

Wykorzystanie metody Chien-Hrones-Reswicka w strojeniu regulatora PID

Autor:

Jakub Ściga

Promotor:

dr inż. Krzysztof Lalik

Zakres pracy

- » Wstęp teoretyczny dotyczący działania układów automatycznej regulacji
- » Przegląd różnych metod doboru nastaw regulatora PID
- » Symulacja metody Chien-Hrones-Reswicka w programie MATLAB/Simulink
- » Implementacja algorytmu autoregulacji na sterowniku PLC

Metody doboru nastaw PID

- » Metoda Zieglera-Nicholsa
- » Metoda Hassena-Offereissena
- » Metoda Cohena-Coona
- » Metoda Strejca
- » Metoda Chiena-Hronesa-Reswicka

Metoda Zieglera-Nicholsa

Parametry	K_p	T_i	T_d
P	$0,5 K_{kr}$	—	—
PI	$0,45 K_{kr}$	$0,85 T_{osc}$	—
PID	$0,6 K_{kr}$	$0,5 T_{osc}$	$0,125 T_{osc}$

Rys. 1. Dobór nastaw regulatora PID wg metody Zieglera-Nicholsa

Metoda Hassena-Offereissena

Parametry	K_p	T_i	T_d
P	$0,45 K_{kr}$	—	—
PI	$0,45 K_{kr}$	$3 T_{i \text{ kryt}}$	—
PID	$0,45 K_{kr}$	$4,5 T_{d \text{ max}}$	$0,3 T_{d \text{ max}}$

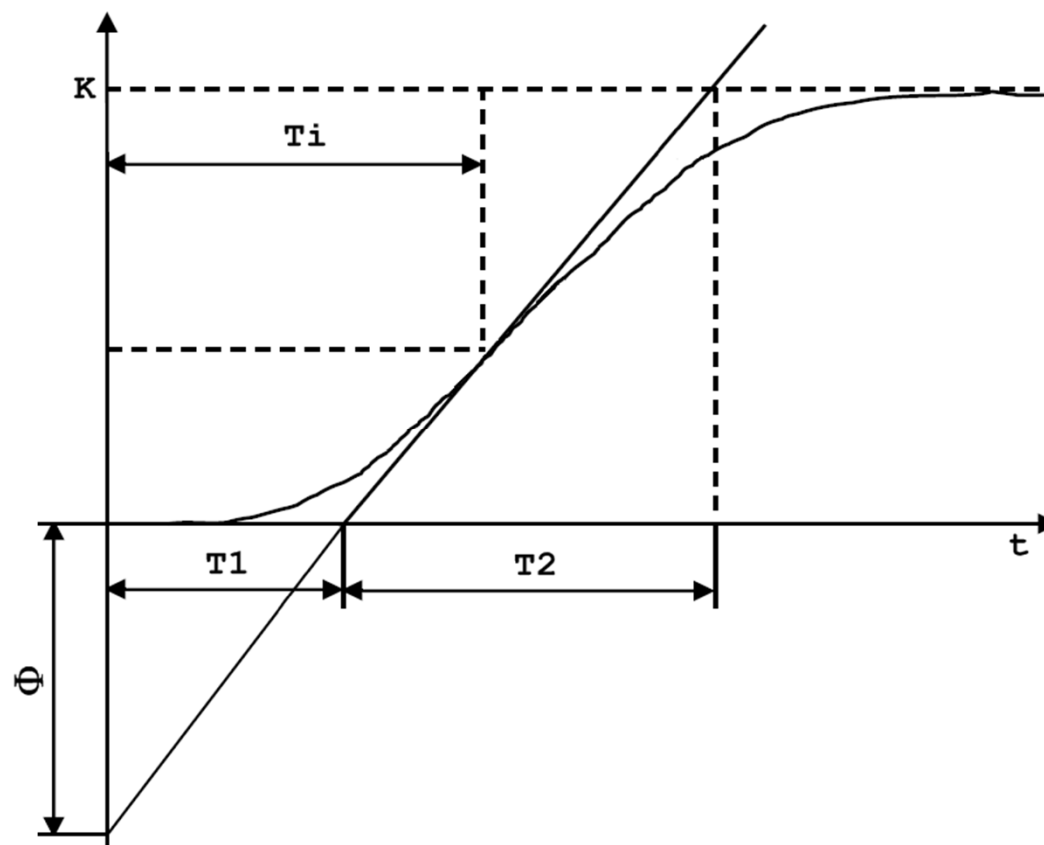
Rys. 2. Dobór nastaw regulatora PID wg metody Hassena-Offereissena

Metoda Cohena-Coona

Parametry	K_p	T_i	T_d
P	$\frac{1}{K_{kr}} (1 + \frac{r}{3})$	—	—
PI	$\frac{1}{K_{kr}} (0,9 + \frac{r}{12})$	$t_d \frac{30 + 3r}{9 + 20r}$	—
PID	$\frac{1}{K_{kr}} (\frac{4}{3} + \frac{r}{4})$	$t_d \frac{32 + 6r}{13 + 8r}$	$t_d \frac{4}{13 + 2r}$

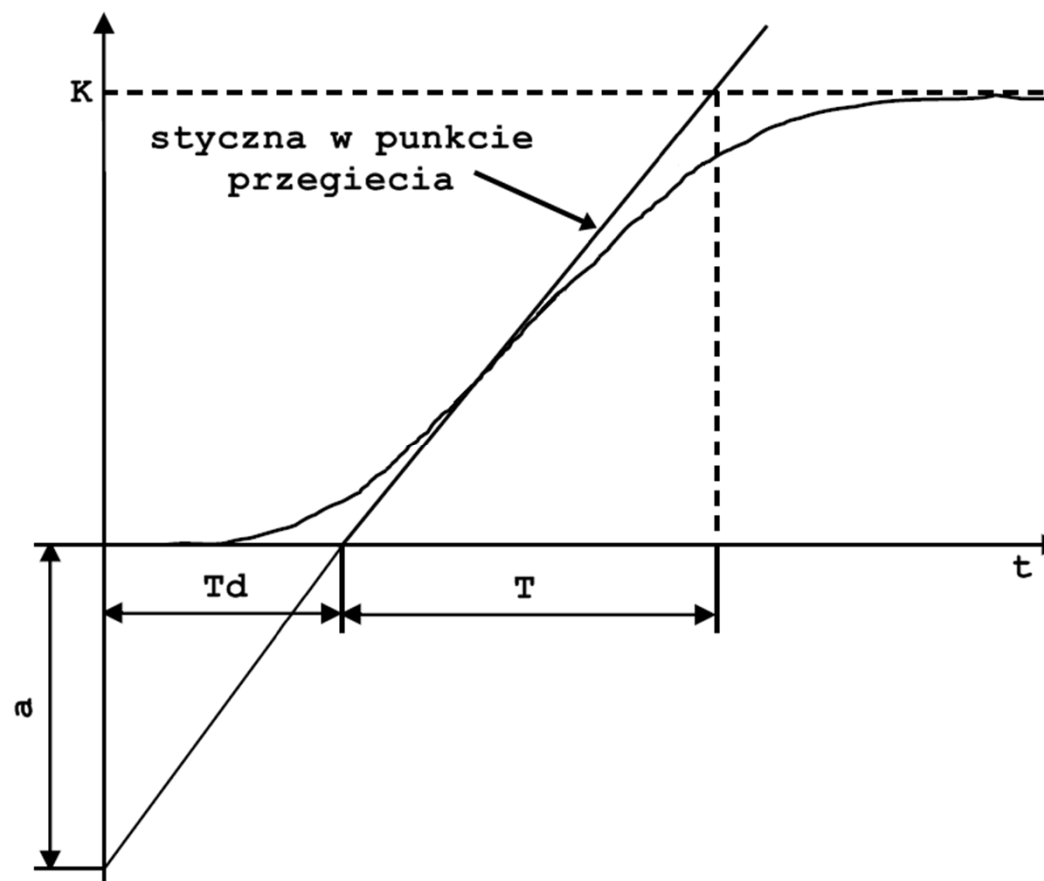
Rys. 3. Dobór nastaw regulatora PID wg metody Cohena-Coona

Metoda Strejca



Rys. 4. Wyznaczenie parametrów pośrednich regulatora PID wg metody Strejca

Metoda Chien-Hrones-Reswicka



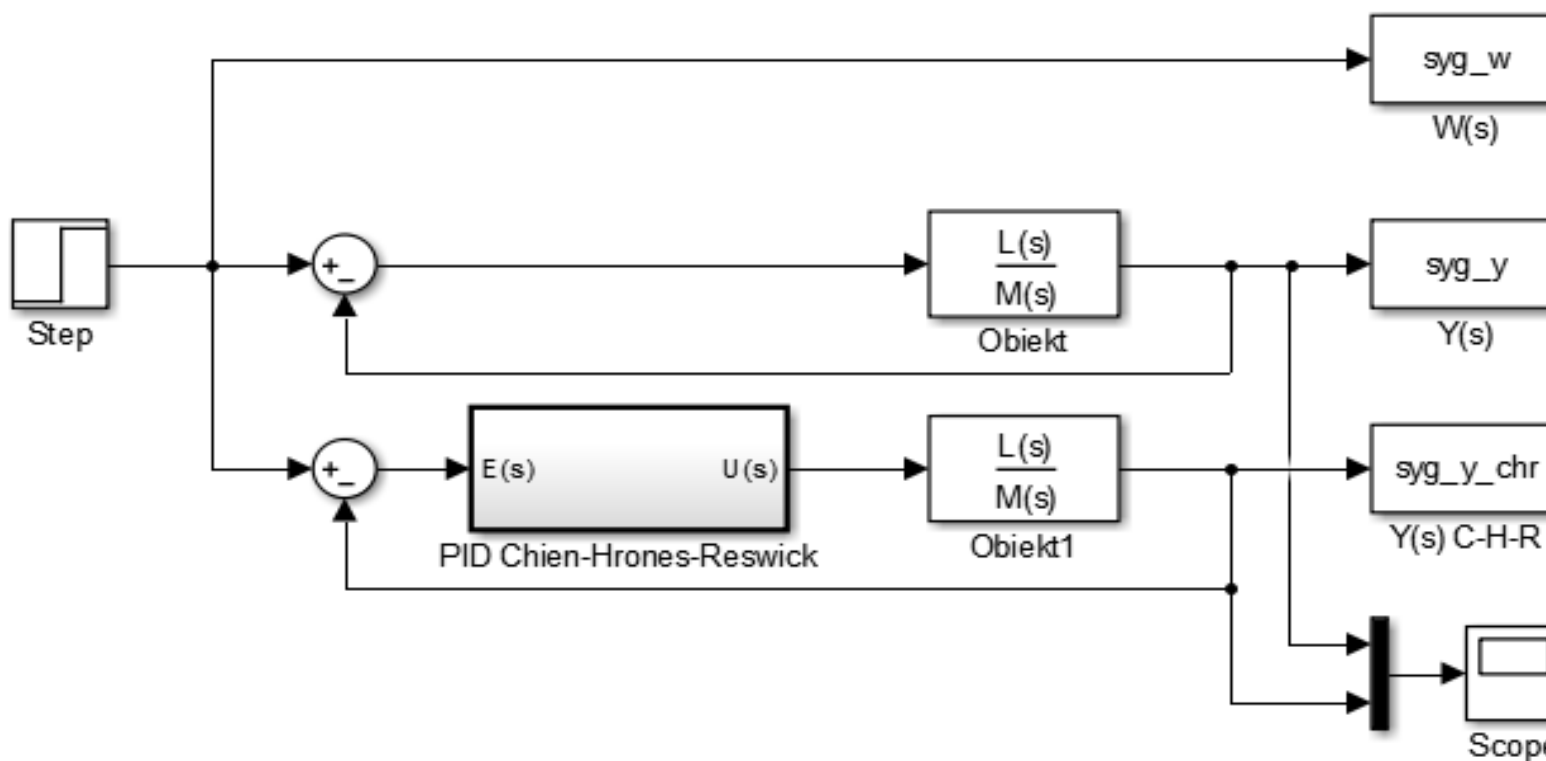
Rys. 5. Wyznaczenie parametrów pośrednich regulatora PID wg metody Chien-Hrones-Reswicka

Metoda Chien-Hrones-Reswicka

Przeregulowanie	0 %			20 %		
Parametry	K_p	T_i	T_d	K_p	T_i	T_d
P	$0,3 \frac{1}{a}$	—	—	$0,7 \frac{1}{a}$	—	—
PI	$0,35 \frac{1}{a}$	$1,2 T$	—	$0,6 \frac{1}{a}$	T	—

Rys. 6. Dobór nastaw regulatora PID wg metody Chien-Hrones-Reswicka

Symulacja w programie MATLAB



Rys. 7. Układ automatycznej regulacji zbudowany za pomocą pakietu Simulink

Symulacja w programie MATLAB

```
%% Obliczanie parametrow regulacji

y1=dy(I_dy)*(0-x(I_dy))+y(I_dy);
x1=(0-y(I_dy))/dy(I_dy)+x(I_dy);
x2=(K-y(I_dy))/dy(I_dy)+x(I_dy);

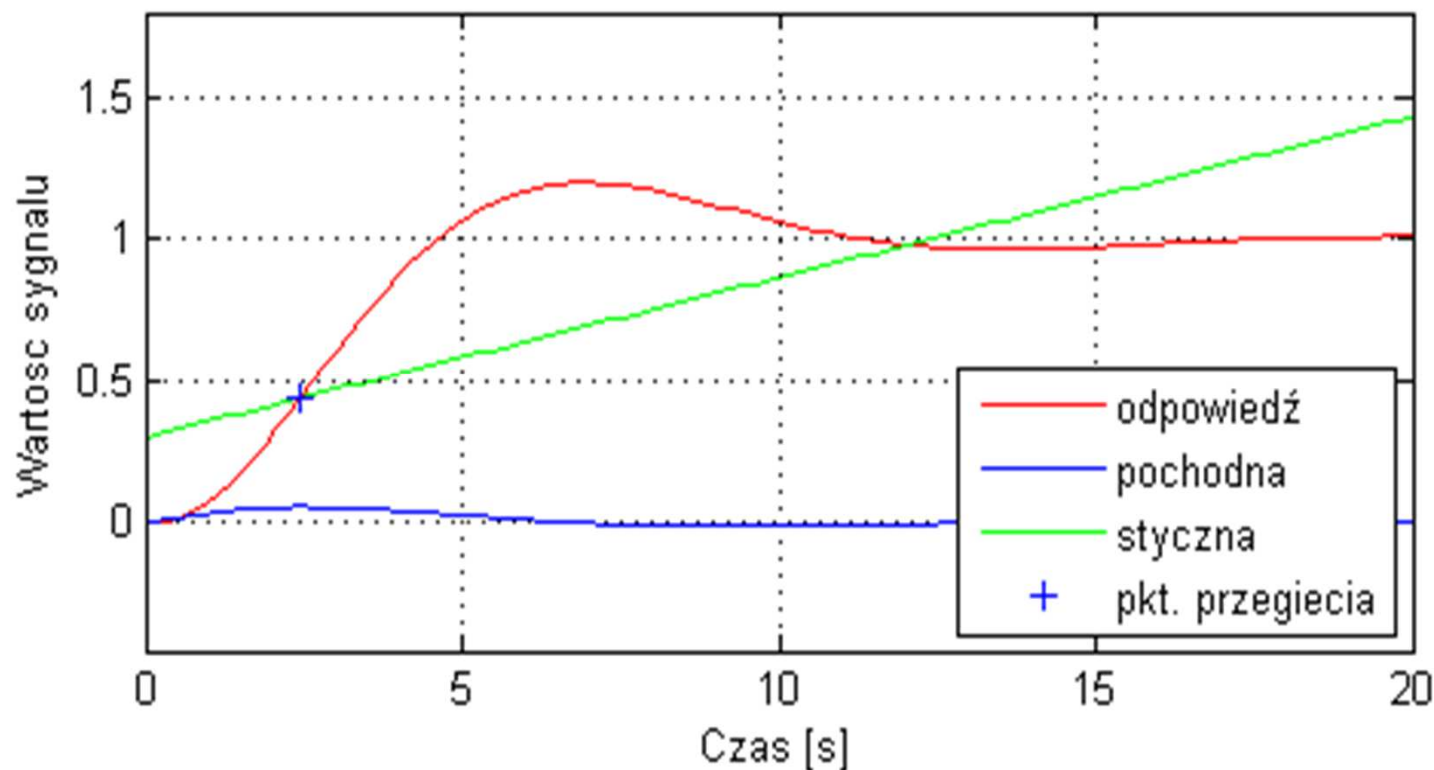
ta=abs(y1);           %wyraz wolny
td=abs(x1);           %odległość stycznej na wys. 0
to=abs(x2-x1);        %odległość stycznej na wys. K

%% Dobor nastawow PID CHR

Kp=0.6/ta;            %wartość wzmacnienia
Ti=1*to;              %stała całkowania
Td=0.5*td;            %stała różniczkująca
```

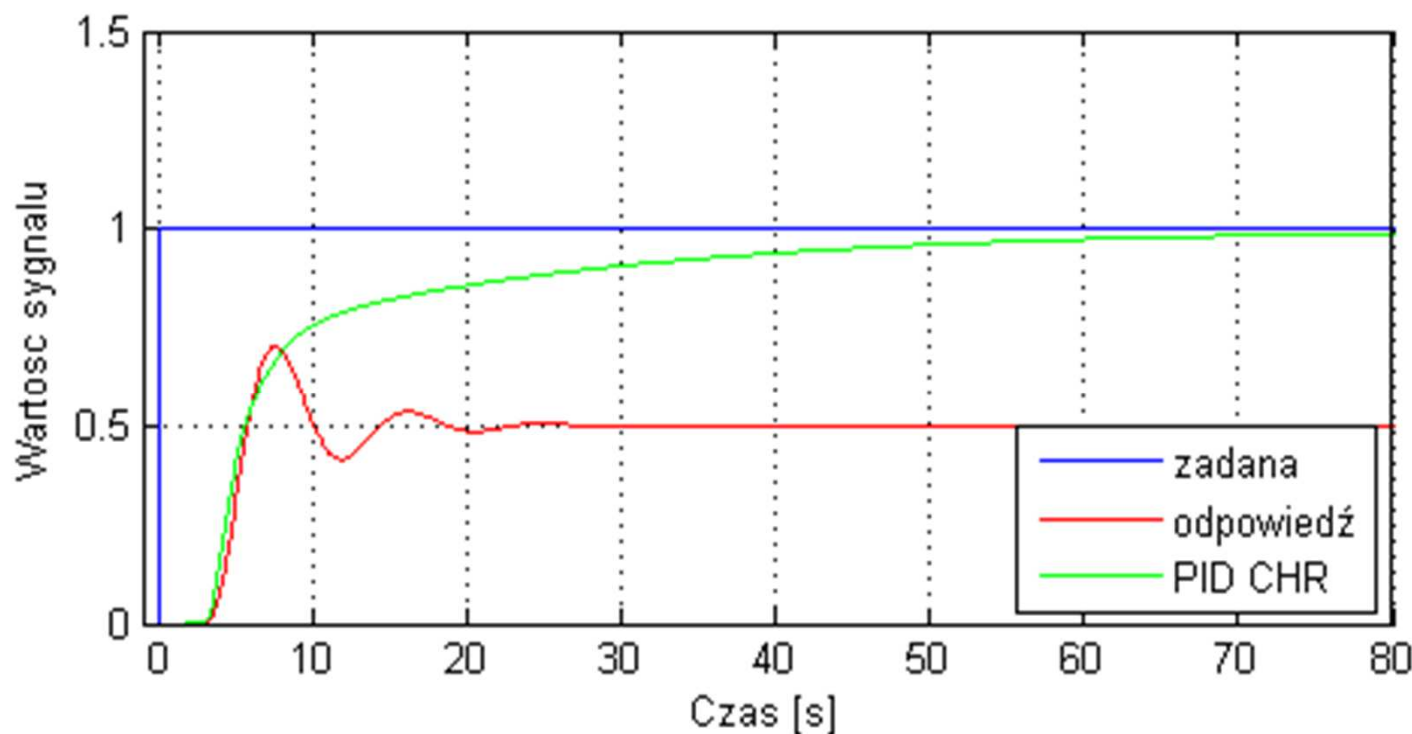
Rys. 8. Fragment programu odpowiedzialny za dobór nastaw regulatora PID

Symulacja w programie MATLAB



Rys. 9. Wyznaczanie stycznej w punkcie przecięcia odpowiedzi układu

Symulacja w programie MATLAB



Rys. 10. Porównanie sygnału wyjściowego regulowanego i bez regulacji

Opis stanowiska



Rys. 11. Sterownik PLC Sinamic S7-1200

Opis stanowiska



Rys. 12. Panel HMI Siemens Simatic HMI KTP600

Opis stanowiska



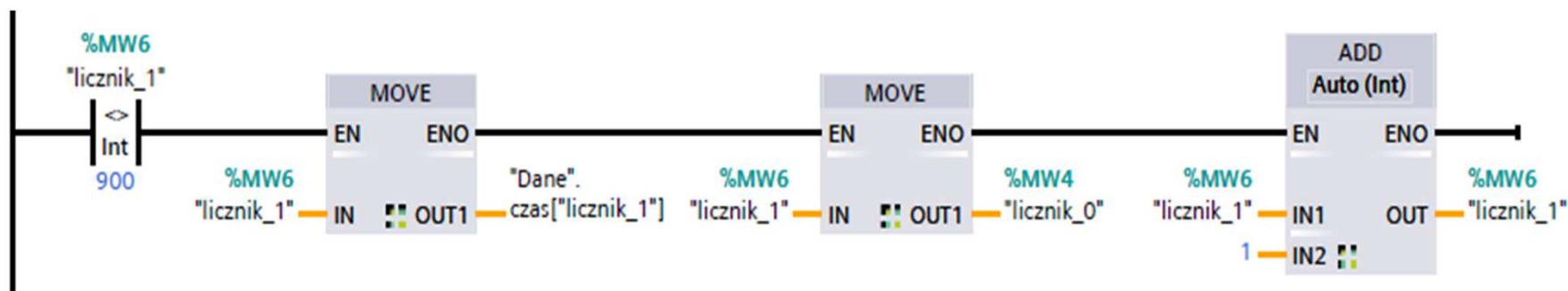
Rys. 13. Silnik prądu stałego i sterownik WOBIT SDC106

Opis stanowiska



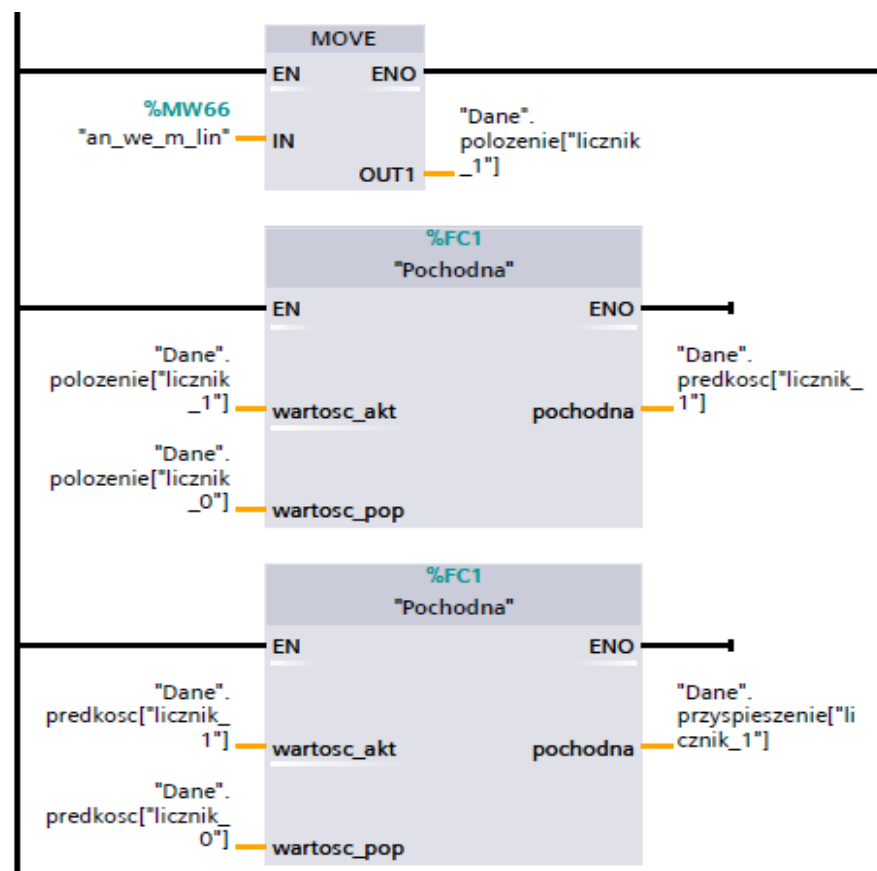
Rys. 14. Obiekt automatycznej regulacji

Działanie programu na PLC



Rys. 15. Przepisanie rejestru i inkrementacja licznika

Działanie programu na PLC



Rys. 16. Zapis aktualnych wartości położenia, prędkości i przyspieszenia

Działanie programu na PLC

```
//Obliczanie parametrow regulacji
```

```
#y1:= "Dane".predkosc["pkt_przeg"] * (0 - "pkt_przeg") + "Dane".polozenie["pkt_przeg"];  
#x1 := (0 - "Dane".polozenie["pkt_przeg"]) / "Dane".predkosc["pkt_przeg"] + "pkt_przeg";  
#x2 := ("zadana" - "Dane".polozenie["pkt_przeg"]) / "Dane".predkosc["pkt_przeg"] + "pkt_przeg";  
  
#punkt_przeg := ABS(#y1);  
#td := #x1;  
#to := #x2- #x1;
```

```
//Dobor nastaw regulatora PID
```

```
"Kp" := 0.6 /#punkt_przeg;           //wartość wzmacnienia  
"Ti" := 1 * #to;                     //stała całkowania  
"Td" := 0.5 * #td;                   //stała różniczkująca  
  
#Wzmocnienie := "Kp";  
#"Stała Ti" := "Ti";  
#"Stała Td" := "Td";
```

Rys. 17. Algorytm doboru nastaw regulatora PID

Dziękuję za uwagę