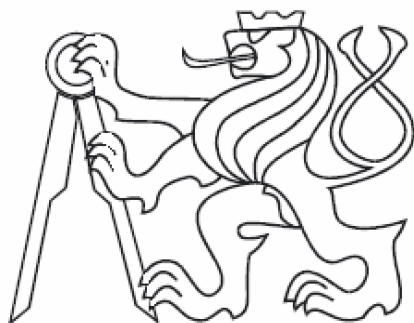


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ
KATEDRA ŘÍDICÍ TECHNIKY



BAKALÁŘSKÁ PRÁCE

**Software pro výpočty a vizualizaci procesních
dat pro řízení CNC frézky**

Praha 2006

Martin Knotek

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze dne 20.6.2006

.....
podpis

Poděkování

Tímto bych rád poděkoval především firmě INTRONIX spol. s r. o. za poskytnutí prostředků a podpory při tvorbě této bakalářské práce. Dále bych rád poděkoval vedoucímu bakalářské práce Ing. Pavlu Burgetovi za jeho ochotu a vstřícný přístup při řešení problémů, a v neposlední řadě svému nejbližšímu okolí za podporu při studiu.

ANOTACE:

Úkolem bakalářské práce bylo vytvořit programové vybavení pro PC připravené na napojení na HW firmy Intronix určeného pro řízení tříosé CNC frézky.

Program byl napsán pro real-time operační systém ETS PharLap a skládá se ze tří bloků.

První blok programu zajišťuje zadání partprogramu pro řízení frézky pomocí G-kódů, možnost editace a uložení tohoto partprogramu, dále kontrolu jeho syntaxe, smyslu a správnosti pro řízení CNC frézky.

Druhý blok vytváří přehledné a snadno ovladatelné grafické prostředí se zobrazením naprogramovaného průběhu obrábění a procesních dat, jako jsou například aktuální polohy os a rychlost pojezdu.

Jako poslední blok byl vytvořen interpolátor pro generování poloh a rychlostí pro obrábění dle naprogramovaného průběhu skutečnou rychlostí v reálném čase a následně simulátor pro grafické zobrazení výstupů interpolátoru. Výstup interpolátoru, kterým je aktuální poloha os, je určen jako vstup do programu zařízení ovládajícího výkonové členy krokových motorů.

THE ANNOTATION:

The goal of this bachelor thesis was to create a PC software prepared for connecting to the Intronix Company hardware. The hardware is designed for three-axis CNC milling machine control.

The program was written for a real-time operating system ETS PharLap and it consists of three blocks.

The first block of the program provides a partprogram specification by means of G-codes, a partprogram editing and saving possibility and a syntax, sense and correctness verification for CNC milling machine control.

The second block creates a well-arranged and easy-to-use graphical environment with the projection of a programmed machining course and of process data, as for example actual positions of axes and the traversing speed.

As the last block, it was written an interpolator for the generating of positions and speeds for the machining according to a programmed course by a real speed in a real time. Subsequently, it was written a simulator for a graphical representation of interpolator outputs. The output of the interpolator, represented by the actual position of axes, is determined as the input to the device program which controls power blocks of step motors.

Obsah

1	ÚVOD.....	1
2	BLOKOVÉ SCHÉMA ŘÍZENÍ CNC FRÉZKY	2
2.1	ZADÁNÍ UŽIVATELSKÉHO PROGRAMU	3
2.2	PRŮMYSLOVÝ POČÍTAČ	3
2.3	VIZUALIZACE	3
2.4	HRADLOVÉ POLE	3
2.5	VÝKONOVÉ ČLENY	4
2.6	FRÉZKA.....	4
3	PROGRAMOVÉ VYBAVENÍ.....	5
3.1	REAL-TIME OPERAČNÍ SYSTEM PHARLAP	5
3.1.1	WIN32 API.....	5
3.1.2	OPERAČNÍ SYSTÉMY PEVNÉHO REÁLNÉHO ČASU.....	5
3.1.3	JEDNOPROCESNÍ OPERAČNÍ SYSTÉM	6
3.1.4	PLÁNOVAČ ROUND-ROBIN	6
3.1.5	CROSS DEBUGGING	6
3.2	VÝVOJOVÉ PROSTŘEDÍ.....	6
4	PROGRAMOVÁNÍ STROJE	7
4.1	ZÁKLADNÍ POJMY.....	7
4.2	ZÁSADY PSANÍ SLOV	7
4.3	TYPY SLOV.....	8
4.4	BLOK PROGRAMU	8
4.5	PODPOROVANÁ SLOVA V PROGRAMU.....	8
4.6	PODROBNÝ POPIS SLOV ZE SKUPINY G.....	9
4.6.1	G0 - POJEZD RYCHLOPOSUVEM.....	9
4.6.2	G1 – LINEÁRNÍ INTERPOLACE	9
4.6.3	G2, G3 - KRUHOVÁ INTERPOLACE	10
4.6.4	G12, G13 – ŠROUBOVICE (ZÁVIT).....	12
4.6.5	PODMÍNĚNÝ A NEPODMÍNĚNÝ SKOK G21, G22	13
4.6.6	POLOMĚROVÉ KOREKCE G40, G41, G42.....	14
4.6.7	POUŽITÍ OFFSETU G43,G44	16
4.6.8	VOLÁNÍ PODPROGRAMU G71.....	17
4.6.9	SPIRÁLOVÁ INTERPOLACE G77, G78	17
4.6.10	ABSOLUTNÍ/RELATIVNÍ SOUŘADNICE G90/G91	18
4.7	PODROBNÝ POPIS SLOV ZE SKUPINY M	19
4.7.1	M0.....	19
4.7.2	M3.....	19
4.7.3	M4.....	19
4.8	PODROBNÝ POPIS OSTATNÍCH SLOV	19

4.8.1	RYCHLOST POSUVU - F	19
4.8.2	RYCHLOST OTÁČENÍ VŘETENA – S	19
4.8.3	FUNKČNÍ SLOVA.....	19
4.8.4	ROZMĚROVÁ SLOVA.....	20
4.8.5	PARAMETRY Q.....	20

5 PROGRAMOVÉ ŘEŠENÍ..... 21

5.1	GRAFICKÉ UŽIVATELSKÉ PROSTŘEDÍ	21
5.1.1	OBRAZOVKA AUTOMATICKÉHO REŽIMU	22
5.1.2	KOMPONENTY OBRAZOVKY AUTOMATICKÉHO ŘÍZENÍ	23
5.2	REPREZENTACE G-KÓDŮ V PROGRAMU.....	24
5.2.1	UCHOVÁNÍ DAT V PROGRAMU	24
5.2.1.1	Převod do spojového seznamu.....	24
5.2.1.2	Typy segmentů spojového seznamu.....	25
5.2.1.3	Ostatní parametry segmentů	26
5.2.2	VÝPOČET KONCOVÝCH SOUŘADNIC SEGMENTŮ.....	26
5.2.2.1	Lineární interpolace	27
5.2.2.2	Kruhová interpolace	27
5.2.2.3	Interpolace po šroubovici	28
5.2.2.4	Interpolace po spirále	28
5.3	VIZUALIZACE PROCESNÍCH DAT	29
5.3.1	GRAF PRŮBĚHU INTERPOLACE.....	29
5.3.1.1	Lineární interpolace	29
5.3.1.2	Kruhová interpolace	29
5.3.1.3	Spirálová interpolace.....	30
5.4	INTERPOLÁTOR	30
5.4.1	PRINCIP INTERPOLÁTORU	30
5.4.1.1	Hodnoty vypočtené před začátkem inerpolace	31
5.4.1.2	Rampa zrychlení, rampa zpomalení.....	31
5.4.1.3	Brzdná dráha.....	32
5.4.2	VÝPOČTY PRO INTERPOLÁTOR.....	32
5.4.2.1	Lineární interpolace	32
5.4.2.2	Kruhová interpolace	32
5.4.2.3	Šroubovicová interpolace	33
5.4.2.4	Spirálová interpolace.....	33

6 ZÁVĚR..... 35

1 Úvod

Frézka je zařízení sloužící k obrábění výrobků z kovových, plastických, dřevěných i jiných materiálů. Pro dosažení požadovaných tvarů je nutno ovládat pohyb nástroje, v případě tříosé frézky ve třech osách, a to X, Y a Z, v požadovaných směrech, zadanou rychlostí.

Rychlost obrábění je mimo jiné závislá na druhu obráběného materiálu a použitého nástroje.

V minulosti se většinou jednalo o konvenční obráběcí stroje, ručně ovládané podle určeného pracovního postupu. Pro zvýšení produktivity práce jsou v dnešní době tyto frézy vybaveny elektronickými prostředky pro výpočty a číslicové řízení (numeric control) pohybů suportu a řízení počtu otáček vřetena podle předem zadaného programu.

V případě číslicového řízení se nejčastěji jedná o CNC stroje (Computer Numerical Control). Tyto stroje lze programovat pomocí takzvaných G-kódů. G-kódy jsou nejrozšířenějším programovacím jazykem pro ovládání NC a CNC strojů. Poprvé byly použity na začátku šedesátých let a konečné úpravy syntaxe byly provedeny v únoru 1980. Díky tomu, že se programovací jazyk dále oficiálně nerozvíjel, vzniklo mnoho různých variant doplňků a změn, které prováděli sami výrobci CNC strojů. Z toho vyplývá nutnost důkladně se seznámit s programováním daného stroje před jeho použitím.

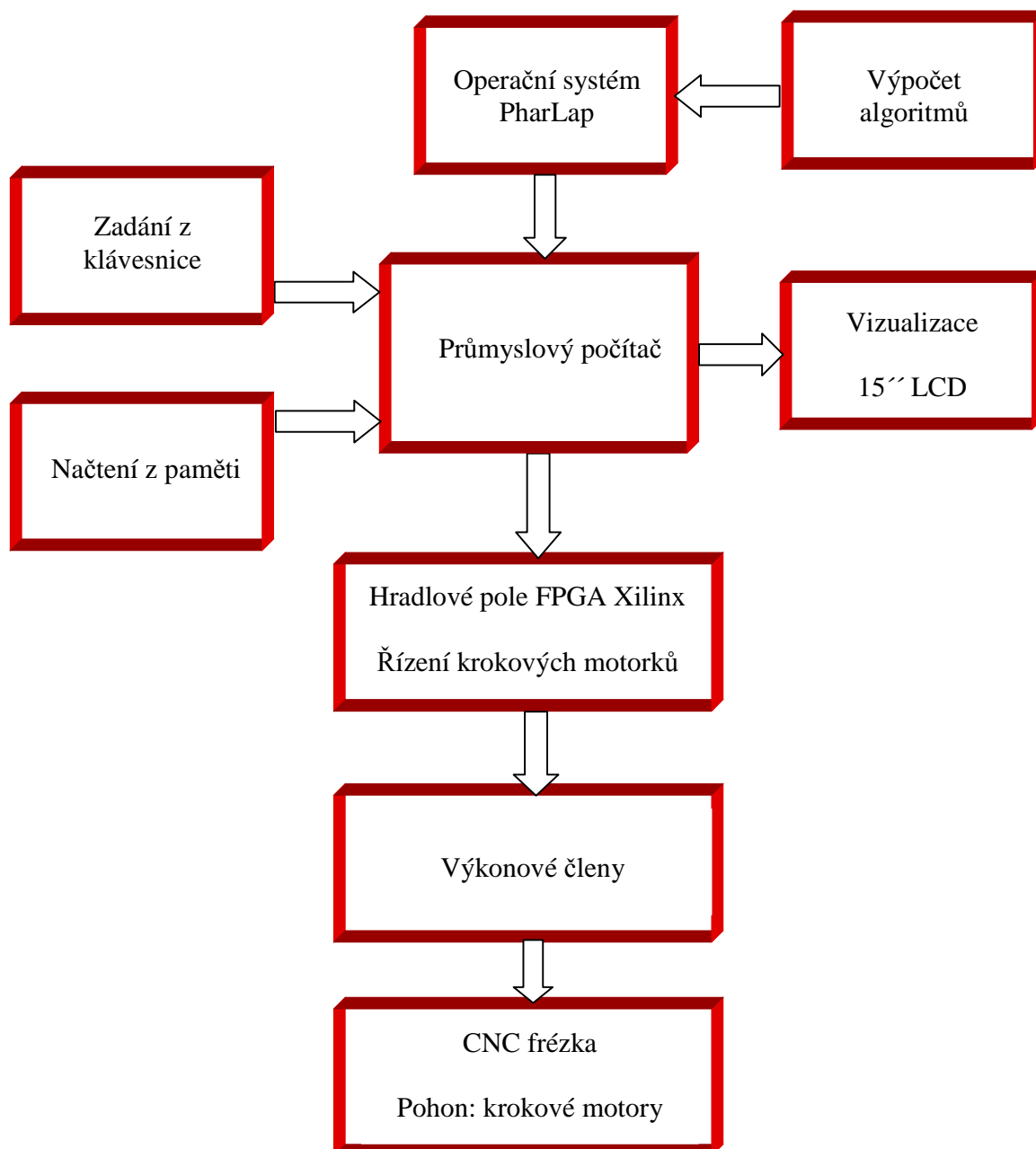
Bakalářská práce, zadaná a vykonávaná ve firmě INTRONIX spol. s r.o., se zabývá realizací programu určeného pro výpočty a vizualizaci procesních dat pro řízení CNC frézky a je rozčleněna do dvou hlavních částí.

První část bakalářské práce je tvořena druhou, třetí a čtvrtou kapitolou. V druhé kapitole je popsán princip a blokové schéma řízení CNC frézky, ve třetí kapitole se zabývám použitým real-time operačním systémem ETS PharLap a v poslední, čtvrté kapitole této části se zabývám pravidly programování, které vycházejí z normy pro řízení obráběcích strojů pomocí G-kódů.

Druhá část bakalářské práce již popisuje způsob, jakým jsem řešil zadané úkoly. Podkapitoly obsahují popis řešení grafického uživatelského prostředí, převodu G-kódů z programu zadaného uživatelem do vhodné reprezentace uživatelského programu, zobrazení uživatelem naprogramovaných průběhů a vytvoření simulátoru průběhu obrábění skutečnou rychlostí v reálném čase.

2 Blokové schéma řízení CNC frézky

Na Obr. 1 je vyznačeno funkční blokové schéma řízení CNC frézky, jak je vyvíjeno ve firmě Intronix. Kapitola dále pokračuje popisem jednotlivých bloků.



Obr. 1: Blokové schéma řízení CNC frézky

2.1 Zadání uživatelského programu

G-kód je programován uživatelem pomocí klávesnice nebo může být načten z paměti. Po dokončení programování je uložen na disk průmyslového počítače a odtud může být později vyvolán k provedení automatického řízení.

2.2 Průmyslový počítač

Použití průmyslového počítače je výhodné především z důvodů jeho malých rozměrů a nízké spotřeby. Vzhledem k výhodám univerzálnosti použití a nízké ceny byl zvolen průmyslový počítač s architekturou 0x86. Použitým operačním systémem je ETS PharLap. Výhodou tohoto real-time operačního systému jsou malé nároky na diskový prostor, protože základní verze se vejde již do 100kB diskového prostoru. Popisu operačního systému ETS PharLap se věnuji v samostatné kapitole. Stejně tak dobře je ale možné místo průmyslového počítače použít standardní PC, přičemž ale ztratíme výhody v podobě malých rozměrů a spotřeby.

Mnou vytvořený program je uložen na disku průmyslového počítače, který se po spuštění počítače zavede do paměti a automaticky spustí.

2.3 Vizualizace

Pro vizualizaci je použit LCD monitor s minimální rozlišovací schopností 1024x768 bodů, který je připojen na digitální grafický výstup průmyslového počítače.

2.4 Hradlové pole

Programovatelná logická pole jsou široce využívanou a efektivní cestou pro realizaci rozsáhlých kombinačních a sekvenčních logických obvodů. Jejich hlavní výhodou je vysoký stupeň integrace obvodu, který tak soustřeďuje v jednom pouzdře všechny prvky nutné pro běh obvodu.

Velkou výhodou obvodů je jejich rychlost. Vnitřní propojení mezi jednotlivými prvky obvodu a tedy jeho funkce je programovatelná.

Hradlové pole se v tomto případě používá k příjmu a transformaci řídicích signálů z průmyslového počítače a následným ovládním výkonových členů krokových motorů.

2.5 Výkonové členy

Výkonové členy mají za úkol zesílit řídicí signály, výkonově je přizpůsobit a ovládat krokové motory

2.6 Frézka

Jedná se o existující frézku menších rozměrů, není ale problémem navržené zapojení a vytvořený program s malými změnami použít pro jiné typy a velikosti zařízení.

3 Programové vybavení

Tato kapitola se zabývá popisem použitého real-time operačního systému PharLap od firmy Ardence a jeho výhodami.



3.1 Real-time operační system PharLap

Ardence ETS je kompaktní operační systém na bázi aplikačních funkcí Win32 API. Má vlastní 32bitové mikrojádro, které zabírá méně než 100 kB, a deterministický plánovač typu Round-robin.

ETS dává programátorům možnost nastavit programovým vláknům libovolnou úroveň z 256 definovaných úrovní priorit. Plánovač běhu vláken programu eliminuje vznik chybového efektu inverze priorit (dead-lock). V jádru ETS jsou implementovány binární synchronizační objekty, jako semaforey, kritické sekce, obsluha událostí (events) a správa paměti, která se programátorům jeví jako lineárně adresovatelný prostor do velikosti 4 GB.

3.1.1 WIN32 API

Díky použití programátorského rozhraní WIN32 API, které používá většina programátorů a to i v řídicích aplikacích, je zajištěna rychlá adaptabilita programátora na programování pro tento operační systém.

3.1.2 Operační systémy pevného reálného času

Operační systém pevného reálného času (hard real-time) je obecně takový systém, jehož správná činnost závisí nejen na výsledcích, které z něj vzejdou, ale i na tom, kdy budou získány. Zajímá nás především nejhorší možná doba odezvy systému. Operační systémy pevného reálného času se používají především pro systémy, kde může mít nedodržení časového limitu pro určitou operaci katastrofické následky, jako například zničení zařízení nebo zranění člověka.

3.1.3 Jednoprocesní operační systém

PharLap ETS používá vlákna pouze k provádění více úloh v rámci jednoho procesu, takže se jedná o jednoprocesní operační systém. Z toho vyplývá, že má nejmenší možnou režii pro přepínání kontextů úloh. Je tomu tak proto, že je zde zapotřebí nejmenší možné množství registrů a dalších informací, které musí být ukládány a obnovovány v průběhu přepínání výkonu základních úloh pro jednotlivá vlákna. Vlákna jsou uložena ve společné paměti, procesy jsou v paměti vzájemně odděleny.

3.1.4 Plánovač Round-robin

Každý proces dostává CPU cyklicky jen po malou jednotku času – časové kvantum (time slice). Po uplynutí časového kvanta dostává CPU nejstarší proces ve frontě připravených procesů dané priority a dosud běžící proces se zařazuje na konec této fronty.

3.1.5 Cross debugging

Aplikace se ladí v hostitelském počítači a cílový strojový kód je při ladění přenesen pomocí sériového nebo paralelního portu, od nejnovější verze i Ethernetem do cílového počítače. Aplikace je laděna pomocí tzv. křížového ladění – cross debugging interaktivně v cílovém počítači, ale na obrazovce hostitelského počítače.

3.2 Vývojové prostředí

Jako vývojové prostředí (IDE) používá ETS Microsoft Visual Studio C++ nebo Microsoft .NET. V tomto prostředí je vedena správa projektu, editace zdrojových textů programů, kompilování a ladění aplikace. K propojení kompilovaného kódu (linking) používá ETS svůj Linker/Locator, který se po nainstalování balíčku SDK (Software Development Kit) stane spolu s nástrojem Crossdebugger součástí aplikace MS Visual Studio.

4 Programování stroje

Řídicí systém umožňuje řídit stroj v automatickém režimu podle předem zadaného programu. Programy je možno uchovávat v souborech v zálohované paměti počítače.

Pro programování je možno použít tzv. G-kódů, v této kapitole je popsán způsob programování stroje pomocí těchto kódů

4.1 Základní pojmy

Program obsahující G-kódy je sestaven z řady jednotlivých elementárních operací, tzv. **bloků**. Blok se skládá z dílčích údajů, tzv. **slov**. Každé slovo pak obsahuje (až na některé výjimky) **adresu**, udávající, o jaký druh informace jde a **číselný údaj**, udávající rozměrovou hodnotu nebo kódové přiřazení k nějaké funkci či operaci.

Příklad bloku partprogramu:

N20 G1 X10.355 Y1.50 Z2.5 F30

Slova v tomto bloku jsou : N20, G1, X10.355, Y1.50, Z2.5, F30

Adresami jsou .: N, G, X, Y, Z, F

Číselné údaje jsou: 20, 1, 10.355, 1.50, 2.5, 30

4.2 Zásady psaní slov

Pro psaní slov je nutné dodržovat následující zásady:

- v rámci slov nejsou povoleny mezery, zatímco mezi slovy jsou mezery vyžadovány
N10 X1 Y1 – korektní zápis
N 10X 1 Y1 – chybný zápis
- při zápisu čísla jsou nuly před číslem ignorovány, tj. zápis **X1** je stejný jako **X001**
- při zápisu desetinného čísla jsou nuly za číslem ignorovány, tj. zápis **X0.1** je stejný jako **X0.100**.
- znaménko „-“ musí být uváděno, zatímco znaménko „+“ uváděno být nemusí.
- slova mající význam nulové souřadnice (**X0**, **Y0**, **I0**, atd.) nemusí být zapisována, např.:
N10 X1 Y0 – má stejný význam jako
N10 X1

4.3 Typy slov

Podle platnosti:

Modální – Slova která mají trvalou platnost, tj. do konce programu nebo do výskytu slova s opačným významem.

Nemodální – Slova která mají význam pouze v bloku, ve kterém jsou použity.

Podle informace kterou nesou:

Rozměrová slova – slova nesoucí údaj o rozměru, který je vždy v milimetrech. Patří sem slova X,Y,Z,I,J,R.

Programová slova – jejich číselná hodnota nemá rozměrový význam.

4.4 Blok programu

Blok programu musí vždy začínat slovem N, hodnota za slovy N musí být v jednotlivých blocích vzestupná.

Komentáře lze do textu vkládat za znaky „//“, znaky následující za těmito dvěma znaky až do konce řádky budou ignorovány.

4.5 Podporovaná slova v programu

Adresa	Číselný údaj	Význam
N	Větší než nula.	Identifikuje bloky, musí být uveden v bloku jako první
G	0	Pojezd rychloposuvem
	1	Lineární interpolace
	2	Kruhová interpolace ve směru hodinových ručiček
	3	Kruhová interpolace proti směru hodinových ručiček
	12	Šroubovice ve směru hodinových ručiček
	13	Šroubovice proti směru hodinových ručiček
	21	Podmíněný skok
	22	Nepodmíněný skok
	40	Zrušení poloměrové korekce
	41	Poloměrová korekce vlevo
	42	Poloměrová korekce vpravo
	43	Zařazení offsetu
	44	Vyřazení offsetu
	71	Volání podprogramu
77	Spirála po směru hodinových ručiček	

	78	Spirála proti směru hodinových ručiček
	90	Následující souřadnice jsou chápány jako absolutní
	91	Následující souřadnice jsou chápány jako relativní
M	3	Start otáčení nástroje
	0	Stop otáčení nástroje
S	Větší než nula	Rychlost otáčení nástroje v ot./min.
F	0 až Fmax	Rychlost posuvu v programovém režimu
X		Poloha v ose X
Y		Poloha v ose Y
Z		Poloha v ose Z
I		Poloha středu v ose X pro kruhovou interpolaci
J		Poloha středu v ose Y pro kruhovou interpolaci
R		Poloměr oblouku pro kruhovou interpolaci
H		Počet stupňů pro realizaci šroubovice
k		Stoupavost poloměru spirály v mm za otočku
Q		Parametr
ABS[]		Absolutní hodnota čísla v závorkách
SQRT[]		Druhá odmocnina čísla v závorkách
SIN[]		Sinus čísla v závorkách
COS[]		Cosinus čísla v závorkách

4.6 Podrobný popis slov ze skupiny G

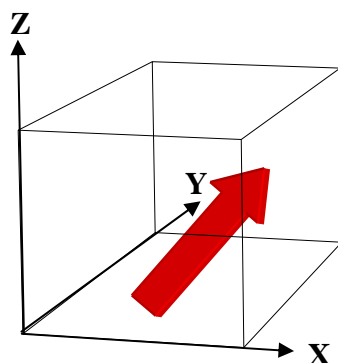
V této kapitole jsou popsána systémem podporovaná slova s adresou G.

4.6.1 G0 - pojezd rychloposuvem

Slouží pro pojezd rychloposuvem do zadaného bodu. V bloku jsou přípustná slova X, Y a Z pro zadání cílové polohy. Podle použití slov G90 nebo G91 jsou zadané souřadnice chápány jako absolutní nebo relativní vzhledem k současné poloze. Hodnota rychlosti při rychloposuvu se zde neprogramuje, je možné ji měnit v uživatelských konstantách stroje.

4.6.2 G1 – lineární interpolace

Slouží pro pojezd po přímce mezi aktuálním a zadaným bodem zadanou rychlostí. Podle použití slov G90 nebo G91 jsou zadané souřadnice chápány jako absolutní nebo relativní vzhledem k současné poloze. Rychlost musí být zadána buď v tomto nebo kterémkoli předešlém bloku.



Obr. 2: Lineární interpolace v prostoru

Příklad:

Systém se nachází v bodě $X=2, Y=1, Z=2$

Blok: **N10 G90 G1 X1 Y1 Z1** způsobí pojezd do bodu $X=1, Y=1, Z=1$

Blok: **N15 G91 G1 X1 Y1 Z1** způsobí pojezd do bodu $X=3, Y=2, Z=3$

4.6.3 G2, G3 - Kruhová interpolace

Slouží pro pojezd po kruhovém oblouku mezi aktuálním a zadaným bodem zadanou rychlostí v rovině X,Y. Rychlost musí být zadána buď v tomto nebo kterémkoli předešlém bloku. Podle použití slov G90 nebo G91 jsou zadané souřadnice cílového bodu chápány jako absolutní nebo relativní vzhledem k současné poloze.

G2– interpolace ve směru hodinových ručiček

G3– interpolace proti směru hodinových ručiček

Střed kružnice lze zadat dvěma způsoby:

a) *Pomocí souřadnic středu I a J.*

V bloku se musí vyskytovat slova I a J udávající polohu středu kružnice **relativně** vzhledem k počátku kružnice nezávisle na použití slov G90 nebo G91. Při překladu je kontrolováno, zda leží počáteční a koncový bod na kružnici s přesností $1\mu\text{m}$. Pokud neleží, je zobrazeno příslušné chybové hlášení.

Příklad:

Systém se nachází v bodě $X=0, Y=1$.

Blok: **N20 G2 X0 Y3 I0 J1** způsobí pojezd po kružnici z bodu $X=0, Y=1$ do bodu $X=0, Y=3$ po směru hodinových ručiček.

b) Pomocí poloměru kružnice R.

V bloku se musí vyskytovat slovo R udávající poloměr kružnice. Při překladu je kontrolováno, zda není koncový bod od počátečního vzdálenější než dvojnásobek poloměru. Pokud je vzdálenější, je zobrazeno příslušné chybové hlášení.

Příklad:

Systém se nachází v bodě $X=1, Y=1$.

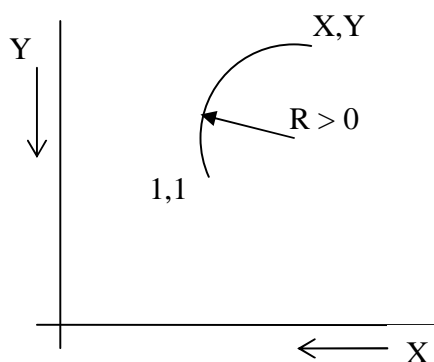
Blok: **N25 G2 X2 Y2 R2** způsobí pojezd po kružnici z bodu $X=1, Y=1$ do bodu $X=2, Y=2$ po směru hodinových ručiček.

Omezení: Použití slov I,J a R současně v jednom bloku není povoleno a způsobí vyvolání chyby.

Na následujících obrázcích jsou zobrazeny kombinace interpolace ve směru a proti směru hodinových ručiček spolu v kladným a záporným poloměrem R při stejném počátečním a konečném bodě interpolace. Výsledná trajektorie může ve většině případů nabývat jeden ze čtyř tvarů

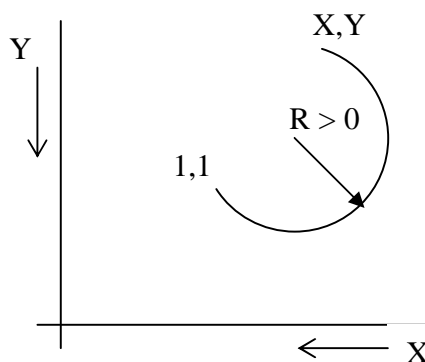
Interpolace

G2

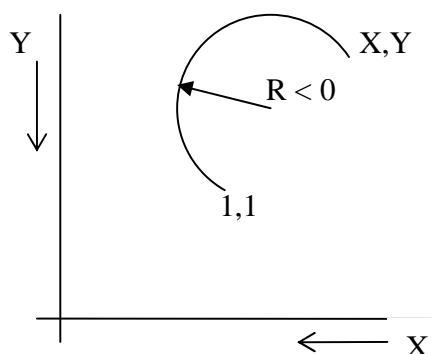
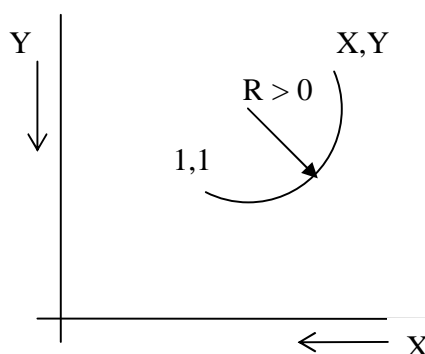


Obr. 3: Interpol. G2 ve směru, $R>0$

G3



Obr. 4: Interpol G3 proti směru, $R>0$

Obr. 5: Interpol. G2 ve směru $R < 0$ Obr. 6: Interpol. G3 proti směru $R < 0$

4.6.4 G12, G13 – Šroubovice (závit)

Slouží pro pojezd po kruhovém tvaru vnitřního nebo vnějšího závitu se zadanou hodnotou stoupání závitu. V rovině X,Y se nástroj pohybuje kruhovou interpolací a ve směru osy Z lineární interpolací. Rychlost musí být zadána buď v tomto nebo kterémkoli předešlém bloku. Podle použití slov G90 nebo G91 je zadaná souřadnice cílového bodu osy Z chápána jako absolutní nebo relativní vzhledem k současné poloze.

Rozlišení jestli jde o frézování vnějšího nebo vnitřního závitu závisí na typu zvolené poloměrové korekce. Střed šroubovice se zadává slovy I,J a to relativně vzhledem k počátku šroubovice, nezávisle na použití slov G90 nebo G91, stejně jako u kruhové interpolace. Slovem H je zadán počet stupňů, které má nástroj v rovině X,Y opsat.

G12 – interpolace ve směru hodinových ručiček

G13 – interpolace proti směru hodinových ručiček

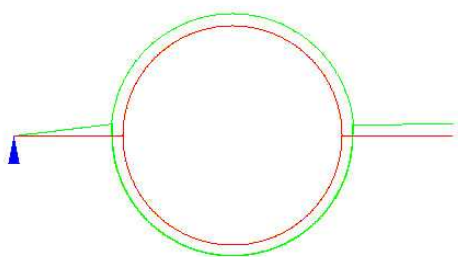
Příklad:

System se nachází v bodě $X=1, Y=1, Z=2$

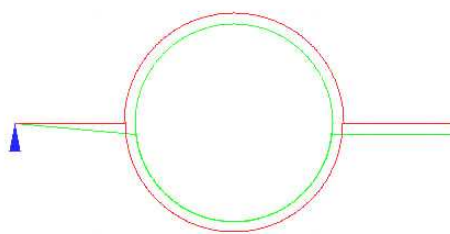
Blok: **N12 G90 G12 I1 J0 Z-10 H540** způsobí pojezd do bodu $X=3, Y=1$, v průběhu interpolace se opíše kružnice v rovině X,Y o délce 540° , při níž v ose Z sestoupí do bodu $Z=-10$. Jde o interpolaci ve směru hodinových ručiček.

Blok: **N15 G91 G13 I1 J0 Z-10 H720** způsobí pojezd do bodu $X=1$, $Y=0$, v průběhu interpolace se opíše kružnice v rovině X,Y o délce 720° , při níž v ose Z sestoupí do bodu $Z=-8$. Jde o interpolaci proti směru hodinových ručiček.

Na následujících obrázcích jsou zachyceny průběhy šroubovice z programu a ukazují rozdíly mezi zvolenými typy poloměrové korekce. Zelenou čarou je vyznačen průběh ekvidistanty, po které se pohybuje nástroj.



Obr. 7: Vnější závit



Obr. 8: Vnitřní závit

4.6.5 Podmíněný a nepodmíněný skok G21, G22

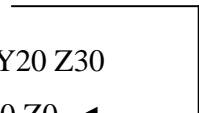
Umožňuje provádět podmíněné a nepodmíněné skoky na zadaná návěští v programu. Návěštím začíná každý platný řádek programu a je uvozen písmenem N a číslem, které je oproti předchozímu řádku stoupající.

Oba typy příkazu obsahují písmeno L a číslo, kterým je udáno návěští, na které má program skočit. U podmíněného skoku navíc následuje podmínka, která je splněna jen tehdy, pokud je nenulová. Pro účely podmínek lze využít parametrů, viz. kapitola 4.8.5

Příklad:

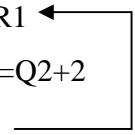
Nepodmíněný skok:

```
N1 G90
N5 G0 X0 Y0 Z0 F200
N10 G22 L20
N15 G1 X50 Y20 Z30
N20 G1 X1 Y0 Z0
```



Podmíněný skok:

```
N1 G90
N5 G0 X0 Y0 Z0 F200
N10 Q1=3 Q2=2
N15 G2 XQ2 Y0 R1
N20 Q1=Q1-1 Q2=Q2+2
N25 G21 L15 Q1
```



Z příkladu nepodmíněného skoku a vyznačené šipky vidíme, že příkaz na řádce s návěstím N15 se nikdy neprovede.

Pro příklad podmíněného skoku se používají operace s parametry, viz. kapitola 4.8.5. Uvedený program provede interpolaci třech po sobě jdoucích půlkružích s poloměrem 1. Podmínkou cyklu je parametr Q1, jehož prvotní hodnotou se nastavuje počet cyklů. Uvnitř cyklu se musí parametr Q1 vhodně zvoleným krokem snižovat. V případě, že je Q1 rovno nule, pokračuje program dalším příkazem na řádce následujícím po N25.

V systému je také zavedena ochrana proti nekonečnému cyklu, nebo cyklu s velmi velkým počtem opakování. V tomto příkladu podmíněného skoku by byl tento mechanismus aktivován v případě, že by poslední příkaz realizoval skok například na návěstí N10.

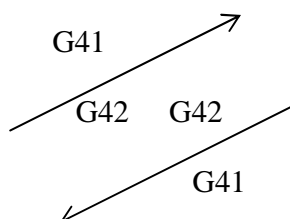
4.6.6 Poloměrové korekce G40, G41, G42

Umožňují programovat obrys výrobku a přitom střed nástroje vést po ekvidistantě, aby mohla být korigována nenulová šířka hrotu nástroje a tvar obrobku odpovídal požadovanému obrysu. Při nespojitosti dvou po sobě následujících bloků dojíždí střed nástroje na průsečík ekvidistant.

Poloměrová korekce je účinná na konci bloku, ve kterém byla vyvolána funkcí G41 nebo G42, tj. začne platit v následujícím bloku. Vyvolání (aktivování funkcí G41 nebo G42) a odvolání (vyřazení funkcí G40) poloměrové korekce může být pouze v bloku s lineární interpolací. Nelze zařazovat a vyřazovat poloměrovou korekci na kružnici, šroubovici nebo spirále.

Pro programování poloměrové korekce G41, G42 platí následující pravidla:

Korekce G41/G42 je trvalá funkce a platí až do odvolání funkcí G40. Korekce může trvat i v blocích bez pohybu.



G41 je korekce vlevo, tj. střed nástroje se pohybuje (pokud je poloměrová korekce kladné číslo) vlevo od programované dráhy ve směru pohybu.

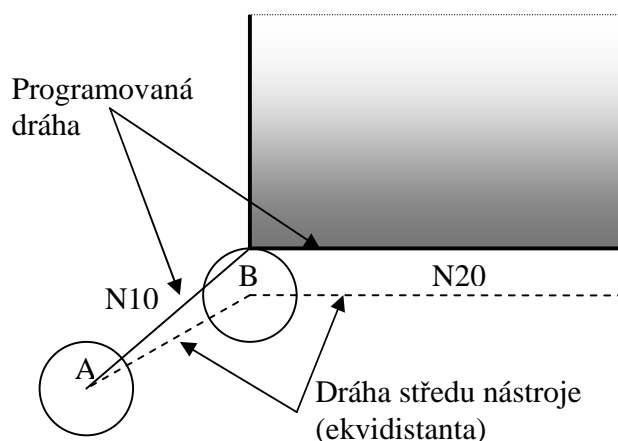
G42 je korekce vpravo, tj. střed nástroje se pohybuje (pokud je poloměrová korekce kladné číslo) vpravo od programované dráhy

Obr. 9: Korekce ve směru pohybu.

Pokud by hodnota korekce bylo záporné číslo, vyměnily by se strany korekcí, tj. G41 se zápornou korekcí je totéž jako G42 s kladnou korekcí.

Zařazení poloměrové korekce

Korekci lze zařadit pouze při lineární interpolaci. Dráha středu nástroje se pohybuje z počátečního bodu bloku N na kolmici k dráze bloku N+1, tj. blok N+1 se celý provede již se zařazenou korekcí. Na následujícím obrázku je v bloku N10 zařazena poloměrová korekce G42, korigovaná dráha je znázorněna čárkovaně z bodu A do bodu B. Bod B, na který najíždí, leží na kolmici k dráze bloku N20. Stejným způsobem se určí bod pro nasazení korekce v případě, že následuje kružnice.



Obr. 10: Zařazení poloměrové korekce

Vyřazení poloměrové korekce

Odvolání poloměrové korekce se provede funkcí G40 a je možné, stejně jako řazení G41, G42, pouze při lineární interpolaci. Koncový korigovaný bod posledního bloku před G40 leží na kolmici k tečně, která prochází koncovým bodem programovým.

4.6.7 Použití offsetu G43,G44

Po zařazení offsetu bude ke všem hodnotám souřadnic v následujících blocích přičítána aktuální hodnota offsetu nezávisle pro souřadnice X, Y a Z. Offset je možno použít k posunu celého grafu v ose X, Y i Z bez nutnosti zásahu do programu.

Pro zařazení offsetu slouží modální slovo G43, tj. bude platit až do výskytu slova G44. Chování funkce G43 závisí na tom zda je aktivní funkce G90 nebo G91.

Zařazení offsetu při aktivní funkci G90 tj. absolutní zadání souřadnic

Při každém následujícím výskytu slova s významem souřadnice v ose X, Y nebo Z tj. i slov I a J bude k jejich číselné hodnotě přičtena hodnota offsetu.

Zařazení offsetu při aktivní funkci G91 tj. relativním zadání souřadnic

Při prvním následujícím výskytu slova s významem souřadnice v ose X, Y nebo Z tj. i slov I a J bude k jejich číselné hodnotě přičtena hodnota offsetu.

Vyřazení offsetu při aktivní funkci G90 tj. absolutním zadání souřadnic

Offset nebude dále přičítán k žádné číselné hodnotě slova.

Vyřazení offsetu při aktivní funkci G91 tj. relativním zadání souřadnic

Při prvním následujícím výskytu slova s významem souřadnice v ose X, Y nebo Z tj. i slov I a J bude od jejich číselné hodnoty odečtena hodnota offsetu.

4.6.8 Volání podprogramu G71

Slouží pro jednoduché nebo opakované vložení volání podprogramu z místa definovaného uživatelem.

Příklad:

N80 G71 tvar3.par 5

V tomto případě dojde v bloku N80 postupně k pětínásobnému zavolání podprogramu s názvem tvar3.par, po skončení opakovaného volání podprogramu se bude pokračovat následujícím blokem v hlavním programu.

4.6.9 Spirálová interpolace G77, G78

Slouží pro pojezd po spirále a tím pro postupné ubírání materiálu se zadaným konečným poloměrem a stoupavostí poloměru spirály v rovině X,Y s konstantní polohou osy Z, a to buď ve směru nebo proti směru otáčení hodinových ručiček. Rychlost musí být zadána buď v tomto nebo kterémkoli předešlém bloku. Použití slov G90 a G91 nemá na samotné souřadnice spirály vliv, protože je její střed dán konečnými souřadnicemi předchozího bloku.

Příklad:

Blok: **N10 G77 R5 k0.1** způsobí pojezd po spirále ve směru hodinových ručiček.

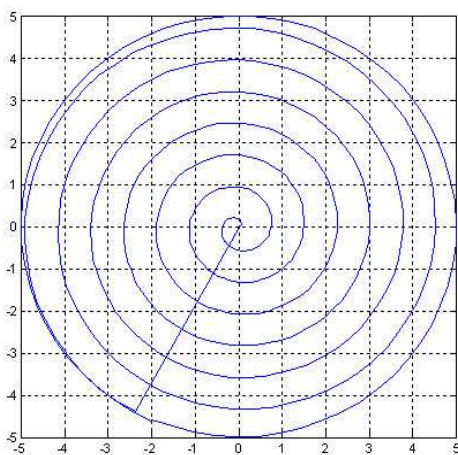
Blok: **N15 G78 R5 k0.1** způsobí pojezd po spirále proti směru hodinových ručiček.

V obou předchozích případech se jedná o spirálu, jejíž konečný poloměr je 5mm. Stoupavost poloměru je dána podílem konečného poloměru a celkového úhlu otočení spirály. Konstantu volí uživatel pomocí písmene 'k'. Díky konstantní stoupavosti poloměru jsou vzdálenosti mezi jednotlivými kruhovými oběhy spirály stejné.

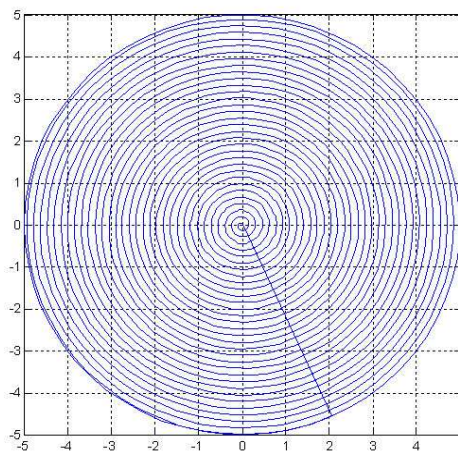
Poté co spirála dosáhne svého konečného poloměru se ještě kruhovou interpolací opíše kruh s konstantním poloměrem, který je rovný konečnému poloměru. Tím se provede začištění spirálového průběhu na kruhový.

Po provedení zakončovacího kruhu se nástroj vrátí zpět do středu spirály a následující pohyb vychází právě z něj.

Zde jsou uvedeny dva příklady ukazující vliv různé hodnoty stoupavosti poloměru k , při stejném poloměru spirály $R = 5$.



Obr. 11: Spirála s velkým k



Obr. 12: Spirála s malým k

4.6.10 Absolutní/relativní souřadnice G90/G91

Slouží pro volbu chápání zadaných souřadnic jako relativní nebo absolutní. Obě funkce jsou trvalé, tj. platí až do příchodu protikladné funkce nebo do konce programu.

Absolutní souřadnice G90

Všechna následující slova s významem souřadnic tj. slova X,Y,Z jsou chápána absolutně vzhledem k aktuálnímu nulovému bodu.

Absolutní souřadnice G91

Všechna následující slova s významem souřadnic tj. slova X,Y,Z jsou chápána relativně vzhledem k počátečnímu bodu pohybového bloku.

Slova I,J jsou chápána vždy relativně!

4.7 Podrobný popis slov ze skupiny M

V této kapitole jsou popsána systémem podporovaná slova s adresou M.

4.7.1 M0

Slovo M0 deaktivuje všechny dosud aktivní slova M.

4.7.2 M3

Slovo M3 aktivuje otáčení vřetena aktuálně zadanou rychlostí S.

4.7.3 M4

Slovo M4 deaktivuje otáčení vřetena.

4.8 Podrobný popis ostatních slov

V této kapitole jsou popsána ostatní systémem podporovaná slova.

4.8.1 Rychlost posuvu - F

Zadává rychlost posuvu použitou v příkazech interpolace. Rychlost je zadaná v mm/min. Zadání rychlosti je chápáno jako globální a není omezeno pro současný blok, je platné až do dalšího výskytu slova F nebo do konce programu.

4.8.2 Rychlost otáčení vřetena – S

Zadává rychlost otáčení vřetena v ot./min.

4.8.3 Funkční slova

Realizují matematické funkce

ABS: Absolutní hodnota ze zadaného čísla, například ABS[-5] je 5

SQRT: druhá odmocnina ze zadaného čísla, například SQRT[4] jsou 2.

SIN: sinus zadaného čísla v radiánech, například SIN[0] je 0

COS: cosinus zadaného čísla v radiánech, například COS[3,14159] je -1

4.8.4 Rozměrová slova

X,Y,I,J,R: Mají význam souřadnice (délky), a to vždy v mm. Jejich použití viz. lineární, kruhová interpolace a interpolace po spirále a šroubovici.

k: Stoupavost poloměru při interpolaci po spirále. Udává, o kolik se zvětší poloměr za jeden oběh spirály. Vzorec pro její výpočet je $k = R/\varphi$, kde R je koncový poloměr spirály a φ je celkový úhel, o který se spirála otočí. Převod hodnot konstanty na velikost změny poloměru v mm se provádí podle vzorce: $\Delta R = k \cdot 2\pi$

H: Udává celkový počet stupňů otočení šroubovice

4.8.5 Parametry Q

Práce s parametry umožňuje ukládat číselnou hodnotu do proměnných a následně s nimi provádět matematické operace. S parametry lze provádět všechny definované matematické operace a především je lze používat jako makra. Použití na způsob maker znamená, že všude tam kde se vyskytne jméno definovaného parametru, vloží se jeho hodnota.

Každý parametr je identifikován svým unikátním číslem. Před prvním použitím musí být parametr definován se syntaxí Q[identifikační číslo]=[vkládaná hodnota].

Jednou definované parametry existují a jsou platné do skončení uživatelského programu.

Příklad:

Definice parametru:

Q1=5

Příklad matematické operace s parametry:

Q1=COS[0] Q2=SIN[1] Q3=Q1+Q2

Použití parametru jako makra:

N10 Q1=2 Q2=4 Q3=6 Q4=200

N15 G0 XQ1 YQ2 ZQ3 FQ4 je tedy stejné jako N15 G0 X2 Y4 Z6 F200

5 Programové řešení

Tato kapitola obsahuje popis vlastního návrhu a následného programového řešení jednotlivých bodů zadání.

V kapitole 5.1 je naznačen popis vytvořeného grafického uživatelského prostředí, které je základem pro automatický režim řízení.

V kapitole 5.2 je popsán způsob uchování a reprezentace dat v paměti, který jsem použil pro uživatelem zadaný G-kód. Následuje výpočet koncových bodů jednotlivých segmentů, rozepsaný do podkapitol pro každý typ interpolace zvlášť.

Následující kapitola 5.3 se věnuje způsobu zobrazení naprogramovaného průběhu na obrazovku.

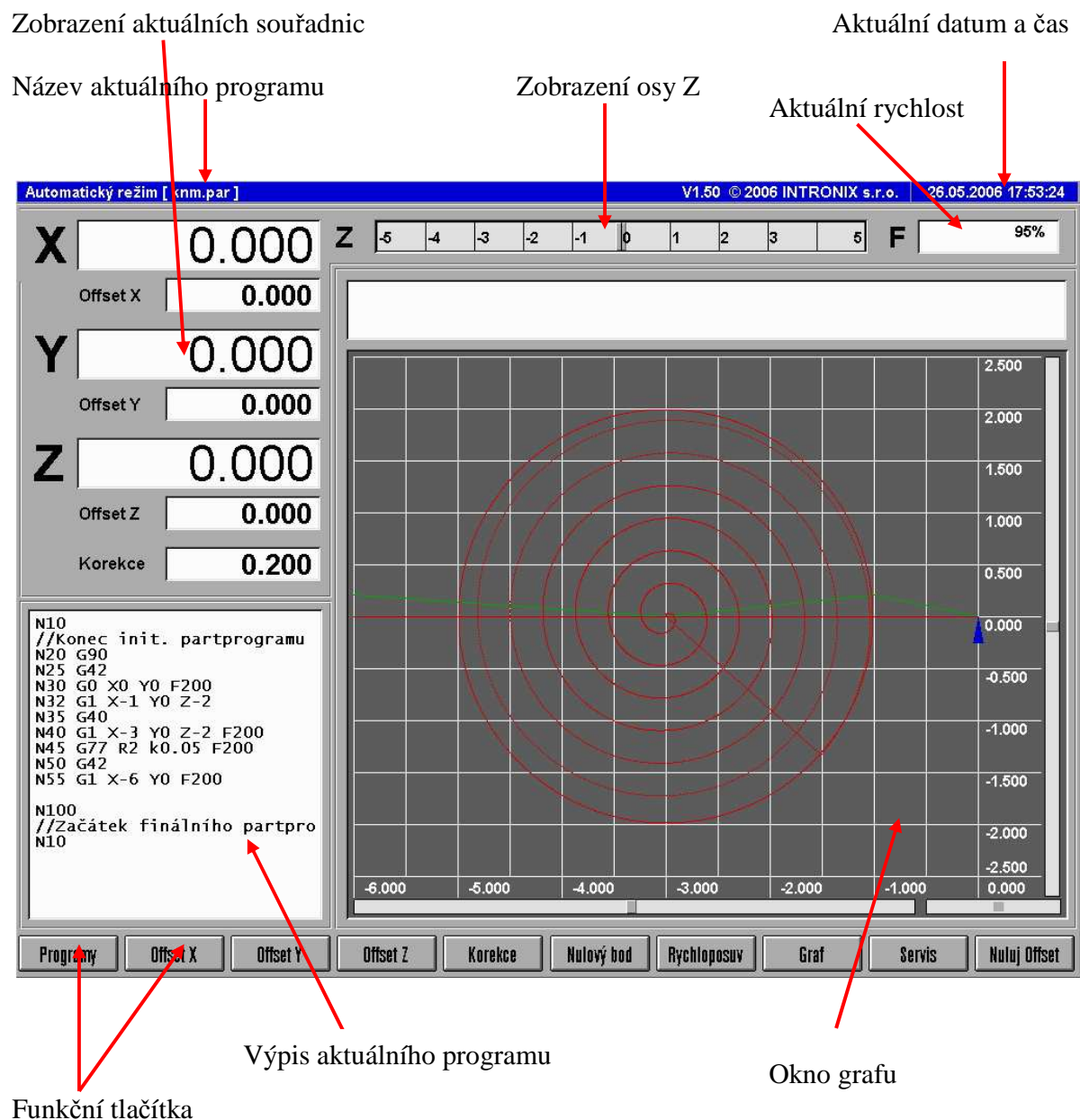
V poslední části, kapitole 5.4, je uveden princip a nejdůležitější výpočty pro naprogramovaný interpolátor zadaného průběhu obrábění.

5.1 Grafické uživatelské prostředí

Nejdůležitějším aspektem vytvořeného grafického uživatelského prostředí je přehlednost a rychlá odezva údajů na změny procesních hodnot.

5.1.1 Obrazovka automatického režimu

V tomto režimu je stroj kompletně řízen podle programu zadaného uživatelem. Vzhled obrazovky v automatickém režimu je na obrázku:



Obr. 13: Obrazovka automatického režimu

5.1.2 Komponenty obrazovky automatického řízení

V tomto oddílu uvádím seznam a popisují účel a vlastnosti jednotlivých prvků obrazovky automatického režimu stroje.

- *Funkční tlačítka ve spodní části obrazovky* – popisují význam tlačítek umístěných pod displejem.
- *Okno aktuálního vykonávaného programu* – je umístěno v levém spodním rohu obrazovky a je v něm zobrazen výpis aktuálního vykonávaného programu. Po nahrání programu tvořeného G-kódy je v okně zobrazen inicializační program, aktuální program a konečný program, které jsou mezi sebou odděleny komentáři. Při vykonávání programu je aktuální řádek zvýrazněn fialově.
- *Aktuální souřadnice* – jsou umístěny v levé horní části obrazovky a zobrazují souřadnice X, Y, Z aktuálního bodu stroje vzhledem k nulovému bodu stroje. Nulový bod stroje je bod zvolený uživatelem, ke kterému se vztahují všechny souřadnice stroje. Pod souřadnicemi jsou menším písmem vypisovány hodnoty aktuálního posuvu souřadnic za nápisy Offset X, Offset Y, Offset Z a poloměrové korekce. Hodnoty všech zobrazovaných souřadnic jsou v mm.
- *Zobrazení osy Z* – Zobrazení osy Z je v horní části obrazovky realizováno pomocí podlouhlého grafického pole, kde se pohybem kurzoru doprava značí kladný směr v ose Z a pohybem doleva pohyb záporný. Při překročení základních mezí, které jsou nastaveny na 5mm, zvětší pole svůj rozsah na dvojnásobek. Naopak při návratu z větších hodnot souřadnic osy Z ať už v kladném či záporném smyslu, se při najetí na hodnotu rovnající se polovině mezí, rozsah zmenší na polovinu, a to opět až do základního rozsahu 5mm.
- *Okno grafu* – zabírá pravou část obrazovky a je v něm zobrazen graf aktuálního programu. Na okrajích grafu jsou zobrazeny souřadnice odpovídající aktuálnímu posunutí/měřítku grafu. Zobrazované hodnoty měřítek jsou v mm. Křivky v grafu mají následující barvu:
 - Červená – přímky (oblouky) přímkové (kruhové) interpolace bez poloměrové korekce.
 - Zelená – přímky (oblouky) přímkové (kruhové) interpolace se zařazenou poloměrovou korekcí.

- Hnědá – přímký při pohybu rychloposuvem.
 - Modrý obdélník – udává hranici pracovní oblasti stroje – po najetí za tuto hranici již není možno pokračovat v původním směru, je možno se pouze vrátit.
 - Modrá šipka – zobrazuje aktuální polohu kotouče při běhu programu.
- *Okna aktuálních rychlostí a otáček* – jsou umístěna v pravém horním rohu obrazovky nad oknem grafu a mají následující význam:
F – rychlost posuvu v mm/min

5.2 Reprezentace G-kódů v programu

V této kapitole se zabývám převodem G-kódů ze souboru do reprezentace vhodné pro následné použití pro řízení frézky a s výpočty s tím spojenými.

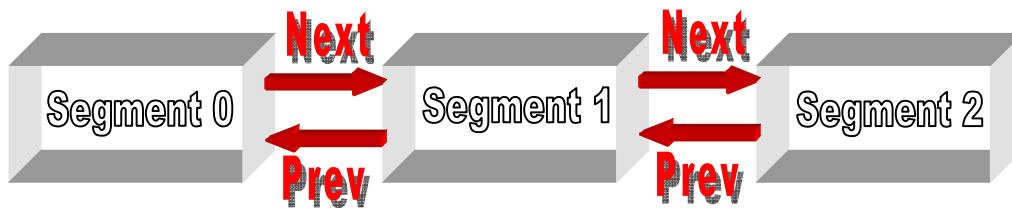
5.2.1 Uchování dat v programu

Rozbor použitého způsobu uložení dat z G-kódů v paměti.

5.2.1.1 Převod do spojového seznamu

Každý jednotlivý G-kód dekodovaný ze souboru s uživatelským programem je interpretován jako jeden segment spojového seznamu. Segmenty jsou za sebe vkládány v pořadí, v jakém jsou zapsány v uživatelském programu. Každý segment referencuje svého předchůdce i následovníka. Tímto způsobem se lze z libovolného segmentu, respektive místa ve spojovém seznamu, dostat k libovolnému předcházejícímu nebo následujícímu segmentu, přičemž je nutno znát pouze směr, kterým se ve spojovém seznamu pohybovat.

Každý segment tvoří ucelený blok dat, který dodává dostatek informací o tom, jakým způsobem daný segment používat. Jsou to tedy i informace o typu pohybu nástroje, cílovém bodu, nastavených rychlostech a použití poloměrové korekce. Vlastnosti, které přímo nevyplývají z významu zadaného G-kódu, se musí dopočítat, a to se nejčastěji děje při založení samotného segmentu.



Obr. 14: Spojový seznam

5.2.1.2 Typy segmentů spojového seznamu

Každý segment v sobě nese informaci o typu G-kódu, který byl do něj vložen. Jsou to následující:

- TYPE_NO_MOVE – segment, který nerealizuje žádný pohyb, je to například segment zapnutí, vypnutí poloměrové korekce nebo segmenty nastavení absolutních či relativních souřadnic
- TYPE_FAST – segment, který realizuje pohyb rychloposuvem
- TYPE_LINE – segment, který realizuje pohyb lineární interpolací
- TYPE_ARCCW respektive TYPE_ARCCCW – segment, který realizuje pohyb kruhovou interpolací ve směru hodinových ručiček, respektive proti směru hodinových ručiček.
- TYPE_CIRCCW respektive TYPE_CIRCCCW – segment, který realizuje pohyb šroubovicovou interpolací (závit) ve směru hodinových ručiček, respektive proti směru hodinových ručiček.
- TYPE_SPIRALCW respektive TYPE_SPIRALCCW – segment realizující pohyb po spirále ve směru hodinových ručiček, respektive proti směru hodinových ručiček.

5.2.1.3 Ostatní parametry segmentů

Ostatní parametry segmentů důležité pro jejich použití:

- **X2,Y2,Z2** Koncové souřadnice segmentů pro pohyb nástroje bez zařazené poloměrové korekce
- **Xeq,Yeq,Zeq** Koncové souřadnice segmentů pro pohyb nástroje se zařazenou poloměrovou korekcí
- **I,J** Souřadnice osy X (I) a osy Y (J) středů kruhových interpolací (kruhová, šroubovicová, spirálová).
- **R** respektive **Req** Velikost poloměrů kruhových interpolací bez zařazené poloměrové korekce, respektive se zařazenou poloměrovou korekcí
- **Eq** Velikost poloměrové korekce
- **F** Rychlost posuvu v programovém režimu
- **S** Rychlost otáčení nástroje v ot./min.

5.2.2 Výpočet koncových souřadnic segmentů

Výpočet koncových souřadnic segmentů je jednou z nejdůležitějších částí interpretace G-kódů pomocí spojového seznamu. Díky tomuto vyjádření známe oba body potřebné pro příslušnou interpolaci. Je to počáteční bod interpolace, který získáme jako koncový bod předchozího segmentu, a koncový bod interpolace, který je k aktuálnímu segmentu vypočítán a je uložen jako vlastnost **X2, Y2, Z2**. Způsob počítání koncového bodu závisí také na zvoleném odečítání souřadnic – absolutním či relativním, a také na možnosti zařazení offsetu ke každé ose zvlášť.

Princip výpočtu koncových bodů segmentu ukáží na segmentu lineární interpolace, protože zde lze nejpřehledněji vysvětlit vzniklé rozdíly a způsoby výpočtu, které jsou v ostatních segmentech obdobné.

5.2.2.1 Lineární interpolace

Při zvolené relativní souřadné soustavě:

Zadané souřadnice se přičtou ke koncovým souřadnicím předchozího segmentu (Prev):

$$X2 = Xi + Prev \rightarrow X2;$$

$$Y2 = Yi + Prev \rightarrow Y2;$$

$$Z2 = Zi + Prev \rightarrow Z2;$$

Jestliže je indikován zapnutý offset, zkontroluje se, jestli byl offset zařazen právě před aktuálním segmentem. Pokud ano, offset se k vypočítaným souřadnicím přičte, pokud ne, offset se nepřičte a neprovede se nic.

Jestliže je indikován vypnutý offset, zkontroluje se, jestli byl offset vypnut právě před aktuálním segmentem. Pokud ano, offset se od vypočítaných souřadnic odečte, pokud ne, neprovede se nic.

Při zvolení absolutní souřadné soustavy:

Souřadnice aktuálního segmentu jsou rovny zadaným souřadnicím v programu (Xi,Yi,Zi):

$$X2 = Xi;$$

$$Y2 = Yi;$$

$$Z2 = Zi;$$

Jestliže je indikován zapnutý offset, přičte se offset k vypočítaným souřadnicím, pokud je offset vypnutý, neprovede se nic.

5.2.2.2 Kruhová interpolace

Jelikož při zadání kruhové interpolace pomocí souřadnic středu I a J i při zadání pomocí poloměru kružnice se v syntaxi G-kódu objevují souřadnice požadovaných X,Y souřadnic, je výpočet koncových souřadnic X,Y stejný jako u lineární interpolace. Malá změna nastává u osy Z. Protože je kruhová interpolace realizována pouze v rovině X,Y, je souřadnice osy Z rovna koncové souřadnici předchozího segmentu: $Z2 = Prev \rightarrow Z2$.

Při zadání kruhové interpolace pomocí souřadnic I a J, která vypadá například následovně: **N20 G2 X0 Y3 I0 J1** musíme pro pozdější účely dopočítat poloměr. To je provedeno jednoduchou analytickou rovnicí:

$$R = \sqrt{(I - X2)^2 + (J - Y2)^2}$$

5.2.2.3 Interpolace po šroubovici

Jelikož je šroubovice složena z potřebného počtu segmentů pro kruhovou interpolaci, odpovídající celkovému úhlu otočení šroubovice, výpočet jejích koncových bodů je obdobný jako u kruhové interpolace.

Jiným způsobem se však určuje souřadnice osy Z, jejíž pokles/vzestup se udává opět relativně či absolutně. Pokud je celková délka šroubovice rozdělena na několik úseků, které jsou realizovány kruhovou interpolací, rozdělí se také změna souřadnice osy Z v odpovídajícím poměru.

V našem případě je šroubovice dělena na úseky po 180°, které jsou realizované kruhovou interpolací.

5.2.2.4 Interpolace po spirále

Pro realizaci spirálové interpolace je využita Archimedova spirála. Zadáním G-kódu je dán pouze konečný poloměr spirály a stoupavost poloměru spirály za úhlovou jednotku.

Jak již vyplývá z definice jednotky stoupavosti poloměru spirály, můžeme ze zadaných hodnot vypočítat celkový úhel otočení spirály podle vzorce:

$$\varphi = \frac{R}{k} ; \text{ kde } k \text{ je zmíněná stoupavost poloměru spirály}$$

Dále již jednoduše dopočteme koncové body z parametrických rovnic pro spirálu a přičteme koncový bod předchozího segmentu, který je pro spirálu bodem jejího středu.

Parametrické rovnice pro otáčení spirály po směru hodinových ručiček:

$$X_2 = k \cdot \varphi \cdot \sin(\varphi) + Pr_{ev} \rightarrow X_2$$

$$Y_2 = k \cdot \varphi \cdot \cos(\varphi) + Pr_{ev} \rightarrow Y_2$$

Parametrické rovnice pro otáčení spirály proti směru hodinových ručiček:

$$X_2 = k \cdot \varphi \cdot \cos(\varphi) + Pr_{ev} \rightarrow X_2$$

$$Y_2 = k \cdot \varphi \cdot \sin(\varphi) + Pr_{ev} \rightarrow Y_2$$

5.3 Vizualizace procesních dat

Tato kapitola se zabývá zobrazením naprogramovaných průběhů do grafu.

5.3.1 Graf průběhu interpolace

Požadované průběhy jsou vykresleny do grafu na základě vlastností dekodovaných z uživatelského programu, dopočítaných koncových bodů a ostatních vlastností segmentu.

S vhodným krokem pro spolehlivé a nepřerušované zobrazení dosazujeme do parametrických rovnic jednotlivých průběhů. U kruhových interpolací je tento krok například volen $1/R$, kde R je údaj v pixelech.

Dále je nutné přepočítat programové vzdálenosti v délkových jednotkách na vzdálenosti vyjádřené v pixelech. Poměr vzdálenosti v délkových jednotkách a v pixelech je závislý na rozměrech obrazovky, velikosti okna pro graf a především na zvoleném zvětšení okna s grafem (Zoom).

Následuje seznam použitých parametrických rovnic.

5.3.1.1 Lineární interpolace

Lineární interpolace je zobrazena pomocí knihovní funkce.

5.3.1.2 Kruhová interpolace a interpolace po šroubovici

Jak již bylo řečeno, interpolace po šroubovici je sled několika kruhových interpolací, proto pro jejich zobrazení budou platit stejné parametrické rovnice.

Jsou to následující:

$$\begin{aligned}x(\varphi) &= I + R \cdot \cos(\varphi) \\y(\varphi) &= J + R \cdot \sin(\varphi)\end{aligned}; \text{ kde } I, J \text{ jsou odpovídající souřadnice středu}$$

5.3.1.3 Spirálová interpolace

Spirálová interpolace se skládá ze tří částí. První je graf samotné spirály, následuje zakončovací kruh a nakonec přímka zpět do středu spirály. Uvádím pouze parametrické rovnice spirály, zbylé dva objekty jsou popsány výše.

Spirála po směru hodinových ručiček:

$$x(\varphi) = k \cdot \varphi \cdot \sin(\varphi)$$

$$y(\varphi) = k \cdot \varphi \cdot \cos(\varphi)$$

Spirála proti směru hodinových ručiček:

$$x(\varphi) = k \cdot \varphi \cdot \cos(\varphi)$$

$$y(\varphi) = k \cdot \varphi \cdot \sin(\varphi)$$

5.4 Interpolátor

Interpolátor je část programu, která vypočítává pohyb po naprogramované křivce. Výstupem interpolátoru je aktuální poloha v prostoru, která je použita jak pro zobrazení na obrazovku, tak pro vstup řídicích obvodů pohonů frézky. Výpočet polohy v prostoru je spouštěn časovačem tisíckrát do vteřiny, to znamená s periodou 1ms.

Interpolátor bere v úvahu programovou rychlost F , se kterou se má nástroj po křivce pohybovat.

5.4.1 Princip interpolátoru

Výpočty pro interpolátor se skládají ze dvou částí:

- Hodnoty vypočtené před začátkem interpolace
- Hodnoty počítané při každé iteraci interpolátoru

Naší snahou je nejkritičtější a nejdelsí výpočty provést již před začátkem interpolace, aby byla při vlastním průběhu interpolace zajištěna perioda výpočtů poloh v prostoru 1ms. Díky tomuto způsobu lze realizovat velmi rychlé řízení pohonů frézky s výkonovou rezervou pro plynulé zobrazování grafických informací.

5.4.1.1 Hodnoty vypočtené před začátkem interpolace

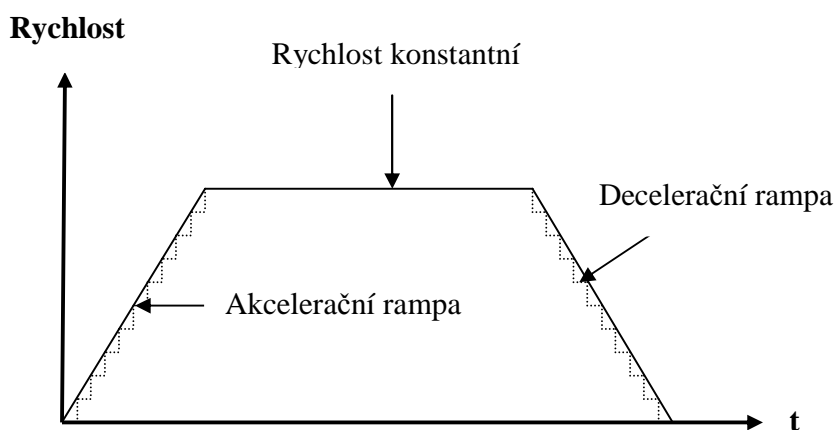
Mezi hodnoty vypočtené před začátkem interpolace patří především **délka křivky**. Rychlost pohybu po křivce je vzdálenost po křivce za jednotku času. Jestliže se rychlost interpolace přepočte na vzdálenost, o kterou se má aktuální bod po křivce posunout, interpolátor se velmi zjednoduší.

Dalšími hodnotami vypočtenými předem jsou úhly, které svírají jednotlivé osy, případně roviny a jejich goniometrické funkce, a to z důvodu již zmíněné časové optimalizace.

5.4.1.2 Rampa zrychlení, rampa zpomalení

Interpolátor má programově zadanou rychlost, kterou se má obrábění realizovat. Na tuto rychlost se ale přechází postupně, a to se strmostí danou v konstantách stroje. Při zrychlování o konstantní hodnotu rychlosti za časový úsek stoupá rychlost po rampě ne plynule, ale po diskrétních skocích.

Stejná pravidla ve smyslu zpomalování platí i pro brždění, které se také neděje skokem, ale se strmostí zadanou v konstantách stroje.



Obr. 15: Průběh rychlosti v rámci jednoho segmentu

5.4.1.3 Brzdná dráha

V každé iteraci interpolátoru se z rychlosti obrábění v příštím kroku a strmosti decelerační rampy vypočítá počet kroků potřebných k zabrždění a následně i brzdná dráha pro uvedení pohonů do klidu.

$$\text{Decelerace: } a_{decel} = \frac{v_{current} + \Delta v}{t} \quad \text{platí pro případ spojitého času } t$$

$$\text{Decelerace: } a_{decel} = \frac{v_{current} + \Delta v}{n} \quad \text{platí pro případ diskrétního času a } n \text{ je tedy počet kroků}$$

k zabrždění z rychlosti v příštím kroku $v_{current} + \Delta v$ se zpomalovací rampou a_{decel} .

Z počtu kroků nutných k zabrždění vypočteme brzdnou dráhu: $BrakeDist = n \cdot (v_{current} + \Delta v)$

a srovnáme ji se zbývající délkou křivky. Jestliže zjistíme, že by se mělo začít brzdit již v příštím kroku, začne se brzdit ihned, tím se sice zmenší sklon decelerační rampy, ale zajistí se tak zabrždění přesně do požadovaném bodu.

5.4.2 Výpočty pro interpolátor

V této kapitole uvádím vzorce pro výpočet délky křivky u jednotlivých interpolací. U spirálové interpolace uvádím obecný způsob výpočtu délky křivky, který lze použít i na mnohem složitější tvary zadané parametrickými rovnicemi.

Ostatní výpočty pro získání aktuální polohy jsou především založeny na postupném dosazování ujeté vzdálenosti do vhodně upravených parametrických rovnic.

5.4.2.1 Lineární interpolace

Délka přímky se vypočte analyticky podle vzorce

$$length = \sqrt{(End.X - Act.X)^2 + (End.Y - Act.Y)^2 + (End.Z - Act.Z)^2}$$

5.4.2.2 Kruhová interpolace

Délka kruhového oblouku (obvod kruhového oblouku):

$$length = R \cdot \varphi_{celk}$$

5.4.2.3 Šroubovicová interpolace

Šroubovicová interpolace je velmi podobná kruhové interpolaci, jen se zde musí brát v úvahu posun po ose Z.

Výpočet délky křivky: $length = \text{SQRT}(\text{ObvodKruhovéČásti}^2 + \text{PosunPoOseZ}^2)$

5.4.2.4 Spirálová interpolace

V této podkapitole je uvedeno odvození pro výpočet délky spirálovité křivky: Parametrické rovnice pro spirálu ve směru hodinových ručiček a proti směru hodinových ručiček sice nejsou identické, odvozený vzorec pro délku křivky ale platí pro oba případy, nezávisle na tom, který typ spirály jsme použili pro výpočet.

Parametrické rovnice spirály:

$$x(\varphi) = k \cdot \varphi \cdot \cos(\varphi)$$

$$y(\varphi) = k \cdot \varphi \cdot \sin(\varphi)$$

Obecný vzorec pro výpočet délky křivky:

$$length = \int_{\varphi_1}^{\varphi_2} \sqrt{\left(\frac{dx(\varphi)}{d\varphi}\right)^2 + \left(\frac{dy(\varphi)}{d\varphi}\right)^2} d\varphi$$

Pro náš případ tedy platí:

$$\frac{dx(\varphi)}{d\varphi} = \frac{d(k \cdot \varphi \cdot \cos(\varphi))}{d\varphi} = k \cdot \cos(\varphi) - k \cdot \varphi \cdot \sin(\varphi)$$

$$\frac{dy(\varphi)}{d\varphi} = \frac{d(k \cdot \varphi \cdot \sin(\varphi))}{d\varphi} = k \cdot \sin(\varphi) + k \cdot \varphi \cdot \cos(\varphi)$$

Dosadíme do obecného vzorce:

$$length = \int \sqrt{(k \cdot \cos(\varphi) - k \cdot \varphi \cdot \sin(\varphi))^2 + (k \cdot \sin(\varphi) + k \cdot \varphi \cdot \cos(\varphi))^2} d\varphi$$

Po integraci dostáváme konečný vzorec:

$$length = |k| \cdot \frac{\ln\left(\left|\sqrt{\varphi^2 + 1} + \varphi\right|\right) + \varphi \cdot \sqrt{\varphi^2 + 1}}{2}$$

Kde k je stoupavost poloměru spirály za úhlovou jednotku a φ je celkový úhel otočení spirály.

Pro použití v interpolátoru musíme do délky křivky také zahrnout délku zakončovacího kruhu, jehož poloměr má stejnou velikost jako koncový poloměr spirály.

Celková délka je tedy:

$$length = |k| \cdot \frac{\ln\left(\left|\sqrt{\varphi^2 + 1} + \varphi\right|\right) + \varphi \cdot \sqrt{\varphi^2 + 1}}{2} + 2 \cdot \pi \cdot R$$

6 Závěr

Výsledkem této bakalářské práce je funkční systém pro výpočty a vizualizaci procesních dat pro řízení CNC frézky fungující pod real-time operačním systémem ETS PharLap.

Program bude v blízké době rozšířen o samotné řízení krokových motorů frézky a prozatím bude využíván pro interní účely firmy INTRONIX spol. s r.o.

V rámci mé bakalářské práce jsem se seznámil, poznal výhody a široké možnosti real-time operačního systému PharLap. Práce byla pro mě mimo jiné přínosem v tom, že jsem si rozšířil obzory ve strojírenském oboru řízení CNC strojů, o kterém jsem měl před začátkem práce jen mlhavé představy.

Možnosti rozšíření mé práce jsou velmi široké. Prvním krokem je samozřejmě výše uvedené napojení programu na řídicí obvody frézky a dořešení problémů z toho plynoucích a momentálně neodhadnutelných. V rámci dalšího vylepšování lze postupovat od přidání mnoha dalších funkcí a frézovaných tvarů, které G-kódy znají, až po napojení řídicího software na programy typu CAD, konvertování tvarů kreslených pomocí těchto programů na G-kódy a jejich následné frézování. Tento výčet není samozřejmě úplný a možností je daleko více, ale nejvíce záleží na požadavcích zákazníka.

Vzhledem k ochotě mého vedoucího Ing. Pavla Burgeta a široké podpoře, které se mi dostalo ve firmě INTRONIX spol. s r.o. pro mě byla tato práce opravdu přínosem, a pokud k tomu budou vhodné podmínky, chtěl bych se jejímu rozšíření věnovat i v diplomové práci.

7 Literatura

- [1] Heidenhain, Bedienungs-Handbuch TNC 151B, TNC 155B
Poslední revize 01/1987 [cit. 15.5.2006]
- [2] Oficiální stránky výrobce operačního systému ETS PharLap [online]
www.ardence.com
- [3] Dodavatel opeačního systému ETS PharLap [online]
www.datapartner.cz
- [4] Rozhraní Win32 API v průmyslové automatizaci [online]
Poslední revize 04/2002 [cit. 15.5.2006]
<http://www.odbornecasopisy.cz/automa/2002/au040234.htm>
- [5] Systémy reálného času [online]
Poslední revize 03/2004 [cit. 15.5.2006]
<http://www.odbornecasopisy.cz/automa/2004/au030481.htm>
- [6] Enhanced Machine Controller project [online]
<http://www.linuxcnc.org/>
- [7] Navrhování systému s výpočetní technikou [online]
[cit. 2.6.2006]
<http://dce.felk.cvut.cz/nst/>
- [8] The Math Works [online]
[cit. 22.5.2006]
<http://www.mathworks.com>
- [9] Wolfram Mathworld Archimedean Spiral [online]
Poslední revize 05/2004 [cit. 28.4. 2006]
<http://mathworld.wolfram.com/ArchimedeanSpiral.html>
- [10] Herout, Pavel; Učebnice jazyka C, 1. díl (III. upravené vydání): Vydavatelství Kopp 2003 269 s. ISBN 80-85828-21-9
- [11] Herout, Pavel; Učebnice jazyka C, 2.díl: Vydavatelství Kopp 2002 236 s. ISBN 80-85828-50-2
- [12] Obráběcí stroje a jejich číslicové řízení [online]
Poslední revize 12/2005
<http://www.automatizace.cz/article.php?a=977>

Seznam příloh

Obsah přiloženého CD:

- elektronická podoba této práce ve formátu .pdf
- zdrojové kódy aplikace:
 - adresář **sources**:
 - graphtransf.cpp (dekódování uživatelského programu)
 - graphobject.cpp (převod uživatelského programu na segmenty a výpočty s tím spojené)
 - gui_automat.cpp (grafické uživatelské prostředí automatického režimu)
 - gui_graph.cpp (zobrazení obráběné křivky)
 - plcthread.cpp (interpolátor)