

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
FAKULTA STROJNÍHO INŽENÝRSTVÍ

**Ing. Jiří Mácha**

**POČÍTAČOVÉ ŘÍZENÍ ŠESTIOSÉHO  
ANTROPOMORFNÍHO ROBOTU**

**COMPUTER CONTROL OF SIX AXIS  
ANTHROPOMORPHIC ROBOT**

Teze disertační práce

PhD Thesis

Studijní obor: Konstrukční a procesní inženýrství  
Školitel: Doc. Ing. Pavel Bělohoubek, CSc. (ÚVSSaR FSI VUT Brno)  
Oponenti: Prof. Ing. Jaroslav Talácko, CSc. (FS ČVUT Praha)  
Doc. Ing. Jiří Skařupa, CSc. (VŠB TU Ostrava)  
Doc. Ing. Jiří Skalický, CSc. (FEI VUT Brno)  
Datum obhajoby: 6. 4. 2001

© 2001 J. Mácha

ISBN 80-214-1888-5

ISSN 1213-4198



# OBSAH

1. ÚVOD	5
2. CÍL A OBSAH PRÁCE	6
3. SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY	7
4. ZVOLENÉ METODY ZPRACOVÁNÍ	10
4.1. Výběr operačního systému	10
4.2. Výběr programovacího jazyku a knihoven	10
4.3. Výběr CAD systému pro návrh modelu	10
4.4. Vstup dat do programu	11
4.5. Programování pohybu koncového efektoru robotu	11
4.6. Řízení pohybu mezi uzlovými body	11
5. HLAVNÍ VÝSLEDKY PRÁCE	13
5.1. Odvození matematických vztahů pro realizaci kinematické analýzy, syntézy a interpolačních funkcí	13
5.2. Propojení návrhu robotizovaného pracoviště v CAD systému s řídicím programem	17
5.3. Vstupní projektový soubor	18
5.4. Realizace interpolačních funkcí	20
5.5. Výstup a zpracování dat	26
6. ZÁVĚR	28
6.1. Přínos práce pro rozvoj vědního oboru	28
6.2. Zhodnocení stávajícího stavu	29
LITERATURA	31
PUBLIKACE AUTORA	34
CURRICULUM VITAE	35
ABSTRACT	37

# 1. ÚVOD

Robot je počítačem řízený integrovaný systém, schopný autonomní cílově orientované iterace s reálným prostředím v souladu s instrukcemi člověka. V současnosti je názvosloví v oblasti robotiky ve světě značně nejednotné a z tohoto důvodu se lze setkat s různými definicemi jak robotu, tak ostatních pojmů z oblasti robotiky. Poslední dobou lze navíc pozorovat též tendence pojmenovat slovem robot software který, zajišťuje sběr nebo zpracování informací z kyberprostoru (Internet).

Robotika je oborem interdisciplinárním a tudíž klade značné požadavky na výzkumné a vývojové pracovníky. Je možné říci, že robotika spojuje obory jako mechanika, mikromechanika, elektronika, informatika, matematika, kybernetika, biologie a mnohé jiné.

Podle zaměření lze v současnosti identifikovat tyto hlavní oblasti, v nichž dochází k rychlému rozvoji:

- Výzkum vesmíru, planet - mobilní roboty, automatické meziplanetární sondy aj.
- Průmysl - průmyslové roboty a manipulátory - ulehčující práci, mechanické úkony na dílně. Důležitými oblastmi přitom je doprava (automobilová) a výroba elektroniky.
- Zdravotnictví - skenery, roboty provádějící přesné chirurgické zákroky.
- Zábava - hračky (roboty) imitující zvířata aj.
- Domácnost - roboty pro úklid domácností.
- Tržní principy (konkurence), vojenství - špiónážní roboty.

Další oblastí, u které lze v blízké budoucnosti předvídat rychlý rozvoj, je

- Nanorobotika - molekulární nanotechnologie (MNT), kde hlavními stavebními prvky „strojů“ jsou atomy.

Dá se přitom předpokládat, že některá komerčně úspěšná oblast nasazení robotů podnítl vývoj a zlevnění v oblastech jiných. Navíc poznatky získané z jedné oblasti lze většinou velmi dobře aplikovat v oblastech ostatních.

V konstrukci řídicích systémů lze pozorovat tendence pro neustálé zjednodušování obvodového zapojení řídicích systémů a přenášení vlastností, dříve realizovaných obvodovým zapojením, na software. Lze pak měnit funkční vlastnosti zařízení (stroje) pouhou výměnou programu (modulů řídicího systému) bez nutnosti měnit obvodové zapojení. Také lze pozorovat snahu nahradit analogové obvody digitálními, které přináší vyšší spolehlivost a jsou snáze ovladatelné procesorovými jednotkami řídicích systémů.

Na druhou stranu přenesení mnoha funkčních vlastností na software s sebou přináší značné požadavky na jeho kvalitu. Chyba v řídicím softwaru tak mívá katastrofální důsledky na funkčnost zařízení a mnohdy vede k jeho ztrátě. Z těchto a dalších důvodů lze proto očekávat další rozvoj a zdokonalování simulačních nástrojů a technik včetně nástrojů pro realizaci virtuální reality.

## 2. CÍL A OBSAH PRÁCE

Disertační práce je zaměřena na průmyslové roboty a manipulátory a to zejména na šestiosý antropomorfní robot, jehož hlavním rysem je schopnost prostorově polohovat a orientovat manipulovaný předmět v rámci svého pracovního prostoru. Práce přitom není zaměřena na konstrukci robotu ani na konstrukci řídicího systému, nýbrž na návrh řídicího softwaru.

- Cílem tedy je sestavit řídicí program pro řízení šestiosého antropomorfního robotu tak, aby bylo dosaženo modularity kódu, aby program využíval vlastností moderních operačních systémů, vývojových prostředků a zároveň aby jej bylo možné dále rozšiřovat o nové funkční vlastnosti. Dále pak by mělo být dosaženo co největší nezávislosti na použitém hardwaru a softwarových technologiích jiných firem.
- Dalším cílem je propojit návrh robotizovaného pracoviště a návrh kinematické struktury v CAD systému s řídicím programem tak, aby bylo dosaženo flexibility během návrhu a simulace modelu. Návrh kinematické struktury v CAD systému má přitom být univerzální a snadno použitelný pro jiné kinematické uspořádání. Zohledněna by dále měla být možnost návrhu robotizovaného pracoviště s několika roboty na pracovišti. Návrhář robotizovaného pracoviště v CAD systému by přitom neměl být omezen při tvorbě výkresové dokumentace.
- Řídicí program má být použitelný jak pro realizaci ON-LINE tak i OFF-LINE řízení a výstupní data z programu mají být použitelné jak pro analýzu tak pro zpracování jinými programy.
- Navrhnout a realizovat základní interpolační funkce pro pohyb koncového efektoru zohledňující orientace a velikost tečné rychlosti během pohybu. Dále pak ověřit výpočtovou náročnost těchto algoritmů, zejména kinematické analýzy, syntézy a interpolačních funkcí a porovnat toto s náročností grafického výstupu.
- Navrhnout způsob programování pohybu koncového efektoru, tak aby uživatel nebyl nucen znát syntaxi vyššího programovacího jazyka a program neustále kompilovat. Program by přitom měl být snadno rozšiřitelný o nové povely.
- Zhodnotit poznatky získané vývojem softwaru.

Cílem práce nebylo zabývat se nejnižšími úrovněmi řídicího softwaru, které jsou implementačně závislé na hardwaru. Je věnována jakési pomyslné střední vrstvě, která je nezbytná pro realizaci vyšších vrstev řídicího softwaru (adaptivní řízení, plánování aj.). Tyto vyšší vrstvy řídicího softwaru jsou pak značně závislé na technologii (manipulace s předměty, stříkání, svařování aj.).

### 3. SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

Problematika řízení strojů je značně obsáhlá a náročná. Naprostá většina existujících řídicích softwarů není schopna alespoň se přiblížit fyzickým možnostem stroje.

Získat přehled o aktuálním stavu vývoje řídicích systémů a CAM softwaru není jednoduché, jelikož jsou většinou strategickým vlastnictvím firem, které jim pomáhají udržet si technologický náskok.

Pro ON-LINE řídicí systémy se převážně používají RealTime operační systémy [20],[21],[46] s dobou regulační smyčky servopohonů od 1 ms do cca 30 ms [3].

Snahou přitom je přiblížit se hodnotě 1ms, kdy je dosaženo větší přesnosti vedení koncového efektoru a lepších dynamických poměrů. V lit. [46] je uvedena hierarchická struktura řídicího systému pro robot PUMA 760, kde nejnižší úroveň řídicího systému je zodpovědná za předání spočtených kloubových souřadnic jednotlivým kartám servo-řízení po sériové lince každých 28ms.

U moderních RealTime operačních systémů, které běží na procesorech kompatibilních s instrukční sadou I386 lze dosáhnout periody přepínání kontextu procesu řádově v jednotkách mikrosekundy [47],[48]. Některé firmy dokonce dodávají rozšiřující moduly ke stávajícím operačním systémům, které neomezují funkční vlastnosti operačního systému a přitom umožňují běh aplikací v reálném čase (RealTime) [48].

Komunikaci s měniči lze v současnosti již realizovat plně digitálně po optickém vlákně – SERCOS interface [50]. Nahrazuje se tak zadávání hodnoty rychlosti napětíovou úrovní  $\pm 10V$  a zpětná polohová vazba se již uzavírá na digitální inteligentní měnič. Digitálním interfacem lze dosáhnout komunikační rychlosti 2-16 Mb/s, přičemž lze do komunikační smyčky zařadit 2 – 120 měničů.

V oblasti konstrukce robotů bylo již dosaženo značné modularity a univerzality, kdy jednotlivé články robotu jsou vybaveny jak pohony tak i měniči. Robot tak lze jednoduše rekonfigurovat na jiné kinematické uspořádání [49].

Funkce kinematické analýzy lze pro otevřené kinematické řetězce poměrně snadno odvodit a realizovat. Pro zpětnou kinematickou úlohu – syntézu lze řešit analyticky pouze v ojedinělých případech a bývá obvykle značně složitá [12]. Pro některé kinematické uspořádání lze v literatuře najít odvozené vztahy pro funkce kinematické analýzy a syntézy [1],[12],[31].

Jiné práce jsou věnovány obecným numerickým metodám pro kinematickou syntézu [51], tyto jsou však značně náročné na strojový čas procesoru. Lze se setkat i s moduly pro MATLAB a jiné systémy, ty však jsou většinou zapouzdřeny do

modulů které nelze bez dostupnosti jejich zdrojových textů modifikovat. Navíc nejsou mnohdy optimalizovány z hlediska výkonosti kvůli jejich univerzálnosti.

Mnohé řídicí CNC a PLC systémy jsou sice vybaveny interpolačními funkcemi (lineární, kruhová interpolace, spline aj.) [21],[40], jedná se však o interpolaci kloubovou, která nezohledňuje kinematické uspořádání stroje. Lineární a kruhová interpolace je tak dobře použitelná pro pravoúhlé kinematické struktury realizované posuvovými jednotkami. Pro obecné kinematické uspořádání a OFF-LINE řízení je možno použít z nabízených pouze spline interpolace.

Některé publikace byly věnovány problematice návrhu jazyka a jeho sémantice pro řízení robotů. Lit. [45] popisuje jazyk COLBERT, jehož syntaxe je podobná jazyku C a který je založen na dvou konceptech – konečný stavový automat a dekompozice na konkurenční procesy. Zde lze také nalézt srovnání jazyka COLBERT s ostatními i komerčními jazyky.

Poslední dobou značně ovlivňují stavbu řídicích systémů a CAM softwaru informační technologie, které dosahují nebývalého rozvoje.

Mnohé současné běžně používané řídicí systémy strojů jsou z hlediska softwarového návrhu „uzavřené“. Tím je myšlen zejména ten fakt, že nelze snadno provádět modifikace řídicího softwaru pouhou výměnou jeho modulů. Je to dáno především tou skutečností, že v době kdy vznikaly, nebyly dostupné současné softwarové technologie. Přitom mnoho komerčních firem těží z této skutečnosti. Zákazník (výrobce stroje a mnohdy i výrobce softwaru pro danou technologii) se pak stává zcela závislý na softwaru jeho dodavatele u kterého je nucen si modifikace řídicího softwaru objednat.

Řídicí systémy jsou většinou navrženy tak, že nelze snadno realizovat adaptivní ON-LINE řízení. Příkladem může být jakýkoliv řídicí systém, který používá NC věty na svém vstupu (interpolační funkce jsou realizovány dodavatelem). Pozn. Po přijetí povelu (NC věty) pracuje řídicí systém v uzavřené interpolační smyčce a nadřazený software (CAM jádro) nemá možnost proces interpolace jakkoliv ovlivňovat (adaptovat na danou technologii).

Zákazník je tak nucen používat pro komunikaci se strojem sadu dodavatelem navržených povelů, bez možnosti jejich modifikace. Jakýkoliv řídicí software vyvinutý nad touto sadou povelů se tak stává mnohdy koncepčně nevhodný pro realizaci ON-LINE adaptivního řízení.

V současnosti lze poměrně snadno odlišit řídicí software (realizovaný dodavatelem) od softwaru CAM jádra (většinou realizovaný výrobcem stroje), jelikož většina řídicích systémů je realizována jako OFF-LINE.

Dá se očekávat rostoucí poptávka po realizaci ON-LINE řídicích systémů, kdy bude nutné integrovat software řídicích systémů (interpolační smyčky)



s nadřazeným CAM jádrem. CAM jádro tak bude mít možnost ovlivňovat proces interpolace a adaptovat jej na danou technologii. Bude se tak dít díky napojení zpětných vazeb ze sensorického subsystému stroje jak na funkce řídicího systému, tak na funkce CAM jádra. Pak již nebude jednoznačné odlišit funkce řídicího systému od funkcí CAM jádra.

Význam ON-LINE řízení vzrůstá s technologickou flexibilitou navržených strojů. Některé současné stroje jsou navrženy tak, že je lze použít pro více technologií, přitom některé z technologií vyžadují realizaci adaptivního řízení, jiné nikoli.

Softwary pro řízení strojů v současnosti daleko zaostávají za fyzickými možnostmi strojů. Cílem vývojáře řídicího softwaru by mělo být co nejvíce se přiblížit reálným možnostem stroje. Přitom je nutné zvážit, zda vývoj softwaru není postaven nad technologiemi řídicích systémů, které omezují vývojáře dosáhnout výše uvedeného cíle.

Současným cílem vývojových skupin řídicích a CAM systémů je snaha využít vyspělých technologií a získat konkurenční výhodu. Mezi tyto technologie, které silně podporují modularitu a nezávislost interfaců jednotlivých modulů jak na použitém programovacím jazyku tak i na operačním systému a dokonce i na místě běhu modulu patří zejména CORBA, COM, DCOM, které byly vyvinuty na začátku 90-tých let. Většího rozšíření dosáhly až v polovině devadesátých let s příchodem WindowsNT a Linuxu. Výhodou komponent založených na těchto technologiích je pak snadná integrace s prakticky libovolným nástrojem, který tuto technologii podporuje (např. tabulkový procesor nebo prohlížeč webových stránek).

Starší zdrojový kód je pak možné zapouzdřit do komponent, nebo jeho nutné jej přepracovat.

Nasazení těchto vyspělých technologií bude patrné zejména v uživatelském interfacu stroje, nicméně cílem bude využít je i směrem k nižším vrstvám CAM jádra a řídicího softwaru. Lze si tak představit ovládání stroje realizované pomocí HTML hypertextového dokumentu, přičemž každý modul stroje je prezentován svou dynamickou stránkou hypertextového dokumentu.

## 4. ZVOLENÉ METODY ZPRACOVÁNÍ

Pro návrh jednotlivých komponent řídicího programu bylo stanoveno pravidlo – co nejvíce se vyhybat použití jakéhokoliv kódu, který by byl závislý na výrobci softwaru.

### 4.1. Výběr operačního systému

Jako operační systém pro vývoj řídicího programu byly zvoleny Windows NT. Operační systém je důležitou částí řídicího programu, jelikož poskytuje řídicímu programu základní funkční vlastnosti které není třeba vyvíjet, ale pouze je vhodně využít. Nejdůležitější vlastnosti operačního systému pro řídicí program jsou především vstup, archivace a výstup dat z/do nejrůznějších periférií a možnost souběžného zpracování několika úloh (multitasking) a popřípadě grafický výstup [4],[18],[41],[42].

Pro ON-LINE řízení je nezbytné, aby operační systém nabízel možnost souběžného zpracování několika úloh. Přitom důležitými vlastnostmi multitaskingu je determinismus, synchronizace, komunikace, plánování a priorita jednotlivých běžících úloh. Pro realizaci vyšších stupňů řízení, jejichž algoritmy dokážou značně zatížit procesor (zpracování obrazu, zpracování dat z hmatových senzorů popřípadě rozpoznávání řeči aj.) je vhodné, aby operační systém podporoval víceprocesorové zpracování úloh.

### 4.2. Výběr programovacího jazyku a knihoven

Pro vývoj aplikace byl použit programovací jazyk firmy Microsoft – Visual C++ 6.0, který je v této verzi již značně propracovaný a pomocí nějž je v případě nutnosti možné psát i kód v Assembleru (pro drivery k hardwaru aj.). Součástí tohoto nástroje je též knihovna STL (Standard Template Library), která je v programu hojně využívána [5], [6], [13].

Pro základní matematické operace jako násobení vektorů, operace s transformačními maticemi 4x4 aj. je použita na rozdíl od práce [2] nová objektová knihovna, která je lépe navržena z hlediska výkonnosti a použití v OpenGL [7].

### 4.3. Výběr CAD systému pro návrh modelu

Pro vytvoření 3D modelu byl použit AutoCAD r.14. Po vytvoření výkresu robotizovaného pracoviště je model uložen ve formě DXF souboru [11]. Jde vlastně o kompletní reprezentaci výkresu v textové formě, která je přístupná ostatním programům. Formát DXF byl stanoven jako světový standard. Řídicí program není nijak s AutoCADem svázan (je pouze vázán na DXF soubor) a proto lze použít při dodržení jistých pravidel jakýkoliv jiný CAD nástroj s možností výstupu do DXF souboru. Pomocí CAD nástroje lze snadno provádět experimenty s umístěním robotu na pracovišti, popřípadě konstrukcí robotu samotného.

#### **4.4. Vstup dat do programu**

Výkres, který je výstupem CAD systému může poskytovat řídicímu programu následující data:

- grafická data používaná pro zobrazení (animaci)
- konfigurační data ( kinematické uspořádání robotu, popř. rozsahy kloubových hodnot atd. )
- pohybová data ( popis trajektorie pohybu )
- pomocná data pro analýzu modelu ( pohybové stopy koncového efektoru atd. )

Kromě geometrických dat je nutné řídicímu programu dodat ještě informace o použitém hardwaru, dále pak požadavky na konfiguraci a data týkající se popisu pohybu koncového efektoru, pokud nejsou obsaženy již ve výkresu aj. Za tímto účelem byl vytvořen projektový soubor a navržena sada konfiguračních a řídicích povelů. Pro překlad projektového souboru byl navržen jednoduchý kompilátor. Projektový soubor se tak stal „hlavním informačním zdrojem“ řídicího programu.

#### **4.5. Programování pohybu koncového efektoru robotu**

Pro programování pohybu koncového efektoru je v řídicím programu použito OFF-LINE textové metody. Jelikož je součástí programu kompilátor DXF výkresu [2], lze poměrně snadno implementovat i CAD metody.

Přímé (ON-LINE) metody programování pohybu koncového efektoru (metoda Teachin a PlayBack) nejsou v řídicím programu implementovány, jelikož jsou silně závislé na existenci a použitém hardwaru. Jejich implementace do stávajícího kódu je však poměrně snadná.

#### **4.6. Řízení pohybu mezi uzlovými body**

Při pohybu koncového efektoru mezi dvěma uzlovými body trajektorie se buďto nebere ohled na tvar trajektorie mezi uzlovými body (metoda PTP), nebo se požaduje, aby se koncový efektor pohyboval po předem stanovené trajektorii a byl přitom správně orientován (metoda CP) [3],[12]. Metoda PTP i metody CP jsou v řídicím programu použity.

Jak pro metodu PTP, tak i pro spojitě řízení (metodu CP) bylo nutné odvodit matematické vztahy, popisující průběh natočení jednotlivých kloubů robotu.

U metody PTP jsou to vztahy, popisující pohyb jednotlivých kloubů robotu vzájemně nezávisle jeden na druhém, při požadavku splnění jistého kritéria (např. snížit nárazové zatížení zdrojů energie, snížit namáhání konstrukce robotu, snížit spotřebu energie robotu nebo provést pohyb v co nekratším čase atd.).

U metody CP jsou to pak matematické vztahy řešící vazbu mezi polohou a orientací koncového efektoru a jednotlivými natočeními kloubů robotu (kinematická syntéza). Průběh polohy a orientace koncového efektoru mezi dvěmi uzlovými body pak popisují interpolační funkce, jejichž úkolem je zajistit dostatečnou aproximaci požadovaného tvaru trajektorie.

## 5. HLAVNÍ VÝSLEDKY PRÁCE

Není cílem této publikace zabývat se detaily v implementaci, proto zde budou uvedeny hlavní tematické oblasti disertační práce a uvedeny hlavní výsledky práce.

Nedílnou součástí disertační práce jsou zdrojové texty programu pro řízení šestiosého antropomorfního robotu v elektronické podobě na CD disku a proto není v práci uveden žádný výpis ani komentář zdrojových textů, jelikož by tím nabyla značného rozsahu.

Pro účely disertační práce je řídicím programem nebo aplikací rozuměn především vyvíjený program.

### 5.1. Odvození matematických vztahů pro realizaci kinematické analýzy, syntézy a interpolačních funkcí

Kapitoly zabývající se odvozením matematických vztahů tvoří jednu třetinu disertační práce. Není možné v rozsahu této publikace čtenáře s matematickými vztahy blíže seznámit, proto jsou zde uvedeny pouze některé jednodušší vztahy které nevyžadují komentář. Na ostatní v disertační práci řešené problémové oblasti je pouze upozorněno+.

Cílem kinematické analýzy je ze známých hodnot natočení jednotlivých kloubů robotu (kloubových hodnot) vypočíst polohu a orientaci koncového efektoru.

Řešení pohybu robotu zahrnuje odvození speciálních vztahů mezi jednotlivými články robotu. Základem všech problémů při řešení kinematické analýzy, kinematické syntézy, interpolace aj. je kinematika současných prostorových pohybů těles, kterou lze s výhodou popsat pomocí vektorové a maticové algebry.

V disertační práci je pro stanovení transformačních matic mezi souřadnými systémy dvou sousedních článků používána metoda, kterou zavedli Denavit a Hartenberg.

Pro odvození matematických vztahů byly na modelu robotu stanoveny dle této metody souřadné systémy. Transformace mezi souřadnými systémy je touto metodou obecně popsána transformační maticí:

$$\mathbf{T}_{i+1,i} = \begin{bmatrix} c(\vartheta_i) & -s(\vartheta_i)c(\alpha_i) & s(\vartheta_i)s(\alpha_i) & a_i c(\vartheta_i) \\ s(\vartheta_i) & c(\vartheta_i)c(\alpha_i) & -c(\vartheta_i)s(\alpha_i) & a_i s(\vartheta_i) \\ 0 & s(\alpha_i) & c(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Rov. 1}$$

Polohu obecného bodu M který je pevně svázán s posledním článkem robotu lze v souřadné soustavě 1 formulovat jako:

$$\bar{\mathbf{r}}_1^M = \mathbf{T}_{21} \mathbf{T}_{32} \mathbf{T}_{43} \mathbf{T}_{54} \mathbf{T}_{65} \mathbf{T}_{76} \bar{\mathbf{r}}_7^M \quad \text{Rov. 2}$$

Obdobně jsou v práci formulovány vztahy pro ostatní články robotu.

Výslednou homogenní transformaci ze souřadného systému 7 do systému 1 lze formulovat jako maticový součin jednotlivých relativních transformací:

$$\mathbf{T}_{71} = \begin{bmatrix} \mathbf{C}_{71} & \mathbf{r}_1^{O_7} \\ \mathbf{0}^T & 1 \end{bmatrix} = \mathbf{T}_{21} \mathbf{T}_{32} \mathbf{T}_{43} \mathbf{T}_{54} \mathbf{T}_{65} \mathbf{T}_{76} = \mathbf{T}_{41} \mathbf{T}_{74} \quad \text{Rov. 3}$$

Jako globální souřadný systém byl zvolen souřadný systém 0. Transformace  $\mathbf{T}_{21}, \mathbf{T}_{32}, \mathbf{T}_{43}, \mathbf{T}_{54}, \mathbf{T}_{65}, \mathbf{T}_{76}$  jsou funkcí kloubové souřadnice. Transformace  $\mathbf{T}_{10}$  byla zvolena pro definování umístění robotu v rámci jeho pracoviště.

Výstupem interpolačních funkcí je obecně transformační matice  $\mathbf{T}_{I0}$  popisující polohu a orientaci v interpolačním bodě. Tuto transformaci by již bylo možné považovat jako vstupní matici pro výpočet natočení kloubů robotu (kinematická syntéza) avšak za podmínky, že na posledním článku robotu není umístěn žádný koncový efektor.

Pro praktické použití je však třeba brát v úvahu koncový efektor, který je pevně navázán na poslední článek robotu. Na tomto koncovém efektoru je pak výhodně stanoven referenční bod popisu trajektorie.

Z výše uvedeného vyplývá potřeba transformace referenčního bodu do souřadné soustavy posledního článku kinematického řetězce robotu (souřadné soustavy 7).

$$\mathbf{T}_{I0} = \mathbf{T}_{10} \cdot \mathbf{T}_{71} \cdot \mathbf{T}_{I7} \quad \text{nebo} \quad \mathbf{T}_{70} = \mathbf{T}_{I0} \cdot (\mathbf{T}_{I7})^{-1} \quad \text{Rov. 4}$$

Kde:  $\mathbf{T}_{I0}$  reprezentuje spočtený interpolační bod interpolační funkcí

$\mathbf{T}_{10}$  zohledňuje umístění robotu v rámci pracovního prostoru

$\mathbf{T}_{71}$  je transformací ze soustavy 7 do 1

$\mathbf{T}_{I7}$  je uživatelem zadaný referenční bod (včetně orientace) v souřadnicích souřadné soustavy č. 7.

Pro zobrazování 3D modelu v rovině stínítka monitoru byla použita transformace, zajišťující přepočtení 3D souřadnic do průmětové roviny. Pak výslednou transformaci lze psát jako:

$$\mathbf{T}_{\text{výsl}} = \mathbf{T}_{3D/2D} \cdot \mathbf{T}_{I0} \quad \text{Rov. 5}$$

Libovolný bod M koncového efektoru lze pak v průmětové rovině zobrazit na pozici:

$$\bar{\mathbf{r}}_{2D}^M = \mathbf{T}_{3D/2D} \mathbf{T}_{10} \mathbf{T}_{21} \mathbf{T}_{32} \mathbf{T}_{43} \mathbf{T}_{54} \mathbf{T}_{65} \mathbf{T}_{76} \bar{\mathbf{r}}_7^M \quad \text{Rov. 6}$$

Načtením souřadných systémů z výkresu CAD systému jsou získány bez jakýchkoliv matematických úprav transformace  $\mathbf{T}_{10}, \mathbf{T}_{20}, \dots, \mathbf{T}_{70}$ . Aby však bylo možné model „oživit“ (animovat) a sestavit pro něj matematický model, bylo nutné získat relativní transformace  $\mathbf{T}_{21}, \mathbf{T}_{32}, \dots, \mathbf{T}_{76}$ . Proto byly odvozeny vztahy zajišťující získání jednotlivých relativních transformací pro jednotlivé články robotu.

Kinematická analýza je v řídicím programu používána mimo jiné pro zobrazení jednotlivých článků (pohybovou animaci) na monitoru počítače a zobrazení trajektorie (stopy) sledovaných bodů robotu.

Kinematická syntéza (zpětná kinematická úloha) řeší problém, jak získat ze známé (požadované) pozice a orientace koncového efektoru kloubové hodnoty (známe  $\mathbf{T}_{71}$  a úkolem je spočítat  $\vartheta_1, \vartheta_2, \vartheta_3, \vartheta_4, \vartheta_5, \vartheta_6$ ). Pro řízení robotu je kinematická syntéza velmi důležitá. Schopnost získat rychle výsledky ze zpětné transformace je rozhodující při vedení robotu po předem stanovené (naprogramované) dráze v kartézských souřadnicích. Kinematická syntéza bývá značně složitá a náročná a je řešitelná v uzavřeném tvaru jen v ojedinělých případech [12].

Vstupem pro kinematickou syntézu je poloha a orientace posledního článku robotu vyjádřená transformační maticí.

$$\mathbf{T}_{71} = \begin{bmatrix} \mathbf{C}_{71} & \mathbf{P} \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{N} & \mathbf{S} & \mathbf{A} & \mathbf{P} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} N_x & S_x & A_x & P_x \\ N_y & S_y & A_y & P_y \\ N_z & S_z & A_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Rov. 7}$$

V disertační práci byly odvozeny analytické vztahy, které na základě spočtených vstupních interpolačních poloh stanoví natočení jednotlivých kloubů robotu. Tyto vztahy pak byly v řídicím programu zapouzdřeny do funkce jejíž vstupem je výše uvedená transformace a výstupem je vektor spočtených kloubových souřadnic.

Interpolace je metoda (funkce) zajišťující přechod z jednoho programovaného bodu do druhého podél definované dráhy, při naprogramovaném stupni posuvu tak, aby byla zajištěna dostatečná aproximace požadovaného tvaru trajektorie [3],[39].

Obecně je vstupem pro interpolační funkci:

- aktuální poloha koncového efektoru  $\mathbf{p}^0 = [p_x^0, p_y^0, p_z^0]^T$  a cílová poloha  $\mathbf{p}^N = [p_x^N, p_y^N, p_z^N]^T$
- aktuální orientace koncového efektoru  $\mathbf{C}^0$  a cílová orientace  $\mathbf{C}^N$ , přičemž změna orientace podél předepsaného typu trajektorie se děje kolem obecné

prostorové osy která je dána změnou natočení  $C^0$  na  $C^N$  (nebo je změna orientace předepsána jiným vhodným způsobem)

- předepsaný typ trajektorie koncového efektoru mezi aktuální a cílovou polohou – lineární (kruhová, parabolická, spline aj.)
- absolutní hodnota tečné rychlosti v bodě  $\mathbf{p}^0$  a v bodě  $\mathbf{p}^N$  ( $v^0, v^N$ )
- interval časového kvantování  $\Delta t$  (obvykle 1 - 20 ms), což je globální konstanta daná použitým hardwarem. Tato hodnota je vlastně doba 1 taktu (regulační smyčky) regulátoru servopohonu.

Předpoklad: zrychlení mezi uzlovými body je konstantní ( $a = konst.$ )

V disertační práci byly odvozeny vztahy pro lineární interpolaci, kde jsou programované body (uzlové body) spojeny přímkami, bez ohledu na vzdálenost mezi body. Pro lineární interpolaci byla uvažována rovnoměrná změna orientace z aktuální na cílovou podél předepsané trajektorie.

Dále byly odvozeny vztahy pro interpolaci kruhovou, u které byly uvažovány dvě varianty:

1. Kruhová interpolace bez změny orientace v jednotlivých interpolačních bodech vůči globálnímu souřadnému systému (dochází pouze ke změně polohy).
2. Kruhová interpolace bez změny orientace vůči ose kruhového oblouku (dochází ke změně polohy a orientace vůči globálnímu souřadnému systému).

Kruhová interpolace se používá všude tam, kde je třeba, aby koncový efektor robotu vykonával úplné nebo částečné kružnice, popřípadě aproximace kružnic. K programování oblouku je zapotřebí programovat polohu středu, radius vektor počátečního bodu, úhel, směr otočení, popřípadě ještě orientaci a rychlost podél interpolované trajektorie.

Pro obecné použití byly v práci odvozeny vztahy s jejichž pomocí byla realizována interpolace pomocí Coonsových kubik [8],[9]. Použití Coonsových kubik pro popis trajektorie koncového efektoru robotu je přínosné zejména v tom, že umožňují popsat obecnou prostorovou trajektorii a přitom zabezpečují plynulou změnu rychlosti. Výhodou polynomiálních křivek třetího stupně (kubik) zejména je, že poskytují širokou škálu tvarů, přičemž jsou nenáročné na výpočet, lze s nimi snadno manipulovat a je možné u nich zaručit spojitost křivky  $C^2$ , což znamená spojitost druhých derivací ve všech bodech křivky [8].

Aby bylo nasazení robotu v průmyslu co nejvíce efektivní, měl by vykonávat pohyby v co nekratším čase. Při programování robotu lze často najít pohyby, u kterých nezáleží na tvaru trajektorie ani na výsledné rychlosti a zrychlení pohybu koncového efektoru, jediným požadavkem bývá zaujmutí požadované polohy a



orientace v cílovém bodě pohybu a to nejlépe v co nejkratším čase. Pro tyto účely byly odvozeny vztahy pro interpolaci PTP (Point To Point).

Na rozdíl od předešlých interpolačních metod není popisován pohyb koncového efektoru, nýbrž je brán zřetel na jednotlivé klouby robotu, které vykonávají pohyb zcela nezávisle.

Jelikož je však řízena reálná soustava, je nutno kontrolovat, zda průběh kloubové hodnoty (natočení) je na reálné soustavě uskutečnitelný, tj. zda nejsou překročeny mezní hodnoty rozsahu kloubové souřadnice (pohyb mimo pracovní prostor robotu), dále pak hodnoty první a druhé derivace (úhlová rychlost a zrychlení). Pro účely kontroly kloubových souřadnic byly odvozeny některé základní vztahy.

Veškeré odvozené matematické vztahy jsou součástí programu pro řízení šestiosého antropomorfního robotu. Na základě odvozených vztahů lze realizovat popřípadě odvodit různé nové, dokonalejší interpolační funkce, nebo stávající modifikovat.

## **5.2. Propojení návrhu robotizovaného pracoviště v CAD systému s řídicím programem**

Pro realizaci propojení byla stanovena taková pravidla, kdy konstruktér není během fáze návrhu modelu v CAD systému nijak omezen, tj. může používat všechny entity a operace nad entitami, které CAD systém nabízí. Pouze pokud chce model použít jako vstup do řídicího programu, musí dodržet jistá pravidla, která vyžaduje řídicí program pro správnou interpretaci dat z výkresu.

Byla stanovena pravidla pro:

- definici geometrie
- definici kinematického uspořádání

Hlavními zásadami pro definici geometrie jsou:

- Definování geometrie (vzhledu) robotu a okolního prostředí není pro řídicí program nezbytné. Lze definovat velmi propracovaný model robotu a okolního prostředí, avšak je nutno si uvědomit, že to s sebou přináší značné nároky na rychlost grafického výstupu.
- Kinematický mechanismus (robot) musí být definován (nakreslen) ve výchozí poloze, která je stanovena při odvozování vztahů pro kinematickou analýzu a syntézu.
- Drátová forma reprezentace modelu a tím omezená sada entit pro definování geometrického vzhledu (Line, 3D Polyline, 3D Face a dále pak všechny entity,

jejichž základem je mesh (sít') a to otevřená nebo jednostranně či oboustranně uzavřená (Box, Sphere, Cylinder, Cone, Torus aj.).

- Každý pohyblivý článek robotu je uložen v samostatné hladině výkresu. Aby řídicí program z výkresu rozpoznal, že daná hladina je určena právě jemu, získal navíc informaci o který článek robotu se jedná je hladina pojmenována dle jistých pravidel.

Hlavními zásadami pro definici kinematického uspořádání jsou:

- Pro definici používat předdefinovaný uživatelský souřadný systém.
- Kinematické uspořádání definovat nejlépe v samostatných hladinách výkresu použitím předdefinovaného uživatelského souřadného systému.
- Souřadné systémy definovat podle pravidel použitých při odvození vztahů pro kinematickou analýzu a syntézu.
- Dodržet syntaxi názvu hladin výkresu.

Jak definice geometrie, tak i definice kinematického uspořádání je značně univerzální a snadno rozšiřitelná na jiné kinematické struktury, popřípadě na modely s několika kinematickým mechanismy na pracovišti.

### 5.3. Vstupní projektový soubor

Jako vstup do programu byl zvolen ASCII textový soubor (projektový soubor), který vedle povelů samotných (konfiguračních a řídicích) může obsahovat i reference na jiné zdrojové soubory.

Obsahem projektového souboru jsou konfigurační instrukce, řídicí instrukce a instrukce určené pro debugging (ladění). Syntaxe a názvy jednotlivých instrukcí byly zvoleny s ohledem na jejich jednoduchou interpretaci kompilátorem projektového souboru. Pro organizaci a syntaxi instrukcí v projektovém souboru byla stanovena pravidla, která jsou v disertační práci popsána.

INSTRUKCE:

- RobotName Určena pro definici názvu stroje.
- ReadDXF Určena pro načtení grafických dat z výkresu (dxf souboru).
- RegulationCycle Definiuje dobu regulační smyčky servopohonu v milisekundách.
- CommunicationDataQuantum Definiuje počet interpolačních poloh, které jsou předány hardwaru na jeho žádost.

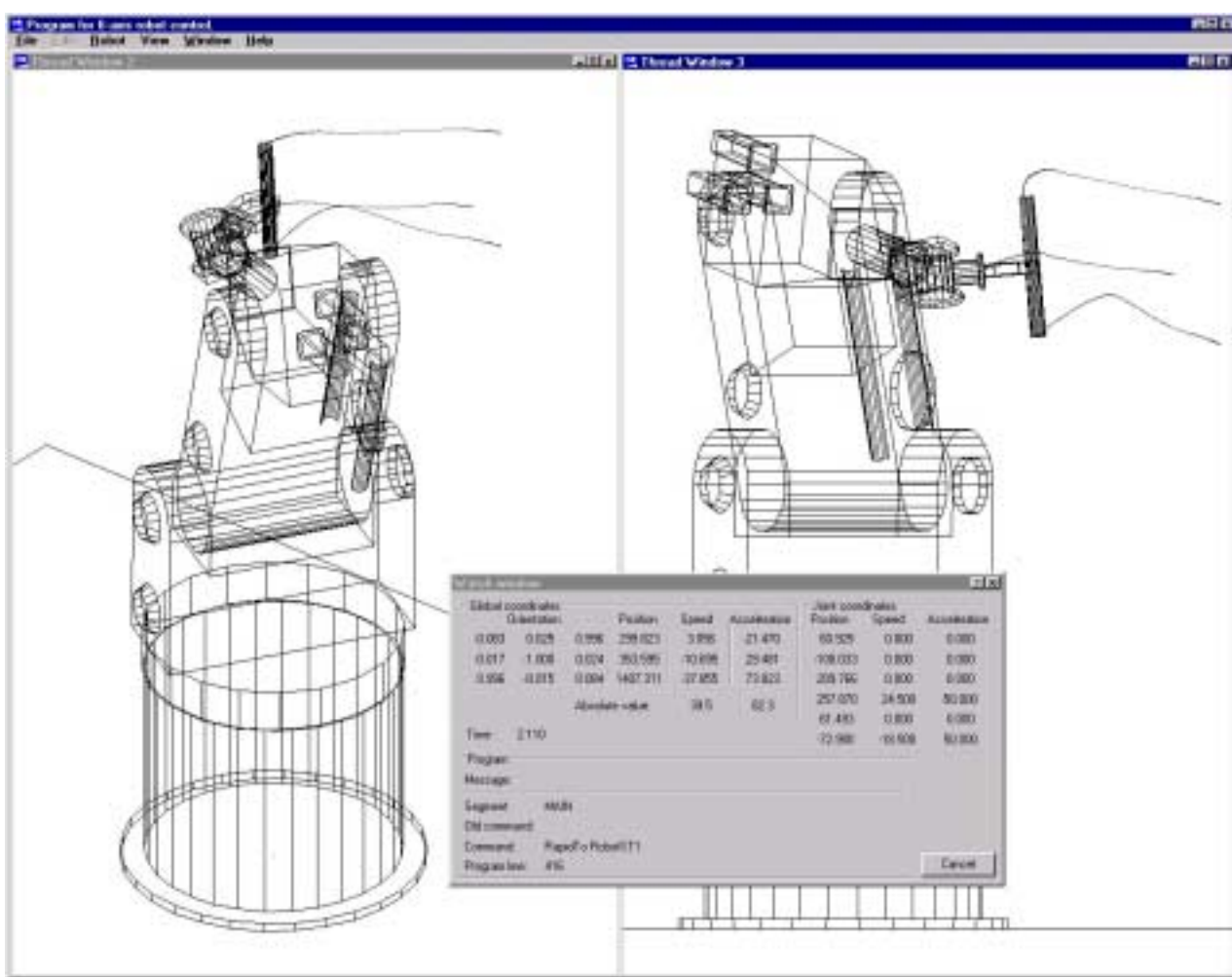
- Simulation Zajistí spuštění řídicího programu v simulačním modu bez ohledu na to, zda je připojen hardware.
- FastSimulation Tatáž funkce jako instrukce Simulation avšak s ignorováním hodnoty nastavené instrukcí RegulationCycle.
- V (Vector) Definuje pojmenovaný 3D vektor.
- R (Rotation) Definuje pojmenovanou orientační matici C
- NP (Nodal Point) Definuje uzlové body trajektorie (Nodal Points).
- Validations Zapne/vypne kontrolu polohy, rychlosti, zrychlení v kloubových souřadnicích stroje.
- MinPosition, MaxPosition Nastavení spodní resp. horní hranice rozsahu polohy všech kloubů.
- MaxSpeed Nastavení maximální hodnoty úhlové rychlosti ve všech kloubech.
- MaxAcceleration Nastavení maximální hodnoty úhlového zrychlení ve všech kloubech.
- CPOINT Nastavení referenčního (řídicího) bodu popisu trajektorie.
- TP (TracePoint) Nastavení sledovacích bodů.
- SEGMENT Definuje začátek segmentu (bloku) trajektorie.
- END Definuje konec segmentu.
- SEG Řídicí povel pro spuštění patřičného segmentu.
- LineTo Příkaz k vykonání pohybu z aktuální polohy do cílové po přímce.
- ArcTo Příkaz k vykonání pohybu z aktuální polohy do cílové po kruhovém oblouku.
- RapidTo Příkaz pro vykonání pohybu do cílové polohy v co nejkratším čase.
- CoonsTo Příkaz pro vykonání obecného prostorového pohybu, který je popsán Coonsovými kubikami.
- Wait Definuje časové prodlevy.
- DumpJoints Umožňuje provést výpis kloubových hodnot do textového souboru.

V disertační práci jsou parametry jednotlivých instrukcí kompletně zdokumentovány. Výše uvedenou sadu instrukcí lze poměrně snadno rozšiřovat o nové.

## 5.4. Realizace interpolačních funkcí

Další asi jedna třetina disertační práce je věnována popisu řídicího programu s uvedením příkladů, které předvádí možnosti programování koncového efektoru robotu. V rámci možností této publikace jsou zde uvedeny pouze obrázky, které vystihují nejdůležitější funkční vlastnosti řídicího programu. V disertační práci jsou navíc k jednotlivým příkladům uvedeny výpisy ze vstupního projektového souboru.

Program umožňuje ON-LINE sledování hlavních kinematických veličin a je koncipován jako více-vláknová (multi-thread) aplikace, kdy není nutné pro běh programu aby byla animace modelu (jakékoliv pohledové okno na model ) včetně okna pro sledování kinematických parametrů (Watch Window) aktivní.

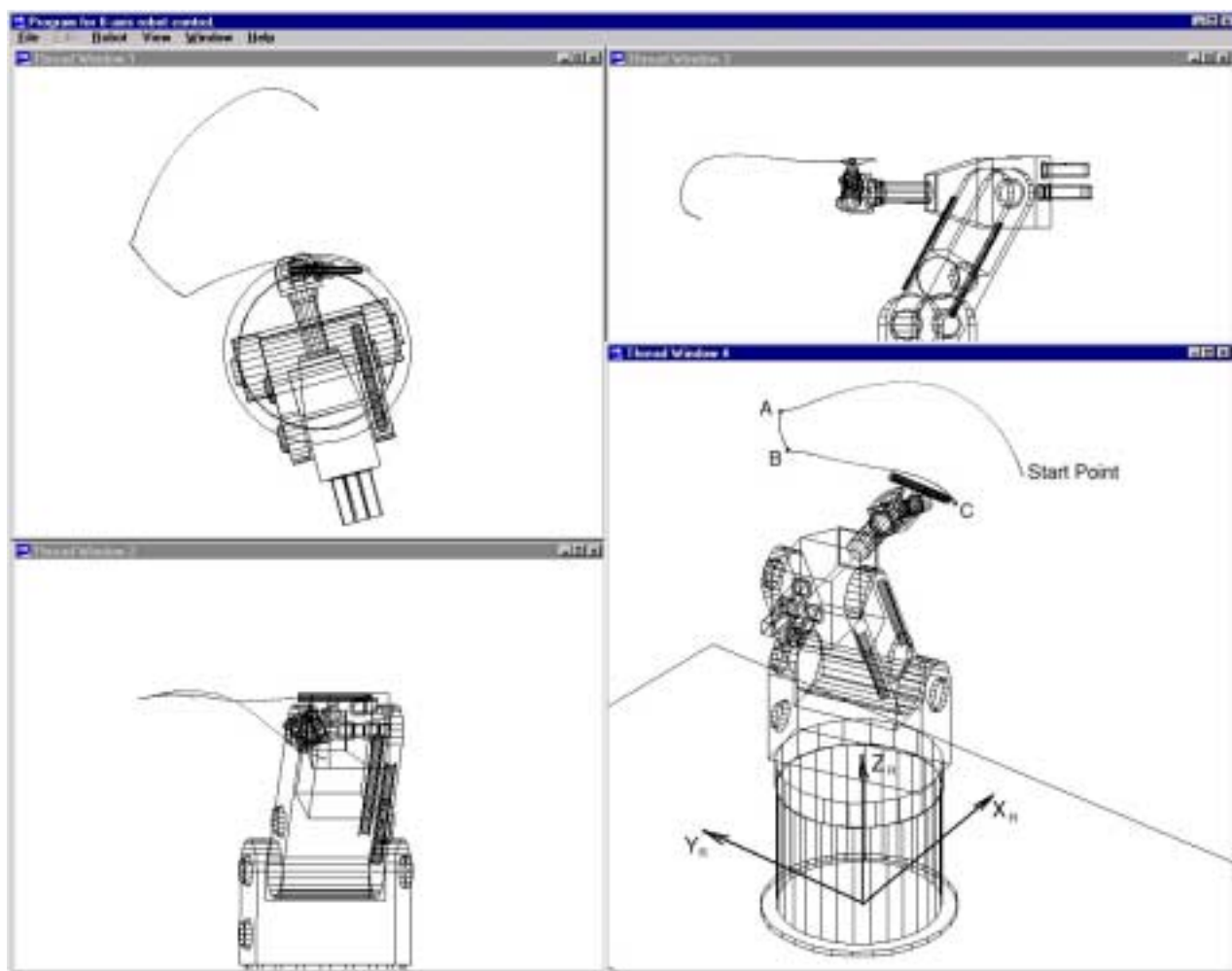


Obr. 1: Sledování hlavních kinematických veličin

Modifikací výkresu (pořízeného CAD systémem) je možné provádět úpravy vzhladu. Instrukcí TP (TracePoint) lze nakonfigurovat sledovací body umístěné na koncovém efektoru, které za sebou zanechávají během pohybu stopu.

Instrukcí CPOINT lze nastavit referenční bod koncového efektoru, ke kterému je vztažen program pohybu.

Interpolace PTP charakteristická tím, že nezáleží na tvaru trajektorie a průběhu orientace během pohybu, pouze je požadováno zaujmutí určité polohy a orientace v prostoru při plnění jistého kritéria (maximální rychlost pohybu, minimální spotřeba energie aj.). Pro interpolaci PTP bylo použito kritéria maximální rychlosti. Díky dostupnosti zdrojových textů je lze snadno rozšířit o další PTP interpolace zohledňující jiná kritéria pohybu.



Obr. 2: Pohyb PTP

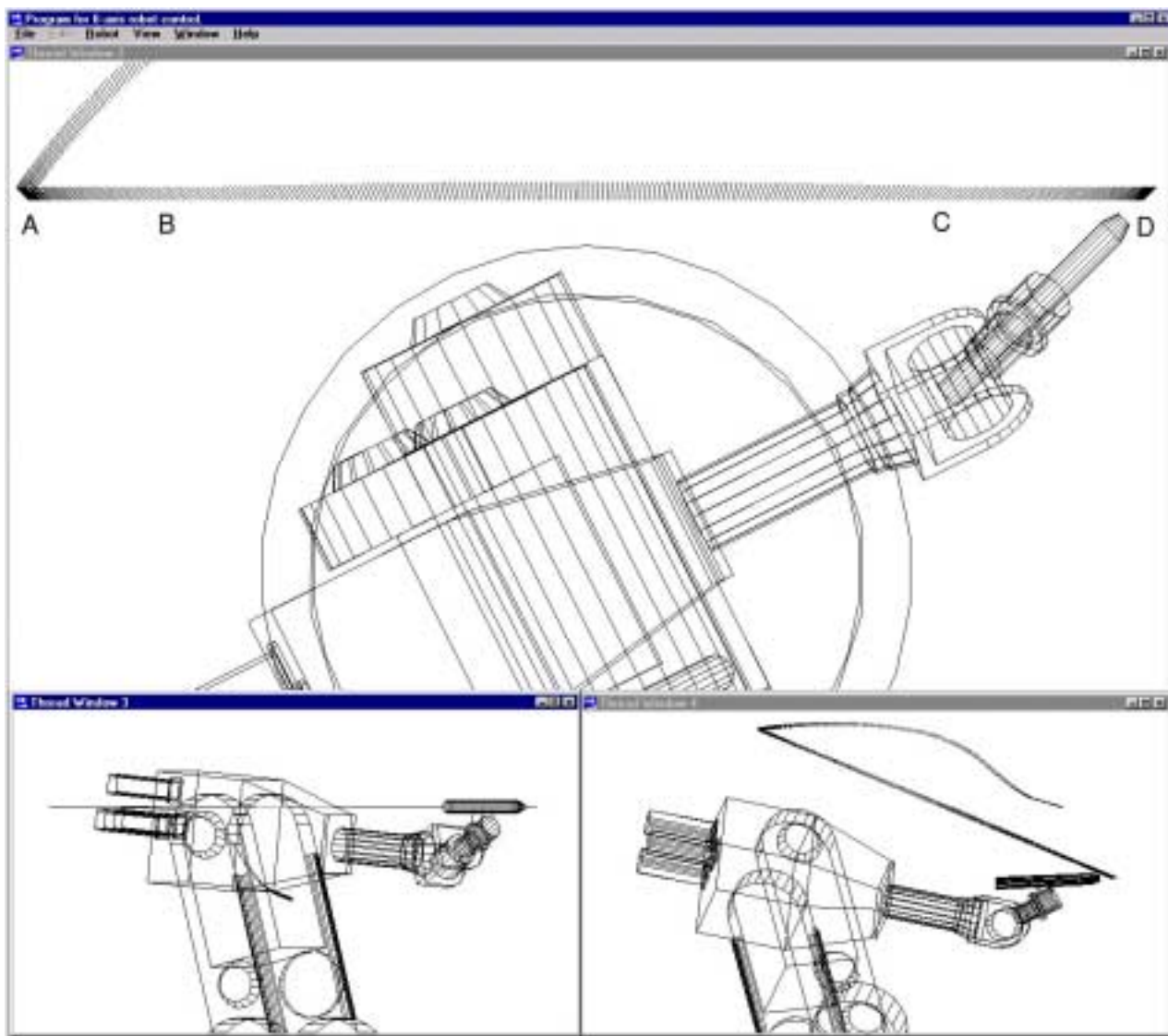
Pro pohyb PTP byl uvažován typický „rampovitý“ průběh rychlosti v každém kloubu robotu. Pohyb je tak charakteristický následujícími etapami:

1. rozběh s maximální možnou hodnotou zrychlení na maximální otáčky
2. pohyb s konstantní maximální rychlostí
3. brždění s maximální možnou hodnotou brzdného zrychlení

Popis trajektorie pro všechny interpolační funkce v programu použité je vztažen k souřadnému systému  $X_R, Y_R, Z_R$ .

Pohybové instrukce projektového souboru lze libovolně řadit za sebou a seskupovat do bloků a ty následně volat. Počet pohybových instrukcí není omezen.

Na dalším obrázku je patrný průběh natočení svařovací hubice během lineárního pohybu a zhuštění stopy na začátku a na konci pohybu, což je dáno postupným rozběhem a brzděním na konci pohybu.



*Obr. 3: Pohyb po přímce*

Nechť zadáním je rovnoměrný pohyb konstantní rychlostí 500 mm/sec po přímce z bodu B do bodu C. Přitom je požadavkem změnit orientaci o  $-90$  stupňů kolem osy  $Z_R$ . Výchozí orientace nechť je dána rotační maticí. Po definici uzlových bodů lze pohyb vykonat následující sadou instrukcí v projektovém souboru:

*SEGMENT MAIN*

*RapidTo A;      PTP pohyb do výchozí polohy*

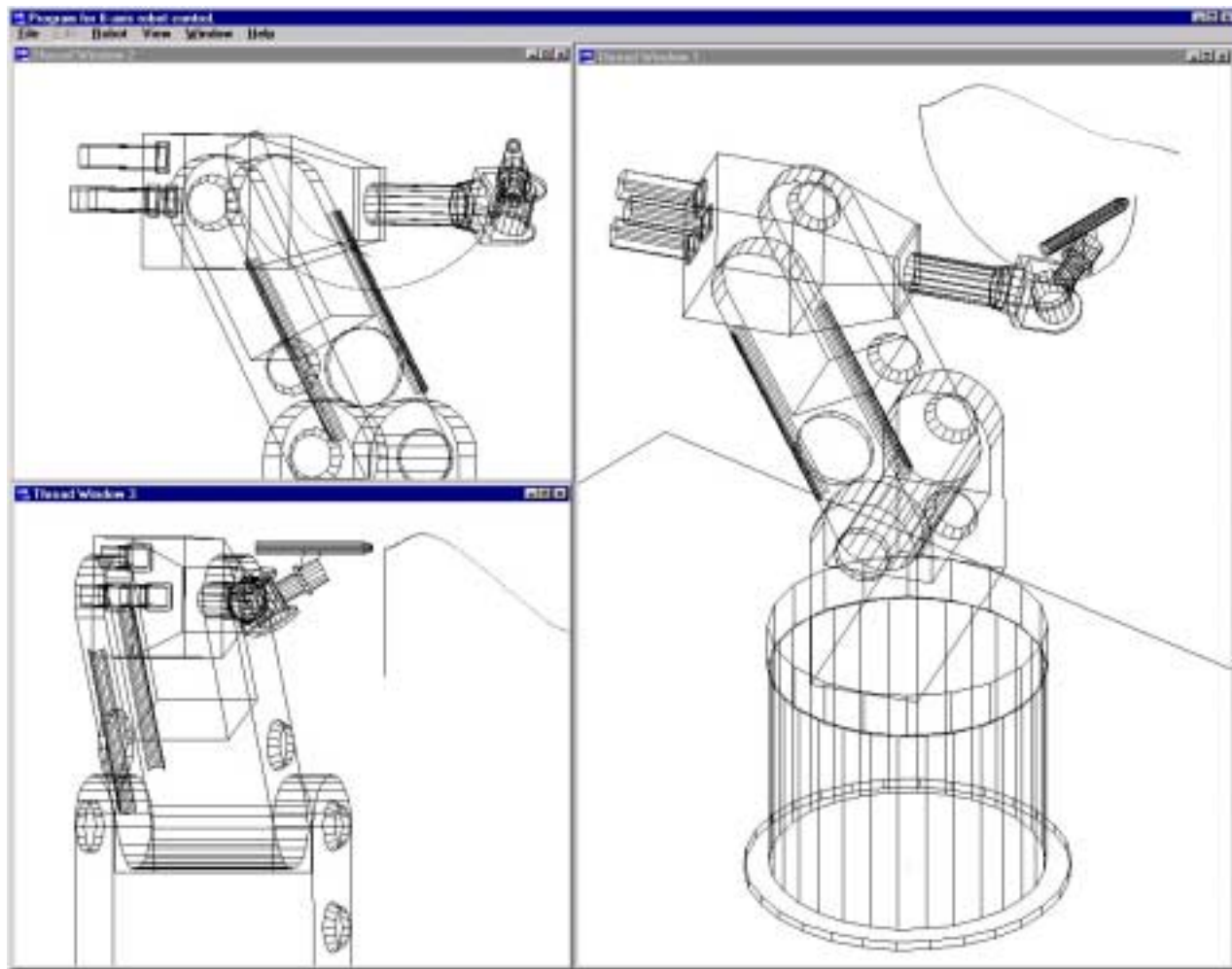
*LineTo B,500;    Rozjezd - lineární pohyb do bodu B tak, aby bylo v tomto bodě  
;dosaženo rychlosti 500 mm/sec*

*LineTo C,500;    Pohyb z bodu B do bodu C dle zadání*

*LineTo D,0;      Brzdění do úplného zastavení*

*END*

Pro kruhovou interpolaci je nutné nastavit polohu středu kruhového oblouku, směrový vektor osy, úhel otočení kolem osy a velikost tečné rychlosti která má být dosažena při průchodu cílovým bodem. Dále pak je možné zvolit metodu změny orientace buďto jako fixní ke globálnímu souřadnému systému nebo k ose kruhového oblouku. Lze navrhnout a realizovat ještě spoustu dalších variant kruhové interpolace s různými požadavky na průběh orientace během pohybu. V disertační práci byly realizovány pouze dvě výše jmenované.



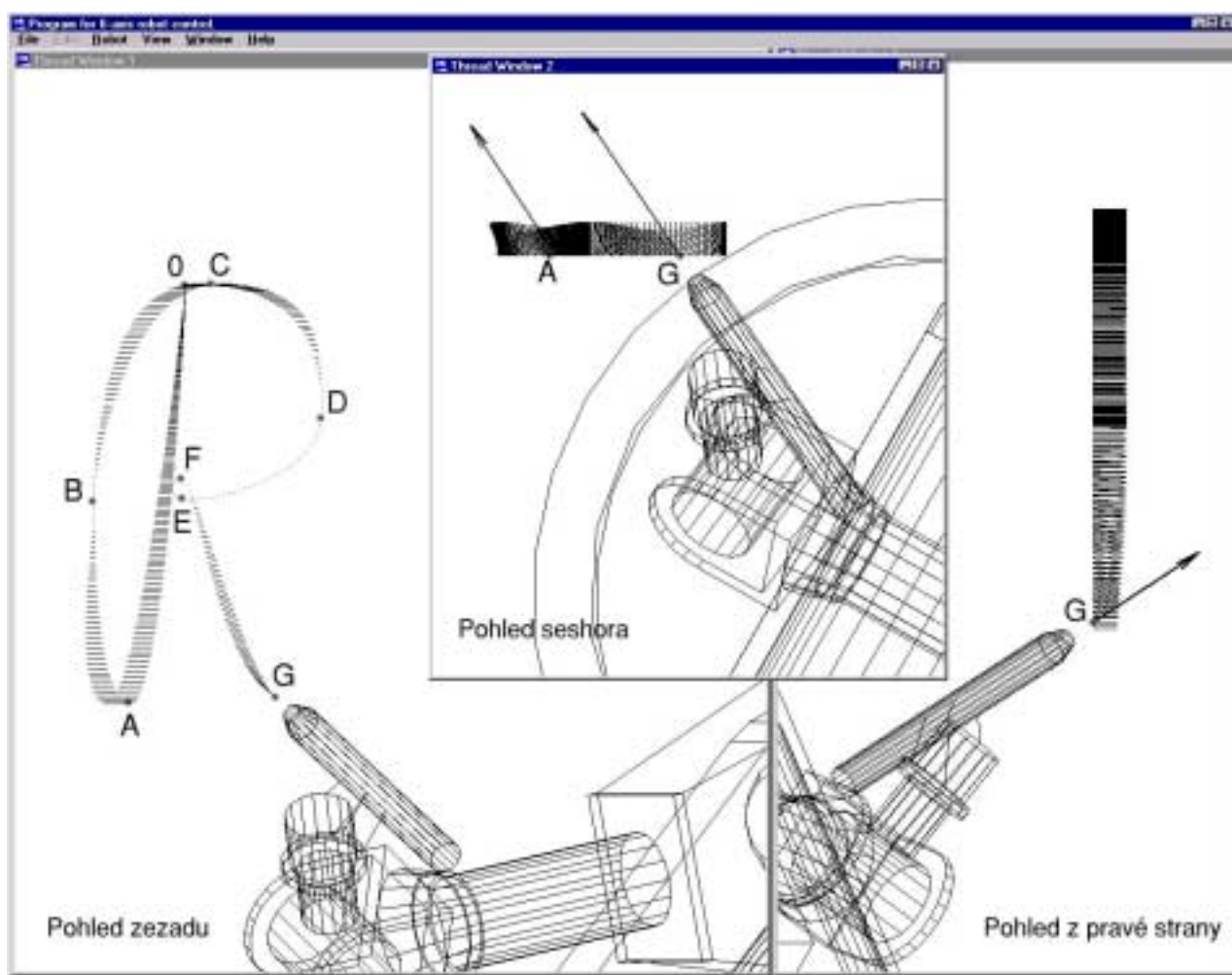
Obr. 4: Pohyb po kruhovém oblouku

Po definování středu Center, osy Axis kruhového oblouku, zvolení metody změny orientace lze splnit požadavek otočení kolem osy kruhového oblouku o -180 stupňů, přičemž 30 stupňů na začátku a na konci pohybu lze použít na urychlení, resp. brždění pohybu a požadavek konstantní tečné rychlosti 300 mm/sec během činné fáze pohybu následující sadou instrukcí:

<i>SEGMENT Arc1;</i>	<i>Seskupení instrukcí do segmentu</i>
<i>ArcTo (1,Center,Axis,-30,300);</i>	<i>Fáze rozjezdu</i>
<i>ArcTo (1,Center,Axis,-120,300);</i>	<i>Pohyb dle zadání</i>
<i>ArcTo (1,Center,Axis,-30,0);</i>	<i>Fáze brždění</i>
<i>END</i>	

Interpolace pomocí Coonsových kubik je nejobecnější a i nejsložitější formou popisu pohybu, která je v řídicím programu dosud implementována. Složitostí jsou zde myšleny značné nároky na představivost programátora. Za pomoci tohoto popisu lze realizovat prostorové pohyby, u nichž lze navíc předepsat průběh natočení a dokonce i průběh velikosti tečné rychlosti během pohybu.

Trajektorie pohybu referenčního bodu koncového efektoru je u této metody interpolace popsána množinou uzlových bodů. Každý uzlový bod sestává z 12-ti složek. 3 složky jsou určeny pro polohu uzlového bodu, 3 složky pro tečnu 3D trajektorie v daném uzlovém bodě, 2 hodnoty definují jaké orientace bude v tomto uzlovém bodě dosaženo vůči referenčnímu souřadnému systému a další 2 složky definují strmost změny orientace v uzlovém bodě. Poslední dvě složky jsou určeny pro stanovení velikosti tečné rychlosti a její směrnice v daném uzlovém bodě.



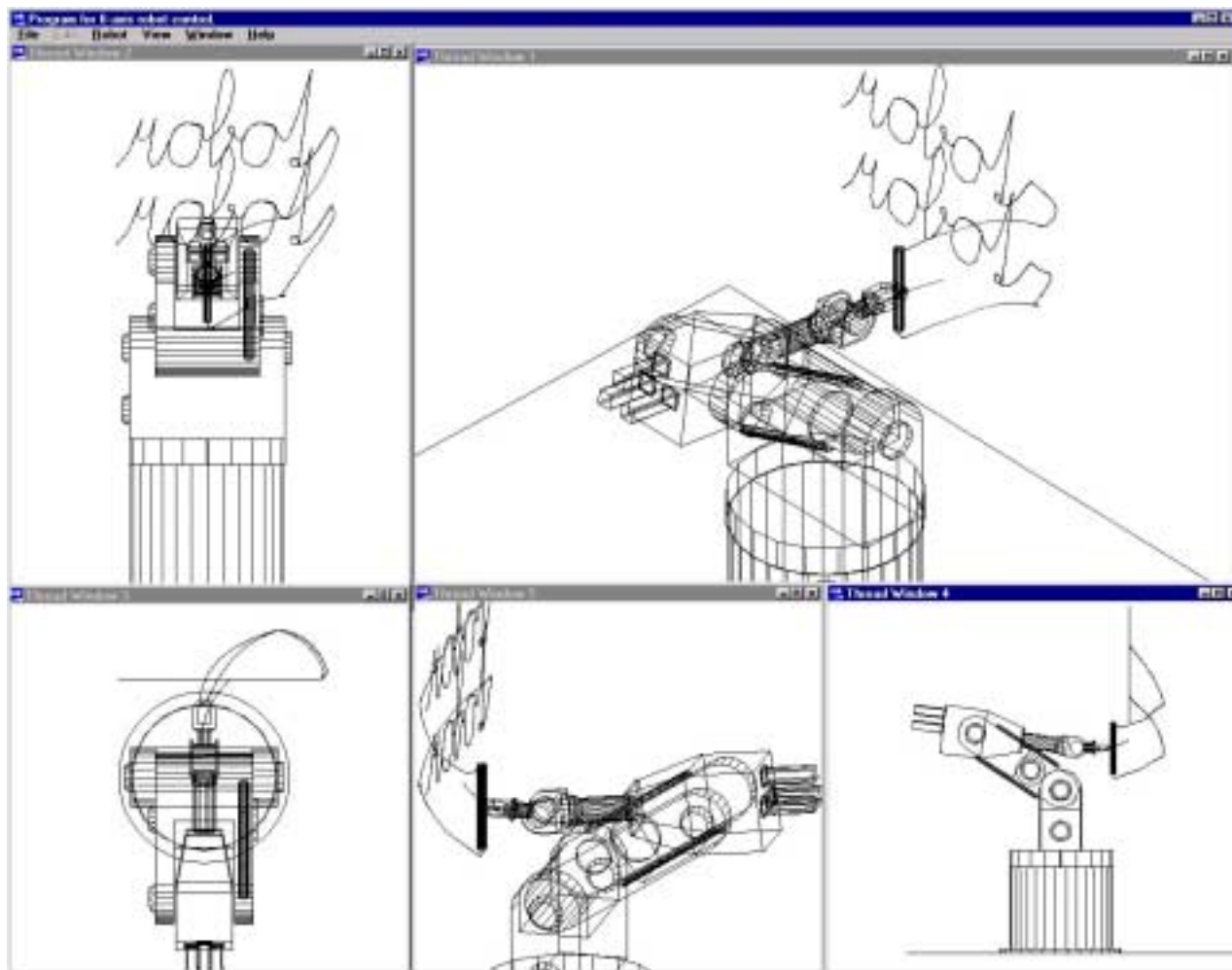
Obr. 5: Obecný prostorový pohyb včetně změny orientace

Orientace vůči globálnímu souřadnému systému je stanovena dvěma úhly - hodnotou rotace a elevace.

Na výše uvedeném obrázku je parná změna orientace (rotace) mezi body 0,A,B,C,D. Mezi body D,E,F je orientace vůči globálnímu souřadnému systému konstantní. Mezi body F a G dochází jak ke změně úhlu rotace tak i úhlu elevace.



Jak je i z následujícího obrázku patrné, lze interpolační metodou založenou na Coonsových kubikách realizovat velmi složité prostorové pohyby.



*Obr. 6: Obecný prostorový pohyb*

Výše uvedený pohyb byl realizován definováním 29 uzlových bodů. S rostoucími požadavky na přesnost popisu pohybu je obecně nutné volit větší počet uzlových bodů trajektorie. Počet uzlových bodů pohybu je pouze omezen systémovými prostředky počítače. Pohyb byl pro názornost realizován v rovině, bez změny orientace během pohybu. Pro zobrazení průběhu trajektorie byly aktivovány 3 sledovací body (dva na koncích neseného předmětu, jeden v místě uchycení koncového efektoru). Kontrola polohy, rychlosti a zrychlení v jednotlivých kloubových souřadnicích byla deaktivována.

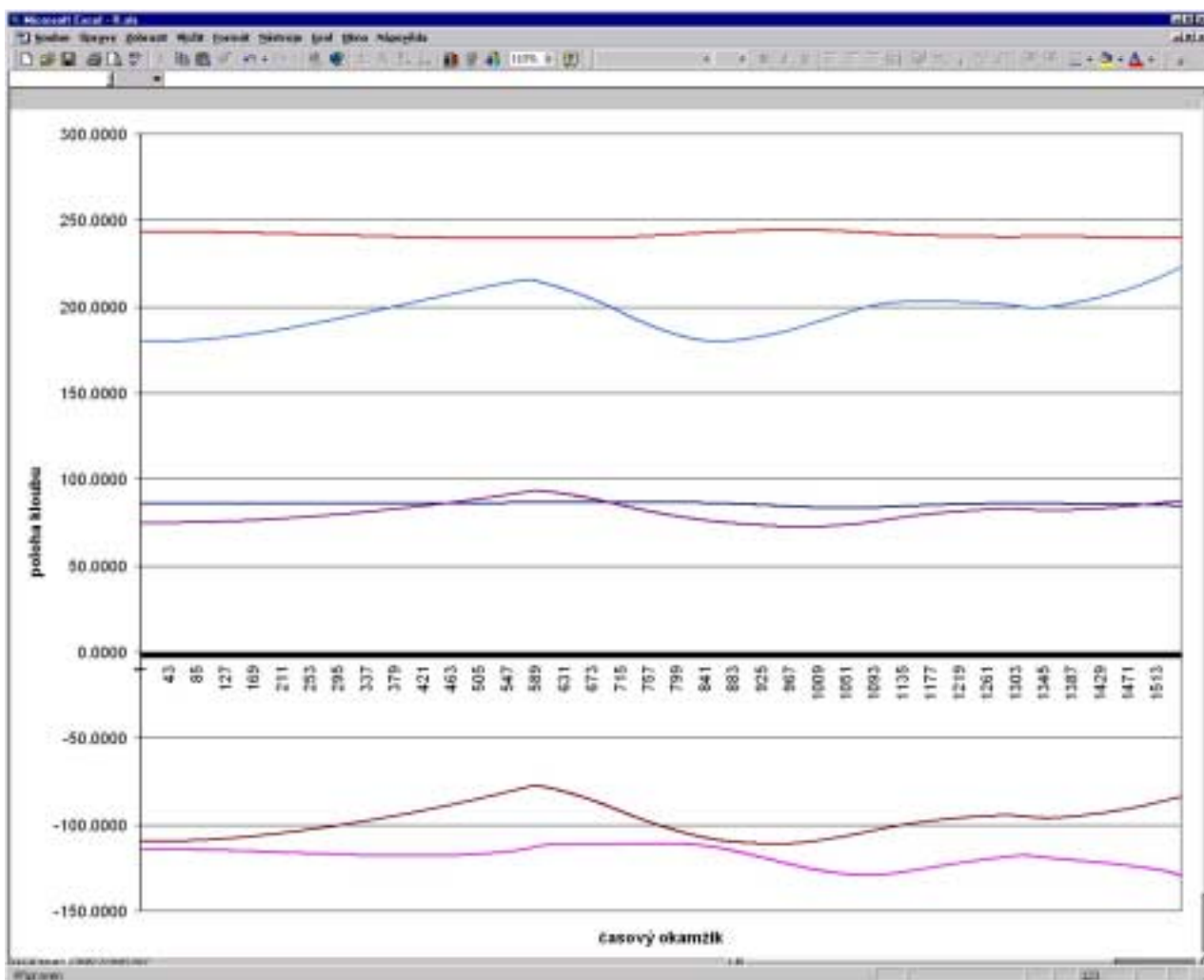
Veškeré v disertační práci uvedené příklady jsou dostupné na doprovodném CD disku ve formě projektových souborů, které lze řídicím programem přeložit a spustit.

Ačkoliv nebyly učiněny pokusy s adaptivním řízením, lze veškeré interpolační funkce poměrně snadno o nejrůznější formy adaptivního řízení doplnit.

## 5.5. Výstup a zpracování dat

Aby bylo možno hodnoty kloubových souřadnic analyzovat, popřípadě použít pro OFF-LINE řízení, bylo potřeba jejich hodnoty zapsat ve vhodném formátu do souboru. K zápisu kloubových souřadnic byla proto navržena instrukce DumpJoints, kterou lze použít kdekoli v programu pro pohyb koncového efektoru k inicializaci, ale i k ukončení výpisu kloubových souřadnic do souboru.

Formát výstupního souboru byl navržen tak, aby jej bylo snadné zpracovat tabulkovým procesorem. Detailnější analýzou v tabulkovém procesoru lze odhalit případné chyby v návrhu interpolačních funkcí.



Obr. 7: Průběh natočení jednotlivých kloubů robotu

Jak je z výše uvedeného obrázku patrné, byl tento záznam pořízen z cca 1540 vzorků, přičemž časová distance mezi vzorky byla 0,01 sec. Pro realizaci ON-LINE řízení je pak nutné řídicímu systému, popřípadě jednotlivým řídicím jednotkám servopohonů předat všechny vzorky.

Na doprovodném CD disku lze najít \*.xls soubory určené pro tabulkový procesor, ve kterých jsou průběhy pro všechny příklady použité v disertační práci.

## 6. ZÁVĚR

S růstem ceny lidské práce a poklesem cen výkonných procesorů a elektroniky všeobecně při pokračujícím zvyšování hustoty integrace a tím i zvyšování spolehlivosti bude jistě docházet i k nárůstu počtu nasazení robotů v praxi. Přitom vyvstane značný tlak na vývoj stále dokonalejšího řídicího softwaru. V současnosti lze pozorovat značné zaostávání řídicích programů nad možnostmi hardwaru a často je oblast softwaru navíc silně podceňována. Zaostávání je do značné míry dáno velkou pracností vývoje řídicího softwaru, snahou stavět na zastaralém nemonulárním softwarovém návrhu a neochota podniknout hloubkové revize stávajícího kódu.

### 6.1. Přínos práce pro rozvoj vědního oboru

Přínos práce pro další rozvoj vědního oboru lze shrnout do několika následujících bodů:

- Kinematické uspořádání stroje – program umožňuje ověření vhodnosti kinematického uspořádání pro jistou technologii, program lze použít k získání představy o pracovním prostoru stroje s jistým kinematickým uspořádáním, základ programu lze použít pro přechod k redundantním kinematickým strukturám aj.
- Konstrukce robotů - možnost ověřit konstrukční návrh z hlediska kolizí během pohybu a to jak jednotlivých článků robotu navzájem tak i vůči předmětům v rámci pracovního prostoru, dosažitelnosti míst v rámci pracovního prostoru robotu, program lze použít pro optimalizaci konstrukce, lze doplnit o funkce zohledňující dynamiku stroje aj.
- Ověřovat návrh koncového efektoru – z hlediska kolizí na pracovišti, dosažitelnosti požadované polohy aj.
- Ověřovat návrh robotizovaného pracoviště – umístění robotů na pracovišti pro dosažení co největší flexibility a optimality výrobního procesu.
- Ladit a ověřovat programy pro robot za pomoci vizuální simulace. Lze tak poměrně snadno odhalit problémová místa na trajektorii pohybu koncového efektoru i v jednotlivých kloubech robotu. Instrukční sadu lze snadno modifikovat a rozšiřovat o nové instrukce.
- Implementovat a ověřovat funkce řídicího systému – interpolační funkce, kinematická analýza a syntéza, lze jej rozšířit o funkce pro komunikaci se sensorickým subsystémem robotu a realizovat funkce adaptivního řízení pro jistou technologii.
- Implementovat a ověřovat funkce nadřazeného CAM jádra pro jisté technologie. Lze rozšířit o moduly CAM jádra které na základě návrhu modelu v CAD systému a dané technologie automaticky generují program pro robot aj.
- Implementovat a ověřovat vyšší formy řízení realizované pomocí neuronových sítí, genetických algoritmů, fuzzy algoritmů aj.

- Dálkový monitoring (simulace) výrobního stroje – monitoring pracovního procesu, pouhým rozšířením o funkce vstupu kloubových souřadnic z čidel polohy.
- Program lze použít jako studijní materiál pro vývoj lépe modularizovaného řídicího systému.

## 6.2. Zhodnocení stávajícího stavu

Pro vstup konfiguračních dat a programu robotu byl zvolen obyčejný textový soubor, k němuž byl vyvinut jednoduchý kompilátor, na místo návrhu mnoha dialogových oken, jejichž implementací by se kód programu stal značně závislý na operačním systému (serveru, který zprostředkuje klientské aplikaci grafický výstup). Ukázalo se, že při dobrém návrhu kompilátoru jej lze snadno realizovat a snadno rozšiřovat o nové povely.

Pro vstup grafických dat do programu byl zvolen dxf soubor, který je kompletní reprezentací výkresu pořízeného CAD systémem. Postupem času se projevila částečná nestálost tohoto formátu, kdy s příchodem nového CAD nástroje docházelo i k drobným úpravám tohoto formátu s nutností upravit zdrojové texty zajišťující jeho interpretaci. Použití CAD systému se ukázalo jako velmi přínosné zejména díky snadné modifikovatelnosti grafického podání během vývoje programu. Díky CAD systému a simulaci je také možné snadno analyzovat pracovní prostor robotu, popřípadě odhalit problémové pohyby (kolize aj.).

Ukázalo se, že bez simulace s grafickým výstupem by bylo odhalení mnoha chyb během vývoje aplikace velmi zdlouhavé ba i nemožné. Navíc simulace pomáhá uživateli během fáze návrhu řídicího programu robotu.

I když program neumožňuje pracovat s jiným kinematickým uspořádáním robotu než výše uvedené, je velká část softwaru navržena tak, aby přechod na jiné kinematické struktury byl snadný. Při přechodu na jinou kinematickou strukturu vyvstává pouze nutnost implementace funkcí kinematické analýzy a syntézy včetně úpravy několika dalších funkcí. Návrh kinematické struktury v CAD systému přitom zůstává stejný a je do jisté míry univerzální.

Funkce kinematické analýzy, tak jak je v programu implementována, je těsně svázána s kinematickým uspořádáním robotu. Ukázalo se však, že tuto funkci lze realizovat univerzálně pro většinu dalších kinematických struktur. U funkce kinematické syntézy nelze univerzálnosti dosáhnout již tak snadno. Možným řešením by však mohla být numerická metoda, avšak s důsledkem na výpočetní náročnost.

Byly rovněž provedeny výkonnostní testy, při nichž bylo zjištěno že většina výpočetního výkonu (90 – 98 %) jde na vrub zobrazování. Díky analytické podobě vztahů pro kinematickou analýzu a syntézu jsou jejich požadavky na výpočetní výkon procesoru minimální. Jelikož je program realizován jako vícevláknová (multi-thread) aplikace, lze vlákna realizující grafický výstup zakázat a provozovat tak program na méně výkonných procesorech (typu I386), což má význam zejména u průmyslových a jednočipových PC !

Z interpolačních funkcí byly pro svou pracnost realizovány jen některé nejzákladnější, ke kterým lze v případě potřeby snadno navrhnout spoustu dalších modifikací. Na interpolační funkce lze nahlížet jako na vrstvu, s níž komunikuje nadřazený software realizující vyšší formu řízení (plánování a tvorba strategie, učení se aj).

Byly provedeny pokusy s časovou synchronizací běhu aplikace s externím hardwarem (jednočipový procesor na paralelním portu počítače). Aplikace se při svém startu pokusila o detekci připojeného hardwaru. Po úspěšné detekci se běh aplikace přepnul do třídy priorit REALTIME a synchronizoval na požadavky hardwaru (hodiny). Během testu bylo dosaženo synchronizace aplikace na hodiny hardwaru se 100 žádostmi o data za sekundu. V tomto režimu běžela i simulace (zobrazování) v reálném čase. Testovacím počítačem bylo PC osazené procesorem Pentium 200 MHz. Pro nasazení v průmyslu by však bylo třeba pro komunikaci a synchronizaci s hardwarem použít rychlejší sběrnice (PCI,USB,SERCOS,CAN Bus nebo IEEE 1394) a vyvinout driver zajišťující komunikaci. Jelikož bylo nutné během vývoje kód s realizací této synchronizace přepracovat, je v současné verzi projektu obsažen pouze použitý ovladač (VirtualDriver), který však není na program napojen.

Jelikož jsou volně dostupné zdrojové texty aplikace, lze aplikaci rozšířit o vlákna (threedy) obsluhující senzorický systém robotu a realizovat tak adaptivní řízení (chování) s různými požadavky. Taktéž pro realizaci softwarových korekcí výrobních nepřesností jednotlivých kloubů robotu by bylo nutné vyměnit funkce kinematické analýzy a syntézy.

S postupem času se ukázala volba platformy osobních počítačů PC (průmyslové PC) jako velmi dobrá a to zejména z následujících důvodů: vysoký výpočetní výkon, dostupnost jak hardwaru tak vývojových nástrojů, vyspělost vývojových prostředků, cena hardwaru a vývojových nástrojů, vyspělost operačních systémů pro tuto platformu včetně pokročilých softwarových technologií. Poslední dobou dochází ke zjednodušení hardwarového návrhu průmyslových PC a lze již pozorovat tendence pro realizaci jednočipových PC systémů.

Pro reálné použití řídicího programu by bylo nutné jej doplnit o další funkční vlastnosti, zejména funkce pro komunikaci s hardwarem, funkce zajišťující nájezd do nulové polohy, kalibrační funkce, funkce pro předzpracování a interpretaci informací získaných ze senzorů aj.

V závěru disertační práce jsou uvedeny náměty pro zdokonalení stávajícího softwaru, týkající se především jeho modularity.

## LITERATURA

- [1] Ránky, P.G., Ho, C.Y.: Robot Modelling.  
IFS (Publications) Ltd., England 1985  
Springer-Verlag Berlin Heidelberg New York Tokyo
- [2] Mácha, J.: Pohybový model šestiosých robotických systémů v návaznosti na návrh robotizovaného pracoviště v AutoCADu.  
Diplomová práce.VUT-FS Brno, Ústav výrobních strojů, systémů a robotiky.,  
Brno 1994.
- [3] Ehrenberger, Z., Kolíbal, Z.: Průmyslové roboty III  
VUT Brno, Brno 1993.
- [4] Helen Custer: Windows NT  
GRADA Publishing , Praha 1994, Original copyright Microsoft Press, 1993.
- [5] Mistrovství ve VISUAL C++, podrobný průvodce tvorbou aplikací.  
Computer Press, Brno 1999, ISBN 80-7226-132-0
- [6] Virius, M.: Programování v C++  
Vydavatelství ČVUT , Praha 1998, ISBN 80-01-01874-1
- [7] Woo, M., Neider, J., Davis, T.: OpenGL Programming Guide  
Addison Wesley Longman, Inc., ISBN 0-201-46138-2
- [8] Piegl, L., Tiller, W.: The NURBS book. – 2nd ed.  
Springer - Verlag Berlin Heidelberg New York, 1997, ISBN 3-540-61545-8
- [9] Sochor, J., Žára, J., Beneš, B.: Algoritmy počítačové grafiky.  
ČVUT Praha, 1998.
- [10] VUKOV : Adaptívny priemyselný robot APR - 20.  
VUKOV Prešov, 1987.
- [11] Autodesk : AutoCAD r. 10, 11. Reference manual.  
Autodesk Ltd.
- [12] Loprais, A.: Mechanika manipulačních zařízení (PRaM).  
Ediční středisko VUT Brno, Brno 1988.
- [13] Dumek, V., Roupec, J.: Programování v jazyku C.  
Ediční středisko VUT Brno, Brno 1992.
- [14] Skalický, J.: Elektrické servopohony výrobních strojů.  
Seminář ICB: Elektrické servopohony a jejich aplikace IX.  
ICB Brno 1999, ISBN 80-902175-9-1
- [15] Bělohoubek, P.: Požadavky na el. servopohony a jejich řízení z hlediska stavby výrobních strojů a průmyslových robotů pro dané technologie.  
Seminář ICB: Elektrické servopohony a jejich aplikace IX.  
ICB Brno 1999, ISBN 80-902175-9-1
- [16] Rubiny, A.,: LINUX device drivers  
O'Reilly & Associates, Inc. 1998, ISBN 1-56592-292-1
- [17] Marcinčin, J., N., Kárník, L.: Problematika zaradenia biorobotiky do štruktúry bioinžinierskych vedeckých disciplín.  
Zborník vedeckých prác. ROBTEP 97, Prešov 1997.
- [18] Fořt, I.: Windows 3.1 techniky programování.

GRADA, Praha 1993.

- [19] Bělohoubek, P.: Filozofie, koncepce a zkušenosti ze stavby řídicích systémů výrobních strojů a průmyslových robotů. Seminář ICB: Řídicí systémy a jejich aplikace I. ICB Brno, 1998, ISBN 80-902175-8-3
- [20] Bernecker und Rainer Industrie-Elektronik Ges.m.b.H. : B&R System 2000. Hardware anwenderhandbuch. v. 1.0 , Eggelsberg 1996.
- [21] Bernecker und Rainer Industrie-Elektronik Ges.m.b.H. : B&R System 2000. PG2000 Software User`s Manual . v. 1.0 , Eggelsberg 1995.
- [22] Skařupa, J., Mostýn, V.: CAD systémy a postupy prediktivního inženýrství aplikované na vývoj PRaM. Zborník vedeckých prác. ROBTEP 97, Prešov 1997.
- [23] Kolíbal, Z., Kadlec, Z.: Robotizované pracoviště pro přípravu polotovárů. Zborník vedeckých prác. ROBTEP 97, Prešov 1997.
- [24] Bělohoubek, P. - Tomek, K. - Šípek, S. - Mácha, J. a kol. : Dílčí zpráva o řešení grantového projektu Řízení a stavba víceosých robotických systémů za rok 1995. VUT-FS, Brno.
- [25] Jančina, J., Pekárek, F.: Kinematika , ALFA Bratislava, 1987.
- [26] Příkryl, H.: Moderní směry ve vývoji digitálních servopohonů. Seminář ICB: Elektrické servopohony a jejich aplikace X. ICB Brno 2000.
- [27] Kašpárek, F., Minárik, M., Nikolov, V., Pecinovský, R., Virius, M.: Co v manuálu nenajdete. Příručka jazyka C ++ pro začátečníky i profesionály. UNIS , Brno 1993.
- [28] REKTORYS a spol.: Přehled užití matematiky I,II Prometheus, Praha 1995, ISBN 80-85849-92-5, 80-85849-62-3
- [29] Bělohoubek, P.: Research in Computer Aided Robot Design in relation to the Virtual Reality. International workshop: Specialized Design Sciences. Pilsen 1994.
- [30] Bělohoubek, P.: Filozofie a vybrané aspekty stavby a řízení moderních výrobních strojů, robotů a výrobních systémů. Seminář ICB: Řídicí systémy a jejich aplikace II. ICB Brno 1999, ISBN 80-86308-01-4
- [31] Štefaňák, P.: Analýza a syntéza struktur mechanismov pohonov a robotov. Veda, Bratislava 1989.
- [32] Chvála, B.,Matička, R.,Talácko, J.: Průmyslové roboty a manipulátory. SNTL, Praha 1990.
- [33] Hosnedl, S., Krátký, J., Černý, J.: CAE of driving mechanisms of machine tools. International workshop: Specialized Design Sciences. Pilsen 1994.
- [34] ČSTVS Pardubice: Řídicí systémy pro průmyslové použití. ČSTVS Pardubice, Pardubice1990, ISBN SO-02-99705-0
- [35] Mostýn, V., Skařupa, J.: Metodika dimenzování pohonů průmyslových robotů a manipulátorů. Zborník vedeckých prác. ROBTEP 97, Prešov 1997.
- [36] Bělohoubek, P., Mácha, J.: Computer Aided Robot Design and Robot Control Příspěvek na mezinárodní konferenci ISMCR '95: Smolenice Castle, Slovakia, 12. až 16. června 1995. , ISBN 80-227-0760-0



- [37] Bělohoubek, P., Šimíček, P., Mácha, J., Tomek, K., Šípek, S.: Řízení robotu ALR-4.1. Příspěvek na konferenci MATAR '96 : Praha 25. až 27. 6. 1996.
- [38] Mácha, J.: Počítačové řízení šestiosého antropomorfního robotu. Příspěvek na konferenci INŽENÝRSKÁ MECHANIKA '97: Svratka 5/1997.
- [39] Kelča, F.: Automatické řízení výrobních strojů. VUT Brno, 1989, ISBN 80-214-0016-1
- [40] SIEMENS : Sinumerik & Simodrive  
Automatisierungssysteme für Beabteilungsmaschinen, Technischer Katalog  
Katalog NC 60.2 · 1996 , Siemens AG 1996.
- [41] Vrátil, Z.: 17. Programátorské rozhraní MS-Windows. GETHON audio and computer, Sokolov 1996.
- [42] Vrátil, Z.: 19. Programátorské rozhraní Windows NT. GETHON audio and computer, Sokolov 1997.
- [43] Šoc, F.: CP řízení a kalibrace robota ASEA IRb6. Zborník vedeckých prác. ROBTEP 97, Prešov 1997.
- [44] Bělohoubek, P.: Elektrické servopohony, jejich řízení a aplikace I. ICB Brno 1998, ISBN 80-902175-4-0.
- [45] Konolige, K.: COLBERT: A language for reactive control in Sapphira.  
[www.ai.sri.com/~konolige/sappihira](http://www.ai.sri.com/~konolige/sappihira)
- [46] Murphy, K., N.: REAL-TIME control system  
User reference manual, 1998, [isd.cme.nist.gov](http://isd.cme.nist.gov)
- [47] QNX Realtime Operating System: [www.qnx.com](http://www.qnx.com)
- [48] Real-time extensions for Microsoft's Windows NT operating system.  
[www.imagination.com](http://www.imagination.com)
- [49] Robotics Research: Dexterous Manipulators: [www.robotics-research.com](http://www.robotics-research.com)
- [50] SERCOS interface: [www.sercos.org](http://www.sercos.org), [www.sercos.com](http://www.sercos.com), [www.sercos.de](http://www.sercos.de)
- [51] RRG Kinematics: [robotics.utexas.edu](http://robotics.utexas.edu)
- [52] Muir P. F., Horner J. W.: Mechatronic objects for real-time control software development.  
International Symposium on Intelligent Systems and Advanced Manufacturing:  
Mechatronics Conference, November 1998, Boston.

## **PUBLIKACE AUTORA**

1. Mácha, J.: Pohybový model šestiosých robotických systémů v návaznosti na návrh robotizovaného pracoviště v AutoCADu.  
Diplomová práce, VUT Brno, 1994.
2. Mácha, J.: Počítačové řízení šestiosého antropomorfního robotu.  
Příspěvek na konferenci INŽENÝRSKÁ MECHANIKA '97: Svratka 12. až 15 května 1997.
3. Mácha, J.: Počítačové řízení šestiosého antropomorfního robotu.  
Teze k rigorózní zkoušce: VUT Brno, květen 1997.
4. Bělohoubek, P., Mácha, J.: Computer Aided Robot Design and Robot Control  
Příspěvek na mezinárodní konferenci ISMCR '95: Smolenice Castle, Slovakia, 12. až 16. června 1995. , ISBN 80-227-0760-0
5. Bělohoubek, P., Šimíček, P., Mácha, J., Tomek, K., Šípek, S.: Řízení robotu ALR-4.1. Příspěvek na konferenci MATAR '96 : Praha 25. až 27. června 1996.
6. Bělohoubek, P., Kolíbal, Z., Mácha, J.: Aplikace statistických metod řízení výrobních systémů. Grantová zpráva (FP 379737) 1997.

# CURRICULUM VITAE

**Ing. Jiří Mácha**  
Podhájí 283  
Uherský Brod - Těšov  
PSC 687 34  
Česká republika  
0042 0633 637906 (So,Ne)  
E-mail : [macha@fme.vutbr.cz](mailto:macha@fme.vutbr.cz)

---

## OSOBNÍ ÚDAJE

NAROZEN : 30.5.1971  
NÁRODNOST : česká

---

## ZAMĚSTNÁNÍ

**Walter s.r.o., Kuřim** **10/1998 - dosud**  
Náplň práce: Vývoj CAM jádra pro 5-ti osý brousící stroj Helitronic Power.

**UNIS spol. s r. o. , Brno** **1/1998 – 10/1998**  
Náplň práce: Vývoj elektronických zařízení - návrh elektrického zapojení, naprogramování jednočipových procesorů, oživení a instalace u zákazníka.

**Realizované projekty:** Návrh jádra měřící části indukčního průtokoměru (software).  
Procesorová jednotka pro regulaci přiměšování pěnídla do vody v požárních vozech (návrh hardwarového zapojení a vývoj softwaru).

---

## VZDĚLÁNÍ

**Třileté postgraduální studium ( doktorandské )** **1994 - dosud**  
VUT Brno , fakulta strojní  
Obor : Konstrukční a procesní inženýrství.

**Vysoká škola** **1989 - 1994**  
VUT Brno , fakulta strojní  
Obor : Výrobní systémy s průmyslovými roboty a manipulátory.

**Střední škola** **1985 - 1989**  
Gymnázium JAK v Uh. Brodě.

---

## ***DALŠÍ ČINNOST***

Vývoj zařízení pro sledování pohybu vozidel pomocí GPS přijímače (návrh hardwaru a softwaru). **1997**

# **ABSTRACT**

## **Computer control of six axis anthropomorphic robot**

The dissertation arise in the context of research in the fields CARD (Computer Aided Robot Design) and CARC (Computer Aided Robot Control) at the Technical University of Brno, Faculty of Mechanical Engineering, Institute of Production Machines and Systems. The direction of the dissertation relates to other researches of this institute.

The dissertation concentrates on the area of anthropomorphic robot control (ALR 4.1), and the connection of the robot computer control and robot design in the CAD system. It has three parts – derivation of mathematical equations, it's implementation in the control program for robot and description of the control program.

It was necessary to derive mathematical equations for forward and inverse kinematics, interpolation functions, the kinematics values validation, model display representation and for data transformation from the CAD system.

The control program is created in object oriented language C++ and its source code is a part of this dissertation. The CAD system is used for robot kinematic structure design and its visual aspect and its output file with data's is used to control program initialization. In addition to the data, the control program is initialized with a file with configuration and movement commands. The control program has interpolation functions for some basic and complex three-dimensional end-effector movements. It is able to graphically simulate movements of the robot and watch all it's joint values on-line. It is possible to make a record of joint coordinate history for other analysis, for example in chart processor. It is possible to use the program for ON-LINE and OFF-LINE control.

The program has a modular structure to be easily extensible to another functional quality like automatic movement instruction generation on the basis CAD drawing, sensor data processing, calibration functions, functions for communication with control units of servodrives, functions for adaptability in specific technology, autodiagnosics, functions for high level control and others.

With connection and outspokenness of researches are facilitated other researches in fields of robot end-effector design, control algorithm development, Virtual Reality, modular construction of control system and construction, sensoric and data processing, the human – computer – robot interaction etc.