Analiza postupka procjene položaja temeljem zadanih pseudoudaljenosti u programski određenom prijamniku za satelitsku navigaciju

Filić, Mia

Master's thesis / Diplomski rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:217:739461

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Repository of the Faculty of Science - University of Zagreb

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO–MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

Mia Filić

ANALIZA POSTUPKA PROCJENE POLOŽAJA TEMELJEM ZADANIH PSEUDOUDALJENOSTI U PROGRAMSKI ODREĐENOM PRIJAMNIKU ZA SATELITSKU NAVIGACIJU

Diplomski rad

Voditelji rada:
izv.prof.dr.sc. Luka Grubišić i
prof.dr.sc. Renato Filjar

Zagreb, rujan, 2017

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Ovaj rad posvećujem svojoj sestri koja je život provela uz mene.

Zahvaljujem se izv.prof.dr.sc. Luki Grubišiću i prof.dr. Renatu Filjaru na vodstvu i savjetima i Antonii Bošnjak na strpljenju i pomoći pri usavršavanju slika koje prate tekst ovoga rada.
Veliko hvala i svima ostalima koji su svjesno i nesvjesno pripomogli izradi ovoga rada svojim načinom razmišljanja, savjetima i preporukama.

Sadržaj

Sadržaj **iv**

Uvod **4**

1 Globalni pozicijski sustav (GPS) **5**
1.1 C/A PRN kod 6
1.2 P kod 7
1.3 Pogreške određivanja položaja i vrste 8
1.4 Navigacijska poruka 11
1.5 Proces određivanja položaja 12

2 Algoritam procjene položaja **17**
2.1 Iterativna metoda najmanjih kvadrata 18
2.2 Metoda zatvorene forme 23
2.3 Metoda najbližeg susjeda 25

3 Programski određen GPS prijamnik **27**
3.1 Model programski određenog radioprijamnika 27
3.2 Pojam programski određenog radioprijamnika 28
3.3 Programski određen GPS prijamnik 29
3.4 Praktična izvedba korisničkog GPS prijamnika 32

4 Praktična izvedba procjene položaja u domeni navigacijske primjene **36**
4.1 Numerička linearna algebra za rješavanje dobivenog sustava problema najmanjih kvadrata 40
4.2 Izvedbe 46

5 Ocjena kvalitete i zaključci **52**
5.1 Osnovni pristup 52
5.2 Poboljšani pristup 56

5.3 Usporedba pristupa i zaključci 61

Bibliografija **63**

A Jakobijeva matrica funkcije h, J **71**

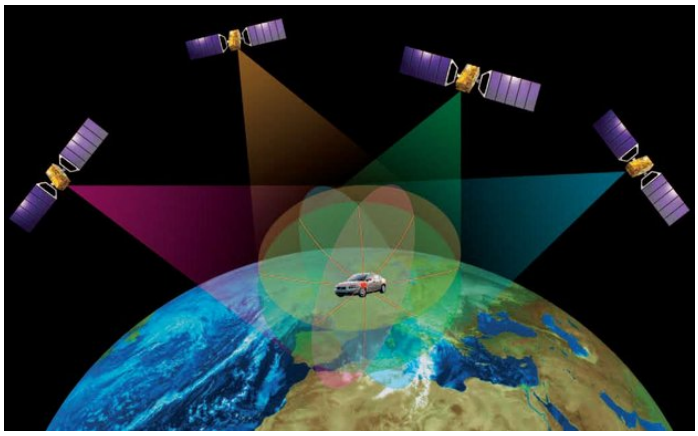
B Mjere kvalitete sazviježđa **72**

C Kodovi izvedbe algoritama **75**
C.1 Osnovni pristup 75
C.2 Poboljšan pristup: uvođenje težina (WLSM) 81

D Sadržaj datoteka ulaza izvedenih algoritama u procesu ocjene kvalitete **86**

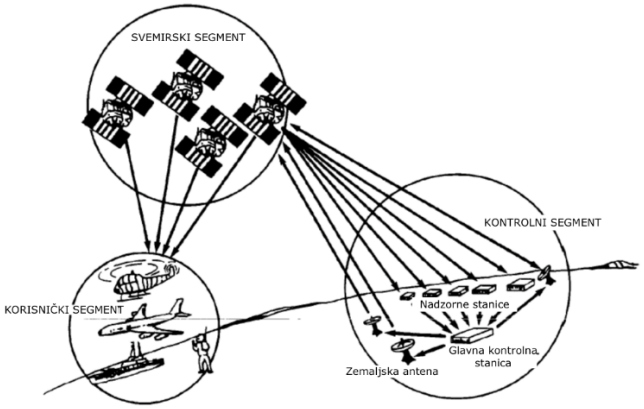
Uvod

Globalni navigacijski satelitski sustav (GNSS) je osmišljen s ciljem da u bilo kojem trenutku i za bilo koji objekt (entitet) na Zemljinoj površini može dati podatak o njegovom trenutnom vremenu, položaju i brzini gibanjem (engl. Position, Velocity and Time), tj. PVT stanju. Kao takav predstavlja temelje rastućem broju tehnoloških i društveno-ekonomskih sustava.



Slika 0.1: Satelitska navigacija[26]

Koristeći pojam GNSS, najčešće se misli na *sazviježđe* satelita koji odašilju signale potrebne za određivanje trenutnog položaja (i/ili brzine i vremena) i dodatne informacije u obliku tzv.*Navigacijske poruke* (engl. Navigation Message (NM)). *Sazviježđe* satelita predstavlja (1) svemirski segment GNSS sustava. U sastavnice GNSS-a spadaju i (2) kontrolni segment koji čine kontrolne i promatračke stanice smještene na Zemlji i (3) korisnički segment, odnosno GNSS prijamnici (slika 0.2). Kontrolni segment nadzire i upravlja radom sustava.



Slika 0.2: Segmenti GNSS-a

Trenutno postoji nekolicina GNS sustava (GNSS). Neki su u potpunosti operativni, a neki samo djelomično. Najrašireniji u civilnoj upotrebi je GPS (Global Positioning System). GPS je u potpunosti operativan i u vlasništnu Vlade SAD-a. Njime upravlja Ministarstvo obrane SAD-a (engl. US Department of Defense). GPS omogućava dvije znatno različite razine korištenja, civilnu i vojnu. Vojna razina korištenja pruža više mogućnosti i točnije određivanje PVT stanja, a dopuštena je samo povlaštenim korisnicima. Civilna razina korištenja je dostupna svima, bez dodatne naknade, uz uvjet posjedovanja GPS prijamnika.

Drugi, također u potpunost operativan GNSS, je GLONASS (Global'naya Navigatsionnaya Sputnikovaya Sistema) u vlasništvu Rusije. Neki od GNSS sustava u razvoju su: (1) Galileo i (2) BeiDou. Galileo-om upravlja Europska unija (EU). Najavljeno je da će postati u potpunosti operativan do 2020 [26]. BeiDou je kineski lokalni navigacijski satelitski sustav. U procesu je projekt proširenja BeiDou-a do globalnog do 2020 [26].

Tablica 0.1: Obilježja različitih satelitskih navigacijskih sustava

	Zemlja	Broj operativnih satelita	Frekvencije valova nosilaca	Brzina slanja navigacijske poruke
GPS	SAD	31	L1 = 1575.42 L2 = 1227.60 L5 = 1176.45	50, 25
GLONASS	Rusija	28	L1 = oko 1602 L2 = 1246	50
Beidou	NR Kina	22	B1 = 1575.42 B2 = 1191.795 B3 = 1268.52	-
Galileo	EU	18 (15 potpuno operativnih)	E1 = 1575.42 E5a-Q = 1176.45 E5b-Q = 1207.14 E6 = 1278.75	-

Primjena GNSS-a dijeli se na pozicioniranje i navigaciju.

Definicija 0.0.1 (Navigacija)**.** *Navigacija obuhvaća trenutno određivanje položaja i brzine objekta u pokretu. Svrha navigacije je praćenje i upravljanje gibanjem objekta.*

Definicija 0.0.2 (Pozicioniranje)**.** *Pozicioniranje nazivamo postupak određivanja položaja točkovnog entiteta (objekta) ili niza točkovnih entiteta u prostoru.*

Ovim radom se ponajprije razmatra bespojena (engl. off-line) navigacijska primjena, u svrhu praćenja točkovnog entiteta. Bespojena navigacija se koristi u prometnoj znanosti u analizi prometnih puteva. Kako ne zahtjeva izračunavanje u realnom vremenu (engl. real-time), svodi se na određivanje položaja točkovnog entiteta koji je statičan u danom vremenu t. Određujući položaj entiteta za niz vremena $t_1, t_2, \ldots, t_n$, dobiva se aproksimacija kretanja entiteta u vremenskom okviru $[t_1, t_n]$. Preciznost aproksimacije kretanja zadaje se veličinom okvira i parametrom n, ili dostupnošću podataka. Praksa ne zathjeva da je n u odnosu na vremenski okvir duljine 1 sata prevelik. Točno kretanje entiteta moguće je odrediti preslikavanjem dobivene aproksimacije na kartu prometnih puteva. U tu se svrhu koriste otprije poznati algoritmi. Dakle, rad se temalji na algoritamu za pozicioniranje (točkovnog entiteta) u konceptu jednog, određenog, GNSS-a: GPS u aspektu civilne razine korištenja.

Cilj rada je opisati, analizirati i izvesti osnovni (referentni) algoritam za pozicioniranje (točkovnog entiteta), uvidjeti potencijalne slabosti te predložiti, izvesti i opravdati[1] njegovo poboljšanje. Pod poboljšanjem se prvenstveno misli na poboljšanje u točnosti procjene položaja.

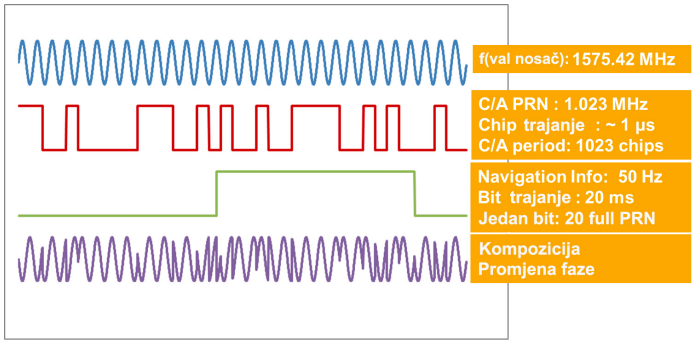
[1]Korsiti se komparativna analiza obilježja predloženog i izvornog algoritma koristeći eksperimentalno prikupljene pseudoudaljenosti.

Poglavlje 1

Globalni pozicijski sustav

Sazvježđe Globalnog pozicijskog sustava (GPS, engl. Global Positioning System) se sastoji od najmanje 24 satelita raspoređenih u 6 jednako odmaknutih orbita, svaka s inklanacijom od 55 stupnjeva od ekvatorijalne ravnine. Sateliti kruže na visini od oko 20200 kilometara od Zemljine površine u srednje visokoj orbiti oko Zemlje (engl. Medium Earth Orbit (MEO)) s periodom rotacije 12 zvjezdanih sati (11 sati i 58 minuta). Sateliti su raspoređeni na način da u svakom trenutku za svako mjesto na Zemljinoj površini postoje barem 4 dostupna satelita. Definicija dostupnosti satelita je dana u kasnijem teksu (stranica 12).

Svi GPS sateliti odašilju (radio)signale s istim frekvencijama valova nosilaca (slika 1.1). U satelitima, vrijeme je praćeno pomoću cezijevih satova koji se sinkroniziraju s univerzalnom GPS atomskom vremenskom skalom. Sinkronizacija se odvija u periodima.



Slika 1.1: GPS signal i njegove komponente (izvor: [22])

1.1 GPS signali: C/A PRN i P kod

C/A PRN kod i primjene

GPS sateliti odašilju signale na dvije frekvencije (vala nosilaca) L_1 i L_2, od kojih je L_1 na 1575.42 MHz namijenjena civilnoj upotrebi [1]. Pojam signal se često u satelitskoj navigaciji koristi samo za dio GPS signala koja sadržava C/A PRN kod (eng. Coarse Acquisition Pseudo Random Noise). Svaki satelit koristi jedinstveni C/A PRN kod koji predstavlja niz 0 i 1 duljine 1023 bit-a. GPS-prijamnik razlikuje signale (signale koji sadrže podatke potrebne za određivanje položaja i *Navigacijske poruke*) različitih satelita temeljem sadržanih C/A PRN kodova. Satelit C/A PRN kodove odašilje neprestano, s početkom na početku svake sekunde. Prijemnik primljeni C/A PRN kod koristi za razlikovanje satelita odašiljetelja, ali i za računanje pseudoudaljenosti.

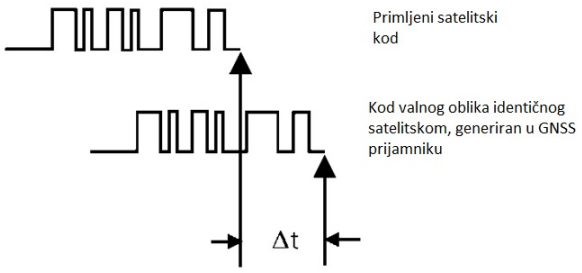
Definicija 1.1.1 (Pseudo-udaljenost)**.** *Neka su svi sateliti numerirani prirodnim brojevima s početkom u 1. Neka je* $\mathbf{S} \in \mathbb{N}$ *neki satelit i* $\mathbf{t}$ *prijamnik koji je u mogućnosti primiti signal koji odašilje satelit* $\mathbf{S}$*. Pseudo-udaljenost između satelita odašiljatelja* $\mathbf{S}$ *i prijamnika primatelja* $\boldsymbol{t}$*:*

$$d_s = c \cdot (t'_s - t_s)$$

gdje je c konstanta koja je jednaka (prosječnoj) brzini putovanja signala od satelita do prijamnika. t'_s *je vrijeme primanja signala, a* t_s *vrijeme slanja signala (po UTC vremenu).*

Pseudoudaljenost je izmjerena udaljenosti (uz sadržane pogreške mjerenja) između satelita odašiljatelja i prijamnika primatelja signala u određenom trenutku. Vrijeme putovanja signala u oznaci $\Delta t := (t'_i - t_i)$, izračunava se poravnavanjem dijela primljenog signala (C/A PRN kod-a) i u prijemniku generiranog C/A PRN koda. Naime, prijamnik i satelit istovremeno generiraju isti C/A PRN kod. Cijelo vrijeme dok satelitski signal s generiranim C/A PRN kodom putuje, prijamnik nastavlja generanje istog C/A PRN koda. Po primitku signala, kodovi se poravnavaju. Temeljem razlike u poravnanju dobivenog i generiranog C/A PRN koda, mjeri se vrijeme putovanja satelitskog signala, tj. Δt (slika 1.2).

[1] Modificirani GPS koristi će i novu frekvenciju vala nosioca $L_5 = 1176.45MHz$, a dio signala odašiljanih na L_2 također će biti dostupni i civilnim korisnicima.



Slika 1.2: Procjena vremena putovanja signala (Δt)

Za vrijednost konstante c se uzima brzina svjetlosti u vakuumu koja predstavlja brzinu putovanja radiovala (poruke satelita) u vakuumu. Ona dovoljno dobro modelira stvarnu prosječnu brzinu putovanja. Naime, satelitski signal približno 90 posto puta se nalazi u uvjetima vakuuma. Budući da se pseudoudaljenost dobiva poravnavanjem kodova, upravo opisani način određivanja pseudoudaljenosti naziva se kodni.

Postoji još i fazni način određivanja pseudoudaljenosti koji se zasniva na poravnanju valova nosilaca (engl. Carrier phase) nakon micanjem C/A PRN i P(Y) kodova iz poruke (GPS signala) [3]. Fazno mjerenje služi kao nadopuna kodnom mjerenju u svrhu poboljšanja točnosti određivanja položaja.

1.2 P kod

P kod je dio GPS signala koji se odašilje na obje frekvencije i rezerviran je za vojnu razinu upotrebe. Kao i C/A PRN kod, sastoji se od karakterističnog pseudo-slučajnog niza nula i jedinica i šalje se brzinom 1023 bit/s. Znatno je dulji. Potrebno je ukupno 37 tjedana kako bi se sekvencijalno poslao cjelokupni P kod. Za razliku od C/A koda, gdje svaki satelit ima svoj jednistveni C/A kod, P kod je distribuiran među satelitima. Isječci P koda koji pripadaju različitim satelitima međusobno su različiti. Svakih 7 dana u točno određeno vrijeme određeni satelit odašilje svoj dio P koda. Na taj način, prijamnik razlikuje pojedinačne satelite. Npr. ukoliko satelit S odašilje 14. tjedan P koda, onda je satelit S zapravo *Space Vehicle 14 (SV 14)*. Kako bi se rezerviralo korištenje P koda samo za vojnu razinu upotrebe, prijamnik signalom ne prima goli P kod, već njegovu kriptiranu verziju, u oznaci $P(Y)$. Također, samo korisnicima s

vojnom razinom upotrebe se prosljeđuje informacija kako dekriptirati $P(Y)$ u P. P kod omogućava točnije određivanje položaja entiteta.

1.3 Pogreške određivanja položaja i vrste

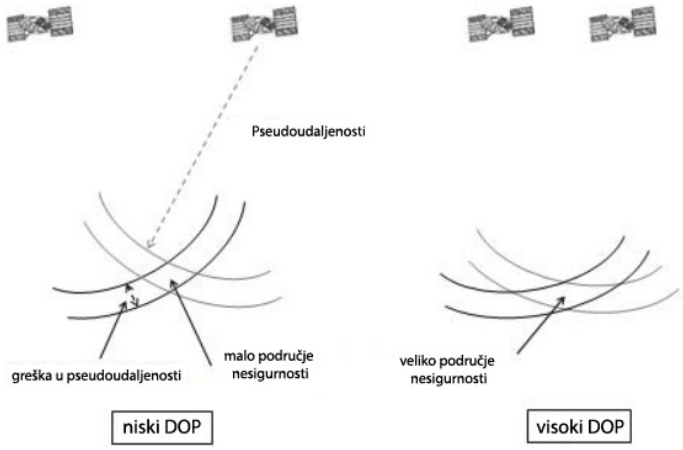
Pogreške određivanja položaja se grubo dijele na dvije vrste: (1) pogreške nastale uslijed konstrukcije ulaza algoritma i (2) uslijed primjene algoritma za određivanje položaja na mjerenim pseudoudaljenostima. Dakle, postoje dva izvora: ulazni podatci algoritma (tip 1) i algoritam (tip 2). Najčešći izvori pogreške tipa 1 su pogreške pri određivanju pseudoudaljenosti ili raspoređenost satelita oko Zemlje (slike 1.3, 1.4 i 1.5).
Dvije skupine utjecajnih veličina (izvori pogrešaka tipa 1) nazivamo:

- korisnička razdioba pogrešaka (engl. User Equivalent Ranging Error, UERE) i
- geometrijska degradacija točnosti (engl. Geometric Dilutation of Precision, GDOP).

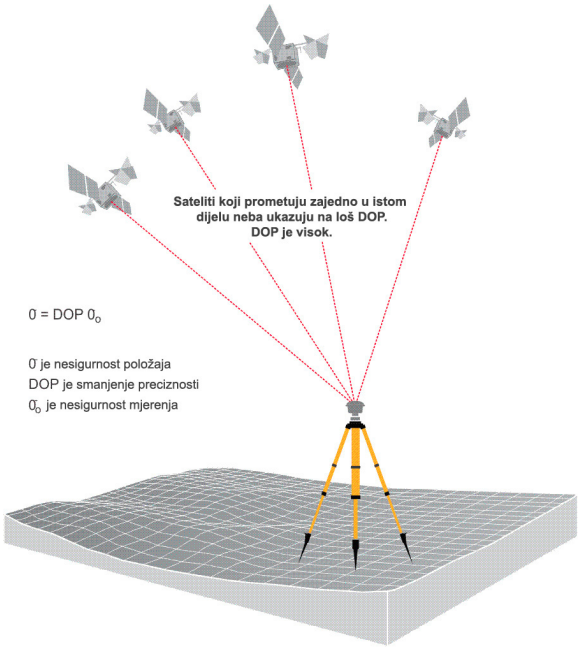
Budući da se gornje utjecajne veličine smatraju statistički neovisnima, ukupna pogreška uslijed djelovanja izvora pogreška tipa 1 dobiva se kao

$$P_{err_1} = UERE \times GDOP.$$

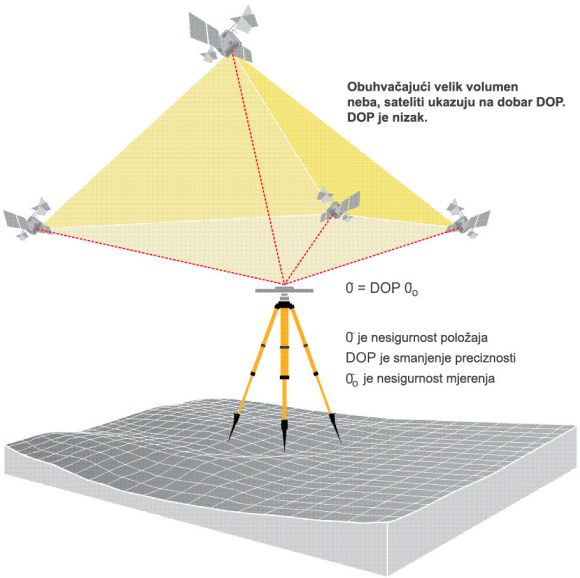
Nepovoljan položaj promatranih satelita može rezultirati i zavisnim jednadžbama sustava jednadžbi za procjenu položaja pomoću satelita. Takvo što može onemogućiti procjenu položaja. U jednostavnijim slučajevima događa se povećanje GDOP-a, a time i ukupna pogreška određivanja položaja.



Slika 1.3: Razlike u razmještaju satelita



Slika 1.4: Nepovoljan razmještaj satelita



Slika 1.5: Povoljan razmještaj satelita

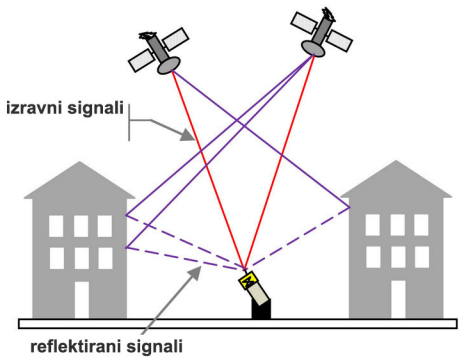
Detaljnija podjela pogrešaka tipa 1 nastalih pri određivanju pseudoudaljenosti (UERE pogreške) i područje utjecaja dano je sljedećom tablicom.

Tablica 1.1: Izvori i utjecaj pogreške tipa 1 na određivanje pseudoudaljenosti

Izvor	Utjecaj
satelit	pogreške orbite pogreška sata satelita
rasprostiranje signala	troposferska refrakcija ionosferska refrakcija
prijamnik	pogreške antene pogreška sata
	pogreška višestaznih puteva

One mogu biti sistemske ili slučajne. Utjecaj sistemskih pogrešaka otklanja se modeliranjem ili kombinacijom opažanja. Korištenjem više prijamnika, otklanjaju se pogreške spacifične za satelite. Pogreške specifične za prijamnike otkanja korištenje većeg broja satelita od potrebnog broja za određivanje položaja. Utjecaje troposfere je najsigurnije otkloniti modeliranjem, a ionosfere korištenjem dva signala različitih frekvencija. Nažalost, ponekad nije moguće korištenje dva signala različitih frekvencije pa se i utjecaj ionosfere otkanja modeliranjem. Ukoliko se utjecaj ionofere otklanja modeliranjem, uvijek ostaje dio slučajne pogreške utjecaja ionosfere koja se može uzeti u obzir prilikom izgradnje algoritma za određivanje položaja u navigacijskoj domeni (Poglavlje 4.2).

Slučajne pogreške nastaju zbog trenutnog mjerenja, slučajne komponente višestaznog signala (engl. multipath) nastalog interferencijom (superponiranjem) izravnog direktnog i reflektiranog signala (slika 1.6) te zbog slučajnog karaktera ionosferskog kašnjena koji se ne ispravlja sistemskim modelom.



Slika 1.6: Nastanak pogreške višestaznih puteva

U poglavlju 4 prvo se izvodi osnovni algoritam za određivanje položaja prijamnika koji polazi od pretpostavke o potpunoj ispravljenosti pseudoudaljenosti.
Kasnije, uvođenjem težinskih koeficijenata (Poglavlje 4.2), reducira se utjecaj pogrešaka pseudoudaljenosti.

Pogreške tipa 2 mogu imati izvor u konstrukciji (dizajnu) izvedbe algoritma ili njegovoj izvedbi, npr. numeričke greške, greške zbog ograničene preciznosti računala, aproksimacije pojedinih vrijednosti.

One se ne modeliraju algoritmima procjene položaja (Poglavlje 2), već prilikom konstrukcije izvedbe odabranog algoritma (Poglavlje 4)

1.4 Navigacijska poruka

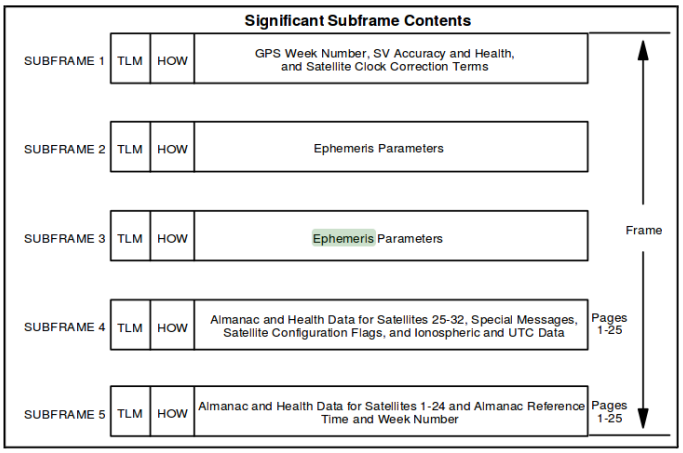
Svaki satelit, uz C/A PRN i P kod, odašilje i dodatne podatke potrebne za ispravo određivanje položaja prijamnika. Odašilje ih u obliku *Navigacijske poruke* koja se šalje zajedno s generiranim C/A PRN kodovima (slika 1.1).

Navigacijska poruka se sastoji od 25 okvira [26]. Jedan okvir se sastoji od 5 podokvira i svaki sadržava vrijeme slanja sljedećeg okvira (slika [22]). Za slanje cjelokupnog podokvira potrebno je 6 sekundi, 6 cjelokupnih C/A PRN kodova. Prijemnik je u mogućnosti računati pseudoudaljenost za novi položaj satelita svakih 6 sekundi. Za slanje cjelokupne NM, potrebno je 12.5 minuta. U nastavku teksta, naziv poruka koristi se sa značenjem potprozora.

Prozor sadrži:
1. GPS vremena odašiljanja,
2. signal prijenosa s P na C/A kod (Potpoglavlja 1.2 i 1.1),
3. podatke o orbitalnoj putanji satelita,
4. podatke o korekciji sata satelita,
5. almanah statusa svih satelita u sazvježđu,
6. koeficijente preračunavanja GPS vremena u UTC,
7. parametre standardnog GPS ionosferskog modela korekcija.

Definicija 1.4.1 (Universal Time Coordinate (UTC vrijeme))**.** *Universal Time Coordinate je vremenski standard zasnovan na međunarodnom atomskom vremenu koji se najčešće koristi u znanstvene i vojne svrhe.* [2]

[2]Drugi nazivi za taj vremenski standard su ZULU vrijeme i Greenwich Mean Time (GMT)



Slika 1.7: Pregled strukture prozora navigacijske poruke (izvor: [22])

Pojedini dijelovi navigacijske poruke pomažu pri otklanjaju pogrešaka tipa 1 (Poglavlje 1.3), određivanju pseudoudaljenosti i trenutnom položaju satelita. Naime, iz podataka o orbitalnoj putanji satelita moguće je za odabrani trenutak izračunati koordinate položaja satelita referentnom koordinatnom sustavu pa i svakom drugom.

Za potrebe ovoga rada, dovoljno je razumjeti sljedeće. Prijemnik svakih 6 sekundi ima dovoljno podataka da izmjeri novu pseudoudaljenost do istog satelita sve dok on ne prestane biti dostupan.

Definicija 1.4.2 (Dostupnost satelita S prijamniku T)**.** *Za satelit S kažemo da je dostupan prijamniku T u trenutku t ako je u sljedećih 6 sekundi u mogućnosti izmjeriti pseudoudaljenost do satelita S i konstruirati sljedeću jednadžbu:*

$$d_s = \sqrt{(x - x_s)^2 + (y - y_s)^2 + (z - z_s)^2} \tag{1.1}$$

gdje su jedine nepoznanice (x, y, z), tj. koordinate položaja prijamnika. (x_s, y_s, z_s) su poznate koordinate položaja satelita.

1.5 Proces određivanja položaja

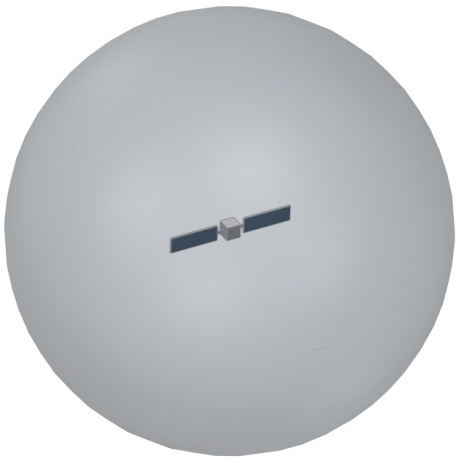
U pravilu, u svakom trenutku, prijamnik ima više dostupnih satelita od kojih dobiva poruke. Za određivanje položaja prijamnika u granicama dopuštene točnosti, zahtjevaju se barem 4 dostupna satelita.

Kako bi prijamnik odredio svoj položaj računa tri nepoznate koordinate položaja koje su obično izražene jednom od sljedećih koordinatnih sustava:

1. geodetskom koordinatnom sustavu koji je izveden referentnim ECEF XYZ koordinatnim sustavom,

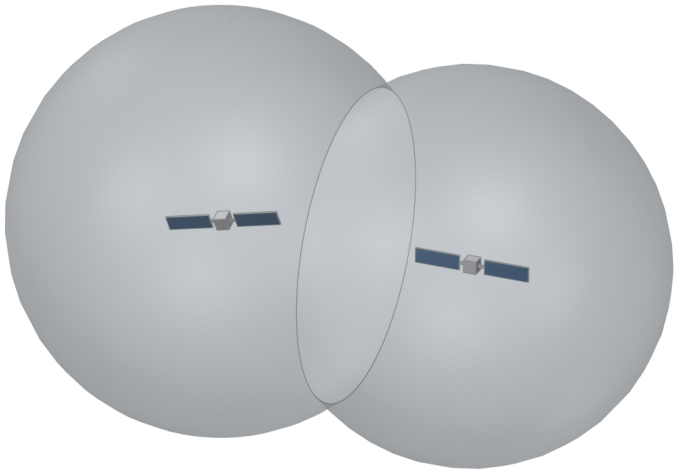
2. geografskom koordinatnom sustavu koji koristi koordinate jednake geografskoj širini, duljini i nadmorskoj visini.

Neka je k broj vidljivih satelita od prijamnika T. Prijemnik T promatrajući poruke dobivene od samo jednog satelita, u vremenu t, izračunava samo jednu pseudoudaljenost i može konstruirati samo jednu jednadžbu (1.1) koja mu omogućava odrediti sferu oko promatranog satelita na kojoj bi se mogao nalaziti (slika 1.8).



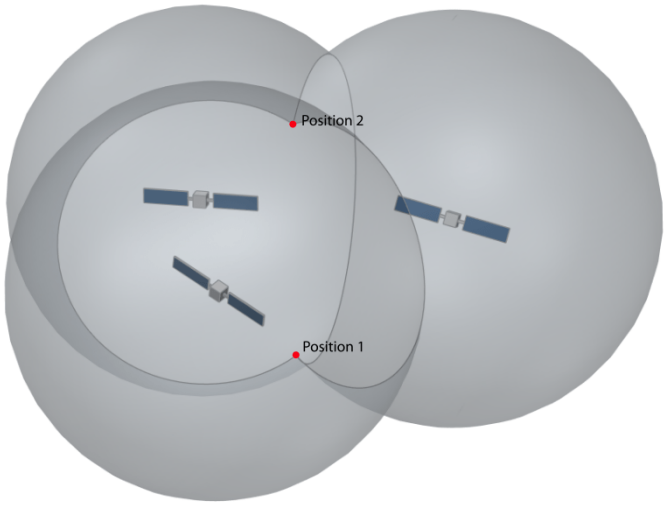
Slika 1.8: Sfera oko promatranog satelita na kojoj bi se prijamnik mogao nalaziti [2]

Uključujući u izračun pridobivene pseudoudaljenosti još jednog satelita dobivamo situaciju prikazanu na Slici 1.9.



Slika 1.9: Sfere oko dva promatrana satelita. Presjek je kružnica na kojoj bi se prijamnik mogao nalaziti. [2]

Uključujući u izračun još jedan satelit, dobivamo situaciju prikazanu na Slici 1.10.



Slika 1.10: Sfere oko tri promatrana satelita. Presjek su dvije točke na kojoj bi se prijamnik mogao nalaziti. [2]

Presjek tri promatrane sfere su dvije točke na kojoj bi se prijamnik mogao nalaziti. Jedna točka se nalazi daleko u svemiru, dok je druga točka točka kandidat položaja prijamnika.

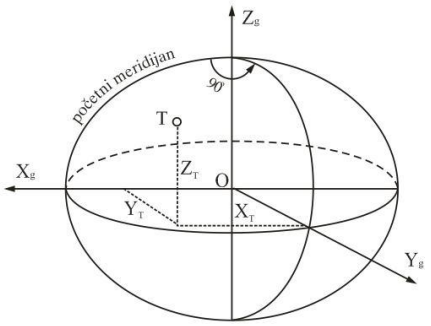
Algebarski, rješavamo sljedeći sustav linearnih jednadžbi u (x, y, z) :

$$\begin{aligned} d_1 &= \sqrt{(x-x_1)^2 + (y-y_1)^2 + (z-z_1)^2} \\ d_2 &= \sqrt{(x-x_2)^2 + (y-y_2)^2 + (z-z_2)^2} \\ d_3 &= \sqrt{(x-x_3)^2 + (y-y_3)^2 + (z-z_3)^2} \end{aligned} \tag{1.2}$$

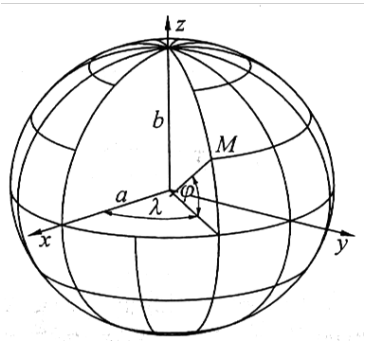
gdje su $1, 2$ i 3, 3 različita satelita, a (x_i, y_i, z_i) pripadajuće koordinate položaja satelita u referentnom ECEF XYZ (egl. Earth-Centered, Earth-Fixed XYZ) koordinatnom sustavu. ECEF XYZ koordinatni sustav ima ishodište u centru Zemljine mase od čega dolazi (engl. Earth-Centered). Sve tri osi (X,Y i Z) koje izlaze iz ishodišta usklađene su s rotacijom Zemlje, tj. rotiraju zajedno sa Zemljom (engl. Earth-Fixed). Z-os prolazi kroz sjeverni pol a XY osi definiraju ekvatorijalnu ravninu.
Zbog kompleksnosti Zemljine površine, uzima se elipsa kao njezina aproksimacija. Trenutni referentni ECEF XYZ koordinatni sustav WGS84 koristi elipsu sljedećih parametara:

- $a = 6378137$,
- $f = \frac{1}{298.257223563}$

i prikazan je slikom 1.11.



Slika 1.11: ECEF X, Y, Z koordinatni sustav (izvor: [32])



Slika 1.12: Geografski koordinatni sustav (izvor: [32])

Svaki prijamnik je sposoban izvesti konverziju iz i u koordinata u referentnom ECEF XYZ sustavu u i iz geografskih (geografska širina, duljina i nadmorska visina) [6]. Dakle, prijamniku su potrebna barem 3 dostupna satelita kako bi odredio položaj. Ali se ipak na stranici 12 se postavlja zahtjev na barem 4.

Primjetimo kako proces određivanja položaja prijamnika indirektno zahtjeva usklađenost satova prijamnika i dostupnih satelita. Satovi svih satelita su međusobno usklađeni usklađenošću stabilnih atomskih satova na satelitima s GPS vremenom. Ukoliko odstupanje ipak postoji, biti će zapisano u navigacijskoj poruci pa se može uzeti u obzir prilikom određivanja položaja prijamnika. Napomenimo, GPS vrijeme nije jednako UTC vremenu. GPS vrijeme je bilo 0 u 06.01.1980. i određeno je protjecanjem vremena u GPS satelitima, tj. njihovim satovima.

Satovi prijamnika nisu iste preciznosti [3] kao satovi satelita. Prijemnici obično koriste satove preciznosti do otprilike 10^{-6} sekundi. Pogreška određivanja vremena od 10^{-6} sekundi dovodi do pogreške u određivanju pseudoudaljenosti od oko 300 metara. Uz pogreške preciznoti sata prijemnika, postoje još pogreške sata prijemnika zbog neapsolutne sinkroniziranosti s GPS vremenom. Uključujući u izračun i obe pogreške sata prijamnika, pseudoudaljenost modeliramo jednadžbom

$$d_i = c \times (t_i' - t_i + d_T) \tag{1.3}$$

gdje d_T predstavlja spomenutu pogrešku korisničkog sata. Budući da se prilikom određivanja položaja, spomenuta pogreška u oznaci d_T ne mijenja u odnosu na satelit koji se promatra, može se izračunati dodavajući ju kao nepoznanicu u sustav

[3] Govorimo o preciznosti, tj. na koliko "decimala" je moguće odrediti vrijeme (koje možda i nije točno ako sat nije dobro usklađen).

jednadžbi (1.2) Dakle, sustav jednadžbi (1.2) prelazi u:

$$\begin{aligned} d_1 &= \sqrt{(x-x_1)^2+(y-y_1)^2+(z-z_1)^2} + c \cdot d_T \\ d_2 &= \sqrt{(x-x_2)^2+(y-y_2)^2+(z-z_2)^2} + c \cdot d_T \\ d_3 &= \sqrt{(x-x_3)^2+(y-y_3)^2+(z-z_3)^2} + c \cdot d_T \end{aligned} \tag{1.4}$$

Kako bi za gornji sustav postojala mogućnost pronalaska rješenja, uvodi se zahtjev na još barem jedan dostupni satelit, što je ukupno 4 (stranica 12). Dobivamo sljedeći sustav jednadžbi u (x, y, z, d_T):

$$\begin{aligned} d_1 &= \sqrt{(x-x_1)^2+(y-y_1)^2+(z-z_1)^2} + c \cdot d_T \\ d_2 &= \sqrt{(x-x_2)^2+(y-y_2)^2+(z-z_2)^2} + c \cdot d_T \\ d_3 &= \sqrt{(x-x_3)^2+(y-y_3)^2+(z-z_3)^2} + c \cdot d_T \\ d_4 &= \sqrt{(x-x_4)^2+(y-y_4)^2+(z-z_4)^2} + c \cdot d_T \end{aligned} \tag{1.5}$$

Upravo opisanim postupkom otklanjamo slučajnu pogrešku nastalu
prilikom mjerenja pseudoudaljenosti s izvorom u pogrešci sata prijamnika. U praksi se može koristiti još veći broj dostupnih satelita što poboljšava točnost određivanja položaja prijamnika. Očekivana pogreška rješenja dobivenog rješavanjem sustava (1.5) je veličina između 10^2 i 10^4 [m]. Tako dobiveno rješenje se profinjuje čime se postiže pogreška veličine 10^1 [m] [4]. Ovim radom se proučava, opisuje, oblikuje i izvodi algoritam za rješavanje sustava (1.5). Naime, rješavanje sustava (1.5) čini temelj procesa određivanja položaja i nužno ga je provesti. U primjenama koje zahtjevaju relativno malu točnost, ono je i dovoljno.
Metode za profinjavanje (poboljšanje) dobivenog rješenja (modeli ispravaka) mogu biti izrazito kompleksne i ovise o primjeni. Svojom kompleksnošću i raznovrsnošću prelaze opseg ovoga rada.

[4] Pofinjenje se obično obavlja kombiniranjem opažanja (modeliranjem nakon primjene algoritma procjene položaja u domeni navigacijske primjene) ili modelima ispravaka primjenjenim na mjerenim pseudoudaljenostima (modeliranjem prije primjene algoritma procjene položaja u domeni navigacijeske primjene).

Poglavlje 2

Algoritam procjene položaja u domeni navigacijske primjene

Postupak procjene položaja satelitskim navigacijskim sustavom traži ispunjavanje sljedećih preduvjeta:

- Korištenje zajedničkog (geoprostornog) referentnog sustava,
- Korištenje zajedničkog vremena (vremenskog okvira) sustava,
- Ispunjavanje pretpostavke o pravocrtnom širenju satelitskih signala jedinstvenom brzinom (brzina svjetlosti u vakuumu).

Uvjeti trebaju biti ispunjeni od strane svih satelita odabranog/odabranih GNSS sustava, ali i korištenog prijamnika. Prvi uvjet je uvijek lako ispuniti. Druga dva se ispunjavaju na razne načine: (E1) modeliranjem prije primjene algoritma za određivanje položaja u navigacijskoj domeni, (E2) korištenjem prekobrojih satelita ili prijamnika (vidi: Poglavlje 1.5), (E3) modeliranjem prilikom primjene algoritma za određivanje položaja u navigacijskoj domeni (samim algoritmom), (E4) modeliranjem nakon primjene algoritma za određivanje položaja u navigacijskoj domeni. Modeliranje prije primjene algoritma za određivanje položaja u navigacijskoj domeni obuhvaća modele ispravaka mjerenih pseudoudaljenosti, tzv. modeli ispravaka [1].

Na primjer, odstupanje sata prijamnika od vremenskog okvira sustava modelira se kao četvrta nepoznanica sustava (vidi: Poglavlje 1.5).

Algoritmom procjene položaja u domeni navigacijske primjene (APP) smatramo svakim algoritmom koji za sustav jednadžbi (1.5) određuje nepoznati položaj prijamnika u koordinatama (x, y, z). Broj jednadži sustava može biti i veći od 4. Tada govorimo o prezasićenom sustavu. Ovisno o odabiru, APP se može temeljiti na rješavanju sustava nelinearnih jednadžbi pronalaženjem rješenja pomoću (M1) me-

[1] Modeli ispravaka se nerjetko zasnivaju na dnevnim vijednostima parametra modela sadržanim u dnevnom almanahu.

tode najmanjih kvadrata (Newton-ova metoda), (M2) metode zatvorene forme, (M3) metode najbližeg susjeda [1].

Općenito, rješava se modificiran sustav jednadžbi (1.5) u koji uključujemo nepoznati parametar (v_1, v_2, v_3, v_4), dodatnu pogreška izračuna:

$$\begin{aligned} d_1 &= \sqrt{(x-x_1)^2+(y-y_1)^2+(z-z_1)^2} + d + v_1 \\ d_2 &= \sqrt{(x-x_2)^2+(y-y_2)^2+(z-z_2)^2} + d + v_2 \\ d_3 &= \sqrt{(x-x_3)^2+(y-y_3)^2+(z-z_3)^2} + d + v_3 \\ d_4 &= \sqrt{(x-x_4)^2+(y-y_4)^2+(z-z_4)^2} + d + v_4 \end{aligned} \tag{2.1}$$

gdje je $d = c \cdot d_T$.
Uz oznake

$$\rho := (d_1, d_2, d_3, d_4)^T \tag{2.2}$$

$$\mathbf{x} := (x, y, z, d_T)^T \tag{2.3}$$

$$\mathbf{x}_{1:3} := \mathbf{x}[1:3] \tag{2.4}$$

$$\mathbf{s}_i := (x_i, y_i, z_i)^T \tag{2.5}$$

$$\mathbf{h}(\mathbf{x}) := \begin{bmatrix} ||(s_1 - \mathbf{x}_{1:3})|| + x_4 \cdot c \\ ||(s_2 - \mathbf{x}_{1:3})|| + x_4 \cdot c \\ ||(s_3 - \mathbf{x}_{1:3})|| + x_4 \cdot c \\ ||(s_4 - \mathbf{x}_{1:3})|| + x_4 \cdot c \end{bmatrix} \tag{2.6}$$

$$\mathbf{v} := (v_i, v_2, v_3, v_4)^T \tag{2.7}$$

(2.1) prelazi u

$$\rho = \mathbf{h}(\mathbf{x}) + \mathbf{v} \tag{2.8}$$

2.1 Iterativna metoda najmanjih kvadrata

Primjetimo kako je $\mathbf{h}(\mathbf{x})$ definiran s (2.6) jednak vektoru udaljenosti satelita i prijamnika za prave vrijednosti $\mathbf{x}$, $\bar{\mathbf{x}}$ i pogrešku sata $d_T = 0$.

Općenito oblikovanje algoritma za određivanje položaja u domeni navigacijske primjene nema utjecaj na pogreške tipa 2, već samo pogreške tipa 1 (stranica 11). Također, pretpostavlja se kako su otklonjene sve pogreške tipa 1 koje imaju izvor u pogreškama izračuna pseudoudaljenosti (osim pogrešaka sata prijamnika) (stranica 11). Ostaje još samo modelirati pogreške koje za izvor imaju trenutni položaj satelita

dostupnih za izračunavanje rješenja sustava.
U tu svrhu, vektor pogrešaka $\mathbf{v}$ ocjenjujemo ovisno o $\mathbf{x}$. Funkciju ocjene pogrešaka, u oznaci $\mathbf{p}(\mathbf{x})$, dobivamo iz (2.8). Uz oznaku $\mathbf{y} := \rho$ (2.8) prelazi u

$$\mathbf{y} = \rho = \mathbf{h}(\mathbf{x}) + \mathbf{p}(\mathbf{x}) \tag{2.9}$$

iz čega slijedi formulu za $\mathbf{p}(\mathbf{x})$:

$$\mathbf{p}(\mathbf{x}) := \mathbf{y} - \mathbf{h}(\mathbf{x}) \tag{2.10}$$

$\mathbf{p}(\mathbf{x})$ opisuje pogrešku razlike u procjeni parametra $\mathbf{x}$ od stvarne vrijednosti. Što je aproksimacija potrebnih vrijednosti za izračun rješenja matrične jednadžbe (2.8) točnija, to je $\mathbf{p}(\mathbf{x})$ bliže nuli za pravu vrijednost $\bar{\mathbf{x}}$. Aproksimaciju za $\bar{\mathbf{x}}$, u oznaci $\hat{\mathbf{x}}$, pronalazimo tražeći nultočke funkcije $\mathbf{p}(\mathbf{x})$. U praksi je uobičajeno da mjerenja sadrže pogreške i tada $\mathbf{p}(\mathbf{x})$ uopće ne mora imati nultočke i $\hat{\mathbf{x}}$ ne možemo pronaći tražeći nultočke funkcije $\mathbf{p}(\mathbf{x})$.

Konceptualno, ideja metode najmanjih kvadrata je pronalaženje $\hat{\mathbf{x}}$ traženjem minimuma funkcije $\mathbf{p}(\mathbf{x})$, tj.

$$\hat{\mathbf{x}} := \arg\min_{\mathbf{x}} \mathbf{p}(\mathbf{x})^T \mathbf{p}(\mathbf{x}) \tag{2.11}$$

Problem opisan jednadžbom (2.11) nije linearan pa ne postoji općeniti način pronalaska njegovog rješenja.

U slučaju kada su funkcija koju treba minimiziati dovoljno dobra, rješenja problema (2.11) moguće je dobiti iterativnim postupkom. Ideja iterativnog postupka je, počevši s unaprijed odabranim $\mathbf{x}_0$ (početna vrijednost), računati $\mathbf{x}_1, \mathbf{x}_2, \ldots$ sve dok se novoizračunate vrijednosti ne prestanu mijenjati ili postanu dovoljno bliske prethodnoj, tj. $\|\mathbf{x}_k - \mathbf{x}_{k-1}\| < \delta$ za dovoljno male $\delta > 0$. δ još nazivamo i zaustavni kriterij. Cilj iterativnog postupka je u koraku k profiniti trenutnu vrijednost $\mathbf{x}_k$ kako bi postala bliža konačnom rješenju sustavu.
Uvijek postoji mogućnost da zaustavljanje iterativnog postupka ne rezultira pronalaskom rješenja ili njegove prihvatljive aproksimacije. Odabirom dobre početne vrijednosti znatno se umanjuje mogućnost zaustavljanja iterativnog postupka bez pronalaska prihvatljivog rješenja sustava.

Prikladni iterativni postupak rješavanja problema (2.11) je Newton-Gaussova metoda (iterativna metoda najmanjih kvadrata). Newton-Gaussova metoda linearizira $\mathbf{p}(\mathbf{x})$ u okolini od $\mathbf{x_k}$ pomoću prvog člana razvoja funkcije u Taylorov red u točki $\mathbf{x}_k$:

$$\mathbf{p}(\mathbf{x_k} + \Delta\mathbf{x_k}) \approx \mathbf{p}(\mathbf{x_k}) + \mathbf{p}'(\mathbf{x_k}) \cdot \Delta\mathbf{x_k} \tag{2.12}$$

$\Delta\mathbf{x_k}$ se odabire na način tako da

$$lim_{k\to\infty} (\mathbf{p}(\mathbf{x_k})) = 0$$

jer za pravu vrijednost $\mathbf{x}$ izraz $\mathbf{p}(\mathbf{x}) = 0$ ili poprima svoj minimum ukoliko postoje pogreške točnosti vrijednosti korištenih za konstrukciju sustava.

Sada, za $\mathbf{p}(\mathbf{x_{k+1}}) := \mathbf{p}(\mathbf{x_k} + \Delta\mathbf{x_k}) \approx \mathbf{p}(\mathbf{x_k}) + \mathbf{p'}(\mathbf{x_k}) \cdot \Delta\mathbf{x_k}$ želimo da je što bliže 0. Dakle, $\Delta\mathbf{x_k}$ odabiremo tražeći minimum funkcije

$$\mathbf{p}(\mathbf{x_k}) + \mathbf{p'}(\mathbf{x_k}) \cdot \Delta\mathbf{x_k} \tag{2.13}$$

u $\Delta\mathbf{x_k}$.

Označimo sada s $\mathbf{J}_k := \mathbf{p'}(\mathbf{x_k}) = \mathbf{h'}(\mathbf{x_k})$. (2.13) prelazi u

$$\mathbf{J}_k\Delta\mathbf{x_k} + \mathbf{p}(\mathbf{x_k}) \tag{2.14}$$

čiji je minimun dan s (stranica 38)

$$\Delta\mathbf{x_k} = -(\mathbf{J}_k^T\mathbf{J}_k)^{-1}\mathbf{J}_k^T\mathbf{p}(\mathbf{x_k}) \tag{2.15}$$

Izraz za $\mathbf{x_{k+1}}$ je sljedeći:

$$\mathbf{x_{k+1}} = \mathbf{x_k} - (\mathbf{J}_k^T\mathbf{J}_k)^{-1}\mathbf{J}_k^T\mathbf{p}(\mathbf{x_k}) \tag{2.16}$$

Prilikom izvedbe algoritma, potrebno je primjereno odrediti početnu vijednost $\mathbf{x_0}$, te kasnije iterirati po formuli (2.16). Uz odabir prikladanog $\mathbf{x}_0$, dovoljno blizu rješenju i ako je druga derivacija od $\mathbf{p}$ u točki $\bar{\mathbf{x}}$ dovoljno mala, niz $\mathbf{x_0}, \mathbf{x_1}, \mathbf{x_2} \ldots$ konvergira prema $\bar{\mathbf{x}}$. Izračun $\mathbf{J}_k$ za idealan slučaj $d = 0$ je dan u Dodatku A.

Algoritam iterativne metode najmanjih kvadrata je dan u nastavku.

Algoritam 2.1: Iterativna metoda najmanjih kvadrata

Data: $\mathbf{p}(\mathbf{x}), \mathbf{x}_0, \delta$
Result: $\hat{\mathbf{x}}$

```
k = 0 ;
while ||x_k − x_{k−1}|| ≥ δ do
    J_k = p'(x_k) ;
    Δx_k = −J_k^{−1} · p(x_k) ;
    x_{k+1} = x_k + Δx_k ;
    k++;
end
x̂ = x_k
```

Prilikom korištenja gornjeg algoritma za određivanje položaja objekta, za $\mathbf{x}_0$ se mogu uzeti koordinate središta zemlje jer su jednadžbe za određivanje položaja dovoljno bliske linearnima.

Ako je poznato da su vrijednosti koje koristimo za konstrukciju jednadžbi za određivanje položaja (1.5) za pojedine jednadžbe točnije, prirodno je pridjeliti im

prednost pred ostalima. Važnost pojedine jednadžbe modeliramo pridavanjem težina pojedinoj jednadžbi. Svakoj jednadžbi se pridružuje težina σ_i koja je proporcionalna preciznosti vrijednosti korištenih prilikom njezine konstrukcije. Najčešće način pronalaženja odgovarajućih težina je korištenjem kovarijancone matrice vektora pogrešaka $\mathbf{v}$ (jednadžba (2.7)),u oznaci $\Sigma := cov(\mathbf{v})$. Minimizacijski problem (2.11) prelazi u

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \mathbf{p}(\mathbf{x})^T \Sigma^{-1} \mathbf{p}(\mathbf{x}) \tag{2.17}$$

Sada, Algoritam 1 prelazi u algoritam 2.

Algoritam 2.2: Iterativna metoda težinskih najmanjih kvadrata

Data: $\mathbf{p}(\mathbf{x}), \mathbf{x}_0, \delta, \Sigma$
Result: $\hat{\mathbf{x}}$
$k = 0$;
while $\|\mathbf{x}_k - \mathbf{x}_{k-1}\| \geq \delta$ **do**
  $\mathbf{J}_k = \mathbf{p}'(\mathbf{x}_k)$;
  $\Delta\mathbf{x}_k = -(\Sigma^{-\frac{1}{2}}\mathbf{J}_k)^{-1}(\Sigma^{-\frac{1}{2}}(\mathbf{p}(\mathbf{x}_k))$;
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k$;
  $k++$;
end
$\hat{\mathbf{x}} = \mathbf{x_k}$

Procjenitelj za $\bar{\mathbf{x}}$ dobiven težinskom metodom najmanjih kvadrata, jednakost (2.17), ima najmanju varijancu među svim procjeniteljima za $\bar{\mathbf{x}}$. Ukoliko pretpostavljamo normalnu distribuciju vektora pogrešaka, procjenitelj (2.17) postaje procjenitelj metode najbližeg susjeda (Potpoglavlje 2.3 i MLE procjenitelj).

Prilikom korištenja iterativne metode najmanjih kvadrata potrebno je modelirati distribuciju vektora pogrešaka, točnije kovarijanconu matricu Σ. Također, potrebno je pripazati na velike pogreške u određivanju vrijednosti pomoći kojih se gradi sustav jednadžbi (1.5) i netipične vrijednosti (engl. outliners) koji se uklanjaju prije primjene algoritma.

Sljedeće poglavlje opisuje izvedbu upravo opisanog algoritma 2 i analizu njegove točnosti. Prije same izvedbe, navodi se zanimljiva posljedica analize pogreške metode najmanjih kvadrata i pregled još nekih metoda za rješavanje sustava (1.5).

Analize pogreške metode najmanjih kvadrata

Uz oznake kao do sada, neka $\bar{\mathbf{y}}$ predstavlja prave udaljenosti između satelita i promatranog objekta (prijamnika) i $\hat{\mathbf{y}}$ izračunate pseudoudaljenosti. Vrijedi $\hat{\mathbf{y}} = \bar{\mathbf{y}} + \Delta\mathbf{y}$. Promotrimo idealan slučaj za metodu iterativnih najmanjih kvadrata (Algoritam 2.1.)

gdje je $\delta = 0$. Neka je $\hat{\mathbf{x}} = \bar{\mathbf{x}} + \Delta\mathbf{x}$ rješenje dobiveno metodom najmanjih kvadarata, tj. $\hat{\mathbf{x}} = \mathbf{x}_{k'}$ i $\forall m \geq k', \mathbf{x}_m = \mathbf{x}_{m+1}$. Uvrštavanjem $\mathbf{x}_k = \hat{\mathbf{x}}$ i $\mathbf{y} = \hat{\mathbf{y}}$ u jednadžbu (2.16) dobivamo

$$\begin{aligned}
\mathbf{x_{k+1}} &= \mathbf{x_k} - (\mathbf{J}_k^T\mathbf{J}_k)^{-1}\mathbf{J}_k^T\mathbf{p}(\mathbf{x_k}) \\
\mathbf{x_{k+1}} - \mathbf{x_k} &= -(\mathbf{J}_k^T\mathbf{J}_k)^{-1}\mathbf{J}_k^T\mathbf{p}(\mathbf{x_k}) \\
0 &= -(\mathbf{J}_k^T\mathbf{J}_k)^{-1}\mathbf{J}_k^T\mathbf{p}(\bar{\mathbf{x}} + \Delta\mathbf{x}) \\
0 &= (\mathbf{J}_k^T\mathbf{J}_k)^{-1}\mathbf{J}_k^T(\mathbf{h}(\bar{\mathbf{x}} + \Delta\mathbf{x}) - (\bar{\mathbf{y}} + \Delta\mathbf{y}))
\end{aligned}$$

Matrica $\mathbf{J}_k$ predstavlja funkciju koja ovisi o parametru $\mathbf{x}$. Ona nije konstantna. Kako se pretpostavlja da je $\Delta\mathbf{x}$ dovoljno blizu nule, opravdano je smatrati $\mathbf{J} := \mathbf{J}_k$ konstantom za susjedstvo od $\bar{\mathbf{x}}$ radijusa $\Delta\mathbf{x}$. Sada se $\mathbf{h}$ u okolini točke $\bar{\mathbf{x}}$ može linearizirati na sljedeći način:

$$\mathbf{h}(\mathbf{x} + \delta) = \mathbf{h}(\mathbf{x}) + \mathbf{J}\delta, \delta > 0$$

.
Dobivamo

$$\begin{aligned}
0 &= (\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T(\mathbf{h}(\bar{\mathbf{x}}) + \mathbf{J}\Delta\mathbf{x} - (\bar{\mathbf{y}} + \Delta\mathbf{y})) \\
0 &= (\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T(\mathbf{J}\Delta\mathbf{x} - \Delta\mathbf{y}) \\
(\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T\mathbf{J}\Delta\mathbf{x} &= (\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T\Delta\mathbf{y} \\
\Delta\mathbf{x} &= (\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T\Delta\mathbf{y}
\end{aligned}$$

Uz pretpostavku normalnosti pogreške izračunavanja pseudoudaljenosti,
$\Delta\mathbf{y} \sim N(0, \Sigma)$, imamo

$$\Delta\mathbf{x} \sim N(0, (\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T\Sigma\mathbf{J}(\mathbf{J}^T\mathbf{J})^{-1}) \tag{2.18}$$

Uz $\Sigma = \sigma^2 I$, $\Delta\mathbf{x} \sim N(0, \sigma^2(\mathbf{J}^T\mathbf{J})^{-1})$. U kontekstu satelitske navigacije, $(\mathbf{J}^T\mathbf{J})^{-1}$ nazivamo DOP (engl. Dilution of Precision) matricom. Iz DOP matrice moguće je izvesti različite mjere kvalitete sazviježđa satelita u danom trenutuku za dani položaj.

1. GDOP $= \sqrt{tr(\mathbf{J}^T\mathbf{J})^{-1}}$
2. PDOP $= \sqrt{tr((\mathbf{J}^T\mathbf{J})^{-1}_{(1:3,1:3)})}$
3. HDOP $= \sqrt{tr((\mathbf{J}^T\mathbf{J})^{-1}_{(1:2,1:2)})}$
4. VDOP $= \sqrt{(\mathbf{J}^T\mathbf{J})^{-1}_{(3,3)}}$

5. TDOP $= \sqrt{(\mathbf{J}^T\mathbf{J})^{-1}_{(4,4)}}$

Opširnije o mjerama kvalitete sazviježđa moguće je naći u Dodatku B.

Dakle, uz neke pretpostavke, iz Jakobijeve matrice funkcije $\mathbf{h}$, $\mathbf{J}$, može se saznati mnogo o kvaliteti određivanja položaja za sustav jednadžbi (1.5), veličini pogreške određivanja s izvorom u kvaliteti sazviježđa. Izračuni gornjih mjera su točni onoliko koliko su pretpostavke o jednakostima za varijance $\Delta\mathbf{y}$ i $\Delta\mathbf{x}$ istinite.

Primjenu metode najmanjih kvadrata moguće je pronaći na stranici 38.

2.2 Metoda zatvorene forme

Metoda zatvorene forme pronalazi izravno rješenje sustava (1.5) [1]. Za razliku od iterativnih metoda, metode zatvorene forme ne zahtjevaju postavljanje početnog rješenja $\mathbf{x}_0$ i uvjeta zaustavljanja δ. Rješenje je egzaktno i ne postoji mogućnost pronalaska krivog rješenja (lokalnog minimuma). Ukoliko postoji više rješenja sustava, zatvorena forma pronalazi sve.

Budući da se metodama zatvorene forme otežano modeliraju pogreške mjerenja, one se ne koriste za pronalazak krajnjeg rješenja sustava. Ipak, zatvorena forma je korisna u pronalasku početnog rješenja sustava iterativnog postupka, istraživanje, razvoj i grafičko predočavanje.

Za mjerene pseudoudaljenosti $y_1, y_2, \dots y_n$ i nepoznatu pogrešku sata prijamnika x_4, rješenje problema najmanjih kvadrata danog jednadžbama

$$\begin{aligned} y_1 &= \|\mathbf{s}_1 - \mathbf{x}_{1:3}\| + \mathbf{x}_4 \\ y_2 &= \|\mathbf{s}_2 - \mathbf{x}_{1:3}\| + \mathbf{x}_4 \\ y_2 &= \|\mathbf{s}_3 - \mathbf{x}_{1:3}\| + \mathbf{x}_4 \\ &\vdots \\ y_n &= \|\mathbf{s}_n - \mathbf{x}_{1:3}\| + \mathbf{x}_4 \end{aligned} \tag{2.19}$$

u $x_{1:3}$ i x_4 dano je sljedećim zatvorenim formama

$$\begin{aligned} \mathbf{x}_{1:3} &= \mathbf{d}\lambda + \mathbf{e} \\ \mathbf{x}_4 &= f\lambda + g \end{aligned} \tag{2.20}$$

gdje λ dobivamo rješavanjem sljedeće jednadžbe

$$(\|\mathbf{d}\|^2 - f^2)\lambda^2 + (2\mathbf{d}^T\mathbf{e} - 2fg - 1)\lambda + \|\mathbf{e}\| - g^2 = 0. \tag{2.21}$$

$\hat{\mathbf{x}}$ je rješenje sustava (2.19) ako i samo ako je rješenje zatvorene forme (2.20).
Parametre $\mathbf{d}$, $\mathbf{e}$, f i g zatvorene forme (2.20) dobivamo iz sustava (2.19) sljedećim

nizom pretvorbi:

$$\begin{aligned}(y_1 - x_4)^2 &= \|\mathbf{s}_1 - \mathbf{x}_{1:3}\|^2 \\ (y_2 - x_4)^2 &= \|\mathbf{s}_2 - \mathbf{x}_{1:3}\|^2 \\ (y_3 - x_4)^2 &= \|\mathbf{s}_3 - \mathbf{x}_{1:3}\|^2 \\ &\vdots \\ (y_n - x_4)^2 &= \|\mathbf{s}_n - \mathbf{x}_{1:3}\|^2\end{aligned} \tag{2.22}$$

$$\begin{aligned}y_1^2 - 2y_1x_4 + x_4^2 &= \|\mathbf{s}_1\|^2 - 2\mathbf{s}_1^T\mathbf{x}_{1:3} + \|\mathbf{x}_{1:3}\|^2 \\ y_2^2 - 2y_2x_4 + x_4^2 &= \|\mathbf{s}_2\|^2 - 2\mathbf{s}_2^T\mathbf{x}_{1:3} + \|\mathbf{x}_{1:3}\|^2 \\ y_3^2 - 2y_3x_4 + x_4^2 &= \|\mathbf{s}_3\|^2 - 2\mathbf{s}_3^T\mathbf{x}_{1:3} + \|\mathbf{x}_{1:3}\|^2 \\ &\vdots \\ y_n^2 - 2y_nx_4 + x_4^2 &= \|\mathbf{s}_n\|^2 - 2\mathbf{s}_n^T\mathbf{x}_{1:3} + \|\mathbf{x}_{1:3}\|^2\end{aligned} \tag{2.23}$$

Uz

$$\lambda := \|\mathbf{x_{1:3}}\|^2 - x_4^2 \tag{2.24}$$

dobivaju se linearne jednadžbe u $\mathbf{x_{1:3}}$, x_4 i λ

$$-\lambda + y_i^2 - \|\mathbf{s}_i\|^2 = 2y_ix_4 - 2\mathbf{s}_i^T\mathbf{x}_{1:3} \tag{2.25}$$

koje čine sustav

$$\begin{bmatrix}2\mathbf{s}_1^T & -2y_1 \\ 2\mathbf{s}_2^T & -2y_2 \\ 2\mathbf{s}_3^T & -2y_3 \\ \vdots & \\ 2\mathbf{s}_n^T & -2y_n\end{bmatrix}\begin{bmatrix}\mathbf{x}_{1:3} \\ x_4\end{bmatrix} = \begin{bmatrix}1 \\ 1 \\ 1 \\ \vdots \\ 1\end{bmatrix}\lambda + \begin{bmatrix}\|\mathbf{s}_1\|^2 & -y_1^2 \\ \|\mathbf{s}_2\|^2 & -y_2^2 \\ \|\mathbf{s}_3\|^2 & -y_3^2 \\ \vdots & \\ \|\mathbf{s}_n\|^2 & -y_n^2\end{bmatrix}. \tag{2.26}$$

Naposljetku, za

$$\begin{bmatrix}\mathbf{d}\\ f\end{bmatrix} := \mathbf{p} = \begin{bmatrix}2\mathbf{s}_1^T & -2y_1\\ 2\mathbf{s}_2^T & -2y_2\\ 2\mathbf{s}_3^T & -2y_3\\ \vdots & \\ 2\mathbf{s}_n^T & -2y_n\end{bmatrix}^T \begin{bmatrix}1\\ 1\\ 1\\ \vdots\\ 1\end{bmatrix} \tag{2.27}$$

$$\begin{bmatrix}\mathbf{e}\\ g\end{bmatrix} := \mathbf{q} = \begin{bmatrix}2\mathbf{s}_1^T & -2y_1\\ 2\mathbf{s}_2^T & -2y_2\\ 2\mathbf{s}_3^T & -2y_3\\ \vdots & \\ 2\mathbf{s}_n^T & -2y_n\end{bmatrix}^T \begin{bmatrix}\|\mathbf{s}_1\|^2 & -y_1^2\\ \|\mathbf{s}_2\|^2 & -y_2^2\\ \|\mathbf{s}_3\|^2 & -y_3^2\\ \vdots & \\ \|\mathbf{s}_n\|^2 & -y_n^2\end{bmatrix} \tag{2.28}$$

dobivamo (2.20).

2.3 Metoda najbližeg susjeda

Metode najbližeg susjeda (maksimalne vjerodostojnosti) opisuju pogrešku mjerenja uvjetnom vjerojatnošću, $\mathbf{p}(y|\mathbf{x})$. $\mathbf{p}(y|\mathbf{x})$ je vjerojatnost da je pseudoudaljenost y izmjerena na položaju s koordinatama $\mathbf{x}_{1:3}$ s pogreškom u izvoru sata prijamnika jednakoj x_4 [1]. Ukoliko se $\mathbf{x}$ postavi za varijablu, a y za konstantu, dobivamo funkciju maksimalne vjerodostojnosti (ML), u oznaci $L(\mathbf{x}|Y) := \mathbf{p}(y|\mathbf{x})$.

Za problem određivanja položaja opisanog jednadžbom (2.8), lako se dobiva ekvivalentan problem određivanja položaja maksimalne vjerodostojnosti. Budući da vrijedi:

$$\mathbf{v} = \rho - \mathbf{h}(\mathbf{x}) \tag{2.29}$$

i uz pretpostavku da je $\mathbf{v}$ slučajni vektor poznate distribucije funkcije vjerojatnosti $\mathbf{p_v}$, imamo:

$$\mathbf{p}(y|\mathbf{x}) = \mathbf{p_v}\left(\rho - \mathbf{h}(\mathbf{x_{1:3}})\right) \tag{2.30}$$

Kada problem određivanja položaja zadamo s (2.30), $\hat{\mathbf{x}}$ pronalazimo pomoću procjenitelja maksimalne vjerodostojnoti za $\mathbf{x}$, tj. $\hat{\mathbf{x}}$ je takav da vrijedi

$$L(\hat{\mathbf{x}}|y) := \max_{\tilde{\mathbf{x}}}(L(\tilde{\mathbf{x}}|y)) \tag{2.31}$$

gdje $\tilde{\mathbf{x}}$ predstavljaju sve dozvoljene koordinate položaja objekta na Zemlji i u zraku. Za poznata mjerenja psudoudaljenosti, $\hat{\mathbf{x}}$ se može pronaći metodom nelinearne optimizacije [1].

Može se pokazati kako metoda najbližeg susjeda i metoda težinskih najmanjih kvadrata daju isto rješenje za $\hat{\mathbf{x}}$ uz normalnu distribuiranost vektora $\mathbf{v}$ i matrice težina postavljene na $\Sigma^{-1} = cov(\mathbf{v})^{-1}$. Naime, za:

$$\mathbf{p}_v(\mathbf{z}) = C \exp\left(-\frac{1}{2}\mathbf{z}^T\Sigma^{-1}\mathbf{z}\right)$$
$$C = (2\pi)^{-\frac{n}{2}} det(\Sigma)^{-\frac{1}{2}}$$

dobivamo:

$$\mathbf{p}(y|\mathbf{x}) = \mathbf{p}_{\mathbf{v}}\left(\rho - \mathbf{h}(\mathbf{x})\right) = C \exp\left(-\frac{1}{2}(\rho - \mathbf{h}(\mathbf{x}))^T\Sigma^{-1}(\rho - \mathbf{h}(\mathbf{x}))\right) \tag{2.32}$$

Budući da je Σ pozitivno definitna matrica, argument eksponencijalne funkcije gornjeg izraza je uvijek negativan. Dakle, problem maksimizacije funkcije (2.32) jednak je minimizaciji izraza:

$$(\rho - \mathbf{h}(\mathbf{x}))^T\Sigma^{-1}(\rho - \mathbf{h}(\mathbf{x}) = \mathbf{v}^T\Sigma^{-1}\mathbf{v} = \mathbf{p}^T(\mathbf{x})\Sigma^{-1}\mathbf{p}(\mathbf{x}). \tag{2.33}$$

Dakle, $\hat{\mathbf{x}} = \arg\min_x \left(\mathbf{p}^T(\mathbf{x})\Sigma^{-1}\mathbf{p}(\mathbf{x})\right)$ što odgovara izrazu (2.17) uz matricu težina jednaku $cov(\mathbf{v})^{-1}$.

Poglavlje 3

Programski određen GPS prijamnik

Za pokretanje izvedbe procesa određivanja položaja, potrebno je prvo prikupiti podatke koji će biti korišteni. Potrebne podatke je moguće prikupiti u *RINEX* obliku koristeći civilni programski određen GPS prijamnik praktično izveden na vlastitom računalu ili koristeći podatke s međunarodne GNSS referentne postaje (engl. International GNSS Service reference station, IGS reference station). Kako bi se formulirali podatci potrebni za ulaza izvedenog procesa određivanja položaja željenog formata, potrebno je obraditi prikupljene *RINEX* podatke i prebaciti u željeni oblik, tekstualnu datoteku s 1 (pseudoudaljenosti) ili 3 stupca podataka (x,y i z koordinate položaja satelita u ECEF XYZ koordinatama [1]).
Obradu prikupljenih *RINEX* podatka je moguće napraviti koristeći programski određen GPS prijamnik praktično izvedenog na vlastitom računalu.

3.1 Model programski određenog radioprijamnika

Svaki radioprijamnik obavlja procesiranje signala i informacija u četiri osnovne domene:

- domena pretvorbe elektromagnetskog vala u električni signal (u anteni),
- domena visokih (radijskih) frekvencija, u kojoj se obrađuje primljeni modulirani signal te obavlja demodulacija i prijenos u osnovno frekvencijsko područje,
- domena osnovnog frekvencijskog područja, u kojoj se procesiraju signali nosioci informacija i iz njih izdvajaju same informacije,
- domena aplikacijskog procesiranja, u kojoj se izdvojene informacije procesiraju s ciljem predstavljanja korisniku u za njega prihvatljivom obliku.

[1] WGS84 koordinatama

Ponekad se obrada u prve tri domene naziva jednim imenom obrada signala i informacija u frekvencijskoj domeni.



Slika 3.1: Funkcionalni model programski određenog radio prijamnika za satelitsku navigaciju

Za područje satelitske navigacije, domena visokih frekvencija izdvojit će i digitalizirati signale koji prenose PRN kodne sekvence i navigacijsku poruku. Nizovi brojeva prosljeđuju se u domenu osnovnog frekvencijskog područja koja identificira i izdvaja prenošene informacije. U satelitskom navigacijskom prijamniku, u ovoj se domeni postupakom unakrsne korelacije primljenih i lokalno generiranih PRN kodnih sekvenci određuju pseudoudaljnosti i izdvajaju elementi navigacijske poruke. U domeni aplikacijskog procesiranja, izlaz osnovnog frekvencijskog područja bit će obrađeni s ciljem predstavljanja (spremanja) informacija u korisniku razumljivom obliku (slika 3.1).

3.2 Pojam programski određenog radioprijamnika

Tradicionalni prijamnik za satelitsku navigaciju je izveden sklopovski. Elektronički sklopovi posebne namjene obavljaju ciljane funkcije unutar segmenata prijamnika. Pri tome, konstrukcija i izvedba sklopova definira uspješnost primjene matematičkih modela u ispunjavanje traženih funkcionalnosti, odnosno postavljenih zahtjeva na kvalitetu procesiranja signala i informacija.

Elektronički sklopovi su po svojoj prirodi nesavršeni i ograničeni. Jednom konstruirani i izvedeni elektronički sklopovi posebne namjene ne mogu se lako značajnije promijeniti. Pokaže li se potreba za proširivanjem ili prilagođavanjem novom statusu sustava kao cjeline, potrebno je napuštanje izvedbe starog sustava i konstrukcija ili kupnja potpuno nove. U slučaju satelitske navigacije, tradicionalni GPS prijamnik, u kojem je generiranje PRN satelitskih sekvenci izvedeno sklopovskim načinom, uvođenje novih satelita i modernizacija sustava izazivaju napuštanje starog i konstrukciju ili kupnju potpuno novog i kompatibilnog GPS prijamnika.

Dvadesete godine dvadesetog stoljeća uvode novi koncept radiokomunikacijske tehnologije. Reducira se broj elektroničkih sklopova posebne namjene i uvode programske komponente za obradu signala i informacija. Time se omogućava lakše

praćenje promjena sustava i izravnija primjena matematičkih modela u algoritamskom obliku na računalnim podlogama opće namjene, npr. osobna računala ili pametni telefoni. Novi koncept se naziva programski određen radio (engl. Software-Defined Radio, SDR). Lakoća prilagodbe promjenama, omogućila je SDR-u da ubrzo postane standard u radiokomunikacijskoj industriji. Brojni uređaju od pametnih telefona do radijskih i televizijskih prijamnika su izvedeni u obliku SDR-a. Takva izvedba im omogućava postizanje bolje prilagodljivosti, proširivosti, iskorištenja energije i lakše komunikacije s drugim računalnim uređajuma.

Programska izvedba prijamnika za satelitsku navigaciju zanimljiva je sa stajališta računarne znanosti. Primjena algoritama za procesiranje signala i informacija podržava raspodjeljivanje arhitekture sustava. Potpuno procesiranje više ne treba biti u potpunosti izvedeno na jednom uređaju (npr. pametnom telefonu ili samostalnom GNSS prijamniku) pa se dijelovi postupka obrade prebacuju na druge uređaje. Svaki korišteni uređaj svoj dio obrade obavlja kvalitetnije i točnije uz jednostavnije korištenje izvora dodatnih informacija koje mogu pridonijeti poboljšanju točnosti procjene položaja [13, 14]. Navedeni pristup omogućava korištenje računalnog okruženja u oblaku što dopušta da se prijamniku ostavi samo obrada signala i informacija u frekvencijskoj domeni. Izlaz obrade signala i informacija u frekvencijskoj domeni pohranjuje se u binarnom obliku u *RINEX* formatu.

RINEX (engl. Receiver Independent Exchange Format) je općeprihvaćena definicija pohranjivanja izlaza obrade (navigacijskih) satelitskih signala u frekvencijskoj domeni (neobrađeni podatci satelitske navigaciju). Definiranje općeprihvaćenog načina pohranjivanja omogućava lako prebacivanje dijelova obrade na druge uređaje u svrhu poboljšanju točnosti procjene položaja [13, 14]. *RINEX* se mijenja kroz vrijeme obuhvaćajući promjene GNS sustava. Trenutna verzija je 3.03 iz 2015 [15].

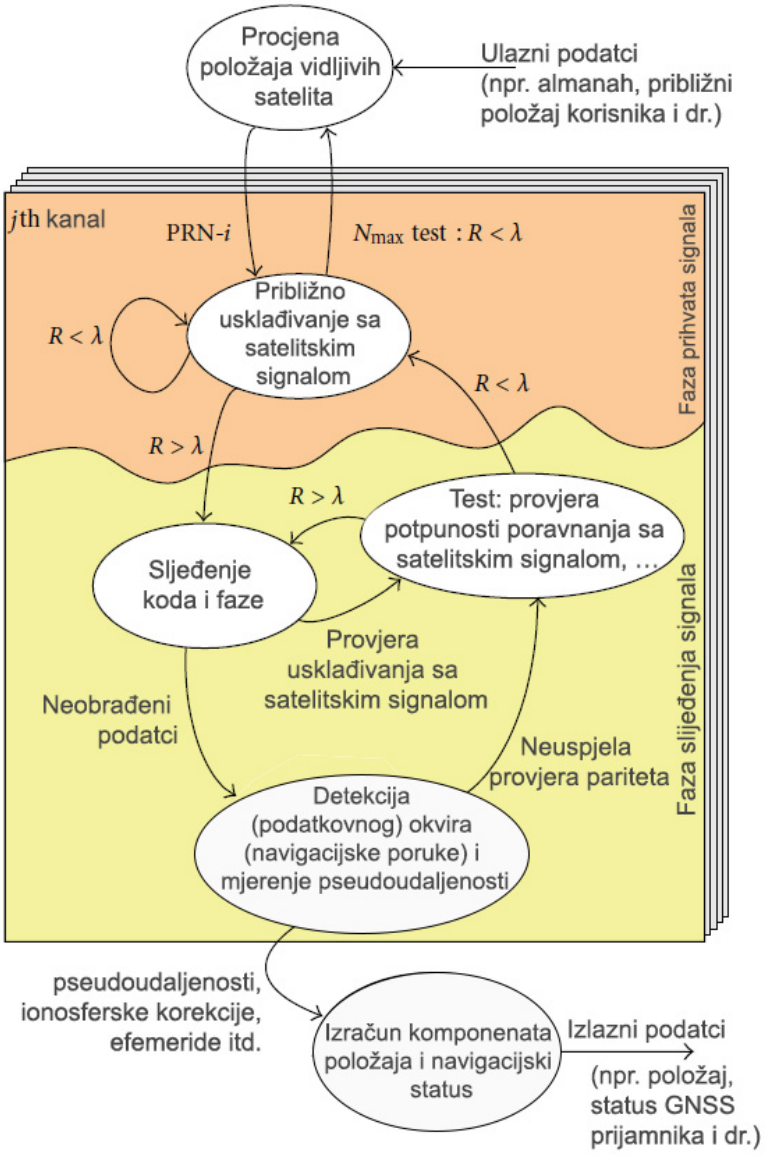
3.3 Programski određen GPS prijamnik

Programsko određen GPS prijamnik predstavlja vrstu programski određenog radioprijamnika posebne namjene za procjenu položaja satelitskim navigacijskim sustavima.

Posebnosti programski određenog radioprijamnika za potrebe satelitske navigacije izražene su karakterističnim postupcima procesiranja signala i informacija u domeni osnovnog frekvencijskog područja i domeni navigacijskog (aplikacijskog) procesiranja. Karakteristični postupci su vezani za:

- prihvat signala (engl. Acquisition), prepoznavanje PRN kodne sekvence pojedinačnog satelitskog signala,
- sljeđenje signala (engl. Tracking), vremensko usklađivanje s primljenim signalom, za potrebe kasnijeg određivanja pseudoudaljenosti,

- procjena vidljivosti satelita,
- procjena položaja, brzine i vremena.



Slika 3.2: Procesiranje signala u domeni osnovnog frekvencijskog područja

Procesiranje signala u domeni osnovnog frekvencijskog područja (slika 3.2) obuhvaća prihvat signala, sljeđenje signala, izdvajanje navigacijske poruke i određivanje pseudoudaljenosti. Obavlja se na razini komunikacijskog kanala, tj. za svaki pojedinačni satelitski signal. U slučaju gubitka vremenske usklađenosti s primljenim

signalom, prijamnik će prijeći na prihvat signala, dok se ne stvore uvjet za ponovni prijelaz u fazu slijeđenja. U slučaju potpunog gubitka signala, prijamnik ponovo započinje postupak prihvata.

Algoritmi obrade signala u domeni osnovnog frekvencijskog područja ovise o spremnosti procjene vidljivosti satelita koja se u navigacijskoj domeni zasniva na pojednostavljenom opisu satelitskih putanja, efemeridama. Almanah o statusu satelita u sazviježđu, kao i satelitske efemeride, prenosi se navigacijskom porukom. Promjene u almanahu događaju se na dnevnoj bazi. Ukoliko prijamnik već poznaje dnevni almanah, u stanju je brže napraviti prvu procjenu položaja, što nazivamo topli start GPS (ili općenito GNSS) prijamnika. Ukoliko su prijamniku poznati i dnevni almanah i efemeride, vrijeme do prve procjene položaja je još kraće i iznosi nekoliko desetaka sekundi. Nazivamo ga vrućim startom GNSS prijamnika. Ako prijamnik nema ni dnevni alamanh ni sateliske efemeride, vrijeme do procjene položaja može biti prilično dugo. Ono ovisi o načinu dobavljanja navigacijske poruke. Ako se poruka prima sa satelita, vrijeme do prve procjene položaja je barem 12.5 min.Takvo stanje GNSS prijamnika se naziva hladan start GNSS prijamnika. Hladan start je moguće ubrzati alternativnom dostavom navigacijske poruke, npr. preko telekomunikacijskih mreža. Naime, elementi telekomunikacijskih mreža su vremenski usklađeni pomoću satelitskih navigacijskih prijamnika pa čvorovi mreže već poznaju navigacijsku poruku i mogu je prenijeti korisničkoj opremi (GNSS prijamniku). Način rada u kojem korisnički prijamnik ne prima sve potrebne podatke za određivanje položaja putem satelita nazivamo potpomognutom satelitskom navigacijom (engl. Augmented GNSS, A-GNSS).

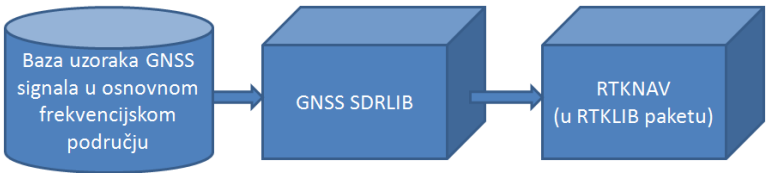
Algoritam procesiranja informacija u domeni navigacijske primjene (vidi: Poglavlje 2) koristi informacije iz navigacijske poruke (satelitske efemeride, alamanah i parametre modela ispravaka pogrešaka) te izmjerene pseudoudaljenosti kako bi se procijenio položaj (i/ili brzinu) prijamnika i ispravio pogrešku korisničkog sata. Prvo se modelima ispravaka ispravljaju poznate sustavne pogreške položaja satelita, ionosferskog i troposferskog kašnjenja te pogreške točnosti [2] korisničkog sata izmjerenih vrijednosti pseudoudaljenosti. Tako se dobivaju ispravljene mjerene pseudoudaljenosti koje se koriste za ulaz algoritma procjene položaja.
Slučajne pogreške ostaju nepokrivene pa procjena položaja nije apsolutno točna. Ipak, postupak procjene položaja omogućuje zadovoljavajuću procjenu pogreške određivanja položaja. Ona se može predstaviti korisniku, zajedno s rezultatima procjene položaja (i/ili brzine i vremena).
Nepokrivenost slučajne pogreške referentnim načinom rada programski određenog radioprijamnika nagnala je mnoge pa i nas na kontrukciju modela redukcije slučajne

[2]Ne misli se na preciznost. Pogreške preciznosti korisničkog sata se modeliraju algoritmom procjene položaja uvođenjem četvrte nepoznanice sustava.

pogreške (vidi: stranica 48). Redukcija slučajne pogreške se temelji na modifikaciji referentnog algoritma procjene položaja za koju postoji mogućnost integracije u izvedeni programski određen radioprijamnik.

3.4 Praktična izvedba korisničkog GPS prijamnika

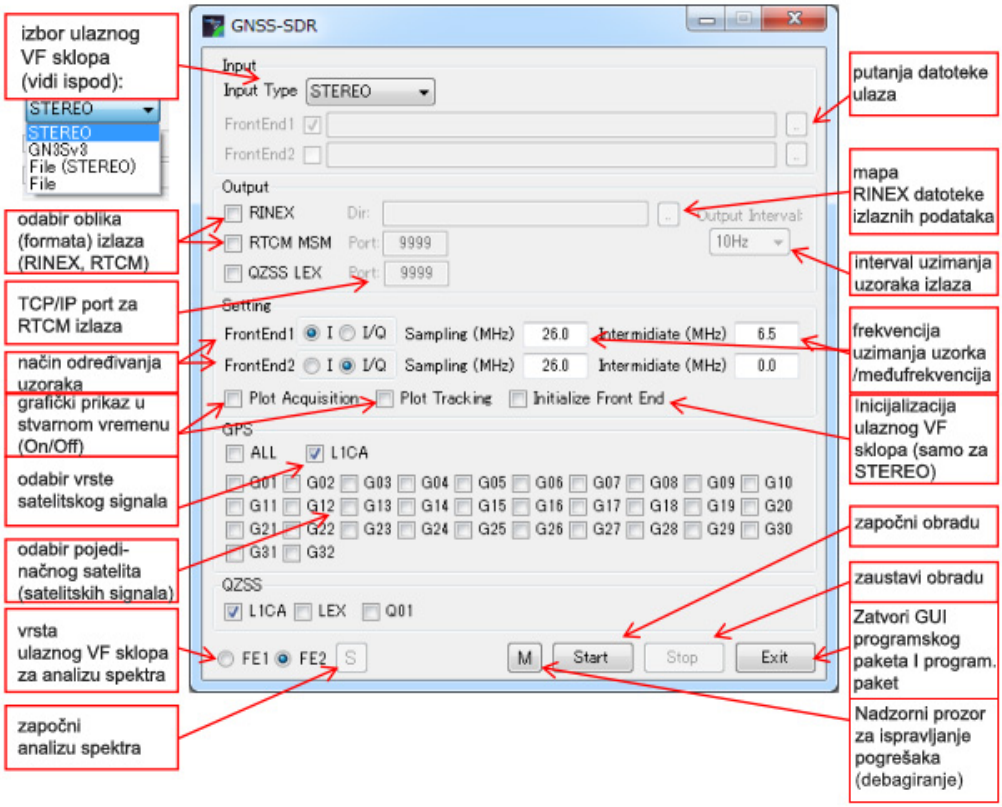
U okviru diplomskog rada izvaden je korisnički *GPS* prijamnik. Izvedeni radioprijamnik je moguće koristiti za obradu satelitskih signala i informacija u domeni osnovnog frekvencijskog područja i domeni navigacijske primjene. Obrada signala u domeni osnovnog frekvencijskog područja se izvodi korištenjem programske knjižnice otvorenog koda GNSS SDRLIB [29]. Obrada informacija u domeni navigacijske primjene je moguće izvesti korištenjem programskog paketa otvorenog koda RTKNAV iz programske knjižnice RTKLIB[30].



Slika 3.3: Shema GNSS radioprijamnika

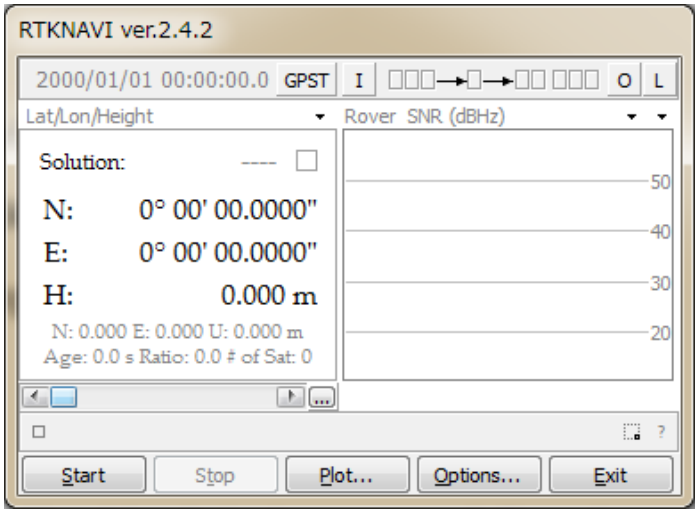
Spomenute knjižnice povezane su klijentsko-poslužiteljskom arhitekturom.

Programska knjižnica GNSS SDRLIB (slika 3.4) omogućava korištenje kompozitnih GPS signala (slika 1.1) u domeni osnovnog frekvencijskog područja dostavljenih strujenjem ili arhivskom datotekom. Omogućuje izbor pojedinačnog GNSS sustava i pojedinačnih satelita pa tako i odabranog *GPS* sustava. Također, omogućava pristup strujanim podatcima potpomognutog GNSS-a, dostavljanim putem internetske veze od trećih strana (dobavljača ispravaka).

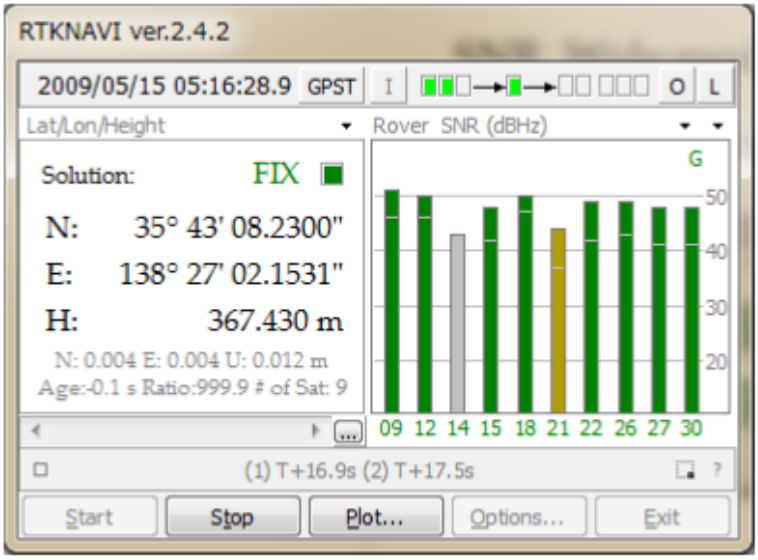


Slika 3.4: Grafičko korisničko sučelje programskog paketa GNSS-SDRLIB

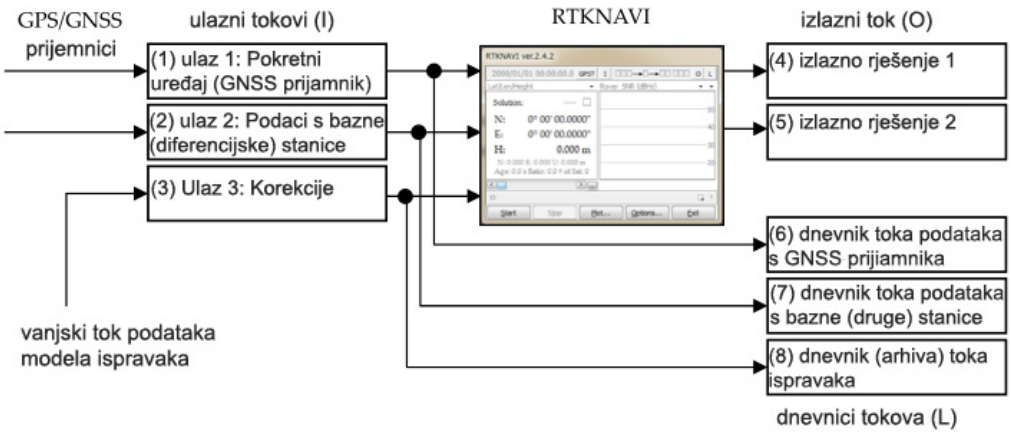
Programski paket RTKNAV je dio programske knjižnice otvorenog koda RTKLIB [30, 8]. Koristi se za procjenu položaja (i/ili brzine i vremena) zasnovanom na podatcima (pseudoudaljenosti i sadržaja navigacijske poruke) koji čine izlaz domene za obradu signala u osnovnom frekvencijskom području. Preko grafičkog korisničkog sučelja (slike 3.5 i 3.6) omogućuje praćenje statusa procesa: toka podataka iz GNSS SDRLIB aplikacije prema RTKNAV aplikaciji, grafičkog predstavljanja jakosti prihvaćenih i slijeđenih satelitskih signala te procjenu navigacijskih parametara (tri komponente položaja: geografska širina, geografska dužina i nadmorska visina, brzina i točno vrijeme) temeljem mjerenih vrijednosti pseudoudaljenosti korištenih satelita te uz korištenje temeljnog postupka procjene položaja i točnog vremena (slika 3.7). Korišteni algoritmi procesiranja informacija i procjene položaja opisani su u dokumentraciji programske knjižnice RTKLIB [30].



Slika 3.5: Grafičko korisničko sučelje programskog paketa RTKNAV



Slika 3.6: GUI RTKNAV aplikacije u radu (zastavica FIX označava ispravnu procjenu položaja)



Slika 3.7: Korištenje aplikacije RTKNAV, s ulaznim i izlaznim informacijama

Korišteni uzorci GPS signala u domeni osnovnog frekvencijskog područja[3] su pribavljeni eksperimantalno u stvarnim uvjetima. Podešavanjem opcija, izvedeni GPS prijamnik obavlja obradu signala u frekvencijskoj domeni. Obavlja se i obrada izmjerenih pseudoudaljenosti referentnim modelima ispravaka. Potrebni podatci za ulaz algoritama procjene položaja u domeni navigacijske primjene su spremljeni u obliku tekstualne datoteke (koordinate satelita i pripadne pseudoudaljenosti). Sadržaj korištenih tekstualnih datoteka dan je u Dodatku D ovoga rada.

[3]Spremljeni u RINEX formatu (RINEX podatci).

Poglavlje 4

Praktična izvedba procjene položaja u domeni navigacijske primjene

Prije korištenja algoritma za procjene položaja, u domeni navigacijske primjene, obavljaju se sljedeće radnje (procesiranje informacija u domeni navigacijske primjene):

1. Prikupljaju se potrebni podatci izlaza domene osnovnog frekvencijskog područja: (I1) djelovi navigacijske poruke (satelitske efemeride, dnevni almanah i parametri modela ispravaka) i (I2) mjerene pseudoudaljenosti,
2. Ispravljaju se mjerene psudo-udaljenosti prikupljenim podatcima modelima ispravaka.

Za potrebe ovoga rada, gornje radnje su obavljene primjereno podešen koristeći izvedeni programski određen radioprijamnik. Izlaz gornjih radnji su dvije datoteke ulaza algoritma procjene položaja: (U1) datoteka ispravljenih pseud-oudaljenosti i (U2) datoteka satelitskih efemerida pripadnih satelita. Primjer njihovog sadržaja je dan u Dodatku D.

Općenito, gornje radnje mogu biti izrazito kompleksne i prelaze obujam ovoga rada. U praksi korišteni algoritmi procjene položaja u domeni navigacijske primjene opisani su dokumentacijom programske knjižnice RTKLIB [30].

Ovo poglavlje najprije opisuje dva načina linearizacije nelinearnog sustava jednadžbi iz Poglavlja 1.5,a zatim numeričke metode koje je moguće iskoristiti u rješavanju lineariziranog sustava jednadžbi iterativnog postupka. Na kraju su opisana dva pristupa izvedbi procjene položaja u domeni navigacijske primjene.

Prvi način linearizacije jednadžbi sustava

Prvi način linearizacije jednadžbi sustava dobivamo promatrajući sustav (1.5) s $v_i = p_i(\mathbf{x})$:

$$\begin{aligned} d_1 &= \sqrt{(x-x_1)^2+(y-y_1)^2+(z-z_1)^2} + c\cdot d_T + p_1(\mathbf{x}) \\ d_2 &= \sqrt{(x-x_2)^2+(y-y_2)^2+(z-z_2)^2} + c\cdot d_T + p_2(\mathbf{x}) \\ d_3 &= \sqrt{(x-x_3)^2+(y-y_3)^2+(z-z_3)^2} + c\cdot d_T + p_3(\mathbf{x}) \\ d_4 &= \sqrt{(x-x_4)^2+(y-y_4)^2+(z-z_4)^2} + c\cdot d_T + p_4(\mathbf{x}) \end{aligned} \tag{4.1}$$

u $\mathbf{x} = (x, y, z, d_T)$.
Sada je:

$$p_i(\mathbf{x}) = \sqrt{(x-x_1)^2+(y-y_1)^2+(z-z_1)^2} + c\cdot d_T - d_i \tag{4.2}$$

Linearizacijom jednadžbi sustava (4.1) na način korišten na stranici 19 (za Newton-Gaussovu metodu), dobivamo:

$$p_i(\mathbf{x}+\Delta\mathbf{x}) = p_i(\mathbf{x}) + \frac{\partial p}{\partial x}\Delta x + \frac{\partial p}{\partial y}\Delta y + \frac{\partial p}{\partial z}\Delta z + \frac{\partial p}{\partial d_T}\Delta d_T \tag{4.3}$$

Koristeći iterativni postupak, svaki korak k definira $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k$ gdje je $\Delta\mathbf{x}_k$ rješenje sustava k-tog koraka. Nastoji se postići $\mathbf{p}(\mathbf{x}_k + \Delta\mathbf{x}_{k+1}) = 0$ što daje

$$p_i(\mathbf{x}_k) + \frac{\partial p_i}{\partial x}\Delta x_k + \frac{\partial p_i}{\partial y}\Delta y_k + \frac{\partial p_i}{\partial z}\Delta z_k + \frac{\partial p_i}{\partial d_T}\Delta(d_T)_k = 0, \tag{4.4}$$

odnosno

$$\begin{aligned} -p_i(\mathbf{x}_k) &= d_i - \sqrt{(x_k-x_i)^2+(y_k-y_i)^2+(z_k-z_i)^2} - c\cdot(d_T)_k \\ &= \frac{\partial f}{\partial x}\Delta x_k + \frac{\partial f}{\partial y}\Delta y_k + \frac{\partial f}{\partial z}\Delta z_k + \frac{\partial f}{\partial d_T}\Delta(d_T)_k \\ &= \begin{bmatrix} \frac{\partial f_i}{\partial x} & \frac{\partial f_i}{\partial y} & \frac{\partial f_i}{\partial z} & \frac{\partial f_i}{\partial d_T} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta d_T \end{bmatrix} \end{aligned} \tag{4.5}$$

Uz

$$\mathbf{A} := \begin{bmatrix} \frac{\partial p_1}{\partial x} & \frac{\partial p_1}{\partial y} & \frac{\partial p_1}{\partial z} & \frac{\partial p_1}{\partial d_T} \\ \frac{\partial p_2}{\partial x} & \frac{\partial p_2}{\partial y} & \frac{\partial p_2}{\partial z} & \frac{\partial p_2}{\partial d_T} \\ \frac{\partial p_3}{\partial x} & \frac{\partial p_3}{\partial y} & \frac{\partial p_3}{\partial z} & \frac{\partial p_3}{\partial d_T} \\ \frac{\partial p_4}{\partial x} & \frac{\partial p_4}{\partial y} & \frac{\partial p_4}{\partial z} & \frac{\partial p_4}{\partial d_T} \end{bmatrix} \tag{4.6}$$

$$= \begin{bmatrix} \frac{(x-x_1)}{\sqrt{(x-x_1)^2+(y-y_1)^2+(z-z_1)^2}} & \frac{(y-y_1)}{\sqrt{(x-x_1)^2+(y-y_1)^2+(z-z_1)^2}} & \frac{(z-z_1)}{\sqrt{(x-x_1)^2+(y-y_1)^2+(z-z_1)^2}} & c \\ \frac{(x-x_2)}{\sqrt{(x-x_2)^2+(y-y_2)^2+(z-z_2)^2}} & \frac{(y-y_2)}{\sqrt{(x-x_2)^2+(y-y_2)^2+(z-z_2)^2}} & \frac{(z-z_2)}{\sqrt{(x-x_2)^2+(y-y_2)^2+(z-z_2)^2}} & c \\ \frac{(x-x_3)}{\sqrt{(x-x_3)^2+(y-y_3)^2+(z-z_3)^2}} & \frac{(y-y_3)}{\sqrt{(x-x_3)^2+(y-y_3)^2+(z-z_3)^2}} & \frac{(z-z_3)}{\sqrt{(x-x_3)^2+(y-y_3)^2+(z-z_3)^2}} & c \\ \frac{(x-x_4)}{\sqrt{(x-x_4)^2+(y-y_4)^2+(z-z_4)^2}} & \frac{(y-y_4)}{\sqrt{(x-x_4)^2+(y-y_4)^2+(z-z_4)^2}} & \frac{(z-z_4)}{\sqrt{(x-x_4)^2+(y-y_4)^2+(z-z_4)^2}} & c \end{bmatrix}$$

$$\mathbf{x} := \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta d_T \end{bmatrix} \tag{4.7}$$

$$\mathbf{b} := \begin{bmatrix} d_1 - \sqrt{(x_k - x_1)^2 + (y_k - y_1)^2 + (z_k - z_1)^2} - c \cdot (d_T)_k \\ d_2 - \sqrt{(x_k - x_2)^2 + (y_k - y_2)^2 + (z_k - z_2)^2} - c \cdot (d_T)_k \\ d_3 - \sqrt{(x_k - x_3)^2 + (y_k - y_3)^2 + (z_k - z_3)^2} - c \cdot (d_T)_k \\ d_4 - \sqrt{(x_k - x_4)^2 + (y_k - y_4)^2 + (z_k - z_4)^2} - c \cdot (d_T)_k \end{bmatrix} \tag{4.8}$$

dobivamo sustav

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{4.9}$$

koji rješavamo metodom iterativnih najmanjih kvadrata.

Drugi način linearizacije jednadžbi sustava

Drugi način linearizacije se dobiva promatrajući isti sustav jednadžbi (1.5). Metoda iterativnih najmanjih kvadrata se ne primjenjuje izvorno na početni sustav, već na njegovu modifikaciju, modifikaciju sustava (2.1). Modificirani sustav, uz uvjet $cd_T < d_i, \forall i \in \{1, 2, 3, 4\}$, je u mogućnosti dati isto dobro rješenje.

Prebacujući član $d = d_T \cdot c$ na lijevu stranu i kvadrirajući obje strane jednadžbi sustava (2.1) dobivamo modificirani sustav jednadžbi:

$$\begin{aligned} (d_1 - d)^2 &= (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 + p_1(\mathbf{x}) \\ (d_2 - d)^2 &= (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 + p_2(\mathbf{x}) \\ (d_3 - d)^2 &= (x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 + p_3(\mathbf{x}) \\ (d_4 - d)^2 &= (x - x_4)^2 + (y - y_4)^2 + (z - z_4)^2 + p_4(\mathbf{x}) \end{aligned} \tag{4.10}$$

gdje je

$$p_i(\mathbf{x}) = 2\sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}v_i + v_i^2. \tag{4.11}$$

i

$$p_i(\mathbf{x}) = (d_i - d)^2 - (x_i - x)^2 - (y_i - y)^2 - (z_i - z)^2. \tag{4.12}$$

Označimo

$$\tilde{\mathbf{p}}(\mathbf{x}) := (p_1(\mathbf{x}), p_2(\mathbf{x}), p_3(\mathbf{x}), p_4(\mathbf{x}))^T \tag{4.13}$$

pa problem minimizacije (2.11) prelazi u

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \tilde{\mathbf{p}}(\mathbf{x})^T \tilde{\mathbf{p}}(\mathbf{x}). \tag{4.14}$$

i

$$\begin{aligned} \tilde{\mathbf{p}}'(\mathbf{x}) &= (p_1'(\mathbf{x}), p_2'(\mathbf{x}), p_3'(\mathbf{x}), p_4'(\mathbf{x}))^T \\ &= \begin{bmatrix} 2(x_1 - x) & 2(y_1 - y) & 2(z_1 - z) & -2c(d_1 - cd_T) \\ 2(x_2 - x) & 2(y_2 - y) & 2(z_2 - z) & -2c(d_2 - cd_T) \\ 2(x_3 - x) & 2(y_3 - y) & 2(z_3 - z) & -2c(d_3 - cd_T) \\ 2(x_4 - x) & 2(y_4 - y) & 2(z_4 - z) & -2c(d_4 - cd_T) \end{bmatrix} \end{aligned} \tag{4.15}$$

Neka je

$$\mathbf{P} := \begin{bmatrix} (x_1 - x) & (y_1 - y) & (z_1 - z) & (d_1 - d) \\ (x_2 - x) & (y_2 - y) & (z_2 - z) & (d_2 - d) \\ (x_3 - x) & (y_3 - y) & (z_3 - z) & (d_3 - d) \\ (x_4 - x) & (y_4 - y) & (z_4 - z) & (d_4 - d) \end{bmatrix} \tag{4.16}$$

$$\tilde{\mathbf{I}} := diag(1, 1, 1, -c) \tag{4.17}$$

$$\tilde{\mathbf{P}'} := 2\mathbf{P}\tilde{\mathbf{I}}. \tag{4.18}$$

Imamo

$$\Delta\mathbf{x_k} = -(2\mathbf{P}\tilde{I})^{-1}(\tilde{\mathbf{p}}(\mathbf{x_k})) \tag{4.19}$$

$$= -\frac{1}{2}(\mathbf{P}\tilde{\mathbf{I}})^{-1}(\tilde{\mathbf{p}}(\mathbf{x_k})). \tag{4.20}$$

Uz oznake

$$\mathbf{A} := \tilde{\mathbf{P}'} \tag{4.21}$$

$$\mathbf{b} := -\tilde{\mathbf{p}}(\mathbf{x_k}) \tag{4.22}$$

svaka iteracija Algoritma 2.1. sa stranice 20, računa $\Delta\mathbf{x_k}$ rješavajući sustav

$$\mathbf{Ax} = \mathbf{b} \tag{4.23}$$

u $\mathbf{x} := \Delta\mathbf{x_k}$.

Analogno, (2.17) prelazi u

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \tilde{\mathbf{p}}(\mathbf{x})^T \Sigma^{-1} \tilde{\mathbf{p}}(\mathbf{x}). \tag{4.24}$$

i uz

$$\begin{aligned} \mathbf{A} &:= \Sigma^{-\frac{1}{2}} \tilde{\mathbf{P}}' \\ \mathbf{b} &:= -\Sigma^{-\frac{1}{2}} \tilde{\mathbf{p}}(\mathbf{x_k}) \end{aligned} \tag{4.25}$$

svaka iteracija Algoritma 2.2 rješava sustav (4.23).

4.1 Numerička linearna algebra za rješavanje dobivenog sustava problema najmanjih kvadrata

Pogreške u mjerenjima ili linearizacija uvjetuju da sustav (4.9) nema uvijek rješenje, tj. $\mathbf{Ax} - \mathbf{b} \neq 0, \forall \mathbf{x} \in \mathbb{R}^m$. Zato ideja rješavanja sustava problema najmanjih kvadrata nije tražiti rješenje sustava, već $\mathbf{x}$ koji minimizira izraz $\|\mathbf{Ax} - \mathbf{b}\|_2$.

Neki od najbitnijih koncepta za pronalazak rješenja sustava u smislu najmanjih kvadrata jesu [27]:

- sustav normalnih jednadžbi,
- QR dekompozicija,
- SVD,

a objašnjeni su u nastavku.

Sustav normalnih jednadžbi

Sustavi normalnih jednadžbi su značajni za rješavanje problema najmanjih kvadrata jer se njihovim rješavanjem dobiva rješenje jednako rješenju pripadnog problema najmanjih kvadrata.
Dakle, za pronalazak rješenja sustava:

$$\mathbf{Ax} = \mathbf{b} \tag{4.26}$$

u smislu najmanjih kvadrata je dovoljno promatrati pripadni sustav (normalnih jednadžbi):

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T b. \tag{4.27}$$

Dokaz je dan teoremom 4.1.1.

Teorem 4.1.1. *Skup svih rješenja problema* $\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2$ *označimo s*

$$S = \{\mathbf{x} \in \mathbb{R}^m | \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \text{ je minimalna}\}$$

Tada je $\mathbf{x} \in S$*, tj.* $\mathbf{x}$ *je rješenje problema najmanjih kvadrata, ako i samo ako vrijedi sljedeća relacija ortogonalnosti*

$$\mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) = 0,$$

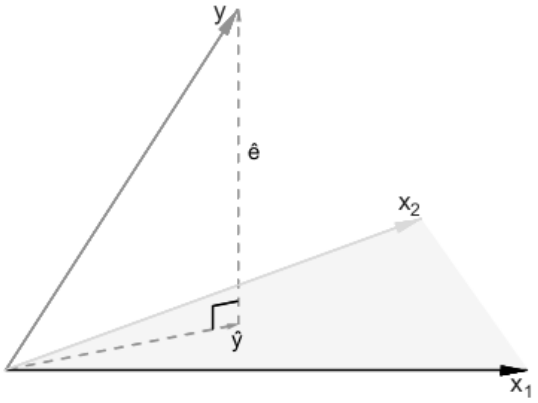
koju obično nazivamo sustav normalnih jednadžbi i pišemo u obliku

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T b$$

.

Dokaz. Rješavanje problema $\mathbf{A}\mathbf{x} = \mathbf{b}$, gdje $\mathbf{x}$ parametar koji je potrebno odrediti, se svodi na prikaz vektora $\mathbf{b}$ u bazi koju čine stupci matrice $\mathbf{A}$. Ukoliko $\mathbf{b}$ nije iz prostora razapetog stupcima matrice $\mathbf{A}$, L_A, tada je potrebno pronaći vektor $\hat{\mathbf{b}} \in L_A$ i najbliži vektoru $\mathbf{b}$ među svim vektorima iz L_A. Po definiciji, $\hat{\mathbf{b}}$ je projekcija $\mathbf{b}$ na L_A dana formulom:

$$\hat{\mathbf{b}} := \mathbf{A}(\mathbf{A}^\mathbf{T}\mathbf{A})^{-\mathbf{1}}\mathbf{A}^\mathbf{T}\mathbf{b}$$



Slika 4.1: Projekcija vektora $\mathbf{y}$ u prostor razapet vektorima x_1 i x_2 (izvor: [23])

Izraz $\mathbf{A}(\mathbf{A^T A})^{-1}\mathbf{A^T}$ nazivamo projektor na prostor razapet stupcima matrice $\mathbf{A}$ i obično se označava s $\mathbf{H}$.
Vrijedi da je $\hat{\mathbf{x}}$ rješenje problema najmanjih kvadrata ako i samo ako vrijedi $\mathbf{A}\hat{\mathbf{x}} = \hat{\mathbf{b}}$, tj. $\hat{\mathbf{x}} := (\mathbf{A^T A})^{-1}\mathbf{A^T b}$. Zaključujemo kako je $\hat{\mathbf{x}}$ rješenje problema najmanjih kvadrata ako i samo ako je $\hat{\mathbf{x}}$ rješenje problema normalnih jednadžbi

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}. \tag{4.28}$$

□

Detaljniji dokaz teorema se može pronaći u [28, stranica 46].
Napomenimo da spomenuti sustav normalnih jednadžbi ima i sljedeća svojstva:

1. Općenito, matrica $\mathbf{A}^T\mathbf{A}$ je simetrična i pozitivno semidefinitna jer za svaki $\mathbf{x} \in \mathbb{R}^m$ vrijedi

$$\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} = (\mathbf{A}\mathbf{x})^T(\mathbf{A}\mathbf{x}) = \|A\mathbf{x}\|_2^2 \geq 0. \tag{4.29}$$

2. Sustav normalnih jednadžbi uvijek ima rješenje i to jedinstveno.

Ona olakšavaju njegovo rješavanje.

Nakon konačne formalizacije problema, potrebno je izabrati način izračunavanja rješenja sustava (4.28).
U praksi se matrica $\mathbf{A}^T\mathbf{A}$ ni ne formulira, nego se rješava početni sustav. Naime, postoji velika mogućnost da će $\mathbf{A}^T\mathbf{A}$ biti loše uvjetovana.
Ipak, u daljnjim poglavljima izvodimo i metodu pronalaska rješanja sustava koja se zasniva na formuliranju sustava (4.28) koji se rješava koristeći dekompoziciju Choleskoga matrice $\mathbf{A}^T\mathbf{A}$. Spomenuta metoda se često spominje prilikom rješavanja problema najmanjh kvadrata određivanja položaja GNSS navigacijskim signalima.
Zbog moguće numeričke nestabilnosti, tako pronađeno rješenje obično nije zadovoljavajuće točnosti [28, stranica 60] i metodu ne preporučujemo za korištenje.

QR dekompozicija

Pristup rješavanja problema najmanjih kvadrata QR dekompozicijom zasniva se na činjenici da je matrica sustava punog stupčanog ranga. Rješenje opisuje na sljedeći način

$$\begin{aligned}\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 &= \min_{\mathbf{x}} \|\mathbf{Q}^T(\mathbf{A}\mathbf{x} - \mathbf{b})\|_2 \\ &= \min_{\mathbf{x}} \|\mathbf{Q}^T\mathbf{A}\mathbf{x} - \mathbf{Q}^T\mathbf{b}\|_2\end{aligned} \tag{4.30}$$

gdje je $\mathbf{Q}$ proizvoljna ortogonalna matrica.
Zbog proizvoljnosti matrice $\mathbf{Q}$, ona može bit odabrana tako da pojednostavljuje izračun za $\mathbf{x}$. Ukoliko se koristi $\mathbf{Q}$ iz QR dekompozicije matrice $\mathbf{A}$ imamo $\mathbf{A} = \mathbf{QR}$, $\mathbf{Q}^T\mathbf{A} = \mathbf{R}$ i $\mathbf{R}$ je gornjetrokutasta matrica. Dalje se rješava sustav

$$\mathbf{Rx} = \mathbf{Q}^T\mathbf{b}. \tag{4.31}$$

tj. traži se

$$\begin{aligned}
\mathbf{x} &= \mathbf{R}^{-1}\mathbf{Q}^T\mathbf{b} \\
&= \mathbf{R}^{-1}\mathbf{R}^{-T}\mathbf{R}^T\mathbf{Q}^T\mathbf{b} \\
&= (\mathbf{R}^T\mathbf{R})^{-1}(\mathbf{QR})^T\mathbf{b} \\
&= (\mathbf{R}^T\mathbf{IR})^{-1}(\mathbf{QR})^T\mathbf{b} \\
&= (\mathbf{R}^T\mathbf{Q}^T\mathbf{QR})^{-1}(\mathbf{A})^T\mathbf{b} \\
&= (\mathbf{A}^T\mathbf{A})^{-1}(\mathbf{A})^T\mathbf{b}.
\end{aligned} \tag{4.32}$$

Gledajući gornju jednakost odozdo prema gore opažamo kako je rješenje sustava normalnih jednadžbi jednako rješenju trokutastog sustava početnog sustava jednadžbi (4.31).
Dakle, dobiveno rješenje također je jednako rješenju problema najmanjih kvadrata dobivenog sustava jednadžbi.

Rješenje sustava (4.31) se također pronalazi kao rješenje pripadnog problema najmanjih kvadrata. Kako je matrica sustava gornjetrokutasta, novi sustav je znatno jednostavnije riješiti. Rješenje dobiveno koristeći QR dekompoziciju je stabilnije i ostvaruje manje odstupanje nego rješenje dobiveno formulacijom sustava normalnih jednadžbi.

Općenito, QR dekompoziciju možemo koristi i prilikom direktnog računa inverza kvadratne matrice punog stupčanog ranga $\mathbf{A}$:

$$\mathbf{A}^{-1} = \mathbf{R}^{-1}\mathbf{Q}^{-1} = \mathbf{R}^{-1}\mathbf{Q}^{\mathbf{T}}. \tag{4.33}$$

Matrice sustava problema najmanjih kvadrata obično nisu kvadratne, ali imaju puni stupčani ili retčani rang. Za takve matrice, moguće je računati pseudoinverz. Pseudoinvez matrice $\mathbf{A}$ je matrica $\mathbf{A}^\dagger$ takva da vrijedi

$$\mathbf{x} = \mathbf{A}^\dagger\mathbf{Ax} \tag{4.34}$$

za sve $\mathbf{x} \in \mathbb{R}^m$. Ukoliko je matrica $\mathbf{A} \in \mathbb{R}^{n\times m}$, $n \geq m$ i ima puni stupčani rang, QR dekompozicija rezultira ortogonalnom matricom $\mathbf{Q}$ i nekvadratnom matricom $\mathbf{R}$ koja

se može zapisati u obliku:

$$\mathbf{R} = \begin{bmatrix} \mathbf{R_1} \\ \mathbf{0} \end{bmatrix} \tag{4.35}$$

gdje je $\mathbf{R_1}$ kvadratna gornjetrokutasta matrica.
Pseudoinverz matrice $\mathbf{A}$ je sada dan sljedećom formulom:

$$\mathbf{A}^\dagger = \begin{bmatrix} \mathbf{R_1}^{-1} & \mathbf{0}^T \end{bmatrix} \mathbf{Q^T} \tag{4.36}$$

Ukoliko je matrica $\mathbf{A} \in \mathbb{R}^{n \times m}$, $n \leq m$ i ima puni retčani rang, QR dekompozicijom dobivamo ortogonalnu matricu $\mathbf{Q}$ i $\mathbf{R}$ koja opet nije kvadratna, ali se može zapisati u obliku

$$\mathbf{R} = \begin{bmatrix} \mathbf{R_1} & \mathbf{0} \end{bmatrix} \tag{4.37}$$

Pseudoinverz za takve matrice se može dobiti po sljedećoj formuli:

$$\mathbf{A}^\dagger = \begin{bmatrix} \mathbf{R_1}^{-1} \\ \mathbf{0}^T \end{bmatrix} \mathbf{Q^T} \tag{4.38}$$

Posebno svojstvo matrice $\mathbf{R}$ rezultira jednostavnijem i stabilnijem izračunavanjem (pseudo)inverza matrice $\mathbf{A}$.

SVD dekompozicija

Definicija 4.1.2 (SVD dekompozicija matrice)**.** *Neka je $\mathbf{A} \in \mathbb{R}^{m \times n}$ ili $\mathbb{C}^{m \times n}$, SVD dekompozicija (engl. Singular Value Decompozition) matrice je $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^*$, $\mathbf{U} \in \mathbb{R}^{m \times m}$ ili $\mathbb{C}^{m \times m}$ unitarna matrica, $\mathbf{V} \in \mathbb{R}^{n \times n}$ ili $\mathbb{C}^{n \times n}$ unitarna matrica i $\mathbf{D} \in \mathbb{R}^{m \times n}$ nenegativna dijagonalna matrica.*
Nadalje, stupci matrice $\mathbf{U}$ su svojstveni vektori matrice $\mathbf{A}\mathbf{A}^*$, dok su stupci matrice $\mathbf{V}$ svojstveni vektori matrice $\mathbf{A}^*\mathbf{A}$. Dijagonalni elementi matrice $\mathbf{D}$ su korijeni svojstvenih vrijednosti matrice $\mathbf{A}^*\mathbf{A}$ ili $\mathbf{A}\mathbf{A}^*$.

Prva primjena SVD dekompozicije je u računu inverza kvadratne matrice punog ranga $\mathbf{A}$:

$$\mathbf{A}^{-1} := \mathbf{V}^{-*}\mathbf{D}^{-1}\mathbf{U}^{-1} := \mathbf{V}\mathbf{D}^{-1}\mathbf{U}^* \tag{4.39}$$

Kako su za izračun inverza sada potrebni samo inverz dijagonalne matrice i hermitski adjugirana matrica proizvoljne matrice, SVD dekompozicija znatno olakšava izračun inverza.

Nadalje, SVD dekompozicija se koristi i prilikom rješavanja sustava linearnih jednadžbi. Kako su u sklopu ovoga rada definirani samo sustavi realnih matrica sustava, može se pretpostaviti kako za $\mathbf{U}$, $\mathbf{D}$ i $\mathbf{V}$ matrice SVD dekompozicije matrice $\mathbf{A}$ vrijedi $\mathbf{U}^* = \mathbf{U}^T$, $\mathbf{D}^* = \mathbf{D}^T = \mathbf{D}$, $\mathbf{V}^* = \mathbf{V}^T$ i $\mathbf{U}$ i $\mathbf{V}$ su ortogonalne matrice. Sada se za $\mathbf{Q}$ iz (4.30) može uzeti da je jednaka matrici $\mathbf{U}$.

Dobiva se sljedeće:

$$\begin{aligned} \|\mathbf{Ax} - \mathbf{b}\|_2 &= \|\mathbf{U}^T(\mathbf{Ax} - \mathbf{b})\|_2 \\ &= \|\mathbf{U}^T\mathbf{Ax} - \mathbf{U}^T\mathbf{b}\|_2 \\ &= \|\mathbf{DV}^T\mathbf{x} - \mathbf{U}^T\mathbf{b}\|_2 \end{aligned} \tag{4.40}$$

Dalje se onda rješava sustav nominalnih jednadžbi

$$(\mathbf{DV}^T)^T\mathbf{DV}^T\mathbf{x} = (\mathbf{DV}^T)^T\mathbf{U}^T\mathbf{b} \tag{4.41}$$

i rješenje za $\mathbf{x}$ je jednako

$$\mathbf{x} = \mathbf{VD}^{-1}\mathbf{U}^T\mathbf{b}. \tag{4.42}$$

Primjer rješavanja problema opisanog sustavom $\mathbf{Ax} = \mathbf{b}$ koristeći SVD dekompoziciju u programskom okruženju *R* dano je izvorom [5].

Problemi najmanjih kvadrata definirani ovim radom opisani su zasićenim ili prezasićenim sustavima jednadžbi. Za njihovo se rješavanja potiče korištenje QR dekompozicije [23].
Pristup rješavanja problema najmanjih kvadrata koji koristi sustav normalnih jednadžbi se smatra najbržim, ali najmanje stabilnim. Pristup nije upotrebljiv[1] već ukoliko je uvjetovanost matrice početnog sustava blizu $\frac{1}{\sqrt{\text{eps}}}$, gdje eps označava strojnu točnost računala na kojem se dekompozicija izvodi. Pristup koji koristi QR dekompoziciju povećava broj korištenih operacija pa i vrijeme rješavanja problema, ali je stabilniji. Nije ga moguće koristiti tek kada je uvjetovanost matrice blizu $\frac{1}{\text{eps}}$. Ono postiže najbolju relativnu pogrešku u smislu najmanjeg kvadratnog odstupanja [20]. SVD pristup je najstabilniji, ali najsporiji. Za razliku od ostalih, jednakom težinom pronalazi i rješenje nezasićenog sustava jednadžbi.

Zanimljivo je da se uz poznato jedno rješenje sustava (4.28), relativno lagano pronalaze i sva ostala.
Naime, uz $\mathbf{Ax} - \mathbf{b} = \mathbf{r}$ i proizvoljan $\hat{\mathbf{x}} \in \mathbb{R}^m$ za koji vrijedi

$$\begin{aligned} \hat{\mathbf{r}} &= \mathbf{A}\hat{\mathbf{x}} - \mathbf{b} \\ &= \mathbf{A}\hat{\mathbf{x}} + \mathbf{r} - \mathbf{Ax} \\ &= \mathbf{r} + \mathbf{A}(\hat{\mathbf{x}} - \mathbf{x}) \end{aligned} \tag{4.43}$$

imamo da je $\hat{\mathbf{x}} \in S$ ako i samo ako $\hat{\mathbf{r}} = \mathbf{r}$, tj. $\mathbf{A}(\hat{\mathbf{x}} - \mathbf{x}) = 0$ i $\hat{\mathbf{x}} - \mathbf{x} \in \mathcal{N}(\mathbf{A})$.
Također, ukoliko je ispunjen jedan od sljedećih uvjeta:

- $\mathbf{A}$ ima puni stupčani rang,
- stupci matrice $\mathbf{A}$ su linearno nezavisni,
- $\mathbf{A}^T\mathbf{A}$ je pozitivno definitna,

$\mathcal{N}(\mathbf{A})$ je trivijalan i rješenje sustava je jedinstveno.

4.2 Izvedbe

Opisuju se dva pristupa izvedbi procesa procjene položaja. Prvi pristup je osnovni pristup i obuhvaća izvedbe osnovnog (referentnog) algoritma. Drugi pristup je poboljšani pristup i predstavlja predložak poboljšanja osnovnog pristupa poboljšanjem osnovnog algoritma. Osnovni algoritam predstavlja algoritam najmanjih kvadrata (Algoritam 2.1., stranica 20). Poboljšanje osnovnog algoritma ostvaruje se uvođenjem težina, odnosno upotrebom algoritma 2.2. sa stranice 21. Oba algoritma za dobivene pseudoudaljenosti i položaj satelita u ECEF XYZ koordinatnom sustavu izračunavaju

[1] Dekompoziciju Choleskoga nije moguće izračunati.

položaj prijamnika i pogrešku sata prijamnika.

Izvedba algoritama je ostvarena korištenjem programskog jezika *R* [16] i R-sučelja *RStudio* na GNU/Linux operativnom sustavu. Općenito, *R* je programski jezik za statističku i drugu matematičku obradbu pomoću računala i ima snažnu grafičku potporu. Između ostalog, podržava postupke zasnovane na linearnoj algebri, analizi i prognozi ponašanja vremenskih nizova [21]. Pogodan je za izvedbu statističke analize, modeliranje i simulacije.

R je dostupan za većinu korištenih platformi (Microsoft Windows, Linux, Mac OS X), a instalacija je poprilično jednostavna. Potrebno je samo preuzeti potrebne datoteke s web-stranice [17] i u skladu s njima instalirati program. Instalirani program nudi grafičko-korisničko R-sučelje (R-GUI) u kojemu se preko naredbene linije zadaju naredbe i pokreću skripte, a dobivaju numerički i grafički rezultati. Postoji i više neslužbenih R-sučelja. Jedan od poznatijih je RStudio [25]. Ovdje se komunikacija opet ostvaruje preko naredbi u konzoli, ali je RStudio opremljen znantno bogatijom grafičkom okolinom (radni prostor, povijest, instalacija paketa, pomoć i sl.). Postoji i mogućnost integracije *R* interpretera u odabrani tekst-editor ili poziva R-funkcija iz drugih programskih jezika (Python, Ruby, SAGE,C, Java).

U izradi ovoga rada uz standardne *R* programske knjižnice, korištene su i dodatne:*MASS*, *matlib*, *limSolve* i *matrixcalc* [9, 10, 21].

Osnovni pristup

Osnovni pristup se temelji na metodi najmanjih kvadrata. Kako je problem određivanja položaja nelinearan, potrebno ga je potrebno prvo **linearizirati**, a tek nekon primijeniti metodu najmanjih kvadrata [19]. Linearizacija se izvodi jednim od predloženih načina.

Izvedba

Izrađene su dvije izvedbe osnovnog algoritma (Algoritam 2.1., stranica 20). Svaka izvedba rješava ne previše drugačiji sustav jednadžbi (2.1) iterativnom metodom najmanih kvadrata. Koristi se QR dekompozicija.
Prva izvedba rješava sustav jednadžbi izvorni sustav (2.1) QR dekompozicijom, tj. sustav (4.31). Druga izvedba rješava sustav (4.23) definiran s (4.21) i (4.22).

Programski kod izvedbi se nalazi u Dodatku C.
Programska izvedba postupka procjene položaja satelitskim navigacijskim sustavom napravljena je u poopćenom obliku. Podržava se slučaj procjene položaja s brojem

izmjerenih pseudoudaljenosti većih ili jednakih 4. Jedna izvedba reprezentirana je jednom *R* skriptom.
Prije pokretanja *R* skripte potrebno je zadovoljiti **uvjete**:

1. Mapa u kojoj se skripta nalazi postavljena za *radnu mapu*,
2. U istoj mapi se nalaze dvije tekstualne datoteke (datoteke ulaza).

Jedna tekstualna datoteka (*pseudorangesb.txt*) treba sadržavati (ispravljene) mjerene pseudoudaljenosti, a druga WGS84 koordinate pripadnih satelita u trenutku mjerenja pseudoudaljenosti (*satellites.txt*). Datoteka s mjerenim pseudoudaljenostima sadrži samo jedan stupac podataka. Broj redaka implicitno određuje broj korištenih satelita u danjem postupku procjene položaja satelitskim navigacijskim sustavom.
Datoteka s koordinatama pripadnih satelita mora sadržavati isti broj redaka. Redak i odgovara WGS84 koordinatama satelita čija se mjerena pseudoudaljenost nalazi u i-tom retku datoteka s mjerenim pseudoudaljenostima. Koordinate su odvojene zarezom. Primjeri datoteka se nalaze u Dodatku D.

Poboljšan pristup: Težinska metoda najmanih kvadrata (WLSM)

Poboljšan pristup ima svrhu smanjenja odstupanja procesa određivanja položaja. Parametar koji može pridonijeti smanjenju odstupanja je utjecaj ionosfere na pogrešku mjerenih pseudoudaljenosti.
Iako pretpostavljamo da su pseudoudaljenosti u potpunosti ispravljene, utjecaj ionosfere nije moguće u potpunosti otkloniti koristeći signal samo jedne frekvencije vala nosioca. Slučajna komponenta se najteže otklanja.
Utjecaj ionosfere gotovo u potpunosti otklanja korištenje dva signala različitih frekvencija koja šalje isti satelit.

Literatura [7], [11], [12], [31] i [26] navodi kako je ionosfera jedna od najznačajnijih uzroka pogreške procjene položaja. Dulji put signala kroz ionosferu povećava negativni utjecaj iononosfere na točnost određivanja položaja prijamnika. Dakle, postoje manje i više točne jednadžbe sustava. Problem najmanih kvadrata (Algoritam 2.1.) može prijeći u problem težinskih najmanih kvadrata (Algoritam 2.2.). Svakoj jednadžbi sustava se pridjeljuje težinski koeficijent k_i proporcionalan njezinoj točnosti, odnosno obrnuto proporcionalan duljini putovanja signala odgovarajućeg satelita kroz ionosferu.

Sa svrhom definicije pripadnog problema težinskih najmanjih kvadrata linearizi-ramo sustav (4.1) na način sa stranice 37, prvim načinom linearizacije, i dobivamo

da $\forall i \in \{1, 2, 3, 4, ...N\}$

$$d_i - \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2 + (z_k - z_i)^2} - c \cdot (d_T)_k = \begin{bmatrix} \frac{\partial p_i}{\partial x} & \frac{\partial p_i}{\partial y} & \frac{\partial p_i}{\partial z} & \frac{\partial p_i}{\partial d_T} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta d_T \end{bmatrix} \tag{4.44}$$

gdje je i indeks pridružen vrijednostima i-te jednadžbe sustava.

Parametar N može biti proizvoljan sve dok se iz nastalog sustava jednadžbi može definirati najmanje onoliko nezavisnih jednadžbi koliko sustav ima nepoznanica.
Do sada se promatrao sustav $N = 4 =$ *broj nepoznanica sustava* s međusobno nezavisnim jednadžbama pa nastavimo na isti način.
Uz definicije sa stranice 37, dobiva se sustav.

$$\mathbf{Ax} = \mathbf{b} \tag{4.45}$$

u koji se uvode težine matričnim množenjem slijeva obe strane gornjeg izraza sa $W^{\frac{1}{2}}$.

Praksa nerijetko koristi samo navigacijske satelitske signale jedne frekvencije i utjecaj ionosfere modelira u skladu s time.

Izvedba

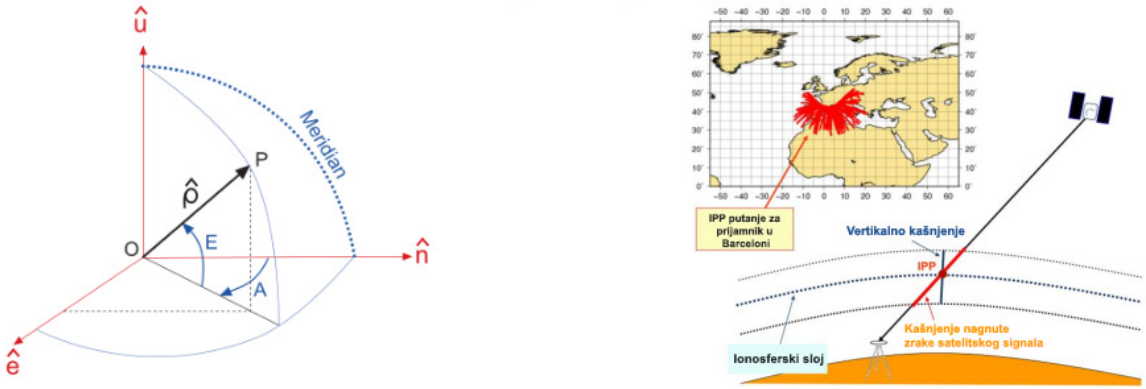
Za potrebe opisa izvedbi najprije uvedimo definiciju kuta elevacije.

Definicija 4.2.1 (Kut elevacije GPS satelita)**.** *Kut elevacije satelita koordinata* (x_i, y_i, z_i) *i prijamnika koordinata* (x_k, y_k, z_k) *se definira kao manji kut između vektora od satelita do prijamnika i vektora od satelita do središta referentnog WGS84 koordinatnog sustava. Egzaktno, kut elevacije* i*-te jednadžbe,* Ele_i*, je definiran sljedećim izrazima:*

$$\cos(l_i) = \frac{-((x_i, y_i, z_i) - (x_k, y_k, z_k))^T}{\|(x_i, y_i, z_i) - (x_k, y_k, z_k)\|} \cdot \frac{-(x_i, y_i, z_i)}{\|(x_i, y_i, z_i)\|} \tag{4.46}$$

$$Ele_i = \left(\frac{\pi}{2} - l_i\right)(l_i \geq 0) + \left(\pi - \left(\frac{\pi}{2} - l_i\right)\right)(l_i < 0) \tag{4.47}$$

Sve koordinate su izražene u WGS84 koordinatnom sustavu.



Slika 4.2: Kut elevacije (izvor)

Uobičajeni način modeliranja težinskih koeficijenata jednadžbi sustava koristi varijancu σ_i^2. Ona modelira varijancu slučajnih varijabli mjerenih podataka i-te jednadžbe. Uz pretpostavku neovisnosti mjerenja, svakoj jednadžbi pridružujemo točno jedan težinski koeficijent definiran izrazom:

$$k_i = \frac{1}{\sigma_i^2} \tag{4.48}$$

i $W = diag(k_1, k_2, \ldots, k_N)^{-1}$.

S ciljem uvođenja ovisnosti težinskih koeficijenata o utjecaju ionosfere, težine modeliramo pomoću varijanci definiranih preko kuta elevacije.
Uz konzultaciju s literaturom [18, 26, 7, 4, 23, 24] varijancu i-te jednadžbe modeliramo pomoću elevacijskog kuta satelita. Dobiveni sustav rješavamo korištenjem QR dekompozicije 4.1 ili sustava normalnih jednadžbi.

Izvodimo tri različita modela težinskih koeficijenata. Težinski koeficijent i-te jednadžbe modeliran j-tim modelom označavamo sa k_{ij}.

$$k_{i1} = \frac{1}{\sigma_{i1}^2} \tag{4.49}$$

$$\sigma_{i1}^2 = \frac{1}{\sin(Ele_i)} \tag{4.50}$$

$$k_{i2} = \frac{1}{\sigma_{i2}^2} \tag{4.51}$$

$$\sigma_{i2}^2 = 1 + \frac{2}{\sin(Ele_i)} \tag{4.52}$$

$$k_{i3} = \frac{1}{\sigma_{i3}^2} \tag{4.53}$$

$$\sigma_{i3}^2 = \frac{1}{(\sin(Ele_i) + 0.5)^2} \tag{4.54}$$

Veći elevacijski kut implicira veći put satelitskog signala kroz ionosferu i pogreška uzrokovana ionosferom je veća (slika 4.2). Kako elevacijski kut poprima vrijednosti između 0 i $\frac{pi}{2}$, veći elevacijski implicira većom vrijednošću funkcije $\sin^2$ i varijance, a manjom vrijednošću težinskog koeficijenta. Jednadžbe većeg elevacijskog kuta se smatraju manje točnima čemu i težimo.

Sadržaj programskog isječka izvedbe moguće je pronaći u sklopu Dodatka C. Podešavanjem parametra $option \in \{1, 2, 3\}$ odabire se model težinskih koeficijenata. Podešavanjem parametra $solution \in \{QR, Snj\}$ odabire se model rješavanja sustava jednadžbi, korištenjem sustava normalnih jednadžbi ili QR dekompozicijom. Ukoliko se odabere $solution = Snj$, sustav se rješava preko sustava normalnih jednadžbi dekompozicijom Choleskoga. Iako se rješavanje preko sustava normalnih jednadžbi ne smatra dobrim kao izravno rješavanje matričnog problema najmanjih kvadrata, često se koristi u postupcima određivanja položaja GNSS navigacijskim signalima. Odabirom $solution = QR$, QR dekompozicije, izbjegava se formulacija pripadnog sustava normalnih jednadžbi.

Prije pokretanja programskog isječaka potrebno je zadovoljiti uvjete sa stranice 48.

Poglavlje 5

Ocjena kvalitete i zaključci

5.1 Osnovni pristup

Kvaliteta procjene položaja satelitskim navigacijskim sustavom razmotrena je sa sljedećih stajališta:

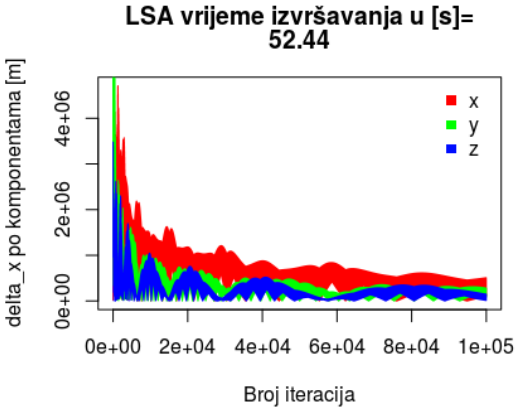
- Potrebnog računalnog vremena za postizanje stabilne procjene u okruženju R,
- Brzine konvergencije iteracijskog postupka u broju iteracija do konvergencije,
- Točnosti procjene koordinata položaja.

Najbitnijim parametrom sa stajališta navigacije se smatra točnost procjene koordinata položja.

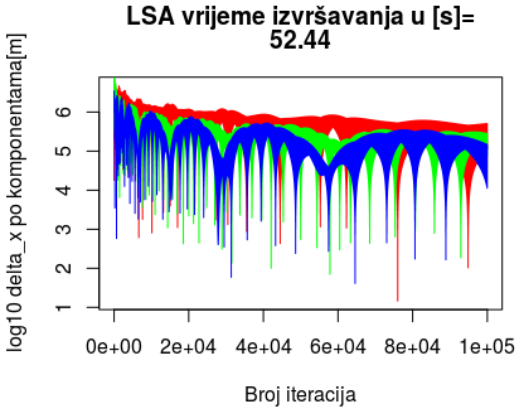
Prilikom procjene točnosti, pokreću se različite izvedbe istih datoteka ulaza sadržaja kako je prikazano u Dodatku C. Prvo dajemo ocjenu kvalitete prve izvedbe osnovnog algoritma predočenu tablicom 5.1 i slikama 5.1 i 5.2.

Tablica 5.1: Kvaliteta prve izvedbe osnovnog algoritma za procjenu položaja u domeni navigacijske primjene

NAZIV PARAMETRA	VRIJEDNOST
vrijeme izvršavanja [s]	52.44 za 10^5, 48h za 10^10 iteracija
broj iteracija do konvergencije	$\approx 10^10$
točnost procjene [m]	10^3



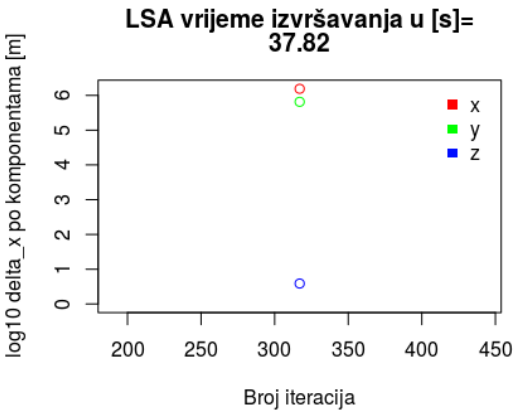
Slika 5.1: Vrijednosti $\Delta\mathbf{x}$ po komponentama kroz iteracije



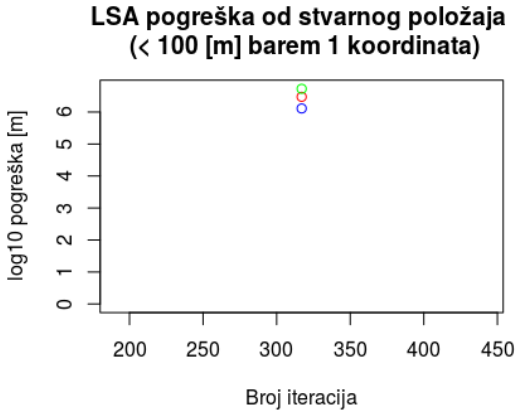
Slika 5.2: Logaritam vrijednosti $\Delta\mathbf{x}$ po komponentama kroz iteracije

Na slikama 5.1 i 5.2 je vidljiv trend smanjivanja izračunate pogreška ($\Delta\mathbf{x}$) procjene položaja. U prvih 10^5 iteracija $\Delta\mathbf{x}$ nikada nije manja od 10^4 m za sve tri komponente istovremeno. Još više, $\Delta\mathbf{x}$ je uvijek veće od 10^4 m u barem dvije komponente, što objašnjava izrazito sporu konvergenciju izvedbe.
Smanjujući zahtjeve konvergencije algoritma toliko da proglašavamo konvergenciju

ako je barem jedna komponenta $\Delta\mathbf{x}$ manja od 10^2 ili 10^3 m, metoda konvergira u 314, odnosno 195 iteracija.

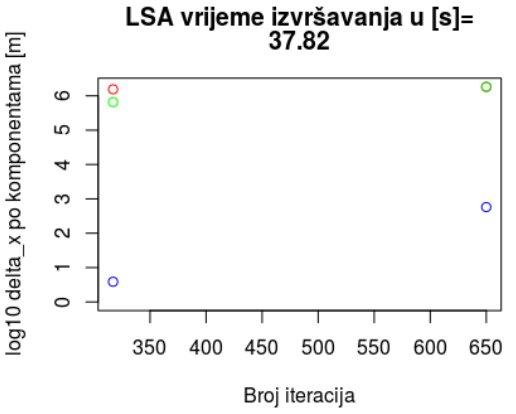


Slika 5.3: Logaritam vrijednosti komponenata $\Delta\mathbf{x}$ kroz iteracije

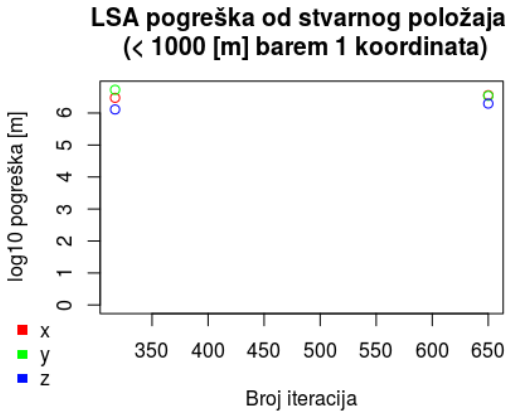


Slika 5.4: Logaritam pogreške određivanja položaja

Slike 5.3 i 5.4 prikazuju odnos izračunate pogreške u procjeni koordinata i stvarne pogreške u prvoj iteraciji u kojoj je barem jedna komponenta $\Delta\mathbf{x}$ manja od 100 m.



Slika 5.5: Logaritam vrijednosti komponenata $\Delta\mathbf{x}$ kroz iteracije



Slika 5.6: Logaritam pogreške određivanja položaja po komponentama

Slike 5.5 i 5.6 prikazuju odnos izračunate pogreške u procjeni koordinata i stvarne

pogreške komponenti za prvu i drugu iteraciju u kojima je barem jedna komponenta $\Delta\mathbf{x}$ manja od 1000 m.

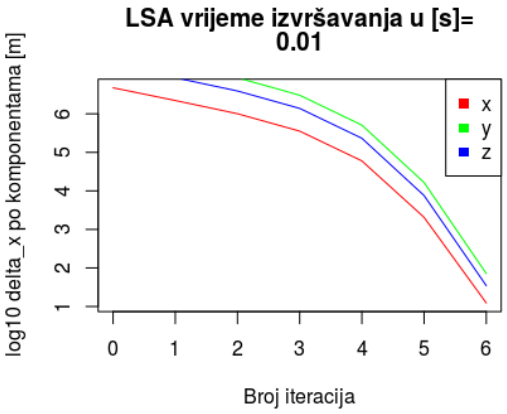
Promijenjen uvjet konvergencije daje relativno brzu konvergenciju, ali ne i konvergenciju prihvatljive točnosti. Prava pogreška određivanja položaja nije manja od 10^6 m.

Općenita konvergencija je jako spora, u smislu vremena izvršavanja i broja iteracija. Pogreška određivanja položaja dobivena općenitom konvegencijom je zadovoljavajuća, ali nije manja od 10^3 m. Dakle, izvedba algoritma zadovoljava kriterij točnosti određivanja položaja, ali ne i vremena potrebnog za određivanje položaja.

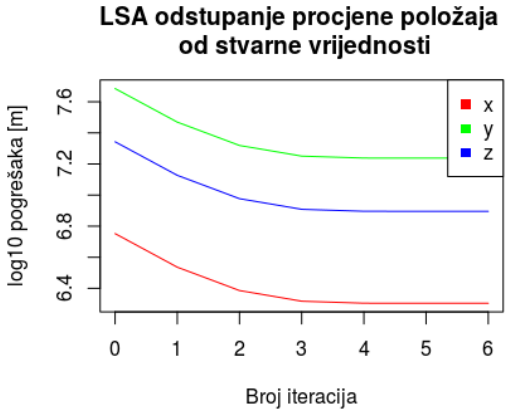
Razmotrimo sada drugu izvedbu osnovnog algoritma za procjenu položaja. Rezultati ocjene kvalitete su dani tablicom 5.2.

Tablica 5.2: Kvaliteta druge izvedbe osnovnog algoritma za procjenu položaja u domeni navigacijske primjene

NAZIV PARAMETRA	VRIJEDNOST
vrijeme izvršavanja [s]	0.01
broj iteracija do konvergencije	6
točnost procjene [m]	$> 10^6$



Slika 5.7: Logaritam vrijednosti Δx po komponentama kroz iteracije



Slika 5.8: Logaritam vrijednosti odstupanja od pravog položaja po komponentama kroz iteracije

Temeljem slika 5.7 i 5.8 zaključujemo kako druga izvedba algoritma 2.1. brzo konvergira, ali ka krivom rješenju.
Dobivena procjena položaja ima odstupanje veće od 10^6 metara.

Zaključak

Razmatrajući rezultate dviju izvedbi osnovnog algoritma ni jedna u potpunosti ne zadovoljava uvjete točnosti konvergencije i potrebnog vremena. Prva izvedba konvergira izrazito sporo k rješenju prihvatljive točnosti, dok druga konvergira brzo, ali ka krivome rješenju.
Budući da prilikom ocjene kvalitete izvedbi na prvo mjesto stavljamo točnost rješenja, boljom smatramo prvu izvedbu osnovnog algoritma. Prva izvedba osnovnog algoritma primjenom prvog načina linearizacije je spora i kao takva neupotrebljiva, ali za razuman broj iteracija daje bolju procjenu položaja.

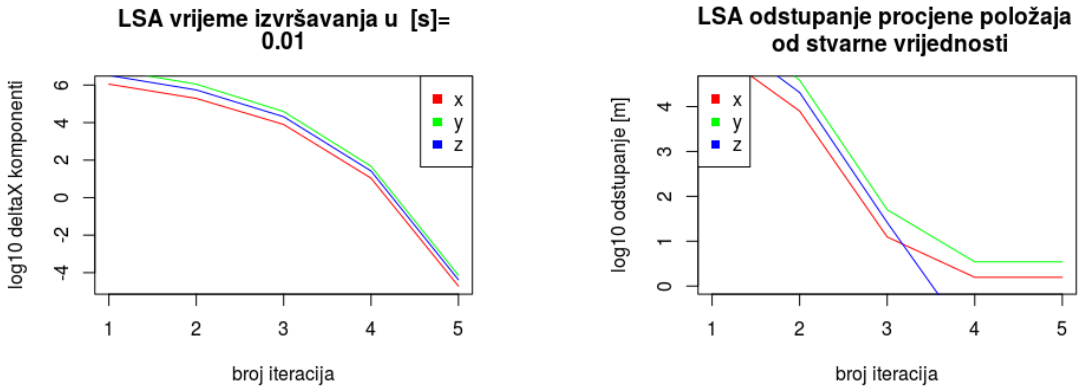
5.2 Poboljšani pristup

Postoji više inačica izvedbe poboljšanoga pristupa. Odabir se ostvaruje podešavanjima parametara *option* za odabir modela težinskih koeficijenata i *solution* za obabir načina rješavanja sustava (stranica 50). Budući da postoje dvije mogućnosti odabira za paramatar *solution* i tri za parametar *option*, sveukupno imamo šest inačica izvedbe poboljšanog pristupa.

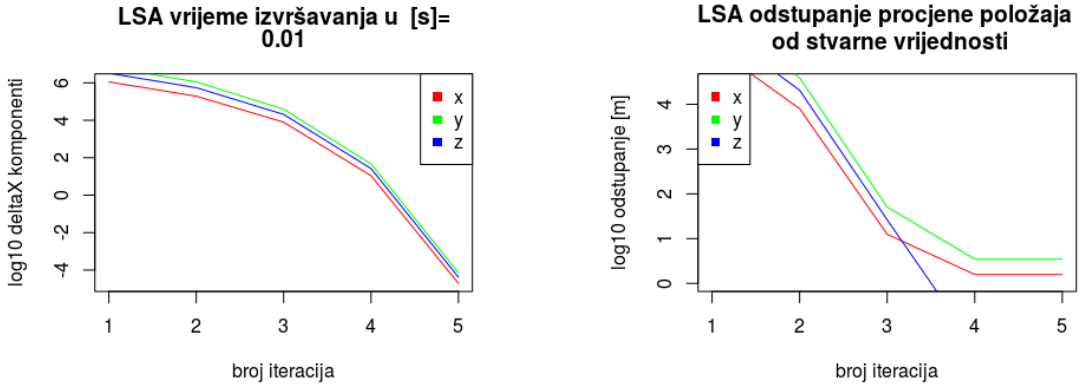
Ocjene kvalitete različitih inačica izvedbe dane su tablicama 5.3 i 5.4 i slikama 5.9, 5.10, 5.11, 5.12, 5.13 i 5.14.

Tablica 5.3: Ocjene kvalitete izvedbi poboljšanog algoritma za procjenu položaja u domeni navigacijske primjene koristeći QR dekompoziciju po modelima težinskih koeficijenata

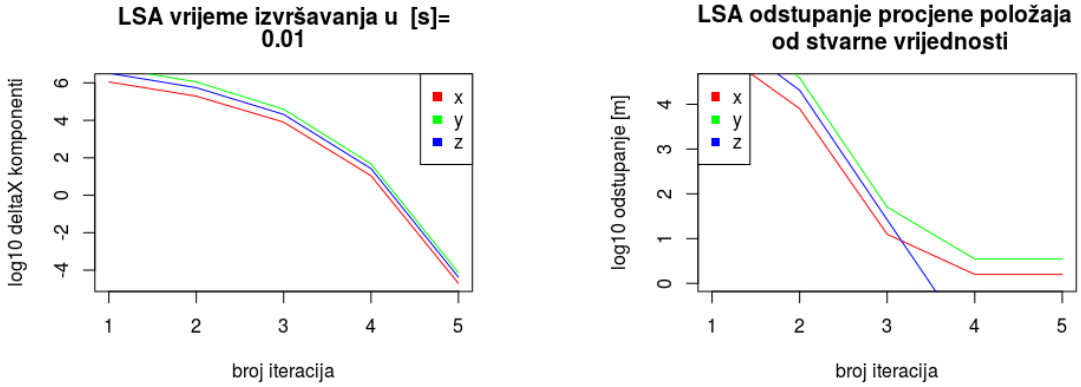
MODEL	1	2	3
NAZIV PARAMETRA	IZNOS	IZNOS	IZNOS
vrijeme izvršavanja	0.01	0.01	0.01
broj iteracija do konvergencije	5	5	5
točnost procjene po koordinatama (x,y,z)	$(10, 10, < 1)$	$(10, 10, < 1)$	$(10, 10, < 1)$



Slika 5.9: QR dekompozicija, prvi model težinskih koeficijenata



Slika 5.10: QR dekompozicija, drugi model težinskih koeficijenata

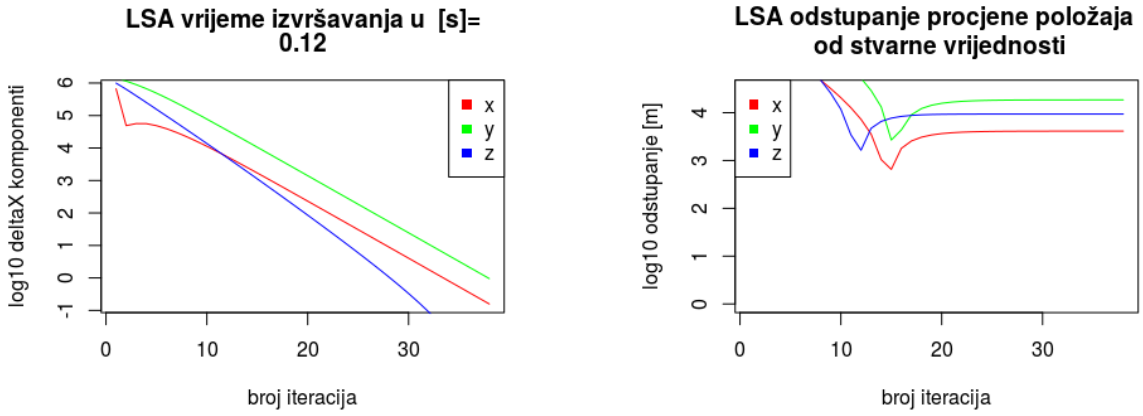


Slika 5.11: QR dekompozicija, treći model težinskih koeficijenata

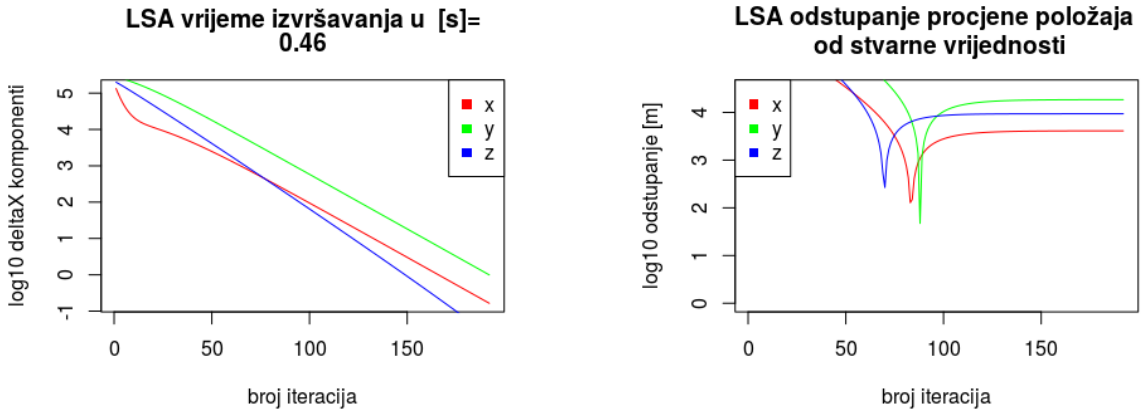
Tablica 5.3 i slike 5.9, 5.10 i 5.11 pokazuju kako nema značajne razlike u kvaliteti inačica izvedbe koje koriste QR dekompoziciju.

Tablica 5.4: Ocjene kvalitete izvedbi poboljšanog algoritma za procjenu položaja u domeni navigacijske primjene koje koriste sustav normalnih jednadžbi, po modelima težinskih koeficijenata

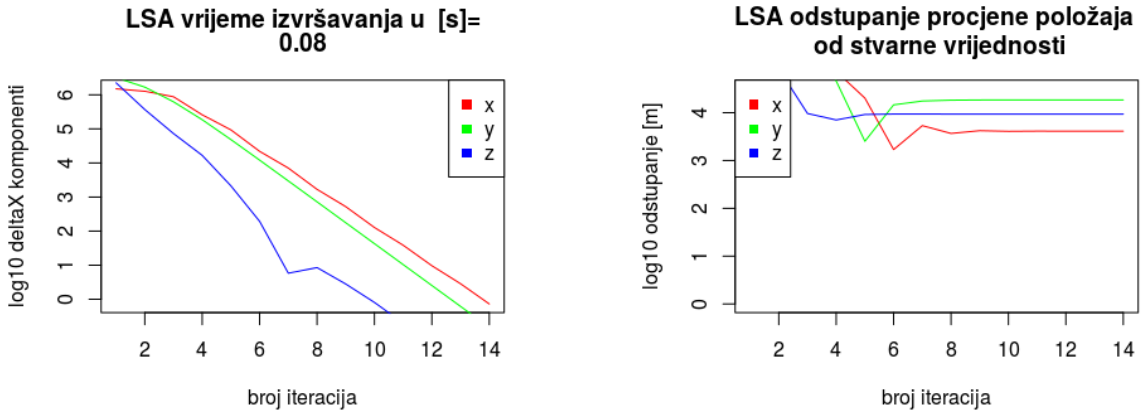
MODEL	1	2	3
NAZIV PARAMETRA	IZNOS	IZNOS	IZNOS
vrijeme izvršavanja	0.12	0.46	0.08
broj iteracija do konvergencije	38	192	14
točnost procjene po koordinatama (x,y,z)	$(10^{3.7}, 10^{4.3}, 10^{4})$	$(10^{3.6}, 10^{4.3}, 10^{4})$	$(10^{3.6}, 10^{4.3}, 10^{4})$



Slika 5.12: Sustav normalnih jednadžbi, prvi model težinskih koeficijenata



Slika 5.13: Sustav normalnih jednadžbi, drugi model težinskih koeficijenata



Slika 5.14: Sustav normalnih jednadžbi, treći model težinskih koeficijenata

Tablica 5.4 i slike 5.12, 5.13 i 5.14 ne pokazuju razlikovanje u točnosti različitih inačica izvedbi poboljšanog pristupa koje koriste sustav normalnih jednadžbi.

Ukoliko bolje usporedimo lijevi i desni graf svake slike posebno, vidljivo je kako do najmanjeg odstupanja procjene položaja ne dolazi nakon konvergencije nego ranije. Prva inačica ima najmanje odstupanje procjene položaja u $(15, 15, 12)$ iteraciji, druga u $(83, 88, 70)$, a treća u $(6, 5, 4)$ po koordinatama, redom. Najmanja odstupanja su jednaka redom $(653.8989, 2684.006, 1645.69)$, $(128.5274, 47.26717, 267.7062)$ i $(7055.983, 2527.817, 1696.067)$ metara po koordinatama. Dakle, prva (x) koordinata prve inačice ima najmanje odstupanje u 15.-oj iteraciji i ono iznosi 653.8989 m, druga koordinata (y) u 15.-oj iteraciji i ono iznosi 2684.006 m, treća koordinata (z) u 12.-oj

iteraciji i ono iznosi 1645.69 m. Prva koordinata druge inačice ima najmanje odstupanje u 83.-oj iteraciji i ono iznosi 128.5274 m, druga koordinata u 88.-oj iteraciji i ono iznosi 47.26717 itd.

Najbolje najmanje odstupanje ima druga inačica. Ono postiže vrijednost 10^2 m u najgorem slučaju.

Zaključak

Budući da se izvedba osnovnog pristupa koja koristi prvi način linearizacije pokazala boljom, sve inačice poboljšanog pristupa predstavljaju poboljšanje prve izvedbe osnovnog pristupa.
Temeljem rezultata inačica izvedbi poboljšanog pristupa, najboljima se pokazuju inačice koja koriste QR dekompoziciju. Među njima nema značajnih razlika. Nemogućnost razlikovanja ocjena kvalitete javlja se i zbog izrazito malog broja ostvarenih iteracija. Naime, mali broj iteracija ne dopušta da se razlike donekle sličnih modela značajno izraze.
Rezultati ocjene kvalitete inačica izvedbi koje koriste sustav normalnih jednadžbi nisu prihvatljivi. Sve inačice konvergiraju k rješenju odstupanja koje nije zadovoljavajuće, ali nije daleko od zadovoljavajućega. Vrijeme izvršavanja je proporcionalno brzini konvergacije. Također, korištenje sustava normalnih jednadžbi pokazuje anomaliju u smislu točnosti određivanja položaja. Postoji zamjetno, ali, u okvirima ovoga rada, neobjašnjivo, poboljšanje u točnosti određivanja položaja prije konvergencije algoritma koje se konvergencijom izgubi. Mogući uzrok značajnog pogoršanja točnosti nakon poboljšanja je u iznimnoj osjetljivosti sustava normalnih jednadžbi na perturbacije koje uzrokuje rješavanje sustava u aritmetici konačne točnosti.

5.3 Usporedba pristupa i zaključci

Promatrajući kvalitetu izvedbe pripadnog osnovnog i poboljšanog algoritma, poboljšani algoritam ostvaruje poboljšanje u točnosti procjene položaja, brzini konvergencije i vremenu izvršavanja.

Tablica 5.5: Usporedba pripadnog osnovnog i poboljšanog pristupa za procjenu položaja u domani navigacijske primjene s najmanjim odstupanjem procjene položaja

	osnovni	poboljšani
NAZIV PARAMETRA	IZNOS	IZNOS
vrijeme izvršavanja	48h	0.01s
broj iteracija do konvergencije	10^{10}	5
točnost procjene po koordinatama (x,y,z) [m]	$\approx (10^3, 10^3, 10^4)$	$(10, 10, < 1)$

Poboljšan pristup smanjuje broj iteracija sa 10^{10} na svega 5. Sukladno tome, smanjeno je i vrijeme izvršavanja (tablica 5.5). Odstupanje u točnosti je smanjeno za barem dva reda veličine, a u z koordinati i za 4.
Pametni odabir modela težinskih koeficijenata poboljšanog algoritma doveo je do poboljšanja u svim parametrima ocjene kvalitete.

Rezultati ocjena kvaliteta ukazuju kako postoji mogućnost djelotvornog ispravka dijela slučajnih pogrešaka primjenom poboljšanog postupka procjene položaja.
Kako bi primjena poboljšanog algoritma za dovoljno dugački niz pojedinačnih procjena položaja zadovoljila industrijske zahtjeve, potrebno je provesti dodatne validacije i po mogućnosti poboljšati brzinu izvršavanja izvedbe.
Danja validacija, razrada i praktične izvedbe navedenih i novih pristupa poboljšanja ostaju predmet budućih istraživanja.

Bibliografija

[1] S. Ali-Löytty, J. Collin, H. Leppäkoski, H. Sairo i N. Sirola, *Mathematics and methods for Positioning*, 2008, Tempare University of Technology, Tempare, Finska [Dostupno na: `http://math.tut.fi/courses/MAT-45806/mathematics_and_methods_for_positioning_2008.pdf`, pregledano 10. rujna 2017.].

[2] Anon, *Vulnerability assessment of the transportation infrastructure relying on the Global Positioning System*, John A. Volpe National Transportation Systems Center, Tech. Rep. (2001), Cambridge, MA, 2001.

[3] D. Blagojević, *Teorija satelitskog pozicioniranja*, [Dostupno na: `http://web.grf.bg.ac.rs/p/learning/teorija_satelitskog_pozicioniranja_1415876930596.pdf`, pregledano 10. rujna 2017.].

[4] R. G. Brown i P. W. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, 2009, John Wiley and Sons. New York, NY.

[5] C. Corcoran, *STAT 6550: Singular Value Decomposition in R*, 2007, Skripta uz predavanja, Utah State University, [Dostupno na: `http://www.math.usu.edu/~corcoran/classes/old/07spring6550/examples/svd.pdf` , pregledano 10. rujna 2017.].

[6] P. H. Dana, *Global positioning system overview*, 1994, University of Colorado, Boulder, CO [Dostupno na: `http://www.colorado.edu/geography/gcraft/notes/`, pregledano 9-Feb-2017].

[7] J. A. Farrell, *Aided Navigation: GPS with High Rate Sensors*, 2008, McGraw-Hill. New York, NY.

[8] M. Filić, *Laboratory Session 4: GNSS positioning performance assessment using RTKLIB/RTKPOST and R, (nastavni materijal)*, 2017, Diplomski studij satelitske navigacije, Regionalni UN centar za akademsko obrazovanje u svemirskim znanostima i tehnologijama za područje Azije i Pacifika, Međunarodna škola, Sveučilište Beihang za aeronautiku i astronautiku. Peking, Kina.

[9] M. Filić, R. Filjar i L. Ruotsalainen, *An SDR-based Study of Multi-GNSS Positioning Performance During Fast-developing Space Weather Storm*, 2016, TRANSNAV, 10(3), 395 – 400. doi:10.12716/1001.10.03.03 [Dostupno na: http://bit.ly/2fxAvph, pregledano 10. rujna 2017.].

[10] M. Filić, L. Grubišić i R. Filjar, *The consequence of a GPS satellite decommission on the quality of positioning for Intelligent Transport Systems*, 2016, Proc of KoREMA Automation in Transport Conference, 14-19. Krapina, Hrvatska, Ljubljana i Maribor, Slovenija.

[11] R. Filjar, *A Study of Direct Severe Space Weather Effects on GPS Ionospheric Delay*, 2008, Journal of Navigation, 61, 115-128, [Dostupno na: http://dx.doi.org/10.1017/S0373463307004420, pregledano 10. rujna 2017.].

[12] R. Filjar, D. Brčić i S. Kos, *Single-frequency Horizontal GPS positioning Error Response to a Moderate Ionospheric Storm over Northern Adriatic*, 2013, Poglavlje u knjizi: Weintrit, A. (editor) Advances in Marine Navigation (2013). Taylor and Francis Group, London, UK.

[13] R. Filjar, D Huljenić i S. Dešić, *Distributed Positioning: A Network-Supported Method for Satellite Positioning Performance Improvement*, 2002, J of Navigation, 55, 477-484.

[14] R. Filjar, D. Huljenić i K. Lenac, *Enhancing Performance of GNSS Position Estimation by Cloud-based GNSS SDR Receiver Architecture Utilisation*, 2013, Proc of IEEE International Symposium ELMAR 2013, 315-318, Zadar, Croatia.

[15] RINEX Working Group & Radio Technical Commission for Maritime Services Special Committee 104 (RTCM-SC104), *RINEX 3.03*, 2015, [Dostupno na: https://igscb.jpl.nasa.gov/igscb/data/format/rinex303.pdf, pregledano 10. rujna 2017.].

[16] R Foundation for Statistical Computing, *R: A language and environment for statistical computing*, 2016, [Dostupno na: http://www.r-project.org, pregledano 10. rujna 2017.].

[17] The R Foundation, *R programming language*, [Dostupno na: https://cran.r-project.org/, pregledano 10. rujna 2017.].

[18] F. Gustafsson, *Statistical sensor fusion, Studentlitteratur*, 2010, Linkoeping University, Linkoeping, Švedska.

[19] Y. He i A. Bilgic, *Iterative least squares method for global positioning system*, 2011, [Dostupno na: `https://pdfs.semanticscholar.org/fcb1/86f5b7feb713e970fd076498e93a77f7f2fc.pdf`, pregledano 10. rujna 2017.].

[20] D. Kao, *QR Factorization and Singular Value Decomposition*, 2016, Skripta uz predavanja, Princeton University, Princeton, NY [Dostupno na: `http://www.cs.princeton.edu/courses/archive/fall11/cos323/notes/cos323_f11_lecture09_svd.pdf` , pregledano 10. rujna 2017.].

[21] J. Maindonald i W. J. Brown, *Data Analysis and Graphics Using R - an Example-Based Approach*, 2010, Cambridge University Press. Cambridge, UK.

[22] United States military, *Global positioning system standard positioning service signal specification*, 1995, [Dostupno na: `http://www.gps.gov/technical/ps/1995-SPS-signal-specification.pdf`, pregledano 10. rujna 2017.].

[23] A. A. Nielsen, *Least Squares Adjustment: Linear and Nonlinear Weighted Regression Analysis*, 2013, Tehničko izvješće, Technical University of Denmark. Lyngby, Danska. [Dostupno na: `http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/2804/pdf/imm2804.pdf` , pregledano 10. rujna 2017.].

[24] N. Rahemi, M.R. Mosavi, A. A. Abedi i S. Mirzakuchaki, *Accurate Solution of Navigation Equations in GPS Receivers for Very High Velocities Using Pseudorange Measurements*, Advances in Aerospace engineering, Vol 20th, Article ID 435891 [Dostupno na: `https://www.hindawi.com/archive/2014/435891/`, pregledano 10. rujna 2017.].

[25] RStudio, *RStudio*, 2016, [Dostupno na: `https://www.rstudio.com/`, pregledano 10. rujna 2017.].

[26] J. Sanz Subirana et al., *GNSS Data Processing – Volume I: Fundamentals and Algorithms*, 2013, European Space Agency (ESA). Nordwijk, Nizozemska. `http://bit.ly/1QV4KAL`, pregledano 10. rujna 2017.].

[27] SIAM, *Linear Least Squares Problems*, [Dostupno na: `https://sites.math.washington.edu/~morrow/498_13/demmelsvd.pdf` , pregledano 10. rujna 2017.].

[28] S. Singer, *Numerička matematika, 7. predavanje*, 2017, [Dostupno na: `http://degiorgi.math.hr/~singer/num_mat/NM_1617/07.pdf`, pregledano 10. rujna 2017.].

[29] T. Suzuki, *Programska knjižnica GNSS-SDRLIB v2.0 Beta*, 2017, [Dostupno na: `https://github.com/taroz/GNSS-SDRLIB`, pregledano 10. rujna 2017.].

[30] T. Takasu, *RTKLIB open-source software package.*, 2017, [Dostupno na: `http://bit.ly/2gXSd42` , pregledano 10. rujna 2017.].

[31] M. Thomas et al., *Global Navigation Space Systems: reliance and vulnerabilities*, 2011, The Royal Academy of Engineering. London, UK. `http://bit.ly/1vrIenu`, pregledano 10. rujna 2017.].

[32] Z. Šimić, *Geodezija 1*, Geodetska tehnička škola Zagreb, [Dostupno na: `http://docplayer.gr/44835179-Zdravka-simic.html`, pregledano 10. rujna 2017.].

Sažetak

Rastući broj tehnoloških i društveno-ekonomskih sustava se zasniva na satelitskom određivanju položaja, npr. snalaženje u prostoru, analiza prometnih puteva, pridjeljivanje vremensko-položajnih žigova financijskim transakcijama itd. Kvaliteta usluge određuje se točnošću procjene položaja satelitskim sustavima.
Ovaj rad obuhvaća analizu (referentnog) postupaka procjene položaja u domeni navigacijske primjene te se otkrivaju potencijalne slabosti i predlaže poboljšanje. Analiza algoritma uključuje izvedbu i ocjenu kvalitete izvedbe algoritma korištenjem izmjerenih pseudoudaljenosti i programski određenog GPS prijemnika. Programski određen GPS prijemnik je praktično izveden na osobnom računalu. Korišten je u svrhu obrade prikupljenih opažanja GPS prijemnika u RINEX podatkovnom formatu za ulaz u izvedbe. Opažanja su prikupljena s referentne međunarodne GNSS stacionarne postaje (engl. International GNSS Service Reference Station, IGS Reference Station) u Padovi, Italija. Poboljšanja algoritma analizirana su usporednom analizom poboljšanog i izvornog algoritma.

Uvodnim se poglavljem uvodi u svijet globalnih satelitskih navigacijskih sustava (engl. Global Navigation Satellite System, GNSS) te se formuliraju ciljevi rada. Prvo poglavlje detaljnije opisuje jedan odabran GNSS, Globalni pozicijski sustav (GPS), i uvodi i egzaktno definira problem određivanja položaja u domeni navigacijske primjene. Odabrani GNSS se koristi pri prikupljanju eksperimentalih podataka.
Drugo poglavlje opisuje načine pronalaska rješenja problema određivanja položaja definiranog prvim poglavljem.
Pri prikupljanju i predobradi eksperimentalih podataka koristi se GNSS korisnička oprema zasnovana na programski određenom radiju. Opis programski određenog radija i korištene GNSS korisničke opreme nalazi se u trećem poglavlju. Četvrto poglavlja donosi praktičnu izvedbu osnovnog (referentnog) algoritma za procjenu položaja u domeni navigacijske primjene, razloge za njegovo poboljšanje te izvedbu i opis poboljšanja. Peto poglavlje ocjenjuje i uspoređuje kvalitetu dva izvedena algoritma, osnovnog i poboljšanog, nakon čega slijede zaključci.

Summary

Satellite positioning has been recognised as an fundamental underlying technology for growing number of modern technology and socio-economic systems, i.e. orientation in space, traffic routes analysis time and space stamping of financial transactions etc. The quality of services provided by those systems relies on the satellite positioning performance, especially positioning accuracy. This paper covers the analysis of (reference) position estimation procedures in the navigation application domain, discovering of potential weaknesses and suggesting the improvement. The algorithm analysis involves practical performance and rating of the algorithm performance quality using measured pseudorange and programmable GPS receiver. A program-specific GPS receiver is practically performed on a personal computer. It was used for processing the GPS receiver's observations in the RINEX data format to form the input of the algorithm performance. GPS receiver's observations are collected from IGS reference station located in Padua, Italy. Algorithm enhancements were analyzed by a parallel analysis of the improved and original algorithm.

The introductory chapter introduces the world of global navigation satellite systems (GNSS) and formulates the objectives. Chapter one details a selected GNSS, Global Positioning System (GPS), and introduces and defines the problem of positioning in the navigation application domain. Selected GNSS is used when collecting experimental data.
The second chapter describes solution finding methods of the problem defined by chapter one.
GNSS user equipment based on program-specific radio is used when collecting and pre-processing experimental data. The description of the programmable radio and the GNSS user equipment used are given with the third chapter. Chapter Four provides a practical performance of the basic (benchmark) algorithm for position assessment in the navigation application domain, the reasons for its improvement and performance and description of the improvement. The fifth chapter evaluates and compares the quality of two derived algorithms, basic and improved, followed by conclusions.

Životopis

Mia Filić je rođena u Zagrebu, Republika Hrvatska. Osnovnu i srednju školu je završila u Sesvetama, Zagreb, Republika Hrvatska u razdoblju od 2000. do 2012. Po završetku srednje škole upisuje, a 2015. završava preddiplomski studij matematike na Prirodoslovno-matematičkom fakultetu (Matematički odsjek) Sveučilišta u Zagrebu. Godine 2015. upisuje, a 2017. završava diplomski studij Računarstva i matematike na Prirodoslovno-matematičkom fakultetu (Matematički odsjek) Sveučilišta u Zagrebu. U sklopu diplomskog studija, zimski semestar akademske godine 2016/17 provodi na Sveučilištu u Ljubljani, Slovenija u okviru projekta Erasmus +. Njezni znanstveni i profesionalni interesi uključuju sljedeća područja: statističko i strojno učenje, procesiranje signala i navigacijski algoritmi za potrebe satelitske navigacije i programski određenog radija, detekcija anomalija i modeliranje pogrešaka satelitskog određivanja položaja, kriptografiju, teorijsko računarstvo ...

Tokom diplomskog studija uključila se u brojne interne znanstveno-istraživačke projekte, što je rezultiralo objavom jednog znanstvenog rada u međunarodnom znanstvenom časopisu TRANSNAV, predstavljanjem šest znanstvenih radova na međunarodnim znanstvenim skupovima i njihovom objavom u zbornicima s međunarodnom recenzijom, kao i pozvanim predavanjem na Boston College-u (Boston, MA, SAD) u okviru Međunarodnog skupa UN-a i SAD-a o svemirskoj meteorologiji i učincima na tehnološke sustave. Mia Filić aktivno djeluje i u području akademskog obrazovanja, sudjelujući kao pozvani asistent u izvođenju nastave na diplomskom studiju satelitske navigacije (prema programu UN-a) na Međunarodnoj školi Sveučilišta Beihang za aeronautiku i astronautiku, Peking, Kina.
Mia Filić je članica (Member) Kraljevskog instituta za navigaciju (The Royal Institut of Navigation, London, Ujedinjeno Kraljevstvo) te članica Međunarodnog programskog i organizacijskog odbora znanstvenog skupa Baška GNSS Conference, koja se tradicionalno svakog svibnja održava u Baški, otok Krk, Republika Hrvatska u organizaciji Kraljevskog instituta za navigaciju.

Osim znanstveno-istraživačkog rada, radi i u industriji. Tokom srednje škole, po-

vremeno radi u pozivnom centru tvrtke GSK d.o.o.. Od 1.07.2013 do 1.10.2013 radi kao pomoćni voditelj skladišta, a od 1.07.2015 do 1.10.2015 kao pomoćni stručni suradnik za analizu poslovnik podataka (engl. Business Analyst Assistant) u Sandoz d.o.o., Maksimirska 120, HR-10 000 Zagreb, Hrvatska.
2016. godine sudjeluje na ljetnom kampu tvrtke Ericsson Nikola Tesla, te ostaje raditi do završetka devetog mjeseca iste godine.
Sudjeluje i na studentskim natjecanjima eStudenta: Case Study Competition i Mozgalo. Na natjecanju Case Study Competition radi u timu na slučaju Zagrebačke Banke i dolazi do finala.

Dodatak A

Jakobijeva matrica funkcije h, J

Iz jednakosti (2.9) i $\mathbf{J} = \mathbf{p}(\mathbf{x})$ dobivamo

$$\mathbf{J} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \tag{A.1}$$

Za

$$\mathbf{h}(\mathbf{x}) := \begin{bmatrix} ||(\mathbf{s}_1 - \mathbf{x}_{1:3})|| \\ ||(\mathbf{s}_2 - \mathbf{x}_{1:3})|| \\ ||(\mathbf{s}_3 - \mathbf{x}_{1:3})|| \end{bmatrix} \tag{A.2}$$

dobivamo

$$\mathbf{J} = \begin{bmatrix} \frac{\partial}{\partial \mathbf{x}}||(s_1 - \mathbf{x}_{1:3})|| \\ \frac{\partial}{\partial \mathbf{x}}||(s_2 - \mathbf{x}_{1:3})|| \\ \frac{\partial}{\partial \mathbf{x}}||(s_3 - \mathbf{x}_{1:3})|| \end{bmatrix} = - \begin{bmatrix} \frac{(s_1 - \mathbf{x}_{1:4})^T}{||(s_1 - \mathbf{x}_{1:3})||} \\ \frac{(s_2 - \mathbf{x}_{1:4})^T}{||(s_1 - \mathbf{x}_{1:3})||} \\ \frac{(s_3 - \mathbf{x}_{1:4})^T}{||(s_1 - \mathbf{x}_{1:3})||} \end{bmatrix} = (\mathbf{J}_n(1:3, 1:3)) \tag{A.3}$$

uz $\hat{\mathbf{x}} = \mathbf{x}_n$

Dodatak B

Mjere kvalitete sazviježđa

Proces određivanja položaja prijamnika je najtočniji ako je međusobni položaj korištenih satelita povoljan. Kvaliteta međusobanog položaja korištenih satelita ovisi o njohovoj prostornojraspodjeli u odnosu na položaj prijamnika. Nepovoljan odnos između satelita rezultira gotovo zavisnim sustavom jednadžbi. Što je sustav jednadžbi bliži zavisnome, veća je mogućnost da prilikom procesa određivanja položaja prijamnika sustav zaista i postane zavisan. Zavisnost sustava uzrokovana je pogreškama zaokruživanja. One se mogu smanjiti, ali nikada u potpunosti izbjeći.

Jednim imenom se mjere međusobnog odnosa među satelitima ili mjere kvalitete sazviježđa nazivaju *degradacija točnosti* (DOP, engl. Dilution of precision). Niske vrijednosti DOP-a znače povoljan, dok visoke vrijednosti znače nepovoljan međusobni položaj satelita. U nastavku navodimo različite DOP mjere [23]. Uz

$$\sigma^2_{0_prior} := \frac{\mathbf{p_{prior}}^T \mathbf{W} \mathbf{p_{prior}}}{N-4} \tag{B.1}$$

$$\mathbf{p_{prior}} := (p_1(\mathbf{x_0}), p_2(\mathbf{x_0}), p_3(\mathbf{x_0}), p_4(\mathbf{x_0}))$$

$$\hat{\sigma}^2_0 := \frac{\hat{\mathbf{p}}^T \mathbf{W} \hat{\mathbf{p}}}{N-4} \tag{B.2}$$

imamo

$$\mathbf{Q}_{\hat{\mathbf{x}}} := \hat{\sigma}_0^2 (\mathbf{A^1}^T \mathbf{W} \mathbf{A^1})^{-1} \tag{B.3}$$

$$\mathbf{Q_{DOP}} := \frac{\mathbf{Q}_{\hat{\mathbf{x}}}}{\hat{\sigma}_0^2 \sigma_{0_prior}^2} \tag{B.4}$$

$$= \frac{(\mathbf{A^1}^T \mathbf{W} \mathbf{A^1})^{-1}}{\sigma_{0_prior}^2} \tag{B.5}$$

$$= \begin{bmatrix} q_X^2 & q_{YX} & q_{ZX} & q_{d_T X} \\ q_{XY} & q_Y^2 & q_{ZY} & q_{cd_T Y} \\ q_{XZ} & q_{YZ} & q_Z^2 & q_{cd_T Z} \\ q_{X cd_T} & q_{Y cd_T} & q_{Z cd_T} & q_{cd_T}^2 \end{bmatrix} \tag{B.6}$$

$$\mathbf{A^1} := \begin{bmatrix} \mathbf{A}[1:N, 1:3] & [1111]^T \end{bmatrix} \tag{B.7}$$

Prostorna degradacija točnosti određivanja položaja ($PDOP$, engl. position DOP) je određena izrazom

$$PDOP = \sqrt{q_X^2 + q_y^2 + q_Z^2} \tag{B.8}$$

Degradacija točnosti određivanja vremena ($TDOP$, engl. time DOP) je određena izrazom

$$TDOP = \sqrt{q_{cd_T}^2} \tag{B.9}$$

DOP formulacija koja objedunjuje prethodne je geometrijska degradacija točnosti ($GDOP$, engl. geometric DOP) određena je izrazom

$$GDOP = \sqrt{q_X^2 + q_y^2 + q_Z^2 + q_{cd_T}^2} \tag{B.10}$$

U praksi najčešće promatramo vrijednosti $PDOP$-a. Vrijednosti $PDOP$-a manje od 2 se smatraju odličnima, između 2 i 4 dobrima, a do 6 prihvatljivima. Vrijednosti iznad 6 su neprihvatljive i sugeriraju nepogodan međusoban položaj satelita.

Dalje definiramo $HDOP$ i $VDOP$.
Nakon transformacije gornje lijeve podmatrice matrice $\mathbf{Q} - \hat{\mathbf{x}}$ veličine 3×3 iz ECEF XYZ koordinata u ENU koordinate u odnosu na položaj prijamnika, dobivano matricu

$\mathbf{Q}_{ENU}$.

$$\mathbf{Q}_{DOP,ENU} := \frac{\mathbf{Q}_{ENU}}{\hat{\sigma}_0^2 \sigma_{0_{prior}}^2} \tag{B.11}$$

$$= \begin{bmatrix} q_E^2 & q_{NE} & q_{UE} \\ q_{EN} & q_N^2 & q_{UN} \\ q_{EU}^2 & q_{NU} & q_U^2 \end{bmatrix} \tag{B.12}$$

$$HDOP := \sqrt{q_E^2 + q_N} \tag{B.13}$$

$$VDOP := \sqrt{q_U^2} \tag{B.14}$$

$$EDOP := \sqrt{q_E^2 + q_N} \tag{B.15}$$

$$NDOP := \sqrt{q_U^2} \tag{B.16}$$

gdje $HDOP$ i $VDOP$ nazivamo horizontalna i vertikalna degradacija točnosti, a $EDOP$ i $NDOP$ su degradacija točnosti u smjeru istoka, odnosno sjevera.

Dodatak C

Kodovi izvedbe algoritama

C.1 Osnovni pristup

Prvi način linearizacije

```
#korišteni novi podatci
library("MASS","matrixcalc")

#pseudo-udaljenosti
c <- 2.99792458E+08 # brzina svjetlosti [m/s], po GPS standardu
R = read.csv('pseudorangesb.txt', header = FALSE);
R <- as.matrix(R[,1])

#učitaj koordinate satelita
S = read.csv('satellites.txt', header = FALSE)
S <- as.matrix(S)
x_0 = c(1,1,1,1) #[x,y,z,d_T] d_t se kasnije množi sa c da bi se oduzeo od [x_i,y_i,
    z_i,d]

delt = c(1000,1000,1000,3)
nRows = dim(S)[1]
nCols = dim(S)[2]+1

realPosition = c(918074.1038,5703773.539,2693918.9285,0)

inside = append(S,rep(-c,nRows))
RS = matrix(inside,nRows,nCols) # [x_i,y_i,z_i,d_i]

iter = 0
niter = 1000

err <- c(11,11,11,11)
#while(norm(t(delt)) > 1){

b = R

iter = 0
start.time <- Sys.time()
while((max(abs(delt[1:3])) > 1 || iter == 0)){
```

```
34
35    x_iter = c(x_0[1:3],0) #delta x_iter
36    AA = t(apply(RS, 1, function(x) (x_iter - x)))
37
38    #udaljenost satelita od procjenjenog položaja
39    D = sqrt(AA**2%*%c(1,1,1,0))
40    DD = matrix(append(rep(D,3),rep(1,nRows)),nRows,nCols)
41
42    A_iter = AA/DD
43    #https://www.math.ucla.edu/~anderson/rw1001/library/base/html/qr.html
44    # rješava sustav Ax=b koristeći QR faktorizaciju
45    delt <- qr.coef(qr(A_iter), b)
46
47    x_0 = x_0 + delt #(x,y,z,dT)
48    b = R - D - c*x_0[nCols]
49
50    #upisivanje vrijednosti dx radi kasnije analize brzine i točnosti postupka
51    if(iter%%100000 == 0){
52      cat(c(iter, delt[1:3]),' \r',file="razmakIteracija.txt", append=TRUE)
53      err <- x_0 - realPosition
54      cat(c(iter, err[1:3]),' \r',file="stvarnoOdstupanje.txt", append=TRUE)
55      print(A_iter)
56    }
57
58    iter = iter +1
59
60  }
61
62  end.time <- Sys.time()
63  timediff <- end.time - start.time
64
65  save.image("~/Documents/Faks/Diplomski/diplomski rad/Rsimulation/gltmp.RData")
66  #grafički prikaz preciznosti i točnosti odredivanja položaja kroz iteracije
67  d_iter <- read.csv('razmakIteracija.txt', header = FALSE, sep = '')
68  err <- read.csv('stvarnoOdstupanje.txt', header = FALSE, sep = '')
69
70  iter <- d_iter$V1[1:1E05]
71  d.x <- d_iter$V2[1:1E05]
72  d.y <- d_iter$V3[1:1E05]
73  d.z <- d_iter$V4[1:1E05]
74
75  plot(iter, log10(abs(d.x)), type = 'p', col = 'red', main = c('LSA vrijeme izvrš
        avanja u [s]=', round(timediff, digits = 2)), xlab = 'Broj iteracija', ylab = '
        log10 delta_x po komponentama[m]')
76  lines(iter, log10(abs(d.y)), type = 'p', col = 'green')
77  lines(iter, log10(abs(d.z)), type = 'p', col = 'blue')
78  legend(
79    "topright", legend=c("x","y", "z"), col=c("red", "green", "blue"),   pch=15
80  )
81
82  plot(iter, (abs(d.x)), type = 'l', col = 'red', main = c('LSA vrijeme izvršavanja u
        [s]=', round(timediff, digits = 2)), xlab = 'Broj iteracija', ylab = 'delta_x po
         komponentama [m]')
83  lines(iter, (abs(d.y)), type = 'l', col = 'green')
84  lines(iter, (abs(d.z)), type = 'l', col = 'blue')
85  legend(
86    "topright", legend=c("x","y", "z"), col=c("red", "green", "blue"),   pch=15
87  )
88
89  iter <- err$V1
```

```
xx <- err$V2
yy <- err$V3
zz <- err$V4

plot(iter, log10(abs(xx)), type = 'l', col = 'red', main = 'LSA pogreška od stvarnog
     položaja', xlab = 'Broj iteracija', ylab = 'log10 pogreška [m]')
lines(iter, log10(abs(yy)), type = 'l', col = 'green')
lines(iter, log10(abs(zz)), type = 'l', col = 'blue')
legend(
  "topright", legend=c("x","y", "z"), col=c("red", "green", "blue"),   pch=15
)

plot(iter, (abs(xx)), type = 'l', col = 'red', main = 'LSA pogreška od stvarnog polo
    žaja', xlab = 'Broj iteracija', ylab = 'pogreška [m]')
lines(iter, (abs(yy)), type = 'l', col = 'green')
lines(iter, (abs(zz)), type = 'l', col = 'blue')

save.image("~/Documents/Faks/Diplomski/diplomski rad/Rsimulation/gltmpCrtanjedata.
    RData")
file.remove('razmakIteracija.txt','stvarnoOdstupanje.txt')
```

Listing C.1: izvedba1.R

Drugi način linearizacije

```
library("MASS","matrixcalc")
#pseudo-udaljenosti
c <- 2.99792458E+08 # brzina svjetlosti [m/s], po GPS standardu
R = read.csv('pseudorangesb.txt', header = FALSE);
R <- as.matrix(R[,1])
#učitaj koordinate satelita
S = read.csv('satellites.txt', header = FALSE)
S <- as.matrix(S)
nRows = dim(S)[1]
nCols = dim(S)[2]+1

#konstante iteracije
W <- diag(nRows)

a_ele = 1
b_ele= 2

a = 1
psi = 0.5

change <- rep(TRUE,nCols-1) #indikatori mijenjanja koordinata (x,y,z) redom.
x_0 = c(1,1,1,1) #[x,y,z,d_T] d_t se kasnije množi sa c da bi se oduzeo od [x_i,y_i,
    z_i,d]
delt = c(3,3,3,3)

tezine = FALSE#ako želimo algoritam težinskih najmanjih kvadrata
#option = 1 # W se bira kao dijaginalna matrica s vrijednostima 1/sin(Ei) na
    dijaginali
#option = 2 # W se bira kao dijaginalna matrica s vrijednostima a_ele**2+b_ele**2/
    sin(Ei) na dijaginali
#option = 3 # W se bira kao dijaginalna matrica s vrijednostima a**2/sin(Ei+psi) na
    dijaginali
```

```
option = 3 #<- ok, izbjegava se singularitet u 0
solution = "Ch" #ili "Ch"; odabir modela rješavanja sustava.
elevation = 1 #način računanja kuta elevacije je jedna 1 za satelit-prijemnik i satelit-ishodište
#2 za prijemnik-satelit i normala tang. ravnine u točki prijemnika na sferu radijusa udaljenosti prijemnika od ishodišta
#slike rezultata spremnjene s nastavkom new.

realPosition = c(918074.1038,5703773.539,2693918.9285,0)

unutar = append(S,R)
RS = matrix(unutar,dim(S)[1],dim(S)[2]+1) # [x_i,y_i,z_i,d]

iter = 0
niter = 100
err <- c(11,11,11,11)
#while(norm(t(delt)) > 1){
start.time <- Sys.time()

while(iter < 100*niter && (max(abs(delt[1:3])) > 1 || iter == 0)){
  x_ = c(x_0[1:3],x_0[4]*c)
  P = t(apply(RS, 1, function(x) (x-x_))) #x-x_0
  D = sqrt((P*P)%*%c(1,1,1,0))
  DD = matrix(append(rep(D,3),rep(1,nRows)),nRows,nCols)

  PI = P
  PI[,dim(S)[2]+1] = - c*P[,dim(S)[2]+1] #4. red s -c

  PX = P*P
  b = PX%*%c(-1,-1,-1,1)
  A_iter = -2*PI

  if(tezine){
    #Procjena kuta elevacije satelita
    #n = (x_iter[i,1],x_iter[i,2],x_iter[i,3])
    #s = (S[i,1],S[i,2], S[i,3] )
    if(elevation == 1){
      xyz.coords.matrix = matrix(S,nRows,nCols-1)
    } else {
      xyz.coords.matrix = matrix(rep(x_iter[1:nCols-1],nRows),nRows,nCols-1)
      print(2)
    }

    xyz.coords.ssv <- sqrt(xyz.coords.matrix**2%*%c(1,1,1))#zbroj svih kooordinata na kvadrat
    xyz.coords.ssv = matrix(rep(xyz.coords.ssv,3),nRows,nCols-1)

    function.ele <- function(angle){
      if(angle < 0){
        a = -angle + pi/2
        return(pi-a)
      }else{
        return(pi/2 - angle)
      }
    }

    if(option==1){
      #budući da uvijek računamo x_iter - x, treba se nanovo izračunati x - x_iter. Smjer
      #vektora je obrnut. -AA
```

```
85
86        D_xyz = -P[1:nRows,1:(nCols-1)] # zraka x_iter do Si
87        D_xyz.ssv <- sqrt(D_xyz**2%*%c(1,1,1))
88        D_xyz.ssv = matrix(rep(D_xyz.ssv,3),nRows,nCols-1)
89
90        E <- acos(((xyz.coords.matrix/xyz.coords.ssv)*(D_xyz/D_xyz.ssv))%*%c(1,1,1))#
              kut izmedu normale tangencijalne ravnine i zrake prijemnik-satelit.
91        ele <- apply(E,1,function.ele)
92
93        Wii = 1/(sin(ele))^2
94        print("ele")
95        print(ele)
96
97      }else if(option==2){
98        D_xyz = -P[1:nRows,1:(nCols-1)]/DD[1:nRows,1:(nCols-1)]
99        D_xyz.ssv <- sqrt(D_xyz**2%*%c(1,1,1))
100       D_xyz.ssv = matrix(rep(D_xyz.ssv,3),nRows,nCols-1)
101
102       E <- acos(((xyz.coords.matrix/xyz.coords.ssv)*(D_xyz/D_xyz.ssv))%*%c(1,1,1))
103       ele <- apply(E,1,function.ele)
104
105       Wii = a_ele**2+b_ele**2/(sin(ele))^2
106       print("ele")
107       print(ele)
108     }else if(option == 3){
109       D_xyz = -P[1:nRows,1:(nCols-1)]/DD[1:nRows,1:(nCols-1)]
110       D_xyz.ssv <- sqrt(D_xyz**2%*%c(1,1,1))
111       D_xyz.ssv = matrix(rep(D_xyz.ssv,3),nRows,nCols-1)
112
113       E <- acos(((xyz.coords.matrix/xyz.coords.ssv)*(D_xyz/D_xyz.ssv))%*%c(1,1,1))
114       ele <- apply(E,1,function.ele)
115
116       Wii = a**2/(sin(ele)+psi)^2
117       print("ele")
118       print(ele)
119
120     }
121
122     W = diag(Wii)
123     b <- R[,1] - D - c*x_0[nCols] #x_0[4] modelira d_T i delta = delta_{d_T}
124
125     if(solution == "Ch"){
126       dx <- chol2inv(t(A_iter) %*% W %*% A_iter) %*% t(A_iter) %*% W %*% b
127     } else if (solution == "QR") {
128       #drugikorijen iz W
129       dx <- qr.coef(qr(sqrt(W)%*%A_iter), sqrt(W)%*%b)
130     } else {
131       print("Nepoznata opcija rješavanja sustava jednadžbi.")
132       exit(-1)
133     }
134
135     x_0 <- x_0 + dx*c(change,TRUE)
136     delt <- dx*c(change,TRUE)
137   }else{
138     delt <- qr.coef(qr(A_iter), b)
139     x_0 = x_0 + delt #(x,y,z,c*dT)
140   }
141
142   cat(c(iter, delt[1:3]),' \r',file="razmakIteracija.txt", append=TRUE) # upisivanje
          vrijednosti dx radi kasnije analize brzine i to?nosti postupka
```

```
  err <- x_0 - realPosition
  cat(c(iter, err[1:3]),' \r',file="stvarnoOdstupanje.txt", append=TRUE)

  iter = iter +1
  print(delt)
}

end.time <- Sys.time()

timediff <- end.time - start.time

d_iter <- read.csv('razmakIteracija.txt', header = FALSE, sep = '')
err <- read.csv('stvarnoOdstupanje.txt', header = FALSE, sep = '')

iter <- d_iter$V1
d.x <- d_iter$V2
d.y <- d_iter$V3
d.z <- d_iter$V4

plot(iter, log10(abs(d.x)),type = 'l', col = 'red', main = c('LSA vrijeme izvršavanja u [s]=', round(timediff, digits = 2)), xlab = 'Broj iteracija', ylab = 'log10 delta_x po komponentama [m]')
lines(iter, log10(abs(d.y)), type = 'l', col = 'green')
lines(iter, log10(abs(d.z)), type = 'l', col = 'blue')
legend(
  "topright", legend=c("x","y", "z"), col=c("red", "green", "blue"),   pch=15
)

plot(iter, (abs(d.x)), type = 'l', col = 'red', main = c('LSA vrijeme izvršavanja u [s]=', round(timediff, digits = 2)), xlab = 'Broj iteracija', ylab = 'delta_x po komponentama [m]')
lines(iter, (abs(d.y)), type = 'l', col = 'green')
lines(iter, (abs(d.z)), type = 'l', col = 'blue')
legend(
  "topright", legend=c("x","y", "z"), col=c("red", "green", "blue"),   pch=15
)

iter <- err$V1
xx <- err$V2
yy <- err$V3
zz <- err$V4

plot(iter, log10(abs(xx)),
     ylim = c(min(min(log10(abs(xx))),min(log10(abs(yy))),min(log10(abs(zz)))), max(max(log10(abs(xx))),max(log10(abs(yy))),max(log10(abs(zz))))),
     type = 'l', col = 'red', main = 'LSA odstupanje procjene položaja \n od stvarne vrijednosti', xlab = 'Broj iteracija', ylab = 'log10 pogrešaka [m]')
lines(iter, log10(abs(yy)), type = 'l', col = 'green')
lines(iter, log10(abs(zz)), type = 'l', col = 'blue')
legend(
"topright", legend=c("x","y", "z"), col=c("red", "green", "blue"),   pch=15
)

plot(iter, (abs(xx)),
     ylim = c(min(min(abs(xx)),min(abs(yy)),min(abs(zz))), max(max(abs(xx)),max(abs(yy)),max(abs(zz)))),
     type = 'l', col = 'red', main = 'LSA odstupanje procjene položaja \n od stvarne vrijednosti', xlab = 'Broj iteracija', ylab = 'pogreška [m]')
lines(iter, (abs(yy)), type = 'l', col = 'green')
lines(iter, (abs(zz)), type = 'l', col = 'blue')
```

```
legend(
  "topright", legend=c("x","y", "z"), col=c("red", "green", "blue"),   pch=15
)

file.remove('razmakIteracija.txt','stvarnoOdstupanje.txt')
```

Listing C.2: izvedba2.R

C.2 Poboljšan pristup: uvođenje težina (WLSM)

```
rm(list=ls())

library(MASS)
library(matlib)
library(limSolve)
library(matrixcalc)

iter = 0
niter = 100
#option = 1 # W se bira kao dijaginalna matrica s vrijednostima 1/sin(Ei) na dijaginali
#option = 2 # W se bira kao dijaginalna matrica s vrijednostima a_ele**2+b_ele**2/sin(Ei) na dijaginali
#option = 3 # W se bira kao dijaginalna matrica s vrijednostima a**2/sin(Ei+psi) na dijaginali
option = 1 #<- ok, izbjegava se singularitet u 0
solution = "Snj" #"QR" ili "Snj"; odabir modela rješavanja sustava.
elevation = 1 #način računanja kuta elevacije je jedna 1 za satelit-prijemnik i satelit-ishodište
  #2 za prijemnik-satelit i normala tang. ravnine u točki prijemnika na sferu radijusa udaljenosti prijemnika od ishodišta
  #slike rezultata spremnjene s nastavkom new.
c <- 2.99792458E+08 # brzina svjetlosti [m/s], po GPS standardu
R = read.csv('pseudorangesb.txt', header = FALSE);
R <- as.matrix(R[,1])
realPosition <- c(918074.1038,5703773.539,2693918.9285)

#ucitaj koordinate satelita
S = read.csv('satellites.txt', header = FALSE)
S <- as.matrix(S)
nRows = dim(S)[1]
nCols = dim(S)[2]+1

#konstante iteracije
RR <- rep(0,nRows)
dpr <- c(1,1)
W <- diag(nRows)

a_ele = 1
b_ele= 2

a = 1
psi = 0.5


if(dim(R)[1] < nCols){
```

```
  stop('Not enough satellites - unable to estimate position. The script will quit.')
}

#pocetni uvjeti
delt <- c(11,11,11,11)
err <- c(11,11,11,11)
x_0 <- c(11, 11, 11, 0) #nije (0,0,0,0) kako bi se izbjegnuo singularitet prilikom prve iteracije i izračunavanja
                  # kuta elevacije

unutar = append(S,rep(-c,nRows))
SC = matrix(unutar,nRows,nCols) # [x_i,y_i,z_i,c]

start.time <- Sys.time()# za regular position estimation (pretpostavljena potpuna kompenzacija pogre?aka)
eps <- 1 #eps > max(delt(x), delt(y), delt(z)), kriterij zaustavljanja
#zanimljive stvari se dogadaju za 10^3.2

A_iter <- matrix(nrow = nRows, ncol=nCols)

rlevel <- 10^5 #max(delt(x), delt(y), delt(z))
b <- R

#ako pomak pojedine iteracije padne ispod određene vrijednosti, ne mijenjamo više tu koordinatu
change <- rep(TRUE,nCols-1) #indikatori mijenjanja koordinata (x,y,z) redom.
chl = FALSE
start.time <- Sys.time()

while(eps < rlevel && iter < 1000){
  # iteracija - sve dok sve pogreske po komponentama ne budu manje od eps
  x_iter = c(x_0[1:3],0) # samo c , a ne c-d_T
  AA = t(apply(SC, 1, function(x) (x_iter - x))) #zbog lakse derivacije je x_-x
  D = sqrt((AA*AA)%*%c(1,1,1,0))
  DD = matrix(append(rep(D,3),rep(1,nRows)),nRows,nCols)

  A_iter = AA/DD #J_k

  #Procjena kuta elevacije satelita
  #n = (x_iter[i,1],x_iter[i,2],x_iter[i,3])
  #s = (S[i,1],S[i,2], S[i,3] )
  if(elevation == 1){
    xyz.coords.matrix = matrix(S,nRows,nCols-1)
  } else {
    xyz.coords.matrix = matrix(rep(x_iter[1:nCols-1],nRows),nRows,nCols-1)
    print(2)
  }

  xyz.coords.ssv <- sqrt(xyz.coords.matrix**2%*%c(1,1,1))#zbroj svih kooordinata na kvadrat
  xyz.coords.ssv = matrix(rep(xyz.coords.ssv,3),nRows,nCols-1)

  function.ele <- function(angle){
    if(angle < 0){
      a = -angle + pi/2
      return(pi-a)
    }else{
      return(pi/2 - angle)
    }
  }
```

```
98
99  if(option==1){
100   #budući da uvijek računamo x_iter - x, treba se nanovo izračunati x - x_iter.
          Smjer
101   #vektora je obrnut. -AA
102
103   D_xyz = -AA[1:nRows,1:(nCols-1)] # zraka x_iter do Si
104   D_xyz.ssv <- sqrt(D_xyz**2%*%c(1,1,1))
105   D_xyz.ssv = matrix(rep(D_xyz.ssv,3),nRows,nCols-1)
106
107   E <- acos(((xyz.coords.matrix/xyz.coords.ssv)*(D_xyz/D_xyz.ssv))%*%c(1,1,1))#kut
          izmedu normale tangencijalne ravnine i zrake prijemnik-satelit.
108   ele <- apply(E,1,function.ele)
109
110   Wii = 1/(sin(ele))^2
111   print("ele")
112   print(ele)
113
114 }else if(option==2){
115   D_xyz = -A_iter[1:nRows,1:(nCols-1)]
116   D_xyz.ssv <- sqrt(D_xyz**2%*%c(1,1,1))
117   D_xyz.ssv = matrix(rep(D_xyz.ssv,3),nRows,nCols-1)
118
119   E <- acos(((xyz.coords.matrix/xyz.coords.ssv)*(D_xyz/D_xyz.ssv))%*%c(1,1,1))
120   ele <- apply(E,1,function.ele)
121
122   Wii = a_ele**2+b_ele**2/(sin(ele))^2
123   print("ele")
124   print(ele)
125 }else if(option == 3){
126   D_xyz = -A_iter[1:nRows,1:(nCols-1)]
127   D_xyz.ssv <- sqrt(D_xyz**2%*%c(1,1,1))
128   D_xyz.ssv = matrix(rep(D_xyz.ssv,3),nRows,nCols-1)
129
130   E <- acos(((xyz.coords.matrix/xyz.coords.ssv)*(D_xyz/D_xyz.ssv))%*%c(1,1,1))
131   ele <- apply(E,1,function.ele)
132
133   Wii = a**2/(sin(ele)+psi)^2
134   print("ele")
135   print(ele)
136
137 }
138
139 W = diag(Wii)
140 b <- R[,1] - D - c*x_0[nCols] #x_0[4] modelira d_T i delta = delta_{d_T}
141
142 if(solution == "Snj"){
143   dx <- chol2inv(t(A_iter) %*% W %*% A_iter) %*% t(A_iter) %*% W %*% b
144 } else if (solution == "QR") {
145   #drugikorijen iz W
146     dx <- qr.coef(qr(sqrt(W)%*%A_iter), sqrt(W)%*%b)
147 } else {
148   print("Nepoznata opcija rješavanja sustava jednadžbi.")
149   exit(-1)
150 }
151 #dx <- svd.inverse(t(A_iter) %*% W %*% A_iter) %*% t(A_iter) %*% W %*% b #
      najpreciznija
152
153 x_0 <- x_0 + dx*c(change,TRUE)
154
```

```
155   iter <- iter + 1
156
157   cat(c(iter, dx[1], dx[2], dx[3]),' \r',file="razmakIteracija.txt", append=TRUE) #
          upisivanje vrijednosti dx radi kasnije analize brzine i to?nosti postupka
158   err[1] <- x_0[1] - 918074.1038
159   err[2] <- x_0[2] - 5703773.539
160   err[3] <- x_0[3] -2693918.9285
161   cat(c(iter, err[1], err[2], err[3]),' \r',file="stvarnoOdstupanje.txt", append=
          TRUE)
162   # Kontrola
163
164   print(cat("iter",iter))
165   print (S)
166   print (dx)
167
168   #dovodi do poboljšanja ukoliko je broj iteracija veći
169   rlevel <- max(abs(dx[1:3,1]))
170
171 }
172
173
174 end.time <- Sys.time()
175
176 timediff <- end.time - start.time
177
178 d_iter <- read.csv('razmakIteracija.txt', header = FALSE, sep = '')
179 err <- read.csv('stvarnoOdstupanje.txt', header = FALSE, sep = '')
180
181 iter <- d_iter$V1
182 d.x <- d_iter$V2
183 d.y <- d_iter$V3
184 d.z <- d_iter$V4
185
186 plot(iter, log10(abs(d.x)), type = 'l', col = 'red', main = c('LSA vrijeme izvrš
        avanja u  [s]=', round(timediff, digits = 2)), xlab = 'broj iteracija', ylab = '
        log10 deltaX komponenti')
187 lines(iter, log10(abs(d.y)), type = 'l', col = 'green')
188 lines(iter, log10(abs(d.z)), type = 'l', col = 'blue')
189 legend(
190   "topright", legend=c("x","y", "z"), col=c("red", "green", "blue"),   pch=15
191 )
192
193 plot(iter, (abs(d.x)), type = 'l', col = 'red', main = c('LSA vrijeme izvršavanja u
         [s]=', round(timediff, digits = 2)), xlab = 'broj iteracija', ylab = 'deltaX
        komponente')
194 lines(iter, (abs(d.y)), type = 'l', col = 'green')
195 lines(iter, (abs(d.z)), type = 'l', col = 'blue')
196 legend(
197   "topright", legend=c("x","y", "z"), col=c("red", "green", "blue"),   pch=15
198 )
199
200 iter <- err$V1
201 xx <- err$V2
202 yy <- err$V3
203 zz <- err$V4
204
205 plot(iter, ylim=c(0,4.5),log10(abs(xx)), type = 'l', col = 'red', main = 'LSA
        odstupanje procjene položaja \n od stvarne vrijednosti', xlab = 'broj iteracija'
        , ylab = 'log10 odstupanje [m]')
206 lines(iter, log10(abs(yy)), type = 'l', col = 'green')
```

```
207 lines(iter, log10(abs(zz)), type = 'l', col = 'blue')
208 legend(
209   "topleft", legend=c("x","y", "z"), col=c("red", "green", "blue"),   pch=15
210 )
211
212 plot(iter, (abs(xx)), type = 'l', col = 'red', main = 'LSA odstupanje procjene polož
      aja \n od stvarne vrijednosti', xlab = 'broj iteracija', ylab = 'odstupanje [m]'
      )
213 lines(iter, (abs(yy)), type = 'l', col = 'green')
214 lines(iter, (abs(zz)), type = 'l', col = 'blue')
215 legend(
216   "topleft", legend=c("x","y", "z"), col=c("red", "green", "blue"),   pch=15
217 )
218
219 file.remove('razmakIteracija.txt','stvarnoOdstupanje.txt')
```

Listing C.3: izvedba3.R

Dodatak D

Sadržaj datoteka ulaza izvedenih algoritama u procesu ocjene kvalitete

Slika D.1: Skup mjerenih pseudoudaljenosti

```
21345372.96948
21123433.31848
23647148.85446
22030908.95548
23234206.55447
20047262.99349
23831204.72647
```

Slika D.2: Skup ispravljenih pseudoudaljenosti

```
21364414.7719640
21133235.1936572
23653202.7278275
22023593.4330683
23223303.5998171
20094101.9563987
23840149.6080318
```

Slika D.3: Skup koordinata položaja satelita u WGS84 koordinatnom sustavu

```
-11443558.1932368,21853722.8998174,9284051.5634504
8849865.3721608,15211504.9991917,19837992.2835602
-12879946.2471086,8427911.5293681,21729597.7908060
-6223856.2333828,25502417.3739922,-3839628.4978272
10426045.9627803,21828156.0737286,-11075665.2807472
1511437.6130666,23698150.4953570,11649872.9017268
19831157.5365209,6560622.8041700,17206279.4938024
```