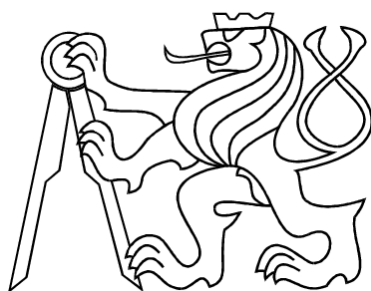


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ



BAKALÁŘSKÁ PRÁCE

Experimenty v Matlab-Aimsun Toolboxu

Praha, 2007

Autor: Marek Elbl

Katedra řídicí techniky

Školní rok: 2006/2007

Z a d á n í b a k a l á ř s k é p r á c e

Student: Marek E l b l

Obor: Kybernetika a měření

Název tématu: Experimenty v Matlab-Aimsun Toolboxu

Z á s a d y p r o v y p r a c o v á n í :

1. Seznamte se s prostředím dopravního simulátoru GETRAM.
2. Prostudujte si stávající dopravní Aimsun-Matlab Interface, vyvíjený na ÚTIA AV ČR, který umožňuje komunikaci Matlabu s dopravním simulátorem Aimsun.
3. Navrhnete systém pro snadné provádění experimentů.
4. Vytvořte funkci, která bude k prováděným experimentům automaticky generovat dokumentaci ve formátu LaTeX obsahující informace, které bude uživatel vyžadovat.
5. Vytvořte dokumentaci a uživatelskou příručku k této části toolboxu.

Seznam odborné literatury:

- [1] Gebouský P.: *AIMSUN-MATLAB Interface - User's Guide*. ÚTIA AV ČR, Praha, 2006.
- [2] Transport Simulation Systems: *AIMSUN Version 4.2 - User manual*. TSS, 2004.
- [3] Dohnal, P.: Bayes Estimation of a Queue Length. In *Proceedings of the 7th International PhD Workshop: Interplay of societal and technical decision-making, Young Generation Viewpoint*. ÚTIA AV ČR, Praha, 2006, str. 1-8.

Vedoucí bakalářské práce: Ing. Pavel Dohnal

Datum zadání bakalářské práce: zimní semestr 2006/07

Termín odevzdání bakalářské práce: 15. 8. 2007

Prof. Ing. Michael Šebek, DrSc.
vedoucí katedry



Prof. Ing. Zbyněk Škvor, CSc.
děkan

V Praze, dne 28. 2. 2007

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb. , o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne _____

podpis

Poděkování

Zde bych v první řadě rád poděkoval Ing. Pavlu Dohnalovi za odborné rady a pomoc při vedení bakalářské práce. Dále pak chci poděkovat své rodině a všem přátelům, kteří mně při studiích a na cestě k vytvoření této práce podporovali.

Abstrakt

Tato práce popisuje provádění simulací a experimentů s různými algoritmy řízení na dopravních oblastech. Pro účely návrhu a simulace oblastí je použit softwarový balík GETRAM. Algoritmy řízení jsou implementovány v MATLABu pomocí Aimsun - MATLAB toolboxu vyvíjeném na ÚTIA AV ČR, který s GETRAMem spolupracuje. Cílem je prostudovat danou problematiku, podílet se na vývoji tohoto toolboxu a vytvořit systém pro automatické generování reportu o proběhlých experimentech.

Abstract

This work describes providing simulations and experiments with many kinds of algorithms of controlling in traffic areas. The software pack GETRAM is used towards simulation. Algorithms of controlling are implemented in MATLAB by means of Aimsun - MATLAB toolbox (the GETRAM co-worker) which is develop in ÚTIA AV ČR. Our purpose is to study this question, then to participate in progression of the toolbox as well and to create the system for automatic generating report on already made experiments.

Obsah

Seznam obrázků	ix
1 Úvod	1
2 Simulace	3
2.1 Rozdělení	3
2.1.1 Mikrosimulace	4
2.2 Pojmy a veličiny	4
3 Getram	5
3.1 Tedi	6
3.1.1 Návrh geometrie dopravní sítě	6
3.1.2 Vytvoření křižovatek a jejich řízení	7
3.1.3 Množství dopravních prostředků na vstupních sekcích	8
3.1.4 Rozmístění detektorů	9
3.2 Aimsun	10
4 Aimsun - Matlab toolbox	13
4.1 Provádění experimentů	14
4.1.1 Inicializace	14
4.1.2 Zjišťování dat v průběhu experimentu	15
4.1.3 Provedení jednoho kroku simulace	16
4.2 Řízení na základě zpětné vazby	16
4.3 Vytváření reportu o experimentu	17
5 Vlastní funkce	19
5.1 Vytvoření L ^A T _E Xového reportu o experimentu	19
5.2 Výpis dostupných oblastí pro experimenty	26

5.3	Uložení figur z MATLABu do \LaTeX ového formátu	27
5.4	Převzorkování dat v matici	30
5.5	Výběr více souborů z různých umístění	32
5.6	Grafické rozhraní inicializace oblasti	35
5.7	Grafické rozhraní vytváření reportu	37
6	Závěr	39
	Literatura	42
A	Obsah přiloženého CD	I
B	Ukázka \LaTeXového reportu o experimentu	III

Seznam obrázků

3.1	Schématické znázornění softwaru GETRAM	5
3.2	Sekce s postranními pruhy	6
3.3	Několik spojených sekcí	6
3.4	Vytváření křižovatek	7
3.5	Editace traffic result	9
3.6	Editace O/D Matrix	10
3.7	Rozmístění detektorů	11
4.1	Předávání dat mezi GETRAM/AIMSUN a Matlabem	13
5.1	Příklad znázornění náhledu oblasti "area 2" v reportu	20
5.2	Příklad vykreslení vjezdů pro ramena 1 a 3 oblasti "area 2" v reportu . .	21
5.3	Příklad vykreslení (intenzity a obsazenosti) z detektoru 13	22
5.4	Příklad vykreslení délek kolon a rozdílu od odhadovaných hodnot z ramene 1	23
5.5	Příklad pro zobrazení popisku ke každé figuře	28
5.6	Příklad pro zobrazení jednoho popisku pro řádek	29
5.7	Příklad pro zobrazení jednoho popisku pro sloupec	30
5.8	Dialogové okno vyvolané funkcí uigetfilelist	33
5.9	Dialogové okno funkce getfilelist vyvolané po stisknutí tlačítka Add . . .	34
5.10	Odstranění položky ze seznamu	35
5.11	Dialogové okno funkce uiinitarea	37
5.12	Dialogové okno funkce uiinitarea (umístění a informace o reportu)	38

Kapitola 1

Úvod

Řízení dopravy se stalo v současnosti ve větších městech již prakticky samozřejmostí. Právě kvalita tohoto řízení ovlivňuje mnoho faktorů. Ať už se jedná o čas účastníka dopravního provozu strávený čekáním v kolonách nebo třeba o dopad vzniklých zplodin na životní prostředí. Dobré řízení dopravy může eliminovat vznik dopravní zácpy v době špiček, a tím například snížit čas průjezdu vozidla danou oblastí. Řídících systémů a algoritmů řízení existuje mnoho, ovšem experimentování s danými algoritmy řízení nelze provádět v reálném provozu. To byl jeden z důvodů, proč byla vyvinuta řada simulačních nástrojů.

Cílem této bakalářské práce je seznámení se s dopravním simulátorem GETRAM, což je softwarový balík obsahující aplikace k návrhu (TEDI) a simulaci (AIMSUN) dopravních oblastí. Dále prostudování Aimsun-Matlab toolboxu, který je v současné době vyvíjen na ÚTIA AV ČR (Ústav teorie informace a automatizace AV ČR) a slouží jak k propojení Matlabu a aplikace AIMSUN, tak k provádění experimentů s různými algoritmy řízení. Posledním krokem je na základě těchto znalostí navrhnout systém pro automatické vytváření výsledného reportu ve formátu \LaTeX o proběhlém experimentu.

V první fázi této práce bude provedeno rozdělení simulačních nástrojů se zaměřením na mikrosimulační nástroje. Následovat bude seznámení se s konkrétním softwarem pro návrh dopravních oblastí a mikrosimulaci dopravy GETRAM, kde bude kladen důraz na kroky potřebné k vytvoření funkční oblasti, na které bude možno provádět experimenty. Dalším krokem bude vysvětlení propojení mezi simulačním programem AIMSUN a Aimsun - Matlab toolboxem a dále popis funkcí toolboxu nutných pro realizaci experimentů na dané oblasti. Nakonec bude popsán vytvořený systém pro automatické generování reportů a popis funkcí, které byly nutné k tomuto účelu naprogramovat.

Kapitola 2

Simulace

Simulace dopravy v různých oblastech má svůj význam hned z několika důvodů. Díky simulacím můžeme snadno sledovat vliv různých řídicích systémů na dopravní situaci a zároveň tyto řídicí systémy můžeme snadno porovnávat. Simulace představuje také pro investora výhodu v podobě minimalizace rizika nevhodné investice. Důležitý význam simulace je také v případě, když nelze provádět experimenty se skutečným systémem. Simulace také může odpovědět na otázky, jestli bude stávající dopravní síť stačit požadavkům za několik let nebo jak se změní dopravní situace při plánované modifikaci oblasti. Na závěr pomáhá při rozhodování, které z navržených dopravních řešení je nejvhodnější použít.

2.1 Rozdělení

Různé druhy simulace dopravy se liší převážně detailností modelu, se kterým pracují. Pokud model zohledňuje jednotlivá auta, hovoříme o takzvané **mikrosimulaci**, jestliže model pracuje s dopravními proudy, které popisuje standardními veličinami, jako například intenzita, obsazenost, rychlost proudu, jedná se o takzvanou **makrosimulaci**. Ve většině jsou používány nástroje mikrosimulační. Může však nastat situace, kdy by byla mikrosimulace příliš pracná, a tudíž i nákladná nebo zbytečně detailní. V tom případě nastupují na řadu makrosimulační prostředky. Výhoda těchto prostředků je v možnosti popisu rozsáhlých sítí a možnost automatizace při vytváření modelu. Na druhou stranu nevýhodou je, že při drobných úpravách dopravní oblasti musí dojít k velkému přepracování modelu. Tato práce se však zabývá mikrosimulačními nástroji, a proto se budu podrobněji věnovat jim.

2.1.1 Mikrosimulace

Ověřováním mikrosimulací nebo také mikroskopickou simulací získáme představu o dopadu konkrétních změn v prakticky jakékoli dopravní oblasti. Může se jednat o změny stavební, změny ve značení atd... . Dále můžeme získat předpověď veličin, jako například intenzity dopravy, spotřeby pohonných hmot, vznik emisí a jiných. Podstata mikrosimulace je v počítačovém modelu dopravní sítě, kde jsou modelovány pohyby jednotlivých vozidel. V těchto modelech mohou být zohledněny velice podrobně parametry vozidel, dopravních sítí a také chování řidičů. U parametrů vozidel lze ovlivňovat hodnoty jako například jejich rozměry, maximální rychlost, zrychlení atd.. Chování řidičů může být definováno například až do úrovně, jak dlouho jsou ochotni čekat, dávat přednost na křižovatkách nebo jejich způsob předjíždění či brždění. U těchto hodnot může být také zadána míra rozptylu od nastavené hodnoty, a tudíž se mohou parametry vozidel stejného typu různě odlišovat. Pomocí těchto nástrojů lze namodelovat jak silnice prvních tříd, tak dálnice, složité městské struktury, chodce, autobusovou dopravu, cyklisty a jiné. Výhodou mikrosimulací je také názornost pomocí vizualizace. Některé programy nabízejí k tomuto účelu také velice propracovanou 3D grafiku. Dostupných mikrosimulačních programů je více. Například: VISSIM, Paramics, nebo GETRAM/AIMSUN 3, se kterým spolupracuje Aimsun - Matlab toolbox 4, a který bude podrobněji popsán dále.

2.2 Pojmy a veličiny

Zde bych chtěl uvést jednotky a pojmy, které se budou dále v textu často vyskytovat. Zpětnovazební řízení rozhoduje o regulačním zásahu na základě dat, získaných z **detektorů** rozmístěných v dopravní síti. Dopravní síť je rozdělena na jednotlivá **ramena**, ke kterým se data vztahují. Na každém rameni může být rozmístěno více detektorů. K řízení jsou použita tato data z detektorů:

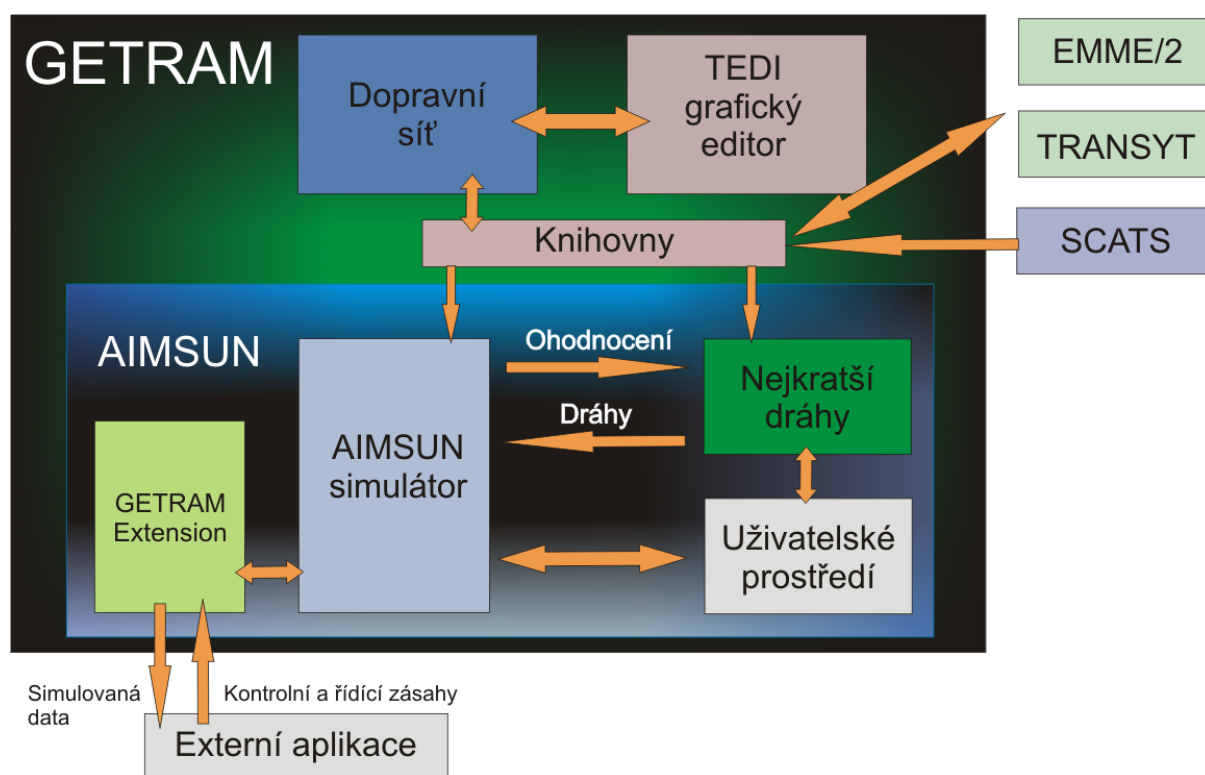
- **Occupancy** (Obsazenost) [%] Udává dobu, po kterou je detektor obsazen vozidlem.
- **Intensity** (Intenzita) [uv/s] Počet vozidel za sekundu. uv (jvoz) je jednotkové vozidlo, neboli průměrné vozidlo v koloně.

Další veličinou je **délka kolony** a pracuje se buď s její maximální, nebo průměrnou délkou. Řídicí systém na základě těchto hodnot ovlivňuje délky **fází**, což je časový úsek, kdy je na semaforech zelená resp. červená.

Kapitola 3

Getram

Prostředí GETRAM (schéma viz obr. 3.1) je softwarový balík obsahující aplikace pro návrh dopravních oblastí a simulaci dopravních situací. Skládá z několika částí. Grafické prostředí TEDI slouží jako editor dopravní sítě. Aplikace AIMSUN se používá pro vlastní simulaci a prostředí AIMSUN 3D pro vizualizaci simulací a zobrazení ve 3D. Dále modul GETRAM Extension umožňuje komunikaci s externími aplikacemi.



Obrázek 3.1: Schématické znázornění softwaru GETRAM

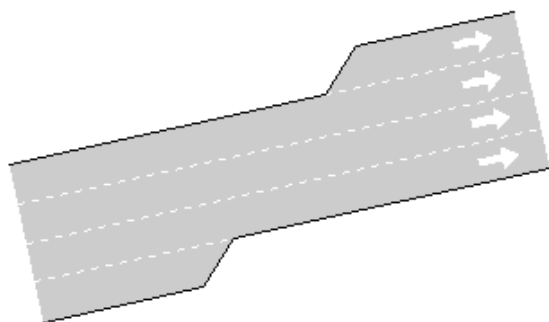
3.1 Tedi

Jde o grafický editor umožňující zadávání geometrie dopravní sítě. Je možné zadávat jak městské, tak meziměstské silnice. U silnic je k dispozici editace různých prvků, jako jsou například vjezdy dopravních prostředků, nájezdy na silnice, velikost postranních čar, semaforey, detektory. Návrh těchto mikromodelů je usnadněn tím, že je možné na pozadí pracovního plátna umístit obrázek dané oblasti v rozlišení 1 metr. Vjezdy automobilů (počty automobilů za hodinu pro danou sekci) mohou být určeny přímo v tomto editoru nebo mohou být zadány externě, což bude zmíněno v dalších kapitolách. Obdobným způsobem může být zadáno řízení křižovatek.

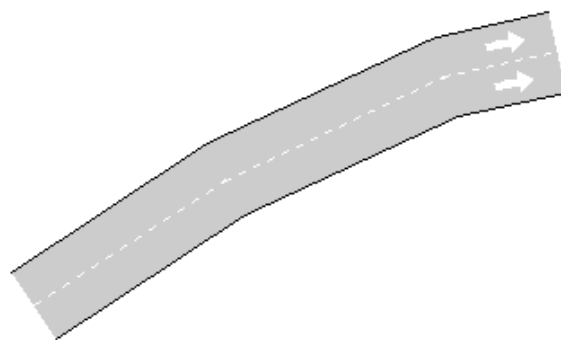
Pro vytvoření funkčního modelu dané oblasti se statickými vjezdy automobilů a řízením křižovatek je nutné provést následující kroky.

3.1.1 Návrh geometrie dopravní sítě

Celá vytvořená oblast se skládá z jednotlivých pospojovaných sekcí (Sections). Každá sekce má svoje ID číslo, jeden směr pohybu vozidel a může mít n pruhů. Začátek a konec sekce může být rozšířen ještě o postranní pruhy nebo o vyhrazené pruhy. Vyhrazené pruhy jsou například pro autobusy, které jsou umístěny vpravo nebo vlevo viz obr. 3.2 (Sekce s třemi pruhy, jedním vstupním a jedním výstupním postranním pruhem). Jednotlivé pruhy v sekcích se mohou rozdělovat v další dvě sekce a naopak se můžou také spojit viz obr. 3.3.



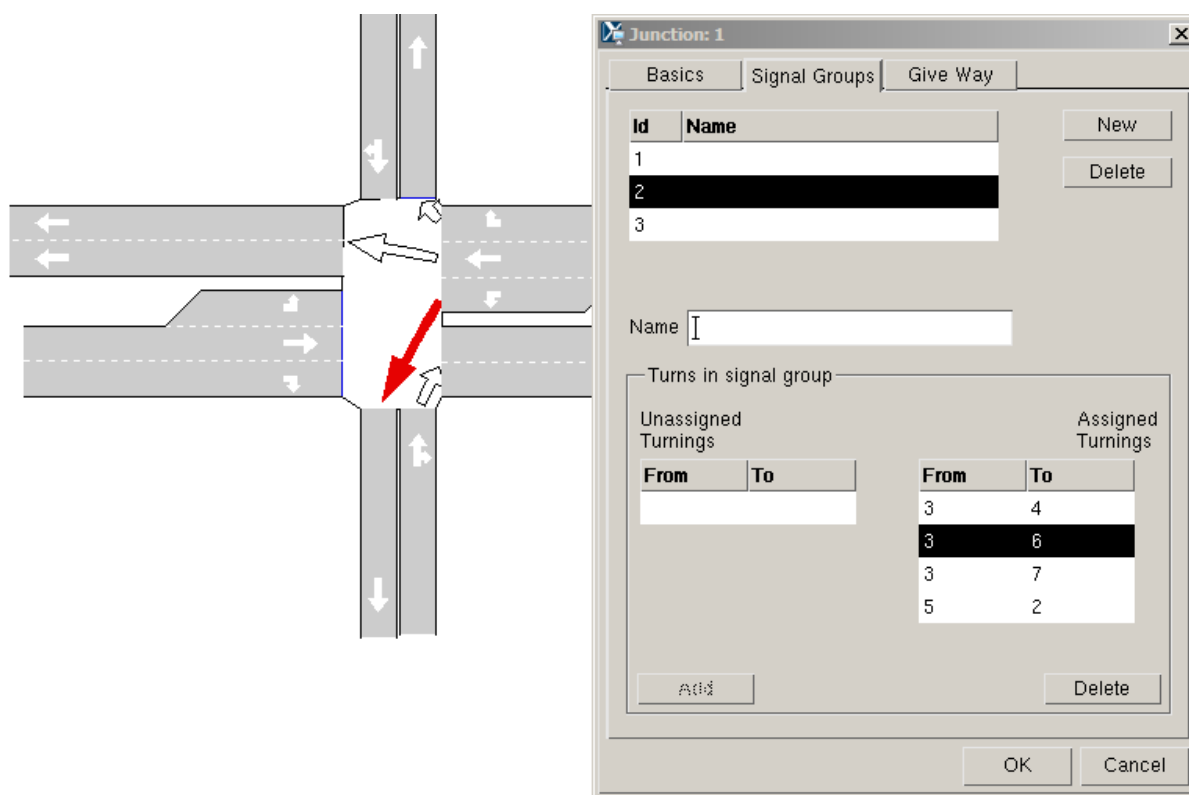
Obrázek 3.2: Sekce s postranními pruhy



Obrázek 3.3: Několik spojených sekcí

3.1.2 Vytvoření křižovatek a jejich řízení

Sekce je možné spojovat do křižovatek (Junctions), kde každá křižovatka má svoje vlastní ID číslo. Křižovatka vzniká postupným přidáváním jednotlivých směrů ze sekce, ze které se startuje k cílové sekci. Tyto směry se dále sdružují do jednotlivých signálových skupin (Signal Groups), což jsou směry, které budou mít ve stejný okamžik zelenou na semaforu, což je zobrazeno na obr. 3.4, kde šipky ukazují všechny směry pro jednu signálovou skupinu. Pro každý směr lze ještě určit kolik vozidel procentuelně z celkového množství odbočí daným směrem.



Obrázek 3.4: Vytváření křižovatek

Po nadefinování signálových skupin je řešeno řízení křižovatek, v programu značené jako takzvaný "Control plan" (v menu položka "Control", "Junctions"). Těchto "Control plans" může být nadefinováno více pro jednu oblast. Pod pojmem řízení se rozumí udání takzvaných "délek fází" (duration) pro jednotlivé signálové skupiny a délek mezifází (interphase). Délka fáze je doba, po kterou má daná signálová skupina zelenou. Mezifáze je doba, kdy je na všech semaforech červená. Při volbě druhu řízení máme na výběr z těchto možností:

- Uncontrolled

Řízení křižovatky je dáno pouze předností v jízdě, semaforey nejsou v provozu.

- Fixed

Délky jednotlivých fází a mezifází jsou pevně dány, jednotlivé Control plans lze měnit v závislosti na denní době.

- External

Zde lze využít GETRAM EXTENSION pro přenechání řízení vnějšímu programu, čehož je využito v našem případě pro externí řízení z Matlabu pomocí Aimsun - Matlab toolboxu.

- Actuated

Řízení křižovatky je dáno NEMA standardem, kterým se v tomto textu nebudeme zabývat.

Pro návrh řízení křižovatek existuje v TEDI ještě další řada parametrů, jako například parametr offset, pomocí kterého je dosažena synchronizace více křižovatek (zelená vlna) nebo yellow time, což je vlastně doba svitu oranžového světla ze začátku fáze s rozsvíceným červeným světlem na semaforu.

3.1.3 Množství dopravních prostředků na vstupních sekcích

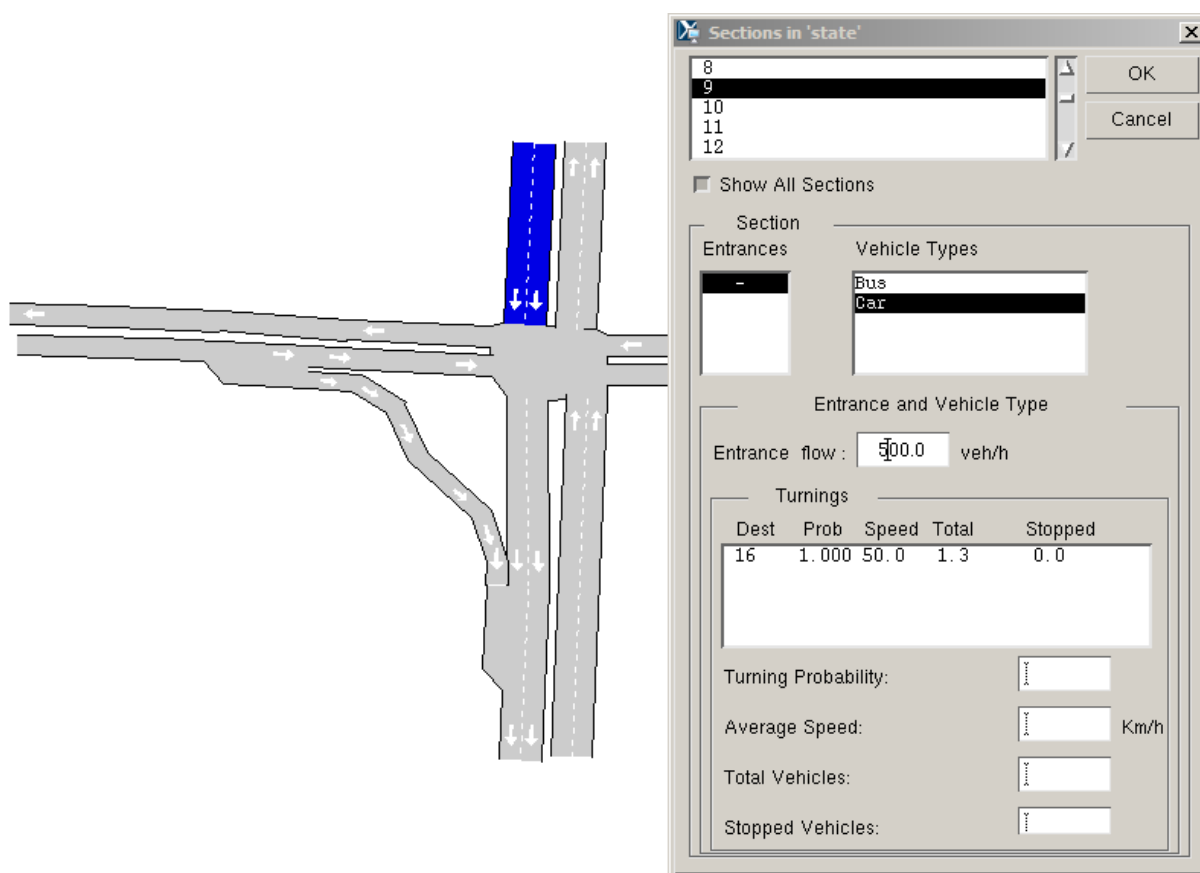
K zadávání množství dopravních prostředků na vstupních sekcích, lze v editoru TEDI použít dva rozdílné přístupy:

- Result container

Tento přístup (v menu položka "Result") přiřazuje každé vstupní sekci kolik dopravních prostředků a jakého typu danou sekci za hodinu projede (položka "State") viz obr. 3.5. Dále je tedy nutné nadefinovat typy aut a jejich vlastnosti (položka "vehicle types"). Tato varianta se předpokládá pro spolupráci s Aimsun - Matlab toolboxem.

- O/D Matrix

V tomto případě se vychází z takzvaných "centroids", u kterých se udá, z jakých sekcí a do jakých sekcí ústí. Následně se vyplní O/D Matice, kde jsou v řádcích zdrojové a ve sloupcích cílové centroidy. Do buněk matice se udá, typ a jaké množství



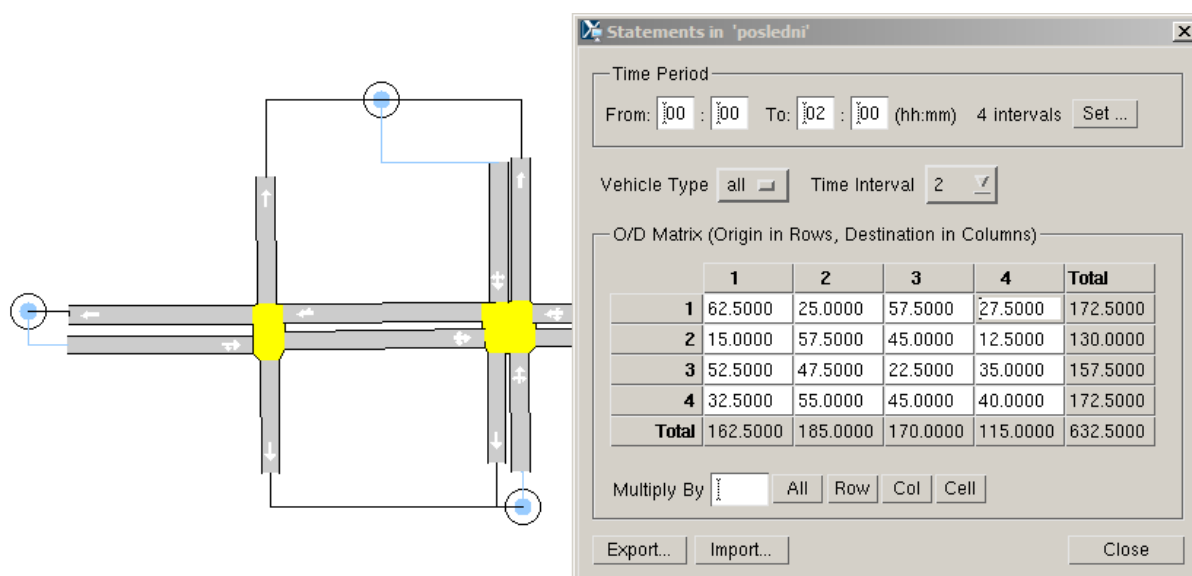
Obrázek 3.5: Editace traffic result

dopravních prostředků z jednoho centroidu do druhého za daný časový úsek projede. Matice a centroidy viz obr. 3.6.

3.1.4 Rozmístění detektorů

Důležitým prvkem pro zpětnou vazbu při řízení křižovatek je rozmístění detektorů. Detektory v editoru TEDI mohou snímat šest veličin.

- Count
Počet vozidel, která projela detektorem
- Speed
Rychlost projíždějícího vozidla
- Presence



Obrázek 3.6: Editace O/D Matrix

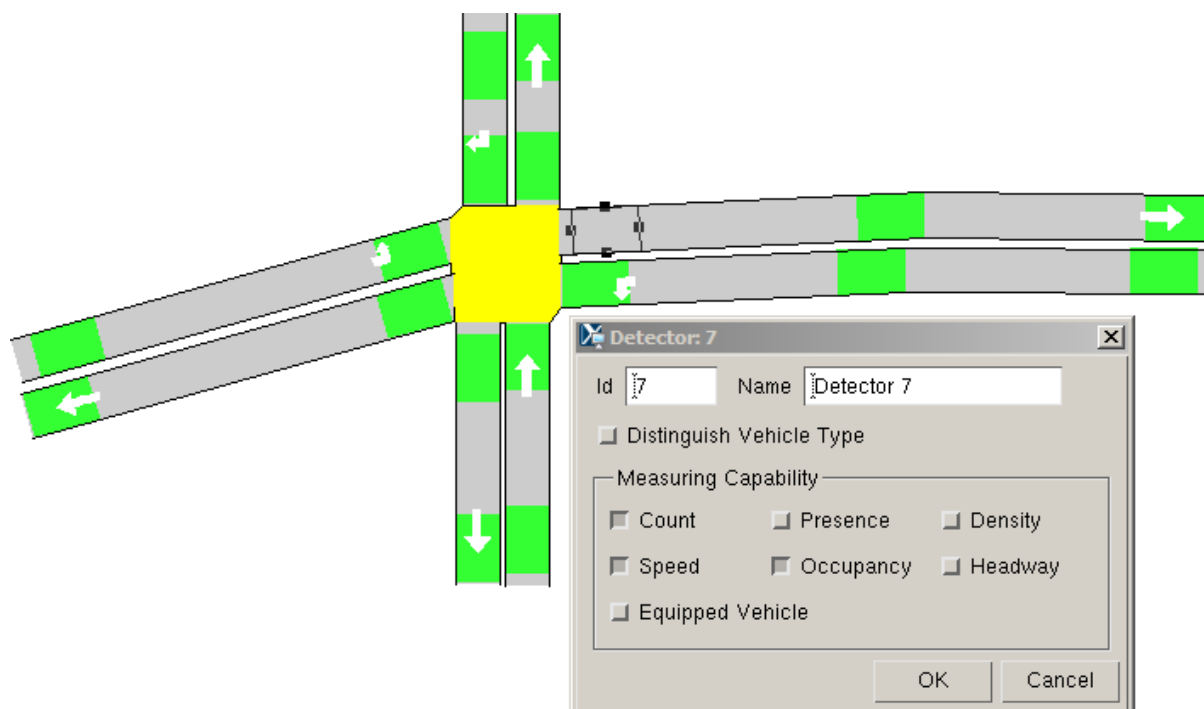
Je nastaveno na 1, pokud při simulaci na konci cyklu je v prostoru detektoru vozidlo, v opačném případě je nastavena 0.

- Occupancy
Obsazenost daného detektoru v procentech.
- Density
Hustota dopravního toku ve vozidlech na kilometr.
- Headway
Průměrná doba, kterou vozidla projíždějí detektorem.

Detektory mohou být nastaveny tak, aby rozlišovaly mezi jednotlivými typy nadefinovaných vozidel (volba Distinguish vehicle type) viz obr. 3.7.

3.2 Aimsun

Aimsun je aplikace pro mikrosimulaci dopravy. Jsou simulovány pohyby jednotlivých aut a díky možnosti nadefinování různých druhů aut a jejich parametrů je možné dosáhnout poměrně přesného modelu v porovnání s realitou. Simulace je definována pro určitý čas



Obrázek 3.7: Rozmístění detektorů

a rychlost. Simulace může být reálná, nebo zrychlená. Pro spuštění simulace je nutné provést tyto kroky:

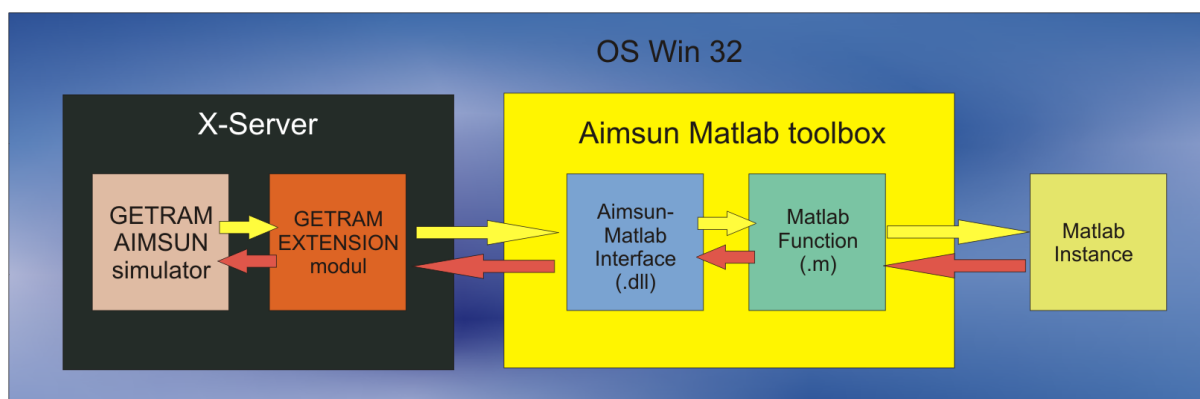
- Načíst dopravní síť vytvořenou v editoru (Network).
- Pro tuto síť načíst vjezdy vozidel jako "Traffic result" (v našem případě), nebo jako "O/D Matrix".
- Načíst řízení křižovatek (Control).
- Pokud byla v editoru nadefinována i veřejná doprava, tak i ta se musí samostatně načíst.

Takto připravenou simulaci lze uložit jako takzvané "Scenario", což je soubor .sce, se kterým pracuje i Aimsun-Matlab toolbox. V případě, že bylo nastaveno řízení všech křižovatek jako "External", je takto nastavená simulace resp. scenario připravena pro externí ovládání pomocí API GETRAM EXTENSION (.dll knihovny), čehož využívá i Aimsun - Matlab Interface toolbox.

Kapitola 4

Aimsun - Matlab toolbox

Aimsun - Matlab je toolbox vyvíjený na ÚTIA AV ČR pro komunikaci Matlabu se simulátorem Aimsun a k provádění experimentů. Je dostupný ke stažení na <http://marabu.utia.cas.cz:1800/svn/doprava/AIMSUN-MATLAB/toolboxASYN>, login a heslo jsou quest. Komunikace externí aplikace s GETRAM/AIMSUN a předávání dat v průběhu simulace je zprostředkováno pomocí rozhraní GETRAM Extensions. GETRAM Extensions poskytuje pomocí řady funkcí data externí aplikaci (například délky kolon, nebo data z detektorů) a externí aplikace může provádět na základě těchto dat řízení, které například v podobě délky fází na křižovatkách zasílá AIMSUNu opět pomocí GETRAM Extensions. Součástí toolboxu jsou knihovny (.dll) obsahující funkce, obstarávající komunikaci mezi GETRAM Extensions a samotnými Matlabovskými funkcemi (.m). Propojení jednotlivých modulů je zobrazeno na obr. 4.1.



Obrázek 4.1: Předávání dat mezi GETRAM/AIMSUN a Matlabem

4.1 Provádění experimentů

Samotný účel vývoje tohoto toolboxu je ověřování algoritmů řízení na modelech dopravních oblastí. Každý experiment s danou oblastí se provádí pomocí čtyř funkcí: `initArea`, `getActualData`, `getActualStatistic`, `setActualControl`. Tyto funkce slouží k inicializaci oblasti, zjištění aktuálních dat simulace a k nastavení řízení křižovatek.

4.1.1 Inicializace

Při inicializaci je nutno zadat, jaká oblast bude pro experiment používána, jaká vozidla budou zastoupena a kam budou vozidla vjíždět.

[MyArea]=initArea(AreaName, VehGenerator, Entrances, ReportPath, Description, Author);

- **MyArea**
Struktura obsahující informace o probíhajícím experimentu
- **AreaName**
Řetězec se jménem oblasti pro experiment
- **VehGenerator**
Jakým způsobem budou vjezdy vozidel generovány
 - 0 - Vjezdy budou generované interně AIMSUNem
 - 1 - Offline, vjezdy budou zasílané v každém kroku simulace z předem uloženého souboru
 - 2 - Realtime, soubor obsahuje pouze informace o poměru zastoupení druhů vozidel
- **VehGenerator**
Řetězec s názvem souboru obsahujícím data o vjezdech do jednotlivých sekcí a poměr zastoupení jednotlivých druhů vozidel.
- **ReportPath**
Řetězec s cestou k adresáři, do kterého bude vytvořen Report o experimentu.
- **Description**
Řetězec s krátkým popisem experimentu.

- Author

Řetězec se jménem autora.

Pro zobrazení dostupných oblastí a vjezdů lze použít funkci `showAreaList` viz. 5.2. Pro příjemnější práci s funkcí `initArea` byla vytvořena funkce `uiinitarea` viz. 5.6, kde se zadávají jednotlivé parametry v dialogovém okně.

4.1.2 Zjišťování dat v průběhu experimentu

Tato funkce slouží pro zjišťování agregovaných dat z detektorů a k nim časových údajů, kdy byla data pořízena.

```
[MyArea, Intensity, Occupancy, Velocity, Day, Time] = getActualData (  
    MyArea);
```

- MyArea

Struktura obsahující informace o probíhajícím experimentu

- Intensity

Vektor intenzit

- Occupancy

Vektor obsazenosti

- Velocity

Vektor rychlosti

- Day

Den, ze kterého pocházejí data

- Time

Čas v sekundách, ze kterého pocházejí data

Další funkce zjišťuje délky kolon pro jednotlivé jízdní pruhy.

```
[MyArea, QueueAverage, QueueMax, Day, Time] = getActualStatistics (  
    MyArea);
```

- MyArea

Struktura obsahující informace o probíhajícím experimentu.

- QueueAverage
Vektor průměrných délek kolony.
- QueueMax
Vektor maximálních délek kolony.
- Day
Den, ze kterého pocházejí data
- Time
Čas v sekundách, ze kterého pocházejí data

4.1.3 Provedení jednoho kroku simulace

Funkce provede jeden krok simulace a nastaví délky fází na všech křižovatkách.

```
[MyArea] = setActualControl (MyArea, PhasesLengths, EstimatedQueues,  
    RealEntrances);
```

- MyArea
Struktura obsahující informace o probíhajícím experimentu.
- PhasesLengths
Matice obsahující délky jednotlivých fází.
- EstimatedQueues
Sloupcový vektor odhadnutých délek fází uživatelem.
- RealEntrances
Sloupcový vektor vjezdů pro případ generování vjezdů v reálném čase.

4.2 Řízení na základě zpětné vazby

Tyto zmíněné čtyři funkce lze využít pro zpětnovazební řízení. Zde je uveden příklad programového kódu z Matlabu pro realizaci zpětnovazební smyčky.

```

% inicializace oblasti a vstupnich dat
Experiment = initArea (AreaName, VehGenerator, Entrances, ReportPath, Description, Author);

% Experiment.state je po dobu simulace nastaven na 1, smyčka se stále opakuje
while Experiment.state,

    % Zjisti agregovanych dat z detektoru
    [MyArea, Intensity, Occupancy, Velocity, TimeData] = ...
    getActualData (MyArea);

    % Ziskani dat o delkach kolom
    [MyArea, QueueAverage, QueueMax, TimeStatistics] = ...
    getActualStatistics (MyArea);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %   Algoritmus řízení vyplněný uživatelem   %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    % Provedeni jednoho kroku simulace a zaslani delek fazi
    Experiment=setActualControl(MyArea, PhasesLengths);

end;

```

Pro demonstraci této zpětné vazby byla vytvořena funkce `demoArea(areaName)`.

4.3 Vytváření reportu o experimentu

Po dokončení experimentu je možné nechat vygenerovat report o experimentu, který obsahuje průběhy snímaných veličin na detektorech po dobu experimentu a další statistická data. Report se vytvoří voláním funkce `uierport 5.7`, kde se zadají soubory obsahující struktury s výsledky experimentů, nebo přímo funkcí `createTEXreport` popsanou v 5.1.

Kapitola 5

Vlastní funkce

V této kapitole je popsána funkce `createTEXreport` 5.1, která automaticky generuje report o proběhlém experimentu v LaTeXovém formátu a další vedlejší funkce, které bylo nutno buď přímo pro tuto funkci, a nebo obecněji pro toolbox vytvořit.

5.1 Vytvoření L^AT_EXového reportu o experimentu

Tato funkce byla vytvořena pro vyhodnocení a zobrazení výsledků proběhlých experimentů. Funkce umožňuje porovnávat mezi sebou výsledky více experimentů na stejné oblasti se stejnými druhy vjezdů, lišící se druhem implementovaného řízení. Funkce vytváří na zadaném místě L^AT_EXový soubor `.tex`.

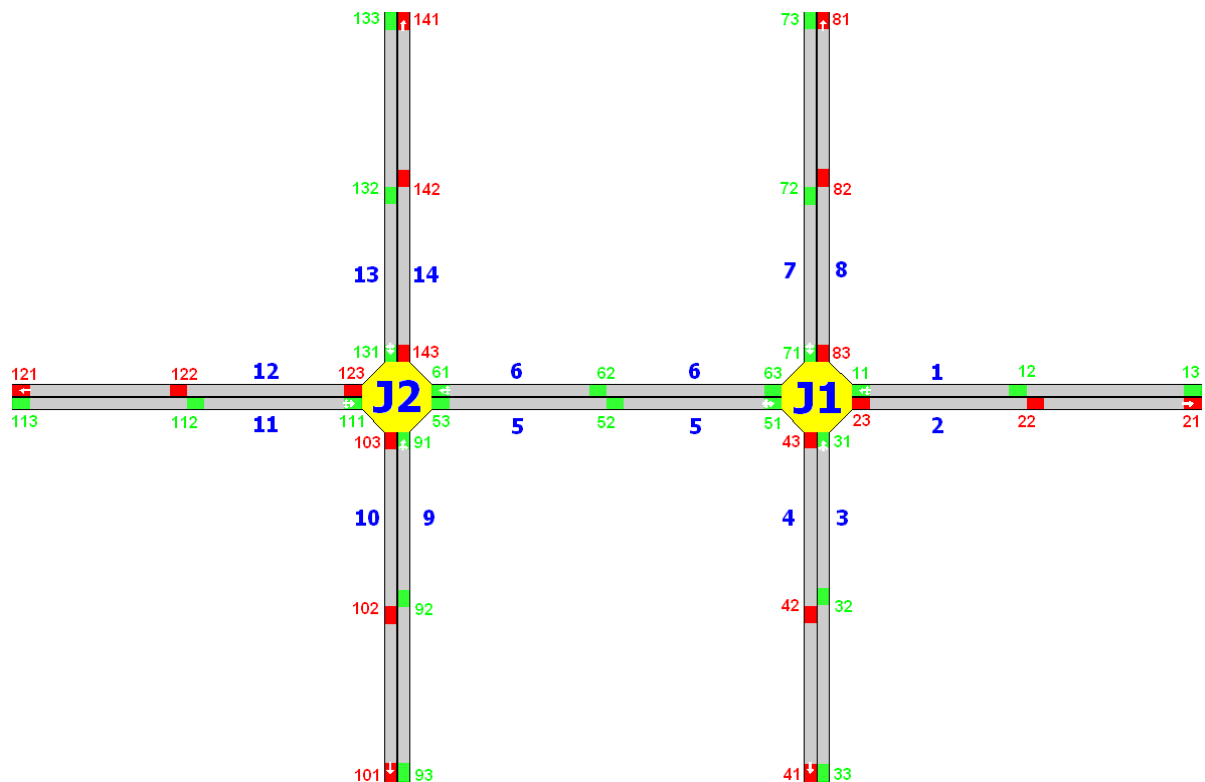
[] = **createTEXReport** (**MyAreas**, **ReportPath**, **Version**)

Popis parametrů:

- **MyAreas** - Vektor se strukturami **MyArea** obsahující informace o experimentu a jeho výsledcích
- **ReportPath** - cesta, kam bude vytvořen adresář "Report" a v něm umístěny zdrojové soubory reportu
- **Version** - verze experimentu

Podrobnější popis funkce:

Univerzální externí soubory pro vytváření reportu jsou umístěny v:



Obrázek 5.1: Příklad znázornění náhledu oblasti "area 2" v reportu

Definice 5.1: ...SVN\AIMSUN-MATLAB\toolboxASYN\areas\report

Externí soubory specifické pro danou oblast jsou umístěny v adresáři s oblastí v adresáři "Source":

Definice 5.2: ...SVN\AIMSUN-MATLAB\toolboxASYN\areas\ (NAZEV_OBL.) \source\

Funkce vytvoří ve výsledném dokumentu úvodní stránku a devět kapitol:

1. Introduction

Pokud se v adresáři se zdrojovými soubory specifickými pro oblast 5.2 nenacházel soubor introduction.tex bude vytvořen a k němu připsán text ze souboru introduction.tex v 5.1.

2. Aims of experiment

Pokud nebyl soubor aims.tex definován v 5.2 bude opět vytvořen. Obsah této kapitoly je ponechán na vyplnění uživatelem.

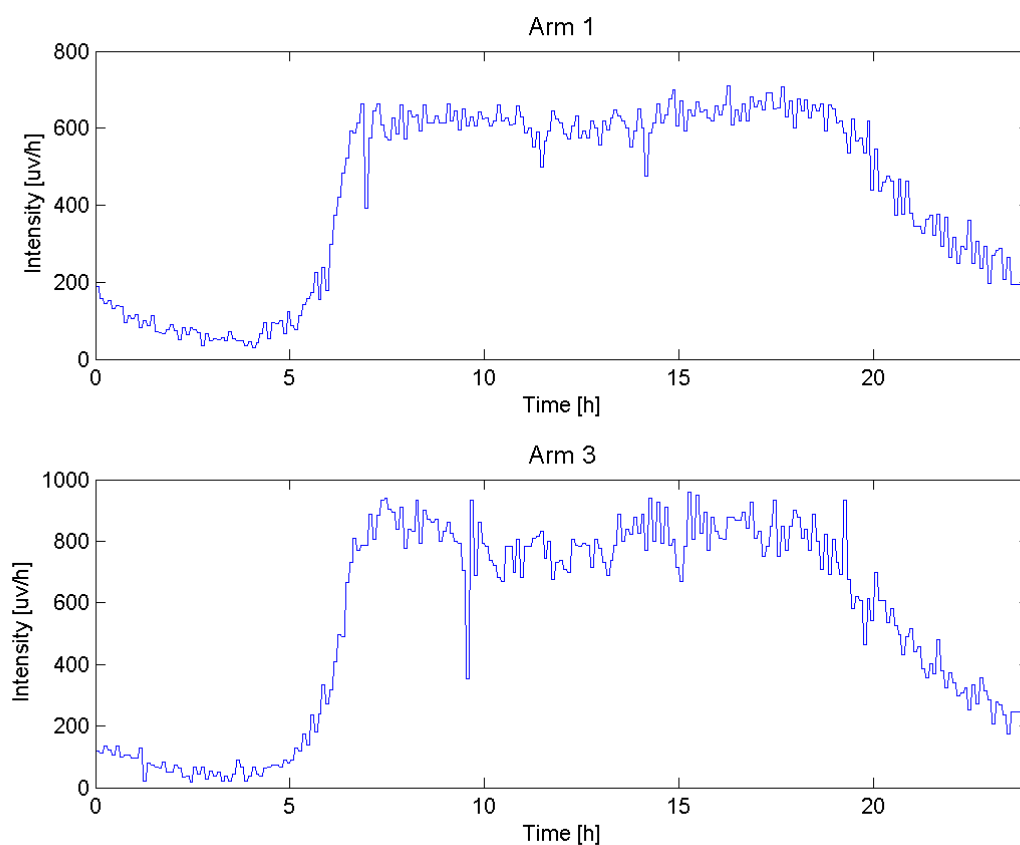
3. Description of Area

V této sekci je popsána oblast daného experimentu. Pokud byl v 5.2 k dispozici soubor NAZEV_OBLASTI.png, je vytvořena podkapitola Network preview s grafickou ukázkou oblasti. Pokud byl v 5.2 k dispozici soubor

”(NAZEV_OBLASTI)_description.tex” je jeho obsah vložen do této kapitoly. Příklad znázornění oblasti ”area 2” viz obr. 5.1.

4. Description of Experiment

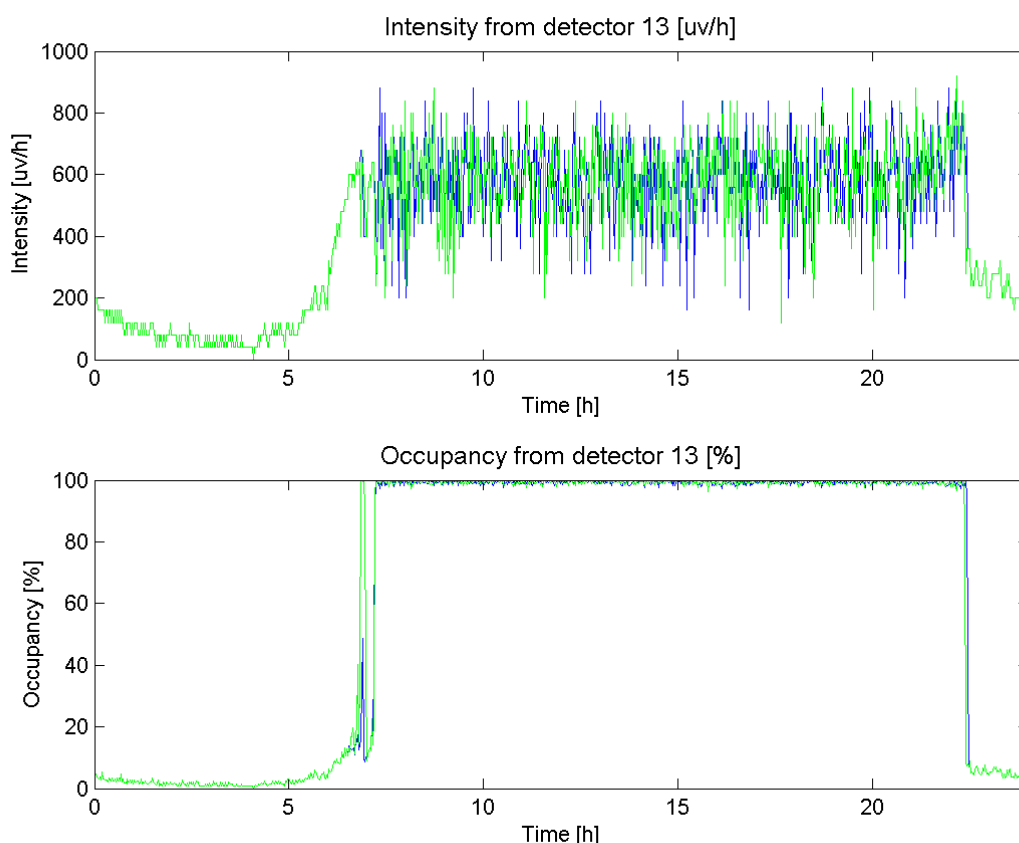
V této kapitole jsou vypsány souhrnné informace o Testované oblasti, o typu vjezdů, délce intervalů detektorů a délce kroku simulace, počet porovnávaných experimentů a popisy k jednotlivým experimentům (Autor, cesta k souboru s výsledky experimentů).



Obrázek 5.2: Příklad vykreslení vjezdů pro ramena 1 a 3 oblasti ”area 2”
v reportu

5. Input data

Tato kapitola již zobrazuje grafy vstupních dat, neboli intenzitu provozu na vstupních ramenech. Předpokládá se, že vstupní data jsou pro všechny experimenty stejná, a proto je zobrazen pouze jeden graf. Kapitola nebude zobrazena v případě generování vjezdů interně AIMSUNem. Na obrázku obr. 5.2 je zobrazen příklad vjezdů pro ramena 1 a 3

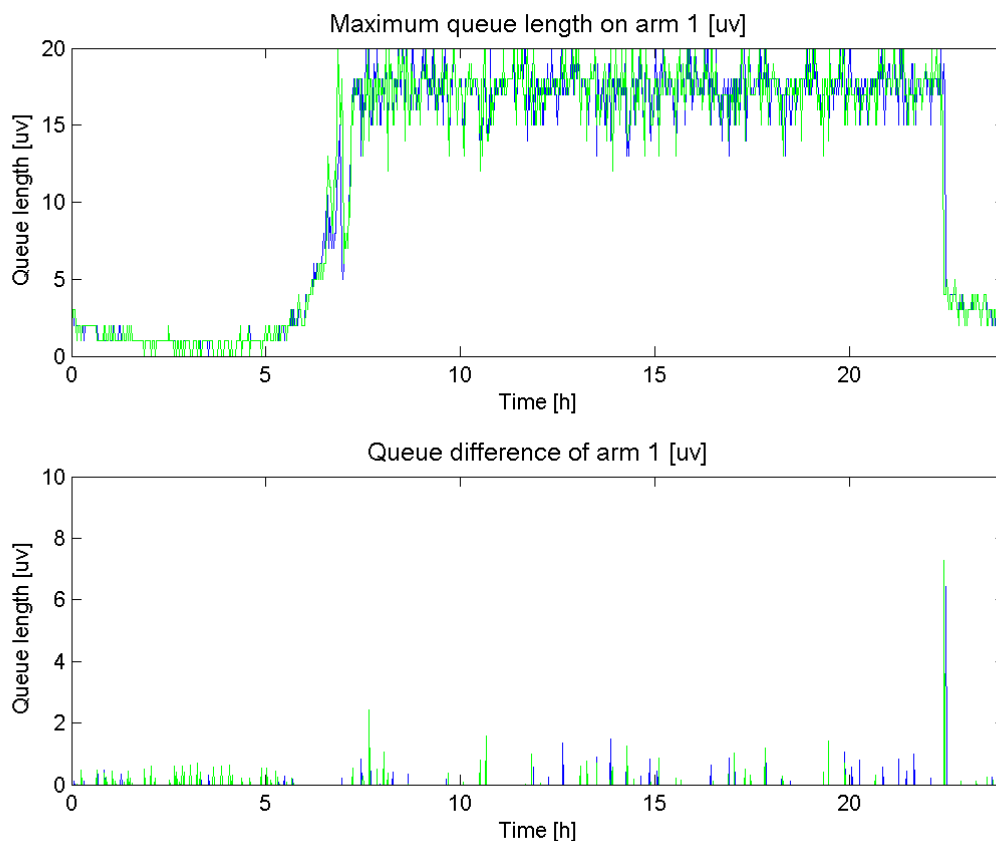


Obrázek 5.3: Příklad vykreslení (intenzity a obsazenosti) z detektoru 13

6. Output data

Zde jsou zobrazena výstupní data z jednotlivých ramen. Jedná se o intenzitu a obsazenost, dále pak maximální délku kolony a její rozdíl od odhadované délky kolony. Tyto charakteristiky jsou zobrazeny nejdříve pro vstupní ramena, pro výstupní ramena nejsou zobrazeny délky kolon. Na obrázku obr. 5.3 jsou zobrazeny příklady vykreslení hodnot intenzit a obsazenosti pro detektor 13 (rameno 1) a dva různé

experimenty. Na obrázku obr. 5.4 jsou zobrazeny průběhy délek kolon a rozdíl od odhadovaných hodnot.



Obrázek 5.4: Příklad vykreslení délek kolon a rozdílu od odhadovaných hodnot z ramene 1

7. Result

- Estimated parameters

V této kapitole jsou zobrazeny výsledky simulace reprezentované konstantami κ , β a λ a střední kvadratická chyba MSE pro jednotlivé experimenty.

- Statistic

Kapitola zobrazuje statistiky získané v době experimentu. Jedná se o hodnoty (Total flow [uv/h], Travel time [s/km], Delay time [s/km], Average speed [km/h], Average space speed [km/h], Density [uv/km], Stop time [s/km],

Number of stops [uv/km], Total intensity [uv/h], Average occupancy [%], Average cycle length [s], Total est. queue length [uv], Total sim. queue length [uv]).

Tyto hodnoty jsou zobrazeny v dalších třech podkapitolách:

– Global statistic

Zde jsou zobrazena jejich minima, maxima, průměrné hodnoty, medián a standardní odchylky.

– Global statistic for spaces of time

Zde jsou zobrazena agregovaná statistická data, jejichž hodnoty jsou zobrazeny pro tříhodinové intervaly.

– Periodical statistic

V této podkapitole je znovu zobrazeno všech třináct údajů, data se ale mění s periodou vzorkování.

8. Conclusion

Soubor "conclusion.tex" bude opět vytvořen, pokud již nebyl nadefinován v 5.2. Text vyplní uživatel.

9. Appendix

Zde jsou ukázány tvary matic obsahující délky fází pro řízení křižovatek, tvary matic obsahující data z detektorů a tvary matic obsahující délky kolon v ramenech.

Poslední kapitolou je závěr obsahující seznam literatury.

Popis algoritmu vytváření T_EXového reportu:

- Při inicializace je zjištěn počet experimentů, ze kterých se vytváří report, jsou vytvořeny konstanty s popisky grafů, vektory se seznamem barev pro vykreslování grafů a do proměnných jsou uloženy cesty k adresáři se zdrojovými soubory k oblasti a k umístění reportu.
- Je vytvořena struktura reportInfo obsahující informace o experimentu a společně se strukturou MyAreas jsou uloženy do souboru v 5.2.
- Z umístění 5.1 a z 5.2 jsou zkopírovány zdrojové soubory pro vytváření reportu.
- Je kontrolováno, jestli jsou v 5.2 soubory introduction.tex, aims.tex, conclusion.tex a bibliography.tex. Pokud ne, jsou vytvořeny a do nich vepsán Předefinovaný text.
- Je vytvořen soubor report.tex, ve kterém je vytvořena hlavička. Jsou nadefinovány použité balíčky a je vytvořena úvodní stránka. V souboru jsou postupně vytvářeny

kapitoly a k nim přidávány dříve vytvořené vstupní soubory s daným textem (Contentst, Introduction, Aims of experiment,). Způsob vytváření kapitoly "Description of area" byl popsán výše.

- Kapitola Description of Experiment je vytvořena pomocí informací ze struktur MyArea.
- Kapitola Input data je vytvořena, pokud nebyly vjezdy generovány AIMSUNem. V případě offline generovaných vjezdů je matice s vjezdy ze struktury "MyArea.experiment.entrances" upravena jen pro čas experimentu (hodnoty mimo tento čas nebudou zobrazeny). Vjezdy pro vstupní ramena jsou postupně vykreslovány do grafů a ukládány jako obrázky do 5.2. Pro vkládání grafů do T_EXových souborů je použita zde i dále funkce fwriteTEXfigure 5.3.
- V kapitole output data jsou zobrazeny délky kolon (struktura MyAreas.experiment.queues), tato matice je převzorkovaná pomocí funkce resampleData 5.4 podle délky simulačního cyklu. Dále jsou zobrazeny data z detektorů (MyAreas.experiment.detectors). Do jednoho grafu jsou zobrazeny průběhy ze všech experimentů pro schodné rameno a stejnou veličinu (obsazenost, intenzita). Pro výstupní ramena nejsou zobrazeny délky kolon.
- V kapitole "Results" v podkapitole "Estimated parameters" jsou zobrazeny v tabulkách, během experimentu vypočtené hodnoty konstant κ , (β) , λ a MSE (struktury MyArea.kappa, MyArea.beta, MyArea.lambda, MyArea.MSE a MyArea.kappa2, MyArea.lambda2, MyArea.MSE2) pro jednotlivá vstupní ramena a pro jednotlivé experimenty. Zde je využito funkce fwriteTEXtable.
- V kapitole "Results" v podkapitole "Estimated parameters" jsou zobrazeny statistické hodnoty ze struktur "MyArea.experiment.periodicalStatistics" (ze kterých jsou vypočtena minima, maxima, průměr, medián a standardní odchylka) a "MyAreas.experiment.experimentStatistics". V podkapitole Global statistic for spaces of time jsou tato data zobrazena do sloupcového grafu po 3-hodinových intervalech a v podkapitole jsou všechna data vykreslena Periodical statistic. Data jsou opět vykreslována v cyklu pro všechny experimenty.
- Je připojen soubor obsahující závěr a vytvořena příloha demonstující tvar matic s délkami fází, dat z detektorů a statistických dat. Pro vykreslení demonstračních matic do TeXového kódu byla využita funkce fwriteTEXmatrix.

5.2 Výpis dostupných oblastí pro experimenty

Funkce proskenuje adresář ".../toolboxASYN/areas" na soubory typu *.sce, zobrazí tento seznam bez koncovek a ze souborů *.mat (které se jmenují stejně jako příslušný *.sce soubor) načte informaci z "areaName" a zobrazí ji do seznamu. To samé provede pro *.mat soubory obsahující data vjezdů k daným oblastem. Název oblasti je totožný s názvem adresáře. Shodné názvy mají také hlavní soubory popisující oblast. Soubory *.sce jsou Aimsun scénáře (např. area2.sce) a *.mat jsou soubory obsahující Matlabovské struktury s informacemi o oblastech nebo vjezdech (např. area2.mat, vehicles.mat).

```
[ ] = showAreaList();
```

Volání dané funkce v Matlabu způsobí následující výpis:

```
>> showAreaList();
```

List of supported traffic areas and entrances:

```
area1          ( Area1 (2 junctions with 3 arms), Proving Ground )
  vehicles      ( 100% cars )

area2          ( Area2 (2 junctions with 4 arms), Proving Ground )
  buses         ( 90% cars, 10% buses )
  realtime      ( 90% cars, 10% buses, ONLY FOR R.TM GEN. MODE!!! )
  vehicles      ( 100% cars )

area3          ( Area3 (1 junction with 4 arms), Proving Ground )
  buses         ( 90% cars, 10% buses )
  realtime      ( 90% cars, 10% buses, ONLY FOR R.TM GEN. MODE!!! )
  vehicles      ( 100% cars )

smichov        ( Smichov, Prague )
  buses         ( 90% cars, 10% buses )
  vehicles      ( 100% cars )

zlicin         ( Zlicin, Prague )
```

Algoritmus funkce pracuje na následujícím principu:

- Funkce proskenuje adresář, ve kterém je sama spuštěna (ve stejném adresáři se nachází adresáře s oblastmi) a vyhledá v něm všechny adresáře.
- Vyhledané adresáře proskenuje, zda neobsahují stejnojmenné *.sce soubory a *.mat soubory. Pokud ano, jedná se o platné oblasti pro experiment a vypíše její název s daným popisem z položky areaName ve struktuře oblasti.
- Po vyhledání adresáře s oblastí je ještě proskenován, neobsahuje-li *.mat soubory s nadefinovanými vjezdy. V případě, že obsahuje, stejným způsobem vypíše názvy vjezdů a k nim dané popisy.

5.3 Uložení figur z MATLABu do L^AT_EXového formátu

Tato funkce má na starosti export figur z Matlabu do souboru TEX. O vykreslení figur se stará uživatel a této funkci pouze zadá ID těch figur (Figures), které chce uložit. Obecně lze exportovat několik figur vedle sebe i pod sebe. Jaké budou mít uspořádání, je určeno formátem matice (vektoru), ve kterém se ID zadají. Např. Figures=[1 2;3 4] zobrazí celkem 4 figury v matici 2x2 (v prvním "řádku" budou vedle sebe figury 1 a 2 a v druhém 3 a 4). V položce Description zadá popisky (číslování obrázků si TEX provádí sám).

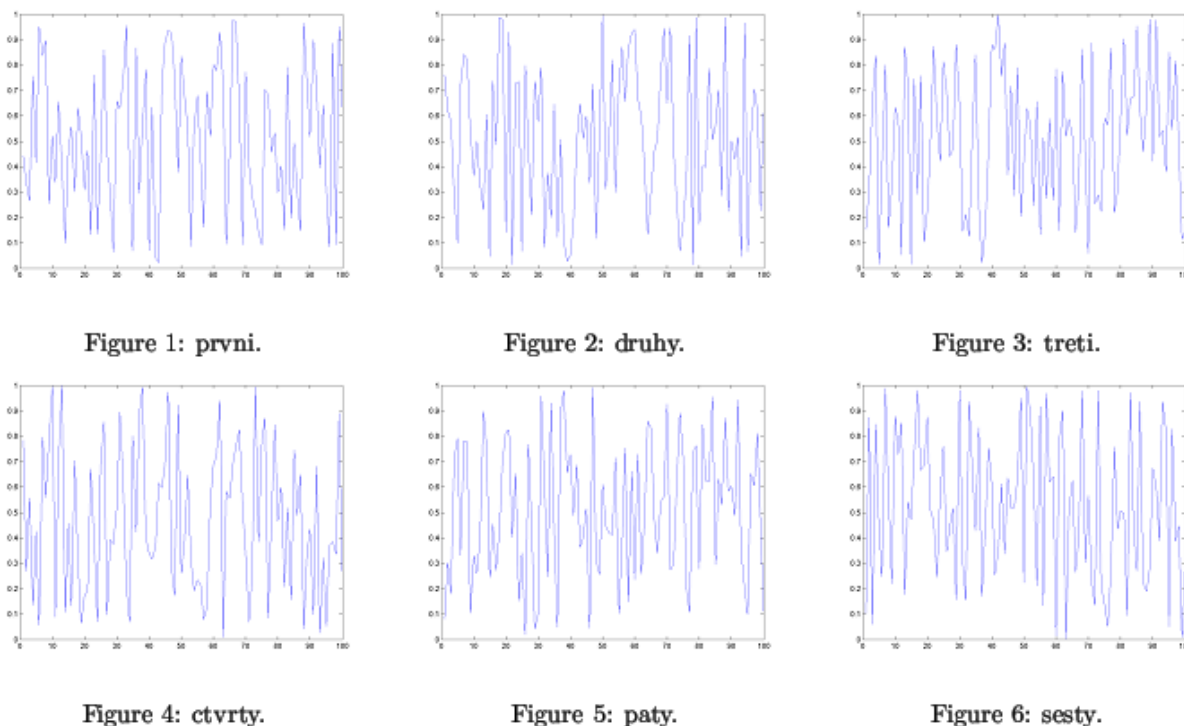
[] = **fwriteTEXfigure**(**TexFile**, **Figures**, **Descriptions**, **Prefix**, **Postfix**, **FigSize**)

Popis parametrů:

- TexFile - File descriptor *.tex souboru otevřeného pro zápis
- Figures - matice s id figur, ve formátu, ve kterém bude zobrazena v .tex souboru
- Descriptions - matice cell objektů obsahující textové popisky k daným figurám
- Prefix - předpona před názvem souboru s obrázkem
- Postfix - přípona za názvem souboru s obrázkem
- FigSize - matice s rozměry jedné figury ve výstupním .tex souboru v cm

Funkce funguje v pěti režimech popisek (jiné je ukončeno chybou):

1. Description bude cell matice o shodném rozměru jako matice s ID figur (Figures), každá figura pak bude mít svou popisku viz. obr. 5.5.



Obrázek 5.5: Příklad pro zobrazení popisku ke každé figurě

2. Description bude mít jeden sloupec a stejný počet řádků jako má matice Figures, a potom tedy bude jedna popiska pro všechny figury v jednom řádku a ta bude uprostřed řádku viz. obr. 5.6.
3. Description bude mít jeden řádek a stejný počet sloupců jako má matice Figures, a potom tedy bude jedna popiska pro všechny figury v jednom sloupci a ta popiska bude pod poslední figurou v daném sloupci viz. obr. 5.7.
4. Description bude pouze jeden řetězec (kombinace 2. a 3.), a pak bude pod všemi figurami (společný pro všechny)
5. Description bude prázdný a pak se nebude příslušný TEX kód generovat

Algoritmus funkce pracuje na následujícím principu:

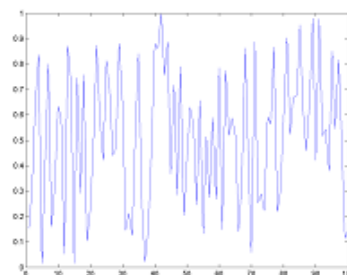
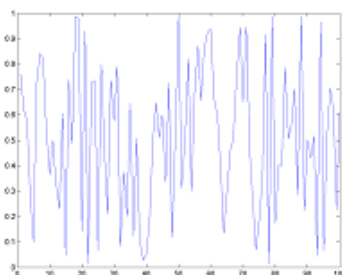
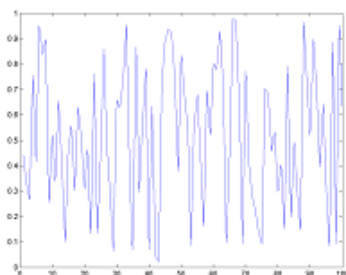


Figure 1: první.

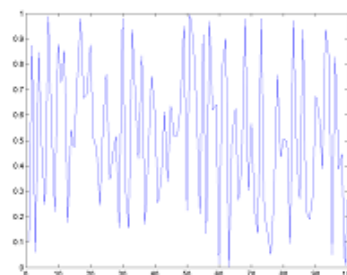
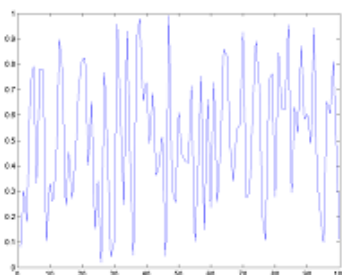
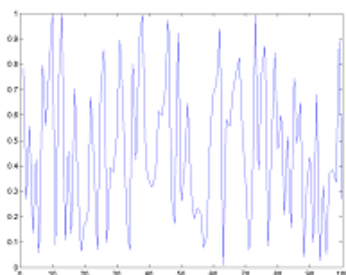


Figure 2: druhy.

Obrázek 5.6: Příklad pro zobrazení jednoho popisku pro řádek

- Jako defaultní typ obrázku je zvolen PNG.
- Funkce prochází po řádcích matice `Figures` a podle zapsaného id ukládá figury do souboru. Název souboru je složen z "prefix" + "id číslo figury" + "Postfix" + ".png".
- Do výstupního .tex souboru je zapsána část TEX kódu, vytvářející "minipage" o rozměrech vypočtených tak, aby se do řádku vešel vedle sebe odpovídající počet "minipage" dle matice `Figures`.
- Dále jsou řešeny popisky obrázků. Pokud je rozměr matice "Description" shodný s rozměrem matice "Figures", proběhne cyklus, který po vykreslení každého řádku s grafikou zapíše do výstupního .tex souboru další řádek s příslušným počtem "minipage" obsahující popisky obázků. Pokud je "Description" sloupcový vektor, je vykreslena pouze jedna "minipage" obsahující pouze jeden popis pro celý řádek s grafikou. V případě, že "Description" obsahuje pouze řetězec obsahující jeden společný popis pro všechny figury, zapíše se do výstupního .tex souboru nejdříve všechny řádky s figurami, a potom teprve jeden společný popis. V případě, že je

”Description” pouze řádkový vektor odpovídající svojí délkou délce matice ”Figures”, je opět volán tento cyklus po vykreslení všech figur.

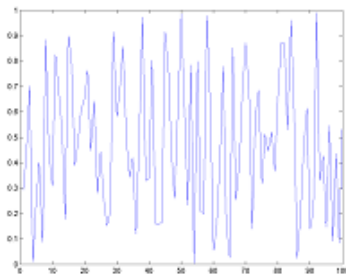


Figure 1: první.

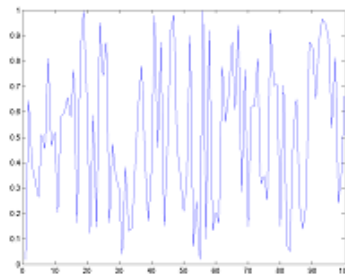


Figure 2: druhý.

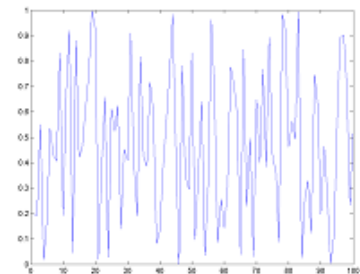


Figure 3: třetí.

Obrázek 5.7: Příklad pro zobrazení jednoho popisku pro sloupec

5.4 Převzorkování dat v matici

Funkce je vytvořena pro dvě matice s daty (např. A a B), kde každá je vzorkovaná s jinou periodou. Druhou matici (B) převzorkuje např. na matici C. Matice C je dána se stejnou periodou, jako matice první (A). Informace o časech (periodách), ze kterých data pocházejí, jsou obsazeny v prvních dvou řádcích (první řádek den a druhý řádek sekundy od začátku simulace, které stále narůstají, tzn. nenulují se pro každý nový den) matice A a B a zbylé řádky jsou vlastní data. Matice B se může skládat z několika submatic B pod sebou (po konstantním počtu řádků se opakují dva řádky s časovými údaji).

```
[C] = resampleData (A, B, subrows, switch);
```

Popis parametrů:

- A - Matice obsahující řádky s časovými údaji, podle kterých se bude převzorkovávat
- B - Matice obsahující data, tato matice bude převzorkována
- subrows - Počet řádků jedné submatice v matici B
- switch - přepínač pro určení hodnot v matici C odpovídající časům v matici A, které jsou mimo časový rozsah matice B
 - 0 - Hodnoty v matici C nahradí její poslední hodnotou.
 - 1 - Hodnoty v matici C nahradí hodnotou NaN.

Zde je ukázán výstup funkce, když matice A začíná v menším čase a její poslední čas je větší než poslední v matici B.

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 100 & 180 & 220 & 250 \\ 5 & 2 & 17 & 4 & 7 \\ 80 & 24 & 17 & 36 & 29 \end{bmatrix} = \begin{bmatrix} den \\ sekunda \\ veličina1 \\ veličina2 \end{bmatrix} \quad (5.1)$$

$$B = \begin{bmatrix} 1 & 1 & 1 \\ 100 & 190 & 240 \\ 56 & 76 & 97 \\ 1 & 1 & 1 \\ 100 & 190 & 240 \\ 11 & 23 & 54 \end{bmatrix} = \begin{bmatrix} den \\ sekunda \\ veličina1 \\ den \\ sekunda \\ veličina2 \end{bmatrix} \quad (5.2)$$

Funkce byla volána s přepínačem switch = 1, subrows bylo nastaveno na 3.

$$C = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 100 & 180 & 220 & 250 \\ NaN & 56 & 74 & 89 & NaN \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 100 & 180 & 220 & 250 \\ NaN & 11 & 22 & 42 & NaN \end{bmatrix} \quad (5.3)$$

Zde byla funkce volána s přepínačem `switch = 0`, `subrows` bylo nastaveno opět na 3.

$$C = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 100 & 180 & 220 & 250 \\ 56 & 56 & 74 & 89 & 89 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 100 & 180 & 220 & 250 \\ 11 & 11 & 22 & 42 & 42 \end{bmatrix} \quad (5.4)$$

Popis algoritmu:

- Po kontrole, jestli nejsou matice shodné nebo nulové, je vytvořena nulová matice, obsahující stejný počet sloupců jako matice A a stejný počet řádků jako matice B.
- V případě, že je matice B tvořena z určitého počtu submatic, jsou do matice C na příslušná místa doplněny řádky, obsahující informace o časech.
- Pro výpočet lineární interpolace je využita Matlabovská funkce `interp1`, která provádí výše zmíněné nahrazování hodnot hodnotou NaN.
- V případě nastavení přepínače `switch = 0`, jsou hodnoty NaN v matici C vyhledány a nahrazeny posledními číselnými hodnotami.

5.5 Výběr více souborů z různých umístění

Funkce byla vytvořena za účelem zadávání více *.mat souborů s výsledky experimentů a následné vytvoření reportu o těchto experimentech. Tato funkce nicméně funguje obecně a lze ji tedy použít pro hromadné vybírání jakéhokoli typu souborů. Je proto volaná z funkce `uireport` 5.7. Po zavolání funkce se otevře dialogové okno.

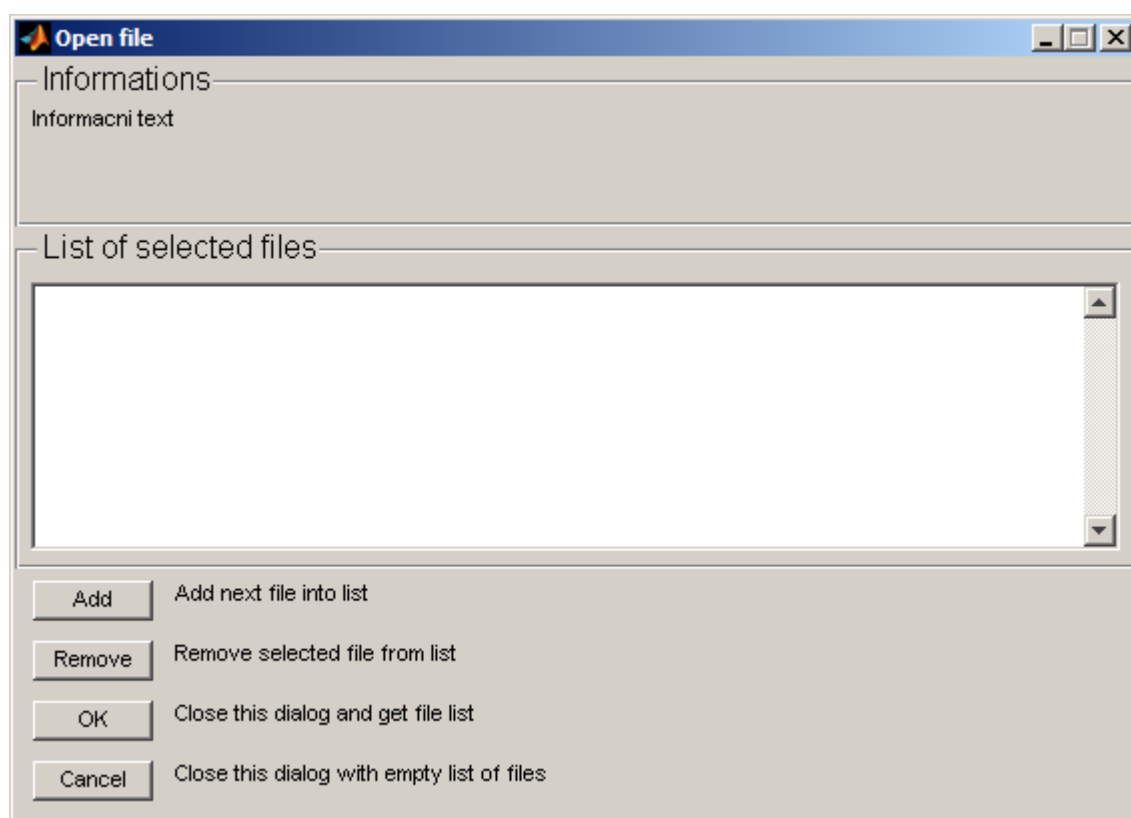
**[FileList] = uigetfilelist (StartPath, FileMask, FigureTitle,
FigureInformations)**

Popis parametrů:

- `FileList` - Řádkový cell vektor obsahující cesty ke všem vybraným souborům
- `StartPath` - Výchozí cesta, kam bude nastaveno dialogové okno `uigetfile` obr. 5.9. Jako defaultní hodnota je nastavena cesta do workspace MATLABu.

- FileMask - sloupcový cell vektor, obsahující masky k zobrazovaným souborům. Může být také matice o dvou sloupcích, obsahující v prvním sloupci masky a ve druhém informativní popisy k typům souborů.
- FigureTitle - Řetězec obsahující titulek dialogového okna
- FigureInformations - Řetězec obsahující informační text zobrazený v dialogovém okně v panelu Informations

Funkce lze spustit i bez vstupních parametrů. Volání funkce se zadanými všemi parametry



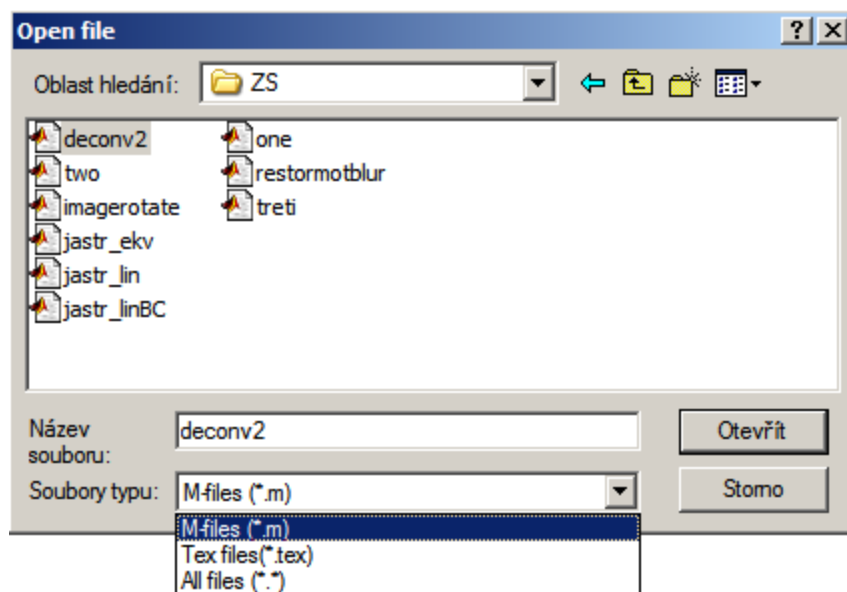
Obrázek 5.8: Dialogové okno vyvolané funkcí `uigetfilelist`

vyvolá toto dialogové okno obr. 5.8:

```
file_list=uigetfilelist('C:\',{ '*.*', 'M-files (*.m)';...
                        '*.tex', 'Tex files (*.tex)';...
                        '*.*', 'All files (*.*)' },...
                        'Open file', 'Informacni text')
```

Dialogové okno obsahuje čtyři tlačítka:

- Add - Vyvolá dialogové okno funkce `uigetfile` obr. 5.9.
- Remove - Odstraní jednu označenou položku z listu obr. 5.10.
- OK - Zavře dialogové okno a nastaví výstupní proměnnou funkce jako cell vektor, obsahující cesty všech souborů v listu.
- Cance - Zavře dialog a vrátí prázdný "cell" vektor.



Obrázek 5.9: Dialogové okno funkce `getfilelist` vyvolané po stisknutí tlačítka Add

Popis algoritmu funkce se dá rozdělit do dvou částí. V první je vytvořeno dialogové okno a ve druhé je ošetřena obsluha stisknutí tlačítek. Tomuto programovacímu stylu se říká "Switched Board Programming" a je popsán na příklad v [13, strana 10].

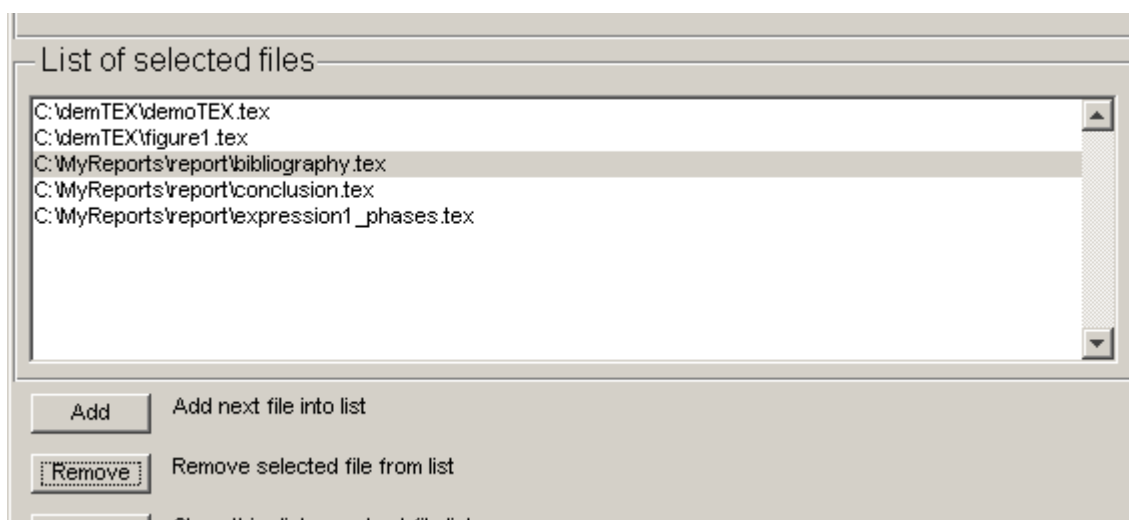
Vytvoření dialogového okna:

- Postupně jsou vytvářeny jednotlivé grafické objekty. U tlačítek je zaregistrována 'Callback' funkce pro obsluhu událostí. Po vytvoření všech objektů je funkce úspěšná pomocí funkce `uiwait`.

Obsluha událostí:

- Při události grafického objektu (např. stisknutí tlačítka) zavolá funkce sama sebe s jedním parametrem pomocí něhož funkce zjistí, že jde o obsluhu události a její druh.

- Při přidávání nových položek do listu musí být data z `uigetfile` dialogu vhodně upravena do tvaru pro list.
- Při odebírání položky z listu jsou data načtena do matice, ve které je následně daný řádek vynechán a po vhodné úpravě data opět zaslána zpět do listu.
- Stisk tlačítka OK spustí uspanou funkci vytváření dialogového okna. Funkce načte data z listu a pošle je na výstup buď jako řádkový cell vektor obsahující úplné cesty k souborům, nebo v případě, že byl vybrán jen jeden soubor, jako řetězec obsahující tuto cestu (z důvodu ekvivalence s funkcí `uigetfile`). V případě, že nebyl vybrán žádný soubor, zachová se funkce stejně jako při stisku tlačítka `Cancle`.



Obrázek 5.10: Odstranění položky ze seznamu

5.6 Grafické rozhraní inicializace oblasti

Pomocí této funkce je umožněna uživatelsky příjemnější inicializace experimentu pomocí dialogového okna (rozšiřuje funkci `initArea` 4.1.1). Po spuštění funkce se zobrazí dialogové okno obr. 5.11. V panelu `Area` je obsažen list obsahující seznam všech dostupných oblastí pro experiment. Při označení oblasti se zobrazí v dolní části panelu krátký popis. V panelu `Vehicles` je obsažen panel `Generator`, ve kterém je možno vybírat ze tří typů generování vjezdů při experimentu. Volba `Aimsun` inicializuje oblast s vjezdy

generovanými interně Aimsunem. Volba offline použije předem uloženou matici vjezdů. Volba realtime je volba pro generování vjezdů v každém kroku simulace. V listu níže se pro dané volby zobrazují dostupné soubory, obsahující dané struktury s vjezdy a k nim stručný popis. Třetím panelem je TeX report. Tento panel je pro určení informací o vytvářeném reportu. Defaultně se report o experimentu nevytváří. Vytváření reportu se nastaví až po zvolení cesty, kam se bude report ukládat, stiskem tlačítka "Browse ...". Po nastavení cesty se zpřístupní okna pro vyplnění autora a krátkého popisu experimentu obr. 5.12. Pokud jedno ze dvou oken nebude vyplněno, bude hodnota ve výsledné struktuře nahrazena řetězcem "Unknown".

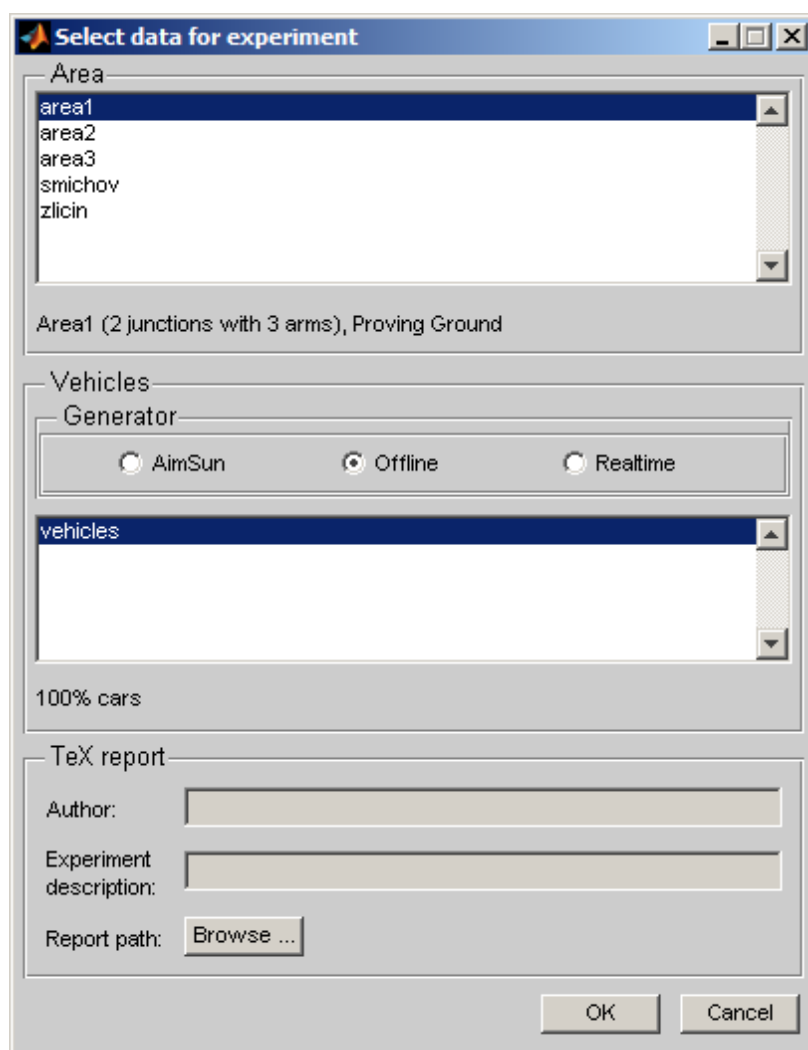
[MyArea] = uiinitarea()

Popis parametrů:

- MyArea - struktura obsahující informace o experimentu.

Popis algoritmu lze stejně jako o funkce `uigetfilelist` 5.5 rozdělit na dvě části. V první se vytvoří grafické objekty a registrují se obsluhy událostí. V druhé jsou už konkrétní obsluhy událostí naprogramovány:

- Po vytvoření grafických objektů je do listu v panelu Area načten pomocí algoritmu popsaného v 5.2 seznam dostupných oblastí.
- Při kliknutí do list s oblastmi je zjištěn druh vybraného generátoru vstupů a název vybrané oblasti. Pokud pro vybranou oblast a vybraný generátor vstupů existuje odpovídající *.mat soubor s odpovídající strukturou, je vypsán do listu v panelu vehicles. Pokud byla jako druh generátoru vjezdů vybrána volba "Aimsun", je tato volba vypsána i do listu.
- Pro volbu generátoru "Realtime" se zobrazují i "offline" vjezdy, protože pro realtime generování vjezdů je potřeba jen informace o poměru zastoupení automobilů a autobusů.
- Pole pro vyplnění autora a popisu experimentu jsou nastavena na disable. Pokud je pomocí tlačítka "Browse ..." nastavena cesta, tato pole se zpřístupní a zvolená cesta je vypsána vedle tlačítka.
- Po stisknutí tlačítka OK je načtena vybraná oblast, druh vjezdu, název vjezdu, cesta pro uložení reportu, jeho popisek a jeho autor. tyto parametry jsou předány funkci `initArea` 4.1.1. Výstup z funkce `initArea` je předán i na výstup funkce `uiinitarea`.



Obrázek 5.11: Dialogové okno funkce uiinitarea

5.7 Grafické rozhraní vytváření reportu

Tato funkce slouží k vytvoření reportu o experimentu a využívá k tomu dostupných výše popsaných funkcí.

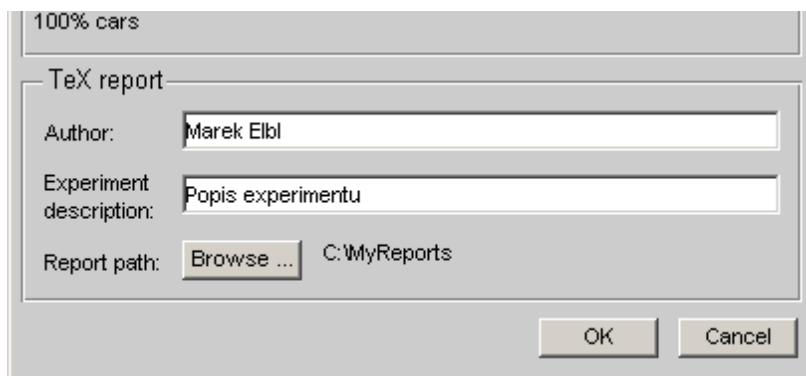
[FileList] = uireport (indir)

popis parametrů:

- indir - řetězec obsahující cestu k místu se zdrojovými daty z experimentů.

popis algoritmu funkce:

- Nejdříve je zavolána funkce uigetfilelist 5.5 pro zjištění cest k souborům, obsahující struktury s výsledky experimentů.



Obrázek 5.12: Dialogové okno funkce `uiinitarea` (umístění a informace o reportu)

- Jsou načteny struktury `MyArea` do workspace a z nich je vytvořena struktura `MaAreas`.
- po zavolání `uigetdir` je zjištěna cesta pro vytvoření výstupního `.tex` souboru s reportem.
- Na závěr je zavolána funkce `createTEXreport 5.1`, které je předána struktura `MyAreas` a řetězec s cestou k výstupnímu souboru.

Kapitola 6

Závěr

Tato bakalářská práce popisuje simulace dopravních oblastí a experimentování na nich s různými algoritmy řízení. V našem případě s využitím softwaru GETRAM. Algoritmy řízení jsou implementovány v MATLABu s využitím Aimsun - MATLAB toolboxu, který je v současné době vyvíjen na ÚTIA AV ČR. Jedním z hlavních cílů této práce bylo vytvoření systému, který bude automaticky generovat výsledný report o proběhlém experimentu.

Po seznámení se s problematikou simulací dopravních oblastí a souvisejících pojmů, byly popsány kroky nezbytné pro sestavení funkčního experimentu. V první fázi byl návrh oblasti v editoru TEDI, zkompletování simulace v simulátoru AIMSUN a konečné vytvoření souboru *.sce, který obsahuje potřebné informace o experimentu a se kterým spolupracuje Aimsun - MATLAB toolbox. Obě aplikace jsou součástí balíku GETRAM. V druhé fázi byl popsán Aimsun - MATLAB toolbox, jeho funkce a spolupráce s GETRAMem, se zaměřením na vytvoření a spuštění experimentu s různým řízením.

V průběhu práce byla vytvořena a postupně upravována a rozšiřována matlabovská funkce **createTEXreport**, zobrazující výsledky experimentů v oblasti jako dokument v L^AT_EXovém formátu. Dokument je rozdělen na několik kapitol. Pokud jsou dostupná všechna zdrojová data pro vytváření reportu, jsou součástí dokumentu souhrnné informace o oblasti na níž je prováděn experiment, dále v grafech zobrazené průběhy intenzit a obsazeností na definovaných ramenech a statistická data o experimentu. Statistická data jsou jednak konstanty κ , λ , β a statistické veličiny popisujících různými způsoby pohyb aut viz 5.1.

Funkce createTEXreport využívá k vytvoření výsledného reportu data, která byla v průběhu experimentu postupně ukládána do objektu MyArea. Vstupním parametrem funkce nemusí být jen jeden objekt MyArea s oblastí, ale může být zadán vektor ob-

sahující více objektů MyArea pro sice stejnou oblast, ale pro různé algoritmy řízení. Funkce následně provede zobrazení sobě odpovídajících hodnot dvou experimentů do jednoho grafu. Takto vygenerovaný report slouží tedy nejen k přehlednému zobrazení informací o proběhlém experimentu, ale také k názornému porovnání dvou algoritmů řízení, a tedy k určení toho, který ze způsobů řízení je pro danou oblast vhodnější.

Spolu s createTEXreport byly zároveň vytvořeny další funkce pro Aimsun - MATLAB toolbox, které buď createTEXreport využívá, nebo tento toolbox rozšiřují. Jednalo se o funkci **showAreaList** zobrazující v příkazovém řádku programu MATLAB dostupné oblasti a vjezdy, dále funkci **fwriteTEXfigure** ukládající dle potřeby MATLABovské figury do výstupního TEXového souboru, funkci **resampleData** soužící k převzorkování jedné matice s daty podle matice s jinou periodou vzorkování. Toolbox byl rozšířen o grafická uživatelská prostředí. Pro inicializaci oblasti před začátkem experimentu funkce **uiinitarea** a pro hromadné zadávání souborů k vytvoření reportu o experimentu **uigetfilelist** resp. **uireport**.

Tímto se podařilo splnit všechny body zadání. Nicméně práce na vývoji tohoto toolboxu nekončí a jeví se zde možnost dalšího pokračování. Další možností v pokračování práce je vytvoření vlatního, uživatelsky přívětivějšího editoru, který by nahradil program TEDI a kde by otevřenost kódu byla jistě výhodou.

Literatura

- [1] Transport Simulation Systems: *AIMSUN Version 4.2 - User manual*.
TSS, Spain, 2004.
- [2] Transport Simulation Systems: *TEDI Version 4.2 - User manual*.
TSS, Spain, 2004.
- [3] Transport Simulation Systems: *GETRAM Extension Version 4.2 - User manual*.
TSS, Spain, 2004.
- [4] Transport Simulation Systems: *GETTING STARTED Version 4.2 - User manual*.
TSS, Spain, 2004.
- [5] Gebouský P.: *AIMSUN-MATLAB Interface - User's Guide*.
Academy of Sciences of the Czech Republic - Institute of Information Theory and Automation, Prague, 2006.
- [6] Dohnal P.: *Matlab Traffic Toolbox*.
Slovak University of Technology in Bratislava, Bratislava, 2007.
- [7] Nagy I., Homolová J. , Pecherková P.: *Dopravně závislé řízení silničního provozu ve městech*.
Academy of Sciences of the Czech Republic - Institute of Information Theory and Automation, Prague, 2007.
- [8] Nagy I. : *AIMSUN - Praktický přehled mikrosimulačního software*.
http://www.fd.cvut.cz/departament/k611/PEDAGOG/K611TH0_soubory/3_AIMSUN.pdf.
- [9] Eltodo: *Modelování dopravy*.
<http://sim.eltodo.cz/teorie.html>.

- [10] Eltodo: *Bezpečnost chodců*.
http://www.eltodo.cz/Webova_prezentace/2_Doprava/Bezpecnost_chodcu.pdf.
- [11] DVH: *GETRAM (AIMSUN/TEDI)*.
<http://www.dhv.cz/article.asp?id=125>.
- [12] Ken Fox: *Introduction to Micro-simulation*.
<http://www.microsimulation.drfox.org.uk/intro.html>.
- [13] Doňar B., Zaplatílek K.: *MATLAB - tvorba uživatelských aplikací*.
BEN - Technická literatura, Praha, 2004.
- [14] Doňar B., Zaplatílek K.: *MATLAB pro začátečníky*.
BEN - Technická literatura, Praha, 2004.

Příloha A

Obsah přiloženého CD

CD přiložené k této práci obsahuje elektronickou verzi této práce a příklad reportu pro oblast Smíchov ve formátu PDF, elektronicky dostupnou literaturu, video-ukázky simulací, zdrojové kódy vytvořených funkcí a poslední verzi Aimsun - MATLAB toolboxu s manuály.

Příloha B

Ukázka L^AT_EXového reportu o experimentu

Ná následujících stránkách této přílohy je uveden příklad automaticky vygenerovaného L^AT_EXového reportu o experimentu na oblasti Smíchov pro dva různé algoritmy řízení.



Department of Adaptive Systems
ÚTIA, AV ČR
P.O.Box 18, 182 08 Prague 8, Czech Republic



RESEARCH REPORT

Marek Elbl (elblm1@fel.cvut.cz)

MATLAB-AIMSUN experiment

Smichov, Prague

Version 0.0.245

August 8, 2007

MDČR 1F43A/003/120, MŠMT 1M0572

Contents

1	Introduction	4
2	Aims of experiments	5
3	Description of Area	6
3.1	Network preview	6
3.2	Network design and notation	6
4	Description of Experiments	7
4.1	Experiment 1: Dynamic Control with Kalman Filter	7
4.2	Experiment 2: Static Control without Feedback	7
5	Input data	8
5.1	Arms 4, 40	8
5.2	Arms 3, 5	8
5.3	Arms 16, 21	9
5.4	Arms 6, 33	9
5.5	Arms 36, 31	10
5.6	Arms 28	10
6	Output data	11
6.1	Input arms	11
6.1.1	Arm 1	11
6.1.2	Arm 2	12
6.1.3	Arm 5	13
6.1.4	Arm 6	14
6.1.5	Arm 8	15
6.1.6	Arm 9	16
6.1.7	Arm 10	17
6.1.8	Arm 13	18
6.1.9	Arm 14	19
6.1.10	Arm 16	20
6.1.11	Arm 21	21
6.2	Output arms	22
6.2.1	Arm 7	22
6.2.2	Arm 10	22
6.2.3	Arm 11	23
6.2.4	Arm 13	23
6.2.5	Arm 14	24
6.2.6	Arm 15	24
6.2.7	Arm 23	25
6.2.8	Arm 24	25
6.2.9	Arm 32	26
6.2.10	Arm 37	26
7	Results	27
7.1	Estimated parameters	27
7.1.1	Parameters κ and λ	27
7.1.2	Parameters κ , β and λ	28
7.2	Statistics	29
7.2.1	Global statistics	29
7.2.2	Global statistics for spaces of time	30
7.2.3	Periodical statistics	34
8	Conclusion	38

9	Appendix	39
9.1	Structures	39
9.1.1	Control matrix	39
9.1.2	Data vectors	39
9.1.3	Statistics vectors	39

1 Introduction

This report contains results from 2 traffic experiments performed in the MATLAB-AIMSUN Toolbox environment [5].

This toolbox is to implement an interface [4] between MATLAB environment and AIMSUN simulator for the purpose of testing of designed traffic control of special macroscopic type. The AIMSUN simulator is powerful tool for simulation of traffic flow while the MATLAB environment provides the rich set of functions that simplify the process of design, optimization and testing of developed traffic control. Toolbox combines together advantages of both of them.

All the actual last data (periodical and global system statistics useful for evaluation of control schemes applied) are gathered in buffer available from MATLAB and stored in their history. Besides detection sampling, statistics sampling and junction sampling intervals can be various in general. Also precise data about applied junction phase timing is gathered and can be gained. Toolbox work in more general asynchronous manner. If external control of junction phases length is used then setting received from MATLAB is not applied immediately but each junction waits for beginning of new control cycle. So setting is applied gradually on individual junctions. It helps us to preserve synchronization and offsets among junctions.

The main structure of algorithms of AIMSUN scenario life cycle and MATLAB traffic toolbox are given close together [4]. In these two algorithms you can follow jumps between algorithms to follow the process of communication and synchronization.

There are four basic steps how add new area in editor TEDI [3] and make experiments in simulator Aimsun [2]:

1. Draw the traffic area
Basic building objects are lanes, which may be connected into sections, junctions and centroids. The data from detectors are needed for feedback control.
2. Create control plan
There are three types of junctions: uncontrolled, internal (fixed) controlled from Aimsun and external controlled from other applications. Steps of creating the control plan are: select lines, setup them into signal groups and assign them into phases.
3. Set entrance flows
The main parameters of entrances are: count of vehicles [uv/h], types of cars (car, bus) and turning conditions.
4. Simulation
New Aimsun scenario made in few steps: loading traffic network, loading traffic control, loading result container and alternatively choosing DLL-library [4, 5], if external control is used.

2 Aims of experiments

Aim of experiment is compare two type of control. first type is dynamic with usage Kalman filter. second is static, with constant control plan during whole time simulation.

3 Description of Area

3.1 Network preview

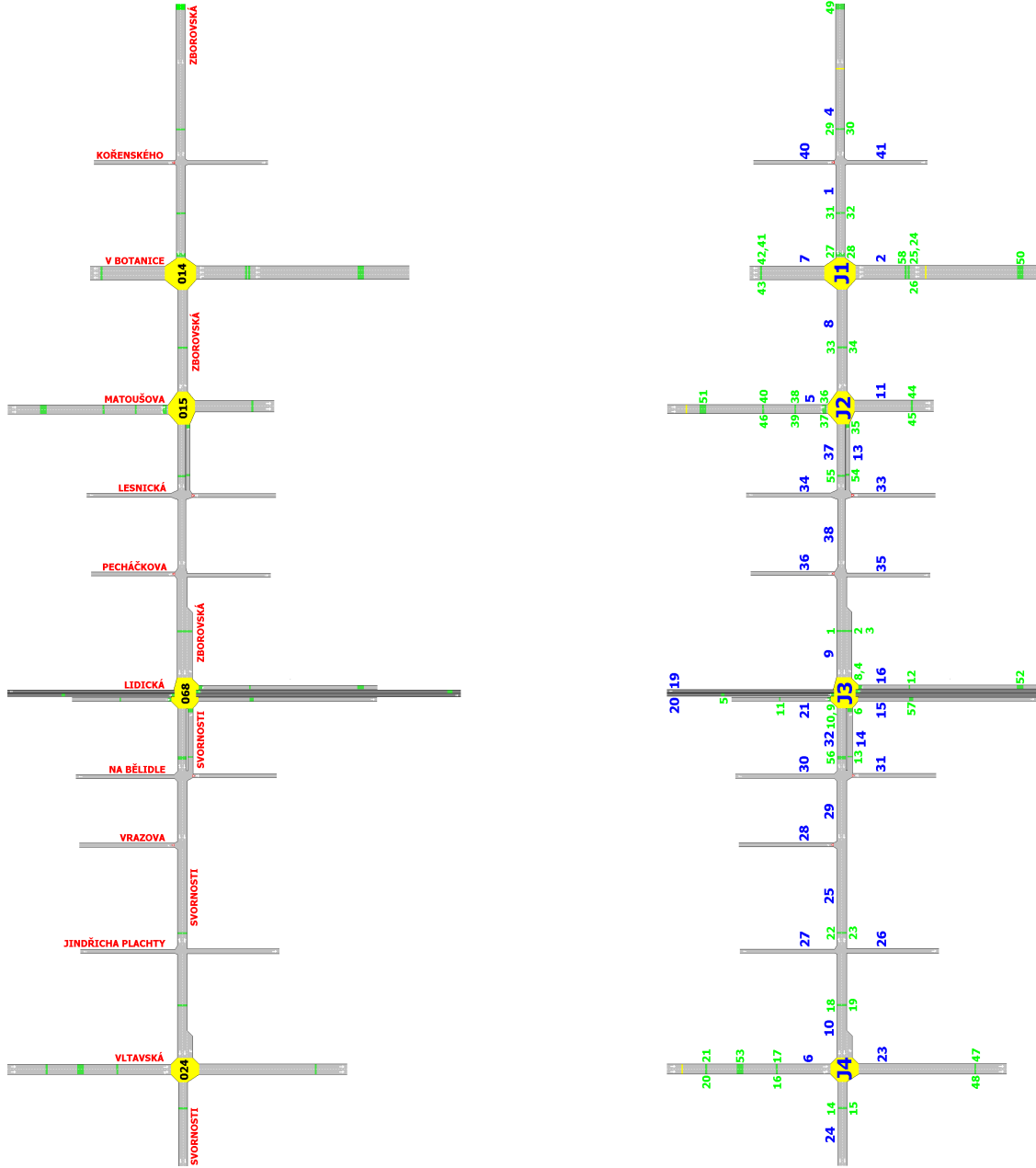


Figure 3.1: Area map of Smichov, Prague.

3.2 Network design and notation

Smichov (Chapter 3) is real area from Prague designed for testing control algorithms. It includes 10 intersections and 4 of them are controlled. The main direction is from right to left (majority is one-way). In this order is the area parsed into single junctions and detailed described.

4 Description of Experiments

Traffic area: Smichov, Prague
Source of input flows: Offline Entrances from File
Traffic composition: 90% cars, 10% buses
Simulation time: 1 day
Detection interval: 90s
Statistics interval: 90s
Number of experiments: 2

4.1 Experiment 1: Dynamic Control with Kalman Filter

Author: Marek Elbl (elblm1@fel.cvut.cz)
Date and time: 2007-08-04, 16:42:04
Workspace: C:\MyReports\smichov_buses_20070804_164203\smichov_buses.mat
Workstation: marek.sh.cvut.cz
Description:
Dynamic Control with Kalman Filter

4.2 Experiment 2: Static Control without Feedback

Author: Pavel Dohnal (dohnalp@utia.cas.cz)
Date and time: 2007-08-02, 20:05:48
Workspace: C:\MyReports\smichov_buses_20070804_165548\smichov_buses.mat
Workstation: axel.pod.cvut.cz
Description:
Static Control without Feedback

5 Input data

There are input traffic flows (Offline Entrances from File).

5.1 Arms 4, 40

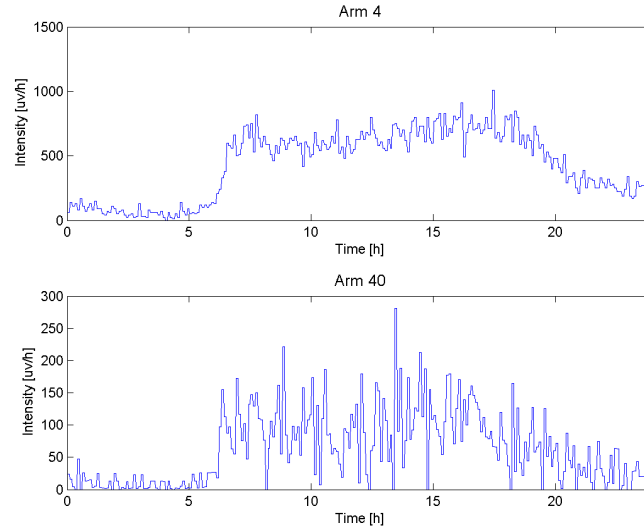


Figure 5.1: Input traffic intensities of arms 4, 40 (see Chapter 3).

5.2 Arms 3, 5

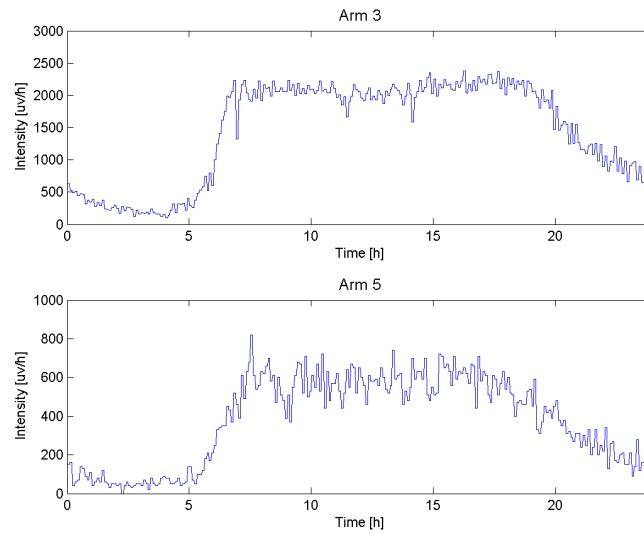


Figure 5.2: Input traffic intensities of arms 3, 5 (see Chapter 3).

5.3 Arms 16, 21

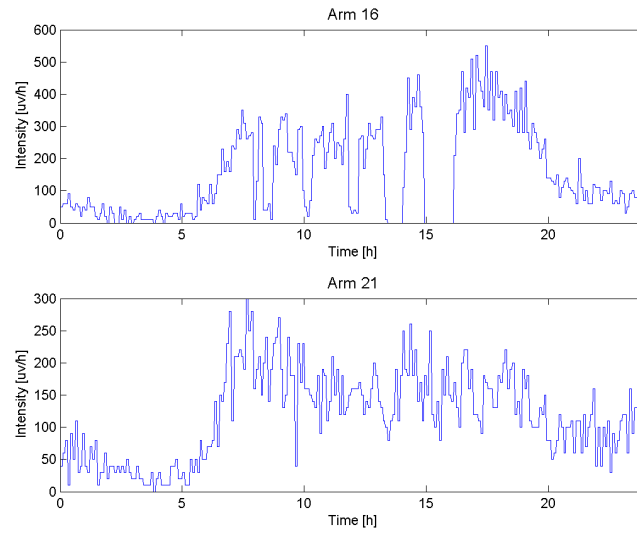


Figure 5.3: Input traffic intensities of arms 16, 21 (see Chapter 3).

5.4 Arms 6, 33

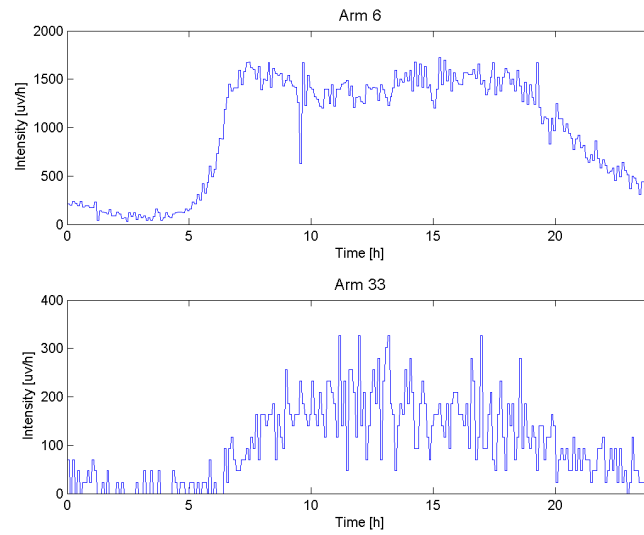


Figure 5.4: Input traffic intensities of arms 6, 33 (see Chapter 3).

5.5 Arms 36, 31

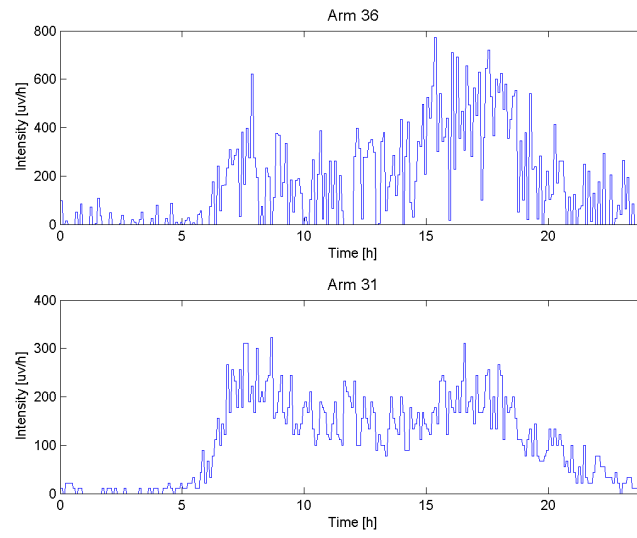


Figure 5.5: Input traffic intensities of arms 36, 31 (see Chapter 3).

5.6 Arms 28

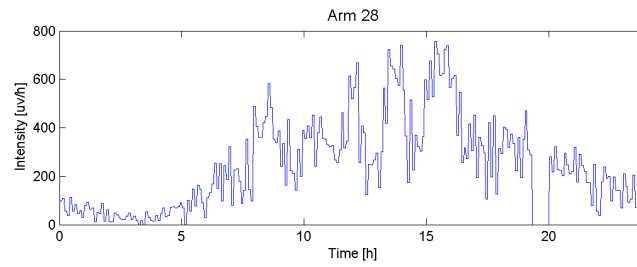


Figure 5.6: Input traffic intensities of arms 28 (see Chapter 3).

6 Output data

6.1 Input arms

6.1.1 Arm 1

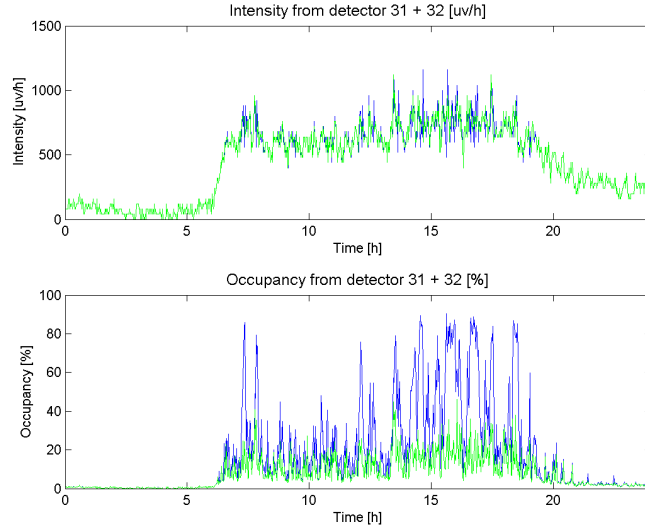


Figure 6.1: Intensity and occupancy from detector 31+32 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

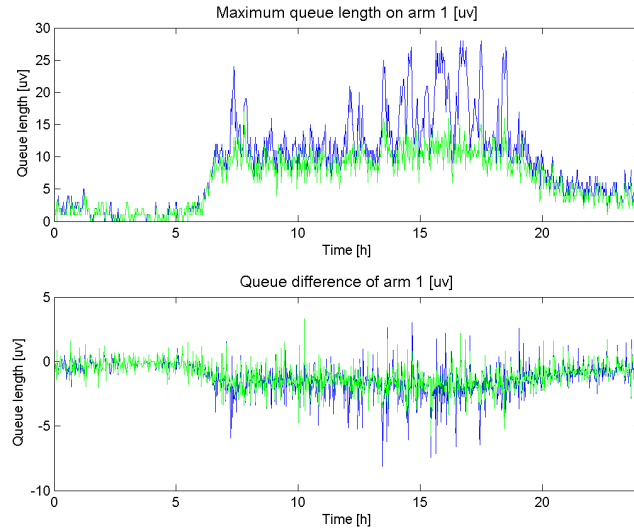


Figure 6.2: Maximum queue length and difference (between simulated and estimated values) on section 1 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

6.1.2 Arm 2

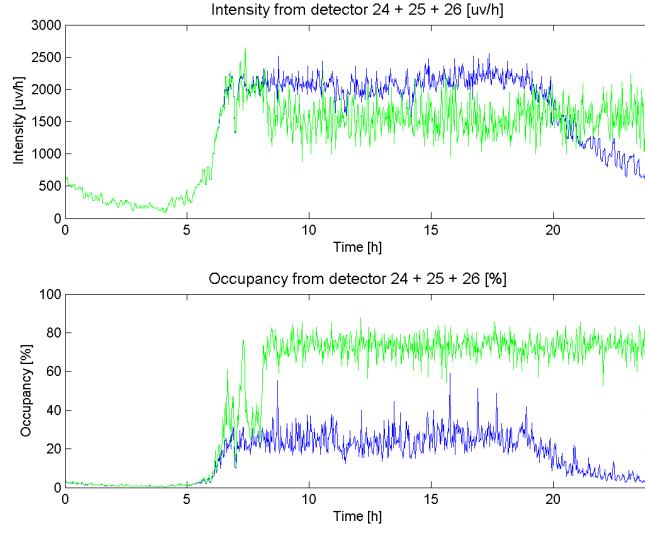


Figure 6.3: Intensity and occupancy from detector 24+25+26 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

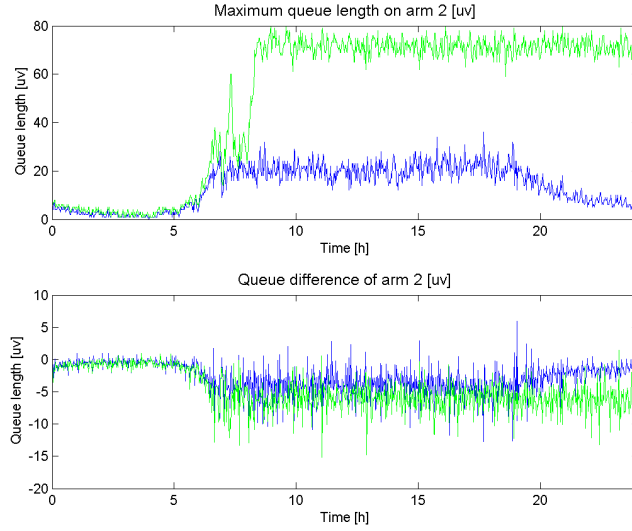


Figure 6.4: Maximum queue length and difference (between simulated and estimated values) on section 2 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

6.1.3 Arm 5

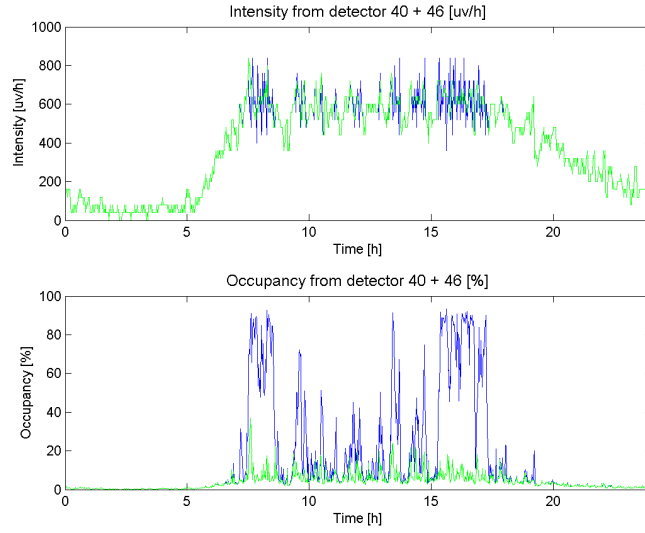


Figure 6.5: Intensity and occupancy from detector 40+46 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

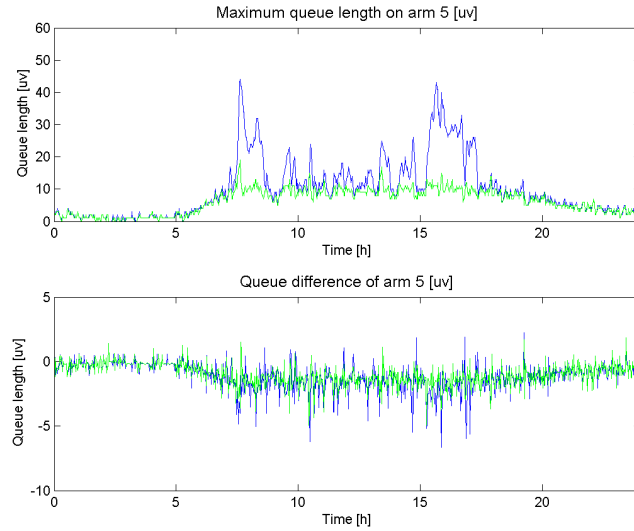


Figure 6.6: Maximum queue length and difference (between simulated and estimated values) on section 5 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

6.1.4 Arm 6

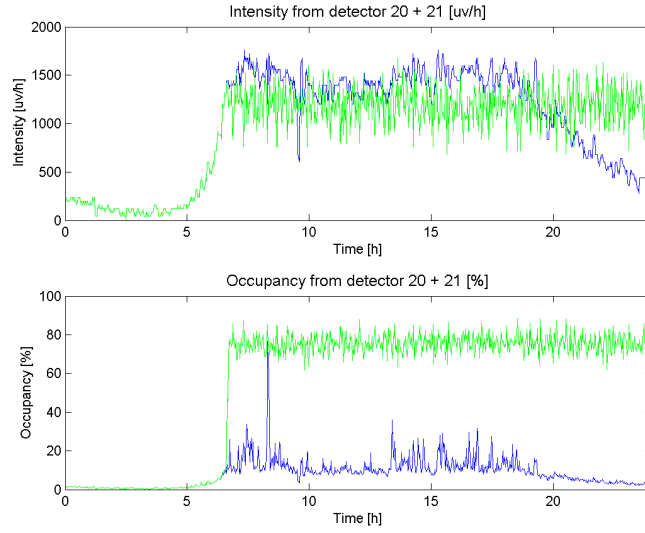


Figure 6.7: Intensity and occupancy from detector 20+21 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

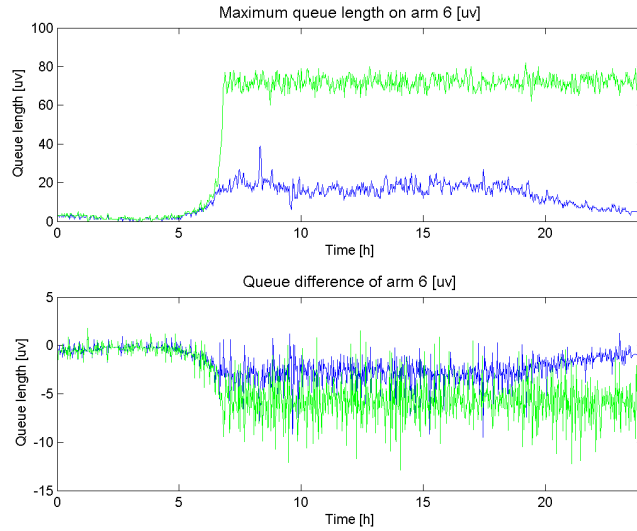


Figure 6.8: Maximum queue length and difference (between simulated and estimated values) on section 6 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

6.1.5 Arm 8

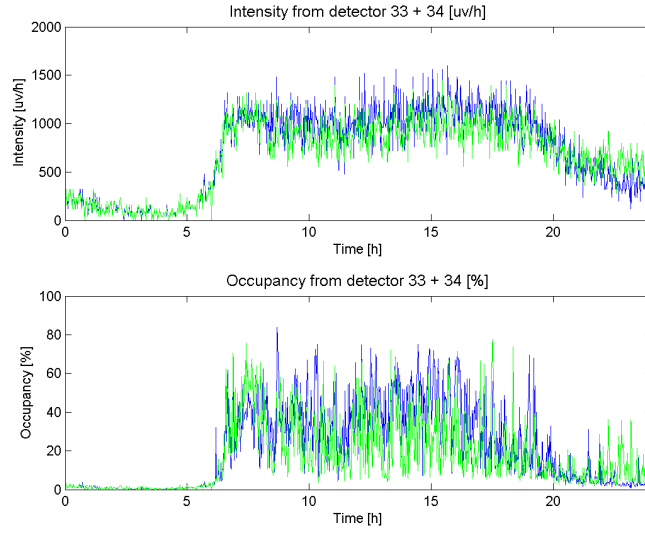


Figure 6.9: Intensity and occupancy from detector 33+34 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

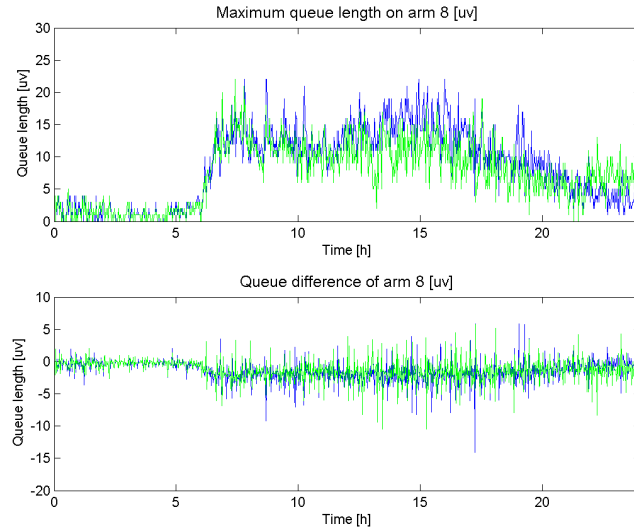


Figure 6.10: Maximum queue length and difference (between simulated and estimated values) on section 8 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

6.1.6 Arm 9

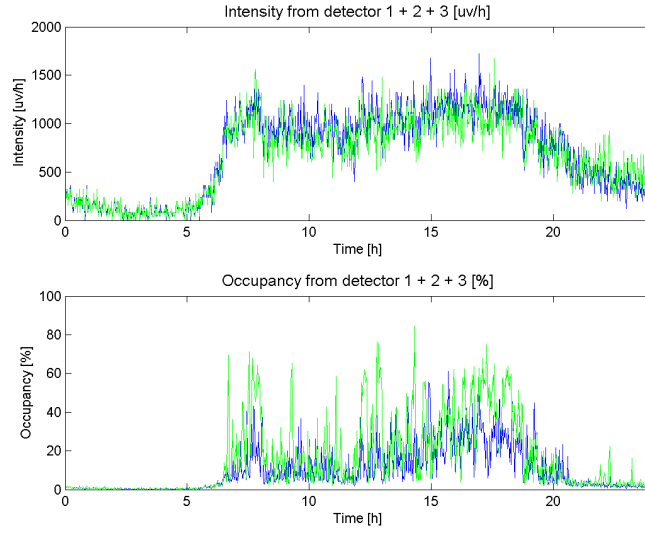


Figure 6.11: Intensity and occupancy from detector 1+2+3 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

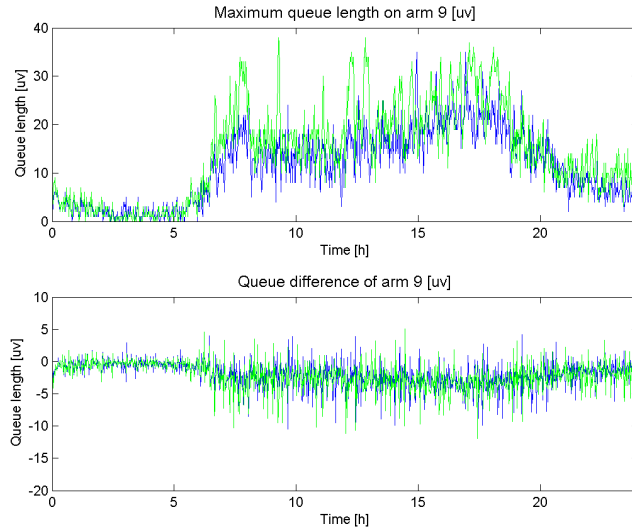


Figure 6.12: Maximum queue length and difference (between simulated and estimated values) on section 9 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

6.1.7 Arm 10

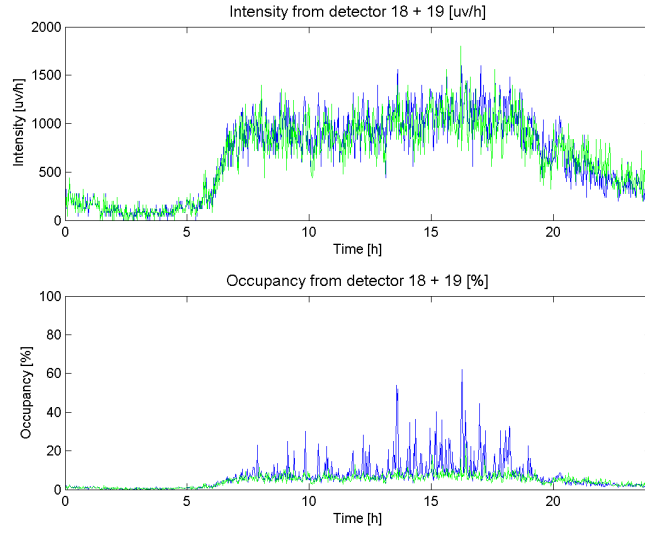


Figure 6.13: Intensity and occupancy from detector 18+19 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

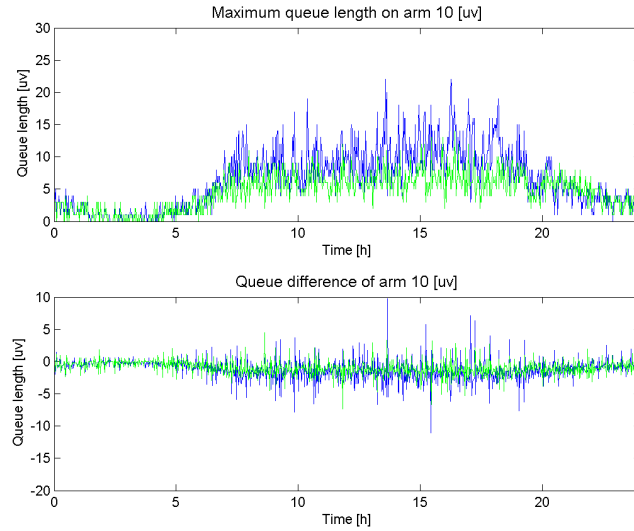


Figure 6.14: Maximum queue length and difference (between simulated and estimated values) on section 10 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

6.1.8 Arm 13

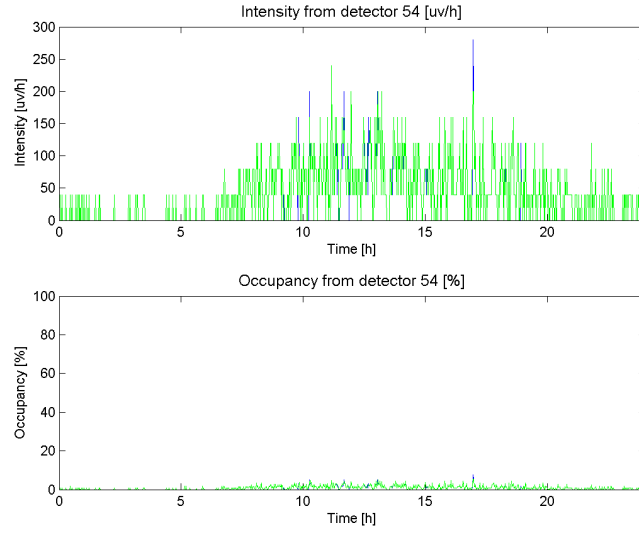


Figure 6.15: Intensity and occupancy from detector 54 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

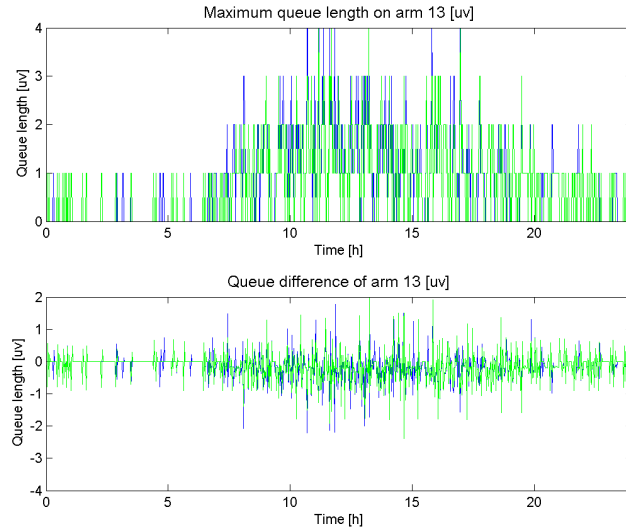


Figure 6.16: Maximum queue length and difference (between simulated and estimated values) on section 13 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

6.1.9 Arm 14

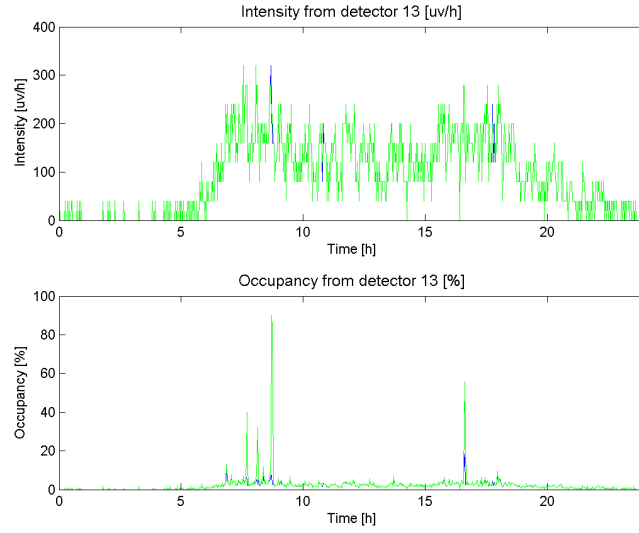


Figure 6.17: Intensity and occupancy from detector 13 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

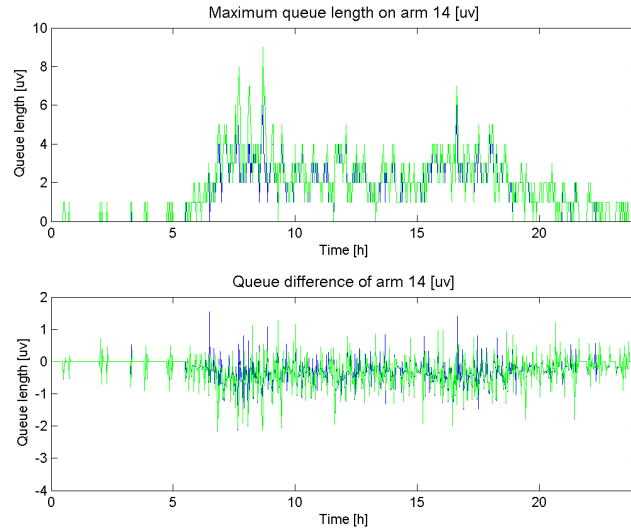


Figure 6.18: Maximum queue length and difference (between simulated and estimated values) on section 14 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

6.1.10 Arm 16

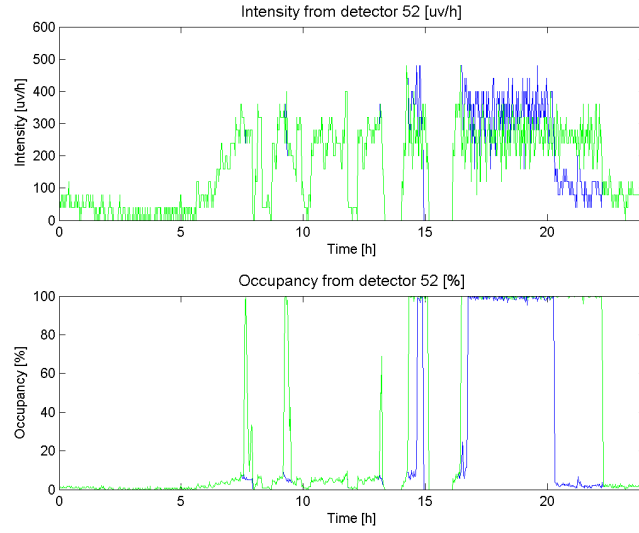


Figure 6.19: Intensity and occupancy from detector 52 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

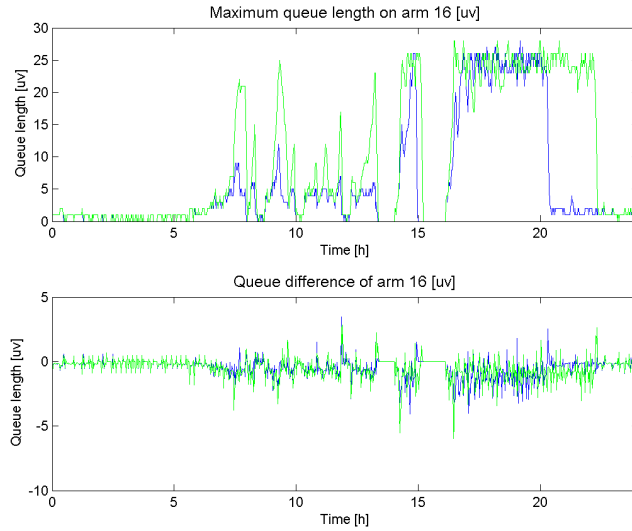


Figure 6.20: Maximum queue length and difference (between simulated and estimated values) on section 16 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

6.1.11 Arm 21

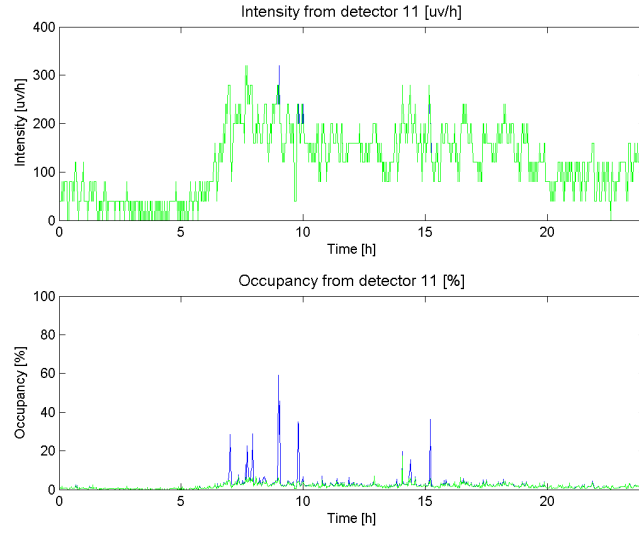


Figure 6.21: Intensity and occupancy from detector 11 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

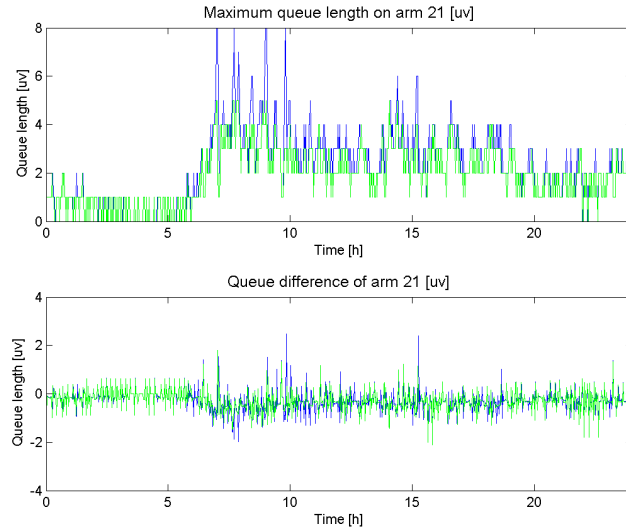


Figure 6.22: Maximum queue length and difference (between simulated and estimated values) on section 21 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

6.2 Output arms

6.2.1 Arm 7

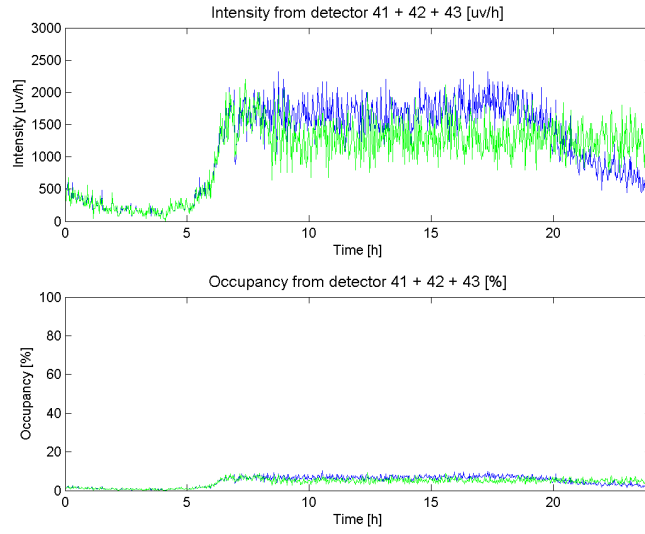


Figure 6.23: Intensity and occupancy from detector 41+42+43 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

6.2.2 Arm 10

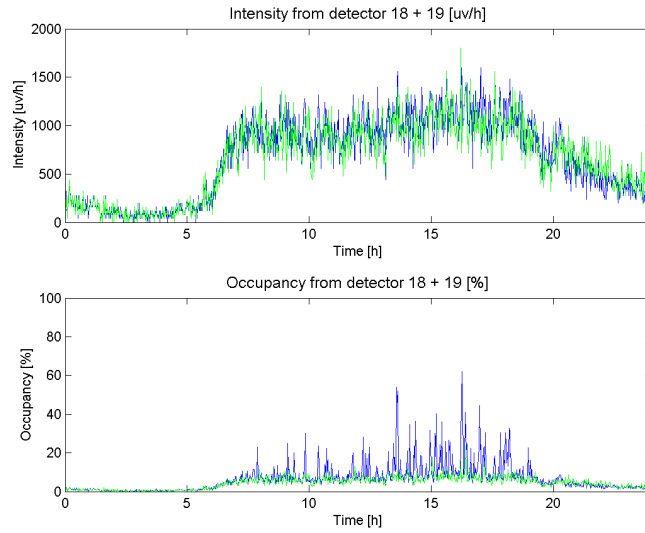


Figure 6.24: Intensity and occupancy from detector 18 + 19 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

6.2.3 Arm 11

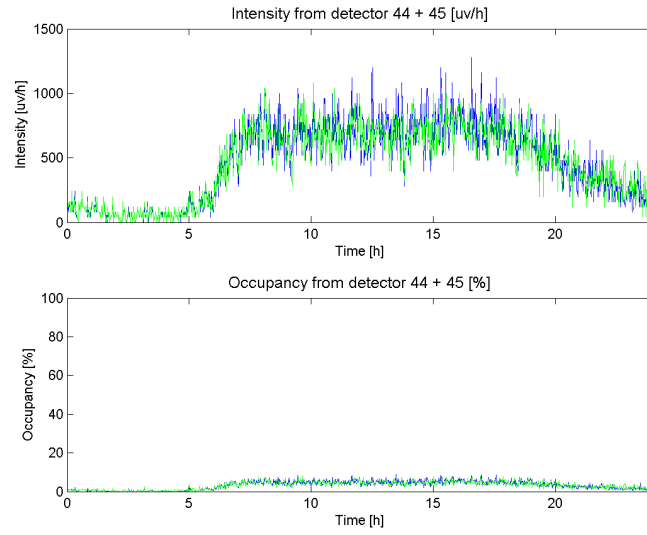


Figure 6.25: Intensity and occupancy from detector 44+45 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

6.2.4 Arm 13

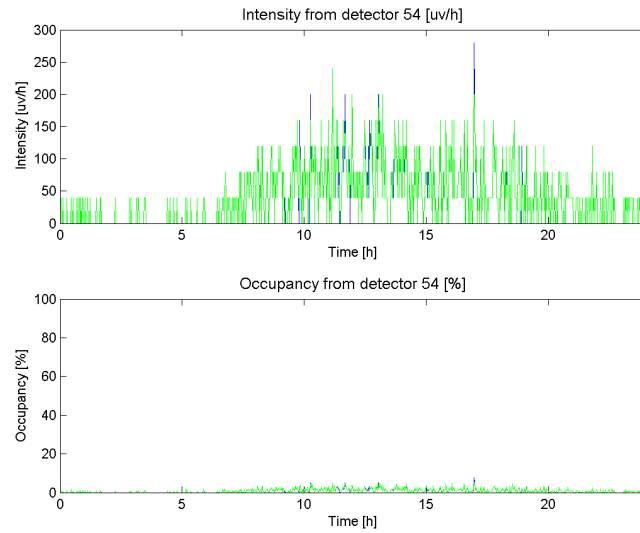


Figure 6.26: Intensity and occupancy from detector 54 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

6.2.5 Arm 14

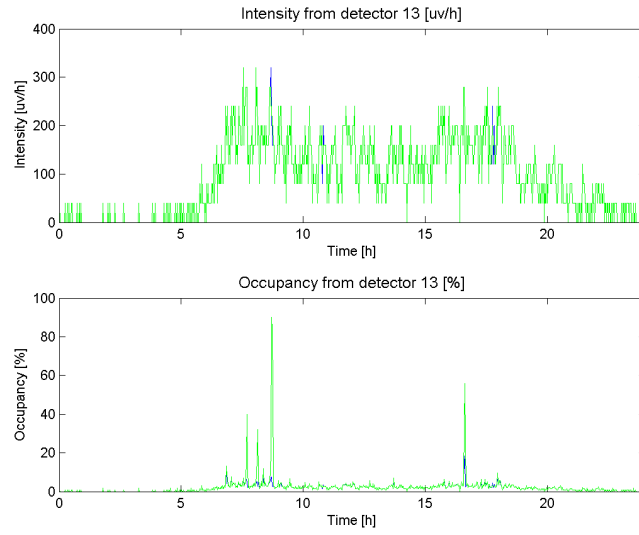


Figure 6.27: Intensity and occupancy from detector 13 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

6.2.6 Arm 15

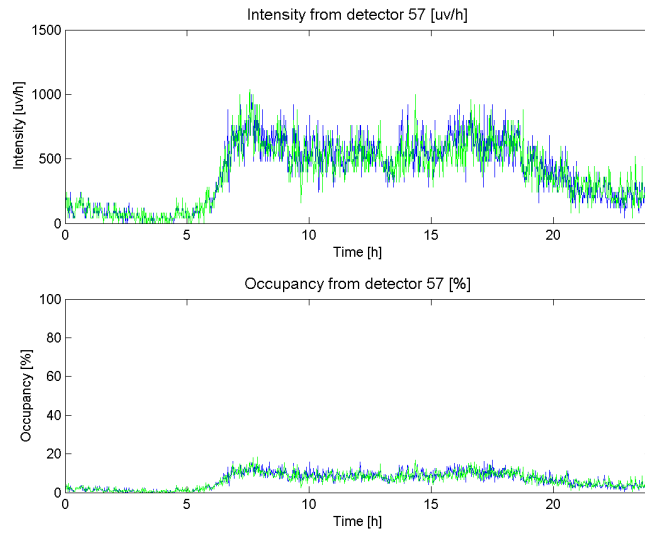


Figure 6.28: Intensity and occupancy from detector 57 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

6.2.7 Arm 23

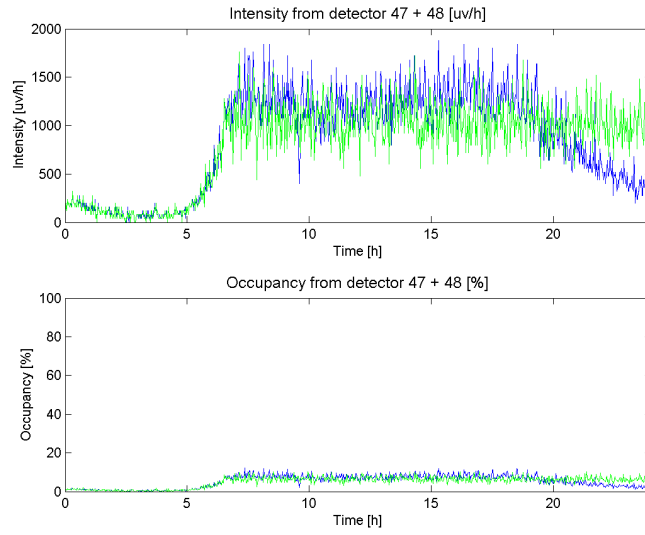


Figure 6.29: Intensity and occupancy from detector 47+48 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

6.2.8 Arm 24

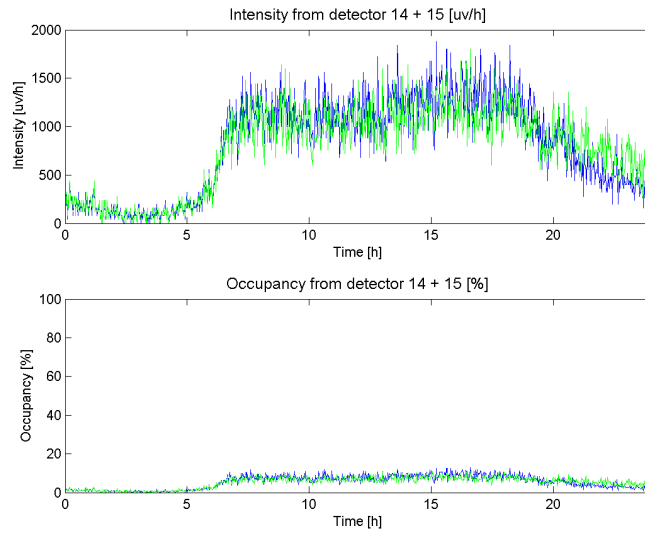


Figure 6.30: Intensity and occupancy from detector 14 + 15 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

6.2.9 Arm 32

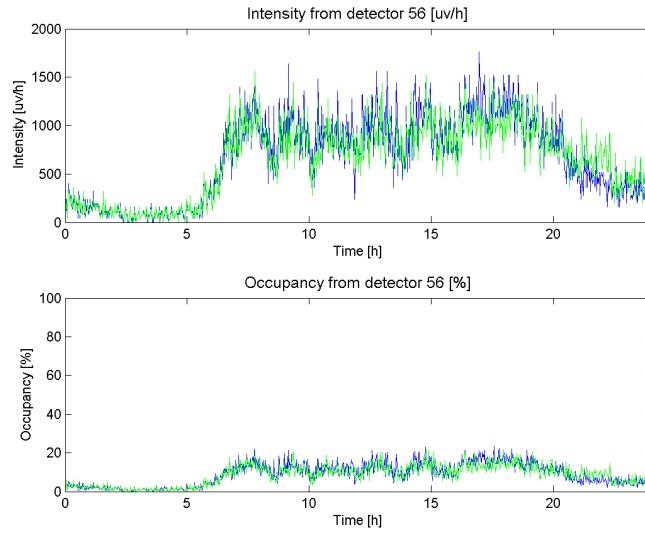


Figure 6.31: Intensity and occupancy from detector 56 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

6.2.10 Arm 37

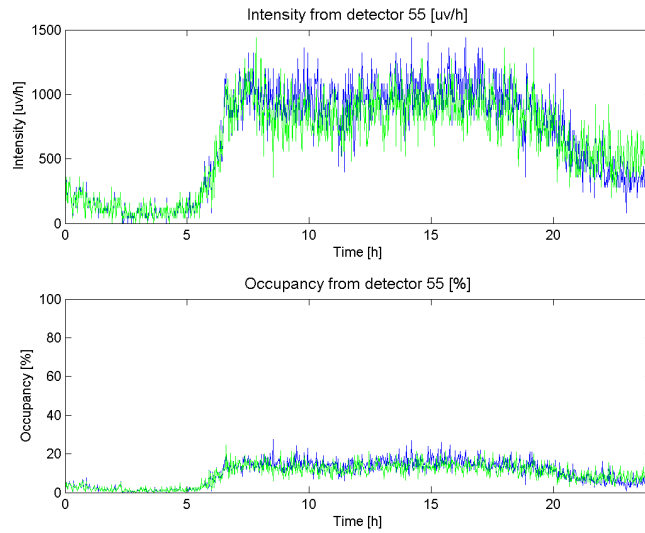


Figure 6.32: Intensity and occupancy from detector 55 (see Chapter 3). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

7 Results

7.1 Estimated parameters

7.1.1 Parameters κ and λ

Parameters are estimated from equation $O_{t+1} = \kappa \xi_t + \lambda$ [7], where: O_t is occupancy in time t , ξ_t is queue length in time t , κ and λ are parameters.

<i>Arm</i>	1	2	5	6	8	9	10	13
κ	0.051700	0.045386	0.035719	0.020819	0.065785	0.026933	0.028919	0.011599
λ	-0.038754	0.023632	-0.040418	0.001582	0.016839	-0.009025	0.009619	0.005959
<i>MSE</i>	0.002977	0.001941	0.002969	0.001154	0.004149	0.002350	0.001644	0.000320
<i>Arm</i>	14	16	21					
κ	0.023229	0.049009	0.032105					
λ	0.004488	-0.019845	-0.009244					
<i>MSE</i>	0.000362	0.003052	0.000967					

Table 1: Table 7.1: Estimated parameters κ , λ and mean squared error *MSE* for Experiment 1 – Dynamic Control with Kalman Filter (see Chapter 4.1).

<i>Arm</i>	1	2	5	6	8	9	10	13
κ	0.043317	0.013964	0.020052	0.015024	0.057276	0.029576	0.034953	0.015548
λ	-0.012938	0.063805	-0.007149	0.017744	0.029633	-0.024064	0.016050	0.005656
<i>MSE</i>	0.001716	0.002941	0.000844	0.002233	0.004069	0.002938	0.000665	0.000308
<i>Arm</i>	14	16	21					
κ	0.041938	0.045905	0.020386					
λ	-0.008294	-0.036116	0.003816					
<i>MSE</i>	0.001320	0.004600	0.000312					

Table 2: Table 7.2: Estimated parameters κ , λ and mean squared error *MSE* for Experiment 2 – Static Control without Feedback (see Chapter 4.2).

7.1.2 Parameters κ , β and λ

Parameters are estimated from equation $O_{t+1} = \kappa\xi_t + \beta O_t + \lambda$ [7], where: O_t is occupancy in time t , ξ_t is queue length in time t , κ , β and λ are parameters.

<i>Arm</i>	1	2	5	6	8	9	10	13
κ	0.026357	0.002070	0.012752	0.009958	0.022342	0.014795	0.019119	0.004226
β	0.463966	0.853500	0.632252	0.493871	0.553002	0.380449	0.273193	0.349360
λ	-0.015179	0.016960	-0.012733	0.003049	0.028929	0.001997	0.010638	0.004693
<i>MSE</i>	0.002841	0.001677	0.002473	0.001060	0.003963	0.002284	0.001616	0.000307

<i>Arm</i>	14	16	21
κ	0.013555	0.004615	0.023038
β	0.371615	0.903517	0.258701
λ	0.003397	-0.001441	-0.006087
<i>MSE</i>	0.000344	0.001488	0.000938

Table 3: Table 7.3: Estimated parameters κ , β , λ and mean squared error *MSE* for Experiment 1 – Dynamic Control with Kalman Filter (see Chapter 4.1).

<i>Arm</i>	1	2	5	6	8	9	10	13
κ	0.030538	0.002262	0.015205	0.001517	0.040698	0.018782	0.008454	0.008429
β	0.266591	0.828879	0.228733	0.888970	0.216218	0.334375	0.664244	0.286057
λ	-0.006881	0.015651	-0.004897	0.007994	0.033447	-0.010430	0.008115	0.004579
<i>MSE</i>	0.001674	0.002001	0.000826	0.001674	0.004043	0.002874	0.000557	0.000299

<i>Arm</i>	14	16	21
κ	0.017728	0.002741	0.015490
β	0.536222	0.936344	0.212367
λ	-0.002606	-0.000928	0.003390
<i>MSE</i>	0.001139	0.001982	0.000307

Table 4: Table 7.4: Estimated parameters κ , β , λ and mean squared error *MSE* for Experiment 2 – Static Control without Feedback (see Chapter 4.2).

7.2 Statistics

7.2.1 Global statistics

There are the global statistical data gathered during the whole simulation, for the whole system. First experiment represents reference. Δ_{x1} represents differences (in %) between applicative experiment x and reference.

<i>Statistic</i>	Δ_1	$\min(\Delta_1)$	$\max(\Delta_1)$	$\text{avg}(\Delta_1)$	$\text{med}(\Delta_1)$	$\text{std}(\Delta_1)$
<i>Total flow [uv/h]</i>	4208	240	7840	4208	5360	2388
<i>Travel time [s/km]</i>	173	85	276	158	153	32
<i>Delay time [s/km]</i>	94	11	196	80	75	31
<i>Average speed [km/h]</i>	27	20	44	29	28	4
<i>Average space speed [km/h]</i>	21	13	43	24	23	5
<i>Density [uv/km]</i>	14	1	33	14	16	9
<i>Stop time [s/km]</i>	76	7	173	64	58	28
<i>Number of stops [uv/km]</i>	3	1	6	3	3	1
<i>Total intensity [uv/h]</i>	14839	1000	26560	14839	19360	8379
<i>Average occupancy [%]</i>	18	0	39	18	21	11
<i>Average cycle length [s]</i>	80	80	80	80	80	0
<i>Total est. queue length [uv]</i>	74	2	181	74	87	48
<i>Total sim. queue length [uv]</i>	88	3	211	88	104	56

Table 5: Table 7.5: Experiment 1 – Dynamic Control with Kalman Filter (see Chapter 4.1).

<i>Statistic</i>	Δ_2	$\min(\Delta_2)$	$\max(\Delta_2)$	$\text{avg}(\Delta_2)$	$\text{med}(\Delta_2)$	$\text{std}(\Delta_2)$
<i>Total flow [uv/h]</i>	3988	200	7160	3988	4800	1990
<i>Travel time [s/km]</i>	351	102	498	306	354	114
<i>Delay time [s/km]</i>	272	28	423	228	275	114
<i>Average speed [km/h]</i>	16	11	40	19	15	8
<i>Average space speed [km/h]</i>	10	7	35	15	10	8
<i>Density [uv/km]</i>	25	1	44	25	32	14
<i>Stop time [s/km]</i>	251	21	405	209	254	110
<i>Number of stops [uv/km]</i>	9	1	14	8	9	4
<i>Total intensity [uv/h]</i>	14192	880	24720	14192	17560	7141
<i>Average occupancy [%]</i>	21	0	37	21	25	11
<i>Average cycle length [s]</i>	80	80	80	80	80	0
<i>Total est. queue length [uv]</i>	146	2	249	146	186	83
<i>Total sim. queue length [uv]</i>	163	3	281	163	208	91

Table 6: Table 7.6: Experiment 2 – Static Control without Feedback (see Chapter 4.2).

7.2.2 Global statistics for spaces of time

There are the time aggregated statistical data gathered during spaces of time, for the whole system.

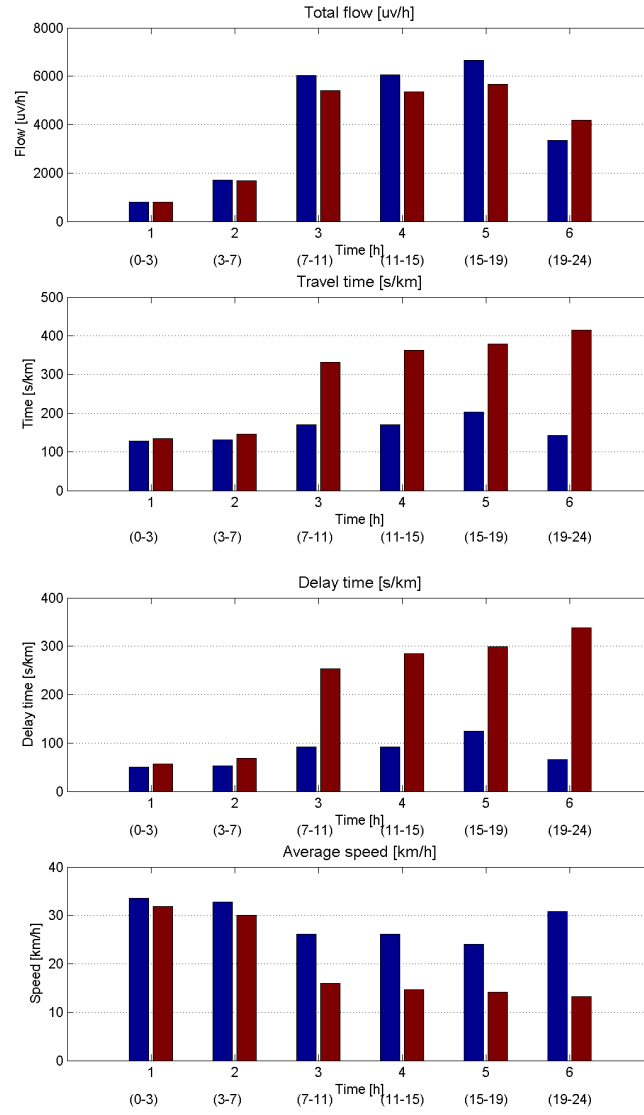


Figure 7.1: Global statistics for spaces of time (1/4). Experiment1 (first column) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (second column) – Static Control without Feedback (see Chapter 4.2).

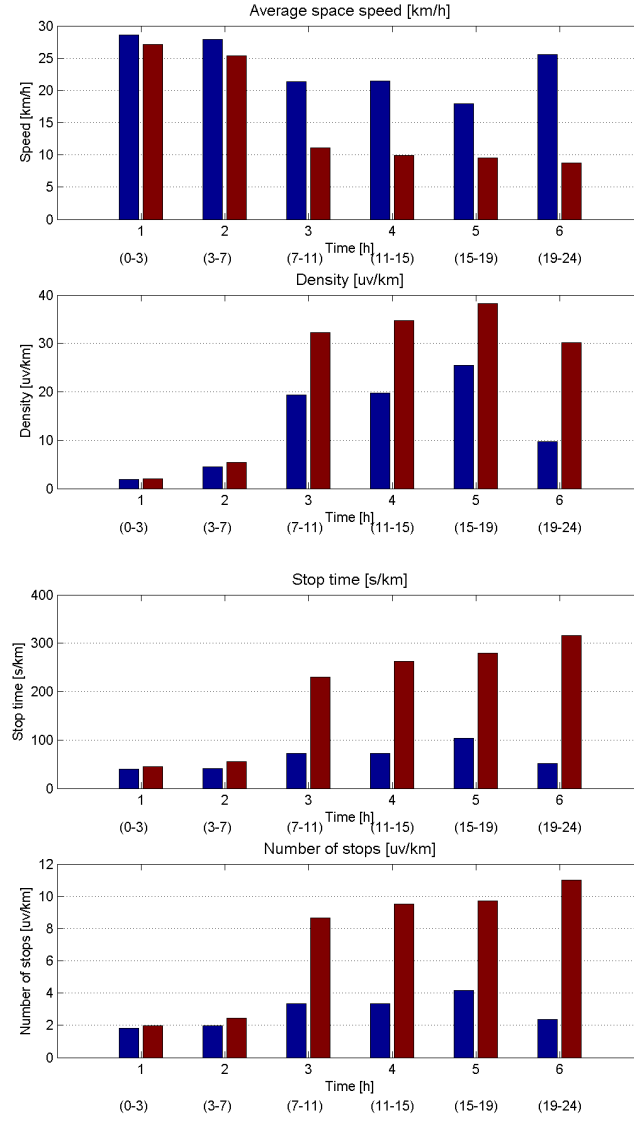


Figure 7.2: Global statistics for spaces of time (2/4). Experiment1 (first column) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (second column) – Static Control without Feedback (see Chapter 4.2).

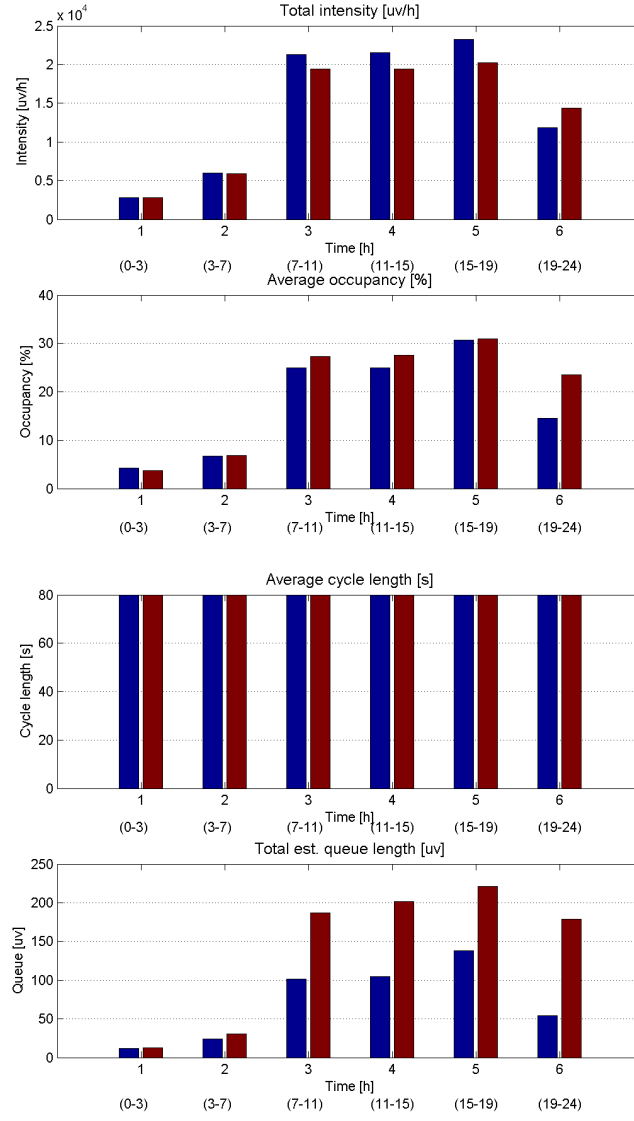


Figure 7.3: Global statistics for spaces of time (3/4). Experiment1 (first column) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (second column) – Static Control without Feedback (see Chapter 4.2).

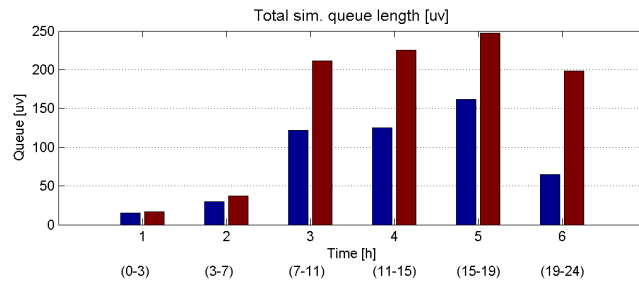


Figure 7.4: Global statistics for spaces of time (4/4). Experiment1 (first column) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (second column) – Static Control without Feedback (see Chapter 4.2).

7.2.3 Periodical statistics

There are the periodical statistical data gathered during each simulation step, for the whole system.

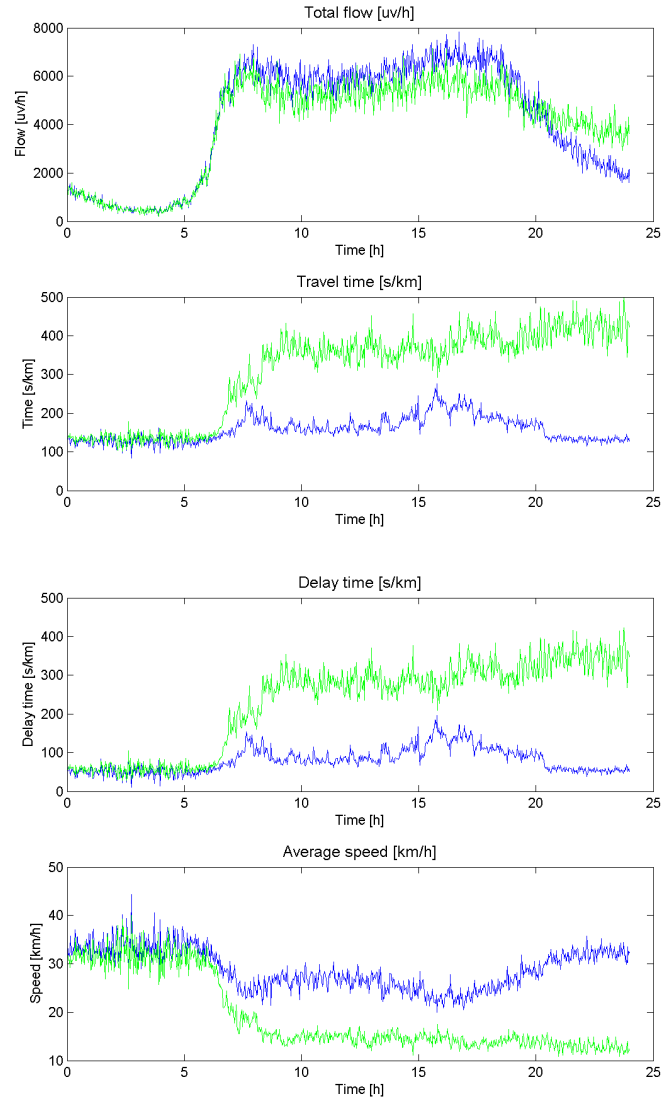


Figure 7.5: Periodical statistics (1/4). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

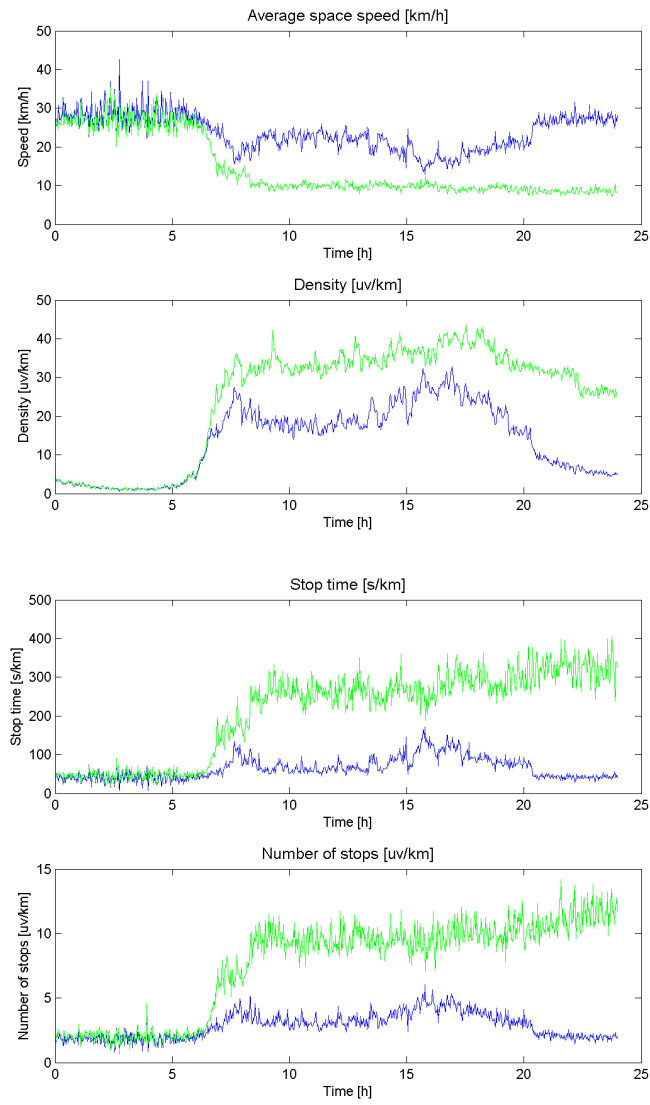


Figure 7.6: Periodical statistics (2/4). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

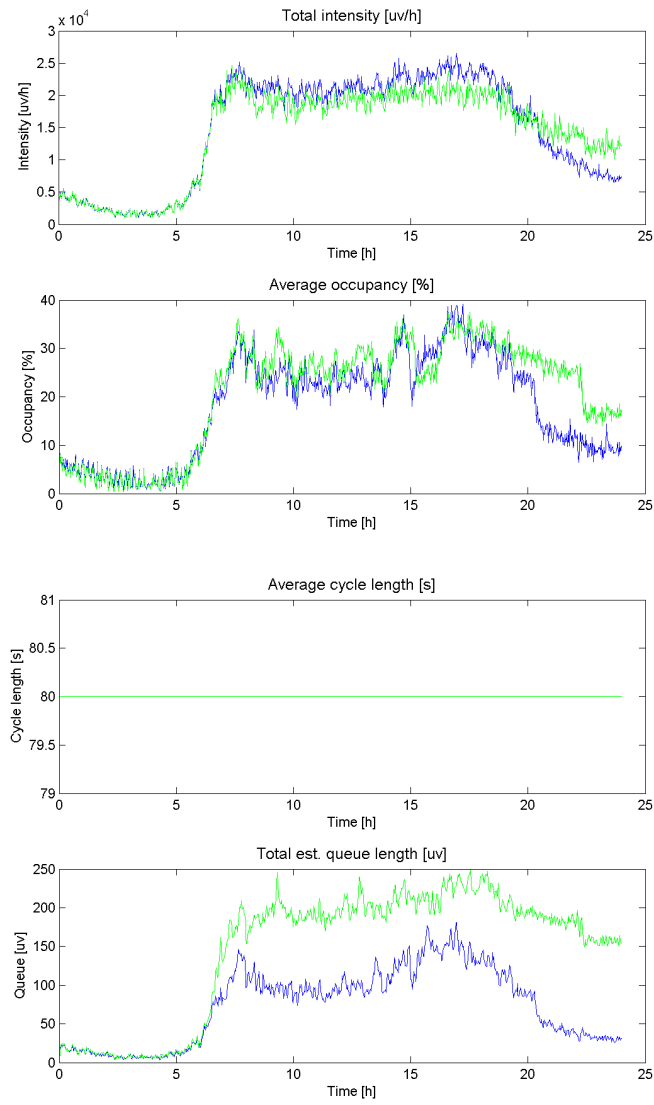


Figure 7.7: Periodical statistics (3/4). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

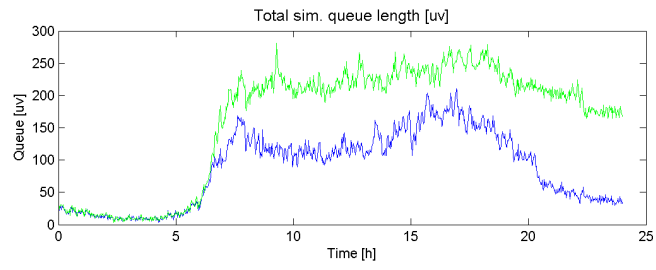


Figure 7.8: Periodical statistics (4/4). Experiment1 (blue line) – Dynamic Control with Kalman Filter (see Chapter 4.1), Experiment2 (green line) – Static Control without Feedback (see Chapter 4.2).

8 Conclusion

As evidenced by results of both experiments (chapter x.x), so first type of control (Dynamic Control with Kalman Filter) get past than second control (Static Control without Feedback), because is reach for higher permeability of areas (Total Flow), decrease of the number of stopping car (Number of Stops), shortening length of queues (Queue Lengths) and time of passage area (Travel Time).

9 Appendix

9.1 Structures

Instructions for experiments and simulations making in MATLAB-AIMSUN Toolbox is described in [5] and [6].

9.1.1 Control matrix

Traffic may be controlled by function *setActualControl(MyArea, PhasesLengths)*. New phase lengths are setting in matrix *PhasesLengths*. Every row represents just one intersection. Phase lengths are in columns. Unused matrix positions have value *NaN*.

$$PhasesLengths = \begin{bmatrix} J_1^1 & J_1^2 & J_1^3 & J_1^4 & J_1^5 & J_1^6 & J_1^7 & J_1^8 \\ J_2^1 & J_2^2 & J_2^3 & J_2^4 & J_2^5 & J_2^6 & J_2^7 & J_2^8 \\ J_3^1 & J_3^2 & J_3^3 & J_3^4 & J_3^5 & J_3^6 & J_3^7 & J_3^8 \\ J_4^1 & J_4^2 & J_4^3 & J_4^4 & J_4^5 & J_4^6 & J_4^7 & J_4^8 \end{bmatrix} \quad (1)$$

9.1.2 Data vectors

Actual traffic data (intensities, occupancies a velocities) from detectors may be getting by function *[MyArea, Intensity, Occupancy, Velocity, Time] = getActualData(MyArea)*, where *Time* is point in time. Data are stored in column-vectors.

$$Intensity = \begin{bmatrix} I_{29} & I_{31} & I_{27} & I_{24} & I_{41} & I_{33} & I_{40} & I_{39} & I_{37} & I_{44} & \dots \\ I_{35} & I_1 & I_{12} & I_4 & I_{11} & I_{10} & I_{13} & I_6 & I_{22} & I_{18} & \dots \\ \dots & \dots & I_{20} & I_{16} & I_{47} & I_{14} & I_{54} & I_{55} & I_{56} & I_{57} & I_{52} \end{bmatrix}^T uv/T_{data} \quad (2)$$

$$Occupancy = \begin{bmatrix} O_{29} & O_{31} & O_{27} & O_{24} & O_{41} & O_{33} & O_{40} & O_{39} & O_{37} & O_{44} & \dots \\ O_{35} & O_1 & O_{12} & O_4 & O_{11} & O_{10} & O_{13} & O_6 & O_{22} & O_{18} & \dots \\ \dots & \dots & O_{20} & O_{16} & O_{47} & O_{14} & O_{54} & O_{55} & O_{56} & O_{57} & O_{52} \end{bmatrix}^T \% \quad (3)$$

$$Velocity = \begin{bmatrix} S_{29} & S_{31} & S_{27} & S_{24} & S_{41} & S_{33} & S_{40} & S_{39} & S_{37} & S_{44} & \dots \\ S_{35} & S_1 & S_{12} & S_4 & S_{11} & S_{10} & S_{13} & S_6 & S_{22} & S_{18} & \dots \\ \dots & \dots & S_{20} & S_{16} & S_{47} & S_{14} & S_{54} & S_{55} & S_{56} & S_{57} & S_{52} \end{bmatrix}^T m/s \quad (4)$$

9.1.3 Statistics vectors

Actual queue lengths from sections may be getting by function *[MyArea, QueueAverage, QueueMaximal, Time] = getActualStatistics(MyArea)*, where *Time* is point in time. Statistics are stored in column-vectors.

$$QueueAverage = \begin{bmatrix} Q_4 & Q_{40} & Q_1 & Q_2 & Q_8 & Q_5 & Q_{37} & Q_{13} & Q_{33} & Q_{38} & \dots \\ Q_{36} & Q_9 & Q_{16} & Q_{21} & Q_{32} & Q_{14} & Q_{31} & Q_{29} & Q_{28} & Q_{25} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & Q_{10} & Q_6 \end{bmatrix}^T \quad (5)$$

$$QueueMaximal = \begin{bmatrix} Q_4 & Q_{40} & Q_1 & Q_2 & Q_8 & Q_5 & Q_{37} & Q_{13} & Q_{33} & Q_{38} & \dots \\ Q_{36} & Q_9 & Q_{16} & Q_{21} & Q_{32} & Q_{14} & Q_{31} & Q_{29} & Q_{28} & Q_{25} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & Q_{10} & Q_6 \end{bmatrix}^T \quad (6)$$

References

- [1] Dohnal P.: *Popis experimentované testovací oblasti Smichov, Prague [online]*.
<http://marabu.utia.cas.cz:1800/svn/doprava/AIMSUN-MATLAB/toolboxASYN/areas/smichov/smichov.pdf>
- [2] Transport Simulation Systems: *AIMSUN Version 4.2 - User manual [online]*.
http://marabu.utia.cas.cz:1800/svn/doprava/Ruzne/AIMSUN_manually/aimsun_42.pdf
- [3] Transport Simulation Systems: *TEDI Version 4.2 - User manual [online]*.
http://marabu.utia.cas.cz:1800/svn/doprava/Ruzne/AIMSUN_manually/tedi_42.pdf
- [4] Gebouský P.: *AIMSUN-MATLAB Interface - User's Guide [online]*.
<http://marabu.utia.cas.cz:1800/svn/doprava/AIMSUN-MATLAB/manualASYN/manualASYN.pdf>
- [5] Dohnal P., Dibelka L.: *AIMSUN-MATLAB Toolbox - User's Guide [online]*.
<http://marabu.utia.cas.cz:1800/svn/doprava/AIMSUN-MATLAB/manualMAT/manualMAT.pdf>
- [6] Dohnal P.: *AIMSUN-MATLAB Toolbox - Nvod k pouit [online]*.
<http://marabu.utia.cas.cz:1800/svn/doprava/AIMSUN-MATLAB/manualMAT/guideMAT.pdf>
- [7] Dohnal P.: Bayes Estimation of a Queue Length. In *Proceedings of the 7th International PhD Workshop: Interplay of societal and technical decision-making, Young Generation Viewpoint*. Academy of Sciences of the Czech Republic - Institute of Information Theory and Automation, Prague, 2006, pp. 1-8.
- [8] Kratochvílová J., Nagy, I.: *Lokální řízení městské dopravy*. Academy of Sciences of the Czech Republic - Institute of Information Theory and Automation, Prague, 2004.