

Synchronizacja z pilotem w radiu FM

Tomasz Zieliński, KT AGH

W dokumencie „Radio_FM_Równania.doc” opisano podstawy wysyłania i odbioru sygnałów w analogowym radiu FM. Zgodnie z tym dokumentem, $y(n)$ to odtworzony w odbiorniku sygnał hybrydowy radia FM. Składa się on z:

- sygnału mono (L+R),
- pilota 19 kHz,
- sygnału stereo (L–R) przesuniętego do częstotliwości 38 kHz („podwojonego” pilota),
- sygnału RDS, przesuniętego do częstotliwości 57 kHz („potrojonego” pilota).

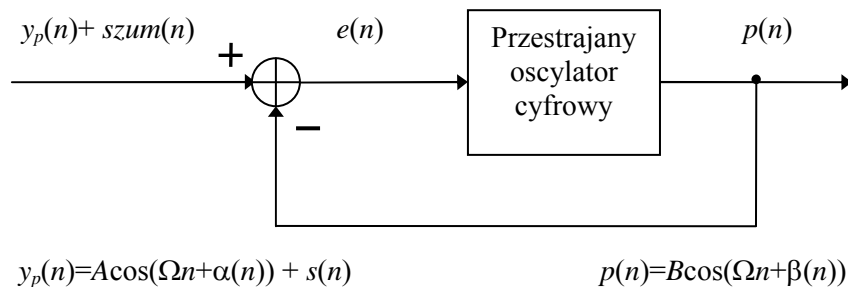
Niech $y_p(n)$ oznacza wynik wąskopasmowej filtracji $y(n)$ wokół 19 kHz:

$$y_p(n) = BP_{19\text{kHz}}(y(n)) = A \cos(\Omega n + \alpha(n)) + s(n) = A \cos(\theta'(n)) + s(n), \quad \Omega = 2\pi \frac{f_{pl}}{f_{pr}}. \quad (1)$$

gdzie $s(n)$ oznacza szum, f_{pl} – częstotliwość pilota 19 kHz, f_{pr} – częstotliwość próbkowania. Aby odzyskać z $y(n)$ sygnały L-R i RDS, musimy odtworzyć kąt $\theta'(n)$ na podstawie sygnału $y_p(n)$. Robimy to za pomocą metody „strojonego modelu”. Jej schemat funkcjonalny jest przedstawiony na rysunku 1. Na początku zakładamy, że pulsacja Ω jest dokładnie znana. Generujemy więc sygnał $p(n)$ postaci:

$$p(n) = B \cos(\Omega n + \beta(n)) = B \cos(\theta(n)). \quad (2)$$

i chcemy iteracyjnie dostroić kąt $\beta(n)$ do kąta $\alpha(n)$ ($\theta(n) \rightarrow \theta'(n)$). Jak to zrobić?



Rys. 1. Układ synchronizacji z sygnałem pilota.

Niech ε oznacza wartość oczekiwaną kwadratu błędu pomiędzy sygnałami (1) i (2):

$$\varepsilon = E \left[\left(y_p(n) - p(n) \right)^2 \right] \quad (3)$$

Podstawmy zależności (1) i (2) do (3). W celu uproszczenia analizy założmy, że amplitudy sygnałów są równe ($A=B$). Załóżmy także chwilowo, że ich przesunięcia fazowe α i β są stałe i niezmiennie w czasie: $\alpha(n)=\text{const}$, $\beta(n)=\text{const}$. W przypadku kiedy w (1) szum $s(n)$ jest gaussowski ($E[s(n)]=0$, $E[s^2(n)]=\sigma_s^2$), to otrzymujemy:

$$\varepsilon = \sigma_s^2 + A^2(1 - \cos(\alpha - \beta)) - E[A^2 \cos(2\Omega n + \alpha + \beta)(1 - \cos(\alpha - \beta))] \quad (4)$$

Ponieważ ostatni składnik jest równy zero, kiedy wartość oczekiwana po zbiorze realizacji jest zastąpiona przez średnią czasową (wówczas średnia wartość z $\cos()$ równa się zero), otrzymujemy:

$$\varepsilon = \sigma_s^2 + A^2(1 - \cos(\alpha - \beta)) \quad (5)$$

czyli **wartość minimalną** $\varepsilon_{\text{minimum}} = \sigma_s^2$ dla $\beta = \alpha$.

Wniosek. Uogólniając, wyciągamy wniosek, że należy adaptacyjnie zmieniać wartość kąta $\beta(n)$ w sygnale $p(n)$ (który generuje cyfrowy syntezytor częstotliwości) i poszukiwać minimum funkcji błędu (3). Dla tego minimum będzie zachodzić równość $\beta(n) = \alpha$.

Usuając operator $E[\cdot]$ wartości oczekiwanej w równaniu (3) otrzymujemy zadanie optymalizacji, sprowadzające się do minimalizacji błędu chwilowego pomiędzy sygnałami (1)(2):

$$\hat{\varepsilon} = [y_{pl}(n) - p(n)]^2. \quad (6)$$

W teorii filtrów adaptacyjnych LMS (w metodzie stochastycznego gradientu) przyjmuje się następujące równanie przestrajania wartości parametru, u nas kąta $\beta(n)$:

$$\beta(n+1) = \beta(n) - \mu \frac{\partial \hat{\varepsilon}}{\partial \beta}. \quad (7)$$

czyli „idzie się” w kierunku przeciwnym (znak minus) do kierunku wzrostu funkcji (gradientu funkcji).

Obliczmy pochodną cząstkową funkcji błędu $\hat{\varepsilon}$ (6) względem $\beta(n)$:

$$\frac{\partial \hat{\varepsilon}}{\partial \beta} = 2A \sin(\Omega n + \beta(n)) [y_{pl}(n) - p(n)]. \quad (8)$$

i podstawmy wynik do (7):

$$\beta(n+1) = \beta(n) - 2\mu A \sin(\Omega n + \beta(n)) \cdot [y_{pl}(n) - p(n)]. \quad (9a)$$

$$\beta(n+1) = \beta(n) - 2\mu A \sin(\Omega n + \beta(n)) \cdot \text{err}(n). \quad (9b)$$

Czyli podczas adaptacji nowa wartość $\beta(n+1)$ powinna być równa poprzedniej wartości $\beta(n)$, do której dodaje się korektę proporcjonalną do błędu $e(n) = y_p(n) - p(n)$.

Dokonujemy dalszych przekształceń (8). Uwzględniamy (2), czyli że $p(n) = A \cdot \cos(\Omega n + \beta(n))$, i otrzymujemy:

$$\frac{\partial \hat{\varepsilon}}{\partial \beta} = 2A \sin(\Omega n + \beta(n)) y_{pl}(n) - 2A^2 \sin(\Omega n + \beta(n)) \cos(\Omega n + \beta(n)) \quad (10)$$

Następnie wykorzystujemy zależności:

$$\sin(a+b) = \sin(a)\cos(b) + \cos(a)\sin(b), \quad \text{dla } a = b \text{ mamy } \sin(2a) = 2 \sin(a)\cos(a)$$

skąd:

$$2A^2 \sin(\Omega n + \beta(n)) \cos(\Omega n + \beta(n)) = A^2 \sin(2\Omega n + 2\beta(n)) \quad (11)$$

Teraz uwzględniamy (11) w (10):

$$\frac{\partial \hat{\varepsilon}}{\partial \beta} = 2A \sin(\Omega n + \beta(n)) y_{pl}(n) - A^2 \sin(2\Omega n + 2\beta(n)). \quad (12)$$

Ponieważ wartość średnia drugiego wyrażenia jest równa zero, to nie ma ono wpływu na zbieżność algorytmu adaptacji. Dlatego zaniedbujemy je:

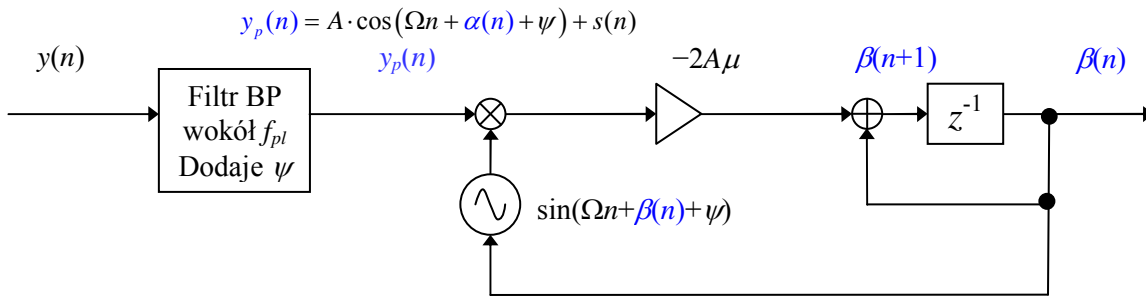
$$\frac{\partial \hat{\varepsilon}}{\partial \beta} \approx 2A \sin(\Omega n + \beta(n)) y_{pl}(n). \quad (13)$$

Wynik podstawiamy do algorytmu adaptacji (7):

$$\beta(n+1) = \beta(n) - 2A\mu \sin(\Omega n + \beta(n)) \cdot y_{pl}(n). \quad (14a)$$

$$\beta(n+1) = \beta(n) - 2A\mu \sin(\Omega n + \beta(n)) \cdot \text{error}(n). \quad (14b)$$

Równanie (14) możemy wykorzystać do uszczegółowienia układu synchronizacji z pilotem 19 kHz, przedstawionym na rysunku 1.



Rys. 2. Schemat blokowy adaptacyjnej pętli, służącej do odzyskiwania (śledzenia) kąta fazowego $\alpha(n)$ sygnału wejściowego $y_p(n)$. Kąt $\beta(n)$ stara się zrównać z $\alpha(n)$: $\beta(n) \rightarrow \alpha(n)$. Przesunięcie fazowe ψ jest wprowadzane przez filtr BP. $s(n)$ oznacza szum.

Stosując oznaczenie:

$$\theta(n) = \Omega n + \beta(n), \quad \Omega = 2\pi(f_{pl} / f_s) \quad (15)$$

z (14) otrzymujemy:

$$\theta(n+1) = \theta(n) + \Omega - 2A\mu \sin(\theta(n)) \cdot \text{error}(n). \quad (16)$$

Powyżej założyliśmy **znajomość** pulsacji Ω i jej **niezmiennność**. Dlatego estymacja (oszacowanie) wartości kąta $\theta(n)$ sprowadziła się do dostrajania wartości kąta $\beta(n)$ (2) do kąta $\alpha(n)$ (1). Teraz „rozluźnijmy” nasze założenia i uwzględnijmy fakt, że wartość Ω_{osc} , przyjętego w oscylatorze cyfrowym może się różnić od Ω sygnału odebranego $y_p(n)$. tzn. oscylator używa wartości $\Omega + \Delta\Omega$. Z (16) mamy:

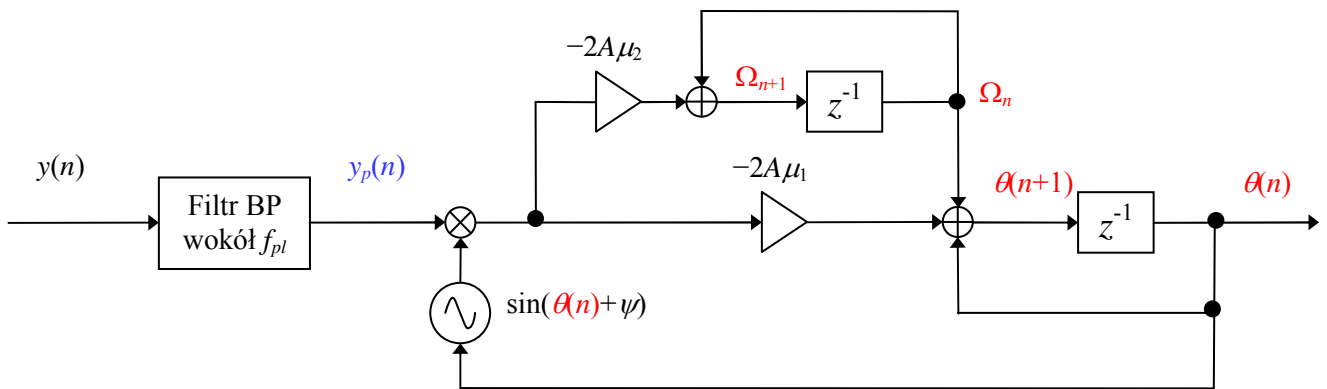
$$\theta(n+1) = \theta(n) + \underbrace{(\Omega + \Delta\Omega)}_{\Omega_{osc}} - 2A\mu \sin(\theta(n)) \cdot \text{error}(n). \quad (17)$$

Dlatego błąd pomiędzy całkowitym kątem $\theta(n)$ sygnału $p(n)$ oscylatora, wyliczony z (17), a całkowitym kątem $\theta'(n)$ sygnału $y_p(n)$ może liniowo maleć lub narastać w zależności od $\Delta\Omega$. Ale błąd ten także będzie wpływał na błąd (6) chwilowego niedopasowania syntezowanego cyfrowo sygnału $p(n)$ do analizowanego sygnału $y_p(n)$. Dlatego wartość Ω_{osc} możemy także przestrajać proporcjonalnie do błędu $\text{error}(n)$ oraz wartości $\sin(\theta(n))$. Końcowe równania zapisujemy w sposób następujący:

$$\theta(n+1) = \theta(n) + \Omega_n - 2A\mu_1 \sin(\theta(n)) \cdot error(n) \quad (17a)$$

$$\Omega_{n+1} = \Omega_n - 2A\mu_2 \sin(\theta(n)) \cdot error(n). \quad (17b)$$

Odpowiada im algorytm przedstawiony poniżej na rysunku 3.



Rys. 3. Schemat blokowy adaptacyjnej pętli synchronizacyjnej, służącej do jednoczesnego odzyskiwania (śledzenia) sumarycznego kąta $\theta'(n) = \Omega n + \alpha(n)$ sygnału wejściowego. Przesunięcie fazowe ψ jest wprowadzane przez filtr BP. $s(n)$ oznacza szum. $y_p(n) = A \cdot \cos(\theta'(n) + \psi) + s(n)$

Końcowy algorytm programowej pętli synchronizacyjnej

W związku z powyższym algorytm programowej pętli adaptacyjnej, stosowanej w zadaniu odtworzenia sygnału pilota w radiu FM, ma następującą postać:

1. Wybór wartości μ_1 i μ_2 .
2. Inicjalizacja: $\Omega(1) = 2\pi(f_{pl} / f_s)$, $\theta(1) = 0$.
3. For $n=1,2,3,\dots$:

$$error(n) = \sin(\theta(n)) y_{pl}(n),$$

$$\theta(n+1) = \theta(n) + \Omega(n) - \mu_1 \cdot error(n).$$

$$\Omega(n+1) = \Omega(n) - \mu_2 \cdot error(n).$$

Odpowiada temu poniższy kod Matlaba, stanowiący część programu [fm_dekoder_stereo_RDS_pilot_norma.m](#). Dodatkowo przykładowy kod Matlaba opisanej metody synchronizacji z pilotem jest w programie [fm_dekoder_PLL.m](#) jako metoda=1, w wersji 1.

Kod Matlaba programowej pętli synchronizacyjnej z pilotem (metoda=1, ver=1)

```

% Odseparowanie sygnału pilota 19 kHz (filtracja pasmowoprzepustowa)
yp = filter(hBP19,1,y); Ny=length(yp);
% Pętla synchronizacyjna do odtworzenia częstotliwości i fazy pilota
% i na tej podstawie sygnałów: c19, c38 i c57
mil = 1e-2; mi2 = mil^2/4;
freq = 2*pi*fpl/f_s; theta = zeros(1,Ny);
for n = 1 : Ny-1
    pherr = sin(theta(n))*yp(n);
    theta(n+1) = theta(n) + freq - mil*pherr;
    freq = freq - mi2*pherr;
end

```

```

c19 = cos(theta(1:end));
c38 = cos(2*theta(1:end));
c57 = cos(3*theta(1:end));

```

Wersja alternatywna programowej pętli synchronizacyjnej

Teraz postaramy się przeprowadzić alternatywne wyprowadzenie dla przypadku, kiedy pulsacje Ω i Ω_{osc} są różne:

$$y_p(n) = BP_{19\text{kHz}}(y(n)) = A \cos(\Omega n + \alpha(n)) + s(n), \quad (18)$$

$$p(n) = A \cos(\Omega_{osc} n + \beta(n)). \quad (19)$$

W takim przypadku (3) jest równe:

$$\begin{aligned}
\varepsilon &= E \left[\left(y_p(n) - p(n) \right)^2 \right] = \\
&= E \left[\left(A \cos(\Omega n + \alpha(n)) + s(n) \right)^2 + \left(A \cos(\Omega_{osc} n + \beta(n)) \right)^2 + \dots \right. \\
&\quad \left. \dots - 2 \left(A \cos(\Omega n + \alpha(n)) + s(n) \right) \left(A \cos(\Omega_{osc} n + \beta(n)) \right) \right] = \\
&= E \left[\left(A^2 \cos^2(\Omega n + \alpha(n)) + A \cos(\Omega n + \alpha(n)) s(n) + s^2(n) \right) + \dots \right. \\
&\quad \left. \dots + \left(A \cos(\Omega_{osc} n + \beta(n)) \right)^2 - 2 \left(A \cos(\Omega n + \alpha(n)) + s(n) \right) \left(A \cos(\Omega_{osc} n + \beta(n)) \right) \right] = \\
&= \sigma_s^2 + E \left[A^2 \cos^2(\Omega n + \alpha(n)) + A^2 \cos^2(\Omega_{osc} n + \beta(n)) + \dots \right. \\
&\quad \left. \dots - 2 A^2 \cos(\Omega n + \alpha(n)) \cos(\Omega_{osc} n + \beta(n)) \right] = \\
&= \sigma_s^2 + A^2 E \left[\cos^2(\Omega n + \alpha(n)) + \cos^2(\Omega_{osc} n + \beta(n)) + \dots \right. \\
&\quad \left. \dots - \cos((\Omega + \Omega_{osc})n + \alpha(n) + \beta(n)) - \cos((\Omega - \Omega_{osc})n + \alpha(n) - \beta(n)) \right] = \\
&= \sigma_s^2 + A^2 \left(1 - E \left[\cos((\Omega - \Omega_{osc})n + \alpha(n) - \beta(n)) \right] \right). \quad (20)
\end{aligned}$$

Poprzednio mieliśmy (5):

$$\varepsilon = \sigma_s^2 + A^2 (1 - \cos(\alpha - \beta))$$

i minimum funkcji kosztu dla $\beta = \alpha$. $\beta(n)$ dostrajało się do $\alpha(n)$ i śledziło wolnozmiennie $\alpha(n)$. Teraz $\beta(n)$ będzie dostrajało się do:

$$\beta(n) \rightarrow (\Omega - \Omega_{osc})n + \alpha(n) \quad (21)$$

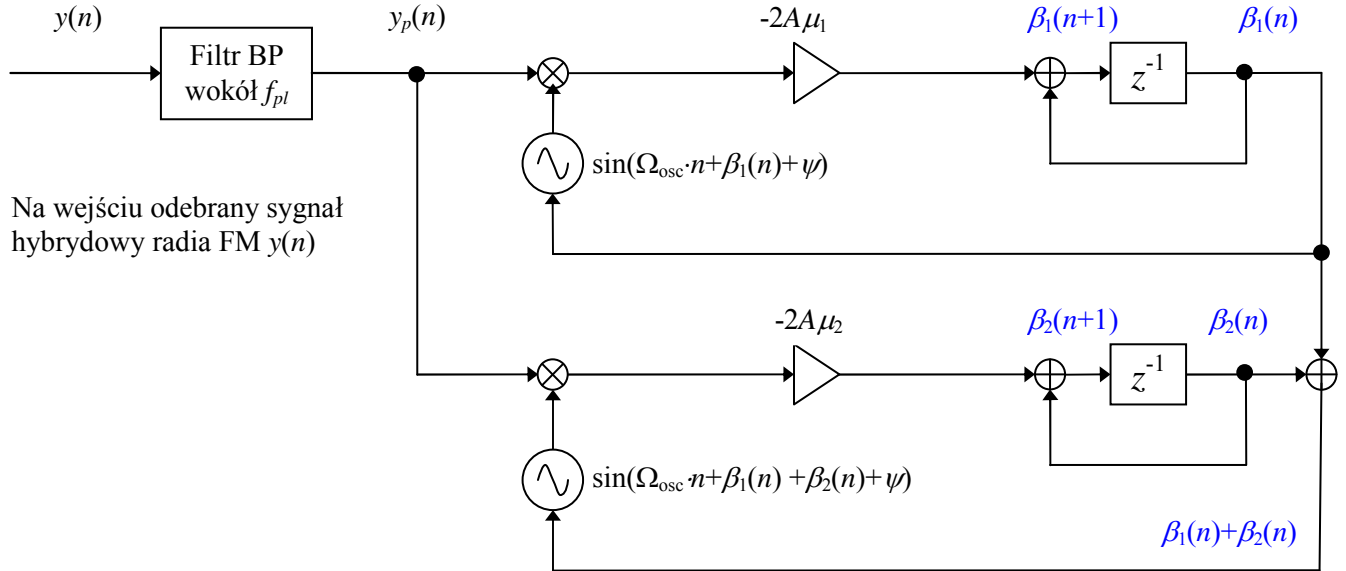
Składnik $(\Omega - \Omega_{osc})n$ narasta (lub maleje liniowo). Jednym z rozwiązań problemu jest zastosowanie drugiej pętli adaptacyjnej. Kąt $\beta_1(n)$ pierwszej pętli zbiega się do (21), a kąt $\beta_2(n)$ drugiej pętli – do $\alpha(n)$. Z (14) mamy:

$$\beta_1(n+1) = \beta_1(n) - 2A\mu_1 \sin(\Omega_{osc} n + \beta_1(n)) \cdot y_{pl}(n), \quad (22)$$

$$\beta_2(n+1) = \beta_2(n) - 2A\mu_1 \sin(\Omega_{osc} n + \beta_1(n) + \beta_2(n)) \cdot y_{pl}(n)$$

$$\begin{aligned}
&= \beta_2(n) - 2A\mu_1 \sin(\Omega_{osc}n + (\Omega - \Omega_{osc})n + \alpha(n) + \beta_2(n)) \cdot y_{pl}(n) \\
&= \beta_2(n) - 2A\mu_1 \sin(\Omega n + \alpha(n) + \beta_2(n)) \cdot y_{pl}(n)
\end{aligned} \tag{23}$$

W konsekwencji odtwarzamy i śledzimy całkowity kąt analizowanego sygnału $y_p(n)$. Schemat blokowy układu, opisanego wyżej, jest przedstawiony na rysunku 4.



Rys. 4. Schemat blokowy alternatywnego układu synchronizacji z pilotem (dwie pętle adaptacyjne), mogącego jednocześnie śledzić wartość kąta fazowego α nośnej Ω jak i sam dryft częstotliwości nośnej Ω . $\Omega_{osc} = 2\pi(f_{osc}/f_{pr})$. Kąt ψ wprowadza filtr BP.

Alternatywnemu układowi synchronizacji z pilotem, opisanemu powyżej, odpowiada poniższy kod Matlaba. Stanowi on część programu **fm_dekoder_PLL.m** jako metoda=1, w wersji 2.

Kod Matlaba programowej pętli synchronizacyjnej z pilotem (metoda=1, ver=2)

```

=====
% Odseparowanie sygnału pilota 19 kHz (filtracja pasmowoprzepustowa)
yp = filter(hBP19,1,y); Ny=length(yp);
% Pętla synchronizacyjna do odtworzenia częstotliwości i fazy pilota
% i na tej podstawie sygnałów: c19, c38 i c57
f0 = fpilot-5; % częstotliwość osc. w odbiorniku
Om = 2*pi*f0/fs; % omega
mu1 = 0.01; mu2 = 0.005; % wsp. szybkości adaptacji
beta1 = zeros(1,Ny); beta1(1) = 0; % inicjalizacja kąta beta1
beta2 = zeros(1,Ny); beta2(1) = 0; % inicjalizacja kąta beta2
theta = zeros(1,Ny); theta(1) = 0; % inicjalizacja theta
for n=1:Ny-1 % podwójna pętla PLL
    theta(n) = Om*(n-1) + beta1(n) + beta2(n); % kąt całkowity
    err1 = yp(n)*sin(Om*(n-1)+beta1(n)); % błąd 1
    err2 = yp(n)*sin(Om*(n-1)+beta1(n)+beta2(n)); % błąd 2
    beta1(n+1) = beta1(n) - mu1*err1; % uaktualnienie 1
    beta2(n+1) = beta2(n) - mu2*err2; % uaktualnienie 2
end
c19 = cos(theta(1:end-1));
c38 = cos(2*theta(1:end-1));
c57 = cos(3*theta(1:end-1));
=====

```