

**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

---

**Fakulta strojní**

**Ústav mechaniky, biomechaniky a mechatroniky**

**Trapezoidální interpolátor pro CNC frézku**

**Trapezoidal interpolator for CNC milling machine**

Bakalářská práce

Studijní program: Teoretický základ strojního inženýrství

Studijní obor: bez oboru

Vedoucí práce: Ing. Martin Nečas MSc., Ph.D.

**Pavel Houfek**

---

**Praha 2016**

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Houfek** Jméno: **Pavel** Osobní číslo: **420514**  
Fakulta/ústav: **Fakulta strojní**  
Zadávající katedra/ústav: **Ústav mechaniky, biomechaniky a mechatroniky**  
Studijní program: **Teoretický základ strojního inženýrství**  
Studijní obor: **bez oboru**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Trapezoidální interpolátor pro CNC frézku**

Název bakalářské práce anglicky:

**Trapezoidal interpolator for CNC mill**

Pokyny pro vypracování:

1. Seznamte se s problematikou řízení CNC strojů a používanými interpolátory 2. Vytvořte simulační model interpolátoru pro 3-osou frézku pro případ lineární interpolace skrze množinu předem definovaných poloh a rychlostí. 3. Vytvořený simulační model experimentálně ověřte na platformě Arduino 4. Kriticky zhodnoťte dosažené výsledky Rozsah grafických prací: cca 20 stran Rozsah práce: cca 50 stran

Seznam doporučené literatury:

Marek, J. a kol.: Konstrukce CNC obráběcích strojů. MM publishing, Praha 2010 W. S. Seames: Computer Numerical Control: Concepts and Programming. Cengage Learning, 2001 C. W. de Silva, Mechatronic Systems: Devices, Design, Control, Operation and Monitoring, CRC Press, 2007

Jméno a pracoviště vedoucí(ho) bakalářské práce:

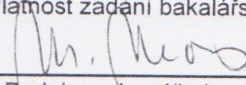
**Ing. Martin Nečas MSc., Ph.D.**

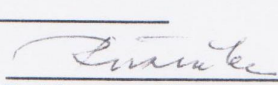
Jméno a pracoviště konzultanta(ky) bakalářské práce:


Datum zadání bakalářské práce: **21.04.2016**

Termín odevzdání bakalářské práce: **12-08-2016**

Platnost zadání bakalářské práce: \_\_\_\_\_

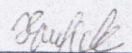
  
Podpis vedoucí(ho) práce

  
Podpis vedoucí(ho) ústavu/katedry

  
Podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

25.4.2016  
Datum převzetí zadání

  
Podpis studenta(ky)

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Trapezoidální interpolátor pro CNC frézku“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce Ing. Martina Nečase MSc., Ph.D. a s použitím odborné literatury uvedené v seznamu literatury.

V Praze dne ..... 2.6.2016 .....

Podpis..... *Houfek* .....

## **PODĚKOVÁNÍ**

Chci poděkovat vedoucímu mé bakalářské práce Ing. Martinu Nečasovi MSc., Ph.D. za jeho věcné připomínky, rady a nápady jak postupovat při tvorbě výpočtového programu jak z hlediska funkčního, tak z hlediska poznávání prostředí programu Matlab Simulink.

## **ABSTRAKT**

Cílem práce je navrhnout interpolační algoritmus v programu Matlab Simulink pro 3-osou CNC frézku ovládanou krokovými motory. Stručný úvod do problematiky je proveden historickým vývojem a popisem vývojových generací CNC strojů. Je probrána teorie CNC řízení a interpolace drah, kde je uveden nezbytný matematický základ. Pro orientaci v prostoru jsou uvedeny používané souřadné systémy a pro zápis hodnot struktura G-kódu. Na závěr teorie jsou uvedeny krokové motory jako pohony jednotlivých os 3-osé frézky. Následuje popis a vysvětlení funkce navrhovaného algoritmu. Jsou uvedeny použité rovnice a rozhodovací podmínky podle jednotlivých programových celků. V závěru je zhodnocena funkčnost vytvořeného algoritmu a jsou předloženy návrhy na budoucí možná vylepšení.

## **KLÍČOVÁ SLOVA**

interpolátor, rychlostní profily, lichoběžníkový profil rychlosti, trajektorový plánovač, teorie CNC řízení

## **ABSTRACT**

The aim is to propose interpolation algorithm in Matlab Simulink for 3-axis CNC milling machine controlled by stepper motors. Historical development and generation of CNC machines is performed brief introduction. The theory CNC control and interpolation paths is analysed and the necessary mathematical background is introduced. Used coordinate systems are introduced for orientation in a space, and for writing the values structure of G-code. At the conclusion of the theory part are analysed stepper motors as actuators of individual axes for 3-axis milling machine. Following topic is a description and explanation of the proposed algorithm. The used equations and determination conditions in individual program units are presented. In conclusion, functionality of the created algorithm is evaluated and some designs for possible future improvements are proposed.

## **KEY WORDS**

interpolator, velocity profiles, trapezoidal velocity profile, trajectory planning, CNC control theory

# OBSAH

<b>ÚVOD</b> .....	<b>7</b>
<b>1 PROBLEMATIKA CNC STROJŮ</b> .....	<b>8</b>
1.1 Historický vývoj .....	8
1.2 Vývojové generace .....	9
<b>2 TEORIE CNC ŘÍZENÍ</b> .....	<b>12</b>
2.1 Generování dráhy .....	12
2.2 Rychlostní profily .....	15
2.3 Souřadné systémy .....	22
2.4 Struktura G-kódu .....	25
2.5 Krokové motory .....	28
<b>3 INTERPOLÁTOR</b> .....	<b>35</b>
3.1 Interpolační profil .....	36
3.2 Rozdělení přirozené souřadnice polohy .....	40
3.3 Generátor pulzů .....	41
3.4 Aktuální poloha v prostoru .....	44
3.5 Souřadnice .....	46
3.6 Trajektorový plánovač .....	47
3.7 Zkušební úloha .....	52
<b>4 ZÁVĚR</b> .....	<b>54</b>
<b>5 SEZNAM POUŽITÝCH SYMBOLŮ</b> .....	<b>56</b>
<b>6 SEZNAM OBRÁZKŮ</b> .....	<b>59</b>
<b>7 POUŽITÁ LITERATURA</b> .....	<b>61</b>
<b>8 SEZNAM PŘÍLOH</b> .....	<b>64</b>
<b>9 PŘÍLOHY</b> .....	<b>65</b>

## ÚVOD

V současné době probíhá prudký rozvoj numericky řízených strojů. Vedou k tomu důvody jak ekonomické v podobně zvyšování produktivity výroby a zkrácení doby, po kterou je výrobek opracováván, tak důvody technické. CNC stroje umožňují vyrábět výrobky ve velkých sériích s vysokou přesností obrábění rozměrů, tvarů i dosažení vyšší jakosti povrchu součástí.

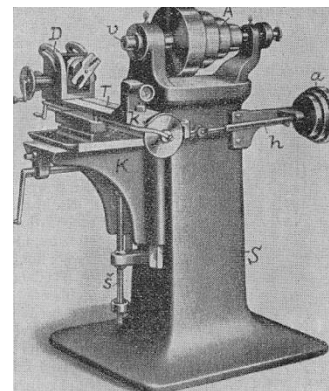
Trendem je zvyšování přesnosti, dosahující k hodnotě 0,001 mm, a zvyšování rychlosti obrábění CNC strojů. Řezné rychlosti se pohybují ve stovkách m/min a rychlosti posuvu v jednotkách m/s. Z těchto důvodů je nutné řídit pohyby jednotlivých os, tak aby nevznikaly nežádoucí setrvačné účinky a kmitání rámu stroje. Nárůst kinematických veličin musí být plynulý, což zajišťují řídicí systémy, kterých jsou implementovány interpolační algoritmy s matematickým modelem stroje. Dochází k propočítávání průběhu dráhy, na jejímž základě jsou řízeny pohony os tak, aby byly v součinnosti při složitějších trajektoriích.

Cílem práce je zmapovat matematický základ interpolátoru CNC stroje s ohledem na stanovený modelový příklad 3-osé frézky ovládané krokovými motory, zjistit jaké zvyklosti a struktury se používají pro vstupní hodnoty do interpolátoru. Na základě získaných znalostí provést návrh interpolačního algoritmu s lichoběžníkovým profilem rychlosti v programu Matlab Simulink, jehož výstupem jsou signály pro krokové motory.

# 1 PROBLEMATIKA CNC STROJŮ

## 1.1 Historický vývoj

Historie obráběcích strojů sahá až do středověku, kde měli řemeslníci manuálně poháněné soustruhy na obrábění dřeva. Avšak stroje pro obrábění kovů tak, jak je známe dnes, vznikly až v průběhu průmyslové revoluce v 18. a 19. století. První typem stroje s konstrukcí shodnou s dnešními stroji byl soustruh Henryho Maudslaye. (ZEITHAMMER, 2000) Od toho okamžiku došlo k rychlému zvýšení přesnosti výroby, protože součásti pro nové stroje byly vyráběny na již funkčních strojích a tudíž byly přesněji vyrobené. (ROSSMANN, 2008) Vznikaly nové typy strojů, které nejdříve vycházely z ručního zpracování, např. hoblovky a obrážky, ale i zcela nové koncepce, zejména frézka.



Obr. 1 Konvenční univerzální frézka z roku 1862 (Tumlik, 2013)

Kovoobráběcí stroje v této době používaly jeden centrální pohon, kterým mohl být parní stroj nebo, v případě polohy u tekoucí vody, kolo vodního mlýnu. Dílnou vedla hlavní hřídel nesoucí výkon pohonu a jednotlivé obráběcí stroje se pomocí řemenic napojovaly. Takto mohly různé typy strojů odebírat potřebný výkon pro obrábění. (ZEITHAMMER, 2000) Další milníkem v historii obráběcí strojů bylo nahrazení centrálního pohonu individuálním pro každý stroj zvlášť. Tento přechod umožnil vynález elektromotoru. Jako první ho do obráběcího stroje zahrnul Werner von Siemens na konci 19. století. (EncyklopedieCoJeCo, 2005) Dále vývoj pokračuje zejména zvyšováním produktivity přesnosti obrábění. (ZEITHAMMER, 2000)

První pokusy o automatizaci obráběcího procesu najdeme opět v 19. století. Hlavní silou pokroku bylo vojenství, kde bylo nutno vyrábět velké počty součástí zbraní za nejnižší možný čas v odpovídající přesnosti. Pro tento účel byly vynalezeny různé kopírovací stroje nebo stroje s řízením posuvů pomocí vaček, dorazů a šablon. Z dnešního pohledu jde o tzv. tvrdou automatizaci, kde jsou veškeré pohyby určeny mechanickou součástkou. (ROSSMANN, 2008) To vede k poměrně vysokým nákladům na stroje, které se vyplatí až při hromadné výrobě.

Velikým pokrokem v oblasti automatizace obrábění bylo vynalezení číslicově řízených strojů. Tento pokrok byl uskutečněn až po 2. světové válce, zejména v 50. letech 20. století. (Ing. POLZER, 2009) První číslicově řízení stroje, tzv. NC stroje (numerical



control), byly řízeny pomocí děrovaných štítků. Tento princip byl již v té době znám ze strojů textilních. (ZEITHAMMER, 2000) Na stroje klasické konstrukce byl umístěn řídicí systém zajišťující pohyb jednotlivých os. V roce 1950 byly použity první elektricky řízené hydromotory, později i řízené elektromotory namontované na konvenčních obráběcích strojích. (ROSSMANN, 2008) V 60. letech se objevilo první obráběcí centrum a koncem tohoto desetiletí začaly být aplikovány integrované obvody do řídicí elektroniky, které zaznamenaly rozmach v 70. letech. (ROSSMANN, 2008)

Postupně se během 80. a 90. let objevovaly nové koncepce numericky řízených obráběcích strojů. Obráběcí centra byla vyráběna jako kombinace soustruhů a frézek, kdy již nebylo možno jednoznačně říci, o který stroj se jedná, proto vznikl nový pojem, tzv. obráběcí centrum. Stroje nově obsahovaly zásobníky nástrojů, jejichž výměna byla možná v rámci programu, bez jeho přerušování. Zabudovávaly se senzory pro sledování pohonů a jejich zpětnou vazbu, pro přesné odměřování polohy nástroje, či senzory teplotní. (SOŠ Bíteš, 2001)

V současné době vede vývoj obráběcí center k neustálému zvyšování přesnosti obrábění, které dosahuje až 0,001 mm u vysoce tuhých strojů s kompenzací teplotních dilatací. Dalším odvětvím je snaha a zvyšování rezní rychlosti a celkové zkrácení mezioperačního času při obrábění. Toho



Obr. 2 Vertikální obráběcí centrum MCV 1220FA (Ing. HALA, 2010)

lze dosáhnout např. snížením počtu upnutí při výrobě součástí. Obráběcí centra obsahují velkokapacitní zásobníky nástrojů, řeší třískové hospodářství i manipulaci s polotovarem a obrobkem. (SOŠ Bíteš, 2001) Trendem v oblasti CNC techniky je integrování CAD/CAM technologií přímo do řídicích počítačů strojů a jejich provázání s externími počítači v řídicích místnostech. (ROSSMANN, 2008)

## 1.2 Vývojové generace

Jak postupoval vývoj CNC obráběcích strojů, lze od sebe odlišit stroje různých vývojových generací. Liší se počtem řízených os, způsoby obrábění a stupni automatizace.

### **1.2.1 První vývojová generace**

První číslicově řízené stroje vycházely z konstrukcí konvenčních strojů doplněných o řídicí systém. (Ing. POLZER, 2009) Pohyb a nastavování souřadnic nástroje bylo možné jen v pravouhlých souřadnicích. Tato koncepce byla vhodná pouze pro stroje, kde je pro pracovní pohyb potřeba jen jedna řízená osa. (SOŠ Bíteš, 2001) Pro požadavky dnešní strojírenské výroby jsou tyto stroje nevyhovující s ohledem na technologické možnosti stroje a efektivitu procesu. (SOŠ Bíteš, 2001)

### **1.2.2 Druhá vývojová generace**

Tato generace strojů je již vybavena řízenými servopohony a umožňuje souvislé řízení ve dvou osách současně. (SOŠ Bíteš, 2001) Stroje byly vybavovány revolverovými hlavami a zásobníky nástrojů pro mezioperační výměnu nástrojů. (SOŠ Bíteš, 2001) Opotřebované nástroje však bylo nutné v zásobníku měnit ručně. Řídicí software byl doplněn pamětí a rozšířen o další operační a pomocné funkce. (Ing. POLZER, 2009)

### **1.2.3 Třetí vývojová generace**

Stroje této generace již umožňují souvislé řízení ve třech osách současně. Zvyšuje se přesnost a produktivita výroby. (SOŠ Bíteš, 2001) Začíná se využívat senzorů pro sledování pohonů a mechanismů. V těchto strojích byla již zaintegrována automatická výměna obrobků a zvětšena kapacita zásobníků nástrojů. (Ing. POLZER, 2009)

### **1.2.4 Čtvrtá vývojová generace**

Charakteristickým prvkem této generace je souvislé řízení ve více než třech osách současně. (SOŠ Bíteš, 2001) Zavádějí se nové metody v konstrukci strojů, objevují se obráběcí centra. (SOŠ Bíteš, 2001) Zajištěna automatická výměna opotřebovaných nástrojů ze zásobníků. Stroje mohou samostatně pracovat v třísměnném provozu. (Ing. POLZER, 2009) Pro zpracování CAD/CAM programů je využíváno postprocesorů.

### **1.2.5 Pátá vývojová generace**

Tato generace strojů s sebou nese mechatronické prvky. Programově je zajištěna např. elektronická kompenzace chyb polohování, kompenzace tepelných deformací nebo měření obrobků mezi jednotlivými operacemi, tzv. improcesní měření. (Ing. POLZER, 2009) Zjišťování skutečných rozměrů probíhá pomocí měřících sond a v řídicím systému stroje jsou porovnány s hodnotami požadovanými výkresem. (Ing. POLZER, 2009) Pokud skutečné rozměry nejsou v požadovaných tolerancích, je obráběcí proces automaticky upraven

a optimalizovány řezné podmínky. Při mezioperační výměně nástrojů může být poloha břitu měřena laserem a rozdíl je uložen do korekční tabulky. (Ing. POLZER, 2009)

### **1.2.6 Šestá vývojová generace**

Zatím poslední generace, která se stále vyvíjí. Konstrukce těchto se zaměřuje na požadavky maximálního snížení časů výměn obrobků a nástrojů, zvýšení tuhosti rámu pro vysokorychlostní obrábění. (Ing. POLZER, 2009) Diagnostika obráběcího procesu je napojena na vzdálené pracoviště. Další zvýšení přesnosti obrábění až na 0,0001 mm s využitím mechatronických prvků. (Ing. POLZER, 2009)

## 2 TEORIE CNC ŘÍZENÍ

### 2.1 Generování dráhy

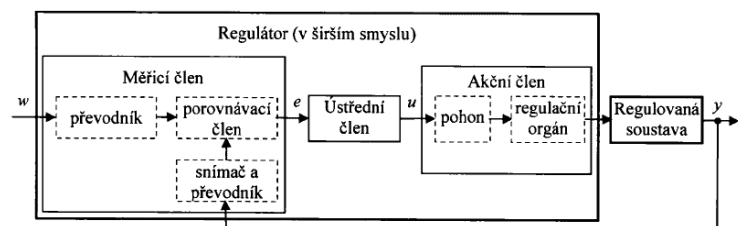
Programátor zadá stroji body, kterými bude procházet dráha nástroje a způsob projetí mezi těmito body. Stroj musí v každém okamžiku znát aktuální polohu programovaného bodu, takže musí v reálném čase dopočítávat souřadnice bodů, které leží mezi zadanými. Tyto informace jsou zpracovávány v interpolátoru, aritmetické výpočtové jednotce a porovnávány s údaji ze senzorů. (Prof. Dr. Ing. SCHMID, 2005)

Podle charakteru dráhy jsou rozlišovány tyto druhy interpolace:

- lineární interpolace
- kruhová interpolace
- interpolace splinem
- spirálová interpolace

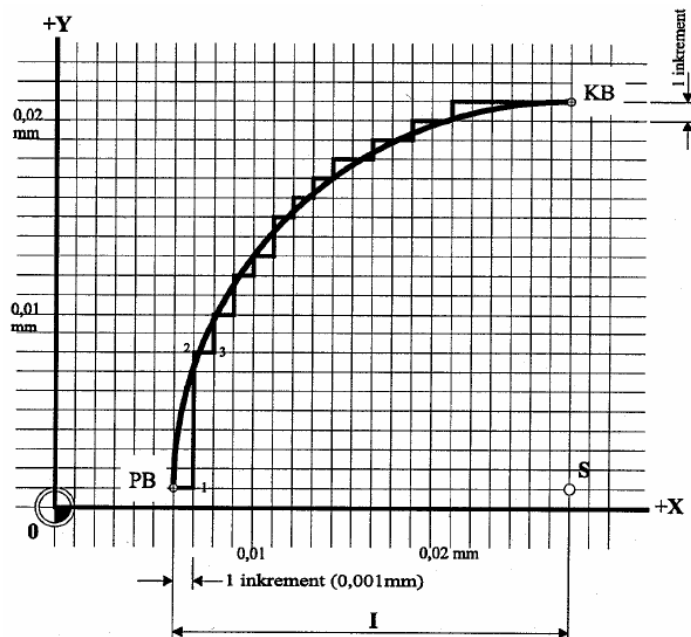
Každá z uvedených druhů interpolace musí mít v řídicím systému stroje zadaný matematický základ, který ze zadaných parametrů polohy a rychlosti interpolaci provede. Pro popis dráhy jsou používány rovnice z analytické geometrie v parametrickém tvaru. (Prof. Dr. Ing. SCHMID, 2005) Společným parametrem pro průběh dráhy je čas. Teoreticky vypočtená dráha a poloha je porovnávána se skutečnou polohou získanou pomocí odměřovacího zařízení. Použity mohou být absolutní snímače polohy nebo inkrementální snímače polohy, u kterých však musí být

provedeno vynulování souřadnic, najetí na referenční bod, při spuštění stroje. (ROSSMANN, 2008) Obě informace, o skutečné a teoretické poloze, se porovnávají a vzniká regulační



Obr. 3 Blokové schéma regulátoru (Prof. Ing. HOFREITER, 2014)

odchylka. Její transformace a zesílení je provedeno v PID regulátoru, který posílá signál na motor až do vyrovnání regulační odchylky a dosažení požadované polohy. (Prof. Ing. HOFREITER, 2014) Snímače polohy neměří spojitě, ale mají danou nejmenší měřitelnou změnu polohy, tzv. inkrement, od kterého se odvíjí i přesnost nastavení polohy a interpolace křivek suportem stroje. (ROSSMANN, 2008)



Obr. 4 Grafické znázornění omezené přesnosti strojů vlivem inkrementů (ROSSMANN, 2008)

### 2.1.1 Lineární interpolace

Umožňuje stroji spojit dva zadané body přímkovou trajektorií s definovanou rychlostí.

Přímku lze parametricky vyjádřit následující rovnicí:

$$\underline{X} = \underline{C} + p \cdot \underline{u}, \quad (1)$$

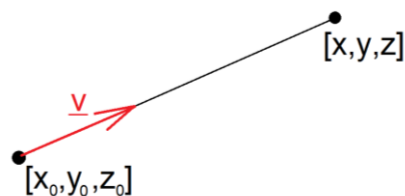
kde vektor  $\underline{A}$  určuje bod, kterým přímka prochází, vektor  $\underline{u}$  udává směrnici a  $p$  je parametr. Pro kinematické řešení přímky je možné tento vztah přepsat do podoby:

$$\underline{X} = \underline{X}_0 + t \cdot \underline{v} \quad (2)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + p \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}, \quad (3)$$

kde  $\underline{X}_0$  je počáteční bod dráhy a  $\underline{v}$  je rychlost posuvu nástroje.

Z lineární interpolace vycházejí i všechny ostatní způsoby interpolace, protože při snižování vzdáleností bodů k nule lze vykreslit jakoukoli křivku a daným rozlišením. Při snižování těchto vzdáleností získáme v každém bodě směrnici požadované křivky a tedy i náhradní úsečky.



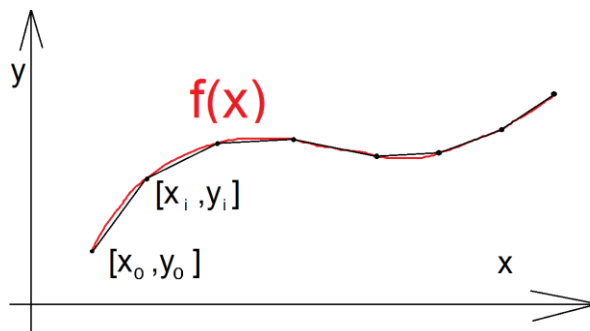
Obr. 5 Interpolace dvou bodů (NEUSTUPA, 2008)

$$k = \operatorname{tg} \psi = f' = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad (4)$$

Což nám dává předpis definující derivaci funkce. Přímku poté můžeme popsat v tzv. úsekovém (explicitním) tvaru přímky (Prof. Dr. Ing. SCHMID, 2005):

$$y = kx + q, \quad (5)$$

kde  $k$  je směrnice přímky a  $q$  je konstanta udávající polohu přímky.



Obr. 6 Interpolace obecné funkce pomocí úseček

### 2.1.2 Kruhová interpolace

Pro správné provedení kruhové interpolace je nutné stroji zadat nejen polohu koncového bodu, ale i střed oblouku a jeho poloměr. Kruhová interpolace může být provedena, na rozdíl od lineární, pouze v rovině. Pokud by se měnila i třetí souřadnice, jednalo by se již o spirálovou interpolaci. Kruhový oblouk lze opět popsat více způsoby. Pokud je kruhová interpolace založena na lineární se vzdáleností mezi body jdoucí k nule, je možné použít explicitní tvar rovnice kružnice. (Prof. Dr. Ing. SCHMID, 2005)

$$(x - m_S)^2 + (y - n_S)^2 = r^2 \quad (6)$$

$$y = n_S + \sqrt{r^2 - (x - m_S)^2}, \quad (7)$$

kde  $S[m_S, n_S]$  udávají polohy středu kružnice a  $r$  její poloměr. Při dosažení bodu přímkovou interpolací je touto rovnicí vygenerována nová dvojice souřadnic s požadovaným rastrem.

Kružnice může být popsána také parametricky, kde její průběh závisí pouze na čase. Tento popis je vhodnější z hlediska řízení rychlosti v průběhu projíždění dráhy. (NEUSTUPA, 2008)

$$\underline{X} = \underline{S} + r \begin{bmatrix} \cos(\omega t + \varphi) \\ \sin(\omega t + \varphi) \end{bmatrix} \quad (8)$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} m_S \\ n_S \end{bmatrix} + r \begin{bmatrix} \cos(\omega t + \varphi_0) \\ \sin(\omega t + \varphi_0) \end{bmatrix}, \quad (9)$$

kde  $\underline{S}$  je vektor souřadnic středu,  $r$  je poloměr kružnice a  $(\omega t + \varphi_0)$  je proměnný parametr je definovaným počátkem oblouku úhlem  $\varphi_0$  a danou úhlovou rychlostí  $\omega$ . Pomocí

kinematických vztahů lze snadno za úhlovou rychlost dosadit obvodovou, která odpovídá rychlosti posuvu.

$$\omega = \frac{v}{r} \quad (10)$$

## 2.2 Rychlostní profily

Pro plynulý pohyb z jednoho bodu do druhého je potřeba řídit rychlost tohoto pohybu. V přírodě nemůže žádná změna nastat nekonečně rychle a tedy ani rychlost nemůže být od počátku na maximální hodnotě. V takovém případě by teoreticky muselo být zrychlení nekonečně velké, řízený motor by se toho snažil dosáhnout a mohl by se poškodit. (LEWIN, 2007) Z tohoto důvodu se pro řízení používají tzv. řízené rychlostní profily. Nejjednodušším profilem je lichoběžníkový (Trapezoidal profile), který má lineární nárůst rychlosti. Dalším používaným profilem je po S-křivce (S-curve profile), který má nárůst rychlosti po plynulejší křivce. Mezi komplexnější profily patří polynomický, kde jsou používány polynomy vyšších řádů, goniometrický nebo pomocí Fourierovy transformace. (Motion Designs Inc., 2010)

### 2.2.1 Lichoběžníkový rychlostní profil

Lichoběžníkový rychlostní profil je tvořen 3 částmi při trajektorii z bodu do bodu. Nejdříve je lineární nárůst rychlosti a zrychlení je konstantní. Souřadnice polohy narůstá po parabole. V další části je rychlost konstantní, zrychlení nulové a souřadnice narůstá po přímce. Poslední část je stejná jako první s tím rozdílem, že dochází ke zpomalení. (LEWIN, 2007) Tato vlastnost vychází ze vztahů mezi těmito veličinami, kdy pohyb řídíme zrychlením. První integrace zrychlení je rychlost a druhá integrace je poloha. Integrační operace vyhlazují křivky, a tedy z 0 vznikne konstanta, z konstanty přímka a z přímky křivka 2. stupně. (LEWIN, 2007)

$$v = \int a dt = at + v_0 \quad (11)$$

$$s = \int v dt = \int at + v_0 dt = \frac{1}{2}at^2 + v_0t + s_0, \quad (12)$$

kde  $a$  je konstantní zrychlení,  $v$  je rychlost v čase  $t$  a  $s$  je poloha v čase  $t$ . Proměnné  $v_0$  a  $s_0$  jsou počáteční hodnoty daných veličin.

Uvedené rovnice platí pro spojitý průběh veličin tak, jak to platí v přírodě. Bohužel tento popis není vhodný pro řídicí systémy, které pracují v diskrétních časových úsecích. Integrál lze s úspěchem nahradit numerickou integrací funkce neurčitého integrálu. Obecný

vztah pro numerickou integraci lichoběžníkovou metodou je uveden v rovnicích (14) a (15). (Mgr. RŮŽIČKOVÁ)

Neurčitý integrál lze pro získání funkčních hodnot primitivní funkce nahradit určitým integrálem z počáteční hodnoty  $a$  do obecného místa  $x$ . Hodnota  $a$  představuje počáteční podmínky pro výpočet konstanty neurčitého integrálu.

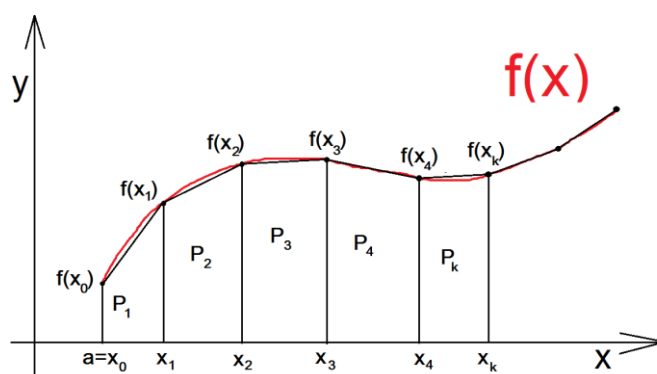
$$F(x) = \int f(x)dx = \int_a^x f(t)dt \quad (13)$$

Integrál je nahrazen sumou, kde pro každou další funkční hodnotu je nutné sečíst všechny předchozí a přidat další krok.

$$F(x_i) \approx \sum_{k=1}^i P_k \quad (14)$$

$$P_k = (x_k - x_{k-1}) \frac{f(x_k) + f(x_{k-1})}{2} = \Delta x_k \frac{f(x_k) + f(x_{k-1})}{2}, \quad (15)$$

kde  $F(x_i)$  je funkční hodnota primitivní funkce pro hodnotu  $x_i$  a  $P_k$  je plocha lichoběžníku pod křivkou mezi hodnotami  $x_{k-1}$  a  $x_k$  při použití aproximace pomocí lichoběžníkové metody.



Obr. 7; Schéma lichoběžníkové metody numerické integrace

Vztahy lze dále upravit dosazením zkoumaných veličin závisících na čase.

$$v_i = \sum_{k=1}^i P_k = \sum_{k=1}^i \Delta t_k \frac{a_k + a_{k-1}}{2} \quad (16)$$

$$s_i = \sum_{k=1}^i \Delta t_k \frac{v_k + v_{k-1}}{2} \quad (17)$$

Sumy v rovnicích definují funkční hodnoty jako součet ploch všech předcházejících lichoběžníků vzniklých mezi dvěma sousedními hodnotami při aproximaci. Výpočtové vztahy lze ještě následujícím způsobem zjednodušit s použitím jednoho sčítacího indexu.



$$v_i = v_{i-1} + \Delta t_i \frac{a_i + a_{i-1}}{2} \quad (18)$$

$$s_i = s_{i-1} + \Delta t_i \frac{v_i + v_{i-1}}{2} \quad (19)$$

Pro lichoběžníkový profil rychlosti je použita konstantní hodnota zrychlení, takže při zrychlování nebo zpomalování pohybu platí  $a_{i-1} = a_i = a$ , při použití konstantní velikosti časového kroku můžeme napsat  $\Delta t_i = \Delta t_{i-1} = \Delta t$

$$v_i = v_{i-1} + \Delta t \cdot a \quad (20)$$

$$\begin{aligned} s_i &= s_{i-1} + \Delta t \frac{v_i + v_{i-1}}{2} = s_{i-1} + \Delta t \frac{v_{i-1} + \Delta t \cdot a + v_{i-1} + \Delta t \cdot a}{2} = \\ &= s_{i-1} + \Delta t \frac{v_{i-1} + v_{i-1}}{2} + a \cdot \Delta t^2 \end{aligned} \quad (21)$$

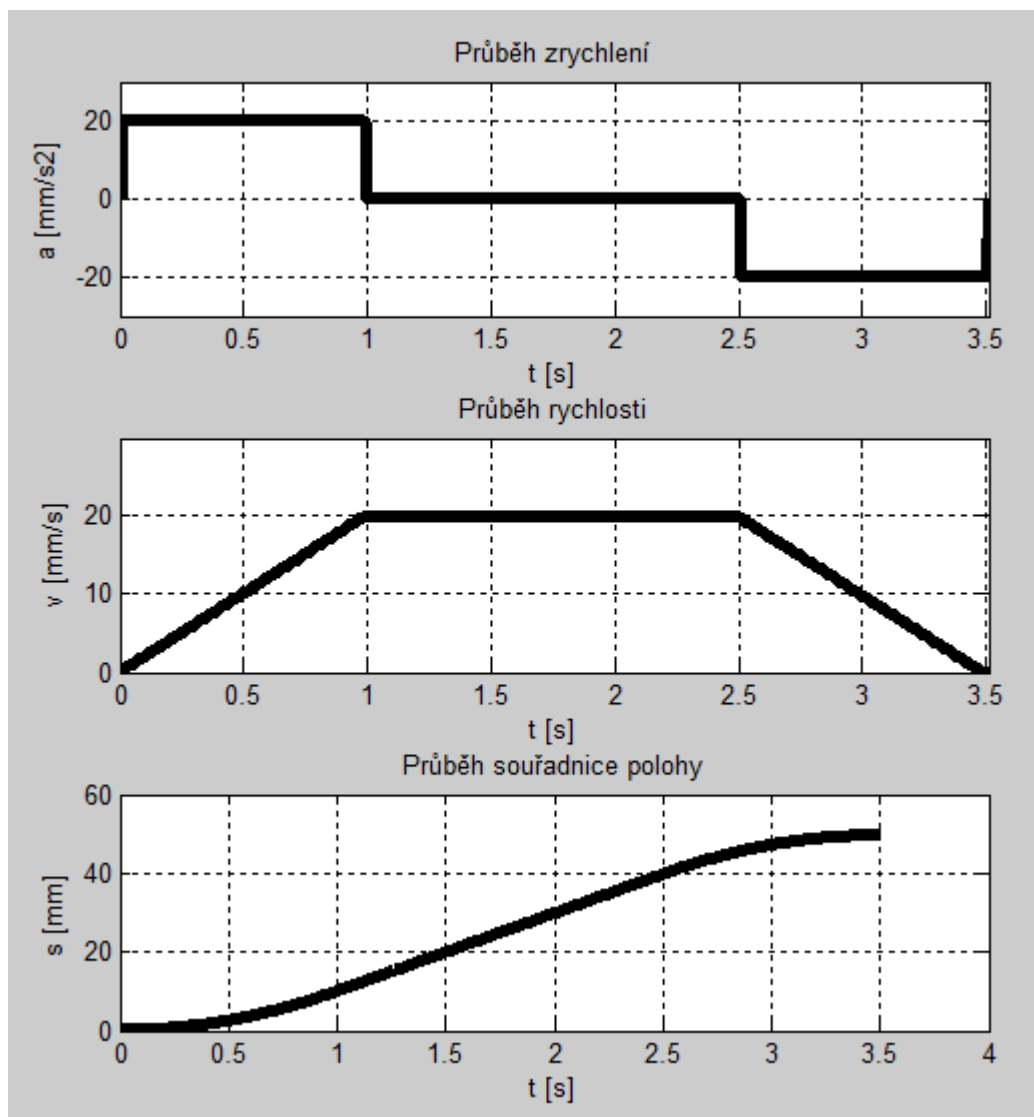
Tyto odvozené vztahy lze snadno zadat do řídicího systému pracujícího s diskrétními hodnotami. Při použití numerické integrace je nutné ověřit velikost chyby zvolené numerické metody. Obecná rovnice pro odhad chyby lichoběžníkové metody. (Mgr. RŮŽIČKOVÁ)

$$|E| \leq \frac{h^3}{12} m M_2, \quad (22)$$

kde  $h$  je velikost kroku,  $m$  je počet kroků a  $M_2$  je definována jako

$$M_2 = \max_{t \in \langle a, b \rangle} |f''(t)|, \quad (23)$$

kde  $f(t)$  je integrovaná funkce.



Obr. 8 Grafické zobrazení kinematických veličin při lichoběžníkovém profilu rychlosti

Pro lichoběžníkový rychlostní profil lze také pomocí základních kinematických vztahů snadno odvodit doplňující informace nutné pro správný výpočet profilu dráhy jako např. celkový čas pohybu a časy jednotlivých úseků za předpokladu známé vzdálenosti ze souřadnic  $s$ , dané maximální rychlosti  $v$ , počáteční rychlosti  $v_{poč}$ , koncové rychlosti  $v_{konc}$ , zrychlení  $a_1$  a zpomalení  $a_3$  stroje.

$$t = t_1 + t_2 + t_3 \quad (24)$$

Dílčí časy zrychlení a zpomalení lze vyjádřit takto:

$$t_1 = \frac{v - v_{poč}}{a_1} ; \quad t_3 = \frac{v - v_{konc}}{a_3} \quad (25)$$

Čas  $t_2$  se vyjádří z podmínky (26) pro dráhy jednotlivých úseků

$$s = s_1 + s_2 + s_3 = \int \int a_1 dt dt + \int v dt + \int \int a_3 dt dt \quad (26)$$

$$s = \frac{1}{2}a_1t_1^2 + v_{poč}t_1 + vt_2 + vt_3 - \frac{1}{2}a_3t_3^2 \quad (27)$$

$$s = \frac{1}{2}a_1 \frac{(v - v_{poč})^2}{a_1^2} + v_{poč} \frac{v - v_{poč}}{a_1} + vt_2 + v \frac{v - v_{konc}}{a_3} - \frac{1}{2}a_3 \frac{(v - v_{konc})^2}{a_3^2} \quad (28)$$

$$s = \frac{(v - v_{poč})^2}{2a_1} + v_{poč} \frac{v - v_{poč}}{a_1} + vt_2 + v \frac{v - v_{konc}}{a_3} - \frac{(v - v_{konc})^2}{2a_3} \quad (29)$$

$$t_2 = \frac{1}{v} \left[ s - \frac{(v - v_{poč})^2}{2a_1} - v_{poč} \frac{v - v_{poč}}{a_1} - v \frac{v - v_{konc}}{a_3} + \frac{(v - v_{konc})^2}{2a_3} \right] \quad (30)$$

Do rovnice pro celkový čas již dosadíme odvozené vztahy dílčích časů

$$t = t_1 + t_2 + t_3 \quad (31)$$

$$t = \frac{v - v_{poč}}{a_1} + \frac{1}{v} \left[ s - \frac{(v - v_{poč})^2}{2a_1} - v_{poč} \frac{v - v_{poč}}{a_1} - v \frac{v - v_{konc}}{a_3} + \frac{(v - v_{konc})^2}{2a_3} \right] + \frac{v - v_{konc}}{a_3} \quad (32)$$

$$t = \frac{v - v_{poč}}{a_1} + \frac{s}{v} - \frac{(v - v_{poč})^2}{2va_1} - v_{poč} \frac{v - v_{poč}}{va_1} - \frac{v - v_{konc}}{a_3} + \frac{(v - v_{konc})^2}{2va_3} + \frac{v - v_{konc}}{a_3} \quad (33)$$

Výsledný vztah pro výpočet celkové doby pohybu.

$$t = \frac{s}{v} - \frac{(v - v_{poč})^2}{2va_1} + \frac{v - v_{poč}}{a_1} \left( 1 - \frac{v_{poč}}{v} \right) + \frac{(v - v_{konc})^2}{2va_3} \quad (34)$$

Tyto odvozené složité rovnice lze zjednodušit, pokud jsou zavedeny určité předpoklady. Prvním předpokladem pro zjednodušení je pohyb z nulové rychlosti  $v_{poč} = 0$  do nulové rychlosti  $v_{konc} = 0$ .

$$t = \frac{s}{v} - \frac{v}{2a_1} + \frac{v}{a_1} + \frac{v}{2a_3} = \frac{s}{v} + \frac{v}{2a_1} + \frac{v}{2a_3} = \frac{s}{v} + \frac{v}{2} \left( \frac{1}{a_1} + \frac{1}{a_3} \right) \quad (35)$$

Pro další zjednodušení odvozeného vztahu lze zavést předpoklad stejné velikosti zrychlení i zpomalení  $a_1 = a_3 = a$

$$t = \frac{s}{v} + \frac{v}{a} \quad (36)$$

V mnoha případech se může stát, že dráha pohybu bude příliš krátká a posuv nestihne zrychlit na požadovanou rychlost. Přestává platit výše odvozený vztah pro celkový čas, protože se pohyb skládá pouze ze zrychleného a zpomaleného. Střední část s konstantní rychlostí chybí. Tento předpoklad lze využít při odvození vztahu pro celkový čas v tomto případě.

$$v_{dos} = v_{poč} + a_1t_1 \quad (37)$$

$$v_{dos} = v_{konc} + a_3t_3 \quad (38)$$

$$s = \frac{1}{2} a_1 t_1^2 + v_{poč} t_1 + v_{dos} t_3 - \frac{1}{2} a_3 t_3^2 \quad (39)$$

Tím vznikly tři rovnice o třech neznámých  $t_1$ ,  $t_3$  a  $v_{dos}$ , jejichž řešením jsou námí hledané vztahy.

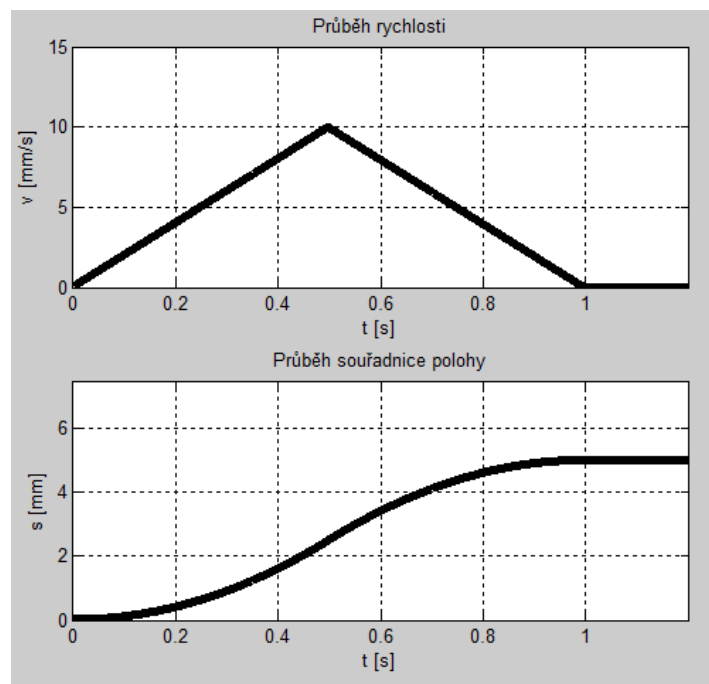
$$v_{dos} = \sqrt{\frac{s + \frac{v_{poč}^2}{2a_1} + \frac{v_{konc}^2}{2a_3}}{\frac{1}{2a_1} + \frac{1}{2a_3}}} \quad (40)$$

$$t_3 = \frac{v_{dos} - v_{konc}}{a_3} \quad (41)$$

$$t_1 = \frac{v_{dos} - v_{poč}}{a_1} \quad (42)$$

Rovnice pro celkový čas pohybu vznikne součtem těchto dvou vztahů

$$t = t_1 + t_3 = \frac{v_{dos} - v_{poč}}{a_1} + \frac{v_{dos} - v_{konc}}{a_3} \quad (43)$$



Obr. 9 Průběh rychlosti a polohy při krátké dráze

## 2.2.2 Rychlostní profil po S-křivce

S-křivkový rychlostní profil používá obecně křivky o jeden řád vyšší než lichoběžníkový profil a průběh je složen z více částí. Použití tohoto profilu vychází z teorie, že v přírodě nemůže nastat nekonečně rychlá změna a tedy existuje ani nárůst zrychlení nemůže být nekonečně rychlý. (NGUYEN, 2008) Tato změna zrychlení je definována veličinou nazývanou jerk, která je definovaná jako časová změna zrychlení.

$$J = \frac{da}{dt} \quad (44)$$

Průběh veličin je o jeden řád posunut. Nejvyšší derivací je jerk, který je konstantou. Zrychlení má přímkový nárůst a tvoří lichoběžníkový profil. Rychlost narůstá po parabole, obecně nazývané S-křivka a poloha narůstá po kubice. (Motion Designs Inc., 2010) Pohyb je plynulejší a nevznikají vibrace, které mohou nastat při použití lichoběžníkového profilu. Matematicky je však pohyb po S-křivce o něco složitější. Vychází se ze stejných rovnic jako u předchozího profilu. (LEWIN, 2007)

$$a = \int J dt = Jt + a_0 \quad (45)$$

$$v = \int a dt = \int Jt + a_0 dt = \frac{1}{2}Jt^2 + a_0t + v_0 \quad (46)$$

$$s = \int v dt = \int \frac{1}{2}Jt^2 + a_0t + v_0 dt = \frac{1}{6}Jt^3 + \frac{1}{2}a_0t^2 + v_0t + s_0 \quad (47)$$

Pro praktické použití v řídicích systémech se opět nahrazuje tato spojitá integrace numerickou. Stejným postupem jako v předchozím případě lze odvodit tytéž obecné rovnice pro kinematické veličiny.

$$a_i = \sum_{k=1}^i \Delta t_k \frac{J_k + J_{k-1}}{2} = a_{i-1} + \Delta t_i \frac{J_i + J_{i-1}}{2} \quad (48)$$

$$v_i = \sum_{k=1}^i \Delta t_k \frac{a_k + a_{k-1}}{2} = v_{i-1} + \Delta t_i \frac{a_i + a_{i-1}}{2} \quad (49)$$

$$s_i = \sum_{k=1}^i \Delta t_k \frac{v_k + v_{k-1}}{2} = s_{i-1} + \Delta t_i \frac{v_i + v_{i-1}}{2} \quad (50)$$

Pro zjednodušení vztahů pro výpočet lze zavést předpoklad konstantní velikosti časového kroku  $\Delta t_i = \Delta t_{i-1} = \Delta t$  a v jednotlivých intervalech lze také předpokládat konstantní hodnotu jerku  $J_{i-1} = J_i = J$ . Vztahy lze analogicky k lichoběžníkovému profilu převést do jednodušší podoby.

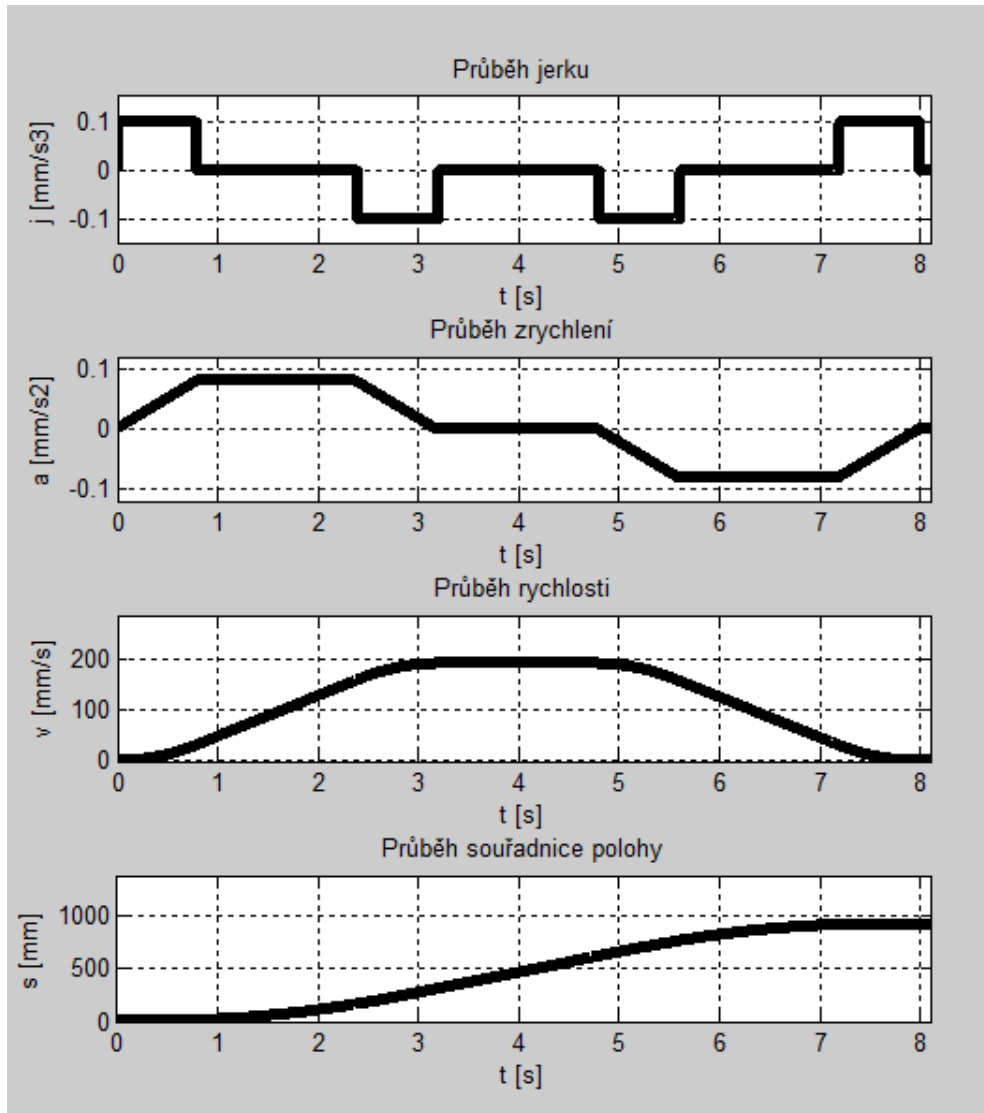
$$a_i = a_{i-1} + \Delta t \cdot J \quad (51)$$

$$\begin{aligned} v_i &= v_{i-1} + \Delta t \frac{a_i + a_{i-1}}{2} = v_{i-1} + \Delta t \frac{a_{i-1} + \Delta t \cdot J + a_{i-1} + \Delta t \cdot J}{2} = \\ &= v_{i-1} + \Delta t \frac{a_{i-1} + a_{i-2}}{2} + J \cdot \Delta t^2 \end{aligned} \quad (52)$$

$$s_i = s_{i-1} + \Delta t \frac{v_i + v_{i-1}}{2} =$$

$$\begin{aligned}
&= s_{i-1} + \Delta t \frac{v_{i-1} + \Delta t \frac{a_{i-1} + a_{i-2}}{2} + J \cdot \Delta t^2 + v_{i-2} + \Delta t \frac{a_{i-2} + a_{i-3}}{2} + J \cdot \Delta t^2}{2} = \\
&= s_{i-1} + \frac{1}{2} \Delta t (v_{i-1} + v_{i-2}) + \frac{1}{4} \Delta t^2 (a_{i-1} + 2a_{i-2} + a_{i-3}) + J \cdot \Delta t^3
\end{aligned} \tag{53}$$

Pro větší názornost je vhodnější vykreslit tyto průběhy graficky.



Obr. 10 Grafické vykreslení profilu rychlosti po S-křivce

### 2.3 Souřadné systémy

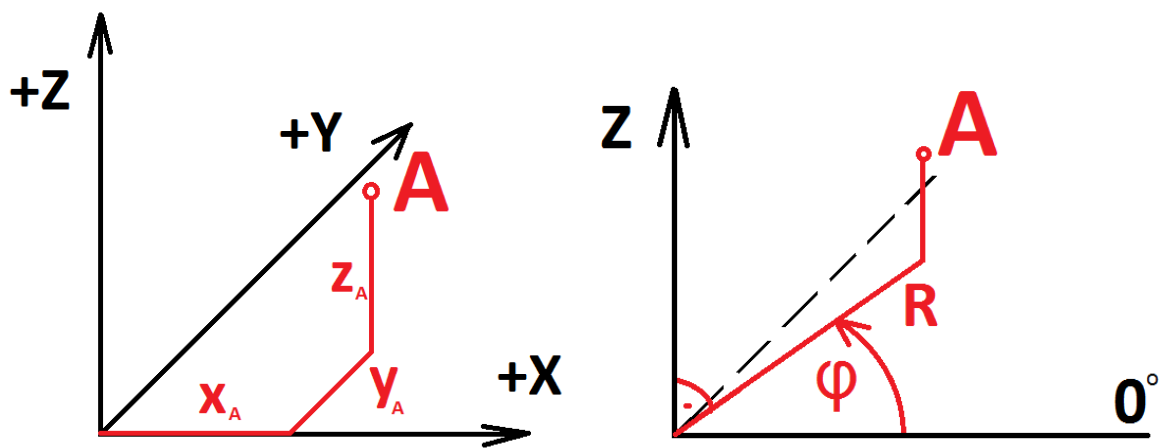
Aby bylo možné obrábět součásti podle definovaných souřadnic, musí mít řídicí systém stroje definován souřadný systém včetně počátku souřadnic. Obráběcí CNC stroje jsou schopny pracovat v různých souřadných systémech podle jejich mechanické konfigurace a možností pohybu.

Základní a nejvíce používaný je pravoúhlý kartézský systém souřadnic. Podle normy je vždy pravotočivý a poloha os je definovaná konstrukcí stroje. Kladný směr osy Z leží na

ose rotace vřetena a směřuje do vřetena. (SOŠ Bíteš, 2001) Směry os X a Y jsou orientovány rovnoběžně s vodicími plochami stroje a kladná orientace dodržuje pravidlo pravotočivého systému. (Prof. Dr. Ing. SCHMID, 2005) Poloha koncového bodu je v tomto souřadném systému definována pomocí odpovídajících souřadnic na jednotlivých osách od počátku. Souřadnice XYZ jsou používány na většině obráběcích strojů

Druhým často používaným systémem souřadnic je polární souřadný systém nebo jeho prostorová varianta cylindrický souřadný systém. Osa Z má stejnou definici jako v předchozím případě, ale v rovině je poloha bodu definovaná pomocí polohového vektoru. Parametry jsou délka vektoru a směr je dán orientovaným úhlem. (SOŠ Bíteš, 2001) Tento souřadný systém je využíván zejména, pokud má řízený stroj jednu rotační osu, jako tomu je např. u radiálních vrtaček. Výhodné je také použití při popisování poloh prvků na roztečných kružnicích, u obrobků s kruhovými oblouky nebo při úhlovém kótování prvků na výkrese. (SOŠ Bíteš, 2001)

Existuje mnoho dalších souřadných systémů pro popis polohy bodu v prostoru, ale z konstrukčních důvodů se u CNC strojů nepoužívají. Své uplatnění nacházejí především u víceosých robotů a průmyslových manipulátorů. (Prof. Dr. Ing. SCHMID, 2005)



Obr. 11 Používané souřadné systémy, vlevo kartézský souřadný systém, vpravo cylindrický souřadný systém

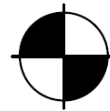
Poloha souřadného systému je definována polohou nulového bodu, kterých může být několik a každý definuje jiný souřadný systém. (Ing. KELLER, 2005) Globální souřadný systém stroje je definován výrobcem a vztahuje se ke konkrétnímu místu na konstrukci stroje, tzv. strojové souřadnice. U soustruhů to bývá v ose vřetene na čelní ploše upínacího dorazu. (Prof. Dr. Ing. SCHMID, 2005) Na frézkách se umísťuje do rohu stolu tak, aby byl operační prostor popisován kladnými hodnotami. Počátek globálního souřadného systému je určen nulovým bodem stroje a značí se M. Od něj se odvozují všechny následující souřadné systémy, které jsou určeny transformačními maticemi.



Obr. 12 Značka nulového bodu stroje M (Ing. KELLER, 2005)

$$\underline{r}_{O1} = \underline{T}_{12}\underline{r}_{O2} \quad (54)$$

S globálním souřadným systémem souvisí referenční bod R. Je to konkrétní bod v pracovním prostoru stroje, který slouží ke zjištění polohy programovaného bodu vůči nulovému bodu M, tzv. najetí do reference. (Ing. KELLER, 2005) Často je umísťován do krajních poloh os, kdy je poloha přesně zjištěna při sepnutí koncových snímačů. Nastavení referenčního bodu je nutností pro stroje s inkrementálním odměřováním polohy, protože při vypnutí napájení se vymaže informace o absolutní poloze a stroj umožňuje měřit polohu pouze od známého bodu. (Prof. Dr. Ing. SCHMID, 2005) Když stroj najede do referenčního bodu, tak zjistí přesnou polohu od nulového bodu a lze začít odměřovat. Poloha referenčního bodu vůči nulovému bodu je zadána v tabulce strojních konstant. (SOŠ Bíteš, 2001) Při použití absolutního odměřování polohy tento problém odpadá. (Ing. KELLER, 2005)



Obr. 13 Značka referenčního bodu R (Ing. KELLER, 2005)

Pro psaní programů je důležitý nulový bod obrobku W. Definuje si ho programátor při sestavování programu pro konkrétní součást tak, aby mohly být snadno zadány rozměrové parametry z výkresu. Poloha bodu W je odvozena od nulového bodu stroje M pomocí transformačních matic. Souřadnice jsou psány v lokálním souřadném systému obrobku, kde počátek je nulový bod obrobku. (ROSSMANN, 2008) Jsou nezávislé na poloze obrobku v pracovním prostoru stroje. Stačí pouze definovat bod W a všechny ostatní souřadnice jsou automaticky přepočítány vzhledem k bodu M.

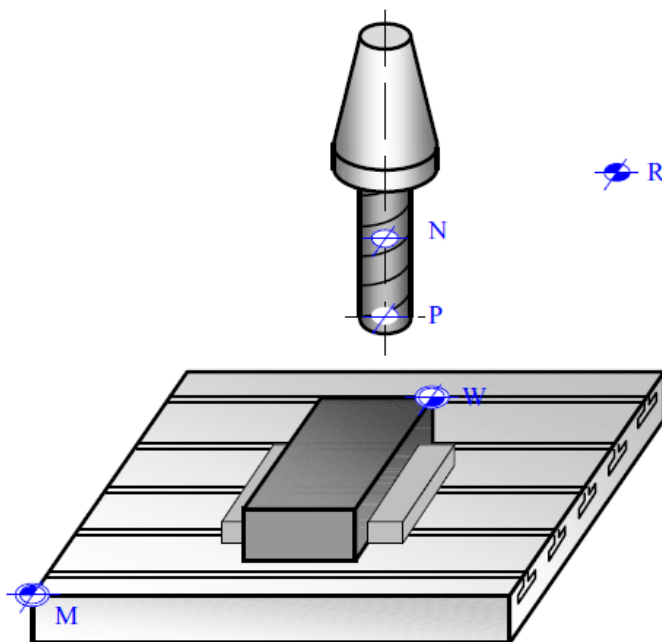


Obr. 14 Značka nulového bodu obrobku (Ing. KELLER, 2005)

V řídicích systémech CNC strojů se vyskytuje mnoho dalších vztažných bodů, které usnadňují používání stroje. Mezi významné body patří bod dorazu A, který je v místě protnutí rotační osy a plochy dorazu obrobku. (Prof. Dr. Ing. SCHMID, 2005) Definuje polohu



obrobku při upínání. Nulový bod nástroje P popisuje polohu teoretické špičky nástroje a zároveň programovaný bod. V případě rotačních nástrojů leží v ose rotace na čele nástroje. Každý nástroj má svoji polohu bodu P (Ing. KELLER, 2005) Pro definování rozměru nástrojů je používán nulový bod nástroje N. Leží v průsečíku osy nástroje a dorazované plochy ve vřetenu. Od tohoto bodu jsou odměřovány délky nástrojů a z nich vypočítávány korekce pro úpravu bodu P. (Prof. Dr. Ing. SCHMID, 2005)



Obr. 15 Vyznačení vztažných bodů na frézce (Ing. KELLER, 2005)

## 2.4 Struktura G-kódu

Pro programování trajektorií CNC obráběcích strojů byl vytvořen programovací jazyk zvaný G-kód nebo také ISO kód. Normou jsou dány základní příkazy, které jsou shodné pro všechny CNC tak, aby byla možná snadná přenositelnost informací a nebylo nutné se učit nové programovací jazyky. Jsou definovány programové bloky, které tvoří základní rozdělení funkcí. V programu je možné rozeznat blok N, který udává číslo řádku, blok funkcí G pro zajištění uvození funkce a blok pomocných funkcí M. Každý z těchto bloků obsahuje funkce povinné, dané normou a společné pro všechny stroje, doporučené a funkce volně k dispozici použitelné pro některý příkaz stroji. (Ing. ŠTAJNORCH, 2015)

### 2.4.1 Povinné G-funkce

Funkce bloku G jsou používány zejména pro řízení dráhy stroje. Zajišťují zadávání souřadnice jednotlivých bodů dráhy nástroje, definování rychlosti posuvu a tvaru trajektorie.

(Ing. ŠTAJNORCH, 2015) Mezi dvěma body lze obecně vytvořit nekonečně mnoho křivek, proto se musí zajistit, aby stroj vytvořil trajektorii z křivky definované. Pro zadávání souřadnic je nutné nejdříve definovat souřadný systém, ve kterém se bude obrábět. (ŠTULPA, 2006) Tuto funkci opět zajišťují funkce z bloku G. Jednotlivé příkazy zadávané stroji je pro názornost lepší vypsát do tabulky.

G00	Rychloposuv
G01	Lineární interpolace
G02	Kruhová interpolace ve směru hodinových ručiček
G03	Kruhová interpolace proti směru hodinových ručiček
G04	Časová prodleva
G90	Absolutní programování
G91	Inkrementální programování

Tab. 1 Povinné funkce bloku G

Funkce G00 a G01 jsou si principiálně velmi podobné. Definují prohyb po přímce mezi dvěma body. Hlavní rozdíl je v tom, že funkce G00 je pro rychloposuv, tedy dráha mezi body je provedena nejvyšší možnou rychlostí. Oproti tomu u funkce G01 rychlost posuvu zadá programátor.

N06	G00	X100	Y100	Z30	
N07	G01	X120	Y150	Z35	F200

Tab. 2 Příklad zadání funkce G00 a G01

V řádku G01 se objevuje, kromě souřadnic X, Y, Z, adresa F, do které je právě zadána rychlost posuvu v mm/min (Prof. Dr. Ing. SCHMID, 2005)

Kruhová interpolace zajišťovaná funkcemi G02 a G03 definuje pohyb po části kružnice. Standardně lze v ISO kódu kruhový oblouk zadat dvěma způsoby. Vždy známe počáteční a koncový bod oblouku a pro přesný tvar jedna informace chybí. Touto informací může být velikost rádiusu oblouku, který je zadáván do adresy R. Je však nutné přesně potřebnou velikost rádiusu spočítat jinak se sice funkce provede, ale výsledkem bude zmetkový výrobek. (ŠTULPA, 2006)

N10	G01	X200	Y200	Z125	-	F150
N11	G02	X300	Y300	-	R100	F150

Tab. 3 Zadání příkazu pro provedení oblouku pomocí rádiusu. Předchozí funkce G01 definuje počáteční bodu oblouku

Druhým způsobem je dodat stroji informaci o poloze středu oblouku. Jeho souřadnice I,J,K je zadávána vždy přírůstkově z počátečního bodu oblouku. Při tomto způsobu zadávání je nižší pravděpodobnost chyby než u zadání rádiusu. (ŠTULPA, 2006)

N10	G01	X200	Y200	Z125	-	-	-	F150
N11	G02	X300	Y300	-	I100	J0	-	F150

Tab. 4 Zadání příkazu pro provedení oblouku pomocí středu. Předchozí funkce G01 definuje počáteční bodu oblouku

Kruhové oblouky je možné provádět pouze v rovině. Pro prostorové provedení některé stroje poskytují šroubovou interpolaci, která se řadí mezi speciální funkce.

Funkce G90 definuje absolutní programování. Při zadávání této funkce se zapíše souřadnice nástroje v aktuální poloze od nulového bodu souřadnicového systému. Celý následující program se vztahuje k tomuto jednomu počátku souřadného systému.

N01	G90	X200	Y200	Z50
-----	-----	------	------	-----

Tab. 5 Tento zápis říká, že v okamžiku spuštění programu je programovaný bod na souřadnicích [200;200;50] od počátku souřadného systému

Funkce G91 zadává inkrementální nebo také přírůstkové programování. Princip tohoto způsobu spočívá v tom, že při pohybech nástroje se souřadný systém posouvá s ním. Vždy tedy programujeme přírůstek dráhy z aktuální polohy.

## 2.4.2 Povinné M-funkce

Dalším blokem jsou funkce typu M, používající se zejména pro pomocné příkazy stroji, aby byl zajištěn správný chod programu, technologické parametry a ovládání vřetene.

M00	Programový stop
M30	Konec hlavního programu
M03	Spuštění otáček vřetena ve směru hodinových ručiček
M04	Spuštění otáček vřetena proti směru hodinových ručiček
M05	Zastavení vřetena

Tab. 6 Povinné funkce bloku M (ŠTULPA, 2006)

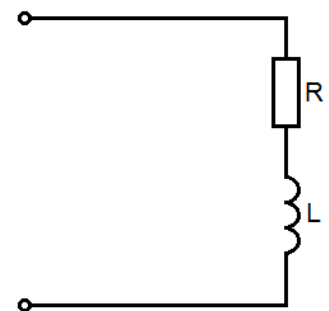
Pro funkce ovládající rotaci vřetena je směr otáčení popisován pro vektor úhlové rychlosti směřující vřetena stroje, nikoli ze strany obsluhy. (ŠTULPA, 2006)

## 2.5 Krokové motory

Na numericky řízených CNC strojích se pro polohování jednotlivých os používají různé druhy elektromotorů. Nejčastěji jsou používány stejnosměrné nebo střídavé servomotory se zabudovaným odměřováním natočení hřídele. Trojfázové servopohony jsou řízeny regulací proudu z hlediska momentu a změnou frekvence proudu z hlediska rychlosti rotace. Pro tento účel jsou využívány frekvenční měniče. (VOJÁČEK, 2014) Z kategorie stejnosměrných motorů jsou používány stejnosměrné komutátorové motory, jejichž rychlost se řídí velikostí napětí. V praxi se využívá PWM pulzů definujících průměrné napětí. Dalším typem stejnosměrného motoru je bezkartáčový DC motor s elektronickou komutací, který se vyznačuje jednoduchou konstrukcí, ale má však složitější řízení. Posledním často používaným typem je krokový motor. Jeho řízením se zabývá tato práce, proto bude popsán podrobněji.

Krokový motor je synchronní DC motor, u kterého je úhel natočení hřídele dán počtem impulsů přivedených na vinutí cívky. (Doc. Ing. BĚLOHOUBEK, 1998) Od klasických DC motorů se liší tím, že se netočí kontinuálně a pouze po jednotlivých krocích. Rychlost otáčení rotoru je dána frekvencí pulzů přivedených na vinutí. Maximální rozsah frekvencí bývá udáván v rozsahu 0 – 20 kHz, ale je třeba vzít v úvahu pokles krouticího momentu při vyšších rychlostech.

Stator motoru je tvořen cívkou, která má vlastní indukčnost a odpor. Při skokovém nárůstu proudu na hraně PWM pulzu vzniká na cívce motoru přechodový jev. Pokud se motor točí pomalu, proud stihne cívku nabít na plnou hodnotu a vše funguje tak, jak má. (SLINTÁK, 2013) Při zvýšení otáček se vinutí cívky nestací nabít a puls vymizí při menší hodnotě proudu, než je přiváděný. To způsobuje prudký pokles momentu při vyšších otáčkách. Pro výpočet přechodového jevu lze stator motoru modelovat jako sériově zapojený rezistor a cívku. (SLINTÁK, 2013) Pro tento obvod lze sestavit rovnici pomocí metody smyčkových proudů.



Obr. 16 Náhradní schéma modelující stator krokového motoru (SLINTÁK, 2013)

$$i \cdot R + L \frac{di}{dt} - U = 0 \quad (55)$$

$$\frac{L}{R} \frac{di}{dt} + i = \frac{U}{R} \quad (56)$$

Podíl indukčnosti a odporu definuje časovou konstantu  $\tau$  obvodu.

$$\tau \frac{di}{dt} + i = \frac{U}{R} \quad (57)$$

Vzniklá diferenciální rovnice prvního řádu se vyřeší pomocí charakteristické rovnice.

$$\tau\lambda + 1 = 0 \quad (58)$$

$$\lambda = -\frac{1}{\tau} \quad (59)$$

Dosadí se do obecného řešení rovnice

$$i(t) = C_1 e^{\lambda t} + i_p = C_1 e^{-\frac{1}{\tau}t} + i_p \quad (60)$$

Partikulární řešení diferenciální rovnice se provede odhadem pravé strany

$$i_p = C_2 \quad (61)$$

$$\frac{di_p}{dt} = 0 \quad (62)$$

$$0 + C_2 = \frac{U}{R} \quad (63)$$

$$i(t) = C_1 e^{-\frac{1}{\tau}t} + \frac{U}{R} \quad (64)$$

Z počátečních podmínek se určí konstanta  $C_1$

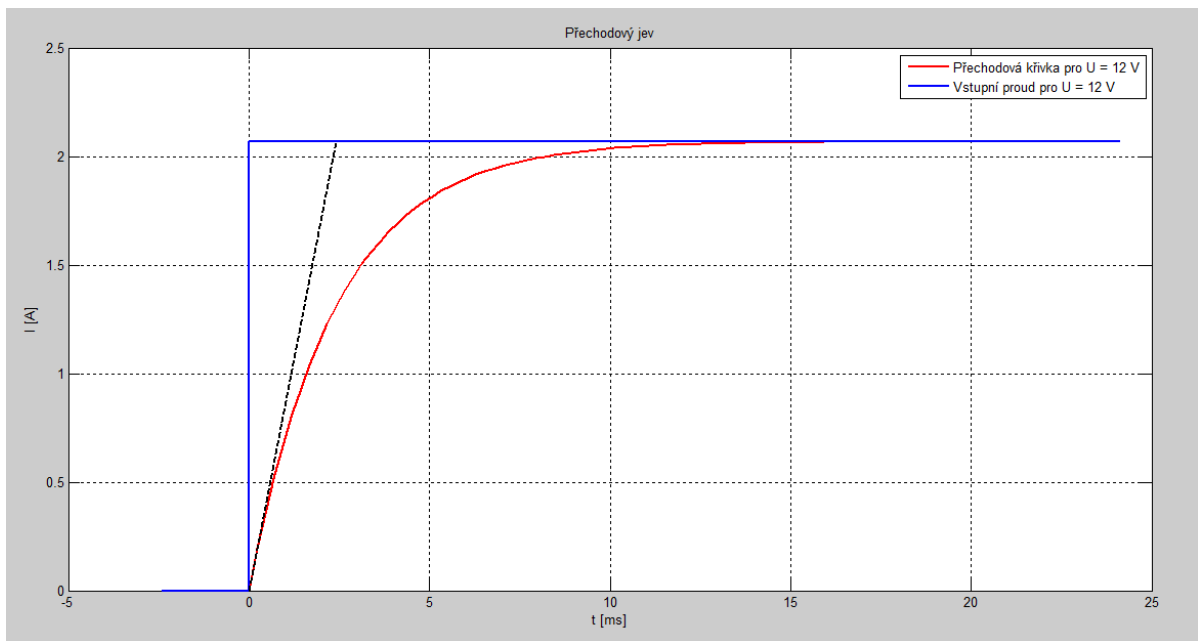
$$t = 0: \quad i(t) = 0$$

$$0 = C_1 e^0 + \frac{U}{R} \quad (65)$$

$$C_1 = -\frac{U}{R} \quad (66)$$

Výsledná rovnice popisující přechodový jev na cívce statoru

$$i(t) = -\frac{U}{R} e^{-\frac{1}{\tau}t} + \frac{U}{R} = \frac{U}{R} \left(1 - e^{-\frac{1}{\tau}t}\right) = \frac{U}{R} \left(1 - e^{-\frac{R}{L}t}\right) \quad (67)$$



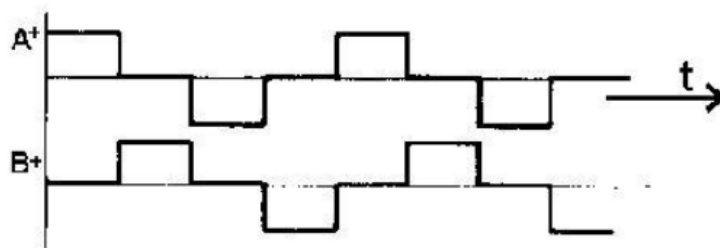
Obr. 17 Grafické znázornění přechodového jevu

Krokové motory je možné řídit v tzv. otevřené vazbě. Natočení či posuv je možné monitorovat odpočítáváním jednotlivých kroků posílaných do motoru bez jejich ověřování senzory zpětné vazby. (Doc. Ing. BĚLOHOUBEK, 1998)

Krokové motory jsou běžně používány v tiskárnách pro nastavování tiskové hlavy, na plotrech, 3D tiskárnách, přesné natáčení klapků a ventilů a jiných zařízeních, kde je potřeba přesné polohování bez vysokých rychlostí. Dosahují menších rychlostí než DC motory, běžně 5- 600 pulzů za sekundu. Krokové motory však vykazují vysoký krouticí moment v nízkých rychlostech a vysoký moment při držení statické polohy. (ONWUBOLU, 2005)

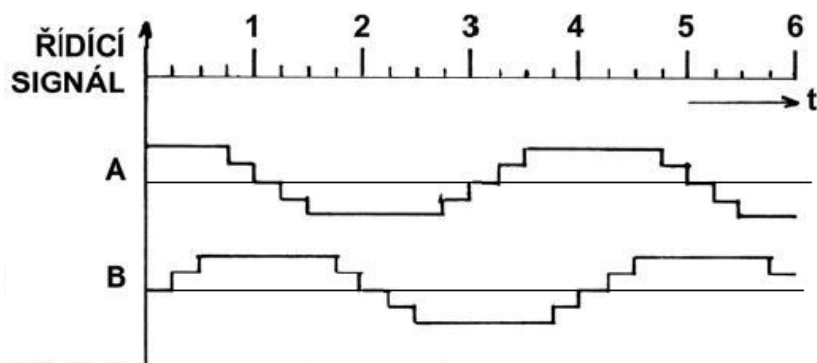
### 2.5.1 Mikrokrokování

Počet a velikost jednotlivých kroků je dána konstrukčním uspořádáním motoru. Úhel kroku může být, dle konstrukce, v rozmezí  $0,9^\circ - 180^\circ$ . Krok u běžně nabízených motorů bývá  $1,8^\circ$  a rozlišení 200 kroků na otáčku. (MAIXNER, 2006) Proud musí být přiváděn do cívek postupně podle dané sekvence. Pokud máme bipolární krokový motor, tak jsou k dispozici dvě vinutí. Zuby rotoru se vždy zarovnají s nejbližším vinutím, které je právě vybuzené. (SEMRÁD, 2012)



Obr. 18 Průběh přiváděného proudu na cívky označené A a B (SEMRÁD, 2012)

Konstrukční náročnost při zmenšování velikosti kroků narůstá, proto se v praxi používá elektronické zmenšení kroku, tzv. mikrokrokování, které dělí jeden krok motor na další dílky stejné velikosti. (SEMRÁD, 2012) V praxi je možné běžně dosáhnout 64, 128 i více mikrokroků na jeden krok. Maximální prakticky použitelné rozlišení je 4000 kroků na otáčku. Menší kroky již zanikají v kmitání hřídele a šumu. Při klasickém buzení teče proud vždy jen do jednoho vinutí motoru, se kterým se zarovná rotor. Při mikrokrokování dochází k rozvážení proudů mezi obě cívky a tím lze získat téměř jakoukoli rovnovážnou stabilní polohu mezi cívkami. (Ing. RYDLO, 2000) Je ale nutné použít vícehladinový spínaný zdroj nebo řídicí elektroniku. Z grafického znázornění na obr. 18 je vidět, že při zvyšování počtu mikrokroků je přiváděným proudem aproximována funkce sinus.



Obr. 19 Průběh přiváděného proudu při použití 4 mikrokroků (Ing. RYDLO, 2000)

V praxi se používají řídicí integrované obvody pro dělení proudu do jednotlivých vinutí. Pomocí spínacích tranzistorů je možné dělit kroky až na zmíněných 128 mikrokroků. (Ing. RYDLO, 2000) Pro jednodušší použití jsou obvody sestaveny tak, aby bylo možné povel pro krok posílat pomocí PWM pulzů, kdy jeden pulz znamená jeden krok. Směr otáčení je řízen dalším signálem, který podle jeho logické hodnoty posílá proudy tak, aby rotace probíhala požadovaným směrem. (SEMRÁD, 2012)

### 2.5.2 Typy krokových motorů

Nejčastější konstrukce krokových motorů je s rotačním výstupem, která může být vyrobena ve více provedeních. (MAIXNER, 2006) Jedním z variant provedení jsou krokové motory s aktivním rotorem. Stator motoru je tvořen elektromagnety, vzájemně natočenými po obvodu o přesně dané úhly. (Prof. Dr. Ing. SCHMID, 2005) Rotor je tvořen plechy s vyfrézovanými zuby a obsahuje permanentní magnety, které mohou být radiálně nebo axiálně polarizované. Motory s aktivním rotorem mají mnohem vyšší momenty než motory s pasivním rotorem.

Druhou variantou provedení rotačního výstupu jsou krokové motory s pasivním rotorem (reluktanční). Rotor je vytvořený svazkem plechů, které jsou nalisovány na hřídel a tvoří zuby. Stator je opět tvořen vinutím elektromagnetů a je opatřen zuby stejné šířky jako na rotoru. Zuby na obou částech se vzájemně nekryjí a jsou posunuty o  $\frac{1}{2}$  rozteče. (Doc. Ing. BĚLOHOUBEK, 1998)

Posledním provedením jsou krokové motory s odvalujícím se rotorem. Při této konstrukci se rotor na několika místech dotýká statoru, po kterém se odvaluje. Zejména jsou určeny pro pohyb po velmi malých krocích a vysokých frekvencích pulzů. (Doc. Ing. BĚLOHOUBEK, 1998)

Krokové motory lze vyrobit i ve variantě lineárního pohonu. Konstrukce lineárních motorů se shodují již zmíněnými, ale jsou rozvinuty do přímky. Stator je v tomto případě primární část, buzený magnetický obvod, a rotorem je kotva se zuby. Mohou být vyrobeny s aktivním i pasivním rotorem. Pohyb je realizován střídavým spínáním elektromagnetů, tak aby se rotor pohyboval v požadovaném směru. (MAIXNER, 2006)

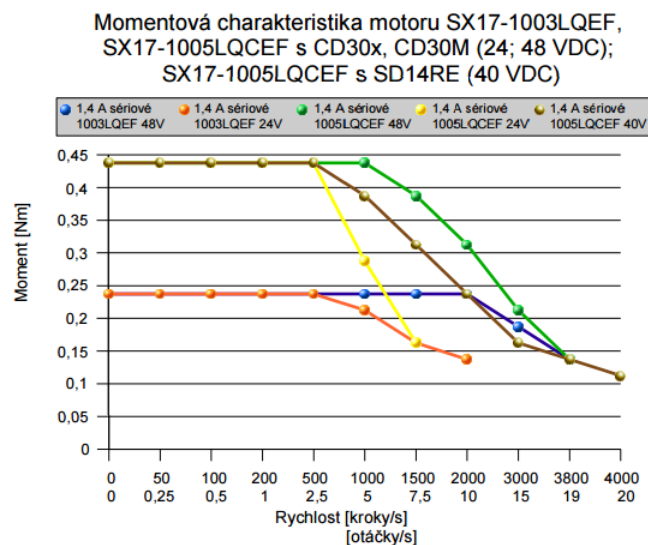
### 2.5.3 Srovnání se servomotory

Mezi hlavní výhody použití krokových motorů patří použití jednodušších řídicích algoritmů, kde je potřeba pouze generovat PWM pulzy a signál řídicí směr otáčení. Krokový motor nemá žádné nepohyblivé části, které by se dotýkaly rotoru jako je tomu v případě kartáčů stejnosměrných motorů a mají tedy vyšší spolehlivost provozu a delší životnost. Pokud se však srovnání provede s bezkartáčovými motory s elektronickou komutací, tak se tato výhoda ztrácí. (Doc. Ing. BĚLOHOUBEK, 1998) Již z principu funkce vychází, že krokový motor má diskrétní počet poloh na jednu otáčku, tedy omezenou přesnost natočení a při malých rychlostech motor výrazně krocuje jednotlivé polohy. O toho se do konstrukce mohou přenášet vibrace. Servomotor je proti tomu řízen spojitě a otáčku vykonává spojitě. Lze nastavit libovolnou polohu rotoru i plynule pracovat při pomalé rychlosti.



Pokud motor není přetěžován, tak lze počítat pulzy poslané na motor a dopočítávat aktuální polohu rotoru, tzv. lze ho řídit v otevřené vazbě (ovládat). S touto vlastností souvisí i jeho hlavní nevýhoda a to je nutnost provozu motoru pouze na přibližně 60% maximálního krouticího momentu jinak motor vypadne ze synchronního režimu a dojde ke ztrátě vztahu mezi počtem pulzů a počtem vykonaných kroků a tím ke ztrátě informace o aktuální poloze. Tuto skutečnost lze eliminovat použitím polohových senzorů, jako jsou inkrementální či absolutní snímače nebo resolvery. Tím se však ztrácí jednoduchost konstrukce motoru a je výhodnější použít elektrický servomotor. (Doc. Ing. BĚLOHOUBEK, 1998)

Druhou velkou nevýhodou krokového motoru je závislost krouticího momentu na rychlosti otáčení, respektive na frekvenci pulzů posílaných na motor. Příčina tohoto chování byla popsána v kapitole 2.5. Jako příklad lze uvést konkrétní momentové charakteristiky poskytnuté výrobcem. Z obrázku je vidět, že tyto konkrétní typy motorů mají stabilní moment do frekvence 500 kroků za sekundu a poté jeho velikost prudce klesá. Nelze tyto motory použít pro vyšší rychlosti. (Microcon)

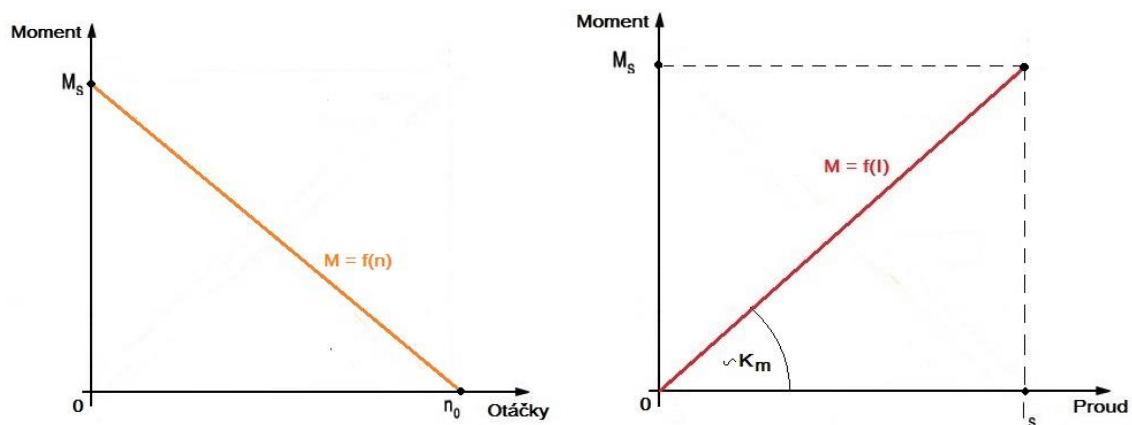


Obr. 20 Momentové charakteristiky krokových motorů řady SX-17, výrobce Microcon, s.r.o. (Microcon)

Na rozdíl od krokových motorů mají servomotory s permanentními magnety pracující při konstantním napětí velikost krouticího momentu jako funkci otáček rotoru  $M = f(n)$  a proudu tekoucího motorem  $M = f(I)$ . Při zvyšování otáček moment motoru lineárně klesá, ale při zvyšování proudu lineárně roste. Vysoký moment lze udržovat i ve vysokých otáčkách. Pro určení momentu servomotoru s permanentními magnety z napájecího proudu je experimentálně zjišťována momentová konstanta a platí vztah

$$M = K_m \cdot I, \quad (68)$$

kde  $M$  je moment motoru,  $I$  napájecí proud a  $K_m$  momentová konstanta motoru.  
(ROHAL, 2007)

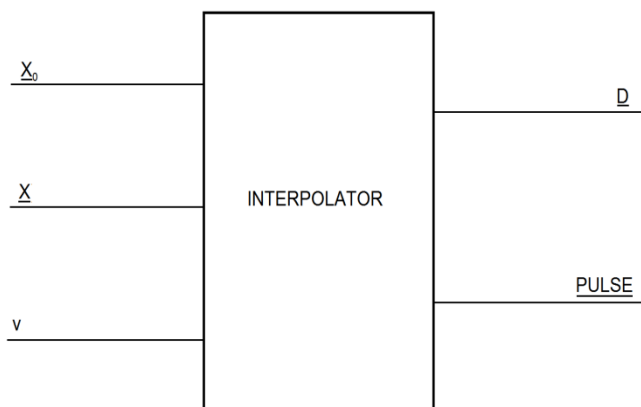


Obr. 21 Momentové charakteristiky stejnosměrného servomotoru s permanentními magnety. Hodnoty s indexem S jsou záběrové a hodnoty s indexem 0 jsou při chodu naprázdno (ROHAL, 2007), upraveno autorem

### 3 INTERPOLÁTOR

V průmyslové praxi jsou firmami vyrábějícími CNC obráběcí stroje používány již předpřipravené struktury programů, které tvoří základ daného řídicího softwaru. V praktické části bakalářské práce jsem psal vlastní variantu algoritmu pro lineární interpolaci dráhy mezi dvěma body.

Nejdříve jsem sestavil blokové schéma řídicího programu a stanovil si obecné funkční vztahy pomocí veličit potřebných k výpočtu. Vstupními hodnotami je počáteční poloha  $\underline{X}_0$ , koncová poloha  $\underline{X}$  a požadovaná rychlost pohybu  $v$ . Výstupními hodnotami jsou pravoúhlé pulzy v proměnné PULSE o

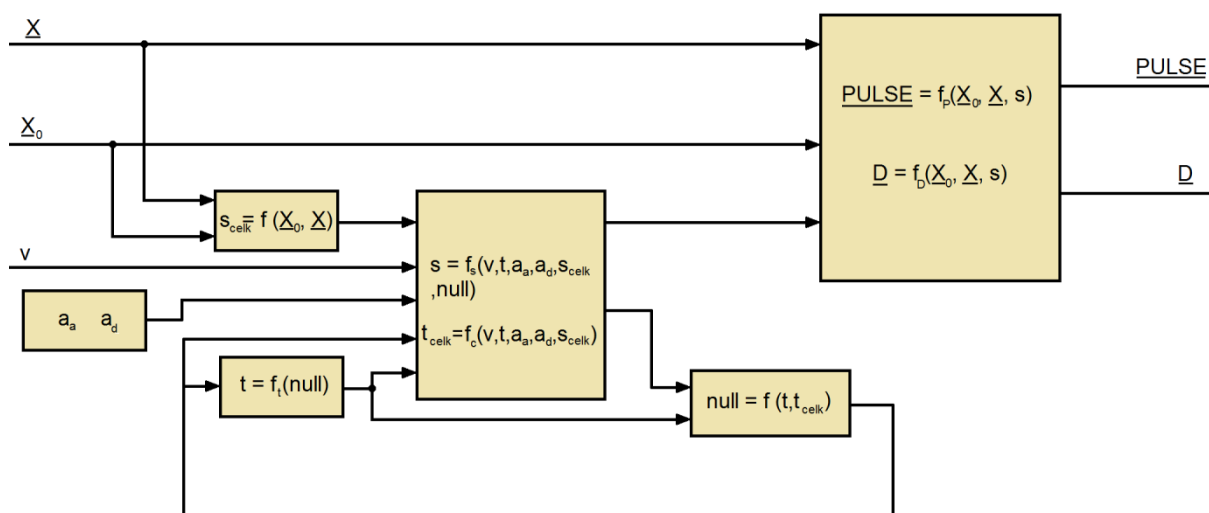


dané frekvenci a požadavek na směr rotace motoru D.

Obr. 22 Znárodnění vstupů a výstupů z bloku interpolátoru

Proměnná PULSE je funkcí tří veličin  $PULSE = f_P(\underline{X}_0, \underline{X}, s)$ , souřadnic počáteční polohy, souřadnic koncové polohy a teoretické křivky profilu polohy, která je rozpočítána na jednotlivé osy. Křivka polohy se vypočítává ze vstupních parametrů a programových konstant  $s = f_s(v, t, a_a, a_d, s_{celk}, null)$ . Je funkcí celkové vzdálenosti mezi body  $s_{celk}$ , požadované rychlosti  $v$ , maximálního zrychlení  $a_a$  a zpomalení  $a_d$ , času  $t$  a signálu indikujícího dosažení polohy  $null$ . Celková vzdálenost je funkcí počáteční a koncových souřadnic  $s_{celk} = f_\Delta(\underline{X}_0, \underline{X})$  a signál  $null$  je funkcí celkového času potřebného k ujetí dráhy a aktuálního času  $null = f_n(t, t_{celk})$ .

Směr rotace D je vypočítáván ve stejném bloku jako PULSE a je funkcí stejných veličin  $\underline{D} = f_D(\underline{X}_0, \underline{X}, s)$ .



Obr. 23 Blokové schéma navrhovaného programu

### 3.1 Interpolační profil

Ze vstupních hodnot do interpolátoru, souřadnic počáteční polohy, souřadnic koncové polohy, požadované rychlosti a požadované koncové rychlosti, je vypočítáván profil průběhu rychlosti a polohy. Teoreticky vypočtená aktuální poloha je porovnávána se známou polohou v generátoru pulzů.

Z vektorů souřadnic počáteční a koncové polohy se určí vektor potřebných přírůstků souřadnic na každé ose.

$$\Delta \underline{X} = \underline{X} - \underline{X}_0 \quad (69)$$

Pomocí Pythagorovy věty se vypočte vzdálenost mezi body v prostoru a tím se získá délka dráhy  $s_{celk}$  pro výpočet rychlostního profilu.

$$s_{celk} = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \quad (70)$$

Pro vytvoření průběhu profilu rychlosti a polohy jsou nejdříve ze známé počáteční rychlosti  $v_0$ , požadované rychlosti  $v_p$  a koncové rychlosti  $v_k$  vypočteny časy potřebné pro zrychlení a zpomalení pohybu.

$$t_a = \frac{v_p - v_0}{a_a} \quad ; \quad t_d = \frac{v_p - v_k}{a_d} \quad (71)$$

Výpočet je proveden pro dva případy, které mohou nastat. V prvním případě je předpokládán plný lichoběžníkový profil rychlosti. Ještě je kontrolována skutečnost, jestli je počáteční rychlost menší nebo větší než požadovaná. Podle toho je čas  $t_a$  vypočten s hodnotou zrychlení nebo zpomalení. V druhém případě je předpokládána kratší dráha tak, že

nevystačí na dosažení požadované rychlosti. Pro výpočet těchto časů je nutné nejdřív určit skutečnou dosaženou rychlost při předpokládané trajektorii.

$$v_{traj} = \sqrt{\frac{s_{celk} + \frac{v_{poč}^2}{2a_1} + \frac{v_{konc}^2}{2a_3}}{\frac{1}{2a_1} + \frac{1}{2a_3}}} \quad (72)$$

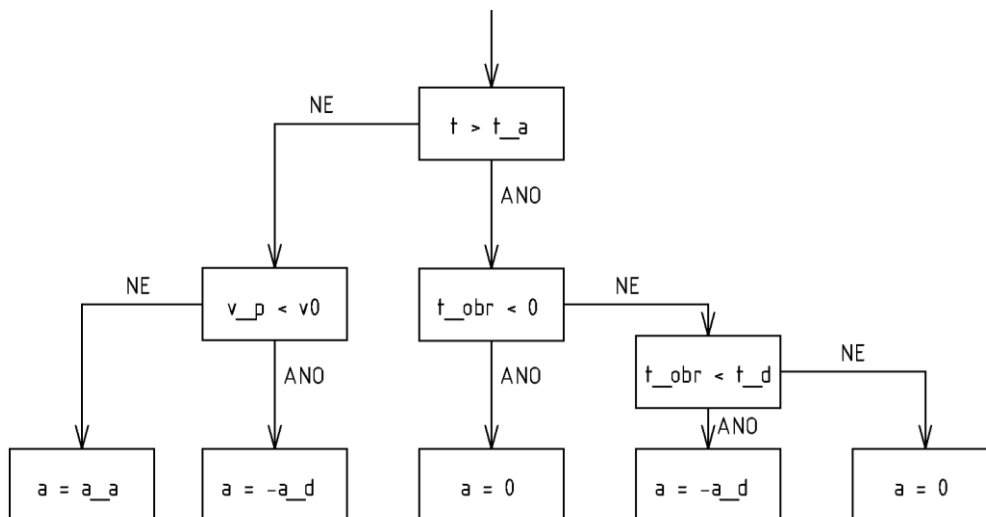
Součet těchto časů je porovnán. Pokud je větší součet z prvního případu, tak nastává tato varianta a dosahovaná rychlost je rovna požadované  $v_p$ . V opačném případě nastává druhý případ pohybu a dosahovaná rychlost je vypočtená  $v_{traj}$ .

Jakmile je identifikován pohybový případ, jsou vypočteny dráhy potřebné pro zrychlení a zpomalení pohybu. Pokud je dosahována požadovaná rychlost, tak je vypočtena i dráha ujetá při této konstantní rychlosti.

$$s_1 = v_0 t_1 + \frac{1}{2} a_1 t_1^2 ; \quad s_2 = v_p t_2 ; \quad s_3 = v_p t_3 - \frac{1}{2} a_3 t_3^2 \quad (73)$$

Následuje kontrola proveditelnosti pohybu. Porovnává se, jestli je požadovaná dráha delší, než je dráha potřebná pro zrychlení nebo zpomalení. V takovém případě by nebyl rychlostní profil při daných podmínkách realizovatelný a program vypíše hlášku informující uživatele o této skutečnosti. Algoritmus pro řešení realizovatelnosti velmi krátkých drah bude popsán v kapitole 3.6.

Z drah jednotlivých úseků se dopočítají zbylé neznámé časy úseků pohybu a určí se celkový čas  $t_{celk}$  pro vykonání dráhy. Jednotlivé časy slouží ke generování průběhu zrychlení. Algoritmus je sestaven tak, aby při dosažení konkrétních časů generoval konstantu zrychlení, zpomalení nebo nulu. Pokud je aktuální čas  $t$  menší než čas potřebný pro zrychlení  $t_a$ , tak se ještě přejde na podmínku, jestli je počáteční rychlost  $v_0$  větší než požadovaná  $v_p$ . Pokud je i tato druhá podmínka splněna, je generována konstanta zpomalení  $-a_d$ , a pokud není splněna, je generována konstanta zrychlení  $a_a$ . Jakmile aktuální čas překročí čas pro zrychlení, hodnota zrychlení se začne rovnat nule a probíhá úsek s konstantní rychlostí. Aby mohl být určen okamžik, kdy je nutné začít zpomalovat, byla zavedena proměnná  $t_{obr} = t_{celk} - t$ , která čas od celkového odpočítává. Jakmile je  $t_{obr}$  menší než čas pro zpomalení, je generována konstanta zpomalení  $-a_d$  až do doby dosažení celkového času  $t_{celk}$ .



Obr. 24 Rozhodovací algoritmus pro generování průběhu zrychlení

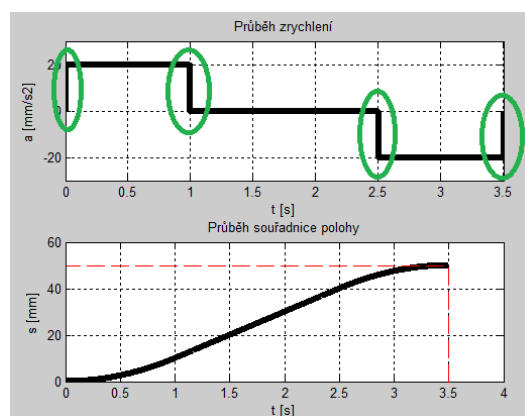
Po provedení uvedeného algoritmu program získá vektor průběhu zrychlení podél celé trajektorie. Pro získání rychlosti již stačí provést numerickou integraci získaného zrychlení. Integrováním konstanty vznikne přímka a z nuly vznikne konstanta. Tím dostaneme lichoběžníkový rychlostní profil. Pro průběh souřadnice polohy se provede ještě druhá numerická integrace rychlostního profilu. Průběh se křivky se zvýší o jeden stupeň a průběh se vyhladí.

Při použití numerických metod je potřeba zkontrolovat velikost chyby zvolené metody, zda není příliš velká, aby znehodnotila vypočítaný průběh. Pro odhad velikosti chyby lze použít rovnici (22), kde při odhadu chyby výpočtu rychlosti potřebujeme druhou derivaci zrychlení a při odhadu chyby polohy druhou derivaci rychlosti. Lichoběžníkový rychlostní profil má obě tyto druhé derivace nulové.

$$a = konst. \quad \rightarrow \quad \dot{a} = \ddot{a} = 0 \quad (74)$$

$$v = at \quad \rightarrow \quad \dot{v} = a ; \ddot{v} = \dot{a} = 0 \quad (75)$$

Při derivování takto jednoduchých funkcí je chyba zvolené numerické metody nulová. Největší chyba, která může nastat je vlivem velikosti výpočtového kroku programu. Při přepínacích časech se může stát, že přepnutí zrychlení vychází do mezikroku, ale prakticky k němu dojde až po čase  $\Delta t$ . Maximální velikost chyby polohy a rychlost může být teoreticky ve velikosti jednoho



Obr. 25 Vyznačení přepínacích míst zrychlení

kroku. Pro jednu dvojici souřadnic mohou být až 4 přepnutí v průběhu zrychlení. Výsledná maximální chyba pro jeden blok je rovna čtyřnásobku jednotlivé chyby.

$$v_{ch} = 4 \cdot a \cdot \Delta t \quad (76)$$

$$s_{ch} = 4 \left( \frac{1}{2} a \Delta t^2 + v_{ch} \Delta t \right) = 2a\Delta t^2 + 4v_{ch}\Delta t \quad (77)$$

Pro zrychlení o velikosti  $a = 10 \text{ mm/s}^2$ , požadované rychlosti  $v_p = 4 \text{ m/s}$  a velikost časového kroku programu  $\Delta t = 0,0001 \text{ s}$  je maximální velikost chyby pro jednu dvojici souřadnic následující.

$$v_{ch} = 4 \cdot a \cdot \Delta t = 4 \cdot 10 \cdot 0,0001 = 0,004 \text{ mm/s} \quad (78)$$

$$s_{ch} = 2a\Delta t^2 + 4v_p\Delta t = 2 \cdot 10 \cdot 0,0001^2 + 4 \cdot 0,0001 = 0,0004 \text{ mm} \quad (79)$$

Z vypočtených hodnot lze usoudit, že chyba numerické integrace je pro zvolený účel zanedbatelná. Ale při velmi dlouhých programech obsahujících stovky souřadnic se bude chyba sčítat a pro např. 2500 souřadnic bude již velikost chyby polohy 1 mm. Program lze sestavit i tak, aby generoval přímo průběh dráhy. Takové řešení je vhodnější pro složitější rychlostní profily, kde by numerickými metodami vznikala příliš velká chyba. Zároveň je generování o něco náročnější při spojitěm napojování jednotlivých úseků.

Takto probíhá výpočet pro jednu dvojici souřadnic. Aby se cyklus mohl opakovat i pro následující dvojice souřadnic, musí být zařazeny ještě další pomocné podprogramy. Pro signalizaci dosažení polohy je nutné monitorovat výpočtový čas a neustále ho porovnávat se celkovým časem potřebným pro vykonání dráhy. Jako signalizační proměnná byla zavedena *null*. Jestliže je výpočtový čas větší než celkový čas, tak proměnná *null* je rovna nule, jinak je rovna jedné. Na tuto signalizaci reagují další části programu.

Jedním z těchto algoritmů je časovač, který generuje výpočtový čas. Během generování dráhy je v každé iteraci přičtena velikost časového kroku.

$$t_i = t_{i-1} + \Delta t \quad (80)$$

Zároveň je čas násoben proměnnou *null*. Když *null* = 1, tak se nic neděje a čas se přičítá, ale pokud *null* = 0, tak se vynuluje i výpočtový čas a pro novou dvojici souřadnic se opět začíná přičítat od nuly.

Druhý algoritmus reagující na proměnnou *null* je paměťový blok pro počáteční rychlost  $v_0$ . Pokud je *null* = 1, tak se počáteční rychlost rovná předchozí hodnotě v interpolaci jedné dvojice souřadnic. Jakmile je interpolace dokončena a *null* = 0, tak si

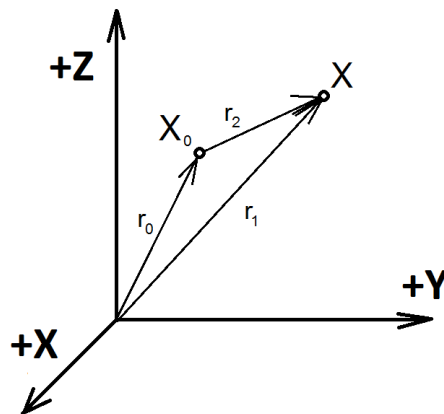
tento blok zapamatuje aktuální rychlost a tím je tato rychlost v následující dvojici souřadnic použita jako  $v_0$ .

### 3.2 Rozdělení přirozené souřadnice polohy

Vstupními parametry do části programu pro rozdělení posuvu na jednotlivé osy jsou tři prostorové souřadnice počátečního bodu  $\underline{X}_0$ , tři prostorové souřadnice požadovaného bodu  $\underline{X}$  a profil vykonané dráhy  $s$ . Pro řízení pohybu je potřeba rozložit přirozenou souřadnici do složek na jednotlivé osy. K rozkladu lze využít zmíněné dva body v prostoru. Nejdříve se určí vzdálenosti mezi body z rozdílu polohových vektorů.

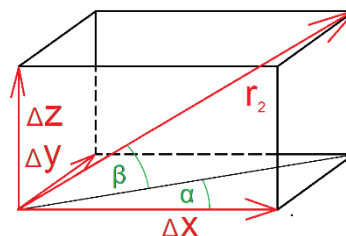
$$\underline{r}_2 = \underline{r}_1 - \underline{r}_0 \quad (81)$$

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \quad (82)$$



Obr. 26 Znáornění polohových vektorů

Z jednotlivých složek polohového vektoru  $\underline{r}_2$  lze vypočítat úhly, které jsou svírány s jednotlivými osami souřadného systému.



Obr. 27 Pomocný obrázek pro určení úhlů  $\alpha$  a  $\beta$

$$\alpha = \arctg \left( \frac{|\Delta y|}{|\Delta x|} \right) \quad (83)$$

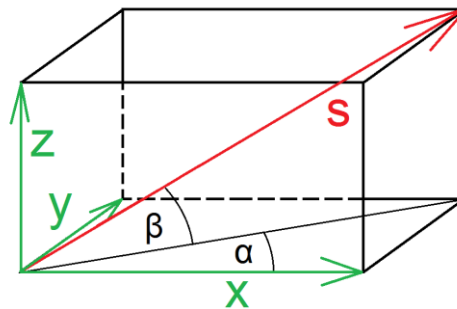


$$\beta = \operatorname{arctg} \left( \frac{|\Delta z|}{\sqrt{\Delta x^2 + \Delta y^2}} \right) \quad (84)$$

Úhly jsou počítány z absolutních hodnot souřadnic, aby jejich velikost vyšla v intervalu  $\langle 0; \frac{\pi}{2} \rangle$ . Vytvoření orientovaného úhlu je pomocí logických funkcí IF, které přičítají k vypočteným úhlům hodnoty  $\frac{\pi}{2}$ ,  $\pi$  nebo  $2\pi$  tak, aby odpovídaly skutečnému směru vektoru  $\underline{r}_2$ . Toto řešení bylo zvoleno z důvodu nespolehlivosti výpočtu správné velikosti úhlu při záporných rozdílech souřadnic na některých osách.

Se získanými směrovými úhly je již možné vypočítat složky přirozené souřadnice do jednotlivých směrů x, y a z

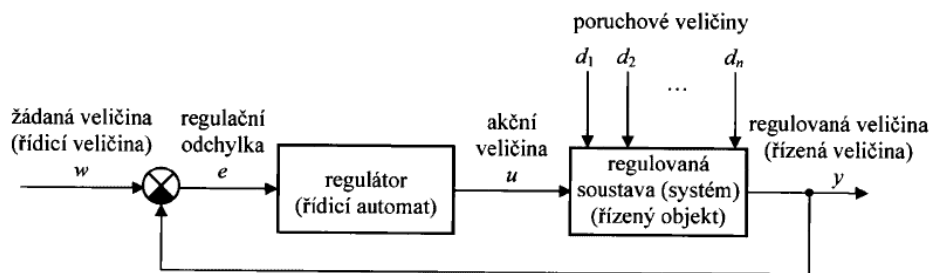
$$\underline{s} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} s \cdot \cos \beta \cdot \cos \alpha \\ s \cdot \cos \beta \cdot \sin \alpha \\ s \cdot \sin \beta \end{bmatrix} = s \begin{bmatrix} \cos \beta \cdot \cos \alpha \\ \cos \beta \cdot \sin \alpha \\ \sin \beta \end{bmatrix} \quad (85)$$



Obr. 28 Rozklad přirozené souřadnice do složek na jednotlivých osách

### 3.3 Generátor pulzů

Generování pulzů pracuje na stejném principu jako zpětnovazební regulační obvod. Žádaná veličina je poloha generovaná podle matematického profilu a rozdělená do jednotlivých os. Je porovnávána s regulovanou veličinou, což je aktuální poloha. Vzniká regulační odchylka, která postupně narůstá. Jakmile je větší nebo rovna velikosti jednoho kroku, tak se tento krok provede vysláním pulzu. Žádaná a regulovaná veličina se na okamžik vyrovnají, ale poté začne žádaná veličina opět narůstat. Tento proces se neustále opakuje a generuje pulzy.

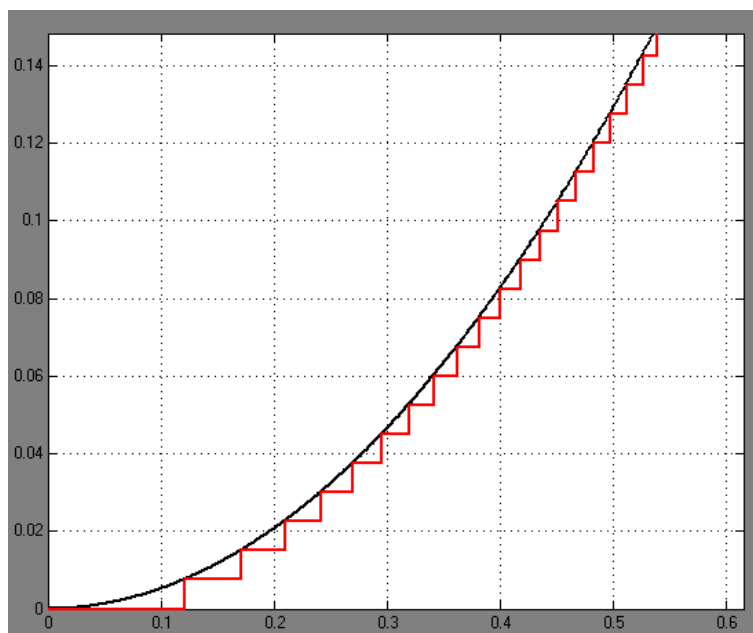


Obr. 29 Regulační obvod (Prof. Ing. HOFREITER, 2014)

Generátor pulzů je v programu umístěn třikrát, pro každou osu zvlášť. Pro příklad je zde uveden generátor na ose  $x$ . Do bloku vstupuje požadovaná poloha  $x$  na dané ose a konstanta pro velikost jednoho kroku. Uvnitř je vypočítána regulační odchylka  $e$ , která je porovnávána s velikostí kroku.

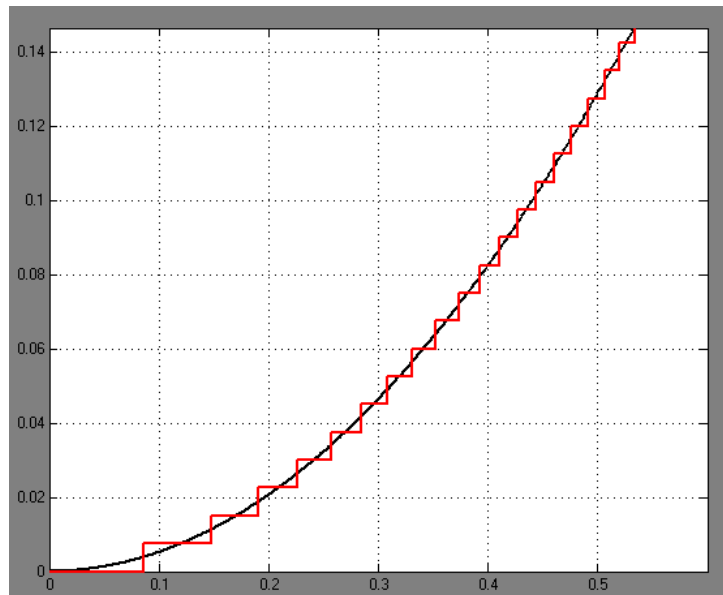
$$e = x - x_{act} \quad (86)$$

Pro narůstání polohy kontroluje podmínka IF, jestli je regulační odchylka větší nebo rovna velikosti kroku. Pokud ano, tak se vygeneruje pulz, pro směr otáčení do proměnné  $D_x$  se запиše hodnota 1 a k aktuální poloze se přičte hodnota o velikosti jednoho kroku. Pro snižování polohy je sestavena stejná podmínka, ale funguje převráceně. Regulační odchylka je záporná, tak se kontroluje, jestli je menší nebo rovna záporné velikosti kroku. Pokud ano, tak se opět vygeneruje pulz, do proměnné  $D_x$  se запиše 0 a od aktuální polohy se odečte hodnota o velikosti jednoho kroku. V případě nesplnění ani jedné z těchto podmínek se aktuální poloha ani směr nemění a žádné pulzy se negenerují.



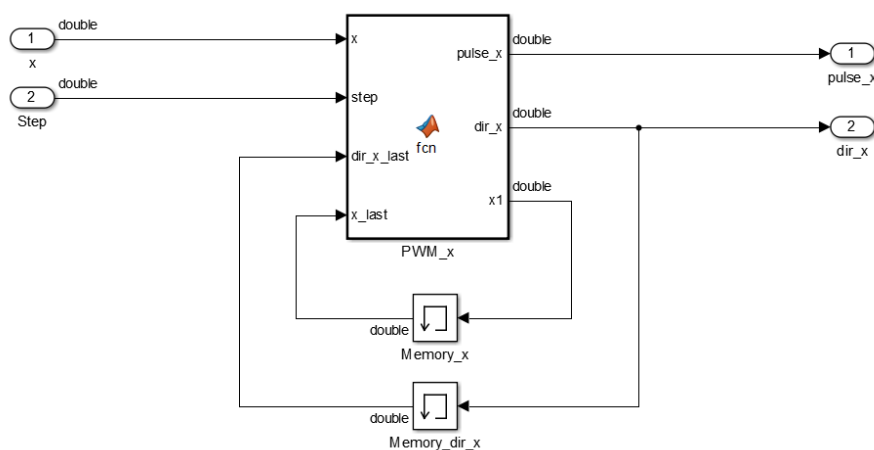
Obr. 30 Požadovaná poloha a generovaná aktuální poloha při regulační odchylce  $e$

Pokud je v podmínce regulační odchylka o velikosti  $e$ , tak se provede krok vždy, když je zjištěn rozdíl mezi aktuální a požadovanou polohou o velikosti jednoho kroku. Aktuální poloha vždy dohání tu požadovanou. Aby byla poloha pomocí kroků lépe aproximována, tak je v programu do podmínky vložena regulační odchylka o velikosti  $2e$ . Ke kroku dojde již v polovičním rozdílu poloh než je velikost kroku a poté se opět čeká na kladný poloviční rozdíl poloh. Generovaná dráha lépe kopíruje požadovanou.



Obr. 31 Požadovaná poloha a generovaná aktuální poloha při regulační odchylce  $2e$

V programu Simulink je algoritmus vytvořen pomocí vložené funkce z Matlabu. Paměť pro aktuální polohu a směr otáčení motoru zajišťují bloky Memory ze Simulink knihovny.



Obr. 32 Simulinkové blokové schéma pro generování pulzů

Za generátor pulzů je ještě zařazen blok, který prodlouží trvání pulzu. Výše popsaným generátorem vznikají pouze pulzy o délce časového kroku  $\Delta t$ . Tak krátký čas je nedostatečný

pro nabití cívky motoru. Doba trvání byla prodloužena funkcí složenou z několika podmínek IF. Pokud je rozdíl aktuální hodnoty proměnné pulz a hodnoty z předchozího kroku roven jedné, tak se začne do proměnné  $n$  přičítat 1 v každém dalším kroku. Na to je navázána podmínka, pokud je  $n > 0$ , tak se hodnota pulzu rovná jedné. Velikost  $v_n$  narůstá až do nastavené velikosti konstantou  $pulse\_width$ . Poté se napočítaná hodnota vynuluje a čeká se, dokud se na vstupu opět neobjeví hrana pulzu. Doba trvání pulzu byla zvolena  $2\tau$  na základě vypočtené přechodové charakteristiky (Obr. 17) tak, aby hodnota byla s určitou rezervou větší než 75% hodnoty ustáleného stavu.

$$2\tau = 0,0048 \text{ s} \quad (87)$$

Největší možná frekvence pulzů vychází z časového kroku programu. Pokud je krok programu  $\Delta t$ , tak nejmenší možná perioda pulzů je rovna dvojnásobku kroku, tedy  $2 \cdot \Delta t$ . Z této skutečnosti vychází maximální frekvence pulzů.

$$f_{max} = \frac{1}{2 \cdot \Delta t} \quad (88)$$

### 3.4 Aktuální poloha v prostoru

Příkazy posílané na motor ve formě jednotlivých pulzů a požadovaného směru otáčení jsou používány k výpočtu aktuální polohy v prostoru. Do podprogramu pro výpočet aktuální polohy vstupuje vektor PULSE a vektor D, aby byla informace ke všem třem osám.

Před vstupem vektoru D je zařazena převodní funkce, protože pro správné určení směru rotace na motor je výstup posílána 1 nebo 0, ale pro výpočet polohy je vhodnější 0 nahradit hodnotou -1. Touto hodnotou lze násobit vektor a změnit jeho orientaci, zatímco 0 by vytvořila nulový vektor. Konverze je provedena podmínkou IF, která sleduje jednotlivé složky vektoru D a pokud je některá z nich rovna 0, tak změní její hodnotu na -1. Příkaz musí být opět prováděn pro každou složku vstupního vektoru zvlášť a vygenerované hodnoty jsou složeny do vektoru výstupního.

V podprogramu je jako konstanta zadána vzdálenost vykonaná při jednom kroku. Bylo nutné ji vypočítat z předpokládané velikosti pohybového šroubu a počtu kroků potřebných pro vykonání jedné otáčky motoru. Krokový motor, pro který je program navržen má konstrukčně dáno  $K_R = 200 \text{ krok/ot}$  a velikost pohybového šroubu je Tr10, který má stoupání  $P = 1,5 \text{ mm}$ . Pro zjištění vykonané dráhy při jednom kroku vydělíme stoupání šroubu počtem kroků motoru.

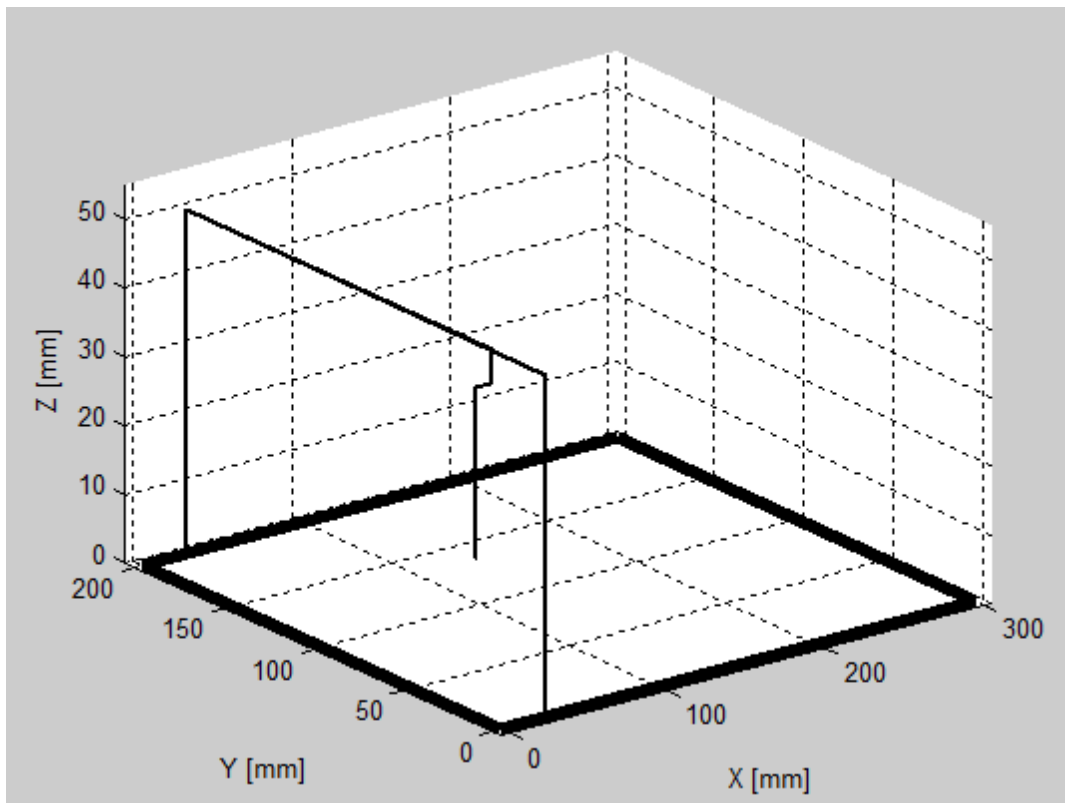
$$\Delta x_{pulse} = \frac{P}{K_R} = \frac{1,5}{200} = 0,007499 \text{ mm/pulse} \quad (89)$$

Poté je kontrolováno vyslání pulzu na motor. Pomocí podmínky IF se pro každou osu zvlášť porovnává pulz z aktuálního časového kroku s předchozím. Předchozí pulz je brán z výstupního, ale pomocí funkčního bloku Memory, který je v knihovně programu Simulink, je zpožděn o jeden časový krok. Pokud je rozdíl těchto pulzů roven jedné, tak je hodnota pro přičtení k poloze rovna vzdálenosti vykonané při jednom kroku, jinak je rovna nule. Pro následné přičítání je sestavena vektorová rovnice.

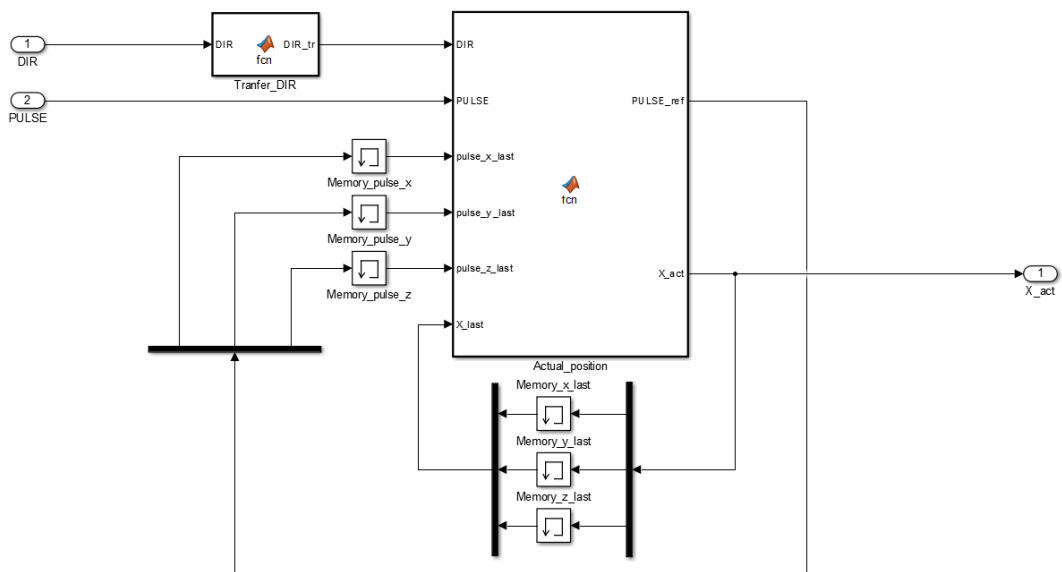
$$\underline{X}_{act} = \underline{X}_{last} + \Delta\underline{X}_{pulse} \cdot \underline{D} \quad (90)$$

$$\begin{bmatrix} x_{act} \\ y_{act} \\ z_{act} \end{bmatrix} = \begin{bmatrix} x_{last} \\ y_{last} \\ z_{last} \end{bmatrix} + \begin{bmatrix} \Delta x_{pulse} \cdot D_x \\ \Delta y_{pulse} \cdot D_y \\ \Delta z_{pulse} \cdot D_z \end{bmatrix} \quad (91)$$

Počáteční hodnoty pro vektor  $\underline{X}_{last}$  odpovídají počáteční poloze v souřadném systému. Aktuální poloha určení vektorem  $\underline{X}_{act}$  je graficky zobrazena v trojrozměrném grafu. Pomocí konstantních vektorů je vytvořen i jednoduchý prostorový model stroje, aby byla lepší představa o aktuální konfiguraci stroje s ohledem na pracovní prostor.



Obr. 33 Grafické znázornění výpočtové polohy v prostoru



Obr. 34 Simulinkové blokové schéma pro výpočet aktuální polohy v prostoru

### 3.5 Souřadnice

Souřadnice jsou zadány v matici o čtyřech sloupcích. V prvních třech je požadovaná poloha postupně v ose X, Y, Z a v posledním sloupci je požadovaná rychlost pohybu  $v$ . V posledním řádku matice musí být vložena referenční hodnota, aby program poznal, že je již na konci a nehlásil chybu neexistujících dat.

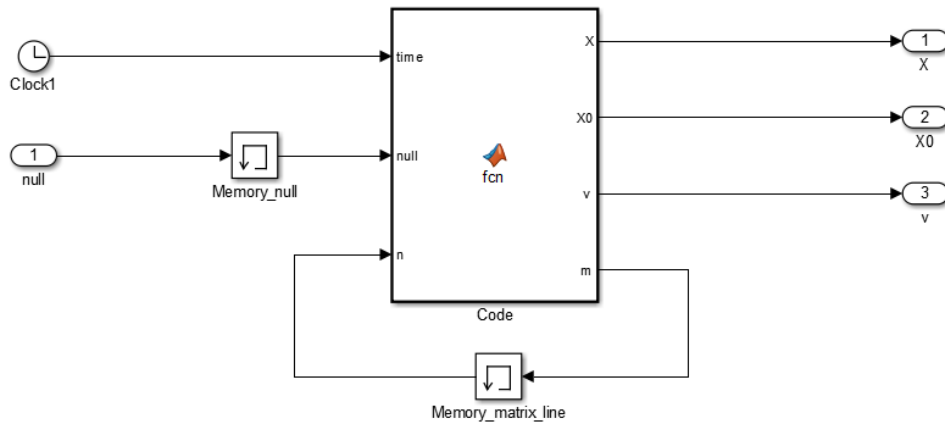
$$\underline{A} = \begin{bmatrix} x_1 & y_1 & z_1 & v_1 \\ x_2 & y_2 & z_2 & v_2 \\ \vdots & \vdots & \vdots & \vdots \\ k_{ref} & k_{ref} & k_{ref} & k_{ref} \end{bmatrix} \quad (92)$$

Podprogram pro generování souřadnic tvoří algoritmus pro vybírání požadovaných prvků z matice.

V prvním kroku programu je definována první dvojice souřadnic. Je zde podmínka, aby pro první krok byla vybrána počáteční souřadnice z prvního řádku matice a koncová souřadnice z druhého řádku matice. První krok programu je definován tak jako  $t_{sim} \leq \Delta t$ . Načítání dalších souřadnic je řízeno proměnnou *null* signalizující dosažení polohy. Pokud je *null* = 0, tak jsou vybrané řádky posunuty o jeden níž. Např. při dosažení první koncové polohy, která byla vybrána z druhého řádku, se výběr souřadnic posune tak, že původní koncová souřadnice se nyní stane počáteční a nová koncová souřadnice se přečte z následujícího řádku. Zároveň je přičteno číslo do proměnné  $n$ , jejíž hodnota určuje vybíraný řádek. Tento cyklus se opakuje, dokud není na řádku nové koncové polohy  $n + 2$  zjištěna

referenční proměnná  $k_{ref}$ . V tom případě se načítání nových souřadnic ukončí a koncová i počáteční poloha se nastaví rovny hodnotě na řádku předposledním, tedy  $n + 1$ .

Zároveň se souřadnicemi je z matice  $A$  vybírána i požadovaná rychlost pohybu. Hodnota rychlosti odpovídá stejnému řádku, ze kterého jsou vybrány počáteční souřadnice pro danou dvojici poloh v prostoru.



Obr. 35 Simulinkové blokové schéma generátoru souřadnic

### 3.6 Trajektorový plánovač

Je zařazen jako inicializační funkce před spuštěním Simulinkové části programu. Načte zadanou matici souřadnic a rychlostí a propočítá průběh celého pohybu předem. K jednotlivým dvojicím souřadnic přiřadí koncové rychlosti podle zadaného algoritmu. Pokud by program při generování průběhu pohybu narazil na problém nerealizovatelnosti na příliš krátké dráze, upraví předchozí koncovou rychlost tak, aby byl tento problém odstraněn. Výstupem je upravená matice souřadnic s přidaným sloupcem. Zde jsou zapsány informace o koncových rychlostech pohybu.

$$\underline{B} = \begin{bmatrix} x_1 & y_1 & z_1 & v_1 & v_{1k} \\ x_2 & y_2 & z_2 & v_2 & v_{2k} \\ \vdots & & & & \\ k_{ref} & k_{ref} & k_{ref} & k_{ref} & k_{ref} \end{bmatrix} \quad (93)$$

Plánovač je tvořen již popsanými bloky pro generování dráhy. Nejdříve je načtena matice souřadnic, ze které jsou brány souřadnice a požadované rychlosti pohybu. Poté je z těchto informací vytvořen profil zrychlení tak, jak je to popsáno v kapitole 3.1. Numerickou integrací se následně získá profil průběhu rychlosti a profil průběhu polohy.

Koncové rychlosti jsou vypočítávány podle změny směru dráhy následující dvojice souřadnic, která je charakterizována úhlem mezi dvěma vektory. Nejdříve se určí vektory pro porovnání.

$$\underline{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} x_1 - x_0 \\ y_1 - y_0 \\ z_1 - z_0 \end{bmatrix} \quad \underline{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{bmatrix} \quad (94)$$

Poté se pro další výpočty určí ještě jejich velikosti.

$$|\underline{u}| = \sqrt{u_1^2 + u_2^2 + u_3^2} \quad (95)$$

$$|\underline{w}| = \sqrt{w_1^2 + w_2^2 + w_3^2} \quad (96)$$

Ze získaných hodnot lze pomocí rovnice pro výpočet úhlu mezi vektory jeho velikost dopočítat.

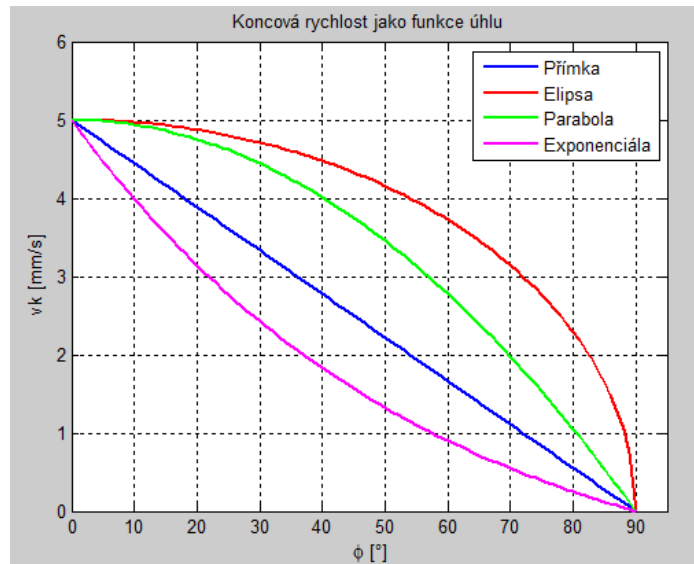
$$\varphi = \arccos\left(\frac{u_1 w_1 + u_2 w_2 + u_3 w_3}{|\underline{u}| \cdot |\underline{w}|}\right) \quad (97)$$

Poté jsem z okrajových podmínek odvodil rovnice různých křivek popisující vztah mezi úhlem mezi vektory a koncovou rychlostí. Pro hodnotu  $\varphi = 0$  je koncová rychlost rovna požadované rychlosti a pro hodnotu  $\varphi = \frac{\pi}{2}$  je koncová rychlost rovna nule. Odvozené rovnice platí pouze pro interval hodnot  $\varphi \in \langle 0; \frac{\pi}{2} \rangle$ . Pro hodnoty větší než  $\frac{\pi}{2}$  je koncová rychlost pohybu rovna nule. Volbou konkrétní rovnice je možné zohlednit dynamiku stroje, tuhost konstrukce a použité pohony.

Typ křivky	Odvozená rovnice
Přímka	$v_k = v_p \left(1 - \varphi \frac{2}{\pi}\right)$
Elipsa	$v_k = v_p \sqrt{1 - \left(\varphi \frac{2}{\pi}\right)^2}$
Parabola	$v_k = v_p \left[1 - \left(\varphi \frac{2}{\pi}\right)^2\right]$
Exponenciála	$v_k = v_p \frac{e^{\frac{\pi}{2}} - e^\varphi}{e^\varphi \left[e^{\frac{\pi}{2}} - 1\right]}$

Tab. 7 Odvozené rovnice pro výpočet koncové rychlosti





Obr. 36 Grafické vykreslení křivek odvozených závislostí

Následuje propočítání parametrů jednoho bloku, jak bylo popsáno v kapitole 3.1. Do kontroly realizovatelnosti pohybu je vložena rovnice přepočítávající rychlosti tak, aby bylo dosaženo realizovatelnosti. Pro zrychlený pohyb se ověřuje podmínka  $s_{celk} \leq s_a$  a zároveň  $v_0 < v_k$ . Pokud je splněna, tak je zjištěn nerealizovatelný případ. Vypočte se maximální dosažitelná změna rychlosti při zrychlování.

$$\Delta v_a = a_a t_a \quad (98)$$

Přepíše se požadovaná koncová rychlost, aby mohla být dosažena, a nová hodnota rychlosti se uloží do matice.

$$v_k = v_0 + \Delta v_a \quad (99)$$

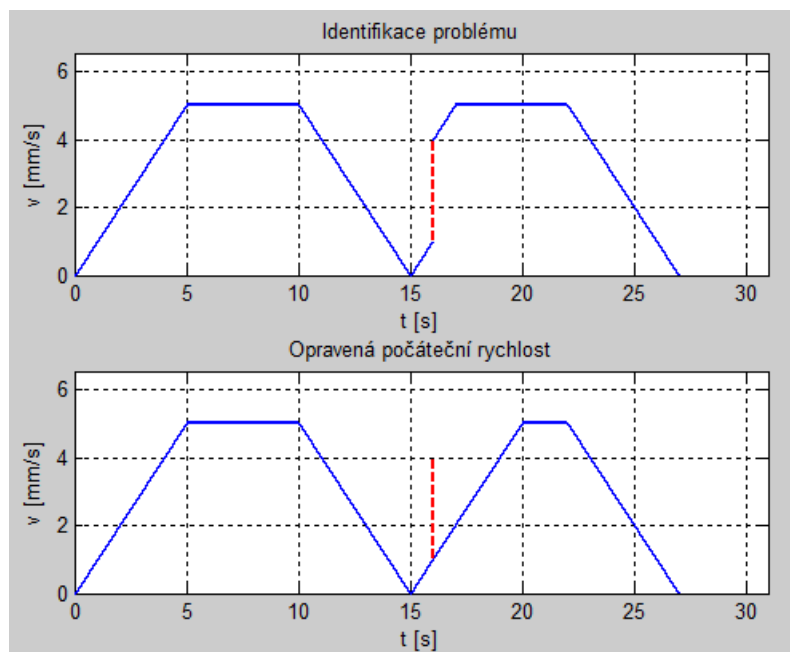
Vynulují se všechny proměnné určující pohyb v matici, sčítací proměnné a výpočtový čas. Výpočet se přesouvá opět na začátek matice a kontroluje dráhy od začátku. Při opakování bloků, kde již byla koncová rychlost vypočtena, případně upravena na dosažitelnou hodnotu, se již pouze její velikost načítá z matice.

V případě nesplnění výše zmíněné podmínky následuje další, která ověřuje, zda jde o zpomalený pohyb  $s_{celk} \leq s_d$  a zároveň  $v_0 > v_k$  a  $s_{celk} \neq 0$ . Při splnění nelze na dané dráze ubrzdít z počáteční rychlosti na koncovou. Opět se vypočítá maximální změna rychlosti.

$$\Delta v_d = a_d t_d \quad (100)$$

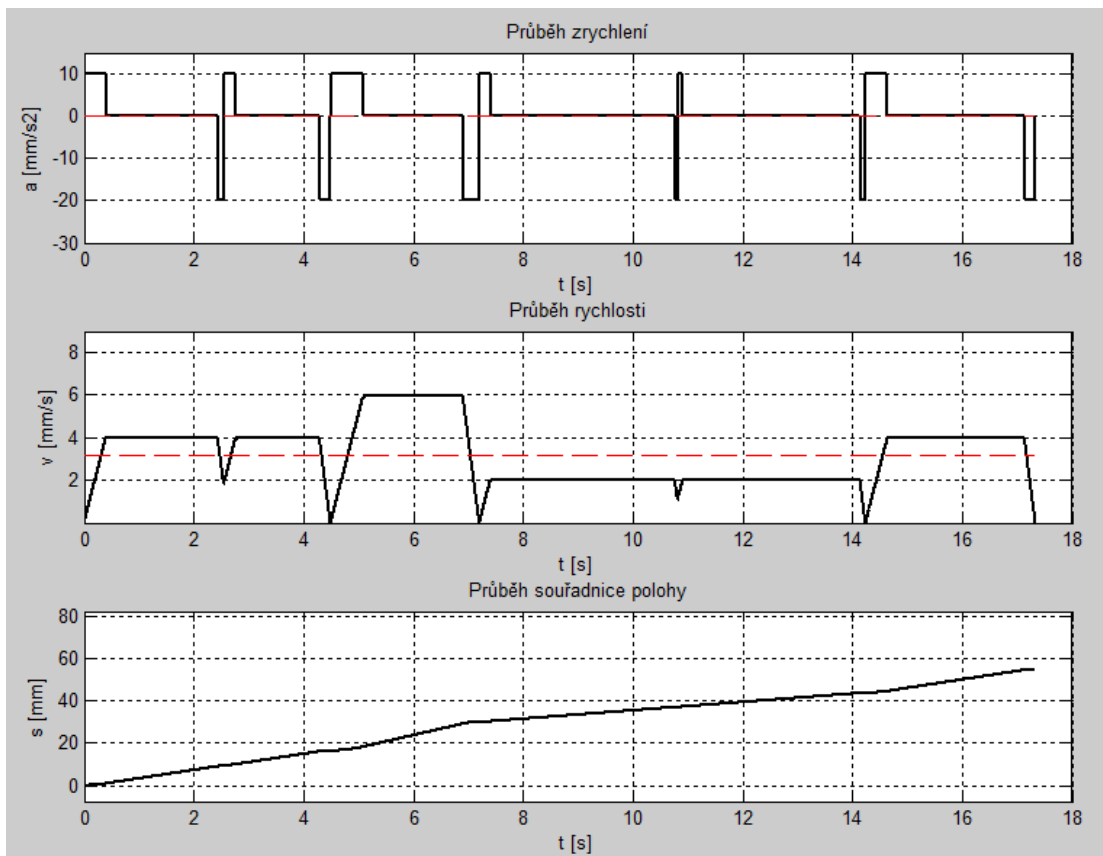
Je upravena počáteční rychlost pohybu, která se uloží do předchozího řádku matice tak, aby vypočtená hodnota byla koncovou rychlostí pro předchozí počítaný blok.

$$v_0 = v_k + \Delta v_d \quad (101)$$



Obr. 37 Ilustrace úpravy počáteční rychlosti

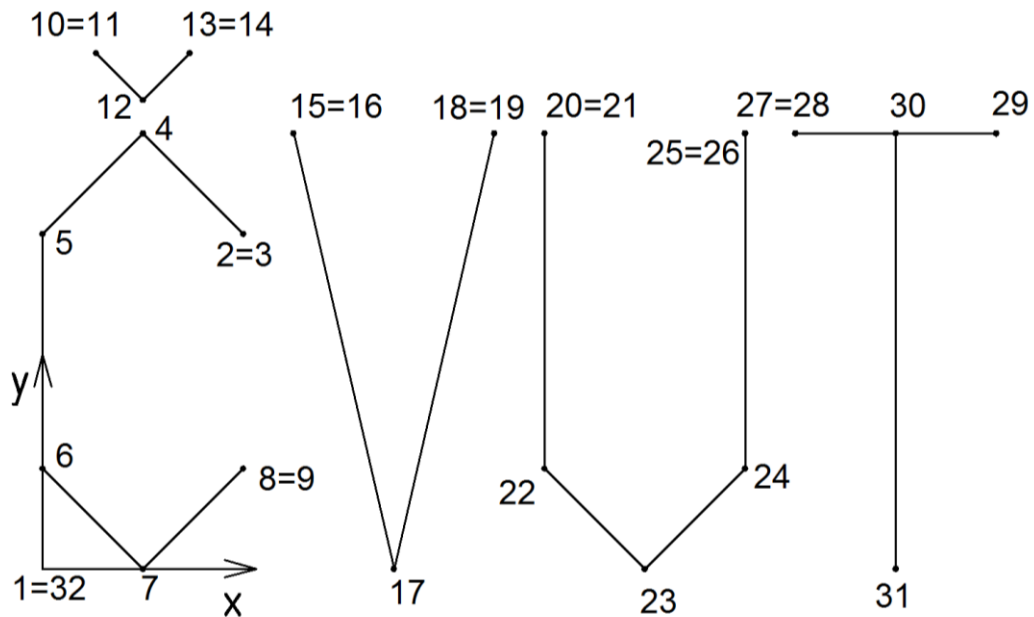
Opět se vynulují všechny proměnné a program se přesouvá na začátek, kde se spouští výpočtový cyklus od začátku. Pokud již program nenarazí na žádnou nerealizovatelnou část, propočítá trajektorie přes všechny zadané souřadnice a vytvoří matici B pro výkonnou část programu v Simulinku. Průběhy kinematických veličin lze vykreslit do grafu před započítáním pohybu a zkontrolovat správnost navržených hodnot.



Obr. 38 Průběhy kinematických veličin přes všechny souřadnice

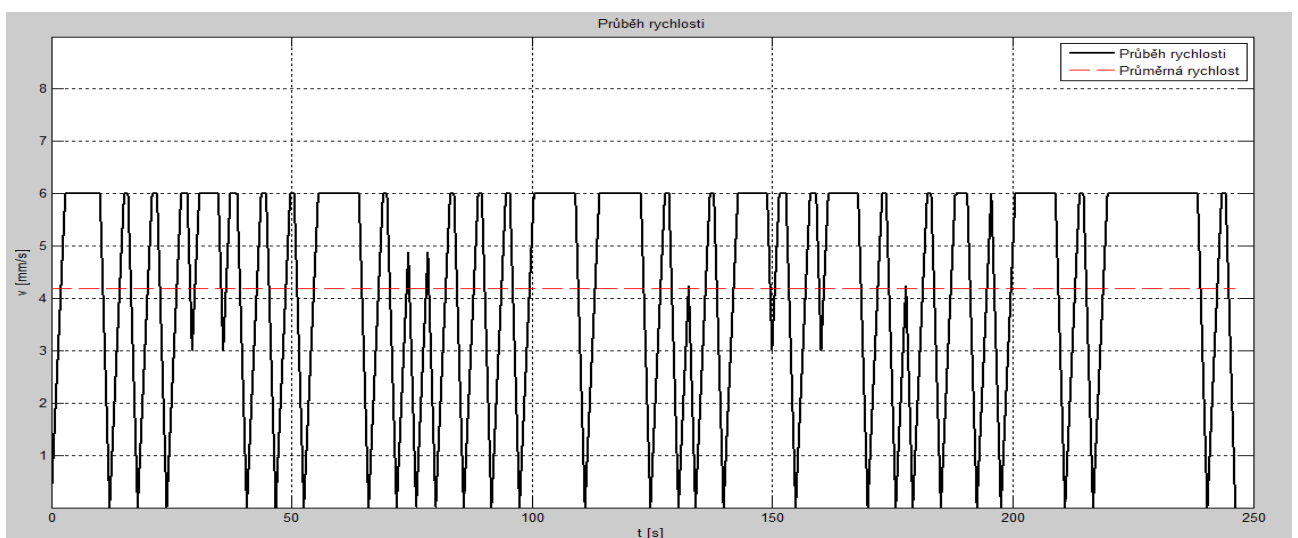
### 3.7 Zkušební úloha

Pro vyzkoušení a demonstrování funkčnosti interpolačního algoritmu jsem zvolil dráhu definovanou více body, která bude zadána v programu. Na ukázkou jsem vybral projetí dráhy v podobě nápisu ČVUT tiskacími písmeny. Jelikož se jedná o lineární interpolátor, bude dráha tvořena pouze přímkami. Souřadnice budou zadány tak, jako by šlo o gravírování nápisu. Osa Z se pohybuje v „záběru“ pouze při objíždění obrysu písmena, ale při přejíždění se zvedne.



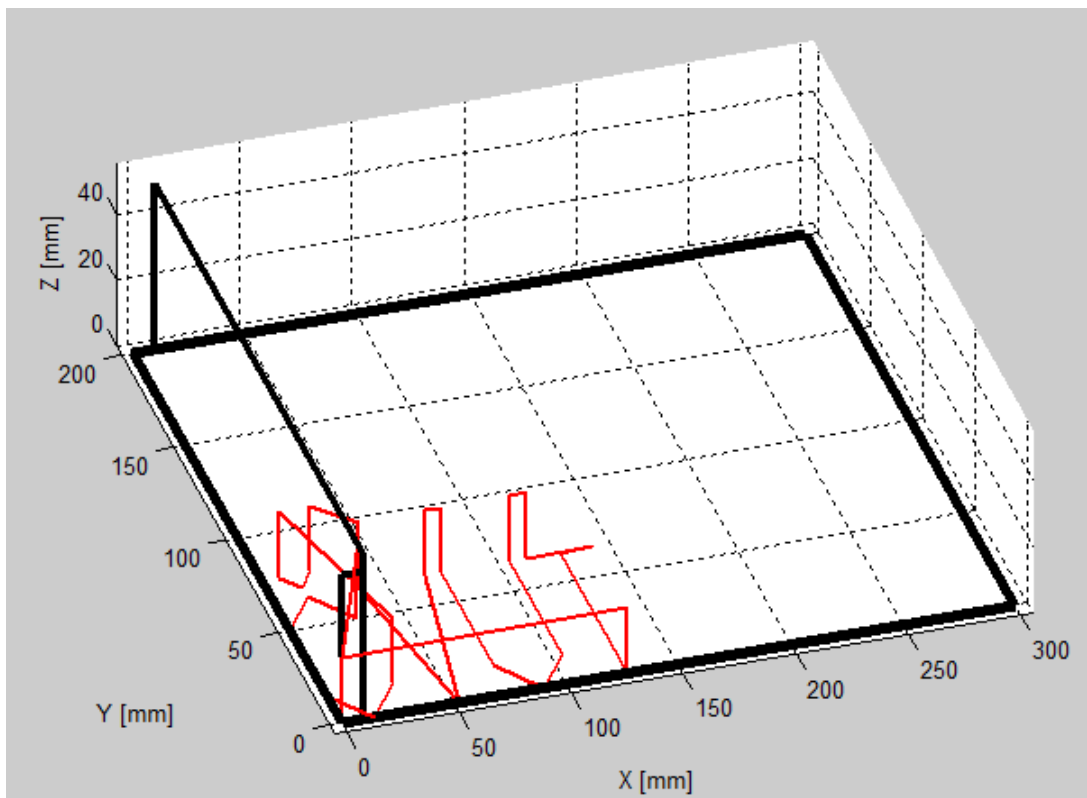
Obr. 39 Nápis ČVUT ze zkušební úlohy

Průjezd dráhy bude definovanou rychlostí  $v = 6 \text{ mm/s}$ . Na některých úsecích je vidět, že je příliš krátká dráha na to, aby bylo definované rychlosti dosaženo.



Obr. 40 Rychlostní profil podél dráhy ze zkušební úlohy

Kód souřadnic nápisu je umístěn v příloze. V grafickém znázornění stroje (Obr. 33) je vykreslen průběh dráhy nápisu ČVUT, aby bylo vidět, že program počítá správné dráhy.



Obr. 41 Zkušební trajektorie vykreslená v pracovním prostoru

## 4 ZÁVĚR

Vytvořený interpolační program spolehlivě pracuje a provádí výpočty v simulačním prostředí Simulink, kde si lze vykreslit časové průběhy libovolných veličin z kterékoli části programu. Z těchto pozorování a vykreslování aktuální polohy v prostoru je možné usuzovat, že vytvořený algoritmus pracuje podle stanovených požadavků. Problematická je však komunikace s reálným světem, kdy se objevují problémy, které simulace neodhalí. Největší překážkou bylo vyladění velikosti časového kroku programu a maximální rychlosti s ohledem na rozlišovací schopnost při daném kroku. Při špatných kombinacích krokové motory ztrácely krok nebo chaoticky měnily směr pohybu.

Jedním z možných budoucích vylepšení celkového chodu je použití snímačů polohy pro zajištění zpětné vazby. Ve vypracovaném řešení je řízení otevřenou smyčkou (ovládání), kde aktuální poloha je pouze teoreticky předpokládána a platí je v případě, kdy nedojde ke ztrátě kroku motoru. Program vykoná přesně daný počet kroků bez ohledu na to, zda byl krok reálně proveden. Pokud nějaký krok vypadne z důvodu přetížení nebo špatného vyladění parametrů, ztrácí se informace o skutečné poloze a program nedojede do správného místa. Senzor polohy by mohl být inkrementální na hřídeli motoru. Plně dostačující by bylo, kdyby rozlišení senzoru bylo stejné veliké jako velikost jednoho kroku motoru, aby byla zajištěna zpětná informace o skutečné poloze a o tom, zda byl požadovaný krok vykonán. Pro generování pulzů by byla porovnávána skutečná poloha ze senzoru a nikoliv jen teoretická (otevřená smyčka), takže počet kroků by byl variabilní, aby bylo požadované polohy skutečně dosaženo.

Pro zvýšení rychlosti výpočtu při nahrání programu do Arduina by bylo vhodné ho vytvořit jiným způsobem než pomocí Simulinku. Při kompilaci blokových schémat nejdříve do jazyka C a potom do strojového kódu vzniká velmi mnoho programových struktur, které jsou pro vlastní výpočet nepodstatné a zbytečně zatěžují procesor. Pokud by byl program napsán přímo v jazyce C nebo C++, obsahoval by o mnoho méně těchto zbytečných částí. Výpočet by se mohl částečně zrychlit a tím i zrychlit pulzování na motory, které by se mohly točit rychleji. Nevýhodou by byla nižší přehlednost v kódu. Předností Simulinku je právě přehlednost algoritmu, možnost vykreslování průběhů veličin a s tím související snazší detekce programových chyb a editace.

Dalším způsobem jak zvýšit rychlost výpočtu programu je použití lepšího hardwaru, jehož procesor má vyšší taktovací frekvenci. Použité testovací Arduino Mega má procesor s taktovací frekvencí 16 MHz, což je dle zkoušení nedostačující pro rychlé výpočty

předcházející generování pulzů pro krokový motor v požadovaném časovém kroku. Pokud by uživatel chtěl zůstat u desek typu Arduino, bylo by vhodnější použít jejich vyšší řadu. Jako příklad lze uvést Arduino Due s frekvencí procesoru 84 MHz. Jinou možností by také bylo rozdělit výpočet mezi více řídicích Arduino desek, kdy jedna by prováděla jen malé výpočty a na základě informace z druhé desky by generovala pulzy. Sloužila by jako ovladač motorů. Naopak druhá deska by prováděla většinu náročných výpočtů. Při tomto řešení by ovšem již bylo nutné použít výše zmiňovaný senzor polohy a bylo by nutné vyřešit vzájemnou komunikaci mezi deskami. K provádění výpočtů by bylo možné použít i počítač, ale opět by musela být vyřešena komunikace s Arduinem.

To byly návrhy na vylepšení současného řešení z hlediska rychlosti a přesnosti výpočtu. Některé z těchto problémů je možno obejít použitím jiného typu motoru než je krokový. Na krokový motor je potřeba vysoká frekvence pulzů pro dosažení vyšší rychlosti otáčení a navíc při těchto rychlostech dochází k prudkému poklesu krouticího momentu. Musela by ale být změněna celá část programu generující signály na motor, aby vytvářela správný typ komunikace s jiným typem motoru.

Interpolační algoritmus je vytvořený pro interpolování lineárních drah mezi soustavou bodů. Dalším rozšířením programu by mohlo být přidání dalších, běžně rozšířených, metod interpolace, mezi které patří kruhová interpolace a složitější interpolace splinem. Všechny další způsoby však vycházejí z přímkové, která tvoří jejich základ. Vhodné by bylo také doplnit přepínání mezi programovým a manuálním ovládáním motorů pomocí např. klávesnice.

Přes určité nedostatky vytvořený program v základní verzi plně funguje a výpočet probíhá dle požadavků. V celkovém řídicím systému obráběcí CNC frézky by mohl tvořit plnohodnotnou algoritmus interpolační části systému pro přímkové trajektorie.

## 5 SEZNAM POUŽITÝCH SYMBOLŮ

Skalární proměnné

$a$	zrychlení pohybu	[mm/s <sup>2</sup> ]
$a_0$	počáteční zrychlení pohybu	[mm/s <sup>2</sup> ]
$a_i$	zrychlení pohybu v kroku $i$	[mm/s <sup>2</sup> ]
$a_1, a_a$	zrychlení pohybu	[mm/s <sup>2</sup> ]
$a_3, a_d$	zpomalení pohybu	[mm/s <sup>2</sup> ]
$C_1, C_2$	konstanty při řešení diferenciální rovnice	
$D_x, D_y, D_z$	složky vektoru $\underline{D}$	
$e$	regulační odchylka	[mm]
$E$	odhad velikosti chyby numerické integrace	
$F$	hodnota primitivní funkce	
$f_{max}$	maximální frekvence pulzů	[Hz]
$h$	velikost kroku	[mm]
$i$	časově proměnný proud	[A]
$I$	napájecí proud	[A]
$j$	přičítací proměnná pro definování prvního kroku	
$J$	jerk	[mm/s <sup>3</sup> ]
$J_i$	jerk v kroku $i$	[mm/s <sup>3</sup> ]
$k$	směrnice přímky	
$k_{ref}$	referenční hodnota posledního řádku matice $A$	
$L$	indukčnost vinutí motoru	[H]
$K_m$	momentová konstanta motoru	[Nm/A]
$K_R$	počet kroků motoru na otáčku	
$m$	počet kroků	
$m_S$	souřadnice středu kružnice na ose $x$	[mm]
$M$	moment motoru	[Nm]
$n$	přičítací proměnná	
$n_S$	souřadnice středu kružnice na ose $y$	[mm]
$null$	nulovací signál při dosažení polohy	
$p$	parametr přímky	
$P$	stoupání pohybového šroubu	[mm/ot]
$P_i$	plocha pod křivkou	[mm <sup>2</sup> ]
$q$	konstanta udávající polohu přímky	
$r$	poloměr kružnice	[mm]
$R$	odpor vinutí motoru	[Ω]
$s$	přirozená souřadnice polohy	[mm]
$s_0$	počáteční přirozená souřadnice polohy	[mm]
$s_{celk}$	celková dráha mezi dvěma body	[mm]
$s_{ch}$	velikost chyby polohy při numerické metodě	[mm]
$s_i$	přirozená souřadnice polohy v kroku $i$	[mm]



$s_1, s_a$	dráha zrychlení pohybu	[mm]
$s_2$	dráha pohybu konstantní rychlostí	[mm]
$s_3, s_d$	dráha zpomalení pohybu	[mm]
$t$	čas	[s]
$t_{obr}$	proměnná odpočítávání času pohybu	[s]
$t_{sim}$	simulační čas	[s]
$t_1$	čas zrychlení pohybu	[s]
$t_2$	čas pohybu konstantní rychlostí	[s]
$t_3$	čas zpomalení pohybu	[s]
$\Delta t$	časový krok	[s]
$U$	napětí na motoru	[V]
$v$	rychlost pohybu požadovaná	[mm/s]
$v_{ch}$	velikost chyby rychlosti při numerické metodě	[mm/s]
$v_i$	rychlost pohybu v kroku $i$	[mm/s]
$v_p$	požadovaná rychlost pohybu	[mm/s]
$v_{poč}, v_0$	počáteční rychlost pohybu	[mm/s]
$v_{konc}, v_k$	koncová rychlost pohybu	[mm/s]
$v_{dos}, v_{traj}$	dosažená rychlost při příliš krátké dráze na dosažení $v$	[mm/s]
$\Delta v_a$	max. dosažitelná změna rychlosti při zrychlování	[mm/s]
$\Delta v_d$	max. dosažitelná změna rychlosti při zpomalování	[mm/s]
$\Delta x, \Delta y, \Delta z$	složky vektoru $\underline{r}_2$	[mm]
$x_{act}$	aktuální poloha na ose $x$	[mm]
$\Delta x_{pulse}$	vzdálenost vykonaná na jeden krok	[mm]
$\alpha$	úhel mezi $\underline{r}_2$ a rovinou $xz$	[rad]
$\beta$	úhel mezi $\underline{r}_2$ a rovinou $xy$	[rad]
$\varphi$	úhel mezi vektory $\underline{u}$ a $\underline{w}$	[rad]
$\varphi_0$	počáteční úhel oblouku	[rad]
$\psi$	úhel sklonu přímky	[rad]
$\tau$	časová konstanta přechodového jevu	[s]
$\omega$	úhlová rychlost	[rad/s]

#### Vektorové proměnné

$\underline{A}$	vstupní matice souřadnic
$\underline{B}$	výstupní matice souřadnic z trajektorového plánovače
$\underline{C}$	vektor souřadnic bodu, kterým prochází přímka při lineární interpolaci
$\underline{D}$	vektor směrů rotace na jednotlivé motory
$\underline{PULSE}$	vektor pulzů na jednotlivé motory
$\underline{r}_{O1}$	polohový vektor souřadného systému 1
$\underline{r}_{O2}$	polohový vektor souřadného systému 2
$\underline{r}_0$	polohový vektor počátečního bodu

$\underline{r}_1$	polohový vektor koncového bodu
$\underline{r}_2$	rozdíl polohových vektorů
$\underline{S}$	vektor souřadnic středu
$\underline{s}$	vektor složek přirozených souřadnic do směrů x, y, z
$\underline{T}_{12}$	transformační matice mezi systémy 1 a 2
$\vec{u}$	vektor směrnice přímky
$\underline{u}$	směrový vektor první dvojice souřadnic
$\underline{v}$	vektor rychlosti posuvu nástroje
$\underline{w}$	směrový vektor druhé dvojice souřadnic
$\underline{X}$	vektor souřadnic aktuálního bodu v závislosti na parametru
$\underline{X}_{act}$	vektor aktuální polohy v prostoru
$\underline{X}_{last}$	vektor polohy v prostoru v předchozím kroku
$\Delta\underline{X}_{pulse}$	vektor přírůstků na jeden krok motoru
$\underline{X}_0$	vektor souřadnic počáteční bod dráhy
$\Delta\underline{X}$	vektor rozdílu souřadnic dvou bodů

## 6 SEZNAM OBRÁZKŮ

Obr. 1 Konvenční univerzální frézka z roku 1862 (Tumlik, 2013).....	8
Obr. 2 Vertikální obráběcí centrum MCV 1220FA (Ing. HALA, 2010) .....	9
Obr. 3 Blokové schéma regulátoru (Prof. Ing. HOFREITER, 2014).....	12
Obr. 4 Grafické znázornění omezené přesnosti strojů vlivem inkrementů (ROSSMANN, 2008).....	13
Obr. 5 Interpolace dvou bodů.....	13
Obr. 6 Interpolace obecné funkce pomocí úseček.....	14
Obr. 7 Grafické zobrazení kinematických veličin při lichoběžníkovém profilu rychlosti.....	18
Obr. 8 Průběh rychlosti a polohy při krátké dráze .....	20
Obr. 9 Grafické vykreslení profilu rychlosti po S-křivce.....	22
Obr. 10 Používané souřadné systémy, vlevo kartézský souřadný systém, vpravo cylindrický souřadný systém .....	23
Obr. 11 Značka nulového bodu stroje M (Ing. KELLER, 2005) .....	24
Obr. 12 Značka referenčního bodu R (Ing. KELLER, 2005).....	24
Obr. 13 Značka nulového bodu obrobku (Ing. KELLER, 2005) .....	24
Obr. 14 Vyznačení vztažných bodů na frézce (Ing. KELLER, 2005) .....	25
Obr. 15 Náhradní schéma modelující stator krokového motoru (SLINTÁK, 2013) .....	28
Obr. 16 Grafické znázornění přechodového jevu.....	30
Obr. 17 Průběh přiváděného proudu na cívky označené A a B (SEMRÁD, 2012).....	31
Obr. 18 Průběh přiváděného proudu při použití 4 mikrokroků (Ing. RYDLO, 2000).....	31
Obr. 19 Momentové charakteristiky krokových motorů řady SX-17, výrobce Microcon, s.r.o. (Microcon).....	33
Obr. 20 Momentová charakteristika stejnosměrného servomotoru s permanentními magnety. Hodnoty s indexem S jsou záběrové a hodnoty s indexem 0 jsou při chodu naprázdno (ROHAL, 2007), upraveno autorem .....	34
Obr. 21 Znázornění vstupů a výstupů z bloku interpolátoru.....	35
Obr. 22 Blokové schéma navrhovaného programu .....	36
Obr. 23 Rozhodovací algoritmus pro generování průběhu zrychlení .....	38
Obr. 24 Vyznačení přepínacích míst zrychlení .....	38
Obr. 25 Znázornění polohových vektorů .....	40
Obr. 26 Pomocný obrázek pro určení úhlů $\alpha$ a $\beta$ .....	40
Obr. 27 Rozklad přirozené souřadnice do složek na jednotlivých osách.....	41
Obr. 28 Regulační obvod (Prof. Ing. HOFREITER, 2014).....	42
Obr. 29 Požadovaná poloha a generovaná aktuální poloha při regulační odchylce $e$ .....	42
Obr. 30 Požadovaná poloha a generovaná aktuální poloha při regulační odchylce $2e$ .....	43
Obr. 31 Simulinkové blokové schéma pro generování pulzů .....	43
Obr. 32 Grafické znázornění výpočtové polohy v prostoru .....	45
Obr. 33 Simulinkové blokové schéma pro výpočet aktuální polohy v prostoru .....	46
Obr. 34 Simulinkové blokové schéma generátoru souřadnic.....	47
Obr. 35 Grafické vykreslení křivek odvozených závislostí .....	49
Obr. 36 Ilustrace úpravy počáteční rychlosti .....	50

Obr. 37 Průběhy kinematických veličin přes všechny souřadnice .....	51
Obr. 38 Nápis ČVUT ze zkušební úlohy.....	52
Obr. 39 Rychlostní profil podél dráhy ze zkušební úlohy.....	52
Obr. 40 Zkušební trajektorie vykreslená v pracovním prostoru.....	53

## 7 POUŽITÁ LITERATURA

**Doc. Ing. BĚLOHOUBEK, Pavel, CSc.** *Elektrické servopohony, jejich řízení a aplikace I.* Brno : Inženýrské centrum Brno, 1998. ISBN 80-902175-4-0.

**EncyklopedieCoJeCo.** Siemens. *CoJeCo.cz.* [Online] 19. 10 2005. [Citace: 24. 10 2015.] Dostupné z:  
[http://www.cojeco.cz/index.php?detail=1&id\\_desc=86663&s\\_lang=2&title=Siemens.](http://www.cojeco.cz/index.php?detail=1&id_desc=86663&s_lang=2&title=Siemens)

**Ing. HALA, Marek, Ing. DEDERLE, Tomáš.** Zlínští strojaři představí obráběcí centrum. *MM Průmyslové spektrum.* [Online] 1. 9 2010. [Citace: 24. 10 2015.] Dostupné z:  
[http://www.mmspektrum.com/clanek/zlinsti-strojari-predstavi-obrabeci-centrum.html.](http://www.mmspektrum.com/clanek/zlinsti-strojari-predstavi-obrabeci-centrum.html)

**Ing. KELLER, Petr, Ph.D.** Programování a řízení CNC strojů. *Technická univerzita v Liberci, fakulta strojí.* [Online] 2005. [Citace: 4. 2 2016.] Dostupné z:  
[http://www.kvs.tul.cz/download/cnc\\_cadcam/pnc\\_2.pdf.](http://www.kvs.tul.cz/download/cnc_cadcam/pnc_2.pdf)

**Ing. POLZER, Aleš , Ph.D.** Akademie CNC obrábění (1). *Technický týdeník.* [Online] 15. 1 2009. [Citace: 10. 24 2015.] Dostupné z:  
[http://www.technickytydenik.cz/rubriky/serialy/akademie-cnc-obrabeni/akademie-cnc-obrabeni-1\\_8536.html.](http://www.technickytydenik.cz/rubriky/serialy/akademie-cnc-obrabeni/akademie-cnc-obrabeni-1_8536.html)

**Ing. POLZER, Aleš, Ph.D.** Akademie CNC obrábění (3). *Technický týdeník.* [Online] 13. 2 2009. [Citace: 28. 11 2015.] Dostupné z:  
[http://www.technickytydenik.cz/rubriky/serialy/akademie-cnc-obrabeni/akademie-cnc-obrabeni-3\\_8538.html.](http://www.technickytydenik.cz/rubriky/serialy/akademie-cnc-obrabeni/akademie-cnc-obrabeni-3_8538.html)

**Ing. RYDLO, Pavel.** KROKOVÉ MOTORY A JEJICH ŘÍZENÍ. *Fakulta mechatroniky a mezioborových inženýrských studií, TECHNICKÁ UNIVERZITA V LIBERCI.* [Online] 2000. [Citace: 15. 2 2016.] Dostupné z:  
[https://www.pslib.cz/pe/skola/studijni\\_materialy/motory/krok\\_motor/krok\\_ucebnice.pdf.](https://www.pslib.cz/pe/skola/studijni_materialy/motory/krok_motor/krok_ucebnice.pdf)

**Ing. ŠTAJNORCH, Lubomír.** Technologie II. [přednáška]. Praha : ČVUT, 16. 11 2015.

**KRATOCHVÍL, Ctirad, a další.** *Mechatronické pohonové soustavy.* Brno : VUT v Brně, ÚMTMB- Centrum mechatroniky, FSI, 2008. ISBN 987-80-214-3790-6.

**LEWIN, Chuck.** Mathematics of Motion Control Profiles. *PMDcorp.* [Online] 2007. [Citace: 27. 1 2016.] Dostupné z:  
[http://www.pmdcorp.com/downloads/Mathematics\\_of\\_Motion\\_Control\\_Profiles.pdf.](http://www.pmdcorp.com/downloads/Mathematics_of_Motion_Control_Profiles.pdf)

**MAIXNER, Ladislav a kolektiv.** *Mechatronika.* Brno : nakladatelství Computer Press, 2006. ISBN 80-251-1299-3.

**Mgr. RŮŽIČKOVÁ, Irena, RNDr. HAVLIČKA, Rudolf, CSc.** Numerické metody. [Online] [Citace: 29. 4 2016.] Dostupné z:  
[http://physics.ujep.cz/~jskvor/NME/DalsiSkripta/Numerika.pdf.](http://physics.ujep.cz/~jskvor/NME/DalsiSkripta/Numerika.pdf)

**Microcon.** Momentové charakteristiky řady SX16, SX17, SX23. [Online] [Citace: 12. 3 2016.] Dostupné z: <http://www.microcon.cz/pdf2015/22-23.pdf>.

**Motion Designs Inc.** Desing trends. *Motion Designs*. [Online] 8 2010. [Citace: 27. 1 2016.] Dostupné z: [http://www.motion-designs.com/images/DTrends\\_Aug\\_2010.pdf](http://www.motion-designs.com/images/DTrends_Aug_2010.pdf).

**NEUSTUPA, Jiří.** *Matematika I*. Praha : ČVUT, 2008. 978-80-01-04111-6.

**NGUYEN, Kim Doang, NG, Teck-Chew, CHEN, I-Ming.** On Algorithms for Planning S-curve. [Online] 2008. [Citace: 27. 1 2016.] Dostupné z: <http://cdn.intechopen.com/pdfs-wm/4267.pdf>. ISSN 1729-8806, pp. 99-106.

**ONWUBOLU, C. Godfrey.** *Mechatronics principles and applications*. Norfolk : Biddles Ltd, King's Lynn, 2005. ISBN 0-7506-6379-0.

**Prof. Dr. Ing. SCHMID, Dietmar a kolektiv.** *Řízení a regulace pro strojírenství a mechatroniku*. Praha : SOBOTÁLES cz, 2005. ISBN 80-86706-10-9.

**Prof. Ing. HOFREITER, Milan, CSc.** *Základy automatického řízení*. Praha : Nakladatelství ČVUT, 2014. ISBN 978-80-01-05007-1.

**ROHAL, Milan.** Princip - Servo motor. *pohonnatechnika.cz*. [Online] 2007. [Citace: 12. 3 2016.] Dostupné z: <http://www.pohonnatechnika.cz/skola/motory/servo-motor>.

**ROSSMANN, Miloslav.** Programování CNC strojů. *COPTTEL-Mechtronika*. [Online] 2008. [Citace: 24. 10 2015.] Dostupné z: <http://coptel.coptkm.cz/reposit.php?action=0&id=22926&instance=2>.

**SEMRÁD, Michal.** *Řízení krokových motorků*. [online] Brno : Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 47s. Vedoucí bakalářské práce byl Ing. Lukáš Kopečný, Ph.D..

**SLINTÁK, Vlastimil.** Napětí krokových motorů. *μArt.cz*. [Online] 1. 6 2013. [Citace: 8. 2 2016.] Dostupné z: <http://uart.cz/1268/napeti-krokovych-motoru/>.

**SOŠ Bíteš.** SOŠ Jana Tiraye Velká Bíteš. *Obecný úvod do problematiky CNC programování*. [Online] 1. 1 2001. [Citace: 24. 10 2015.] Dostupné z: [http://www.sosbites.cz/images/stories/VUKOV\\_TEXT\\_-\\_1.ST.pdf](http://www.sosbites.cz/images/stories/VUKOV_TEXT_-_1.ST.pdf).

**Studijní materiály.** Numerické výpočty. *VUT v Brně, Fakulta stavební*. [Online] 2014. [Citace: 28. 1 2016.] Dostupné z: <http://www.fce.vutbr.cz/studium/materialy/BU01/mac/numerickeVypocty.pdf>.

**ŠTULPA, Miroslav.** *CNC obráběcí stroje a jejich programování*. Praha : Nakladatelství BEN, 2006. ISBN 80-7300-207-8.

**Tumlik.** Z historie vývoje fréz, frézovacích strojů a frézování. *TUMLIKOVO.CZ*. [Online] 23. 11 2013. [Citace: 24. 10 2015.] Dostupné z: <http://www.tumlikovo.cz/z-historie-vyvoje-frez-frezovacich-stroju-a-frezovani/>.

**VOJÁČEK, Antonín.** El. motory a jejich řízení- základní přehled- 1. díl. *automatizace.hw*. [Online] 7. 7 2014. [Citace: 8. 2 2016.] Dostupné z: <http://automatizace.hw.cz/el-pohony-mereni-a-regulace/el-motory-a-jejich-rizeni-zakladni-prehled.html>.

**ZEITHAMMER, Karel.** *Vývoj techniky*. Praha : Vydavatelství ČVUT, 2000. 80-01-01725-7.

## **8 SEZNAM PŘÍLOH**

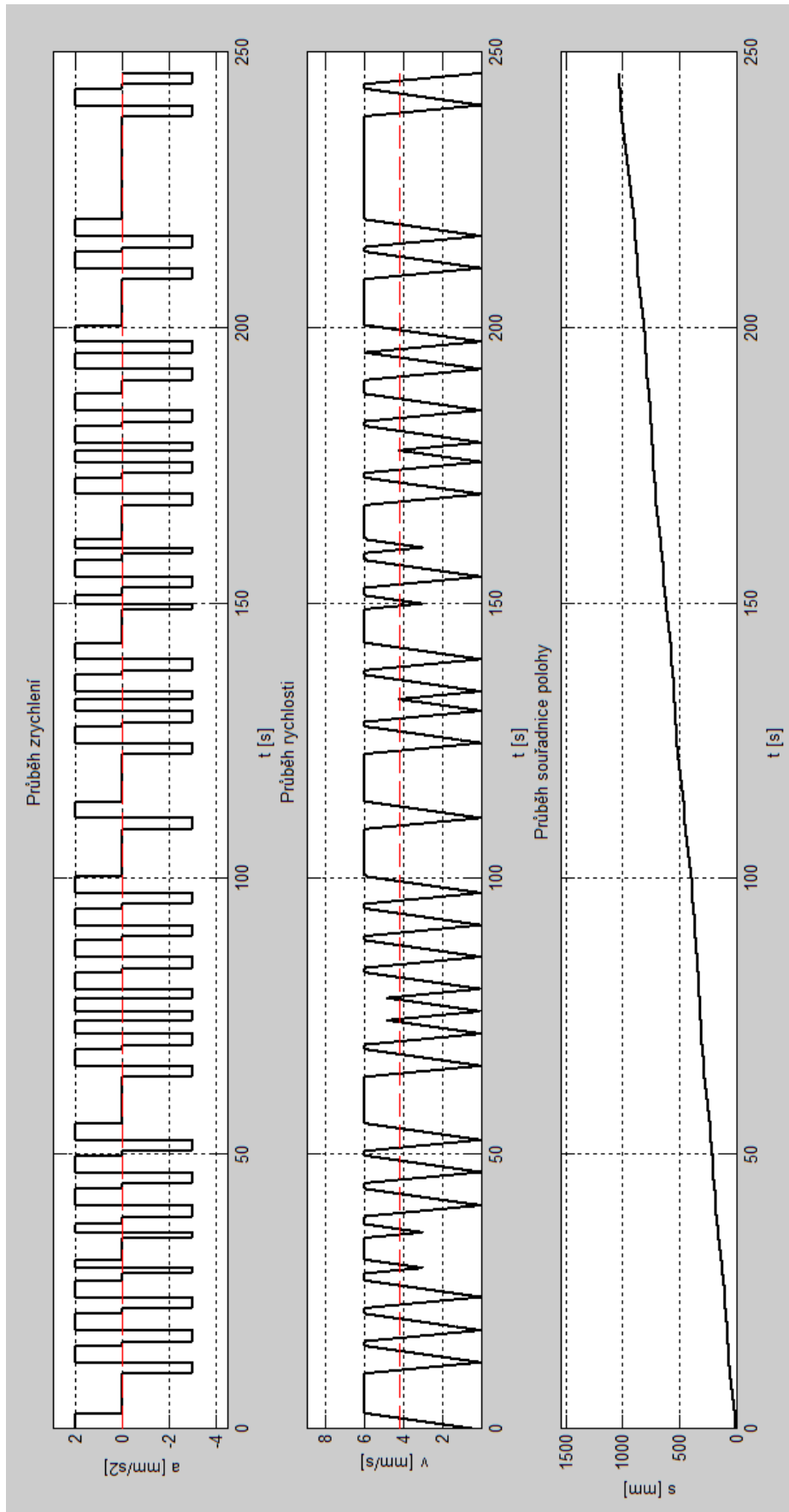
- I. Souřadnice dráhy zkušební úlohy
- II. Průběhy kinematických veličin podél zkušební dráhy
- III. Srovnání celkových časů zkušební dráhy v závislosti na zvolené funkci výpočtu koncových rychlostí
- IV. Vytvořený program na CD



I. Souřadnice dráhy zkušební úlohy

Bod	X [mm]	Y [mm]	Z [mm]	v [mm/s]	Poznámka
1	0	0	30	6	Počátek
2	30	50	30	6	Písmeno C
3	30	50	0	6	
4	15	65	0	6	
5	0	50	0	6	
6	0	15	0	6	
7	15	0	0	6	
8	30	15	0	6	
9	30	15	30	6	
10	8	77	30	6	
11	8	77	0	6	
12	15	70	0	6	
13	22	77	0	6	
14	22	77	30	6	
15	37,5	65	30	6	Písmeno V
16	37,5	65	0	6	
17	52,5	0	0	6	
18	67,5	65	0	6	
19	67,5	65	30	6	
20	75	65	30	6	Písmeno U
21	75	65	0	6	
22	75	15	0	6	
23	90	0	0	6	
24	105	15	0	6	
25	105	65	0	6	
26	105	65	30	6	
27	112,5	65	30	6	Písmeno T
28	112,5	65	0	6	
29	142,5	65	0	6	
30	127,5	65	0	6	
31	127,5	0	0	6	
32	127,5	0	30	6	
33	0	0	30	6	Návrat do počátku

## II. Průběhy kinematických veličin podél zkušební dráhy



III. Srovnání celkových časů zkušební dráhy v závislosti na zvolené funkci výpočtu koncových rychlostí

Typ křivky $v_k = f(\varphi)$	Celkový čas [s]	Časový rozdíl [s]	Procentuální rozdíl
Přímka	246,36	2,22	0,89%
Elipsa	244,04	4,54	1,83%
Parabola	244,49	4,09	1,65%
Exponenciála	248,58	0,00	0,00%