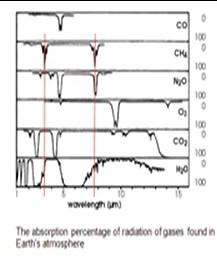
**Primjena z-transformacije, linearne algebre i genetskog algoritma na procjenjivanje svjetskih antropogenih emisija metana**

Autor: Teo Samaržija

SAŽETAK: Među vegetarijancima na internetu dobro je poznata činjenica da su svjetske antropogene emisije metana dostigle vrhunac negdje 1980-ih i da otada opadaju. Međutim, neki misle da je tu riječ o krivoj statističkoj analizi podataka. Ovaj tekst pokušava odgovoriti na to pitanje.

**O metanu kao stakleničkom plinu**

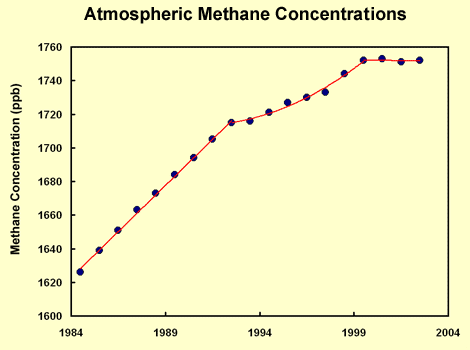
Metan ili CH4 jedan je od glavnih uzročnika globalnog zatopljenja. On ima širi emisijski spektar od CO2-a i na prvi pogled bi se činilo da ima veći utjecaj na globalno zatopljenje od CO2. No, to nije točno iz tri razloga:

1. Metana u atmosferi ima daleko manje nego što ima CO2-a. Koncentracija metana u atmosferi mjeri se u parts per billion (dijela na milijardu), dok se koncentracija CO2-a mjeri u parts per million (dijela na milijun).
2. Iako CH4 ima širi emisijski spektar nego što ima CO2, frekvencije koje on pokriva uglavnom se poklapaju s frekvencijama u spektru vode. Voda je najbrojniji i najjači staklenički plin i prema tome emisijski spektri ostalih stakleničkih plinova nisu važni u onoj mjeri u kojoj se poklapaju sa spektrom vode. CO2, iako ima uži spektar nego što ima CH4, svojim spektrom pokriva mnoge frekvencije koje voda svojim spektrom ne pokriva.
3. Metan ima kratak životni vijek. Vrijeme poluživota metana u atmosferi procjenjuje se na između 9 i 12 godina. Glavni način na koji metan nestaje je da on kemijski reagira s ozonom te se pretvara u ugljikov dioksid i vodu. Negdje ćete možda naći podatak da je prosječno vrijeme života molekule metana u atmosferi oko 17 godina, ali primijetite da vrijeme poluživota i prosječno vrijeme života nisu isto. Prosječno vrijeme života znači zbroj vremena života molekula metana podijeljen s brojem molekula metana, dok je vrijeme poluživota vrijeme potrebno da pola molekula metana nestane. Prosječno vrijeme života jednako je vremenu poluživota podijeljenim s prirodnim logaritmom od dva.

Iz tih razloga računalni modeli uglavnom procjenjuju da metan ima oko 20% utjecaja na globalno zatopljenje, dok ugljikov dioksid ima oko 80%[[1]](#footnote-2).

**Tvrdnja da su naše emisije metana dostigle vrhunac 1980-ih**

Ako ste sudjelovali u raspravama o vegetarijanstvu, vjerojatno ste vidjeli ovaj dijagram ili slične dijagrame:



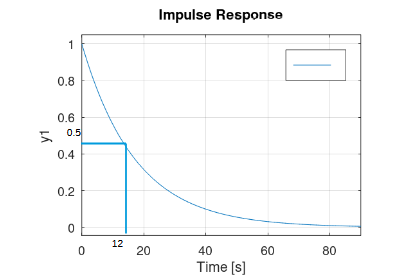
Preuzeto s: <https://skepticalscience.com/images/atmospheric_methane_conc.gif>

Argument ide da naše emisije metana (antropogene emisije metana iz cijelog svijeta) moraju biti malo veće nego prva derivacija (brzina rasta) ove krivulje, te da se s grafa može očitati da su naše emisije metana dostigle svoj vrhunac u 1980-ima i da od tada opadaju. To je teško objasniti ako pretpostavimo da naše emisije metana primarno dolaze od zemnog plina koji mi koristimo (jer mi danas koristimo znatno više zemnog plina nego što smo koristili 1980-ih), ali se lagano može objasniti ako pretpostavimo da većina emisija metana dolazi od krava. Naime, vegetarijanci uglavnom vjeruju da krave koje se hrane travom emitiraju oko 3 puta više metana po litri mlijeka nego krave koje se hrane zrnjem, a danas ima sve manje krava koje se hrane travom, a sve više krava koje se hrane zrnjem. Koliko su te tvrdnje istinite izvan dosega je ovog teksta.

Kada sam taj argument prezentirao korisniku Quore magistru kemije Malcolmu Forsteru, on mi je rekao da se njemu taj dijagram čini kao otprilike ono što bismo očekivali da su naše emisije metana konstantne, i da probam napraviti kompjutersku simulaciju i da ću ubrzo shvatiti zašto. Na prvi pogled mi se ta tvrdnja činila suludom, ali sad mi se više ne čini.

**Kompjuterska simulacija u Octavi**

Na prvi bi se pogled činilo da su emisije metana u atmosferi analogne IT1 sustavima o kojima smo učili na Osnovama automatskog upravljanja, ali je li stvarno tako? Probajmo izvesti prijenosnu funkciju. Sada, na prvi pogled možda nije očito kako izvesti prijenosnu funkciju. Na Osnovama automatskog upravljanja prijenosna funkcija obično se računa iz diferencijalnih jednadžbi, a ovdje nije očito kako postaviti diferencijalne jednadžbe. Međutim, očito je koja je težinska funkcija (odziv na Diracovu delta-funkciju):



A, sjećamo se, za linearne vremenski nepromjenjive sustave (kao što je metan u atmosferi), prijenosna funkcija jednaka je Laplaceovoj transformaciji težinske funckije. Po teoremu eksponencijalnog raspada Laplaceove transformacije:

Prijenosna funckija je:

Primijetite da se ova prijenosna funkcija ne može svesti na prijenosnu funkciju IT1 člana. Prijenosna funkcija IT1 člana je:

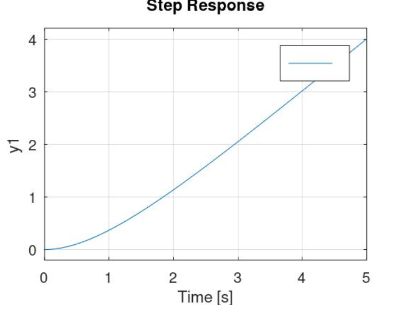
Kako bi sustav opisan prijenosnom funkcijom koju smo upravo izračunali reagirao na konstantne emisije metana? Koja je njegova prijelazna funkcija (odziv na Heavisideovu jediničnu stepenicu)? Pa, pitajmo program Octave, ovim kodom:

1 pkg load control;

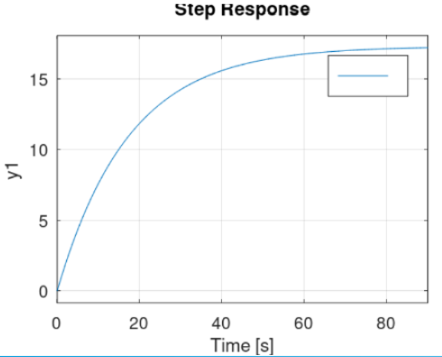
2 s=tf('s');

3 step(1**/**(s**+log**(2)**/**12))

Što očekujemo? Pa, ja sam očekivao ovako nešto (približno linearan rast, i sigurno ne konvergencija):



Međutim, Octave je ispisao ovo:



Problem sada postaje očit: ovaj dijagram izgleda slično kao empirijski mjerena koncentracija metana u atmosferi s vremenom.

Pitao sam se: konvergira li ta krivulja? Ili je ona kao logaritamska krivulja, da raste sve sporije s vremenom, ali da ne doseže nikakvu konačnu vrijednost? Sjećamo se iz Osnova automatskog upravljanja da se takvi problemi (kao i problemi određivanja greške u stacionarnom stanju ovisno o tome kakav regulator koristimo) rješavaju pomoću teorema o konačnoj vrijednosti Laplaceove transformacije. S obzirom na to da je Laplaceova transformacija Heavisideove jedinične stepenice jednaka 1/s, Laplaceova transformacija prijelazne funckije jednaka je:

Dakle, po teoremu konačne vrijednosti Laplaceove transformacije:

Dakle, naše emisije metana iznose približno, prema podacima koje očitamo s grafa:

I približno su konstantne, a ne da su funkcija koja je dostigla vrhunac negdje 1980-ih i od tada da opada.

**Rekonstrukcija naših emisija metana**

Međutim, postavlja se pitanje možemo li biti još precizniji od toga. Možemo li odrediti točnu funckiju kako se naše emisije metana mijenjaju s vremenom? Pa, mislim da je taj problem lakše riješiti u z-domeni nego u s-domeni. Ako se sjećamo iz Signalnih sustava, Laplaceova transformacija pretvara se u z-transformaciju po formuli:

Prema tome bi z-transformacija bila:

Ne znam kako tu formulu srediti dalje, ali čini mi se očitim da vrijedi:

Hajdemo napisati to u programskom jeziku JavaScript (jer je to programski jezik koji ja znam najbolje), zadati mu da izračuna prijelaznu funkciju, i usporediti rezultat s Octavinim:

1 **const** vrijeme\_poluzivota\_metana = 12;

2 **const** step\_funkcija = [];

3 **for** (let i = 0; i < 80; i++)

4 step\_funkcija.push(1);

5 function koncentracija\_metana\_u\_atmosferi(emisije\_metana\_po\_godinama) {

6 **const** koncentracija\_metana\_u\_atmosferi = [];

7 **for** (let i = 0; i < emisije\_metana\_po\_godinama.length; i++) {

8 let zbroj = 0;

9 **for** (let j = 0; j <= i; j++)

10 zbroj += emisije\_metana\_po\_godinama[i - j] \* Math.E \*\*

11 (Math.log(1 / 2) / vrijeme\_poluzivota\_metana \* j);

12 koncentracija\_metana\_u\_atmosferi.push(zbroj);

13 }

14 **return** koncentracija\_metana\_u\_atmosferi;

15 }

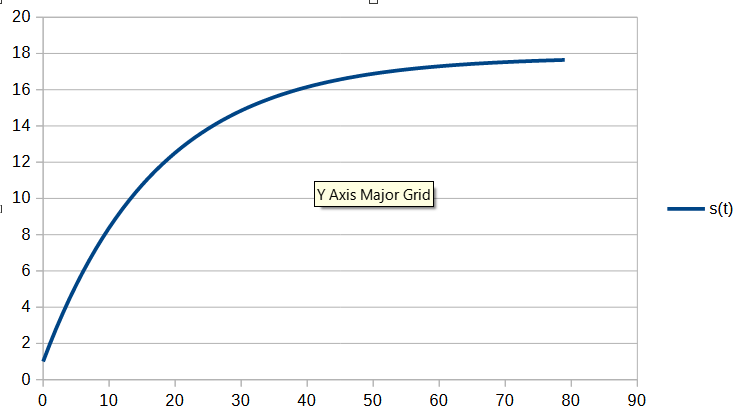
16 **const** prijelazna\_funkcija = koncentracija\_metana\_u\_atmosferi(step\_funkcija);

17 console.log("t\ts(t)");

18 **for** (let i = 0; i < prijelazna\_funkcija.length; i++)

19 console.log(i + "\t" + prijelazna\_funkcija[i]);

Evo što ispisuje:



Vidimo da je to poprilično slično prijelaznoj funkciji koju ispisuje Octave. Onda bismo mogli, recimo, aproksimirati naše emisije metana pomoću Gaussove metode iz kolegija Linearna algebra i popraviti aproksimaciju tehnikom mekanog programiranja (genetski algoritam), ovako:

1 **const** koncentracije\_metana\_ocitane\_s\_grafa = [

2 1625, 1640, 1650, 1665, 1675, 1685, 1695, 1705, 1715, 1720,

3 1725, 1730, 1735, 1737, 1745, 1750, 1755, 1755, 1750, 1755

4 ];

5 **const** krivulja =

6 koncentracije\_metana\_ocitane\_s\_grafa.map((x) => { **return** x - 1620; });

7 **const** vrijeme\_poluzivota\_metana = 12; // Procjene se kreću između 9 i 12.

8 let jedinke = [];

9 **const** koliko\_jedinki\_imamo = 12;

10 function nasumicno\_generiraj\_pokusaj() {

11 **const** niz = [];

12 let trenutni = 2 \* (1760 - 1600) / (vrijeme\_poluzivota\_metana / Math.log(2)) \*

13 Math.random();

14 **while** (niz.length < krivulja.length) {

15 niz.push(trenutni);

16 trenutni += Math.random() \* 3 - 3 / 2;

17 }

18 **return** niz;

19 }

20 function aproksimiraj\_Gaussovom\_metodom() {

21 **const** niz = [];

22 **for** (let i = 0; i < krivulja.length; i++) {

23 let razlika = krivulja[i];

24 **for** (let j = 1; j <= i; j++)

25 razlika -= niz[i - j] \* Math.E \*\*

26 (Math.log(1 / 2) / vrijeme\_poluzivota\_metana \* j);

27 razlika += Math.random() - 1 / 2;

28 niz.push(razlika);

29 }

30 **return** niz;

31 }

32 jedinke.push(aproksimiraj\_Gaussovom\_metodom());

33 **while** (jedinke.length < koliko\_jedinki\_imamo \*\* 2)

34 jedinke.push(nasumicno\_generiraj\_pokusaj());

35 function koncentracija\_metana\_u\_atmosferi(emisije\_metana\_po\_godinama) {

36 **const** koncentracija\_metana\_u\_atmosferi = [];

37 **for** (let i = 0; i < emisije\_metana\_po\_godinama.length; i++) {

38 let zbroj = 0;

39 **for** (let j = 0; j <= i; j++)

40 zbroj += emisije\_metana\_po\_godinama[i - j] \* Math.E \*\*

41 (Math.log(1 / 2) / vrijeme\_poluzivota\_metana \* j);

42 koncentracija\_metana\_u\_atmosferi.push(zbroj);

43 }

44 **return** koncentracija\_metana\_u\_atmosferi;

45 }

46 function izracunaj\_gresku(koncentracije\_metana\_u\_atmosferi) {

47 let zbroj = 0;

48 **if** (krivulja.length != koncentracije\_metana\_u\_atmosferi.length)

49 **throw** "Neki dio programa je promijenio velicinu niza, prekidamo!";

50 **for** (let i = 0; i < krivulja.length; i++)

51 zbroj += (krivulja[i] - koncentracije\_metana\_u\_atmosferi[i]) \*\* 2;

52 **return** zbroj;

53 }

54 function krizaj\_dva\_pokusaja(prvi, drugi) {

55 **if** (prvi.length != drugi.length)

56 **throw** "Neki dio programa je promijenio velicinu niza, prekidamo!";

57 **const** novi\_niz = [];

58 **if** (Math.random() < 1 / 2) {

59 novi\_niz.push((prvi[0] + drugi[0]) / 2);

60 let trenutni = novi\_niz[0];

61 **for** (let i = 1; i < prvi.length; i++)

62 **if** (Math.random() < 1 / 2) {

63 trenutni += prvi[i] - prvi[i - 1];

64 novi\_niz.push(trenutni);

65 } **else** {

66 trenutni += drugi[i] - drugi[i - 1];

67 novi\_niz.push(trenutni);

68 }

69 } **else** {

70 **for** (let i = 0; i < prvi.length; i++)

71 **if** (Math.random() < 1 / 2)

72 novi\_niz.push(prvi[i]);

73 **else**

74 novi\_niz.push(drugi[i]);

75 }

76 **return** novi\_niz;

77 }

78 function nasumice\_izmijeni(pokusaj) {

79 **const** novi\_pokusaj =

80 pokusaj.map((x) => { **return** x + Math.random() \* 3 - 3 / 2; });

81 **return** novi\_pokusaj;

82 }

83 function sortiraj\_jedinke\_i\_zadrzi\_najbolje() {

84 jedinke.sort((prva, druga) => {

85 **return** izracunaj\_gresku(koncentracija\_metana\_u\_atmosferi(prva)) <

86 izracunaj\_gresku(koncentracija\_metana\_u\_atmosferi(druga));

87 });

88 jedinke = jedinke.slice(0, koliko\_jedinki\_imamo);

89 }

90 let brojac = 0;

91 **while** (brojac < 1\_000) {

92 sortiraj\_jedinke\_i\_zadrzi\_najbolje();

93 **for** (let i = 0; i < koliko\_jedinki\_imamo; i++) {

94 **for** (let j = 0; j < koliko\_jedinki\_imamo; j++) {

95 **if** (i == j)

96 **continue**;

97 **if** (Math.random() < 1 / koliko\_jedinki\_imamo)

98 jedinke.push(

99 nasumice\_izmijeni(krizaj\_dva\_pokusaja(jedinke[i], jedinke[j])));

100 **else**

101 jedinke.push(krizaj\_dva\_pokusaja(jedinke[i], jedinke[j]));

102 }

103 jedinke.push(nasumice\_izmijeni(jedinke[0]));

104 jedinke.push(

105 nasumicno\_generiraj\_pokusaj()); // Biološki krajnje nerealistično

106 // (abiogeneza se ne događa danas), ali

107 // izgleda da program daje bolje

108 // rezultate ovako.

109 }

110 **for** (let i = 0; i < 6; i++) // Dakle, neka najbolja jedinka ima 6+1=7 djece sa

111 // svakom drugom jedinkom.

112 **for** (let j = 1; j < koliko\_jedinki\_imamo; j++)

113 jedinke.push(krizaj\_dva\_pokusaja(jedinke[0], jedinke[j]));

114 brojac++;

115 }

116 sortiraj\_jedinke\_i\_zadrzi\_najbolje();

117 **const** izlaz\_simulacije = koncentracija\_metana\_u\_atmosferi(jedinke[0]);

118 console.log(

119 "Godine\tSimulirane emisije metana\tSimulirana koncentracija metana");

120 **for** (let i = 0; i < jedinke[0].length; i++)

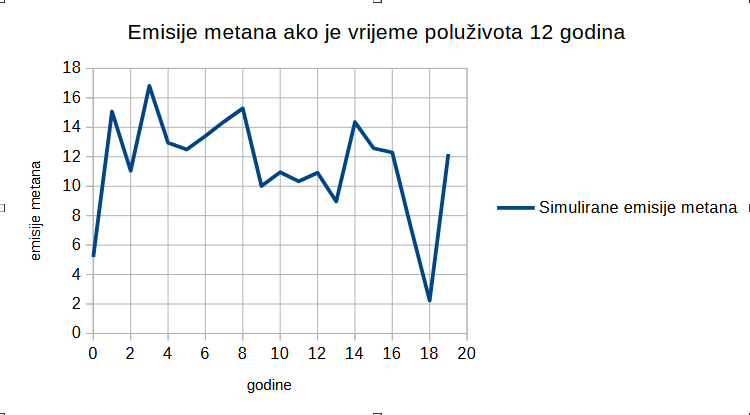
121 console.log(i + "\t" + jedinke[0][i] + "\t" + izlaz\_simulacije[i]);

122 console.log(

123 "Ukupno kvadratno odstupanje simulirane koncentracije metana od mjerenih podataka: " +

124 izracunaj\_gresku(izlaz\_simulacije));

Ako mu se zada da pretpostavi da je vrijeme poluživota metana 12 godina, taj program ispisuje ovo:



Ovdje se vidi blagi trend prema dolje. Međutim, ako mu se zada da je poluživot metana 9 godina, on ispisuje ovo:



I ovdje se ne vidi nikakav trend prema dolje.

Pomalo mi je začuđujuće to što je 9. godina (dakle, 1984+9=1993) u prvom dijagramu dolina, a u drugom dijagramu je vrh. Volio bih da mi neki statističar objasni kako je to moguće.

**Zaključak**

Tvrdnja da su naše emisije metana dostigle svoj vrhunac 1980-ih i od tada brzo opadaju uistinu nije bazirana na dobroj evidenciji, nego na pogrešnoj statističkoj analizi. Ako pretpostavimo da je vrijeme poluživota metana 12 godina, računalna simulacija pokazuje blagi pad emisija metana, ali, ako pretpostavimo da je ono 9 godina, ne pokazuje nikakav pad.

1. Podatak preuzet s uvoda na Wikipedijinu stranicu o metanu: https://en.wikipedia.org/wiki/Methane [↑](#footnote-ref-2)