

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.

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

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

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

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

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:

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

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

## 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)

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

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

### 1.3 Ważona metoda rekurencyjna

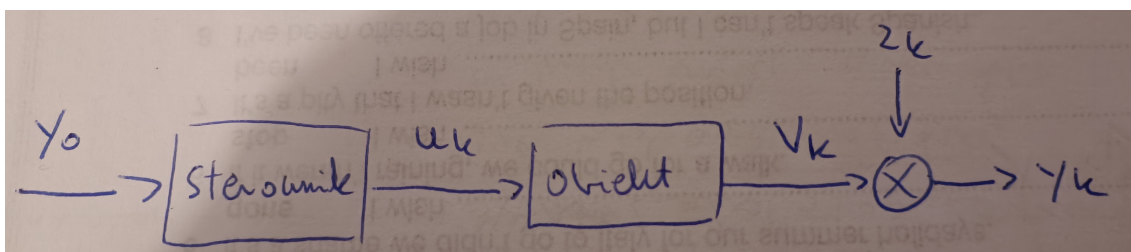
W tej metodzie wyznaczamy przybliżenia wartości  $a_0$  i  $a_1$  korzystając z próbki czasu teraźniejszego oraz przeszłego, ale dodatkowo korzystamy z współczynnika zapomnienia  $\lambda \in (0, 1]$ . Im próbka młodsza tym ma większą wagę. Tutaj też trzeba wprowadzić macierz kowariancji  $P_k$ .

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

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

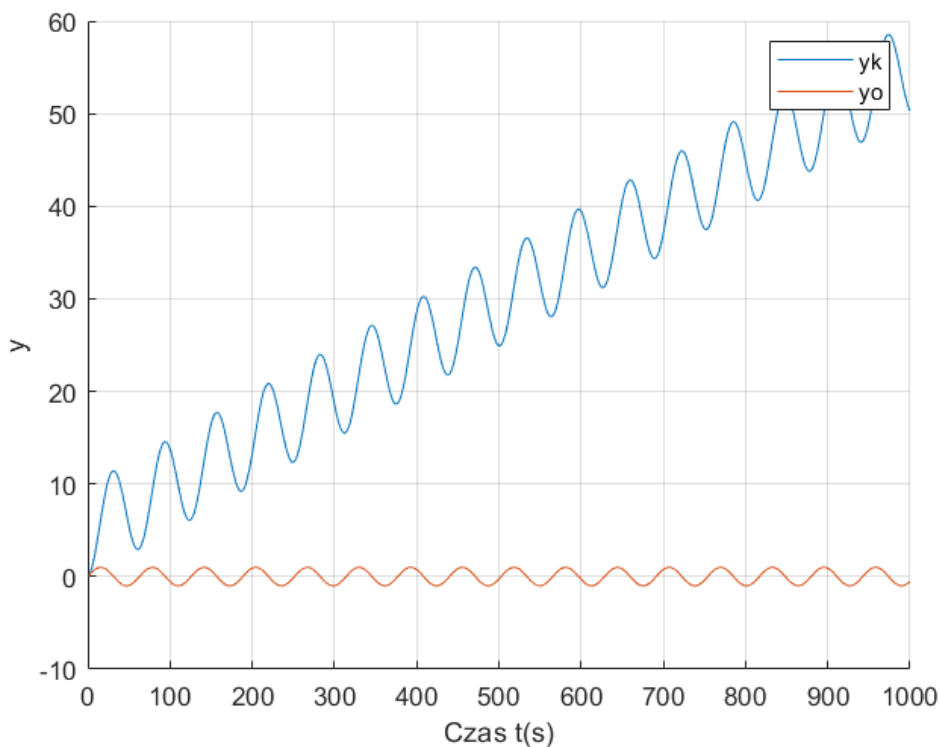
$$P_k = \frac{1}{\lambda} \left( P_{k-1} - \frac{P_{k-1} * X_k * X_k^T * P_{k-1}}{\lambda + X_k^T * P_{k-1} * X_k} \right)$$

## 2 Schemat



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

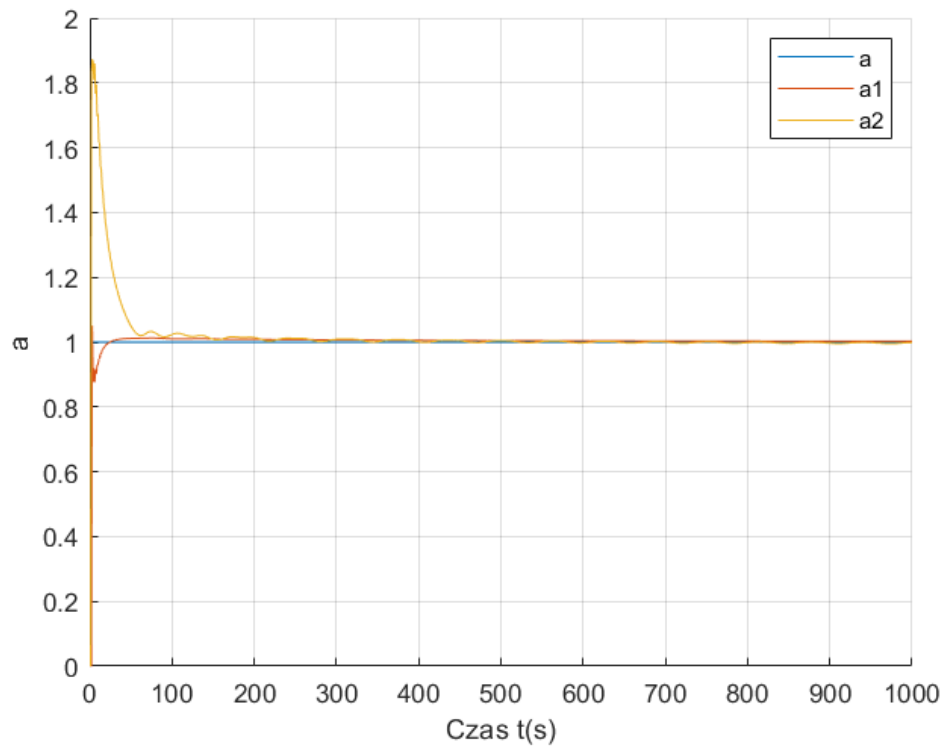
### 3.1 $y$ na tle przebiegu zadanego $y_0 = \sin(0.1k)$



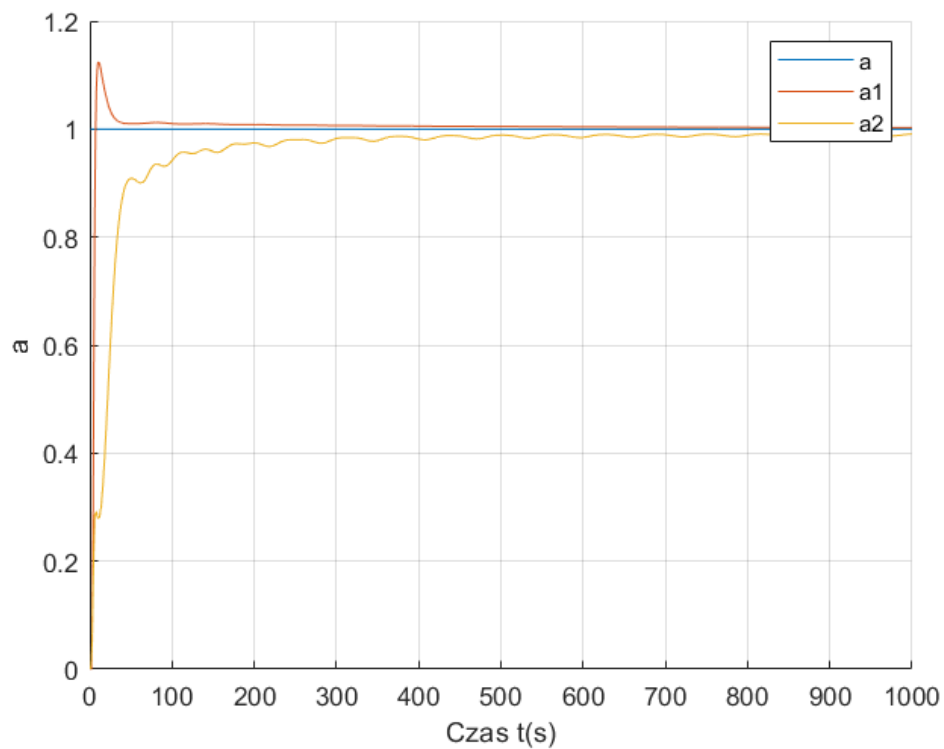
### 3.2 Zbieżność w czasie estymatora $\hat{a}$ do wartości prawdziwej

$a$

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

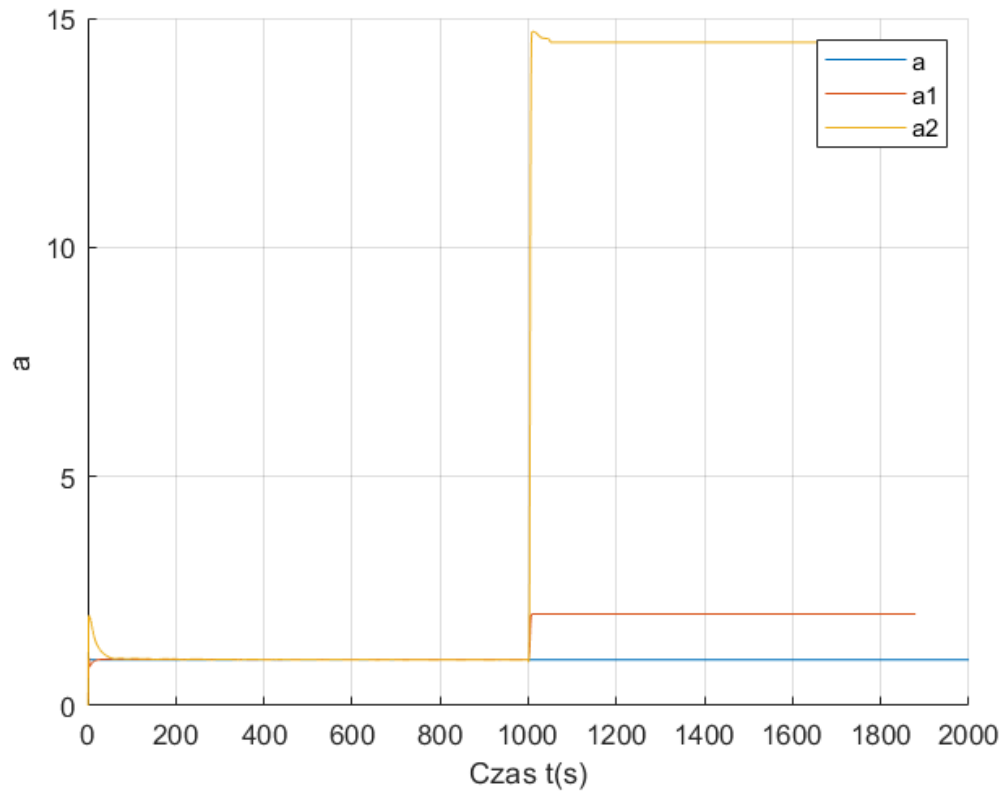


$P=\text{eye}(2);$



Widzimy, że łatwiej do stanu równowagi dochodzi  $a_1$ , lecz  $a_2$  również da się dobrze wyregulować, ale na to potrzeba więcej czasu.

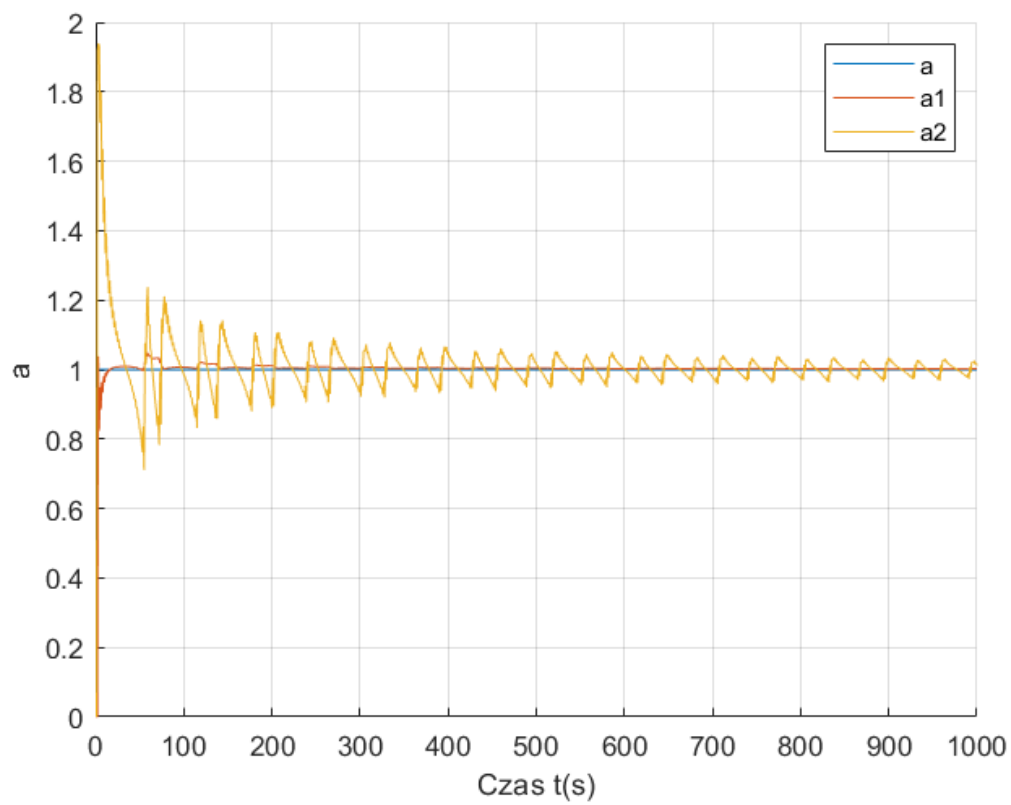
## 4 Niestacjonarność przy zmianie $a_1$



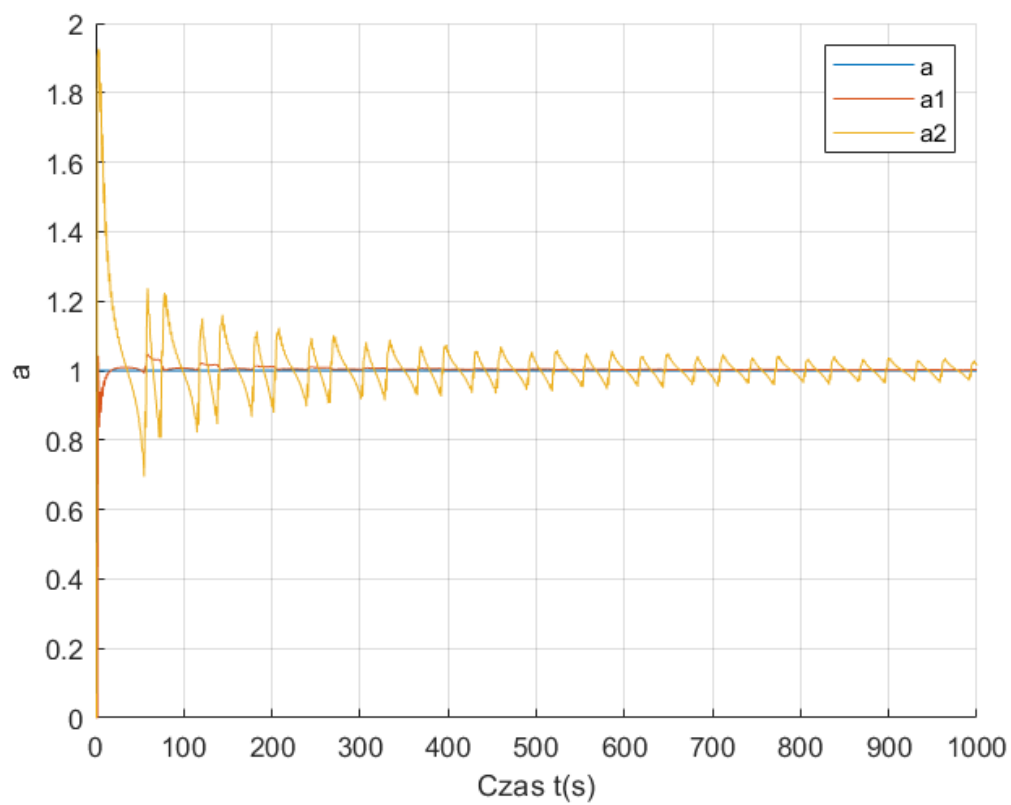
Można zaobserwować utratę stacjonarności. Widzimy, że  $a_1$ , którego wartość zmieniamy z 1 na 2 w momencie dojścia do 1000 próbek, zmienia się i stabilizuje swoją wartość na ok. 2. Natomiast estymator  $a_2$ , którego wartości nie zmieniamy, sprawia, że zmiana wartości jest o wiele za duża, przez co mamy błędną identyfikację.

## 5 Ważona metoda rekurencyjna

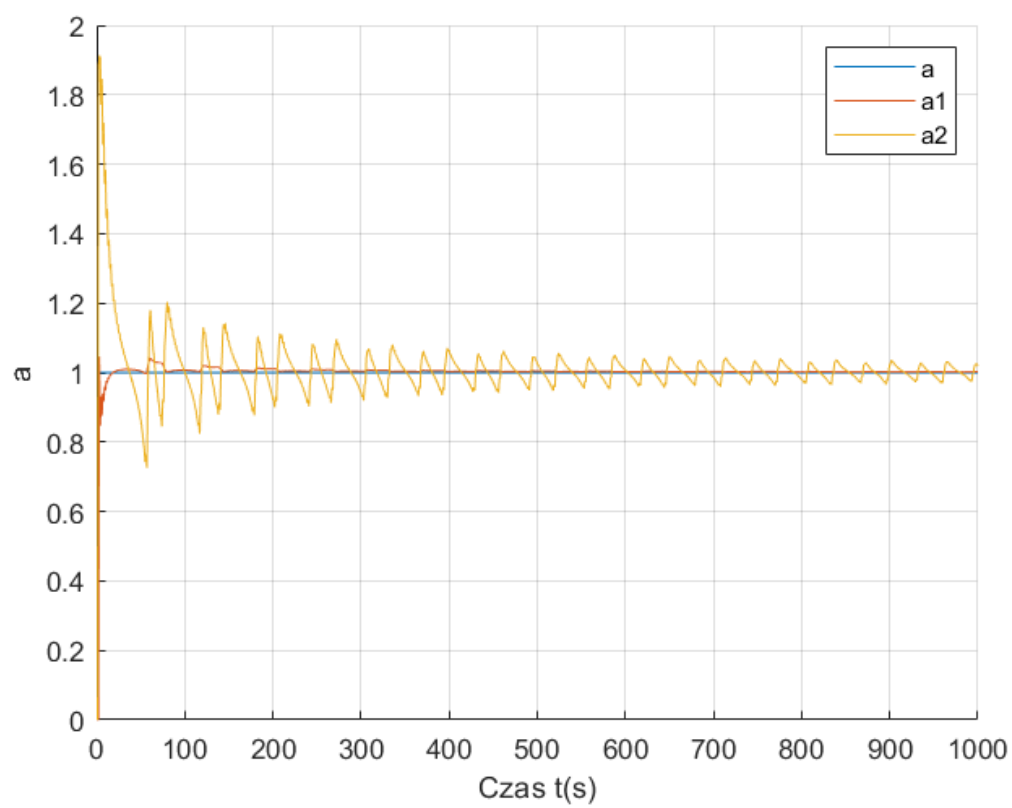
$$\lambda = 0.5$$



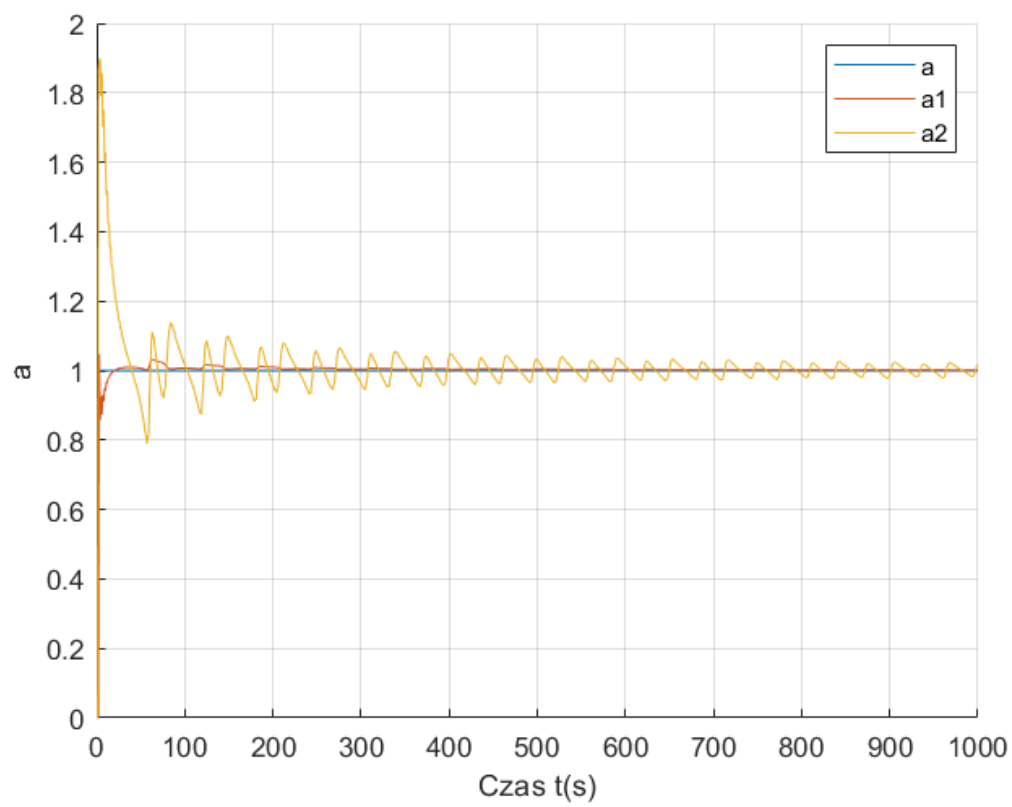
$$\lambda = 0.6$$



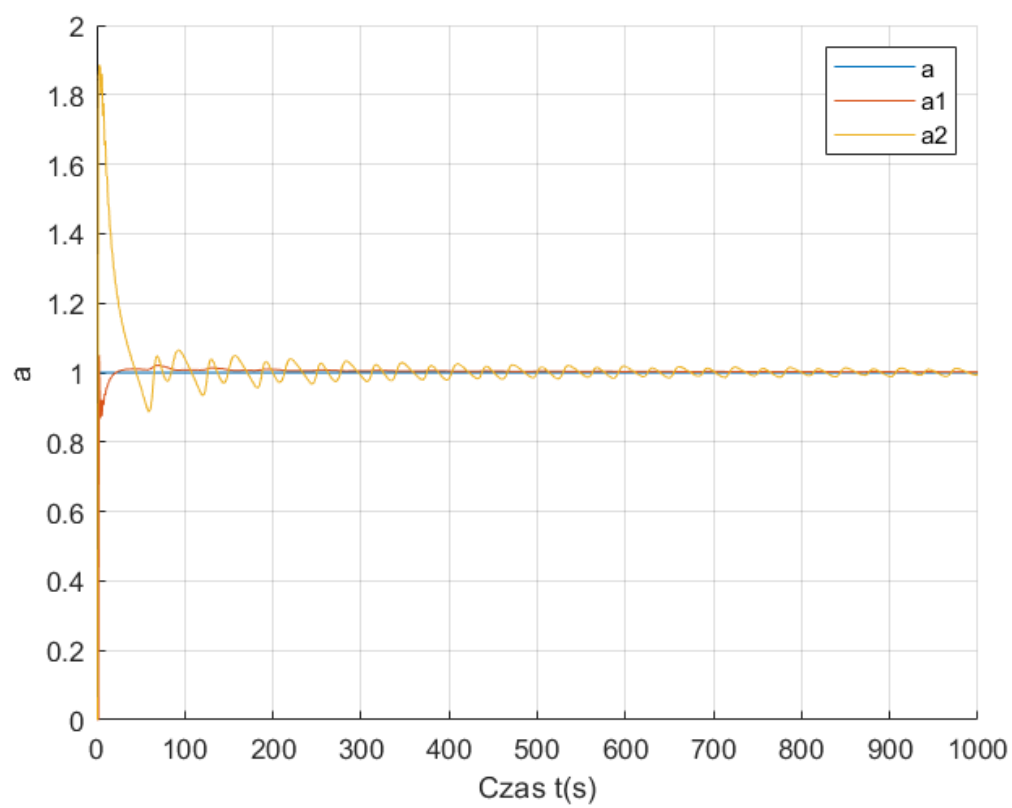
$$\lambda = 0.7$$



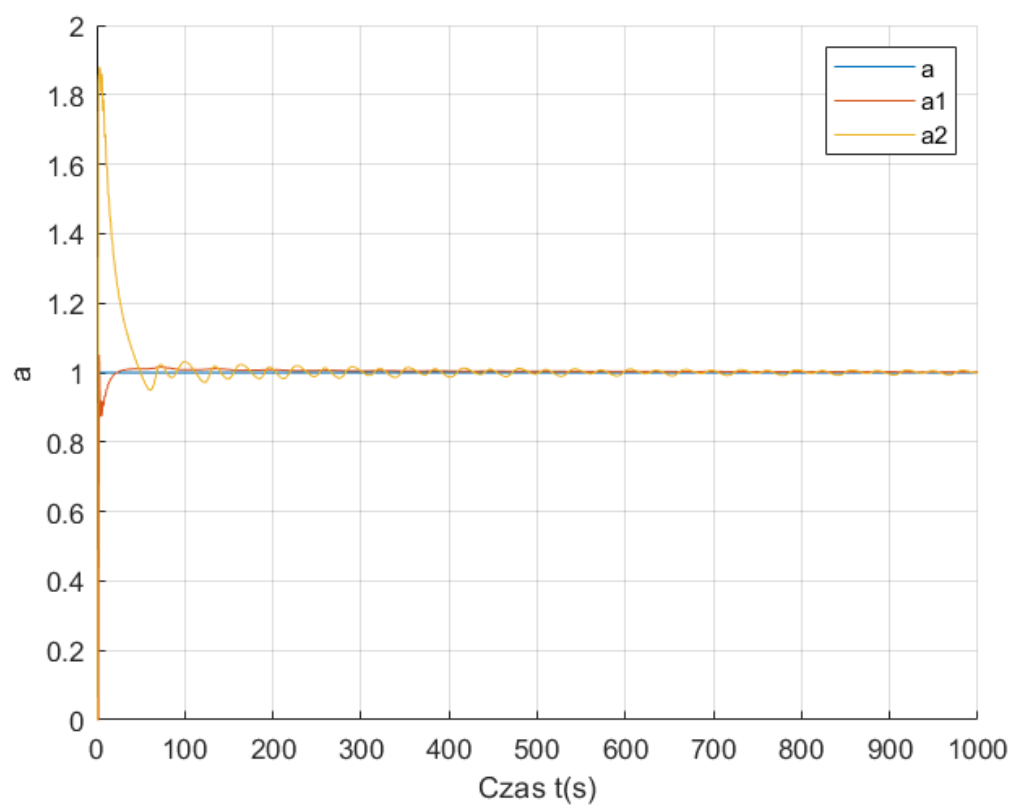
$$\lambda = 0.8$$



$$\lambda = 0.9$$

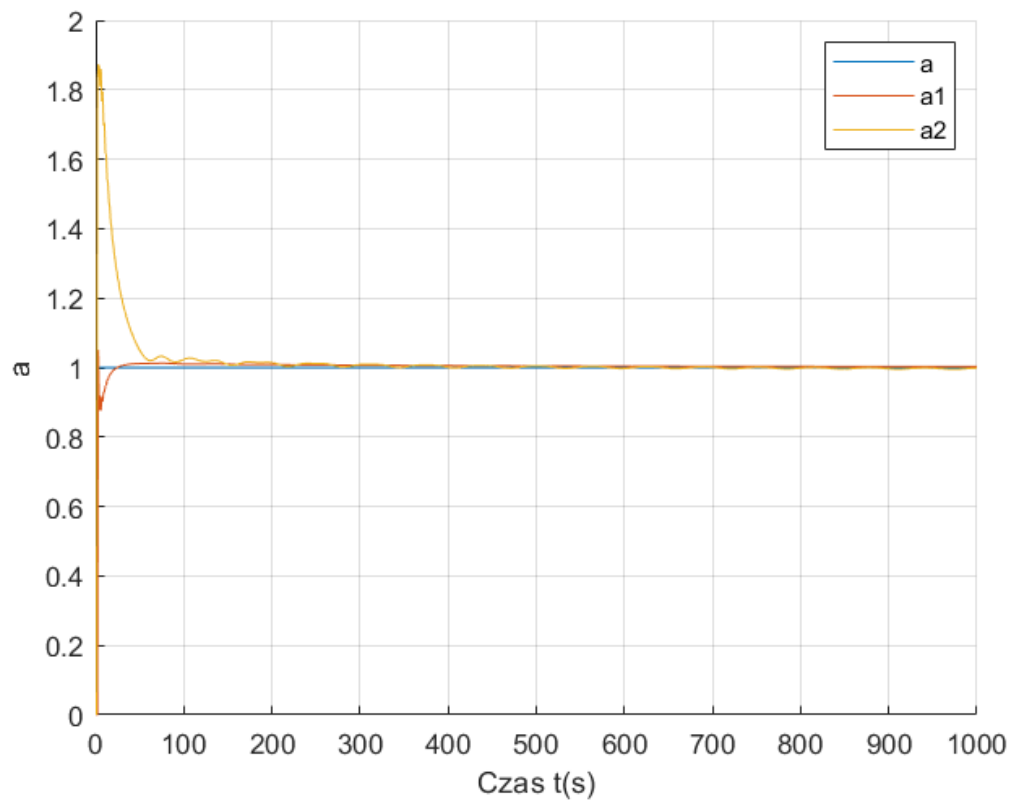


$$\lambda = 0.95$$





$$\lambda = 1$$



Widzimy, że wraz ze wzrostem  $\lambda$  stabilizacja przebiega coraz szybciej zarówno dla  $a1$  oraz dla  $a2$ . Co ciekawe, przy mniejszych  $\lambda$  estymator  $a2$  daje bardzo ostry wykres, a dla większych  $\lambda$  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

```
1 a=[1;1];
2 N=1000;
3 uk=zeros(1,N);
4 yo=zeros(1,N);
5 yk=zeros(1,N);
6 %P=eye(2);
7 P=[N,0;0,N];
8 est=[0;0];
9 zk=0.1;
10
11 for k=2:1:N
12     yo(k)=(sin(0.1*k));
13     uk(k)=(yo(k)-a(2)*uk(k-1)-zk)/a(1);
14
15     phi=[yk(k-1);uk(k)];
16     yk(k)=(phi')*a+zk;
17
18     P=P-(P*phi*(phi')*P)/(1+(phi')*P*phi);
19     est=est+P*phi*(yk(k)-(phi')*est);
20     a1(k)=est(1);
21     a2(k)=est(2);
22 end
23
24 t=[1:1:N];
25 an=ones(1,N);
26
27 figure(1)
28 grid on;
29 hold on;
30 plot(t,an);
31 plot(t,a1);
32 plot(t,a2);
33 legend('a','a1','a2');
34 xlabel('Czas t(s)');
35 ylabel('a');
36
37
38 figure(2)
39 grid on;
40 hold on;
41 plot(t,yk);
42 plot(t,yo);
43 legend('yk','yo');
44 xlabel('Czas t(s)');
45 ylabel('y');
46
47
48
49
50 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%ZAD2
51 ap=[2;1];
52 N1=2000;
53 P=[N1,0;0,N1];
```

```

54 uk=zeros(1,N1);
55 yo=zeros(1,N1);
56 yk=zeros(1,N1);
57
58 for k=2:1:N1
59     if k<1000
60         yo(k)=(sin(0.1*k));
61         uk(k)=(yo(k)-a(2)*uk(k-1)-zk)/a(1);
62
63         phi=[yk(k-1);uk(k)];
64         yk(k)=(phi')*a+zk;
65
66         P=P-(P*phi*(phi')*P)/(1+(phi')*P*phi);
67         est=est+P*phi*(yk(k)-(phi')*est);
68         a1(k)=est(1);
69         a2(k)=est(2);
70
71
72     else k>999
73         yo(k)=(sin(0.1*k));
74         uk(k)=(yo(k)-ap(2)*uk(k-1)-zk)/ap(1);
75
76         phi=[yk(k-1);uk(k)];
77         yk(k)=(phi')*ap+zk;
78
79         P=P-(P*phi*(phi')*P)/(1+(phi')*P*phi);
80         est=est+P*phi*(yk(k)-(phi')*est);
81         a1(k)=est(1);
82         a2(k)=est(2);
83     end
84 end
85
86 t=[1:1:N1];
87 an=ones(1,N1);
88
89 figure(3)
90 grid on;
91 hold on;
92 plot(t,an);
93 plot(t,a1);
94 plot(t,a2);
95 legend('a','a1','a2');
96 xlabel('Czas t(s)');
97 ylabel('a');
98
99 figure(4)
100 grid on;
101 hold on;
102 plot(t,yk);
103 plot(t,yo);
104 legend('yk','yo');
105 xlabel('Czas t(s)');
106 ylabel('y');
107
108

```

```

109
110
111
112 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%ZAD3
113 a=[1;1];
114 N=1000;
115 uk=zeros(1,N);
116 yo=zeros(1,N);
117 yk=zeros(1,N);
118 %P=eye(2);
119 P=[N,0;0,N];
120 est=[0;0];
121 zk=0.1;
122 alpha=0.95;
123
124 for k=2:1:N
125     yo(k)=(sin(0.1*k));
126     uk(k)=(yo(k)-a(2)*uk(k-1)-zk)/a(1);
127
128     phi=[yk(k-1);uk(k)];
129     yk(k)=(phi')*a+zk;
130
131     P=1/alpha*(P-(P*phi*(phi')*P)/(alpha+(phi')*P*phi));
132     est=est+P*phi*(yk(k)-(phi')*est);
133     a11(k)=est(1);
134     a22(k)=est(2);
135 end
136
137 t=[1:1:N];
138 an=ones(1,N);
139
140 figure(5)
141 grid on;
142 hold on;
143 plot(t,an);
144 plot(t,a11);
145 plot(t,a22);
146 legend('a','a1','a2');
147 xlabel('Czas t(s)');
148 ylabel('a');
149
150
151 figure(6)
152 grid on;
153 hold on;
154 plot(t,yk);
155 plot(t,yo);
156 legend('yk','yo');
157 xlabel('Czas t(s)');
158 ylabel('y');

```