

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

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

Zagreb, 12. srpnja 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. _____

Na kraju

Sadržaj

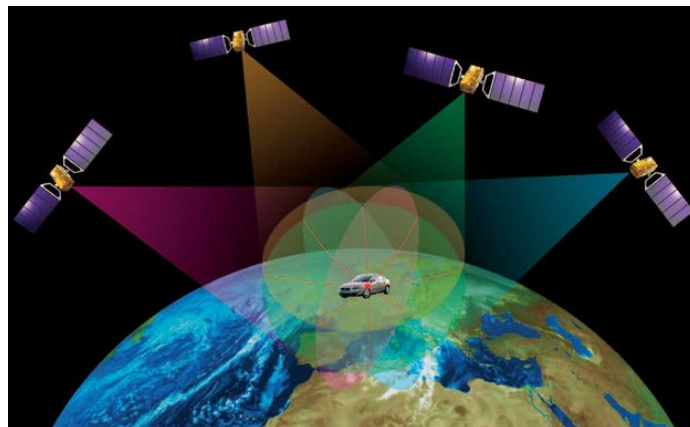
| | |
|--|----|
| Sadržaj | v |
| Uvod | 1 |
| 1 Globalni navigacijski satelitski sustav (GNSS) | 3 |
| 2 Globalni pozicijski sustav (GPS) | 7 |
| 2.1 C/A PRN kod | 8 |
| 2.2 P kod | 9 |
| 2.3 Pogreške određivanja položaja i vrste | 9 |
| 2.4 Navigation Message | 12 |
| 2.5 Proces određivanja položaja | 14 |
| 3 Algoritam procjene položaja (APP) | 19 |
| 3.1 Iterativna metoda najmanjih kvadrata | 20 |
| 3.2 Metoda zatvorene forme | 25 |
| 3.3 Metoda najbližeg susjeda (maksimalne vjerodostojnosti) | 27 |
| 4 Dizajn i izvedba algoritma, procjena točnosti | 29 |
| 4.1 Programski jezik R | 29 |
| 4.2 Zahtjevi algoritma | 29 |
| 4.3 Izvedba | 30 |
| 4.4 Zaključak | 42 |
| Bibliografija | 43 |
| A Taylorov red potencija | 51 |
| B Jakobijeva matrica funkcije h , J | 53 |
| C Mjere kvalitete "zviježđda" | 55 |

Uvod

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

Poglavlje 1

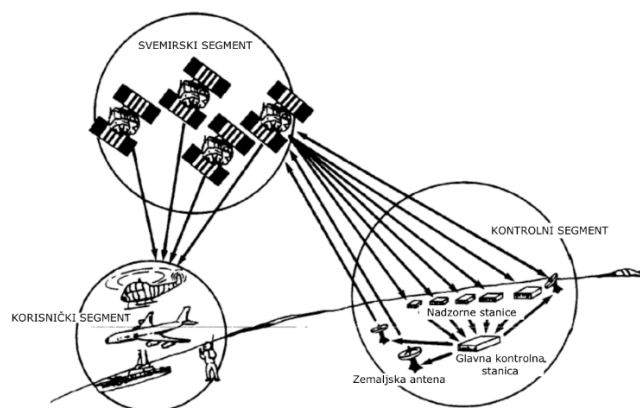
Globalni navigacijski satelitski sustav (engl. Global Navigation Satellite System (GNSS))



Slika 1.1: Satelitska navigacija[6]

Spominjući GNSS, najčešće se misli na *"sazviježđe"* satelita koji odašilju signale, potrebne za određivanje trenutne pozicije, i *Navigacijske poruke* (engl. Navigation Messages (NM)). *"Sazviježđe"* satelita predstavlja (1) svemirski segment GNSS sustava. Postoji još (2) kontrolni segment koji čine kontrolne stanice smještene na Zemlji i (3) korisnički segment, tj. GNSS prijemnici (Slika 1.2). Kontrolni segment nadzire i upravlja radom sustava.

Trenutno postoji više GNS sustava (GNSS). Neki su u potpunosti operativni, a neki samo djelomično. Najraširaniji u civilnoj upotrebi je GPS (Global Positioning



Slika 1.2: Segmenti GNS sustava (GNSS)

System). GPS je u potpunosti operativan i u vlasništvu Vlade SAD-a. Njime upravlja Ministarstvo obrane SAD-a (engl. US Air Force). GPS omogućava dvije znatno različite razine korištenja, civilnu i vojnu. Vojna razina korištenja pruža mogućnosti, a dopuštena je samo određenim korisnicima. Civilna razina korištenja je dostupna svima, bez dodatne naknade, uz uvjet posjedovanja GPS-prijemnika.

Drugi, također u potpunost operativan GNSS, je GLONASS (Global'naya Navigatsionnaya Sputnikovaya Sistema) koji je u vlasništvu Rusije. Postoje i GNSS sustavi u razvoju. Jedan od njih je Galileo. Galileo-om upravlja Europska unija (EU). Najavljeno je da će postati u potpunosti operativan do 2020 [6]. Kina posjeduje lokalni navigacijski satelitski sustav BeiDou na kojemu se radi da postane globalni najkasnije do 2020[6].

Primjena GNSS-a dijeli se na pozicioniranje i navigaciju.

Definicija 1.0.1 (Navigacija). *Navigacija obuhvaća trenutno određivanje položaja i brzine entiteta u pokretu. Svrha navigacije je praćenje i upravljanje gibanja entiteta.*

Definicija 1.0.2 (Pozicioniranje). *Pozicioniranje nazivamo postupak određivanja položaja točkovnog entiteta ili niza točkovnih entiteta u prostoru.*

Ovaj rad se bavi isključivo bespojenom (engl. off-line) navigacijskom primjenom, u svrhu praćenja 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 , tj. pozicioniranje. Određujući položaj entiteta za niz vremena t_1, t_2, \dots, t_n , dobiva se aproksimacija kretanja entiteta u vremenskom okviru $[t_1, t_n]$.

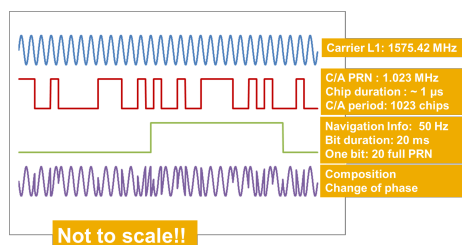
Preciznost aproksimacije kretanja zadaje se veličinom okvira i parametrom n , ili dostupnošću podataka. Praksa ne zathjeva da je n u odnosu na vremenski okvir duljine 1 sata prevelik. Točno kretanje entiteta moguće je odrediti preslikavanjem dobivene aproksimacije na kartu prometnih puteva. U tu svrhu se koriste otprije poznati algoritmi. Dakle, u ovom radu se bavi algoritmom za pozicioniranje (statičkog entiteta) u konceptu jednog određenog GNSS-a, GPS-a u aspektu civilne razine korištenja.

Poglavlje 2

Globalni pozicijski sustav (GPS, engl. Global Positioning System)

Sazvježđe GPS-a se sastoji od najmanje 24 satelita raspoređenih u 6 jednako odmaknutih orbita, svaka s inklanacijom od 55 stupnjeva od ekvatorijalne ravnine (engl. Medium Earth Orbit (MEO)). Sateliti kruže na visini od oko 20200 kilometara od Zemljine površine s periodom rotacije 12 zvjezdanih sati. 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 tekstu (Stranica 14).

Svi GPS sateliti odašilju signale na istoj osnovnoj frekvenciji/frekvencijama (Slika 2.1). U satelitima, vrijeme je praćeno pomoću cezijevih satova koji se sinkroniziraju s univerzalnom GPS atomskom vremenskom skalom. Sinkronizacija se odvija u periodima.



Slika 2.1: GPS signal i njegove komponente [7]

2.1 GPS signali: C/A PRN i P kod

C/A PRN kod i primjene

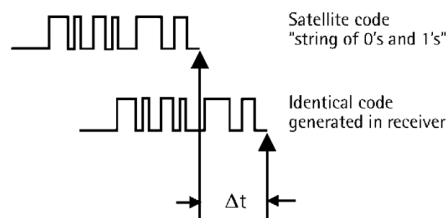
GPS sateliti odašilju signale na dvije frekvencije (nosača) L_1 i L_2 , od kojih je L_1 na 1575.42 MHz namjenjena civilnoj upotrebi. Pojam signal se često u satelitskoj navigaciji koristi samo za dio GPS signala koja sadržava C/A PRN kod (eng. Coarse Acquisition Pseudo Random Noise). Svaki satelit posjeduje jedinstveni C/A PRN kod koji predstavlja niz 0 i 1 duljine 1023 bit-a. GPS-prijemnik 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 odašiljatelja, ali i za računanje pseudo-udaljenosti.

Definicija 2.1.1 (Pseudo-udaljenost). *Naka su svi sateliti numerirani prirodnim brojevima s početkom u 1. Naka je $i \in \mathbb{N}$ neki satelit i t prijemnik koji je u mogućnosti primiti signal koji odašilje satelit i . Pseudo-udaljenost između satelita odašiljatelja i i prijemnika primatelja t :*

$$d_i = c \cdot (t'_i - t_i)$$

gdje je c konstanta koja je jednaka (prosječnoj) brzini putovanja signala od satelita do prijemnika. t'_i je vrijeme primanja signala, a t_i vrijeme slanja signala (po UTC vremenu).

Pseudo-udaljenost je aproksimacija udaljenosti između satelita odašiljatelja i prijemnika primatelja signala u određenom trenutku. Neka je $\delta t := (t'_i - t_i)$. Vrijeme putovanja signala izračunava se poravnavanjem odgovarajućih dijelova signala, tj. C/A PRN kodova. Naime, prijemnik i satelit istovremeno generiraju isti C/A PRN kod. Budući da dok signala putuje, prijemnik još uvijek generira C/A PRN kod, po primitku signala, ta 2 koda se uspoređuju, poravnavaju. Temeljem razlike u poravnanju, dobivenog i generiranog C/A PRN koda, računa se procjena vremena putovanja, tj. δt (Slika 2.2).



Slika 2.2: Procjena vremena putovanja signala (δt)

Za vrijednost konstante c se uzima brzina svjetlosti koja predstavlja brzinu putovanja poruke satelita u vakuumu. Ona dovoljno dobro modelira stvarnu prosječnu brzinu putovanja.

Budući da se psudo-udaljenost dobiva poravnavanjem kodova, upravo opisani način određivanja pseudo-udaljenosti naziva se kodni.

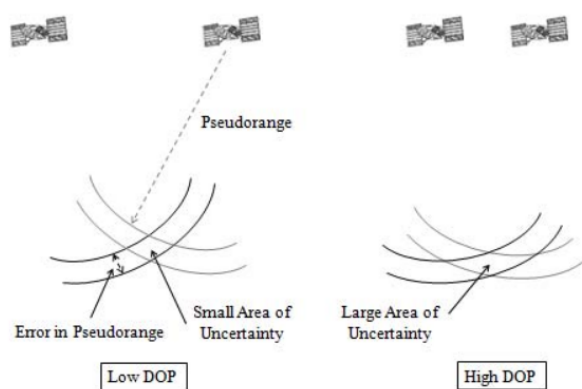
Postoji još i fazni način određivanja psudo-udaljenosti koji se zasniva na poravnanju valova nosača (engl. Carrier phase) nakon micanjem C/A PRN i P(Y) kodova iz poruke (Slika 2.1). Fazno mjerenje služi kao nadopuna kodnom mjerenju u svrhu poboljšanja točnosti određivanja položaja.

2.2 P kod

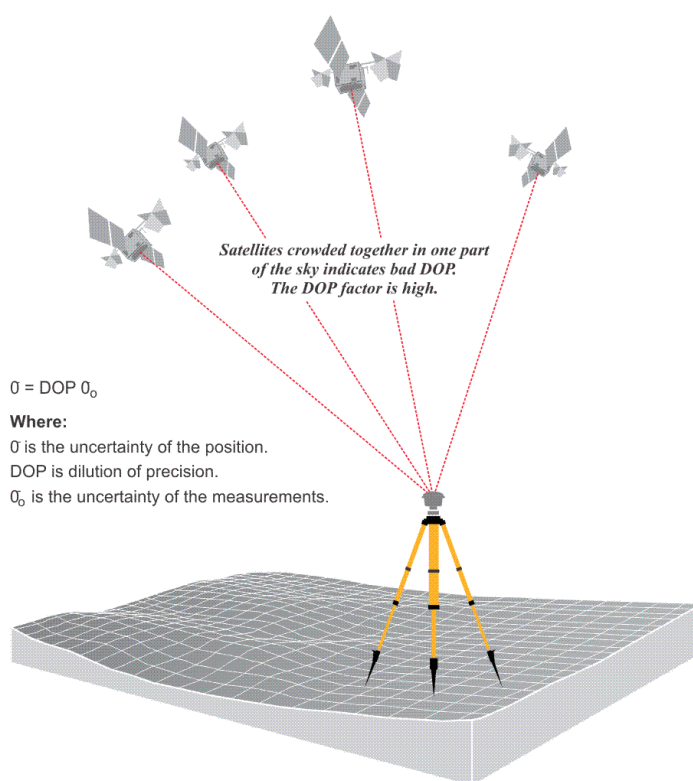
P kod je dio GPS signala koji se odašilje na obje frekvencije i rezerviran je za vojnu razinu upotrebe. Kao i C/A PRN kod, sastoji se od niza nula i jedinica i šalje se brzinom 1023 bit/s, ali je znatno dulji. Potrebno je ukupno 37 tjedana kako bi se sekvencijalno poslao cjelokupni P kod. Za razliku od C/A koda, gdje svaki satelit ima svoj jedinstveni 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, prijemnik razlikuje jedan satelit od drugoga. 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, prijemnik 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 pozicije entiteta.

2.3 Pogreške određivanja položaja i vrste

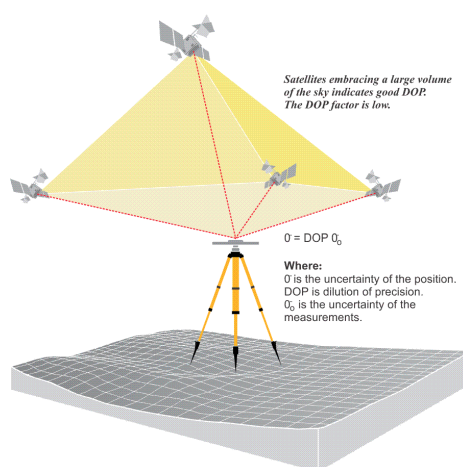
Pogreške određivanja položaja se grubo dijele na dvije vrste: (1) pogreške nastale usljed konstrukcije ulaza algoritma i (2) usljed primjenjivosti algoritma za određivanje položaja na mjerenim psudo-udaljenostima. Dakle, postoje dva izvora: ulazni podaci algoritma (tip 1) i algoritam (tip 2). Izvori pogreške tipa 1 su najčešće pogreške pri određivanju pseudo-udaljenosti ili raspoređenost satelita oko Zemlje (Slike 2.3, 2.4 i 2.5). Nepovoljan položaj promatranih satelita može rezultirati skoro pa zavisnim jednadžbama u 2.4.



Slika 2.3: Razlike u razmještanju satelita



Slika 2.4: Loš razmještaj satelita



Slika 2.5: Dobar razmještaj satelita

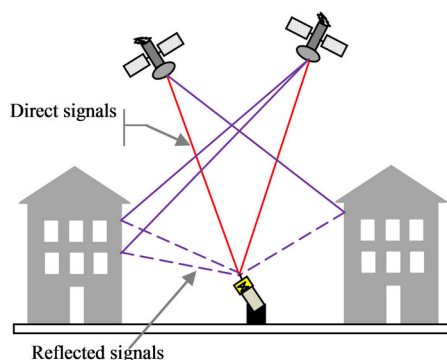
Detaljnija podjela pogrešaka tipa 1 nastalih pri određivanju pseudo-udaljenosti i na što koja utječe dana je sljedećom tablicom.

Tablica 2.1: Pogreške određivanja pseudo-udaljenosti

| izvor | utjecaj |
|------------------------|---|
| satelit | pogreške orbite pogreška sata satelita |
| rasprostiranje signala | troposferska refrakcija ionosferska refrakcija |
| prijemnik | pogreške antene pogreška sata |

Utjecaj sistemskih pogrešaka otklanja se modeliranjem ili kombinacijom opažanja. Korištenjem više prijemnika, otkanjaju se pogreške specifične za satelite. Pogreške specifične za prijemnike otkanja korištenje viška satelita. Utjecajem troposfere se najčešće otklanja modeliranjem, a ionosfere korištenjem dva signala različitih frekvencija.

Uz sistemske pogreške, postoje još i slučajne pogreške nastale zbog trenutnog mjerenja i slučajnog dijela višestruke refleksije signala (multipath) nastalog interferencijom direktnog i reflektiranog signala (Slika 2.6).



Slika 2.6: Višestruka refleksija signala

U ovom radu, sistemske i slučajne pogreške otklonjene su koristeći RTK-LIB. Prije konstrukcije ulaza algoritma za određivanja položaja, smatramo da je pogreška s izvorom u pseudo-udaljenostima maksimalno reducirana. Dakle, pretpostavljamo kako ona više nije značajna neuzimajući ju u obzir u daljnjem postupku izračuna položaja.

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

One se ne modeliraju algoritmima procjene položaja (Poglavlje 3), već prilikom dizajna izvedbe odabranog algoritma (Poglavlje 4)

U poglavlju 4, biti će obrađena analiza pogreške tipa 2 za odabrani algoritam.

2.4 Navigation Message

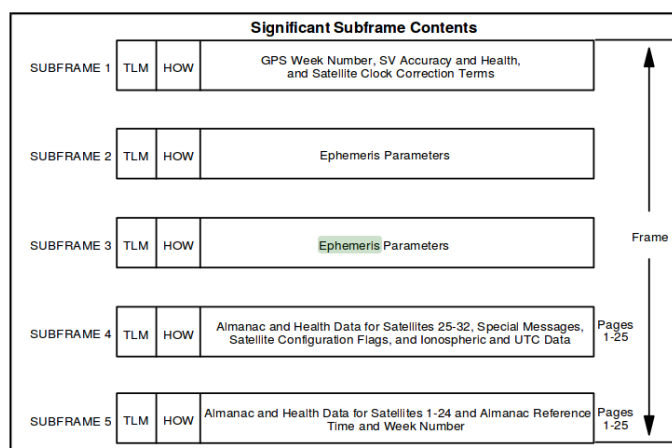
Svaki satelit, uz C/A PRN i P kod, odašilje i dodatne podatke potrebne za ispravo pozicioniranje prijemnika. Odašilje ih u obliku *Navigacijske poruke* koja se šalje zajedno s generiranim C/A PRN kodovima (Slika 2.1).

Navigacijska poruka se sastoji od 25 okvira [6]. Jedan okvir se sastoji od 5 podokvira i svaki sadržava vrijeme slanja sljedećeg okvira (Slika [7]). Za slanje cjelokupnog podokvira potrebno je 6 sekundi, 6 cjelokupnih C/A PRN kodova. Prijemnik je u mogućnosti računati pseudo-udaljenost za novu poziciju satelita svakih 6 sekundi. Za slanje cjelokupne NM, potrebno je 12.5 minuta. U nastavku termin poruka koristi se misleći na podprozor.

Prozor sadrži:

1. GPS vremena odašiljanja,
2. signal prijenosa s P na C/A kod (potpoglavlja 2.2 i 2.1),
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. ionosferski model korekcije za koji se smatra da je potrebno koristiti.

Definicija 2.4.1 (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. Drugi nazivi za taj vremenski standard su ZULU vrijeme i Greenwich Mean Time (GMT).*



Slika 2.7: Pregled strukture prozora navigacijske poruke[7]

Pojedini dijelovi navigacijske poruke pomažu pri otklanjanju pogrešaka tipa 1 (Potpoglavlje 2.3), određivanju pseudo-udaljenosti i trenutnoj poziciji satelita. Naime, iz podataka o orbitalnoj putanji satelita moguće je za odabrani trenutak izračunati poziciju (koordinate) satelita u orbitalnom koordinatnom sustavu pa i svakom drugom.

Za razumjevanje ovoga rada, dovoljno je razumjeti sljedeće. Prijemnik svakih 6 sekundi ima dovoljno podataka da odredi novu pseudo-udaljenost do istog satelita sve dok on ne prestane biti dostupan.

Definicija 2.4.2 (Dostupnost satelita S prijemniku T). *Za satelit S kažemo da je dostupan prijemniku T u trenutku t ako je u sljedećih 6 sekundi u mogućnosti izračunati pseudo-udaljenost do satelita S i konstruirati sljedeću jednadžbu:*

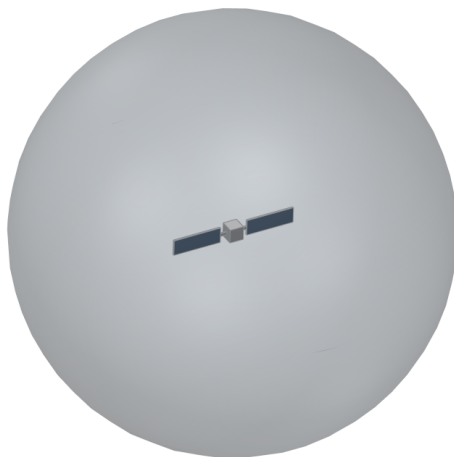
$$d_s = \sqrt{(x - x_s)^2 + (y - y_s)^2 + (z - z_s)^2} \quad (2.1)$$

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

2.5 Proces određivanja položaja

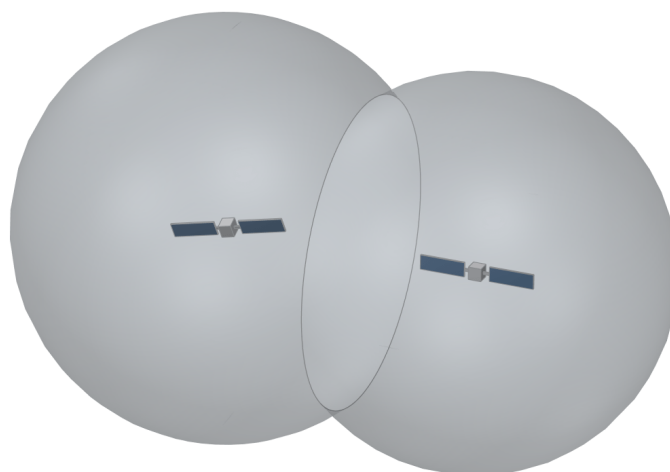
U pravilu, u svakom trenutku, prijemnik ima više dostupnih satelita od kojih dobiva poruke. Za određivanje položaja prijemnika u granicama dopuštene točnosti, zahtjevaju se barem 4 dostupna satelita.

Kako bi prijemnik odredio svoju poziciju računa tri nepoznanice: geografsku širinu, duljinu i nadmorsku visinu. Neka je k broj vidljivih satelita od prijemnika T . Prijemnik T promatrajući poruke dobivene od samo jednog satelita, u vremenu t , izračunava samo jednu pseudo-udaljenost i može konstruirati samo jednu jednadžbu 2.1 koja mu omogućava odrediti sferu oko promatranog satelita na kojoj bi se mogao nalaziti (Slika 2.8).



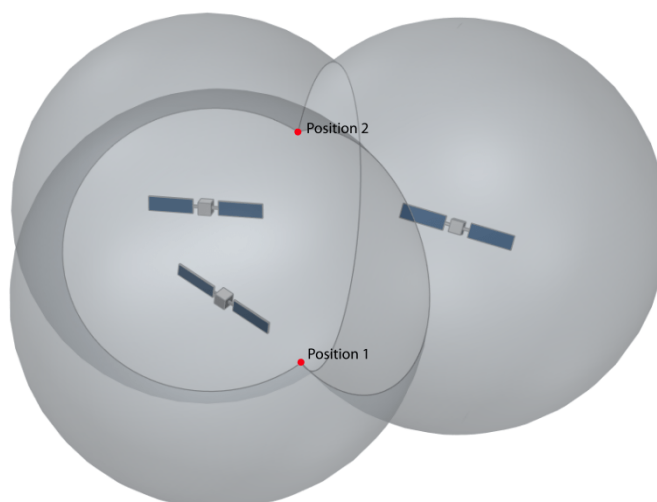
Slika 2.8: Sfera oko promatranog satelita na kojoj bi se prijemnik mogao nalaziti [2]

Uključujući u izračun pridobivene pseudo-udaljenosti od još jednog satelita dobivamo situaciju prikazanu na Slici 2.9.



Slika 2.9: Sfere oko 2 promatrana satelita, presjek je kružnica na kojoj bi se prijemnik mogao nalaziti. [2]

Uključujući u izračun još jedan satelit, dobivamo situaciju prikazanu na Slici 2.10.



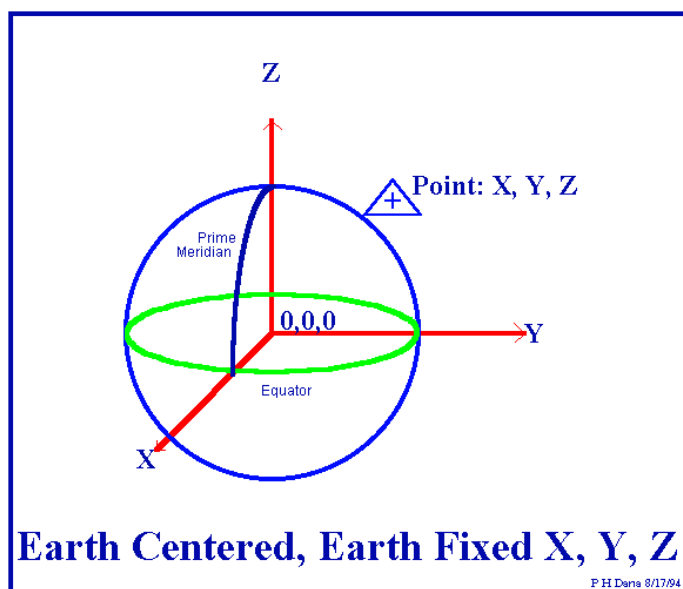
Slika 2.10: Sfere oko 3 promatrana satelita, presjek su 2 točke na kojoj bi se prijemnik mogao nalaziti. [2]

Presjek 3 promatrane sfere su 2 točke na kojoj bi se prijemnik mogao nalaziti. Jedna točka se nalazi daleko u svemiru, dok je druga točka točka kandidat pozicije prijemnika.

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

$$\begin{aligned} d_1 &= \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2} \\ d_2 &= \sqrt{(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2} \\ d_3 &= \sqrt{(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2} \end{aligned} \quad (2.2)$$

gdje su 1, 2 i 3, 3 različita satelita, a (x_i, y_i, z_i) pripadajuće koordinate položaja satelita u (ECEF XYZ) koordinatnom sustavu. ECEF XYZ koordinatni sustav je prikazan na Slici 2.11. Ishodište (ECEF XYZ) koordinatnog sustava je središte zemlje.



Slika 2.11: Earth-Centered, Earth-Fixed X, Y, Z coordinate system (ECEF XYZ koordinatni sustav) [3]

Svaki prijemnik je sposoban izvesti konverziju iz i u koordinata u ECEF XYZ sustavu u i iz geografskih (geografska širina, duljina i nadmorska visina) [3]. Dakle, prijemniku su potrebna barem 3 dostupna satelita kako bi odredio poziciju. Ali se ipak na stranici 14 se postavlja zahtjev na barem 4.

Primjetimo kako proces određivanja položaja prijemnika indirektno zahtjeva usklađenost satova prijemnika i dostupnih satelita. Satovi svih satelita su međusobno usklađeni usklađenošću s GPS vremenom. Ukoliko odstupanje ipak postoji, biti će zapisano u navigacijskoj poruci pa se može uzeti u obzir prilikom određivanja položaja prijemnika. 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 prijemnika nisu iste preciznosti 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 pseudo-udaljenosti od oko 300 metara. Uključujući u izračin i pogrešku sata prijemnika, pseudo-udaljenost modeliramo jednadžbom:

$$d_i = c \times (t'_i - t_i + d_T)$$

gdje d_T predstavlja spomenutu pogrešku. 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 jednadžbi 2.2. Dakle, sustav jednadžbi 2.2 prelazi u:

$$\begin{aligned} d_1 &= \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2} + c \cdot d_T \\ d_2 &= \sqrt{(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2} + c \cdot d_T \\ d_3 &= \sqrt{(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2} + c \cdot d_T \end{aligned} \quad (2.3)$$

Kako bi za gornji sustav postojala mogućnost pronalaska rješenja, uvodi se zahtjev na još barem jedan dostupni satelit, što je ukupno 4 (Stranica 14). Dobivamo sljedeći sustav jednadžbi u (x, y, z, d_T) :

$$\begin{aligned} d_1 &= \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2} + c \cdot d_T \\ d_2 &= \sqrt{(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2} + c \cdot d_T \\ d_3 &= \sqrt{(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2} + c \cdot d_T \\ d_4 &= \sqrt{(x - x_4)^2 + (y - y_4)^2 + (z - z_4)^2} + c \cdot d_T \end{aligned} \quad (2.4)$$

Upravo opisanom postupkom otklanjamo pogrešku nastalu prilikom određivanja pseudo-udaljenosti s izvorom u pogrešci sata prijemnika. U praksi se može koristiti još veći broj dostupnih satelita što poboljšava preciznost pozicioniranja prijemnika.

Poglavlje 3

Algoritam procjene položaja u domeni navigacijske primjene

Algoritam procjene položaja u domeni navigacijske primjene (APP) smatramo svakim algoritmom koji za sustav jednažbi 2.4 određuje nepoznatu poziciju prijemnika u koordinatama (x, y, z) . Broj jednažbi sustava može biti i veći od 4. Tada govorimo o prezasićenim sustavima. Ovisno o odabiru, APP se može temeljiti na rješavanju sustava nelinearnih jednažbi pronalaženjem rješenja pomoću (1) metode najmanjih kvadrata (Newton-ova metoda), (2) metode zatvorene forme, (3) metode najbližeg susjeda [1].

Općenito, rješava se moduficiran sustav jednažbi 2.4 uz $d = c \cdot d_T$:

$$\begin{aligned}d_1 &= \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2} + d + v_1 \\d_2 &= \sqrt{(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2} + d + v_2 \\d_3 &= \sqrt{(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2} + d + v_3 \\d_4 &= \sqrt{(x - x_4)^2 + (y - y_4)^2 + (z - z_4)^2} + d + v_4\end{aligned}\tag{3.1}$$

u koji uključujemo nepoznati parametar (v_1, v_2, v_3, v_4) , dodatnu pogreška izračuna.

Uz oznake

$$\rho := (d_1, d_2, d_3, d_4)^T \quad (3.2)$$

$$\mathbf{x} := (x, y, z, d_T)^T \quad (3.3)$$

$$\mathbf{s}_i := (x_i, y_i, z_i)^T \quad (3.4)$$

$$\mathbf{h}(\mathbf{x}) := \begin{bmatrix} \|(s_1 - \mathbf{x}_{1:3})\| + x_4 \cdot c \\ \|(s_2 - \mathbf{x}_{1:3})\| + x_4 \cdot c \\ \|(s_3 - \mathbf{x}_{1:3})\| + x_4 \cdot c \\ \|(s_4 - \mathbf{x}_{1:3})\| + x_4 \cdot c \end{bmatrix} \quad (3.5)$$

$$\mathbf{v} := (v_1, v_2, v_3, v_4)^T \quad (3.6)$$

prelazi u

$$\rho = \mathbf{h}(\mathbf{x}) + \mathbf{v} \quad (3.7)$$

3.1 Iterativna metoda najmanjih kvadrata

Primjetimo kako je $\mathbf{h}(\mathbf{x})$ jednako pravom vektoru pseudo-udaljenosti (udaljenosti, ako pogreška sata, $d_T = 0$) između satelita i prijemnika za prave vrijednosti $\mathbf{x}, \bar{\mathbf{x}}$.

Algoritam određivanja položaja se ne bavi pogreškama tipa 2, već samo pogreškama tipa 1 (Stranica 2.3). Također, može se pretpostaviti kako su otklonjene sve pogreške tipa 1 koje imaju izvor u pogreškama izračuna pseudo-udaljenosti (osim pogreški sata prijemnika) (Stranica 2.3). Ostaje samo modelirati pogreške koje imaju za izvor trenutni položaj satelita dostupnih za izračunavanje željenog položaja $\bar{\mathbf{x}}_{(1:3)}$. U tu svrhu modeliramo vektor pogrešaka \mathbf{v} , funkcijom $\mathbf{p}(\mathbf{x})$ koja ovisi o nepoznatom parametru \mathbf{x} . Uz oznaku $\mathbf{y} := \rho$, jednadžba 3.7 prelazi u

$$\mathbf{y} = \rho = \mathbf{h}(\mathbf{x}) + \mathbf{p}(\mathbf{x}) \quad (3.8)$$

Preciznije, član $\mathbf{p}(\mathbf{x})$ modelira pogrešku razlike u procjeni parametra \mathbf{x} od stvarne vrijednosti. Što je aproksimacija potrebnih vrijednosti za izračun rješenja matrične jednadžbe 3.7 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 $\mathbf{p}(\mathbf{x})$.

Ideja metode najmanjih kvadrata je pronalazak $\hat{\mathbf{x}}$ tražeći minimum $\mathbf{p}(\mathbf{x})$, tj.

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \mathbf{p}(\mathbf{x})^T \mathbf{p}(\mathbf{x}) \quad (3.9)$$

Problem opisan jednađbom 3.9 nije linearan pa ne postoji općeniti način pronalaska njegovog rješenja.

U slučaju kada su funkcija koju treba minimizirati i početna vrijednost \mathbf{x}_0 (iterativnog postupka) dovoljno dobre (vidi: dodatak A), rješenja problema 3.9 možemo dobiti iterativnim postupkom. Ideja iterativnog postupka je počevši s \mathbf{x}_0 računati $\mathbf{x}_1, \mathbf{x}_2, \dots$ 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.

Jedan iterativni postupak rješavanja problema 3.9 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 :

$$\mathbf{p}(\mathbf{x}_k + \Delta\mathbf{x}_k) \approx \mathbf{p}(\mathbf{x}_k) + \mathbf{p}'(\mathbf{x}_k) \cdot \Delta\mathbf{x}_k \quad (3.10)$$

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

$$\lim_{k \rightarrow \infty} (\mathbf{p}(\mathbf{x}_k)) = 0$$

jer za pravu vrijednost \mathbf{x} izraz $\mathbf{p}(\mathbf{x}) = 0$ ili poprima svoj minimum ukoliko postoje pogreške točnosti vrijednosti koje se koriste prilikom konstrukcije 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 trežeći minimum funkcije

$$\mathbf{p}(\mathbf{x}_k) + \mathbf{p}'(\mathbf{x}_k) \cdot \Delta\mathbf{x}_k \quad (3.11)$$

u $\Delta\mathbf{x}_k$.

Označimo sada s $J_k := \mathbf{p}'(\mathbf{x}_k) = \mathbf{h}'(\mathbf{x}_k)$. 3.11 prelazi u

$$J_k \Delta\mathbf{x}_k + \mathbf{p}(\mathbf{x}_k) \quad (3.12)$$

čij je minimum dan s (vidi stranicu 31)

$$\Delta\mathbf{x}_k = -(J_k^T J_k)^{-1} J_k^T \mathbf{p}(\mathbf{x}_k) \quad (3.13)$$

Izraz za \mathbf{x}_{k+1} je sljedeći:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (J_k^T J_k)^{-1} J_k^T \mathbf{p}(\mathbf{x}_k) \quad (3.14)$$

Prilikom izvedbe algoritma, potrebno je pametno odrediti početnu vrijednost \mathbf{x}_0 , te kasnije iterirati po formuli 3.14. Ukoliko odaberemo dovoljno dobar \mathbf{x}_0 , dovoljno blizu rješenju i ako je druga derivacija od p u točki $\bar{\mathbf{x}}$ dovoljno mala, niz x_0, x_2, \dots konvergira prema $\bar{\mathbf{x}}$. Izračun J_k za idealan slučaj $d = 0$ se može naći u prilogu A.

Algoritam iterativne metode najmanjih je dan u nastavku.

Algoritam 1: Iterativna metoda najmanjih kvadrata

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

Result: $\hat{\mathbf{x}}$

```

1  $k = 0$  ;
2 while  $\|\mathbf{x}_k - \mathbf{x}_{k-1}\| \geq \delta$  do
3    $J_k = \mathbf{p}'(\mathbf{x}_k)$  ;
4    $\Delta\mathbf{x}_k = -J_k^{-1} \cdot \mathbf{p}(\mathbf{x}_k)$  ;
5    $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k$  ;
6    $k++$ ;
7 end
8  $\hat{\mathbf{x}} = \mathbf{x}_k$ 

```

Prilikom korištenja gornjeg algoritma za određivanje pozicije entiteta, 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 2.4 za pojedine jednadžbe točnije, pametno je dati prednost tim jednadžbama pred ostalima. Važnost pojedine jednadžbe simuliramo pridavanjem težina pojedinoj jednadžbi. 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 3.6), u oznaci $\Sigma := \text{cov}(\mathbf{v})$. Minimizacioni problem 3.9 prelazi u

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \mathbf{p}(\mathbf{x})^T \Sigma^{-1} \mathbf{p}(\mathbf{x}) \quad (3.15)$$

Sada, algoritam 1 prelazi u algoritam 2.

Algoritam 2: Iterativna metoda težinskih najmanjih kvadrata

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

Result: $\hat{\mathbf{x}}$

```

1  $k = 0$  ;
2 while  $\|\mathbf{x}_k - \mathbf{x}_{k-1}\| \geq \delta$  do
3    $J_k = \mathbf{p}'(\mathbf{x}_k)$  ;
4    $\Delta\mathbf{x}_k = -(\Sigma^{-\frac{1}{2}} J_k)^{-1} (\Sigma^{-\frac{1}{2}} (\mathbf{p}(\mathbf{x}_k)))$  ;
5    $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k$  ;
6    $k++$ ;
7 end
8  $\hat{\mathbf{x}} = \mathbf{x}_k$ 

```

Procjenitelj za $\bar{\mathbf{x}}$ dobiven težinskom metodom najmanjih kvadrata, jednakost 3.15, ima najmanju varijancu među svim procjeniteljima za $\bar{\mathbf{x}}$. Ukoliko je vektor pogrešaka \mathbf{v} normalno distribuiran, procjenitelj 3.15 postaje procjenitelj metode najbližeg susjeda (Podpoglavlje 3.3 i MLE procjenitelj).

Prilikom korištenja iterativne metode najmanjih kvadrata potrebno je modelirati distribuciju vektora pogrešaka, točnije kovarijanconu matricu Σ . Također, potrebno je pripaziti na velike pogreške u određivanju vrijednosti pomoći kojih se gradi sustav jednačbi 2.4 i netipične vrijednosti ("outlinere") 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 2.4.

Analize pogreške metode najmanjih kvadrata

Uz oznake kao do sada, neka $\bar{\mathbf{y}}$ predstavlja prave udaljenosti između satelita i promatranog entiteta (prijemnika) i $\hat{\mathbf{y}}$ izračunate pseudo-udaljenosti. Vrijedi $\hat{\mathbf{y}} = \bar{\mathbf{y}} + \Delta\mathbf{y}$. Promatramo idealan slučaj za metodu iterativnih najmanjih kvadrata (Algoritam 1), $\delta = 0$. Neka je $\hat{\mathbf{x}} = \bar{\mathbf{x}} + \Delta\mathbf{x}$ rješenje metode najmanjih kvadrata konvergirala, 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 jednačbu 3.14 dobivamo

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

Matica J_k predstavlja funkciju koja ovisi o parametru \mathbf{x} i nije konstantna. Kako se pretpostavlja da je $\Delta\mathbf{x}$ blizu nule, opravdano je promatrati $J := J_k$ konstantnom u susjedstvu od $\bar{\mathbf{x}}$ radijusa $\Delta\mathbf{x}$. Sada se \mathbf{h} u okolini točke $\bar{\mathbf{x}}$ može linearizirati na sljedeći način:

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

Dobivamo

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

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

$$\Delta\mathbf{x} \sim N(0, (J^T J)^{-1} J^T \Sigma J (J^T J)^{-1}) \quad (3.16)$$

Također, uz $\Sigma = \sigma^2 I$, $\Delta\mathbf{x} \sim N(0, \sigma^2 (J^T J)^{-1})$. U kontekstu satelitske navigacije, $(J^T J)^{-1}$ se naziva DOP matrica (engl. Dilution of Precision). Iz DOP matrice moguće je izvesti različite mjere kvalitete "zvijezdada" satelita u danom trenutku za danu poziciju.

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

Opširnije o mjerama kvalitete "zvijezdada" moguće je naći u dodatku C ovoga rada.

Dakle, uz neke pretpostavke, iz Jakobijeve matrice funkcije \mathbf{h} , J , može se saznati mnogo o kvaliteti određivanja položaja za sustav jednadžbi 2.4, veličini pogreške određivanja s izvorom u kvaliteti "zvijezdada". Izračuni gornjih mjera su točni onoliko koliko su pretpostavke o jednakosti varijance za $\Delta\mathbf{y}$ i $\Delta\mathbf{x}$ istinite.

Primjenu metode najmanjih kvadrata moguće je pronaći na stranici 31.

3.2 Metoda zatvorene forme

Metoda zatvorene forme pronalazi direktno rješenje sustava 2.4. Za razliku od iterativnih metoda, metode zatvorene forme ne zahtijevaju postavljanje početnog rješenja 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 teško 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 vizualizaciju.

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

$$\begin{aligned} y_1 &= \|\mathbf{s}_1 - \mathbf{x}_{1:3}\| + x_4 \\ y_2 &= \|\mathbf{s}_2 - \mathbf{x}_{1:3}\| + x_4 \\ y_3 &= \|\mathbf{s}_3 - \mathbf{x}_{1:3}\| + x_4 \\ &\vdots \\ y_n &= \|\mathbf{s}_n - \mathbf{x}_{1:3}\| + x_4 \end{aligned} \quad (3.17)$$

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

$$\begin{aligned} \mathbf{x}_{1:3} &= \mathbf{d}\lambda + \mathbf{e} \\ x_4 &= f\lambda + g \end{aligned} \quad (3.18)$$

gdje λ dobivamo rješavanjem sljedeće jednadžbe

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

$\hat{\mathbf{x}}$ je rješenje sustava 3.17 ako i samo ako je rješenje zatvorene forme 3.18.

Parametre \mathbf{d} , \mathbf{e} , f i g zatvorene forme 3.18 dobivamo iz sustava 3.17 sljedećim nizom pretvorbi:

$$\begin{aligned} (y_1 - x_4)^2 &= \|\mathbf{s}_1 - \mathbf{x}_{1:3}\|^2 \\ (y_2 - x_4)^2 &= \|\mathbf{s}_2 - \mathbf{x}_{1:3}\|^2 \\ (y_3 - x_4)^2 &= \|\mathbf{s}_3 - \mathbf{x}_{1:3}\|^2 \\ &\vdots \\ (y_n - x_4)^2 &= \|\mathbf{s}_n - \mathbf{x}_{1:3}\|^2 \end{aligned} \quad (3.19)$$

$$\begin{aligned}
y_1^2 - 2y_1x_4 + x_4^2 &= \|\mathbf{s}_1\|^2 - 2\mathbf{s}_1^T \mathbf{x}_{1:3} + \|\mathbf{x}_{1:3}\|^2 \\
y_2^2 - 2y_2x_4 + x_4^2 &= \|\mathbf{s}_2\|^2 - 2\mathbf{s}_2^T \mathbf{x}_{1:3} + \|\mathbf{x}_{1:3}\|^2 \\
y_3^2 - 2y_3x_4 + x_4^2 &= \|\mathbf{s}_3\|^2 - 2\mathbf{s}_3^T \mathbf{x}_{1:3} + \|\mathbf{x}_{1:3}\|^2 \\
&\vdots \\
y_n^2 - 2y_nx_4 + x_4^2 &= \|\mathbf{s}_n\|^2 - 2\mathbf{s}_n^T \mathbf{x}_{1:3} + \|\mathbf{x}_{1:3}\|^2
\end{aligned} \tag{3.20}$$

Uz

$$\lambda := \|\mathbf{x}_{1:3}\|^2 - x_4^2$$

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

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

koje čine sustav

$$\begin{bmatrix} 2\mathbf{s}_1^T & -2y_1 \\ 2\mathbf{s}_2^T & -2y_2 \\ 2\mathbf{s}_3^T & -2y_3 \\ \vdots & \\ 2\mathbf{s}_n^T & -2y_n \end{bmatrix} \begin{bmatrix} \mathbf{x}_{1:3} \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \lambda + \begin{bmatrix} \|\mathbf{s}_1\|^2 & -y_1^2 \\ \|\mathbf{s}_2\|^2 & -y_2^2 \\ \|\mathbf{s}_3\|^2 & -y_3^2 \\ \vdots & \\ \|\mathbf{s}_n\|^2 & -y_n^2 \end{bmatrix}. \tag{3.21}$$

Naposljetku, za

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

dobivamo 3.18.

3.3 Metoda najbližeg susjeda (maksimalne vjerodostojnosti)

Metode najbližeg susjeda opisuju pogrešku mjerenja uvjetnom vjerojatnošću, $\mathbf{p}(y|\mathbf{x})$. $\mathbf{p}(y|\mathbf{x})$ je vjerojatnost da je psudoudaljenost y izmjerena na položaju s koordinatama $\mathbf{x}_{1:3}$ s pogreškom u izvoru sata prijemnika jednako \mathbf{x}_4 . 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 3.7, lako se dobiva ekvivalentan problem određivanja položaja maksimalne vjerodostojnosti. Budući da vrijedi

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

i \mathbf{v} je poznate distribucije, dobivamo:

$$\mathbf{p}(y|\mathbf{x}_{1:3}) = \mathbf{p}_{\mathbf{v}}(\rho - \mathbf{h}(\mathbf{x}_{1:3})) \quad (3.22)$$

Ukoliko je problem određivanja položaja zadan s 3.22, $\hat{\mathbf{x}}$ pronalazimo pomoću procjenitelja maksimalne vjerodostojnosti za \mathbf{x} , tj. $\hat{\mathbf{x}}$ je takav da vrijedi

$$L(\hat{\mathbf{x}}|y) := \max_{\tilde{\mathbf{x}}} (L(\tilde{\mathbf{x}}|y)) \quad (3.23)$$

gdje $\tilde{\mathbf{x}}$ predstavljaju sve dozvoljene koordinate položaja entiteta na Zemlji i u zraku.

Za poznata mjerenja psudoudaljenosti, $\hat{\mathbf{x}}$ se može pronaći metodom nelinearne optimizacije.

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 $\Sigma^{-1} = \text{cov}(\mathbf{v})^{-1}$. Naime, za

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

imamo

$$\mathbf{p}(y|\mathbf{x}) = \mathbf{p}_{\mathbf{v}}(\rho - \mathbf{h}(\mathbf{x})) = C \exp \left(-\frac{1}{2} (\rho - \mathbf{h}(\mathbf{x}))^T \Sigma^{-1} (\rho - \mathbf{h}(\mathbf{x})) \right) \quad (3.24)$$

Budući da je Σ pozitivno definitna matrica, argument eksponencijalne funkcije gornjeg izraza je uvijek negativan. Dakle, problem maksimizacije funkcije 3.24 jednak je minimizaciji izraza $(\rho - \mathbf{h}(\mathbf{x}))^T \Sigma^{-1} (\rho - \mathbf{h}(\mathbf{x})) = \mathbf{v}^T \Sigma^{-1} \mathbf{v} = \mathbf{p}^T(\mathbf{x}) \Sigma^{-1} \mathbf{p}(\mathbf{x})$.

Dakle, $\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} (\mathbf{p}^T(\mathbf{x}) \Sigma^{-1} \mathbf{p}(\mathbf{x}))$ što odgovara izrazu 3.15 uz matricu težina jednaku $\text{cov}(\mathbf{v})^{-1}$.

Poglavlje 4

Dizajn i izvedba algoritma, procjena točnosti

Nakon proučavanja literature, odlučili smo se za izvedbu temeljnog algoritma za određivanje položaja: algoritma težinskih najmanjih kvadrata (vidi stranicu 22, algoritam 2).

Za dobivene psudo udaljenosti (vidi 4.2), i položaj satelita u *ECEF XYZ* koordinatnom sustavu u *RINEX* formatu, izračunavamo položaj prijemnika i pogrešku sata prijemnika.

4.1 Programski jezik R

4.2 Zahtjevi algoritma

Za pokretanje procesa određivanja položaja, potrebno je prvo prikupiti podatke u *RINEX* obliku. Podatci su prikupljeni koristeći programski određen GPS prijemnik izveden na vlastitom računalu.

RINEX

-objašnjenje pojma + primjeri. + otkuda nam.

Programski određen GPS prijemnik

+zašto i kako smo ga koristili. i dobili podatke. (User manual?)

4.3 Izvedba

Direktna izvedba algoritma sa stranice 22, dobivamo izvedbu koja nije općepromjenjiva. Naime, matrica sustava ima veliku uvjetovanost pa sustav postaje numerički nestabilan.

Naivan pristup

-neki od onih

Izvedba

R kod

Odabran pristup

Budući da je sustav koji opisuje problem određivanja položaja nelinearan, možemo ga prvo **linearizirati**, a tek zatim primjeniti metodu najmanjih kvadrata [5]. Rješenja lineariziranog i nelineariziranog sustava nisu u potpunosti jednaka, ali su u ovom slučaju dovoljno bliska jer su jednadžbe sustava skoro pa linearne [9].

Prvo se promatra se sustav 2.4:

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

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

Označimo s $f_i(\mathbf{x}) = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} + c \cdot d_T$.

Linearizacijom jednadžbi sustava na isti način kao $\mathbf{p}(\mathbf{x})$ sa stranice 21 (za Gauss-Newtonovu metodu), dobivamo:

$$f_i(\mathbf{x} + \Delta\mathbf{x}) = f_i(\mathbf{x}) + \frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial y} \Delta y + \frac{\partial f}{\partial z} \Delta z + \frac{\partial f}{\partial d_T} \Delta d_T$$

Koristeći iterativnu metodu, $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k$. Budući da se (d_1, d_2, d_3, d_4) ne mijenjaju kroz iteracije, dobivamo izraz:

$$d_i = f_i(\mathbf{x}_{k+1}) = f_i(\mathbf{x}_k + \Delta\mathbf{x}_k) = f_i(\mathbf{x}_k) + \frac{\partial f}{\partial x} \Delta x_k + \frac{\partial f}{\partial y} \Delta y_k + \frac{\partial f}{\partial z} \Delta z_k + \frac{\partial f}{\partial d_T} \Delta(d_T)_k$$

, tj.

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

Uz

$$\mathbf{A} := \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} & \frac{\partial f_1}{\partial d_T} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z} & \frac{\partial f_2}{\partial d_T} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial z} & \frac{\partial f_3}{\partial d_T} \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial z} & \frac{\partial f_4}{\partial d_T} \end{bmatrix} \quad (4.2)$$

$$= \begin{bmatrix} \frac{(x-x_1)}{\sqrt{(x-x_1)^2 + (y-y_1)^2 + (z-z_1)^2}} & \frac{(y-y_1)}{\sqrt{(x-x_1)^2 + (y-y_1)^2 + (z-z_1)^2}} & \frac{(z-z_1)}{\sqrt{(x-x_1)^2 + (y-y_1)^2 + (z-z_1)^2}} & c \\ \frac{(x-x_2)}{\sqrt{(x-x_2)^2 + (y-y_2)^2 + (z-z_2)^2}} & \frac{(y-y_2)}{\sqrt{(x-x_2)^2 + (y-y_2)^2 + (z-z_2)^2}} & \frac{(z-z_2)}{\sqrt{(x-x_2)^2 + (y-y_2)^2 + (z-z_2)^2}} & c \\ \frac{(x-x_3)}{\sqrt{(x-x_3)^2 + (y-y_3)^2 + (z-z_3)^2}} & \frac{(y-y_3)}{\sqrt{(x-x_3)^2 + (y-y_3)^2 + (z-z_3)^2}} & \frac{(z-z_3)}{\sqrt{(x-x_3)^2 + (y-y_3)^2 + (z-z_3)^2}} & c \\ \frac{(x-x_4)}{\sqrt{(x-x_4)^2 + (y-y_4)^2 + (z-z_4)^2}} & \frac{(y-y_4)}{\sqrt{(x-x_4)^2 + (y-y_4)^2 + (z-z_4)^2}} & \frac{(z-z_4)}{\sqrt{(x-x_4)^2 + (y-y_4)^2 + (z-z_4)^2}} & c \end{bmatrix} \quad (4.3)$$

$$\mathbf{x} := \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta d_T \end{bmatrix} \quad (4.4)$$

$$\mathbf{b} := \begin{bmatrix} d_1 - \sqrt{(x_k - x_1)^2 + (y_k - y_1)^2 + (z_k - z_1)^2} - c \cdot (d_T)_k \\ d_2 - \sqrt{(x_k - x_2)^2 + (y_k - y_2)^2 + (z_k - z_2)^2} - c \cdot (d_T)_k \\ d_3 - \sqrt{(x_k - x_3)^2 + (y_k - y_3)^2 + (z_k - z_3)^2} - c \cdot (d_T)_k \\ d_4 - \sqrt{(x_k - x_4)^2 + (y_k - y_4)^2 + (z_k - z_4)^2} - c \cdot (d_T)_k \end{bmatrix} \quad (4.5)$$

dobivamo sustav:

$$\mathbf{Ax} = \mathbf{b} \quad (4.6)$$

koji rješavamo metodom iterativnih najmanjih kvadrata. Ideja je da, budući da zbog pogreške u mjerenjima ili linearizacije, što je slučaj u ovome radu, sustav 4.6 nema uvijek rješenje, tj. $\mathbf{Ax} - \mathbf{b} \neq 0, \forall \mathbf{x} \in \mathbb{R}^m$, ne tražiti rješenje sustava, već \mathbf{x}

koji minimizira izraz $\|\mathbf{Ax} - \mathbf{b}\|_2$. Prema sljedećem teoremu, dovoljno je promatrati sustav

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

.

Teorem 4.3.1. *Skup svih rješenja problema $\min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\|_2$ označimo s*

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

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

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

koju obično nazivamo sustav normalnih jednadžbi i pišemo u obliku

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

.

Dokaz teorema se može naći u [9], stranica 46.

Dakle, rješenje minimizacijskog problema najmanjih kvadrata sustava 4.6 je jednako rješenju sustava

$$A^T \mathbf{Ax} = A^T \mathbf{b} \tag{4.7}$$

koji se naziva **sustav normalnih jednadžbi**.

Zanimljiva činjenica je da ukoliko znamo jedno rješenje sustava 4.7, lako je pronaći i sva ostala, ukoliko ona postoje.

Neka je $\mathbf{Ax} - \mathbf{b} = r$ i $\hat{\mathbf{x}} \in \mathbb{R}^m$ proizvoljan.

Neka je

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

Tada je $\hat{\mathbf{x}} \in S$ ako i samo ako $\hat{r} = r$, $\mathbf{A}(\hat{\mathbf{x}} - \mathbf{x})$, odnosno $\hat{\mathbf{x}} - \mathbf{x} \in \mathcal{N}(A)$.

Ukoliko vrijedi nešto od sljedećega:

- A ima puni stupčani rang,

- stupci matrice A su linearno nezavisni,
- $A^T A$ je pozitivno definitna,

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

Vrijede i sljedeće tvrdnje.

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

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

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

Nakon formalizacije problema, potrebno je jednostavan način za izračunavanje rješenja sustava 4.7. Jasno je da se matrica $A^T A$ ne invertira, nego se rješava sustav 4.7. Sustav možemo riješiti tako što koristimo faktORIZACIJU Choleskoga matrice $A^T A$. Tako pronađeno rješenje nije naročito točno[9],stranica 60.

Numerička metoda koja dovodi do poboljšanja točnosti rješenja

Budući da će korištena matrica A imati puni stupčani rang, rješenje problema minimizacije možemo zapisati kao:

$$\|A\mathbf{x} - \mathbf{b}\|_2 = \|\mathbf{Q}^T(A\mathbf{x} - \mathbf{b})\|_2 \quad (4.8)$$

$$= \|\mathbf{Q}^T A \mathbf{x} - \mathbf{Q}^T \mathbf{b}\|_2 \quad (4.9)$$

gdje je \mathbf{Q} proizvoljna ortogonalna matrica.

\mathbf{Q} može bit proizvoljna pa možemo odabrati \mathbf{Q} takvu da je lagano računati \mathbf{x} . Korištenjem QR faktORIZACIJE, $\mathbf{A} = \mathbf{Q}\mathbf{R}$ i $\mathbf{Q}^T \mathbf{A} = \mathbf{R}$ gdje je \mathbf{R} gornjetrokutasta matrica. Na ovaj način dobiveno rješenje se pokazuje znatno točnije.

Sada rješavamo sustav

$$\mathbf{R}\mathbf{x} = \mathbf{Q}^T \mathbf{b}. \quad (4.10)$$

tj. tražimo

$$\begin{aligned}
 \mathbf{x} &= \mathbf{R}^{-1} \mathbf{Q}^T \mathbf{b} \\
 &= \mathbf{R}^{-1} \mathbf{R}^{-T} \mathbf{R}^T \mathbf{Q}^T \mathbf{b} \\
 &= (\mathbf{R}^T \mathbf{R})^{-1} (\mathbf{Q} \mathbf{R})^T \mathbf{b} \\
 &= (\mathbf{R}^T \mathbf{I} \mathbf{R})^{-1} (\mathbf{Q} \mathbf{R})^T \mathbf{b} \\
 &= (\mathbf{R}^T \mathbf{Q}^T \mathbf{Q} \mathbf{R})^{-1} (\mathbf{A})^T \mathbf{b} \\
 &= (\mathbf{A}^T \mathbf{A})^{-1} (\mathbf{A})^T \mathbf{b}
 \end{aligned}$$

Gledajući gornju jednakost odozdo prema gore, i sustav nominalnih jednadžbi se svoji na trokutast sustav 4.10 koji je dalje pogodan za rješavanje problema predstavljenog ovim radom. Kada to ne bi bio slučaj, moglo bi se dogoditi da sustav nema rješenja. Naime, prije primjene metode najmanjih kvadrata za pronalazak rješenja, linearizira se početni sustav jednadžbi, uvode se pogreške.

Izvedba

Sustav 4.10 dalje rješavamo iterativnim postupkom.

Programski kod je dan u nastavku:

```

library("MASS", "matrixcalc")
#pseudo-udaljenosti
c <- 2.99792458E+08 # brzina svjetlosti [m/s], po GPS standardu
R = read.csv('pseudoranges5a.txt', header = FALSE);
R <- as.matrix(R[,1])
#ucitaj koordinate satelita
S = read.csv('satellites5.txt', header = FALSE)
S <- as.matrix(S)
x_0 = c(1,1,1,1) #[x,y,z,d.T] d_t se kasnije mnozi sa c da bi se oduzeo od
delt = c(3,3,3,3)
nRows = dim(S)[1]
nCols = dim(S)[2]+1

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

unutar = append(S,rep(c,nRows))
RS = matrix(unutar,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){
start.time <- Sys.time()

b = R
while(iter < niter){
  x_iter = c(x_0[1:3],0) s
  AA = t(apply(RS, 1, function(x) (x_iter - x)))
  D = sqrt(AA**2%%c(1,1,1,0))
  DD = matrix(append(rep(D,3),rep(1,nRows)),nRows,nCols)

  A_iter = AA/DD
  delt <- qr.coef(qr(A_iter), b)
  #rjesava sustav Ax=b koristeći QR faktORIZACIJU
  x_0 = x_0 + delt #(x,y,z,dT)

  b = R - D - c*x_0[nCols]
  if(iter%%10 == 0){
    cat(c(iter, delt[1:3]),
        '\r', file="razmakIteracija.txt", append=TRUE)

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

    print(A_iter)
  }
  iter = iter +1
}

```

Meroda ipak nažalost ne konvergira. Pokušajmo na sljedeći način.

Poboljšan pristup: linearizacija dijela

Ideja je linearizirati samo nelinearni dio jednadžbi sustava 2.4, odnosno $\sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} =: f(x, y, z)$.

$$f(x + \Delta x, y + \Delta y, z + \Delta z) = f_i(x, y, z) + \frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial y} \Delta y + \frac{\partial f}{\partial z} \Delta z$$

Analogno kao zu prošlom podpoglavlju, dobivamo:

$$\begin{aligned} d_i = f_i(\mathbf{x}_{k+1}) + c(d_T)_{k+1} &= f_i(\mathbf{x}_k + \Delta \mathbf{x}_k) + c(d_T)_k \\ &= f_i(\mathbf{x}_k) + \frac{\partial f}{\partial x} \Delta x_k + \frac{\partial f}{\partial y} \Delta y_k + \frac{\partial f}{\partial z} \Delta z_k + \frac{\partial f}{\partial d_T} \Delta(d_T)_k + c(d_T)_k \end{aligned}$$

, tj.

$$\begin{aligned} d_i - \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2 + (z_k - z_i)^2} + c((d_T)_{k+1} - (d_T)_k) = \\ \frac{\partial f}{\partial x} \Delta x_k + \frac{\partial f}{\partial y} \Delta y_k + \frac{\partial f}{\partial z} \Delta z_k + \frac{\partial f}{\partial d_T} \Delta(d_T)_k = \begin{bmatrix} \frac{\partial f_i}{\partial x} & \frac{\partial f_i}{\partial y} & \frac{\partial f_i}{\partial z} & \frac{\partial f_i}{\partial d_T} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta d_T \end{bmatrix} \end{aligned}$$

U_z

$$\mathbf{A} := \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} & c \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z} & c \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial z} & c \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial z} & c \end{bmatrix} \quad (4.11)$$

$$= \begin{bmatrix} \frac{(x-x_1)}{\sqrt{(x-x_1)^2+(y-y_1)^2+(z-z_1)^2}} & \frac{(y-y_1)}{\sqrt{(x-x_1)^2+(y-y_1)^2+(z-z_1)^2}} & \frac{(z-z_1)}{\sqrt{(x-x_1)^2+(y-y_1)^2+(z-z_1)^2}} & c \\ \frac{(x-x_2)}{\sqrt{(x-x_2)^2+(y-y_2)^2+(z-z_2)^2}} & \frac{(y-y_2)}{\sqrt{(x-x_2)^2+(y-y_2)^2+(z-z_2)^2}} & \frac{(z-z_2)}{\sqrt{(x-x_2)^2+(y-y_2)^2+(z-z_2)^2}} & c \\ \frac{(x-x_3)}{\sqrt{(x-x_3)^2+(y-y_3)^2+(z-z_3)^2}} & \frac{(y-y_3)}{\sqrt{(x-x_3)^2+(y-y_3)^2+(z-z_3)^2}} & \frac{(z-z_3)}{\sqrt{(x-x_3)^2+(y-y_3)^2+(z-z_3)^2}} & c \\ \frac{(x-x_4)}{\sqrt{(x-x_4)^2+(y-y_4)^2+(z-z_4)^2}} & \frac{(y-y_4)}{\sqrt{(x-x_4)^2+(y-y_4)^2+(z-z_4)^2}} & \frac{(z-z_4)}{\sqrt{(x-x_4)^2+(y-y_4)^2+(z-z_4)^2}} & c \end{bmatrix} \quad (4.12)$$

$$\mathbf{x} := \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ d_T \end{bmatrix} \quad (4.13)$$

$$\mathbf{b} := \begin{bmatrix} d_1 - \sqrt{(x_k - x_1)^2 + (y_k - y_1)^2 + (z_k - z_1)^2} \\ d_2 - \sqrt{(x_k - x_2)^2 + (y_k - y_2)^2 + (z_k - z_2)^2} \\ d_3 - \sqrt{(x_k - x_3)^2 + (y_k - y_3)^2 + (z_k - z_3)^2} \\ d_4 - \sqrt{(x_k - x_4)^2 + (y_k - y_4)^2 + (z_k - z_4)^2} \end{bmatrix} \quad (4.14)$$

Budući da $(d_T)_{k+1}$ pripada budućem koraku, njega ne možemo modelirati pa se pretpostavlja da će $c((d_T)_{k+1} - (d_T)_k)$ biti modelirano korištenjem normalnih jednadžbi. Dakle, dobivamo sustav:

$$\mathbf{Ax} = \mathbf{b} \quad (4.15)$$

Modifikacija ne dovodi do poboljšanja. Naime, dobiva se gotovo identičan sustav jednadžbi s istim problemima. Najveći je što matrica \mathbf{A} (ili vektor \mathbf{x} , ovisno o definiciji) imaju elemente veličine $10E-1$ i $10E6$ što predstavlja veliki numerički problem. Znatno utječe na točnost izračuna pojedine iteracije i konvergencije sustava. Predstavljen problem je riješen uvođenjem *SVD* dekompozicije i reformulacije problema. Donosi se nakon sljedećeg poglavlja. Promotrimo prvo rješavanje problema određivanja položaja uvođenjem težina, tj. koristeći algoritam 2.

Poboljšan pristup: reformulacija jednadžbi

Budući da upravo predložen pristup rješavanja problema jako sporo konvergira, traži se novi, efikasniji pristup rješavanju problema. Ponovno se promatra isti sustav jed-

nadžbi 2.4, ali se metoda iterativnih najmanjih kvadrata ne primjenjuje direktno na početni sustav, već na njegovu modifikaciju, modifikaciju sustava 3.1. Modificirani sustav je u mogućnosti dati isto dobro rješenje uz uvijet $cd_T < d_i \forall i \in \{1, 2, 3, 4\}$.

Prebacujući član $d = d_T \cdot c$ na lijevu stranu i kvadrirajući obje strane jednadžbi sustava 3.1 dobivamo modificirani sustav jednadžbi:

$$\begin{aligned} (d_1 - d)^2 &= (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 + p_1(\mathbf{x}) \\ (d_2 - d)^2 &= (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 + p_2(\mathbf{x}) \\ (d_3 - d)^2 &= (x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 + p_3(\mathbf{x}) \\ (d_4 - d)^2 &= (x - x_4)^2 + (y - y_4)^2 + (z - z_4)^2 + p_4(\mathbf{x}) \end{aligned} \quad (4.16)$$

gdje je

$$\begin{aligned} p_i(\mathbf{x}) &= 2\sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}v_i + v_i^2 \\ &= (d_i - d)^2 - (x_i - x)^2 - (y_i - y)^2 - (z_i - z)^2. \end{aligned}$$

Označimo

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

pa minimizacijski problem 3.9 prelazi u

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \tilde{\mathbf{p}}(\mathbf{x})^T \tilde{\mathbf{p}}(\mathbf{x}). \quad (4.17)$$

Analogno 3.15 prelazi u

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \tilde{\mathbf{p}}(\mathbf{x})^T \Sigma^{-1} \tilde{\mathbf{p}}(\mathbf{x}). \quad (4.18)$$

Nadalje,

$$\tilde{\mathbf{p}}'(\mathbf{x}) = (p'_1(\mathbf{x}), p'_2(\mathbf{x}), p'_3(\mathbf{x}), p'_4(\mathbf{x}))^T = \begin{bmatrix} 2(x_1 - x) & 2(y_1 - y) & 2(z_1 - z) & -2c(d_1 - cd_T) \\ 2(x_2 - x) & 2(y_2 - y) & 2(z_2 - z) & -2c(d_2 - cd_T) \\ 2(x_3 - x) & 2(y_3 - y) & 2(z_3 - z) & -2c(d_3 - cd_T) \\ 2(x_4 - x) & 2(y_4 - y) & 2(z_4 - z) & -2c(d_4 - cd_T) \end{bmatrix}$$

Označimo s

$$\begin{aligned} P &:= \begin{bmatrix} (x_1 - x) & (y_1 - y) & (z_1 - z) & (d_1 - d) \\ (x_2 - x) & (y_2 - y) & (z_2 - z) & (d_2 - d) \\ (x_3 - x) & (y_3 - y) & (z_3 - z) & (d_3 - d) \\ (x_4 - x) & (y_4 - y) & (z_4 - z) & (d_4 - d) \end{bmatrix} \\ \tilde{I} &:= \text{diag}(1, 1, 1, -c). \end{aligned}$$

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

Imamo

$$\Delta \mathbf{x}_k = -(2\Sigma^{-\frac{1}{2}}P\tilde{I})^{-1}(\Sigma^{-\frac{1}{2}}\tilde{\mathbf{p}}(\mathbf{x}_k)) = -\frac{1}{2}(\Sigma^{-\frac{1}{2}}P\tilde{I})^{-1}(\Sigma^{-\frac{1}{2}}\tilde{\mathbf{p}}(\mathbf{x}_k))$$

Opisanim perturbacijama, dobivamo sljedeću izvedbu algoritma 1.

```
library("MASS", "matrixcalc")
#pseudo-udaljenosti
c <- 2.99792458E+08 # brzina svjetlosti [m/s], po GPS standardu
R = read.csv('pseudoranges5a.txt', header = FALSE);
R <- as.matrix(R[,1])
#ucitaj koordinate satelita
S = read.csv('satellites5.txt', header = FALSE)
S <- as.matrix(S)
x_0 = c(1,1,1,1) #[x,y,z,d_T] d_t se kasnije mnozi sa c da bi se oduzeo
delt = c(3,3,3,3)

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)
start.time <- Sys.time()

while(iter < niter){ #ili while(norm(t(delt)) > 1){
x_ = c(x_0[1:3],x_0[4]*c)
P = t(apply(RS, 1, function(x) (x-x_))) #x-x_0
PP = P
PP[,dim(S)[2]+1] = - c*P[,dim(S)[2]+1] #4. red s -c

PX = P*P
PX = PX%*%c(-1,-1,-1,1)

delt <- -(1/2)*svd.inverse(PP)%*%PX
delt <- qr.coef(qr(-2*PI), PX)#
```

```

x_0 = x_0 + delt # (x, y, z, c*dT)

cat(c(iter, delt[1:3]), '\r', file="razmakIteracija.txt", append=TRUE) # upi
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

```

Ukoliko bolje promotrimo gornji program sustav rješavamo invertirajući matricu $P\hat{I}$ koristeći SVD rastav ili pomoću QR faktorizacije.

Definicija 4.3.2 (QR faktorizacija matrice). *Neka je $\mathbf{A} \in \mathbb{R}^{m \times n}$ ili $\mathbb{C}^{m \times n}$, SVD dekompozicija 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 $\mathbf{M}\mathbf{M}^*$, dok su stupci matrice \mathbf{V} svojstveni vektori matrice $\mathbf{M}^*\mathbf{M}$. Dijagonalni elementi matrice \mathbf{D} su korijeni svojstvenih vrijednosti matrice $\mathbf{M}^*\mathbf{M}$ ili $\mathbf{M}\mathbf{M}^*$.*

Problem računanja inverza proizvoljne matrice direktnim računom poprilično je kompleksan i numerički nestabilan, gotovo nikada se ne koristi. Primjenjiviji izračun inverza je korištenjem SVD dekompozicije koji je korišten u prikazanom programu.

U oba slučaja $\Delta \mathbf{x}$ postaje dovoljno mali već nakon 4. iteracije, ali dobivena greška u procjeni položaja je znatna (10E6). Dakle, algoritam konvergira, ali prema lokalnom minimumu. Treba nam bolje rješenje.

Poboljšan pristup: uvođenje težina (WLSM)

Prijašnje metode ne određuju poziciju prijemnika u okvirima dopuštene točnosti. Moguće je da su pojedine jednadžbe točnije, a druge manje točne. Neka to ima utjecaj na određivanje položaja.

Upravo uvođenjem težina modelira se točnost pojedine jednadžbe sustava.

Ponovno lineariziramo desne strane jednadžbi 4.1 i dobivamo da $\forall i \in \{1, 2, 3, 4, \dots\}$

$$d_i - \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2 + (z_k - z_i)^2} - c \cdot (d_T)_k =$$

$$\frac{\partial f}{\partial x} \Delta x_k + \frac{\partial f}{\partial y} \Delta y_k + \frac{\partial f}{\partial z} \Delta z_k + \frac{\partial f}{\partial d_T} \Delta (d_T)_k = \begin{bmatrix} \frac{\partial f_i}{\partial x} & \frac{\partial f_i}{\partial y} & \frac{\partial f_i}{\partial z} & \frac{\partial f_i}{\partial d_T} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta d_T \end{bmatrix}$$

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

Općenito, i može biti proizvoljan. Glavno sadrži najmanje onoliko nezavisnih jednadžbi koliko sustav ima nepoznanica. Do sada se većinom promatrao sustav $i = 4 = \text{broj nepoznanica sustava}$ s međusobno nezavisnim jednadžbama. Nastavimo u istom tonu.

Uz

$$\mathbf{A} := \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} & \frac{\partial f_1}{\partial d_T} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z} & \frac{\partial f_2}{\partial d_T} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial z} & \frac{\partial f_3}{\partial d_T} \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial z} & \frac{\partial f_4}{\partial d_T} \end{bmatrix} \quad (4.19)$$

$$= \begin{bmatrix} \frac{(x-x_1)}{\sqrt{(x-x_1)^2 + (y-y_1)^2 + (z-z_1)^2}} & \frac{(y-y_1)}{\sqrt{(x-x_1)^2 + (y-y_1)^2 + (z-z_1)^2}} & \frac{(z-z_1)}{\sqrt{(x-x_1)^2 + (y-y_1)^2 + (z-z_1)^2}} & c \\ \frac{(x-x_2)}{\sqrt{(x-x_2)^2 + (y-y_2)^2 + (z-z_2)^2}} & \frac{(y-y_2)}{\sqrt{(x-x_2)^2 + (y-y_2)^2 + (z-z_2)^2}} & \frac{(z-z_2)}{\sqrt{(x-x_2)^2 + (y-y_2)^2 + (z-z_2)^2}} & c \\ \frac{(x-x_3)}{\sqrt{(x-x_3)^2 + (y-y_3)^2 + (z-z_3)^2}} & \frac{(y-y_3)}{\sqrt{(x-x_3)^2 + (y-y_3)^2 + (z-z_3)^2}} & \frac{(z-z_3)}{\sqrt{(x-x_3)^2 + (y-y_3)^2 + (z-z_3)^2}} & c \\ \frac{(x-x_4)}{\sqrt{(x-x_4)^2 + (y-y_4)^2 + (z-z_4)^2}} & \frac{(y-y_4)}{\sqrt{(x-x_4)^2 + (y-y_4)^2 + (z-z_4)^2}} & \frac{(z-z_4)}{\sqrt{(x-x_4)^2 + (y-y_4)^2 + (z-z_4)^2}} & c \end{bmatrix} \quad (4.20)$$

$$\mathbf{x} := \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta d_T \end{bmatrix} \quad (4.21)$$

$$\mathbf{b} := \begin{bmatrix} d_1 - \sqrt{(x_k - x_1)^2 + (y_k - y_1)^2 + (z_k - z_1)^2} - c \cdot (d_T)_k \\ d_2 - \sqrt{(x_k - x_2)^2 + (y_k - y_2)^2 + (z_k - z_2)^2} - c \cdot (d_T)_k \\ d_3 - \sqrt{(x_k - x_3)^2 + (y_k - y_3)^2 + (z_k - z_3)^2} - c \cdot (d_T)_k \\ d_4 - \sqrt{(x_k - x_4)^2 + (y_k - y_4)^2 + (z_k - z_4)^2} - c \cdot (d_T)_k \end{bmatrix} \quad (4.22)$$

imamo sustav:

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

i

$$\mathbf{p}(\mathbf{x}) := \mathbf{Ax} - \mathbf{b}$$

Izvedba

Prilikom izvedbe algoritma koristimo sustav s $i = 7$ jednadžbi. Slijedi programski isječak koji izvodi opisanu metodu.

f

Usporedba osnovnog i poboljšanog algoritma

+ zaključak zašto je bolji.

4.4 Zaključak

Potrebno promatrati i druge parametre, utjecaje refleksije, ionosfere itd.

Bibliografija

- [1] Simo Ali-Löytty, Jussi Collin, Helena Leppäkoski, Hanna Sairo i Niilo Sirola, *Mathematics and methods for Positioning*, 2008, http://math.tut.fi/courses/MAT-45806/mathematics_and_methods_for_positioning_2008.pdf [Online; accessed 12. srpnja 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).
- [3] DANA, P. H., *Global positioning system overview*, <http://www.colorado.edu/geography/gcraft/notes/>, 1994, [Online; accessed 9-Feb-2017].
- [4] Zlatko Drmač, Miljenko Marušić, Sanja Singer, Vedran Hari, Mladen Rogina i Saša Singer, *Numerička matematika*, 2003, https://web.math.pmf.unizg.hr/~singer/num_mat/num_anal.pdf [Online; accessed 12. srpnja 2017.].
- [5] Y. He i A. Bilgic, *Iterative least squares method for global positioning system*, 2011, <https://pdfs.semanticscholar.org/fcb1/86f5b7feb713e970fd076498e93a77f7f2fc.pdf> [Online; accessed 12. srpnja 2017.].
- [6] J. Sanz Subirana, J.M. Juan Zornoza and M. Hernández-Pajares, *Fundamentals and Algorithms*, sv. 1, May 2013, http://gage.upc.edu/sites/default/files/TEACHING_MATERIAL/GNSS_Book/ESA_GNSS-Book_TM-23_Vol_I.pdf [Online; accessed 12. srpnja 2017.].
- [7] United States military, *GLOBAL POSITIONING SYSTEM STANDARD POSITIONING SERVICE SIGNAL SPECIFICATION*, <http://www.gps.gov/technical/ps/1995-SPS-signal-specification.pdf>, 1995, [Online; accessed 30-Jan-2017].
- [8] PMF, *Redovi potencija i Taylorovi redovi*, https://web.math.pmf.unizg.hr/nastava/analiza/files/ch3_3.pdf, 2009, [Online; accessed 9-Feb-2017].

- [9] Saša Singer, *Numerička matematika*, 7. predavanje, 2017, http://degiorgi.math.hr/~singer/num_mat/NM_1617/07.pdf[Online; accessed 12. srpnja 2017.].

Sažetak

Satelitsko određivanje položaja predstavlja temeljnu tehnologiju rastućeg broja tehnoloških i društveno-ekonomskih sustava. Kvaliteta njihovih usluga određena je točnoću procjene položaja satelitskim sustavima. Programski određen radioprijamnik za satelitsku navigaciju procesira signale za određivanje položaja i podatke iz navigacijske poruke u tri osnovne domene: radiofrekvencijskoj, u domeni osnovnog frekvencijskog područja te u domeni navigacijske primjene. Ovaj rad analizira postupak procjene položaja u domeni navigacijske primjene. U tu svrhu, koriste se na osobnom računalu izveden programski određen GPS prijamnik i ulazni podatci o opaženim pseudoudaljenostima spremljeni u RINEX podatkovnom formatu. Analiza korištenog algoritma procjene položaja temelji se na izmjerenim pseudoudaljenosti (Sanz Subirana et al, 2013, Chapter 6.1) te se otkrivaju potencijalne slabosti algoritma s učincima na točnost procjene položaja. Na kraju, predlažu se poboljšanja algoritma te ih se izvodi u programskom okruženju R. Poboljšanja algoritma su vrednovana komparativnom analizom obilježja poboljšanog i izvornog algoritma.

Summary

In this ...

Životopis

Dana ...

Dodatak A

Taylorov red potencija

Primjetimo kako smo na stranici 21 pretpostavili kako će za rezidualnu funkciju $\mathbf{p}(\mathbf{x})$ postojati njezin razvoj u Taylorov red oko svake točke \mathbf{x}_k . Ipak, Taylorov red nije definiran za svaku funkciju na $\mathbb{R}^n, n \in \mathbb{N}$. Prilikom primjene Iterativne metode najmanjih kvadrata, treba zahtijevati da funkcija $\mathbf{p}(\mathbf{x})$ i točka x_k zadovoljava uvjete definicije razvoja funkcije u Taylorov red oko točke x_k [8].

Definicija A.0.1. *Neka je $f : \mathbf{I} \rightarrow \mathbb{R}$ funkcija klase $C^\infty(\mathbf{I})$ definirana na otvorenom intervalu $\mathbf{I} \subseteq \mathbb{R}^n$ i neka je $c \in \mathbf{I}$. Red potencija*

$$T[f, c] := \sum_{n=0}^{\infty} \frac{f^{(n)}}{n!} (x - c)^n \quad (\text{A.1})$$

*nazivamo **Taylorov red** funkcije f oko točke c .*

Također, pretpostavlja se kako je

$$\mathbf{p}(\mathbf{x}_{k+1} + \Delta \mathbf{x}_k) = T[\mathbf{p}, x_k] \quad (\text{A.2})$$

što općenito nije točno. Naime, Taylorov red $T[f, c]$ funkcije $f \in C^\infty(\mathbf{I})$ nužno ne konvergira za svaki $x \neq c, x \in \mathbf{I}$ ili može konvergirati prema nekoj drugoj funkciji. Uvjete pod kojima zaista vrijedi A.2 opisani su teoremima u nastavku.

Definicija A.0.2 (Analitička funkcija). *Za $f \in C^\infty(\mathbf{I})$ kažemo da je **analitička u točki** $c \in \mathbf{I}$ ako njezin Taylorov red:*

$$T[f, c] := \sum_{n=0}^{\infty} \frac{f^{(n)}}{n!} (x - c)^n$$

ima radijus konvergencije $R > 0$ i ako postoji $0 < \delta \leq R$ takav da vrijedi

$$f(x) = T[f, c] := \sum_{n=0}^{\infty} \frac{f^{(n)}}{n!} (x - c)^n, \forall x \in \langle c - \delta, c + \delta \rangle \cap \mathbf{I}$$

U oznaci: $f \in C^\omega(\mathbf{I})$.

Teorem A.0.3. Neka je $\sum_{n=0}^{\infty} a_n (x - c)^n$ red potencija s radijusom konvergencije $R > 0$. Za $\mathbf{I} := \langle c - R, c + R \rangle$, funkcija $f : \mathbf{I} \rightarrow \mathbb{R}$ definirana s

$$f(x) := \sum_{n=0}^{\infty} a_n (x - c)^n \quad (\text{A.3})$$

je analitička na čitavom \mathbf{I} . Nadalje, za svaki $\alpha \in \mathbf{I}$ pripadni Taylorov red

$$T[f, \alpha] = \sum_{n=0}^{\infty} \frac{f^{(n)}}{n!} (x - \alpha)^n \quad (\text{A.4})$$

ima radijus konvergencije $\rho \leq R - (c - \alpha)$ i vrijedi

$$f(x) = T[f, \alpha] = \sum_{n=0}^{\infty} \frac{f^{(n)}}{n!} (x - \alpha)^n \quad (\text{A.5})$$

Teorem A.0.4. Neka je $f : \mathbf{I} \rightarrow \mathbb{R}$ funkcija klase $C^\infty(\mathbf{I})$ definirana na otvorenom intervalu $\mathbf{I} \subseteq \mathbb{R}^n$. Tada je $f \in C^\omega(\mathbf{I})$ ako i samo ako za svaki $c \in \mathbf{I}$ postoje $\delta > 0$ i konstante $C > 0$ i $r > 0$ takve da za sve $n \in \mathbb{Z}_+$ vrijedi:

$$\left| f^{(n)}(x) \right| \leq C \frac{n!}{r^n} \quad \forall x \in \mathbf{J} := \langle c - \delta, c + \delta \rangle \cap \mathbf{I} \quad (\text{A.6})$$

U tom slučaju $f(x) = T[f, c](x) \quad \forall x \in \langle c - r, c + r \rangle \cap \mathbf{J}$.

Za primjenu iterativne metode najmanjih kvadrata \mathbf{p} mora biti klase $C^\infty(\mathbf{I})$ gdje je \mathbf{I} unija otvorenih okolina oko svih izračinatih $\mathbf{x}_k, k \in \mathbb{N}$, osim zadnjega. Također, otvorena okolina oko \mathbf{x}_k mora barem sadržavati otvorenu kuglu $K(\mathbf{x}_k, \Delta \mathbf{x}_k)$ i za \mathbf{p} mora vrijediti teorem A.0.3 ili A.0.4.

Dodatak B

Jakobijeva matrica funkcije \mathbf{h} , J

Iz jednakosti 3.8 i $J = \mathbf{p}(\mathbf{x})$ dobivamo

$$J = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \quad (\text{B.1})$$

Za

$$\mathbf{h}(\mathbf{x}) := \begin{bmatrix} \|(s_1 - \mathbf{x}_{1:3})\| \\ \|(s_2 - \mathbf{x}_{1:3})\| \\ \|(s_3 - \mathbf{x}_{1:3})\| \end{bmatrix} \quad (\text{B.2})$$

dobivamo

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

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

Dodatak C

Mjere kvalitete "zviježđda"