

# Simulace křižovatky

Michal Průšek  
FJFI ČVUT

leden 2025

## 1 Úvod

Tato seminární práce se zaměřuje na simulaci dopravní křižovatky typu T pomocí Inteligentního řidičského modelu (IDM). Cílem simulace je nalézt vhodná počáteční rozdělení rychlostí a časových rozestupů mezi vozidly (při generování vozidel na začátku silnice) spolu s optimalizovanými parametry IDM, aby výsledné rozdělení časových intervalů průjezdů vozidel křižovatkou odpovídalo reálným datům získaným na mnichovské křižovatce. Reálná data z této křižovatky jsou dodána profesorem Krbálkem. Data obsahují jak časové intervaly mezi průjezdy dvou po sobě jedoucích aut na hlavní silnici křižovatkou, tak sloupec určující počet aut z vedlejší silnice, které se do vzniklé mezery zařadily. Dalším cílem je porovnat simulované počty zařazených vozidel do mezery mezi auty na hlavní silnici s reálnými počty.

## 2 Simulace provozu na přímé silnici

Tento kód simuluje provoz na silnici pomocí modelu **IDM (Intelligent Driver Model)** a generování časových intervalů mezi vozidly dle **GIG (Generalized Inverse Gaussian)** a **log-normálního** rozdělení. Volba těchto dvou rozdělení vychází z předpokladu, že odpovídají reálným situacím na pozemní komunikaci, kdy se často předpokládá, že právě GIG rozdělení popisuje intervaly mezi průjezdy vozidel. Na začátku jsou nastaveny realistické parametry, protože hledání optimálních parametrů IDM a daných rozdělení by byl složitější a výpočetně náročnější úkol.

Parametry GIG rozdělení jsou:

- **lambda** = 2.71
- **beta** = 1.03

- `loc = 0`

Parametry byly nalezeny pomocí optimalizačního kódu `find_optimal_GIG.py`.

Parametry log-normálního rozdělení jsou nastaveny tak, aby střední hodnota odpovídala rychlosti 50 km/h se směrodatnou odchylkou 5.4 km/h. Celý proces sbírá a vyhodnocuje data, jako jsou časy mezi průjezdy (roadcross) a intervaly pro spawn nových aut, což slouží k optimalizaci parametrů IDM.

## 2.1 Princip fungování

### 1. Spuštění simulace

- Načte se konfigurace (maximální počet aut, délka silnice, parametry rozdělení, atd.).
- Vytvoří se instance simulace a připraví se okno pro vizualizaci (pokud je zapnuta).

### 2. Generování nových vozidel

- V pravidelných intervalech (vzorkovaných z GIG rozdělení) se “spawnuje” nové vozidlo.
- Každé vozidlo získá **počáteční rychlost** z log-normálního rozdělení, aby se simulovala variabilita vstupních rychlostí.

### 3. Aktualizace polohy a rychlosti vozidel

- V každém simulačním kroku se vyhodnocuje akcelerace a brzdění podle modelu **IDM**, který zohledňuje:
  - Bezpečnostní rozestup od vozidla vpředu.
  - Omezení maximálního zrychlení a zpomalení.
- Vozidla se pohybují po silnici a při průjezdu definovaným měřicím bodem (roadcross) se zaznamená jejich čas.

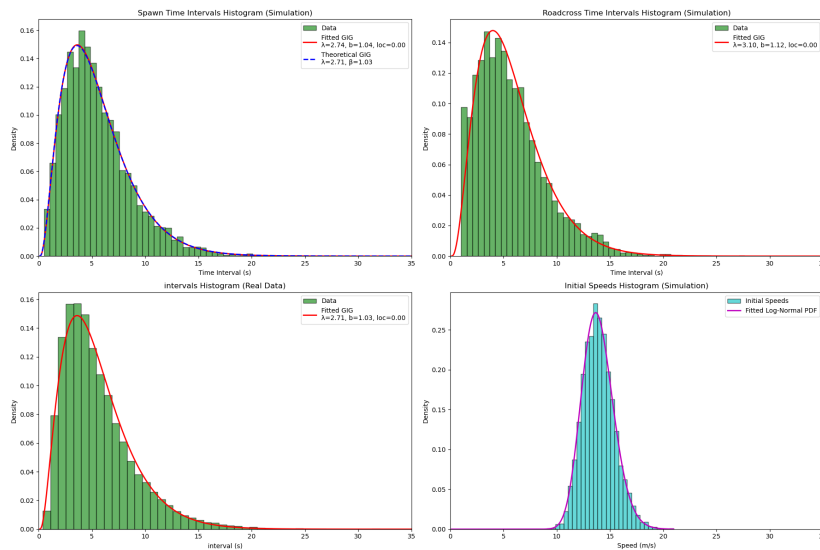
### 4. Ukončení simulace

- Simulace končí, pokud:
  - Všechna vozidla dojedou mimo oblast měření, nebo
  - Dojde ke kolizi (v případě, že nastanou nevhodné podmínky pro spawn nebo parametry IDM).

### 5. Vyhodnocení výsledků

- Po ukončení simulace (pokud není zapnut silent režim) se vykreslí grafy a histogramy:
  - Časové intervaly mezi spawny vozidel.
  - Časové intervaly průjezdů měřícím bodem.
  - Počáteční rychlosti vozidel.
- Porovnání se reálnými daty (načítanými z Excel souboru).
- Provádí se **Kolmogorov-Smirnov** testy pro kontrolu shody empirických dat s GIG/log-normální hypotézou. Pozor, u reálných dat je KS test s GIG hypotézou zamítnut, ačkoliv nafitované GIG rozdělení velmi dobře aproximuje histogram díky velkému množství dat (cca 23000 vzorků).

Vpravo nahoře je zobrazen histogram intervalů průjezdů aut křižovatkou, kdy vozidla řídí model IDM s parametry nalezenými metodou popsanou dále. Je patrné, že nafitované parametry GIG odpovídají parametrům získaným z reálných dat.



Obrázek 1: Výsledky simulace na přímé silnici

## 3 Proces hledání optimálních parametrů IDM

### 3.1 Význam jednotlivých parametrů

- **min\_gap:** Minimální vzdálenost (v metrech), kterou chce řidič udržovat od vozidla před sebou, i když obě auta stojí. Odpovídá “osobnímu prostoru” při nízkých rychlostech či na semaforech.
- **max\_acc:** Maximální možné zrychlení ( $\text{m/s}^2$ ), které vozidlo může dosáhnout, případně které je řidič ochoten využít. Udává agresivitu zrychlení.
- **max\_dec:** Maximální (komfortní) zpomalení ( $\text{m/s}^2$ ) vozidla, které je použito při brzdění, aniž by to bylo nekomfortní či nebezpečné.
- **react\_time:** Časová prodleva (v sekundách) potřebná k reakci řidiče na změny v provozu (např. zpomalení vozidla vpředu). Vyšší hodnota znamená pomalejší reakci.
- **desired\_speed:** Cílová rychlost (v  $\text{m/s}$ ), které chce řidič dosáhnout za optimálních podmínek. Reprezentuje preferovanou rychlost na volné silnici.
- **delta:** Exponent ovlivňující, jak rychle se mění akcelerace vozidla v závislosti na vzdálenosti a relativní rychlosti. Typicky se nastavuje kolem 4; vyšší hodnota značí větší citlivost modelu.

Kód `straight_line.py` slouží k simulacím, které jsou volány z `config_parameters.py`. Tento kód pomocí Grid search algoritmu hledá optimální parametry IDM (`min_gap`, `max_acc`, `max_dec`, `react_time`, `desired_speed`, `delta`) s cílem minimalizovat chi-kvadrát metriku mezi dvěma histogramy. Cílem je, aby při daném počátečním rozdělení rychlostí a intervalů spawnů aut byly časové intervaly průjezdů po 300 metrech stejné jako naměřené na mnichovské křižovatce.

Simulace pro danou volbu parametrů proběhla pětkrát a zaznamenal se průměr získaných chi-kvadrát hodnot. Simulace běžely paralelně na 16 výpočetních uzlech Intel XEON Gold 6134@3.2GHz CPU na FJFI HELIOS clusteru. I když byla zkoušena metoda regresní analýzy pro aproximaci logaritmicke transformované chi-kvadrát metriky, z důvodu vysokého počtu regresorů (cca 5000) a overfittingu byl predikční interval příliš široký. Proto byly optimální parametry IDM zvoleny podle nejnižší průměrné chi-kvadrát hodnoty získané Grid search algoritmem:

- `min_gap` = 1.625
- `max_acc` = 2
- `max_dec` = 4
- `react_time` = 0.3
- `desired_speed` = 19.44
- `delta` = 2

Průměrná chi-kvadrát hodnota pro tuto volbu byla 0.0139. Výsledky Grid search jsou uloženy v souboru `data/IDM_parameters.xlsx`.

Celkově při použití těchto parametrů IDM a zvoleného rozdělení počátečních rychlostí a intervalů spawnů aut mají časové intervaly průjezdů po 300 metrech rozdělení velmi podobné reálným datům z mnichovské křižovatky.

## 4 `T_cross_new.py`

Tento kód implementuje simulaci křižovatky typu T. Předpokládá se, že vozidla na hlavní silnici jsou primární a do mezer mezi nimi se připojují vozidla z vedlejší silnice. Parametry IDM jsou nastaveny dle optimálních hodnot nalezených v předchozím kódu, s tím rozdílem, že `desired_speed` na vedlejší silnici je nastavena na 50 km/h (13.88 m/s) oproti 70 km/h (19.44 m/s) na hlavní silnici. Pro obě silnice jsou rychlosti generovány z log-normálního rozdělení s odlišnými parametry a intervaly mezi spawny aut jsou generovány z GIG rozdělení s různými parametry. Parametry rozdělení však odpovídají těm, které byly použity ve `straight_line.py`. Křižovatka je umístěna na hlavní silnici přesně 300 metrů od jejího začátku.

### 4.1 Zařazování aut z vedlejší silnice

Proces probíhá na základě bezpečné mezery mezi vozidly na hlavní silnici a zahrnuje následující kroky:

#### 1. Zpomalení při přibližování ke křižovatce

- Vozidla z vedlejší silnice (červená auta) se přibližují ke křižovatce a začínají postupně zpomalovat.
- Zpomalení začíná 50 metrů před křižovatkou, kdy se cílová rychlost snižuje z 50 km/h na 10 km/h. Během této fáze se kontroluje, zda je na hlavní silnici volná mezera.

- Pokud není volná, pokračuje auto v postupném zpomalování.

## 2. Kontrola mezery na hlavní silnici

- Po dosažení křižovatky se ověřuje, zda existuje dostatečná mezera pro bezpečné zařazení vozidla.
- Kontrola zahrnuje:
  - **Gap ahead:** Vzdálenost od vozidla před křižovatkou k nejbližšímu vozidlu na hlavní silnici.
  - **Gap behind:** Vzdálenost od vozidla za křižovatkou k nejbližšímu vozidlu na hlavní silnici, přičemž se bere v potaz rozdíl rychlostí mezi čekajícím vozidlem z vedlejší a vozidlem na hlavní silnici.

## 3. Zařazení na hlavní silnici

- Pokud je mezera dostatečná, vozidlo:
  - Změní barvu na modrou (indikace úspěšného zařazení).
  - Obdrží parametry odpovídající hlavní silnici (vyšší rychlost a odlišný model chování).
  - Pokračuje v jízdě po hlavní silnici.
- Pokud není mezera dostatečná, vozidlo čeká u křižovatky na nízké rychlosti.

## 4. Sledování dat o zařazení

- Po každém úspěšném zařazení se zaznamená:
  - Počet vozidel zařazených z vedlejší silnice.
  - Časový odstup mezi průjezdy modrých vozidel křižovatkou.
- Data se ukládají do CSV souboru a mohou být použita pro srovnání s reálnými daty.

# 5 Porovnání reálných dat z T křižovatky a simulace

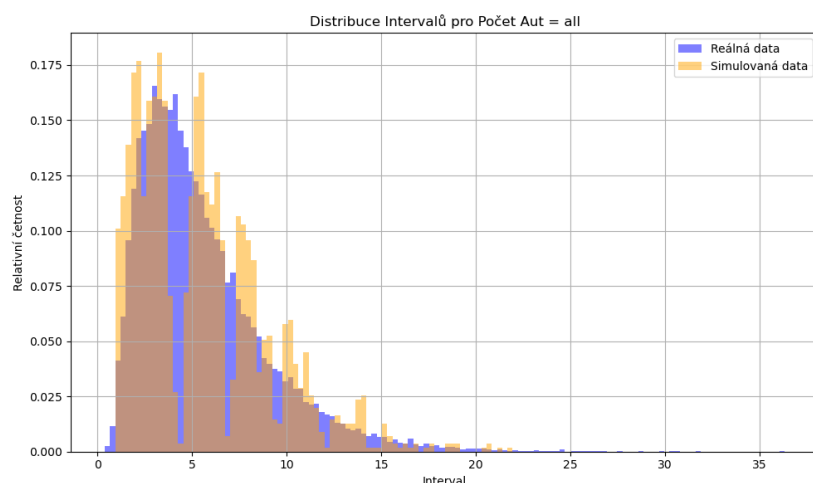
Naměřená data časových intervalů mezi průjezdy na hlavní silnici a počet aut zařazených z vedlejší silnice jsou analyzována a porovnávána s reálnými daty v kódu `merging_data_analysis.py`. Data jsou rozdělena do kategorií

podle počtu zařazených aut a odpovídající kategorie jsou porovnány. Pro každou kategorii je testována podobnost rozdělení (časových intervalů průjezdů vozidel na hlavní silnici) pomocí Kolmogorov-Smirnov GoF testu. Ve všech kategoriích byla nulová hypotéza zamítnuta. Rovněž byl proveden KS GoF test pro celkové výběry intervalů, který byl rovněž zamítnut, ačkoli vizuální porovnání histogramů ukazuje poměrně blízkou shodu.

Následují histogramy:

- Histogram všech časových intervalů průjezdů.
- Histogramy rozdělené dle počtu zařazených aut (0, 1 a 2).

S rostoucím počtem zařazených aut se histogram zužuje a posouvá vlevo.

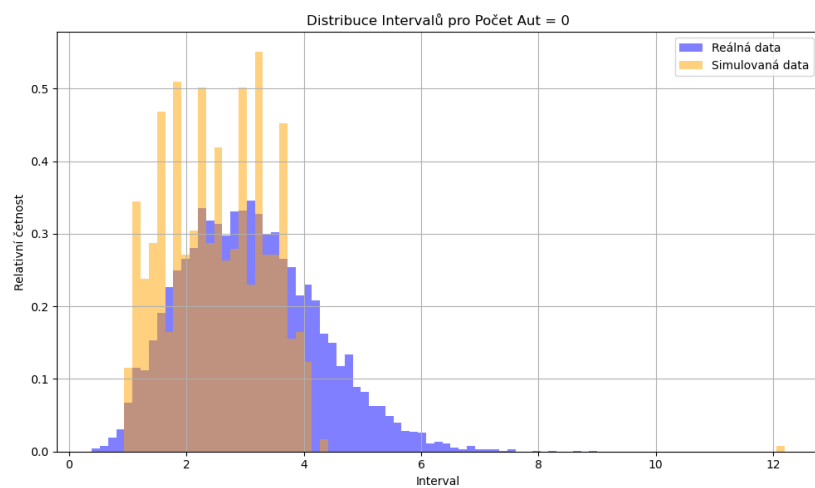


Obrázek 2: Histogram všech průjezdů

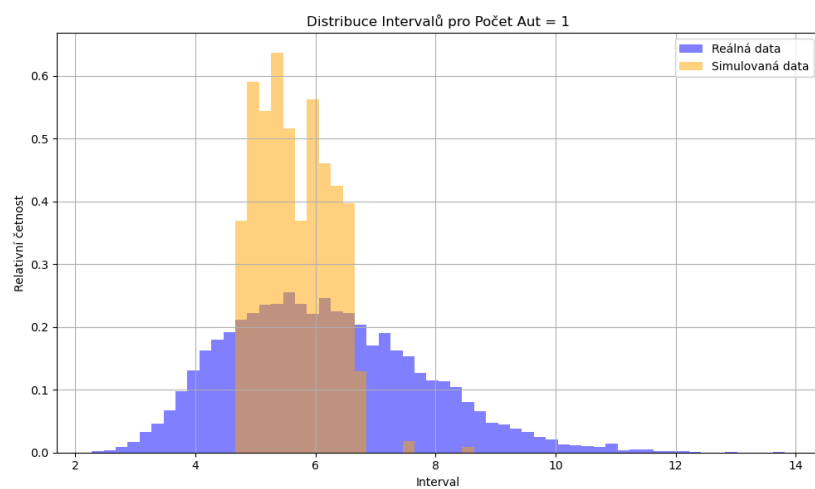
Data z reálných měření a simulace jsou uložena ve složce `data` pod názvy `real_data.xlsx` a `simulation_data.csv`.

## 6 Závěr

Celkově se nepodařilo sladit simulaci s reálnými daty, což může být způsobeno několika faktory. Jednou z možných chyb je nevhodná kalibrace parametrů spawn rozdělení, zejména u rozdělení GIG a log-normálního, které nemusí dostatečně odrážet variabilitu reálného provozu. Další potenciální zdroj nesouladu spočívá v samotném modelu IDM, jenž předpokládá homogenní chování řidičů, zatímco v reálném světě se chování značně liší, což ovlivňuje dynamiku vozidel.

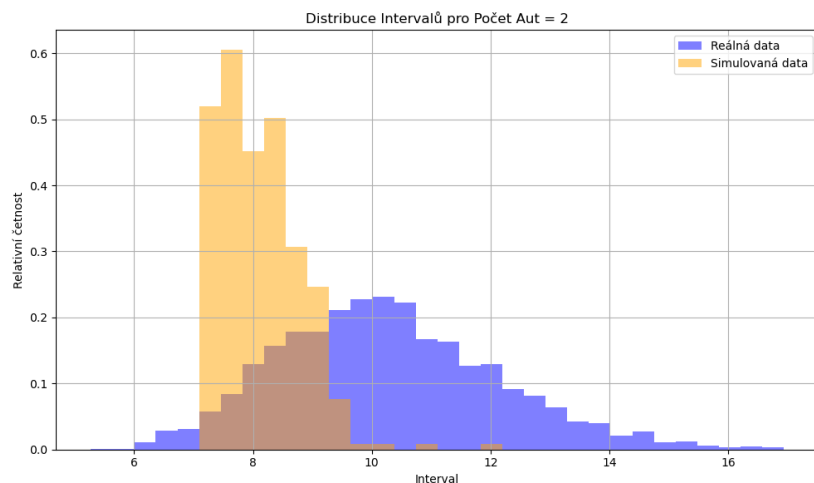


Obrázek 3: Histogram průjezdů: 0 aut



Obrázek 4: Histogram průjezdů: 1 auto





Obrázek 5: Histogram průjezdů: 2 auta

Další (více pravděpodobnou) oblastí, kde mohla vzniknout chyba, je logika zařazování vozidel z vedlejší silnice. Tento proces nemusí dostatečně napodobovat složitý rozhodovací proces řidičů při vjezdu na hlavní silnici, což vede k odchylkám od reálných výsledků.

Pro zlepšení simulace bych navrhoval:

- Provést detailnější kalibraci statistických rozdělení na základě rozsáhlejších reálných dat.
- Implementovat heterogenní chování řidičů, například variabilní reakční doby a rozdílné požadavky na minimální vzdálenost.
- Vylepšit logiku zařazování vozidel z vedlejší silnice tak, aby lépe reflektovala reálné dopravní situace.

## 7 Cluster

Složka `cluster` obsahuje soubory, které byly použity pro simulaci na školním FJFI HELIOS clusteru. Dokumentace clusteru je k dispozici zde: <http://helios.fjfi.cvut.cz/>.