

# Asimilacija podataka s primjenama

---

Gašparić, Iskra

Master's thesis / Diplomski rad

2021

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://um.nsk.hr/um:nbn:hr:217:135190>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-29**



Repository / Repozitorij:

[Repository of the Faculty of Science - University of Zagreb](#)



**SVEUČILIŠTE U ZAGREBU**  
**PRIRODOSLOVNO–MATEMATIČKI FAKULTET**  
**MATEMATIČKI ODSJEK**

Iskra Gašparić

**ASIMILACIJA PODATAKA S**  
**PRIMJENAMA**

Diplomski rad

Voditelj rada:  
prof.dr.sc. Zlatko Drmač

Zagreb, 2021.

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. \_\_\_\_\_

*Obitelji i prijateljima za konstantnu potporu.*

# Sadržaj

<b>Uvod</b>	<b>1</b>
<b>1 Asimilacija podataka u diskretnom vremenu</b>	<b>4</b>
1.1 Formulacija problema . . . . .	4
1.2 Primjer: Položaj i brzina objekta . . . . .	5
1.3 Primjer: širenje epidemije . . . . .	6
1.4 Primjer: Lorenzov atraktor . . . . .	8
<b>2 Varijacijske metode</b>	<b>12</b>
2.1 Izvod 3D-Var i 4D-Var metode . . . . .	12
2.2 Bayesova formulacija . . . . .	13
2.3 Inkrementalna 4D-Var metoda . . . . .	22
<b>3 Kalmanov filter</b>	<b>26</b>
3.1 Izvod metode . . . . .	26
3.2 Nelinearni slučaj . . . . .	32
3.3 Aproksimacije nižeg ranga . . . . .	33
<b>A MATLAB kodovi</b>	<b>39</b>
A.1 Lorenzovi atraktori . . . . .	39
A.2 3D-Var metoda . . . . .	41
A.3 Kalmanovi filteri (položaj i brzina objekta) . . . . .	42
<b>Bibliografija</b>	<b>45</b>

# Uvod

Modeliranje stvarnih procesa podrazumijeva kombiniranje matematičkih modela s dostupnim podacima. Ukoliko su nakon definiranja modela dostupni novi podaci, poželjno je prilagoditi model i njegove parametre kako bi što bolje odgovarali novim informacijama. Taj koncept nazivamo asimilacijom podataka. Prirodno se pojavljuje u brojnim primjenama od kojih ćemo za početak istaknuti meteorologiju, oceanografiju te epidemiologiju. Meteorologija proučava izuzetno velik broj parametara: temperaturu zraka, tlak, vjetar, smjer vjetra i sl. Sve navedene veličine mjerene su u određenim točkama na Zemlji (odnosno, primjerice, u zraku ili oceanu), svakoga dana po nekoliko puta. Ukoliko su ti podaci, primjerice, korišteni za vremensku prognozu, napravljena je predikcija za sljedećih nekoliko dana na temelju svih prethodnih podataka. Nakon određenog vremena pristižu nova mjerenja koja bi prognozu mogla učiniti preciznijom. Tada je nove podatke potrebno uklopiti u model. Oni su često prirodno strukturirani u obliku vektora ili matrica. Velika količina podataka rezultira matricama velikih dimenzija (npr.  $10^9$ ). Zbog toga se često pribjegava metodama numeričke linearne algebre, poput aproksimacije matricama manjeg ranga. Naravno, sam postupak asimilacije podataka mora se odvijati brzo i efikasno. Pretpostavimo da predviđamo vremenske uvjete u trenutku  $t_i$  za trenutak  $t_{i+1}$ . Jasno je da je važno da izračun bude gotov prije nego trenutak  $t_{i+1}$  dođe, a cilj je da proces završimo i još puno brže. Osim toga, nisu dostupna mjerenja u svakoj točki u kojoj želimo raditi predikcije. Zbog toga je potrebno na odgovarajući način upotrijebiti podatke u diskretnim točkama u svrhu aproksimiranja stanja u cijeloj domeni. Pojavljuju se (moguće nelinearna) preslikavanja koja stvaraju problem u implementaciji. Sama veza između dva stanja na susjednim indeksima je u primjeni također često nelinearna, što može stvoriti različite izazove u algoritmima.

U prvom poglavlju ovog rada formalno je zapisan problem. Uveden je *operator modela*; općenito nelinearan operator, koji opisuje kako je prethodno stanje preslikano u tre-

nutno. Model u stvarnosti ne mora potpuno odgovarati problemu, već se pojavljuje *greška modela*. Zatim je opisana veza između opažanja i varijabli. Najčešće opažanja nisu same vrijednosti varijabli u tom trenutku, već neka njihova funkcija. Primjerice, pretpostavimo da promatramo djelovanje sila na neku točku u prostoru. Opažanje neće biti svaka od komponenti u  $3D$  prostoru, već njihova rezultanta. Ukoliko varijable definiramo kao jačinu sile u svakom od 3 smjera, tada je opažanje vrijednost neke funkcije te tri varijable. Opisana funkcija naziva se *operator opažanja* koji ponovno dolazi sa pripadnom greškom. Nakon što su uvedeni nabrojani pojmovi, naveden je primjer određivanja položaja i brzine objekta na temelju opaženog približnog položaja u svakom trenutku. Pripadni operatori modela i opažanja bit će linearni - opisani matricom. Zatim je definiran Lorenzov sustav koji ilustrira moguću nelinearnost modela. Na kraju poglavlja nalazi se rezultat algoritma asimilacije podataka primijenjenog na Lorenzov atraktor.

U drugom poglavlju opisane su varijacijske metode za asimilaciju podataka. Ideja je u svakom trenutku pronaći vrijednosti varijabli koje uvažavaju i apriornu predikciju i opažanja. Dakle, varijacijske metode minimiziraju funkciju cilja koja opisuje udaljenost parametara od obje navedene vrijednosti.  $3D$ -Var metoda najbolju predikciju nalazi koristeći opažanja u jednom diskretnom vremenskom trenutku, dok  $4D$ -Var optimizaciju vrši nad vremenskim intervalom. Osim njih, opisana je i inkrementalna  $4D$ -Var metoda u kojoj se, kao numeričko poboljšanje, umjesto nelinearne funkcije cilja iz  $4D$ -Var algoritma minimizira niz linearnih konveksnih funkcija.

Konačno, u trećem poglavlju uvodimo *Kalmanov filter* kao primjer sekvencijalne metode asimilacije podataka. U svakom diskretnom vremenskom koraku, koristeći samo predviđanje iz prethodnog koraka i novo opažanje, kreira se novo predviđanje. Sam algoritam podijeljen je u dva koraka - predikciju i korekciju. U koraku predikcije kreira se apriorna procjena vrijednosti u trenutnom koraku pomoću predviđene vrijednosti u prethodnom koraku. Zatim, u koraku korekcije, u određenom omjeru uvažavaju se i nova opažanja kako bi procjena bila točnija. Dva navedena dijela algoritma iteriraju u svakom vremenskom koraku. Međutim, ukoliko u nekom trenutku nisu dostupna nova mjerenja, korekcija se preskače, a ako je dostupno više opažanja (na primjer, sa više mjernih uređaja), korekcija se može obaviti za svako opažanje. Kalmanov filter pretpostavlja da su operatori modela i opažanja linearni što predstavlja veliko ograničenje. Ono se rješava u *Extended Kalman Filter* algoritmu u kojem se za određene korake algoritma koriste linearizacije tih operatora. Matrice koje se pojavljuju u algoritmu su često velikih dimenzija, što nega-

tivno utječe na efikasnost algoritma. Štoviše, budući da se pojavljuje invertiranje matrica te traženje pozitivnog korijena matrica, algoritam može biti teško provesti na računalu. Zato koristimo aproksimacije matrica nižeg ranga i dolazimo do *SEEK filter* te popularnog *Ensemble Kalman filter (EnKF)* algoritma.



# Poglavlje 1

## Asimilacija podataka u diskretnom vremenu

U ovom poglavlju formaliziramo problem asimilacije podataka. Zapisan je dinamički sustav koji opisuje proces te veza između opažanja i promatranih varijabli. Uz određene pretpostavke o greški modela i greški mjerenja, definirani su svi elementi potrebni za izvod različitih metoda za asimilaciju podataka. Nadalje, navedena su dva primjera - jedan linearan te jedan nelinearan na kojima ćemo kasnije testirati neke od metoda.

### 1.1 Formulacija problema

Signal  $x$  zadan je preslikavanjem

$$x_{j+1} = \mathcal{M}_j(x_j), j \in \mathbb{N}_0 \quad (1.1)$$

$$x_0 \sim N(m_0, C_0). \quad (1.2)$$

Dakle,  $x_0$  pripada normalnoj distribuciji. Vrijednost njenog medijana je  $m_0$ , a  $C_0$  je varijacijska matrica. Dodatno,  $\mathcal{M}_j \in C(\mathbb{R}^n, \mathbb{R}^n), \forall j \in \mathbb{N}_0$ ,  $m_0$  naziva se *pozadinski medijan*, a  $C_0$  *pozadinska varijanica*. U opisanom modelu nije u obzir uzet šum. Takav model naziva se *determinističkom dinamikom*. U suprotnom, izrazu (1.1) dodajemo *grešku modela*  $w_j \sim N(0, \Sigma)$ , gdje je matrica  $\Sigma$  realna pozitivno definitna matrica ( $\Sigma > 0$ ):

$$x_{j+1} = \mathcal{M}_j(x_j) + w_j, j \in \mathbb{N}_0 \quad (1.3)$$

te govorimo o *stohastičkoj dinamici*. Nisu poznate točne vrijednosti  $x_j$  u diskretnim vremenskim trenutcima  $j$ , već su poznata opažanja  $y_j$ . Formalno, za  $j \in \mathbb{N}_0$  dane su vrijednosti

$$y_{j+1} = \mathcal{H}_{j+1}(x_{j+1}) + v_{j+1}, j \in \mathbb{N}_0, \quad (1.4)$$

gdje je  $\mathcal{H}_{j+1} \in C(\mathbb{R}^n, \mathbb{R}^m)$  općenito nelinearna funkcija koju zovemo *operator opažanja*. Opažanja često nisu same vrijednosti varijabli u promatranom trenutku, već vrijednosti funkcije tih varijabli,  $\mathcal{H}_{j+1}$ .  $v = \{v_j\}_{j \in \mathbb{N}}$  je niz nezavisnih jednako distribuiranih slučajnih varijabli iz normalne distribucije  $N(0, \Gamma)$ ,  $\Gamma > 0$  koji predstavlja *grešku u mjerenju*. Pretpostavke da greške  $w$  i  $v$  pripadaju normalnoj distribuciji nisu potrebne u samoj definiciji problema, no koristit će se u daljnjoj analizi. Dodatno, pretpostavljamo da su  $x_0$ ,  $w$  i  $v$  nezavisne slučajne varijable.

Cilj asimilacije podataka je popravak modela koji opisuje nepoznati signal  $x$  pomoću opažanja  $y$  koja nisu korištena u oblikovanju postojećeg modela.

Promotrimo diskretna opažanja do trenutka  $t_N$  i označimo ih sa  $y_0, y_1, \dots, y_N$ . Ukoliko na temelju tih podataka želimo procijeniti vrijednost signala u nekom trenutku  $t \in [0, t_N)$ , nailazimo na problem *interpolacije (izgladivanja)*. Ako je  $t = t_N$ , radi se o *filtriranju*, a ukoliko je  $t > t_N$ , bavimo se problemom *predikcije*.

## 1.2 Primjer: Položaj i brzina objekta

Pretpostavimo da želimo pratiti lokaciju nekog objekta u pokretu, npr. vozila na cesti ili pruzi. Osim početnog položaja, uređaj za praćenje lokacije svakih  $T$  vremenskih jedinica šalje približnu lokaciju vozila. Dakle, varijable u trenutku  $t$  su položaj  $x_t$  i brzina  $v_t$ . Opišimo model pomoću operatora navedenih u prethodnom poglavlju. Najprije želimo vezu između varijabli u dva susjedna diskretna trenutka  $t_k$  i  $t_{k-1}$ . Tražimo sustav oblika  $\mathbf{x}_k = M\mathbf{x}_{k-1} + \mathbf{w}_k$ , gdje je  $\mathbf{x}_k = [x_k, v_k]^T$ . Položaj objekta promijenio se za  $T \cdot v_{k-1}$ . Dobivamo traženi oblik

$$\mathbf{x}_k = M\mathbf{x}_{k-1} + \mathbf{w}_k = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \mathbf{w}_k. \quad (1.5)$$

Greška modela  $w_k$  ima kovarijacijsku matricu  $\Sigma$ . Prema [17], najčešći izbor  $\Sigma$  je *random acceleration (RA)*:

$$\Sigma = \begin{bmatrix} T^4/4 & T^3/2 \\ T^3/2 & T^2 \end{bmatrix} \sigma_q^2. \quad (1.6)$$

Parametre koji opisuju grešku uobičajeno je određivati empirijski. Izbor konstante  $\sigma_q^2$  uvelike mijenja efikasnost asimilacije podataka. Nažalost, ne postoji generalna metoda za određivanje  $\sigma_q$  tako da i on također podilazi istom postupku. Odabir  $\Sigma$  je diskutabilan te se u literaturi pojavljuju i druge opcije, kao što je npr. *random velocity model* koji zanemaruje korelacije u greški pa je  $\Sigma$  dijagonalna.

Dosad smo definirali operator modela  $M$ . Osim njega, još je potrebno izraziti operator opažanja  $H$ . Budući da dobivamo samo informaciju o položaju, zapišimo

$$\mathbf{y}_k = H\mathbf{x}_k + \mathbf{v}_k = [1 \ 0] \mathbf{x}_k + \mathbf{v}_k, \quad (1.7)$$

gdje je  $\mathbf{v}_k$  greška opažanja s kovarijacijskom matricom  $\Gamma = [B_x]$ . Vrijednost  $B_x$  je varijanca greške mjerenja. Ukoliko osim položaja, uređaj mjeri i brzinu, operator opažanja glasio bi

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

a kovarijacijska matrica greške mjerenja  $\Sigma = \begin{bmatrix} B_x & 0 \\ 0 & B_v \end{bmatrix}$  gdje su  $B_x$  i  $B_v$  varijance greške mjerenja položaja i brzine, respektivno. Ovaj problem može se promatrati i u više dimenzija. Međutim, možemo primijetiti da će, na primjer, u  $2D$  prostoru položaj na  $x$ -osi biti ovisan isključivo o starom položaju na  $x$ -osi te o komponenti brzine u tom istom smjeru. Zbog toga bismo zapravo mogli promatrati dva smjera neovisno jedan o drugome.

### 1.3 Primjer: širenje epidemije

Motivirani aktualnom epidemijom virusa *n-Cov2019*, modeliramo širenje općenite epidemije u društvu u nekom vremenskom periodu  $[t, t + r]$ . Ako pretpostavimo da je na promatranom području veličina populacije  $N$  fiksna, tj. da zanemarujemo novorođene, doseljene, odseljene i slično, onda cijelu populaciju u svakom trenutku možemo podijeliti u tri skupine: inficirani ( $I$ ), oporavljeni ( $R$ ) i podložni ( $S$ ). Pritom u skupinu oporavljenih uvrštavamo i sve one koji su bili zaraženi i preminuli (neovisno o uzroku smrti). Dakle, oporavljeni su svi oni koji više ne mogu imati bolest.

Vrijeme  $t$  mjerimo u danima. Funkcije koje promatramo su

$$s(t) = \text{postotak podložnih}$$

$i(t)$  = postotak inficiranih

$r(t)$  = postotak oporavljenih.

Važna informacija za model je koliko ljudi prosječno zarazi jedna zaražena osoba svaki dan. Ako sa  $b$  označimo broj kontakata koje jedna osoba ima u danu, tada će svaka inficirana osoba prosječno zaraziti  $bs(t)$  osoba dnevno. Sada nije teško vidjeti da promjena broja podložnih zadovoljava jednadžbu

$$\frac{ds}{dt} = -bs(t)i(t).$$

Pretpostavimo da se, kad bi infekcija trajala 5 dana, dnevno prosječno oporavi jedna petina trenutno zaraženih. Označimo s  $k$  taj postotak. Dakle, dnevno se oporavi  $ki(t)$  ljudi pa imamo sljedeću jednadžbu

$$\frac{dr}{dt} = ki(t).$$

Dodatno, pretpostavljamo da svaka osoba koja preboli bolest ima imunitet pa se nitko iz skupine oporavljenih ne može vratiti u skupinu podložnih.

Sada je iz te dvije jednadžbe jasno kako se mijenja broj zaraženih. On raste kako broj podložnih pada, a pada kako broj oporavljenih raste pa imamo

$$\frac{di}{dt} = bs(t)i(t) - ki(t).$$

Kako smo na početku pretpostavili da se ukupna veličina populacije u modelu ne mijenja, vrijedi da je

$$\frac{ds}{dt} + \frac{di}{dt} + \frac{dr}{dt} = 0$$

pa vidimo da se model slaže s tom pretpostavkom.

*Adaptivni DA-SIR model* koristi nove podatke (u smislu asimilacije podataka) kako bi poboljšao model. Ukoliko diskretiziramo jednadžbe po vremenu, dobivamo relacije

$$S_{i+1} = S_i + b \frac{I_i S_i}{N} \quad (1.8)$$

$$I_{i+1} = I_i + b \frac{I_i S_i}{N} \quad (1.9)$$

$$R_{i+1} = R_i + kI_i, \quad (1.10)$$

gdje smo sa  $S_i, I_i, R_i$  označili vrijednosti  $s(t_i), i(t_i), r(t_i)$ , respektivno. Označimo sa  $R_i^{obs}$  opažanja varijable  $R_i$ . Tada se problem asimilacije svodi na minimiziranje funkcije cilja

$$J(I) = \sum_{i=t+1}^{t+r} \|R_i^{obs} - R_i^{pred}(I, b, k)\|_{C_0^{-1}} + \|I - I_t^{pred}\|_{\Sigma^{-1}} \quad (1.11)$$

i

$$I_t^{DA} = \operatorname{argmin}_I J(I). \quad (1.12)$$

Vrijednost  $R^{pred}$  označava predikciju iz SIR modela, a matrice  $C_0$  i  $\Sigma$  su, kao i ranije, pozadinske kovarijance grešaka. Kako bismo riješili problem, minimiziramo

$$\beta_t, \gamma_t = \operatorname{argmin}_{\beta, \gamma} \sum_{i=t+1}^{t+r} \|R_i^{obs} - R_i^{pred}(I_t^{DA}, \beta, \gamma)\|_{Q_t^{-1}}. \quad (1.13)$$

Uz dostupne službene podatke o brojevima  $S_t, I_t, R_t$ , ovaj problem je potpuno postavljen i spreman za algoritme asimilacije podataka.

## 1.4 Primjer: Lorenzov atraktor

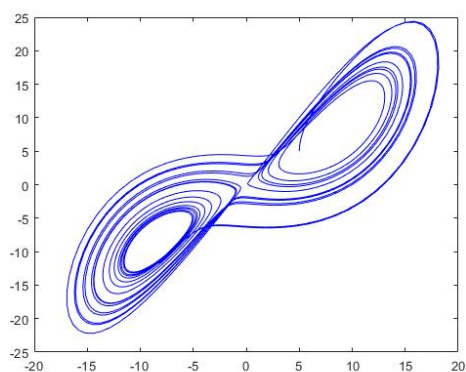
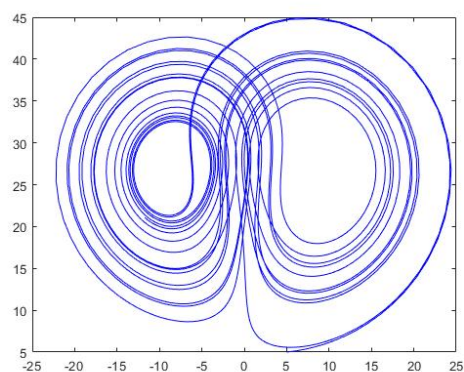
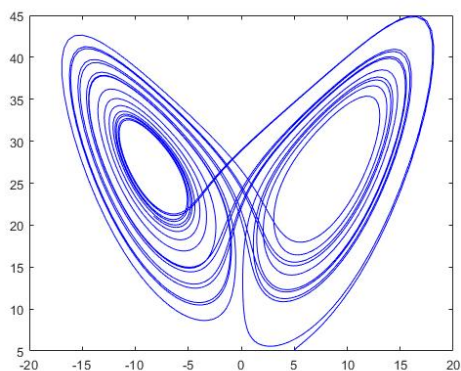
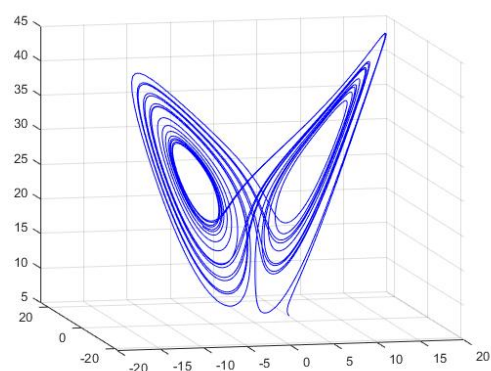
Lorenzov atraktor zadan je sustavom diferencijalnih jednadžbi

$$\begin{aligned} \frac{dv_1}{dt} &= a(v_2 - v_1) \\ \frac{dv_2}{dt} &= -av_1 - v_2 - v_1v_3 \\ \frac{dv_3}{dt} &= v_1v_2 - bv_3 - b(r + a), \end{aligned} \quad (1.14)$$

gdje su konstante  $a, b, r$  najčešće zadane s  $a = 10, b = \frac{8}{3}, r = 28$ .

Ovaj sustav, uz navedene parametre, dobar je primjer budući da je vrlo osjetljiv na početne uvjete. Slika 1.1 prikazuje numeričko rješenje sustava (1.14) uz točne početne uvjete:  $(x, y, z) = (5, 5, 5)$ .

U problemima iz primjene koje asimilacija podataka rješava, početni uvjeti nisu eksplicitno poznati već su dobiveni kao rezultat mjerenja. Zbog toga dolaze s greškom. Pretpostavimo sada da nam nisu poznate početne vrijednosti sustava (1.14), već su one izmjerene. Ovakvu situaciju moguće je simulirati tako da perturbiramo prave vrijednosti slučajnim

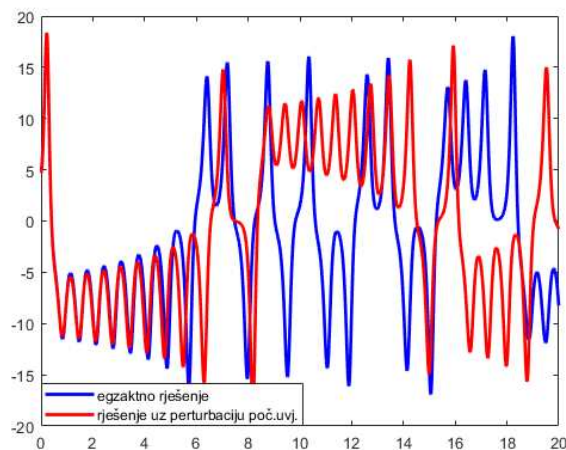
(a)  $xy$ -projekcija(b)  $yz$ -projekcija(c)  $xz$ -projekcija

(d) 3D prikaz rješenja

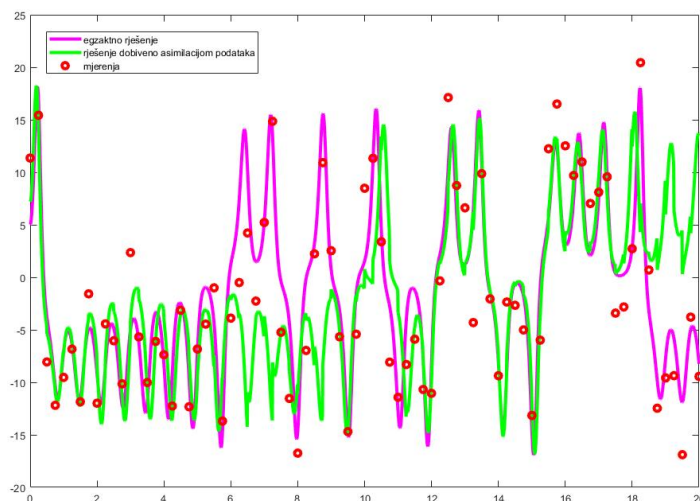
Slika 1.1: Rješenje Lorenz'63 sustava (1.14) uz početne uvjete  $[5, 5, 5]$ 

brojevima iz normalne distribucije  $N(0, 1)$ . Slika 1.2 uspoređuje prvu koordinatu rješenja dobivenog uz perturbirane te uz točne početne uvjete. Možemo primijetiti da se rješenje drastično promijenilo. To je rezultat nestabilnosti sustava na početni uvjet.

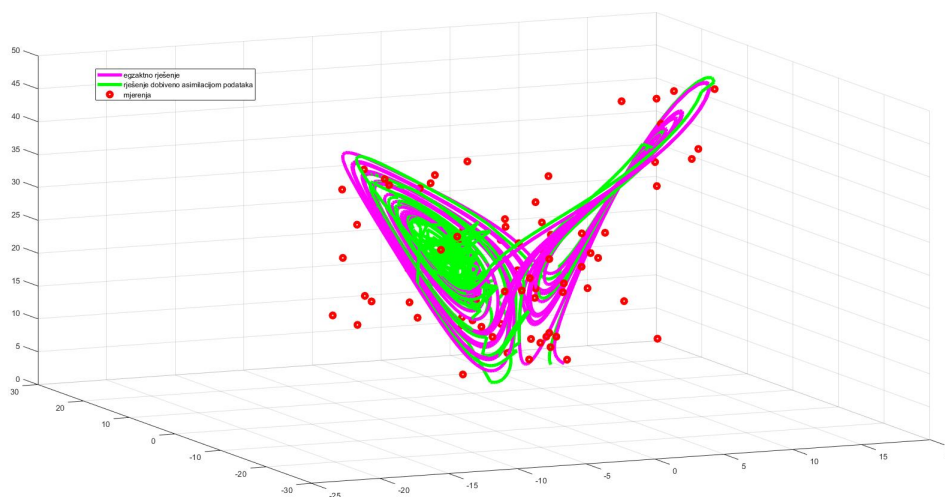
Radi ilustracije, promotrimo kako jedan od algoritama koje ćemo navesti u nastavku rada asimilira rješenje novim podacima. Na Slici 1.3 napravljena je usporedba stvarnog rješenja i podataka s rezultatom algoritma za asimilaciju podataka. Jasno je vidljivo da rezultat dobiven asimilacijom podataka većinom prati liniju stvarnog rješenja. Ukoliko dođe do njihovog razilaženja (npr. u 8. iteraciji), krivulje se ponovno približavaju zahvaljujući mjerenjima. U ovom slučaju napravljeno je 20 iteracija.



Slika 1.2: Prikaz  $x$  koordinate rješenja Lorenz'63 sustava 1.14 uz egzaktno i perturbirane početne uvjete. Možemo primijetiti značajnu razliku u vrijednostima. Otprilike do točke  $x = 8$ , rješenja su bliska. Zatim se trajektorije počinju značajno razlikovati te nakon toga njihovo ponašanje nije ni približno slično.



Slika 1.3:  $x$  koordinata rješenja sustava (1.14) dobivenog asimilacijom podataka i mjerenja. Možemo primijetiti da je krivulja približnog rješenja puno bliža egzaktnom rješenju nego na Slici 1.2. One se ponovno ponekad razilaze, ali zahvaljujući mjerenjima rješenje se modificira i postaje preciznije.



Slika 1.4: 3D prikaz rješenja sustava (1.14) dobivenog asimilacijom podataka. Rješenje je uspoređeno s egzaktnim. Kao početni uvjet za asimilaciju podataka uzet je vektor  $[5, 5, 5]$  perturbiran za slučajne vrijednosti. U svakoj iteraciji "mjerenje" se imitira perturbiranjem egzaktnog rješenja. Možemo primijetiti da ovo numeričko rješenje ostaje na ispravnom *leptiru*, što je i odlika atraktora.



## Poglavlje 2

### Varijacijske metode

U ovom poglavlju opisane su varijacijske *3D-Var* i *4D-Var* metode. Sam izvod napravljen je na dva načina. Najprije intuitivno opisujemo pristup i definiramo metodu, a zatim do istog algoritma dolazimo pomoću Bayesove statistike.

#### 2.1 Izvod 3D-Var i 4D-Var metode

Promotrimo jednu diskretnu vremensku točku  $t_j$  u kojoj su nam poznate vrijednosti mjerenja  $y_j$  te vrijednost  $x_j^F$  dobivena iz modela. Ideja *3D-Var* metode je aproksimaciju iz modela u tom vremenskom trenutku približiti mjerenjima. Precizno, metoda minimizira udaljenost između svake od tih vrijednosti i nove aproksimacije  $x_j^A$ . U modeliranju funkcije troška važno je u obzir uzeti greške mjerenja i modela. Pretpostavimo da greške imaju Gaussovu distribuciju pa ih je dovoljno opisati očekivanjem i varijancom. Prirodno je za distribuciju greške pretpostaviti očekivanje 0 pa preostaje promotriti kovarijacijsku matricu. Kovarijacijsku matricu pozadinske greške označavamo sa  $C_0$ , a sa  $R_j$  kovarijacijsku matricu greške mjerenja u diskretnom trenutku  $t_j$ . Ukoliko pretpostavimo međusobnu nezavisnost tih dviju slučajnih varijabli, definiran je problem:

$$x_j^A = \operatorname{argmin}_x \left( \frac{1}{2} \|x - x_j^F\|_{C_0^{-1}}^2 + \frac{1}{2} \|\mathcal{H}_j(x) - y_j\|_{R_j^{-1}}^2 \right) \quad (2.1)$$

Podsjetimo,  $\|x\|_M = x^T M x$  je *Mahalanobisova udaljenost (Mahalanobis distance)*. Dakle, greške (kovarijance) su uzete u obzir kao težine u normi udaljenosti. Operator  $\mathcal{H}_j$  je kao i ranije operator opažanja.

Do sada smo promatrali stacionarni problem. Umjesto jednog stanja  $x$  u fiksnom trenutku  $t$ , uzmimo u obzir prozor od posljednjih  $J + 1$  vremenskih trenutaka. Dolazimo do formulacije *4D-Var* problema: pronaći  $x$  koji minimizira funkciju troška

$$x_j^A = \operatorname{argmin}_x \left( \frac{1}{2} \|x - x_j^F\|_{C_0^{-1}}^2 + \frac{1}{2} \sum_{j=0}^J \|\mathcal{H}_j(\mathcal{M}_j(x)) - y_j\|_{R_j^{-1}}^2 \right), \quad (2.2)$$

gdje je  $\mathcal{M}_j$  operator modela:  $\mathcal{M}_j(x(t_0)) = x(t_j)$ . Rješenje  $x_j^A$  je optimalan inicijalni uvjet za svaki vremenski interval u smislu podudaranja / sličnosti trajektorije modela tih  $J$  opažanja. Primijetimo da u (2.2) pretpostavljamo da je model savršen, tj. da je greška modela 0. Takvu postavku nazivamo *jakom 4D-Var formulacijom (strong-constraint 4D-Var)*. Promotrimo sada općenitiji slučaj. Kako bismo uveli što manje uvjeta na grešku, ponovno pretpostavljamo Gaussovu razdiobu greške. Označimo sa  $q_j$  grešku modela te sa  $Q_j$  njenu kovarijacijsku matricu. Proširimo funkciju troška:

$$x_j^A = \operatorname{argmin}_x \left( \frac{1}{2} \|x - x_j^F\|_{C_0^{-1}}^2 + \frac{1}{2} \sum_{j=0}^J \|\mathcal{H}_j(x) - y_j\|_{R_j^{-1}}^2 + \frac{1}{2} \sum_{j=0}^J \underbrace{\|x_j - \mathcal{M}_j(x_{j-1}) - \bar{q}\|_{Q_j^{-1}}^2}_{q_j} \right) \quad (2.3)$$

te dobivamo *slabu 4D-Var formulaciju (weak-constraint 4D-Var)*.

## 2.2 Bayesova formulacija

Prisjetimo se Bayesove formule za uvjetnu vjerojatnost:

$$\pi_{X|Y}(x|y) = \frac{\pi_{Y|X}(y|x)\pi_X(x)}{\pi_Y(y)} \propto \pi_{Y|X}(y|x)\pi_X(x). \quad (2.4)$$

$X$  je slučajna varijabla na čiju distribuciju utječu podatci - realizacije slučajne varijable  $Y$ . Podatci nisu korišteni u računanju *apriorne distribucije*  $\pi_X$ . Distribucija  $\pi_{Y|X}$  opisuje vjerojatnost opažanja  $Y$  uz dostupan model - slučajnu varijablu  $X$ .

$$\pi_Y(y) = \int_{\mathbb{R}^n} \pi_{Y|X}(y|x)\pi_X(x) dx$$

je *marginalna distribucija* koja ovisi samo o  $y$ . Njen značaj je normalizacija  $\pi_{X|Y}$ , *aposteriorne distribucije*, dobivene nakon što su dostupna opažanja.

Asimilacija podataka može se shvatiti kao aproksimacija stvarnog stanja sustava u danom trenutku, kombiniranjem opažanja distribuiranih u vremenu i modela. Taj problem prirodno je opisati pomoću Bayesove statistike.  $X$  je slučajna varijabla koja opisuje model, a  $Y$  slučajna varijabla opažanja. Apriorna distribucija jest model prije nego što su dostupni novi podaci, a posteriorna distribucija je model prilagođen novim informacijama. Prema (2.4), za daljnju analizu najprije je potrebno izračunati funkciju gustoće  $\pi_{Y|X}$  te  $\pi_X$ .

Podsjetimo se dinamičkog sustava koji promatramo:

$$x_{j+1} = \mathcal{M}_j(v_j) + w_j, j \in \mathbb{N}_0$$

$$x_0 \sim N(m_0, C_0),$$

gdje je  $\mathcal{M}_j \in C(\mathbb{R}^n, \mathbb{R}^n)$ ,  $\forall j$ , a  $w = \{w_j\}_{j \in \mathbb{N}_0}$  je n.j.d. niz slučajnih varijabli  $w_j \sim N(0, \Sigma)$ ,  $\Sigma > 0$ . Opažanje  $y$  zadovoljava relaciju

$$y_{j+1} = \mathcal{H}_{j+1}(x_{j+1}) + v_{j+1}, j \in \mathbb{N}_0,$$

za  $\mathcal{H}_j \in C(\mathbb{R}^n, \mathbb{R}^m)$ ,  $\forall j$  te  $v = \{v_j\}_{j \in \mathbb{N}}$ . Dodatno smo pretpostavili da su  $x_0$ ,  $w$  i  $v$  međusobno nezavisne.

Promotrimo signal na diskretnom vremenskom intervalu  $\mathbb{J}_0 = \{0, \dots, J\}$  kojeg nazivamo *prozor asimilacije podataka*. Označimo  $x = \{x_j\}_{j \in \mathbb{J}_0}$ ,  $y = \{y_j\}_{j \in \mathbb{J}}$ ,  $w = \{w_j\}_{j \in \mathbb{J}_0}$ ,  $v = \{v_j\}_{j \in \mathbb{J}}$ . Pronađimo apriori vjerojatnost  $\mathbb{P}(x)$ . Računamo:

$$\begin{aligned} \mathbb{P}(x) &= \mathbb{P}(x_J, x_{J-1}, \dots, x_0) \\ &= \mathbb{P}(x_J | x_{J-1}, \dots, x_0) \mathbb{P}(x_{J-1}, \dots, x_0) \\ &\stackrel{(1)}{=} \mathbb{P}(x_J | x_{J-1}) \mathbb{P}(x_{J-1}, \dots, x_0) \\ &= \dots \\ &\stackrel{(2)}{=} \prod_{j=0}^{J-1} \mathbb{P}(x_{j+1} | x_j) \mathbb{P}(x_0), \end{aligned}$$

gdje (1) vrijedi zbog Markovljevog svojstva od  $x$ , a posljednja jednakost (2) dobiva se induktivno. Iz pretpostavke o distribuciji varijable  $x_0$  znamo da vrijedi

$$\mathbb{P}(x_0) \propto e^{-\frac{1}{2} \left( C_0^{-\frac{1}{2}} (x_0 - m_0) \right)^2}.$$

Sa  $A^{\frac{1}{2}}$  označavamo jedinstveni pozitivni korijen simetrične pozitivno semidefinitne matrice  $A \in \mathbb{R}^{n \times n}$ ,  $n \in \mathbb{N}$ . Pozitivni korijen matrice je također pozitivno semidefinitna matrica istog

ranga. Zato je korijen pozitivno definitne matrice također pozitivno definitna matrica te je samim time i regularna pa postoji njen jedinstveni inverz. Inverz pozitivnog korijena matrice označavamo sa  $A^{-\frac{1}{2}}$ . Iz relacije (1.3) vidimo da je

$$\mathbb{P}(x_{j+1}|x_j) = \mathbb{P}(w_j) \propto e^{-\frac{1}{2}\left(\Sigma_0^{-\frac{1}{2}}(x_{j+1}-\mathcal{M}_j(x_j))\right)^2}.$$

Označimo

$$\begin{aligned} J(x) &:= \frac{1}{2}\left(C_0^{-\frac{1}{2}}(x_0 - m_0)\right)^2 + \sum_{j=0}^{J-1} \frac{1}{2}\left(\Sigma^{-\frac{1}{2}}(x_{j+1} - \mathcal{M}_j(x_j))\right)^2 \\ &= \|x_0 - m_0\|_{C_0}^2 + \sum_{j=0}^{J-1} \frac{1}{2}\|x_{j+1} - \mathcal{M}_j(x_j)\|_{\Sigma}^2. \end{aligned} \tag{2.5}$$

Tada je

$$\mathbb{P}(x) \propto e^{-J(x)}.$$

To znači da je apriori vjerojatnost  $\mathbb{P}(x) = \rho_0(x)$  proporcionalna funkciji  $e^{-J(x)}$ . Primijetimo da ova vjerojatnost ne potječe nužno iz Gaussove distribucije jer nismo uveli pretpostavku o linearnosti funkcija  $\mathcal{M}_j$ .

Preostalo je odrediti funkciju vjerodostojnosti. Računamo

$$\begin{aligned} \mathbb{P}(y|x) &= \prod_{j=0}^{J-1} \mathbb{P}(y_{j+1}|x) \\ &= \prod_{j=0}^{J-1} \mathbb{P}(y_{j+1}|x_{j+1}) \\ &\propto \prod_{j=0}^{J-1} e^{-\Phi(x;y)}, \end{aligned}$$

gdje je  $a \propto b$  ukoliko postoji konstanta  $\alpha$  takva da je  $a = \alpha \cdot b$ . Sa  $\Phi$  označen je *funkcional neusklađenosti podataka i modela (model-data misfit)*,

$$\Phi(x; y) = \sum_{j=0}^{J-1} \frac{1}{2}\|y_{j+1} - \mathcal{H}_j(x_{j+1})\|_{\Gamma}^2.$$

Sada vidimo da je funkcija vjerodostojnosti proporcionalna funkciji  $e^{-\Phi(x;y)}$ .

Posteriorna distribucija sada se dobiva pomoću Bayesove formule:

**Teorem 2.2.1.** *Posteriorna izglađujuća distribucija varijable  $X|Y$  za dinamički model (1.3), (1.4) je vjerojatnostna mjera  $\mu$  na  $\mathbb{R}^{|\mathbb{J}_0| \times n}$  čija je vjerojatnosna funkcija gustoće  $\mathbb{P}(x|y) = \rho(x)$  proporcionalna funkciji  $e^{-I(x;y)}$ , gdje je  $I(x; y) = J(x) + \Phi(x; y)$  negativni log-posterior.*

*Dokaz.* Tvrdnja slijedi direktno iz Bayesove formule:

$$\begin{aligned} \pi_{X|Y}(x|y) &\propto \pi_{Y|X}(y|x)\pi_X(x_0) \\ &\propto e^{-\Phi(x;y)} e^{-J(x)} \\ &= e^{-I(x;y)} \end{aligned}$$

□

Primijetimo da posteriorna distribucija općenito nije Gaussova. Za Gaussovu razdiobu potrebna je pretpostavka o linearnosti  $\mathcal{M}$  i  $\mathcal{H}$ .

Srž varijacijskih metoda maksimiziranje je aposteriorne vjerojatnosti:

$$\max_x (\pi_{X|Y}(x|y))$$

Podsjetimo se strukture distribucije koju maksimiziramo u Teoremu 1.4.1 te zaključujemo da tražimo

$$x^* = \operatorname{argmin}_{x \in \mathbb{R}^{|\mathbb{J}_0| \times n}} I(x; y).$$

Iskažimo teorem o egzistenciji i jedinstvenosti minimuma  $x^*$  za stohastičku dinamiku (1.3).

**Teorem 2.2.2.** *Ako je  $\mathcal{M} \in C^1(\mathbb{R}^n, \mathbb{R}^n)$  te  $\mathcal{H} \in C^1(\mathbb{R}^n, \mathbb{R}^m)$ , onda vrijedi:*

1. *Infimum funkcije  $I(\cdot; y)$  postiže se u barem jednoj točki  $x^* \in \mathbb{R}^{|\mathbb{J}_0| \times n}$ . Dakle, funkcija gustoće  $\rho(x) = \mathbb{P}(x|y)$ , odnosno aposteriorna distribucija, postiže maksimum u točki  $x^*$ .*

2. *Označimo sa  $K(x, r) \subseteq \mathbb{R}^{|\mathbb{J}_0| \times n}$  kuglu sa središtem u  $x$  radijusa  $r$ . Tada je:*

$$\lim_{r \rightarrow 0} \frac{\mathbb{P}^\mu(K(x_1, r))}{\mathbb{P}^\mu(K(x_2, r))} = e^{(I(x_2; y) - I(x_1; y))}, \quad \forall x_1, x_2 \in \mathbb{R}^{|\mathbb{J}_0| \times n}. \quad (2.6)$$

**Napomena 2.2.3.** *Tvrdnja 2 zapravo je reformulacija prve tvrdnje. Za fiksiran  $x_2$ , maksimum desne strane izraza (2.6) postiže se u točki  $x_1$  koja minimizira  $I(\cdot; y)$ . Dakle, vjerojatnost kugle malenog radijusa sa središtem u  $x_1$ , neovisno o izboru  $x_2$ , maksimizirana je u središtima koja minimiziraju funkciju  $I(\cdot; y)$ . Prednost ove tvrdnje je to da se u dokazu ne*

koristi Lebesgueova mjera te se stoga dokaz može provesti i u beskonačnodimenzionalnim prostorima. Dakle, zbog toga je moguće proširiti teorem u slučaj kada se promatra neprekidno vrijeme.

*Dokaz.* Dokaz je preuzet iz [13].

1. Prisjetimo se funkcije  $I$ :

$$I(x; y) = \Theta(x; y) + J(x) = \sum_{j=0}^{J-1} \|y_{j+1} - \mathcal{H}(x_{j+1})\|_{\Gamma}^2 + \|x_0 - m_0\|_{C_0}^2 + \sum_{j=0}^{J-1} \frac{1}{2} \|x_{j+1} - \mathcal{M}(x_j)\|_{\Sigma}^2.$$

Ona je očito neprekidna i nenegativna pa je njezin infimum  $\bar{I}$  konačan i nenegativan. Pokažimo da se infimum postiže. Uvedimo minimizirajući niz  $x^{(n)}$  te pretpostavimo bez smanjenja općenitosti da

$$I(x^{(n)}; y) \leq \bar{I} + 1, \forall n \in \mathbb{N}.$$

Iz strukture funkcije  $I(\cdot; y)$  jasno je da

$$x_0 = m_0 + C_0^{\frac{1}{2}} r_0$$

$$x_{j+1} = \mathcal{M}(x_j) + \Sigma^{\frac{1}{2}} r_{j+1}, j \in \mathbb{Z}^+,$$

gdje je  $\frac{1}{2}|r_j|^2 \leq \bar{I} + 1, \forall j \in \mathbb{Z}^+$ . Pomoću ove ocjene možemo zaključiti da postoji  $K > 0$  takav da  $|x^{(n)}| \leq K, \forall n \in \mathbb{N}$ . Dakle, budući da je taj niz ograničen, postoji konvergentan podniz sa limesom u  $x^*$ . Od sada sa  $x^{(n)}$  radi jednostavnosti označavamo taj podniz. Stoga,  $x^{(n)} \rightarrow x^*$  te za svaki  $\epsilon > 0$  postoji  $N = N(\epsilon)$  takav da

$$\bar{I} \leq I(x^{(n)}; y) \leq \bar{I} + \epsilon, \text{ za sve } n \geq N.$$

Budući da je  $I(\cdot; y)$  neprekidna funkcija, ista ocjena vrijedi i za limes  $x^*$ :

$$\bar{I} \leq I(x^*; y) \leq \bar{I} + \epsilon.$$

Puštanjem limesa  $\epsilon \rightarrow 0$ , zaključujemo da  $I(x^*; y) = \bar{I}$ .

Neka je s  $\mu_0$  označena apriorna, a s  $\mu$  aposteriorna vjerojatnosna mjera pridružena (1.3). Neka su sa  $\rho_0$  i  $\rho$  označene Lebesgueove funkcije gustoće pridružene  $\mu_0$  i  $\mu$ , respektivno. Iz (2.5) znamo da je onda

$$\rho_0(x) = \frac{1}{Z_0} e^{-J(x)} \quad \rho(x) = \frac{1}{Z} e^{-J(x) - \phi(x; y)},$$

gdje su  $Z_0 = \int_{\mathbb{R}^{|J_0| \times n}} e^{-J(x)} dx$  i  $Z = \int_{\mathbb{R}^{|J_0| \times n}} e^{-J(x) - \phi(x;y)} dx$ . Sada odmah slijedi da je

$$e^{-J(x)} dx = Z_0 \rho_0(x) dx = Z_0 \mu_0(dx)$$

te također

$$\mu(dx) = \frac{1}{Z} e^{-I(x;y)} dx = \pi_{X|Y}(x; y) dx = \rho(x) dx,$$

pa je  $x^*$  ujedno i točka maksimuma aposteriorne distribucije.

**2.** Funkcija  $I(\cdot, y)$  je klase  $C^1$  zbog istog svojstva funkcija  $\mathcal{M}$  i  $\mathcal{H}$ . Zato vrijedi:

$$\begin{aligned} \mathbb{P}^\mu(K(u, r)) &= \frac{1}{Z} \int_{|x-u| < r} e^{-I(x;y)} dx \\ &= \frac{1}{Z} \int_{|x-u| < r} (e^{-I(u;y)} + e(u; x-u)) dx, \end{aligned}$$

gdje je

$$e(u; x-u) = \left\langle - \int_0^1 D_y I(u + s(x-u); y) ds, x-u \right\rangle.$$

Sada za  $K^\pm > 0$ :

$$-K^-|r| \leq e(u; x-u) \leq K^+|r|$$

za  $|x-u| < r$ . Označimo  $E := e^{I(x_2;y) - I(x_1;y)}$  i ocijenimo izraz iz iskaza:

$$\frac{\mathbb{P}^\mu(K(x_1, r))}{\mathbb{P}^\mu(K(x_2, r))} \leq E \frac{\int_{|x-x_1| < r} e^{K^+|r|} dx}{\int_{|x-x_2| < r} e^{-K^-|r|} dx} = E \frac{e^{K^+|r|}}{e^{-K^-|r|}}.$$

Analogno dobivamo

$$\frac{\mathbb{P}^\mu(K(x_1, r))}{\mathbb{P}^\mu(K(x_2, r))} \geq E \frac{\int_{|x-x_1| < r} e^{-K^-|r|} dx}{\int_{|x-x_2| < r} e^{K^+|r|} dx} = E \frac{e^{-K^-|r|}}{e^{K^+|r|}}.$$

Uz limes  $r \rightarrow 0$ , slijedi tvrdnja. □

**Napomena 2.2.4.** *Pretpostavke o glatkoći funkcija  $\mathcal{M}$  i  $\mathcal{H}$  mogu se ublažiti, no u tom slučaju ne dobivaju se eksplicitne granice kao u ovom dokazu.*

Sada je posve definiran algoritam: potrebno je izračunati

$$x = \operatorname{argmin}_{u \in \mathbb{R}^{|I_0| \times n}} I(u; y).$$

Funkcije troška u obje metode minimiziraju se iterativno:

$$x_0^{(l+1)} = x_0^{(l)} + \alpha^{(l)} d^{(l)},$$

gdje je  $x^{(0)}$  unaprijed zadan. Smjer  $d^{(l)}$  ovisi o metodi. Na primjer, ako je odabran gradijentni spust onda je  $d^{(l)} = -\nabla J(x_0^{(l)})$ . Funkciju troška *4D-Var* metode optimiziramo uz uvjet  $x_i = \mathcal{M}_{i-1}(x_{i-1})$ . Kako bismo riješili problem minimizacije, definiramo *Lagrangeovu funkciju*

$$\mathcal{L}(x_i, \lambda_i) = J(x_0) + \sum_{i=0}^{N-1} \lambda_{i+1}^T (x_{i+1} - \mathcal{M}_i(x_i))$$

i postavljamo Karush-Kuhn-Tuckerove uvjete:

$$\frac{\partial \mathcal{L}(x_i, \lambda_i)}{\partial x_0} = C_0^{-1}(x_0 - x_0^B) + \left( \frac{\partial \mathcal{H}_0}{\partial x}(x_0) \right)^T R_0^{-1}(\mathcal{H}_0(x_0) - y_0) - \left( \frac{\partial \mathcal{M}_0}{\partial x}(x_0) \right)^T \lambda_1 = 0$$

$$\frac{\partial \mathcal{L}(x_i, \lambda_i)}{\partial x_i} = \left( \frac{\partial \mathcal{H}_i}{\partial x}(x_i) \right)^T R_i^{-1}(\mathcal{H}_i(x_i) - y_i) - \left( \frac{\partial \mathcal{M}_i}{\partial x}(x_i) \right)^T \lambda_{i+1} + \lambda_i = 0 \quad , i = 1, \dots, N$$

$$\frac{\partial \mathcal{L}(x_i, \lambda_i)}{\partial \lambda_i} = x_i - \mathcal{M}_{i-1}(x_{i-1}) = 0 \quad , i = 1, \dots, N$$

Sada su jednadžbe za varijable  $\lambda_i, i = 1, \dots, N + 1$  koje mjere osjetljivost funkcije troška na promjenu  $x_i$  dane sa:

$$\lambda_{N+1} = 0 \quad (2.7)$$

$$\lambda_i = \left( \frac{\partial \mathcal{M}_i}{\partial x}(x_i) \right)^T \lambda_{i+1} - \left( \frac{\partial \mathcal{H}_i}{\partial x}(x_i) \right)^T R_i^{-1}(\mathcal{H}_i(x_i) - y_i), i = N, \dots, 0 \quad (2.8)$$

Sada možemo izračunati gradijent funkcije troška:

$$\nabla J(x_0) = -\lambda_0 + C_0^{-1}(x_0 - x_0^B)$$

Gornji izvod proveden je za funkciju troška iz jake formulacije *4D-Var*. Optimizacija za slabu formulaciju puno je kompleksnija. Potrebno je minimizirati funkciju u tri prostorne dimenzije i tri dodatne dimenzije. Osim što je taj račun velike vremenske složenosti,



važno je naglasiti i da  $\mathcal{M}_i$  najčešće nastaje diskretizacijom nelinearne parcijalne diferencijalne jednačbe.

Dodatno, važno je napomenuti da navedeni algoritam kao rješenje daje samo aposteriornu srednju vrijednost (*MAP estimator*). Moguće je aproksimirati i kovarijacijsku matricu  $A$  aposteriorne distribucije računanjem inverza  $A = (\nabla^2 J(x))^{-1}$  (koji je moguće dobiti i eksplicitno ako je model linearan i distribucija stvarno jest Gaussova).

---

### Algoritam 1 4D-Var

---

**Input:** Kovarijacijske matrice grešaka  $R_i$  i  $C_0$

Rutine za primjenu operatora modela i opažanja,  $\mathcal{M}_i$  te  $\mathcal{H}_i$

Linearizacije operatora  $\mathcal{M}_i$ ,  $M_i$  te  $\mathcal{H}_i$ ,  $H_i$  za  $i = 0, \dots, N$

Maksimalan broj iteracija  $l_{\max}$

- 1: inicijaliziraj  $l = 0$  i  $x^{(0)} = x_0^B$
- 2: **while**  $\|\nabla J(x_0^{(l)})\| > \epsilon$  **and**  $l \leq l_{\max}$  **do**
- 3:     izračunaj funkciju troška  $J(x_0^{(l)})$
- 4:     izračunaj  $\nabla J(x_0^{(l)})$
- 5:     izračunaj smjer silaska  $d^{(l)}$  odabranom metodom spusta
- 6:      $x_0^{(l+1)} = x_0^{(l)} + \alpha^{(l)} d^{(l)}$
- 7:      $l = l + 1$
- 8: **end while**

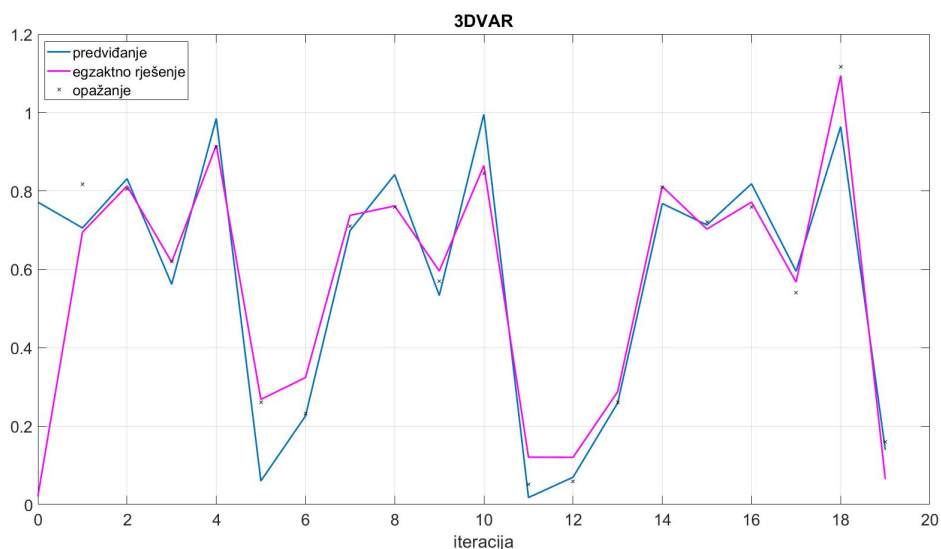
**Output:** Ukupan broj izvršenih iteracija

$x_0^{(0)}, \dots, x_0^{(L)}$

---

### Primjer

Promotrimo rezultate primjene 3D-Var algoritma na dinamički sustav zadan jednačbom  $v_{j+1} = c \cdot v_j(1 - v_j)$ . Na 2.1 uspoređeno je egzaktno rješenje s predviđanjima u svakoj točki. Dodatno su označena i opažanja u svakoj točki.



Slika 2.1: 3D-Var metoda primijenjena na dinamički sustav  $v_{j+1} = c \cdot v_j(1 - v_j)$ . U svakoj iteraciji dostupno je novo opažanje koje utječe na predviđanje.

Ukoliko je  $\mathcal{H} : \mathbb{R}^n \rightarrow \mathbb{R}^p$  linearan operator reprezentiran matricom  $H \in \mathbb{R}^{p \times n}$ , rješenje 3D-Var problema može se odmah izračunati. Definirajmo *Kalmanovu matricu* (*Kalman gain*)  $K$ , koju ćemo kasnije formalno izvesti:

$$K = BH^T(HBH^T + R)^{-1}. \quad (2.9)$$

Sada je minimum, prema [10], dan sa:

$$x_j^A = x_j^F + K(y - H(x_j^F)).$$

Iako je minimum sada zadan eksplicitno, u primjeni je problematičan sam izračun matrice  $K$ . Matrice iz njene definicije su vrlo velike te je računanje inverza neizvedivo. Ponekad nije moguće niti pohraniti cijele matrice.

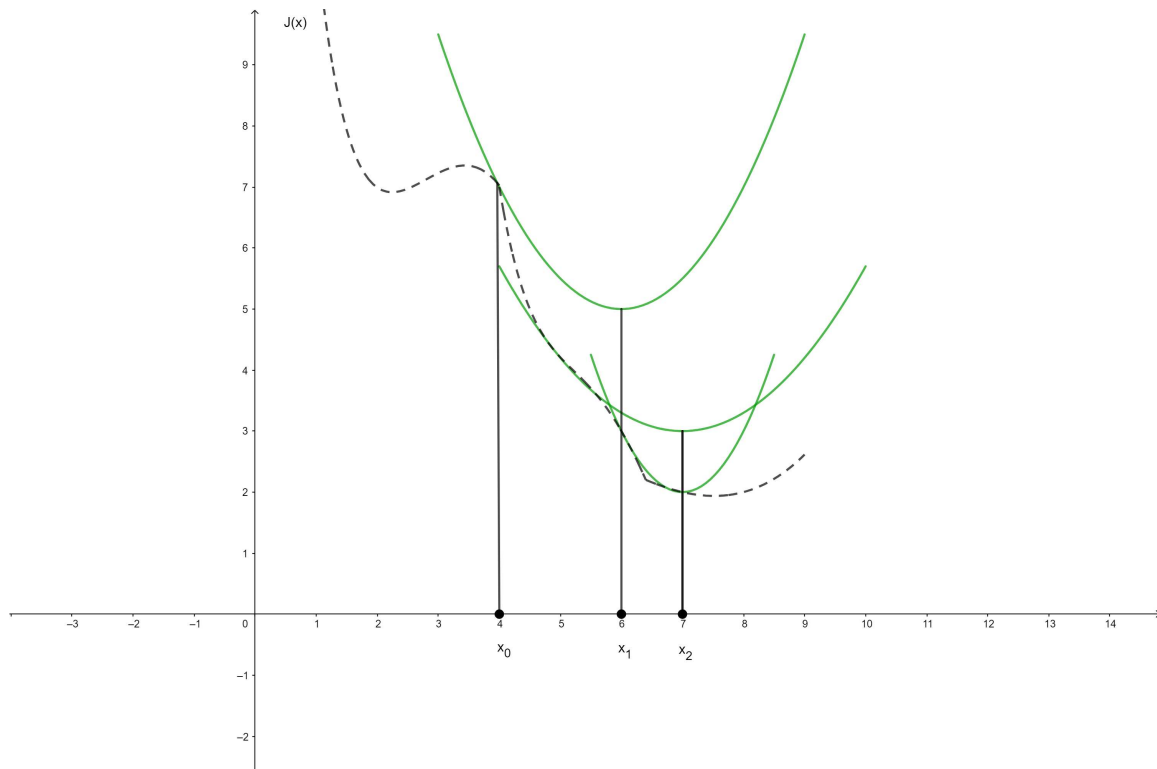
Vratimo se 4D-Var metodi. Funkcija troška definirana je kao:

$$J(x) = \frac{1}{2} \|x - x_j^F\|_{C_0^{-1}}^2 + \frac{1}{2} \sum_{j=0}^J \|\mathcal{H}_j(\mathcal{M}_j(x)) - y_j\|_{R_j^{-1}}^2,$$

gdje je  $x_{j+1} = \mathcal{M}_j(x_j)$  općenito nelinearna funkcija ovisna o vremenu koja opisuje evoluciju dinamičkog sustava. Odmah primijetimo da regularizacijski izraz u  $J(x)$  sadrži samo informacije u trenutku  $t_0$ .

## 2.3 Inkrementalna 4D-Var metoda

Kao što smo vidjeli, varijacijski pristup dovodi do kompleksnog optimizacijskog problema. Prisjetimo se, uvjeti optimizacije su parcijalne diferencijalne jednačbe. Moguće rješenje je aproksimirati minimum pomoću jednostavnijih funkcija cilja. Uvodimo *inkrementalnu 4D-Var metodu* prvo opisanu u [16]. Umjesto nelinearne optimizacije, rješenje se aproksimira nizom minimizacija kvadratičnih funkcija cilja. Budući da je kvadratna funkcija konveksna, u svakoj iteraciji postoji jedinstveno optimalno rješenje. Metoda je zapravo ekvivalentna Gauss-Newtonovoj metodi za nelinearan problem najmanjih kvadrata. Kao što će biti opisano, u strukturi inkrementalne *4D-Var* razlikujemo dvije petlje. Ako je  $x_0^{(l)}$  aproksimacija rješenja u  $x_0$  u  $l$ -toj iteraciji, vanjska petlja modificira vrijednost inkrementa  $\delta x_0^{(l)} = x_0^{(l+1)} - x_0^{(l)}$  pomoću točke minimuma linearizirane funkcije cilja, koja se aproksimira u unutarnjoj petlji. Promotrimo ilustraciju metode na Slici 2.2.



Slika 2.2: Inkrementalna 4D-Var metoda

Inkrementalni 4D-Var algoritam može se opisati pomoću vanjske i unutarnje petlje. U

vanjskoj petlji određuju se linearizacijsko stanje  $x_{0:n}^g$  u vremenskom prozoru koji promatramo,  $[t_0, t_n]$ , te inovacije  $\delta\hat{y}$ . Linearizacijsko stanje dano je sa

$$x_{0:n}^g = \begin{bmatrix} x_0^g \\ \mathcal{M}_1(x_0^g) \\ \mathcal{M}_2(x_0^g) \\ \dots \\ \mathcal{M}_n(x_0^g) \end{bmatrix},$$

gdje je  $x_0^g$  trenutna aproksimacija inicijalnog stanja. Sada je inovacija dana kao

$$\delta\hat{y} = \hat{y} - \hat{\mathcal{H}}(x_0^g),$$

tj. kao razlika između opažanja i aproksimacije inicijalnog stanja preslikane u prostor opažanja. Za računanje  $\hat{\mathcal{H}}(x_0^g)$  koristimo linearizacijsko stanje:

$$[\hat{\mathcal{H}}(x_0^g)]_i = \mathcal{H}_i[\mathcal{M}_i(x_0^g)] = \mathcal{H}_i(x_i^g).$$

Cilj unutarnje petlje je minimizirati linearnu aproksimaciju funkcije troška. Nekoliko iteracija unutarnje petlje ilustrirano je na Slici 2.2. Plavom bojom označena je funkcija cilja  $J(x)$ :

$$J(x) = (x - x^F)^T C_0^{-1} (x - x^F) + \sum_{j=0}^J (\mathcal{H}_j(\mathcal{M}_j(x)) - y_j)^T R_j^{-1} (\mathcal{H}_j(\mathcal{M}_j(x)) - y_j).$$

U početnoj aproksimaciji  $x_0^b$  aproksimirana je kvadratnom funkcijom (zelena boja) dobivenom na sljedeći način. Označimo sa  $H$  linearizaciju operatora opažanja,  $H = \frac{\partial \mathcal{H}}{\partial x}$  i sa  $d = y - \mathcal{H}(x^F)$  odstupanje predviđanja od opažanja. Zapišimo lineariziranu funkciju troška preko  $\delta x = x - x^F$ :

$$J(\delta x) = \delta x^T C_0^{-1} \delta x + (H\delta x - d)^T R^{-1} (H\delta x - d). \quad (2.10)$$

Zbog konveksnosti ove funkcije, postoji jedinstveni minimum. Označimo ga sa  $x_0^g$  (po oznakama sa Slike 2.2). U toj točki nastavlja se isti postupak sve do kad ne dođemo do točke  $x_0^a$ . Kada je zadovoljen kriterij zaustavljanja petlje, ažuriramo aproksimaciju inicijalnog stanja

$$x_0^g = x_0^a + \delta x_0.$$

Nakon što vanjska petlja iskonvergira (ili nakon dovoljnog broja iteracija), vektor  $x_0^g$  predstavlja novu aproksimaciju. Detaljniji opis postupka i rezultati o konvergenciji mogu se pronaći u [18].

Pokažimo još da je inkrementalna 4D-Var doista varijanta Gauss-Newtonove metode dane Algoritmom 2:

---

**Algoritam 2** Gauss-Newtonova metoda

---

**Input:** Rutine za računanje funkcije  $f(x)$  te njenog gradijenta  $G(x)$

Maksimalan broj iteracija  $l_{max}$

Početna iteracija  $x_0$

- 1: inicijaliziraj  $l = 0$  i  $x^{(0)} = x_0$
- 2: **while**  $\|G(x^{(l)})^T f(x^{(l)})\| > \epsilon$  **and**  $l \leq l_{max}$  **do**
- 3:     riješi  $G(x^{(l)})^T G(x^{(l)}) \delta x^{(l)} = -G(x^{(l)})^T f(x^{(l)})$
- 4:      $x^{(l+1)} = x^{(l)} + \delta x^{(l)}$
- 5:      $l = l + 1$
- 6: **end while**

**Output:** Ukupan broj izvršenih iteracija  $L$

$x^0, \dots, x^{(L)}$

---

Najprije pobliže promotrimo ovu metodu. Općenit nelinearan problem najmanjih kvadrata za  $f \in C^2(\mathbb{R}^n, \mathbb{R}^q)$  dan je sa  $\min_x \Phi(x)$ , gdje je

$$\Phi(x) = \frac{1}{2} f(x)^T f(x) = \frac{1}{2} \|f(x)\|_2^2.$$

Ako, kao u algoritmu, označimo sa  $G(x) = f'(x) \in \mathbb{R}^{q \times n}$  Jacobijevu matricu funkcije  $f$ , možemo zapisati gradijent i Hesseovu matricu funkcije  $\Phi(x)$  :

$$\nabla \Phi(x) = G(x)^T f(x)$$

$$\nabla^2 \Phi(x) = G(x)^T G(x) + \sum_{j=1}^q f_j(x) \nabla^2 f_j(x)$$

Iz algoritma možemo primijetiti da Gauss-Newtonova metoda zanemaruje drugi pribrojnik ovog izraza, koji je mnogo kompliciraniji i teži za računanje. Definirajmo sada

$$f(x) = \begin{bmatrix} C_0^{-1/2}(x - x_0^B) \\ R_0^{-1/2}(y_0 - \mathcal{H}_0(x_0)) \\ \dots \\ R_N^{-1/2}(y_N - \mathcal{H}_N(x_N)) \end{bmatrix}$$

Ukoliko primijetimo da je u algoritmu Gauss-Newtonove metode korak

$$G(x^{(l)})^T G(x^{(l)}) \delta x^{(l)} = -G(x^{(l)})^T f(x^{(l)})$$

ekvivalentan rješavanju problema najmanjih kvadrata

$$\min_{s \in \mathbb{R}^n} \|G(x^{(l)})s + f(x^{(l)})\|,$$

jasno je da je minimizacija iz inkrementalne 4D-Var metode zapravo varijanta Gauss-Newtonove metode.

# Poglavlje 3

## Kalmanov filter

### 3.1 Izvod metode

Pretpostavimo da je dinamika sustava dana sa

$$x_{i+1} = M_i x_i + w_i \quad (3.1)$$

$$y_i = H_i x_i + v_i \quad (3.2)$$

gdje su  $w_i \sim \mathcal{N}(0, Q_i)$  greške opažanja te  $v_i \sim \mathcal{N}(0, R_i)$  greške u mjerenju i međusobno su nezavisne. U prethodnom poglavlju vidjeli smo koliko se glavni problem pojednostavio uvođenjem pretpostavki linearnosti. Pretpostavimo ponovno da su operatori  $M_i \in \mathbb{R}^{n \times n}$  i  $H_i \in \mathbb{R}^{p \times p}$  linearni. Pretpostavimo da smo u vremenskom trenutku  $t_i$  pomoću tada dostupnih mjerenja i informacija izračunali najbolju procjenu stanja  $x_i^A$ . Tada se proces može podijeliti na dva dijela. U *koraku predikcije* pomoću modela (3.1) izračunamo predviđanje  $x_{i+1}^F$ . U trenutku  $t_{i+1}$  postaju dostupna i nova opažanja  $y_{i+1}$  pa kombiniranjem nalazimo najbolji linearni nepristrani procjenitelj (Best Linear Unbiased Estimator, *BLUE*)  $x_{i+1}^A$ . Sada je opisan i *korak analize/korekcije*.

Greška predviđanja u trenutku  $t$  (uz egzaktno stanje  $x_t^t \in \mathbb{R}^n$ ) dana je sa  $e_t^F = x_t^F - x_t^t$ , a greška analize  $e_t^A = x_t^A - x_t^t$ . Sada kovarijacijske matrice tih dviju grešaka možemo definirati sa

$$P_t^F = \text{cov}(e_t^F) = \mathbb{E}[e_t^F (e_t^F)^T] \quad (3.3)$$

$$P_t^A = \text{cov}(e_t^A) = \mathbb{E}[e_t^A (e_t^A)^T] \quad (3.4)$$

U koraku predikcije računamo  $x_{i+1}^F = M_i x_i^A$ . Jasno je da nas zanimaju svojstva te greške pa računamo kovarijacijsku matricu  $P_{i+1}^F$ . Najprije raspišimo  $e_{i+1}^F$ :

$$\begin{aligned}
 e_{i+1}^F &= x_{i+1}^F - x_{i+1}^t \\
 &= M_i x_i^A - x_{i+1}^t \\
 &\stackrel{(1)}{=} M_i x_i^A - (M_i x_i^t - w_i) \\
 &= M_i (x_i^A - x_i^t) + w_i \\
 &= M_i e_i^A + w_i
 \end{aligned}$$

U (1) je uvrštena relacija (3.1). Uvrstimo sada dobiveni izraz u definiciju  $P_{i+1}^F$ :

$$\begin{aligned}
 P_{i+1}^F &= \mathbb{E}[e_{i+1}^F (e_{i+1}^F)^T] \\
 &= \mathbb{E}[(M_i e_i^A + w_i)(M_i e_i^A + w_i)^T] \\
 &= \mathbb{E}[(M_i e_i^A + w_i)((e_i^A)^T M_i^T + w_i^T)] \\
 &= \mathbb{E}[M_i e_i^A (e_i^A)^T M_i^T + M_i e_i^A w_i^T + w_i (e_i^A)^T M_i^T + w_i w_i^T] \\
 &\stackrel{(1)}{=} M_i \mathbb{E}[e_i^A (e_i^A)^T] M_i^T + M_i \mathbb{E}[e_i^A w_i^T] + \mathbb{E}[w_i (e_i^A)^T] M_i^T + \mathbb{E}[w_i w_i^T] \\
 &\stackrel{(2)}{=} M_i P_i^A M_i^T + \mathbb{E}[w_i w_i^T] \\
 &\stackrel{(3)}{=} M_i P_i^A M_i^T + Q_i
 \end{aligned}$$

gdje (1) slijedi zbog linearnosti operatora modela i očekivanja, (2) zbog nekoreliranosti grešaka analize  $e_i^A$  i modela  $w_i$ , a u (3) uvrstimo znanje o distribuciji  $w_i \sim \mathcal{N}(0, Q_i)$  pa imamo  $Q_i = \mathbb{E}[w_i w_i^T]$ .

U koraku analize/korekcije računamo a-posteriori procjenitelj  $x_i^A$  pomoću a-priori procjenitelja  $x_i^F$  te opažanja  $y_i$ . Ukoliko pretpostavimo da je procjenitelj  $x_i^A$  vektora  $x_i^t$  linearan i nepristran, onda je  $x_i^A$  oblika

$$x_i^A = x_i^F + K_i (y_i - H_i x_i^F) \quad (3.5)$$

$K_i$  je Kalmanova matrica. Računamo kovarijacijsku matricu  $P_i^A = \mathbb{E}[e_i^A (e_i^A)^T]$ . Najprije



raspišimo  $e_i^A$  :

$$\begin{aligned}
e_i^A &= x_i^A - x_i^t \\
&\stackrel{(1)}{=} x_i^F + K_i y_i - K_i H_i x_i^F - x_i^t \\
&\stackrel{(2)}{=} x_i^F + K_i H_i x_i^t + K_i v_i - K_i H_i x_i^F - x_i^t \\
&= K_i H_i (x_i^t - x_i^F) + e_i^F + K_i v_i \\
&= K_i H_i (-e_i^F) + e_i^F + K_i v_i \\
&= K_i (-H_i e_i^F + v_i) + e_i^F
\end{aligned}$$

U (1) je uvrštena relacija (3.5), a u (2) relacija (3.2). Izračunajmo sada kovarijacijsku matricu:

$$\begin{aligned}
P_i^A &= \mathbb{E}[(K_i(-H_i e_i^F + v_i) + e_i^F)(K_i(-H_i e_i^F + v_i) + e_i^F)^T] \\
&= \mathbb{E}[(-K_i H_i e_i^F + K_i v_i + e_i^F)(-e_i^F)^T H_i^T K_i^T + v_i^T K_i^T + (e_i^F)^T] \\
&= \mathbb{E}[K_i H_i e_i^F (e_i^F)^T H_i^T K_i^T - K_i H_i e_i^F v_i^T K_i^T - K_i H_i e_i^F (e_i^F)^T - K_i v_i (e_i^F)^T H_i^T K_i^T + K_i v_i v_i^T K_i^T \\
&\quad + K_i v_i (e_i^F)^T - e_i^F (e_i^F)^T H_i^T K_i^T + e_i^F v_i^T K_i^T + e_i^F (e_i^F)^T] \\
&= K_i H_i \mathbb{E}[e_i^F (e_i^F)^T] H_i^T K_i^T - K_i H_i \mathbb{E}[e_i^F v_i^T] K_i^T - K_i H_i \mathbb{E}[e_i^F (e_i^F)^T] - K_i \mathbb{E}[v_i (e_i^F)^T] H_i^T K_i^T \\
&\quad + K_i \mathbb{E}[v_i v_i^T] K_i^T + K_i \mathbb{E}[v_i (e_i^F)^T] - \mathbb{E}[e_i^F (e_i^F)^T] H_i^T K_i^T + \mathbb{E}[e_i^F v_i^T] K_i^T + \mathbb{E}[e_i^F (e_i^F)^T] \\
&\stackrel{(1)}{=} K_i H_i \mathbb{E}[e_i^F (e_i^F)^T] H_i^T K_i^T - K_i H_i \mathbb{E}[e_i^F (e_i^F)^T] + K_i \mathbb{E}[v_i v_i^T] K_i^T - \mathbb{E}[e_i^F (e_i^F)^T] H_i^T K_i^T + \mathbb{E}[e_i^F (e_i^F)^T] \\
&\stackrel{(2)}{=} K_i H_i P_i^F H_i^T K_i^T - K_i H_i P_i^F + K_i R_i K_i^T - P_i^F H_i^T K_i^T + P_i^F
\end{aligned}$$

U (1) je uvažena pretpostavka da su  $e_i^F$  i  $v_i$  nekorelirani pa je  $\mathbb{E}[e_i^F v_i^T] = \mathbb{E}[v_i (e_i^F)^T] = 0$ . Ukoliko definiramo  $P_i^F = \mathbb{E}[e_i^F (e_i^F)^T]$  te  $R_i = \mathbb{E}[v_i v_i^T]$ , dobivamo (2). Sada zapišemo dobiveni izraz u pogodnom obliku:

$$P_i^A = (I - K_i H_i) P_i^F (I - K_i H_i)^T + K_i R_i K_i^T \quad (3.6)$$

Kalmanovu matricu  $K_i$  biramo tako da minimizira aposteriornu varijancu  $\text{tr}(P_i^A)$ . Pronađimo njen oblik kao rješenje od  $\frac{\partial}{\partial K_i} \text{tr}(P_i^A) = 0$ . Izvod je preuzet iz [14].

Najprije iskažimo dva matična identiteta. Označimo sa  $\frac{d}{dA}$  derivaciju skalarne funkcije po matrici  $A$ . Rezultat je matrica derivacija te funkcije parcijalno po elementima matrice:  $(\frac{d}{dA} f)_{ij} = \frac{\partial}{\partial a_{ij}} f$ . Za matrice  $A$  i  $B$  takve da je produkt  $AB$  kvadratna matrica, vrijedi

$$\frac{d}{dA} \text{tr}(AB) = B^T. \quad (3.7)$$

Nadalje, ako je  $C$  simetrična,

$$\frac{d}{dA} \text{tr}(ACA^T) = 2AC. \quad (3.8)$$

Za primjer, dokažimo prvi. Neka su  $a_1^T, a_2^T, \dots, a_n^T$  retci matrice  $A$ , a  $b_1, b_2, \dots, b_n$  stupci matrice  $B$ . Računamo:

$$\begin{aligned} \text{tr}(AB) &= \text{tr} \left( \begin{bmatrix} a_1^T \\ a_2^T \\ \dots \\ a_n^T \end{bmatrix} \begin{bmatrix} b_1 & b_2 & \dots & b_n \end{bmatrix} \right) \\ &= \text{tr} \begin{pmatrix} a_1^T b_1 & a_1^T b_2 & \dots & a_1^T b_n \\ a_2^T b_1 & a_2^T b_2 & \dots & a_2^T b_n \\ \dots & \dots & \dots & \dots \\ a_n^T b_1 & a_n^T b_2 & \dots & a_n^T b_n \end{pmatrix} \\ &= \sum_{i=1}^m a_{i1} b_{i1} + \sum_{i=1}^m a_{i2} b_{i2} + \dots + \sum_{i=1}^m a_{in} b_{in} \end{aligned}$$

Sada je  $\frac{\partial}{\partial a_{ij}} \text{tr}(AB) = b_{ji}$  pa slijedi

$$\frac{d}{dA} \text{tr}(AB) = B^T.$$

□

Vratimo se na problem pronalaska Kalmanove matrice  $K_i$  koja minimizira  $\text{tr}(P_i^A)$ . Zapišimo matricu  $P_i^A$  u pogodnijem obliku:

$$\begin{aligned} P_i^A &= (I - K_i H_i) P_i^F (I - K_i H_i)^T + K_i R_i K_i^T \\ &= P_i^F - K_i H_i P_i^F - P_i^F H_i^T K_i^T + K_i (H_i K_i H_i^T + R_i) K_i^T. \end{aligned}$$

Sada je

$$\begin{aligned} \text{tr}(P_i^A) &= \text{tr}(P_i^F) - \text{tr}(K_i H_i P_i^F) - \text{tr}(P_i^F H_i^T K_i^T) + \text{tr}(K_i (H_i K_i H_i^T + R_i) K_i^T) \\ &= \text{tr}(P_i^F) - \text{tr}(K_i H_i P_i^F) - \text{tr}(K_i H_i P_i^F) + \text{tr}(K_i (H_i K_i H_i^T + R_i) K_i^T), \end{aligned}$$

jer je  $\text{tr}(A^T) = \text{tr}(A)$ ,  $\forall A$ . Sada koristeći identitete 3.7 i 3.8 te distributivnost traga matrica prema zbrajanju dobivamo:

$$\frac{d}{dK_i} \text{tr}(P_i^A) = -2(H_i P_i^F)^T + 2K_i (H_i K_i H_i^T + R_i).$$

Uvjet  $\frac{d}{dk_i} \text{tr}(P_i^A)$  zadovoljen je za

$$K_i = P_i^F H_i^T (H_i P_i^F H_i^T + R_i)^{-1}. \quad (3.9)$$

Uvršavanjem 3.9 u 3.6 i dobivamo

$$P_i^A = (I - K_i H_i) P_i^F. \quad (3.10)$$

Intuitivno, malen  $P_i^F$  implicira i malen  $P_i^A$ , što znači da će  $x_i^A$  biti blizu predikcije  $x_i^F$ . Sada možemo zapisati kompletan algoritam.

---

### Algoritam 3 Kalman filter

---

**Input:** Kovarijacijske matrice grešaka  $R_i, Q_i$

Operatori  $M_i, H_i$

Opažanja  $y_i, i = 0, \dots, N$

Početno stanje  $x_0^F$  i pripadna kovarijacijska matrica greške  $P_0^F$

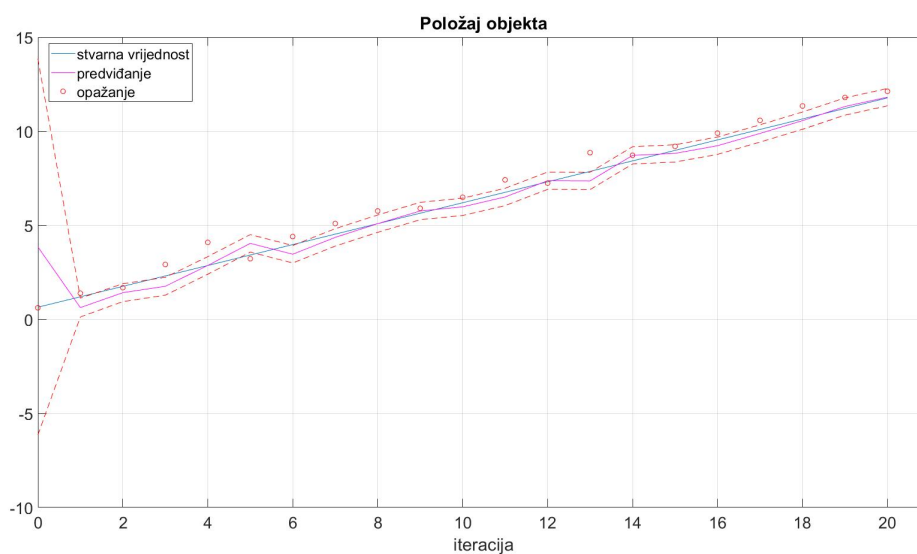
- 1: **for**  $i=0, \dots, N$
- 2:   izračunaj Kalmanovu matricu  $K_i = P_i^F H_i^T (H_i P_i^F H_i^T + R_i)^{-1}$
- 3:   izračunaj predviđanje  $x_i^A = x_i^F + K_i (y_i - H_i(x_i^F))$
- 4:   izračunaj kovarijacijsku matricu greške  $P_i^A = (I - K_i H_i) P_i^F$
- 5:   izračunaj predviđanje  $x_{i+1}^F = M_i(x_i^A)$
- 6:   izračunaj kovarijacijsku matricu greške predviđanja  $P_{i+1}^F = M_i P_i^A M_i^T + Q_i$
- 7: **end for**

**Output:** Vektori  $x_0^A, \dots, x_N^A$  te matrice  $P_0^A, \dots, P_N^A$   
 Vektori  $x_1^F, \dots, x_{N+1}^F$  te matrice  $P_1^F, \dots, P_{N+1}^F$

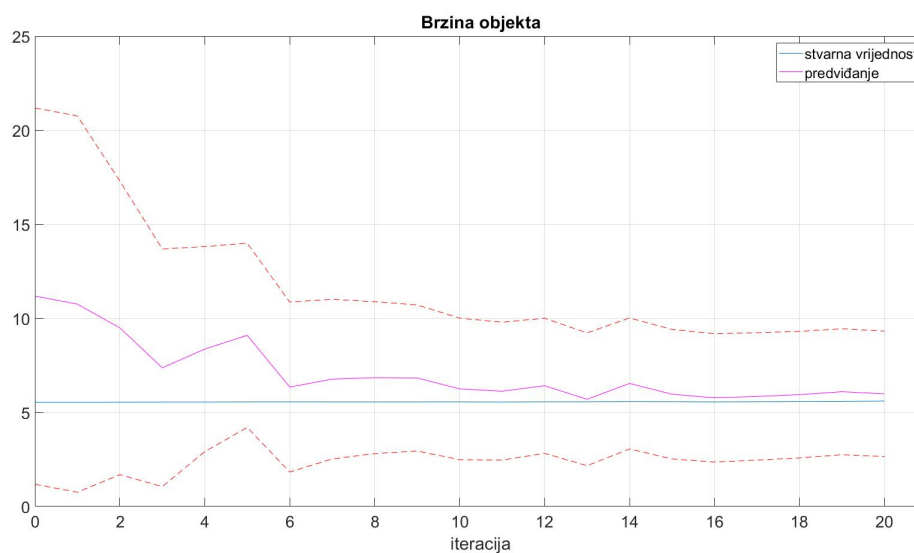
---

### Primjer

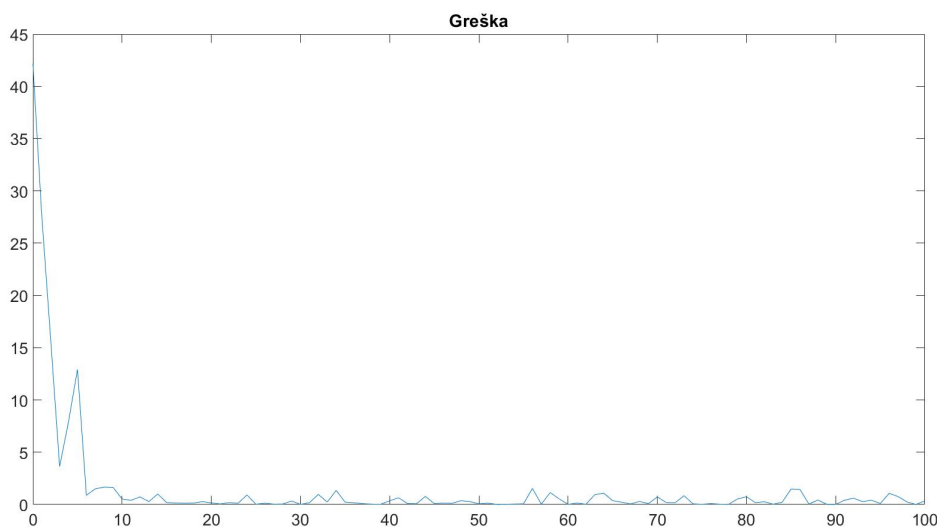
Prisjetimo se dinamičkog sustava 1.7 za praćenje kretanja objekta. Svi elementi algoritma definirani su u uvodnom poglavlju. Početni položaj je 0, a brzina 10. Na slikama 3.1 i 3.2 nalaze se rezultati. Na slici 3.3 prikazana je greška po iteracijama. Možemo primijetiti da se rješenje asimptotski približava pravim vrijednostima.



Slika 3.1: Na slici je prikazana prva komponenta rješenja. Graf obuhvaća prvih 20 iteracija od ukupno izračunatih 100. Početna pozicija je 0, a brzina 7. Za vrijednost parametra  $T$  iz 1.7 odabrano je  $T = 0.1$ . Na slici plavom linijom prikazano je stvarno rješenje, a ljubičastom bojom rješenje dobiveno pomoću navedenog algoritma. Dodatno, označena su i diskretna opažanja te varijanca.



Slika 3.2: Na ovoj slici prikazana je druga komponenta rješenja. Graf obuhvaća prvih 20 iteracija od ukupno izračunatih 100. Početna pozicija je 0, a brzina 7. Za vrijednost parametra  $T$  iz 1.7 odabrano je  $T = 0.1$ . Na slici je plavom bojom prikazano stvarno rješenje, a ljubičastom bojom rješenje dobiveno pomoću navedenog algoritma. Dodatno je na grafu i odgovarajuća varijanca.



Slika 3.3: Greška po iteracijama. Prikazano je svih 100 iteracija. Početna pozicija je 0, a brzina 7. Za vrijednost parametra  $T$  iz 1.7 odabrano je  $T = 0.1$ . Greška je, naravno, najveća u početnoj iteraciji. Međutim, greška već u nekoliko iteracija pada prema nuli.

## 3.2 Nelinearni slučaj

Prethodni algoritam konstruiran je za slučaj da su operatori  $M_i$  i  $H_i$  linearni. *Prošireni Kalmanov filter (Extended Kalman Filter, EKF)* izvedenica je iz navedenog algoritma prilagođena problemima u kojima nisu zadovoljene navedene pretpostavke. Sustav je oblika

$$x_{i+1} = \mathcal{M}_i(x_i) + w_i$$

$$y_i = \mathcal{H}_i(x_i) + v_i.$$

Neka su sa  $M_i$  i  $K_i$  označene linearizacije operatora  $\mathcal{M}_i$  te  $\mathcal{H}_i$ . Nelinearizirani operatori i dalje se koriste pri računanju vektora  $x_i^A$  te  $x_{i+1}^F$  pa dobivamo Algoritam 4.

Početni primjer Lorenzovih jednadžbi sustav je povezanih nelinearnih diferencijalnih jednadžbi.

Kalmanovi filteri, iako jednostavni za implementaciju i korištenje, ograničeni su na primjenu samo na modele malih dimenzija. Kao što smo vidjeli, algoritam prvotno kreiran

**Algoritam 4** Extended Kalman filter

---

**Input:** Kovarijacijske matrice grešaka  $R_i, Q_i$   
 Operatori  $\mathcal{M}_i, \mathcal{H}_i$  te njihove linearizacije  $M_i, H_i$   
 Opažanja  $y_i, i = 0, \dots, N$   
 Početno stanje  $x_0^F$  i pripadna kovarijacijska matrica greške  $P_0^F$

- 1: **for**  $i=0, \dots, N$
- 2:   izračunaj Kalmanovu matricu  $K_i = P_i^F H_i^T (H_i P_i^F H_i^T + R_i)^{-1}$
- 3:   izračunaj predviđanje  $x_i^A = x_i^F + K_i(y_i - \mathcal{H}_i(x_i^F))$
- 4:   izračunaj kovarijacijsku matricu greške  $P_i^A = (I - K_i H_i) P_i^F$
- 5:   izračunaj predviđanje  $x_{i+1}^F = \mathcal{M}_i(x_i^A)$
- 6:   izračunaj kovarijacijsku matricu greške predviđanja  $P_{i+1}^F = M_i P_i^A M_i^T + Q_i$
- 7: **end for**

**Output:** Vektori  $x_0^A, \dots, x_N^A$  te matrice  $P_0^A, \dots, P_N^A$   
 Vektori  $x_1^F, \dots, x_{N+1}^F$  te matrice  $P_1^F, \dots, P_{N+1}^F$

---

za linearne modele može se proširiti i na nelinearan slučaj ukoliko dozvolimo linearizaciju operatora za računanje Kalmanove matrice i kovarijacijskih matrica grešaka. Sada još preostaje problem velikih dimenzija sustava. U primjenama nailazimo na modele velikih dimenzija (npr. reda veličine  $10^8$ ) na koje je gotovo nemoguće primijeniti opisan proces. Ukoliko promotrimo naveden algoritam, vidimo da jedan korak petlje zahtijeva nekoliko evaluacija operatora na vektoru, množenje matrica dimenzija jednakih dimenziji modela te invertiranje izrazito velikih matrica. Zbog toga je sljedeći zadatak prilagoditi algoritam velikim dimenzijama. Generalni pristup je aproksimirati model u prostoru manje dimenzije ili reducirati prostor stanja ili prostor grešaka.

### 3.3 Kalmanovi filteri i aproksimacije nižeg ranga

Navedimo neke algoritme koji koriste aproksimacije matrica nižeg ranga. Najprije izvodimo tzv. *SEEK filter algoritam* (*Singular evolutive extended Kalman filter*) koji koristi aproksimacije kovarijacijskih matrica  $P_i^F$  i  $P_i^A$  faktorizacijom oblika  $P = S S^T$ , gdje je  $S \in \mathbb{R}^{n \times r}$  matrica ranga  $r \ll n$  (značajno manjeg od  $n$ ). Jedan od mogućih načina dobivanja ovakve faktorizacije je skraćena svojstvena dekompozicija. Zapišimo Kalmanovu matricu

i ostale izraze iz algoritma koristeći faktorizacije  $P_i^F = S_i^F(S_i^F)^T$  i  $P_i^A = S_i^A(S_i^A)^T$ :

$$\begin{aligned} K_i &= P_i^F H_i^T (H_i P_i^F H_i^T + R_i)^{-1} \\ &= (S_i^F(S_i^F)^T) H_i^T (H_i (S_i^F(S_i^F)^T) H_i^T + R_i)^{-1} \\ &= S_i^F (H_i S_i^F)^T (H_i S_i^F (H_i S_i^F)^T + R_i)^{-1} \end{aligned}$$

Pomoću Sherman-Morrison-Woodbury identiteta za matrice

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}U(I + V^T A^{-1}U)^{-1}V^T A^{-1}$$

Kalmanovu matricu možemo zapisati i kao

$$K_i = S_i^F \left[ I_r + (H_i S_i^F)^T R_i^{-1} H_i S_i^F \right]^{-1} (H_i S_i^F)^T R_i^{-1} \quad (3.11)$$

Kovarijacijska matrica  $R_i$  često je blok-dijagonalna pa je ovaj izraz povoljniji za računanje. Znamo da je inkrement jednak  $x_i^A - x_i^F = K_i(y_i - H_i(x_i^F))$ , a iz oblika Kalmanove matrice vidimo da je to linearna kombinacija stupaca matrice  $S_i^F$ . Uvrstimo dobivenu Kalmanovu matricu u jednadžbu 3.10:

$$P_i = \left( S_i^F [I_r + (H_i S_i^F)^T R_i^{-1} H_i S_i^F]^{-1/2} \right) \left( S_i^F [I_r + (H_i S_i^F)^T R_i^{-1} H_i S_i^F]^{-1/2} \right)^T = S_i^A (S_i^A)^T$$

uz  $S_i^A = S_i^F [I_r + (H_i S_i^F)^T R_i^{-1} H_i S_i^F]^{-1/2}$ . U koraku predikcije računamo

$$P_{i+1}^F = (M_i S_i^A)(M_i S_i^A)^T + Q_i = \tilde{S}_{i+1}^F (\tilde{S}_{i+1}^F)^T + Q_i$$

gdje smo sa  $\tilde{S}_{i+1}^F$  označili matricu  $M_i S_i^A$ . Ukoliko je model nelinearan, matrica se aproksimira konačnim diferencijama te po stupcima ozačenima sa  $[\cdot]_l$  imamo

$$[\tilde{S}_{i+1}^F]_l = \mathcal{M}_i(x_i^A + [\tilde{S}_{i+1}^A]_l - \mathcal{M}_i(x_i^A)), l = 1, \dots, r$$

Matrica  $Q_i$  mora zadovoljavati određene pretpostavke kako bi vrijedilo  $P_{i+1}^F = S_{i+1}^F (S_{i+1}^F)^T$  te kako bi rang ostao  $r$  (naglasak je na tome da se rang ne poveća). Opisani algoritam naziva se *Singular Evolutive Extended Kalman filter algorithm (SEEK algoritam)* te u nastavku navodimo konkretan Algoritam 5 preuzet iz [10].

**Algoritam 5** SEEK filter**Input:** Kovarijacijske matrice grešaka  $R_i, Q_i$ Rutine za primjenu operatora modela i opažanja  $\mathcal{M}_i$  te  $\mathcal{H}_i, i = 0, \dots, N$ Linearizacije operatora  $M_i$  i  $H_i, i = 0, \dots, N$ Opažanja  $y_i, i = 0, \dots, N$ Početno stanje  $x_0^F$  i pripadna kovarijacijska matrica greške niskog ranga  $P_0^F = S_0^F (S_0^F)^T$ 1: **for**  $i=0, \dots, N$ 

2: izračunaj Kalmanovu matricu niskog ranga

$$K_i = S_i^F \left[ I_r + (H_i S_i^F)^T R_i^{-1} H_i S_i^F \right]^{-1} (H_i S_i^F)^T R_i^{-1}$$

3: izračunaj predviđanje  $x_i^A = x_i^F + K_i(y_i - \mathcal{H}_i(x_i^F))$ 4: izračunaj kovarijacijsku matricu greške  $P_i^A = S_i^A (S_i^A)^T$  uz

$$S_i^A = S_i^F \left[ I_r + (H_i S_i^F)^T R_i^{-1} H_i S_i^F \right]^{-1/2}$$

5: izračunaj predviđanje  $x_{i+1}^F = \mathcal{M}_i(x_i^A)$ 6: izračunaj kovarijacijsku matricu greške predviđanja  $P_{i+1}^F = \tilde{S}_{i+1}^F (\tilde{S}_{i+1}^F)^T + Q_i$ , uz

$$[\tilde{S}_{i+1}^F]_l = \mathcal{M}_i(x_i^A + [\tilde{S}_{i+1}^A]_l) - \mathcal{M}_i(x_i^A)$$

matricu zadanu po stupcima  $[\cdot]_l, l = 1, \dots, r$ 7: **end for****Output:** Vektori  $x_0^A, \dots, x_N^A$  te matrice  $P_0^A, \dots, P_N^A$ Vektori  $x_1^F, \dots, x_{N+1}^F$  te matrice  $P_1^F, \dots, P_{N+1}^F$ 

*Ensemble Kalman filter (EnKF)* još je jedna metoda koja koristi aproksimacije matrica nižeg ranga. Postoje brojne verzije ovog algoritma, od kojih ćemo istaknuti *stohastički EnKF* te *ensemble transform Kalman Filter (ETKF)*. Ponovno zanemarimo vremensku dimenziju te pretpostavimo da je zadano  $r$  apriornih uzoraka  $x_1^F, \dots, x_r^F$ . Kovarijacijsku matricu predviđanja aproksimiramo empirijskom kovarijancom

$$P^F = \frac{1}{r-1} \sum_{k=1}^r (x_k^F - \bar{x}^F)(x_k^F - \bar{x}^F)^T = X^F (X^F)^T \quad (3.12)$$

Sa  $\bar{x}^F$  je označena sredina  $\bar{x}^F = \frac{1}{r} \sum_{k=1}^r x_k^F$ , a sa  $X^F$  matrica dimenzija  $n \times r$  čiji su stupci  $[X^F]_k = \frac{x_k^F - \bar{x}^F}{\sqrt{r-1}}$ . Nakon izračuna predviđanja za svaki član ansambla,

$$x_k^A = x_k^F + K(y - \mathcal{H}(x_k^F)),$$



dobivamo posteriorni ansambl  $X^A$ ,

$$[X^A]_k = \frac{x_k^A - \bar{x}^A}{r-1} \quad (3.13)$$

Preostaje izračunati kovarijacijsku matricu  $P^A = X^A(X^A)^T$ . Pokazuje se da je kovarijanca u takvoj matrici manja nego je ona stvarno. Rješenje je perturbacija vektora opažanja:  $y_k = y + v_k, v_k \sim \mathcal{N}(0, R)$ . Definirajmo

$$[L^F]_k = \frac{Hx_k^F - H\bar{x}^F - (v_k - \bar{v})}{\sqrt{r-1}}$$

Može se pokazati (detalji u [14]) da je za  $[X^A]_k = [X^F]_k - K[L^F]_k$  precizno aproksimirana  $P^A = X^A(X^A)^T$ . Kalmanova matrica može se izračunati pomoću relacije

$$K = X^F(L^F)^T(L^F(L^F)^T)^{-1}.$$

Slijedi cijeli algoritam preuzet iz [10]. Indeks  $k$  odnosi se na indeks u ansamblu, a  $i$  je vremenski indeks.

Opišimo još i *Ensemble Transform Kalman Filter (ETKF)* algoritam. Kovarijacijska matrica greške predviđanja ponovno je aproksimirana empirijskom kovarijancom,

$$P^F = X^F(X^F)^T, [X^F]_k = \frac{x_k^F - \bar{x}^F}{\sqrt{r-1}}$$

Pretpostavimo da je vektor predviđanja  $x^A$  oblika

$$x^A = \bar{x}^F + X^F w^A,$$

uz ansamble  $\{x_k^F\}_{k=1}^r$ , vektor koeficijenata  $w^A$  iz potprostora ansambla  $\mathbb{R}^r$  ( $r$  je mala dimenzija), te  $X^F \in \mathbb{R}^{n \times r}$ . Znamo da je  $x^A = \bar{x}^F + K(y - \mathcal{H}(\bar{x}^F))$  pa uvrstimo u prethodnu relaciju:

$$\bar{x}^F + K(y - \mathcal{H}(\bar{x}^F)) = \bar{x}^F + X^F w^A.$$

Uvrstimo i aproksimaciju kovarijacijske matrice  $P^F = X^F(X^F)^T$  te kao u (3.11) uz nove oznake dobivamo:

$$\begin{aligned} w^A &= (X^F)^{-1} K(y - \mathcal{H}(\bar{x}^F)) \\ &= (X^F)^{-1} X^F \left[ I_r + (\mathcal{H}X^F)^T R^{-1} (\mathcal{H}X^F) \right]^{-1} (\mathcal{H}X^F)^T R^{-1} (y - \mathcal{H}(\bar{x}^F)) \\ &\stackrel{(1)}{=} \left[ I_r + (L^F)^T R^{-1} L^F \right]^{-1} (L^F)^T R^{-1} (y - \mathcal{H}(\bar{x}^F)) \\ &\stackrel{(2)}{=} W^{-1} (L^F)^T R^{-1} (y - \mathcal{H}(\bar{x}^F)), \end{aligned}$$

**Algoritam 6** Ensemble Kalman Filter (EnKF)**Input:** Kovarijacijske matrice grešaka  $R_i$ Rutine za primjenu operatora modela i opažanja  $\mathcal{M}_i$  te  $\mathcal{H}_i$ ,  $i = 0, \dots, N$ Opažanja  $y_i$ ,  $i = 0, \dots, N$ Početna stanja  $x_{k,0}^F$ ,  $k = 1, \dots, r$ 1: **for**  $i=0, \dots, N$ 2: perturbiraj opažanja  $y_{k,i} = y_i + v_k$ ,  $v_k \sim \mathcal{N}(0, R_i)$ ,  $k = 1, \dots, r$ 

3: izračunaj sve aritmetičke sredine

$$\bar{x}_i^F = \frac{1}{r} \sum_{k=1}^r x_{k,i}^F \quad \bar{v} = \frac{1}{r} \sum_{k=1}^r v_k \quad \bar{y}_i^F = \frac{1}{r} \sum_{k=1}^r \mathcal{H}_i(x_{k,i}^F)$$

4: izračunaj normalizirane matrice ansambla

$$[X^F]_{k,i} = \frac{x_{k,i}^F - \bar{x}_i^F}{\sqrt{r-1}} \quad [L^F]_{k,i} = \frac{\mathcal{H}_i(x_{k,i}^F) - \bar{y}_i^F - (v_k - \bar{v})}{\sqrt{r-1}}$$

5: izračunaj aproksimativnu Kalmanovu matricu:

$$K_i = X_i^F (L_i^F)^T (L_i^F (L_i^F)^T)^{-1}$$

6: korak analize:

$$x_{k,i}^A = x_{k,i}^F + K_i (y_{k,i} - \mathcal{H}_i(x_{k,i}^F)), k = 1, \dots, r$$

7: korak predviđanja:

$$x_{k,i+1}^F = \mathcal{M}_i(x_{k,i}^A), k = 1, \dots, r$$

8: **end for****Output:** Vektori  $x_0^A, \dots, x_N^A$  te matrice  $P_0^A, \dots, P_N^A$ Vektori  $x_1^F, \dots, x_{N+1}^F$  te matrice  $P_1^F, \dots, P_{N+1}^F$ 

gdje smo u (1) uvrstili  $L^F \stackrel{\text{def.}}{=} \mathcal{H}X^F$ , a u (2)  $W \stackrel{\text{def.}}{=} I_r + (L^F)^T R^{-1} L^F$ . Primijetimo da su matrice u posljednjem izrazu reda  $r$  što smo htjeli i postići zbog računanja. Kovarijacijsku matricu  $X^A$  računamo kao

$$X^A = X^F (I_r + (L^F)^T R^{-1} L^F)^{-1/2} V = X^F W^{-1/2} V,$$

gdje je  $V$  proizvoljna ortogonalna matrica za koju vrijedi  $V\mathbf{1} = \mathbf{1}$  (na taj način je i dalje ispunjen uvjet  $X^A\mathbf{1} = 0$  koji opravdava da su perturbacije centrirane u  $x^A$ ). Ova formula dolazi iz *Reduced rank square root filter (RRSQRT)* algoritma. Računamo posteriorne članove ansambla

$$x_k^A = x^A + \sqrt{r-1} X^F [W^{1/2} V]_k = \bar{x}^F + X^F (w^A + \sqrt{r-1} [W^{1/2} V]_k).$$

Primijetimo da su sve invertirane i korijenovane matrice malih dimenzija  $r \times r$ , a upravo to smo i htjeli postići. Osim toga, operatori  $\mathcal{M}_i$  i  $\mathcal{H}_i$  se primjenjuju samo  $r$  puta. Istaknimo da je mana algoritama koji koriste ovakve potprostore zadržavanje inkrementa unutar prostora razapetog stupcima matrice reda  $r$ . Međutim, postoje rješenja tog problema koja koriste lokalizaciju i mogu se pronaći npr. u [14].

---

**Algoritam 7** Ensemble Transform Kalman Filter (ETKF)
 

---

**Input:** Kovarijacijske matrice grešaka  $R_i$

Rutine za primjenu operatora modela i opažanja  $\mathcal{M}_i$  te  $\mathcal{H}_i$ ,  $i = 0, \dots, N$

Opažanja  $y_i$ ,  $i = 0, \dots, N$

Ortogonalna matrica  $V \in \mathbb{R}^{r \times r}$  koja zadovoljava  $V\mathbf{1} = \mathbf{1}$

Početna stanja  $x_k^F$ ,  $k = 1, \dots, r$  za  $i = 0$ . Definiramo  $E^F = [x_1^F, \dots, x_r^F]$

- 1: **for**  $i=0, \dots, N$
  - 2: izračunaj sredinu i matricu  $X^F$   

$$\bar{x}^F = \frac{1}{r} E^F \quad X^F = \frac{1}{\sqrt{r-1}} (E^F - \bar{x}^F \mathbf{1}^T)$$
  - 3: izračunaj sredinu opažanja  

$$\bar{y} = \frac{1}{r} Y \mathbf{1}, \text{ gdje je } Y = \mathcal{H}_i(E^F)$$
  - 4: izračunaj normalizirani ansambl opažanja  

$$\tilde{L}^F = \frac{1}{\sqrt{r-1}} R_i^{-1/2} (Y - \bar{y} \mathbf{1}^T)$$
  - 5: izračunaj normalizirani vektor inovacije  $d = R_i^{-1/2} (y_i - \bar{y})$  i matricu  $W$   

$$W = (I_r + (\tilde{L}^F)^T \tilde{L}^F)^{-1}$$
  - 6: izračunaj vektor koeficijenata  $w^A$   

$$w^A = W (\tilde{L}^F)^T d$$
  - 7: izračunaj ansambl analize  

$$E^A = \bar{x}^F \mathbf{1}^T + X^F (w^A \mathbf{1}^T + \sqrt{r-1} W^{1/2} V)$$
  - 8: izračunaj ansambl predviđanja  

$$E^F = \mathcal{M}_i(E^A)$$
  - 9: **end for**
-

# Dodatak A

## MATLAB kodovi

U ovom dijelu nalaze se MATLAB kodovi koji rekonstruiraju grafove iz cijelog rada.

### A.1 Lorenzovi atraktori

```
t = 0:0.01:20;
sigma = 10;
beta = 8/3;
rho = 28;

x0 = [5 5 5]; % egzaktni pocetni uvjet
[t, x_egz] = ode45('lor_rhs', t, x0, [], sigma, beta, rho);

x_true = x_egz(:,1);
y_true = x_egz(:,2);
z_true = x_egz(:,3);

% perturbirani pocetni uvjeti
sigma_pert = 1;
xic = x0 + sigma_pert * randn(1,3); % perturbirani x0

%%
% opazanja kreirana kao perturbirane egzaktne vrijednosti
```

```

t_opazanja = t(1:25:end);
n = length(t_opazanja);

sigma_error = 4;
xdata = x_true(1:25:end) + sigma_error * randn(n,1);
ydata = y_true(1:25:end) + sigma_error * randn(n,1);
zdata = z_true(1:25:end) + sigma_error * randn(n,1);
% data vs the truth for x
figure(1), plot(t_opazanja, xdata, 'ro', t,
    x_true, 'k', 'linewidth', [2])
hold;grid;xlabel('iteracija');
legend('opažanje', 'stvarna vrijednost');
title('Usporedba opažanja i stvarne vrijednosti prve koordinate')

%% Kalman filter

x_kalman = []; % rjesenje iz asimilacije podataka
for j = 1:length(t_opazanja) - 1
    tspan = 0:0.01:0.25;
    [tspan, x_sol] = ode45('lor_rhs', tspan,
        xic, [], sigma, beta, rho);
    xic0 = [x_sol(end,1); x_sol(end, 2); x_sol(end, 3)];
    xdat = [xdata(j+1); ydata(j+1); zdata(j+1)]; % mjerjenje
    K = sigma_pert / (sigma_pert + sigma_error);
    xic = xic0 + (K * (xdat - xic0));
    x_kalman = [x_kalman; x_sol(1:end-1, :)];
end

x_kalman = [x_kalman; x_sol(end, :)];

figure(2), plot(t, x_true, 'm', t, x_kalman(:,1), 'g',
    'linewidth', [2])

```

```

figure(3), plot3(x_true, y_true, z_true, '-m', 'linewidth', [4]);
hold on
plot3(x_kalman(:,1), x_kalman(:,2), x_kalman(:,3), '-g',
      'linewidth', [4])
plot3(xdata, ydata, zdata, 'ro', 'linewidth', [4])
grid on
legend('egzaktno rješenje', 'rješenje dobiveno asimilacijom podataka',
      'mjerenja')

figure(4), plot(t, x_true, 'm', t, x_kalman(:,1), 'g',
      t_opazanja, xdata, 'ro', 'linewidth', [3])
legend('egzaktno rješenje', 'rješenje dobiveno asimilacijom podataka',
      'mjerenja')

```

## A.2 3D-Var metoda

```

J=1e3; % broj koraka
r=4;
gamma=1e-1;% korijen varijance greske opazanja
sigma=0;% korijen varijance greske modela
sd=10;rng(sd);

m=zeros(J,1);v=m;y=m;
v(1)=rand;% inicijalna "istina" u [0,1]
m(1)=rand;% inicijalno predvidanje u [0,1]
eta=2e-1;
C=gamma^2/eta; % kovarijanca
H=1;
K=(C*H')/(H*C*H'+gamma^2); % Kalmanova matrica (u 1D slucaju broj)

%% metoda

for j=1:J

```

```

v(j+1)=r*v(j)*(1-v(j)) + sigma*randn; % istina
y(j)=H*v(j+1)+gamma*randn; % opazanje

m_p=r*m(j)*(1-m(j));

d=y(j)-H*m_p; % inovacija
m(j+1)=m_p+K*d; % update
end

n_plot=20;
figure(1)
plot([0:n_plot-1],v(1:n_plot), 'linewidth', [2]);
hold on;
plot([0:n_plot-1],m(1:n_plot),'m', 'linewidth', [2]);
plot([1:n_plot-1],y(1:n_plot-1),'kx');
hold;grid;xlabel('iteracija');
legend('predviđanje', 'egzaktno rješenje', 'opažanje');
title('3DVAR')

```

### A.3 Kalmanovi filteri (položaj i brzina objekta)

```

J=1e2; % broj koraka
N=2;
I=eye(N);
gamma=0.5;
sigma=1;
C0=eye(N); % apriorna varijanca
m0=[0;7];
rng('default');
sd=10;rng(sd); % random number seed

T = 0.1;
M = [1, T; 0, 1];

```

```

sigma_q = 0.1;
Q = [T^4 / 4, T^3/2; T^3/2, T^2] * sigma_q^2;

m=zeros(N,J);v=m;y=zeros(J,1);c=zeros(N,N,J);
v(:,1)=m0+sqrtm(C0)*randn(N,1);
m(:,1)=10*randn(N,1);
c(:,:,1)=100*C0; % inicijalna kovarijanca
H=[1,0]; % operator opazanja

%% asimilacija

for j=1:J
    mu = [0,0];
    w = mvnrnd(mu,Q,1) '

    v(:,j+1)=M*v(:,j) + sigma*w; % stvarna vrijednost
    y(j)=H*v(:,j+1)+gamma*randn; % opazanje

    m_p=M*m(:,j); % predikcija
    c_p=M*c(:,:,j)*M'+sigma^2*I; % predikcija kovarijance

    d=y(j)-H*m_p;
    K=(c_p*H')/(H*c_p*H'+gamma^2); % Kalmanova matrica
    m(:,j+1)=m_p+K*d;
    c(:,:,j+1)=(I-K*H)*c_p;
end

n_plot = 21;
figure;plot([0:n_plot-1],v(1,1:n_plot));hold;
plot([0:n_plot-1],m(1,1:n_plot), 'm');
plot([0:n_plot-1], y(1:n_plot), 'ro');
plot([0:n_plot-1],m(1,1:n_plot)+reshape(sqrt(c(1,1,1:n_plot)),
    1,n_plot), 'r--');

```



```
plot([0:n_plot-1],m(1,1:n_plot)-reshape(sqrt(c(1,1,1:n_plot)),
    1,n_plot),'r--');
hold;grid;xlabel('iteracija');
title('Položaj objekta');
xlim([0 n_plot]);
legend('stvarna vrijednost', 'predviđanje', 'opažanje');

figure;plot([0:n_plot-1],v(2,1:n_plot));hold;
plot([0:n_plot-1],m(2,1:n_plot),'m');
plot([0:n_plot-1],m(2,1:n_plot)+reshape(sqrt(c(2,2,1:n_plot)),
    1,n_plot),'r--');
plot([0:n_plot-1],m(2,1:n_plot)-reshape(sqrt(c(2,2,1:n_plot)),
    1,n_plot),'r--');
hold;grid;xlabel('iteracija');
title('Brzina objekta');
xlim([0 n_plot]);
legend('stvarna vrijednost', 'predviđanje');

figure;plot([0:J],sum((v-m).^2));
hold;xlabel('iteracija');
title('Greška');
```

# Bibliografija

- [1] Amos S. Lawless Alison M. Fowler, *An Idealized Study of Coupled Atmosphere-Ocean 4D-Var in the Presence of Model Error*, (2016).
- [2] Amit Apte, *An Introduction to Data Assimilation*, str. 31–42, siječanj 2015, ISBN 978-81-322-2546-1.
- [3] Ross N. Bannister, *Elementary 4D-Var*, (2007).
- [4] Dale Barker, Xiang Yu Huang, Z. Liu, Thomas Auligné, Xiaoyan Zhang, Steven Rugg, Raji Ajjaji, Al Bourgeois, John Bray, Yongsheng Chen, Meral Demirtaş i Yong Run Guo, *The Weather Research and Forecasting Model's Community Variational/Ensemble Data Assimilation System: WRFDA*, Bulletin of The American Meteorological Society - BULL AMER METEOROL SOC **93** (2012), 831–843.
- [5] Jose Bernardo i Adrian Smith, *Bayesian Theory*, sv. 15, siječanj 2000, ISBN 0 471 49464 X.
- [6] Marc Bocquet, H. Elbern, Henk Eskes, Marcus Hirtl, Gregory Carmichael, Johannes Flemming, Antje Inness, Mariusz Pagowski, Juan Camaño, Pablo Saide, Roberto García, Mikhail Sofiev, J. Vira, Alexander Baklanov, Claudio Carnevale, Georg Grell i Christian Seigneur, *Data assimilation in atmospheric chemistry models: Current status and future prospects for coupled chemistry meteorology models*, Atmospheric Chemistry and Physics **15** (2015), 5325–5358.
- [7] A. Carrassi, Marc Bocquet, Laurent Bertino i Geir Evensen, *Data Assimilation in the Geosciences - An overview on methods, issues and perspectives*, Wiley Interdisciplinary Reviews: Climate Change **9** (2017).

- [8] Ralf Engbert, Maximilian Rabe, Reinhold Kliegl i Sebastian Reich, *Sequential Data Assimilation of the Stochastic SEIR Epidemic Model for Regional COVID-19 Dynamics*, *Bulletin of Mathematical Biology* **83** (2021).
- [9] Geir Evensen, *Data Assimilation: The Ensemble Kalman Filter*, Springer-Verlag, Berlin, Heidelberg, 2006, ISBN 354038300X.
- [10] M. Freitag, *Numerical linear algebra in data assimilation*, (2020).
- [11] Nils Gustafsson, *Discussion on ‘4D-Var or EnKF?’*, *Tellus A* **59** (2008), 774 – 777.
- [12] Nicholas J. Higham, *Functions of Matrices*, Society for Industrial and Applied Mathematics, 2008, <https://locus.siam.org/doi/abs/10.1137/1.9780898717778>.
- [13] K. C. Zygalakis K. J. H. Law, A. M. Stuart, *Data Assimilation: A Mathematical Introduction. Texts in Applied Mathematics*, Springer, 2015.
- [14] M. Nodet M. Asch, M. Bocquet, *Data Assimilation: Methods, algorithms and applications*, SIAM, 2016.
- [15] Philip Nadler, Shuo Wang, Rossella Arcucci, Xian Yang i Yike Guo, *An epidemiological modelling approach for COVID-19 via data assimilation*, *European Journal of Epidemiology* **35** (2020), br. 8, 749–761, ISSN 1573-7284, <http://dx.doi.org/10.1007/s10654-020-00676-7>.
- [16] A. Hollingsworth P. Courtier, J.-N. Thépaut, *A strategy for operational implementation of 4D-Var, using an incremental approach*, (1994), 1367–1387.
- [17] Kenshi Saho, *Kalman Filter for Moving Object Tracking: Performance Analysis and Filter Design*, (2018).
- [18] Yannick Tremolet, *Incremental 4D-Var Convergence Study*, (2005).

# Sažetak

Asimilacija podataka je skup matematičkih metoda usklađivanja modela s podacima. Metode obuhvaćene ovim radom mogu se podijeliti u dvije kategorije: varijacijske te sekvencijalne. Varijacijske metode minimiziraju funkciju cilja koja mjeri točnost predikcije u odnosu na model i opažanja, unutar određenog vremenskog intervala. Dva najpoznatija algoritma su *4D-Var* algoritam te *3D-Var* kao njegov poseban slučaj. Inkrementalni *4D-Var* algoritam umjesto nelinearne funkcije cilja minimizira niz linearnih konveksnih funkcija. Sekvencijalne metode asimilacije podataka opažanja uvažavaju čim ona postanu dostupna. Jedan takav algoritam je Kalmanov filter, u kojem se pretpostavlja da su operatori korišteni u definiciji modela linearni, a greške normalno distribuirane. Ovo ograničenje rješava se u *Extended Kalman filter* algoritmu koji za neke korake koristi linearizaciju opratora. Osim toga, zbog numeričkih performansi, matrice korištene u metodi mogu se aproksimirati matricama manjeg ranga te tako dolazimo do sasvim novog skupa metoda. U ovom radu predstavljena je i na praktičnim primjerima ilustrirana samo nekolicina iz ogromnog skupa algoritama za asimilaciju podataka.

# Summary

Data assimilation is a group of mathematical methods aimed at adjusting a model of interest with available data. The methods covered in this thesis can be divided into two categories: variational and sequential methods. Variational methods minimize the cost function that measures the accuracy of the prediction compared to the model and observations, in a specified time interval. Two most commonly used algorithms are *4D-Var* algorithm and *3D-Var* as its special case. Incremental *4D-Var* algorithm minimizes a sequence of linear convex cost functions, instead of a nonlinear one as in *4D-Var*. Sequential methods for data assimilation make use of observations as soon as they become available. One such sequential algorithm is the *Kalman filter*, in which the operators used in the model definition are assumed to be linear, and errors are assumed to be normally distributed. These limitations are dropped in *Extended Kalman filter* algorithm which uses linearizations of those operators for some steps. Furthermore, the matrices used in the method can be replaced by their low-rank approximations, for better numerical performance. That leads to a completely new set of methods. In this thesis only a small part of a rather large set of algorithms for data assimilation is introduced and illustrated on practical examples.

# Životopis

Iskra Gašparić rođena je 20.8.1998. u Varaždinu, Hrvatska. Nakon što je završila prirodoslovno - matematičku gimnaziju u Prvoj gimnaziji u Varaždinu, 2016. upisuje preddiplomski studij na Prirodoslovno-matematičkom fakultetu u Zagrebu, smjer Matematika; istraživački. Zatim, 2019. upisuje diplomski studij Primijenjene matematike na istom odsjeku.