

Radarski sistem za pameten nadzor prometa: Zajem, obdelava in prikaz podatkov

Luka Podbregar, Andrej Kos, Urban Sedlar, Aljaž Blatnik, Boštjan Batagelj

Univerza v Ljubljani, Fakulteta za elektrotehniko, Katedra za informacijske in komunikacijske tehnologije

E-pošta: luka.podbregar@fe.uni-lj.si

Radar System for Smart Traffic Control: Data Capturing, Processing and Display

The rapid development of technology presents growing challenges in processing, storing, and visualizing the data generated by various devices and sensors. As system precision improves, the volume of data increases, emphasizing the need for careful system architecture planning and efficient data storage solutions.

Radar systems — renowned for their ability to accurately measure speed and distance — have become indispensable in a wide range of applications, from the automotive industry to meteorology. This paper introduces a radar system developed for smart traffic monitoring, encompassing data acquisition, processing, storage, and visualization through a user-friendly interface. The system collects radar measurements, processes them on a computer, and stores the results in a cloud-based MongoDB database. These measurements are then displayed on a user interface that interacts dynamically with the database, utilizing the Express.js framework to ensure an efficient and responsive user experience.

1 Uvod

S hitrim razvojem tehnologije se soočamo z vedno večjimi izzivi pri obdelavi, shranjevanju in vizualizaciji podatkov, ki jih generirajo različne naprave in senzorji. Povečana natančnost sistemov prinaša večje količine podatkov, kar izpostavlja nujnost skrbnega načrtovanja arhitekture sistemov in učinkovitega hranjenja podatkov.

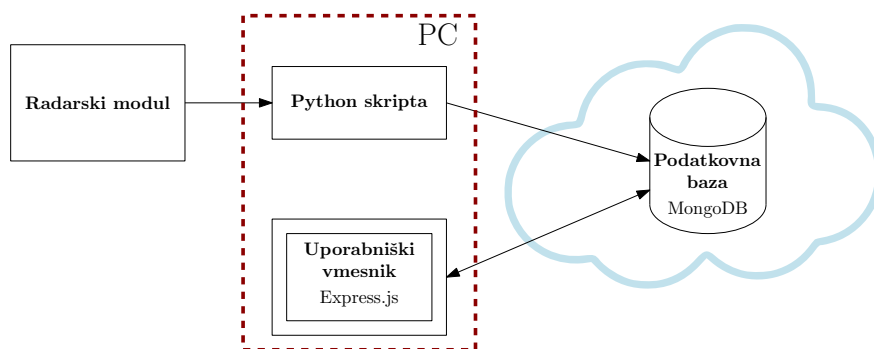
Radarski sistemi, znani po svoji sposobnosti natančnega merjenja hitrosti in razdalje, so postali ključni na številnih področjih uporabe, od avtomobilske industrije do meteorologije. Ta prispevek predstavlja radarski sistem za pameten nadzor prometa, ki združuje zajem, obdelavo in shranjevanje podatkov ter njihov prikaz na intuitivnem uporabniškem vmesniku. Zasnova sistema omogoča zajem radarskih meritev, njihovo obdelavo na računalniku in shranjevanje rezultatov v oblak, pri čemer uporabljamo podatkovno bazo MongoDB. Meritve so nato prikazane na uporabniškem vmesniku, ki dinamično komunicira s podatkovno bazo, z uporabo ogrodja Express.js za učinkovito uporabniško izkušnjo.

2 Arhitektura sistema

Blokovni načrt arhitekture sistema je prikazan na Sliki 1. Sistem je sestavljen iz radarskega modula, Python skripte za zajem podatkov iz radarskega modula, podatkovne baze v oblaku in uporabniškega vmesnika. V nadaljevanju so podrobneje opisani posamezni deli sistema.

2.1 Radarski modul

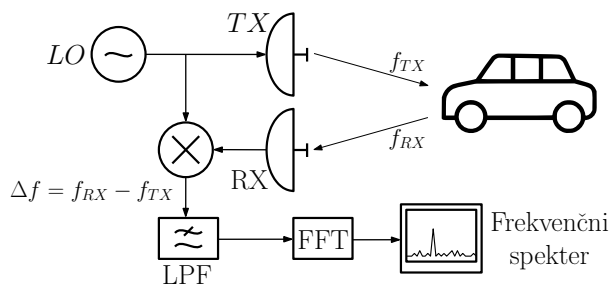
Prvotno vojaški sistemi so se v zadnjih desetletjih tehnološko nadgradili in razširili na številna nova področja [1, 2]. Med njimi so avtomobilski radarji [3], senzorji za zaznavo in preklapljanje luči ali vrat, sistemi za nadzor avtomatizacije [4], radarji za medicinsko uporabo [5], slikanje reliefa [6], zaznavanje vremenskih pojavov [7] in zaznavanje astronomskih objektov [8]. Poseben primer uporabe radarjev je tudi nadzor cestnega prometa [9], kjer



Slika 1: Blokovni načrt arhitekture izvedenega radarskega sistema za pameten nadzor prometa. PC: osebni računalnik.

je merjenje hitrosti vozil ključnega pomena za zagotavljanje varnosti na cestah.

Dopplerjev radar deluje na principu Dopplerjevega pojava, ki je značilen za vsako valovanje, pri katerem se oddajnik ali sprejemnik valovanja premika [10]. V primeru stacionarnega Dopplerjevega radarja je oddajnik stacionaren in meri premikajoče se tarče. Blokovni načrt na sliki 2 prikazuje delovanje preprostega stacionarnega Dopplerjevega radarskega sistema. Če se tarča premika v smeri radarskega sistema, je frekvenca sprejetega signala višja od frekvence oddanega signala. Če se tarča premika stran od radarskega sistema, je frekvenca sprejetega signala nižja od oddane frekvence.



Slika 2: Blokovni načrt delovanja preprostega Dopplerjevega radarja. LO: lokalni oscilator; TX: oddajnik, RX: sprejemnik; LPF: nizko prepustno sito; FFT: hitra Fourierjeva transformacija.

Radarske sisteme delimo glede na obliko radarskega signala, ki je odvisna od želene lastnosti radarskega sistema. Oblika signala se večinoma prilagaja glede na željeno ločljivost izmerjene razdalje in hitrosti. V grobem jih delimo na dve skupini: impulzne radarje in radarje s konstantno oddajo (angl. Continuous-Wave – CW). V obeh primerih lahko radar uporablja signal s konstantno frekvenco ali pa frekvenčno oziroma fazno moduliran signal. Čeprav so frekvenčno modulirani CW radarski sistemi (angl. Frequency-Modulated Continuous-Wave – FMCW) zelo priljubljeni, je bil za namen tega prispevka uporabljen nemoduliran CW radarski sistem.

FMCW omogoča zaznavo relativne hitrosti in razdalje do tarče, medtem ko nemoduliran CW omogoča zaznavo samo relativne hitrosti tarče. Ker je bil radarski sistem načrtovan za nadzor prometa, kjer sta pomembni samo relativna hitrost vozila in velikost radarske odmevne površine (angl. Radar Cross-Section – RCS), je bil izbran nemoduliran CW radarski signal. Velikost RCS se lahko izračuna iz moči sprejetega odbitega signala. Prednost uporabe nemoduliranega CW je v tem, da FMCW potrebuje dodatno zunanjo fazno uklenjeno zanko, kar podraži in zakomplicira izvedbo končnega produkta. Razvoj in množična proizvodnja integriranih vezij omogočata, da lahko radarski čip kupimo za nekaj evrov in ga uporabimo kot Dopplerjev radar. Radarski sistem, uporabljen za namen tega prispevka, je bil izveden z Infineonovim 24 GHz radarskim čipom BGT24LTR11N16.

Radarski čip je krmiljen z mikrokrmilnikom STM32F301K8U, ki prek USART povezave pošilja

meritve na čip CH340G. Ta nato po serijski komunikaciji prenese podatke na osebni računalnik, kjer Python skripta zajema in shranjuje podatke v podatkovni sistem.

2.2 Podatkovni sistem

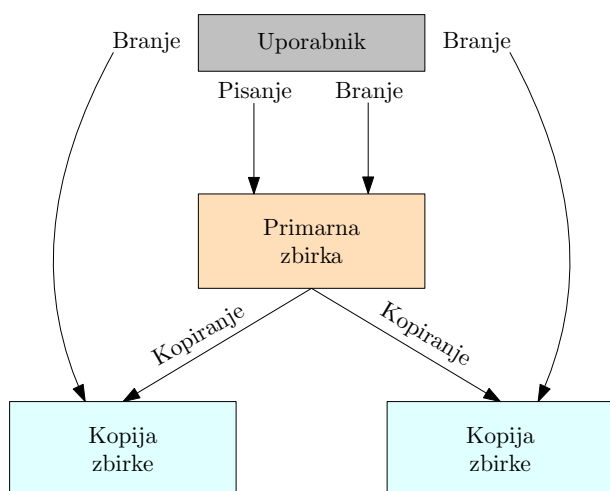
Za vzpostavitev podatkovnega sistema za pametni nadzor prometa je bila izbrana odprtokodna dokumentno usmerjena podatkovna zbirka MongoDB [11]. To je nerelacijski (angl. Not only Standard Query Language – NoSQL) podatkovni sistem, zasnovan za shranjevanje velike količine podatkov. Zaradi svoje strukture in funkcionalnosti omogoča zelo učinkovito shranjevanje ter upravljanje s podatki.

Sistem je sestavljen iz zbirk (angl. Collections), ki vsebujejo dokumente tipa binarni JSON (BSON). Prav ta struktura omogoča bolj učinkovito shranjevanje in iskanje po podatkih. Vsak dokument predstavlja en vnos v zbirko in je sestavljen iz polj, ki so lahko različno zapletenih tipov (od preprostih številskih vrednosti do zapletenih seznamov). Omogoča tudi gnezdenje dokumentov, saj lahko v polje dokumenta shranimo celoten nov dokument. Zaradi dinamične strukture dokumentov lahko v eni zbirki sobivajo dokumenti z različnimi polji, kar pomeni, da zbirka ne zahteva vnaprej določene sheme, kot to velja za tabele v relacijskih bazah. Vsak dokument ima unikatno identifikacijsko polje, ki se uporablja kot primarni ključ.

Medtem ko relacijski podatkovni sistemi podpirajo zgolj vertikalno razširljivost z nadgradnjo strojne opreme, MongoDB omogoča horizontalno razširljivost z razdelitvijo podatkov na manjše dele (angl. Sharding). Posamezni deli podatkov so shranjeni na ločenih strežnikih, povezani pa so s porazdelitvenim ključem. Horizontalna razširljivost je običajno cenejša in omogoča boljše dostopnost ter hitrejši dostop do podatkov. Vendar pa porazdelitev podatkov zahteva dodatno napravo ali storitev za usmerjanje poizvedb, kar lahko poveča stroške in zakasnitve v primeru neoptimalne porazdelitve. Pravilna izbira porazdelitvenega ključa je zato ključnega pomena za doseganje optimalne zmogljivosti in skalabilnosti sistema.

MongoDB omogoča enostavno doseganje redundance in boljše dosegljivost podatkov s kopiranjem zbirk (angl. Replication). Kopije podatkov se lahko shranijo na ločene strežnike, kar zagotavlja nemoten dostop do podatkov tudi v primeru odpovedi enega od strežnikov. Slika 3 prikazuje shemo izvedbe s kopijami podatkov. Primarna zbirka sprejema vse operacije pisanja, medtem ko so preostale kopije namenjene branju. Kopiranje zbirk omogoča tudi branje iz več različnih kopij zbirke in posledično zmanjša latence in čakalne vrste (v primeru več uporabnikov) pri dostopu do podatkov.

MongoDB ponuja preprosto izvedbo podatkovne baze v oblaku z uporabo storitve MongoDB Atlas. Storitev omogoča preprost nadzor poizvedb v podatkovnem sistemu in ima vgrajeno kopiranje zbirk na ločene strežnike. Na voljo je brezplačna različica z omejenim prostorom za shranjevanje in deljenimi računskimi viri. Izvedba z uporabo MongoDB Atlasa je lahko zaradi deljene računske moči precej počasnejša kot izvedba baze



Slika 3: Shema izvedbe podatkovnega sistema s kopirano zbirko podatkov.

podatkov, ki se izvaja lokalno, kar je opazno predvsem, kadar shranjujemo veliko število podatkov.

Pomanjkljivost MongoDB v primerjavi z relacijskimi podatkovnimi sistemi je predvsem v tem, da porabi več prostora za shranjevanje podatkov. Omejena je tudi velikost posameznega dokumenta, ki lahko meri največ 16 MB.

Za končno rešitev je bila uporabljena podatkovna zbirka v oblaku s kopijami za zagotavljanje redundance.

2.3 Zajem podatkov

Radarske meritve se izvajajo na osebem računalniku s pomočjo Python skripte, ki zajema podatke prek serijske povezave in jih shranjuje kot nize 256 števil. Ti podatki so nato pakirani v BSON dokument in naloženi v podatkovni sistem. Dokument vsebuje niz 256 števil, ki predstavljajo eno radarsko meritve, časovni žig ter identifikacijsko številko zajete slike. Ker se slike zajemajo le, ko je zaznano vozilo, meritve brez zaznanega vozila nimajo identifikacijske številke slike. Prag za zaznavanje vozila je določen na vrednosti 4500, kar je nekoliko nad pragom šuma. Za zajem serijskih podatkov se uporablja Python knjižnica *Serial*, za zajem slik knjižnica *OpenCV*, podatki pa se nalagajo v MongoDB s pomočjo knjižnice *PyMongo*.

Zaradi želje po hitrejših poizvedbah so bile slike shranjene v ločeno zbirko. Zajem slik je bil izveden s preprosto USB kamero, slike pa so bile pretvorjene v nize števil, nato pa stisnjene za zmanjšanje velikosti podatkov. Vsaka slika je bila v BSON dokumentu opremljena s časovnim žigom in tipom slike. Ko je bil dokument uspešno naložen, je podatkovni sistem vrnil identifikacijsko številko, ki omogoča povezavo med radarskimi podatki in ustreznimi slikami. Med 10-urnimi meritvami je bilo zbranih približno 65000 meritev RCS in posnetih okoli 1300 fotografij vozil.

2.4 Uporabniški vmesnik

Za postavitev spletne aplikacije z uporabniškim vmesnikom je bil uporabljen odprtokodni strežnik Express.js. To je hitro, prilagodljivo in minimalistično spletno ogrodje za Node.js [12]. Orodje poenostavlja gradnjo spletnih aplikacij in aplikacijskih vmesnikov z uporabo JavaScripta na strani strežnika. Omogoča preprosto komunikacijo s podatkovno bazo MongoDB z uporabo knjižnice *Mongoose*. V izvedbi je bil uporabljen lokalno zagnan strežnik, možna bi bila pa tudi izvedba, kjer bi strežnik tekel v oblaku.

Slika 4 prikazuje uporabniški vmesnik. Sestavljen je iz dveh delov: prikaza spektra meritev radarske odmevne površine in prikaza pripadajoče fotografije. Za pohitritev delovanja uporabniškega vmesnika so uporabljena polja, s katerimi filtriramo podatke in s tem zmanjšamo število podatkov, ki jih je treba prenesti ob posameznih poizvedbah v podatkovni sistem. S poljem *Časovni interval* lahko izberemo časovni interval, v katerem so nastale meritve, s poljem *Hitrost vozila* pa izberemo filtriranje po hitrosti zaznanega vozila. Filtrirane meritve se pokažejo v polju *Meritve*. Dodan je bil tudi gumb za osvežitev že naloženih podatkov, ki je priročen, kadar se meritve izvajajo v realnem času.

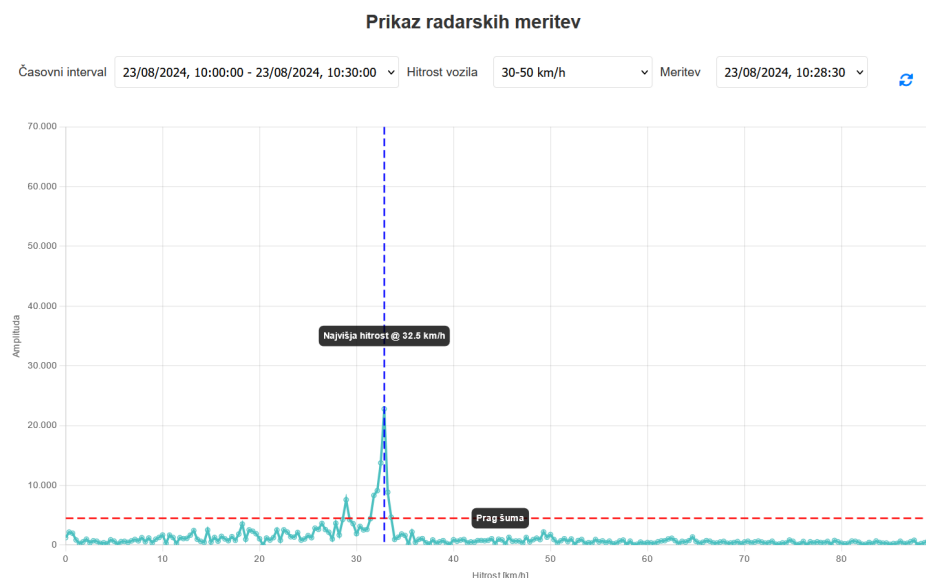
Za lepši prikaz grafov je bila uporabljena knjižnica *Chart.js*, ki omogoča zelo prilagodljiv in preprost prikaz grafikonov. Graf meritev prikazuje spekter meritev radarske odmevne površine, kjer x-os prikazuje hitrost vozila v km/h, y-os pa izmerjeno amplitudo. Prikazana je tudi odločitvena meja za zaznavo vozila in hitrost zaznanega vozila. Če je bila za izbrano meritev zajeta tudi fotografija vozila, se ta prikaže pod grafom.

3 Zaključek

V prispevku je bil predstavljen radarski sistem za pamenten nadzor prometa, ki omogoča zajem, shranjevanje in prikaz podatkov radarskih meritev. Sistem temelji na uporabi nemoduliranega radarja, ki meri hitrost vozil in velikost njihove radarske odmevne površine. Meritve so shranjene v dokumentno usmerjenem podatkovnem sistemu MongoDB, ki zagotavlja učinkovitost in skalabilnost, potrebni za obdelavo velike količine podatkov. Uporabniški vmesnik, razvit z Express.js, omogoča enostaven dostop do podatkov in njihovo vizualizacijo preko spletne aplikacije. Izvedba je pokazala, da je mogoče s pomočjo preprostih, a učinkovitih komponent zagotoviti zmogljiv in razširljiv sistem za nadzor cestnega prometa, ki se lahko prilagodi različnim potrebam, vključno z izvedbo v oblaku.

4 Zahvala

Delo je nastalo pod okriljem projektov J2-50072, V2-24005, L7-60164 in programa P2-0246 Javne agencije za znanstvenoraziskovalno in inovacijsko dejavnost Republike Slovenije.



Pripadajoča slika:



Slika 4: Uporabniški vmesnik. Zgoraj je prikazan spekter meritev radarske odmevne površine. Spodaj je prikazana pripadajoča fotografija.

Literatura

- [1] L. Podbregar, B. Batagelj, A. Blatnik in A. Lavrič, "Advances in and Applications of Microwave Photonics in Radar Systems: A Review," *Photonics*, vol. 12, str. 529, št. 6, 2025.
- [2] A. Blatnik, L. Zmrzljak in B. Batagelj, "Radio Front-End for Frequency Agile Microwave Photonic Radars," *Electronics*, vol. 13, str. 4662, 2024.
- [3] J. Dickmann et al., "Automotive radar the key technology for autonomous driving: From detection and ranging to environmental understanding," *Proc. IEEE Radar Conf. (RadarConf)*, str. 1–6, Philadelphia, ZDA, 2016.
- [4] BGT24LTR11 Product Brief. [Online]. Dostopno: www.infineon.com/dgdl/Infineon-BGT24LTR11-PB-v0100-EN.pdf?fileId=5546d4625696ed7601569d063799153e. [Dostopano: 10. 07. 2025].
- [5] S. Pisa, E. Pittella in E. Piuze, "A survey of radar systems for medical applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 31, št. 11, str. 64–81, november 2016.
- [6] W. M. Brown in L. J. Porcello, "An introduction to synthetic-aperture radar," *IEEE Spectrum*, vol. 6, št. 9, str. 52–62, september 1969.
- [7] R. J. Doviak, D. S. Zrnic in D. S. Sirmans, "Doppler weather radar," *Proceedings of the IEEE*, letn. 67, št. 11, november 1979, str. 1522–1553.
- [8] K. v. Klooster, M. Petelin, M. Thumm in M. Aloisio, "Ka-band groundstation antenna aspects for deep space telecommunication and radar," *Proc. 7th European Conf. on Antennas and Propagation (EuCAP)*, str. 242–246, Gothenburg, Švedska, 2013.
- [9] J. Sánchez-Oro, D. Fernández-López, R. Cabido, A. S. Montemayor in J. J. Pantrigo, "Radar-based road-traffic monitoring in urban environments," *Digital Signal Processing*, vol. 23, št. 1, str. 364–374, januar 2013.
- [10] E. J. Barlow, "Doppler Radar," *Proceedings of the IRE*, letn. 37, št. 4, str. 340–355, april 1949.
- [11] MongoDB Manual. [Online]. Dostopno: www.mongodb.com/docs/manual/. [Dostopano: 10. 07. 2025].
- [12] Express. [Online]. Dostopno: <https://expressjs.com/>. [Dostopano: 10. 07. 2025].