Model drsečega povprečja z n zakasnitvami

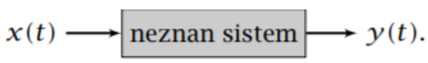
1. domača naloga pri Matematičnem modeliranju

Anže Mur

7. marec 2018

1 Opis problema

Imamo neznan sistem, ki nam pri vhodnem signalu $x(t)$ vrne izhodni signal $y(t)$:



Naše podatke tvori $N+1$ vhodov $x_0, x_1, \ldots, x_N$ in $N+1$ pripadajočih izhodov $y_0, y_1, \ldots, y_N$, ki jih izmerimo pri celoštevilskih časih t iz intervala $[0, N]$. Vhod in izhod sistema, ki pri vhodnem signalu $x(t)$ vrne izhodni signal $y(t)$ označimo:

$$x_i = x(i)$$
$$y_i = y(i)$$

Naš model predpostavlja, da lahko izhod $y(t)$ ob času t zapišemo kot linearno kombinacijo n vhodov $x(t), x(t-1), \ldots, x(t-n+1)$ iz časovnega intervala $[t-n+1, t]$ in tako dobimo enačbo:

$$y_k = h_1 x_{k-n+1} + h_2 x_{k-n} + \ldots + h_{n-1} x_{k-1} + h_n x_k,$$

ki velja za vsak $k = n-1, n, \ldots, N$. Imamo torej sistem $N-n+1$ enačb z n neznankami $h_1, \ldots, h_n$ in predpostavljamo, da je $N-n+1 \geq n$. Z A označimo matriko tega sistema, s $h = [h_1, ..., h_n]^T$ pa vektor neznank in z y desno stran. Poiskati želimo rešitev sistema $Ah = y$ v smislu linearne metode najmanjših kvadratov.

2 Naloge

1. Naloga: Izpelji in zapiši/opiši splošno obliko matrike $\mathbf{A}$ in desne strani $\mathbf{y}$. Kako bi ob znanem $\mathbf{h}$ napovedal izhod sistema $y_0, ..., y_m$, če veš le, kaj so vhodni podatki $x_0, ..., x_m$? Katere komponente izhoda lahko napoveš?

Rešitev:

Matriko A izpeljemo tako, da za x_k vstavimo vse $k = [n-1, n, \ldots, N]$.

Izpeljava matrike $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} x_0 & x_1 & x_2 & \ldots & x_{n-1} \\ x_1 & x_2 & x_3 & \ldots & x_n \\ x_2 & x_3 & x_4 & \ldots & x_{n+1} \\ \vdots & \vdots & \vdots & & \vdots \\ x_{N-n+1} & x_{N-n+2} & x_{N-n+3} & \ldots & x_N \end{bmatrix}$$

Izpeljava matrike $\mathbf{y}$ po sistemu $Ah = y$, kjer je $\mathbf{h}$ oblike $h = [h_1, ..., h_n]^T$.

$$\mathbf{y} = \begin{bmatrix} x_0 \cdot h_1 + x_1 \cdot h_2 + x_2 \cdot h_3 & + & \ldots & + & x_{n-1} \cdot h_n \\ x_l \cdot h_1 + x_2 \cdot h_2 + x_3 \cdot h_3 & + & \ldots & + & x_n \cdot h_n \\ x_2 \cdot h_1 + x_3 \cdot h_2 + x_4 \cdot h_3 & + & \ldots & + & x_{n+1} \cdot h_n \\ \vdots & & & & \vdots \\ x_{N-n+1} \cdot h_1 + x_{N-n+2} \cdot h_2 + x_{N-n+3} \cdot h_3 & + & \ldots & + & x_N \cdot h_n \end{bmatrix} = \begin{bmatrix} y_{n-1} \\ y_n \\ y_{n+1} \\ \vdots \\ y_N \end{bmatrix}$$

Ob znanem $\mathbf{h}$ bi in s podanimi vhodnimi podatki $x_0, \ldots, x_m$ izhod sistema $y_0, \ldots, y_m$ napovedal tako, da bi vhodne podatke pravilno vstavil v matrkiko sistema $\mathbf{A}$ (po n podatkov v vsako vrstico, kjer bi n tekel od $[n-1, N]$ in s tem premikal okvir velikosti n med vhodnimi podatki) in rešil sistem $Ah = y$, kot je prikazano v zgornji izpeljavi matrike $\mathbf{y}$.

Komponente izhoda, ki jih lahko napovemo so $y_n - 1, \ldots, y_N$. Se pravi komponent z indeksom manjšim od n-1 ne moremo napovedati.

2. Naloga: V octave-u napiši funkcijo $h = movavg(x, y, n)$, ki za vhodne podatke $x = [x_0, \ldots, x_N]$ in izhodne podatke $y = [y_0, \ldots, y_N]$ poišče koeficiente $h = [h_1, \ldots, h_n]$ za model drsečega povprečja z n zakasnitvami. (Upoštevaj specifikacije: x, y in h so vrstice.)

Rešitev:
Nalogo rešimo tako, da najprej sestavimo matriko A, tako da iz vhodnih podatkov x, v vsaki iteraciji vzamemeo okvir velikosti n, ter ga premaknemo za eno naprej, dokler ne pridemo do konca vhodnih podatkov x. Nato pa po metodi najmanjših kvadratov rešimo sistem.

Rešitev v octave-u:

```
function h = movavg(x, y, n)
%h = movavg(x, y, n)je funckija ki za vhodne podatke x = [x0, ..., xN]
%in izhodne podatke y = [y0, ..., yN] poisce koeficiente
%h = [h1, ..., hn] za model drsecega povprecja z n zakasnitvami.
%x,y sta podana kot vrsticna vektorja.

%sestavimo matriko A iz podatkov x, tako da vedno vzamemo
%okvir velikosti n iz vhodnih podatkov x,
%in ga v vsaki iteraciji premaknemo za eno naprej

for i = 1:length(x)-n+1;
A(i ,: ) = x(i : i+n-1);
end


%Resimo sistem po metodi najmanjsih kvadratov
%in pri tem zanemarimo prvih n vrednosti y
h = A\(y(n:end))';
%v navodilih je napisano da mora biti h vrsticni vektor
h = h';

%!test
%! x=[2 6 8 3];
%! y=[2 10 20 19];
%! h=[2 1];
%! n=2;
%! assert(movavg(x,y,n),h,eps*10)

%!test
%! x=[1 2 3 4 5 6 7 8 9 10];
%! y=[1 3 6 9 12 15 18 21 24 27];
%! h=[1 1 1];
%! n=3;
%! assert(movavg(x,y,n),h,eps*10)
```

3. Naloga: Recimo, da so vsi vhodi in izhodi konstantni, t.j. $x_i = y_i = 1$. Najprej ugani, nato pa še izpelji rešitev **h** za poljubna N in n. Primerjaj z izhodom, ki ga vrne movavg.

Intuitivna rešitev:

V primeru da so vsi vhodi in izhodi konstantni je vektor neznank sistema h, tudi konstantne vrednosti. Vektor h je sestavljen iz n elementov, ki imajo vrednosti $\frac{1}{n}$.

Naj bosta vhod izhod sestavljena iz neke konstane vrednosti j, torej $x_i = y_i = j$. Rešitev sistema lahko uganemo, saj vemo, da mora biti vrednost, ki jo dobimo s seštevanjem n vrednosti vhoda x po množenju z vektorem h enaka j. In ker je h vektor velikosti n, do iskane vrednosti $y_i = j$ lahko pridemo samo tako da je h sestavljen iz n elementov vrednosti $\frac{1}{n}$.

Izpeljava vrednosti poljubnega elementa h, h_i:

$$\begin{aligned} n \cdot x_i \cdot h_i &= y_i \\ h_i &= \frac{y_i}{n \cdot x_i} \\ h_i &= \frac{1}{n}; \quad \text{ker velja } x_i = y_i \end{aligned}$$

Oblika vektorja neznank $\mathbf{h} = [\frac{1}{n}_1, \ldots, \frac{1}{n}_n]$.

Izhod funkcije **movavg** ustreza izpeljavi za računanje vrednosti vektorja h.

Rešitev s pomočjo Moore-Penrose inverza:

Če imamo matriko A, ki je sestavljena iz samih enakih konstanih vrednosti 1 in je velikosti mxn vemo, da je njen MP inverz dimenzij nxm in je sestavljen iz vrednosti $\frac{1}{n \cdot m}$. Spodnja izpeljava velja v primeru, če sta $y_i = x_i = 1$.

$$A = \begin{bmatrix} 1_1 & 1_2 & 1_3 & . & . & . & 1_n \\ . & & & & & & \\ . & & & & & & \\ . & & & & & & \\ 1_m & 1_{m+1} & 1_{m+2} & . & . & . & 1_{mn} \end{bmatrix} \text{; dimenzij m x n}$$

$$A^+ = \begin{bmatrix} \frac{1}{n \cdot m}_1 & \frac{1}{n \cdot m}_2 & \frac{1}{n \cdot m}_3 & . & . & . & \frac{1}{n \cdot m}_m \\ . & & & & & & \\ . & & & & & & \\ . & & & & & & \\ \frac{1}{n \cdot m}_{n+1} & \frac{1}{n \cdot m}_{n+2} & \frac{1}{n \cdot m}_{n+3} & . & . & . & \frac{1}{n \cdot m}_{nm} \end{bmatrix} \text{; dimenzij n x m}$$

$$h = A^{+} \cdot y; \text{ kjer je } y_i = 1$$

$$h_i = \frac{1}{n \cdot m_1} + \ldots + \frac{1}{n \cdot m_m}$$
$$h_i = \frac{1}{n \cdot m} \cdot m$$
$$h_i = \frac{1}{n}$$

4. Naloga: Napiši še funkcijo $y = prediction(x, h)$, ki iz vhodnih podatkov x in koeficientov h napove izhod y na podlagi našega modela.

Rešitev:
Nalogo rešimo tako, da najprej sestavimo matriko A, tako da iz vhodnih podatkov x, v vsaki iteraciji vzamemeo okvir velikosti n(izračunamo ga iz dolžine vektorja h), ter ga premaknemo za eno naprej, dokler ne pridemo do konca vhodnih podatkov x. Nato pa rešimo sistem $y = A \cdot h$.

Rešitev v octave-u:

```
function y = prediction(x, h)
%y = prediction(x, h) je funkcija, ki iz vhodnih podatkov x
%in vektorja koeficientov h napove izhod y na podlagi nasega modela
%x in h sta podana kot vrsticna vektorja
n = length(h);

for i = 1:length(x)-n+1;
A(i ,: ) = x(i : i+n-1);
end

y = A*h';


%!test
%! x=[1 2 3 4 5 6 7 8 9 10];
%! y=[6 9 12 15 18 21 24 27];
%! h=[1 1 1];
%! assert(prediction(x,h),y',eps*10)

%!test
%! x=[2 6 8 3];
%! y=[10 20 19];
%! h=[2 1];
%! assert(prediction(x,h),y',eps*10)
```

5. Naloga: Preveri delovanje na podatkih iz datotek **io-train.txt** in **io-test.txt** za $n = 1, 2, 3, 5, 10$. (io-train.txt je učni nabor, uporabi ga za izračun h. io-test.txt je testni nabor, napoved modela testiraj na podatkih iz tega nabora.) Kako je točnost napovedi odvisna od števila zakasnitev n?

Rešitev:
Funkcijo **movavg** dopolnimo z demonstracijsko funkcijo **demo**, ki jo kličemo z ukazom: *demo movavg*. Ta iz učnih primerov prebere podatke in s pomočjo tesnih primerov izračuna naš model, ter napake izračuna modela. Ob klicu funckije se nam izrišejo tudi grafi napak, ter grafi primerjav y_{test} in y_{model} ob različnih zakasnitvah n.

Dopolnitev funkcije movavg:

```
%!demo
%! n=[1 2 3 5 10];
%! train_data = load("io-train.txt");
%! test_data = load("io-test.txt");
%!
%! x_train = train_data(1,: );
%! y_train = train_data(2,: );
%!
%! x_test = test_data(1,: );
%! y_test = test_data(2,: );
%!
%! count = 1;
%! for i = n
%!
%!  h = movavg(x_train, y_train, i);
%!  y_model = prediction(x_test, h);
%!  tmp = y_test-y_model;
%!  error(i) = norm(tmp);
%!  err1 = mean(abs(tmp), 1);
%!  abs_error(i) = mean(err1,2);
%!  figure(count)
%!  plot(y_test, 'g'); hold on; plot (y_model, 'r');
%!  title(['n = ' num2str(i) ''])
%!  legend("y_{test}", "y_{model}")
%!
%!  count = count + 1;
%!
%! end
%!
%! figure(6)
%! plot(n, error(n), 'r');
%! title("Napaka modela v odvisnosti od stevila zakasnitev n")
%!
%! figure(7)
%! plot(n, abs_error(n), '-o');
%! title("Absolutna napaka modela v odvisnosti od stevila zakasnitev n")
```

Razlaga rezultatov dopolnjene funkcije movavg:

Pri tesnih n-jih je rezultat najbolj optimalen pri n=10, kjer je prileganje našemu modelu največje. Glede na naše rezultate, bi lahko sklepali, da se napaka modela s povečevanjem števila zakasnitev n začne zmanjševati. Vendar zaradi testnih primerov pri vrednostih, ko je n enak 1, 2, 3 in 5 vidimo, da temu ni res, saj se točnost napovedi ne povečuje, kljub temu da povečujemo n. Iz spodnjih grafov lahko razberemo, da je pri $n = 5$ ujemanje veliko manjše kot pri manjših n-jih.

Pri $n = 5$ napaka zelo poskoči. Taka odstopanja, bi se mogoče dalo omiliti, če bi za izračun modela uporabili Moore-Penrose inverz.

Pri velikih n-jih prihaja do odstopanj, ki pa absolutno ne variirajo v neke ekstremne vrednosti in odstopanja niso tako velika, vendar pa prileganje modela ni tako dobro, kot pri $n = 10$.

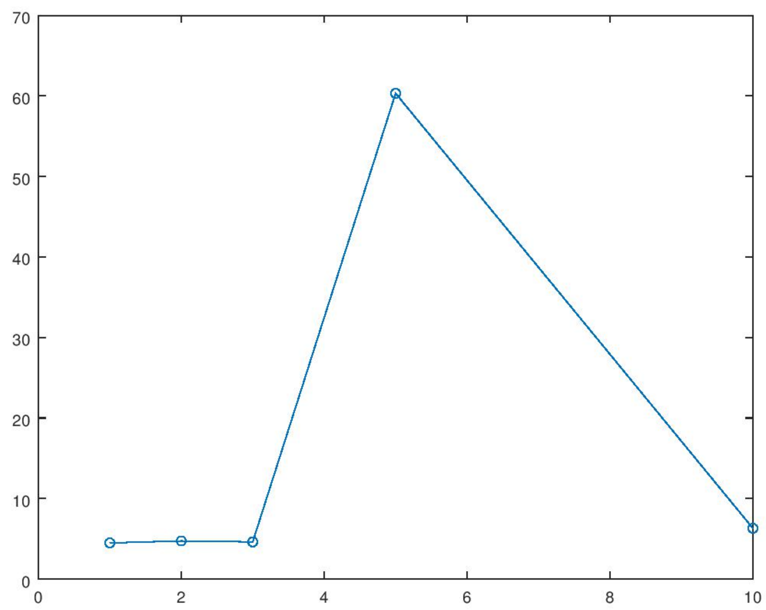


Figure 1: Absolutna napaka modela v odvisnosti od števila zakasnitev n.

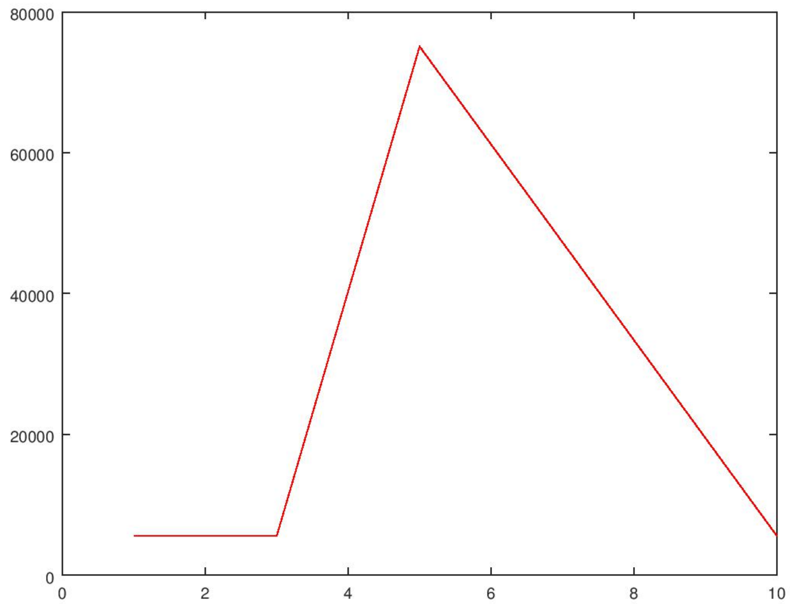


Figure 2: Napaka modela v odvisnosti od števila zakasnitev n.

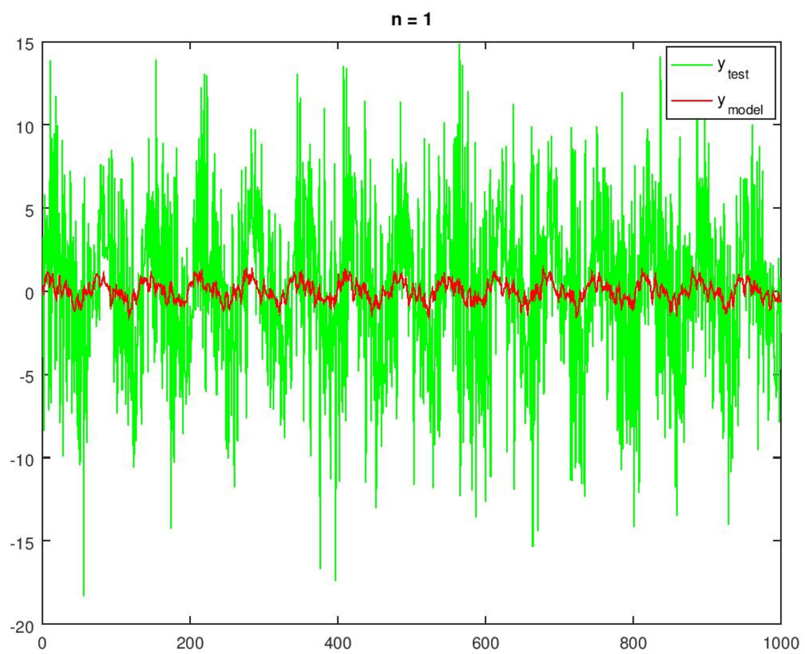


Figure 3: Prikaz y_{test} in y_{model} pri $n = 1$

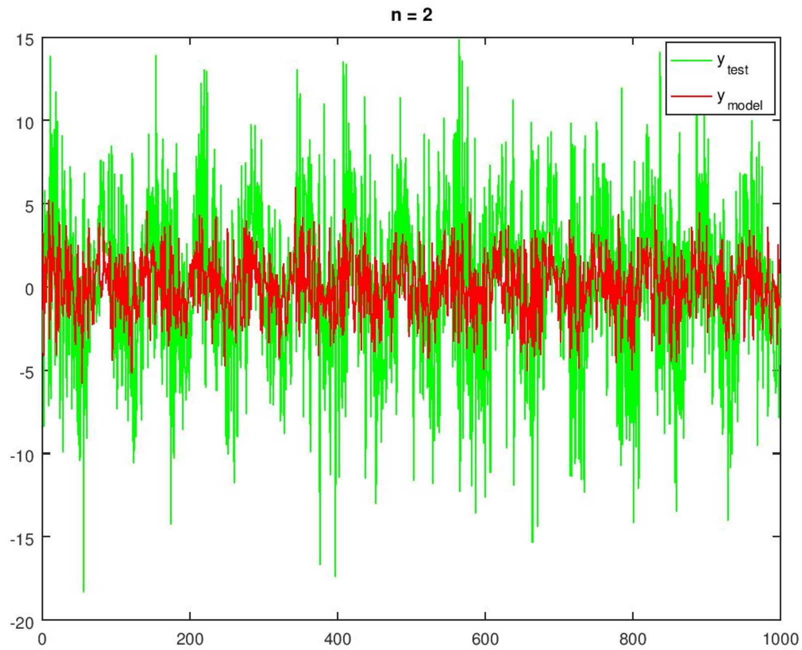


Figure 4: Prikaz y_{test} in y_{model} pri $n = 2$

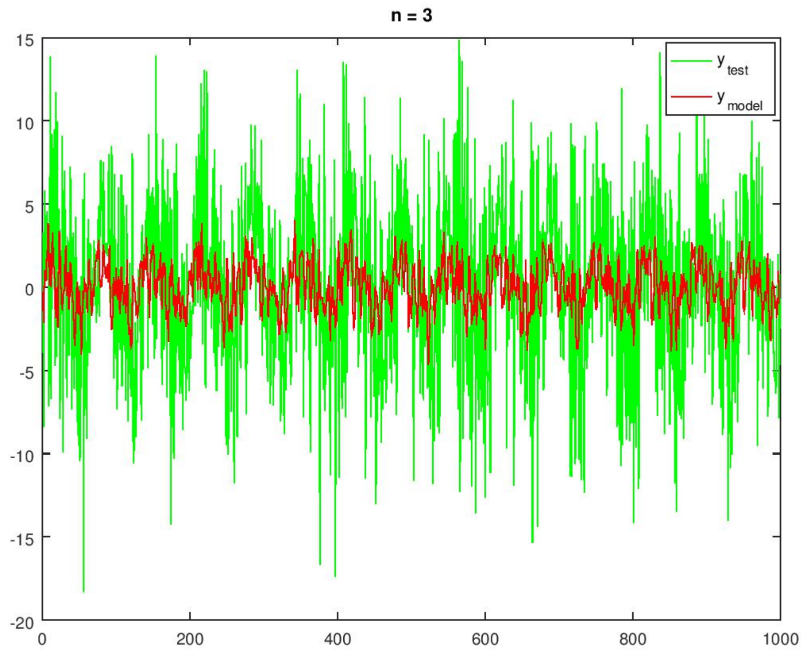


Figure 5: Prikaz y_{test} in y_{model} pri $n = 3$

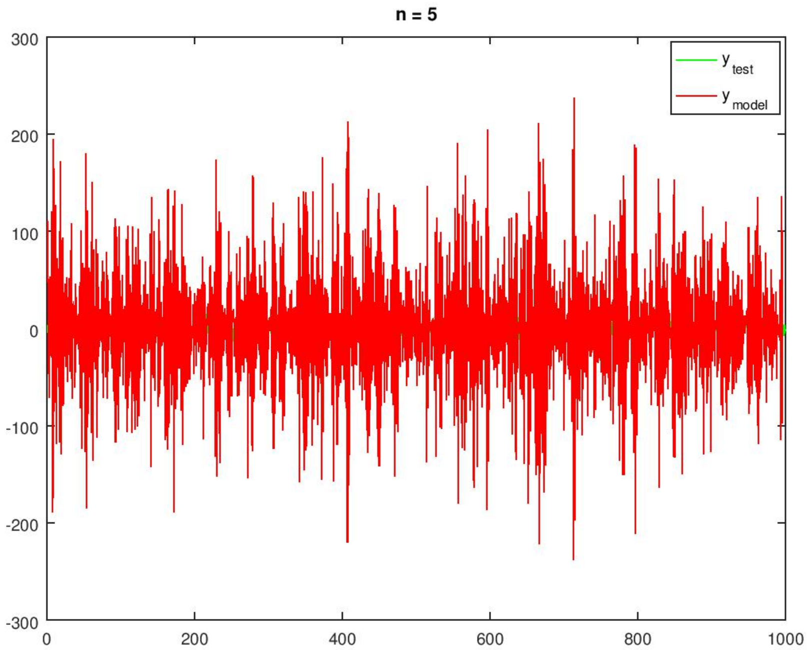


Figure 6: Prikaz y_{test} in y_{model} pri $n = 5$

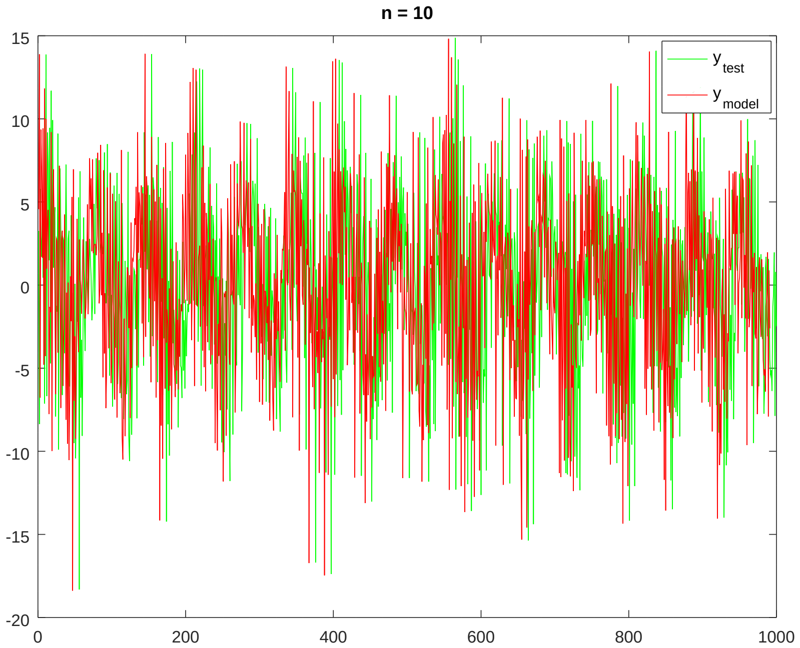


Figure 7: Prikaz y_{test} in y_{model} pri $n = 10$

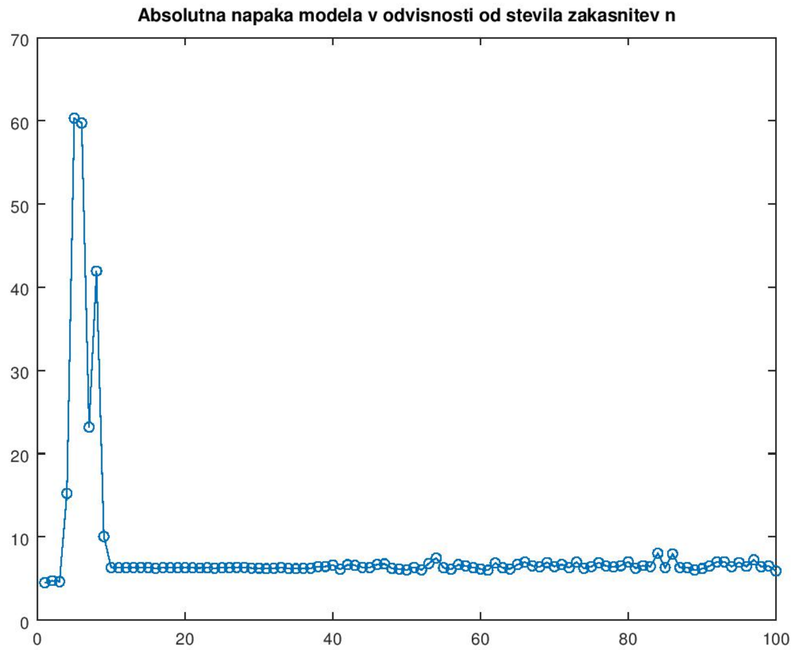


Figure 8: Absolutna napaka modela v odvisnosti od števila zakasnitev n na intervalu [1,100].