

Moderne metode asimilacije podataka - numeričke implementacije i primjene

Bilušković, Juraj

Master's thesis / Diplomski rad

2023

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:217:726303>

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

Download date / Datum preuzimanja: **2024-07-09**



Repository / Repozitorij:

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



SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO–MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

Juraj Bilušковиć

MODERNE METODE ASIMILACIJE
PODATAKA - NUMERIČKE
IMPLEMENTACIJE I PRIMJENE

Diplomski rad

Voditelj rada:
prof. dr. sc. Zlatko Drmač

Zagreb, srpanj, 2023.

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. _____

Najveća hvala dragom Bogu na svim blagoslovima i darovima. Zahvaljujem svojoj obitelji na stalnoj podršci tokom studiranja. Zahvaljujem mentoru, prof. dr. dc. Zlatku Drmaču na pomoći pri radu, pristupačnosti i savjetima. Posvećujem ovaj rad svojoj obitelji i svim nastavnicima i profesorima iz matematike koje sam imao kroz svoje obrazovanje.

Sadržaj

Sadržaj	iv
Uvod	2
1 Teorijski uvod	3
1.1 Oznake	3
1.2 Vjerojatnost	4
1.3 Linearna algebra	9
1.4 Dinamički sustavi	10
1.5 Numeričko rješavanje inicijalnog problema za ODJ	12
2 Asimilacija podataka	15
2.1 Matematički okvir	15
2.2 Sekvencijalna asimilacija podataka	17
2.3 Kvaliteta algoritama asimilacije podataka	18
3 Algoritmi sekvencijalne asimilacije podataka	21
3.1 Primjeri dinamičkih sustava	21
3.2 Kalmanov filter	25
3.3 3DVAR i ExKF	29
3.4 Algoritmi testirani na primjerima	31
4 Epidemiološki modeli	37
4.1 SIR model	37
4.2 SEIR model	39
4.3 SEIRD model	40
5 Sekvencijalna asimilacija podataka na COVID-19 podacima	43
5.1 COVID-19 podaci	43
5.2 Jednodnevne predikcije	45

<i>SADRŽAJ</i>	v
5.3 Predikcije većih valova širenja zaraze	53
Bibliografija	59
Octave kodovi	67

Uvod

Glavni izazov matematičkih znanosti u 21. stoljeću je brza integracija sve većih skupova podataka i matematičkih modela. Velike baze prikupljenih podataka postaju dostupne u sve više područja iz znanosti, inženjerstva i tehnologije. Matematički modeli koji opisuju sustave i metode trebaju godine, ako ne i stoljeća razvoja. Proces kombiniranja matematičkog modela i podataka dobivenih iz odabranog sustava kako bi dobili bolji i informativniji sustav se zove asimilacija podataka.

Asimilacija podataka svoje korijene u istraživanjima vuče iz područja numeričke predikcije vremena i modeliranja oceana. Danas ima sve više primjena u geološkim znanostima, ekologiji, biologiji, kemiji, inženjerstvu, medicini, podatkovnim znanostima, strojnom učenju, pa čak i u društvenim znanostima. To je dovelo do razvoja mnogih algoritamskih pristupa i velikog broja algoritamskih inovacija.

Asimilacija podataka je metoda koja kombinira opservacije stanja sustava s rezultatima modela kako bi se poboljšala procjena stanja sustava pa samim time i rezultati modela. Model je često reprezentiran sustavom parcijalnih ili običnih diferencijalnih jednadžbi.

Model sustava čine matematičke jednadžbe, često parcijalne diferencijalne jednadžbe koje opisuju procese iz fizike, biologije, kemije, Većinom te jednadžbe pojednostavljuju dinamiku sustava izbacujući procese koji su nevažni, tj. nemaju prevelik utjecaj na sustav ili su prekomplikirani da bi ih se modeliralo. Štoviše, parametri korišteni u modelu mogu biti tek djelomično poznati ili potpuno nepoznati. Matematički model kroz diskretizaciju diferencijalnih jednadžbi rezultira greškama u procesu. Čak i kada bismo sustav mogli savršeno modelirati, često početni i/ili rubni uvjeti nisu poznati sa zadovoljavajućom točnošću. Uza sve greške i probleme koji nastaju u samoj konstrukciji matematičkog modela podaci koje dobivamo dolaze s greškama mjerenja. Stoga vidimo kako je asimilacija podataka inverzni problem koji koristi pogrešne i nepotpune podatke kako bi opisao sustav čiji model je i sam netočan te ga djelomično opisuje.

Većina je algoritama za asimilaciju podataka razvijena u 20. stoljeću. Kako su podaci sve više dostupni i sve su veće baze podataka, problemi koji se trebaju riješiti sve su kompleksniji i pojavljuju se u sve višim dimenzijama.

Razlikujemo varijacijske i sekvencijalne metode asimilacije podataka. Ovaj rad će se primarno fokusirati na sekvencijalne metode, tj. na filtriranje.

Kao primarna literatura korištena je knjiga [5] te su poglavlja 2 i 3 rađena po njoj. Kao suplement za bolje i dublje razumijevanje korišten je članak [4].

Poglavlje 1 iznosi matematičku teoriju i osnovne pojmove potrebne za izgradnju matematičkog okvira asimilacije podataka i za njeno razumijevanje. Ključni su Bayesova formula i dinamički sustavi.

U poglavlju 2 uvodi se diskretni matematički okvir asimilacije podataka, osnovne komponente sekvencijalne asimilacije te se diskutira o tome kako procijeniti kvalitetu asimilacije i asimilacijskih algoritama.

Poglavlje 3 iznosi primjere dinamičkih sustava i odabrane algoritme sekvencijalne asimilacije podatka. Obradeni su Kalmanov filter (KF), 3DVAR i Extended Kalman Filter (ExKF) te su testirani na navedenim primjerima.

Matematički modeli širenja infekcija koji će biti potrebni za primjenu znanja iz asimilacije podataka na podacima COVID-19 zaraze obradeni su u poglavlju 4. Fokus je na klasifikacijskim modelima širenja zaraze.

U zadnjem poglavlju 5 donose se rezultati predikcije širenja zaraze COVID-19 dobiveni korištenjem teorije i algoritama prikazanih u poglavljima 2 i 3 te epidemioloških modela iz poglavlja 4.

Cilj ovog rada bio je primijeniti teoriju asimilacije podataka i matematičkog modeliranja širenja infekcija na jednom stvarnom problemu. COVID-19 pandemija izvrstan je problem zbog svoje velike i lako dostupne baze podataka. Predviđanje naglih širenja valova određene infekcije potencijalno može spasiti puno života stoga nije potrebno naglasiti važnost takvih predikcija.

Poglavlje 1

Teorijski uvod

U ovom poglavlju iznosimo matematičku teoriju potrebnu za izgradnju matematičkih modela i algoritama asimilacije podataka. Poglavlje je konstruirano po uzoru na [5, pogl. 1]. Prvotno se iznose definicije i rezultati iz teorije vjerojatnosti koja je esencijalna jer se na cijeli koncept asimilacije podataka može gledati kao Bayesovski inverzni problem. Kao izvor za teoriju vjerojatnosti korištena je knjiga [6]. Teorija dinamičkih sustava je također važna zbog koncepta kretanja modela i opservacija ([5, pogl. 1.2]). Mnogi modeli zadani su kao sustavi običnih (ili parcijalnih) diferencijalnih jednačbi (ODJ) te stoga iznosimo dvije jednokoračne metode rješavanja inicijalnog problema za sustav ODJ-a. Za posljednje potpoglavlje 1.5 je kao izvor korištena knjiga [1].

1.1 Oznake

\mathbb{N}	$\{1, 2, 3, \dots\}$
\mathbb{N}_0	$\{0, 1, 2, 3, \dots\}$
\mathbb{R}	$(-\infty, \infty)$; skup realnih brojeva
\mathbb{R}_+	$[0, \infty)$; skup nenegativnih realnih brojeva
\mathbb{R}^n	skup realnih n -torki
\otimes	Kroneckerov produkt
$\ \cdot\ $	Eukidska norma na \mathbb{R}^n
$\ \cdot\ _A = \ A^{-\frac{1}{2}} \cdot\ $	norma inducirana s A (A pozitivno definitna matrica)
$C(\mathbb{R}^n, \mathbb{R}^m)$	skup neprekidnih funkcija iz \mathbb{R}^n u \mathbb{R}^m
$D\psi(x)$	Jacobijeva matrica vektorske funkcije ψ u točki x
\propto	proporcionalno

1.2 Vjerojatnost

U ovom potpoglavlju iznosimo osnovne pojmove iz vjerojatnosti. Prvo definiramo vjerojatnosni prostor i navodimo koji točno mi koristimo. Potom uvodimo pojmove slučajnih varijabli i vektora, pa zatim govorimo što znači da su dvije slučajne varijable nezavisne te iznosimo Bayesovu formulu s osnovnom terminologijom koju ona donosi.

Vjerojatnosni prostor

Pokus je svaka dobro definirana procedura. Rezultati (ili pojave) pokusa nazivaju se ishodi (ili elementarni događaji). Skup svih ishoda pokusa zove se prostor elementarnih događaja i tradicionalno se označava s Ω . Neformalno, događaj je podskup prostora elementarnih događaja Ω .

Definicija 1.2.1. *Neka je Ω neprazan skup. Familija podskupova \mathcal{F} od Ω zove se σ -algebra ako vrijede sljedeća tri svojstva:*

1. $\Omega \in \mathcal{F}$.
2. Ako je $A \in \mathcal{F}$, onda je i $A^c \in \mathcal{F}$.
3. Ako su $A_j \in \mathcal{F}$, $j \in \mathbb{N}$, onda je i $\cup_{j=1}^{\infty} A_j \in \mathcal{F}$.

Uređeni par (Ω, \mathcal{F}) zove se izmjerivi prostor.

Skup elementarnih događaja bit će \mathbb{R}^n , a za izlaganje teme ovog rada ćemo trebati dovoljno bogatu familiju podskupova kako bi mogli računati očekivanja.

S \mathcal{B} označimo σ -algebru generiranu familijom svih otvorenih skupova na \mathbb{R} . Nju zovemo σ -algebra Borelovih skupova na \mathbb{R} , a elemente σ -algebre \mathcal{B} zovemo Borelovi skupovi. Iz definicije slijedi da je svaki otvoreni interval (a, b) , $a, b \in \mathbb{R}$ Borelov skup. Isto tako je i svaki zatvoreni interval $[a, b]$ Borelov skup kao komplement te su i intervali oblika $(a, b]$ i $[a, b)$ također Borelovi skupovi.

Dovoljno bogata familija podskupova potrebna za ovaj rad je upravo Borelova σ -algebra.

Definicija 1.2.2. *Neka je Ω neprazan skup i \mathcal{F} σ -algebra događaja. Vjerojatnost na izmjerivom prostoru (Ω, \mathcal{F}) je funkcija $\mathbb{P} : \mathcal{F} \rightarrow [0, 1]$ koja zadovoljava sljedeća tri aksioma:*

1. $\mathbb{P}(A) \geq 0$, $\forall A \in \mathcal{F}$.
2. $\mathbb{P}(\Omega) = 1$.

3. Za svaki niz $(A_j)_{j \in \mathbb{N}}$ po parovima disjunktних događaja $A_j \in \mathcal{F}$ ($A_i \cap A_j = \emptyset$ za $i \neq j$) vrijedi

$$\mathbb{P}\left(\bigcup_{j=1}^{\infty} A_j\right) = \sum_{j=1}^{\infty} \mathbb{P}(A_j).$$

Uređena trojka $(\Omega, \mathcal{F}, \mathbb{P})$ zove se vjerojatnosni prostor.

Slučajne varijable i vektori

Definicija 1.2.3. Neka je $(\Omega, \mathcal{F}, \mathbb{P})$ vjerojatnosni prostor. Funkcija $X : \Omega \rightarrow \mathbb{R}$ je slučajna varijabla (na Ω) ako je $X^{-1}(B) \in \mathcal{F}$ za proizvoljno $B \in \mathcal{B}$, tj. $X^{-1}(\mathcal{B}) \subseteq \mathcal{F}$.

U daljnjem tekstu slučajne varijable označavat ćemo velikim slovima latinice (npr. X, Y, Z, \dots). U teoriji mjere slučajna varijabla je realna Borel-izmjeriva funkcija, tj. funkcija sa Ω u \mathbb{R} izmjeriva u paru σ -algebri $(\mathcal{F}, \mathcal{B})$.

Sada želimo poopćiti pojam slučajne varijable u više dimenzija. S \mathcal{B}^n označimo σ -algebru na \mathbb{R}^n generiranu familijom svih otvorenih podskupova od \mathbb{R}^n . \mathcal{B}^n zovemo σ -algebra Borelovih skupova na \mathbb{R}^n , a elemente od \mathcal{B}^n zovemo Borelovi skupovi (na \mathbb{R}^n).

Definicija 1.2.4. Neka je $(\Omega, \mathcal{F}, \mathbb{P})$ vjerojatnosni prostor i $X : \Omega \rightarrow \mathbb{R}^n$. Kažemo da je X n -dimenzionalan slučajni vektor (ili, kraće, slučajni vektor) (na Ω) ako je $X^{-1}(B) \in \mathcal{F}$ za svako $B \in \mathcal{B}^n$, tj. $X^{-1}(\mathcal{B}^n) \subseteq \mathcal{F}$.

Propozicija 1.2.5. Neka je $X : \Omega \rightarrow \mathbb{R}^n$, $X = (X_1, \dots, X_n)$. Tada je X slučajni vektor ako i samo ako je X_k slučajna varijabla za svaki $k = 1, 2, \dots, n$.

Dokaz. \Leftarrow : Neka je X_k slučajna varijabla za svako $k = 1, \dots, n$. Tada a svako $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ vrijedi

$$X^{-1}((-\infty, x]) = \bigcap_{k=1}^n X^{-1}((-\infty, x_k]) \in \mathcal{F}.$$

Kako znamo da je X slučajni vektor ako i samo ako

$$X^{-1}((-\infty, x]) = \{X \leq x\} \in \mathcal{F}, \quad x \in \mathbb{R}^n,$$

slijedi da je X slučajni vektor.

\Rightarrow : Neka je X slučajni vektor. Tada za proizvoljno $k = 1, \dots, n$ i $x_k \in \mathbb{R}$ vrijedi

$$X_k^{-1}((-\infty, x_k)) = X^{-1}((-\infty, b)) \in \mathcal{F},$$

gdje je $b = (\infty, \dots, \infty, x_k, \infty, \dots, \infty)$.

Kako znamo da je X_k slučajna varijabla ako i samo ako

$$X_k^{-1}((-\infty, x_k)) = \{X < x\} \in \mathcal{F}, \quad x \in \mathbb{R},$$

slijedi da je X_k slučajna varijabla. □

Iz gornje propozicije slijedi da je n -dimenzionalni slučajni vektor zapravo uređena n -torka slučajnih varijabli.

Za $B \in \mathcal{B}$ stavimo:

$$\mathbb{P}_X(B) = \mathbb{P}(X^{-1}(B)) = \mathbb{P}\{\omega \in \Omega; X(\omega) \in B\} = \mathbb{P}\{X \in B\}.$$

Gornjom relacijom definirana je funkcija $\mathbb{P}_X : \mathcal{B} \rightarrow [0, 1]$ i lako je provjeriti da je to vjerojatnosna mjera na \mathcal{B} . \mathbb{P}_X zovemo vjerojatnosna mjera inducirana s X i često ju zovemo zakon razdiobe od X .

Definicija 1.2.6. *Neka je X slučajna varijabla na Ω . Funkcija distribucije od X je funkcija $F_X : \mathbb{R} \rightarrow [0, 1]$ definirana s*

$$F_X(x) = \mathbb{P}_X((-\infty, x]) = \mathbb{P}(X^{-1}((-\infty, x])) = \mathbb{P}(X \leq x), \quad x \in \mathbb{R}.$$

Stavljat ćemo $F_X = F$ ako je jasno o kojoj se slučajnoj varijabli radi.

Definicija 1.2.7. *Neka je X slučajna varijabla na vjerojatnosnom prostoru $(\Omega, \mathcal{F}, \mathbb{P})$ i neka je F_X njezina funkcija distribucije. Kažemo da je X apsolutno neprekidna ili neprekidna slučajna varijabla ako postoji nenegativna realna Borelova funkcija $f : \mathbb{R} \rightarrow \mathbb{R}_+$ takva da je*

$$F_X(x) = \int_{-\infty}^x f(t) d\lambda(t), \quad x \in \mathbb{R}.$$

Gornji integral je Lebesgueov integral funkcije f u odnosu na Lebesgueovu mjeru λ . Ako je X neprekidna slučajna varijabla, tada se funkcija f zove funkcija gustoće vjerojatnosti od X , tj. gustoća od X i označujemo ju sa f_X ili f ako je jasno o kojoj se slučajnoj varijabli radi.

Gornju izgradnju funkcije distribucije i funkcije gustoće slučajne varijable poopćujemo i na slučajne vektore te i za njih uvodimo iste pojmove.

Za $B \in \mathcal{B}^n$ stavimo:

$$\mathbb{P}_X(B) = \mathbb{P}(X^{-1}(B)) = \mathbb{P}\{\omega \in \Omega; X(\omega) \in B\} = \mathbb{P}\{X \in B\}.$$

Gornjom relacijom definirana je funkcija $\mathbb{P}_X : \mathcal{B}^n \rightarrow [0, 1]$ i \mathbb{P}_X je vjerojatnost na \mathcal{B}^n . \mathbb{P}_X zovemo vjerojatnosna mjera inducirana slučajnim vektorom X ili zakon razdiobe slučajnog vektora X .

Definicija 1.2.8. Neka je $X : \Omega \rightarrow \mathbb{R}^n$ slučajni vektor, $X = (X_1, \dots, X_n)$. Funkcija distribucije slučajnog vektora X je funkcija $F_X = F : \mathbb{R}^n \rightarrow [0, 1]$ definirana s

$$F(x) = \mathbb{P}_X((-\infty, x]) = \mathbb{P}\{X \leq x\} = \mathbb{P}\{X_1 \leq x_1, \dots, X_n \leq x_n\}, \quad x = (x_1, \dots, x_n) \in \mathbb{R}^n.$$

Definicija 1.2.9. Neka je $X = (X_1, \dots, X_n)$ n -dimenzionalni slučajni vektor na vjerojatnosnom prostoru $(\Omega, \mathcal{F}, \mathbb{P})$ i neka je F njegova funkcija distribucije.

X je apsolutno neprekidan, ili, kraće, neprekidan slučajni vektor ako postoji nenegativna Borelova funkcija $f : \mathbb{R}^n \rightarrow \mathbb{R}$ takva da je

$$F(x) = \int_{-\infty}^x f(t) d\lambda(t), \quad x \in \mathbb{R}^n.$$

Gornji integral je Lebesgueov integral i funkciju f zovemo gustoća slučajnog vektora X .

Za kraj ovog odjeljka definiramo najvažniji primjer neprekidne slučajne varijable kako za teoriju vjerojatnosti i statistiku tako i za ovaj rad.

Definicija 1.2.10. Slučajna varijabla X ima normalnu (Gaussovu) razdiobu s parametrima $\mu \in \mathbb{R}$ i $\sigma^2 > 0$ i pišemo $X \sim N(\mu, \sigma^2)$ ako joj je funkcija gustoće

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

Nezavisnost slučajnih varijabli

Općenito je nezavisnost jedan od najvažnijih pojmova u teoriji vjerojatnosti te je stoga i pogotovo važan u asimilaciji podataka.

Definicija 1.2.11. Neka su X_1, \dots, X_n slučajne varijable definirane na vjerojatnosnom prostoru $(\Omega, \mathcal{F}, \mathbb{P})$. Kažemo da su X_1, \dots, X_n nezavisne ako za proizvoljne $B_i \in \mathcal{B}$, $i = 1, \dots, n$ vrijedi

$$\mathbb{P}\{X_1 \in B_1, \dots, X_n \in B_n\} = \prod_{i=1}^n \mathbb{P}\{X_i \in B_i\}.$$

Nezavisnost možemo u slučaju neprekidnih slučajnih varijabli karakterizirati pomoću pripadnih funkcija gustoće.

Teorem 1.2.12. Ako slučajni vektor $X = (X_1, \dots, X_n)$ ima gustoću f , tada svaka slučajna varijabla X_i ima gustoću f_i , $i = 1, \dots, n$. Osim toga, u tom slučaju X_1, \dots, X_n su nezavisne ako i samo ako je

$$f(x_1, \dots, x_n) = f_1(x_1) \cdots f_n(x_n),$$

za sve $(x_1, \dots, x_n) \in \mathbb{R}^n$, osim eventualno Borelovog podskupa od \mathbb{R}^n Lebesgueove mjere nula.

Iz dokaza teorema 1.2.12, koji se može naći u [6], direktno slijedi

Korolar 1.2.13. *Ako su X_1, \dots, X_n nezavisne i X_i ima gustoću f_i ($i = 1, \dots, n$), tada $X = (X_1, \dots, X_n)$ ima gustoću danu sa*

$$f(x_1, \dots, x_n) = f_1(x_1) \cdots f_n(x_n), \quad (x_1, \dots, x_n) \in \mathbb{R}^n.$$

Općenito ako X_1, \dots, X_n nisu nezavisne, iz egzistencije gustoća od X_i za $i = 1, \dots, n$ ne slijedi egzistencija gustoće za $X = (X_1, \dots, X_n)$.

Uvjetna vjerojatnost

Vjerojatnost događaja A ako se dogodio događaj B naziva se uvjetna vjerojatnost od A uz dano B i označava s $\mathbb{P}(A|B)$.

Definicija 1.2.14. *Neka je $(\Omega, \mathcal{F}, \mathbb{P})$ vjerojatnosni prostor te $B \in \mathcal{F}$ događaj takav da je $\mathbb{P}(B) > 0$. Uvjetna vjerojatnost događaja A uz dano B definira se formulom*

$$\mathbb{P}(A|B) := \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}.$$

Za $B \in \mathcal{F}$ takav da je $\mathbb{P}(B) > 0$ uvodimo oznaku

$$\mathbb{P}_B(A) := \mathbb{P}(A|B), \quad A \in \mathcal{F}.$$

Gornjom relacijom definirana je vjerojatnost \mathbb{P}_B na \mathcal{F} , koju zovemo uvjetna vjerojatnost uz uvjet B .

Definicija 1.2.15. *Neka je X slučajna varijabla na vjerojatnosnom prostoru $(\Omega, \mathcal{F}, \mathbb{P})$ te neka je B događaj iz \mathcal{F} takav da je $\mathbb{P}(B) > 0$. Uvjetna funkcija gustoće od X uz dano B je funkcija $f(\cdot|B) : \mathbb{R} \rightarrow [0, 1]$ definirana s*

$$f(x|B) := \mathbb{P}(X = x|B) = \mathbb{P}_B(X = x), \quad x \in \mathbb{R}.$$

Ako je (X, Y) dvodimenzionalan neprekidan slučajni vektor s gustoćom $f_{X,Y}$. Uvjetnu gustoću od X za dano $Y = y$ definiramo kao:

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x, y)}{f_Y(y)}, \quad \text{ako je } f_Y(y) > 0.$$

Uz događaj $A = \{X \in C\}$ za proizvoljni $C \in \mathcal{B}$ i uz $B = \{Y = y\}$ pokaže se da vrijedi

$$\mathbb{P}(X \in C|Y = y) = \int_C \frac{f_{X,Y}(x, y)}{f_Y(y)} dx \quad (\text{g.s.} - \mathbb{P}_Y).$$

Funkciju $\mathbb{P}(X \in C|Y = y)$, gdje je $C \in \mathcal{B}$, zovemo uvjetna razdioba od X za dano $Y = y$.

Bayesova formula

Neka su $A, B \in \mathcal{F}$ dva događaja takva da je $\mathbb{P}(A), \mathbb{P}(B) > 0$. Tada iz definicija uvjetnih vjerojatnosti vrijedi:

$$\mathbb{P}(A \cap B) = \mathbb{P}(A|B)\mathbb{P}(B),$$

$$\mathbb{P}(A \cap B) = \mathbb{P}(B|A)\mathbb{P}(A).$$

Izjednačavanjem i sređivanjem dobivamo Bayesovu formulu:

$$\mathbb{P}(A|B) = \frac{1}{\mathbb{P}(B)}\mathbb{P}(B|A)\mathbb{P}(A).$$

Neka je (X, Y) neprekidan slučajni vektor na vjerojatnosnom prostoru $(\Omega, \mathcal{F}, \mathbb{P})$. Bayesova formula u terminima od (X, Y) dobiva se na analogan način preko definicija:

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)},$$

$$f_{Y|X}(y|x) = \frac{f_{X,Y}(x,y)}{f_X(x)},$$

iz čega slijedi:

$$f_{X|Y}(x|y) = \frac{1}{f_Y(y)}f_{Y|X}(y|x)f_X(x).$$

Značaj ove formule sam po sebi nije isprva očit i jasan. Ona dobiva svoju vrijednost kada želimo odrediti uvjetnu funkciju gustoće $f_{X|Y}(x|y)$ koja se teško odredi, ali su zato $f_X(x)$ i $f_{Y|X}(y|x)$ relativno jednostavne za odrediti.

Kada se Bayesova formula koristi u statistici onda su Y tipično opservacije i prikupljeni podaci, a X je ono nepoznato o čemu želimo pronaći informacije koristeći podatke. U tom kontekstu gustoću $f_X(x)$ zovemo apriorna distribucija od X , $f_{Y|X}(y|x)$ vjerodostojnost i $f_{X|Y}(x|y)$ aposteriorna distribucija.

1.3 Linearna algebra

Prvo se podsjetimo definicije pozitivne definitnosti realnih simetričnih matrica.

Definicija 1.3.1. Neka je $A \in \mathbb{R}^{n \times n}$, $n \in \mathbb{N}$ takva da vrijedi $A^T = A$, tj. simetrična je. Matrica A je pozitivno definitna ako vrijedi $x^T A x > 0$, $\forall x \in \mathbb{R}^n \setminus \{0\}$. Koristimo oznaku $A > 0$ za pozitivno definitne matrice.

Navodimo poznati identitet koji će nam biti potreban kasnije u poglavlju 3 u dokazu korolara koji je "recept" za algoritam Kalmanovog filtera. Identitet ćemo iskazati i dokazati uz jače pretpostavke nego što su u općenitom slučaju.

Teorem 1.3.2 (Woodburyjev matični identitet). *Neka su $A \in \mathbb{R}^{p \times p}$, $U \in \mathbb{R}^{p \times q}$, $C \in \mathbb{R}^{q \times q}$ i $V \in \mathbb{R}^{q \times p}$. Ako su A i C pozitivno definitne te $V = U^T$, onda je $A + UCV$ invertibilna i*

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}.$$

(Umjesto samo zahtjeva za regularnost od A i C pretpostavili smo dodatno da su $A > 0$ i $C > 0$. Također smo pretpostavili $V = U^T$ što u općenitom slučaju nije potrebno.)

Dokaz. Prvo provjerimo jesu li sve matrice u izrazu dobro definirane.

Matrica A je pozitivno definitna pa je stoga i regularna. Analogno vrijedi za matricu C iz čega slijedi da su A^{-1} i C^{-1} dobro definirane. Također zbog pozitivne definitnosti od A i C slijedi da je $A + UCV$ pozitivno definitna i time regularna. Kako su A i C pozitivno definitne matrice onda su i njihovi inverzi A^{-1} i C^{-1} pozitivno definitni pa samim time i regularni. Iz toga slijedi da je $C^{-1} + VA^{-1}U$ pozitivno definitna i stoga regularna.

Formula se može provjeriti izravno:

$$\begin{aligned} & (A + UCV)[A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}] \\ &= \{I - U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}\} + \{UCVA^{-1} - UCVA^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}\} \\ &= \{I + UCVA^{-1}\} - \{U(C^{-1} + VA^{-1}U)^{-1}VA^{-1} + UCVA^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}\} \\ &= I + UCVA^{-1} - (U + UCVA^{-1}U)(C^{-1} + VA^{-1}U)^{-1}VA^{-1} \\ &= I + UCVA^{-1} - UC(C^{-1} + VA^{-1}U)(C^{-1} + VA^{-1}U)^{-1}VA^{-1} \\ &= I + UCVA^{-1} - UCVA^{-1} = I. \end{aligned}$$

□

1.4 Dinamički sustavi

U ovom potpoglavlju iznosimo osnovnu teoriju diskretnih dinamičkih sustava potrebnu za razumijevanje vremenski diskretne asimilacije podataka koju proučavamo u radu.

Neka je $\psi \in C(\mathbb{R}^n, \mathbb{R}^n)$. Od velike će nam važnosti biti iterirajuća preslikavanja, tj. diskretni dinamički sustavi oblika

$$v_{i+1} = \psi(v_i), \quad v_0 = u.$$

Važno je promatrati i stohastičke dinamičke sustave

$$v_{i+1} = \psi(v_i) + \xi_i, \quad v_0 = u,$$

gdje je $\xi = \{\xi_i\}_{i \in \mathbb{N}}$ niz nezavisnih i jednakodistribuiranih slučajnih varijabli na \mathbb{R}^n te je u također slučajna varijabla na \mathbb{R}^n nezavisna od ξ .

Dinamički sustavi s kontrolom

Često je korisno dodati takozvanu kontrolu $w = \{w_i\}_{i=0}^{\infty}$ u diskretni dinamički sustav čime dobivamo:

$$v_{i+1} = \psi(v_i) + w_i. \quad (1.1)$$

Svrha kontrole je da "navigira" dinamički sustav k nekom cilju. Od više mogućih i zanimljivih primjera za kontrolu mi ćemo se fokusirati na jedan koji je najrelevantniji za asimilaciju podataka.

Uz dane $y = \{y_i\}_{i \in \mathbb{N}}$ gdje je $y_i \in \mathbb{R}^m$, koji predstavljaju opservacije o modelu, i uz danu funkciju $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ biramo kontrolu w tako da je $\|y_i - h(v_i)\|$ malo. Fokusirat ćemo se stoga na kontrole oblika $w = K(y_i - h(v_i))$, pa uvrštavanjem u (1.1) slijedi:

$$v_{i+1} = \psi(v_i) + K(y_i - h(v_i)).$$

Glavno pitanje je kako odabrati K tako da osiguramo željeno svojstvo.

Primjer 1.4.1. *Uzmimo $n = m = 1$, $\psi(v) = \lambda v$ i $h(v) = v$. Pretpostavimo da su opservacije $y = \{y_i\}_{i \in \mathbb{N}}$ dobivene iz dinamičkog sustava $v^+ = \{v_i^+\}_{i \in \mathbb{N}_0}$ bez dodavanja ikakvog šuma: $y_{i+1} = v_{i+1}^+$, $\forall i \in \mathbb{N}_0$, gdje je $v_{i+1}^+ = \lambda v_i^+$, $\forall i \in \mathbb{N}_0$. Podaci y su generirani dinamičkim sustavom bez kontrole v^+ . Sada želimo preko dinamičkog sustava s kontrolom dobiti model koji pokušava aproksimirati opservacije y te samim time odrediti dinamički sustav bez kontrole v^+ koji je generirao te podatke.*

Uzmimo sljedeći dinamički sustav s kontrolom:

$$\begin{aligned} v_{i+1} &= \psi(v_i) + K(y_i - h(v_i)) \\ &= \lambda v_i + K(y_i - v_i) \end{aligned}$$

i pretpostavimo $v_0 \neq v_0^+$. Zanima nas približava li se v_i k v_i^+ kako $i \rightarrow \infty$.

Neka je $e_i = v_i - v_i^+$ greška u i -tom trenutku. Tada vrijedi:

$$\begin{aligned} e_{i+1} &= v_{i+1} - v_{i+1}^+ \\ &= \lambda v_i + K(y_i - v_i) - \lambda v_i^+ \\ &= \lambda(v_i - v_i^+) + K(v_i^+ - v_i) \\ &= (\lambda - K)e_i, \end{aligned}$$

iz čega vidimo

$$|e_{i+1}| = |\lambda - K||e_i|.$$

Ako pretpostavimo da K biramo tako da je $|\lambda - K| < 1$ slijedi da $|e_i| \rightarrow 0$, $i \rightarrow \infty$. Dinamički sustav s kontrolom v približava se k dinamičkom sustavu bez kontrole v^+ kako $i \rightarrow \infty$. Ovo je tipično za asimilacijske algoritme koje ćemo proučiti u poglavlju 3.

1.5 Numeričko rješavanje inicijalnog problema za ODJ

Ovo potpoglavlje odnosi se na proces određivanja dinamičkog sustava iz modela opisanog diferencijalnim jednačbama. Nas će interesirati samo inicijalni problem i to običnih diferencijalnih jednačbi. Navodimo i obrađujemo dvije metode koje koristimo u ovom radu. Restrikcija na samo inicijalni problem običnih diferencijalnih jednačbi je zbog prirode problema kojeg proučavamo u zadnjem poglavlju, a koji je opisan modelima iz poglavlja 4. Također, mnogi se sustavi parcijalnih diferencijalnih jednačbi svode na upravo na IP sustava ODJ-a (metoda linija).

Neka je zadan sustav diferencijalnih jednačbi koji opisuju nekakav model od interesa:

$$\begin{aligned} \frac{dv}{dt} &= f(t, v(t)), \quad t \geq t_0 \\ v(t_0) &= v_0, \end{aligned} \quad (1.2)$$

pri čemu je $f : I \times \mathbb{R}^n \rightarrow \mathbb{R}^n$, gdje je $I \subseteq \mathbb{R}$ otvoren interval, $t_0 \in I$, $n \geq 1$ i $v_0 \in \mathbb{R}^n$. Po komponentama sustav glasi

$$\begin{pmatrix} \frac{dv_1(t)}{dt} \\ \frac{dv_2(t)}{dt} \\ \dots \\ \frac{dv_n(t)}{dt} \end{pmatrix} = \begin{pmatrix} f_1(t, v_1(t), v_2(t), \dots, v_n(t)) \\ f_2(t, v_1(t), v_2(t), \dots, v_n(t)) \\ \dots \\ f_n(t, v_1(t), v_2(t), \dots, v_n(t)) \end{pmatrix}, \begin{pmatrix} v_1(t_0) \\ v_2(t_0) \\ \dots \\ v_n(t_0) \end{pmatrix} = \begin{pmatrix} (v_0)_1 \\ (v_0)_2 \\ \dots \\ (v_0)_n \end{pmatrix}.$$

Kažemo da je (1.2) inicijalni problem (IP) za sustav običnih diferencijalnih jednačbi (ODJ).

Ako je

$$f(t, x_1, \dots, x_n) = A(t) \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} + \begin{pmatrix} b_1(t) \\ b_2(t) \\ \dots \\ b_n(t) \end{pmatrix}$$

onda kažemo da je ODJ sustav linearan.

Riješiti (1.2) numerički znači izračunati aproksimacije vrijednosti od $v(t_i)$ u konačno točkaka $t_0, t_1, \dots, t_N = T$ u zadanom intervalu $[t_0, T]$. Navodimo i opisujemo dvije jednokoračne metode koje su korištene u radu: Eulerova metoda kao najjednostavnija i kompleksnija Runge-Kutta metoda reda 3.

Eulerova metoda

Ključni element numeričkog rješavanja inicijalnog problema (1.2) je kako iskoristiti diferencijalnu jednačbu i od poznate vrijednosti $v(t_0)$ dobiti što bolju aproksimaciju za $v(t_1)$ i

tako dalje $t_0 \rightarrow t_1 \rightarrow \dots$ sve do zadnje točke t_N . Primijetimo:

$$v(t_{i+1}) - v(t_i) = \int_{t_i}^{t_{i+1}} v'(\tau) d\tau = \int_{t_i}^{t_{i+1}} f(\tau, v(\tau)) d\tau, \quad \forall i = 0, 1, \dots, N-1.$$

Sada uočimo:

$$v(t_{i+1}) = v(t_i) + \int_{t_i}^{t_{i+1}} f(\tau, v(\tau)) d\tau, \quad \forall i = 0, 1, \dots, N-1. \quad (1.3)$$

Sada pokušavamo numerički aproksimirati vrijednost integrala. Kako podintegralnu funkciju poznamo samo u lijevom rubu intervala $[t_i, t_{i+1}]$ pa je najjednostavnije što možemo aproksimirati:

$$\int_{t_i}^{t_{i+1}} f(\tau, v(\tau)) d\tau \approx (t_{i+1} - t_i) f(t_i, v(t_i)). \quad (1.4)$$

Iz (1.3) i (1.4) dobivamo:

$$v(t_{i+1}) \approx v(t_i) + (t_{i+1} - t_i) f(t_i, v(t_i)).$$

Naravno, osim u $t = t_0$ kada je $v(t_0)$ poznata zadana vrijednost, sve vrijednosti $v(t_i)$ su zamijenjene aproksimacijama v_i . Diskretne vrijednosti t_0, t_1, \dots, t_N možemo birati s varijabilnim koracima $h_i = t_{i+1} - t_i$ ili, jednostavnije, ekvidistantno $h = t_{i+1} - t_i = \text{const.}$ za sve $i = 0, 1, \dots, N-1$.

Eulerova metoda je najjednostavniji primjer jednokoračne eksplicitne metode - vrijednost v_{i+1} je dobivena eksplicitnim izrazom koji koristi informaciju samo iz koraka t_i .

Runge-Kutta metoda reda 3

Želimo poboljšati jednostavnu Eulerovu metodu. Polazimo od osnovne relacije:

$$v(t_{i+1}) = v(t_i) + \int_{t_i}^{t_{i+1}} f(\tau, v(\tau)) d\tau$$

i u njoj pokušajmo numeričku vrijednost integrala aproksimirati koristeći dodatne čvorove u $[t_i, t_{i+1}]$ i dodatne slobodne parametre koje naknadno odredimo na najbolji mogući način. Za Runge-Kutta metodu reda 3 uzimamo tri dodatna čvora ξ_1, ξ_2, ξ_3 , tri dodatna parametra c_1, c_2, c_3 i potražimo formulu oblika:

$$v_{i+1} = v_i + h[c_1 f(\xi_1, v(\xi_1)) + c_2 f(\xi_2, v(\xi_2)) + c_3 f(\xi_3, v(\xi_3))], \quad (1.5)$$

pri čemu ćemo slobodne parametre iskoristiti da dobijemo što veći red konzistentnosti, tj. što je moguće manju lokalnu pogrešku diskretizacije.

Odaberimo $\xi_1 = t_i$, $\xi_2 = t_i + a_2h$ i $\xi_3 = t_i + a_3h$, gdje su $0 < a_2, a_3 < 1$. Kako nam vrijednosti $v(\xi_j)$, $j = 1, 2, 3$ nisu dostupne zamijenit ćemo ih prediktorima-aproksimacijama, ali opet sa slobodnim parametrima A_{21}, A_{31}, A_{32} :

$$\begin{aligned} v(\xi_1) &\approx v_i, \\ v(\xi_2) &\approx v_i + hA_{21}f(t_i, v_i), \\ v(\xi_3) &\approx v_i + hA_{31}f(t_i, v_i) + hA_{32}f(t_i + a_2h, v_i + hA_{21}f(t_i, v_i)). \end{aligned} \quad (1.6)$$

Iz (1.5) i (1.6) dobivamo opći oblik i -tog koraka:

$$\begin{aligned} \psi_1 &= f(t_i, v_i), \\ \psi_1 &= f(t_i + a_2h, v_i + hA_{21}\psi_1), \\ \psi_1 &= f(t_i + a_3h, v_i + h(A_{31}\psi_1 + A_{32}\psi_2)), \\ v_{i+1} &= v_i + h(c_1\psi_1 + c_2\psi_2 + c_3\psi_3). \end{aligned} \quad (1.7)$$

Sada osam slobodnih parametara treba odrediti tako da shema bude što je moguće bolja. Taj postupak ne prikazujemo jer nije od važnosti za razumijevanje teme ovoga rada, već navodimo rješenje.

Rješenje kompaktno zapisujemo u obliku

$$\begin{array}{c|ccc} a_1 & & & \\ a_2 & A_{21} & & \\ a_3 & A_{31} & A_{32} & \\ \hline & c_1 & c_2 & c_3 \end{array},$$

a za Runge-Kutta metodu reda 3 ona je dana kao

$$\begin{array}{c|ccc} 0 & & & \\ 1/2 & 1/2 & & \\ 1 & -1 & 2 & \\ \hline & 1/6 & 2/3 & 1/6 \end{array}.$$

Poglavlje 2

Asimilacija podataka

U ovome poglavlju predočujemo osnovni matematički okvir za razumijevanje asimilacije podataka. Poglavlje je napisano po uzoru na [5, pogl. 2], preciznije na potpoglavlja koja se odnose na sekvencijalnu asimilaciju podataka (u [5] nazvano filtriranje).

Cilj asimilacije podataka je inkorporirati opservacije u model kako bi dobili najprecizniju moguću procjenu stvarnog stanja fizikalnog sustava. Opservacija koju dobivamo je funkcija našega modela uz koju moramo računati i grešku opservacije. Istodobno naš model ima svoju vlastitu dinamiku koja može biti deterministička ili stohastička.

2.1 Matematički okvir

Na asimilaciju podataka možemo gledati kao na inverzni Bayesovski problem. Neka je $v \in \mathbb{R}^n$ stanje modela koje želimo procijeniti. U Bayesovskoj statistici v modeliramo kao realizaciju slučajne varijable $V : \Omega \rightarrow \mathbb{R}^n$. Ako je Θ neka druga slučajna varijabla koja sa srednjom vrijednosti 0 modelira opservacijski šum, onda opservacije možemo modelirati preko slučajne varijable Y definirane kao:

$$Y = h(V) + \Theta, \quad (2.1)$$

gdje je $h \in C(\mathbb{R}^n, \mathbb{R}^m)$. Funkcija h se još zove opservacijski operator. Za V i Θ pretpostavljamo da su nezavisne slučajne varijable. Želimo dobiti informaciju o stanju sustava v uz dane realizacije y od Y .

Ako pretpostavimo da je f_Θ funkcija gustoće od Θ te da su y i v realizacije slučajnih varijabli Y i V onda vrijedi $f_{Y|V}(y|v) = f_\Theta(y - h(v))$. U vidu Bayesovske statistike uvjetnu funkciju gustoće od $Y = y$ uz dano $V = v$ zovemo vjerodostojnost. Funkciju gustoće od V , $f_V(v)$, zovemo apriorna distribucija od V . Na kraju, Bayesovom formulom dobivamo

formulu za izračun aposteriorne distribucije:

$$f_{V|Y}(v|y) = \frac{f_{Y|V}(y|v)f_V(v)}{f_Y(y)} \propto f_{Y|V}(y|v)f_V(v), \quad (2.2)$$

gdje je $f_Y(y) = \int_{\mathbb{R}^n} f_{Y|V}(y|v)f_V(v)dv$ normalizacijska konstanta koja ovisi samo o y .

Uvedimo konkretniji i manje apstraktan matematički okvir za asimilaciju podataka u diskretnom vremenu. Neka je $t = \{t_i\}_{i \in \mathbb{N}_0}$ niz vremenskih trenutaka u kojima opažamo sustav i dobivamo opservacije sadržane u $y = \{y_i\}_{i \in \mathbb{N}_0}$. Stanja našega modela u trenucima iz t su sadržana u $v = \{v_i\}_{i \in \mathbb{N}_0}$.

Neka je $\psi \in C(\mathbb{R}^n, \mathbb{R}^n)$ funkcija dinamičkog sustava. Preslikavanje ψ često je rješenje sustava običnih ili parcijalnih diferencijalnih jednačnji. Prvotno opisujemo naš model kao dinamički sustav.

Dinamiku modela $v = \{v_i\}_{i \in \mathbb{N}_0}$, kojeg ćemo zvati **signal**, opisujemo deterministički:

$$v_{i+1} = \psi(v_i), \quad i \in \mathbb{N}_0 \quad (2.3)$$

ili stohastički:

$$v_{i+1} = \psi(v_i) + \xi_i, \quad i \in \mathbb{N}_0, \quad (2.4)$$

gdje je $\xi = \{\xi_i\}_{i \in \mathbb{N}_0}$ niz nezavisnih jednakodistribuiranih slučajnih varijabli koje predstavljaju greške (šum) modela te pretpostavljamo da je slučajna varijabla v_0 nezavisna od ξ . U nastavku rada uz pretpostavljamo $v_0 \sim N(m_0, C_0)$ te $\xi \sim N(0, \Sigma)$, $\Sigma > 0$. Primijetimo da je u slučaju korištenja determinističkog modela realizacija u trenutku t_0 , tj. v_0 , varijabla od interesa. Signal je poznat u svim trenucima kada nam je poznata realizacija od v_0 .

Za **opservacije** $y = \{y_i\}_{i \in \mathbb{N}_0}$ vrijedi:

$$y_{i+1} = h(v_{i+1}) + \theta_{i+1}, \quad i \in \mathbb{N}_0,$$

gdje je h već spomenuti opservacijski operator i $\theta = \{\theta_i\}_{i \in \mathbb{N}}$ je niz nezavisnih jednakodistribuiranih slučajnih varijabli koje modeliraju opservacijski šum u trenucima t_i . U nastavku rada uz pretpostavljamo $\theta \sim N(0, \Gamma)$, $\Gamma > 0$.

U vidu Bayesovskog pogleda s početka potpoglavlja ((2.1) i (2.2)) naš je cilj odrediti uvjetnu funkciju gustoće u svakom trenutku t_i preko sljedeće relacije:

$$f_{V|Y}(v_i|y) \propto f_{Y|V}(y|v_i)f_V(v_i),$$

gdje je $f_{Y|V}(y|v_i) = f_{\theta_i}(y - h(v_i))$. Ono što razlikuje varijacijsku od sekvencijalne asimilacije podataka je odabir opservacija koje koristimo za procjenu uvjetne funkcije gustoće. Označimo s t_N sadašnji trenutak. Varijacijska asimilacija za procjenu svake uvjetne funkcije gustoće (aposteriorne distribucije) prethodnih stanja modela koristi sve dostupne opservacije:

$$f_{V|Y}(v_i|y) = f_{V|Y}(v_i|y_1, \dots, y_N), \quad \forall i = 1, \dots, N.$$

Sekvencijalna asimilacija podataka se postavlja u sadašnjost, tj. na rub vremenskog horizonta i koristi samo opservacije iz prošlosti:

$$f_{V|Y}(v_i|y) = f_{V|Y}(v_i|y_1, \dots, y_i), \quad \forall i = 1, \dots, N.$$

Ponovimo, osnovni cilj asimilacije podataka je dobiti vrijednosti **signala** v preko informacija koje dobivamo, tj. preko **opservacija** y .

2.2 Sekvencijalna asimilacija podataka

U sekvencijalnoj asimilaciji podataka (tzv. filtriranje) korekcije procjene modela radimo kad god su nam dostupne nove opservacije, tj. opservacije uključujemo u model sekvencijalno. Pretpostavimo da se nalazimo u vremenskom trenutku t_i u kojem imamo dostupne samo opservacije prikupljene do tog vremena. Označimo skup opservacija do trenutka t_i sa $Y_i = \{y_l\}_{l=1}^i$. Uz poznati Y_i i danu apriornu distribuciju $f_V(v_i)$ tražimo ažuriranu aposteriornu distribuciju $f_{V|Y}(v_i|Y_i)$. Koristeći vjerodostojnost $f_{Y|V}(y_i|v_i)$ i Bayesovu formulu imamo

$$f_{V|Y}(v_i|Y_i) = f_{V|Y}(v_i|y_i, Y_{i-1}) \propto f_{Y|V,Y}(y_i|v_i, Y_{i-1})f_V(v_i) = f_{Y|V}(y_i|v_i)f_V(v_i).$$

Filtriranje u principu pokušava odrediti uvjetnu funkciju gustoće slučajne varijable $V|Y$ u trenutku t_i . U idealnom slučaju uspjeh ćemo odrediti cijelu funkciju gustoće $f_{V|Y}(v_i|Y_i)$ u svakom trenutku t_i . Preciznije rečeno ono pokušava sekvencijalno ažurirati funkciju gustoće kako se vremenski indeks i povećava. Ažuriranje definirano sljedećim postupkom je recept za izračunavanje $f(v_{i+1}|Y_{i+1})$ iz $f(v_i|Y_i)$. Činimo ga kroz dva koraka: **predikciju** i **analizu**. Predikcija računa preslikavanje $f(v_i|Y_i) \mapsto f(v_{i+1}|Y_i)$, a analiza $f(v_{i+1}|Y_i) \mapsto f(v_{i+1}|Y_{i+1})$ primjenom Bayesove formule.

Predikcija

Koristeći definiciju uvjetne vjerojatnosti imamo:

$$\begin{aligned} f(v_{i+1}|Y_i) &= \int_{\mathbb{R}^n} f(v_{i+1}|Y_i, v_i)f(v_i|Y_i)dv_i \\ &= \int_{\mathbb{R}^n} f(v_{i+1}|v_i)f(v_i|Y_i)dv_i. \end{aligned} \tag{2.5}$$

Uočimo da je $f(v_{i+1}|v_i, Y_i) = f(v_{i+1}|v_i)$ jer Y_i sadrži lošiju informaciju o stanju modela od v_{i+1} . Također primijetimo ako koristimo determinističku dinamiku izračun predikcije se pojednostavljuje. Tada sve što trebamo je primijeniti funkciju ψ na $f(v_i|Y_i)$.

Analiza

Koristeći Bayesovu formulu računamo:

$$\begin{aligned}
 f(v_{i+1}|Y_{i+1}) &= f(v_{i+1}|Y_i, y_{i+1}) \\
 &= \frac{f(y_{i+1}|v_{i+1}, Y_i)f(v_{i+1}|Y_i)}{f(y_{i+1}|Y_i)} \\
 &= \frac{f(y_{i+1}|v_{i+1})f(v_{i+1}|Y_i)}{f(y_{i+1}|Y_i)}.
 \end{aligned} \tag{2.6}$$

Uočimo da je $f(y_{i+1}|v_{i+1}, Y_i) = f(y_{i+1}|v_{i+1})$ jer Y_i sadrži lošiju informaciju o stanju modela od v_{i+1} .

Zajedno, predikcija i analiza, definiraju preslikavanje iz $f(v_i|Y_i)$ na $f(v_{i+1}|Y_{i+1})$.

Slika 2.1 vizualizira što zapravo radimo prilikom sekvencijalnog asimiliranja podataka te je dobar intuitivni vodič. Imamo početnu vrijednost modela koju smo procijenili ili pretpostavili. Potom računamo predikciju u kojem će stanju model biti u trenutku i (zelena boja). U trenutku i istodobno dobivamo podatak o stvarnom stanju sustava. Na slici je to opservacija označena plavom bojom. Analizom potom usklađujemo našu predikciju s opservacijom koja je aproksimativno stvarno stanje sustava (crvena boja). Nakon toga nastavljamo s predikcijom stanja modela u trenutku $i + 1$ samo nam je ovog puta početna točka pomaknuta iz predikcije za trenutak i u točku dobivenu analizom u trenutku i . Analogno nastavljamo do kraja intervala kojeg želimo asimilirati.

2.3 Kvaliteta algoritama asimilacije podataka

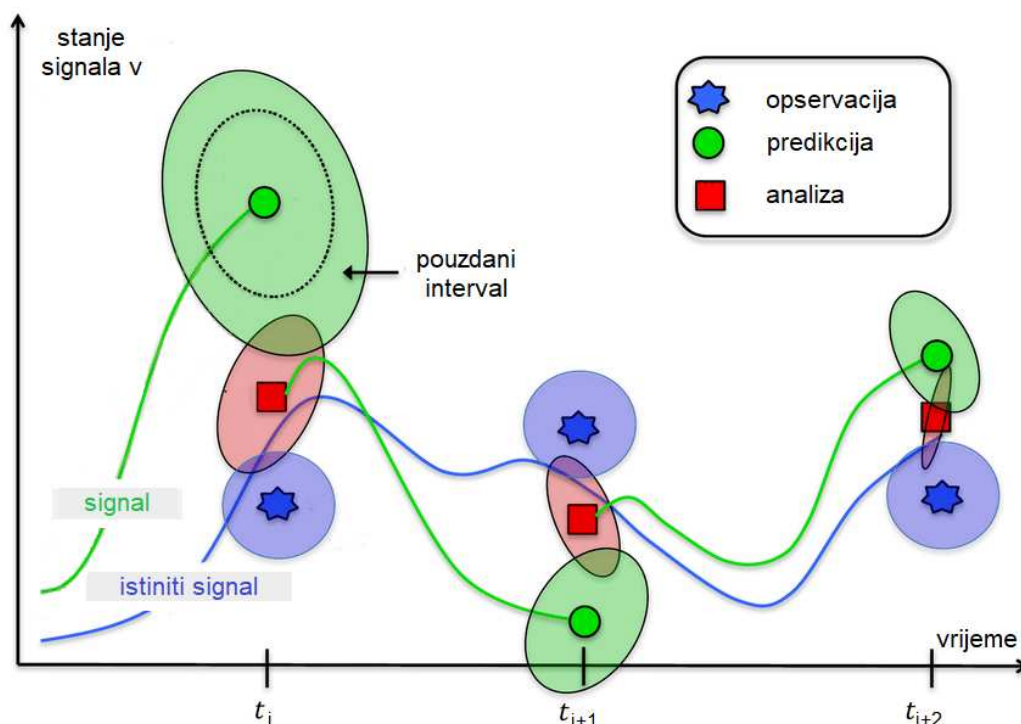
Postavljamo si dva pitanja:

1. Koliko informacija nose opservacije koje dobivamo?
2. Koliko dobro naš algoritam ekstrahira te informacije?

Kako bi razumjeli koliko kvalitetan signal dobivamo koristeći model i podatke važno je odgovoriti na ova dva pitanja. Prvo pitanje neovisno je o algoritmu kojeg koristimo te se tiče svojstava Bayesove aposteriorne distribucije.

U testiranju algoritama koristimo podatke dobivene iz nekog sustava koji nam je poznat. Taj poznati sustav ćemo zvati istiniti signal. U tom kontekstu ćemo s $y^+ = \{y_i^+\}_{i=1}^N$ označiti opservacije dobivene kao realizacije istinitog signala kojeg označavamo s $v^+ = \{v_i^+\}_{i=0}^N$.

Aposteriorna konzistencija je svojstvo konvergencije aposteriorne distribucije $f(v|y^+)$ k istinitoj i pravoj distribuciji kada $N \rightarrow \infty$ (velika baza podataka, tj. opservacija) ili kada $|\Gamma| \rightarrow 0$ (mali šum). Za problem koji ima aposteriornu konzistenciju također kažemo da ima Bayesovu aposteriornu konzistenciju. I tek kada je ovo svojstvo prisutno možemo



Slika 2.1: Grafički opis sekvencijalne asimilacije podataka.

nas zanimati i kojom brzinom ono konvergira k pravoj. Ovo pitanje neovisno je o izboru algoritma te se tiče isključivo kvalitete i količine dostupnih podataka. Kada govorimo o sekvencijalnoj asimilaciji ovo pitanje se tiče marginalne distribucije. Zanima nas konvergira li aposteriorna distribucija $f(v_N|y_N^+)$ k $f(v_N^+)$.

Prikladno je ovdje spomenuti koncept **pogrešnog modela**. Fizikalni sustav koji generira podatke $\{y_i\}$ može biti potpuno drugačiji od matematičkog modela kojeg koristimo. Ili barem u nekim aspektima. Konceptualno to poimamo zamišljanjem kako su podaci y^+ generirani istinitim signalom $v^+ = \{v_i^+\}$ koji je, na primjer, određen determinističkom dinamikom:

$$v_{i+1}^+ = \psi_{true}(v_i^+), \quad i \in \mathbb{N}_0,$$

$$v_0^+ \sim N(m_0, C_0).$$

Vidimo kako preslikavanje ψ_{true} određuje dinamiku istinitog signala. Najčešće istiniti operator ψ_{true} nije poznat. Stoga podatke koje dobivamo često kombiniramo sa stohastičkim modelom u kojem šum $\xi = \{\xi_i\}_{i \in \mathbb{N}_0}$ služi kako bi pokrio razliku istinitog i korištenog mo-

dela ψ . Statističari ovakvu situaciju, kada se izvor podataka razlikuje od modela, zovu pogrešna specifikacija modela.

Do sada smo proučavali kvalitetu podataka, a sada se okrećemo proučavanju kvalitete algoritama za asimilaciju podataka. Tri pristupa procjeni kvalitete su: **Bayesovska procjena kvalitete, procjena kvalitete signala i točnost predikcije**. Nas će zanimati samo zadnji pristup jer se u njemu kvaliteta algoritma vrednuje prema sposobnosti da točno predviđa i stoga se često koristi u numeričkim predikcijama vremena.

Točnost predikcije

Često je u asimilaciji podataka cilj pružiti bolja predviđanja budućnosti. U tom kontekstu algoritmi asimilacije podataka mogu biti uspoređivani prema sposobnosti što preciznijeg predviđanja i prema sposobnosti predviđanja vremenski što dalje budućnosti. Ako pretpostavimo da je naš asimilacijski prozor zadan od t_0 do t_N onda nas zanima koliko dobro algoritam predviđa budućnost određen broj koraka unaprijed. Označimo sa k koliko iteracija unaprijed želimo predviđati. Postavlja se pitanje kako mjeriti kvalitetu predikcije? Nudimo odgovore iz aspekta Bayesovske procjene kvalitete i procjene kvalitete signala. Bayesovska procjena proučava udaljenost aproksimativne funkcije gustoće $f_{approx}(v|y)$ i stvarne funkcije gustoće $f(v|y)$ kada je model primijenjen od kraja asimilacijskog prozora k puta. U vidu procjene signala analogno se model primjenjuje k puta, ali se uspoređuje signal s istinitim signalom u trenutku t_{N+k} . Neka je $v_{approx}(= v_N)$ aproksimacija stvarnog stanja signala v_N^+ na kraju asimilacijskog prozora. U determinističkoj dinamici koristeći v_{approx} kao inicijalnu točku model primjenjujemo k puta i rezultat uspoređujemo sa v_{N+k}^+ . U stvarnim primjenama ovakva metodologija nailazi na mnogo problema, od pogrešne specifikacije modela koja vodi predikcije u potencijalno pogrešnom smjeru do problema kvalitete podatka i kvalitete algoritma kojeg se koristi.

Poglavlje 3

Algoritmi sekvencijalne asimilacije podataka

Po uzoru na [5, pogl. 4] strukturiramo ovo poglavlje na sljedeći način. U 3.1 iznosimo primjere dinamičkih sustava. Uzeti su osnovni primjeri linearnog i nelinearnog inicijalnog problema sustava ODJ-a. Njih smo riješili pomoću metoda prikazanih u 1.5. Njihovim numeričkim rješavanjem dobili smo diskretne dinamičke sustave koje koristimo u testiranju algoritama opisanih u potpoglavljima 3.2 i 3.3.

Obrađujemo algoritme sekvencijalne asimilacije podataka. Prisjetimo se iz 2.2 da u sekvencijalnoj asimilaciji podataka pokušavamo odrediti uvjetnu distribuciju stanja modela v_i obzirom na već poznate podatke $Y_i = \{y_l\}_{l=1}^i$. Sekvencijalno ažuriranje iz vremena i u $i + 1$ smo rastavili na dva dijela: predikciju (2.5) i analizu (2.6). Svi algoritmi obrađeni u ovom poglavlju reflektiraju i sadrže ova dva koraka.

Na kraju, u potpoglavljju 3.4 testiramo metode na navedenim primjerima i prikazujemo rezultate.

3.1 Primjeri dinamičkih sustava

U ovom odjeljku prikazujemo primjere dinamičkih sustava, tj. modela koje dobivamo iz sustava običnih diferencijalnih jednažbi. Prvo navodimo primjer linearne dinamike koju ćemo asimilirati koristeći Kalmanov filter. To je algoritam sekvencijalne asimilacije koji pretpostavlja linearnu dinamiku i normalnost šumova te je obrađen u sljedećem potpoglavljju. Nakon toga nudimo primjer nelinearne dinamike koja će biti iskorištena kasnije za testiranje 3DVAR i ExKF algoritama.

Linearni dinamički sustav

Pretpostavimo da je naš model zapisan kao linearni sustav diferencijalnih jednadžbi:

$$\begin{aligned}\frac{dv}{dt} &= Lv, \quad L \in \mathbb{R}^{n \times n}, \\ v(0) &= v_0.\end{aligned}$$

Neka je $t = \{t_i\}_{\mathbb{N}_0}$ ekvidistantna mreža s razmakom $h = t_{i+1} - t_i$. Sa v_i ćemo označavati procjenu, tj. aproksimaciju od $v(t_i)$. Vrijedi:

$$v_{i+1} = v_i + \int_{t_i}^{t_{i+1}} f(\tau, v(\tau)) d\tau = v_i + \int_{t_i}^{t_{i+1}} Lv(\tau) d\tau.$$

Integral možemo procijeniti najjednostavnijom Eulerovom metodom:

$$\int_{t_i}^{t_{i+1}} f(\tau, v(\tau)) d\tau \approx (t_{i+1} - t_i) f(t_i, v(t_i)) = hf(t_i, v(t_i)),$$

pa dobiti dinamiku sustava:

$$v_{i+1} = v_i + hf(t_i, v_i) = v_i + hLv_i = (I + hL)v_i = Mv_i = \psi(v_i).$$

Ili možemo koristiti Runge-Kutta metodu reda 3:

$$\psi_1 = f(t_i, v_i) = Lv_i,$$

$$\psi_2 = f(t_i + a_2h, v_i + hA_{21}\psi_1) = Lv_i + hA_{21}L^2v_i,$$

$$\psi_3 = f(t_i + a_3h, v_i + h(A_{31}\psi_1 + A_{32}\psi_2)) = Lv_i + hA_{31}L^2v_i + hA_{32}L^2v_i + h^2A_{32}A_{21}L^3v_i$$

i dobiti:

$$\begin{aligned}v_{i+1} &= v_i + h(c_1\psi_1 + c_2\psi_2 + c_3\psi_3) \\ &= Iv_i + hc_1Lv_i + hc_2(Lv_i + hA_{21}L^2v_i) + hc_3(Lv_i + hA_{31}L^2v_i + hA_{32}L^2v_i + h^2A_{32}A_{21}L^3v_i) \\ &= (I + hc_1L + hc_2(L + hA_{21}L^2) + hc_3(L + hA_{31}L^2 + hA_{32}L^2 + h^2A_{32}A_{21}L^3))v_i \\ &= Mv_i.\end{aligned}$$

Otuda vidimo kako rješavanjem Runge-Kutta metodom dobijemo da je funkcija ψ linearni operator kao i u Eulerovoj metodi:

$$v_{i+1} = Mv_i = \psi(v_i).$$

Primjer 3.1.1. Kao konkretan primjer ćemo uzeti $n = 2$ i matricu $L = \begin{pmatrix} -3/2 & 1/2 \\ 1/4 & 0 \end{pmatrix}$.

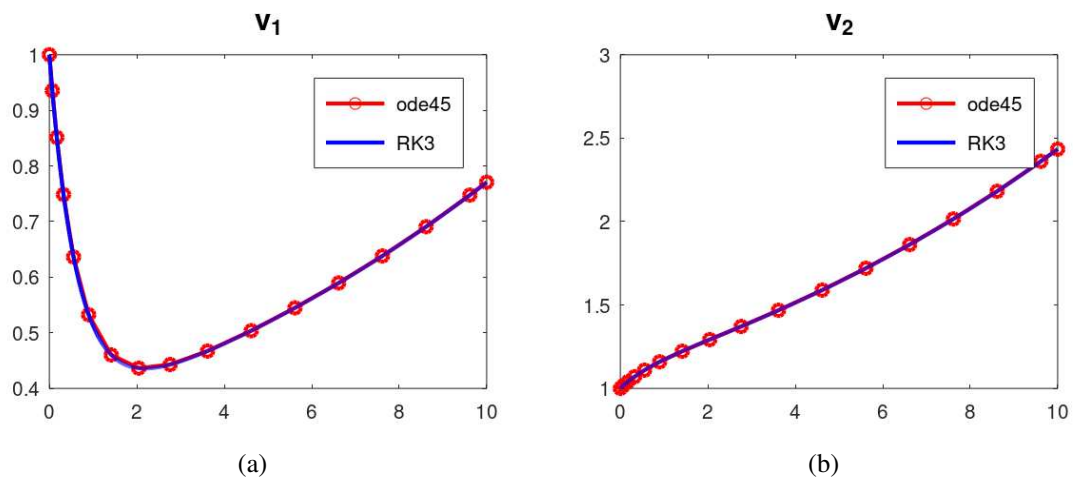
Imamo zadan sustav:

$$\begin{aligned} \frac{dv_1}{dt}(t) &= -1.5v_1(t) + 0.5v_2(t), \\ \frac{dv_2}{dt}(t) &= 0.25v_1(t) \end{aligned}$$

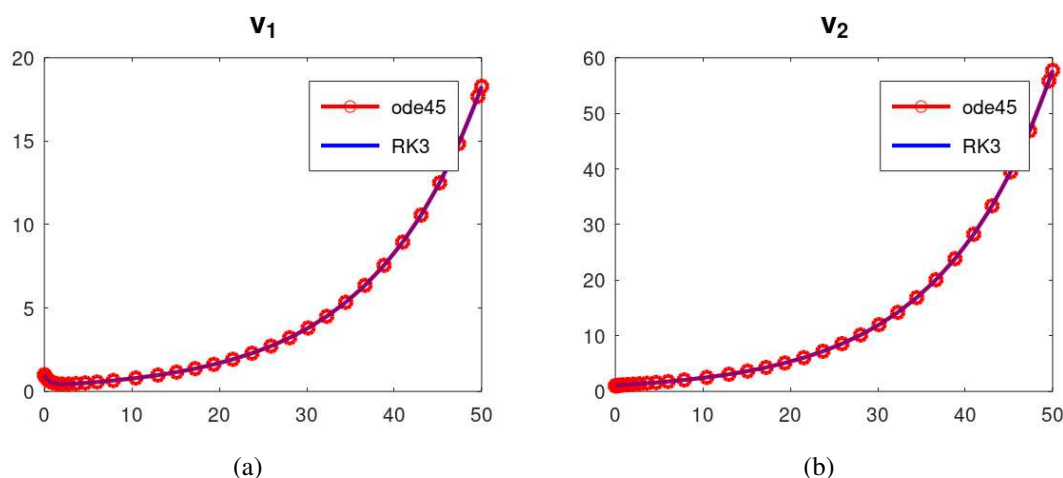
i početni uvjet:

$$v(0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Prikažimo numeričko rješenje dobiveno u Octave-u. Rješenje ćemo prikazati na dva intervala kako bi dobili bolji dojam dinamike sustava. Uzevši ekvidistantnu mrežu na intervalu $[0, 10]$ i $[0, 50]$ s korakom $h = 0.001$ numerički rješavamo sustav koristeći Runge-Kutta metodu reda 3 (RK3). Dobiveni rezultati su prikazani na 3.1 i 3.2 te su uspoređeni s rezultatima dobivenima koristeći gotovu funkciju ode45 koja rješava sustave diferencijalnih jednadžbi koristeći RK4 ili RK5.



Slika 3.1: Linearni sustav ODJ riješen s RK3 na $[0, 10]$. (a) Rješenje funkcije v_1 na $[0, 10]$. (b) Rješenje funkcije v_2 na $[0, 10]$.



Slika 3.2: Linearni sustav ODJ riješen s RK3 na $[0, 50]$. (a) Rješenje funkcije v_1 na $[0, 50]$. (b) Rješenje funkcije v_2 na $[0, 50]$.

Nelinearni dinamički sustav

Uzmimo slučaj kada imamo sustav običnih diferencijalnih jednadžbi:

$$\begin{aligned} \frac{dv}{dt} &= f(t, v(t)), \\ v(0) &= v_0, \end{aligned}$$

gdje je f neka nelinearna, dovoljno glatka funkcija.

Primjer 3.1.2. (Lorenzov sustav)

Uzmimo za primjer Lorenzov 63 model koji je jedan od važnijih modela za bolje razumijevanje asimilacije podataka. 1963. godine Edward Lorenz razvio je ovaj pojednostavljeni model kako bi opisao atmosfersku konvekciju. Originalni rad je citiran preko 20,000 puta.

$$\begin{aligned} \frac{dv_1}{dt} &= \sigma(v_2 - v_1), \\ \frac{dv_2}{dt} &= v_1(r - v_3) - v_2, \\ \frac{dv_3}{dt} &= v_1v_2 - bv_3. \end{aligned}$$

Detaljna analiza ovog modela nije nam od interesa. Stoga za parametre σ , r i b ćemo koristiti poznate klasične vrijednosti $(\sigma, r, b) = (10, 28, \frac{8}{3})$. Ove vrijednosti se često koriste jer su odličan primjer kaotičnog sustava. Za malu perturbaciju početnog uvjeta dobiva se potpuno drugačija dinamika sustava što je pokazatelj velike osjetljivosti. Za rješenje sustava

diferencijalnih jednadžbi može se iskoristiti Eulerova ili Runge-Kutta metoda. Vektorska oznaka koju koristimo za aproksimaciju od $v(t_i) = \begin{pmatrix} v_1(t_i) \\ v_2(t_i) \\ v_3(t_i) \end{pmatrix}$ je $v_i = \begin{pmatrix} v_{1,i} \\ v_{2,i} \\ v_{3,i} \end{pmatrix}$. Eulerovom metodom dobivamo sljedeću dinamiku modela:

$$v_{i+1} = v_i + hf(t_i, v_i) = \psi(v_i),$$

tj.

$$v_{i+1} = \begin{pmatrix} v_{1,i} \\ v_{2,i} \\ v_{3,i} \end{pmatrix} + h \begin{pmatrix} v_{1,i} + \sigma(v_{2,i} - v_{1,i}) \\ v_{2,i} + v_{1,i}(r - v_{3,i+1}) - v_{2,i} \\ v_{3,i} + v_{1,i}v_{2,i} - bv_{3,i+1} \end{pmatrix} = \psi(v_i).$$

Korištenjem Runge-Kutta metode reda 3 imamo sljedeći opis:

$$v_{i+1} = \psi(v_i) = v_i + h(c_1\psi_1 + c_2\psi_2 + c_3\psi_3),$$

gdje su:

$$\begin{aligned} \psi_1 &= f(t_i, v_i), \\ \psi_2 &= f(t_i + a_2h, v_i + hA_{21}\psi_1), \\ \psi_3 &= f(t_i + a_3h, v_i + h(A_{31}\psi_1 + A_{32}\psi_2)). \end{aligned}$$

Izgled rješenja Lorenzovog sustava kada istovremeno na osima prikažemo odabrane komponente prikazujemo na slici 3.3. Sustav je riješen koristeći RK3 metodu s ekvidistantnim korakom $h = 0.01$ uz parametre $(\sigma, r, b) = (10, 28, \frac{8}{3})$ i početni uvjet $v(0) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$.

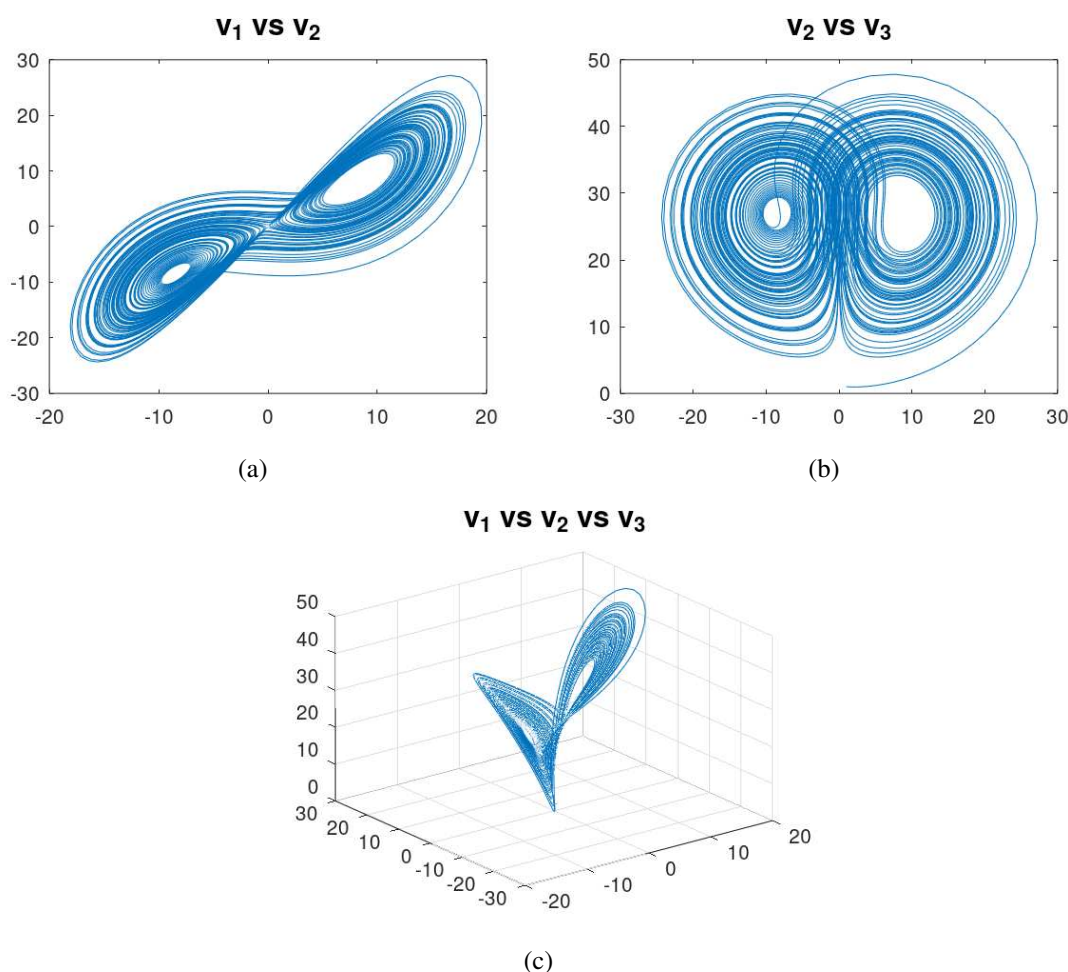
3.2 Kalmanov filter

Kalmanov filter je sekvencijalna metoda za ažuriranje uvjetne funkcije gustoće $f_{v|Y}(v_i|Y_i)$ iz vremena i u vrijeme $i+1$ kada su šum modela i opservacijski šum normalno distribuirani ($\xi_i \sim N(0, \Sigma)$ i $\theta_i \sim N(0, \Gamma)$) te su ψ i h linearni operatori:

$$\psi(v) = Mv, \quad h(v) = Hv,$$

gdje su $M \in \mathbb{R}^{n \times n}$ i $H \in \mathbb{R}^{m \times n}$. Pretpostavljamo $m \leq n$ te $\text{rang}(H) = m$.

Primijetimo kako je uz ove pretpostavke uvjetna funkcija gustoće $f_{v|Y}(v_i|Y_i)$ u biti normalno distribuirana te je stoga u potpunosti određena srednjom vrijednosti i kovarijacijskom matricom. S (m_i, C_i) ćemo označavati srednju vrijednost i kovarijacijsku matricu



Slika 3.3: Lorenzov sustav riješen koristeći RK3. (a) Na osima su rješenja funkcija v_1 i v_2 . (b) Na osima su rješenja funkcija v_2 i v_3 . (c) 3-dimenzionalni graf gdje su na osima rješenja od v_1 , v_2 i v_3 .

slučajne varijable $v_i|Y_i$ te analogno $(\hat{m}_{i+1}, \hat{C}_{i+1})$ označavaju srednju vrijednost i kovarijacijsku matricu od $v_{i+1}|Y_i$. Naglasimo da su (m_{i+1}, C_{i+1}) srednja vrijednost i kovarijacijska matrica slučajne varijable $v_{i+1}|Y_{i+1}$. Kalmanovim filterom zapravo izvodimo preslikavanje $(m_i, C_i) \mapsto (m_{i+1}, C_{i+1})$ preko varijabli $(\hat{m}_{i+1}, \hat{C}_{i+1})$. Oznaka $\hat{\cdot}$ će u nastavku rada uvijek biti u konotaciji s predikcijom.

Teorem 3.2.1. *Pretpostavimo $C_0, \Gamma, \Sigma > 0$. Tada je $C_i > 0, \forall i \in \mathbb{N}_0$ te vrijedi:*

$$C_{i+1}^{-1} = (MC_i M^T + \Sigma)^{-1} + H^T \Gamma^{-1} H,$$

$$C_{i+1}^{-1}m_{i+1} = (MC_iM^T + \Sigma)^{-1}Mm_i + H^T\Gamma^{-1}y_{i+1}.$$

Dokaz. Dokaz tvrdnje $C_i > 0, \forall i \in \mathbb{N}_0$ provodimo indukcijom. Baza nam je kao što smo i pretpostavili $C_0 > 0$. Pretpostavimo da za neki $i \in \mathbb{N}_0$ vrijedi $C_i > 0$. Predikcijski korak u slučaju stohastičke dinamike je sljedeći

$$v_{i+1} = \psi(v_i) + \xi_i = Mv_i + \xi_i, \quad \xi_i \sim N(0, \Sigma).$$

Iz ovoga slijedi

$$\mathbb{E}(v_{i+1}|Y_i) = \mathbb{E}(Mv_i|Y_i) + \mathbb{E}(\xi_i|Y_i).$$

Kako je ξ_i nezavisan od Y_i imamo:

$$\hat{m}_{i+1} = Mm_i. \quad (3.1)$$

Na sličan način dobivamo

$$\begin{aligned} \mathbb{E}((v_{i+1} - \hat{m}_{i+1}) \otimes (v_{i+1} - \hat{m}_{i+1})|Y_i) &= \mathbb{E}((M(v_i - m_i) + \xi_i) \otimes (M(v_i - m_i) + \xi_i)|Y_i) \\ &= \mathbb{E}(M(v_i - m_i) \otimes M(v_i - m_i)|Y_i) + \mathbb{E}(\xi_i \otimes \xi_i|Y_i) \\ &\quad + \mathbb{E}(M(v_i - m_i) \otimes \xi_i|Y_i) + \mathbb{E}(\xi_i \otimes M(v_i - m_i)|Y_i), \end{aligned}$$

a kako je ξ_i nezavisan od Y_i i v_i slijedi

$$\begin{aligned} \hat{C}_{i+1} &= M\mathbb{E}((v_i - m_i) \otimes (v_i - m_i)|Y_i)M^T + \Sigma \\ &= MC_iM^T + \Sigma. \end{aligned} \quad (3.2)$$

Primijetimo da je $\hat{C}_{i+1} > 0$ jer je $C_i > 0$ iz pretpostavke indukcije te je $\Sigma > 0$ iz pretpostavke teorema.

Promotrimo korak analize. Iz (2.6) koristeći pretpostavku o normalnosti imamo

$$\begin{aligned} \exp(-\frac{1}{2}\|v - m_{i+1}\|_{C_{i+1}}^2) &\propto \exp(-\frac{1}{2}\|\Gamma^{-\frac{1}{2}}(y_{i+1} - HC)\|^2 - \frac{1}{2}\|\hat{C}_{i+1}^{-\frac{1}{2}}(v - \hat{m}_{i+1})\|^2) \\ &= \exp(-\frac{1}{2}\|y_{i+1} - Hv\|_{\Gamma}^2 - \frac{1}{2}\|v - \hat{m}_{i+1}\|_{\hat{C}_{i+1}}^2). \end{aligned}$$

Izjednačavanjem kvadratnih termina desne i lijeve strane dobivamo

$$C_{i+1}^{-1} = \hat{C}_{i+1}^{-1} + H^T\Gamma^{-1}H, \quad (3.3)$$

a izjednačavanjem linearnih

$$C_{i+1}^{-1}m_{i+1} = \hat{C}_{i+1}^{-1}\hat{m}_{i+1} + H^T\Gamma^{-1}y_{i+1}. \quad (3.4)$$

Uvrštavanjem (3.1) i (3.2) umjesto \hat{m}_{i+1} i \hat{C}_{i+1}^{-1} dobivamo traženi rezultat. Preostaje pokazati korak indukcije $C_{i+1} > 0$. No to slijedi iz (3.3) jer smo pretpostavili $\Gamma^{-1} > 0$ i prethodno pokazali da $\hat{C}_{i+1}^{-1} > 0$. \square

Metodu Kalmanovog filtera možemo preformulirati tako da koristimo kovarijacijsku matricu (C_{i+1}), a ne matricu preciznosti (C_{i+1}^{-1}).

Korolar 3.2.2. *Pod istim pretpostavkama kao u 3.2.1 formule Kalmanovog filtera možemo preformulirati na sljedeći način:*

$$\begin{aligned}d_{i+1} &= y_{i+1} - H\hat{m}_{i+1}, \\S_{i+1} &= H\hat{C}_{i+1}H^T + \Gamma, \\K_{i+1} &= \hat{C}_{i+1}H^T S_{i+1}^{-1}, \\m_{i+1} &= \hat{m}_{i+1} + K_{i+1}d_{i+1}, \\C_{i+1} &= (I - K_{i+1}H)\hat{C}_{i+1},\end{aligned}$$

gdje su $(\hat{m}_{i+1}, \hat{C}_{i+1})$ zadani kao u (3.1) i (3.2).

Dokaz. Iz (3.3) primjenom Woodburyjevog matričnog identiteta (1.3.2) dobivamo

$$\begin{aligned}C_{i+1} &= \hat{C}_{i+1} - \hat{C}_{i+1}H^T(\Gamma + H\hat{C}_{i+1}H^T)^{-1}H\hat{C}_{i+1} \\&= (I - \hat{C}_{i+1}H^T(\Gamma + H\hat{C}_{i+1}H^T)^{-1}H)\hat{C}_{i+1} \\&= (I - \hat{C}_{i+1}H^T S_{i+1}^{-1}H)\hat{C}_{i+1} \\&= (I - K_{i+1}H)\hat{C}_{i+1},\end{aligned}$$

gdje koristimo oznake kao u iskazu korolara

$$\begin{aligned}S_{i+1} &= H\hat{C}_{i+1}H^T + \Gamma, \\K_{i+1} &= \hat{C}_{i+1}H^T S_{i+1}^{-1}.\end{aligned}$$

Uz ove oznake iz (3.4) dobivamo

$$\begin{aligned}m_{i+1} &= C_{i+1}\hat{C}_{i+1}^{-1}\hat{m}_{i+1} + C_{i+1}H^T\Gamma^{-1}y_{i+1} \\&= (I - K_{i+1}H)\hat{m}_{i+1} + C_{i+1}H^T\Gamma^{-1}y_{i+1}.\end{aligned}\tag{3.5}$$

Ponovno koristeći (3.3) i množeći je slijeva sa C_{i+1} dobivamo

$$\begin{aligned}C_{i+1}H^T\Gamma^{-1}H &= I - C_{i+1}\hat{C}_{i+1}^{-1} \\&= I - (I - K_{i+1}H) \\&= K_{i+1}H,\end{aligned}$$

a kako je $\text{rang}(H) = m$ zaključujemo

$$C_{i+1}H^T\Gamma^{-1} = K_{i+1}.$$

Stoga iz (3.5) za kraj dobivamo

$$m_{i+1} = (I - K_{i+1}H)\hat{m}_{i+1} + K_{i+1}y_{i+1} = \hat{m}_{i+1} + K_{i+1}d_{i+1}.$$

□

Ključna razlika između dviju formulacija metode Kalmanovog filtera iz 3.2.1 i 3.2.2 je ta što se u 3.2.2 inverzija matrica odvija u prostoru opservacija dimenzije m , a ne u prostoru signala dimenzije n . U mnogo primjena je $m \ll n$ (m puno manji od n) te su inverzije stoga puno efikasnije.

Veličina d_{i+1} se zove inovacija u koraku $i + 1$ (eng. *innovation at time step $i + 1$*) te je mjera udaljenosti predviđenog stanja od opservacije u $i + 1$. Matrica K_{i+1} je poznatija po svojem engleskom nazivu *Kalman gain*.

3.3 3DVAR i ExKF

Uvodimo familiju metoda koje se pozivaju na minimizacijski problem koji je baza za Kalmanov filter, a koji ima prirodnu generalizaciju na ne-Gaussovske probleme. Jednadžba ažuriranja buduće procijenjene srednje vrijednosti glasi:

$$m_{i+1} = \arg \min_v I_{filter}(v),$$

gdje je

$$I_{filter}(v) := \frac{1}{2} \|y_{i+1} - Hv\|_{\Gamma}^2 + \frac{1}{2} \|v - \hat{m}_{i+1}\|_{\hat{C}_{i+1}}^2. \quad (3.6)$$

\hat{m}_{i+1} dobiven je iz (3.1), a \hat{C}_{i+1} iz (3.2).

Iako je Kalmanov filter definiran za linearne Gaussovske probleme, formulacijom preko minimizacije generaliziramo na nelinearne probleme. Prirodna generalizacija koja slijedi iz (3.6) je:

$$I_{filter}(v) := \frac{1}{2} \|y_{i+1} - h(v)\|_{\Gamma}^2 + \frac{1}{2} \|v - \hat{m}_{i+1}\|_{\hat{C}_{i+1}}^2,$$

gdje je

$$\hat{m}_{i+1} = \psi(m_i) + \xi_i,$$

a ažurirana procjena stanja modela dobije se iz:

$$m_{i+1} = \arg \min_v I_{filter}(v).$$

Primijetimo kako je minimizacijska funkcija intuitivno i prirodno definirana. Traži se procjena stanja modela koja minimizira zbroj udaljenosti od predikcije modela \hat{m}_{i+1} i

opservacije y_{i+1} . Zbog jednostavnosti uzet ćemo slučaj kada je funkcija h linearni operator, tj. $h(v) = Hv$. Ovakav kvadratni minimizacijski problem je eksplicitno rješiv i koristeći 3.2.2 dobivamo sljedeće formule za ažuriranje sustava:

$$\begin{aligned} m_{i+1} &= (I - K_{i+1}H)\hat{m}_{i+1} + K_{i+1}y_{i+1}, \\ K_{i+1} &= \hat{C}_{i+1}H^T S_{i+1}^{-1}, \\ S_{i+1} &= H\hat{C}_{i+1}H^T + \Gamma. \end{aligned} \quad (3.7)$$

Sada iznosimo dva algoritma dobivena na upravo opisan način, naime minimizacijom funkcije $I_{filter}(v)$ i različitim odabirom kovarijacijske matrice modela \hat{C}_{i+1} .

3DVAR

Naziv metode slijedi iz činjenice da je varijacijska metoda te da sekvencijalno ažurira model za svako fiksirano vrijeme i . Kada se primjeni na stvarne fizikalne probleme algoritam minimizira tri prostorne dimenzije. Algoritam dobijemo iz (3.7) fiksiranjem kovarijacijske matrice modela: $\hat{C}_i = \hat{C}, \forall i \in \mathbb{N}_0$. Uz

$$\begin{aligned} S &= H\hat{C}H^T + \Gamma, \\ K &= \hat{C}H^T S^{-1}, \end{aligned}$$

formuliramo predikciju i analizu na sljedeći način:

$$\begin{aligned} \hat{m}_{i+1} &= \psi(m_i), \\ m_{i+1} &= (I - KH)\hat{m}_{i+1} + Ky_{i+1}. \end{aligned}$$

Extended Kalman Filter(ExKF)

Ideja ExKF algoritma je propagirati kovarijacijsku matricu linearizacijom nelinearne dinamike (2.4), a srednju vrijednost koristeći (2.3). Modifikacijom korolara 3.2.2 slijedi:

Predikcija:

$$\begin{aligned} \hat{m}_{i+1} &= \psi(m_i), \\ \hat{C}_{i+1} &= D\psi(m_i) C_i D\psi(m_i)^T + \Sigma. \end{aligned}$$

Analiza:

$$\begin{aligned} S_{i+1} &= H\hat{C}_{i+1}H^T + \Gamma, \\ K_{i+1} &= \hat{C}_{i+1}H^T S_{i+1}^{-1}, \\ m_{i+1} &= (I - K_{i+1}H)\hat{m}_{i+1} + K_{i+1}y_{i+1}, \\ C_{i+1} &= (I - K_{i+1}H)\hat{C}_{i+1}. \end{aligned}$$

3.4 Algoritmi testirani na primjerima

U ovom potpoglavlju iznosimo rezultate dobivene prilikom testiranja algoritama navedenih u 3.2 i 3.3 na primjerima 3.1.1 i 3.1.2. Kalmanov filter je metoda koja pretpostavlja linearnu dinamiku te ćemo ju stoga testirati na linearnom primjeru i vidjeti kako reagira na promjene šumova. 3DVAR i ExKF algoritme ćemo testirati i usporediti na nelinearnom primjeru, tj. na Lorenzovom sustavu. Kako smo već spomenuli svaki algoritam sadrži dva koraka: predikciju i analizu. U ovome testiranju algoritama predikcije su rađene samo na sljedeći trenutak (iteraciju), a ne na više njih.

Kalmanov filter testiran na linearnom primjeru

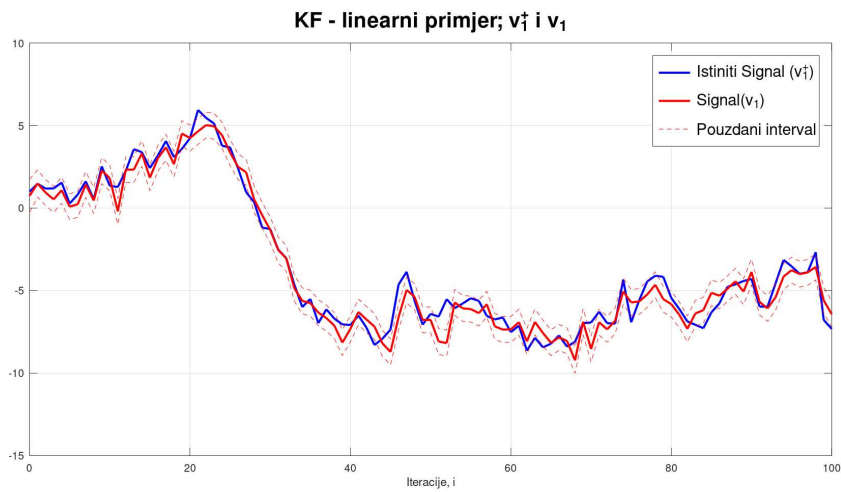
Provjerimo koliko uspješno Kalmanov filter aproksimira istiniti signal na već spomenutom primjeru linearne dinamike 3.1.1 u kojem je ψ linearni operator.

Model je numerički riješen koristeći RK3 metodu uz korak $h = 0.01$. Kako je kao početni uvjet istinitog signala uzet $v(0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, početnu vrijednost signala odabiremo kao $v_0 \sim N(v(0), I)$, gdje je $I \in \mathbb{R}^{2 \times 2}$ identiteta. Za opservacijski operator uzimamo identitetu, $h(v) = Iv$.

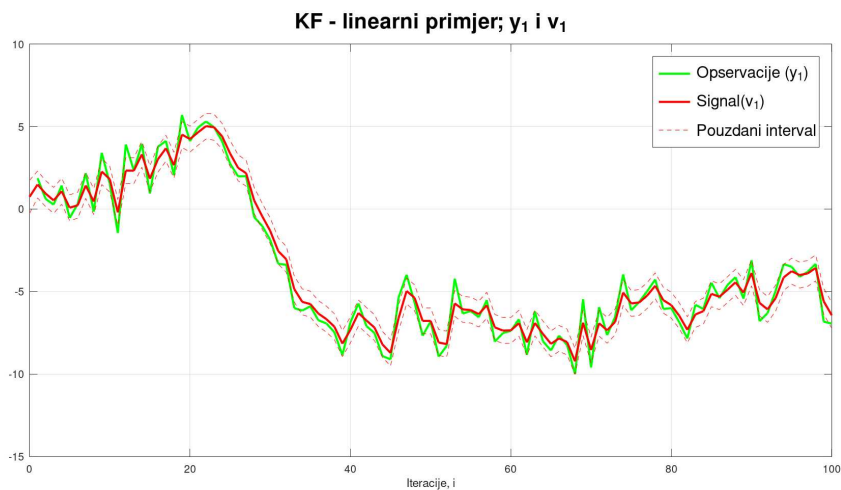
Napravit ćemo dva slučaja s obzirom na varijancu opservacijskih šumova kako bi vidjeli koliko uspješno Kalmanov filter određuje stanje istinitog signala kojeg koristimo za simuliranje opservacija. Usporedit ćemo uspješnost određivanja istinitog signala kada je opservacijski šum velik i malen. Usporedbom želimo naglasiti važnost dobivanja preciznih i pouzdanih podataka u određivanju istinitog signala.

Zbog preglednosti rada prikazujemo samo rješenja dobivena za prvu komponentu funkcije $v(v_1)$. Također, prikazujemo asimilaciju prvih 100 iteracija dinamičkog sustava na široj mreži zbog vizualno kvalitetnijih grafova.

Prvotno pretpostavimo $\xi \sim N(0, 1)$ i $\theta \sim N(0, 1)$. Rezultate vidimo na 3.4. Primjećujemo kako algoritam uspješno određuje stanja istinitog signala preko opservacija.



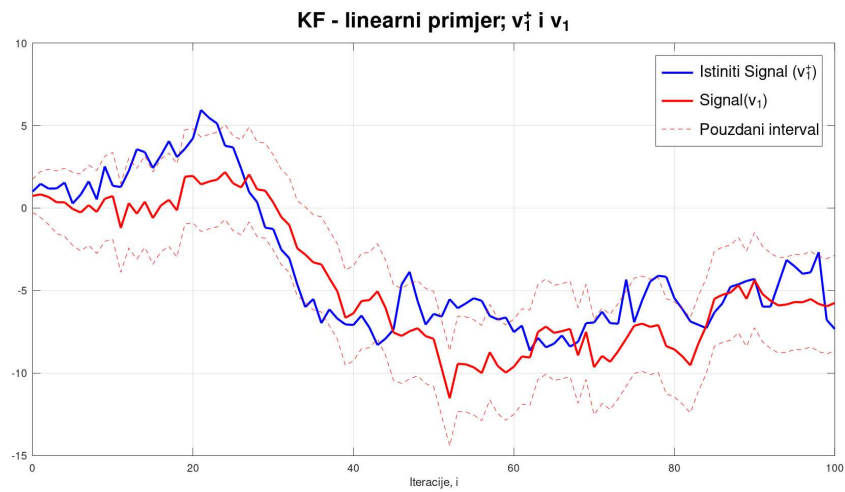
(a)



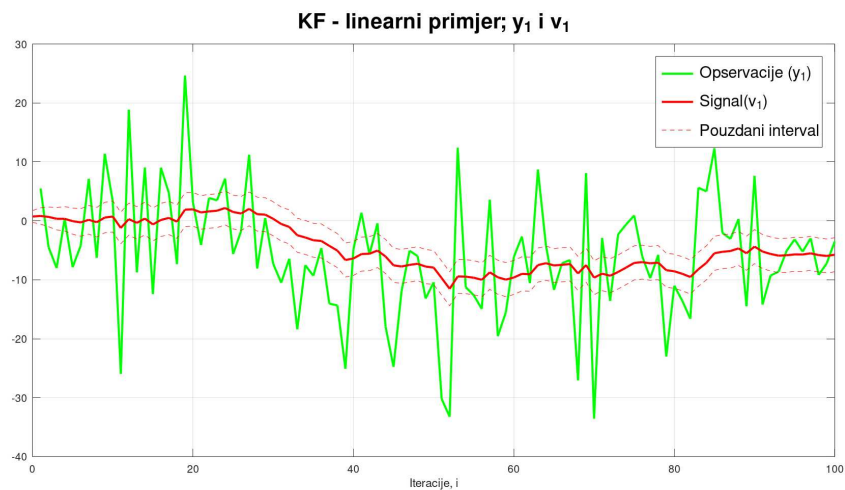
(b)

Slika 3.4: Kalmanov filter uz $\xi \sim N(0, 1)$ i $\theta \sim N(0, 1)$. (a) Usporedba istinitog signala s asimilacijskim signalom. (b) Prikaz asimilacije modela (signala) s opservacijama.

Pogledajmo koliko uspješno se model asimilira s obzirom na veću varijancu opservacijskih greški. Na primjer, pretpostavimo $\theta \sim N(0, 10^2)$. Na slici 3.5 primjećujemo kako uz veći opservacijski šum naš signal lošije aproksimira istiniti signal kojeg smo koristili za simuliranje opservacija. Unatoč očekivanom lošijem aproksimiranju istinitog signala algoritam se i dalje asimilira podacima zadovoljavajuće.



(a)



(b)

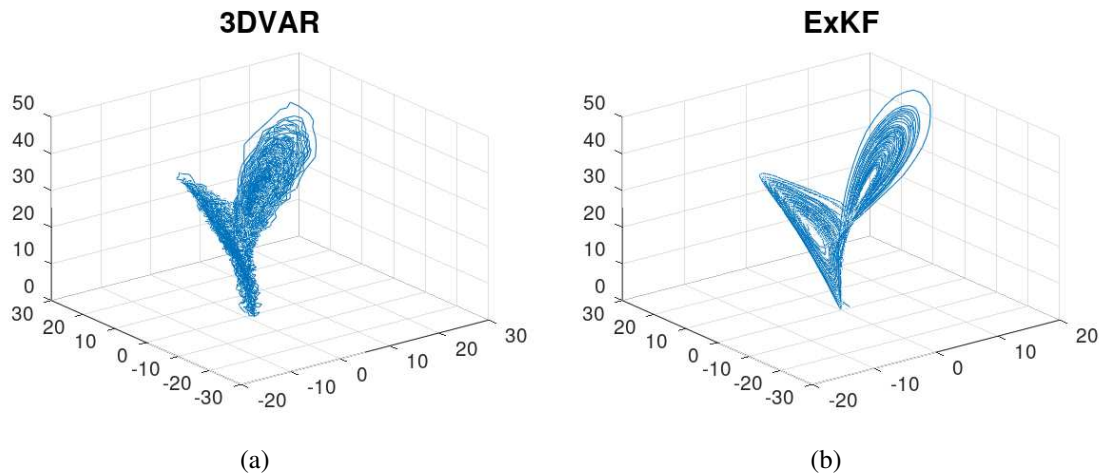
Slika 3.5: Kalmanov filter uz $\xi \sim N(0, 1)$ i $\theta \sim N(0, 10^2)$. (a) Usporedba istinitog signala s asimilacijskim signalom. (b) Prikaz asimilacije modela (signala) s opservacijama.

Na slici 3.5(a) je crvena linija (u legendi označena sa Signal) ista kao i na slici 3.5(b), samo je zbog velike perturbacije opservacija y -os skalirana drugačije.

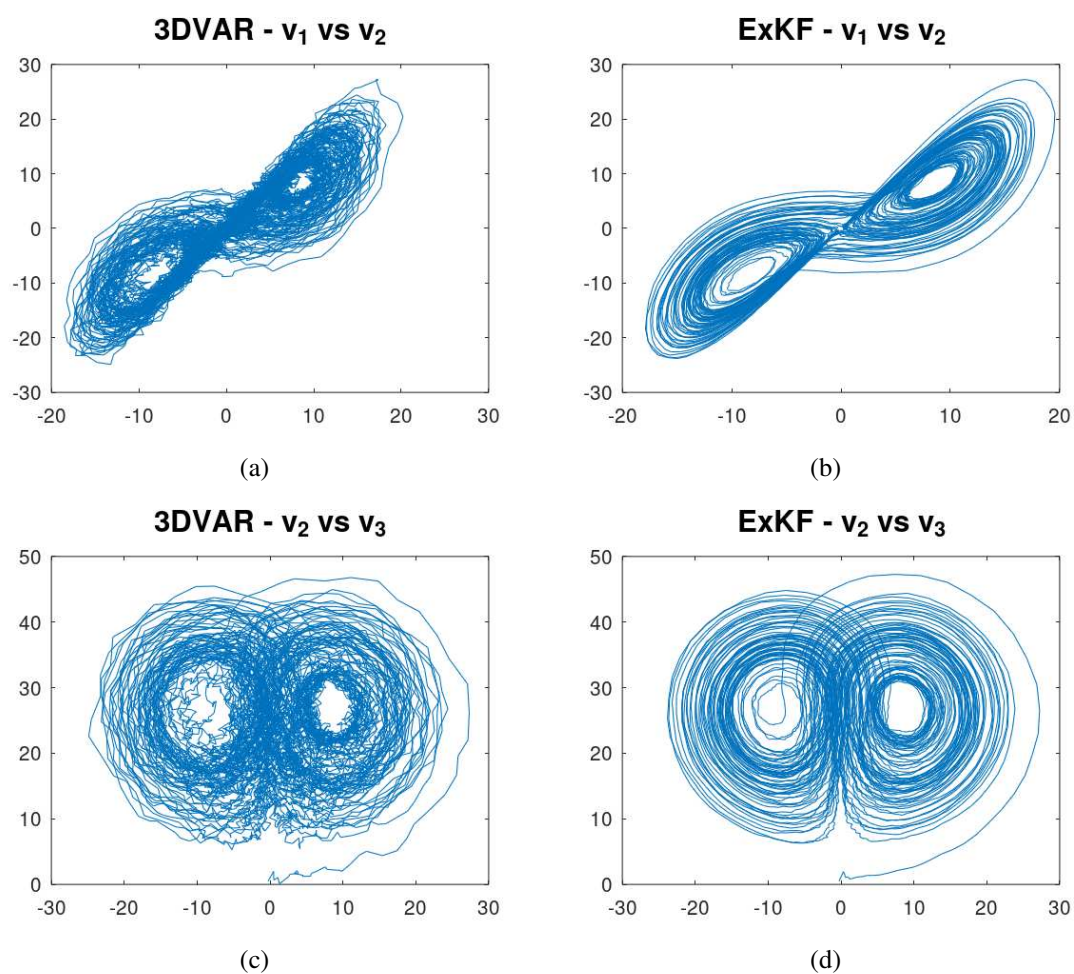
3DVAR i ExKF testirani na Lorenzovom sustavu

Uzmimo ψ kao u primjeru 3.1.2. Neka su opservacije jednakih dimenzija kao i signal, $m = n = 3$ te neka je $h(v) = Iv$, gdje je $I \in \mathbb{R}^{3 \times 3}$ identiteta. Sustav je riješen RK3 metodom uz ekvidistantni korak $h = 0.01$. Neka su šumovi modela $\xi \sim N(0, 0.1^2)$ i opservacijski šumovi $\theta \sim N(0, 1)$. Neka je $v_0 \sim N(v(0), C_0)$, gdje je $v(0) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ i $C_0 = I$.

Na slikama 3.6 i 3.7 vidimo rezultate dvaju algoritama. Primijetimo kako je rješenje od ExKF algoritam puno glade i bolje odgovara slici 3.3 koja je prikaz rješenja sustava bez asimilacije.



Slika 3.6: Usporedba 3DVAR i ExKF algoritama testiranih na Lorenzovom sustavu riješenom RK3 metodom. (a) 3DVAR rješenje prikazano u 3 dimenzije. (b) ExKF rješenje prikazano u 3 dimenzije.



Slika 3.7: Usporedba 3DVAR i ExKF algoritama testiranih na Lorenzovom sustavu. Lijevi stupac ((a) i (c)) je prikaz rješenja korištenjem 3DVAR metode. Desni stupac ((b) i (d)) je prikaz rješenja korištenjem ExKF metode.

Poglavlje 4

Epidemiološki modeli

Prethodna poglavlja su se bavila teorijskim postavljanjem matematičkog okvira za asimilaciju podataka, tj. preciznije za sekvencijalnu asimilaciju podataka. Tu teoriju želimo testirati na jednom stvarnom problemu. Uzet ćemo podatke o pandemiji COVID-19. Pokušat ćemo koristeći asimilaciju podataka napraviti korisne predikcije o širenju COVID-19 u odabranoj populaciji. Za ovo istraživanje moramo uvesti matematičke modele širenja virusa i infekcija.

Koristit ćemo klasifikacijske modele za širenje infektivnih bolesti. Korištena je knjiga [2] kao izvor. Počeci ovakvih modela kreću u ranom 20. stoljeću. Najčešće su zadani kao sustav običnih ili parcijalnih diferencijalnih jednadžbi kojima se opisuje dinamika prelaska jedinice iz populacije iz jedne u drugu klasu. Modeli klasificiraju populaciju u grupe (klase) sa zajedničkim svojstvima u odnosu na virus koji se promatra. Tako ćemo na primjer gledati kretanje broja ljudi koji su potencijalni kandidati za zarazu, broja zaraženih, ukupan broj oporavljenih od bolesti itd. U sljedećim potpoglavljima opisujemo svaki od odabranih modela.

Najjednostavniji temeljni model je SIR model. Na njega se nadograđuju dodatna dva modela koja ćemo koristiti u asimilaciji COVID-19 podataka, a to su SEIR i SEIRD modeli.

4.1 SIR model

Model se sastoji od 3 kategorije: S - (Susceptible) broj potencijalnih kandidata za zarazu, I - (Infected) broj zaraženih osoba i R - (Recovered) broj osoba koje su se oporavile od virusa. Dinamika modela se može opisati dijagramom $S \rightarrow I \rightarrow R$. Zdrava osoba iz promatrane populacije koja dođe u kontakt sa zaraženom osobom biva i sama zaražena s nekom vjerojatnošću. Ona se iz S kategorije premješta u I kategoriju. Zaražene osobe (kategorija I) koje se uspiju oporaviti od bolesti se sele u kategoriju R.

Sljedeći sustav običnih diferencijalnih jednadžbi opisuje ovu dinamiku:

$$\begin{aligned} S'(t) &= -I(t)\beta\frac{S(t)}{N}, \\ I'(t) &= I(t)\beta\frac{S(t)}{N} - \lambda I(t), \\ R'(t) &= \lambda I(t), \end{aligned}$$

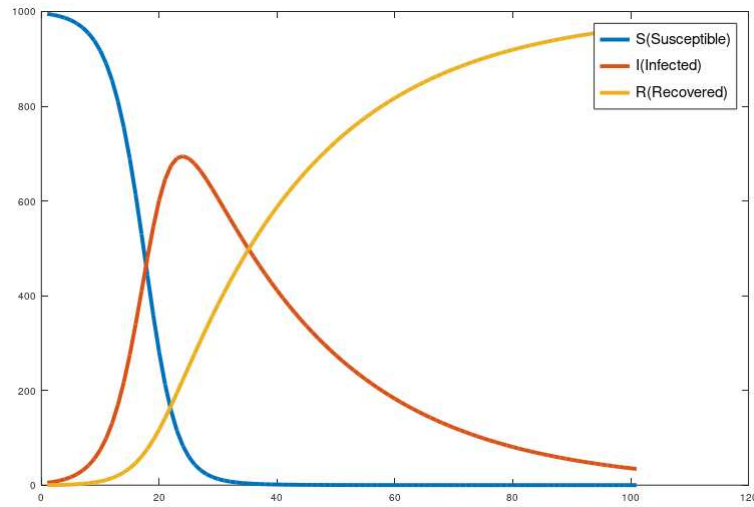
gdje je N veličina populacije koju proučavamo (broj stanovnika). Pretpostavljamo da je veličina populacije konstantna ($N = \text{const.}$) te da je zbroj svih klasa u svakom trenutku jednak N . U slučaju SIR modela vrijedi $N = S + I + R$. Parametri β i λ također imaju svoju interpretaciju. Parametar β predstavlja prosječni broj kontakata koje jedna zaražena osoba može imati i potencijalno inficirati u jedinici vremena. Parametar λ predstavlja dio zaraženih koji se prosječno oporavi u jednom danu (najjednostavnija opcija je uzeti $\lambda = \frac{1}{D}$, gdje je D broj dana koliko dugo je zaražena osoba bolesna i može zaraziti druge, prije nego se oporavi).

Pogledajmo na slici 4.1 kretanje SIR modela uz zadane početne uvjete i navedene parametre. Sustav je numerički riješen koristeći Eulerovu metodu uz korak $h = 1$ (što predstavlja 1 dan).

Slika je u skladu s očekivanim kretanjima i intuicijom. Broj potencijalnih kandidata (S) s vremenom pada dok istovremeno broj zaraženih (I) naglo raste. Oko 25. iteracije broj zaraženih dolazi do svojeg maksimuma nakon čega naglo pada. Broj oporavljenih (R) s malim odmakom raste kroz cijeli period promatranja. To je zato što nas zanima kumulativan broj oporavljenih.

Primijetimo koliko model ima mjesta za poboljšanje. Dodatno možemo promatrati broj umrlih od bolesti kako bismo dobili informaciju koliko je smrtonosan virus; kako pojava cjepiva i cijepljenje populacije utječe na širenje infekcije; možemo pratiti broj osoba koje su bile izložene virusu i nije poznato jesu li zaražene; praćenje hospitaliziranih osoba kojima je potrebna medicinska briga u prebolijevanju infekcije; osobe koje su se oporavile od infekcije možda imaju određen period jakog imuniteta u kojem se ne mogu opet zaraziti istom infekcijom.

Mnogobrojne su modifikacije, a u nastavku obrađujemo samo dvije dodatne nadogradnje na SIR model. Uvodimo dodatnu klasu eksponiranih osoba E (Exposed), koje su bile u kontaktu s nekom zaraženom osobom i trenutno je upitno jesu li zaražene. I na kraju uvodimo klasu osoba preminulih od zaraze D (Deceased).



Slika 4.1: SIR simulacija. Broj iteracija = 100. $N = 1000$, $\beta = 0.4$, $\lambda = 0.04$. Početni uvjeti: $S(0) = N - I(0)$, $I(0) = 5$ i $R(0) = 0$.

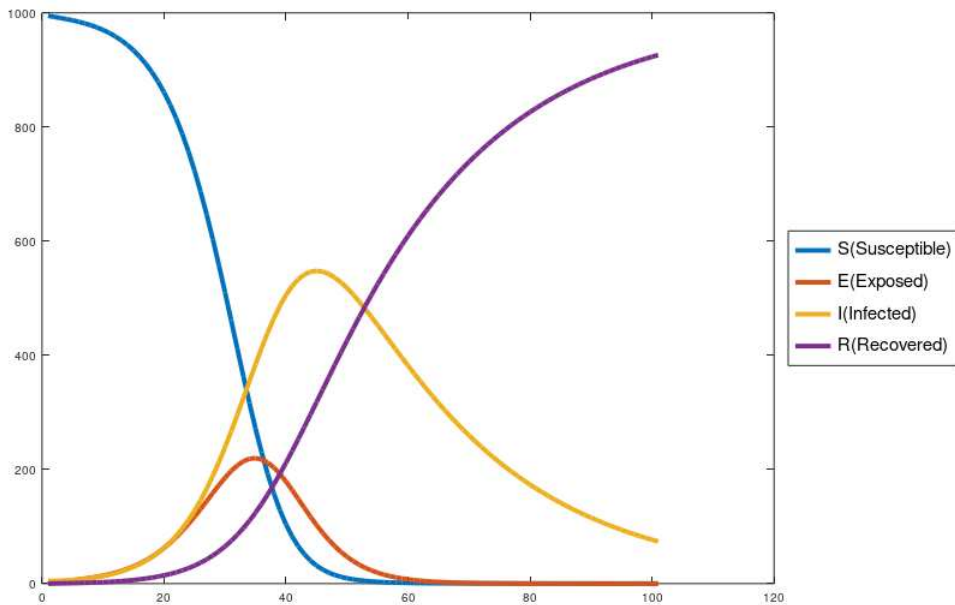
4.2 SEIR model

Za mnoge infektivne bolesti postoji period latencije. Period latencije je vrijeme u kojem su osobe koje jesu zaražene, ali nisu u stanju prenijeti zarazu, tj. zaraziti drugu osobu. Pretpostavljamo da je duljina perioda latencije eksponencijalna varijabla s parametrom ϵ . Slijedi da je očekivano vrijeme latencije $\frac{1}{\epsilon}$.

Na slici 4.2 vidimo istu dinamiku kao i u slučaju SIR modela samo je još uključena klasa E. Korišteni su isti parametri kao i u prethodnom slučaju.

Stoga u SIR model uvodimo dodatnu već spomenutu klasu ljudi eksponiranih infekciji E (Exposed). To je klasa u kojoj su ljudi koji se nalaze u periodu latencije. Dinamika modela s dodatnom klasom je $S \rightarrow E \rightarrow I \rightarrow R$. Matematički model opisan sustavom diferencijalnih jednačbi je:

$$\begin{aligned} S'(t) &= -I(t)\beta\frac{S(t)}{N}, \\ E'(t) &= I(t)\beta\frac{S(t)}{N} - \epsilon E(t), \\ I'(t) &= \epsilon E(t) - \lambda I(t), \\ R'(t) &= \lambda I(t). \end{aligned}$$



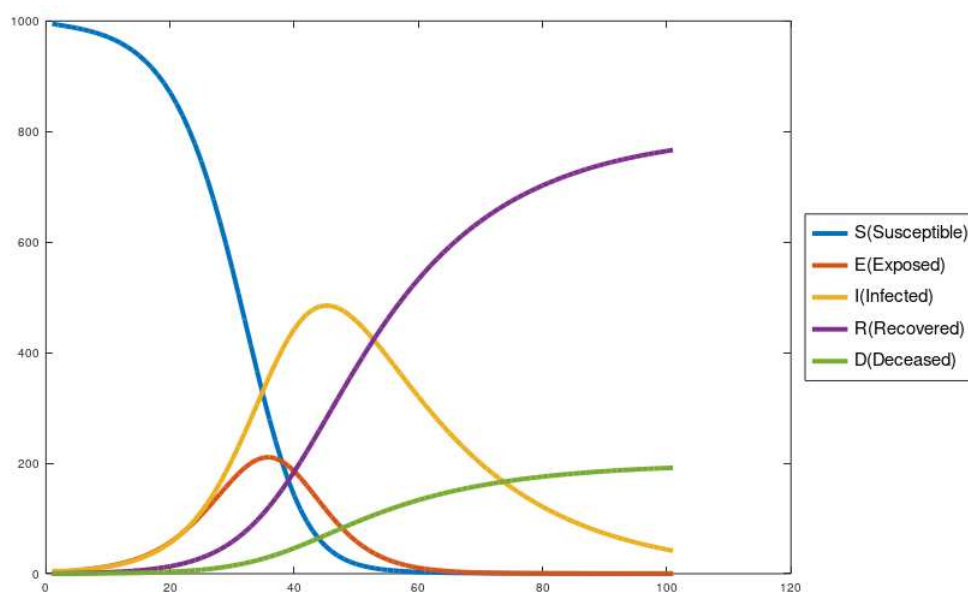
Slika 4.2: SEIR simulacija. Broj iteracija = 100. $N = 1000$, $\beta = 0.4$, $\lambda = 0.04$, $\epsilon = 0.2$. Početni uvjeti: $S(0) = N - I(0)$, $E(0) = 0$, $I(0) = 5$ i $R(0) = 0$.

4.3 SEIRD model

Zadnja nadogradnja koju uvodimo je klasa osoba preminulih od infekcije D (Deceased). Kretanje između klasa je $S \rightarrow E \rightarrow I \rightarrow R \rightarrow D$. U dinamici klase D koristimo parametar μ koji predstavlja stopu smrtnosti od infekcije:

$$\begin{aligned} S'(t) &= -I(t)\beta\frac{S(t)}{N}, \\ E'(t) &= I(t)\beta\frac{S(t)}{N} - \epsilon E(t), \\ I'(t) &= \epsilon E(t) - \lambda I(t) - \mu I(t), \\ R'(t) &= \lambda I(t), \\ D'(t) &= \mu I(t). \end{aligned}$$

Motivacija i razlog za uvođenje ove klase je u tome što je broj preminulih javno dostupan podatak i relativno pouzdano mjerilo. Naravno, problem u broju preminulih je koliko njih je uistinu preminulo zbog infekcije koju modeliramo, a ne iz drugih razloga. Primijetimo kako će naše opservacije biti dvodimenzionalne jer će komponente biti broj



Slika 4.3: SEIRD simulacija. Broj iteracija = 100. $N = 1000$, $\beta = 0.4$, $\lambda = 0.04$, $\epsilon = 0.2$, $\mu = 0.01$. Početni uvjeti: $S(0) = N - I(0)$, $E(0) = 0$, $I(0) = 5$, $R(0) = 0$ i $D(0) = 0$.

zaraženih i broj preminulih u nekom vremenskom trenutku. Osim toga informacija o smrtnosti određene zaraze je iznimno važna.

Slika 4.3 prikazuje dinamiku SEIRD modela. Jedina novost u odnosu na prethodne simulacije je prikaz klase preminulih od infekcije (D). Očekivano ona raste kroz cijeli period promatranja dokle god je infekcija prisutna, tj. dokle god ima zaraženih ($I > 0$).

Poglavlje 5

Sekvencijalna asimilacija podataka na COVID-19 podacima

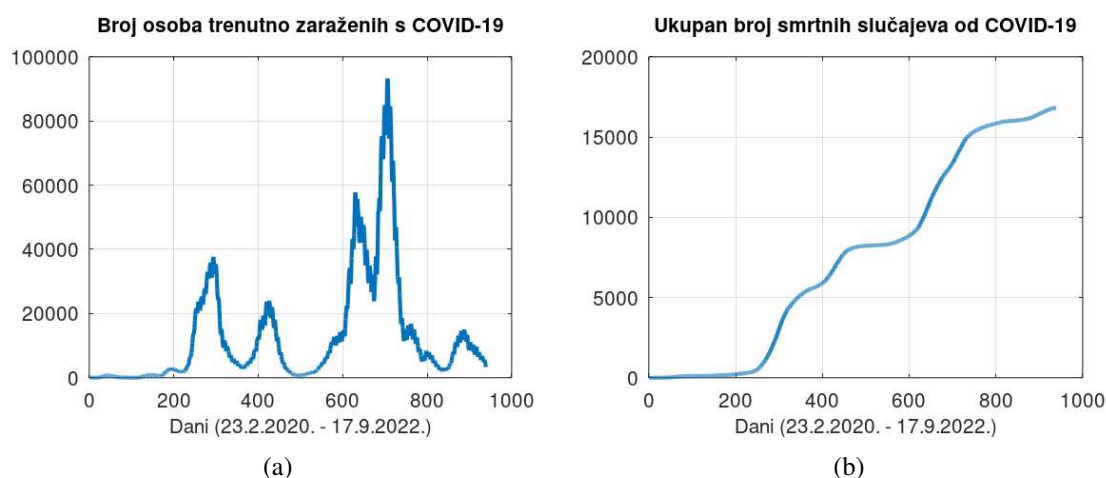
Do sada prikazanu i obrazloženu matematičku teoriju ćemo pokušati primijeniti na jednom stvarnom problemu. Svijet je početkom 2020. godine zadesila pandemija zaraze COVID-19. Pandemija je trajala 3 godine te je u tom razdoblju došlo do velike potrebe za matematičkim modelima širenja zaraze. Modeli koji mogu predvidjeti nagli val velikog broja zaraženih osoba bi potencijalno mogli spasiti na desetke tisuća života. Znanjem iz asimilacije podataka i podacima o širenju COVID-a pokušat ćemo retroaktivno predvidjeti jedan od valova širenja zaraze.

5.1 COVID-19 podaci

Podaci o COVID-19 zarazi koje koristimo u ovome radu su preuzeti iz baze podataka tvrtke Google zvane "COVID-19 Open Data Repository". Ova baza podataka sadrži jako puno podataka i parametara koji mogu biti povezani sa COVID-19 i to za više od 20 000 lokacija diljem svijeta. Podaci su sakupljeni od autoritativnih i pouzdanih organizacija. Ovaj rad će se fokusirati isključivo na podatke vezane uz Hrvatsku. Od mnogobrojnih podataka dostupnih u bazi koristimo samo broj trenutno (aktivno) zaraženih osoba u danu i ukupan broj preminulih osoba od zaraze.

Na slici 5.1 prikazujemo kretanje odabranih brojeva u Hrvatskoj u vremenu od 23. veljače 2022. godine kada je zabilježen prvi slučaj zaraze s COVID-19 do 17. rujna 2022. godine kada baza podataka jednostavno prestaje s praćenjem pandemije.

Važno je naglasiti kako su prikupljeni podaci koje asimiliramo izrazito nepouzdati i kontaminirani. Kada kažemo da su kontaminirani mislimo na grešku koja dolazi u mjerenju broja zaraženih. Ne možemo sa sigurnošću znati koliki je pravi broj zaraženih u Hrvatskoj u određenom danu. Osim tehničkih problema pri testiranju populacije na COVID-19



Slika 5.1: (a) Prikaz kretanja aktivnih slučajeva zaraze COVID-19. (b) Prikaz kretanja kumulativnih smrtnih slučajeva od COVID-19 od početka zaraze.

imamo i problem s ljudima koji se ne žele testirati. Imamo ljude koji nemaju simptome ili ih imaju, ali jednostavno ne učine ništa po tom pitanju i svrstaju simptome pod običnu prehladu. Puno je razloga zašto treba biti oprezan prema prikazanim brojkama zaraženih.

S druge strane broj smrtnih slučajeva od zaraze COVID-19 je također teško precizno odrediti. S kojom točnošću to možemo ustvrditi razliku između osobe preminule s COVID-19 i od COVID-19 zaraze? Kako odrediti je li to što je osoba u trenutku smrti bila zaražena utjecalo na to hoće li ona umrijeti ili neće, i u kojoj mjeri. Jasnije vidimo kako imamo jako nepouzdanu opservaciju.

5.2 Jednodnevne predikcije

Prvo pokažimo rezultate asimilacije COVID-19 podataka uz jednodnevne predikcije. Kako širenje zaraze modeliramo sa SEIR i SEIRD koji su nelinearni koristimo 3DVAR i ExKF algoritme u asimiliranju. Oba modela su riješena Eulerovom metodom.

Asimiliranje podataka provodimo u dva slučaja s obzirom na to kakve su nam opservacije. Prvi slučaj je onaj u kojem su nam opservacije jednodimenzionalne te predstavljaju broj aktivno zaraženih osoba. U ovom slučaju ćemo zarazu modelirati sa SEIR modelom. U drugom slučaju imamo dvodimenzionalne opservacije koje su broj aktivno zaraženih i ukupni broj smrtnih slučajeva. U drugom slučaju koristimo SEIRD model.

Parametri β , λ , ϵ i μ korišteni u modelima SEIR i SEIRD su uzeti kao fiksni i preuzeti iz članka [3] uz male promjene i modifikacije ako je rezultiralo poboljšanjem u asimilaciji. Vrijednosti parametara su: $\beta = 0.15$, $\lambda = 1/D$, gdje je $D = 10$ s obrazloženjem kao u 4.1, $\epsilon = 1/5$ i $\mu = 0.01$.

SEIR model i jednodimenzionalne opservacije

Kako je naš signal oblika

$$v_i = v(t_i) = \begin{pmatrix} S(t_i) \\ E(t_i) \\ I(t_i) \\ R(t_i) \end{pmatrix},$$

gdje je t_i neki dan u promatranom periodu naš opservacijski operator je zapravo $h(v_i) = H v_i$, gdje je $H = \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}$. Opservacije su jednodimenzionalne ($m = 1$) dok je signal 4-dimenzionalan ($n = 4$).

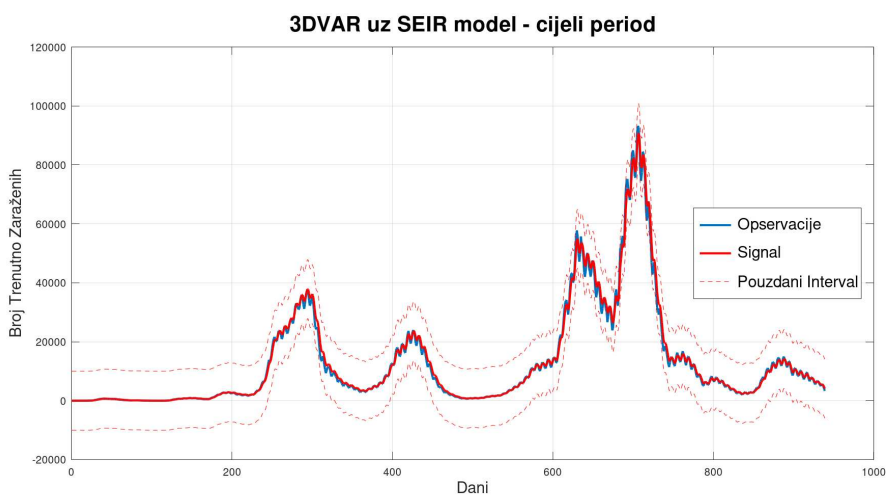
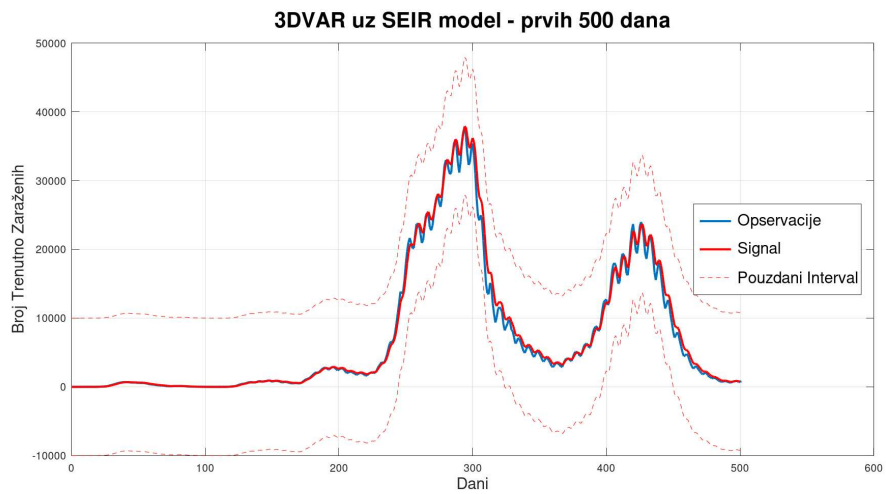
Osim gore navedenih parametara modela i opservacijskog operatora moramo odrediti i šumove. Zbog velike kontaminacije podataka i uzevši u obzir veličinu promatrane populacije za varijancu opservacijske greške je odabrana vrijednost $\Gamma = 10^4$ ($\theta \sim N(0, \Gamma)$). Za varijancu greške modela kod ExKF algoritma je uzeta $\Sigma = I_4 * 10^3$, gdje je $I_4 \in \mathbb{R}^{4 \times 4}$ identiteta.

Kao početne vrijednosti signala i kovarijacijske matrice uzeti su:

$$v_0 = \begin{pmatrix} S_0 \\ E_0 \\ I_0 \\ R_0 \end{pmatrix} = \begin{pmatrix} N-1 \\ 1 \\ 1 \\ 0 \end{pmatrix} \quad i \quad C_0 = I_4 * \Gamma^2,$$

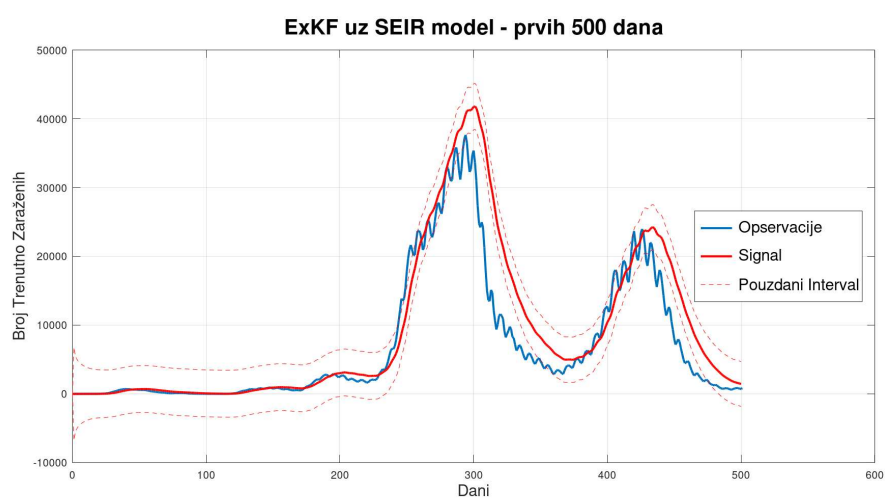
gdje je $N = 4 \times 10^6$.

Pogledajmo na slici 5.2 rezultate 3DVAR algoritma testiranog na podacima o aktivnom broju zaraženih uz SEIR model.

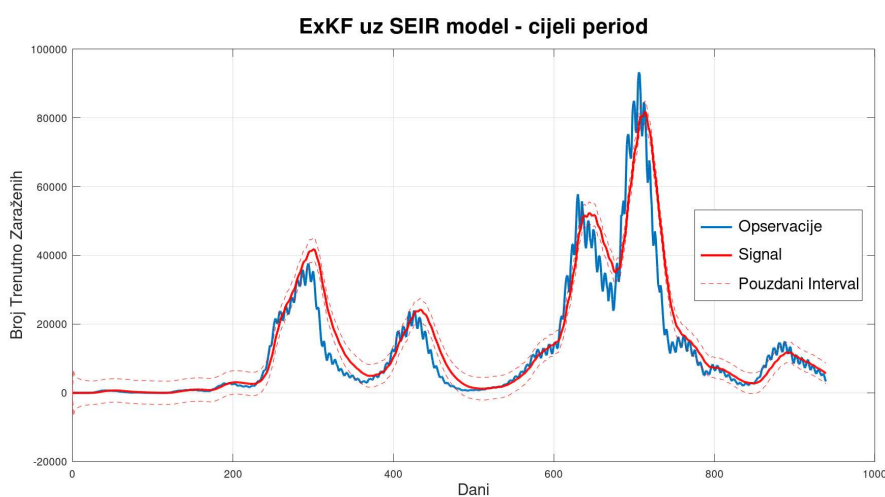


Slika 5.2: 3DVAR uz jednodimenzionalne opservacije. (a) Prikaz asimilacije prvih 500 dana od početka zaraze. (b) Prikaz asimilacije kroz cijeli promatrani period.

Na slici 5.3 vidimo rezultate ExKF metode. Primijetimo kako ExKF lošije asimilira podatke od 3DVAR algoritma.



(a)



(b)

Slika 5.3: ExKF uz jednodimenzionalne opservacije. (a) Prikaz asimilacije prvih 500 dana od početka zaraze. (b) Prikaz asimilacije kroz cijeli promatrani period.

No, je li 3DVAR asimilacija korisnija? Čini se kao da je 3DVAR metoda previše prilagođena (eng. *overfitted*) dok se za ExKF algoritam čini kao da samo određuje trend zaraze i nije ga briga za male oscilacije. Ovakvo obrazloženje nije nerealno jer, prisjetimo se, imamo jako kontaminirane i nepouzdana opservacije.

SEIRD model i dvodimenzionalne opservacije

Sada je $H = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$ te je naš signal oblika:

$$v_i = v(t_i) = \begin{pmatrix} S(t_i) \\ E(t_i) \\ I(t_i) \\ R(t_i) \\ D(t_i) \end{pmatrix},$$

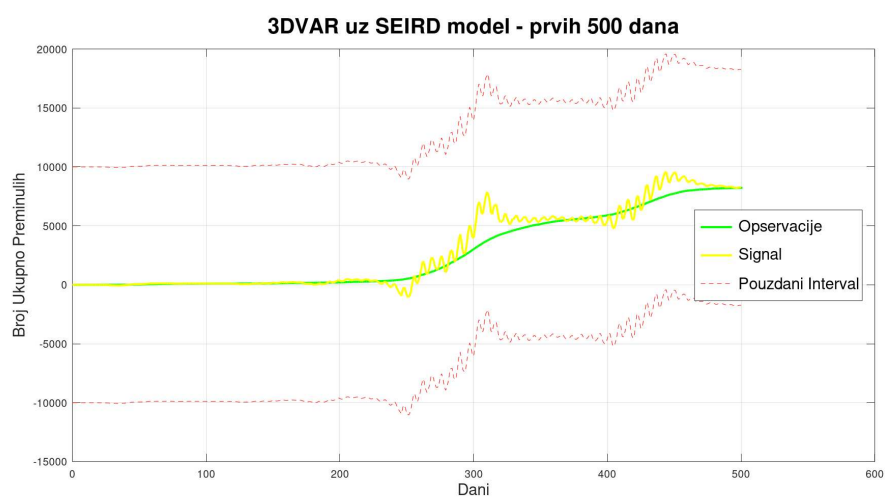
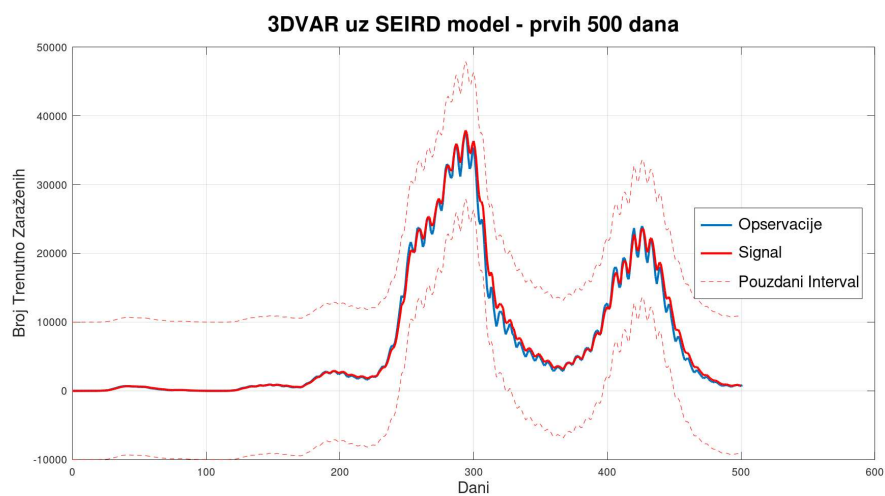
gdje je t_i neki dan u promatranom periodu. Novododana komponenta D prati ukupan broj preminulih od zaraze.

Opservacijska greška je $\Gamma = I_2 * 10^4$ ($\theta \sim N(0, \Gamma)$), gdje je $I_2 \in \mathbb{R}^{2 \times 2}$ identiteta. Za grešku modela kod ExKF algoritma je uzet $\Sigma = I_5 * 10^3$, gdje je $I_5 \in \mathbb{R}^{5 \times 5}$ identiteta.

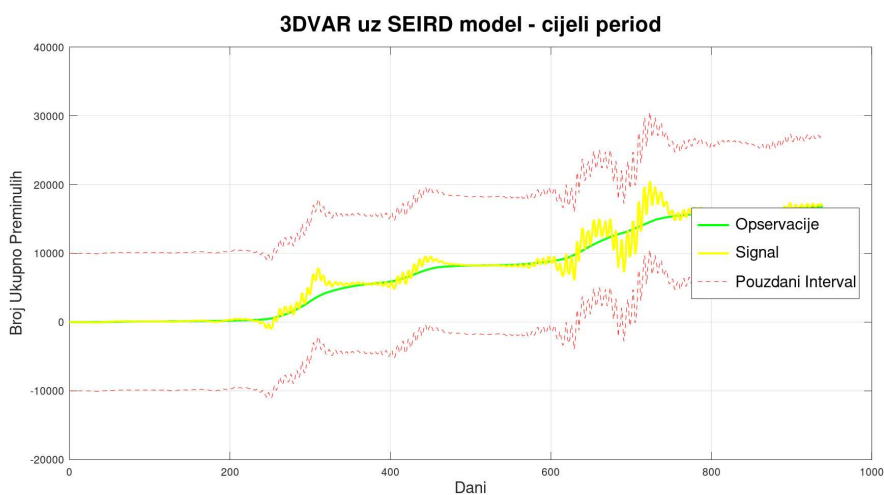
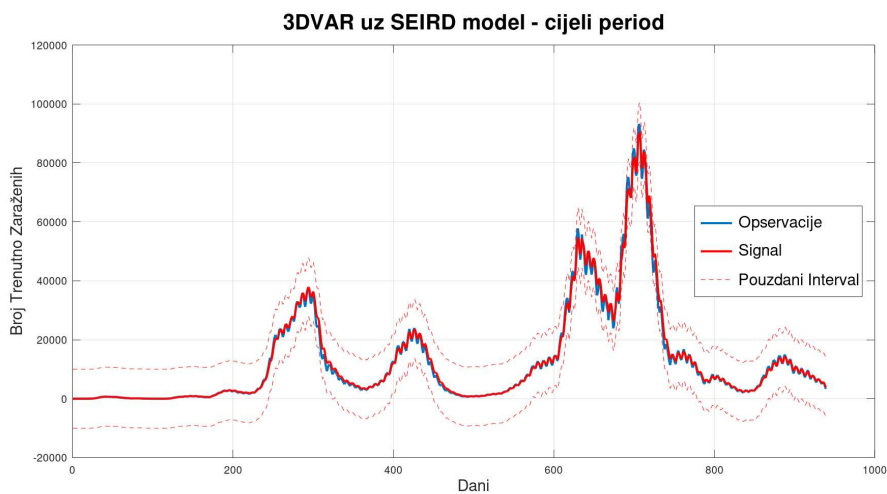
Početne vrijednosti signala i kovarijacijske matrice su:

$$v_0 = \begin{pmatrix} S_0 \\ E_0 \\ I_0 \\ R_0 \\ D_0 \end{pmatrix} = \begin{pmatrix} N - 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad i \quad C_0 = I_5 * \Gamma^2.$$

Na slikama 5.4 i 5.5 prikazujemo asimilaciju SEIRD modela koristeći 3DVAR algoritam.

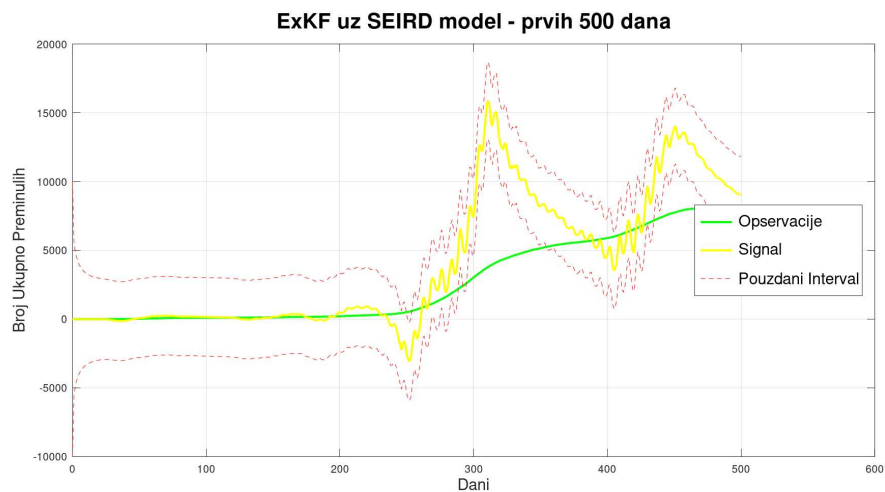
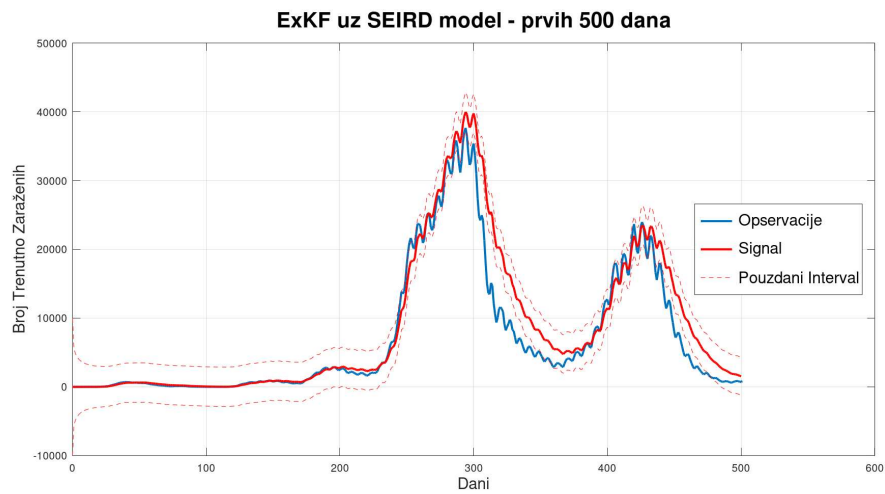


Slika 5.4: 3DVAR asimiliran prvih 500 dana od početka zaraze. (a) Prikaz asimilacije broja aktivno zaraženih. (b) Prikaz asimilacije ukupnog broja smrtnih slučajeva.



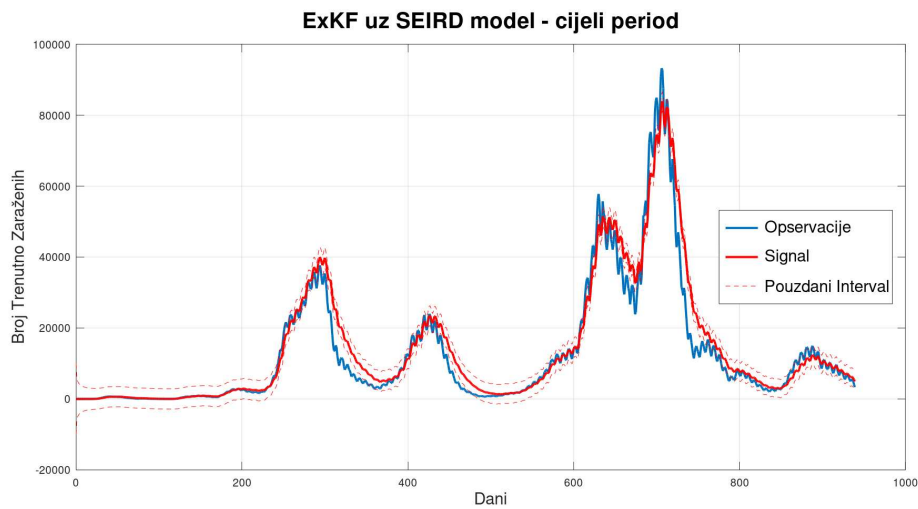
Slika 5.5: 3DVAR asimilirani cijeli promatrani period zaraze. (a) Prikaz asimilacije broja aktivno zaraženih. (b) Prikaz asimilacije ukupnog broja smrtnih slučajeva

Sada prikazujemo rezultate ExKF algoritma. Vidimo ih na slikama 5.6 i 5.7.

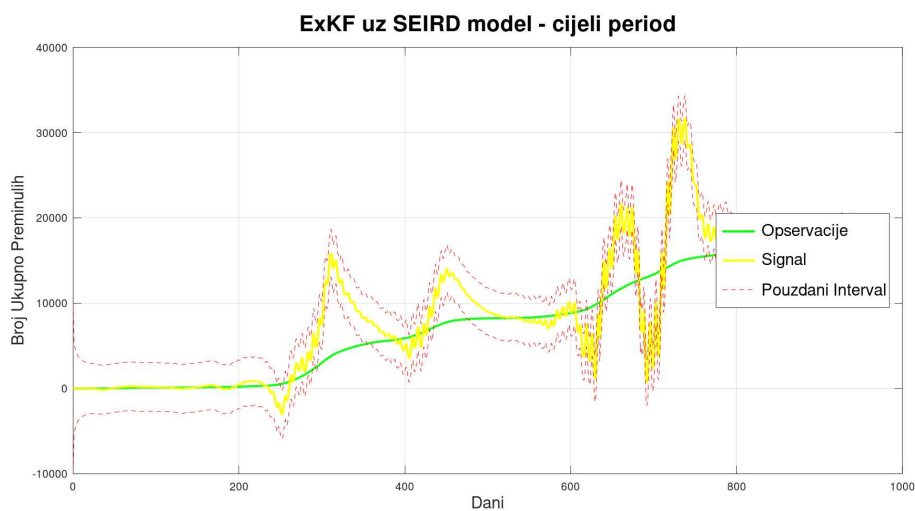


Slika 5.6: ExKF asimiliran prvih 500 dana od početka zaraze. (a) Prikaz asimilacije broja aktivno zaraženih. (b) Prikaz asimilacije ukupnog broja smrtnih slučajeva.

Rezultati asimilacije SEIRD modela vode nas na analogan zaključak u kojem vidimo kako je 3DVAR malo previše prilagođen opservacijama dok je ExKF metoda više usmjerena na trend što može biti korisnije u slučaju nepouzdatih podataka kakve mi imamo.



(a)



(b)

Slika 5.7: ExKF asimiliran cijeli promatrani period zaraze. (a) Prikaz asimilacije broja aktivno zaraženih. (b) Prikaz asimilacije ukupnog broja smrtnih slučajeva

Obje metode, a pogotovo ExKF, imaju problema pri asimiliranju broja smrtnih slučajeva te vidimo u trenucima velikih skokova broja zaraženih kako signal kreće jako oscilirati. Moguće obrazloženje je u SEIRD modelu kojem je prirodno naglo povećati broj smrtnih slučajeva kako broj zaraženih osoba raste. Oscilacije su stoga "borba" između modela i prilagodbe opservacijama.

5.3 Predikcije većih valova širenja zaraze

U ovom odjeljku ćemo pokušati napraviti predikcije na više od jednog dana u budućnost. Dugoročnije predikcije bi u slučaju širenja zaraze bile korisnije i vrijednije. Kada bi smo znali da će sljedećih tjedan ili mjesec dana nastupiti veliki val širenja virusa mogli bismo adekvatnije postupiti u preveniranju velikog broja zaraženih i samim time smanjiti broj preminulih.

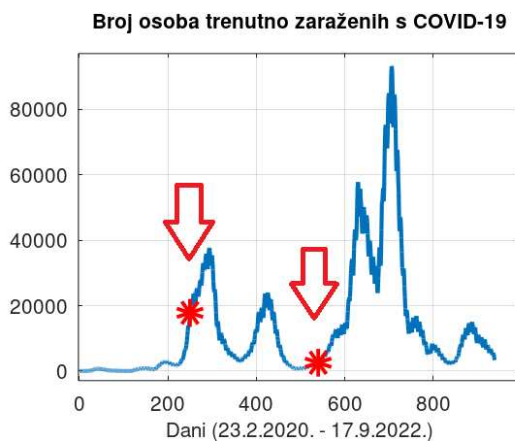
Podsjetimo se kako je definirana dinamika modela, tj. signala:

$$v_{i+1} = \psi(v_i) + \xi_i, \quad i \in \mathbb{N}_0.$$

Dugoročne predikcije radimo višestrukom primjenom funkcije dinamičkog sustava ψ . Neka je t_i sadašnji trenutak (u našem slučaju t_i je današnji dan), a v_i današnja procjena signala. Naša predikcija broja zaraženih k dana unaprijed je stoga $\psi^{(k)}(v_i)$, gdje je $\psi^{(k)} = \psi \circ \dots \circ \psi$ oznaka za k kompozicija funkcije ψ .

Širenje zaraze ćemo modelirati sa SEIRD modelom. Model je numerički riješen Eulerovom metodom iz koje je dobivena funkcija dinamičkog sustava ψ . Parametri SEIRD modela i veličine šumova isti su kao i u 5.2.

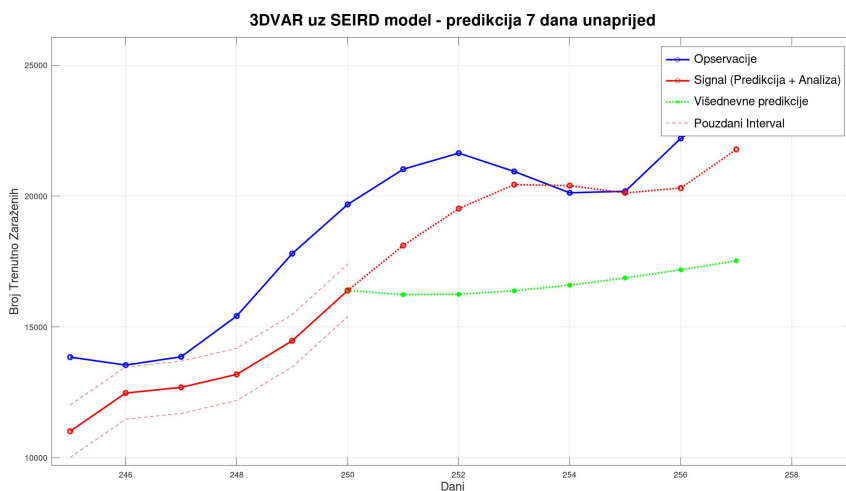
Pogledajmo ponovno graf kretanja dnevnog broja aktivnih slučajeva zaraze. Na slici 5.8 su označena dva ključna dana neposredno prije nastupanja velikog vala zaraženih. Oda-brani su 250. i 540. dan od početka pandemije. Iz ove dvije točke pokušavamo dobiti što bolje predikcije za 7 i 30 dana unaprijed.



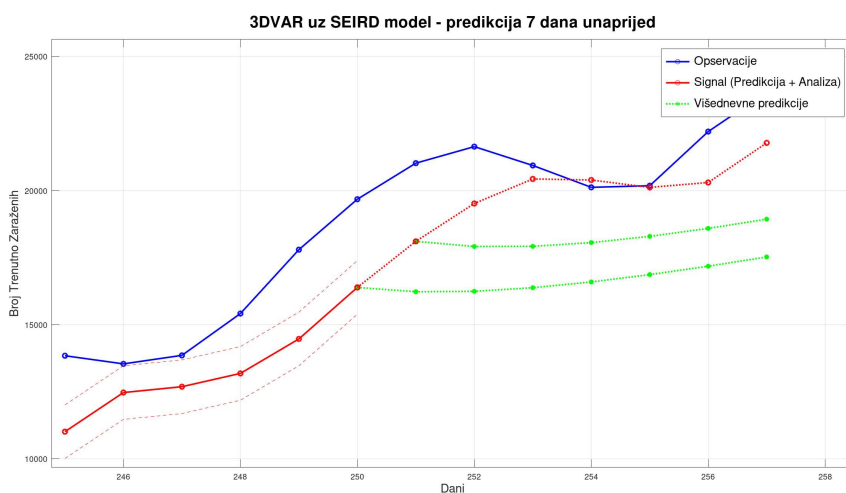
Slika 5.8: Prikaz kretanja aktivnih slučajeva zaraze COVID-19 na kojem su označena dva dana iz kojih će se raditi predikcije.

Primijetimo kako je 250. dan na slici 5.8 usred naglog širenja zaraze dok je 540. dan neposredno prije još većeg vala. Zbog toga očekujemo kako će predikcije biti puno bolje

za 250. dan nego za 540. dan. Postoji puno parametara i slučajnosti koje nisu sadržane u SEIRD modelu i ne možemo očekivati kako će model uspješno, ako i ikako predvidjeti val koji slijedi iza 540. dana.



(a)



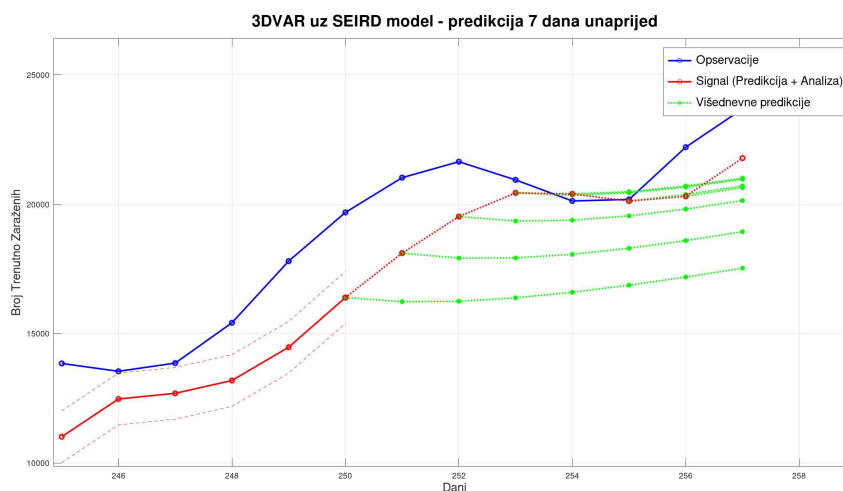
(b)

Slika 5.9: (a) Predikcija širenja zaraze COVID-19 sljedećih 7 dana s početkom u 250. danu. (b) Ažurirana predikcija širenja zaraze COVID-19 u 251. danu.

Krenimo od sedmodnevne predikcije u 250. danu od početka pandemije. Na slici 5.9(a) vidimo rezultat. Zelenom bojom je označena predikcija i primjećujemo kako podcjenjuje rast zaraženih osoba. No to neznamo jer se budućnost očito još nije dogodila.

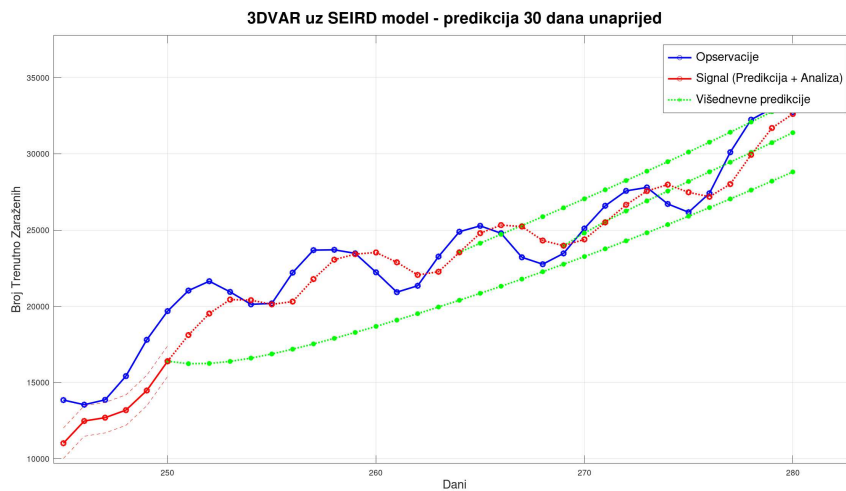
Sutradan dobivamo podatak o broju zaraženih (opservacija) i uzevši tu opservaciju asimiliramo našu predikciju. Analizom usklađujemo predikciju i opservaciju kako bi dobili procjenu stanja sustava. Sada ponovno radimo predikciju, ali ovog puta na sljedećih 6 dana kako bismo usporedili sadašnje predviđanje s jučerašnjim. Nova šestodnevna predikcija je prikazana na slici 5.9(b) kao gornja zelena linija. Uspoređivanjem vidimo razliku, ali i sličnost u trendu kretanja broja aktivno zaraženih.

Analogno radimo za svaki dan predikcijskog intervala. Rezultat predikcija tijekom 6 narednih dana vidimo na slici 5.10. Ne možemo biti zadovoljni prikazanim predviđanjima koja podcjenjuju nadolazeći val zaraze.



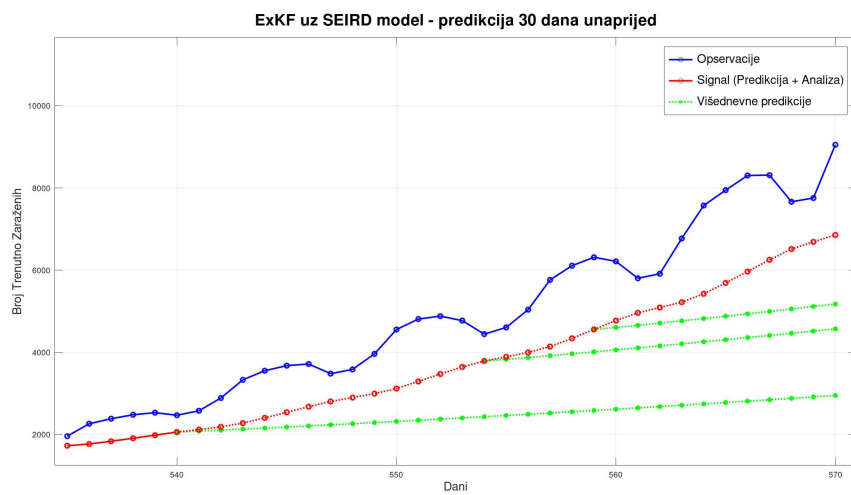
Slika 5.10: Predikcija širenja zaraze COVID-19 sljedećih 7 dana s početkom u 250. danu te predikcije ažurirane i napravljene svakog dana predikcijskog intervala.

Pogledajmo kako će izgledati predikcije ako želimo znati kretanje broja zaraženih kroz sljedećih 30 dana. Rezultati su prikazani na slici 5.11 gdje su rađene predikcije na početku predikcijskog intervala, nakon 15 dana i nakon 20 dana. Možemo primijetiti kako predikcije jako dobro određuju trend kretanja broja zaraženih te bi ovakva predikcija bila od velike koristi.



Slika 5.11: Predikcija širenja zaraze COVID-19 sljedećih 30 dana s početkom u 250. danu te predikcije ažurirane i napravljene 15. i 20. dana predikcijskog intervala. (3DVAR)

Pokušajmo napraviti predikciju sljedećih 30 dana na 540. danu od početka pandemije koristeći ExKF metodu za asimiliranje. Rezultati su na slici 5.12. Ne možemo biti zadovoljni ovakvim predikcijama jer podcjenjuju val zaraze, ali s obzirom na jednostavnost modela teško je tražiti nešto bolje. Razlog ovakve lošije procjene je, kao što smo već spomenuli, u tome što se predikcija radi u trenutku kada nema velikog trenda zaraženih.



Slika 5.12: Predikcija širenja zaraze COVID-19 sljedećih 30 dana s početkom u 250. danu te predikcije ažurirane i napravljene 15. i 20. dana predikcijskog intervala. (ExKF)

Bibliografija

- [1] Iserles A., *A First Course in the Numerical Analysis of Differential Equations*, Cambridge University Press, 2008.
- [2] Norman T. J. Bailey, *The mathematical theory of infectious diseases and its applications*, Griffin London, 1975, <https://nla.gov.au/nla.cat-vn702926>.
- [3] O. N. Bjørnstad, K. Shea, M. Krzywinski i N. Altman, *The SEIRS model for infectious disease dynamics.*, (2020), <https://doi.org/10.1038/s41592-020-0856-2>.
- [4] M. A. Freitag, *Numerical Linear Algebra in Data Assimilation*, (2020), <https://arxiv.org/abs/1912.13336v2>.
- [5] K. Law, A. Stuart i K. Zygalakis, *Data Assimilation - A Mathematical Introduction*, Springer, 2015.
- [6] Sarapa N., *Teorija vjerojatnosti*, Školska knjiga, 2002.

Sažetak

Ovaj rad predstavio je teoriju asimilacije podataka te ju primijenio na konkretnom problemu. Asimilacija podataka relativno je novo i interdisciplinarno područje koje se i dalje razvija. To je metoda koja kombinira opservacije stanja sustava s rezultatima modela kako bi se poboljšala procjena stanja sustava pa samim time i rezultati modela. Koristeći asimilaciju podataka napravljene su predikcije širenja zaraze COVID-19 u Hrvatskoj.

Prvotno je iznesena matematička teorija potrebna za daljnju izgradnju matematičkih modela i algoritama asimilacije podataka. Obrađeni su osnovni pojmovi iz teorije vjerojatnosti s naglaskom na Bayesovskoj statistici, teorije dinamičkih sustava s naglaskom na dinamičke sustave s kontrolom i numeričkog rješavanja inicijalnog problema sustava običnih diferencijalnih jednadžbi.

Opisan je osnovni matematički okvir za razumijevanje asimilacije podataka. Naglasak je na sekvencijalnoj asimilaciji podataka (tzv. filtriranje). U sekvencijalnoj asimilaciji podataka korekcije procjene modela rade se kad god su dostupne nove opservacije, tj. opservacije se uključuju u model sekvencijalno.

Potom su uzeti osnovni primjeri linearnog i nelinearnog inicijalnog problema sustava ODJ-a. Obrađena je metoda Kalmanovog filtera koja radi samo za linearne probleme sa šumovima koji su normalno distribuirani. Na nju su nadograđene dodatne dvije metode: 3DVAR i Prošireni Kalmanov filter (ExKF - Extended Kalman Filter). Ove dvije metode su dobivene generalizacijom minimizacijskog pristupa te rješavaju i nelinearne probleme. Na primjerima su testirana sva tri algoritma.

Uvedeni su matematički modeli širenja virusa i infekcija. Korišteni su klasifikacijski modeli za širenje infektivnih bolesti. Teorijski je opisan osnovni SIR model. Na njega su nadograđena dodatna dva modela: SEIR i SEIRD. Oni su bili korišteni u analizi širenja COVID-19.

Sintezom ovih znanja analizirana je pandemija COVID-19 u Hrvatskoj. Napravljene su dugoročne predikcije kretanja infekcije (7 i 30 dana). Rezultati su daleko od savršenih i daljnji rad je potreban. Cilj je pokazati potencijal sekvencijalne asimilacije podataka u predviđanju valova širenja infekcije što je i postignuto. Uz mnogobrojne nadogradnje koje su moguće kako na samom modelu širenja infekcije pa tako i na samoj metodi asimiliranja može se dobiti kvalitetna predikcija naglog širenja virusa koja spašava živote.

Summary

This work presented the theory of data assimilation and applied it to a concrete problem. Data assimilation is a relatively new and interdisciplinary field which is still developing. It is a method that combines the observations of the system state with model results to improve the estimation of the system state and thus the model results. Predictions of the spread of the COVID-19 infection in Croatia are made using data assimilation.

Firstly, mathematical theory needed for further development of mathematical models and data assimilation algorithms is carried out. Basic terms from probability theory focusing on Bayesian statistics, dynamical systems theory focusing on controlled dynamical systems and numerical solving initial-value problems of ordinary differential equations are introduced.

Description of the basic mathematical framework for understanding data assimilation is given. The emphasis is on sequential data assimilation (the so-called filtering). In sequential data assimilation corrections of model prediction are made whenever new observations are available, i.e. observations are included into model sequentially.

Basic examples of linear and non-linear initial-value problem of ordinary differential equations are given. Kalman filter is described, an algorithm that works for linear dynamics and noises with normal distribution. Two additional methods are built up on it: 3DVAR and Extended Kalman filter (ExKF). These two methods are obtained through generalization of minimization principle and they solve non-linear problems. All three algorithms are tested on examples.

Then there are given mathematical models of virus spread. The focus is on compartmental models. Theory of SIR model as basic one is described. On top of it are built SEIR and SEIRD models. They are used in COVID-19 analysis.

With everything given so far, the spread of COVID-19 in Croatia is analyzed. Attempts of long-term predictions of infection spread (7 and 30 days) are made. Results are far from perfect and further work is needed. The aim is to show the potential of sequential data assimilation in predicting big waves of virus spread and it was accomplished. With multitude of possible upgrades on the epidemic model and on the assimilation method used, it can be obtained to get high quality predictions of sudden virus spread which saves lives.

Životopis

Juraj Bilušković rođen je 10. prosinca 1999. godine u Zagrebu. Završio je Gimnaziju Lucijana Vranjanina u Zagrebu 2018. godine. Iste godine upisao je Preddiplomski sveučilišni studij Matematika na Prirodoslovno-matematičkom fakultetu Sveučilišta u Zagrebu. Nakon završetka Preddiplomskog studija upisao je Diplomski sveučilišni studij Financijska i poslovna matematika tijekom kojeg je imao studentski posao u području umjetne inteligencije i podatkovnih znanosti (tvrtka *AITech*). Prilikom obrazovanja usavršio je tečno govorenje engleskog te pasivnu razinu njemačkog jezika.

Dodatak - Octave kodovi

Kodovi Dinamičkih sustava

Kod za slike 3.1 i 3.2 (linearni sustav ODJ-a):

```
1 J = 10; % J = 50;
2 n=2; I=eye(n,n);
3
4 % Parametri modela
5 a=-1.5;b=0.5;c=0.25;d=0;
6 L = [a,b;c,d];
7
8 % RK3 parametri
9 c1=1/6;c2=2/3;c3=1/6;
10 a1=0;a2=1/2;a3=1;
11 A21=1/2;A31=-1;A32=2;
12
13 v = zeros(n,J*100+1);
14 v(:,1)=[1,1]';
15
16 h = 0.01;
17 t_plot = 0:h:J;
18 M = I + h*(c1*L + c2*(L + h*A21*(L^2)) + c3*(L + h*(A31 + ...
    A32)*(L^2) + (h^2)*A32*A21*(L^3)));
19
20 for j=1:(J*100)
21     v(:,j+1) = M*v(:,j);
22 end
23
24 % ODE45 rjesenje
25 f = @(t,vs) [-1.5*vs(1) + 0.5*vs(2);
26             0.25*vs(1)];
27 [t,vt] = ode45(f,[0 J],[1 1]); % Rjesenje dobiveno s RK4/5 metodom
28
29 figure;
30 plot(t,vt(:,1),'ro-','LineWidth',2),hold;
```

```

31 plot(t_plot,v(1,:), 'b', 'LineWidth',2);
32 lgd=legend('ode45', 'RK3');
33 set(lgd, 'FontSize',10);
34 title('v_1', 'FontSize',15);
35
36 figure;
37 plot(t,vt(:,2), 'ro-', 'LineWidth',2),hold;
38 plot(t_plot,v(2,:), 'b', 'LineWidth',2);
39 lgd=legend('ode45', 'RK3');
40 set(lgd, 'FontSize',10);
41 title('v_2', 'FontSize',15);

```

Kod za sliku 3.3 (Lorenzov sustav):

```

1 % Potrebne funkcije:
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 function retval = lorenz63 (v)
4 sigma = 10;
5 r = 28;
6 b = 8/3;
7 retval = [-sigma*v(1) + sigma*v(2);
8           r*v(1) - v(2) - v(1)*v(3);
9           -b*v(3) + v(1)*v(2)];
10 endfunction
11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12
13 J = 100;n=3;
14
15 % Parametri Lorenzovog sustava
16 sigma = 10;
17 r = 28;
18 b = 8/3;
19
20 % RK3 parametri
21 c1=1/6;c2=2/3;c3=1/6;
22 a1=0;a2=1/2;a3=1;
23 A21=1/2;A31=-1;A32=2;
24
25 % Korak h
26 h = 0.01;
27
28 vt = zeros(n,J+1);
29 vt(:,1)=[1;1;1];
30 for j=1:(10000)
31     Psi1 = lorenz63(vt(:,j));
32     Psi2 = lorenz63(vt(:,j) + h*A21*Psi1);
33     Psi3 = lorenz63(vt(:,j) + h*(A31*Psi1 + A32*Psi2));

```

```

34 vt(:,j+1) = vt(:,j) + h*(c1*Psi1 + c2*Psi2 + c3*Psi3);
35 end
36
37 figure;plot(vt(1,:),vt(2,:));
38 title('v_1 vs v_2', 'FontSize',15);
39 figure;plot(vt(2,:),vt(3,:));
40 title('v_2 vs v_3', 'FontSize',15);
41 figure;plot3(vt(1,:),vt(2,:),vt(3,:));grid on;
42 title('v_1 vs v_2 vs v_3', 'FontSize',15);

```

Kodovi algoritama (KF, 3DVAR, ExKF) testiranih na primjerima

Kod za slike 3.4 i 3.5 (Kalmanov filter testiran na linearnom primjeru):

```

1 J = 10;
2 n=2;I=eye(n,n);
3
4 % Varijance sumova
5 theta = 1;gamma = 1; % gamma = 10;
6 E = I*(theta^2);G = I*(gamma^2);
7
8 % RK3 parametri
9 c1=1/6;c2=2/3;c3=1/6;
10 a1=0;a2=1/2;a3=1;
11 A21=1/2;A31=-1;A32=2;
12
13 % Parametri dinamičkog sustava
14 a=-1.5;b=0.5;c=0.25;d=0;
15 L = [a,b;c,d];
16 h = 0.01;
17 t = 0:h:J;
18 M = I + h*(c1*L + c2*(L + h*A21*(L^2)) + c3*(L + h*(A31 + ...
    A32)*(L^2) + (h^2)*A32*A21*(L^3)));
19 H = I;
20
21 seed = 7; randn('state',seed);
22 %%% Istiniti Signal
23 vt = zeros(n,J*100+1);y = zeros(n,J*100);
24 vt(:,1)=[1;1];
25 for j=1:(J*100)
26 vt(:,j+1) = M*vt(:,j) + theta*randn(n,1);
27 y(:,j) = H*vt(:,j+1) + gamma*randn(n,1);
28 end
29
30 %%% Asimilacija podataka

```

```

31 m = zeros(n,J*100+1);C = zeros(n,n,J*100+1);
32
33 % Pocetne vrijednosti
34 m0 = [1;1];C0 = I;
35 m(:,1) = m0 + sqrtm(C0)*randn(n,1);
36 C(:, :, 1) = C0;
37
38 for j=1:(J*100)
39     mhat = M*m(:, j);
40     Chat = M*C(:, :, j)*M' + E;
41
42     S = H*Chat*H' + G;
43     K = Chat*H'*inv(S);
44     m(:, j+1) = (I - K*H)*mhat + K*y(:, j);
45     C(:, :, j+1) = (I - K*H)*Chat;
46 end
47
48 %%% Grafovi
49 labelFont = 15;
50 titleFont = 20;
51 lineWidth = 2;
52 lgdLocation = 'east';
53 lgdFontSize = 15;
54
55 figure;js=J*10;
56 plot([0:js],vt(1,1:js+1),'b','LineWidth',2);hold;
57 plot([0:js],m(1,1:js+1),'r','LineWidth',2);
58 plot([0:js],m(1,1:js+1)+reshape(sqrt(C(1,1,1:js+1)),1,js+1),'r--');
59 plot([0:js],m(1,1:js+1)-reshape(sqrt(C(1,1,1:js+1)),1,js+1),'r--');
60 hold;
61 grid;xlabel('Iteracije, i');
62 lgd = legend('Istiniti Signal (v^+1)', 'Signal(v1)', 'Pouzdaní ...
        interval');
63 set(lgd,'FontSize',lgdFontSize);
64 title('KF - linearni primjer; v^+1 i v1', 'FontSize',titleFont);
65
66 figure;js=J*10;
67 plot([1:js],y(1,1:js),'g','LineWidth',2);hold;
68 plot([0:js],m(1,1:js+1),'r','LineWidth',2);
69 plot([0:js],m(1,1:js+1)+reshape(sqrt(C(1,1,1:js+1)),1,js+1),'r--');
70 plot([0:js],m(1,1:js+1)-reshape(sqrt(C(1,1,1:js+1)),1,js+1),'r--');
71 hold;
72 grid;xlabel('Iteracije, i');
73 lgd = legend('Opservacije (y1)', 'Signal(v1)', 'Pouzdaní ...
        interval');
74 set(lgd,'FontSize',lgdFontSize);
75 title('KF - linearni primjer; y1 i v1', 'FontSize',titleFont);

```

```

76
77 figure;js=J*10;
78 plot ([0:js],vt (2,1:js+1), 'b', 'LineWidth',2);hold;
79 plot ([0:js],m(2,1:js+1), 'r', 'LineWidth',2);
80 plot ([0:js],m(2,1:js+1)+reshape (sqrt (C(2,2,1:js+1)),1,js+1), 'r--');
81 plot ([0:js],m(2,1:js+1)-reshape (sqrt (C(2,2,1:js+1)),1,js+1), 'r--');
82 hold;
83 grid;xlabel ('Iteracije, i');
84 lgd = legend ('Istiniti Signal (v^+2)', 'Signal(v.2)', 'Pouzdanje ...
interval');
85 set (lgd, 'FontSize',lgdFontSize);
86 title ('KF - linearni primjer; v^+2 i v.2', 'FontSize',titleFont);
87
88 figure;js=J*10;
89 plot ([1:js],y(2,1:js), 'g', 'LineWidth',2);hold;
90 plot ([0:js],m(2,1:js+1), 'r', 'LineWidth',2);
91 plot ([0:js],m(2,1:js+1)+reshape (sqrt (C(2,2,1:js+1)),1,js+1), 'r--');
92 plot ([0:js],m(2,1:js+1)-reshape (sqrt (C(2,2,1:js+1)),1,js+1), 'r--');
93 hold;
94 grid;xlabel ('Iteracije, i');
95 lgd = legend ('Opservacije (y.2)', 'Signal(v.2)', 'Pouzdanje ...
interval');
96 set (lgd, 'FontSize',lgdFontSize);
97 title ('KF - linearni primjer; y.2 i v.2', 'FontSize',titleFont);

```

Kodovi za slike 3.6 i 3.7 (3DVAR i ExKF testirani na Lorenzovom sustavu):

```

1  %%% 3DVAR testiran na Lorenzovom sustavu %%%
2
3  % Potrebne funkcije:
4  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5  function retval = lorenz63 (v)
6  sigma = 10;
7  r = 28;
8  b = 8/3;
9  retval = [-sigma*v(1) + sigma*v(2);
10           r*v(1) - v(2) - v(1)*v(3);
11           -b*v(3) + v(1)*v(2)];
12 endfunction
13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14 J = 10000;n=3;I=eye (n,n);
15 theta = 0.1;gamma = 1;
16 E = I*(theta^2);G = I*(gamma^2);
17 H = I;
18 h = 0.01;
19
20 % RK3 parametri

```

```

21 c1=1/6;c2=2/3;c3=1/6;
22 a1=0;a2=1/2;a3=1;
23 A21=1/2;A31=-1;A32=2;
24
25 seed = 7; randn('state',seed);
26 %% Istiniti Signal
27 vt = zeros(n,J+1);y = zeros(n,J);
28 vt(:,1)=[1;1;1];
29 for j=1:J
30     Psi1 = lorenz63(vt(:,j));
31     Psi2 = lorenz63(vt(:,j) + h*A21*Psi1);
32     Psi3 = lorenz63(vt(:,j) + h*(A31*Psi1 + A32*Psi2));
33     vt(:,j+1) = vt(:,j) + h*(c1*Psi1 + c2*Psi2 + c3*Psi3) + ...
        theta*randn(n,1);
34     y(:,j) = H*vt(:,j+1) + gamma*randn(n,1);
35 end
36
37 %% Asimilacija podataka
38 m = zeros(n,J+1);C = I;
39 % Pocetne vrijednosti
40 m0 = [1;1;1];C0 = C;
41 m(:,1) = m0 + sqrtm(C0)*randn(n,1);
42
43 S = H*C*H' + G;
44 K = C*H'*inv(S);
45
46 for j=1:J
47     % Predikcija
48     mhat = zeros(n,1);
49     Psi1 = lorenz63(m(:,j));
50     Psi2 = lorenz63(m(:,j) + h*A21*Psi1);
51     Psi3 = lorenz63(m(:,j) + h*(A31*Psi1 + A32*Psi2));
52     mhat = m(:,j) + h*(c1*Psi1 + c2*Psi2 + c3*Psi3);
53
54     % Analiza
55     m(:,j+1) = (I - K*H)*mhat + K*y(:,j);
56 end
57
58 %% Grafovi
59 figure;plot(m(1,:),m(2,:))
60 title('3DVAR - v_1 vs v_2', 'FontSize',15);
61 figure;plot(m(2,:),m(3,:))
62 title('3DVAR - v_2 vs v_3', 'FontSize',15);
63 figure;plot3(m(1,:),m(2,:),m(3,:));grid on;
64 title('3DVAR', 'FontSize',15);

```

```

1  %%% ExKF testiran na Lorenzovom sustavu %%%
2
3  % Potrebne funkcije:
4  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5  function retval = lorenz63 (v)
6  sigma = 10;
7  r = 28;
8  b = 8/3;
9  retval = [-sigma*v(1) + sigma*v(2);
10           r*v(1) - v(2) - v(1)*v(3);
11           -b*v(3) + v(1)*v(2)];
12 endfunction
13
14 function retval = DPhi_lorenz63 (m,h,Psi1,Psi2)
15
16 c1=1/6;c2=2/3;c3=1/6;
17 a1=0;a2=1/2;a3=1;
18 A21=1/2;A31=-1;A32=2;
19 I = eye(3,3);
20
21 DPsi1 = Dlorenz63(m);
22 DPsi2 = Dlorenz63(m + h*A21*Psi1);
23 DPsi3 = Dlorenz63(m + h*(A31*Psi1 + A32*Psi2));
24
25 retval1 = [c1*(DPsi1(1,:)) + c2*(DPsi2(1,:)) + c3*(DPsi3(1,:));
26           c1*(DPsi1(2,:)) + c2*(DPsi2(2,:)) + c3*(DPsi3(2,:));
27           c1*(DPsi1(3,:)) + c2*(DPsi2(3,:)) + c3*(DPsi3(3,:))];
28 retval = I + h*retval1;
29
30 endfunction
31 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
32 J = 10000;n=3;I=eye(n,n);
33 theta = 0.1;gamma = 1;
34 E = I*(theta^2);G = I*(gamma^2);
35 H = I;
36 h = 0.01;
37
38 % RK3 parametri
39 c1=1/6;c2=2/3;c3=1/6;
40 a1=0;a2=1/2;a3=1;
41 A21=1/2;A31=-1;A32=2;
42
43 seed = 7; randn('state',seed);
44 %%% Istiniti Signal
45 vt = zeros(n,J+1);y = zeros(n,J);
46 vt(:,1)=[1;1;1];
47 for j=1:J

```

```

48 Psi1 = lorenz63(vt(:,j));
49 Psi2 = lorenz63(vt(:,j) + h*A21*Psi1);
50 Psi3 = lorenz63(vt(:,j) + h*(A31*Psi1 + A32*Psi2));
51 vt(:,j+1) = vt(:,j) + h*(c1*Psi1 + c2*Psi2 + c3*Psi3) + ...
    theta*randn(n,1);
52 y(:,j) = H*vt(:,j+1) + gamma*randn(n,1);
53 end
54
55 %% Asimilacija podataka
56 m = zeros(n,J+1);C = zeros(n,n,J+1);
57 % Pocetne vrijednosti
58 m0 = [1;1;1];C0 = I;C(:, :, 1) = I;
59 m(:,1) = m0 + sqrtm(C0)*randn(n,1);
60
61 for j=1:J
62     % Predikcija
63     mhat = zeros(n,1);
64     Psi1 = lorenz63(m(:,j));
65     Psi2 = lorenz63(m(:,j) + h*A21*Psi1);
66     Psi3 = lorenz63(m(:,j) + h*(A31*Psi1 + A32*Psi2));
67     mhat = m(:,j) + h*(c1*Psi1 + c2*Psi2 + c3*Psi3);
68     Dphi = DPhi_lorenz63 (m(:,j),h,Psi1,Psi2);
69     Chat = Dphi*C(:, :, j)*Dphi' + E;
70
71     % Analiza
72     S = H*Chat*H' + G;
73     K = Chat*H'*inv(S);
74     m(:,j+1) = (I - K*H)*mhat + K*y(:,j);
75     C(:, :, j+1) = (I - K*H)*Chat;
76 end
77
78 %% Grafovi
79 figure;plot(m(1,:),m(2,:))
80 title('ExKF - v_1 vs v_2', 'FontSize',15);
81 figure;plot(m(2,:),m(3,:))
82 title('ExKF - v_2 vs v_3', 'FontSize',15);
83 figure;plot3(m(1,:),m(2,:),m(3,:));grid on;
84 title('ExKF', 'FontSize',15);

```


Kodovi simulacija epidemioloških modela (SIR, SEIR, SEIRD)

Kod za sliku 4.1 (simulacija SIR modela):

```
1 J = 100;
2 N = 1000;
3 beta=0.4; lambda=0.04;
4
5 S = zeros(J+1,1); I = zeros(J+1,1); R = zeros(J+1,1);
6 I(1)=5; S(1)=N-I(1); R(1)=0;
7 seed = 5; randn('state',seed);
8
9 for j=1:J
10     S(j+1) = S(j) - I(j)*beta*(S(j)/N);
11     I(j+1) = I(j) + I(j)*beta*(S(j)/N) - lambda*I(j);
12     R(j+1) = R(j) + lambda*I(j);
13 end
14
15 figure(1); plot(S, 'LineWidth', 2);
16 hold on; plot(I, 'LineWidth', 2);
17 hold on; plot(R, 'LineWidth', 2);
18 lgd=legend('S(Susceptible)', 'I(Infected)', 'R(Recovered)');
19 set(lgd, 'FontSize', 15);
```

Kod za sliku 4.2 (simulacija SEIR modela):

```
1 J = 100;
2 N = 1000;
3 beta=0.4; lambda=0.04; eps=1/5;
4
5 S = zeros(J+1,1); E = zeros(J+1,1); I = zeros(J+1,1); R = zeros(J+1,1);
6 I(1)=5; S(1)=N-I(1); E(1)=0; R(1)=0;
7 seed = 5; randn('state',seed);
8
9 for j=1:J
10     S(j+1) = S(j) - I(j)*beta*(S(j)/N);
11     E(j+1) = E(j) - eps*E(j) + I(j)*beta*(S(j)/N);
12     I(j+1) = I(j) - lambda*I(j) + eps*E(j);
13     R(j+1) = R(j) + lambda*I(j);
14 end
15
16 figure(1); plot(S, 'LineWidth', 2);
17 hold on; plot(E, 'LineWidth', 2);
18 hold on; plot(I, 'LineWidth', 2);
19 hold on; plot(R, 'LineWidth', 2);
20 lgd = legend('S(Susceptible)', 'E(Exposed)', 'I(Infected)');
```

```

21         , 'R(Recovered) ');
22 set(lgd, 'FontSize', 15);
23 set(lgd, 'location', 'eastoutside');

```

Kod za sliku 4.3 (simulacija SEIRD modela):

```

1 J = 100;
2 N = 1000;
3 beta=0.4;lambda=0.04;eps=1/5;mu=0.01;
4
5 S = zeros(J+1,1);E = zeros(J+1,1);I = zeros(J+1,1);R = ...
   zeros(J+1,1);D = zeros(J+1,1);
6 I(1)=5;S(1)=N-I(1);E(1)=0;R(1)=0;D(1)=0;
7 seed = 5; randn('state',seed);
8
9 for j=1:J
10  S(j+1)= S(j) - I(j)*beta*(S(j)/N);
11  E(j+1)= E(j) - eps*E(j) + I(j)*beta*(S(j)/N);
12  I(j+1)= I(j) - lambda*I(j) + eps*E(j) - mu*I(j);
13  R(j+1)= R(j) + lambda*I(j);
14  D(j+1)= D(j) + mu*I(j);
15 end
16
17 figure(1);
18 plot(S, 'LineWidth', 2);
19 hold on; plot(E, 'LineWidth', 2);
20 hold on; plot(I, 'LineWidth', 2);
21 hold on; plot(R, 'LineWidth', 2);
22 hold on; plot(D, 'LineWidth', 2);
23 lgd = legend('S(Susceptible)', 'E(Exposed)', 'I(Infected)'
24             , 'R(Recovered)', 'D(Deceased)');
25 set(lgd, 'FontSize', 15);
26 set(lgd, 'location', 'eastoutside');

```

Kodovi jednodnevnih predikcija

Kod 3DVAR algoritma uz SEIR model (slika 5.2):

```
1  %%% Ucitavanje podataka %%%
2  data = csvread('0-Data/COVID-19_data.csv');
3  y = data(:,11); % Trenutno zarazeni s COVID-19
4
5  %%% Parametri %%%
6  J = length(y);
7  H = [0,0,1,0];
8
9  %%% Opservacijski sum
10 gamma = 10000;
11 G = gamma^2;
12
13 %%% SEIR parametri
14 N = 4000000;
15 beta=0.15;D=10;lambda=1/D;eps=1/5;
16
17 %%% Pocetne vrijednosti
18 m = zeros(4,J+1);I = eye(4);
19 m(:,1) = [N-1;1;1;0];
20 C = I*gamma^2;
21
22 %%% 3DVAR
23 S = H*C*H' + G;
24 K = C*H'*inv(S);
25
26 %%% Asimilacija
27 for j=1:J
28     % Predikcija
29     mhat = zeros(4,1);
30     mhat(1) = m(1,j) - m(3,j)*beta*(m(1,j)/N);
31     mhat(2) = m(2,j)*(1-eps) + m(3,j)*beta*m(1,j)/N;
32     mhat(3) = m(3,j)*(1-lambda) + eps*m(2,j);
33     mhat(4) = m(4,j) + lambda*m(3,j);
34
35     % Analiza
36     m(:,j+1) = (I - K*H)*mhat + K*y(j);
37 end
38
39 %%% GRAFOVI
40 labelFont = 15;
41 titleFont = 20;
42 lineWidth = 2;
43 lgdLocation = 'east';
```

```

44 lgdFontSize = 15;
45
46 %% PRVIH 500 DANA
47 js=501;
48 figure;plot([1:js],y(1:js),'LineWidth',lineWidth);hold;
49 plot([0:js-1],m(3,1:js),'r','LineWidth',lineWidth);
50 plot([0:js-1],m(3,1:js)+sqrt(C(3,3)),'r--');
51 plot([0:js-1],m(3,1:js)-sqrt(C(3,3)),'r--');
52 hold;grid;
53 ylabel('Broj Trenutno Zarazenih','FontSize',labelFont);
54 xlabel('Dani','FontSize',labelFont);
55 title('3DVAR uz SEIR model - prvih 500 dana','FontSize',titleFont);
56 legend('Opservacije','Signal','Pouzdaní ...
Interval','location',lgdLocation,'FontSize',lgdFontSize);
57 %% SVI DANI
58 js = J;
59 figure;plot([1:js],y,'LineWidth',lineWidth);hold;
60 plot([0:js],m(3,:),'r','LineWidth',lineWidth);
61 plot([0:js],m(3,:)+sqrt(C(3,3)),'r--');
62 plot([0:js],m(3,:)-sqrt(C(3,3)),'r--');
63 hold;grid;
64 ylabel('Broj Trenutno Zarazenih','FontSize',labelFont);
65 xlabel('Dani','FontSize',labelFont);
66 title('3DVAR uz SEIR model - cijeli period','FontSize',titleFont);
67 legend('Opservacije','Signal','Pouzdaní ...
Interval','location',lgdLocation,'FontSize',lgdFontSize);

```

Kod 3DVAR algoritma uz SEIRD model (slike 5.4 i 5.5):

```

1 %% Ucitavanje podataka %%
2 data = csvread('0-Data\COVID-19_data.csv');
3 y = zeros(2,length(data(:,1)));
4 y(1,:) = data(:,11); % Trenutno zarazeni
5 y(2,:) = data(:,6); % Kumulativno preminuli od COVID-19
6
7 %% Parametri %%
8 J = length(y);
9 H = [0,0,1,0,0;
10      0,0,0,0,1];
11
12 %% Opservacijski sum
13 gamma = 10000;
14 G = gamma^2;
15
16 %% SEIRD parametri
17 N = 4000000;
18 beta=0.15;D=10;lambda=1/D;eps=1/5;mu=0.01;

```

```

19
20 %%% Pocetne vrijednosti
21 I = eye(5);
22 C = I*gamma^2;
23 m = zeros(5,J+1);
24 m(:,1) = [N-1;1;1;0;0];
25
26 %%% 3DVAR
27 S = H*C*H' + G;
28 K = C*H'*inv(S);
29
30 %%% Asimilacija
31 for j=1:J
32     mhat = zeros(5,1);
33     mhat(1) = m(1,j) - m(3,j)*beta*(m(1,j)/N);
34     mhat(2) = m(2,j) - eps*m(2,j) + m(3,j)*beta*(m(1,j)/N);
35     mhat(3) = m(3,j) + eps*m(2,j) - lambda*m(3,j) - mu*m(3,j);
36     mhat(4) = m(4,j) + lambda*m(3,j);
37     mhat(5) = m(5,j) + mu*m(3,j);
38
39     m(:,j+1) = (I - K*H)*mhat + K*y(:,j);
40 end
41
42 %%% GRAFOVI
43 labelFont = 15;
44 titleFont = 20;
45 lineWidth = 2;
46 lgdLocation = 'east';
47 lgdFontSize = 15;
48
49 %%% PRVIH 500 DANA
50 js=501;
51 figure;plot([1:js],y(1,1:js),'LineWidth',lineWidth);hold;
52 plot([0:js-1],m(3,1:js),'r','LineWidth',lineWidth);
53 plot([0:js-1],m(3,1:js)+sqrt(C(3,3)),'r--');
54 plot([0:js-1],m(3,1:js)-sqrt(C(3,3)),'r--');
55 hold;grid;
56 ylabel('Broj Trenutno Zarazenih','FontSize',labelFont);
57 xlabel('Dani','FontSize',labelFont);
58 title('3DVAR uz SEIRD model - prvih 500 dana','FontSize',titleFont);
59 legend('Opservacije','Signal','Pouzdaní ...
        Interval','location',lgdLocation,'FontSize',lgdFontSize);
60
61 figure;plot([1:js],y(2,1:js),'g','LineWidth',lineWidth);hold;
62 plot([0:js-1],m(5,1:js),'y','LineWidth',lineWidth);
63 plot([0:js-1],m(5,1:js)+sqrt(C(5,5)),'r--');
64 plot([0:js-1],m(5,1:js)-sqrt(C(5,5)),'r--');

```

```

65 hold;grid;
66 ylabel('Broj Ukupno Preminulih','FontSize',labelFont);
67 xlabel('Dani','FontSize',labelFont);
68 title('3DVAR uz SEIRD model - prvih 500 dana','FontSize',titleFont);
69 legend('Opservacije','Signal','Pouzdaní ...
        Interval','location',lgdLocation,'Fontsize',lgdFontSize);
70
71 %%% SVI DANI
72 js = J;
73 figure;plot([1:js],y(1,1:js),'LineWidth',lineWidth);hold;
74 plot([0:js],m(3,:),'r','LineWidth',lineWidth);
75 plot([0:js],m(3,)+sqrt(C(3,3)),'r--');
76 plot([0:js],m(3,)-sqrt(C(3,3)),'r--');
77 hold;grid;
78 ylabel('Broj Trenutno Zarazenih','FontSize',labelFont);
79 xlabel('Dani','FontSize',labelFont);
80 title('3DVAR uz SEIRD model - cijeli period','FontSize',titleFont);
81 legend('Opservacije','Signal','Pouzdaní ...
        Interval','location',lgdLocation,'Fontsize',lgdFontSize);
82
83 figure;plot([1:js],y(2,1:js),'g','LineWidth',lineWidth);hold;
84 plot([0:js-1],m(5,1:js),'y','LineWidth',lineWidth);
85 plot([0:js-1],m(5,1:js)+sqrt(C(5,5)),'r--');
86 plot([0:js-1],m(5,1:js)-sqrt(C(5,5)),'r--');
87 hold;grid;
88 ylabel('Broj Ukupno Preminulih','FontSize',labelFont);
89 xlabel('Dani','FontSize',labelFont);
90 title('3DVAR uz SEIRD model - cijeli period','FontSize',titleFont);
91 legend('Opservacije','Signal','Pouzdaní ...
        Interval','location',lgdLocation,'Fontsize',lgdFontSize);

```

Kod ExKF algoritma uz SEIR model (slika 5.3):

```

1 % Potrebne funkcije:
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 function Dphi = DPhi_SEIR (m,beta,eps,lambda,N)
4     bn = beta/N;
5     Dphi = [1-m(3)*bn, 0, -m(1)*bn, 0;
6             m(3)*bn, 1-eps, m(1)*bn, 0;
7             0, eps, 1-lambda, 0;
8             0, 0, lambda, 1];
9 endfunction
10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
11 %%% Ucitavanje podataka %%%
12 data = csvread('0-Data\COVID-19_data.csv');
13 y = data(:,11); % Trenutno zarazeni
14

```

```

15  %%% Parametri %%%
16  J = length(y);
17  H = [0,0,1,0];
18
19  %%% Sumovi
20  n = 4; I = eye(n);
21  gamma = 10000; sigma = 1000;
22  G = gamma^2; E = I*sigma^2;
23
24  %%% SEIR parametri
25  N = 4000000;
26  beta=0.15; D=10; lambda=1/D; eps=1/5;
27
28  %%% Pocetne vrijednosti
29  C = zeros(n,n,J+1); m = zeros(n,J+1); cov_plot = zeros(n,J+1);
30  m(:,1) = [N-1;1;1;0];
31  C(:, :, 1) = I*gamma^2;
32  cov_plot(1) = 1;
33
34  %%% Asimilacija
35  for j=1:J
36      % Predikcija
37      mhat = zeros(n,1);
38      mhat(1) = m(1,j) - m(3,j)*beta*(m(1,j)/N);
39      mhat(2) = m(2,j) - eps*m(2,j) + m(3,j)*beta*(m(1,j)/N);
40      mhat(3) = m(3,j) + eps*m(2,j) - lambda*m(3,j);
41      mhat(4) = m(4,j) + lambda*m(3,j);
42      Dphi = Dphi_SEIR(m(:,j),beta,eps,lambda,N);
43      Chat = Dphi*C(:, :, j)*Dphi' + E;
44
45      % Analiza
46      S = H*Chat*H' + G;
47      K = Chat*H'*inv(S);
48      m(:,j+1) = (I - K*H)*mhat + K*y(j);
49      C(:, :, j+1) = (I - K*H)*Chat;
50
51      cov_plot(j+1) = C(3,3,j+1);
52  end
53
54  %%% GRAFOVI
55  labelFont = 15;
56  titleFont = 20;
57  lineWidth = 2;
58  lgdLocation = 'east';
59  lgdFontSize = 15;
60
61  %%% PRVIH 500 DANA

```

```

62 js=501;
63 figure;plot([1:js],y(1:js),'LineWidth',lineWidth);hold;
64 plot([0:js-1],m(3,1:js),'r','LineWidth',lineWidth);
65 plot([0:js-1],m(3,1:js)+sqrt(cov_plot(1:js)),'r--');
66 plot([0:js-1],m(3,1:js)-sqrt(cov_plot(1:js)),'r--');
67 hold;grid;
68 ylabel('Broj Trenutno Zarazenih','FontSize',labelFont);
69 xlabel('Dani','FontSize',labelFont);
70 title('ExKF uz SEIR model - prvih 500 dana','FontSize',titleFont);
71 legend('Opservacije','Signal','Pouzdaní ...
        Interval','location',lgdLocation,'Fontsize',lgdFontSize);
72
73 %%% SVI DANI
74 js = J;
75 figure;plot([1:js],y,'LineWidth',lineWidth);hold;
76 plot([0:js],m(3,:),'r','LineWidth',lineWidth);
77 plot([0:js],m(3,:)+sqrt(cov_plot(1:js+1)),'r--');
78 plot([0:js],m(3,:)-sqrt(cov_plot(1:js+1)),'r--');
79 hold;grid;
80 ylabel('Broj Trenutno Zarazenih','FontSize',labelFont);
81 xlabel('Dani','FontSize',labelFont);
82 title('ExKF uz SEIR model - cijeli period','FontSize',titleFont);
83 legend('Opservacije','Signal','Pouzdaní ...
        Interval','location',lgdLocation,'Fontsize',lgdFontSize);

```

Kod ExKF algoritma uz SEIRD model (slike 5.6 i 5.7):

```

1 % Potrebne funkcije:
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 function Dphi = DPhi_SEIRD (m,beta,eps,lambda,mu,N)
4     bn = beta/N;
5     Dphi = [1-m(3)*bn, 0, -m(1)*bn, 0, 0;
6             m(3)*bn, 1-eps, m(1)*bn, 0, 0;
7             0, eps, 1-lambda-mu, 0, 0;
8             0, 0, lambda, 1, 0;
9             0, 0, mu, 0, 1];
10 endfunction
11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12 %%% Ucitavanje podataka %%%
13 data = csvread('0-Data\COVID-19_data.csv');
14 y = zeros(2,length(data(:,1)));
15 y(1,:) = data(:,11); % Trenutno zarazeni
16 y(2,:) = data(:,6); % Kumulativno preminuli
17
18 %%% Parametri %%%
19 J = length(y(1,:));
20 H = [0,0,1,0,0;

```



```

21         0,0,0,0,1];
22
23     %%% Sumovi
24     n = 5; I = eye(n);
25     gamma = 10000; sigma = 1000;
26     G = gamma^2; E = I*sigma^2;
27
28     %%% SEIRD parametri
29     N = 4000000;
30     beta=0.15; D=10; lambda=1/D; eps=1/5; mu=0.01;
31
32     %%% Pocetne vrijednosti
33     C = zeros(n,n,J+1); m = zeros(n,J+1);
34     cov_plot = zeros(n,J+1); cov_d_plot = zeros(n,J+1);
35     C(:, :, 1) = I*gamma^2;
36     m(:, 1) = [N-1; 1; 1; 0; 0];
37     cov_plot(1) = gamma^2; cov_d_plot(1) = gamma^2;
38
39     %%% Asimilacija
40     for j=1:J
41         % Predikcija
42         mhat = zeros(n,1);
43         mhat(1) = m(1,j) - m(3,j)*beta*(m(1,j)/N);
44         mhat(2) = m(2,j) - eps*m(2,j) + m(3,j)*beta*(m(1,j)/N);
45         mhat(3) = m(3,j) + eps*m(2,j) - lambda*m(3,j) - mu*m(3,j);
46         mhat(4) = m(4,j) + lambda*m(3,j);
47         mhat(5) = m(5,j) + mu*m(3,j);
48         Dphi = DPhi_SEIRD(m(:,j),beta,eps,lambda,mu,N);
49         Chat = Dphi*C(:, :, j)*Dphi' + E;
50
51         % Analiza
52         S = H*Chat*H' + G;
53         K = Chat*H'*inv(S);
54         m(:, j+1) = (I - K*H)*mhat + K*y(:, j);
55         C(:, :, j+1) = (I - K*H)*Chat;
56
57         % Za graf pouzdanog intervala
58         cov_plot(j+1) = C(3,3,j+1);
59         cov_d_plot(j+1) = C(5,5,j+1);
60     end
61
62     %%% GRAFOVI
63     labelFont = 15;
64     titleFont = 20;
65     lineWidth = 2;
66     lgdLocation = 'east';
67     lgdFontSize = 15;

```

```

68 %% PRVIH 500 DANA
69 js=501;
70 figure;plot([1:js],y(1,1:js),'LineWidth',lineWidth);hold;
71 plot([0:js-1],m(3,1:js),'r','LineWidth',lineWidth);
72 plot([0:js-1],m(3,1:js)+sqrt(cov_plot(1:js)),'r--');
73 plot([0:js-1],m(3,1:js)-sqrt(cov_plot(1:js)),'r--');
74 hold;grid;
75 ylabel('Broj Trenutno Zarazenih','FontSize',labelFont);
76 xlabel('Dani','FontSize',labelFont);
77 title('ExKF uz SEIRD model - prvih 500 dana','FontSize',titleFont);
78 legend('Opservacije','Signal','Pouzdaní ...
    Interval','location',lgdLocation,'FontSize',lgdFontSize);
79
80 figure;plot([1:js],y(2,1:js),'g','LineWidth',lineWidth);hold;
81 plot([0:js-1],m(5,1:js),'y','LineWidth',lineWidth);
82 plot([0:js-1],m(5,1:js)+sqrt(cov_d_plot(1:js)),'r--');
83 plot([0:js-1],m(5,1:js)-sqrt(cov_d_plot(1:js)),'r--');
84 hold;grid;
85 ylabel('Broj Ukupno Preminulih','FontSize',labelFont);
86 xlabel('Dani','FontSize',labelFont);
87 title('ExKF uz SEIRD model - prvih 500 dana','FontSize',titleFont);
88 legend('Opservacije','Signal','Pouzdaní ...
    Interval','location',lgdLocation,'FontSize',lgdFontSize);
89 %% SVI DANI
90 js = J;
91 figure;plot([1:js],y(1,1:js),'LineWidth',lineWidth);hold;
92 plot([0:js],m(3,:), 'r','LineWidth',lineWidth);
93 plot([0:js],m(3,)+sqrt(cov_plot(1:js+1)),'r--');
94 plot([0:js],m(3,)-sqrt(cov_plot(1:js+1)),'r--');
95 hold;grid;
96 ylabel('Broj Trenutno Zarazenih','FontSize',labelFont);
97 xlabel('Dani','FontSize',labelFont);
98 title('ExKF uz SEIRD model - cijeli period','FontSize',titleFont);
99 legend('Opservacije','Signal','Pouzdaní ...
    Interval','location',lgdLocation,'FontSize',lgdFontSize);
100
101 figure;plot([1:js],y(2,1:js),'g','LineWidth',lineWidth);hold;
102 plot([0:js-1],m(5,1:js),'y','LineWidth',lineWidth);
103 plot([0:js-1],m(5,1:js)+sqrt(cov_d_plot(1:js)),'r--');
104 plot([0:js-1],m(5,1:js)-sqrt(cov_d_plot(1:js)),'r--');
105 hold;grid;
106 ylabel('Broj Ukupno Preminulih','FontSize',labelFont);
107 xlabel('Dani','FontSize',labelFont);
108 title('ExKF uz SEIRD model - cijeli period','FontSize',titleFont);
109 legend('Opservacije','Signal','Pouzdaní ...
    Interval','location',lgdLocation,'FontSize',lgdFontSize);

```

Kodovi višednevnih predikcija

Kod za 3DVAR algoritam uz SEIRD model (slike 5.9, 5.10 i 5.11):

```
1  %%% Ucitavanje podataka %%%
2  data = csvread('..\0-Data\COVID-19_data.csv');
3  y = zeros(2,length(data(:,1)));
4  y(1,:) = data(:,11); % Trenutno zarazeni
5  y(2,:) = data(:,6); % Kumulativno preminuli
6
7  %%% Parametri %%%
8  J = length(y);
9  H = [0,0,1,0,0;
10      0,0,0,0,1];
11
12 %%% Opservacijski sum
13 gamma = 1000;
14 G = gamma^2;
15
16 %%% SEIRD parametri
17 N = 4000000;
18 beta=0.15;D=10;lambda=1/D;eps=1/5;mu=0.01;
19
20 %%% Pocetne vrijednosti
21 I = eye(5);
22 C = I*gamma^2;
23 m = zeros(5,J+1);
24 m(:,1) = [N-1;1;1;0;0];
25
26 %%% 3DVAR
27 S = H*C*H' + G;
28 K = C*H'*inv(S);
29
30 windowSize = 7; % windowSize = 30;
31 startPrediction = 250; % startPrediction = 540;
32 predictions = cell(windowSize);
33
34 # Asimiliraj do startPrediction
35 for i=1:startPrediction
36     mhat = zeros(5,1);
37     mhat(1) = m(1,i) - m(3,i)*beta*(m(1,i)/N);
38     mhat(2) = m(2,i) - eps*m(2,i) + m(3,i)*beta*(m(1,i)/N);
39     mhat(3) = m(3,i) + eps*m(2,i) - lambda*m(3,i) - mu*m(3,i);
40     mhat(4) = m(4,i) + lambda*m(3,i);
41     mhat(5) = m(5,i) + mu*m(3,i);
42
43     m(:,i+1) = (I - K*H)*mhat + K*y(:,i);
```

```

44 end
45 # Predvidi windowSize
46 for l=1:windowSize
47     mhat = zeros(5,1);
48     mhat(1) = m(1,startPrediction+l) - ...
         m(3,startPrediction+l)*beta*(m(1,startPrediction+l)/N);
49     mhat(2) = m(2,startPrediction+l) - eps*m(2,startPrediction+l) ...
         + m(3,startPrediction+l)*beta*(m(1,startPrediction+l)/N);
50     mhat(3) = m(3,startPrediction+l) + eps*m(2,startPrediction+l) ...
         - lambda*m(3,startPrediction+l) - mu*m(3,startPrediction+l);
51     mhat(4) = m(4,startPrediction+l) + lambda*m(3,startPrediction+l);
52     mhat(5) = m(5,startPrediction+l) + mu*m(3,startPrediction+l);
53     m(:,startPrediction+l+1) = mhat;
54 endfor
55 predictions{1} = m(3,startPrediction+2:startPrediction+windowSize+1);
56
57 # Asimiliraj i predvidaj startPrediction+windowSize-1
58 for j=1:(windowSize-1)
59     # Asimiliraj
60     mhat = zeros(5,1);
61     mhat(1) = m(1,startPrediction+j) - ...
         m(3,startPrediction+j)*beta*(m(1,startPrediction+j)/N);
62     mhat(2) = m(2,startPrediction+j) - eps*m(2,startPrediction+j) + ...
         m(3,startPrediction+j)*beta*(m(1,startPrediction+j)/N);
63     mhat(3) = m(3,startPrediction+j) + eps*m(2,startPrediction+j) - ...
         lambda*m(3,startPrediction+j) - mu*m(3,startPrediction+j);
64     mhat(4) = m(4,startPrediction+j) + lambda*m(3,startPrediction+j);
65     mhat(5) = m(5,startPrediction+j) + mu*m(3,startPrediction+j);
66     m(:,startPrediction+j+1) = (I - K*H)*mhat + ...
         K*y(:,startPrediction+j);
67
68     # Predvidi ostatak
69     for l=1:(windowSize-j)
70         mhat = zeros(5,1);
71         mhat(1) = m(1,startPrediction+l+j) - ...
             m(3,startPrediction+l+j)*beta*(m(1,startPrediction+l+j)/N);
72         mhat(2) = m(2,startPrediction+l+j) - ...
             eps*m(2,startPrediction+l+j) + ...
             m(3,startPrediction+l+j)*beta*(m(1,startPrediction+l+j)/N);
73         mhat(3) = m(3,startPrediction+l+j) + ...
             eps*m(2,startPrediction+l+j) - ...
             lambda*m(3,startPrediction+l+j) - ...
             mu*m(3,startPrediction+l+j);
74         mhat(4) = m(4,startPrediction+l+j) + ...
             lambda*m(3,startPrediction+l+j);
75         mhat(5) = m(5,startPrediction+l+j) + mu*m(3,startPrediction+l+j);
76         m(:,startPrediction+l+1+j) = mhat;

```

```

77     endfor
78     predictions{j+1} = ...
           m(3,startPrediction+2+j:startPrediction+windowSize+1);
79 end
80
81 mhat = m(:,startPrediction+windowSize+1);
82 m(:,startPrediction+windowSize+1) = (I - K*H)*mhat + ...
           K*y(:,startPrediction+windowSize);
83
84 %%% Grafovi %%%
85 labelFont = 15;
86 titleFont = 20;
87 lineWidth = 2;
88 lgdLocation = 'east';
89 lgdFontSize = 15;
90 addOps = 0;
91 startPlot = startPrediction - 5;
92
93 % 1
94 js=startPrediction+windowSize+1;
95 figure;
96 plot([startPlot:js+addOps-1],y(1,startPlot+1:js+addOps)
97      , 'bo-', 'LineWidth', lineWidth);
98 hold;
99 plot([startPlot:startPrediction],m(3,startPlot+1:startPrediction+1)
100     , 'LineWidth', lineWidth, 'ro-');
101 plot([startPrediction:js-1],[m(3,startPrediction+1),predictions{1}]
102     , 'g*:', 'LineWidth', lineWidth);
103 plot([startPlot:startPrediction]
104     ,m(3,startPlot+1:startPrediction+1)+sqrt(C(3,3)), 'r--');
105 plot([startPlot:startPrediction]
106     ,m(3,startPlot+1:startPrediction+1)-sqrt(C(3,3)), 'r--');
107 plot([startPrediction:startPrediction+windowSize]
108     ,m(3,startPrediction+1:startPrediction+windowSize+1), 'ro:'
109     , 'LineWidth', lineWidth);
110 hold;grid;
111 ylabel('Broj Trenutno Zarazenih','FontSize',labelFont);
112 xlabel('Dani','FontSize',labelFont);
113 title('3DVAR uz SEIRD model - predikcija 7 dana ...
           unaprijed','FontSize',titleFont);
114 legend('Opservacije','Signal (Predikcija + Analiza)','Visednevne ...
           predikcije','Pouzdani Interval','FontSize',lgdFontSize);
115
116 % 2
117 js=startPrediction+windowSize+1;
118 figure;
119 plot([startPlot:js+addOps-1],y(1,startPlot+1:js+addOps)

```

```

120     , 'bo-', 'LineWidth', lineWidth);
121 hold;
122 plot([startPlot:startPrediction], m(3, startPlot+1:startPrediction+1)
123     , 'LineWidth', lineWidth, 'ro-');
124 plot([startPrediction:js-1], [m(3, startPrediction+1), predictions{1}]
125     , 'g*', 'LineWidth', lineWidth);
126 plot([startPrediction+1:js-1], [m(3, startPrediction+2), predictions{2}]
127     , 'g*', 'LineWidth', lineWidth);
128 plot([startPlot:startPrediction]
129     , m(3, startPlot+1:startPrediction+1)+sqrt(C(3,3)), 'r--');
130 plot([startPlot:startPrediction]
131     , m(3, startPlot+1:startPrediction+1)-sqrt(C(3,3)), 'r--');
132 plot([startPrediction:startPrediction+windowSize]
133     , m(3, startPrediction+1:startPrediction+windowSize+1)
134     , 'ro:', 'LineWidth', lineWidth);
135 hold; grid;
136 ylabel('Broj Trenutno Zarazenih', 'FontSize', labelFont);
137 xlabel('Dani', 'FontSize', labelFont);
138 title('3DVAR uz SEIRD model - predikcija 7 dana ...
139     unaprijed', 'FontSize', titleFont);
140 legend('Opservacije', 'Signal (Predikcija + Analiza)', 'Visednevne ...
141     predikcije', 'FontSize', lgdFontSize);
142 % SVE
143 js=startPrediction+windowSize+1;
144 figure;
145 plot([startPlot:js+addOps-1], y(1, startPlot+1:js+addOps)
146     , 'bo-', 'LineWidth', lineWidth);
147 hold;
148 plot([startPlot:startPrediction], m(3, startPlot+1:startPrediction+1)
149     , 'LineWidth', lineWidth, 'ro-');
150 plot([startPrediction:js-1], [m(3, startPrediction+1), predictions{1}]
151     , 'g*', 'LineWidth', lineWidth);
152 plot([startPrediction+1:js-1], [m(3, startPrediction+2), predictions{2}]
153     , 'g*', 'LineWidth', lineWidth);
154 plot([startPrediction+2:js-1], [m(3, startPrediction+3), predictions{3}]
155     , 'g*', 'LineWidth', lineWidth);
156 plot([startPrediction+3:js-1], [m(3, startPrediction+4), predictions{4}]
157     , 'g*', 'LineWidth', lineWidth);
158 plot([startPrediction+4:js-1], [m(3, startPrediction+5), predictions{5}]
159     , 'g*', 'LineWidth', lineWidth);
160 plot([startPrediction+5:js-1], [m(3, startPrediction+6), predictions{6}]
161     , 'g*', 'LineWidth', lineWidth);
162 plot([startPrediction+6:js-1], [m(3, startPrediction+7), predictions{7}]
163     , 'g*', 'LineWidth', lineWidth);
164 plot([startPlot:startPrediction]
165     , m(3, startPlot+1:startPrediction+1)+sqrt(C(3,3)), 'r--');

```

```

165 plot([startPlot:startPrediction]
166       ,m(3,startPlot+1:startPrediction+1)-sqrt(C(3,3)), 'r--');
167 plot([startPrediction:startPrediction+windowSize]
168       ,m(3,startPrediction+1:startPrediction+windowSize+1)
169       , 'ro:', 'LineWidth', lineWidth);
170 hold;grid;
171 ylabel('Broj Trenutno Zarazenih', 'FontSize', labelFont);
172 xlabel('Dani', 'FontSize', labelFont);
173 title('3DVAR uz SEIRD model - predikcija 7 dana ...
        unaprijed', 'FontSize', titleFont);
174 legend('Opservacije', 'Signal (Predikcija + Analiza)', 'Visednevne ...
        predikcije', 'Fontsize', lgdFontSize);

```

Kod za ExKF algoritam uz SEIRD model (slika 5.12):

```

1  %%% Ucitavanje podataka %%%
2  data = csvread('..\0-Data\COVID-19_data.csv');
3  y = zeros(2, length(data(:,1)));
4  y(1, :) = data(:,11); % Trenutno zarazeni
5  y(2, :) = data(:,6); % Kumulativno preminuli
6
7  %%% Parametri %%%
8  J = length(y);
9  H = [0,0,1,0,0;
10      0,0,0,0,1];
11
12 %%% Sumovi
13 n = 5; I = eye(n);
14 gamma = 10000; sigma = 1000;
15 G = eye(2)*gamma^2; E = I*sigma^2;
16
17 %%% SEIRD parametri
18 N = 4000000;
19 beta=0.15; D=10; lambda=1/D; eps=1/5; mu=0.01;
20
21 %%% Pocetne vrijednosti
22 m = zeros(n, J+1); C = zeros(n, n, J+1);
23 C(:, :, 1) = I*gamma^2;
24 m(:, 1) = [N-1; 1; 1; 0; 0];
25
26 windowSize = 30;
27 startPrediction = 540;
28 predictions = cell(windowSize);
29
30 # Asimiliraj do startPrediction
31 for i=1:startPrediction
32     mhat = zeros(5,1);

```

```

33 mhat(1) = m(1,i) - m(3,i)*beta*(m(1,i)/N);
34 mhat(2) = m(2,i) - eps*m(2,i) + m(3,i)*beta*(m(1,i)/N);
35 mhat(3) = m(3,i) + eps*m(2,i) - lambda*m(3,i) - mu*m(3,i);
36 mhat(4) = m(4,i) + lambda*m(3,i);
37 mhat(5) = m(5,i) + mu*m(3,i);
38 Dphi = DPhi_SEIRD(m(:,i),beta,eps,lambda,mu,N);
39 Chat = Dphi*C(:, :, i)*Dphi' + E;
40
41 % Analiza
42 S = H*Chat*H' + G;
43 K = Chat*H'*inv(S);
44 m(:,i+1) = (I - K*H)*mhat + K*y(:,i);
45 C(:, :, i+1) = (I - K*H)*Chat;
46 end
47
48 # Predvidi windowSize
49 for l=1>windowSize
50     mhat = zeros(5,1);
51     mhat(1) = m(1,startPrediction+l) - ...
52         m(3,startPrediction+l)*beta*(m(1,startPrediction+l)/N);
53     mhat(2) = m(2,startPrediction+l) - eps*m(2,startPrediction+l) ...
54         + m(3,startPrediction+l)*beta*(m(1,startPrediction+l)/N);
55     mhat(3) = m(3,startPrediction+l) + eps*m(2,startPrediction+l) ...
56         - lambda*m(3,startPrediction+l) - mu*m(3,startPrediction+l);
57     mhat(4) = m(4,startPrediction+l) + lambda*m(3,startPrediction+l);
58     mhat(5) = m(5,startPrediction+l) + mu*m(3,startPrediction+l);
59     m(:,startPrediction+l+1) = mhat;
60 endfor
61 predictions{1} = m(3,startPrediction+2:startPrediction>windowSize+1);
62
63 # Asimiliraj i predvidaj do startPrediction>windowSize-1
64 for j=1:(windowSize-1)
65     # Asimiliraj
66     mhat = zeros(5,1);
67     mhat(1) = m(1,startPrediction+j) - ...
68         m(3,startPrediction+j)*beta*(m(1,startPrediction+j)/N);
69     mhat(2) = m(2,startPrediction+j) - eps*m(2,startPrediction+j) + ...
70         m(3,startPrediction+j)*beta*(m(1,startPrediction+j)/N);
71     mhat(3) = m(3,startPrediction+j) + eps*m(2,startPrediction+j) - ...
72         lambda*m(3,startPrediction+j) - mu*m(3,startPrediction+j);
73     mhat(4) = m(4,startPrediction+j) + lambda*m(3,startPrediction+j);
74     mhat(5) = m(5,startPrediction+j) + mu*m(3,startPrediction+j);
75     Dphi = DPhi_SEIRD(m(:,startPrediction+j),beta,eps,lambda,mu,N);
76     Chat = Dphi*C(:, :, startPrediction+j)*Dphi' + E;
77     % Analiza
78     S = H*Chat*H' + G;
79     K = Chat*H'*inv(S);

```



```

74 m(:,startPrediction+j+1) = (I - K*H)*mhat + ...
    K*y(:,startPrediction+j);
75 C(:, :, startPrediction+j+1) = (I - K*H)*Chat;
76
77 # Predvidi ostatak
78 for l=1:(windowSize-j)
79     mhat = zeros(5,1);
80     mhat(1) = m(1,startPrediction+l+j) - ...
        m(3,startPrediction+l+j)*beta*(m(1,startPrediction+l+j)/N);
81     mhat(2) = m(2,startPrediction+l+j) - ...
        eps*m(2,startPrediction+l+j) + ...
        m(3,startPrediction+l+j)*beta*(m(1,startPrediction+l+j)/N);
82     mhat(3) = m(3,startPrediction+l+j) + ...
        eps*m(2,startPrediction+l+j) - ...
        lambda*m(3,startPrediction+l+j) - ...
        mu*m(3,startPrediction+l+j);
83     mhat(4) = m(4,startPrediction+l+j) + ...
        lambda*m(3,startPrediction+l+j);
84     mhat(5) = m(5,startPrediction+l+j) + mu*m(3,startPrediction+l+j);
85     m(:,startPrediction+l+1+j) = mhat;
86 endfor
87 predictions{j+1} = ...
    m(3,startPrediction+2+j:startPrediction+windowSize+1);
88 end
89
90 mhat = m(:,startPrediction+windowSize+1);
91 Dphi = ...
    DPhi_SEIRD(m(:,startPrediction+windowSize),beta,eps,lambda,mu,N);
92 Chat = Dphi*C(:, :, startPrediction+windowSize)*Dphi' + E;
93 % Analiza
94 S = H*Chat*H' + G;
95 K = Chat*H'*inv(S);
96 m(:,startPrediction+windowSize+1) = (I - K*H)*mhat + ...
    K*y(:,startPrediction+windowSize);
97 C(:, :, startPrediction+windowSize+1) = (I - K*H)*Chat;
98
99 %%% Grafovi %%%
100 labelFont = 15;
101 titleFont = 20;
102 lineWidth = 2;
103 lgdLocation = 'east';
104 lgdFontSize = 15;
105 addOps = 0;
106 startPlot = startPrediction - 5;
107
108 # SVE
109 js=startPrediction+windowSize+1;

```

```

110 figure;
111 plot([startPlot:js+addOps-1],y(1,startPlot+1:js+addOps)
112     , 'bo-', 'LineWidth',lineWidth);
113 hold;
114 plot([startPlot:startPrediction],m(3,startPlot+1:startPrediction+1)
115     , 'LineWidth',lineWidth, 'ro-');
116 plot([startPrediction:js-1],[m(3,startPrediction+1),predictions{1}]
117     , 'g*:', 'LineWidth',lineWidth);
118 plot([startPrediction+14:js-1]
119     , [m(3,startPrediction+15),predictions{15}]
120     , 'g*:', 'LineWidth',lineWidth);
121 plot([startPrediction+19:js-1]
122     , [m(3,startPrediction+20),predictions{20}]
123     , 'g*:', 'LineWidth',lineWidth);
124 #plot([startPlot:startPrediction]
125     , m(3,startPlot+1:startPrediction+1)+sqrt(C(3,3)), 'r--');
126 #plot([startPlot:startPrediction]
127     , m(3,startPlot+1:startPrediction+1)-sqrt(C(3,3)), 'r--');
128 plot([startPrediction:startPrediction+windowSize]
129     , m(3,startPrediction+1:startPrediction+windowSize+1)
130     , 'ro:', 'LineWidth',lineWidth);
131 hold;grid;
132 ylabel('Broj Trenutno Zarazenih', 'FontSize',labelFont);
133 xlabel('Dani', 'FontSize',labelFont);
134 title('ExKF uz SEIRD model - predikcija 30 dana ...
135     unaprijed', 'FontSize',titleFont);
135 legend('Opservacije', 'Signal (Predikcija + Analiza)', 'Visednevne ...
136     predikcije', 'FontSize',lgdFontSize);

```