Control an IT1-Process with a P-Controller (numerically):

```
by L.Billmann 06/2011
File: IT1withP_num.m
Doku: IT1withP_num.pdf
Task: To check the controlability of a given IT1-process with a simple
      P-controller and determine the expected time rsponses.
ext. Files:   -
M-Block:      -
LBtools/math: -
```

```
clc, clear all, close all
s = polynom(0,'s');
```

Given process:

```
Ti = 10; fprintf('    integration time Ti = %g s\n',Ti)
T1 = 5;  fprintf('    time constant    T1 = %g s\n',T1)
Fs = lintf('c',1/(Ti*s*(1+T1*s)))
```

```
    integration time Ti = 10 s
    time constant    T1 = 5 s
Fs =
    '       1       '
    '---------------'
    ' 10*s + 50*s^2 '
```

Check stability by pole-placement (manual or symbolic):

```
syms Kr p
```

- Determine the reference transfer function Fw

```
Fr = Kr;
fprintf('    with   Fr(p) = %s\n',char(Fr))
Fs = 1/(Ti*p*(1+T1*p));
fprintf('    and    Fs(p) = %s\n',char(Fs))
disp(   '    we get Fw(p) =')
Fw = Fr*Fs/(1 + Fr*Fs); Fw=simplify(Fw);pretty(Fw)
```

```
    with   Fr(p) = Kr
    and    Fs(p) = 1/10/p/(1+5*p)
    we get Fw(p) =

                                  Kr
                           -----------------
                                     2
                           10 p + 50 p  + Kr
```

- Look for stable pole placement

```
[num,den] = numden(Fw); poles = solve(den,p);
fprintf('    from the characteristic polynomial Q(p) = %s\n',...
         [char(den) '= 0'])
disp('    we get p1 ='), pretty(poles(1))
disp('    and    p2 ='), pretty(poles(2))

disp(' ')
Krsemi = solve(poles(1),Kr);
fprintf('    For the stability limit case (poles=0), we get Kr = %s\n',...
        char(Krsemi))
disp('    So we can state out the stability condition:')
fprintf('\n    >>> stable, if Kr > %s\t\t\t\t(1)\n',char(Krsemi))
```

```
    from the characteristic polynomial Q(p) = 10*p+50*p^2+Kr= 0
    we get p1 =

                                                1/2
                      - 1/10 + 1/10 (1 - 2 Kr)
    and    p2 =

                                                1/2
                      - 1/10 - 1/10 (1 - 2 Kr)

    For the stability limit case (poles=0), we get Kr = 0
    So we can state out the stability condition:

    >>> stable, if Kr > 0                               (1)
```

- Check for aperiodic/periodic poles

```
disp('    Behavior becomes periodic if a pole becomes complex,')
disp('    which happens if radicant becomes negative.')
disp('    Looking at the so called aperiodic limit case,')
disp('    where we change from periodic to aperiodic while')
disp('    the radicant Ti^2-4*Ti*T1*Kr becomes zero.')
disp('    With this we get:')
apg=solve(Ti^2-4*Ti*T1*Kr,Kr);
fprintf('\n    >>> aperiodic if Kr < %s\t\t\t(2)\n',char(apg))
```

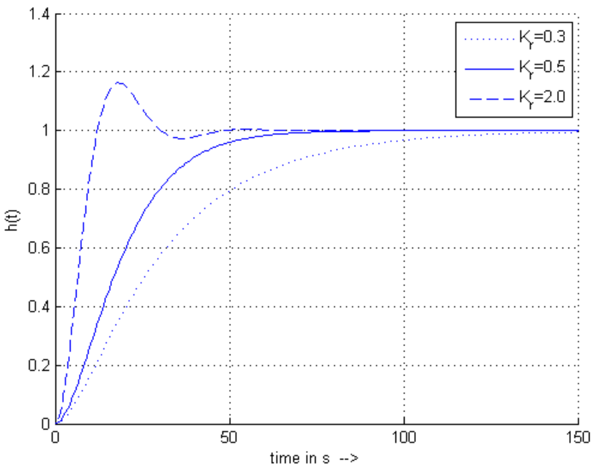
```
    Behavior becomes periodic if a pole becomes complex,
    which happens if radicant becomes negative.
    Looking at the so called aperiodic limit case,
    where we change from periodic to aperiodic while
    the radicant Ti^2-4*Ti*T1*Kr becomes zero.
    With this we get:

    >>> aperiodic if Kr < 1/2                   (2)
```

```
For further investigations we assume 3 different controller settings:
Kr = 0.3    aperiodic
Kr = 0.5    aperiodic limit case
Kr = 2.0    periodic
```

Determine the expected reference reaction:

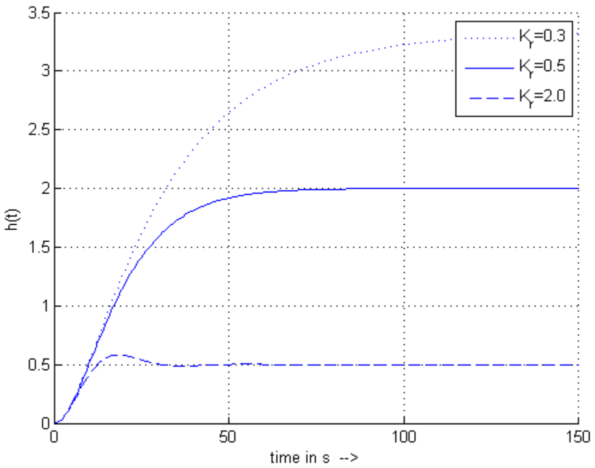
```
Fs = lintf('c',1/(Ti*s*(1+T1*s)));
figure(1), set(gcf,'color','white'), hold on
for Kr = [0.3 0.5 2]
    Fw = Kr*Fs/(1+Kr*Fs); pstep(Fw,150)
end
d=get(gca,'children');      set(d(1),'LineStyle','--')
set(d(2),'LineStyle','-'), set(d(3),'LineStyle',':')
legend('K_r=0.3','K_r=0.5','K_r=2.0')
```




```
- No permanent control deviation expected for setpoint changes
- Higher values of Kr speed up the reaction but increase
  oscillations
```

Determine the expected input disturbance reaction:

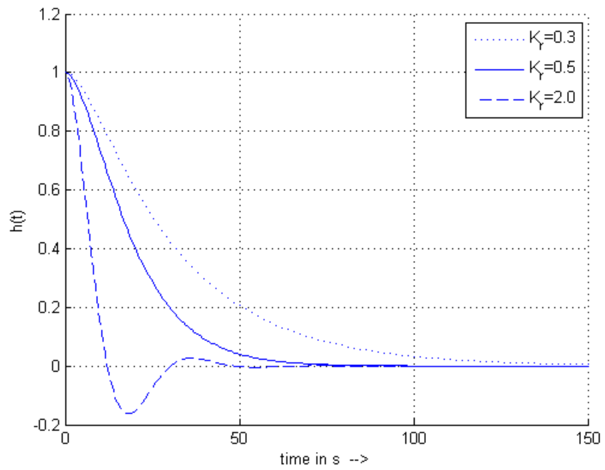
```
figure(2), set(gcf,'color','white'), hold on
for Kr = [0.3 0.5 2]
    Fzy = Fs/(1+Kr*Fs); pstep(Fzy,150)
end
d=get(gca,'children');      set(d(1),'LineStyle','--')
set(d(2),'LineStyle','-'), set(d(3),'LineStyle',':')
legend('K_r=0.3','K_r=0.5','K_r=2.0')
```




```
- Permanent control deviation expected for input disturbances
- Higher values of Kr speed up the reaction and reduce the
  disturbance strength
```

Determine the expected output disturbance reaction:

```
figure(3), set(gcf,'color','white'), hold on
for Kr = [0.3 0.5 2]
    Fzi = 1/(1+Kr*Fs); pstep(Fzi,150)
end
d=get(gca,'children');      set(d(1),'LineStyle','--')
set(d(2),'LineStyle','-'), set(d(3),'LineStyle',':')
legend('K_r=0.3','K_r=0.5','K_r=2.0')
```




```
- No permanent control deviation expected for output disturbances
- Higher values of Kr speed up the reaction but increase
  oscillations
```

```
close all
```

Published with MATLAB® 7.4