

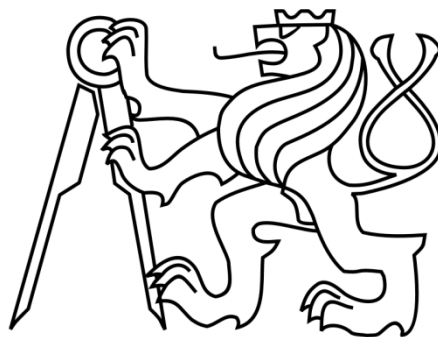
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

---

FAKULTA STROJNÍ

ÚSTAV MECHANIKY, BIOMECHANIKY A MECHATRONIKY

ODBOR MECHANIKY A MECHATRONIKY



## **DIPLOMOVÁ PRÁCE**

**Asistent pro nalezení optimální trajektorie  
při couvání s přívěsem**

**Praha, 2018**

**Bc. Vladislav Kocián**



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kocián** Jméno: **Vladislav** Osobní číslo: **420527**  
Fakulta/ústav: **Fakulta strojní**  
Zadávající katedra/ústav: **Ústav mechaniky, biomechaniky a mechatroniky**  
Studijní program: **Strojní inženýrství**  
Studijní obor: **Mechatronika**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Asistent pro nalezení optimální trajektorie při couvání s přívěsem**

Název diplomové práce anglicky:

**Assistance system for optimal trajectory planning when backing up with a trailer**

Pokyny pro vypracování:

- 1) Seznamte se s principy funkce asistenčních systémů pro couvání s přívěsem.
- 2) Připravte simulační model vozidla s přívěsem.
- 3) Navrhněte metodu nalezení optimální trajektorie při znalosti situace za vozidlem.
- 4) Ověřte navržené řešení na několika trajektoriích.
- 5) Navrhněte podobu asistenčního systému pro skutečné vozidlo.

Seznam doporučené literatury:

- 1) Lavallo, S.M., Planning Algorithms, Illinois, University of Illinois, 1999.
- 2) Roh, J., Chung, W., Reversing control of a car with a trailer using a driver assistance, IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), Seoul, 2010, pp. 99-104.
- 3) Bazjuk, L., Couvání vozidla s přívěsem, bakalářská práce, Praha, ČVUT v Praze, 2014.
- 4) Vrbský, L., Experimentální model vozidla s asistentem pro couvání s přívěsem, bakalářská práce, Praha, ČVUT v Praze, 2016.

Jméno a pracoviště vedoucí(ho) diplomové práce:

**Ing. Petr Beneš, Ph.D., odbor mechaniky a mechatroniky FS**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **25.04.2018**

Termín odevzdání diplomové práce: **17.08.2018**

Platnost zadání diplomové práce: \_\_\_\_\_

Ing. Petr Beneš, Ph.D.  
podpis vedoucí(ho) práce

prof. Ing. Milan Růžička, CSc.  
podpis vedoucí(ho) ústavu/katedry

prof. Ing. Michael Valášek, DrSc.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta



## Anotační list

<b>Jméno autora:</b>	Bc. Vladislav Kocián
<b>Název diplomové práce:</b>	Asistent pro nalezení optimální trajektorie při couvání s přívěsem
<b>Anglický název:</b>	Assistance system for optimal trajectory planning when backing up with a trailer
<b>Akademický rok:</b>	2017/2018
<b>Obor studia:</b>	Mechatronika
<b>Ústav/odbor:</b>	Ústav mechaniky, biomechaniky a mechatroniky Odbor Mechaniky a mechatroniky
<b>Vedoucí diplomové práce:</b>	Ing. Petr Beneš, Ph.D.
<b>Bibliografické údaje:</b>	Počet stran: 94 Počet obrázků: 45 Počet tabulek: 2 Počet příloh: 3 Příloha mimo výtisk – CD
<b>Klíčová slova:</b>	Optimalizace, trajektorie, Bézierova křivka, couvání, automobil, přívěs, asistenční systém
<b>Keywords:</b>	Optimization, trajectory, Bézier curve, reversing, car, trailer, assistance system



## **Abstrakt:**

Tato diplomová práce se zabývá návrhem asistenčního systému pro nalezení optimální trajektorie při couvání s přívěsem. V úvodních kapitolách je provedena rešerše dané problematiky, včetně popisu existujících asistenčních systémů. Je zvolen model vhodný pro popis systému automobilu s přívěsem a model je následně implementován do prostředí Matlab Simulink. V dalších částech práce je popsán návrh algoritmu pro vyhledání a následnou optimalizaci trajektorie. Na tuto část navazuje ukázka funkce algoritmu na několika mapách prostředí. Poslední kapitoly uvádí návrh podoby asistenta ve skutečném vozidle.

## **Klíčová slova:**

Optimalizace, trajektorie, Bézierova křivka, couvání, automobil, přívěs, asistenční systém

## **Abstract:**

This Master's thesis deals with designing of an assistance system for planning an optimal trajectory when backing up with a trailer. In the first section of the present thesis the research is made on the given issue, including description of existing solutions on assistance systems. A suitable model for description of a car-trailer system is selected and the model is consequently implemented into Matlab Simulink environment. In the following chapters of the thesis the design of an algorithm for path planning and optimization is described. The design of the algorithm is followed by the demonstration of use on several different environment maps. The last chapters of the thesis propose the design of an assistant for a real vehicle.

## **Keywords:**

Optimization, trajectory, Bézier curve, reversing, car, trailer, assistance system

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

V Praze, dne .....

.....

Podpis



## Poděkování

Chtěl bych touto formou poděkovat vedoucímu mé diplomové práce, panu Ing. Petru Benešovi, Ph.D., za cenné rady, vstřícný přístup a za čas investovaný do konzultací a do kontroly této práce.

Chtěl bych také poděkovat mým přátelům a mé přítelkyni za morální podporu.

Svůj vděk chci vyjádřit zejména mým rodičům, kteří mi pomohli určit si směr při putování životem a na této pouti jsou mi vždy podporou.

Děkuji.

## Obsah

1. Přehled použitých značek a zkratek.....	12
2. Úvod.....	17
3. Cíle práce .....	19
4. Existující řešení asistentů pro parkování s přívěsem.....	20
4.1. Trailer Assist (Volkswagen).....	20
4.2. Trailer Reverse Guidance a Pro Trailer Backup Assist™ (Ford).....	21
4.3. Další případy užití asistentů pro jízdu s přívěsem .....	22
5. Přehled poznatků z vědeckých publikací v problematice couvání s přívěsem.....	23
6. Navrhovaný způsob řešení .....	25
7. Sestavení kinematických modelů .....	27
7.1. Kinematický model automobilu .....	27
7.2. Kinematický model přívěsu .....	32
7.3. Kinematický model vlečky .....	34
7.4. Souhrnný popis sestavených kinematických modelů .....	38
7.5. Implementace kinematických modelů v prostředí Matlab Simulink .....	39
8. Vytvoření mapy prostoru pro parkovací manévr .....	41
9. Způsob popisu trajektorie – Bézierovy křivky.....	44
9.1. Teorie Bézierových křivek, definice, vlastnosti, příklady užití.....	44
9.2. Bézierova kubika .....	45
9.3. Bézierova křivka pátého řádu.....	53
10. Plánování a optimalizace trajektorie .....	55
10.1. Globální smyčka pro nalezení prvotní bezkolizní trajektorie .....	56
10.2. Detekce kolize.....	58
10.3. Lokální optimalizace navržené trajektorie.....	59
10.4. Minimalizace délky trajektorie .....	60

10.5.	Omezení křivosti trajektorie .....	63
10.6.	Zachování bezkoliznosti trajektorie.....	66
10.7.	Volba váhových koeficientů .....	68
11.	Vstupy do simulačního modelu .....	69
12.	Ověření navrženého řešení.....	73
13.	Návrh podoby asistenta pro skutečné vozidlo .....	77
14.	Závěr .....	82
15.	Seznam obrázků a tabulek .....	84
16.	Použité zdroje a literatura .....	86
17.	Seznam příloh .....	90

## 1. Přehled použitých značek a zkratk

ABS	.....	<i>Anti-lock Breaking System</i>
ESP	.....	<i>Electronic Stability Program</i>
ICR	.....	<i>Instant Centre of Rotation</i>
$B(t)$ [m]	.....	<i>Bézierova křivka</i>
$B'$ [m]	.....	<i>První derivace Bézierovy křivky podle jejího parametru <math>t</math></i>
$B''$ [m]	.....	<i>Druhá derivace Bézierovy křivky podle jejího parametru <math>t</math></i>
$B'_x(t)$ [m]	.....	<i>x-ová složka první derivace Bézierovy křivky podle jejího parametru <math>t</math></i>
$B''_x(t)$ [m]	.....	<i>x-ová složka druhé derivace Bézierovy křivky podle jejího parametru <math>t</math></i>
$B'_y(t)$ [m]	.....	<i>y-ová složka první derivace Bézierovy křivky podle jejího parametru <math>t</math></i>
$B''_y(t)$ [m]	.....	<i>y-ová složka druhé derivace Bézierovy křivky podle jejího parametru <math>t</math></i>
$P_i$ [m]	.....	<i><math>i</math>-tý řídicí bod Bézierovy křivky</i>
$b_{i,n}(t)$ [–]	.....	<i><math>i</math>-tý Bernsteinův polynom Bézierovy křivky řádu <math>n</math></i>
$x_i$ [m]	.....	<i>x-ová souřadnice tělesa <math>i</math> v prostoru</i>
$y_i$ [m]	.....	<i>y-ová souřadnice tělesa <math>i</math> v prostoru</i>
$\dot{x}_i$ [ $ms^{-1}$ ]	.....	<i>Derivace x-ové souřadnice tělesa <math>i</math> podle času</i>

$\dot{y}_i$ [ $ms^{-1}$ ]	.....	Derivace y-ové souřadnice tělesa i podle času
$\theta_0$ [rad]	.....	Úhel natočení automobilu v prostoru
$\theta_1$ [rad]	.....	Úhel natočení přívěsu/přední nápravy vlečky v prostoru
$\theta_2$ [rad]	.....	Úhel natočení zadní nápravy vlečky v prostoru
$\dot{\theta}_i$ [ $ms^{-1}$ ]	.....	Derivace úhlu natočení i-tého tělesa v prostoru
$\omega_0$ [rad/s]	.....	Úhlová rychlost automobilu
$\omega_1$ [rad/s]	.....	Úhlová rychlost přívěsu/přední nápravy vlečky
$\omega_2$ [rad/s]	.....	Úhlová rychlost zadní nápravy vlečky
$v_0$ [ $ms^{-1}$ ]	.....	Rychlost automobilu
$v_1$ [ $ms^{-1}$ ]	.....	Rychlost přívěsu/přední nápravy vlečky
$v_2$ [ $ms^{-1}$ ]	.....	Rychlost zadní nápravy vlečky
$CF$ [–]	.....	Celková cílová funkce
$CF_i$ [–]	.....	i-tá dílčí cílová funkce
$s$ [m]	.....	Celková délka trajektorie
$t$ [s]	.....	Čas
$\tau$ [s]	.....	Čas
$\kappa$ [ $m^{-1}$ ]	.....	Křivost trajektorie
$\kappa_{Dmax}$ [ $m^{-1}$ ]	.....	Maximální povolená křivost trajektorie

$M_K [-]$	.....	<i>Matice kolize</i>
$M_P [-]$	.....	<i>Matice překážek</i>
$M_T [-]$	.....	<i>Matice trajektorie</i>
$t [-]$	.....	<i>Parametr Bézierovy křivky</i>
$n_p [-]$	.....	<i>Počet volných parametrů Bézierovy křivky</i>
$R [m]$	.....	<i>Poloměr otáčení</i>
$R_{Dmin} [m]$	.....	<i>Minimální povolený poloměr otáčení</i>
$p_f [-]$	.....	<i>Převodový poměr převodky řízení</i>
$\eta_1, \eta_2 [rad]$	.....	<i>Úhel tečny trajektorie v počátku, konci trajektorie</i>
$\varphi_{ij} [rad]$	.....	<i>Úhel zalomení mezi tělesy <math>i</math> a <math>j</math></i>
$\phi_v [rad]$	.....	<i>Úhel natočení volantu</i>
$\phi [rad]$	.....	<i>Úhel natočení předních kol automobilu</i>
$k_i [-]$	.....	<i>Váhový koeficient příslušející <math>i</math>-té cílové funkci</i>
$\vec{p} [m]$	.....	<i>Vektor volných parametrů trajektorie</i>
$\Delta_1, \Delta_2 [m]$	.....	<i>Vzdálenost mezi prvními dvěma/posledními dvěma řídicími body Bézierovy křivky</i>
$L [m]$	.....	<i>Rozvor náprav automobilu</i>
$L_0 [m]$	.....	<i>Vzdálenost mezi zadní nápravou automobilu a místem spojení automobilu s přívěsem/vlečkou</i>

$L_1$ [m]	.....	Vzdálenost mezi místem spojení automobilu s přívěsem/vlečkou a nápravou přívěsu/přední nápravou vlečky
$L_2$ [m]	.....	Vzdálenost mezi přední a zadní nápravou vlečky
$L_R$ [m]	.....	Rozchod kol přední nápravy automobilu





## 2. Úvod

V současné době dochází s ohledem na rapidní vývoj technologií obzvláště v oboru elektrotechniky k velkému nárůstu elektronických komponent osobních automobilů. Za ty základní lze v dnešní době považovat řídicí jednotky a datové sběrnice, na které se nabalují stále nové systémy, zajišťující jak bezpečnost, tak i pohodlí a zábavu posádky osobního automobilu. Mezi základní elektronické bezpečnostní prvky automobilu patří například ABS a ESP. O zábavu posádky se stará tzv. „Infotainment“ (z anglických *Information* – informace a *Entertainment* – zábava), který je dostupný ve většině moderních aut v podobě dotykové obrazovky a sdružuje funkce jako jsou rádio, přehrávač hudby, nastavení klimatizace, navigace a další – příklad takového systému uvádí Obr. 1. Pohodlí řidiče umožňuje řada systémů, mimo samozřejmý posilovač řízení se může jednat o spoustu sekundárních funkcí a asistenčních systémů, lze zmínit například adaptivní tempomat, elektronické polohování světlometů, automatické stěrače, asistent pro jízdu v pruhu a mnoho jiných.



Obr. 1 – Infotainment automobilů Škoda (fdrive.cz)

O úspěšnosti konkrétního modelu automobilu na trhu může rozhodovat právě tato dodatečná výbava, důsledkem toho je snaha ze strany výrobců nabízet zákazníkům co nejširší škálu možných nadstandardních služeb a systémů. Mezi takové systémy patří v dnešní době už i asistenty pro jízdu s přívěsem.

Jízda s přívěsem vpřed se víceméně neliší od jízdy automobilem bez přívěsu, řidič pouze musí mít na mysli větší prostorové dispozice řízené soupravy, v zatáčkách případně mírně

upravit styl průjezdu. Jako mnohem složitější se jeví úloha couvání s přívěsem, která může být zprvu pro řidiče značně neintuitivní a vyžaduje určité zkušenosti. Zacouvat na určité místo s vozem s dvěma nápravami (vlečkou) je pro nezkušeného řidiče úloha téměř nezvládnutelná. Problémy plynou z toho, že při couvání je úhel mezi automobilem a přívěsem nestabilní a je třeba ho regulovat zásahy řidiče, v opačném případě narůstá, až nakonec dojde ke kolizi přívěsu s automobilem, či dosažení maximálního možného úhlu, který bývá limitován mechanickým dorazem. Cílem asistenta pro jízdu s přívěsem je usnadnění manévru couvání. Toho lze v závislosti na složitosti asistenta dosáhnout různými způsoby. Asistent může podávat řidiči pokyny pro ovládání vozidla, může stabilizovat úhel mezi automobilem a přívěsem s cílem, aby jel přívěs v určitém směru, nebo může v nejsložitějších verzích sám připravit trajektorii pro průjezd a tu poté autonomně projet.

### 3. Cíle práce

Cílem práce je vytvoření návrhu asistenta pro plánování trajektorií při couvání automobilu s přívěsem. Z toho plynou následující dílčí cíle: seznámení se s funkcemi existujících asistenčních systémů, sestavení matematických vztahů pro popis modelovaného systému, návrh algoritmu pro nalezení a optimalizaci trajektorie a následné ověření jeho funkčnosti. Dále uvedu souhrn zamýšleného postupu řešení.

V první části této diplomové práce provedu rešerši existujících řešení asistentů pro couvání s přívěsy, uvedu jejich princip funkce a klíčové vlastnosti. Pokusím se vyhledat vědecké články a práce zabývající se danou problematikou a zvážím možné postupy při návrhu vlastního jízdního asistenta.

V další části práce se zaměřím na sestavení vhodného kinematického modelu pro popis soustavy automobilu s přívěsem, popřípadě s vlečkou. Sestavený model musí dostatečně přibližovat realitu pro účely parkování, nemusí popisovat komplexní dynamiku jízdy autem při vyšších rychlostech.

Dále bude mým cílem vytvořit algoritmus pro nalezení trajektorie potřebné k provedení parkovacího manévru. Budu vycházet ze znalosti situace za vozidlem, tudíž znám polohu překážek, kterým je potřeba se vyhnout. Nalezenou cestu budu dále optimalizovat pro co nejefektivnější možnost průjezdu.

V závěrečné části práce propojím za účelem simulace kinematický model s vypočítanou trajektorií a ověřím její průjezdnost. Zároveň získám ze simulace kinematické veličiny automobilu potřebné pro správné sledování trajektorie přívěsem, především se bude jednat o úhlovou rychlost automobilu, ze které budu získávat potřebné natočení volantu.

## 4. Existující řešení asistentů pro parkování s přívěsem

Na trhu existuje několik řešení asistentů pro jízdu s přívěsem, ve většině případů se jedná o asistenty implementované přímo v určitých modelech automobilů a jsou nabízeny jako příplatková výbava. Mezi vědeckými publikacemi lze také dohledat řadu článků, zabývajících se danou problematikou, většina z nich pojednává o strategiích řízení soustavy automobilu s přívěsem, případně s několika přívěsy. V následující části uvedu několik příkladů asistenčních systémů dodávaných jako příplatková výbava.

### 4.1. Trailer Assist (Volkswagen)

Automobilka Volkswagen uvedla svůj asistenční systém pro jízdu s přívěsem jako jedna z prvních v roce 2014 [1]. Spojuje již dříve vyvinuté technologie parkovacího asistenta (Park Assist – 2006) a kamery sledující prostor za vozidlem za účelem usnadnění parkovacího manévru (Rear View – 2008).

Na základě zpracování obrazu ze zadní kamery automobilu se vyhodnocuje úhel odklonu přívěsu od automobilu v horizontální rovině. Do systému není třeba zadávat údaje o taženém zařízení, po průjezdu několika zatáček se systém sám „učí“ a vyhodnotí klíčové kinematické vlastnosti taženého zařízení. Při provádění samotného manévru couvání přebírá kontrolu nad volantem elektromechanický posilovač řízení a řidič pouze určuje rychlost vozidla. Směr couvání přívěsu je nastavován polohováním ovládacího prvku (Volkswagen nemá dedikovaný ovládací prvek, ale využívá joystick pro polohování zpětných zrcátek, který při zapnutí funkce Trailer Assist řídí přívěs). Řidič může nastavit požadovaný úhel mezi autem a přívěsem a systém řídí natočení kol přední nápravy automobilu tak, aby nastaveného úhlu zalomení bylo dosaženo na co nejkratší trajektorii. Je nutno podotknout, že nastavovanou veličinou je úhel zalomení mezi automobilem a přívěsem, ne absolutní natočení přívěsu v prostoru. Po srovnání přívěsu do požadovaného směru v prostoru je tedy třeba upravit úhel zalomení na nulovou hodnotu, poté bude asistent stabilizovat úhel zalomení tak, aby celá souprava jela rovně. Ilustraci poskytuje Obr. 2. Manévr couvání lze dle potřeby přerušit jízdou dopředu při nutnosti srovnat přívěs, po zařazení zpáteční

rychlosti pokračuje asistent ve své funkci [2], [3].



Obr. 2 – Ukázka funkce systému Trailer Assist (převzato z [3])

## 4.2. Trailer Reverse Guidance a Pro Trailer Backup Assist™ (Ford)

Asistenty pro jízdu s přívěsem jako možnou doplňkovou výbavu nabízí k některým svým modelům i automobilka Ford. Vyvíjí je ve spolupráci se společností Magna, která na svých webových stránkách uvádí systémy s názvem „Trailer Angle Detection“ [4] které mají řidiče upozorňovat na polohy, při kterých již nelze bez přerušení couvání stabilizovat úhel zalomení přívěsu, či polohy, při kterých hrozí bezprostřední kolize přívěsu s automobilem. Systémy pracují na základě zpracování obrazu, přičemž je na tažené zařízení třeba přilepit šachovnicovou nálepku (Obr. 3 a Obr. 4), která slouží k určení orientace přívěsu vzhledem k tažnému zařízení. Na rozdíl od systému automobilky Volkswagen je třeba před použitím asistenta zadat několik rozměrů, které určují polohu nálepky vůči kameře a klíčové rozměry přívěsu [5].



Obr. 3 – Šachovnicová nálepka umístěná na oji přívěsu (převzato z [5])

Obr. 4 – Vzdálenost mezi zadní parkovací kamerou a nálepkou (převzato z [5])

Systém Trailer Reverse Guidance je dodáván s přídatnou kamerou, kterou je možné nainstalovat na libovolné tažené zařízení. Jeho funkce spočívá ve zprostředkování lepšího úhlu pohledu pro řidiče při couvání, kdy pomocí sedmi kamer umístěných na automobilu rekonstruuje 3D obraz okolí, čímž zlepšuje viditelnost. Na displeji také animačně vykresluje aktuální stav couvající soustavy a čarami naznačuje směr pohybu při stávajícím natočení volantu. Zobrazuje taktéž mezní úhly zalomení pro bezpečné couvání. Systém může řidiči ukazovat potřebnou polohu volantu pro jízdu přívěsem rovně vzad, tedy mu radí, jak stabilizovat úhel zalomení. Točení volantem však provádí sám řidič.

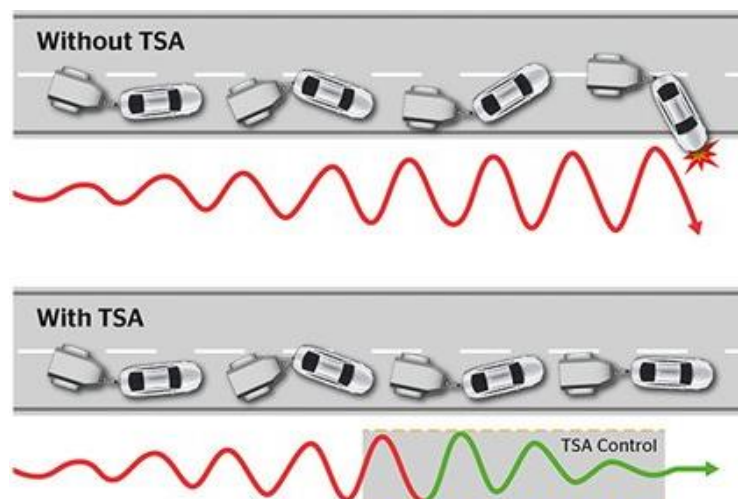
Systém Pro Trailer Backup Assist™ se velice podobá systému automobilky Volkswagen, funkcionalitou se neliší, drobné rozdíly lze najít v technické realizaci, kdy Ford využívá, stejně jako u jeho druhého zmíněného systému, ke zpracování obrazu šachovnicovou nálepkou. Na rozdíl od Volkswagenu také vyčlenil v kabině dedikovaný ovládací prvek pro ovládání asistenta. Je také třeba zadávat parametry přívěsu, které lze ukládat do paměti systému, který podporuje až 10 různých přívěsů s rozdílnými parametry [6], [7].

### 4.3. Další případy užití asistentů pro jízdu s přívěsem

Za zmínění stojí dále systémy automobilky Audi a Škoda, které se však funkcionalitou nijak neliší od provedení systému Trailer Assist od Volkswagenu, jelikož patří do koncernu Volkswagen Group, který sdružuje (mimo jiné) technologicky všechny tři uvedené automobilky [8].

Vlastní originální řešení nabízí automobilka Land Rover se systémem „*Transparent trailer technology*“, který však nelze užít s libovolným přívěsem. Asistenční systém využívá obrazu z kamer umístěných na speciálním přívěsu v kombinaci s obrazem z kamer umístěných na voze a rekonstruuje obraz, na kterém se přívěs jeví transparentní. Odstraňuje tím slepé úhly za přívěsem při jízdě vpřed, při jízdě vzad slouží pro snadnější provedení parkovacího manévru. V přívěsu jsou také instalovány vnitřní kamery a jiné senzory, které slouží ke kontrole přívěsu a jeho nákladu jak během jízdy, tak i po opuštění vozidla prostřednictvím mobilního telefonu. Mimo to nabízí systém „*Advanced Tow Assist*“, který nabízí stejné funkce, jako Trailer Assist, či Pro Trailer Backup Assist™ [9], [10].

Společnost Continental AG se zabývá v rámci elektronických stabilizačních programů systémem TSA (Trailer Stability Assist). Tento systém je spíše než asistentem prvkem bezpečnostním. Při jízdě s přívěsem může při vyšších rychlostech docházet k oscilacím soupravy, síly vznikající při tomto kmitání musí být reakčně zachyceny mezi vozovkou a pneumatikou. Při dosažení kritických hodnot dojde k odtržení kontaktu mezi koly a vozovkou a souprava přechází do nekontrolovatelného smyku – viz Obr. 5. Více o jízdních vlastnostech přívěsů se lze dočíst například ve zdroji [11]. Systém TSA je schopný zachytit vznik kmitání a vhodnými zásahy do točivého momentu motoru a přibrzděním jednoho či více kol upravit dynamiku soupravy, čímž dokáže automobil s přívěsem stabilizovat a utlumit účinky kmitání [12]. Společnost Continental AG se také zabývá vývojem asistenta pro parkování s přívěsem, který je možno ovládat mimo vozidlo prostřednictvím mobilního telefonu. Přes Wi-Fi spojení komunikuje mobilní telefon s asistentem v automobile, řidič má tedy možnost určovat směr pohybu přívěsu, zrychlovat a brzdit. Ovládání vozidla při couvání s přívěsem mimo kabinu s sebou nese výhodu mnohem lepší kontroly situace



kolem vozidla, kdy řidič není omezen výhledem z kabiny řidiče, či odkázán na kamerový systém [13].

Obr. 5 – Ukázka funkce asistenta pro stabilizaci přívěsu (převzato z [13])



## 5. Přehled poznatků z vědeckých publikací v problematice couvání s přívěsem

Mnoho vědeckých článků dostupných online se zabývá problematikou jízdy s přívěsem. Základním krokem je sestavení matematického modelu, který bude dostatečně věrně popisovat chování soupravy. Vhodná úroveň detailnosti modelu závisí na tom, jak věrně má daný model popisovat realitu a jakým účinkům je modelované těleso vystaveno v reálném světě. Hlavní otázkou je rozhodnout, zda je dostačující sestavit model kinematický, který nebere v potaz setrvačné účinky působící na modelovaný objekt ani pasivní účinky mezi jednotlivými tělesy, nebo je třeba sestavit model dynamický. Článek [14] se zabývá modelováním automobilu v souvislosti s úlohou sledování několika předem daných trajektorií. Jednotlivé trajektorie se od sebe liší, mají jiné průběhy křivostí, spojitostí derivace i rychlostních profilů podél trajektorie. Autor navrhuje různé taktiky řízení vozidla, které jsou založeny na různě složitých modelech. Srovnává následující typy modelů:

- geometrický,
- kinematický,
- dynamický.

S každým dalším stupněm modelu narůstá jeho složitost a jsou používány sofistikovanější metody řízení. V závěru práce je uvedena tabulka toho, jak si který daný model a způsob řízení „vedl“ na jednotlivých trajektoriích a navrhuje vhodnost modelů a způsobů řízení pro různé aplikace. Pro účely parkování se ukazuje, že kinematický model dostatečně věrně přibližuje realitu.

V článku [14] i mnoha dalších se pro popis automobilu s řízením podle Ackermanna používá zjednodušení z dvounápravového čtyřkolového automobilu na vozidlo jednostopé s pevnou zadní nápravou a přední nápravou říditelnou. Tento model uvažuje pohyb buď po přímce nebo po jiné obecné křivce, přičemž střed zadní nápravy dokonale rotuje kolem okamžitého středu otáčení, který je dán rozvorem náprav a polohou kol přední nápravy.

Dalším tématem, které má spojitost s návrhem asistenta je plánování trajektorie. Zdroj [15] je vhodnou knihou ke studiu dané problematiky. Uvádí řadu různých přístupů k plánování trajektorií, způsoby vyhodnocování kolizí s překážkami i kinematické pojmy z teorie pohybů pro popis pohybu těles ve 2D i 3D. Části s vysvětlením jednotlivých plánovacích algoritmů jsou proloženy pseudokódem pro systematictější představu o funkci algoritmu. V zdroji je též zmínka o tzv. NURBS křivkách, což jsou zobecněné Bézierovy křivky [16], příklad konkrétní aplikace jsem však nedohledal. Jelikož mám v úmyslu pro účely popisu trajektorie využívat právě Bézierovy křivky, zaměřil jsem se na jejich konkrétní aplikace v oblasti plánování trajektorií. Většina mnou dohledaných zdrojů je však používá pouze k úpravě trajektorie nalezené jiným plánovacím algoritmem. Například zdroj [17] je užívá pro zaoblení ostrých neprůjezdných zatáček. Zdroj [18] využívá k nalezení prvotní trajektorie

Voroného diagramy prostoru, čímž získá průjezdnou trajektorii v podobě několika na sebe navazujících úseček. Tuto trajektorii následně za účelem vyhlazení ostrých napojení jednotlivých úseků aproximuje Bézierovou křivkou.

Lze rovněž dohledat množství publikací, které se zabývají autonomní jízdou s taženým zařízením. V těchto publikacích jsou navrženy různé strategie řízení, např. článek [19] uvádí možnost využití jednoduchého PI regulátoru pro řízení úhlu zalomení nápravy přívěsu za účelem sledování předepsané trajektorie. Zdroj [20] uvádí návrh sofistikovaného řízení pro systém jednonápravového robota s přívěsem, je navržen regulátor založený na klouzavém řízení (SMC – sliding mode control) i adaptivní verze regulátoru (ASMC – adaptive dynamic sliding mode control) a dále CTC regulátor (computed torque control).



## 6. Navrhovaný způsob řešení

Za účelem návržení vlastního asistenčního systému pro couvání s přívěsem je třeba rozhodnout o způsobu realizace následujících bodů:

- jaký model systému zvolit,
- jakým způsobem popsat prostředí pro manévr,
- jakým způsobem popsat trajektorii,
- jakým způsobem vyhledávat trajektorii.

Na základě rešeršní studie jsem se rozhodl použít pro popis systému kinematický model, jelikož při předpokládaných nízkých rychlostech pohybu lze zanedbat dynamické účinky. Zvolený model musí umožňovat vyjádření kinematických veličin pro popis pohybu tělesa v prostoru (ideálně vždy polohu bodu a úhel natočení). Je třeba spojit rovnice pro popis pohybu automobilu s rovnicemi pro popis pohybu přívěsu a je třeba zachovat logiku systému, tudíž za vstupy lze považovat pouze rychlost automobilu a polohu volantu automobilu, jelikož přívěs je pouze pasivně poháněný člen.

Prostor, ve kterém je prováděn úkon couvání (či parkování), považuji za známý, získaný například na základě zpracování obrazu z kamer, jimiž bude automobil s vlečkou vybaven. Takový prostor je omezený dosahem kamer, avšak pro vypočítání trajektorie za účelem zajetí do parkovacího místa je dostatečný. V dnešní době jsou často parkoviště v městech vybavena bezpečnostními kamerami, pro získání popisu okolního prostředí by bylo možno užít i obraz z nich. Mimo to jsou k dispozici digitalizované mapy či živé satelitní snímky. Pro účely domácího užití je možné vytvořit mapy manuálně, zcela bez použití technologie zpracování obrazu. V takovém případě by systém ani nemusel být vybaven kamerami, předpokladem pro použitelnost takové mapy je, že se v dříve zmapovaném prostředí neobjevily nové překážky. Pro mapu použiji síť buněk, každá buňka bude obsahovat informaci, zda je průjezdná, či nikoli.

Pro popis trajektorie jsem zvolil Bézierovy křivky. Jedná se o křivky hojně užívané v počítačové grafice. Umožňují jednoduše parametricky popsat celou trajektorii, splňují požadavky na spojitost i spojitost derivací, snadno se upravují a tvarují. Jejich vlastnosti rozvedu detailněji v dalších kapitolách při samotné aplikaci v parkovacím asistentu.

Otázka způsobu vyhledávání trajektorie souvisí s předchozím bodem. Způsob prohledávání prostoru musí po nalezení konečného řešení poskytnout výsledek ve formě Bézierovy křivky. Nemohu tedy používat klasické algoritmy pro vyhledávání trajektorie z jednoho bodu do druhého, protože jako výstup poskytují sadu buněk. Z těchto buněk by sice bylo možné rekonstruovat křivku například proložením spojovacích elementů středy zmíněných buněk, avšak výsledek by nebyl křivkou v požadované formě. Další možností by bylo identifikovat parametry Bézierovy křivky tak, aby co nejlépe aproximovala nalezenou trajektorii. Problémem by v takovém případě byla volba vhodného stupně křivky tak, aby

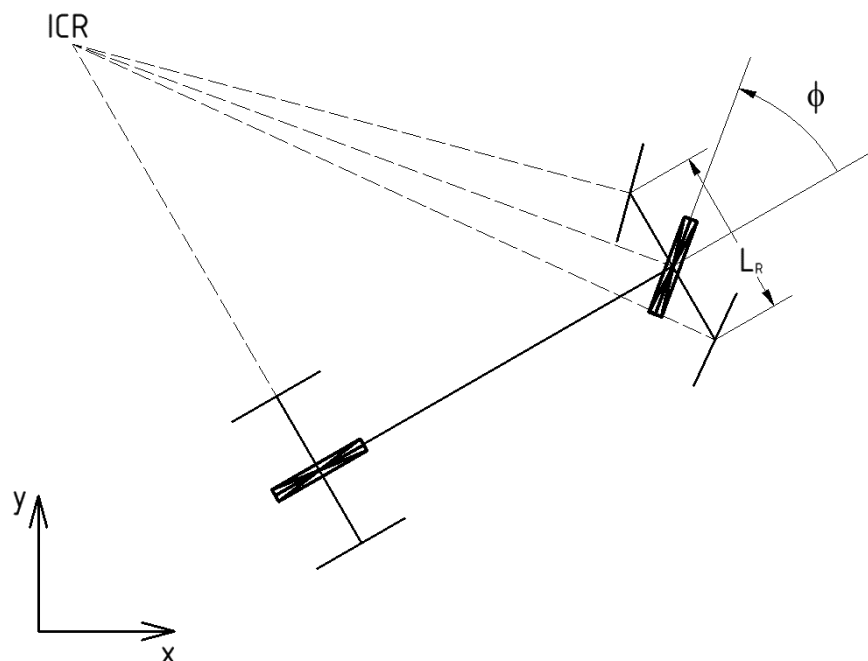
dostatečně věrně sledovala průběh vypočítané trajektorie. Pokud by nebyl zvolen dostatečný stupeň křivky, mohlo by například vlivem chyby aproximace dojít místně k průniku identifikované trajektorie do prostoru překážky. Tomu by bylo možné zabránit nastavením dalších omezujících podmínek, celý postup by se tím však již značně komplikoval. Za účelem nalezení trajektorie jsem tedy navrhl vlastní algoritmus, který prohledává zmapovaný prostor a snaží se proložit počáteční a koncové místo Bézierovou křivkou, která je následně optimalizována tak, aby umožnila co nejlepší průjezd pro soustavu automobilu s taženým zařízením.

## 7. Sestavení kinematických modelů

Při sestavování kinematických modelů jednotlivých těles (automobil, přívěs, vlečka) budu vycházet především z poznatků načerpaných z rešeršní studie. Model budu sestavovat v prostředí Matlab Simulink, který umožňuje pohodlnou integraci jednotlivých veličin v čase a snadné vyčítání a zaznamenávání průběhů veličin, které jsou předmětem mého zájmu.

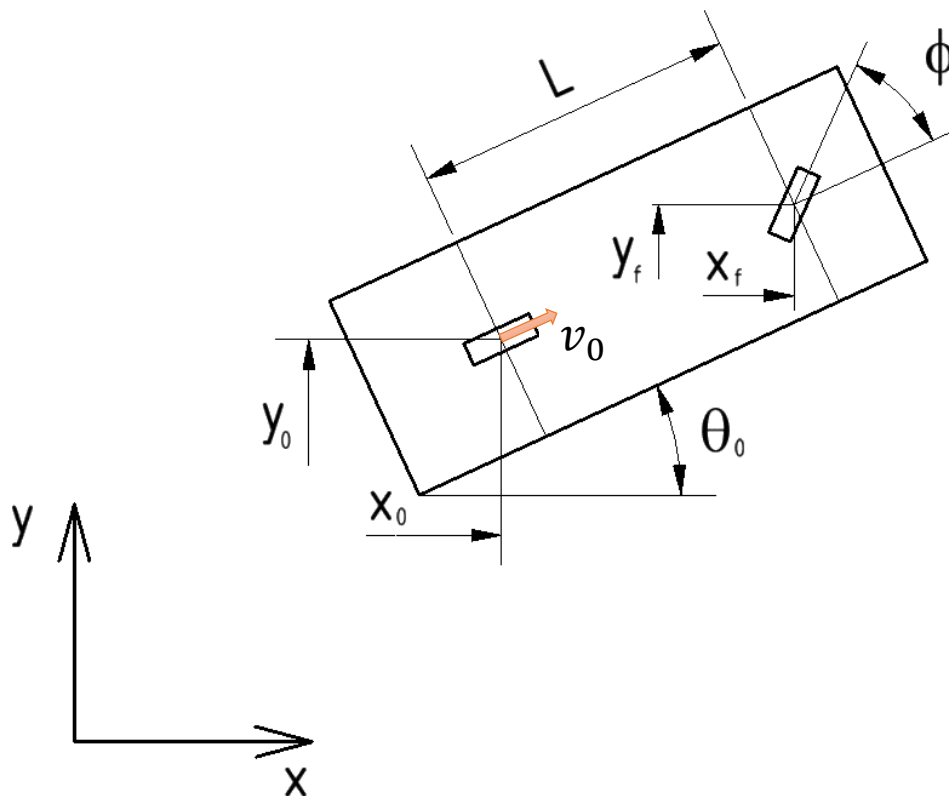
### 7.1. Kinematický model automobilu

Pro kinematický model vozidla budu uvažovat zjednodušený jednostopý model s pevnou zadní nápravou dle Obr. 6. Řiditelné kolo automobilu je umístěno uprostřed přední nápravy. Natočení kola přední nápravy  $\phi$  v rovině je dáno polohou volantu a příslušným převodem mezi natočením volantu a kola. Jelikož uvažujeme zjednodušený model Ackermannova řízení, tak je úhel natočení myšleného kola přední nápravy průměrem natočení dvou reálných kol, která jsou od sebe ve skutečnosti vzdálena o rozchod kol přední nápravy. Za rozchod kol nápravy  $L_R$  považujeme přibližně vzdálenost středů otisků pneumatik reálného kola. Modelované kolo má zanedbatelnou šířku a jeho natočení je takové, aby se okamžitý střed otáčení automobilu (ICR) shodoval se středem otáčení podle Ackermanna [21].



Obr. 6 – Zjednodušení modelu automobilu na jednostopý model

Použitý model neuvažuje rychlost otáčení volantem, poloha volantu se mění bez časové prodlevy. Můžeme si to dovolit vzhledem k vlastnostem Bézierovy křivky – má spojitý průběh křivosti. Z toho plyne, že při průjezdu trajektorie se natočení kol bude měnit také spojitě.



Obr. 7 – Odvození pohybových rovnic automobilu

Pohybové rovnice automobilu v rovině lze dohledat v literatuře. Postup odvození těchto rovnic uvádí například zdroj [14], podle něhož budu postupovat. Odvození vychází z Obr. 7. Nejdříve si zapíšeme neholonomní vazbové rovnice pro přední a zadní kolo. Vycházíme z předpokladu, že středy obou náprav (dány souřadnicemi  $[x_0, y_0]$  a  $[x_f, y_f]$  a s rozvorem  $L$ ) se vždy pohybují ve směru příslušného kola, úhel natočení vozidla v globálním souřadnicovém systému je  $\theta_0$  a úhel natočení předního kola je  $\phi$ . Kladná orientace obou úhlů je proti směru hodinových ručiček, úhel  $\theta_0$  je vztažen ke kladnému směru x-ové osy, úhel  $\phi$  je vztažen k úhlu  $\theta_0$  (odpovídá natočení automobilu). Rychlost kola zadní nápravy  $v_0$  známe a můžeme ji vyjádřit následovně:

$$v_0 = \sqrt{\dot{x}_0^2 + \dot{y}_0^2}. \quad (7.1)$$

Z geometrického vztahu podle Obr. 7 lze psát:

$$\tan \theta_0 = \frac{\dot{y}_0}{\dot{x}_0} = \frac{\sin(\theta_0)}{\cos(\theta_0)}, \quad (7.2)$$

z čehož plyne (obdobně i pro přední kolo):

$$\dot{x}_0 \sin(\theta_0) - \dot{y}_0 \cos(\theta_0) = 0, \quad (7.3)$$

$$\dot{x}_f \sin(\theta_0 + \phi) - \dot{y}_f \cos(\theta_0 + \phi) = 0. \quad (7.4)$$

Pro vztah mezi středy náprav platí:

$$x_f = x_0 + L \cos(\theta_0), \quad (7.5)$$

$$y_f = y_0 + L \sin(\theta_0). \quad (7.6)$$

Ze vztahů (7.5) a (7.6) vyjádříme časové derivace a dosadíme do rovnice (7.4), dostaneme

$$\dot{x}_f = \frac{dx_f}{dt} = \dot{x}_0 - L\dot{\theta}_0 \sin(\theta_0), \quad (7.7)$$

$$\dot{y}_f = \frac{dy_f}{dt} = \dot{y}_0 + L\dot{\theta}_0 \cos(\theta_0), \quad (7.8)$$

$$(\dot{x}_0 - L\dot{\theta}_0 \sin(\theta_0)) \sin(\theta_0 + \phi) - (\dot{y}_0 + L\dot{\theta}_0 \cos(\theta_0)) \cos(\theta_0 + \phi) = 0. \quad (7.9)$$

Roznásobením vztahů v rovnici (7.9) a použitím součtových vzorců pro goniometrické funkce vyjádříme vztah pro  $\dot{\theta}_0$ :

$$\begin{aligned} & \dot{x}_0 \sin(\theta_0 + \phi) - \dot{y}_0 \cos(\theta_0 + \phi) \\ & - L\dot{\theta}_0 \cos(\theta_0) (\cos(\theta_0) \cos(\phi) - \sin(\theta_0) \sin(\phi)) \\ & - L\dot{\theta}_0 \sin(\theta_0) (\sin(\theta_0) \cos(\phi) + \cos(\theta_0) \sin(\phi)) = 0, \end{aligned} \quad (7.10)$$

$$\begin{aligned} & \dot{x}_0 \sin(\theta_0 + \phi) - \dot{y}_0 \cos(\theta_0 + \phi) \\ & - L\dot{\theta}_0 (\cos^2(\theta_0) \cos(\phi) - \cos(\theta_0) \sin(\theta_0) \sin(\phi) \\ & + \sin^2(\theta_0) \cos(\phi) + \sin(\theta_0) \cos(\theta_0) \sin(\phi)) = 0, \end{aligned} \quad (7.11)$$

$$\dot{x}_0 \sin(\theta_0 + \phi) - \dot{y}_0 \cos(\theta_0 + \phi) - L\dot{\theta}_0 \cos(\phi) = 0. \quad (7.12)$$

Z poslední rovnice (7.12) už můžeme jednoduše vyjádřit  $\dot{\theta}_0$ :

$$\dot{\theta}_0 = \frac{\dot{x}_0 \sin(\theta_0 + \phi) - \dot{y}_0 \cos(\theta_0 + \phi)}{L \cos(\phi)}. \quad (7.13)$$

Neholonomní vazbová rovnice (7.3) je splněna pro  $\dot{x}_0 = v_0 \cos(\theta_0)$  a  $\dot{y}_0 = v_0 \sin(\theta_0)$ . Dosadíme do rovnice (7.13) a upravíme:

$$\dot{\theta}_0 = \frac{v_0 \cos(\theta_0) \sin(\theta_0 + \phi) - v_0 \sin(\theta_0) \cos(\theta_0 + \phi)}{L \cos(\phi)}, \quad (7.14)$$

$$\begin{aligned} \dot{\theta}_0 &= \frac{v_0 \cos(\theta_0) (\sin(\theta_0) \cos(\phi) + \cos(\theta_0) \sin(\phi))}{L \cos(\phi)} \\ &\quad - \frac{v_0 \sin(\theta_0) (\cos(\theta_0) \cos(\phi) - \sin(\theta_0) \sin(\phi))}{L \cos(\phi)}, \end{aligned} \quad (7.15)$$

$$\begin{aligned} \dot{\theta}_0 &= \frac{v_0 (\cos(\theta_0) \sin(\theta_0) \cos(\phi) + \cos^2(\theta_0) \sin(\phi))}{L \cos(\phi)} \\ &\quad - \frac{v_0 (\sin(\theta_0) \cos(\theta_0) \cos(\phi) - \sin^2(\theta_0) \sin(\phi))}{L \cos(\phi)}, \end{aligned} \quad (7.16)$$

$$\dot{\theta}_0 = \frac{v_0 \sin(\phi)}{L \cos(\phi)} = \frac{v_0}{L} \tan(\phi). \quad (7.17)$$

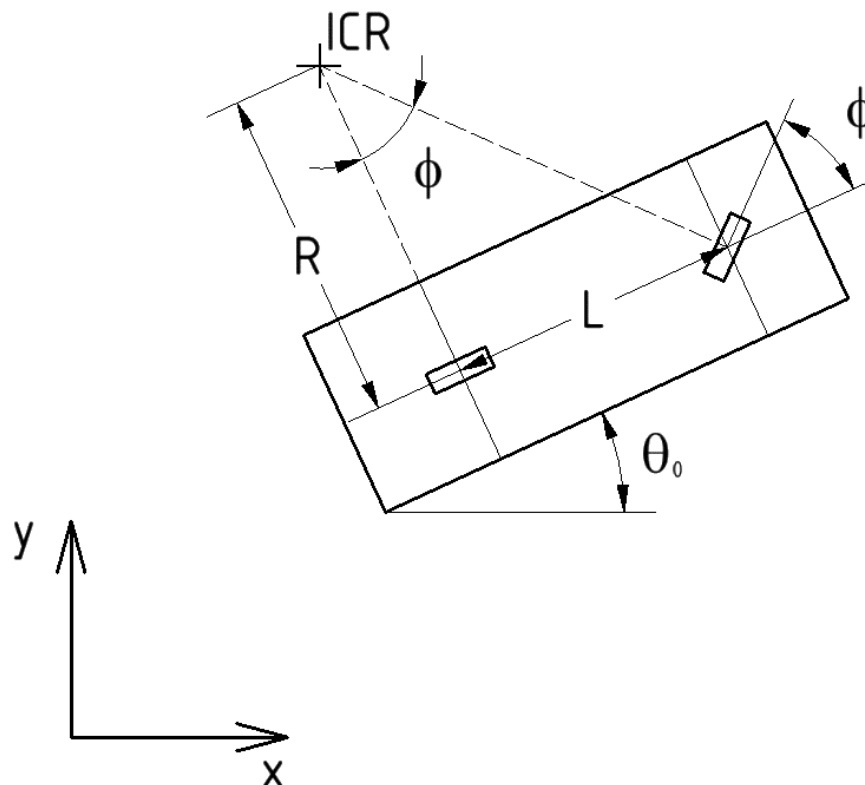
Získali jsme tři pohybové rovnice pro popis pohybu automobilu v rovině:

$$\dot{x}_0 = v_0 \cos(\theta_0), \quad (7.18)$$

$$\dot{y}_0 = v_0 \sin(\theta_0), \quad (7.19)$$

$$\dot{\theta}_0 = \omega_0 = \frac{v_0}{L} \tan(\phi). \quad (7.20)$$

Získané pohybové rovnice odpovídají geometrické představě podle Obr. 8.



Obr. 8 – Poloha středu otáčení automobilu v závislosti na natočení kola přední nápravy

Z Obr. 8 vidíme, že pro úhel  $\phi$  platí:

$$\tan(\phi) = \frac{L}{R}, \quad (7.21)$$

kde  $R$  značí poloměr otáčení vozidla kolem okamžitého středu otáčení ICR (*Instant Centre of Rotation*). Dosazením rovnice (7.21) do rovnice (7.20) získáme vztah pro úhlovou rychlost v závislosti na rychlosti a poloměru otáčení:

$$\dot{\theta}_0 = \omega_0 = \frac{v_0 L}{L R} = \frac{v_0}{R}. \quad (7.22)$$

V následující části uvedu odvození pro kinematické veličiny pasivně taženého přívěsu.

## 7.2. Kinematický model přívěsu

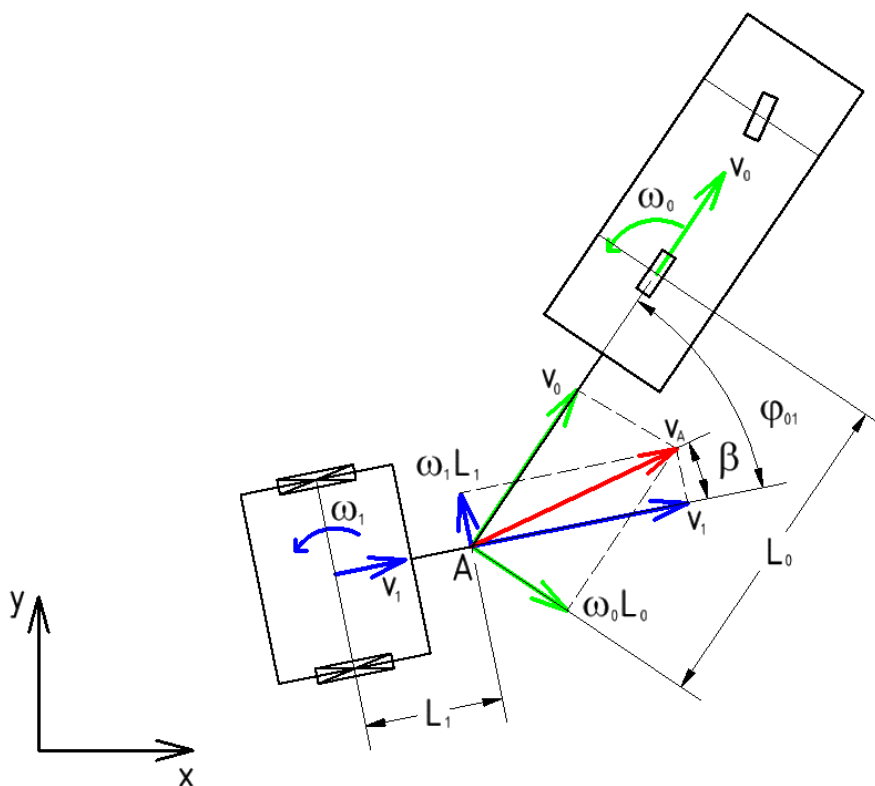
Problematice odvození vztahu kinematických veličin mezi automobilem a přívěsem se věnuje závěrečná práce [22], téma dále rozvíjí [23], obě práce vznikly stejně jako tato na Odboru mechaniky a mechatroniky. Automobil je s přívěsem spojen prostřednictvím tažného zařízení, při rovinné úloze lze toto spojení nahradit rotační vazbou. Automobil i přívěs konají obecný rovinný pohyb složený z translace bodů (odpovídajících středům náprav) a rotace kolem těchto bodů.

Rychlost libovolného bodu tělesa konajícího obecný rovinný pohyb lze vyjádřit jako superpozici rychlosti translace a rotace:

$$\vec{v}_A = \vec{v}_S + \vec{\rho} \times \vec{\omega}_S, \quad (7.23)$$

kde  $\vec{v}_A$  je vektor rychlosti libovolného bodu tělesa,  $\vec{v}_S$  je vektor rychlosti posuvu bodu  $S$ ,  $\vec{\omega}_S$  je vektor rychlosti otáčení tělesa kolem bodu  $S$  a  $\vec{\rho}$  je vektor z bodu  $A$  do bodu  $S$ .

Rychlost automobilu i vlečky musí být shodná pro obě tělesa v bodě jejich spojení, tedy v místě rotační vazby. Toto graficky interpretuje Obr. 9.



Obr. 9 – Grafické znázornění rychlostí automobilu a přívěsu (vytvoreno podle [22])



Podle Obr. 9 lze odvodit vztah mezi rychlostí a úhlovou rychlostí automobilu a přívěsu, musí platit:

$$v_A \cos(\beta) = v_1 \quad (7.24)$$

a zároveň

$$v_A \cos(\varphi_{01} - \beta) = v_A (\cos(\varphi_{01}) \cos(\beta) + \sin(\varphi_{01}) \sin(\beta)) = v_0. \quad (7.25)$$

Z rovnice (7.24) vyjádříme  $v_A$  a dosadíme do (7.25), získáme vztah mezi  $v_0$  a  $v_1$ :

$$v_0 = v_1 \frac{\cos(\varphi_{01}) \cos(\beta) + \sin(\varphi_{01}) \sin(\beta)}{\cos(\beta)}, \quad (7.26)$$

$$v_0 = v_1 (\cos(\varphi_{01}) + \sin(\varphi_{01}) \tan(\beta)). \quad (7.27)$$

Úhel  $\beta$  lze z obrázku vyjádřit jako

$$\tan(\beta) = \frac{\omega_1 L_1}{v_1}. \quad (7.28)$$

Vztah (7.28) dosadíme do (7.27) a získáme konečný tvar pro rychlost automobilu  $v_0$ :

$$v_0 = v_1 \cos(\varphi_{01}) + \omega_1 L_1 \sin(\varphi_{01}). \quad (7.29)$$

Obdobným způsobem vyjádříme úhlové rychlosti:

$$v_A \sin(\beta) = \omega_1 L_1, \quad (7.30)$$

$$v_A \sin(\varphi_{01} - \beta) = v_A (\sin(\varphi_{01}) \cos(\beta) - \cos(\varphi_{01}) \sin(\beta)) = \omega_0 L_0. \quad (7.31)$$

Po vyjádření  $v_A$  z (7.30) a dosazení do (7.31) získáme:

$$\omega_0 L_0 = \omega_1 L_1 \frac{\sin(\varphi_{01})}{\tan(\beta)} - \omega_1 L_1 \cos(\varphi_{01}). \quad (7.32)$$

Za  $\tan(\beta)$  dosadíme opět ze vztahu (7.28) a dostaneme tvar pro úhlovou rychlost automobilu  $\omega_0$ :

$$\omega_0 = \frac{v_1}{L_0} \sin(\varphi_{01}) - \frac{\omega_1 L_1}{L_0} \cos(\varphi_{01}). \quad (7.33)$$

Získané vztahy lze vyjádřit v maticovém tvaru, který použijí při vytváření modelu v Simulinku:

$$\begin{bmatrix} v_0 \\ \omega_0 \end{bmatrix} = \begin{bmatrix} \cos(\varphi_{01}) & L_1 \sin(\varphi_{01}) \\ \frac{\sin(\varphi_{01})}{L_0} & -\frac{L_1}{L_0} \cos(\varphi_{01}) \end{bmatrix} \begin{bmatrix} v_1 \\ \omega_1 \end{bmatrix}. \quad (7.34)$$

Pro účely opačného převodu tvar s inverzní maticí:

$$\begin{bmatrix} v_1 \\ \omega_1 \end{bmatrix} = \begin{bmatrix} \cos(\varphi_{01}) & L_0 \sin(\varphi_{01}) \\ \frac{\sin(\varphi_{01})}{L_1} & -\frac{L_0}{L_1} \cos(\varphi_{01}) \end{bmatrix} \begin{bmatrix} v_0 \\ \omega_0 \end{bmatrix}. \quad (7.35)$$

### 7.3. Kinematický model vlečky

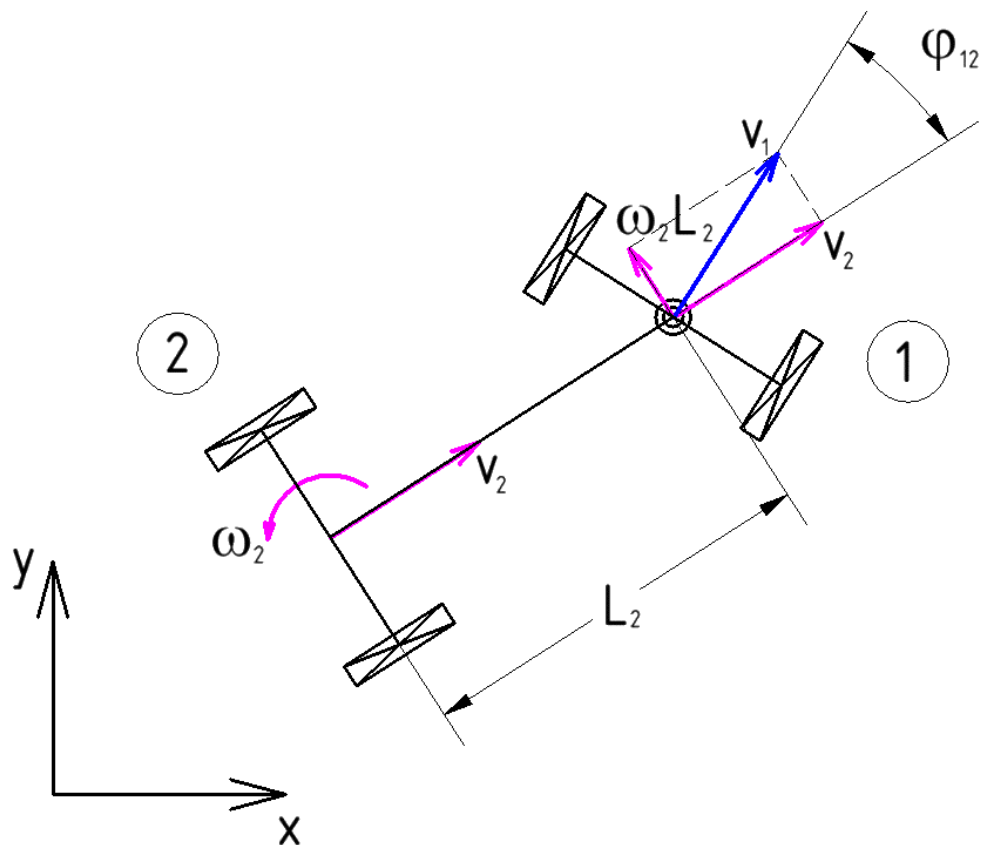
Pojmem vlečka budu dále v této práci odkazovat na přívěs se dvěma nápravami, z nichž zadní je pevná a přední je otočná kolem čepu umístěného uprostřed nápravy. Obě nápravy jsou pasivní a jsou vlečeny prostřednictvím tažného zařízení umístěného na oji, která je kolmá na přední nápravu. Příklad vlečky je znázorněn na Obr. 10.



Obr. 10 – Model vlečky Fliegl ([www.darky-berk.cz](http://www.darky-berk.cz))

Úkon couvání s vlečkou je co do požadavků na dovednosti řidiče obecně považován za mnohem náročnější než couvání s obyčejným přívěsem. Za vozidlem, které táhne vůz, jsou totiž připojena dvě pasivní tělesa (lze pomyslně rozdělit na přední a zadní nápravu vlečky), jejichž vzájemné úhly zalomení určují okamžitý směr pohybu. Řidič musí tedy stabilizovat hned dva úhly za účelem jízdy po zamýšlené křivce.

Z kinematického hlediska lze vlečku považovat za dva přívěsy zapojené za sebe. Přední nápravu vlečky lze modelovat stejným způsobem, jako pasivní přívěs v kapitole 7.2. Vztah mezi kinematickými veličinami vozidla a přední nápravy je tedy dán vztahem (7.34), případně inverzně (7.35). Zbývá tedy vyřešit poměr mezi kinematickými veličinami zadní a přední nápravy. Zadní náprava je připojena za přední prostřednictvím čepu, který lze modelovat jako rotační vazbu. V bodě rotační vazby, který je společný pro obě tělesa, musí opět platit rovnost rychlostí obou těles. Geometrickou interpretaci poskytuje Obr. 11, podle kterého provedu odvození poměrů rychlostí a úhlových rychlostí těles.



Obr. 11 – Grafické znázornění rychlostí přední (1) a zadní (2) nápravy vlečky

Při připojení zadní nápravy vlečky na střed její přední nápravy se vztahy pro vyjádření vzájemných kinematických veličin oproti případu, kdy je připojení mimo střed nápravy značně zjednoduší. Rychlost tělesa 1 (přední nápravy) je totiž v bodě rotační vazby přímo rovna posuvné rychlosti  $v_1$ , úhlová rychlost nemá na celkovou rychlost účinek.

Podle Obr. 11 lze tedy psát:

$$v_2 = v_1 \cos(\varphi_{12}), \quad (7.36)$$

$$\omega_2 L_2 = v_1 \sin(\varphi_{12}) \rightarrow \omega_2 = \frac{v_1}{L_2} \sin(\varphi_{12}). \quad (7.37)$$

Jak je patrné z rovnic (7.36) a (7.37), rychlost ani úhlová rychlost zadní nápravy není funkcí úhlové rychlosti přední nápravy, nýbrž pouze funkcí úhlu zalomení mezi těmito dvěma nápravami. Při dalším postupu při navrhování asistenta však budeme muset znát průběh úhlové rychlosti přední nápravy  $\omega_1$ . Abychom ho získali vyjádříme si nejdříve rychlost  $v_1$  a úhel  $\varphi_{12}$  jako funkci  $v_2$  a  $\omega_2$ . Z rovnice (7.37) vyjádříme rychlost  $v_1$  a dosadíme do (7.36):

$$v_1 = \frac{\omega_2 L_2}{\sin(\varphi_{12})}, \quad (7.38)$$

$$v_2 = \frac{\omega_2 L_2}{\sin(\varphi_{12})} \cos(\varphi_{12}) = \frac{\omega_2 L_2}{\tan(\varphi_{12})}. \quad (7.39)$$

Vyjádříme  $\varphi_{12}$  a dosadíme do (7.38):

$$\varphi_{12} = \operatorname{atan}\left(\frac{\omega_2 L_2}{v_2}\right), \quad (7.40)$$

$$v_1 = \frac{\omega_2 L_2}{\sin\left(\operatorname{atan}\left(\frac{\omega_2 L_2}{v_2}\right)\right)}. \quad (7.41)$$

V další části odvození budeme uvažovat systém v globálním souřadnicovém systému podle Obr. 12. Musíme zavést souřadnice popisující polohu všech těles soustavy v prostoru, k těmto souřadnicím přísluší odvozené kinematické veličiny v předchozí části.

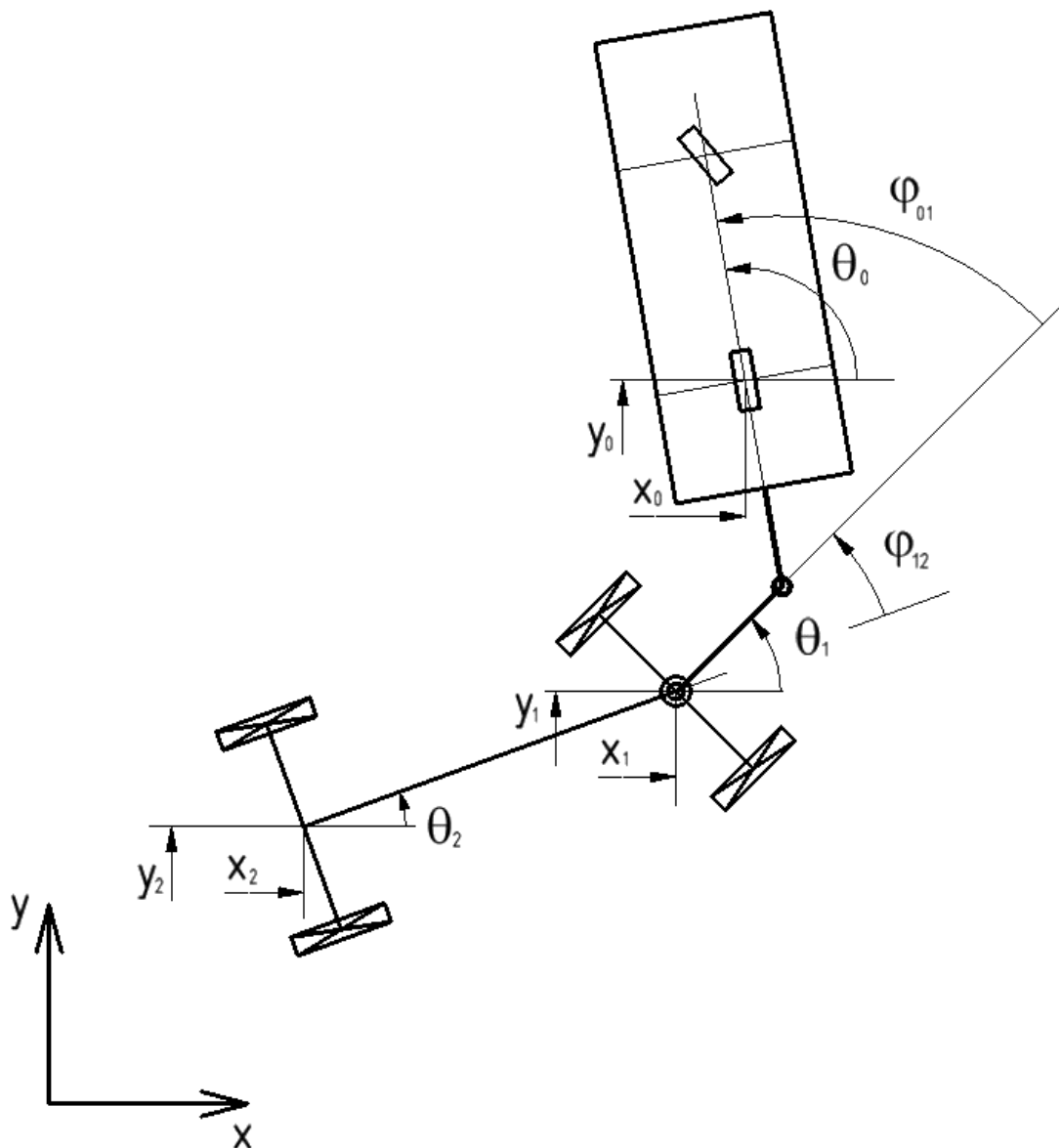
Budeme předpokládat, že známe časový průběh rychlostí  $\omega_2(t)$  a  $v_2(t)$ . V takovém případě známe v každém časovém okamžiku podle rovnice (7.40) i úhel  $\varphi_{12}(t)$ . Tento úhel je podle Obr. 12 roven:

$$\varphi_{12}(t) = \theta_1(t) - \theta_2(t). \quad (7.42)$$

Jelikož známe průběh úhlové rychlosti  $\omega_2 = \dot{\theta}_2 = \frac{d\theta_2}{dt}$  v čase, známe při doplnění počáteční podmínky  $\theta_2(0) = \theta_{20}$  i časový průběh úhlu  $\theta_2(t)$ . Z něj můžeme jednoduše vyjádřit

$$\theta_1(t) = \theta_2(t) + \varphi_{12}(t). \quad (7.43)$$

Známe tedy i časový průběh úhlu natočení přední nápravy  $\theta_1(t)$ .



Obr. 12 – Soustava automobilu s přívěsem v globálním souřadnicovém systému

Dále budu uvažovat diskrétní uzavřený časový interval  $T \in \langle T_0, T_N \rangle$ , pro který platí  $T_0 = 0$ ,  $T_k = T_0 + k \cdot \Delta T$  a  $k = 0, 1, 2, \dots, N$  a průběhy všech časových veličin diskretizuji v těchto časových okamžicích  $T_k$ .

Hledané diskrétní hodnoty úhlové rychlosti  $\omega_1$  získáme náhradou derivace diferencí podle vztahu:

$$\omega_1(T_k) = \frac{\theta_1(T_{k+1}) - \theta_1(T_k)}{\Delta T} \quad (7.44)$$

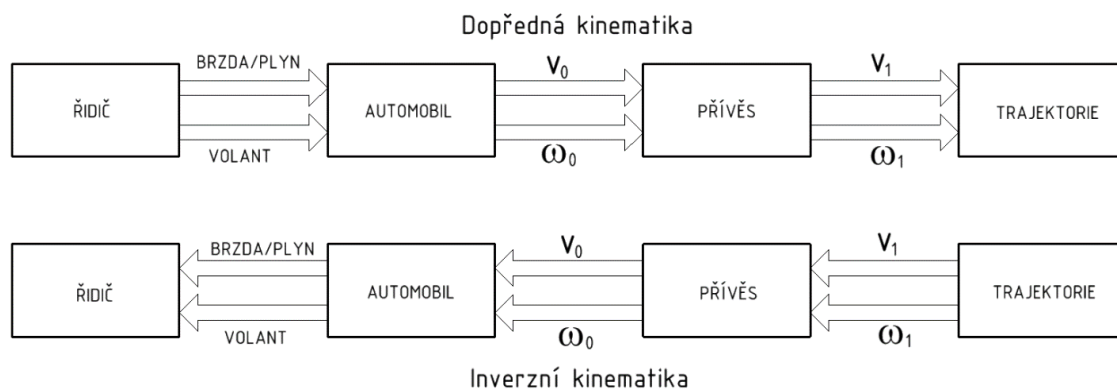
Pokud chceme vyjádřit úhlovou rychlost přední nápravy pomocí rychlostí zadní nápravy, můžeme za  $\theta_1(T_{k+1})$  dosadit z (7.43) a získáme:

$$\omega_1(T_k) = \frac{\theta_2(T_k) + \omega_2(T_k) \cdot \Delta T + \text{atan}\left(\frac{\omega_2(T_{k+1})L_2}{v_2(T_{k+1})}\right) - \theta_1(T_k)}{\Delta T} \quad (7.45)$$

Pro vyjádření  $\omega_1(t)$  ve spojitě časové oblasti můžeme využít interpolaci hodnot  $\omega_1(T_k)$ . Tyto hodnoty budeme potřebovat pro přepočítání kinematických veličin přední nápravy na automobil.

### 7.4. Souhrnný popis sestavených kinematických modelů

Sestavené rovnice umožňují propagaci kinematických veličin v řetězci buď z automobilu na přívěs nebo v případě rovnic pro vlečku z automobilu přes přední nápravu vlečky až na zadní nápravu vlečky. Propagaci směrem od automobilu budu nazývat dopředná kinematika, propagaci pomocí inverzních vyjádření v opačném směru budu nazývat inverzní kinematika. V reálné soupravě vždy dochází k propagaci kinematických veličin směrem od automobilu „dále“, jelikož vstupy do systému jsou právě rychlost automobilu a v závislosti na poloze volantu úhlová rychlost automobilu, které ovlivňuje řidič. Schematicky



Obr. 13 – Schéma propagace veličin v dopředné a inverzní kinematice

znázorněno na Obr. 13. Pro potřeby asistenta pro jízdu s přívěsem je vhodné využívat inverzní kinematiky, jelikož trajektorie přívěsu je předem dána. V souladu s předpokladem ve výše uvedeném odvození tedy známe z trajektorie průběh úhlové rychlosti taženého zařízení – pro vlečku  $\omega_2$ , v případě přívěsu  $\omega_1$  (vždy požadujeme, aby poslední náprava soupravy sledovala požadovanou trajektorii). Rychlost poslední nápravy soupravy nastavíme prozatím na konstantu.

## 7.5. Implementace kinematických modelů v prostředí Matlab Simulink

Rovnice odvozené v předchozí části této kapitoly použijí pro sestavení simulačního modelu v Simulinku. Rovnice pro propagaci kinematických veličin jsou sestaveny jak pro model s přívěsem, tak pro model s vlečkou. V simulačním modelu tyto vztahy rozšířím ještě o rovnice, které převádí pohyb soustavy do globálního souřadnicového systému. Lze je zobecněně zapsat pro  $i$ -té těleso v řetězci jako:

$$\dot{x}_i = v_i \cos(\theta_i), \quad (7.46)$$

$$\dot{y}_i = v_i \sin(\theta_i), \quad (7.47)$$

$$\dot{\theta}_i = \omega_i. \quad (7.48)$$

Pro každou diferenciální rovnici je třeba doplnit počáteční podmínky  $(x_{i0}, y_{i0}, \theta_{i0})$ , počáteční podmínky odpovídají stavu soupravy v globálním souřadnicovém systému na začátku trajektorie (odpovídá spuštění vyhledání trajektorie v další části práce), jedná se tedy o  $x$ -ovou a  $y$ -ovou souřadnici tělesa spolu s jeho úhlem natočení. Po integraci tedy dostáváme:

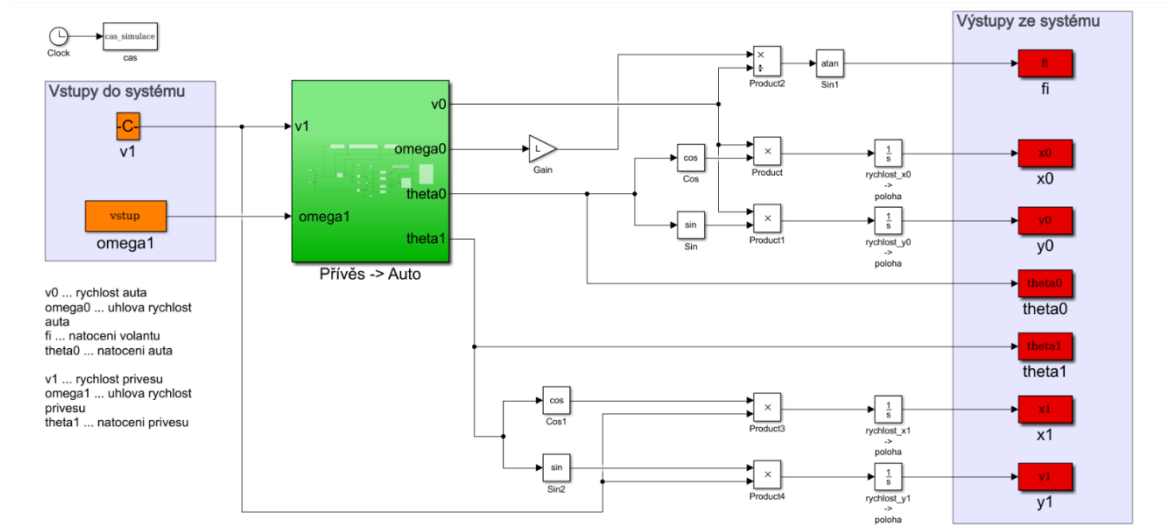
$$x_i = \int \dot{x}_i dt + x_{i0}, \quad (7.49)$$

$$y_i = \int \dot{y}_i dt + y_{i0}, \quad (7.50)$$

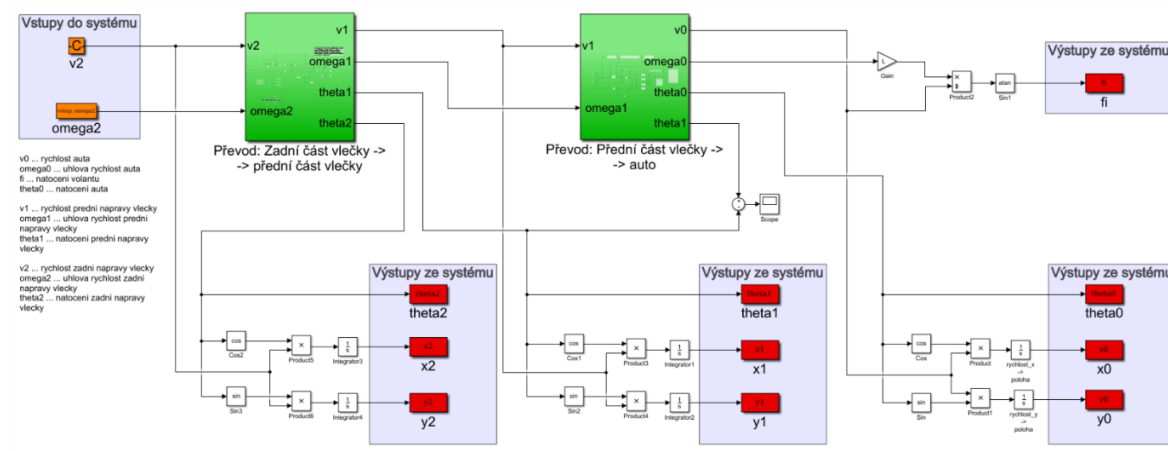
$$\theta_i = \int \omega_i dt + \theta_{i0}. \quad (7.51)$$

## Asistent pro nalezení optimální trajektorie při couvání s přívěsem

Modely inverzní kinematiky pro vlečku a přívěs jsou znázorněny na Obr. 14 a Obr. 15, zobrazena je pouze nejvyšší úroveň modelu, přepočet kinematických veličin probíhá v podsystémech (v schématech označeny zelenou barvou). Kompletní modely jsou součástí příloh (CD příloha) této práce.



Obr. 14 – Model soupravy automobilu s přívěsem v Simulinku



Obr. 15 – Model soupravy automobilu s vlečkou v Simulinku

Oranžovou barvou jsou ve schématech znázorněny vstupy do systému, rychlost zadní nápravy přívěsu je konstantní (zatím pro zjednodušení), úhlová rychlost je dopočítávána z trajektorie určené pro jízdu soupravy (bude vysvětleno v dalších kapitolách). Červenou barvou jsou zvýrazněny výstupy ze systému, jedná se o souřadnice těles v globálním souřadnicovém systému, přesněji o jejich časový průběh podle simulace. Dalším výstupem je průběh natočení kol, který lze přepočítat na průběh polohování volantu potřebný pro průjezd vypočítané trajektorie.



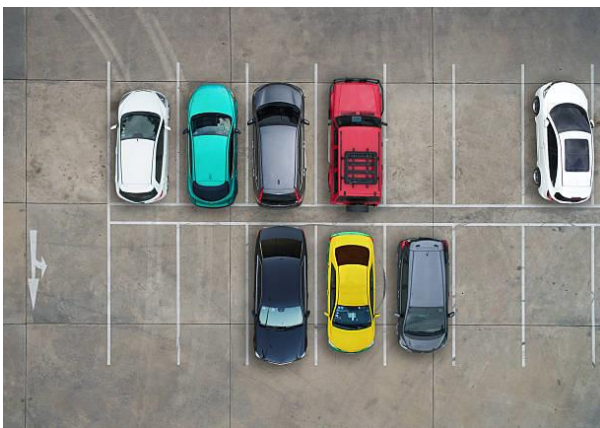
## 8. Vytvoření mapy prostoru pro parkovací manévr

Jak jsem uváděl již v předchozích kapitolách, úkol vytvořit mapu okolí, ve které bude prováděn parkovací manévr je řešitelný více způsoby. Jedním z možných přístupů je vytvoření digitální mapy prostředí pomocí technologie zpracování obrazu. Této problematice se ve své diplomové práci věnuje kolega Bc. Patrik Vacek [24]. Cílem zpracování obrazu je z obrazu kamer sestavit 2D scénu prostředí, ve které by byly identifikované průjezdné a neprůjezdné zóny a na základě této informace již mohou aplikovat svůj algoritmus pro vyhledávání optimální trajektorie, který budu popisovat dále v práci.

Při návrhu svého jízdního asistenta jsem se problematikou zpracování obrazu tedy nezabýval a předpokládal jsem znalost prostoru, ve kterém budu vyhledávat trajektorii a poté ve spojení s dříve odvozeným kinematickým modelem provádět manévr couvání. Scény pro provádění parkovacího manévru jsem tedy sestavoval „ručně“. Tento způsob manuálního vytvoření map prostředí je pro funkce asistenta zcela postačující, systém ke své funkci nepotřebuje kamery. Jejich nepřítomnost snižuje možnosti systému co do bezpečnosti (kontrola možné kolize), tak i do komfortu (výhled do prostoru pouze pomocí zpětných zrcátek). Na druhou stranu však značně snižuje výpočetní náročnost celého asistenčního systému a při případné realizaci i náklady na výrobek. Manuální vytváření map prostředí je vhodné, pokud je manévr opakovaně prováděn ve stejném prostředí, může se jednat například o domácí užití (zajetí do garáže), nebo firemní (zavážení zboží do firmy). Nevýhodou může být, že pokud dojde ke změně dříve zmapovaného místa (uprostřed příjezdové cesty do firmy leží nevyzvednuté zboží), nemá asistenční systém o této nové překážce žádné informace a nemůže ji zohlednit při výpočtu trajektorie. V případě jakékoliv změny prostředí by tedy bylo nutné manuálně vložit novou překážku.

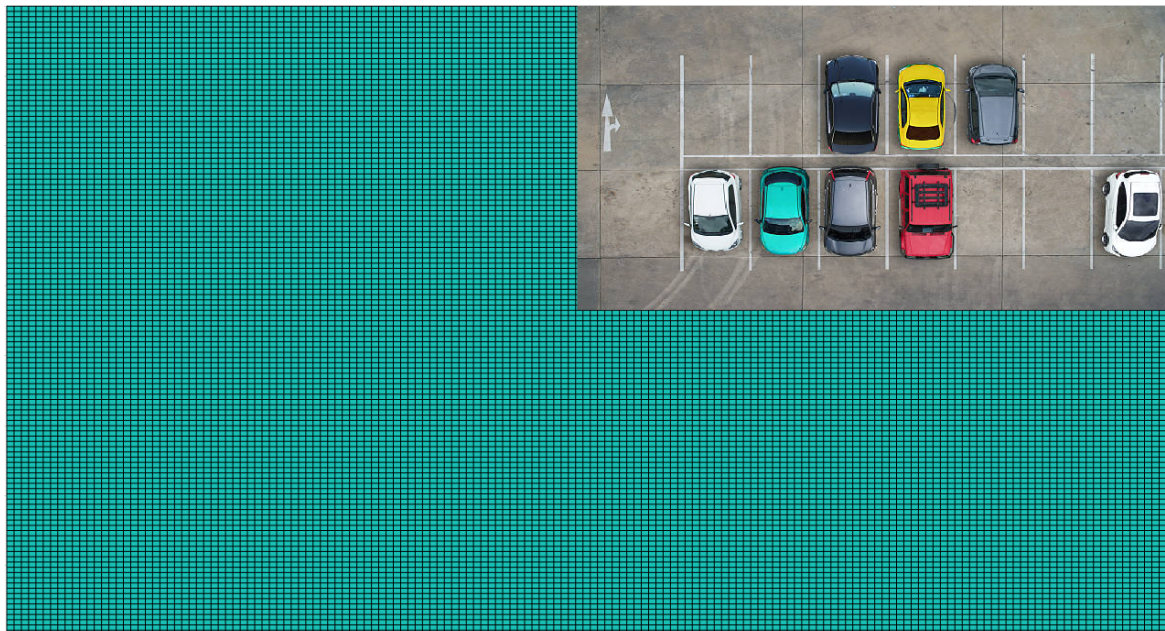
V dalších částech své práce budu využívat model systému automobil-přívěs, pro systém automobil-vlečka by se postup zásadně nelišil, bylo by pouze třeba uvažovat jinou prostorovou náročnost celé soupravy.

Pro demonstraci uvedu logiku postupu vytváření scény, která by byla v dalších částech práce použitelná pro vytvoření a testování algoritmu pro vyhledání optimální trajektorie.

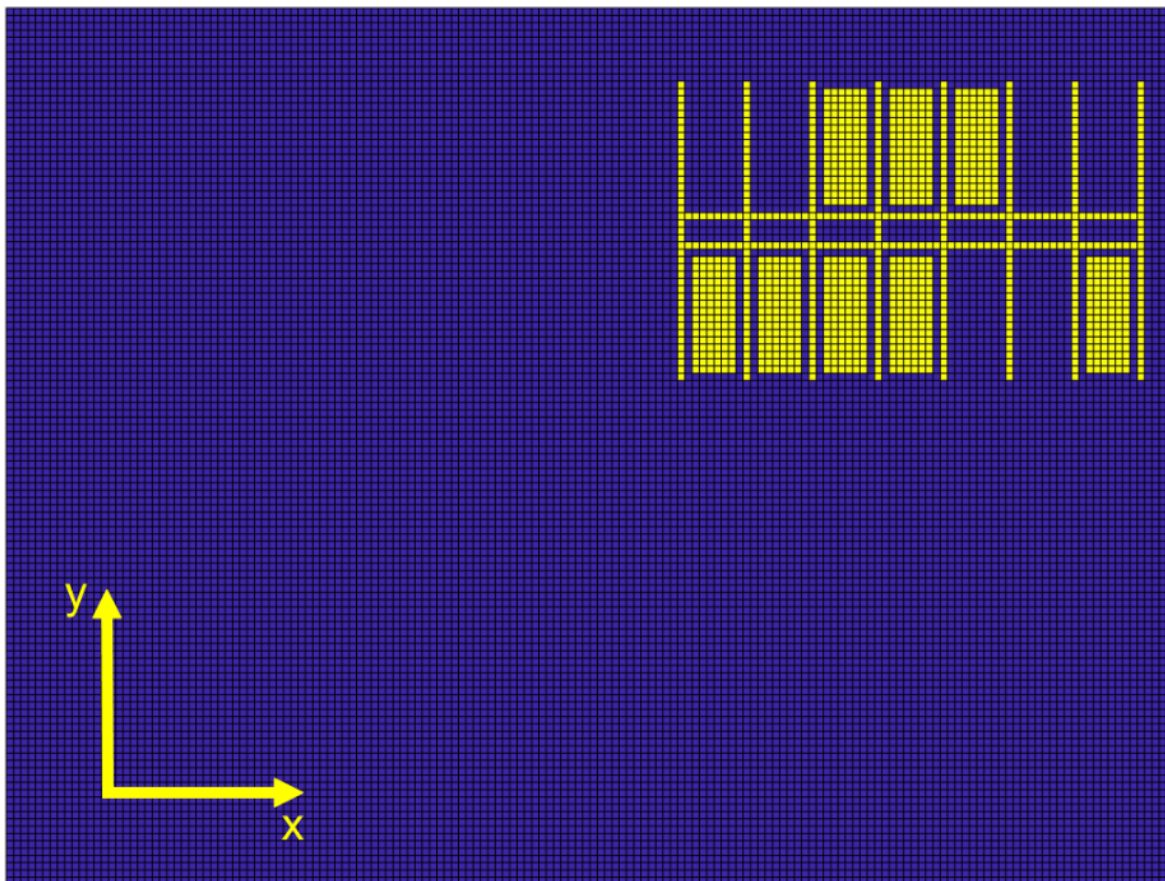


Obr. 16 – Reálné prostředí pro úkol parkování (istockphotos.com)

Na Obr. 16 je vrchní snímek parkoviště, na kterém jsou některá místa obsazena, některá volná. Při znalosti měřítka fotografie mohu vytvořit mapu prostředí, do které toto parkoviště zasadím (přidám navíc okolní prostředí). Tuto mapu znázorňuje Obr. 17 (bez měřítka a popisu os, zobrazení je pouze schematické).



Obr. 17 – Ustavení parkoviště do prostoru



Obr. 18 – Konečná podoba mapy prostředí

Prostředí z Obr. 17 dále upravím do konečné podoby, ve které bude v digitální podobě v binární logice popis celé scény. Konečnou podobu mapy prostředí znázorňuje Obr. 18 (opět pouze schematicky, bez popisu os, pouze naznačena orientace globálního souřadnicového systému). Při vytváření mám možnost dle potřeby zvolit vzdálenost ve směru obou os, kterou mapa pokrývá. Dále mohu nastavit libovolné síťování prostoru (počet buněk) ve směru obou os, je vhodné zvolit dostatečně hustou síť pro přesnější popis překážek v prostoru. Po vytvoření měřítka prostoru a síťování zapíši do příslušných buněk binární informaci, zda je průjezdná, či ne (0 odpovídá průjezdné buňce – není překážka; 1 odpovídá neprůjezdné buňce – je překážka). Na Obr. 18 jsou průjezdné buňky znázorněny modrou barvou, neprůjezdné buňky jsou znázorněny žlutě. Při vytváření map prostředí budu vždy kolem překážek přidávat „bezpečnostní zónu“. Tato zóna zvětší virtuální překážku oproti té reálné. Je to vhodné pro snadnější průjezdnost trajektorie, pokud by trajektorie příliš přiléhala k překážce, i malé nepřesnosti v řízení soupravy by mohly způsobit kolizi s překážkou, navíc i s přihlédnutím na řidičův pocit bezpečí není vhodné provádět příliš „těsné“ manévry.

V dalších částech práce budu používat různé mapy prostředí, které budou podobné té z Obr. 18, bude zachována logika mapy, tzn. průjezdné části budou modré, neprůjezdné žluté. Budu však používat sítě s menší hustotou buněk kvůli přehlednější možnosti zobrazení. Překážky zanesené na všech mapách budou již s „bezpečnostními zónami“, skutečné překážky v prostoru jsou tedy o vhodně zvolenou vzdálenost menší. I těsný průjezd kolem překážky v mapě tedy znamená průjezd s bezpečnou vzdáleností od skutečné překážky v reálném prostředí.

## 9. Způsob popisu trajektorie – Bézierovy křivky

Pro popis trajektorie jsem zvolil vzhledem k jejich vhodným vlastnostem Bézierovy křivky. V této kapitole se budu věnovat popisu těchto křivek, uvedu definiční vztahy, příklady užití a uvedu svůj postup při volbě vhodného řádu Bézierovy křivky pro mou úlohu. Budu vycházet z poznatků uvedených v [25] a [26].

### 9.1. Teorie Bézierových křivek, definice, vlastnosti, příklady užití

Matematický základ Bézierových křivek tvoří Bernsteinovy polynomy, které byly známy již od roku 1912, své využití v grafice našly však až asi o 50 let později. Zasloužil se o to francouzský inženýr a jeden ze zakladatelů fyzikálního modelování Pierre Étienne Bézier, který je využíval při navrhování karoserií automobilky Renault. První studii těchto křivek provedl však už roku 1959 francouzský fyzik a matematik Paul de Casteljau, který vyvinul pro automobilku Citroën de Casteljau algoritmus konstrukce bodu na Bézierově křivce.

Bézierova křivka je parametrická křivka daná svými řídicími body. Jeden segment křivky řádu  $n$  je určen  $n + 1$  řídicími body, uspořádanými do řídicího polygonu. Křivka samotná se nachází vždy uvnitř svého řídicího polygonu. Krajními řídicími body (prvním a posledním) křivka prochází, ostatní body pouze aproximuje. Každý řídicí bod má vliv na celou křivku, ne pouze na její určité segmenty, změna polohy jednoho bodu tudíž ovlivní tvar celé křivky. Matematický zápis Bézierovy křivky řádu (stupně)  $n$  je následující:

$$\mathbf{B}(t) = \sum_{i=0}^n b_{i,n}(t) \mathbf{P}_i = b_{0,n}(t) \mathbf{P}_0 + b_{1,n}(t) \mathbf{P}_1 + \dots + b_{n,n}(t) \mathbf{P}_n, \quad t \in [0,1], \quad (9.1)$$

kde  $\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_n$  jsou řídicí body,  $t$  je parametr křivky a kde:

$$b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i} = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}, \quad t \in [0,1], \quad i = 0, \dots, n, \quad (9.2)$$

jsou Bernsteinovy polynomy  $n$ -tého stupně.

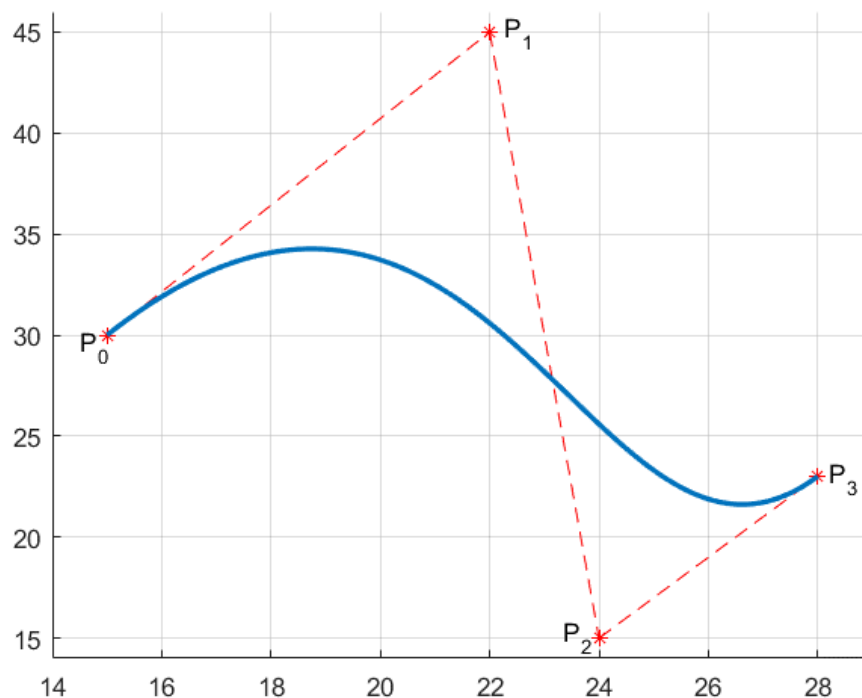
Některé vlastnosti Bézierovy křivky:

- Bézierova křivka prochází pouze svými krajními řídicími body, tedy  $\mathbf{B}(t = 0) = \mathbf{P}_0$  a  $\mathbf{B}(t = 1) = \mathbf{P}_n$ .
- Tečný vektor  $\mathbf{B}'(0)$  v počátku Bézierovy křivky je roven  $n$ -násobku vektoru určeného počátečním ramenem řídicího polygonu:  $\mathbf{B}'(0) = n(\mathbf{P}_1 - \mathbf{P}_0)$ .
- Tečný vektor  $\mathbf{B}'(1)$  v koncovém bodě Bézierovy křivky je roven  $n$ -násobku vektoru určeného koncovým ramenem řídicího polygonu:  $\mathbf{B}'(1) = n(\mathbf{P}_n - \mathbf{P}_{n-1})$ .
- Leží-li všechny řídicí body křivky na společné přímce, křivka se redukuje na úsečku.

Bézierovy křivky se v současnosti hojně využívají v počítačové grafice a souvisejících oborech. Díky jejich matematickému popisu nachází užití ve vektorové grafice, která nepodléhá rastrování, používá se řada křivek vzájemně na sebe napojených pro vytvoření obrazců se spojitými liniemi. Jejich užití je díky způsobu popisu velice jednoduché a intuitivní, většina grafických softwarů používajících Bézierovy křivky znázorňuje jejich řídicí body a umožňuje s nimi tahem myši manipulovat a tím tvarovat křivku. Další funkcionalitou je možnost nastavení řádu spojitosti křivek v místě jejich napojení, tím dochází k vzájemnému zavazbení poloh některých řídicích bodů. Příklad grafického softwaru pracujícího s Bézierovými křivkami je například Adobe Illustrator. V animačních aplikacích, například Adobe Flash, jsou Bézierovy křivky použity pro popis trajektorie animovaného tělesa, na základě těchto trajektorií pak program generuje snímky tělesa, které odpovídají pohybu po zadané křivce. Mimo popis trajektorie samotné se používají i pro popis rychlosti podél trajektorie, díky jejich vlastnostem je zaručena vždy plynulá změna rychlosti animovaného tělesa. Využívají se také jako základ znaků některých fontů implementovaných přímo v operačních systémech počítačů.

## 9.2. Bézierova kubika

Nejčastěji užívanou Bézierovou křivkou je křivka třetího řádu – Bézierova kubika. Je dána čtyřmi řídicími body  $P_0, P_1, P_2, P_3$ . Složitější obrazce se skládají napojováním více kubik za sebou, případně se vhodně transformují na jiné typy křivek. Pro svou práci jsem se prvotně rozhodl využít právě Bézierových kubik, příklad této křivky uvádí Obr. 19.



Obr. 19 – Příklad Bézierovy kubiky





stabilizovatelný jízdou vzad. Uvedenou orientaci přívěsu v prostoru jsem zvolil, protože pokud bude systém vybaven zpracováním obrazu, bude možné vytvořit mapu prostředí (a tím orientovat globální souřadnicový systém  $xy$ ) tak, aby byl rovnoběžný s přívěsem a orientován „po směru couvání“.

Jelikož tedy znám orientaci přívěsu na počátku, vím také, že řídicí bod  $\mathbf{P}_1$  Bézierovy kubiky musí mít stejnou  $y$ -ovou souřadnici, jako bod  $\mathbf{P}_0$ , pro který na počátku manévru platí:

$$\mathbf{P}_0 = \begin{bmatrix} x_{1s} \\ y_{1s} \end{bmatrix}, \quad (9.4)$$

kde souřadnice  $x_{1s}$  a  $y_{1s}$  jsou počáteční  $x$ -ová, respektive  $y$ -ová souřadnice polohy přívěsu. Pro bod  $\mathbf{P}_1$  tedy musí platit:

$$\mathbf{P}_1 = \begin{bmatrix} p_1 \\ y_{1s} \end{bmatrix}, \quad (9.5)$$

kde  $p_1$  je volný parametr Bézierovy kubiky.

Uživatel jízdního asistenta bude jako vstup do systému zadávat požadavek na koncovou polohu přívěsu v prostoru (na jaké místo si přeje dojet a jakým způsobem si přeje být natočen). Tímto požadavkem jsou tedy vázány stejně jako v případě počáteční orientace i body  $\mathbf{P}_2$  a  $\mathbf{P}_3$ . Pro jednodušší vyjádření budu zatím uvažovat, že koncová orientace přívěsu má být  $\pi/2$ . V takovém případě musí být  $x$ -ová souřadnice bodu  $\mathbf{P}_2$  stejná, jako bodu  $\mathbf{P}_3$ , pro tyto body tedy platí:

$$\mathbf{P}_3 = \begin{bmatrix} x_{1f} \\ y_{1f} \end{bmatrix}, \quad (9.6)$$

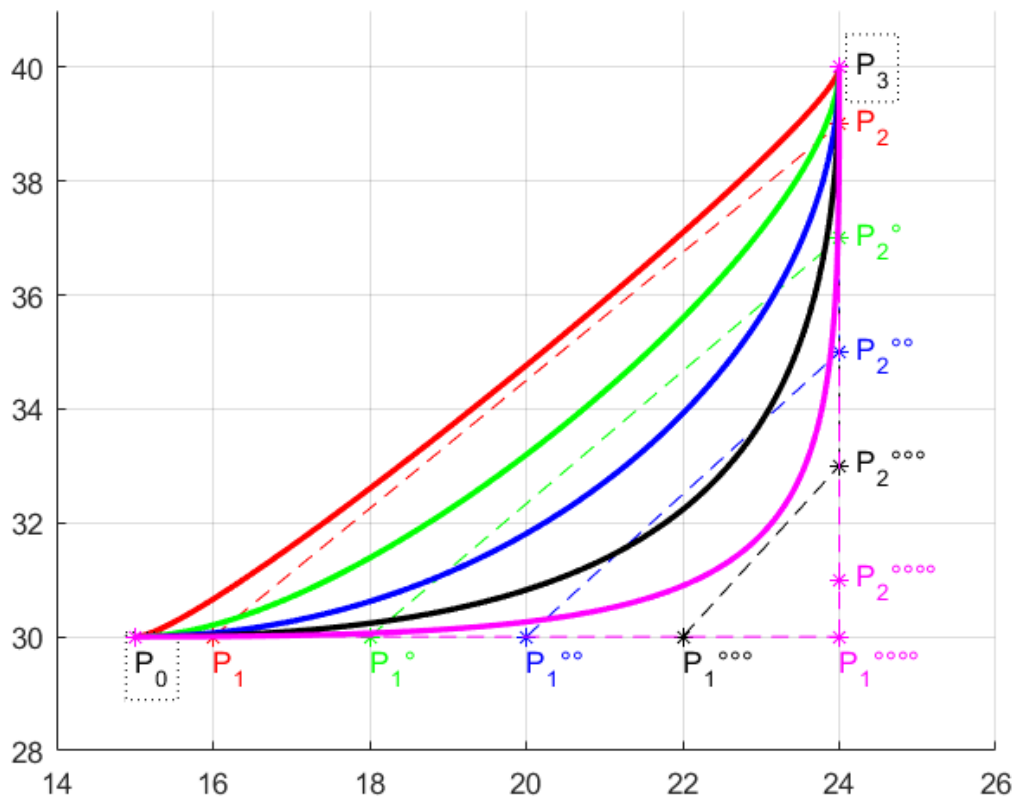
kde souřadnice  $x_{1f}$  a  $y_{1f}$  jsou koncová (final)  $x$ -ová, respektive  $y$ -ová souřadnice polohy přívěsu. A bod  $\mathbf{P}_2$  tedy:

$$\mathbf{P}_2 = \begin{bmatrix} x_{1f} \\ p_2 \end{bmatrix}, \quad (9.7)$$

kde  $p_2$  je volný parametr Bézierovy kubiky.

Po rozboru situace vidíme, že místo obecně osmi volných parametrů Bézierovy kubiky (osm souřadnic čtyř řídicích bodů) máme k dispozici pro optimalizaci trajektorie pouze dva volné

parametry  $p_1$  a  $p_2$ . Možnost tvarování takové křivky je tím pádem značně omezena, pro ukázkou uvedu Obr. 21, plné čáry jsou možné tvary křivky.



Obr. 21 – Možnosti tvarování Bézierovy kubiky s dvěma volnými parametry

Jak je z Obr. 21 patrné, je možné pomocí dvou volných parametrů pouze omezeně měnit tvar křivky, pro účely vyhledávání a optimalizace trajektorie není tento rozsah úpravy křivky dostačující. V další fázi návrhu popisu trajektorie jsem se tedy rozhodl vytvářet trajektorii jako několik křivek třetího stupně napojených za sebou. Za tímto účelem jsem musel odvodit vztahy pro napojování Bézierových křivek tak, aby odpovídaly požadavkům na spojení. Tyto vztahy lze dohledat i v literatuře, např. ve zdroji [26]. Budu požadovat spojitost do druhého řádu derivace v místě napojení křivky. Spojitost nultého řádu odpovídá tomu, že obě křivky mají společný bod napojení, trajektorie nemůže být přerušena. Spojitost prvního řádu derivace zajistí to, že výsledná křivka nebude v místě napojení obou kubik zalomena, s ohledem na soupravu by zalomení trajektorie znamenalo, že se přívěs musí na nekonečně malé vzdálenosti natočit o úhel konečné velikosti. Spojitost druhého řádu derivace zajistí spojitost křivosti podél trajektorie, pokud by v průběhu křivosti v místě napojení docházelo ke skokové změně, musela by se skokově měnit i úhlová rychlost pohybu soupravy, z čehož by plynul nespelnitelný požadavek na nekonečně velké zrychlení. V následující části této kapitoly proto odvodím vztahy mezi řídicími body dvou



křivek, které musí platit, aby v místě jejich napojení byla splněna spojitost derivací až do druhého řádu.

V rovnici (9.8) uvádím pro snadnější orientaci v textu již dříve zmíněný tvar Bézierovy kubiky, v rovnici (9.9) ekvivalentní zápis téže křivky. Druhý tvar uvádím, protože jde o tvar, který je snadnější pro derivování a vhodnější pro vyjádření hodnoty v  $t = 0$ .

$$\mathbf{B}(t) = (1-t)^3 \mathbf{P}_0 + 3t(1-t)^2 \mathbf{P}_1 + 3t^2(1-t) \mathbf{P}_2 + t^3 \mathbf{P}_3, \quad t \in [0,1], \quad (9.8)$$

$$\begin{aligned} \mathbf{B}(t) = & (-\mathbf{P}_0 + 3\mathbf{P}_1 - 3\mathbf{P}_2 + \mathbf{P}_3)t^3 + (3\mathbf{P}_0 - 6\mathbf{P}_1 + 3\mathbf{P}_2)t^2 \\ & + (-3\mathbf{P}_0 + 3\mathbf{P}_1)t + \mathbf{P}_0, \quad t \in [0,1]. \end{aligned} \quad (9.9)$$

Je vhodné mít k dispozici oba uvedené tvary. Tvar v rovnici (9.8) využijeme pro popis první kubiky, jelikož nás zajímají vlastnosti v jejím koncovém bodě, za parametr  $t$  budeme tedy dosazovat hodnotu 1. Tvar rovnice (9.9) využijeme pro popis druhé z kubik, kde naopak zkoumáme vlastnosti v počátečním bodě křivky, za parametr  $t$  tedy budeme dosazovat hodnotu 0. První kubiku budu značit  $\mathbf{B}_I$ , druhou budu značit  $\mathbf{B}_{II}$ , řídicí body příslušící té či oné kubice budu indexovat podle stejné logiky.

Vyjádřím si vztah pro první a druhou derivaci obou zápisů Bézierovy kubiky (nebudu dále uvádět rozsah parametru  $t$ , pokud to nebude nutné):

$$\begin{aligned} \mathbf{B}'(t) = & -3(1-t)^2 \mathbf{P}_0 + 3(1-t)^2 \mathbf{P}_1 - 6t(1-t) \mathbf{P}_1 \\ & + 6t(1-t) \mathbf{P}_2 - 3t^2 \mathbf{P}_2 + 3t^2 \mathbf{P}_3, \end{aligned} \quad (9.10)$$

$$\begin{aligned} \mathbf{B}'(t) = & 3(-\mathbf{P}_0 + 3\mathbf{P}_1 - 3\mathbf{P}_2 + \mathbf{P}_3)t^2 + 2(3\mathbf{P}_0 - 6\mathbf{P}_1 + 3\mathbf{P}_2) \\ & + (-3\mathbf{P}_0 + 3\mathbf{P}_1), \end{aligned} \quad (9.11)$$

$$\begin{aligned} \mathbf{B}''(t) = & 6(1-t) \mathbf{P}_0 - 6(1-t) \mathbf{P}_1 - 6(1-t) \mathbf{P}_1 \\ & + 6t \mathbf{P}_1 + 6(1-t) \mathbf{P}_2 - 6t \mathbf{P}_2 - 6t \mathbf{P}_2 + 6t \mathbf{P}_3, \end{aligned} \quad (9.12)$$

$$\mathbf{B}''(t) = 6(-\mathbf{P}_0 + 3\mathbf{P}_1 - 3\mathbf{P}_2 + \mathbf{P}_3)t + 2(3\mathbf{P}_0 - 6\mathbf{P}_1 + 3\mathbf{P}_2). \quad (9.13)$$

Jako první vyšetřím vztah mezi řídicími body pro spojitost nulté derivace. Podle očekávání dostaneme jednoduchý vztah:

$$\mathbf{B}_I(t=1) = \mathbf{B}_{II}(t=0), \quad (9.14)$$

který po dosazení z (9.8) a (9.9) dává první podmínku pro počáteční řídicí bod druhé z kubik:

$$\mathbf{P}_{0II} = \mathbf{P}_{3I}. \quad (9.15)$$

Zcela logicky tudíž musí mít počáteční bod druhé křivky shodné souřadnice s koncovým bodem křivky první.

Dále vyšetřím vztah mezi řídicími body pro zajištění spojitosti první derivace:

$$\mathbf{B}'_I(t = 1) = \mathbf{B}'_{II}(t = 0). \quad (9.16)$$

Po dosazení z vztahů (9.10) a (9.11) dostáváme:

$$-3\mathbf{P}_{2I} + 3\mathbf{P}_{3I} = -3\mathbf{P}_{0II} + 3\mathbf{P}_{1II}. \quad (9.17)$$

Za bod  $\mathbf{P}_{0II}$  dosadíme ze vztahu (9.15) a dostáváme:

$$-3\mathbf{P}_{2I} + 3\mathbf{P}_{3I} = -3\mathbf{P}_{3I} + 3\mathbf{P}_{1II}, \quad (9.18)$$

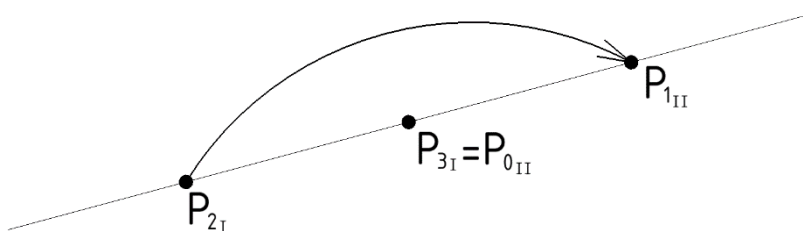
$$-3\mathbf{P}_{2I} + 6\mathbf{P}_{3I} = 3\mathbf{P}_{1II}. \quad (9.19)$$

Z čehož dostáváme podmínku pro druhý řídicí bod druhé z kubik ve tvaru:

$$\mathbf{P}_{1II} = 2\mathbf{P}_{3I} - \mathbf{P}_{2I} = \mathbf{P}_{3I} + (\mathbf{P}_{3I} - \mathbf{P}_{2I}) = \mathbf{P}_{3I} + \overrightarrow{\mathbf{P}_{2I}\mathbf{P}_{3I}}. \quad (9.20)$$

Geometrickou interpretací (Obr. 22) této rovnice je, že bod  $\mathbf{P}_{1II}$  je od bodu  $\mathbf{P}_{3I}$  vzdálen o stejnou vzdálenost, jako  $\mathbf{P}_{3I}$  od  $\mathbf{P}_{2I}$  a zároveň leží na jedné přímce (je pouze posunut o vektor  $\overrightarrow{\mathbf{P}_{2I}\mathbf{P}_{3I}}$  z bodu  $\mathbf{P}_{3I}$ ). Tento výsledek je v souladu s vlastnostmi tečen Bézierovy křivky, které byly uvedeny výše v kapitole 9.1. Při zpětném dosazení podle (9.15) získáme rovnost tečen obou křivek v daném bodě napojení:

$$\mathbf{P}_{1II} - \mathbf{P}_{0II} = \mathbf{P}_{3I} - \mathbf{P}_{2I}. \quad (9.21)$$



Obr. 22 – Geometrická interpretace spojitosti první derivace v bodě napojení

Zbývá ještě vyšetřit vztahy pro spojitost druhé derivace v bodě napojení obou kubik. Odvození probíhá zcela analogicky těm minulým:

$$\mathbf{B}'_I(t = 1) = \mathbf{B}''_{II}(t = 0). \quad (9.22)$$

Po dosazení ze vztahů (9.12) a (9.13):

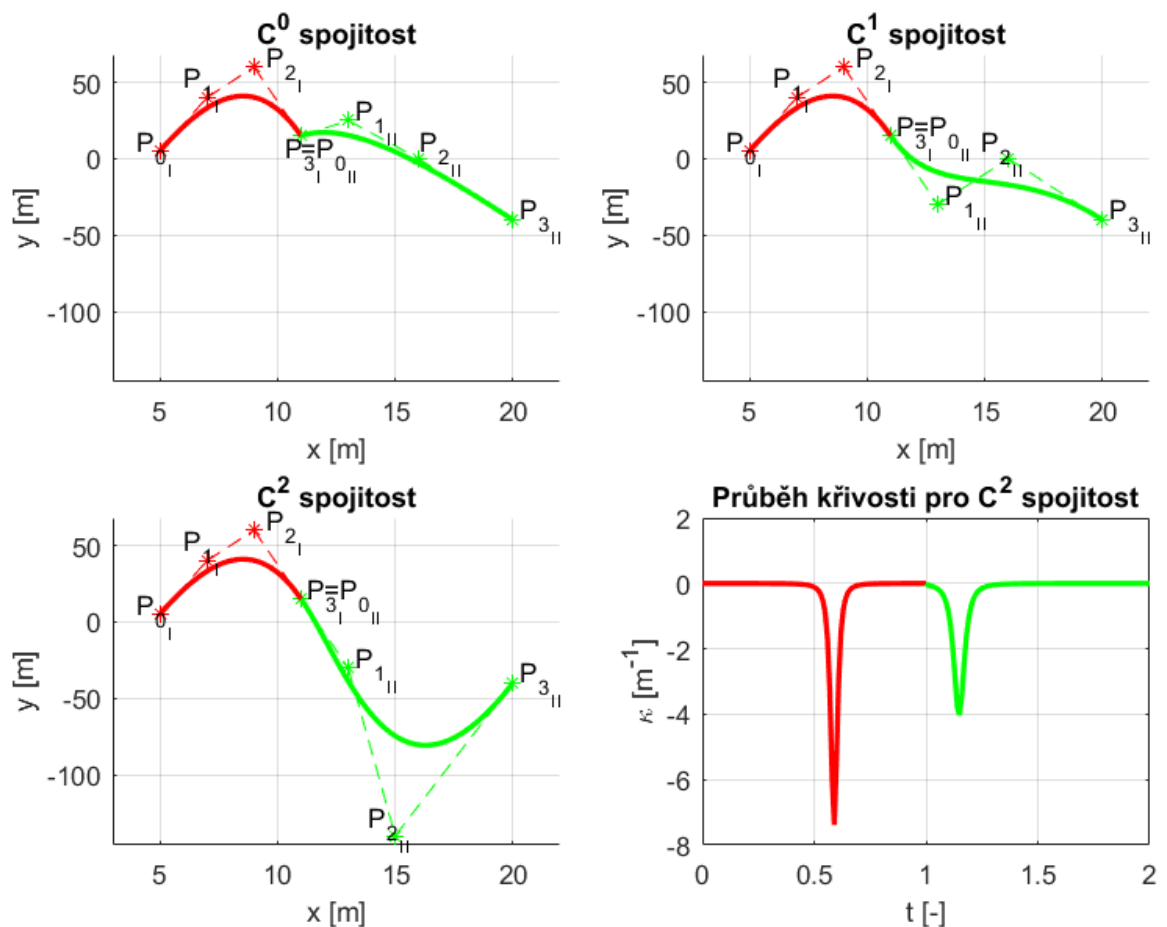
$$6\mathbf{P}_{1_I} - 12\mathbf{P}_{2_I} + 6\mathbf{P}_{3_I} = 6\mathbf{P}_{0_{II}} - 12\mathbf{P}_{1_{II}} + 6\mathbf{P}_{2_{II}}. \quad (9.23)$$

Za body  $\mathbf{P}_{0_{II}}$  a  $\mathbf{P}_{1_{II}}$  dosadíme z dříve odvozených vztahů (9.15) a (9.20):

$$6\mathbf{P}_{1_I} - 12\mathbf{P}_{2_I} + 6\mathbf{P}_{3_I} = 6\mathbf{P}_{3_I} - 12(2\mathbf{P}_{3_I} - \mathbf{P}_{2_I}) + 6\mathbf{P}_{2_{II}} \quad (9.24)$$

a dále upravíme, až dostaneme konečný tvar pro vyjádření třetího bodu druhé kubiky:

$$\mathbf{P}_{2_{II}} = \mathbf{P}_{1_I} - 4\mathbf{P}_{2_I} + 4\mathbf{P}_{3_I}. \quad (9.25)$$

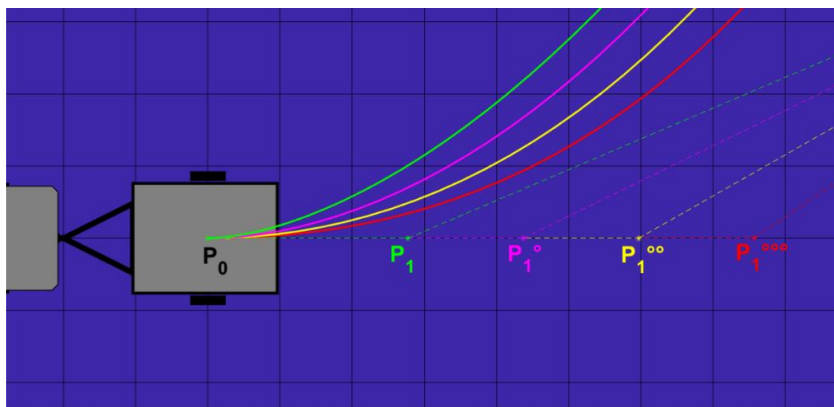


Obr. 23 – Ukázka různých řádů spojitosti derivace křivek v místě jejich napojení

Získal jsem výše popsaným postupem podmínky, které musí splňovat řídicí body druhé křivky, aby byl splněn požadavek na spojitost obou křivek až do druhého řádu derivace. Ukázkou křivek s různým řádem spojitosti derivace v místě napojení poskytuje Obr. 23. Z grafu průběhu křivostí obou křivek na tomto obrázku je také vidět spojitost křivostí obou křivek v místě jejich napojení.

Pomocí odvozených vztahů mohu libovolně napojovat více Bézierových kubik za účelem popisu trajektorie, která bude mít po celém svém průběhu spojitou křivost. Bohužel se podle odvozených vztahů ukazuje, že jediný řídicí bod z druhé kubiky, kterým lze dle libosti manipulovat je pouze bod poslední. Poloha prvních tří řídicích bodů druhé kubiky je dána svou závislostí na řídicích bodech kubiky první. Navíc pro možnost volby natočení přívěsu v koncovém bodě trajektorie je nutné správně zavázat poslední dva body kubiky, která tvoří poslední segment řetězce křivek. Podle odvozených vztahů však tyto dva poslední body není možné libovolně volit a bylo by nutné složitým způsobem vyjadřovat závislosti opačným způsobem, čímž by se ale omezila tvarovatelnost křivek tvořících řetězec „před“ posledním úsekem. Možným řešením by mohlo být použití velkého množství kubik napojených za sebou, čímž by se ve výsledku získalo dostatečné množství volných parametrů pro správné natvarování trajektorie. Jelikož však hodlám v další části práce užít pro hledání výsledného tvaru trajektorie optimalizačních algoritmů, vedl by velký počet pospojovaných kubik k příliš velkému počtu optimalizačních parametrů. Rychlost výpočtu by tak pro účely parkovacího asistenta nejspíše nebyla dostatečná, navíc celý algoritmus by narůstal na složitosti.

Na základě výše zmíněné úvahy jsem se rozhodl, že metoda napojování kubik není pro účely vyhledávání optimální trajektorie vhodná. V dalším postupu jsem proto ustoupil od klasicky užívaných kubik a rozhodl se pro popis trajektorie využít jedinou Bézierovu křivku vyššího řádu. Jak bylo výše uvedeno, křivka řádu  $n$  má  $n + 1$  řídicích bodů, tento počet řídicích bodů představuje v rovině obecně  $2(n + 1)$  volných parametrů křivky. Při popisu trajektorie je vždy pevně dána poloha počátečního a koncového bodu a poloha druhého a předposledního bodu je omezena požadavkem na směr odjezdu z počátku, respektive příjezdu do cíle. Počet volných parametrů křivky je tedy třeba snížit o 4 za počáteční a koncový bod (2 za každý) a dále o 2 za bod druhý a předposlední (1 za každý). Druhý a



předposlední bod jsou vázány požadovanou orientací vzhledem k bodu prvními, respektive poslednímu, umožňují však volit vzdálenost od těchto bodů – viz Obr. 24.

Obr. 24 – Omezení polohy bodu  $P_1$  vůči  $P_0$

Ve výsledku tak získávám pro mnou definovanou úlohu popisu trajektorie pro Bézierovu křivku řádu  $n$  počet volných parametrů odpovídající:

$$n_p = 2(n + 1) - 6 = 2n - 4. \quad (9.26)$$

Je třeba zvolit řád křivky, který poskytne dostatečný počet volných parametrů, není ovšem vhodné volit vzhledem k výpočetní náročnosti zbytečně vysoký řád křivky. Pro volbu řádu křivky  $n = 5$  dostávám podle vztahu (9.26) 6 volných parametrů pro tvarování křivky, tento počet považuji za vhodný kompromis mezi dobrou tvarovatelností a nepřilíš vysokými výpočetními požadavky pro optimalizaci.

### 9.3. Bézierova křivka pátého řádu

Rozhodl jsem se na základě výše uvedených úvah pro popis trajektorie přívěsu využít Bézierovu křivku pátého řádu. Na rozdíl od často používaných kubik se mi nepodařilo dohledat tvar této křivky v rozepsané formě v mých zdrojích, musel jsem ho tedy vypočítat z definice křivky pomocí Bernsteinových polynomů podle rovnic (9.1) a (9.2). Nejdříve si vyjádřím Bernsteinovy polynomy pátého řádu:

$$b_{0,5}(t) = \binom{5}{0} t^0 (1-t)^{5-0} = (1-t)^5, \quad (9.27)$$

$$b_{1,5}(t) = \binom{5}{1} t^1 (1-t)^{5-1} = 5t(1-t)^4, \quad (9.28)$$

$$b_{2,5}(t) = \binom{5}{2} t^2 (1-t)^{5-2} = 10t^2(1-t)^3, \quad (9.29)$$

$$b_{3,5}(t) = \binom{5}{3} t^3 (1-t)^{5-3} = 10t^3(1-t)^2, \quad (9.30)$$

$$b_{4,5}(t) = \binom{5}{4} t^4 (1-t)^{5-4} = 5t^4(1-t), \quad (9.31)$$

$$b_{5,5}(t) = \binom{5}{5} t^5 (1-t)^{5-5} = t^5. \quad (9.32)$$

Dosadím do rovnice (9.1) a získávám první tvar Bézierovy křivky pátého řádu:

$$\begin{aligned} B(t) = & (1-t)^5 P_0 + 5t(1-t)^4 P_1 + 10t^2(1-t)^3 P_2 \\ & + 10t^3(1-t)^2 P_3 + 5t^4(1-t) P_4 + t^5 P_5. \end{aligned} \quad (9.33)$$

Pro získání tvaru s řídicími body jako koeficienty u jednotlivých řádů parametru provedu roznásobení závorek ve vztahu (9.33) a upravím do příslušného tvaru, jednotlivé kroky postupu zde nebudu uvádět. Po provedení matematických úprav získáváme alternativní zápis Bézierovy křivky pátého řádu:

$$\begin{aligned} B(t) = & (-P_0 + 5P_1 - 10P_2 + 10P_3 - 5P_4 + P_5)t^5 \\ & + (5P_0 - 20P_1 + 30P_2 - 20P_3 + 5P_4)t^4 \\ & + (-10P_0 + 30P_1 - 30P_2 + 10P_3)t^3 \\ & + (10P_0 - 20P_1 + 10P_2)t^2 \\ & + (-5P_0 + 5P_1)t \\ & + P_0. \end{aligned} \quad (9.34)$$

I pro křivku pátého řádu jsem provedl odvození vztahů mezi řídicími body, které při napojování zajistí splnění požadavku spojitosti derivace do druhého řádu v místě napojení křivek. V další části postupu jsem však napojování křivek pátého řádu nevyužil, protože počet volných parametrů u křivky pátého řádu se ukázal jako dostatečný pro tvarování trajektorie. Odvozené vztahy přesto uvedu, avšak bez postupu, odvození jsem prováděl stejným způsobem jako u křivek třetího řádu. Ukázalo se, že vztahy pro napojování křivek pátého řádu vedou na analogické vztahy, jako u napojování křivek třetího řádu, konkrétně tedy:

$$P_{0II} = P_{5I}, \quad (9.35)$$

$$P_{1II} = 2P_{5I} - P_{4I}, \quad (9.36)$$

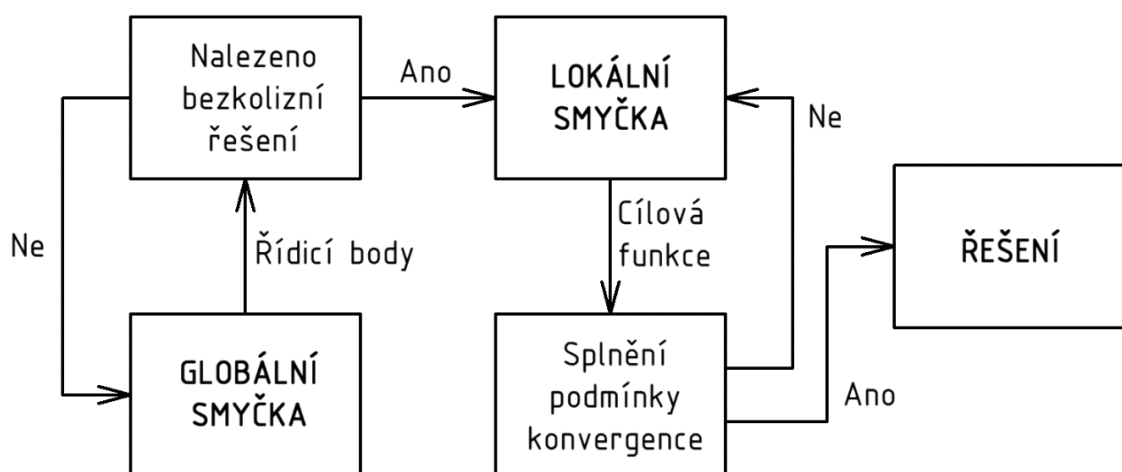
$$P_{2II} = P_{3I} - 4P_{4I} + 4P_{5I}. \quad (9.37)$$

Získal jsem tedy matematický popis křivky, kterou budu dále využívat pro popis trajektorie, dalším krokem postupu bude vytvořit algoritmus pro vyhledání a optimalizaci této trajektorie.

## 10. Plánování a optimalizace trajektorie

V této části diplomové práce se budu věnovat algoritmu pro hledání a optimalizaci trajektorie, který jsem vytvořil pro účely parkovacího asistenta. Spojuje poznatky a postupy popsané v minulých kapitolách. Vycházím z mapy prostředí vytvořené podle popisu v kapitole 8, vyhledávaná trajektorie je ve formě Bézierovy křivky pátého řádu podle kapitoly 9 a uvažuji soupravu automobilu s přívěsem popsanou v kapitole 7.

Pro optimalizaci trajektorie používám v Matlabu lokální optimalizační metodu založenou na simplexové metodě. Optimalizační algoritmus implementovaný jako funkce v Matlabu se jmenuje *fminsearch* [27], [28]. Jako nadstavbu jsem přidal smyčku programu, která nejdříve prohledává zmapovaný prostor globálně (volí náhodně řídicí body křivky), dokud nenajde řešení, které je podél trajektorie bezkolizní, dále předává vyhovující řídicí body do algoritmu lokální optimalizace, který dále na základě mnou vytvořené cílové funkce optimalizuje polohu řídicích bodů pro nalezení ideální trajektorie. Nedá se říci, že by se výsledná optimalizační metoda dala považovat za algoritmus globální optimalizace, ale vykazuje určité příslušné vlastnosti. Splňuje požadavek na náhodný výběr prvotního řešení, po nalezení první vyhovující konfigurace řídicích bodů však dále ve výpočtu nezohledňuje jiná možná řešení. Není tedy zaručeno nalezení zcela optimálního řešení, to však nezaručují ani algoritmy globální. Vyhledávání a optimalizaci trajektorie jsem se během navrhování algoritmu pokusil řešit i přímo globálními metodami, které jsou implementované v Matlabu, výpočetní doba však byla násobně vyšší (pro asistenční systém do automobilu zásadní problém, i když se nejedná o proces, který by musel běžet v reálném čase) a dosažené výsledky ve většině situací nevykazovaly zlepšení, často spíše zhoršení. Rozhodl jsem se tedy zůstat u výše zmíněného řešení s „globální nadsmyčkou“ a následnou optimalizací na lokální úrovni. Schematicky běh algoritmu znázorňuje obrázek Obr. 25.



Obr. 25 – Schéma nejvyšší úrovně algoritmu parkovacího asistenta

### 10.1. Globální smyčka pro nalezení prvotní bezkolizní trajektorie

Úlohou této programové smyčky algoritmu parkovacího asistenta je nalezení prvotní trajektorie, která se bude vyhýbat překážkám v okolním prostředí. Trajektorie je popsána Bézierovou kubikou pátého řádu a je dána svými šesti řídicími body:

$$\mathbf{P}_0 = \begin{bmatrix} x_{P_0} \\ y_{P_0} \end{bmatrix}, \quad (10.1)$$

$$\mathbf{P}_1 = \begin{bmatrix} x_{P_1} \\ y_{P_1} \end{bmatrix}, \quad (10.2)$$

$$\mathbf{P}_2 = \begin{bmatrix} x_{P_2} \\ y_{P_2} \end{bmatrix}, \quad (10.3)$$

$$\mathbf{P}_3 = \begin{bmatrix} x_{P_3} \\ y_{P_3} \end{bmatrix}, \quad (10.4)$$

$$\mathbf{P}_4 = \begin{bmatrix} x_{P_4} \\ y_{P_4} \end{bmatrix}, \quad (10.5)$$

$$\mathbf{P}_5 = \begin{bmatrix} x_{P_5} \\ y_{P_5} \end{bmatrix}. \quad (10.6)$$

Body  $\mathbf{P}_0$  a  $\mathbf{P}_5$  jsou dány začátkem a koncem trajektorie, body  $\mathbf{P}_1$  a  $\mathbf{P}_4$  jsou omezeny tečnou k trajektorii v počátečním a koncovém bodě. Tečna trajektorie svírá s kladnou osou  $x$  ve svém počátku úhel  $\eta_1$  a na svém konci úhel  $\eta_2$ . Pak platí pro body  $\mathbf{P}_1$  a  $\mathbf{P}_4$  rovnice:

$$\mathbf{P}_1 = \begin{bmatrix} x_{P_1} \\ y_{P_1} \end{bmatrix} = \begin{bmatrix} x_{P_0} + \Delta_1 \cos(\eta_1) \\ y_{P_0} + \Delta_1 \sin(\eta_1) \end{bmatrix}, \quad (10.7)$$

$$\mathbf{P}_4 = \begin{bmatrix} x_{P_4} \\ y_{P_4} \end{bmatrix} = \begin{bmatrix} x_{P_5} - \Delta_2 \cos(\eta_2) \\ y_{P_5} - \Delta_2 \sin(\eta_2) \end{bmatrix}. \quad (10.8)$$



Z 12 řídicích bodů zbývá, jak bylo výše vysvětleno, 6 volných parametrů. Tyto parametry jsou:

$$\vec{p} = \begin{bmatrix} \Delta_1 \\ x_{P_2} \\ y_{P_2} \\ x_{P_3} \\ y_{P_3} \\ \Delta_2 \end{bmatrix}, \quad (10.9)$$

kde  $\Delta_1$  a  $\Delta_2$  jsou vzdálenosti mezi prvními dvěma, respektive posledními dvěma řídicími body křivky a  $x_{P_i}$  a  $y_{P_i}$  jsou x-ová, respektive y-ová souřadnice i-tého řídicího bodu.

Těchto šest parametrů je náhodně voleno, dokud neposkytnou jako výsledek prvotní bezkolizní trajektorii. Vyhodnocení, zda podél trajektorie dochází či nedochází ke kolizi provádí funkce *Detekce\_kolize.m*. Na Obr. 26 je ukázka kódu z programu.

```

6
7  %% Počáteční poloha soupravy, koncová poloha soupravy
8 - eta1 = 0/4*pi; % Úhel tečny trajektorie v jejím počátečním bodě
9 - eta2 = 0/4*pi; % Úhel tečny trajektorie v jejím koncovém bodě
10 - delta1_0 = 10+randi([1 5]); % Vzdálenost mezi prvními dvěma řídicími body
11 - delta2_0 = 10+randi([1 5]); % Vzdálenost mezi posledními dvěma řídicími body
12
13 - P0 = [5 5]'; % počáteční bod trajektorie
14 - P5 = [48 36.2]'; % koncový bod trajektorie
15
16 - P1_0 = [P0(1)+delta1_0*cos(eta1) P0(2)+delta1_0*sin(eta1)]';
17 - DOTAZ = 'Přejete si zvolit dva body podél trajektorie? [ANO/NE]';
18 - ODPOVED = input(DOTAZ, 's');
19 - if string(ODPOVED) == 'ANO'
20 -     P2_0 = [15 10]'; % V asistentovi se otevře okno pro zvolení bodů
21 -     P3_0 = [20 25]'; % V asistentovi se otevře okno pro zvolení bodů
22 - else
23 -     P2_0 = [randi([10 60]) randi([10 60])]';
24 -     P3_0 = [randi([10 60]) randi([10 60])]';
25 - end
26 - P4_0 = [P5(1)-delta2_0*cos(eta2) P5(2)-delta2_0*sin(eta2)]';
27
28  %% Nalezení bezkolizní trajektorie
29 - while Detekce_kolize(P0, P1_0, P2_0, P3_0, P4_0, P5)>0
30 -     delta1_0 = randi([1 10]);
31 -     delta2_0 = randi([1 10]);
32 -     P1_0 = [P0(1)+delta1_0*cos(eta1) P0(2)+delta1_0*sin(eta1)]';
33 -     P2_0 = [randi([10 60]) randi([10 60])]';
34 -     P3_0 = [randi([10 60]) randi([10 60])]';
35 -     P4_0 = [P5(1)-delta2_0*cos(eta2) P5(2)-delta2_0*sin(eta2)]';
36 - end
37

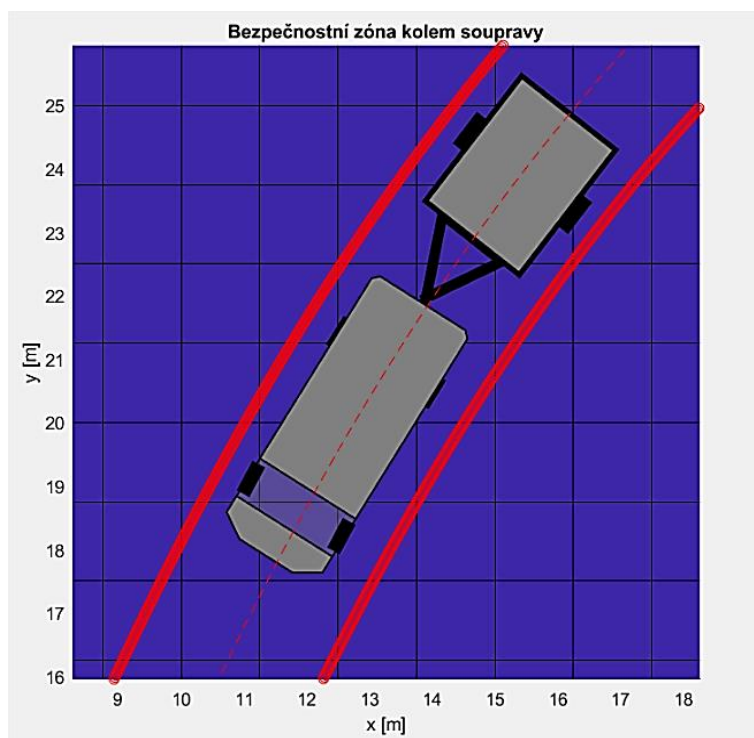
```

Obr. 26 – Ukázka kódu – globální smyčka hledání trajektorie

Počáteční odhad bodů  $P_2$  a  $P_3$  je také možno zvolit v asistentovi manuálně, tím lze urychlit proces vyhledání bezkolizní trajektorie. Také je tím možno v případě více průjezdných cest naznačit asistentovi, kudy si přeje řidič manévr provádět. Křivka sice těmito body přímo neprochází, avšak volbou jejich polohy ji lze „přetvarovat“.

### 10.2. Detekce kolize

Funkce *Detekce\_kolize.m* má za úkol kontrolovat průjezdnost navržené trajektorie. Provádí tak srovnáváním dvou map, které v sobě mají zanesené binární informace. Tyto mapy jsou reprezentovány maticemi, každý prvek matice je buď jedna nebo nula. První z těchto matic odpovídá mapě okolního prostředí, kterou jsem si již dříve připravil a popisoval v kapitole 8. Označím tuto matici  $M_P$  (matice překážek). Matice překážek je rozměru  $(m \times n)$ , kde  $m$  odpovídá počtu buněk v mapě okolního prostředí v y-ovém směru a  $n$  odpovídá počtu buněk v x-ovém směru. Prvky matice  $M_P$ , které odpovídají buňkám obsazeným překážkami mají hodnotu 1, ostatní prvky matice  $M_P$  mají hodnotu 0. Druhou matici označím  $M_T$  (matice trajektorie). Matice trajektorie má stejný rozměr jako matice překážek  $(m \times n)$ . Tato matice je vytvořena na základě zpracování navržené trajektorie. Každému bodu této trajektorie jsem přiřadil buňku, do které spadá v mapě okolního prostředí. Prvky matice  $M_T$ , které odpovídají buňkám, kterými prochází navržená trajektorie mají hodnotu 1, ostatní prvky matice  $M_T$  mají hodnotu 0. Takovýmto způsobem jsem schopen identifikovat buňky, kterými prochází navržená Bézierova křivka, tato křivka však popisuje pohyb pouze středu nápravy přívěsu a neuvažuje rozměry soupravy. K navržené trajektorii jsem tedy sestrojil z obou stran ekvidistanty, které kolem trajektorie vytyčují „bezpečnou zónu“.



nepřesnosti mapy a zaručuje minimální vzdálenost od překážek během průjezdu. Ukázka podle Obr. 27, kde červená čárkovaná čára je vypočítaná trajektorie středu nápravy přívěsu, tlustá červená čára ohraničuje bezpečnou zónu kolem soupravy. Do matice trajektorie tedy zahrnu i buňky, kterými sice neprochází sama trajektorie, ale jsou uvnitř bezpečné zóny.

Obr. 27 – Ilustrace bezpečnostní zóny kolem soupravy

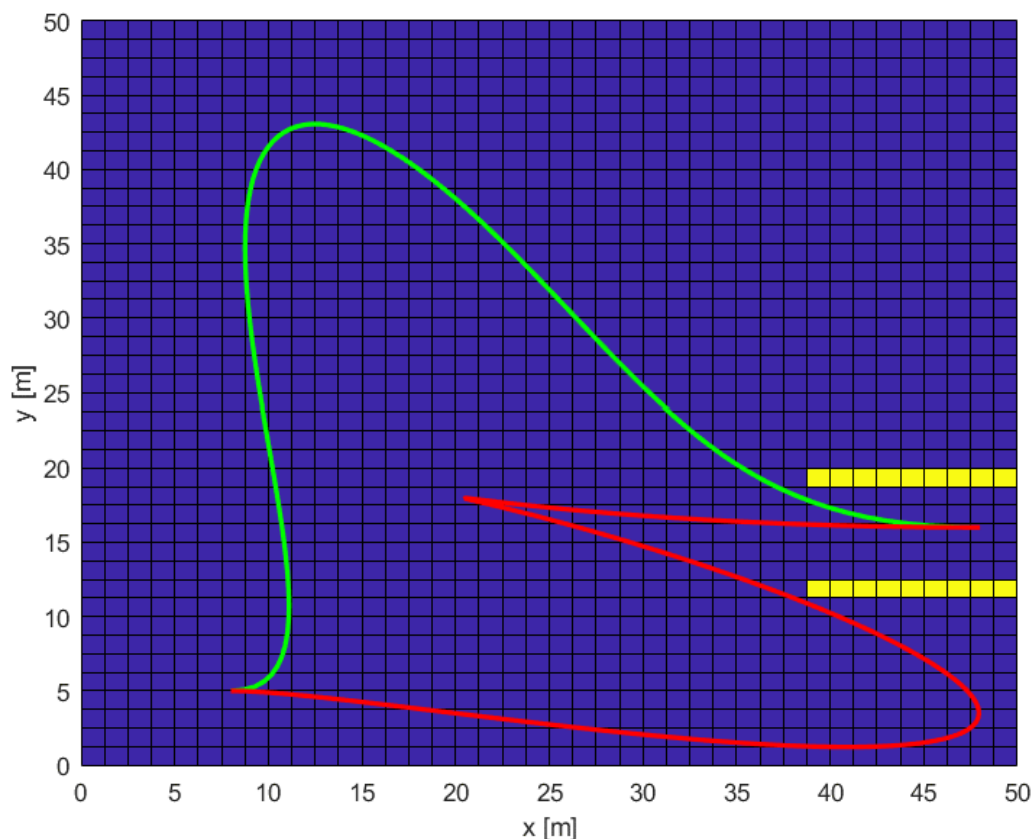
Výsledné vyhodnocení přítomnosti či nepřítomnosti kolize provedu vynásobením matic  $\mathbf{M}_P$  a  $\mathbf{M}_T$  systémem „každý s každým“, výsledkem je matice  $\mathbf{M}_K$  (matice kolize):

$$\mathbf{M}_{K_{ij}} = \mathbf{M}_{P_{ij}} \mathbf{M}_{T_{ij}}, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n. \quad (10.10)$$

Pokud alespoň jeden prvek výsledné matice  $\mathbf{M}_K$  splňuje  $\mathbf{M}_{K_{ij}} > 0$ , pak je detekována kolize a je třeba dále vyhledávat bezkolizní trajektorii, viz Obr. 25. Zvolí se nový vektor parametrů  $\vec{p}$  a celý proces se opakuje, dokud se nepodaří nalézt průjezdnou trajektorii, nebo dokud není dosaženo maximálního počtu pokusů, po kterém se vyhledávání ukončí bez nalezení vyhovujícího řešení, o čemž je uživatel informován chybovou hláškou.

### 10.3. Lokální optimalizace navržené trajektorie

Po získání prvotního návrhu trajektorie v globální smyčce jsou vyhovující parametry v programu předány dále jako vstupy do funkce *fminsearch*. Návrh trajektorie z globální smyčky sice zaručuje, že nedojde ke kolizi s překážkami, trajektorie však není optimalizována z hlediska její délky či ostroty zatáček. Příklady neoptimalizovaných trajektorií uvádí Obr. 28.



Obr. 28 – Ukázka neoptimalizovaných trajektorií

Zelená i červená trajektorie z Obr. 28 jsou sice bezkolizní, rozhodně je však nelze považovat za optimální. Červená trajektorie je již od pohledu neprůjezdná kvůli příliš velké křivosti v jedné ze zatáček. Zelená trajektorie vypadá od pohledu co do křivosti průjezdná, avšak je zbytečně dlouhá a rozhodně nepředstavuje pro řidiče intuitivní a pohodlný manévr.

Za účelem optimalizace trajektorie jsem navrhl cílovou funkci, při jejíž minimalizaci je dosaženo ideální trajektorie pro provedení manévru couvání. Cílová funkce je navržena jako funkce vektoru volných parametrů Bézierovy křivky  $\vec{p}$  a její minimum je hledáno simplexovou metodou pomocí výše zmíněného optimalizačního algoritmu *fminsearch* v Matlabu. Cílová funkce v sobě zahrnuje požadavek na bezkoliznost, zohledňuje délku trajektorie a průběh křivosti podél trajektorie. Výsledná cílová funkce je dána součtem více funkcí, které popisují tyto vlastnosti, jedná se tedy o multikriteriální optimalizaci. Jednotlivé cílové funkce tvořící výslednou cílovou funkci rozvedu v dalších částech této kapitoly.

Výslednou cílovou funkci jsem navrhl jako:

$$CF = k_1 CF_1 + k_2 CF_2 + k_3 CF_3 + k_4 CF_4, \quad (10.11)$$

kde  $CF_i$  je  $i$ -tá cílová funkce a  $k_i$  je váhový koeficient  $i$ -té cílové funkce.

Dříve nezmiňovanou výhodou popisu trajektorie formou Bézierovy křivky je to, že mám k dispozici přesný analytický popis trajektorie, čehož využiji při vyjadřování vztahů v dalších částech kapitoly 10.

### 10.4. Minimalizace délky trajektorie

Aby byl optimalizační algoritmus schopen nalézt optimální parametry (řídící body Bézierovy křivky) pro minimalizaci délky trajektorie, musím tuto délku vyjádřit jako funkci těchto parametrů. Délku parametricky<sup>1</sup> vyjádřené křivky  $\mathbf{B}(t)$  je možno vypočítat pomocí křivkového integrálu integrací podle parametru [29] podle vztahu:

$$s = \int_{\mathbf{B}} 1 ds = \int_0^1 \sqrt{B'_x(t)^2 + B'_y(t)^2} dt, \quad (10.12)$$

kde  $s$  je výsledná délka křivky,  $B'_x(t)$  je derivace x-ové složky křivky  $\mathbf{B}$  podle parametru  $t$  a  $B'_y(t)$  je derivace y-ové složky křivky  $\mathbf{B}$  podle parametru  $t$ .

---

<sup>1</sup> Poznámka: V textu je třeba rozlišovat: 1) optimalizační parametry, které jsou dány vektorem  $\vec{p}$  – jsou to hledané řídící body Bézierovy křivky a 2) parametr Bézierovy křivky  $t$  – křivka je dána svým parametrickým popisem  $\mathbf{B}(t)$ .

Derivaci Bézierovy křivky jsem uváděl již dříve v kapitole 9.2, ne však pro Bézierovu křivku pátého řádu, kterou zde používám. Nejdříve zavedu pro větší přehlednost substituci za koeficienty Bézierovy křivky:

$$\begin{aligned}
 \mathbf{B}(t) = & \underbrace{(-\mathbf{P}_0 + 5\mathbf{P}_1 - 10\mathbf{P}_2 + 10\mathbf{P}_3 - 5\mathbf{P}_4 + \mathbf{P}_5)}_a t^5 \\
 & + \underbrace{(5\mathbf{P}_0 - 20\mathbf{P}_1 + 30\mathbf{P}_2 - 20\mathbf{P}_3 + 5\mathbf{P}_4)}_b t^4 \\
 & + \underbrace{(-10\mathbf{P}_0 + 30\mathbf{P}_1 - 30\mathbf{P}_2 + 10\mathbf{P}_3)}_c t^3 \\
 & + \underbrace{(10\mathbf{P}_0 - 20\mathbf{P}_1 + 10\mathbf{P}_2)}_d t^2 \\
 & + \underbrace{(-5\mathbf{P}_0 + 5\mathbf{P}_1)}_e t \\
 & + \underbrace{\mathbf{P}_0}_f
 \end{aligned} \tag{10.13}$$

Její derivaci pak vyjádřím jako:

$$\mathbf{B}'(t) = 5\mathbf{a}t^4 + 4\mathbf{b}t^3 + 3\mathbf{c}t^2 + 2\mathbf{d}t + \mathbf{e}. \tag{10.14}$$

Po rozepsání do složek:

$$\mathbf{B}'(t) = \begin{bmatrix} B'_x(t) \\ B'_y(t) \end{bmatrix} = \begin{bmatrix} 5a_x t^4 + 4b_x t^3 + 3c_x t^2 + 2d_x t + e_x \\ 5a_y t^4 + 4b_y t^3 + 3c_y t^2 + 2d_y t + e_y \end{bmatrix}, \tag{10.15}$$

kde  $a_x$  je x-ová složka koeficientu  $\mathbf{a}$  Bézierovy křivky podle (10.13) a  $a_y$  je jeho y-ová složka. Pro ostatní koeficienty platí totéž.

Výrazy z (10.15) dosadím do (10.12) a získávám délku trajektorie v závislosti na řídicích bodech Bézierovy křivky pátého řádu. Integraci provádím numericky v Matlabu, výraz je příliš složitý pro analytické vyjádření.

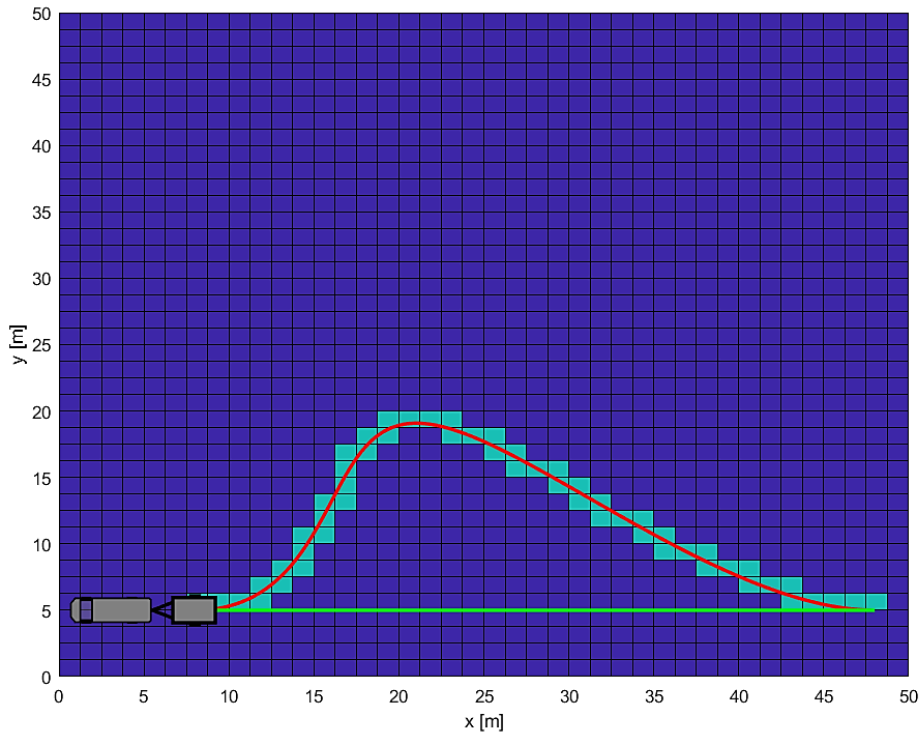
Cílovou funkci s jejím váhovým koeficientem zohledňující délku křivky jsem definoval jako:

$$k_1 C F_1 = s, \tag{10.16}$$

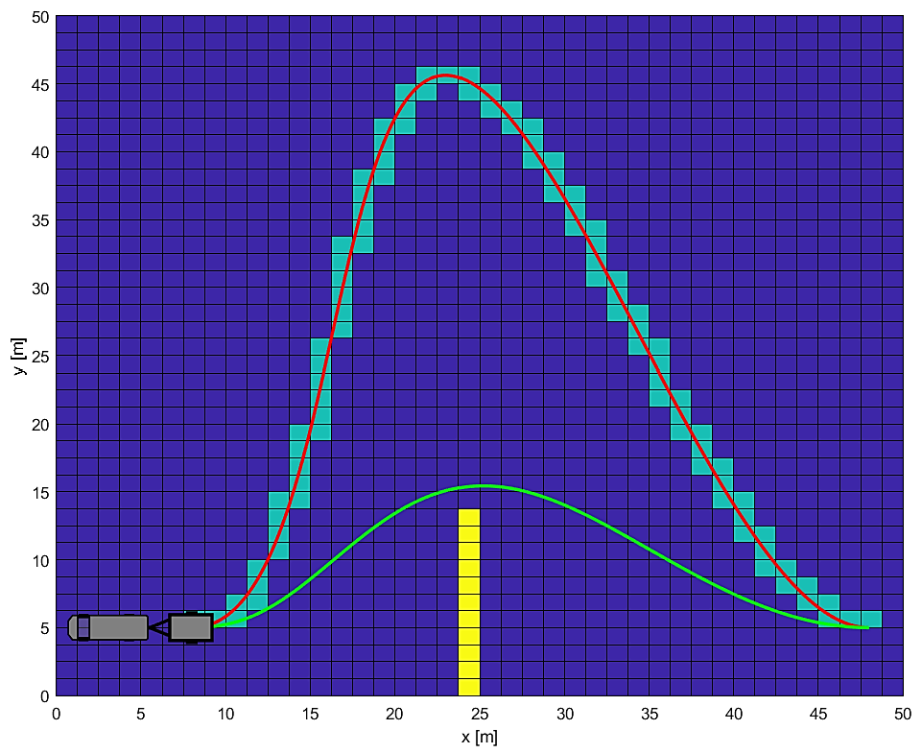
kde  $s$  je délka křivky podle (10.12) a koeficient  $k_1 = 1$ .

Vhodná volba váhového koeficientu příslušné cílové funkce je problematická a volby jednotlivých váhových koeficientů okomentuji souhrnně v dalších částech práce. Vliv cílové funkce s jejím koeficientem na tvar optimalizované trajektorie demonstrují Obr. 29 a Obr. 30. Červená křivka představuje na obou obrázcích návrh trajektorie z globální smyčky,

zelená křivka pak představuje křivku s optimalizovanou délkou. Z Obr. 29 je patrné, že pokud není mezi počátečním a koncovým bodem přítomna překážka a pokud se nemění natočení přívěsu, křivka obecného tvaru přejde v úsečku, což je nejkratší spojnice obou bodů. V tomto případě se jedná o speciální případ Bézierovy křivky, kdy všechny řídicí body leží na společné přímce a křivka prochází všemi těmito body.



Obr. 29 – Minimalizace délky trajektorie (bez překážek)



Obr. 30 – Minimalizace délky trajektorie (s překážkou)

## 10.5. Omezení křivosti trajektorie

Průběh křivosti podél trajektorie nesmí přesáhnout stanovenou maximální hodnotu. Toto omezení je dáno minimálním poloměrem otáčení přívěsu kolem jeho okamžitého středu otáčení. Hodnota minimálního poloměru otáčení přívěsu je dána mimo jiné i maximálním přípustným úhlem zalomení přívěsu vůči automobilu, ten může být omezen i mechanickým dorazem v tažném zařízení. Jelikož ve své práci neuvažuji žádné konkrétní tažné zařízení, budu hodnotu minimálního poloměru otáčení volit. Ve funkci asistenta se jedná o konstantu, kterou lze upravit pro užití na konkrétních soupravách automobilu s přívěsem.

Po rozvaze o hodnotě maximální povolené křivosti je dalším krokem vyjádřit za účelem optimalizace křivost trajektorie jako funkci řídicích bodů Bézierovy křivky, postup bude obdobný jako při vyjadřování délky. Absolutní hodnota křivosti parametricky vyjádřené křivky  $\mathbf{B}(t)$  je pro konkrétní volbu parametru rovna [30]:

$$\kappa(t) = \frac{1}{R(t)} = \frac{|B'_x B''_y - B'_y B''_x|}{(B'^2_x + B'^2_y)^{\frac{3}{2}}} \quad (10.17)$$

Hodnoty derivací  $B'_x$  a  $B'_y$  jsou dány vztahem (10.15), hodnoty druhých derivací  $B''_x$  a  $B''_y$  vypočítám z rovnice (10.14):

$$\mathbf{B}''(t) = 20at^3 + 12bt^2 + 6ct + 2d. \quad (10.18)$$

Po rozepsání do složek:

$$\mathbf{B}''(t) = \begin{bmatrix} B''_x(t) \\ B''_y(t) \end{bmatrix} = \begin{bmatrix} 20a_x t^3 + 12b_x t^2 + 6c_x t + 2d_x \\ 20a_y t^3 + 12b_y t^2 + 6c_y t + 2d_y \end{bmatrix}. \quad (10.19)$$

Po dosazení vztahů z (10.19) do (10.17) mohu určit z řídicích bodů Bézierovy křivky křivost v každém jejím bodě daném parametrem  $t$ .

Za účelem optimalizace křivosti jsem navrhl dvě cílové funkce. Jedna z nich obstarává snižování křivosti podél trajektorie, čímž ji vyhlazuje, druhá zaručuje, že maximální křivost nepřesáhne maximální povolenou hodnotu danou jako:

$$\kappa_{Dmax} = \frac{1}{R_{Dmin}}, \quad (10.20)$$

kde  $R_{Dmin}$  je minimální dovolený poloměr otáčení přívěsu podle úvahy v úvodním odstavci této kapitoly.

Cílovou funkci, jež zajišťuje snižování křivosti podél trajektorie jsem navrhl jako:

$$k_2 CF_2 = 75 \max_t \kappa(t) = 75 \kappa_{max}, \quad (10.21)$$

kde  $\kappa_{max}$  je maximální křivost podél navrhnuté trajektorie a váhový koeficient  $k_2 = 75$ .

Minimalizací této funkce se snižuje maximální křivost podél trajektorie, alternativním přístupem k řešení problému křivosti by bylo například snižovat průměrnou křivost, či kvadratickou odchylku křivosti, mnou zvolený způsob však poskytoval uspokojivé výsledky.

Cílová funkce, která zajistí, že křivost podél trajektorie nepřesáhne maximální povolenou hodnotu pracuje opět s výpočtem křivosti, avšak má složitější tvar. Navrhl jsem ji jako:

$$k_3 CF_3 = 500 \left( \text{sign} \left( \kappa_{Dmax} - \max_t \kappa(t) \right) + 1 \right), \quad (10.22)$$

kde váhový koeficient  $k_3 = 500$ .

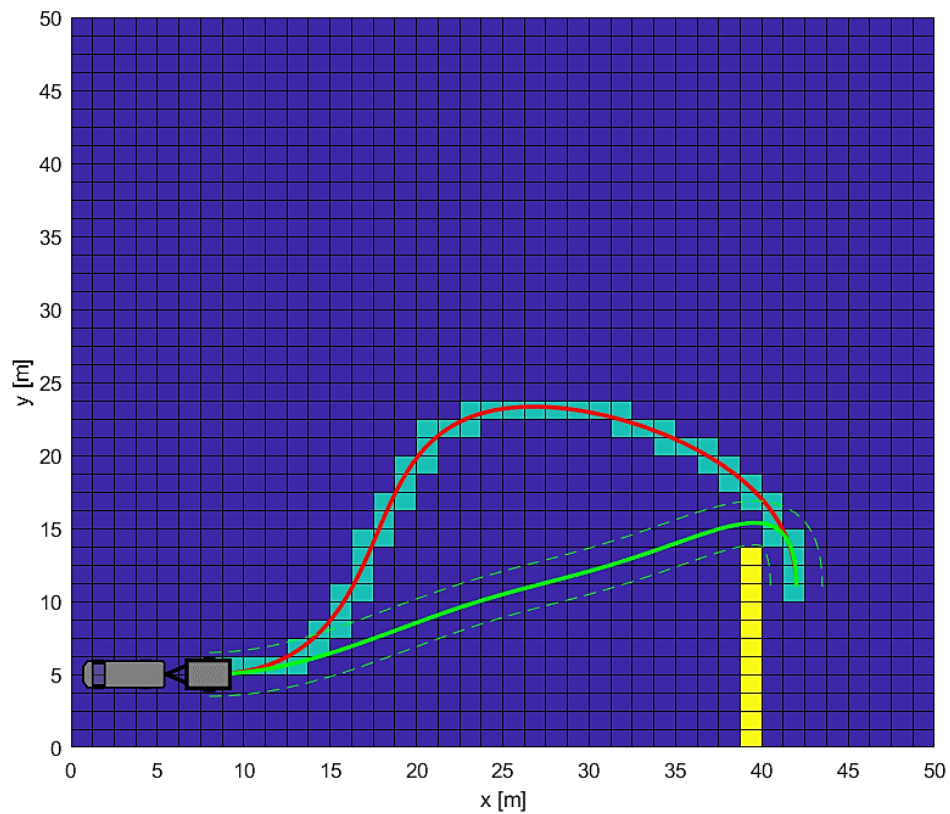
Navržená cílová funkce je nespojitá, jelikož je založena na funkci *signum*. Slovní popis funkce je následující: Pokud je maximální křivost podél navržené trajektorie vyšší než maximální dovolená křivost, přičte se prostřednictvím této cílové funkce s jejím koeficientem do celkové cílové funkce  $CF$  penalizace rovna  $2k_3$ , pokud jsou si obě křivosti rovny, přičte se penalizace  $k_3$  a pokud je maximální křivost podél trajektorie menší než maximální povolená, nepřičítá se do celkové cílové funkce žádná penalizace.

Ačkoli pracují obě cílové funkce  $CF_2$  i  $CF_3$  s křivostí, musí být rozděleny do dvou, jelikož je pro ně třeba volit rozdílné váhové koeficienty.

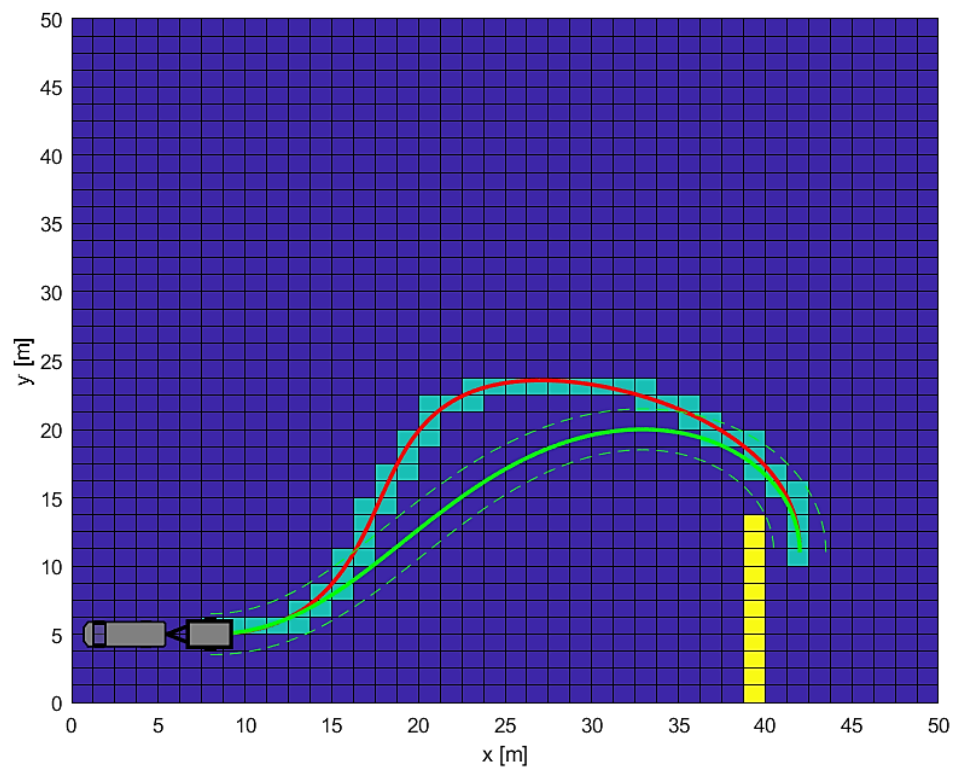
Vliv těchto dvou cílových funkcí na výsledný tvar trajektorie po optimalizaci ilustrují Obr. 31, Obr. 32 a Obr. 33. Na Obr. 31 je červenou čarou znázorněn počáteční odhad trajektorie, zelená křivka pak představuje trajektorii optimalizovanou, avšak bez uvažování omezení křivosti. Je vidět, že výsledná trajektorie se řídí pouze požadavkem na minimalizaci délky, kdy těsně obepíná překážku. Na Obr. 32 jsou znázorněny ty samé trajektorie, avšak tentokrát byl při optimalizaci brán v potaz požadavek na vyhlazení křivosti a na omezení maximální dovolené křivosti. Na Obr. 33 je znázorněn graf absolutní hodnoty křivosti v závislosti na parametru křivky  $t$ . Červená křivka představuje průběh křivosti optimalizované trajektorie z Obr. 31 a zelená křivka znázorňuje průběh křivosti optimalizované trajektorie z Obr. 32. Na zelené křivce je vidět výrazné snížení křivosti podél trajektorie, což je v souladu s požadavky navrhnutých cílových funkcí, tato trajektorie je ovšem na úkor lepšího průběhu křivosti o něco delší. Při optimalizaci jsem nastavil minimální dovolený poloměr otáčení  $R_{Dmin} = 5m$ , což odpovídá maximální povolené



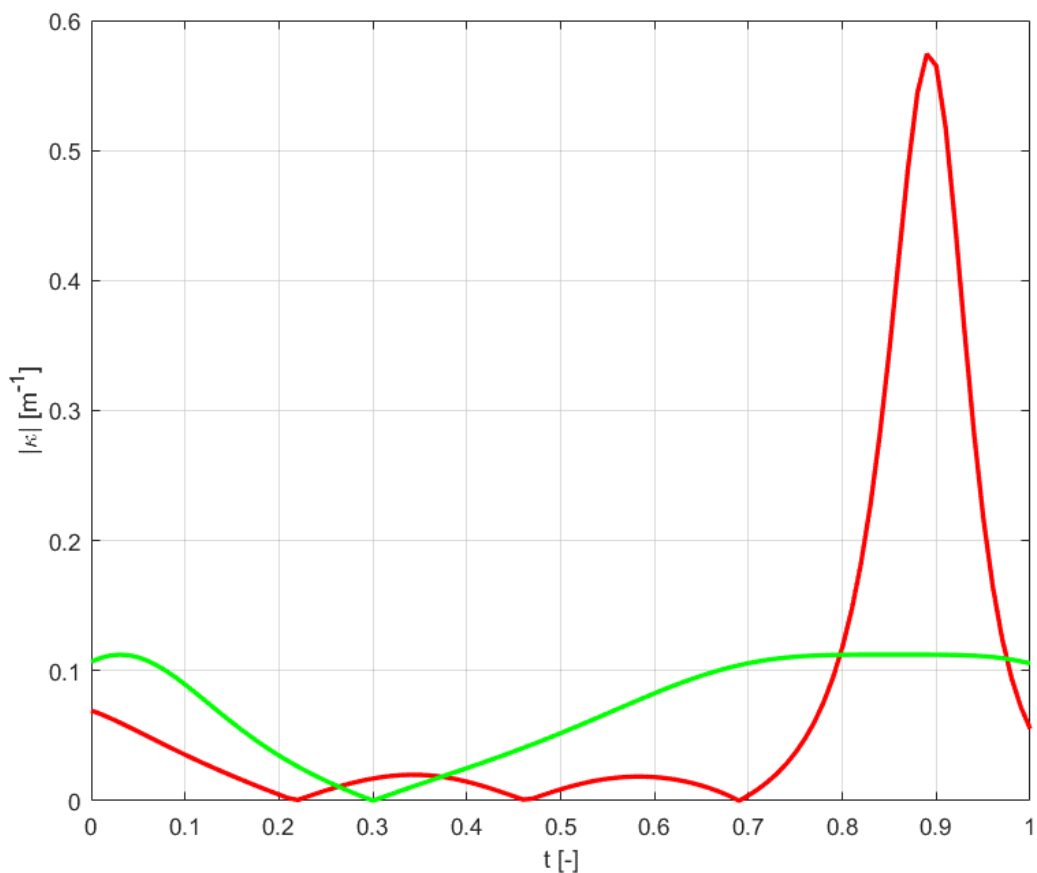
křivosti  $\kappa_{Dmax} = \frac{1}{R_{Dmin}} = 0,2 \text{ m}^{-1}$ . Z grafu křivosti je patrné, že po zahrnutí požadavku na křivost do optimalizace byla tato podmínka splněna.



Obr. 31 – Optimalizace trajektorie bez zohlednění křivosti



Obr. 32 – Optimalizace trajektorie se zohledněním křivosti



Obr. 33 – Graf průběhů křivosti s a bez optimalizace křivosti

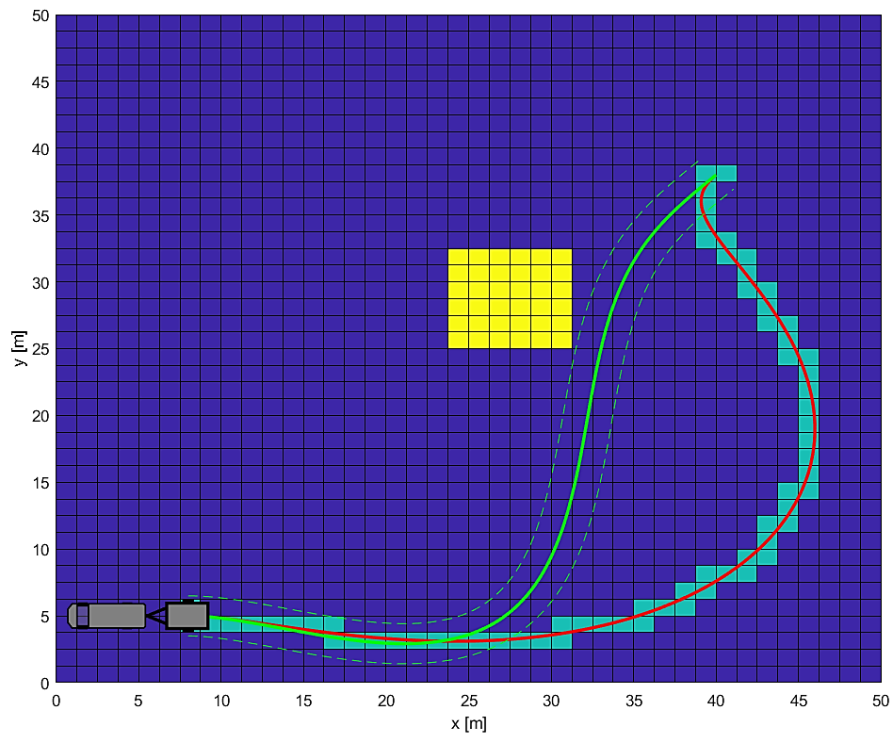
## 10.6. Zachování bezkoliznosti trajektorie

Prvotní návrh řídicích bodů Bézierovy křivky poskytuje globální smyčka optimalizačního algoritmu, která zajišťuje, že křivka daná těmito body bude podél trajektorie bezkolizní. Tyto řídicí body (respektive jejich část, která odpovídá vektoru volných parametrů  $\vec{p}$ ) jsou dále optimalizovány v lokální smyčce, za účelem snížení délky trajektorie a zajištění průjezdné křivosti trajektorie. Během tohoto optimalizačního procesu je třeba brát zřetel na to, že úpravou polohy řídicích bodů se může z původně bezkolizní trajektorie stát trajektorie, která protíná překážky. I během lokální optimalizace je tedy nutno výpočetně zaručit, že se trajektorie nepřesune do prostoru vymezeného pro překážky. Cílovou funkci, která toto zaručí jsem navrhl jako:

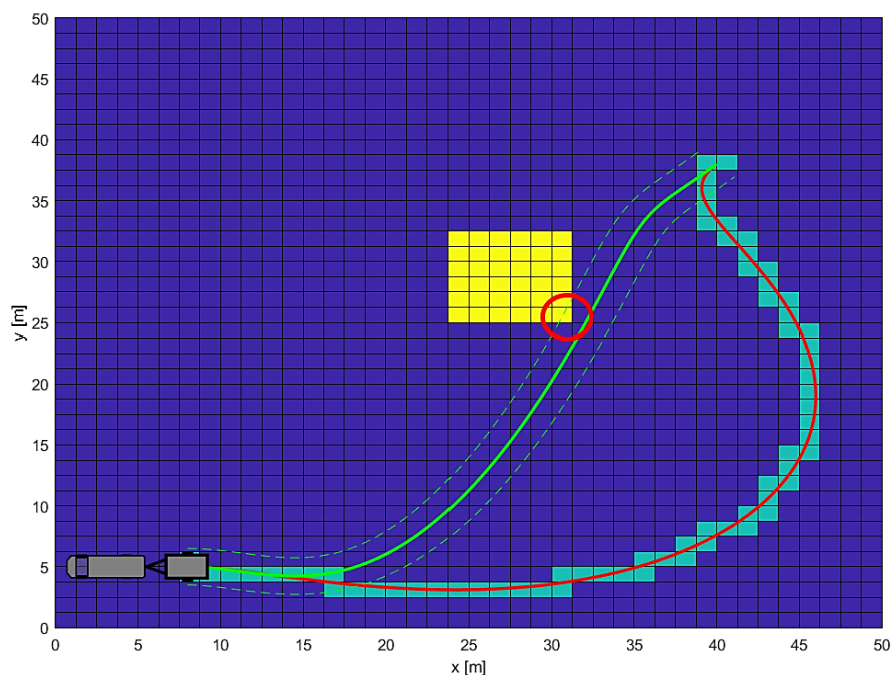
$$k_4 CF_4 = 2000 \max_{ij} \mathbf{M}_{Kij}, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n, \quad (10.23)$$

kde  $k_4 = 2000$  je váhový parametr a  $\mathbf{M}_K$  je matice kolize podle (10.10).

Funkce, která vypočítává matici kolize, je stejná jako při vyhledávání prvotního návrhu trajektorie v globální smyčce, opět je v ní zakomponovaná bezpečnostní zóna kolem přívěsu. Pokud by posunutím řídicích bodů křivky mělo dojít k průniku trajektorie s nějakou překážkou, přičte se prostřednictvím této cílové funkce do výsledné cílové funkce penalizace rovna  $k_4$ , tato penalizace je dostatečně velká, aby takové posunutí bodů nemohlo být přijato jako nové nejlepší řešení. Ukázku funkce poskytují Obr. 34 a Obr. 35. Na druhém z obrázků je vypnuta funkce zohledňující průnik trajektorie s překážkou, v místě červeného kroužku by došlo při jízdě ke kolizi soupravy s překážkou.



Obr. 34 – Optimalizace trajektorie s aktivní kontrolou kolize



Obr. 35 – Optimalizace trajektorie bez kontroly kolize

## 10.7. Volba váhových koeficientů

Volba vhodných váhových koeficientů při jednotlivých cílových funkcích je složitá záležitost a nelze ji vhodným způsobem matematicky popsat či jinak exaktně vyjádřit. Obecně se dá říci, že čím je který z daných faktorů důležitější, tím větší váhový koeficient mu přísluší. Souhrn zvolených váhových koeficientů v parkovacím asistentu:

Váhový koeficient	Funkce	Hodnota [-]
$k_1$	Minimalizace délky	1
$k_2$	Vyhlazení trajektorie	75
$k_3$	Omezení maximální křivosti	500
$k_4$	Zamezení kolize s překážkou	2000

Tabulka 1 – Souhrn zvolených váhových koeficientů v jednotlivých cílových funkcích

Například pro cílovou funkci zohledňující možnou kolizi s překážkou v prostoru je zvolena hodnota váhového koeficientu  $k_4 = 2000$ . Tato hodnota je násobně vyšší, než všechny ostatní a zaručuje, že bezkoliznost trajektorie má při optimalizaci nejvyšší prioritu. Výpočet trajektorie je určen pro relativně krátké trajektorie, kdyby však hypoteticky mohla být navržena trajektorie velice dlouhá, například  $s = 3000 \text{ m}$ , tak by výsledná „důležitost“ délky mohla být v optimalizačním algoritmu vyšší než důležitost bezkoliznosti. V takovém případě by mohl algoritmus poskytnout jako výsledek trajektorii krátkou, ale protínající překážku. Volba vah tedy není univerzální, ale je třeba uvažovat konkrétní aplikaci a různé scénáře, které mohou u dané aplikace nastat. Druhou nejvyšší váhu jsem zvolil pro zaručení průjezdnosti trajektorie ve smyslu křivosti. Je tím zaručeno, že trajektorie, která obsahuje průjezdné „zatáčky“ nebude algoritmem považována za horší než například trajektorii velmi krátkou, ale s neprůjezdnými zatáčkami. Pokud algoritmus dokáže najít bezkolizní trajektorii, ale nedokáže ji optimalizovat tak, aby splňovala požadované omezení křivosti, bude o tom uživatel informován a bude doporučeno provést nové vyhledávání trajektorie. Nejnižší prioritu jsem určil pro minimalizaci délky, která není z hlediska průjezdnosti klíčová.

K volbě výsledných hodnot váhových koeficientů jsem dospěl metodou pokus omyl, výše zmíněná kombinace poskytuje při vyhledávání dobré a stabilní výsledky, není však zaručeno, že jiná kombinace by neposkytovala výsledky lepší. Také by bylo možné přidat další cílové funkce, které by zohledňovaly jiné aspekty couvání automobilu s přívěsem, zvyšovala by se tím však zároveň výpočetní náročnost. Mnou navržené cílové funkce berou v potaz klíčové aspekty pro nalezení vhodné trajektorie pro soupravu vozidla s přívěsem. Přesnější odlaďování váhových koeficientů by bylo předmětem dlouhodobějšího testování.

## 11. Vstupy do simulačního modelu

V minulé kapitole jsem získal výsledný tvar trajektorie pro provedení parkovacího manévru. V souladu se schématem inverzní kinematiky podle Obr. 13 musím dále získat z vypočítané trajektorie kinematické veličiny pro její průjezd. Tyto kinematické veličiny použiji jako vstupy do modelu v Simulinku podle Obr. 14. Potřebné vstupy jsou průběh rychlosti a úhlové rychlosti přívěsu podél trajektorie v závislosti na zvolené veličině. Za nezávislou veličinu je možné zvolit například dráhu, či čas, já budu užívat čas. Toto rozhodnutí dělám mimo jiné proto, že integrace všech veličin v Simulinku probíhá podle času. Navíc lze jednoduše vytvořit vstup do modelu jako tzv. *timeseries*, což je dvousloupcová matice. Jeden ze sloupců obsahuje hodnoty požadované veličiny a druhý sloupec čas, kdy se má tato hodnota nastavit. Hodnoty veličin mezi těmito časy přepnutí je také možno interpolovat. Pro zjednodušení budu zatím uvažovat konstantní rychlost přívěsu podél trajektorie. Platí tak:

$$v_1(\tau) = konst. \quad (11.1)$$

V této kapitole budu čas značit symbolem  $\tau$ , aby nemohlo dojít k záměně s parametrem Bézierovy křivky  $t$ . Časový interval má rozsah  $\langle 0, \tau_{max} \rangle$ , kde  $\tau_{max}$  je čas potřebný k projetí celé trajektorie. Ze známé délky trajektorie podle vztahu (10.12) lze vypočítat čas  $\tau_{max}$  jako:

$$\tau_{max} = \frac{s}{v_1}. \quad (11.2)$$

Tímto vztahem je dán celkový čas, po který poběží simulace v Simulinku.

Zbývá určit průběh úhlové rychlosti přívěsu  $\omega_1$  v čase podél trajektorie. Tato úhlová rychlost je dána vztahem:

$$\omega_1 = \frac{d\theta_1}{d\tau}. \quad (11.3)$$

Průběh úhlu  $\theta_1$  podél trajektorie odpovídá až na konstantu průběhu úhlu tečny Bézierovy křivky (tato konstanta je rovna  $\pi$ , jelikož přívěs je otočen „po směru couvání“, viz např. Obr. 24). Mezi úhlem tečny ke křivce v určitém bodě a derivací téže křivky ve stejném bodě platí:

$$\tan(\alpha) = \frac{B'_y}{B'_x} \quad (11.4)$$

kde  $B'_x$  a  $B'_y$  jsou derivace Bézierovy křivky podle jejího parametru  $t$  (ne podle času  $\tau$ ) podle rovnice (10.15) a  $\alpha$  je úhel tečny této křivky.

Platí tedy:

$$\theta_1 = \operatorname{atan}\left(\frac{B'_y}{B'_x}\right) + \pi. \quad (11.5)$$

Derivací tohoto vztahu podle času by bylo možné vyjádřit úhlovou rychlost  $\omega_1$ .

Jelikož však budu, jak jsem výše zmiňoval, používat jako vstup do systému data ve formátu *timeseries*, není nutné vyjadřovat analyticky závislost úhlu  $\theta_1$  na čase, stačí získat diskrétní hodnoty úhlu  $\theta_1$  v sadě časových okamžiků. Za tímto účelem diskretizuji Bézierovu křivku  $\mathbf{B}(t)$  (tudíž i trajektorii) jako body  $\mathbf{B}(t_k)$  dané konkrétní volbou parametru  $t_k$ , pro který platí:

$$t_k = k\Delta t, \quad k = 0, 1, 2, \dots, N, \quad \Delta t = \frac{1}{N}, \quad (11.6)$$

kde  $N$  je počet bodů v kterých je trajektorie diskretizována a  $\Delta t$  je krok parametru mezi dvěma body trajektorie.

Úhel  $\theta_1$  z rovnice (11.5) mohu tedy vyjádřit pro konkrétní volbu parametru  $t_k$  jako:

$$\theta_1(t_k) = \operatorname{atan}\left(\frac{B'_y(t_k)}{B'_x(t_k)}\right) + \pi, \quad k = 0, 1, 2, \dots, N. \quad (11.7)$$

Úhlovou rychlost  $\omega_1$  vyjádřím náhradou derivace diferencí pomocí vztahu:

$$\omega_1(t_k) = \frac{\theta_1(t_{k+1}) - \theta_1(t_k)}{\Delta \tau_k}, \quad k = 0, 1, 2, \dots, (N - 1), \quad (11.8)$$

kde  $\omega_1(t_k)$  je úhlová rychlost v bodě trajektorie  $\mathbf{B}(t_k)$  potřebná ke změně úhlu  $\theta_1$  z hodnoty  $\theta_1(t_k)$  na hodnotu  $\theta_1(t_{k+1})$  za čas  $\Delta \tau_k$ .

Čas  $\Delta \tau_k$  je čas, za který projede souprava  $k$ -tý úsek trajektorie konstantní rychlostí  $v_1$ . Trajektorie má celkem  $(N - 1)$  úseků a každý tento úsek má obecně jinou délku, přestože krok mezi dvěma parametry je vždy stejný. Délka  $k$ -tého úseku trajektorie  $s_k$  je dána vztahem:

$$s_k = \int_{t_k}^{t_{k+1}} \sqrt{B'_x(t)^2 + B'_y(t)^2} dt, \quad k = 0, 1, 2, \dots, (N - 1). \quad (11.9)$$

Čas  $\Delta\tau_k$  pro dosažení do rovnice (11.8) lze tedy vyjádřit jako:

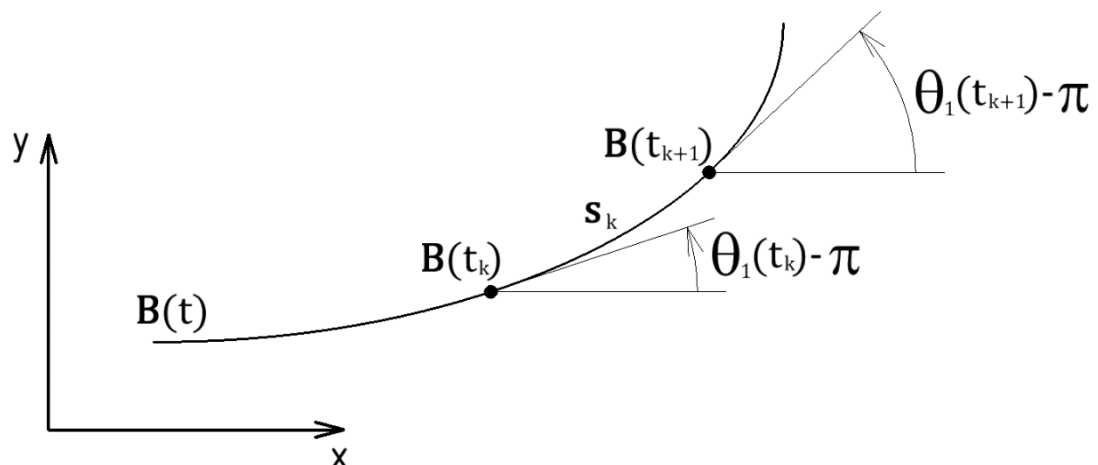
$$\Delta\tau_k = \frac{s_k}{v_1}, \quad k = 0, 1, 2, \dots, (N - 1). \quad (11.10)$$

Jelikož je trajektorie poskládána z jednotlivých úseků, musí také platit vztahy:

$$s = \sum_{k=0}^{N-1} s_k, \quad (11.11)$$

$$\tau_{max} = \sum_{k=0}^{N-1} \Delta\tau_k. \quad (11.12)$$

Grafickou představu výše popsané diskretizace trajektorie představuje Obr. 36:



Obr. 36 – Diskretizace trajektorie

Pro snadnější pochopení uvedu slovní popis výpočtu úhlových rychlostí a časů, které tvoří vstup do simulačního modelu: Na počátku simulace je čas simulace  $\tau = 0$ , nultý úsek trajektorie je dán body  $\mathbf{B}(t_0) = \mathbf{P}_0$  a  $\mathbf{B}(t_1)$ , má délku  $s_0$  a rychlostí  $v_1$  ho přívěs projede za čas  $\Delta\tau_0$ . Za stejný čas se musí změnit úhel natočení přívěsu z hodnoty  $\theta_1(t_0)$  na hodnotu  $\theta_1(t_1)$ , čímž je určena hodnota úhlové rychlosti v bodě  $\mathbf{B}(t_0)$ . Po uplynutí času  $\Delta\tau_0$  se přívěs nachází v bodě  $\mathbf{B}(t_1)$  a jeho úhel natočení je  $\theta_1(t_1)$ . Další úsek je dán body  $\mathbf{B}(t_1)$  a  $\mathbf{B}(t_2)$ , má délku  $s_1$  a přívěs ho projede při stejné konstantní rychlosti  $v_1$  za čas  $\Delta\tau_1$ . Za stejný čas se musí změnit jeho úhel natočení z hodnoty  $\theta_1(t_1)$  na  $\theta_1(t_2)$ , čímž je určena hodnota úhlové rychlosti v bodě  $\mathbf{B}(t_1)$ . Stejným způsobem výpočet pokračuje, dokud není projeta celá trajektorie.

Touto formou jsem tedy připravil vstup do simulace jako datový typ *timeseries*. Strukturu tohoto vstupu znázorňuje následující tabulka:

Index řádku <i>timeseries</i> (odpovídá parametru $t_k$ )	Hodnota $\omega_1$	Čas přepnutí $\tau_k$
0	$\omega_1(0)$	0
1	$\omega_1(\Delta t)$	$\Delta\tau_0$
2	$\omega_1(2\Delta t)$	$\Delta\tau_0 + \Delta\tau_1$
3	$\omega_1(3\Delta t)$	$\Delta\tau_0 + \Delta\tau_1 + \Delta\tau_2$
⋮	⋮	⋮
$i$	$\omega_1(i\Delta t)$	$\sum_{k=0}^i \Delta\tau_k - \Delta\tau_i$
⋮	⋮	⋮
$(N - 1)$	$\omega_1((N - 1)\Delta t)$	$\sum_{k=0}^{N-1} \Delta\tau_k - \Delta\tau_{N-1}$

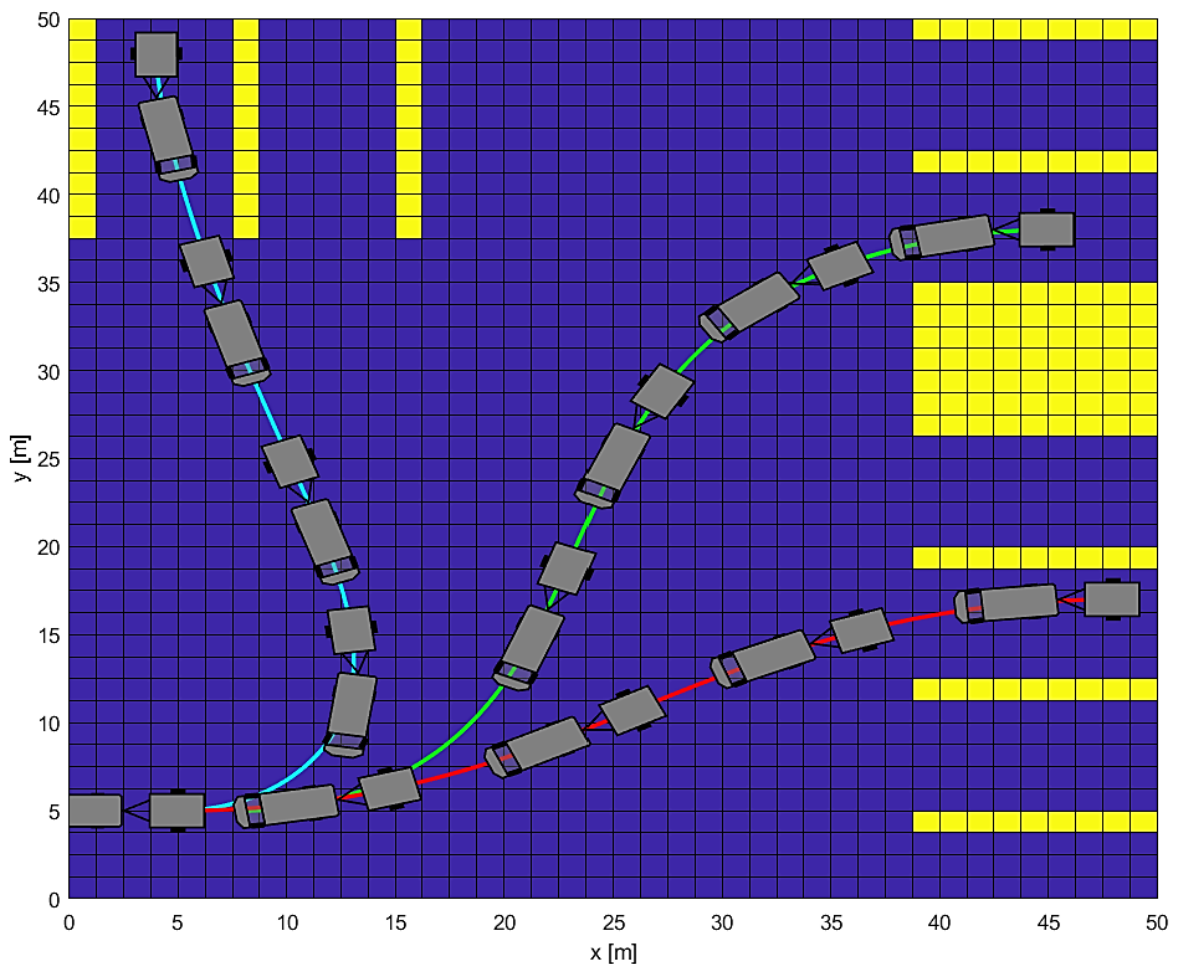
Tabulka 2 – Struktura datového typu *timeseries* jako vstup do simulace



## 12. Ověření navrženého řešení

Ověření funkčnosti navrženého řešení provedu na několika mapách okolního prostředí. Algoritmus vyhledání a optimalizace trajektorie nechám vyhledat trajektorie, z nichž následně připravím podle postupu z minulé kapitoly vstupy pro simulaci systému v Simulinku. Po provedení simulace porovná vypočítanou trajektorii s trajektorií projetou během simulace.

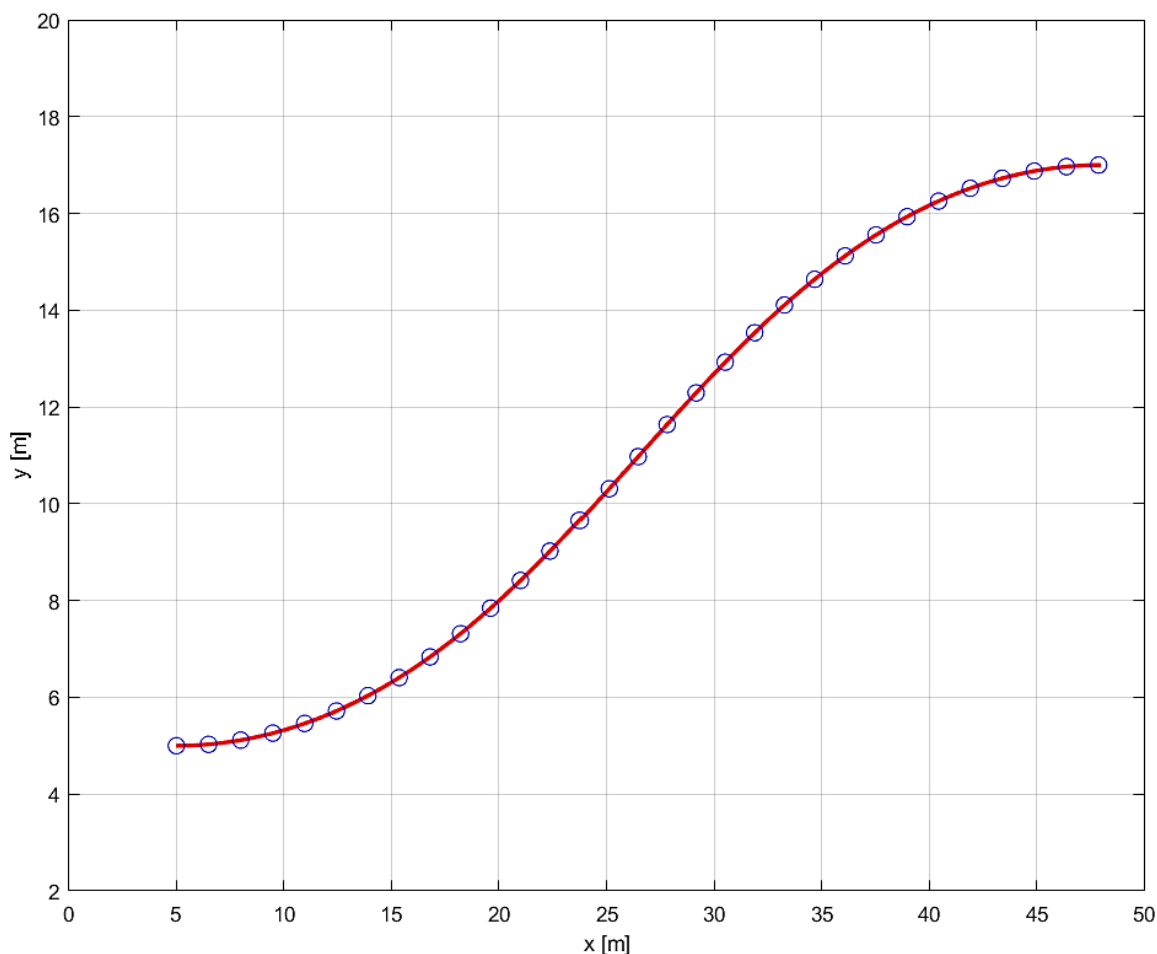
První mapou prostředí bude smyšlené parkoviště dle Obr. 37, pro ukázkou jsem provedl simulace pro několik možných způsobů zaparkování, všechny tři znázorněné trajektorie dokázal systém vypočítat po prvním spuštění při zcela náhodné volbě volných parametrů  $\vec{p}$ . Souprava je pro každou trajektorii vykreslena v několika zvolených bodech průjezdu. Data použitá pro vykreslení soupravy jsou ze simulace, je tedy vidět, že simulovaná souprava dokáže spolehlivě sledovat požadovanou trajektorii.



Obr. 37 – Simulace pohybu soupravy pro několik koncových poloh (mapa parkoviště)

Pro přesnější kontrolu správnosti projeté trajektorie uvedu například pro červenou trajektorii z Obr. 37 graf srovnání trajektorie určené optimalizačním algoritmem s trajektorií, která je dána výstupem ze simulačního modelu – toto srovnání poskytuje Obr.

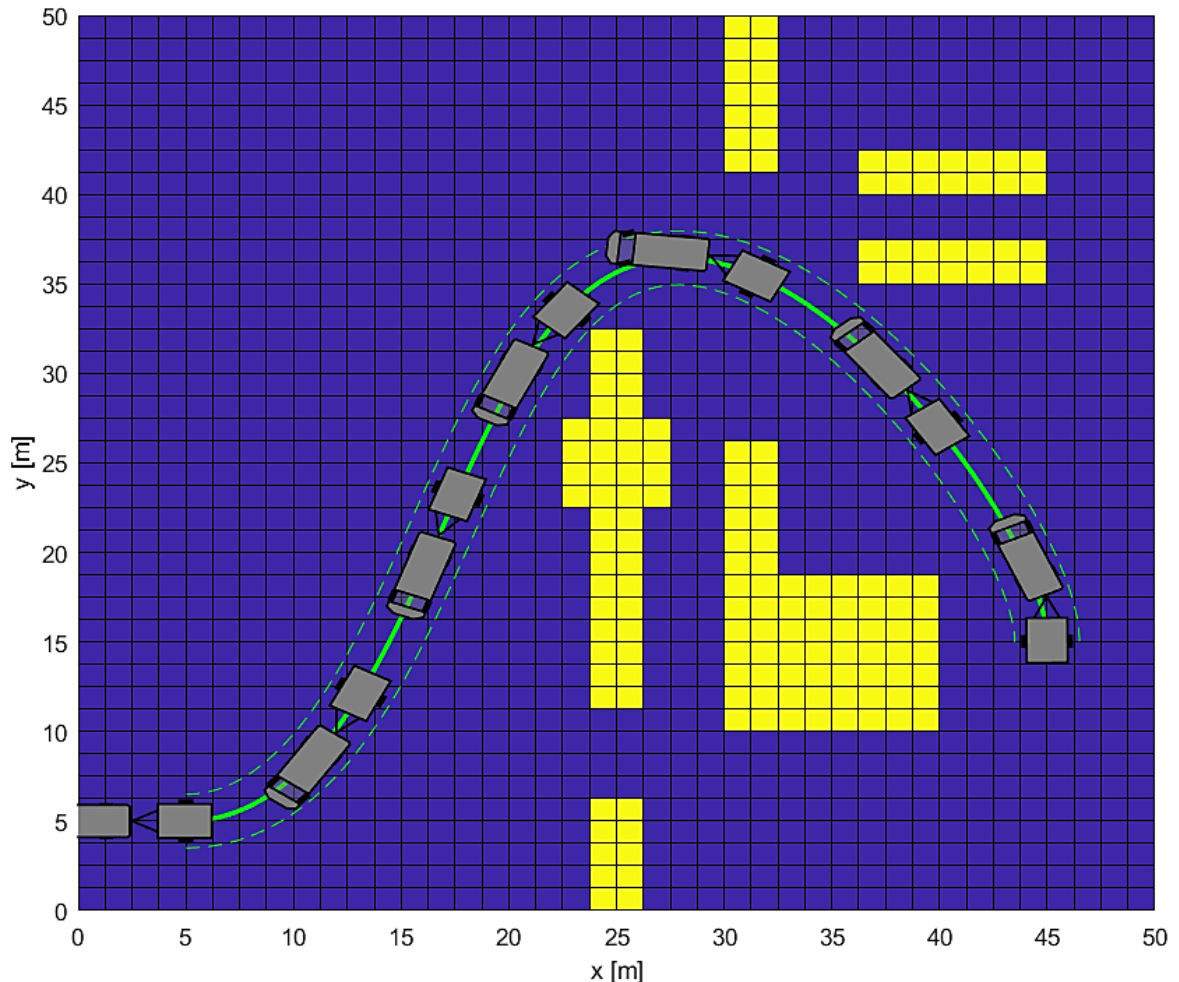
38, kde červená křivka je shodná s červenou trajektorií z Obr. 37 (jiné měřítko os) a modré body jsou vybrané polohy středu nápravy přívěsu během simulace parkovacího manévru.



Obr. 38 – Detailní porovnání přesnosti sledování předepsané trajektorie modelem soustavy

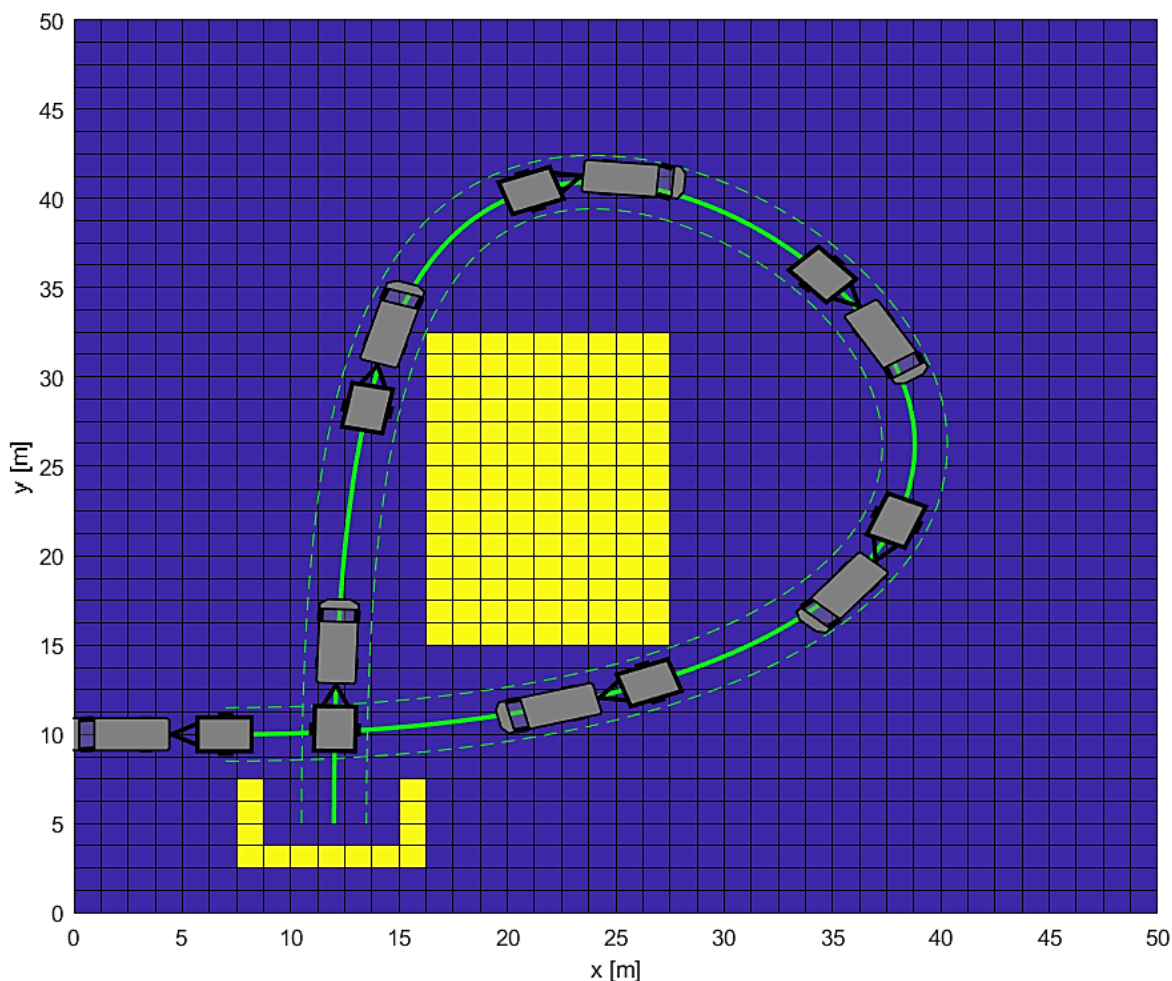
Na Obr. 39 je znázorněna vypočítaná trajektorie (plná zelená) i s bezpečnostní zónou (čárkovaná zelená) na smyšlené mapě, kde se vyskytuje relativně velké množství překážek. Algoritmus dokázal přesto relativně rychle nalézt a optimalizovat průjezdnou trajektorii. Z obrázku je patrné, že při průjezdu ostřejší ze zatáček se automobil dostává svou přední částí mimo bezpečnostní zónu a mohla by tím pádem hrozit kolize s překážkou. Řešením by mohlo být zvětšení bezpečnostní zóny kolem trajektorie, avšak tento přístup by znamenal omezení možnosti nalezení trajektorie v úzkých, avšak průjezdných místech. Lepším přístupem k problému by mohlo být navržení adaptivní bezpečnostní zóny, její velikost by závisela na křivosti trajektorie, v místech s vyšší křivostí by se proporčně zvětšovala za účelem pokrytí většího vybočení automobilu. Ideálním řešením by bylo provést již při vyhledávání trajektorie simulaci průjezdu soupravy. V takovém případě by bylo možné již při plánování trajektorie brát v potaz polohu soupravy v každém okamžiku průjezdu trajektorie. Provádění simulace pohybu podél trajektorie v každém optimalizačním kroku za účelem určení skutečného průjezdu jsem při vytváření mého návrhu asistenta realizoval,

avšak výpočetní doba narostla nepříjemným způsobem. Při případné lepší optimalizaci běhu simulace i plánovacího algoritmu by to však mohla být cesta k nalezení zaručeně průjezdné a bezpečné trajektorie bez zbytečného omezování plánování trajektorie přidavnou bezpečnostní zónou.



Obr. 39 – Mapa prostředí s větším množstvím překážek

Při navrhování systémů, jako je asistent pro couvání s přívěsem je nevyhnutelná práce s goniometrickými funkcemi. Při jejich vyčíslování, či při inverzním použití bývá často problematický definiční obor a obor hodnot, úhel vyčíslený určitou funkcí může například spadat do jiného kvadrantu, než je očekáváno. Jelikož využívám ve svých algoritmech k výpočtům úhlů i funkci arkus tangens, rozhodl jsem se systém testovat při přechodu hodnoty úhlu přes hranice kvadrantů. Toto testování odhalilo chybné určování úhlů ve zvláštních případech trajektorií, například, když trajektorie protínala sama sebe, tudíž úhel natočení těles proběhl více než 180 stupňů. Při počítání běžných tvarů trajektorií se tento problém neprojevil, po jeho odhalení se mi ho podařilo odstranit. Příklad trajektorie, která protíná sama sebe uvádí Obr. 40, je na něm opět vykreslen sled poloh dosažených při simulaci a je zjevné, že se model chová po odstranění chyb správně.



Obr. 40 – Příklad trajektorie, která protíná sama sebe

Ověření funkčnosti navrženého řešení jsem provedl i na jiných mapách, nebudu zde však s ohledem na přehlednost práce uvádět další. Zároveň jsem správnost řešení ověřil i pro systém automobilu s vlečkou, avšak ten není zcela odladěn co do rychlosti výpočtu. Příklad průjezdu se systémem automobil-vlečka uvádí přílohová část práce.

Souhrnně lze konstatovat, že na mapách s nepříliš velkým počtem překážek dokáže asistent velice rychle vyhledat a optimalizovat trajektorii. Na mapách, kde se vyskytuje větší počet překážek je rychlost algoritmu omezena globální smyčkou, kdy nalezení prvotní průjezdné trajektorie může trvat delší dobu. V případě takových map je vhodné využít možnost pro ruční zadání počátečního odhadu dvou bodů podél trajektorie, které může nalezení prvotní průjezdné trajektorie značně urychlit. Sestavený simulační model funguje správně a provedením simulace dojde k projetí žádané trajektorie.

## 13. Návrh podoby asistenta pro skutečné vozidlo

V této kapitole se budu zabývat možnou realizací asistenčního systému pro skutečné vozidlo. Základním předpokladem pro funkci takového asistenčního systému je naplnění potřeby možnosti vyčítat některé potřebné údaje z řídicí jednotky vozidla. Jednalo by se především o rychlost pohybu a natočení volantu. Většina novějších automobilů umožňuje tyto informace vyčítat, pokud by to nebylo možné, bylo by třeba na automobil instalovat dodatečnou výbavu v podobě enkodérů, které by zprostředkovaly tyto informace. Takový postup by však mohl být problematický, jelikož by vyžadoval zásah a možnou úpravu prvků, které zajišťují bezpečnost vozidla a podléhají přísným legislativním požadavkům a normám. Dalším požadavkem nutným pro mnou navržený asistenční systém je možnost vyhodnocovat úhel zalomení mezi automobilem a přívěsem. Toho by bylo možné dosáhnout pomocí speciálního elektronického tažného zařízení pro estimaci úhlu přímo v místě spojení. Jinou možností je nepřímé měření tohoto úhlu, například pomocí měření vzdálenosti přívěsu od automobilu a následným přepočtem na úhel zalomení. V současnosti už se pro vyhodnocení úhlu zalomení využívá také zpracování obrazu. Způsob určování mapy prostředí na základě zpracování obrazu jsem popisoval v práci již dříve, automobil nemusí být nutně vybaven systémem pro rekonstrukci okolního prostředí, ačkoli je tím částečně omezena funkcionality systému.

Nezbytnou součástí asistenčního systému je ovládací prvek, který umožní zadávání příkazů do systému. Ovládání asistenta zahrnuje např. nastavení konstant automobilu, uložení profilů přívěsů či vleček, nastavení velikosti bezpečnostní zóny a další. Hlavní funkcionalitou, kterou musí splňovat, je možnost zobrazení mapy prostředí, ve které se automobil nachází a zadání požadavku na cílovou polohu přívěsu a úhel natočení přívěsu v cílové poloze. Ovládacím prvkem by tak v dnešní době nejspíše byla dotyková obrazovka, kterou by uživatel instaloval do kabiny automobilu. Alternativním ovládacím prvkem by mohl být i mobilní telefon s nainstalovanou aplikací parkovacího asistenta, komunikace se systémem by probíhala bezdrátově.

Mnou navržený asistenční systém pro jízdu s přívěsem by mohl fungovat i při zcela autonomní jízdě vozidla, jelikož systém má na rozdíl od asistentů zmíněných v kapitole 4 informace o trajektorii určené k průjezdu. V takovém případě by řidič pouze zvolil cílový stav soupravy a automobil by vykonal celý parkovací manévr samostatně. Asistent by mohl být užíván i v „semiautonomním“ režimu, kdy by řidič pouze zrychloval či zpomaloval automobil a točení volantem by na základě pokynů z asistenta prováděl elektromechanický posilovač řízení. Takovýto způsob užití se zdá v dnešní době nejrealizovatelnější, jelikož veškeré náročné polohování soupravy zajišťuje počítač, řidič pouze přizpůsobuje rychlost průjezdu. To dává zároveň řidiči dostatečnou kontrolu nad systémem a může manévr v případě nebezpečí okamžitě ukončit. V další části této kapitoly budu uvažovat právě takovýto způsob realizace. V nejnižším stupni autonomnosti systému by i úkon točení volantem prováděl sám řidič. Na zobrazovacím zařízení (ať už speciální dotyková obrazovka,

či mobil) by se zobrazoval aktuální stav vozidla v prostoru a pokyny pro řidiče, které ho provedou parkovacím manévrem.

Ve své práci jsem zatím ověřil správnost vytvořeného simulačního modelu při konstantní rychlosti přívěsu. Při simulaci jsem použil inverzní kinematický model, výstupy z něj jsou podle Obr. 13 údaje o požadované rychlosti  $v_0$  a úhlové rychlosti  $\omega_0$  automobilu v čase. Hodnotu požadované úhlové rychlosti lze pomocí vztahu (7.20) přepočítat na požadované natočení kol:

$$\phi = \operatorname{atan}\left(\frac{\omega_0 L}{v_0}\right). \quad (13.1)$$

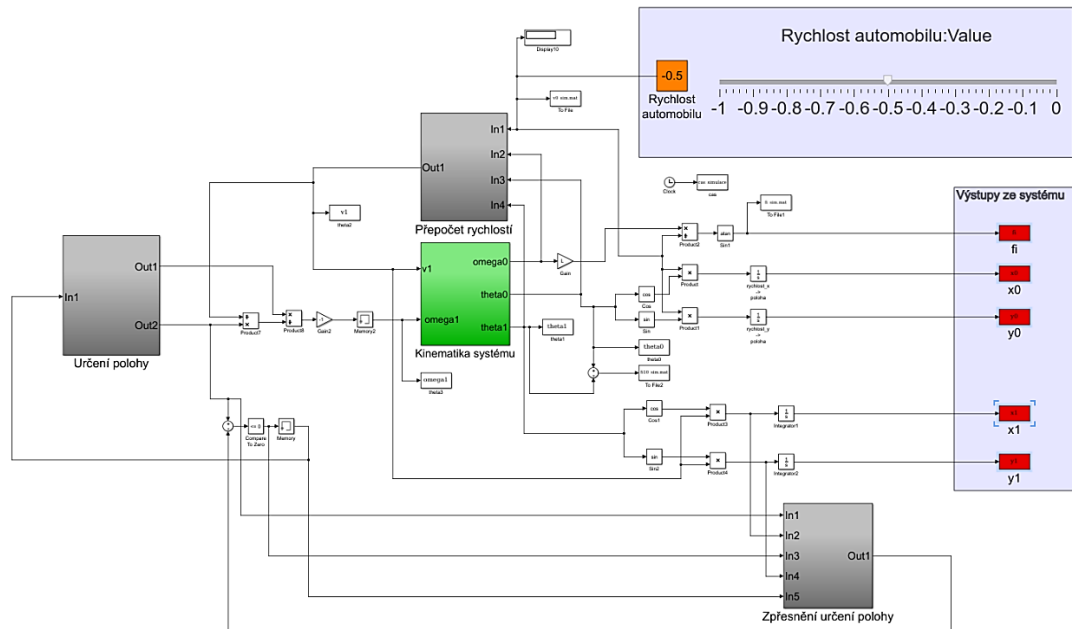
A to následně přes převodový poměr v převodce řízení na natočení volantu:

$$\phi_v = p_{\ddot{r}} \phi. \quad (13.2)$$

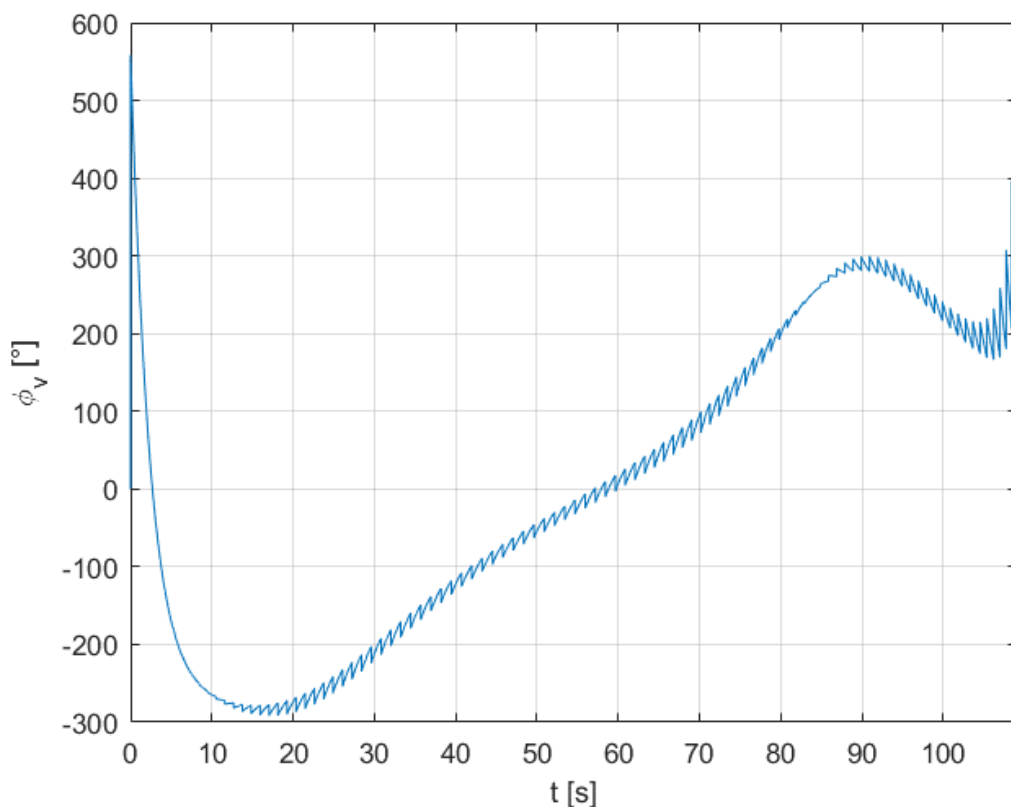
Pokud bude tedy systém polohovat volant tak, aby jeho natočení bylo v ideálním případě v každém místě trajektorie rovno hodnotě  $\phi_v$  podle rovnice (13.2), tak přívěs projede při konstantní rychlosti předepsanou trajektorií.

Podmínka konstantní rychlosti průjezdu je pro skutečný systém nesplnitelná už z požadavku klidové polohy při vyhledávání trajektorie. Pro případnou aplikaci asistenčního systému pro skutečné vozidlo je třeba okamžitou rychlost pohybu automobilu vyčítat z řídicí jednotky. Na základě této rychlosti je třeba změnit časový průběh natočení volantu za účelem správného sledování trajektorie. Za účelem vytvoření reálnějšího simulačního modelu jsem tedy upravil model z Obr. 14. Upravený model vychází z kombinace rovnic (7.34) a (7.35). V upraveném modelu podle Obr. 41 je možné za běhu simulace měnit libovolně rychlost automobilu, protože tuto rychlost podle návrhu „semiautonomní“ verze systému ovládá libovolně řidič. Na základě této rychlosti se vyhodnocuje ujetá vzdálenost, z níž se následně získá upravený průběh natočení volantu. Upravený průběh natočení volantu má co do tvaru shodný průběh s vypočítaným natočením volantu při předpokladu konstantní rychlosti. Liší se však v nelineárně naškálované časové ose. Celkový čas trvání simulace taky není předem znám a její ukončení se provádí vyhodnocením, zda došlo k průjezdu celé trajektorie. V praxi by tedy řidič do asistenta zadal požadavek na nalezení trajektorie, systém by po nalezení a optimalizaci trajektorie předpočítal průběh natočení volantu podél trajektorie a spustil simulaci. V simulačním modelu by na základě údaje o rychlosti automobilu systém počítal aktuální úsek trajektorie, na kterém se souprava nachází a přiřazoval by příslušné natočení volantu. Jinak řečeno, systém musí přepočítat časově závislé natočení volantu ze zjednodušené simulace na natočení volantu jako funkci aktuální polohy v rámci trajektorie. Až by se souprava blížila do cílové polohy, řidič by byl upozorněn, aby začal snižovat rychlost a po příjezdu do cíle, aby zcela zastavil.

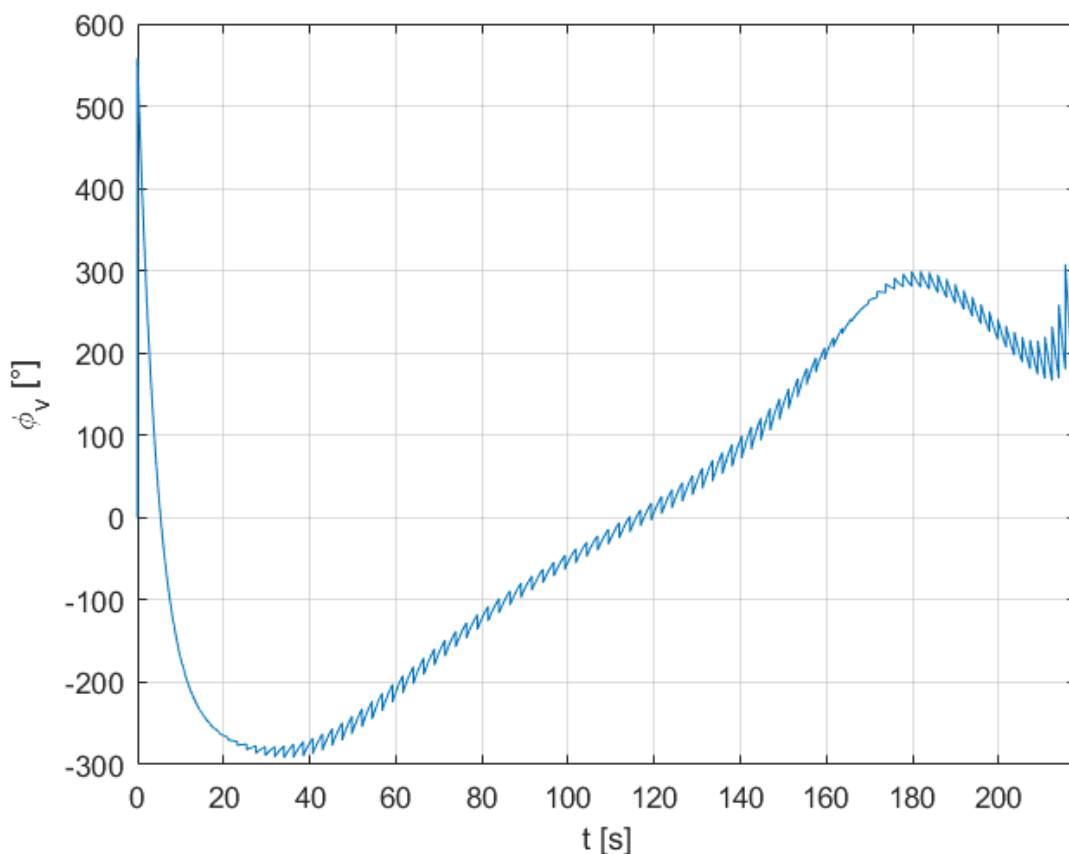
Na Obr. 42 je znázorněn průběh natočení volantu v čase při konstantní rychlosti automobilu  $v_0 = -0,5 \text{ ms}^{-1}$ . Převodový poměr v převodce řízení byl zvolen jako  $p_{\text{ř}} = 24$ . Na Obr. 43 je znázorněn průběh natočení volantu pro stejnou trajektorii a stejnou hodnotu  $p_{\text{ř}}$ , avšak s konstantní hodnotou rychlosti  $v_0 = -0,25 \text{ ms}^{-1}$ . Z časové osy je patrné, že došlo pouze k lineárnímu přeškálování, čas průjezdu se zdvojnásobil.



Obr. 41 – Reálnější simulační model

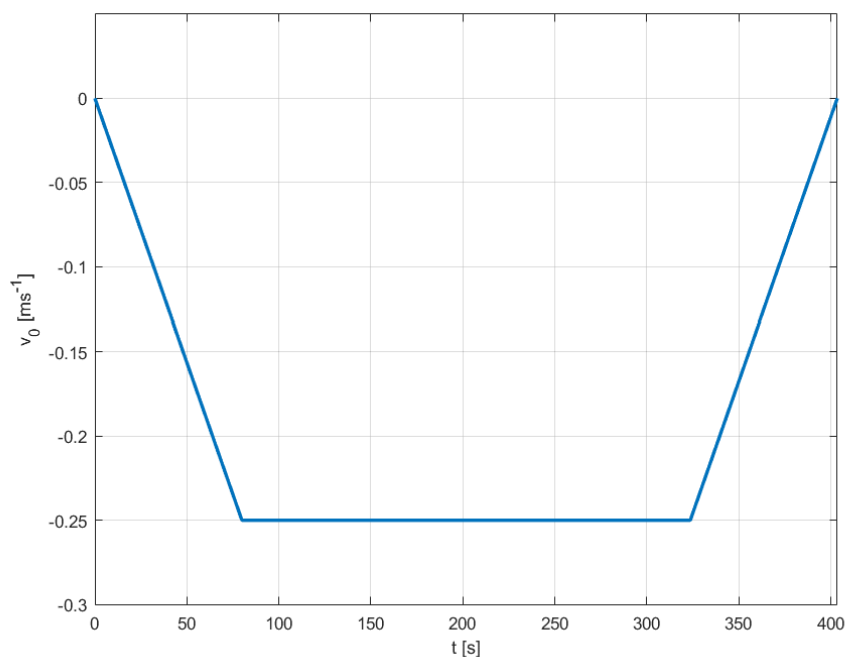


Obr. 42 – Průběh natočení volantu při konstantní rychlosti  $v_0 = -0,5 \text{ ms}^{-1}$



Obr. 43 – Průběh natočení volantu při konstantní rychlosti  $v_0 = -0,25 \text{ ms}^{-1}$

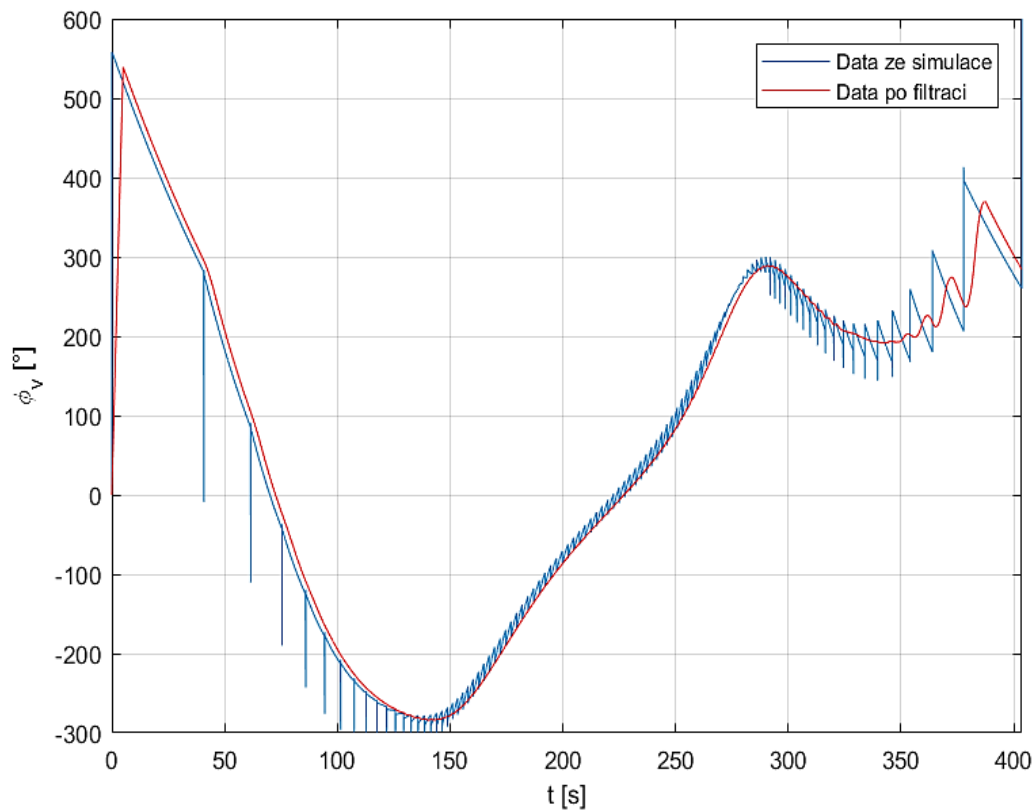
Dále provedu průjezd této trajektorie, který bude simulovat případ skutečného užití asistenčního systému. Na počátku bude mít automobil nulovou rychlost, následně se uvede do pohybu, systém bude řídit polohu volantu, aby souprava jela po trajektorii, a nakonec automobil zastaví. Profil rychlosti  $v_0$  znázorňuje Obr. 44.



Obr. 44 – Profil rychlosti během průjezdu trajektorie



Na Obr. 45 je modře znázorněn průběh natočení volantu v čase při průjezdu trajektorií rychlostí podle Obr. 44. Je vidět, že průběh je stejný, jako při průjezdech konstantní rychlostí, avšak tentokrát je časová osa škálována nelineárně.



Obr. 45 – Průběh natočení volantu při nekonstantní rychlosti průjezdu

Velké překmity v jinak zřetelném trendu vývoje natočení volantu v čase jsou způsobeny nedokonalostmi modelu ve spojení s numerickými nepřesnostmi, pro případné užití průběhu natočení volantu jako vstupu do systému by bylo třeba tento průběh vhodně filtrovat. Filtrovaný průběh natočení volantu v čase je na Obr. 45 zobrazen červeně. Nezabýval jsem se výběrem nejlepšího filtru pro danou aplikaci, filtrovaný průběh dat je pouze ilustrativní.

Průběh rychlosti  $v_0$  z Obr. 44 spolu s průběhem natočení volantu  $\phi_v$  z Obr. 45 tvoří všechny skutečné fyzikální vstupy, které má modelovaná souprava. Nyní ji lze simulovat přímo pomocí dopředné kinematiky dle Obr. 13. Sestavil jsem simulační model, kde vstupy tvoří pouze rychlost  $v_0$  a poloha volantu  $\phi_v$ . V průběhu simulace přibližně v polovině projeté trajektorie ztratí systém vlivem numerických nepřesností stabilitu a úhel zalomení přívěsu přesáhne povolenou hodnotu. Pro silně nestabilní systém, jako je automobil s přívěsem je pro účely řízení nezbytný návrh regulátoru, který bude stabilizovat úhel zalomení přívěsu.

### 14. Závěr

Cílem této diplomové práce bylo vytvořit asistenta pro nalezení optimální trajektorie při couvání soupravy automobilu s přívěsem.

V úvodních kapitolách této práce jsem uvedl přehled existujících řešení asistenčních systémů pro couvání s přívěsem a popsal jsem principy jejich funkcí. V další části rešeršní studie uvádím poznatky, které jsem nabył při studiu zdrojů souvisejících s danou problematikou, jedná se o přístup k sestavování matematického modelu pro soupravu automobilu s přívěsem, různé návrhy strategií řízení takové soupravy a přístupy k vyhledávání trajektorií.

Na základě poznatků z rešeršní studie jsem dále vytvořil kinematický model automobilu s přívěsem a implementoval ho do prostředí Matlab Simulink za účelem pozdější simulace systému. Nad rámec zadání jsem dále sestavil kinematický model pro soupravu automobilu s vlečkou, jakožto dalším častým zařízením užívaným k přepravě nákladu. Kinematický model automobilu s vlečkou jsem rovněž implementoval do prostředí Matlab Simulink.

Hlavní částí mé práce bylo vytvoření algoritmu pro nalezení a následnou optimalizaci trajektorie za předpokladu znalosti situace za soupravou. K řešení tohoto problému jsem přistoupil s využitím Bézierových křivek, jelikož mají vhodné vlastnosti pro popis trajektorií. Navrhl jsem vlastní algoritmus, který dokáže ve známé mapě okolního prostředí navrhnout průjezdnou trajektorii ve formě Bézierovy křivky a tuto trajektorii následně optimalizuje tak, aby byla průjezdná pro soupravu automobilu s přívěsem. Optimalizační algoritmus tvaruje trajektorii tak, aby byla co nejkratší a měla rovnoměrný průběh křivosti, zároveň omezuje maximální křivost trajektorie kinematickými možnostmi soupravy. Při procesu optimalizace je také zaručena bezkoliznost trajektorie a bezpečná vzdálenost od překážek, které souprava podél trajektorie míjí.

Mnou vytvořený algoritmus pro vyhledání a optimalizaci trajektorie jsem ověřil na řadě smyšlených map okolního prostředí, ve kterých jsem prováděl parkovací manévry. Během testování algoritmu jsem objevil některé jeho nedostatky a ty úspěšně odstranil. Algoritmus dokázal téměř ve všech zkoušených případech užití vyhledat dostatečně rychle průjezdnou trajektorii. Z navržených trajektorií jsem dále pomocí inverzní kinematiky připravil vstupy do dříve vytvořených simulačních modelů. Provedením simulací jsem ověřil správnost a funkčnost navržených modelů. Během simulace dokázala souprava přesně sledovat vypočítanou trajektorii.

V závěrečných kapitolách práce je uveden návrh asistenčního systému pro skutečné vozidlo. Zmiňuji požadavky na vozidlo, které jsou nutné pro možnost správné funkce asistenčního systému a uvádím možné způsoby realizace asistenta, které se liší v úrovni autonomnosti celého systému. Dále jsem provedl úpravu simulačního modelu pro soupravu automobilu s přívěsem, aby byl použitelný pro případné užití v reálném systému.

Model není zdaleka dokonalý, avšak umožňuje relativně realisticky pojatý průjezd vypočítané trajektorie, kdy se na základě aktuální rychlosti vozidla upravuje poloha volantu tak, aby se celá souprava držela předepsané trajektorie.

Při případné budoucí práci na navrženém asistenčním systému by byla hlavní výzvou hardwarová implementace do skutečné soupravy automobilu s přívěsem, jako první krok by postačoval zmenšený model automobilu. Po testování systému na skutečném modelu by bylo třeba zhodnotit dostatečnou detailnost modelu, případně ho doplnit dalšími vztahy, které více přiblíží model realitě. Algoritmus pro plánování trajektorie by měl být použitelný ve stavu popsaném v této práci, možné úpravy by mohly zasloužit váhové koeficienty použité při jednotlivých cílových funkcích. Použití speciálního optimalizačního algoritmu pro řešení minimalizace mnou navržené cílové funkce by také mohlo přinést zkrácení výpočetní doby nutné pro nalezení optimalizované trajektorie. Také by bylo nutné navrhnout zpětnovazební regulátor pro řízení vstupů do systému a případnou korekci chyb, které nedokáže systém modelovat.

## 15. Seznam obrázků a tabulek

Obr. 1 – Infotainment automobilů Škoda (fdrive.cz).....	17
Obr. 2 – Ukázka funkce systému Trailer Assist (převzato z [3]) .....	20
Obr. 3 – Šachovnicová nálepka umístěná na oji přívěsu (převzato z [5]).....	21
Obr. 4 – Vzdálenost mezi zadní parkovací kamerou a nálepkou (převzato z [5]) .....	21
Obr. 5 – Ukázka funkce asistenta pro stabilizaci přívěsu (převzato z [13]).....	22
Obr. 6 – Zjednodušení modelu automobilu na jednostopý model .....	27
Obr. 7 – Odvození pohybových rovnic automobilu .....	28
Obr. 8 – Poloha středu otáčení automobilu v závislosti na natočení kola přední nápravy .....	31
Obr. 9 – Grafické znázornění rychlostí automobilu a přívěsu (vytvoreno podle [22]).....	32
Obr. 10 – Model vlečky Fliegl (www.darky-berk.cz).....	34
Obr. 11 – Grafické znázornění rychlostí přední (1) a zadní (2) nápravy vlečky .....	35
Obr. 12 – Soustava automobilu s přívěsem v globálním souřadnicovém systému .....	37
Obr. 13 – Schéma propagace veličin v dopředné a inverzní kinematice .....	38
Obr. 14 – Model soupravy automobilu s přívěsem v Simulinku .....	40
Obr. 15 – Model soupravy automobilu s vlečkou v Simulinku.....	40
Obr. 16 – Reálné prostředí pro úkol parkování (istockphotos.com).....	41
Obr. 17 – Ustavení parkoviště do prostoru .....	42
Obr. 18 – Konečná podoba mapy prostředí .....	42
Obr. 19 – Příklad Bézierovy kubiky.....	45
Obr. 20 – Orientace soupravy na počátku manévru .....	46
Obr. 21 – Možnosti tvarování Bézierovy kubiky s dvěma volnými parametry .....	48
Obr. 22 – Geometrická interpretace spojitosti první derivace v bodě napojení.....	50
Obr. 23 – Ukázka různých řádů spojitosti derivace křivek v místě jejich napojení.....	51
Obr. 24 – Omezení polohy bodu $P_1$ vůči $P_0$ .....	52
Obr. 25 – Schéma nejvyšší úrovně algoritmu parkovacího asistenta.....	55

Obr. 26 – Ukázka kódu – globální smyčka hledání trajektorie.....	57
Obr. 27 – Ilustrace bezpečnostní zóny kolem soupravy .....	58
Obr. 28 – Ukázka neoptimalizovaných trajektorií.....	59
Obr. 29 – Minimalizace délky trajektorie (bez překážek).....	62
Obr. 30 – Minimalizace délky trajektorie (s překážkou).....	62
Obr. 31 – Optimalizace trajektorie bez zohlednění křivosti .....	65
Obr. 32 – Optimalizace trajektorie se zohledněním křivosti .....	65
Obr. 33 – Graf průběhů křivostí s a bez optimalizace křivosti.....	66
Obr. 34 – Optimalizace trajektorie s aktivní kontrolou kolize.....	67
Obr. 35 – Optimalizace trajektorie bez kontroly kolize .....	67
Obr. 36 – Diskretizace trajektorie .....	71
Obr. 37 – Simulace pohybu soupravy pro několik koncových poloh (mapa parkoviště) .....	73
Obr. 38 – Detailní porovnání přesnosti sledování předepsané trajektorie modelem soustavy.....	74
Obr. 39 – Mapa prostředí s větším množstvím překážek.....	75
Obr. 40 – Příklad trajektorie, která protíná sama sebe.....	76
Obr. 41 – Reálnější simulační model .....	79
Obr. 42 – Průběh natočení volantu při konstantní rychlosti $v_0 = -0,5 \text{ ms}^{-1}$ .....	79
Obr. 43 – Průběh natočení volantu při konstantní rychlosti $v_0 = -0,25 \text{ ms}^{-1}$ .....	80
Obr. 44 – Profil rychlosti během průjezdu trajektorie .....	80
Obr. 45 – Průběh natočení volantu při nekonstantní rychlosti průjezdu.....	81
Tabulka 1 – Souhrn zvolených váhových koeficientů v jednotlivých cílových funkcích .....	68
Tabulka 2 – Struktura datového typu timeseries jako vstup do simulace.....	72

## 16. Použité zdroje a literatura

- [1] "Trailer Assist": Jednoduché, rychlé a bezpečné manévrování se soupravou. *Volkswagen Česká republika* [online]. 2018 [cit. 2018-08-13]. Dostupné z: <https://www.volkswagen.cz/svet-volkswagen/novinky/5814-trailer-assist-jednoduche-rychle-a-bezpecne-manevrovani-se-s>
- [2] Practical Caravan tests Volkswagen's Trailer Assist. *YouTube* [online]. 2018 [cit. 2018-08-13]. Dostupné z: <https://www.youtube.com/watch?v=3ruB5XYObtQ>
- [3] NEW VW Passat | Trailer Assist. *YouTube* [online]. 2018 [cit. 2018-08-22]. Dostupné z: <https://www.youtube.com/watch?v=AFjOjHtG6Oc>
- [4] Trailer Angle Detection. *MAGNA* [online]. 2018 [cit. 2018-08-13]. Dostupné z: <https://www.magna.com/products/power-vision/product/trailer-angle-detection>
- [5] Setting Up Trailer Reverse Guidance | Ford How-To | Ford. *YouTube* [online]. 2018 [cit. 2018-08-22]. Dostupné z: <https://www.youtube.com/watch?v=M0Y7QXobCD8>
- [6] What is Pro Trailer Backup Assist™?. *Ford.com* [online]. 2018 [cit. 2018-08-22]. Dostupné z: <https://owner.ford.com/support/how-tos/safety/driver-assist-technology/trailers-towing-loading/expedition-pro-trailer-backup-assist.html>
- [7] Trailer reverse guidance. *Ford.com* [online]. 2018 [cit. 2018-08-22]. Dostupné z: <http://trailerreverseguidance.com/>
- [8] Volkswagen Group. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2018 [cit. 2018-08-15]. Dostupné z: [https://cs.wikipedia.org/wiki/Volkswagen\\_Group](https://cs.wikipedia.org/wiki/Volkswagen_Group)

- [9] Does it work? Land Rover's Advanced Tow Assist tested. *CAR Magazine* [online]. Německo: Bauer Consumer Media, 2018 [cit. 2018-08-22]. Dostupné z: <https://www.carmagazine.co.uk/car-news/tech/does-it-work-land-rovers-advanced-tow-assist-tested/>
- [10] LAND ROVER REVEAL CONCEPT TRANSPARENT TRAILER TECHNOLOGY. *Land Rover* [online]. 2018 [cit. 2018-08-22]. Dostupné z: <https://www.landrover.com/experiences/news/transparent-trailer.html>
- [11] KARNOPP, Dean. *Vehicle, dynamics, stability and control*. 2nd ed. Boca Raton: CRC Press, c2013. ISBN 978-146-6560-857.
- [12] Trailer Stability Assist. *Continental Automotive* [online]. 2017 [cit. 2018-08-22]. Dostupné z: <https://www.continental-automotive.com/en-gl/Passenger-Cars/Chassis-Safety/Software-Functions/Active-Safety/Trailer-Stability-Assist>
- [13] Trailer Reverse Assist. *Continental Automotive* [online]. 2017 [cit. 2018-08-15]. Dostupné z: <https://www.continental-automotive.com/en-gl/Passenger-Cars/Chassis-Safety/Software-Functions/Maneuvering/Trailer-Reverse-Assist>
- [14] SNIDER, Jarrod M. *Automatic Steering Methods for Autonomous Automobile Path Tracking*. Pittsburgh, PA: Carnegie Mellon University, 2009. CMU-RI-TR-09-08.
- [15] LAVALLE, Steven Michael. *Planning algorithms*. New York: Cambridge University Press, 2006. ISBN 978-0521862059.
- [16] NURBS. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2018 [cit. 2018-08-22]. Dostupné z: <https://cs.wikipedia.org/wiki/NURBS>

- [17] CHOI, Ji-wung, Renwick CURRY a Gabriel ELKAIM. Path Planning Based on Bézier Curve for Autonomous Ground Vehicles. In: *Advances in Electrical and Electronics Engineering – IAENG Special Edition of the World Congress on Engineering and Computer Science 2008*. IEEE, 2008, 2008, s. 158-166. DOI: 10.1109/WCECS.2008.27. ISBN 978-0-7695-3555-5. Dostupné také z: <http://ieeexplore.ieee.org/document/5233184/>
- [18] ZHOU, F., B. SONG a G. TIAN. Bezier curve based smooth pathplanning for mobile robot. *Journal of Information and Computational Science*. 2011, **8**, 2441-2450. ISSN 1548-7741.
- [19] PRADALIER, Cedric a Kane USHER. A simple and efficient control scheme to reverse a tractor-trailer system on a trajectory. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation* [online]. IEEE, 2007, 2007, s. 2208-2214 [cit. 2018-08-22]. DOI: 10.1109/ROBOT.2007.363648. ISBN 1-4244-0602-1. ISSN 1050-4729. Dostupné z: <http://ieeexplore.ieee.org/document/4209412/>
- [20] KEYMASI KHALAJI, Ali a S Ali A MOOSAVIAN. Adaptive sliding mode control of a wheeled mobile robot towing a trailer. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* [online]. 2014, **229**(2), 169-183 [cit. 2018-08-22]. DOI: 10.1177/0959651814550539. ISSN 0959-6518. Dostupné z: <http://journals.sagepub.com/doi/10.1177/0959651814550539>
- [21] Ackermann steering geometry. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2018 [cit. 2018-08-22]. Dostupné z: [https://en.wikipedia.org/wiki/Ackermann\\_steering\\_geometry](https://en.wikipedia.org/wiki/Ackermann_steering_geometry)
- [22] BAZJUK, Ludvík. *Couvání vozidla s přívěsem* [online]. Praha, 2014. Bakalářská práce. České vysoké učení technické v Praze.



- [23] VRBSKÝ, Ladislav. *Experimentální model vozidla s asistentem pro couvání s přívěsem*. Praha, 2016. Bakalářská práce. České vysoké učení technické v Praze.
- [24] VACEK, Patrik. *Zpracování obrazu pro systém couvacího asistenta v automobilu*. Praha, 2018. Diplomová práce. České vysoké učení technické v Praze.
- [25] Bézier curve. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2018 [cit. 2018-08-22]. Dostupné z: [https://en.wikipedia.org/wiki/Bézier\\_curve](https://en.wikipedia.org/wiki/Bézier_curve)
- [26] LINKEOVÁ, Ivana. *Základy počítačového modelování křivek a ploch*. V Praze: České vysoké učení technické, 2008. ISBN 978-80-01-04011-9.
- [27] Fminsearch. *MathWorks* [online]. 2018 [cit. 2018-08-13]. Dostupné z: <https://www.mathworks.com/help/matlab/ref/fminsearch.html>
- [28] LEDERER, Pavel. *Teorie a optimalizace mechanických systémů I*. Praha: České vysoké učení technické, 1988.
- [29] NEUSTUPA, Jiří. *Matematika II*. 3. vyd. V Praze: České vysoké učení technické, 2015. ISBN 978-80-0105-684-4.
- [30] Curvature. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2018 [cit. 2018-08-13]. Dostupné z: <https://en.wikipedia.org/wiki/Curvature>

## 17. Seznam příloh

- Příloha 1 – Ukázka parkování s vlečkou č. 1
- Příloha 2 – Ukázka parkování s vlečkou č. 2
- Příloha 3 – Obsah přiloženého CD

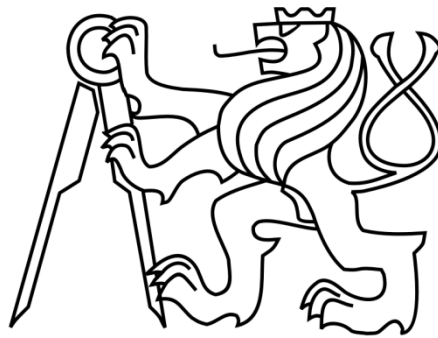
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

---

FAKULTA STROJNÍ

ÚSTAV MECHANIKY, BIOMECHANIKY A MECHATRONIKY

ODBOR MECHANIKY A MECHATRONIKY



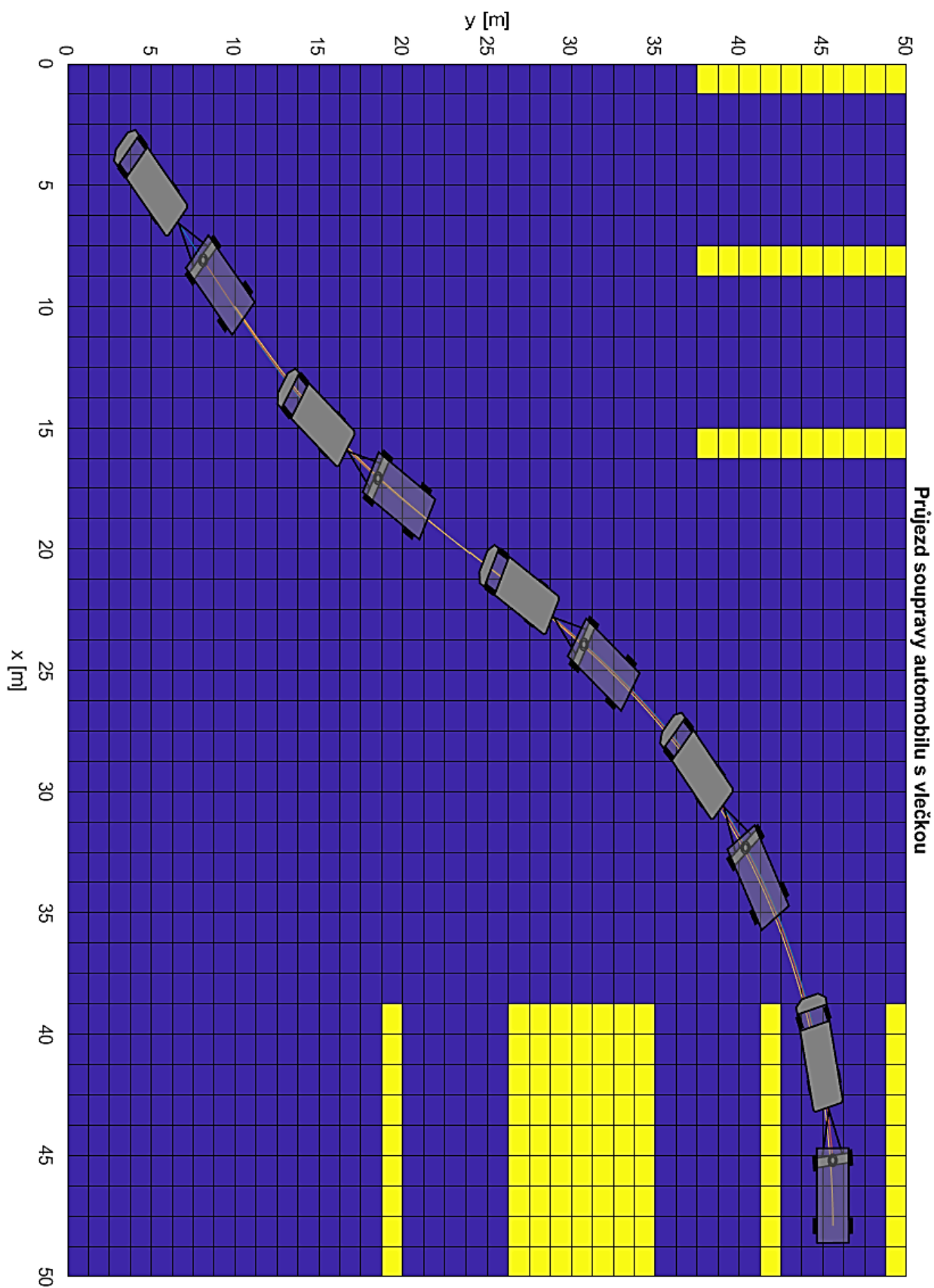
**DIPLOMOVÁ PRÁCE**

**PŘÍLOHOVÁ ČÁST PRÁCE**

**Asistent pro nalezení optimální trajektorie**

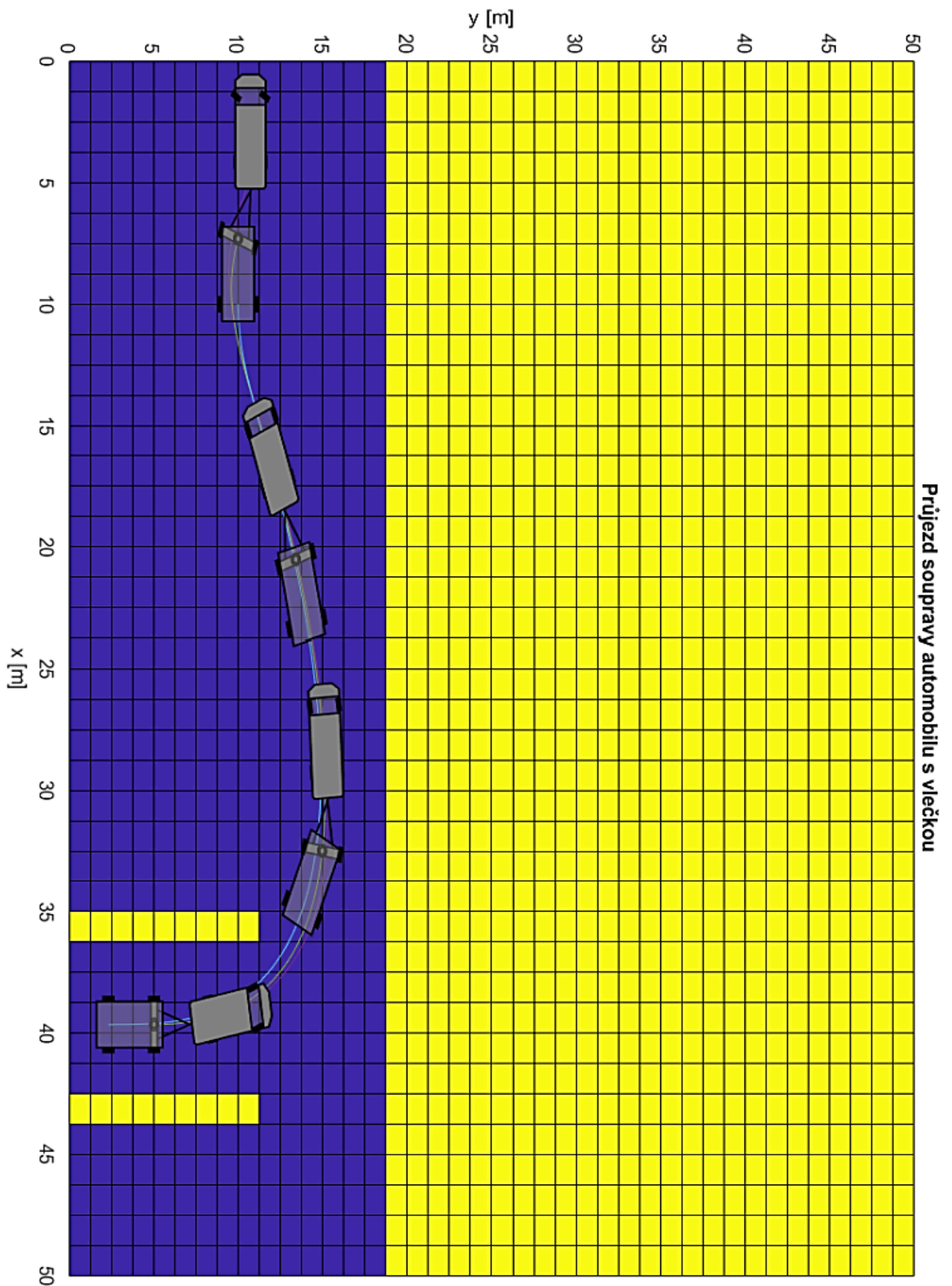
**při couvání s přívěsem**

Příloha 1 – Ukázka parkování s vlečkou č. 1



Průjezd soupravy automobilu s vlečkou

## Příloha 2 – Ukázka parkování s vlečkou č. 2



## Příloha 3 – Obsah přiloženého CD

### Obsah přiloženého CD dle adresářů:

\\Diplomová práce:	Elektronické verze této práce ve formátech <i>.docx</i> a <i>.pdf</i>
\\Model s nekonz. rychl.:	Verze modelu dle kapitoly 13 této práce
\\Parkovací asistent:	Modely inverzní kinematiky přívěsu i vlečky
\\Data_parkovaci_asistent:	Pomocné soubory potřebné pro chod programu
\\Parkovací asistent demo:	Ukázka funkce asistenta (zpracováno podobou „aplikace“)
\\Data_parkovaci_asistent:	Pomocné soubory potřebné pro chod programu

### Poznámky:

Všechny modely jsou založeny na navrženém algoritmu hledání a optimalizace trajektorie, ten je součástí všech složek s Matlab realizací asistenta.

V každé ze tří složek s verzemi asistenta je přiložen soubor *Popis funkce.txt*, který uvádí stručný popis funkce každé ze tří verzí.