

Okruženje za simulacije sustava za upravljanje promjenjivih ograničenja brzine na gradskim autocestama zasnovano na dubokom učenju

Vrbanić, Filip

Master's thesis / Diplomski rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:119:060236>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom](#).

Download date / Datum preuzimanja: **2024-09-10**



Repository / Repozitorij:

[Faculty of Transport and Traffic Sciences - Institutional Repository](#)



**SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI**

Filip Vrbanić

**OKRUŽENJE ZA SIMULACIJE SUSTAVA ZA
UPRAVLJANJE PROMJENJIVIH OGRANIČENJA BRZINE
NA GRADSKIM AUTOCESTAMA ZASNOVANO NA
DUBOKOM UČENJU**

DIPLOMSKI RAD

Zagreb, 2019.

Zagreb, 11. travnja 2019.

Zavod: **Zavod za inteligentne transportne sustave**
Predmet: **Umjetna inteligencija**

DIPLOMSKI ZADATAK br. 5202

Pristupnik: **Filip Vrbanić (0036472710)**
Studij: **Inteligentni transportni sustavi i logistika**
Smjer: **Inteligentni transportni sustavi**

Zadatak: **Okruženje za simulacije sustava za upravljanje promjenjivih ograničenja
brzine na gradskim autocestama zasnovano na dubokom učenju**

Opis zadatka:

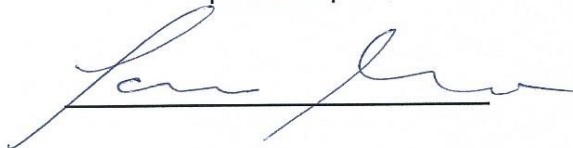
U uvjetima povećane prometne potražnje na gradskim autocestama javlja se potreba za upravljanje prometom s ciljem smanjenja i eliminiranja posljedica nastajanja prometnih zagušenja. Jedno od rješenja iz područja inteligentnih transportnih sustava je sustav za upravljanje promjenjivim ograničenjem brzine na gradskim autocestama. Noviji pristupi koriste algoritme zasnovane na dubokom učenju. U radu je potrebno opisati upravljanje prometom na gradskim autocestama zasnovano na upravljanju promjenjivim ograničenjem brzine, analizirati mogućnost primjene dubokog učenja u upravljanju promjenjivim ograničenjem brzine te izraditi okruženje za simulaciju takvog upravljanja korištenjem mikroskopskog simulatora VISSIM i programskog alata Python/Keras.

Mentor:



izv. prof. dr. sc. Edouard Ivanjko

Predsjednik povjerenstva za
diplomski ispit:



Sveučilište u Zagrebu
Fakultet prometnih znanosti

DIPLOMSKI RAD

**OKRUŽENJE ZA SIMULACIJE SUSTAVA ZA
UPRAVLJANJE PROMJENJIVIH OGRANIČENJA BRZINE
NA GRADSKIM AUTOCESTAMA ZASNOVANO NA
DUBOKOM UČENJU**

**FRAMEWORK FOR SIMULATION OF VARIABLE SPEED
LIMIT CONTROL SYSTEMS ON URBAN MOTORWAYS
BASED ON DEEP LEARNING**

Mentor: izv. prof. dr. sc. Edouard Ivanjko

Student: Filip Vrbanić
JMBAG: 0036472710

Zagreb, rujan 2019.

Zahvale

Veliku zahvalnost dugujem svom mentoru i profesoru izv. prof. dr. sc. Edouardu Ivanjku na pruženoj prilici i savjetima pri izradi diplomskog rada.

Također bih želio posebno zahvaliti neposrednom voditelju dr. sc. Martinu Greguriću na velikodušnoj i nesebičnoj pomoći pri izradi diplomskog rada.

Ovim putem se zahvaljujem i tvrtki PTV na pruženoj akademskoj licenci u svrhu izrade rada. Zahvaljujem se mojoj obitelji i djevojci na velikoj podršci i razumijevanju tijekom studiranja.

Sažetak

Naslov: Okruženje za simulacije sustava za upravljanje promjenjivih ograničenja brzine na gradskim autocestama zasnovano na dubokom učenju

Radi povećane prometne potražnje dolazi do nastanka zagušenja na gradskim autocestama te se javlja potreba za uvođenjem upravljanja prometom. Upravljanje prometom ima za cilj smanjenje i prevenciju posljedica nastajanja prometnih zagušenja. Sustav za upravljanje ograničenjem brzine na gradskim autocestama predstavlja jedno od mogućih rješenja iz područja inteligentnih transportnih sustava. Takav pristup u upravljanju promjenjivim ograničenjem brzine na gradskim autocestama može koristiti algoritme zasnovane na dubokom učenju. U ovom radu je opisano upravljanje prometom na gradskim autocestama zasnovano na upravljanju promjenjivim ograničenjem brzine, analizirana je mogućnost primjene dubokog učenja s posebnim naglaskom na algoritmu učenja dualne duboke Q-mreže u upravljanju promjenjivim ograničenjem brzine te je izrađeno okruženje za simulaciju navedenog upravljanja korištenjem mikroskopskog simulatora VISSIM i programskog okruženja Python/Keras. Dobiveni rezultati dokazuju da je primjenom dubokog učenja moguće iskoristiti informaciju o brzini i položaju svakog vozila za povećanje razine usluznosti gradske autoceste.

KLJUČNE RIJEČI: upravljanje promjenjivim ograničenjem brzine; simulacijsko okruženje; duboko učenje; ojačano učenje; inteligentni transportni sustavi

Summary

Title: Framework for Simulation of Variable Speed Limit Control Systems on Urban Motorways Based on Deep Learning

In the case of congestion on urban motorways due to increased traffic demand, there is a need to introduce dynamic traffic management to reduce and prevent the effects of traffic congestion. The urban motorway speed limit control system is one of the possible solutions in the field of Intelligent Transport System. Such an approach in managing variable speed limits on urban motorways can utilize a deep learning algorithm. This thesis describes urban motorway traffic management based on the variable speed limit control, analyzes the possibility of applying deep learning algorithm with special emphasis on the dueling deep Q-network learning algorithm in variable speed limit control, and creating an environment for simulating such system using the microscopic simulator VISSIM and the Python/Keras programming environment. The results prove that by applying such an algorithm it is possible to use information about the speed and position of each vehicle to increase the level of service of the urban motorway.

KEYWORDS: variable speed limit control; framework for simulation; deep learning; reinforcement learning; intelligent transport systems

Sadržaj

1. Uvod	1
2. Problem upravljanja promjenjivim ograničenjem brzine	3
3. Primjena dubokog učenja u upravljanju promjenjivim ograničenjem brzine	7
3.1. Ojačano učenje	7
3.1.1. Q-učenje.....	7
3.1.2. Algoritam učenja na principu duboke Q-mreže	8
3.1.3. Algoritam učenja na principu dualne duboke Q-mreže.....	11
3.2. Upravljanje promjenjivim ograničenjem brzine zasnovano na algoritmu učenja dualne duboke Q-mreže	12
4. Značajke simulatora VISSIM i programskog okruženja Python/Keras	14
5. Povezivanje Python/Keras-a i VISSIM-a	18
6. Simulacijski rezultati	23
6.1. Model i podaci.....	23
6.2. Dobiveni rezultati.....	27
6.2.1. Protok, gustoća i brzina na promatranj dionici l	30
6.2.2. VSLC - ograničenja brzine za dionicu autoceste l	32
6.2.3. Duljina repa čekanja na rampir ₂	34
6.2.4. Funkcija nagrade/kazne algoritma	35
7. Zaključak	37
Popis kratica	39
Literatura.....	40
Popis slika	42
Popis tablica	43
Popis grafikona.....	44

1. Uvod

U suvremenom dobu 21. stoljeća velika se pozornost pridodaje mobilnosti u svakodnevnom životu u gradovima. Nerijetko ljudi moraju putovati s jednog kraja grada u drugi radi raznih privatnih i poslovnih razloga. Uzevši u obzir trend porasta osobnih vozila na području Republike Hrvatske, zaključujemo kako je osobni automobil glavno prijevozno sredstvo u gradovima na ovom području [1]. Povećanjem broja vozila koja se koriste u privatne svrhe, povećava se i prometna potražnja u gradovima. Povećanje prometne potražnje izravno uzrokuje nedostatak kapaciteta postojećih prometnica u gradovima, pogotovo u vršnim satima tijekom radnog tjedna ili većeg broja putovanja tijekom blagdana odnosno godišnjih odmora. Radi naglog porasta broja vozila u gradovima zadnjih godina, širenje i izgradnja nove prometne infrastrukture unutar gradova nije dovoljno dobro ni dostatno rješenje radi manjka prostora i neuklapanja u urbanističke planove gradova.

Jedno od mogućih i tehnički izvedivih rješenja je izgradnja gradskih obilaznica koje još nazivamo i gradskim autocestama. Izgradnjom gradskih autocesta rasterećuje se prometna potražnja na gradskim prometnicama. Gradske autoceste također imaju svrhu preuzimanja tranzitnog prometa koji nužno ne mora prolaziti gradskim prometnicama kroz samo središte grada. Gradske autoceste su projektirane i izgrađene kako bi osigurale i pružale veći prometni kapacitet te veće brzine kretanja vozila u odnosu na gradske prometnice.

Unatoč izgradnji gradskih autocesta u cilju smanjenja prometnog opterećenja i potražnje za kapacitetom gradskih prometnica, u vršnim satima se sve češće dešava zagušenje na gradskim autocestama. Preopterećenje gradske autoceste se najčešće javlja kod gradskih autocesta s velikim brojem bliskih ulaznih i izlaznih rampi. S obzirom na to da su ulazne i izlazne rampe izravno povezane s gradskim područjima, one predstavljaju velike generatore prometne potražnje prema kapacitetu glavnog toka gradske autoceste. Radi česte nemogućnosti i financijske neisplativosti širenja postojećih gradskih autocesta, javlja se potreba za pronalaskom rješenja pomoću pametnog upravljanja prometom. Moguće rješenje dolazi iz domene inteligentnih transportnih sustava (ITS) implementacijom upravljačkih i nadzornih sustava. Takvim sustavima primarni je cilj podizanje razine uslužnosti na gradskoj autocesti i povećanje sigurnosti prometa, ali i informiranje korisnika te smanjenje emisija štetnih plinova.

Jedan od takvih sustava je upravljanje promjenjivim ograničenjem brzine (engl. *Variable Speed Limit Control* - VSLC) na gradskim autocestama. Njegova je svrha promjena ograničenja brzine na pojedinim dijelovima autoceste u svrhu harmoniziranja prometnog toka i smanjenja zagušenja. Postoji mnogo algoritama na kojima se zasniva upravljačka logika navedenih sustava. Danas su sve više zastupljeni algoritmi koji su zasnovani na strojnom odnosno najnovije dubokom učenju. U ovom radu testiran je algoritam za upravljanje promjenjivim ograničenjem brzine koji je zasnovan na dubokom učenju primjenom realističnog modela upravljanje dionice gradske autoceste.

U svrhu testiranja efikasnosti ovakvih rješenja prije implementacije u stvarno okruženje potrebno je takav sustav testirati korištenjem odgovarajućih programskih alata. U

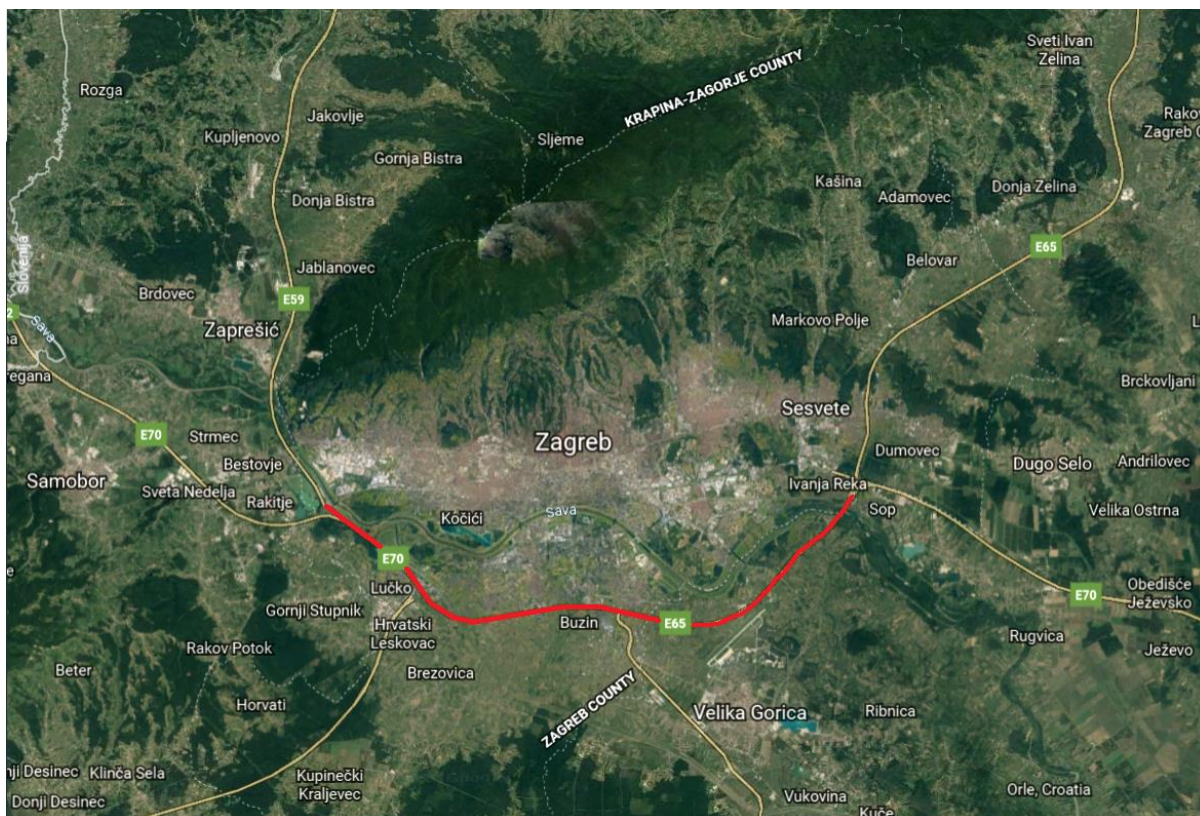
ovom diplomskom radu napravljeno je simulacijsko okruženje korištenjem programskog alata VISSIM i programskog okruženja Python/Keras unutar kojeg je testiran algoritam VSLC zasnovan na dubokom učenju. Svrha ovog diplomskog rada je pružiti uvid u okruženje za simulaciju sustava VSLC-a na osnovi kojeg se vrši provjera i testiranje upravljačkog algoritma zasnovanog na dubokom učenju. Dodatna svrha je provjeriti mogućnosti primjene mikroskopskog simulatora VISSIM i programskog okruženja Python/Keras za analizu VSLC sustava zasnovanih na dubokom učenju. Cilj ovog diplomskog rada je analiza VSLC algoritma zasnovanog na dubokom učenju implementiranog u razvijenom okruženju, što podrazumijeva detaljan opis implementacije sustava u simulacijsko okruženje i testiranje istog.

Rad je podijeljen u sedam poglavlja. U drugom poglavlju je opisana problematika upravljanja promjenjivim ograničenjem brzine. Treće poglavlje sadrži osvrt na primjenu strojnog učenja i različitih metoda ojačanog učenja u upravljanju promjenjivim ograničenjem brzine. U tom poglavlju napravljen je kratak pregled osnovnih pojmova o ojačanom učenju i algoritmima koji se danas koriste. Na kraju tog poglavlja opisan je implementacijski postupak VSLC sustava zasnovanog na učenju dualne duboke Q-mreže, koji je kasnije implementiran u programskom okruženju Python/Keras-u u obliku programskog kôda te testiran u simulacijskom okruženju VISSIM. U četvrtom poglavlju dan je kratak opis mikroskopskog simulatora VISSIM i programskog jezika Python, te knjižnice Keras za duboko učenje unutar Python-a. Peto poglavlje objašnjava na koji način su povezani programski alat i Python skripta koji zajedno čine okruženje za simulaciju VSLC-a zasnovano na dubokom učenju. U šestom poglavlju je opisan testni prometni model gradske autoceste s ulaznim podacima u simulatoru, detaljnije su definirane postavke učenja u algoritmu, te su na kraju prikazani rezultati simulacije testiranog sustava s analizom istih. Sedmo poglavlje odnosi se na zaključak i spoznaje do kojih se došlo tijekom izrade diplomskog rada. Također sadrži prijedloge za nastavak istraživanja na ovoj temi.

2. Problem upravljanja promjenjivim ograničenjem brzine

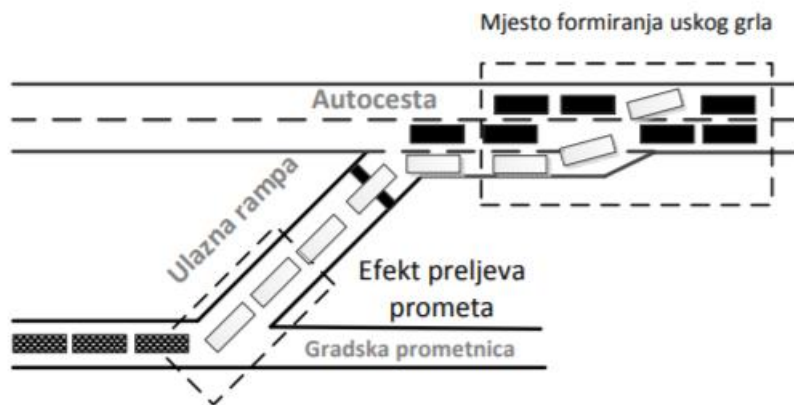
Gradske autoceste su prometnice koje ne sadrže infrastrukturu kao što su semafori i raskrižja. Konstruirane su kako bi pružale mogućnost kretanja većim brzinama u odnosu na gradske prometnice. Gradske autoceste pružaju veći prometni kapacitet i smanjuju opterećenje gradskih prometnica. Samim tim gradske autoceste pružaju veću razinu uslužnosti (engl. *Level of Services* - LoS) u odnosu na gradske prometnice. No unatoč većoj razini uslužnosti autoceste, sve većim rastom broja vozila povećava se pojava prometnih zagušenja, produljuje se vrijeme čekanja i smanjuje sigurnost na prometnicama i autocestama. Prilikom pojave zagušenja i povećanjem vremena čekanja povećava se potrošnja goriva, što uzrokuje povećanje emisije štetnih ispušnih plinova, odnosno nepotrebno zagađenje okoliša [2].

Gradske autoceste se najčešće nalaze na rubnim dijelovima gradova. U gradu Zagrebu izgrađena je zagrebačka obilaznica prikazana na slici 1. Obilaznica prati grad od zapadnog ulaza u grad (Jankomir) do Ivanje Reke na istoku. Obilaznica dva puta prelazi rijeku Savu i sastoji se od mosta preko kanala za prelijevanje poplavne vode. Zagrebačka obilaznica spaja i bliske gradove kao što su Velika Gorica, Sveta Nedjelja, Zaprešić itd. Budući da najprometnija dionica između čvorova Jankomir i Buzin ima prosječni godišnji dnevni promet (engl. *Annual Average Daily Traffic* – AADT) od oko 45.000 vozila, predstavlja najkorišteniju dionicu autoceste u Hrvatskoj [3].



Slika 1. Zagrebačka obilaznica

Izgradnja velikog broja ulaznih i izlaznih rampi rezultira dobrom povezanosti s gradskom prometnom mrežom. Tijekom vršnih sati radi povećanja prometnih zahtjeva prometna potražnja često se približava maksimalnom kapacitetu gradske autoceste. U slučaju kada prometna potražnja premaši kapacitet prometnice dolazi do pojave zagušenja u prometnom toku, a ponekad dovodi i do potpunog zastoja prometa. Stohastička priroda ponašanja prometa otežava točno predviđanje ponašanja prometa. Zagušenje se najčešće javlja u blizini ulaznih rampi, odnosno u blizini ulaza na autocestu. Mjesto gdje se susreću prometni tok s ulazne rampe i glavni prometni tok naziva se nizvodno usko grlo [4]. Na slici 2 prikazan je slučaj pojave nizvodnog uskog grla (engl. *Downstream Bottleneck*) u neposrednoj blizini ulazne rampe, te učinak preljeva prometa na lokalnu cestu. Usred nastanka zagušenja na dionici autoceste gdje nastaje nizvodno usko grlo dolazi do stvaranja zagušenja koje se širi kao šok val usmjeren unazad, uzduž glavnog toka autoceste suprotno smjeru kretanja glavnog prometnog toka. U tom slučaju dolazi do smanjenja brzine i povećanja vremena putovanja na dionici autoceste, što uvelike smanjuje sigurnost u prometu [4].



Slika 2. Mjesto nastanka uskog grla i posljedice preljeva prometa

Izvor: [4]

Tijekom dimenzioniranja kapaciteta autocesta teško je točno predvidjeti buduće promjene prometne potražnje na glavnom toku i pripadnim priljevnim tokovima. Neizravno izgradnja takve prometnice uvijek privlači i gospodarske subjekte (trgovačke centre, tvornice, skladišta itd.) u neposrednu blizinu zbog bolje prometne povezanosti. Posljedično tome povećava se prometna potražnja za kapacitetom gradske autoceste. Poseban naglasak daje se budućem porastu prometne potražnje koja će utjecati na propusnost glavnog toka na mjestima gdje se nalaze ulazne rampe, odnosno gdje se priljevni tokovi spajaju s glavnim tokom gradske autoceste. Zagušenja na autocestama najčešće se pojavljuju na područjima gdje se nalazi velik broj ulaznih i izlaznih rampi na malim međusobnim udaljenostima. Prometni zastoji su izraženiji na dijelovima autoceste u blizini velikih gradskih područja tijekom ranih jutarnjih i kasnih poslijepodnevih (vršnih) sati. Problemi nastaju zbog dnevnih migracija na relaciji privatne/poslovne obaveze i doma. Tijekom tog perioda značajno je narušena propusnost i sigurnost prometnih sudionika na gradskoj autocesti [2].

Upravljanje prometom na gradskim autocestama moguće je na osnovi nekoliko kriterija. Sigurnost i upravljanje sigurnošću prometa predstavljaju najvažnije kriterije za

upravljanje prometa na autocestama. U svrhu smanjenja zagušenja potrebno je i upravljati protočnošću prometa. Upravljanje protočnošću je glavni kriterij za smanjenje zagušenja u ovom radu. Prilikom takve vrste upravljanja susrećemo se s dva glavna problema koja uzrokuju zagušenje, a to su nehomogene brzine kretanja vozila na glavnom toku i priljevni tokovi prometa. U slučaju veće potražnje za nekom rutom, u urbanoj sredini, dolazi do zagušenja na ulaznoj rampi te novog zagušenja na autocesti oko ulazne rampe. Posljedično se produljuje vrijeme putovanja svim vozilima na pripadnoj dionici autoceste. Kako se prilikom dimenzioniranja kapaciteta autocesta teško može predvidjeti prometna potražnja na priljevnim tokovima, potrebno je pronaći druga rješenja za dani problem korištenjem upravljačkih pristupa iz domene ITS-a [2].

Poboljšanje kvalitete usluge prometnice moguće je postići primjenom naprednih upravljačkih sustava iz domene ITS-a, kao što je dinamičko upravljanje prometom. Ovi sustavi zasnivaju se na razmjeni podataka u stvarnom vremenu prikupljenih s osjetila (engl. *Detectors*) postavljenih na prometnoj infrastrukturi. Da bi razmjena podataka bila moguća, potrebna je odgovarajuća informacijsko-komunikacijska infrastruktura putem koje se podaci šalju, prikupljaju i analiziraju te razmjenjuju između korisnika i pružatelja usluga unutar sustava. Dva pristupa koriste se prilikom dinamičkog upravljanja prometom pomoću kojih je moguće upravljati prometom na gradskoj autocesti prilikom povećane prometne potražnje.

Prvi pristup upravljanja prometom tzv. upravljanje priljevnim tokom (engl. *Ramp Metering* - RM) zasniva se na upravljanju količinom vozila na ulaznim rampama koja se priključuju u glavni tok gradske autoceste. Ovim pristupom regulira se broj vozila koja se uključuju na glavni prometni tok autoceste (npr. pomoću algoritma ALINEA [5]). Drugi pristup se zasniva na VSLC sustavu. VSLC sustav izravno upravlja ponašanjem prometnog toka vozila na gradskim autocestama korištenjem promjenjivih znakova (engl. *Variable Message Signs* - VMS), to jest izmjenom ograničenja brzina. Korištenjem odgovarajućeg algoritma moguće je mijenjati vrijednosti ograničenja brzine u odnosu na trenutnu prometnu situaciju. Ako dođe do preopterećenja glavnog toka gradske autoceste zbog povećane prometne potražnje s priljevne rampe stvara se zagušenje i postoji mogućnost stvaranja prometnog čepa. Sigurnosni problem se javlja na mjestima gdje vozila s velikim prilazećim brzinama dolaze do kraja repa čekanja zbog potrebe za naglim smanjenjem brzine što uzrokuje povećanjem rizika od prometnih nesreća. Nastanak zagušenja uzrokuje usporavanje prometnog toka, te uz povećanje rizika od sudara unosi nezadovoljstvo vozačima jer im onemogućuje putovanje ujednačenom brzinom.

VSLC sustavi se najčešće dijele na dvije grupe [2]: (i) VSLC sustav zasnovan na vremenskim (meteorološkim) uvjetima; i (ii) VSLC sustav zasnovan na zagušenjima. Prvi sustav se koristi na gradskim autocestama gdje nepovoljni vremenski uvjeti stvaraju sigurnosni problem u sigurnom odvijanju prometa. Prilikom pogoršavanja vremenskih uvjeta do konfigurirane kritične točke u sustavu smanjuje se ograničenje brzine na prikladnu brzinu u cilju smanjenja mogućnosti nastanka incidentnih situacija. Takvi sustavi su česti u nordijskim zemljama s mnoštvom snježnih padalina tijekom zime. Druga vrsta sustava se koristi prilikom povećanja prometnog volumena i pojave zagušenja. Upravljačka strategija kojom se smanjuje ograničenje brzine aktivira se kada su volumen i/ili gustoća toka prešli

određenu konfiguriranu granicu. Smanjenje brzine ograničenja može se postaviti automatski prema određenom algoritmu upravljanja ili ručno preko centra za upravljanje autocestom. U ovom radu se analizira automatski pristup promjene ograničenja brzine zasnovan na upravljačkoj logici unutar algoritma zasnovanog na dubokom učenju. Glavni operativni razlozi uvođenja VSLC sustava na autocesti su [6]:

- Smanjenje vremena putovanja,
- Pouzdanija vremena putovanja,
- Smanjenje broja pada prosječne brzine glavnog toka,
- Smanjenje emisije ispušnih plinova,
- Veća iskorištenost prometnih traka,
- Ujednačeniji vremenski interval slijeđenja vozila,
- Smanjenje broja incidentnih situacija.

VSLC tehnika upravljanja prometom za autoceste se već primjenjuje i koristi se npr. na njemačkim autocestama (njem. *Autobahn*), na prometnici M26 u Velikoj Britaniji i u ostalim zemljama u Europi. VSLC ima pozitivan učinak na sigurnost prometa uslijed smanjenja brzine i homogenizacije brzine, koji su u korelaciji sa smanjenjem vjerojatnosti nastanka prometne nesreće. Višegodišnje procjene utjecaja VSLC-a na sigurnost prometa ukazuju na smanjenje broja nesreća za čak 20-30% nakon implementacije VSLC-a. Neki od prijašnjih znanstvenih radova na temu upravljanja VSLC-om dostigli su određene zaključke o dobrobiti VSLC-a kao što su [6] i [7].

3. Primjena dubokog učenja u upravljanju promjenjivim ograničenjem brzine

Znanstvena zajednica i struka sve više proučavaju rješenja prometnih problema koje računalo može rješavati. Računalo može izvršiti i izračunati rješenja dovoljno dobro u stvarnom vremenu, a da istodobno imaju mogućnost prilagodbe i učenja tijekom primjene. Razvitkom tehnike i tehnologije, razvijeno je nekoliko metoda na temelju koje računalo može obavljati i pronalaziti rješenja. Jedna od tih metoda je ojačano učenje. Ojačano učenje može se interpretirati kao skupina različitih vrsta algoritama koji uče izravno iz interakcije s okolinom. Okolina može biti stvarni svijet, računalna igra, simulacija ili čak igra na ploči poput šaha. Poput čovjeka, agent uči iz posljedica svojih akcija, a ne iz toga što mu je eksplicitno zadano pravilima i formulama. Postoji više metoda ojačanog učenja a neke od metoda su; Q-učenje (engl. *Q-learning*), učenje duboke Q-mreže (engl. *Deep Q-network* - DQN) i učenje dualne duboke Q-mreže (engl. *Dual Deep Q-network* - DDQN).

3.1. Ojačano učenje

Ojačano učenje (engl. *Reinforcement Learning* - RL) je grana strojnog učenja kod koje sustavi uče izravno iz podataka dobivenih iz okoline tijekom rada. Zasniva se na agentima koji su u interakciji sa svojom okolinom. Za razliku od klasičnih pristupa strojnog učenja, agent unutar RL-a mora biti sposoban sam učiti iz vlastitih iskustava metodom pokušaja i pogrešaka. Cilj učenja agenta je povećanje kumulativne nagrade tijekom učenja u definiranom okruženju. Nagrada predstavlja povratnu informaciju koju agent prima nakon izvršenja pojedine akcije [8].

Okolina u kojoj se nalazi agent opisana je i ima svojstva Markovljevog procesa odlučivanja (engl. *Markov Decision Process* - MDP). Njime je definiran skup stanja i akcija koje se mogu izvršiti u svrhu upravljanja okolinom. Praktični problemi poput balansiranja letjelice, hoda robota i upravljanja bespilotnom letjelicom mogu se modelirati kao MDP. Takvo učenje gdje je model okoline poznat naziva se učenje zasnovano na modelu (engl. *Model-Based Learning*) [9].

U slučaju kada je okolina nepoznata, učenje se naziva učenje bez modela (engl. *Model-Free Learning*). Takvi modeli uče optimalni upravljački zakon (engl. *Policy*) na osnovi iskustva, odnosno dobivene nagrade nakon primjene akcija. Prethodno spomenuti praktični problemi se ponekad zbog složenosti okoline ne opisuju kao MDP. Umjesto toga opisani su prikupljenim nizovima uzoraka stečenih iskustvom koje agent prima izravno iz okoline [9].

3.1.1. Q-učenje

Algoritam Q-učenja predstavlja *off-policy* učenje. Konvergencija Q funkcije zasniva se na sljedećoj hipotezi. Ako agent obiđe svaki par stanja i akcija beskonačno puta, tada prema [9] algoritam Q-učenja konvergira prema optimalnoj funkciji stanja i akcije (engl. *action-state*) (1).

$$Q(s, a) \rightarrow q_*(s, a) \quad (1)$$

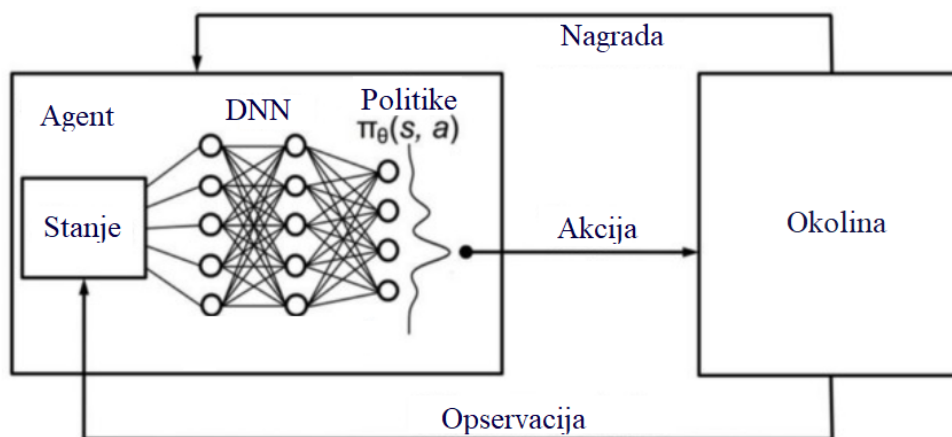
Model okoline kod ovog algoritma nije potrebno poznavati. Informacije o okolini agent dobiva preko nagrada ili kazni putem povratne veze gdje je poduzeta akcija u stvari reakcija okoline. Kod Q-učenja agent može učiti promatrajući druge agente prilikom izvođenja akcija. Također ima sposobnost korištenja stečenih iskustva iz prijašnjih upravljačkih zakona $\pi_1, \pi_2, \dots, \pi_{\pi-1}$ u svrhu unaprjeđenja učenja. Algoritam Q-učenja ima sposobnost istraživanja i istovremeno učenja optimalnih upravljačkih zakona. Iterativni proces ažuriranja Q funkcije kod Q-učenja dan je izrazom (2) [10]:

$$Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a)). \quad (2)$$

Algoritam Q-učenja nakon odabrane akcije π i preslikavanja na osnovi te akcije u novo stanje $s \rightarrow s'$, gleda još korak unaprijed kako bi odabrao odgovarajuću akciju a' koja daje najveću nagradu za korak unaprijed. Na osnovi ta dva koraka ažurira se vrijednost Q funkcije za stanje s i akciju a . No algoritam ne mora uvijek slijediti optimalni upravljački zakon iz stanja s' , bitno je samo da osvježi Q funkciju na poziciji $Q(s, a)$ s tim vrijednostima, te se zato još naziva *off-policy* algoritam.

3.1.2. Algoritam učenja na principu duboke Q-mreže

Duboka Q-mreža je naziv za algoritam koji integrira rad konvolucijske neuronske mreže (engl. *Convolutional Neural Network* - CNN) i Q-učenja, gdje CNN predstavlja aproksimator Q-tablice stanja i akcija. Kod DQN-a neuronska mreža uzima ulogu naučenog preslikavanja stanja i akcija. Neuronska mreža ima izravno djelovanje na okolinu. Promatra trenutno stanje okoline i na osnovi prikupljenih podataka daje odluku koju će akciju poduzeti na osnovi trenutnog stanja i dosadašnjih iskustava. Na osnovi poduzete akcije agent dobiva nagradu. Visina nagrade određuje kvalitetu poduzetih akcija s obzirom na rješavanje zadanog stanja. Cilj agenta je naučiti poduzimati akcije u bilo kojim okolnostima koje maksimiziraju akumuliranu nagradu tijekom vremena. Shematski prikaz dubokog ojačanog učenja prikazan je na slici 3 [11], [12].



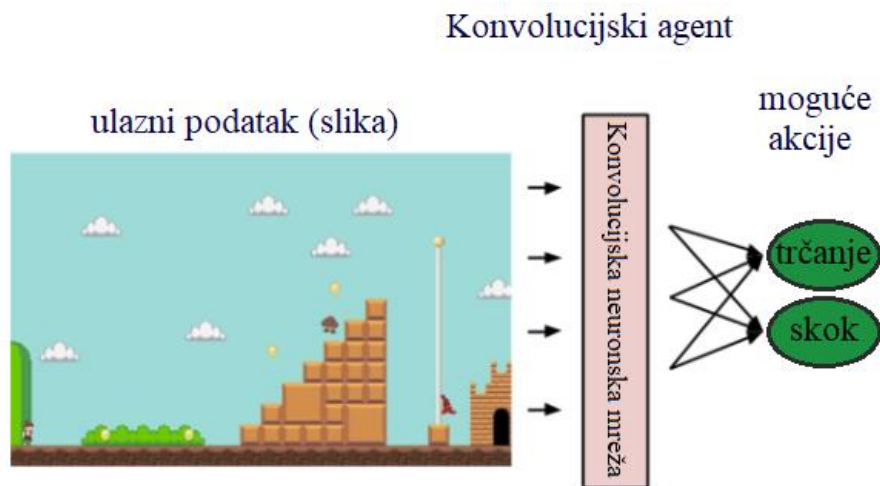
DNN - Duboka neuronska mreža

Slika 3. Shematski prikaz dubokog ojačanog učenja

Izvor: [11]

Neuronska mreža je kod DQN-a sredstvo koje uči mapirati parove stanja i akcija u nagradu. Kao i sve neuronske mreže, one koriste koeficijente za aproksimaciju funkcije koja se odnosi na ulaze i izlaze, a njihovo se učenje sastoji od pronalaženja pravih koeficijenata ili težina. Tako se iterativno podešavaju težine duž gradijenata koji zatim daju manju grešku. Kod ojačanog učenja mogu se koristiti neuronske mreže za prepoznavanje stanja agenata; na primjer zaslon na kojem je lik iz igrice, ili teren ispod bespilotne letjelice. Odnosno, one obavljaju zadatak prepoznavanja slike. Konvolucijske neuronske mreže dobivaju različita tumačenja od slika u ojačanom učenju nego u učenju pod nadzorom. Mreža primjenjuje oznaku na sliku, to jest, postavlja imena pikselima [13].

Kod učenja DQN-a, imajući na umu da slika predstavlja stanje, konvolucijska neuronska mreža može rangirati radnje koje je moguće izvesti u tom stanju. Na primjer, može predvidjeti da će trčanje udesno vratiti nagradu od 5 bodova, skok 7 i trčanje ulijevo nijedan u nekoj računalnoj igri, kao što je prikazano na slici 4 [13].



Slika 4. Ilustracija mapiranja stanja agenta

Izvor: [13]

Akcija je skup svih mogućih poteza koje agent može izvršiti. Radnja je gotovo nerazumljiva za agenta, ali treba napomenuti da agent bira među popisom mogućih radnji. U video igrama popis može uključivati trčanje udesno ili lijevo, skakanje uvis, naginjanje ili stajanje. Popis radnji na burzama može uključivati kupnju, prodaju ili držanje bilo kojeg od vrijednosnih papira i njihovih derivata. Pri rukovanju zračnim bespilotnim letjelicama, alternative bi uključivale mnogo različitih brzina i ubrzanja u trodimenzionalnom prostoru [13]. Okolina uzima trenutno stanje i radnje agenta kao ulaz i kao rezultat vraća nagradu agenta i njegovo sljedeće stanje. Okolina mogu biti zakoni fizike i pravila društva koja obrađuju postupke pojedinca i utvrđuju njihove posljedice. Stanje je konkretna i neposredna situacija u kojoj se agent nalazi, tj. određeno mjesto i trenutak, trenutna konfiguracija koja stavlja agenta u odnosu na druge značajne stvari kao što su prepreke, neprijatelji ili nagrade. Stanje može biti trenutna situacija koju vraća okolina ili bilo koja buduća situacija. Nagrada je povratna informacija kojom se mjeri uspjeh ili neuspjeh radnji koje poduzima agent. Iz bilo kojeg stanja, agent šalje izlaz u obliku radnji u okolinu, a okruženje vraća agentu novo stanje (koje je nastalo djelovanjem na prethodno stanje) kao i nagradu, ako je ima. Nagrada može

biti trenutačna ili odgođena. Nagrada učinkovito ocjenjuje djelovanje agenta. Upravljački zakon je strategija koju agent koristi za određivanje sljedeće akcije na osnovi trenutnog stanja. To preslikava stanja u akcije, radnje koje obećavaju najveću nagradu. Matematički gledano, naučeni upravljački zakon je distribucija u svim radnjama danim upravljačkim zakonom [13]. Pravilo određuje preslikavanje iz stanja s u akciju a koju mora poduzeti agent:

$$a = \pi(s), \quad (3)$$

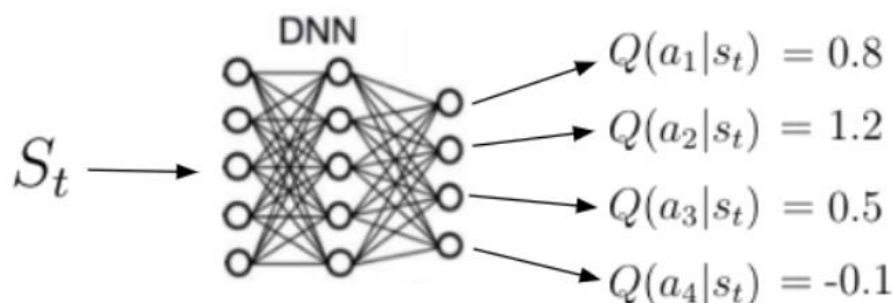
$$\pi(a|s) = \mathbb{P}[A_t = a|S_t = s], \quad (4)$$

gdje su:

- A – označava skup akcija,
- a – je pojedina akcija,
- S – je skup svih stanja,
- s – je pojedino stanje,
- π – je funkcija upravljačkog zakona.

Očekivani dugoročni povrat s pripadnim smanjenjem (popustom), za razliku od kratkoročne nagrade definira se kao očekivani dugoročni povrat trenutnog stanja u skladu s upravljačkim zakonom. Nagrade snižavamo ili smanjujemo njihovu procijenjenu vrijednost, što se dalje događa u budućnosti. Q-vrijednost se odnosi na dugoročni povrat trenutnog stanja i poduzimanja akcija pod određenim upravljačkim zakonom [13].

Funkcija vrijednosti akcije je očekivani povrat koji se dobiva započinjući u stanju s , poduzimajući akciju a slijedeći upravljački zakon π . Za stanje s , $q(s, a)$ se može pretpostaviti nekoliko vrijednosti jer agent može poduzeti bilo koju akciju u stanju s . Naučeno preslikavanje $Q(s, a)$ sprema se u neuronsku mrežu. Dajući stanje s kao ulaz, mreža izračunava kvalitetu za svaku moguću radnju u ovom stanju kao što je prikazano na slici 5. Veći broj na izlazu mreže znači i bolje djelovanje s obzirom na zadani cilj [11].



Slika 5. Prikaz funkcije vrijednosti akcije

Izvor: [11]

Q-vrijednost ili vrijednost akcije (Q) se odnosi na dugoročni povrat trenutnog stanja s , poduzimanja akcije a pod upravljačkim zakonom π . Q preslikava parove djelovanja upravljačkih zakona u najvišu kombinaciju neposrednih nagrada sa svim budućim nagradama koje bi mogle biti prikupljene kasnijim akcijama. Jednadžba za osvježavanje Q glasi (5):

$$Q(s_t, a_t) \leftarrow (1 - \alpha) * Q(s_t, a_t)' + \alpha * (r_t + \gamma * \max Q(s_{t+1}, a)), \quad (5)$$

gdje su:

- $Q(s_t, a_t)'$ – prethodno stanje,
- α – stopa učenja,
- r_t – nagrada,
- γ – stopa popusta odnosno smanjenja nagrade,
- $\max Q(s_{t+1}, a)$ – estimacija optimalne buduće vrijednosti.

Dodjeljujući vrijednosti očekivanim nagradama, Q-funkcija jednostavno odabire par stanja i akcije s najvišom takozvanom Q-vrijednošću. Na početku učenja DQN-a koeficijenti neuronske mreže mogu se inicijalizirati stohastički ili nasumično. Koristeći povratne informacije iz okoline, neuronska mreža može upotrijebiti razliku između očekivane nagrade i stvarne nagrade da bi prilagodila svoje težine te poboljšala tumačenje parova stanja i akcija. Ova petlja za povratne informacije analogna je uzvraćanju pogreške kod nadziranog učenja. Međutim, nadzirano učenje započinje poznavanjem oznaka temeljne istine koje neuronska mreža pokušava predvidjeti. Učenje DQN-a oslanja se na okolinu i šalje skalarni broj kao odgovor na svaku novu akciju. Nagrade koje okolina vraća mogu se mijenjati, odgađati ili utjecati na njih. To nas vodi do cjelovitijeg izraza Q-funkcije, koja uzima u obzir ne samo trenutne nagrade proizvedene djelovanjem, već i odgođene nagrade koje se mogu vratiti nekoliko vremenskih koraka dalje u nizu [13].

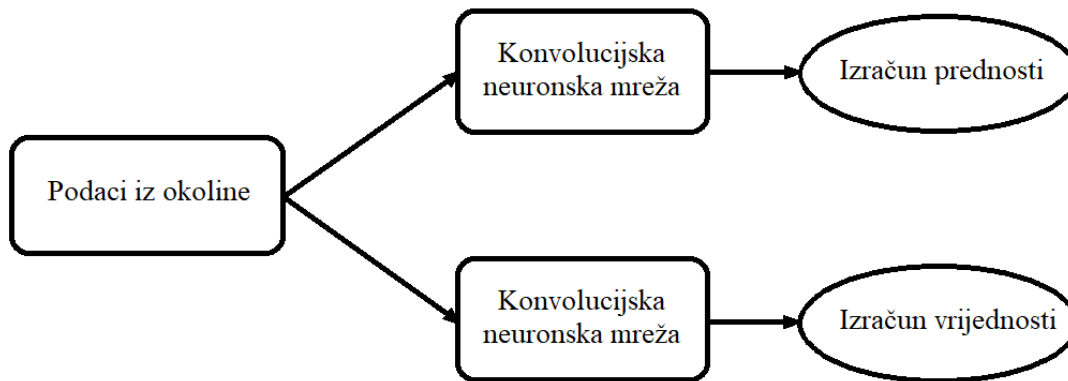
3.1.3. Algoritam učenja na principu dualne duboke Q-mreže

Poznato je da algoritam učenja duboke Q-mreže ponekad uči nerealno visoke vrijednosti akcija jer uključuje korak maksimizacije nad procijenjenim vrijednostima akcija. Među znanstvenom zajednicom još je otvoreno pitanje utječe li precjenjivanje vrijednosti na performanse agenata u praksi. Preoptimistične procjene vrijednosti ne moraju nužno predstavljati problem sam po sebi. Ako bi sve vrijednosti bile ujednačeno više, sačuvane su relativne preferencijalne akcije i ne bi bilo očekivano da rezultirajući upravljački zakon bude pogoršan. Ako međutim precjenjivanja nisu koncentrirana na stanja o kojima se želi više učiti, onda mogu negativno utjecati na kvalitetu upravljačkog zakona [12].

Algoritam učenja na principu DDQN-a algoritam je ojačanog učenja koji pokušava naučiti Q-vrijednost putem dva procjenitelja funkcija: jednog koji procjenjuje funkciju prednosti i drugog koji procjenjuje funkciju vrijednosti. Funkcija vrijednosti izračunava vrijednost određenog ulaza, a funkcija prednosti izračunava nagradu na osnovi izvršene akcije. Zajedno pružaju dobru procjenu uspješnosti sljedećeg ulaza s obzirom na određeni par akcija i stanja. Nadalje, ova strategija uči koji su ulazi vrijedni umjesto da uči iz posljedica reakcije na svaki od ulaza. U okruženjima poput ovih, ovaj način predstavlja veliku prednost jer za puno ulaza postupci ne određuju ishod. Često postoji samo jedna pogrešna akcija, pa ostale akcije imaju izravne posljedice. Ovakav algoritam bolje razumije koji su ulazi bolji u stvarnom vremenu, a ne nakon 20 ili 30 instanci. Dualna mrežna struktura pokazuje bolje rezultate od pojedinačnih mrežnih struktura kao što je DQN, jer se svaki procjenjivač

funkcija može usredotočiti na različit dio slike, koristeći različite strategije za stvaranje boljeg procjenitelja za Q funkciju [11], [13].

U pogledu strukture algoritma, napravljene su dvije neuronske mreže gdje jedna mjeri funkcije vrijednosti, a druga funkcije prednosti. Neuronska mreža koja mjeri funkcije prednosti naziva se ciljna mreža [12]. Pojednostavljena blok shema rada algoritma prikazan je na slici 6.



Slika 6. Pojednostavljena blok shema algoritma DDQN

Q-vrijednosti odgovaraju koliko je dobro biti u narednom stanju i poduzeti akciju u tom stanju $Q(s, a)$. Tako se $Q(s, a)$ može rastaviti kao zbroj [14]:

- $V(s)$ - vrijednost bivanja u stanju,
- $A(s, a)$ - nagrada za poduzimanje akcije u stanju (koliko je bolje poduzeti narednu akciju u odnosu na sve druge moguće akcije u stanju).

$$Q(s, a) = A(s, a) + V(s) \quad (6)$$

Pomoću DDQN-a se želi odvojiti mreža koji procjenjuje ova dva elementa na dvije mreže; ona koja procjenjuje vrijednost stanja $V(s)$ i ona koja procjenjuje nagradu za svaku akciju $A(s, a)$. Razdvajanjem se dobiva mogućnost izračunavanja $V(s)$. Ovo je posebno korisno za stanja u kojima njihove akcije ne utiču na okolinu na odgovarajući način. U ovom slučaju, nepotrebno je izračunavati vrijednost svake akcije.

3.2. Upravljanje promjenjivim ograničenjem brzine zasnovano na algoritmu učenja dualne duboke Q-mreže

Kao što je već navedeno, noviji pristup upravljanja promjenjivim ograničenjem brzine na gradskim autocestama zasniva se i na strojnom učenju. Regulator koji upravlja ograničenjem brzine može se poistovjetiti s agentom u strojnom učenju. Model okoline u kojoj agent djeluje je predstavljen prometnim tokom i dionicom gradske autoceste. Stanja su predstavljena osnovnim veličinama koje opisuju parametre prometnog toka. Akcije su opisane kao izvršne brzine ograničenja kojima agent upravlja i postavlja putem VMS-ova. Agent odabire akciju iz skupa konačnih akcija na osnovi stanja okoline. Ovisno o poduzetoj akciji okolina se mijenja, te agent (regulator) dolazi u novo stanje. Agent dobiva povratnu

informaciju u obliku nagrade odnosno kazne ovisno o tome koliko su dobre njegove poduzete akcije.

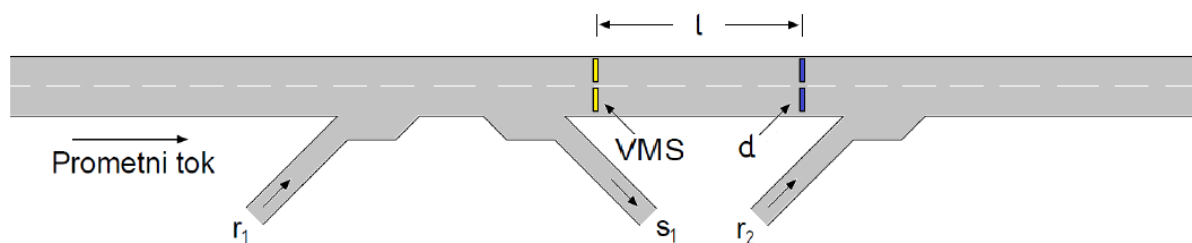
U nastavku će biti opisana implementacija regulatora za VSLC koji je zasnovan na algoritmu učenja dualne duboke neuronske Q-mreže. VSLC regulator stalno uzorkuje parametre okoline (parametri prometnog toka) te donosi odluke koju akciju primijeniti nakon kojih regulator dobiva određenu nagradu ili kaznu koja predstavlja mjeru uspješnosti poduzetih akcija. Podaci o prometnom toku prikupljaju se periodično u intervalima uzorkovanja podataka o vozilima $T_u = 60$ [s]. Prikupljeni prometni podaci računaju se u intervalima akcija $T = 300$ [s] (5 [min]). Cijela simulacija podijeljena je na korake koji traju $T_k = 0,1$ [s], odnosno u svakoj sekundi se napravi 10 koraka. Ovakvom izvedbom i brzinom kojom se algoritam izvršava dobiva se dojam da je simulacija kontinuirana (ne primjećuju se ti koraci, odnosno „rezolucija simulacije“). Atribut u skupu koji sadržava stanja je ograničenje brzine v_{VSLC} [$\frac{km}{h}$] na dionici s implementiranim sustavom VSLC. Vrijednosti za stanja ograničenja brzine kodirane su kako slijedi:

- 60 → 0,
- 70 → 1,
- 80 → 2,
- 90 → 3,
- 100 → 4,
- 110 → 5,
- 130 → 6.

Skup akcija kojima regulator raspolaže u svrhu upravljanja predstavljen je konačnim skupom (7) koji sadrži moguća ograničenja brzine. Nakon što regulator odabere ograničenje brzine, ono se postavlja na promjenjive prometne znakove na početku dionice l kao što je prikazano na slici 7.

$$A = \{60, 70, 80, 90, 100, 110, 130\} \quad (7)$$

Akcije su također kôdirane redom od 0 do 6 prema redoslijedu navođenja iznosa ograničenja brzina u izrazu (7). Na slici 7 dan je grafički prikaz modela gradske autoceste. Na slici 7 je vidljiva dionica l na početku koje se primjenjuje algoritam za upravljanje promjenjivim ograničenjem brzine.



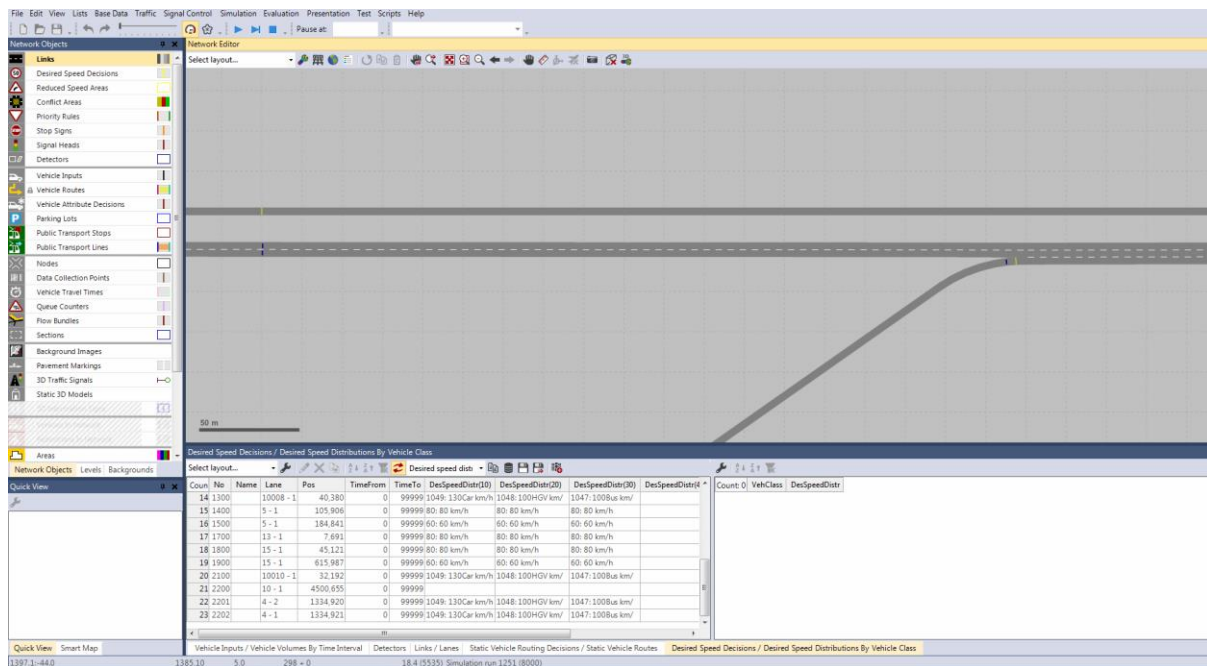
Slika 7. Grafički prikaz modela gradske autoceste

4. Značajke simulatora VISSIM i programskog okruženja Python/Keras

U ovom radu za izvedbu simulacija prometa korišten je mikrosimulacijski alat VISSIM tvrtke PTV. Kada se govori o simulacijama dijele se na mikroskopske i makroskopske simulacije, te na hibride spomenutih odnosno mezoskopske. Kod mikroskopskih simulacija svaki se prometni entitet (vozilo, osoba, biciklist, itd.) gleda kao jedinka, odnosno promatra se zasebno unutar simulacija. Makroskopske simulacije promatraju promet kao skup vozila koja se kreću uniformno, odnosno promatraju prometni tok. Ponašanje prometa na makroskopskoj razini možemo poistovjetiti s ponašanjem fluida prema pravilima mehanike fluida. VISSIM je programski alat koji vrši simulacije na mikroskopskoj razini. Zasniva se na Wiedemann-ovom modelu. Postoje dva Wiedemann-ova modela, Wiedemann 74 i Wiedemann 99. U Wiedemann-ovom modelu se pretpostavlja da reakcija vozača može biti jedan od četiri modela ponašanja vozila [15]:

- 1) slobodna vožnja, što znači da vozač u ovom slučaju ne percipira ostala vozila te vozi svojom individualno određenom brzinom,
- 2) prilazak drugom vozilu, koji predstavlja adaptivno upravljanje brzinom kretanja vozila na osnovi percepcije drugih vozila, odnosno smanjenje brzine prilikom prilaska drugom vozilu,
- 3) praćenje drugog vozila, gdje vozač prati vozilo ispred sebe i drži dostatan razmak među vozilima,
- 4) model kočenja, gdje vozač primjenjuje srednju ili visoku stopu usporavanja ako udaljenost među vozilima padne ispod željene sigurnosne udaljenosti.

U modelu vozača se također postavljaju postavke za promjenu traka i ponašanje vozila u traci. Postavke promjene traka odnose se na svojevolsnu promjenu traka, tj. na pretjecanje vozila i na nužnu promjenu traka što se odnosi na sužavanje ceste, odnosno praćenje zadane rute. U ovom radu korišten je Wiedemann-ov 99 model jer je prikladniji za simuliranje prometa na autocestama. Parametri koji se koriste u opisu kretanja vozila na razini svakog tipa vozila su: distribucija brzine pojedinog tipa vozila, ubrzanje i usporavanje, interval slijeđenja, razmak vozila, model prestrojavanja itd.. Sučelje simulacijskog alata VISSIM prikazano na slici 8 je veoma pristupačno i pregledno za korištenje.



Slika 8. Prikaz korisničkog sučelja simulatora VISSIM

Python je objektu orijentiran programski jezik, kojeg je razvio Guido van Rossum 1990. godine. Python kao programski jezik ne nudi nove revolucionarne značajke u programiranju, nego ujedinjuje sve najbolje principe i načela rada drugih programskih jezika. Vrlo je jednostavan i moćan programski jezik. Više nego drugi jezici on omogućuje programeru više vremena za razmišljanje o problemu nego o sintaksi jezika. U neku ruku može ga se smatrati hibridom: nalazi se između tradicionalnih skriptnih jezika (kao što su Tcl, Schema i Perl) i sistemskih jezika (kao što su C, C++ i Java). To znači da nudi jednostavnost i lako korištenje skriptnih jezika, uz napredne programske alate koji se tipično nalaze u sistemskim razvojnim jezicima. Python je besplatan, s izuzetno dobrom potporom, literaturom i dokumentacijom [16]. Jezične značajke Python-a su [16]:

1) Interpretacija međukôda;

Python kôd živi u tekst datotekama koje završavaju nastavkom „.py“. Program prevodi kôd u niz bytecodeova koji se spremaju u „.pyc“ datoteke koje su prenosive na bilo koje platforme gdje se mogu izvoditi interpretacijom tog međukôda. Brzina izvođenja Python kôda istog je reda veličine kao u Javi ili Perlu. Python je napisan u ANSI-C i dostupan je za cijeli niz računala i operacijskih sustava uključujući Windows, Unix/Linux i Macintosh.

2) Jezik visoke razine;

osim standardnih tipova podataka (brojevi, nizovi znakova i sl.) Python ima ugrađene tipove podataka visoke razine kao što su liste, n-terci i rječnici.

3) Interaktivnost;

Python se može izvoditi u različitim okruženjima. Za razvitak programa najbolji je interaktivni način rada u kojem se programski kôd piše naredbu za naredbom. Ne postoji razlika u razvojnom i izvedbenom (engl. *runtime*) okolišu.

4) Čista sintaksa;

Sintaksa jezika je jednostavna i vidljiva. Uvlake zamjenjuju posebne znakove za

definiranje blokova kôda, pa je program vrlo pregledan i jednostavan za čitanje.

5) Napredne značajke jezika;

Python nudi sve značajke kao i kod suvremenog programskog jezika: objektu orijentirano programiranje s višestrukim nasljeđivanjem, dohvatanje izuzetaka, redefiniranje standardnih operatora, pretpostavljene argumente, prostore imena i pakete.

6) Proširivost;

Python je pisan u modularnoj C arhitekturi. Zato se može lako proširivati novi značajkama ili tzv. API-ima (engl. *Application Programming Interface*).

7) Bogate knjižnice programa;

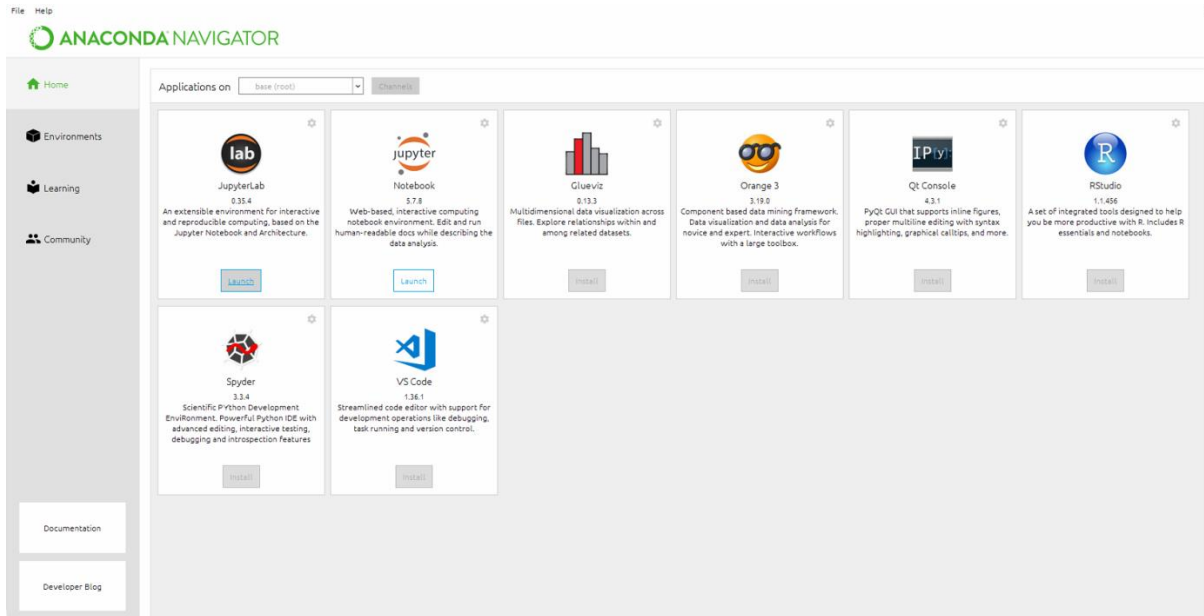
Python-ova knjižnica (engl. *library*) uključuje standardnu instalaciju i preko 200 modula, što pokriva sve od funkcija operacijskog sustava do struktura podataka. Glavna Python internetska stranica (www.python.org) nudi sažeti popis mnogih Python projekata i različitih drugih knjižnica.

8) Potpora;

Python ima veliku zajednicu korisnika koja rapidno raste iz godine u godinu.

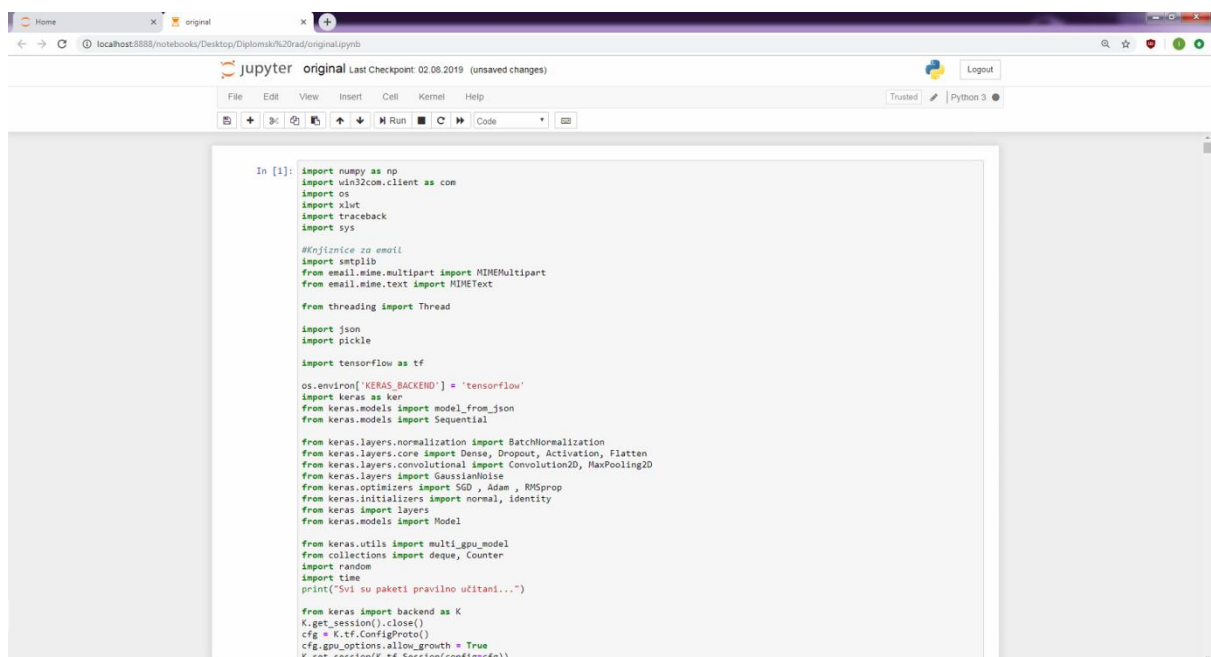
Keras je knjižnica unutar programskog jezika Python koja se koristi za dizajniranje i programiranje neuronskih mreža kod primjene dubokog učenja. Dizajniran je kako bi omogućio eksperimentiranje s dubokim neuronskim mrežama, usredotočen je na pristupačnost korisnicima, modularnost i proširivost. Razvijen je kao dio istraživačkog napora projekta ONEIROS (engl. *Open-end Neuro-Electronic Intelligent Robot Operating System*). Nudi intuitivniji skup apstrakcija na višoj razini koji olakšava razvoj modela dubokog učenja [17].

U ovom radu programski jezik Python i knjižnica Keras instalirani su i pokretani putem platforme Anaconda. Anaconda je platforma otvorenog kôda i predstavlja najlakši način za strojno učenje. Anaconda omogućava korisnicima brzo i jednostavno preuzimanje više od 1.500 Python paketa kao što su upravljanje knjižnicama, ovisnostima i okruženjima, razvijanje i izradu modela strojnog učenja i dubokog učenja, analizu podatka i vizualizaciju rezultata. Sučelje Anaconda platforme je veoma jednostavno i pregledno kao što je prikazano na slici 9.



Slika 9. Korisničko sučelje platforme Anaconda

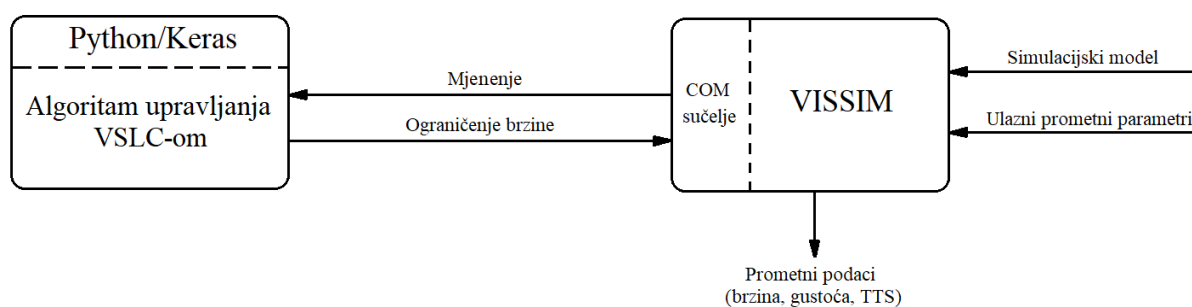
Unutar platforme putem web aplikacije jupyter Notebook otvara se okruženje gdje se piše i izvršava kôd programskog jezika Python. Spomenuto okruženje je veoma jednostavno i pregledno za korištenje kako je prikazano na slici 10.



Slika 10. Korisničko sučelje jupyter Notebook

5. Povezivanje Python/Keras-a i VISSIM-a

Model gradske autoceste napravljen je unutar mikroskopskog simulatora VISSIM u svrhu testiranja upravljačkog algoritma. Upravljački algoritam implementiran je unutar skripte napisane u programskom jeziku Python korištenjem knjižnice Keras. Povezivanje skripte s algoritmom učenja i simulatora vrši se preko COM (engl. *Component Object Model*) sučelja, čime se omogućuje komunikacija i upravljanje. Povezivanjem preko spomenutog sučelja moguće je upravljati prometnom simulacijom pomoću programskog kôda u programskom jeziku Python. Simulacija je u potpunosti upravljana pomoću Python skripte. Podaci o prometnom toku (brzina, gustoća, ukupno provedeno vrijeme svih vozila u prometnoj mreži (engl. *Total Time Spent* – TTS)) prikupljaju se i računaju pomoću funkcija izrađenih u programskoj skripti u Python-u koje pristupaju osjetilima u VISSIM-u. VMS-ovi u VISSIM-u izravno zaprimaju odluke o ograničenju brzine koje predlaže algoritam VSLC-a, te primjenjuju donesene odluke od strane algoritma. Na slici 11 prikazana je blokovska shema simulacijskog okruženja za testiranje VSLC-a.



Slika 11. Python/Keras-VISSIM simulacijska shema

Proces simuliranja počinje pokretanjem Python skripte. Skripta prvo učitava sve potrebne knjižnice unutar programskog jezika, zatim slijedi inicijalizacija VISSIM modela, uspostava COM sučelja između oba programa i na kraju se učitava VISSIM model (*inpx* i *layx* datoteke). Prva datoteka sadrži podatke o parametrima mikrosimulacijskog modela kao što su poveznice (engl. *Links*) od kojih je izgrađen model, iznose zadane prometne potražnje, podatke o tipovima vozila, distribucije brzina pojedinog tipa vozila, model slijeđenja vozila, model prestrojavanja vozila itd.. Datoteka s nastavkom *layx* sadrži podatke o okolini koja se nalazi oko same prometne mreže kao što su npr. veličina radnog sučelja, povećanje pojedinog dijela mreže, izbor atributa u radnom sučelju itd.. Nakon inicijalizacije VISSIM modela i učitavanja knjižnica poput Keras-a koja je potrebna za izradu i učenje neuronske mreže, otvara se sučelje simulatora VISSIM. Idući korak definira ulazne varijable poput ukupnog broja simulacija, trajanja opservacija, trajanja učenja mreže, simulacijska ograničenja, itd.. Pomoću različitih ulaznih vrijednosti i ograničenja, napravljen je algoritam za generiranje vozila za svaku simulaciju čine se postiče efekt stohastičkog ponašanja prometnog toka. Također unutar skripte izrađene su konvolucijske neuronske mreže koje rade na principu algoritma DDQN, koje uče na osnovi podataka iz prometa za vrijeme opservacije, gdje mreža ne poduzima akcije, i za vrijeme učenja mreže kada ona zapravo upravlja VMS-ovima. Unutar skripte napravljen je algoritam DDQN, algoritam nagrade i buduće nagrade. S

izvršavanjem naredbi u Python skripti dolazi se do glavne petlje unutar koje se pokreće simulacija u VISSIM-u. Svaka iteracija u toj petlji pokrene simulaciju za jedan vremenski korak (T_k) simulatora. Jedna iteracija simulacije simulira $T_s = 7.200$ [s] kretanja prometnog toka u stvarnom vremenu. Unutar svake od akcija u iteraciji, algoritam izračunava srednju brzinu i gustoću svih vozila prikupljenih na osjetilu koje je udaljeno jedan kilometar od postavljenog i upravljano VMS-a, prosječan minutni TTS, i red vozila na drugoj ulaznoj rampi. Pojednostavljena blok shema prikazana je na slici 12. Prema definiranoj funkciji unutar algoritma za svaku iteraciju simulacije VISSIM dobiva podatke o broju vozila na glavnom toku autoceste. Broj vozila na ulaznim rampama je stalan. Na prvoj rampi r_1 definirano je $800 \left[\frac{\text{voz}}{h}\right]$ a na drugoj r_2 definirano je $1050 \left[\frac{\text{voz}}{h}\right]$. Nakon svake iteracije simulacije, podaci se spremaju u određene varijable. Nakon završetka svih simulacija rezultati se spremaju u Microsoft Excel tablicu, kod koje svaki red predstavlja jednu iteraciju, a stupci predstavljaju akcije unutar iteracija. Svaka iteracija ima 24 akcije kao što je vidljivo u formuli (8):

$$A = \frac{T_s}{T}, \quad (8)$$

gdje su:

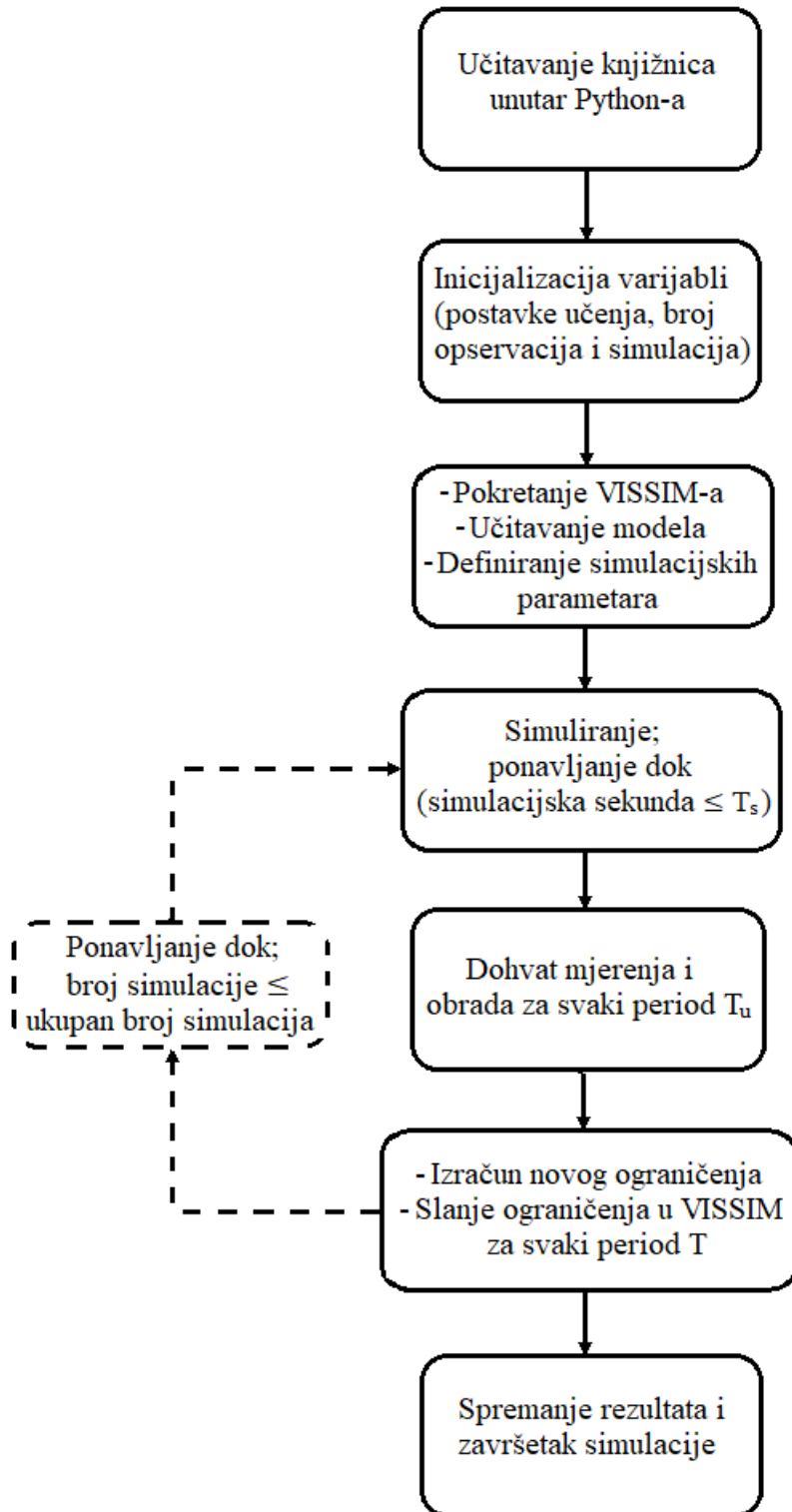
- T_s – vrijeme trajanja jedne iteracije,
- T – interval uzorkovanja i djelovanja agenta (mreže),
- A – broj akcija u iteraciji.

Prije u radu spomenuto je kako je interval akcije T jednak 300 [s], što znači u jednoj iteraciji simulacije, agent (neuronska mreža) napravi 24 akcije promjene brzine na VMS-ovima. Za svaku akciju algoritam pristupa podacima gustoće (9) i srednje brzine vozila na dionici od VMS-a, do osjetila d točno udaljenog jedan kilometar od VMS-a. U svakoj akciji algoritam radi 5 slika gdje se mjere svi navedeni parametri (svaki period T_u). Jedna minuta traje 600 simulacijskih jedinica što znači da se u jednoj sekundi izvrši 10 simulacijskih jedinica. Proces kreće ispočetka pokretanjem nove simulacije i nastavlja se do zadanog ukupnog broja simulacija. U ovom radu napravljeno je 2.100 simulacija od čega se prvih 400 simulacija odnosi na period opservacije, a 1.700 simulacija je bilo predodređeno za testiranje i rad algoritma DDQN. Gustoća se izračunava prema formuli:

$$g = \frac{N_{ulaz} - N_{izlaz}}{l_d}, \quad (9)$$

gdje su:

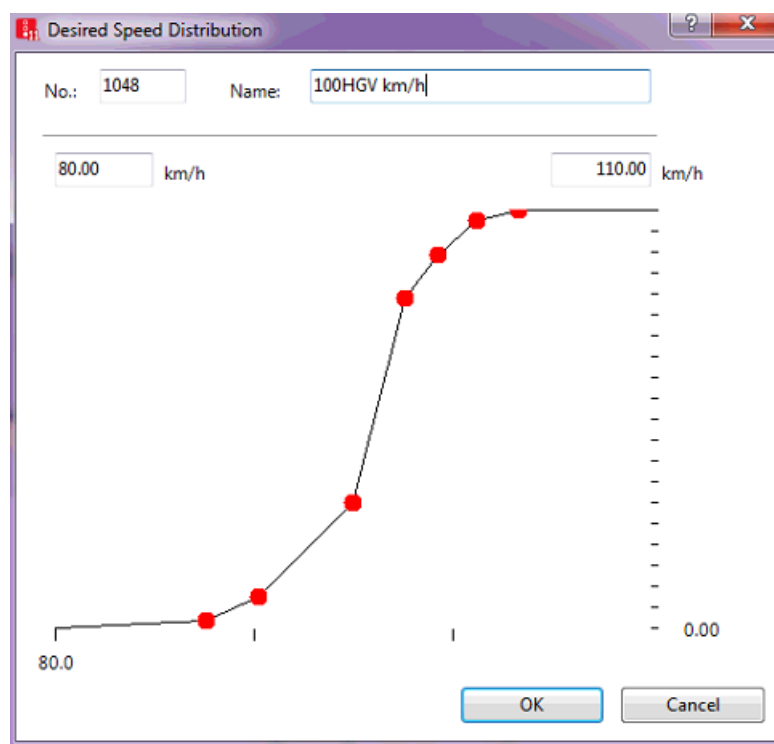
- g – gustoća na dionici poslije VMS-a,
- N_{ulaz} – broj vozila koja ulaze u dionicu,
- N_{izlaz} – broj vozila koja izlaze iz dionice,
- l_d – duljina dionice (u ovom slučaju 1.000 metara).



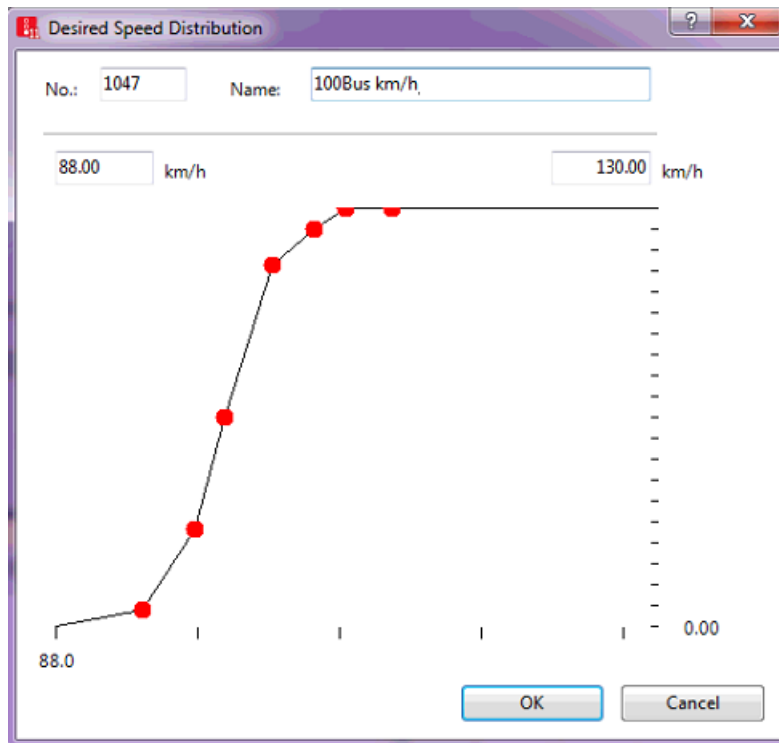
Slika 12. Pojednostavljena blok shema izvršavanja simulacije

Nakon svake akcije algoritam VSLC-a pristupa podacima brzine i gustoće te na osnovi njih postavlja ograničenje brzine. Agent odnosno neuronska mreža na osnovi iskustva mijenja i podešava težine unutar skrivenih konvolucijskih slojeva. U njoj je spremljeno svo znanje (iskustvo) koje stječe tijekom učenja. Kako agent djeluje izravno na simulacijsku okolinu (promjena prometnih parametara) postavljanjem ograničenja brzine, VSLC ima

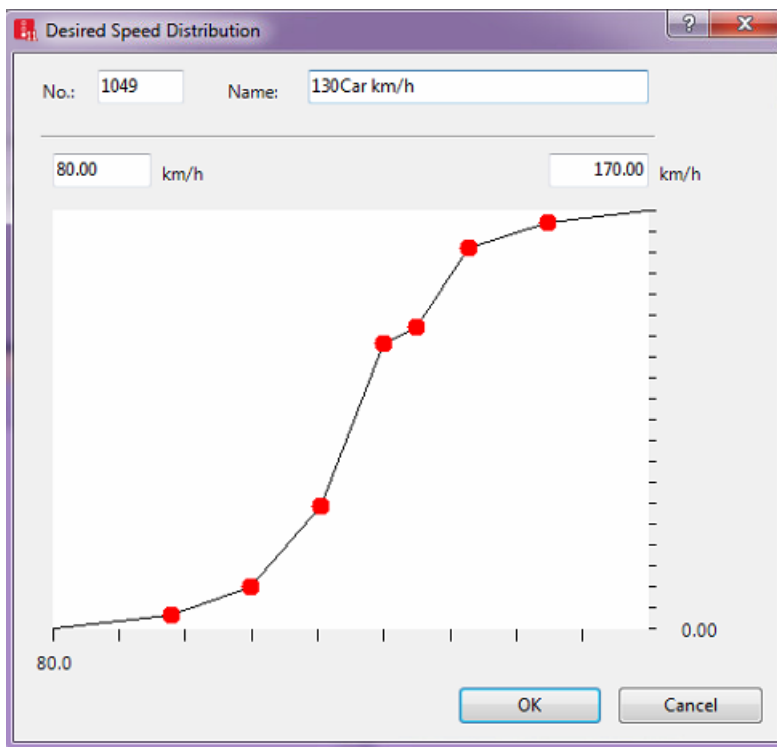
spособnost učenja što se manifestira boljim odabirom akcija. U mikroskopskom simulatoru VISSIM je također riješen problem poštivanja ograničenja brzina za teretna vozila i autobuse. Naime, odabirom većih brzina ograničenja od $100 \left[\frac{km}{h} \right]$ VSLC regulator predlaže svim vozilima da voze tom brzinom. Poštujući prometne propise nije ispravno da teška vozila (kamioni i autobusi) voze nedozvoljenim brzinama ovisno o njihovoj klasifikaciji. Oni u tom slučaju voze najvećom dozvoljenom brzinom, koja se ravna po zadanoj distribuciji unutar VISSIM-a prikazanoj na slici 13 za kamione, i slici 14 za autobuse. Distribucija brzine teretnih vozila dana je tako da najveći broj populacije vozi brzinom između $90 \left[\frac{km}{h} \right]$ i $110 \left[\frac{km}{h} \right]$, a najveća populacija autobusa vozi brzinama između $90 \left[\frac{km}{h} \right]$ i $130 \left[\frac{km}{h} \right]$. Distribucija brzine osobnih vozila je nešto drugačija, te kod nje najveća populacija osobnih vozila vozi brzinom između $90 \left[\frac{km}{h} \right]$ i $170 \left[\frac{km}{h} \right]$ kako je i prikazano na slici 15.



Slika 13. Definirana distribucija brzina za teretna vozila unutar VISSIM-a



Slika 14. Definirana distribucija brzina za autobuse unutar VISSIM-a



Slika 15. Definirana distribucija brzina za osobna vozila unutar VISSIM-a

Nakon što se izvrše sve simulacije, rezultati se spremaju u Microsoft Excel tablicu te se nad njima vrši analiza uspješnosti implementiranog algoritma. Analiza dobivenih simulacijskih rezultata slijedi u idućem poglavlju rada.

6. Simulacijski rezultati

U ovom poglavlju dana je analiza i prikaz dobivenih rezultata simulacije prometnog modela gradske autoceste. Rezultati opisuju karakteristične parametre prometnog toka na osnovi kojih je ocijenjen rad VSLC-a zasnovan na učenju dualne duboke Q-mreže. Osim parametara prometnog toka prikazana su izlazna ograničenja brzine koje izračunava VSLC implementiran na dionici autoceste l te su korištenjem zavisnosti prometnog toka o ograničenjima brzine izneseni određeni zaključci o radu implementiranog sustava.

6.1. Model i podaci

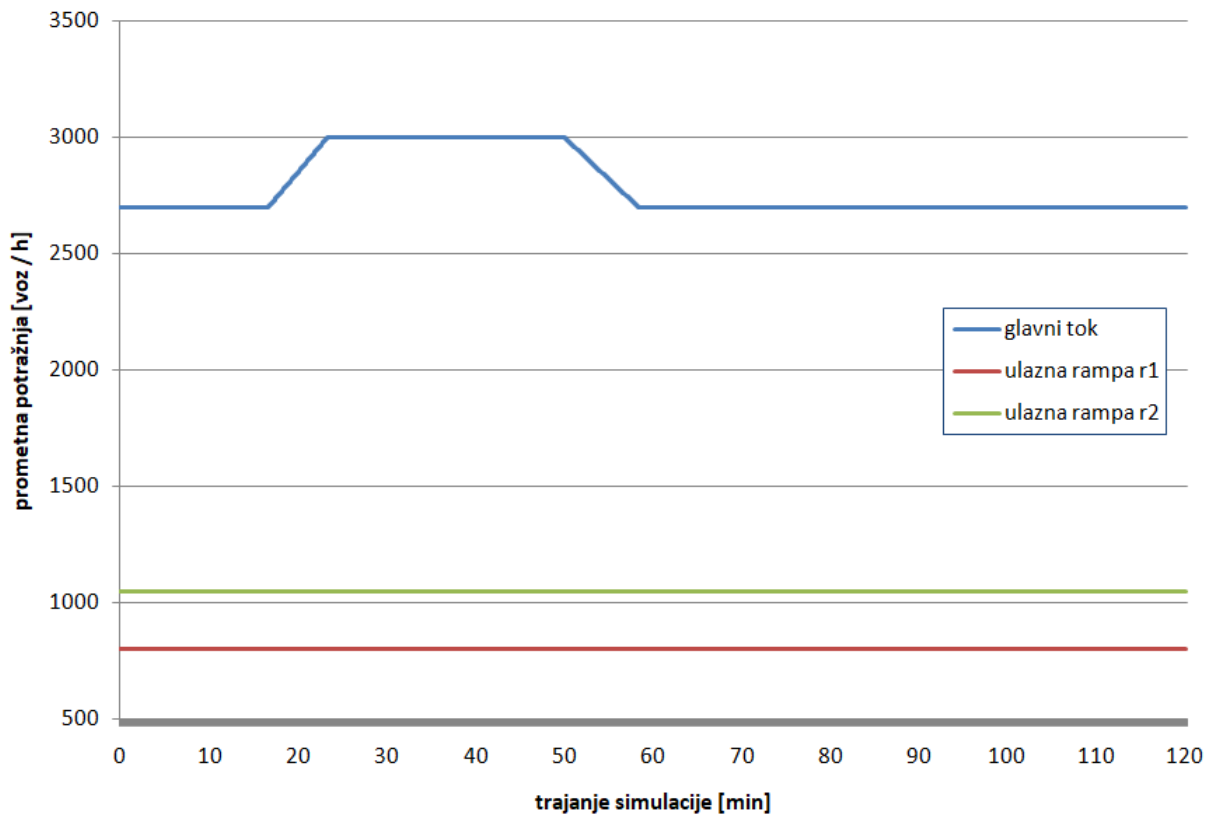
Upravljačka logika VSLC-a zasnovana na učenju dualne duboke Q-mreže implementirana je u skripti napisanoj u programskom okruženju Python/Keras, dok je model gradske autoceste s dvije prometne trake prikazan na slici 6 napravljen u mikroskopskom simulatoru VISSIM. U modelu definirana je dionica l na kojoj mjerimo određene prometne parametre. Traka za ubrzanje prve ulazne rampe r_1 počinje na 2.377 metru autoceste, traka za ubrzanje druge ulazne rampe r_2 počinje na 5.395 metru autoceste, a traka za usporenje izlazne rampe s_1 počinje na 3.380 metru autoceste.

Promjenjivi znakovi za ograničenje brzine (VMS-ovi) koji prikazuju ograničenja brzine nalaze se na početku dionice l te se pomoću njih obavještavaju vozači kolika je dozvoljena brzina kretanja. Raspodjela vozila prometnog toka u slučaju simulacijskog testiranja u ovom radu čine automobili (96 %), teretna vozila (2 %) i autobusi (2 %) kako je prikazano u tablici 1.

Tablica 1. Zastupljenosti različitih tipova vozila

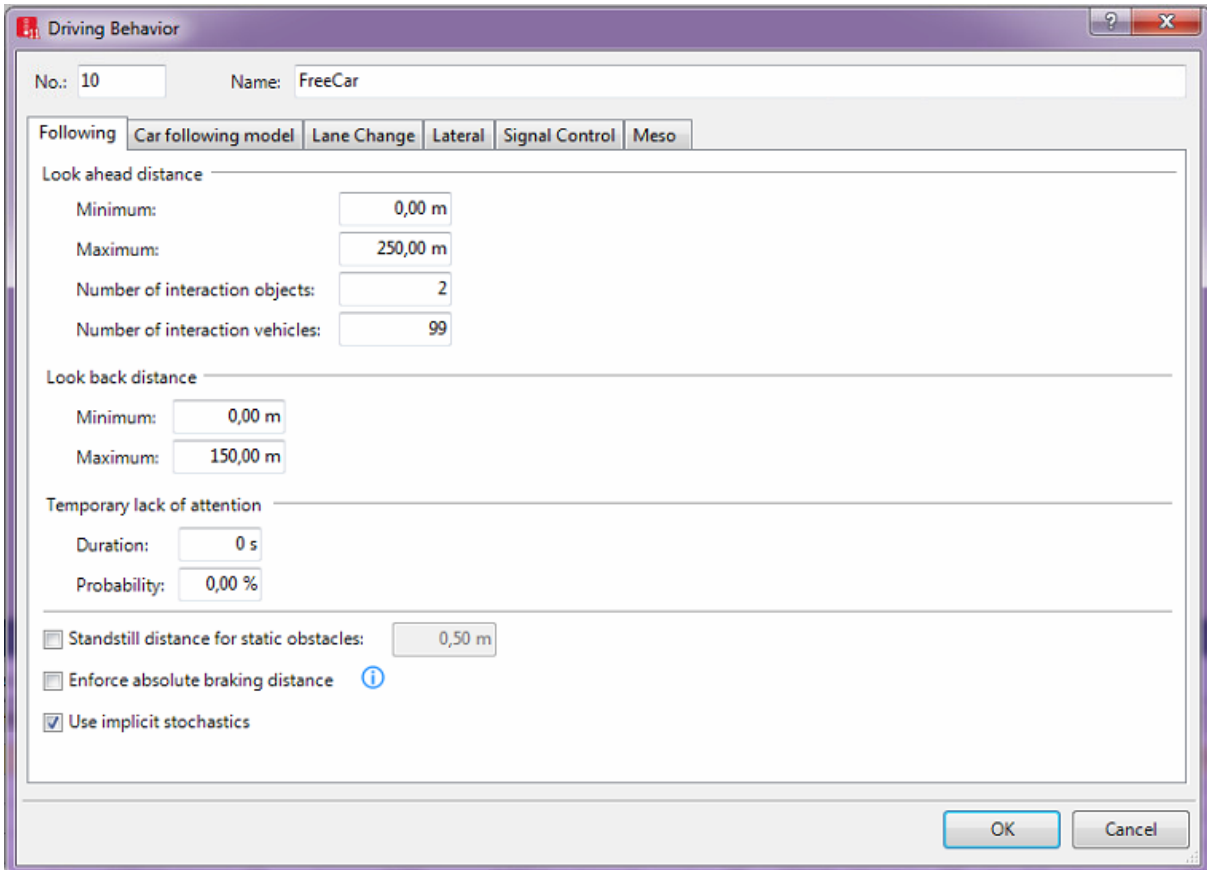
Tip vozila	Distribucija brzine	Raspodjela vozila (%)
Automobil	130 Car km/h	96
Teretno vozilo	100 HGV km/h	2
Autobus	130 Bus km/h	2

Prva ulazna rampa r_1 ima stalan ulazni prometni tok i iznosi $800 \left[\frac{voz}{h} \right]$ a na drugoj ulaznoj rampi r_2 definiran je stalan prometni tok od $1050 \left[\frac{voz}{h} \right]$. Glavni tok ima promjenjivu prometnu potražnju prema definiranim jednadžbama unutar kôda. Maksimalna vrijednost opterećenja postiže se nakon 1.400 sekundi simulacije i traje 1.600 sekundi, odnosno je 26 minuta i 40 sekundi. Vrijednost prometne potražnje na glavnom prometnom toku za taj period iznosi $3.000 \left[\frac{voz}{h} \right]$. Na grafikonu 1 prikazana je prometna potražnja na glavnom prometnom toku i ulaznim rampama (r_1 i r_2).

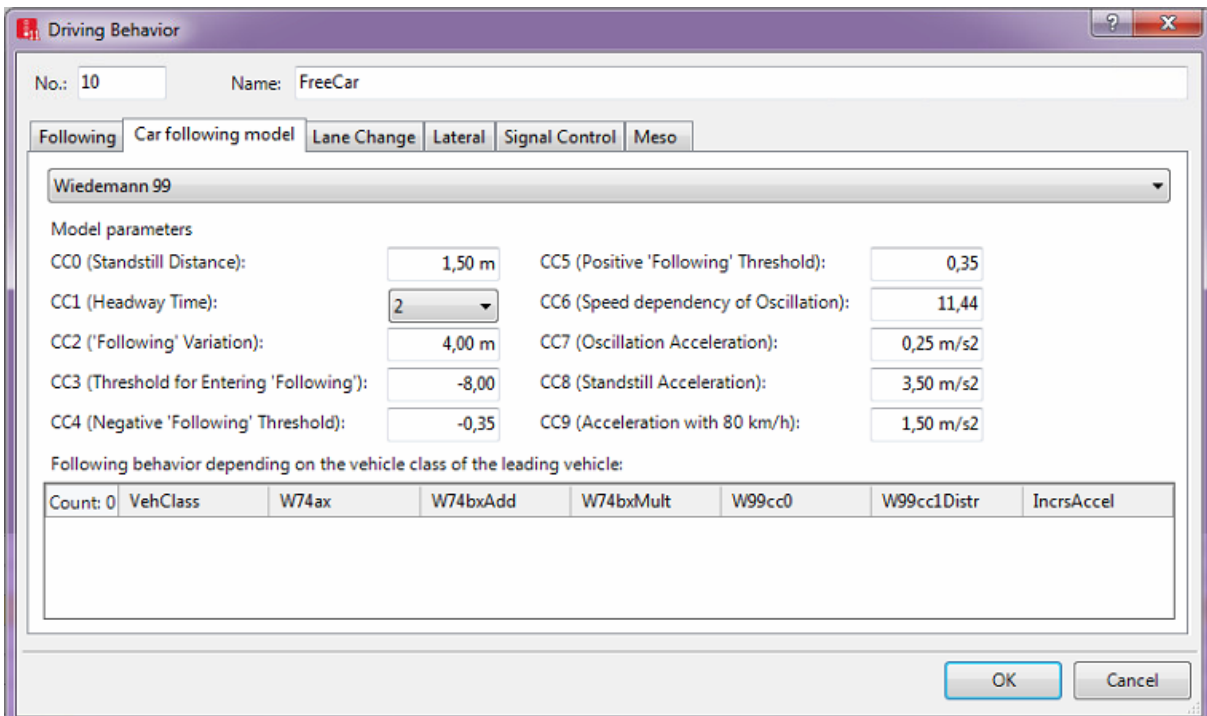


Grafikon 1. Prometna potražnja na glavnom toku i ulaznim rampama

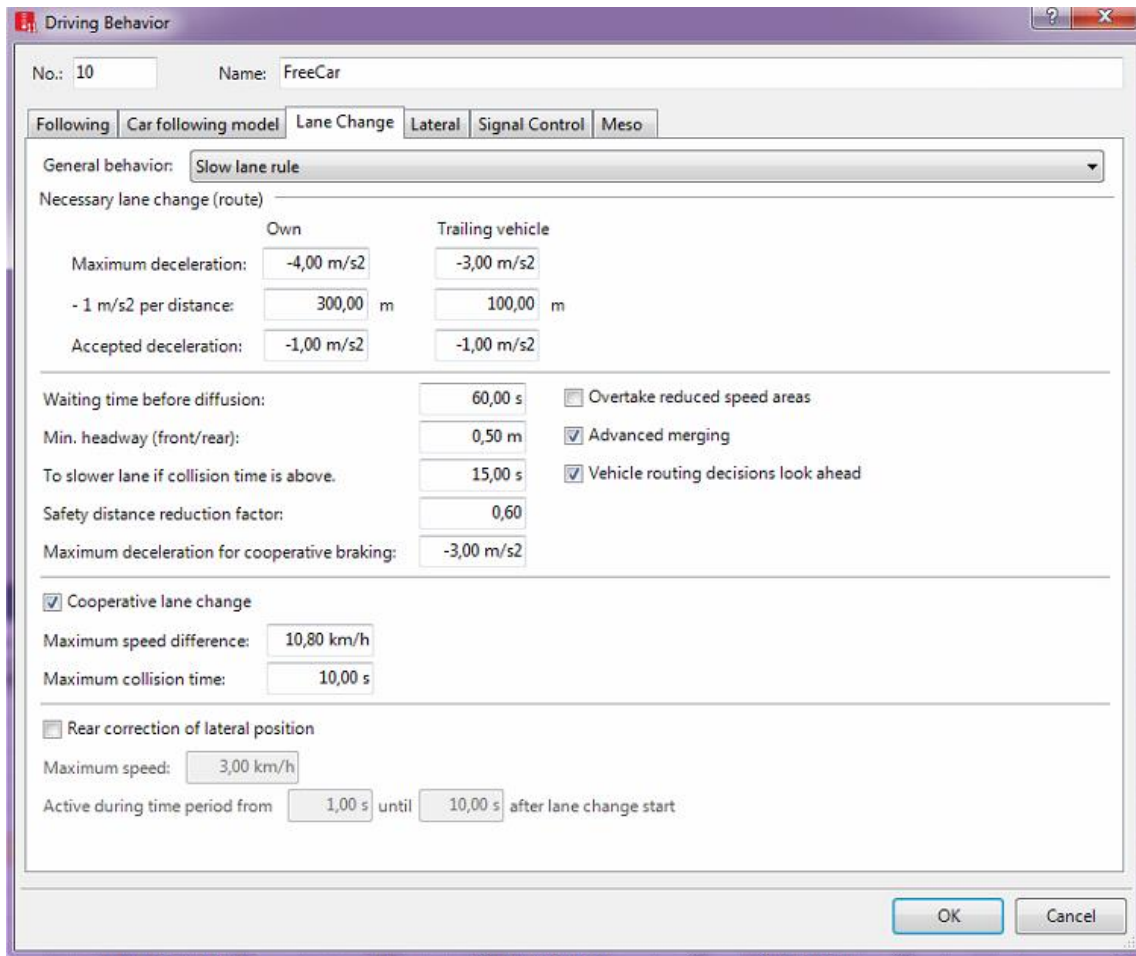
Ovim povećanjem prometne potražnje potencirano je stvaranje povećanog prometnog opterećenja u području oko ulazne rampe r_2 . Zagušenje uzrokuje postepeno širenje šok vala glavnim tokom unatrag prelijevajući se u dionicu l te uzrokuje nemogućnost normalnog odvijanja prometnog toka. Spomenuti period predstavlja najtežu prepreku koju agent na osnovi stečenog iskustva mora riješiti upravljanjem brzinama na postavljenim VMS-ovima. Predstavljeni prometni model nije potrebno umjeravati (u smislu provjere iznosa tokova i sl.) zato što model ne predstavlja simulaciju stvarne okoline za koju se traži/testira varijantno rješenje, već služi u svrhu testiranja okruženja. Parametri za potrebe ovog rada su dobro umjereni u mikroskopskom simulatoru VISSIM koji definiraju karakteristike i ponašanja vozača na autocesti. Dinamika glavnog toka ovisi o nekoliko bitnih parametara kao što su: interval slijeđenja vozila (engl. *Time Headway*), najvećim vrijednostima ubrzanja (akceleracije i deceleracije) pojedinih tipova vozila, parametri kod promjene prometnih traka (udaljenost bočnih vozila, uočavanje vozila koja prilaze od iza itd.). Navedene postavke, ali i mnoge druge prikazane su na slikama 16, 17, 18 i 19.



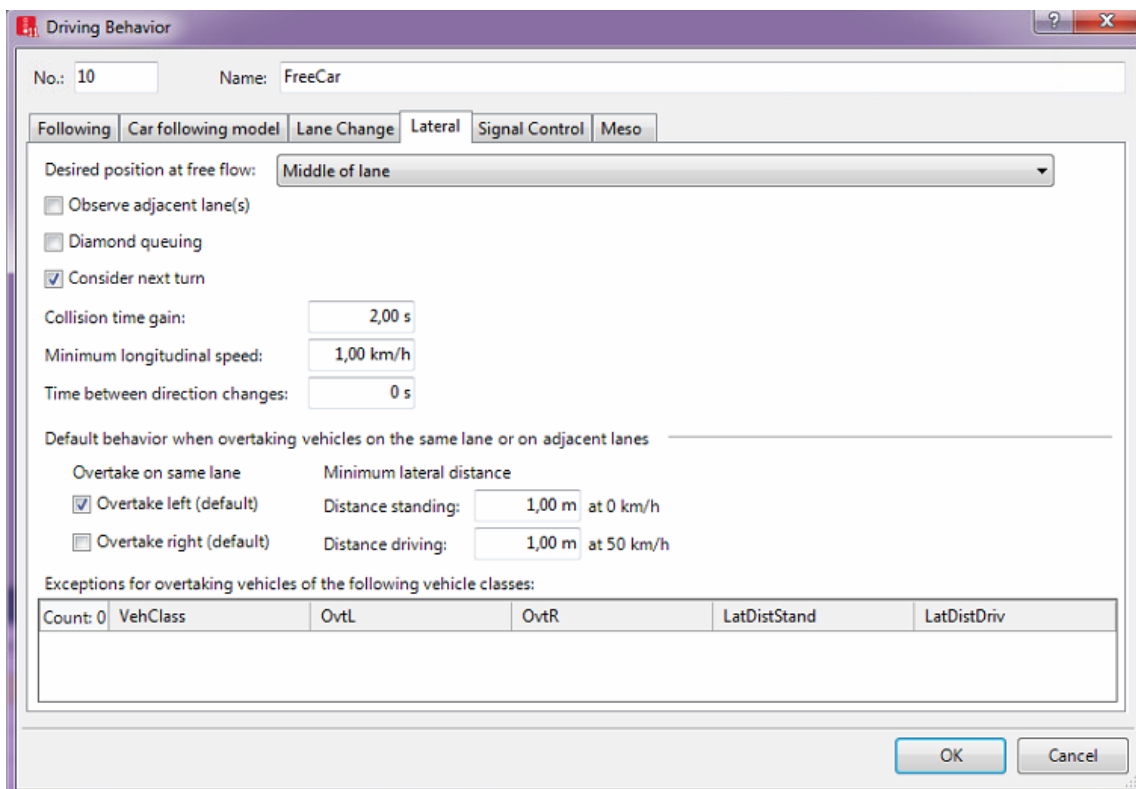
Slika 16. Parametri slijeđenja vozila u VISSIM-u



Slika 17. Dodatni parametri slijeđenja vozila u VISSIM-u

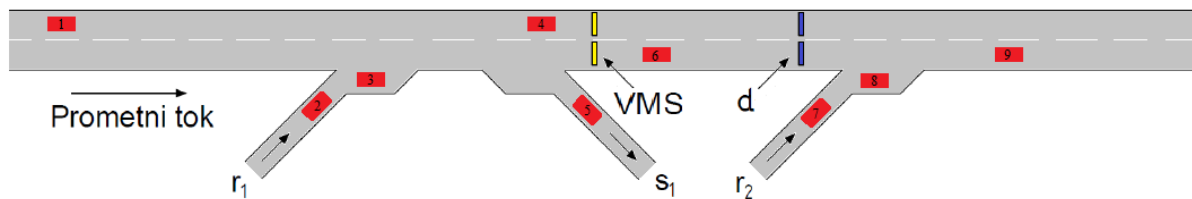


Slika 18. Parametri promjene prometnih traka



Slika 19. Parametri bočnog kretanja vozila

Podaci o vozilima u mreži spremaju se u matricu svakih T_u (60 [s]) i bilježi se njihova pozicija u mreži, brzina, link na kojem se vozilo nalazi te u kojoj se traci vozilo kreće na zabilježenom linku. Prikupljeni podaci spremaju se u matricu gdje stupci predstavljaju spomenute vrijednosti, a redovi predstavljaju podatke svakog pojedinog vozila. Primjer uzorkovanja prikazan je na slici 20 gdje brojevi na vozilima predstavljaju red u tablici o kojem se vozilu radi, a prikaz podataka kao matrice prikazan je u tablici 2. Prvi stupac u tablici 2 je dodan radi preglednosti, no u samoj skripti ne postoji. Važno je napomenuti kako su vrijednosti pozicije vozila u mreži i brzina vozila pretvorene u cjelobrojne vrijednosti radi jednostavnosti implementacije algoritma i obrade podataka.



Slika 20. Prikaz uzorkovanja podataka o vozilima

Tablica 2. Primjer matrice podataka prikupljenih za svako vozilo u mreži

Indeks vozila (red u matrici)	Pozicija u mreži	Brzina	Poveznica	Prometna traka u poveznici
1	221	88	4	1
2	1.321	62	10	1
3	1.375	60	10	1
...
9	6.950	102	4	2

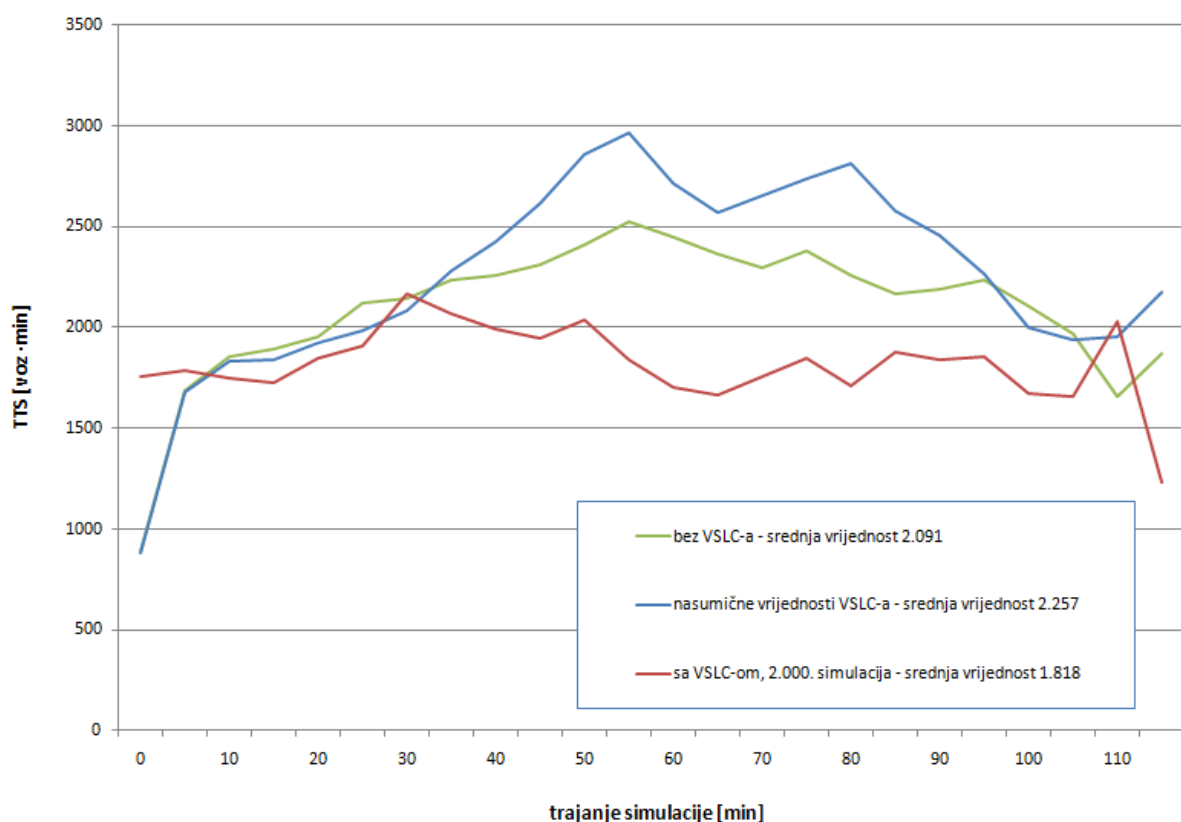
6.2. Dobiveni rezultati

U ovom potpoglavlju dan je prikaz rezultata i analiza dobivenih rezultata prikupljenih tijekom izvršenih simulacija. U svrhu rada napravljeno je ukupno 2.100 simulacija te je za prikaz rezultata odabran slučaj s početka opservacije algoritma (21. simulacija) te slučaj iz 2.000-te simulacije. Na osnovi dobivenih rezultata iz spomenutih simulacija napravljena je procjena VSLC algoritma zasnovanog na algoritmu DDQN s ciljem ocjene kvalitete upravljanja ograničenjem brzine na promatranom dionici testnog modela gradske autoceste. Određeni zaključci o sposobnosti učenja upravljačkog algoritma doneseni su na osnovi prikupljenih rezultata tijekom simulacija. Dobiveni rezultati ovise o odlukama agenta o prilagodbi ograničenja brzine koje ovise o prometnim uvjetima na simuliranom modelu.

6.2.1. TTS kao glavna mjera uspješnosti učenja

Parametar TTS koji se odnosi na ukupno provedeno vrijeme svih vozila u prometnoj mreži predstavlja kvantitativnu mjeru koja dobro opisuje stanje prometnog toka. TTS u ovom radu predstavlja glavni pokazatelj uspješnosti učenja algoritma za upravljanje VSLC-om

zasnovanog na učenju dualne duboke Q-mreže. U bilo kom slučaju TTS proporcionalno raste s povećanjem gustoće prometnog toka ako nastane zagušenje unutar simulacijskog modela. Parametar TTS se opisuje mjernom jedinicom umnoška vozila i vremena provedenog u prometnoj mreži [$voz \cdot min$]. Isto tako u slučaju kada algoritam reagira smanjenjem izlaznih ograničenja brzina (npr. indeks brzine 2, $v_{VSLC} = 70 [\frac{km}{h}]$) uzrokovat će povećanjem vremena putovanja vozila za isti prevaljeni put, samim time raste i vrijednost TTS-a. Korištenjem ovog parametra kao pokazatelja kvalitete algoritma se sprječava da algoritam u nastojanju da smanji nastalo zagušenje previše smanji ograničenja brzine koja se šalju na VMS-ove. Na ovaj način algoritam mora računati optimalne vrijednosti ograničenja brzina tako da traži ravnotežu između prethodno spomenutih pojava s ciljem minimiziranja vrijednosti TTS-a koja je predstavljena kao kazna unutar algoritma nakon svake poduzete akcije. Treba napomenuti da je TTS mjeren unutar svakog intervala uzorkovanja T_u (60 [s]), a za svaki upravljački interval T (300 [s]) izračunat je prosjek jednominutnih uzoraka TTS-a. Na grafikonu 2 prikazane su vrijednosti za slučaj kada ne upravljamo VSLC-om (zeleno krivulja) odnosno ograničenje je nepromijenjenih $130 [\frac{km}{h}]$, slučaj kada je algoritam u stanju opservacije (nasumične promjene ograničenja) što je u 21. simulaciji (plava krivulja) te kada je u fazi primjenjivanja ograničenja brzina ma VMS-ovima (crvena krivulja) odnosno za vrijeme 2.000. simulacije. Može se vidjeti da je vrijednost prosječnog TTS-a nakon 2.000 simulacija više uniformna i ima manje oscilacija te kako polako opada s povećanjem broja simulacija.

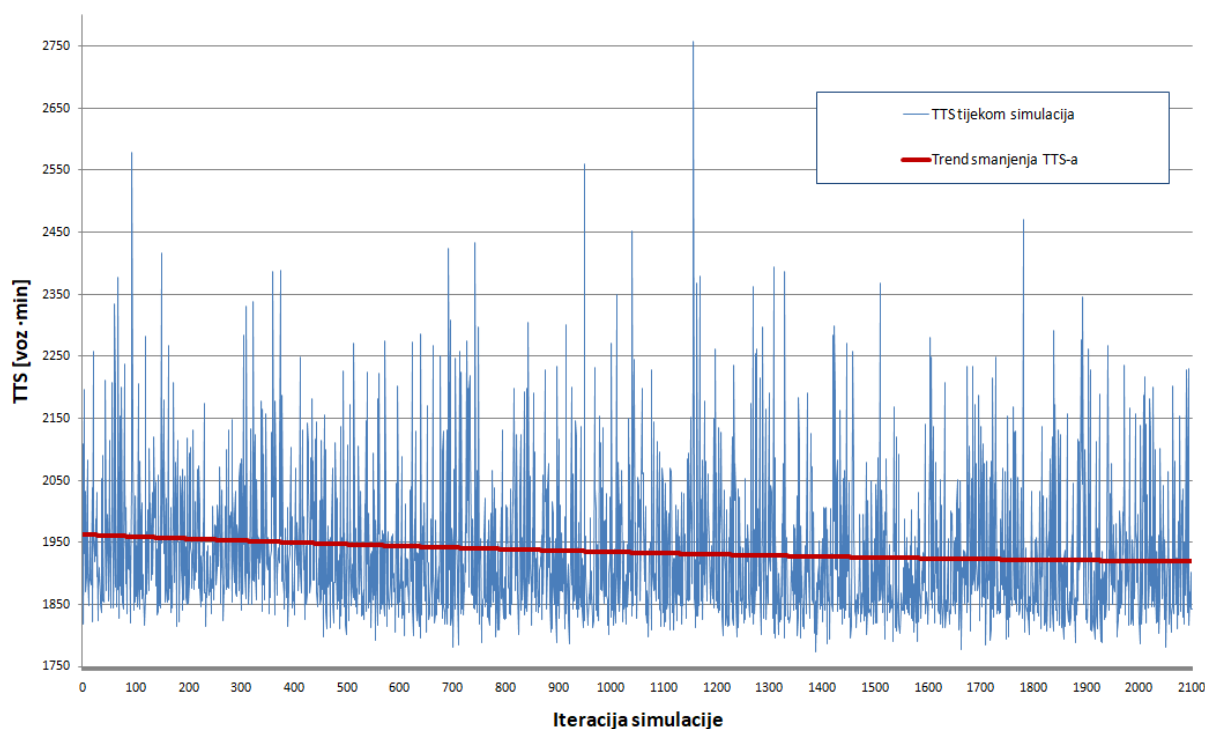


Grafikon 2. Prosječna vrijednost TTS-a za svaki interval T

Tablica 3. Prosječne vrijednosti TTS-a u prometnom modelu za intervale mjerenja T

	Bez VSLC-a – stalno ograničenje $130 \left[\frac{km}{h} \right]$	Nasumične vrijednosti VSLC-a – vrijeme opservacije (21. simulacija)	Implementiran DDQN algoritam VSLC (2.000. simulacija)
Maksimalan TTS [voz · min]	2.056	2.967	2.164
Srednja vrijednost TTS [voz · min]	2.091,3	2.257,5	1.818

Na osnovi vrijednosti krivulja na grafikonu 2 napravljena je tablica 3. Na osnovi podataka srednjih vrijednosti vidljivo je da primjenom naučenog VSLC algoritma postignuto smanjenje TTS-a računano unutar svakog intervala T tijekom simulacije. Smanjenje srednje vrijednosti TTS-a naučenog algoritma u odnosu na stalno ograničenje brzine iznosi otprilike 13,06 %, a u odnosu na nasumične vrijednosti VSLC-a iznosi 19,4 %. Ovaj rezultat predstavlja čvrstu činjenicu da algoritam VSLC-a zasnovan na učenju dualne duboke Q-mreže ima sposobnost i velik potencijal učenja.

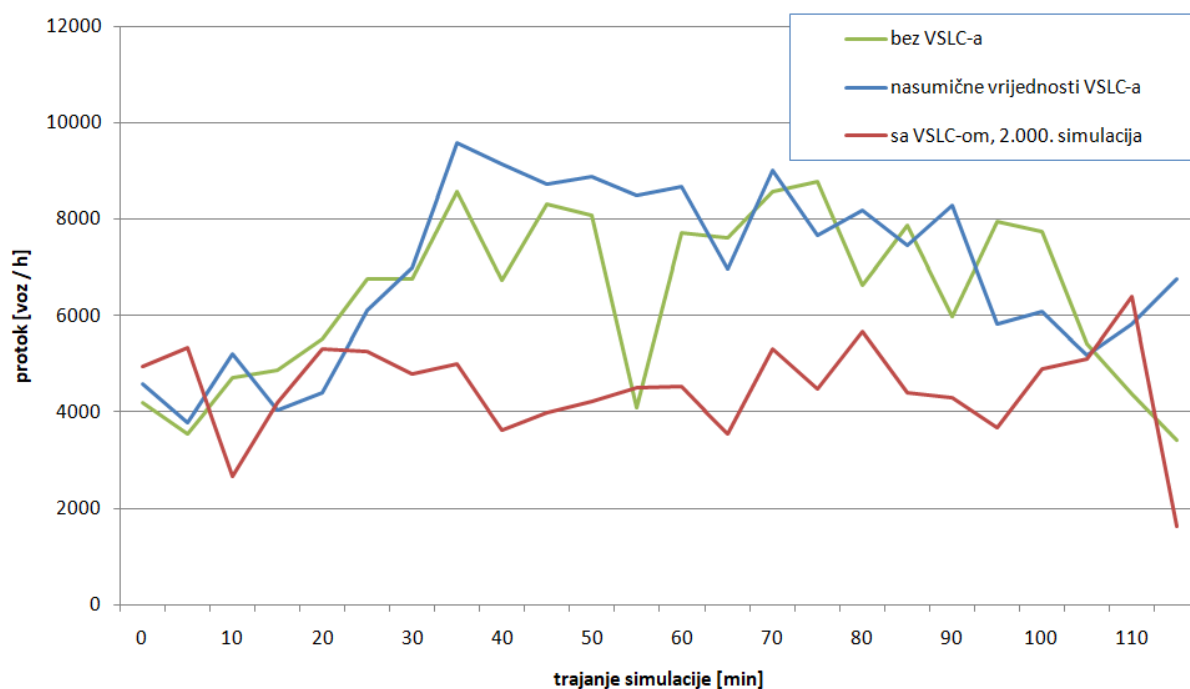


Grafikon 3. Trend smanjenja prosječnog TTS-a tijekom provođenja simulacija

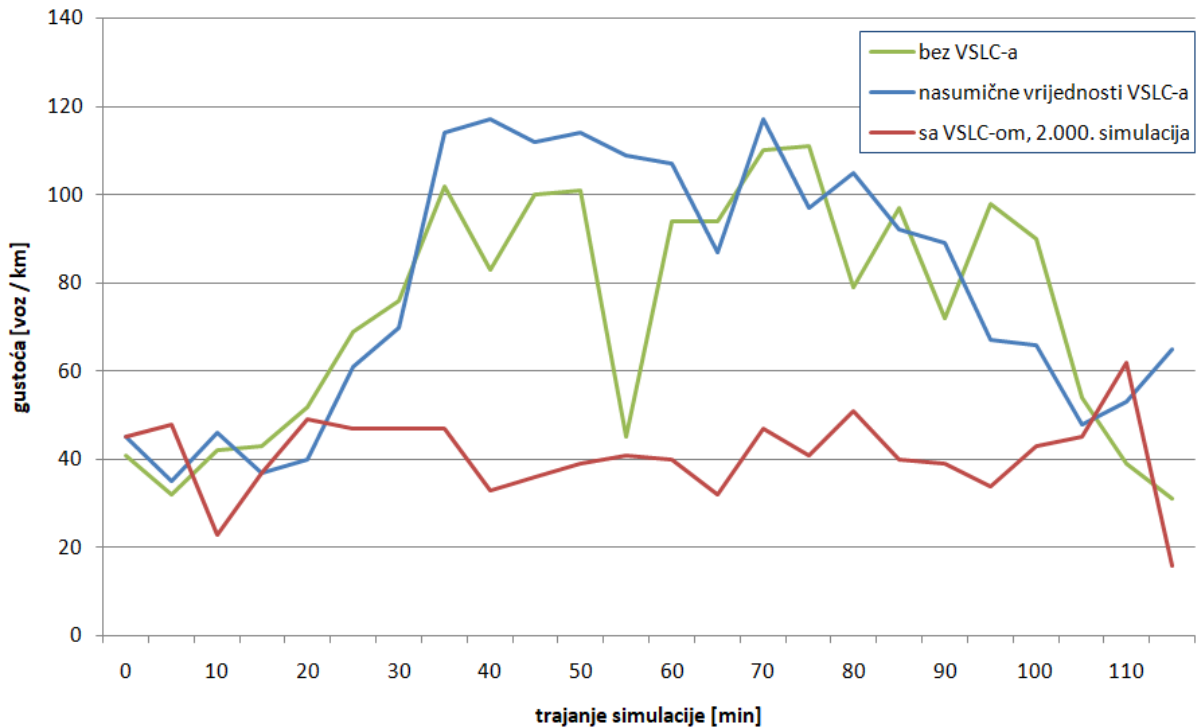
Na grafikonu 3 vidljiv je trend promjene prosječnog TTS-a za svaku iteraciju simulacije tijekom simuliranja. Što je veći broj simulacije, prosječni TTS opada, čime je dokazano kako primijenjeni algoritam ima sposobnost učenja i djeluje u cilju smanjenja i sprječavanja zagušenja.

6.2.1. Protok, gustoća i brzina na promatranjoj dionici l

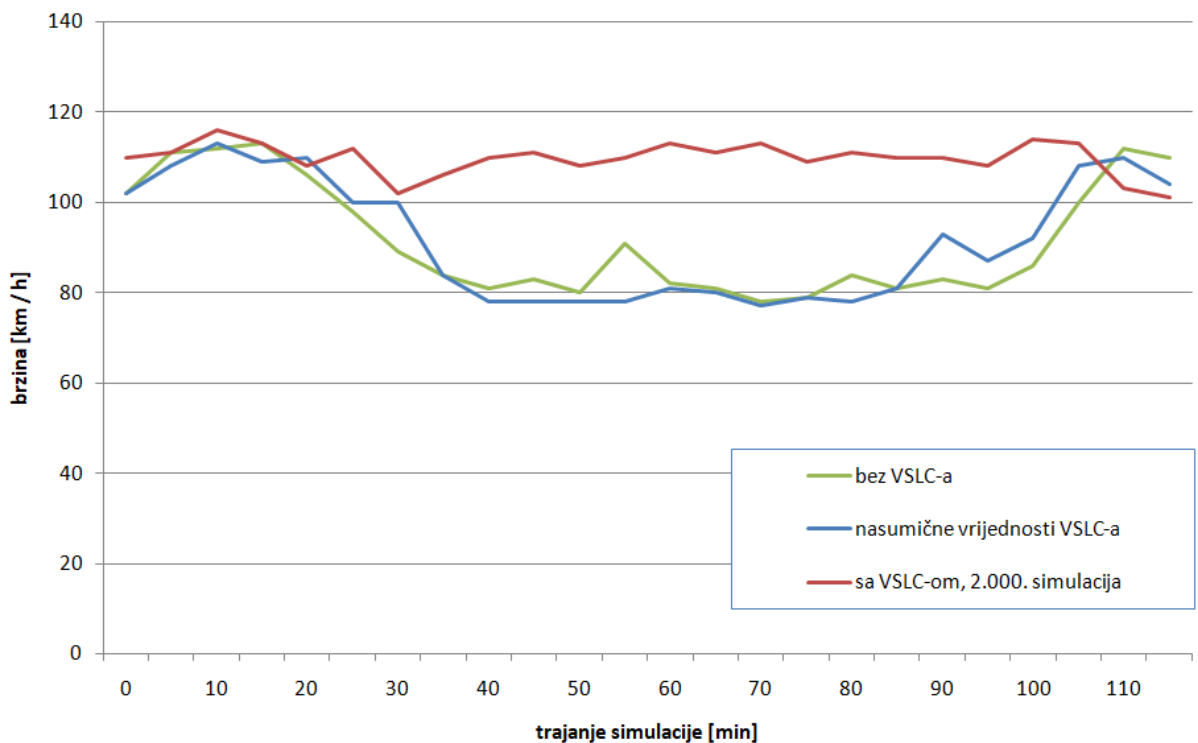
U ovom potpoglavlju su prikazani rezultati za izmjerene i izračunate prometne veličine (protok, gustoća i brzina). Gustoća prometnog toka mjerena je na dionici l . Vrijednosti su izračunate nakon svakog intervala akcije Δt (5 [min]) tijekom trajanja simulacije. Na grafikonu 4, 5 i 6 zelena krivulja predstavlja slučaj bez VSLC-a, plava krivulja slučaj nasumičnih ograničenja VSLC-a a crvena naučen algoritam VSLC-a nakon 2.000 simulacija što karakterizira algoritam VSLC s naučenim znanjem o svojoj okolini te kako pojedine akcije (nizovi akcija) djeluju na promjene okoline. Grafikon 4 prikazuje protok vozila, grafikon 5 prikazuje gustoću vozila dok je na grafikonu 6 prikazana srednja brzina vozila na dionici l .



Grafikon 4. Protok vozila na dionici l tijekom simulacije



Grafikon 5. Gustoća vozila na dionici *l* tijekom simulacije



Grafikon 6. Srednja brzina vozila na dionici *l* tijekom simulacije

Na grafikonu 4 jasno je vidljivo kako vrijednosti krivulja koje opisuju protok na dionici autoceste *l* poprimaju umanjene vrijednosti nakon naučene mreže na upravljanje VSLC-om. Protok je izračunat na temelju mjerenja gustoće i srednje brzine. Protok se računa prema formuli [18]:

$$q = g \cdot v, \tag{10}$$

gdje su:

- q – protok,
- g – gustoća,
- v – srednja brzina vozila.

Iz krivulja koje opisuju gustoću toka na dionici autoceste l vidljive su promjene koje neposredno slijede zbog nedjelovanja regulatora VSLC-a (promjene ograničenja brzine). Uzrok tome je poremećaj prometnog toka u okruženju rampe čija se karakteristika širi unatrag niz glavni tok (šok val). Prema krivulji srednje brzine vozila (crvena krivulja na grafikonu 6) na dionici autoceste l vidljivo je kako je brzina umjerena i nema velikih oscilacija u promjenama brzine, čime je zaključeno kako je algoritam dobro naučio funkciju optimalnog upravljačkog zakona. Cilj ovakvog pristupa je preventivno djelovati na dionici l ispred u odnosu na zagušeni dio autoceste kako bi se popravila prometna situacija na zagušenom dijelu dionice (oko rampe r_2). Algoritam VSLC-a zasnovan na algoritmu DDQN mora naučiti prepoznati određena stanja gustoće toka na promatranoj dionici na osnovi kojih zaključuje je li nastalo zagušenje, te na osnovi toga preventivno smanjuje prosječne brzine prometnog toka (harmonizacija brzine prometnog toka) na dionici l . Smanjenjem brzine nadolazećeg toka vozila na zagušeni dio dionice smanjuje brzinu propagacije šok valova izazvanih zagušenjem u blizini rampe r_2 uzvodno uz autocestu. U tablici 4 su prikazane srednje vrijednosti rezultata na dionici autoceste l .

Tablica 4. Srednje vrijednosti prometnih parametara na dionici autoceste l

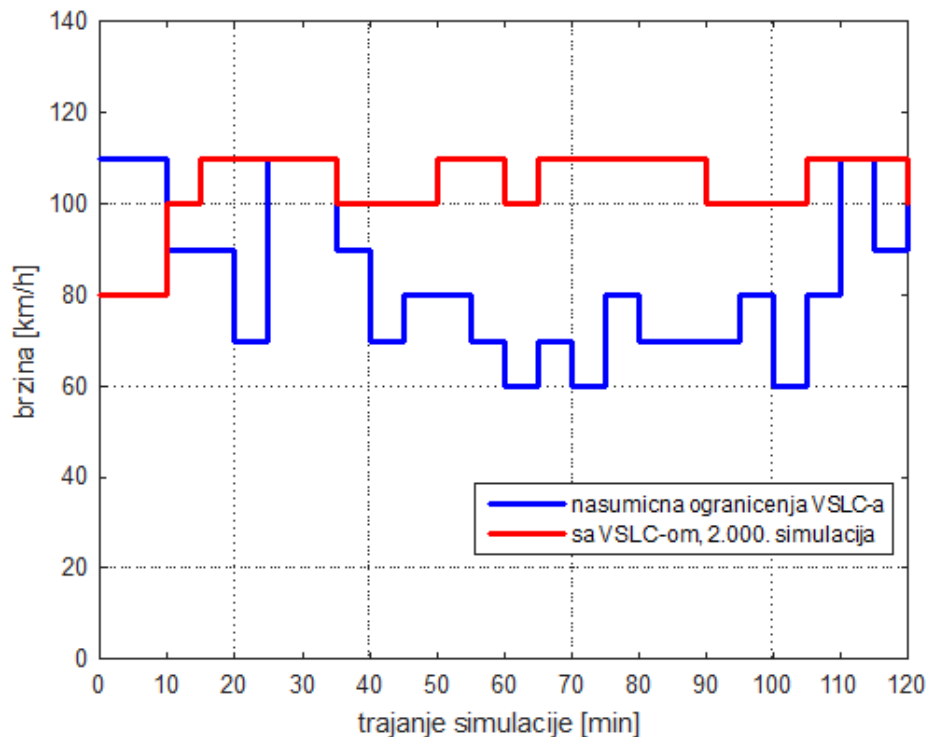
Dionica autoceste l	Bez VSLC – stalno ograničenje 130 [$\frac{km}{h}$]	Nasumične vrijednosti VSLC – vrijeme opservacije (21. simulacija)	Implementiran DDQN algoritam VSLC (2.000. simulacija)
Protok [voz / h]	6.421	6.909	4.486
Gustoća [voz / km]	73	79	41
Brzina [km / h]	91.9	92	109,7

Grafikon 5 prikazuje gustoće vozila na dionici autoceste l te je primjećeno smanjenje prosječne gustoće naučenog algoritma u odnosu na slučaj bez VSLC-a za 44,05 %, a u odnosu na nasumične vrijednosti VSLC-a za 48,12 %. Uzročno smanjenju gustoće, povećala se srednja brzina kretanja vozila na dionici autoceste l naučenog algoritma u odnosu stalno ograničenje brzine za 16,22 %, a u odnosu na nasumične vrijednosti VSLC-a za 16,13 %, što je i vidljivo na grafikonu 6. Također valja napomenuti kako prema grafikonu 6 vidimo da je na osnovi naučenog algoritma VSLC brzina ujednačena i nema velikih oscilacija promjena brzine.

6.2.2. VSLC - ograničenja brzine za dionicu autoceste l

Stepenasti oblik krivulja na grafikonu 7 ograničenja brzina proizlazi iz činjenice da VSLC daje izlazne cjelobrojne vrijednosti ograničenja brzine koja se postavljaju na VMS-

ove. Raspon ograničenja brzine koje generira VSLC kreće se od minimalnih 60 [km/h] do maksimalnih 130 [km/h]. Zbog specifičnosti i dinamičnosti prometnog toka čije se karakteristike nerijetko mijenjaju u prostoru i vremenu bitna je pravovremena vrijednost ograničenja brzine ovisno o prometnoj potražnji na promatranoj dionici. VSLC sustav tijekom simulacije kontinuirano izračunava i šalje ograničenja brzine na VMS-ove na dionici l . Svako ograničenje brzine traje koliko i vremenski interval Δ (5 [min]) te se nakon svake akcije izračuna i pošalje novo ograničenje na VMS-ove. U slučaju ovakvog režima rada sustava, može se desiti da sustav ponekad kasni s odgovarajućim promjenama ograničenja brzine jer ponekad ne detektira kratkotrajne manje poremećaje u prometnom toku što može utjecati na krajnji rezultat. Interval Δ bi u teoriji mogao biti i kraći npr. 2,5 [min]. VSLC bi djelovao pouzdanije i točnije ako bi se koristio interval uzorkovanja i primjene algoritma u trajanju od dvije minute i 30 sekundi ovisno o trenutnoj prometnoj situaciji. Problem koji se javlja je implementacija sustava s tako malim vremenima uzorkovanja i promjene ograničenja jer se pretpostavlja da ne bi imao velik utjecaj na ponašanje i razumijevanje vozača. Na grafikonu 7 prikazane su vrijednosti ograničenja brzine za slučaj s VSLC-om za vrijeme opservacije (plava krivulja) i za slučaj kada je algoritam stekao određeno znanje nakon 2.000 iteracija simulacija (crvena krivulja).



Grafikon 7. Iznos ograničenja brzine na dionici autoceste l

Iz grafikona se može zaključiti da za vrijeme opservacije algoritam za upravljanje VSLC-om samo proučava svoju okolinu velikim brojem slučajno odabranih akcija (ograničenja brzine). Crvena krivulja predstavlja slučaj za algoritam upravljanja VSLC-om koji je s vremenom naučio kako djelovati na nastala stanja okoline odabirom pravih i dobrih akcija. Crvena krivulja ima manje oscilacije u promjenama brzina među koracima u simulaciji što uzrokuje stabilnost i harmonizaciju prometnog toka. Ova karakteristika je definirana

unutar algoritma kao funkcija nagrade kojom se kažnjavaju velike razlike ograničenja brzine. Kod slučaja naučenog algoritma VSLC (crvena krivulja na grafikonu 7) najveće promjene ograničenja brzine ne prelaze 20 [km/h], što nije slučaj za vrijeme nasumičnih ograničenja VSLC-a kao što je vidljivo kod plave krivulje na grafikonu 7. Na osnovi izvršenih simulacija (2.100) VSLC algoritam još uvijek ponekad napravi preveliku promjenu brzine na VMS-ovima. Potreban je još veći broj iteracija algoritma u svrhu boljeg podešavanja težina unutar konvolucijske neuronske mreže. Generalno usporedivši crvenu i plavu krivulju vidi se značajan napredak kvalitete rada što se pripisuje učenju algoritma VSLC-a primjenom učenja nad dualnom dubokom Q-mrežom, te način upravljanja ograničenjem brzine sve većim brojem iteracija postaje sve bolje.

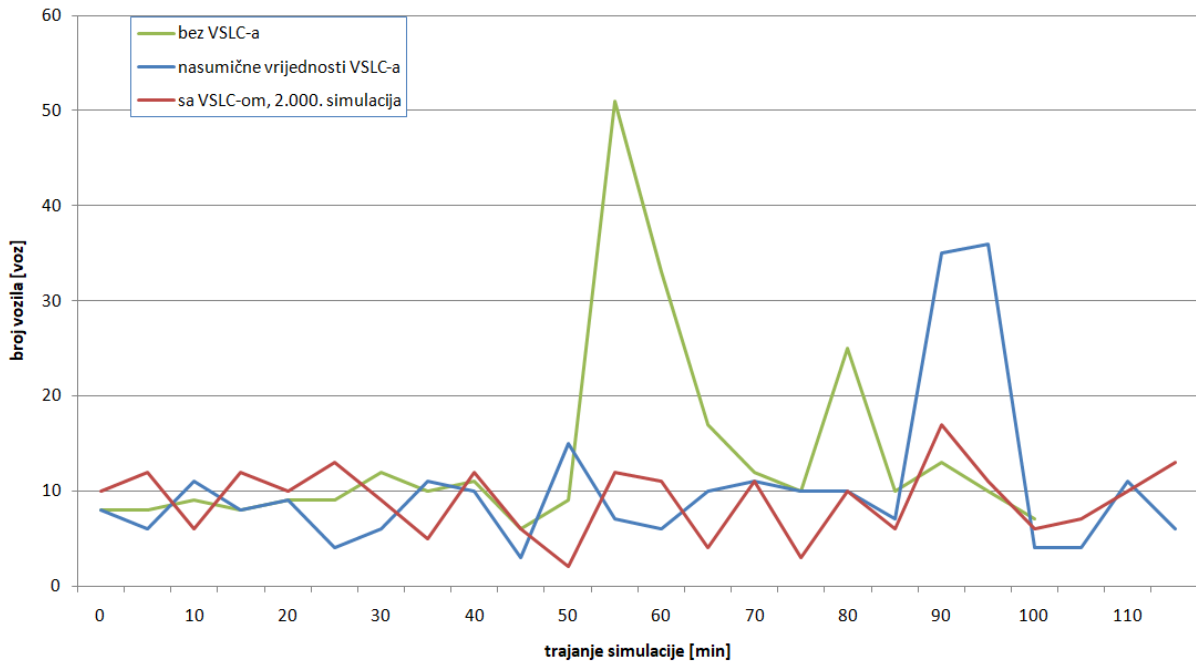
Varijabilnost prometnog toka moguće je numerički kvantificirati kao srednje odstupanje vrijednosti ograničenja brzine x_i od aritmetičke sredine brzine \bar{x} . Spomenute vrijednosti izračunate su prema formuli standardne devijacije (11) [19]:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}. \quad (11)$$

Srednja vrijednost ograničenja brzine kod nasumičnog ograničenja (plava krivulja na grafikonu 7) iznosi 82 [km/h] sa standardnom devijacijom 16,07 [km/h]. U slučaju na kraju učenja algoritma (crvena krivulja na grafikonu 7) srednja vrijednost ograničenja brzine iznosi 105 [km/h] sa standardnom devijacijom 7,07 [km/h]. Time se potvrđuje da je učenjem postignuta manja oscilacija ograničenja brzina pri čemu regulator stalno nastoji vozilima omogućiti najveću moguću brzinu vožnje uz sprječavanje pojave zagušenja.

6.2.3. Duljina repa čekanja na rampi r_2

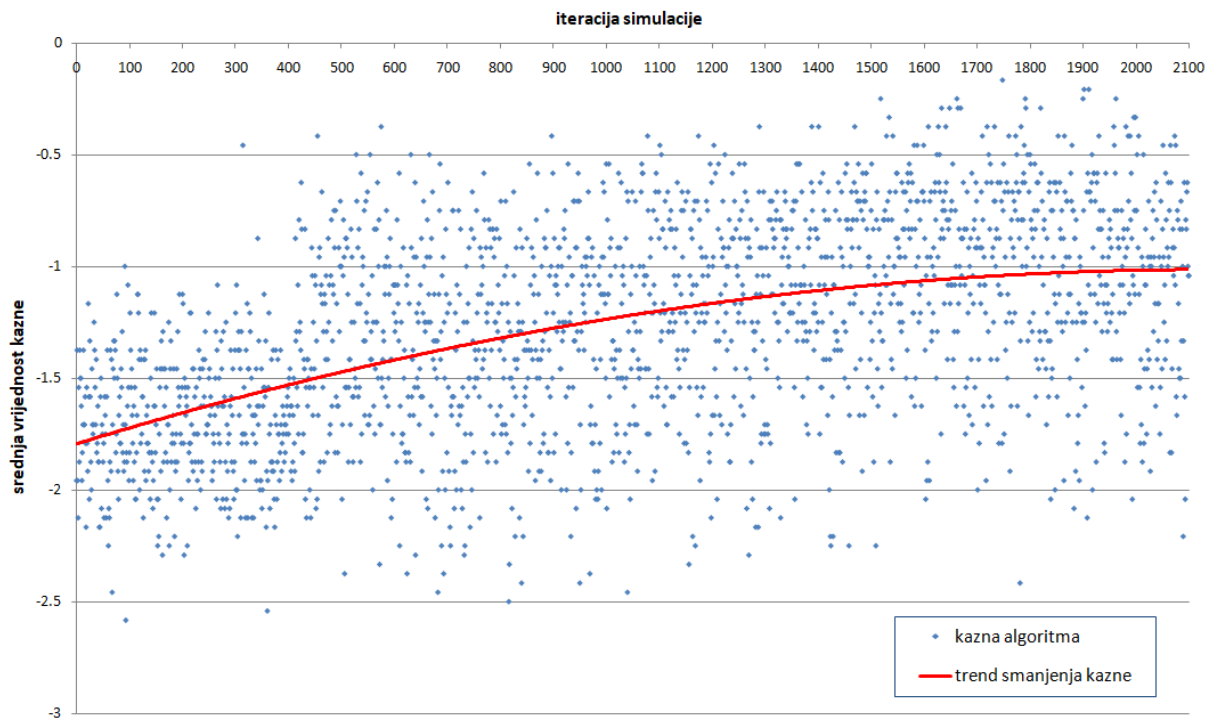
Duljina repa čekanja vozila na rampi r_2 predstavlja broj vozila koji je u zavisnosti o učinku VSLC-a na promatranom dijelu dionice autoceste l . Kada sustav umjeri prometni tok, odnosno smanji zagušenje na glavnom toku, to rezultira lakšim uključivanjem vozila s prilazne rampe r_2 u glavni tok autoceste. Time se dobiva smanjena duljina repa čekanja na pristupnoj cesti (rampa r_2). Na grafikonu 8 su prikazane krivulje za sva tri slučaja. Srednje vrijednosti duljine reda vozila na rampi u slučaju nasumične vrijednosti VSLC-a iznosi 10,3 [voz], slučaj s implementiranim algoritmom upravljanja VSLC-om iznosi 9 [voz], dok je kod slučaja bez VSLC-a srednja vrijednost 12,95 [voz]. Slučaj s naučenim VSLC-om je nešto bolji te je duljina repa čekanja na ulaznoj rampi r_2 u odnosu na slučaj nasumične vrijednosti VSLC-a smanjena za 12,9 %, a u odnosu na slučaj bez VSLC-a za 30,5 %. Također je primijećena velika oscilacija za vrijeme opservacije i za vrijeme bez VSLC-a, gdje je zabilježen velik rep čekanja od 36 vozila i 51 vozilo.



Grafikon 8. Duljina repa čekanja vozila na rampi r_2

6.2.4. Funkcija nagrade/kazne algoritma

U početku karakteristika VSLC-a zasnovanog na DDQN algoritmu je istraživanje prostora tako da je broj nasumično odabranih akcija visok. S vremenom algoritam učenja smanjuje slučajni broj odabranih akcija (izlazna ograničenja brzine) te počinje slijediti naučenu funkciju upravljačkog zakona. Nakon većeg broja simulacija algoritam polako konvergira ka optimalnom upravljačkom zakonu. Pod pojmom optimalan podrazumijeva se najbolji upravljački zakon (niz odabranih akcija) koje je algoritam naučio nakon većeg broja simulacija što ne znači da algoritam s još većim brojem simulacija ne bi naučio još kvalitetniju funkciju upravljačkog zakona. Na grafikonu 9 prikazan je trend smanjenja kazne zbog sve bolje promjene brzine tijekom simulacija, čime se potvrđuje da je algoritam sposoban učiti. Plave točkice predstavljaju srednji iznos nagrade (kazne), a crvena krivulja predstavlja trend smanjenja kazne. Negativna vrijednost nagrade zapravo predstavlja kaznu algoritma.



Grafikon 9. Trend smanjenja prosječne kazne tijekom provođenja simulacija

7. Zaključak

U svrhu izrade ovog rada i testiranja upravljačkog algoritma napravljen je model gradske autoceste s upravljanom dionicom l unutar mikroskopskog simulatora VISSIM. Upravljački algoritam implementiran je unutar programskog jezika Python korištenjem knjižnice Keras. Povezivanje skripte s algoritmom učenja i simulatora izvršeno je preko COM sučelja, čime je omogućena komunikacija i upravljanje. Povezivanjem preko spomenutog sučelja upravljano je prometnom simulacijom pomoću programskog kôda u programskom jeziku Python. Simulacija je u potpunosti upravljana pomoću skripte unutar Python-a. Podaci o prometnom toku prikupljeni su i računani pomoću funkcija izrađenih u programskom kôdu u Python-u koje pristupaju prometnim podacima u VISSIM-u.

Primjenom upravljačkog algoritma VSLC iz domene ITS-a zasnovanog na učenju dualne duboke Q-mreže na testnom modelu gradske autoceste pokazao se kao dobra osnova za daljnji razvitak algoritma. Princip rada algoritma je zasnovan na izravnom upravljanju ponašanja prometnog toka na osnovi izmjene ograničenja brzine na postavljenim VMS-ovima. Na osnovi prikupljenih i izračunatih prometnih parametara unutar svake simulacije algoritam mijenja ograničenja brzine i uči neuronsku mrežu na osnovi prikupljenih podataka s ciljem upravljanja prometa na upravljanoj dionici autoceste. Svakom iteracijom simulacije neuronska mreža postaje robusnija i točnija. U svrhu izrade rada i testiranja sustava napravljeno je simulacijsko okruženje korištenjem programskog okruženja Python/Keras i mikroskopskog simulatora VISSIM unutar kojeg je testiran implementirani sustav upravljanja ograničenjem brzine zasnovan učenju dualne duboke Q-mreže. Upravljačka logika implementirana je u programskom jeziku Python korištenjem knjižnice Keras, a prometni model je simuliran unutar programskog paketa VISSIM.

Obuhvatna zona testnog modela gradske autoceste na kojem je simuliran rad algoritma uključuje glavni tok autoceste s dvije ulazne i jednom izlaznom rampom. Model je prilagođen u svrhu omogućavanja analize utjecaja promjenjivog ograničenja brzina na prometni tok. U svrhu analize rezultata napravljen je određen broj opservacija prometa gdje algoritam ne upravlja promjenjivim ograničenjem brzine (postavljaju se nasumične vrijednosti ograničenja brzine) i određen broj simulacija za vrijeme koje algoritam uči i upravlja ograničenjem brzine. Prikazom vrijednosti prosječnog TTS-a kao glavne mjere uspješnosti algoritma prikazano je poboljšanje u smanjenju vrijednosti za promatrani model gradske autoceste.

Implementacijom algoritma primijećeno je smanjenje zagušenja i povećanje stupnja harmonizacije brzine na kritičnim područjima gradske autoceste na dionici l . Na spomenutoj dionici prilikom povećane prometne potražnje stvara se kritično zagušenje prometnog toka. Primjenom algoritma za upravljanje ograničenjem brzina znatno je smanjenja gustoća prometnog toka vozila u tom području. Mjerenjem gustoće prometnog toka primijećeno je smanjenje prosječne gustoće prometnog toka u slučaju primjene naučenog algoritma u odnosu na slučaj bez VSLC-a za 44,05 %, a u odnosu na nasumične vrijednosti VSLC-a za 48,12 %. Povećanje srednje brzine kretanja vozila naučenog algoritma u odnosu na stalno ograničenje brzine za 16,22 %, a u odnosu na nasumične vrijednosti VSLC-a za 16,13 %. Na osnovi ovih rezultata zaključeno je kako implementirani algoritam pridonosi smanjenju

intenziteta zagušenja i brzine širenja šok valova, što posljedično smanjuje ukupno provedeno vrijeme svih vozila u prometnoj mreži (parametar TTS). Unutar svakog upravljačkog intervala algoritma formulirana je povratna veza iz okoline koja je proporcionalna vrijednosti parametra TTS-a koja predstavlja funkciju nagrade (kazne) odluke algoritma. Dokazano je smanjenje prosječnog TTS-a nakon određenog broja iteracija simulacija što je prikazano u poglavlju rezultata. Nakon 2.000-te simulacije prosječna vrijednost TTS-a je smanjena u odnosu na slučaj bez VSLC-a za od prilike 13,06 %, a u odnosu na period postavljanja nasumičnih vrijednosti (opservacijski period) za od prilike 19,46 %. Na osnovi dobivenih rezultata zaključeno je da algoritam DDQN ima potencijal za primjenu u upravljanju cestovnim prometom odnosno za VSLC na gradskim autocestama.

Glavni cilj ovog diplomskog rada je implementacija i testiranje okruženja za simulaciju sustava za VSLC zasnovanog na učenju dualne duboke Q-mreže. Kod testiranja algoritama zasnovanih na učenju dualne duboke Q-mreže neophodna je nešto naprednija konfiguracija računala kako proces simuliranja ne bi trajao predugo, kao u slučaju ovog rada gdje su simulacije trajale više od 8 dana. Bolji rezultati bili bi postignuti kada bi bilo moguće napraviti još više simulacija.

Kao prijedlog za daljnji rad bio bi nastavak rada na ovoj problematici prvenstveno u pogledu implementacije mehanizama koji bi optimizirali proces učenja. Prvi korak bi bilo rješenje kojim bi se dalo više pažnje na samu strukturu konvolucijskih neuronskih mreža. Pod time se misli na prilagodbu strukture duboke neuronske mreže u smislu broja slojeva i neurona u njima. Također kvaliteta naučenog upravljačkog zakona ovisi o funkciji nagrade koja se može prilagoditi zahtjevima koji uključuju smanjenje zagađenja, povećanje sigurnosti prometa i sl. te bi bilo dobro napraviti analizu utjecaja različitih funkcija nagrada na kvalitetu naučenog upravljačkog zakona.

Popis kratica

ITS	(Intelligent Transport System) inteligentni transportni sustavi
VSLC	(Variable Speed Limit Control) upravljanje promjenjivim ograničenjem brzine
VMS	(Variable Message Signs) promjenjivi prometni znakovi
DQN	(Deep Q-Network) duboka Q-mreža
DDQN	(Dueling Deep Q-Network) dualna duboka Q-mreža
TTS	(Total Time Spent) ukupno provedeno vrijeme svih vozila u prometnoj mreži

Literatura

- [1] Centar za vozila Hrvatske, Broj novih i rabljenih vozila na prvom tehničkom pregledu (2007-2018), <https://www.cvh.hr/tehnicki-pregled/statistika/> [pristupljeno 29. Lipnja 2019.].
- [2] Korent N, Kušić K. Analiza utjecaja promjenjivog ograničenja brzine na protočnost prometa i zagađenje okoliša, Rektorova nagrada, Fakultet prometnih znanosti, Zagreb, 2016.
- [3] Wikipedia, Zagreb bypass, https://en.wikipedia.org/wiki/Zagreb_bypass [pristupljeno 2. Srpnja 2019.].
- [4] Galić I. Usporedba metoda za upravljanje priljevnim tokovima na primjeru zagrebačke obilaznice, Diplomski rad: Sveučilište u Zagrebu, Fakultet prometnih znanosti, 2015.
- [5] Gregurić M, Ivanjko E, Korent N, Kušić K. Short Review of Approaches for Variable Speed Limit Control, ZIRP2016, Zagreb, Croatia, 12.04.2016. str. 41-52, 2016.
- [6] Kušić K. Okruženje za simulaciju sustava za upravljanje promjenjivim ograničenjem brzine na urbanim autocestama zasnovanih na učenju, Diplomski rad: Sveučilište u Zagrebu, Fakultet prometnih znanosti, 2017.
- [7] Korent N. Usporedba algoritama za upravljanje promjenjivim ograničenjem brzine na urbanim autocestama, Diplomski rad: Sveučilište u Zagrebu, Fakultet prometnih znanosti, 2017.
- [8] Kušić K, Ivanjko E, Gregurić M. A Comparison of Different State Representations for Reinforcement Learning Based Variable Speed Limit Control, 26th Mediterranean Conference on Control and Automation (MED), Zadar, 2018.
- [9] Sutton RS, Barto AG. Reinforcement Learning: An Introduction, The MIT Press Cambridge, Massachusetts London, England, 2002.
- [10] Walraven E. Traffic Flow Optimization using Reinforcement Learning, Master's Thesis, Faculty EEMCS, Delft University of Technology, Delft, the Netherlands, 2014.
- [11] Oppermann A. Self Learning AI-Agents Part II: Deep (Double) Q-Learning, Towards Data Science, 2018.
- [12] Calvo JJA. Heterogeneous Multi-Agent Deep Reinforcement Learning for Traffic Lights Control, Dissertation thesis, University of Dublin, Trinity College, 2018.

- [13] Skymind, A Beginner's Guide to Deep Reinforcement Learning, <https://skymind.com/wiki/> [preuzeto 8. Srpnja 2019.].
- [14] Simonini T. Improvements in Deep Q Learning: Dueling Double DQN, Prioritized Experience Replay, and fixed..., <https://www.freecodecamp.org/news/improvements-in-deep-q-learning-dueling-double-dqn-prioritized-experience-replay-and-fixed-58b130cc5682/> [preuzeto 10. Srpnja 2019.].
- [15] Group PTV, PTV VISSIM, <http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/> [preuzeto 7. Srpnja 2019.].
- [16] Essert M, Vazler I. Python – osnove, Odjel za matematiku Sveučilišta u Osijeku, Osijek, 2007.
- [17] Keras, Guiding principles, <https://keras.io/> [preuzeto 8. Srpnja 2019.].
- [18] Dadić I, Kos G, Ševrović M. Teorija prometnog toka, Zagreb: Sveučilište u Zagrebu, Fakultet prometnih znanosti, 2014.
- [19] Kovač Striko E, Fratrović T, Ivanković B. Vjerojatnost i statistika s primjerima iz tehnologije prometa, Zagreb: Sveučilište u Zagrebu, Fakultet prometnih znanosti, 2008.

Popis slika

Slika 1. Zagrebačka obilaznica.....	3
Slika 2. Mjesto nastanka uskog grla i posljedice preljeva prometa [4]	4
Slika 3. Shematski prikaz dubokog ojačanog učenja [11]	8
Slika 4. Ilustracija mapiranja stanja agenta [13]	9
Slika 5. Prikaz funkcije vrijednosti akcije [11]	10
Slika 6. Pojednostavljena blok shema algoritma DDQN.....	12
Slika 7. Grafički prikaz modela gradske autoceste	13
Slika 8. Prikaz korisničkog sučelja simulatora VISSIM.....	15
Slika 9. Korisničko sučelje platforme Anaconda	17
Slika 10. Korisničko sučelje jupyter Notebook.....	17
Slika 11. Python/Keras-VISSIM simulacijska shema	18
Slika 12. Pojednostavljena blok shema izvršavanja simulacije	20
Slika 13. Definirana distribucija brzina za teretna vozila unutar VISSIM-a	21
Slika 14. Definirana distribucija brzina za autobuse unutar VISSIM-a	22
Slika 15. Definirana distribucija brzina za osobna vozila unutar VISSIM-a.....	22
Slika 16. Parametri slijeđenja vozila u VISSIM-u	25
Slika 17. Dodatni parametri slijeđenja vozila u VISSIM-u	25
Slika 18. Parametri promjene prometnih traka.....	26
Slika 19. Parametri bočnog kretanja vozila	26
Slika 20. Prikaz uzorkovanja podataka o vozilima	27

Popis tablica

Tablica 1. Zastupljenosti različitih tipova vozila	23
Tablica 2. Primjer matrice podataka prikupljenih za svako vozilo u mreži	27
Tablica 3. Prosječne vrijednosti TTS-a u prometnom modelu za intervale mjerenja T	29
Tablica 4. Srednje vrijednosti prometnih parametara na dionici autoceste l	32

Popis grafikona

Grafikon 1. Prometna potražnja na glavnom toku i ulaznim rampama	24
Grafikon 2. Prosječna vrijednost TTS-a za svaki interval T	28
Grafikon 3. Trend smanjenja prosječnog TTS-a tijekom provođenja simulacija	29
Grafikon 4. Protok vozila na dionici l tijekom simulacije	30
Grafikon 5. Gustoća vozila na dionici l tijekom simulacije.....	31
Grafikon 6. Srednja brzina vozila na dionici l tijekom simulacije.....	31
Grafikon 7. Iznos ograničenja brzine na dionici autoceste l	33
Grafikon 8. Duljina repa čekanja vozila na rampi r_2	35
Grafikon 9. Trend smanjenja prosječne kazne tijekom provođenja simulacija	36



Sveučilište u Zagrebu
Fakultet prometnih znanosti
10000 Zagreb
Vukelićeva 4

IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOST

Izjavljujem i svojim potpisom potvrđujem kako je ovaj _____ diplomski rad isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu što pokazuju korištene bilješke i bibliografija.

Izjavljujem kako nijedan dio rada nije napisan na nedozvoljen način, niti je prepisan iz necitiranog rada, te nijedan dio rada ne krši bilo čija autorska prava.

Izjavljujem također, kako nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

Svojim potpisom potvrđujem i dajem suglasnost za javnu objavu _____ diplomskog rada pod naslovom **OKRUŽENJE ZA SIMULACIJU SUSTAVA ZA UPRAVLJANJE PROMJENJIVIM OGRANIČENJEM BRZINE NA GRADSKIM AUTOCESTAMA ZASNOVANO NA DUBOKOM UČENJU** na internetskim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom akademskom repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.

U Zagrebu, 9/11/2019 _____

Student/ica:

Filip Urbanić

(potpis)