Okruženje za simulaciju sustava za upravljanje promjenjivim ograničenjem brzine na urbanim autocestama zasnovanih na učenju

Kušić, Krešimir

Master's thesis / Diplomski rad

2017

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:119:016117

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**



Repository / Repozitorij:

Faculty of Transport and Traffic Sciences - Institutional Repository







SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

Krešimir Kušić

OKRUŽENJE ZA SIMULACIJU SUSTAVA ZA UPRAVLJANJE PROMJENJIVIM OGRANIČENJEM BRZINE NA URBANIM AUTOCESTAMA ZASNOVANIH NA UČENJU

DIPLOMSKI RAD

Zagreb, 2017.

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI
POVJERENSTVO ZA DIPLOMSKI ISPIT

Zagreb, 24. travnja 2017.

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

DIPLOMSKI ZADATAK br. 3931

Pristupnik: **Krešimir Kušić (0135227681)**
Studij: Inteligentni transportni sustavi i logistika
Smjer: Inteligentni transportni sustavi

Zadatak: **Okruženje za simulaciju sustava za upravljanje promjenjivim ograničenjem brzine na urbanim autocestama zasnovanih na učenju**

Opis zadatka:

U uvjetima povećane prometne potražnje na urbanim autocestama javlja se potreba za uvođenjem upravljanja prometom s ciljem prevencije i smanjenja posljedica nastajanja prometnih zagušenja. Jedno od rješenja iz područja ITS-a je sustav za upravljanje ograničenjem brzine na urbanim autocestama. Noviji pristup u upravljanju promjenjivim ograničenjem brzine na urbanim autocestama koristi algoritme zasnovane na učenju. U radu je potrebno opisati upravljanje prometom na urbanim autocestama zasnovano na upravljanju promjenjivim ograničenjem brzine, analizirati mogućnosti primjene učenja u upravljanju promjenjivim ograničenjem brzine te izraditi okruženje za simulaciju takvog upravljanja korištenjem mikroskopskog simulatora VISSIM i programskog paketa MATLAB.

Zadatak uručen pristupniku: 28. travnja 2017.

Mentor:

doc. dr. sc. Edouard Ivanjko

Predsjednik povjerenstva za diplomski ispit:

Sveučilište u Zagrebu
Fakultet prometnih znanosti

DIPLOMSKI RAD

OKRUŽENJE ZA SIMULACIJU SUSTAVA ZA UPRAVLJANJE PROMJENJIVIM OGRANIČENJEM BRZINE NA URBANIM AUTOCESTAMA ZASNOVANIH NA UČENJU

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

Mentor: doc. dr. sc. Edouard Ivanjko
Student: Krešimir Kušić, 0135227681

Zagreb, srpanj 2017.

Zahvala

U prvom redu dugujem veliku zahvalnost svom mentoru i profesoru doc. dr. sc. Edouardu Ivanjku na pruženoj prilici i stručnim savjetima pri izradi ovog diplomskog rada.

Zahvaljujem se mojoj obitelji na velikoj podršci i razumijevanju tijekom mog studiranja.

Također, hvala i mojim prijateljima koji su mi bili potpora u studiranju.

Sažetak

U uvjetima povećane prometne potražnje na urbanim autocestama javlja se potreba za uvođenjem upravljanja prometom s ciljem prevencije i smanjenja posljedica nastajanja prometnih zagušenja. Jedno od rješenja iz područja inteligentnih transportnih sustava je sustav za upravljanje ograničenjem brzine na urbanim autocestama. Noviji pristup u upravljanju promjenjivim ograničenjem brzine na urbanim autocestama koristi algoritme zasnovane na učenju. U radu je opisano upravljanje prometom na urbanim autocestama zasnovano na upravljanju promjenjivim ograničenjem brzine, analizirana je mogućnost primjene učenja s posebnim naglaskom na algoritmu *Q-učenja* u upravljanju promjenjivim ograničenjem brzine te je izrađeno okruženje za simulaciju takvog upravljanja korištenjem mikroskopskog simulatora VISSIM i programskog paketa MATLAB.

KLJUČNE RIJEČI: upravljanje promjenjivim ograničenjem brzine; simulacijsko okruženje; algoritmi zasnovani na učenju

Summary

In the conditions of increased traffic demand on urban motorways, there is a need for dynamic traffic management. Its aim is preventing and reducing the consequences of traffic congestion on urban motorways. One of the acceptable solutions is the application of services from the domain of Intelligent Transport System. One of such services is variable speed limit control (VSLC). The newer approach to managing variable speed limitation on urban motorways utilizes “algorithm based on learning”. This master’s thesis describes dynamic traffic management on urban motorways that is based on VSLC. In addition, it describes the ability to apply learning for VSLC with an emphasis on the Q-learning algorithm. An environment for simulating such system is developed using the microscopic simulator VISSIM and the MATLAB software package.

KEYWORDS: variable speed limit control; framework for simulation; reinforcement learning; Q-learning

Sadržaj

1. Uvod 1
2. Problem upravljanja promjenjivim ograničenjem brzine i najčešće metode 3
3. Primjena učenja u upravljanju promjenjivim ograničenjem brzine 7
   3.1. Potporno učenje 7
   3.2. Metoda učenja s vremenskom razlikom u potpornom učenju 12
      3.2.1. SARSA algoritam 12
      3.2.2. Algoritam *Q-učenja* 13
   3.3. Upravljanje promjenjivim ograničenjem brzina zasnovano na algoritmu *Q-učenja* 14
4. Značajke mikroskopskog simulatora VISSIM i programskog paketa MATLAB 19
5. Povezivanje VISSIM-a i MATLAB-a 22
6. Simulacijski rezultati 28
   6.1. Model i podaci 28
   6.2. Postavke učenja 31
   6.3. Dobiveni rezultati 33
      6.3.1. TTS kao glavna mjera uspješnosti učenja 33
      6.3.2. Protok, gustoća i brzina 35
      6.3.3. VSLC - ograničenja brzine za ćeliju L_2 38
      6.3.4. Duljina reda čekanja na rampi r_2 40
7. Zaključak 42

Literatura 44

Popis kratica 46

Popis slika 47

Popis tablica 48

Popis grafikona 49

1. Uvod

Danas način života u urbanim sredinama podrazumijeva veliku mobilnost građana. Čest je slučaj da građani obavljajući svoje svakodnevne zadatke (posao, fakultet, škola, kupovina, zabava itd.) moraju putovati u udaljene dijelove gradova u odnosu na mjesto stanovanja. Takve dnevne migracije uzrokuju dodatnu prometnu potražnju unutar gradova. S obzirom na mentalitet ljudi na ovim područjima najkorištenije prijevozno sredstvo je osobni automobil. Uz sve veću prometnu potražnju na prometnicama u gradovima javlja se problem manjka kapaciteta istih. U vršnim satima kapacitet na pojedinim prometnicama nije dostatan. Mogućnosti širenja prometne infrastrukture (cestovne prometne mreže) unutar gradova je ograničena pomanjkanjem prostora.

Ne ulazeći u analizu zašto javni gradski i prigradski promet nije bolje iskorišten (subvencije, bolja povezanost, češće linije itd.) u cilju smanjenja osobnih automobila na gradskim prometnicama, jedno od rješenja je izgradnja gradskih obilaznica. One preuzimaju dio gradskog prometa rasterećujući tako gradske prometnice. Osim toga obilaznice imaju svrhu preuzimanja tranzitnog prometa koji više ne mora na svom putu prolaziti kroz središta gradova. Tako projektirane urbane cestovne prometnice se zovu urbane autoceste. Dimenzionirane su da pružaju veći prometni kapacitet i veće brzine prometovanja. Uz sve veće širenje cestovnog prometa ponekad su i one opterećene iznad svog maksimalnog kapaciteta. Hipotetski mogle bi se one i proširiti (dodatni prometni trak) u svrhu povećanja kapaciteta, no to baš nije svaki puta ekonomski isplativo.

Uz sve strože ekološke, sigurnosne i eksploatacijske standarde rješenje zagušenja na urbanim autocestama predstavlja veliki izazov za prometne stručnjake u traženju odgovarajućih optimalnih rješenja. Iz tog razloga znanstvena i stručna zajednica sve više pridaje pozornost razvoju upravljačkih rješenja kojima bi se moglo unaprijediti vođenje prometa u kritičnim situacijama s ciljem povećanja maksimalne propusnosti na već postojećoj prometnoj infrastrukturi urbane autoceste. Danas postoje razni pristupi iz područja inteligentnih transportnih sustava (ITS) kojima se pokušavaju odrediti odnosno izračunati određena stanja prometa, te na osnovu njih u stvarnom vremenu preko izvršnih elemenata kao što su promjenjivi prometni znakovi (VMS), paneli za obavijesti vozačima itd. upravljati prometom za i nakon vršnih opterećenja. Jedno od takvih sustava upravljanja prometom je i upravljanje promjenjivim ograničenjem brzine (engl. *variable speed limit control* - VSLC) na urbanim autocestama koje izravno utječe na dinamiku prometnog toka promjenom ograničenja brzine na pojedinim dijelovima promatrane dionice autoceste. Algoritmi na kojima se zasniva upravljačka logika takvih sustava ima mnogo. Danas su popularni i sve se više istražuju te testiraju algoritmi zasnovani na strojnom učenju. Jedan takav pristup je primijenjen i testiran u ovom radu, a riječ je o upravljanju promjenjivim ograničenjem brzine koje je zasnovano na *Q-učenju*. Da bi se uopće dokazala efikasnost ovakvih rješenja prije implementacije u realno okruženje potrebno je takav sustav testirati u „prometnom laboratoriju“. Prometni laboratorij u ovom smislu predstavlja simulacijsko okruženje za testiranje različitih rješenja na simulacijskom modelu unutar odgovarajućeg programskog simulacijskog alata. U ovom diplomskom radu napravljeno je simulacijsko okruženje korištenjem programskih alata VISSIM i MATLAB unutar kojeg je

testiran VSLC zasnovan na *Q-učenju*. Svrha ovog diplomskog rada je dati uvid u kreiranje okruženja za simulaciju sustava VSLC-a na osnovu koje se vrši provjera i testiranje upravljačkog algoritma zasnovanog na učenju. Dodatna svrha je provjeriti mogućnosti primjene mikroskopskog simulatora VISSIM i programskog paketa MATLAB za analizu VSLC sustava zasnovanih na strojnom učenju. Cilj ovog diplomskog rada je analiza VSLC algoritma zasnovanog na *Q-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 brzina. Treće poglavlje opisuje primjenu učenja u upravljanju promjenjivim ograničenjem brzina. Tu je dat pregled osnovnih pojmova o potpornom učenju i algoritmima koji se danas koriste. Na kraju poglavlja napravljen je implementacijski postupak VSLC sustava zasnovanog na *Q-učenju*, koji je kasnije implementiran u MATLAB-u u obliku programskog k*ô*da te testiran u simulacijskom okruženju. U četvrtom poglavlju dan je kratak opis mikroskopskog simulatora VISSIM i programskog paketa MATLAB. Peto poglavlje objašnjava na koji način su povezana ova dva programska alata koji zajedno čine okruženje za simulaciju VSLC-a zasnovanog na učenju. U šestom poglavlju je opisan testni prometni model urbane autoceste s ulaznim podacima u simulatoru, detaljnije su definirane postavke učenja u Q-algoritmu, te na kraju prikazani rezultati simulacije testiranog sustava. Sedmo poglavlje odnosi se na zaključak i spoznaje do kojih se došlo tijekom izrade ovog diplomskog rada.

2. Problem upravljanja promjenjivim ograničenjem brzine i najčešće metode

Prometni sustav iz dana u dan postaje opterećeniji, a taj porast je naročito vidljiv u cestovnom prometu. Naročito u urbanim sredinama. Raste broj cestovnih vozila, a time se povećava pojava prometnih zagušenja, produljuje se vrijeme čekanja i smanjuje sigurnost na prometnicama. To je posebno prisutno na autocestama gdje su brzine veće. Povećanjem zagušenja i vremena čekanja povećava se potrošnja goriva, što kao posljedicu ima i povećanu emisiju štetnih ispušnih plinova, odnosno povećano zagađenje okoliša. Iz tog razloga grade se obilaznice, odnosno urbane autoceste oko većih gradova. Urbane autoceste su projektirane na način da omogućuju vrlo brzo i sigurno odvijanje prometa. One su veoma bitne u prometnoj mreži velikih gradova jer rasterećuju gradske prometnice [1].

Urbane autoceste pružaju veći prometni kapacitet s ciljem da smanje prometno opterećenje unutar gradske prometne mreže. Najčešće se nalaze na rubnim dijelovima gradova (npr. zagrebačka obilaznica). Veliki broj ulaznih i izlaznih rampi dobro ju povezuje s lokalnom urbanom mrežom. S povećanjem prometnih zahtijeva (naročito tijekom vršnih sati) prometna potražnja često se približava maksimalnom kapacitetu prometnice. U takvim uvjetima kvaliteta usluge (engl. *level of services* - LoS) se smanjuje, što se izražava kroz dulja vremena putovanja, manje brzine te učestale promjene brzine (kočenje i ubrzanje). Kada prometna potražnja premaši kapacitet autoceste dolazi do poremećaja u prometno toku što se izražava u obliku povremenih zagušenja što ponekad dovodi do potpunih zastoja u odvijanju prometa. Kako je promet stohastičke prirode i njegovo trenutno stanje ovisi o ponašanju sudionika, vrlo je teško predvidjeti kada će doći do zagušenja, iako je moguće donekle predvidjeti gdje će doći do smanjenja propusnosti autoceste. Posljedično smanjenjem propusnosti dolazi do prekoračenja kapaciteta autoceste. Takve situacije su česte u blizini ulaznih rampi (ulaz na autocestu). Zagušenje cestovnog prometa je stanje karakterizirano smanjenjem prosječne brzine prometnog toka na prometnici, produljenjem vremena putovanja i povećanjem duljine repova čekanja, odnosno duljine kolone.

Tijekom dimenzioniranja kapaciteta autocesta teško je točno predvidjeti buduće promjene prometne potražnje na glavnom toku i pripadnim priljevnim tokovima. Indirektno izgradnja takve prometnice uvijek privlači i gospodarske subjekte (tvornice, skladišta itd.) u svoju blizinu zbog bolje prometne povezanosti. To posljedično dovodi još više vozila na samu autocestu. Naročito je tu bitan budući porast 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 urbane autoceste. Zagušenja na autocestama su tako najčešća 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 urbanih područja tijekom ranih jutarnjih i kasnih poslijepodnevnih (vršnih) sati. Problemi nastaju zbog dnevnih migracija na relaciji posao/škola/vrtić - dom. Tijekom tog perioda značajno je narušena propusnost i sigurnost prometnih sudionika na urbanoj autocesti.

Kako bi se izbjegle ovakve situacije, jedno rješenje je povećanje kapaciteta autoceste izgradnjom dodatne infrastrukture (izgradnja dodatnog prometnog traka). Većinu vremena dodatna izgrađena infrastruktura je neiskorištena, osim za vremena vršnih sati. Takvo rješenje je skupo za realizaciju, a kasnije i neekonomično s eksploatacijskog aspekta. Struka se sve više okreće metodama i tehnologijama kojima je moguće dodatno iskoristiti već postojeću prometnu infrastrukturu u svrhu povećanja propusnosti prometnica ili preraspodjelom prometne potražnje na manje opterećene prometnice kada je to potrebno. Klasičan način gradnje dodatne prometne infrastrukture pokušava se izbjeći ukoliko nije neophodno.

Upravljanje prometom na urbanim autocestama moguć je na osnovu nekoliko kriterija. Jedan od njih je sigurnost i upravljanje sigurnošću prometa na autocestama koje postaje sve značajnije. Dodatni kriterij je upravljanje protočnošću prometa kako bi se smanjilo zagušenje. Ovaj rad se odnosi na kriterij vezan za smanjivanje zagušenja pomoću upravljanja protočnošću. U takvom upravljanju susrećemo se s dva glavna problema koja uzrokuju zagušenje, a to su priljevni tokovi te nehomogene brzine vozila u glavnom toku. U slučaju veće potražnje za nekom relacijom, u urbanoj sredini, dolazi do zagušenja na pripadnoj ulaznoj rampi te novog zagušenja na autocesti oko pripadne ulazne rampe. Posljedično se produljuje vrijeme putovanja svim vozilima. Kako se pri kapacitetnom dimenzioniraju autocesta teško može predvidjeti prometna potražnja na priljevnim tokovima, potrebno je pronaći druga rješenja za dani problem korištenjem upravljačkih pristupa iz domene ITS-a [1].

Primjena naprednih sustava iz domene ITS-a, kao što je dinamičko upravljanje prometom omogućuje poboljšanje kvalitete usluge upravljane prometnice. Osnova rada ovakvih sustava je trenutna razmjena podataka prikupljenih od senzora na prometnoj infrastrukturi. Da bi to bilo ostvarivo potrebna je odgovarajuća informacijsko-komunikacijska infrastruktura putem koje se podaci šalju, analiziraju te razmjenjuju između korisnika i pružatelja usluga unutar samog sustava. Dinamičko upravljanje prometom uključuje dva konkretna pristupa pomoću kojih je moguće kontrolirati prometne uvjete na urbanoj autocesti pri povećanoj prometnoj potražnji.

Prva strategija zasniva se na upravljanju količinom vozila na ulaznim rampama koje se priključuju u glavni tok urbane autoceste, tzv. upravljanje priljevnim tokom (engl. *ramp metering* - RM). Konkretno regulira se broj vozila koja se uključuju na autocestu (npr. pomoću algoritma Alineja [2]). Drugi sustav se zasniva na upravljanju promjenjivim ograničenjem brzine (engl. *variable speed limit control* - VSLC). VSLC sustav izravno regulira prometni tok vozila na urbanim autocestama izmjenom ograničenja brzina na promjenjivim prometnim znakovima (engl. *variable message signs* - VMS). VSLC sustav pokazuje informaciju o ograničenju brzine na prikladnim promjenjivim znakovima. Prema tome, vrijednosti ograničenja brzine se mogu mijenjati u odnosu na trenutnu prometnu situaciju korištenjem odgovarajućeg algoritma upravljanja. U slučaju pojave zagušenja na nekom dijelu autoceste, veliki problem predstavlja i nehomogena brzina kretanja vozila. Ako dođe do preopterećenja dionice urbane autoceste zbog dolaska vozila s priljevne rampe stvara se zagušenje i postoji mogućnost stvaranja prometnog čepa. U tom slučaju prometni zastoj se širi unatrag uzduž urbane autoceste. Također je poznato da pri velikim odstupanjima brzina kretanja vozila od prosječne brzine prometnog toka postoji velika vjerojatnost nastanka incidentnih situacija. Na mjestima gdje vozila s velikim prilazećim brzinama dolaze do kraja prometnog reda, ona

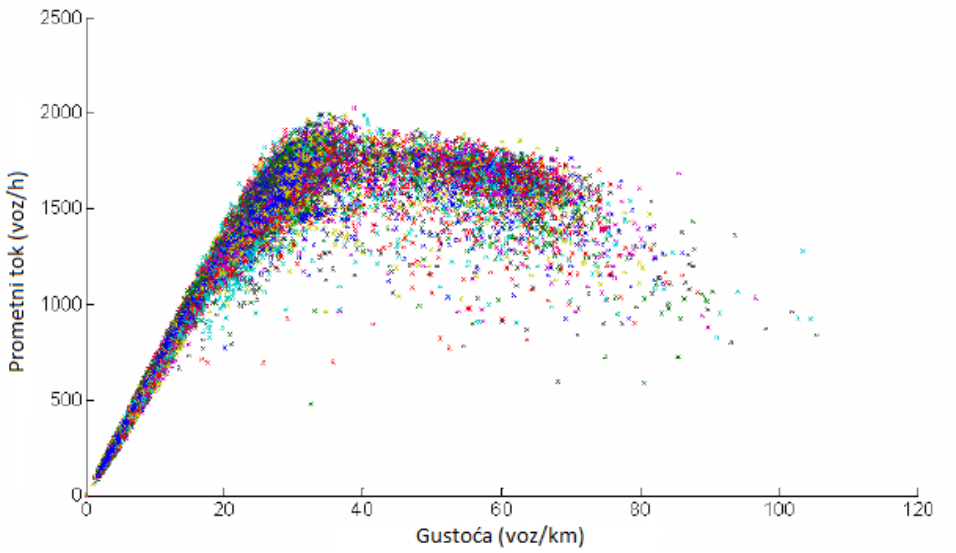
moraju naglo smanjivati brzinu time povećavajući mogućnost sudara. Takva vožnja izaziva nepravilnosti u odvijanju prometnog toka, te uz povećanje rizika od sudara, izaziva stres i frustraciju vozača jer im onemogućuje putovanje ujednačenom brzinom.

VSLC sustavi se najčešće dijele na dvije grupe [1]: (i) VSLC sustav zasnovan na vremenskim uvjetima; i (ii) VSLC sustav zasnovan na zagušenjima. Prvi sustav se koristi na urbanim autocestama gdje su pojave magle, leda, kiše ili drugih vremenski nepovoljnih uvjeta često utjecajan faktor na sigurnost prometnice. Kada se pogoršaju vremenski uvjeti do predefinirane kritične točke smanjuje se ograničenje brzine na prikladnu brzinu kako bi se smanjile mogućnosti nastanka incidentnih situacija. Druga vrsta sustava se koristi kada se povećava prometni volumen i izgledna je pojava zagušenja. Kada su volumeni i/ili gustoća toka prešle određenu predefiniranu granicu, aktivira se upravljačka strategija kojom se smanjuje ograničenje brzine. Smanjenje brzine ograničenja može se desiti automatski prema već 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 *Q-učenju*.

Glavni operativni razlozi uvođenja VSLC sustava na autocesti su:

- 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.

U radu je naglasak na drugu vrstu VSLC sustava zasnovanog na detekciji zagušenja prometnog toka. Dva klasična algoritma na kojima se zasniva rad VSLC regulatora su (engl. *mainline virtual metering* - MVM) i (engl. *simple proportional speed limit controller* - SPSC) [3]. Oba su reaktivna i zasnovana na osnovnoj zavisnosti (toka-gustoće kao što je prikazano na slici 1) preslikanoj u iznos brzine.



Slika 1. Osnovna zavisnost prometnog toka o gustoći [4]

Problem kod ove vrste regulatora predstavlja njihova ograničenost u smislu širine spektra djelovanja. Naime sposobni su polučiti rezultat u uskom području varijacije prometne potražnje, a preduvjet toga su dobro kalibrirani parametri unutar samog algoritma.

Danas znanstvena zajednica sve više pridaje pozornost razvoju i testiranju VSLC čiji se princip rada zasniva na strojnom učenju. Jedan takav pristup učenja zasnovan je na algoritmu *Q-učenja*. Prednosti ovakvih sustava očituju se u mogućnosti samoučenja, te kontinuiranog poboljšanja upravljačkog zakona tijekom rada. Također su sposobni obraditi veliki broj prikupljenih ulaznih parametara koji su potrebni za kvalitetan opis prometne situacije, te na osnovu njih predlagati rješenja. U odnosu na klasične analitičke pristupe nude određenu dozu robusnosti i mogućnost bržeg donošenja zadovoljavajućeg rješenja za konkretan problem. Naročito kada se s vremenom mijenja karakteristika prometnice i prometnog toka kojim upravljaju.

3. Primjena učenja u upravljanju promjenjivim ograničenjem brzine

Danas uz napredak računalne tehnologije (poboljšanje performansi u obradi velike količine podataka) rješenja koja su proučavana polovicom prošlog stoljeća zasnovana na iterativnim postupcima računanja zaživjela su u punom smislu. Veliki broj istraživačkih radova u području dinamičkog upravljanja prometnim tokom zasnivaju se na obradi velikog broja ulaznih podataka u stvarnom vremenu. Glavni razlog primjene računala kod rješavanja takvih problema zasniva se na činjenici da znanstvena zajednica i struka sve više proučavaju „zadovoljavajuća“ rješenja (koja su približna točnim rješenjima). Točna rješenja kod dinamičkih sustava kao što je prometni tok ponekad nisu ni moguća (npr. opis prometnog toka kao fluida sadrži sustave jednadžbi koje sadrže derivacije s nekoliko varijabli, te je pronalazak funkcije rješenja koja bi zadovoljavala takav „nelinearni sustav“ težak posao). Danas aproksimativne metode iz numeričke matematike uz primjenu brzih računala vrlo dobro (do zadovoljavajuće točnosti) rješavaju takve i slične teške probleme iz područja tehničkih znanosti [5].

U potpornom učenja situacija je slična. Jedna od metoda potpornog učenja je algoritam *Q-učenja* zasnovan na rekurzivnim Bellman-ovim jednadžbama. Da bi *Q-učenje* konvergiralo ka određenoj optimalnoj politici (engl. *optimal policy*) upravljanja potreban je veliki broj iterativnih izračuna Q vrijednosti. Više detalja je dano u narednim potpoglavljima. Osnova ovakvog učenja je metoda pokušaja i pogrešaka na osnovu kojih računalo (agent) uči. Da bi učenje bilo uspješno, potreban je veliki broj epizoda učenja (moguće uz moderna računala). Neki od pionira u razvoju algoritama zasnovanih na učenju u području prometa (adaptivno upravljanje semaforiziranim raskrižjima) je istraživačka skupina s odsjeka za ITS tehnologije sa Sveučilišta Toronto [6]. Daljnji radovi danas su prošireni na upravljanje promjenjivim ograničenjem brzina na urbanim autocestama. Tu postoje radovi sa Sveučilišta u Nizozemskoj, fakultet EEMCS [7]. U idućim potpoglavljima objašnjena je osnova potpornog učenja, te su detaljnije opisane metode učenja s vremenskom razlikom (engl. *Temporal difference*) pod koju spada i algoritam *Q-učenja* primijenjen u ovom radu. Od drugih pristupa još je moguće spomenuti rad sa Sveučilišta u Zagrebu, Fakultet prometnih znanosti gdje je testiran sustav upravljanja priljevnih tokova na rampama RM zasnovan na *Q-učenju* [8], te rad sa Sveučilišta Sv. Kliment Ohridski, Fakultet tehničkih znanosti gdje su napravljena istraživanja na primjeni *Q-učenja* u adaptivnom upravljanju izoliranim semaforiziranim raskrižjem u gradu Bitoli, Makedonija [9].

3.1. Potporno učenje

Potporno učenje (engl. *Reinforcement Learning* - RL) je grana strojnog učenja koja proučava sustave koji su u stanju učiti iz podataka. RL se zasniva na agentima koji su u interakciji sa svojom okolinom. Klasični pristupi strojnog učenja zasnivaju se na nadgledanom učenju dok kod RL-a agent mora biti sposoban sam učiti iz vlastitih iskustava metodom pokušaja i pogrešaka. Tijekom učenja agent uči kako izvršiti akcije u okruženju s ciljem

povećanja kumulativne nagrade. Nagrada postoji kao povratna informacija koju agent prima nakon izvršenja akcije.

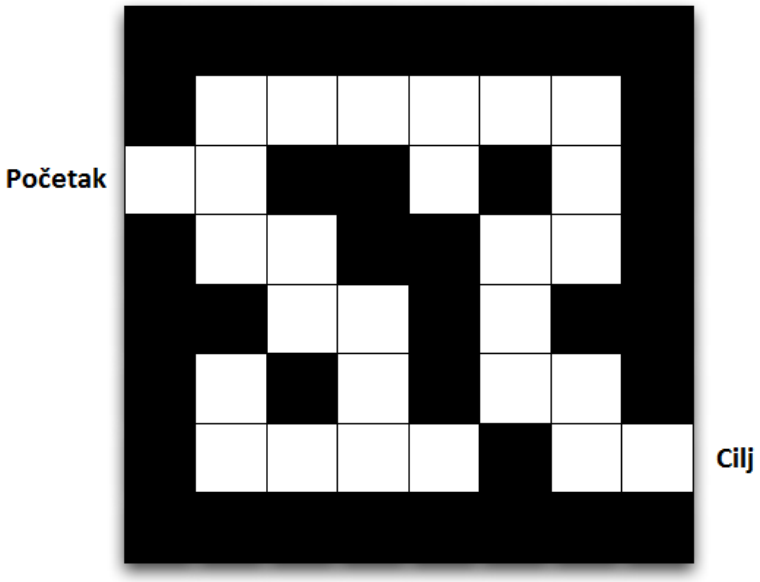
Okolina (model) u kojoj se nalazi agent opisana je ili se pretpostavlja da ima svojstva Markovljevog procesa odlučivanja (engl. *Markov Decision Process* - MDP). Proces definira skup stanja i akcija koje se mogu izvršiti s ciljem upravljanja okolinom. Praktični problemi, poput optimizacije dizala, hod robota, upravljanje helikopterom, upravljanje brodom mogu se modelirati kao MDP. Takvo učenje gdje je model okoline poznat naziva se učenje zasnovano na modelu (engl. *Model-Based Learning*) [10].

S druge strane kada okolina nije poznata učenje se naziva učenja bez modela (engl. *Model-Free Learning*). Takvi modeli uče optimalnu politiku (engl. *policy*) (optimiziraju funkciju strategije odabira akcija) na osnovu iskustva, kontinuirano promatrajući dobivene nagrade nakon primijenjenih akcija. Prethodno formulirani praktični problemi se ponekad zbog složenosti dinamike okoline ne opisuju kao MDP. Umjesto toga opisani su prikupljenim nizovima uzoraka stečenih iskustvom koje agent prima izravno iz okoline. Isto i u slučaju kada je model okoline poznat ali je prevelik za pretraživanje. Upravljanje bez modela (engl. *Model-free control*) rješava ovakve probleme [10], [11].

Osnovni elementi koji opisuju proces potpornog učenja, opis modela i nagrada u ojačanom učenju su [12]:

- agent (apstrakcija upravljačkog regulatora),
- stanja koja opisuju okolinu agenta,
- akcije koje agent poduzima,
- strategija odabira akcija ili politika (engl. *Policy*),
- funkcija vrijednosti ili (korisnost) za predviđanje budućih nagrada (engl. *Value function*),
- funkcija prijelaza (engl. *Transition function*),
- funkcija nagrade (engl. *Reward*),
- model okoline koji reagira na poduzete akcije (engl. *Model*).

Na slici 2 prikazan je primjer jedne pojednostavljene okoline koja predstavlja 2D labirint u kojem se nalazi agent. Agent ima zadatak doći do cilja, te s vremenom minimizirati negativne kumulativne nagrade (mora naučiti najkraći put do cilja neovisno u koje polje ga se postavi). Na ovom primjeru su ilustrativno pojašnjeni prethodni pojmovi. Za svaki korak (akcija) agent trenutno dobiva nagradu u obliku negativnog broja bodova iznosa -1. Akcije predstavljaju smjerove kretanja agenta. One su ograničene na smjerove gore, dolje, lijevo i desno. Trenutna pozicija agenta predstavlja stanje okoline [11].



Slika 2. Prikaz jednostavnog primjera okoline 'labirinta' u kojem agent traži cilj [11]

Prema [10] „politika“ (engl. *Policy*) predstavlja naučenu funkciju koja predstavlja strategiju odabira akcija, može se reći da preslikava stanja u akcije. Izraz (1) predstavlja funkciju politike za deterministčke procese. Izraz (2) opisuje stohastičke procese odlučivanja poduzimanja pojedinih akcija na osnovu određene vjerojatnosti, što omogućuje istraživanje okoline:

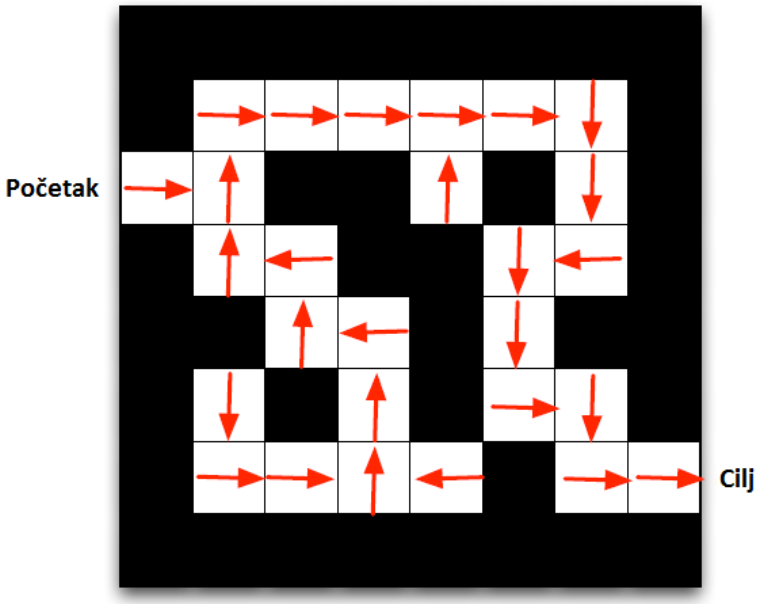
$$a = \pi(s), \tag{1}$$

$$\pi(a|s) = \mathbb{P}[A = a|S = s], \tag{2}$$

gdje su:

- A - označava skup akcija,
- a - je pojedina akcija,
- S - je skup stanja,
- s - je pojedino stanje, a
- π - je funkcija politike (strategija odabira akcija).

Na slici 3 dan je ilustrativan primjer optimalne funkcije (politike) $\pi(s)$ u obliku strelica. Strelice predstavljaju akcije a koje vode agenta iz bilo kojeg stanja s ravno prema cilju odnosno „najkraća ruta“ [11].



Slika 3. Strategija odabira akcija (funkcija politike $\pi(s)$ „crvene strelice“) [11]

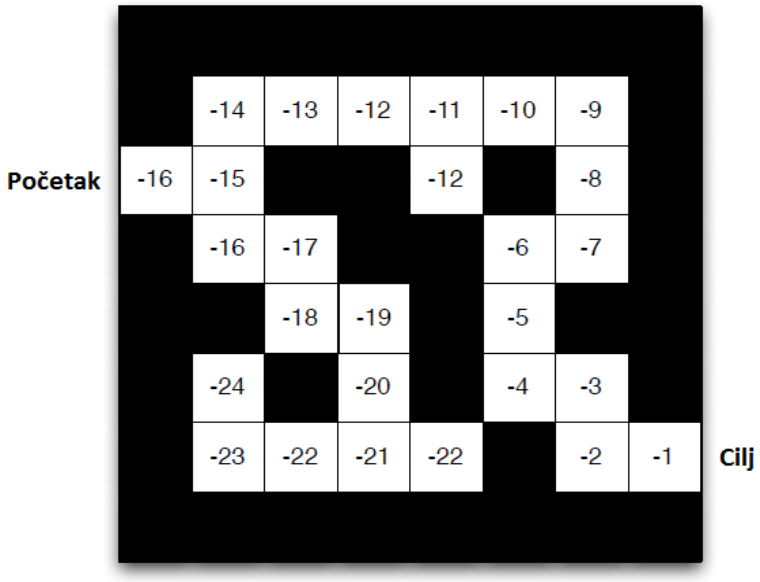
Izraz (3) predstavlja funkciju vrijednosti. Ta funkcija predstavlja očekivanu vrijednost budućih nagrada ako se agent pridržava određene politike. Na osnovu nje se određuje koliko je neko stanje dobro/loše, te se ovisno o tome odabiru potrebne akcije:

$$V_\pi(s) = E_\pi[R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \ldots | S_t = s], \quad (3)$$

gdje su:

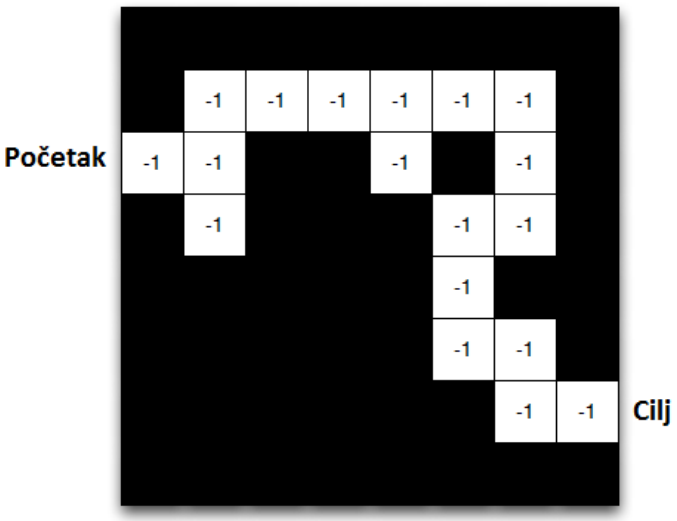
- R - nagrada, može biti pozitivna ili negativna,
- γ - faktor propadanja koji regulira utjecaj budućih nagrada, $\gamma \in [0, 1]$ (za vrijednosti oko 1 odgođene nagrade su jako bitne, dok je za $\gamma = 0$ bitna samo trenutna nagrada).

U skladu s idejom nagrađivanja, želi se doći do one strategije koja će rezultirati najvećom ukupnom nagradom. Tako počevši od početnog stanja s i koristeći strategiju π za odabir akcija, dobiva se ukupna vrijednost (korisnost) $V_\pi(s)$ [11]. Brojevi u poljima na slici 4 predstavljaju vrijednost funkcije korisnosti (3) za pojedina stanja s.



Slika 4. Funkcija vrijednosti $V_\pi(s)$ [11]

Model predstavlja apstrakciju kako agent zamišlja svoju okolinu. Tragovi rute „bijela polja“ na slici 5 tijekom jedne epizode predstavljaju ono što agent zamišlja da je njegova okolina. No kao što se vidi okolina je nepotpuna jer nedostaju polja iz donjeg lijevog ugla koji nije istražen. Da bi agent dobio potpunu sliku svoje okoline potrebno je dodatno istraživanje odabirom različitih kombinacija akcija. U potpornom učenju ovo je jedno od najbitnijih pitanja kako odrediti granicu do kada istraživati, a kada početi koristiti spoznato znanje o okolini i kako pojedina akcija djeluje na nju. Brojevi u poljima na slici predstavljaju trenutnu nagradu $\mathcal{R}_s^a$ za prelazak iz polja u polje. U ovom primjeru za svako preslikavanje stanje-akcija ta nagrada je ista [11].



Slika 5. Model okoline iz perspektive agenta u početku učenja [11]

Korištenjem prethodnih ilustracija sa slika 2 do 5 dana je predodžba o osnovnim pojmovima kod RL-a. Iz navedenog slijedi da je cilj učenja pronaći optimalnu politiku π^* tako da je [11]:

$$\pi^* \equiv argmax V^{\pi}(s), \forall s. \quad (4)$$

Odnosno cilj je pronaći onaj skup akcija koji za bilo koje stanje s daje maksimalnu vrijednost funkcije korisnosti V. Izraz (4) predstavlja slučaj koji kaže da je optimalna politika identična maksimalnoj vrijednosnoj funkcije (ako agent „robot" bira nova stanja optimalno) [11]. Ovakav pristup daje na izbor da li računati optimalnu funkciju korisnosti V^{π} ili do istog rješenja doći pomoću funkcije optimalne politike. Često se do istog rezultata dolazi puno brže računajući funkciju politike. Neke metode koriste oba pristupa simultano gdje se nakon izračunate vrijednosti V izračuna politika π te se na osnovu nje ponovno osvježava funkcija V i tako u krug ovisno koliko točno se želi aproksimirati optimalna politika π^* [10].

3.2. Metoda učenja s vremenskom razlikom u potpornom učenju

Klasa metoda učenja s vremenskom razlikom (engl. *Temporal Difference* - TD) predstavlja *model - free* učenje. Metoda objedinjuje principe *Monte Carlo* - MC metode i dinamičkog programiranja (engl. *Dynamic Programming* - DP). Ideja TD je da agent procjenjuje funkciju korisnosti $V(\pi)$ izravno iz svakog novog iskustva slijedeći politiku π. Razlika između MC i TD očituje se u tome da prva metoda treba čekati kraj epizode jednog učenja da formalizira određeno znanje, dok TD uči nakon svakog koraka tijekom učenja. Dakle, TD ima sposobnost učiti iz necjelovitih sljedova događaja. Ovo svojstvo omogućava primjenu učenja u kontinuiranoj okolini gdje epizoda učenja nikada ne prestaje. Da bi algoritam nešto naučio nije potrebno uzeti beskonačno mnogo sljedova događaja iz takve okoline, dovoljno je tek nekoliko događaja na osnovu kojih TD donosi zaključke. Formalno se proces učenja i traženja optimalne politike može predstaviti sustavom jednadžbi te je jedno od rješenja takvog sustava primjena alata iz linearne algebre. Ti sustavi jednadžbi mogu biti veliki ovisno o modelu i akcijama koje opisuju agenta i okolinu. Drugi pristup je iterativni postupak traženja nepoznate funkcije politike ili vrijednosne funkcije. Osnovni princip evaluacije funkcije politike zasniva se na *Bellman*-ovim rekurzivnim formulama koje omogućuju iterativno računanje nepoznatih vrijednosti funkcija umjesto rješavanja velikih sustava jednadžbi. U nastavku su opisani algoritmi zasnovani na TD-u metodi [7], [10], [11].

3.2.1. SARSA algoritam

SARSA algoritam predstavlja *On-Policy* algoritam učenja zasnovan na Q funkciji. Pošto učenje Q funkcije odgovara učenju idealne politike, time odgovara učenju i njene vrijednosne funkcije V^* „pogledaj (4)". Proces iterativnog učenja Q funkcije u ovom slučaju ovisi samo o trenutnom stanju u kojem se nalazi agent i o poduzetoj akciji. Te vrijednosti se bilježe i osvježavaju u takozvanoj Q-matrici. Slijedeći izraz opisuje iterativni proces kod SARSA algoritma [7], [11]:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\big(r + \gamma Q(s',a') - Q(s,a)\big), \quad (5)$$

gdje su:

- Q - takozvana Q-matrica,

- α - koeficijent učenja,
- r - predstavlja trenutnu nagradu,
- γ - faktor odgode nagrade,
- s' - je novo stanje i
- a' - predstavlja novu akcija iz stanja s'.

3.2.2. Algoritam *Q-učenja*

Algoritam *Q-učenja* predstavlja *off-policy* učenje. Konvergencija Q funkcije zasniva se na sljedećoj hipotezi. Ako agent obiđe svaki par stanja - akcija beskonačno puta, tada prema [7] algoritam *Q-učenja* konvergira prema optimalnoj funkciji „akcija-stanja“ (engl. *action-state*) (6).

$$Q(s,a) \rightarrow q_*(s,a) \tag{6}$$

Model okoline (MDP) kod ovog algoritma nije potrebno poznavati. Informacije o okolini agent dobiva preko nagrada ili kazni, u smislu povratne veze kao reakcije okoline na poduzete akcije. Te iste koristi prilikom ažuriranja vrijednosti Q-funkcije. Kod *Q-učenja* agent ima sposobnost učiti promatrajući druge agente kako izvode akcije. Također ima sposobnost ponovnog korištenja iskustva stečenih iz prijašnjih politika $\pi_1, \pi_2, \dots, \pi_{t-1}$ u svrhu još boljeg učenja. Algoritam *Q-učenja* ima sposobnost istraživanja i istovremeno učenja optimalne politike. Iterativni proces ažuriranja Q funkcije kod *Q-učenja*-a dan je izrazom [7], [11]:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\big(r + \gamma max_{a'} Q(s',a') - Q(s,a)\big). \tag{7}$$

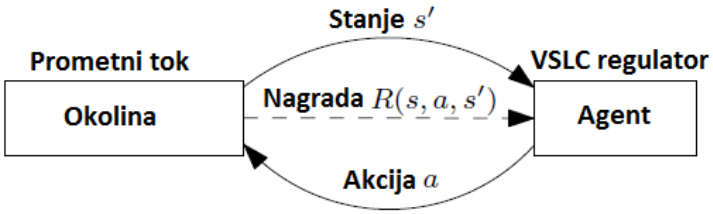
Algoritam *Q-učenja* nakon odabrane akcije a i preslikavanja na osnovu te akcije u novo stanje $s \rightarrow s'$, gleda još korak unaprijed s ciljem da odabere odgovarajuću novu akciju a' koja daje najveću nagradu za taj novi korak. Na osnovu ta dva koraka ažurira se vrijednost Q funkcije za stanje s i akciju a. No algoritam ne mora uvijek slijediti optimalnu politiku 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. Dok s druge strane SARSA algoritam iz stanja s' odbire akciju a' te ažurira $Q(s,a)$. U novoj iteraciji slijedi tu istu akciju a' zato se još naziva i *on-policy* algoritam [10].

Kod potpornog učenja, postoji problem kako uravnotežiti istraživanje i iskorištavanje naučenog znanja (engl. *exploration - explotation dilemma*). Istraživanje je korisno za otkrivanje novih stanja (prostora) u okolini koja prije nisu bila poznata agentu. Istraživanje je moguće postići slučajnim odabirom akcija. S druge strane postojeće znanje također može biti iskorišteno za dobivanje većih nagrada. Važno je uravnotežiti istraživanje i eksploataciju s ciljem izbjegavanja lokalnih optimalnih rješenja, a s druge strane maksimalno iskoristiti postojeće znanje. Jednostavniji pristup rješenja ovog problema je takozvani ϵ-pohlepni (engl. ϵ - *greedy*) algoritam pretraživanja prostora. Postoje i bolja rješenja koja se temelje na istraživanju prostora čija vrijednost nagrada nije još sa sigurnošću ustanovljena tj. prostor nije dovoljno puta pretražen te se ne može sa sigurnošću reći da li je to stanje dobro ili loše. Jedan od poznatijih algoritama koji se bavi ovom problematikom je tako zvani gornji stupanj povjerenja (engl. *upper confidence bounds* - UCB). Više o problemu pretraživanja prostora i postavkama ε-

pohlepnog mehanizma implementiranog u algoritmu VSLC-a slijedi u idućem poglavlju [11], [10].

3.3. Upravljanje promjenjivim ograničenjem brzina zasnovano na algoritmu *Q-učenja*

U današnje vrijeme algoritmi sustava promjenjivog ograničenja brzina na urbanim autocestama se mogu zasnovati i na strojnom učenju. Regulator koji upravlja ograničenjem brzine može se poistovjetiti s agentom u RL-u. Prometni tok i dionica urbane autoceste predstavlja model okoline u kojoj agent djeluje. Osnovne veličine koje opisuju prometni tok predstavljaju stanja. Akcije su kodirane kao izvršne brzine ograničenja koje agent šalje na VMS-ove. Za svako stanje okoline agent može odabrati pojedinu akciju iz konačnog skupa akcija. Ovisno o odabranoj akciji okolina se mijenja te agent (regulator) dolazi u novo stanje. Ovisno o tom preslikavanju $s \rightarrow s'$ agent dobiva povratnu informaciju u obliku nagrade/kazne koliko su dobre njegove poduzete akcije. Ovaj proces je prikazan na slici 6 [7].



Slika 6. Povratna veza akcija i stanja okoline [7]

U nastavku će biti opisana implementacija regulatora za VSLC koji je zasnovan na *Q-učenju*. Rad VSLC regulatora zasnovanog na *Q-učenju* zasniva se na konstantnom uzorkovanju okoline (parametri prometnog toka na dionici autoceste) te donošenja odluka koju akciju primijeniti nakon kojih regulator dobiva ocjenu (nagradu/kaznu) kao mjeru uspješnosti poduzetih akcija. Podaci o prometnom toku prikupljaju se uzastopno u vremenu uzorkovanja od $T = 30$ sekundi. VSLC mijenja brzine svakih T_c sekundi (u ovom radu je uzeta vrijednost od 300 sekundi) prema izrazu:

$$T_c = cT, \tag{8}$$

gdje su:

- T - vrijeme uzorkovanja podataka,
- c - konstanta koja određuje duljinu intervala uzrokovanja regulatora ($c \in \mathbb{N}$).

Iz kojih se dalje izračuna upravljački interval regulatora:

$$T_c = 10 * 30\ [s] = 300\ [s] = 5\ [min].$$

Stanje prometa na autocesti dobro opisuje gustoća toka $\rho\left[\frac{veh}{km}\right]$. Dodatni atribut u skupu koji sadržava stanja je ograničenje brzine $v_{VSLC}\left[\frac{km}{h}\right]$ u ćeliji s implementiranim sustavom VSLC. Sljedeći izraz predstavlja konačni skup stanja koja opisuju okolinu:

$$S = \{\rho_i^t, \rho_{i+1}^t, \rho_{i+2}^t, VSLC_i^{t-1}\}, \quad (9)$$

gdje su:

- S - konačan skup stanja koja opisuju okolinu agenta (regulatora),
- ρ_i^t - predstavlja gustoću prometnog toka u ćeliji i za trenutak t,
- $VSLC_i^{t-1}$ - je ograničenje brzine iz prijašnjeg upravljačkog vremenskog interval $(t-1)$ za ćeliju i.

U slučaju gustoća ρ vrijednosti koje varijabla može poprimiti kodirane su kako slijedi:

- $\rho < 10 \rightarrow 1$,
- $10 \leq \rho < 15 \rightarrow 2$,
- $15 \leq \rho < 22 \rightarrow 3$,
- $22 \leq \rho < 30 \rightarrow 4$,
- $30 \leq \rho \rightarrow 5$.

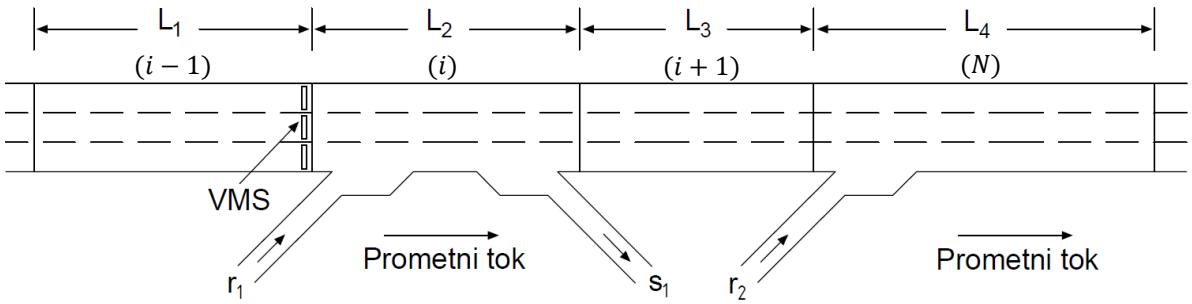
Isto je sa vrijednostima za stanja ograničenja brzine trenutak prije, koje su kodirane kako slijedi:

- $130 \rightarrow 1$,
- $110 \rightarrow 2$,
- $100 \rightarrow 3$,
- $80 \rightarrow 4$,
- $60 \rightarrow 5$.

Skup akcija koje regulator (agent) ima na raspolaganju poduzimati predstavljen je konačnim skupom (10) koji sadrži ograničenja brzine. Nakon što regulator odabere ograničenje ono se šalje na promjenjive prometne znakove VMS na autocesti na početku ćelije L_2.

$$A = \{130, 110, 100, 80, 60\} \quad (10)$$

Akcije su također kodirane redom od 1 do 5 prema redoslijedu navođenja iznosa ograničenja brzina u izrazu (10). Slika 7 prikazuje prikaz modela apstraktne urbane autoceste podijeljene na ćelije [1] koja će se primijeniti za simulaciju. Može se vidjeti koje ćelije opisuju stanja i u kojoj se primjenjuju akcije (VSLC algoritam).



Slika 7. Model urbane autoceste podijeljen u četiri ćelije

Odabrane akcije agenta dovode do promjene stanja okoline. Agent nakon svake odabrane akcije a u stanju s posljedično završava u novom stanju s'. Na osnovu tog preslikavanja $s \rightarrow s'$ agent dobiva nagradu. Nagrada ne mora uvijek biti pozitivna. Ona može biti i kazna u slučaju ako agent poduzima loše akcije. U *Q-učenju* ona predstavlja povratnu informaciju iz okoline koja agentu pomaže da na svojevrstan način stvori predodžbu (sliku) o okolini u kojoj se nalazi.

Slijedeći jednostavni primjer pojašnjava ulogu povratne veze u obliku nagrada kod *Q-učenja*, na primjeru robota koji uči prijeći most bez zaštitne ograde. Robot ima ograničene kretnje ravno, lijevo i desno. Nitko mu prije nije rekao kako izgleda most i da se s njega može pasti u rijeku. Za svaki korak dobije nagradu (npr. kaznu iznosa -1), a ako završi u rijeci (-10). Nakon većeg broja pokušaja saznaje da nešto nije u redu ako ide previše u lijevo ili desno, upravo preko akumuliranih nagrada u Q-tablici (matrici). Iako nije imao znanja kako izgleda njegova okolina, polako sa svakim pokušajem prelaska mosta on ju formulira (apstraktno) upravo preko povratne veze u obliku nagrada koje dobiva za poduzete akcije. Sa svakim novim pokušajem dobiva bolju sliku o tome gdje se nalazi i kuda ići. U slučaju VSLC-a princip je isti samo što je problem puno složeniji jer se okolina konstantno mijenja (kod robota okolina je statična). Za opis okoline u kojoj radi regulator (u ovom slučaju za VSLC) broj varijabli koje opisuju stanja okoline i njihove kombinacije puno je veći.

Nagrada (u slučaju VSLC-a 'kazna ili nagrada') određuje se uz male preinake prema [7]:

$$r(t) = \begin{cases} -\delta & (VSLC_i^{t-2} = VSLC_i^t) \wedge (VSLC_i^{t-1} \neq VSLC_i^t) \\ -\delta & \left|VSLC_i^{t-1} - VSLC_i^t\right| > 20 \\ 0 & 105 < \min\{\bar{V}_i^{t+1} \,|\, i = 1, \dots, N\} < 110 \\ \delta & \min\{\bar{V}_i^{t+1} \,|\, i = 1, \dots, N\} \geq 110 \\ -TTS(t) & inače \end{cases}, \tag{11}$$

gdje su:

- r - funkcija koja vraća nagradu,
- δ - konstanta za prva tri uvjeta u (11),
- $\bar{V}_i^{t+1}$ - srednja brzina toka vozila za ćeliju i u trenutku $(t+1)$,
- TTS - ukupno provedeno vrijeme vozila na dijelu autoceste (obuhvaćene ćelije L_2, L_3, L_4). Mjereno unutar upravljačkog intervala (VSLC akcija - pet minutni intervali).

Prvi uvjet u izrazu (11) kažnjava akcije regulatora koje uzrokuju oscilirajuće promjene ograničenja brzine. Drugi uvjet kažnjava promjenu brzine ograničenja kada je razlika između dva uzastopna upravljačka intervala regulatora veća od 20 $\left[\frac{km}{h}\right]$. Ovime se sprječavaju nagle značajne promjene iznosa ograničenja brzine. Cilj je postepeno smanjivati ili povećavati ograničenja brzine. Treći uvjet predstavlja slučaj u kojem je u tri obuhvaćene ćelije (L_2, L_3, L_4) prosječna izmjerena brzina u trenutku $(t+1)$ veća od 105 $\left[\frac{km}{h}\right]$. Ta činjenica dovodi do zaključka da na promatranoj dionici autoceste ne postoji zagušenje. U tome slučaju regulator (VSLC) radi dobru stvar tj. odabire dobre akcije, te na osnovu toga dobiva najmanju kaznu (0). Ovaj uvjet može biti ispunjen samo ako regulator drži ograničenje brzine u ćeliji L_2 iznad 100 $\left[\frac{km}{h}\right]$. Kako bi se spriječilo konvergiranje izlaznih brzina na vrijednost 110 $\left[\frac{km}{h}\right]$ u slučaju slobodnog toka, dodan je četvrti uvjet. Četvrti uvjet je sličan kao treći samo što je pretpostavka da nema zagušenja još jača, a uvjet postignutih prosječnih brzina od 110 $\left[\frac{km}{h}\right]$ u sve tri obuhvaćene ćelije lakše ispunjen ako je ograničenje brzine iznosa 130 $\left[\frac{km}{h}\right]$.U svim ostalim situacijama regulator dobiva kaznu proporcionalnu izmjerenom ukupnom vremenu koja su sva vozila provela u prometnoj mreži (engl. *total time spent* - TTS). TTS se računa prema slijedećoj formuli [7]:

$$TTS(t) = T \sum_{p=ct}^{(t+1)c} \sum_{j=i}^{N} M_i L_i \rho_i(p)\,, \tag{12}$$

gdje su:

- M_i - broj prometnih traka u ćeliji i,
- L_i - duljina ćelije i,
- ρ_i - gustoća prometnog toka u ćeliji i.

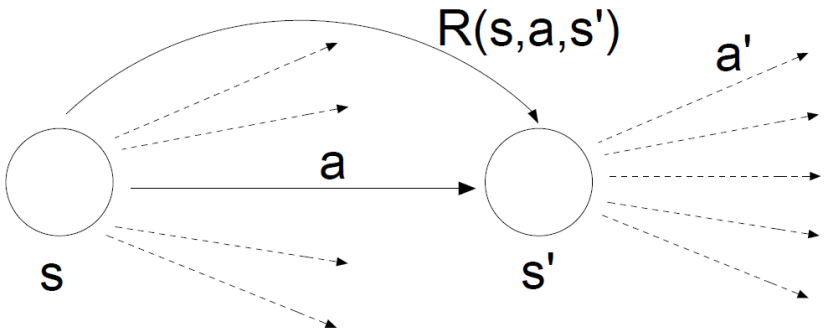
TTS kao nagrada predstavlja izvrsnu kvantitativnu mjeru kažnjavanja iz dva razloga. Ako regulator drži veće iznose brzina ograničenja u trenutku zagušenja, nalet tog brzog toka vozila prouzročit će brže širenje zagušenja unazad po autocesti (šok val), samim time i veći TTS. S druge strane ako bi regulator stalno držao ograničenja brzine na najnižim vrijednostima, poštujući ograničenja vozila bi se kretala sporije, čime se povećava vrijeme putovanja za isti prevaljeni put. To opet znači povećanje TTS vrijednosti. Ova proporcionalna mjera kažnjavanja (TTS) daje mogućnost regulatoru da optimalno odabire akcije s ciljem minimiziranja vrijednosti TTS-a kada je prometno opterećenje povećano. Konstanta δ prema [7] mora biti odabrana na način da je uvijek veća od najveće izmjerene vrijednosti TTS-a unutar upravljačkog intervala regulatora ($T_c = 300\,[s]$). U radu je konstanta δ postavljena na vrijednost dvostruko veću od najvećeg izmjerenog iznosa TTS-a.

Ideja algoritma VSLC-a zasnovanog na *Q-učenju* sastoji se u tome da se konstantno ažuriraju Q vrijednosti za pojedini par stanja-akcija (s,a). S vremenom regulator konvergira prema optimalnoj VSLC politici akumulirajući nagrade u Q-matricu. Izraz (13) ispod pojašnjava ovaj postupak:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma * max_{a'} Q(s',a') - Q(a,s)]. \quad (13)$$

(nova Q vrijednost = stara Q vrijednost) + α(nova informacija – stara informacija)

Na slici 8 shematski je prikazan jedan iterativan postupak u *Q-učenju*.



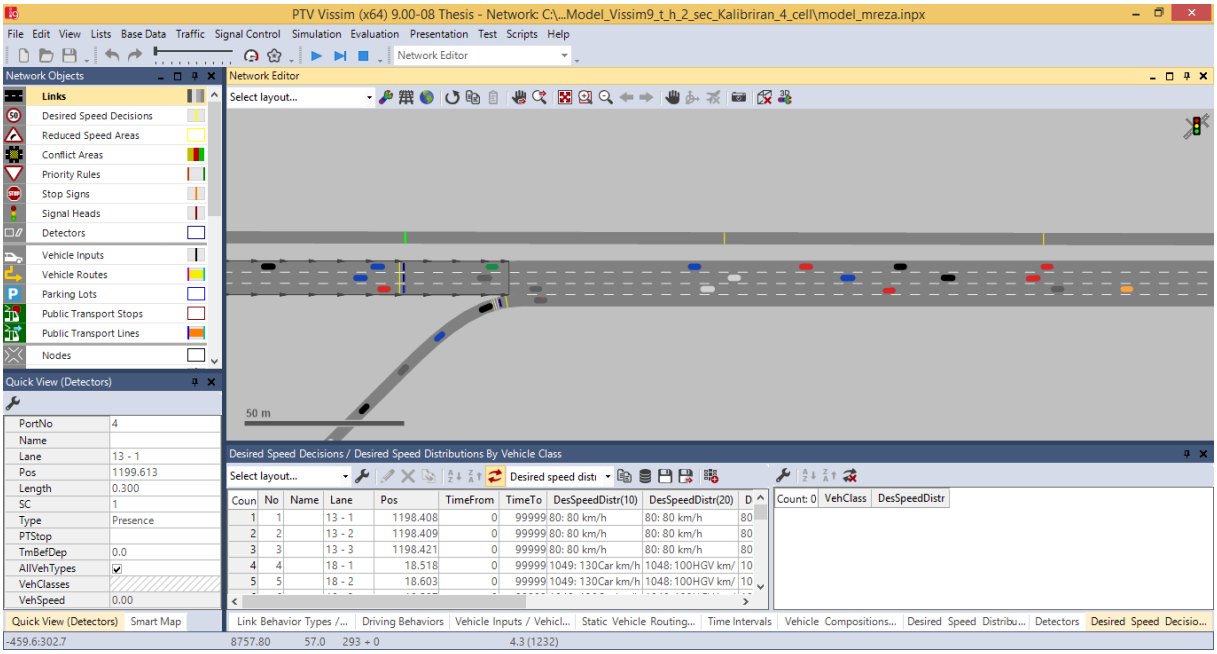
Slika 8. Shematski prikaz preslikavanja $s \rightarrow s'$

Neka je agent (VSLC) u stanju s i odabrao je akciju a (npr. pomoću ϵ-*greedy* algoritma). Poduzeta akcija a vodi ga u novo stanje s'. Ovisno o tom preslikavanju agent dobiva nagradu. Ako odabere lošu akciju dobiva veću kaznu. Na kraju kada se nalazi u novom stanju s' mora odabrati novu akciju a' koja mu garantira najveći iznos iz Q-matrice. Na kraju ovog postupka ažurira se Q-matrica na poziciji $Q(a,s)$. Nakon toga postupak kreće ispočetka gdje je sada $s \leftarrow s'$.

Na red dolazi dio koji pojašnjava na koji način agent odabire akcije. Kod *Q-učenja* veliko pitanje predstavlja problem istraživanja, odnosno iskorištavanja naučenog znanja. Algoritam *Q-učenja* u početku ne zna ništa o svojoj okolini, mora ju istražiti. To se postiže slučajnim odabirom akcija. Jednostavniji pristup u rješavanju ovog problema postiže se korištenjem ϵ-*pohlepnog* algoritma. Algoritam se zasniva na uniformnoj distribuciji vjerojatnosti odabira akcija. Stohastički odabir akcija kontrolira se pomoću parametra ε koji poprima vrijednosti $0 < \epsilon < 1$. Za male vrijednosti parametra, agent će koristiti stečeno znanje kako bi odabrao najbolje poznate akcije s ciljem maksimiziranja nagrade (iskorištavanje). Obrnuto kada $\epsilon \rightarrow 1$ agent odabire akcije slučajnim odabirom (istražuje) [10], [11]. U praksi se pojavljuje problem odabira najbolje vrijednosti ϵ. Ako agent ne istraži dovoljno svoj prostor može zaglaviti u lokalnom minimumu smatrajući da je naučio dovoljno, a može još pronaći puno bolje rješenje odnosno doći do globalnog optimuma. Tijekom testiranja VSLC algoritma zasnovanog na *Q-učenju* korišten je veći broj simulacija, te je ϵ postepeno smanjivan u svakoj od njih počevši od 1 prema 0.

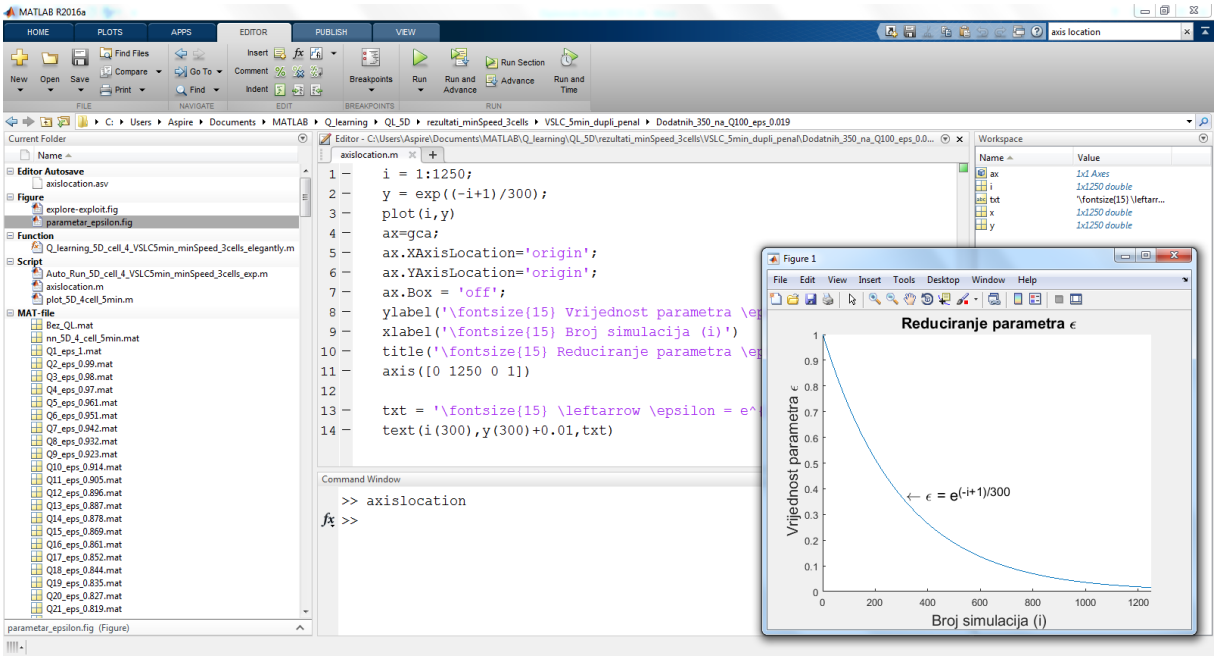
4. Značajke mikroskopskog simulatora VISSIM i programskog paketa MATLAB

Simulacija prometa odvija se u cestovnom mikroskopskom simulacijskom programu VISSIM tvrtke PTV. Općenito postoje mikroskopske i makroskopske simulacije, te hibridi obje spomenute. U mikroskopskoj simulaciji se svaki entitet (automobil, kamion, vlak, osoba, itd.) simulira na individualnoj razini, tj. predstavljen je zasebnim entitetom unutar simulacije. Jednako vrijedi i za međusobnu interakciju između entiteta. Makroskopska simulacija promatra prometni tok, zbog dovoljno poopćene razine promatranja, kao da je fluid i ponaša se po pravilima mehanike fluida. VISSIM je alat koji se zasniva na mikroskopskom modelu simulacije. Osnova ovog simulatora je Wiedemann-ov model koji opisuje interakciju između prometnih entiteta (vozila) na osnovu psihofizičkih reakcija vozača (individualna percepcija vozača o brzini vozila koja se kreču ususret, percepcija udaljenosti od drugih vozila, reakcije na zaustavljeno vozilo, „blaga/nagla“ usporenja (koćenja) i ubrzanja, sigurnosni razmak između vozila itd.) [13]. Postoje dva osnovna modela Wiedemann 77 i 99. U ovom radu korišten je drugi zbog toga što je prikladniji za opis cesta izvan naselja i autocesta. Dakle, mikroskopski modeli opisuju ponašanje jednog vozila u toku, uz pretpostavku da se takvo ponašanje može primijeniti na sva ostala vozila. Zbog toga se koriste parametri koji opisuju kretanje na razini svakog tipa vozila, a neki od njih su: distribucija brzine pojedinog tipa vozila, interval slijeđenja, razmak vozila, ubrzanje i usporenje, model prestrojavanja itd.. Sučelje simulacijskog alata VISSIM prikazano na slici 9 je grafički pregledno i jednostavno za korištenje. Sam alat nudi mogućnost primjene za testiranje različitih inačica rješenja ne samo unutar simulatora nego i putem povezivanja s drugim aplikacijama napravljenih pomoću MATLAB, C#, JAVA, C++ itd. Ovo se prvenstveno odnosi na testiranje različitih upravljačkih rješenja implementiranih u obliku programskog kôda u jednom od viših programskih jezika. Uz sam program VISSIM moguće je instalirati programski dodatak EnViVer koji korištenjem simulacijskih podataka iz VISSIM-a (brzine, akceleracije, tipovi vozila itd.) izračunava emisije štetnih plinova (CO_2, NO_x) i čestica (PM10) [1], [13].



Slika 9. Korisničko sučelje mikroskopskog simulatora VISSIM

MATLAB je matematički programski paket koji nalazi primjenu u praktičnom inženjerstvu i znanosti, te intenzivnu primjenu nalazi i u industriji. Program je u samim početcima zamišljen kao potpora za rješavanje linearne algebre. Budući da je zamišljen i razvijen kao interpreterski programski jezik visoke razine, koji se u početku zasnivao na kompleksnoj matrici kao osnovnom tipu podatka, imenovan je kraticom od matrični laboratorij [14]. Gotova funkcijska rješenja unutar programskog paketa MATLAB za različita područja primjene kontinuirano se dorađuju i proširuju. Po svojoj formi blizak je načinu na koji se inače zapisuju matematičke formule, pa jedna linija u MATLAB-u može zamijeniti više linija kôda napisanih u nekom drugom višem programskom jeziku kao što je to npr. C++ ili Java. MATLAB je stoga programski jezik visoke učinkovitosti u tehničkom računanju. On objedinjuje računanje, vizualizaciju i programiranje u praktičnom i preglednom prozorskom okolišu ugodnom za korisnika kao što je prikazano na slici 10, gdje su problemi i rješenja izraženi uobičajenim elegantnim matematičkim zapisom [14].

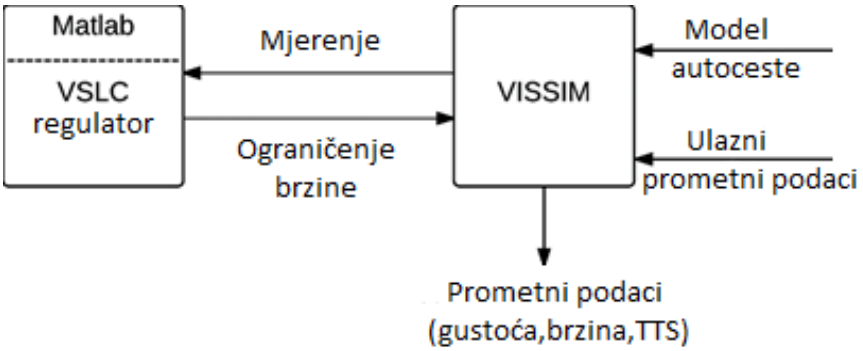


Slika 10. Korisničko sučelje programskog paketa MATLAB

Današnje karakteristike MATLAB-a daleko prelaze originalni "matrični laboratorij" [15]. Uz osnovni paket postoje i brojni dodatni programski alati (engl. *toolboxes*) koji pokrivaju gotovo sva područja inženjerske djelatnosti: obradu signala, slike, 2D i 3D grafičko oblikovanje, automatsko upravljanje, statističke obrade, neuronske mreže, neizrazitu logiku, financijsku matematiku, simboličku matematiku i mnogo drugih alata. Paket SIMULINK je dodatak MATLAB-u koji omogućuje simulaciju različitih „modela" sustava pomoću funkcijskih blok dijagrama i dijagrama stanja. MATLAB je otvoren sustav u kojem korisnik može graditi svoje vlastite alate (funkcije) unutar takozvanih MATLAB skripti. Omogućeno je preuzimanje novih i modificiranje postojećih, jer su dostupne u obliku izvornog kôda „on-line". Složeniji programi mogu biti programirani u kraćem vremenu, za razliku od vremena koje je potrebno kod "klasičnih" programskih jezika. Na taj se način omogućuje brža provjera različitih inačica pojedinog algoritama prije implementacije u realni sustav gdje izvršni kôd zbog zahtijeva sustava mora biti u nekom drugom programskom jeziku [14], [15].

5. Povezivanje VISSIM-a i MATLAB-a

U svrhu testiranja upravljačkog algoritma za VSLC zasnovanog na *Q-učenju* napravljen je model urbane autoceste sa četiri ćelije unutar mikroskopskog simulatora VISSIM. Upravljački algoritam implementiran je unutar programskog paketa MATLAB. Oba alata povezana su preko COM sučelja, čime im je omogućena komunikacija. Tim načinom se može upravljati prometnom simulacijom pomoću programskog kôda napisanog u programskom paketu MATLAB. Simulacijom se u potpunosti upravlja pomoću MATLAB-a. Podaci o prometnom toku (brzina, gustoća, protok) prikupljaju se i računaju pomoću funkcija kreiranih u programskom kôdu u MATLAB-u koje pristupaju postavljenim senzorima u VISSIM-u. Trenutne odluke o ograničenju brzine koje predlaže algoritam VSLC-a, izravno se šalju u VISSIM na pripadne VMS-ove. Na slici 11 prikazana je blokovska shema simulacijskog okruženja za testiranje VSLC-a zasnovanog na *Q-učenju*.

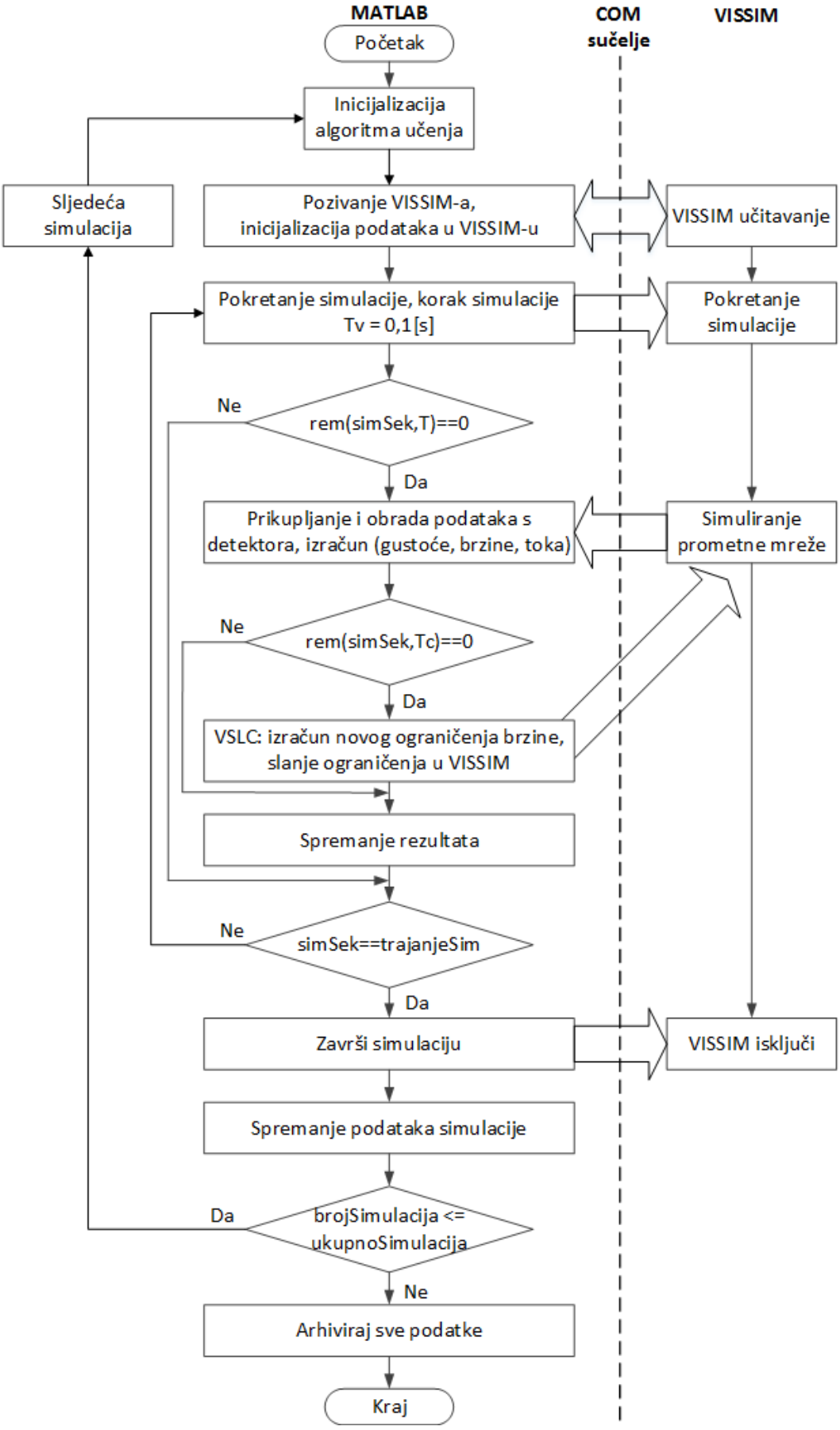


Slika 11. VISSIM-MATLAB simulacijski okvir [3]

Na slici 12 prikazan je dijagrama toka izvršavanja simulacija. Dijagram toka prikazuje interaktivni proces između programskog paketa MATLAB i mikroskopskog simulatora VISSIM. Proces počinje pokretanjem MATLAB skripte. Skripta prvo uspostavlja COM sučelje između oba programa te se učitava VISSIM model (*inpx* i *layx* datoteke). Prva sadrži podatke o parametrima mikrosimulacijskog modela kao što su poveznice (engl. *links*) 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 što se sve učita otvara se sučelje simulatora VISSIM. Idući korak definira ulazne varijable poput trajanja simulacije, broja simulacija, sjemena simulacije (engl. *simulation seed*), te parametara T i T_c. Pomoću različitih ulaznih vrijednosti parametra sjemena za generiranje vozila se za svaku simulaciju postiže efekt stohastičkog ponašanja prometnog toka tj. sve se razlikuju po distribucijama tokova zavisno o ulaznoj vrijednosti navedenog parametra [13]. Osim toga potrebna je i inicijalizacija ulaznih parametara algoritma učenja (ε, α, γ, ulazna Q-matrica itd.). S izvršavanjem slijedećih naredbi u MATLAB 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 simulatora. Jedna iteracija navedene petlje odgovara trajanju koraka simulacije T_v u iznosu od 0,1 $[s]$. Ovakvom izvedbom i brzinom kojom se petlja izvršava dobiva

se dojam da je simulacija kontinuirana (ne primjećuju se ti koraci „rezolucija simulacije“). Također unutar svakog tog koraka detektori u VISSIM-u reagiraju na podražaj ukoliko se na njima nalazi vozilo. Tako dobiveni „impulsi“ kontinuirano se pohranjuju unutar MATLAB-a u za to predviđenim matricama, ovisno o tipu podatka prikupljenom s detektora (prisutnost vozila, brzina vozila). Nakon svakih 30 $[s]$ (parametar T) pomoću gotove MATLAB funkcije *rem* (engl. *remainder after division*) provjerava se ostatak dijeljenja trenutne simulacijske sekunde i parametra T. Funkcija *rem* vrača ostatak dijeljenja dva broja te se pomoću nje detektiraju definirani vremenski trenuci predviđeni za proračun parametara toka i ograničenje brzine. Ako je ostatak dijeljena jednak nuli ulazi se u dio naredbi unutar kôda koji dohvaća akumulirane podatke pomoću funkcija kreiranih u MATLAB skripti koje izračunavaju gustoću, srednju brzinu i protok vozila za svaku ćeliju. Ove varijable predstavljaju osnovu za ocjenu kvalitete i sam rad VSLC algoritma. One se pohranjuju u matricu rezultata. Nakon svakih 300 $[s]$ (parametar T_c) kada je ostatak dijeljenja simulacijske sekunde i parametra T_c jednak nuli aktivira se sustav VSLC-a zasnovan na *Q-učenju* koji je također implementiran u MATLAB skripti. VSLC dohvaća prethodno izračunate vrijednosti varijabli iz matrice rezultata te na osnovu njih i onoga što je naučio dosad izračuna novu vrijednost ograničenja brzine. Nakon toga također se sve zabilježi u matricu rezultata, te se istodobno ažurira Q-matrica algoritma učenja. Proces se ponavlja tijekom cijele simulacije. Nakon što se petlja završi zaustavlja se simulacija te se prikupljeni (izračunati) podaci varijable toka i odluke VSLC-a pohranjuju unutar MATLAB mape. Proces kreće ispočetka pokretanjem nove simulacije inicijalizacijom podataka (mijenjaju se parametri ε i sjemena te se učitava prethodno popunjena Q-matrica) i nastavlja se do zadanog ukupnog broja simulacija (u ovom radu napravljeno je 1.250 simulacija).

Kod algoritma *Q-učenja* važno je naglasiti da je kod svake nove simulacije potrebno učitati prethodno popunjenu Q-matricu iz prethodne simulacije. Postupak ovog učitavanja Q-matrice implementiran je unutar MATLAB skripte te se izvršava automatski prije pokretanja nove simulacije. Potrebno je pristupiti prethodno pohranjenoj Q-matrici unutar MATLAB mape nakon čega se ista učita u radnu okolinu MATLAB-a (engl. *workspace*). Nakon toga se inicijalizira kao nova ulazna Q-matrica za novu simulaciju. Time se ostvaruje prijenos znanja (onoga što je algoritam VSLC-a zasnovan na potpornom učenju do sada naučio) u novu simulaciju. Ovaj proces je neophodan kod algoritama zasnovanih na učenju, zbog samog procesa učenja. Osim toga kako je kod učenja VSLC-a korišten ε- pohlepni pristup odabira akcija potrebno je tijekom svakog novog pokretanja simulacije umanjiti parametar ε za određeni iznos (kao što je prikazano na grafikonu 2). Više o detaljima o ovome procesu je dano u ostalim dijelovima ovog poglavlja.



Slika 12. Dijagram toka izvršne simulacije

Prethodno pojašnjen dijagram toka simulacijskog procesa unutar simulacijskog okruženja može se prikazati kao pseudokôd m-funkcije implementirane unutar MATLAB skripte.

Struktura m-funkcije:

Inicijalizacija varijabli; Ponavljaj dok (broj simulacije <= ukupan broj simulacija)
    Q-matrica
    Parametri: ε, α, γ
Pokretanje VISSIM-a
    COM sučelje
    Učitavanje modela
    Definiranje parametara simulacije: trajanje simulacije, sjemena simulacije itd.
    Simuliranje; Ponavljaj dok (simulacijska sekunda <= trajanje simulacije)
        Dohvat mjerenja
        Obrada mjerenja
            Izvršavanje VSLC
                Dohvat obrađenih mjerenja
                Izračun novog ograničenja
                Slanje ograničenja u VISSIM
        Spremanje varijabli
    Zaustavljanje simulacije
Arhiviranje podataka

za svaki T_c | za svaki T | za svaki T_v

Kao što je već spomenuto simulacija unutar VISSIM-a pokretana je unutar for petlje u MATLAB-u. Period trajanja simulacije iznosi 9.000 $[s]$ odnosno 2,5 $[h]$. Simulacija unutar spomenute petlje tijekom jedne iteracije ima korak od 0,1 $[s]$, te je time omogućena precizna detekcija vozila na postavljenim detektorima. Tako prikupljeni podaci akumuliraju se unutar MATLAB-a te se svakih 30 $[s]$ na osnovu istih računaju potrebne varijable koje opisuju prometni tok (gustoća, brzina, protok).koje se spremaju u matricu rezultata.

Izračun gustoće toka zasniva se na razlici između broja ulaznih i izlaznih vozila u pojedinoj ćeliji podijeljen s duljinom ćelije. Toj vrijednosti pribroji se gustoća u ćeliji iz prethodnog intervala uzorkovanja. Iterativni postupak izračuna gustoće za i-tu ćeliju izgleda:

$$\rho_i^t = \rho_i^{t-1} + \left(\frac{N_{ulaz_i}^t - N_{izlaz_i}^t}{l_i} \right), \tag{14}$$

gdje su:

- ρ_i^t - trenutna gustoća u i-toj ćeliji,
- ρ_i^{t-1} - gustoća prije u i-toj ćeliji,
- $N_{ulaz_i}^t$ - broj ulaznih vozila u i-tu ćeliju,
- $N_{izlaz_i}^t$ - broj izlaznih vozila iz i-te ćelije,
- l_i - duljina i-te ćelije.

Izraz (14) predstavlja zadovoljavajuću točnost izračuna gustoće iz razloga što u VISSIM-u svi detektori rade bez „greške“. U stvarnom okruženju ovaj pristup ima nedostatke zbog stvarne greške detektora (senzora) koju nije moguće izbjeći. Nakon dužeg vremena akumulirana greška bi mogla biti prevelika te je potreban drugačiji pristup kod određivanja gustoće toka. No to ionako nije u fokusu ovog rada. Valja napomenuti da je još potrebno dobivenu vrijednost gustoće podijeliti s brojem prometnih traka unutar pojedine ćelije (ovisno o modelu autoceste).

Srednja brzina predstavlja kvantitativnu mjeru koja dobro opisuje stanje prometnog toka. U ovom radu izračun srednje brzine predstavlja aritmetičku sredinu prosječne brzine na ulazu u ćeliju te na izlazu iz iste:

$$\bar{v}_i^t = \left(\frac{\frac{\sum_n v_n}{n} + \frac{\sum_m v_m}{m}}{2} \right), \tag{15}$$

gdje su:

- $\bar{v}_i^t$ - srednja brzina u ćeliji i,
- $\frac{\sum_n v_n}{n}$ - suma brzina detektiranih vozila na detektorima na ulazu u i-tu ćeliju izmjerena unutar vremenskog intervala $\langle t-1, t \rangle$, te podijeljena s brojem detektiranih vozila n,
- $\frac{\sum_m v_m}{m}$ - suma brzina detektiranih vozila na detektorima na izlazu i-te ćelije izmjerena unutar vremenskog intervala $\langle t-1, t \rangle$, te podijeljena s brojem detektiranih vozila m,
- $\langle t-1, t \rangle$ - vremenski interval između svakog izračuna prometnih parametara iznosi 30 $[s]$.

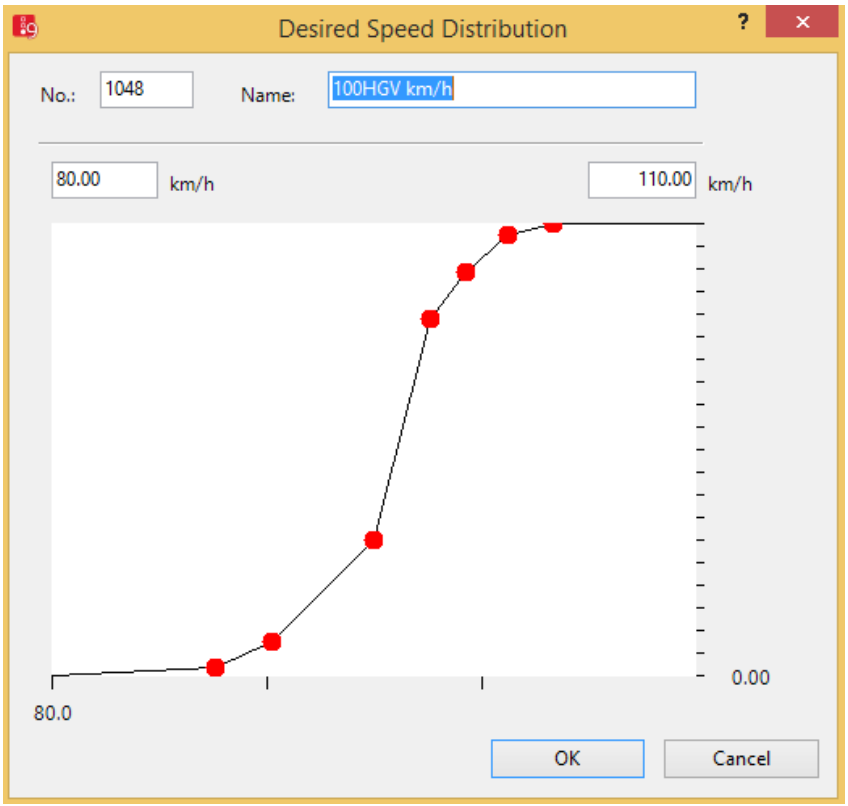
Izraz (15) ima jedan nedostatak. U slučaju da nema vozila na početku ćelije ili na kraju (unutar intervala $\langle t-1, t \rangle$) jedan od izraza u brojniku poprima vrijednost nula, te se sve dijeli s dva. Tako dobivena brzina ustvari je upola manja od stvarne. Da se spriječi ovakva situacija potrebno je dodatno provjeriti stanja na ulazu i izlazu te osigurati da se izbjegne dijeljenje nazivnikom u slučaju da je jedan od izraza u brojniku jednak nuli. Brzina bi mogla biti još preciznija ako bi se implementirala još jedna grupa detektora na sredini svake ćelije. Cilj rada je bio držati se stvarnih mogućnosti pri implementaciji sustava u realno okruženje u kojem uvijek postoje ograničenja u smislu broja senzora i ostale opreme (cijena samog sustava). Tako da je uvijek potrebno tražiti i ispitivati rješenja u simulatoru s realnim brojem senzorskih elemenata i druge opreme npr. broj VMS-ova. Inače približna brzina izračunata izrazom (15) sasvim dobro opisuje srednju brzinu toka vozila u ćeliji.

Izračun protoka na detektorima zasniva se na ukupnom broju detektiranih vozila na samom detektoru podijeljeno s vremenom trajanja uzorkovanja $\langle t-1, t \rangle$. Kako je to vrijeme kraće od jednoga sata potrebno je prilagoditi izraze kako bi se protok vozila prikazao u mjernoj jedinici $\left[\frac{voz}{h}\right]$.

Nakon svakog pet minutnog ciklusa regulator za VSLC pristupa podacima brzine i gustoće te na osnovu njih predlaže ograničenje brzine. Sam regulator ima svoj dio memorije unutar algoritma u MATLAB-u za spremanje naučenog upravljačkog zakona, tzv. Q-matricu.

U njoj je spremljena svo znanje (iskustvo) koje regulator stječe tijekom učenja. Djelujući izravno na simulacijsku okolinu (promjena prometnih parametara) odabirom određenih akcija (ograničenja brzine) VSLC ima sposobnost učenja što se manifestira boljim odabirom akcija.

Također je riješen problem poštivanja ograničenja kod teretnih vozila i autobusa u mikroskopskom simulatoru VISSIM. 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 tolikim brzinama. Iz tog razloga uvedene su dodatne provjere unutar izvršnog dijela kôda koji predlaže i postavlja nova ograničenja brzina na VMS-ove. Provjera se sastoji od ispitivanja iznosa izlazne brzine regulatora te u koliko je ona veća od 100 $\left[\frac{km}{h}\right]$ ignoriraju se teretna vozila i autobusi. Oni u tom slučaju voze najvećom dozvoljenom brzinom, koja se ravna po zadanoj distribuciji unutar VISSIM-a prikazanoj na slici 13. Distribucija brzine definirana je na način da najveći broj populacije (teretna vozila i autobusi) voze brzinom između 90 $\left[\frac{km}{h}\right]$ i 110$\left[\frac{km}{h}\right]$.



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

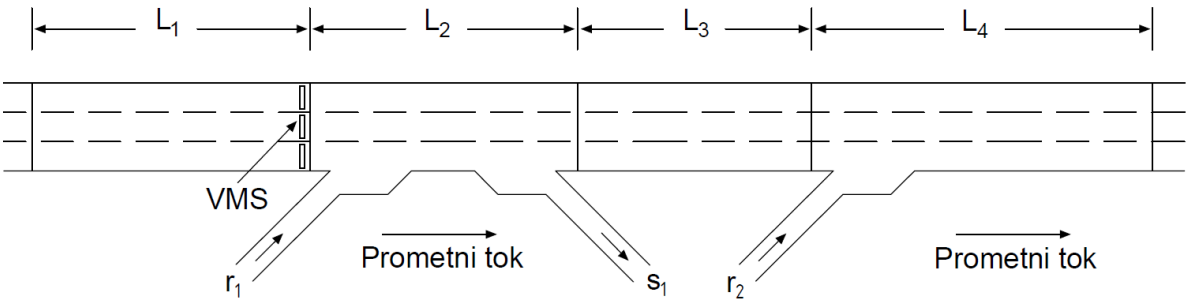
Nakon što simulacijska petlja izvrši sve iteracije (nakon 9.000 [s]) simulacija se prekida. Tako prikupljeni podaci (u matrici rezultata) analiziraju se i grafički prikazuju pomoću alata u MATLAB-u. Prikaz rezultata slijedi u idućem poglavlju.

6. Simulacijski rezultati

U ovom poglavlju slijedi analiza i prikaz dobivenih rezultata simulacije prometnog modela urbane autoceste. Rezultati opisuju karakteristične parametre prometnog toka na osnovu kojih je ocijenjen rad VSLC zasnovanog na *Q-učenju*. Osim parametara prometnog toka prikazane su izlazna ograničenja brzine koje izračunava VSLC implementiran u ćeliji L_2 te su korištenjem zavisnosti prometnog toka o ograničenjima brzine izneseni određeni zaključci o radu VSLC sustava.

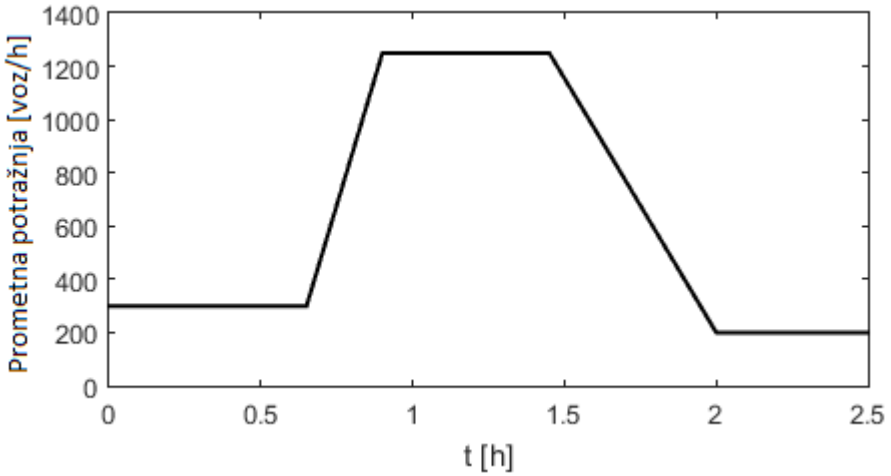
6.1. Model i podaci

Model urbane autoceste s tri prometne trake prikazan na slici 14 napravljen je u mikroskopskom simulatoru VISSIM, dok je upravljačka logika VSLC-a zasnovana na *Q-učenju* implementirana u programskom paketu MATLAB. Model je podijeljen na četiri ćelije. Podjela autoceste na ćelije omogućuje parcijalnu evaluaciju stanja pojedinih dijelova autoceste kao i djelovanje VSLC-a na određenu ćeliju ili više njih. Duljine ćelija redom iznose 1, 0,8, 0,7 i 2 $[km]$. U ćeliji L_2 nalazi se prva ulazna rampa r_1 i izlazna rampa s_1, dok se u ćeliji L_4 nalazi druga ulazna rampa r_2. Model i ulazni podaci tokova su zasnovani na modelu iz [1] uz male preinake u odnosu na original koje se očituju u zamjeni duljina druge i treće ćelije, čime su obje rampe (r_1 i s_1) smještene u drugoj ćeliji. Protok vozila na r_1i glavnom toku u prosjeku je povećan za 10 % u odnosu na model iz [1].



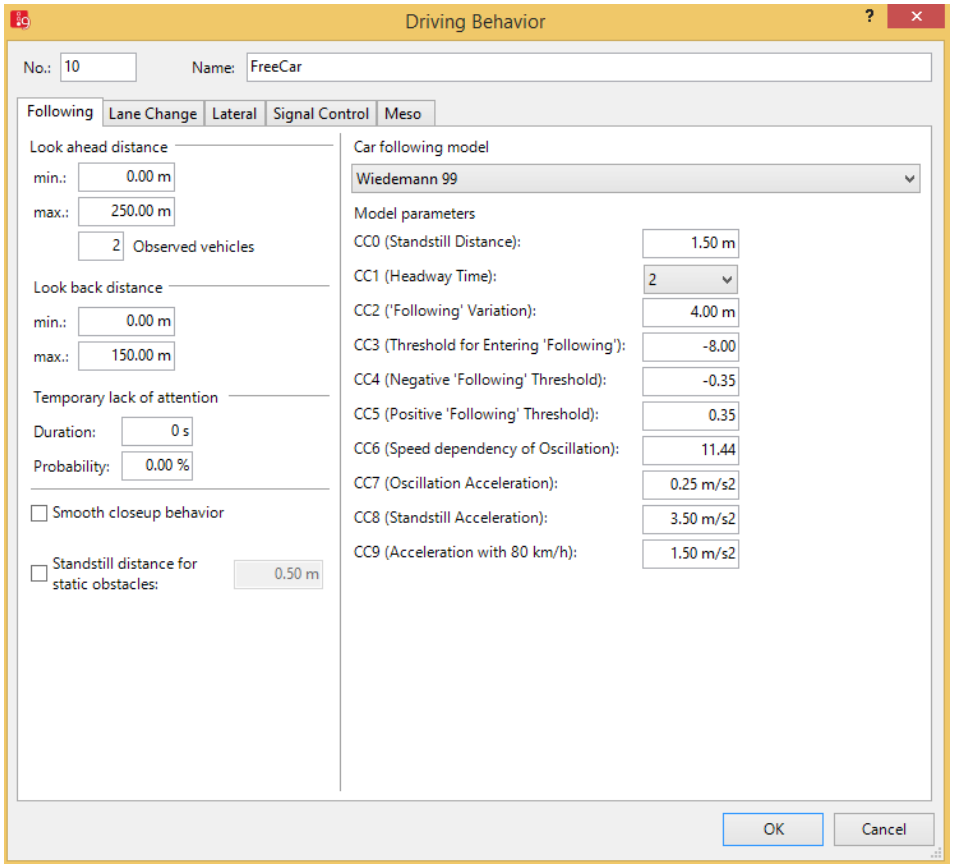
Slika 14. Model urbane autoceste podijeljen u četiri ćelije L_1, L_2, L_3 i L_4

VMS-ovi za prikaz promjenjivog ograničenja brzine nalaze se na početku ćelije L_2 te se pomoću njih obavještavaju vozači kolika je dozvoljena brzina kretanja u ćeliji L_2. Prometni tok u slučaju simulacijskog testiranja u ovom radu čine automobili (96 %), teretna vozila (2 %) i autobusi (2 %). Glavni tok ima konstantnu prometnu potražnju od 4.500 $\left[\frac{voz}{h}\right]$ od toga 95 % prometa ostaje na glavnom toku dok ostali silaze na izlaznoj rampi s_1. Isto je i s prvom ulaznom rampom gdje je prometni tok konstantan i iznosi 1.350 $\left[\frac{voz}{h}\right]$. Ulazni tok vozila na drugoj ulaznoj rampi mijenja se tijekom simulacije prema funkciji koja je prikazana na grafikonu 1.

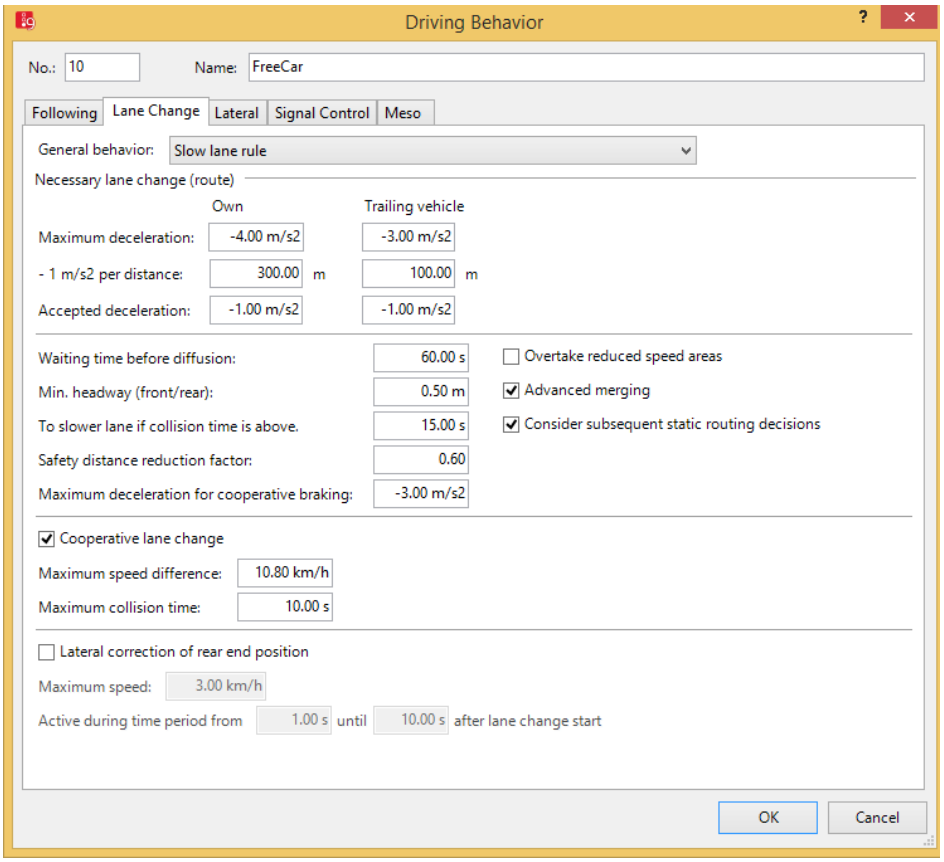


Grafikon 1. Generirana prometna potražnja na ulaznoj rampi r_2

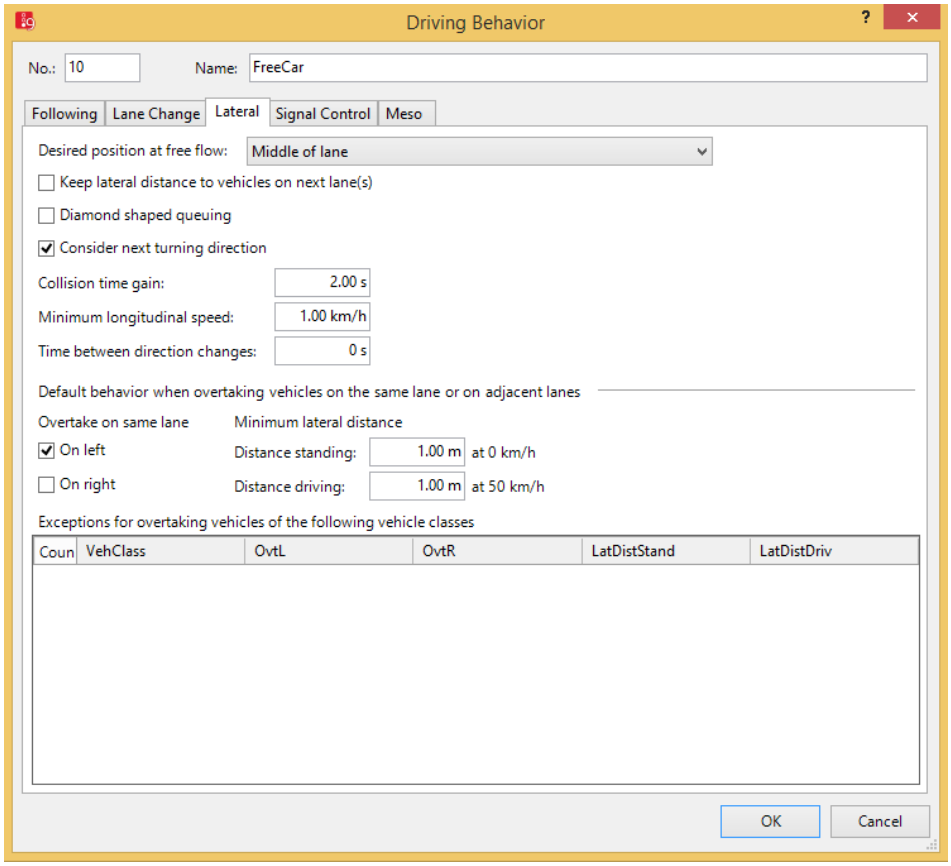
Maksimalna vrijednost opterećenja postiže se oko prvog sata simulacije i traje otprilike pola sata. Taj vremenski interval dalje u radu se označava kao t_k (*t-kritično*) Vrijednost prometne potražnje za taj period iznosi 1.250 $\left[\frac{voz}{h}\right]$. Ovakvim podešavanjem prometne potražnje potencirano je stvaranje prometne gužve u području oko ulazne rampe r_2. Zagušenje se postepeno širi glavnim tokom unatrag prelijevajući se u ćeliju L_3 te stvara poremećaj u normalnom odvijanju prometa. Ovako stvorena prometna situacija predstavlja glavni test za implementirani VSLC regulator zasnovan na *Q-učenju* implementiran u ćeliji L_2. Umjeravanje prometnog modela (u smislu provjere iznosa tokova i sl.) nije potrebno iz razloga što je model u osnovi testnog karaktera tj. ne predstavlja realnu okolinu za koju se traži/testira varijantno rješenje. Potrebno je napomenuti da su poprilično dobro umjereni parametri u VISSIM-u koji definiraju karakteristike i ponašanja vozača na autocestama. Velika pozornost je pridodana stvaranju što realnije prometne situacije prilikom priključivanja ulaznih tokova na glavni tok, te kod izlaza vozila s autoceste. Dinamika glavnog toka ovisi o nekoliko bitnih parametara kao što su: interval slijeđenja vozila (engl. *time headway*), najvećim vrijednostima ubrzanja (akceleracija/deceleracija) 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 15, 16 i 17.



Slika 15. Parametri slijeđenja vozila u VISSIM-u



Slika 16. Parametri promjene prometnih traka



Slika 17. Parametri bočnog gibanja

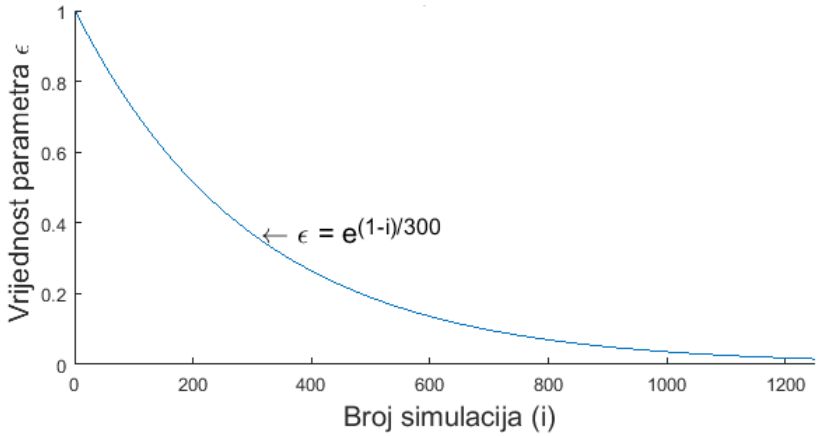
6.2. Postavke učenja

Kod algoritma *Q-učenja* postoje tri bitna parametra α, γ *i* ε. Parametar α predstavlja koeficijent učenja. Ovisno o iznosu koeficijenta α učenje može biti brže ili sporije. Za $\alpha = 0$ agent ima originalno *Q-učenje* (razmatra samo jedan korak razlike u procjeni Q funkcije). Povećanjem parametra α algoritam daje sve veći značaj budućim aproksimacijama vrijednosti u Q-matrici. Ovisno o modelu na kojem agent uči preporuka je da u početku parametar bude veći te ga je s vremenom potrebno postepeno smanjivati. U ovo radu parametar je odabran prema preporukama iz relevantnih radova u ovom području [7], [10], te iznosi $\alpha = 0{,}5$ tijekom svih simulacija (iteracija učenja).

Parametar γ predstavlja eksponencijalni faktor propadanja buduće nagrade. Svaki korak koji agent napravi umanjuje nagradu množenjem s faktorom γ. Kada ($\gamma \rightarrow 1$) algoritam *Q-učenja* razmatra udaljene nagrade s većom vrijednosti, težeći odgoditi nagrade. Dok za male vrijednosti ovog parametra ($\gamma \rightarrow 0$) agent razmatra samo trenutnu nagradu, buduće nagrade se ignoriraju. U praksi se vrijednost faktora γ kreću između (0,8 - 1). U ovom radu je faktor γ konstantan i iznosi 0,8.

Kod *Q-učenja* pojavljuje se problem koliko dugo istraživati, te kada početi iskorištavati stečeno iskustvo. Ovaj problem se u stručnoj literaturi naziva dilema istraživanja i iskorištavanja (engl. *exploration-exploitation dilemma*). Istraživanje se odnosi na isprobavanje nepoznatih akcija sa ciljem prikupljanja novih informacija iz okoline. Iskorištavanje predstavlja

korištenje stečenog znanja. To da algoritam odabire najbolju akciju s ciljem (maksimizacije/minimizacije) funkcije nagrade. Rješenje ovog problema predstavlja algoritam zvan ε -pohlepni pristup (engl. *ε-greedy approach*). Parametar ε predstavlja varijablu unutar ε -pohlepnog algoritma koja upravlja slučajnošću odabira akcija tijekom *Q-učenja*. S vjerojatnošću ε odabiru se nasumične akcije, dok se s vjerojatnošću $1-\varepsilon$ agent ponaša prema trenutnoj optimalnoj politici. To znači da kada vrijednosti ε teže ka jedinici $\varepsilon \to 1$ agent odabire slučajne akcije, dok za vrijednosti epsilona oko nule $\varepsilon \to 0$ agent bira optimalnu akciju. Odabrati vrijednost najboljeg ε nije lagano, te se u praksi primjenjuje postepeno smanjivanje vrijednosti ovog parametra od jedan prema nuli [10], [12]. Kod okoline koja se konstantno mijenja dobro je uvijek držati vrijednost ε iznad nule, tako da agent ima mogućnost povremenih slučajnih odabira akcija s ciljem novog učenja. U ovom radu ε je postepeno smanjivan tijekom svake nove simulacije prema funkciji prikazanoj na grafikonu 2.



Grafikon 2. Smanjivanje parametra ε tijekom broja simulacija

Kod jednostavnijih problema odlučivanja formulirani model MDP u kojem se nalazi agent, može imati posebno stanje koje predstavlja cilj. Kada agent dostigne taj cilj dobiva nagradu i završava jedna epoha učenja. Nakon toga počinje nova takozvana epizoda učenja, te se proces nastavlja ovisno o tome koliko dobro se želi naučiti agenta.

U ovom radu problem nije tako formuliran. Agent stalno uči tj. ne postoji stanje tijekom simulacije koje bi predstavljalo krajnji cilj nakon kojeg bi ponovno krenula nova epoha učenja. Može se reći da je krajnji cilj agenta kontinuirano minimiziranje kazne tijekom svake simulacije. Po završetku svake simulacije jedino što se korigira je parametar ϵ.

Trag prethodnog znanja (iskustva iz prethodnih simulacija) zapisuje se u Q-matricu. Na početku tijekom prve simulacije Q-matrica se popuni s nulama. U svakoj novoj simulaciji učita se Q-matrica iz prethodne simulacije. Tako se ovim procesom polako popunjava Q-matrica te nakon velikog broja simulacija ona predstavlja znanje agenta. Ponekad, ovisno o problematici i stručnosti osoba za pojedina područja Q-matrica može biti popunjena određenim vrijednostima koje popunjava stručnjak ili se za to primjeni neki standardni algoritam za rješavanje problema (u ovom slučaju bi to bio jedan od standardnih algoritama za upravljanje promjenjivim ograničenje brzine). Ovakvim pristupom može se dodatno ubrzati učenje agenta.

Učitavanje stare Q-matrice u svaku novu simulaciju izvršava se automatski pomoću MATLAB skripte u kojoj je napisan programski kôd u obliku for petlje. Unutar petlje se poziva

glavna skripta (kôd koji pokreće VISSIM simulaciju i implementirani VSLC algoritam zasnovan na *Q-učenju*), te joj se prosljeđuju ulazni podaci: Q-matrica, parametar ε itd. Po završetku svake simulacije podaci se spremaju u predviđenu mapu unutar MATLAB-a, te se proces nastavlja iterativno do zadanog broja simulacija (trajanje for petlje). Iz tako dobivenih podataka rade se analize, grafovi i donose zaključci o uspješnosti VSLC algoritma. Rezultati implementiranog algoritma nakon 1.250 simulacija prikazani su u slijedećem potpoglavlju.

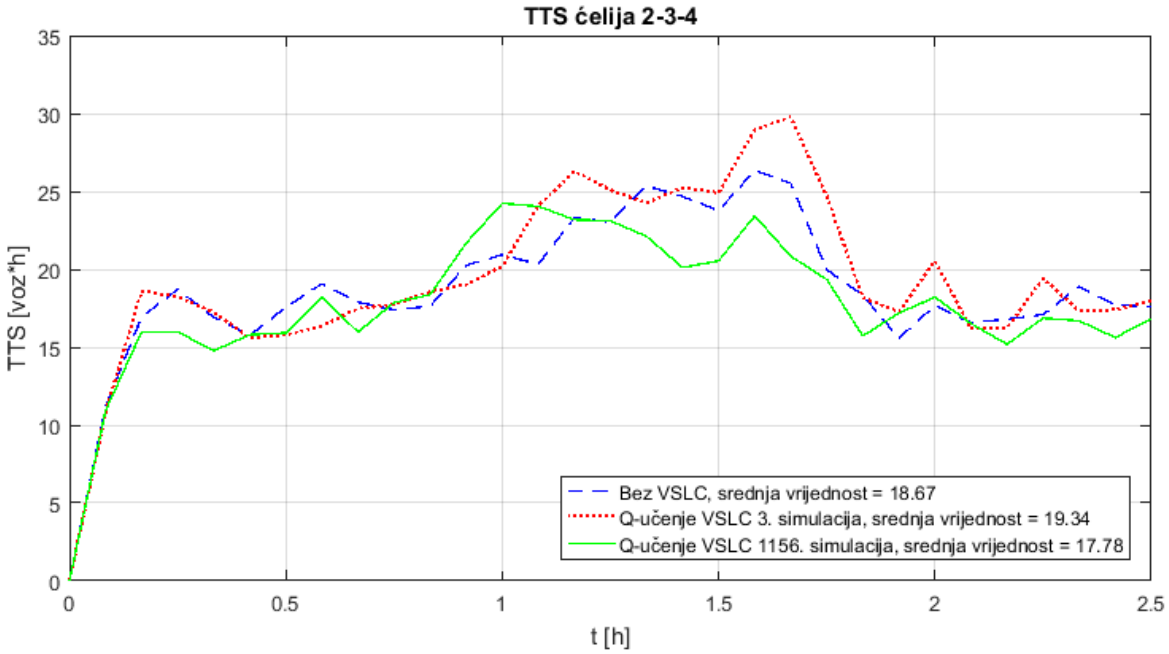
6.3. Dobiveni rezultati

U ovom potpoglavlju prikazani su te analizirani dobiveni rezultati prikupljeni tijekom simulacija. U radu je napravljeno ukupno 1.250 simulacija te za prikaz rezultata odabran je slučaj s početka učenja algoritma (3. simulacija) te slučaj iz 1.156-te simulacije. Na osnovu tih rezultata napravljena je evaluacija VSLC algoritma zasnovanog na potpornom učenju s ciljem ocjene kvalitete upravljanja ograničenjem brzine na promatranoj dionici testnog modela autoceste. Na osnovu prikupljenih rezultata mogu se donijeti određeni zaključci o sposobnosti učenja upravljačkog algoritma. Rezultati ovise o izračunatim odlukama o prilagodbi ograničenja brzine ovisno o prometnim uvjetima na simuliranom modelu.

6.3.1. TTS kao glavna mjera uspješnosti učenja

Ukupno provedeno vrijeme svih vozila u prometnoj mreži TTS predstavlja kvantitativni podatak koji dobro opisuje stanje prometnog toka. U ovom radu TTS predstavlja glavni pokazatelj uspješnosti učenja VSLC-a zasnovanog na potpornom učenju (izračun ograničenja brzine). Ukoliko nastane zagušenje unutar simuliranog modela TTS proporcionalno raste s povećanjem intenziteta zagušenja (povećanje gustoće prometnog toka). TTS se opisuje mjernom jedinicom umnoška vozila i vremena provedenog u prometnoj mreži $[voz \cdot h]$. Također u slučaju kada VSLC reagira povećanim smanjenjem izlaznih ograničenja brzina (npr. $v_{VSLC} = 60\ \left[\frac{km}{h}\right]$) povećava se vrijeme putovanja vozila za isti prevaljeni put (duljina ćelije L_2) samim time i vrijednost TTS-a. Ovime se sprječava da VSLC u nastojanju da smanji nastalo zagušenje previše smanjuje izlazna ograničenja brzine koja se šalju na VMS-ove. Na ovaj način VSLC mora računati optimalne vrijednosti ograničenja brzina tako da mora tražiti ravnotežu između prethodno spomenutih pojava s ciljem minimiziranja vrijednosti TTS-a koja mu se vraća u obliku kazne nakon svake poduzete akcije. Treba napomenuti da je TTS mjeren unutar svakog upravljačkog intervala VSLC sustava ($T_c = 300\ [s]$). Obuhvaćene ćelije ovim mjerenjem su L_2, L_3 i L_4. Na grafikonu 3 prikazane su vrijednosti za slučaj kada je algoritam učenja isključen (plava krivulja), kada je u fazi zagrijavanja tj. na samom početku (crvena krivulja) i nakon 1.156 simulacija učenja (zelena krivulja). Može se vidjeti da TTS vrijednost polako opada s povećanjem broja simulacija. U početku karakteristika VSLC-a zasnovanog na potpornom učenju je istraživanje prostora tako da je broj nasumično odabranih akcija visok. S vremenom algoritam učenja smanjuje slučajan broj odabranih akcija (izlazna ograničenja brzine) te počinje slijediti naučenu funkciju politike. Nakon velikog broja simulacija VSLC je polako konvergira ka optimalnoj politici. Pod pojmom optimalna podrazumijeva se najbolja politika (niz odabranih akcija) koje je algoritam naučio nakon većeg broja simulacija. To ne

znači da VSLC algoritam s još većim brojem simulacija ne bi naučio i bolju takozvanu funkciju politike ukoliko se parametar ε ne smanji skroz do nule,.



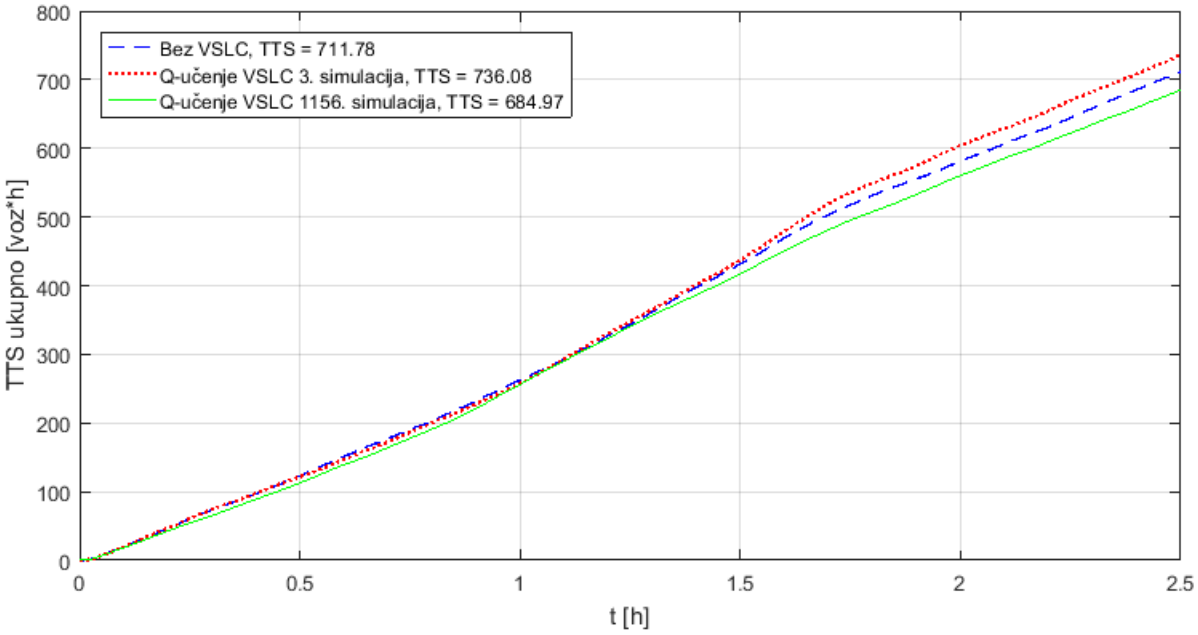
Grafikon 3. TTS $[voz \cdot h]$ mjera efikasnosti učenja VSLC-a mjerena unutar intervala T_c

Tablica 1. Vrijednosti TTS-a za ćelije L_2, L_3 i L_4, intervali mjerenja (5 $[min]$)

Obuhvaćene ćelije L_2, L_3 i L_4	**Bez VSLC**	**Q-učenje VSLC 3. simulacija**	**Q-učenje VSLC 1156. simulacija**
Maksimalan TTS $[voz \cdot h]$	26,34	29,78	24,23
Srednja vrijednost TTS $[voz \cdot h]$	18,67	19,34	17,78

Na osnovu vrijednosti krivulja na grafikonu 3 napravljena je tablica 1. Iz prikazanih prosječnih vrijednosti vidi se da je s primjenom naučenog VSLC-a postignuto smanjenje TTS-a mjereno unutar svakog pet minutnog intervala tijekom simulacije. Smanjenje iznosi otprilike 8,7 %. Ovaj rezultat predstavlja čvrstu činjenicu da algoritam VSLC-a zasnovan na *Q-učenju* ima sposobnost učenja.

Usporedno s prethodnim rezultatom može se prikazati i ukupno provedeno vrijeme svih vozila u mreži mjereno unutar intervala trajanja cjelokupne simulacije (2,5 $[h]$). Iz prikazanih krivulja na grafikonu 4 iščitane su vrijednosti ukupnog kumulativnog TTS za cijelu simulaciju za sva tri scenarija.

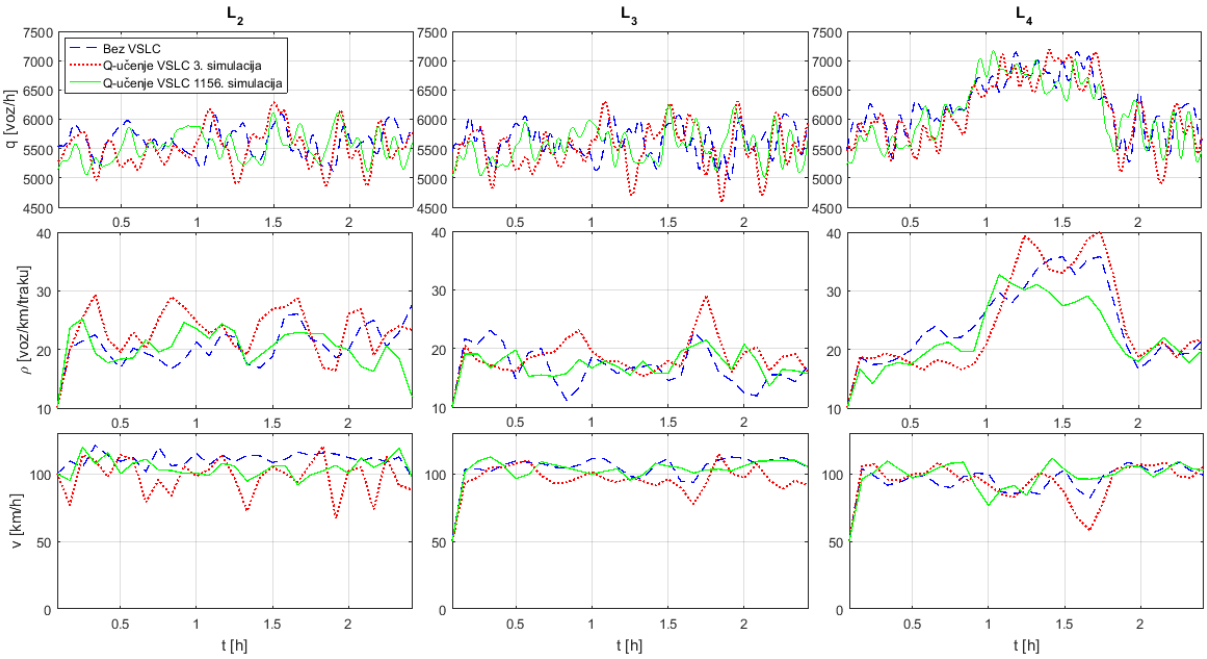


Grafikon 4. Kumulativni iznos TTS-a tijekom cijele simulacije

Na grafu se vidi da slučaj s naučenim VSLC (1.156 simulacija) ima najbolji rezultat odnosno ukupan TTS je najmanji. U odnosu na početak učenja smanjenje ukupnog TTS-a iznosi otprilike 7,4 %. Dok u usporedbi sa simuliranom prometnom potražnjom bez VSLC-a poboljšanje iznosi 3,9 %.

6.3.2. Protok, gustoća i brzina

Ovdje su prikazani rezultati za izmjerene prometne veličine (protok, gustoća, brzina). Protok vozila mjeren je na izlaznim detektorima ćelija L_2, L_3 i L_4. Gustoća prometnog toka mjerena je unutar svake ćelije kao i vrijednosti prosječnih brzina. Vrijednosti su izračunate nakon svakog intervala uzorkovanja T (30 $[s]$) tijekom trajanja simulacije. Na grafikonu 5 plava krivulja predstavlja slučaj bez VSLC-a, crvena za početno učenje VSLC-a i zelena krivulja za slučaj nakon 1.156 simulacija što karakterizira algoritam VSLC s naučenim znanjem o svojoj okolini te kako pojedine akcije (nizovi akcija) djeluju na promjene okoline.



Grafikon 5. Protok, gustoća i brzina toka

Duljina druge ćelije L_2 iznosi 0,8 $[km]$ te se na njenom početku nalazi ulazna rampa r_1, a na kraju ćelije nalazi se izlazna rampa s_1. Ulazni tok na rampi je konstantan tijekom cijele simulacije i iznosi 1.350 $\left[\frac{voz}{h}\right]$. Vrijednosti krivulja koje opisuju protok na tom dijelu dionice poprimaju približno iste vrijednosti. Iz krivulja koje opisuju gustoću toka u ćeliji L_2 vidljive su promjene koje neposredno slijede zbog djelovanja regulatora VSLC-a (promjene ograničenja brzine). Promjene u gustoći su izraženije unutar vremenskog intervala t_k. Uzrok tomu je poremećaj prometnog toka u okruženju rampe r_2 čija se karakteristika širi unatrag niz glavni tok (šok val) prelijevajući se iz ćelija $L_4 \rightarrow L_3$ do L_2. Preventivno djelovanje VSLC-a u vidu povremenog smanjenja/povećanja ograničenja brzine u ćeliji L_2 rezultira osciliranjem prosječnih brzina što je izraženije kod crvene krivulje gdje VSLC tek uči. U slučaju bez VSLC i s naučenim algoritmom VSLC-a brzine su ujednačenije tijekom trajanja simulacije. Cilj ovakvog pristupa je preventivno djelovati u ćeliji ispred u odnosu na zagušeni dio dionice kako bi se popravila prometna situacija na zagušenom dijelu dionice (oko rampe r_2). Algoritam VSLC-a zasnovan na potpornom učenju mora naučiti prepoznati određena stanja gustoće toka na promatranoj dionici na temelju kojih zaključuje da nastaje zagušenje na osnovu toga preventivno smanjuje prosječne brzine prometnog toka (harmonizacija brzine prometnog toka) u kontroliranoj ćeliji L_2. Smanjenjem brzine nadolazećeg toka vozila na zagušeni dio dionice smanjuje brzinu propagacije šok valova izazvanih zagušenjem u blizini rampe r_2 (područje ćelija L_3 i L_4) uzvodno uz autocestu. U tablici 2 su prikazane srednje vrijednosti rezultata za ćeliju L_2.

Tablica 2. Izmjerene srednje vrijednosti prometnih parametara u ćeliji L_2

Ćelija L_2	Bez VSLC	Q-učenje VSLC 3. simulacija	Q-učenje VSLC 1156. simulacija
Protok $[voz/h]$	5.538	5.445	5.445
Gustoća $[voz/km/trak]$	20,05	22,37	19,44
Brzina $[km/h]$	106,6	94,62	100,90

Zelena krivulja nalazi se povrh u odnosu na crvenu te su njene vrijednosti bliže vrijednosti brzine specifične za slobodni tok. Takva karakteristika je poželjna, a to je rezultat naučenog znanja algoritma (pravovremeni odabir ispravnih akcija). Kada zagušenje prođe cilj je da brzina toka što prije dosegne brzine specifične za slobodni tok. U slučaju crvene krivulje vrijednosti su najlošije tako da na početku učenja VSLC sustav čak i pogorša prometnu situaciju svojim djelovanjem u odnosu na simulaciju bez VSLC-a. Prosječno algoritam je popravio svoj rezultat povećanjem prosječne brzine za 6,2 % u odnosu na početak učenja. Navedene promjene u smanjenju brzine prometnog toka u ćeliji L_2 nisu značajnije utjecale na sam protok vozila.

Slijedi opis parametara treće ćelije (L_3) duljine 0,7 $[km]$ koja predstavlja dionicu bez ulaznih i izlaznih rampi. Prostorno predstavlja ćeliju koja je najbliža drugoj ulaznoj rampi. Karakterizira ju to što se nalazi u neposrednoj blizini ulazne rampe r_2. Zbog toga je najizloženija nepovoljnim prometnim uvjetima koje generira ulazna rampa r_2 unutar vremenskog intervala t_k. Izlazne vrijednosti krivulja na grafu u zavisnosti su o ograničenju brzine iz prethodne ćelije dok najveći utjecaj na njih direktno ovisi o zagušenju u području druge ulazne rampe. Na grafikonu 5 su vidljive izraženije oscilacije krivulja koje opisuju parametre prometnog toka ćelije L_3. U tablici 3 su prikazane vrijednosti za analiziranu ćeliju.

Tablica 3. Izmjerene srednje vrijednosti prometnih parametara u ćeliji L_3

Ćelija L_3	Bez VSLC	Q-učenje VSLC 3. simulacija	Q-učenje VSLC 1156. simulacija
Protok $[voz/h]$	5.531	5.433	5.438
Gustoća $[voz/km/trak]$	16,22	17,85	16,67
Brzina $[km/h]$	100,60	92,00	99,17

Povećanje prosječne brzine u odnosu na početak učenja može se pripisati određenom stečenom znanju algoritma nakon 1.156 simulacija. U postotcima je prosječna brzina povećana za 7,2 %.

Nakon što su analizirane ćelije L_2 i L_3 na red dolazi posljednja četvrta ćelija L_4. Njena duljina iznosi 2 $[km]$. Na početku iste nalazi se već spomenuta kritična ulazna rampa r_2. Područje oko rampe karakterizira zagušenje koje nastaje međudjelovanjem prevelike prometne potražnje na drugoj rampi (tijekom intervala t_k) i glavnog toka. Priljevni tok na drugoj ulaznoj

rampi odražava se na ukupan protok četvrte ćelije što se može vidjeti na krivuljama protoka. U ćeliji L_4 krivulje poprimaju veće iznose u trenutku najveće potražnje na r_2. Posljedice djelovanja VSLC-a u ćeliji L_2 vidljive su i u ovoj ćeliji na pripadnim krivuljama. Veće oscilacije su vidljive kod parametra gustoće i brzine toka. Primjenom VSLC-a postignuto je poboljšanje u trenutku vršnog opterećenja na r_2 održavanjem gustoće u prihvatljivim granicama što smanjuje intenzitet poremećaja u prometnom toku. Prometni tok upravljan VSLC ima u prosjeku veću prosječnu brzinu. Prosječne vrijednosti u ćeliji L_4 prikazane su u tablici 4.

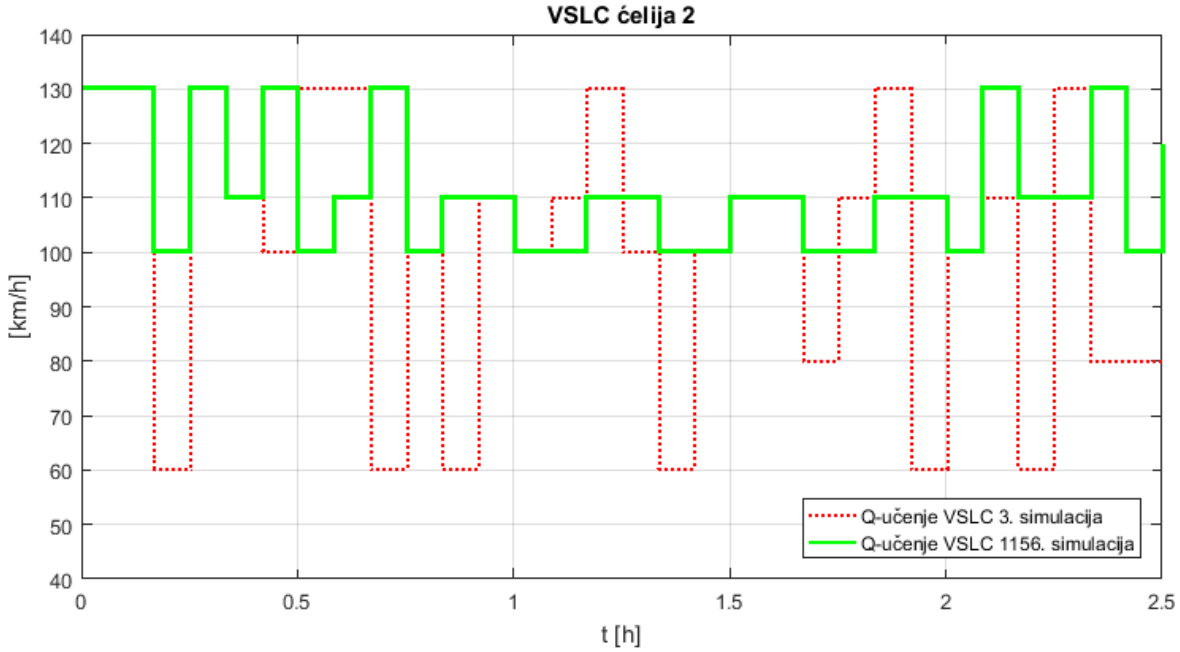
Tablica 4. Izmjerene srednje vrijednosti prometnih parametara u ćeliji L_4

Ćelija L_3	**Bez VSLC**	**Q-učenje VSLC 3. simulacija**	**Q-učenje VSLC 1156. simulacija**
Protok $[voz/h]$	6.099	5.980	5.971
Gustoća $[voz/km/trak]$	23,44	23,39	21,42
Brzina $[km/h]$	92,23	90,87	94,75

VSLC algoritam zasnovan na potpornom učenju uspio je smanjiti prosječnu vrijednost gustoće, a time je povećana prosječna brzina u L_4. Gustoća je smanjena za 9,2 % dok je brzina porasla za otprilike 4 %. Vrijednosti protoka ne odstupaju značajno.

6.3.3. VSLC - ograničenja brzine za ćeliju L_2

Stepenasti oblik krivulja na grafikonu 6 ograničenja brzina proizlazi iz činjenice da VSLC daje izlazne cjelobrojne vrijednosti ograničenja brzine. Raspon ograničenja brzine koje generira VSLC kreće se od minimalnih 60 $[km/h]$ do maksimalnih 130 $[km/h]$. Zbog specifičnosti prometnog toka čije se karakteristike dinamično mijenjaju u prostoru i vremenu bitna je pravovremena i ažurirana vrijednost ograničenja brzine ovisno o prometnoj situaciji na promatranoj dionici. VSLC sustav tijekom simulacije kontinuirano izračunava i šalje ograničenja brzine na VMS-ove u ćeliji L_2. Svako ograničenje brzine traje $T_c = 5$ $[min]$ te se nakon isteka izračuna novo ograničenje i šalje ponovo na VMS-ove. U slučaju pet minutnog režima rada VSLC sustava može malo kasniti s odgovarajućim promjenama ograničenja brzine (ponekad ne detektira kratkotrajne manje poremećaje u prometnom toku) što može utjecati na krajnji rezultat. Teoretski interval T_c bi moga biti i kraći npr. 2 $[min]$. Djelovanje VSLC u intervalima od dvije minute rezultiralo bi pouzdanijim i točnijim djelovanjem VSLC-a ovisno o trenutnoj prometnoj situaciji. Postavlja se pitanje implementacije takvog sustava u stvarnu okolinu i kako bi na česte promjene ograničenja brzine reagirali vozači. Na grafikonu 6 prikazane su vrijednosti ograničenja brzine za slučaj s VSLC-om u početku učenja (crvena krivulja) i za slučaj kada je VSLC naučio određeno znanje (popunio Q-matricu vrijednostima nakon 1.156 simulacija).



Grafikon 6. Iznos ograničenja brzine u ćeliji L_2

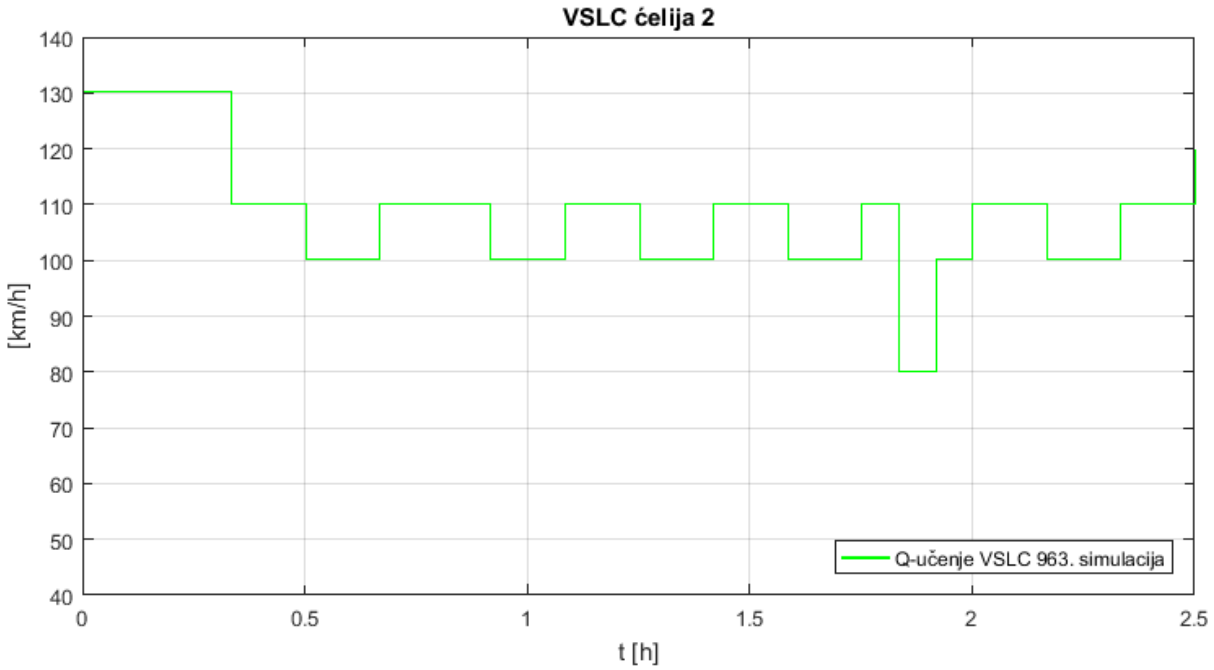
Iz grafa se može zaključiti da je u početku algoritam za VSLC znatiželjan te proučava svoju okolinu velikim brojem slučajno odabranih akcija (ograničenja brzine). Zelena krivulja predstavlja slučaj za VSLC koji je s vremenom naučio kako djelovati na pojedino stanje okoline (parametri toka) odabirom odgovarajućih akcija. Važna karakteristika zelene krivulje je manja promjena izlaznih brzina za ograničenje između pojedinih vremenskih koraka. To je definirano uvjetima funkcije nagrade kojom se kažnjavaju velike uzastopne razlike ograničenja brzine (veće od 20 $[km/h]$). Drugi uvjet u funkciji nagrade kažnjava osciliranje uzastopnih ograničenja npr. $(100 \rightarrow 110 \rightarrow 100 \rightarrow 110 \rightarrow 100\ [km/h])$. Na grafikonu 6 zelena krivulja u nekim intervalima izgleda kao da povremeno oscilira ali uglavnom s „periodom $2T_c$ i više“ npr. $(100 \rightarrow 100 \rightarrow 110 \rightarrow 110 \rightarrow 100 \rightarrow 100\ [km/h])$, a to je dozvoljeno. Na osnovu dosadašnjih izvršenih simulacija (1.250) VSLC algoritam još uvijek povremeno napravi prethodno spomenute loše radnje. Potreban je veći broj iteracija učenja kako bi u potpunosti savladao loše akcije. Generalno usporedivši crvenu i zelenu krivulju vidi se značajan napredak kvalitete rada što se pripisuje učenju algoritma VSLC-a primjenom potpornog učenja tako da upravljanje s vremenom postaje sve bolje.

Prethodno spomenuta varijabilnost može se numerički kvantificirati kao prosječno odstupanje vrijednosti numeričkog obilježja x_i (ograničenje brzine) od aritmetičke sredine $\bar{x}$. Vrijednosti su izračunate korištenjem formule za izračun standardne devijacije [16]:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}\,. \tag{16}$$

Srednja vrijednost zelene krivulje iznosi 111 $[km/h]$ sa standardnom devijacijom 11,47 $[km/h]$. U slučaju crvene krivulje srednja vrijednost iznosi 99 $[km/h]$ sa standardnom devijacijom 25,23 $[km/h]$.

Na grafikonu 7 prikazana je suboptimalna funkcija politike VSLC-a tijekom 963. simulacije. Algoritam VSLC-a je na ovom primjeru slijedio politiku koja nije optimalna iz razloga što na početku i na kraju simulacije izraženijeg zagušenja nema te je poželjno držati ograničenja na višim vrijednostima. Razlog ovakvog ponašanja leži u činjenici nedovoljnog broja posjeta svih parova (stanje-akcija). Kako se konvergencija Q funkcije ka optimalnoj politici zasniva na hipotezi beskonačno mnogo posjeta svih parova (stanje-akcija) proizlazi činjenica da je broj izvršenih simulacija (963) nedovoljan da se obiđu svi elementi prostora okoline veći broj puta. Razlog tomu je što ε-pohlepni mehanizam slučajnog odabira akcija polako smanjuje iznos parametra ε prema nuli tijekom svake nove simulacije te ponekad za određenu prometnu situaciju algoritam zaglavi u suboptimaloj politici. Iz ovoga slijedi zaključak da je ukupnih 1.250 simulacija ipak nedovoljno da bi se cijeli prostor pet dimenzionalne Q-matrice s 3.125 elemenata dovoljno dobro pretražio. Rješenje je puno veći broj simulacija od dosad izvršenih s osvrtom na pretraživanje (veći ε). Situacija je ohrabrujuća zbog toga što ipak politika odabira akcija nije konvergirala nizu akcija npr. ($60 \rightarrow 60 \rightarrow 80 \rightarrow 80 \rightarrow 60 \rightarrow 60$) tijekom cijele simulacije što dokazuje da algoritam ipak uči s vremenom. Postoje i druge metode pomoću kojeg je učenje velikih prostora brže (npr. uz pomoć neuronskih mreža). Implementirani algoritam *Q-učenja* u ovom radu prvenstveno se zasniva na čistom pretraživanju cijelog prostora stanja te ga treba pustiti u rad na duži period učenja kako bi se donesli čvrsti zaključci o konvergenciji učenja VSLC-a.

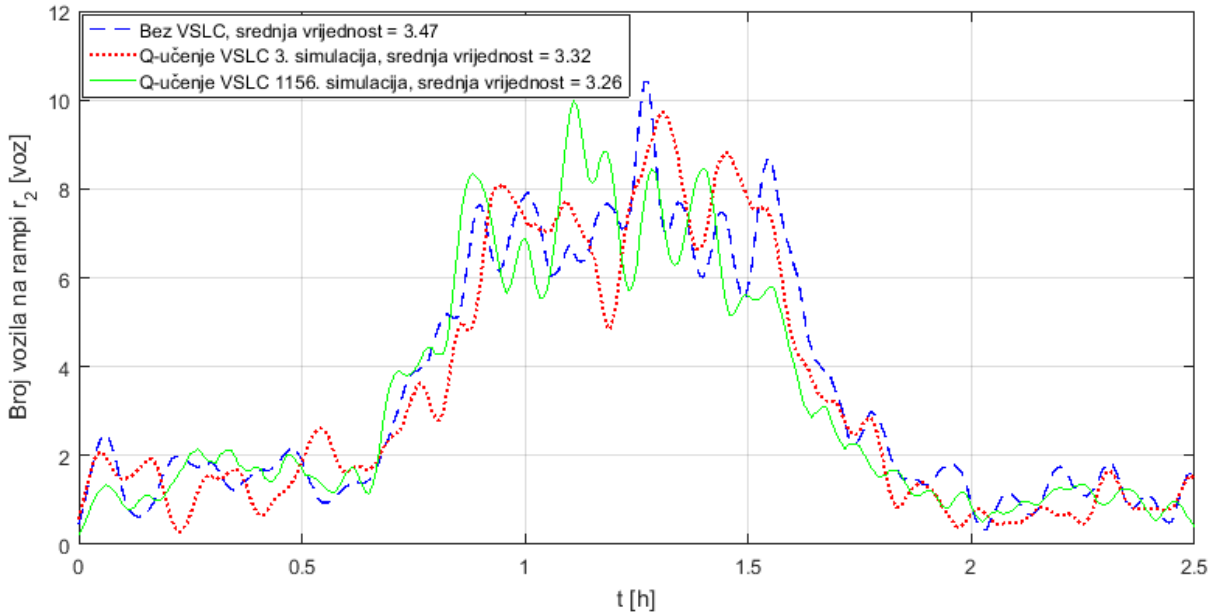


Grafikon 7. Iznos ograničenja brzine u ćeliji L_2, suboptimalna funkcija politike

6.3.4. Duljina reda čekanja na rampi r_2

Duljina repa čekanja vozila na rampi r_2 (broj vozila) u zavisnosti je o učinku VSLC na promatranom dijelu dionice (ćelija L_2). Ako je sustav smanjio zagušenje na glavnom toku u blizini ulazne rampe omogućuje se lakše uključivanje vozila s rampe na glavni tok. Čime se smanjuje duljina reda čekanja na pristupnoj cesti (rampa r_2). Na grafikonu 8 su prikazane krivulje za sva tri slučaja. Srednje vrijednosti duljine reda vozila na rampi su približno jednake

za sva tri slučaja, no ipak je slučaj s naučenim VSLC-om nešto bolji te je duljina reda čekanja na ulaznoj rampi r_2 u prosjeku smanjena za 6,4 %.



Grafikon 8. Red čekanja na rampi r_2

Duljina reda čekanja na rampi r_2 također može biti obuhvaćena unutar izračuna TTS-a kojim se kažnjava agent (VSLC) kako se radi o mjeri koliko vremena su sva vozila provela u prometnoj mreži. Uobičajeno je koristiti rampe kao mjeru TTS-a kod sustava koji integriraju strategije upravljanja ograničenjem brzine VSLC i sustava upravljanja priljevnih tokova na ulaznim rampama (engl. *ramp metering*). Kako je u ovom radu testiran sustav VSLC-a, duljina reda na rampi r_2 služi kao dodatan kvantitativni podatak za ocjenu rada sustava VSLC-a zasnovanog na potpornom učenju.

7. Zaključak

Primjena naprednih upravljačkih rješenja iz područja ITS-a omogućava dodatno poboljšanje razine uslužnosti na urbanim autocestama. Jedan takav sustav rješenja je VSLC odnosno upravljanje promjenjivim ograničenjem brzine na autocesti. U radu je implementiran i testiran algoritam za VSLC zasnovan na potpornom učenju. Princip rada algoritma se zasniva na izravnom upravljanju prometnim tokom izmjenom ograničenja brzine na upravljanoj dionici urbane autoceste. Algoritam mijenja ograničenja brzine na osnovi izmjerenih trenutnih prometnih parametara (gustoće i brzine prometnog toka) i trenutnim znanjem kojim raspolaže u obliku Q vrijednosti (Q-matrica) s ciljem regulacije prometa na upravljanoj dionici autoceste. Za testiranje sustava napravljeno je simulacijsko okruženje korištenjem programskih paketa MATLAB i VISSIM unutar kojeg je testiran sustav upravljanja ograničenjem brzine zasnovan na potpornom učenju. Upravljački zakon implementiran je u programskom paketu MATLAB, a simuliran je na mikroskopskom prometnom modelu u programskom paketu VISSIM.

Testni model autoceste na kojem je simuliran rad VSLC predstavlja dionicu autoceste s dvije ulazne i jednom izlaznom rampom. Model je prilagođen da omogućuje analizu utjecaja promjenjivog ograničenja brzina. Analiza rezultata napravljena je na simulacijama provedenim na modelu autoceste s primjenom i bez primjene VSLC sustava. Primjenom već spomenutog algoritma postignuto je poboljšanje u smanjenju TTS-a za simuliranu dionicu autoceste.

Povećanjem stupnja harmonizacije brzine primjenom VSLC smanjena su zagušenja na kritičnim područjima urbane autoceste oko druge ulazne rampe. U tom području u trenutku prevelike prometne potražnje dolazi do kritičnih zagušenja (povećana gustoća prometnog toka). Uz primjenu algoritma VSLC znatno je smanjenja gustoća prometnog toka vozila u tom području. Izmjereno poboljšanje s algoritmom u odnosu na rezultate simulacije bez algoritma doseže u pojedinim trenucima smanjenje od 9,2 %, a prosječna brzina povećanje za 4 %. Ovakvo poboljšanje pridonosi smanjenju intenziteta zagušenja i brzine širenja šok valova, što za posljedicu ima smanjenje ukupnog provedenog vremena svih vozila u prometnoj mreži (TTS). Funkcija nagrade koja je formulirana kao povratna veza iz okoline izravno je proporcionalna vrijednosti TTS-a unutar svakog upravljačkog intervala algoritma. Kao što je prikazano u poglavlju rezultata dokazano je smanjenje TTS-a s porastom broja učenja (veći broj simulacija). Nakon 1.156-te simulacije TTS je smanjen u odnosu na početak učenja algoritma za 8,7 %. Ovo je upravo činjenica koja dokazuje da VSLC zasnovan na *Q-učenju* ima sposobnost učenja.

Kod *Q-učenja* potreban je veliki broj iterativnih sekvenci učenja kako bi algoritam konvergirao ka optimalnoj Q vrijednosti. Kako je glavni cilj ovog rada implementacija okruženja za simulaciju sustava za VSLC zasnovanog na učenju, izostavljeno je dovoljno dobro pretraživanje prostora rješenja zbog premalog broja simulacija (ograničeni vremenski resursi). Kod testiranja algoritama zasnovanih na potpornom učenju neophodna su brza računala kako proces ne bi trajao predugo, kao u slučaju ovog rada gdje su simulacije trajale nekoliko dana. Uz više napravljenih simulacija za učenja moguće je postići bolje rezultate.

Kao prijedlog za daljnji rad bio bi nastavak rada na ovoj problematici prvo u pogledu implementacije mehanizama koji bi ubrzali proces učenja. Prvi korak bi bilo rješenje kojim bi se dalo više pažnje manje posječenim parovima stanje-akcija u Q-matrici s ciljem utvrđivanja prave Q vrijednosti za pojedini par. Nakon toga bi se išlo u proučavanje i implementaciju naprednijih rješenja poput učenja uz primjenu neuronskih mreža kojima bi se izbjeglo dugotrajno pretraživanje Q-matrice.

Literatura

[1] Korent, N., Kušić, K., "Analiza utjecaja promjenjivog ograničenja brzine na protočnost prometa i zagađenje okoliša", Rektorova nagrada, *Fakultet prometnih znanosti*, Zagreb, 2016.

[2] Gregurić, M., Ivanjko, E., Korent, N. and Kušić, K., "Short Review of Approaches for Variable Speed Limit Control", *ZIRP2016, Zagreb, Croatia, 12.04.2016. pp. 41-52,* 2016.

[3] Kušić, K., Korent, N., Gregurić, M., Ivanjko, E., "Comparison of Two Controllers for Variable Speed Limit Control", *ELMAR-2016, Zadar, Croatia, 12.-14.09.2016. pp. 101-106,* 2016.

[4] Dervisoglu, G., "Automatic Calibration of the Fundamental Diagram and Empirical Observations on Capacity", *For Presentation and Publication on the 88. Annual Transport Research Board Meeting, Siječanj 2009,* Whashington, D.C., 2008.

[5] Drmač, Z. i dr., Numerička analiza: osnovni udžbenik, Zagreb: skripta PMF–Matematičkog odjela, 2003.

[6] Abdulhai, B., Pringle, R., Karakoulas, G. J., "Reinforcement Learning for True Adaptive Traffic Signal Control", *Journal of Transportation Engineering,* vol. 129, no. 3, 2003.

[7] Walraven, E., "Traffic Flow Optimization using Reinforcement Learning", "Master's Thesis", *Faculty EEMCS, Delft University of Technology*, Delft, the Netherlands, 2014.

[8] Ivanjko, E., Kotlovska Nečoska. D., Gregurić, M., Vujić, M., Jurković, G., Mandžuka, S., "Ramp Metering Control Based on the Q-Learning Algorithm", *The Journal of Institute of Information and Communication Technologies of Bulgarian Academy of Sciences,* vol. 15, no. 5, pp. 11-22, 2015.

[9] Kotlovska, D., Bombol, K., "Intelligent Agent Based Traffic Signal Control on Isolated Intersection", *TEM Journal,* vol. 3, no. 3, pp. 216-222, 2014.

[10] Sutton, R. S., Barto, A. G., Reinforcement Learning: An Introduction, The MIT Press Cambridge, Massachusetts London, England, 2012.

[11] Silver, D., “UCL Course on RL”, 2015. [internet izvor]. Dostupno: http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html. [preuzeto 4. lipnja 2017].

[12] Dabelo Bašić, B., Šnajder, J., “Potporno učenje”, *Umjetna inteligencija*, Zagreb, Sveučilište u Zagrebu Fakultet Elektrotehnike i Računarstva (FER), dostupno na: https://www.fer.unizg.hr/_download/repository/AI-11-ReinforcementLearning%5B1%5D.pdf, 2015, p. 56.

[13] Group PTV, “PTV Vissim simulation software”, http://vision-traffic.ptvgroup.com. [preuzeto 7. Svibnja 2017.].

[14] Essetr, M., Žilić, T., MATLAB - Matrični laboratorij, Zagreb: Fakultet strojarstva i brodogradnje u Zagrebu, 2014.

[15] Ban, Ž., Matuško, J., Petrović, I., Primjena programskog sustava MATLAB za rješavanje tehničkih problema, Zagreb: Graphis, 2010.

[16] Kovač Striko, E., Fratrović, T., Ivanković, B.,, Vjerojatnost i statistika s primjerima iz tehnologije prometa, Zagreb: Sveučilište u Zagrebu, Fakultet prometnih znanosti, 2008.

Popis kratica

ITS	(Intelligent Transport System) inteligentni transportni sustavi
VSLC	(Variable Speed Limit Control) sustav promjenjivog ograničenja brzine
VMS	(Variable Message Signs) promjenjivi prometni znakovi
TTS	(Total Time Spent) ukupno provedeno vrijeme svih vozila u prometnoj mreži

Popis slika

Slika 1. Osnovna zavisnost prometnog toka o gustoći [4] 6
Slika 2. Prikaz jednostavnog primjera okoline 'labirinta' u kojem agent traži cilj [11] 9
Slika 3. Strategija odabira akcija (funkcija politike $\pi(s)$ „crvene strelice“) [11] 10
Slika 4. Funkcija vrijednosti $V\pi(s)$ [11] 11
Slika 5. Model okoline iz perspektive agenta u početku učenja [11] 11
Slika 6. Povratna veza akcija i stanja okoline [7] 14
Slika 7. Model urbane autoceste podijeljen u četiri ćelije 16
Slika 8. Shematski prikaz preslikavanja $s \rightarrow s'$ 18
Slika 9. Korisničko sučelje mikroskopskog simulatora VISSIM 20
Slika 10. Korisničko sučelje programskog paketa MATLAB 21
Slika 11. VISSIM-MATLAB simulacijski okvir [3] 22
Slika 12. Dijagram toka izvršne simulacije 24
Slika 13. Definirana distribucija brzina za teretna vozila unutar VISSIM-a 27
Slika 14. Model urbane autoceste podijeljen u četiri ćelije $L1$, $L2$, $L3$ i $L4$ 28
Slika 15. Parametri slijeđenja vozila u VISSIM-u 30
Slika 16. Parametri promjene prometnih traka 30
Slika 17. Parametri bočnog gibanja 31

Popis tablica

Tablica 1. Vrijednosti TTS-a za ćelije $L2$, $L3$ i $L4$, intervali mjerenja (5 min)..... 34
Tablica 2. Izmjerene srednje vrijednosti prometnih parametara u ćeliji $L2$..... 37
Tablica 3. Izmjerene srednje vrijednosti prometnih parametara u ćeliji $L3$..... 37
Tablica 4. Izmjerene srednje vrijednosti prometnih parametara u ćeliji $L4$..... 38

Popis grafikona

Grafikon 1. Generirana prometna potražnja na ulaznoj rampi $r2$ 29
Grafikon 2. Smanjivanje parametra ε tijekom broja simulacija 32
Grafikon 3. TTS $voz \cdot h$ mjera efikasnosti učenja VSLC-a mjerena unutar intervala Tc 34
Grafikon 4. Kumulativni iznos TTS-a tijekom cijele simulacije 35
Grafikon 5. Protok, gustoća i brzina toka 36
Grafikon 6. Iznos ograničenja brzine u ćeliji $L2$ 39
Grafikon 7. Iznos ograničenja brzine u ćeliji $L2$, suboptimalna funkcija politike 40
Grafikon 8. Red čekanja na rampi $r2$ 41

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 URBANIM AUTOCESTAMA ZASNOVANIH NA UČENJU**

na internetskim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom akademskom repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.

Student/ica:

U Zagrebu, 14.6.2017

Krešimir Kušić
(potpis)