



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA STROJNÍHO INŽENÝRSTVÍ**

FACULTY OF MECHANICAL ENGINEERING

**ÚSTAV MATEMATIKY**

INSTITUTE OF MATHEMATICS

**MATEMATICKÝ POPIS TRAJEKTORIE POHYBU VOZIDLA**

MATHEMATICAL DESCRIPTION OF VEHICLE MOTION TRAJECTORY

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. Jiří Lorenczyk**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**doc. Ing. Petr Porteš, Ph.D.**

**BRNO 2020**

# Zadání diplomové práce

Ústav: Ústav matematiky  
Student: **Bc. Jiří Lorenczyk**  
Studijní program: Aplikované vědy v inženýrství  
Studijní obor: Matematické inženýrství  
Vedoucí práce: **doc. Ing. Petr Porteš, Ph.D.**  
Akademický rok: 2019/20

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

## Matematický popis trajektorie pohybu vozidla

### Stručná charakteristika problematiky úkolu:

Práce je zaměřena na matematický popis trajektorie vozidla pro účely optimalizace jízdy automobilu po zadané vozovce. Předpokládá se využití znalostí konstruktivní a počítačová geometrie a diferenciální geometrie. Pro orientaci v tématu je rovněž nezbytná znalost optimalizačních metod.

### Cíle diplomové práce:

- Rešerše typů křivek použitelných pro popis trajektorie vozidla.
- Výběr typů křivek. Matematický popis musí poskytovat informace o délce křivky, směru tečny a o dalších kinematických veličinách nezbytných pro popis pohybu vozidla.
- Vhodná parametrizace křivek se zaměřením na optimalizaci průjezdu vozovkou se zadanou trajektorií.
- Sestavení modelu trajektorie v software Matlab.

### Seznam doporučené literatury:

VLK, F. Dynamika motorových vozidel: jízdní odpory, hnací charakteristika, brzdění, odpružení, řiditelnost, ovladatelnost, stabilita. Brno: Nakladatelství a vydavatelství Vlk, 2000. ISBN 80-238-52-3-6.

GILLESPIE, T. D. Fundamentals of vehicle dynamics. Warrendale, PA: Society of Automotive Engineers, c1992. ISBN 15-609-1199-9.

MARTIŠEK, D. Počítačová geometrie a grafika. Brno: VUT IUM, 2000. ISBN 80-214-1632-7.

DOUPOVEC, M. Diferenciální geometrie a tenzorový počet. Brno: PC-DIR Real, 1999. Učební texty vysokých škol. ISBN 80-214-1470-7.

BAZARAA, M. S., SHERALI, H. D. and SHETTY, C. M. Nonlinear programming: theory and algorithms. 3rd ed. Hoboken, N.J.: Wiley-Interscience, c2006. ISBN 978-0471486008.

LINKEOVÁ, I. Základy počítačového modelování křivek a ploch. Praha: České vysoké učení technické, 2008. ISBN 978-80-01-04011-9.

LINKEOVÁ, I. NURBS křivky: NeUniformní Racionální B-Spline křivky. Praha: Nakladatelství ČVUT, 2007. ISBN 978-80-01-03893-2.

PIEGL, L. A. a TILLER, W. The NURBS book. 2. vyd. New York: Springer, c1997. Monographs in visual communication. ISBN 35-406-1545-8.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2019/20

V Brně, dne

L. S.

---

prof. RNDr. Josef Šlapal, CSc.  
ředitel ústavu

---

doc. Ing. Jaroslav Katolický, Ph.D.  
děkan fakulty

## Abstrakt

Tato diplomová práce má za cíl najít křivky, které by mohly sloužit k popisu trajektorie vozidla po vozovce. Ukazuje se, že takové křivky by měly mít spojitou křivost, a z toho důvodu jsou blíže představeny tři typy křivek. Klotoidy, pomocí kterých lze plynule spojit rovinku s kruhovým obloukem, uzlové interpolační spline křivky 5. stupně, ty mají spojitou druhou derivaci, čímž je zajištěna i spojitá křivost a  $\eta$ -spline křivky, což jsou také interpolační polynomiální křivky 5. stupně, avšak jejich tvar je navíc určen vektorem parametrů  $\eta$ . Součástí této diplomové práce je i aplikace umožňující manuální vytvoření trajektorie složené ze spline křivek. .

## Summary

The goal of this thesis is to find types of curves which would allow for the construction of a path that could be traversed by a vehicle. It seems that a minimal constraint for such a path is the continuity of curve's curvature. This leads to a closer look at the three types of curves: Clothoids, which are able to smoothly connect straights with arcs of a constant curvature, interpolation quintic splines, which are  $C^2$  smooth in the interpolation nodes and  $\eta$ -splines, these belong to the family of quintic polynomial curves too, however, they are characterised by the vector of parameters  $\vec{\eta}$  which modifies the shape of the curve. The thesis is accompanied by an application allowing for manual construction of the path composed of spline curves.

## Klíčová slova

Trajektorie vozidla, B-spliny, NURBS, interpolační křivky, klotoida, Dubinsovy křivky,  $\eta$ -spline

## Keywords

Path planning, B-splines, NURBS, interpolation curves, clothoid, Dubins' paths,  $\eta$ -spline

LORENCZYK J. *Matematický popis trajektorie pohybu vozidla*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2020. 48 s. Vedoucí diplomové práce doc. Ing, Petr Porteš, Oh.D. .

Prohlašuji, že svou diplomovou práci na téma „Matematický popis trajektorie pohybu vozidla“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Brně dne 26.6. 2020.

.....

Jiří Lorenczyk

# Obsah

<b>1 Úvod</b>	<b>2</b>
<b>2 Popis trajektorie pomocí křivek</b>	<b>3</b>
2.1 Rovinky, části kružnic a klotoidy . . . . .	3
2.1.1 Dubinovy křivky . . . . .	3
2.1.2 Klotoidy . . . . .	3
2.2 Interpolační uzlové B-spliny . . . . .	6
2.3 $\eta$ -spliny . . . . .	7
<b>3 B-spline a NURBS křivky</b>	<b>10</b>
3.1 Vlastnosti aproximačních B-spline křivek . . . . .	10
3.2 Otevřená, uzavřená a ukotvená křivka . . . . .	14
3.2.1 Otevřená křivka . . . . .	14
3.2.2 Ukotvená křivka . . . . .	15
3.2.3 Uzavřená křivka . . . . .	17
3.3 Interpolační NURBS a B-spline křivky . . . . .	18
3.3.1 Prostá interpolace . . . . .	18
3.3.2 Uzlová interpolace . . . . .	20
<b>4 Aplikace pro modelování trajektorie</b>	<b>22</b>
4.1 Popis aplikace . . . . .	22
4.1.1 Uživatelský manuál . . . . .	22
4.1.2 Technická dokumentace . . . . .	26
4.2 Možná vylepšení aplikace a její nedostatky . . . . .	31
<b>5 Závěr</b>	<b>34</b>
5.1 Rešerše typů křivek . . . . .	34
5.2 Aplikace pro manuální modelování dráhy . . . . .	35
<b>A Parametrické křivky</b>	<b>37</b>
A.1 Definice křivky . . . . .	37
A.2 Vlastnosti křivek . . . . .	38
<b>B Skript pro výpočet <math>\eta</math>-spline křivky</b>	<b>42</b>
<b>Literatura</b>	<b>45</b>
<b>Seznam použitých zkratk a symbolů</b>	<b>47</b>
<b>Seznam obrázků</b>	<b>48</b>

# 1. Úvod

Nalezení vhodné trajektorie, kterou by mohlo následovat pro vozidlo popř. robota je problematika, která v posledních letech nabývá na popularitě, hlavně díky rozvoji autonomních aut a navyšování početního výkonu strojů tak, že jsou schopné zpracovávat a vyhodnocovat senzorické informace v reálném. Kromě tohoto oboru se o tuto disciplínu zajímají samozřejmě také závodní inženýři. Téměř všechna odvětví motorsportu jsou mimořádně nákladná a o úspěchu a porážce často rozhodují pouhé setiny, ne-li tisíce vteřiny, jako tomu bylo při kvalifikaci na (mezi fanoušky Formule 1) nechvalně proslulou velkou cenu Evropy roku 1997, kdy byl prvním třem jezdcům naměřen totožný čas a o pořadí na startu tak rozhodlo, kdo dokončil ono kvalifikační kolo na dráze v Jerezu jako první. Ironií však bylo, že na nejvyšší stupínek se v nedělním závodě nepostavil ani jeden z těchto tří jezdců. Z takovýchto důvodů jsou postupy, jak může pilot zlepšit svůj čas, byť i o nepatrný rozdíl, velice žádané.

Existuje několik metod, jak vypočíst optimální trajektorii vozidla. A jsou zřejmě dva směry, kterými se lze při řešení tohoto problému vydat. Prvním je, že se nejdříve vygeneruje trajektorie a poté se provede simulace, kdy model vozidla tuto křivku následuje, druhý směr se na tuto problematiku dívá tak, že jsou to vlastnosti vozidla samotného, které ovlivňují tvar takové křivky.

Jedna ze stěžejních prací, která využívá právě druhý pohled na dané téma je [5]. Autor v ní využil teorii optimálního řízení pro manévrování v minimálním čase a aplikoval ji k obdržení všech hodnot řídicích členů v čase, což mu dovolilo zkonstruovat časově optimální trajektorii.

Prvním směrem se vydali například autoři [17] nebo [4]. V [17] bylo využito ke konstrukci výsledné trajektorie spojení rovinek, klotoid a kruhových oblouků, tato metoda je blíže popsána v kapitole 2. Zatímco autoři [4] využili segmentaci okruhu a za použití genetického algoritmu našly hodnoty bodů trajektorie, tak aby křivka byla výsledkem minimalizace vůči vhodné kombinaci minimální průměrné křivosti a délky.

Tato práce se však nebude přímo zabývat časovou optimalizací vymodelované trajektorie. Pouze shrne poznatky o různých typech křivek vhodných k jejímu modelování a seznámí čtenáře s aplikací vyvinutou za účelem manuálního vytvoření validní trajektorie na daném okruhu. Podrobněji se bude zabývat Klotoidami, jejichž křivost je lineárně závislá na jejich délce, snadno modelovatelnými B-spline křivkami a  $\eta$ -spliny, což jsou polynomičké interpolační křivky pátého stupně, které umožňují interpolovat body se zadanou křivostí.

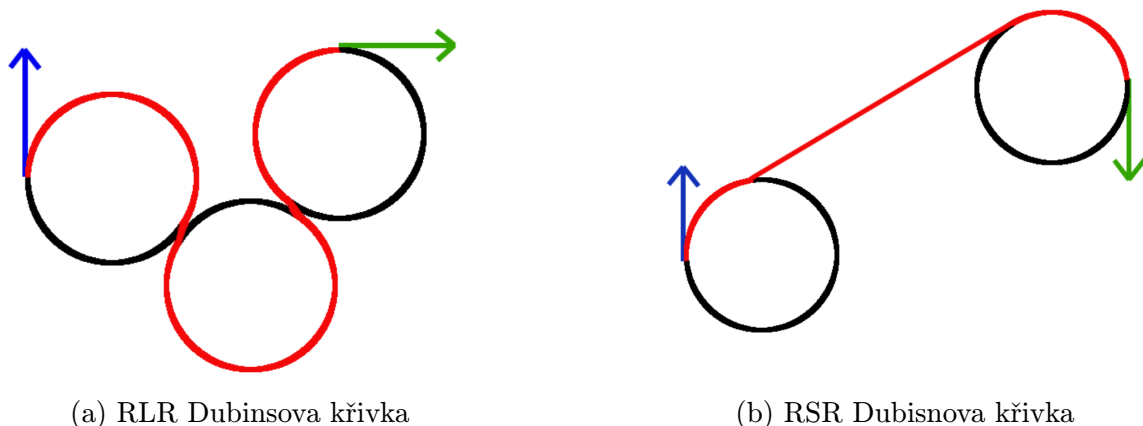
## 2. Popis trajektorie pomocí křivek

### 2.1. Rovinky, části kružnic a klotoidy

#### 2.1.1. Dubinovy křivky

Jedna zřejmě z prvních prací zabývajících se problematikou nalezení trajektorie vozidla pomocí úseků daných křivek byl článek L.E. Dubinse vydaný v roce 1957 v *American Journal of Mathematics* [7], ve kterém pomocí Pontrjaginova principu maxima ukázal, že pro vozidlo pohybující se pouze vpřed jednotkovou rychlostí se zadaným minimálním poloměrem zatáčení (tedy maximální křivostí) mezi dvěma body zadanými polohou a příslušným vektorem první derivace jsou křivky s nejkratší dráhou splňující tato omezení složeny ze segmentů přímek a kružnic. Vozidlo by se po takové křivce tedy pohybovalo pouze ve třech režimech, rovně, maximální zatočení vpravo a maximální zatočení a všechny trajektorie opsány pohybem takového vozidla by vznikly kombinací těchto režimů.

Dubins ve svém článku dále dokázal, že všechny nejkratší dráhy by se skládaly pouze z 6 možných kombinací těchto režimů. Pokud bychom označili zatočení vpravo písmenem **R**, zatočení vlevo **L** a přímočarou jízdu jako **S**, potom by se jednalo o tyto kombinace: **RSR**, **LSL**, **RSL**, **LSR**, **RLR**, **LRL**. Postup, jak tyto křivky konstruovat, je popsán v [9]. Tento model byl dále zkoumán a rozšiřován, např. o možnost zpětného chodu vozidla [15].



Obrázek 2.1: Příklad Dubinových křivek. Zdroj [9].

Při konstrukci trajektorie pouze za použití rovinek a kružnic však nedochází k napojení se spojitou křivostí (přímka má křivost rovnu nule a kružnice nenulové konstantě), což znamená, že vozidlo, které by přesně následovalo takovou trajektorii, by muselo v každém bodě nespojitosti křivosti zastavit, upravit své řízení, a až potom se znovu rozjet [8].

#### 2.1.2. Klotoidy

Z toho důvodu, např. Theodosis [17] obohatil tuto skupinu křivek o další - klotoidu. Klotoida je křivka, jejíž fundamentální vlastností je, že její křivost roste lineárně s její délkou. Díky tomu je z klotoidy ideální přechodnicová křivka, umožňuje totiž propojit



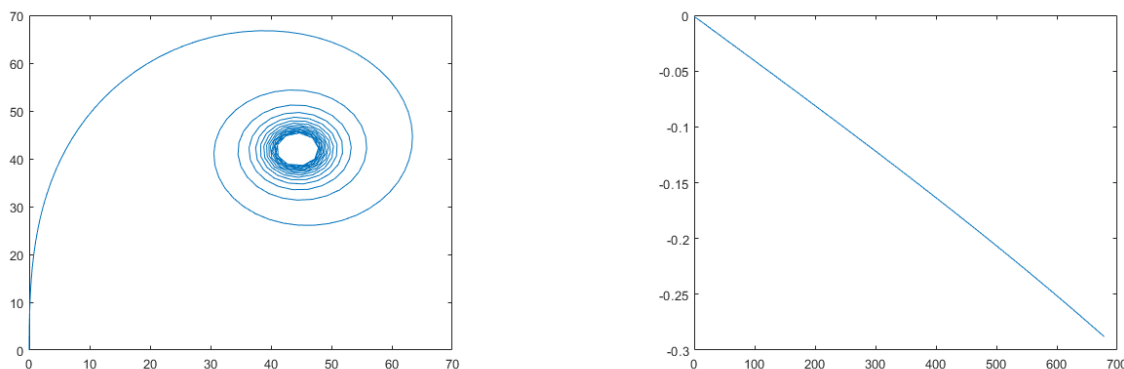
## 2.1. ROVINKY, ČÁSTI KRUŽNIC A KLOTOIDY

přímku s kružnicí a to tak, že výsledná křivka má spojitou křivost bez nějakých výchylek. Modelováním trajektorie pomocí napojování segmentů rovinek, klotoid a kružnic tedy lze vytvořit  $G^2$  spojitou křivku. Díky tomu si klotoidy našly uplatnění i v praxi. Využívají se zejména při napojování silnic nebo železnic [12].

Analyticky normovanou klotoidu popisují Fresnelovy integrály:

$$\begin{aligned} x(s) &= \int_0^s \cos t^2 dt \\ y(s) &= \int_0^s \sin t^2 dt \end{aligned} \tag{2.1}$$

Ty lze odvodit z věty A.1 dosazením  $\kappa(t) = 2t$ .



Obrázek 2.2: Část klotoidy a její křivost.

Ve své práci [17] autor nahlíží na problém modelování trajektorie vozidla z pohledu závodního jezdce, konkrétně se opírá o knihu italského závodníka a inženýra Piera Taruffiho „The Technique of Motor Racing [16].“ Na základě jeho poznatků potom vypracoval metodu, jak najít závodní stopu pomocí výše zmíněných křivek, tu ověřoval na okruhu ve Willows v Kalifornii. Vychází z toho, že na rovinkách pilot maximálně akceleruje, ještě na rovince před zatáčkou maximálně brzdí, při nájezdu do zatáčky potom zároveň zatáčí a brzdí na limitu trakce pneumatik, ve vrcholu zatáčky, potom maximálně zatáčí a na výjezdu ze zatáčky opět akceleruje na limitu trakce a zároveň zatáčí, dokud na rovince pilot nevyrovná řízení a znovu maximálně akceleruje. Tyto 4 fáze zhruba odpovídají geometrickému rozkladu na úsečky, klotoidy a kružnice.

Jeho metoda spočívá v tom, že jako počáteční odhad výsledné trajektorie uvažuje tzv. středovou křivku okruhu, tedy křivku, která má od obou okrajů tratě stejnou vzdálenost. Poté zanalyzuje profil křivosti okruhu a identifikuje oblasti s rovinkami a oblasti se zatáčkami. Rovinky jsou samozřejmě na úsecích s nulovou křivostí a zatáčky tam, kde je křivost nenulová. U rovinek pak zjistí jejich natočení vůči souřadnému systému a jejich počátek a konec. Poté pomocí kombinace bisekce a Newtonovy metody minimalizuje maximální křivost a délku trajektorie tak, aby v krajních bodech byla křivka tečná k rovinkám a v zatáčce se dotýkala vrcholu.

Trajektorie jednotlivých segmentů navazujících na sebe jsou potom popsány následujícími rovnicemi:

## Rovinka

$$\begin{aligned}x(s) &= x_0 + s \cos(\psi(s)) \\y(s) &= y_0 + s \sin(\psi(s)) \\ \psi(s) &= \psi_0,\end{aligned}\tag{2.2}$$

kde  $x_0$  a  $y_0$  je poloha počátečního bodu a  $\psi_0$  označuje směr rovinky. Segment končí dosažením konce rovinky, tedy  $s = L_1$ ,  $x(L_1) = x_1$ ,  $y(L_1) = y_1$  a  $\psi(L_1) = \psi_0 = \psi_1$

## Vstup do zatáčky

Tato fáze začíná koncem té předchozí a je tvořena klotoidou.

$$\begin{aligned}x(s) &= x_1 + \int_{L_1}^s \cos(\psi(t))dt \\y(s) &= y_1 + \int_{L_1}^s \sin(\psi(t))dt \\ \psi(s) &= \psi_1 + \frac{\kappa(s - L_1)^2}{2L_{C1}},\end{aligned}\tag{2.3}$$

kde  $x_1, y_1$  a  $\psi_1$  je počáteční pozice a směr klotoidy,  $\kappa$  je křivost kružnice, na kterou se bude klotoida dále napojovat a  $L_{C1}$  je délka oblouku klotoidy. Konec tohoto segmentu a začátek dalšího nastane pro  $s = L_2 = L_1 + L_{C1}$ .

## Kruhový oblouk

Na klotoidu se nyní napojí kruhový oblouk s konstantní křivostí  $\kappa$ .

$$\begin{aligned}x(s) &= x_2 + \int_{L_2}^s \cos(\psi(t))dt \\y(s) &= y_2 + \int_{L_2}^s \sin(\psi(t))dt \\ \psi(s) &= \psi_2 + \kappa(s - L_2),\end{aligned}\tag{2.4}$$

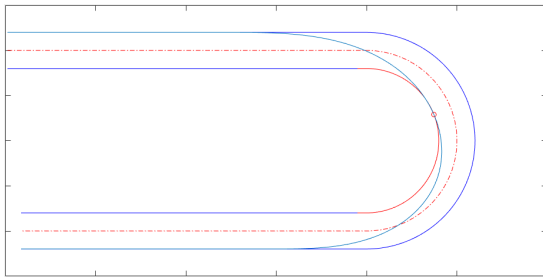
Pokud označíme délku kruhového oblouku jako  $L_a$ , potom konec tohoto úseku nastane pro  $s = L_3 = L_1 + L_{C1} + L_a$ .

## Výjezd ze zatáčky

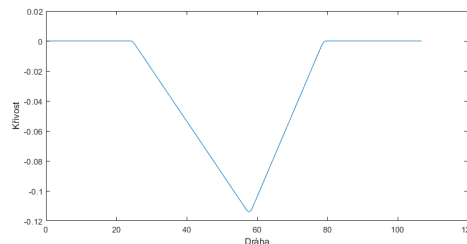
Po projetí oblouku s konstantní křivostí následuje výjezd ze zatáčky po klotoidě, která se rozvírá a hladce se napojí na následující rovinku.

$$\begin{aligned}x(s) &= x_3 + \int_{L_3}^s \cos(\psi(t))dt \\y(s) &= y_3 + \int_{L_3}^s \sin(\psi(t))dt \\ \psi(s) &= \psi_3 + \frac{\kappa L_{C2}}{2} - \frac{\kappa(L_4 - s)^2}{2L_{C2}},\end{aligned}\tag{2.5}$$

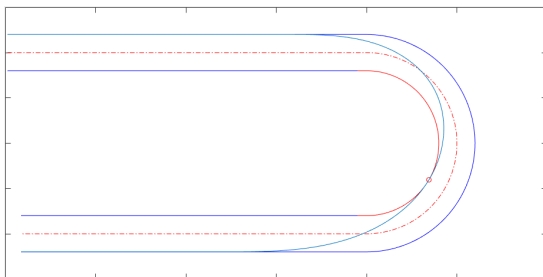
## 2.2. INTERPOLAČNÍ UZLOVÉ B-SPLINY



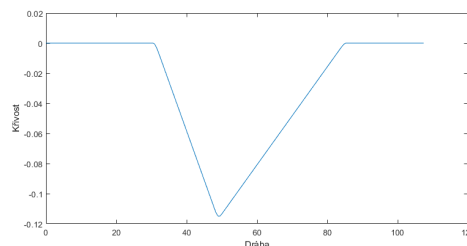
(a) Trajektorie s brzkým apexem. Podle postupu v kapitole 2.1.2



(b) Křivost trajektorie složené z klotoid a rovinek s brzkým apexem



(c) Trajektorie s pozdním apexem. Podle postupu v kapitole 2.1.2



(d) Křivost trajektorie složené z klotoid a rovinek s pozdním apexem.

Obrázek 2.3: Trajektorie zkonstruována z úseků rovinek a klotoid s nulovou délkou kruhového oblouku.

kde  $L_{C2}$  je délka této klotoidy, konec celého úseku nastává, pokud platí  $s = L_4 = L_1 + L_{C1} + L_a + L_{C2}$ .

Z výše uvedených rovnic tedy autor [17] obdržel parametry:  $x_0, y_0, \psi_0, L_1, L_{C1}, L_a, L_{C2}$  a  $\kappa$ , které určují výsledný tvar trajektorie. V kterém okamžiku dojde k dotyku s vrcholem zatáčky, potom ovlivňuje vztahy mezi relativními délkami oblouku segmentů. Tato volba má velký vliv při simulaci průjezdu zatáčky vozidlem, jelikož ovlivňuje momenty, kdy vozidlo brzdí a kdy již akceleruje pod plynem. Křivost kruhového oblouku určuje z rozdílu směrů mezi rovinkami

$$\kappa = \frac{2(\psi_4 - \psi_0)}{L_{C1} + L_a + L_{C2}}$$

Poté již následuje optimalizační část, ve které využívá metodu bisekce a Newtonovu metodu k určení hodnot zbylých parametrů tak, aby došlo k tečnému dotyku mezi koncem klotoidy na výjezdu ze zatáčky a následující rovinkou. Nevýhodou této metody je, že pokud chceme přesně určit vrchol zatáčky, kterého se má trajektorie dotknout, potom jeden ze vstupních parametrů takové optimalizační úlohy je délka počáteční rovinky a výstupem je bod napojení klotoidy a následující rovinky, což znamená, že nelze kontrolovat pozici těchto dvou okrajových bodů. To potom ztěžuje konstrukci trajektorie v úsecích, mezi kterými je pouze krátká rovinka, popř. inflexní bod.

## 2.2. Interpolační uzlové B-spliny

Toto jsou křivky, které jsou nakonec využity v aplikaci. Konkrétně jsou v ní použity po částech napojované Beziérovky 3. a 5. stupně. Z toho důvodu je kapitola 3 věnována základům teorie B-splínových křivek a vzorcům k jejich výpočtům.

## 2.3. $\eta$ -spliny

Další skupina křivek, která se nabízí k využití pro modelování trajektorie vozidla, jsou tzv.  $\eta$  kvitnické spliny. Ty umožňují nalézt takovou křivku, pro kterou budou platit následující okrajové podmínky:

$$\begin{aligned}
\mathbf{C}(0) &= \mathbf{Q}_A, \\
\mathbf{C}(1) &= \mathbf{Q}_B \\
\mathbf{e}(0) &= \begin{pmatrix} \cos \theta_A \\ \sin \theta_A \end{pmatrix} \\
\mathbf{e}(1) &= \begin{pmatrix} \cos \theta_B \\ \sin \theta_B \end{pmatrix} \\
\kappa(0) &= \kappa_A, \\
\kappa(1) &= \kappa_B
\end{aligned} \tag{2.6}$$

Kde  $\mathbf{e}(t)$  je jednotkový tečný vektor, určený vztahem  $\frac{\mathbf{C}'(t)}{\|\mathbf{C}'(t)\|}$ . To znamená, že při interpolaci lze zajistit spojitost křivosti napojované křivky v interpolačních bodech. Kvintická polynomiální křivka, která splňuje podmínky 2.6 je vyjádřena následujícími rovnicemi odvozenými v [14]:

$$\begin{aligned}
x(u) &= x_0 + x_1 u + x_2 u^2 + x_3 u^3 + x_4 u^4 + x_5 u^5, \\
y(u) &= y_0 + y_1 u + y_2 u^2 + y_3 u^3 + y_4 u^4 + y_5 u^5,
\end{aligned} \tag{2.7}$$

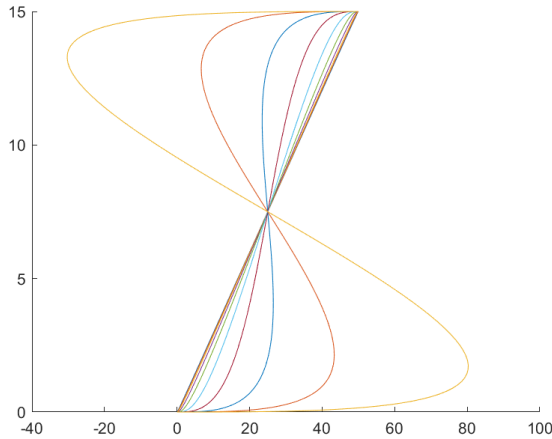
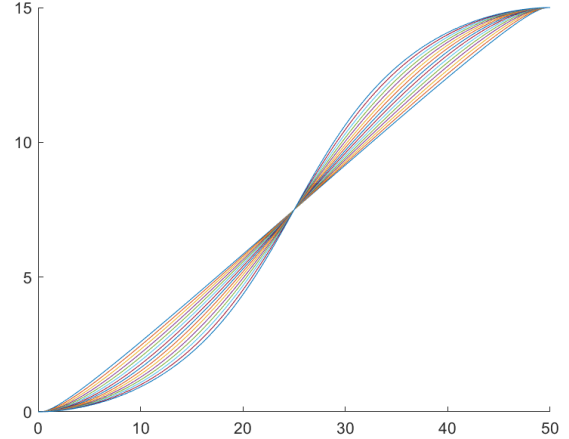
kde

$$\begin{aligned}
x_0 &= x_A \\
x_1 &= \eta_1 \cos \theta_A \\
x_2 &= \frac{1}{2}(\eta_3 \cos \theta_A - \eta_1^2 \kappa_A \sin \theta_A) \\
x_3 &= 10(x_B - x_A) - (6\eta_1 + \frac{3}{2}\eta_3) \cos \theta_A - (4\eta_2 - \frac{1}{2}\eta_4) \cos \theta_B \\
&\quad + \frac{3}{2}\eta_1^2 \kappa_A \sin \theta_A - \frac{1}{2}\eta_2^2 \kappa_B \sin \theta_B \\
x_4 &= -15(x_B - x_A) + (8\eta_1 + \frac{3}{2}\eta_3) \cos \theta_A + (7\eta_2 - \eta_4) \cos \theta_B \\
&\quad - \frac{3}{2}\eta_1^2 \kappa_A \sin \theta_A + \eta_2^2 \kappa_B \sin \theta_B \\
x_5 &= 6(x_B - x_A) - (3\eta_1 + \frac{1}{2}\eta_3) \cos \theta_A - (3\eta_2 - \frac{1}{2}\eta_4) \cos \theta_B \\
&\quad + \frac{1}{2}\eta_1^2 \kappa_A \sin \theta_A - \frac{1}{2}\eta_2^2 \kappa_B \sin \theta_B
\end{aligned} \tag{2.8}$$

$$\begin{aligned}
y_0 &= y_A \\
y_1 &= \eta_1 \sin \theta_A \\
y_2 &= \frac{1}{2}(\eta_3 \sin \theta_A + \eta_1^2 \kappa_A \cos \theta_A) \\
y_3 &= 10(y_B - y_A) - (6\eta_1 + \frac{3}{2}\eta_3) \sin \theta_A - (4\eta_2 - \frac{1}{2}\eta_4) \cos \theta_B \\
&\quad + \frac{3}{2}\eta_1^2 \kappa_A \cos \theta_A + \frac{1}{2}\eta_2^2 \kappa_B \cos \theta_B \\
y_4 &= -15(y_B - y_A) + (8\eta_1 + \frac{3}{2}\eta_3) \sin \theta_A + (7\eta_2 - \eta_4) \sin \theta_B \\
&\quad + \frac{3}{2}\eta_1^2 \kappa_A \cos \theta_A - \eta_2^2 \kappa_B \cos \theta_B \\
y_5 &= 6(y_B - y_A) - (3\eta_1 + \frac{1}{2}\eta_3) \sin \theta_A - (3\eta_2 - \frac{1}{2}\eta_4) \sin \theta_B \\
&\quad - \frac{1}{2}\eta_1^2 \kappa_A \cos \theta_A + \frac{1}{2}\eta_2^2 \kappa_B \cos \theta_B,
\end{aligned}$$

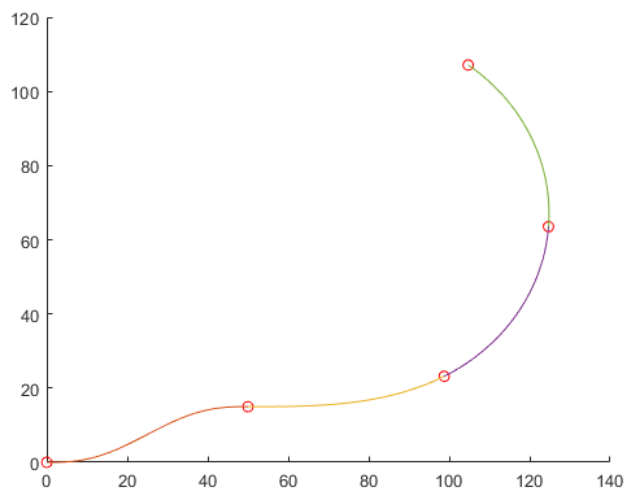
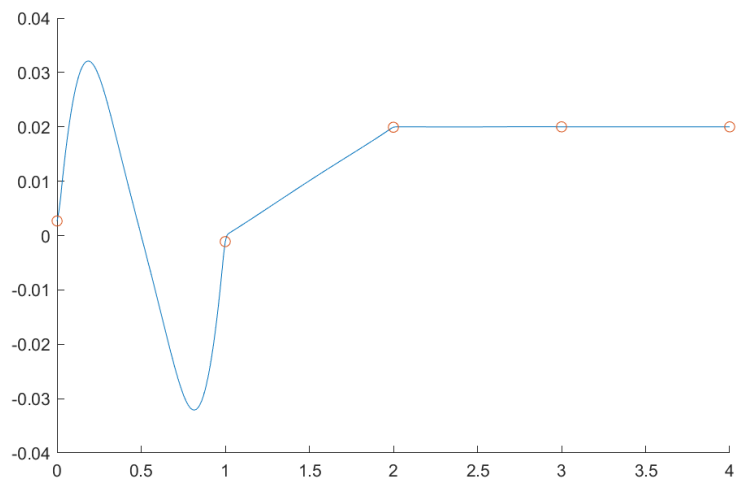
kde parametry  $\eta_1, \eta_2, \eta_3$  a  $\eta_4$  lze uspořádat do vektoru  $\vec{\eta}$ . Potom křivku zadanou rovnicemi 2.7 a 2.8 můžeme parametrizovat pomocí parametrů  $u$  a  $\vec{\eta}$ , tedy  $\mathbf{C}(t, \vec{\eta}) = (x(u, \vec{\eta}), y(u, \vec{\eta}))$  a nazýváme ji  $\eta$ -spline.

Autoři [14] dále dokázali tvrzení, že pro libovolné interpolační data 2.6, bude parametrická křivka  $\mathbf{C}(t, \vec{\eta})$  splňovat tyto okrajové podmínky a to pro všechny  $\vec{\eta} \in \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R} \times \mathbb{R}$ . A naopak pro libovolnou kvintickou polynomiální křivku  $\mathbf{C}(u)$ , která splňuje okrajové podmínky 2.6 a  $(C')'(0) \neq 0$ ,  $(C')'(1) \neq 0$ , existuje vektor parametrů  $\vec{\eta} \in \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R} \times \mathbb{R}$  takový, že křivku  $\mathbf{C}(u)$  lze vyjádřit jako  $\mathbf{C}(u, \vec{\eta})$ .

(a) Lineární změny parametrů  $\eta_1$  a  $\eta_2$ .(b) Lineární změny parametrů  $\eta_3$  a  $\eta_4$ .

Obrázek 2.4: Vliv změny hodnot parametrů na tvar křivky.

Autoři ve svém článku ukázali, že parametry  $\eta_1$  a  $\eta_2$  souvisí s velikostí rychlosti v interpolačních bodech a parametry  $\eta_3$  a  $\eta_4$  se zakřivením křivky, avšak bližší informace o vlivu těchto parametrů na tvar křivky, popř. nějaká doporučení či tabulkové hodnoty se mi dohledat nepodařilo. Samotní autoři byli ve svém článku udivení, že pro stejnou sadu parametrů  $\vec{\eta}$  se jim podařilo aproximovat kružnici i klotoidu. Zdá se tedy, že křivka získána tímto způsobem by musela být podrobena buď optimalizaci, nebo by byly parametry volitelné manuálně, popř. by byly zadány pevně. Tento typ křivek by si zřejmě zasloužil podrobnější prozkoumání, avšak to by již bylo zřejmě nad rámec této práce. V příloze B,

(a) 4 interpolační  $\eta$ -spline křivky(b) Křivost uvedených  $\eta$ -spline křivek.Obrázek 2.5: Interpolační  $\eta$ -spliny podle dat z [14]

alespoň uvádím skript, který na základě dat poskytnutých v [14] vygeneroval grafy 2.4 a 2.5, který by mohl sloužit jako malý odrazový můstek pro další výzkum.

### 3. B-spline a NURBS křivky

NURBS a B-spline křivky jsou po částech definované křivky, kde na jednotlivých segmentech jsou tyto křivky zkonstruovány pomocí lineární kombinace bazových funkcí. Tyto segmenty jsou na sebe napojeny s  $C^{p-1}$  spojitostí, kde  $p$  je stupeň křivky. Písmeno B v názvu těchto křivek je tedy zkratkou pro ony bazové funkce. NURBS křivky jsou zobecněním B-spline křivek a to ve dvou ohledech. Prvním zobecněním je, že délka intervalů, na kterých jsou definovány B-spline křivky, je pro všechny segmenty stejná, tyto křivky jsou tedy uniformní, u křivek NURBS tato vlastnost nemusí být nutně splněna, proto jsou neuniformní. Druhé zobecnění se pak týká tvaru bazových funkcí, B-spline křivky obdržíme jako lineární kombinaci polynomů, kdežto NURBS jsou konstruovány lineární kombinací racionálních funkcí, ty obdržíme tak, že každému řídicímu bodu křivky přiřadíme jistou váhu, pomocí které udáváme významnost daných řídicích bodů na tvar křivky ve srovnání s vlivem ostatních bodů.

Podle typů zadaných dat rozlišujeme dva přístupy ke konstrukci křivky zadané pomocí posloupností bodů a to aproximaci a interpolaci.

Při aproximaci křivka zadaná řídicími body  $P_0, P_1, \dots, P_n$  nemusí těmito body procházet, zatímco interpolační křivka zadaná definičními body  $Q_0, Q_1, \dots, Q_n$  těmito body procházet musí.

Nejdříve si ukážeme konstrukci aproximačních křivek a později s jejich pomocí i konstrukci křivek interpolačních.

Následující definice jsou převzaty z [10] a [11]

#### 3.1. Vlastnosti aproximačních B-spline křivek

Jednotlivé segmenty B-spline resp. NURBS křivky uspořádáváme do tzv. uzlového vektoru.

**Definice 3.1** (Uzlový vektor). *Nechť  $U$  je neklesající posloupnost  $(m+1)$  reálných čísel*

$$u_0 \leq u_1 \leq u_2 \leq \dots \leq u_m,$$

*Potom čísla  $u_i$ ,  $i = 0, \dots, m$  nazýváme uzly, množinu  $U = (u_i)_{i=0}^m = (u_0, u_1, u_2, \dots, u_m)$  nazýváme uzlovým vektorem a interval  $[u_i, u_{i+1})$  nazveme  $i$ -tou uzlovou roztečí (uzlovým intervalem).*

*$K$ -násobným uzlem nazveme uzel, pro který platí*

$$u_i = u_{i+1} = \dots = u_{i+k-1}, \quad k > 1.$$

*Jednoduchý uzel je uzel, pro který platí*

$$u_i \neq u_k, \quad \text{pro } i \neq k.$$

*Jednoduchý nebo  $k$ -násobný uzel na začátku a na konci uzlového vektoru nazveme krajní, ostatní uzly se nazývají vnitřní.*

*Pokud mají všechny uzlové intervaly stejnou délku,*

$$u_{i+1} - u_i = \Delta u = \text{konst.}, \quad i = 0, \dots, m-1,$$

*pak řekneme, že uzlový vektor je uniformní, v ostatních případech je uzlový vektor neuniformní. Výjimkou jsou krajní uzly, které mohou být násobné.*

### 3.1. VLASTNOSTI APROXIMAČNÍCH B-SPLINE KŘIVEK

**Definice 3.2** (B-spline báze funkce). *Nechť je dán uzlový vektor  $U = (u_i)_{i=1}^m$ ,  $u_0 = a, u_m = b$ ,  $a, b \in \mathbb{R}$ , potom báze funkce  $N_{i,p}(u)$ ,  $i = 0, \dots, n$ ,  $u \in [a, b]$ , stupně  $p$  definujeme rekurentním vzorcem*

$$\begin{aligned} N_{i,0} &= \begin{cases} 1, & u \in [u_i, u_{i+1}), \\ 0, & u \notin [u_i, u_{i+1}) \text{ nebo na intervalu nulové délky,} \end{cases} \\ N_{i,k} &= \frac{u - u_i}{u_{i+k} - u_i} N_{i,k-1}(u) + \frac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} N_{i+1,k-1}(u), \quad k = 1, \dots, p. \end{aligned} \quad (3.1)$$

Výrazy  $\frac{0}{0}$ , které obdržíme pro uzlové intervaly nulové délky, se položí rovny nule.

**Definice 3.3** (B-spline křivka). *Nechť je dán řídící polygon určený řídícími body  $(\mathbf{P})_{i=0}^n = (P_0, P_1, \dots, P_n)$ , uzlový vektor  $U = (u_i)_{i=0}^m$ ,  $u_0 = a, u_m = b$ ,  $a, b \in \mathbb{R}$  a stupeň křivky  $p$ . Potom je B-spline křivka  $\mathbf{C}(u)$  stupně  $p$  dána následující vektorovou rovnicí*

$$\mathbf{C}(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i, \quad u \in [a, b], \quad (3.2)$$

kde  $N_{i,p}(u)$ ,  $i = 0, \dots, n$  jsou B-spline báze funkce (3.1).

**Definice 3.4** (Váhy řídících bodů). *Nechť je dán řídící polygon  $(\mathbf{P}_i)_{i=0}^n$  a posloupnost nezáporných reálných čísel  $W = (w_0, w_1, \dots, w_n) = (w_i)_{i=0}^n$  přiřazených odpovídajícím řídícím bodům,  $\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_n$ . Potom čísla  $w_0, w_1, \dots, w_n$  nazveme váhami řídících bodů.*

**Definice 3.5** (Racionální báze funkce). *Nechť jsou dány váhy řídících bodů  $(w_i)_{i=0}^n$ . Potom racionální báze funkce  $R_{i,p}(u)$ ,  $i = 0, 1, \dots, n$ ,  $u \in [a, b]$  stupně  $p$  definujeme jako*

$$R_{i,p}(u) = \frac{N_{i,p}(u) w_i}{\sum_{j=0}^n N_{j,p}(u) w_j}, \quad (3.3)$$

kde  $N_{i,p}(u)$ ,  $i = 0, 1, \dots, n$  jsou B-spline báze funkce (3.1).

**Definice 3.6** (NURBS křivka). *Nechť je dán řídící polygon  $(P_i)_{i=0}^n$ , váhy odpovídajících řídících bodů  $(w_i)_{i=0}^n$ , uzlový vektor  $U = (u_i)_{i=0}^m$ ,  $u_0 = a, u_m = b$ ,  $a, b \in \mathbb{R}$  a stupeň křivky  $p$ . Potom je NURBS křivka  $\mathbf{C}(u)$  stupně  $p$  dána následující vektorovou rovnicí*

$$\mathbf{C}(u) = \sum_{i=0}^n R_{i,p}(u) \mathbf{P}_i, \quad u \in [a, b], \quad (3.4)$$

kde  $R_{i,p}$ ,  $i = 0, 1, \dots, n$  jsou racionální báze funkce (3.3).

**Věta 3.1** (Počet uzlových roztečí uzlového vektoru). *Pro  $n + 1$  řídících bodů a stupeň  $p$  je počet  $m$  uzlových roztečí uzlového vektoru roven*

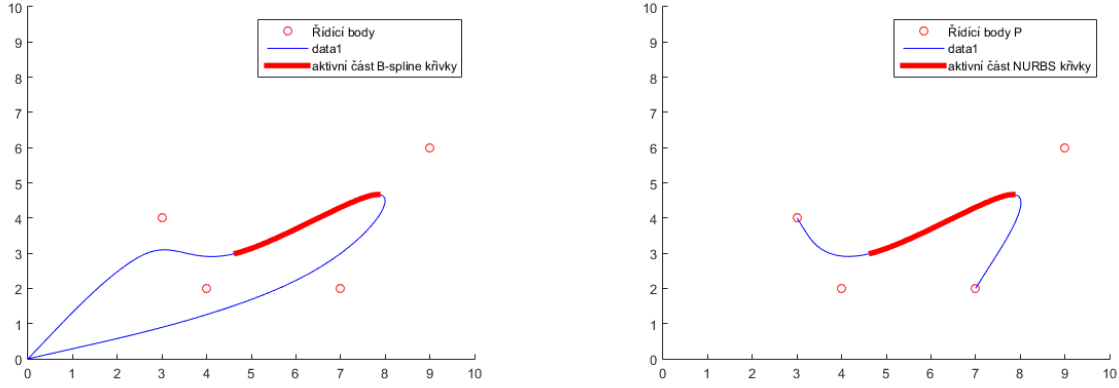
$$m = n + p + 1. \quad (3.5)$$

*Důkaz viz [10].*

Nyní si uvedeme některé významné vlastnosti báze funkcí a B-spline resp. NURBS křivek.



### 3.1. VLASTNOSTI APROXIMAČNÍCH B-SPLINE KŘÍVEK



(a) Aproximační B-Spline křivka

(b) Aproximační NURBS křivka

Obrázek 3.1: Aproximační B-spline a NURBS křivky, aktivní části křivek jsou vyznačeny červeně.

*Vlastnost 3.1* (Lokální podpora). B-spline básová funkce  $N_{i,p}(u)$ , resp. racionální básová funkce  $R_{i,p}(u)$  je nenulová na intervalu  $[u_i, u_{i+p+1})$ . Tato vlastnost vyplývá z definice básových funkcí, kde se na konstrukci básové funkce stupně  $p$  využívá lineární kombinace dvou básových funkcí stupně  $p - 1$ .

Tato vlastnost vyjadřuje fakt, že pokud změním polohu řídicího bodu  $\mathbf{P}_i$ , změní se tvar křivky pouze na segmentu odpovídajícímu intervalu  $[u_i, u_{i+p+1})$ , mimo tento interval se tvar křivky zachová.

*Vlastnost 3.2.* Na libovolné uzlové rozteči  $[u_i, u_{i+1}]$  je nejvýše  $p + 1$  B-spline, resp. racionálních básových funkcí  $p$ -tého stupně nenulových.

Tuto vlastnost získáme z faktu, že pro B-spline básově funkce nultého stupně platí

$$N_{i,0}(u) = \begin{cases} 1, & u \in [u_i, u_{i+1}), \\ 0, & \text{jinde} \end{cases}$$

Tyto dvě vlastnosti vyjadřují tzv. plnou podporu křivky básovými funkcemi na uzlových roztečích, na kterých je právě  $p + 1$  básových funkcí nenulových. Tu část křivky, která je plně podporována básovými funkcemi, tj. je zde  $p + 1$  básových funkcí nenulových, nazveme aktivní část křivky. Příslušné segmenty (resp. uzlové rozteče, resp. uzly) křivky nazýváme aktivní segmenty (resp. uzlové rozteče, resp. uzly). Segmenty křivky, na jejichž uzlových roztečích je méně než  $p + 1$  básových funkcí nenulových, nejsou svými básovými funkcemi plně podporovány. Tuto část křivky tedy budeme ignorovat a nazveme ji stejně jako odpovídající segmenty, uzlové rozteče a uzly pasivní. [10]

*Vlastnost 3.3.* V případě, že jsou všechny váhy rovny stejné kladné konstantě, budou racionální básově funkce na aktivních uzlových roztečích totožné s B-spline básovými funkcemi, B-spline básově funkce jsou tedy speciálním případem racionálních básových funkcí a B-spline křivka je na aktivních uzlových roztečích totožná s NURBS křivkou. Na pasivních částech se však B-spline a NURBS křivky liší.

*Vlastnost 3.4.* Pro všechna  $i = 0, 1, \dots, n, p, n$  platí, že

$$\begin{aligned} N_{i,p}(u) &\geq 0, \quad \forall u \in [a, b], \\ R_{i,p}(u) &\geq 0, \quad \forall u \in [a, b]. \end{aligned}$$

### 3.1. VLASTNOSTI APROXIMAČNÍCH B-SPLINE KŘIVEK

B-spline resp. racionální báze funkce jsou tedy totálně pozitivní.

*Vlastnost 3.5.* Pokud je uzlová rozteč  $[u_i, u_{i+1})$  aktivní, potom pro B-spline báze funkce stupně  $p$  platí

$$\sum_{j=i-p}^p N_{j,p} = 1.$$

Na pasivních uzlových roztečích pak součet B-spline báze funkcí není roven jedné. Součet racionálních báze funkcí je díky váhám roven jedné jak na aktivních, tak na pasivních segmentech křivky.

*Vlastnost 3.6.* B-spline křivka leží celá v konvexním obalu svého řídicího polygonu. Bod B-spline křivky  $\mathbf{C}(u_0)$ ,  $u_0 \in [u_i, u_{i+1})$ , leží v konvexním obalu dílčího řídicího polygonu  $(\mathbf{P})_{j=i-p}^p$ .

*Vlastnost 3.7.* B-spline křivka je invariantní vůči afinním transformacím.

Poslední vlastnost znamená, že ať již transformujeme výslednou křivku, nebo nejdříve řídicí body a až poté křivku vygenerujeme, bude výsledek stejný, tato vlastnost je velice praktická, jelikož šetří výpočetní čas.

*Vlastnost 3.8.* Uvnitř každé uzlové rozteče existují všechny derivace B-spline i racionálních báze funkcí. V uzlech jsou potom báze funkce  $p - k$  spojitě diferencovatelné, kde  $k$  je násobnost uzlu.

*Vlastnost 3.9.*  $K$ -tou derivaci B-spline báze funkce vypočteme vztahem

$$N_{i,p}^{(k)}(u) = \frac{p}{p-k} \left[ \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}^{(k)}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}^{(k)}(u) \right], \quad k = 0, \dots, p-1 \quad (3.6)$$

*Vlastnost 3.10* ( $k$ -tá derivace B-spline křivky). Označme řídicí bod  $\mathbf{P}_i = \mathbf{P}_i^{(0)}$  a B-spline křivku jako

$$\mathbf{C}(u) = \mathbf{C}^0(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i^{(0)},$$

potom lze vyjádřit  $k$ -tou derivaci B-spline křivky  $\mathbf{C}^{(k)}(u)$  jako

$$\mathbf{C}^{(k)}(u) = \sum_{i=0}^{n-k} N_{i,p-k}(u) \mathbf{P}_i^{(k)}, \quad (3.7)$$

kde

$$\mathbf{P}_i^{(k)} = \begin{cases} \mathbf{P}_i, & \text{pro } k = 0, \\ \frac{p-k+1}{u_{i+p+1} - u_{i+k}} \left( \mathbf{P}_{i+1}^{(k-1)} - \mathbf{P}_i^{(k-1)} \right), & \text{pro } k > 0, \end{cases}$$

$N_{i,p-k}(u)$  jsou pak B-spline báze funkce stupně  $p - k$  definované na uzlovém vektoru  $U^k$ , což je původní uzlový vektor  $U$  zkrácen o  $k$  počátečních a koncových uzlů.

*Vlastnost 3.11* ( $k$ -tá derivace NURBS křivky). Nurbs křivka  $\mathbf{C}(u)$  je zadána jako

$$\mathbf{C}(u) = \frac{\sum_{i=0}^n N_{i,p}(u) w_i \mathbf{P}_i}{\sum_{i=0}^n N_{i,p}(u) w_i}$$

### 3.2. OTEVŘENÁ, UZAVŘENÁ A UKOTVENÁ KŘIVKA

označme

$$\mathbf{S}(u) = \sum_{i=0}^n N_{i,p}(u) w_i \mathbf{P}_i$$

$$w(u) = \sum_{i=0}^n N_{i,p}(u) w_i,$$

tedy

$$\mathbf{C}(u) = \frac{\mathbf{S}(u)}{w(u)}.$$

$k$ -tou derivaci  $\mathbf{C}^{(k)}(u)$  lze potom vyjádřit rekurentním vzorcem

$$\mathbf{C}^{(k)}(u) = \frac{1}{w(u)} \left[ \mathbf{S}^{(k)}(u) - \sum_{i=0}^k \binom{k}{i} w^{(i)}(u) \mathbf{C}^{(k-i)}(u) \right], \quad (3.8)$$

kde  $\mathbf{S}^{(k)}(u)$  a  $w^{(i)}(u)$  spočteme podle (3.7).

K odvození těchto vlastností viz [13].

## 3.2. Otevřená, uzavřená a ukotvená křivka

B-spline a NURBS křivky můžeme podle vztahu mezi tvarem uzlového vektoru a řídícího polygonu rozdělit na tři typy: otevřenou, uzavřenou a ukotvenou.

### 3.2.1. Otevřená křivka

Pro otevřenou křivku není podmínka plné podpory báze funkcí zajištěna v celém rozsahu parametru  $u \in [a, b]$ , ale pouze na intervalu  $[u_p, u_{m-p}]$ . Pasivní segmenty křivky z pasivních roztečí  $[a, u_p]$  a  $[u_{m-p}, b]$  se ignorují. Oborem parametrizace je pak interval  $[u_p, u_{m-p}]$ . Křivka zadaná takovým uzlovým vektorem

$$U = (a = u_0 \leq u_1 \leq \dots \leq u_m = b) \quad (3.9)$$

se pak nazývá otevřená (anglicky open), neboť neprochází žádným svým řídícím bodem.

Konkrétním příkladem otevřené křivky je například Coonsova kubika.

**Definice 3.7.** *Coonsova kubika*

*Nechť jsou dány řídící body  $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2$  a  $\mathbf{P}_3$  a k nim příslušné váhy  $w_0 = w_1 = w_2 = w_3 = C > 0$ , uzlový vektor  $U = (-3, -2, -1, 0, 1, 2, 3, 4)$  a stupeň  $p = 3$ . Aktivní část NURBS křivky potom leží na intervalu  $u \in [0, 1]$  a je popsána vektorovou rovnicí*

$$\mathbf{C}(u) = \sum_{i=0}^3 C_i(u) \mathbf{P}_i, \quad u \in [0, 1], \quad (3.10)$$

### 3.2. OTEVŘENÁ, UZAVŘENÁ A UKOTVENÁ KŘIVKA

kde

$$\begin{aligned} C_0(u) &= \frac{1}{6}(1-u)^3, \\ C_1(u) &= \frac{1}{6}(3u^3 - 6u^2 + 4), \\ C_2(u) &= \frac{1}{6}(-3u^3 + 3u^2 + 3u + 1), \\ C_3(u) &= \frac{1}{6}u^3. \end{aligned} \tag{3.11}$$

Křivku (3.10) nazýváme Coonsovou kubikou a funkce (3.11) Coonsovými polynomy.

Coonsova kubika je otevřená NURBS křivka tvořená jediným aktivním segmentem, lze však sestavit NURBS resp. B-spline křivku tvořenou více aktivními segmenty. Taková křivka se pak nazývá segmentovaná. Příkladem otevřené segmentované NURBS křivky je Coonsův kubický B-spline.

**Definice 3.8.** Coonsův kubický B-spline Nechť je dán řídící polygon  $\{\mathbf{P}_i\}_{i=0}^n, n > 3$ , stupeň  $p=3$ , váhy  $w_i \equiv C, C > 0$ , pro  $\forall i = 0, 1, \dots, n$  a uniformní uzlový vektor

$$U = (u_i)_{i=0}^m = (-3, -2, \dots, m-3).$$

Aktivní část NURBS křivky (3.4) potom leží na intervalu  $u \in [u_3, u_{m-3}]$  a nazývá se Coonsův kubický B-spline.

#### 3.2.2. Ukotvená křivka

Ukotvenou křivku obdržíme, pokud bychom na NURBS resp. B-spline křivku vznesli přirozený požadavek, aby se aktivní segment uzlového vektoru, kde je křivka plně podpořována, rovnal celému intervalu, na kterém je uzlový vektor definován. Toho dosáhneme jednoduše tím, že položíme prvních  $p+1$  uzlů rovných počátku intervalu a posledních  $p+1$  uzlů rovných konci intervalu. Dostaneme tak uzlový vektor ve tvaru

$$U = (a = u_0 = u_1 = \dots = u_p \leq u_{p+1} \leq \dots \leq u_{m-p-1} \leq u_{m-p} = \dots = u_m = b). \tag{3.12}$$

Pasivní segmenty křivky se tak zredukovaly pouze na body  $\mathbf{P}_0$  a  $\mathbf{P}_n$ , báze funkce se pro tyto body rovnají jedné, a proto je křivka interpoluje, ostatní body řídícího polynomu však pouze aproximuje, odtud název pro tento typ křivky (v angličtině clamped). Nejznámějším příkladem ukotvené křivky je Bézierova křivka.

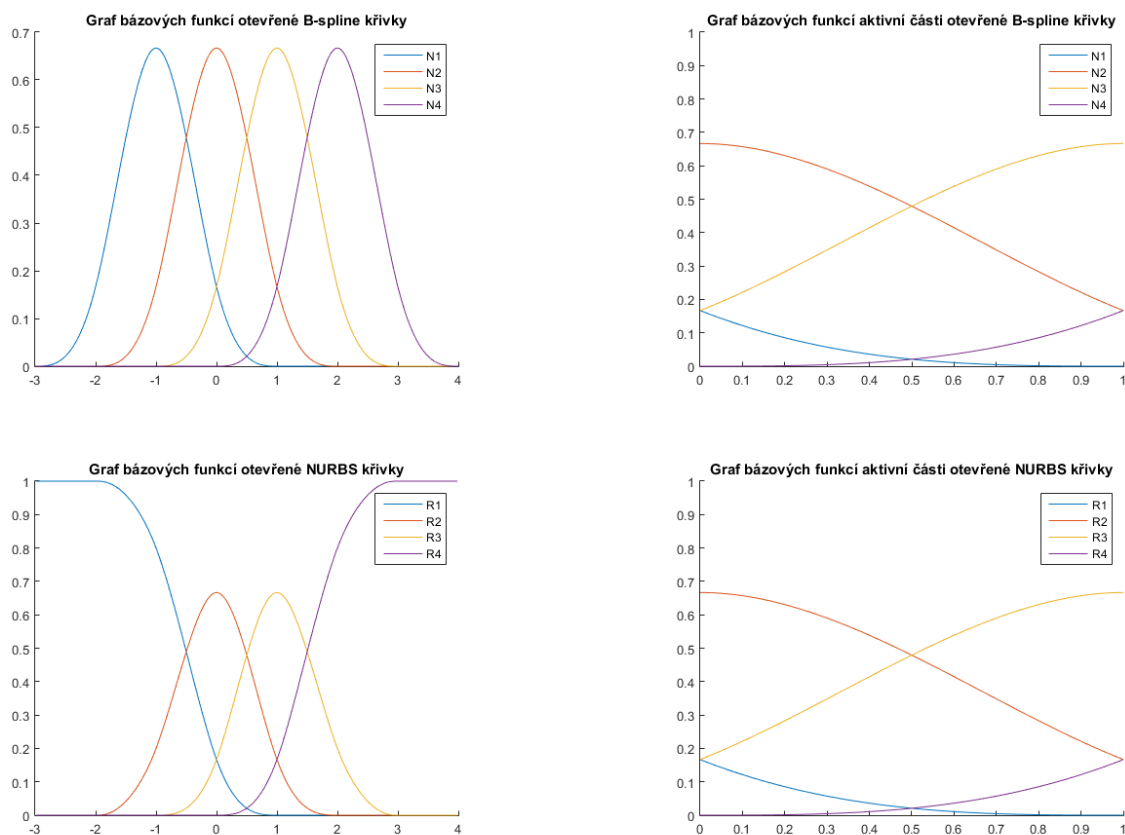
**Věta 3.2.** Nechť je dán řídící polygon  $(\mathbf{P}_i)_{i=0}^n, n \geq 0$ , váhy  $w_i \equiv C, C > 0$ , pro  $\forall i = 0, 1, 2, \dots, n$ , uzlový vektor

$$U = (u_0, \dots, u_p, u_{m-p}, \dots, u_m) = (0, \dots, 0, 1, \dots, 1)$$

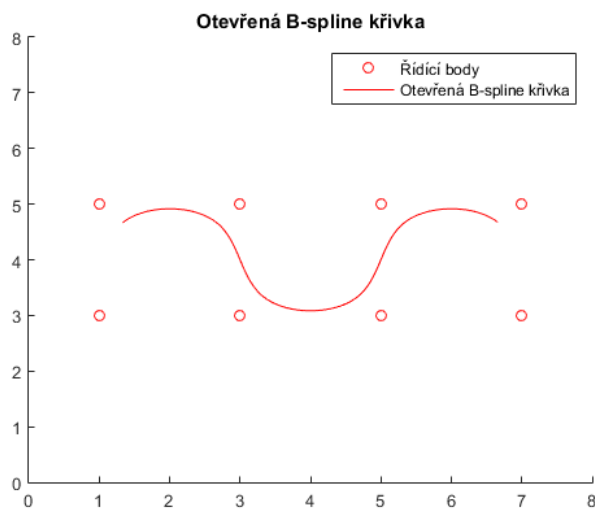
a stupeň  $p = n$ . Potom NURBS křivku (3.4) na intervalu  $u \in [0, 1]$  nazveme Bézierovou křivkou. Ta je popsána vektorovou rovnicí

$$\mathbf{C}(u) = \sum_{i=0}^n B_{i,n}(u) \mathbf{P}_i, \quad u \in [0, 1], \tag{3.13}$$

### 3.2. OTEVŘENÁ, UZAVŘENÁ A UKOTVENÁ KŘIVKA



Obrázek 3.2: Grafy báзовých funkcí otevřených B-spline a NURBS křivek, vlevo celý interval parametru, vpravo aktivní část báзовých funkcí



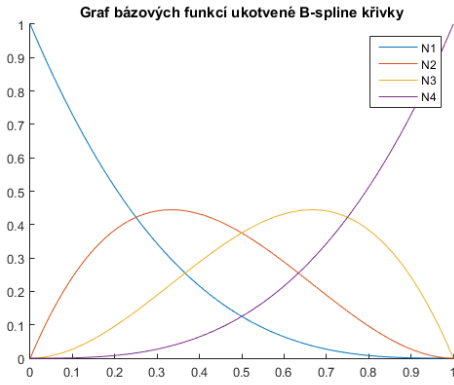
Obrázek 3.3: Otevřená B-spline křivka.

,kde

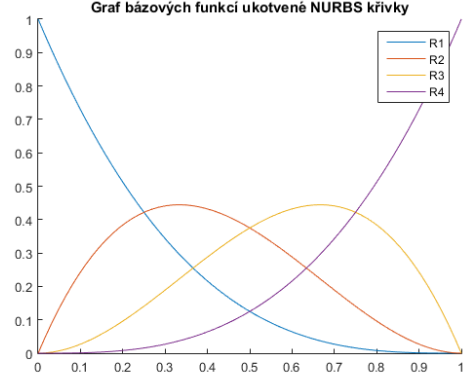
$$B_{i,n} = \binom{n}{i} u^i (1-u)^{n-1}, \quad i = 0, 1, \dots, n, \quad (3.14)$$

jsou Bernsteinovy polynomy  $n$ -tého stupně.

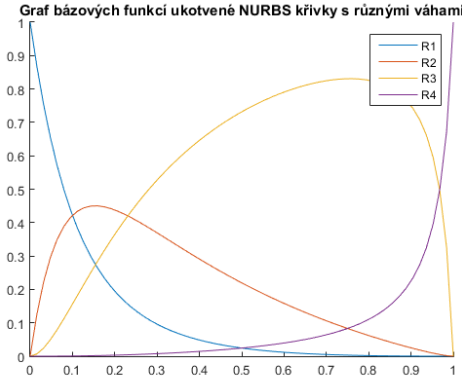
### 3.2. OTEVŘENÁ, UZAVŘENÁ A UKOTVENÁ KŘIVKA



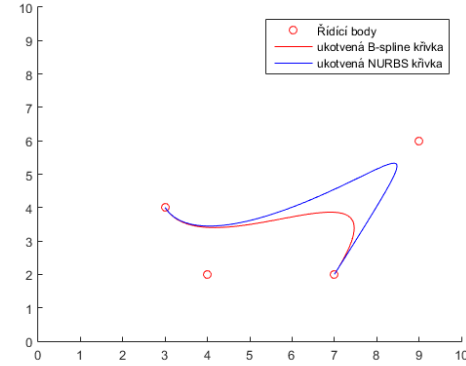
(a) Báзовé funkce ukotvené B-spline křivky.



(b) Báзовé funkce ukotvené NURBS křivky, váhy rovny konstantě.



(c) Báзовé funkce ukotvené NURBS křivky s váhami [1, 3, 10, 1].



(d) Porovnání ukotvené B-spline křivky a NURBS křivky s různými váhami.

Obrázek 3.4: Ukotvené B-spline a NURBS křivky

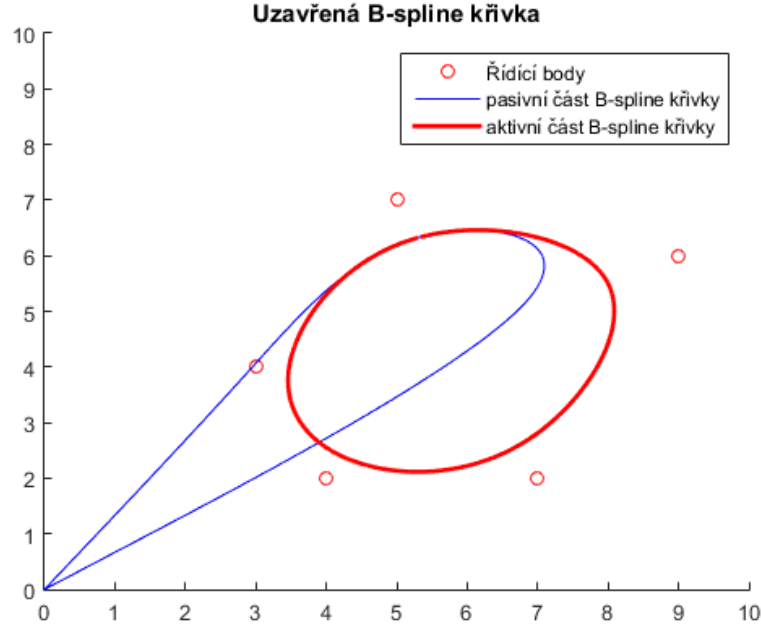
Stejně jako Coonsovu kubiku, lze i Bézierovu křivku rozšířit na segmentovanou ukotvenou B-spline (resp. NURBS) křivku tím, že zvýšíme počet ramen řídicího polygonu a zároveň zachováme stupeň křivky, přičemž náležitě upravíme uniformní uzlový vektor.

#### 3.2.3. Uzavřená křivka

Uzavřená křivka je taková, že se její počáteční bod rovná koncovému bodu křivky, přičemž zde zůstává zachována spojitost, kterou požadujeme podél celé křivky. Takové napojení lze docílit tím, že na konci řídicího polygonu zopakujeme tolik řídicích bodů, jaký je stupeň křivky, včetně vah, které jim jsou přidruženy.

$$\begin{aligned} P_{n-p+1} &= P_0, P_{n-p+2} = P_1, \dots, P_n = P_{p-1} \\ w_{n-p+1} &= w_0, w_{n-p+2} = w_1, \dots, w_n = w_{p-1} \end{aligned} \quad (3.15)$$

Řídicí polygon musí začínat nejméně  $p+1$  různými řídicími body a uzlový vektor je stejný jako u otevřené křivky (3.9), stejně tak aktivní segmenty uzavřené křivky leží na intervalu  $[u_p, u_{m-p}]$ .



Obrázek 3.5: Uzavřená B-spline křivka.

### 3.3. Interpolační NURBS a B-spline křivky

V předchozí sekci jsme se věnovali aproximačním křivkám, které nemusely procházet zadanými body. Nyní naši pozornost přesuneme ke křivkám interpolačním, které prochází tzv. definičními body. Známe tedy polohu bodů, které na křivce leží, avšak neznáme polohu řídicích bodů, které její tvar určují. Jedná se tedy o inverzní problém k nalezení aproximační křivky, vyřešíme jej tak, že nalezneme řešení soustavy rovnic, se zadanou známou polohou (popř. navíc derivacemi v nich) bodů, kterými křivka prochází, obdržíme tak polohu řídicích bodů a pak již můžeme napsat vektorovou rovnici interpolační křivky.

Ke konstrukci interpolační NURBS resp. B-spline křivky přistupujeme dvěma způsoby. Prvním je prostá interpolace, kdy je počet řídicích bodů stejný jako počet definičních bodů a druhým způsobem je uzlová interpolace, kdy je stejný počet aktivních segmentů a ramen definičního polygonu. Při tomto způsobu interpolace, je více řídicích bodů než definičních, a proto je třeba zavést dodatečné podmínky k jednoznačnému určení tvaru křivky.

#### 3.3.1. Prostá interpolace

K vyřešení interpolační úlohy musíme nejdříve zavést tzv. vektor parametrizace, který nám říká, které hodnotě parametru křivky odpovídá definiční bod, který na ní leží.

**Definice 3.9.** *Nechť  $H$  je neklesající posloupnost  $\tilde{n} + 1$  reálných čísel  $h_0 \leq h_1 \leq h_{\tilde{n}}$ , ty nazveme hodnoty parametru, množinu  $H = (h_i)_{i=0}^{\tilde{n}}$  nazveme vektor parametrizace a interval  $[h_i, h_{i+1})$  nazveme  $i$ -tou roztečí vektoru parametrizace.*

*Uniformní vektor parametrizace je takový, jehož rozteče mají stejnou délku. Pokud je délka roztečí různá, hovoříme o neuniformním vektoru parametrizace.*

**Definice 3.10.** *Prostá interpolační B-spline křivka. Nechť je dán definiční polygon  $(Q_i)_{i=0}^{\tilde{n}}$ , vektor parametrizace  $H = h_{\tilde{n}}$ , uzlový vektor  $U = (u_i)_{i=0}^m$  a stupeň křivky  $p$ . Potom*

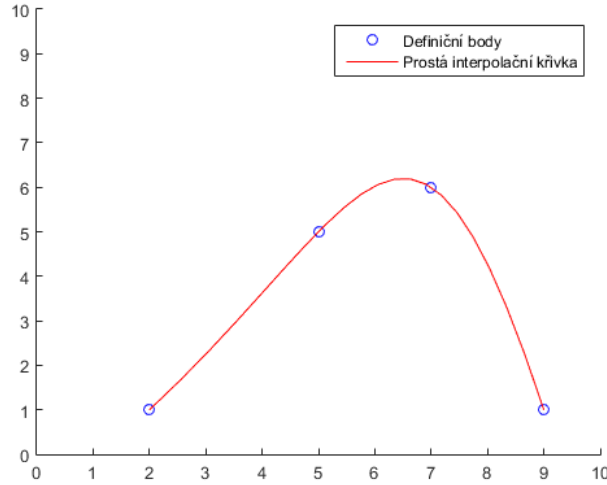
### 3.3. INTERPOLAČNÍ NURBS A B-SPLINE KŘIVKY

řídící body  $P_i, i = 0, 1, \dots, \tilde{n}$  prosté interpolační B-spline křivky  $\mathbf{C}_u$  obdržíme jako řešení soustavy rovnic

$$\begin{pmatrix} N_{0,p}(h_0) & N_{1,p}(h_0) & \dots & N_{n,p}(h_0) \\ N_{0,p}(h_1) & N_{1,p}(h_1) & \dots & N_{n,p}(h_1) \\ \vdots & & & \vdots \\ N_{0,p}(h_{\tilde{n}}) & N_{1,p}(h_{\tilde{n}}) & \dots & N_{n,p}(h_{\tilde{n}}) \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ \vdots \\ P_{\tilde{n}} \end{pmatrix} = \begin{pmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_{\tilde{n}} \end{pmatrix}, \quad (3.16)$$

kde  $N_{i,p}(h_i)$  jsou B-spline bázové funkce pro hodnotu parametru  $h_i$ . Rovnice B-spline křivky  $\mathbf{C}(u)$  stupně  $p$  má potom tvar

$$\mathbf{C}(u) = \sum_{i=0}^{\tilde{n}} N_{i,p}(u) \mathbf{P}_i, u \in [u_0, u_m]. \quad (3.17)$$



Obrázek 3.6: Interpolační B-spline křivka.

**Definice 3.11.** Nechť je dán definiční polygon  $(\mathbf{Q}_i)_{i=0}^{\tilde{n}}$ , vektor parametrizace  $H = h_{\tilde{n}}$ , uzlový vektor  $U = (u_i)_{i=0}^m$ , stupeň křivky  $p$  a váhy  $W = (w_i)_{i=0}^{\tilde{n}}$ . Potom řídící body  $R_i, i = 0, 1, \dots, \tilde{n}$  prosté interpolační NURBS křivky  $\mathbf{C}_u$  obdržíme jako řešení soustavy rovnic

$$\begin{pmatrix} R_{0,p}(h_0) & R_{1,p}(h_0) & \dots & R_{n,p}(h_0) \\ R_{0,p}(h_1) & R_{1,p}(h_1) & \dots & R_{n,p}(h_1) \\ \vdots & & & \vdots \\ R_{0,p}(h_{\tilde{n}}) & R_{1,p}(h_{\tilde{n}}) & \dots & R_{n,p}(h_{\tilde{n}}) \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ \vdots \\ P_{\tilde{n}} \end{pmatrix} = \begin{pmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_{\tilde{n}} \end{pmatrix}, \quad (3.18)$$

kde  $R_{i,p}(h_i)$  jsou racionální bázové funkce pro hodnotu parametru  $h_i$ . Rovnice NURBS křivky  $\mathbf{C}(u)$  stupně  $p$  má potom tvar

$$\mathbf{C}(u) = \sum_{i=0}^{\tilde{n}} R_{i,p}(u) \mathbf{P}_i, u \in [u_0, u_m]. \quad (3.19)$$



### 3.3.2. Uzlová interpolace

Při uzlové interpolaci chceme docílit toho, aby byl počet aktivních uzlových roztečí shodný s počtem ramen definičního polygonu a aby k napojování segmentů docházelo právě v definičních bodech, tím pádem je vektor parametrizace  $H$  aktivní částí uzlového vektoru  $U$ .

V této interpolační úloze se tedy počet řídicích bodů liší od počtu definičních bodů. Celkový počet uzlových roztečí  $m$  obdržíme jako součet aktivních a pasivních uzlových roztečí

$$m = \tilde{n} + 2p.$$

Ze vzorce pro počet uzlových roztečí můžeme vyjádřit počet ramen  $n$  řídicího polygonu

$$n = m - p - 1.$$

Dosadíme za  $m$  a obdržíme

$$n = \tilde{n} + p - 1. \quad (3.20)$$

Pro  $\tilde{n}$  definičních bodů je tedy třeba přidat  $p - 1$  dodatečných podmínek. Nejčastěji zadáváme vektory derivací v krajních bodech. Odtud lze také vidět, že pokud bychom chtěli zadat stejný počet podmínek na každém okraji, musíme použít interpolační křivku lichého stupně. Abychom dále zajistili při napojování křivek  $C^2$  spojitost a zároveň lokalitu křivky, je třeba použít křivku 5. stupně.

**Definice 3.12.** *Uzlová interpolační B-spline křivka 5. stupně. Nechť je dán definiční polygon  $(\mathbf{Q}_i)_{i=0}^{\tilde{n}}$ , stupeň křivky  $p = 5$ , vektor parametrizace  $H = (h_i)_{i=0}^{\tilde{n}}$  a v počátečním bodě  $\mathbf{Q}_0$ , resp. v koncovém bodě  $\mathbf{Q}_{\tilde{n}}$  definičního polygonu, vektor první a druhé derivace  $\mathbf{Q}'_0, \mathbf{Q}''_0$ , resp.  $\mathbf{Q}'_{\tilde{n}}, \mathbf{Q}''_{\tilde{n}}$ . Potom souřadnice řídicích bodů  $P_i, i = 0, \dots, n$ , obdržíme jako řešení soustavy rovnic*

$$\begin{pmatrix} N_{0,5}(h_0) & N_{1,5}(h_0) & \dots & N_{n,5}(h_0) \\ N_{0,5}(h_1) & N_{1,5}(h_1) & \dots & N_{n,5}(h_1) \\ \vdots & & & \vdots \\ N_{0,5}(h_{\tilde{n}}) & N_{1,5}(h_{\tilde{n}}) & \dots & N_{n,5}(h_{\tilde{n}}) \\ N'_{0,5}(h_0) & N'_{1,5}(h_0) & \dots & N'_{n,5}(h_0) \\ N''_{0,5}(h_0) & N''_{1,5}(h_0) & \dots & N''_{n,5}(h_0) \\ N'_{0,5}(h_{\tilde{n}}) & N'_{1,5}(h_{\tilde{n}}) & \dots & N'_{n,5}(h_{\tilde{n}}) \\ N''_{0,5}(h_{\tilde{n}}) & N''_{1,5}(h_{\tilde{n}}) & \dots & N''_{n,5}(h_{\tilde{n}}) \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ \vdots \\ P_{n-4} \\ P_{n-3} \\ P_{n-2} \\ P_{n-1} \\ P_n \end{pmatrix} = \begin{pmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_{\tilde{n}} \\ Q'_0 \\ Q''_0 \\ Q'_{\tilde{n}} \\ Q''_{\tilde{n}} \end{pmatrix}. \quad (3.21)$$

Interpolační B-spline křivka je potom dána vektorovou rovnicí

$$\mathbf{C}(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i, u \in [a, b] \quad (3.22)$$

Pokud nebudeme při napojování křivek požadovat tak silnou spojitost, můžeme využít i křivku třetího stupně, ta je definována obdobně jako předchozí.

**Definice 3.13.** *Uzlová interpolační B-spline křivka 3. stupně. Nechť je dán definiční polygon  $(\mathbf{Q}_i)_{i=0}^{\tilde{n}}$ , stupeň křivky  $p = 3$ , vektor parametrizace  $H = (h_i)_{i=0}^{\tilde{n}}$  a v počátečním*

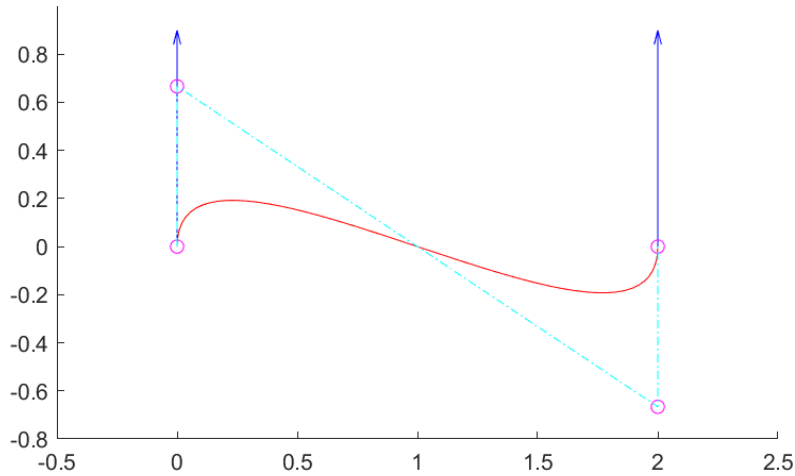
### 3.3. INTERPOLAČNÍ NURBS A B-SPLINE KŘIVKY

bodě  $\mathbf{Q}_0$ , resp. v koncovém bodě  $\mathbf{Q}_{\tilde{n}}$  definičního polygonu, vektor první derivace  $\mathbf{Q}'_0$ , resp.  $\mathbf{Q}'_{\tilde{n}}$ . Potom souřadnice řídicích bodů  $P_i, i = 0, \dots, n$ , obdržíme jako řešení soustavy rovnic

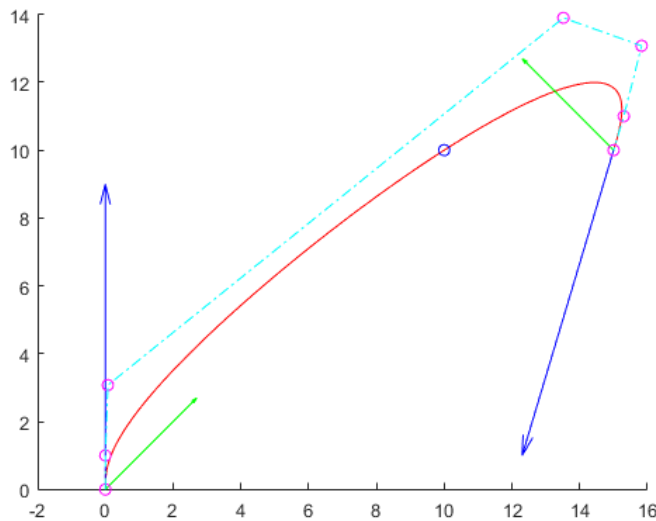
$$\begin{pmatrix} N_{0,3}(h_0) & N_{1,3}(h_0) & \dots & N_{n,3}(h_0) \\ N_{0,3}(h_1) & N_{1,3}(h_1) & \dots & N_{n,3}(h_1) \\ \vdots & & & \vdots \\ N_{0,3}(h_{\tilde{n}}) & N_{1,3}(h_{\tilde{n}}) & \dots & N_{n,3}(h_{\tilde{n}}) \\ N'_{0,3}(h_0) & N'_{1,3}(h_0) & \dots & N'_{n,3}(h_0) \\ N'_{0,3}(h_{\tilde{n}}) & N'_{1,3}(h_{\tilde{n}}) & \dots & N'_{n,3}(h_{\tilde{n}}) \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ \vdots \\ P_{n-2} \\ P_{n-1} \\ P_n \end{pmatrix} = \begin{pmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_{\tilde{n}} \\ Q'_0 \\ Q'_{\tilde{n}} \end{pmatrix}. \quad (3.23)$$

Interpolační B-spline křivka je potom dána vektorovou rovnicí

$$\mathbf{C}(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i, u \in [a, b]. \quad (3.24)$$



Obrázek 3.7: Uzlová interpolační křivka 3. stupně. Řídicí body  $\mathbf{P}$  fialově. Pozn. první a poslední bod řídicího polygonu (čerchovaná čára) jsou totožné s definičními body  $\mathbf{Q}$ .



Obrázek 3.8: Uzlová interpolační křivka 5. stupně.

## 4. Aplikace pro modelování trajektorie

Aplikace vytvořená pro tuto diplomovou práci by měla umožňovat manuálně navrhnout trajektorii složenou ze segmentů různých typů křivek. Vybrat si lze mezi přímkou, bézierovou kubikou a uzlovou interpolační B-spline křivkou 5. stupně. Tyto křivky byly k modelaci zvoleny z důvodu nízké náročnosti jejich implementace a jednoduché obsluhy vstupních parametrů definujících výsledný tvar křivky na úseku určeném interpolačními body. Např. při využití klotoid, nelze pevně zvolit koncový bod segmentu. Ten je určen výsledkem Newtonovy minimalizační metody pro vzdálenost mezi koncovým bodem klotoidy a přímkou, na kterou se má klotoida napojit. Tento fakt komplikuje modelaci trajektorie např. v tzv. "esíčku", kde může nastat taková situace, ve které nebude minimalizační proces mít dostatečně dlouhou přímkou k nalezení vhodného místa k napojení. Tento problém lze vyřešit [17], avšak mně se bohužel implementace nezdařila.

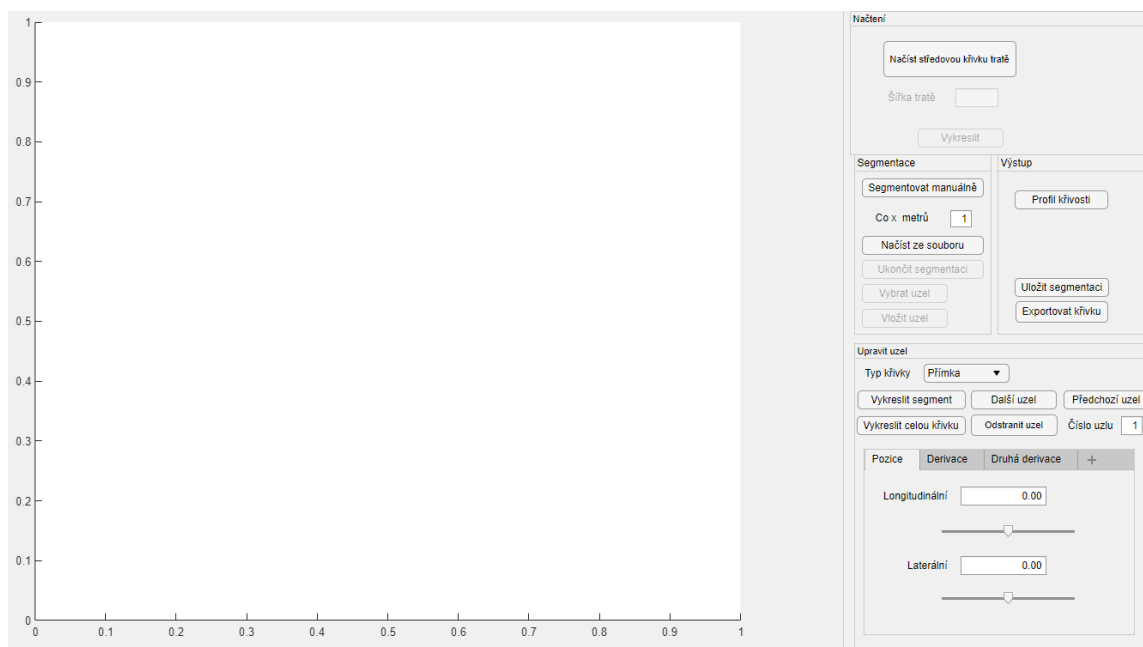
### 4.1. Popis aplikace

#### 4.1.1. Uživatelský manuál

Hlavní okno programu lze rozdělit na dvě hlavní části. Pravá část je vyhrazena pro ovládací prvky a zbytek okna pro grafický výstup. Ovládací prvky jsou uskupeny do čtyř panelů.

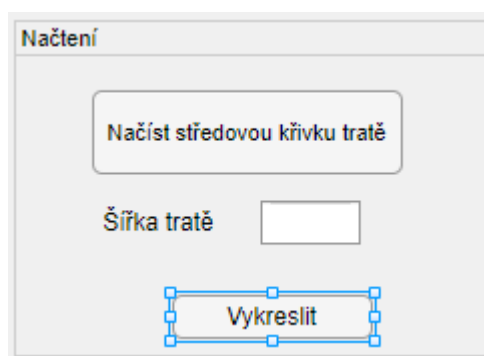
- Načtení
- Segmentace
- Výstup
- Upravit uzel

Při zapnutí aplikace je aktivní pouze panel *Načtení*, který slouží k zadání vstupu - csv soubor se souřadnicemi středové čáry okruhu a její případnou parametrizaci. Je však potřeba dodržet daný formát. Soubor musí mít na prvním řádku hlavičku tvořenou stringy. X-ové souřadnice musí být ve 4. sloupci, Y-ové souřadnice v 6. a parametrizace, která je nepovinná, v 5. sloupci. Hodnoty X-ových, Y-ových souřadnic a parametru musí být pouze čísla, ať už integrity, floaty nebo doubly. Po načtení dat se zadá do textového políčka šířka tratě, která bude po celou dobu konstantní a po kliknutí na tlačítko *Vykreslit* se v grafickém poli vykreslí trať a odemkne panel *Segmentace*.

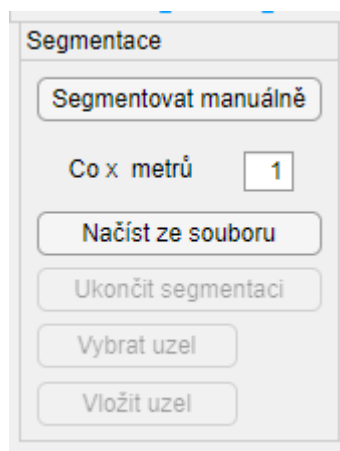


Obrázek 4.1: Hlavní okno aplikace

Navíc, k zajištění správné funkčnosti aplikace, je třeba zajistit, aby trať byla regulární křivka a její šířka byla zadána tak, aby se okraje tratě nekřížily.

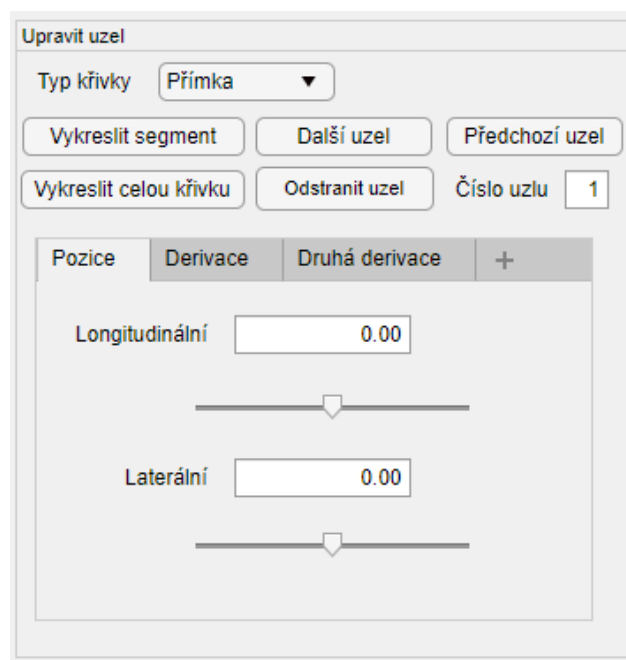
Obrázek 4.2: Panel *Načtení*

Panel *Segmentace* slouží k zadávání uzlů, které budou definovat výslednou křivku. Program nabízí tři způsoby, jak uzel přidat. Manuálně, co  $x$  jednotek vzdálenosti a načtením ze souboru. Při ručním zadávání se automaticky vytvoří první uzel v prvním bodě tratě a následně se volí umístění dalších uzlů levým kliknutím myši na středovou čáru, pravým kliknutím na středovou čáru pak lze vždy poslední uzel smazat. Místo manuálního zadávání uzlů lze také trať segmentovat ekvidistantním dělením, zadáním vzdálenosti mezi jednotlivými uzly do textového pole vedle popisku *Co x metrů*, poté se stačí jen vykliknout z textového pole a uzly se vytvoří. Třetí možnost je nahrát ze souboru již uloženou konfiguraci, opět se načítá soubor ve formátu csv.

Obrázek 4.3: Panel *Segmentace*

Po ukončení segmentační fáze se tyto tři možnosti uzamknou a v panelu *Segmentace* se odemknou tlačítka *Vybrat uzel* a *Vložit uzel*, zároveň se zpřístupní panely *Výstup* a *Upravit uzel* a automaticky se vybere první uzel. Tlačítko *Vybrat uzel* umožní zvolit si uzel kliknutím na již vykreslený uzlový bod. Zvolený uzel je dále zvýrazněn zeleným křížkem. Tlačítkem *Vložit uzel* lze vložit nový uzel mezi dva již vytvořené uzly. Správná funkčnost tohoto tlačítka je zajištěna splněním dvou následujících podmínek:

- Trajektorie již musí být vykreslená
- Nový uzel nesmí být vložen za poslední

Obrázek 4.4: Panel *Upravit uzel*

Parametry zvoleného uzlu lze editovat v panelu *Upravit uzel*. Právě zde se vybírá typ křivky v úseku mezi následujícím a zvoleným uzlem, jeho pozice a případné derivace. Pomocí tlačítek *Další uzel* a *Předchozí uzel*, popř. zadáním pořadového čísla uzlu lze navíc mezi uzly přepínat. *Vykreslit segment* vypočte souřadnice křivky od předcházejícího

uzlu po následující. *Vykreslit celou křivku* postupně propočítá hodnoty pro danou křivku segment po segmentu, avšak ne vždy se tato funkce chová správně a někdy způsobí chybu. Aplikace umožňuje generaci tří typů křivek:

- Přímka
- Uzlový interpolační spline 3. stupně
- Uzlový interpolační spline 5. stupně

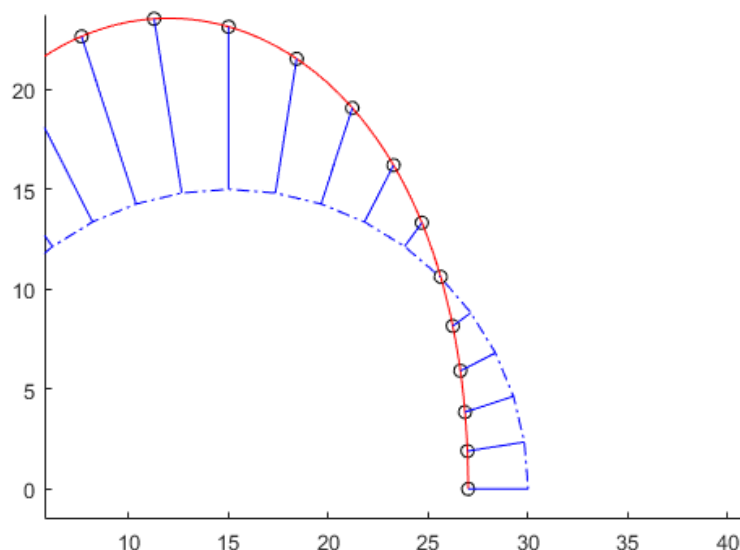
Typ křivky je definován vždy na segmentu začínajícím zvoleným uzlem a končícím následujícím uzlem. Tzn., pro zvolený uzel  $n$  vybereme typ křivky na segmentu  $[n, n + 1]$ , ale pokud zmáčkne *Vykreslit segment*, dojde k přepočtu křivky a vykreslení na uzlech  $[n - 1, n, n + 1]$ .

Výsledná vykreslená křivka je před vykreslením a uložením interpolovaná z vypočtených hodnot tak, aby byla namapovaná na body středové čáry tratě, mapování tak vytvoří z křivky funkci středové čáry

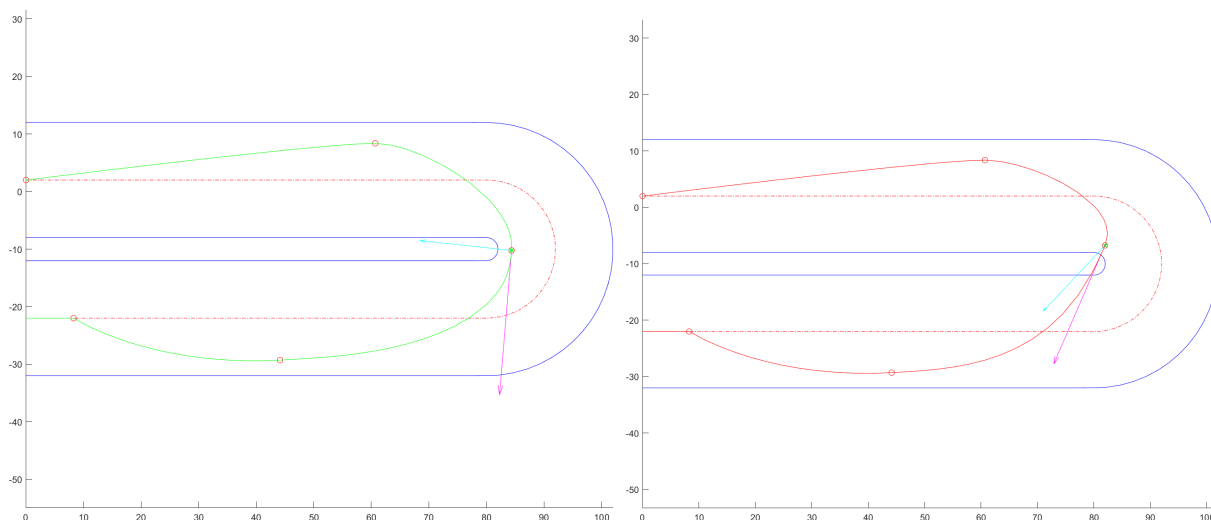
$$\mathbf{C} = f(s, n),$$

kde  $s$  je vzdálenost od počátku středové čáry, až do místa, kde by se na ni bod kolmo promítl a  $n$  je délka kolmice vedené ze středové čáry k danému bodu. Znaménko  $n$  potom udává, na kterou stranu od středové čáry bod připadá, nalevo je  $n$  kladné, napravo záporné.

Avšak takové zobrazení může vést ke ztrátě informací, pokud by na vypočtené křivce vznikla např. smyčka, potom mapování nebude brát v potaz vzdálenější body, popř. dojde k pádu programu. Na druhou stranu tato reprezentace umožňuje detekci toho, jestli se křivka nalézá uvnitř okruhu. Pokud ano, vykreslí se zeleně, pokud ne, bude vykreslená červeně.



Obrázek 4.5: Mapování bodů křivky (červená plná čára) vůči středové čáře (modrá čerchovaná). Interpolované v modrém kolečku.



Obrázek 4.6: Příklad validní a nevalidní křivky, na pravém obrázku prochází křivka mimo trať

Vůči středové čáře je určovaná i pozice uzlového bodu. Nezadávají se tedy kartézské souřadnice bodu. Hodnota derivace, popř. druhé derivace je reprezentovaná pomocí velikosti a směru vektoru. Úhel vektoru je relativní vůči směru středové čáry tratě, kladný směr otáčení je levotočivý, bohužel je to opačný směr, než při pojíždění koněm, kterým lze hodnoty měnit. Navíc rozpětí úhlu pro první derivaci je interval  $[-\pi/2, \pi/2]$ , nelze se tedy "vracet" a pro druhou derivaci  $[-\pi, \pi]$ , lze tedy "zpomalovat". Změny v nastavení hodnot pozice a derivací uzlového bodu by se měly projevit v grafickém poli téměř okamžitě, avšak k přepočtení křivky dojde až po kliknutí na jedno z vykreslujících tlačítek. V grafickém poli jsou derivace reprezentovány šipkami vycházejícími ze zvoleného uzlového bodu, první derivace je magentová, druhá azurová.

#### 4.1.2. Technická dokumentace

Aplikace je navržena v Matlabu s využitím toolu App Designer, který je dostupný od verze 2016 a využívá některé další funkcionality dostupné až od verze 2019, ale nevyužívá žádné další nadstandardní balíčky. Spouští se ze souboru *Main\_app.mlapp*.

Srdcem programu jsou tři třídy:

- SectionClass
- TrackClass
- CurveClass

*SectionClass* reprezentuje úsek mezi dvěma uzly. Objekty této třídy nesou informace o souřadnicích a derivacích krajních bodů a typu křivky, která má být na takto zadaném segmentu vypočtena. Což tedy znamená, že jsou zde uloženy všechny potřebné informace ke zkonstruování křivky. Z toho důvodu je v této třídě také definovaná metoda *ConstructCurve* pro provedení správného výpočtu v závislosti na typu křivky.

Třída *TrackClass* nese informace o souřadnicích středové čáry tratě, její šířce a délce v každém jejím definovaném bodě. Dále obsahuje mj. tyto důležité metody:

*SetBorders* pro výpočet souřadnic levého a pravého okraje okruhu v závislosti na šířce tratě v daném bodě.

*GetCurvilinearCoordsForPoint*, která vrací k poloze bodu zadané v kartézském souřadném systému jeho pozici vzhledem ke středové čáře.

*GetAbsolutePointPosition*, která naopak vrací kartézské souřadnice pro bod zadaný polohou vůči trati.

Třída *CurveClass* potom slouží k uložení souřadnic výsledné křivky a to jak kartézských, tak vztažených ke středové čáře.

## Konstrukce křivek

Souřadnice úsečky se vypočítají standardně pomocí parametrické rovnice přímky, kterou lze získat z koncových bodů úseku  $P_1$  a  $P_2$ .

$$\vec{u} = \mathbf{P}_2 - \mathbf{P}_1, \text{ v aplikaci } \mathbf{P}_2, \mathbf{P}_1 \in \mathbb{R}^k$$

$$\mathbf{C}(t) = \mathbf{P}_1 + t\vec{u}, \quad t \in [0, 1]$$

K výpočtu interpolačních křivek se využívají definice 3.13 (implementováno v souboru *bsplineinterpolace3stupen*) a 3.12 (implementováno v souboru *bsplineinterpolace5stupen*).

V obou výše zmíněných souborech je definovaná příslušná funkce pro výpočet souřadnic interpolační křivky na uniformně děleném intervalu. Obě mají jako vstup matici a skalár, ten udává jemnost dělení.

Pro interpolační křivku třetího stupně je potom matice složena z  $n$  interpolačních bodů. Matice má  $n+2$  řádků, kde až na poslední dva řádky odpovídá každý řádek jednomu interpolačnímu bodu, v předposledním řádku je derivace v prvním bodě a v posledním řádku matice je zadaná derivace v posledním bodě. Tzn. např.

$$A = \begin{pmatrix} 0 & 0 \\ 7 & 10 \\ 15 & 10 \\ 10 & 0 \\ 0 & -10 \end{pmatrix}$$

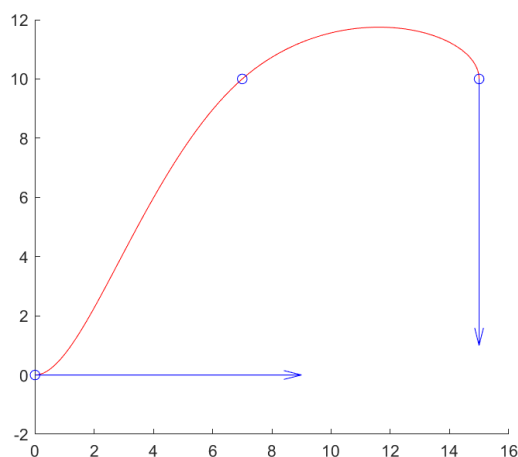
bude procházet body  $Q1 = (0,0)$ ,  $Q2 = (7,10)$  a  $Q3 = (15,10)$ , derivace v bodě  $Q1$  bude mít hodnotu  $Q1' = (10,0)$ , v bodě  $Q3$  zase  $Q3' = (0,-10)$ .

Obdobně vypadá vstupní matice pro interpolační křivku pátého stupně, avšak s tím rozdílem, že jsou pro derivace vyhrazeny poslední čtyři řádky, jelikož se musí v okrajových interpolačních bodech zadat navíc druhé derivace. Nejdříve se zadají první derivace

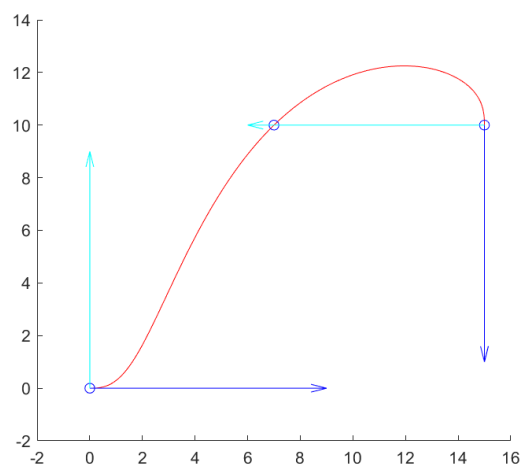


pro první a poslední interpolační bod a nakonec druhé derivace pro tyto body. Obdobně jako u matice  $A$  výše názorný příklad:

$$B = \begin{pmatrix} 0 & 0 \\ 7 & 10 \\ 15 & 10 \\ 10 & 0 \\ 0 & -10 \\ 0 & 10 \\ -10 & -0 \end{pmatrix}$$



(a) Uzlová interpolační křivka 3. stupně zadaná maticí  $A$ .



(b) Uzlová interpolační křivka 5. stupně zadaná maticí  $B$ .

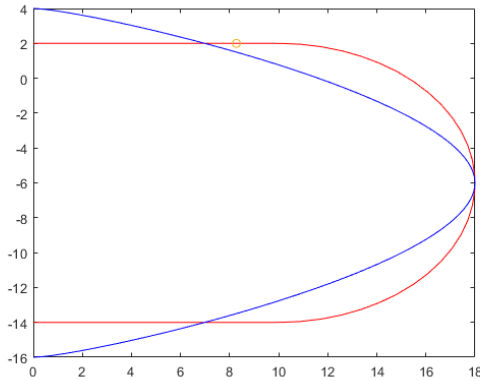
Obrázek 4.7: Příklad interpolačních křivek.

V samotné aplikaci se však jako interpolační body berou pouze dva uzly. Toto je určitě jedno z potenciálních vylepšení aplikace. Na druhou stranu v těchto vnitřních bodech nelze kontrolovat hodnoty derivací, je v nich však zajištěno hladké napojení stupně o jeden nižší, než je stupeň spline křivky [10].

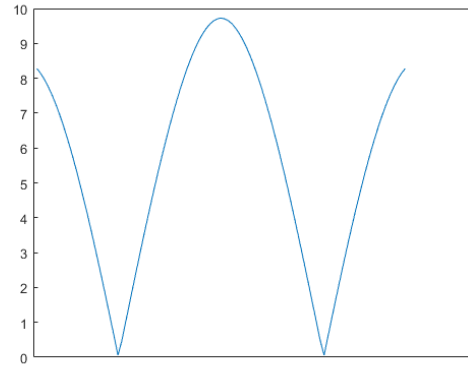
Z výše uvedeného vyplývá, že k výpočtu křivky se na vstupu výpočtu podílí pouze relevantní data. Tedy přímka je definována pouze dvěma body a neuvažují se žádné derivace, interpolační křivka třetího stupně je určena pozicí dvou bodů a prvními derivacemi a nakonec interpolační křivka pátého stupně oba výše zmíněné vstupy, a navíc druhé derivace. Také tedy není nutně zajištěna vyšší spojitost křivek než  $C^0$  pro přímku s libovolnou jinou křivkou a  $C^1$  pro interpolační křivku 3. stupně s interpolační křivkou 5. stupně.

### Parametrizace křivky vůči středové čáře

Po vypočtení křivky proběhne transformace výsledku z kartézského souřadnicového systému do kurvilinearního systému vztaženého vůči středové čáře trati. Tato transformace se provádí pomocí metody *MapCurveRelativeToCenterLine*, která nejdříve voláním funkce *MapMinDotProducts* najde pro každý bod středové čáře jeden bod na vypočtené křivce algoritmem zjednodušeně popsaným v 4.1 tak, aby byl co nejblíže ke kolmici vedené z bodu na středové čáře, a poté lineárně interpoluje hodnoty křivky tak, aby skutečně ležely na této kolmici ležely viz obrázek 4.5.



(a) Bod na středové čáře (znázorněn kolečkem), vůči kterému se počítá skalární součin z bodů modré křivky



(b) Absolutní hodnota skalárního součinu daného bodu vůči bodům křivky.

Obrázek 4.8: Příklad nekonvexního průběhu skalárního součinu

Jak je z algoritmu patrné, k detekci nulového skalárního součinu je využita Bolzanova věta a algoritmus prochází všechny body křivky, je to tedy bohužel *brute force* metoda. Obecně totiž není zajištěno, že průběh absolutní hodnoty skalárního součinu na daném intervalu bude konvexní funkce. Přesto v kódu aplikace je při volání těchto funkcí snaha o co nejúspornější průběh výpočtu a to např. na vstupu do druhého for cyklu uvažovat již pouze dosud nezpracované body vstupní křivky nebo ukončit výpočet po překročení jisté dostatečné maximální vzdálenosti.

---

**Algorithm 4.1** Nalezení odpovídajících bodů

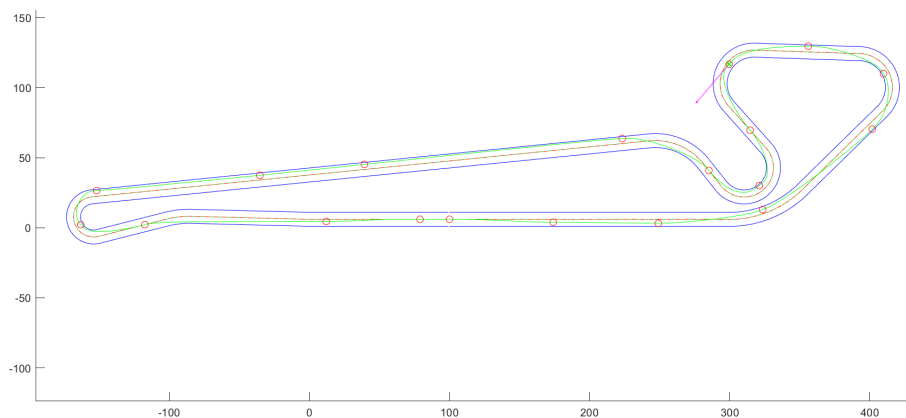
---

```

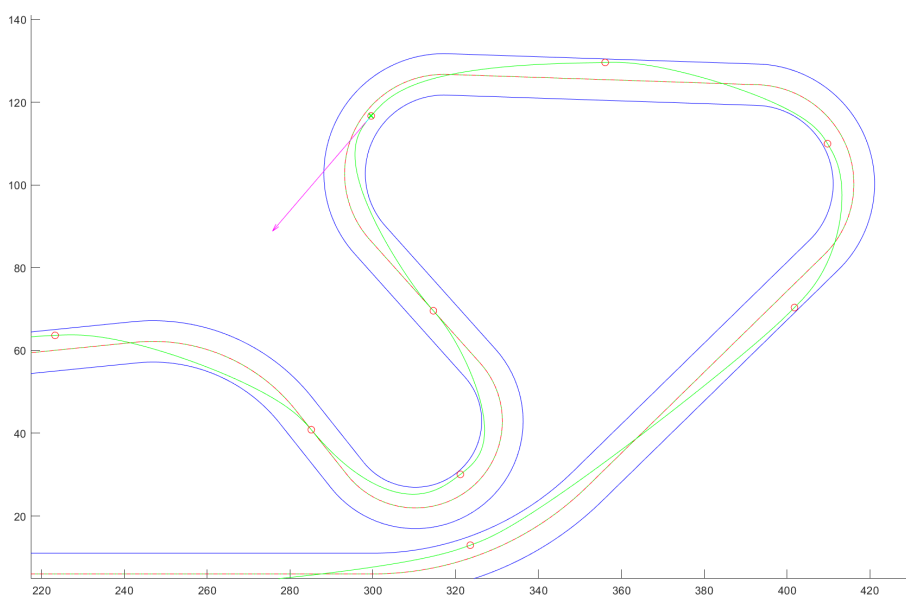
1: Vstupy: vypočtená_křivka, středová_čára, parametrizace_středové_čáry
2: minimum = 5000;
3: for all bod_st in středová_čára do
4:   index = 1;
5:   vektor_derivace = Derivace(parametrizace_středové_čáry[bod_st], středová_čára[bod_st]);
6:   for all bod_k in vypočtená_křivka-1 do
7:     soucin1 = Skalární_součin(vektor_derivace, [bod_k-bod_st]);
8:     soucin2 = Skalární_součin(vektor_derivace, [následující_bod_k - bod_st])
9:     dist = Norma(bod_st - bod_k)
10:    if soucin1 * soucin2 < 0 && minimum then
11:      minimum = dist;
12:      min_bod = bod_k
13:    end if
14:  end for
15:  Nalezené_body[index] = min_bod;
16:  index = index+1;
17: end for
18: Výstup: Nalezené_body;

```

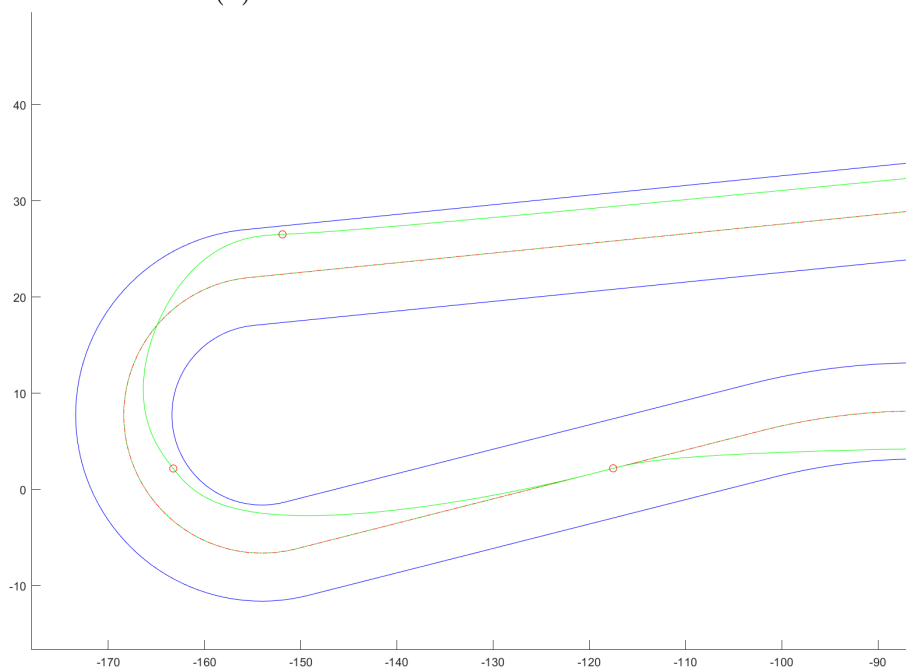
---



(a) Příklad vymodelované trajektorie.



(b) Přiblížení k technické sekci okruhu.



(c) Přiblížení k vlásence. Trajektorie s pozdním apexem

Obrázek 4.9: Příklad trajektorie na testovací trati.

## 4.2. Možná vylepšení aplikace a její nedostatky

### Možná vylepšení

Návrhu k vylepšení aplikace je hned několik, pomineme-li nyní optimalizaci kódu a zaměříme-li se pouze na přidání nových funkcionalit. Potom existují tři hlavní funkcionality, o které by program mohl být v budoucnu obohacen. Jsou to:

- Přidání více typů křivek
- Výpočet rychlostního profilu pro vozidlo
- Automatická optimalizace trajektorie

První bod z tohoto seznamu zní přímočaře. Na první pohled by se mohlo zdát, že by stačilo přidat typ křivky do seznamu a do třídy *SectionClass.m* přidat výpočetní metodu. Avšak v závislosti na typu křivky a vstupních parametrech existuje možnost, že by bylo třeba upravit celou logiku pro vstupy výpočtu křivky na daném segmentu, změnit význam vstupních parametrů, např. z druhé derivace přejít na křivost, nebo zavést jako parametr směr přímky a jejich délku, a v neposlední řadě zohlednit typ segmentu vůči svému okolí. Nynější řešení tato kritéria nemusí zohledňovat.

Implementace výpočtu rychlostního profilu pro dané vozidlo, by na druhou stranu měla být jednodušší. Jako vstupní parametry přímo z aplikace postačují souřadnice, popř. křivost, dané křivky, poté by bylo potřeba dodat potřebné parametry definující model vozidla. Tyto parametry jsou závislé na složitosti modelu, který se uvažuje. Pro nejjednodušší možný případ, tj. reprezentace vozidla pomocí hmotného bodu s kontinuální převodovkou a konstantním výkonem, jsou těmito parametry hmotnost, maximální dopředné zrychlení a maximální laterální zrychlení. Z teorie optimálního řízení potom vyplývá algoritmus 4.2 potřebný k výpočtu rychlostního profilu [18].

---

#### Algorithm 4.2 Výpočet rychlostního profilu

---

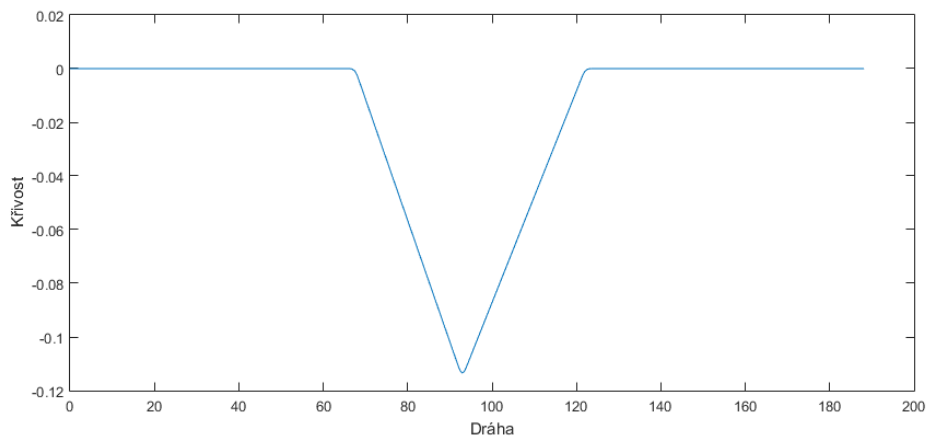
```

1: Vstupy: Křivka, hmotnost vozidla, maximální laterální zrychlení, maximální dopředné
   zrychlení
2: Vrcholy = Najdi_vrcholy(vypočti_křivost(Křivka));
3: for all vrchol in vrcholy do
4:   v_krit = Vypočti_maximální_rychlost_ve_vrcholu(maximální laterální zrychlení,
   Křivost, Vrcholy);
5:   v_dopředná[vrchol : vrchol+1] = Rychlost_v_dopředném_směru(v_krit, Křivost,
   maximální laterální zrychlení, maximální dopředné zrychlení);
6:   v_zpětná[vrchol : -1 : vrchol-1] = Rychlost_ve_zpětném_směru(v_krit, Křivost, ma-
   ximální laterální zrychlení, maximální dopředné zrychlení);
7: end for
8: for all bod in Křivka do
9:   v_výsledná[bod] = min(v_dopředná[bod], v_zpětná[bod]);
10: end for
11: Výstup: v_výsledná

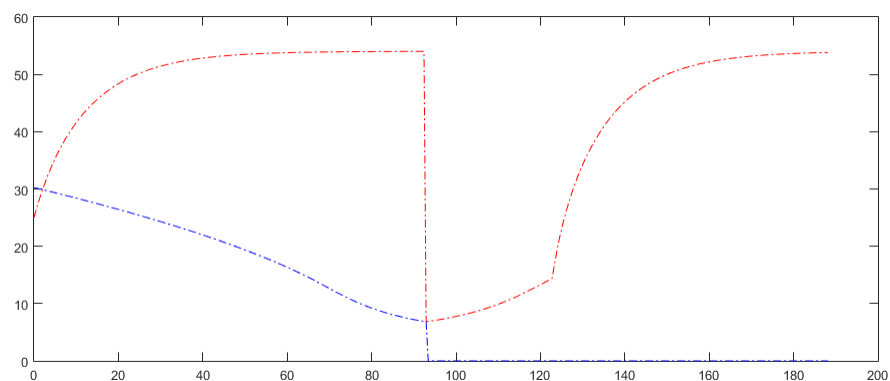
```

---

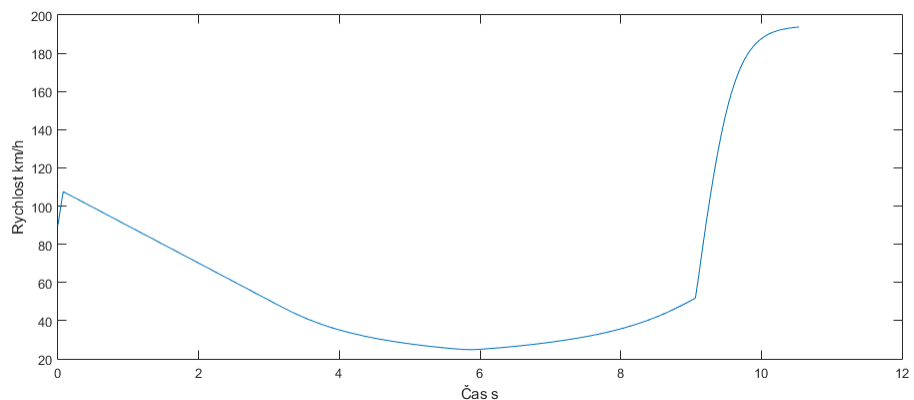
## 4.2. MOŽNÁ VYLEPŠENÍ APLIKACE A JEJÍ NEDOSTATKY



(a) Křivost charakterizující křivku vstupující do algoritmu 4.2



(b) Rychlost v dopředném směru - červeně, rychlost ve zpětném směru - modře



(c) Výsledná rychlost - nenulové minimum dopředné a zpětné rychlosti

Obrázek 4.10: Příklad výstupu z algoritmu 4.2. V tomto příkladu je chyba ve výpočtu rychlosti z maximálního zrychlení. Znázorněný princip je však správný.

Automatická optimalizace by spočívala v určení takových parametrů, které by minimalizovaly čas, po který by vozidlo projelo danou křivkou. Problém nalezení takové trajektorie

## 4.2. MOŽNÁ VYLEPŠENÍ APLIKACE A JEJÍ NEDOSTATKY

se v tomto případě většinou řeší jako minimalizace křivosti a délky křivky v poměru určeném váhovými koeficienty, minimalizuje se tedy následující funkcionál:

$$F^2 = (1 - \epsilon)\Gamma^2 + \epsilon S^2,$$

kde  $\epsilon$  váhový koeficient,  $\Gamma$  je křivost a  $S$  je délka dráhy. Viz např. Cardamone [4], kde tuto minimalizaci provádí pomocí genetických algoritmů nebo Brahina, aj. [3], kteří generují trajektorii pomocí přirozených kubických splinů a funkcionál minimalizují jako kvadratickou formu.

Další drobné rozšíření, které by ale mohlo mít velký přínos by mohla být validace křivky v závislosti na maximální křivosti, ta totiž implikuje jistý poloměr zatáčení a při příliš vysokém poloměru, by vozidlo nemuselo být schopné po dané křivce projet.

### Nedostatky

V první řadě je potřeba zmínit, že se v programu chybně přiřazuje velikost a směr derivací, když se překlikává mezi nakonfigurovanými uzly. Myslel jsem, že to je způsobeno překlepem při přiřazování směrů, ale bohužel v kódu zůstala ještě někde chyba, kterou se mi nedaří objevit. Také někdy dojde k pádu aplikace při kliknutí na tlačítko vykreslit celou křivku. Zejména se to stává, pokud ještě nejsou nakonfigurovány všechny uzly, což způsobí, že se mezi nimi vytvoří přímky, které vedou klidně mimo okruh a v některých situacích toto může vést ke špatné detekci nulových skalárních součinů a tedy nemožnosti přiřadit body vůči středové čáře. Velký nedostatek z hlediska kódu spatřuji v reprezentaci uzlových bodů. Uzly jsou v kódu chápány spíše jako segmenty s počátkem a koncem. Konec daného uzlu je však v konečném důsledku stejný jako počátek následujícího. K této volbě jsem se uchýlil v rané fázi návrhu aplikace a držel jsem se jí po celou dobu vývoje. Avšak ve finišovací fázi vývoje se tato volba ukázala, jako redundantní, ne-li vyloženě nevhodná. Výhodou této reprezentace je, že jeden objekt nezávisle na ostatních může definovat celý tvar křivky. Nevýhodou je duplikace při přiřazování hodnot koncovým bodům segmentů, např. při vkládání nových uzlů mezi dva již existující nebo při jakékoli změně vlastností uzlového bodu. Tato duplikace zvyšuje šanci na chybu v kódu a vede ke komplikacím v logice jistých funkcí, jako je například přidělování hodnot derivací pro jednotlivé uzly při obnovování grafických prvků.

Další slabinou aplikace je zvolená tříbodová metoda numerické derivace ve funkci *Derivace*, která hlavně v krajních bodech křivky vrací hodnoty derivace s velkou chybou, to způsobuje problémy při hledání nulového skalárního součinu mezi středovou čarou trati a zvoleným bodem, neboť je tak vychýlen směr vektoru derivace v okrajových bodech na zkoumaných úsecích, což může mít za následek chybnou detekci mapovaného bodu.

## 5. Závěr

### 5.1. Rešerše typů křivek

Jedním z primárních úkolů této práce bylo prozkoumat různé typy křivek, které lze využít k popisu trajektorie vozidla. Ze zpracovaných zdrojů vyplynulo, že v ideálním případě by takové křivky, popř. kombinace různých křivek, měly splňovat požadavek na geometrickou spojitost druhého stupně, tzn. tyto křivky by měly mít spojitou křivost. Je to tak z toho důvodu, že pokud by vozidlo následovalo trajektorii s nespojitou křivostí, buď by muselo v bodě nespojitosti zastavit, nebo pokud by se vozidlu podařilo tuto trajektorii následovat, musela by v jeho řízení být také nespojitost.

Vzhledem k tomuto poznatku se jeví, jako vhodné k popisu trajektorie vozidla následující typy křivek:

- Klotoidy
- $\eta$ -spliny
- Uzlové interpolační spliny 5. stupně

Všemi těmito křivkami lze zajistit, spojitou křivost trajektorie.

#### Klotoidy

Klotoidy jsou křivky charakterizovány lineárním průběhem křivosti v závislosti na své délce. Popisují je Fresnelovy integrály 2.1. Při tvoření trajektorie se využívají jako přechodnicové křivky mezi rovinkami a kruhovými oblouky, díky svému lineárnímu průběhu křivosti zajistí spojitost křivosti výsledné křivky. Průjezd zatáčkou pomocí tohoto modelu by se pak podle [17] skládal ze 4 po sobě jdoucích fází: počáteční rovinka, utahující se klotoida, kruhový oblouk, zde by byla maximální křivost, otevírající se klotoida napojující se na následující rovinku. Aby se dalo tuto trajektorii zhotovit, je potřeba na okruhu identifikovat zatáčky a rovinky a poté zbývá určit délky jednotlivých úseků tak, aby byl zajištěn tečný dotek mezi klotoidou na výjezdu a rovinkou. Nevýhodou této metody je, že pokud chceme přesně určit vrchol zatáčky, kterého se má trajektorie dotknout, potom jeden ze vstupních parametrů takové optimalizační úlohy je délka počáteční rovinky a výstupem je bod napojení klotoidy a následující rovinky, což znamená, že nelze kontrolovat přesnou pozici těchto dvou okrajových bodů. To potom ztěžuje konstrukci trajektorie v úsecích, mezi kterými je pouze krátká rovinka, popř. inflexní bod.

#### $\eta$ -spliny

Tato skupina křivek umožňuje interpolaci pro zadané počáteční a koncové body, úhel směru derivace v nich a křivost.  $\eta$  v názvu křivky odkazuje na fakt, že tyto křivky jsou parametrizovány vektorem parametrů  $\vec{\eta} \in \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R} \times \mathbb{R}$ . Parametry  $\eta_1$  a  $\eta_2$  jsou podle [14] spjatý s velikostí „rychlosti“ a parametry  $\eta_3$  a  $\eta_4$  se zakřivením křivky. Bohužel, kromě několika základních poznatků se mi nepodařilo vypátrat o těchto křivkách podrobnější informace, proto by mohly být vhodným kandidátem pro další výzkum. V příloze B je uveden skript v Matlabu, který tyto křivky vypočte.

### Uzlové interpolační spliny 5.stupně

Tyto interpolační křivky zajišťují spojitost v druhé derivaci, čímž je automaticky zajištěná spojitost křivosti křivky. Relativně lehce se s těmito spliny manipuluje, a proto byly zvoleny jako vhodné pro aplikaci, která je také výstupem této práce. Avšak nejsou úplně ideální, protože v některých případech u nich může docházet k nechtěnému vlnění. Definice splinových křivek jsou popsány v kapitole 3 společně s jejich základními vlastnostmi. Společně s B-spline křivkami jsou zde popsány i NURBS křivky, které bývají často využívány k modelování povrchů na počítači, avšak mně se nepodařilo nalézt jejich vhodné vyjádření, přestože v praxi bývají ke generování trajektorie vozidla používány, např. [2]

## 5.2. Aplikace pro manuální modelování dráhy

Součástí výstupu této diplomové práce je i program, který by měl umožnit ruční vytvoření modelu trajektorie. Aplikace je podrobně popsána v kapitole 4. Byla vytvořena v Matlabu ve verzi 2020 se studenskou licencí a využívá tool App Designer, který umožňuje v Matlabu vytvořit relativně responzivní aplikace, navíc v něm lze využívat lokální a globální parametry a funkce, díky čemu se v něm pracuje s objekty mnohem lépe, než při tvoření obyčejných matlabovských figure, navíc je zde oproti normálnímu prostředí Matlabu vylepšené našeptávání, které snižuje pravděpodobnost hloupých překlepů. Na druhou stranu z něj byly odebrány některé callbacky, které ulehčovaly manipulaci s grafy.

Aplikace se spouští ze souboru *Main\_app.mlapp*. Po jejím spuštění je potřeba zadat jako vstup csv soubor se souřadnicemi tratě a do textového pole zadat šířku okruhu. Poté se může v grafickém panelu tato trať vykreslit.

Dalším krokem je rozdělení tratě na úseky, na kterých se posléze definuje typ křivky k vykreslení. Toto rozdělení se může provést třemi různými způsoby a to: Ručně, uniformně, načtením ze souboru. Uniformní rozdělení probíhá tak, že se zadá vzdálenost, která bude jednotlivé body oddělovat. Ze souboru lze načíst již uloženou konfiguraci uzlů a to nejen jejich polohu, ale také derivace. A nakonec ruční zadávání probíhá klikáním levého tlačítka myši na středovou křivku tratě.

Po této segmentaci si lze volit jednotlivé uzly a zadávat v nich polohu v závislosti na pozici vůči středové čáře a velikost a směr první nebo druhé derivace. Také si lze vybrat mezi třemi různými typy křivek: Přímkou, kubickým interpolačním splinem a interpolačním splinem 5. stupně. Jak bylo popsáno výše z těchto tří křivek je spojitost křivosti zajištěna vždy jen mezi křivkami 5. stupně, avšak vhodným nasměrováním derivací lze teoreticky dosáhnout i spojitosti mezi jinými typy křivky. Výsledná vykreslená křivka je však interpolována takovým způsobem, aby byl každému bodu středové čáry přidělen bod, který by byl průsečíkem kolmice vedené z tohoto bodu vůči středové čáře a vypočtené křivky.

Výstupy z této aplikace, tedy jak konfiguraci uzlových bodů, tak výslednou křivku, lze uložit do csv souboru exportovacími tlačítky.



## Nedostatky aplikace

Aplikace má několik nedostatků, někdy se stává, že dojde k chybě během zmíněné interpolace a spadne. Také zde není žádná kontrola míry křivosti výsledné křivky, což určitě rozšíření, které by se hodilo, bohužel z časových důvodů již nebylo implementováno. Dalším nedostatkem se ukázala být špatně promyšlená reprezentace uzlových bodů, ve skutečnosti jsou chápány jako segmenty s počátkem a koncem, avšak takovýto objekt se v konečném důsledku ukázal být zbytečně složitý. Další slabinou aplikace je funkce *Derivace*, ve které jsem implementoval tříbodovou metodu numerické derivace, avšak v krajních bodech počítá nepřesné výsledky.

## Možná vylepšení

Možná rozšíření aplikace by se daly shrnout do tří hlavních bodů:

- Další typy křivek
- Rychlostní profil pro vozidlo
- Optimalizace trajektorie

První bod v sobě skrývá potíž s přeprogramováním funkcí v rozhraní. Například  $\eta$ -spliny by totiž měly jako vstupní parametr kromě pozice interpolačních bodů a směru derivace také křivost a právě vektor parametrů  $\vec{\eta}$ , který by také ovlivňoval jejich tvar. Avšak na tyto možnosti není ovládací panel vůbec přizpůsoben.

Rychlostní profil se mi téměř podařilo implementovat, narazil jsem však na chybu ve výpočtu při vyjádření závislosti rychlosti na čase při řešení diferenciální rovnice pomocí Runge-Kuttovy metody.

Optimalizace trajektorie by pak již zřejmě byla nad rámec této práce. Spočívala by však v nalezení takové konfigurace parametrů, aby se našel vhodný kompromis při minimalizaci křivosti a délky tratě.

# A. Parametrické křivky

## A.1. Definice křivky

Na křivku můžeme nahlížet buď z fyzikálního hlediska, kdy ji chápeme jako dráhu pohybujícího se bodu v rovině či prostoru v závislosti na čase, nebo z geometrického hlediska, pak se na křivku díváme jako na jednoparametrickou množinu bodů v prostoru, které obdržíme jako funkční hodnoty bodové funkce jedné proměnné. Často je však výhodnější namísto bodů křivky uvažovat jejich polohové vektory, které jsou funkčními hodnotami vektorové funkce. Definice v následující kapitole jsou přejaty z [6] a [10].

**Definice A.1** (Vektorová funkce). *Nechť  $V_n$  je vektorové zaměření  $n$ -rozměrného euklidovského prostoru  $E_n$ , tj.  $V_n = Z(E_n) = \{\vec{XY}; X, Y, \in E_n\}$  a  $I$  je otevřený interval  $(a, b)$ , který se nazývá oborem parametrizace vektorové funkce. Potom zobrazení  $\mathbf{F}: I \rightarrow V_n$  se nazývá vektorová funkce na intervalu  $I$ .*

*Poznámka.* Volbou počátku  $P \in E$  můžeme v  $n$ -rozměrném euklidovském prostoru  $E$  každý bod  $X \in E$  ztotožnit s vektorem  $\vec{PX}$ . Můžeme tedy euklidovský prostor  $E_n$  i jeho zaměření pomocí souřadnic ztotožnit s  $\mathbb{R}^n$ .

Každému  $t \in I$  je tedy vektorovou funkcí přiřazen vektor. Reálné funkce

$$x_1 = x_1(t), \quad x_2 = x_2(t), \quad \dots, \quad x_n = x_n(t), \quad t \in I,$$

se nazývají souřadnicové funkce vektorové funkce jedné proměnné.

**Definice A.2.** Řekneme, že vektorová funkce  $\mathbf{F}: I \rightarrow V_n$  má v bodě  $t_0 \in I$  limitu a píšeme  $\mathbf{F}_0 = \lim_{t \rightarrow t_0} \mathbf{F}(t)$ , jestliže ke každému  $\varepsilon > 0$  existuje  $\delta > 0$  tak, že

$$|t - t_0| < \delta, \quad t \neq t_0, \quad \text{potom} \quad \|\mathbf{F}(t) - \mathbf{F}(t_0)\| < \varepsilon$$

Vektorová funkce  $\mathbf{F}: I \rightarrow V_n$  je spojitá v bodě  $t_0$ , jestliže  $\lim_{t \rightarrow t_0} \mathbf{F}(t) = \mathbf{F}(t_0)$ .

Řekneme, že vektorová funkce  $\mathbf{F}: I \rightarrow V_n$  je spojitá na intervalu  $I$ , právě když je spojitá ve všech bodech intervalu  $I$ .

**Definice A.3.** Derivací vektorové funkce v bodě  $t_0$  nazveme limitu  $\lim_{t \rightarrow t_0} \frac{\mathbf{F}(t) - \mathbf{F}(t_0)}{t - t_0}$  a značíme ji  $\frac{d\mathbf{F}(t_0)}{dt}$  nebo  $\mathbf{F}'(t_0)$ .

Vektorová funkce  $\mathbf{F}(t)$  je spojitá v bodě  $t_0 \in I$ , právě když jsou v tomto bodě spojitě všechny její složky, navíc řekneme, že je spojitá na intervalu  $I$ , pokud jsou na  $I$  spojitě všechny její složky. Stejně tak má derivaci v bodě  $t_0 \in I$ , právě tehdy když mají všechny její složky  $x_1(t)$ ,  $x_2(t)$ ,  $\dots$ ,  $x_n(t)$  derivaci v tomto bodě, přičemž platí

$$\frac{d\mathbf{F}(t_0)}{dt} = \left( \frac{dx_1(t_0)}{dt}, \frac{dx_2(t_0)}{dt}, \dots, \frac{dx_n(t_0)}{dt} \right).$$

Iterací můžeme zavést derivace vyššího řádu a jejich spojitosti. [6]

**Definice A.4.** Řekneme, že vektorová funkce  $\mathbf{F}: I \rightarrow V_n$  je třídy  $C^k$  na intervalu  $I$ , právě tehdy když má  $\mathbf{F}$  na  $I$  spojitě všechny derivace do  $k$ -tého řádu včetně.

**Definice A.5** (Křivka). *Křivka je každá souvislá podmnožina prostoru  $k \subset \mathbf{R}$ , která je spojitým obrazem intervalu  $I$ . V případě, že  $n=2$ , nazýváme křivku rovinnou, pro  $n=3$  prostorovou. Je-li analytickou reprezentací rovinné křivky vektorová funkce, která je definovaná, spojitá a minimálně třídy  $C^1$  na intervalu  $I$ , pak říkáme, že rovinná křivka dána vektorovou rovnicí*

$$\mathbf{C}(t) = (x(t), y(t)), \quad t \in I, \quad (\text{A.1})$$

obdobně pro  $n=3$

$$\mathbf{C}(t) = (x(t), y(t), z(t)), \quad t \in I. \quad (\text{A.2})$$

Rozepíšeme-li souřadnicové funkce křivky (A.1), obdržíme parametrické rovnice rovinné křivky

$$\begin{aligned} x &= x(t), \\ y &= y(t), \quad t \in I. \end{aligned} \quad (\text{A.3})$$

Analogicky pro prostorovou křivku

$$\begin{aligned} x &= x(t), \\ y &= y(t), \\ z &= z(t), \quad t \in I. \end{aligned} \quad (\text{A.4})$$

Je-li interval  $I = [a, b]$  uzavřený, hovoříme o oblouku křivky. [10] V dalším výkladu budeme pod pojmem křivka chápat oblouk křivky.

**Definice A.6** (Bod křivky). *Hodnotu polohového vektoru vektorové funkce  $\mathbf{F}$  pro  $\alpha \in I = [a, b]$*

$$\mathbf{F}(\alpha) = (x(\alpha), y(\alpha), z(\alpha)) = \mathbf{C}(\alpha)$$

*nazveme bod křivky. Hodnota parametru  $t = \alpha \in I = [a, b]$  se nazývá parametrická souřadnice bodu křivky. Orientace křivky je určena vektorovou funkcí, počáteční bod má parametrickou souřadnici  $a$ , koncový bod má parametrickou souřadnici  $b$ . Tyto body pak nazveme krajními body křivky.*

## A.2. Vlastnosti křivek

**Definice A.7** (Regulární a singulární bod křivky). *Bod křivky  $\mathbf{C}(t_0)$ ,  $t_0 \in I = [a, b]$  nazveme regulárním, jestliže pro první derivace rovnice křivky je vektor  $\mathbf{C}'(t_0) \neq \mathbf{0}$  a odpovídá-li mu právě jedna hodnota parametru  $t = t_0 \in (a, b)$ . Ostatní body nazveme singulárními.*

**Definice A.8** (Inflexní bod křivky). *Bod křivky  $\mathbf{C}(t_0)$ ,  $t_0 \in I = [a, b]$  nazveme inflexním, platí-li pro vektory  $\mathbf{C}'(t_0)$  a  $\mathbf{C}''(t_0)$ , že jsou lineárně závislé.*

**Definice A.9** (Tečný vektor). *Tečný vektor křivky  $\mathbf{C}(t)$ ,  $t \in I$  obdržíme jako derivaci její vektorové funkce*

$$\mathbf{C}'(t) = \left( \frac{dx(t)}{dt}, \frac{dy(t)}{dt}, \frac{dz(t)}{dt} \right) = (x'(t), y'(t), z'(t)). \quad (\text{A.5})$$

Jednotkový tečný vektor  $\mathbf{t}(t_0)$  v regulárním bodě  $\mathbf{C}(t)$  křivky  $\mathbf{C}(t_0)$  obdržíme jako

$$\mathbf{t}(t_0) = \frac{\mathbf{C}'(t_0)}{|\mathbf{C}'(t_0)|} \quad (\text{A.6})$$

Přímka určena bodem  $\mathbf{C}(t_0)$  a směrovým vektorem (A.6) se nazývá tečna křivky v bodě  $\mathbf{C}(t_0)$ .

**Definice A.10** (Vektor binormály). Vektorový součin první a druhé derivace vektorové funkce křivky  $\mathbf{C}(t)$ ,  $t \in I$

$$\mathbf{b}(t) = \mathbf{C}'(t) \times \mathbf{C}''(t)$$

nazveme vektorem binormály, její jednotkový vektor v bodě  $\mathbf{C}(t_0)$ ,  $t_0 \in I$  opět obdržíme znormováním vektoru binormály a dosazením bodu  $t_0$ ,

$$\mathbf{b}(t_0) = \frac{\mathbf{C}'(t_0) \times \mathbf{C}''(t_0)}{|\mathbf{C}'(t_0) \times \mathbf{C}''(t_0)|}. \quad (\text{A.7})$$

Přímka určena bodem  $\mathbf{C}(t_0)$  a směrovým vektorem  $\mathbf{b}(t_0)$  se nazývá binormála křivky v bodě  $\mathbf{C}(t_0)$ .

**Definice A.11.** Vektor normály je definován jako vektorový součin vektoru binormály a tečného vektoru křivky  $\mathbf{C}(t)$ ,  $t \in I$  v bodě  $\mathbf{C}(t_0)$ . Jednotkový vektor normály  $\mathbf{n}(t_0)$  v bodě  $\mathbf{C}(t_0)$  potom obdržíme jako

$$\mathbf{n}(t_0) = \mathbf{b}(t_0) \times \mathbf{t}(t_0) \quad (\text{A.8})$$

Přímka určena bodem  $\mathbf{C}(t_0)$  a směrovým vektorem  $\mathbf{n}(t_0)$  se nazývá normála křivky v bodě  $\mathbf{C}(t_0)$ .

**Definice A.12.** Normovaný pravoúhlý a kladně orientovaný (pravotočivý) trojhran tvořený v regulárním bodě  $\mathbf{C}(t_0)$ ,  $t_0 \in I$  křivky  $\mathbf{C}(t)$ ,  $t \in I$  jednotkovým tečným vektorem  $\mathbf{t}(t_0)$ , jednotkovým vektorem normály  $\mathbf{n}(t_0)$  a jednotkovým vektorem binormály  $\mathbf{b}(t_0)$  se nazývá Frenetův průvodní trojhran křivky.

**Definice A.13.** Nechť  $\mathbf{C}(t_0)$ ,  $t_0 \in I$  je regulární bod křivky  $\mathbf{C}(t)$ ,  $t \in I$ . Rovina, která je kolmá k tečně, určená normálou a binormálou v bodě  $\mathbf{C}(t_0)$  se nazývá normálová rovina  $\nu$ , dána rovnicí

$$\nu: (\mathbf{N} - \mathbf{C}(t_0)) \cdot \mathbf{t}(t_0) = 0, \quad (\text{A.9})$$

kde  $\mathbf{N}$  je polohový vektor bodu normálové roviny  $\nu$ .

Rovina, kolmá k normále, určená binormálou a tečnou v bodě  $\mathbf{C}(t_0)$  se nazývá rektifikační rovina  $\rho$ , dána rovnicí

$$\rho: (\mathbf{R} - \mathbf{C}(t_0)) \cdot \mathbf{n}(t_0) = 0, \quad (\text{A.10})$$

kde  $\mathbf{R}$  je polohový vektor bodu rektifikační roviny  $\rho$ .

Rovina, kolmá k normále, určená tečnou a normálou v bodě  $\mathbf{C}(t_0)$  se nazývá rektifikační rovina  $\omega$ , dána rovnicí

$$\omega: (\mathbf{W} - \mathbf{C}(t_0)) \cdot \mathbf{n}(t_0) = 0, \quad (\text{A.11})$$

kde  $\mathbf{W}$  je polohový vektor bodu rektifikační roviny  $\omega$ .

Pomocí Frenetova průvodního trojhranu křivky můžeme vyjádřit první křivost (flexi), která v regulárním bodě  $\mathbf{C}(t_0)$  charakterizuje odchylku křivky  $k$  od tečny  $t_{t_0}$ , a druhou křivost (torzi), která v regulárním bodě  $\mathbf{C}(t_0)$  charakterizuje odchylku křivky  $k$  od oskulační roviny  $\omega_{t_0}$ . [11]

**Definice A.14** (První křivost křivky). *Nechť  $\mathbf{C}(t_0)$ ,  $t \in I$  je regulární bod křivky  $\mathbf{C}(t)$ ,  $t \in I$ , potom číslo*

$$\kappa(t_0) = \frac{|\mathbf{C}'(t_0) \times \mathbf{C}''(t_0)|}{|\mathbf{C}'(t_0)|^3} \quad (\text{A.12})$$

*nazveme první křivostí  $\kappa(t_0)$  v bodě  $\mathbf{C}(t_0)$ .*

*Obrácenou hodnotu první křivosti  $r(t_0) = \frac{1}{\kappa(t_0)}$  nazýváme poloměrem křivosti v bodě  $\mathbf{C}(t_0)$ . Bod  $\mathbf{S}(t_0) = \mathbf{C}(t_0) + r(t_0)\mathbf{n}(t_0)$  ležící v oskulační rovině  $\omega_{t_0}$  na polopřímce určené bodem  $\mathbf{C}(t_0)$  a vektorem  $\mathbf{n}(t_0)$  ve vzdálenosti  $r(t_0)$  od bodu  $\mathbf{C}(t_0)$  se nazývá střed křivosti v bodě  $\mathbf{C}(t_0)$ . Střed křivosti  $\mathbf{S}(t_0)$  a poloměr křivosti  $r(t_0)$  určuje oskulační kružnici v bodě  $\mathbf{C}(t_0)$ .*

Pokud je křivost  $\kappa(t)$ ,  $t \in I$  na intervalu  $I = [a, b]$  identicky rovná nule, tedy

$$\kappa(t) = 0, \quad \forall t \in I,$$

potom je křivka  $\mathbf{C}(t)$  přímka. [11]

**Definice A.15** (Druhá křivost křivky, torze). *Nechť  $\mathbf{C}(t_0)$  je neinflexní bod křivky  $\mathbf{C}(t)$ . Potom číslo*

$$\tau(t_0) = \frac{\mathbf{C}'(t_0) \cdot (\mathbf{C}''(t_0) \times \mathbf{C}'''(t_0))}{|\mathbf{C}'(t_0) \times \mathbf{C}''(t_0)|}, \quad (\text{A.13})$$

*nazýváme druhou křivostí  $\tau(t_0)$  křivky  $\mathbf{C}(t)$  v bodě  $\mathbf{C}(t_0)$*

Pokud je torze  $\tau(t)$ ,  $t \in I$  na intervalu  $I = [a, b]$  identicky rovná nule,

$$\tau(t) = 0, \quad \forall t \in I,$$

potom je křivka  $\mathbf{C}(t)$  rovinná. [11]

Pro rovinné křivky je první křivost jedna z jejich nejzákladnějších vlastností. V rovině se (A.12) zredukuje na rovnici

$$\kappa(t_0) = \frac{|x'(t_0) * y''(t_0) - y'(t_0) * x''(t_0)|}{(x'(t_0)^2 + y'(t_0)^2)^{\frac{3}{2}}}. \quad (\text{A.14})$$

Bez absolutní hodnoty v čitateli dostaneme orientovanou křivost a ta určuje tvar křivky jednoznačně až na posunutí, což je znění základní věty o křivkách v rovině.

**Věta A.1** (Základní věta o rovinných křivkách). *Mějme rovinnou křivku  $\mathbf{C}(s)$   $s \in (a, b)$ , parametrizovanou obloukem, tzn. velikost tečného vektoru křivky se v každém bodě rovná jedné, jejíž křivost je po částech spojitá funkce  $\kappa(a, b) \rightarrow \mathbb{R}$ , potom je křivka parametrizována:*

$$\begin{aligned} \mathbf{C}(s) &= \left( \int_0^s \cos(\theta(t))dt + c, \int_0^s \sin(\theta(t))dt + d \right) \\ \theta(s) &= \int_0^s \kappa(t)dt + \theta_0, \end{aligned} \quad (\text{A.15})$$

kde  $c, d, \theta_0$  jsou integrační konstanty a  $s$  je dráha křivky.

**Definice A.16** (Spojitost křivky). *Křivka  $\mathbf{C}(t)$ ,  $t \in I$  má spjitost  $k$ -tého řádu v bodě  $t_0 \in I$ , právě když má její vektorová funkce v bodě  $t_0$  spjitost  $k$ -tého řádu, potom řekneme, že je křivka  $C^k$  spojitá v bodě  $t_0 \in I$ .*

*Křivka  $\mathbf{C}(t)$ ,  $t \in I$  má spjitost  $k$ -tého řádu na intervalu  $I$ , právě když má její vektorová funkce na  $I$  spjitost  $k$ -tého řádu a řekneme, že taková křivka je  $C^k$  spojitá na intervalu  $I$ .*

**Definice A.17** (Spojitost napojení dvou křivek). *Nechť jsou dány křivky  $\mathbf{C}(t)$ ,  $t \in I = [a, b]$ ,  $\mathbf{D}(u)$ ,  $u \in J = [c, d]$ , které jsou na  $I$ , resp.  $J$  spojitě a  $k$ -krát diferencovatelné. Řekneme, že křivka  $\mathbf{D}(u)$  je napojena svým počátečním bodem  $c$  na koncový bod  $b$  křivky  $\mathbf{C}(t)$  s  $C^k$  spjitostí, pokud platí*

$$\mathbf{C}^{(i)}(b) = \mathbf{D}^{(i)}(c), \quad i = 0, 1, \dots, k. \quad (\text{A.16})$$

*Toto napojení dvou křivek označujeme jako parametrickou spjitost.*

## B. Skript pro výpočet $\eta$ -spline křivky

```
P1 = [5 4];
P2 = [10 8];
alfa = pi/4;
beta = 4/3*pi;
kappa1 = 0;
kappa2 = 0;

eta= [50 50 0 0];

P=[0 0 0 0;
    50 15 0 0;
    98.76 23.19 0.5 1/50;
    124.67 63.53 1.5 1/50;
    104.72 107.12 2.50 1/50];
[C1,u1] = etaSplineInterpolation(P(1,1:2), P(2,1:2),P(1,3), P(2,3), ...
    P(1,4), P(2,4), eta);
[C2,u2] = etaSplineInterpolation(P(2,1:2), P(3,1:2), P(2,3), P(3,3), ...
    P(2,4), P(3,4), eta);
[C3,u3] = etaSplineInterpolation(P(3,1:2), P(4,1:2), P(3,3), P(4,3), ...
    P(3,4), P(4,4), eta);
[C4,u4] = etaSplineInterpolation(P(4,1:2), P(5,1:2), P(4,3), P(5,3), ...
    P(4,4), P(5,4), eta);

C = [C1; C2(2:end,:); C3(2:end,:); C4(2:end,:)];
u = [u1,u1(end) + u2(2:end),u1(end)+u2(end) + u3(2:end), u1(end) + ...
    u2(end) + u3(end) + u4(2:end)];
indeces = [1,length(C1), length(C1)*2-1, length(C1)*3-2, length(C1)*4-3];
kappa = Krivost2DKrivky(C,u);

figure
hold on
plot(P(:,1), P(:,2), 'o', 'color', 'red');
plot(C1(:,1), C1(:,2));
plot(C2(:,1), C2(:,2));
plot(C3(:,1), C3(:,2));
plot(C4(:,1), C4(:,2));

figure
hold on
plot(u, kappa);
plot(u(indeces), kappa(indeces), 'o');

figure
eta=[10, 10, 1, -1];
hold on
for ii = 1:15
C=etaSplineInterpolation(P(1,1:2), P(2,1:2), P(1,3), P(2,3), P(1,4), ...
    P(2,4), eta);
plot(C(:,1),C(:,2))
eta(3) = eta(3)+sign(eta(3))*50;
```

```

eta(4) = eta(4) + sign(eta(4))*50;
end

figure
eta=[1, 1, 1, -1];
hold on
for ii = 1:15
C=etaSplineInterpolation(P(1,1:2), P(2,1:2), P(1,3), P(2,3), P(1,4), ...
    P(2,4), eta);
plot(C(:,1),C(:,2))
eta(1) = eta(1)+sign(eta(1))*50;
eta(2) = eta(2) + sign(eta(2))*50;
eta(3) = eta(3)+sign(eta(3))*50;
eta(4) = eta(4) + sign(eta(4))*50;
end

function [C,t] = etaSplineInterpolation(P0, ...
    P1,thetaA,thetaB,kappaA,kappaB,eta)
%P0 = [x0, y0]
%P1 = [x1, y1]
x0 = P0(1);
y0 = P0(2);
x1 = P1(1);
y1 = P1(2);
    x= etasplinuxkoefficients(x0,x1, thetaA, thetaB,kappaA,kappaB,eta);
    y= etasplineykoefficients(y0,y1, thetaA, thetaB,kappaA,kappaB,eta);

    cx = @(t) x(1) + x(2)*t+x(3)*t.^2+x(4)*t.^3+x(5)*t.^4+x(6)*t.^5;
    cy = @(t) y(1) + y(2)*t+y(3)*t.^2+y(4)*t.^3+y(5)*t.^4+y(6)*t.^5;
    t=linspace(0,1);
    C = [cx(t)', cy(t)'];
end

function [x] = etasplinuxkoefficients(x0,x1,thetaA,thetaB, ...
    kappaA,kappaB, eta)
    kappaA,kappaB, eta)
if(length(eta) ≠ 4 || eta(1)<0 || eta(2) <0)
    ME = MException('eta:Cannotbenegative', ...
        'Eta must have four components, and first two have to be positive');
    throw(ME);
end
eta1 = eta(1);
eta2 = eta(2);
eta3 = eta(3);
eta4 = eta(4);
x(1) = x0;
x(2) = eta1*cos(thetaA);
x(3) = 1/2*(eta3*cos(thetaA) - eta1^2*kappaA*sin(thetaA));
x(4) = 10*(x1-x0) - (6*eta1+3/2*eta3)*cos(thetaA) - ...
    (4*eta2 - 1/2 * eta4)*cos(thetaB) + 3/2 * eta1^2 * ...
    kappaA*sin(thetaA)...
    - 1/2* eta2^2*kappaB*sin(thetaB);
x(5) = -15*(x1-x0) + (8*eta1 + ...
    3/2*eta3)*cos(thetaA)+(7*eta2-eta4)*cos(thetaB)-...
    3/2*eta1^2*kappaA*sin(thetaA)+eta2^2*kappaB*sin(thetaB);

```



```

x(6) = 6*(x1-x0) - (3*eta1 + ...
    1/2*eta3)*cos(thetaA)-(3*eta2-1/2*eta4)*cos(thetaB)+...
    1/2*eta1^2*kappaA*sin(thetaA)-1/2*eta2^2*kappaB*sin(thetaB);
end

function [y] = etasplineykoeficients(y0,y1,thetaA,thetaB, ...
    kappaA,kappaB, eta)
if(length(eta) ≠ 4 || eta(1)<0 || eta(2) <0)
    ME = MException('eta:Cannotbenegative', ...
        'Eta must have four components, and first two have to be positive');
    throw(ME);
end
eta1 = eta(1);
eta2 = eta(2);
eta3 = eta(3);
eta4 = eta(4);
y(1) = y0;
y(2) = eta1*sin(thetaA);
y(3) = 1/2*(eta3*sin(thetaA) + eta1^2*kappaA*cos(thetaA));
y(4) = 10*(y1-y0) - (6*eta1+3/2*eta3)*sin(thetaA) - ...
    (4*eta2 - 1/2 * eta4)*sin(thetaB) - 3/2 * eta1^2 * ...
    kappaA*cos(thetaA)...
    + 1/2* eta2^2*kappaB*cos(thetaB);
y(5) = -15*(y1-y0) + (8*eta1 + ...
    3/2*eta3)*sin(thetaA)+(7*eta2-eta4)*sin(thetaB)+...
    3/2*eta1^2*kappaA*cos(thetaA)-eta2^2*kappaB*cos(thetaB);
y(6) = 6*(y1-y0) - (3*eta1 + ...
    1/2*eta3)*sin(thetaA)-(3*eta2-1/2*eta4)*sin(thetaB)-...
    1/2*eta1^2*kappaA*cos(thetaA)+1/2*eta2^2*kappaB*cos(thetaB);
end

```

# Literatura

- [1] ABBENA, E., SALAMON, S., GRAY, A. *Modern Differential Geometry of Curves and Surfaces with Mathematica*. 3rd edition. 2006. ISBN-13: 978-1584884484
- [2] ALEOTTI, J., CASELLI, S., MACCHEROZZI, G. *Trajectory Reconstruction with NURBS Curves for Robot Programming by Demonstration*. [Online] University of Parma, Italy, 2005. [Cit: 20.6. 2020] DOI: 10.1109/CIRA.2005.1554257 Dostupné z: <http://robocare.istc.cnr.it/papers/a8026.pdf>
- [3] BRAGHIN, F., aj. *Race Driver Model* Computers and Structures 86 [Online]. s. 1503-1516, 2008. [Cit. 13.6.2020] Dostupné z: <https://www.sciencedirect.com/science/article/abs/pii/S0045794908000163>
- [4] CARDAMONE, L. aj. *Searching for the Optimal Racing Line Using Genetic Algorithms*. Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games. Dublin, 2010. s. 388 - 394. doi: 10.1109/ITW.2010.5593330.
- [5] CASANOVA, D. *On Minimum Time Vehicle Manoeuvring: The Theoretical Optimal Lap* [Online] Cranfield University, 200. [Cit. 20.6. 2020] Dostupné z: [https://dspace.lib.cranfield.ac.uk/bitstream/1826/1091/1/PAT\\_TAYLOR\\_CASANOVA\\_THESIS.P](https://dspace.lib.cranfield.ac.uk/bitstream/1826/1091/1/PAT_TAYLOR_CASANOVA_THESIS.P)
- [6] DOUPOVEC, M. *Diferenciální geometrie a tenzorový počet*. Brno: PC-DIR Real, 1999. Učební texty vysokých škol. ISBN 80-214-1470-7.
- [7] DUBINS, L.E. *On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents* American Journal of Mathematics. The Johns Hopkins University Press, 1957. DOI: 10.2307/2372560
- [8] FRAICHARD, T., SCHEUER, A. *Continuous-Curvature Path Planning for Car-Like Vehicles*. Conference: Intelligent Robots and Systems, 1997. 997 - 1003 vol.2. IROS.1997.655130
- [9] GIESE, A. *A Comprehensive, Step-by-Step Tutorial on Computing Dubin's Curves*. [Online]. 2012 [Cit. 23.06.2020] Dostupné z: <https://gieseanw.files.wordpress.com/2012/10/dubins.pdf>
- [10] LINKEOVÁ, I. *NURBS křivky: NeUniformní Racionální B-Spline křivky*., Praha: Nakladatelství ČVUT, 2007. ISBN 978-80-01-03893-2
- [11] LINKEOVÁ, I. *Základy počítačového modelování křivek a ploch*, Praha: Nakladatelství ČVUT, 2008. ISBN 978-80-01-04011-9
- [12] MARZBANI, H., JAZAR, R.N., FARD, M. *Better Road Design Using Clothoids*. Sustainable Automotive Technologies. Springer, Cham, 2014. ISBN 978-3-319-17999-5.
- [13] PIEGL, Les A., Tiller W. *The NURBS book. 2nd ed.* New York: Springer, 1997. Monographs in visual communication. ISBN 3-540-61545-8.

- [14] BIANCO, C. G., PIAZZI A. *Optimal trajectory planning with quintic  $G^2$ -splines*. [online] Conference: Intelligent Vehicles Symposium, 2000. [Cit. 20.6.2020] DOI: 10.1109/IVS.2000.898417 Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.700.9504rep=rep1type=pdf>
- [15] REEDS, J. A., SHEPP, L. A. *Optimal paths for a car that goes both forwards and backwards*. Pacific J. Math. 145, 1990, no. 2. s. 367-393. Dostupné z: <https://projecteuclid.org/euclid.pjm/1102645450>
- [16] TARUFFI, P. *The Technique of Motor Racing*. R. Bentley, 1988. ISBN: 0-837-60228-9
- [17] THEODOSIS, P. A. *Path Planning for an Automated Vehicle Using Professional Racing Techniques* [online]. 2014. [Cit. 13. 6. 2020]. Dostupné z: [https://stacks.stanford.edu/file/druid:zn992vv3694/Theodosis\\_thesis-augmented.pdf](https://stacks.stanford.edu/file/druid:zn992vv3694/Theodosis_thesis-augmented.pdf)
- [18] VELENIS, E., TSIOTRAS, P. *Minimum-Time Travel for a Vehicle with Acceleration Limits: Theoretical Analysis and Receding-Horizon Implementation*. J Optim Theory Appl [Online]. s. 275–296, 2008.[Cit. 13.6.2020]. Dostupné z: <https://doi.org/10.1007/s10957-008-9381-7>

# Seznam použitých zkratek a symbolů

$\mathbf{C}(t)$	Křivka parametrizovaná parametrem $t$
$\mathbf{C}'(t)$	Derivace křivky
$\omega(u)$	Váhová funkce NURBS křivky
$U$	Uzlový vektor
$\mathbf{C}(t, \vec{\eta})$	$\eta$ -spline křivka
$\vec{\eta}$	Vektor parametrů $\eta_1, \eta_2, \eta_3$ a $\eta_1$
$\mathbf{P}_i$	Řídicí bod spline křivky
$\mathbf{Q}_i$	Definiční bod interpolační spline křivky
$\kappa(t_0)$	Křivost křivky v bodě $\mathbf{C}(t_0)$
$\tau(t_0)$	Torze křivky v bodě $\mathbf{C}(t_0)$

# Seznam obrázků

2.1	Příklad Dubinsových křivek. Zdroj [9]. . . . .	3
2.2	Část klotoidy a její křivost. . . . .	4
2.3	Trajektorie zkonstruována z úseků rovinek a klotoid s nulovou délkou kruhového oblouku. . . . .	6
2.4	Vliv změny hodnot parametrů na tvar křivky. . . . .	8
2.5	Interpolační $\eta$ -spliny podle dat z [14] . . . . .	9
3.1	Aproximační B-spline a NURBS křivky, aktivní části křivek jsou vyznačeny červeně. . . . .	12
3.2	Grafy bazových funkcí otevřených B-spline a NURBS křivek, vlevo celý interval parametru, vpravo aktivní část bazových funkcí . . . . .	16
3.3	Otevřená B-spline křivka. . . . .	16
3.4	Ukotvené B-spline a NURBS křivky . . . . .	17
3.5	Uzavřená B-spline křivka. . . . .	18
3.6	Interpolační B-spline křivka. . . . .	19
3.7	Uzlová interpolační křivka 3. stupně. Řídící body <b>P</b> fialově. Pozn. první a poslední bod řídícího polygonu (čerchovaná čára) jsou totožné s definičními body <b>Q</b> . . . . .	21
3.8	Uzlová interpolační křivka 5. stupně. . . . .	21
4.1	Hlavní okno aplikace . . . . .	23
4.2	Panel <i>Načtení</i> . . . . .	23
4.3	Panel <i>Segmentace</i> . . . . .	24
4.4	Panel <i>Upravit uzel</i> . . . . .	24
4.5	Mapování bodů křivky (červená plná čára) vůči středové čáře (modrá čerchovaná). Interpolované v modrém kolečku. . . . .	25
4.6	Příklad validní a nevalidní křivky, na pravém obrázku prochází křivka mimo trať . . . . .	26
4.7	Příklad interpolačních křivek. . . . .	28
4.8	Příklad nekonvexního průběhu skalárního součinu . . . . .	29
4.9	Příklad trajektorie na testovací trati. . . . .	30
4.10	Příklad výstupu z algoritmu 4.2. V tomto příkladu je chyba ve výpočtu rychlosti z maximálního zrychlení. Znázorněný princip je však správný. . .	32