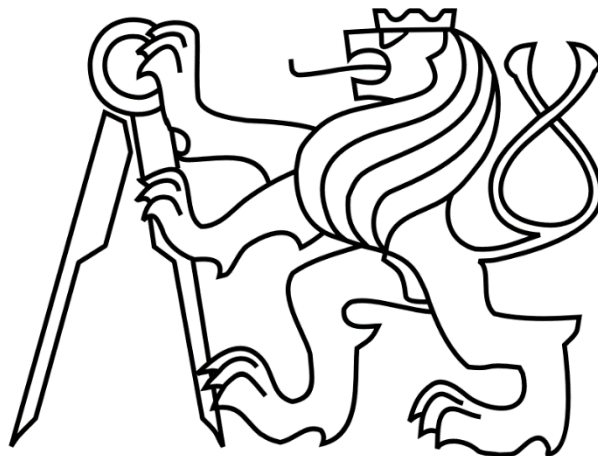


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STROJNÍ

Ústav výrobních strojů a zařízení



# Bakalářská práce

**Simulační model robota KUKA KR60 HA s výměnnými pracovními hlavicemi**

2022

**Strahinja Protić**

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Protič** Jméno: **Strahinja** Osobní číslo: **484725**  
Fakulta/ústav: **Fakulta strojní**  
Zadávající katedra/ústav: **Ústav výrobních strojů a zařízení**  
Studijní program: **Teoretický základ strojního inženýrství**  
Studijní obor: **bez oboru**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Simulační model robota KUKA KR60 HA s výměnnými pracovními hlavicemi**

Název bakalářské práce anglicky:

**Simulation model of a robot KUKA KR60 HA with interchangeable working heads**

Pokyny pro vypracování:

Popis tématu: Student bude pracovat na vytvoření simulačního modelu a emulátoru řídicího systému pro robot KUKA KR60 HA s výměnnými pracovními hlavicemi (end-effektory). Řešení bude provedeno pro Siemens NX CAM; Osnova práce: Programování průmyslových robotů. Rešerše možností realizace simulačních modelů strojů a robotů a databáze pomůcek v Siemens NX CAM. Rešerše tvorby emulátoru řídicího systému pro Siemens NX CAM. Rešerše funkcí ř.s. Sinumerik 840D pro řízení robotů. Úprava simulačního modelu robota a výměnných pracovních hlavic (end-efektorů) dle disponibilního vybavení robota KUKA KR60 HA v buňce na pracovišti CIIRC. Úprava emulátoru řídicího systému pro CAM systém Siemens NX pro realizaci výměny pracovních hlavic při odbavení NC programů. Příprava drah v NX CAM pro simulaci obrábění s využitím modelu robota a výměny pracovních hlavic. Testování vytvořeného simulačního modelu s emulátorem řídicího systému. Rozsah textové části: 40 - 60 stran; Rozsah grafické části: vybrané vývojové diagramy.

Seznam doporučené literatury:

[1] KRATĚNA, T.: Postprocessor a simulační model pro robota s přidavnými osami. Diplomová práce na ČVUT v Praze, FS, Ú12135, 2019.; [2] [15] SINUMERIK Run MyRobot /Machining V2.1 for KUKA robots. 2019. Dostupné z: <https://support.industry.siemens.com/cs/document/109758486/sinumerik-run-myrobot-direct-control-plc?dti=0&lc=en-WW>; [3] SIEMENS PLM SOFTWARE. NX CAM Robotics Programming [online]. 2016. Dostupné z: <https://www.geoplmm.com/knowledge-base-resources/GEOPLM-Siemens-PLM-NX-CAM-Robotics-Programming.pdf>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Petr Vavruška, Ph.D. ústav výrobních strojů a zařízení FS**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

**Ing. Tomáš Kratěna ústav výrobních strojů a zařízení FS**

Datum zadání bakalářské práce: **29.03.2022**

Termín odevzdání bakalářské práce: **25.07.2022**

Platnost zadání bakalářské práce: **30.09.2022**

Ing. Petr Vavruška, Ph.D.  
podpis vedoucí(ho) práce

Ing. Matěj Sulitka, Ph.D.  
podpis vedoucí(ho) ústavu/katedry

prof. Ing. Michael Valášek, DrSc.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta

# Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a že jsem uvedl v příloženém seznamu veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací, vydaným ČVUT v Praze 1. 7. 2009.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 25.7.2022

.....

podpis

# **Poděkování**

Za konzultace, rady a korekce bych rád poděkoval vedoucímu práce Ing. Petru Vavruškovi, Ph.D. Dále pak konzultantovi Ing. Tomáši Kratěnovi za poskytnutí praktické pomoci při zpracování práce. Také bych rád poděkoval svojí rodině, přítelkyni Natálii a její rodině za podporu během celého studia.

# Anotace

Autor:	Strahinja Protić
Název BP:	Simulační model robota KUKA KR60 HA s výměnnými pracovními hlavicemi
Rozsah práce:	63 str., 35 obr., 6 tab.
Akad. rok vyhotovení:	2021/2022
Škola:	ČVUT v Praze – Fakulta strojní
Ústav:	Ú12135 – Ústav výrobních strojů a zařízení
Vedoucí BP:	Ing. Petr Vavruška, Ph. D
Konzultant:	Ing. Tomáš Kratěna
Zadavatel:	ČVUT FS, Ú12135
Využití:	Příprava a verifikace NC programů pro pracoviště s robotickou buňkou na aditivní výrobu
Klíčová slova:	Obráběcí robot, Simulace obrábění, Řídicí systém, Emulátor řídicího systému, Výměnné pracovní hlavice
Anotace:	Bakalářská práce se zabývá problematikou simulačních modelů s výměnnými pracovními hlavicemi. Na robotické buňce s výměnnými pracovními hlavicemi byl aplikován postup rozšíření simulačního modelu o emulátor řídicího systému. Byly demonstrovány potřebné úpravy pro konkrétní robot a byly vytvořeny podprogramy realizující výměnu pracovních hlavic. Výsledkem je funkční simulační model pro prostředí Siemens NX 12.

# Annotation

Author:	Strahinja Protić
Title of bachelor dissertation:	Simulation model for robot KUKA KR60 HA with interchangeable working heads
Extent:	63 p., 35 fig., 6 tab.
Academic year:	2021/2022
University:	CTU in Prague – Faculty of mechanical engineering
Department:	Ú12135 – Research Centre of Manufacturing Technology
Supervisor:	Ing. Petr Vavruška, Ph. D
Consultant:	Ing. Tomáš Kratěna
Submitter of theme:	CTU – Faculty of Mechanical Engineering
Application:	Preparation and verification of NC programs for a robotic cell for additive manufacturing
Key words:	Machining with industrial robot, Simulation of Machining, CNC system, CNC system emulator, Interchangeable working heads
Annotation:	This bachelor's thesis deals with simulation models of robots with interchangeable working heads. The methodical process of creating such model was explained in this thesis, and applied to a robotic cell with interchangeable working heads. All necessary changes to the emulator and its subprograms were made to realize the changing of working heads during the simulation. The result is a simulation model working in Siemens NX 12.

## Obsah

1.	Úvod a cíle práce .....	10
2.	Programování průmyslových robotů .....	11
2.1.	Rozdíly mezi průmyslovými roboty a CNC obráběcími stroji .....	11
2.1.1.	Konstrukce CNC obráběcích strojů .....	11
2.1.2.	Konstrukce průmyslových robotů.....	12
2.2.	Řízení a programování průmyslových robotů.....	13
2.2.1.	Siemens NX CAM .....	14
2.2.2.	Siemens NX CAM pro řízení robotů .....	15
2.2.3.	Úrovně simulace v Siemens NX.....	18
3.	Simulační modely .....	21
3.1.	Kinematické modely .....	21
3.1.1.	CAD modely .....	22
3.1.2.	Prvky Machine Component .....	22
3.2.	Řídicí systémy a emulátory řídicích systémů.....	25
3.2.1.	Řídicí systémy.....	25
3.2.2.	Siemens Sinumerik 840D sl.....	28
3.2.3.	Funkce řídicího systému Siemens Sinumerik 840D sl .....	29
3.2.4.	Sinumerik Run MyRobot.....	30
3.2.5.	Emulátory řídicích systémů .....	31
3.2.6.	Machine Configurator.....	32
3.2.7.	Podprogramy emulátoru řídicího systému .....	33
3.3.	Popis robotické buňky.....	35
3.3.1.	Specifické funkce generované postprocesorem .....	36
3.4.	Závěr řešerše .....	38
4.	Návrh vlastního řešení .....	39
5.	Aplikace metodiky na robotické buňce .....	40
5.1.	Úprava kinematického modelu .....	40
5.2.	Tvorba podprogramů emulátoru řídicího systému.....	41
5.2.1.	Přehled podprogramů emulátoru řídicího systému pro robotickou buňku41	

---

5.2.2.	Indexování pracovních hlavic a nástrojů .....	42
5.2.3.	Podprogramy pro výměnu pracovních hlavic a nástrojů .....	43
5.2.4.	Podprogramy pro navařovací procesy .....	46
5.3.	Úprava parseru .....	47
5.3.1.	Knihovny Lex a Tokens.....	48
5.3.2.	Knihovna Yacc .....	49
5.4.	Tvorba programů pro parser v prostředí Machine Configurator.....	50
5.4.1.	Funkce H52.....	51
5.4.2.	Funkce H10 a M89 .....	52
5.4.3.	Funkce M10 .....	53
5.5.	Implementace a použití souborů simulačního modelu.....	53
5.6.	Ověření funkčnosti simulačního modelu robotické buňky a porovnání s reálným strojem .....	55
5.6.1.	Ověření funkčnosti simulačního modelu .....	55
5.6.2.	Srovnání simulačního modelu s robotickou buňkou.....	56
6.	Závěr .....	58
	Seznamy.....	59
	Seznam literatury.....	59
	Seznam obrázků .....	61
	Seznam tabulek.....	62
	Seznam použitého software.....	62
	Seznam příloh.....	62



## Seznam zkratk

AC	Any-Controller / Typ programovacího jazyka
CAD	Computer Aided Design / Počítačem podporované konstruování
CAM	Computer Aided Manufacturing / Počítačem podporované konstruování
CCF	Controller Configuration File / Konfigurační soubor řídicího systému
CFC	Compact Flash Card / Kompaktní flash karta
CL	Cutter Location / Poloha nástroje
CNC	Computer Numerical Control / Počítačové numerické řízení
COM	Communication module / Komunikační modul
CSE simulace	Common Simulation Engine simulace / Simulace na základě NC kódu
DLL	Dynamic-link library / Dynamický propojovaná knihovna
HMI	Human-machine interface / Rozhraní člověk-stroj
IPW	In Process Workpiece / Dílec ve výrobě
MCF	Machine Configuration File / Strojní konfigurační soubory
NC	Numerical Control / Číslicové řízení
NCU	Numerical Control Unit / Číslicově řízená jednotka
PLC	Programmable Logic Controller / Programovatelný logický automat
TCU	Thin Controller Unit
TPS	Tool path based simulation / Simulace na základě nástrojových drah
USB	Universal Serial Bus / Univerzální sériová sběrnice
VNCK	Virtual Numerical Control Kernel
XML	Extensible Markup Language / Rozšířitelný značkovací jazyk

## 1. Úvod a cíle práce

V dnešní době se stále více uplatňuje při projektování obráběcích a aditivních operací NC (Numerical control) programování. Nejefektivnějším způsobem, jak toto provádět u složitých dílců, je s pomocí CAM (Computer aided machining) softwarů. Tyto softwary umožní více úrovní simulace, které slouží k ověření realizovatelnosti dané operace. Aby tato simulace byla plnohodnotná, je zapotřebí použitý stroj virtuálně reprezentovat simulačním modelem. Simulační model zahrnuje jak reprezentaci samotného stroje (pomocí kinematického modelu), tak i jeho řídicího systému (pomocí emulátoru řídicího systému). Emulátor řídicího systému je tedy zapotřebí pro odbavení generovaného NC kódu při simulaci v prostředí CAM, který je generován pro syntaxi řídicího systému Siemens Sinumerik 840D sl. Tento emulátor by proto měl věrohodně reprezentovat funkci reálného řídicího systému a funkce stroje, resp. robotické buňky. To zahrnuje odbavení všech potřebných funkcí obsažených v NC kódu. Může se jednat například o výměnu pracovních hlavic, která probíhá za pomoci řetězce podprogramů. Emulátor musí při výměně být schopen nejen pracovní hlavici uchopit, ale také najet do její pozice. Tímto bude zaručena funkční i vizuální věrohodnost výměny pracovních hlavic. Stroj, kterým se tato práce zabývá, je průmyslový robot od firmy KUKA, který je používán pro výzkumné účely. Výzkum probíhá v rámci robotické buňky, která je navržena pro robotické navařování. Samotná robotická buňka se nachází v laboratoři CIIRC u – Český institut informatiky, robotiky a kybernetiky (Czech Institute of Informatics, Robotics and Cybernetics). Robotická buňka je vyvíjena na míru, a tak pro ni není k dispozici komerčně dostupný simulační model. Je tedy zapotřebí takový model vytvořit a odladit tak, aby byl použitelný pro simulace výrobních operací. V rámci předchozího vývoje robotické buňky byl již vytvořen postprocesor pro syntaxi řídicího systému Siemens Sinumerik 840D sl a kinematický model robotické buňky.

Cílem této práce je vytvoření simulačního modelu robotu KUKA KR60 HA pro prostředí výzkumné robotické buňky v CIIRC u, rozšířeného o emulátor řídicího systému a jeho podprogramy. Také dojde k úpravě kinematického modelu. Konečný simulační model bude schopen odbavit NC program spolu s výměnou pracovních hlavic (end-effektorů) nacházejících se v buňce.

V rámci vlastního řešení vznikne následující:

- Emulátor řídicího systému Siemens Sinumerik 840D sl
- Podprogramy starající se o výměnu pracovních hlavic
- Simulace výrobní operace v prostředí Siemens NX

## 2. Programování průmyslových robotů

V dnešní době se začínají rozrůstat možnosti využívání průmyslových robotů. Toto zapříčinily dva trendy – jedním je postupné zvyšování celkové přesnosti, tuhosti a možnost manipulace vyšší zátěží, a druhým je nízký počet kvalifikovaných pracovníků. Zatímco v minulosti byly průmyslové roboty využívány zejména k operacím jako je sváření, manipulace nebo polohování dílců, tak se v dnešní době začínají využívat i pro operace obrábění, broušení a různé dokončovací operace, jako je např. odstraňování otřepů. Tento nárůst využívání robotů v průmyslu si vyžádal nejen vývoj možností jejich programování, ale i virtualizace a simulace. [2]

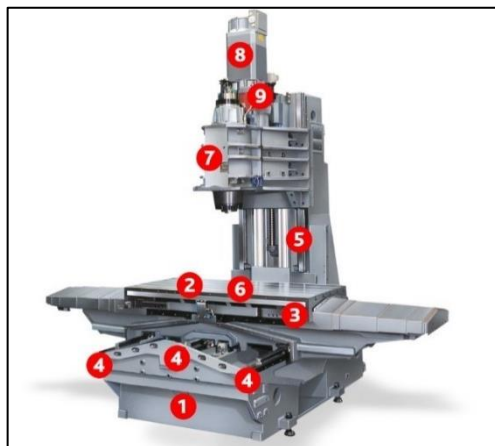
### 2.1. Rozdíly mezi průmyslovými roboty a CNC obráběcími stroji

V této kapitole jsou popsány rozdíly mezi CNC obráběcími stroji a průmyslovými roboty. Pozornost je věnována zejména konstrukčním rozdílům, ze kterých pak vyplýne potřeba dodatečných nástrojů pro programování a řízení robotů.

#### 2.1.1. Konstrukce CNC obráběcích strojů

CNC obráběcí stroje se dají kategorizovat podle mnoha kritérií. Podle počtu řízených os se dělí na tři největší skupiny. Těmi jsou tříosé obráběcí stroje, pětiosé obráběcí stroje a multifunkční obráběcí centra. Tříosé stroje mají tři řízené osy - X, Y a Z, tzv. lineární osy. Pětiosé stroje jsou doplněny o další dvě rotační osy. Ty jsou nejčastěji realizovány otočně-sklopným stolem. Multifunkční obráběcí centra jsou dále doplněna o další osy a kombinují více technologických procesů – většinou soustružení a frézování. CNC obráběcí stroje se vyznačují jejich vysokou tuhostí a přesností, které jsou dány konstrukčním uspořádáním jednotlivých komponent.

Na Obr. 1 je vyobrazeno vertikální obráběcí centrum MCV 1000 od firmy Kovosvit MAS, které je zde zastupitelem skupiny tříosých strojů. Části tohoto stroje dle obrázku jsou: 1 - Lože stroje, 2 - Upínací plocha stolu, 3 - Lineární valivé vedení – osa X, 4 - Lineární valivé vedení – osa Y, 5 - Lineární valivé vedení – osa Z, 6 - Stůl, 7 - Vřeteník, 8 – Motor, 9 – Dvoustupňová převodovka.



Obr. 1 Vertikální obráběcí centrum MCV 1000 [19]

### 2.1.2. Konstrukce průmyslových robotů

Průmyslové roboty jsou na rozdíl od CNC obráběcích strojů konstruovány pouze otočnými osami, nazývanými klouby (angl. joints). Tělo se typicky dělí na dvě části – rameno a zápěstí. V případě robotu KUKA KR60 (Obr. 2), jenž je předmětem této práce, je rameno tvořeno třemi klouby a zápěstí dalšími třemi klouby. Konstrukce robotů jsou oproti klasickým konstrukcím CNC obráběcích strojů méně tuhé a méně přesné. Tento typ konstrukce je i přes své nedostatky účelový, a umožňuje robotům vyšší flexibilitu výroby a širší spektrum uplatnění. Průmyslové roboty se také dají obdobně



Obr. 2 Robot KUKA KR60-3 [3]

jako CNC obráběcí stroje doplnit o dvouosý stůl, či umístit na lineární pojezd, který umožňuje pohyb základny robotu v prostoru. [4]

Vlivem odlišné konstrukce je u robotů zapotřebí dodatečných nástrojů pro jejich řízení. To by byly například přepočty prostorových bodů z X, Y a Z souřadnic na natočení v kloubových pohonech robotu, či třeba nastavení preferovaných konfigurací robotu. Tato dodatečná robotická nastavení jsou blíže popsána později v této práci.

Další možností, jak využít průmyslové roboty, je jejich spolupráce s CNC obráběcími stroji, nebo jinými roboty, například jako obráběcí nebo manipulační zařízení. Na Obr. 3 je vyobrazena robotická výrobní linka sloužící k bodovému svařování karoserií automobilů, realizovaná pomocí robotů postavených sériově za sebou. Mezi největší výrobce robotů ve světě patří Fanuc, ABB a KUKA.



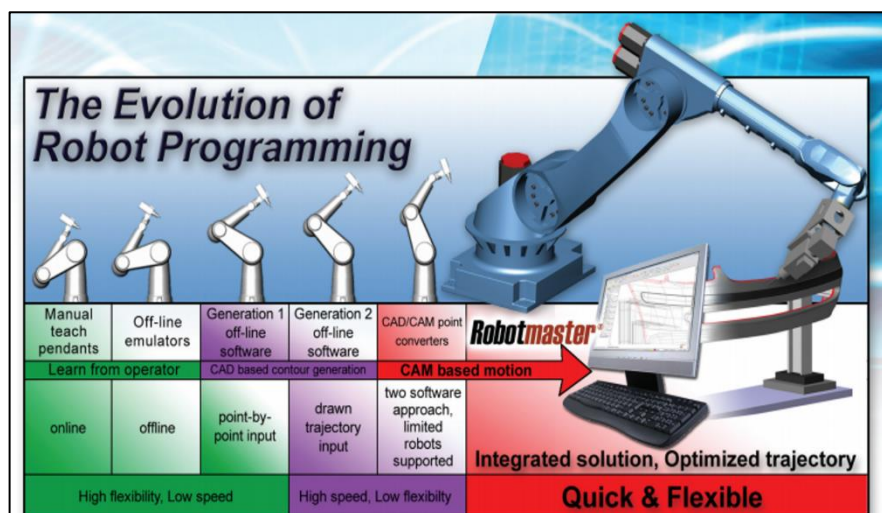
*Obr. 3 Robotická linka pro bodové svařování karoserií automobilů [2]*

## **2.2. Řízení a programování průmyslových robotů**

Průmyslové roboty se dají programovat třemi způsoby – ručně, dílensky a automaticky. V rámci této práce je podstatné pouze automatické programování probíhající pomocí počítačových softwarů CAM (Computer aided manufacturing). Tyto softwary umožňují obrábění tvarově složitých dílců. Mimo to poskytují možnost do výroby zapojit přídatné osy, otočně-sklopné stoly či použít více strojů najednou (např. robot + CNC obráběcí stroj). Technolog tvoří NC program na počítači

prostřednictvím vybraného CAM softwaru. Do programu se vloží geometrie dílce a zadají se technologické a výrobní parametry, program následně vygeneruje dráhy nástroje.

Vývoj programování robotů se odvíjel od jejich využití v průmyslu a od vývoje schopností řídicích systémů. V první generaci bylo možné roboty ovládat ručně, robot byl manuálně veden po dráze a potom tento pohyb opakoval. Toho se využívalo například při svařování nebo natírání. Další fází bylo ruční programování pomocí ovládacího panelu. Tento způsob byl doplněn o možnost programování drah pomocí počítačového softwaru, jedná se o takzvané offline programování. Toto programování nejdříve probíhalo zadáním drah bod po bodu, potom bylo možné dráhy generovat pomocí CAM softwarů. Postupem času byly CAM systémy vylepšeny a rozšířeny o prostředí pro programování robotů. Díky tomuto vznikla aktuálně nejvyšší úroveň programování robotů umožňující jejich plnohodnotné ovládání. Na Obr. 4 od firmy Hypertherm je vidět postupný vývoj programování robotů. [6]



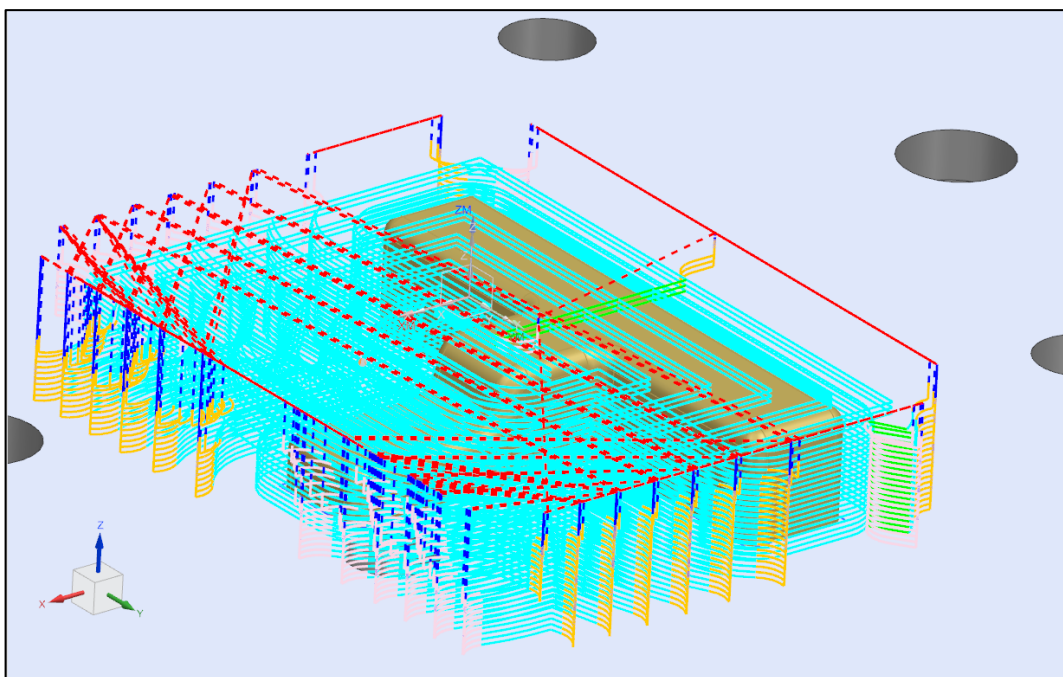
Obr. 4 Vývoj programování robotů [21]

### 2.2.1. Siemens NX CAM

Siemens NX CAM je CAM software od firmy Siemens. Používá se pro plánování výrobních procesů. Program má možnost generovat dráhy nástroje (Obr. 5), včetně technologických podmínek. Dále tu jsou například knihovny nástrojů. Uživatel si tyto nástroje může upravovat nebo si je sám tvořit. V programovacím prostředí je také možná simulace obrábění s využitím simulačního modelu reálného stroje a pracovní hlavice. Při ní jsou vidět reálné pohyby stroje a přídavných os. Také je možné nastavit vizualizaci úběru či přídávku materiálu. Pro simulaci jsou dostupné celkem čtyři úrovně – TPS



simulace (tool path based simulation), CSE simulace (common simulation engine simulace, někdy také nazývaná machine code base simulation), simulace pomocí externího kódu a VNCK simulace (Virtual Numerical Controller Kernel). Tyto úrovně jsou blíže popsány později v textu. [1]

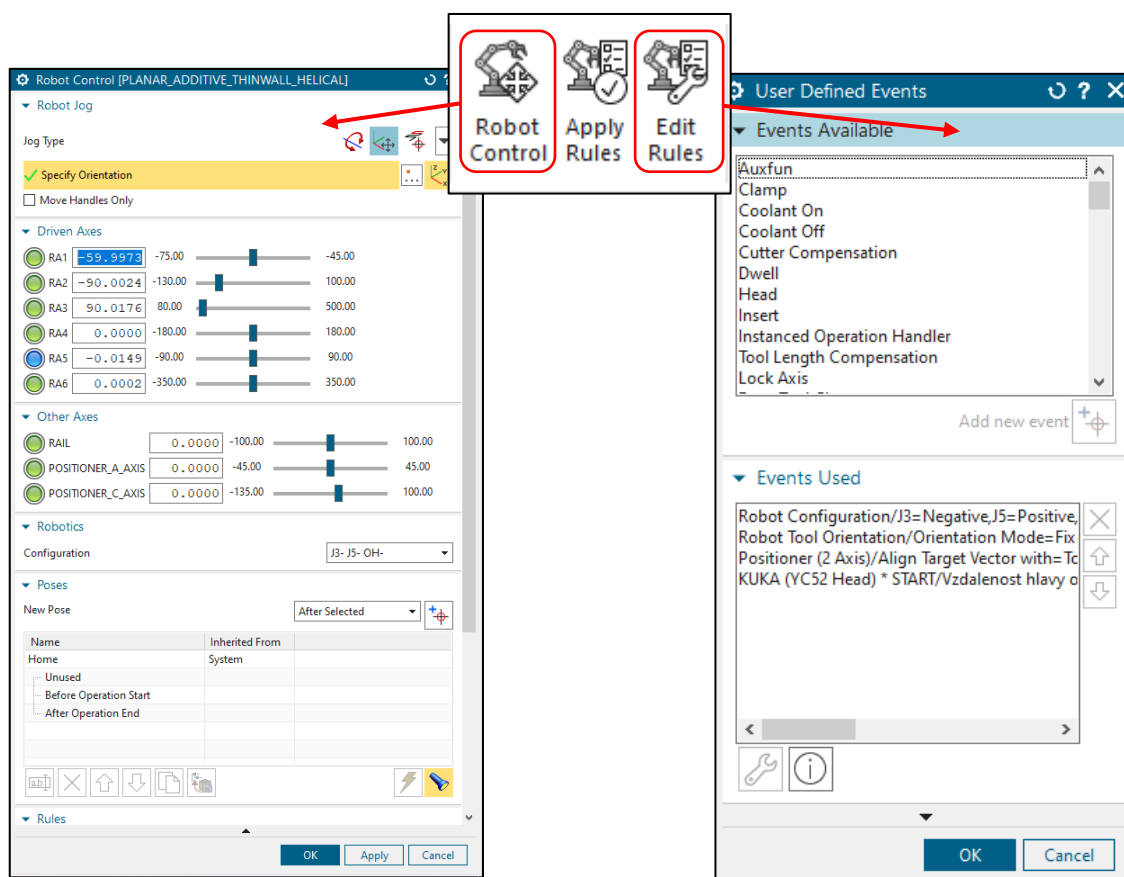


*Obr. 5 Vizualizace drah obrábění v prostředí Siemens NX CAM*

### **2.2.2. Siemens NX CAM pro řízení robotů**

Software Siemens NX CAM umožňuje využít pro programování robotů prostředky ve stejném rozsahu jako pro programování CNC obráběcích strojů. To znamená, že je schopen generovat dráhy nástrojů, včetně přepočtu souřadnic X, Y a Z na pohyby robotu. Software je pro programování robotů vybaven mnoha pravidly. Tato pravidla jsou specifická robotům a pro reálné programování a simulaci výroby je zapotřebí je specifikovat. Pro plnohodnotné programování a virtualizaci robotů jsou také potřeba stejné komponenty jako pro programování CNC obráběcích strojů – kinematický model, postprocesor a emulátor řídicího systému. Tyto komponenty jsou více přiblíženy v kapitole č. 3 Simulační modely.

Prvním pravidlem pro programování robotů je možnost definovat polohy robotu před a po obrábění, nebo mezi jednotlivými operacemi. Toto se provádí v dialogovém okně Robot Control (Robotické ovládání) nebo Edit Rules (Upravit pravidla). Zde je možnost nastavit libovolný počet pozicí robotu, což pomáhá při výrobě – např. najetím do volného prostoru pro ruční výměnu nástrojů nebo obrobků. Okna Robot Control a Edit Rules jsou propojena. Robot Control má lepší grafické prostředí (např. při nastavování pozicí jsou vypsané do přehledného seznamu, dále taky ukazuje aktuální hodnoty natočení v kloubech), zatímco Edit Rules nabízí větší výběr parametrů, které lze nastavit – např. zapínání chlazení, pouštění laseru apod. Na Obr. 6 jsou obě okna znázorněna. Pravidla se nastavují pro každou operaci zvlášť. [6]

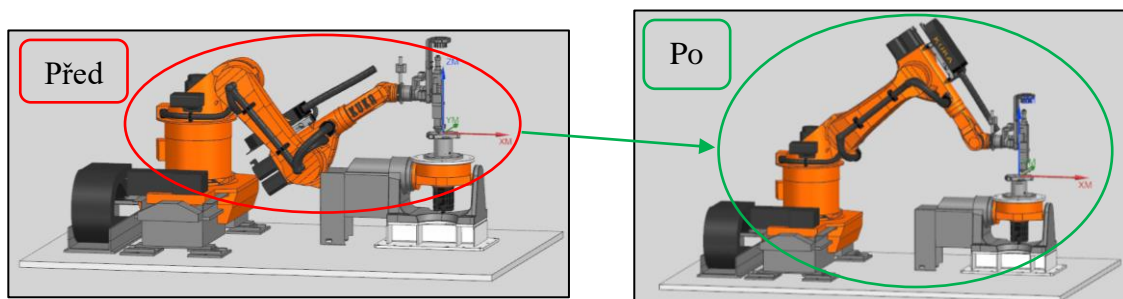


Obr. 6 Dialogová okna Robot Control (vlevo) a Edit Rules (vpravo)

Dalším pravidlem je nastavení konfigurace robotu při operacích (viz Obr. 8). Kvůli své konstrukci může robot dosahovat jedné polohy více kloubovými konfiguracemi. Některé z těchto poloh mohou být fyzicky nedosažitelné. Nastavením konfigurace robotu je možné zvolit žádanou orientaci částí robotické ruky pro danou operaci. Na Obr. 8 jsou

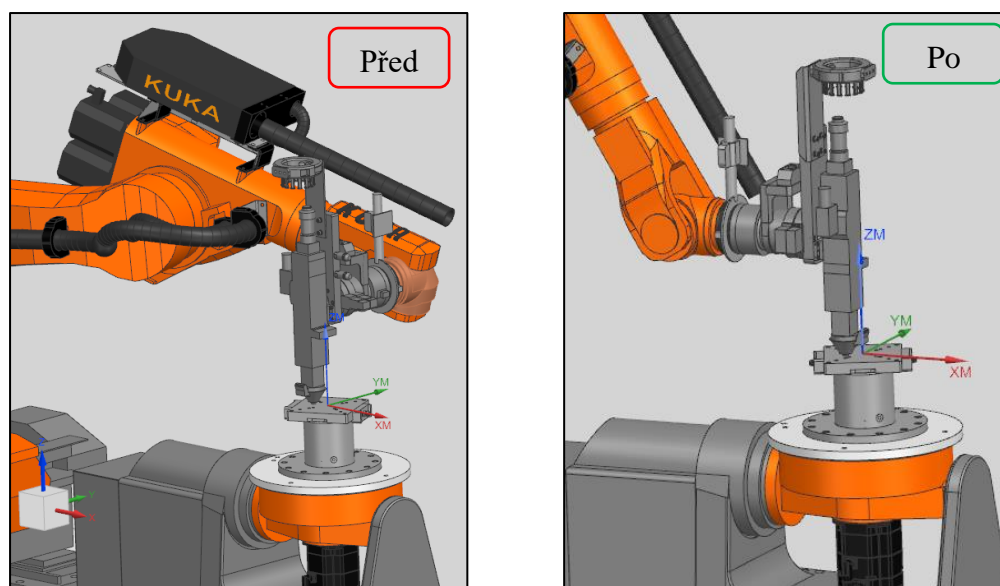


zobrazeny dvě konfigurace robotu – vpravo dosažitelná a vlevo nedosažitelná. To, zda je pozice reálně dosažitelná, lze ověřit při simulaci s kinematickým modelem, kdy Siemens NX ohlašuje přesažení limitu v jednotlivých osách. [6]



*Obr. 8 Aplikace pravidla pro nastavení konfigurace robotu*

Třetím pravidlem je nastavení orientace nástroje. Na roboty nelze narozdíl od CNC obráběcích strojů upnout nástroj pouze pomocí nástrojové hlavičky. Ty mohou mít motorizované prvky (např. vřeteno). Konec nástroje či efektoru (nástrojové hlavičky) může dosahovat nekonečně mnoho poloh pro každý bod dráhy operace. Nastavením orientace nástroje či nástrojové hlavičky lze danému problému zamezit. Na Obr. 7 je vidět nástrojová hlavička orientovaná tak, aby robot byl ve správné konfiguraci. [6]



*Obr. 7 Aplikace pravidla pro nastavení orientace nástrojové hlavičky robotu*

Posledním pravidlem, jenž lze pro roboty nastavit, je zapojení externích os. Například lze do výroby zapojit otočně-sklopný stůl nebo lineární pojezd. Otočně-sklopný stůl může

být nastaven tak, aby otáčel pouze jednou osou, či aby sledoval vygenerovanou dráhu. Možností je také uzamknout osy stolu na určitou hodnotu. [6]

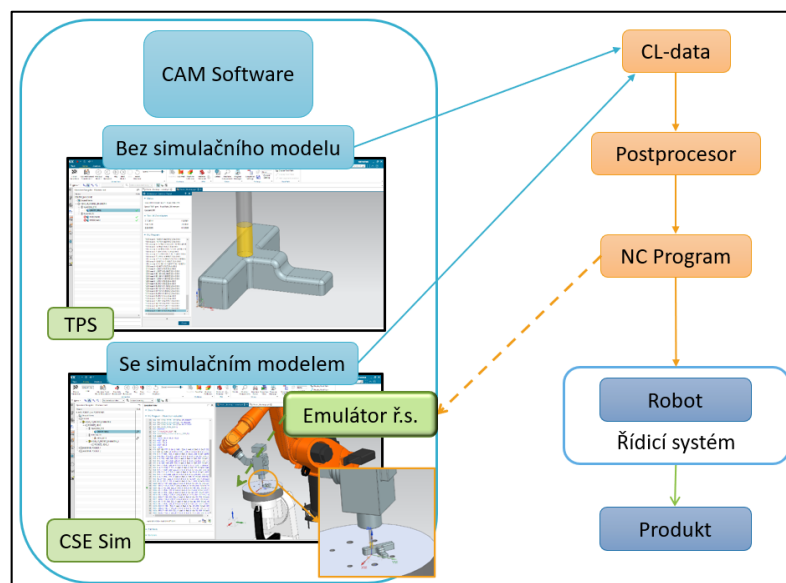
Po nastavení potřebných pravidel je nutné tato pravidla na danou operaci aplikovat. To se projeví v konečném NC programu. V prostředí Siemens NX CAM je vedle dané operace vidět změna ikonky potvrzující nastavení robotických pravidel zvolené operace, viz Obr. 9 (vlevo bez pravidla, vpravo s pravidlem). [6]



Obr. 9 Vizuální indikace uplatnění robotických pravidel

### 2.2.3. Úrovně simulace v Siemens NX

Jak již bylo řečeno, Siemens NX umožňuje celkem čtyři úrovně simulace, ze kterých má každá své uplatnění. Na Obr. 10 je vyobrazeno jednoduché schéma TPS (tool path based) a CSE (common simulation engine) simulací, které jsou v následujících kapitolách blíže popsány. Na vstupu jsou buď dráhy nástroje bez definovaného kinematického modelu, nebo dráhy s definovaným kinematickým modelem. Pomocí těchto údajů jsou generována CL-data (Cutter Location data), která se buď dají rovnou využít k TPS simulaci, nebo se posílají dále do postprocesoru, který je převádí na NC (Numerical control) kód odbavitelný pomocí emulátoru řídicího systému. Výstupem je pak CSE simulace.



Obr. 10 Schéma simulace

### **2.2.3.1. TPS simulace**

TPS neboli tool path based simulace je nejjednodušším typem simulace v prostředí Siemens NX. TPS simulací je možné vizualizovat buď jen pohyb nástroje, nebo i pohyb celého stroje. Pro simulaci pohybů stroje je potřeba do Siemens NX zavést kinematický model stroje, nebo využít jeden z již zavedených kinematických modelů z knihovny Siemens NX. TPS Simulace probíhá na základě interních CL dat v CAM systému a je schopna sledovat kolize nástroje a obrobku, popřípadě upínacích prvků. To ovšem není možné u samotného těla stroje, protože nejsou brány v potaz fyzikální rozměry stroje. Při tomto typu simulace nelze verifikovat NC kód a také není počítán strojní čas. Kvůli tomuto se TPS považuje za neplnohodnotný typ simulace. Simulace slouží především k ověření dráhy z hlediska kinematických rozsahů stroje – překročení limitů os. Také se dá využít pro kontrolu vhodného umístění obrobku nebo pro kontrolu obrobeného povrchu. Tato simulace neprobíhá pomocí emulátoru řídicího systému, a tak není předmětem této práce. [7]

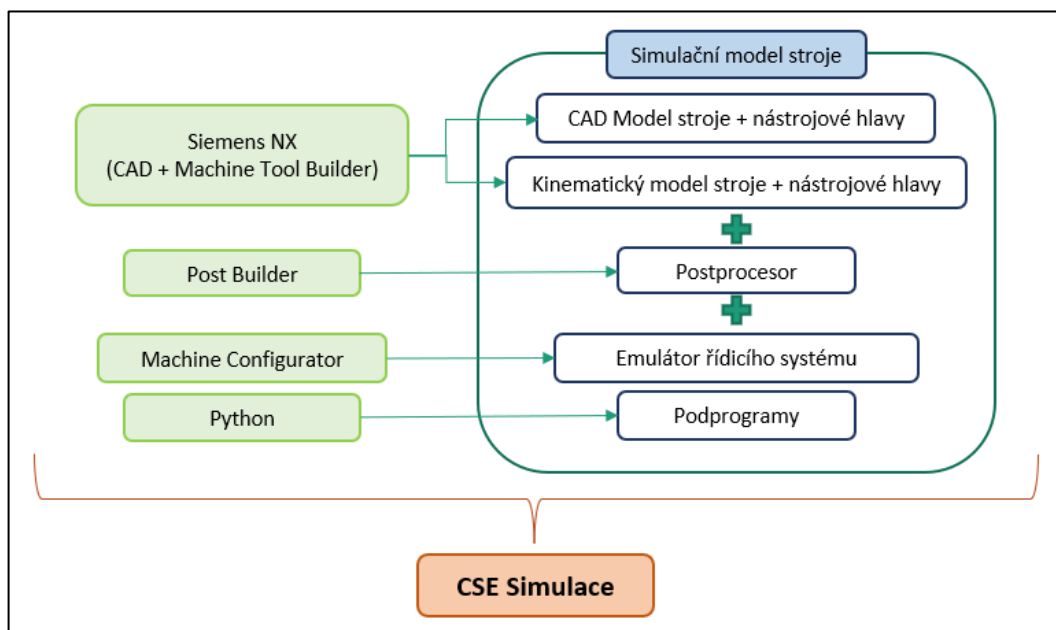
### **2.2.3.2. CSE Simulace**

CSE simulace je simulace řízená emulátorem řídicího systému. V prostředí Siemens NX CAM se tato simulace nazývá Machine code based simulation (simulace probíhající na základě NC kódu). Jedná se o technologicky pokročilejší simulaci, než je TPS. Pro její odbavení je potřebný emulátor řídicího systému a jeho podprogramy, postprocessor a kinematický model stroje. Tyto tři komponenty společně tvoří simulační model stroje. Samotné řízení je definováno CSE drivery, které obsahují informace o osách stroje a řídicím systému. Zde jsou zavedené příslušné kanály, přiřazeny osy a zavedeny funkce a cykly řídicího systému. Tyto drivery se pro prostředí Siemens NX CAM programují v softwaru Machine Configurator, který také vyvíjí firma Siemens. Drivery obsahují informace o rozsazích a rychlostech jednotlivých os. Program Machine Configurator je více přiblížen v kapitole č. 3.2.6 Machine Configurator.

Jak již bylo zmíněno, CSE simulace probíhá na základě vygenerovaného NC kódu, nikoliv interních CL-datech. Tento kód je generován postprocesorem a poté poslán do emulátoru řídicího systému. Emulátor NC kód překládá pomocí tzv. parseru. Přeložený NC kód se pak odbaví v simulačním prostředí Siemens NX. Parser je tvořen několika python soubory, jmenovitě se jedná o soubory yacc, lex a tokens. Tyto soubory budou

přibliženy ve vlastním řešení práce. Simulace potom vizualizuje reálnou dráhu stroje a počítá i velice přesný strojní čas. Pokud se při simulaci počítá s adekvátní časovou konstantou, tak je tento strojní čas identický s tím reálným. Právě díky zavedení této konstanty je Siemens NX schopen do strojního času započítat například čas potřebný pro upnutí nástroje (jedná se o dobu potřebnou k samotnému upnutí nástroje, čas pohybu stroje k zásobníku se bere potaz i bez použití časové konstanty).

Na Obr. 11 je zobrazeno zjednodušené schéma potřebných komponent pro CSE simulaci. V levé části obrázku jsou vidět softwary potřebné k realizaci těchto komponent. [7]



Obr. 11 Schéma komponentů potřebných k CSE simulaci

### 2.2.3.3. External code based simulace

External code based simulace probíhá na základě externího kódu. Ten může být napsán ručně nebo vygenerován jiným CAM systémem a jiným postprocesorem. Takto odbavovaný kód se dá ručně za běhu optimalizovat. Kód psaný ve správném jazyce řídicího systému je nejdříve načten z PC úložiště, následně celý odbaven pomocí software tlačítek, nebo pouštěn po řádcích. Další možností je vstupovat do funkcí volajících určité podprogramy. Toho lze využít např. při kontrole funkčnosti podprogramů emulátoru řídicího systému, které by se např. mohly starat o výměnu nástroje. Tento typ simulace je v rámci této práce využit pro ověření a odladění programovatelných kódů podprogramů emulátoru řídicího systému. V této simulaci lze obdobně jako u CSE Simulace verifikovat

NC kód a detekovat kolize v plném rozsahu. Simulace na základě externího kódu a CSE simulace tedy probíhají na stejné úrovni.

#### **2.2.3.4. VNCK simulace (Virtual numerical controller kernel)**

VNCK simulace je nejvyšší dostupnou úrovní simulace v Siemens NX CAM a probíhá za využití veškerých funkcí řídicího systému Siemens Sinumerik 840D, což ji dělá velice náročnou. Obsahuje interpolátor tohoto řídicího systému, pomocí kterého odbavuje NC kód, stejně jako reálný stroj. VNCK simulace nijak nevyužívá emulátor řídicího systému a tudíž není předmětem této práce. [7]

Pro všechny čtyři typy simulace je Siemens NX také schopen ukazovat takzvaný IPW (In Process Workpiece) model, viz Obr. 5. Tento model ukazuje úběr materiálů na předem zadaném polotovaru.

### **3. Simulační modely**

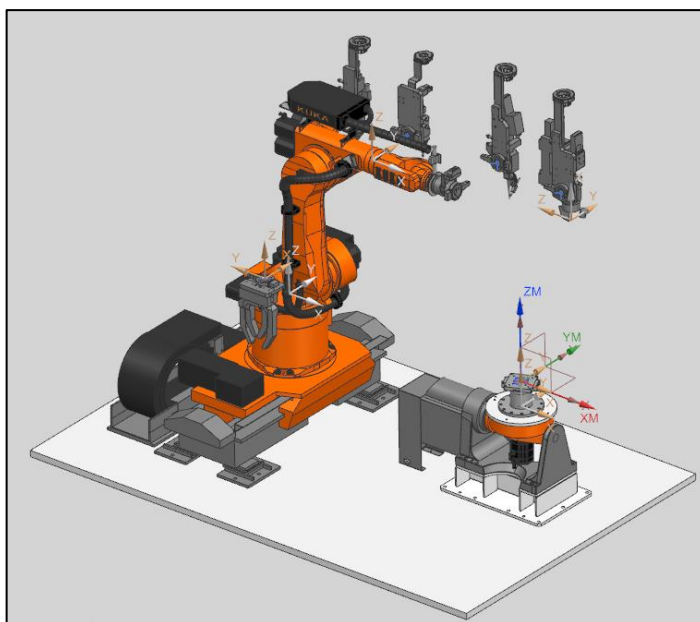
V této kapitole jsou popsány možnosti tvorby simulačních modelů pro offline CSE simulaci, která je předmětem této práce. Offline simulace umožňují verifikaci operací bez spuštění stroje, toto slouží například k rychlejšímu odladění NC kódu. Tyto simulace je také možné pouštět, aniž by technolog byl fyzicky přítomen u stroje, například z domova. Simulační modely jsou tvořeny třemi základními složkami – kinematickým modelem, postprocesorem a emulátorem řídicího systému. V rámci této práce je popsán pouze charakter kinematických modelů a emulátorů řídicích systémů. Postprocesor není předmětem řešení ani praktické části, protože je pro danou robotickou buňku již vytvořen. [8]

#### **3.1. Kinematické modely**

Kinematické modely jsou virtuální reprezentací fyzických částí stroje nebo nástrojových hlavic. Jsou tvořeny 3D modelem daného stroje, který je doplněn o kinematické vlastnosti. Mezi tyto vlastnosti patří souřadné systémy jednotlivých pohonů, jejich rozsahy, rychlosti, ale i směr pohybu či otáčení. Právě díky tomuto modelu je možné z CSE simulace získat reálný čas výrobního procesu nebo jednotlivých operací. V následujících dvou kapitolách jsou stručně popsány části těchto modelů. [8]

### 3.1.1. CAD modely

CAD modely stroje (viz Obr. 12) se dají získat externě, typicky od výrobce. Nicméně si je může uživatel také sám vytvořit (pomocí CAD/CAM softwaru), nebo výrobcem dodávaný model upravit. Mimo modelu stroje jsou potřebné také modely nástrojových hlavic, úchopných zařízení, nástrojů, otočně-sklopných stolů, lineárních pojezdů apod. Siemens ve svém software Siemens NX CAM zahrnuje knihovnu modelů několika strojů a nástrojových hlavic. Všechny tyto modely se získávají obdobně a je potřeba je zahrnout do celkové kinematiky stroje, a to i v případě, že se nepodílí na výrobě. Ukázkovým příkladem je základna či stůl, na kterém je stroj umístěn – zahrnutím do geometrie se zachová vzájemná poloha stroje a obrobku. Musí tedy být zahrnuty, právě kvůli tomu, aby výsledný NC kód byl odbavitelný na reálném stroji. Je tedy třeba dané pracoviště vždy reálně reprezentovat v softwarovém prostředí.



*Obr. 12 CAD model robotické buňky v prostředí Siemens NX CAM*

### 3.1.2. Prvky Machine Component

Jak již bylo zmíněno, CAD model musí být doplněn o kinematické vazby, což se provádí pomocí tzv. k-komponent (v prostředí Siemens NX pod názvem Machine component – strojní komponenta). Jedná se komponenty, jež se vážou na určitou část stroje a obsahují informace o pozici os pohonů a definice těchto pohonů. Samotné komponenty se tvoří v modulu Siemens NX CAM pod názvem Machine Tool Builder. Tvorba by se dala rozdělit do čtyř fází. [8]



V první fázi se tvoří samotné k-komponenty, kterými se definuje jejich umístění v 3D modelu stroje. Komponenty na sebe vzájemně působí a navazují, proto je nutné, aby byly umístěny v kinematickém stromě ve správném pořadí (viz Obr. 13). [8]

KUKA_KR60_L30_HA_DKP400			
[-] MACHINE_BASE	_MACHINE_BASE	MACHINE_BASE_JCT*	
[-] ROBOT_BASE		ROBOT_BASE_JCT*	
[-] J1		J1_JCT	J1
[-] J2		J2_JCT	J2
[-] J3		J3_JCT	J3
[-] J4		J4_JCT	J4
[-] J5		J5_JCT	J5
[-] J6		J6_JCT	J6
[-] POCKET	_DYNAMIC HOLDER	POCKET_JCT*	
[-] POSITIONER_BASE		POSITIONER_BASE_JCT*	
[-] POSITIONER_A1_AXIS		POSITIONER_A1_AXIS_JCT	POSITIONER_A1_AXIS
[-] POSITIONER_A2_AXIS		POSITIONER_A2_AXIS_JCT	POSITIONER_A2_AXIS
[-] POSITIONER_TOOL		POSITIONER_TOOL_JCT*	

Obr. 13 K-komponenty uspořádané do stromu v prostředí Siemens NX CAM

V druhé fázi dochází k přiřazení lokálních a řídicích souřadných systémů. Ty se orientují podle souřadného systému základny stroje nacházejícím se v komponentě *MACHINE\_BASE* (viz Obr. 13). Tento souřadný systém představuje nulový bod stroje. Takové nastavení zaručuje správný pohyb lineárních os, potažmo také rotačních os. [8]

Ve třetí fázi je nutné definovat osy. Ty obsahují informace o své pozici v prostoru, podle jakého osového systému jsou definovány a parametry potřebné k správnému odbavení simulace. Mezi tyto parametry patří například:

- **Axis type** – Parametr udává, o který typ osy se jedná - lineární, rotační, spindle (vřeteno) – spindle se používá např. i pro pohon otočného stolu
- **Axis direction** – Parametr udává směr pohybu dané osy; u rotační vazby nebo spindle se jedná o směr otáčení (u samotného robotu se vždy jedná o směr otáčení – volí se např. X+, to by znamenalo otáčení se kolem osy X po směru hodinových ručiček)
- **Upper/Lower limit** – rozsah os; nemusí být nutně uveden plný rozsah, pokud je stroj např. omezen prostorově
- **Max velocity** – maximální rychlost, kterou lze vyvinout při souvislém pohybu. Při volání funkce rychloposuvu se osa bude pohybovat touto rychlostí
- **Max acceleration/deceleration** – maximální zrychlení/zpomalení
- **Jerk limit** – limit Rychlosti strojných os, kdy Rychlost je veličina popisující změnu rychlosti; jde tedy o časovou derivaci zrychlení

- **Kv** – veličina představující odchylku dosažené polohy od požadované polohy v závislosti na zrychlení pohybu. Pomocí této veličiny lze určit jakou maximální rychlost může stroj vyvinout, tak aby stále splnil podmínku přesnosti
- **Coarse precision** – Hrubá přesnost – pokud provedeme nástrojem pohyb a pokud se nástroj nachází v konečné poloze v této toleranci, je část kódu, která provedla pohyb nástroje, považována za odbavenou
- **Fine precision** – Jemná přesnost – stejná veličina jako coarse precision, pouze nižší hodnota
- **Current Value** – počáteční stav osy

[9]

V Tab. 3.1 je znázorněn přehled některých proměnných nastavovaných v Machine Tool Builderu a jim odpovídající proměnné v řídicím systému Siemens Sinumerik 840D.

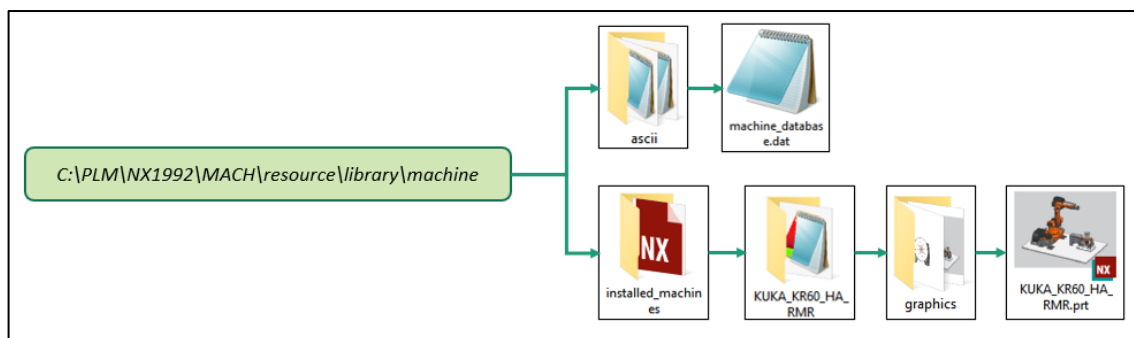
*Tab. 3.1 Přehled proměnných v Siemens NX Machine Tool Builder [9]*

Proměnná v Machine Tool Builderu	Proměnná v Sinumerik 840D
Max. Velocity	MD32000 – \$MAX_AX_VELO
Max. Acceleration/Deceleration	MD32300 – \$MAX_AX_ACCEL
Jerk Limit	MD32431 – \$MAX_AX_JERK
Jump Velocity	MD32310 – \$MAX_AX_OVL_FACTOR
Kv	MD32200 – POSCTRL_GAIN
Coarse Precision	MD 3600 – \$STOP_LIMIT_COARSE
Fine Precision	MD36010 – \$STOP_LIMIT_FINE

Ve čtvrté, a to poslední fázi, je třeba u k-komponent definovat tzv. setup elementy, tj. definovat, zda se jedná o obrobek, polotovár či upínací prvek. Mezi tyto setup elementy patří i prvek *\_DEVICE*, který se v Siemens NX mimo jiné používá k označení nástrojových hlavic. [8]



Po vytvoření celého kinematického modelu je zapotřebí tento model zavést do databáze strojů Siemens NX. Tato databáze je zastoupena souborem `installed_machines` (pro modely nástrojových hlavic by se jednalo o soubor `installed_devices`). Tento soubor se nachází v `C:\PLM\NX1992\MACH\resource\library\machine\installed_machines`. Zde se vytvoří složka s názvem stroje, do které je model zaveden. Název této složky a název souboru s modelem stroje musí být stejné. Nakonec je nutné tuto složku zavést do knihovny ve složce `ascii`, která obsahuje odkazy na jednotlivé stroje. Pomocí této knihovny Siemens NX dokáže jednotlivé stroje načíst a umožňuje jejich použití pro simulace. Schéma umístění souborů kinematického modelu pro Siemens NX je vidět na Obr. 14.



Obr. 14 Schéma umístění souborů pro Siemens NX

## 3.2. Řídicí systémy a emulátory řídicích systémů

Tato kapitola charakterizuje základní komponenty a využití řídicích systémů. Konkrétně je zde popsán řídicí systém Siemens Sinumerik 840D sl, kterým se práce zabývá. Dále jsou popsány emulátory řídicích systémů, princip jejich tvorby a úpravy a jejich podprogramy.

### 3.2.1. Řídicí systémy

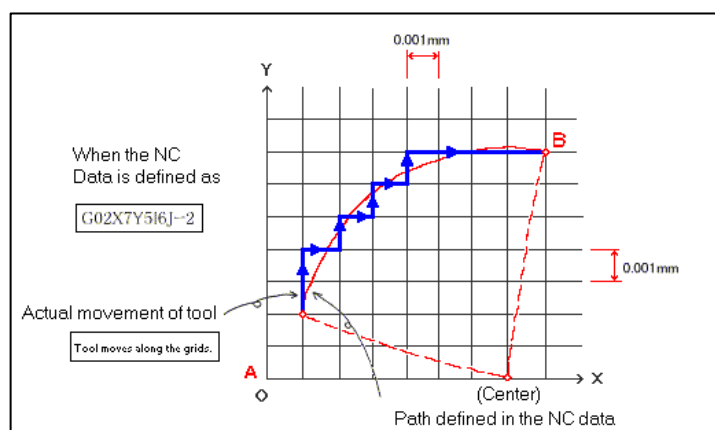
Řídicí systémy obecně slouží k číslicovému řízení a existují pro všechny možné typy strojů a náročnosti řízení. V rámci této práce jsou popsány pouze řídicí systémy používané pro průmyslové CNC obráběcí stroje a roboty.

Řídicí systémy jsou tvořeny počítačem s programovatelným a řídicím softwarem. V prostředí tohoto systému lze programovat požadovaný CNC software, spravovat databáze nástrojů, nastavovat strojní a technologické parametry stroje a ovládací prvky stroje. Mezi tyto prvky patří veškeré části stroje jako jsou například motory, snímače

poloh v motorech, hydraulická a pneumatická vedení. V případě robotické buňky, která je předmětem této práce, je řídicím systémem také řízeno spouštění a regulace zdroje laseru, míchání kovového prášku pro svařování, nebo také přívod drátu pro svařování.

Pro řízení stroje se používá průmyslový počítač dodávaný přímo výrobcem. Na tento počítač je nahrán příslušný řídicí systém. V dnešní době patří mezi největší výrobce Siemens, Heidenhain a Fanuc. Řídicí systém se volí na základě složitosti technologických procesů, které bude stroj řídit. Jedním z parametrů je například počet řízených os. Takový počítač je mimo jiné vybaven ovládacím panelem (viz Obr. 16) s displejem (může být i dotykový). Tento panel je osazen ovládacími tlačítky a strojním panelem s ovládacími prvky pro provoz stroje, USB (Universal Serial Bus) nebo CFC (Compact Flash Card) rozhraním pro nahrání NC kódu z externího zdroje (např. CAM software). Na počítači je možnost ručně spravovat databáze nástrojů nebo programovat NC kód. K tomu se používá klávesnice, která se na panelu také nachází. Tento panel reprezentuje rozhraní člověk-stroj. Samotný řídicí systém by se dal rozdělit na tři nejdůležitější složky – NC složku (Interpolátor), PLC (Programmable Logic Controller) složku a COM (Communication module) složku. [10]

NC složka řídicího systému je tvořena primárně interpolátorem. Je zde počítána dráha nástrojů. Tato dráha závisí na geometrických rozměrech a geometrických parametrech pro danou operaci. Dráha se tzv. interpoluje, tj. dochází zde k převodu jednotkového signálu na spojitou čáru, která nejkratší cestou spojuje počáteční a koncový bod daného úseku, viz Obr. 15. Před vstupem do interpolátoru je signál skokový, a na počátku každé části dráhy musí stroj maximálně zrychlit, pak maximálně zpomalit. Uhlazením tohoto



Obr. 15 Kruhová interpolace [20]

signálu do podoby spojitě křivky se toto chování eliminuje a ve stroji pak nevznikají rázy. Celý běh stroje je pak hladší a přesnější. Důležitou vlastností interpolátoru je jeho schopnost se do NC kódu dívat dopředu. Díky tomu dokáže počítat dráhu v reálném čase a posílat signál do regulátorů pohonů stroje. Toto je také důvodem, proč NC složka řídicího systému musí mít nízkou časovou odezvu. V interpolátoru také dochází k ovládní rychlosti, zrychlení a jerku stroje. [10]

Obecně se nejvíce využívá tří typů interpolace:

- **Lineární interpolace** – Signál se interpoluje na lineární (přímkový) pohyb. Jedná se o nejjednodušší typ interpolace a je vhodná pro jednoduché dílce. U tvarově složitějších dílců dochází k linearizaci povrchu. Tato interpolace je výpočetně náročná. Může u ní dojít ke skokovým změnám rychlosti. To má za následek vznik rázů ve stroji. Rázy vzniklé při lineární interpolaci jsou méně časté než u dráhy před interpolací. Toto je dáno délkou lineárních úseků. Programově je zastoupena funkcí G01.
- **Kruhová interpolace** – Signál se interpoluje na kruhový pohyb. Prokládá povrch obrobku kruhovými dráhami. Eliminuje rázy a linearizaci povrchu vznikající při lineární interpolaci a je méně výpočetně náročná. Programově je zastoupena funkcemi G02 a G03.
- **Křivková interpolace** – Spline interpolace. Prokládá body dráhy křivkou. Eliminuje nepřesnosti lineární interpolace proložením několika lineárních úseků jedním křivkovým. Nedochozí tak k linearizaci povrchu. Tyto křivky mohou dosahovat vysokých řádů.

[18]

Tyto tři typy interpolace se dále dělí na:

- **Interpolace rovinná** – 2D interpolace, pro dvě souřadné osy
- **Interpolace prostorová** – 3D interpolace, pro tři souřadné osy

Další částí řídicího systému je PLC část. Jedná se o část starající se o periferie stroje. Ovládá například pohony motorů vřetena, ale také přídatná zařízení – např. již dříve zmíněné hydraulické a pneumatické agregáty a jejich čerpadla, laser apod. Řízení v této složce nemá tak rychlý takt jako interpolátor, pro porovnání je odezva NC

části zpravidla circa 1ms a odezva PLC části circa 10ms. Jedním z úkolů PLC složky je odlehčit NC části řídicího systému. Tato část se také dá programovat uživatelským softwarem, což může způsobovat dodatečnou časovou odezvu vinou pomalého kódu. V případě robotické buňky v CIIRCu řídí PLC část veškeré náležitosti týkající se svařování – těmi jsou například spouštění laseru, míchání kovových prášků, přívod drátu nebo přesměrování laseru na správnou pracovní hlavici. PLC část je zastoupena fyzickými PLC prvky (viz Obr. 16). [17]

Poslední částí řídicího systému je COM rozhraní – komunikační modul. Je zde obsaženo prostředí člověk-stroj (HMI – Human-Machine Interface), které je reprezentováno zejména uživatelským panelem připojeným k počítači (viz Obr. 16), ale také síťovým připojením, nebo například připojením k dalšímu kooperujícímu stroji či diagnostickým připojením k výrobci. [17]

### 3.2.2. Siemens Sinumerik 840D sl

Robotická buňka, která je předmětem této práce, je vybavena řídicím systémem Siemens Sinumerik 840D sl. Zkratka sl znamená Solution Line. Systém je také nabízen ve verzi Power Line, která je vybavená jinou NCU jednotkou a jiným systémem pohonů. V této kapitole budou popsány dílčí části a konfigurace tohoto řídicího systému. Také jsou zde popsány některé jeho funkce

Počítač řídicího systému Siemens Sinumerik 840D sl je představován jednotkou NCU (Numerical Control Unit). V této jednotce jsou obsaženy NC, PLC a COM části řídicího systému, spolu s regulačními a komunikačními funkcemi. Konkrétně se jedná o jednotku NCU 710.3 (pro 6 řízených os), pro větší počet os se také dodávají varianty 720.3 a 730.3. V této jednotce pak probíhá zpracování NC programu (interpolace). PLC jednotka dodávaná s tímto systémem je řady Simatic S7-300 s řídicí jednotkou CPU317F-3PN/DP. Komunikace PLC a NC části probíhá oboustranně. HMI rozhraní je u řídicího systému Siemens Sinumerik 840D sl realizováno strojním panelem, popřípadě i menším přenosným panelem. Strojní panel se nabízí v několika konfiguracích - je zde možnost vybrat například velikost a typ displeje (dotykový či ne). Připojení HMI jednotky k NCU jednotce je provedeno pomocí tzv. TCU (Thin controller unit) jednotky. Tato jednotka umožňuje prostorové oddělení uživatelského panelu od počítačové části řídicího systému prostřednictvím bezdrátového připojení. [10]

Řídicí systém Sinumerik 840D sl je dále vybaven integrovaným systémem pohonů Sinamics S120. Kompletní řídicí systém včetně PLC části Sinamics S120 pohonů a ovládacího panelu průmyslového počítače je na Obr. 16. [11]



*Obr. 16 HMI panel řídicího systému Sinumerik 840D sl s jeho PLC částí a spolu s pohony Sinamics S120 [20]*

Řídicí systém Siemens Sinumerik 840D sl se řadí mezi nejvíce modulární řídicí systémy na trhu. Toto je jedním z důvodů jeho popularity pro výzkumné potřeby a výroby, kde je třeba vysoká úroveň personalizace stroje a jeho periferií. [12]

### **3.2.3. Funkce řídicího systému Siemens Sinumerik 840D sl**

Funkce řídicího systému se dělí na několik typů. Dvě největší skupiny jsou G funkce a M funkce. G funkce zastupují interpolační (NC) část řídicího systému a patří sem například dříve zmíněné funkce G01, G02 a G03. Dalším zástupcem těchto funkcí je funkce G00, která se používá k rychloposuvu. [13]

M funkce řídicího systému sdělují PLC části řídicího systému, kdy má provést určité operace. Tyto funkce jsou spolu se svými parametry přenášeny na PLC rozhraní řídicího systému, kde jsou zpracovány uživatelským PLC softwarem. Jedna z funkcí v této skupině je funkce M6 značící výměnu nástroje. Zavedení funkce M6 do emulátoru řídicího systému je jedním z úkolů této práce. V Tab. 3.2 jsou k nahlédnutí některé další M-funkce. Pro další typy funkcí viz literatura [13]. [13]

*Tab. 3.2 Přehled M-funkcí řídicího systému Siemens Sinumerik 840D sl/828D [13]*

<b>M-funkce</b>	<b>Význam</b>
M0	Programovatelné zastavení
M1	Volitelné zastavení
M2	Konec hlavního programu s návratem na začátek programu
M3	vřeteno se otáčí doprava
M4	vřeteno se otáčí doleva
M5	zastavení vřetena
M6	Výměna nástroje (standardní nastavení)
M17	Konec podprogramu
M19	Polohování vřetena
M30	Konec programu (stejně jako M2)
M40	Automatické přepínání stupňů převodovky
M41	1. stupeň převodovky
M42	2. stupeň převodovky
M43	3. stupeň převodovky
M44	4. stupeň převodovky
M45	5. stupeň převodovky
M70	přepnutí vřetena do osového režimu

### **3.2.4. Sinumerik Run MyRobot**

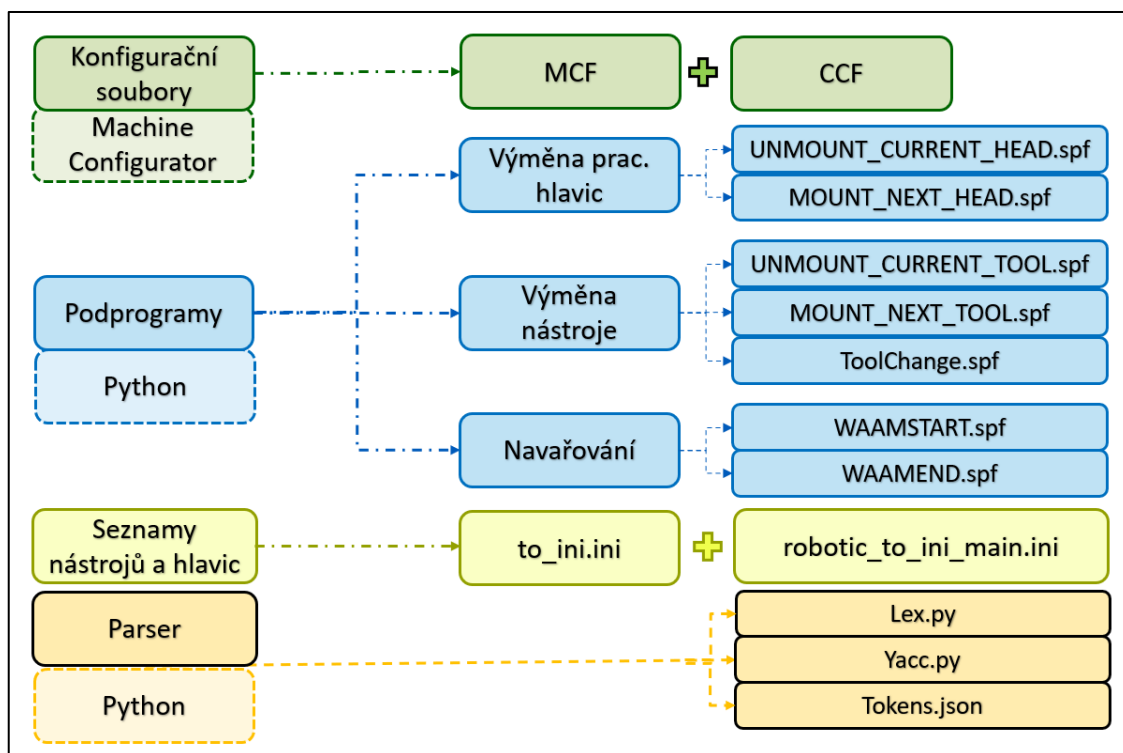
Sinumerik Run MyRobot je softwarové řešení firmy Siemens. Představuje komunikační spojení řídicích systémů Siemens přímo s robotem, bez potřeby pro další ovládání robotu externí jednotkou (vlastním počítačem). Umožňuje komunikaci již dříve vyvinutých řídicích systémů (jako je Sinumerik 840D sl) s roboty. Integruje do řídicího systému kinematickou konfiguraci robotu. Také umožňuje přepočítání souřadnic X, Y a Z na natočení kloubových pohonů robotů. Díky tomuto řešení je ovládání robotů podobné řízení CNC obráběcích strojů. Prostředí Run MyRobot je nejlépe přizpůsobeno robotům KUKA. [15]

### 3.2.5. Emulátory řídicích systémů

Emulátory řídicích systémů jsou virtuální, zjednodušená zastoupení řídicích systémů. Obecně se dají rozdělit na interpolační (NC) část a PLC část. Narozdíl od reálného řídicího systému nejsou zastoupeny žádným počítačem, pouze programovatelnými soubory. Obdobně jako u reálného řídicího systému se tyto dvě části starají o dvě různé části. Interpolační část se stará o dráhu nástroje a obrobku, PLC část se stará o periferie virtuálního stroje.

Emulátory řídicích systémů v Siemens NX jsou tvořeny několika typy souborů, mezi nejvíce zastoupené se řadí MCF (Machine Configurator File), CCF (Controller configuration file), Python soubory a knihovny nástrojů. CCF soubory jsou konfigurační soubory řídicího systému. Obsahují chráněné knihovny specifického řídicího systému. Obsahují obecné definice, jako je například: příkaz G0 značí rychloposuv, příkaz F značí posuv, S značí otáčky vřetene apod. MCF soubory jsou strojní konfigurační soubory. V těchto souborech jsou specifikovány vlastnosti os, kanály řízení a parametry pohonů stroje. Tyto soubory je nutné přesně specifikovat pro každou kombinaci řídicího systému a geometrických vlastností stroje. Pro tvorbu funkcí v souborech MCF je třeba znát syntaxi XML (Extensible markup language) jazyka a příkazy v CSE a AC (Any-Controller) jazyku pro řídicí systémy stroje. Python soubory se starají o překlad syntaxe jazyka příslušného řídicího systému, neboli o takzvaný parsing. Tento parser je tvořen několika soubory, mezi ty patří soubory yacc, lex a tokens. Bez parseru není emulátor schopen číst ani interpretovat NC kód. Obsah NC kódu je určen použitým postprocesorem a tak je pro funkčnost simulace nezbytné parser rozšířit o příkazy generované tímto postprocesorem. V python syntaxi jsou dále také napsány podprogramy emulátoru. Poslední částí emulátoru řídicího systému jsou knihovny nástrojů, jedná se o soubory *to\_ini.ini* a *robotic\_to\_ini\_main.ini*. Na Obr. 17 je schéma dílčích částí emulátoru řídicího systému. [16]



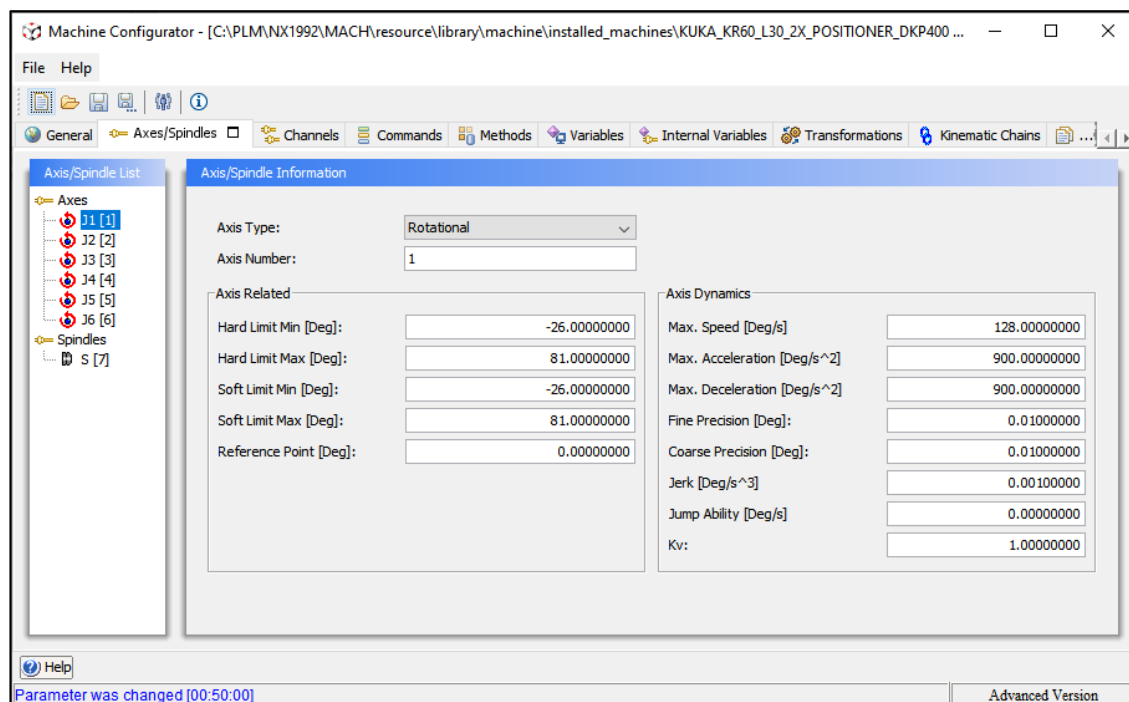


Obr. 17 Schéma dílčích částí emulátoru řídicího systému

### 3.2.6. Machine Configurator

Machine Configurator je samostatný program ze skupiny Technomatix od firmy Siemens. Je provázán s programem Siemens NX CAM. V programu Machine Configurator probíhá tvorba a úprava souborů MCF a CCF. V programu lze dále konfigurovat i rozsahy, rychlosti a zrychlení os, nicméně toto od verze Siemens NX 10 lze definovat přímo v nadstavbě Machine Tool Builder. Pro tvorbu a úpravu funkcí je třeba znát syntaxi XML (Extensible Markup Language) programovacího jazyka a příkazy používané v CSE (Common Simulation Engine) a AC (Any Controller) jazycích pro řídicí systémy stroje. Prostředí softwaru Machine Configurator je vidět na Obr. 18. [16]





Obr. 18 Machine Configurator

### 3.2.7. Podprogramy emulátoru řídicího systému

Podprogramy emulátoru řídicího systému jsou samostatné bloky kódu sloužící k odbavení různých PLC funkcí v rámci simulace. Tyto soubory jsou uživatelsky upravitelné, je také možnost vytvořit si podprogramy nové. V NC kódu se označují příkazy, jako je například M6 – výměna nástroje. Když se v NC kódu zavolá funkce M6, tak emulátor řídicího systému zavolá podprogram na ní navázaný. V případě robotické buňky v CIIRCu to je podprogram *ToolChange.spf*, který pak volá další podprogramy, jako je *UNMOUNT\_CURRENT\_TOOL.spf*, starající se o odepnutí nástroje. Po odbavení všech spolu navázaných funkcí se v simulaci provede výměna nástroje, včetně vizualizace najetí robotické ruky do pozice zvoleného nástroje v zásobníku. Podrobné popisy a vývojové diagramy průběhu těchto funkcí jsou blíže popsány v praktické části této práce. Na Obr. 19 je vidět jeden z kódů podprogramů starajících se o výměnu pracovní hlavice. Tento kód provádí upnutí další pracovní hlavice v pořadí. Jedná se o kód pro vzorovou simulaci robotu v prostředí Siemens NX.

```
##LANGUAGE AC

INT nNextIndex;
nNextIndex = GetRobotNextToolCarrierIndex();

STRING sHeadName;
STRING sHeadTargetKinCarrier;

sHeadName = GetRobotHead(nNextIndex);
sHeadTargetKinCarrier = GetRobotHeadTargetKinCarrier(nNextIndex);

IF(sHeadName != "");
    IF(getCurrentHead(sHeadTargetKinCarrier) != "");
        visibility(getCurrentHead(sHeadTargetKinCarrier),OFF,TRUE);
        unmountHead(sHeadTargetKinCarrier);
    ENDIF;

    mountHead (sHeadName, sHeadTargetKinCarrier);
    grasp (sHeadName, sHeadTargetKinCarrier);
    position (sHeadName, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0);
    visibility (sHeadName, ON, TRUE);
ENDIF;

##LANGUAGE NATIVE

M17
```

*Obr. 19 Kód podprogramu pro výměnu pracovní hlavy*

### 3.3. Popis robotické buňky

Robotická buňka, kterou se tato práce zabývá, je využívána pro výzkumné činnosti a je tedy stále ve vývoji. K buňce je vytvořen funkční a odladěný postprocessor. Celá robotická buňka je ovládána řídicím systémem Siemens Sinumerik 840D sl. K buňce je také již vytvořen kinematický model pro prostředí Siemens NX CAM. Robot je umístěn na lineární ose a je doplněn o otočně sklopný stůl.

K práci robot používá čtyři pracovní hlavice. První hlavice používá směs kovových prášků, druhá navařuje pomocí drátu, třetí má také přívod drátu (hlavice typu wobbler) a čtvrtá hlavice je řezací. Na Obr. 21 jsou k nahlédnutí všechny čtyři tyto hlavice. V Tab. 3.3 je vidět označení hlavic v kódu. Ve zbytku práce bude na hlavice odkazováno pomocí těchto označení.

*Tab. 3.3 Označení pracovních hlavic v kódu*

Název pracovní hlavice	Název hlavice v kódu
Prášková hlavice	YC52
Hlavice s přívodem drátu	COAX
Hlavice typu wobbler	WOBBLE
Řezací laserová hlavice	SOLID

Všechny NC ovládané části robotické buňky jsou již zavedeny do kinematického modelu, který zahrnuje samotný stroj, jeho otočně-sklopný stůl, pracovní hlavice a lineární pojezd. V současné době není pro robotickou buňku vytvořen emulátor řídicího systému.

### 3.3.1. Specifické funkce generované postprocesorem

V této kapitole jsou vyjmenovány některé funkce generované dodaným postprocesorem. Pro všechny pracovní hlavice se generuje určitý blok funkcí před a po navařovací operaci. Generování těchto funkcí lze nastavit při navrhování výrobního procesu. Na Obr. 20 je k nahlédnutí NC kód generovaný dodaným postprocesorem pro pracovní hlavici YC52. V Tab. 3.4 je pak vidět přehled funkcí, kterými se volají určité podprogramy. Toto bude dále přiblíženo v praktické části práce. Výpis všech funkcí generovaných postprocesorem pro každou hlavici je vidět v příloze č. 20.

```

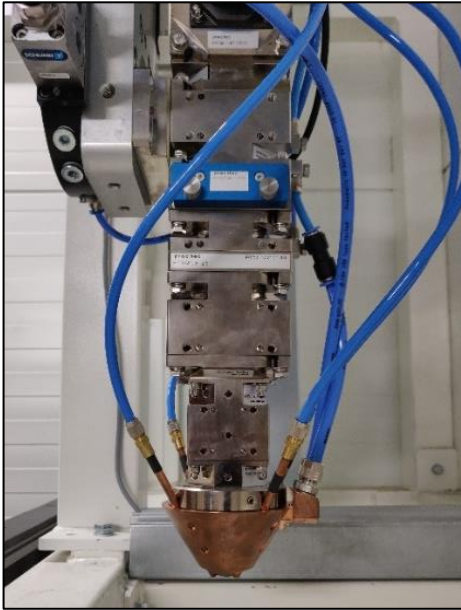
N10 T="YC52" D1 M6
N20 M81 ;start YC52 procesu
N30 ;
N40 STOPRE
N50 TRAORI
N60 TRAORI_ROBOT_TABLE
N70 G54
N80 G1 PTP X-49. Y0.0 Z9. A=0.0 B=0.0 C=-0.003 STAT=6 TU=27 RB8=0.000000 RA9=0.000000 RL7=0.000000 F5000.
N90 STOPRE
N100 H19=0.100 ;koeficient modulace výkonu
N110 H20=0 ;koeficient modulace posuvové rychlosti
N120 H17=3 ;aktivace modreho svetla (11010)
N130 M12 ;termokamera (zapnuti)
N140 H10=0 ;vykon laseru
N150 M10 ;zapnuti laserove emise
N160 H31=5.0 ;rotace disku (1)
N170 H33=11.0 ;prutok plynu (1)
N180 H35=11 ;mixer (1)
N190 STOPRE
N200 M31 ;zapnuti prasek 1
N210 G4 F0.020 ;casova prodleva pro sypani prasku
N220 H10=1200 ;vykon laseru
N230 G4 F0.020 ;casova prodleva pro rozjezd robota
N240 STOPRE

```

*Obr. 20 NC kód generovaný dodaným postprocesorem*

*Tab. 3.4 Funkce generované postprocesorem, jejich využití a příslušné podprogramy*

Název funkce	Využití funkce	Volaný podprogram
M6	Výměna nástrojové sestavy	ToolChange.spf
H10	Regulace výkonu laseru – ovládání navařování	WAAMSTART.spf a WAAMEND.spf



*Obr. 21 Pracovní hlavice robotické buňky*

*Vlevo nahoře – prášková hlavice, vpravo nahoře – hlavice s přívodem drátu, vlevo dole – wobbler, vpravo dole – laserová řezací hlavice*

### 3.4. Závěr rešerše

Úkolem rešeršní části této práce bylo přiblížit problematiku virtualizace stroje pro prostředí Siemens NX. Bylo vysvětleno, že pro programování robotů je zapotřebí v softwaru Siemens NX využít dodatečných nástrojů. Také byla přiblížena problematika simulací v prostředí Siemens NX a jaké softwarové komponenty jsou pro její vytvoření nezbytné.

Jelikož je robotická buňka soukromě vyvíjená, neexistuje pro ni dostupné komerční řešení, a tudíž je nutno navázat na předchozí vývoj. Tvorba simulačního modelu proběhne úpravou souborů a modelů již dodávaných v Siemens NX. Charakter práce je primárně programátorský. Kinematický model stroje je již z části vyvinut, nicméně musí dojít k jeho rozšíření.

Softwary použité pro samotnou realizaci jsou Siemens NX spolu s rozšířením Machine Tool Builder, software Machine Configurator a libovolný textový editor pro skripty v jazyce Python. Bakalářská práce navazuje na již rozpracovaný simulační model. Díky tomuto jsou zvolené softwary od firmy Siemens nejvhodnější, jelikož by využití jiného softwaru vyžadovalo předělání již existujících softwarových komponent buňky. To znamená, že neproběhne výběr variant řešení.

Pomocí informací a softwarů představených v rešeršní části této práce vzniknul emulátor řídicího systému a podprogramy provádějící výměnu nástrojových hlavic a podprogramy na start a konec navařování.

## 4. Návrh vlastního řešení

V rámci realizace simulačního modelu neproběhl výběr varianty řešení. Emulátor řídicího systému byl vyvinut pro prostředí Siemens NX. Tato varianta byla jedinou použitelnou variantou, z důvodu navazování na již vytvořené části simulačního modelu. Při vytváření dílčích částí simulačního modelu je možné začít tyto části tvořit od začátku. Nicméně při použití řídicího systému, jako je Siemens Sinumerik, je možné pouze upravit již existující emulátory, postprocesory a kinematické modely dostupné v Siemens NX. V této práci byl emulátor řídicího systému realizován úpravou právě již existujícího emulátoru v Siemens NX. Tento výchozí vzorový emulátor byl vytvořen pro kombinaci robotu KUKA KR60HA s řídicím systémem Siemens Sinumerik 840D, tudíž je vhodný pro aplikaci v robotické buňce. Celková úprava simulačního modelu probíhala ve třech fázích.

V první fázi byl upraven kinematický model. Úprava proběhla v prostředí Siemens NX 12 a byl použit již dodaný kinematický model. Práce s kinematickými modely byla popsána v kapitole 3.1, samotná úprava modelu je popsána v kapitole 4.

V druhé fázi byly upraveny podprogramy emulátoru řídicího systému. Při úpravě se vycházelo z podprogramů obsažených v Siemens NX 12. Dále bylo v rámci věrohodnosti využito podprogramů použitých u reálného stroje. Úprava podprogramů je blíže vysvětlena v kapitole 5.2.

Ve třetí fázi byl upraven parser emulátoru řídicího systému. V rámci této úpravy byly doplněny některé knihovny parseru a byly k nim vytvořeny odpovídající funkce.

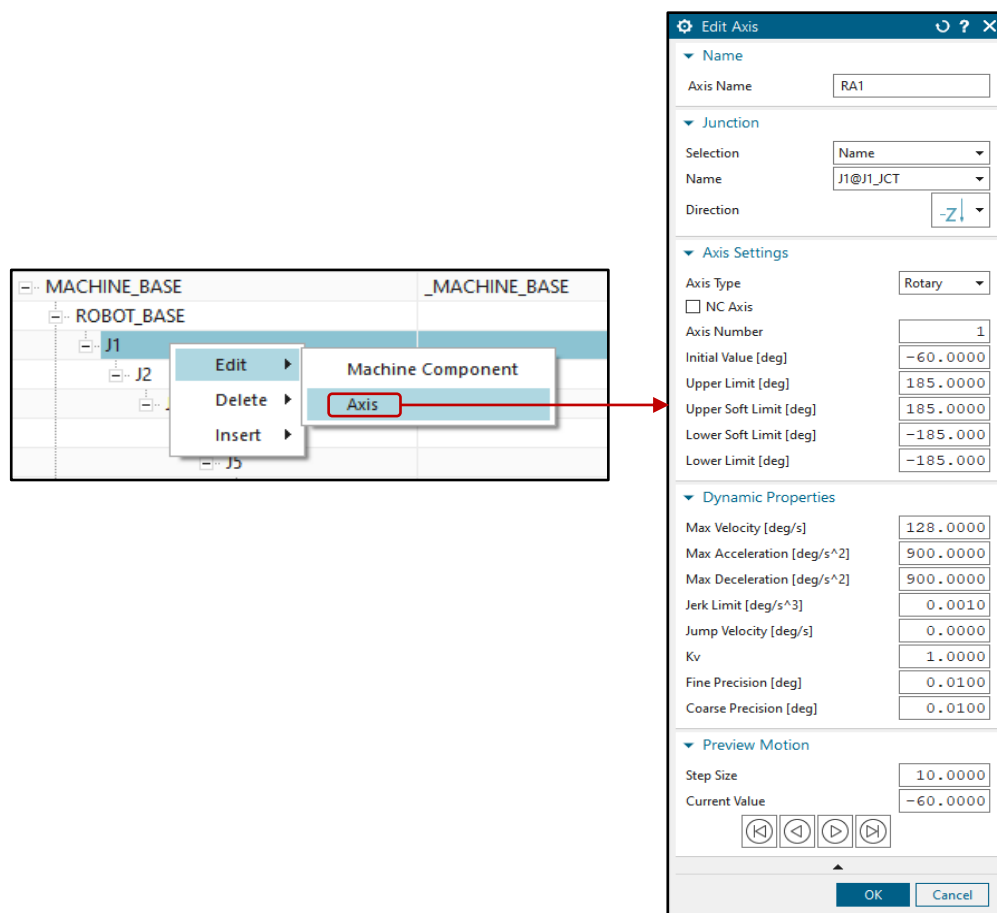
Posledním krokem bylo připravení a odbavení vzorové simulace. Blíže popsáno v kapitole 5.6.

## 5. Aplikace metodiky na robotické buňce

V této kapitole je na základě analyzovaných informací k robotické buňce, a metodiky popsané v rešeršní části, vytvořen funkční emulátor řídicího systému v Siemens NX 12.

### 5.1. Úprava kinematického modelu

Pro zajištění správného chodu simulace je potřebný kinematický model odpovídající reálnému stroji. Pro robotickou buňku byl dodán již vytvořený kinematický model obsahující samotný robot, jeho pracovní hlavice, lineární pojezd a otočně-sklopný stůl. Simulačnímu modelu byly poté upraveny rozsahy os tak, aby odpovídaly reálnému nastavení stroje. Pro změnu parametrů os bylo potřebné si pro danou komponentu otevřít nastavení „Axis“ a zadat správné údaje (viz. Obr. 22). Hodnoty pro rozmezí os, jejich rychlosti pohybu a zrychlení byly získány z katalogu firmy KUKA. Model byl upraven ve dříve zmíněné nadstavbě programu Siemens NX zvané Machine Tool Builder.



Obr. 22 Nastavení rozsahu os v Siemens NX



## 5.2. Tvorba podprogramů emulátoru řídicího systému

Podstatnou částí emulátoru řídicího systému jsou jeho podprogramy, které zajišťují různé funkce. V rámci této práce byly upraveny podprogramy zajišťující výměnu nástroje a nástrojové hlavice a podprogramy starající se o navařovací procesy. Z praktického důvodu byly pro práci použity již existující vzorové podprogramy pro robot KUKA KR60 HA dostupné v souborech pro Siemens NX. Tyto soubory byly postupně upravovány. V první řadě byly podprogramy přizpůsobeny již vytvořenému postprocesoru. To znamená, že podprogramy musely umět pracovat s názvem robotické hlavice, zatímco původní vzorové podprogramy pracovaly s názvem nástroje. Z toho plynula potřeba vytvoření nového způsobu indexování nástrojů i hlavic. Dalším krokem bylo doplnění potřebných pohybů pro věrohodnou simulaci výměny pracovních hlavic.

### 5.2.1. Přehled podprogramů emulátoru řídicího systému pro robotickou buňku

Pro doplnění simulačního modelu je zapotřebí vytvořit řadu podprogramů. Mezi ty se řadí podprogramy starající se o výměnu pracovních hlavic a podprogramy pro navařovací procesy. Pro správnou funkčnost těchto podprogramů je také nutné vytvořit databáze nástrojů, které podprogramy při své funkci využívají. V Tab. 5.1 je k nahlédnutí přehled všech podprogramů a databází, které emulátor využívá a které bude potřeba upravit.

*Tab. 5.1 Seznam podprogramů a knihoven emulátoru řídicího systému pro robotickou buňku*

<b>Podprogramy pro výměnu pracovních hlavic</b>	<b>Funkce podprogramu</b>
ToolChange.spf	Hlavní program výměny hlavic
UNMOUNT_CURRENT_TOOL.spf	Odepnutí stávajícího nástroje
MOUNT_NEXT_TOOL.spf	Upnutí dalšího nástroje
UNMOUNT_CURRENT_HEAD.spf	Odepnutí stávající hlavice
MOUNT_NEXT_HEAD.spf	Upnutí další hlavice
<b>Podprogramy pro navařovací procesy</b>	<b>Funkce podprogramu</b>
WAAMSTART.spf	Start navařování
WAAMEND.spf	Konec navařování
<b>Knihovny nástrojů a nástrojových hlavic</b>	<b>Funkce knihovny</b>
to_ini_main.ini	Knihovna nástrojů
robotic_to_ini_main.ini	Knihovna nástrojů a nástrojových hlavic

### 5.2.2. Indexování pracovních hlavic a nástrojů

Při úpravě podprogramů pro jejich využití v robotické buňce byl zjištěn problém se způsobem indexování a rozpoznávání nástrojů. Přejaté vzorové programy pro poskytnutí čísla použitých hlavic a nástrojů spoléhaly na vzorovém postprocesoru. Ten při postprocesování vytvořil soubory *to\_ini.ini* a *robotic\_to\_ini\_main.ini*, ve kterých byly vypsány použité hlavice a jím odpovídající číselné indexy. Postprocesor dodaný k robotické buňce však tuto funkčnost nemá, bylo tedy zapotřebí tyto soubory vytvořit ručně. Nově vytvořené soubory jsou statické – nikdy se nemění. To je možné díky tomu, že se v buňce stále nacházejí stejné nástrojové hlavice.

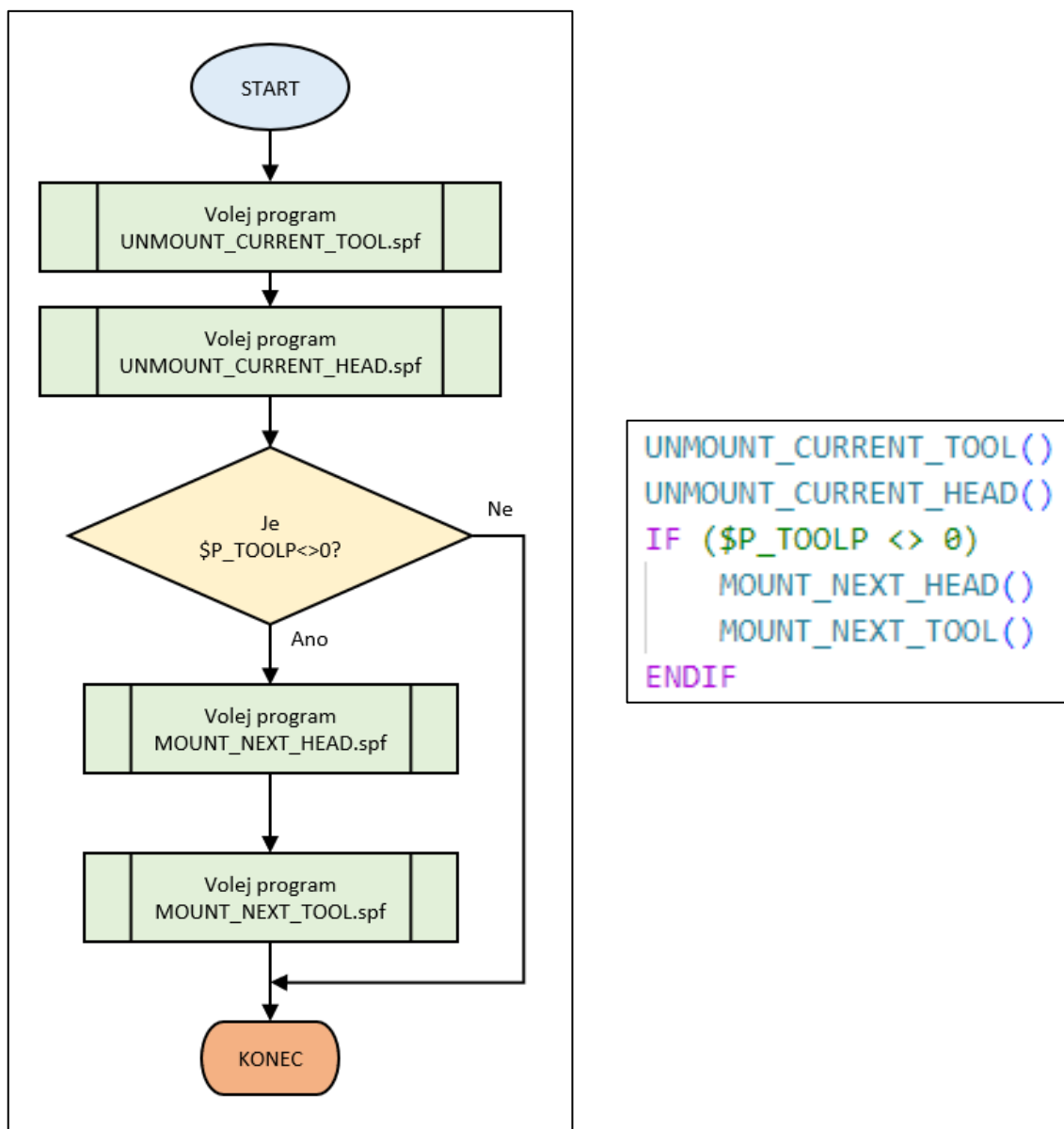
Dalším problémem bylo pracování s číselnými indexy hlavic. Přejatý vzorový emulátor získával indexy z původních souborů *to\_ini.ini* a *robotic\_to\_ini\_main.ini*. K těmto datům se dostával přes řetězec funkcí obsažených v interních programech simulátoru pro Siemens NX. To však s novými statickými soubory nefungovalo, což vedlo ke změně způsobu, kterým emulátor číselné indexy získává. Realizace proběhla přímým odkazováním na soubory *to\_ini\_bunka.ini* a *robotic\_to\_ini\_main\_bunka.ini*. Číselný index hlavice, u které dochází k upínání, se ukládá do proměnné *\$P\_TOOLP*, číselný index současně upnuté hlavice se ukládá do proměnné *\$P\_CURRT*. Obě tyto proměnné se nachází v hlavním MCF souboru emulátoru řídicího systému (název souboru je *KUKA\_KR60\_L30\_2X\_POSITIONER\_DKP400.MCF*). Na Obr. 23 je k nahlédnutí část kódu souboru *robotic\_to\_ini\_main\_bunka.ini*. Tato část náleží pracovní hlavici pod názvem YC52. Je zde vypsán název hlavice, nástroje, kinematického řetězce a nosičů nástrojů a hlavic (položky s příponou Carrier). Kinematický řetězec a nosiče se definují při tvorbě kinematického modelu (viz kapitola 3.1.2).

```
SetRobotTool(1, "YC52_T");  
SetRobotHead(1, "YC52");  
SetRobotChain(1, "CHAIN");  
SetRobotToolInitialKinCarrier(1, "POCKET_HEAD1");  
SetRobotToolTargetKinCarrier(1, "POCKET_HEAD1");  
SetRobotHeadInitialKinCarrier(1, "TOOL_RACK_T1_JCT");  
SetRobotHeadTargetKinCarrier(1, "POCKET_JCT");
```

Obr. 23 Část kódu databáze *robotic\_to\_ini\_main\_bunka.ini*

### 5.2.3. Podprogramy pro výměnu pracovních hlavic a nástrojů

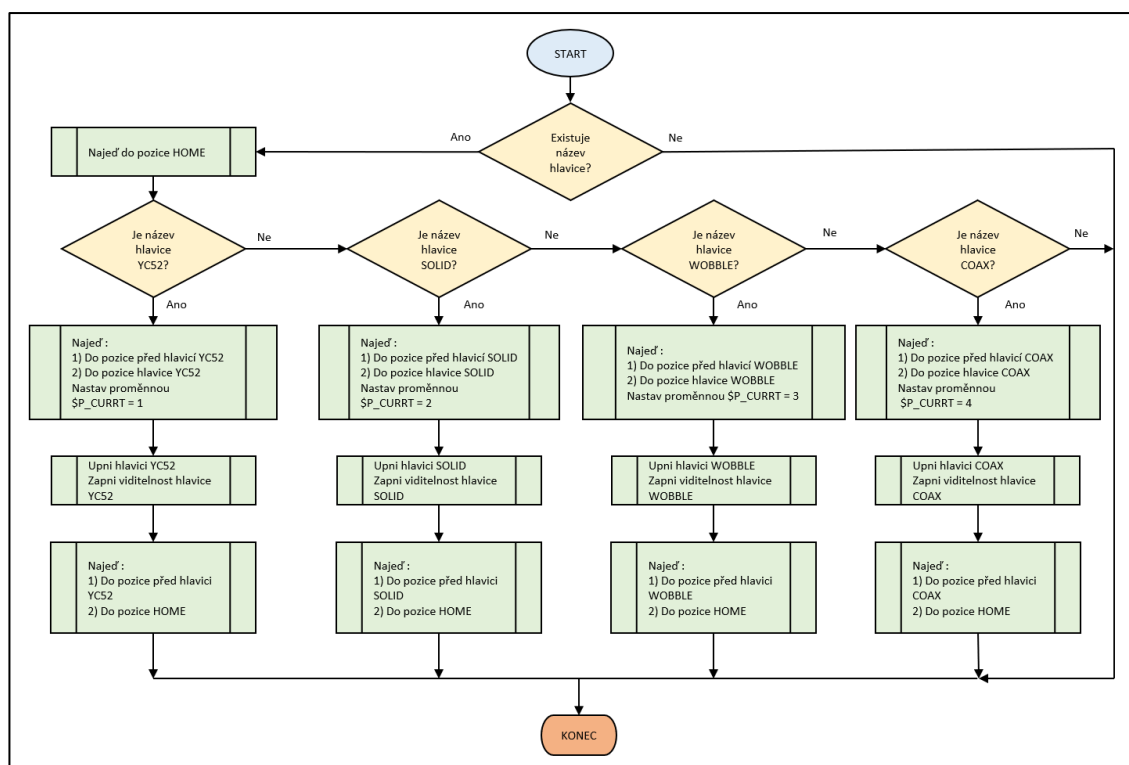
O výměnu nástrojových hlavic a nástrojů se stará soubor pěti programů. Při zavolání funkce M6 v NC kódu se v rámci simulace vstupuje do podprogramu výměny nástroje pod názvem *ToolChange.spf*. Jedná se o jednoduchý program, který pouze volá a odbavuje další podprogramy. Na něj navazující podprogramy jsou blíže popsány v této kapitole. Schéma podprogramu *ToolChange.spf* a část jeho kódu je vidět na Obr. 24. Proměnná  $\$P\_TOOLP$  zde představuje číselný index upínané hlavice (viz kapitola 5.2.2).



Obr. 24 Pracovní diagram podprogramu *ToolChange.spf* (vlevo) a část kódu volající další podprogramy výměny nástrojové sestavy (vpravo)

## ***UNMOUNT\_CURRENT\_HEAD a MOUNT\_NEXT\_HEAD***

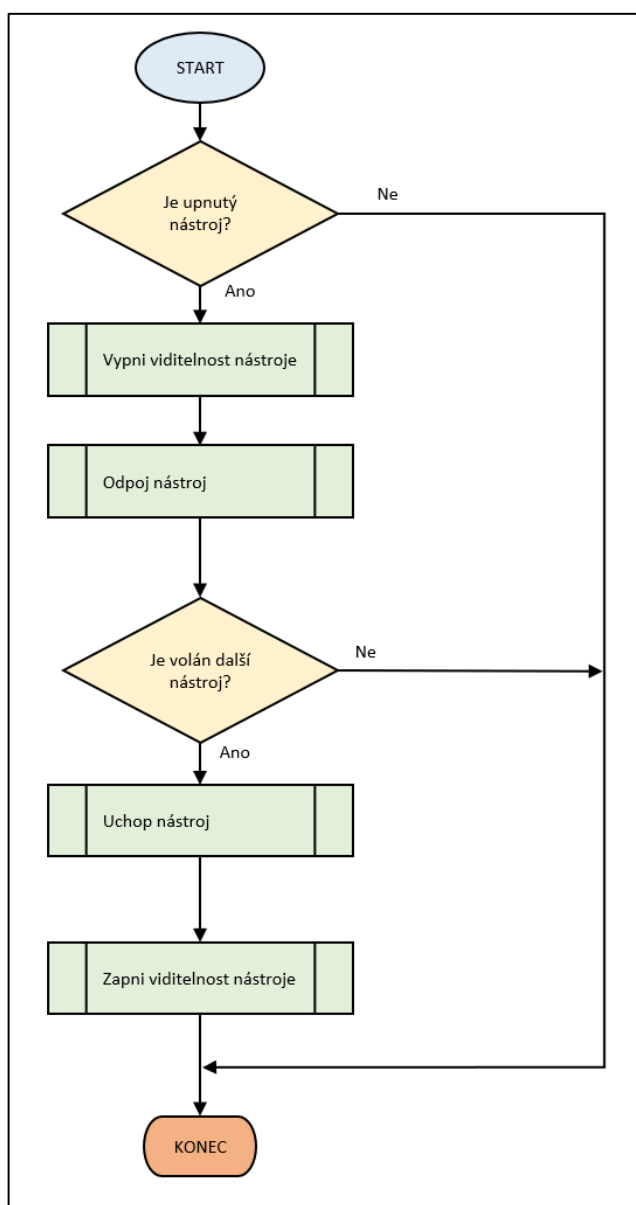
Jedná se o podprogramy starající se o výměnu samotné hlavice. Pracují s indexy ze souborů *to\_ini\_bunka.ini* a *robotic\_to\_ini\_main\_bunka.ini*. Podprogram postupně porovnává vstupní jméno hlavice z NC kódu se jmény hlavic v kódu samotného podprogramu. Název hlavice se získá z jejího číselného indexu. Propojení číselných indexů hlavic s jejich jmény je provedeno v souboru *robotic\_to\_ini\_main.ini*, kde také dochází k provázání nástrojové hlavice s odpovídajícím nástrojem. Po shodě se vždy odbaví část kódu obsahující souřadnice pohybu robotu k zvolené hlavici. V tomto kroku se také nastavuje hodnota proměnné *\$P\_CURRT*. Do této proměnné se ukládá číslo aktuálně připnuté hlavice. Toho se později využívá v programu pro odepnutí hlavice. Vývojový diagram funkce *MOUNT\_NEXT\_HEAD.spf* je zobrazen na Obr. 25. Diagram funkce *UNMOUNT\_CURRENT\_HEAD.spf* je analogický a je k nahlédnutí v příloze. Hlavní úpravou těchto dvou podprogramů bylo doplnění pohybů robotu při výměně pracovní hlavice. V rámci simulace je toto u podprogramů nejpodstatnější úkon a umožňuje to kontrolu kolizí při výměně hlavice.



Obr. 25 Vývojový diagram podprogramu *MOUNT\_NEXT\_HEAD.spf*

## ***UNMOUNT\_CURRENT\_TOOL a MOUNT\_NEXT\_TOOL***

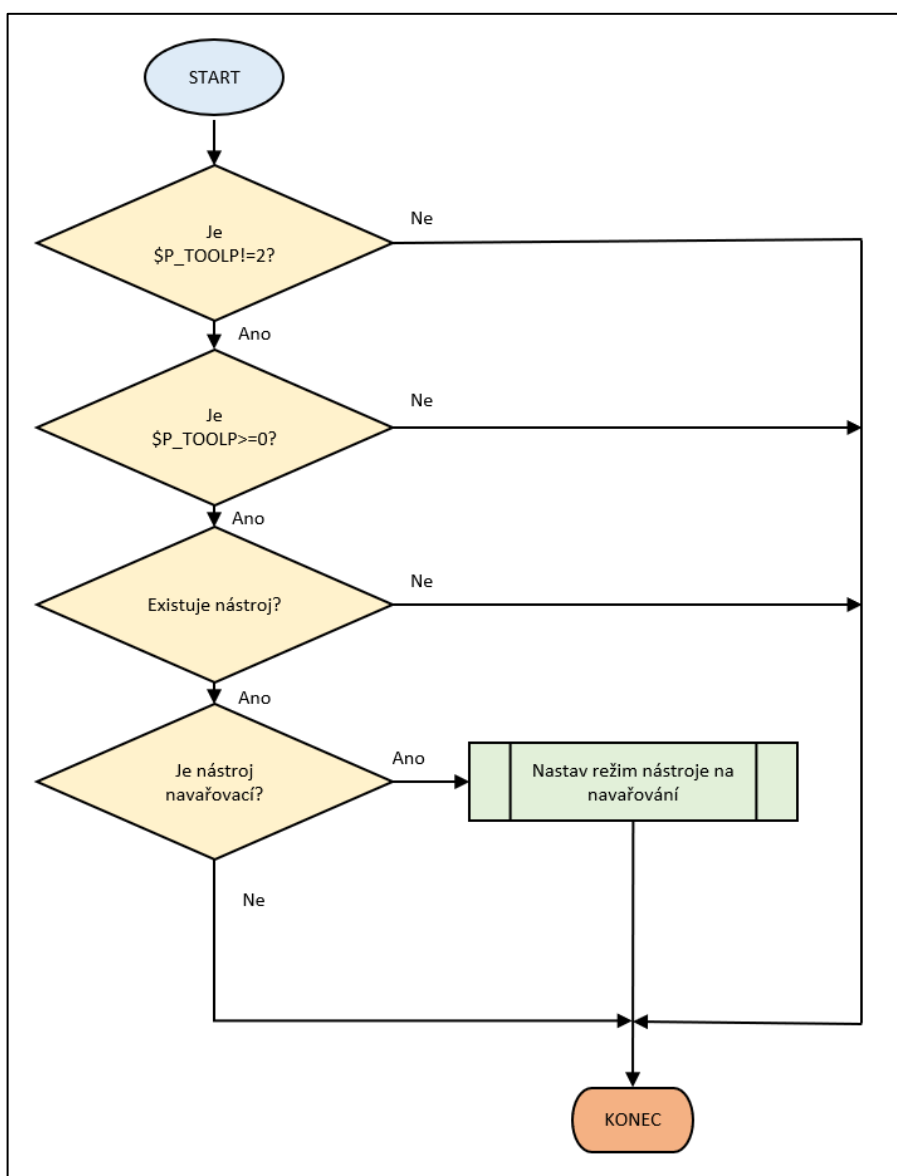
Tyto dva podprogramy se starají o výměnu nástroje. Opět využívají číselných a jmenných indexů ze souborů *to\_ini\_bunka.ini* a *robotic\_to\_ini\_main\_bunka.ini*. Při jejich chodu dochází k přečtení názvu pracovní hlavice z NC kódu. Podle tohoto si program v souborech *.ini* najde odpovídající nástroj (nikoliv hlavici) a následně dojde k jeho upnutí do nástrojové hlavice. Při upínání nástroje nedochází k žádným pohybům, jelikož všechny byly provedeny během výměny nástrojové hlavice. Pracovní diagram podprogramu starajícího se o upnutí dalšího volaného nástroje je vidět na Obr. 26.



Obr. 26 Podprogram *MOUNT\_NEXT\_TOOL.spf* starající se o výměnu nástroje

### 5.2.4. Podprogramy pro navařovací procesy

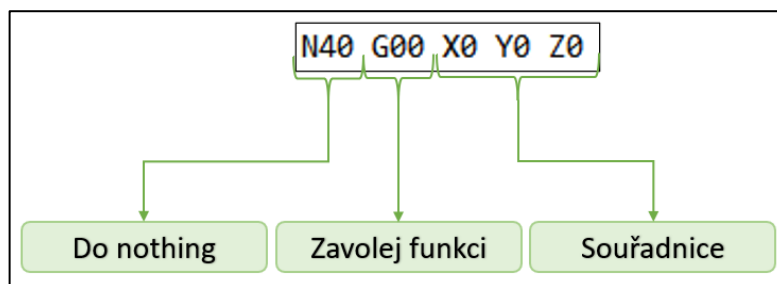
O navařovací procesy se starají dva podprogramy – *WAAMSTART.spf* a *WAAMEND.spf*. Oba jsou v NC kódu volány jako funkce H10, respektive M89 (blíže vysvětleno v kapitole 4.4). Pro aplikaci v robotické buňce v těchto souborech došlo pouze ke změně přiřazování indexů nástrojů. Pracovní diagram funkce *WAAMSTART.spf* je vidět na Obr. 27. Diagram funkce *WAAMEND.spf* vypadá analogicky a je k nahlédnutí v příloze. Programy *WAAMSTART.spf* i *WAAMEND.spf* nespouští ani nevypínají navařovací procesy v případě použití hlavice SOLID (viz Obr. 27), protože tato hlavice slouží k řezání materiálu, nikoliv k aditivnímu navařování.



Obr. 27 Vývojový diagram podprogramu *WAAMSTART.spf*

### 5.3. Úprava parseru

Parser je část emulátoru řídicího systému starající se o komunikaci mezi NC kódem a emulátorem. Jedná se o překladač jazyka, který překládá jazyk NC kódu na jazyk programů simulátoru pro prostředí Siemens NX (ten používá převážně syntaxi jazyka python). Skládá se z několika souborů a databází. Funguje na základě postupného čtení řádku, kde si každý znak, číslo či písmeno přečte a ověří, zda se na řádku smí vyskytnout. Dále pak zjišťuje, zda má obsah řádku spustit nějaký proces (například spuštění funkce), nebo se jedná jen o prázdnou informaci. Na Obr. 28 je vidět jednoduchý příklad interpretace NC kódu parserem. V prvním kroku parser přečte „N40“, což interpretuje jako číslování řádku. V druhém kroku se vyskytuje příkaz G00, jenž značí spuštění funkce G00 – rychloposuv. Tato funkce pracuje s proměnnými (souřadnicemi), ty parser získává z příkazů X0 Y0 Z0.

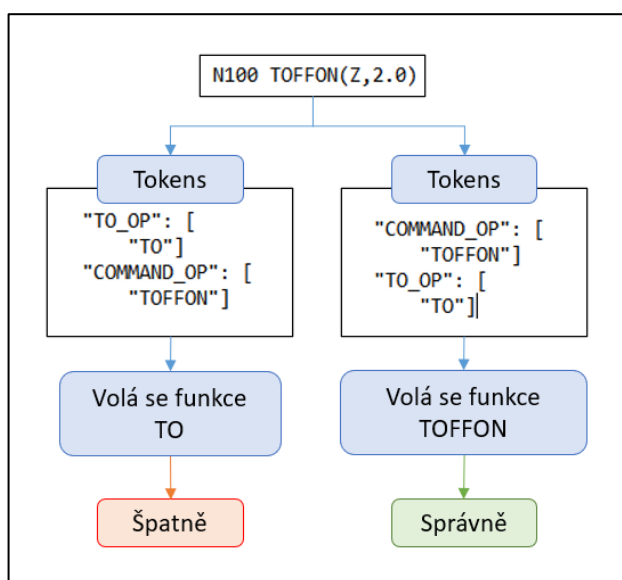


Obr. 28 Zjednodušené zobrazení překladač parserem

Jak již bylo zmíněno, parser je tvořen několika soubory, v rámci bakalářské práce došlo k úpravě pouze některých z nich – jedná se o soubory *Yacc*, *Tokens* a *Lex*. Konkrétní úpravy jsou v této kapitole dále přiblíženy.

### 5.3.1. Knihovny Lex a Tokens

V kapitole 3.2.5 bylo popsáno využití parseru v rámci simulace a jeho rozdělení do dílčích knihoven. Nyní budou přiblíženy funkce jednotlivých knihoven a jejich úpravy. Knihovny *Lex* a *Tokens* obsahují názvy některých funkcí a příkazů, na které se odkazuje v knihovně *Yacc*. Parser při hledání příkazu čte soubor *Tokens*. Pokud narazí na část hledaného výrazu, ihned si vezme první slovo, na které narazí. Například, v NC kódu se vyskytne příkaz „TOFFON“ - parser začne tohle slovo hledat v souboru *Tokens*. První narazí na slovo „TO“, jelikož je v souboru *Tokens* napsáno výše než slovo „TOFFON“. Jelikož parser čte NC kód po částech, myslí si, že našel slovo „TO“, hledá tedy dále slovo „FFON“. Toto je nicméně špatně, proto je třeba slovo „TOFFON“ zadat v souboru *Tokens* dříve než slovo „TO“. Při hledání slova „TOFFON“ pak narazí v souboru *Tokens* nejdříve na slovo „TOFFON“. Pokud se ale bude v NC kódu naopak vyskytovat slovo „TO“, tak parser v souboru *Tokens* nejdřív narazí na slovo „TOFFON“, které mu neodpovídá, dále na slovo „TO“, které již vyhovuje. Tímto je zaručeno, že parser dokáže určit obě slova správně. Na Obr. 29 je stručně znázorněno, jak parser požadovaná slova hledá.



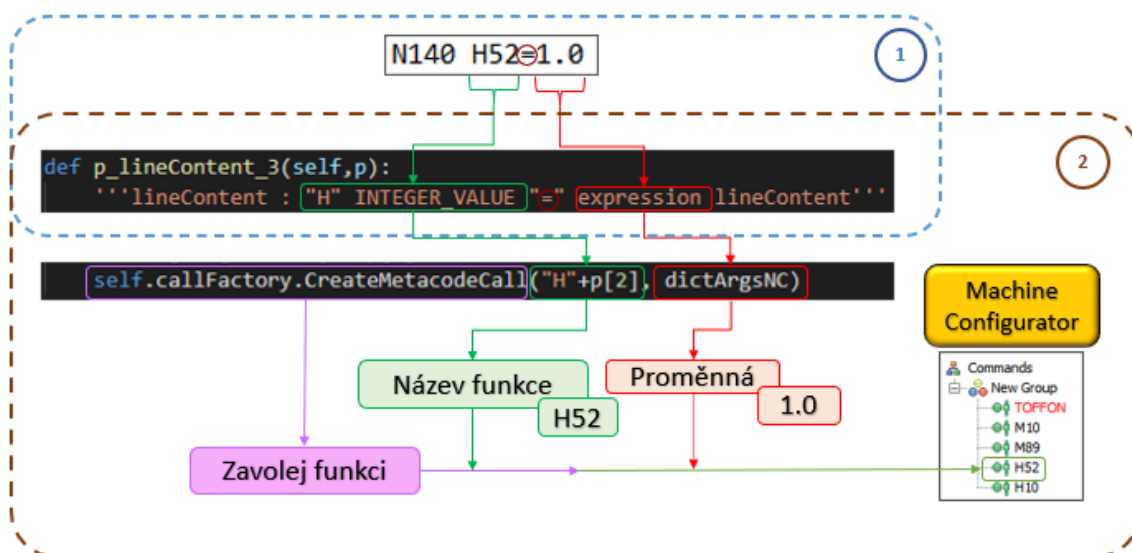
Obr. 29 Zjednodušené znázornění čtení souboru *Tokens* parserem



### 5.3.2. Knihovna Yacc

Zkratka *Yacc* znamená „Yet Another Compiler-Compiler“. Jedná se o knihovnu obsahující definice funkcí a příkazů obsažených v řádku NC kódu. Tento soubor se stará o to, aby byl obsah řádku parserem úspěšně propuštěn, a aby byl příkazům přidělen význam. V této práci byla knihovna *Yacc* rozšířena o dvě funkce – TOFFON a H10. Schéma překlada funkce H10 a její vstup z NC kódu je vidět na Obr. 30.

Parser zpracovává NC kód ve dvou krocích. Pro lepší pochopení je obrázek rozdělen do dvou částí (část 1 – modrá, a část 2 - hnědá).



Obr. 30 Schéma čtení a překlada vstupu z NC kódu parserem

V modré části jsou obsaženy příkazy, které dovolují obsahu NC kódu projít parserem. V druhém řádku prvního černého bloku je vidět, že parser dovoluje výskyt písmene „H“ na řádku. Dále se smí v přesném pořadí vyskytnout celé číslo – „INTEGER\_VALUE“, znak rovnosti – „=“, a proměnná – „expression“. Do prvku „expression“ je možné uložit různé typy proměnných – celé číslo, písmeno, desetinné číslo apod.

V hnědé části je obsahu řádku přiřazen význam. V druhém černém bloku je vidět, že se první řadě volá příkaz „self.callFactory.CreateMetacodeCall“. Tento příkaz volá funkci s definovaným názvem a proměnnou. V případě funkce H52 je název definován jako „‘H‘+p[2]“. Příkaz p[2] značí obsah na pozici 2 – v tomto případě celé číslo INTEGER\_VALUE = 52. Název volané funkce je pak H + 52, tudíž H52. Druhým volaným parametrem je proměnná „dictArgsNC“, do které je uložena proměnná

„expression“. Emulátor pak volá funkci H52 s proměnnou „1.0“. Všechny funkce volané parserem jsou obsaženy v hlavním MCF souboru emulátoru řídicího systému a upravují se v prostředí Machine Configurator.

## 5.4. Tvorba programů pro parser v prostředí Machine Configurator

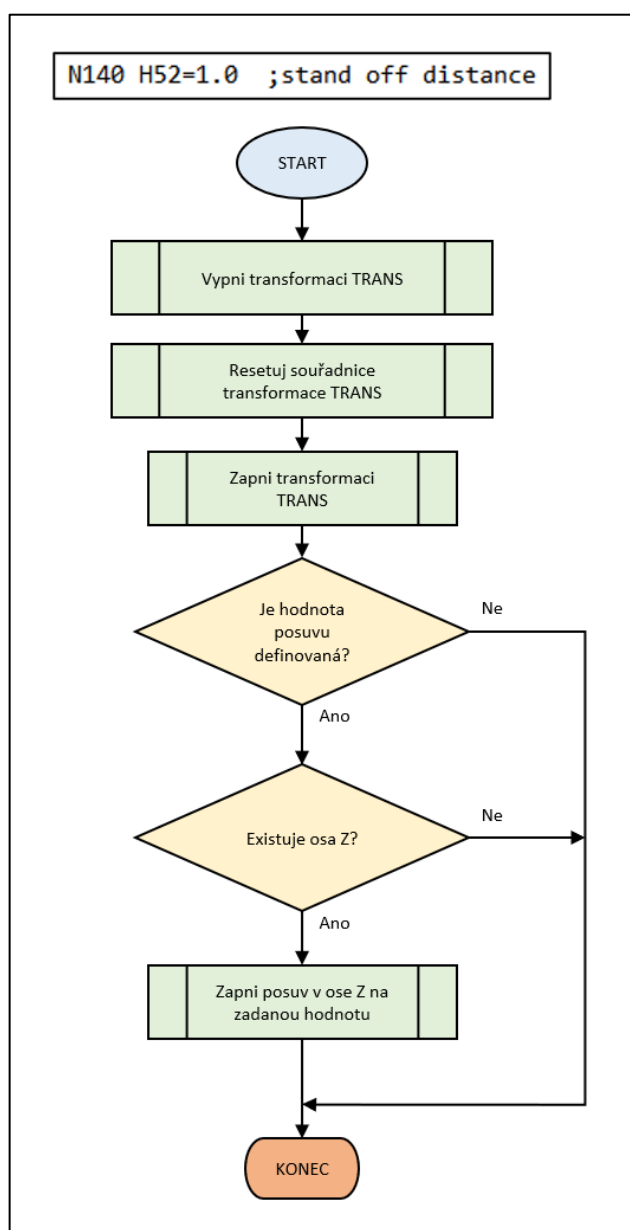
V programu Machine Configurator dochází k úpravě CCF a MCF souborů (viz kapitola 3.2.6). Právě v hlavním MCF souboru (název souboru je KUKA\_KR60\_L30\_2X\_POSITIONER\_DKP400.MCF) jsou obsaženy některé funkce, na které odkazuje parser. V této práci bylo vytvořeno několik funkcí (viz Tab. 5.2). Analýzou generovaného NC kódu proběhl výběr funkcí, které bylo potřebné do parseru přidat. Bylo potřeba přidat nejen veškeré funkce, které parser neznal, ale také rozšířit funkčnost již existujících funkcí. Samotný kód funkce se edituje v programu Machine Configurator a je psán v syntaxi XML programovacího jazyka.

*Tab. 5.2 Přehled funkcí přidaných do emulátoru řídicího systému*

Název funkce	Využití funkce
TOFFON	Nastavení stand-off distance
H10	Ovládání navařovacích procesů
H52	Nastavení stand-off distance
M10	Do-nothing
M89	Vypnutí navařovacích procesů

### 5.4.1. Funkce H52

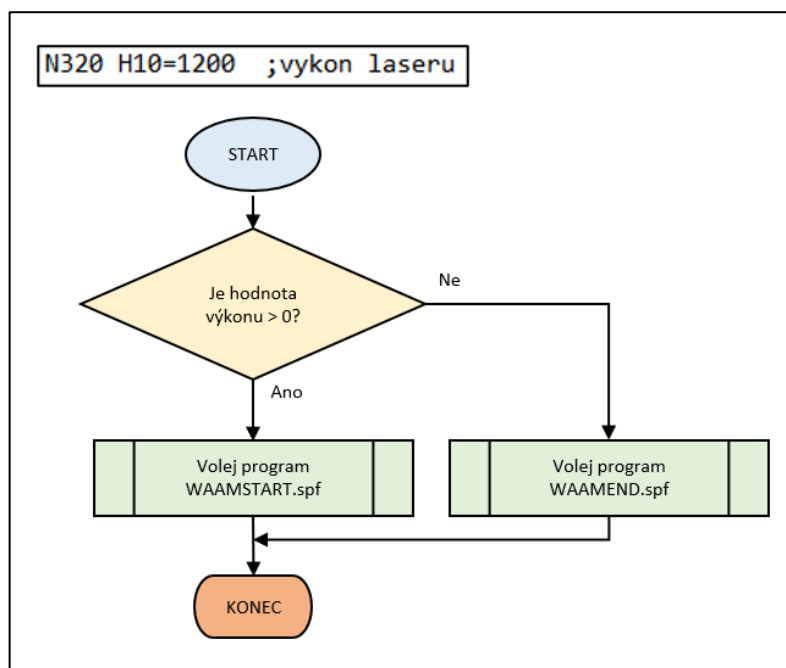
První přidanou funkcí je funkce H52. Úkolem této funkce je nastavení takzvaného stand-off distance, což je vzdálenost špičky nástroje od materiálu. Vzorem pro programování této funkce byla funkce TRANS. Ta se stará o posunutí souřadného systému. Obě funkce plní stejný úkol (fungují analogicky), ale bylo potřeba upravit kód pro práci pouze s osou Z. Také bylo potřeba změnit způsob, jakým pracuje s proměnnou. Schéma funkce H52 je zobrazeno na Obr. 31. Funkce TOFFON funguje stejně jako funkce H52, byla pouze vytvořena pro první verzi postprocesoru.



Obr. 31 Vývojový diagram funkce H52

### 5.4.2. Funkce H10 a M89

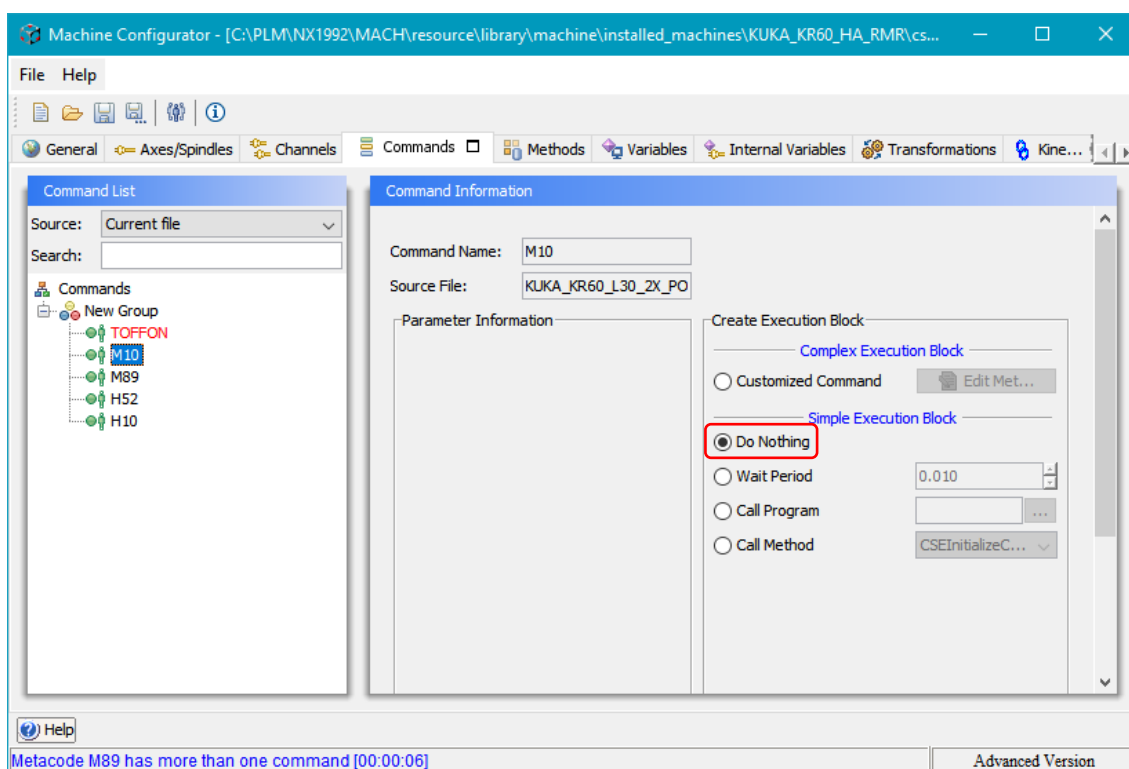
Dalšími funkcemi přidány do Machine Configuratoru byly funkce H10 a M89. Funkce H10 se stará o zapínání a vypínání navařovacích procesů. Při vstupu do funkce kontroluje hodnotu výkonu laseru a volá podprogramy *WAAMSTART.spf* (start navařování), nebo *WAAMEND.spf* (konec navařování). Pokud je požadovaný výkon laseru vyšší než nula, bude zavolán podprogram *WAAMSTART.spf*. Pokud je požadovaný výkon rovný nule, bude zavolán podprogram *WAAMEND.spf*. Funkce M89 se vyskytuje pro případ, že by funkce H10 byla volána pouze pro zapnutí laseru. V takovém případě zajistí zavolání podprogramu *WAAMEND.spf*, čímž ukončí navařování. Diagram funkce H10 je vidět na Obr. 32.



Obr. 32 Vývojový diagram funkce H10

### 5.4.3. Funkce M10

Poslední přidanou funkcí je funkce M10. Ta u reálného stroje slouží k zapnutí laserové emise. Tento úkon v rámci simulace už provádí funkce H10, proto byl funkci M10 v prostředí Machine Configurator přidělen příkaz „Do Nothing“ (viz Obr. 33). Funkce H10 byla zvolená místo funkce M10 kvůli schopnosti volat dva různé podprogramy za použití proměnné. Funkce M10 proměnnou nemá, mohla by tedy volat pouze jeden podprogram, a proto by byla méně užitečná.



Obr. 33 Nastavení příkazu "Do Nothing" v prostředí Machine Configurator

## 5.5. Implementace a použití souborů simulačního modelu

Pro kompletaci simulačního modelu je potřeba do knihoven Siemens NX zanést kinematický model robotické buňky a její emulátor řídicího systému. Do složky na adrese:

*Instalační\_složka\_NX\NX12\MACH\resource\library\machine\installed\_machines\*

je zanesena složka obsahující kinematický model, postprocesor, a emulátor řídicího systému (včetně všech souborů potřebných pro parser). Tato složka také obsahuje veškeré

podprogramy potřebné pro emulátor. Dále je zapotřebí na tento model odkázat v souboru databáze strojů na adrese:

*Instalační\_složka\_NX\NX12\MACH\resource\library\machine\ascii\machine\_database.dat*

Dále jsou do složky na adrese:

*Instalační\_složka\_NX\NX12\MACH\resource\library\device\graphics\*

zavedeny 3D modely výměnných pracovních hlavic. V rámci zanesení těchto modelů do knihovny je potřeba upravit soubor s databází nástrojů na adrese:

*Instalační\_složka\_NX\NX12\MACH\resource\library\device\ascii\device\_database.dat*

V poslední řadě je třeba rozšířit knihovnu nástrojů o nástroje používané u nástrojových hlavic. U reálného stroje tyto nástroje neexistují – pracuje pouze hlavice, nicméně v rámci simulace nástroje potřebné jsou. Toto je řešeno přidáním nástrojů s minimálními rozměry – simulace pak bude vizuálně vypadat jako reálná operace (pozice hlavy nebude od dílce posunutá vlivem délky nástroje). Nástroje musí zároveň mít správná jména, aby simulace fungovala. Kvůli těmto důvodům byly pro každou hlavici vytvořeny vzorové nástroje. To znamená, že nebude potřeba nástroje pokaždé tvořit a přejmenovávat. Musí tedy být doplněna databáze nástrojů na adrese:

*Instalační\_složka\_NX\NX12\MACH\resource\library\tool\installed\_machines\metric\tool\_database.dat*

a také databáze na adrese:

*Instalační\_složka\_NX\NX12\MACH\resource\library\tool\installed\_machines\metric\tool\_database.dat\_bak*

Pro simulaci výroby navrženého dílce se simulační model robotické buňky přidá tlačítkem Retrieve Machine from Library. Ze seznamu se zvolí model pod názvem KUKA KR60 HA ON RAIL z nabídky Robots. Dále je potřeba přidat používané nástrojové hlavice, ty se přidají možností Retrieve Devices from Library. Pořadí nástrojových hlavic v reálné buňce je následující: 1-YC52, 2-SOLID, 3-COAX, 4-WOBBLE. V poslední řadě se do nástrojových hlavic musí přidat správné nástroje. Toto

se provede přes tlačítko Retrieve Tools from Library. Při hledání nástroje se zvolí možnost „Deposition“ a do řádku „Libref“ se napíše název hlavice (např YC52). Po vypsaní vyhovujících výsledků z knihovny se pak zvolí nástroj s názvem YC52\_T. Toto se musí provést pro každou hlavici. Takto přidáný simulační model je schopen plnohodnotné simulace v prostředí Siemens NX.

## **5.6. Ověření funkčnosti simulačního modelu robotické buňky a porovnání s reálným strojem**

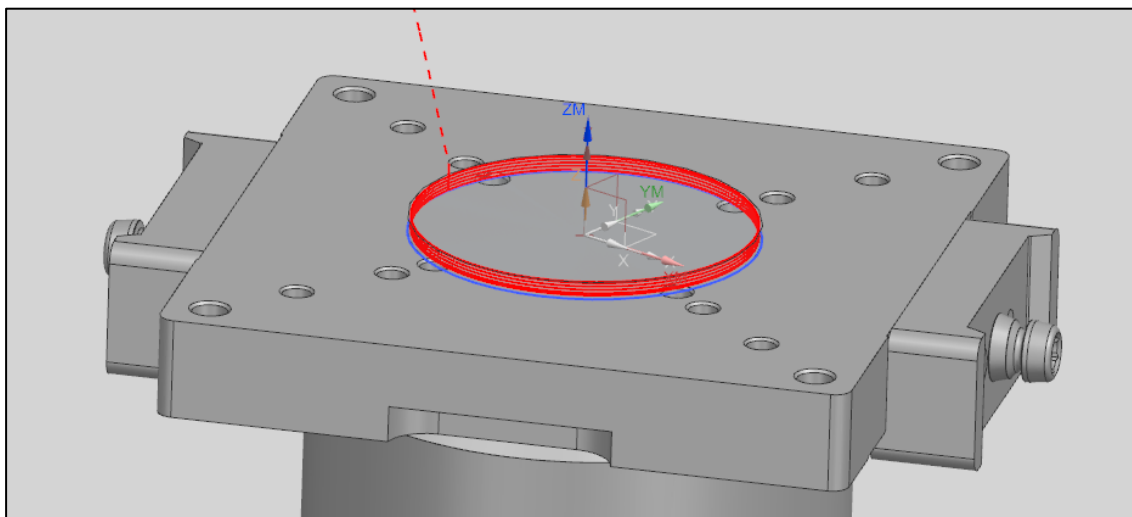
Funkčnost a věrnost emulátoru řídicího systému a kinematického modelu byla provedena testováním samotné simulace, a poté porovnáním s reálným strojem. Pro tento účel byla vytvořena jednoduchá vzorová simulace navařování stěny válce (viz Obr. 34). Při této simulaci byla uplatněna výměna pracovní hlavice, start a konec navařování, i funkce H52 pro posunutí souřadného systému.

Při vývoji a testování emulátoru řídicího systému se vyskytla komplikace s poskytnutým postprocesorem. Bylo zjištěno, že postprocesor nepovoluje spuštění Machine Code Based simulace v Siemens NX. Důvod, proč postprocesor brání spuštění této simulace dosud nebyl zjištěn. Nicméně, pokud by došlo k opravě postprocesoru, tak by nic nemělo bránit spuštění Machine Code Based simulace. Funkčnost všech kódu a odbavení simulace tedy probíhala prostřednictvím External Code Based simulace (simulace na základě externího kódu). Jak již bylo řečeno, tato simulace probíhá na stejné úrovni jako Machine Code Based simulace. Jedním ze dvou zjištěných nedostatků simulace na základě externího kódu je nutnost NC kód nejdříve generovat, a teprve poté ho odbavit. Druhým nedostatkem je to, že v prostředí Siemens NX není možné během simulace na základě externího kódu provádět vizualizaci přídavku materiálu při aditivním navařování. Při spuštění simulace je možná pouze vizualizace drah nástroje, nicméně toto je dostačující pro detekci kolizí, singulárních poloh a přibližné predikci strojního času. Simulace běžící na této úrovni je tedy plně schopná zjištění jakýchkoliv nedostatků při navrhování výrobního procesu.

### **5.6.1. Ověření funkčnosti simulačního modelu**

Simulační model byl otestován na již zmiňovaném dílci, jednalo se o stěnu válce. Pro tento dílec byly připraveny dráhy pro pracovní hlavici YC52 s výměnou na pracovní

hlavici COAX. Na počátku simulace bylo potřeba správnou hlavici upnout, dále celý NC kód musel projít parserem a musel být celý odbaven, tímto byla ověřena funkčnost rozšířeného parseru. Během volání funkcí pro zapínání a vypínání navařování bylo ověřeno, že simulace tyto úkony provádí, tímto tedy byla ověřena funkčnost podprogramů pro navařování. V půlce NC programu došlo k výměně pracovní hlavy z hlavy YC52 na hlavici COAX, tímto byla ověřena funkčnost podprogramu na výměnu pracovních hlavic. Celé testování proběhlo v rámci simulace na základě externího kódu. Emulátor byl schopen celý kód správně a bez problému odbavit. Dráhy nástroje pro výsledný dílec jsou k nahlédnutí na Obr. 34. Celý průběh simulace je vidět v příloze č. 11.



*Obr. 34 Zobrazení dráh nástroje pro simulaci navařování vzorového dílce*

### **5.6.2. Srovnání simulačního modelu s robotickou buňkou**

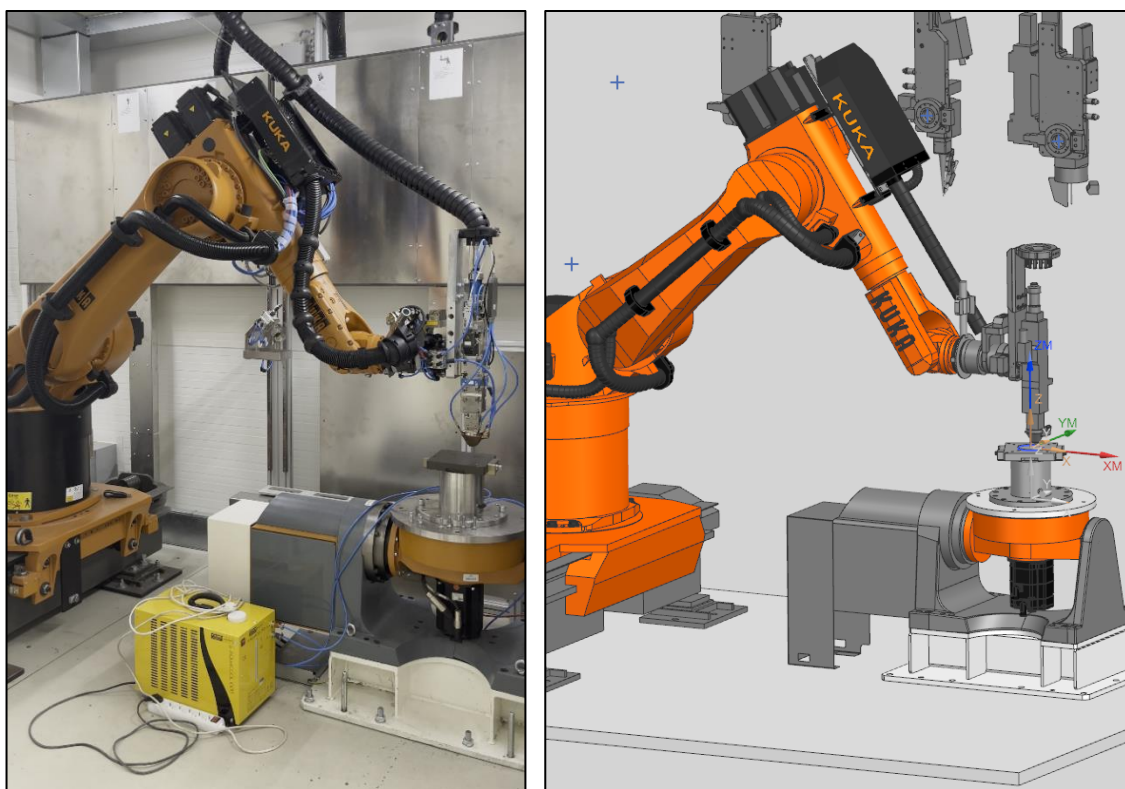
V této fázi testování bylo podstatné ověřit, zda pohyby robotu při výměně pracovních hlavic v simulaci odpovídají pohybům robotu v reálné buňce. Srovnání simulačního modelu s robotickou buňkou proběhlo za pomoci jednoduššího NC kódu. Nebyla zde využita celá výměna nástrojové sestavy, pouze upnutí a odepnutí jedné pracovní hlavy. Jednalo se o hlavici YC52, jelikož při jejím upnutí a odepnutí robot provádí nejsložitější pohyb. Samotný dílec a dráhy nástroje zůstaly stejné.

V rámci testování nebylo možné ověřit strojní čas. Důvodem bylo to, že reálný stroj byl pro testování spuštěn v režimu „Dry run“. V tomto režimu stroj nespouští navařování a pohyby provádí pomaleji. Nicméně při vývoji podprogramů pro simulaci bylo využito podprogramů použitých pro řídicí systém Siemens Sinumerik 840D pro reálnou robotickou buňku. Díky tomu jsou veškeré pohyby v simulaci prováděny při rychlostech,



při kterých probíhají v běžné výrobě za pomoci reálného robotu. To znamená, že by výměna měla probíhat při stejné rychlosti.

Pohyby robotu v simulaci vizuálně odpovídaly pohybům reálného stroje, a to jak při výrobě, tak i při upnutí a odepnutí nástrojové hlavice. Na Obr. 35 je vidět reálný stroj (vlevo) a simulační model (vpravo) při odbavování NC kódu. Ve video přílohách jsou k nahlédnutí video průběhu simulace (Příloha č.11) a videa reálného stroje (Přílohy č.12 až č.15).



*Obr. 35 Porovnání reálného robotu (vlevo) a jeho simulačního modelu (vpravo) při odbavování NC kódu*

## 6. Závěr

Na základě zadání této bakalářské práce byla vypracována rešerše možností číslicového řízení robotů. Zmíněná byla také jejich konstrukce a způsoby využití. Došlo ke zjištění, že se v dnešní době rozšiřuje využití robotů i do oblastí složitějších výrobních procesů jako je obrábění a navařování, nikoliv pouze manipulace. Z toho plyne potřeba roboty řídit počítačově. K počítačové verifikaci těchto složitějších výrobních procesů je však nutný simulační model. Tyto modely jsou tvořeny dílčími částmi. Pro tvorbu těchto částí byla zvolena kombinace programů Siemens NX a Machine Configuratoru. Toto bylo zapříčiněno tím, že pro robot již byl dodán postprocesor vyvinutý pro prostředí Siemens NX. Dále byly rozebrány části řídicího systému Siemens Sinumerik 840D sl. Následovala rešerše možnosti tvorby emulátoru řídicích systému.

Bakalářská práce byla na základě rešerše a zvolené varianty doplněna o praktickou část. V první řadě došlo k úpravě kinematického modelu. U tohoto modelu bylo zapotřebí nastavit správné rozsahy otočných os, jejich rychlosti a zrychlení. Dále byl emulátor řídicího systému přizpůsoben a doplněn o funkční podprogramy. Ty byly vyvinuty na základě dostupných vzorových podprogramů v Siemens NX, a také na základě podprogramů reálného stroje. Dalším krokem bylo upravení a rozšíření parseru (jako části emulátoru řídicího systému) o potřebné funkce. Takto doplněný parser je schopen interpretovat NC kód generovaný stávající verzí postprocesoru. V případě dalšího vývoje postprocesoru je samozřejmě možné tento parser dále doplnit o dodatečné funkce. V poslední řadě došlo k odbavení simulace za použití vzniklého simulačního modelu. K ověření byl navržen jednoduchý válcový dílec a při simulaci bylo využito všech funkcí přidanych do simulačního modelu. Tento stejný NC kód byl také odbaven na reálném stroji a porovnán se simulací. Pohyby při výměně nástroje i samotné odbavení výrobního procesu simulací odpovídaly reálnému stavu. Takto rozšířený simulační model může tedy sloužit k plnohodnotné verifikaci výrobního procesu. Cíle stanovené pro řešení bakalářské práce byly splněné.

## Seznamy

### Seznam literatury

- [1] *Siemens NX Cam Robotics programming* [online]. 2016 [cit. 2021-12-25]. Dostupné z: <https://www.geopl.m.com/knowledge-base-resources/GEOPLM-Siemens-PLM-NX-CAM-Robotics-Programming.pdf>
- [2] KOLÍBAL, Z. *Minulost a budoucnost robotů* [Článek]. Brno: FCC Public s.r.o. 2009, č. AUTOMA 5/2009, [cit. 2021-12-20]. ISSN 1210-9592. Dostupné z: [http://www.automa.cz/Aton/FileRepository/pdf\\_articles/39014.pdf](http://www.automa.cz/Aton/FileRepository/pdf_articles/39014.pdf)
- [3] HepcoMotion. *HepcoAutomation partnership agreement to expand KUKA robot activities in general automation* [online]. [cit. 2022-01-09]. Dostupné z: <https://www.hepcotion.com/blog/latest-news/hepcoautomation-partnership-agreement-to-expand-kuka-robot-activities-in-general-automation/#.YdroZ1ko9EY>
- [4] ANDRLÍK, V a KOL. *Průmyslové roboty a manipulátory*. Praha: ČVUT, Fakulta strojní, Ústav výrobních strojů a zařízení, 2012.
- [5] Robots Done Right. *KUKA KR60-3* [online]. [cit. 2022-01-09]. Dostupné z: <https://robotsdoneright.com/KUKA/KRC2-robots/kuka-kr-60-3.html>
- [6] KRATĚNA, T. *Postprocesor a simulační model pro robota s přidavnými osami*. Praha, 2019. Diplomová práce. České vysoké učení technické v Praze, Fakulta strojní. Vedoucí práce Ing. Vavruška Petr, Ph. D.
- [7] *Pokročilé simulace obrábění v prostředí NX CAM* [online]. 2015 [cit. 2022-01-05]. Dostupné z: <https://www.konstrukter.cz/pokrocile-simulace-obrabeni-v-prostredii-nx-cam/>
- [8] HÁJÍČEK, Z. *Testování virtuálního NC stroje v prostředí PLM systému NX*. Plzeň, 2013. Diplomová práce. Západočeská univerzita v Plzni, Fakulta strojní. Vedoucí práce Ing. Jan Hnátík, Ph. D.
- [9] ZBÍRAL, J. *Simulační model obráběcího stroje v CAM systému Siemens NX s verifikací strojního času*. Praha, 2021. Bakalářská práce. České vysoké učení technické v Praze, Fakulta strojní. Vedoucí práce Ing. Vavruška Petr, Ph. D.
- [10] PLOCEK, J. *Řízení jed noučelového obráběcího stroje*. Brno, 2015. Diplomová práce. Vysoké učení technické v Brně. Vedoucí práce Ing. Pavel Houška, Ph. D.

- [11] SIEMENS. *Sinumerik CNC automation systems: Siemens 840D sl* [online]. [cit. 2021-12-22]. Dostupné z: <https://mall.industry.siemens.com/mall/en/us/Catalog/Products/10233717#>
- [12] HRABAL, J. *Možnost systému Sinumerik 840D při programování obráběcích strojů*. Brno, 2011. Diplomová práce. Vysoké učení technické v Brně. Vedoucí práce Ing. Aleš Polzer, Ph.D.
- [13] *SINUMERIK 840D sl/828D Základy* [online]. 2010 [cit. 2022-01-09]. Dostupné z: <https://support.industry.siemens.com/cs/mdm/28705635?c=15783420171&dl=cs&lc=d e-DE>
- [14] *SINUMERIK, SINAMICS: SINUMERIK 840D sl, SINAMICS S120 Guidelines for machine configuration*. 2009. Dostupné z: [https://cache.industry.siemens.com/dl/files/027/40963027/att\\_63024/v1/MA1\\_10\\_2009\\_en\\_en-US.pdf](https://cache.industry.siemens.com/dl/files/027/40963027/att_63024/v1/MA1_10_2009_en_en-US.pdf)
- [15] *SINUMERIK Run MyRobot /Machining V2.1 for KUKA robots*. 2019. Dostupné z: <https://support.industry.siemens.com/cs/document/109758486/sinumerik-run-myrobot-direct-control-plc?dti=0&lc=en-WW>
- [16] KRATĚNA, T. *Simulace obrábění s modelem obráběcího stroje s výměnnými vřetenovými jednotkami*. Praha, 2016. Bakalářská práce. České vysoké učení technické v Praze, Fakulta strojní. Vedoucí práce Ing. Vavruška Petr, Ph. D.
- [17] ING. ČERNÝ, R. *CNC systémy*. Praha. České vysoké učení technické v Praze, Fakulta strojní.
- [18] NEZBEDA, F. *Deskripce možností číslicového řízení obráběcích strojů*. Brno, 2010. Bakalářská práce. Vysoké učení technické v Brně. Vedoucí práce Doc. Ing. Petr Blecha, Ph.D.
- [19] Kovosvit MAS. *MCV 1000* [online]. [cit. 2022-01-09]. Dostupné z: <https://www.kovosvit.cz/mcv-1000/>
- [20] *Circular Interpolation* [online]. [cit. 2022-01-10]. Dostupné z: [http://moldviet.net/Tai-Lieu/yougo2/nc\\_arch.php](http://moldviet.net/Tai-Lieu/yougo2/nc_arch.php)
- [21] HYPER THERM, I. Robotmaster. *About Robotmaster* [online]. 2019 [cit. 2022-01-18]. Dostupné z: <https://www.robotmaster.com/en/about>

## Seznam obrázků

Obr. 1 Vertikální obráběcí centrum MCV 1000 [19] .....	12
Obr. 2 Robot KUKA KR60-3 [3] .....	12
Obr. 3 Robotická linka pro bodové svařování karoserií automobilů [2] .....	13
Obr. 4 Vývoj programování robotů [21].....	14
Obr. 5 Vizualizace drah obrábění v prostředí Siemens NX CAM.....	15
Obr. 6 Dialogová okna Robot Control (vlevo) a Edit Rules (Vpravo).....	16
Obr. 7 Aplikace pravidla pro nastavení orientace nástrojové hlavy robotu.....	17
Obr. 8 Aplikace pravidla pro nastavení konfigurace robotu.....	17
Obr. 9 Vizuální indikace uplatnění robotických pravidel.....	18
Obr. 10 Schéma simulace .....	18
Obr. 11 Schéma komponentů potřebných k CSE simulaci.....	20
Obr. 12 CAD model robotické buňky v prostředí Siemens NX CAM.....	22
Obr. 13 K-komponenty uspořádané do stromu v prostředí Siemens NX CAM.....	23
Obr. 14 Schéma umístění souborů pro Siemens NX .....	25
Obr. 15 Kruhová interpolace [20].....	26
Obr. 16 HMI panel řídicího systému Sinumerik 840D sl s jeho PLC částí a spolu s pohony Sinamics S120 [20].....	29
Obr. 17 Schéma dílčích částí emulátoru řídicího systému.....	32
Obr. 18 Machine Configurator.....	33
Obr. 19 Kód podprogramu pro výměnu pracovní hlavy .....	34
Obr. 20 NC kód generovaný dodaným postprocesorem.....	36
Obr. 21 Pracovní hlavy robotické buňky .....	37
Obr. 22 Nastavení rozsahu os v Siemens NX.....	40
Obr. 23 Část kódu databáze robotic_to_ini_main_bunka.ini .....	42
Obr. 24 Pracovní diagram podprogramu ToolChange.spf (vlevo) a část kódu volající další podprogramy výměny nástrojové sestavy (vpravo).....	43
Obr. 25 Vývojový diagram podprogramu MOUNT_NEXT_HEAD.spf .....	44
Obr. 26 Podprogram MOUNT_NEXT_TOOL.spf starající se o výměnu nástroje .....	45
Obr. 27 Vývojový diagram podprogramu WAAMSTART.spf.....	46
Obr. 28 Zjednodušené zobrazení překladu parserem.....	47
Obr. 29 Zjednodušené znázornění čtení souboru Tokens parserem .....	48

Obr. 30 Schéma čtení a překladu vstupu z NC kódu parserem .....	49
Obr. 31 Vývojový diagram funkce H52 .....	51
Obr. 32 Vývojový diagram funkce H10 .....	52
Obr. 33 Nastavení příkazu "Do Nothing" v prostředí Machine Configurator .....	53
Obr. 34 Zobrazení drah nástroje pro simulaci navařování vzorového dílce .....	56
Obr. 35 Porovnání reálného robotu (vlevo) a jeho simulačního modelu (vpravo) při odbavování NC kódu .....	57

## Seznam tabulek

Tab. 3.1 Přehled proměnných v Siemens NX Machine Tool Builder [9] .....	24
Tab. 3.2 Přehled M-funkcí řídicího systémů Siemens Sinumerik 840D sl/828D [13]... ..	30
Tab. 3.3 Označení pracovních hlavic v kódu.....	35
Tab. 3.4 Funkce generované postprocesorem, jejich využití a příslušné podprogramy .	36
Tab. 5.1 Seznam podprogramů a knihoven emulátoru řídicího systému pro robotickou buňku .....	41
Tab. 5.2 Přehled funkcí přidanych do emulátoru řídicího systému .....	50

## Seznam použitého software

1. Siemens NX 12
2. Machine Configurator
3. Microsoft Visual Studio
4. Notepad
5. XML Notepad

## Seznam příloh

### *Textové přílohy:*

Příloha č. 1 : Vývojový diagram podprogramu ToolChange.spf.....	P2
Příloha č. 2 : Vývojový diagram podprogramu MOUNT_NEXT_HEAD.spf.....	P3
Příloha č. 3 : Vývojový diagram podprogramu UNMOUNT_CURRENT_HEAD.spf .	P4
Příloha č. 4 : Vývojový diagram podprogramu MOUNT_NEXT_TOOL.spf .....	P5
Příloha č. 5 : Vývojový diagram podprogramu UNMOUNT_CURRENT_TOOL.spf..	P6
Příloha č. 6 : Vývojový diagram podprogramu WAAMSTART.spf.....	P7

Příloha č. 7 : Vývojový diagram podprogramu WAAMEND.spf .....	P8
Příloha č. 8 : Vývojový diagram funkce H52 .....	P9
Příloha č. 9 : Vývojový diagram funkce H10 .....	P10
Příloha č. 10 : Vývojový diagram funkce M89.....	P11

***Elektronické přílohy:***

Příloha č. 11: Video testovacího NC programu pro ověření simulačního modelu

Příloha č. 12: Video upnutí pracovní hlavice YC52 robotickou buňkou

Příloha č. 13: Video odepnutí pracovní hlavice YC52 robotickou buňkou

Příloha č. 14: Video odbavení vzorového NC kódu robotickou buňkou

Příloha č. 15: NC kód pro testování simulačního modelu

Příloha č. 16: NC kód pro robotickou buňku

Příloha č. 17: Testovací dílec

***Přílohy na CD:***

Příloha č. 11: Video testovacího NC programu pro ověření simulačního modelu

Příloha č. 12: Video upnutí pracovní hlavice YC52 robotickou buňkou

Příloha č. 13: Video odepnutí pracovní hlavice YC52 robotickou buňkou

Příloha č. 14: Video odbavení vzorového NC kódu robotickou buňkou

Příloha č. 15: NC kód pro testování simulačního modelu

Příloha č. 16: NC kód pro robotickou buňku

Příloha č. 17: Testovací dílec

Příloha č. 18: PDF bakalářské práce a textových příloh

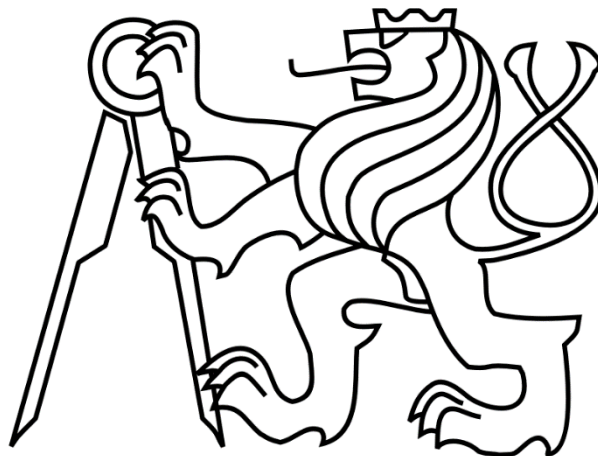
Příloha č. 19: Simulacni\_model.zip

Příloha č. 20: NC\_kody\_hlavic.zip

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STROJNÍ

Ústav výrobních strojů a zařízení

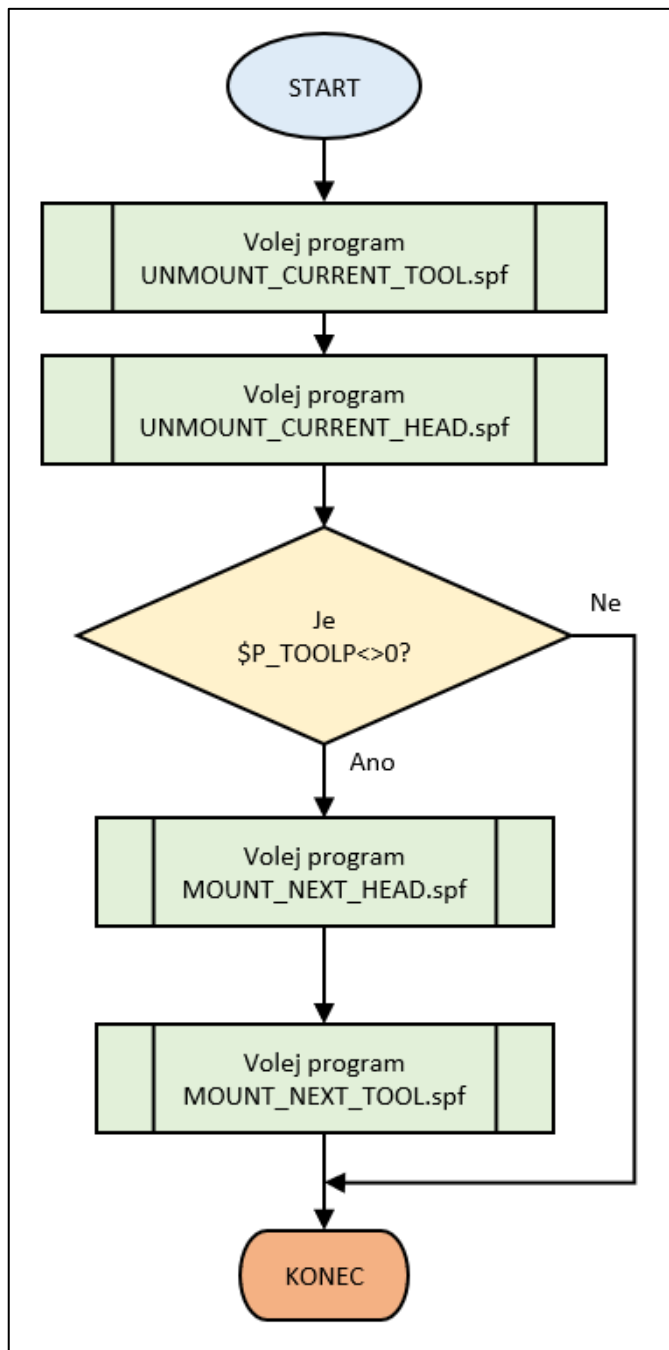


# Bakalářská práce

Textové přílohy



## Příloha č. 1 : Vývojový diagram podprogramu ToolChange.spf



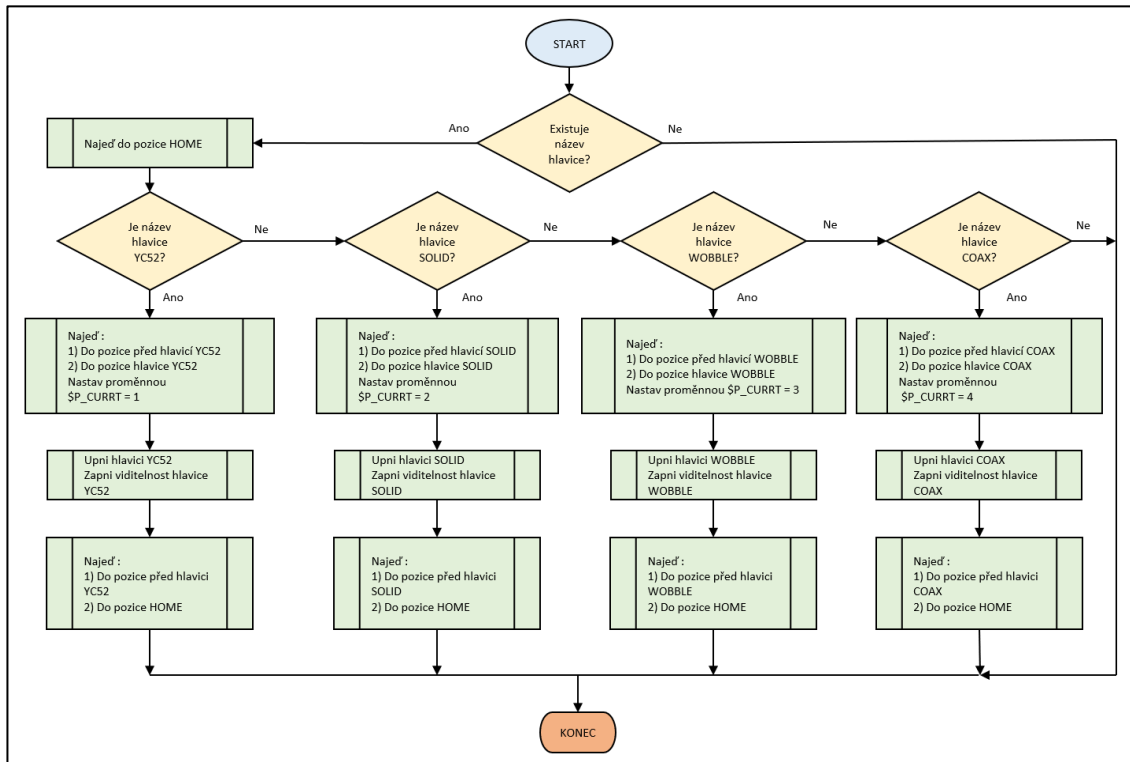
### Popis

Tento diagram znázorňuje průběh funkce ToolChange.spf starající se o celkovou výměnu nástrojové sestavy (nástroj + hlavice). Jedná se pouze o funkci, která postupně volá další podprogramy.

### Vysvětlivka proměnných v diagramu

\$P\_TOOLP – proměnná obsahující číslo volaného nástroje

## Příloha č. 2 : Vývojový diagram podprogramu MOUNT\_NEXT\_HEAD.spf



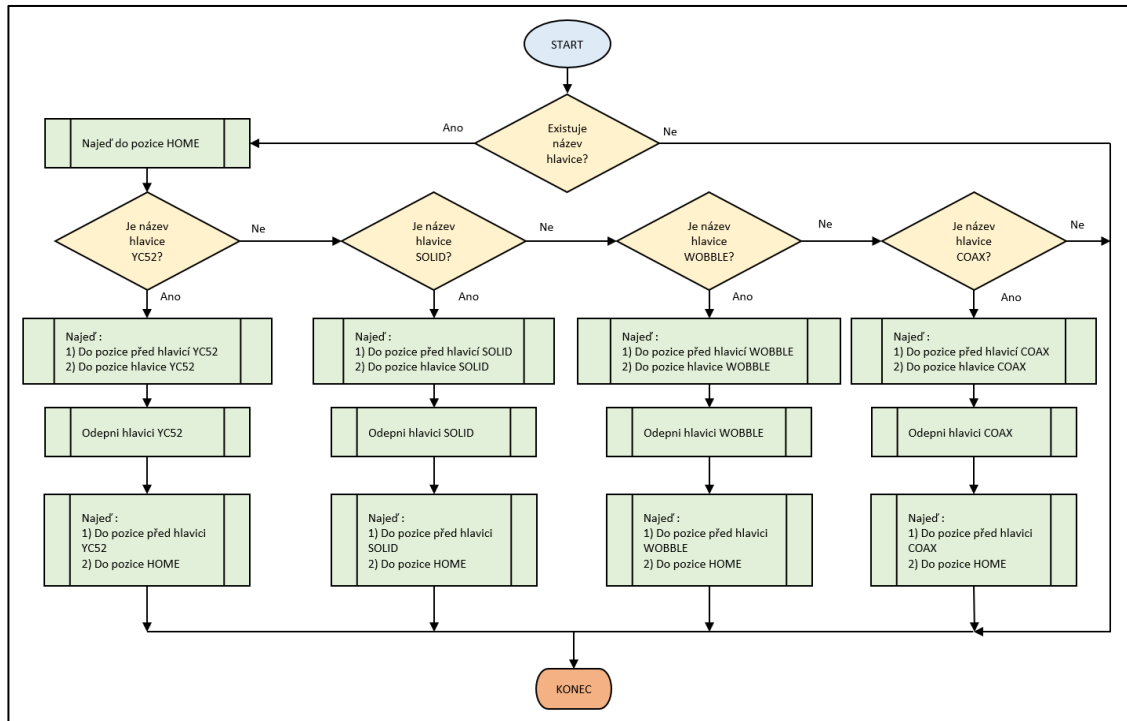
### Popis

Tento diagram znázorňuje průběh podprogramu MOUNT\_NEXT\_HEAD.spf, který se stará o upnutí volané nástrojové hlavice. Při vstupu do podprogramu se zkontroluje, zda existuje název volané hlavice. Poté robot najede do pozice HOME, která je definována souřadnicemi v samotném podprogramu. Dále se porovnává název volané hlavice se seznamem hlav v kódu podprogramu. Po shodě názvů robot najede do správné pozice. Dojde k upnutí hlavice a nastavení proměnné \$P\_CURRT. Na konci znovu dojde k najetí do pozice před hlavicí a poté do pozice HOME.

### Vysvětlivka proměnných v diagramu

\$P\_CURRT – proměnná obsahující číslo právě upnutého nástroje, s touto proměnnou pracuje podprogram UNMOUNT\_CURRENT\_HEAD.spf a podprogram UNMOUNT\_CURRENT\_TOOL.

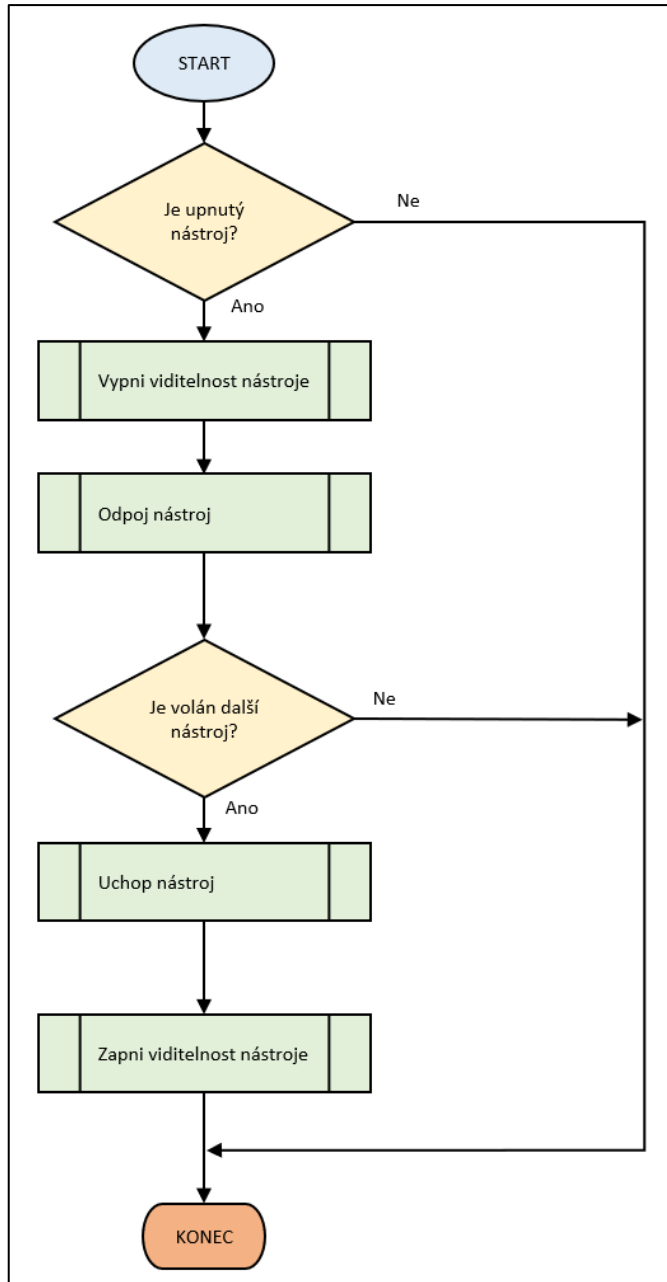
### Příloha č. 3 : Vývojový diagram podprogramu UNMOUNT\_CURRENT\_HEAD.spf



#### Popis

Tento diagram znázorňuje průběh podprogramu UNMOUNT\_CURRENT\_HEAD.spf, který se stará o upnutí volané nástrojové hlavice. Při vstupu do podprogramu se zkontroluje, zda existuje název volané hlavice. Podprogram se v této fázi dívá na název volané hlavice, kterou se NC kód snaží upnout, nikoliv na název aktuálně upnuté hlavice. Důvod je ten, že se do tohoto podprogramu vstupuje pouze při upínání nové hlavice, kdy je podstatné zjistit, zda má smysl do programu vstupovat – tj. zda se NC kód snaží upnout hlavu, která v simulačním modelu existuje. Poté robot najede do pozice HOME, která je definována souřadnicemi v samotném podprogramu. Dále se porovnává název volané hlavice se seznamem hlav v kódu podprogramu. Po shodě se najede do správné pozice. Dojde k odepnutí hlavice. Na konci znovu dojde k najetí do pozice před hlavicí a poté do pozice HOME.

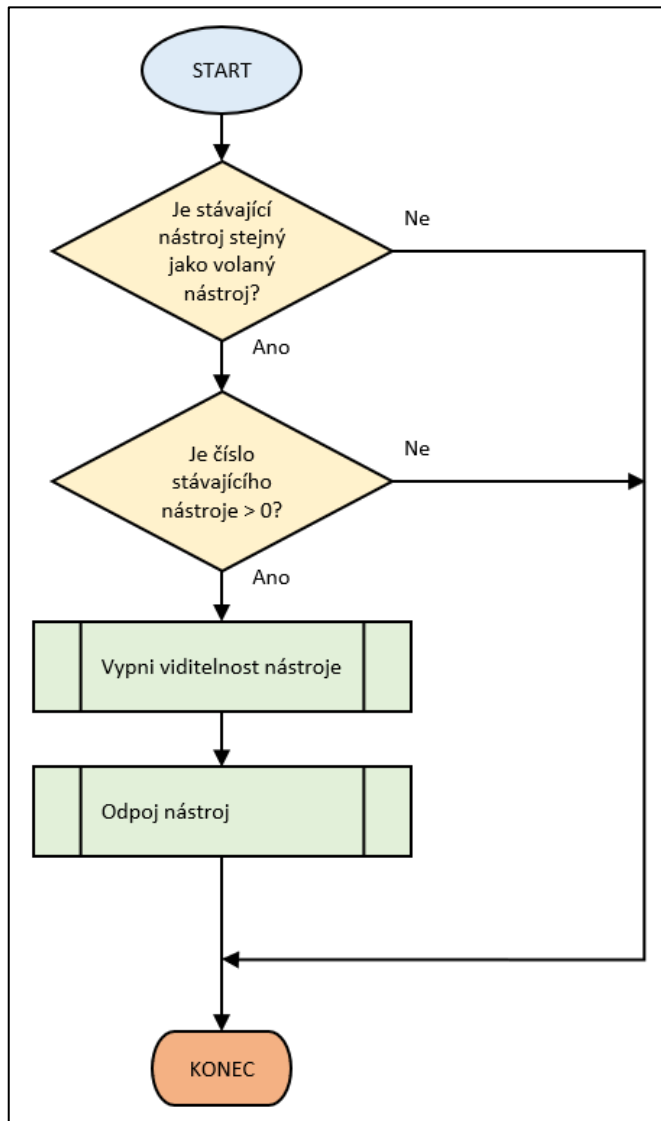
## Příloha č. 4 : Vývojový diagram podprogramu MOUNT\_NEXT\_TOOL.spf



### Popis

Tento diagram znázorňuje průběh podprogramu MOUNT\_NEXT\_TOOL.spf. Program v počátku nejdříve zkontroluje, zda je upnutý nějaký nástroj. Pokud ano, tento nástroj nejdříve odepne, pak se podívá, zda je volán další nástroj. Pokud ano, dojde k upnutí tohoto nástroje, a poté se zapne jeho viditelnost. Při tomto podprogramu nedochází k žádným pohybům, není jich zde potřeba.

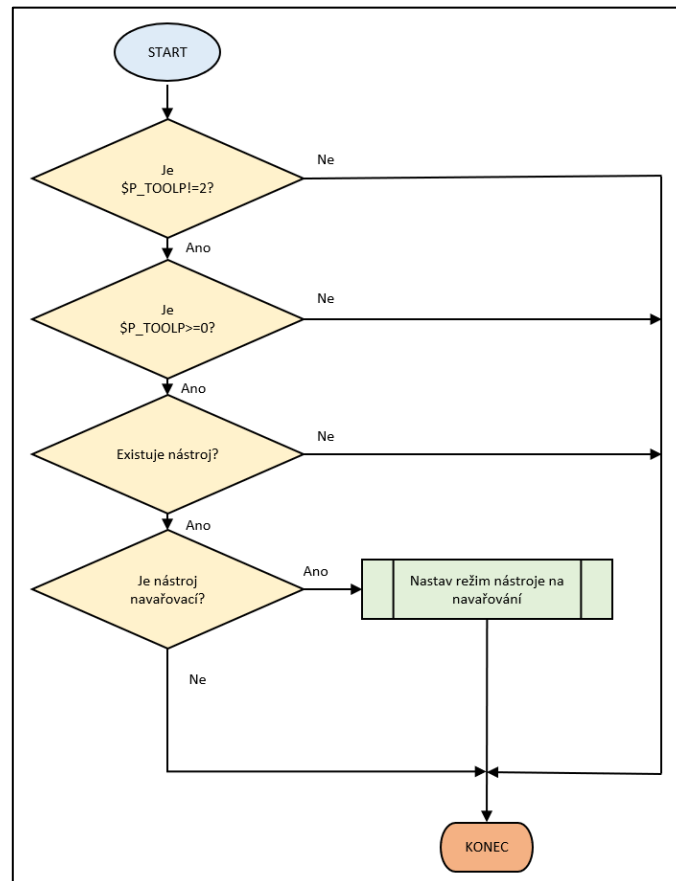
## Příloha č. 5 : Vývojový diagram podprogramu UNMOUNT\_CURRENT\_TOOL.spf



### Popis

Tento diagram znázorňuje průběh podprogramu UNMOUNT\_CURRENT\_TOOL.spf. Nejdříve se kontroluje, zda se NC kód pokouší upnout nástroj, který je již upnutý. Pokud tomu tak není, kontroluje zda je číslo upnutého nástroje větší než nula. Pokud ano, u tohoto nástroje vypne jeho viditelnost a nástroj poté odepne.

## Příloha č. 6 : Vývojový diagram podprogramu WAAMSTART.spf



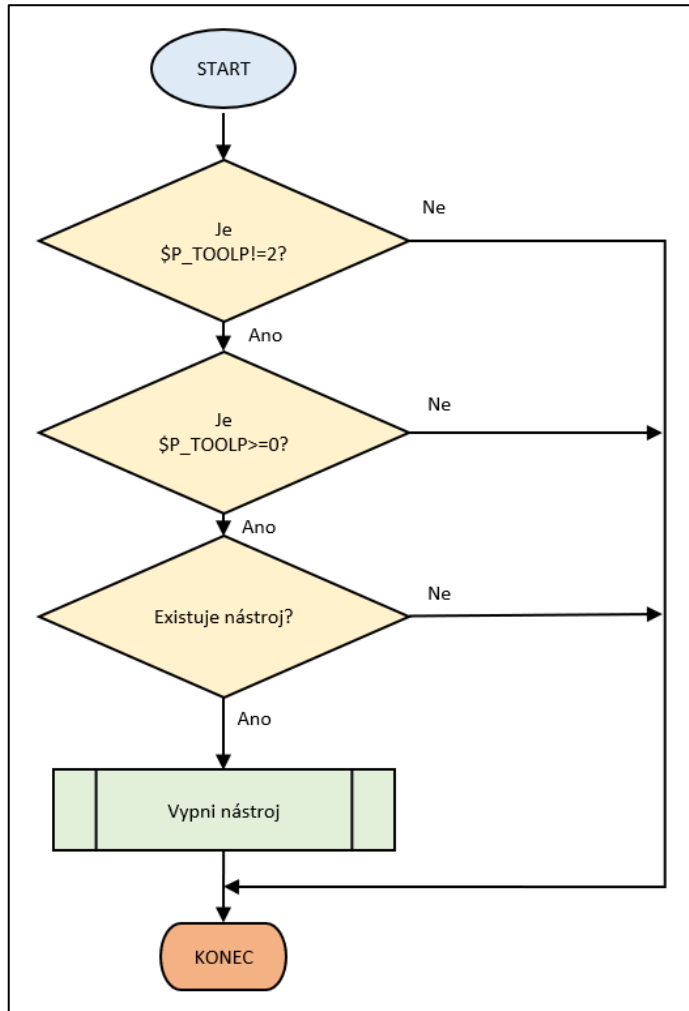
### Popis

Tento diagram znázorňuje průběh podprogramu WAAMSTART.spf. V prvním kroku kontroluje zda se číslo volaného nástroje rovná 2. Pokud ano, znamená to, že volaný nástroj je hlavice SOLID. Toto znamená, že zbytek programu bude přeskočen. Poté se kontroluje, zda je číslo volaného nástroje větší než nula, pak zda volaný nástroj existuje. Nakonec se zjistí, zda je volaný nástroj navařovací a poté se zapne navařovací režim nástroje. Tímto je navařování zahájeno.

### Vysvětlivka proměnných v diagramu

\$P\_TOOLP – proměnná obsahující číslo volaného nástroje

## Příloha č. 7 : Vývojový diagram podprogramu WAAMEND.spf

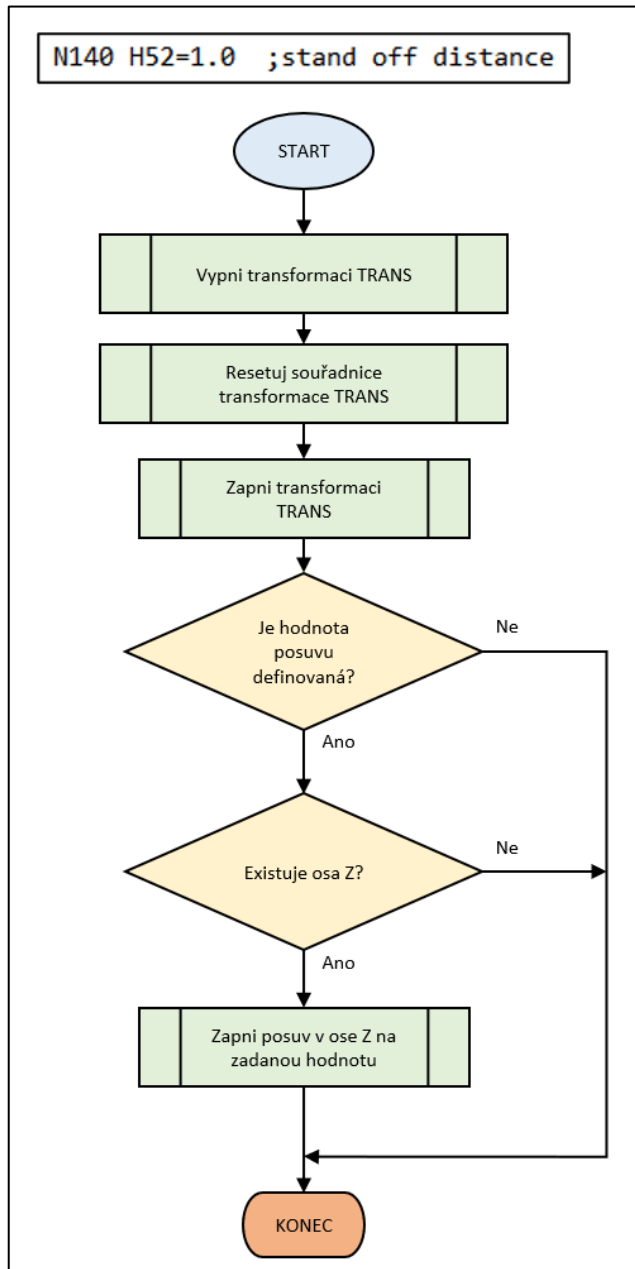


### Popis

Tento diagram znázorňuje průběh podprogramu WAAMEND.spf. V prvním kroku kontroluje zda se číslo volaného nástroje rovná 2. Pokud ano, znamená to, že volaný nástroj je hlavice SOLID. Toto znamená, že zbytek programu bude přeskočen. Poté se kontroluje, zda volaný nástroj existuje. V posledním kroku pak dojde k vypnutí navařovací funkce nástroje. Tímto je navařování ukončeno.



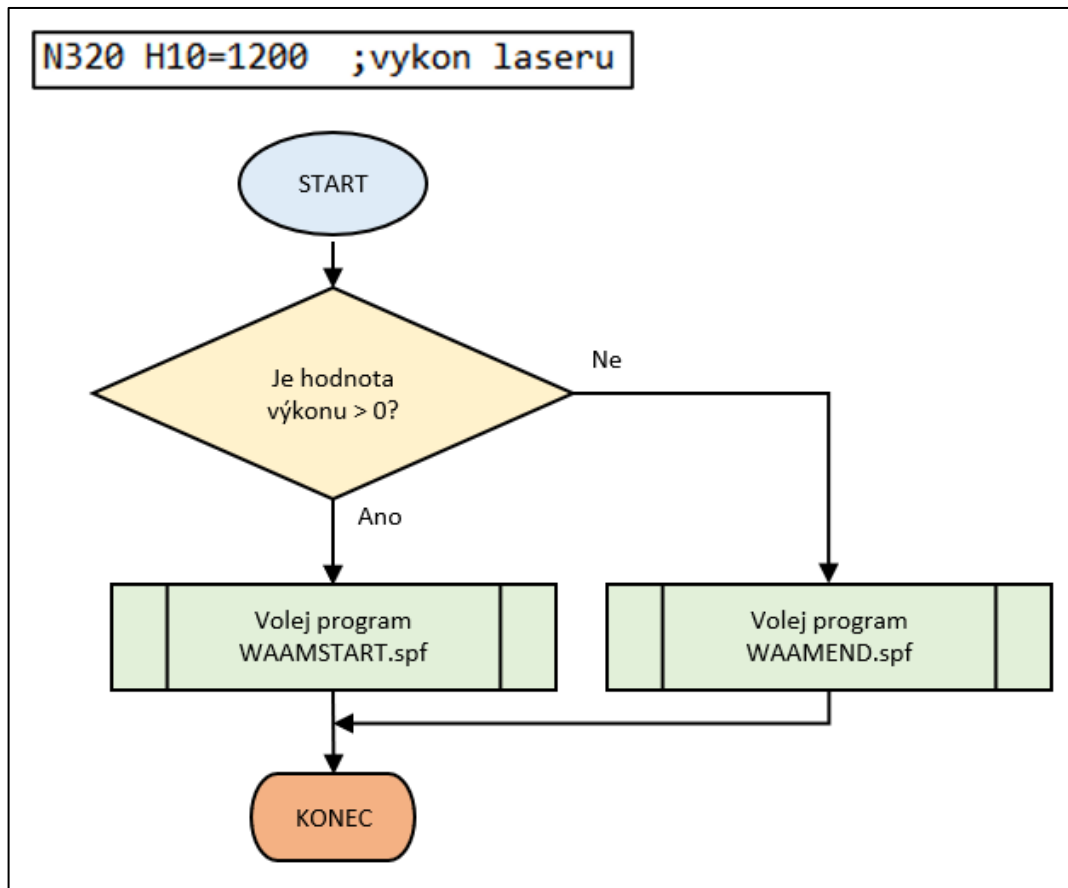
## Příloha č. 8 : Vývojový diagram funkce H52



### Popis

Tento diagram znázorňuje průběh funkce H52 a její volání z NC kódu. Funkce nejdříve vypíná a resetuje hodnoty transformace TRANS souřadného systému. Toto zajistí, že se transformace pokaždé zapíná od nulové souřadnice. Poté funkce opět transformaci TRANS zapne. Dále se podívá zda je definovaná nějaká hodnota posuvu (viz ve vstupu z NC kódu hodnota 1.0). Potom kontroluje zda existuje osa Z, pokud ano, zapíná posuv v ose Z na zadanou hodnotu.

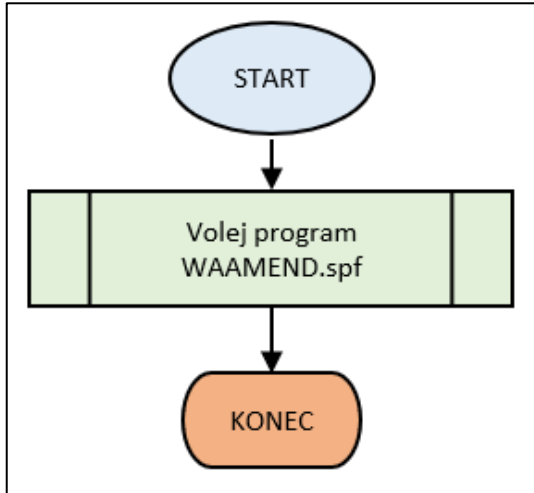
## Příloha č. 9 : Vývojový diagram funkce H10



### Popis

Tento diagram znázorňuje průběh funkce H10. Tato funkce se stará o volání podprogramů pro zapínání a vypínání navařovacích procesů. Funkce zkontroluje je, zda je hodnota výkonu větší než nula. Pokud ano, zavolá podprogram WAAMSTART.spf. V opačném případě volá podprogram WAAMEND.spf.

## Příloha č. 10 : Vývojový diagram funkce M89



### Popis

Tento diagram znázorňuje průběh funkce M89. Tato funkce pouze volá podprogram WAAMEND.spf. Vyskytuje se pouze pro případ, kdy se v NC kódu funkce H10 vyskytuje jen pro zapnutí navařovacího procesu.