

Sterowanie obiektem cieplnym

Celem ćwiczenia jest realizacja i przetestowanie układu regulacji stałowartościowej z użyciem regulatora PI oraz obiektu cieplnego, który zidentyfikowano w poprzednim ćwiczeniu dla zadanego punktu pracy.

1. Dobór nastaw regulatora PI dla obiektu inercyjnego z opóźnieniem:

- a) Wybierz lepszą dopasowaną transmitancję obiektu cieplnego zidentyfikowanego podczas poprzedniego ćwiczenia (T_i , k_o , τ):

$$G(s) = \frac{k_o}{Ts+1} e^{-s\tau}$$

- b) Wybierz parametry $p\%$, k i b – kolumnę z tabeli poniżej:

$p\%$	0 ^{*)}	4,6	9,5	16,3	25,4	37,2
k	0,17	0,27	0,32	0,38	0,46	0,55
b	4,8	5,5	5,9	6,5	7,4	9,0

^{*)} przebiegi aperiodyczne krytyczne

- c) Za czas całkowania przyjmij stałą czasową:
 $T_i = T$

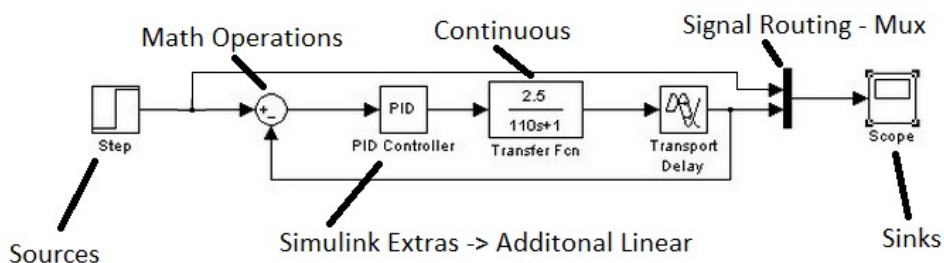
- d) Oblicz wzmacnienie regulatora PI na podstawie wybranych wcześniej parametrów:

$$k_p = 2k \frac{T_i}{k_o \tau}$$

- e) Oblicz projektowy czas regulacji:

$$t_r = b\tau$$

2. Przeprowadź w pakiecie Simulink symulację układu regulacji dla obliczonych nastaw regulatora PI oraz parametrów zidentyfikowanego obiektu:



Jeżeli układ działa zgodnie z założeniami projektowymi to przejdź do dalszej części ćwiczenia.

3. Wykonaj eksperyment sprawdzający jakość regulacji z użyciem regulatora PI z użyciem TwinCAT 3:

a) Zadeklaruj zmienne:

PROGRAM MAIN

VAR

```
w : LREAL := 40;
y : LREAL := 10;
u : LREAL;
fSetpointGen : LREAL;
fSetpointValue : LREAL := 50;
fManSyncValue : LREAL := 50;
bSync : BOOL;
bHold : BOOL;
stCTRL_PI_PARAMS : ST_CTRL_PI_PARAMS;
eErrorId : E_CTRL_ERRORCODES;
bError : BOOL;
bARWactive : BOOL;
bMode : BOOL := FALSE;
(* hardware *)
hwIn AT%I* : INT := 0;
hwInFix16 : T_FIX16;
hwOut AT%Q* : INT := 0;
hwOutwent AT%Q* : INT := 0;
hwWentylator : INT := 0;
hwOutFix16 : T_FIX16;
hwOutWentylatorFix16 : T_FIX16;
(* controller *)
fbCTRL_PI : FB_CTRL_PI;
bInit : BOOL := TRUE;
(* SP generator *)
stCTRL_GEN_PARAMS : ST_CTRL_SIGNAL_GENERATOR_PARAMS;
fbCTRL_GEN : FB_CTRL_SIGNAL_GENERATOR;
eSinState : E_CTRL_STATE;
eGenMode : E_CTRL_MODE;
eSinErrorId : E_CTRL_ERRORCODES;
bSinError : BOOL;
```

END_VAR

b) Napisz kod programu podstawiając obliczone parametry regulatora PI (podstaw za **stCTRL_PI_PARAMS.tTn** i **stCTRL_PI_PARAMS.fKp** obliczone nastawy):

IF bInit THEN

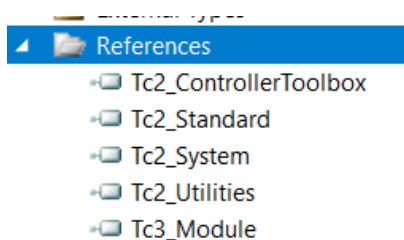
```
(* konfiguracja bloku regulatora FB_CTRL_PI *)
stCTRL_PI_PARAMS.tCtrlCycleTime := T#10MS;
stCTRL_PI_PARAMS.tTaskCycleTime := T#10MS;
stCTRL_PI_PARAMS.tTn := T#161.459S;(* TUTAJ WSTAWIC wyliczone Ti*)
stCTRL_PI_PARAMS.fKp := 2.6891;(* TUTAJ WSTAWIC wyliczone Kp *)
stCTRL_PI_PARAMS.fOutMaxLimit := 100; (* gorne ograniczenie wyjscia *)
stCTRL_PI_PARAMS.fOutMinLimit := 0;(* dolne ograniczenie wyjscia*)
(* konfiguracja generatora sygnalowego FB_CTRL_SIGNAL_GENERATOR *)
stCTRL_GEN_PARAMS.tCtrlCycleTime := T#10MS;
stCTRL_GEN_PARAMS.tTaskCycleTime := T#10MS;
stCTRL_GEN_PARAMS.eSignalType := eCTRL_SINUS; (* wybor rodzaju przebiegu,
wybrac odpowiednio: eCTRL_SINUS lub eCTRL_TRAINGLE *)
```

```

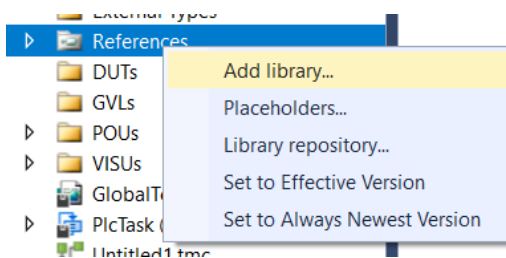
stCTRL_GEN_PARAMS.fAmplitude := 15; (* amplituda *)
stCTRL_GEN_PARAMS.fOffset := w; (* offset - skladowa stala *)
stCTRL_GEN_PARAMS.tCylceDuration := T#300S; (* okres przebiegu *)
stCTRL_GEN_PARAMS.tStart := T#0S;(* faza pocz1tkowa, odniesiona do okresu *)
(* flaga inicjujaca - zerowana, aby inicjalizacja wykonana zostala jednorazowo *)
bInit := FALSE;
END_IF
IF bMode THEN
    fSetpointValue := fSetpointGen;
    eGenMode := eCTRL_MODE_ACTIVE;
ELSE
    fSetpointValue := w;
    eGenMode := eCTRL_MODE_MANUAL;
END_IF
fbCTRL_GEN(
    fManValue := w,
    eMode := eGenMode,
    stParams := stCTRL_GEN_PARAMS,
    fOut => fSetpointGen,
    eState => eSinState,
    eErrorId => eSinErrorId,
    bError => bSinError
);
hwInFix16.value := hwIn;
hwInFix16.n := 0;
y := 100.0 / 16#7FFF * FIX16_TO_LREAL(hwInFix16);
(* call controller *)
fbCTRL_PI(
    fSetpointValue := fSetpointValue,
    fActualValue := y,
    fManSyncValue := fManSyncValue,
    bSync := bSync,
    eMode := eCTRL_MODE_ACTIVE,
    bHold := bHold,
    stParams := stCTRL_PI_PARAMS,
    fOut => u,
    bARWactive => bARWactive,
    eErrorId => eErrorId,
    bError => bError
);
hwOutFix16 := LREAL_TO_FIX16(u * 16#7FFF / 100.0, 0);
hwOutWentylatorFix16 := LREAL_TO_FIX16(hwWentylator * 16#7FFF / 100.0,0);
hwOutwent := hwOutWentylatorFix16.value;
hwOut := hwOutFix16.value;

```

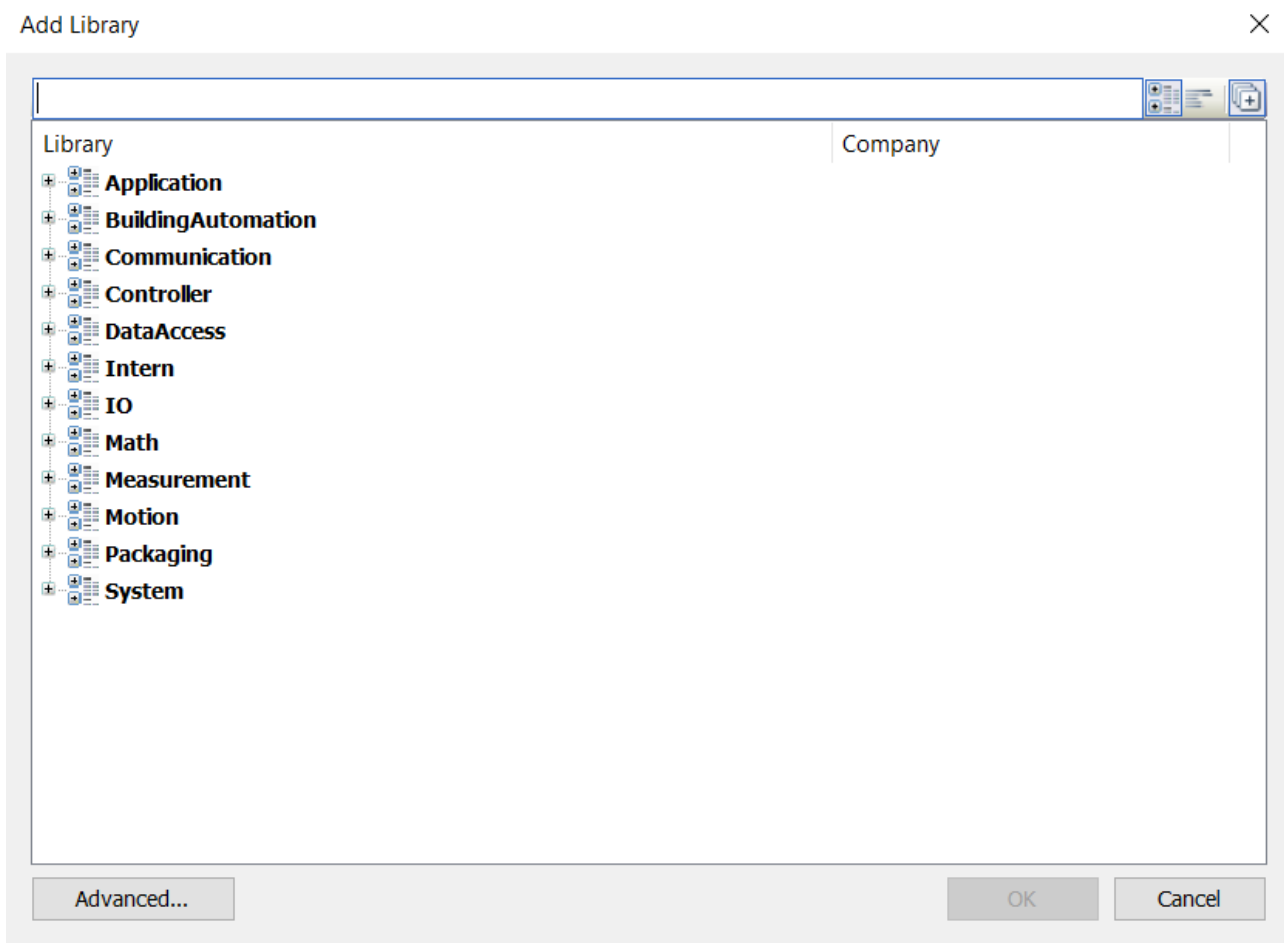
c) Dodaj następujące biblioteki do **References** (niektóre mogą być dodane domyślnie):



W tym celu kliknij prawym przyciskiem na **References** i wybierz **Add library**:



Następnie wyszukaj i dodaj odpowiednie biblioteki poprzez okno:



- d) Powiąż zmienną **temperatura** z odpowiednim wejściem sterownika z modułu modułu **EL3054** (wybierz to samo wejście co na poprzednim laboratorium).
- e) Powiąż zmienną **grzalka** z 2 wyjściem modułu **EL4028**.
- f) Uruchom program i ustaw zmienną **w** na punkt pracy w jakim był identyfikowany obiekt cieplny na poprzednich zajęciach np. dla przebiegu identyfikacyjnego z poprzednich zajęć:



zmienną w należałoby ustawić na 44.5.

- g) Wyświetl przebiegi zmiennych u , w i y w **TwinCAT Measurement Project** (analogicznie do poprzednich zajęć) i po ustabilizowaniu temperatury (y) wykonaj dwa skoki zmiennej w (wartości zadanej) o 5 i 10°C (od punktu pracy dla identyfikacji), np.:



- h) Porównaj wartości projektowe przeregulowania i czasu regulacji 2% z rzeczywistymi obliczonymi na podstawie eksperymentu.

4. W sprawozdaniu zamieść:

- obliczenia nastaw regulatora PI,
- odpowiedź skokową symulacji z simulinka,
- screeny 2 odpowiedzi skokowych rzeczywistego obiektu cieplnego,
- obliczenia przeregulowania i czasu regulacji 2% dla każdego skoku najlepiej

- z powiększonymi odpowiedziami skokowymi, tak żeby było wiadomo na podstawie jakich punktów przeprowadzono obliczenia,
- e) porównanie w tabelce otrzymanych wyników z wartościami projektowymi,
 - f) wnioski jeśli posiadasz.