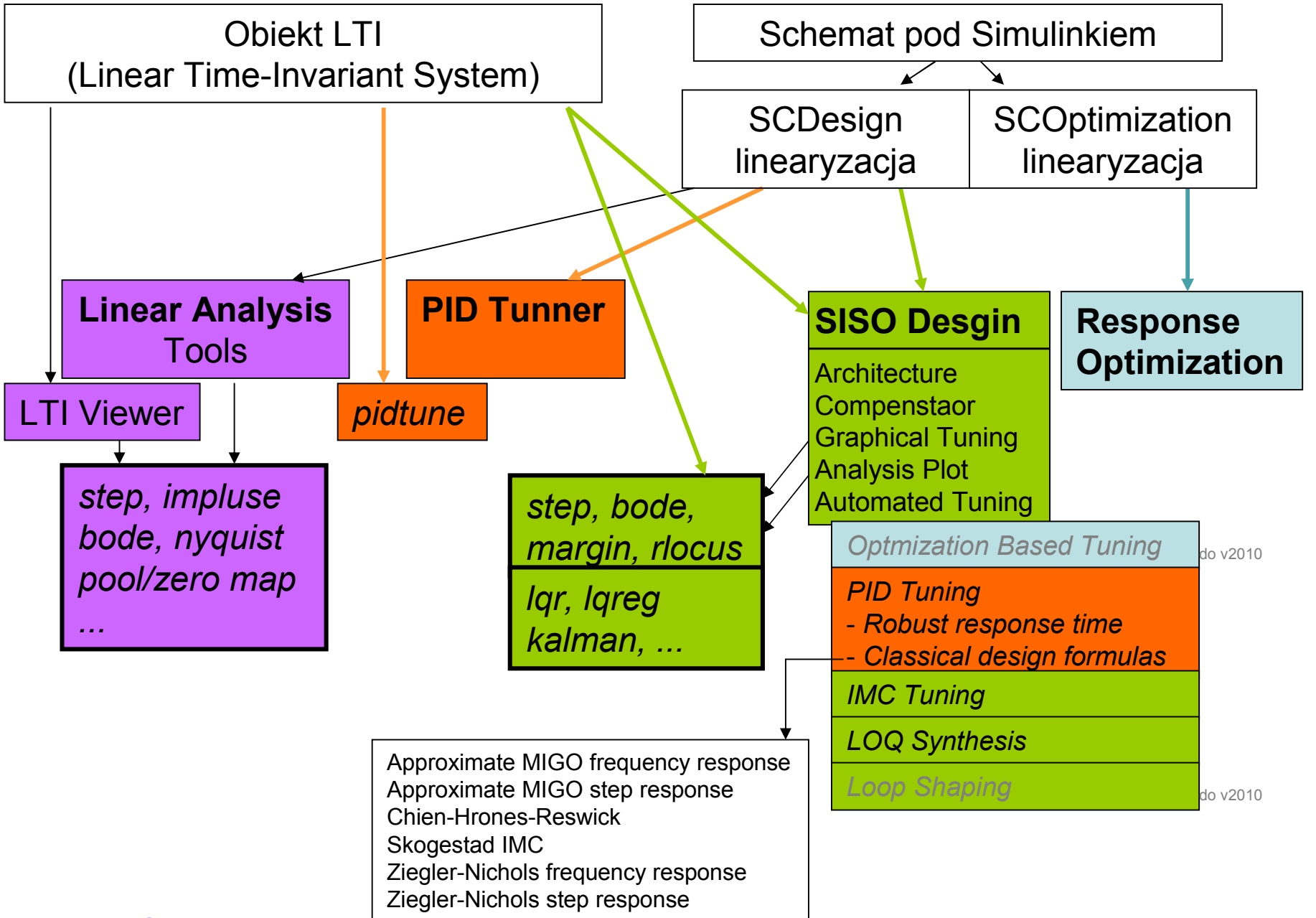
Narzędzia wspomagające projektowanie UR – SISO Design



Funkcje dla modeli LTI (wywoływane z linii komend) – Matlab+Control:

- ltiview(obiekt LTI) – okno pt.: „LTI Viewer”
- pidtune(obiekt LTI, typ_regulatora) – nastawy na ekranie
- sisotool(obiekt LTI) – okno pt.: „Control and Estimation Tools Manager”, (oraz „SISO Design”, „LTI View”)

Interaktywne interfejsy użytkownika wspomagające projektowanie - Matlab

Linear Analysis Tools

→ LA
→ SCD/2.2

- okno pt. „Linear Analysis Tools”
- moduły: Matlab + Simulink + Simulink Control Design + Control
- wywołanie:

-schemat → Tools → Control Design → Linear Analysis

PID Tuner

→ PID_Tuner

- okno pt. „PID Tuner”
- moduły: Matlab + Simulink + Simulink Control Design + Control
- wywołanie:

• blok PID (Simulink/Continues) → okno parametrów bloku → PID Tuner

Compensator Design (SISO Design)

→ SCD/2.1

- okno pt. „Control and Estimation Tools Manager” (oraz „SISO Design”, „LTI View”)
- moduły: Matlab + Simulink + Simulink Control Design + Control
- wywołanie:

- schemat → Tools → Control Design → Compensator Design → okno Control and Estimation Manager → Tune Block

Response Optimization

→ SDO
→ SDO
→ SCD/1

- okno pt. „Response Optimization”
- moduły: Matlab + Simulink + Simulink Optimization Design + Optimization
- wywołanie:

- schemat → Tools → Response Optimization
- blok Check ... (Simulink Optimization Design) → okno parametrów bloku → Response Optimization
Check Against Reference, Check Custom Bounds, **Check Step Response Characterisitics**
- blok Check ... (Simulink Control Design) → okno parametrów bloku → Response Optimization
Check Bode Characteristics, Check Gain and Phase Margins, **Check Linear Step Response Characterisitics**
Check Nichols Characteristics, Check Pole-Zero Characteristics, Check Singular Value Characteristics

Narzędzia wspomagające projektowanie - Matlab

Matlab Product Family:

Matlab	**9 540**	
Control System Toolbox	**4 500**	**14 040**
System Identification Toolbox	4 500	
Robust Control	8 460	
Model Predictive Control	8 460	**23 940**
Optimization Toolbox	4 500	
Neural Network Toolbox	4 500	
Fuzzy Logic Toolbox	4 500	

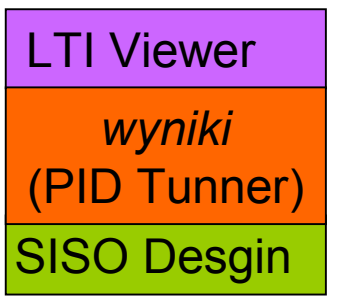
Simulink Product Family:

Simulink	14 400	
Simulink Control Design (wymaga Control ST)	4 500	**32 940**
Simulink Design Optimization (wymaga Optimization T)	4 500	**18 540**

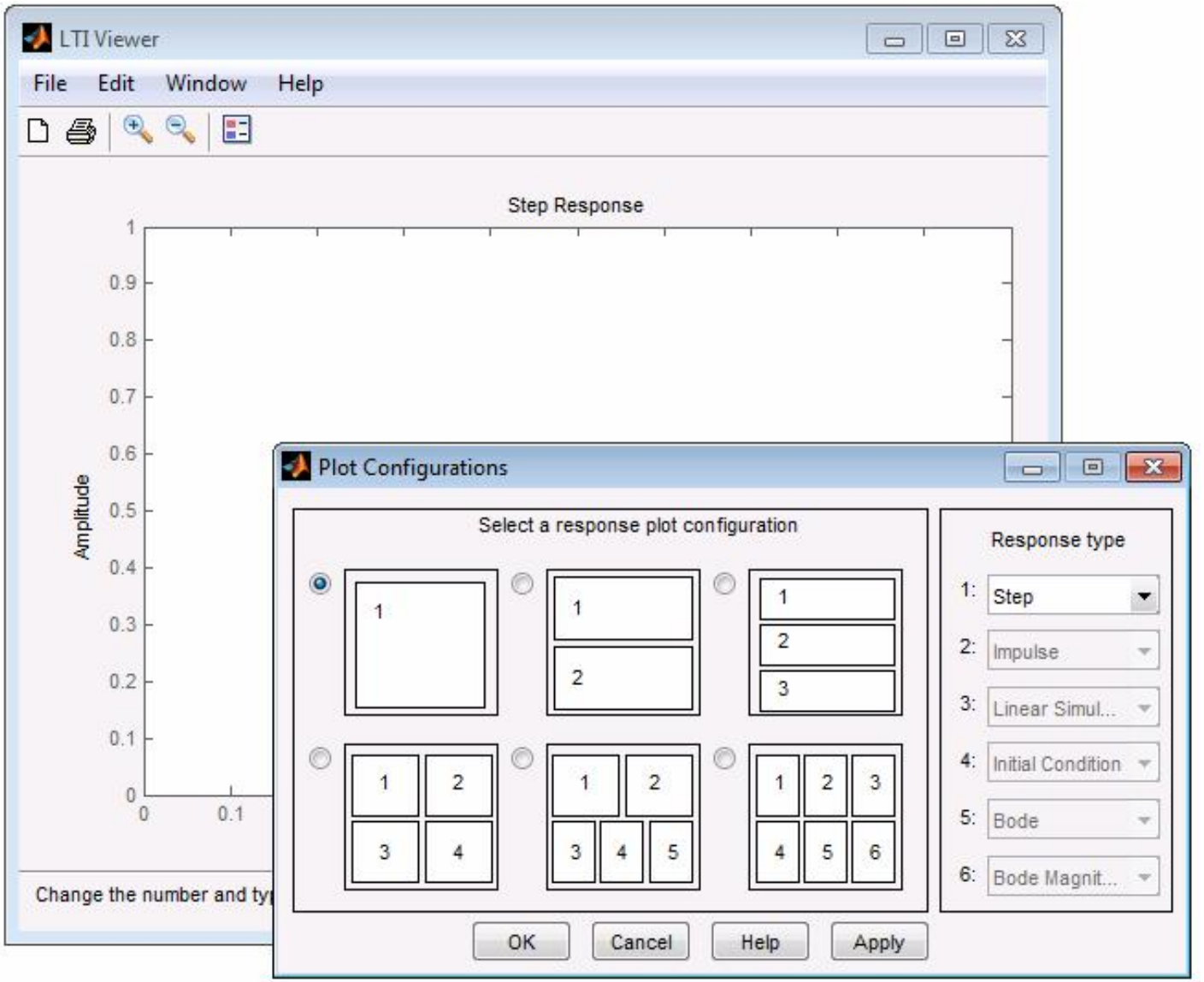
Lic. komercyjna, 1 stanowisko, netto (2011r.)

Matlab + Control

- modele (transfer function, state-space, pole-zero-gain, frequency-response)
- konwersje
- połączenia (series, parallel, feedback, …)
- funkcje (step response, Bode, Nyquist, ..)
- metody projektowania (Root locus, Bode diagram, LQR, LQG, …)
- narzędzia interaktywne
    - ltiview
    - pidtool
    - sisotool, sisoinit



help control



Wywołanie z linii komend: ltiview(obiekt LTI)
Analiza obiektów LTI (linowych, stacjonarnych)

regul = pidtune(obiekt, typ_reg)

[regul info] = pidtune(obiekt, typ_reg)

obiekt= model tf, ss, zpk, np.: obiekt=tf(1, [1 1])

typ_reg = ('p', 'i', pi', 'pd', 'pdf', 'pid', 'pidf')

regul= Kp + Ki * 1/s

with Kp=0.473, Ki=1.6

Continues-time PI controller in parrallel form

info= Stable: 1

CrossoverFrequency: 1.1237

PhaseMargin: 60.0000

[regul info] = pidtune(obiekt, typ_reg, opcje)

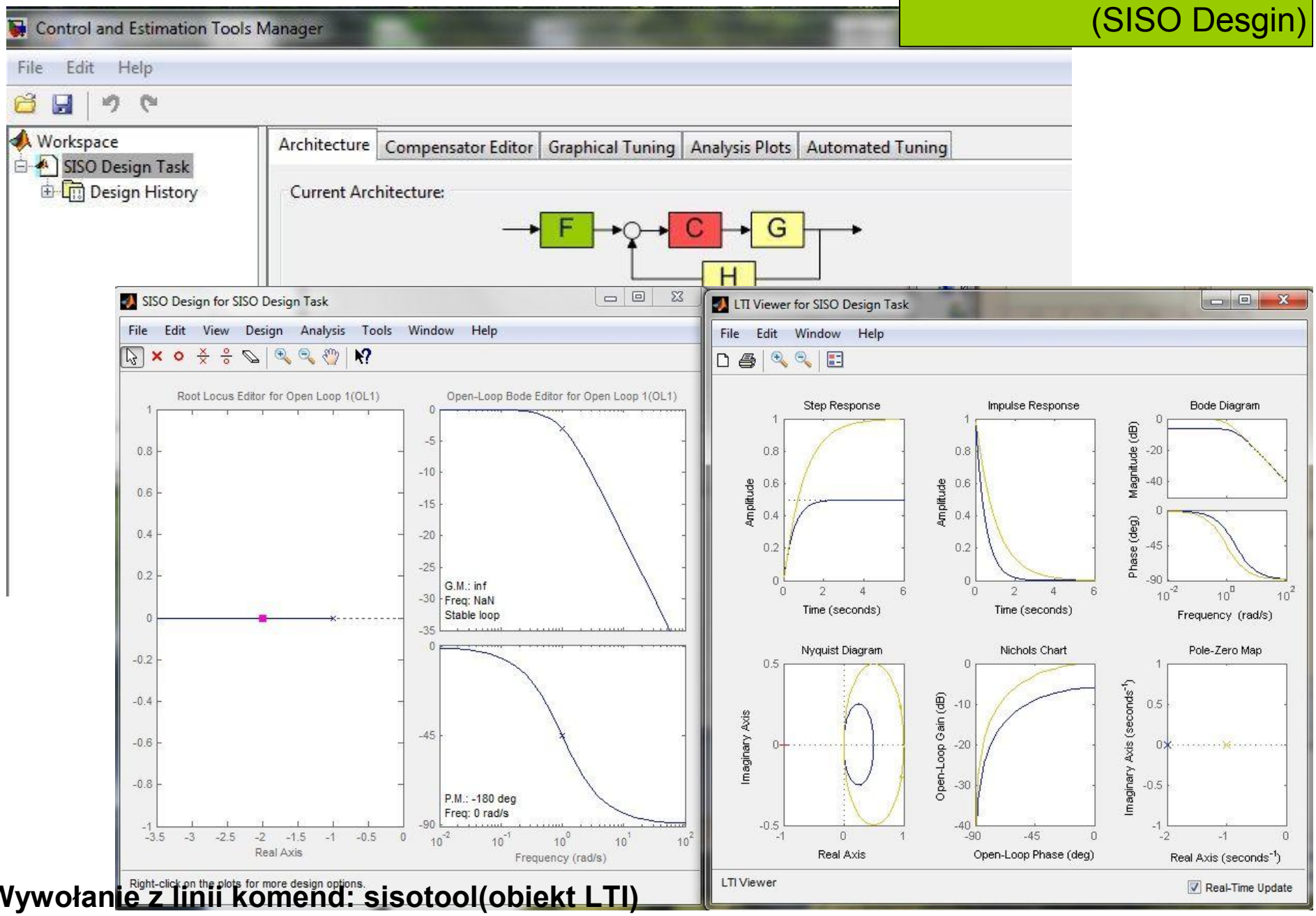
opcje = pidtuneOptions('CrossoverFrequency',1.2,'PhaseMargin',45);

[C info] = pidtune(obiekt,typ_reg, opcje)

Wywołanie z linii komend: pidtune(obiekt LTI, typ_reg [,opcje])

Matlab + Control

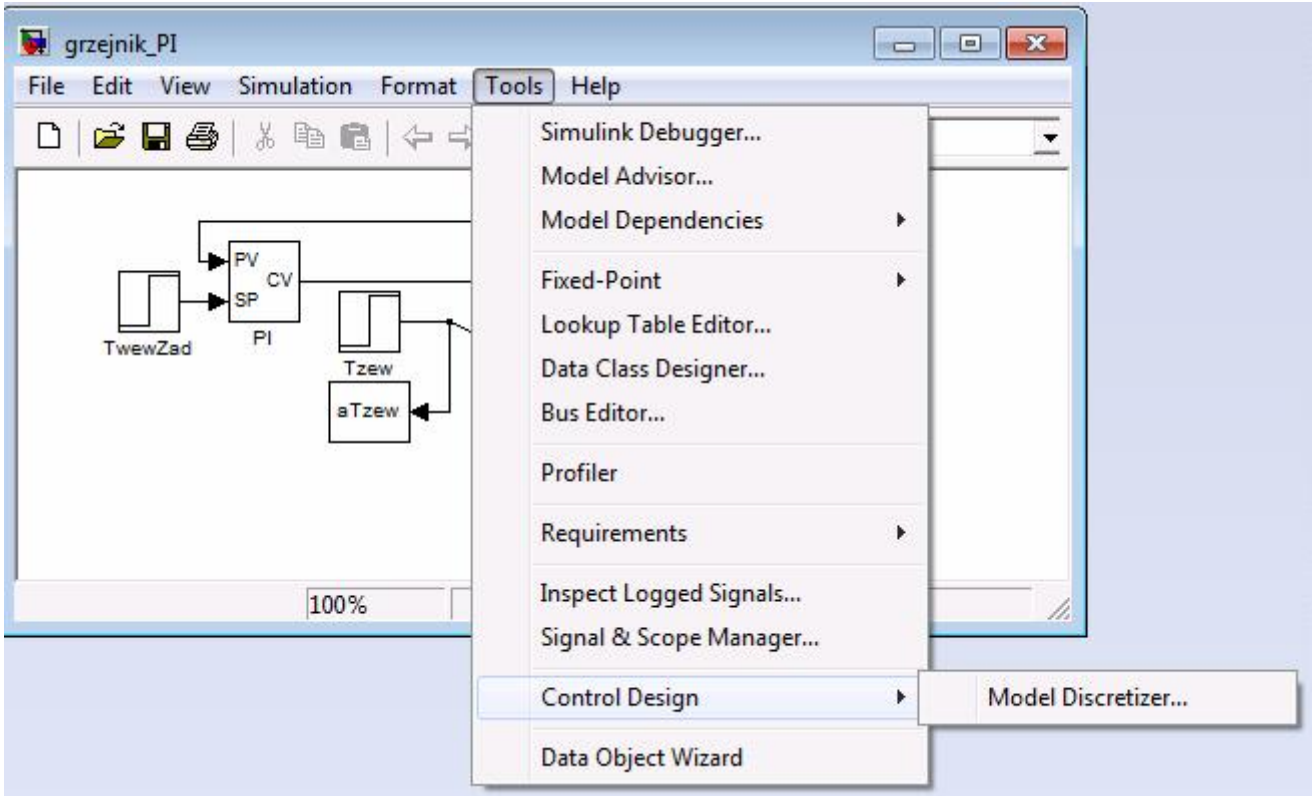
Control and Estimation TM (SISO Desgin)



Wywołanie z linii komend: sisotool(obiekt LTI)

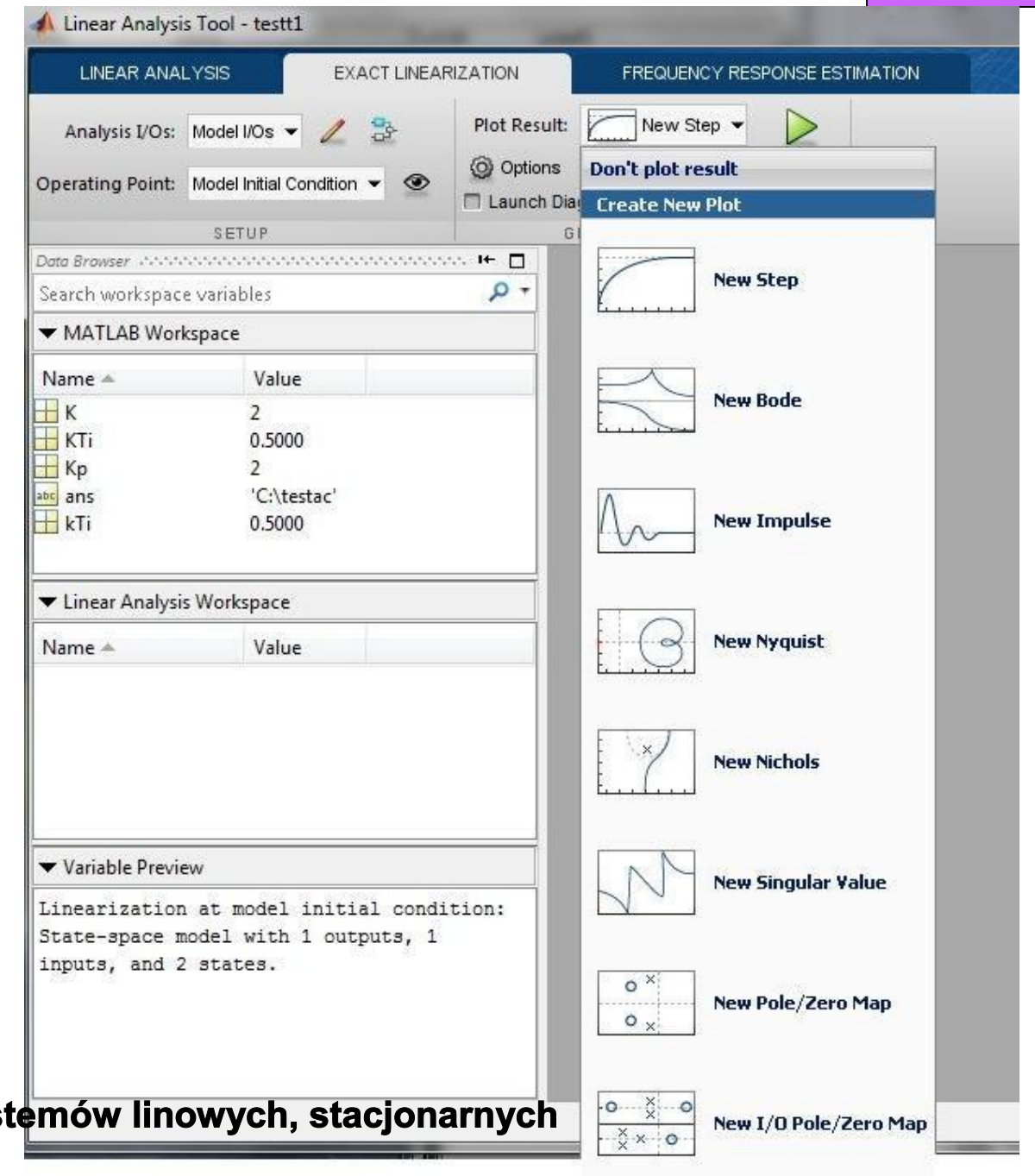
Projektowanie wybranego układu regulacji dla obiektu LTI (układ SISO)

Matlab + Simulnik



Matlab + Simulnik + Simulink Control Design + Control

Linear Analysis

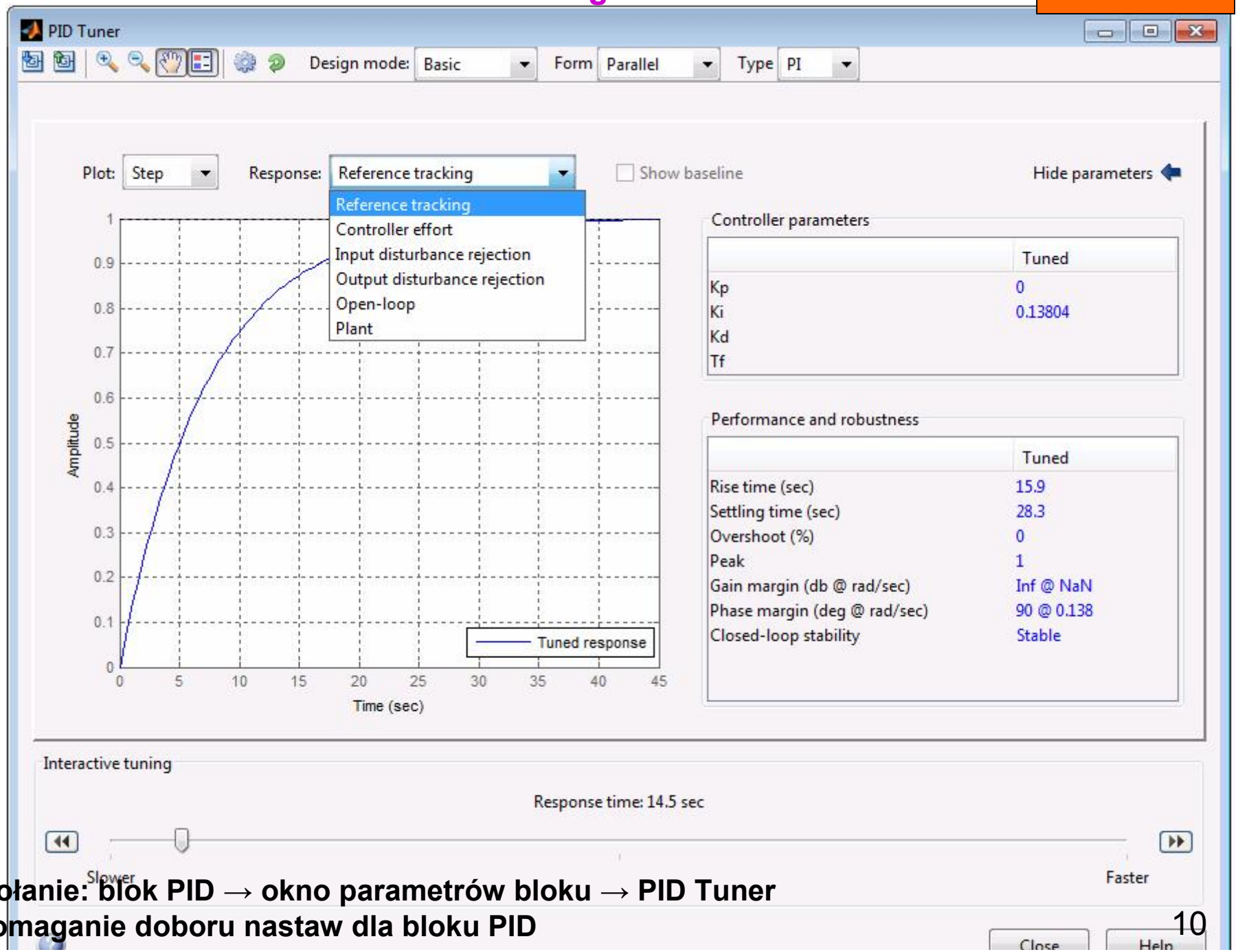


Wywołanie: schemat
→ Tools
→ Control Design
→ Linear Analysis

Linearyzacja i analiza systemów linowych, stacjonarnych

Matlab + Simulink + Simulink Control Design + Control

PID Tunner

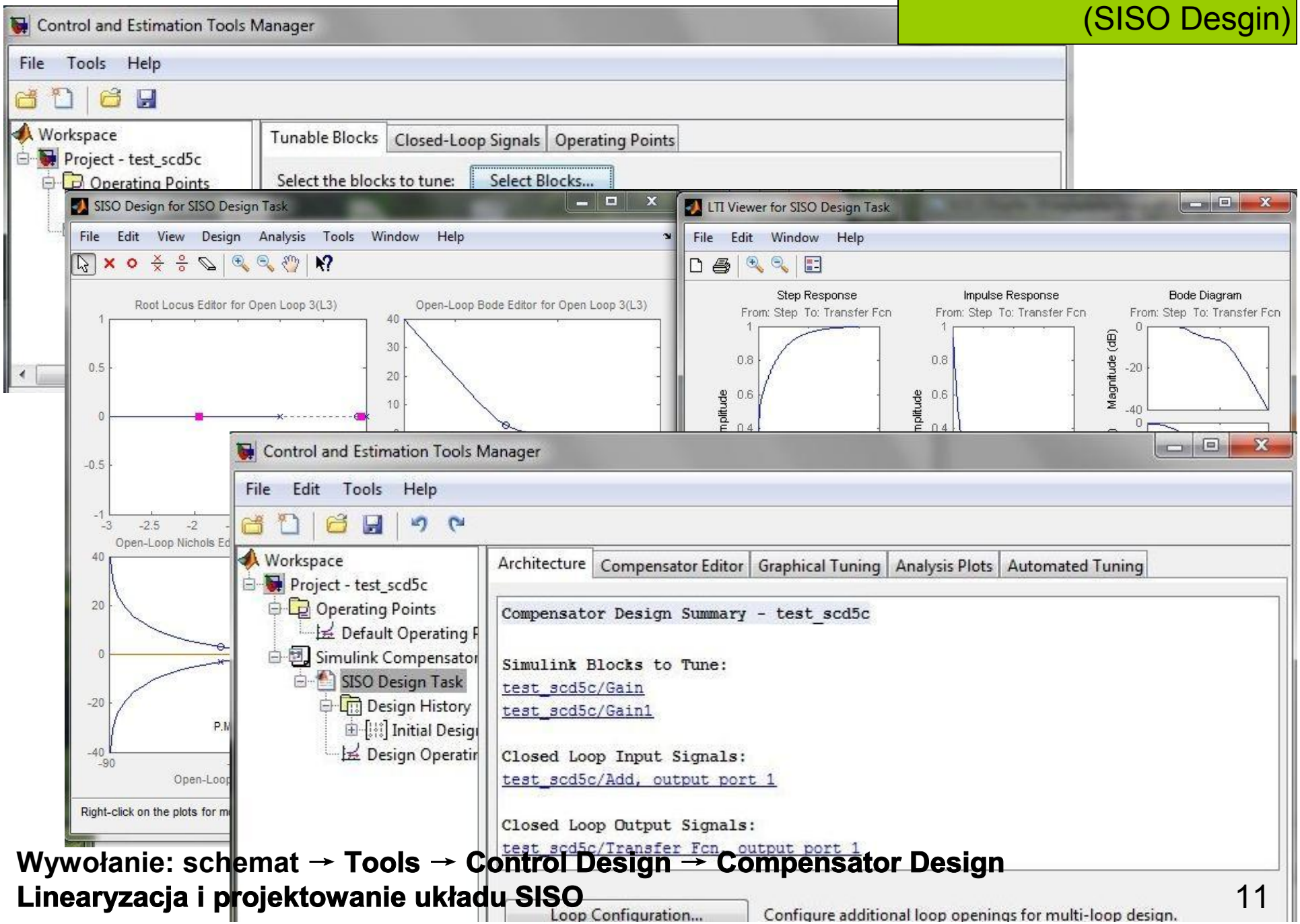


Wywołanie: blok PID → okno parametrów bloku → PID Tuner

Wspomaganie doboru nastaw dla bloku PID

Matlab + Simulnik + Simulink Control Design + Control

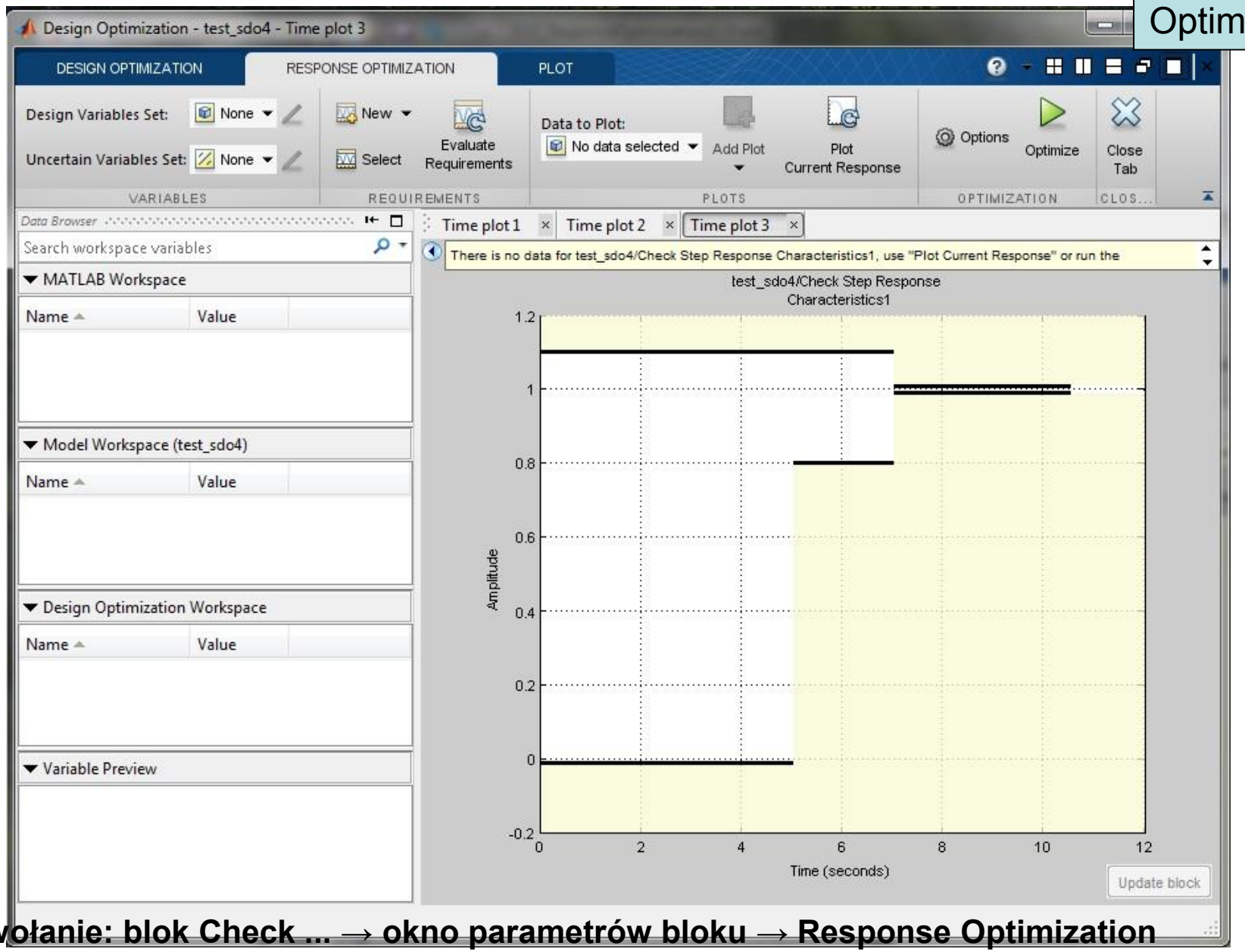
Control and Estimation TM (SISO Desgin)



Wywołanie: schemat → Tools → Control Design → Compensator Design

Matlab + Simulink + Simulink Optimization Design + Optimization

Response Optimization

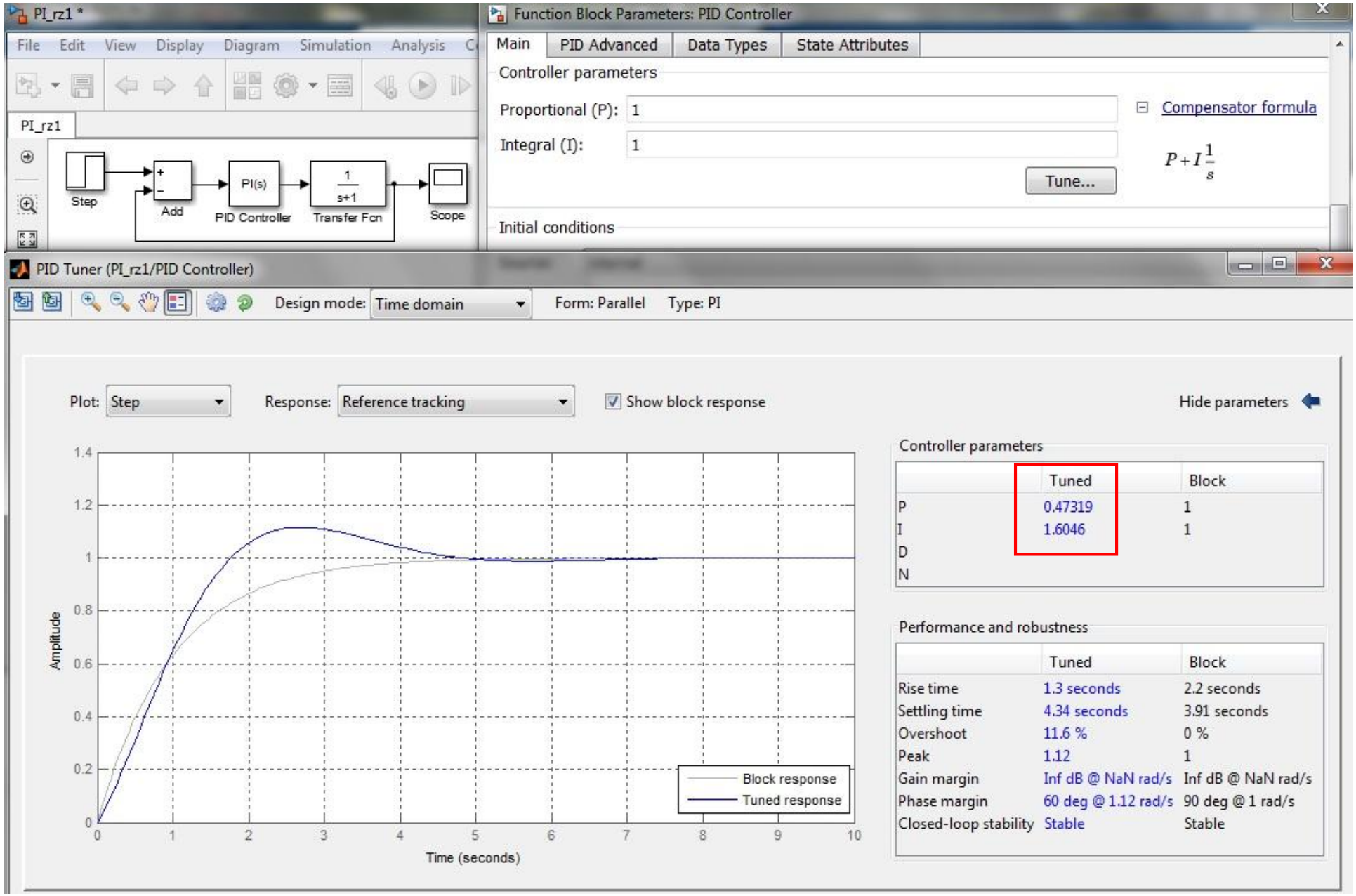


Wywołanie: blok Check ... → okno parametrów bloku → Response Optimization

np. Check [Linear] Step Response Characterisitics

Przykład 1: obiekt = 1/(s+1), regulator = Kp + Ki/s

blok PID + Tune

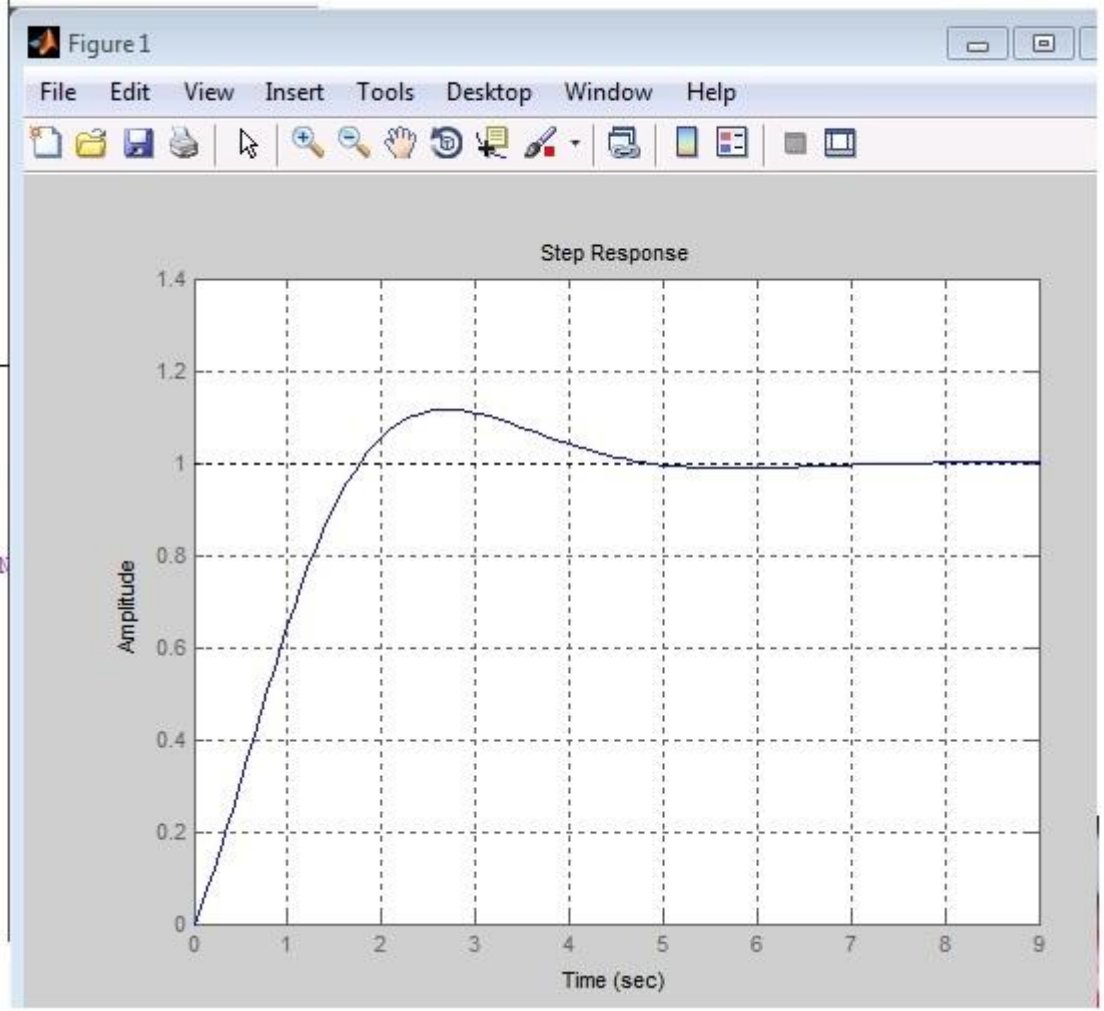


Przykład 1: obiekt = 1/(s+1), regulator = Kp + Ki/s

funkcja pidtune

```
close all, clear all
s=tf('s');
K1=1; T1=1;
G1=K1/(T1*s+1);

[C,info]=pidtune(G1,'pi')
R=C;                                  %1.sposob
%Kp=C.Kp; Ki=C.Ki; R=Kp+Ki/s;         %2.sposob
Gz1=feedback(G1*R,1);
step(Gz1);
stepinfo(Gz1), %allmargin(G1*R) i info
```

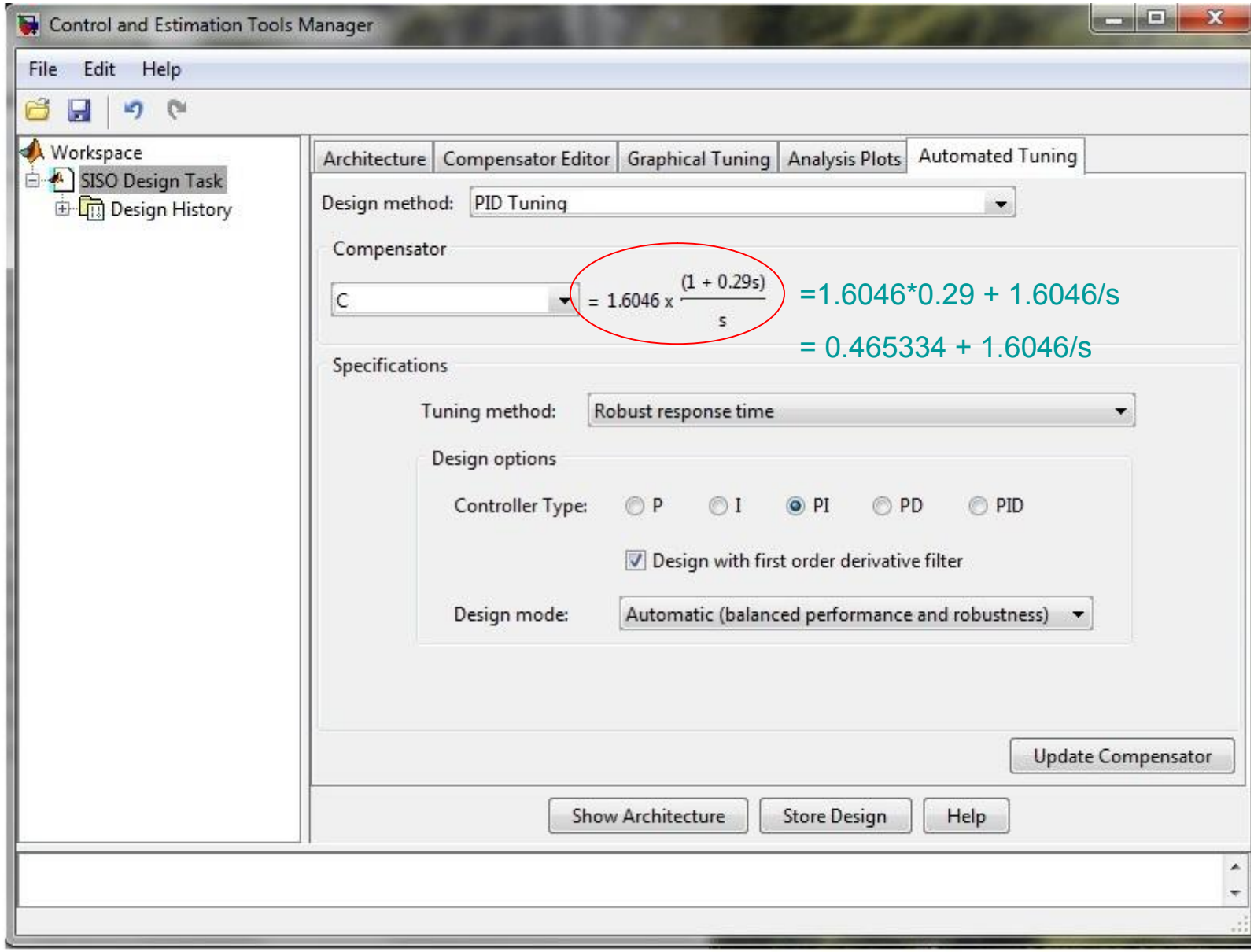
```
Continuous-time PI controller in parallel form:
                 1
Kp + Ki * ---                                   i;
                 s
with Kp = 0.47319, Ki = 1.6046
info =                                          (ZN
                 Stable: 1
    CrossoverFrequency: 1.1237
            PhaseMargin: 60.0000
ans =
        RiseTime: 1.3003
    SettlingTime: 4.3377
     SettlingMin: 0.9053
     SettlingMax: 1.1158
       Overshoot: 11.5817
      Undershoot: 0
            Peak: 1.1158
        PeakTime: 2.7063
```



Przykład 1: obiekt = 1/(s+1), regulator = Kp + Ki/s

sisotool

```
s=tf('s');
G=1/(1*s+1);

Kp=1; Ki=1;
R=Kp+Ki/s;

%Gz=feedback(G*R,1);
sisotool(G,R)
```



Przykład 2: obiekt = 1/(10s+1)*exp(-s), regulator = Kp + Ki/s

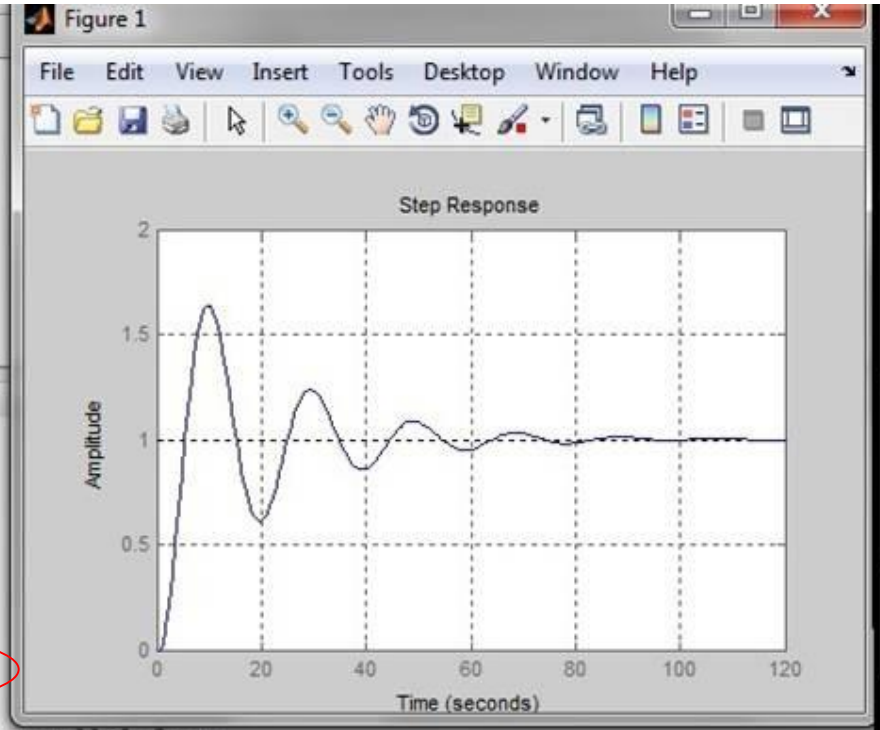
funkcja pidtune

PI_rz10to_txt.m

```
s=tf('s');
G=1/(10*s+1)*exp(-s);
Kp=1; Ki=1;
R=Kp+Ki/s;
[C info]= pidtune(G,'pi')
R=C;
Gz=feedback(G*R,1);
step(Gz)
stepinfo(Gz)
```

Command Window

```
C =

             1
  Kp + Ki * ---
             s

  with Kp = 0.552, Ki = 0.141

Continuous-time PI controller in parallel form.

info =
                 Stable: 1
    CrossoverFrequency: 0.1049
            PhaseMargin: 60.0000
ans =
        RiseTime: 3.2878
    SettlingTime: 71.4931
     SettlingMin: 0.6121
     SettlingMax: 1.6382
       Overshoot: 63.8171
      Undershoot: 0
            Peak: 1.6382
        PeakTime: 9.5895
>>
```

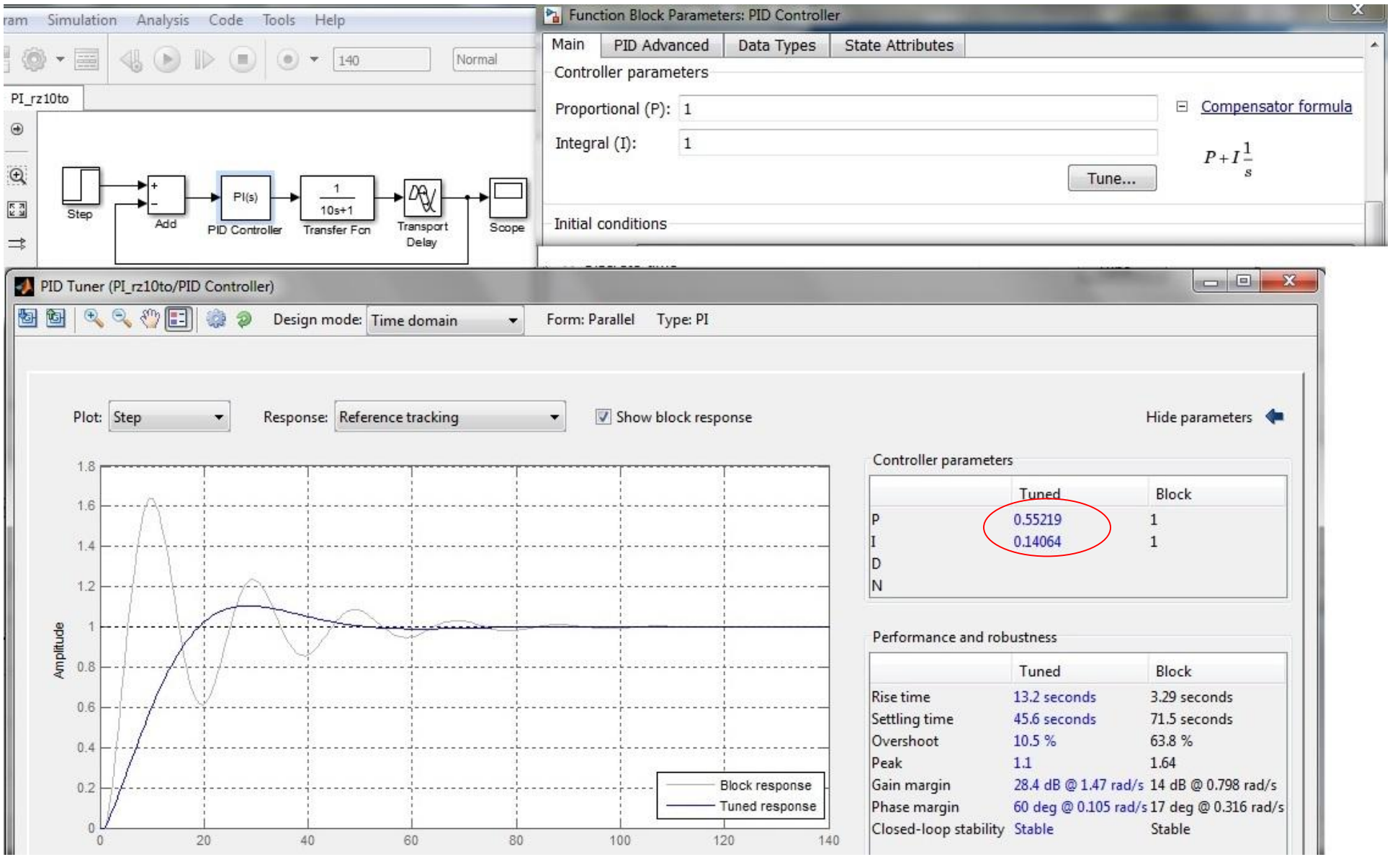



```
s=tf('s');
G=1/(10*s+1)*exp(-s);
Gp=1/(10*s+1)*pade(exp(-s),1);
Kp=1; Ki=1;
R=Kp+Ki/s;

%Matlab > R2010b
[C info]= pidtune(G,'pi') R=C;
Gz=feedback(G*R,1);
step(Gz)
stepinfo(Gz)

%zawsze
[C info]= pidtune(Gp,'pi')  R=C;
Gz=feedback(Gp*R,1);
step(Gz)
stepinfo(Gz)
```

```
             1
  Kp + Ki * ---
             s

  with Kp = 0.61518, Ki = 0.15101

info =

                 Stable: 1
    CrossoverFrequency: 0.1110
            PhaseMargin: 60.0000
ans =

        RiseTime: 3.3021
    SettlingTime: 71.3813
     SettlingMin: 0.6148
     SettlingMax: 1.6368
       Overshoot: 63.6762
      Undershoot: 1.8514
            Peak: 1.6368
        PeakTime: 9.7800
```

Przykład 2: obiekt = 1/(10s+1)*exp(-s), regulator = Kp + Ki/s

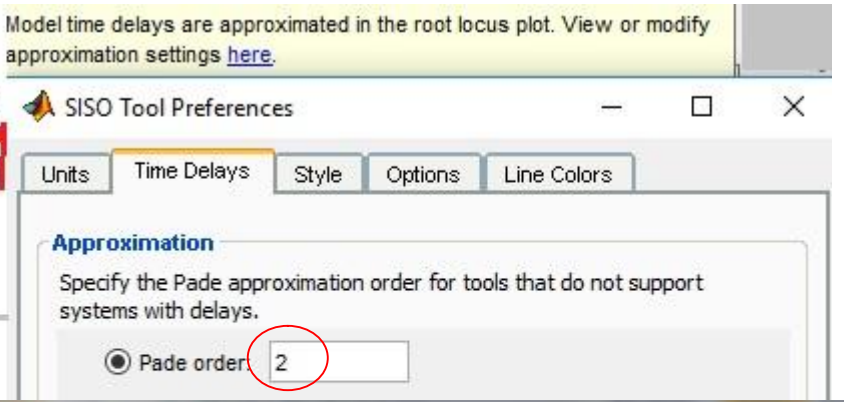
blok PID + Tune

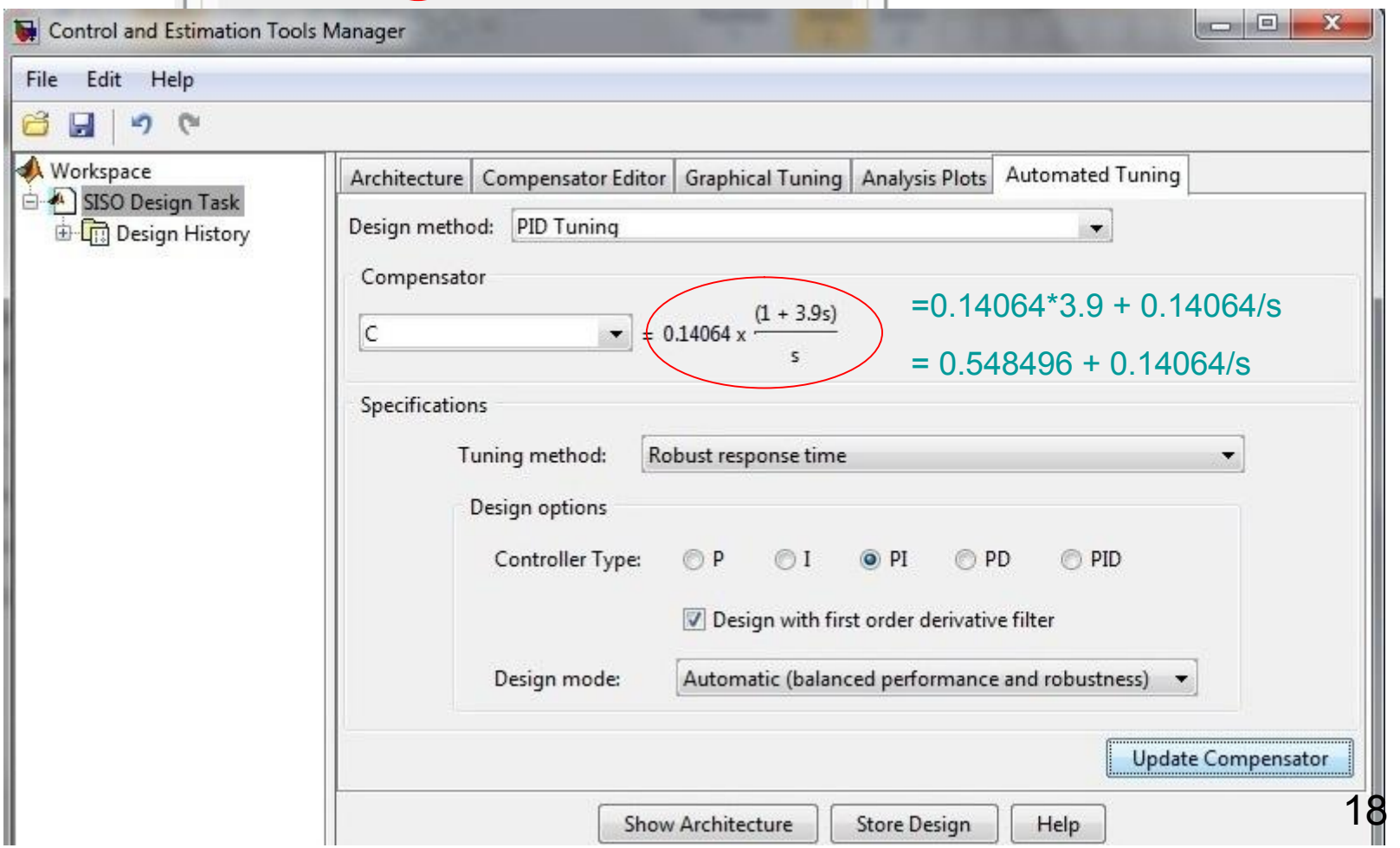


Przykład 2: obiekt = 1/(10s+1)*exp(-s), regulator = Kp + Ki/s

sisotool

```
s=tf('s');
G=1/(10*s+1)*exp(-s);
Gp=1/(10*s+1)*pade(exp(-s),1);
Kp=1; Ki=1;
R=Kp+Ki/s;

%aproksymacja automatyczna
sisotool(G,R)

%aproksymacja ręczna
sisotool(Gp,R)
```





=0.14064*3.9 + 0.14064/s

= 0.548496 + 0.14064/s