

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra kybernetiky

BAKALÁŘSKÁ PRÁCE

Plzeň, 2022

Jan Kasík

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd
Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Jan KASÍK**
Osobní číslo: **A19B0296P**
Studijní program: **B0714A150005 Kybernetika a řídicí technika**
Specializace: **Automatické řízení a robotika**
Téma práce: **Plánování trajektorie v systémech řízení pohybu**
Zadávající katedra: **Katedra kybernetiky**

Zásady pro vypracování

1. Seznamte se s metodami plánování trajektorie pohybu v mechatronických systémech a technikami matematické reprezentace polohy a orientace.
2. Navrhněte vhodné algoritmy umožňující interpolaci uživatelských dat v podobě sekvence lineárních a kruhových úseků do formy fyzikálně realizovatelných hladkých pohybů.
3. Otestujte navržené algoritmy na simulačních modelech a v případě možnosti na fyzikálním systému.

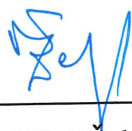
Rozsah bakalářské práce: **30 – 40 stránek A4**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

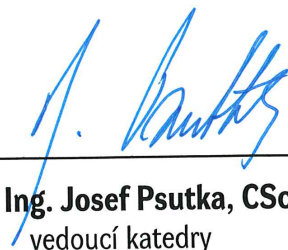
- GOUBEJ, M. a MELICHAR, J. Lineární systémy 1. Učební text, Západočeská univerzita, Fakulta aplikovaných věd, 2017.
GOUBEJ, M. a MELICHAR, J. Lineární systémy 2. Učební text, Západočeská univerzita, Fakulta aplikovaných věd, 2019.
GOUBEJ, M., ŠVEJDA, M., SCHLEGEL, M. Úvod do mechatroniky, robotiky a systémů řízení pohybu. Skriptum pro studenty doktorských programů v oboru automatické řízení, Západočeská univerzita, Fakulta aplikovaných věd, 2012.
ŠVEJDA, M. Úvod do robotiky a mechatroniky [přednášky]. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, říjen – prosinec 2021.

Vedoucí bakalářské práce: **Ing. Martin Goubelj, Ph.D.**
Katedra kybernetiky

Datum zadání bakalářské práce: **15. října 2021**
Termín odevzdání bakalářské práce: **23. května 2022**



Doc. Ing. Miloš Železný, Ph.D.
děkan



Prof. Ing. Josef Psutka, CSc.
vedoucí katedry

PROHLÁŠENÍ

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 16. května 2022

.....

Jan Kasík

PODĚKOVÁNÍ

Tímto bych chtěl poděkovat vedoucímu bakalářské práce, Ing. Martinu Goubajovi, Ph. D., za odborné vedení, věnovaný čas, cenné rady, podněty a připomínky při zpracování mé práce.

Abstrakt

Cílem této bakalářské práce je představit problematiku plánování trajektorie v systémech řízení pohybu. V úvodu se práce zabývá rozdělením obecného mechatronického systému na tři základní vrstvy. Dále je vysvětlena kinematika tělesa se zaměřením na druhy popisu jeho orientace. Poté je popsán pojem interpolace dat a uvedeno několik základních v praxi používaných typů interpolace včetně problémů s nimi spojených. Spolu s touto prací byl také navržen program, který plní funkci generátoru trajektorie.

Klíčová slova : orientace tělesa, rotace, pohyb, interpolace, trajektorie, generátor

Abstract The aim of this bachelor thesis is to introduce the problem of trajectory planning in motion control systems. The thesis starts by dividing a general mechatronic system into three basic layers. Then, the kinematics of the body is explained, focusing on the types of descriptions of its orientation. The concept of data interpolation is then described and several basic types of interpolation used in practice are introduced, including the problems associated with them. In parallel with this work, a program has been designed to serve as a trajectory generator.

Key words : body orientation, rotation, motion, interpolation, trajectory, generator

Obsah

Seznam obrázků	8
1 Úvod	9
2 Mechatronika	10
2.1 Řídicí systém	10
2.1.1 Vrstva pro monitorování a diagnostiku	11
2.1.2 Generátor trajektorie	11
2.1.3 Regulátor pohybu	12
3 Kinematika tělesa	13
3.1 Popis orientace tělesa v prostoru	13
3.1.1 Eulerovy úhly	13
3.1.2 Matice rotace	14
3.1.3 Jednotkový kvaternion	17
3.2 Rozklad pohybu	18
3.3 Výpočet polohy a orientace tělesa	19
4 Plánování trajektorie pohybu	21
4.1 Souřadnicové systémy	21
4.2 Interpolace	22
4.2.1 Přímková interpolace	23
4.2.2 Kruhová interpolace	24
4.2.3 Hélická interpolace	24
4.2.4 Interpolace spline křivkami	25
5 Návrh generátoru trajektorie	29
5.1 Geometrická reprezentace trajektorie	29
5.2 Pohyb s konstantní rychlostí	30
5.3 Fyzikálně realizovatelný pohyb	31
5.3.1 Příklady vygenerování rychlostních profilů	33
5.4 Časový průběh pohybu	34
5.5 B-spline interpolace	37
5.5.1 Převod na fyzikální pohyb	39
5.6 Orientace pomocí jednotkového kvaternionu	41
5.7 Vyhodnocení a kompenzace chyby B-spline interpolací	42

6 Závěr	46
Použitá literatura	47

Seznam obrázků

1	Mechatronika jako věda	10
2	Eulerovy úhly	14
3	Příklad hélického pohybu mezi dvěma body	25
4	Příklad interpolace dat B-spline křivkou	28
5	Vygenerovaná trajektorie na základě vstupních dat	30
6	Tečná rychlost a zrychlení při požadavku na rovnoměrný pohyb s konstantní rychlostí	31
7	Lichoběžníkový rychlostní profil	33
8	Trojúhelníkový rychlostní profil	34
9	Časový průběh rychlosti a zrychlení v jednotlivých osách při interpolaci přímkami a kružnicemi	35
10	Časový průběh tečné rychlosti a tečného zrychlení	36
11	Nespojitost rychlosti	36
12	Příklad interpolace dat B-spline křivkou 5. stupně	38
13	Porovnání B-spline křivek 3. a 5. stupně při interpolaci shodných dat	38
14	Časový průběh rychlosti a zrychlení v jednotlivých osách při B-spline interpolaci	40
15	Časový průběh tečné rychlosti a zrychlení při B-spline interpolaci	41
16	Vývoj jednotlivých složek jednotkové vektoru po B-spline interpolaci orientace	42
17	Referenční trajektorie tvořená sekvencí přímek a kružnic	44
18	B-spline interpolace testovacích dat pro kompenzaci chyby interpolace	44
19	Výsledná trajektorie při využití první korekční funkce	45
20	Výsledná trajektorie při využití druhé korekční funkce	45

1 Úvod

Hlavním cílem této bakalářské práce bylo navrhnout program v prostředí MATLAB, který bude plnit úlohu generátoru trajektorie. Generátor trajektorie je jedna z vrstev každého systému řízení pohybu. Ať už se jedná o průmyslové robotické manipulátory či CNC (Computer Numeric Control) obráběcí stroje, je jednou z klíčových úloh těchto mechatronických systémů generování trajektorie na základě požadavků operátora. Kromě toho je také nutné, aby generátor respektoval fyzikální omezení stroje a realizovatelnost samotného generovaného pohybu.

V první části se práce zabývá úvodem do celé problematiky. Ve zkratce je představena mechatronika jako věda a také popsán obecný mechatronický systém, se zaměřením na vrstvu generátoru trajektorie a regulace. Druhá kapitola je věnována samotnému pohybu se zaměřením na popis orientace tělesa v prostoru. Budou představeny vybrané způsoby reprezentace včetně jejich nedostatků. Dále je pak rozebrán rozklad pohybu a jeho následné využití při výpočtech polohy tělesa. Závěr teoretické části práce je věnován základním myšlenkám plánování trajektorie pohybu. Budou popsány v praxi používané typy interpolace od nejjednodušší lineární interpolace až po komplexní NURBS křivky.

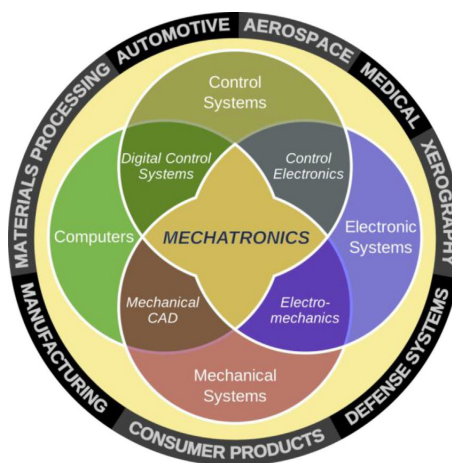
Součástí práce je také praktická část, která se zaměřuje na popis návrhu generátoru trajektorie v celém rozsahu. Nejdříve budou naprogramované jednoduché lineární a kruhové interpolace, a dále pak metoda, která umožní navázání jednotlivých segmentů. Poté bude vytvořena část generátoru, která převede geometrickou reprezentaci trajektorie na fyzikální pohyb v reálném čase. Nejprve bude navrhnout algoritmus pro konstantní rychlost na celé dráze, který bude následně rozšířen na realizovatelný pohyb "z klidu do klidu". Dále bude navržena metoda pro interpolaci dat pomocí B-spline křivky, pomocí které bude interpolována jak poloha, tak i orientace. Posledním úkolem bude navrhnout vhodný algoritmus, jehož cílem bude vyhodnotit a kompenzovat chybu, které se dopustí B-spline interpolace oproti původní trajektorii tvořené přímkami a kružnicemi.

Zvláště praktická část je doplněna o grafickou vizualizaci.

2 Mechatronika

Mechatronika je komplexní vědní disciplína spojující obdobné obory jako jsou mechanika, elektrotechnika, softwarové inženýrství, automatické řízení a další. Pojem "Mechatronika" je užíván již od roku 1969, kdy jej zavedla japonská firma Yaskawa zaměřující se na výrobu motorů, servopohonů a robotů [13].

S rozvojem výpočetní techniky bylo možné standardní mechanické systémy vylepšit o další funkce, které umožní využít potenciálu kombinace zpracování dat z různých vestavěných senzorů spolu s vhodně umístěnými aktuátory k efektivnímu řízení daného systému [4]. Za vhodný příklad lze uvažovat aktivní řízení podvozku u automobilu, kdy původní mechanický systém "pružina a tlumič" bude rozšířen o aktuátor a senzory snímající polohou a rychlost kola automobilu vůči podvozku. S pomocí implementace vhodného řídicího systému je možné vylepšit kvalitu tlumení a zajistit tak komfortnější jízdu cestujících.



Obrázek 1: Mechatronika jako věda [8]

2.1 Řídicí systém

Jak již bylo řečeno v předchozí kapitole, mechatronický systém je vybaven řídicím systémem, který zajišťuje rozšířené funkce. Obecně lze každý takový řídicí systém rozdělit z hlediska funkcionality na tři hlavní části: vrstva pro monitorování a diagnostiku, generátor trajektorie a regulátor pohybu [4].

2.1.1 Vrstva pro monitorování a diagnostiku

Tato vrstva slouží, jak již název napovídá, k monitorování zbylých dvou vrstev a pochodů v nich probíhajících. Klíčovou funkcí je pak detekce havarijních stavů a chyb v systému. Může se jednat například o poškození nebo výpadek senzoru, ztrátu energie a další. Sofistikovanější systémy mají v současné době i software, který umožňuje těmto stavům předcházet a snížit následná poškození v co největší míře [4].

2.1.2 Generátor trajektorie

Vyšší vrstva systému pracuje interně ve dvou fázích. V první fázi dojde k tzv. interpolaci dat, neboli ke geometrickému propojení operátorem zadaných bodů požadovanou křivkou. V druhé fázi dojde k převodu této geometrické reprezentace na fyzikální pohyb v reálném čase, který je poté předán do nižší vrstvy, tj. do regulátorů. V praxi však samozřejmě existují mnohá omezení, se kterými musí generátor operovat [4]. Typickým příkladem mohou být překážky v prostoru. Přirozeným požadavkem je pak navrhnout dráhu tak, aby nedošlo ke kolizi stroje s překážkou. V těchto úlohách je následně potřeba využít metody umělé inteligence a strojového učení z důvodu obtížného nalezení analytického řešení [4].

Při generování fyzikálního pohybu je nutné dále počítat s fyzickými omezeními stroje, respektive aktuátorů. V reálném prostředí samozřejmě není možné vykonat nekonečně rychlý pohyb. Každý aktuátor má svůj maximální výkon, který spolu s fyzikálními parametry stroje omezuje maximální zrychlení, kterého je možné dosáhnout. Je proto nutné omezit akční zásahy a docílit tak realizovatelného pohybu. Souvislostí jsou také vibrace stroje. Pokud by stroj zrychloval/zpomaloval příliš prudce, mohlo by dojít k nežádoucím vibracím stroje, které by mohly způsobit mimo jiné jeho poškození.

Z důvodu současných požadavků v průmyslu je kladen velký důraz na to, aby vše proběhlo v co nejkratším čase. To znamená řešit problém časové optimalizace. Je tedy potřeba hledat takový časový průběh trajektorie, kdy je čas jeho vykonání co nejkratší, ovšem je nutné opět dbát na výše zmíněné požadavky, tj. realizovatelnost samotného pohybu a také omezení fluktuací stroje. Dalšími problémy, které musí generátor trajektorie řešit, jsou například zkoordinování více aktuátorů nebo zajištění pohybu ve více osách najednou [4].

2.1.3 Regulátor pohybu

Poslední (nižší) vrstvou je regulátor pohybu, jehož úkolem je sledování referenčního signálu v podobě časového průběhu převzatého z generátoru trajektorie, a regulování potřebných veličin tak, aby byl dosažen požadovaný pohyb. V průmyslové praxi se nejčastěji využívá zpětnovazební kaskádní struktura o třech regulátorech, jejíž hlavní výhoda je zajištění minimálního zpoždění mezi měřeným výstupem a poruchou. Vnitřní smyčka reguluje elektrický proud, prostřední reguluje otáčky motoru a vnější smyčka pak polohu aktuátoru [4].

Nejčastěji používané typy regulátorů jsou pro svou jednoduchost, efektivitu a zároveň univerzálnost tzv. PID (Proporcionálně Integračně Derivační) regulátory. P-složka představuje zesílení, díky kterému je možné zrychlit odezvu a potlačit nízkofrekvenční poruchy. I-složka zajišťuje přesnost regulace v podobě nulové regulační odchylky v ustáleném stavu. D-složka může při správném nastavení snížit kmitavost a zvýšit robustnost. Generované řízení má pak tvar [2,3]:

$$u(t) = K \left[e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt} \right],$$

kde $e(t)$ je regulační odchylka, K proporcionální zesílení, T_I integrační časová konstanta a T_D derivační časová konstanta.

Kromě sledování trajektorie musí vrstva regulace řešit i jiné problémy, například zajištění robustnosti vůči poruchám. Poruchy mohou nastat v podobě vnějších vlivů, případně změnou parametrů systému. Dále je nutné regulovat a případně eliminovat například nežádoucí vzniklé vibrace [4].

3 Kinematika tělesa

Dalším tématem, kterým se bude práce zabývat, je popis pohybu tělesa se zaměřením na jeho orientaci.

Spojitá křivka, která kopíruje dráhu pohybujícího se bodu v prostoru, se nazývá trajektorie. Na základě tvaru trajektorie je možné rozdělit pohyb na dvě základní skupiny [1]:

- a) Přímočarý pohyb (v případě, že dráhou je přímka).
- b) Křivočarý pohyb (pokud je trajektorií obecná křivka).

Dalším možným dělením by mohlo být rozdělení na pohyb posuvný (translační), otáčivý (rotační) a pohyb obecný. Při posuvném pohybu platí, že všechny body tělesa se pohybují po stejných vzájemně posunutých křivkách se stejnou rychlostí a zrychlením. Kinematika celého tělesa je tak určena kinematikou jednoho libovolného hmotného bodu. Rotační pohyb je pak, zjednodušeně řečeno, popsán pohybem libovolného referenčního bodu okolo středu (těžiště) tělesa. Výše zmíněné principy jsou platné jak v rovině, tak v prostoru.

3.1 Popis orientace tělesa v prostoru

V této podkapitole se práce bude zabývat popisem rotace, respektive orientace. Obecně se jedná o úlohu nalezení transformačního vztahu, který transformuje souřadnicový systém tělesa na souřadnicový systém pevného rámu. Typickým příkladem by mohlo být určení orientace koncového efektoru manipulátoru v souřadnicovém systému stroje.

Pro účely této práce budou uvedeny tři způsoby reprezentace: Eulerovy úhly, matice rotace a jednotkový kvaternion.

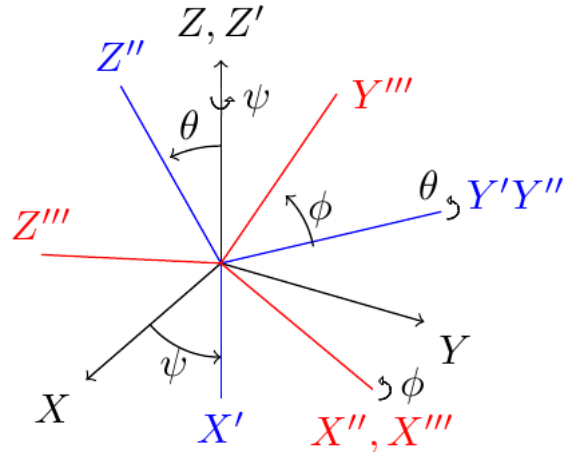
3.1.1 Eulerovy úhly

Jedním z možných popisů orientace jsou tzv. Eulerovy úhly. Grafické znázornění Eulerových úhlů je na obrázku č. 2.

Eulerovy úhly zakreslené na obrázku výše jsou definovány následovně [5]:

- 1) úhel precese ψ - natočení z pevného (původního) systému (X, Y, Z) do systému (X', Y', Z')
- 2) úhel nutace θ - natočení ze systému (X', Y', Z') do systému (X'', Y'', Z'')

- 3) úhel vlastní rotace ϕ - natočení ze systému (X'', Y'', Z'') do výsledného systému (X''', Y''', Z''')



Obrázek 2: Eulerovy úhly [12]

3.1.2 Matice rotace

Dalším v praxi používaným způsobem reprezentace orientace (rotace) je pomocí matice rotace. Pro účely následujících podkapitol necht' jsou definovány dva souřadnicové systémy. Kartézský souřadnicový systém (x_1, y_1, z_1) a souřadnicový systém tělesa (x_2, y_2, z_2) . Namísto použití Eulerových úhlů popsanych v sekci výše je zde orientace definována pomocí následující matice:

$$R_{12} = \begin{bmatrix} \cos(x_1, x_2) & \cos(y_1, x_2) & \cos(z_1, x_2) \\ \cos(x_1, y_2) & \cos(y_1, y_2) & \cos(z_1, y_2) \\ \cos(x_1, z_2) & \cos(y_1, z_2) & \cos(z_1, z_2) \end{bmatrix},$$

kde se na jednotlivých pozicích nacházejí cosiny úhlů, které spolu svírají dané osy obou soustav souřadnic.

Blokově se také matice zapisuje následovně [13]:

$$R_{12} = \begin{bmatrix} \mathbf{x}_1^2 & \mathbf{y}_1^2 & \mathbf{z}_1^2 \end{bmatrix}, \quad (3.1.1)$$

Z tohoto tvaru je patrné, že matice rotace je po sloupcích složena z vektorů definujících osy souřadnicového systému tělesa v prostoru kartézské soustavy souřadnic.

Alternativní variantou je sestavení matice pomocí jednotkových směrových vektorů jednotlivých os systémů, konkrétně jejich součinů. Matice poté vypadá takto [10]:

$$R_{12} = \begin{bmatrix} \vec{e}_{1x} \cdot \vec{e}_{2x} & \vec{e}_{1y} \cdot \vec{e}_{2x} & \vec{e}_{1z} \cdot \vec{e}_{2x} \\ \vec{e}_{1x} \cdot \vec{e}_{2y} & \vec{e}_{1y} \cdot \vec{e}_{2y} & \vec{e}_{1z} \cdot \vec{e}_{2y} \\ \vec{e}_{1x} \cdot \vec{e}_{2z} & \vec{e}_{1y} \cdot \vec{e}_{2z} & \vec{e}_{1z} \cdot \vec{e}_{2z} \end{bmatrix}, \quad (3.1.2)$$

kde se jednotlivé pozice skládají ze skalárního součinu dvou os systémů.

Obecně platí, že libovolnou rotaci je možné rozložit na rotace kolem jednotlivých os souřadnicového systému o úhly α, β, γ . Matice rotace těchto tzv. elementárních rotací plynou z obecného tvaru (3.1.1), jelikož vždy jedna osa zůstává neměnná. Matice rotace elementárních rotací jsou definovány následovně [13]:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Matice rotace libovolné orientace tělesa lze poté zapsat jako součin výše popsanych elementárních matic.

Při násobení matic je však nutné dbát na pořadí rotací, jelikož u maticového počtu platí, že násobení matic není komutativní. Z tohoto hlediska lze uvést dvě standardní formy rotace, a to fixní rotaci a postupnou rotaci.

Fixní rotace

Klíčovým aspektem této verze je, že všechny rotace probíhají kolem pevného systému (x_1, y_1, z_1) . Výsledná matice bude součinem matic R_x, R_y a R_z v postupném pořadí [13, 5]:

$$R_{RPY} = R_z R_y R_x$$

V praxi se této variantě také říká zkráceně RPY. Zkratka RPY pochází ze tří anglických slov: Roll, Pitch, Yaw; které lze z angličtiny přeložit jako klonění, klopení, bočení. Typickým příkladem využití varianty RPY je letectví, konkrétně popis orientace letadla ve vzdušném prostoru.

Postupná rotace

Tato modifikace se liší od předchozí varianty pouze tím, že rotace probíhají postupně podle aktuálního souřadnicového systému. Nejprve se tedy provede rotace kolem pevné osy x , dále kolem nově vzniklé osy y a v poslední řadě kolem nově vzniklé osy z . Matematický zápis je pak [13]:

$$R = R_x R_y R_z$$

Rotace kolem obecné osy

Další alternativní variantou je rotace kolem obecné osy o úhel θ . Zatím byly uvedeny pouze metody, které se zaměřují na rotace kolem jednotlivých os původního nebo nově vznikajícího souřadnicového systému. Při rotaci kolem obecné osy se využijí známé vztahy pro rotace kolem os x , y a z . Nejprve bude uveden vztah pro tuto operaci [13]:

$$R_o = R_z(\alpha)R_y(\beta)R_z(\theta)R_y(-\beta)R_z(-\alpha)$$

Celá operace je tedy složena z rotace kolem původní osy z o úhel α , poté rotací kolem původní osy y o úhel β . Tyto úhly svírají původní systém a obecná osa. Výsledkem těchto dvou operací je nová osa z , mířící do směru obecné osy. Poté je možné otočit nově vzniklý systém kolem obecné osy (pomocí matice rotace kolem osy z) o úhel θ . Nakonec se provedou první dvě operace se zápornými úhly, čímž se anulují první dvě rotace a zachová se tak původní systém¹.

Problém singularity

Doposud byla výše definována pouze dopředná transformace, tedy vztah, kterým lze z daných úhlů sestavit matici rotace, kdy tato transformace je vždy jednoznačná. V inverzní situaci však již transformace jednoznačná není a může dojít k tzv. singularitě.

Při úvaze rotace kolem všech tří os, bude mít matice rotace po pronásobení tvar [13]:

$$R(\alpha, \beta, \gamma) = \begin{bmatrix} c_\beta c_\gamma & -c_\beta s_\gamma & s_\beta \\ s_\alpha s_\beta c_\gamma + c_\alpha s_\gamma & -s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & -s_\alpha c_\beta \\ -c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma & c_\alpha s_\beta s_\gamma + s_\alpha c_\gamma & c_\alpha c_\beta \end{bmatrix},$$

kde s_α a c_α značí $\sin(\alpha)$ a $\cos(\alpha)$.

¹Vzhledem k vlastnosti lichosti funkce sinus: $\sin(x) = -\sin(-x)$

Pro úhel $\beta = -\frac{\pi}{2}$ degraduje matice na tvar [13]

$$R(\alpha, \beta, \gamma) = \begin{bmatrix} 0 & 0 & -1 \\ -\sin(\alpha - \gamma) & \cos(\alpha - \gamma) & 0 \\ \cos(\alpha - \gamma) & \sin(\alpha - \gamma) & 1 \end{bmatrix}$$

a pro úhel $\beta = +\frac{\pi}{2}$ na tvar:

$$R(\alpha, \beta, \gamma) = \begin{bmatrix} 0 & 0 & 1 \\ \sin(\alpha + \gamma) & \cos(\alpha + \gamma) & 0 \\ \cos(\alpha + \gamma) & \sin(\alpha + \gamma) & 1 \end{bmatrix}$$

Z degradovaných matic výše lze jasně vidět, že je možné vypočítat pouze součet, respektive rozdíl úhlů α a γ , nikoliv úhly jako takové.

Pro popis singularity při rotaci kolem obecné osy bude zvolena jiná reprezentace než pomocí úhlů, které svírá pevný rám s obecnou osou. Osa bude vyjádřena jako vektor $r = [r_x, r_y, r_z]^T$. Odpovídající matice rotace bude mít poté následující tvar [11]:

$$R(r, \theta) = \begin{bmatrix} r_x^2(1 - c_\theta) + c_\theta & r_x r_y(1 - c_\theta) + r_z s_\theta & r_x r_z(1 - c_\theta) + r_y s_\theta \\ r_x r_y(1 - c_\theta) + r_z s_\theta & r_y^2(1 - c_\theta) + c_\theta & r_y r_z(1 - c_\theta) + r_x s_\theta \\ r_x r_z(1 - c_\theta) + r_y s_\theta & r_y r_z(1 - c_\theta) + r_x s_\theta & r_z^2(1 - c_\theta) + c_\theta \end{bmatrix},$$

kde c_θ a s_θ značí $\cos(\theta)$ a $\sin(\theta)$.

Zde singularita tkví v tom, že $R(r, \theta) = R(-r, -\theta)$, a tedy, že odpovídající matice rotace bude shodná pro tytéž parametry v opačné polaritě. Nelze tudíž určit, jaká rotace byla, či má být provedena [11].

3.1.3 Jednotkový kvaternion

V předchozí podkapitole se ukázalo, že matice rotace je poměrně jednoduchý způsob, jak popsat rotaci respektive orientaci tělesa. Klíčovou nevýhodou aplikace matice rotace však byla singularita. Dalším způsobem, kterým je možné popsat rotaci či orientaci tělesa je jednotkový kvaternion, jehož výhodou je především právě odstranění problému singularity. Principiálně je podobný popisu pomocí rotace kolem obecné osy. Kvaternion je definován následovně [11, 13]:

$$\Pi = [\eta, \epsilon^t]^T,$$

kde $\eta = \cos(\frac{\theta}{2})$ a $\epsilon = \sin(\frac{\theta}{2})r$.

Protože se jedná o jednotkový kvaternion, bude platit [11, 13]:

$$||\Pi|| = \eta^2 + \epsilon_x^2 + \epsilon_y^2 + \epsilon_z^2 = 1$$

Jednotkový kvaternion je možné převést na odpovídající matici rotace [11, 13]:

$$R(\eta, \epsilon) = \begin{bmatrix} 2(\eta^2 + \epsilon_x^2) - 1 & 2(\epsilon_x\epsilon_y - \eta\epsilon_z) & 2(\epsilon_x\epsilon_z - \eta\epsilon_y) \\ 2(\epsilon_x\epsilon_y - \eta\epsilon_z) & 2(\eta^2 + \epsilon_y^2) - 1 & 2(\epsilon_y\epsilon_z - \eta\epsilon_x) \\ 2(\epsilon_x\epsilon_z - \eta\epsilon_y) & 2(\epsilon_y\epsilon_z - \eta\epsilon_x) & 2(\eta^2 + \epsilon_z^2) - 1 \end{bmatrix}$$

V případě kvaternionu nedochází k singularitě, jelikož rotaci (r, θ) i rotaci $(-r, -\theta)$ odpovídá tentýž kvaternion, při řešení inverzního problému převodu matice rotace na kvaternion tak existuje jednoznačné řešení [11].

3.2 Rozklad pohybu

Popis obecného (složeného) pohybu bývá zpravidla komplikovaný, a tak se místo přímého výpočtu využívá tzv. rozkladu pohybu.

Nejprve obecná rovnice rozkladu:

$$3_1 = 3_2 + 2_1, \tag{3.2.1}$$

kde číslo 1 označuje polohu pozorovatele (pevný rám) a číslem 3 je bod daného tělesa.

Levá strana rovnice (3.2.1) vyjadřuje souřadnice bodu tělesa 3 v souřadnicovém systému 1, tedy výsledné (absolutní) souřadnice, které je cílem zjistit. Dále se zavede nový souřadnicový systém pevně spojený s tělesem, jehož počátek bude ležet v referenčním bodu tělesa. Jako referenční bod se z praktických důvodů volí například těžiště tělesa.

Na pravé straně rovnice (3.2.1) se nachází součet dvou pohybů. První sčítanec vyjadřuje pohyb bodu tělesa 3 v nově vytvořeném souřadnicovém systému 2. Tomuto pohybu se říká relativní (druhotný). Druhý sčítanec popisuje pohyb souřadnicového systému 2 v souřadnicovém systému pevného rámu 1. Tento pohyb se nazývá unášivý.

Speciálním případem je situace, kdy unášivý pohyb je posuvný a relativní pohyb sférický, respektive rotační. Pak se tento rozklad nazývá základní prostorový, respektive rovinný [10].

²Parametry θ a r odpovídají parametrům při popisu rotace kolem obecné osy

3.3 Výpočet polohy a orientace tělesa

Poznatky a vztahy z předchozích kapitol budou nyní využity k popsaní výpočtu polohy a orientace tělesa v prostoru.

Při úvaze základního rozkladu je unášivý pohyb pohybem translačním. Pro výpočet polohy tělesa postačí vypočítat polohu jednoho bodu, podle kterého lze vypočítat posun všech ostatních bodů tělesa. Jako bod nechť je zvoleno těžiště tělesa O_2 , které je zároveň počátkem souřadnicového systému tělesa. Složky zvoleného bodu O_2 v souřadnicovém systému 1 udávají jednotlivé složky translačního pohybu. Matematický zápis je pak³:

$$\vec{O}_2^1(t) = \left[O_{2x}^1(t) \quad O_{2y}^1(t) \quad O_{2z}^1(t) \right]^T,$$

kde \vec{O}_2^1 je vektor reprezentující translační pohyb.

Nechť je zvolen jiný referenční bod L, který je součástí tělesa. Jelikož je uvažováno dokonale tuhé těleso, bude vektor \vec{r}_L^2 konstantní velikosti. Protože je souřadnicový systém 2 pevně spojen s tělesem, jedná se o úlohu rotace souřadnicového systému 2 okolo v daném okamžiku pevného bodu O_2 . Samotná rotace nechť je vyjádřena pomocí matice rotace R_{12} .

Praktickou úlohou je pak vyjádření polohy bodu L v souřadnicovém systému 1. Polohu bodu je možné vypočítat jako [10]:

$$\vec{r}_L^1 = \vec{O}_2^1 + R_{12} \vec{r}_L^2 \quad (3.3.1)$$

Protože je uvažováno dokonale tuhé těleso, není připuštěna žádná deformace. Tím pádem matice rotace zachovává velikost vektoru a úhel dvou os. Platí tedy vztah [10]:

$$R_{12}^T R_{12} = I, \quad (3.3.2)$$

kde I je jednotková matice. Pokud bude rovnice (3.3.2) vynásobena zprava inverzní maticí R_{12}^{-1} , pak [10]:

$$R_{12}^T = R_{12}^{-1} \quad (3.3.3)$$

Vztah (3.3.3) dává předpis, jak snadno vypočítat inverzní matici. Inverzní matice může být užitečná, neboť popisuje inverzní operaci, čímž umožňuje vypočítat převod souřadnic ze systému 1 do systému 2. Matematicky tedy:

³V označení X_i^k udává i označení daného bodu a k souřadnicový systém, ve kterém je bod i popsán [13]

$$R_{21} = R_{12}^{-1}, \quad (3.3.4)$$

kde matice R_{21} je matice rotace ze systému 1 do systému 2. Za užití vztahu (3.3.4) a (3.3.1) lze vypočítat souřadnice bodu v systému 2 [13]:

$$\vec{r}_{L2} = R_{21}\vec{r}_L^1 - R_{21}\vec{O}_2^1$$

Další vlastností matice R_{12} je její determinant, který je vždy roven [10]:

$$\det(R_{12}) = \pm 1,$$

kdy v případě $\det(R_{12}) = +1$ platí vztahy pro pravotočivý systém a v případě $\det(R_{12}) = -1$ pro systém levotočivý.

4 Plánování trajektorie pohybu

V minulé kapitole se práce věnovala obecným zákonům pohybu, které budou v praktické části využity. V této části bude pozornost zaměřena na generování trajektorie a všech jejích součástí.

Obecně se jedná o problém nalezení vhodné interpolující funkce v interním parametru u a následný převod na časový průběh trajektorie, který již může být předán do řídicího systému, který tento pohyb vykoná. Parametr u může nabývat hodnot z intervalu $\langle 0, 1 \rangle$, kdy se pro $u = 0$ nachází systém v počátečním bodě a pro $u = 1$ v bodě koncovém.

Z hlediska typu pohybu je možné rozdělit úlohy na dva typy: diskrétní a spojitý. Hlavní rozdíl mezi těmito typy je v podobě popisu dráhy mezi počátečním a koncovým bodem [11, 4]. V diskrétním případě se jedná o úlohu tzv. Point-to-Point. Při této variantě není důležitý průběh mezi body, hlavním a přirozeným požadavkem je pouze dosáhnout cílového bodu co nejrychleji a nejplynuleji. Je samozřejmé, že generátor musí dbát na fyzikální realizaci pohybu. Je tedy nutné přizpůsobit rychlost, zrychlení a případně vyšší derivace polohy daného systému. Dalším důvodem k omezení těchto veličin může být redukce vibrací a plynulý chod stroje. U spojitě varianty kromě startovního a koncového bodu zadá uživatel body popisující trajektorii, kterou musí stroj sledovat [4, 11, 7].

Jiným typem dělení by mohlo být podle počtu os pohybu, a to na pohyb jednoosý a víceosý. Jak už název napovídá, u jednoosého pohybu bude sledován pohyb pouze v jedné ose. Tento pohyb nalezne využití u jednoduchých manipulátorů a obráběcích strojů, ale také u dopravníků. Další možností využití je rozdělení víceosého pohybu na nezávislý pohyb v jednotlivých osách. U víceosého pohybu je nutné pohyb zkoordinovat ve všech osách a využít jej pak lze u složitějších manipulátorů a pokročilejších obráběcích strojů [4].

4.1 Souřadnicové systémy

Pro víceosý pohyb je nutné definovat souřadnicové systémy pro snadnější popis samotného pohybu. Publikace [4] definuje 3 souřadnicové systémy:

- 1) Souřadnicový systém programu/produktu (PCS) - tento systém je vztažený k prostředí nebo produktu, se kterým stroj interaguje. Výhodou této varianty je vlastnost, že je systém zaveden staticky nezávisle na konfiguraci přístroje, čímž je možné snadněji popsat trajektorii.

- 2) Souřadnicový systém stroje (MCS) - v tomto případě je systém pevně spojen se statickou základnou stroje. V těchto souřadnicích je pak určena pozice koncového efektoru pomocí translace a rotace jednotlivých částí stroje.
- 3) Souřadnicový systém aktuátorů (ACS) - tento systém popisuje polohu a orientaci jednotlivých aktuátorů, které vytváří pohyb efektoru podél trajektorie

Mezi těmito systémy existují jednotlivé transformace. Mezi systémem PCS a MCS platí klasická kartézská transformace popsaná v kapitole č. 3. Pro převod mezi MCS a ACS slouží tzv. kinematické transformace, které se uplatňují především v úloze řízení manipulátorů. Zjednodušeně řečeno se jedná o převod nastavení aktuátorů (poloha, rychlost a zrychlení) v jednotlivých kloubech stroje na polohu/orientaci efektoru. Nastavení aktuátorů se nazývá "kloubové souřadnice Θ ", poloha a orientace efektoru pak tzv. "zobecněné souřadnice X " [13].

Z hlediska převodu je pak možné definovat dvě úlohy: přímou a inverzní kinematickou úlohu [13].

- 1) Přímá kinematická úloha řeší problém převedení známých kinematických souřadnic na souřadnice efektoru. Matematicky je možné tento vztah zapsat následovně:

$$X = G(\Theta)$$

- 2) Opačnou úlohu, tedy nalezení nastavení aktuátorů na základě známé polohy a orientace koncového efektoru, se nazývá inverzní kinematická úloha. Matematicky se pak tento problém zapíše jako:

$$\Theta = G^{-1}(X)$$

Funkce G je hledaná funkce, která převede známé souřadnice na požadovanou konfiguraci kloubů, respektive polohu a orientaci efektoru.

4.2 Interpolace

Jak již bylo řečeno v předchozích kapitolách, problém interpolace by se dal popsat jako problém hledání vhodné funkce, která bude na určitém intervalu procházet zadanými body. Nejpoužívanějšími funkcemi pak bývají funkce polynomiální (především pak lineární) či funkce kruhové. Interpolace může být ale obecně složitá úloha, která vede k velmi komplikovanému řešení například v podobě funkce vysokého řádu. Problémem předchozího příkladu je jednak z matematického hlediska nalezení

příslušných parametrů, kterých se zvyšujícím se řádem polynomu přibývá. Zároveň z praktického hlediska bude výsledná trajektorie nadměrně zvlněná, což je samozřejmě nežádoucí [4].

Řešením tohoto problému je následné rozdělení trajektorie na menší segmenty. Na jednotlivých segmentech je poté hledána interpolující funkce, která proloží body na daném úseku. Výsledkem bude sada "jednodušších" funkcí, pomocí nichž je možné propojit všechny zadané body. V tomto případě dochází ale k jiné překážce, a to v podobě nespojitosti na okrajích jednotlivých segmentů. To může způsobit problém při řešení úlohy derivací polohy. Z matematické analýzy je známo, že derivace funkce neexistuje v bodech nespojitosti. Je tedy nutné v těchto bodech funkce vhodně "napojit", aby nedošlo k nespojitosti v poloze a bylo možné polohu derivovat pro výpočet rychlosti a zrychlení. V opačném případě by v praxi došlo k neúměrnému skoku v rychlosti a tedy požadavku na velké zrychlení, který může být násobně větší, než je maximální možná hodnota zrychlení. Mimo to, tyto fluktuace mohou například u obráběcích strojů způsobit nekvalitní obrábění a při častém opakování by mohlo dojít i k poškození samotného stroje. Z tohoto důvodu je třeba navrhnout trajektorii v případě jejího rozdělení na více menších segmentů tak, aby segmenty na sebe plynule navazovaly, čímž se zajistí plynulý "přechod" z jednoho segmentu do druhého [4].

V rámci této práce budou uvedeny čtyři příklady interpolace. Zvláště pak u lineární a kruhové interpolace je nutné dbát na výše zmíněný problém, tj. nespojitost polohy vedoucí na skoky v rychlosti a zrychlení. Dále také bude popsána tzv. hélická interpolace a interpolace pomocí tzv. NURBS křivek.

4.2.1 Přímková interpolace

V nejjednodušším případě se využívá lineární interpolace, tedy proložení zadaných bodů přímkou, respektive více přímkami spojující průjezdové body. Přímka se vyjádří parametricky v závislosti na parametru u ve tvaru:

$$p(u) = p_0 + au, \quad u \in \langle 0, 1 \rangle, \quad (4.2.1)$$

kde p_0 je počáteční bod a a je sklon přímky, který je možné vyjádřit jako $a = p(1) - p(0)$. Lineární interpolace je tzv. přirozeně parametrizovaná, to znamená, že ujetá dráha stroje je lineárně závislá na inkrementaci parametru u [4].

4.2.2 Kruhová interpolace

Dalším příkladem přirozeně parametrizované křivky je kružnice. Kružnice může být definována v prostoru několika způsoby [4]:

- 1) Krajní body a střed - v tomto případě nastává problém duality, protože mohou být vygenerovány dvě různé cesty (po směru/proti směru hodinových ručiček). Musí být tedy požadován další parametr, který bude definovat směr trajektorie. V rámci G-kódu u CNC obráběcích strojů je pak tento problém řešen například dvěma funkcemi (G2 - po směru ručiček, G3 - proti směru ručiček)
- 2) Tři body v prostoru - dva body reprezentují počáteční a koncový bod, třetí definuje průjezdní bod a řeší tak problém předchozí varianty
- 3) Dva body a orientace - v této variantě jsou zadány dva krajní body a vektor ležící v rovině kružnice s velikostí rovnou poloměru kružnice

Parametricky popsat pohyb po kružnici v prostoru je poměrně náročné, a proto se pro zjednodušení zavede nový vhodný souřadnicový systém. Počátek nového systému bude ležet ve středu kružnice, nová osa x bude směřovat k počátečnímu bodu a nová osa y bude ležet ve směru vektoru orientace. Nová osa z bude kolmá na zbylé dvě osy, přičemž orientace bude určena tak, aby byl nový systém pravotočivý [11, 4].

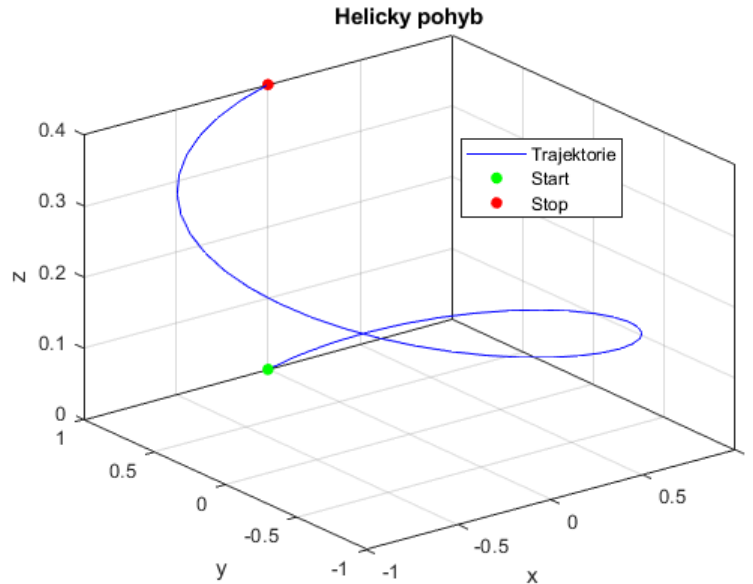
V takto definovaném systému je možné křivku parametricky popsat [11, 4]:

$$p(u) = \begin{bmatrix} r \cos\left(\frac{s_f u}{r}\right) \\ r \sin\left(\frac{s_f u}{r}\right) \\ 0 \end{bmatrix}, \quad (4.2.2)$$

kde r je poloměr kružnice, s_f je délka opisovaného oblouku a u je interní parametr nabývající opět hodnot z intervalu $\langle 0, 1 \rangle$.

4.2.3 Hélická interpolace

Hélická interpolace je spojení dvou předchozích způsobů interpolace. Při této variantě dochází k současnému pohybu po kružnici (oblouku kruhu) v osách x a y , zatímco se lineárně zvyšuje/snižuje ve třetí souřadnici z . Klíčová výhoda této interpolace je její jednoduchost, kdy za použití dvou jednoduchých typů interpolace je možné dosáhnout celkem komplexního pohybu. Hélická varianta se využívá především v rámci obrábění [6]. Příklad hélického pohybu je vidět na obrázku č. 3.



Obrázek 3: Příklad hélického pohybu mezi dvěma body

4.2.4 Interpolace spline křivkami

V praktických úlohách však lineární a kruhová interpolace a jejich případné kombinace nemusí dostatečně přesně interpolovat uživatelská data. Může tak dojít k nespojitosti v poloze při navazování jednotlivých segmentů, a proto se přechází k obecnějším spline křivkám, které jsou již data schopné proložit.

Nejobecnějším případem spline křivek jsou tzv. NURBS (Non-uniform rational basis spline). NURBS vznikly jako dvojí zobecnění tzv. B-spline křivek. Písmeno "B" označuje slovo báze, jejichž lineární kombinací je popsána křivka na daném segmentu. Slovo "spline" značí, že křivka je definována po částech (segmentech) [9]. NURBS křivky jsou tedy po částech racionální funkce popsané následujícími parametry [4, 9]:

- 1) Stupeň křivky p - obecně platí, že polynomem vyššího stupně lze interpolovat data přesněji a je také zajištěna spojitost vyšších derivací polohy, ale za cenu nežádoucího velkého zvlnění. Z tohoto důvodu se jako kompromis používá nejčastěji kvintický polynom, tedy polynom 5. stupně, který zajistí spojitost až do jerku a zároveň je zvlněný v "rozumných" mezích.
- 2) Stupeň křivky k - udává počet řídicích bodů, které ovlivňují libovolný bod křivky. Je možné jej určit jako $k = p + 1$.
- 3) Uzlový vektor $U = \{u_0, u_1, \dots, u_m\}$, kde $m = \tilde{n} + p + 1$ - udává úsek křivky ovlivněný

jednotlivými řídicími body. Jedná se o neklesající posloupnost čísel, které se nazývají uzly. Křivka, pro kterou platí $u_{i+1} - u_i = konst, \forall i$, je nazývána uniformní. Z praktických důvodů je vhodné například zvolit neuniformní parametrizaci v závislosti vzdálenosti mezi řídicími body a přiblížit se tak přirozené parametrizaci. Prvních $p + 1$ prvků je voleno rovných nule, čímž se zajistí, že průběh křivky začne přímo v počátečním bodu. Ze stejného důvodu pak posledních $p + 1$ prvků budou rovny jedné, aby křivka skončila přímo v konečném bodě.

- 4) Bázové funkce - definují chování křivky v jednotlivých segmentech. Je možné je matematicky definovat jako:

$$N_{i,1}(u) = \begin{cases} 1 & \text{pro } u_i \leq u < u_{i+1} \\ 0 & \text{jinak} \end{cases}, \quad i = 0, 1, \dots, n$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

V rámci rekurzivního počtu se může dojít k výrazu $\frac{0}{0}$. V takovém případě je výsledek této operace rovný 0.

- 5) Řídicí polygon $P = \{P_0, P_1, \dots, P_n\}$ - pomocí řídicích bodů lze tvarovat funkci na daném segmentu bez nutnosti změny například již nalezených parametrů křivky
- 6) Vektor váhových koeficientů $W = \{w_0, w_1, \dots, w_n\}$ - tento vektor ovlivňuje váhu řídicích bodů, tedy jak moc se bude křivka k těmto řídicím bodům přimykát.

V tuto chvíli je již možné uvést matematický předpis pro B-spline křivku:

$$C(u) = \sum_{i=0}^n N_{i,p}(u) P_i, \quad u \in \langle 0, \dots, 1 \rangle \quad (4.2.3)$$

Dvojitě výše zmíněné zobecnění B-spline spočívá v tom, že uzlový vektor nemusí být uniformní a jednotlivým řídicím bodům je možné přiřadit váhu. Zobecnění se tak dotkne i bázových funkcí, které jsou pro NURBS křivky definovány následovně [9]:

$$R_{i,p}(u) = \frac{N_{i,p}(u) w_i}{\sum_{j=0}^n N_{j,p}(u) w_j} \quad (4.2.4)$$

Nyní bude uveden matematický předpis pro obecnou NURBS křivku [4]:

$$\Phi(u) = \frac{\sum_{i=0}^n N_{i,p}(u)P_i w_i}{\sum_{i=0}^n N_{i,p}(u)w_i} = \sum_{i=0}^n R_{i,p}(u)P_i, \quad u \in \langle 0, \dots, 1 \rangle \quad (4.2.5)$$

Ve tvaru, jak byla křivka popsána výše, se jedná o funkci aproximační. To znamená, že křivka nebude procházet všemi body (kromě počátečního a koncového), ale bude je aproximovat. Aproximaci může být vhodné použít v případech, kdy je velký počet zadaných bodů (například při opakování pohybu na základě dat ze senzorů pohybu) a očekává se chyba měření. Je proto žádoucí body neprocházet přímo, ale zachytit základní tvar plánované trajektorie [4].

V případě potřeby navrhnutí interpolační funkce, je nutné určit tzv. vektor parametrizace. Vektor parametrizace je neklesající posloupnost čísel $h_i \in \langle 0, \dots, 1 \rangle$ [9]:

$$H = (h_i)_{i=0}^{\tilde{n}}$$

kde \tilde{n} je počet bodů interpolace.

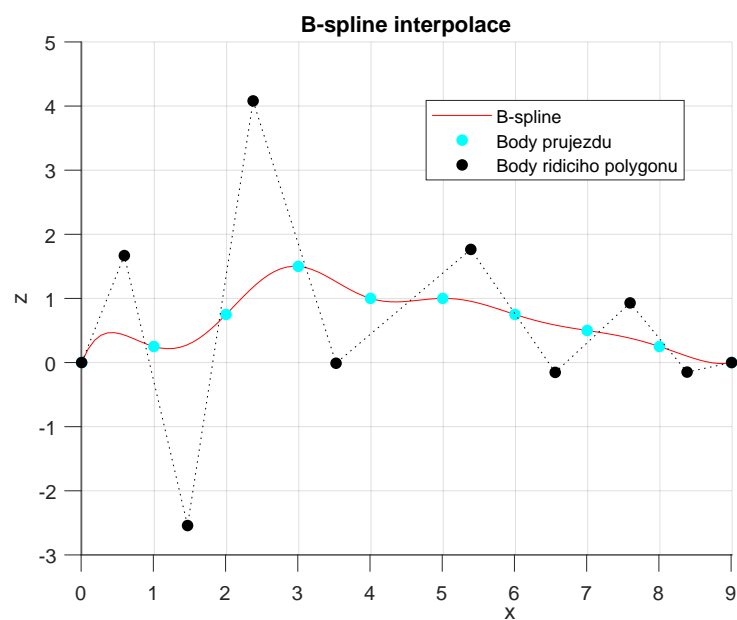
Vektor parametrizace je dále využit pro výpočet bodů řídicího polygonu P . Řídící body jsou určeny z následující podmínky [9]:

$$\sum_{j=0}^{\tilde{n}} N_{j,p}(h_i)P_j = Q_i, \quad i = 0, 1, \dots, \tilde{n} \quad (4.2.6)$$

Výše uvedenou rovnici je možné také často najít v maticové formulaci následovně:

$$\begin{bmatrix} N_{0,p}(h_0) & N_{1,p}(h_0) & \dots & N_{\tilde{n},p}(h_0) \\ N_{0,p}(h_1) & N_{1,p}(h_1) & \dots & N_{\tilde{n},p}(h_1) \\ \vdots & \vdots & \ddots & \vdots \\ N_{0,p}(h_{\tilde{n}}) & N_{1,p}(h_{\tilde{n}}) & \dots & N_{\tilde{n},p}(h_{\tilde{n}}) \end{bmatrix} \cdot \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_{\tilde{n}} \end{bmatrix} = \begin{bmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_{\tilde{n}} \end{bmatrix} \quad (4.2.7)$$

Příklad interpolace dat B-spline křivkou je vidět na obrázku č. 4. Na tomto obrázku lze pozorovat jak průběh trajektorie, tak řídicí polygon.



Obrázek 4: Příklad interpolace dat B-spline křivkou

5 Návrh generátoru trajektorie

V této kapitole bude popsána praktická část práce, která se zabývá návrhem generátoru trajektorie v celém rozsahu. Program byl vytvořen ve vývojovém prostředí MATLAB ve verzi "2019a". Jak bylo zmíněno v předchozích kapitolách, generátor trajektorie lze rozdělit na dvě části. První je část geometrická, která pro zadaná vstupní data vygeneruje požadovanou trajektorii. Následně druhá část, která na základě této trajektorie vytvoří časový průběh pohybu, jenž je již možné předat nižší vrstvě mechatronického systému. Nejprve bude vysvětlena geometrická část.

5.1 Geometrická reprezentace trajektorie

Jako první byly vytvořeny funkce, které interpolují vstupní data. Vstupní data byla reprezentována pomocí struktury "cell". První prvek každého interpolačního bodu obsahuje jeho samotné souřadnice. Druhá pozice pak udává způsob interpolace, kterým je potřeba předchozí a současný bod proložit. Startovní, tedy první bod, má na druhé pozici číslo "0". Dále pak číslo "1" označuje lineární interpolaci. Pokud se na druhé pozici nachází vektor, je to příkaz ke kruhové interpolaci, kdy daný vektor reprezentuje vektor osy y nového souřadnicového systému kružnice. Přímka lineární interpolace je vytvořena dle předpisu (4.2.1). V případě kruhové interpolace je křivka reprezentována dle předpisu (4.2.2). Pro následné převedení mezi souřadnicovým systémem oblouku kružnice a Kartézským souřadnicovým systémem byla použita matice rotace ve tvaru (3.1.2).

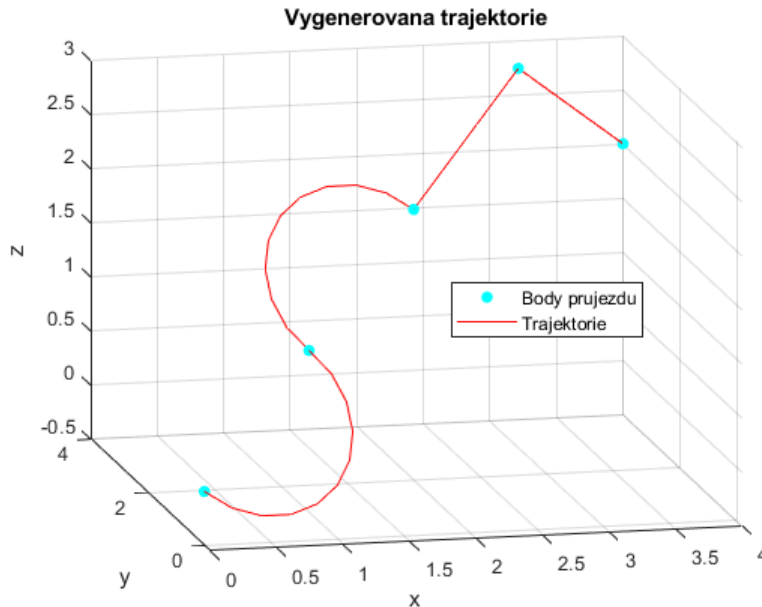
Jako další byla vytvořena funkce, která převezme vstupní data. Na základě výše popsaného určení rozhodne o typu interpolace. Jednotlivé segmenty na sebe napojí a vytvoří tak geometrickou reprezentaci výsledné trajektorie.

Testovací trajektorie je tvořena ze dvou kruhových a dvou lineárních segmentů. Ty byly voleny tak, aby se zde ukázaly problémy vznikající při napojování jednotlivých segmentů. Jednotlivé body jsou definovány následovně:

```
start = {[0 0 0], 0};  
pass1 = {[1 1 1], [1,-1,-1]};  
pass2 = {[2 2 2], [-1,1,1]};  
pass3 = {[3 3 3], 1};  
stop = {[4 4 2], 1};
```

Tedy nejprve bude propojen počátek s prvním bodem "pass1" pomocí kruhové interpolace se směrovým vektorem $[1, -1, -1]^T$. Obdobně je pak provedena další

kruhová interpolace pouze s odlišným směrovým vektorem. Následují dva lineární segmenty. Výslednou vygenerovanou trajektorii je poté možné vidět na obrázku č. 5.



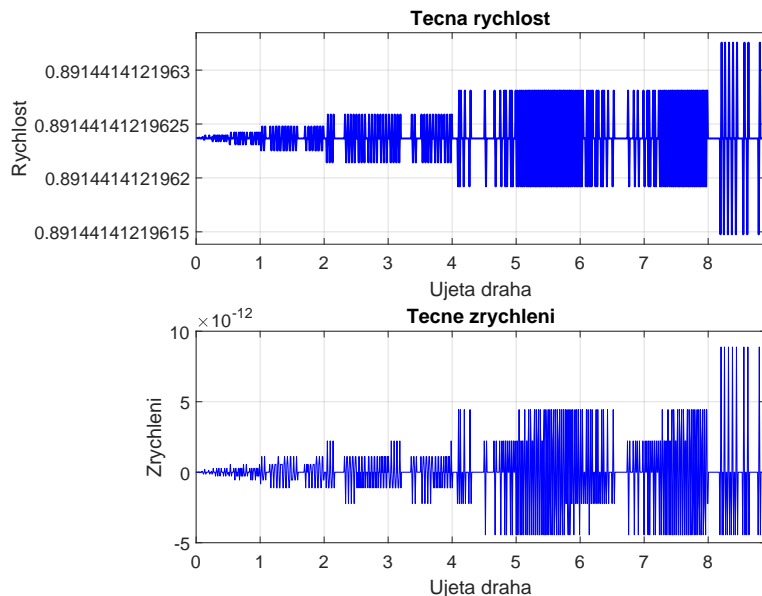
Obrázek 5: Vygenerovaná trajektorie na základě vstupních dat

5.2 Pohyb s konstantní rychlostí

Nyní, když je vytvořena trajektorie, je nutné ji převést na fyzikální pohyb. Je potřeba vytvořit požadovaný průběh ujeté dráhy, rychlosti a zrychlení. Nejjednodušším možným pohybem by byl pohyb s konstantní rychlostí a tedy i nulovým zrychlením. Toho lze dosáhnout snadno vygenerováním dostatečně hustého vektoru, jehož jednotlivé prvky budou ekvidistantní body ujeté dráhy v rozsahu od nuly do celkové délky trajektorie. Výsledný profil tečné rychlosti a zrychlení je vidět na obrázku č. 6.

Ač to na první pohled není patrné, jedná se skutečně o profil s konstantní rychlostí a nulovým zrychlením. Tyto derivace polohy jsou však vypočteny numericky, čímž zde přirozeně vzniká numerická chyba. Chyba je ale tak malá (řádově 10^{-12}), že by bylo možné ji zanedbat.

Ačkoli je pohyb s konstantní rychlostí velmi důležitý pro dosažení požadované kvality, například při obrábění, je nepřekonatelným problémem takového profilu jeho fyzikální nerealizovatelnost. Je nutné počítat s tím, že se stroj před realizací pohybu nachází v klidné poloze, tedy má nulovou rychlost a nulové zrychlení. Není tak



Obrázek 6: Tečná rychlost a zrychlení při požadavku na rovnoměrný pohyb s konstantní rychlostí

fyzikálně možné, aby během okamžiku zrychlil na požadovanou konstantní rychlost, nýbrž je nutné naplánovat plynulý rozjezd a dojezd z klidu do klidu s respektováním fyzikálních omezení stroje. Naplánováním takového profilu se bude zabývat následující podkapitola.

5.3 Fyzikálně realizovatelný pohyb

Jak bylo naznačeno v předchozí podkapitole, je potřeba naplánovat rychlostní profil tak, aby byl fyzikálně realizovatelný, to znamená tzv. pohyb z klidu do klidu. Zároveň bývá častým požadavkem udržovat po co možná nejdéle dobu konstantní rychlost. Dalším přirozeným požadavkem je projet trajektorii v nejkratším čase.

V této práci bude vytvářen program tak, aby splnil dvě výše uvedená kritéria. Zrychlení bude vždy konstantní. Při rozjezdu bude zrychlení rovné maximálnímu možnému zrychlení stroje a_{max} , při zpomalování naopak záporné hodnotě maximálního zrychlení. Tím se zajistí, že dráha bude ujetá v nejkratším možném čase. Jelikož integrací zrychlení je vypočtena rychlost, bude mít rychlostní profil lineární průběh. Při rozjezdu a zpomalení bude rychlost lineárně růst či klesat se sklonem rovným zrychlení, tedy $v = \pm a_{max} \cdot t$. Jakmile je dosažena maximální rychlost stroje v_{max} , zůstane rychlost konstantní až do chvíle zpomalení. Opětovnou integrací rychlosti bude vypočten průběh ujeté dráhy. S ohledem na průběh dráhy je také

tento profil nazýván "S-curve" (viz. obrázek č. (7) či (8)).

Kód funkce tvořící rychlostní profil pak začíná výpočtem dráhy potřebné pro zrychlení z nulové rychlosti na maximální rychlost a zpomalení opět do klidu v poměru ku celkové délce pohybu. Na základě tohoto poměru je pak vyhodnoceno, zda-li se vygeneruje tzv. "lichoběžníkový" profil či profil "trojúhelníkový".

Čas potřebný pro zrychlení na maximální rychlost je daný jako:

$$t_a = \frac{v_{max}}{a_{max}}$$

Z tohoto času je pak možné vypočítat dráhu potřebnou pro zrychlení:

$$s_a = \frac{a_{max} \cdot t^2}{2} \quad (5.3.1)$$

Pokud je dráha potřebná na zrychlení větší, než polovina celkové dráhy, pak je jasné, že stroj nemůže dosáhnout maximální rychlosti tak, aby stihl opět plynule zabrzdít.

Pohyb samotný se pak vypočítá dle následujících vztahů. Pro zrychlení platí:

$$\begin{aligned} a &= a_{max} \\ v &= \int a \, dt = a_{max} \cdot t \\ s &= \int v \, dt = \frac{a_{max} \cdot t^2}{2} \end{aligned} \quad (5.3.2)$$

Při pohybu konstantní (maximální) rychlostí platí:

$$\begin{aligned} a &= 0 \\ v &= \int a \, dt = v_{max} \\ s &= s_a + \int v \, dt = s_a + v_{max} \cdot t \end{aligned} \quad (5.3.3)$$

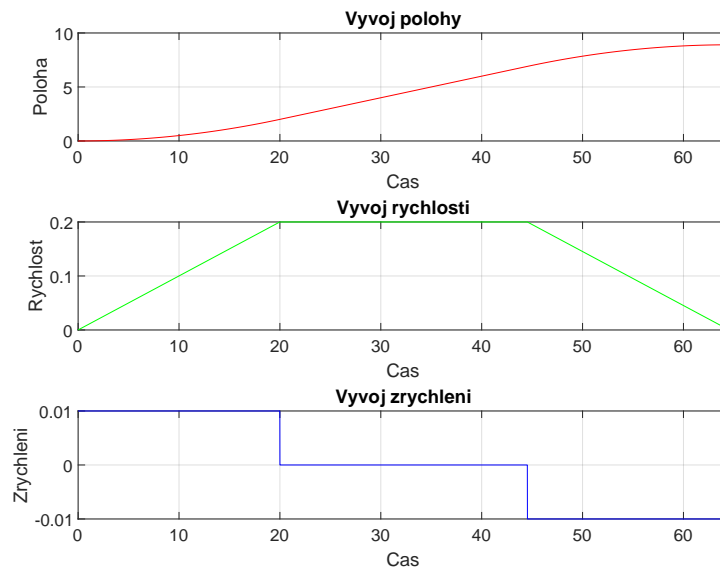
A nakonec při zpomalování je pohyb určen následovně:

$$\begin{aligned} a &= -a_{max} \\ v &= v_{max} - \int a \, dt = v_{max} - a_{max} \cdot t \\ s &= s_d + \int v \, dt = s_d + v_{max} \cdot t - \frac{a_{max} \cdot t^2}{2} \end{aligned} \quad (5.3.4)$$

Hodnota s_d je předem vypočtený bod na dráze, při kterém je nutné začít pohyb zpomalovat. Jelikož pro zrychlení i zpomalení je potřeba stejná dráha, je možno tuto hodnotu určit jako $s - s_a$.

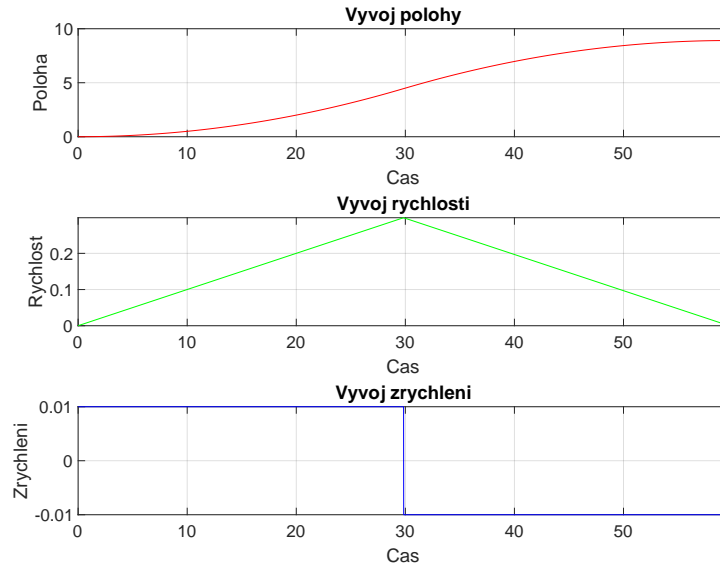
5.3.1 Příklady vygenerování rychlostních profilů

Lichoběžníkový profil se vygeneruje v případě, že stroj stihne zrychlit na maximální rychlost, určitý čas touto konstantní rychlostí jet a zároveň pak stihne opět plynule zastavit. Pro příklad byl tento profil vygenerován s parametry $a_{max} = 0.01$ a $v_{max} = 0.2$ a je vidět na obrázku č. (7).



Obrázek 7: Lichoběžníkový rychlostní profil

Trojúhelníkový profil vznikne ve chvíli, kdy za daných parametrů stroj nestihne při maximálním zrychlení dosáhnout maximální rychlosti do doby, než je potřeba opět začít zpomalovat, aby stroj stihl zastavit. Rychlost tedy bude růst do poloviny dráhy, poté začne lineárně klesat do nuly. Takový profil lze vidět na obrázku č. (8) a byl vygenerován s parametry $a_{max} = 0.01$ a $v_{max} = 0.4$.



Obrázek 8: Trojúhelníkový rychlostní profil

5.4 Časový průběh pohybu

Výstup funkce popsané v předchozí podkapitole je vektor popisující ujetou dráhu vygenerovanou s respektem k požadované rychlosti a zrychlení v daných časech. Nyní je potřeba na základě tohoto vektoru vytvořit sekvence polohy, rychlosti a zrychlení v daných okamžicích s periodou vzorkování, které je následně možné předat regulátoru pohybu jako setpointy po celou dobu výkonu pohybu. Je tedy potřeba vytvořit algoritmus, který na základě požadované změny dráhy vypočítá skutečnou polohu v daném čase. Protože jsou obě křivky, přímka i kružnice definované parametricky, je vhodné vypočítat hodnotu interního parametru u tak, aby vzdálenost ujetá od posledního bodu odpovídala požadované délce ujetého oblouku. Jelikož jsou přímka i kružnice přirozeně parametrizované křivky, je výpočet interního parametru celkem jednoduchý.

Celková trajektorie je až na výjimečné triviální případy složena z několika částí (segmentů). Jako první je tedy potřeba zjistit segment, na kterém se stroj teoreticky nachází. To lze snadno určit, jelikož jsou známé délky z interpolace jednotlivých segmentů. Poté, co je určen segment, stačí vypočítat, jak velkou část daného segmentu již stroj ujel. Díky přirozené parametrizovatelnosti bude u odpovídat poměru zjištěné délky ku celkové délce segmentu.

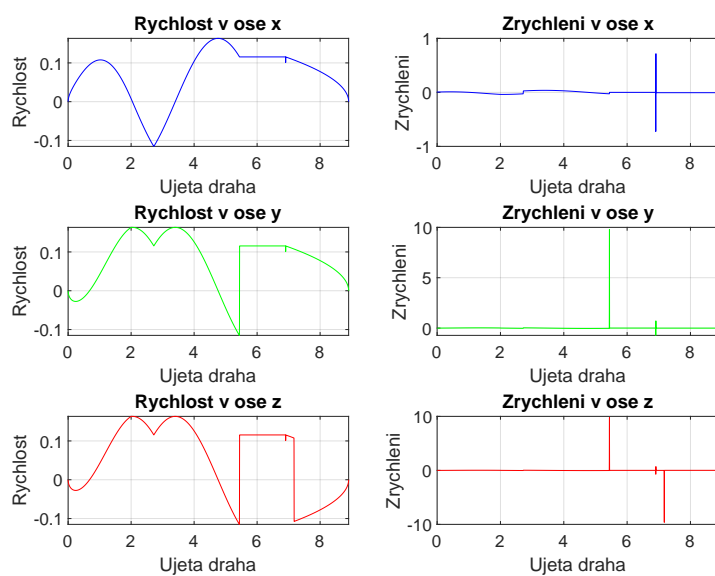
Ve chvíli, kdy je znám segment i parametr u , je nutné vypočítat polohu stroje na základě těchto dvou údajů. Pro tento účel byla vytvořena metoda, která pro konkrétní u a segment dosadí do parametrických předpisů funkcí a určí tak konkrétní

polohu stroje. Pokud tento cyklus projde přes celou délku dráhy, vytvoří se sekvence bodů ve trojdimenzionálním prostoru, které je již možné předat regulátoru pohybu jako referenční hodnoty v daných časech definovaných periodou vzorkování.

Pro sekvenci bodů reprezentující polohu je vhodné vypočítat odpovídající rychlost a zrychlení v jednotlivých osách. Tyto derivace polohy byly vypočteny numericky. Rychlost a zrychlení v osách jsou pak také předávány regulátoru pohybu jako referenční hodnoty. Průběh těchto dvou veličin v čase je vidět na obrázku č. 9.

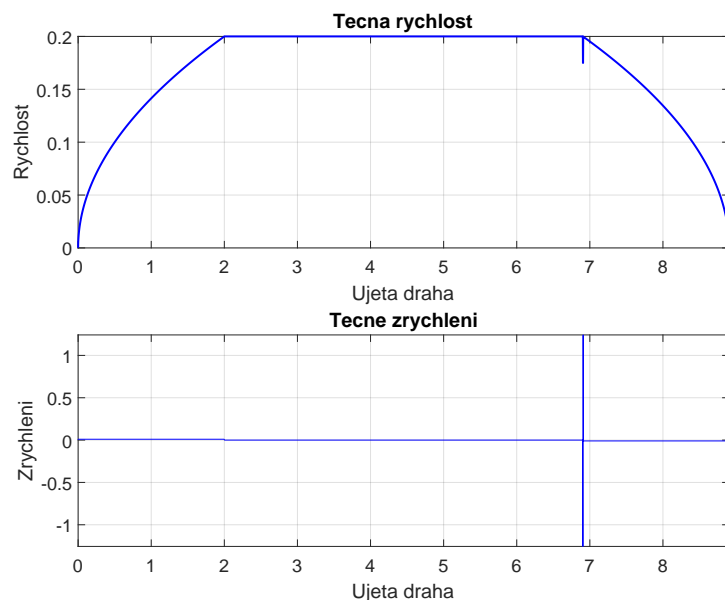
Na závěr byla také vypočítána a vykreslena tečná rychlost a tečné zrychlení. Tyto veličiny by měly odpovídat rychlostnímu profilu vytvořenému generátorem pojezdu. Tečnou rychlost a zrychlení lze vidět na obrázku č. 10.

Oba výše zmíněné grafy obsahují průběh rychlosti a zrychlení vypočtených pro lichoběžníkový rychlostní profil popsany v předchozí podkapitole.



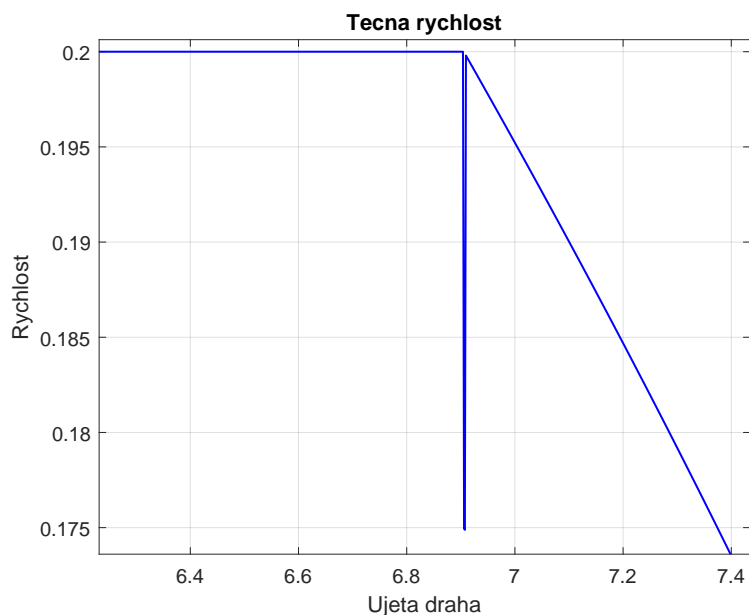
Obrázek 9: Časový průběh rychlosti a zrychlení v jednotlivých osách při interpolaci přímkami a kružnicemi

Při pohledu na grafy rychlostí a zrychlení je patrná nespojitost v okamžiku, kdy by stroj měl začít zpomalovat. Toto vychýlení vzniká v důsledku diskretizace. Momenty, do kterého by měl stroj zrychlovat, a od kterého zpomalovat, byly vypočteny ve spojitěm čase. Profil je však dán jako sekvence bodů s danou periodou vzorkování. Časy jsou vypočteny s větší přesností, než je perioda vzorkování, a proto může dojít (a ve výše uvedených případech došlo) k situaci, kdy přesný čas není násobkem



Obrázek 10: Časový průběh tečné rychlosti a tečného zrychlení

periody vzorkování. Výsledkem je, že tyto dva body budou u sebe blíže, než by dle rychlosti měly být. Tato situace je zobrazena na obrázku č. (11). Jelikož je hlavním cílem udržet konstantní rychlost po co nejdéle, není pro účely této práce tento problém nutné řešit.



Obrázek 11: Nespojitosť rychlosti

5.5 B-spline interpolace

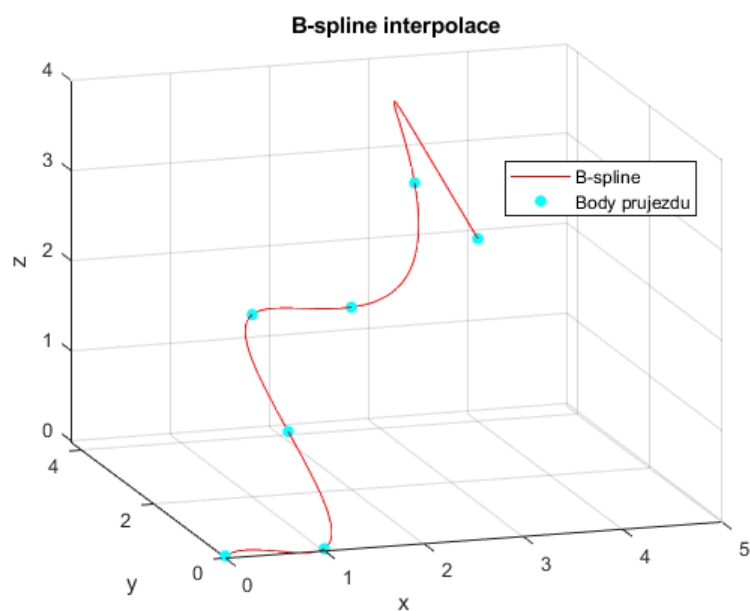
Doposud byl představen návrh generátoru trajektorie od interpolace až po vytvoření požadovaného fyzikálního pohybu, prozatím však pouze pro sekvenci přímk a kružnic. Jak již bylo popsáno v teoretické části této práce, u takové trajektorie dochází k nespojitosti v místech, kde se napojuje jeden segment na druhý. Tato nespojitost v poloze způsobí přirozeně i nespojitost jejích vyšších derivací. Takové nespojitosti je možné vidět na obrázku č. 9 v předchozí podkapitole, kde v místě napojení třetího segmentu (první přímka) na segment druhý (druhá kružnice) dochází ke skokové změně rychlosti v ose y a z , což způsobí prudký nárůst zrychlení, které by požadovanou změnu rychlosti umožnilo. Takové zrychlení však není fyzikálně realizovatelné. Tento problém bude v této práci řešen pomocí interpolace B-spline křivkou.

Reprezentace vstupních dat je stejná jako v případě lineární interpolace, jen namísto čísla 1 je na druhé pozici všech bodů kromě prvního uvedeno číslo 2. V rámci této práce byl navržen algoritmus pro B-spline křivku 3. stupně, který byl poté rozšířen s minimálními úpravami na křivku 5. stupně. Program pro generování B-spline 3. stupně byl navržen podle řešeného příkladu v publikaci [9].

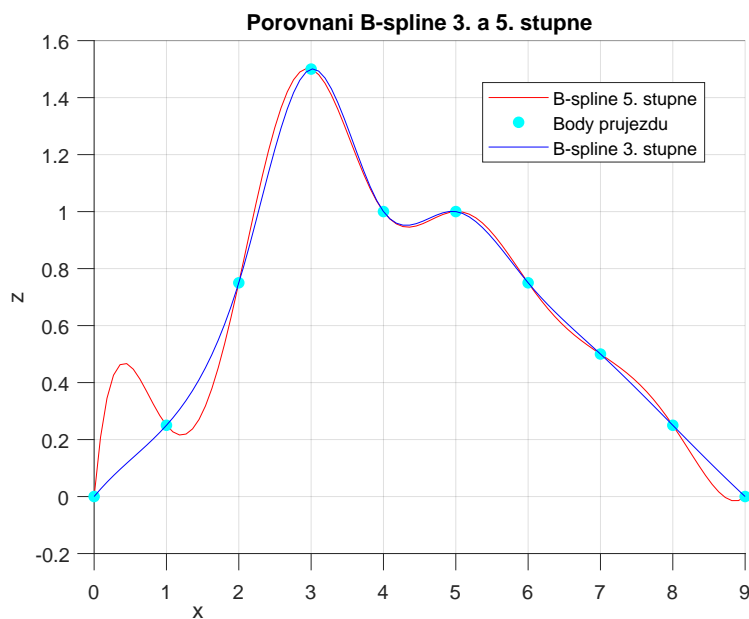
Na obrázku č. 12 je zobrazena B-spline interpolace (5. stupně), která by měla interpolovat data uvedená na začátku kapitoly. Pro interpolaci byly oproti původním datům přidány body na vrcholech kružnic, aby došlo k "zachycení" správného tvaru trajektorie. U této trajektorie je patrná velká odchylka zvláště v oblasti napojení dvou přímk z důvodu "ostrého" navázání, a proto musí B-spline opsat velký oblouk. Algoritmus, který bude automaticky "fitovat" B-spline křivku na segmenty kružnic a přímk, a tím kompenzovat chybu, bude popsán na závěr praktické části.

Na obrázku č. 13 je pak vidět porovnání B-spline funkce 3. a 5. stupně. Je patrné, že křivka 3. stupně je méně "zvlněná" oproti křivce vyššího stupně, jak bylo uvedeno v teoretické části práce. Zároveň je však nutné podotknout, že vyšší stupeň poskytne spojitost vyšších derivací polohy oproti křivce nižšího stupně.

V obou výše uvedených grafech nebyl vykreslen z důvodu přehlednosti řídicí polygon.



Obrázek 12: Příklad interpolace dat B-spline křivkou 5. stupně



Obrázek 13: Porovnání B-spline křivek 3. a 5. stupně při interpolaci shodných dat

5.5.1 Převod na fyzikální pohyb

Narozdíl od přímky a kružnice není B-spline křivka přirozeně parametrizovaná. To znamená, že bude potřeba komplexnější algoritmus pro určení interního parametru u tak, aby stroj přesně ujel danou vzdálenost.

Prvním problémem je vypočítat délku oblouku obecné B-spline křivky. Pro účely programu je potřeba tuto vzdálenost vypočítat na libovolném intervalu interního parametru $\Delta u = \langle u_1, u_2 \rangle$. Matematicky je možná hledanou vzdálenost definovat následovně [4]:

$$L_{1,2} = \int_{u_1}^{u_2} \frac{ds}{du} du \quad (5.5.1)$$

Vztah (5.5.1) lze poté rozepsat na tvar:

$$L_{1,2} = \int_{u_1}^{u_2} \sqrt{\left(\frac{dx}{du}\right)^2 + \left(\frac{dy}{du}\right)^2 + \left(\frac{dz}{du}\right)^2} du = \int_{u_1}^{u_2} \|C'(u)\| du, \quad (5.5.2)$$

kde $C'(u)$ je derivace B-spline funkce. Vztah (5.5.2) nelze obecně řešit analyticky, a je tedy potřeba přejít k numerickému řešení. Nechť je $f(u) = \|C'(u)\|$ a $u_3 = \frac{u_1+u_2}{2}$. Potom lze integrál (5.5.2) vyřešit pomocí tzv. Simpsonova pravidla jako [4]:

$$L_{1,2} = \frac{u_2 - u_1}{6} \cdot (f(u_1) + 4 \cdot f(u_3) + f(u_2)) \quad (5.5.3)$$

Obdobně jsou pak vypočteny vzdálenosti $L_{1,3}$ a $L_{3,2}$ a poté vyhodnocena chyba následovně [4]:

$$\epsilon = |L_{1,3} + L_{3,2} - L_{1,2}| \quad (5.5.4)$$

V případě, že je ϵ větší než maximální povolená chyba, je Simpsonovo pravidlo použito znovu, jen tentokrát na intervalech $\langle u_1, u_3 \rangle$ a $\langle u_3, u_2 \rangle$. Výsledná délka $L_{1,2}$ a chyba ϵ bude součtem hodnot vypočtených na těchto subintervalech. Takto se bude pokračovat až do chvíle, kdy je ϵ menší než maximální povolená chyba.

Ve chvíli, kdy je již zkonstruován algoritmus pro výpočet délky opsaného oblouku, je možné přejít na výpočet odpovídajícího interního parametru u pro posun o zadanou vzdálenost Δs . Rovnici lze definovat následovně:

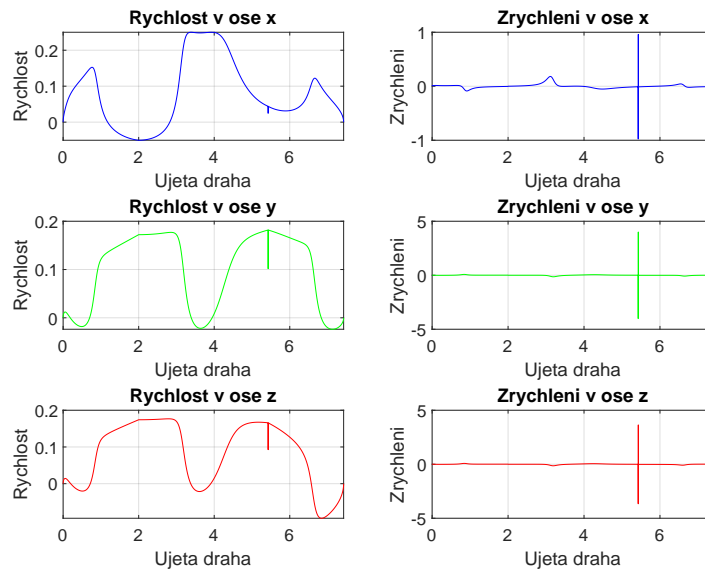
$$L_{1,2} = \int_{u_1}^{u_2} f(u) du \stackrel{!}{=} \Delta s \quad (5.5.5)$$

Jelikož se tato rovnice řeší pro celý vygenerovaný vektor ujeté dráhy, jsou v každé iteraci známé $\|C'(u)\|$, u_1 a Δs . Úkolem je tedy nalézt u_2 tak, aby výše uvedená

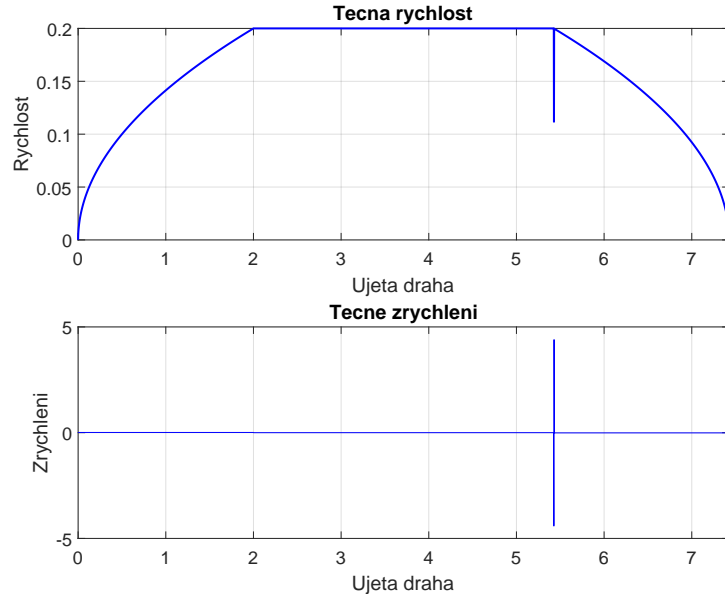
rovnice byla splněna. Vzhledem ke složitosti $\|C'(u)\|$ není možné opět tuto rovnici řešit analyticky. Rovnici (5.5.5) lze přepsat do tvaru:

$$L_{1,2} - \Delta s = 0 \quad (5.5.6)$$

Tuto rovnici je tedy třeba řešit numericky. Jako první byla vyzkoušena metoda tečen (Newton-Raphsonova). V praxi však nebyla příliš použitelná, jelikož se příliš často stávalo, že algoritmus nekonvergoval. Proto byla namísto toho použita metoda bisekce, neboli metoda půlení intervalu. Metoda bisekce patří ke spíše pomalejším numerickým metodám, ale pro účely této práce byla dostatečně rychlá (vygenerování celé sekvence parametru "u" pro celou délku trajektorie trvala v průměru 4 sekundy). Výsledný časový vývoj rychlosti a zrychlení v jednotlivých osách je vidět na obrázku č. 14 a na obrázku č. 15 je poté vidět průběh tečné rychlosti a zrychlení. Z obrázků je patrné, že narozdíl od trajektorie tvořené přímkami a kružnicemi zde nedochází (až na místo vzniklé v důsledku diskretizace rychlostního profilu v generátoru pojezdu, viz. kapitola 5.4) k nespojitosti v rychlosti a tedy ani k nerealizovatelným požadavkům na zrychlení.



Obrázek 14: Časový průběh rychlosti a zrychlení v jednotlivých osách při B-spline interpolaci



Obrázek 15: Časový průběh tečné rychlosti a zrychlení při B-spline interpolaci

5.6 Orientace pomocí jednotkového kvaternionu

V předchozích podkapitolách byla zatím řešena pouze poloha a její vyšší derivace. V praktických úlohách je ještě navíc potřeba uvažovat i orientaci stroje. V teoretické části bylo uvedeno několik možností, jak popsat orientaci tělesa, a v rámci této práce byla zvolena reprezentace pomocí jednotkového kvaternionu. Z praktických důvodů byla interpolace provedena pomocí B-spline křivky.

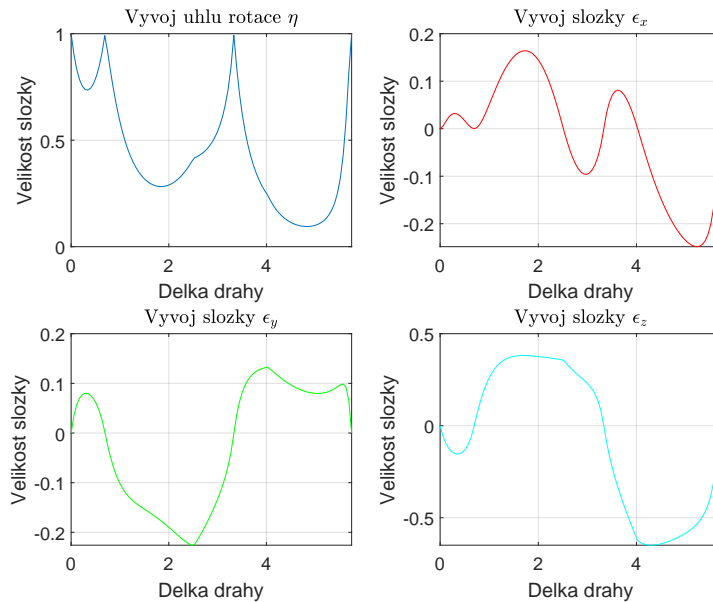
Pro kompletní interpolaci (poloha i orientace) bylo potřeba upravit strukturu vstupních dat. Každý bod nyní bude mít kromě prvních dvou pozic ještě pozici třetí a čtvrtou, kde na třetí pozici je požadovaná orientace v daném bodě a na čtvrté způsob interpolace (pro B-spline interpolaci číslo "2"). Vstupní data tedy poté vypadají například následovně:

```
start = {[0 0 0], 0, [0 0 -1 pi/2], 0};
pass1 = {[1 0 0], 2, [0 0 -1 (3*pi)/4], 2};
pass2 = {[2 1 0], 2, [1 1 -1 (3*pi)/4], 2};
pass3 = {[3 2 1], 2, [1 0 -1 pi/2], 2};
pass4 = {[4 2 3], 2, [-1 0 -1 (3*pi)/2], 2};
stop = {[5 2 2], 2, [0 0 -1 0], 2};
```

Informace na třetí pozici udává tři složky osy rotace r a čtvrté číslo reprezentuje

úhel rotace kolem dané osy θ v radiánech.

Pro interpolaci byla upravena metoda pro B-spline křivku 5. stupně. Nejprve je provedena samotná interpolace, čímž je vypočtena sekvence požadované orientace po celé délce trajektorie. Jelikož je známa osa i úhel rotace, je možné kvaternion sestavit dle předpisu (3.1.3). Aby měla takto sestavená reprezentace fyzikální smysl, je nutné provést korekci v podobě normování, aby byla velikost kvaternionu rovna jedné. Průběh jednotlivých složek kvaternionu je možné vidět na obrázku č. 16.



Obrázek 16: Vývoj jednotlivých složek jednotkové vektoru po B-spline interpolaci orientace

5.7 Vyhodnocení a kompenzace chyby B-spline interpolací

V kapitole 5.5 byl zmíněn problém interpolace v případě, kdy je dán požadavek na naplánování trajektorie B-spline křivkou, která by měla odpovídat trajektorii vytvořené sekvencí přímk a kružnic. Prvním problémem bylo sledování kružnice. Aby bylo možné "kopírovat" trasu kružnice, bylo nutné ručně přidat bod průjezdu do vrcholu kružnice. Druhým problémem byla chyba, kterou se dopustí B-spline křivka oproti původní trajektorii. Proto budou v této práci navrženy dvě korekční funkce, které by měly tyto chyby do jisté míry minimalizovat.

Než bude popsán způsob minimalizace chyby, je nejprve nutné určit způsob, jakým lze chybu vůbec vyhodnotit. Nejprve je vygenerována trajektorie přímk a

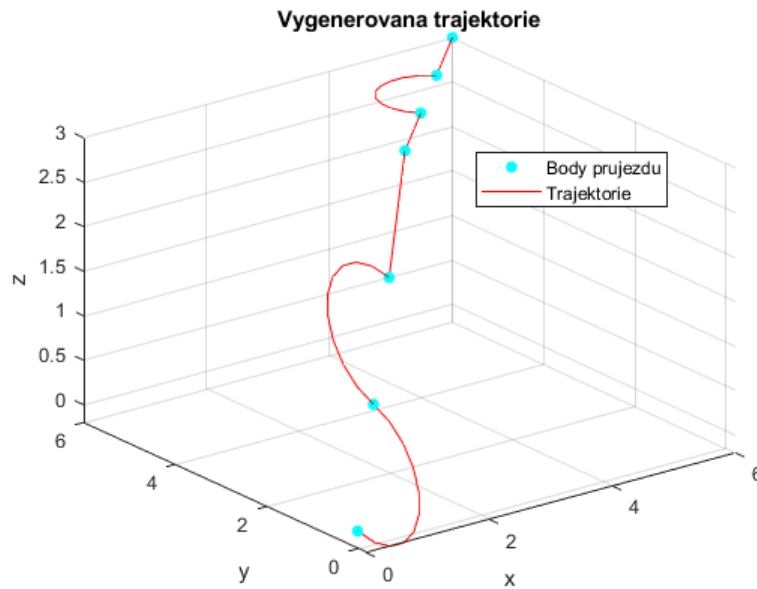
kružnic s určitým počtem vzorků. Hodnoty interního parametru u pro B-spline křivku jsou pak ekvidistantně rozloženy na intervalu $< 0, 1 >$ a počet vzorků odpovídá počtu vzorků původní trajektorie. Na základě analytického předpisu B-spline křivky z minulé iterace (pro první iteraci je předpis vstupním parametrem funkce) jsou vypočteny body polohy v kartézském prostoru pro odpovídající hodnotu u . Tím je získána sekvence bodů B-spline, přímek a kružnic. Mezi odpovídajícími body je poté vypočtena vzdálenost, která bude definovat chybu, které se B-spline interpolací dopustilo oproti původní trajektorii.

Nyní, když je definované vyhodnocení chyby, budou popsány dvě funkce, jejichž cílem je dopuštěnou chybu kompenzovat. Obě funkce pracují na podobném principu. V první iteraci dojde k přidání vrcholů všech kružnic na trajektorii. To způsobí, že v další interpolaci již bude základní tvar trajektorie odpovídat trajektorii původní. Od druhé iterace se funkce liší. První funkce vyhodnotí chyby pro jednotlivé hodnoty parametru u a v místě, kde je chyba největší, přidá bod průjezdu pro B-spline interpolaci v místě původní trajektorie. To zajistí, že B-spline křivka projde přesně tímto bodem. V další iteraci se poté vyhodnotí chyba mezi původní trajektorií s trajektorií B-spline křivky, u které byl přidán výše popsaný bod. Tato lokální úprava křivky způsobí, že na jiném místě dojde k většímu zakřivení, což potenciálně může vést na větší chybu, než byla před přidáním nového bodu. Pokud došlo ke zvětšení maximální chyby, je funkce ukončena a výsledkem je finální trajektorie bez posledního přidaného bodu. Druhá funkce pracuje velmi podobně. Rozdíl je v kompenzaci chyby, kdy narozdíl od první funkce je zde největší chyba řešena na každém segmentu. Je-li maximální chyba na daném úseku větší než předem stanovená maximální možná chyba, pak je stejným způsobem jako u první funkce zařazen nový bod průjezdu. Ač se může na první pohled zdát, že druhá funkce bude dosahovat lepších výsledků v důsledku přidání více bodů v rámci jedné iterace, a tedy přesnější interpolace, opak může být někdy pravdou. Přidání více bodů totiž způsobuje větší zvlnění mezi těmito body a tedy i větší celkové chyby. Proto mohou nastat situace, kdy pomocí první funkce lze minimalizovat chybu efektivněji.

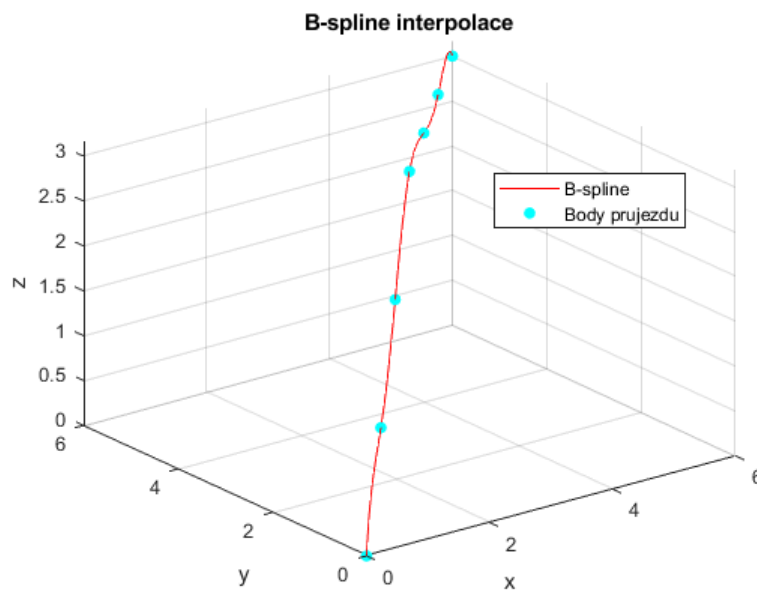
Alternativní variantou ukončení metody bylo nechat vygenerovat stanovený počet iterací a poté vybrat interpolaci s nejmenší chybou. Na všech testovacích trajektoriích se ukázalo, že jakmile se chyba přidáním bodu zvětší, již nedojde ke zlepšení výsledku. Proto i metoda s takovýmto ukončením vypočítala jako nejlepší možnou variantu tu trajektorii, kterou vypočítaly první dvě funkce.

Pro příklad byla sestavena následující trajektorie (obrázek č. 17 a 18). Jelikož při generování B-spline křivky nejsou zadané vrcholy kružnic jako body průjezdu,

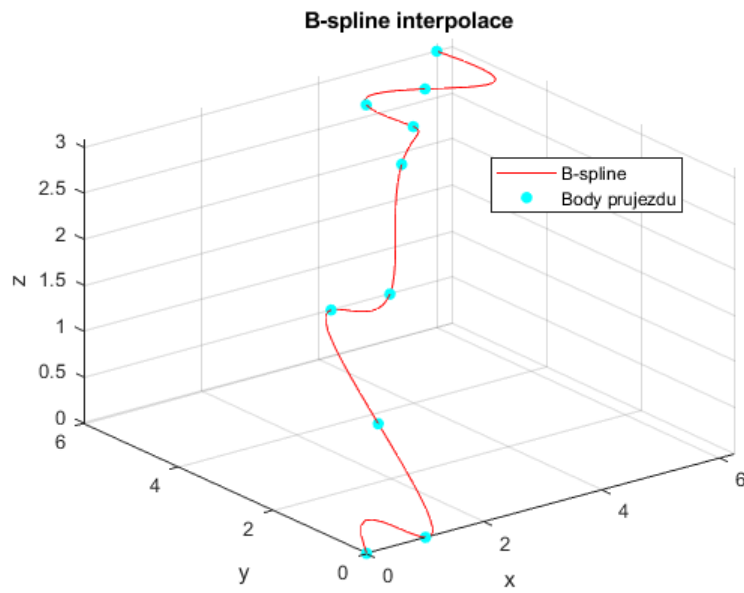
neodpovídá tato křivka vůbec původní trajektorii. Proto budou použity výše popsané funkce. Výsledky jsou k vidění na obrázcích č. 19 a 20. Zde se jedná o případ, kdy druhá funkce dosáhla přesnějšího sledování původní trajektorie. Konkrétně maximální chyba u první funkce byla rovna 0.8568 a tohoto výsledku bylo dosaženo ve dvou iteracích. Druhá funkce minimalizovala chybu na 0.6113 ve třech iteracích.



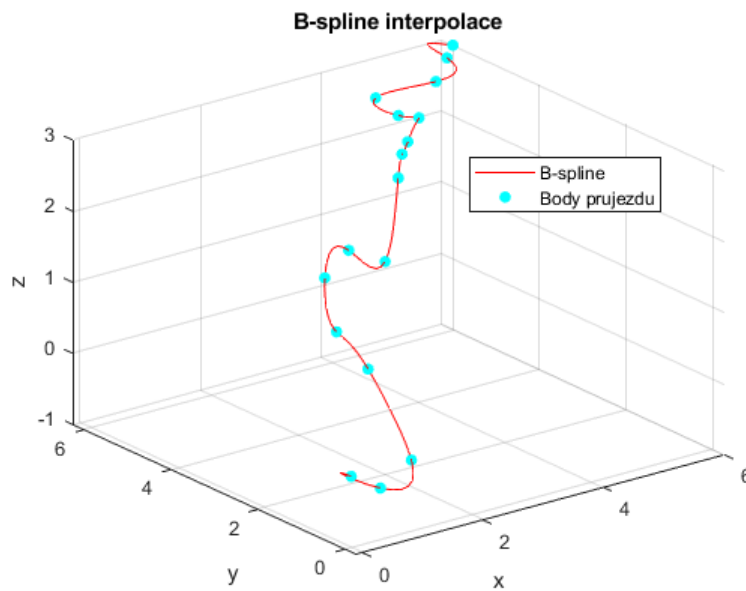
Obrázek 17: Referenční trajektorie tvořená sekvencí přímek a kružnic



Obrázek 18: B-spline interpolace testovacích dat pro kompenzaci chyby interpolace



Obrázek 19: Výsledná trajektorie při využití první korekční funkce



Obrázek 20: Výsledná trajektorie při využití druhé korekční funkce

6 Závěr

Hlavním cílem této práce bylo seznámit čtenáře s problematikou generování pohybu v celém rozsahu, tj. od interpolace dat (tedy vygenerování trajektorie, která projde všemi požadovanými body) přes návrh rychlostního profilu, až po vygenerování fyzikálního pohybu.

V teoretické části práce byl nejprve čtenáři představen vědní obor mechatronika se zaměřením na obecné rozdělení mechatronického mechanismu na jednotlivé vrstvy. Důraz byl kladen na představení generátoru trajektorie a několik typických úloh, které musí tato vrstva řešit. V další části se práce zaměřila na kinematiku tělesa a speciálně na jeho orientaci. Byly popsány základní principy pohybu, konkrétně rozklad pohybu. Poté byly představeny nejčastěji používané reprezentace orientace tělesa, včetně jejich předností a nedostatků. V poslední kapitole teoretické části byla vysvětlena problematika plánování trajektorie pohybu. Byly představeny vybrané souřadnicové systémy stroje, které je možné k popisu dat použít, včetně vzájemných transformací. Dále byly objasněny běžně používané způsoby interpolace uživatelských dat. Pozornost byla věnována hlavně B-spline a obecnějším NURBS křivkám.

Praktická část dokumentuje jednotlivé kroky návrhu generátoru trajektorie. Nejprve byly navrženy jednoduché interpolátory pro lineární a kruhovou interpolaci, dále i složitější metody pro B-spline křivky třetího a pátého stupně. Poté byl vytvořen generátor pojezdu, nejprve zjednodušeně pro konstantní rychlost, následně pro fyzikálně realizovatelný pohyb s ohledem na maximální rychlost a zrychlení. V dalším kroku byl navržen podprogram, který na základě požadovaného posuvu podél dráhy vygeneruje sekvence polohy, rychlosti a zrychlení, jež je již možné předat regulátorům jako referenční hodnoty. Tato část byla vytvořena jak pro přímozeně parametrizované křivky (přímka a kružnice), tak pro nepřímozeně parametrizované křivky v podobě B-spline. Interpolátor byl později rozšířen o interpolaci orientace B-spline křivkou. Poslední částí celého programu bylo navrhnutí funkcí, které stanoví a minimalizují chybu B-spline interpolace oproti původní trajektorii danou sekvencí přímek a kružnic.

Pro případné budoucí navázání na tuto práci by mohlo být zajímavým konceptem rozšíření metody interpolace B-spline křivek na metodu interpolace pomocí NURBS křivek. Hlavním přínosem je umožnění přidání váhy řídicím bodům, což by potenciálně mohlo zlepšit kvalitu interpolace. Dále by bylo vhodné navrhnout komplexnější a efektivnější metody kompenzace chyby interpolace B-spline křivkou oproti původní trajektorii tvořené sekvencí přímek a kružnic.

Použitá literatura

- [1] BRÁT, Vladimír, JÁČ, Václav a ROSENBERG, Josef. *Kinematika*. 1. vyd. Praha: SNTL, 1987. 250 s.
- [2] GOUBEJ, Martin a MELICHAR, Jiří. *Lineární systémy 1*. Učební text, Západočeská univerzita, Fakulta aplikovaných věd, 2017.
- [3] GOUBEJ, Martin. a MELICHAR, Jiří. *Lineární systémy 2*. Učební text, Západočeská univerzita, Fakulta aplikovaných věd, 2019.
- [4] GOUBEJ, Martin, ŠVEJDA, Martin, SCHLEGEL, Miloš. *Úvod do mechatroniky, robotiky a systémů řízení pohybu*. Skriptum pro studenty doktorských programů v oboru automatické řízení, Západočeská univerzita, Fakulta aplikovaných věd, 2012.
- [5] GREPL, Robert. *Kinematika a dynamika mechatronických systémů*. Vyd. 1. Brno: Akademické nakladatelství CERM, 2007. 158 s. ISBN 978-80-214-3530-8.
- [6] HENDLEY, Nate. *Making holes with helical interpolation*. Canadian Metalworking - Metal Fabricating News, Products, Articles, and More [online]. Copyright © 2022 FMA Communications Canada, Inc. All rights reserved. [cit. 28.02.2022]. Dostupné z: <https://www.canadianmetalworking.com/canadianmetalworking/article/cuttingtools/making-holes-with-helical-interpolation>
- [7] KHALIL, Wisama a DOMBRE, Étienne. *Modeling identification and control of robots*. 1st ed. New York: Taylor and Francis, 2002. xv, 480 s. ISBN 1-56032-983-1.
- [8] KIRUBA, Shankar R., A workbench for motion control experiments using programmable automation controllers in industrial automation laboratory at Kongu Engineering College. *ResearchGate* [online]. únor 2017. [cit. 8.3.2022]. Dostupné z: https://www.researchgate.net/figure/Mechatronics-system-courtesy-Aerial-Euler-diagram-from-RPIs-website-describes-the_fig1_322941733
- [9] LINKEOVÁ, Ivana. *NURBS křivky: NeUniformní Racionální B-Spline křivky*. Vyd. 1. Praha: Nakladatelství ČVUT, 2007. 208 s. ISBN 978-80-01-03893-2.
- [10] ROSENBERG, Josef. *Kinematika. I. část*. 3. uprav. vyd. Plzeň: ZČU, 1992. 203 s. ISBN 80-7082-070-5.

- [11] SICILIANO, Bruno et al. *Robotics: modelling, planning and control*. London: Springer, ©2009. xxiv, 632 s. Advanced textbooks in control and signal processing. ISBN 978-1-84628-641-4.
- [12] SILVA, André Luis da. Attitude Determination For A Brazilian CubeSat Mission Using The Kalman Filter. *ResearchGate* [online]. prosinec 2017. [cit. 2.3.2022]. Dostupné z: https://www.researchgate.net/figure/Euler-angles-sequence-of-standard-rotations_fig2_3227119
- [13] ŠVEJDA, Martin. *Úvod do robotiky a mechatroniky* [přednášky]. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, říjen - prosinec 2021