinematický řetězec průmyslového robota FANUC.....	16
Obr. 2:	Zobrazení polohy jednotlivých os na uživatelském terminálu .....	17
Obr. 3:	Zobrazení kartézských souřadnic TCP na uživatelském panelu.....	17
Obr. 4:	Základní FRAME robota FANUC.....	18
Obr. 5:	Singularita osy J5.....	19
Obr. 6:	Singularita osy J3.....	19
Obr. 7:	Singularita os J1 a J6 .....	20
Obr. 8:	Robot FANUC M-20iA v prostředí ROBOGUIDE [23].....	24
Obr. 9:	Výrobní štítek robota FANUC M-20iA [23] .....	25
Obr. 10:	Výrobní štítek kontroléru robota FANUC M-20iA [23] .....	26
Obr. 11:	Popis kontroléru [23].....	26
Obr. 12:	Popis uživatelského panelu [23].....	27
Obr. 13:	Graf interpolací pohybů PTP a LIN .....	28
Obr. 14:	Rychlosti jednotlivých os při synchronizovaném PTP [23].....	29
Obr. 15:	PTP pohyb.....	29
Obr. 16:	Programovací příkaz LIN pohybu.....	30
Obr. 17:	Programovací příkaz CIRC pohybu .....	31
Obr. 18:	Příklady generovaných pohybů.....	31
Obr. 19:	Datové formáty pro CAD a OLP.....	44
Obr. 20:	OFF-LINE programování.....	50
Obr. 21:	Komunikace mezi CAD/CAM a OLP.....	51
Obr. 22:	Průběh OFF-LINE programování .....	52
Obr. 23:	Průvodce vytvořením nového projektu .....	54
Obr. 24:	Panel nástrojů prostředí ROBOGUIDE [23].....	55
Obr. 25:	Cellbrowser [23].....	56
Obr. 26:	Obálka dosahu robota ve vývojového prostředí ROBOGUIDE [23].....	58
Obr. 27:	Robotická cela ŠKODA ACADEMY [23] .....	59
Obr. 28:	Model robotické cely.....	59
Obr. 29:	Reálný nástroj robota FANUC M-20iA .....	60
Obr. 30:	Modelovaný nástroj v softwaru FreeCad .....	61
Obr. 31:	Naimportovaný nástroj do prostředí ROBOGUIDE .....	61
Obr. 32:	Nastavení UFRAME .....	64
Obr. 33:	Kalibrace objektů .....	65
Obr. 34:	3 kalibrační kroky .....	66
Obr. 35:	Výsledek kalibrace .....	67
Obr. 36:	Výsledky kalibrace.....	68
Obr. 37:	Nastavení FRAME robota.....	70
Obr. 38:	Konfigurace pohybu.....	71
Obr. 39:	Konfigurace pohybu.....	72
Obr. 40:	Konfigurace přístupového a koncové bodu.....	73
Obr. 41:	Nastavení offsetů.....	74
Obr. 42:	Návrh loga v softwaru FreeCad .....	76
Obr. 43:	Naimportované ŠKODA logo v prostředí ROBOGUIDE .....	76



## Seznam tabulek

Tab.1:	Tabulka rozsahů jednotlivých os. (Závisí na HW a SW dorazech robota) .....	18
Tab.2:	Parametry jednotlivých os. ....	25
Tab.3:	Logické operátory [20] .....	32
Tab.4:	Podporované formáty pro prostředí ROBOGUIDE.....	81
Tab.5:	Ovládací zkratky prostředí ROBOGUIDE .....	82

## Úvod

Velké výrobní závody a automobilový průmysl zvláště, plánují výrobu na mnoho let dopředu s požadavky na co nejvyšší efektivitu. Při očekávaném zásahu do výroby, jako je zavedení produkce nového výrobku, změně výrobní technologie, rekonstrukce výrobních prostor, rozšiřování či přesouvání výroby se nová produkční linka plánuje v souladu s nasazovanou technologií dlouho před samotným zásahem do stávající výroby.

Výsledkem takového plánování jsou dnes simulační modely budoucích linek, které slouží k určování taktovacích časů, výrobní kapacity, množství a typu technologie, kterou bude třeba nasadit, energetické náročnosti a proveditelnosti vůbec. Výhody takového postupu jsou zřejmé. Je možné bez zásahu do linky samotné provádět neomezeně cyklů pokus/oprava. V případě robotů lze takovýto simulační prostředí využít též k tzv. OFF-LINE programování (OLP), kdy samotné testovací procesy zpracovávané virtuálními roboty, jsou ve finále exportovány a použitelné jako výchozí programy pro reálné roboty ve výrobních linkách. Jinak by bylo možné tvořit takové programy s nemalým časovým nákladem až v lince samotné. Lze tedy říci, že nepostižená produkce staré a tvorba linky nové jsou z části paralelní procesy, ačkoliv se jedná většinou o zástavbu do stejného prostoru. Výpadek produkce se tedy omezí pouze na nezbytně krátkou dobu nutnou k instalaci nových výrobních prostředků, zasíťování a ověření importovaných programů. Že se čas poskytování k přestavbě linky neustále zkracuje, není nutné připomínat. Simulace a OLP se tedy stává nutnými podmínkami při a pro návrh linky, na jejichž přesnosti se do budoucna budou klást stále větší nároky.

Zde je třeba též zmínit, že výstup ze simulací a OFF-LINE programy pro import do robotů v lince jsou použitelné jen do té míry, do které model, na kterém byly prováděny, odpovídal realitě. Zvláště se toto projevuje v případě tzv. integrací do linek, kdy to stávající výroby se technologie přidává. Např. pokud se pouze rozšiřuje sortiment výrobků. Zde se musí postupovat zvláště obezřetně, neboť jsme omezeni již stávajícím vybavením a cyklem linky. Získání přesných dat o topologii linky je tedy esenciální.

Obor robotiky se neustále rozvíjí a simulace se stávají detailnějšími a náročnějšími. Vývojových prostředí je dost, jejich možnosti se ale liší, z části v závislosti na podpoře od výrobců technologií. Výrobci robotů nabízejí téměř vždy vlastní simulační prostředí, které poskytuje věrnější chování vlastních robotů. S rozšířenějšími simulačními produkty třetích stran si tak vlastně konkurují.

I z těchto důvodů je nutné mít dané problematice dostatečně hluboké znalosti a dovednosti, aby bylo možné výsledky interpretovat a popřípadě korigovat.

## Obsahem této práce je

Popsání aktuálního stavu robotizace v automobilce ŠKODA AUTO a.s. A to včetně prostředí CAD používaných pro OFF-LINE programování a přípravu robotů.

Následně jsem se z celé škály omezil na roboty firmy FANUC, které jsou počtu 500 ks aktuálně používané na projektech OCTAVIA, RAPID, FABIA.

K robotům ve výrobě samotné byly instalovány i tréninkové roboty v prostorách ŠKODA ACADEMY, které se využívají ke školení personálu výroby údržby a

projektantů linek. K těmto robotům mi byl udělen přístup a budou použity k vývoji a testování v rámci této práce.

Tyto roboty doplňuje i SW výrobcem poskytované vybavení ROBOGUIDE. Jedná se o OFF-LINE programovací prostředí s virtuálním robotem (roboty). V tomto prostředí se bude vytvářet model robotického pracoviště podle reálné předlohy a následně se použije pro vytváření a testování výrobních programů.

Důvodem pro volbu právě těchto produktů byla jejich dostupnost a nabízené možnosti, o kterých se v této práci zmíním.

Demonstrovat se budou v zásadě věci dvě. Modelování pracoviště a jeho kalibraci s reálným pracovištěm. Následně pak proces CAD model – OFF-LINE prostředí – program pro robota – výroba na skutečném robotu.

### **Obsahem práce není**

Kompletní popis a implementace řešení na všech produktech aktuálně používaných v průmyslu, což je samo o sobě nad rámec této práce.

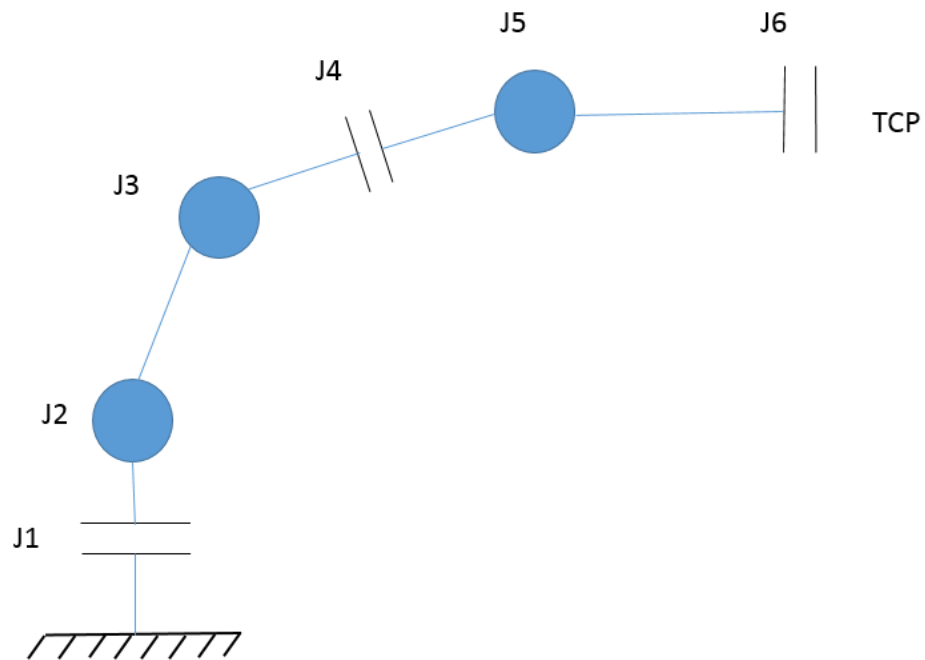
Tato práce se rovněž nezabývá simulací na jiné technologii než na samotném robotu. Modely jiných technologií jako jsou lepičky, MIG/MAG svářečky atd. zde zapracovány nebudou.

Rovněž se zde nerozebírá import veškerých formátů a verzí, které produkty deklarují ovládat.

## 1 Možnosti a programovací postupy robota FANUC

Následující kapitoly používají terminologii firmy FANUC. Jsou zde nastíněny základní principy a chování robotů obecně.

### 1.1 Datové formáty uložených poloh v průmyslových robotech FANUC

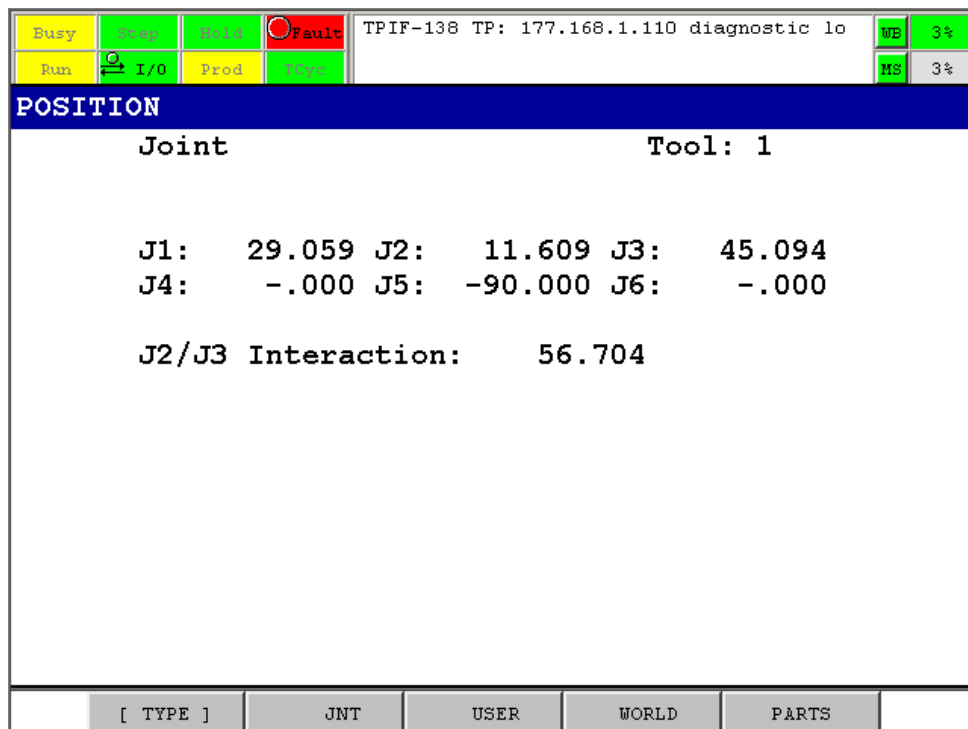


Obr. 1: Kinematický řetězec průmyslového robota FANUC

#### 1.1.1 Základní (nativní) datový formát

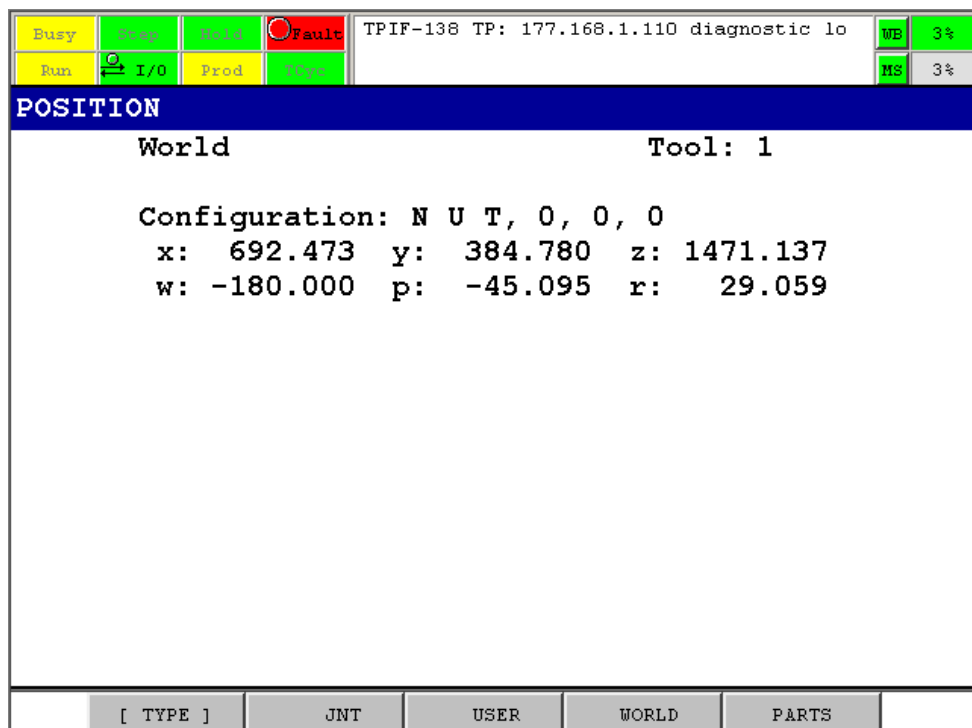
$$Jn = (J1, J2, J3, J4, J5, J6) \quad (1.1)$$

Je vektor úhlů ramen, které reprezentují natočení jednotlivých os. Používané jednotky jsou stupně. Pro výpočet kinematiky jsou k tomu nutné prvotní seřízení robota tzv. kalibrace nebo justáž a parametry ze strojních dat, což je ekvivalent elektronického katalogového listu každé mechanické jednotky. Tyto parametry obsahují délky jednotlivých ramen a další údaje nutné k propočtu kinematiky.



Obr. 2: Zobrazení polohy jednotlivých os na uživatelském terminálu

### 1.1.2 Kartézské souřadnice



Obr. 3: Zobrazení kartézských souřadnic TCP na uživatelském panelu

## Základní FRAME robota FANUC

$$\left\{ \underbrace{X, Y, Z}_{TCP}, \underbrace{W, P, R}_{orientace} \right\}$$

Obr. 4: Základní FRAME robota FANUC

TCP (Obr. 4:) neboli koncový pracovní bod robota se udává v milimetrech a počítán je od nuly prostoru. Pokud robot končí přírubou šesté osy, na kterou zákazník nepřipevnil žádný svůj nástroj, tak je TCP od výroby umístěn ve středu této příruby. Takové TCP je potom výchozí pro ostatní zákaznické TCP.

Orientace (Obr. 4:) určuje natočení TCP vůči souřadnému systému robota.

CONFIGURATION obsahuje relativní polohy ramen robota ve vztahu k singularitám (tzv. figuru) a dále počet celých otáček těch os, které nejsou omezeny na interval jedné otáčky tj. J1, J4, J6. U osy J1 se jedná o rozsah  $\pm 185^\circ$ . Osy J4 a J6 jsou osy na zápěstí robota, které jsou konstrukčně řešeny jako nekonečné převody. V praxi je omezuje pouze vnitřní či vnější kabeláž a softwarové dorazy.

Hodnoty celých otáček jsou reprezentovány v Tab.1: . Např. je možné pozorovat to, že pokud se pohybujeme pouze v základním rozsahu tj.  $\pm 180^\circ$  tak indikujeme na uživatelském panelu hodnotu 0, jsme tedy v základním rozsahu dané osy. Pokud bychom se přetočili např. v kladném směru otáčení o dalších  $360^\circ$ , tak si můžeme si povšimnout toho, že v tabulce indikujeme hodnotu 1, ale na samotných ramenech mechanické jednotky toto protočení poznat není.

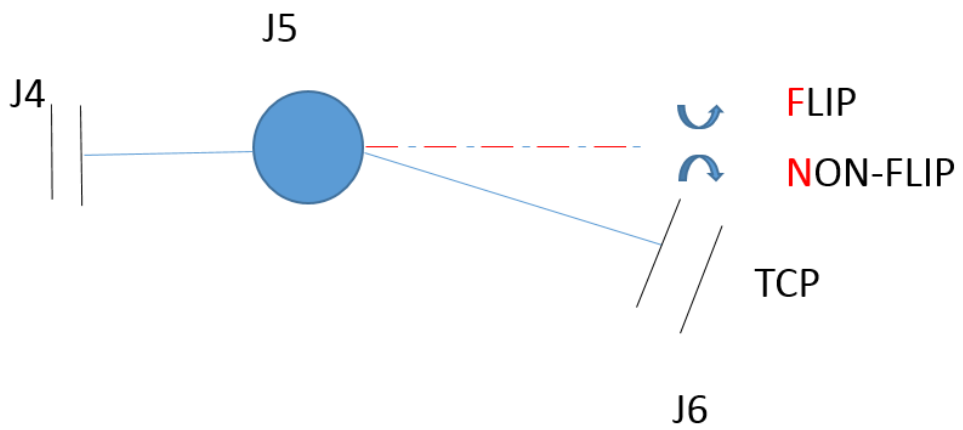
Tab.1: Tabulka rozsahů jednotlivých os.  
(Závisí na HW a SW dorazech robota)

Rozsah		Počet celých otáček		
OD[°]	DO[°]	J1	J2	J3
-900	-541	-2	-2	-2
-540	-181	-1	-1	-1
-180	180	0	0	0
181	540	1	1	1
541	900	2	2	2

### 1.1.3 Druhy singularit jednotlivých os

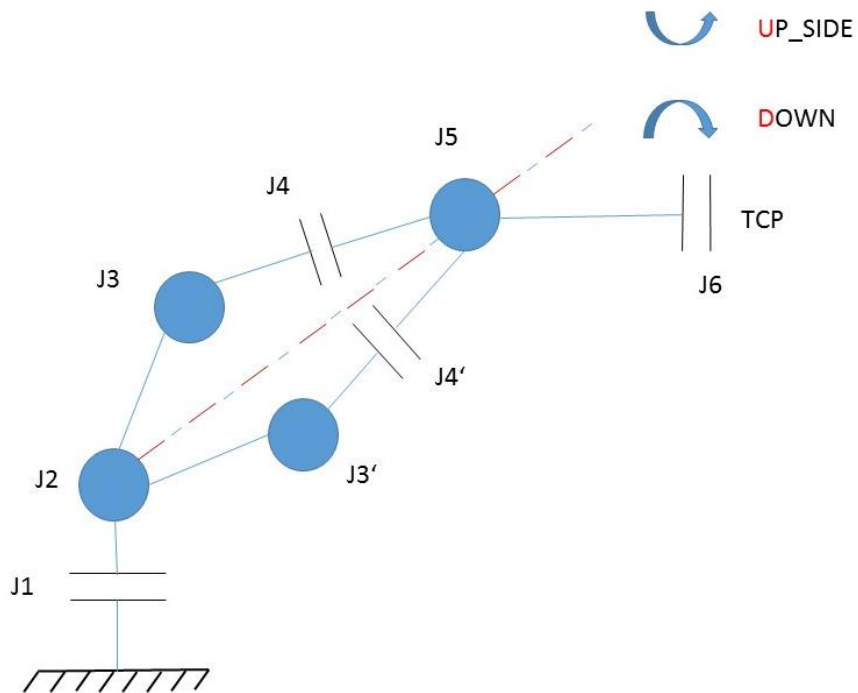
Singularita je stav, kdy nastává u robota tzv. neurčitost a to v našem případě taková, že má robot k zajištění cílového bodu několik jiných možností. Tento stav je indikován na uživatelském panelu v FRAMU robota, kde jsou tyto stavy singularit indikovány jednotlivými písmeny. Pro lepší názornost je problematika znázorněna na následujících obrázcích.

Na tomto obrázku je možné spatřit singularitu osy J5, u které mohou nastat dva stavy, a to FLIP a NON-FLIP, které jsou na uživatelském panelu značeny F a N. Tento parametr je jako jediný závislý pouze na jedné ose.



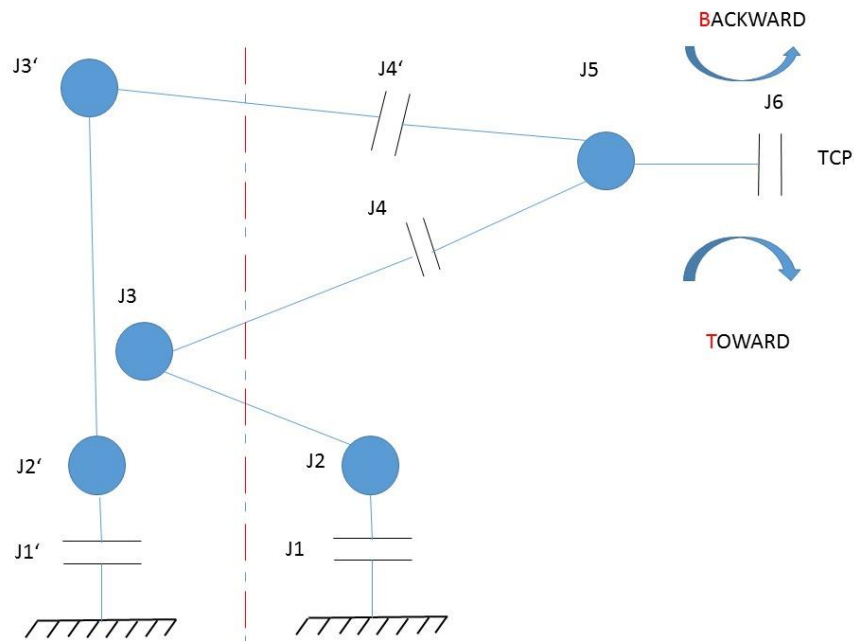
Obr. 5: Singularita osy J5

Na tomto obrázku je znázorněna singularita osy J3, která je na uživatelském panelu indikována písmeny U a D neboli UP\_SIDE a DOWN.



Obr. 6: Singularita osy J3

Na tomto obrázku je znázorněna singularita osy J1 a J6. Na uživatelském panelu je indikována písmeny B a T neboli BACKWARD a TOWARD.



Obr. 7: Singularita os J1 a J6

Standardní šestiosý průmyslový robot se třemi protínajícími se osami může teoreticky zaujmout 8 poloh, tzv. figur pro dosažení požadované souřadnice TCP. Jedná se o teoretickou možnost, která nezohledňuje omezení kabelážemi a dorazy. Pro jednoznačný výpočet IKÚ je nutné tyto možnosti uvažovat. To byla původní motivace pro zavedení parametrů CONFIGURACE.



## 1.2 Úvod do problematiky řízení robotů

### Orientace a poloha souřadných systémů

Přímá úloha je založena na znalosti jednotlivých úhlů natočení jednotlivých pohyblivých členů robota, jsme tedy schopni určit v jaké poloze a orientaci se nachází koncový bod.

Inverzní úloha určuje jednotlivé úhly natočení jednotlivých pohyblivých členů na základě pozice koncového bodu. Jedná se o opak přímé úlohy.

Obě tyto úlohy jsou vztaženy k bázi robota. V současné době se touto problematikou uživatelé příliš nezabývají, jelikož tyto složité matematické operace jsou počítány v kontroléru automaticky na základě algoritmů od výrobce.

### 1.2.1 Rotace a translace souřadných systémů za pomoci Eulerových úhlů

#### Rotace

Tyto rotace jsou prováděny za pomoci Eulerových úhlů vůči nově vzniklé ose  $(x_1, y_1, z_1)$ . Pokud uvažujeme natočení kolem osy  $z_0$  základního souřadného systému  $(x_0, y_0, z_0)$ , tak soustava rovnic natočení jednotlivých os nového souřadného systému  $(x_1, y_1, z_1)$  vůči základnímu souřadnému systému  $(x_0, y_0, z_0)$  je popsána rovnicemi (1.2), (1.3), (1.4):

$$x_0 = x_1 \cdot \cos \alpha - y_1 \cdot \sin \alpha + 0 \cdot z_1 \quad (1.2)$$

$$y_0 = x_1 \cdot \sin \alpha + y_1 \cdot \cos \alpha + 0 \cdot z_1 \quad (1.3)$$

$$z_0 = 0 \cdot x_1 + 0 \cdot y_1 + 1 \cdot z_1 \quad (1.4)$$

Pak můžeme tuto soustavu rovnic, která reprezentuje rotaci kolem osy  $z_0$  přepsat v maticovém zápisu tedy:

#### Rot $z_0$ (1.5):

$$\begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \quad (1.5)$$

Totéž můžeme provést pro rotaci kolem zbývajících os  $x_0, y_0$ :

#### Rot $x_0$ (1.6):

$$\begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \quad (1.6)$$

#### Rot $y_0$ (1.6):

$$\begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} = \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \quad (1.7)$$

Souřadnice  $x_1, y_1, z_1$  jsou odrazem nového souřadného systému, který vzniknul díky těmto rotacím.

Pokud bychom chtěli získat matici, která reprezentuje natočení kolem všech tří os tj. kolem  $x, y, z$ , tak je možné využít tzv. matici rotace. Tuto maticovou závislost, která reprezentuje natočení všech os, získáme tak, že provedeme maticový součin jednotlivých matic, které reprezentují natočení kolem jednotlivých os. Tím pádem získáme matici ve tvaru (1.9):

$$R(\alpha, \beta, \gamma) = R_z(\alpha) \cdot R_y(\beta) \cdot R_x(\gamma) \quad (1.8)$$

$$R(\alpha, \beta, \gamma) = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{pmatrix}$$

$$R(\alpha, \beta, \gamma) = \begin{pmatrix} c\alpha \cdot c\beta & c\alpha \cdot s\beta \cdot s\gamma - s\alpha \cdot c\gamma & c\alpha \cdot s\beta \cdot c\gamma + s\alpha \cdot s\gamma \\ s\alpha \cdot c\beta & s\alpha \cdot s\beta \cdot s\gamma + c\alpha \cdot c\gamma & s\alpha \cdot s\beta \cdot c\gamma - c\alpha \cdot s\gamma \\ -s\beta & c\beta \cdot s\gamma & c\beta \cdot c\gamma \end{pmatrix} \quad (1.9)$$

### Translace

Translaci neboli posunutí je možné reprezentovat za pomoci tzv. vektoru posunutí  $\vec{P}(x, y, z)$ , kde jednotlivé složky určují posun v ose  $x, y$  a  $z$ . Pak platí (1.10).

$$\vec{P}(x, y, z) = \begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} \quad (1.10)$$

Pokud bychom chtěli vyjádřit translaci v ose  $z$ , tak v maticovém tvaru platí (1.11).

$$T_{tran z} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.11)$$

### 1.2.2 Denavit-Hartenbergova koncepce

Denavit-Hartenbergova koncepce neboli také (D-H) je princip, který umožňuje obecně sjednotit libovolně posunuté a orientované souřadné systémy za pomoci čtyř základních pohybů. Tato realizace je možná po maticovém součinu těchto jednotlivých transformačních matic (rotace kolem osy  $z$ , translace podél osy  $z$ , translace podél osy  $x$  a rotací kolem osy  $x$ ), tyto pohyby lze realizovat za pomoci těchto matic homogenních transformací (1.12), (1.13), (1.14), (1.15).

$$T_{rot z} = \begin{pmatrix} \cos \alpha_i & -\sin \alpha_i & 0 & 0 \\ \sin \alpha_i & \cos \alpha_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.12)$$

$$T_{tran z} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.13)$$

$$T_{tran x} = \begin{pmatrix} 1 & 0 & 0 & b_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.14)$$

$$T_{rot x} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \gamma_i & -\sin \gamma_i & 0 \\ 0 & \sin \gamma_i & \cos \gamma_i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.15)$$

Výsledná D-H koncepce mezi sousedními souřadnými systémy vznikne vynásobením jednotlivých dílčích pohybů transformačních matic v pořadí, tak jak by byly reálně prováděny (1.16).

$$T_{D-H} = T_{rot z} \cdot T_{tran z} \cdot T_{tran x} \cdot T_{rot x}$$

$$T_{D-H} = \begin{pmatrix} \cos \alpha_i & -\sin \alpha_i \cdot \cos \gamma_i & \sin \alpha_i \cdot \sin \gamma_i & b_i \cdot \cos \alpha_i \\ \sin \alpha_i & \cos \alpha_i \cdot \cos \gamma_i & -\cos \alpha_i \cdot \sin \gamma_i & b_i \cdot \sin \alpha_i \\ 0 & \sin \gamma_i & \cos \gamma_i & a_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.16)$$

Parametry  $\alpha_i, a_i, b_i, \gamma_i$  plně reprezentují vztahy mezi sousedními souřadnými systémy spojenými rotační nebo translační vazbou [6].

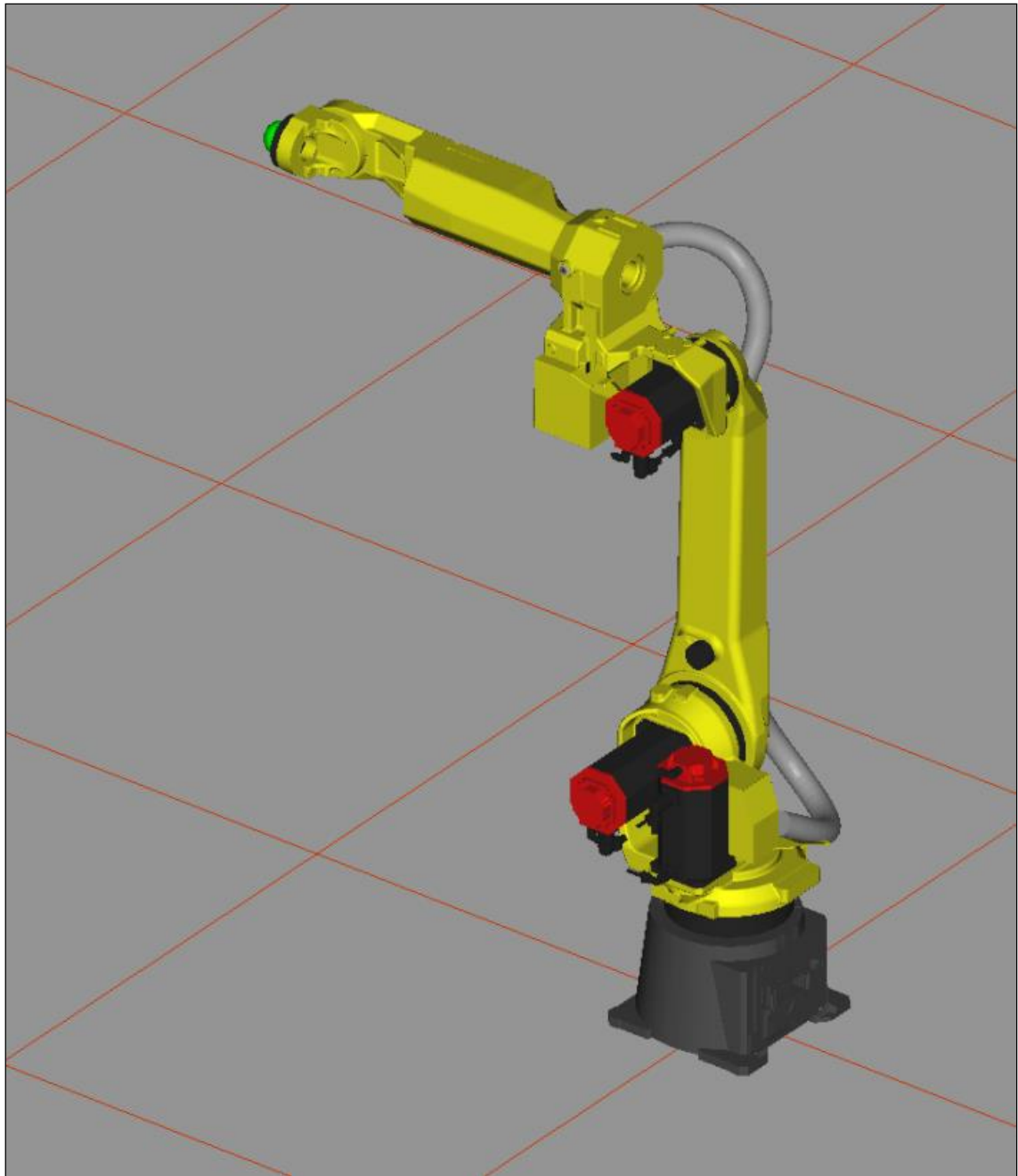
### 1.3 Robot FANUC M-20iA

Při realizaci této práce byl využit robot od firmy FANUC s konkrétním typovým označením M-20iA. Jedná se o šestiosého robota. Výrobce rozděluje tyto roboty podle využití, řada M-20iA je určena především k:

- Svařování CO2
- Manipulaci
- Zakládání do/Vykládání ze stroje
- Manipulace při sestavování a u vstřikovacích lisů
- Řezání-broušení-odjehlování-leštění
- Balení-sbírání-lepení-těsnění

#### 1.3.1 Mechanická jednotka FANUC M-20iA

Maximální nosnost jeho šesté osy bez omezení jeho přesnosti a rychlosti je garantována výrobcem až do zatížení 20 kg. Opakovací přesnost tohoto robota dostat se do uložené pozice je  $\pm 0.08$  mm. Roboty od firmy FANUC lze rozeznat od konkurenčních robotů už na dálku a to jednoznačným žlutým designem, který je pro většinu typů robotu FANUC jednotný. Výrobce garantuje dosah robota M-20iA na vzdálenost 1811 mm. Tento robot využívá ke svému pohybu cykloidní převodovku. Tento robot se stejně jako většina průmyslových robotů skládá z kontroléru a ručního ovladače zvaného teachpendant [23].

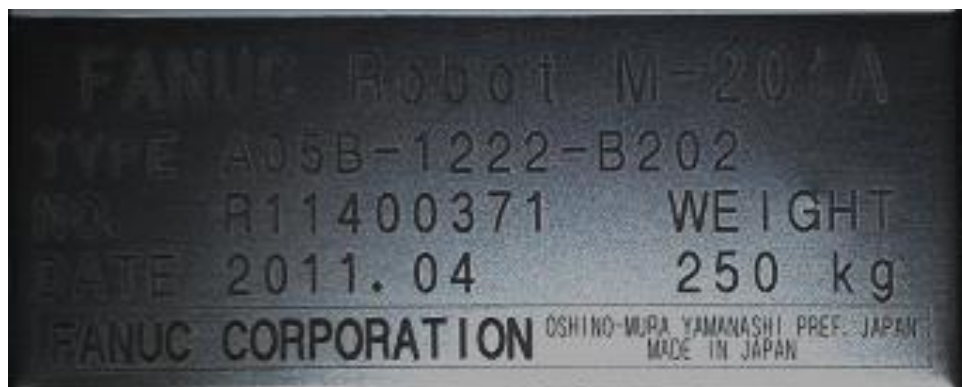


Obr. 8: Robot FANUC M-20iA v prostředí ROBOGUIDE [23]

Tab.2: Parametry jednotlivých os.

Jednotlivé osy	J1	J2	J3	J4	J5	J6
Rozsah pohybu [°]	340/370	260	458	400	360	900
Maximální rychlost [°/s]	95	175	180	360	360	550

Firma FANUC vyvinula pro snadnější programování a simulaci prostředí ROBOGUIDE. Jedná se o prostředí, ve kterém je možné tvořit reálné simulace robota od firmy FANUC a následně vytvořit reálný program, který lze posléze implementovat do robota [23].



Obr. 9: Výrobní štítek robota FANUC M-20iA [23]

### 1.3.2 Kontrolér

Kontrolér zajišťuje veškeré výpočty, komunikaci, bezpečnost a regulaci mechanické jednotky na dráze. Robota od firmy FANUC má až 250 softwarových funkcí, které jsou klíčové pro výkon robota. Kontroléry od firmy FANUC jsou vybaveny průmyslovým počítačem vlastní výroby a mají velmi rychlý náběh systému po zapnutí (cca 30s). U konkurenčních firem jsou to řádově minuty.

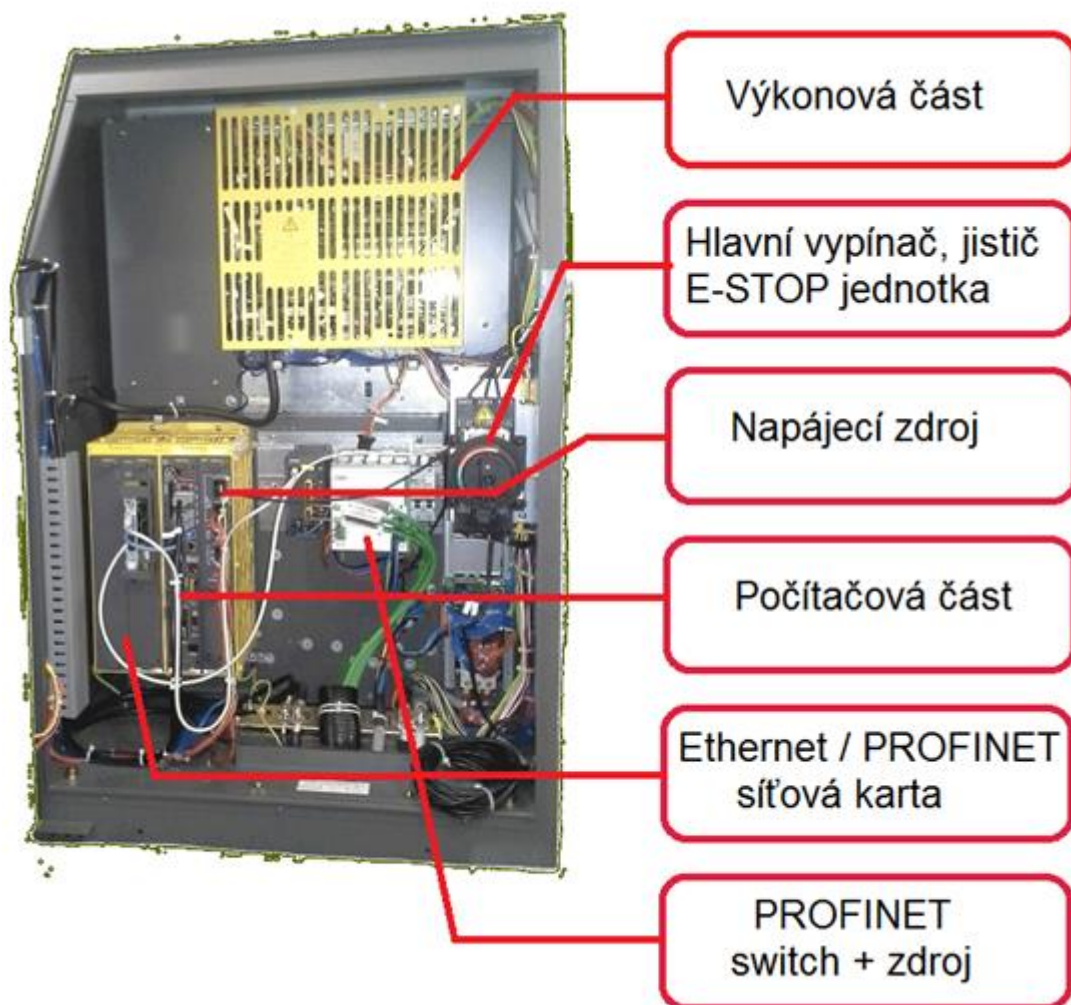
Operační systém stejně jako veškerý HW je vlastní produkce a veškerá paměťová média jsou polovodičové. Mechanické jednotky jako rotační pevné disky se nepoužívají. Celý systém si vystačí s 32 MB RAM. Při vytváření záloh máme možnost volit mezi zálohováním samotných souborů v souborovém systému FAT/FAT32 na paměťové karty PCMCIA/CF a pomalejší USB flash disky nebo bitového obrazu celé paměti (tzv. IMAGE) včetně systému na paměťové karty PCMCIA/CF.

Kontrolér má dva chladicí okruhy, které zajišťují vhodné ochlazování výkonových prvků, u kterých je třeba odvádět teplotu. Vnitřní „čistý“ okruh je s vnějším „špinavým“ spojen tepelným výměníkem. Je tedy zajištěno optimální chlazení všech prvků bez znečištění elektroniky robota nehledě na technologii, na kterou robot použít.

Kontrolér robota FANUC má hmotnost cca 180 kg. Pracuje s napájecím napětím 400V / 50 Hz [23].



Obr. 10: Výrobní štítek kontroléru robota FANUC M-20iA [23]



Obr. 11: Popis kontroléru [23]

### 1.3.3 Uživatelský panel

Jedná se o ruční ovladač díky, kterému je umožněno využívat veškeré funkce Robota. Každý uživatelský panel je vybaven jisticím prvkem, který se nazývá „tlačítko mrtvého muže“. Tento jisticí prvek je nutné při obsluze tohoto zařízení držet v prostřední poloze, tj. musí být stisknuto a zároveň nesmí být promáčknuto, jinak dojde k automatickému zablokování všech funkcí.

Uživatelský panel je vybaven obrazovkou, kterou nahradil v posledních letech díky modernímu vývoji technologii barevný dotykový display. Zařízení je vybaveno nalepovací klávesnicí díky, které je umožněno celé zařízení přehledně ovládat. Výhodou těchto klávesnic je to, že pokud se v provozu nějakým způsobem poškodí, je možné ji velice rychle a efektivně vyměnit, jelikož se jedná o samolepicí plochu s vyvedenými tlačítky, dříve bylo nutné poslat celý uživatelský panel na opravu na několik tisíc Kč. Dnes, v případě této závady, zaplatíme pouze cca 150 Kč za novou klávesnici.

Uživatelský panel je vybaven STOP tlačítkem, které v případě kolize okamžitě zastaví robota. A v neposlední řadě je zařízení vybaveno USB portem, který především slouží k nahrání zálohy robota [23].



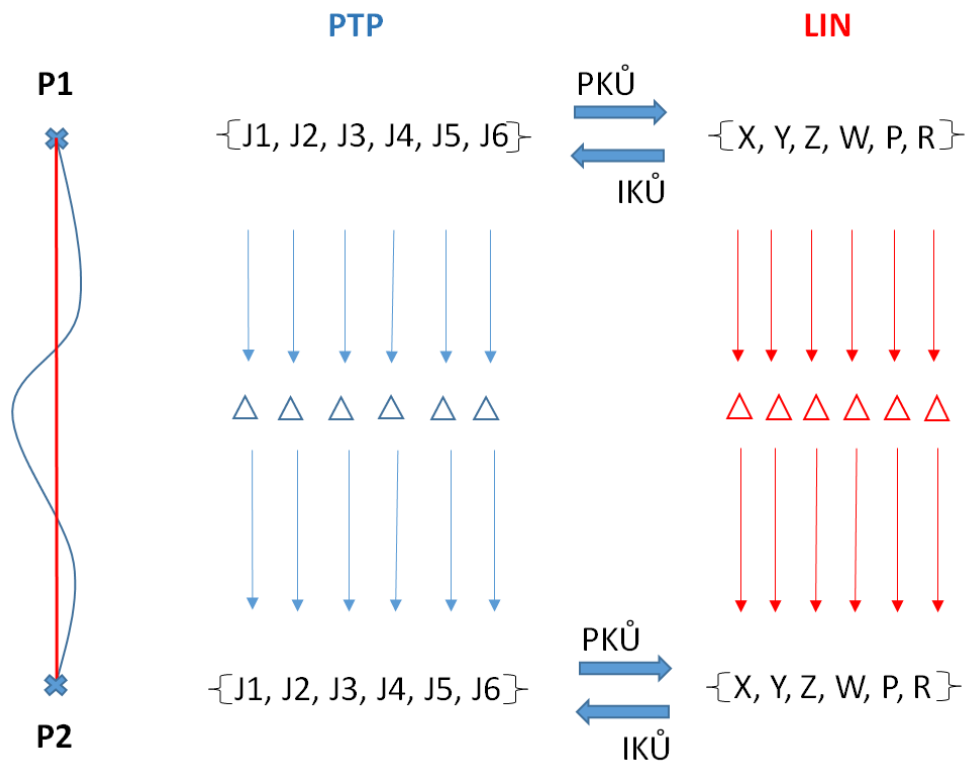
Obr. 12: Popis uživatelského panelu [23]

## 1.4 Pohybové příkazy průmyslových robotů

V robotových programech, nezávisle jakým způsobem vznikly, se uchovávají souřadnice bodů v prostoru a druhy pohybů mezi těmito body. Zatímco souřadnice bodů vyplývají z technologického zadání, konstrukce linky a orientace nástroje robota, volba pohybů je věcí programátora.

V zásadě se lze držet pravidla, že pohyby, při kterých robot pracuje v součinnosti s technologií, pro kterou byl určen, jsou vykonávány pomaleji a s vyššími nároky na přesnost a rychlost. Při těchto pohybech robot firmě přidanou hodnotou vydělává.

Naproti tomu pohyby mimo obrobek a bez součinnosti s technologií, kde se robot pouze přesouvá mezi technologiemi, kladou nárok na rychlost pohybu. Tradičně již od dob CNC a portálových jeřábů se takovému pohybu říká rychlopřejezd. Tímto pohybem robot může šetřit čas.

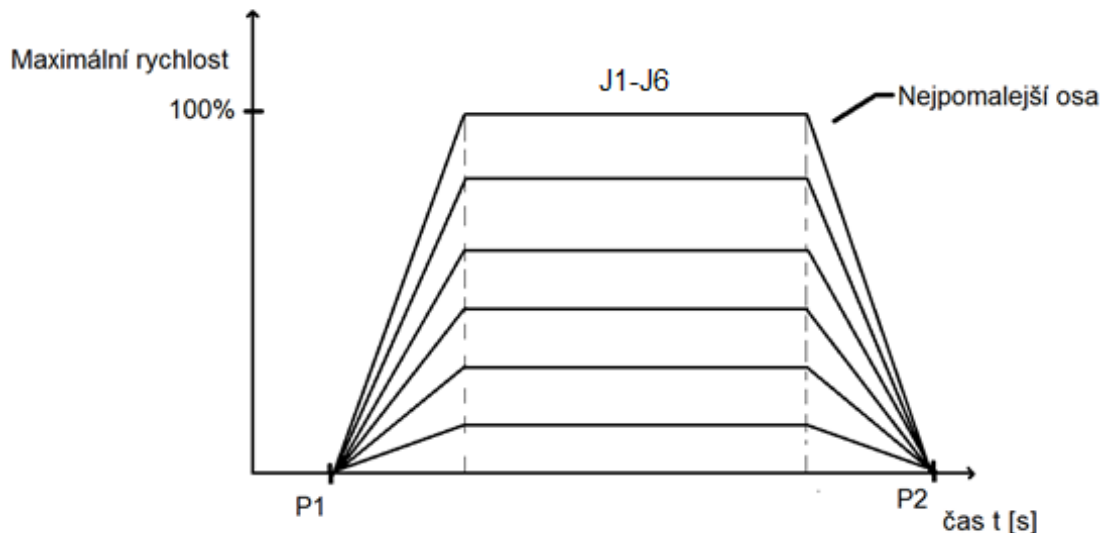


Obr. 13: Graf interpolací pohybů PTP a LIN



### 1.4.1 Rychlopřejezd

Rychlopřejezd, který se u robotů FANUC nazývá pohyb jointový a v programu je značen písmenem J, se obecně nazývá pohybem Point To Point. Při takovémto pohybu každá osa sama za sebe odtočí nebo natočí rozdíl úhlů mezi počátečním a koncovým bodem svého pohybu. Pozitivem takového pohybu, je to, že každá osa v průběhu pohybu provádí pouze monotónní pohyb, což šetří převodovky na osách a poskytuje nejrychlejší možný pohyb.

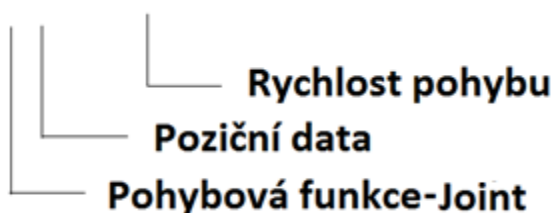


Obr. 14: Rychlosti jednotlivých os při synchronizovaném PTP [23]

Speciální verzí takového pohybu je potom synchronizované PTP, kdy robot dle svých strojních dat a rozdílů úhlů na jednotlivých osách provede vyhodnocení trvání pohybu, dle osy nejpomalejší. Tato osa jediná se potom pohybuje maximální rychlostí. Ostatní osy potom jedou sníženou rychlostí, tak aby dosáhly polohy ve stejný okamžik (viz Obr. 14:). Daný princip lze popsat slovy, že robot se zastaví stejně až ve chvíli, kdy se zastaví jeho poslední osa, tak proč by se některá z jeho os měla točit zbytečně rychle a potom čekat na pomalejší.

Jedná se o pohyb nativní, kdy robot nepočítá žádnou kinematickou úlohu, TCP se nemá pohybovat po definované dráze a na rychlost TCP nejsou kladeny žádné požadavky. Robot po takovém pohybu zaujímá vždy figuru kompletně definovanou cílovým bodem. Toto je další výhoda takovýchto pohybů. Z jakékoliv pozice lze robota „rozmotat“.

**J P[1] 100% FINE** — **Způsob ukončení pohybu FINE/ CNT**



Obr. 15: PTP pohyb

## 1.4.2 Technologické pohyby

Další skupinou pohybů, které nám poskytuje průmyslový robot, jsou pohyby technologické, při kterých se robot synchronizuje s nějakou technologií. Dnes nejčastějšími jsou lepení, MIG/MAG sváření, laserové sváření, řezání vodním paprskem, broušení atd. Použité technologie předepisují mimo požadované trajektorie též technologické rychlosti, či přímo vzájemnou komunikaci, kdy robot předává hodnotu rychlosti po průmyslové sběrnici do technologie.

Při takovýchto pohybech robot již kinematickou úlohu řešit musí a lze očekávat, že pohyby jednotlivých os rozhodně nebudou monotónní. U velkých průmyslových robotů s těžkými nástroji jsou potom zajímavé i rozběhové a brzdící rampy.

Vedle jednoduchých pohybů, které si pro popis dráhy vystačí s počátečním a koncovým bodem máme i pohyby, které se pro svojí složitou trajektorii po celé své délce lépe popisují analyticky. K tomu slouží tzv. generátory trajektorií.

### 1.4.2.1 LIN

Robot pohybuje TCP z počátečního bodu do koncového bodu po přímce (lineárně). Zde je třeba zdůraznit, že se pohyb týká všech složek vektoru souřadnice. Tedy i pouhé přeorientování nástroje na obrobku složkami W, P, R lze provádět příkazem LIN.

Zadání pro robota zní držet TCP na dráze a pohyb jednotlivých os se tomu dle IKÚ podřizuje. V koncovém bodě nelze vyžadovat, aby robot zaujmul figuru, která je v tomto bodě uložena. Pouze souřadnice samotného TCP jsou dosaženy.

Během těchto pohybů se robot často pohybuje kolem singularit na jednotlivých osách. Je úkolem programátora, aby při testování robota na toto dbal a vzal v úvahu.

Při výpadku robota na singulární bod, lze upravit pozici bodu tak, aby nedocházelo k přetáčení jednotlivých os robota. Řešení tohoto problému je možné pouze tak, že snížíme rychlost pohybu a rozdělíme dráhu robota na několik dílčích částí.

**L P[1] 100 mm/s FINE** — **Způsob ukončení pohybu FINE/ CNT**

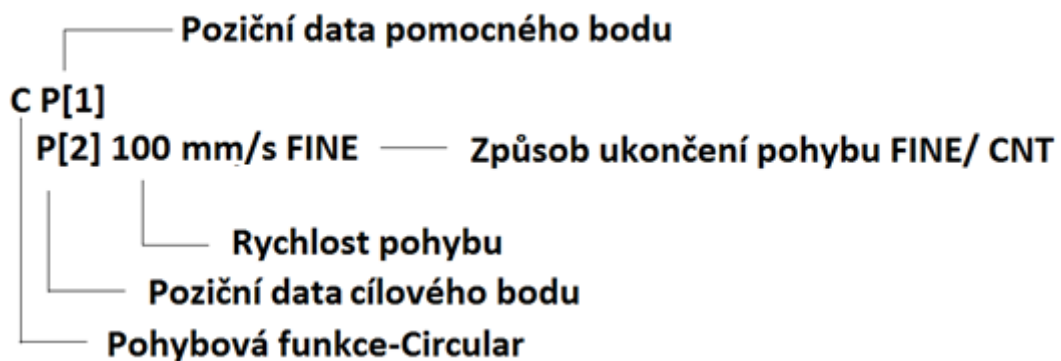
**Rychlost pohybu**  
**Poziční data**  
**Pohybová funkce-Linear**

Obr. 16: Programovací příkaz LIN pohybu

### 1.4.2.2 CIRC

Při tomto druhu pohybu se robot pohybuje po kružnici z počátečního bodu přes pomocný bod do bodu koncového. Plynule se mění nejen poloha TCP ale i orientace během pohybu. Pomocný bod je nutný pro definování poloměru a směru kružnice k cílovému bodu.

Metoda definování kružnice pomocným bodem se ukázala jako obecně nejvýhodnější, protože pomocný bod leží na požadované trajektorii a lze jej tudíž robotem dosáhnout. Pokud bychom volili postup definice kružnice pomocí středu a poloměru, pak bychom museli být schopni prostorové souřadnice bodu buďto zadat ručně nebo nechat robota nájezdem to středu tuto hodnotu odměřit. Při svařování trubek a vnějších oblouků je ale zřejmé, že střed nemusí být pro robota vůbec dosažitelný.

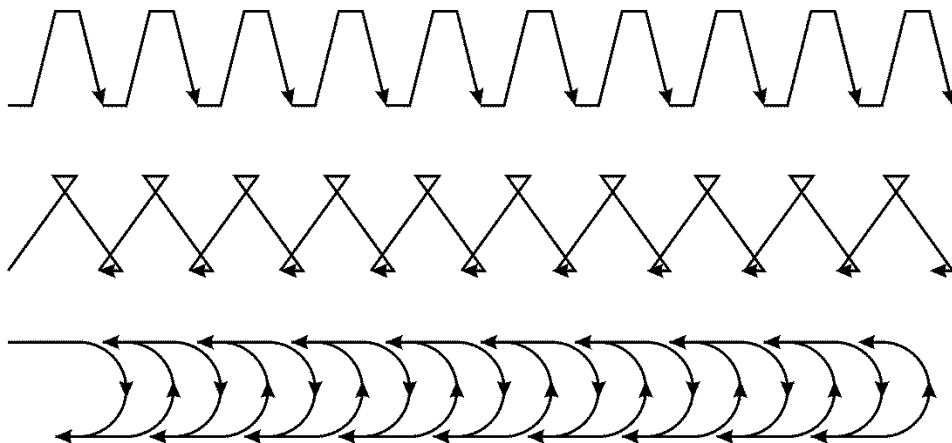


Obr. 17: Programovací příkaz CIRC pohybu

### 1.4.2.3 Generované trajektorie

Další méně využívanou možností robota jsou tzv. generované periodické trajektorie. Využívají se tehdy, pokud je pohyb robota periodický (podobá se periodicky generované funkci). V tomto případě je možné naprogramovat pouze jednu periodu jako kombinaci jednotlivých technologických pohybů LIN a CIRC.

Tento druh generování trajektorie se využívá např. při svaření, kdy je potřeba dodržet určitou periodickou dráhu svařování.



Obr. 18: Příklady generovaných pohybů

## 1.5 Logické příkazy (SPS)

Řídicí logika programu pracuje se vstupy a výstupy robota, provádí výpočetní operace atd. Jedná se o nepohybové instrukce.

Principiálně mohou být zadávány samostatně na řádek programu, nebo v rámci programovaného bodu. Dle standardu MQB musí být SPS funkce programovány na konci pohybového řádku prostřednictvím instrukce P-SPS (výjimka – PAYLOAD, USER FRAME, USER TOOL).

Tab.3: Logické operátory [20]

<table border="1"><tr><td colspan="2"><b>Aritmetické operátory</b></td></tr><tr><td>+</td><td>Sčítání</td></tr><tr><td>-</td><td>Odčítání</td></tr><tr><td>*</td><td>Násobení</td></tr><tr><td>/</td><td>Dělení</td></tr><tr><td>MOD</td><td>Zbytek po dělení 1=je, 0=není</td></tr><tr><td>DIV</td><td>celočíslné dělení (10 DIV 3 = 3)</td></tr></table>	<b>Aritmetické operátory</b>		+	Sčítání	-	Odčítání	*	Násobení	/	Dělení	MOD	Zbytek po dělení 1=je, 0=není	DIV	celočíslné dělení (10 DIV 3 = 3)	<table border="1"><tr><td colspan="2"><b>Relační operátory</b></td></tr><tr><td>=</td><td>Rovno</td></tr><tr><td>&lt;&gt;</td><td>Nerovno</td></tr><tr><td>&lt;</td><td>Menší</td></tr><tr><td>&gt;</td><td>Větší</td></tr><tr><td>&lt;=</td><td>Menší nebo rovno</td></tr><tr><td>&gt;=</td><td>Větší nebo rovno</td></tr></table>	<b>Relační operátory</b>		=	Rovno	<>	Nerovno	<	Menší	>	Větší	<=	Menší nebo rovno	>=	Větší nebo rovno
<b>Aritmetické operátory</b>																													
+	Sčítání																												
-	Odčítání																												
*	Násobení																												
/	Dělení																												
MOD	Zbytek po dělení 1=je, 0=není																												
DIV	celočíslné dělení (10 DIV 3 = 3)																												
<b>Relační operátory</b>																													
=	Rovno																												
<>	Nerovno																												
<	Menší																												
>	Větší																												
<=	Menší nebo rovno																												
>=	Větší nebo rovno																												
<table border="1"><tr><td colspan="2"><b>Logické operátory</b></td></tr><tr><td>AND</td><td>A</td></tr><tr><td>OR</td><td>Nebo</td></tr><tr><td>!</td><td>Negace (NOT)</td></tr></table>	<b>Logické operátory</b>		AND	A	OR	Nebo	!	Negace (NOT)																					
<b>Logické operátory</b>																													
AND	A																												
OR	Nebo																												
!	Negace (NOT)																												

### 1.5.1 Vstupy/Výstupy

**Vstupní signály** – I (E) jsou informace, které přicházejí do řízení robota z periférií (Simatic, přípravek, nástroj, sousední robot...).

**Výstupní signály** – O (A) vysílá řízení robota k perifériím (ovládání nástroje, upínek, info k jinému řídicímu systému..).

**DO/DI** – Digitální vstupy/ výstupy – 2 stavy vypnuto OFF/ zapnuto ON (0/1). Jedná se o předdefinovanou skupinu signálů, která slouží ke komunikaci, ovládání nebo zjištění stavu periférií. Digitální I/O mohou nabývat pouze dvě hodnoty (24 V = ON/0 V = OFF). V řízení je předdefinováno 4096 DI a DO. Některé jsou systémové (nelze je použít jako uživatelské).

**AO/AI** – Analogové vstupy/výstupy – spojité – číselná hodnota (například tlak, svářecí proud). Jedná se o předdefinovanou skupinu signálů, která slouží k ovládání či snímání veličiny v určitém rozsahu. Analogové I/O mohou nabývat nejčastěji 0-10 V hodnotou 0-2000 - dle připojené analogové karty PLS. V řízení je předdefinováno 64 AI a AO. Některé jsou systémové (nelze je použít jako uživatelské).

**GO/GI** – Jedná se o předdefinovanou skupinu po sobě jdoucích signálů (DI/DO), která nejčastěji slouží k odeslání decimálního čísla v binárním kódu. V řízení je předdefinováno 64 GI a GO.

**RO/RI** – Robotické digitální – nevyužívá se v rámci MQB.

**SO/SI** – Vnitřní signály, které využívá sám systém, nelze upravovat.

**F** – Flag – Jedná se o předdefinovanou skupinu virtuálních signálů. Flagy mohou nabývat pouze dvě hodnoty (ON/OFF). V programu se využívají obdobně jako DO/DI, ale není k nim připojeno žádné zařízení - jsou pouze virtuální. V řízení je předdefinováno 1024 Flagů. Některé jsou systémové (nelze je použít jako uživatelské).

**M** – Marker – soubor podmínek (I/O) (D, A, G). Jedná se o virtuální BIT, který může nabývat pouze dvou hodnot (ON/OFF logická 1/0). Po zpracování instrukce je Marker zpracováván cyklicky, tj. že jeho hodnota je v každém okamžiku zpracování programu aktuální. V programu je možné použít až 200 Markerů.

## 1.5.2 Vnitřní paměti

Registry (R[ ]) neboli paměťové místo, do kterého je možné ukládat číslo typu REAL (od  $2.9 \cdot 10^{-39}$  do  $1.7 \cdot 10^{+38}$ ). K dispozici je 820 registrů. Numerická hodnota v jakékoli instrukci může být nahrazena obsahem registru. K programování registrů je možné využít logické operátory, které umožňují základní numerické operace mezi registry.

### Adresace registru

**Přímá** – Hodnota registru je zapsána přímo do tabulky registrů (manuální zápis), nebo prostřednictvím příkazu – např. R[2]=2.

**Nepřímá** – Hodnota registru je do registru zapsána prostřednictvím dalšího registru např. R[R[3]]=34

## 1.6 Řízení a větvení programu

### 1.6.1 Bezpodmínečné větvení programu

Jelikož zde nedochází k vykonání žádné speciální podmínky, je způsobeno větvení programu vždy. Mezi instrukce, které zajišťují toto větvení patří především tyto instrukce:

**CALL** umožňuje volání podprogramu/ makra. Po vykonání příslušné funkce se systém vrací zpět do hlavního programu na následující instrukci.

**CALL MAKRO080** ; Volání modulu MAKRO080

**RUN** spustí paralelní proces. Takovýto proces nesmí nárokovat přístup na již alokovanou mechanickou jednotku. Volající program běží dále a nečeká na ukončení volaného programu

**JMP/LBL** je operace pro nepodmíněný skok v programu. Kde LBL[ ] číslo návěští na, které je proveden skok. Za to příkaz JMP LBL[ ] - je příkaz skoku na námi zvolené návěští.

**LBL[ ]** číslo návěstí na které je proveden skok.

**JMP LBL[ ]** příkaz skoku na zvolené návěstí.

### 1.6.2 Podmíněné větvení programu

Skok z jednoho místa v programu na jiné, pokud je splněna určitá podmínka.

**IF**, „jestliže“ - podmíněné porovnání: Způsobuje přeskočení na specifikované návěstí či program, pokud je splněna definovaná podmínka. Při nesplnění podmínky pokračuje program na následujícím řádku.

**IF R[2] >= R[3], LBL[1]** Skok na návěstí LBL[1] při splnění podmínky  $R[2] \geq R[3]$

**SELECT/ELSE** - Podmíněný výběr: Sestává z několika příkazů porovnání podmínky. Přeskočí na specifikované návěstí či program dle splněné definované podmínky. V případě, že žádná z podmínek nevyhovuje, pokračuje program příkazem **ELSE** a jím specifikované příkazy.

**3: SELECT R[2] = 1, JMP LBL[15]** ; Pokud je splněna podmínka  $R[2]=1$  je proveden skok na návěstí LBL[15].

**4: =2, JMP LBL[3]** ; Pokud  $R[2]=2$  je proveden skok na návěstí LBL[3].

**5: =3, JMP LBL[4]** ; Pokud  $R[2]=3$  je proveden skok na návěstí LBL[4].

**6: ELSE, CALL MAKRO0025** ; Pokud je hodnota registru R[2] jiná než předchozí hodnoty tak je proveden skok na MAKRO0025

## 1.7 OFF-LINE programování (OLP)

OFF-LINE programování je oproti ON-LINE programování rozdílné v tom, že robot je reprezentován svým virtuálním modelem a pohybuje se ve virtuálním modelu pracoviště. Pro tento způsob programování je zvláště důležitá věrnost modelu, která má zásadní vliv na množství práce při přenosu takto vzniklých programů do reálného robota. Modely pracovišť většinou vznikají na základě technických výkresů nebo 3D scanů reálných pracovišť. Takovéto programování umožňuje libovolné simulace na základě, kterých se dnes rozhoduje o technických realizacích.

Software ROBOGUIDE je vývojové prostředí pro tvorbu OFF-LINE programů pro roboty FANUC a jejich následnou simulaci na reálném modelu. Použití tohoto softwaru značně podporuje výrobu. Jeho velkou výhodou je to, že nemusíme být fyzicky u robota, ale můžeme software použít kdekoliv mimo pracoviště.

Tento software je schopný provést kalibraci pracovní buňky a souřadného systému uživatele. Nespornou výhodou je to, že je možné tyto parametry kdykoliv pružně změnit aby případně odpovídali realitě.

Tento model je posléze možné graficky odsimulovat a výstup uložit ve video formátu \*.AVI, což má za výhodu velice přehlednou prezentaci dané problematiky. A je pak tedy mnohem snazší rozhodovat o případném řešení.

Nespornou výhodou je to, že lze předem odhalit kolize a kolizní stavy robotu, které by mohli na reálném pracovišti vzniknout. Pokud by tento problém nebyl nikterak ošetřen, mohly by vzniknout až milionové škody a to si v dnešní době nemůže žádná firma dovolit, nehledě na ztrátu způsobenou zastavením výroby. ROBOGUIDE automaticky hlásí chybové stavy a uživatel je prakticky hned upozorněn.

Tento software je schopen pracovat s CAD daty ve formátu IGES.

Další funkcí tohoto prostředí je možné využití virtuálního uživatelského panelu, který odpovídá reálnému ovladači robota. Za pomoci tohoto ovladače je možné vyzkoušet správnou funkci modelu, který odpovídá reálně simulované skutečnosti. Tento virtuální uživatelský panel obsahuje stejné displejové zobrazení a ovládání, jako fyzicky reálný ovladač robota. Není zde třeba absolvovat, žádné speciální školení na obsluhu tohoto virtuálního ovladače, jelikož zde nehrozí žádné škody způsobené neproškolenou osobou. Proto je možné i tento virtuální ovladač použít jako určitý druh „trenažeru“.

S prvky buňky lze vždy kdykoliv pružně manipulovat a je možné jejich atributy jakkoliv měnit. Dále je možné zkoušet jednotlivé funkce jednotlivě nezávisle na sobě. Je umožněn Import obrobků, nástrojů, upínadel a překážek ve formátu IGES. Je možné zjistit přesnou dobu cyklu výroby a podle toho uzpůsobit výrobu.

## 1.8 Generování trajektorie podle hran importovaných objektů

OFF-LINE programování nabízí velice zajímavou možnost využití, a to takovou, že je možné generovat trajektorii podle importovaných obrobků. Vývojové prostředí pro tvorbu simulačních modelů umožňuje importování CAD souborů v základních formátech a nadále s nimi pracovat.

S těmito objekty je možné pracovat v robotickém simulačním prostředí FANUC ROBOGUIDE v jejich reálném měřítku. Je nutné zadat softwaru, o jaký typ objektu se jedná, aby bylo možné s nimi podle konkrétních vlastností jednotlivých objektů pracovat. Je tedy nutné určit, zdali se jedná pouze o pevnou překážku (Obstacle), přípravek kde se generují díly (Fixture) nebo o obrobek (Part), na kterém je možné vygenerovat trajektorii podle jeho hran.

Nás v tomto případě zajímá hlavně možnost generování trajektorie. Pokud tedy nadefinujeme objekt v této skupině, je možné pracovat s jeho hranami jako s nositelem souřadnice. Prostředí ROBOGUIDE automaticky na základě těchto informací vygeneruje program pro robota, který tuto dráhu velice přesně zajede. Takovýto objekt budeme importovat z programu CAD.

Velkou výhodou je, že jsme schopni poměrně jednoduše dosáhnout složitých trajektorií, s mnoha body, což by s reálným robotem na hranách reálného obrobku trvalo nepoměrně déle. Není zde totiž potřeba pro každý ukládaný bod pohybovat do tohoto bodu robotem.



## 2 Rešerše CAD systémů

Technická dokumentace existovala již od nepaměti, kdy bylo potřeba přesně zakreslit nejrůznější objekty. Dříve se využívala pouze standardní „ruční“ forma, kde bylo zapotřebí zakreslit jednotlivé prostorové zobrazení předmětu. Tato zobrazení se běžně nazývají nárýs, bokorys, půdorys. V praxi to není nic jiného než jen zakreslení součástky ze tří různých pohledů, které jsou ovšem na sobě závislé.

První počítačovou snahou o technickou dokumentaci bylo 2D modelování, kde bylo možné si prohlížet předmět pouze, z 3 zakreslených pohledů.

Od tohoto způsobu dokumentace bylo později upuštěno, jelikož byly vyvinuty 3D softwary, které dovedou předmět zachytit v reálné podobě.

Díky těmto softwarům je mnohem snazší předmět zakreslit a i následně získat potřebná data, se kterými lze dále mnohem pohodlněji pracovat.

Ve firmě ŠKODA AUTO a.s. je technická dokumentace nezbytnou součástí. Za zmínku stojí především známý AUTOCAD, Pro Engeneer, Catia, Creo Parametric.

Díky CAD systému schopni rychleji upravovat dané technické výkresy a tím pádem aplikovat přesné korekce. Tato data je poté velice jednoduché sdílet.

Dále je možné tyto data velice snadno a pohodlně exportovat pro další práci a to například pro tvářecí techniku, obráběcí techniku.

Následovalo využití CAD systému pro sestavování kusovníků.

V současné době je posledním krokem využití těchto dat pro simulaci kinematiky i dynamiky. Následovalo i využití numerických metod FEM (metoda konečných prvků – Finite element's metod) a výkonných počítačů díky kterým mohly být prováděny i zátěžové testy jednotlivých součástek.

V současné době těží CAD programy z nepřeberného množství knihoven. Stalo se zvyklostí, že výrobci součástek takové knihovny samy vytvářejí a poskytují je potenciálním zákazníkům, což zásadním způsobem urychluje práci.

## 2.1 Historie

**1959** – V tomto roce byl vyvinut software pro zápis grafiky na pásku, která umožňovala zpětné načtení na obrazovku.

**1960** – V tomto roce byl vyvinut AUTOCADový programovací jazyk nazývaný LISP, který se hojně používá dodnes.

**1963** – V tomto roce byl vyvinut program SKETCHPAD od firmy ITEK umožňující základní principy technického kreslení. Tento software byl první bezplatnou provizorní verzí CAD systému.

**1964** – V tomto roce byl vpuštěn na trh první placený CAD systém, který byl následníkem CAD softwaru od firmy ITEK. Jeho cena se pohybovala něco okolo 500.000,- USD. Tento software byl pojmenován DAC (Design Automated by Computer). Tento systém využíval interaktivní grafiku (umožnil uživateli např. rotování objektu a pohled pod různým sklonem).

**1966** – V tomto roce byly vyvinuty CAD softwary, které byly schopny pracovat s různými geometrickými problémy. Tyto softwary pracovaly tehdy ještě na velmi drahých a rozměrově velkých sálových počítačích.

**1968** – První CAD systémy určené pro domácí použití.

**1969** – Koncem šedesátých let se začínaly vydávat první placené CAD systémy, které vyvíjelo čím dál více firem, mezi které patřili např. Auto-trol, Computervision, Evans & Sutherland atd.

**1972** – V tomto roce vyvinula firma SynthaVision software zvaný MAGI (Mathematics Application Group, Inc.) což byl první software pracující s 3D modely. Nejednalo se o CAD systém, ale program pro analýzu radiačního záření. Tyto 3D modely jsou zde reprezentovány podobně jako v budoucích verzích CAD systému.

**1973** – V tomto roce uvedla na trh firma United Computing software zvaný UNI-GRAPHICS. Tento program již plnohodnotně zvládal základní 2D grafiku.

**1975** – V tomto roce vyvinula Francouzská letecká firma zvaná Avions Marcel Dassault (AMD) celosvětově známý systém CATIA (Computer Aided Three Dimensional Interactive Application).

**1976** – V tomto roce firma MCS vydala software AD-2000. Což byl systém pro první 32-bitové počítače.

**1979** – V tomto roce vzniknul formát CAD dat zvaný IGES. Firma M&S Computing (dnes již vystupující pod názvem Intergraph) vyvinula svůj CAD systém nazvaný IGDS.

**1980** – V tomto roce vyvinula firma T&W software zvaný Versa CAD.

**1981** – V tomto roce byl vyvinut software zabývající se 3D modelováním zvaný Unisolid. V další řadě byla vyvinuta další verze softwaru CATIA verze 1. od francouzské firmy Avions Marcel Dassault (AMD), která byla přímo uzpůsobena pro 3D návrh.

**1982** – V tomto roce byl vyvinut software CAD-plan od firmy P-CAD. Byl založen AUTODESK, kde bylo cílem vyvinout CAD program, který by byl k dispozici běžným uživatelům za cenu cca 1000USD a použitelný na běžných PC. Od této firmy vzniknul software AUTOCAD 1. V tuto dobu se pracovalo pouze s datovými formáty DWG a DXF.

**1983** – V tomto roce se začínal vyvíjet datový formát zvaný STEP. Oficiálně byl vydán až v roce 1991.

**1984** – V tomto roce vznikla další verze CATIE – verze 2, která byla především určena pro návrh dílů pro letecký průmysl.

**1988** – V tomto roce vznikla další verze CATIE – verze 3. Firma PTC zahájila prodej softwaru zvaného Pro/ENGINEER.

**1989** – V tomto roce se začíná v USA prodávat software zvaný ArchiCAD.

Byla vydána první verze softwaru zvaného KOMPAS-Graphic.

Firma CSC vpustila na trh software zvaný MicroCADAM, - CAD/CAM.

**1990** – AUTODESK vydává další verzi AUTOCADu R11.

**1992** – AUTODESK vydává další verzi grafického softwaru AUTOCAD R12. Vyvinut software zvaný Visio Technical

**1993** – V tomto roce začala firma PTC prodávat Pro/ENGINEER pro Windows.

První verze ArchiCADu pro Windows. Zároveň vychází další verze AUTOCADu, která je přizpůsobena jak pro DOS i pro Windows. Tato verze umožňovala export do DWF formátu. Francouzská firma Dassault vyvinula novou verzi CAD softwaru – CATIA 4.

**1994** – Byl vyvinut software zvaný DesignCAD 3D

**1997** – Vychází první verze softwaru zvaného KOMPAS-3D určená pro Windows. Firma AUTODESK vyvinula další verzi AUTOCAD R14

**1998** – V tomto roce vychází AUTOCAD pro Windows.

Francouzská firma Dassault vyvinula software CATIA ver. 5

Veškeré informace ohledně historie byly čerpány z pramene [22]

## **2.2 AUTOCAD**

Prvním softwarem na světě, který je určen k technické dokumentaci a který nahradil v technických výkresech tužku, je software AUTOCAD.

Tento software plní svojí úlohu již od roku 1982, kdy byla vytvořena první verze tohoto softwaru. Dříve byl tento software určen k projektování 2D výkresů postupem vývoje došlo k upgradu na 3D, která umožnila projektování 3D výkresů, které jsou pro uživatele velice přehledné a výhodné.

Tento software je neustále vyvíjen a vylepšován. V současné době existuje již nová verze AUTOCAD 2015 (R20.0), která vyšla v březnu 2014. Pracuje s formáty DWG a DXF, které je možné číst i v konkurenčních CAD programech.

DWG je základním formátem softwaru AUTOCAD, tato zkratka byla vytvořena z anglického slova DraWinG což znamená v překladu výkres nebo kreslit.

Následníkem tohoto softwaru se v průmyslovém odvětví staly především softwary Catia a Pro Engineer, které pracují na podobném principu, jako AUTOCAD.

### 2.3 Catia

Catia je vývojové prostředí pro počítačový návrh konstrukcí a výroby jednotlivých prvků. Tento systém byl vyvinut již v roce 1975 v letecké firmě Avions Marcel Dassault, která se zabývá výrobou stíhacích letounů a vyvinula tento software pro usnadnění práce. V současné době je tento software hojně používán v automobilovém průmyslu. Nejnovější verze rovněž umožňují, tzv. „rozhybání“ modelu, čímž ověříme kinematické vazby. Tento systém pracuje především s CAD daty. V tomto systému je možné vyvíjet 2D a 3D modely, pracuje pod běžně známými operačními systémy, jako jsou např. Microsoft Windows, Linux a IBM AIX.

### 2.4 Creo Parametric

Tento grafický software spadá do skupiny tzv. Parametrických kreslicích programů. Jelikož se jedná o Parametrický software, je zde každý krok naší práce uložen do určité posloupnosti. Tím pádem stoupají nároky na výkon PC. Creo Parametric podporuje většinu grafických CAD formátů. Velkou výhodou tohoto softwaru je kontrola správnosti náčrtu a možnost vracení se k jednotlivým předchozím krokům. Tento software pracuje plnohodnotně pod operačním systémem Windows. Je zde ovšem vyžadována operační paměť o velikosti aspoň 4 GB a poměrně vysoké rozlišení grafické karty tj. 1280 x 1024 px.

### 2.5 AUTODESK Inventor Professional

Jedná se o velice využívaný CAD software v průmyslu. Tento software je téměř každoročně upgradován na novou verzi. Pracuje plnohodnotně s formátem \*.DWG a je zde přizpůsobena spolupráce se softwarem AUTOCAD.

Tento software pracuje s technologií zvanou DWG TrueConnect, která má na starosti kompatibilitu 2D a 3D dat. Dále je zde funkce zvaná DWG Read, díky které se tyto soubory otevírají bez zbytečných časových prodlev.

### 2.6 DraftSight

Jedním z nejznámějších je produkt od konkurenční společnosti Autodesku – společnosti Dassault Systems (tvůrce 3D modelovacího software SolidWorks) – který se jmenuje DraftSight.

Tento produkt je volně ke stažení a je možné jej používat i ke komerčním účelům v neomezeném množství kopií.

DraftSight je téměř identický s produktem AutoCad, jeho podobnost je až zarážející. Je v něm možné tvořit a editovat všechny soubory typu DWG a DXF a to včetně všech edicí.

Bylo odzkoušeno, že s produkty Autodesk je tento program plně kompatibilní, lze v něm AutoCadové výkresy otevírat a uložené jsou čitelné opět v AutoCadu.

Pro uživatele, kteří jsou zvyklí na AutoCad bude práce v DraftSight naprosto samozřejmá, prostředí je identické, všechny příkazy najdeme na místech kde jste zvyklí, jen některé mají trochu jiná označení (např. příkaz „ekvilidistanta“ je zde řešena příkazem „odsadit“) [24].

## 2.7 SOLIDWORKS

Tento CAD systém, který je v současné době využíván ve strojírenství. Byl vytvořen společností SolidWorks Corporation. Je schopen vytvářet 3D objekty, výrobní výkresy atd. Tento software zvládá prakticky velice efektivně objemové modelování, navíc je ovládání velice snadné a přehledné. Je zde možné vytvářet výrobní dokumentaci.

Díky tomuto softwaru je uživatel schopen prakticky hned bez jakýchkoliv zbytečných formalit a složitostí jednoduše modelovat požadované objekty.

## 2.8 FreeCad

Jedná se o OpenSorce software. Ovládání softwaru je velice obdobné jako u jiných komerčních produktů např. AUTOCAD. Tento software má podstatně širokou oblast využití např. strojírenství, robotika, stavebnictví atd. Tento software nabízí mnoho druhů vstupních a výstupních formátů a to: STEP, IGES, OBJ, STL, DXF, SVG, STL, atd.

Tento software využívá kompletní české rozhraní včetně českého menu, dialogového okna atd. Obsahuje plně lokalizovaný příkazový řádek ve, kterém je možné využívat české ale i standardní anglické příkazy s podtržítkem přesně jako v softwaru AUTOCAD.

Podpora DWG, DXF formátu je možná pouze za použití konvertoru AUTOCADových čar, tedy souborů s příponou \*.lin. Dále je zde pak možná standardní podpora vyšrafovaných ploch stejně jako v AUTOCADu, což jsou soubory typu \*.pat.

Obsahuje základní editační funkce podobné AUTOCADu. Veškeré parametry objektů je možné přehledně editovat v tabulce hodnot. Do výkresu je možné vkládat i rastrové obrázky.

## 2.9 Google SketchUp

SketchUp je původně produktem firmy @Last Software, kterou v březnu 2006 odkoupil Google.

Vylepšená placená verze SketchUp 7 Pro se vyznačuje zvýšenou funkcionalitou v oblasti vytváření inteligentních a dynamických objektů, možností exportu do mnoha rozšířených 2D (PDF, EPS, EPX, DWG, DXF) i 3D (3DS, DWG, DXF, DAE, FBX, OBJ, XSI, VRML) formátů. Možné je také generování animací v rozšířených video formátech MOV nebo AVI nebo převedení do specifických formátů užitím přídatných modulů, včetně geodatabázi podporovaných systémem ArcGIS. Další součástí rozšířené verze je

aplikace vytvářející propracované prezentace modelů (LayOut 2) nebo Style Builder, sloužící k návrhu prostředí a přizpůsobení stylů při modelování ve SketchUp. Testování i modelování proběhlo pomocí plné verze SketchUp 7 Pro, k níž firma Google zdarma poskytla roční studentskou licenci.

Produkt je založen na zcela jiných principech 3D modelování než tradiční systémy. Záměrně potlačuje detail CAD aplikací výměnou za jednoduchost, názornost a intuitivnost; nutno však říct, že to nijak neovlivňuje přesnost. Stavební elementy jsou jednoduché, ovládací prvky inteligentní a modelovací nástroje rychlé s okamžitě kontrolovatelným výsledkem [25].

## **2.10 Další možnosti importu dat do prostředí ROBOGUIDE**

Tyto softwary nepatří sice do skupiny CAD systému, ovšem jsou schopny komunikovat s prostředím ROBOGUIDE. Na základě tohoto faktu je zde uvedena i malá zmínka o těchto programech.

### **2.10.1 3D Studio MAX**

3D Studio MAX je prostředí, které je určeno pro 3D grafiku, vizualizace a animace. Tento software podporuje spoustu druhů CAD formátů, jako jsou např. běžné DWG a DXF ale také spousta dalších CAD formátů. Je možné ukládat výstupní animace ve formátu MOV a AVI.

Tento software byl vyvinut firmou AUTODESK. Z tohoto softwaru je možný import dat do OFF-LINE programovacího prostředí ROBOGUIDE. Nespornou výhodou tohoto softwaru je silná podpora při tvorbě grafických objektů. Tento software bývá většinou běžně používán v průmyslových aplikacích, ale není problém jej využít i na tvorbu reklam a grafiky do počítačových her.

### **2.10.2 Process Simulate**

Asi největší pozornost si zaslouží software Process Simulate od firmy Siemens. Výhodou tohoto softwaru je jeho univerzálnost, jelikož tento software umožňuje programovat i jiné roboty od jiných výrobců. Toto je možné pouze tehdy, pokud příslušný výrobce poskytne model robota v požadovaném datovém formátu \*.JT nebo \*.COJT, pro popis mechanické jednotky a tzv. virtuálního kontroléru pro výpočet kinematiky, dynamiky a obsluhu vstupu/výstupů daného robota.

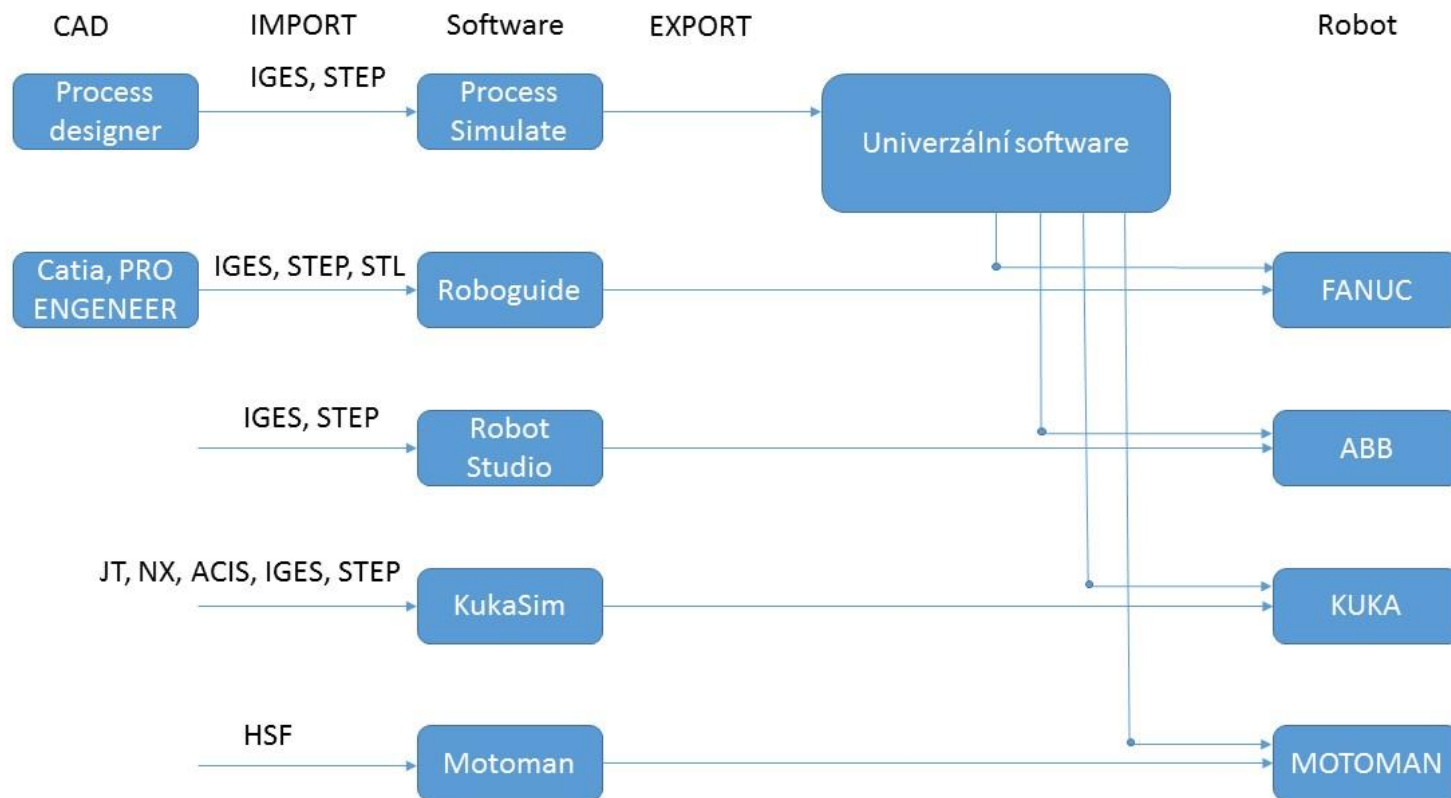
Ke správné implementaci programu je u tohoto softwaru nejprve použito vývojové prostředí Process Designer. Jedná se o prostředí, kde je vytvořen jakýsi počáteční koncept – 3D layout. Tento koncept většinou vytváří projektant, který se stará v první řadě hlavně o to, aby odpovídaly počty jednotlivých prvků v simulaci, jako jsou například počet robotů, přípravků, překážek atd. Tento koncept není odsimulován a je tedy velice nepřesný. Hlavním cílem, proč se toto provádí, je především cena, jelikož tento layout slouží v tuto chvíli jako kusovník jednotlivých artiklů. Jak je tedy patrné všechny data, jsou v Process Designeru STATICKÁ.

Pokud je kusovník artiklů hotov a jsou zde obsaženy všechny potřebné komponenty, je i nadále nutné tento layout odeslat externím dodavatelům jako jsou např. EDAC atd. a začnou se upravovat kinematické modely z Catie do dynamického formátu \*.JT.

Soubory s formátem \*.JT lze poměrně jednoduše odsimulovat v softwaru Process Simulate a je možné si ověřit zdali jsou všechny mechanické vazby funkční. S kinematikou lze hýbat pouze v Process Simulate.

Model robota poskytuje výrobce, jako například předně známí výrobci robotů jako jsou KUKA, FANUC, ABB, MOTOMAN atd. Od těchto výrobců je poskytnut adresář \*.COJT, ve kterém je mimo jiné obsažen přesný dynamický model robota od výrobce ve formátu \*.JT. Díky tomu je software Process Simulate považován za univerzální software.

Na diagramu (Obr. 19:) je možné pozorovat jednotlivé výrobce robotů, jejich vývojové prostředí a formáty dat, které lze importovat a exportovat.



Obr. 19: Datové formáty pro CAD a OLP



## 2.11 CAD produkty

Pokud bychom se hlouběji ponořili do současného trhu CAD produktů, tak patrně zjistíme, že je značné množství softwaru, které pracují s CAD daty. Ovšem správnou otázkou je to, který produkt se jeví pro běžného uživatele jako nejvýhodnější. Většinu těchto kreslicích softwarů je možné zakoupit v určitých balíčcích, které jsou případně omezeny. U CAD produktů zabývajících se např. návrhem plošných spojů je software omezen např. počtem součástek v modelu atd. Zase obecně platí, že pokud bychom chtěli produkt bez jakýchkoliv omezení, budeme nuceni si u výrobce zakoupit větší balíček s více funkcemi.

Z důvodu řešení diplomové práce byl zvolen OPEN SOURCE software FreeCad (2.8).

V porovnání s ostatními konkurenčními produkty, které jsou do značné míry velice podobné, vycházel nejlépe produkt FreeCad. Tento produkt je v první řadě úplně zdarma, což je velké plus pro mnoho uživatelů, kteří potřebují nárazově využít software pro 3D modely. V druhé řadě tento produkt vykazoval značnou funkcionalitu a jevil se v porovnání s ostatními placenými produkty jako plně konkurenceschopný. Tento produkt zabírá cca 120 MB a pracuje i jako přenosná aplikace na flash disku bez instalace. Velikost je menší především proto, že neobsahuje např. oproti AUTOCADU knihovnu součástek a dílů, které jsou většinou importovány přímo od výrobce.

Obdobně je to realizováno i v robotických CAD programech jako je v našem případě software ROBOGUIDE, kde jsou naimportovány veškeré knihovny robotu FANUC od výrobce. Opět i zde platí, že tento software je plnohodnotně použitelný pro roboty FANUC. Pro širší podporu je výhodnější využít software Process Simulate, který je více univerzální.

## 2.12 Formáty CAD dat

### 2.12.1 Formát STL

Patří mezi jeden z nejpoužívanějších CADovských formátů \*.STL. Tento formát je používán např. při 3D tisku nebo při 3D skenování. Reprezentovaný model v tomto formátu je sestaven z malých polygonových plošek, které odborně nazýváme fazety. Z těchto plošek je vytvořena optimalizovaná síť.

STL formát je možné reprezentovat dvěma základními způsoby, buď za pomoci ASCII formátu, jehož hlavní výhodou je, že je zápis poměrně přehledný, nebo máme možnost využít binárního formátu.

### 2.12.2 Formát IGES

IGES neboli (Initial Graphics Exchange Specification). Jedná se o datový formát, který umožňuje přenos 2D, ale i 3D dat. Tento formát obsahuje pouze geometrická data, nikoliv grafické informace.

#### Struktura formátu IGES

Data formátu IGES jsou reprezentována v ASCII kódu, který je uspořádán v podobě vět, které obsahují 80 znaků. Tyto informace jsou v souboru rozděleny do pěti skupin:

- Start (S) section
- Global (G) section
- Directory (D) section
- Parameter (P) Data Section
- Termination (T) Section

Jelikož formát IGES pracuje pouze s geometrickými informacemi, může docházet při přenosu k nekompatibilitě.

Nevýhodou tohoto formátu je to, že nedokáže přenést kompletní 3D model, ale pouze povrch modelu složený z ploch. Tím pádem dochází ke ztrátě vnitřní struktury 3D modelu, ale také ke ztrátě návaznosti ploch tvořících povrch modelu, proto vznikají na povrchu nepřesnosti, které je nutné ručně odstranit.

Úlohu formátu IGES v současné době postupně přebírá formát STEP.

### 2.12.3 Formát STEP

STEP neboli (Standard for the Exchange of Product Model Data) je relativně novým formátem, který nahradil formát IGES. Kromě 2D i 3D dat dovede přenášet i jiné údaje o modelu. Tento formát udává při převodu 3D dat lepší výsledky než formát IGES.

STEP dovede přenést uzavřené 3D těleso složené z ploch, na rozdíl od IGES zde odpadají problémy s nespojitostmi přeneseného 3D tělesa. O vnitřní topologii tohoto modelu přijdeme i za použití tohoto formátu.

Import a export STEP formátu je součástí většiny dnešních CAD aplikací. Formát STEP obsahuje několik tzv. aplikačních protokolů.

STEP je v současné době jedním z nejvíce vyvíjených formátů pro převod dat CAD.

### 2.12.4 Formát DXF

DXF neboli (Drawing Exchange Format) je jeden ze základních CADovských formátů. Byl vyvinut firmou AUTODESK v roce 1982 a postupem času byl rozšířen do většiny CAD systémů. Tento formát je podporován i většinou kreslicích programů jako je např. Star Office, Corel Draw atd. V tomto formátu je možné reprezentovat technické výkresy v „drátové“ podobě ať už jen ve 2D, ale i ve 3D. V současné době je i přes své nedostatky je jedním z nejvíce používaných formátů pro výměnu CAD dat.

### 2.12.5 Formát DWG

DWG neboli (Drawing) je de facto uzavřený formát firmy AUTODESK, tím pádem se neřadí mezi standardní výměnné CAD formáty. Postupem času byl tento formát využíván i u jiných výrobců CAD systémů. Tito výrobci převedli formát DWG na otevřenou platformu, která se nazývá Alliance Open-DWG. Tento formát je v poslední době čím dál více podporován jednotlivými CAD systémy a je velice možné, že v budoucnu se bude jednat o nový standard v oblasti přenosu 2D dat.

### 2.13 Výhody používání CAD/CAM

Využití CAD/CAM systému umožnilo podstatně zmenšit dobu konstrukční přípravy. Díky projektování modelů na PC je celý proces návrhu značně zefektivněn. V dnešní době máme 2D a 3D CAD/CAM systémy. Výhodou je, že tyto softwary mezi sebou navzájem komunikují a mohou si vyměňovat navzájem data. Tyto systémy počítačového návrhu jsou využívány již od 70. let 20. století. Nyní jsou tyto softwary na vysoké úrovni s celou řadou funkcí, které značně usnadní návrh modelu.

Díky CAD/CAM softwarům je v podstatě v dnešní době umožněno získat 3D data skenováním reálných objektů a pak s nimi nadále pracovat. V další řadě, je možné vyexportovat přehlednou technickou dokumentaci, výrobní kalkulaci nákladů nebo různé 2D a 3D šablony. Tyto novodobé možnosti značně ušetří čas i práci vývojářům.

### 2.14 Porovnání nejpoužívanějších průmyslových CAD systémů.

V tomto srovnání jsou rozebrány jednotlivé výhody a nevýhody současně nejpoužívanějších CAD softwarů v průmyslu. Jedná se především o softwary CATIA V5 R21, AUTODESK Inventor Professional, SolidWorks 2009. Tyto softwary jsou obecně považovány za velice spolehlivé s poměrně vysokou funkcionalitou.

#### 2.14.1 CATIA V5 R21

Mezi hlavní výhody tohoto rozšířeného softwaru patří např. využívání plošného modelování, kde je výborná propojenost s modelem. Lze vytvářet velice složité prvky, které ostatní softwary nezvládávají např. složitá zaoblení atd. Mezi hlavní nevýhody patří především to, že je zde postrádána česká verze softwaru. Tento problém je částečně řešen za pomoci TECHNODATU, ale překlad je značně nepřesný a může uživatele spíše zmást. V další řadě je zde velice složité a nepřehledné uživatelské rozhraní, které působí na uživatele velice chaoticky. V neposlední řadě je zde postrádána studentská verze tohoto softwaru.

#### 2.14.2 AUTODESK Inventor Professional

Výrobce tohoto systému je firma AUTODESK. Mezi hlavní výhody patří např. kvalitní český překlad, velice přehledné ovládání programu, velice přehledná nápověda a na závěr i možnost využití studentské licence. Hlavními nevýhodami je především to, že jsou zde vyšší nároky na výkon PC, operační paměť a grafickou kartu. Je zde poměrně náročná realizace plošného modelování.

### 2.14.3 SolidWorks 2009

Výrobce tohoto softwaru je francouzská firma Dassault Systemes. Hlavními výhodami tohoto softwaru je především velice přehledné uživatelské rozhraní, možnost českého překladu, plošné modelování a kvalitně zpracovaná nápověda. Mezi hlavní nevýhody patří především to, že tento software neumožňuje některé náročné operace, které jsou v jiných softwarech možné, např. složité zaoblení a srážení, které zvládá např. CATIA.

### 3 Digitalizace robotické buňky pro tvorbu knihovny nástrojů a pracovních objektů

#### 3.1 ROBOGUIDE a CAD

Tato práce bude realizována za pomoci softwaru ROBOGUIDE a FREECAD. Tyto softwary mají poměrně širokou možnost využití. Prostředí ROBOGUIDE je možné využít pro OFF-LINE programování a simulaci robotů FANUC. Je zde možné vymodelovat celé pracoviště včetně všech objektů a reálně odsimulovat správnou funkčnost modelu. Následně je možné nahrát program z tohoto modelu do reálného robota. Simulační model nám prozradí mnoho potřebných informací jako je např. ověření dosahu jednotlivých poloh, zjišťování kolizních stavů v pracovním prostředí, správná funkčnost senzorů atd. Velkou výhodou je, že pokud máme přesné informace o reálném pracovišti tak je adjustace modelu velice přesná, rychlá, přehledná a pohodlná. Navíc OFF-LINE programování umožňuje práci vykonávat mimo pracoviště, tím pádem je možné programovat třeba i z pohodlí domova. Jelikož jak již bylo řečeno, veškerá simulace se provádí na PC v softwaru ROBOGUIDE.

V softwaru FREECAD je možné veškeré chybějící objekty vytvořit. Velice užitečnou vlastností těchto softwarů je to, že je možné tyto objekty poměrně snadno naimportovat do prostředí ROBOGUIDE. Tím pádem mohou být tyto objekty plnohodnotně využívány v prostředí ROBOGUIDE, jelikož tento software podporuje většinu CAD dat.

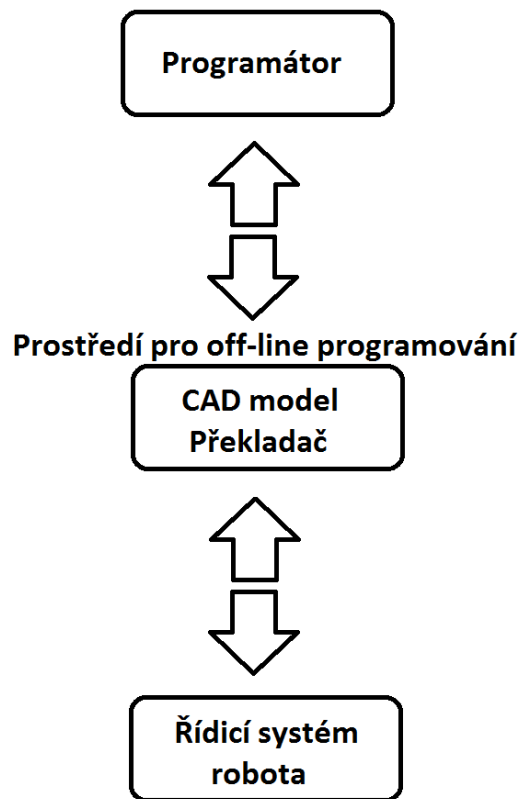
V této práci bude vytvořen základní program, který dokáže na reálném pracovišti svázat tyto dva softwary. V softwaru FREECAD založíme nový projekt, ve kterém vytvoříme námi potřebné objekty, se kterými budeme dále pracovat. Těmto objektům můžeme v prostředí ROBOGUIDE přidělit určité vlastnosti a parametry. Tyto naimportované objekty je možné používat v prostředí ROBOGUIDE jako např. Fixtures, Obstacles, Parts. Jak bude podrobně rozebráno v kapitole 3.6 CELLBROWSER, je možné v softwaru ROBOGUIDE vygenerování jednotlivých dílů, překážek a drah pro robota.

Jelikož nám prostředí ROBOGUIDE umožňuje některé základní objekty přímo vytvořit, tak v prostředí FREECAD budeme tvořit pouze objekty, které se v prostředí ROBOGUIDE využijí jako objekty typu Parts, tzn. jejich hrana bude sloužit jako dráha robota.

V námi vytvořeném simulačním modelu využijeme nakloněnou plexisklovou desku, která je v robotických laboratořích reálně obsažena. Tuto desku využijeme ke generování dílů, tzn. v softwaru ROBOGUIDE bude nadefinována jako objekt typu Fixtures což znamená, že se na této desce mohou generovat jednotlivé díly pro robota.

Pokud bude naším cílem např. nakreslení libovolného obrázku či textu na nakloněné plošině, využijeme software FREECAD kde jsou možnosti modelování téměř neomezené. Je zde možné zhotovit téměř libovolný technický výkres, který se dá do softwaru ROBOGUIDE naimportovat. Hrana tohoto objektu bude sloužit jako dráha pro koncový bod robota. Tímto jednoduchým krokem a vhodnou adjustací můžeme poměrně

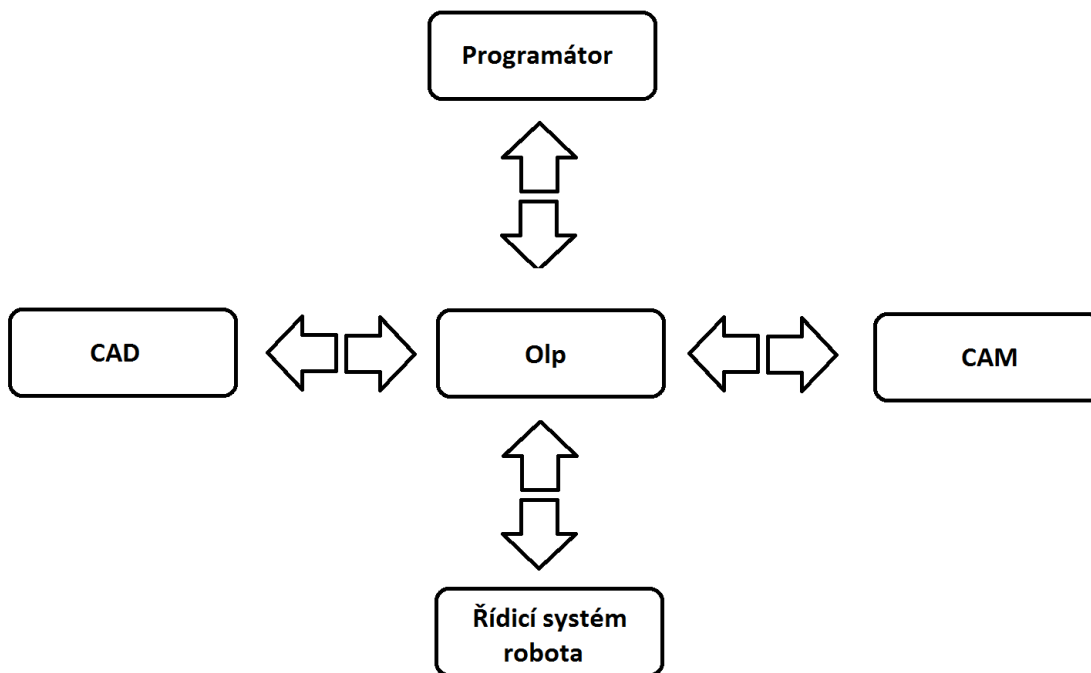
rychle a přesně zhotovit i zdánlivě složité dráhy pro robota, které by byli v ručním ON-LINE režimu takřka neproveditelné.



Obr. 20: OFF-LINE programování

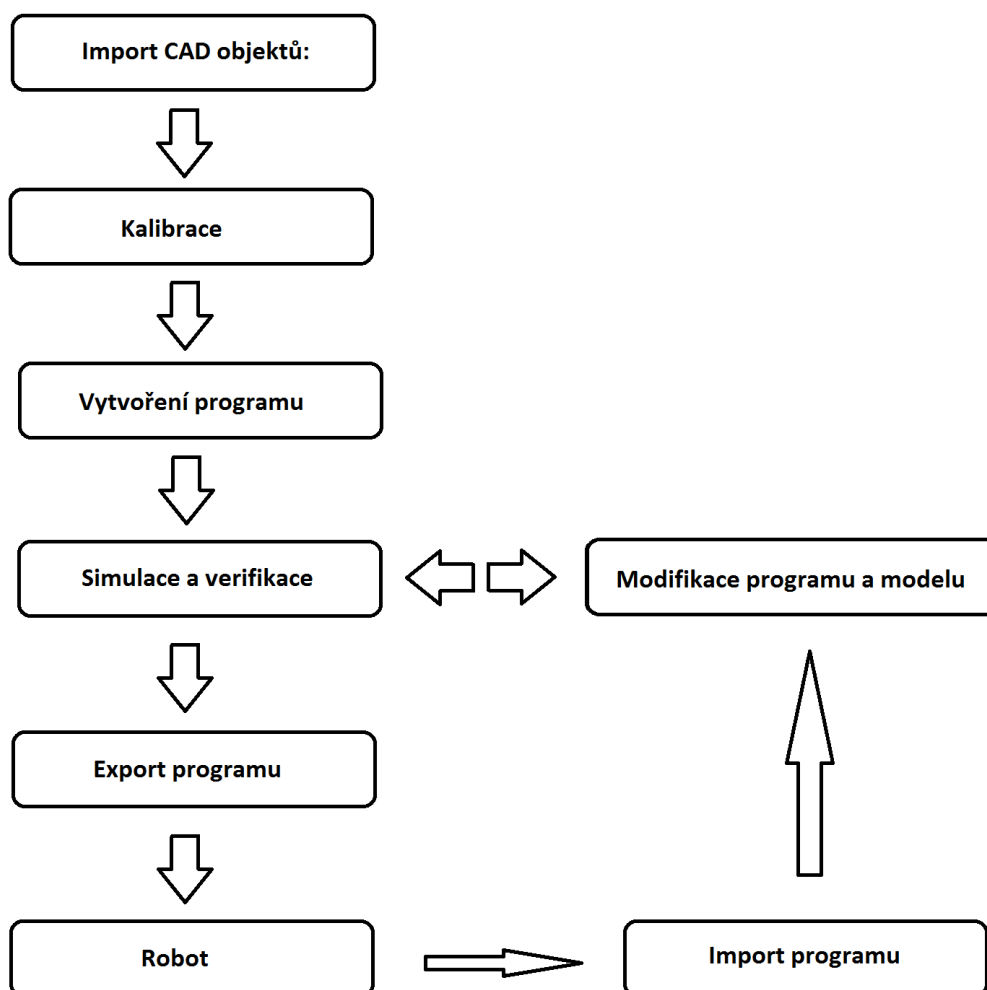
Jak již bylo řečeno OFF-LINE programování robotů je rozdílné od ON-LINE programování tím, že se programování uskutečňuje za pomoci 3D modelu, který je v našem případě vytvořen v prostředí ROBOGUIDE. Tento software potřebuje ke správné funkci překladač, díky kterému je možné veškerá data přetransformovat do podoby, která umožňuje naimportování těchto zdrojových kódů pro konkrétního reálného robota.

Prostředí ROBOGUIDE pracuje s CAD/CAM modelem kde si jednotlivé informace vyměňuje za pomoci knihoven objektů, které jsou do simulačního modelu naimportovány. Nejčastěji jsou v těchto knihovnách jednotlivé modely robotů, kontrolérů, výrobní díly, typy nejběžnějších překážek atd.



Obr. 21: Komunikace mezi CAD/CAM a OLP

Průběh celé simulace lze simulovat v reálném čase, a je tedy možné upravit a optimalizovat pracovní cyklus. Celý model musí být maximálně věrný realitě, aby byla oprava případných nepřesností a chyb pokud možno co nejmenší.



Obr. 22: Průběh OFF-LINE programování

### 3.2 Zaměření nástroje – TOOL FRAME

Nadefinování polohy koncového bodu robota je nezbytné pro další práci s robotem. Zajišťuje nám, že se souřadnice koncového bodu neodečítají od příruby ale až od koncového nástroje.

V praxi máme celkem 3 možnosti, jak toho můžeme docílit a to za pomoci třibodové metody (Three Point), šestibodové metody (Six point) a přímého zadání hodnot (Direct Entry).

Jako plnohodnotně dostačující se jeví základní třibodová metoda, která zjistí pouze pozici koncového bodu nikoliv jeho orientaci. Nastavení se provádí v MENU robota FANUC, kde zvolíme nabídku Tool Frame, kde jsme vyzváni k výběru metody zaměření nástroje.

Definování TPC se u třibodové metody provádí zpravidla tak, že za pomoci ručního ovládání najedeme na pevný bod ze tří různých směrů. Směr X směřuje ve směru nástroje kolmo k přírubě J6.



Hodnoty koncového bodu - TCP jsou vztaženy ke středu příruby, která se nachází na konci šesté osy značíme J6 (Joint).

Šestibodová metoda vychází ze třibodové metody s tím rozdílem, že se zde navíc definuje směr nástroje. Je zde nutné nadefinovat výchozí bod pro určení orientace a směr osy X a Z. Jelikož víme, že osa X u robotu FANUC směřuje ve směru nástroje, je nutné ji nadefinovat ručním pojezdem cca o 20 cm. Hodnotu uložíme, vrátíme se do počáteční pozice a totéž provedeme pro novou osu Z.

Na závěr je zde možnost zvaná Direct Entry – přímé zadání. Díky této možnosti můžeme přímo zadat konkrétní hodnoty, nejčastěji to jsou data z OFF-LINE modelů např. data ze softwaru ROBOGUIDE.

### **3.3 Zaměření souřadného systému – USER FRAMES / EXTERN TOOL**

Jedná se o nadefinování souřadného systému, který uživateli umožňuje se s robotem pohybovat např. na nakloněné desce. Je tedy možné jednotlivé osy nadefinovat tak, že jsou rovnoběžné s nakloněnou deskou, tím pádem nemusíme s robotem manipulovat podle nejběžnějšího souřadného systému WORLD, ale podle vlastního souřadného systému, který značně usnadní manipulaci s robotem.

Zaměření souřadného systému se provádí přes ruční ovladač (teachpendant), kde vstoupíme do jeho MENU a následně rozklikneme možnost SETUP, kde zvolíme možnost FRAMES. Poté rozklikneme možnost OTHER a zvolíme nabídku USER/RTCP, kde dále vybereme šipkami číslo položky, pod kterou se má náš USER FRAME uložit (1 až 20).

Stejně jako u zaměření nástroje i zde máme celkem 3 možnosti jak zaměřit vlastní souřadný systém. Tato možnost je realizovatelná díky třibodové metodě, čtyřbodové metodě a přímé metodě.

U všech metod je nezbytně nutné aktivovat nástroj, aby robot zajížděl souřadnice podle skutečného koncového bodu a ne podle středu konce příruby šesté osy J6.

Plně dostačující je třibodová metoda, kde se v první řadě zvolí počáteční bod, ze kterého bude robot vycházet a následně se uloží. Poté je nezbytně nutné nadefinovat osy X a Y. Od tohoto počátečního bodu jedeme cca 20 cm ve směru X, kde hodnotu uložíme a následně se vrátíme do počátečního bodu a to samé provedeme i pro směr Y. Tímto jsme si nadefinovali souřadný systém, ve kterém chceme, aby robot pracoval.

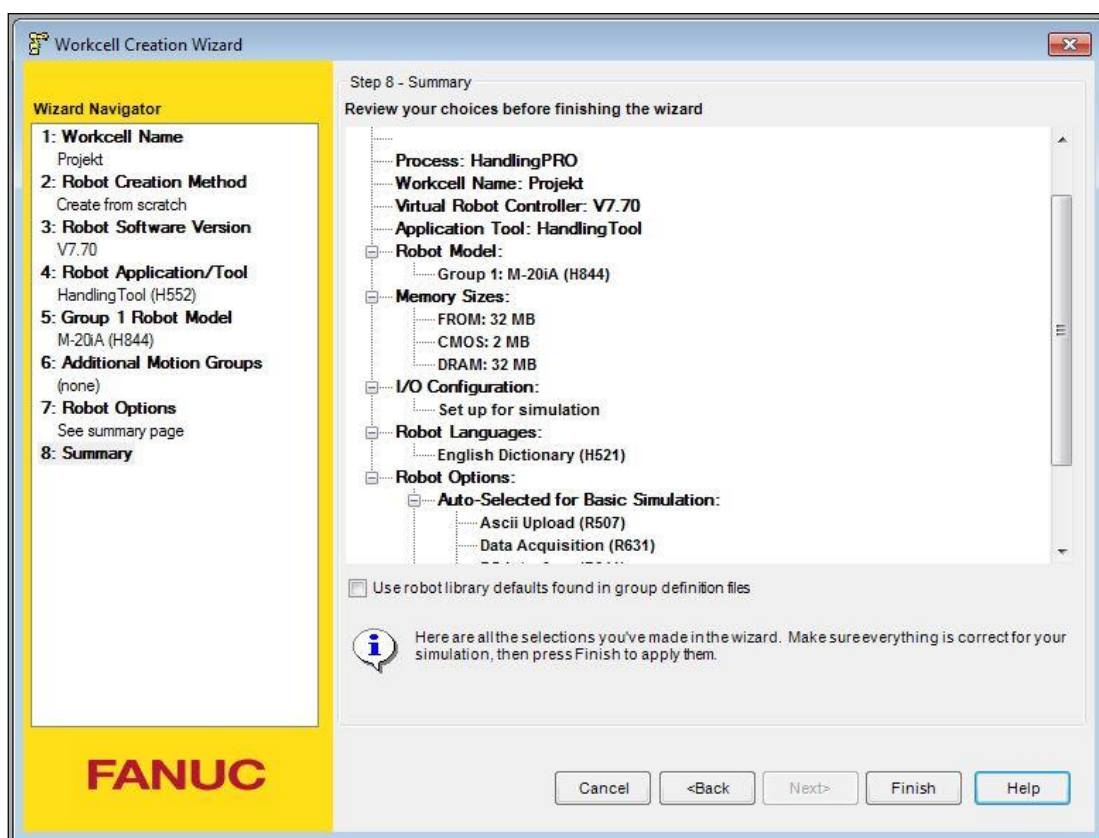
### **3.4 Založení nového projektu**

Při založení nového projektu v softwaru ROBOGUIDE je třeba dávat pozor na několik zásadních věcí. Nový projekt založíme tak, že klikneme na položku „New Cell“ dále se nás průvodce táže, jak by se měl tento nový projekt (výrobní cela) jmenovat. Po uvedení vhodného názvu, který popisuje naši aplikaci, můžeme pokračovat tlačítkem „next“ na další výběrové okno.

V dalším kroku se nás ROBOGUIDE ptá, jakou chceme použít softwarovou verzi pro robota FANUC. Ve firmě ŠKODA AUTO je zaveden standard verze: V7.63 – 7.6310.07.03 pokud bychom dali jinou verzi, tak by nebylo možné robota ovládat a pracovat s ním. Po vybrání příslušné volby pokračujeme opět tlačítkem „next“ na další výběrové okno.

V dalším kroku je zapotřebí vybrat konkrétní typ robota, se kterým pracujeme. V našem případě se jedná o robota FANUC M-20iA a proto tedy zaškrtneme možnost H844 M-20iA.

V dalších 2 krocích jsme pouze vyrozuměni s parametry vybraných voleb.



Obr. 23: Průvodce vytvořením nového projektu

### 3.5 Vytvoření modelu vývojového pracoviště

Prostředí ROBOGUIDE umožňuje uživateli na základě vstupních parametrů pracoviště zhotovit velice přesný model robotického zázemí. V prvním kroku je nezbytně nutné zvolit správný typ robota, který odpovídá naší skutečnosti.

V našem případě to byl robot od firmy FANUC s typovým označením M-20iA. Po založení nového projektu jsme dostali prakticky „holý“ model pouze s námi vybraným robotem.

V první řadě bylo nezbytně nutné zajistit správnou pozici robota. Ve školícím pracovišti je robot umístěn na zhruba 2 cm vysoké podložce a je úmyslně natočen vůči oplocence. Tyto malé nepodstatné detaily bylo třeba nutně zahrnout do modelu. Při jejich

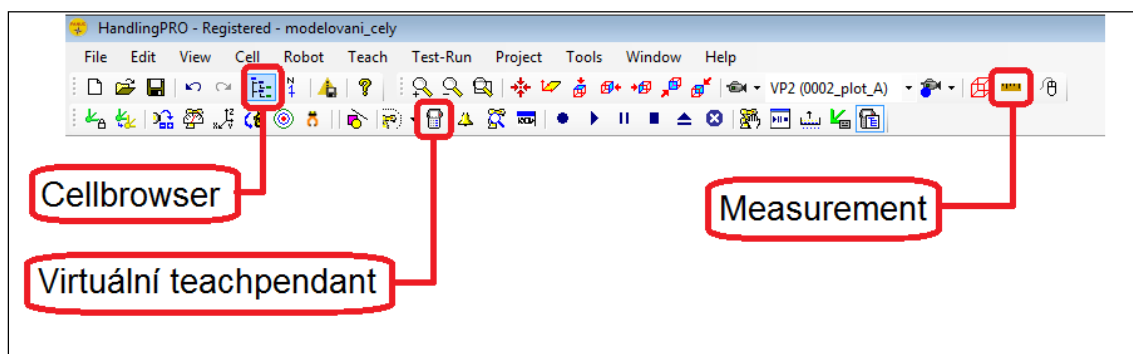
zanedbání by v případě programování data jednotlivých souřadnic nekorespondovaly a dostavila by se nám chyba ve značné nepřesnosti.

Tuto desku bylo třeba přesně nadefinovat jako první objekt v podskupině objektu Obstacles, což jsou veškeré objekty, které slouží jako překážka robota. Bylo nezbytně nutné zadat její rozměry, pozici a následně ji „svázat“ s robotem za pomoci volby „LUMP“, která umožňuje pevné spojení dvou či více objektů.

Po této základní „kalibraci“ bylo zapotřebí namodelovat základní pevné prvky tj. oplocenka, zeď, kontrolér. Bylo nezbytně nutné provést fyzické proměření pracoviště, aby byly zjištěny reálné souřadnice a rozměry těchto objektů.

Prostředí ROBOGUIDE má některé objekty předdefinované, to se týkalo zejména kontroléru, který bylo možné přesně nalézt v knihovně objektů Obstacles a následně ho podle jeho skutečných souřadnic vložit do modelu.

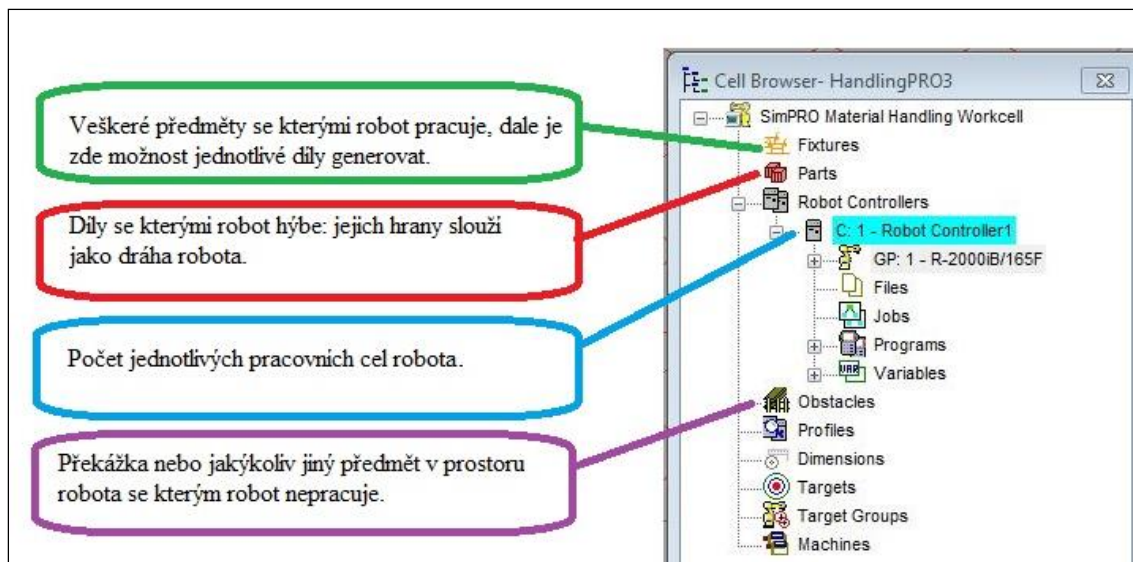
Většinu souřadnic modelu bylo možné proměřit přímo samotným robotem a zjistit tak skutečné vzdálenosti s prakticky minimální chybou. Na ostatní prvky jakými byly např. sloupky a oplocenka bylo zapotřebí použít pásmový metr, z důvodu malého dosahu robota [23].



Obr. 24: Panel nástrojů prostředí ROBOGUIDE [23]

### 3.6 Cellbrowser

Cellbrowser je základní nabídka prostředí ROBOGUIDE při tvorbě objektů a pracování s nimi. Díky tomuto formuláři máme přehledně zobrazeno, jaké jsme použili objekty a jakého jsou charakteru. Jak je již patrné z Obr. 25., je možné rozdělit objekty do několika základních skupin: Fixtures, Parts, Obstacles. Tyto objekty se podstatně liší svým charakterem. Objekty typu Fixtures, jsou de facto veškeré předměty, se kterými robot pracuje, ovšem s tou výhradou, že jsou schopny na sobě generovat nejružnější výrobní díly, aby bylo možné simulovat v modelu výrobní procesy [23].



Obr. 25: Cellbrowser [23]

Dále je zde položka Parts, kde hrana objektů slouží jako dráha pro robota. V položce Obstacles jsou veškeré objekty, se kterými nelze pracovat de facto překážky v pracovním prostoru robota.

V nabídce Cellbrowser můžeme spatřit všechny aktuálně použité objekty a podle potřeb je upravit. Nespornou výhodou Cellbrowseru je to, že pokud si uživatel ROBOGUIDE udělá při tvoření modelu určitý systém (např. vhodné názvy objektů), jsou veškeré úpravy na modelu velice rychlé, jednoduché a hlavně přehledné [23].

### 3.7 Reálný dosah robota

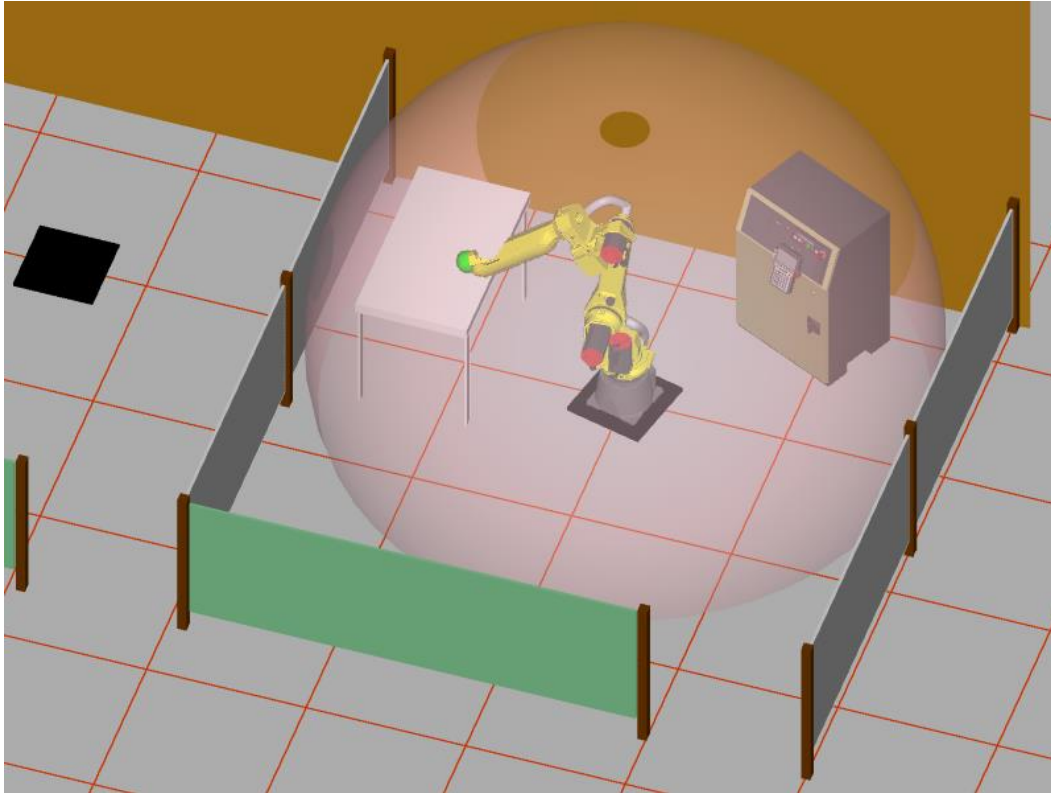
Reálný dosah robota je parametr, který by neměl být žádným způsobem podceňován a opomíjen hlavně z hlediska bezpečnosti. Každý průmyslový robot má určitou pracovní obálku, kam až je schopný se dostat. Na základě této informace jsme nuceni učinit určitá opatření.

V praxi je nezbytná „oplocenka“ robota, která má za cíl zabránit kontaktu osob s robotem za provozu a tím pádem zabránit případným zraněním. Často opomíjenou funkcí oplocenky je i ochrana osob před technologií provozovanou robotem. Charakter a provedení oplocenky musí odpovídat používané technologii a podřizuje se bezpečnostním předpisům. Jako zvláštní případy lze zmínit svařování laserem, řezání vodou, obloukové svařování, materiálový Handling při vyšších rychlostech apod.

V ideálním případě by mělo být ochranné ohrazení až za maximálním možným dosahem robota, aby v případě kolize nedošlo ke střetu robota s oplocenkou nebo jinými ochrannými prvky. Ovšem jak je patrné z obrázku, který věrně vystihuje reálné pracoviště, ne vždy tomu tak je.

V prostředí ROBOGUIDE je možné tuto obálku maximálního dosahu na základě technických parametrů od výrobce zobrazit a přesvědčit se, zdali jsme či nejsme v kolizním pásmu. Jak je patrné z obrázku, tak zde v případě kolize může robot nabourat do levé části oplocenky, stolu, kontroléru a zdi.

Jelikož se ovšem jedná o školící pracoviště, kde je téměř každý krok obsluhy pod dohledem školících pracovníků, není třeba se příliš tohoto „přízpusobení“ obávat. Navíc je robot provozován v ručním režimu a dosahuje pouze omezených rychlostí (do 25mm/s), které nemohou člověku ublížit. Pokud je robot přepnut do automatického režimu, kde se již pohybuje plnou produkční rychlostí, které mohou člověku ublížit, tak musí být striktně vyklizen vnitřní prostor, odzkoušena správnost programu v ručním režimu a zajištěna bezpečnostní pojistka na dveřích oplocenky [23].



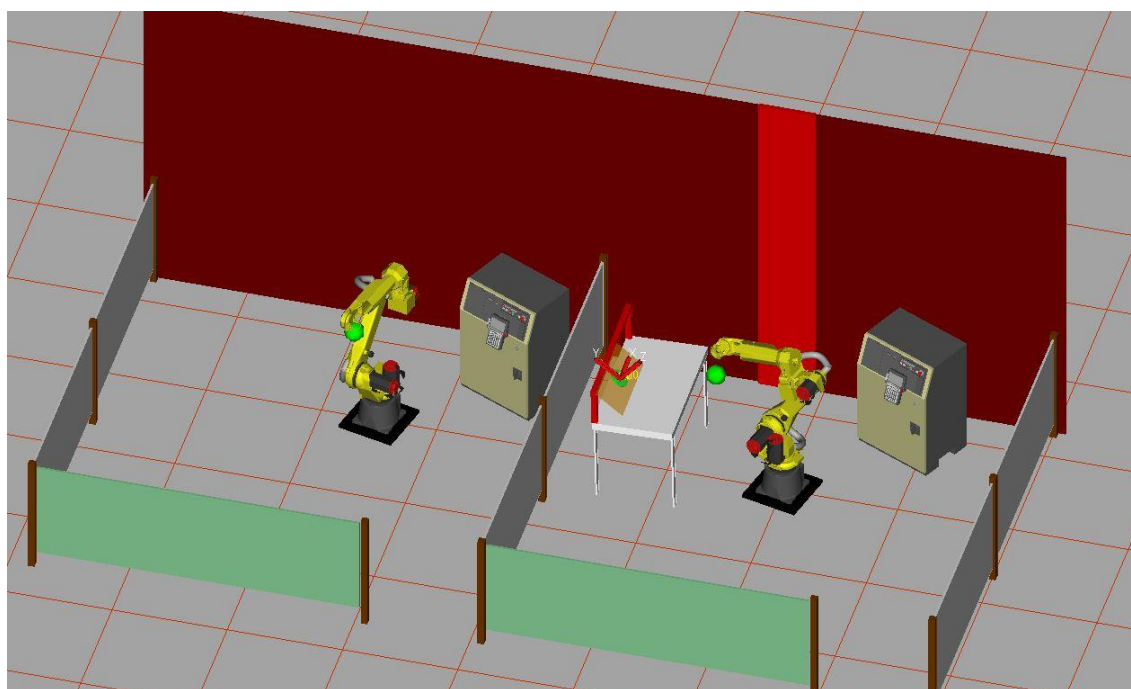
Obr. 26: Obálka dosahu robota ve vývojového prostředí ROBOGUIDE [23]

### 3.8 Model reálného pracoviště

Na obrázku je možné spatřit reálný model pracoviště, který je možný porovnat s realitou. Cílem bylo vytvořit objekty, které se budou pokud možno co nejvíce blížit realitě. Na obrázku je možné vidět dvě vymodelované cely s robotem typu FANUC M-20iA, které jsou paralelně vedle sebe. Objekty jsou namodelovány s mm odchylkami, jejich přesná adjustace je rovněž součástí této diplomové práce [23].



Obr. 27: Robotická cela ŠKODA ACADEMY [23]



Obr. 28: Model robotické cely

### 3.9 Modelování koncového nástroje TPC

Vzhledem k demonstraci modelu a následným simulacím, bylo součástí této práce i vymodelování koncového nástroje TCP.

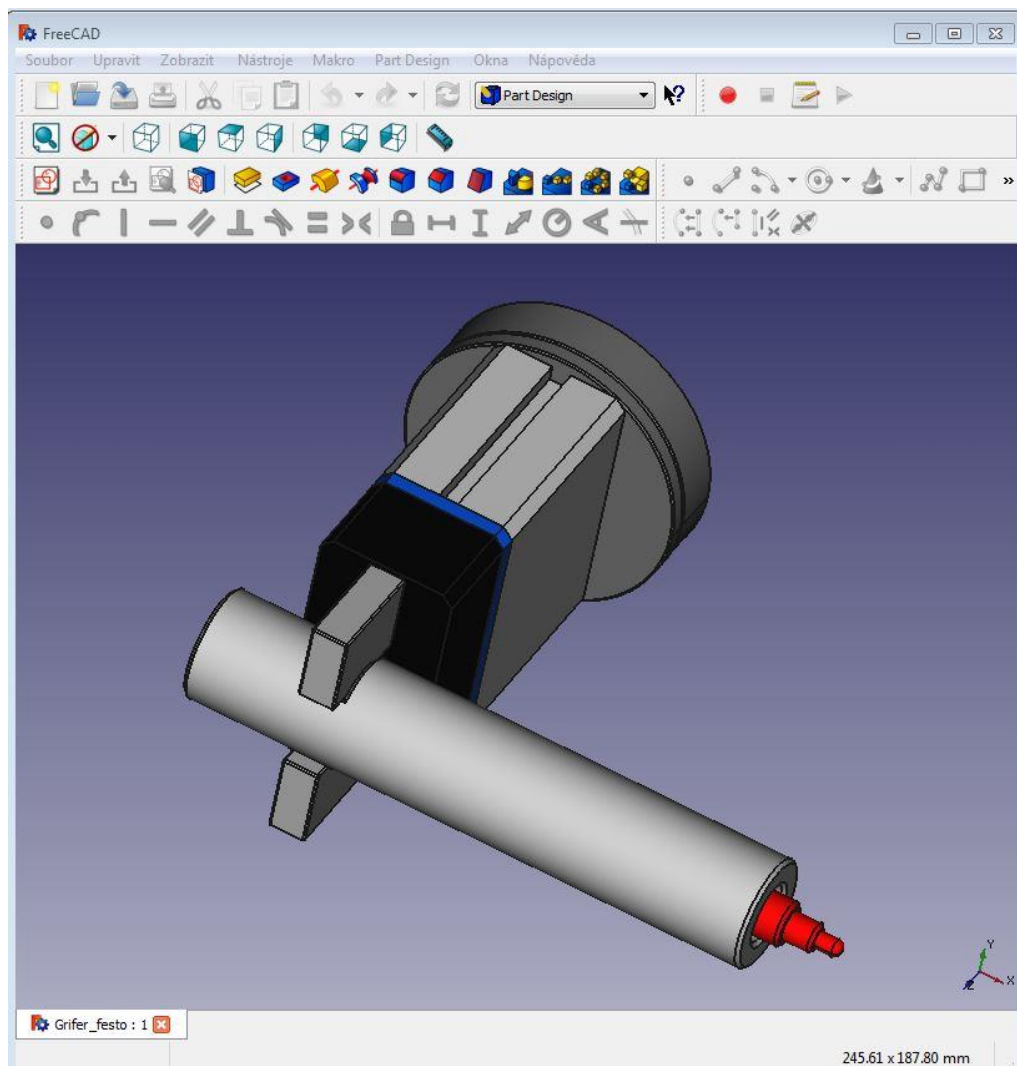
Tento pracovní nástroj, který je v našem případě konkrétně FIXA vložená v držáku zvaný GRIFER byl vymodelován v softwaru FREECAD a následně naimportován do softwaru ROBOGUIDE. Tento nástroj byl sestaven z několika dílčích částí, které bylo třeba spolu zkombinovat a následně svázat. Poté byl nutný export do formátu IGES, jelikož žádný jiný formát není s ROBOGUIDE plně kompatibilní. Při importu formátu STEP bylo možné výrobek naimportovat, ovšem nebylo možné s tímto formátem vůbec pracovat. Při importu dat ztratil technický výkres data o barevném rozložení.

Následně se v nabídce CELLBROWSER otevřel požadovaný robot do, kterého chceme tento CAD model TCP naimportovat a zvolí se „IMPORT CAD FILE“.

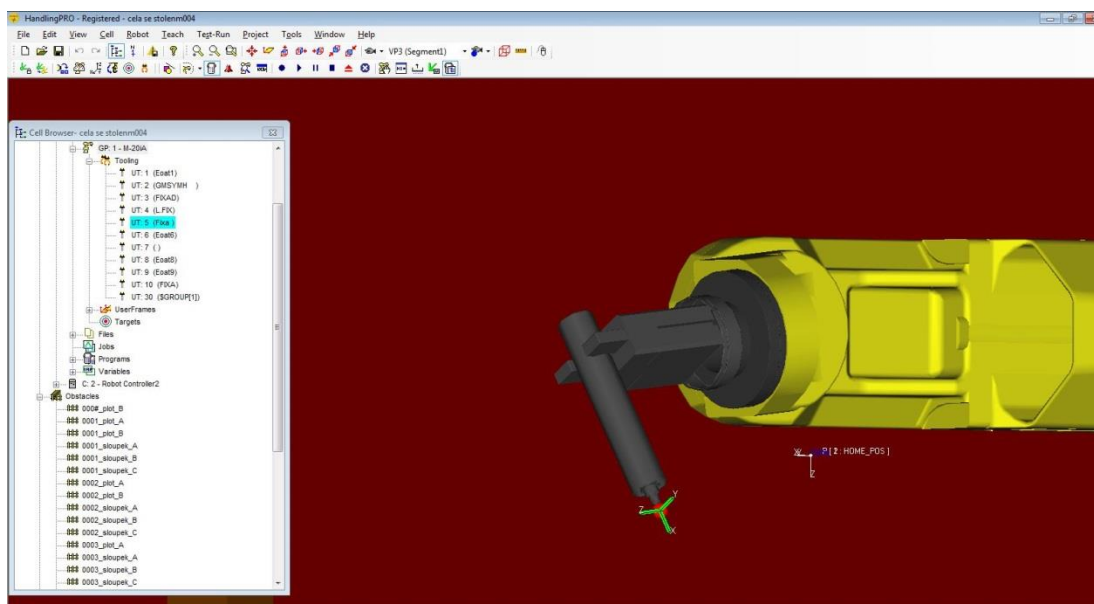


Obr. 29: Reálný nástroj robota FANUC M-20iA





Obr. 30: Modelovaný nástroj v softwaru FreeCad



Obr. 31: Nainportovaný nástroj do prostředí ROBOGUIDE

## 4 Adjustace modelu

### 4.1 Adjustace výrobního stolu v software ROBOGUIDE

Při výrobě tohoto výrobního stolu bylo nutné zajistit přesné rozměry jednotlivých částí stolu. Tento stůl bohužel nenalezneme v knihovně ROBOGUIDE čili je nutné ho za pomoci jednotlivých objektů vytvořit. Prvním krokem bylo rozdělit stůl na jednoduché konstrukční části tzn. na čtyři nosné sloupky a jednu hlavní desku. Při vytvoření těchto jednotlivých částí je nutné tyto objekty dostat do vzájemné správné polohy a následně je mezi sebou pevně spárovat za pomoci možnosti LUMP, která z více objektů vytvoří jeden.

U tohoto objektu je prozatím jedno, jestli je nadefinován jako OBSTACLE nebo FIXTURES (3.6 Cellbrowser). V tuto chvíli zbývá pouze vytvoření pouze plexisklové desky, která už musí být definována jako FIXTURES a její nosná konstrukce, která je typu OBSTACLE. Po správném napolohování a natočení desky tyto objekty opět společně spárujeme za pomoci možnosti LUMP v jeden výsledný objekt.

Nyní je nutné zjistit přesné reálné souřadnice středu plexisklové desky. Tyto souřadnice poslouží v modelu jako počáteční bod. V tuto chvíli se pouze změří robotem skutečná vzdálenost od robota a hodnota se nastaví v modelu. Tímto krokem zajistíme minimální chybu. Navíc je toto opatření vhodné i při změně pracovního prostředí např. pozice stolu, jelikož tento stůl není nikterak připevněn a tím pádem je možné téměř vždy zajistit přesnou pozici.

### 4.2 Kontrola kolizních stavů

Pokud je simulační model v prostředí ROBOGUIDE zcela hotov, nezbyvá nám nic jiného, než provést odladění chybových stavů aby nedošlo ke škodám na reálném pracovišti. V první řadě je nezbytně nutné odladit simulační model v softwaru ROBOGUIDE, kde zjistíme, zdali se robot pohybuje v námi definovaném prostoru.

Pokud je vše v pořádku můžeme námi vytvořený skript naimportovat do robota. Při prvním spuštění programu musíme být maximálně ostražití, jelikož skutečnost se může vždy oproti simulačnímu modelu značně lišit. Je tedy nezbytně nutné při prvním spuštění robota dbát na dodržení co nejmenší pohybové rychlosti, jelikož může dojít k poškození pohyblivých hadiček tlakového vzduchu, přírodních kabelů atd. S těmito možnostmi OFF-LINE simulace nepočítá a je tedy nutné v případě největší potřeby program zastavit, aby nedošlo k poničení jednotlivých konstrukčních částí.

### 4.3 Kontrola kolizí v softwaru ROBOGUIDE

Jak již bylo řečeno v prostředí ROBOGUIDE je nezbytně nutné nadefinovat objekty podle jejich typu zjistit, zdali se jedná o překážku, výrobní díl atd. Na základě těchto informací je tento software schopný detekovat kolize s pevnými objekty.

Pokud na základě vykonaného programu dojde ke stavu, kdy robot plánuje trajektorii v místě překážky, tak jsme okamžitě vyrozuměni.

Většinou je vždy realita jiná než model, ať už na základě drobných měřících nepřesností, špatných hodnot z technické dokumentace nebo i případnou drobnou změnou

pracovního prostředí. Na základě těchto předpokladů je nutné přistupovat k odladění programu maximálně ostražitě. Pokud nám výsledný skript jeví známky nepřesností, nastává zde zpětná vazba pro programátora, který zajistí import nahraného programu zpátky do prostředí ROBOGUIDE a následně se provádí vhodná modifikace. Tyto úpravy by měly být prováděny na základě reálného chování robota.

Velice zřídka se stává, že je program po prvním exportu do robota finálně přesný. Většinou se při spuštění programu objeví drobné odchylky, které mohou být od několika mm až do několika cm.

## 4.4 Kalibrace pracovní buňky

### 4.4.1 Obecné informace

System ROBOGUIDE umožňuje vykonat proceduru na kalibraci naší virtuální pracovní buňky s reálným, fyzickým pracovištěm a zařízeními v buňce. Pokud potřebujeme importovat/exportovat programy do a z prostředí OFF-LINE softwaru ROBOGUIDE, je důležité, aby náš model pracovní buňky přesně odpovídal reálné pracovní buňce.

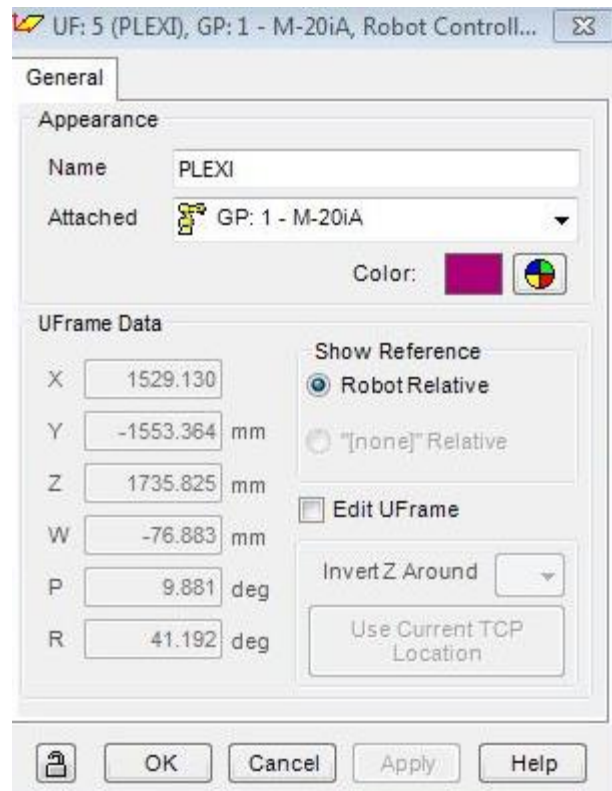
Následující body nám umožní lépe pochopit přesun programů mezi systémem ROBOGUIDE a reálným pracovištěm:

- Programy na skutečném robotu se chovají stejně jako v prostředí ROBOGUIDE. Nejsou zde třeba žádné konverze a úpravy programu a je tedy možné přesouvat programy přímo z ROBOGUIDE do robota. Pokud pracovní buňka v ROBOGUIDE není zkalibrovaná s reálným pracovištěm, může se zdát, že programy nebudou běžet správně.
- Pokud je skutečná buňka rozdílná od simulace v OFF-LINE softwaru, tak se naučené pozice nemusí jevit jako správné.
- Pomocí kalibrační procedury se pokusíme umístit pozice objektů v simulačním softwaru tak, jako se nacházejí v reálné pracovní buňce. Kalibrace pracovní buňky se provede tak, že vytvoříte program, kde naučíte 3 body v reálném pracovišti a potom naučíte ty samé 3 body v simulačním softwaru ROBOGUIDE. Objekty v ROBOGUIDE se následně posunou na základě vypočítaných rozdílů mezi naučenými body.
- Používáním UFRAME se přesun pohybových programů z/do OFF-LINU značně zjednoduší. Pokud jsou programy relativně vztažené k objektu v prostředí ROBOGUIDE, který je spojený s UFRAME robota, tak se jednotlivé souřadnice automaticky při kalibraci posunou.

### 4.4.2 Postup pro kalibraci objektů

Aby se pozice, které se mají posunout, posunovali společně s objektem, musí být objekt nadefinovaný pomocí USERFRAME.

V tomto příkladu jsem, jako objekt použil ze skupiny Fixture (Deska stolu plex). Pro kontrolu je nezbytně nutné ověřit následující nastavení:



Obr. 32: Nastavení UFRAME

Vybereme UFRAME, který chceme přiřadit k objektu v buňce. V tabulce základních nastavení zvolíme objekt v buňce a přiřadíme mu UFRAME. Následně potvrdíme „Apply“.

V pracovní buňce se může nacházet více objektů, kalibraci však stačí vykonat jen na kritických objektech. Jsou to ty objekty, vůči kterým vykonáváme programy např., místo kde odebíráme díly či stroje, včetně stroje, kde vkládáme kusy atd. V našem případě se jedná o plexisklovou desku na, které bude náš program realizován.

## Kalibrace je rozdělena do pěti základních bodů

### A) Otevřeme stránku s vlastnostmi požadovaného objektu (3.6 Cellbrowser).

Můžeme kalibrovat Přípravky, Překážky nebo Mechanismy. Součástky typu (Parts) přiřazené k přípravkům se dají kalibrovat na stránce vlastnosti přípravku.

### B) Zvolte záložku kalibrace „Calibration“

Ta má políčka, které umožňují kalibraci objektu v pracovní buňce. Vyberte objekt, který, se má zkalibrovat v dropdown menu „Object to be Calibrated“ např. náš objekt typu „Fixture“ (deska stolu plex).

## Metody kalibrace

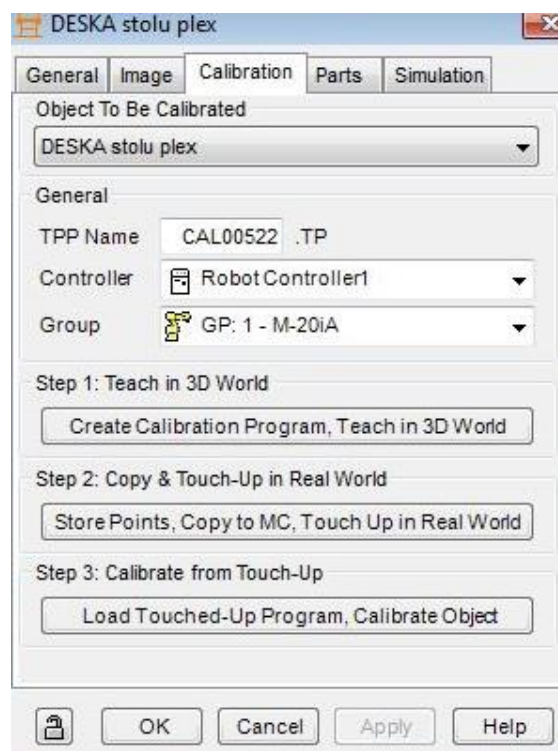
Je možné použít klasický postup - touch up za pomoci Teach pendantu a nebo metodu přímého zadání „Direct Entry Method“.

### Pro Direct Entry můžeme použít

**Offsets** – je možné použít offsety objektu, které mohou kalibrovat odchylku, která vznikla na základě drobných nepřesností měření při tvorbě modelu. Konkrétně řečeno např. jednotlivé vzdálenosti mezi pracovním nástrojem TPC a výrobním dílem v jednotlivých osách.

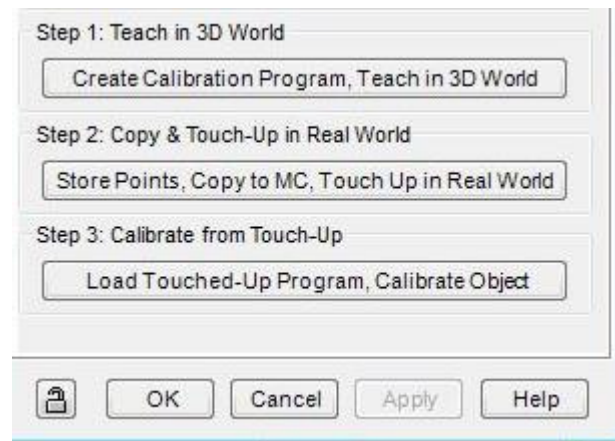
**Wiew preview** – Pokud je zaškrtnuté, zobrazí se nám preview objektu na kalibrované pozici

**Calibrate Object** – Pokud je potvrzené, objekt se zkalibruje a posune.



Obr. 33: Kalibrace objektů

### C) Krok 1 – Teaching v 3D



Obr. 34: 3 kalibrační kroky

Ve vytvořeném kalibračním programu umístíme 3 body v prostředí ROBOGUIDE tak, aby se později dali upravit i na reálném robotu.

Následně je potřeba upravit tyto jednotlivé body, které se zapíší do programu. Postup je takový, že se zajedou tyto 3 body na objektu, který chceme kalibrovat. V ideálním případě je vhodné zaměřit rohy objektu, tam máme totiž větší jistotu, že jsme najeli prakticky na „stejně“ místo. Kalibrační program vyhotovíme jednak v prostředí ROBOGUIDE ale také i za pomoci uživatelském panelu s reálným robotem. Tyto oba dva programy software následně vyhodnotí a následně skalibruje.

### D) Krok 2 – Uložení a Touch Up reálných objektů

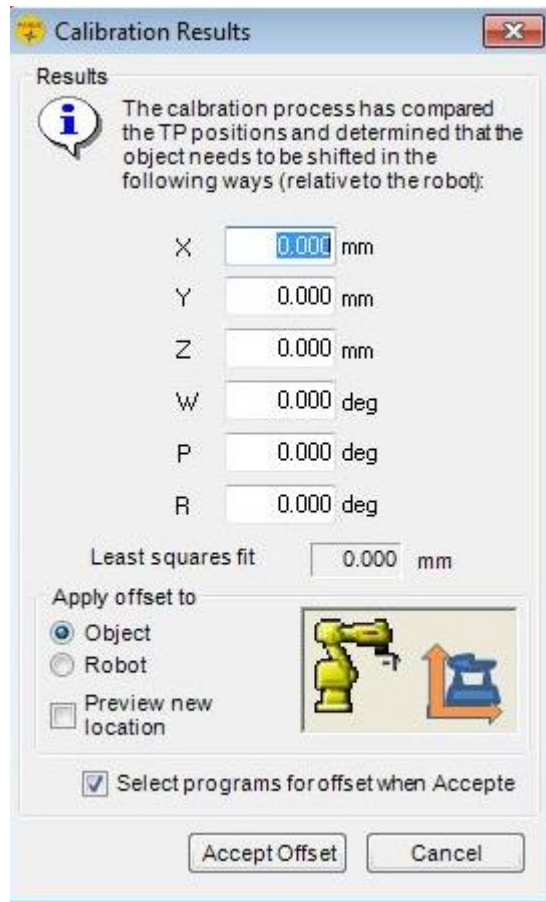
V tuto chvíli je nutné uložit naučený program. Zobrazí se hláška, kde nás ROBOGUIDE informuje o umístění programu a jeho názvu.

Následně je program zkopírován na USB flash disk a nahrán do reálného robota. V tuto chvíli je nutné vykonání rutiny „Touch Up“ objektu na reálném pracovišti a následně je nutné opět program uložit na USB flash disk a přepsat původní program v adresáři kde byl uložený.

Následně přepíšeme původní program novým. Pro ověření můžeme spustit nový program a uvidíme rozdíl mezi naprogramovanou dráhou a objektem.

### E) Krok 3 – Kalibrace

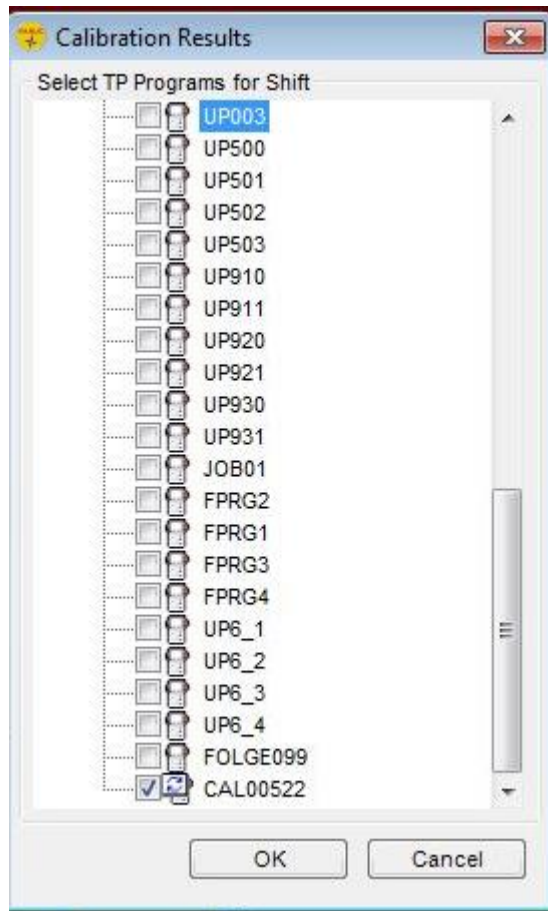
Když klikneme na Krok 3 – ROBOGUIDE načítá kalibrační údaje ze souboru a porovná původně naučené 3D body získané v OFF-LINE módu se skutečným – fyzickým pracovištěm. Použije se algoritmus výpočtu metodou nejmenších čtverců. Zobrazí se nám následující okno (Obr. 35:, Obr. 35:) Výsledky kalibrace



Obr. 35: Výsledek kalibrace

ROBOGUIDE porovnal rozdíly a zjistil offsety v jednotlivých souřadnicích a natočení přípravku, které jsou ve výsledcích kalibrace zobrazeny. Dále zobrazí hodnotu nejmenšího čtverce v nabídce s názvem „Least squares fit“. Nyní si můžeme vybrat, zda daný offset použijeme na relativní posunutí objektu, nebo objekt ponecháme a posuneme základnu robota viz možnost „Apply offset to“. Před kalibrací můžeme zobrazit náhled objektu po kalibraci. Nyní stačí potvrdit kalibraci akceptováním offsetu „Accept offset“.

Po akceptování offsetu se zobrazí okno, kde si můžeme dané změny promítnout na zvolené programy v řídicí jednotce nebo úplně na všechny programy. Po potvrzení volby se posunou všechny zvolené programy. Tímto krokem je kalibrace objektu úspěšně dokončena.



Obr. 36: Výsledky kalibrace

#### 4.4.3 Postup pro kalibraci součástky (Part) umístěné v přípravku (Fixture)

Kalibrace součástky probíhá způsobem, že kalibrujeme polohu součástky relativně vůči poloze přípravku ve, které je součástka uložena.

Pokud se součástka vyskytuje ve více přípravcích, můžeme kalibrovat každou součástku zvlášť pro každý přípravek.

##### Postup

Otevřeme okno s vlastnostmi přípravku, na kterém je umístěná součástka. Následně klikněte na záložku „Calibration“.

Vybereme součástku, u které chceme danou pozici kalibrovat (Obr. 33:), kde pouze zvolíme nabídku „Object To Be Calibrated“ a zvolíme požadovaný objekt ke kalibraci.

Následně vykonáme Krok 1,2 a 3 stejně jako v předchozím případě a kalibrujeme pozici dílu v přípravku.



#### 4.4.4 Postup pro kalibraci metodou přímého zadání Offsetu

##### Obecný popis

V této metodě kalibrace uživatel může zadat offsety přímo a kalibrovat objekty. Offset by měl být relativní vůči originální pozici objektu a jeho aktuální pozici. Tato kalibrační metoda podporuje posunutí bodů „shifting“ TP programů, které jsou uloženy s referencí vůči „nulovému bodu“ robota a nebo vzhledem k FRAME pracoviště.

Tento způsob je výhodné použít v případech, když uživatel pozná anebo změří offsety a zadá je přímo pro posunutí jednotlivých bodů. Tato metoda neumožňuje možnost kalibrovat a přesunout polohu robota.

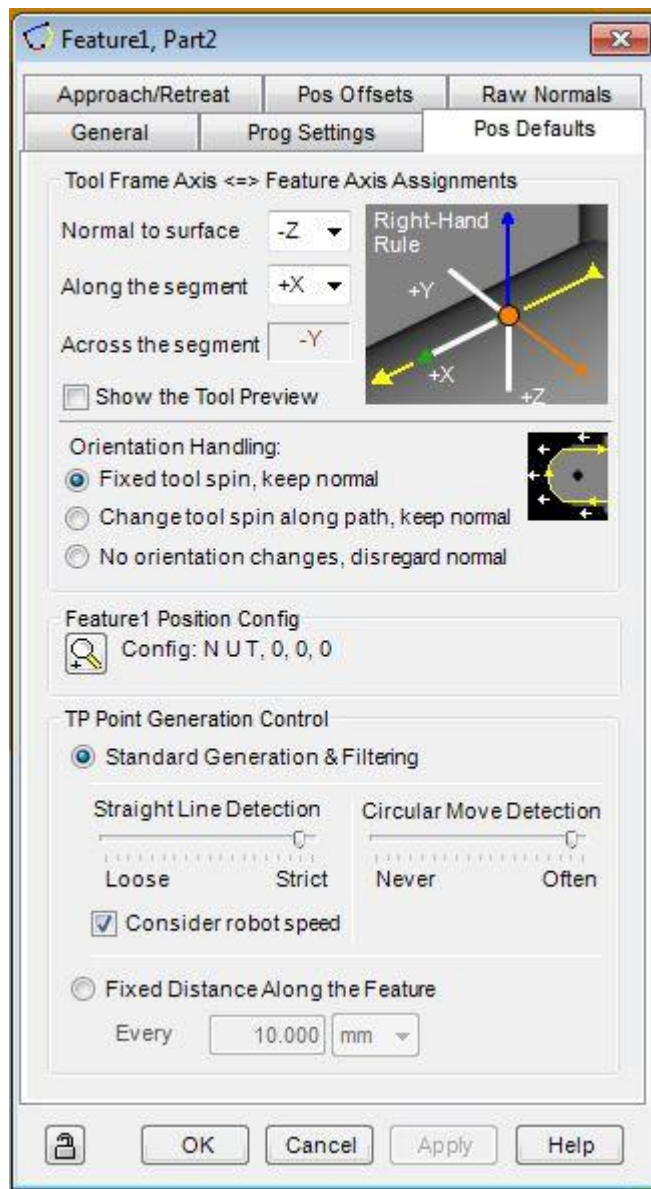
##### Postup

Otevřeme okno s vlastnostmi objektu, který chceme zkalibrovat, např. náš objekt typu Fixtures Deska stolu plex. A následně zvolíme záložku kalibrace a jako metodu kalibrace na přímé zadání „Direct Entry Method“. Poté do políček offsetu zadáme požadované posunutí a stiskneme tlačítko „Calibrate Object“ a vybereme veškeré programy, které se mají posunout. Kalibraci dokončíme potvrzením tlačítkem OK.

#### 4.5 Adjustace modelu

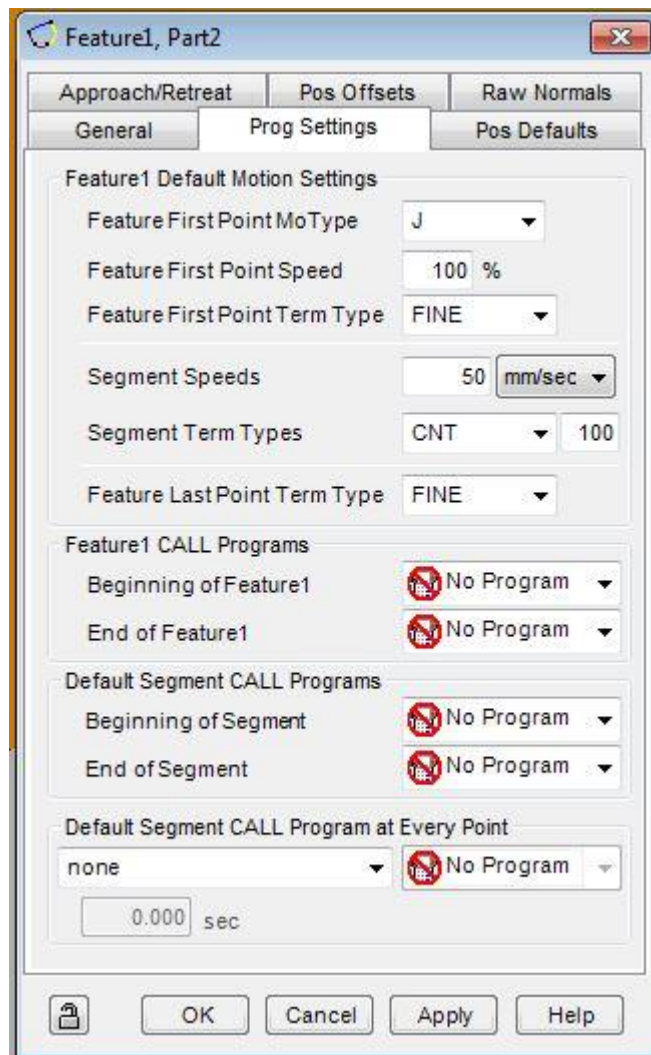
V softwaru ROBOGUIDE bylo nezbytně nutné zvolit správný souřadný systém koncového bodu. V tomto programu se automaticky pracuje s pracovním směrem nástroje ve směru osy Z, zatímco podle interních směrnic, kterým podléhá programování robotů ve ŠKODA AUTO se předpokládá pracovní směr nástroje ve směru X.

Přesná kalibrace směru nástroje byla prováděna v nabídce „Pos Defaults“ Obr. 37:



Obr. 37: Nastavení FRAME robota

V další řadě je nutné nastavit jakým druhem pohybu a rychlostí má robot najet do prvního bodu. Co se týče druhu pohybu, můžeme zde ponechat JOINTový pohyb, jelikož v našem případě nehrozí žádný kolizní stav robota s okolím. Rychlost je možné nechat na 100%.(Obr. 38:)



Obr. 38: Konfigurace pohybu

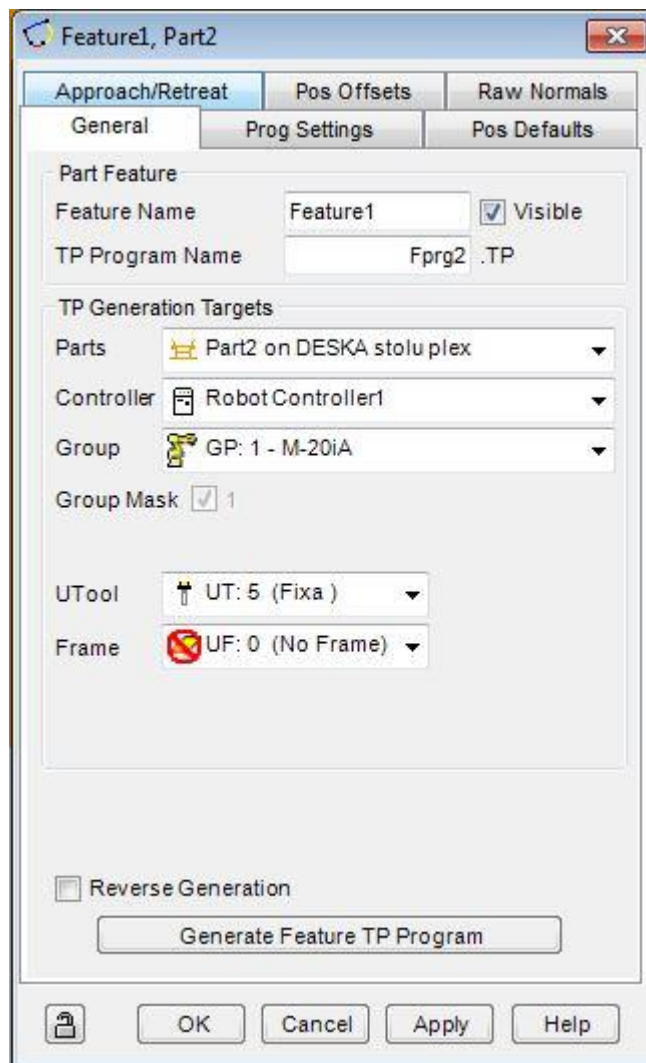
V další nabídce je nutné zadat název programu, který bude vygenerován. Tento skript je ve formátu \*.TP, což je standardní formát, který je podporován většinou průmyslových robotů. Přenos programu můžeme zrealizovat buď za pomoci ethernetové sítě nebo nahráním ručně za pomoci FLASH disku. Co se týče ostatních nastavení, je zde nutné zadat, na jakém objektu se budou generovat díly. V našem případě se jedná o plexisklovou desku.

V další řadě je třeba vybrat vhodný kontrolér a typ robota. Toto nastavení by mělo být správné, jelikož tyto parametry byly vybírány při vytváření projektu. Jedním z důvodů tohoto nastavení je to, že pokud máme v modelu více robotů, tak je díky této předvolbě snazší rozdělení úloh mezi jednotlivé roboty.

V neposlední řadě se zde určí FRAME pracovního nástroje. V našem případě se jedná o FRAME nástroje zvaného Fixa, kde:

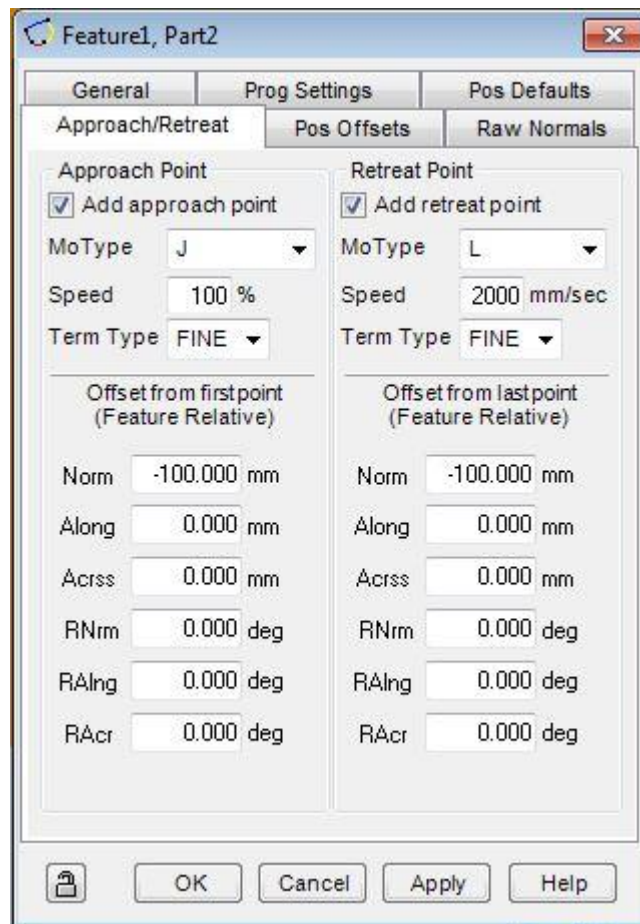
- $X = -142,4$
- $Y = 0$
- $Z = 115,8$
- $W = -180^\circ$
- $P = 90^\circ$
- $R = 0^\circ$

Dále je nutné zvolit souřadný systém, ve kterém bude robot jezdit. Pokud je zvolena nabídka NO FRAME tak robot zajiždí souřadnice v systému WORLD.



Obr. 39: Konfigurace pohybu

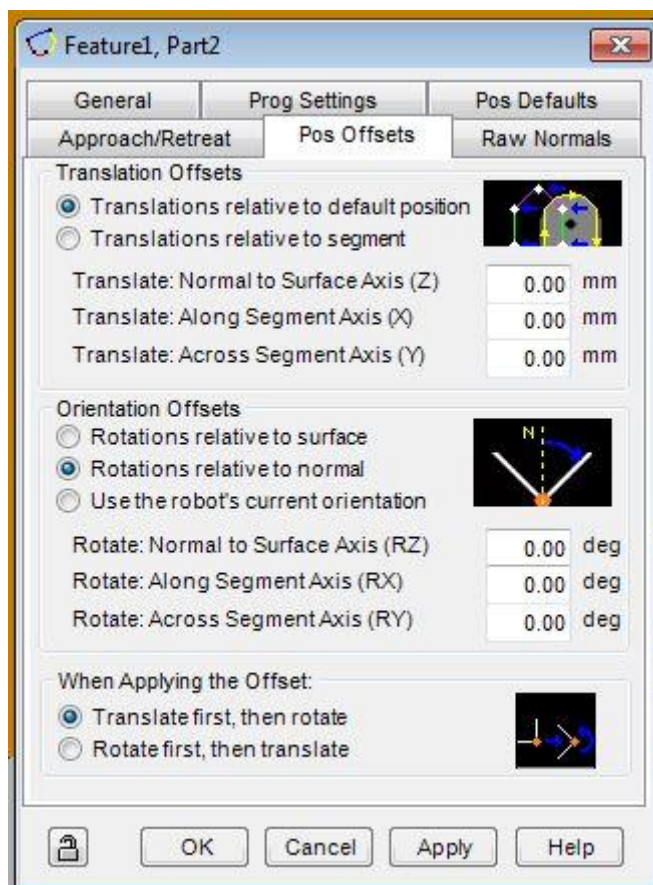
V této nabídce se volí konfigurace přístupového a koncového bodu. Dále je možné vybrat požadovaný typ pohybu JOINT, LIN nebo CIRC. A v neposlední řadě rychlost tohoto pohybu.



Obr. 40: Konfigurace přístupového a koncové bodu

Ruční nastavení offsetů v jednotlivých osách nebo rotacích je možné provést ručně v nabídce „Pos Offsets“ (Obr. 41:). Tyto hodnoty je vhodné měnit pouze tehdy, pokud máme program plně odladěný a jeví nám známky pouze drobných nepřesností, které jsme schopni změřit či jakkoliv určit.

Tyto jednotlivé vzdálenosti neboli nepřesnosti programu se defaultně přičtou či odečtou jako konstanty podle potřeby tak, aby byl koncový nástroj TCP v požadované pozici pouze s minimální chybou.



Obr. 41: Nastavení offsetů

## 4.6 Vygenerování dráhy podle CAD objektu

Prostředí ROBOGUIDE nabízí možnost generovat trajektorie robota z předlohy. Tou předlohou musí být objekty typu obrobek (part). Na takovémto obrobku lze body pro dráhu určovat několika způsoby. Mohou to být v podstatě libovolné body na povrchu, ke kterým ROBOGUIDE naorientuje v žádaném předdefinovaném směru TCP robota. Podstatně jednodušeji se dráha generuje na hranách a rozích obrobku. K tomu aby ROBOGUIDE rozpoznal hrany, se osvědčil formát IGES pro import CAD modelů obrobků. S formáty STEP a STL jsme měli problémy.

Pro uložení obrobku do robotické cely slouží tzv. přípravky (Fixtures), na kterých se obrobky mohou „vyskytovat“.

Výsledná trajektorie pohybu robota obsahuje i „příletové“ a „odletové“ body od obrobku, všechny body na obrobku mají jasně definovanou orientaci, rychlosti a druh pohybu.

Takovouto trajektorii je potom možné uložit do programu a ten přenést do skutečného robota.

## 4.7 Praktický výstup práce

Praktickým výstupem této práce bylo zdigitalizování reálné robotické laboratoře ve firmě ŠKODA AUTO (Obr. 27:), která se nachází v Mladé Boleslavi na ŠKODA AUTO ACADEMY. V první řadě bylo zapotřebí tuto robotickou buňku přesně zanalyzovat a následně změřit. Tyto parametry byly posléze využity jako vstupní data pro digitalizaci této buňky v prostředí ROBOGUIDE (Obr. 28:).

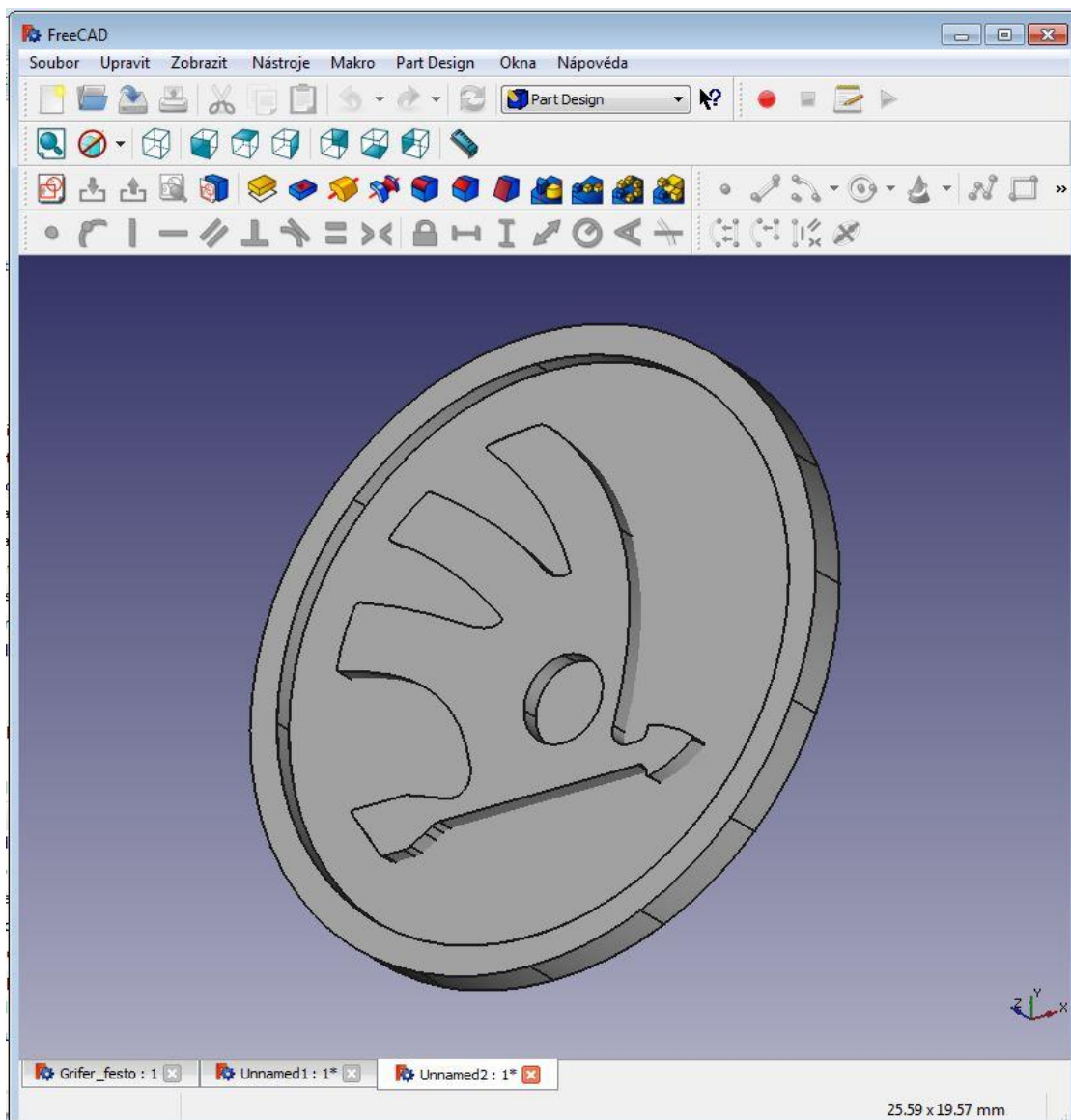
Na základě tohoto „věrného“ modelu pracoviště bylo zapotřebí vytvořit v tomto OFF-LINE prostředí program, který by splňoval funkčnost jednak v prostředí ROBOGUIDE, ale také i na reálném robotu.

Prostředí ROBOGUIDE umožňuje na základě vstupních dat vygenerovat jednotlivé trajektorie. Jelikož u tohoto robota FANUC M-20iA (1.3) je fyzicky použit jako pracovní nástroj lihový FIX, jevila se nám jako nejrozmumnější možnost ta, která by realizovala nakreslení např. konkrétního obrázku, nápisu či loga. Tato varianta vyžaduje ke své demonstraci vysokou přesnost a funkčnost programu.

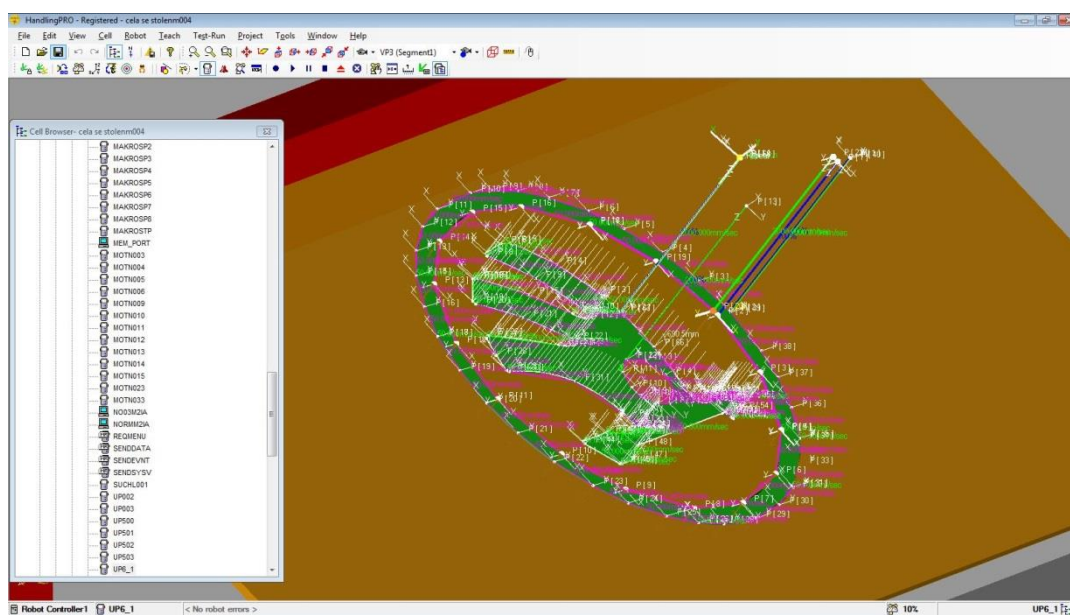
Jak již bylo řečeno, byly zde poměrně vysoké nároky na přesnost, tím pádem bylo i nutné následné korekce programu, jelikož software ROBOGUIDE neuvažuje při simulaci pozice jednotlivých přívodních kabelů, hadiček se vzduchem atd. Což by znamenalo za normálních okolností jejich poškození. Na základě tohoto faktu bylo třeba vytvořit „přiletové“ a „odletové“ body, ze kterých robot začíná práci na obrobku a ve kterých robot na obrobku končí, aniž by se zbytečně přetáčel.

Postup byl takový, že byl použit CAD software zvaný FREECAD, ve kterém bylo z poskytnuté 2D předlohy navrženo logo jako 3D objekt ve formátu IGES (Obr. 42:). Tento objekt byl naimportován do prostředí ROBOGUIDE (Obr. 43:) a následně umístěn na výrobní desku. Na základě možnosti generování trajektorie pomocí hran objektu bylo vytvořeno několik bodů, které se uložily jako nositelé souřadnic do programu. Na základě těchto jednotlivých bodů software vygeneroval po potvrzení program, který splňoval základní pohybové přejezdy mezi těmito body. Tento program bylo nutné modifikovat, aby nedošlo ke kolizním stavům.

Na základě přesného odladění a adjustace, bylo možné ověřit toto chování i na reálném robotu. Jediné co bylo zapotřebí, bylo přenesení, nahrání a spuštění vytvořeného programu, který se choval stejně jako v prostředí ROBOGUIDE.



Obr. 42: Návrh loga v softwaru FreeCad



Obr. 43: Naimportované ŠKODA logo v prostředí ROBOGUIDE



Tato práce má veliký přínos pro většinu větších firem, kde se výroba často různě modifikuje a mění. Díky OFF-LINE programování máme vše velice přehledně pod kontrolou a jsme schopni poměrně efektivně ladit technologii. Nejdůležitějším faktem zůstává, že jsme schopni velice pružně a rychle změnit výrobní trajektorii, což u ON-LINE programování znamená velice pracné vytvoření celého programu znovu, tím pádem velkou časovou náročnost na vytvoření, na základě velkého množství nepřesností, kterou s sebou ON-LINE programování nese. U OFF-LINE programování pouze nahrajeme do vytvořeného modelu pracovní cely v ROBOGUIDE příslušný CAD soubor obrobku ve formátu IGES a vygenerujeme základní trajektorie.

## 5 Závěr

Díky poskytnuté podpoře firmy ŠKODA AUTO ve formě skutečného robota firmy FANUC a vývojového prostředí ROBOGUIDE téže firmy, jsem měl možnost osvojit si základy programování a obsluhy průmyslového robota výše zmíněného výrobce.

Díky tomu se podařilo digitalizovat robotickou laboratoř s reálnými roboty v profesionálním prostředí ROBOGUIDE, jehož obsluhu jsem si v rámci této diplomové práce též osvojil.

Výsledky této digitalizace budou použity nejen pro tuto diplomovou práci, ale i pro výuku a prezentaci robotických laboratoří zde ve firmě ŠKODA AUTO. Plánuje se též využití těchto výsledků pro budoucí projekty v rámci těchto laboratoří.

Při realizaci bylo třeba zohlednit několik zásadních faktů, které se na první pohled zdály být triviální. V první řadě se jednalo o formát importovaného CAD výkresu do prostředí ROBOGUIDE. Po konečných úpravách CAD výkresu se jevil jako jediný kompatibilní formát IGES. Ostatní formáty bylo sice možné naimportovat, ovšem nebylo možné s těmito formáty plnohodnotně pracovat.

Dalším úskalím v cestě bylo určení směru nástroje. Škodoväcký STANDARD MQB uvažuje jiný defaultní směr TCP než prostředí ROBOGUIDE. Na základě správného ověření bylo nutné zajistit správný směr nástroje, aby adjustáž byla provedena bez jakýchkoliv komplikací.

Při adjustaci modelu bylo nutné dbát zvýšené ostražitosti při ladění programu na reálném robotu. Prostředí ROBOGUIDE v simulaci nezohledňuje hraniční pozice přívodních kabelů, tlakových hadiček se vzduchem atd., což by při zanedbání za normálních okolností vedlo k jejich zničení. Robot je v základním režimu vyzván „pouze“ k tomu, aby vykonal vytvořený program, ovšem některé přechody mezi jednotlivými pozicemi automaticky přepočítává, tím pádem se zde objevuje nežádoucí přetáčení těchto jednotlivých os.

Řešení této situace bylo takové, že se „kreslený“ objekt rozdělil do více dílčích částí, které jsou volány pomocí jednotlivých podprogramů. U každého podprogramu bylo třeba nadefinovat základní výchozí bod, aby nedocházelo k automatickému přepočítávání jednotlivých souřadnic.

Zadání této práce se podařilo naplnit a rozhodně skýtá potenciál pro další rozvoj a studium.

## 6 Literatura

- [1] *Oficiální stránky výrobce FANUC*[ON-LINE].[cit. 2015-01-28]. Dostupné z: [http://www.FANUCrobotics.cz/cs/products/a\\_industrial-robots/M-20iA/](http://www.FANUCrobotics.cz/cs/products/a_industrial-robots/M-20iA/)
- [2] *Oficiální stránky výrobce FANUC*[ON-LINE].[cit. 2015-01-28]. Dostupné z: <http://www.FANUC.com/>
- [3] M. Baumruk, *Process Designer Basic / Dikab*, 2014. 296 s.
- [4] M. Baumruk, *Process Designer Logistics / Advanced*, 2014. 177 s.
- [5] *MAINTENANCE MANUAL, FANUC - M-20iA*, 2010. 131 s.
- [6] V. Mostýn, *Teorie průmyslových robotů*, 2000. 147 s.
- [7] *FANUC - ROBOGUIDE*, 2008. 48 s.
- [8] *FANUC - Obsluha a programování*, 2008. 1151 s.
- [9] *Oficiální web výrobce robotů KUKA*[ON-LINE].[cit. 2015-02-10]. Dostupné z: [http://www.kuka-robotics.com/de/products/software/kuka\\_sim/kuka\\_sim\\_detail/PS\\_KUKA\\_Sim\\_Pro.htm/](http://www.kuka-robotics.com/de/products/software/kuka_sim/kuka_sim_detail/PS_KUKA_Sim_Pro.htm/)
- [10] *Oficiální web výrobce robotů FANUC*[ON-LINE].[cit. 2015-02-11]. Dostupné z: <http://www.FANUCrobotics.cz/cs/products/software/simulation%20and%20development/ROBOGUIDE/>
- [11] *Oficiální web výrobce robotů ABB*[ON-LINE].[cit. 2015-02-15]. Dostupné z: <http://new.abb.com/products/robotics/robotstudio/downloads/>
- [12] *Wikipedie - využití robotů v praxi*[ON-LINE].[cit. 2015-02-15]. Dostupné z: [www.wikipedia.com](http://www.wikipedia.com)
- [13] Heřman, Špicar, *Řídicí systém FANUC - R-30iA (MQB) - ELEKTROÚDRŽBA*, 2009, 49 s.
- [14] *Internetový server IFR*[ON-LINE].[cit. 2015-02-17]. Dostupné z: [www.ifr.org](http://www.ifr.org)
- [15] *FANUC - Bezpečnostní příručka*, 2009. 36 s.
- [16] Černohorský, *učební text k předmětu základy robotiky*, 2010, 45 s.
- [17] SCHMID, D. a kol. *Řízení a regulace pro strojírenství a mechatroniku. 1. vyd.* Praha, 1998. 420 s. ISBN 80-86706-10-9.
- [18] PIRES, J. N. *Industrial Robots Programming: Building Applications for the Factories of the Future*. Springer, 2007. 282 s. ISBN 978-0-387-23325-3.
- [19] Kelemen, M.: *Komponenty a moduly mechatronických systémů*, Košice, 2004. 119 s. ISBN 80-8073-212-4.
- [20] V. Špicar, *Řídicí systém FANUC - R-30iA (MQB) - PROGRAMOVÁNÍ*, 2013. ZA - ŠKODA Akademie Mladá Boleslav. 65 s.

- [21] *CAD systémy*[ON-LINE].[cit. 2015-03-15]. Dostupné z:  
[http://dspace.k.utb.cz/bitstream/handle/10563/24788/ma%C5%88as\\_2013\\_bp.pdf?sequence=1](http://dspace.k.utb.cz/bitstream/handle/10563/24788/ma%C5%88as_2013_bp.pdf?sequence=1)
- [22] *Historie CAD systému* [ON-LINE].[cit. 2015-03-15]. Dostupné z:  
<http://www.FREECAD.sk/cz/CAD-history>
- [23] Jelínek, M.: *Digitalizace školícího robotického pracoviště Škoda Auto*, 2015, TUL Liberec, 32 s.
- [24] Bureš, P.: *Ekonomická rozvaha využití CNC strojů ve výrobě basketbalu*, 2011, Lukavec, 95 s.
- [25] MALÝ, M.: *3D VIZUALIZACE VYBRANÉHO ZASTAVĚNÉHO AREÁLU*, 2009, Brno, 65 s.

## Příloha A: Podporované typy souborů v prostředí ROBOGUIDE

Tab.4: Podporované formáty pro prostředí ROBOGUIDE

KL	(all KAREL source)
CF	(all command files)
TX	(all text files)
LS	(all KAREL listings)
DT	(all KAREL data files)
PC	(all KAREL p-code)
TP	(all TP programs)
MN	(all MN programs)
VR	(all variable files)
SV	(all system files)
IO	(I/O config data)
DF	(all DEFAULT files)
ML	(all part model files)
BMP	(all bit-map images)
PMC	(all PMC files)
VA	(all Variable Listings)
DG	(all Diagnostic files)
VD	(all Vision VD files)
IBG	(all IBG files)
IBA	(all IBA files)
IMG	(all IMG files)
HTM	(all HTM files)
STM	(all STM files)
GIF	(all GIF files)
JPG	(all JPG files)
XML	(all XML files)

## Příloha B: Ovládání prostředí ROBOGUIDE

Tab.5: Ovládací zkratky prostředí ROBOGUIDE

Rotate view	Rotace	Pravé tlačítko myši
Pan view	Posun	[CTRL] + pravé tlačítko myši
Zoom in/out	Přiblížení/oddálení	Kolečko myši
Select object	Označení objektu	Levé tlačítko myši
Object property page	Vlastnosti objektu	2x tlačítko levé myši
Move object, one axis	Posun objektu v jednotlivých osách	Označení levým tlačítkem myši jednotlivou osu v osovém kříži
Move object, multiple	Pohyb objektu	[CTRL] + levé tlačítko myši
Rotate object	Rotace objektu	[SHIFT] + levé tlačítko myši
Move robot to surface	Pohyb robotem po povrchu	[CTRL] + [SHIFT] + levé tlačítko myši
Move robot to edge	Pohyb robotem po hraně	[CTRL] + [ALT] + levé tlačítko myši
Move robot to vertex	Pohyb robotem do rohu	[CTRL] + [ALT] + [SHIFT] + levé tlačítko myši
Move robot to center	Pohyb robotem do středu hrany	[SHIFT] + [SHIFT] + levé tlačítko myši

## Příloha C: Hlavní program FOLGE099

```
/PROG FOLGE099
/ATTR
OWNER          = MNEDITOR;
COMMENT        = "";
PROG_SIZE      = 1272;
CREATE         = DATE 15-04-16 TIME 13:43:36;
MODIFIED       = DATE 15-04-24 TIME 11:08:26;
FILE_NAME      = ;
VERSION        = 0;
LINE_COUNT     = 18;
MEMORY_SIZE    = 2344;
PROTECT        = READ_WRITE;
TCD: STACK_SIZE = 0,
    TASK_PRIORITY = 50,
    TIME_SLICE = 0,
    BUSY_LAMP_OFF = 0,
    ABORT_REQUEST = 0,
    PAUSE_REQUEST = 0;
DEFAULT_GROUP  = 1,*,*,*,*;
CONTROL_CODE   = 00000000 00000000;
/MN
1: UTOOL_NUM=1 ;
2: UFRAME_NUM=0 ;
3: PAYLOAD[10] ;
4: J P[1:HOME_POS] 100% CNT0 ACC100 TB 0.00sec,P-SPS ;
5: CALL UP6_1 ;
6: CALL UP6_2 ;
7: CALL UP6_3 ;
8: CALL UP6_4 ;
9: ;
10: UTOOL_NUM=1 ;
11: UFRAME_NUM=0 ;
12: PAYLOAD[10] ;
13: J P[2:HOME_POS] 100% CNT0 ACC100 TB 0.00sec,P-SPS ;
14: ;
15: ;
16: ;
17: ;
18: ;
/POS
P[1:"HOME_POS"]{
    GP1:
        UF : 0, UT : 1,
```

```
        J1=  -.000 deg,      J2= -45.000 deg,      J3=  0.000 deg,
        J4=  -.000 deg,      J5= -90.000 deg,      J6=  .000 deg
};
P[2:"HOME_POS"]{
  GP1:
    UF : 0, UT : 1,
    J1=  -.000 deg,      J2= -45.000 deg,      J3=  0.000 deg,
    J4=  -.000 deg,      J5= -90.000 deg,      J6=  .000 deg
};
/END
```



## Příloha D: Podprogram UP6\_3 „ŠKODA LOGO“

```
/PROG UP6_4
/ATTR
OWNER          = MNEDITOR;
COMMENT        = "SimPRO Auto-Gen";
PROG_SIZE      = 4408;
CREATE         = DATE 15-04-16 TIME 13:33:58;
MODIFIED       = DATE 15-04-16 TIME 13:34:00;
FILE_NAME      = ;
VERSION        = 0;
LINE_COUNT     = 59;
MEMORY_SIZE    = 4668;
PROTECT        = READ_WRITE;
TCD: STACK_SIZE = 0,
    TASK_PRIORITY = 50,
    TIME_SLICE = 0,
    BUSY_LAMP_OFF = 0,
    ABORT_REQUEST = 0,
    PAUSE_REQUEST = 0;
DEFAULT_GROUP  = 1,*,*,*,*;
CONTROL_CODE   = 00000000 00000000;
/MN
1: !SimPRO Auto-Generated TPP ;
2: !Part6, Feature4 ;
3: ;
4: UFRAME_NUM[GP1]=0 ;
5: UTOOL_NUM[GP1]=5 ;
6: !Approach Position ;
7: J P[1] 100% FINE ;
8: ;
9: !Segment1 ;
10: J P[2] 100% FINE ;
11: L P[3] 50mm/sec CNT100 ;
12: C P[4]
    : P[5] 50mm/sec CNT100 ;
13: L P[6] 50mm/sec CNT100 ;
14: L P[7] 50mm/sec CNT100 ;
15: L P[8] 50mm/sec CNT100 ;
16: C P[9]
    : P[10] 50mm/sec CNT100 ;
17: L P[11] 50mm/sec CNT100 ;
18: C P[12]
    : P[13] 50mm/sec CNT100 ;
19: L P[14] 50mm/sec CNT100 ;
20: C P[15]
```

: P[16] 50mm/sec CNT100 ;  
21:L P[17] 50mm/sec CNT100 ;  
22:L P[18] 50mm/sec CNT100 ;  
23:L P[19] 50mm/sec CNT100 ;  
24:L P[20] 50mm/sec CNT100 ;  
25:C P[21]  
: P[22] 50mm/sec CNT100 ;  
26:L P[23] 50mm/sec CNT100 ;  
27:C P[24]  
: P[25] 50mm/sec CNT100 ;  
28:L P[26] 50mm/sec CNT100 ;  
29:L P[27] 50mm/sec CNT100 ;  
30:C P[28]  
: P[29] 50mm/sec CNT100 ;  
31:L P[30] 50mm/sec CNT100 ;  
32:C P[31]  
: P[32] 50mm/sec CNT100 ;  
33:C P[33]  
: P[34] 50mm/sec CNT100 ;  
34:C P[35]  
: P[36] 50mm/sec CNT100 ;  
35:L P[37] 50mm/sec CNT100 ;  
36:C P[38]  
: P[39] 50mm/sec CNT100 ;  
37:L P[40] 50mm/sec CNT100 ;  
38:L P[41] 50mm/sec CNT100 ;  
39:L P[42] 50mm/sec CNT100 ;  
40:L P[43] 50mm/sec CNT100 ;  
41:L P[44] 50mm/sec CNT100 ;  
42:L P[45] 50mm/sec CNT100 ;  
43:L P[46] 50mm/sec CNT100 ;  
44:L P[47] 50mm/sec CNT100 ;  
45:L P[48] 50mm/sec CNT100 ;  
46:L P[49] 50mm/sec CNT100 ;  
47:C P[50]  
: P[51] 50mm/sec CNT100 ;  
48:L P[52] 50mm/sec CNT100 ;  
49:L P[53] 50mm/sec CNT100 ;  
50:C P[54]  
: P[55] 50mm/sec CNT100 ;  
51:L P[56] 50mm/sec CNT100 ;  
52:C P[57]  
: P[58] 50mm/sec CNT100 ;  
53:L P[59] 50mm/sec CNT100 ;  
54:C P[60]

```

: P[61] 50mm/sec CNT100 ;
55:C P[62]
: P[63] 50mm/sec CNT100 ;
56:C P[64]
: P[65] 50mm/sec CNT100 ;
57:C P[66]
: P[67] 50mm/sec FINE ;
58: !Retreat Position ;
59:L P[68] 2000mm/sec FINE ;
/POS
P[1]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 430.131 mm,        Y = -1265.767 mm,    Z = 561.306 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[2]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 446.016 mm,        Y = -1326.099 mm,    Z = 483.155 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[3]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 450.570 mm,        Y = -1334.253 mm,    Z = 490.375 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[4]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 464.224 mm,        Y = -1347.706 mm,    Z = 503.536 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[5]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 482.881 mm,        Y = -1357.267 mm,    Z = 514.709 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[6]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 489.519 mm,        Y = -1356.814 mm,    Z = 515.708 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};

```

```

};
P[7]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 501.437 mm,        Y = -1352.444 mm,    Z = 514.757 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[8]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 503.317 mm,        Y = -1350.489 mm,    Z = 513.626 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[9]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 483.479 mm,        Y = -1340.482 mm,    Z = 501.869 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[10]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 467.784 mm,        Y = -1325.740 mm,    Z = 487.299 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[11]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 466.654 mm,        Y = -1322.509 mm,    Z = 484.576 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[12]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 471.800 mm,        Y = -1321.943 mm,    Z = 485.185 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[13]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 476.487 mm,        Y = -1323.643 mm,    Z = 487.450 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[14]{
  GP1:

```

```

UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
X = 477.175 mm,        Y = -1324.040 mm,    Z = 487.896 mm,
W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[15]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 495.454 mm,        Y = -1332.359 mm,    Z = 498.034 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[16]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 520.531 mm,        Y = -1339.141 mm,    Z = 508.367 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[17]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 522.018 mm,        Y = -1339.216 mm,    Z = 508.727 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[18]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 522.789 mm,        Y = -1338.787 mm,    Z = 508.553 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[19]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 532.876 mm,        Y = -1328.100 mm,    Z = 502.352 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[20]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 532.484 mm,        Y = -1326.743 mm,    Z = 501.225 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[21]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 509.732 mm,        Y = -1320.971 mm,    Z = 492.145 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};

```

```

};
P[22]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 488.439 mm,        Y = -1311.245 mm,    Z = 480.308 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[23]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 490.645 mm,        Y = -1308.376 mm,    Z = 478.542 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[24]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 495.757 mm,        Y = -1308.053 mm,    Z = 479.332 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[25]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 500.477 mm,        Y = -1308.361 mm,    Z = 480.529 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[26]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 539.286 mm,        Y = -1311.425 mm,    Z = 490.781 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[27]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 541.435 mm,        Y = -1310.065 mm,    Z = 490.168 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[28]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 543.149 mm,        Y = -1301.905 mm,    Z = 484.218 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[29]{
  GP1:

```

```

UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
X = 543.719 mm,         Y = -1294.881 mm,      Z = 478.910 mm,
W = 172.575 deg,       P = -37.989 deg,      R = -63.295 deg
};
P[30]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 542.700 mm,         Y = -1294.340 mm,      Z = 478.286 mm,
    W = 172.575 deg,       P = -37.989 deg,      R = -63.295 deg
};
P[31]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 506.704 mm,         Y = -1290.906 mm,      Z = 468.319 mm,
    W = 172.575 deg,       P = -37.989 deg,      R = -63.295 deg
};
P[32]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 489.291 mm,         Y = -1274.370 mm,      Z = 452.013 mm,
    W = 172.575 deg,       P = -37.989 deg,      R = -63.295 deg
};
P[33]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 489.003 mm,         Y = -1271.376 mm,      Z = 449.644 mm,
    W = 172.575 deg,       P = -37.989 deg,      R = -63.295 deg
};
P[34]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 489.483 mm,         Y = -1269.379 mm,      Z = 448.199 mm,
    W = 172.575 deg,       P = -37.989 deg,      R = -63.295 deg
};
P[35]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 490.203 mm,         Y = -1268.652 mm,      Z = 447.785 mm,
    W = 172.575 deg,       P = -37.989 deg,      R = -63.295 deg
};
P[36]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 491.951 mm,         Y = -1267.369 mm,      Z = 447.150 mm,
    W = 172.575 deg,       P = -37.989 deg,      R = -63.295 deg
};

```

```

};
P[37]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 491.998 mm,        Y = -1267.339 mm,    Z = 447.138 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[38]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 492.676 mm,        Y = -1266.976 mm,    Z = 446.995 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[39]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 493.614 mm,        Y = -1266.585 mm,    Z = 446.884 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[40]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 495.735 mm,        Y = -1266.266 mm,    Z = 447.069 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[41]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 509.699 mm,        Y = -1266.332 mm,    Z = 449.957 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[42]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 511.132 mm,        Y = -1266.191 mm,    Z = 450.140 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[43]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 528.622 mm,        Y = -1261.668 mm,    Z = 450.203 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[44]{
  GP1:

```



```

UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
X = 528.759 mm,        Y = -1260.483 mm,    Z = 449.317 mm,
W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[45]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 516.171 mm,        Y = -1251.444 mm,    Z = 439.779 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[46]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 514.684 mm,        Y = -1251.369 mm,    Z = 439.419 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[47]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 505.705 mm,        Y = -1254.152 mm,    Z = 439.741 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[48]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 497.085 mm,        Y = -1260.633 mm,    Z = 442.992 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[49]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 428.915 mm,        Y = -1278.263 mm,    Z = 442.746 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[50]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 427.054 mm,        Y = -1278.086 mm,    Z = 442.231 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[51]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 425.280 mm,        Y = -1276.675 mm,    Z = 440.781 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};

```

```

};
P[52]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 425.142 mm,        Y = -1276.136 mm,    Z = 440.337 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[53]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 424.481 mm,        Y = -1275.848 mm,    Z = 439.979 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[54]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 417.019 mm,        Y = -1280.401 mm,    Z = 441.977 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[55]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 411.102 mm,        Y = -1285.538 mm,    Z = 444.741 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[56]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 411.150 mm,        Y = -1286.303 mm,    Z = 445.344 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[57]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 418.367 mm,        Y = -1287.766 mm,    Z = 447.941 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[58]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 427.594 mm,        Y = -1288.025 mm,    Z = 450.016 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[59]{
  GP1:

```

```

UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
X = 427.952 mm,        Y = -1287.128 mm,    Z = 449.397 mm,
W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[60]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 428.400 mm,        Y = -1285.591 mm,    Z = 448.301 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[61]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 430.375 mm,        Y = -1283.974 mm,    Z = 447.454 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[62]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 436.976 mm,        Y = -1283.076 mm,    Z = 448.103 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[63]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 438.776 mm,        Y = -1283.525 mm,    Z = 448.815 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[64]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 440.003 mm,        Y = -1285.280 mm,    Z = 450.418 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[65]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 440.457 mm,        Y = -1287.228 mm,    Z = 452.014 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[66]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 440.997 mm,        Y = -1309.385 mm,    Z = 469.230 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};

```

```
};
P[67]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 445.852 mm,        Y = -1325.747 mm,    Z = 482.847 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
P[68]{
  GP1:
    UF : 0, UT : 5,          CONFIG : 'N U T, 0, 0, 0',
    X = 429.968 mm,        Y = -1265.414 mm,    Z = 560.999 mm,
    W = 172.575 deg,      P = -37.989 deg,    R = -63.295 deg
};
/END
```