

Dinamička modalna dekompozicija u empirijskoj identifikaciji nelinearnih dinamičkih sustava

Boras, Ivana

Master's thesis / Diplomski rad

2020

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

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:217:451109>

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

Download date / Datum preuzimanja: **2024-04-24**



Repository / Repozitorij:

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



SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO–MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

Ivana Boras

DINAMIČKA MODALNA DEKOMPOZICIJA U
EMPIRIJSKOJ IDENTIFIKACIJI NELINEARNIH
DINAMIČKIH SUSTAVA

Diplomski rad

Voditelj rada:
prof. dr. sc. Zlatko Drmač

Zagreb, 2020.

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

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

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

*Ovaj rad posvećujem svojim roditeljima i svojoj sestri. Hvala vam za svu podršku i žrtvu!
Bez vas ne bih bila ovdje gdje jesam.*

Sadržaj

Sadržaj	iv
Uvod	1
1 Osnovni pojmovi	2
1.1 Dinamički sustav	2
1.2 Dekompozicije matrica i primjene	4
2 Dinamička modalna dekompozicija (DMD)	8
2.1 Uvod	8
2.2 Matematička pozadina	8
2.3 Formulacija DMD	11
3 DMD i Koopmanov operator	17
3.1 Uvod	17
3.2 Koopmanov operator	18
3.3 Veza DMD i Koopmanovog operatora	20
4 Procesiranje videa	23
4.1 RPCA u procesiranju videa	23
4.2 DMD u procesiranju videa	24
4.3 Testiranje algoritma	27
5 Varijacije DMD algoritma	32
5.1 Određivanje parametra redukcije dimenzije	32
5.2 Poboľjšani Rayleigh-Ritzovi parovi	33
5.3 Redukcija dimenzije korištenjem QR dekompozicije	39
5.4 DMDSP	42
Bibliografija	49

Uvod

Dinamička modalna dekompozicija (DMD) je algoritam redukcije dimenzionalnosti i analize strukture dinamičkih sustava koji je razvio Peter Schmid 2008. godine. DMD je algoritam treniran podacima (*data driven*) koje dobijemo ili numerički ili eksperimentalno i za DMD metodu nije potrebno poznavati jednadžbe koje opisuju dinamiku promatranog sustava, tj. kažemo da je *equation free* metoda, te kao takva pronalazi primjenu u raznim područjima, uključujući mehaniku fluida, procesiranje videa, neuroznanost, epidemiologiju, financijska tržišta itd.

Dinamička modalna dekompozicija nudi dekompoziciju visoko-dimenzionalnog kompleksnog dinamičkog sustava u prostorno i vremensko ovisne strukture koji dominiraju u danim podacima i koje se kasnije koriste npr. za predviđanje budućeg stanja samog sustava. DMD je prvo nastala kao alat za proučavanje dinamike fluida, tj. kao metoda za dekompoziciju kompleksnog toka u jednostavnije reprezentacije. Kasnije je dokazano da je s nelinearnom dinamikom povezana preko teorije Koopmanovog operatora ili tzv. operatora kompozicije.

U prvom poglavlju ovog rada ponovit ćemo neke osnovne pojmove i metode koje će se kasnije provlačiti kroz rad. U drugom poglavlju ćemo detaljno opisati DMD, njegovu matematičku pozadinu i formulaciju DMD algoritma. U trećem poglavlju ćemo opisati Koopmanove operatore, dati uvid u spektralnu teoriju Koopmanovih operatora te pokazati vezu sa DMD metodom. U četvrtom poglavlju ćemo prikazati primjenu DMD metode kod procesiranja videa, tj. razdvajanju pozadine od prvog plana i u petom poglavlju ćemo dati ideje kako poboljšati DMD algoritam i kako pomoću DMD algoritma rekonstruirati uzorke promatranog dinamičkog sustava.

Poglavlje 1

Osnovni pojmovi

Objasnit ćemo ukratko pojmove koji će se spominjati u radu. Počet ćemo s definicijom dinamičkog sustava, zatim ćemo spomenuti neke dekompozicije matrica i dati kratak opis metoda koje se baziraju na tim dekompozicijama.

1.1 Dinamički sustav

U ovom radu centralnu ulogu imaju dinamički sustavi te njihova rekonstrukcija matematičkim modelom. Za kratak opis dinamičkog sustava koristit ćemo [10]. Dinamički sustav je sustav u kojem funkcija opisuje vremensku ovisnost točke u geometrijskom prostoru. Neka je zadan skup početnih položaja $E \subset \mathbb{R}^n$, dinamički sustav nam opisuje gibanje tog skupa kroz vrijeme.

Definicija 1.1.1. *Autonomni dinamički sustav je sustav običnih diferencijalnih jednadžbi koje ne ovise eksplicitno o nezavisnoj varijabli.*

Definicija 1.1.2. *Neka je $E \subset \mathbb{R}^n$ otvoren skup. Dinamički sustav na skupu E je funkcija $\phi = \phi(t, x) : \mathbb{R} \times E \rightarrow E$ klase C^1 tako da*

1. $\phi(0, x) = x, x \in E$
2. $\phi(t + s, x) = \phi(t, \phi(s, x)), t, s \in \mathbb{R}, x \in E.$

Ako uvedemo oznaku $\phi^t(x) = \phi(t, x)$ svojstva dinamičkog sustava možemo zapisati kao

1. $\phi^0 = id$
2. $\phi^{t+s} = \phi^t \circ \phi^s.$

Primjer 1.1.3. Pogledajmo definiraju li sljedeće funkcije dinamički sustav:

1. $\varphi : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad \varphi(t, x) = 4t + x;$
2. $\eta : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad \eta(t, x) = 2t + 3x.$

Promotrimo prvo funkciju φ , naime $\varphi(0, x) = x$ i vrijedi

$$\varphi(t, \varphi(s, x)) = 4t + \varphi(s, x) = 4t + 4s + x = 4(t + s) + x = \varphi(t + s, x)$$

pa vidimo da funkcija φ definira dinamički sustav. Za funkciju η vrijedi $\eta(0, x) \neq x$ pa funkcija η nije dinamički sustav.

Teorem 1.1.4. Neka je zadan dinamički sustav $\phi : \mathbb{R} \times E \rightarrow E$. Neka je $x(t) = \phi(t, x_0)$ za neki $x_0 \in E$ i neka je

$$f(x) = \frac{\partial \phi}{\partial t}(0, x).$$

Tada je funkcija x rješenje inicijalnog problema

$$\begin{aligned} x' &= f(x) \\ x(0) &= x_0. \end{aligned} \tag{1.1.1}$$

Dokaz. Prema prvom svojstvu dinamičkog sustava imamo $x(0) = \phi(0, x_0) = x_0$ a prema drugom svojstvu imamo

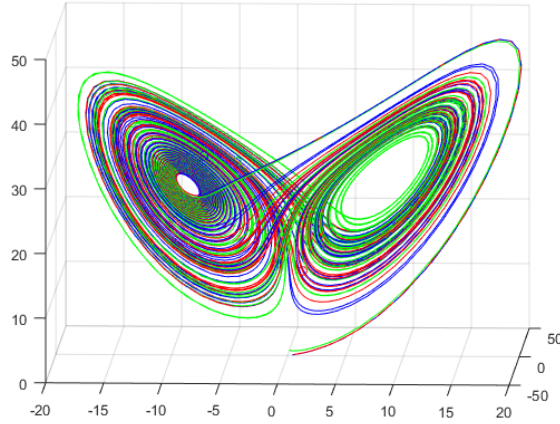
$$\frac{dx}{dt}(t) = \frac{\partial}{\partial t} \phi(t, x_0) = \frac{\partial}{\partial t} \phi(0 + t, x_0) = \frac{\partial}{\partial t} \phi(0, \varphi(t, x_0)) = \frac{\partial}{\partial t} \phi(0, x(t)) = f(x(t)) \tag{1.1.2}$$

i time je dokaz gotov. □

Primjer dinamičkog sustava je Lorenzov sustav. Lorenzov sustav je nelinearan, trodimenzionalan i glasi

$$\begin{aligned} x' &= \alpha(y - x) \\ y' &= \beta x - y - xz \\ z' &= -\gamma z + xy \end{aligned} \tag{1.1.3}$$

gdje su α, β, γ parametri sustava. Karakterizira ga leptiroliki oblik.



Slika 1.1: Ilustracija Lorenzovog dinamičkog sustava s parametrima $\alpha = 10$, $\beta = 28$ i $\gamma = \frac{8}{3}$.

1.2 Dekompozicije matrica i primjene

Prilikom matematičkog analiziranja nekog dinamičkog sustava, primjerice toka fuida, praksa je da izdvajamo bitne karakteristike sustava te pomoću njih opisujemo dinamiku sustava. Prvi korak u tome je dekompozicija matrica. Jedan od načina dekompozicije matrica je SVD i QR dekompozicija, te kasnije koristeći njih razvijamo algoritme POD i DMD.

Spektralna dekompozicija matrice

Dekompozicija pomoću svojstvenih vrijednosti i svojstvenih vektora matrice nam daje smjerove duž kojih matrica djeluje na vektor što je ekvivalentno množenju vektora skalarom. Za neku matricu $X \in \mathbb{C}^{n \times n}$, vektor $\mathbf{v} \in \mathbb{C}^n$, $\mathbf{v} \neq 0$ i skalar $\lambda \in \mathbb{C}$ zovemo svojstveni vektor i svojstvena vrijednost (redom) matrice X ako vrijedi

$$X\mathbf{v} = \lambda\mathbf{v}. \quad (1.2.1)$$

Ako je $\mathbf{v}$ svojstveni vektor tada je i $\alpha\mathbf{v}$ svojstveni vektor za $\alpha \in \mathbb{C}$, $\alpha \neq 0$. Skup svih svojstvenih vrijednosti matrice X zovemo spektar matrice X . Ako matrica X ima n linearno nezavisnih svojstvenih vektora $\mathbf{v}_j$ sa odgovarajućom svojstvenom vrijednosti λ_j tada imamo

$$XV = V\Lambda$$

gdje je $V = (\mathbf{v}_1 \dots \mathbf{v}_n) \in \mathbb{C}^{n \times n}$ a $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n) \in \mathbb{C}^{n \times n}$. Množeći zdesna s V^{-1} dobijemo

$$X = V\Lambda V^{-1}$$

i to zovemo svojstvena ili spektralna dekompozicija matrice X . Ovakva dekompozicija je moguća samo za dijagonalizabilne matrice. Mi ćemo dalje u radu raditi s matricama koje su dijagonalizabilne pa ovo vrijedi.

Dekompozicija singularnih vrijednosti

Dekompozicija singularnih vrijednosti (SVD) (eng. *Singular value decomposition*) je generalizirani oblik spektralne dekompozicije opisane prethodno te vrijedi za svaku matricu $X \in \mathbb{C}^{n \times m}$.

Definicija 1.2.1. Neka je dana matrica $X \in \mathbb{C}^{n \times m}$ za $n, m \in \mathbb{N}$. Dekompozicija singularnih vrijednosti (SVD) matrice X je dana kao $X = U\Sigma V^*$, gdje je $U \in \mathbb{C}^{n \times n}$, $\Sigma \in \mathbb{R}^{n \times m}$ i $V \in \mathbb{C}^{m \times m}$. V^* označava konjugirano transponiranu matricu matrice V . Matrice U i V su unitarne tj. vrijedi $U^*U = UU^* = I$, $V^*V = VV^* = I$, Σ je dijagonalna matrica, $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_{\min(n,m)})$ gdje su σ_i singularne vrijednosti matrice X i vrijedi $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min(n,m)} \geq 0$. Stupce matrice U nazivamo lijevi, a stupce matrice V desni singularni vektori matrice X (redom).

POD

Prava ortogonalna dekompozicija (POD) (eng. *Proper orthogonal decomposition*) je postupak redukcije dimenzije tako da se konstruira potprostor koji najbolje aproksimira dane vektore u smislu najmanjih kvadrata. POD ekstrahira podatke koji sadrže najviše "energije"¹ sustava. Konstruira se najčešće korištenjem SVD matrice. Naime, neka je $X \in \mathbb{C}^{n \times m}$ matrica ranga $r \leq \min(n, m)$. Tada, prema definiciji 1.2.1, znamo da postoje matrice $U \in \mathbb{C}^{n \times n}$ i $V \in \mathbb{C}^{m \times m}$ tako da vrijedi

$$U^*XV = \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} = \Sigma, \quad D = \text{diag}(\sigma_1, \dots, \sigma_r), \quad \sigma_1 \geq \dots \geq \sigma_r > 0.$$

Sada umjesto $X = U\Sigma V^*$ možemo pisati $X = U_r D V_r^*$ gdje su U_r i V_r dobivene tako da uzmemo prvih r stupaca matrice U i V redom. Vidimo da stupce matrice X možemo zapisati u bazi koju čine stupci matrice U_r . Sada ćemo iskazati teorem koji pokazuje da je ta baza optimalna u smislu najmanjih kvadrata.

Teorem 1.2.2. Neka je $X \in \mathbb{C}^{n \times m}$ i neka je $X = U\Sigma V^*$ dan SVD matrice X . $\Sigma = \text{diag}(\sigma_i)_{i=1}^{\min(n,m)}$, $\sigma_1 \geq \dots \geq \sigma_{\min(n,m)} \geq 0$. Za $r \in \{1, \dots, \min(n, m)\}$ definirajmo $U_r = U(:, 1 : r)$, $\Sigma_r = \Sigma(1 : r, 1 : r)$, $V_r = V(:, 1 : r)$ i $X_r = U_r \Sigma_r V_r^*$. X_r je optimalna aproksimacija matrice X ranga r u $\|\cdot\|_2$ i u $\|\cdot\|_F$:

$$\min_{\text{rang}(N) \leq r} \|X - N\|_2 = \|X - X_r\|_2 = \sigma_{r+1} \quad (1.2.2)$$

¹Ono što smatramo "energijom" sustava ovisi o promatranom sustavu

$$\min_{rang(N) \leq r} \|X - N\|_F = \|X - X_r\|_F = \sqrt{\sum_{i=r+1}^{\min(n,m)} \sigma_i^2} \quad (1.2.3)$$

gdje $\|\cdot\|_2$ predstavlja najveću singularnu vrijednost matrice, tj. $\|A\|_2 = \sigma_{\max}(A)$ za neku matricu A , a $\|\cdot\|_F$ je Frobeniusova norma koja je jednaka $\|A\|_F = \sqrt{\sum_{j=1}^n \sum_{k=1}^m |a_{jk}|^2}$.

Stupci matrice U_r razapinju najbolji r -dimenzionalni potprostor koji zahvaća podatke u smislu najmanjih kvadrata pa vidimo da POD bazu možemo dobiti računanjem SVD dekompozicije matrice.

Rayleigh-Ritzova metoda

Jedna od metoda za približno računanje svojstvene dekompozicije matrice je Rayleigh-Ritzova metoda. Ona se bazira na generiranju potprostora $\mathcal{W}$ koji sadržavaju aproksimacije traženih svojstvenih vektora. Neka je $XV = V\Lambda$ svojstvena dekompozicija matrice $X \in \mathbb{C}^{n \times n}$ gdje je V matrica svojstvenih vektora a $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ gdje su λ_i , $i = 1, \dots, n$ svojstvene vrijednosti. Neka je X matrica čiju dekompoziciju tražimo, Rayleigh-Ritzova metoda pruža aproksimaciju $(\tilde{\Lambda}, \tilde{V})$ para (Λ, V) na sljedeći način:

1. izračunamo ortonormalnu bazu $W \in \mathbb{C}^{n \times k}$ potprostora $\mathcal{W}$
2. izračunamo $S = W^* X W$, $S \in \mathbb{C}^{k \times k}$
3. izračunamo svojstvenu dekompoziciju matrice S ; $SZ = Z\tilde{\Lambda}$, gdje je $Z = (z_1, \dots, z_k)$, a matrica $\tilde{\Lambda} = \text{diag}(\tilde{\lambda}_1, \dots, \tilde{\lambda}_k)$ tako da za $\tilde{\lambda}_i$ vrijedi $Sz_i = \tilde{\lambda}_i z_i$, $i = 1, \dots, k$
4. $\tilde{V} = WZ$

Matricu S zovemo Rayleighev kvocijent. Vrijednost $\tilde{\lambda}_i$ se zove Ritzova vrijednost a vektor $\tilde{v}_i = Wz_i$ zovemo Ritzov vektor i vrijedi $X\tilde{v}_i \approx \tilde{\lambda}_i \tilde{v}_i$.

Pseudo-inverz matrice

Pomoću SVD dekompozicije možemo računati pseudo-inverz matrice. Pogledajmo prvo što je pseudo-inverz.

Definicija 1.2.3. *Pseudo-inverz je poopćeni oblik matričnog inverza za matrice koje nisu kvadratne. Zove se još Moore-Penroseov inverz. Neka je $X \in \mathbb{C}^{m \times n}$ matrica, njezin pseudo-*

inverz je jedinstvena matrica $X^\dagger \in \mathbb{C}^{n \times m}$ tako da vrijedi:

$$\begin{aligned} XX^\dagger X &= X \\ X^\dagger XX^\dagger &= X^\dagger \\ (XX^\dagger)^* &= XX^\dagger \\ (X^\dagger X)^* &= X^\dagger X. \end{aligned} \tag{1.2.4}$$

Neka je $X = U\Sigma V^*$ SVD dekompozicija matrice X ranga r . Generalizirani inverz matrice oblika

$$\Sigma = \begin{pmatrix} \Sigma_1 \\ 0 \end{pmatrix} \in \mathbb{R}^{m \times n}, m \geq n \text{ gdje je } \Sigma_1 = \begin{pmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \\ 0 & & \sigma_n \end{pmatrix}, \sigma_1 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_n = 0,$$

je matrica

$$\Sigma^\dagger = \begin{pmatrix} \tilde{\Sigma}^\dagger & 0 \end{pmatrix} \in \mathbb{R}^{n \times m}, \quad \tilde{\Sigma}^\dagger = \begin{pmatrix} \sigma_1^\dagger & \dots & 0 \\ \vdots & \ddots & \\ 0 & & \sigma_n^\dagger \end{pmatrix}$$

gdje je $\sigma_i^\dagger = 1/\sigma_i$ ako je $\sigma_i > 0$ i $\sigma_i^\dagger = 0$ ako je $\sigma_i = 0$. Sada je pseudo-inverz matrice X dan s $X^\dagger = V\Sigma^\dagger U^* \in \mathbb{C}^{n \times m}$. Ako je X kvadratna regularna matrica tada je $X^\dagger = X^{-1}$.

Pseudo-inverz se najčešće koristi za računanje rješenja problema najmanjih kvadarata $\min_x \|Ax - b\|$ gdje je $x = A^\dagger b$ rješenje problema koje je i samo najmanje $\|\cdot\|_2$ norme.

QR dekompozicija

Definicija 1.2.4. Neka je dana matrica $X \in \mathbb{C}^{n \times m}$, $n \geq m$. QR dekompozicija matrice X je rastav matrice X na produkt $X = QR$ pri čemu je $Q \in \mathbb{C}^{n \times n}$ unitarna matrica, tj. vrijedi $Q^*Q = QQ^* = I_n$, a $R \in \mathbb{C}^{n \times m}$ gornje trokutasta. Štoviše, vrijedi:

$$X = QR = Q \begin{pmatrix} R_1 \\ 0 \end{pmatrix} = (Q_1, Q_2) \begin{pmatrix} R_1 \\ 0 \end{pmatrix} = Q_1 R_1$$

gdje se zadnjih $(n - m)$ redova matrice R sastoje samo od 0, stoga je $R_1 \in \mathbb{C}^{m \times m}$ gornje trokutasta matrica, $Q_1 \in \mathbb{C}^{n \times m}$ i $Q_2 \in \mathbb{C}^{n \times (n-m)}$ i stupci matrica Q_1 i Q_2 su ortogonalni i međusobno okomiti.

Poglavlje 2

Dinamička modalna dekompozicija (DMD)

2.1 Uvod

Dinamička modalna dekompozicija je dekompozicija koja izdvaja iz visoko-dimenzionalnog sustava osnovne podatke nižeg ranga, što omogućava bolje i preciznije interpretiranje u prostorno-vremenskoj ovisnosti. DMD nam vraća modove (prostorne) kao i njima pridodane vrijednosti koji opisuju frekvenciju oscilacije i brzinu širenja koja je zapisana u pripadnoj svojstvenoj vrijednosti. Bazira se na prostornoj redukciji (POD) linearnog operatora A koji najbolje aproksimira dinamiku sustava. Zbog velike pretpostavljene dimenzije n dekompozicija matrice A može nekad biti teška za računanje stoga prvo projiciramo podatke na potprostor nižeg ranga razapet POD modovima i kasnije pomoću njih računamo aproksimaciju nižeg ranga.

2.2 Matematička pozadina

Fokus DMD metode su dinamički sustavi. Kao što smo rekli u prvom poglavlju, dinamički sustav nam opisuje evoluciju fizikalnih varijabli preko diferencijalnih jednadžbi koje su većinom nelinearne. Neka nam je zadan neki dinamički sustav

$$\frac{dx}{dt} = f(x, t; \mu) \quad (2.2.1)$$

gdje je $x(t) \in \mathbb{R}^n$, $n \gg 1$ je vektor koji predstavlja stanje promatranog dinamičkog sustava u vremenu t , tj. on nam predstavlja uzorak dinamičkog sustava u vremenu t (*eng. snapshot*) a μ sadrži parametre sustava. Sustav (2.2.1) je neprekidan i kao takvi se oni prirodno

pojavljaju ali radi praktičnijeg računanja u praksi se promatra vremenski diskretiziran sustav jer se i sami podaci o sustavu prikupljaju na taj način. Vremenskom diskretizacijom sustava (2.2.1) dobijemo nove vrijednosti vektora $x_k = x(k\Delta t)$, $k \in \mathbb{N}$ gdje je Δt korak diskretizacije sustava. Neka je F preslikavanje dobiveno evolucijom sustava za Δt

$$x_{k+1} = F(x_k). \quad (2.2.2)$$

Napomena 2.2.1. *Preslikavanje F može biti rezultat mjerenja, modeliranja ili primjerice numeričke simulacije gdje F predstavlja neki softverski paket.*

Mjere promatranog sustava su opisane nekom funkcijom $y_k = g(x_k)$ u vremenu t_k no većinom mjera sustava u vremenu t_k je upravo stanje sustava u tom vremenu ukoliko nemamo neke parametre sustava, tj. $y_k = x_k$ i pomoću njih aproksimiramo dinamiku sustava jer često ne znamo desnu stranu sustava (2.2.1). Neka nam je sada dana lokalna linearna aproksimacija sustava (2.2.1):

$$\frac{dx}{dt} = \mathcal{A}x \quad \mathcal{A} \in \mathbb{C}^{n \times n} \quad (2.2.3)$$

s početnim uvjetom $x(0) = x_0$. Neka je sada $\mathcal{A}Z = Z\Omega$ svojstvena dekompozicija matrice $\mathcal{A}$ (pogledati poglavlje 1). Rješenje (2.2.3) je $x(t) = e^{\mathcal{A}t}x_0 = Ze^{\Omega t}Z^{-1}x_0$ tj. imamo

$$x(t) = \sum_{k=1}^n z_k e^{\omega_k t} b_k \quad (2.2.4)$$

gdje su koeficijenti b_k koordinate početnog uvjeta u bazi svojstvenih vektora. Sada vremenskom diskretizacijom sustava (2.2.3) za vremenski korak Δt imamo:

$x_{k+1} = x((k+1)\Delta t) = e^{\mathcal{A}((k+1)\Delta t)}x_0 = e^{\mathcal{A}(k\Delta t)}e^{\mathcal{A}\Delta t}x_0 = e^{\mathcal{A}\Delta t}x(k\Delta t) = e^{\mathcal{A}\Delta t}x_k$ pa dobijemo sustav:

$$x_{k+1} = Ax_k \quad (2.2.5)$$

gdje je $A = e^{\mathcal{A}\Delta t}$. Rješenje ovog sustava je oblika :

$$x_k = \sum_{j=1}^n z_j \lambda_j^k b_j = Z\Lambda^k b \quad (2.2.6)$$

gdje su λ_j svojstvene vrijednosti a z_j svojstveni vektori matrice A a b_j kao i prije, koeficijenti početnog uvjeta u bazi svojstvenih vektora.

DMD nam daje svojstvenu dekompoziciju matrice A koja dobro aproksimira odgovarajuće podatke x_k , $k = 1, \dots, m$ u smislu najmanjih kvadrata, tj. vrijednost

$$\|x_{k+1} - Ax_k\|_2 \quad (2.2.7)$$

je minimalna za svaki k . Ako sada vektore stanja posložimo po stupcima u matrice:

$$X = \begin{pmatrix} x_1 & x_2 & \dots & x_m \end{pmatrix}$$

$$Y = \begin{pmatrix} x_2 & x_3 & \dots & x_{m+1} \end{pmatrix}$$

te matrice će biti *visoke i mršave* ($n \gg m$). U terminima ovih matrica sada možemo zapisati (2.2.5) kao:

$$Y \approx AX \quad (2.2.8)$$

Čije rješenje je dano s

$$A = YX^\dagger \quad (2.2.9)$$

gdje $\dagger$ označava Moore-Penroseov pseudo-inverz (definicija 1.2.3). Ovako dano rješenje minimizira grešku

$$\|Y - AX\|_F \quad (2.2.10)$$

gdje je $\|\cdot\|_F$ Frobeniusova norma $\|X\|_F = \sqrt{\sum_{j=1}^n \sum_{k=1}^m X_{jk}^2}$. Po (2.2.8) vidimo da vrijedi $X^T A^T = Y^T$, pa sada po definiciji 1.2.3 i poopćenju primjene pseudo-inverza kao alata za računanje rješenja problema najmanjih kvadrata, vidimo da zaista vrijedi $A^T = (X^T)^\dagger Y^T$, tj. za matricu A vrijedi $A = X^\dagger Y$.

Definicija 2.2.2. Dinamička modalna dekompozicija: Pretpostavimo da imamo zadan dinamički sustav

$$\frac{dx}{dt} = f(x, t; \mu)$$

i dva skupa podataka

$$X = \begin{pmatrix} x_1 & x_2 & \dots & x_m \end{pmatrix} \quad (2.2.11)$$

$$Y = \begin{pmatrix} y_1 & y_2 & \dots & y_m \end{pmatrix} \quad (2.2.12)$$

gdje su $x_k \in \mathbb{R}^{n \times 1}$ vektori stanja a $y_k = F(x_k)$ za $k = 1, \dots, m$ gdje je F preslikavanje tako da vrijedi $x_{k+1} = F(x_k)$ koje odgovara evoluciji sustava (2.2.1) za vrijeme Δt . Dinamička modalna dekompozicija računa vodeću svojstvenu dekompoziciju linearnog operatora A koji najbolje opisuje podatke $Y \approx AX$:

$$A = YX^\dagger. \quad (2.2.13)$$

DMD modovi (dinamički modovi), su svojstveni vektori matrice A i svaki DMD mod odgovara jednoj svojstvenoj vrijednosti matrice A

2.3 Formulacija DMD

DMD nam daje spektralne informacije o matrici A lineariziranog diskretnog dnamičkog sustava (2.2.5). Zbog velike dimenzije n problema nailazimo na poteškoće u direktnoj analizi matrice A stoga koristimo reducirani SVD matrice aproksimirajući matricu podataka X matricom nižeg ranga u terminima POD modova (poglavlje 1.2).

Neka su matrice X, Y kao u (2.2.11) i (2.2.12).

1. U prvom koraku DMD algoritma tražimo SVD matrice $X \in \mathbb{C}^{n \times m}$:

$$X = U\Sigma V^*. \quad (2.3.1)$$

Kako radimo reduciranu SVD dekompoziciju ranga r matrice, reducirane matrice iz singularne dekompozicije su $U_r \in \mathbb{C}^{n \times r}$, $\Sigma_r \in \mathbb{C}^{r \times r}$ i $V_r \in \mathbb{C}^{m \times r}$. Stupci matrice U_r su POD modovi i čine ortonormalnu bazu potprostora koji sadrži tražene aproksimacije svojstvenih vektora matrice A .

2. Sada kada imamo ortonormalnu bazu želimo $r \times r$ aproksimaciju matrice $A = YV\Sigma^{-1}U^*$ koju dobijemo kao

$$S_r = U_r^* A U_r = U_r^* Y V_r \Sigma_r^{-1}. \quad (2.3.2)$$

Zaista, kako vrijedi $AX = Y$ imamo $AU\Sigma V^* = Y$, tj. $AU = YV\Sigma^{-1}$ pa (2.3.2) zaista vrijedi. Matrica S_r predstavlja Raylieghev kvocijent (poglavlje 1.2)

3. Računamo svojstvenu dekompoziciju matrice S_r

$$S_r W_r = W_r \Lambda_r \quad (2.3.3)$$

gdje su stupci matrice W_r svojstveni vektori matrice S_r a dijagonalni elementi matrice Λ_r pripadne svojstvene vrijednosti.

4. U zadnjem koraku sada rekonstruiramo svojstvenu dekompoziciju matrice A . Svojstvene vrijednosti matrice A su dani u matrici Λ_r , a svojstvene vektore, tj. DMD modove dobijemo kao stupce matrice $Z_r = U_r W_r$, ti modovi predstavljaju Ritzove vektore (poglavlje 1.2).

Sažmimo sada prethodne korake u algoritam:

Ulaz: $X = (x_1 \dots x_m)$, $Y = (x_2 \dots x_{m+1})$; matrice podataka

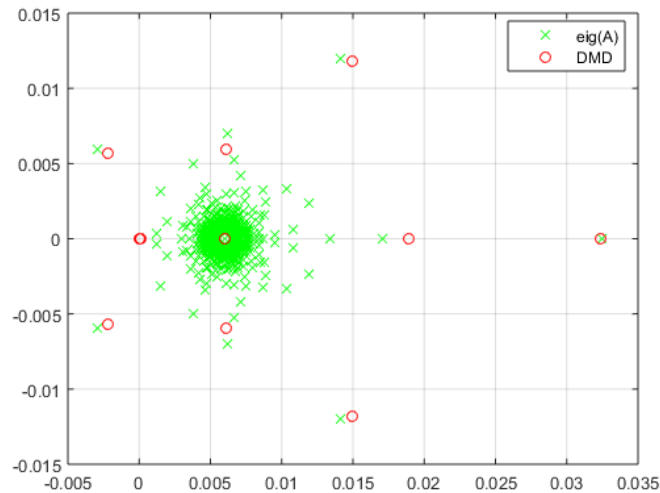
Izlaz: Z_r, Λ_r

- 1: $[U, \Sigma, V] = \text{svd}(X)$
- 2: odredi parametar redukcije r
- 3: $U_r = U(:, 1 : r)$, $\Sigma_r = \Sigma(1 : r, 1 : r)$, $V_r = V(:, 1 : r)$
- 4: $S_r = U_r^* Y V_r \Sigma_r^{-1}$
- 5: $[W_r, \Lambda_r] = \text{eig}(S_r)$
- 6: $Z_r = U_r W_r$

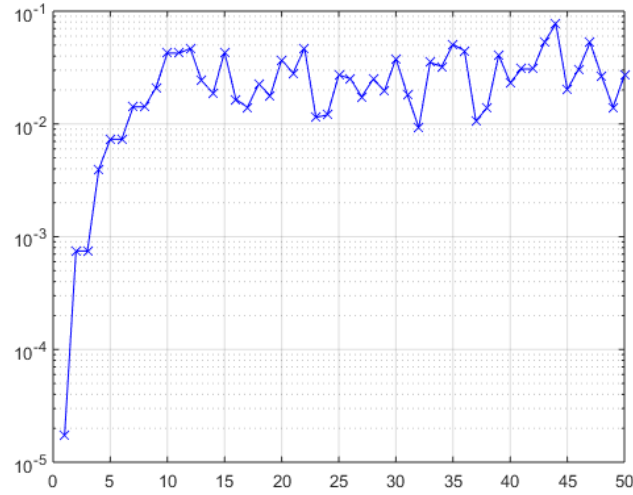
Algoritam 1: DMD

Napomena 2.3.1. Zasad ćemo se zadržati na ovoj formulaciji DMD algoritma, kasnije u poglavlju 5 ćemo dati ideje kako poboljšati algoritam.

Primjer 2.3.2. Nasumično smo generirali podatke, prvo smo generirali vektor x_1 dimenzije 1000×1 pomoću Matlabove funkcije `rand`, zatim matricu A , 1000×1000 kao $A = \exp(-\text{inv}(B))$, gdje je $B = \text{rand}(1000)$ i skalirali $A = A/\|A\|$. Zatim smo gradili matricu podataka po stupcima kao $x_{i+1} = Ax_i$, $i = 1, \dots, 199$. Kao parametar redukcije koristili smo $r = 50$.



Slika 2.1: Na slici su prikazane svojstvene vrijednosti dobivene DMD algoritmom kao i usporedba sa svojstvenim vrijednostima matrice A koju smo dobili pomoću Matlabove funkcije `eig()`.



Slika 2.2: Na slici su prikazani reziduali koje smo dobili egzaktno, koristeći matricu A ; $\|Az_i - \lambda_i z_i\|_2$.

Vratimo se sada na formulaciju DMD algoritma. Nakon što izračunamo niže-dimenzionalnu aproksimaciju ranga r svojstvenih vrijednosti i svojstvenih vektora rješenje (2.2.6) možemo konstruirati za sva buduća vremena. Kako je $A = e^{\mathcal{A}\Delta t}$, svojstvene vrijednosti operatora $\mathcal{A}$ i matrice A su povezane relacijom $\omega_k = \ln(\lambda_k)/\Delta t$ i imamo aproksimaciju rješenja:

$$x(t) \approx \sum_{k=1}^r z_k e^{\omega_k t} b_k \quad (2.3.4)$$

gdje su matrice $Z_r = (z_1, \dots, z_r)$ i $\Lambda_r = (\lambda_1, \dots, \lambda_r)$ matrice opisane u formulaciji DMD metode. Matricu svojstvenih vektora Z_r i svojstvenih vrijednosti $\Omega = \text{diag}(\omega_1, \dots, \omega_r)$ znamo kako konstruirati, pa za računanje rješenja sad još preostaje odrediti koeficijente b_k početnog uvjeta. U početnom vremenu $t = 0$ imamo zadan početni uvjet x_1 pa nam (2.3.4) daje $x_1 = Z_r b$ tj. vektor koeficijenata b možemo izračunati kao

$$b = Z_r^\dagger x_1 \quad (2.3.5)$$

gdje je $Z_r^\dagger$ Moore-Penroseov pseudoinverz matrice Z_r . Sada možemo upotrijebiti prethodni algoritam:

Ulaz: $X = (x_1 \dots x_m)$, $Y = (x_2 \dots x_{m+1})$; matrice podataka
dt; korak vremenske diskretizacije

Izlaz: Z_r , Λ_r , X_{DMD}

1. $[U, \Sigma, V] = \text{svd}(X)$;
2. odredi parametar redukcije r ;
3. $U_r = U(:, 1 : r)$, $\Sigma_r = \Sigma(1 : r, 1 : r)$, $V_r = V(:, 1 : r)$;
4. $S_r = U_r^* Y V_r \Sigma_r^{-1}$;
5. $[W_r, \Lambda_r] = \text{eig}(S_r)$, ($\Lambda_r = \text{diag}(\lambda_1, \dots, \lambda_r)$);
6. $Z_r = U_r W_r$;
7. $\omega = \log(\lambda)/dt$;
8. $b = Z_r^\dagger x_1$;
9. **Za** $i=1:m$ **radi**
| $T(:, i) = b e^{\omega t_i}$; (t je vektor vremena; $t = (0 : m - 1)dt$)
- kraj**
10. $X_{DMD} = Z_r T$;

Algoritam 2: DMD

U zadnjem koraku algoritma, matrica X_{DMD} predstavlja rekonstruiranu matricu podataka X dobivenu DMD algoritmom.

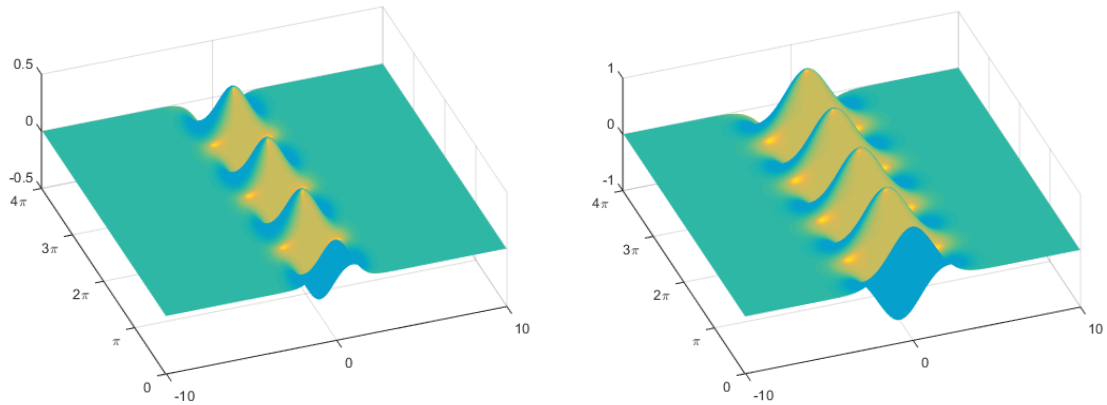
Pogledajmo sad sljedeći primjer:

Primjer 2.3.3. Neka su nam dane funkcije $f_1(x, t) = e^{-0.7x^2-1} e^{i1.8t}$ i $f_2(x, t) = e^{-0.3x^2} e^{i2.3t}$ koji predstavljaju signale i funkcija $f(x, t)$ tako da je

$$f(x, t) = f_1(x, t) + f_2(x, t) = e^{-0.7x^2-1} e^{i1.8t} + e^{-0.3x^2} e^{i2.3t} \quad (2.3.6)$$

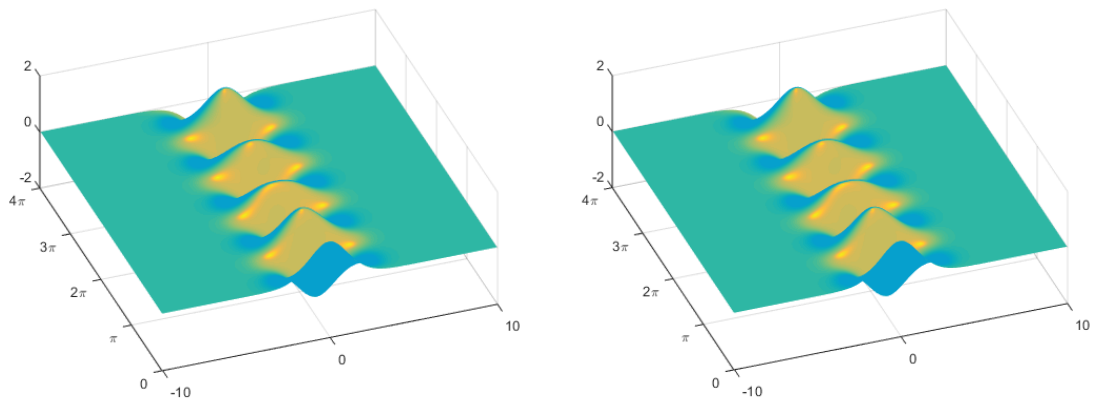
gdje je $i = \sqrt{-1}$.

Funkcije f_1 , f_2 i f kao i aproksimacija funkcije f DMD algoritmom prikazane su na sljedećim slikama:

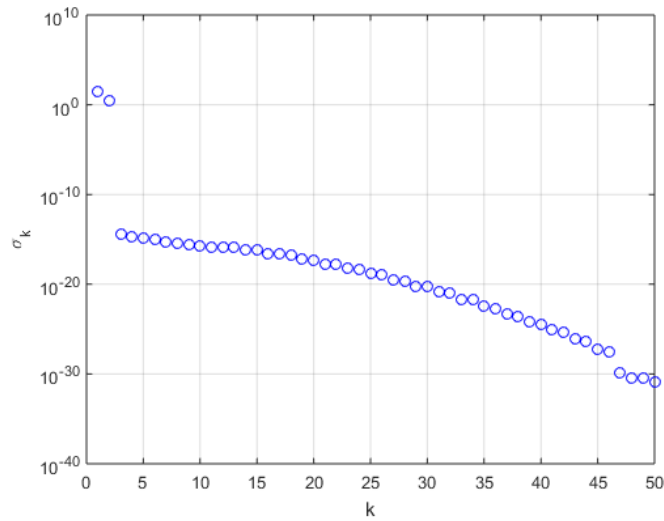


Slika 2.3: Na slici lijevo prikazan je realni dio signala $f_1(x, t)$ a na slici desno realni dio signala $f_2(x, t)$.

Kako je funkcija zadana preko dva različita signala kao parametar redukcije dimenzije uzet ćemo $r = 2$. Ako promotrimo singularne vrijednosti matrice podataka X , koju smo generirali funkcijom $f(x, t)$ na slici 2.5, vidimo da sustav lako reduciramo na sustav ranga 2.



Slika 2.4: Na slici lijevo prikazan je realni dio funkcije $f(x, t)$ a na slici desno realni dio funkcije koju smo dobili rekonstruiranjem funkcije $f(x, t)$ pomoću DMD algoritma.



Slika 2.5: Na slici su prikazane singularne vrijednosti matrice podataka X koju smo generirali funkcijom $f(x, t)$ i vidimo da je opravdano za parametar redukcije dimenzije uzeti $r = 2$.

Algoritam je također vratio svojstvene vrijednosti $\omega_1 = 0.0000 + 2.3000i$ i $\omega_2 = 0.0000 + 1.8000i$ i vidimo da se podudaraju sa frekvencijama oscilacija signala f_1 i f_2 .

Poglavlje 3

DMD i Koopmanov operator

3.1 Uvod

Nelinearne dinamičke sustave želimo linearizirati jer onda možemo koristiti već poznate alate za rješavanje linearnih sustava. Jedan od načina pomoću kojeg možemo to postići je pomoću Koopmanovih operatora. Koopmanov operator je beskonačnodimenzionalan linearan operator koji opisuje kako mjere nekog dinamičkog sustava evoluiraju kroz vrijeme, te mjere su funkcije stanja, tzv. *opservacije*. Primjerice, ako promatramo tok inkompresibilnog fluida, gibanje tog toka definira neki dinamički sustav, primjer opservacija su tlak ili brzina toka ali mogu biti i neke druge matematičke konstrukcije. Koopmanovi operatori se prvi put spominju u ranim 1930-im kao alat za demonstriranje kako Hamiltonove sustave opisati konačno dimenzionalnim linearnim operatorom u prostoru funkcija stanja, kasnije se proširio i u dinamičkim sustavima trenirani podacima. Svodeći tako nelinearni sustav na linearni, iako beskonačnodimenzionalni, Koopmanov operator omogućava primjenu spektralne teorije i drugih alata iz numeričke linearne algebre. Cilj Koopmanove analize je identificirati svojstvene funkcije iz danih podataka koje opisuju koordinatni sustav u kojem je dinamika promatranog sustava linearna. Da bismo lakše računali s beskonačnodimenzionalnim prostorima aproksimiramo ih konačnom sumom modova. Te modove će nam upravo dati DMD. Sljedeća razrada teorije Koopmanovih operatora i spektralne dekompozicije istog preuzeta je iz [8] i [7].

3.2 Koopmanov operator

Neka je dan autonomni dinamički sustav

$$\frac{dx}{dt} = f(x(t)) = \begin{pmatrix} f_1(x(t)) \\ \vdots \\ f_n(x(t)) \end{pmatrix}, \quad (3.2.1)$$

gdje je $x \in \mathcal{M}$ stanje u nekoj n -dmenzionalnoj mnogostrukosti $\mathcal{M}$ a $f : \mathcal{M} \rightarrow \mathcal{M}$. Neka je sada $\phi^t : \mathcal{M} \rightarrow \mathcal{M}$ funkcija kao u 1.1.2 koja opisuje taj dinamički sustav, koja stanje $x(t_0)$ preslika u buduće stanje $x(t_0 + t)$, tzv. funkcija toka. Vrijedi:

$$x(t_0 + t) = \phi^t(x(t_0)) = x(t_0) + \int_{t_0}^{t_0+t} f(x(\tau))d\tau.$$

Neka su sada dane funkcije stanja $g : \mathcal{M} \rightarrow \mathbb{C}$ gdje je prostor $\mathcal{F} \ni g$ dan kao Banachov ili Hilbertov prostor, npr. $L^p(\mathcal{M}, \mu)$, $1 \leq p \leq \infty$, gdje je μ neka mjera. U ovom radu nećemo davati više detalja o konstrukciji i svojstvima samog prostora $\mathcal{F}$.

Koopmanov operator $\mathcal{K}$ je beskonačnodimenzionalan linearni operator koji djeluje na sve funkcije stanja dinamičkog sustava g tako da

$$\mathcal{K}_{\phi^t} g = g \circ \phi^t \quad g \in \mathcal{F}. \quad (3.2.2)$$

Za Koopmanov operator kažemo još da je operator kompozicije. Linearnost Koopmanovog operatora slijedi iz linearnosti kompozicije :

$$\mathcal{K}_{\phi^t}(g_1 + g_2) = (g_1 + g_2) \circ \phi^t = g_1 \circ \phi^t + g_2 \circ \phi^t = \mathcal{K}_{\phi^t} g_1 + \mathcal{K}_{\phi^t} g_2.$$

Neka je sada dan vremensko-diskretiziran dinamički sustav sustava (3.2.1):

$$x_{k+1} = F(x_k) \quad (3.2.3)$$

gdje je $F : \mathcal{M} \rightarrow \mathcal{M}$, $x_k = x(kt)$, $k \in \mathbb{N}$. Koopmanov operator $\mathcal{K} \equiv \mathcal{K}_F$ za diskretiziran dinamički sustav se definira analogno:

$$\mathcal{K}g = g \circ F \quad (3.2.4)$$

$$\mathcal{K}g(x_k) = g(F(x_k)) = g(x_{k+1}). \quad (3.2.5)$$

Pretpostavimo sada da nam je prostor funkcija stanja $\mathcal{F}$ dan kao Hilbertov prostor $L^2(\mathcal{M}, \mu)$, gdje je $\mathcal{M}$ kompaktan i dovoljno velik da sadrži sva stanja i promotrimo spektralnu dekompoziciju Koopmanovog operatora $\mathcal{K}$.

Neka je $\varphi_j : \mathcal{M} \rightarrow \mathbb{C}$ funkcija i $\lambda_j \in \mathbb{C}$. Za φ_j kažemo da je svojstvena funkcija Koopmanovog operatora sa pripadnom svojstvenom vrijednošću λ_j ako vrijedi:

$$\mathcal{K}\varphi_j = \lambda_j\varphi_j. \quad (3.2.6)$$

Pretpostavimo da je $g \in \text{span}\{\varphi_j\}$, tj. možemo $g(x)$ prikazati kao:

$$g(x) = \sum_{k=1}^{\infty} \varphi_k(x)v_k \quad (3.2.7)$$

gdje je v_k , k -ti Koopmanov mod povezan sa svojstvenom funkcijom φ_k . To neće uvijek biti moguće ukoliko Koopmanov operator ima kontinuirani spektar no, te detalje ćemo preskočiti. Sada evoluciju funkcija stanja možemo opisati kao¹:

$$\mathcal{K}g(x) \approx \mathcal{K} \sum_{k=1}^{\infty} \varphi_k(x)v_k = \sum_{k=1}^{\infty} \mathcal{K}\varphi_k(x)v_k = \sum_{k=1}^{\infty} \lambda_k\varphi_k v_k. \quad (3.2.8)$$

Sve ovo možemo proširiti i na vektorske funkcije stanja $g = (g_1, \dots, g_n) : \mathcal{M} \rightarrow \mathbb{C}^n$, $g_j : \mathcal{M} \rightarrow \mathbb{C}$, $j = 1, \dots, n$. Definiramo

$$\mathcal{K}_n g = \begin{pmatrix} g_1 \circ F \\ g_2 \circ F \\ \vdots \\ g_n \circ F \end{pmatrix} = \begin{pmatrix} \mathcal{K}g_1 \\ \mathcal{K}g_2 \\ \vdots \\ \mathcal{K}g_n \end{pmatrix}. \quad (3.2.9)$$

Neka je $h \in \mathcal{F}$, $h : \mathcal{M} \rightarrow \mathbb{C}^d$,

$$h(x) = \begin{pmatrix} h_1(x) \\ \vdots \\ h_d(x) \end{pmatrix} = \sum_{j=1}^{\infty} \varphi_j(x)v_j$$

gdje je

$$h_i(x) = \sum_{j=1}^{\infty} \varphi_j(x)(v_j)_i \quad v_j = \begin{pmatrix} (v_j)_1 \\ \vdots \\ (v_j)_d \end{pmatrix}.$$

Sada evoluciju funkcije stanja h u buduća stanja dobijemo kao:

$$h(F(\mathbf{x})) = \begin{pmatrix} h_1(F(\mathbf{x})) \\ \vdots \\ h_d(F(\mathbf{x})) \end{pmatrix} = \begin{pmatrix} (\mathcal{K}h_1)(\mathbf{x}) \\ \vdots \\ (\mathcal{K}h_d)(\mathbf{x}) \end{pmatrix} = (\mathcal{K}_d g)(\mathbf{x}) = \sum_{j=1}^{\infty} \lambda_j \varphi_j(\mathbf{x})v_j. \quad (3.2.10)$$

¹Izostavit ćemo detalje konvergencije.

Vidimo da buduće stanje sustava možemo dobiti samo množeći funkciju stanja s Koopmanovom svojstvenom vrijednošću. Jednakost (3.2.10) zovemo **Koopmanova modalna dekompozicija (KMD)** funkcije stanja h a vektor v_k je Koopmanov mod funkcije stanja h sa svojstvenom vrijednošću λ_k . Primijetimo da za diskretiziran sustav i x_0 početni uvjet, iterativno dođemo:

$$h(x_k) = \mathcal{K}(h(x_{k-1})) = \mathcal{K}^2((h(x_{k-2}))) = \dots = \mathcal{K}^k((h(x_0))) \quad (3.2.11)$$

$$h(x_k) = \mathcal{K}^k \sum_{j=1}^{\infty} \varphi_j(x_0) \mathbf{v}_j = \sum_{j=1}^{\infty} \mathcal{K}^k \varphi_j(x_0) \mathbf{v}_j = \sum_{j=1}^{\infty} \lambda_j^k \varphi_j(x_0) \mathbf{v}_j. \quad (3.2.12)$$

3.3 Veza DMD i Koopmanovog operatora

Za računanje Koopmanovih modova i općenito Koopmanovih svojstvenih vrijednosti i svojstvenih funkcija izračunate podatke (uzorke stanja) slažemo u matricu podataka $M \in \mathbb{C}^{n \times m}$ gdje svaki red matrice M predstavlja vrijednosti funkcije stanja kroz vrijeme u kojem promatramo sustav:

$$M = \begin{pmatrix} g_1(x_1) & g_1(x_2) & \dots & g_1(x_m) \\ g_2(x_1) & g_2(x_2) & \dots & g_2(x_m) \\ \vdots & \vdots & & \vdots \\ g_n(x_1) & g_n(x_2) & \dots & g_n(x_m) \end{pmatrix}. \quad (3.3.1)$$

Iako radimo na nelinearnom sustavu, želimo naći matricu $A \in \mathbb{C}^{n \times n}$ tako da $Ag(x_k) = g(x_{k+1})$. Ako uzmemo $X = M(:, 1 : m-1)$ i $Y = M(:, 2 : m)$ takva matrica zadovoljava

$$AX = Y.$$

Općenito, matrice X i Y su takve da i -ti stupac u matrici Y dobijemo djelovanjem Koopmanovog operatora $\mathcal{K}$ na i -ti stupac u X :

$$X = \begin{pmatrix} g(x_1) & g(x_2) & \dots & g(x_m) \end{pmatrix} \quad (3.3.2)$$

$$Y = \begin{pmatrix} \mathcal{K}g(x_1) & \mathcal{K}g(x_2) & \dots & \mathcal{K}g(x_m) \end{pmatrix}; Y = \begin{pmatrix} g(F(x_1)) & g(F(x_2)) & \dots & g(F(x_m)) \end{pmatrix}. \quad (3.3.3)$$

Matrica A je takva da minimizira $\|Y - AX\|_F$ i možemo ju dobiti kao $A = YX^\dagger$ (pogledati definiciju 1.2.3). Ako je X punog retčanog ranga tada je optimalna matrica A jedinstvena a ako je X punog stupčanog ranga tada $A = YX^\dagger$ zadovoljava $AX = Y$ egzaktno. Mi ćemo dalje pretpostaviti da X je punog ranga (stupčanog ili retčanog).

Kao što smo rekli na početku poglavlja, želimo Koopmanov operator aproksimirati kompresijom na konačno dimenzionalan prostor.

Napomena 3.3.1. *Prisjetimo se kako možemo linearni operator prikazati matricno u nekoj bazi. Neka je $A : X \rightarrow Y$ linearni operator i neka su X i Y konačno dimenzionalni vektorski prostori nad istim poljem. Odaberimo bazu $\{f_1, \dots, f_n\}$ u X i bazu $\{g_1, \dots, g_m\}$ u Y . Tada rastav $Af_j = \sum_{i=1}^m a_{ij}g_i$, $j = 1, \dots, n$ definira matricu $A = (a_{ij})$ dimenzija $m \times n$. Ako je $A : X \rightarrow X$ onda uzimamo istu bazu.*

Transponirajmo sada matricu M^T (3.3.1) i izdvojimo matrice X^T i Y^T tako da $X^T = M^T(1 : m-1, :)$ i $Y^T = M^T(2 : m, :)$ i promatrajmo djelovanje operatora $\mathcal{K}$ na prostor $\mathcal{F}_n \subset \mathcal{F}$ razapet skalarnim funkcijama $g_1, \dots, g_n$. Sada kada imamo konačno dimenzionalan prostor možemo linearni operator $\mathcal{K}$ reprezentirati matricno. Matricnu reprezentaciju tražimo od kompresije $\Psi_{\mathcal{F}_n} \mathcal{K}|_{\mathcal{F}_n} : \mathcal{F}_n \rightarrow \mathcal{F}_n$, gdje je $\Psi_{\mathcal{F}_n}$ projekcija sa slikom $\mathcal{F}_n$. Baza u kojoj tražimo reprezentaciju je $\mathcal{N} = \{g_1, \dots, g_n\}$ pa stoga po napomeni (3.3.1) imamo:

$$(\mathcal{K}g_i)(x) = g_i(F(x)) = \sum_{j=1}^n k_{ji}g_j(x) + \rho_i(x) \quad i = 1, \dots, n \quad x \in \mathcal{M}. \quad (3.3.4)$$

Matricu $K = (k_{ji})_{j,i}^n \in \mathbb{C}^{n \times n}$ sada možemo konstruirati stupac po stupac minimizirajući rezidual ρ_i za sva stanja x :

$$\frac{1}{m+1} \sum_{k=0}^m |\rho_i(x)|^2 = \frac{1}{m+1} \sum_{k=0}^m \left| \sum_{j=1}^n k_{ji}g_j(x) - g_i(F(x)) \right|^2. \quad (3.3.5)$$

Jednakost (3.3.5) označava L^2 rezidual s empirijskom mjerom definiranom kao Diracova mjera (normirana) koncentrirana u stanju x_k , $\delta_m = \frac{1}{m+1} \sum_{k=0}^m \delta_{x_k}$. Tako konstruirana matrica K minimizira $\|X^T K - Y^T\|_F$, tj. $K = (X^T)^\dagger Y^T = (YX^\dagger)^T = A^T$ i sada iz (3.3.4) djelovanje Koopmanovog operatora $\mathcal{K}$ možemo zapisati kao

$$\mathcal{K}(g_1(x) \dots g_n(x)) = (g_1(x) \dots g_n(x))K + (\rho_1(x) \dots \rho_n(x)). \quad (3.3.6)$$

Nadalje pretpostavimo da je X punog retčanog ranga, to onda znači da je X^T punog stupčanog ranga pa K postoji i jedinstvena je. Pretpostavimo još da je matrica K dijagonalizabilna, tj postoji matrica Q i dijagonalna matrica Λ tako da $K = Q\Lambda Q^{-1}$. Sada iz (3.3.6) imamo:

$$\begin{aligned} \mathcal{K}(g_1(x) \dots g_n(x))Q &= (g_1(x) \dots g_n(x))KQ + (\rho_1(x) \dots \rho_n(x))Q = \\ &= (g_1(x) \dots g_n(x))Q\Lambda + (\rho_1(x) \dots \rho_n(x))Q \end{aligned} \quad (3.3.7)$$

pa su aproksimacije svojstvenih funkcija φ operatora $\mathcal{K}$ oblika $(g_1, \dots, g_n)Q$. Neka je sada $f(x)^T = (f_1(x) \dots f_n(x))$ neka vektorska funkcija stanja i $f_i(x) = \sum_{j=1}^n \gamma_{ji}g_j(x) + e_i(x)$, ako je $f_i \in \mathcal{N}$, $\mathcal{N} = \{g_1, \dots, g_n\}$ tada je $e_i = 0$, za svaki i a ako je $f_i = g_i$ tada je $\Gamma = (\gamma)_{ji} = I_n$. Nadalje, vrijedi

$$\begin{aligned} f(x)^T &= (g_1(x) \dots g_n(x))\Gamma + E(x) = (g_1(x) \dots g_n(x))QQ^{-1}\Gamma + E(x) = \\ &= (\varphi_1(x) \dots \varphi_n(x))Q^{-1}\Gamma + E(x) \end{aligned}$$

gdje je $\Gamma = (\gamma)_{ji} \in \mathbb{C}^{n \times n}$, $E(x) = (e_1(x) \dots e_n(x))^T$. Ako sada uzmemo $V = \Gamma^T Q^{-T} = (v_1 \dots v_n)$ imamo

$$\begin{pmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{pmatrix} = \Gamma^T Q^{-T} \begin{pmatrix} \varphi_1(x) \\ \vdots \\ \varphi_n(x) \end{pmatrix} + E(x)^T = \sum_{j=1}^n v_j \varphi_j(x) + E(x)^T \approx \sum_{j=1}^n v_j \varphi_j(x). \quad (3.3.8)$$

Sada ćemo pretpostaviti da je $f_i = g_i$, za svaki i pa je $\Gamma = I_n$, tj. $V = Q^{-T}$. Sada imamo

$$\begin{aligned} KQ &= Q\Lambda \\ Q^T K^T &= \Lambda^T Q^T \\ Q^T A &= \Lambda Q^T \\ A &= Q^{-T} \Lambda Q^T \\ AQ^{-T} &= Q^{-T} \Lambda \end{aligned}$$

gdje smo u trećem redu iskoristili to da je $K = A^T$ i $\Lambda^T = \Lambda$. Vidimo da su stupci matrice Q^{-T} svojstveni vektori matrice A pa je za računanje Koopmanovih modova dovoljno izračunati svojstvene vektore matrice A a to možemo pomoću DMD metode.

Napomena 3.3.2. *Kao što smo mogli primijetiti, ovu teoriju smo gradili na izboru funkcija stanja g_i . Problem kojeg nailazimo prilikom analiziranja Koopmanovih operatora je upravo kako izabrati funkcije stanja koje razapinju potprostor u kojem tražimo aproksimaciju.*

Poglavlje 4

Procesiranje videa

U ovom poglavlju objasniti ćemo kako pomoću DMD algoritima možemo prilikom procesiranja videa odvojiti pozadinu video (*background*) i prvi plan (*foreground*). Usporedit ćemo rezultate s rezultatima dobiveno RPCA metodom (*Robust Principal Component Analysis*), jednom od najrasprostranjenijih metoda trenutno koja se koristi u separaciji videa na pozadinu i prvi plan. Kao primjer video procesiranja promatrat ćemo snimke nadzornih kamera. Nadzorne kamere snimaju 20-30 kadrova (*frames*) u sekundi, i kao takve su jako pristupačne za DMD dekompoziciju jer te slike možemo promatrati kao evoluciju nekog nelinearnog dinamičkog sustava. DMD modovi koji sadrže frekvencije bliže izvoru (nuli) su povezani sa stanjima koji se ili ne mijenjaju kroz vrijeme ili mijenjanju jako sporo pa kao takvi predstavljaju stacionarni dio video, dok one koji su udaljeni od izvora se mijenjaju brže kroz vrijeme i predstavljaju dinamički dio videa, tj. prvi plan. Prilikom razrade ovog poglavlja koristili smo [6] i [7].

4.1 RPCA u procesiranju videa

Analiza glavnih komponenti (eng. *Principal component analysis*), (PCA) je metoda koja se koristi u obradi podataka kao alat za redukciju dimenzije problema. To je jednostavna neparametrizirana metoda koja se koristi za ekstrakciju važnih informacija (prediktora), tj. onih koji najbolje opisuju varijabilnosti nekog skupa podataka. RPCA (eng. *Robust Principal component analysis*) je proširenje PCA, to je dekompozicija matrice X na matricu nižeg ranga L i rijetku matricu S tako da $X = L + S$. PCA metoda koristi L^2 normu za optimizacijske probleme zbog pogodnih matematičkih svojstava te kao dobar opis "energije" sustava, no L^2 norma je osjetljiva na velika odstupanja u podacima i zato PCA može dati krive rezultate davajući veću važnost tim podacima nego što želimo. Norma pogodna za takvu obradu podataka je L^1 norma jer ona bolje regulira težine koje pridajemo tim podacima. RPCA algoritam kombinira te dvije norme i daje snažniju redukciju dimenzi-

onalnosti za potencijalno rijetke podatke i time poboljšava klasični PCA. RPCA algoritam radi tako da na osnovu danih podataka, traži rijetke strukture, dok istovremeno ”nado-punjuje” ostale podatke podacima nižeg ranga. Ukoliko se ulazni podaci mogu svesti na niže-dimenzionalan potprostor i ima rijetke komponente, RPCA algoritam egzaktno razdvaja matricu $X = L + S$.

Ovu dekompoziciju moguće je postići rješavajući konveksni optimizacijski problem PCP (*Principal Component Pursuit*)

$$\begin{aligned} \arg \min_{L, S} \|L\|_* + \lambda \|S\|_1 \\ \text{t.d. } L + S = X. \end{aligned} \quad (4.1.1)$$

PCP minimizira težinsku kombinaciju nuklearne norme $\|M\|_* = \text{tr}(\sqrt{M^*M})$ i L^1 norme $\|M\|_1 = \sum_{ij} |m_{ij}|$. Parametar λ je nenegativan i kako $\lambda \rightarrow 0$, struktura nižeg ranga postaje čitav skup danog podatka a kako se λ povećava, rijetki dio podataka obuzima sve više početnih podataka. Pokazano je da za $\lambda = 1/\sqrt{\max(n, m)}$, gdje su n i m dimenzije matrice podataka X , postoji velika vjerojatnost uspjeha u aproksimaciji matrica L i S . RPCA algoritam koji smo koristili u ovom radu rješava optimizacijski problem (4.1.1) ADMM metodom (*Alternating Direction Method of Multipliers*).¹

4.2 DMD u procesiranju videa

Za razliku od RPCA koji nam, pod pretpostavkom da je dani skup podataka zaista ovakve prirode, tj. da leži na niže-dimenzionalnom potprostoru i sadrži rijetke dijelove, daje egzaktnu separaciju na dio nižeg ranga i na rijetki dio, ne možemo to isto reći i za DMD, no DMD metoda je brža jer se bazira na samo jednoj SVD dekompoziciji i rješavanju jedne linearne jednačbe. Prisjetimo se kako smo rekonstruirali rješenja dinamičkog sustava u poglavlju 2.3, dobili smo

$$x(t) \approx \sum_{k=1}^r z_k e^{\omega_k t} b_k$$

kao rekonstrukciju rješenja. Kao što smo rekli na početku poglavlja, podaci nižeg ranga se sporo mijenjaju kroz vrijeme pa im je pridružena frekvencija $|\omega_k| \approx 0$, stoga, kod DMD-a uspješnost separacije ovisi o interpretaciji ω_k frekvencija. Vidimo da matricu nižeg ranga

¹Kod za RPCA algoritam korišten u ovom radu se može preuzeti na <https://github.com/dlaptev/RobustPCA>

L i rijetku matricu S možemo reprezentirati kao

$$\begin{aligned} L &\approx \sum_{|\omega_k| \leq \epsilon} b_k z_k e^{\omega_k t} \\ S &\approx \sum_{|\omega_k| > \epsilon} b_k z_k e^{\omega_k t} \end{aligned} \quad (4.2.1)$$

gdje je $\epsilon \ll 1$ neka zadana tolerancija.

Neka je dan video koji sadrži m kadrova gdje se svaki kadar sastoji od $n_x \times n_y = n$ piksela koje vektoriziramo u $n \times 1$ vektore $x_1, x_2, \dots, x_m$. Te vektore slažemo u matrice kao u (2.2.11) i (2.2.12). Pretpostavimo da smo pomoću DMD algoritma (algoritam 2) rekonstruirali video:

$$X_{DMD} = \sum_{k=1}^r b_k z_k e^{\omega_k t} \quad (4.2.2)$$

Kako je $x_1 = Z_r b$ gdje je $Z_r = (z_1, \dots, z_r)$ matrica dobivena algoritmom 2, zaključujemo da $Z_r b$ prikazuje prvi kadar videa s redukcijom dimenzionalnosti određena parametrom r . Stoga, dijagonalna matrica svojstvenih vrijednosti $\Omega = \text{diag}(\omega_1, \dots, \omega_r)$ određuje kako se taj prvi kadar promijeni kroz vrijeme, pa postaje očito da svaki dio prvog kadra koji se ne mijenja kroz vrijeme (ili se mijenja jako sporo) ima pridruženu vrijednost ω_k blizu nule. Pretpostavimo sada da je $|\omega_p| \approx 0$, $p \in \{1, 2, \dots, r\}$ i da za svaki $k \neq p$, $|\omega_k|$ je "daleko" od nule. Sada možemo razdvojiti matricu (4.2.2) kao

$$X_{DMD} = b_p z_p e^{\omega_p t} + \sum_{k \neq p} b_k z_k e^{\omega_k t} \quad (4.2.3)$$

gdje prvi dio sumacije predstavlja ono što je u pozadini videa a drugi dio predstavlja prvi plan videa. Matricu nižeg ranga L dobijemo kao:

$$L = b_p z_p e^{\omega_p t} \quad (4.2.4)$$

a kako treba vrijediti

$$X = L + S, \quad (4.2.5)$$

matricu S možemo dobiti kao

$$S = X - |L| \quad (4.2.6)$$

gdje je $|\cdot|$ modul svakog elementa u matrici. Ovako računanje matrice S je pogodno jer time izbjegavamo moguće kompleksne elemente u matrici. Također, zbog toga može doći do pojave negativnih vrijednost kod rekonstrukcije rijetke matrice, a nema smisla govoriti

o negativnim pikselima stoga te negativne vrijednosti možemo spremiti u $n \times m$ matricu R koju kasnije dodamo natrag u L :

$$\begin{aligned} L &\leftarrow R + |L| \\ S &\leftarrow S - R. \end{aligned} \quad (4.2.7)$$

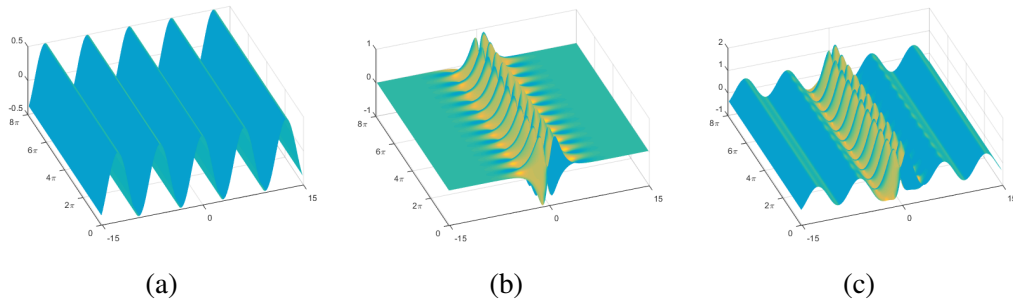
Na ovaj način osiguravamo to da su aproksimacija matrica L i S realne vrijednosti i da ne dođe do negativnih piksela.

Primjer 4.2.1. Pogledajmo prvo na jednostavnom primjeru kako separacija DMD algoritmom funkcionira. Neka nam je dana funkcija

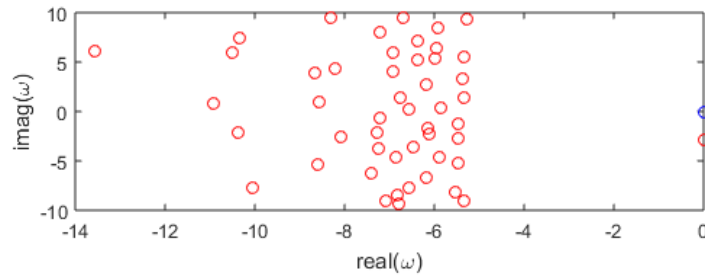
$$f(x, t) = f_1(x) + f_2(x, t) = 0.5 \cos(x) + 2 \operatorname{sech}(x) \tan(x) e^{i2.8t} \quad (4.2.8)$$

gdje je $i = \sqrt{-1}$.

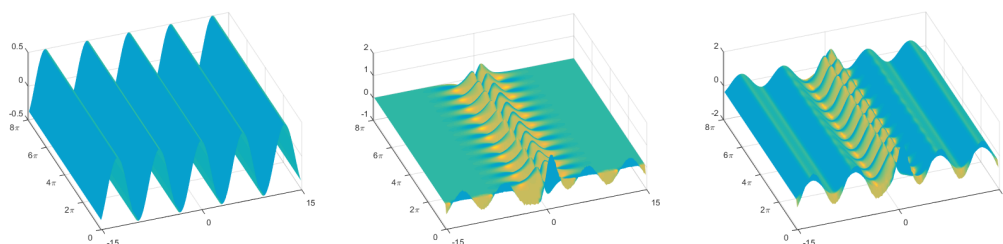
kao što vidimo, funkcija f_1 ne ovisi o vremenu i očekujemo da će ju DMD klasificirati kao pozadinu.



Slika 4.1: Na slici (a) je prikazan realni dio funkcije $f_1(x)$, na slici (b) je prikazan realni dio funkcije $f_2(x, t)$ a na slici (c) je prikazan realni dio funkcije $f(x, t)$.



Slika 4.2: Na slici je prikazana distribucija svojstvenih vrijednosti ω_j gdje se vidi da sa samo jednim modom (plavi kružić) možemo rekonstruirati pozadinu.



Slika 4.3: Na slici (a) je prikazana "pozadina" koji nam daje DMD algoritam a na slici (b) je prikazan "prvi plan". Vidimo da je DMD klasificirao funkciju $f_1(x)$ kao "pozadinu", tj. nešto što se ne mijenja kroz vrijeme. Na slici (c) je prikazan realni dio funkcije dobivene rekonstrukcijom funkcije $f(x, t)$ DMD algoritmom.

4.3 Testiranje algoritma

VIRAT Video dataset

Za testiranje algoritma prvo smo koristili snimak nadzorne kamere preuzet iz VIRAT Video dataset [9]. Video je u .mp4 formatu, traje 24 sekunde i sadrži 584 kadrova. Radi lakšeg izvođenja, rezoluciju videa smo reducirali na 96×120 te je prebačen iz *RGB* u *grayscale* format.



Slika 4.4

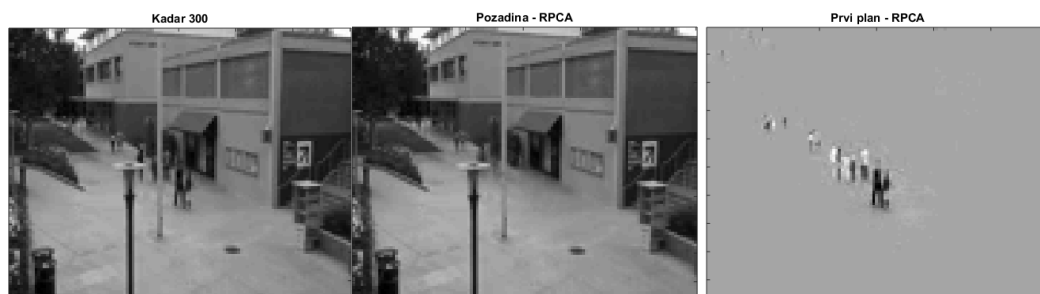
Slika 4.5: Na gornjim slikama prikazan je kadar broj 45 u videu i njegova separacija na pozadinu i prvi plan dobivena DMD algoritmom, kao parametar redukcije dimenzije uzeli smo $r = 2$.

Vidimo da s parametrom redukcije dimenzije $r = 2$ DMD daje dobru aproksimaciju

pa smo i kasnije u testiranju iste baze koristili isti parametar. Kao parametar λ u RPCA algoritmu smo koristili $\lambda = 1 / \sqrt{\max(n, m)}$ gdje su n i m dimenzije ulazne matrice podataka.



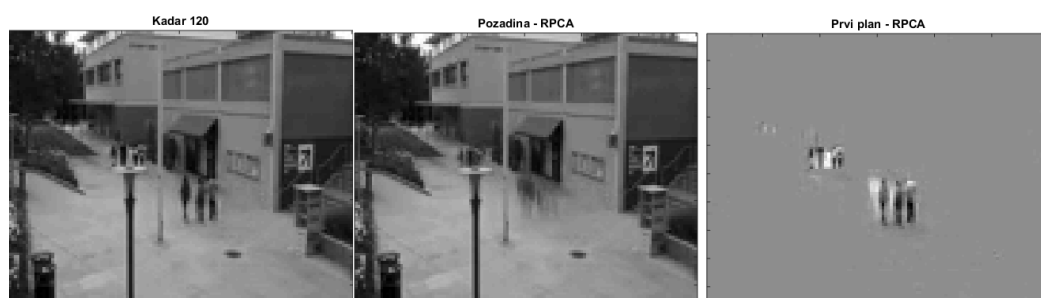
Slika 4.6: Na gornjim slikama prikazan je kadar broj 300 u videu i njegova separacija na pozadinu i prvi plan dobivena DMD algoritmom



Slika 4.7: Na gornjim slikama prikazan je kadar broj 300 u videu i njegova separacija na pozadinu i prvi plan dobivena RPCA algoritmom.



Slika 4.8: Na gornjim slikama prikazan je kadar broj 120 u videu i njegova separacija na pozadinu i prvi plan dobivena DMD algoritmom.



Slika 4.9: Na gornjim slikama prikazan je kadar broj 120 u videu i njegova separacija na pozadinu i prvi plan dobivena RPCA algoritmom.

AVSS Parked Vehicle

Druga baza na kojoj smo testirali algoritam je AVSS "Parked Vehicle" snimak nadzorne kamere [1]. Video je u .avi formatu, traje 2.54 min. Video smo reducirali na dimenzije 96×120 te prebacili iz *RGB* u *grayscale* format. Prvih 315 kadrova video smo zanemarili jer nisu bitne za treniranje algoritma, te nam tako ostane 4010 kadrova videa. Radi lakšeg računanja, video smo podijelili u manje videe po 100 kadrova, te s njima trenirali algoritam i uspoređivali sa RPCA algoritmom.



Slika 4.10: Na gornjim slikama prikazan je kadar broj 100 u videu i njegova separacija na pozadinu i prvi plan dobivena DMD algoritmom, kao parametar redukcije dimenzije uzeli smo $r = 2$.

Vidimo opet da s parametrom redukcije $r = 2$ dobivamo dobru aproksimaciju pa smo i dalje u testiranju iste baze koristili taj parametar. Kao parametar λ u RPCA algoritmu smo koristili $\lambda = 1/\sqrt{\max(n, m)}$ gdje su n i m dimenzije ulazne matrice podataka.



Slika 4.11: Na gornjim slikama prikazan je kadar broj 100 u videu i njegova separacija na pozadinu i prvi plan dobivena RPCA algoritmom.



Slika 4.12: Na gornjim slikama prikazan je kadar broj 200 u videu i njegova separacija na pozadinu i prvi plan dobivena DMD algoritmom.



Slika 4.13: Na gornjim slikama prikazan je kadar broj 200 u videu i njegova separacija na pozadinu i prvi plan dobivena RPCA algoritmom.

Poglavlje 5

Varijacije DMD algoritma

U drugom poglavlju prilikom formulacije DMD algoritma rekli smo da koristimo reducirani SVD matrice no nismo davali više informacija o tom parametru redukcije. U ovom poglavlju ćemo vidjeti kako možemo odrediti rang r i općenito kako poboljšati DMD algoritam, pokazat ćemo kako bolje izabrati Rayleigh-Ritzove parove i pokazati da umjesto SVD možemo koristiti QR dekompoziciju da bismo zadovoljili dodatne uvjete.

5.1 Određivanje parametra redukcije dimenzije

Parametar redukcije dimenzije r nije trivijalan i cilj je da bude jednak numeričkom rang matrice podataka X . Po teoremu 1.2.2 znamo da stupci matrice U_r razapinju najbolji r -dimenzionalni potprostor koji zahvaća podatke u smislu najmanjih kvadrata. Naime, ako je W neka $n \times r$ matrica tako da $W^*W = I_r$, tada po 1.2.2 imamo:

$$\begin{aligned} \sum_{i=1}^m \|x_i - WW^*x_i\|_2^2 &= \|X - W(W^*X)\|_F^2 \geq \text{jer je } \text{rang}(WW^*X) \leq r \\ &\geq \|X - U_r \Sigma_r V_r^* X\|_F^2 = \|X - U_r U_r^* X\|_F^2 = \sum_{i=1}^m \|x_i - U_r U_r^* x_i\|_2^2 = \sum_{i=r+1}^m \sigma_i^2 \end{aligned}$$

Primijetimo da za ovako konstruiran potprostor ne mora vrijediti da je x_i u slici matrice U_r . Sada iz 1.2.2 numerički rang r možemo izračunati kao

$$r = \max\{i : \sigma_i > \epsilon \sigma_1\} \quad (5.1.1)$$

gdje je $\epsilon \in (0, 1)$ zadana tolerancija a σ_i singularne vrijednosti matrice. Ovakav način određivanja parametra redukcije r ćemo koristiti dalje u radu.

5.2 Poboljšani Rayleigh-Ritzovi parovi

Općenito, nije svaka Rayleigh-Ritzova aproksimacija (λ_i, z_i) dobra aproksimacija traženog svojstvenog para matrice A . Ovdje ćemo dati ideju kao poboljšati te aproksimacije. Za razradu ovog poglavlja koristili smo [4] i [5]. Neka su sada $Z_r = U_r W_r$ i Λ_r matrice dobivene DMD algoritmom (algoritam 1) gdje je Z_r matrica svojstvenih vektora, a Λ_r dijagonalna matrica svojstvenih vrijednosti.

Cilj je minimizirati rezidual $Az_i - \lambda_i z_i$:

$$r_i = \|Az_i - \lambda_i z_i\|_2 = \|A(U_r W_r(:, i)) - \lambda_i(U_r W_r(:, i))\|_2 = \|(YV_r \Sigma_r^{-1})W_r(:, i) - \lambda_i(U_r W_r(:, i))\|_2 \quad (5.2.1)$$

gdje smo u trećoj jednakosti iskoristili $AU_r = YV_r \Sigma_r^{-1}$. Vidimo da traženi rezidual možemo računati bez direktnog računanja matrice A . No vektor $z = U_r w$ možda nije uvijek najbolji za minimizaciju reziduala (5.2.1), za vrijednost λ možda postoji bolji vektor $\tilde{z}$ u slici matrice U_r koji ima manji rezidual. Smisleno je promatrati sljedeći optimizacijski problem:

$$\min_{z \in \mathcal{U}_r \setminus \{0\}} \frac{\|Az - \lambda z\|_2}{\|z\|_2} = \min_{w \neq 0} \frac{\|AU_r w - \lambda U_r w\|_2}{\|U_r w\|_2} = \min_{\|w\|_2=1} \|(AU_r - \lambda U_r)w\|_2 \quad (5.2.2)$$

gdje je $\mathcal{U}_r$ predstavlja sliku matrice U_r . Rješenje ovog problema je $\sigma_{\min}(AU_r - \lambda U_r)$ gdje $\sigma_{\min}(\cdot)$ označuje najmanju singularnu vrijednost matrice i ona se postiže u desnom singularnom vektoru w_λ koji odgovara singularnoj vrijednosti $\sigma_\lambda \equiv \sigma_{\min}(AU_r - \lambda U_r)$. Kao rezultat toga, poboljšani Ritzov vektor $\tilde{z}$ koji odgovara singularnoj vrijednosti λ je $\tilde{z} = U_r w_\lambda$ i optimalni rezidual jednak je σ_λ . Sada, minimizaciju reziduala (5.2.1) možemo dobiti računanjem singularne dekompozicije matrice $(AU_r - \lambda U_r) = (YV_r \Sigma_r^{-1} - \lambda U_r)$ tj. najmanje singularne vrijednosti i pripadnog singularnog vektora. Trošak računanja možemo dalje smanjiti, naime, za računanje reziduala (5.2.1) potrebno je računati singularnu dekompoziciju matrice $n \times r$, tu dimenziju možemo dalje reducirati na sljedeći način. Promotrimo QR dekompozicije sljedeće matrice, radi lakšeg zapisa neka je $B_r = YV_r \Sigma_r^{-1}$:

$$(U_r \quad B_r) = QR, \quad R = \begin{pmatrix} R_{[11]} & R_{[12]} \\ 0 & R_{[22]} \end{pmatrix} \quad (5.2.3)$$

gdje su $R_{[11]}$ i $R_{[12]} \in \mathbb{C}^{r \times r}$ a $R_{[22]} \in \mathbb{C}^{r' \times r}$, gdje je $r' = \min(n - r, r)$. Sada možemo $B_r - \lambda U_r$ izraziti kao

$$B_r - \lambda U_r = Q \left(\begin{pmatrix} R_{[12]} \\ R_{[22]} \end{pmatrix} - \lambda \begin{pmatrix} R_{[11]} \\ 0 \end{pmatrix} \right) \equiv QR_\lambda \quad (5.2.4)$$

gdje je

$$R_\lambda = \begin{pmatrix} R_{[12]} - \lambda R_{[11]} \\ R_{[22]} \end{pmatrix}. \quad (5.2.5)$$

Sada se problem svodi na računanje najmanje singularne vrijednosti i pripadnog vektora $2r \times r$ matrice $R_{[1]}$, a $r \leq m \ll n$.

Navodimo sada teorem koji nam govori kako možemo izračunati Rayleighev kvocijent.

Propozicija 5.2.1. *U QR faktorizaciji (5.2.3), definirajmo $\Phi = \text{diag}((R_{[11]})_{ii})_{i=1}^r$. Tada je $\Phi^* R_{[12]} = U_r^* B_r \equiv S_r$*

Dokaz. Ako matricu Q iz (5.2.3) particioniramo kao $Q = (Q_1 \quad Q_2)$, gdje Q_1 ima r stupaca, tada je QR faktorizacija matrice U_r dana kao $U_r = Q_1 R_{[11]}$. U_r je unitarna matrica, tj. vrijedi $U_r^* U_r = I_r$ i kako je QR faktorizacija jedinstvena, $\Phi \equiv R_{[11]}$ je dijagonalna unitarna matrica. Sada imamo $U_r = U_r I_r = (Q_1 \Phi)(\Phi^* R_{[11]}) = (Q_1 \Phi) I_r$ pa je $U_r = Q_r \Phi$. Stoga, $U_r^* B_r = \Phi^* Q_1^* B_r = \Phi^* R_{[12]}$. $\square$

Prije nego krenemo na sam algoritam, prokomentirat ćemo podatke sadržane u uzorcima x_i i njihovo ponašanje kada sustav aproksimiramo sustavom nižeg ranga. Naime, vektori x_i koji su male norme u odnosu na $\max_i \|x_i\|$ se mogu izgubiti prilikom reduciranja sustava što nije uvijek poželjno ako nam ti vektori nose važne informacije. Da bismo to izbjegli, skaliramo ulazne matrice podataka $X \in \mathbb{C}^{n \times m}$ i $Y \in \mathbb{C}^{n \times m}$. Primjetimo da ako vrijedi $Y = AX$, tada za svaku D invertibilnu, dijagonalnu $m \times m$ matricu vrijedi $(YD) = A(XD)$. Kao matricu D možemo izabrati $D_X = \text{diag}(1/\|X(:, i)\|_2)_1^m$ ili $D_Y = \text{diag}(1/\|Y(:, i)\|_2)_1^m$.

Sumirajući prethodne rezultate dobijemo sljedeći algoritam:

Ulaz: $X = (x_1 \dots x_m)$, $Y = (x_2 \dots x_{m+1})$; matrice podataka
 ϵ ; tolerancija za određivanje numeričkog ranga

Izlaz: Z_r , Λ_r , res_r , ρ_r

1. $D = \text{diag}(\|X(:, i)\|_2)_{i=1}^m$; $X^{(1)} = XD^\dagger$, $Y^{(1)} = YD^\dagger$;
2. $[U, \Sigma, V] = \text{svd}(X^{(1)})$;
3. odredi parametar redukcije r s tolerancijom ϵ ;
4. $U_r = U(:, 1 : r)$, $\Sigma_r = \Sigma(1 : r, 1 : r)$, $V_r = V(:, 1 : r)$;
5. $B_r = Y^{(1)}(V_r \Sigma_r^{-1})$;
6. $[Q, R] = \text{qr}(U_r \quad B_r)$;
7. $S_r = \text{diag}((\bar{R}_{ii}))_{i=1}^r R(1 : r, r + 1 : 2r)$;
8. $\Lambda_r = \text{eig}(S_r)$, $(\Lambda_r = \text{diag}(\lambda_i)_{i=1}^r)$;
9. **Za** $i=1:r$ **radi**

$$[\sigma_{\lambda_i}, w_{\lambda_i}] = \text{svd}_{\min} \left(\begin{pmatrix} R(1 : r, r + 1 : 2r) - \lambda_i R(1 : r, 1 : r) \\ R(r + 1 : 2r, r + 1 : 2r) \end{pmatrix} \right);$$

$$W_r(:, i) = w_{\lambda_i};$$

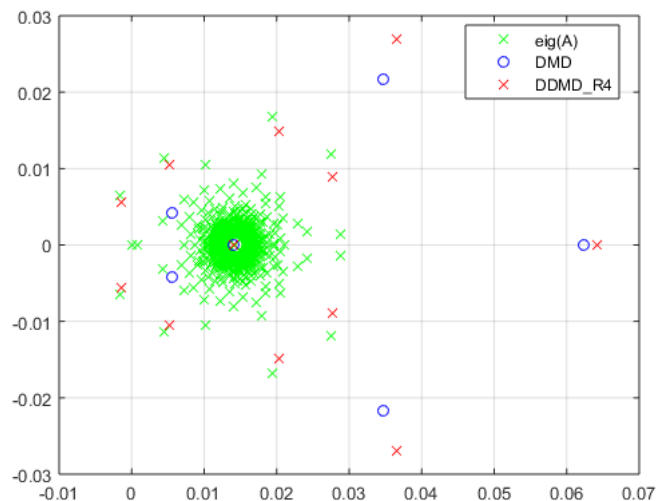
$$res_r(i) = \sigma_{\lambda_i};$$

$$\rho_r(i) = w_{\lambda_i}^* S_r w_{\lambda_i}$$

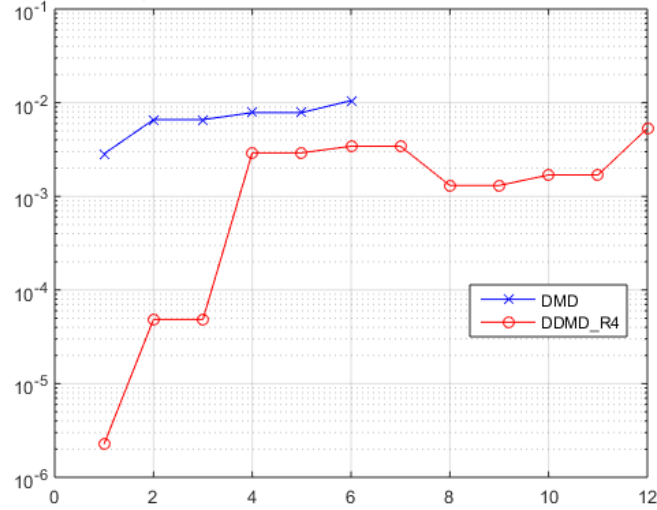
kraj

10. $Z_r = U_r W_r$

Algoritam 3: DDMD_R4 {Refined Rayleigh-Ritz Data Driven Modal Decomposition}



Slika 5.1: Na slici je prikazana usporedba svojstvenih vrijednosti dobivenih *DMD* i *DDMD_R4* algoritmima na nasumično generiranim podacima. Prvo smo generirali matricu B dimenzija 1000×1000 i vektor stanja x_1 dimenzije 1000×1 pomoću Matlabove funkcije *rand*, dalje smo matricu A izračunali kao $A = \exp(-\text{inv}(B))$ te skalirali $A = A/\|A\|_2$ i dalje računali vektore stanja kao $x_i = Ax_{i-1}$, $i = 2, \dots, 100$. Kao toleranciju za parametar redukcije koristili smo $\epsilon = n\varepsilon$, gdje je $\varepsilon = 1e-12$. Za isti kriterij *DMD* je sustav reducirao na rang $r = 6$ a *DMD_R4* $r = 12$. Također, kako je poznata matrica A izračunali smo svojstvene vrijednosti matrice A pomoću Matlabove funkcije *eig*.



Slika 5.2: Na slici je prikazana usporedba reziduala dobivena *DMD* i *DDMD_R4* algoritmima na nasumično generiranom podatku kao na slici 5.1. Rezidual smo računali egzaktno; $\|Az_i - \lambda_i z_i\|_2$

Pogledajmo sada kako dalje možemo reducirati dimenziju problema. Neka su opet dane matrice podataka $X \in \mathbb{C}^{n \times m}$ i $Y \in \mathbb{C}^{n \times m}$ tako da $Y = AX$ i neka je QR faktorizacija sljedeća:

$$(X \ Y) = Q_{xy} \begin{pmatrix} R_{[11]} & R_{[12]} \\ 0 & R_{[22]} \\ 0 & 0 \end{pmatrix} = \widehat{Q}_{xy} R_{xy} \quad (5.2.6)$$

gdje je

$$R_{xy} = \begin{pmatrix} R_{[11]} & R_{[12]} \\ 0 & R_{[22]} \end{pmatrix}$$

a $\widehat{Q}_{xy} = Q_{xy}(:, 1 : 2m)$.

Neka su matrice $R_x = R_{xy}(1 : 2m, 1 : m)$, $R_y = R_{xy}(1 : 2m, m + 1 : 2m)$, sada vrijedi $X = \widehat{Q}_{xy} R_x$ i $Y = \widehat{Q}_{xy} R_y$. U bazi stupaca matrice $\widehat{Q}_{xy}$ možemo identificirati matrice $X \equiv R_x$, $Y \equiv R_y$ tj. možemo naše početne matrice podataka X i Y sada promatrati kao m vektora stanja u $2m$ -dimenzionalnom prostoru. Ako sada na te matrice primijenimo algoritam 3¹ dobit ćemo matricu aproksimacija svojstvenih vrijednosti Λ_r i odgovarajuće svojstvene vektore kao stupce matrice $\widehat{Z}_r \in \mathbb{C}^{(2m) \times r}$. Da bismo prikazali te svojstvene vektore u terminima polaznih podataka, tj. kao elemente $\mathbb{C}^n$ moramo pomnožiti $Z_r = \widehat{Q}_{xy} \widehat{Z}_r$ i time dobijemo

¹ Isto to smo mogli primijeniti i na algoritam 1

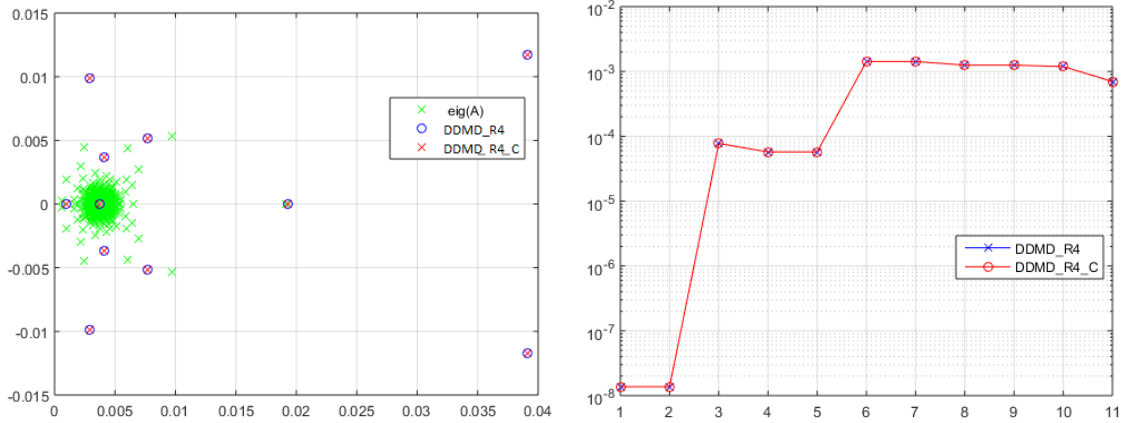
kompresiranu verziju algoritma 3 (*DDMD_R4_C*). Ovim načinom smo reducirali dimenziju n na mnogo manju dimenziju $2m$. Sada dobijemo sljedeći algoritam:

Ulaz: $X = (x_1 \dots x_m)$, $Y = (x_2 \dots x_{m+1})$; matrice podataka
 ϵ ; tolerancija za određivanje numeričkog ranga

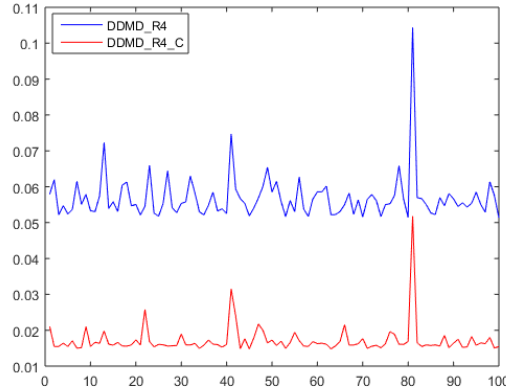
Izlaz: Z_r , Λ_r , res_r , ρ_r

1. $[Q_{xy} \ R_{xy}] = qr(X \ Y)$;
2. $\widehat{Q}_{xy} = Q_{xy}(:, 1 : 2m)$, $R_x = R_{xy}(1 : 2m, 1 : m)$, $R_y = R_{xy}(1 : 2m, m + 1 : 2m)$;
3. $[\widehat{Z}_r, \Lambda_r, res_r, \rho_r] = DMD_R4(R_x, R_y, \epsilon)$;
4. $Z_r = \widehat{Q}_{xy} \widehat{Z}_r$

Algoritam 4: *DDMD_R4_C* {*Compressed Refined Rayleigh-Ritz Data Driven Modal Decomposition*}



Slika 5.3: Nasumično smo generirali matricu A dimenzija 500×500 i vektor x_1 dimenzija 500×1 pomoću Matlabove funkcije *rand* i dalje računali niz podataka kao $x_i = Ax_{i-1}, i = 2, \dots, 50$. Na lijevoj slici su prikazane svojstvene vrijednosti dobivene DDMD_R4 i DDMD_R4_C algoritmom kao i svojstvene vrijednosti matrice A koje smo izračunali pomoću Matlabove funkcije *eig*. Na slici desno je prikazan rezidual dobiven algoritmom. Kao toleranciju za parametar redukcije dimenzije koristili smo $\epsilon = n\epsilon$, gdje je $\epsilon = 1e - 12$. Parametar redukcije dimenzije u oba algoritma je $r = 11$.



Slika 5.4: Na slici je prikazano ukupno vrijeme potrebno za izvršavanje DDMD_R4 i DDMD_R4_C algoritma u sekundama (y-os), na nasumično generiranim podacima kao kod slike 5.3. Algoritme smo pokrenuli 100 puta (svaki put na različitim podacima).

5.3 Redukcija dimenzije korištenjem QR dekompozicije

Dosada je svaki prikazani DMD algoritam u ovom radu baziran na SVD-u. No međutim računanje SVD-a nekada može biti skupo i još važnije, može se dogoditi da se uzorci x_i ne nalazi u prostoru koji razapinju stupci matrice U_r , tj. x_i nije u slici matrice U_r , gdje je U_r matrica singularnih vektora dobivena reduciranim SVD-om matrice podataka. Ponekad je baš od koristi da nam aproksimaciju nižeg ranga razapinju točno odabrani uzorci x_i a to možemo postići ako umjesto SVD primijenimo QR dekompoziciju na ulaznu matricu podataka. Sljedeća razrada poglavlja je preuzeta iz [4]. Mi ćemo dati kratak opis metode, za detalje pogledati literaturu.

Neka su $X \in \mathbb{C}^{n \times m}$ i $Y = \mathbb{C}^{n \times m}$ matrice tako da vrijedi $AX = Y$. Radimo QR faktorizaciju matrice X s pivotiranjem:

$$X\Pi = QR, \quad Q \in \mathbb{C}^{n \times m}, \quad R \in \mathbb{C}^{m \times m}. \quad (5.3.1)$$

Koristimo QR faktorizaciju s pivotiranjem da bismo lakše očitali rang matrice R . Ako je numerički rang matrice X jednak $r < m$ tada će $R_r = R(1 : r, 1 : r)$ biti dominantan dio matrice R . Najčešće korištenu strategiju za pivotiranje su razvili Businger i Golub [2] i ona osigurava

$$|R_{ii}| \geq \sqrt{\sum_{r=1}^j |R_{kj}|^2}, \quad 1 \leq i \leq j \leq m.$$

Stoga, ako matrica X nije punog ranga, tada će R imati sljedeću blok strukturu, za neki $\mu > 0$ i $1 \leq r < m$

$$R = \begin{pmatrix} R_{[11]} & R_{[12]} \\ 0 & R_{[22]} \end{pmatrix}, \quad R_{[11]} \in \mathbb{C}^{r \times r}, \quad |R_{rr}| \geq \mu \mid R_{r+1,r+1} \mid \geq \frac{\mu}{\sqrt{m-r}} \|R_{[22]}\|_F.$$

Kako je numerički rang matrice X jednak r , $R_{[22]}$ biti malen u usporedbi sa $R_{[11]}$ pa imamo:

$$X\Pi \approx Q_r(R_{[11]} \quad R_{[12]}) \quad (5.3.2)$$

gdje je $Q_r = Q(:, 1 : r)$. Sada iz (5.3.1) vidimo da vrijedi $X \approx Q_r(R_{[11]} \quad R_{[12]})\Pi^T$, pa iz toga slijedi da matricu X u bazi matrice Q_r možemo prikazati kao $(R_{[11]} \quad R_{[12]})\Pi^T$ i time smo odredili ortonormalnu bazu, sada preostaje odrediti Rayleigh-Ritzove parove.

$$\begin{aligned} X\Pi &= QR \\ AX\Pi &= AQR \\ Y\Pi &= AQR. \end{aligned}$$

Neka je $R_r = R(1 : r, 1 : r)$, sada je $(Y\Pi)_r = (Y\Pi)(:, 1 : r) = AQR_r$, iz čega slijedi

$$AQR_r = (Y\Pi)_r R_r^{-1}. \quad (5.3.3)$$

Sada možemo računati Rayleighov kvocijent $S_r = Q_r^* AQR_r$ kao

$$S_r = (Q_r^* (Y\Pi)_r) R_r^{-1}. \quad (5.3.4)$$

Ostatak je analogan kao u DMD algoritmu samo sada umjesto U_r , Σ_r i V_r imamo Q_r , R_r i Π_r (redom), gdje je $\Pi_r = \Pi(:, 1 : r)$. Primijetimo da Q_r razapinje isti potprostor kao i r stupaca matrice X koje odaberemo matricom permutacije Π . Mi ćemo dalje nastaviti sa poboljšanim DMD algoritmom objašnjenim u poglavlju 5.2 (DDMD_R4). Tražene svojstvene vrijednosti bit će svojstvene vrijednosti matrice S_r a pripadne svojstvene vektore dobijemo kao $z = Q_r w$, gdje su w poboljšani Ritzovi vektori dobiveni kao u DDMD_R4 algoritmu.

Ulaz: $X = (x_1 \dots x_m)$, $Y = (x_2 \dots x_{m+1})$; matrice podataka
 ϵ ; tolerancija za određivanje numeričkog ranga

Izlaz: Z_r , Λ_r , res_r , ρ_r

1. $D = \text{diag}(\|X(:, i)\|_2)_{i=1}^m$; $X^{(1)} = XD^\dagger$, $Y^{(1)} = YD^\dagger$;
2. $[Q_x, R_x, \Pi_x] = qr(X^{(1)})$;
3. odredi rang r ;
4. $Q_r = Q_x(:, 1 : r)$, $\Pi_r = \Pi_x(:, 1 : k)$, $R_r = R_x(1 : r, 1 : r)$;
5. $B_r = (Y\Pi_r)R_r^{-1}$;
6. $[Q, T] = qr(Q_k \quad B_r)$;
7. $S_r = \text{diag}(\overline{T_{ii}})_{i=1}^r T(1 : r, r + 1 : 2r)$ (ili $S_r = Q_r^* B_r$);
8. $\Lambda_r = \text{eig}(S_r)$ ($\Lambda_r = \text{diag}(\lambda_i)_{i=1}^r$);
9. **Za** $i = 1, \dots, r$ **radi**

$$[\sigma_{\lambda_i}, w_{\lambda_i}] = \text{svd}_{\min} \left(\begin{pmatrix} T(1 : r, r + 1 : 2r) - \lambda_i T(1 : r, 1 : r) \\ T(r + 1 : 2r, r + 1 : 2r) \end{pmatrix} \right);$$

$$W_k(:, i) = w_{\lambda_i};$$

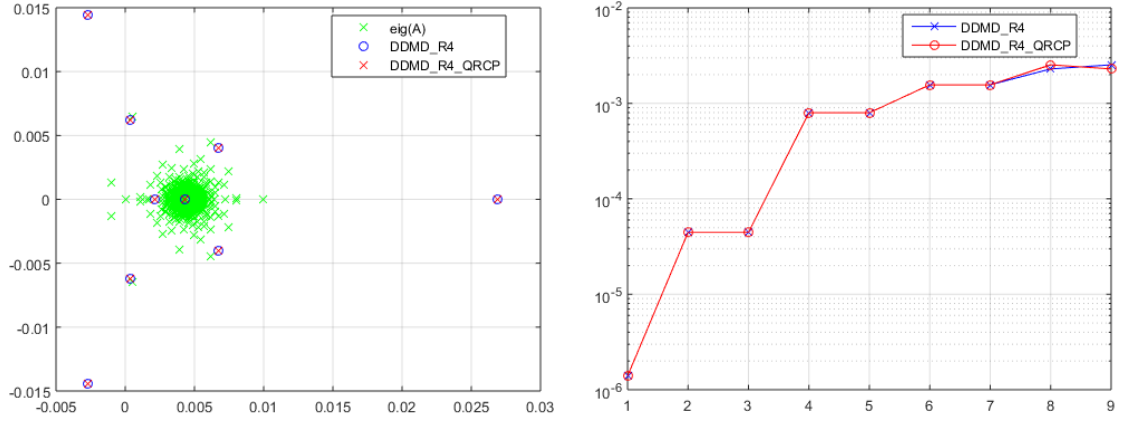
$$res_k(i) = \sigma_{\lambda_i};$$

$$\rho_r(i) = w_{\lambda_i}^* S_k w_{\lambda_i};$$

kraj

10. $Z_r = Q_r W_r$

Algoritam 5: DDMD_R4 {*Refined Rayleigh-Ritz Data Driven Modal Decomposition using QR with column pivoting*}



Slika 5.5: Nasumično smo generirali matricu A dimenzija 1000×1000 i vektor x_1 dimenzija 1000×1 pomoću Matlabove funkcije *rand* i dalje računali niz podataka kao $x_i = Ax_{i-1}, i = 2, \dots, 100$. Na lijevoj slici su prikazane svojstvene vrijednosti dobivene algoritmima DDMD_R4 i DDMD_R4_QRCP kao i svojstvene vrijednosti matrice A koje smo dobili pomoću Matlabove funkcije *eig*. Na slici desno su prikazani reziduali dobiveni algoritmima. Kao toleranciju za parametar redukcije dimenzije koristili smo $\epsilon = n\epsilon$, gdje je $\epsilon = 1e - 12$. Parametar redukcije dimenzije u oba algoritma je $r = 9$.

5.4 DMDSP

U ovom dijelu ćemo pokazati kako pomoću rezultata dobivenih DMD algoritmom možemo rekonstruirati dinamiku sustava na još jedan način, tj. kako rekonstruirati pojedine uzorke sustava x_i . Razrada ovog poglavlja je preuzeta iz [3].

Ritzov par (λ_j, z_j) koji nam daje DMD kao aproksimaciju svojstvenog para matrice A (pogledati algoritam 1) možemo iskoristiti za rekonstruiranje uzoraka x_i . Neka je $X = (x_1, \dots, x_m)$ matrica uzoraka, rekonstrukciju uzoraka dobijemo na način:

$$x_i = \sum_{j=1}^m z_j \alpha_j \lambda_j^{i-1}, \quad i = 1, \dots, m. \quad (5.4.1)$$

Kao i dosad, pokušavamo smanjiti dimenziju problema pa ćemo pokušati sa što manjim brojem modova z_j dobro aproksimirati dinamiku. Želimo odrediti $l < m$ i indekse $\zeta_1 < \dots < \zeta_l$ tako da aproksimacija idalje bude dobra:

$$x_i \approx \hat{x}_i = \sum_{j=1}^l z_{\zeta_j} \alpha'_j \lambda_{\zeta_j}^{i-1}, \quad i = 1, \dots, m. \quad (5.4.2)$$

Nakon što odredimo najbolji podskup modova, rješavamo problem najmanjih kvadrata:

$$\sum_{i=1}^m \|x_i - \hat{x}_i\|_2^2 \longrightarrow \min_{\alpha'_j} \quad (5.4.3)$$

U matičnom zapisu (5.4.1) izgleda kao:

$$X = \begin{pmatrix} z_1 & z_2 & \dots & z_m \end{pmatrix} \begin{pmatrix} \alpha_1 & & & \\ & \alpha_2 & & \\ & & \ddots & \\ & & & \alpha_m \end{pmatrix} \begin{pmatrix} 1 & \lambda_1 & \dots & \lambda_1^{m-1} \\ 1 & \lambda_2 & \dots & \lambda_2^{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_m & \dots & \lambda_m^{m-1} \end{pmatrix}. \quad (5.4.4)$$

Koeficijenti $(\alpha_j)_{j=1}^m$ se dobiju kao

$$(\alpha_j)_1^m = Z^\dagger x_1 \quad (5.4.5)$$

gdje je x_1 prvi stupac matrice X . Kako je $Z = UW$ (pogledati algoritam 1) i kako vrijedi $U^*U = I$ imamo

$$Z^\dagger x_1 = W^{-1}(U^* x_1) \quad (5.4.6)$$

pa vidimo da koeficijente α_j možemo dobiti bez da direktno primjenjujemo pseudo-inverz na matricu Z . Cilj reprezentacije (5.4.1) i (5.4.4) je vidjeti možemo li sa što manje modova što bolje aproksimirati dinamiku sustava, tj. matricu X . Pitanje je kako odrediti najbolji $l < m$ tako da vrijedi

$$X \approx \begin{pmatrix} z_{\zeta_1} & z_{\zeta_2} & \dots & z_{\zeta_l} \end{pmatrix} \begin{pmatrix} \alpha'_{\zeta_1} & & & \\ & \alpha'_{\zeta_2} & & \\ & & \ddots & \\ & & & \alpha'_{\zeta_l} \end{pmatrix} \begin{pmatrix} 1 & \lambda_{\zeta_1} & \dots & \lambda_{\zeta_1}^{m-1} \\ 1 & \lambda_{\zeta_2} & \dots & \lambda_{\zeta_2}^{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_{\zeta_l} & \dots & \lambda_{\zeta_l}^{m-1} \end{pmatrix} \equiv Z_\zeta D_\alpha \mathbb{V}_\zeta. \quad (5.4.7)$$

Želimo odabrati $z_{\zeta_1}, \dots, z_{\zeta_l}$ tako da je greška $\|X - Z_\zeta D_\alpha \mathbb{V}_\zeta\|_F^2$ što manja. Jedan od načina na koji možemo to postići je korištenjem metoda s rijetko popunjenim ograničenjima, primjerice ADMM (*Alternating Directions Method of Multipliers*) metoda koju koristimo u *Sparsity promoting DMD*. Nakon što izaberemo najvažnije modove, umjesto da izaberemo koeficijente α_{ζ_l} od potpune rekonstrukcije, rješavajući problem najmanjih kvadrata možemo doći do novih koeficijenata α'_{ζ_i} i tako postići optimalnije rezultate.

DMDSP (*Sparsity promoting DMD*)

Formulirajmo problem kao problem najmanjih kvadrata s rijetko popunjenim ograničenjem:

$$\|X - Z_r D_\alpha \mathbb{V}_{r,m}\|_F^2 + \gamma \|\alpha\|_0 \rightarrow \min_{\alpha} \quad (5.4.8)$$

gdje je $\|\alpha\|_0$ broj ne-nul elemenata u vektoru $\alpha = (\alpha'_i)_{i=1}^r$ novih koeficijenata, parametar $\gamma \geq 0$ penalizira rijetku popunjenost rješenja α . Oskudnost $\|\alpha\|_0$ možemo izračunati praktično koristeći l_1 normu, $\|\alpha\|_1$ pa sada imamo konveksni optimizacijski problem:

$$\|X - Z_r D_\alpha \mathbb{V}_{r,m}\|_F^2 + \gamma \|\alpha\|_1 \rightarrow \min_{\alpha} \quad (5.4.9)$$

koji rješavamo ADMM metodom i optimalni α prorjeđujemo postavljajući one vrijednosti koji su po modulu manji od nekog zadanog praga, na 0. Neka su sada $j_1, \dots, j_s$ indeksi tih vrijednosti koje smo postavili na 0 i neka je matrica E matrica čiji su stupci stupci identitete s indeksima $j_1, \dots, j_s$. Tada dobivenu rijetku popunjenost vektora α možemo prikazati kao $E^T \alpha = \mathbf{0}$. Nakon što odredimo tu rijetku strukturu dalje računamo koeficijente α'_i rješavajući problem najmanjih kvadrata s ograničenjem da je rješenje rijetko popunjeno:

$$\|X - Z_r D_\alpha \mathbb{V}_{r,m}\|_F^2 \rightarrow \min_{\alpha} \text{ s ograničenjem } E^T \alpha = \mathbf{0}. \quad (5.4.10)$$

Ovaj korak zovemo tzv. *poliranje*.

Pogledajmo sada malo detaljnije problem najmanjih kvadrata koji definira koeficijente α'_j u (5.4.2). Pretpostavimo da smo odabrali l modova i posložili ih tako da su vodeći, $\zeta_j = j$, $j = 1, \dots, l$ i potražimo aproksimaciju uzorka x kao u (5.4.2). Neka je par (λ_j, z_j) Ritzov par dobiven DMD algoritmom (algoritam 1) i $Z_l \equiv (z_1 \dots z_l) = U_r \tilde{W}_l$ gdje je $\tilde{W}_l$ neka $r \times l$ matrica, $l \leq k \leq \min(n, m)$. Želimo za dane (λ_j, z_j) i nenegativne težine w_i odrediti α'_j tako da

$$\sum_{i=1}^m w_i^2 \|x_i - \sum_{j=1}^l z_j \alpha'_j \lambda_j^{i-1}\|_2^2 \rightarrow \min_{\alpha'_j}. \quad (5.4.11)$$

Težine $w_i > 0$ nam govore koji uzorci su važniji. Prikažimo sada matrično (5.4.11)

$$\alpha = \begin{pmatrix} \alpha'_1 \\ \alpha'_2 \\ \vdots \\ \alpha'_l \end{pmatrix}, \Delta_\alpha = \begin{pmatrix} \alpha'_1 & 0 & \dots & 0 \\ 0 & \alpha'_2 & \dots & 0 \\ \vdots & \dots & \ddots & 0 \\ 0 & \dots & 0 & \alpha'_l \end{pmatrix}, \Lambda_l = \mathbb{V}_{l,m}(:, i) = \begin{pmatrix} \lambda_1^{i-1} \\ \lambda_2^{i-1} \\ \vdots \\ \lambda_l^{i-1} \end{pmatrix}, \Delta_{\Lambda_i} = \begin{pmatrix} \lambda_1^{i-1} & 0 & \dots & 0 \\ 0 & \lambda_2^{i-1} & \dots & 0 \\ \vdots & \dots & \ddots & 0 \\ 0 & \dots & 0 & \lambda_l^{i-1} \end{pmatrix} = \Lambda^{i-1}. \quad (5.4.12)$$

Sada možemo (5.4.11) prikazati kao funkciju od α

$$\Omega^2(\alpha) \equiv \|[X - Z_l \Delta_\alpha (\Lambda_1 \ \Lambda_2 \ \dots \ \Lambda_m)] \mathcal{W}\|_F^2 \rightarrow \min \quad (5.4.13)$$

gdje je $\mathcal{W} = \text{diag}(w_i)_{i=1}^m$ a $Z_l = (z_{\zeta_1}, \dots, z_{\zeta_l})$. Iskazat ćemo teorem koji nam daje rješenje α minimizacijskog problema (5.4.8).

Teorem 5.4.1. *Neka su matrice $Z_l, \mathbb{V}_{l,m}, \mathcal{W}$ kao u (5.4.13) i neka je $\mathbf{e} = (1 \ \dots \ 1)^T$, jedinstveno rješenje α problema najmanjih kvadrata (5.4.8) je dano kao*

$$\alpha = [(Z_l^* Z_l) \circ (\overline{\mathbb{V}_{l,m} \mathcal{W}^2 \mathbb{V}_{l,m}^*})]^{-1} [(\overline{\mathbb{V}_{l,m} \mathcal{W}} \circ (Z_l^* X \mathcal{W})) \mathbf{e}] \quad (5.4.14)$$

gdje je $\circ$ Hadamardov produkt matrice². Za dokaz teorema pogledati [3].

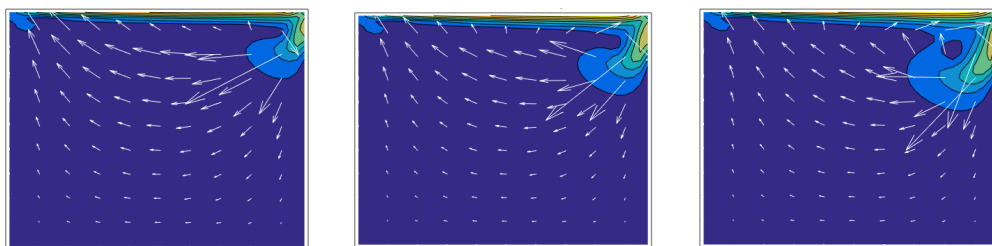
Primjer 5.4.2. *Pogledajmo sada inkomresibilnu Navier-Stokesovu jednadžbu koja opisuje 2D protok kroz šupljinu sa pomičnim vrhom (cavity flow). Inkomresibilna Navier-Stokesova jednadžba je dana kao*

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} &= -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} \\ \nabla \cdot \mathbf{u} &= 0; \end{aligned}$$

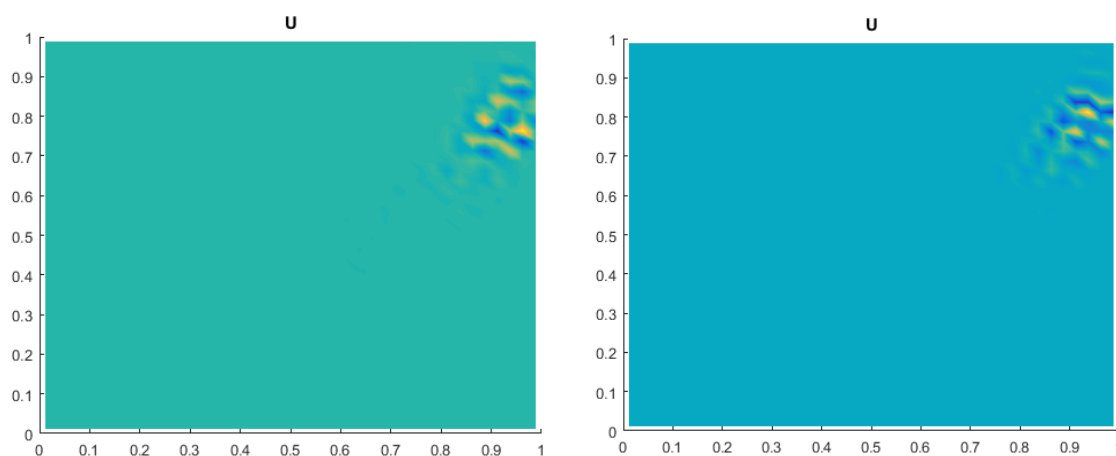
Re je Reynoldsov broj, u našem slučaju $Re = 1000$. Jednadžba je numerički riješena jednostavnom metodom konačnih diferencija.³ Metoda generira vrijednost brzine $\mathbf{u} = (u, v)$, gdje je u vrijednost brzine u smjeru x -osi a v vrijednost brzine u smjeru y -osi. Koristili smo mrežu 100×100 te vremenski korak diskretizacije $dt = 0.01$. Podatke smo skupljali za 400 vremenskih koraka i tako smo za svaki vremenski korak dobili dvije matrice U i V dimenzija 100×100 . Matrice U i V smo zatim vektorizirali u vektor stupac dimenzije 10000×1 za svako vrijeme t , te smo time kao matrice podataka koju smo provlačili kroz algoritam, dobili nove matrice $U = (u_1, \dots, u_{400})$ i $V = (v_1, \dots, v_{400})$ gdje su u_i i v_i vektori dimenzija 10000×1 .

² $C = A \circ B$; $c_{ij} = a_{ij} b_{ij}$

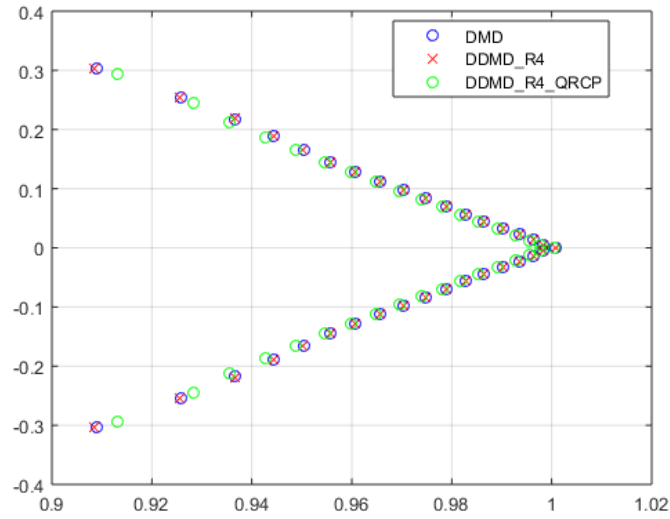
³Kod korišten za generiranje podataka korištenih u ovom primjeru se može pronaći na <https://github.com/mathworks/2D-Lid-Driven-Cavity-Flow-Incompressible-Navier-Stokes-Solver>



Slika 5.6: Radi lakše intuicije na slici je prikazano gibanje fluida u nasumično odabranim trenucima.

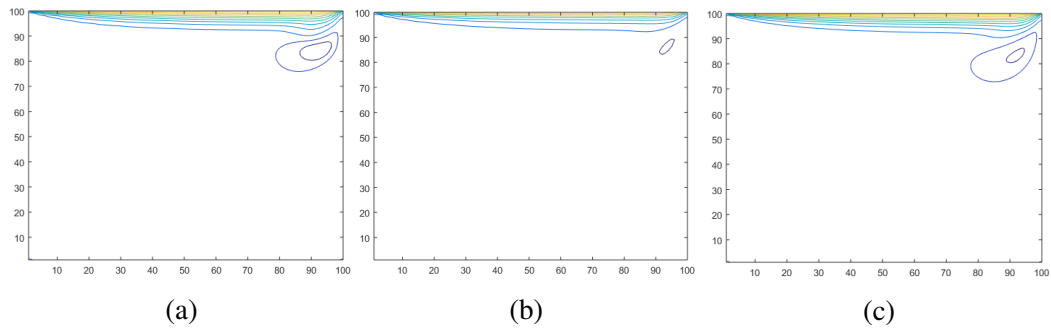


Slika 5.7: Na gornjim slikama prikazana su prva dva moda matrice U dobivena DMD algoritmom.

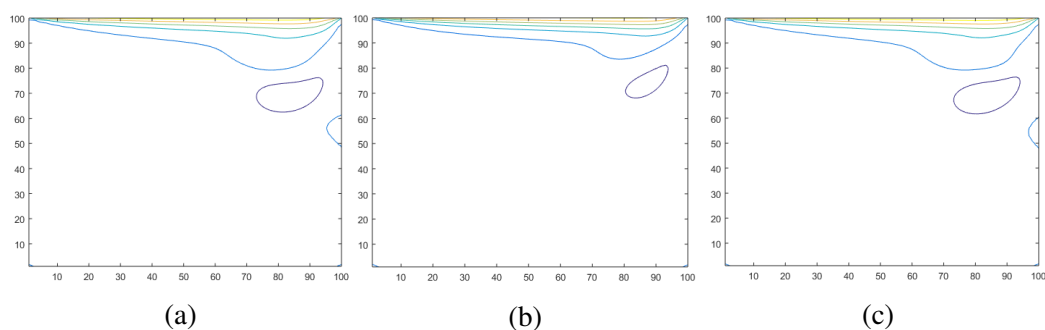


Slika 5.8: Na slici su prikazane svojstvene vrijednosti matrice U dobivene DMD, DDMD_R4 i DDMD_R4_QRCP algoritmom. Kao toleranciju za parametar redukcije dimenzije koristili smo $\epsilon = n\varepsilon$, gdje je $\varepsilon = 1e - 12$. Parametar redukcije dimenzije za sva tri algoritma je $r = 35$.

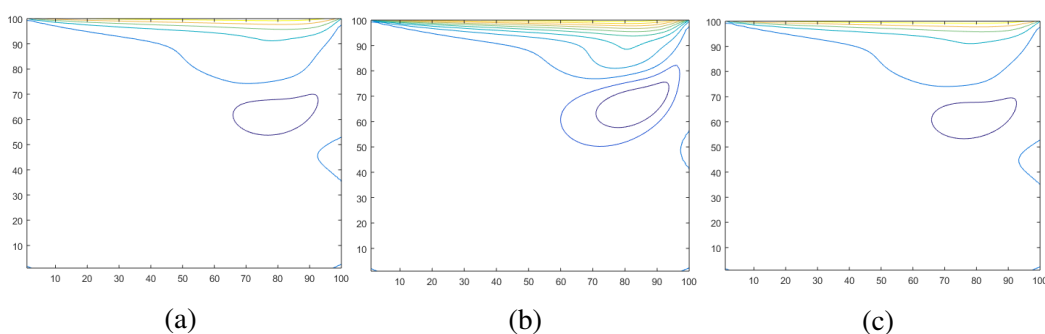
Pogledajmo sada rekonstrukciju nasumično odabranih uzoraka x_i matrice U pomoću DMDS metode. Za parametar l koji određuje s koliko modova želimo aproksimirati dinamiku birali smo l Ritzovih vektora s najmanjim rezidualom kao što je opisano u poglavlju 5.2.



Slika 5.9: Na gornjim slikama prikazan je uzorak u_1 matrice U te njegova rekonstrukcija. Na slici (a) je prikazan egzaktni uzorak u_1 , na slici (b) je prikazana rekonstrukcija uzorka gdje smo za l uzeli $l = 5$, na slici (c) je prikazana rekonstrukcija gdje smo za l uzeli $l = 10$.



Slika 5.10: Na gornjim slikama prikazan je uzorak u_{131} matrice U te njegova rekonstrukcija. Na slici (a) je prikazan egzaktni uzorak u_{131} , na slici (b) je prikazana rekonstrukcija uzorka gdje smo za l uzeli $l = 5$, na slici (c) je prikazana rekonstrukcija gdje smo za l uzeli $l = 10$.



Slika 5.11: Na gornjim slikama prikazan je uzorak u_{220} matrice U te njegova rekonstrukcija. Na slici (a) je prikazan egzaktni uzorak u_{220} , na slici (b) je prikazana rekonstrukcija uzorka gdje smo za l uzeli $l = 5$, na slici (c) je prikazana rekonstrukcija gdje smo za l uzeli $l = 10$.

Bibliografija

- [1] *i-Lids dataset for AVSS 2007*, 2007, http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html.
- [2] P. Businger i G. Golub, *Linear least squares solutions by householder transformations*, Numerische Mathematik **7** (1965), 269–276.
- [3] Zlatko Drmac, Igor Mezic i Ryan Mohr, *On least squares problems with certain Vandermonde-Khatri-Rao structure with applications to DMD*, ArXiv **abs/1811.12562** (2018).
- [4] Z. Drmač, *Dynamic Mode Decomposition - A Numerical Linear Algebra Perspective*, str. 161–194, veljača 2020, ISBN 978-3-030-35712-2.
- [5] Z. Drmač, I. Mezić i R. Mohr, *Data driven modal decompositions: Analysis and Enhancements*, SIAM Journal on Scientific Computing **40** (2017).
- [6] Jacob Grosek i J. Nathan Kutz, *Dynamic Mode Decomposition for Real-Time Background/Foreground Separation in Video*, CoRR **abs/1404.7592** (2014), <http://arxiv.org/abs/1404.7592>.
- [7] J. Kutz, Steven Brunton, Bingni Brunton i Joshua Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*, str. 1–70, studeni 2016, ISBN 978-1-611974-49-2.
- [8] I. Mezić, Z. Drmač, N. Črnjarić Žic, S. Maćešić, M. Fonoberova, R. Mohr, A. Avila i I. Manojlović, *A Koopman Operator-Based Prediction Algorithm and its Applications to COVID-19 Pandemic Supplementary Material*, Technical report (2020).
- [9] A. Perera N. Cuntoor C. Chen J. T. Lee S. Mukherjee J.K. Aggarwal H. Lee L. Davis E. Swears X. Wang Q. Ji K. Reddy M. Shah C. Vondrick H. Pirsiavash D. Ramanan J. Yuen A. Torralba B. Song A. Fong A. Roy Chowdhury S. Oh, A. Hoogs i M. Desai, *A Large-scale Benchmark Dataset for Event Recognition in Surveillance Video*, 2011.

- [10] M. Starčević, *Dinamički sustavi, Skripta*, PMF Matematički odsjek (2017).

Sažetak

U ovom radu opisujemo DMD metodu kao alat za numeričku analizu nelinearnih dinamičkih sustava. DMD metoda je metoda trenirana podacima i nije potrebno poznavati jednadžbu gibanja promatranog sustava, zbog tih poželjnih svojstava danas pronalazi primjenu u raznim područjima. Analiziramo matematičku pozadinu i vidimo da DMD metoda izdvaja važne strukture iz visoko dimenzionalnog dinamičkog sustava na prostorno i vremensko ovisne podatke. Te strukture su dane u svojstvenim vektorima operatora A koji opisuje promatranu dinamiku sustava. Operator A je općenito nepoznat, no pomoću prikupljenih uzoraka sustava, DMD određuje linearnu aproksimaciju operatora A , matricu S_r i traži svojstvenu dekompoziciju te matrice. To postiže koristeći SVD dekompoziciju matrice i Rayleigh-Ritzovu metodu za računanje svojstvene dekompozicije.

Pokazujemo kako pomoću DMD algoritma rekonstruirati dinamički sustav. Zatim definiramo Koopmanove operatore i opisujemo djelovanje Koopmanovog operatora na funkcije stanja sustava. Dajemo kratak opis svojstvene dekompozicije Koopmanovih operatora te povezanost s DMD metodom.

Pokazujemo primjenu DMD metode u procesiranju videa, tj. u odvajanju pozadine videa od prvog plana videa. Metodu testiramo na dvije baze, te dobivene rezultate uspoređujemo s RPCA metodom koja je trenutno najrasprostranjenija u tom području.

U zadnjem poglavlju dajemo varijacije DMD algoritma. Pokazujemo kako odabrati parametar redukcije dimenzije, kako poboljšati Rayleigh-Ritzove parove dobivene DMD algoritmom i tako opisujemo još dva nova algoritma: DDMD_R4 i DDMD_R4.C. Zatim, pokazujemo kako aproksimirati operator A ako želimo da nam prostor u kojem tražimo aproksimaciju razapinju odabrani uzorci stanja. Vidimo da tražene rezultate možemo postići ako umjesto SVD dekompozicije matrice koristimo QR dekompoziciju s pivotiranjem, stoga dajemo opis još jedne metode, DDMD_R4_QRCP. Zatim pokazujemo još jedan način kako, koristeći DMD metodu, možemo rekonstruirati promatranu dinamiku sustava, tj. kako rekonstruirati željene uzorke sustava (DMDSP). Naposljetku, opisane metode primijenjujemo na dinamičkom sustavu koji opisuje Navier-Stokesovu jednadžbu.

Summary

In this paper we study the DMD method as a tool for numerical analysis of nonlinear dynamical systems. The DMD method is data driven and equation free method, hence, with many applications in many fields of engineering. We analyze the mathematical background and show that DMD method extracts important structures from a high dimensional dynamical system – the so called coherent spatio-temporal structures. These structures are given in the eigenvectors of an operator A which describes given dynamical system. The operator A is mostly unknown, but, based on given measurements (observables) of the system, the DMD provides a linear approximation of the operator A , the matrix S_r and later finds its eigendecomposition. DMD is based on the SVD of the matrix of the snapshots and Rayleigh-Ritz method for approximating the eigendecomposition.

We show how to reconstruct dynamical system using DMD algorithm. We define Koopman operator and described its action on the observables. We briefly describe eigendecomposition of the Koopman operator and show its connection to the DMD.

We show DMD application in video processing, that is, we show how to separate background and foreground of a video. We test DMD method on two datasets and compare given results with RPCA method, which is most commonly used nowadays for that purpose.

In the last section we give variations of the DMD algorithm. We present how to determine parameter of dimension reduction, we analyze how to improve Rayleigh-Ritz pairs and describe two new algorithms: DDMD_R4 and DDMD_R4_C. We show how to approximate the operator A if we want for the approximation to reside in the span of selected snapshots, and we see we can apply the QR decomposition with column pivoting instead of the SVD, and in that way we obtain another method, DDMD_R4_QRCP. We also show another way of reconstructing dynamical system in application using the DMD method, that is, how to reconstruct wanted snapshots (DMDSP). As a final example, we apply the presented algorithms on a dynamical system given by the Navier-Stokes equation.

Životopis

Ivana Boras rođena je 28.4.1996. u Splitu, Hrvatska. Odrasla je u mjestu Vitina, Ljubuški (BiH) gdje nakon završene Osnovne škole Tin Ujević pohađa Gimnaziju Ljubuški. Školovanje nastavlja na Fakultetu prirodoslovno-matematičkih i odgojnih znanosti Sveučilišta u Mostaru, gdje završava preddiplomski studij Matematike i stječe titulu: Prvostupnica matematike. Nakon toga upisuje diplomski studij Primijenjene matematike na Prirodoslovno-matematičkom fakultetu Sveučilišta u Zagrebu.