Sterowanie Procesami Ciągłymi
Laboratoria
03.01.2021

Sprawozdanie 5
Regulacja adaptacyjna i predykcyjna

Autor
Mikołaj Zapotoczny
(252939)

Prowadzący
mgr inż. Paweł Mielcarek

1 Wstęp

Metoda identyfikacji najmniejszych kwadratów polega na tym, że na podstawie wejścia i jego odpowiedzi wyznacza się przybliżoną wartość parametrów rzeczywistych obiektu.

$$\begin{cases} V_k = & a_0 * u_k + a_1 * u_{k-1} \\ y_k = & a_0 * u_k + a_1 * u_{k-1} + z_k \end{cases}$$

$$u_k = \frac{y_k - \widehat{a}_1 * u_{k-1}}{\widehat{a}_0}$$

$$\widehat{a} = \begin{bmatrix} \widehat{a}_0 \\ \widehat{a}_1 \end{bmatrix}$$

$$Y_k = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}$$

Są trzy typy metody identyfikacji parametrów systemów:

1.1 Metoda offline

Jest to najprostsza metoda, gdyż operuje ona tylko na próbce czasu rzeczywistego. W celu obliczenia estymatora korzystamy ze wzoru:

$$\widehat{a} = (X_k^T * X_k)^{-1} * X_k^T * Y_k$$

$$X_k = \begin{bmatrix} y_0 & u_1 \\ y_1 & u_2 \\ y_2 & u_3 \\ \vdots & \vdots \\ y_{k-1} & u_k \end{bmatrix}$$

1.2 Metoda rekurencyjna

W tej metodzie wyznaczamy przybliżenia wartości a_0 i a_1 korzystając z próbki czasu teraźniejszego oraz przeszłego. Tutaj trzeba wprowadzić macierz kowariancji P_k.

$$X_k = \begin{bmatrix} y_{k-1} \\ u_k \end{bmatrix}$$

(tutaj k to numer próbki w teraźniejszości)

$$P_k = P_{k-1} - \frac{P_{k-1} * X_k * X_{k-1}^T * P_{k-1}}{1 + X_k^T * P_{k-1} * X_k}$$

$$\widehat{a_k} = \widehat{a}_{k-1} + P_k * X_k(Y_k - X_k^T * \widehat{a}_{k-1})$$

1.3 Ważona metoda rekurencyjna

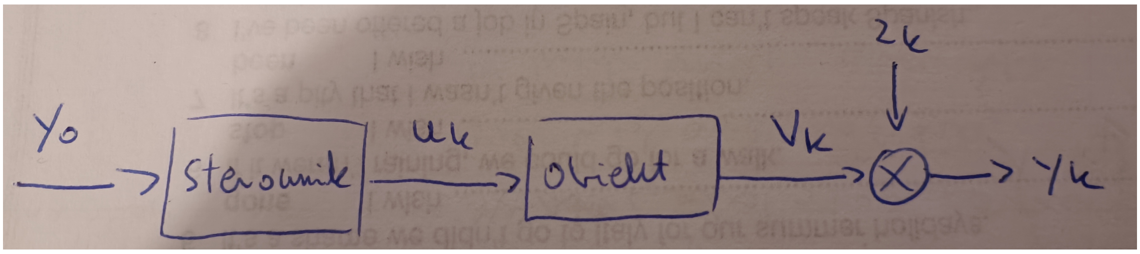
W tej metodzie wyznaczamy przybliżenia wartości a_0 i a_1 korzystając z próbki czasu teraźniejszego oraz przeszłego, ale dodatkowo korzystamy z współczynnika zapomnienia $\lambda \in (0, 1]$. Im próbka młodsza tym ma większą wagę. Tutaj też trzeba wprowadzić macierz kowariancji P_k.

$$\widehat{a_k} = \widehat{a}_{k-1} + P_k * X_k(Y_k - X_k^T * \widehat{a}_{k-1})$$

$$X_{k-1} = \begin{bmatrix} y_{k-1} \\ u_k \end{bmatrix}$$

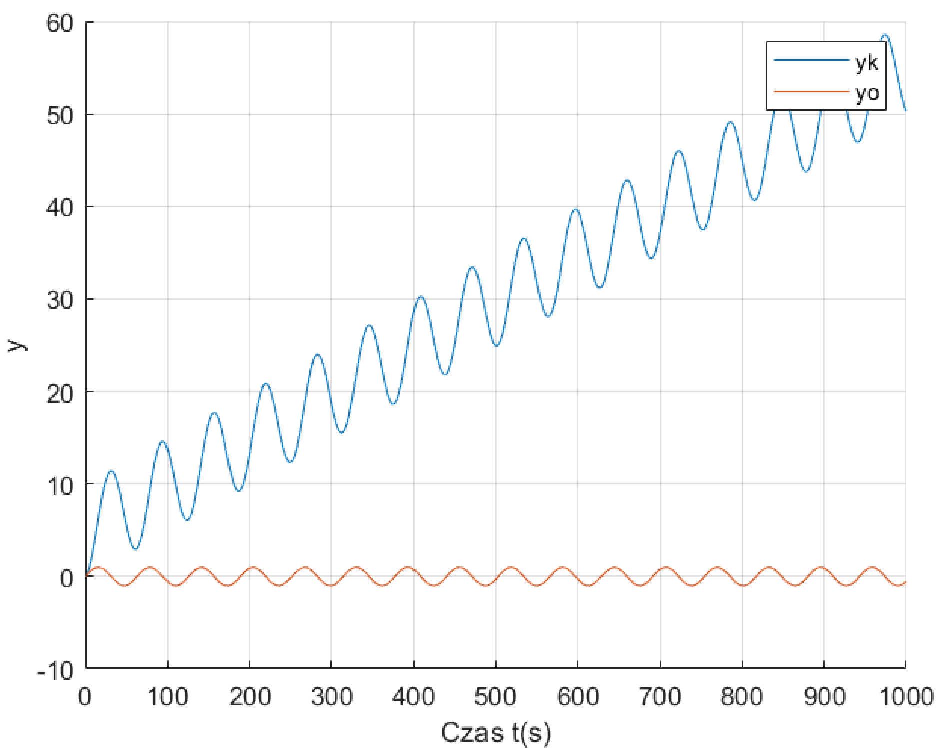
$$P_k = \frac{1}{\lambda}(P_{k-1} - \frac{P_{k-1} * X_k * X_{k-1}^T * P_{k-1}}{\lambda + X_k^T * P_{k-1} * X_k})$$

2 Schemat



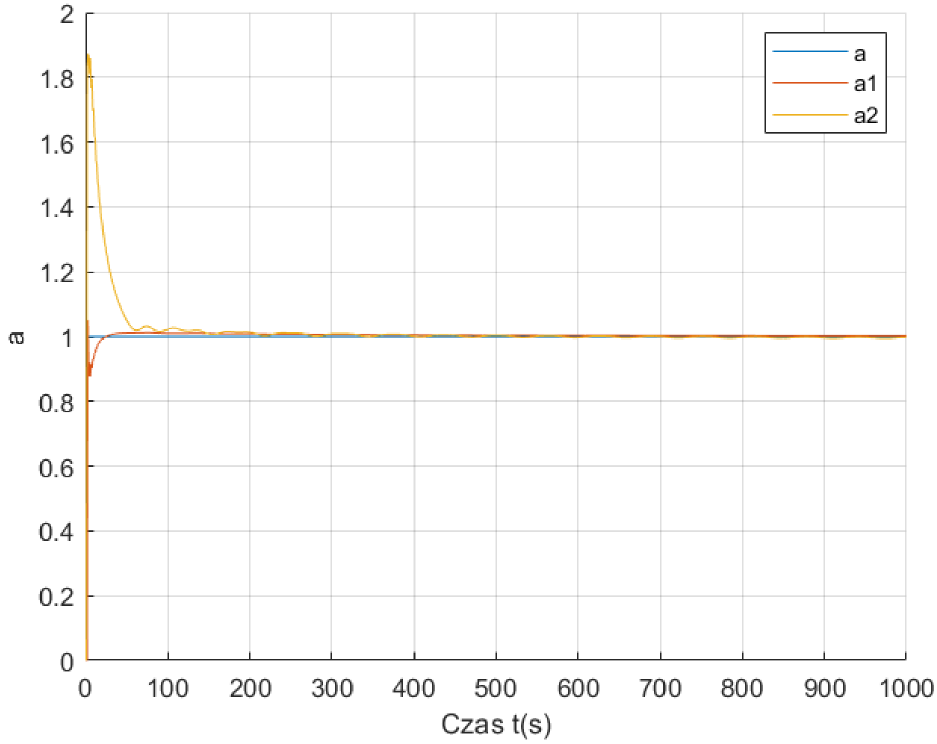
3 Badania symulacyjne UAR dla $a = (1, 1)^T$

3.1 y na tle przebiegu zadanego $y_0 = sin(0.1k)$

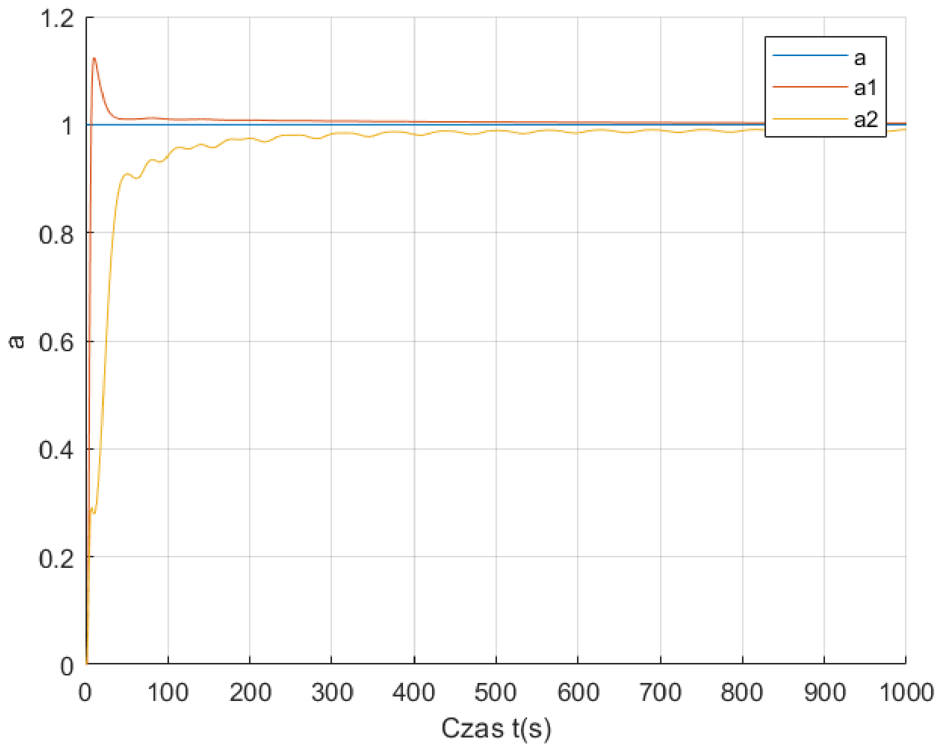


3.2 Zbieżność w czasie estymatora $\hat{a}$ do wartości prawdziwej a

P=[N,0;0,N];

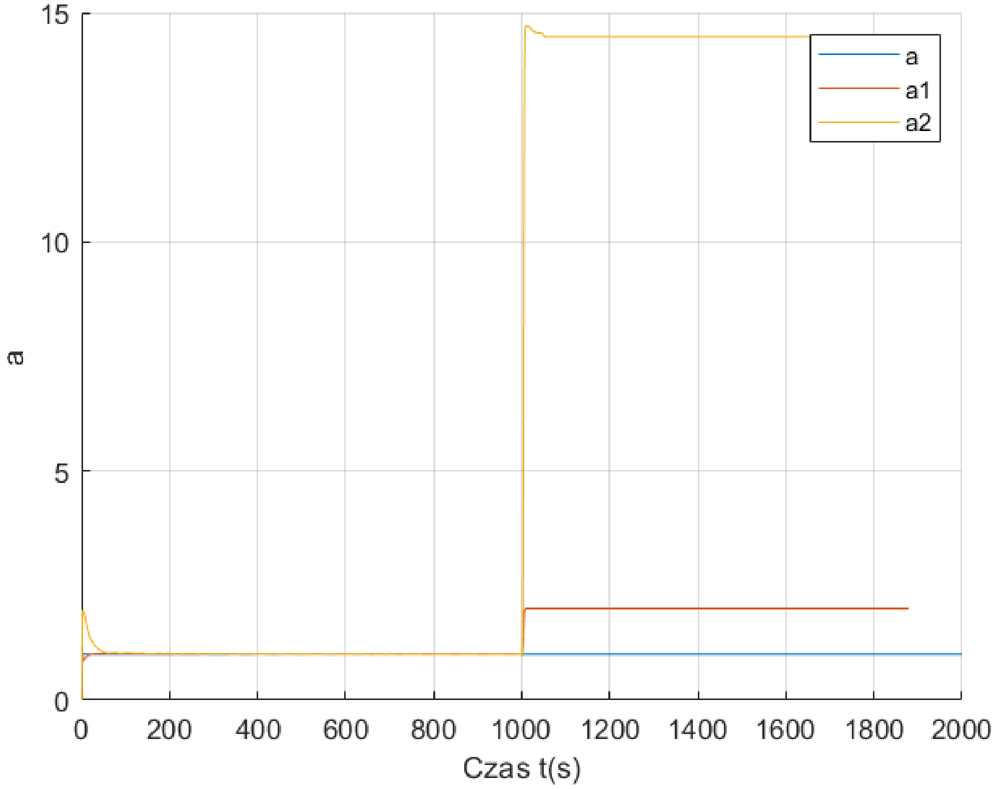


P=eye(2);



Widzimy, że łatwiej do stanu równowagi dochodzi a1, lecz a2 również da się dobrze wyregulować, ale na to potrzeba więcej czasu.

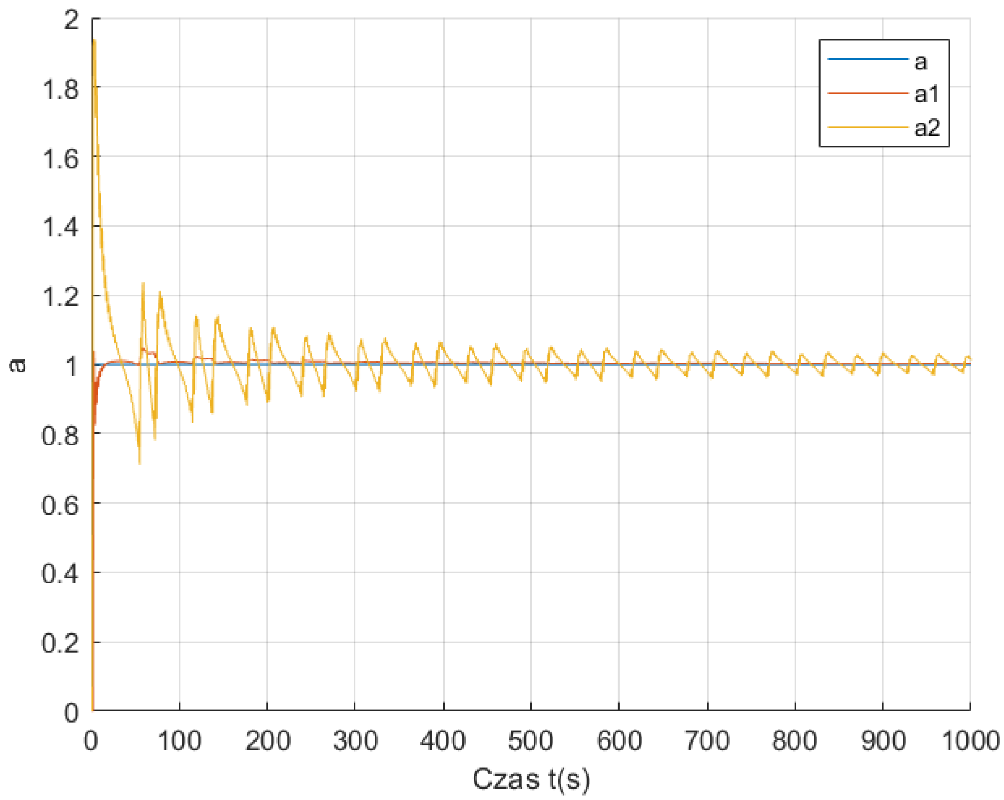
4 Niestacjonarność przy zmianie a_1



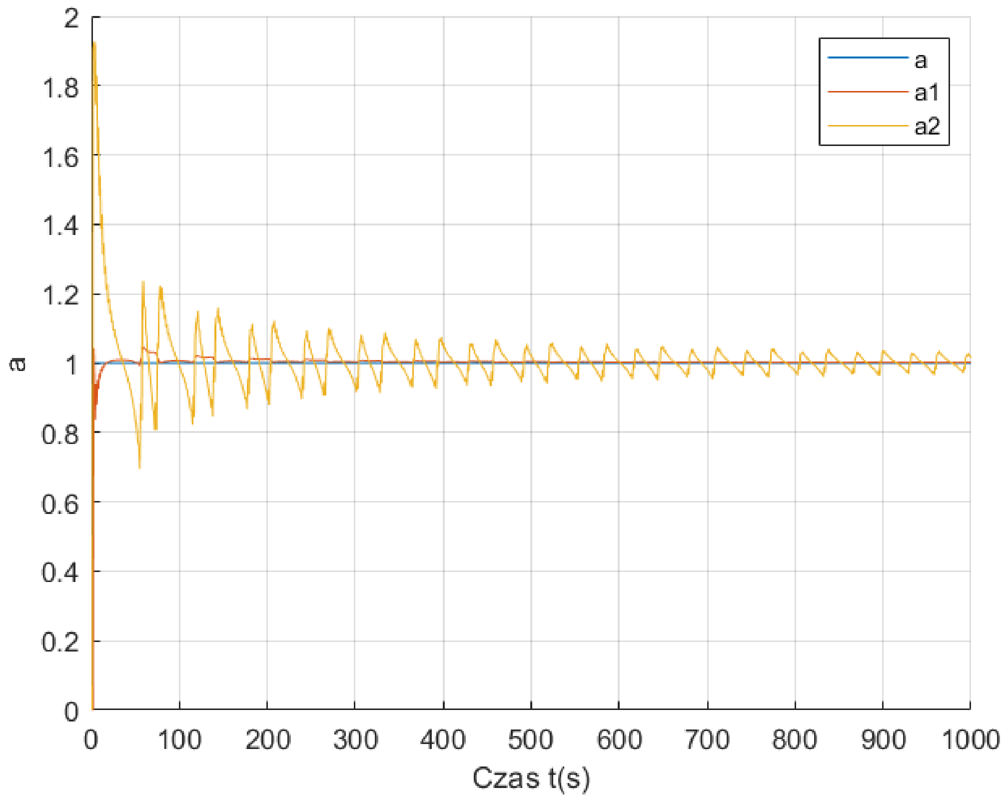
Można zaobserwować utratę stacjonarności. Widzimy, że a1, którego wartość zmieniamy z 1 na 2 w momencie dojścia do 1000 próbek, zmienia się i stabilizuje swoją wartość na ok. 2. Natomiast estymator a2, którego wartości nie zmieniamy, sprawia, że zmiana wartości jest o wiele za duża, przez co mamy błędną identyfikację.

5 Ważona metoda rekurencyjna

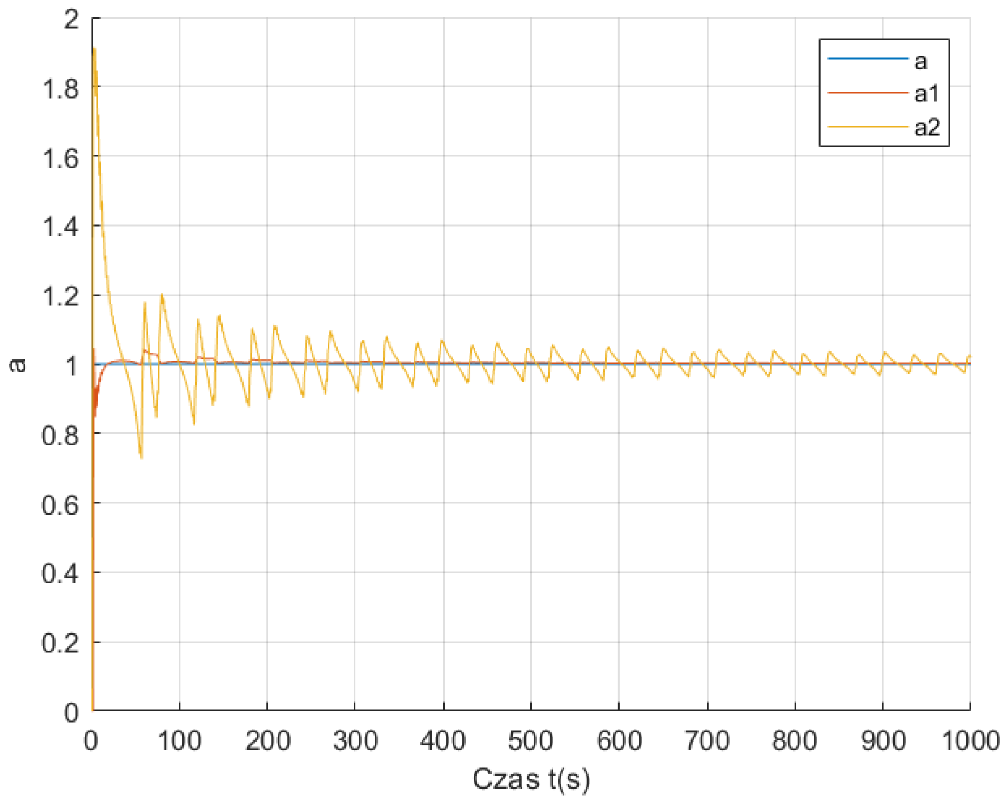
$\lambda = 0.5$



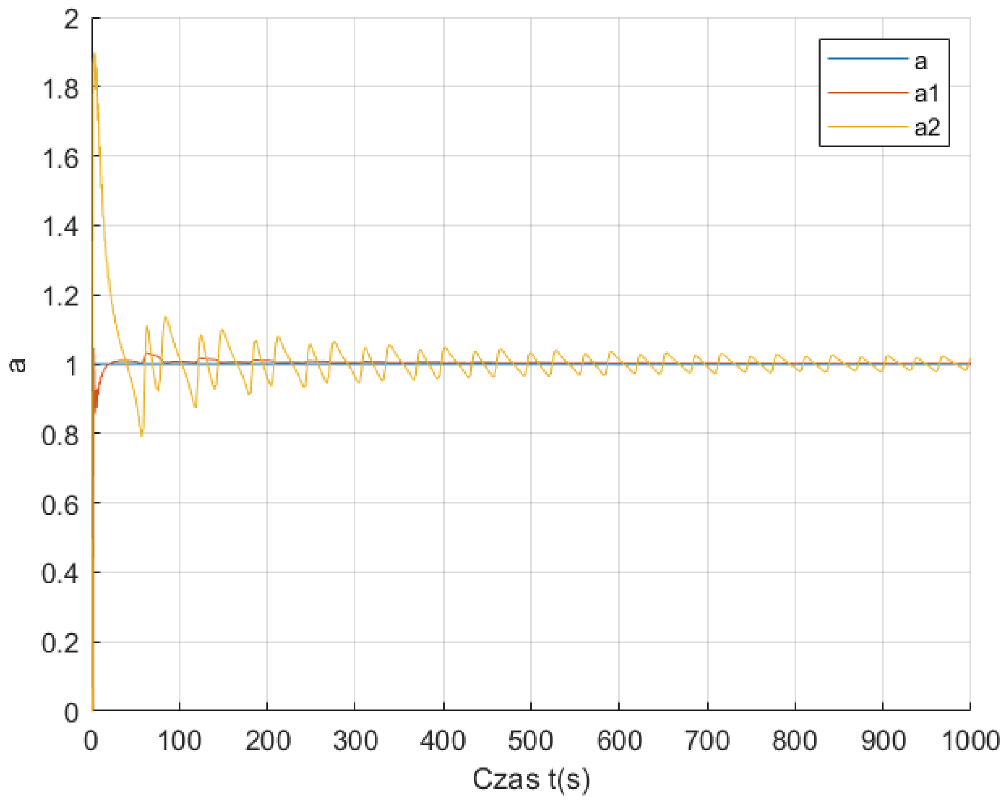
$\lambda = 0.6$



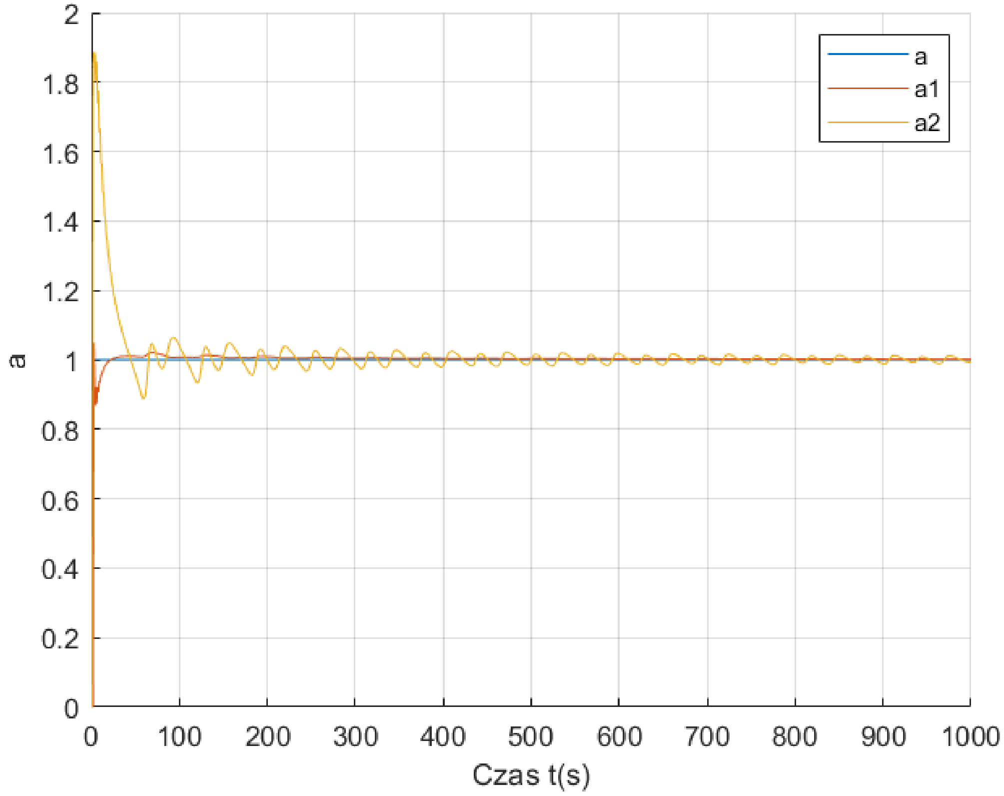
$\lambda = 0.7$



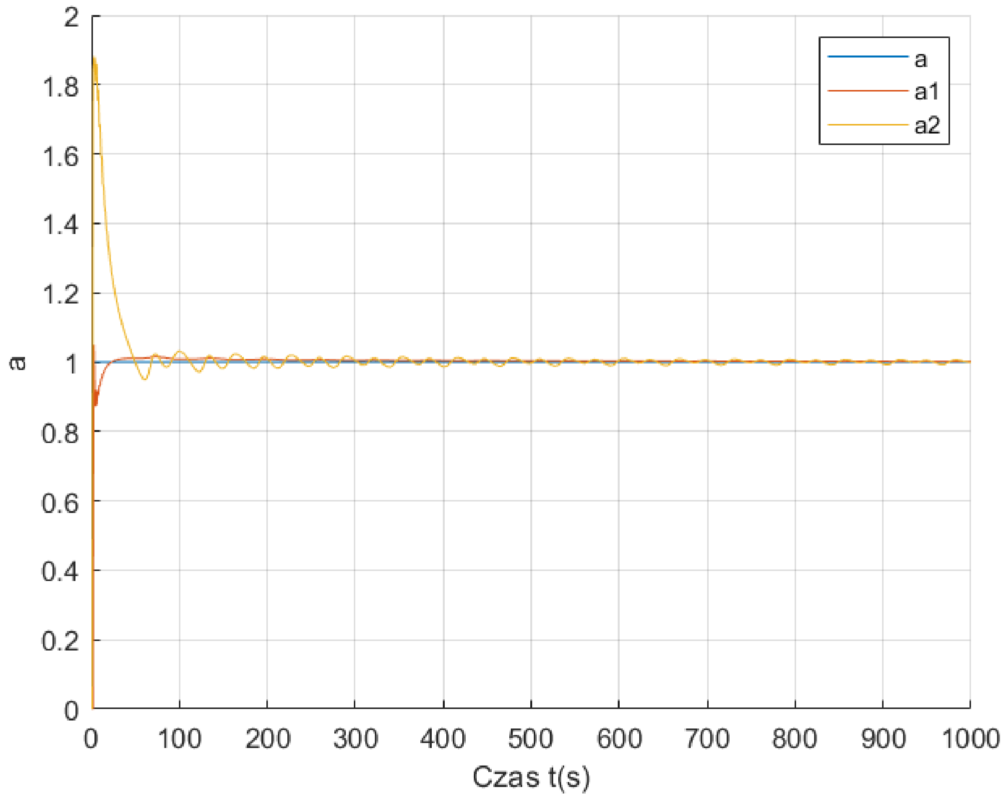
$\lambda = 0.8$



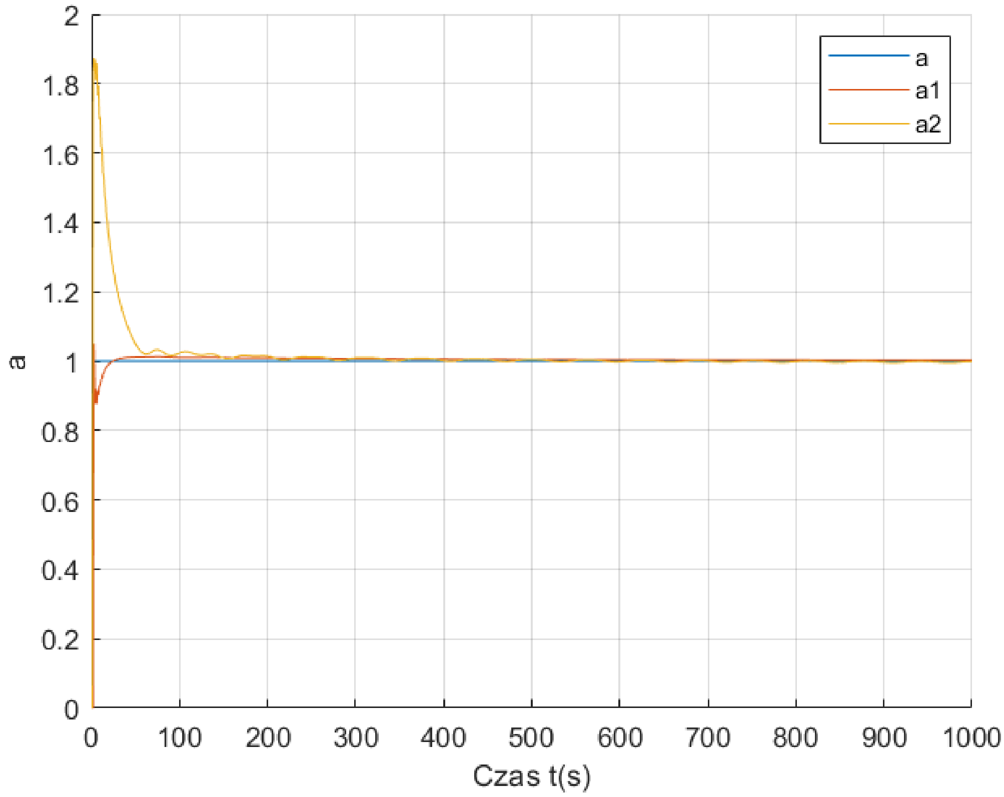
$\lambda = 0.9$



$\lambda = 0.95$



$\lambda = 1$



Widzimy, że wraz ze wzrostem λ stabilizacja przebiega coraz szybciej zarówno dla a1 oraz dla a2. Co ciekawe, przy mniejszych lambdach estymator a2 daje bardzo ostry wykres, a dla większych lambd wykres wydaje się być całkiem wygładzony. W naszym przypadku wyszło, że zapamiętywanie poprzednich wyników jest mniej efektywne niż nie zapamiętywanie, ale to tylko symulacja i nie należy wyciągać tak pochopnych wniosków.

6 Skrypt Matlab'a

```
a=[1;1];
N=1000;
uk=zeros(1,N);
yo=zeros(1,N);
yk=zeros(1,N);
%P=eye(2);
P=[N,0;0,N];
est=[0;0];
zk=0.1;

for k=2:1:N
    yo(k)=(sin(0.1*k));
    uk(k)=(yo(k)-a(2)*uk(k-1)-zk)/a(1);

    phi=[yk(k-1);uk(k)];
    yk(k)=(phi')*a+zk;

    P=P-(P*phi*(phi')*P)/(1+(phi')*P*phi);
    est=est+P*phi*(yk(k)-(phi')*est);
    a1(k)=est(1);
    a2(k)=est(2);
end

t=[1:1:N];
an=ones(1,N);

figure(1)
grid on;
hold on;
plot(t,an);
plot(t,a1);
plot(t,a2);
legend('a','a1','a2');
xlabel('Czas t(s)');
ylabel('a');


figure(2)
grid on;
hold on;
plot(t,yk);
plot(t,yo);
legend('yk','yo');
xlabel('Czas t(s)');
ylabel('y');




%%%%%%%%%%%%%%%%%%%ZAD2
ap=[2;1];
N1=2000;
P=[N1,0;0,N1];
```

```
uk=zeros(1,N1);
yo=zeros(1,N1);
yk=zeros(1,N1);

for k=2:1:N1
    if k<1000
    yo(k)=(sin(0.1*k));
    uk(k)=(yo(k)-a(2)*uk(k-1)-zk)/a(1);

    phi=[yk(k-1);uk(k)];
    yk(k)=(phi')*a+zk;

    P=P-(P*phi*(phi')*P)/(1+(phi')*P*phi);
    est=est+P*phi*(yk(k)-(phi')*est);
    a1(k)=est(1);
    a2(k)=est(2);


    else k>999
    yo(k)=(sin(0.1*k));
    uk(k)=(yo(k)-ap(2)*uk(k-1)-zk)/ap(1);

    phi=[yk(k-1);uk(k)];
    yk(k)=(phi')*ap+zk;

    P=P-(P*phi*(phi')*P)/(1+(phi')*P*phi);
    est=est+P*phi*(yk(k)-(phi')*est);
    a1(k)=est(1);
    a2(k)=est(2);
    end
end

t=[1:1:N1];
an=ones(1,N1);

figure(3)
grid on;
hold on;
plot(t,an);
plot(t,a1);
plot(t,a2);
legend('a','a1','a2');
xlabel('Czas t(s)');
ylabel('a');

figure(4)
grid on;
hold on;
plot(t,yk);
plot(t,yo);
legend('yk','yo');
xlabel('Czas t(s)');
ylabel('y');
```

```
%%%%%%%%%%%%%%%%%%%%ZAD3
a=[1;1];
N=1000;
uk=zeros(1,N);
yo=zeros(1,N);
yk=zeros(1,N);
%P=eye(2);
P=[N,0;0,N];
est=[0;0];
zk=0.1;
alpha=0.95;

for k=2:1:N
    yo(k)=(sin(0.1*k));
    uk(k)=(yo(k)-a(2)*uk(k-1)-zk)/a(1);

    phi=[yk(k-1);uk(k)];
    yk(k)=(phi')*a+zk;

    P=1/alpha*(P-(P*phi*(phi')*P)/(alpha+(phi')*P*phi));
    est=est+P*phi*(yk(k)-(phi')*est);
    a11(k)=est(1);
    a22(k)=est(2);
end

t=[1:1:N];
an=ones(1,N);

figure(5)
grid on;
hold on;
plot(t,an);
plot(t,a11);
plot(t,a22);
legend('a','a1','a2');
xlabel('Czas t(s)');
ylabel('a');


figure(6)
grid on;
hold on;
plot(t,yk);
plot(t,yo);
legend('yk','yo');
xlabel('Czas t(s)');
ylabel('y');
```