

Modelarea seriilor de timp cu funcții de baza Fourier

Bene Adrian Cristian

Florișteanu Antonio Pavel

Ognean Naomi Lorena

Index : product\_14

# CUPRINS

[1. Introducere 3](#_Toc118907299)

[2. Descrierea funcției 3](#_Toc118907300)

[3. Descrierea proiectului 4](#_Toc118907301)

[4. Rezultate 5](#_Toc118907302)

[5. Concluzii 6](#_Toc118907303)

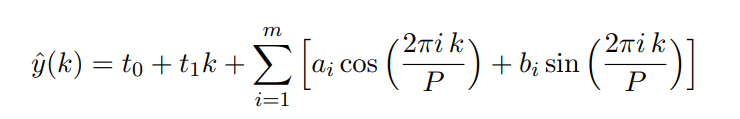
[6. Bibliografie 7](#_Toc118907304)

# Introducere

Documentul prezent susține soluția găsită de noi pentru prima parte a proiectului de la materia “Identificarea Sistemelor’’.

Această parte a proiectului abordează modelarea seriilor de timp cu funcții de bază Fourier. Enunțul problemei prezintă o problemă reala, și anume vânzările unui magazin de instalații și a produselor care se vând mai bine în diferite perioade ale anului.

# Descrierea funcției



*Fig 1. Forma formulei folosite*

Seria conține o componentă liniara, dar și o componenta Fourier neliniara cu un numar variabil de termeni. Componenta liniara este de forma . Fiecare termen conține o funcție *sin* și o funcție *cos* de diverse frecvențe. Seria de timp conține și un vector de parametrii de forma

# 

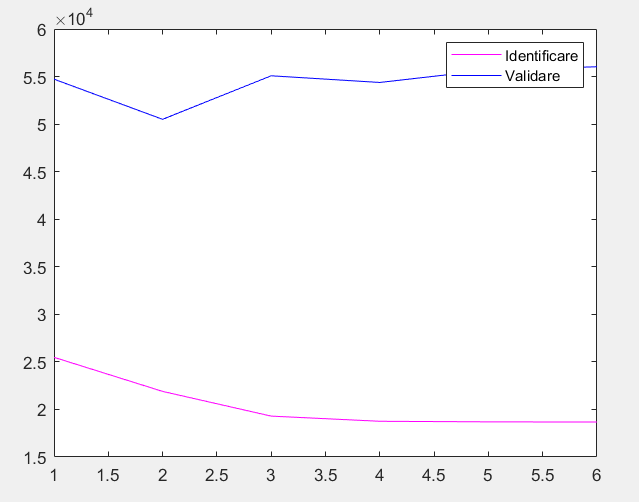
# Descrierea proiectului

În setul de date furnizat există doi vectori de aceeași lungime. Unul care exprimă timpul (un index pentru fiecare lună) iar celălalt vector, y, exprimă cantitatea de produs vândut din luna adiacentă. Setul de date este împărțit în două părți. Prima parte yc(80%) reprezintă datele de identificare iar restul de yval(20%) reprezintă datele de validare.

În codul realizat de noi ca soluție la găsirea unei valori ‘m’ dat modelului constă din aflarea vectorului optim de parametri θ pentru care yc si yval se apropie cel mai mult de y. Prima parte realizează încărcarea și definirea constantelor, apoi a doua parte este pentru împărțirea vectorului y în 80% pentru identificare, iar restul de 20% pentru validare. Ultima parte reprezintă crearea repetat al fi-urilor( matrice care conține seria de componente pentru fiecare timp dat), cât și calculul repetat al vectorului θ(denumit Teta in cod) *yc=fi\_id\*θ* și *yval= fi\_val\*θ* . Vom afisa yc și y\_val pentru fiecare m în comparație cu y inițial, calculul erorilor find E\_id=y-yc, E\_val=y-yval. Astfel la final putem calcula MSE-urile pentru identificare si validare. Calcurile se repeta de m(denumit MAX în cod) ori, pentru a găsi cel mai bun posibil vector optim de parametrii θ.

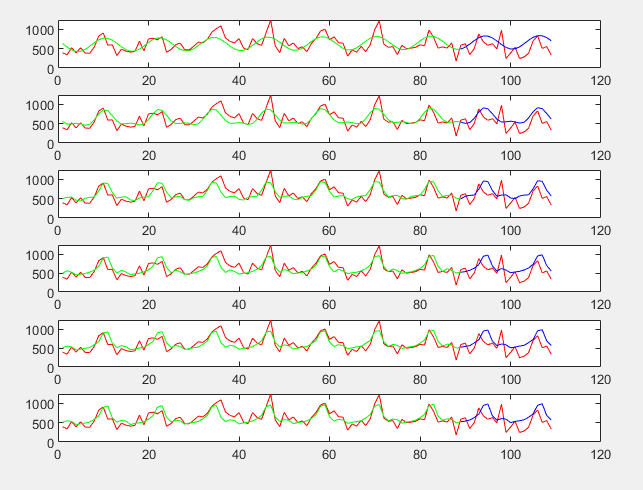
La final se afișează grafic evolutia MSE-urilor in functie de m-ul ales.

# Rezultate



În Figura 2 putem observa evolutia MSE de validare si identificare in functie de m-ul ales intre 1 si 6.

*Fig 2. Evolutia MSE*

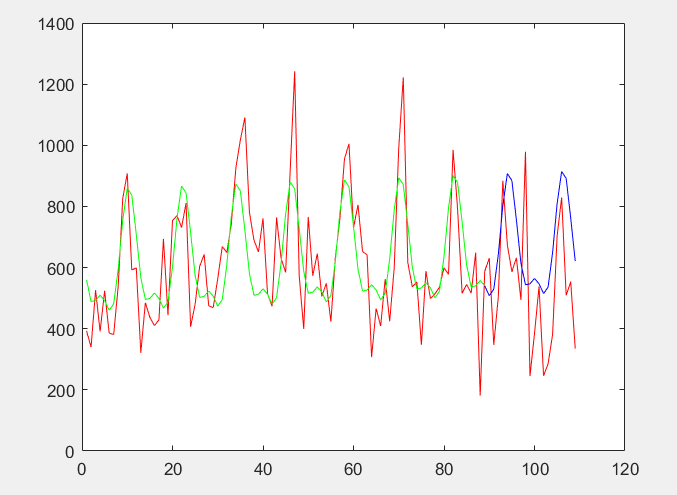


În Figura 3 putem observa comparația dintre y-urile aproximate(reprezentate cu verde si albastru) si datele primite de la mgazin(cu roșu) in funcție de timpul oferit.

*Fig 3. Forma y date, y aproximativ*

# Concluzii

Conform Figurii 2 cel mai bun ‘m’ pentru model este unde MSE-ul de validare este minim, pentru cazul nostru în m=2. Chiar dacă se preferă și MSE identificare să fie minim, nu putem alege un ‘m’ mai mare, deoarece MSE validare va crește.



*Fig 4. Forma y date, y aproximativ pentru m=2*

# Bibliografie

* <https://busoniu.net/teaching/sysid2022/projectro.pdf>
* Matlab

load('product\_14 (3).mat') close all, p=12, MAX=2, MSE\_id=zeros(1,MAX), MSE\_val=zeros(1,MAX);

k=y'; ndata = size(k, 2); procent = 82; newpoints = floor(ndata\*procent/100);

k = k(:); yc = k(1:newpoints); yval=k(length(yc):length(y))

for m=1:MAX fi\_id=[ones(length(time(1:length(yc))),1), time(1:length(yc)), time(1:length(yc)).^3]; fi\_val=[ones(length(time(length(yc):length(y))),1), time(length(yc):length(y)) , time(length(yc):length(y)).^3];

for i=1:m fi\_id=[fi\_id,cos(2\*pi\*i\*time(1:length(yc))/p), sin(2\*pi\*i\*time(1:length(yc))/p)]; end Teta=fi\_id\y(1:length(yc)); yc=fi\_id\*Teta; E\_id=y(1:length(yc))-yc;

for i=1:length(yc) MSE\_id(m)=MSE\_id(m)+E\_id(i).^2; end

for i=1:m fi\_val=[fi\_val,cos(2\*pi\*i\*time(length(yc):length(y))/p), sin(2\*pi\*i\*time(length(yc):length(y))/p)]; end

yval=fi\_val\*Teta; E\_val=y(length(yc):length(y))-yval;

for i=1:length(yval) MSE\_val(m)=MSE\_val(m)+E\_val(i).^2; end

subplot(MAX,1,m), plot(time,y,'r', 1:length(yc),yc, 'g', time(newpoints:length(y)),yval, 'b')

m, MSE\_id(m)=MSE\_id(m)/length(E\_id), MSE\_val(m)=MSE\_val(m)/length(E\_val) end

figure, plot((1:MAX),MSE\_id, 'm', (1:MAX), MSE\_val, 'b'), legend('Identificare', 'Validare')

*p = 12*

*MAX = 2*

*MSE\_id = 0 0*

*yval =*

*588*

*631*

*347*

*504*

*884*

*674*

*586*

*632*

*494*

*979*

*245*

*384*

*538*

*246*

*285*

*381*

*704*

*829*

*509*

*554 335*

*m = 1*

*MSE\_id =*

*1.0e+04 \**

*2.2650 0*

*MSE\_val =*

*1.0e+04 \**

*4.7274 0*

*m = 2*

*MSE\_id =*

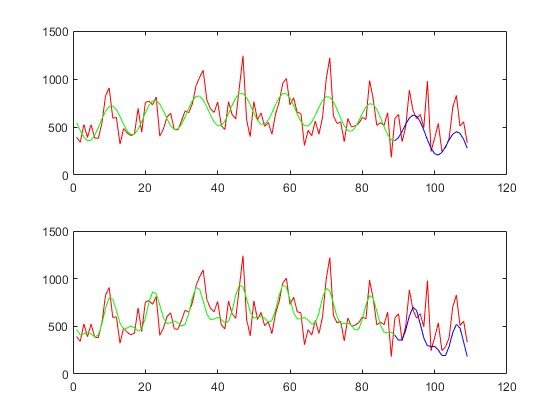
*1.0e+04 \**

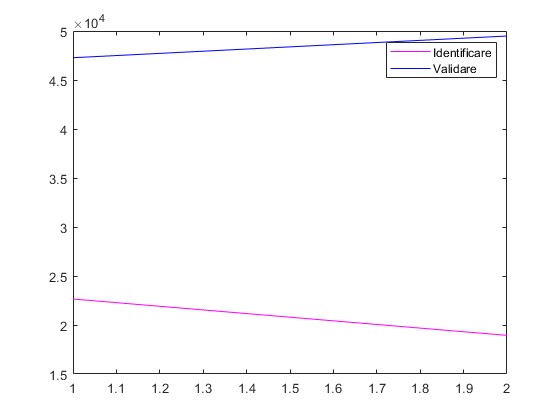
*2.2650 1.8941*

*MSE\_val =*

*1.0e+04 \**

*4.7274 4.9489*





*Published with MATLAB® R2021b*