



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV VÝROBNÍCH STROJŮ, SYSTÉMŮ A
ROBOTIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF PRODUCTION MACHINES, SYSTEMS AND
ROBOTICS

PROGRAMOVÁNÍ PRŮMYSLOVÝCH ROBOTŮ KUKA

PROGRAMMING OF KUKA INDUSTRIAL ROBOTS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

RADIM DVOŘÁK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ALEŠ POCHYLÝ

BRNO 2015

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav výrobních strojů, systémů a robotiky

Akademický rok: 2014/15

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): Radim Dvořák

který/která studuje v **bakalářském studijním programu**

obor: **Základy strojního inženýrství (2341R006)**

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

Programování průmyslových robotů KUKA

v anglickém jazyce:

Programming of KUKA industrial robots

Stručná charakteristika problematiky úkolu:

Analyzujte aktuální možnosti programování průmyslových robotů KUKA. Zaměřte se na metody programování online, offline, případně i na problematiku pro CAD/CAM aplikace. V druhé části práce vypracujte program pro manipulační aplikaci s robotem KUKA dle zadání.

Cíle bakalářské práce:

1. Analyzujte aktuální možnosti pro programování průmyslových robotů (online, offline, CAD/CAM aplikace).
2. Vypracujte obecný algoritmus řešení manipulační úlohy s robotem KUKA dle zadání a poté vytvořte pro danou úlohu funkční demonstrační aplikaci.

Seznam odborné literatury:

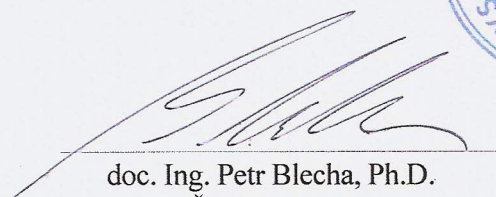
1. PIRES, J. N. Industrial Robots Programming: Building Applications for the Factories of the Future. Springer, 2007. 282 s. ISBN 978-0-387-23325-3
2. WOLF, A., STEINMANN, R. SCHUNK, H. Grippers in Motion: The Fascination of Automated Handling Tasks. Springer, 2005. 242 s. ISBN 978-3-540-27718-7
3. KUKA - Operating and Programming Instructions - For End Users, KSS V5.2, V5.3, V5.4, 2006
4. KUKA KR C2 / KR C3 - Expert programming, 2006

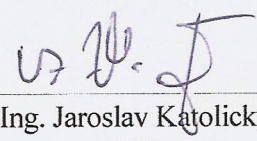
Vedoucí bakalářské práce: Ing. Aleš Pochylý

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2014/15.

V Brně, dne 18.11.2014




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


doc. Ing. Jaroslav Katolický, Ph.D.
Děkan

Abstrakt

Tato bakalářská práce pojednává o analýze aktuálních možností programování průmyslových robotů. Postupně se zabývá průmyslovými roboty používanými k manipulaci, jednotlivými druhy programování, jejich vlastnostmi, porovnáním a seznamuje s novými trendy v oblasti on-line, off-line a CAD/CAM programování. V závěrečné části této práce je řešena demonstrační aplikace pomocí podtlakové úchopné hlavice, jejímž výsledkem je obecný algoritmus a manipulační program pro robot KUKA.

Klíčová slova

průmyslový robot, programování, manipulace, algoritmus, úchopná hlavice

Abstract

This bachelor's thesis discusses with analysis of current options of industrial robots programming. It deals with manipulating industrial robots, their different programming types, properties, comparison and introduces with new trends in the field of on-line, off-line and CAD/CAM programming. Demonstration gripping task with vacuum gripper, whose resulting in a general algorithm and manipulation program for KUKA robot, is solved in final part of this thesis.

Key words

industrial robot, programming, manipulation, algorithm, gripper

Bibliografická citace

DVOŘÁK, R. *Programování průmyslových robotů KUKA*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2015. 42 s. 5 příloh. Vedoucí bakalářské práce Ing. Aleš Pochylý.

Prohlášení

Prohlašuji, že jsem bakalářskou práci na téma programování průmyslových robotů KUKA vypracoval samostatně, pouze s použitím uvedených pramenů, publikací a na základě konzultací s vedoucím bakalářské práce.

V Brně dne 27. 05. 2015


.....
Radim Dvořák

Poděkování

Tímto bych rád poděkoval mému vedoucímu Ing. Aleši Pochylému za odborné vedení, rady, informace a cenné připomínky, které mi byly poskytnuty při vypracování této bakalářské práce.

Obsah

Úvod	10
1 Robotika a robotizace	11
2 Průmyslové roboty a manipulátory	12
2.1 Definice průmyslových robotů	12
2.2 Struktura a kinematická konstrukce	12
2.3 Nejpoužívanější druhy robotů	13
2.4 Souřadnicové systémy	14
2.5 Řízení pohybu.....	15
3 Programování robotů	16
3.1 On-line programování.....	16
3.1.1 Uživatelská rozhraní	17
3.1.2 Programování teach-pendanty	18
3.1.3 Kooperace robot/člověk.....	19
3.2 Off-line programování.....	21
3.2.1 Obecná metodika	22
3.2.2 Virtuální kontrolér	23
3.2.3 Simulační softwary	24
3.3 CAD/CAM programování	28
3.3.1 Metodika CAD/CAM	28
3.3.2 Softwarové systémy	29
4 Manipulační úloha	32
4.1 Zadání a popis úlohy.....	32
4.2 Robotizované pracoviště.....	33
4.3 Proměnné, vstupy a výstupy	33
4.4 Procesy a podprogramy	35
4.5 Tvorba a funkce řídicího programu	36
Závěr	37
Seznam použitých zdrojů	38
Seznam obrázků	41
Seznam grafů a tabulek	42
Seznam příloh	42

	Ústav výrobních strojů, systémů a robotiky	Str. 10
	BAKALÁŘSKÁ PRÁCE	

Úvod

Roboty se v posledních letech již tradičně používají pro práci v průmyslovém prostředí. Představují nejflexibilnější stávající automatizační technologii a oproti původní myšlence, číslicově řízenému univerzálnímu manipulátoru, ušly už dlouhou cestu [19]. Vzhledem k současným potřebám, kdy se neustále zvyšují požadavky na kvalitu a kvantitu, stále rostou nároky kladené na robotizaci spolu s automatizací. Avšak díky nim jsou ustavičně modernizována i vyvíjena nová automatizovaná a robotická zařízení [3, 32].

Na vývoj a budoucí tendenci průmyslového robotu se ale vyskytují, hlavně v řadách odborníků, odlišné i mnohdy si odporující představy, jenž se odráží například v jeho nejasné definici [32]. Roboty jsou však stále více autonomní, s větší mírou uživatelských úprav a možností konfigurace pro různé využití napříč všemi průmyslovými odvětvími [19]. A v tomhle směru je klíčovým faktorem právě ono programování robotů, jež slouží k budování různých aplikací a dalo vzniknout i téhle práci, která se jím zabývá.

Úvod je věnován základním pojmům, kde proběhne seznámení s robotikou a její ekonomickou perspektivou (kap. 1). Současně, jako podklad pro závěrečnou praktickou část, také nastíní strukturu, jednotlivé typy pohybů i souřadných systémů (kap. 2).

Dále se práce dostává k jednomu ze dvou hlavních cílů, kterým bylo analyzovat aktuální možnosti v oblasti programování průmyslových robotů. Postupně pojednává o principech, rozdílech i uplatněních jednotlivých metod (kap. 3). Menší, ale neméně důležitá část, zařazená pod on-line programování, je pak zaměřena na do budoucna velmi perspektivní přímou spolupráci mezi člověkem a robotem.

Následně bylo druhým cílem vypracování obecného algoritmu a dané demonstrační aplikace dle zadání. Řešení této části, zpracované v závěrečné kapitole, dále obsahuje doplňující materiál dodaný v přílohách (kap. 4).



1 Robotika a robotizace

Robotikou se rozumí obor pokrývající speciální oblast techniky. Je to věda zabývající se studiem robotů, jejich konstrukcí, designem a aplikacemi. Jako doplňující definici pak lze vzít výrok jednoho amerického profesora M.Bradyho, který řekl: „*Robotika je inteligentním spojením mezi vnímáním a činností.*“ Jedná se tedy o široký pojem, jenž čerpá z mnoha vědních disciplín [29]. Proto ji lze často členit dle různých kritérií, kdy rozdělení na robotiku průmyslovou a experimentální bylo pro účely práce nevhodnější. Průmyslová robotika se tedy zabývá konstrukcí i použitím průmyslových robotů a dala vzniknout hned dvěma důležitým pojmům, které spolu jdou ruku v ruce – automatizace a robotizace [27, 29].



Obr. 1.1 Příklad robotizovaného pracoviště [13]

Automatizaci lze definovat jako období technického vývoje, které je charakteristické používáním řídicích systémů k řízení výrobních procesů a zařízení bez přímého zásahu člověka [3]. Zatímco robotizace má za úkol nahrazovat člověka při fyzicky namáhavé a jednotvárné práci, případně mu pomoci vyvarovat se nebezpečným prostředím [23].

Obě procedury tak dopomáhají ke snižování výrobních nákladů a soudobému zvyšování kvality i produktivity práce. Investice se tedy vyplácí nejen v technickém, ale hlavně i ekonomickém směru a robotizace se tak v posledních letech těší vysoké popularitě, což dokazuje i Graf č. 1. Například v České republice vzrostl prodej robotů v roce 2013 o celých 29%, který má na svědomí hlavně automobilový průmysl, jenž kraluje odběratelům i ve světovém měřítku [2, 20]. A i když by se mohlo zdát, že robotizace přispívá k nezaměstnanosti, jde jen o falešný mýtus. Nejnovější výzkumy totiž dokazují vznik tří až pěti pracovních míst na každého nově zavedeného robota [11].

Graf 1 Roční poptávka průmyslových robotů [10]



2 Průmyslové roboty a manipulátory

2.1 Definice průmyslových robotů

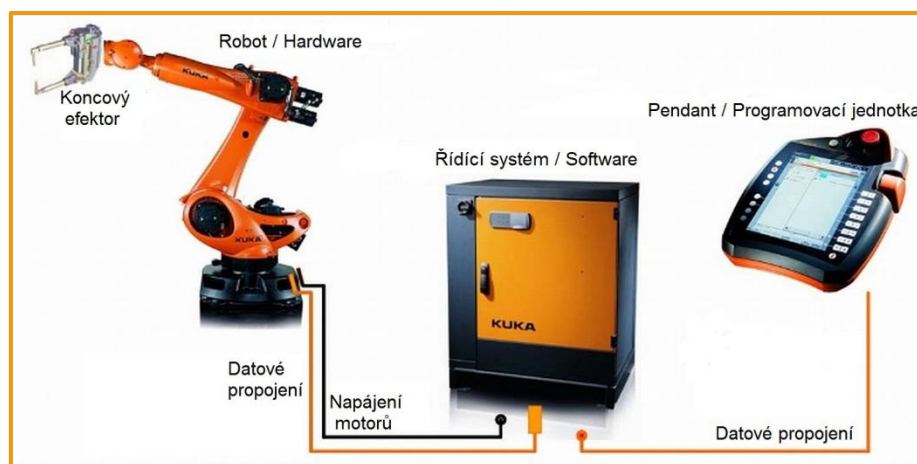
Jak již bylo zmíněno v úvodu, existuje mnoho rozdílných definic. Avšak ve snaze poskytnou jednu všeobecně přijatelnou, byl průmyslový robot definován organizací ISO¹ dle normy ISO 8373 jako: „Automaticky řízený, opětovně programovatelný, víceúčelový manipulátor pro činnost ve třech nebo více osách, který může být upevněn na místě nebo mobilní k užití v průmyslových automatických aplikacích.“ [2]

Využívá se tedy pro práce vyžadující přesnost, rychlost i jistou odolnost proti okolním vlivům. Roboty jsou schopny vykonávat úkony spolehlivěji než lidé a i přes jejich současné nedostatky se stávají se stabilnějšími i potřebnějšími, což dokazuje předchozí kapitola [2].

2.2 Struktura a kinematická konstrukce

Průmyslový robot má specifickou strukturu, která obsahuje tři hlavní části (Obr. 2.1) [3].

- Programovací část – programování robota je realizováno pomocí uživatelského rozhraní. Při on-line programování se jedná o tzv. teach-pendant, zatímco u programování off-line nebo CAD/CAM jde o uživatelské softwary. Oba způsoby jsou podrobněji rozebrány v následující kapitole [3].
- Řídicí systém – jeho úlohou je na základě vstupů a informací řídit činnost robota a jeho periferie. Řízení zajišťuje mikroprocesorový systém pracující na bázi multitaskingu [3].
- Mechanická část – je složena z podstavy, akčního systému a koncového efektoru sloužícího jako funkční částí robota [3].



Obr. 2.1 Struktura průmyslového robota [3]

Akční systém robota zahrnuje v podstatě celý pohybový mechanismus. Je poskládan z binárních členů spojených kinematickými dvojicemi² (dále jen KD), které odebírají zpravidla 1 stupeň volnosti [32]. Používají se nejčastěji translační (T) a rotační (R) KD, jejichž druh a uspořádání určuje pracovní prostor, konstrukci i požadavky na řízení. K dosažení libovolné pozice v prostoru robota je zapotřebí tří stupňů volnosti a tedy i tří KD, jejichž kombinací lze dosáhnout různých kinematických řetězců [5, 24].

¹ mezinárodní organizace pro normalizaci; z řeckého isos = stejný, ve smyslu shodného pojmenování

² představuje 2 členy robota spojené kinematickou vazbou



2.3 Nejpoužívanější druhy robotů

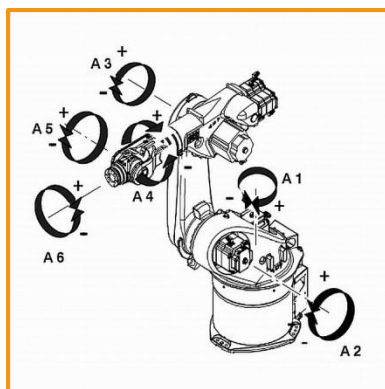
V návaznosti na předchozí konstrukci se dostáváme k nyní nejpoužívanějším průmyslovým robotům. Jelikož byl tento obecný základ zamýšlen jako podklad pro závěrečnou manipulační úlohu, je tedy zaměřen jen na 3 nejdůležitější a k manipulaci nejčastěji používané roboty – 6DOF, SCARA a roboty paletizační.

6DOF³ – robot operuje se šesti stupni volnosti, které jsou rozšířením základního kinematického řetězce RRR. Tři stupně volnosti totiž poskytují jen nastavení libovolné pozice v prostoru, což může být u některých operacích nedostačující a vyžaduje se i změna orientace uchopeného předmětu nebo nástroje. Aby se dosáhlo potřebné orientace, jsou zavedeny další tři stupně volnosti zajištěné koncovým členem ramene. Celkem tedy šest stupňů volnosti, které odpovídají šesti manipulačním osám a jsou vyobrazeny na obrázku č. 2.2 [9, 24].

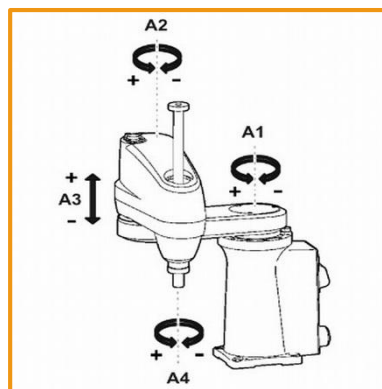
Jedná se typ robota využívající sériovou kinematiku s univerzálním použitím. Jeho použití není omezeno jen na manipulaci, ale je používán i pro aplikace jako svařování, vrtání, montáž a další. Rameno je složeno pouze z vazeb rotačních, které mohou vykonávat pohyb nezávisle na sobě [32]. Z čehož vyplynuly jeho největší výhody, kterými jsou vysoká variabilita použití, pružnost a větší pracovní prostor. I přes nevýhody jakými může být nižší rychlost, zrychlení a méně přesné polohování jde o nejjednoduššího robota, jehož zdokonalováním se zabývají největší výrobci robotů jako KUKA, ABB, Fanuc, Comau, Stäubli a další [9, 32].

SCARA⁴ – je typ využívající sériovou kinematiku se čtyřmi osami s hlavním využitím při montážích. Jedná se základní řetězec RRT se specifickým vodorovným otočným ramenem rozšířeným o možnost další rotace (Obr. 2.3). Výhody představují hlavně jednoduchost, přesnost, rychlost a jako nevýhodu lze zmínit sníženou manipulační schopnost v ose Z [16, 32].

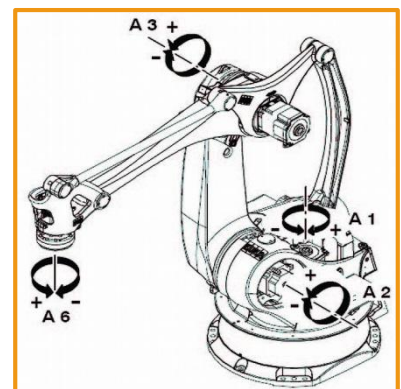
Paletizační roboty – využívají základní řetězec RRR, přičemž bývají rozšířeny o další stupně volnosti. Obvyklá konstrukce tedy čítá 4 nebo 5 os s tím rozdílem, že některá z ramen jsou spojena a otáčení je tak na sobě závislé (Obr. 2.4). Využití těchto robotů vyplývá už z názvu a díky větší robustnosti dosahují únosnosti až 1300 kg [9, 32].



Obr. 2.2 Schéma robotu 6DOF [13]



Obr. 2.3 Schéma robotu SCARA [13]



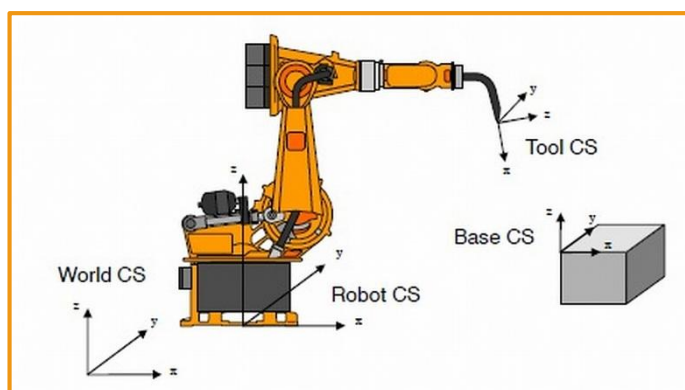
Obr. 2.4 Schéma paletizačního robotu [5]

³ Degrees Of Freedom = stupně volnosti

⁴ Selective Compliance Assembly Robot Arm = robot vhodný pro montáž

2.4 Souřadnicové systémy

Pro všechny představené roboty je jedním z nejdůležitějších úkolů pohyb. Pro jeho řízení potřebuje robot přesný popis polohy vzhledem k pracovišti a ve vztahu k manipulovanému objektu. Tento popis polohy je realizován pomocí souřadnic v jednotlivých souřadných systémech. V robotickém systému jsou definovány čtyři (Obr. 2.5) [14, 24].



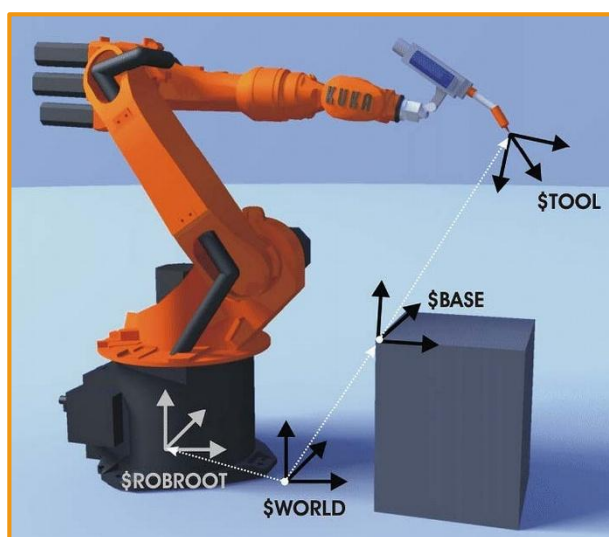
Obr. 2.5 Souřadnicové systémy [14]

World – je trvale definovaný světový kartézský souřadný systém. Ve výchozím nastavení je definován v patě robotu a slouží jako základní pro systémy Robot i Base [14, 15].

Robot – je souřadný systém robotu, který se vždy nachází ve spodní části a používá se jako referenční systém pro mechanické konstrukce robotu. Současně definuje pozici robotu vzhledem ke globálnímu systému World [14, 15].

Base – je souřadný systém vztahovaný ke světovému systému a definuje polohu. Probíhá v něm programování robotu. Definuje jej uživatel [14, 15].

Tool – je pak souřadný systém nástroje umístěný ve středním bodu nástroje (TCP⁵). Systém si definuje sám uživatel a je vztahován k souřadnému systému Base. V případě pohybu středního bodu nástroje se pohybuje s ním [14, 15].



Obr. 2.6 Vzájemné vztahy souřadných systémů [15]

⁵ Tool Centre Point = střední bod nástroje



2.5 Řízení pohybu

Samotné řízení pohybu je pak možno provádět dvěma různými způsoby.

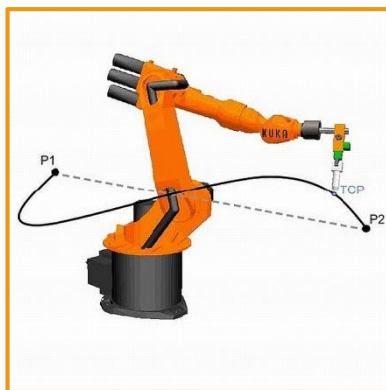
- Pohyb v kartézských souřadnicích – změna polohy středního bodu nástroje v kladném nebo záporném směru podél os (X, Y, Z) s možností natočení (A, B, C) [15].
- Osově specifický pohyb – samostatná změna natočení jednotlivých os v kladném a záporném směru [15].

Z důvodu nepraktického programování v osově specifickém režimu je programování prováděno v režimu kartézském a řídicí systém souřadnice automaticky převede na natočení jednotlivých os [14]. Ještě předtím je však zapotřebí jednotlivé pohyby definovat pomocí příkazů. Existují tři druhy a u všech je zapotřebí definovat koncový bod, protože bod startovní je obvykle koncovým bodem předchozího pohybu [14, 15].

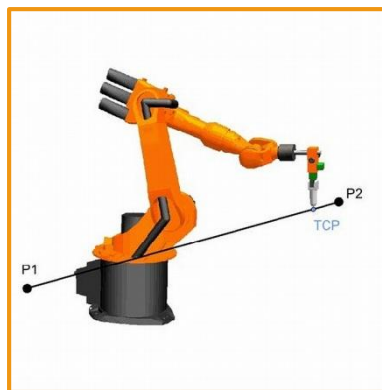
PTP⁶ – je pohyb, který vede TCP nejrychlejší cestou do koncového bodu (Obr. 2.9). Dráhu robotu však nelze předvídat, protože nejkratší cesta obvykle nemusí být tou nejrychlejší a díky rotačním osám robotu mohou být zakřivené dráhy prováděny rychleji. Pohyb začíná i končí ve všech osách současně [15].

LIN – je pohybem lineárním a TCP opisuje geometricky definovanou přímku, která je nejkratší cestou do koncového bodu (Obr. 2.8) [15].

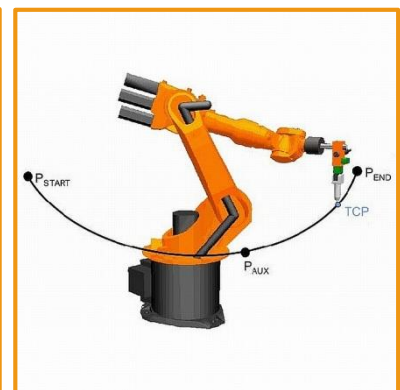
CIRC – je pohyb vykonávaný po kruhové dráze. Kruh však nelze popsat jen dvěma body a musí být tedy zadáván i bod pomocný, který dráhu jasně definuje. TCP pak prochází přes tento bod do bodu koncového (Obr. 2.7) [15].



Obr. 2.9 Pohyb PTP [15]



Obr. 2.8 Pohyb LIN [15]



Obr. 2.7 Pohyb CIRC [15]

⁶ Point To Point = z bodu do bodu

3 Programování robotů

Od základního seznámení s roboty se postupně dostáváme k jejich programování. Roboty svou činnost vykonávají na základně dopředu připraveného programu, který je definován jako posloupnost příkazů vedoucích k vykonání požadované úlohy. Programování tedy lze definovat jako sestavování jednotlivých příkazů dle daného algoritmu a jejich formování do koncové podoby – programu [3].

Průmyslových robotů a jejich výrobců však existuje nemalé množství. Zatímco z mechanického a elektronického hlediska vyvíjí různí výrobci podobné produkty, větší rozdíly jsou obvykle v jejich programovacích jazycích a systémech. Jednotlivé jazyky od různých výrobců tak mají své odlišné syntaxe a specifikace [16]. Z tohoto důvodu není lehké danou problematiku lehce vystihnout. Následující strany jsou tedy zaměřeny spíše na deskripci jednotlivých metod než na syntaxi a aplikace. I přes různé specifikace však lze obvykle využít různé programovací přístupy, které lze dělit na [5]:

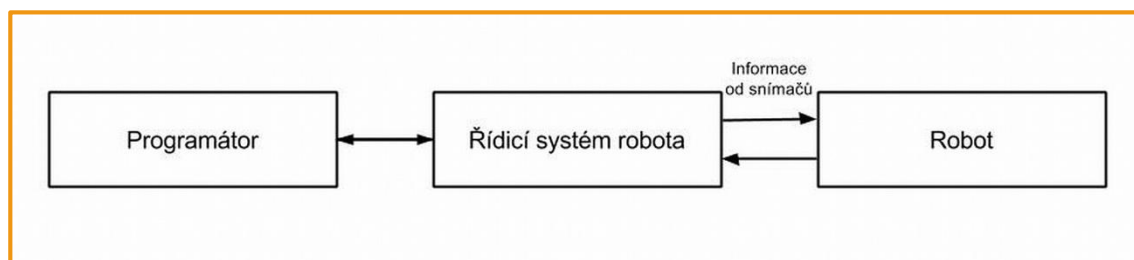
- on-line programování,
- off-line programování,
- CAD/CAM programování (lze řadit pod off-line).

3.1 On-line programování

On-line programování umožňuje přímou interakci člověka s robotem. Programování je vykonáváno naváděním robota přes požadované manipulační body, které jsou zapisovány do řídicího systému robota. Po navádění do jednotlivých pozic pak následuje logické řízení koncového efektoru, nástroje nebo případných periferních zařízení. Programování provádí uživatel přímo na pracovišti pomocí uživatelského rozhraní tzv. teach-pendantu, který plní funkci ovládacího panelu [5, 8].

Výhodou této metody programování je znalost trajektorie robota a práce v reálném prostředí s možností současné kontroly funkčnosti programu. Mezi nevýhody pak lze zařadit poměrně dlouhou dobu programování při složitějších manipulačních pohybech nebo cyklech. Hlavní nevýhodou však představuje odstávka pracoviště, která způsobuje ztrátu na výrobě. Pouze ojedinělé případy pak dovolují některým zařízením omezeně pracovat během programování [3].

Pod on-line programování, jehož postup je znázorněn na obrázku č. 3.1, lze také zařadit poměrně novou kooperaci robota a člověka, která je spolu s programováním pomocí teach-pendantu a samotnými uživatelskými rozhraními popsána níže.



Obr. 3.1 Postup on-line programování [8]



3.1.1 Uživatelská rozhraní

Jak již bylo zmíněno, uživatelská rozhraní sloužící k on-line programování se nazývají teach-pendants. Tyto současné moderní programovací jednotky jsou postavené na bázi osobních počítačů. Typické jednotky bývají propojeny s řídicím systémem pomocí kabelu a dokáží do něj přenášet informace přímo. Avšak vyskytuje se také varianta s bezdrátovým připojením pomocí zabezpečené bezdrátové sítě (Comau robotics) [3, 5].

Pendantsy zprostředkovávají programování skrze klávesnici s barevným displejem. Jejich ergonomický tvar umožňuje pohodlné držení v ruce a dovoluje přímé ovládání pohybů robotu pouhým stiskem několika funkčních tlačítek nebo pohyby joysticku [16]. Displej plní funkci jako u normálního počítače, je schopen zobrazit více oken pro zobrazení funkcí robotu, parametrů nebo programů. Teach-pendantsy umožňují monitorování i programování vstupů a výstupů spolu s nastavením pracovních parametrů. Jsou také schopny také zobrazovat systémové informace a případně upravovat nebo i přímo zapisovat technologický program v editoru [3]. Vzhledem k tomu, že většinu současných teach-pendantů lze vlastně považovat za pokročilé počítače s výkonnými operačními systémy, možnosti pro rozvoj moderních rozhraní jsou obrovské a flexibilní [19].

Nynější teach-pendantsy je možno dle zpracování dělit na 2 kategorie.

„Landscape” design – zpracování, pro které je charakteristickým rozměrem šířka. V porovnání s následující skupinou disponuje větším displejem a joystickem nebo prostorovou myší pro manuální ovládání. Zpravidla se drží jednou rukou, zatímco druhá zajišťuje ovládání. Typickým příkladem je pendant firmy ABB na obrázku č. 3.2. Lehce polemizovat lze nad nejnovějším pendantem firmy KUKA, který se svými rozměry přiklání spíše k druhé variantě. Avšak díky prostorové myši a typickému úchytu s bezpečnostními tlačítky na zadní straně se jedná o „landscape” design (Obr. 3.3) [5].

„Portrait” design – charakteristickým rozměrem je výška a k ovládání je obvykle nutné použití obou rukou. Oproti předchozí skupině disponuje menším displejem, pod kterým jsou většinou umístěna funkční ovládací tlačítka. S touto kategorií se ztotožňují designy firem Fanuc (Obr. 3.4), Comau a další [5].



Obr. 3.4 Teach-pendant firmy Fanuc [7]



Obr. 3.3 Teach-pendant firmy KUKA [13]



Obr. 3.2 Teach-pendant firmy ABB [1]

3.1.2 Programování teach-pendanty

Jedná se o metodu on-line programování, která k navádění robota využívá již zmíněná uživatelská rozhraní – teach-pendanty. Při programování tzv. postupným učením je rameno robotu postupně naváděno operátorem do jednotlivých pozic v prostoru, kde má být následně vykonána určitá akce. Slovo postupně znamená z bodu do bodu přesně tak, jak je vyžadováno, aby robot následně pracoval. Navádění je prováděno pomocí funkčních tlačítek, joysticku nebo prostorové myši umístěné na pendantu [24].

Orientace nástroje a souřadnice požadované pozice, kterou může obsluha nastavovat libovolně dlouho, se pak pomocí určeného tlačítka uloží do paměti. V paměti je pak zaznamenána určitá posloupnost údajů o poloze a orientaci, ke které se před spuštěním robotu v automatickém režimu musí přiřadit údaje o způsobu propojení jednotlivých bodů v prostoru [27]. Pomocí těchto dodatečných údajů je řídicí systém schopen dopočítat skutečnou trajektorii ramene robotu. Jedná se tak o inverzní kinematickou úlohu, kdy systém ze známých počátečních a koncových pozic dopočítává potřebnou změnu natočení jednotlivých ramenních kloubů [3, 27].

Při spuštění automatického režimu pak robot využívá údajů z paměti a postupně prochází všemi naprogramovanými pozicemi. Pohyb mezi jednotlivými pozicemi však nemusí být obsluze dostatečně známý (PTP), což lze považovat za nevýhodu. Výhodou je pak velmi jednoduchá a přirozená cesta programování, umožnění velmi přesného nastavení jednotlivých parametrů polohy, orientace a možnost jejich vizuální kontroly [16, 27].



Obr. 3.5 Programování pomocí teach-pendantu [3]

Nicméně programování pomocí teach-pendantu neumožňuje jen zaznamenávání požadovaných bodů podél dráhy pro pozdější přehrávání. Nabízí také mnoho dalších užitečných programovacích doplňků a funkcí. Jako například testování prováděného programu, zobrazení uživatelských a chybových zpráv, výzvy, nabídky, zobrazení a úpravu polohových dat, programové proměnné, vstupy, výstupy a v neposlední řadě provádění operací se soubory (jako kopírování, mazání, přenos atd.) [21]. Důležitou částí jsou také bezpečnostní prvky a opatření, ze kterých se na programovacím pendantu vykytují bezpečnostní tlačítka, bez jejichž stlačení nelze s robotem dále operovat a tlačítko pro centrální zastavení. Modernější jednotky také disponují analytickými funkcemi pro optimalizaci práce robota. Některé klávesy s funkcemi jsou popsány na další straně na obrázku č. 3.6 [3, 21].



Obr. 3.6 Jednotlivá funkční tlačítka teach-pendantu [3]

Jako příklad lze vzít manipulační úlohu ze závěru práce. Operace vyžaduje vyzvednutí břemene a jeho následné umístění na místo se dvanácti pozicemi. Obsluha tedy pendantem programuje a ukládá pozice robota postupně krok za krokem, z čehož vyplývá poměrně velký počet kroků, ale na druhou stranu potřeba jen minimálního zaškolení pro obsluhu [21]. Lze tedy konstatovat, že programování pomocí teach-pendantů je vhodné pro menší a jednodušší úkony. Ovšem s novějšími a modernější programovacími jednotkami, které umožňují zpracování komplexnějších ploch jako je tahle, se vhodnost této metody, i přes nesporné výhody off-line programování, dále rozšiřuje [21].

3.1.3 Kooperace robot/člověk

Vzhledem k současným trendům je zapotřebí zmínit i kooperaci mezi člověkem a robotem, která se ubírá velmi perspektivním směrem. Jedná se o tzv. interaktivní programování, kdy operátor přímo spolupracuje nebo jinak i interaguje s robotem [26].

Stavba, zvyšování flexibility robotů a strategie kooperačních pracovišť patří k hlavním úkolům současného vývoje robotů. Ještě nedávno byly robotizovaná pracoviště díky možnému nebezpečí úrazu přísně uzavřenými prostory [26]. Avšak díky projektu EU, zaměřeného na podporu konkurenceschopnosti malých a středních podniků, začali postupně vznikat nové bezpečnostní systémy a pracoviště, kde člověk přímo spolupracuje s robotem. Došlo ke zjištění, že stejný princip lze využít také v klasických chodech robotizovaných pracovišť a začala tak vznikat nová éra robotů spolupracujících s lidmi [1, 26].

Kombinace průmyslových robotů a lidské obsluhy tak současně využívá silných stránek obou faktorů. Všude, kde je zapotřebí například vysoké rychlosti pracovního cyklu, opakovatelné přesnosti, případně pokud není místo zcela přístupné nebo se nachází pro člověka v nevhodné ergonomické pozici, je velmi výhodné zapojit do výroby roboty. Všechny tyto výhody jsou navíc ještě zvýšeny zapojením člověka do výrobního procesu, čímž je dosaženo zvýšené flexibility, vizuálního vyhodnocování a schopnosti pružné reakce na okolní podněty. Avšak žádné z těchto výhod by nebylo možné dosáhnout bez odstranění klasických omezení pro řešení bezpečnosti na pracovišti – oplocení [31].

Odstranění oplocení a následný provoz kooperačních pracovišť však musí podléhat pevným bezpečnostním pravidlům a předpisům. Předpokladem pro budoucnost je pak nárůst kooperačních pracovišť a rozmach robotů i do dalších výrobních úseků [26]. Přičemž přední výrobci robotů již pracují na vývoji robotů pro kooperaci a postupně představují nové koncepce robotů.

KUKA LBR iiwa – je kooperační robot od firmy KUKA. LBR⁷ iiwa⁸ dokáže úzce spolupracovat s člověkem a tak řešit vysoce citlivé úlohy. Existuje ve dvou provedeních s max. hmotností 7 a 14 kg a pracovním dosahem 800 a 820 mm [13]. Člověk se může robotu dotýkat přímo i v průběhu jeho činnosti, což je zajištěno sedmi citlivými momentovými senzory, které neustále komunikují s řídicím systémem a tím nedovolí ohrozit bezpečnost obsluhy. Přínosná je také možnost programování pomocí navádění ramene robotu rukou, protože umožňuje pružně reagovat na změny výrobního procesu [31].



Obr. 3.7 Robot KUKA LBR iiwa [13]

ABB YuMi – byl vyvinut společností ABB. Jedná se o průmyslového robota se dvěma pažemi pro montáž drobných součástek, který dokáže spolupracovat s člověkem. Zařízení je tvořeno flexibilními pažemi, systémem podávání součástek a kamerovým systémem pro jejich umístění s hlavním využitím v odvětví spotřební elektroniky. Robot YuMi⁹ je vyráběn s pracovním dosahem 500 mm, nosnou zátěží 0,5 kg a možností ručního programování. Jedná se o prvního kooperačního robota firmy ABB, ale dle jejich oficiálního prohlášení o koupi společností Gomtec plánují zkombinovat výzkum obou společností a vyvinout nového robota zaměřeného na spolupráci mezi člověkem a robotem [1].



Obr. 3.8 Robot ABB YuMi [1]

⁷ „leichtbauroboter“ = robot lehké konstrukce

⁸ Intelligent Industrial Work Assistant = inteligentní průmyslový asistent

⁹ „You and Me“ = „ty a já“



Fanuc CR-35iA – byl vytvořen firmou Fanuc. Robot CR^{10-35iA} je kooperační robot vybavený vestavěnými silovými snímači. Obložení měkkou pryží pak zajišťuje schopnost pracovat v těsné blízkosti lidí bez nutnosti jiných bezpečnostních zařízení. Obsluha tedy může robota jednoduše navádět nebo odstrčit bez nebezpečí, které by u normálních robotů hrozilo. Robot je vyráběn s pracovním dosahem 1813mm a na kooperační roboty velmi velkou užitečnou zátěží 35 kg. Využití nachází při opakujících se úlohách v automobilovém průmyslu, při balení, distribuci a může být vybaven různými kamerovými systémy [7]



Obr. 3.9 Robot Fanuc CR-35i [7]

3.2 Off-line programování

Mimo použití klasického teach-pendantu při on-line programování robotů je zde také možnost využití tzv. off-line programování. Metoda je založena na softwaru umožňujícím 3D virtuální simulaci robotizovaného pracoviště (buňky) včetně jejího okolí. Programování se provádí v předstihu a softwary umožňují definici drah, pohybů a jiných činností robotu s ohledem na příslušné aplikace, které se pak exportují v příslušných formátech programovacích jazyků (KRL, RAPID atd.). Takto vygenerovaný program je pak dále možné nahrát do řídicího systému robotu [5, 8].



Obr. 3.10 Virtuální model pracoviště v programu RobotStudio [1]

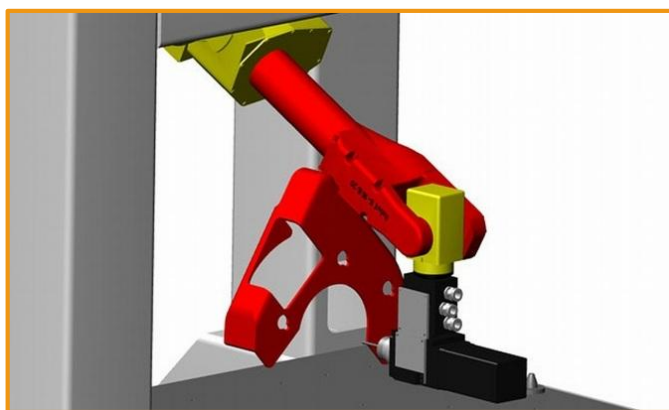
Na základě simulace činností je dále možno optimalizovat a upravovat chod robotu, sledovat celkový reálný pracovní čas, testovat dosažitelnosti bodů, dráhy, pracovní prostory, případně využívat různé doplňky a komponenty [5]. V poslední době se stále zvětšuje možnost integrace různých simulačních balíčků, které jsou specializovány na různé aplikace

¹⁰ Collaborative Robot = spolupracující robot

jako svařování, obrábění a další. Systémy také často podporují import součástí z různých CAD¹¹ systémů a umožňují automatické generování dráhy s extrémně přesnými výpočty časů a pohybů vzhledem k vymodelovaným součástem [5, 30].

U off-line programovacích softwarů však nejde jen o simulaci a jejich bezesporu důležité funkce. Klíčovým atributem je ona možnost převést vytvořené dráhy a simulace do programovacího jazyka konkrétního robotu. Přestože se však jedná o pravděpodobně nejdůležitější vlastnost, spoléhat se jen na vyexportovaný program by nebylo moudré [5]. Z praktického hlediska je nutné generovaný program více či méně upravit a odladit jeho nedostatky, které mohou být dány nepřesností simulačního modelu vzhledem k realitě. Důležitým úkolem programátora je tedy vytvořit co nejvěrnější model fyzického systému tak, aby bylo zapotřebí co nejmenšího množství korekcí přímo na pracovišti [4, 8].

Využití všech výše uvedených vlastností poskytuje mnoho výhod a důvodů proč zvolit právě programování off-line. Nejvíce sledovanou výhodou je možnost vytvářet program v mimo reálném prostředí bez časového stresu a s dostatečným předstihem před realizací projektů, což on-line programování neumožňuje [8]. Používání běžně dostupných víceúčelových počítačů pro softwary, pak kromě zjevné ekonomické výhodnosti eliminuje potřebu školit programátory pro specializovaná robotická prostředí. Oproti on-line programování se snižuje čas potřebný k vytvoření programu i výrobě a provádění změn jde realizovat výrazně jednodušším způsobem. Také rozvíjení programu v prostředí provozu vytváří nezanedbatelné nepříjemnosti, které našťastí tato metoda eliminuje, čímž výrazně zvyšuje produktivitu práce a snižuje prostoje robotu [8, 21].



Obr. 3.11 Detekce kolize robotu [25]

Nevýhodu pak představují další investice mimo robota, které tento programovací přístup vyžaduje. Na druhou stranu je však třeba poznamenat, že programovací software obsáhne všechny případné roboty i zařízení [8].

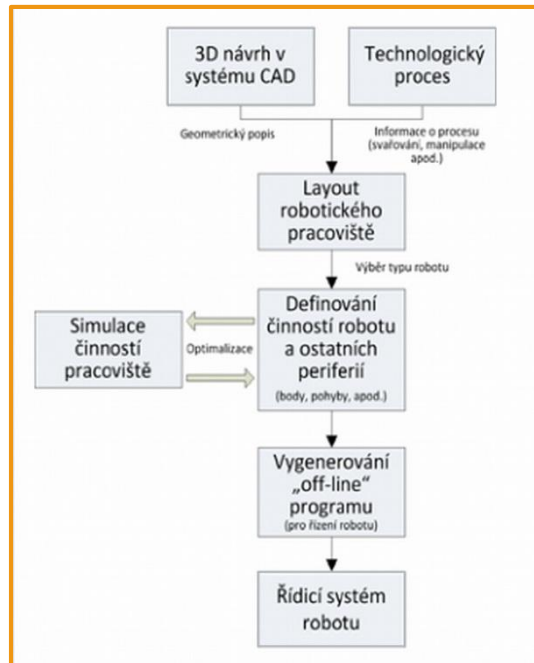
3.2.1 Obecná metodika

Nejprve je nutné definovat 3D model geometrie objektu, který lze navrhnout přímo v softwarovém produktu nebo ho jednoduše importovat z jiného programu [4]. Druhým bodem je vytvoření procesního popisu úlohy, následované načtením simulačního modelu robotu z knihovny a návrhem robotické buňky – layoutem. Dále je pak možné definovat dráhy, body a činnosti robotu spolu s ostatními periferiemi. Simulace posléze odhaluje nedostatky plánování, které se optimalizují. Zefektivňuje se pracovní cyklus, čímž se maximalizuje výrobnost celé aplikace [3, 30]. Po dosažení optimální varianty je následně

¹¹ Computer Aided Design = počítačem podporované projektování



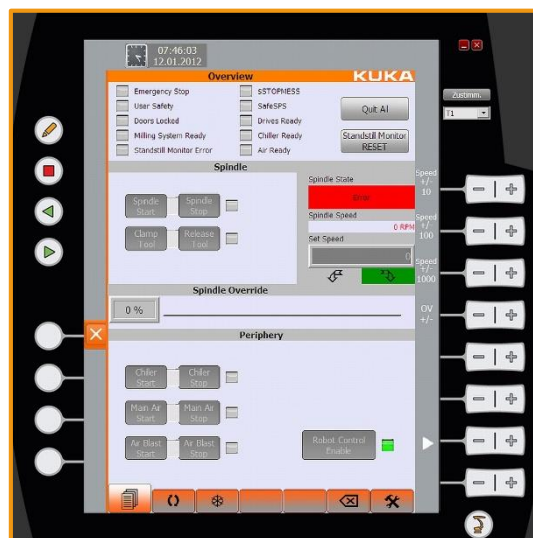
vygenerován program ve tvaru odpovídajícím programovacímu jazyku robotu. Ten se posléze importuje do reálného řídicího systému, kde ho čeká korekce a ostré spuštění v provozu [4]. Obecná metodika off-line programování je znázorněna na obrázku č. 3.12.



Obr. 3.12 Metodika off-line programování [4]

3.2.2 Virtuální kontrolér

Většina níže zmíněných simulačních off-line softwarů od výrobců využívají k programování virtuální kontrolér. Jedná se o teach-pendant vytvořený ve virtuálním prostředí počítačového softwaru. Ovládání a programování kontroléru je stejné jako u reálného teach-pendantu v on-line režimu [21]. Obsahuje stejné funkce a lze pomocí něj simulovat a měřit časy pracovních cyklů. Off-line programování pomocí virtuálního kontroléru využívá stejné inverzní kinematiky jako u on-line programování robotu a vytvořený pracovní cyklus tak odpovídá reálné situaci [3].



Obr. 3.13 Virtuální kontrolér KUKA [18]

3.2.3 Simulační softwaru

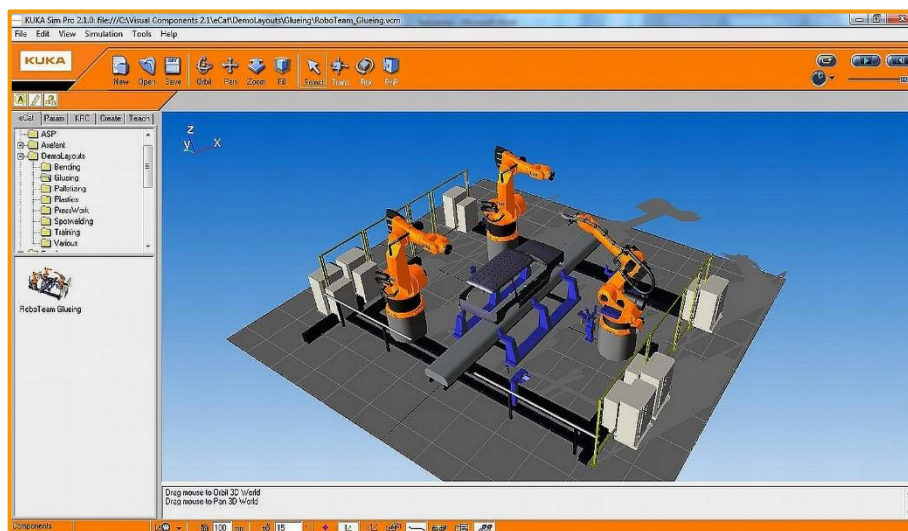
Simulační softwaru fungují jako oboustranné 3D systémy, které umožňují přenášet provedené změny mezi režimy on-line a off-line. Cílem většiny výrobců je, aby se programování všech robotů provádělo v off-line režimu. Umožňuje to vykonávat rychlé změny ve výrobě a spoustu dalších dříve popsanych výhod. Z tohoto důvodu, si tak největší a nejznámější robotické firmy vyvíjí své off-line programovací systémy a simulátory samy. Ovšem existují také univerzálnější softwaru od různých společností. Komerčně dostupné softwaru tedy lze rozdělit na dvě kategorie [5]:

- systémy vyvíjené výrobcí průmyslových robotů,
- univerzální systémy od softwarových společností.

Využití softwarů z první kategorie zpravidla funguje jen s průmyslovými roboty výrobce, jež systém vyvinul. Nejsou tedy univerzální, což by se dalo považovat za nevýhodu. Avšak výhody, které z těchto systému plynou, tuhle nevýhodu převyšují. Fungují na principu virtuálního kontroléru, a tak umožňují shodné programování jako na teach-pendantu a generování programu v programovacím jazyku příslušného robotu. Podrobněji jsou níže představeny systémy od největších firem Kuka, ABB a Fanuc. [5].

Druhou kategorií jsou systémy univerzální, z jejichž názvu vyplívá použití, které je vhodné pro různé programovací jazyky od různých výrobců robotů. Výhodu tak při tvorbě layoutu představuje možnost zvolit nejlépe vyhovujícího robota. Nevýhodou pak může být nižší přesnosti při výpočtech a možné odchylky simulované dráhy od skutečné z důvodu rozdílných výpočtových algoritmů. Podrobněji jsou zde přestaveny systémy RobCAD a Workspace [5].

KUKA.Sim Pro + Office.Lite – jsou softwaru sloužící k off-line programování od firmy KUKA, která patří mezi přední výrobce průmyslových robotů. Nástroj KUKA.Sim Pro je programovací a simulační software, zahrnující mimo jiné i funkce dalších nástrojů od firmy jako KUKA.Sim Layout [13]. Umožňuje tak pomocí konstrukční knihovny vytvářet a prezentovat simulace robotů, dopravníků a ostatních zařízení navrhované robotické buňky. Lze srovnávat i optimalizovat různé koncepce a v reálném čase znázornit průběhy naprogramovaných pohybů. Současné verze také detekují kolize, analyzují dosažitelnost jednotlivých pracovních bodů a umožňují import CAD souborů. Programování probíhá přímo pro jazyk KRL (KUKA Robot Language) [13].



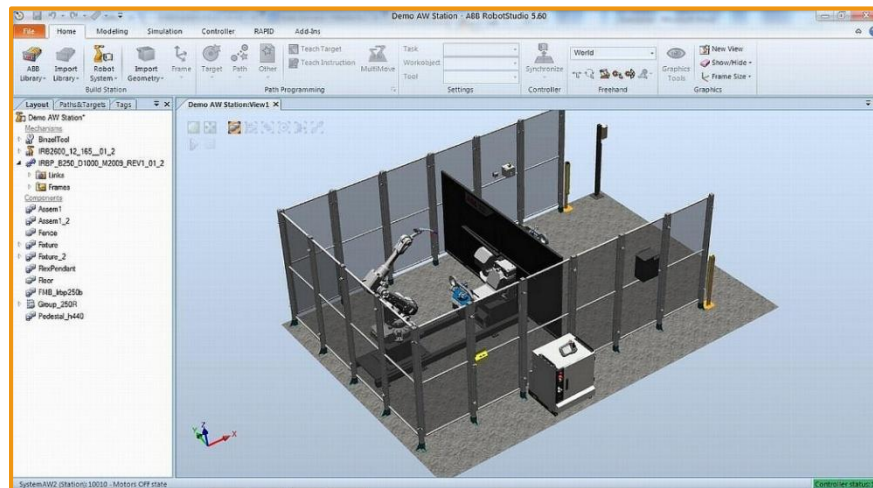
Obr. 3.14 Ukázka prostředí Kuka.Sim Pro 2.1 [9]



KUKA.OfficeLite pak díky stejným vlastnostem jakými disponuje řídicí systém robotu, umožňuje editovat programy na kterémkoliv osobním počítači a je nedílnou součástí toho programovacího celku. Samotné Sim Pro totiž umožňuje jen simulace. Hotové programy pak stačí zkopírovat do robotu a v on-line režimu opravit případné nesrovnalosti [13].

RobotStudio – simulační a off-line programovací software pro roboty vyvinuté společností ABB. Stejně jako všechny softwary umožňuje RobotStudio programování na počítači bez nutnosti zastavení výroby, čímž zvyšuje celkovou produktivitu. Předem připravený program poskytuje možnosti ke zvýšení ziskovosti tím, že zprostředkovává možnost školení, programování a optimalizaci mimo robota [1].

RobotStudio je založeno na virtuálním řídicím systému od společnosti ABB, který je přesnou kopií skutečného softwaru. Proto poskytuje relativně realistické simulace. Dokáže importovat data v hlavních CAD formátech, díky kterým lze generovat přesnější programy. Nechybí samozřejmě možnost zjištění dob cyklů, detekce kolizí a analýza dosažitelnosti pracovních bodů. Poskytuje také možnost připojení programovacích balíčků pro technologické operace jako obrábění, ohýbání, lakování a další [1].



Obr. 3.15 Ukázka prostředí RobotStudio 5.6 [9]

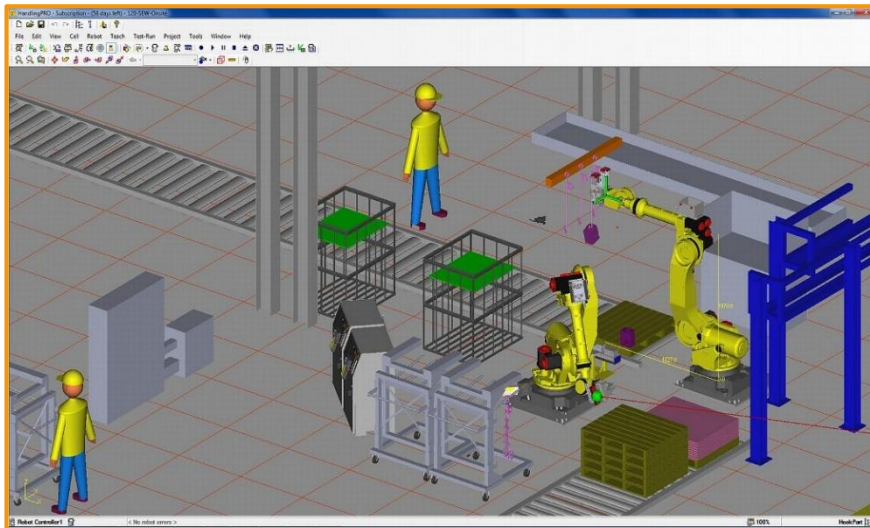
RoboGuide – je posledním ze tří zmíněných softwarů od výrobců. Software se věnuje čtyřem základním balíčků, kterými jsou HandlingPRO (manipulace), PaintPRO (lakování), PalletPRO (paletizační systémy) a WeldPRO (svařování). Všechny čtyři základní balíky i samotný software byl vyvinut společností Fanuc [7].

HandlingPRO umožňuje uživatelům simulovat robotický proces v 3D prostoru nebo provádět studie proveditelnosti manipulačních úloh bez fyzické potřeby a nákladů na instalaci pracovní buňky. PaintPRO pak dokáže zjednodušovat naučenou robotickou dráhu. Je speciálně navržen, aby vytvářel dráhu vhodnou pro lakovací nástroje skutečného robotu obsahujícího software Fanuc PaintTool. Náklady i čas na vývoj a integraci paletizačních případně depaletizačních softwarů jsou pak šetřeny pomocí softwaru PalletPRO. Poslední a nejnovější balíček je WeldPRO umožňující simulaci robotického svařování elektrickým obloukem [7].

Stejně jako oba dva předešlé softwary, poskytuje RoboGuide možnost importu dat v CAD formátu, které ulehčují projektování layoutu. Zvládá i zobrazení detekce kolize, měření časů, analýzu dosažitelnosti a vytvoření kompletního programu dle všech potřebných náležitostí programovacího jazyka firmy Fanuc [7]. Ukázka prostředí je zobrazena na obrázku č. 3.16 na následující straně.



BAKALÁŘSKÁ PRÁCE

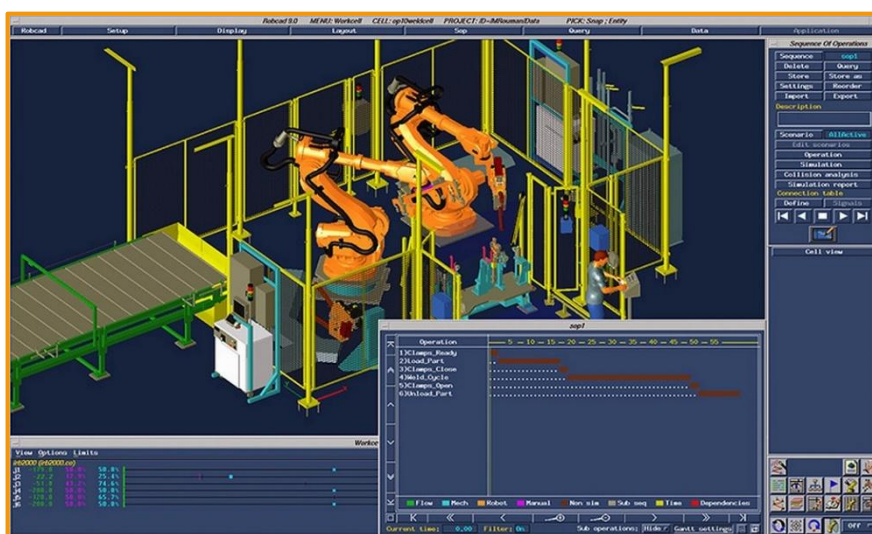


Obr. 3.16 Ukázka prostředí RoboGuide [7]

Dalšími simulačními softwary od výrobců průmyslových robotů jsou:

- Stäubli – Robotics Suite,
- Motoman – MotoSim,
- Denso – Wincaps III,
- Mitsubishi – Melfa Works,
- Kawasaki – K-ROSET,
- a další.

RobCAD – je prvním ze dvou podrobněji představených univerzálních softwarů. Byl vyvinut firmou Siemens a řadí se pod komplexnější digitální výrobní řešení Tecnomatix sloužícího nejen k robotizaci a automatizaci výroby. RobCAD je nástroj pro design a simulaci robotických buněk umožňující simulaci, optimalizaci a off-line programování procesů [22].



Obr. 3.17 Ukázka prostředí RobCad 9.0 [22]

V softwaru lze zadat výpočet délky cyklů v různých fázích časového plánu vývoje. Poskytuje také detekci kolize a možnost importu nebo případné spolupráce s CAD systémy, jak tomu je i u softwarů vyvinutých výrobcí průmyslových robotů. Dle brožury výrobce pak

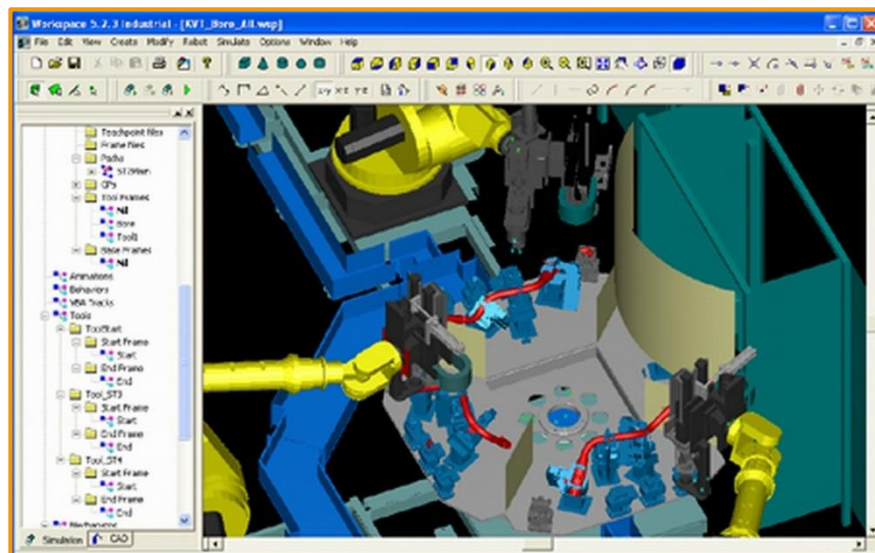


dokáže simulovat technologické aplikace jako bodové nebo obloukové svařování, lakování, řezání vodou, laserem a manipulaci s materiálem [22].

Obsahuje rozsáhlou knihovnu standartních robotů (50+ druhů rozhraní) a konfigurací konkrétních kontrolérů. Specifické informace zahrnující pohyby a vlastnosti programovaného procesu jsou pak pomocí kontrolérů přidávány do generované dráhy robotu. RobotCAD poté na základě simulací vytvoří program, který se přenesou do skutečného robotu [22]. Vygenerovanému kódu je však zapotřebí věnovat zvýšené pozornosti, protože programy generované těmito softwary nebudou nikdy přesné jako ty od výrobců průmyslových robotů.

Workspace – je druhý a poslední podrobněji představený softwarový nástroj od společnosti WAT Solutions. Nabízí simulace a off-line programování robotů bez ohledu na výrobce, jako všechny ostatní univerzální softwary. Vybírat lze z mnoha různých druhů robotů, jejichž fyzické rozvržení lze optimalizovat a tím se vyhnout nákladnému přepracování. Při programování složitějších pohybů a operací může být ovládání obyčejnou myší zdouhavé i pracné, proto se vývojáři rozhodli pro podporu 3D myši, která zvyšuje úroveň kontroly [33].

Tvorbu 3D CAD modelů je možno provádět přímo v programu nebo lze použít import dat ve formátech IGES a SAT. Simulovat lze programy vytvořené interně i převzaté ze skutečné robotické buňky. Automatické generování kódu je pak doplněno o nástroj pro detekci a dynamickou i statickou analýzu kolizí. V softwarovém systému lze také jednoduše vytvořit vlastní koncový efektor, robot nebo jiné mechanické zařízení. Vykresluje pracovní prostory, kontroluje dosažitelnost pracovních bodů a počítá doby cyklů. Možnosti softwarového uplatnění jsou tedy nemalé a svoje uplatnění nachází i ve výuce, pro kterou má vytvořen speciální software Workspace LT [33].



Obr. 3.18 Ukázka prostředí Workspace5 [33]

Dalšími univerzálními simulačními softwary jsou:

- Delmia IGRIP,
- RobotExpert,
- Sim X,
- EASY-ROB,
- RobotWorks,
- CimStation,
- a další.



3.3 CAD/CAM programování

Jak již bylo poznamenáno, dalším a posledním představeným způsob programování zajišťují CAD/CAM¹² systémy. Oproti simulačním softwarům, však jednostranné CAD/CAM systémy neumožňují přenášet provedené změny mezi off-line a on-line režimem. Softwary jsou zaměřeny na CAM obrábění a s jejich pomocí lze tvořit programy pro obrábění součástí nebo těles různých tvarů, velikostí a geometrií [35].

Off-line programování tohoto typu vychází z počítačového modelu pracovní buňky a program je tvořen na základě modelu objektu. Modelování se preferuje ve 3D zobrazení a doprovází jej tvorba drah a výběr potřebného nástroje [8]. Integrovaná knihovna obsahující modely robotů pak zajišťuje načtení libovolného robotu a jeho sladění s dříve vytvořenými drahami. Simulace pak odhalí případné chyby, po jejichž odladění je vygenerován výsledný kód. Obvykle se využívají nestandardní programovací jazyky, což vyžaduje překladač, který vytvoří požadovaný program pro konkrétního robota [8, 35].

Řídicí program se posléze, jako u všech druhů off-line programování, nahrává do řídicí jednotky robotu. V případě výskytu jakékoliv kolize robotu je zapotřebí opravit dráhy přímo v softwaru a vygenerovat nový řídicí program. Kolizní kód tedy nelze opravit pouhým přepsáním některého z příkazů jako u simulačních softwarů, z čehož plyne přívlastek „jednostranné“ systémy. Sledované výhody a nevýhody CAD/CAM stylu programování jsou pak stejné jako u zmíněných off-line simulačních softwarů, kde hlavní přednost představuje programování mimo pracoviště bez přerušení výroby [35].

3.3.1 Metodika CAD/CAM

Obecnou metodiku CAD/CAM programování lze vidět na obrázku č. 3.19. Prvním krokem při vypracování programu pro obrábění je tvorba nebo import a úprava 3D modelu součásti, následovaná programováním drah vybraných nástrojů v závislosti na tvaru součásti. Tyto hlavní přednosti jsou základními stavebními kameny CAD/CAM, protože umožňují o hodně rychlejší a jednodušší programování. Jedná se tedy o zásadní rozdíl oproti stávajícím řešením,



Obr. 3.19 Průběh CAD/CAM programování [6]

¹² Computer Aided Manufacturing = počítačem podporovaná výroba

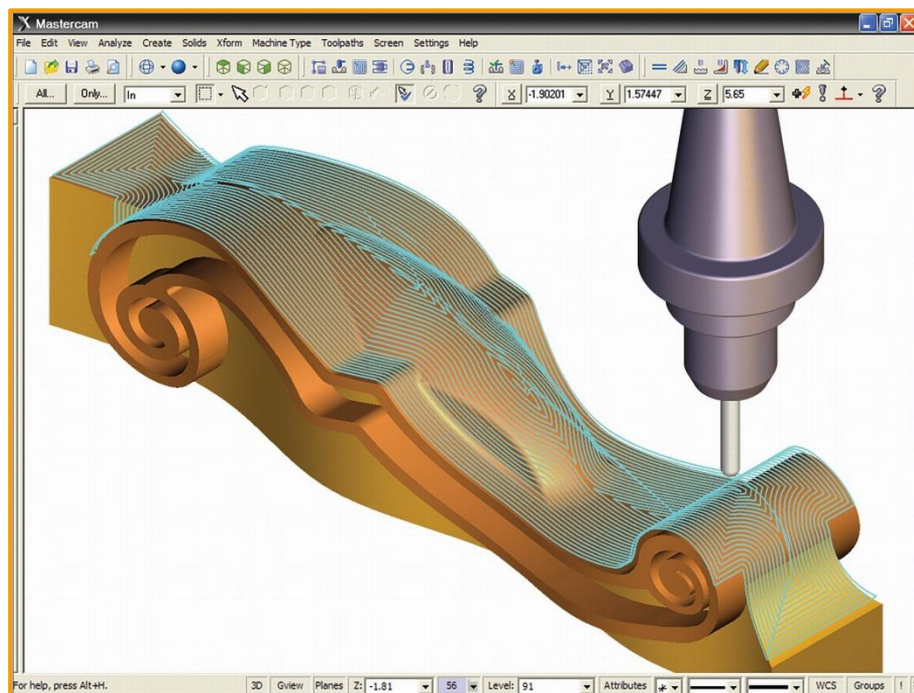


kteří jsou většinou závislé na manuálním programování bod po bodu. Z knihovny se pak nahrávají k vypočteným drahám a nástrojům modely robotů, případně se konstruuje robotické buňky. Spolu s následujícími dalšími moduly, které slouží pro úpravy a ladění pohybů, simulace a vygenerování výsledného kódu, tak CAD/CAM systémy vytváří komplexní způsob programování [17, 35].

3.3.2 Softwarové systémy

Softwarové systémy představují skupinu uživatelských rozhraní – souborů nástrojů, které jsou uživateli používány k ovládní, programování a komunikaci se samotnými softwary. Z těchto softwarových systémů práce představuje 2 nejznámější skupiny.

Mastercam + Robotmaster – je kombinace dvou softwarových nástrojů sloužících pro CAD/CAM programování. Pomocí nástroje Mastercam lze definovat, korigovat i simulovat modely součástí a dráhy nástroje, zatímco nastavení Robotmaster pak slouží k vytvoření potřebného kódu [35].



Obr. 3.20 Ukázka prostředí Mastercam [25]

Mastercam funguje jako modulový systém, kdy má každý uživatel možnost konfigurace modulů pro vlastní potřeby. Jedná se o současně nepoužívanější CAD/CAM software s téměř trojnásobným množstvím prodaných licencí oproti jeho nejbližšímu konkurentovi. Celý systém je složen z 6 základních modulů [25].

- **Mastercam Design** – slouží jako 2D i 3D CAD modelář, ve kterém lze modelovat drátové geometrie, plochy i objekty s pomocí asociativního kótování [25].
- **Mastercam frézování** – je stejně jako celý systém řazen do modulů. Základem je pak frézování 2,5D kontur a kapes. Dále frézování 3D ploch různými technologiemi a víceosé frézování 5-ti i 6-ti osých strojů a robotů [25].
- **Mastercam soustružení** – nabízí možnost programovat soustružení, jak pro dvousé soustruhy, tak i pro soustruhy s poháněnými nástroji a více nástroji i nástrojovými hlavami [25].

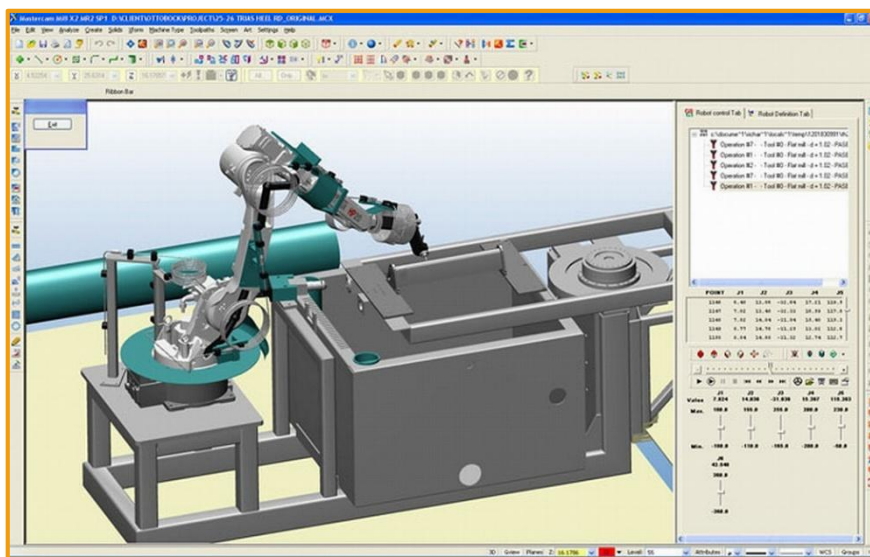


BAKALÁŘSKÁ PRÁCE

- Mastercam Router – speciální modul určený pro obrábění objektů ze dřeva. Modul se velmi podobá frézovacímu, ale je zaměřen na dřevoobrábění [25].
- Mastercam drátové řezání – poskytuje kontrolu nad pohybem drátu, úhlem, vstupy, výstupy a mnoho dalšími prvky při řezání drátem [25].
- Mastercam Art – je poslední z modulů, který je navržen pro řemeslníky využívající ke své práci umělecký talent [25].

Robotmaster pak, jak již bylo zmíněno, plní funkci nastavby pro tvorbu řídicích kódů. Robotmaster je nainstalován a plně propojen včetně ovládání s CAD/CAM systémem Mastercam. Jedná se o off-line programovací software pro programování robotů různých značek [25]. Po nastavení drah nástroje a vytvoření 3D modelů v programu Mastercam, lze následně naprogramovat simulaci robotu s možností odladění nájezdů, výjezdů a rychlostí tak, aby nedošlo ke kolizím s opracovávanými dílci nebo okolním prostředím [25, 35].

Robotmaster využívá vyspělé CAD/CAM techniky pro generování a automatizaci optimalizovaných trajektorií. Všechny tyto funkce kombinuje s vizuálním interaktivním prostředím a umožňuje flexibilitu úprav pomocí tvarování a ohýbání křivkových trajektorií [17]. Robotmaster je schopen naprogramovat většinu výrobních robotických aplikací včetně obrábění, dočišťování, broušení, leštění a samozřejmě i lakování, svařování a manipulace. Všechny tyto možnosti si plně uvědomují i přední výrobci robotů, a proto většina značek, jako KUKA, ABB, Fanuc, Stäubli či Yaskawa, oficiálně podporují software jako hlavního partnera pro programování [12, 35].



Obr. 3.21 Ukázka prostředí Robotmaster [25]

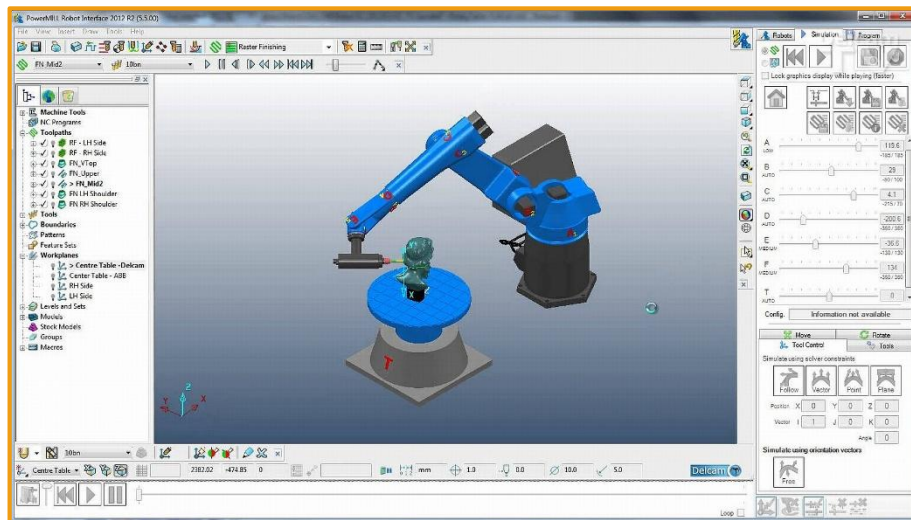
Delcam PowerMILL Robot – je software od firmy Delcam a jedná se o rozšíření či nastavbu systému PowerMILL, stejně jako je tomu u softwarů Mastercam a Robotmaster. Samotný PowerMILL představuje CAD/CAM systém pro import a výpočet drah, zatímco nastavba Robot slouží k následnému přiřazení a programování samotného robotu. Při editaci software poskytuje plnou kontrolu nad dráhou nástroje. To znamená, že mohou být odstraněny části drah nástroje, měněny vazby, vedení, případně posuvy v požadovaných oblastech bez toho, aby došlo k softwarovému přepočtu dráhy [35].

PowerMILL Robot umožňuje programovat roboty s externími osami, jako jsou otočné stoly nebo lineární koleje, které poskytují zvýšení pracovní oblasti při obrábění rozměrných dílů. Právě při obrábění velkých těžkých dílů se použití robotů považuje za velmi výhodné, díky vřetenou umístěnému na místě koncového efektoru [6].



Trojdimenzionální simulace pak zobrazuje chování robotu v nepřítomnosti programovacího systému. K dosažení požadované simulace robotu umožňuje software definovat strategii řešení a různé proměnné, jako limity jednotlivých os, roviny pracovních nástrojů a další. Eliminuje také potřebu třetí strany při překladu programového kódu do jazyku robotu. Simulace může být přeložena do rodného jazyka řady robotů od společností KUKA, ABB, Fanuc, Stäubli a Motoman [6, 35].

Systém obsahuje řadu analytických nástrojů sloužících k optimalizaci operací robotu. Pro lepší pochopení toho, jak se bude robot pohybovat, jsou v softwaru obsaženy pokročilé grafy zobrazující zrychlení, zpomalení a jiné vlastnosti v závislosti na čase [6]. PowerMILL Robot je stejně jako předchozí softwary složen z jednotlivých modulů, které byly přizpůsobeny pro jednotlivé aplikace, jakými jsou například obrábění, frézování, řezání laserem i plasmou a další [6, 35].



Obr. 3.22 Ukázka prostředí PowerMILL Robot [6]

Dalšími CAD/CAM softwarovými systémy jsou:

- IRBCAM,
- Eureka Robot,
- SprutCAM,
- Fastsimu,
- a další.

4 Manipulační úloha

Proces manipulace je ve výrobě často podceňován jako technicky jednoduchý, případně triviální, avšak z časového hlediska tomu tak zdaleka není. Výrobní čas lze rozdělit na časy strojní a manipulační, přičemž oba tyto časy mají zásadní vliv na produktivitu práce a její ekonomické ohledy. Plánování výroby je proto zaměřeno na jejich synchronizaci [34].

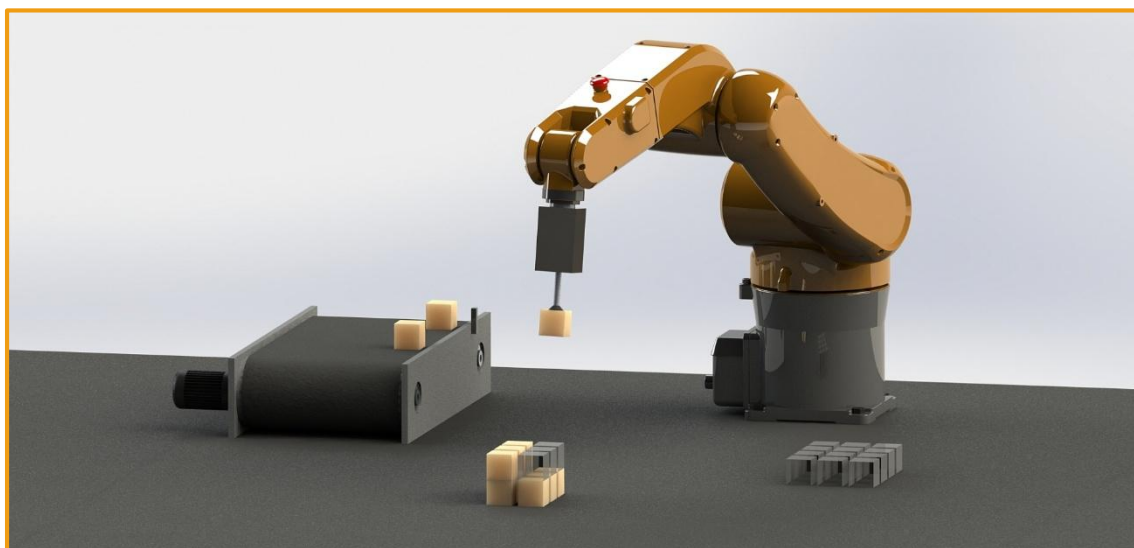
Manipulační proces může být v podstatě charakterizován jako počet přesunutých objektů za jednotku času. Přesun součásti z bodu A do bodu B se však může stát velmi komplexním a složitým procesem, který není radno podceňovat. Z praktických zkušeností v automatizaci je známo, že se nečekané technické a ekonomické problémy objevují zejména jako následek nedostatečné analýzy a vyhodnocení manipulačních procesů v rané fázi [34].

Jak již bylo několikrát zmíněno, manipulační úloha je praktickou částí této bakalářské práce. Byla zpracována na základě informací z manuálů, rad i připomínek vedoucího práce a poznatků z úvodní rešeršní části. Cílem bylo vytvoření programu v průmyslovém jazyce KRL pro průmyslového robota od firmy KUKA.

4.1 Zadání a popis úlohy

Manipulační úloha je zaměřena na paletizaci a depaletizaci předmětů, které jsou zde představovány platovými krychlemi s délkami hran 40 mm. Původním zadáním bylo efektivní přemístění 8 kostek ze vstupního rozložení $2 \times 2 \times 2$ na výstupní tvar 4×2 pomocí podtlakového chapadla. Avšak z vlastní iniciativy a se schválením bylo zadání rozšířeno o 4 další kostky na výsledných 12. Paletizační část úlohy tak představuje přemístění kostek z pásového dopravníku snímaného optickým senzorem a jejich poskládání do výsledného tvaru $3 \times 2 \times 2$. Depaletizační část pak představuje přeskládání z tohoto pomyslného paletového uspořádání do lépe přístupného tvaru 4×3 .

Úloha měla být strukturovaná do podprogramů a řešena cyklicky s použitím startovního a koncového bodu. Dále měla obsahovat startovací tlačítko a časovače pro zjištění času celého cyklu i čistého manipulačního času bez pomocných pohybů.



Obr. 4.1 Zobrazení manipulační úlohy (robot [28])



Zpracovaná manipulační úloha je tak vzdálenější interpretací skutečného procesu. Krychle putující po dopravníku zde mohou představovat obrobené či jinak opracované součásti, které jsou paletizovány kvůli přesunu na jiné pracoviště, kde jsou následně rozloženy do přístupnějšího tvaru pro následné hromadné odmašťování, lakování či jiný technologický proces.

4.2 Robotizované pracoviště

Robotizované pracoviště, jehož fotografie je umístěna v příloze č. 1, se skládá z:

- průmyslového robotu KUKA KR3 – starší verze současně vyráběných robotů se šesti stupni volnosti od firmy KUKA,
- teach-pendantu a řídicího systému KUKA,
- podtlakového chapadla se snímačem podtlaku,
- pneumatického kompresoru,
- osobního počítače,
- pásového dopravníku s optickým senzorem.



Obr. 4.2 Detail podtlakového chapadla

4.3 Proměnné, vstupy a výstupy

Vstupy jsou v řídicím systému označeny jako $\$IN[x]$, kde písmeno „x“ představuje čísla přiřazená k jednotlivým vstupům. Mají binární tvar a nabývají hodnot 0 nebo 1 (FALSE nebo TRUE). Do úlohy byly zahrnuty celkem čtyři vstupní proměnné.

- $\$IN[19]$ – analogové tlačítko pro spuštění paletizace, které je v programu označeno jako TLACITKO_P (TRUE = spuštění procesu paletizace).
- $\$IN[22]$ – analogové tlačítko pro spuštění depaletizace, které je v programu označeno jako TLACITKO_D (TRUE = spuštění procesu depaletizace).
- $\$IN[18]$ – podtlakový snímač poskytující informaci o uchopení objektu, který je v programu označen jako SMIMAC_V (TRUE = aktivní podtlak).
- $\$IN[161]$ – optický senzor na dopravníku, který je v programu označen jako DOPR (TRUE = zaznamenám objekt pro manipulaci).

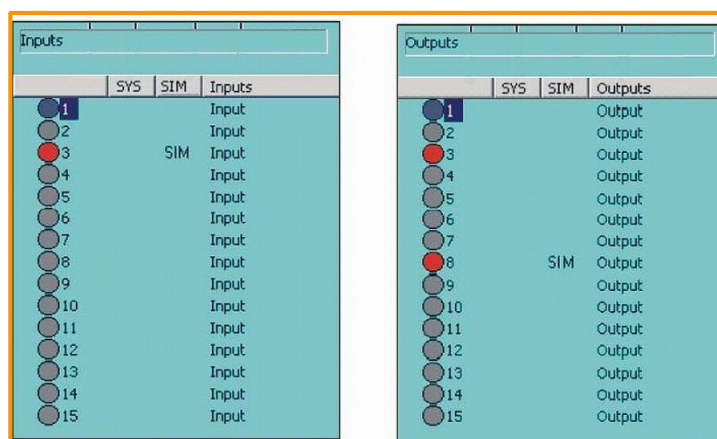
Výstupy jsou pak v řídicím systému označeny jako $\$OUT[x]$, kde písmeno „x“ představuje číslo jako v předešlém případě. Tvar je opět binární, tudíž i hodnoty, kterých mohou výstupy nabývat.

- $\$OUT[1]$ – výstup č. 1 pro polohu rozvaděče vzduchu u kompresoru
- $\$OUT[2]$ – výstup č. 2 pro polohu rozvaděče vzduchu u kompresoru

V případě kdy se $\$OUT[1] = FALSE$ a $\$OUT[2] = TRUE$ je vzduch nasáván, čímž se vytváří podtlak a dochází k uchopení kostky. V opačném případě, při nastavení výstupů na $\$OUT[1] = TRUE$ a $\$OUT[2] = FALSE$, je podtlak neaktivní (není nasáván vzduch) a dochází k uvolnění manipulované kostky.

Všechny **proměnné** v řídicím systému musí být řádně deklarovány, a jelikož jsou hodnoty proměnných celá čísla, deklarace je zajištěna příkazem INT¹³. Názvy proměnných pak nesmí mít stejné názvy jako systémové funkce nebo proměnné a jejich názvy nesmí být v rozporu s programovacím jazykem [14].

- **ST** – proměnná označující počet kroků (manipulovaných objektů) v programu, která nabývá hodnot 1–13.
- **TASK** – slouží k výběru manipulačního procesu a nabývá hodnot 0–2 (hodnota 1 – proces paletizace, hodnota 2 – proces depaletizace).
- **COLUMN2** – proměnná procesu paletizace i depaletizace, která nabývá hodnot 1–3 a přiřazuje souřadnice objektům v ose X pro uspořádání 3×2×2.
- **LINE2** – proměnná procesu paletizace i depaletizace, která nabývá hodnot 1–2 a přiřazuje souřadnice objektům v ose Y pro uspořádání 3×2×2.
- **LEVEL2** – proměnná procesu paletizace i depaletizace, která nabývá hodnot 1–2 a přiřazuje souřadnice objektům v ose Z pro uspořádání 3×2×2.
- **COLUMN3** – proměnná procesu depaletizace, která nabývá hodnot 1–4 a přiřazuje souřadnice objektům v ose X pro uspořádání 4×3.
- **LINE3** – proměnná procesu depaletizace, která nabývá hodnot 1–3 a přiřazuje souřadnice objektům v ose Y pro uspořádání 4×3.



Obr. 4.3 Ukázka zobrazení vstupů a výstupů na teach-pendantu [15]

Pro zjištění potřebných časů pak byly v řídicím programu použity čtyři časovače. Jejich deklarace je provedena pomocí zápisu `$TIMER_STOP[x] = TRUE` a inicializace pomocí `$TIMER[x] = 0`, která nastavuje jejich počáteční hodnotu na nulu. Spuštění časovače se provádí nastavením časovače na hodnotu FALSE a vypnutí nastavením hodnoty TRUE. Písmeno „x“ představuje čísla jednotlivých časovačů.

- `$TIMER_STOP[1]` – celkový čas paletizace
- `$TIMER_STOP[2]` – čistý manipulační čas paletizace
- `$TIMER_STOP[3]` – celkový čas depaletizace
- `$TIMER_STOP[4]` – čistý manipulační čas depaletizace

¹³ INTeger = celé číslo



4.4 Procesy a podprogramy

Hlavní program je složen ze dvou procesů (paletizace, depaletizace) a dalších podprogramů, které definují posloupnost a chod programu. Grafické znázornění je uvedeno v příloze č. 2 a samotný tvar jednotlivých procesů a podprogramů lze vidět ve výsledném řídicím programu uvedeném v příloze č. 4 a 5.

Paletizace() – jeden ze dvou procesů manipulační úlohy. Skládá se z podmíněného příkazu IF a cyklu FOR. Oba tyto příkazy zajišťují správnou posloupnost podprogramů a jejich opakování pro každou ze 12 použitých kostek. Robot čeká na optický senzor zaznamenávající kostku připravenou k přemístění. Poté objekty postupně přesouvá z pásového dopravníku to výsledného tvaru $3 \times 2 \times 2$.

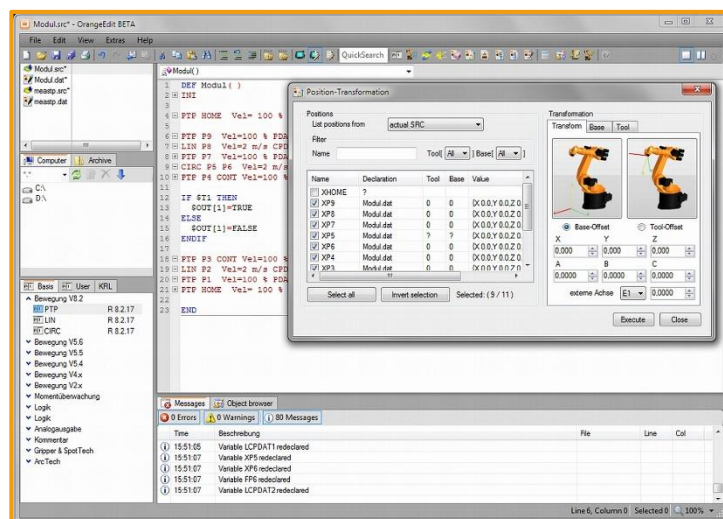
Depaletizace() – druhý hlavní proces, který se od prvního liší jen počáteční inicializací proměnných, čítači a posloupností podprogramů. Slouží k přeskládání kostek z výchozího tvaru $3 \times 2 \times 2$ do výsledného tvaru 4×3

Vypocet1() a vypocet2() – jsou podprogramy zajišťující výpočet a zápis jednotlivých souřadnic pro nájezdové body (POINT2 a POINT3) do proměnných COLUMN2, LINE2, LEVEL2 a COLUMN3, LINE3. Výpočty zajišťuje kombinace FOR cyklů, která v závislosti na zmíněných proměnných přičítá inkrementy délky k souřadnicím startovního a koncového bodu (ST_P a END_P).

Najezd1() – slouží k pohybu do bodu POINT2 a výběru jeho souřadnic v uspořádání $3 \times 2 \times 2$. Nájezd je realizován pohybem PTP, který byl vysvětlen v úvodní části. Výběr souřadnic je pak realizován sérií podmíněných příkazů IF v závislosti na aktuálním procesu. K počáteční hodnotě proměnné COLUMN2 je postupně přičítáno číslo jedna do doby, než dojde k naplnění podmínky, která změní hodnotu proměnné LINE2 případně LEVEL2. Souřadnice jsou zapsány v proměnných ve tvaru POINT2 [COLUMN2, LINE2, LEVEL2].

Najezd2() – slouží k pohybu do bodu POINT3 a výběru jeho souřadnic v uspořádání 4×3 . Nájezd a výběr souřadnic je realizován stejně jako v předchozím případě. Souřadnice jsou zapsány v proměnných ve tvaru POINT3 [COLUMN3, LINE3].

Uchopení() a uvolnění() – jsou podprogramy zajišťující uchopení a uvolnění manipulovaných kostek. Skládají se ze z pohybů LIN_REL, které umožňují pohyb robotu jen v jedné ose (Z). Samotné uchopení a uvolnění je realizováno nastavením hodnot výstupů pro rozvaděč vzduchu, jak bylo popsáno v předchozí podkapitole.



Obr. 4.4 Ukázka prostředí OrangeEdit 2.0 [18]

4.5 Tvorba a funkce řídicího programu

Při tvorbě řídicího programu bylo nejdříve zapotřebí vytvoření předběžné struktury programu se zjednodušeným vývojovým diagramem obsahujícím základními prvky, který je vhodný zejména u strukturovaných programů s více větvemi a podprogramy. Napomáhá tak ke snadné přehlednosti i orientaci při dalším programování a je uveden v příloze č. 3. Sktruktura programu byla následně převedena do programovacího prostředí OrangeEdit (Obr. 4.4), které slouží k off-line editaci řídicích programů v jazyce KRL. Struktura byla rozčleněna do jednotlivých podprogramů a dále upravována.

```

1
2 DEF my_program( )
3   INI
4
5   PTP HOME Vel= 100 % DEFAULT
6   ...
7   LIN point_5 CONT Vel= 2 m/s CPDAT1 Tool[3] Base[4]
8   ...
9   PTP point_1 CONT Vel= 100 % PDAT1 Tool[3] Base[4]
10  ...
11  PTP HOME Vel= 100 % DEFAULT
12
13  END

```

Obr. 4.5 Ukázka struktury programu [15]

Dále pak díky dvěma procesům (paletizace a depaletizace) byl program během tvorby rozdělen do dvou větví, kde každá větev představuje jeden proces. Výběr mezi jednotlivými procesy je realizován příkazem LOOP, který vytvoří opakující se smyčku. Smyčka se opakuje, dokud obsluha nestiskne příslušné tlačítko. Stisknutím tlačítka (TLACITKO_P nebo TLACITKO_D) dojde k zápisu hodnoty do proměnné TASK a opuštění opakující se smyčky. Následují podprogramy pro výpočet souřadnic a v závislosti na hodnotě proměnné dojde pomocí příkazu SWITCH ke spuštění jednotlivých procesů (TLACITKO_P pro TASK = 1 a paletizaci nebo TLACITKO_D pro TASK = 2 a depaletizaci).

Po rozvětvení a následné úpravě, která obsahovala zápis deklarací, inicializací a všech ostatních náležitostí, byl program přenesen do řídicího systému robotu. Úloha byla programována v základním souřadném systému World a centrální bod nástroje bylo nutné posunout o hodnotu 175 mm v ose Z. V on-line režimu pak bylo zapotřebí odstranit příslušné chyby, které souviseli hlavně s globálními a lokálními deklaracemi proměnných. Následovalo nastavení souřadnic základních bodů spolu se vstupy a výstupy. Jednotlivé nájezdové polohy se nachází vždy 100 mm nad manipulovanými kostkami, které jsou od sebe vzdáleny 10 mm. Po eliminaci chyb byl program spuštěn a odstraněním přebytečných přejezdů optimalizován do výsledné podoby uvedené v přílohách č. 4 a 5.

Program tak slouží k efektivnímu přemístění objektů z pásového dopravníku, nebo k jejich přeskládání do přístupnější polohy v závislosti na tom, které tlačítko bude na počátku stišťeno. Na závěr byly do programu připsány příkazy pro jednotlivé čítače, jejichž výsledné časy při spuštění programu v automatickém režimu jsou zapsány v následující tabulce.

Tabulka 1 Výsledné časy manipulační úlohy

Jednotlivé časy	[ms]	[s]
Celkový čas paletizace	42480	42,48
Celkový čas depaletizace	37392	37,392
Čistý manipulační čas paletizace	19224	19,224
Čistý manipulační čas depaletizace	18996	18,996



Závěr

Cílem této bakalářské práce bylo v její první části analyzovat současné možnosti programování průmyslových robotů s ohledem na on-line, off-line i CAD/CAM aplikace. V druhé části následně na základě teoretické pasáže a informací z manuálů vypracovat obecný algoritmus řešení manipulační úlohy spolu s funkční demonstrační aplikací.

Úvodní pasáž práce je věnována základním pojmům ze světa průmyslové robotiky. Mimo ekonomického hlediska zde byly popsány i běžně využívané struktury a nejčastěji k manipulaci používané typy průmyslových robotů. Spolu se základními druhy pohybů a souřadnými systémy tak tato část vytváří komplexní podklad nejen pro praktickou část práce.





Jednotlivé možnosti programování průmyslových robotů byly dále rozebrány v části následující. On-line programování umožňující přímou interakci člověka s robotem je vnímáno jako nejpřirozenější způsob výuky, a i přes postupně se rozvíjející off-line softwary, jejichž výběr byl součástí analýzy, hraje při programování stále hlavní roly. Nelze však opomenout i off-line programovací CAD/CAM softwary zaměřené na obrábění a neustále se rozvíjející oblast kooperace robot/člověk, která v sobě skrývá velký potenciál a jeví se do budoucna jako velmi perspektivní.

Každá programovací metoda má své vlastní výhody, nevýhody i aplikační domény, jež byly popsány v rámci analýzy jednotlivých možností. Vhodného optimálního řešení je pak možno dosáhnout pomocí programování hybridního, které využívá výhod obou hlavních programovacích metod (on-line a off-line). Z komfortního hlediska je hlavním cílem všech způsobů co nejvíce uživatelsky příjemný proces, a v konečném shrnutí tak volba programovací metody závisí hlavně na konkrétní aplikaci robotu a zkušenostech programátora.

Závěrečná pasáž je věnována praktické části této práce, jejímž výstupem je řídicí program ve specifickém programovacím jazyku KRL firmy KUKA. Manipulační úloha byla zaměřena na paletizační a depaletizační procesy, které bývají ve výrobě často podceňovány. Úloha zahrnuje přemístování objektů (plastových krychlí) za pomoci podtlakového chapadla. Představuje tak zjednodušenou a vzdálenější interpretaci skutečného manipulačního procesu, u kterého je čas stěžejním faktorem. Z toho důvodu byly do programu přiřazeny časovače pro čisté manipulační a celkové časy jednotlivých procesů.

Z porovnání výsledných časů lze konstatovat, že paletizace představuje z obou časových hledisek delší proces, což je převážně způsobeno delšími drahami robotického ramene. Celkový čas paletizačního procesu však nelze považovat za směrodatný, protože je značně zkreslen dobou, kdy robot čekal na objekt připravený k manipulaci.

Úlohu lze ve výrobním procesu dále optimalizovat dalším zkrácením pomocných nájezdů, přiblížením odkládacích poloh, případně lepším umístěním dopravníku, což nebylo v rámci robotizovaného pracoviště při zpracování úlohy možné.

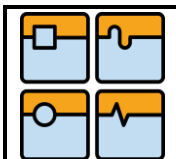
 	Ústav výrobních strojů, systémů a robotiky	Str. 38
 	BAKALÁŘSKÁ PRÁCE	

Seznam použitých zdrojů

- [1] ABB. *ABB Robotika* [online]. 2015 [cit. 2015-05-14]. Dostupné z: <http://new.abb.com/products/robotics/cs?gelid=CI7Zo5KZwcUCFagfwwod9HoAOw>
- [2] *Automatizace a robotizace I.* [online]. Učební text SPŠ a JŠ Kolín, [2011], 74 s. [cit. 2015-05-13]. Dostupné z: http://www.sps-ko.cz/documents/ARO_prorok/Pr%C5%AFmyslov%C3%A9%20roboty.pdf
- [3] *Automatizačná a robotická technika* [online]. Učební text VITRALAB Košice, 2011, 64 s. [cit. 2015-05-13]. Dostupné z: http://www.sjf.tuke.sk/vitralab/files/vystupy/prirucka_sk_final.pdf
- [4] BLECHA, P., T. NOVOTNÝ, A. POCHYLÝ a S. SIMEONOV. Potenciál technologie virtuální reality a simulací pro výrobní techniku [online]. *Svět strojírenské techniky: časopis svazu strojírenské technologie*. Praha: SST, 2010, (6): 14-16. ISSN 1803-5736. Dostupné z: http://www.sst.cz/download/pdf/svet_stroj_tech201006_complete.pdf
- [5] BLECHA, P., Z. KOLÍBAL, R. KNOFLÍČEK, A. POCHYLÝ, T. KUBELA, R. BLECHA a T. BŘEZINA. *Mechatronika: Modul10: Robotika* [on-line]. Učební text VUT Brno, 157 s. [cit. 2015-05-13]. Dostupné z: <http://www.adam-europe.eu/prj/3810/prd/1/7/Modul%2010%20Tschechisch%20komplett.pdf>
- [6] DELCAM. *Advanced manufacturing solutions: PowerMILL Robot* [online]. 2015 [cit. 2015-05-14]. Dostupné z: <http://www.delcam-robotics.com/>
- [7] FANUC. *Průmyslové roboty FANUC* [online]. 2015 [cit. 2015-05-14]. Dostupné z: <http://www.fanuc.eu/cz/cs/robots>
- [8] HAJDUK, M., V. BALÁŽ a N. DANESHJO. Simulácia a off-line programovanie priemyselných robotických systémov [online]. *AT&P journal: priemyselná automatizácia a informatika*. Bratislava: HMH, 2005, (2): 84-85. ISSN 1336-233X. Dostupné z: <http://www.idbjournal.sk/buxus/docs/atp-2005-02-84.pdf>
- [9] HAJNÝ, P. *Návrh robotické buňky pro manipulační operace*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2014. 55 s. Vedoucí bakalářské práce Ing. Aleš Pochylý.
- [10] Industrial Robot Statistic. *International federation of robotics* [online]. 2014 [cit. 2015-05-13]. Dostupné z: <http://www.ifr.org/news/ifr-press-release/accelerating-demand-for-industrial-robots-will-continue-658/>
- [11] KABEŠ, Karel. Průmyslové roboty překonávají všechny rekordy. *Automa: časopis pro automatizační techniku*. Praha: FCC Public, 2015, (2): 44-45. ISSN 1210-9592. Dostupné také z: <http://automa.cz/res/pdf/53440.pdf>
- [12] KREJČIŘÍKOVÁ, Lucie. Programování robotů tématem konference v Lisabonu. *MM Průmyslové spektrum* [online]. Praha: Vogel Publishing, 2013, (3): 34 [cit. 2015-05-14]. ISSN 1212-2572. Dostupné z: <http://www.mmspektrum.com/clanek/programovani-robotu-tematem-konference-v-lisabonu.html>
- [13] KUKA ROBOTER CEE GMBH. *KUKA Robotics* [online]. 2015 [cit. 2015-05-13]. Dostupné z: http://www.kuka-robotics.com/czech_republic/cs/



- [14] KUKA ROBOTER GMBH. *KUKA KR C2 / KR C3: Expert programming. 5.2.* Augsburg, 2006, 203 s. [cit. 2015-05-13].
- [15] KUKA ROBOTER GMBH. *KUKA Operating and Programming Instructions: For End User, KSS V5.2, V5.3, V5.4. 1.1.* Augsburg, 2006, 124 s. [cit. 2015-05-13].
- [16] LEGNANI, Giovanni a Irene FASSI. *Robotics: state of the art and future trends* [online]. New York: Nova Science Publishers, 2012, viii, 362 p. [cit. 2015-05-13]. Computer science, technology and applications. ISBN 9781621004783. Dostupné z: ebrary ProQuest
- [17] NETOPIIL, Jiří. Software boří bariéry v programování robotů. *MM Průmyslové spektrum* [online]. Praha: Vogel Publishing, 2014, (3): 42 [cit. 2015-05-14]. ISSN 1212-2572. Dostupné z: <http://www.mmspektrum.com/clanek/software-bori-bariery-v-programovani-robotu.html>
- [18] ORANGEAPPS. *OrangeEdit, ObjectBrowser* [online]. 2015 [cit. 2015-05-14]. Dostupné z: <http://www.orangeapps.de/?lng=en&page=apps>
- [19] PIRES, J. *Industrial robots programming: building applications for the factories of the future.* New York: Springer, 2007, xiv, 282 s. ISBN 03-872-3325-3.
- [20] Poptávka po průmyslových robotech roste. *TechMagazín: měsíčník pro technické obory, vědu, výzkum, strojírenství, plastikářský a automobilový průmysl, IT a technické školství.* Praha: Tech Media Publishing, 2015, (2): 50. ISSN 1804-5413. Dostupné také z: http://www.techmagazin.cz/ke_stazeni/TM022015.pdf
- [21] RAHIMI, Mansour a Waldemar KARWOWSKI. *Human-robot interaction* [online]. Washington, DC: Taylor, 1992, viii, 378 p. [cit. 2015-05-13]. ISBN 0-203-21093-X. Dostupné z: ebrary ProQuest
- [22] RobCAD. *Siemens PLM Software* [online]. 2015 [cit. 2015-05-14]. Dostupné z: http://www.plm.automation.siemens.com/cz_cz/products/tecnomatix/robotics_automation/robcad/
- [23] RUMÍŠEK, Pavel. *Automatizace: roboty a manipulátory* [online]. Učební text FSI VUT Brno, 2003, 31 s. [cit. 2015-05-13]. Dostupné z: http://ust.fme.vutbr.cz/tvareni/opory_soubory/mechanizace_a_automatizace__roboty__rumisek.pdf
- [24] SCHMID, Dietmar. *Řízení a regulace pro strojírenství a mechatroniku.* 1. vyd. Praha: Europa – Sobotáles, 2005, 420 s. ISBN 80-867-0610-9.
- [25] SONETECH. *Mastercam, Robotmaster* [online]. 2012 [cit. 2015-05-14]. Dostupné z: <http://www.sonetech.cz/>
- [26] ŠMÍD, Jiří. SPARC a roboty na Automatica 2014. *MM Průmyslové spektrum* [online]. Praha: Vogel Publishing, 2014, (7): 60 [cit. 2015-05-14]. ISSN 1212-2572. Dostupné z: <http://www.mmspektrum.com/clanek/sparc-a-roboty-na-automatica-2014.html>
- [27] ŠOLC, František a Luděk ŽALUD. *Robotika* [online]. Učební text FEKT VUT Brno, 2002, 61 s. [cit. 2015-05-13]. Dostupné z: <http://media1.wgz.cz/files/media1:5100dca52f8f1.pdf.upl/Robotika.pdf>
- [28] KUKA KR-5 R850: KUKA 6-axis robotic arm. In: TJIU, Sugi. *GrabCAD* [online]. 2011 [cit. 2015-05-19]. Dostupné z: <https://grabcad.com/library/kuka-kr-5-r850#>

	Ústav výrobních strojů, systémů a robotiky	Str. 40
	BAKALÁŘSKÁ PRÁCE	

- [29] TOCHÁČEK, Daniel a Jakub LAPEŠ. *Edukační robotika* [online]. 1. vyd. Praha: Univerzita Karlova, Pedagogická fakulta, 2012, 52 s. [cit. 2015-05-13]. ISBN 978-80-7290-577-5. Dostupné z: https://kraken.pedf.cuni.cz/~lapej2ap/robo/skripta_edurobo.pdf
- [30] TRNKA, Kamil a Pavol BOŽEK. Simulácia a off-line programovanie robotov. *Strojárstvo* [online]. Žilina: MEDIA/ST, 2011, (12): 82-84 [cit. 2015-05-14]. ISSN 1335-2938. Dostupné z: <http://www.floowie.com/sk/citaj/stroj12-el/#/strana/82/zvacsenie/100/>
- [31] VELEBIL, Radek. Nová éra robotiky přichází. *Strojárstvo* [online]. Žilina: MEDIA/ST, 2014, (9): 58-59 [cit. 2015-05-14]. ISSN 1335-2938. Dostupné z: <http://www.floowie.com/sk/citaj/stroj9-2014-el/#/strana/58/zvacsenie/100/>
- [32] VELÍŠEK, Karol, Branko KATALINIČ a Angela JAVOROVÁ. *Priemyselné roboty a manipulátory*. 1. vyd. Bratislava: Vydavateľstvo STU, 2006, 183 s. ISBN 80-227-2492-0.
- [33] W.A.T. SOLUTIONS. *Workspace: Robot simulation* [online]. 2015 [cit. 2015-05-14]. Dostupné z: <http://www.workspaceit.com/>
- [34] WOLF, Andreas, Ralf STEINMANN a Henrik SCHUNK. *Grippers in motion: the fascination of automated handling tasks*. Berlin: Springer, 2005, 248 p. ISBN 35-402-5657-1.
- [35] ZAPLETAL, J. *Simulační nástroje pro CAD/CAM programování průmyslových robotů*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2014. 48 s. Vedoucí bakalářské práce Ing. Aleš Pochylý.

Seznam obrázků

Obr. 1.1 Příklad robotizovaného pracoviště [13]	11
Obr. 2.1 Struktura průmyslového robotu [3]	12
Obr. 2.2 Schéma robotu 6DOF [13]	13
Obr. 2.3 Schéma robotu SCARA [13]	13
Obr. 2.4 Schéma paletizačního robotu [5]	13
Obr. 2.5 Souřadnicové systémy [14]	14
Obr. 2.6 Vzájemné vztahy souřadných systémů [15]	14
Obr. 2.7 Pohyb CIRC [15]	15
Obr. 2.8 Pohyb LIN [15]	15
Obr. 2.9 Pohyb PTP [15]	15
Obr. 3.1 Postup on-line programování [8]	16
Obr. 3.2 Teach-pendant firmy ABB [1]	17
Obr. 3.3 Teach-pendant firmy KUKA [13]	17
Obr. 3.4 Teach-pendant firmy Fanuc [7]	17
Obr. 3.5 Programování pomocí teach-pendantu [3]	18
Obr. 3.6 Jednotlivá funkční tlačítka teach-pendantu [3]	19
Obr. 3.7 Robot KUKA LBR iiwa [13]	20
Obr. 3.8 Robot ABB YuMi [1]	20
Obr. 3.9 Robot Fanuc CR-35i [7]	21
Obr. 3.10 Virtuální model pracoviště v programu RobotStudio [1]	21
Obr. 3.11 Detekce kolize robotu [25]	22
Obr. 3.12 Metodika off-line programování [4]	23
Obr. 3.13 Virtuální kontrolér KUKA [18]	23
Obr. 3.14 Ukázka prostředí Kuka.Sim Pro 2.1 [9]	24
Obr. 3.15 Ukázka prostředí RobotStudio 5.6 [9]	25
Obr. 3.16 Ukázka prostředí RoboGuide [7]	26
Obr. 3.17 Ukázka prostředí RobCad 9.0 [22]	26
Obr. 3.18 Ukázka prostředí Workspace5 [33]	27
Obr. 3.19 Průběh CAD/CAM programování [6]	28
Obr. 3.20 Ukázka prostředí Mastercam [25]	29
Obr. 3.21 Ukázka prostředí Robotmaster [25]	30
Obr. 3.22 Ukázka prostředí PowerMILL Robot [6]	31
Obr. 4.1 Zobrazení manipulační úlohy (robot [28])	32
Obr. 4.2 Detail podtlakového chapadla	33
Obr. 4.3 Ukázka zobrazení vstupů a výstupů na teach-pendantu [15]	34
Obr. 4.4 Ukázka prostředí OrangeEdit 2.0 [18]	35
Obr. 4.5 Ukázka struktury programu [15]	36

Seznam grafů a tabulek

Graf 1 Roční poptávka průmyslových robotů [10] 11

Tabulka 1 Výsledné časy manipulační úlohy.....36

Seznam příloh

Příloha č. 1 Robotizované pracoviště

Příloha č. 2 Grafické znázornění procesů a podprogramů

Příloha č. 3 Vývojový diagram

Příloha č. 4 Řídicí program – papírová verze

Příloha č. 5 Řídicí program – přiložené CD (manipulacni_uloha.src, manipulacni_uloha.dat)



Ústav výrobních strojů, systémů a robotiky
BAKALÁŘSKÁ PRÁCE
ROBOTIZOVANÉ PRACOVIŠTĚ

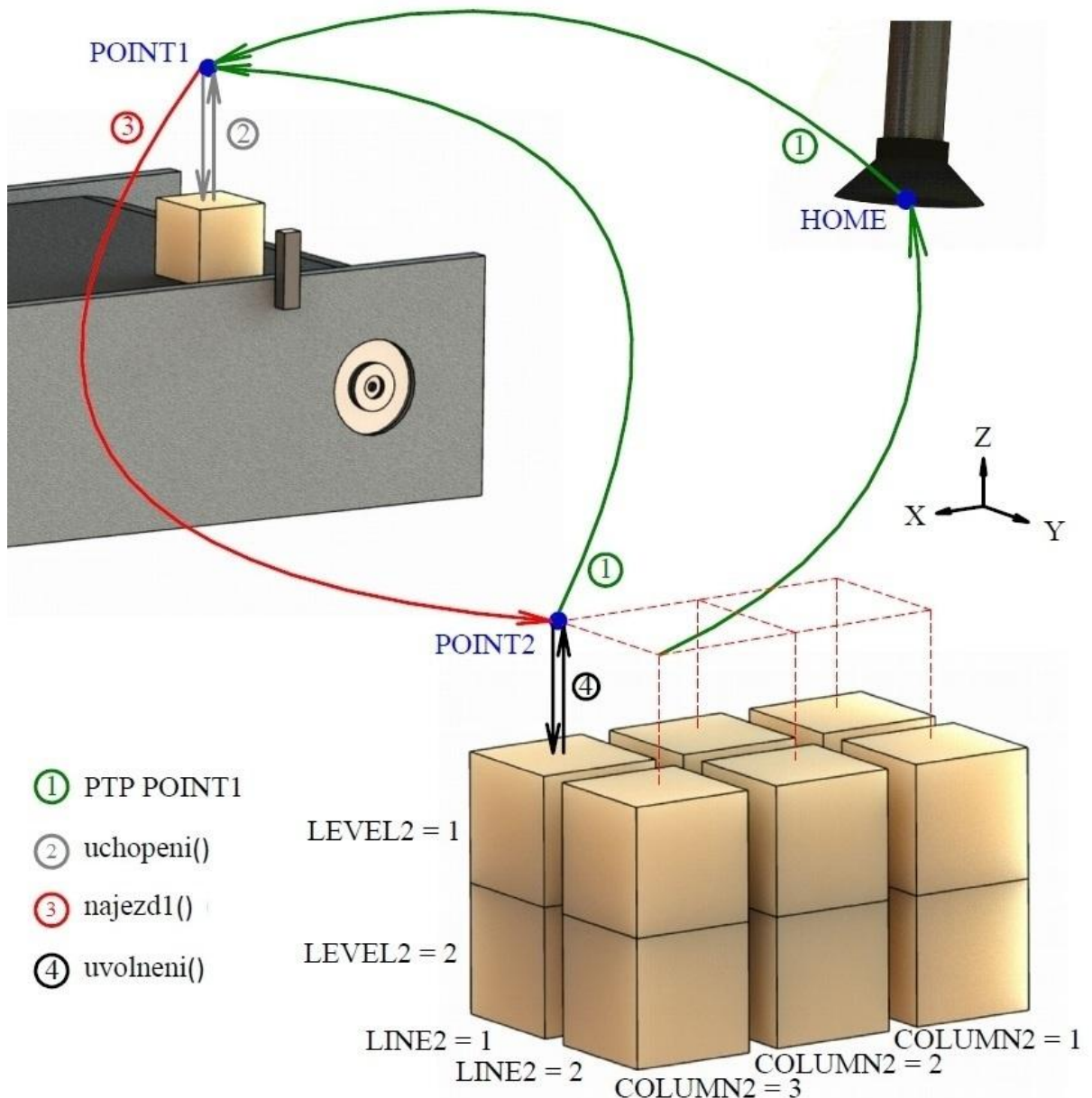
Příloha
č. 1





Paletizace

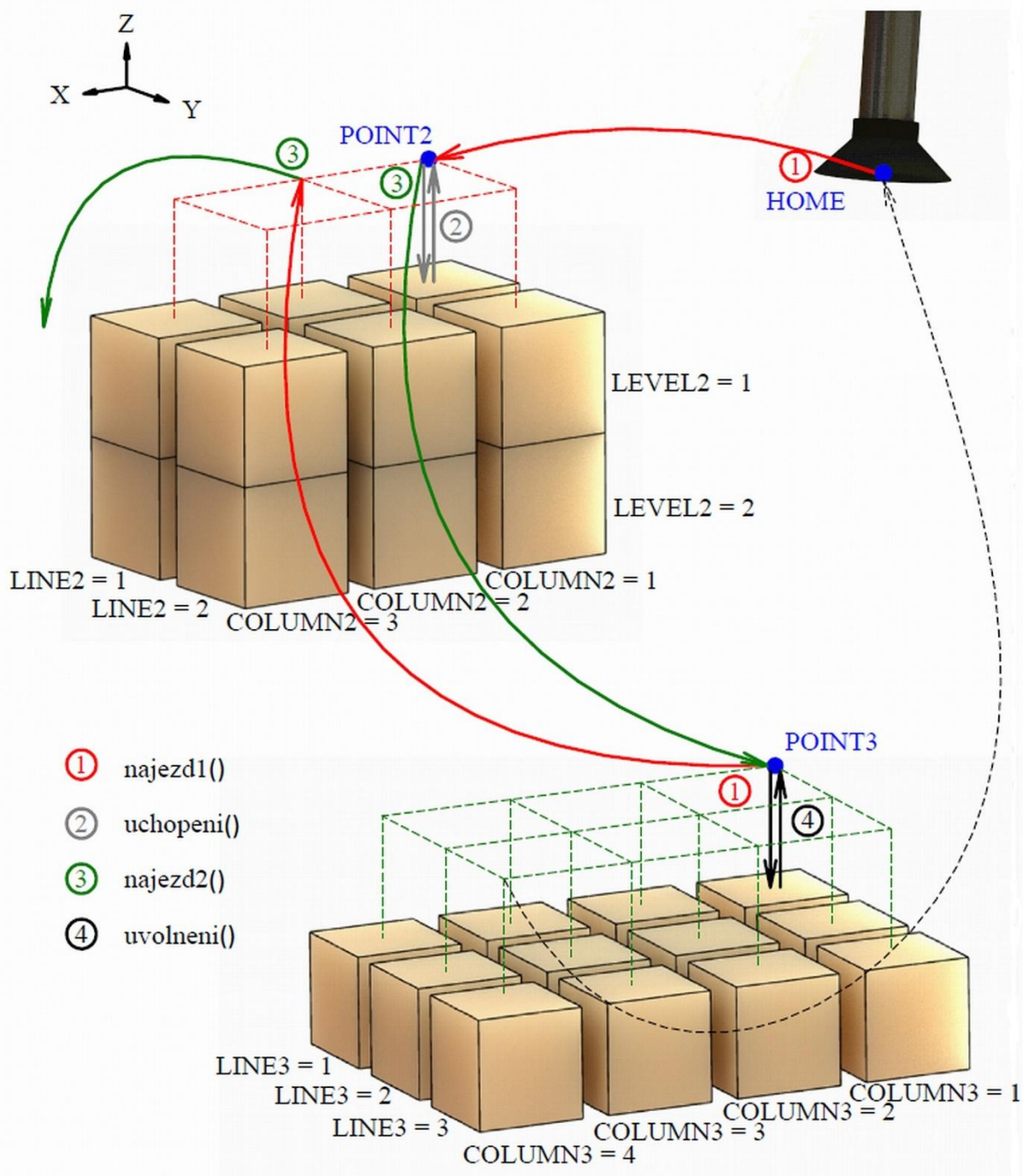
Pro zachování přehlednosti byl zakreslen a barevně rozlišen jen jeden pracovní cyklus (plná čára). Dále byly naznačeny polohy ostatních bodů (čárkovaná čára) a závěrečný odjezd do výchozího bodu.



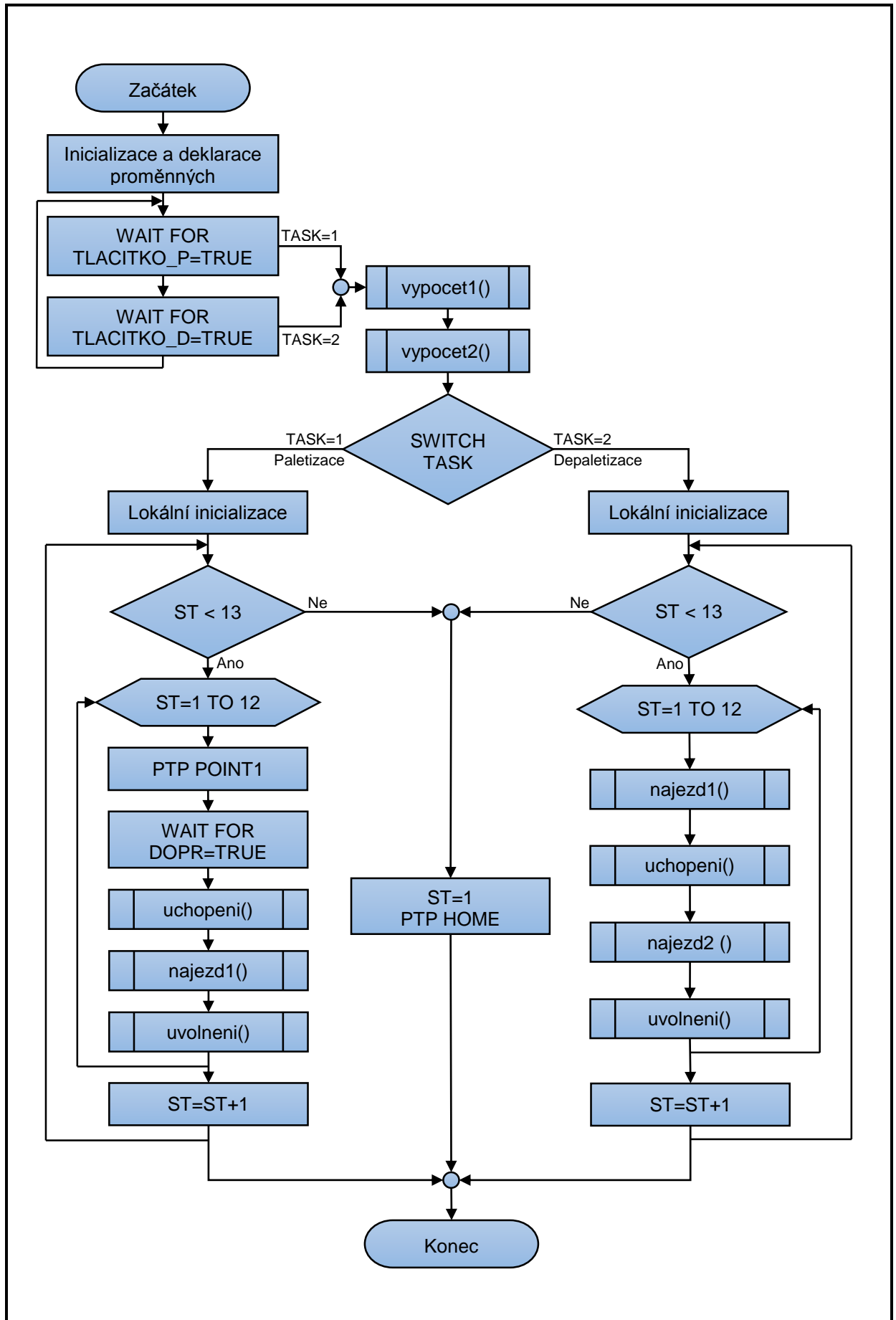
ST (krok)	1	2	3	4	5	6
POINT2	[1,1,2]	[2,1,2]	[3,1,2]	[1,2,2]	[2,2,2]	[3,2,2]
ST (krok)	7	8	9	10	11	12
POINT2	[1,1,1]	[2,1,1]	[3,1,1]	[1,2,1]	[2,2,1]	[3,2,1]



Depaletizace



ST (krok)	1	2	3	4	5	6
POINT2	[1,1,1]	[2,1,1]	[3,1,1]	[1,2,1]	[2,2,1]	[3,2,1]
POINT3	[1,1]	[2,1]	[3,1]	[4,1]	[1,2]	[2,2]
ST (krok)	7	8	9	10	11	12
POINT2	[1,1,2]	[2,1,2]	[3,1,2]	[1,2,2]	[2,2,2]	[3,2,2]
POINT3	[3,2]	[4,2]	[1,3]	[2,3]	[3,3]	[4,3]





```
DEFDAT manipulacni_uloha
```

```
EXTERNAL DECLARATIONS
```

```
SIGNAL SNIMAC_V $IN[18]
```

```
SIGNAL DOPR $IN[161]
```

```
DECL AXIS HOME_POSITION
```

```
DECL POS POINT1
```

```
DECL POS ST_P
```

```
DECL POS END_P
```

```
DECL POS POINT2[3,2,2]
```

```
DECL POS POINT3[4,3]
```

```
INT COLUMN2, LINE2, LEVEL2
```

```
INT COLUMN3, LINE3
```

```
INT TASK
```

```
INT ST
```

```
ENDDAT
```

```
DEF manipulacni_uloha()
```

```
;Deklarace a inicializace promennych
```

```
SIGNAL TLACITKO_D $IN[22]
```

```
SIGNAL TLACITKO_P $IN[19]
```

```
BAS (#INITMOV,0)
```

```
BAS (#TOOL,1)
```

```
BAS (#BASE,0)
```

```
$TIMER_STOP[1] = TRUE ;Celkovy cas paletizace
```

```
$TIMER[1] = 0
```

```
$TIMER_STOP[2] = TRUE ;Cisty manipulacni cas pal.
```

```
$TIMER[2] = 0
```

```
$TIMER_STOP[3] = TRUE ;Celkovy cas depaletizace
```

```
$TIMER[3] = 0
```

```
$TIMER_STOP[4] = TRUE ;Cisty manipulacni cas dep.
```

```
$TIMER[4] = 0
```

```
ST = 1
```

```
HOME_POSITION = {A1 0, A2 -90, A3 90, A4 0, A5 0, A6 0}
```

```
POINT1 = {X -5, Y -437, Z 217, A 180, B 0, C 180, S 2., T 2.}
```

```
ST_P = {X 360, Y -250, Z 165, A 180, B 0, C 180, S 2., T 10.}
```

```
END_P = {X 405, Y 150, Z 125, A 180, B 0, C 180}
```

```
;Hlavní program
```

```
PTP HOME_POSITION
```

```
LOOP ;Smyčka zapisující hodnotu promenne pro volbu procesu
```

```
IF TLACITKO_P==TRUE THEN
```

```
TASK = 1
```

```
$TIMER_STOP[1] = FALSE
```

```
EXIT
```

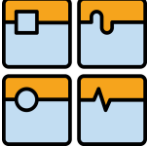
```
ELSE
```

```
IF TLACITKO_D==TRUE THEN
```

```
TASK = 2
```

```
$TIMER_STOP[3] = FALSE
```

```
EXIT
```

	Ústav výrobních strojů, systémů a robotiky	Příloha č. 4
	BAKALÁŘSKÁ PRÁCE	
	ŘÍDICÍ PROGRAM	

```

ELSE
  TASK=0
ENDIF
ENDIF
ENDLOOP

vypocet1() ;Volani podprogramu pro vypocet souradnic
vypocet2()

SWITCH TASK ;Prikaz zajistujici vyber hlavniho procesu
CASE 1
  paletizace()
CASE 2
  depaletizace()
ENDSWITCH

END

;Proces paletizace (TASK=1)
DEF paletizace()


COLUMN2 = 1 ;Pocatecni inicializace pro proces paletizace
LINE2 = 1
LEVEL2 = 2
IF ST < 13 THEN ;Hlavni cast zajistujici posloupnost a opakovani podprogramu
  FOR ST = 1 TO 12 STEP 1
    PTP POINT1
    WAIT FOR DOPR==TRUE
    uchopeni() ;Volani podprogramu
    najezd1()
    uvolneni()
  ENDFOR
  ST = ST + 1
ELSE ;Cast zajistujici reset a najezd do poc.polohy
  PTP HOME_POSITION
  $TIMER_STOP[1] = TRUE
  ST = 1
ENDIF
PTP HOME_POSITION
$TIMER_STOP[1] = TRUE

END

;Proces depaletizace (TASK=2)
DEF depaletizace()

COLUMN2 = 1 ;Pocatecni inicializace pro proces depaletizace
LINE2 = 1
LEVEL2 = 1
COLUMN3=1
LINE3=1
IF ST < 13 THEN ;Hlavni cast zajistujici posloupnost a opakovani podprogramu
  FOR ST = 1 TO 12 STEP 1
    najezd1() ;Volani podprogramu
    uchopeni()
    najezd2()
    uvolneni()
  ENDFOR
  ST = ST + 1
ELSE ;Cast zajistujici reset a najezd do poc.polohy

```

	Ústav výrobních strojů, systémů a robotiky	Příloha č. 4
	BAKALÁŘSKÁ PRÁCE	
	ŘÍDICÍ PROGRAM	

```

PTP HOME_POSITION
$TIMER_STOP[3] = TRUE
ST = 1
ENDIF
PTP HOME_POSITION
$TIMER_STOP[3] = TRUE

END

```

```

;Podprogram - nájezd1
DEF najezd1()

PTP POINT2 [COLUMN2, LINE2, LEVEL2]

;Serie prikazu zajistujici vyber souradnic pro usporadani 3x2x2 zapsanych v
jednotlivych promenych
IF (COLUMN2==3) AND (LINE2==2) THEN
    COLUMN2 = 1
    LINE2 = 1
    IF TASK==1 THEN
        LEVEL2 = 1
    ELSE
        LEVEL2 = 2
    ENDIF
ELSE
    IF COLUMN2==3 THEN
        COLUMN2 = 1
        LINE2 = 2
    ELSE
        COLUMN2 = COLUMN2 + 1
    ENDIF
ENDIF
ENDIF

END

```

```

;Podprogram - nájezd2
DEF najezd2()

PTP POINT3 [COLUMN3, LINE3]

;Serie prikazu zajistujici vyber souradnic pro usporadani 4x3 zapsanych v
jednotlivych promenych
IF COLUMN3==4 THEN
    COLUMN3 = 1
    LINE3 = LINE3 + 1
ELSE
    COLUMN3 = COLUMN3 + 1
ENDIF
ENDIF

END

```

```

;Podprogram - uchopeni
DEF uchopeni()

LIN_REL {Z -100}
$OUT[1] = FALSE ;Prisati
$OUT[2] = TRUE
WAIT FOR SNIMAC_V==TRUE

IF TASK==1 THEN

```



Ústav výrobních strojů, systémů a robotiky

BAKALÁŘSKÁ PRÁCE

ŘÍDICÍ PROGRAM

Příloha
č. 4

```
$TIMER_STOP[2] = FALSE  
ELSE  
  $TIMER_STOP[4] = FALSE  
ENDIF
```

```
LIN_REL {Z 100}
```

```
END
```

```
;Podprogram - uvolneni
```

```
DEF uvolneni()
```

```
LIN_REL {Z -100}
```

```
$OUT[1] = TRUE           ;Uvolneni
```

```
$OUT[2] = FALSE
```

```
WAIT FOR SNIMAC_V==FALSE
```

```
IF TASK==1 THEN
```

```
  $TIMER_STOP[2] = TRUE
```

```
ELSE
```

```
  $TIMER_STOP[4] = TRUE
```

```
ENDIF
```

```
LIN_REL {Z 100}
```

```
END
```

```
;Podprogram - vypocet souradnic bodu POINT2
```

```
DEF vypocet1()
```

```
FOR COLUMN2 = 1 TO 3
```

```
  FOR LINE2 = 1 TO 2
```

```
    FOR LEVEL2 = 1 TO 2
```

```
      POINT2 [COLUMN2, LINE2, LEVEL2].X = ST_P.X + ((COLUMN2 - 1) * 50)
```

```
      POINT2 [COLUMN2, LINE2, LEVEL2].Y = ST_P.Y + ((LINE2 - 1) * 50)
```

```
      POINT2 [COLUMN2, LINE2, LEVEL2].Z = ST_P.Z - ((LEVEL2 - 1) * 40)
```

```
      POINT2 [COLUMN2, LINE2, LEVEL2].A = ST_P.A
```

```
      POINT2 [COLUMN2, LINE2, LEVEL2].B = ST_P.B
```

```
      POINT2 [COLUMN2, LINE2, LEVEL2].C = ST_P.C
```

```
    ENDFOR
```

```
  ENDFOR
```

```
ENDFOR
```

```
END
```

```
;Podprogram - vypocet souradnic bodu POINT3
```

```
DEF vypocet2()
```

```
FOR COLUMN3 = 1 TO 4
```

```
  FOR LINE3 = 1 TO 3
```

```
    POINT3 [COLUMN3, LINE3].X = END_P.X + ((COLUMN3 - 1) * 50)
```

```
    POINT3 [COLUMN3, LINE3].Y = END_P.Y + ((LINE3 - 1) * 50)
```

```
    POINT3 [COLUMN3, LINE3].Z = END_P.Z
```

```
    POINT3 [COLUMN3, LINE3].A = END_P.A
```

```
    POINT3 [COLUMN3, LINE3].B = END_P.B
```

```
    POINT3 [COLUMN3, LINE3].C = END_P.C
```

```
  ENDFOR
```

```
ENDFOR
```

```
END
```