Sustav potpore eko-vožnje za osobna vozila

Gruičić, Massimo

Sveučilište u Zagrebu
Fakultet prometnih znanosti

Završni rad

Sustav potpore eko-vožnje za osobno vozilo

Eco-Driving Support System For Personal Vehicle

Mentor: dr. sc. Pero Škorput
Student: Massimo Gruičić, 0135229665

Zagreb, rujan, 2016.

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI
ODBOR ZA ZAVRŠNI RAD

Zagreb, 20. travnja 2016.

Zavod: **Zavod za inteligentne transportne sustave**
Predmet: **Inteligentni transportni sustavi I**

ZAVRŠNI ZADATAK br. 3380

Pristupnik: **Massimo Gruičić (0135229665)**
Studij: Inteligentni transportni sustavi i logistika
Smjer: Inteligentni transportni sustavi

Zadatak: **Sustav potpore eko-vožnje za osobna vozila**

Opis zadatka:

Na razini Europske unije u području inteligentnih transportnih sustava, eko vožnja je prepoznata kao jedna od učinkovitijih mjera za poticanje energetske učinkovitosti u prometu. Istraživanje i razvoj mjera i tehničko-tehnoloških rješenja, doprinijeli su ekološki i energetski učinkovitijoj vožnji. Eko-vožnja, kao jedna od važnih komponenti održive mobilnosti značajno doprinosi zaštiti okoliša i smanjenju emisija štetnih plinova. Sukladno navedenom, u završnom radu potrebno je identificirani i analizirati relevantni parametri eko-vožnje, opisati primjere implementacija sustava eko-vožnje, objasniti karakteristike i prednosti eko-vožnje te komparativno usporediti programske jezike s kojima bi se u budućnosti mogla izraditi potencijalne aplikacije.

Zadatak uručen pristupniku: 4. ožujka 2016.

Mentor:

dr. sc. Pero Škorput

Predsjednik povjerenstva za završni ispit:

Sažetak

Na razini Europske unije u području inteligentnih transportnih sustava, eko vožnja je prepoznata kao jedna od učinkovitijih mjera za poticanje energetske učinkovitosti u prometu. Istraživanje i razvoj mjera i tehničko-tehnoloških rješenja, doprinijeli su ekološki i energetski učinkovitijoj vožnji. Eko-vožnja, kao jedna od važnih komponenti održive mobilnosti značajno doprinosi zaštiti okoliša i smanjenju emisija štetnih plinova. Sukladno navedenom, u završnom radu potrebno je identificirani i analizirati relevantni parametri eko-vožnje, opisati primjere implementacija sustava eko-vožnje, objasniti karakteristike i prednosti eko-vožnje te komparativno usporediti programske jezike s kojima bi se u budućnosti mogla izraditi potencijalne aplikacije.

Ključne riječi: inteligentni, energetski, parametri, implementacija sustava

Abstract

At the EU level in the field of intelligent transport systems, eco- driving is recognized as one of the effective measures to encourage energy efficiency in transport. Research and development measures and tehnical-tehnological solutions, have contributed to the ecological and energy-efficient driving. Eco-driving, as an important component of sustainable mobility contributes significantly to environmental protection and reducing emissions. Accordingly, the final work needed is identified and analyze the relevant parameters of eco-driving, describe examples of the implementation of the eco-driving, explain the features and benefits of eco-driving and comparative compare programming languages with which in the future could make potential applications.

Keywords: inteligent, energy, parameters, implementation of the system

Table of Contents

1. Uvod

Potrebe za smanjenjem emisije ispušnih plinova motornih vozila pogonjenih motorima s unutarnjim izgaranjem implementirana su najsuvremenija tehničko-tehnološka rješenja kako bi se povećala ekološka i energetska učinkovitost vozila.

Analize dominantnih stilova vožnji kod vozača, ukazuju na nisku razinu svijesti vozača vezano za primjenu pravila ekološke i energetski učinkovite vožnje. To je jedan od najvećih razloga zašto proizvođači vozila/ automobilske industrije ulažu u razvijanje i pružanje novih naprednih sustava za potporu vozaču.

U prvom poglavlju su pojašnjeni neki od osnovnih principa eko-vožnje koji bi svaki vlasnik vozila trebao znati, ukoliko želi izbjeći veće popravke, kvarove, potrošnju i zagađenje.

U drugom poglavlju je opisana arhitektura kooperativnih sustava funkcionalnostima tehničko-tehnoloških sustava potpore eko-vožnje. Detaljno su objašnjene mogućnosti vozača i vozila i napredna komunikacija u vozilima, pa skroz do infrastrukture na prometnicama.

U trećem poglavlju su pojašnjeni programski jezici, njih 4 preciznije. Jezik na kojem je baziran cijeli sustav u vozilu i koji se u 2016. godini u automobilskoj industriji smatra primarnim jezikom, no razvijanjem tehnologije, pojavljuju se noviji, brži, lakši za primjenu i pisanje koda jezici sa kojima se mogu pružiti potencijalne aplikacije za vozače ili vozilo.

U četvrtom poglavlju je kratko objašnjena komunikacija uređaja u vozilu i mobilnog uređaja i kako ta tehnologija funkcionira. Za primjer je dana već jedna gotova aplikacija koja se smatra jednom od najpopularnijih aplikacija za potporu vozaču.

2. Principi Eko-vožnje

Eko-vožnja je ujedno i vožnja na način koji vozaču, vozilu i okolišu ekološki najviše odgovara. Takvom vožnjom smanjuje se potrošnja goriva, emisija stakleničkih plinova, stopa nesreća, razina buka i mnogih drugih prednosti. Dakle, bitna stavka eko-vožnje je da ima pozitivan učinak na vozila i vozača.

Među opcijama politike za smanjenje CO2 emisije iz prometa koja je proučavala eko-vožnju je uvijek istaknuta među izborima s velikim potencijalom za smanjenje tj. značajno smanjenje potrošnje goriva učeći vozače kako promijeniti svoje ponašanje u smislu promjene načina vožnje. Takva promjena ponašanja je vrlo isplativ način za smanjenje potrošnje energije i emisija.

Dodatne preporuke za povećanje energetske učinkovitosti vozila odnose se na ekonomično korištenje klima uređaja te ostalo elektroničke opreme koju treba ugasiti ako se ne koristi. Također, električna energija dobiva se iz dodatnog izgaranja goriva u motoru s unutarnjim izgaranjem tako da je potrošnju potrebno reducirati. Svakako dodatne preporuke se odnose i na izbjegavanje dodatnog tereta te i aerodinamični otpor tereta na vozilo.

Točke dolje navedene ukratko objašnjavaju i pojašnjavaju principe kada se dolazi do eko-vožnje. Oni su ne ovisni o vrsti automobila i svrsi putovanja.

2.1 Priprema vozila prije putovanja

Vozači bi prije bilo kakvoga putovanja trebali osigurati da je njihovo vozilo pažljivo pregledano i osigurano. Odlazak kod mehaničara za detaljan pregled ili jednostavniji pregledi koje vozači mogu obaviti samostalno. Pregled ili odlazak kod mehaničara cik prije planiranog odlaska, nije najbolja praksa jer mnogo toga može poći po zlu.

Najjednostavnija solucija je ostaviti vozilo kod mehaničara dovoljno dana/tjedana prije odlaska na put.

Ukoliko je vozilo nedavno pregledano ili je vlasnik siguran da odlazak kod mehaničara nije potreban, onda bi bilo poželjno da vozači sami naprave osnovnu provjeru.

Osnovna provjera uključuje sljedeće:

1. Provjera tekućina. To uključuje ulje, ulje za servo volan, ulje u mjenjačkoj kutiji, antifriz, tekućina za pranje stakla i ulje u kočnicama. Ukoliko ulje nije promijenjeno u narednih 4 830 kilometara, trebalo bi ga promijeniti.

2. Nadopuniti sve tekućine koje su poprilično niske. Kupnja rezervnih boca u slučaju da se neka tekućina istroši.

3. Provjeriti brisače da li su ispravni i zamijeniti gumu na njima ukoliko je potrebno.

4. Provjeriti da li je akumulator ispravan i da li su sve žice dovoljno pričvršćene i očistiti koroziju. Tipičan vijek akumulatora u vozilu je od 3-5 godina.

5. Provjeriti kočnice. Ukoliko su kočnice na ispod 50%, a putovanje je duže od 1 000 kilometara, trebalo bi zamijeniti kočnice.

6. Provjeriti sva svijetla na vozilu da li su ispravna.

7. Provjeriti gume da li su istrošene i naravno ukoliko su gume za zimu, a putovanje je po ljeti, preporuča se promjena guma za ljeto.

8. Provjeriti da li u vozilu svijetli bilo kakva nepotrebna lampica ili lampica koja ne bi trebala svijetliti. Ako svijetli, trebalo bi popraviti problem.

2.2 Paljenje i gašenje vozila

Prilikom paljenja vozila po prvi puta u danu, vozač treba dopustiti vozilu da radi na optimalnoj brzini sve dok se ne podigne točan tlak motornog ulja.

Ovo je veoma važno pogotovo pri paljenju dok je motor hladan, zbog toga što hladan motor troši više ulja nego zagrijani motor, posebice na visokim okretajima.

VTL studije su dokazale da vožnja sa hladnim, ne zagrijanim motorom troši 15% više ulja nego dok je motor u zagrijanom stanju.

Motor će se zagrijati do optimalne temperature tek nakon otprilike 30 km (ovisi i o vanjskim faktorima) uključujući i nježnu vožnju prilikom zagrijavanja.

Prilikom zaustavljanja vozila, na primjer na semaforu, vozač bih trebao ugasiti vozilo da uštedi na gorivu, no konstantno paljenje i gašenje vozila nije uvijek najbolja opcija. Ovaj način funkcionira jedino ako je vozilo namijenjeno takvome pristupu, onda nema gubitaka prilikom konstantnog paljenja i gašenja.

2.3 Vožnja u većem stupnju prijenosa

Učinkovitost motora ovisi o brzini i o okretnom momentu samog motora. Za vožnju konstantnom brzinom ne može se odabrati radna točka za motor, a postoji određeni iznos energije potrebne za održavanje odabrane brzine.

Ručni mjenjač omogućuje vozaču odabir između nekoliko točak. Za turbo dizel prenizak stupanj prijenosa će pomaknuti motor u visoke okretaje motora, a kod niskog okretnog momenta, odnosno višeg stupnja prijenosa postiže se ušteda na gorivu i manja emisija ispušnih plinova.

Kod benzinskog motora, učinkovitost obično pada brže nego u dizel zbog gubitaka prigušenja pri dodavanju gasa.

Optimalna radna točka za vožnju male snage u pravilu je na vrlo niskom broju okretaja motora oko ili ispod 1000 okretaja u minuti. Na primjer, mali automobil možda treba samo 10-15 konjskih snaga (7,5 do 11,2 kilowata) za vožnju na 60 milja na sat (97 km / h).

Prilikom kretanja vozila iz mirujućeg položaja preporuča se da vozač promjeni u drugu brzinu čim prijeđe dužinu vozila i ukoliko se tako osjeća da prilagodi brzinu u odnosu na ostali promet.

Vožnja pri 50 km/h u petoj brzini danas ne predstavlja problem vozilima i vozači bi trebali prebacivati u veću brzinu čim prije ukoliko žele uštediti na gorivu, smanjiti količinu ispušnih plinova i razinu buke.

Prednost imaju vozila sa automatskim mjenjačem jer su prilagođeni za najoptimalniju vožnju gradom, autocestom ili drugim cestama što se tiče mjenjača.

2.4 Ubrzanje, usporavanje i korištenje kočnice

Učinkovitost goriva varira s obzirom na vozilo. Učinkovitost goriva tijekom ubrzavanja općenito se poboljšava sve do točke negdje blizu vršnog momenta, odnosno specifična potrošnja goriva pri kočenju.

BSFC je mjera za učinkovitost goriva za bilo koji pokretač koji troši gorivo, proizvodi rotaciju i snagu. Inače se koristi za usporedbu učinkovitosti motora s unutarnjim izgaranjem.

Jednostavna formula za izračun (BSFC-a):

$BSFC = r/P$ (g/(kW·h))

BSFC se uobičajeno izražava: (grami po kilowat/satu)

Gdje je:

r – potrošnja u gramima po sekundi (g/s)

P – proizvedena snaga u watima $P = \omega * t$ (w)

ω (omega) je brzina motora u radijanima po sekundi (rad/s)

t (tau) je moment motora u newton metrima (N * m) [5]

Ubrzavanje do nepotrebne brzine bez obraćanja pažnje što je ispred uzrokuje većem korištenju kočnice i nakon toga dodatnom ubrzavanju.

Puštanje vozila u trenutnoj brzini i dopustiti utjecaj momenta da uspori vozilo, odnosno okretaje, a prilikom toga i brzinu samom vozilu.

Najbolje vrijeme za upotrebu ove metode je prilikom spuštanja niz ulicu ili cestu ili prilikom približavanja crvenom svijetlu na semaforu.

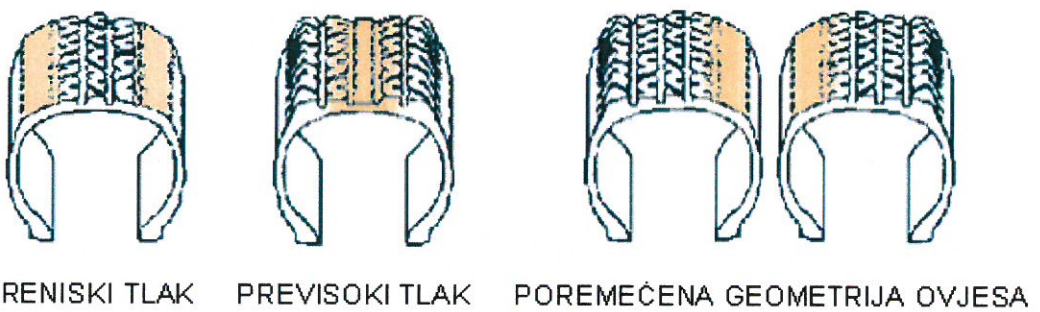
Bitna stavka ovakve metode je to što se potrošnja goriva smanjuje s obzirom na regularan način pritiskanja kočnice. [2]

2.5 Gume

Ripne na gumama prenose snagu vozila s obzirom na površinu ceste, nose vodu, prljavštinu i hlade gumu. Sa time da postoje i zakoni o propisanim dubinama ripni i uz pravilno održavanje rezultira se većoj sigurnosti i manjoj potrošnji goriva.

Nepotpuno napumpane gume povećavaju otpor kotrljanja, a nizak tlak u gumama zahtjeva više energije za pokrenuti vozilo što rezultira većoj potrošnji goriva do 15%.

Slika 1. Vizualizacija tlakova u gumama



Slika 1. Prikazuje izgled gume prilikom preniskog tlaka, što u gore navedenom dovodi do veće potrošnje goriva, lošijoj upravljivosti i većem trošenju gume, previsokog tlaka što također nije dobro jer se povećava trešnja ovjesa, što nije dobro za trajnost ležajeva, kuglastih zglobova, spona, amortizera, no nešto viši tlak manje je opasan od nešto nižeg, poremećena geometrija ovjesa koja je za svaki automobil propisana točna geometrija kotača, no ona se tijekom vremena mijenja. Geometrija bitno utječe na stabilnost, kočenje i vozna svojstva automobila. Ipak, o geometriji ovjesa najčešće se povede računa kada automobil počne vući u stranu ili kada se gume počnu nepravilno trošiti.

Prilikom kupnje guma kupci, vozači mogu pogledati koja guma ima najmanji mogući otpor kotrljanja da bi uštedili 3% potrošnje goriva i sa time da takve vrste guma proizvode manju buku.

Prilikom brže vožnje, konstantno jakog kočenja i neispravno upravljanje upravljačem vozila dolazi do većeg trošenja guma neovisno o kvaliteti.

Korisno za znati je da gume gube pritisak prilikom zahlađenja zraka (oko 7 kPa ili 1 psi za svakih 5 °C pada u temperaturi). Gume također gube određenu količinu pritiska (oko 14 kPa ili 2 psi po mjesecu).

psi - (per square inch) po kvadratnom inču

kPa – kilopascal [6]

„MythBusters" analiza/test potrošnje u odnosu na psi

Test vozilo – Ford Taraus 2004-2007

35 psi (preporuka proizvođača)

Gume pri 10psi = 3.7% veća potrošnja

Gume pri 30psi = 1.2% veća potrošnja

Gume pri 40psi = 6.2% manja potrošnja

Gume pri 60psi = 7.6% manja potrošnja [7]

2.6 Nepotreban teret

Vozači također mogu potrošnju goriva smanjivanjem mase vozila, tj. broja osoba, količine tereta, alata i opreme u vozilu. Svaki kilogram prtljage uzrokuje većoj potrošnji goriva. Preciznije, 100 kg može uzrokovati većoj potrošnji za 0,31 l/100 km, što znači da punjenje spremnika goriva do gornje granice nije potrebno, a isto tako nepotrebna prtljaga koja zauzima prostor i povećava ukupnu masu vozila.

Uklanjanje uobičajenih nepotrebnih dodataka kao što su krovni nosači, prednji branici, deflektori za vjetar (ili „spojleri"), „bumperi" i uske i niže profile guma. Sve to će poboljšati učinkovitost goriva smanjivanjem težine i aerodinamičkog otpora.

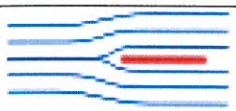
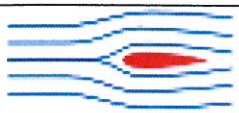
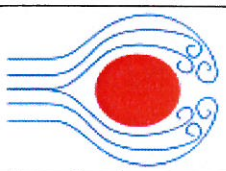
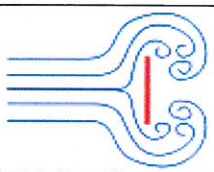
2.7 Dobra aerodinamika

Bitna stavka za uštedu goriva je dobra aerodinamika, pogotovo prilikom velikih brzina.33% veći aerodinamički otpor može povećati potrošnju goriva za 2 l/100 km pri brzini od 160 km/h. [1][2][3]

Manji otpor strujanju zraka oko automobila smanjuje potrošnju, znaci sto je koeficjent otpora manji to će i potrošnja biti manja, samim time automobil će sa jednakom snagom bolje ubrzavati, postizati veću maksimalnu brzinu, zato što motor treba savladati manji otpor, znaci gura manje zraka. Isto tako, strujanje zraka oko automobila utiče na njegovo ponašanje.

Sada neka aerodinamička svojstva se koriste da bi se automobil po potrebi usmjerio u željenom smjeru, sportski automobili se usmjeravaju prema podlozi, da bi dobili veći downforce, odnosno bolje prianjanje uz podlogu. Ali sada tim boljim prianjanjem uz podlogu se povećava otpor zraka, pa samim tim je i potrošnja veća, ali je to praktično neophodno kod automobila koji postižu velike brzine. Znači sto je otpor zraka manji to je manja potrošnja, sto je downforce jaci i potrošnja veća.

Tablica 1. Prikaz ovisnosti oblika objekta o protoku zraka

Oblik i protok	Otpor	Protok
	0%	100%
	~10%	~90%
	~90%	~10%
	100%	0%

U tablici 1. vidimo da što je otpor zraka manji, a protok zraka veći oko objekta, to je manja potrošnja goriva, emisija plinova i dr. Povećavanjem otpora zraka i manjem protoku zraka donosi većoj potrošnji goriva i zagađenju.

3. Kooperativni sustavi potpore eko-vožnje

Odgovornost svjetskih proizvođača vozila prema okolišu ne zaustavlja se na unapređivanju tehnologije za smanjenje CO2 emisije. Dugo vrijeme na ovom području je bio potpuni fokus svih autoindustrija s ciljem manjeg zagađenja okoliša. Rezultatom tog cilja vozila s motorom s unutarnjim izgaranjem imaju sve manju emisiju CO2.

Industrija se fokusirala na tehnologijska rješenja vozila, međutim, smanjenje CO2 emisije može biti postignuto već i razmišljajući o ponašanju vozača u prometu - ne, samo gledajući na vozila koje vozači voze već i kako ih voze. To je cilj sustava Eko-vožnje za smanjenje emisije CO2, koje može biti primijenjeno od strane bilo kojeg vozača u bilo kakvom vozilu.

Jednostavne promjene načina vožnje vozača mogu značajno utjecati na zagađenje okoliša s smanjenjem emisije CO2. Sila kočenja, sila ubrzanja i vrijeme mijenjanja stupnja prijenosa su bitne činjenice u načinu vožnje.

S željom za praćenje i snimanje sile ubrzanja, sila kočenja i vrijeme mijenjanja stupnja prijenosa proizvođači vozila su proizveli :

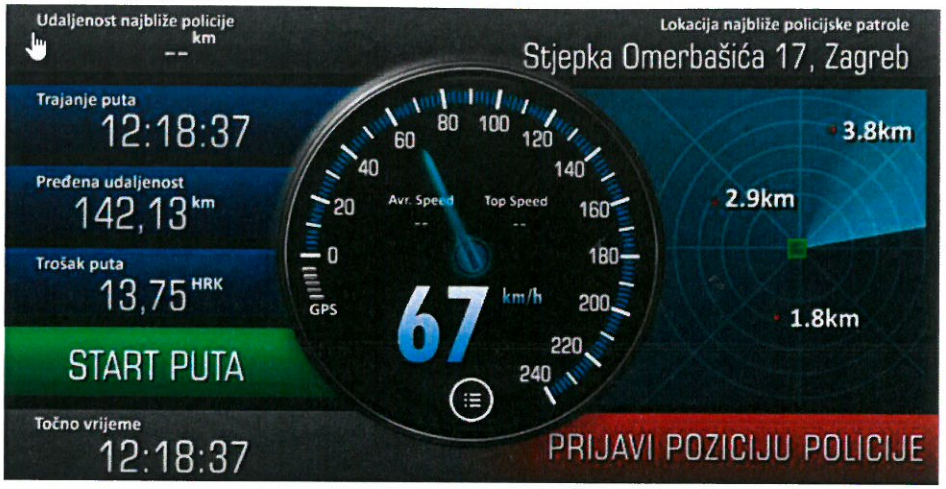
- mobilne aplikacije (slika 2)

- integrirane aplikacije u vozilima (slika 3)

- uređaje za naknadnu ugradnju u vozilo s praćenjem preko web-a (slika 4)

Same aplikacije rade na način da snimaju sve te podatke pomoću senzora za mjerenje G-sile. Na tržištu imamo široki izbor aplikacija koje pružaju podršku Eko vožnji, ali sve imaju isto načelo rada. Svaka aplikacija analizira podatke pomoću senzora za količinu G-sile.

Određivanje granice prekoračenja G-sile je proizvoljno od strane svakog proizvođača za svaku vrstu vozila drugačije (motocikl, osobni automobil, teretno vozilo, priključno vozilo). [8]

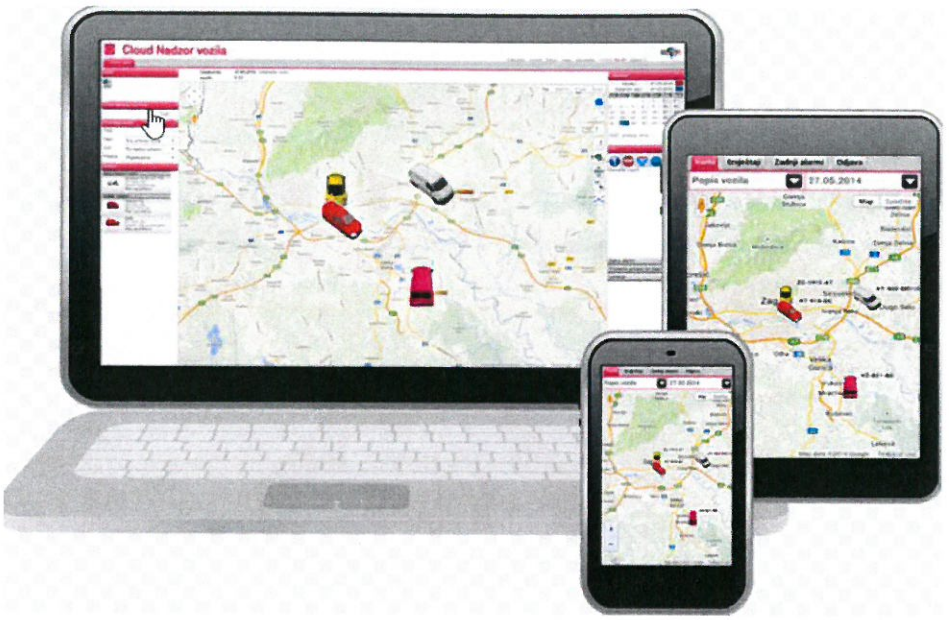


Slika 2. Vizualni primjer sučelja mobilne aplikacije za vozilo

Na slici 2. vidimo prikaz sučelja mobilne aplikacije koja pruža vozaču informacije o trenutnoj brzini, troškovima, vremenu putovanja, lokaciju (GPS) i pruža informaciju sa drugim vozilima, odnosno korisnicima aplikacije o lokaciji policijskih vozila ili patrola.

Slika 3. Vizualni primjer implementirane aplikacije GPS u monitor vozila

Na slici 3. vidimo primjer implementiranog GPS sustava u vozilu koji omogućuje vozaču preglednu i sigurnu vožnju tokom putovanja. Prednost ove aplikacija implementiranog u vozilo je da vozač ne mora koristiti mobilni uređaj prilikom vožnje i dovoditi u opasnost sebe i druge suvozače.



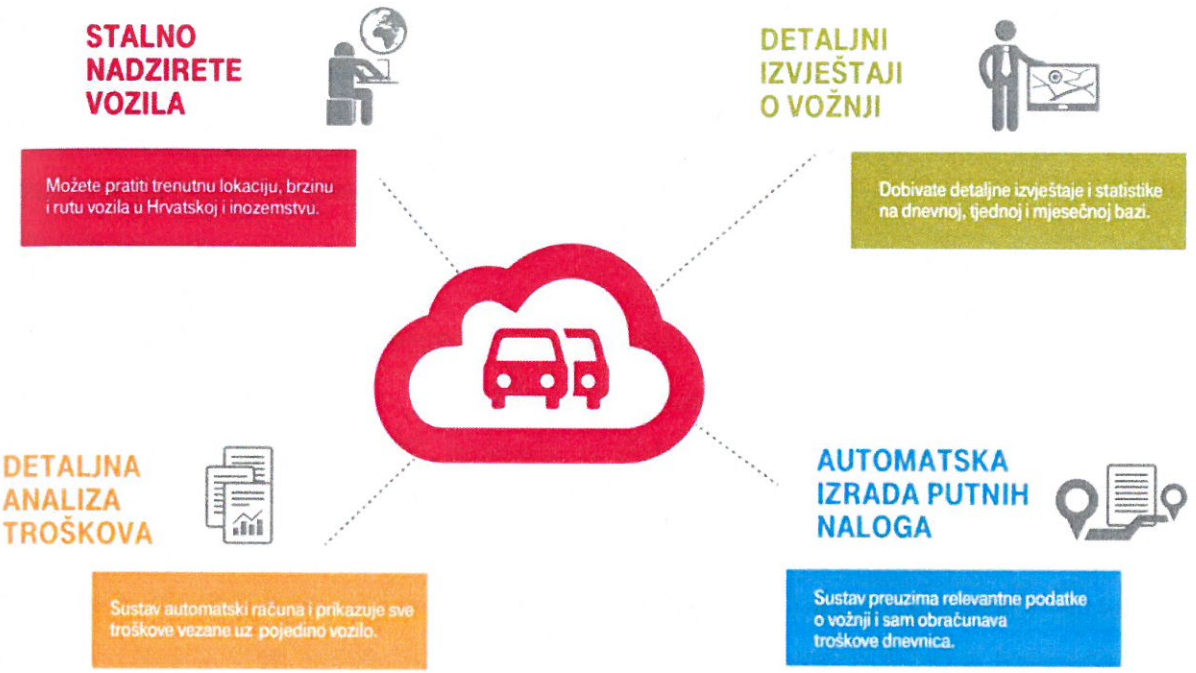
Slika 4. Primjer „Cloud" praćenja svih aktivnih vozila

Slika 4. prikazuje primjer sustava (Telecoma) za nadziranje i kontrolu putanje aktivnih vozila i moguće dojave informacija vozaču za koje vozač još nije svijestan.

3.1 Arhitektura kooperativnih sustava potpore eko-vožnji

Arhitektura kooperativnih sustava potpore eko-vožnji predstavlja obuhvaća tehničke aspekte i povezana organizacijska, pravna i poslovna pitanja arhitekture kooperativnih sustava. One pomažu da se rezultirajuća upotreba sustava potpore eko-vožnji može planirati na logičan način i uspješno integrirati sa drugim tehničkim podsustavima potpore eko-vožnji o kojima će biti više riječi u narednim pod poglavljima.

Na slici 5. prikazan proces sustava za potporu eko-vožnje koja omogućava sustavsko definiranje željenih performansi, naznačava ponašanje i omogućuje proširivanje funkcionalnosti prema budućim očekivanjima korisnika ili sustavskih zahtjeva.



Slika 5. Proces sustava potpore vozaču

Slika 5. pojednostavljeni prikaz sustava komunikacije, potpore i analize za vozila.

Sustavi potpore eko-vožnje prihvatili su široku paletu aplikacija zasnovanih na suvremenim komunikacijskim tehnologijama koje su svoju primjenu pronašle u područjima kao što je:

- povećanje sigurnosti putovanja
- smanjenje utjecaja na okoliš
- poboljšanje upravljanja prometom 14
- maksimiziranje koristi od prijevoza i za korisnika i za davatelja usluge
- poboljšanja u korištenju javnog prijevoza i sl. [9]

Automobilska industrija na tržište automobila komercijalno postepeno pušta rješenja i sustave kao što su:

- sustavi kontrole sigurnog razmaka

- sustavi održavanja brzine
- autonomno vođenje vozila unutar prometne trake
- izbjegavanje pretjecanja na nesigurnim mjestima
- siguran prolazak kroz raskrižje i sl.

Međutim, navedeni sustavi te koristi i prednosti koje oni pružaju korisnicima prometnog sustava mogle bi se dodatno uvećati kada bi pojedinačna vozila bila u mogućnosti kontinuirano međusobno komunicirati ili komunicirati sa prometnom infrastrukturom.

Kooperativni sustavi su sljedeći veliki izazov za opremu u vozilu i inteligentne transportne sustave (ITS). U suštini takvi sustavi omogućuju kvalitetan pristup informacijama o praćenju 17 prometa o drugim vozilima i korisnicima prijevoza, omogućujući na taj način efikasnu sigurnost i mobilnost.

Kao takvi kooperativni sustavi pružaju unaprijeđene informacije o vozilima, njihovoj lokaciji, smjeru orijentacije i infrastrukturi putem cestovne mreže, a time utječu na prepreke i nesreće.

Prednosti kooperativnih sustava proizlaze iz povećane dostupnosti informacijama o vozilima i njihovoj okolini. Isti skup informacija može se iskoristiti kako bi se proširila funkcionalnost sigurnih sustava u vozilu, a kroz vozilo se postiže komunikacija sa infrastrukturom za učinkovitiju kontrolu i upravljanje prometom.

Prednosti podrazumijevaju:

- Povećan kapacitet cestovne mreže
- smanjenje zagušenja i zagađenja
- niži operativni troškovi vozila
- učinkovitija logistika
- povećanje učinkovitosti sustava javnog prijevoza
- kraće i više predvidljivo putovanje
- bolji i učinkovitiji odgovori na opasnosti, incidente i nesreće

Dizajneri kooperativnih sustava napravili su pretpostavke o tome kako će i gdje njihov sustav djelovati i graditi sigurne osobine koje će mu omogućiti željenu strukturu koja neće biti podložna čestim i naglim promjenama. Svaki dizajner nekog sustava nastoji osigurati stabilnu osnovu za upotrebljiv i izvediv sustav koji mora biti razumljiv i jasan.[10]

Funkcioniranje sustava bazira se na međusobnom djelovanju podsustava i suradnji sa svim funkcionalnim zahtjevima i ciljevima sustava, jer kao takav postaje jednostavan i za upravljanje i održavanje tijekom životnog vijeka sustava samom korisniku. Samim time arhitektura sustava obuhvaća sve ciljno orijentirane funkcije koje osiguravaju djelovanje sustava i potpomaže funkcije koje osiguravaju djelovanje i izvedbu sustava.

Sustavi se razvijaju na razne načine i u raznim područjima i svaki sadrži određen karakteristike i prednosti koje utječu na razvoj prometa.

U odnosu na postojeće sustave, tehnologija kooperativnih sustava omogućuje dvosmjernu komunikaciju:

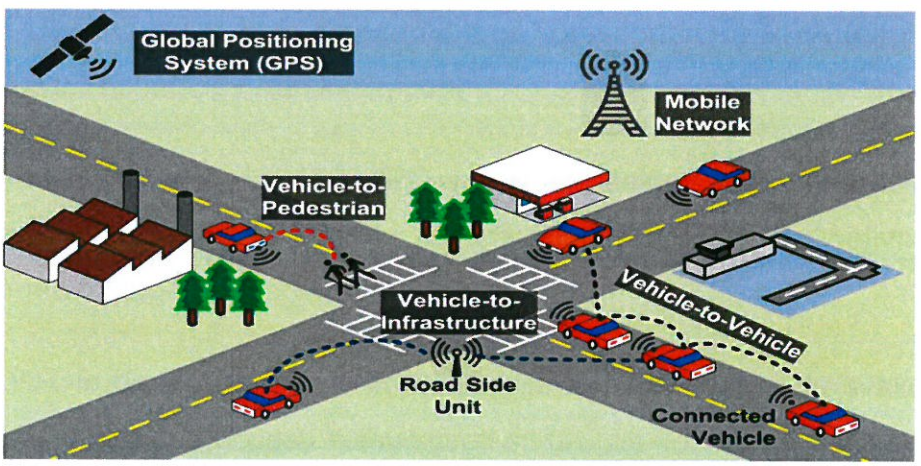
- V2V - vozila s vozilom
- V2I – vozilo s infrastrukturom
- V2U - vozilo s ostalim korisnicima
- I2U – infrastruktura s ostalim korisnicima

Ovaj pristup se približava prirodnoj komunikaciji više učesnika u nekom poslu. Kao posljedica, ostvaruje se mogućnost izgradnje socijalne inteligencije. [9]

3.2 V2X kooperativni sustav

Povezanost vozila je ključ kako bi se omogućila široka paleta opcija informacijskih usluga za vozače, proizvođače vozila, tijela javne vlasti. Nove tehnološke komunikacije omogućuju dinamičke informacije u stvarnom vremenu za povećanje sigurnosti vozača i učinkovitosti u prometu, pružanje usluga platnog prometa, komercijalni multimedijski sustav i omogućavanje prikupljanja vrijednih podataka za zaštitu okoliša.

Vozilo s vozilom (V2V) i vozilo s infrastrukturom (V2I) zajedno čine (V2X) je jedno od glavnih rješenja za „povezano vozilo" okruženje. Cestovni operatori, infrastruktura, vozila, njihovi vozači i ostali sudionici u prometu moraju surađivati da bi se osiguralo najučinkovitije, najsigurnije i najudobnije putovanje. V2X će biti glavni parametar za ovaj koncept kooperativne mobilnosti.



Slika 6. Prikaz komunikacije

Na slici 6. Poopćeni prikaz kooperativne mobilnosti za vozila, infrastrukturu i pješake putem komunikacije.

V2X se temelji na 5.9 GHz DSRC radio komunikacije, dvosmjerna, kratkog dometa, bežična komunikacijska tehnologija dizajnirana posebno za pokretne objekte kao što su vozila. U principu to omogućuje vozilima razmjenu podataka sa drugim vozilima, jedinicama uz cestu, senzorima i ponašanje je slično Wi-Fi komunikaciji, ali uz korištenje učinkovitog WANET umrežavanja.

U principu sve ovisi o broju vozila koja su opremljena sa DSRC tehnologijom. Sa V2X svako vozilo je u stanju osjetiti njegovu okolinu. Može iskoristiti podatke o vozilima u blizini za izračun njegove sadašnje i buduće pozicije i na taj način omogućuje vozaču da bude spreman na nepredviđene situacije. To omogućuje sigurnost za izbjegavanje sudara ili predviđanja opasnih situacija. [10]

3.2.1 V2V komunikacija

Vozilo s vozilom (V2V) komunikacija za sigurnost je bežična razmjena podataka između vozila koja su u neposrednoj blizini i pruža mogućnosti za značajna sigurnosna poboljšanja.

V2V komunikacije za sigurnost je ključna komponenta istraživačkog programa povezanosti vozila u ITS JPO (engl. Intelligent Transportation Systems Joint Program Office) od USDOT (engl. U.S. Department of Transportation) RITA (engl. Research and Innovative Technology Administration)

Kroz više modalni program ITS JPO i privatni sektor su u mogućnosti dijeliti informacije između vozila kako bi se postigla sigurnosna transformativna prednost sektor više-modalnog transporta

Vizija USDOT je da V2V program komunikacije za sigurnost je da svako vozilo na cesti u prometu je u mogućnosti komunicirati sa drugim vozilom ili vozilima i da će taj bogati skup podataka i komunikacije podržati novu generaciju aktivnih sigurnosnih aplikacija i sustava

Četiri glavna cilja V2V programa komunikacije za sigurnost su :

- Razviti V2V aktivne sigurnosne aplikacije koje se bave najkritičnijim scenarijima sudara
- Razviti strogu procjenu sigurnosnih prednosti koje će doprinijeti cjelokupnoj procjeni 2013 National NHTSA (engl. Highway Traffic Safety Administration) agenciji za odluke
- Rad s industrijom i omogućiti tržišne čimbenike koji će ubrzati V2V prednosti kroz V2V tehnologije u vozilu i kroz korištenje opcija oprema i modifikacija

kako bi se osiguralo da prvo vozilo opremljeno sa V2V tehnologijom da vlasnik bude zadovoljan sa uloženim

- Izgradnja iz rezultata VII programa „(engl. proof-of-concept)" testova, kompletiranje razvoja i testiranje V2V komunikacijske tehnologije i standarda

Dodatni ciljevi:

- Uvesti napredu V2V bežičnu tehnologiju za smanjenje, ublažavanje i sprječavanje značajnog postotka sudara vozila od strane loših vozača
- Uspostaviti snažne DSRC standarde sigurnosnih aplikacija za kritične situacije
- Ubrzati tehnologiju u vozilu koja osigurava vrijednost prvih V2V vozila

Istraživački program :

1 – identifikacija scenarija kritičnih sudara za V2V je završena. Početna mjerila za funkcijsku sigurnost aplikacije, preformanse i učinkovitosti su razvijene

2 – osigurati interoperabilnost i odrediti prateće infrastrukturalne potrebe za V2V implementaciju. Sigurnosne aplikacije moraju raditi na svim tipovima opremljenih vozila i u skladu s komunikacijskim standardima kako bi se osigurala sigurnost i integritet poruke

3 – razviti rigorozne procjene sigurnosne prednosti. Razvoj mjera uspjeha objektivnim i ispitnim postupcima, te adaptacija napredne tehnologije izbjegavanja sudara će pomoći u potvrđivanju sigurnosne prednosti

4 – razviti prototip aplikacije za aktivnu sigurnost i proći kroz objektivne testove i terenska ispitivanja. Razvoj ovih aplikacija ovisi o i pomaže u analizi za zahtjeve funkcionalnosti i performanse za temeljne tehnologije, kao što su pozicioniranje i komunikacija. Međutim, dodatna istraživanja treba svesti na adresu složenijih scenarija sudara za broj dodatnih scenarija. Dodatni napori u ovoj točci će biti kooperativno istraživanje i razvoj sigurnosne aplikacije s partnerima Europske Unije

5 – razviti učinkovita sučelja u vozilu za vozače. Učinkovitost sustava za upozorenje sudara ovisi o kvaliteti sučelja, što može utjecati na vozačeve reakcije

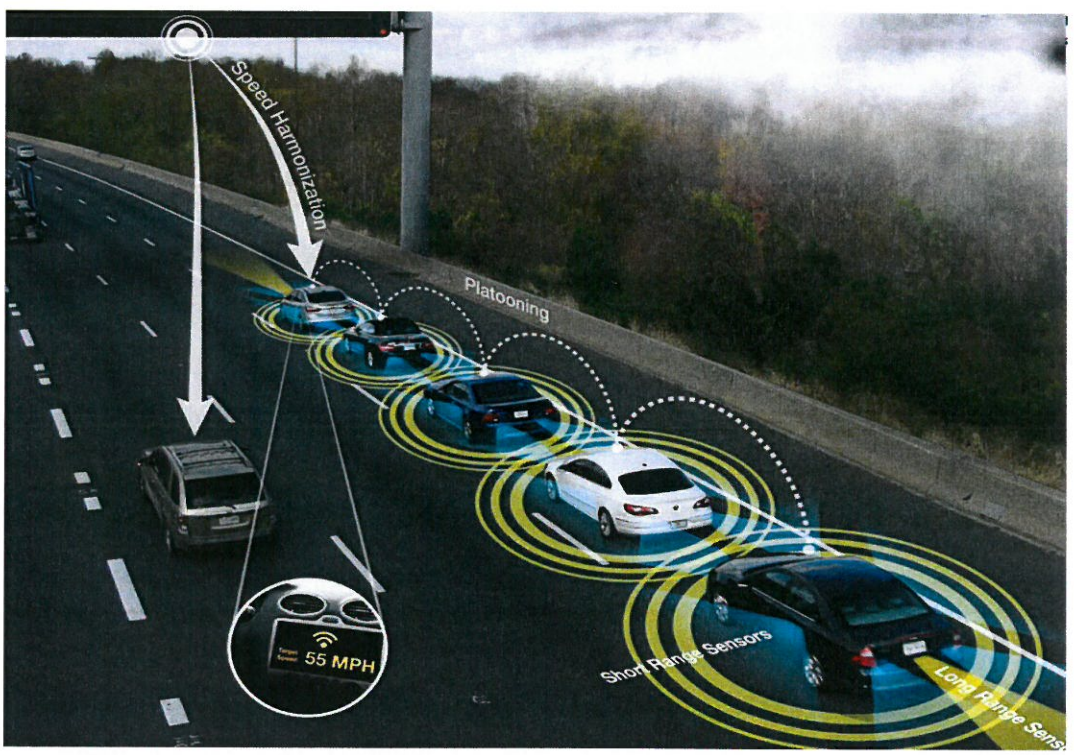
6 – istražiti politička pitanja i formulirati regulatorne odluke u okviru šireg program

7 – izraditi i ocijeniti V2V sigurnosne aplikacije koje zahtijevaju jedinstvene potrebe i dinamiku komercijalnih vozila, velikih kamiona, i motornih vagona. Istraživanje iz NHTSA procjenjuje da V2V aplikacije mogu riješiti značajan postotak sudara kod svih teških vozila. Ostale aplikacije za operatore komercijalnih vozila će se također ocjenjivati

8 – razviti tranzitne sigurnosne aplikacije. Koristeći rad od automobilskih sigurnosnih aplikacija i prenositi primjenjivost na tranzitna vozila može pozitivno utjecati na industriju

V2V komunikacija se očekuje da će biti učinkovitija od postojećih tehnologija automobilskih proizvođača originalne opreme (OEM) kao što su:

- Ugrađeni sustavi za napuštanje traka ili prestrojavanje u/iz trake
- Sinkronizacija brzina sa drugim vozilima
- Adaptivni tempomat
- Senzor i kamera za parkiranje unatrag (V2V tehnologija omogućuje 360 stupnjeva svijesti u okolini vozila i mogućim prijetnjama) [12]



Slika 7. Prikaz komunikacije vozilo s vozilom

Na slici 7. vidimo primjer sinkronizacije brzina za više vozila. Prvo vozilo (na slici 7.) se vozi nekakvom određenom brzinom, no vozila iza njega sinkroniziraju brzine s obzirom na prvo vozilo. Ukoliko prvo vozilo poveća brzinu ili smanji brzinu, vozila iza njega će biti upozorena na vrijeme.

3.2.2 V2I komunikacija

Vozilo s infrastrukturom (V2I) je komunikacija za sigurnost preko bežične razmjene kritičke sigurnosti i operativnih podataka između vozila i cestovne infrastrukture, prvenstveno kako bi se izbjeglo što više nesreća.

V2I komunikacija za sigurnost je ključni istraživački program za ITS JPO programa u U.S. DOT RITA

Vizija za V2I istraživanja je omogućiti dizajniranje sigurnosnih aplikacija za izbjegavanje ili ublažavanje nesreća, osobito za scenarije nesreća koji nije predviđen za V2I komunikaciju. Drugi važan cilj V2I istraživanja je nacionalna interoperabilnost za potporu implementacije infrastrukture i vozila.

Četiri glavna cilja V2I programa istraživanja tehničke sigurnosti su :

- Razviti V2I aktivne sigurnosne zahtjeve koji se bave nekim najkritičnijim scenarijima nesreća, uključujući aplikacije koje koriste fazu semafora i vremena SPaT (engl. signal phase and timing) i te podatci se šalju na vozila putem bežične mreže
- Razviti strogu procjenu sigurnosnih prednosti koje će doprinijeti procjeni mogućih sigurnosnih preporuka i/ili uputa za praktičare
- Osigurati objektivne podatke informacije koje će podržati donošenje odluka između praktičara u pogledu implementacije infrastrukture
- Osigurati odgovarajuće implementirane strategije za privatnost, sigurnost i certifikaciju sustava, interoperabilnost, skalabilnost, upravljačke strukture, javno prihvaćanje i održivo tržište za učinkovito pokretanje i održavanje infrastrukturnih komponenti

U istraživanje V2I komunikacije za program sigurnosti uključuje višestruke transportne agencije i načine. Usredotočena je na ključ RITA, FTA, FMCSA i FHWA područja interesa aplikacije, uključujući sigurnost u raskrižju, upravljanje brzine, sigurnost pješaka, operacije tranzitnih i komercijalnih vozila

Dodatni ciljevi su:

- Uvesti napredne V2I bežične tehnologije za smanjenje, ublažavanje i sprječavanje dodatnih 12% scenarija nesreća koji ne mogu biti predviđeni V2V programom
- Razviti signale upozorenja koji podržavaju aktivnu sigurnost

Istraživački program :

1 – identificirati i analizirati kritične scenarije sudara za V2I aplikaciju. Preliminarne studije pokazuju da bi se dodatnih 12% sudara moglo riješiti V2I sigurnosnim primjenama

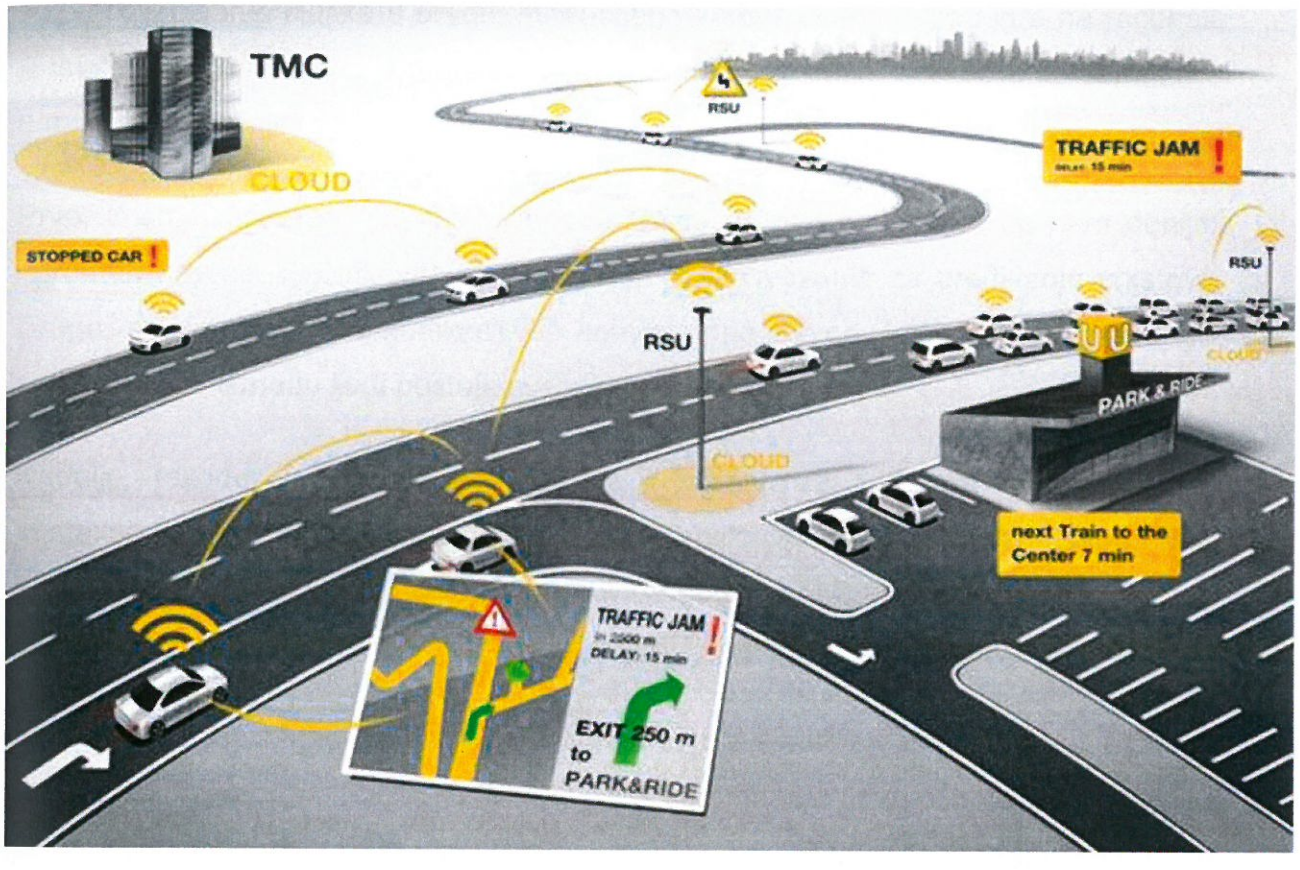
2 – razviti prototip aplikacije koje uključuju identifikaciju i profinjenost zahtjevima i protumjere. Dodatni napori u ovoj točci će biti kooperativno istraživanje i razvoj jedne sigurnosne aplikacije u partnerstvu Europske Unije

3 – adrese infrastrukturne komunikacije i elementi interoperabilnosti koji omogućavaju razmjenu informacija između više sustava i komponenata kao sto je mapiranje, pozicioniranje, standardi, sigurnosti i bežične komunikacije

4 – kroz kontrolirane demonstracije testna operacijska područja koristi procjene prikupljanjem i analizom stvarnih podataka, dok za procjenu inženjerstvo, arhitekturu i dizajn. Te će analize rezultirati boljem razumijevanju tržišnog potencijala, infrastrukturnih zahtjeva i razinu tržišne penetracije potrebne za omogućavanje buduće V2I sigurnosne aplikacije

5 – provesti planiranje implementacije preko razvijenih alata i smjernice koje će dati potrebne informacije praktikantima za donošenje dobrih odluka za održavanje i implementaciju samog V2I sustava. Na kraju, rezultati V2I sigurnog istraživačkog

programa će razviti sigurnosne aplikacije koje su u mogućnosti podržavati komunikaciju infrastrukture i vozila [13]



Slika 8. Prikaz komunikacije vozilo s infrastrukturom

Na slici 8. vidimo primjer u kojoj se stvorila prometna gužva i povodom toga vozila šalju informacije infrastrukturi koja šalje informacije nadolazećim vozilima dovoljno rano da bez opasnosti mogu skrenuti u alternativnu rutu da se izbjegne prometna gužva.

4. Komparativna analiza programskih jezika

Softver je proširio mnoga područja suvremenog života. Ljudi bi mogli koristiti softver za pisanje članka u uredu, međusobno komuniciranje, igrati igre na računalu ili telefonu za opuštanje. U prethodnim godinama softverske aplikacije su dominirale u mnogim područjima našeg života.

Prvo, imamo office softver kao Microsoft Office i Open Office koji su nam donijeli ogroman napredak učinkovitosti istraživanja i rada vezanih za uređivanje tekstova. Drugo, sa primjenom komunikacijskih softvera kao Skype, MSN, Yahoo Message, komunikacija između ljudi postaje sve više prigodna.

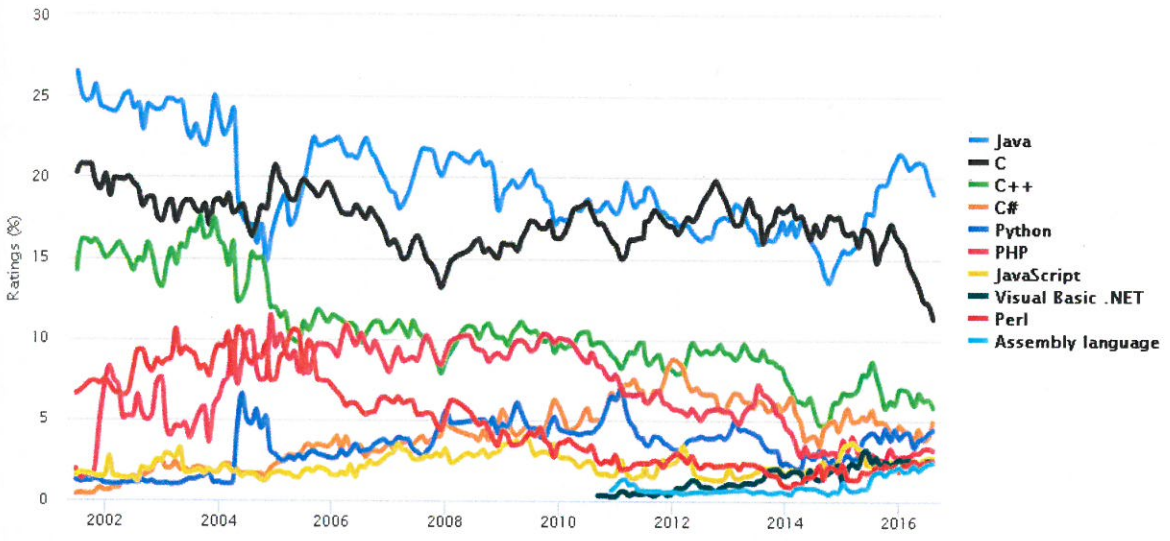
Nadalje, također je postala sve jeftinija i jeftinija za ljude da se međusobno kontaktiraju. Na primjer, u sadašnjosti postoje aplikacije kao „Viber, Whatsapp,Fring" preko kojih ljudi mogu kontaktirati druge ljude putem bežične mreže ili putem 3G ILI 4G mreže.

Jasno je da je javnost iskusila moć softvera. I svi softveri u podlozi su napisani programskim jezicima. Razvoj u tehnologiji programiranja i razvoj programskih jezika je veoma popularno u današnjici i sve više se razvija. Najnovija generacija programskih jezika ima za cilj da računalo može riješiti bilo kakav problem bez prisustva programera.

Paralelno programiranje postaje vodeći način programiranja u budućnosti za razvoj softvera. Uz ograničenja o tome koliko brzo jedan procesor može izračunati, zbog toga dolazi do razvoja procesora sa više jezgri.

Glavni razlog prestanka razvoja procesora sa jednom jezgrom je količina energije potrebna za povećanje brzine frekvencije procesora. Današnjim kompjutorima sa procesorima s više jezgri je vrlo malo vremena i malo energije potrebno za izvršavanje programa. [14]

Na slici 1. je prikazano prema TIOBE Programming Community index uvid i prikaz popularnosti i korištenosti programskih jezika od 2002.godine, pa do današnje 2016. godine.



Grafikon 1. Prikaz popularnosti i korištenosti programski jezika od 2002 - 2016.godine

Na slici 1. je prikazano prema TIOBE (engl. Programming Community indeks) uvid i prikaz popularnosti i korištenosti programskih jezika od 2002.godine, pa do današnje 2016. godine.

TIOBE je pokazatelj popularnosti programskih jezika kroz godine. Indeks se ažurira jedanput mjesečno. Ocjene (engl. Raitngs) kao na grafikonu, ovise o broju kvalificiranih inženjera širom svijeta. Popularne Internet tražilice kao Google, Bing, Yahoo, Wikipedia, Amazon, i druge se koriste za izračunavanje ocjena.

Važno je napomenuti da indeks TIOBE ne ovisi o najboljem programskom jeziku ili jeziku kojem je potrebno najviše linija koda. [16]

4.1 Povijest C, C++, C# i Java programskog jezika

Povijest C programskog jezika

C programski jezik je rođen u SAD-u od strane „AT & T Bell Laboratories" 1972. godine. Napisao ga je Dennis Ritchie. Ovaj jezik je stvoren za specifičnu svrhu: da dizajnira UNIX operativni sustav (koji se koristi na mnogim računalima).

Od samog početka, C je namijenjen da bude koristan kako bi se omogućilo programerima da obave posao. Nakon toga, C se počeo koristiti sve više i više izvan „Bell Laboratories" jer je bio učinkovitiji od drugih programskih jezika u to vrijeme. U kasnim 1970., C je zauzeo dominantnu poziciju među programskim jezicima.

Odbor formiran od strane „American National Standards Institute (ANSI)" odobrio je 1989. godine verziju C-a koja je poznata kao ANSI C. Uz nekoliko iznimaka, svaki moderan C programski prevodilac ima sposobnost pridržavanja ovog standarda. ANSI C je tada odobren od strane International ISO (engl. Standards Organization) u 1990.godini.

```
/* Hello World program */

#include<stdio.h>

main()
{
    printf("Hello World");


}
```

Slika 9. Primjer „Hello world" koda u C programskom jeziku

Povijest C++ programskog jezika

C++ je napisao Bjarne Stroustrup u „AT & T Bell Laboratories" tijekom 1983.-1985.godine. C++ je proširenje programskog jezika C. Bjarne Stroustrup je dodao značajke u C i formirao nešto što je nazvao „C with Classes".Skombinirao je klase i objektno-orijentirane značajke sa snagom i učinkovitošću C jezika. Izraz C++ je prvi puta korišten 1983.godine.

```
#include <iostream.h>

main()
{
    cout << "Hello World!";
    return 0;
}
```

Slika 10. Primjer „Hello world" koda u C++ programskom jeziku

Povijest C# programskog jezika

Primarni arhitekti C# jezika su Peter Golde, Eric Gunnerson, Anders Hejlsberg, Peter Sollichy i Scott Wiltamuth. Naravno, principijalni dizajner C# jezika je Anders Hejlsberg, vodeći arhitekt u Microsoft-u.

C# je dizajniran da bude čisti objektno-orijentirani programski jezik. C# je 2000.godine debitirao u Professional PDC (engl. Developers Conference) gdje je osnivač Microsoft-a Bill Gates bio glavni govornik.

```
public class Hello1
{
   public static void Main()
   {
      System.Console.WriteLine("Hello, World!");
   }
}
```

Slika 11. Primjer „Hello World" koda u C# programskom jeziku

Povijest Java programskog jezika

Java se počela razvijati u 1991.godini od strane James Gosling-a iz „Sun Microsystems" i njegov tim. Originalna verzija Jave je namijenjena za programiranje kućanskih aparata.

1994., James Gosling je počeo razvijati konekciju između Jave i Interneta. 1995., Netscape incorporated je objavio najnoviju verziju Netscape preglednika koji je bio sposoban podržavati Java programe.

Izvorni naziv Jave je hrast (engl. Oak), no moralo se promijeniti naziv jer je hrast naziv već bio korišten za drugi programski jezik. [14]

```
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!"); // Display the string.
    }
}
```

Slika 12. Primjer „Hello world" koda u Java programskom jeziku

4.2 Teorijska pozadina i paralelno programiranje

U moru programskih jezika postoje razna grupiranja i kategoriziranja različitih programskih jezika.

Glavne tri grupe su :

- Programski jezici niske razine
- Programski jezici srednje razine
- Programski jezici visoke razine

Programski jezici niske razine često imaju vrhunske preformanse i pružaju malo ili ništa apstrakcije programskih koncepata. To znači da programer ima potpunu kontrolu nad time što se događa i u mogućnosti je dodati dovoljno sredstva za rješenje problema, ali zbog problema apstrakcije to može biti prilično komplicirano zbog toga što se teško vidi tok programa i zbog toga programer mora znati što radi. Neki primjeri programskih jezika niske razine su Assembly i C.

Programski jezici srednje razine imaju višu razinu apstrakcije i djeluju kao sloj na vrhu programskih jezika niske razine. Zato oni često dolaze zajedno sa virtualnom mašinom (engl. Virtual Machine) koji djeluje kao prvi korak za prvo kompajliranje koda srednje razine na kod niske razine prije nego što je preveden u kod mašine. Viša razina apstrakcije čini korištenje objekata mogućim što čini kod puno lakšim za pratiti tok programa i programer se više ne mora zamarati sa registrima i raspodjeli memorije. Loša strana je da programer gubi preciznost određivanja programskih jezika niže razine. Neki primjeri programski jezika srednje razine su Java i C#.

Programerski jezici više razine su dovedeni na ekstremnu razinu apstrakcije. Oni su, dakle, vrlo dinamični i daju programeru veliku količinu fleksibilnosti u dizajniranju algoritama što programski jezici niže i srednje razine ne mogu pružiti. Zbog takve fleksibilnosti, može biti poprilično teško pratiti tok programa i zbog brojnih slojeva apstrakcije nekoliko uputa se može prevesti u tisuće strojnih riječi, što je loše. Neki primjeri programskih jezika više razine su Ruby i Perl.

Za razliku od tradicionalnog razvoja softvera sa sekvencijalnim programiranjem, paralelno programiranje uvodi nove izazove i svojstva.

Postoje različiti načini za vidjeti paralelno programiranje, bilo fokusirano na strukturalni model ili na paralelnu implementaciju koji uključuju kakva je vrsta ili paralelni model, na primjer

Postoje tri tipa klasifikacije u paralelnom programiranju ako se gleda na paralelni program :

- Sitno-zrnatost (fine-grained)
- Krupno-zrnatost (coarse-grained)
- Neugodni paralelizam (embarrassing parallelism)

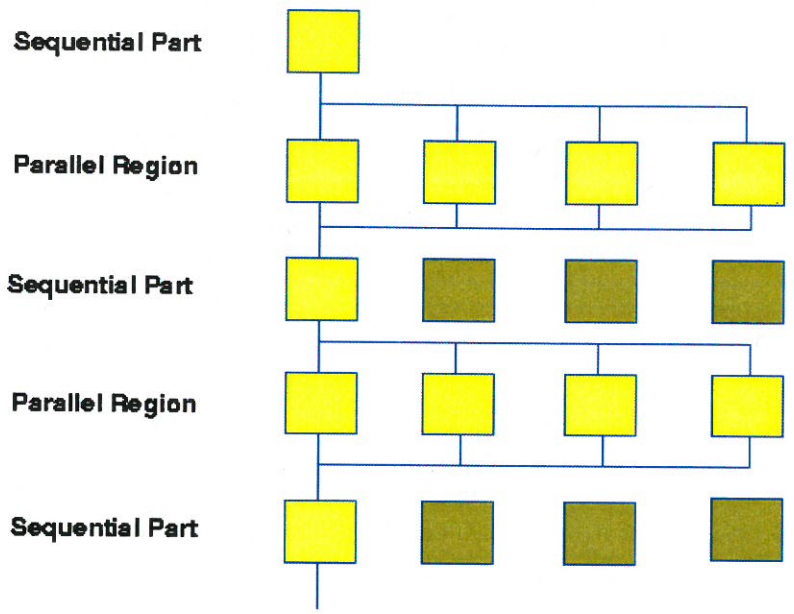
Sitno-zrnatost je kada se paralelni pod-zadaci programa trebaju konstantno usklađivati i komunicirati jedni s drugima dovoljno često da bi funkcionirali pravilno.

Krupno-zrnatost je slično kao sitno-zrnatost, ali se ne treba usklađivati i komunicirati tako često.

Neugodni paralelizam rijetko kada se mora usklađivati i komunicirati jedan s drugim.

Ove klasifikacije daju dosta informacija kako će se program sa paralelnom implementacijom ponašati.

Pod-zadaci, gore navedeni se također nazivaju dretve (engl. Threads) i imaju različite vrste sinkronizacije i metode komuniciranja koji se uzajamno isključuju i pauziraju. Međusobno isključivanje se obično koristi da bi se jedna dretva zaključala i obavila sigurno kritičnu sekciju bez ometanja drugih dretvi. Pauziranje se obično koristi kada dretve trebaju pričekati druge dretve dok ne završe da bi one počele ili ako dretva treba rezultat od druge dretve, pa zbog toga mora pričekati da prva dretva završi. Ova svojstva paralelno programiranje čine i jednostavnim i složenim. [15][17]



Slika 13. Prikaz modela za više procesa

OpenMP – je sučelje za programiranje aplikacija (API) koji podržava više procesnu memoriju s više platformi

Izvršenje programa uvijek započinje u serijskom načinu rada. Na slici 2. vidimo da čim se dođe do prve paralelne regije, skupina dretvi se formira ovisno o korisničkim zahtjevima (4 dretve na slici) i svaka dretva izvršava kod u svojoj zatvorenoj paralelnoj regiji. Na kraju paralelne regije sve dretve se spajaju. Dretve su završene tek kada je sam program gotov.

Svojstva paralelnih algoritama

- definiramo poželjna svojstva koja bi paralelni algoritmi trebali imati:
  - istodobnost (engl. concurrency) – mogućnost izvođenja više radnji istovremeno nužno za razvoj algoritama
  - skalabilnost (engl. scalability) – mogućnost prilagođavanja proizvoljnom broju fizičkih procesora (odnosno mogućnost iskorištavanja dodatnog broja računala)

- lokalnost (engl. locality) – veći omjer lokalnog u odnosu na udaljeni pristup u memoriji – korištenje lokalne ili priručne memorije (engl. cache)
- modularnost (engl. modularity) – mogućnost uporabe dijelova algoritama unutar različitih programa [18]

4.3 Usporedba C, C++, C# I Java programskih jezika

4.3.1 Tipovi Podataka i veličine

U C, C++, C# I Java, sve varijable moraju biti deklarirane prije nego što su korištene, obično na početku funkcije prije bilo kakvog izvršenja.

Neke uobičajene vrste tipova podataka koje se koriste u programskim jezicima su nazvane primitivnim tipovima kao znakovi, brojevi, brojevi sa plivajućim zarezom itd.

C

U C jeziku postoje 4 osnovna tipa i 5 tipova specifikatora:
Osnovni:

- Int
- Float
- Double
- Char

Specifikatori:

- Long
- Long long
- Short
- Unsigned
- Signed

Tablica 2. Osnovni tipovi, primjeri i kvalifikatori u C programskom jeziku

Type	Constant Examples	printf chars
char	'a', '\n'	%c
_Bool	0, 1	%i, %u
short int	—	%hi, %hx, %ho
unsigned short int	—	%hu, %hx, %ho
int	12, -97, 0XFFE0, 0177	%i, %x, %o
unsigned int	12u, 100U, 0XFFu	%u, %x, %o
long int	12L, -2001, 0xffffL	%li, %lx, %lo
unsigned long int	12UL, 100ul, 0xffeeUL	%lu, %lx, %lo
long long int	0xe5e5e5e5LL, 50011	%lli, %llx, &llo
unsigned long long int	12ull, 0xffeeULL	%llu, %llx, %llo
float	12.34f, 3.1e-5f, 0x1.5p10, 0x1P-1	%f, %e, %g, %a
double	12.34, 3.1e-5, 0x.1p3	%f, %e, %g, %a
long double	12.341, 3.1e-51	%Lf, $Le, %Lg

Tablica 2. prikazuje sve rezervirane riječi u C jeziku. Ove riječi se ne mogu skratiti, koristiti kao imena varijabli ili kao bilo koji drugi tip identifikatora.

Tablica 3. Rezervirane riječi u C programskom jeziku

auto	else	long	switch
break	enum	register	typedef
case	extern	return	union
char	float	short	unsigned
const	for	signed	void
continue	goto	sizeof	volatile
default	if	static	while
do	int	struct	_Packed
double			

U C jeziku, svaki tip podataka kao što je znak, broj ili broj s plivajućim zarezom ima raspon vrijednosti povezanih s njima. Raspon se odlučuje preko količine prostora za pohranu određene vrste podataka u memoriji računala. Na primjer, broj bi mogao zauzeti 32 bita na računalu ili bi mogao biti spremljen kao 64 bita na drugom računalu. Nemoj te pisati bilo koji program koji uzima veličinu tipova podataka U C-u.

C++

U C++, osnovni tipovi podataka su skoro isti kao i u C jeziku.

Tablica 4. Osnovni tipovi podataka i kvalifikatori u C++ jeziku

Type	*Size*	*Values*
`bool`	1 byte	true or false
`unsigned short int`	2 bytes	0 to 65,535
`short int`	2 bytes	-32,768 to 32,767
`unsigned long int`	4 bytes	0 to 4,294,967,295
`long int`	4 bytes	-2,147,483,648 to 2,147,483,647
`int` (16 bit)	2 bytes	-32,768 to 32,767
`int` (32 bit)	4 bytes	-2,147,483,648 to 2,147,483,647
`unsigned int` (16 bit)	2 bytes	0 to 65,535
`unsigned int` (32 bit)	4 bytes	0 to 4,294,967,295
`char`	1 byte	256 character values
`float`	4 bytes	1.2e-38 to 3.4e38
`double`	8 bytes	2.2e-308 to 1.8e308

U C++, postoje više logičkih tipova. logičkih, odnosno bool ima dvije vrijednosti istina ili laž. logičkih je korišten za izražavanje rezultata kod logičkih operacija.

Tablica 5. Rezervirane riječi u C++ jeziku

auto	else	long	switch
break	enum	register	typedef
case	extern	return	union
char	float	short	unsigned
const	for	signed	void
continue	goto	sizeof	volatile
default	if	static	while
do	int	struct	double

Iz tablice 5 vidimo da C++ ima iste rezervirane riječi kao i C. Jedino riječ _Packed se više ne koristi u C++.

Postoje 30 novih riječi koje nisu u C, a C++ ih ima, zato i C++ zovemo nadogradnjom C jezika.

Tablica 6. Nove riječi u C++ jeziku

asm	dynamic_cast	namespace	reinterpret_cast
bool	explicit	new	static_cast
catch	false	operator	template
class	friend	private	this
const_cast	inline	public	throw
delete	mutable	protected	true
try	typeid	typename	using
virtual	wchar_t		

Bez obzira na veliku prihvaćenost programskog jezika C++ postoje mnoge osobe koje su autoriteti na području programiranja kritiziraju same sposobnosti koje su dostupne u programskom jeziku, i mnoge implementacije koje su opisane kao spore, nedovoljne, i siromašno u ponudi standardnih biblioteka rutina, loše izvedbe kako predlošci rade, sakupljane smeća, nedostatka refleksivnih sposobnosti. Zbog dodatnih zahtjeva za standardne biblioteke i na veličinu završnog binarnog programa C++ se koristi manje od programera koji zahtijevaju prijenosne programe koje se rabe u kritičnim aplikacijama

C#

Logički tip u C# ima malu razliku između tog tipa u C++. U C#, logički tip može imati samo dvije vrijednosti istina ili laž. Dok u C++, logički tip može imati i 0 kao vrijednost koja znači laž, a sve suprotno je istina.

Integralni tipovi u C# su također različiti nego kod C i C++ jezika. U C i C++ jeziku, integralni tip je samo jedan tip, ali u C# integralni tip je kategorija tipova. To su brojevi, pozitivni ili negativni i znakovi. Tip znak je Unicode vrsta znakova i definirana je Unicode standardom.

U C#, long tip podataka je veličine 64-bita, dok je u C++ 32-bita.

Tablica 7. Integralni tipovi, veličina i raspon

Type	Size (in bits)	Range
sbyte	8	-128 to 127
byte	8	0 to 255
short	16	-32768 to 32767
ushort	16	0 to 65535
int	32	-2147483648 to 2147483647
uint	32	0 to 4294967295
long	64	-9223372036854775808 to 9223372036854775807
ulong	64	0 to 18446744073709551615
char	16	0 to 65535

Tip s plivajućim zarezom u C# je float ili double, što je isto kao i u C i C++. No, postoji još jedna vrsta podataka za decimalne brojeve koje treba koristiti kod predstavljanja financijskih ili novčanih vrijednosti.

Tablica 8. Float tipovi i Decimal tip

Type	Size (in bits)	precision	Range
float	32	7 digits	1.5 x 10-45 to 3.4 x 1038
double	64	15-16 digits	5.0 x 10-324 to 1.7 x 10308
decimal	128	28-29 decimal places	1.0 x 10-28 to 7.9 x 1028

Postoji mnogo više ključnih riječi u C# nego u C i C++. Ukupno 87 rezerviranih riječi postoje u C#.

Tablica 9. Sve ključne riječi u C#

C# keywords / C# reserved words			
abstract	as	base	bool
break	by [2]	byte	case
catch	char	checked	class
const	continue	decimal	default
delegate	do	double	descending [2]
explicit	event	extern	else
enum	false	finally	fixed
float	for	foreach	from [2]
goto	group [2]	if	implicit

in	int	interface	internal
into [2]	is	lock	long
new	null	namespace	object
operator	out	override	orderby [2]
params	private	protected	public
readonly	ref	return	switch
struct	sbyte	sealed	short
sizeof	stackalloc	static	string
select [2]	this	throw	true
try	typeof	uint	ulong
unchecked	unsafe	ushort	using
var [2]	virtual	volatile	void
while	where [2]	yield [1]	

[1,2] To nisu zapravo prave ključne riječi, naime moguće je definirati varijable i tipove koristeći ta imena, dok se prave ne mogu, ali se ponašaju kao ključne riječi u novijim verzijama C# - 2.0(1) i 3.0(2).

Sintaksa C# jezika jeste slična onima iz ostalih C-stilskih jezika kao što su C, C++ i Java. U biti:

- Točka-zarez se koristi da se opiše kraj tvrdnje
- Zaobljene zagrade se koriste da grupiraju tvrdnje
- Varijable su pridružene korištenje znaka jednakosti
- Oštre zagrade se koriste sa nizovima, da bi se deklarirali i da bi se uzela vrijednost danog indeksa od jednog od njih

Java

Logički tip u Javi je isti kao i u C#. Postoje samo dvije vrijednosti, a to su istina ili laž.

Integralni tipovi u Javi se također razlikuju od C i C++, ali su zato slični C#. U C# integralni tip je kategorija koja sadrži 9 tipova podataka, dok u Javi postoje 5 tipova podataka. Kod Jave više ne postoje sbyte, uint, ulong tipovi podataka.

Tablica 10. Java primitivni tipovi podataka

Data Type	Description	Size	Default Value
boolean	true or false	1-bit	false
char	Unicode Character	16-bit	\u0000
byte	Signed Integer	8-bit	(byte) 0
short	Signed Integer	16-bit	(short) 0
int	Signed Integer	32-bit	0
long	Signed Integer	64-bit	0L
float	Real number	32-bit	0.0f
double	Real number	64-bit	0.0d

Tablica 11. Float i double

Type	Size		Range	Precision
name	bytes	bits	approximate	in decimal digits
float	4	32	+/- 3.4 * 1038	6-7
double	8	64	+/- 1.8 * 10308	15

Java ima ukupno 50 ključnih riječi. Neke ključne riječi se pojavljuju samo u Javi, a to su na primjer: synchronized, instanceof.

Tablica 12. Ključne riječi u Javi

abstract	continue	For	new
switch	assert	default	goto
package	synchronized	boolean	do
if	private	this	break
double	implements	protected	throw
byte	else	import	public
throws	case	enum	instanceof
return	transient	catch	extends
int	short	try	char
final	interface	static	void
class	finally	long	
volatile	const	float	native
super	while		

Iako je Java nastala od C++, ali za razliku od C++, koji dozvoljava strukturirane kao i objektno orijentirane principe, u Javi su objektno orijentirani principi obavezni. Sve je u Javi objekt, a sav izvorni kod je pisan unutar klasa. [14]

4.4 Benchmarking

Benchmarking je proces usporedbe dva ili više sličnih proizvoda, programa, metoda ili strategija da bi se odredilo koji od njih ima najbolje preformanse.

Usporedba se vrši pokretanjem komparativne analize programa koji mjere različite aspekte preformansi.

Ciljevi usporedba su kako bi se utvrdilo gdje su poboljšanja i analizirati kako ostale organizacije postižu visoke razine preformanse i koristiti te informacije da bi se preformanse više poboljšale.

Kada se uspoređuju drugačiji programski jezici, odabrani programi za usporedbu su kodirani u svakom odgovarajućem jeziku i onda ga pokrenuti na istom stroju pod istim uvjetima. Obično ti programi usporedbe izvode teške proračune ili procese velikih količina podataka. Preformanse se najčešće mjere u izvršenom vremenu ili broju operacija.

4.4.1 SparseLU - Sparse Linear Algebra

SparseLU program usporedbe koristi rijetku linearnu algebru za izračun LU(engl. lower-upper) faktorizaciju rijetke matrice. Rijetke matrice su implementirane kao block matrica veličine 50 x 50 blokova s veličinom bloka 100 x 100 jedinica umjesto jedne relativno velike 5000 x 5000 matrice.

Rijetka matrica je matrica koja uglavnom sadrži nule u tablici. To je korišteno u znanosti ili inženjerstvu pri rješavanju parcijalnih diferencijalnih jednadžbi, a također se primjenjuje u područjima koja imaju malo značajnih podataka ili konekcija.

LU faktorizacija faktorira matricu kao proizvod niže i više trokutaste matrice. Proizvod ponekad uključuje permutaciju matrice također. [15]

4.4.2 Linije koda LOC (engl. lines of codes)

Tablica 12. Prikazuje finalni broj linija u kodu nakon usporedbe (isti zadatak za sve programske jezike)

Tablica 13. Broj linija u kodu na kraju

Language	Final size in LOC
C	256
C++ with TBB	250
C#	415
Java	381

C++ sa TBB – (engl. Threading building blocks) (Intelova tehnologija za mogućnost pisanja paralelnih C++ programa koji mogu koristiti punu snagu više-jezgrenih preformansi)

C# je trebao dodatne linije koda za kontrolu broja zadataka u isto vrijeme, zbog nedostatka kontrole broja dretvi.

Java može kontrolirati broj korištenih dretvi, ali je potrebno imati duplicirane funkcije za serijske i paralelne verzije. To je zbog toga što su funkcije morale biti u svojoj vlastitoj klasi. [14]

5. Implementacija sustava potpore eko-vožnje

5.1 OBD

Od 1998 god. standardizirano je komunikacijsko sučelje sa vozilima pod nazivom OBD (eng. On Board Diagnostic).

Slika 14. OBD uređaj

Ovo sučelje prvenstveno je namijenjeno aktivnostima dijagnostike kvarova na vozilima. Pored navedenog na tržištu su se pojavili tzv. OBD dongle-ovi koji omogućavaju razmjenu informacija sa različitim vanjskim uređajima, kao što su npr. aplikacije za pametne telefone. Također, OBD dongle-ovi svoju primjenu su pronašli za potreba naprednog praćenja flote vozila. Pomoću takvih uređaja moguće je pratiti rad vozila te nadzirati istovremeno većinu parametara uz funkcionalnosti pridruživanja tih istih parametara i satelitskog pozicioniranja vozila. Navedene funkcionalnosti OBD dongle-ova moguće je dodatno proširiti mobilnom komunikacijskom mrežom. Pojavom kooperativnih inteligentnih transportnih sustava i njihovom integracijom sa sustavima potpore eko-vožnje nameće se kao neizbježan scenarij za budućnost. Dakle, gotovo pa većina današnjih automobila na cestama posjeduje ovakav priključak, a broj modela i proizvođača automobila koji ga koriste broji se u tisućama. [19]

5.2 ELM 327

ELM327 predstavlja sučelje koji omogućava komunikaciju drugih uređaja sa automobilom korištenjem standardiziranog OBD2 priključka koji se u većini automobilima nalazi još od kraja 90-ih godina.

Slika 15. ELM327 adapter

S tehničke strane, ELM327 je PIC18F2480 mikrokontroler sa programom koji otprilike funkcionira tako da sa računala zaprimi komandu i zatim je pretvori u oblik koji je propisan OBD2 standardom i pošalje prema vozilu putem K i L komunikacijske linije ili CAN sabirnice, ovisno o tome koja se podvrsta OBD2 protokola koristi (inače, ELM327 IC podržava 12 podvrsta komunikacijskih protokola).

Prevedeno, ništa drugo nego omogućava drugim uređajima da iskoriste ovaj priključak (OBD2) i ostvare komunikaciju sa vozilom. Osim što će se povezati sa vozilom, ELM327 priključak će se pomoću bežične Bluetooth veze povezati sa drugim uređajem i biti svojevrsni posrednik između vozila i drugog uređaja. Ovaj drugi uređaja zapravo može biti bilo koji uređaj koji ima mogućnost pokrenuti kompatibilnu aplikaciju i povezati se pomoću Bluetooth veze. Takvih uređaja je puno, a generalno, najčešće su to pametni telefoni, tableti, računala i prijenosnici, ali i neki drugi, specijalizirani uređaji za ovu namjenu.

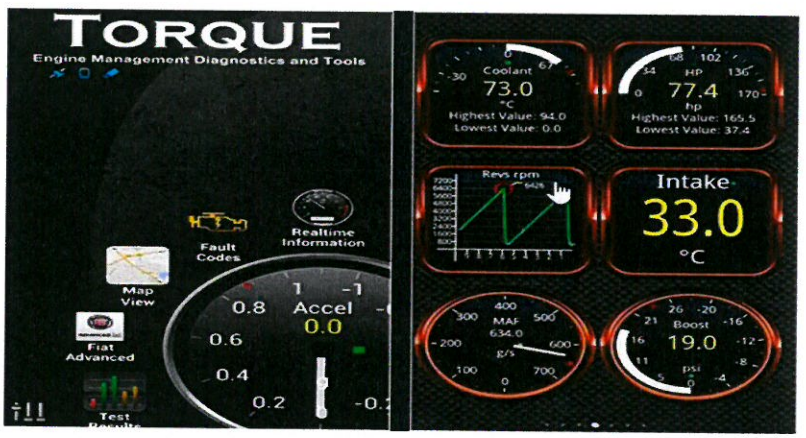
Za Android uređaje biti će dovoljan ELM327 sučelje u Bluetooth izvedbi. Kod IOS uređaja (iPhone, iPad) spajanje na Bluetooth vanjske uređaje je ograničeno zbog restrikcija operativnog sustava pa će za rad biti potrebna izvedba ELM327 uređaja koja komunicira putem Wi-Fi mreže. [20][21]

5.3 Mobilne aplikacije za potporu eko-vožnje

Većina smartphone aplikacija za potporu eko-vožnji koriste inerciometar. Aplikacije memoriraju izmjerene vrijednosti inerciometra, te prikazuju na način da je vozilo previše ubrzavalo, prenaglo se zaustavljalo ili kretalo prevelikom brzinom zavojima. To su jako bitne informacije prilikom vožnje za koju vozači nisu svjesni kako utječu na sigurnost prometa, potrošnju goriva i količinu emisije ispušnih plinova.

Ovakve aplikacije omogućavaju smanjenje emisije CO2 i smanjenje rizika u prometu, ali postoje nedostaci u korištenju same aplikacije tokom vožnje. Najveći nedostatak se uviđa prilikom korištenja aplikacije i telefoniranja.

Jedna od najpoznatijih aplikacija potpore Eko-vožnji za smartphone uređaja na Android platformi je „Torque".



Slika 16. Torque sučelje

Slika 16. Prikazuje Torque sučelje (desno) za očitanje informacija prilikom vožnje i glavni meni (lijevo). Ponuda informacija Torque aplikacije vozaču je širokog spektra. Na slici je prikazan (desno) termometar za očitanje temperature motora, koliko konjskih snaga se koristi tijekom vožnje, temperaturu okoline, broj okretaja motora i akceleraciju. Torque sučelje se može uređivati po volji.

Postoje dvije verzije, a to su: Torque Lite i Torque Pro, s time da Lite je besplatna verzija, a Pro nije

Torque mobilna aplikacija je dijagnostički alat za bilo koji uređaj koji ima Android operativni sustav. Omogućuje pristup mnogim senzorima ugrađenim u sustav upravljanja vozila, i također omogućuje vozačima pregled neispravnosti. Povezuje se preko OBD Bluetooth adaptera.

Ima mogućnost dijagnostike grešaka motora i praćenja raznovrsnih podataka koji se mogu prikazati u grafičkom ili brojčanom obliku. Uz upaljen GPS Torque će pratiti vožnju i ponuditi kasniju analizu puta. Moguće je i snimati vožnju kamerom telefona, a Torque će na snimak dodati željene podatke kao što su brzina, broj okretaja, potrošnja i sl. [22][23]

6. Zaključak

Svakim danom sve više i više vozila prometuje cestovnim prometnicama i štetni ispušni plinovi postaju sve veći problem za okoliš. Autoindustrije razvijaju razne elektroničke sklopove kako bi se smanjila emisija i to veoma uspješno. U zadnjih par godina vozilima se smanjuje potrošnja goriva s najjednostavnijim dijelovima kao što su auto gume, pa sve do sofisticiranijih rješenja poput start-stop sustava i slično.

Problem nastaje u tome što će se emisije kod motora na unutarnje sagorijevanje moći samo smanjivati, no ne i ukloniti. Potencijalno rješenje (već u primjeni) su električna vozila koja nemaju zagađenja, ne stvaraju buku i troškovi su drastično manji. Očekuje se nagli porast primjene električnih vozila kroz nekoliko godina.

Primjenom sustava koji su integriranim u vozilo, naknadno ugrađeni ili aplikacijama za smartphone uređaje moguće je znatno smanjiti emisiju štetnih ispušnih plinova, potrošnju goriva i povećati pomoć vozaču. Takvi jednostavni sustavi informiraju vozača kakve je pogreške učinio prilikom vožnje te kako drugačije pristupiti problemu.

Na temelju spoznaja o doprinosu smartphone aplikacija, integriranih sustava te naknadno ugrađenih sustava došlo se je do zaključka da eko vožnja i sustavi potpore eko vožnji imaju znatan istraživački potencijal kako bi se implementirali sustavi koji ne samo da mogu promatrati eko način vožnje jednog vozila, nego putem kooperativnih sustava optimizirati vožnju više vozila koja su u međusobnoj interakciji.

Takvi sustavi zahtijevaju i određene tehnologije koje će se moći koristiti za razvijanje tih sustava. Programiranje aplikacija postaje veoma popularno u današnjici i potraga za novim inovativnim i kreativnim aplikacijama je u rastu. Na današnjem tržištu, Java se koristi široko i dosljedno tamo gdje preteže brzina razvoja programskog sustava nad zahtjevima do brzine rada programa. Java pruža bolji stupanj sigurnosti i pouzdanosti zahvaljujući VM-u i hermetički zatvorenom okolišu u kome svaki program operira, a dok na Javi se program brže razvija s manje pogrešaka.

Već postoji tisuće aplikacija pisanih u Javi i koje su pridonijele korisnicima zadovoljstvo i nove inovacije. Bitna činjenica je da su ovo tek nove ideje i da aplikacijama nije kraj.

Popis literature

[1] https://www.smead.com/Director.aspx?NodeId=2140

[2] https://energypedia.info/wiki/Eco_Driving

[3] http://www.cepta.sk/index.php/en/clean-air-ciste-ovzdusie-projects-736/545-zasady-ekosoferovania

[4] https://en.wikipedia.org/wiki/Energy-efficient_driving#Choice_of_gear_.28manual_transmissions.29

[5] https://en.wikipedia.org/wiki/Brake_specific_fuel_consumption

[6] https://www.pca.state.mn.us/living-green/check-your-tire-pressure-reduce-pollution

[7] http://ecomodder.com/forum/showthread.php/mythbusters-tests-tyre-tire-pressure-17151.html

[8] H. Vojvodić: Sustavi potpore eko-vožnji, Završni rad, 2013

[9] https://www.sigmobile.org/mobicom/2015/papers/p358-kangA.pdf

[10] H. Vojvodić: Analiza i značajke kooperativnih sustava, Završni rad, 2013.

[11] Alexander Paier, Refi-Tugrul Güner, Wolfgang Brückler - V2X COOPERATIVE SYSTEMS – ON THE WAY TO NEXT GENERATION ITS

[12] http://www.its.dot.gov/factsheets/v2v_factsheet.htm

[13] http://www.its.dot.gov/factsheets/v2isafety_factsheet.htm

[14] http://publications.theseus.fi/bitstream/handle/10024/16995/Chen_Hao.pdf

[15] http://kth.diva-portal.org/smash/get/diva2:648395/FULLTEXT01.pdf

[16] http://www.tiobe.com/tiobe-index/

[17]http://www.tutorialspoint.com/parallel_algorithm/parallel_algorithm_introduction.htm

[18] http://www.lrz.de/services/software/parallel/openmp/

[19] https://en.wikipedia.org/wiki/On-board_diagnostics

[20] https://en.wikipedia.org/wiki/ELM327

[21] http://www.glas-slavonije.hr/250690/23/Kako-dise-vas-automobil

[22] https://play.google.com/store/apps/details?id=org.prowl.torque&hl=en

[23] http://www.kdijagnostika.hr/shop/elm327-bezicni/

Popis slika

Popis tablica

Popis grafikona

METAPODACI

Naslov rada: Sustav potpore eko-vožnje za osobna vozila

Student: Massimo Gruičić

Mentor: Dr. sc. Pero Škorput

Naslov na drugom jeziku (engleski):

Eco-Driving Support System for Vehicles

Povjerenstvo za obranu:

- Izv. prof. dr. sc. Tonči Carić, predsjednik
- Dr. sc. Pero Škorput, mentor
- Mr. sc. Goran Jurković, član
- Izv. prof. dr. sc. Sadko Mandžuka, predsjednik zamjena

Ustanova koja je dodijelila akademski stupanj: Fakultet prometnih znanosti Sveučilišta u Zagrebu

Zavod: Zavod za Inteligentne transportne sustave

Vrsta studija: Preddiplomski

Studij: Inteligentnih transportni sustavi (npr. Promet, ITS i logistika, Aeronautika)

Datum obrane završnog rada: 13. rujan 2016.

Napomena: pod datum obrane završnog rada navodi se prvi definirani datum roka obrane.

Sveučilište u Zagrebu
Fakultet prometnih znanosti
10000 Zagreb
Vukelićeva 4

IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOST

Izjavljujem i svojim potpisom potvrđujem kako je ovaj završni 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 završnog rada pod naslovom **Sustav potpore eko-vožnje za osobna vozila**

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

U Zagrebu, 6.9.2016

Student/ica:

(potpis)