

Određivanje dinamike fizikalnih sustava pomoću strojnog učenja

Vinko Dragušica*

Fizički odsjek, Prirodoslovno-matematički fakultet, Bijenička 32, Zagreb.

Mentor: izv.prof.dr.sc. Davor Horvatić

(Datum: 21. siječnja 2023.)

U radu se opisuje PDE-FIND (PDE Functional Identification of Nonlinear Dynamics) algoritam koji je dodatno modificiran tako da može indentificirati diferencijalne jednadžbe oblika $\ddot{u} = F(u, \dot{u}, u_x, \dots, \mathbf{r}, t, \mu)$ iz podataka o dinamici sustava. Pokazuje se da algoritam uspješno indentificira diferencijalne jednadžbe koje opisuju slobodni pad, harmonijski oscilator, gušeni oscilator, Van der Polov oscilator i valnu jednadžbu. Također se pokazuje da algoritam može indentificirati dinamiku i u slučaju kada se na podatke doda određena razina šuma generirana iz normalne distribucije.

Kodovi i primjeri: <https://github.com/VinkoGitHub/Samostalni-seminar>

I. UVOD

Čovječanstvo je kroz povijest oduvijek pokušavalo opisati prirodne fenomene matematičkim formalizmom. Tako je Newton pronašao tri zakona koji opisuju kompletnu klasičnu fiziku, Maxwell je nakon toga konstruirao jednadžbe koje opisuju klasičnu elektrodinamiku, Schrödinger je pronašao jednadžbu koja opisuje kvantne sustave i tako dalje. Sve te jednadžbe primjeri su diferencijalnih jednadžbi koje zapravo predstavljaju fizikalne zakone prema kojima se neka veličina mijenja u ovisnosti o drugoj veličini (u većini slučajeva vremenu i prostoru).

Kada imamo jednadžbe, cilj je riješiti ih te iz dobivenih rješenja dobiti određena predviđanja. Vjerodostojnost tih predviđanja provjerava se eksperimentom te se teorija zasnovana na danoj diferencijalnoj jednadžbi potvrđuje ili odbacuje.

Povijesno gledano, proces otkrivanja većine fizikalnih zakona odvijao se na gore opisan način, ali s razvojem novih računalnih metoda i algoritama te strojnog učenja moguće je fizikalne zakone otkriti na način koji je obrnut od onog prethodno opisanog. Naime, dosta je slučajeva kod kojih jednadžbe koje opisuju dinamiku sustava nisu poznate, ali imamo vremensku ili prostornu snimku same dinamike. Na primjer u biologiji imamo gibanje raznih struktura za koje ne postoji opis u obliku jednadžbe ali možemo dobiti snimku gibanja tih struktura. Također, u ekonomiji za mnoge pojave nemamo opis u vidu jednadžbi ali ipak imamo podatke o gibanju veličina od interesa.

U ovom seminaru ideja je opisati formalizam pomoću kojeg se može iz ovisnosti neke veličine u o prostoru i vremenu doći do diferencijalne jednadžbe koja bi

opisivala taj sustav i čije rješenje bi bio početni skup podataka $u(\mathbf{r}, t)$.

Metode strojnog i dubokog učenja već su korištene (na primjer [1] i [2]) u ovakvim problemima, ali u ovom radu koriste se rijetki modeli kako bi se na brži i kontroliraniji način moglo indentificirati dinamiku.

II. TEORIJSKA POZADINA

Pretpostavimo da imamo podatke koji predstavljaju dinamiku sustava u prostoru i vremenu u formi funkcije $u(\mathbf{r}, t)$. Na primjer, možemo imati gibanje nekog tijela na opruzi u ovisnosti o vremenu i prostoru ili gibanje valova na površini vode. Iz tih izmjerenih podataka želja nam je dobiti diferencijalnu jednadžbu koja bi za rješenje imala danu funkciju $u(\mathbf{r}, t)$. Metoda koja se koristi u tu svrhu zasnovana je na linearnoj regresiji uz nekoliko modifikacija i radi se po uzoru na [3].

U algoritmu koji je predstavljen u ovom radu koristi se sljedeći model:

$$u_{tt} \equiv \ddot{u} = F(u, u_x, u_{xy}, \dots, \mathbf{r}, t, \mu). \quad (1)$$

Pretpostavlja se da F ovisi o raznim kombinacijama funkcije u , koordinatama, vremenu i raznim drugim parametrima. Skup svih tih funkcija nazivamo funkcije kandidati i koje će funkcije biti uključene u taj skup ovisi o modelu koji konstruiramo. Na primjer, F može ovisiti o prostornim koordinatama x , y , i z , kvadratu funkcije u , umnošku $|u|u_x$ i vremenskoj varijabli t .

A. Numerička obrada podataka

Da bi mogli numerički obrađivati podatke, moramo imati diskretizirani vremenski interval, to jest imamo vremenske trenutke t_1, t_2, \dots, t_m za neki konačan m . Isto vrijedi i za prostornu dimenziju (prostornih točaka ima n). U tim diskretnim vremenskim i prostornim točkama

* vdragusica.phy@pmf.hr

imamo definiranu funkciju $u_{ij} \equiv u(x_i, t_j)$.

Iz diskretiziranih podataka možemo napraviti vektor na način da u vektor-stupac posložimo vrijednosti $u(x_i, t_0)$ za svaki i , zatim dodamo u vektor-stupac vrijednosti $u(x_i, t_1)$ za svaki i i tako dalje za svaki t_j . Konačni vektor nazvat ćemo \mathbf{U} i on izgleda ovako:

$$\mathbf{U} \equiv \begin{bmatrix} u(x_0, t_0) \\ u(x_1, t_0) \\ \vdots \\ u(x_{n-1}, t_m) \\ u(x_n, t_m) \end{bmatrix}. \quad (2)$$

Prvo, treba pronaći drugu derivaciju funkcije $u(\mathbf{r}, t)$ koja se nalazi s lijeve strane jednadžbe (1). Problem derivacije diskretiziranih podataka sa ili bez šuma je problem za sebe i bit će obrađen u jednom od narednih potpoglavlja. Drugi korak je prikaz funkcije F u matricnom formalizmu. Cijeli algoritam prevodimo u diskretiziranu formu na sljedeći način. Pretpostavljamo da je $\mathbf{U}_{tt} = \Theta(\mathbf{U}, \mathbf{Q})\xi$ gdje $\Theta(\mathbf{U}, \mathbf{Q})$ predstavlja skup funkcija kandidata¹, a ξ je vektor koji sadrži koeficijente s kojima pojedina funkcija kandidat ulazi u jednadžbu. Funkcije kandidati se preoblikuju u vektore na isti način kao što je opisano i za vektor \mathbf{U} .

Primjer modela za harmonijski oscilator frekvencije $\omega = 1$ je sljedeći:

$$\begin{bmatrix} | \\ | \\ \mathbf{U}_{tt} \\ | \\ | \end{bmatrix} = \begin{bmatrix} | & | & | & | \\ \mathbf{1} & \mathbf{U} & \mathbf{U}^2 & \mathbf{U}\mathbf{U}_t \\ | & | & | & | \end{bmatrix} \begin{bmatrix} | \\ | \\ \xi \\ | \end{bmatrix} \quad (3)$$

Ukoliko je algoritam dobro konstruiran, trebao bi kao rješenje pružiti vektor

$$\xi = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \end{bmatrix} \quad (4)$$

to jest rješenje bi trebala biti jednadžba $u_{tt} = -u$. Na taj način je iz podataka koji predstavljaju gibanje jednog oscilatora dobivena diferencijalna jednadžba pomoću koje možemo raditi daljnja predviđanja o gibanju sustava ili o raznim drugim ponašanjima sustava.

Do sada je zbog jednostavnosti opisan 1D sustav, ali generalizacija opisane metode na višedimenzionalne sustave je trivijalna. Jedino što treba napraviti je umjesto 1D vektora \mathbf{U} i ξ koristiti višedimenzionalne vektore, tj matrice. Svaki stupac u novoj višedimenzionalnoj matrici

U sadržava podatke o jednoj od dimenzija (na primjer, jedan stupac može označavati dinamiku 2D oscilatora u x smjeru, a drugi stupac dinamiku u y smjeru), a svaki stupac u novoj matrici ξ označava s kojom težinom koja funkcija kandidat ulazi u jednadžbu gibanja dane koordinate (x i y koordinate u primjeru 2D oscilatora).

B. Metode za pronalazak koeficijenata

Kada postavimo neki model, jedino što preostaje je pronaći koeficijente ξ_0 za koje je rješenje takvo da se minimizira određena funkcija gubitka. Koeficijente se može izračunati pomoću metode najmanjih kvadrata²

$$\xi_0 = \underset{\xi}{\operatorname{argmin}} \|\Theta\xi - \mathbf{U}_{tt}\|_2^2. \quad (5)$$

Ova metoda dobro radi kada su podatci čisti, odnosno bez šuma, što je u stvarnosti rijetkost. Kada podatci posjeduju određenu količinu šuma, koeficijenti ξ će biti različiti od onih koji se dobiju kod podataka bez šuma. Naime, većina koeficijenata koji odgovaraju nekoj funkciji kandidatu bit će različiti od 0. To znači da bi za podatke sa šumom dobivena jednadžba trebala imati puno aktivnih članova i time bi mogla biti proizvoljno kompleksna. Način na koji se ovo izbjegne je da se u funkciju gubitka ubaci dodatni član koji odgovara penalizaciji. Drugim riječima, ako je vektor ξ prevelike norme, on neće biti zadovoljavajući. Tada vrijedi da je rješenje danog problema jednako

$$\xi_0 = \underset{\xi}{\operatorname{argmin}} \left\{ \|\Theta\xi - \mathbf{U}_{tt}\|_2^2 + \lambda \|\xi\|_p^2 \right\}. \quad (6)$$

Primjetimo da je norma u drugom članu varijabilna. Kada se koristi 1-norma, metoda se zove Lasso regresija, a kada se koristi 2-norma, metoda se zove Ridge regresija. Obje metode primjeri su takozvanih rijetkih modela. Parametar λ zove se hiperparametar i pomoću njega reguliramo kolika će biti penalizacija modela (što je veći λ , manje će članova biti u ξ). S povećanjem hiperparametra, određeni članovi u ξ postaju 0 kako bi se smanjila norma vektora ξ i dobivamo jednadžbu s manje članova.

Iako bi u principu Lasso algoritam mogao rezultirati rijetkim vektorom ξ , ispostavilo se da postoji i algoritam koji bolje funkcionira. Taj algoritam zove se STRidge³ (*Sequential Threshold Ridge Regression*) i funkcionira tako da više puta napravi Ridge regresiju definiranu hiperparametrom λ na podacima te u svakoj iteraciji odstrani član koji je manji od parametra tolerancije tol . STRidge algoritam koristi se u svim primjerima u ovom radu pri pronalasku jednadžbi gibanja.

¹ $\bar{\mathbf{Q}}$ predstavlja kontrolne varijable i sve dodatne funkcije koje nisu intrinzične sustavu, na primjer tjeranje oscilatora.

² Broj u indeksu norme označava o kojoj se normi radi (ovdje se radi o 2-normi).

³ Više informacija o algoritmu može se pronaći u [4].

Slika 1. Ilustracija PDE-FIND algoritma (preuzeto iz [3])

C. Deriviranje podataka

Kao što je već navedeno, numerička derivacija diskretiziranih podataka je poseban problem. U ovom potpoglavlju ukratko je objašnjen način kako se računa derivacija diskretiziranih podataka sasumom i bezsuma.

Da bismo dobili diferencijalnu jednačbu, trebamo na neki način numerički izračunati prostorne i vremenske derivacije podataka. Ako podatci nemaju suma, onda je zgodno koristiti metodu konačnih razlika (vidi npr. [5]). Ta metoda daje dosta sličan rezultat kao i egzaktna, analitička derivacija podataka u slučaju podataka bezsuma. Formula koja se može koristiti je

$$f'(x) = \frac{f(x+a) - f(x-a)}{2a} + O(a^2) \quad (7)$$

gdje je a neki mali broj a $O(a^2)$ je numerička pogreška kod računanja derivacije.

Ukoliko podatci ipak sadrže određenu količinu suma, mogu se koristiti razne metode. Ipak, metoda koja se pokazala kao najbolja je metoda interpolacije polinoma. Naime, kada tražimo derivaciju u nekoj točki kod podataka sasumom, prvo u toj točki radimo prilagodbu nekog polinoma⁴ reda p (na primjer $p = 3$) i onda deriviramo dobiveni polinom u toj točki i rezultat koji dobijemo smatramo derivacijom početne funkcije u danoj točki. Drugi način je da se prvo funkcija učini glatkom (na primjer konvolucijom s Gaussijanom) i da se onda uzima derivacija pomoću metode konačnih razlika, ali takva metoda

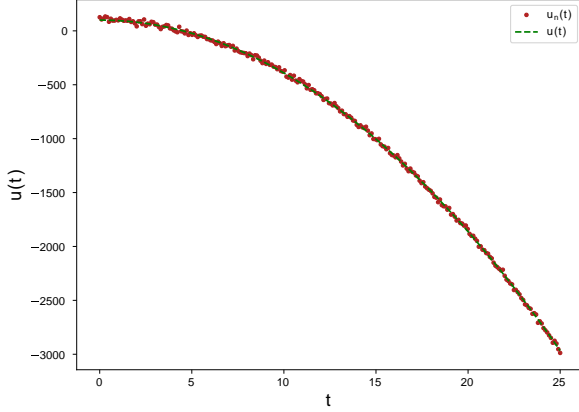
u pravilu daje lošije rezultate od interpolacije polinoma. Dakako, postoje i druge metode koje se mogu koristiti kod ovakvih izračuna.

III. REZULTATI METODE

Opisanim algoritmom uspješno je identificirana diferencijalna jednačba za 5 primjera. Kako su za sve korištene primjere korištene poznate diferencijalne jednačbe i njihova rješenja, prvo su u kodu umjetno stvoreni podatci koji predstavljaju dinamiku danih sustava. Zatim je na tim podacima korišten predstavljani algoritam. Promatrani su koeficijenti u dobivenoj jednačbi i njihova pogreška u vidu odstupanja od egzaktnih vrijednosti.

Zatim je na egzaktnu podatke dodana sumacija je raspodjela normalna. Time su simulirani realni podatci koji u pravilu posjeduju određenu količinu suma. Postupak identifikacije diferencijalne jednačbe sa sumom ponovljen je u 10 iteracija, svaki put sa različitim normalnim sumom (ali sa istim parametrima μ i σ normalne raspodjele). Promatrana je pogreška parametara u svakoj od iteracija i iz tih podataka dobivena je srednja vrijednost i devijacija pogreške za svaki parametar. Također, može se primjetiti da se u nekoj od iteracija pojavljuju dodatni članovi u jednačbama, ali kako se za krajnju jednačbu uzima srednja vrijednost koeficijenata dobivenih u svim iteracijama (tu jednačbu nazivamo srednja jednačba), na kraju se za te članove dobiju značajno manje vrijednosti nego za druge pa ih zapravo možemo zanemariti. Ti članovi u seminaru su istaknuti crvenom bojom. Algoritam je i u slučaju sa sumom također identificirao članove diferencijalne jednačbe i time je potvrđena robustnost algoritma i njegova sposobnost rada s realnim podacima.

⁴ U ovom algoritmu se koriste Čebševljevi polinomi, iako se u principu mogu koristiti bilo koji drugi.



Slika 2. Podatci za slobodni pad (sa šumom)

U priloženom kodu mogu se vidjeti razni parametri koji radi jednostavnosti nisu navedeni u seminaru kao i skup funkcija kandidata koji je korišten za svaki pojedini primjer.

A. Slobodni pad

Algoritam je prvo iskušan na najjednostavnijem primjeru, na slobodnom padu točkastog tijela u konstantnom gravitacijskom polju. Jednadžba takvog gibanja je poznata i glasi⁵

$$\ddot{u} = -g \quad (8)$$

gdje je $g = 9.81$. Algoritam je u slučaju čistog signala u potpunosti točno identificirao diferencijalnu jednadžbu i greška u koeficijentima je stoga jednaka 0%.

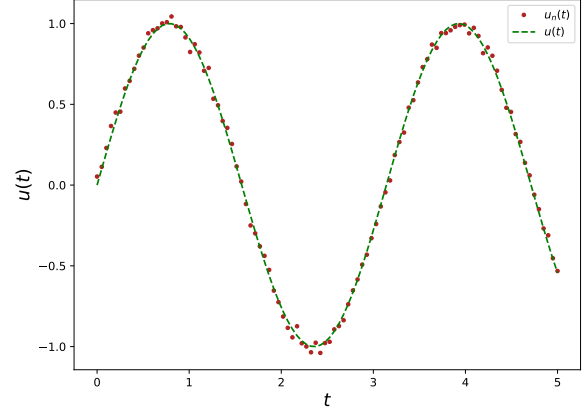
Kada je na čiste podatke dodan šum ($\mu_N = 0, \sigma_N = 15$) kao što je prethodno opisano, dobije se sljedeća srednja jednadžba

$$\ddot{u} = -9.795279 \quad (9)$$

što u odnosu na egzaktnu jednadžbu predstavlja pogrešku od $(1.0 \pm 0.5)\%$. Navedeni primjer može se vidjeti na slici 2.

B. Klasični harmonijski oscilator

Sljedeći primjer je harmonijski oscilator. Ovaj primjer ima dovoljno jednostavnu dinamiku da bude egzaktno



Slika 3. Podatci za harmonijski oscilator (sa šumom)

rješiv a s druge strane je dovoljno kompleksan da bude netrivialan primjer za PDE-FIND algoritam. Jednadžba harmonijskog oscilatora glasi

$$\ddot{u} = -\omega^2 u \quad (10)$$

i u primjeru koji je korišten u kodu (slika 3) odabrana je frekvencija $\omega = 2$. Na čistim podatcima dobivena je jednadžba

$$\ddot{u} = -3.99687u \quad (11)$$

što implicira pogrešku od 0.08% u koeficijentima. Kada se uvede šum ($\mu_N = 0, \sigma_N = 0.03$) kao na slici 3, dobije se srednja jednadžba

$$\ddot{u} = -3.879961u \quad (12)$$

te slijedi da je u tom slučaju pogreška u koeficijentu jednaka $(3.0 \pm 1.6)\%$.

C. Gušeni harmonijski oscilator

Još jedan zanimljiv primjer je harmonijski oscilator s gušenjem. Primjer je sličan prethodnom, ali je uvedeno gušenje koje eksponencijalno smanjuje amplitudu oscilatora u vremenu (slika 4).

U jednadžbi gušenog oscilatora

$$\ddot{u} = -2\zeta\dot{u} - \omega^2 u \quad (13)$$

radi jednostavnosti je odabrano $\omega = 1$ i $\zeta = 0.1$. Kada se obrade podatci bez šuma, dobije se jednadžba

$$\ddot{u} = -0.200179\dot{u} - 0.999128u \quad (14)$$

dok se sa šumom ($\mu_n = 0, \sigma_N = 0.05$) dobiva srednja jednadžba

$$\ddot{u} = -0.188519\dot{u} - 0.982673u. \quad (15)$$

⁵ Zbog jednostavnosti, u svim primjerima koriste se jednadžbe bez jedinica.

Tablica I. Diferencijalne jednadžbe dobivene za primjer Van der Polovog oscilatora bez šuma

μ	PDE-FIND			Max. greška
0.1	$\ddot{u} = 0.099726\dot{u}$	$0.999290u$	$0.099848u^2\dot{u}$	0.27%
2	$\ddot{u} = 1.941602\dot{u}$	$0.972596u$	$1.942509u^2\dot{u}$	2.92%
5	$\ddot{u} = 4.926123\dot{u}$	$0.976853u$	$4.905555u^2\dot{u}$	2.31%

Slijedi da je pogreška u slučaju bez šuma za oba koeficijenta jednaka 0.09%, a u slučaju kada je dodan šum pogreška za \dot{u} iznosi $(6 \pm 2)\%$ dok je pogreška za u jednaka $(1.7 \pm 0.7)\%$.

D. Van der Polov oscilator

Još jedan tip oscilatora koji je korišten kao primjer je Van der Polov oscilator. Taj sustav je zanimljiv jer posjeduje član koji je nelinearan. Jednadžba Van der Polovog oscilatora glasi

$$\ddot{u} = \mu(1 - u^2)\dot{u} - \omega^2 u \quad (16)$$

i opet je zbog jednostavnosti odabrano $\omega = 1$. Za μ su uzete tri različite vrijednosti i ujedno su postavljeni različiti početni uvjeti kako bi se obuhvatila malo bogatija dinamika. Rezultati su prikazani u tablicama I i II dok slika 5 prikazuje primjer dinamike Van der Polovog oscilatora za $\mu = 2$.

E. Valna jednadžba

Zadnji primjer je valna jednadžba prvog reda. Ta je jednadžba slična valnoj jednadžbi ali njezina rješenja

mogu se gibati samo u jednom smjeru pa se ona još naziva i *One-way* valna jednadžba.

Jednadžba glasi

$$\dot{u} = cu' \quad (17)$$

gdje je c brzina gibanja vala i rješenja jednadžbe su ravni valovi (slika 6) koji se gibaju u jednom smjeru. Brzina c odabrana je da bude $c = -2$ kako bi se valovi gibali udesno. Ovaj primjer razlikuje se od prethodnih jer se ovdje ne radi o dinamici točkaste čestice nego jednodimenzionalnog polja. Također, radi se o jednadžbi koja je prvog reda u vremenu pa se ne koristi PDE-FIND algoritam iz ovog rada nego originalni PDE-FIND iz [3].

Primjer je provjeren na podacima bez šuma i dobivena je jednadžba

$$\dot{u} = -2.002414u' \quad (18)$$

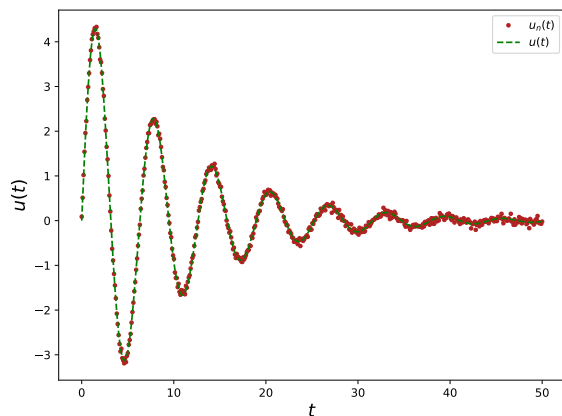
dok je za podatke sa šumom ($\mu_n = 0, \sigma_N = 0.1$) dobivena srednja jednadžba

$$\dot{u} = -1.969783u'. \quad (19)$$

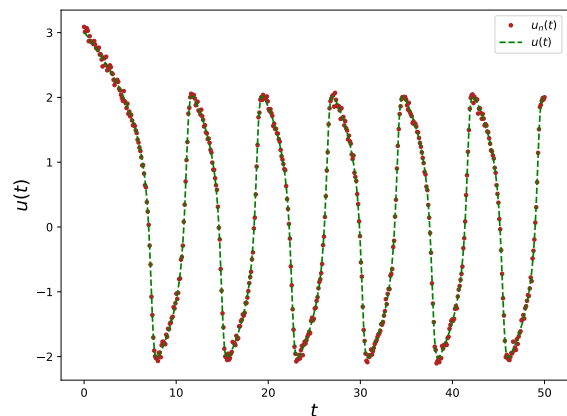
Greška je u prvom slučaju jednaka 0.1% dok u drugom slučaju iznosi $(1.5 \pm 0.3)\%$.

IV. ZAKLJUČAK

U radu je predstavljen algoritam sličan algoritmu PDE-FIND iz [3], ali modificiran tako da identificira diferencijalne jednadžbe drugog reda. Objašnjena je teorijska pozadina na kojoj takav algoritam počiva i osnovni dijelovi algoritma. Na kraju su predstavljene diferencijalne jednadžbe koje je PDE-FIND identificirao



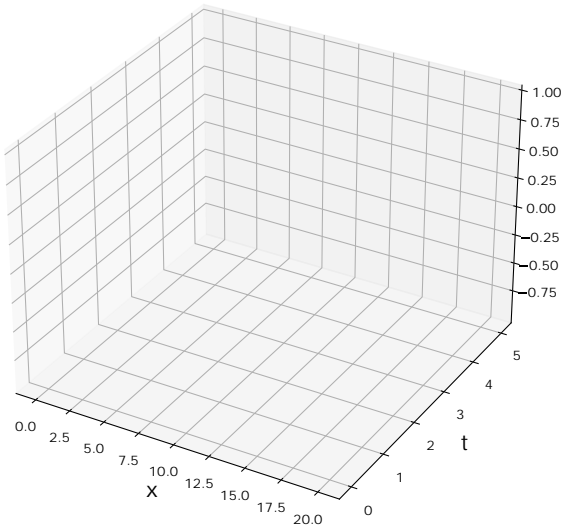
Slika 4. Podatci za gušeni oscilator (sa šumom)



Slika 5. Podatci za Van der Polov oscilator (sa šumom)

Tablica II. Diferencijalne jednadžbe dobivene za primjer Van der Polovog oscilatora sa šumom

μ	PDE-FIND srednja jednadžba	Parametri μ_N i σ_N	Pogreška uz \dot{u}	Pogreška uz u	Pogreška uz $u^2\dot{u}$
0.1	$\ddot{u} = 0.016188 + 0.076330\dot{u} - 0.983806u$ $+ 0.006127u^2 - 0.002214u\dot{u} - 0.088699u^2\dot{u}$	$\mu_N = 0, \sigma_N = 0.05$	(26 12)%	(11 5)%	(1.6 1.1)%
2	$\ddot{u} = 0.033608 + 1.412393\dot{u} - 0.928338u$ $+ 0.023849u^2 - 0.004304u\dot{u} - 1.357691u^2\dot{u}$	$\mu_N = 0, \sigma_N = 0.05$	(29 3)%	(32 3)%	(7 4)%
5	$\ddot{u} = 0.010158 + 3.237894\dot{u} - 1.027299u$ $+ 0.002301u^2 - 0.005109u\dot{u} - 3.034661u^2\dot{u}$	$\mu_N = 0, \sigma_N = 0.05$	(35 3)%	(39 3)%	(7 4)%



Slika 6. Podatci za valnu jednadžbu

za slobodni pad, harmonijski oscilator, gušeni oscilator, Van der Polov oscilator i valnu jednadžbu. Pokazano je da algoritam radi u svim ovim slučajevima čak i sa dodanim normalnim šumom. Time je dokazana stabilnost algoritma i njegova sposobnost za rad na realnim podacima sa šumom. Također, može se primjetiti da je vrijeme izvršavanja kodova tek nekoliko sekundi što čini algoritam jako brzim i pouzdanim kod rješavanja

problema identifikacije dinamike.

Kod nekih primjera (na primjer Van der Polov oscilator) vidljiva je veća greška kada se na čiste podatke nadoda normalni šum. Razlog tome je što u određenim dijelovima grafa postoje veliki skokovi u funkcionalnim vrijednostima i tu su razmaci među točkama veliki (slika 5) pa su neki dijelovi podataka efektivno rijetko raspoređeni u vremenu. Taj problem može se riješiti tako da uzmemo veću vremensku rezoluciju, odnosno više točaka u vremenu ili tako da snimamo sustav dulje vremena.

Također, značajan problem algoritma je veliki broj parametara koje korisnik samostalno mora odabirati. Iako se određeni parametri mogu eliminirati na način da se odabiru automatski Pareto analizom, ipak je broj preostalih parametara značajan i nije očito koju vrijednost odabrati za koji parametar. Ipak, većina primjera u ovom radu radi za veliki raspon vrijednosti parametara pa je taj problem u ovom slučaju zaobiden, ali treba imati na umu da će algoritam za neke primjere raditi dobro samo za uski raspon parametara koje korisnik treba pronaći.

Iako ima niz problema, PDE-FIND je algoritam koji je brz, pouzdan i efikasan u traženju diferencijalnih jednadžbi koje opisuju razne sustave s potencijalnom upotrebom u fizici, kemiji, medicini, biologiji, ekonomiji i mnogim drugim poljima.

-
- [1] R. González-García, R. Rico-Martínez, and I. Kevrekidis, Identification of distributed parameter systems: A neural net based approach, *Computers and Chemical Engineering* **22**, S965 (1998).
- [2] H. U. Voss, P. Kolodner, M. Abel, and J. Kurths, Amplitude equations from spatiotemporal binary-fluid convection data, *Phys. Rev. Lett.* **83**, 3422 (1999).
- [3] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, Data-driven discovery of partial differential equations, *Science Advances* **3**, e1602614 (2017), <https://www.science.org/doi/pdf/10.1126/sciadv.1602614>.
- [4] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proceedings of the National Academy of Sciences* **113**, 3932 (2016), <https://www.pnas.org/doi/pdf/10.1073/pnas.1517384113>.
- [5] R. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*, Other Titles in Applied Mathematics (Society for Industrial and Applied Mathematics, 2007).