

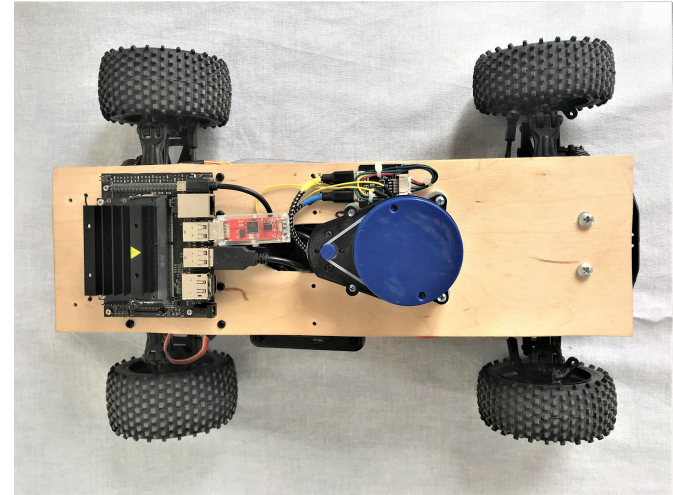
# Trajectory planning for fast moving cars

Šimon Rozsival

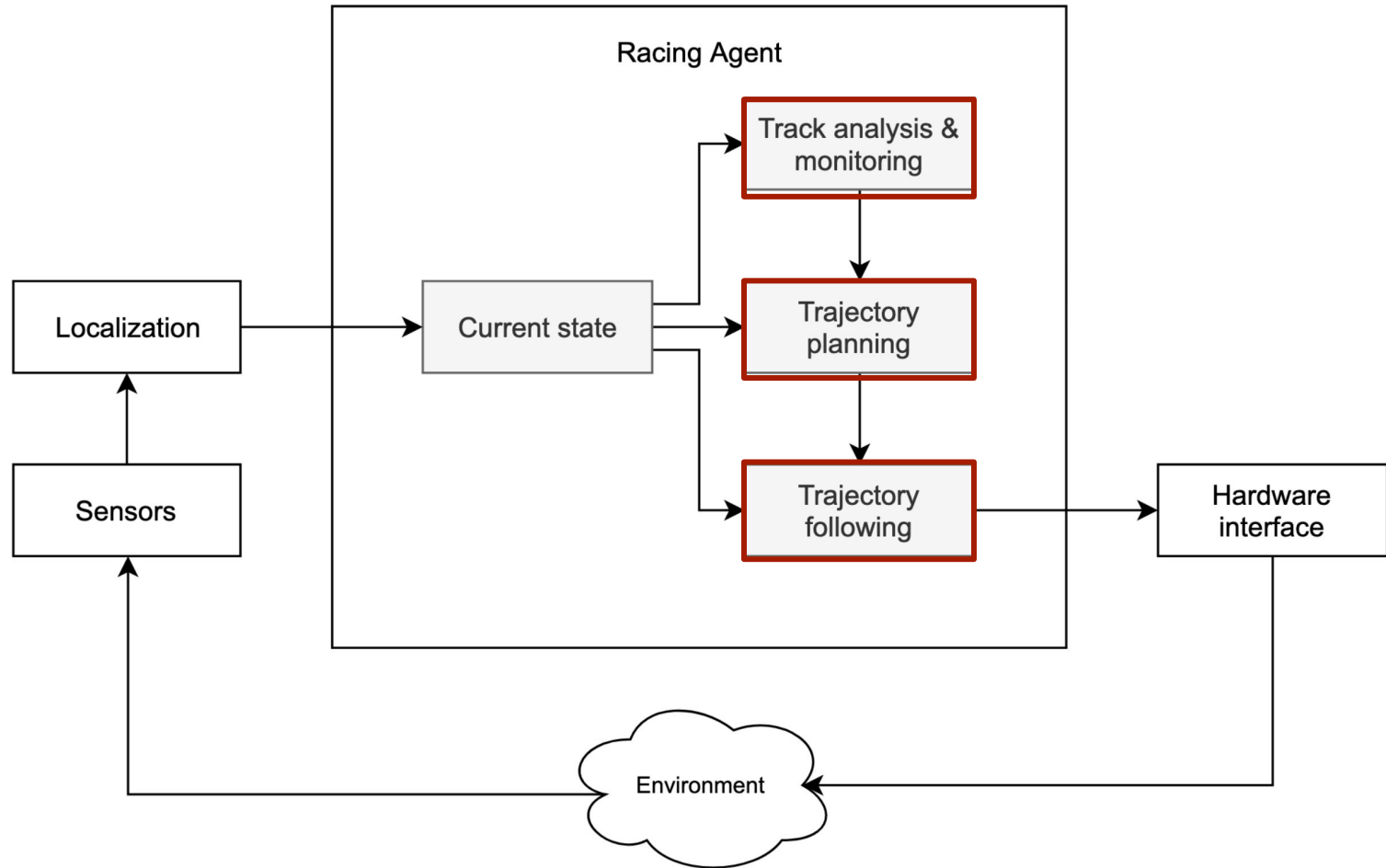
Vedoucí práce: prof. RNDr. Roman Barták, Ph.D.  
Katedra Teoretické Informatiky a Matematické Logiky

# Cíl diplomové práce

- Vytvořit autonomní závodní automobil, který bude schopen závodit na libovolném okruhu
  - Se znalostí mapy okruhu
  - Bez znalosti překážek umístěných na trati
- Otestovat implementaci na skutečném robotickém vozidle
  - Inspirace soutěží F1/10 - UPenn

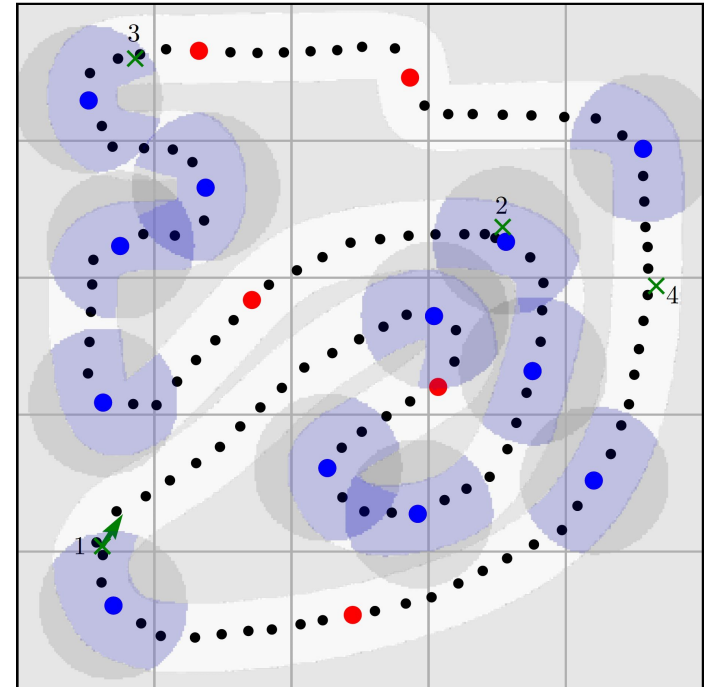


# Autonomní závodní agent



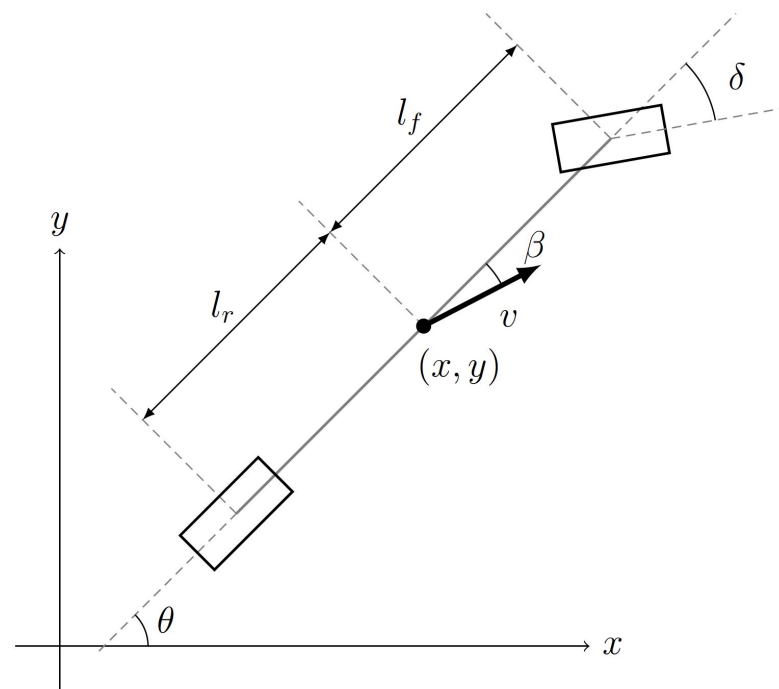
# Analýza mapy okruhu

- *Předpoklad:* Robot nebude schopen perfektně sledovat naplánovanou trasu
  - Po chvíli bude nutné trajektorii přeplánovat
- Plánování soustředěno jen na 2 zatáčky bezprostředně před vozidlem
  - Výrazné zmenšení prohledávaného prostoru
  - Umožňuje real-time plánování za jízdy
- Jednoduchý algoritmus pro detekci zatáček
  - Spuštěn jen jednou před začátkem závodu pro konkrétní okruh
  - Idea:
    - Najdeme “středovou čáru” trati (●) v bitmapě
    - Vynecháme všechny body, které jsou “viditelné” z předchozího validního bodu
    - Sloučíme body, které jsou blízko
    - Vynecháme body, ve kterých se příliš nemění směr cesty (●)
    - Zbývající body prohlásíme za zatáčky (●)



# Plánování trajektorie

- Cíl:
  - najít trajektorii z výchozího stavu skrze 2 následující záchytné body bez kolize
  - co nejlepší čas do cíle
- Spojitý stavový prostor
  - Konfigurace vozidla  $(x, y, \theta)$
  - Stav serva a motoru  $(\delta, \omega)$
  - Počet navštívených záchytných bodů  $(m)$
- Akce: dvojice  $(\delta_t, \tau_t)$
- Testované algoritmy:
  - Hybrid A\*
  - SEHS

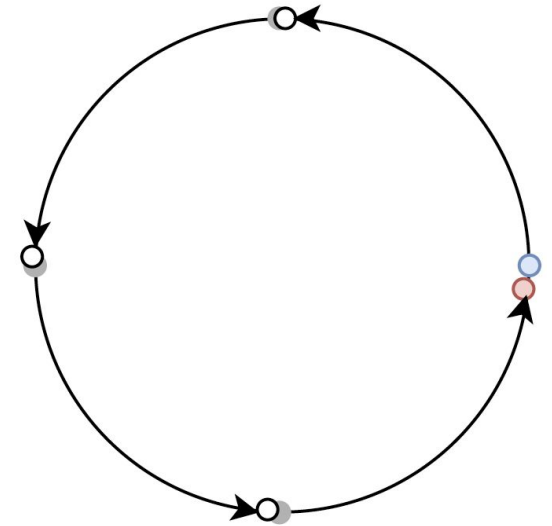


$$X = \{(x, y, \theta, m, \delta, \omega) \mid x, y, \omega \in \mathbb{R}, m \in \{0, 1, 2, \dots, l\}, \theta, \delta \in [0, 2\pi)\}.$$

$$U = \{(\delta_t, \tau_t) \mid \delta_t \in [\delta_{min}, \delta_{max}] \tau_t \in [-1, 1]\}$$

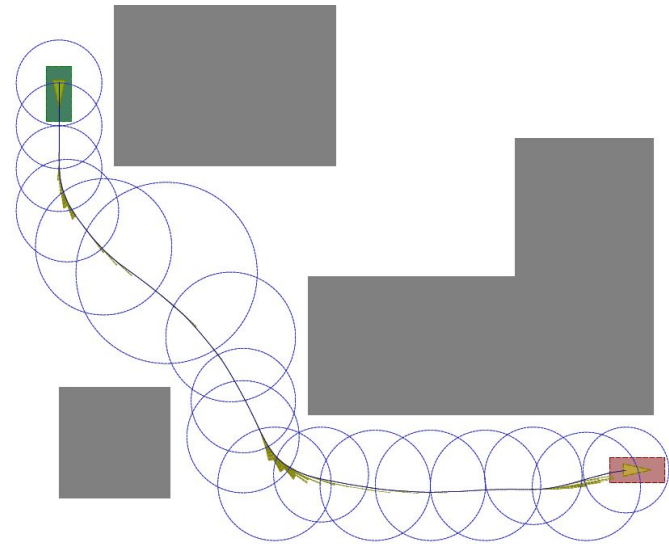
# Hybrid A\*

- Efektivní hledání ve spojitém stavovém prostoru
  - Stavový prostor uniformě rozdělen do diskretních oblastí
  - Při otevření vrcholu si zapamatujeme jak přesný stav vozidla, tak i jeho diskretní souřadnice
  - Každou diskretní oblast expandujeme nejvýše jednou
- Úspěšně použit při DARPA Urban Challenge
  - Stanford Racing Team, vozidlo Junior
- Hrubost diskretizace má zásadní vliv na dobu hledání řešení i na kvalitu nalezeného řešení
  - Kvůli diskretizaci můžeme přijít o některá řešení



# Space Exploration guided Heuristic Search (SEHS)

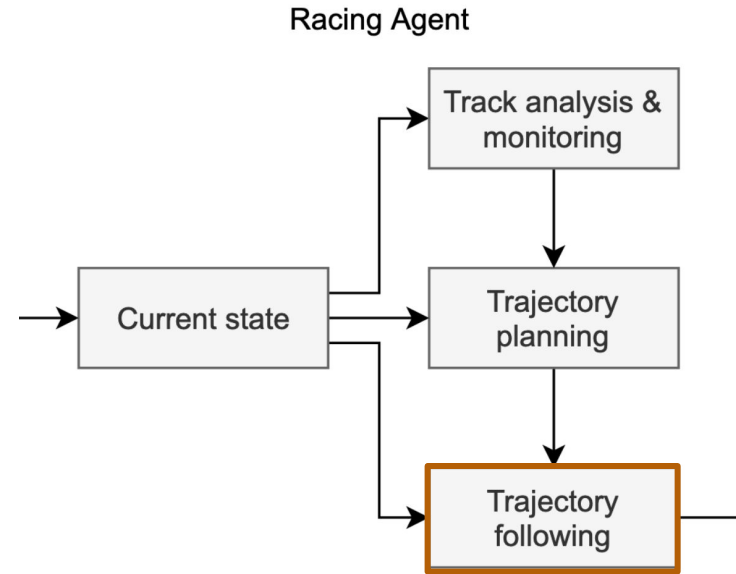
- Také diskretizuje stavový prostor
- 1. Fáze: Space Exploration
  - Najde cestu z překrývajících se kruhů z počátečního stavu do cíle
  - Všesměrový pohyb
- 2. Fáze: Heuristic Search
  - Algoritmus A\*
  - Místo uniformního rozdělení souřadnic  $(x, y)$  je vybrán index nejbližšího kruhu
- Autor: Chao Chen
  - Disertační práce na Technické univerzitě v Mnichově, 2016



Zdroj: Chao Chen, "Combining Space Exploration and Heuristic Search in Online Motion Planning for Nonholonomic Vehicles", 2016

# Sledování trajektorie

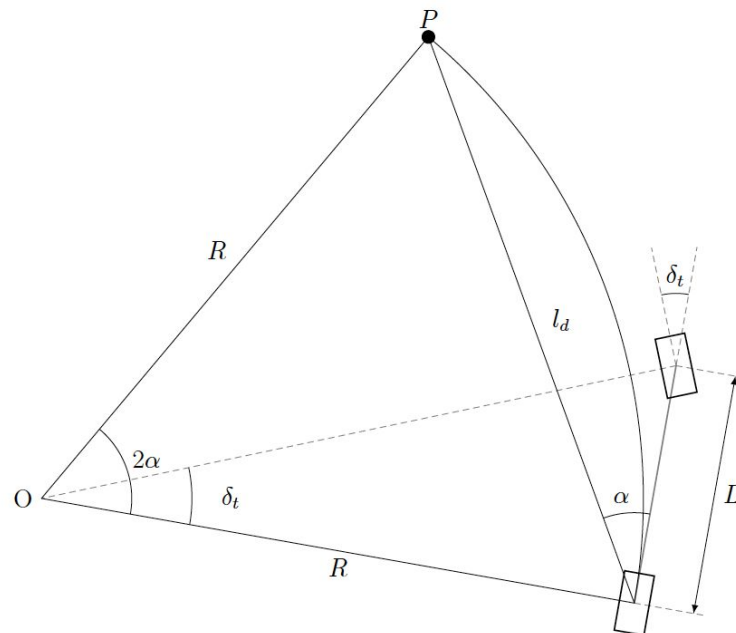
- Vstup:
  - Poslední naplánovaná trajektorie
  - Aktuální odhad stavu vozidla
- Výstup:
  - Řídící signál pro servo a motor
- Implementované a testované algoritmy:
  - Pure Pursuit
  - Dynamic Window Approach (DWA)





# Pure Pursuit

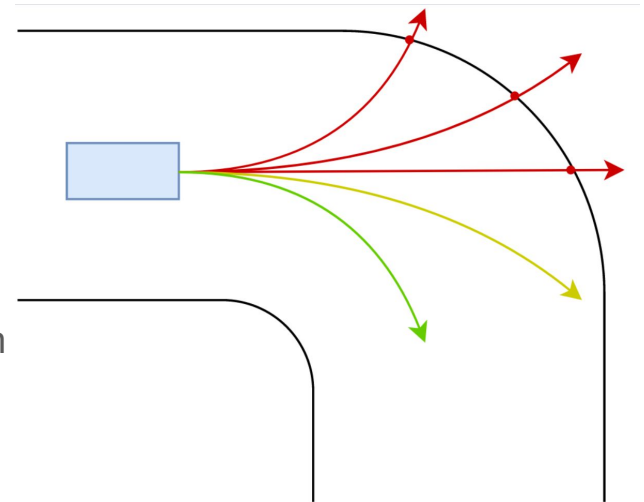
- Výpočet úhlu natočení předních kol
  - uvažuje jen geometrii vozidla a pozici cíle
- Cíl  $P$  je nejzazší bod naplánované dráhy vzdálený od středu zadní nápravy maximálně  $l_d$
- Vzdálenost  $l_d$  závisí na rychlosti vozidla
  - Čím rychlejší jízda - tím větší výhled
- Cílovou rychlost akce volí algoritmus podle rychlosti v nejbližším bodě naplánované trajektorie



$$R = \frac{l_d}{2 \sin \alpha}$$
$$\delta_t = \arctan \frac{L}{R} = \arctan \frac{2L \sin \alpha}{l_d} = \arctan \frac{2L \sin \alpha}{Kv}$$

# Dynamic Window Approach (DWA)

- Vybírá se akce z množiny přípustných akcí pro poslední známý stav vozidla
  - Pro každou vyzkouší, jak by se stav vozidla vyvíjel, kdyby byla tato akce zafixovaná pro nějaký konstantní časový interval (např. 0.5s).
  - Vyřadí se akce, které by vedly ke kolizi.
  - Pro zbývající akce se spočítají “chyby” sledování trajektorie:
    - Vzdálenost od překážek
    - Vzdálenost od naplánované dráhy
    - Rozdíl mezi naplánovanou a predikovanou rychlostí vozidla
    - Rozdíl mezi naplánovaným a predikovaným úhlem natočení vozidla
  - Vybere se akce s nejmenším váženým průměrem chyb

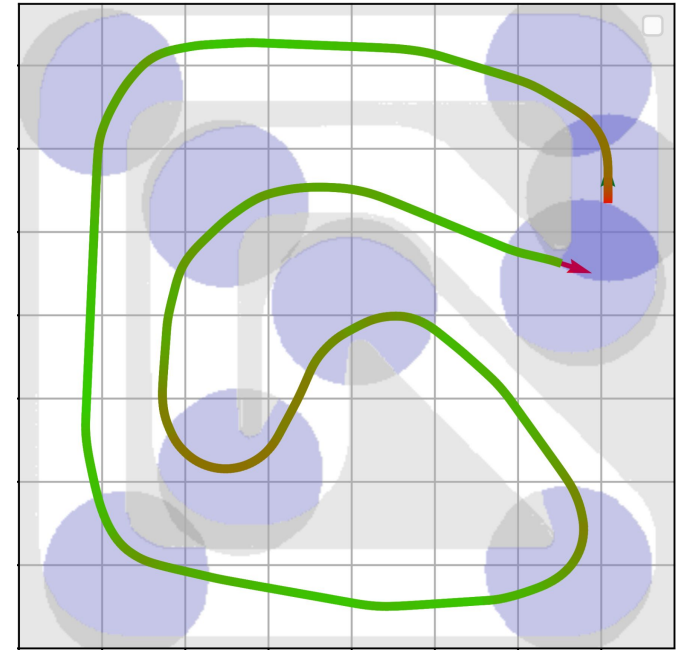


# Experimenty

- Plánování trajektorie pro celé kolo
- Experimenty na robotickém vozítku
- Simulovaný závod

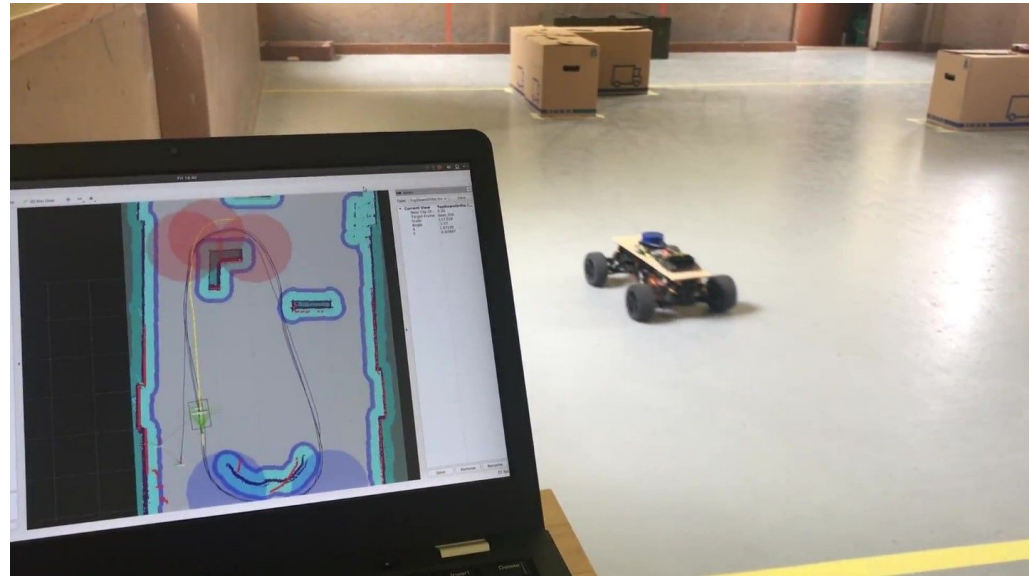
# Plánování trajektorie pro celé kolo

- Srovnání Hybrid A\* a SEHS
- Ze startovací pozice skrze všechny zatáčky
  - 6 různých závodních okruhů
- Zkoušeli jsme různé kombinace parametrů pro diskretizaci stavového prostoru
  - Zajímalo nás vliv zvolených parametrů na kvalitu nalezeného řešení a na dobu hledání řešení
- Cílem bylo zjistit, zda je jeden z algoritmů výrazně lepší než druhý
  - Při vhodně zvolených parametrech
- Ani jeden z algoritmů však nebyl výrazně lepší
  - Každý z algoritmů “vyhrál” právě 3 závody ze 6 (čas do cíle)
  - Hybrid A\* měl často kratší výpočet, avšak ne výrazně



# Robotické vozítko

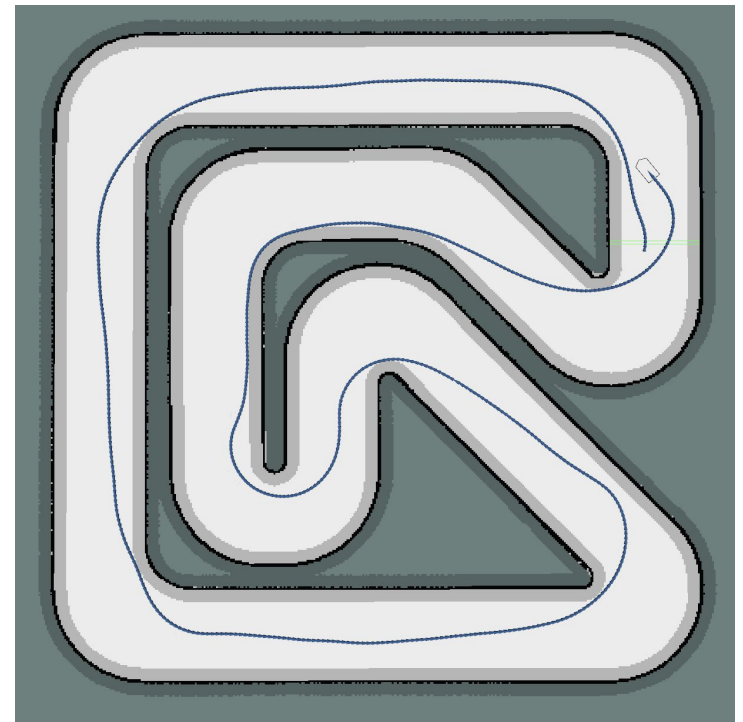
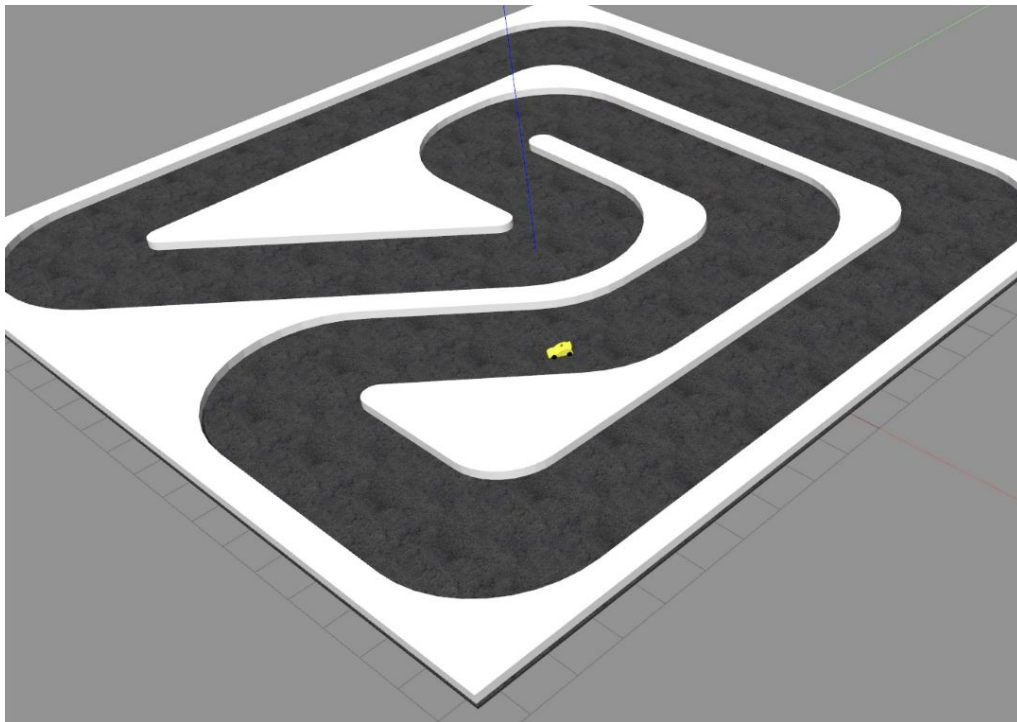
- Vlastní robotické vozítko se senzory a počítačem
  - LIDAR, IMU, encoder
  - Nvidia Jetson Nano
- Problém s odometrií při vyšších rychlostech
- Při nižších rychlostech bylo auto schopno bezpečně jezdit po okruhu
  - cca 1 m/s



Experimenty

# Simulovaný závod

- ROS Gazebo + konfigurace a testovací trať F1/10
  - [https://github.com/f1tenth/f1tenth\\_simulator](https://github.com/f1tenth/f1tenth_simulator)



Hybrid A\* + DWA, čas kola 25.274s

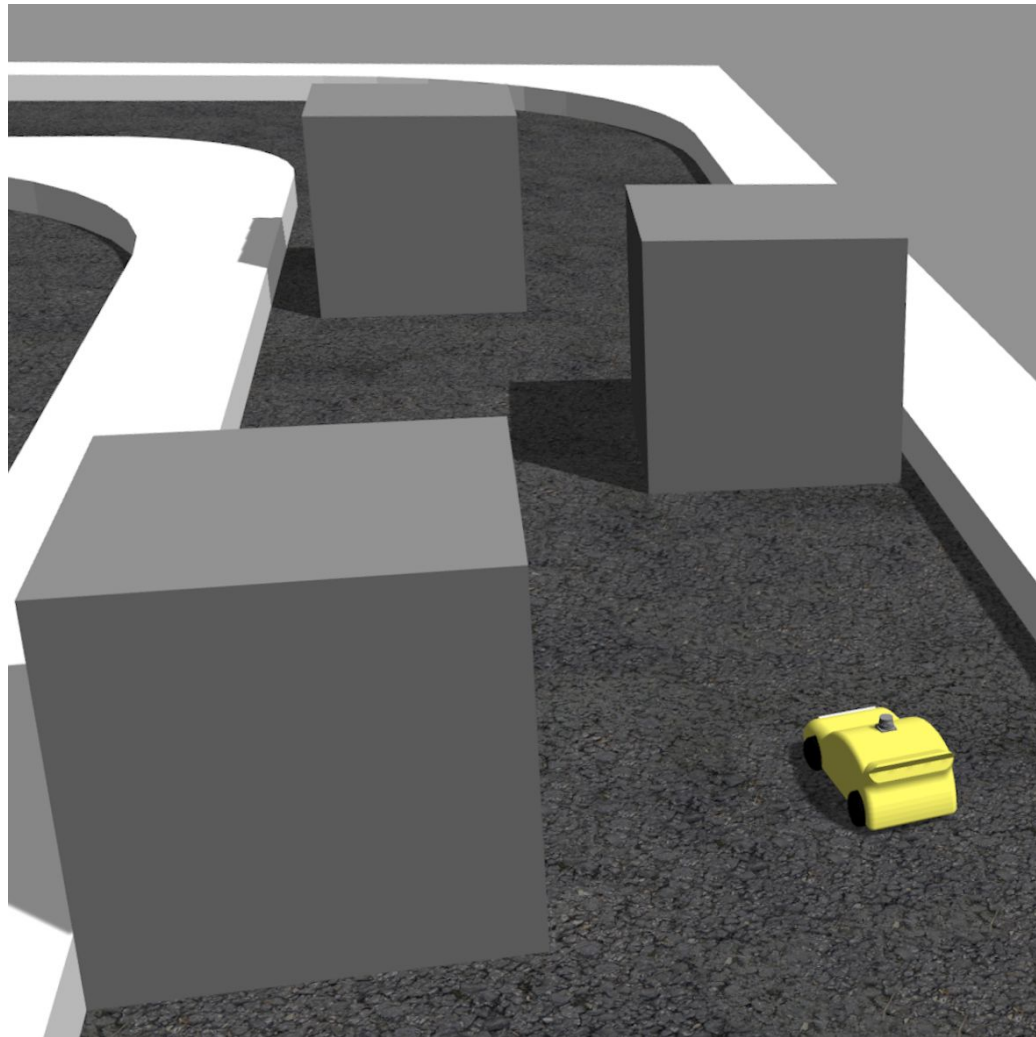
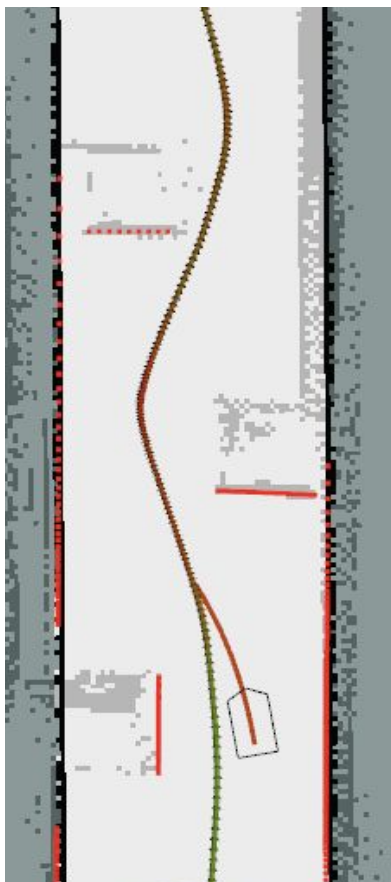
# Simulovaný závod

- Testovali jsme úspěšnost všech kombinací algoritmů
- Nejlepších výsledků jsem dosáhl s pomocí kombinace SEHS + DWA
- Podařilo se překonat referenční řešení pro danou trať
  - Pure Pursuit podél předdefinované trajektorie
  - Implementace od týmu F1/10

Planning alg.	Following alg.	Avg. lap time [s]	Best lap time [s]	Max. speed [m s <sup>-1</sup> ]
SEHS	DWA	<b>25.577</b>	<b>23.815</b>	4.04
Hybrid A*	DWA	25.802	24.295	4.05
SEHS	Pure Pursuit	27.168	25.850	4.03
Hybrid A*	Pure Pursuit	27.484	25.667	4.01
Referenční řešení:			<b>25.911</b>	<b>4.05</b>

Experimenty

# Překážky





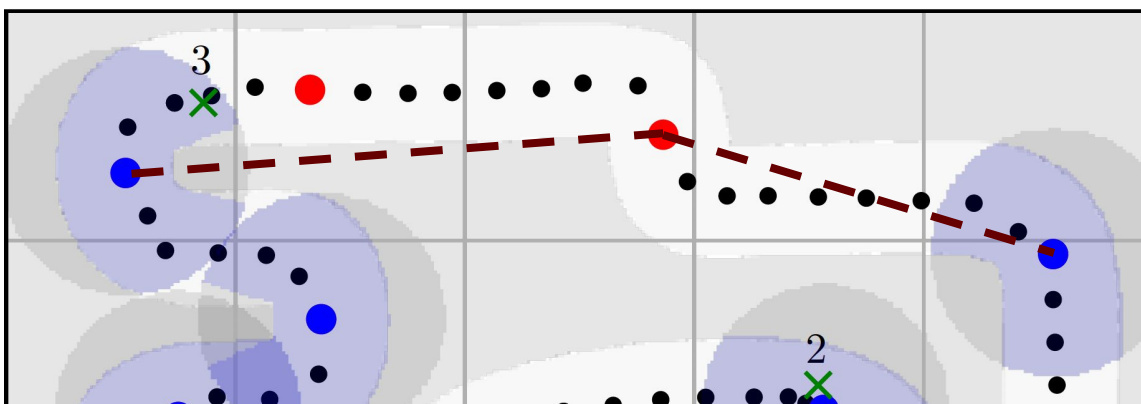
# Shrnutí výsledků

- Implementace algoritmů pro autonomní závodní auto
  - Díky segmentaci trati na jednotlivé úseky mezi zatáčkami zvládne auto přepřánovávat bez potíží během jízdy nehledě na délku okruhu
- Testování na robotickém vozítku
  - Nutnost omezit maximální rychlost na 1 m/s
  - Při vyšších rychlostech již docházelo k problémům s určováním odometrie z dat dostupných z použitých senzorů
- Ověření funkčnosti algoritmů v simulátoru
  - Dosažená rychlost 4.1 m/s (14.76 km/h) u modelu auta 1:10
    - To odpovídá rychlosti 41 m/s (147 km/h) u běžného auta
  - Podařilo se překonat referenční řešení, které sleduje předdefinovanou trajektorii
  - Auto se umí vyhnout i statickým překážkám na trati

Děkuji za pozornost

# Nedetekovaná zatáčka

- V jednom konkrétním případě algoritmus v ukázce selhává
  - Důvod: “Vynecháme body, ve kterých se příliš nemění směr cesty”
    - Hraniční úhel stanoven na:  $\frac{1}{2}\pi = 0.80\pi$  ( $144^\circ$ )
    - Úhel v tomto případě:  $0.88\pi$  ( $159^\circ$ )
- Při vyšší hodnotě hraničního úhlu však přibylo více falešně detekovaných zatáček u jiných okruhů



# Srovnání s jinými řešeními F1/10

- Soutěží F1/10 již proběhlo 6, avšak zdrojové kódy byly zveřejněny pouze pro řešení ze závodu F1tenth Porto Grand Prix 2018
  - <https://github.com/f1tenth/F110CPSWeek2018>
  - Z jiných ročníků nebyly zdrojové kódy zveřejněny
- Ve zdrojových kódech je ukázka 5 řešení
  - Všechna řešení jsou čistě reaktivní
    - žádná lokalizace
    - vybere řídicí výstup jen z dat z LIDARu
  - Nejzajímavější je práce ČVUT (vítěz)
    - algoritmus Follow The Gap
    - snaží se držet uprostřed trati
    - rychlost je odvislá od úhlu zatáčení

