**SVEUČILIŠTE U ZAGREBU**

**FAKULTET ORGANIZACIJE I INFORMATIKE**

**V A R A Ž D I N**

**Paula Kokić**

**SIMULACIJA POZICIONIRANJA AUTONOMNIH LETJELICA U PROSTORU**

**ZAVRŠNI RAD**

**Varaždin, 2016.**

**SVEUČILIŠTE U ZAGREBU**

**FAKULTET ORGANIZACIJE I INFORMATIKE**

**V A R A Ž D I N**

**Paula Kokić**

**Matični broj: 41987/13-R**

**Studij: Informacijski sustavi**

**SIMULACIJA POZICIONIRANJA AUTONOMNIH LETJELICA U PROSTORU**

**ZAVRŠNI RAD**

**Mentor:**

prof.dr.sc. Neven Vrček

**Varaždin, kolovoz 2016.**

**Sadržaj**

[1. Uvod 1](#_Toc460451712)

[2. SOA 2](#_Toc460451713)

[2.1. GPS 2](#_Toc460451714)

[2.2. Autonomna letjelica 3](#_Toc460451715)

[2.3. RF propagacija 3](#_Toc460451716)

[3. Hipoteze 5](#_Toc460451717)

[3.1. Minimizacija GPS greške u crowd sourced sustavima 5](#_Toc460451718)

[3.2. Ciljevi 5](#_Toc460451719)

[4. Aplikacija 6](#_Toc460451720)

[4.1. Korisnički zahtjevi 6](#_Toc460451721)

[4.2. Struktura sustava 6](#_Toc460451722)

[4.3. Model podataka 8](#_Toc460451723)

[4.4. Algoritam za korekciju 8](#_Toc460451724)

[4.5. Simulacija ulaznih parametara 12](#_Toc460451725)

[4.6. Implementacija 12](#_Toc460451726)

[5. Simulacija 13](#_Toc460451727)

[5.1. Scenariji 13](#_Toc460451728)

[5.2. Rezultati 13](#_Toc460451729)

[6. Zaključak 14](#_Toc460451730)

[7. Literatura 15](#_Toc460451731)

1. Uvod

U posljednjih par godina, na području tehnologije, sve se više spominju pojmovi poput „autonomne letjelice“, „dronovi“, „Internet of Things“, „crowdsourcing“, itd... Upravo zato će se i ovaj rad orijentirati tom području, odnosno dotaknuti se barem jednog njegovog dijela i na taj način pokušati doprinjeti njegovom razvoju.

Dakle, ovaj rad bavit će se pozicioniranjem autonomnih letjelica, odnosno simulacijom njihovog pozicioniranja te ispravljanjem GPS pogreške njihove pozicije, tj. lokacije. Naime, radi se o tome da postoji mogućnost minimiziranja pogreške koju pojedini čvor dobije skupa s lokacijom od GPS sustava i to očitavanjem jačine WiFi signala drugih čvorova koji se nalaze u njegovoj blizini. Prema dobivenim očitavanjima, računat će se udaljenost između čvorova te na temelju toga korigirati lokacija pojedinog čvora, odnosno smanjiti pogreška GPS sustava.

Rad će najprije pojasniti teoretski dio, tj. detaljnije objasniti sve pojmove koji će nam biti potrebni za shvaćanje problema i algoritma, a to su: GPS sustav, pojam autonomne letjelice i RF propagacije. Nakon toga bit će iznesena hipoteza na kojoj se zasniva ovaj rad te nakon toga prikazan i opisan rad aplikacije te rezultati koji su nastali korištenjem aplikacije.

1. SOA
   1. GPS

GPS (eng. *Global Positioning System*) je američki sustav koji korisnicima omogućava pozicioniranje, navigaciju i vremenske usluge. Razvoj ove tehnologije započeo je još u ranim 70-tima prošlog stoljeća, da bi danas već bilo korišten od gotovo svih uređaja kojima je potrebno odrediti lokaciju na Zemlji. Sustav se može koristiti neovisno o vremenskim prilikama, dobu dana ili lokaciji na Zemlji, važno je samo da pojedini čvor ima nesmetan signal prema najmanje četiri satelita GPS-a.

Postoje tri glavna segmenta rada GPS-a: svemirski, kontrolni i korisnički. Svemirski segment sastoji se od 24-31 satelita, koji se nalaze u srednjoj Zemljinoj orbiti i korisnicima odašilju radio signale. Svaki satelit dnevno obiđe Zemlju dva puta, ali na način da su u svakom trenutku, na gotovo svakoj lokaciji na Zemlji dostupna minimalno četiri satelita kako bi sustav mogao pravilno raditi. Kontrolni sustav sastoji se svjetske mreže koja prati GPS satelite, s njima razmjenjuje podatke, nadzire njihove transmisije i obavlja analize. Konačno, korisnički segment sastoji se od samih GPS prijemnika, bilo samostalnih, bilo ugrađenih u različite uređaje, koji očitavaju signale GPS satelita i određuju lokaciju korisnika.

Korisnički segment GPS sustava dovodi upravo do njegove primjene. Glavna podjela upotrebe GPS-a je na civilne (dostupne svim građanima na Zemlji) i vojne svrhe (dostupne američkoj vojsci i njihovim vojnim saveznicima). Budući da je ova usluga besplatna, otvorena i poprilično neovisna, omogućila je razvoj stotine aplikacija bez kojih se današnji život ne bi mogao ni zamisliti, krenuvši od mobitela i pametnih telefona, preko ručnih satova do buldožera, bankomata i sl. Ovaj sustav koristi se u poljoprivredi, građevini, rudarstvu, dostavljanju paketa i logističkim sustavima, bankarskim sustavima, komunikacijskim mrežama... (prema: GPS.GOV, 2016) O primjeni GPS sustava može se danima govoriti, no to nije glavna tema ovog rada. Umjesto toga, u nastavku se iznosi zašto je ovom radu potreban GPS sustav i kako će se koristiti.

Svaki GPS prijemnik (pa tako i oni koji se nalaze u autonomnim letjelicama), od GPS satelita prima veću količinu podataka, a najbitnije su one koje se tiču same lokacije: geografska širina (eng. *latitude*) i geografska dužina (eng. *longitude*). Osim toga, jako je bitan i podatak GPS pogreške (eng. *error*) jer ćemo se njime najdetaljnije baviti. Geografska širina i dužina izražene su u stupnjevima, minutama i sekundama, dok je greška prikazana u metrima.

* 1. Autonomna letjelica

Pojam autonomna ili bespilotna letjelica (eng. *Unmanned Aerial Vechile – UAV*) odnosi se na letjelicu (zrakoplov) bez posade, odnosno pilota koji leti s njom. Takvu letjelicu se može nadzirati i upravljati na daljinu (ručno), a neke imaju i sposobnost autonomno letjeti pomoću unaprijed isprogramiranih planova leta ili kompeksnih dinamičkih automatskih sustava. Letjelice su često opremljene različitim senzorima, kamerama, odašiljačima i prijemnicima (kao što je GPS prijemnik ili RF antena), ovisno o svrhi/zadaći koju obavlja. A zadaće ovih letjelica su jednako različite: koriste se u vojne svrhe, ali i civilne – za snimanje iz zraka, dostavu, različita istraživanja, spašavanje, itd.

Razina autonomije je dio koji se zapravo još uvijek razvija, tako da čovjek ima sve manju ulogu u upravljanju samom letjelicom. Neka od područja koja će biti značajna za napredak autonomije jesu: spajanje podataka iz senzora, komunikacije između pojedinih letjelica u svrhu nadopunjavanja i korigiranja podataka, planiranje kretanja u svrhu pronalaska optimalnog puta između dvije točke (kako prijeđi određene zapreke, itd.), raspodjela posla i kooperacija između pojedinih letjelica kako bi se najoptimalnije riješili dani zadaci ili maksimizirala šansa za uspjeh u danoj misiji. (prema: theuav.com, 2016)

* 1. RF propagacija

Radio frekvencija (eng. *radio frequency*, RF) definira se kao bilo koja frekvencija elektromagnetskog vala koja leži u rasponu između 3 kHz i 300 GHz, a koristi se u komunikacijske svrhe ili kao signal za radar. (prema: Merriam Webster, 2016) Nas zapravo najviše zanimaju frekvencije od 2.4 GHz i 5GHz, jer na njima leži Wi-Fi signal, koji će odašiljati i primati naši čvorovi, tj. autonomne letjelice.

RF propagacija je pojam koji označava način na koji se radio valovi šire između dvije točke na Zemlji. Neću previše ulaziti u detalje, jer je to područje fizike, već ću samo naglasiti par stvari koje su nam važne za ovaj rad. Naime, propagacija signala važna nam je jer ćemo upravo pomoću nje pokušati odrediti udaljenost između dvije točke. Ono što trebamo znati jest da se svaki signal širi iz jedne točke određenom jačinom (mjereno u decibel-miliwatima, dBm; označava električnu snagu u dB koja odgovara 1 mW), i šireći se kroz prostor njegova jačina slabi. Signal nikad neće prestati, samo će slabiti, ali postoji prag, tj. minimalna jačina signala koju neki prijemnik može očitati, a ona iznosi oko -90 dBm. (prema: Experts Exchange, 2013) Formula koja će se koristiti za izračun udaljenosti između dva čvora je tzv. formula za gubitke propagacije u neomeđenom prostoru (eng. *free-space path loss*, FSPL), a glasi ovako:

gdje je:

* R – jačina primljenog signala, mjeri se putem odgovarajuće WiFi opreme na letjelici (dBm)
* T – jačina odašiljanja signala, iznosi 17 dBm
* K – konstanta gubitka propagacije, iznosi -147.55
* f – frekvencija WiFi signala, iznosi 2450 MHz
* n – eksponent gubitka propagacije, iznosi 2, jer se radi o slobodnom prostoru
* r – udaljenost između dva čvora (rezultat u metrima, m)

(prema Tomaš, 2013)

Prikupljajući na taj način podatke o udaljenosti između čvorova, tj. onih čvorova koji se međusobno „vide“, odnosno mogu očitati jedan drugome jačinu signala, pokušat će se preciznije odrediti lokacija pojedninog čvora i paralelno smanjiti GPS pogreška.

1. Hipoteze
   1. Minimizacija GPS greške u crowd sourced sustavima

Masovna podrška (eng. *crowdsourcing*) termin je koji predstavlja postupak dobivanja potrebnih usluga, ideja ili podataka od neodređene skupine ljudi. (prema: Frančula, 2015) Prema tome, crowd sourced sustavi (sustavi masovne podrške) su oni sustavi koji se sastoje od jedinica koje međusobno samostalno razmjenjuju podatke. U ovom slučaju, radi se o autonomnim letjelicama koje međusobno razmjenjuju podatke, odnosno konkretno (između ostalog) svoju lokaciju.

Problem koji je postavljen je sljedeći: GPS pogreška koju autonomne letjelice dobivaju skupa s lokacijom od GPS sustava mogu varirati, i iznositi i do 30 metara (prema: GPS SPS PS, 2008)

Hipoteza koja se postavlja glasi: Koristeći RF propagacijski model može se minimizirati GPS greška u crowd sourced sustavima. U prijašnjem poglavlju objašnjen je pojam RF propagacije i formule na temelju koje bi se ovakva postavka mogla ostvariti.

* 1. Ciljevi

Ovim radom pokušat će se dokazati navedena hipoteza koristeći osmišljeni algoritam za korekciju GPS pogreške. Bit će napravljena jednostavna aplikacija za simuliranje GPS podataka i kretanje čvorova te primijenjen algoritam za korekciju. S dobivenom preciznijom lokacijom svakog pojedinog čvora, algoritam će biti primjenjiv u stvarnim crowd sourced sustavima. U konačnici to rezultira lakšim nadgledanjem i upravljanjem pojednih čvorova, odnosno u ovom slučaju bespilotnih letjelica.

1. Aplikacija
   1. Korisnički zahtjevi

Prije razrade samog algoritma, definirat će se korisnički zahtjevi, odnosno specifikacija i domene aplikacije koja će koristiti taj algoritam, kako bi ga mogli testirati.

Aplikacija, koju možemo nazivati i Simulator pozicioniranja dronova, treba ispunjavati sljedeće zahtjeve:

* mogućnost unošenja podataka o svakom dronu: ID, naziv, lokaciju u obliku x i y koordinata, početni smjer kretanja (u stupnjevima) i brzinu
* mogućnost dodavanja unesenog drona na listu dronova koji će biti u simulaciji
* mogućnost brisanja dodanih dronova
* mogućnost pokretanja, zaustavljanja/pauziranja i ponovnog pokretanja (reset) simulacije
* vizualni prikaz kretanja dronova, njihove GPS greške i greške korigirane algoritmom
* mogućnost spremanja dobivenih rezultata simulacije u vanjsku datoteku
  1. Struktura sustava

U nastavku slijedi opis strukture sustava, odnosno način na koji je osmišljena aplikacija. Za bolji opis, za prikaz će se koristiti UML dijagram klasa, koji sadrži sve klase, atribute, metode i odnose među klasama.

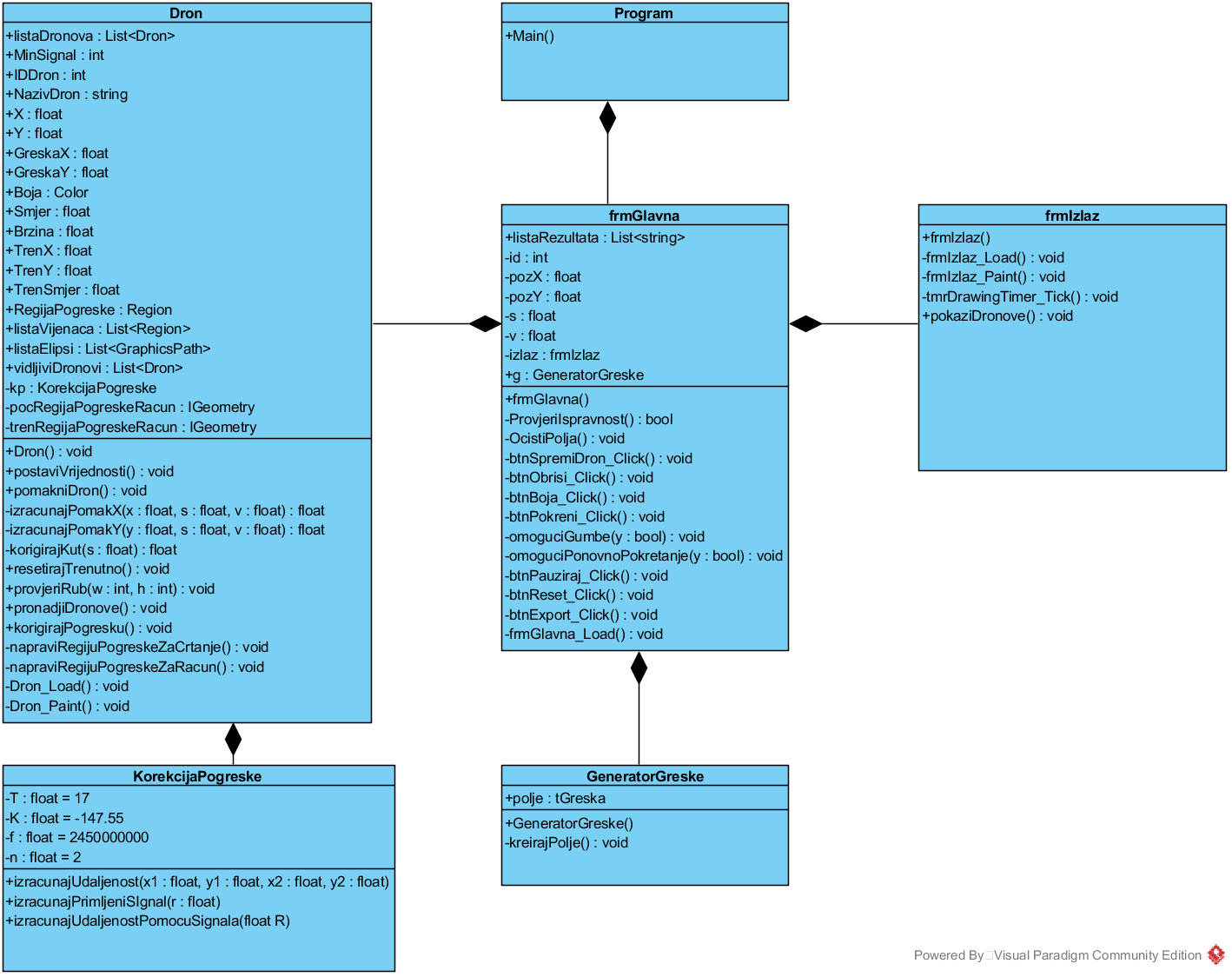
Slika 1. prikazuje navedeni UML dijagram klasa koji opisuje aplikaciju. S obzirom da se radi o jako maloj aplikaciji, dijagram je poprilično jednostavan i sadrži samo šest klasa. Strukturno se sustav može razložiti na dva sloja: klase koje služe za ulaz i izlaz (prikaz) podataka, odnosno komunikaciju s korisnikom (a to su FrmGlavna i FrmIzlaz) te klase koje služe za samu obradu podataka (Program, Dron, KorekcijaPogreske, GeneratorGreske).

Klasa Program zapravo je početna klasa, iz koje se instancira i poziva prva klasa sa svojom funkcijom, a to je klasa FrmGlavna.

Klasa FrmGlavna je klasa koja služi za prikaz početne forme, koja služi za unos podataka te sadrži glavne kontrole za upravljanje simulacijom. Prema tome, uglavnom sadrži metode za određene događaje (za klik miša na određeni gumb), te metode za uključivanje/isključivanje određenih gumbova, sigurnosnih provjera podataka i sl.

Klasa FrmIzlaz pokreće se odmah nakon glavne forme, a služi samo za prikaz čvorova, odnosno iscrtavanje njihovog kretanja. Sadrži i okidač pomoću kojeg se odvija cijela simulacija kretanja.

Klasa GeneratorGreske još je jedna mala klasa, čiji se objekt također jedino instancira unutar klase FrmGlavna. Ta klasa služi generiranju vrijendosti GPS pogreške i jednako iz nje čitaju (koriste je) svi objekti klase Dron.

Klasa Dron je najveća klasa, a sadrži sve podatke o čvoru. Osim osnovnih obilježja (atributa), kao što su ID, naziv, lokacija (x,y), greška (x,y), boja (koristi se za prikaz), smjer kretanja i brzina, sadrži još neke atribute. Atributi TrenX i TrenY označavaju trenutnu lokaciju (x,y), atribut MinSignal označava najmanju jačinu signala koju dron može „čuti“ (rečeno je da je to -90dBm, ali ovdje je ostavljeno da se čak može i podesiti za svaki čvor drugačija vrijednost), RegijaPogreske označava GPS regiju pogreške (površinu elipse) koja će se crtati na zaslonu, atribut kp je objekt klase KorekcijaPogreske (o kojoj će biti riječi kasnije), a služi za pristup metodama za izračun podataka za simulaciju. Tu su još dva atributa, pocRegijaPogreskeRacun i trenRegijaPogreskeRacun, koja služe za pohranu samog izračuna površine pogreške (početne i trenutne). Osim ovih, tu su još neki atributi, odnosno liste. Lista dronova je statična lista koja predstavlja popis svih unesenih dronova. ListaVijenaca i listaElipsi služe za pohranu međurezultata (vijenaca i elipsi), za svaki pojedini dron posebno, odnosno omogućuju njihovo iscrtavanje na zaslon. Lista vidljiviDronovi je lista koja sadrži sve dronove koji promatrani dron „vidi“ (više o tome bit će rečeno u poglavlju 4.4. Algoritam za korekciju). Ako pogledamo metode, možemo izdvojiti tri skupine. Prva skupina metoda služi za osnovnu manipulaciju podacima, odnosno upravljaju događajima (metode postaviVrijednosti(), resetirajTrenutno(), Dron\_Load(), Dron\_Paint() ). Druga skupina metoda služi za simulaciju kretanja dronova (metode pomakniDron(), izracunajPomakX(), izracunajPomakY(), korigirajKut(), provjeriRub() ). Treća skupina metoda koristi se za izračun, tj. korekciju GPS pogreške (metode pronadjiDronove(), korigirajPogresku(), napraviRegijuPogreskeZaCrtanje(), napraviRegijuPogreskeZaRacun() ).

Slika 1. Dijagram klasa za aplikaciju Simulator pozicioniranja dronova

Klasa KorekcijaPogreske još je jedna manja klasa, koja služi za izračun simuliranih vrijednosti (metode izracunajUdaljenost() i izracunajPrimljeniSignal() ) te same udaljenosti između čvorova (metoda izracunajUdaljenostPomocuSignala() ) te kao atribute sadrži konstante koje se koriste u formulama.

Naravno, sve ovo će biti puno jasnije u idućim poglavljima, gdje se bude govorilo o samom algoritmu korekcije i računanju simuliranih podataka.

* 1. Model podataka

neki tekst ovdje

* 1. Algoritam za korekciju

U nastavku je dan opis funkcioniranja algoritma za korekciju GPS pogreške.

Ulazni parametri za algoritam jesu: podaci od čvora čija lokacija se korigira (trenutna pozicija (x,y) i greška (x,y) ), podaci od svih dronova koji čine listu vidljivih dronova navedenog čvora (također trenutna pozicija (x,y,) i greška (x,y)), te jačine signala svih vidljivih dronova. Budući da se gotovo svi ovi podaci na neki način simuliraju, način na koji se to čini bit će opisan u idućem potpoglavlju.

Izlazni parametar jest određena regija (nepravilan geometrijski lik), koja predstavlja površinu na kojoj se može nalaziti čvor kojemu se korigira lokacija.

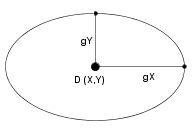
Cilj algoritma jest da izlazni parametar, tj. dobivena regija bude manja od početne regije (elipse) koju čine početni podaci o trenutnoj lokaciji i pogrešci (iz ulaznih parametara).

Rad algoritma riječima može se ukratko ovako opisati:

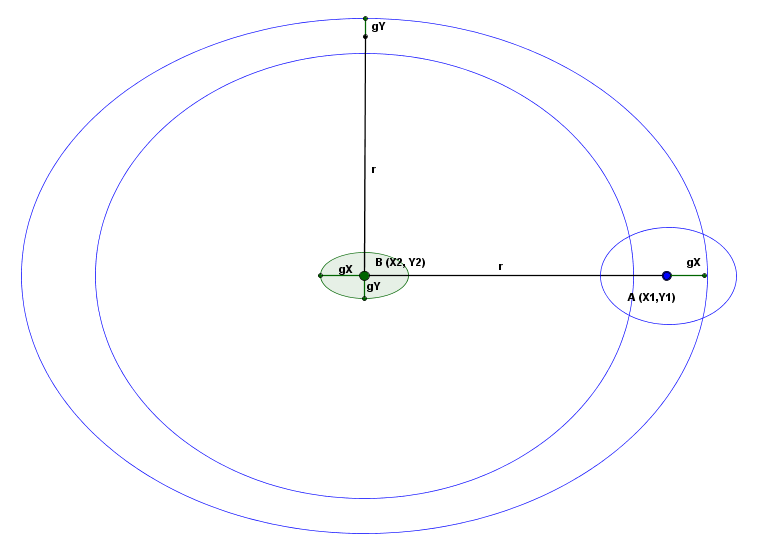
1. na temelju jačine RF signala napravi listu „vidljivih“ čvorova
2. izračunaj vlastitu elipsu pogreške
3. za svaki vidljivi čvor učini sljedeće:
   1. pomoću očitane jačine signala izračunaj udaljenost do vidljivog čvora
   2. prihvati podatke o lokaciji vidljivog čvora
   3. izračunaj širinu i visinu pomoću udaljenosti i greške vidljivog čvora za „malu“ elipsu
   4. izračunaj širinu i visinu pomoću udaljenosti i greške vidljivog čvora za „veliku“ elipsu
   5. označi prostor (vijenac) između izračunate „male“ i „velike“ elipse
4. izračunaj presjek između vlasite elipse pogreške i svih izračunatih „vijenaca“ – dobiveni presjek rezultat je algoritma i predstavlja korigiranu pogrešku

U nastavku su detaljno opisane sve faze, odnosno koraci navedenog algoritma.

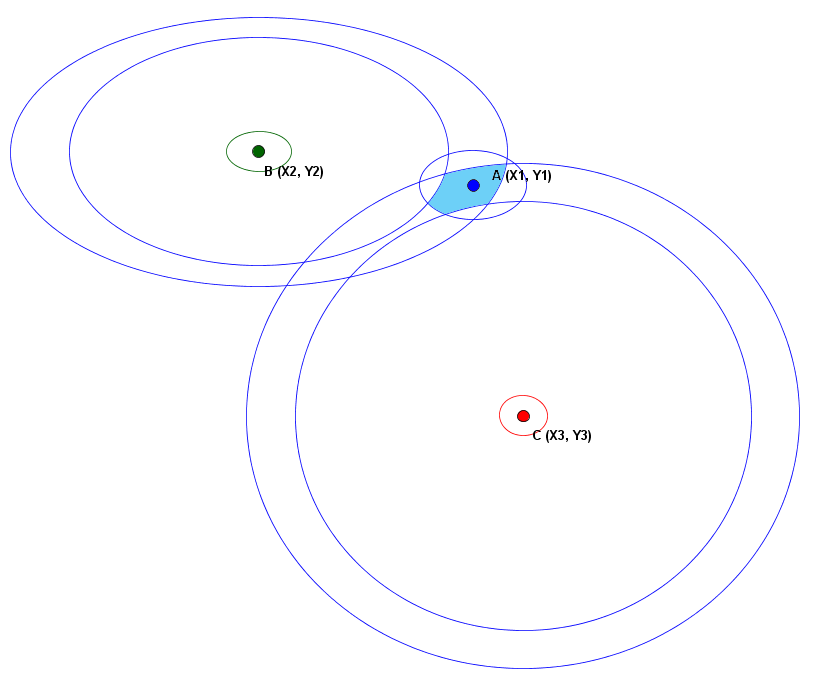
Prvi korak – izrada liste „vidljivih“ čvorova. Čvor očitava podatke, tj. jačinu signala svakog čvora sa svog WiFi uređaja i na temelju toga izrađuje se lista „vidljivih“ čvorova.

Drugi korak – izrada elipse pogreške. S GPS prijemnika očitavaju se podaci o lokaciji promatranog čvora te njegove x,y pogreške. Prema tome, elipsa pogreške bi bila elipsa koja opisuje čvor, na udaljenosti jednakoj grešci po x u vodoravnom smjeru, te udaljenosti jednakoj grešci po y u okomitom smjeru (vidi Sliku 1. – gY predstavlja vrijednost greške po y, analogno tome gX vrijednost greške po x, a točka D(X,Y) predstavlja lokaciju čvora) Dakle, površina ove elipse predstavlja prostor u kojem bi se prema GPS sustavu trebao nalaziti promatrani čvor. Površina ove elipse uspoređivat će se s krajnjim rezultatom kako bi se dobio postotak poboljšanja preciznosti, odnosno smanjenja pogreške.

Slika 2. Elipsa   
pogreške čvora

Treći korak – izračun „vijenaca“. Algoritam prolazi sve vidljive čvorove promatranog čvora i učitava njihove podatke: lokaciju (*x,y*), grešku (*gX, gY*) i jačinu RF signala (*R*) koju prima od svakog. Pomoću jačine signala najprije se računa udaljenost (*r*) do tog vidljivog čvora, pri tome koristi se formula za gubitke propagacije u neomeđenom prostoru koju smo spomenuli u ranijem poglavlju o RF propagaciji. Nakon toga uzimaju se u obzir greške po x i y vidljivog čvora. Kreira se velika elipsa oko vidljivog čvora (sa središtem u točki koja predstvalja lokaciju čvora, D (X,Y) ), kojoj je polumjer po širini jednak zbroju izračunate udaljenosti *r* i greške po x (*gX*), a polumjer po visini jednak zbroju izračunate udaljenosti *r* i greške po y (*gX*). Prema istoj logici, kreira se i mala elipsa oko istog čvora, samo manjeg polumjera – umjesto zbroja riječ je o razlici između udaljenosti *r* i grešci po x *gX*, odnosno po y *gY*. Razlika površina između te dvije elipse kreirat će neku vrstu eliptičnog vijenca oko čvora, a jedan dio tog vijenca će prelaziti preko promatranog čvora (za koji računamo korigiranu pogrešku), odnosno njegove elipse pogreške (vidi Sliku 2.)

Slika 3. Odnos između dva čvora - nastanak eliptičnog vijenca

Četvrti korak – izračun površine korigirane pogreške. U ovom koraku skupljaju se podaci iz prethodnog koraka, odnosno svi izračunati vijenci. Računa se presjek između površina tih vijenaca i površine elipse pogreške promatranog čvora – izračunati presjek predstavlja korigiranu pogrešku promatranog čvora. Na Slici 3. rezultat algoritma, odnosno presjek je osjenčan svijetlo-plavom bojom. U ovom primjeru su dva čvora (B i C) u vidljivom rasponu čvora A, tako da utječu na korekciju njegove pogreške.

Slika 4. Skica rezultata algoritma

Naravno, u nastavku algoritma, i čvor A će utjecati na korekciju pogrešaka čvorova B i C, itd...

Zašto je odabran baš ovakav način računanja i zašto bi trebao funkcionirati? GPS lokacija (X,Y) i dana greška (gX, gY), odnosno ova elipsa koja je crtana oko čvora, znači da se promatrani čvor može nalaziti bilo gdje unutar te elipse. Kada se izračuna udaljenost *r* između dva čvora, i uzme u obzir da se čvor može nalaziti na jednom ili skroz suprotnom kraju elipse, dobivamo upravo ova dva (odnosno četiri) polumjera za izradu velike i male elipse koje će činiti vijenac.

* 1. Simulacija ulaznih parametara

S obzirom na to da je ovaj rad fokusiran na sam algoritam korekcije, aplikacija (tj. algoritam) ne koristi prave ulazne parametre, već ih simulira.

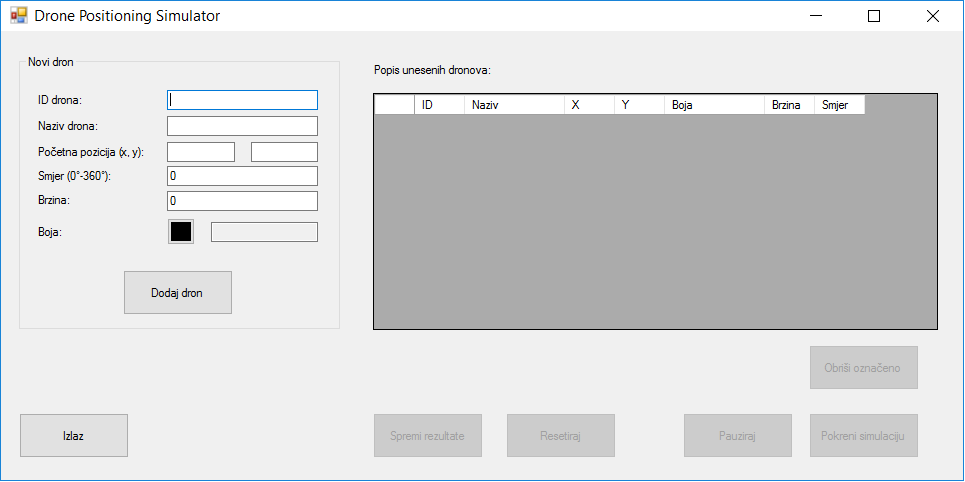
Lokacije čvorova korisnik upisuje skupa s ostalim podacima o čvoru (smjer kretanja, brzina, naziv, itd.) Sve veličine izražene su u pikselima, tako i lokacija, greška, udaljenost i sl. Smjer se unosi u stupnjevima (između 0 i 360), a brzina predstavlja broj piksela koji će prijeći čvor u zadanom intervalu, a ono iznosi 30ms. Simulacija kretanja provedena je prema matematičkim zakonima Pitagorinog poučka, uz pomoć sinusa i kosinusa. Čvorovi se kreću po fiksno određenoj veličini karte, a kad se dođe do ruba, čvor se odbija od njega pod istim kutem pod kojim je ušao.

Pogreška GPS-a (*gX, gY*) simulira se pomoću karte veličine izlazne forme, podijeljene na nekoliko nejednakih regija. Svaka regija ima svoje vrijednosti pogreške koje generira, dakle svi čvorovi koji se nalaze u istoj regiji imaju istu pogrešku. Na ovaj način se pokušava simulirati različit teren na kojem se mogu nalaziti čvorovi i njegova povezanost s GPS-om.

Jačina signala (*R*) simulira se zapravo preko iste formule za gubitke propagacije u neomeđenom prostoru iz udaljenosti čvorova. Udaljenost čvorova za ovu svrhu računa se na temelju lokacija oba čvora, pomoću matematičke formule za udaljenost dvije točke u koordinatnom sustavu.

* 1. Implementacija

Za implementaciju ovakve aplikacije odabran je jezik C#, odnosno .NET tehnologija i alat Microsoft Visual Studio te je napravljena Windows Forms aplikacija. Osim osnovnih paketa i biblioteka, preuzete su NetTopologySuite i GeoAPI biblioteke za lakši rad s geometrijskim likovima. Za korištenje aplikacije, potrebno je imati Windows operacijski sustav te podršku za .NET 4.5 Framework.

Nakon pokretanja aplikacija otvaraju se dvije jednostavne forme, jedna pored druge. Prva je glavna forma (o kojoj je bilo riječ u poglavlju 4.2. Struktura sustava), a njen izgled može se vidjeti na Slici 4. Kao što možemo vidjeti, s lijeve strane nalazi se prostor za unos podataka o dronu: ID, naziv, početna pozicija, smjer (0-360), brzina (0-15) i boja. Klikom na gumb 'Dodaj dron' dron se dodaje u listu prikazanu s desne strane.

Slika 5. Glavna forma aplikacije

Nakon dodavanja drona, na raspolaganju stoje ostale mogućnosti. Brisanje drona izvršava se označavanjem reda drona kojeg se želi obrisati i klikom na gumb 'Obriši označenog'. Ako treba pokazati ili sakriti vijence tijekom simulacije, potrebno je označiti željeni dron i kliknuti na gumb 'Pokaži/sakrij vijence označenog'. Gumb 'Pokreni simulaciju' pokreće simulaciju kretanja i izračuna pogreške dronova. Gumb 'Pauziraj' zaustavlja, točnije samo pauzira trenutnu simulaciju, a moguće ju je nastaviti ponovnim klikom na gumb 'Pokreni simulaciju'. Gumbom 'Resetiraj' možemo resetirati cijelu simulaciju i vratiti dronove na početna mjesta. Gumb 'Spremi rezultate' sprema rezultate simulacije u vanjsku .txt datoteku. Rezultati simulacije sadrže ID i naziv drona, trenutnu lokaciju (x,y), vrijednost GPS površine elipse pogreške, vrijednost površine regije korigirane pogreške te postotak za koliko je korigirana površina manja od početne (tj. GPS-ove).

1. Simulacija
   1. Scenariji

bla bla bla

* 1. Rezultati

bla bla bla

1. Zaključak

Fuck yea :D

1. Literatura
2. Tsui, JBY (2000) *Fundamentals of Global Positioning System Receivers*. New York: John Wiley & Sons, Inc. Preuzeto 17.08.2016. s <http://kakukoto.free.fr/F/019/Fundamentals%20of%20Global%20Positioning%20System%20Receivers/Fundamentals%20of%20Global%20Positioning%20System%20Receivers.pdf>
3. GPS.GOV (2016) *Official U.S. Government information about the Global Positioning System (GPS) and related topics*. Dostupno 18.08.2016. na <http://www.gps.gov/>
4. The UAV (2016) Dostupno 18.08.2016. na <http://www.theuav.com/index.html>
5. Sayler K (2015) *A World of Proliferated Drones: A Technology Primer.* Dostupno 20.08.2016. na <http://www.cnas.org/sites/default/files/publications-pdf/CNAS%20World%20of%20Drones_052115.pdf>
6. Merriam Webster (2016) *Dictionary – Radio Frequency*. Dostupno 20.08.2016. na <http://www.merriam-webster.com/dictionary/radio%20frequency>
7. Experts Exchange (2013) *Industry standard for minimum Wifi signal strength?* Dostupno 20.08.2016. na <https://www.experts-exchange.com/questions/28103112/Industry-standard-for-minimum-Wifi-signal-strength.html>
8. Tomaš B (2013) *WiFi roaming in urban multi-sensor environment*.
9. Frančula N (2015) *Terminologija*. Dostupno 27.08.2016. na <https://bib.irb.hr/datoteka/793247.Masovna_podrska.pdf>
10. Department of Defense USA (2008) *Global Positioning System Standard Positioning Service Performance Standard.* Dostupno 27.08.2016. na <http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf>