

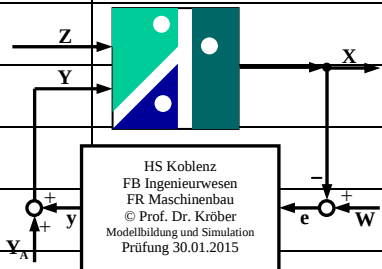
Modellbildung und Simulation WS 14/15
 Prof. Dr. W. Kröber

Zur Bewertung der Aufgaben muss der gesamte Lösungsweg ersichtlich sein.

- Bearbeitungszeit : 90 min
- Erlaubte Hilfsmittel :
 - selbstständige Lösung
 - keine weiteren Einschränkungen

Note : _____

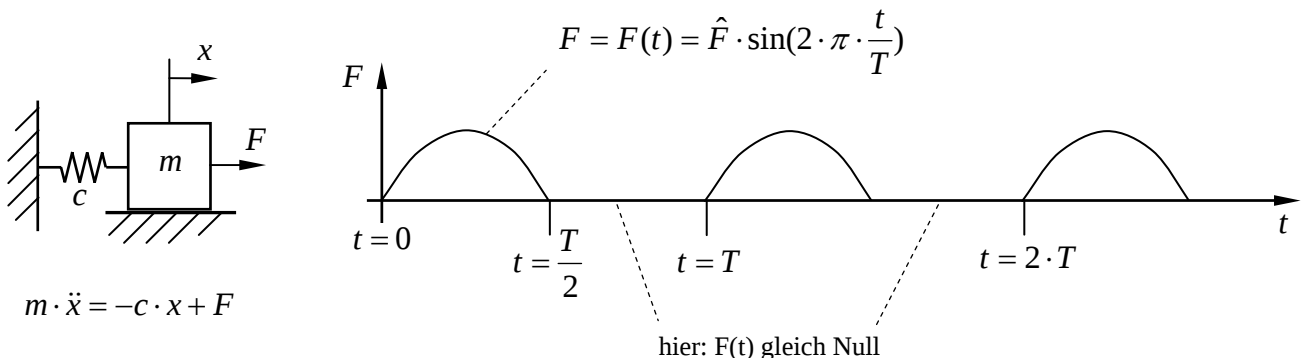
Aufgabe	erreichte Punkte
1	
2	
3	
4	
5	
6	
Summe	



Hinweis zum Abspeichern: Die abzuspeichernden Ergebnisdateien beginnen mit den sechs Buchstaben *nnnvvv*. Dabei steht *nnn* für die ersten drei Buchstaben des Nachnamens und *vvv* für die ersten drei Buchstaben des Vornamens.

Aufgabe 1 (16P) Ergebnisdateien → *nnnvvv1.m* und *nnnvvv1_sl.slx*

Auf das abgebildete Feder-Masse-System wirkt eine äußere Kraft F . Der zeitliche Verlauf der Kraft ist eine Sinusfunktion, wobei nur die "positive Halbwelle" ungleich Null ist und die sogenannte "negative Halbwelle" stets gleich Null ist. Der abgebildete Kraftverlauf wiederholt sich nach jedem Zeitintervall T (Periodendauer T). Die Anfangsauslenkung und Anfangsgeschwindigkeit der Koordinate x sind gleich Null.



Die Daten sollen in MATLAB vorgegeben werden. Die weitere Lösung soll komplett in SIMULINK erfolgen. In einem Scope-Block soll die Ausgabe des zeitlichen Verlaufes von x erfolgen. Die Simulationsdauer soll 5 Sekunden betragen. Verwenden Sie als festen Zeitschritt $\Delta t = 1$ ms. Das vorgeschlagene Lösungsverfahren zur Lösung der Differentialgleichungen (ode3 Bogacki-Shampine) soll so belassen werden.

Geg.: $T = 0,25$ s; $m = 3$ kg; $c = 7580$ N/m; $\hat{F} = 60$ N

Hinweis für "Wissensdurstige": Die Erregung erfolgt (etwa) mit der Hälfte der Eigenfrequenz. Die Oberschwingung $i=2$ des Kraftverlaufes erzeugt einen Resonanzeffekt.

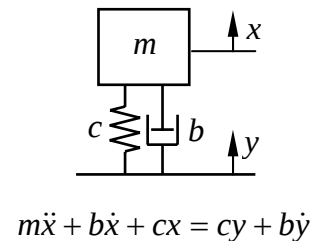
Aufgabe 2 (18P)

Ergebnisdateien → *nnnvvv2.m* und *nnnvvv2_sl.slx*

Bei der hier zu untersuchenden Fußpunkterregung ändert sich die Koordinate y sinusförmig. Der Verlauf von y bzw. $\dot{y}$ kann beschrieben werden durch:

$$y = \hat{y} \cdot \sin(\omega t) \quad \dot{y} = \hat{y} \cdot \omega \cdot \cos(\omega t)$$

Die Anfangsauslenkung und Anfangsgeschwindigkeit der Koordinate x sind gleich Null. Ein möglicher Einfluss der Erdbeschleunigung wird nicht betrachtet.



Die Daten sollen in MATLAB vorgegeben werden. Die Simulation erfolgt in SIMULINK. Die graphische Ausgabe von $x(t)$ soll wieder in MATLAB erfolgen. Die Simulationsdauer soll 10 Sekunden betragen. Verwenden Sie als festen Zeitschritt $\Delta t = 1$ ms. Das vorgeschlagene Lösungsverfahren zur Lösung der Differentialgleichungen (ode3 Bogacki-Shampine) soll so belassen werden.

Die Zahlenwerte sind so gewählt, dass sich etwa in der ersten Hälfte des Simulationszeitraumes ein Einschwingvorgang ergibt. In der zweiten Hälfte ist das System eingeschwungen (stationärer Zustand). Der Maximalausschlag und Minimalausschlag von $x(t)$ im Zeitraum $8s \leq t \leq 10s$ soll im Workspace in einer geeigneten Variable ersichtlich sein.

Geg.: $\hat{y} = 0,05m$; $\omega = 11,314 \text{ s}^{-1}$; $m = 50 \text{ kg}$; $b = 120 \text{ Ns/m}$; $c = 3200 \text{ N/m}$

Hinweis für "Wissensdurstige": Für $\frac{\omega}{\omega_0} = \sqrt{2}$ gilt bei der Fußpunkterregung $|G| = \frac{\hat{x}}{\hat{y}} = 1$.

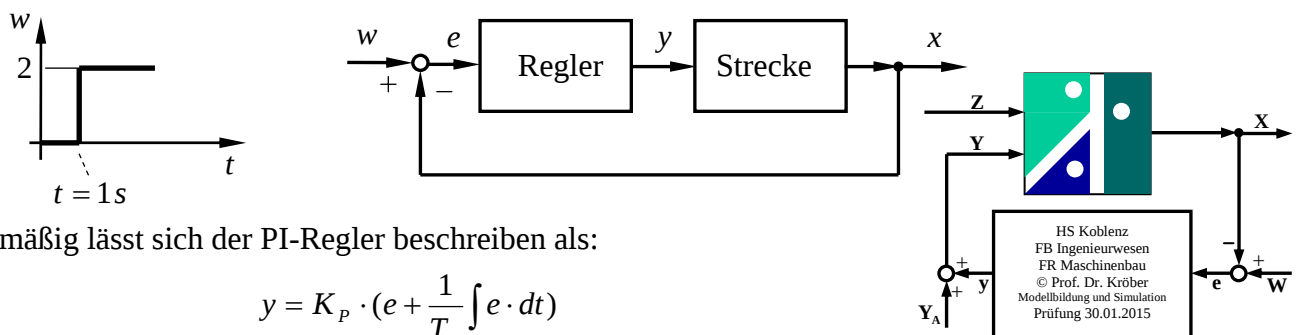
Aufgabe 3 (18P)

Ergebnisdateien → *nnnvvv3.m* und *nnnvvv3_sl.slx*

Die Durchflussregelstrecke im REG-Labor kann durch folgenden Frequenzgang beschrieben werden:

$$G_s = \frac{K}{(1 + j\omega \cdot T_1) \cdot (1 + j\omega T_2)} \cdot e^{-j\omega T_t} = \frac{K}{1 + j\omega \cdot T_1} \cdot \frac{1}{1 + j\omega T_2} \cdot e^{-j\omega T_t}$$

Dies ist ein PT_2 -Glied mit nachgeschalteter Totzeit. Als Regler wird ein PI-Regler eingesetzt. Hier soll ein Sollwertsprung bei $t = 1s$ betrachtet werden (Sollwertsprung von 0 auf den Wert 2). Alle Anfangsbedingungen seien gleich Null.



Formelmäßig lässt sich der PI-Regler beschreiben als:

$$y = K_p \cdot (e + \frac{1}{T_n} \int e \cdot dt)$$

Geg.: $K=1,2$; $T_1 = 1 \text{ s}$; $T_2 = 0,3 \text{ s}$; $T_t = 0,3 \text{ s}$; $K_p = 0,5$; $T_n = 1s$

Nach der Vorgabe der Daten in MATLAB soll die weitere Lösung komplett in SIMULINK erfolgen. In einem gemeinsamen Scope-Block soll w , y , und x graphisch dargestellt werden. Die Simulationsdauer soll 8 Sekunden betragen. Verwenden Sie als festen Zeitschritt $\Delta t = 10$ ms. Das vorgeschlagene Lösungsverfahren zur Lösung der Differentialgleichungen (ode3 Bogacki-Shampine) soll so belassen werden.

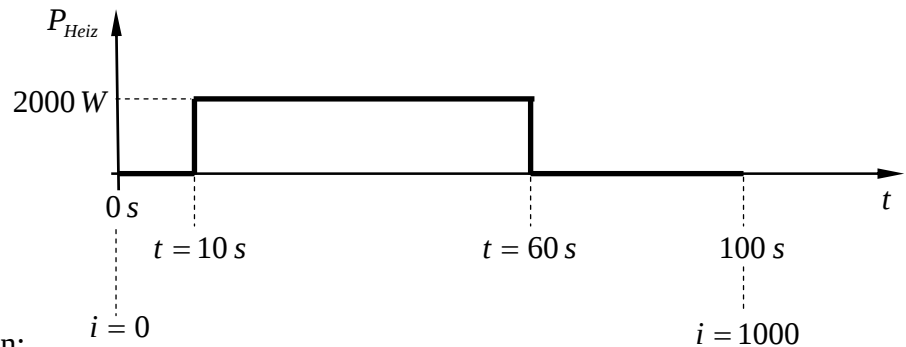
Aufgabe 4 (16P)

Ergebnisdatei → *nnnvvv4.xlsm*

Die folgende Aufgabe soll in einem Excel-Makro gelöst werden.

Einem Wasserbad wird elektrische Heizleistung zugeführt. Die Temperatur des Wasserbades wird mit einem Temperatursensor gemessen. Betrachtet wird ein Zeitraum von 100 s. Als Zeitschritt soll $\Delta t = 0,1$ s gewählt werden. Dazu sind 1000 Zeitschritte erforderlich. Zu bestimmen ist die Temperatur des Wasserbades ϑ_W und die angezeigte Temperatur ϑ_F des Temperaturfühlers.

Heizleistung über der Zeit:



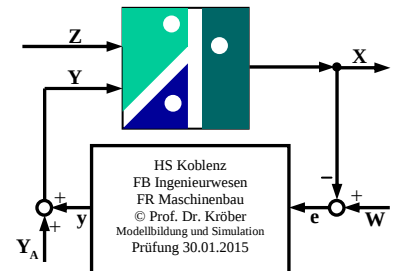
Zugrunde zu legende Gleichungen:

$$P_{Heiz} = m \cdot c \cdot \frac{d\vartheta_W}{dt} \quad \rightarrow \quad \vartheta_{W\ i+1} = \vartheta_{W\ i} + \frac{P_{Heiz}}{m \cdot c} \cdot \Delta t$$

$$\vartheta_F + T \cdot \frac{d\vartheta_F}{dt} = \vartheta_W \quad \rightarrow \quad \vartheta_{F\ i+1} = \vartheta_{F\ i} \cdot \left(1 - \frac{\Delta t}{T}\right) + \vartheta_{W\ i} \cdot \frac{\Delta t}{T}$$

Weitere gegebene Zahlenwerte:

$m = 1,2$ kg; $c = 4183$ J/(kg·K); $T = 10$ s; Anfangstemperatur sei 20°C



In der Excel-Spalte 1 soll die Zeit stehen, in Spalte 2 die Temperatur ϑ_W und in Spalte 3 die Temperatur ϑ_F . Eine graphische Darstellung ist nicht erforderlich aber zur Ergebniskontrolle hilfreich. Das Makro soll mit der Tastenkombination „Strg y“ gestartet werden.

Aufgabe 5 (16P)

Ergebnisdatei → *nnnvvv5.vi*

Grundsätzlich soll hier das gleiche Problem wie in Aufgabe 4 bearbeitet werden. Die Lösung soll in einem Formelknoten in LabVIEW gelöst werden. Die Kurvenverläufe sollen hier (nur) graphisch in einem gemeinsamen XY-Schrieb dargestellt werden (Beide Temperaturverläufe über der Zeit).

Aufgabe 6 (16P)

Ergebnisdatei → *nnnvvv6.vi*

Mit dem USB6008 ist durch das Steckmodul ein NTC-Widerstand angeschlossen. Als Ausgangspunkt kann ein vorhandenes Programmbeispiel verwendet werden, bei dem die Temperatur [in $^\circ\text{C}$] in Funktion der Zeit angezeigt wird.

- Das Programm ist dahingehend zu ergänzen: Es soll ein zusätzliches numerisches Ausgabefenster eingefügt werden in dem die momentane 1. Ableitung der Temperatur [in $^\circ\text{C}/\text{sec}$] angezeigt wird.

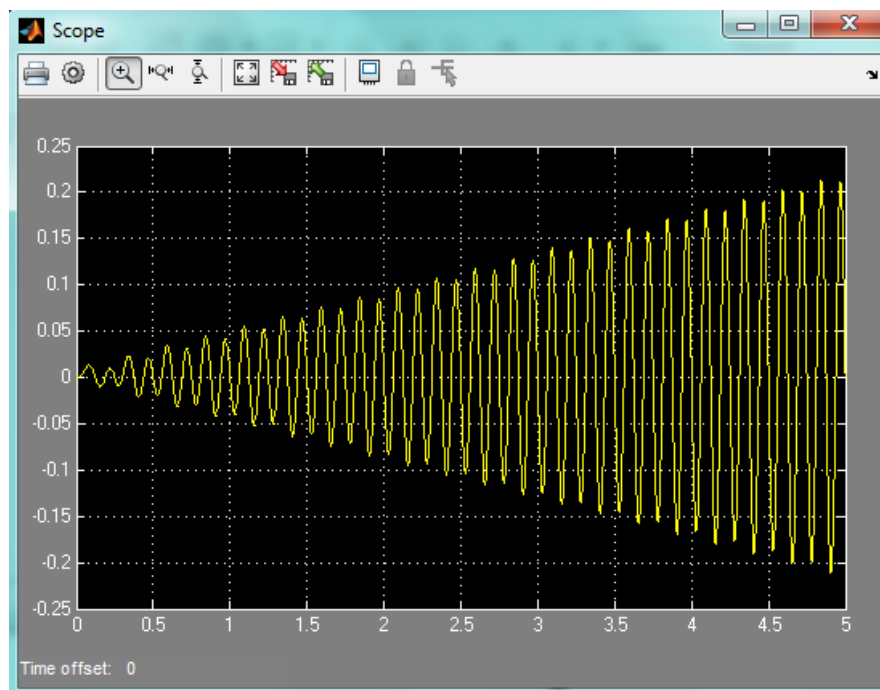
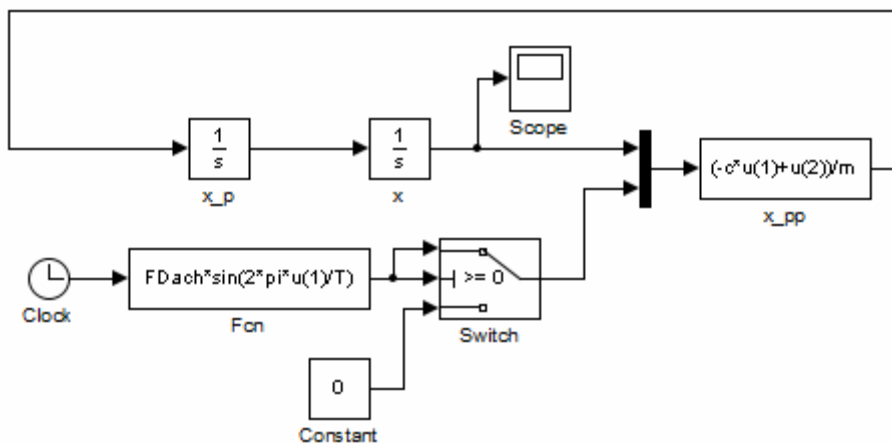
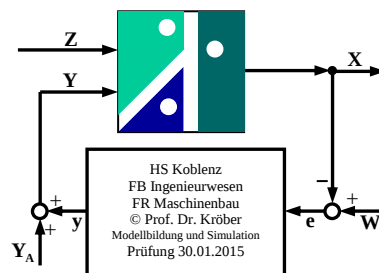
$$\text{Hilfestellung: 1. Ableitung} = \frac{\Delta \vartheta}{\Delta t} = \frac{\vartheta_{\text{aktuell}} - \vartheta_{\text{zuvor}}}{\Delta t} = \frac{\vartheta_i - \vartheta_{i-1}}{\Delta t}$$

- In einem weiteren Fenster soll die maximale Steigung seit Programmstart angezeigt werden (hier nur positive Steigungen, beim Aufheizvorgang).

Nebensächliche Bemerkung: Das Programm wird z.B. durch Betätigen der Stop-Schaltfläche beendet.

Prüfung 30.01.2015 / Lösung zur Aufgabe 1

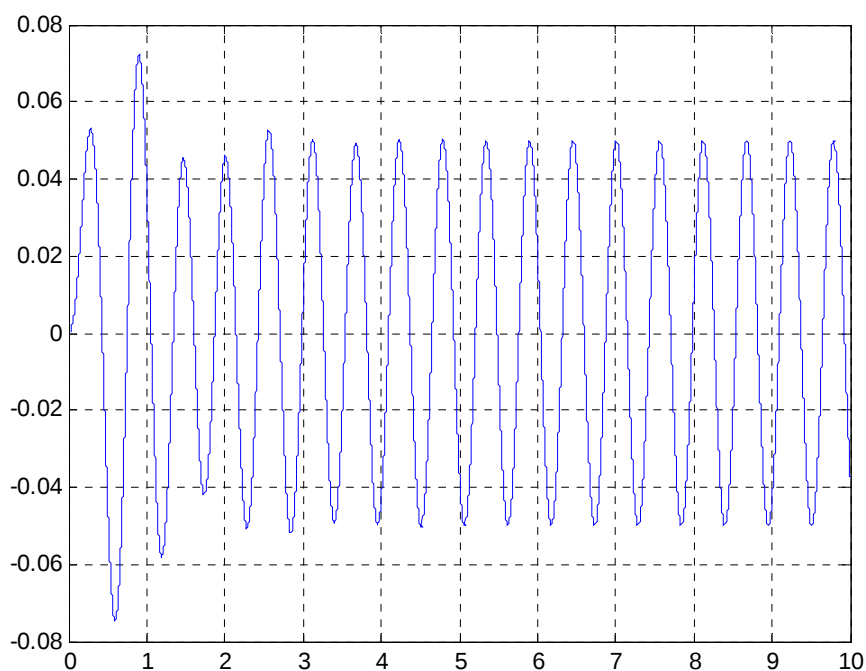
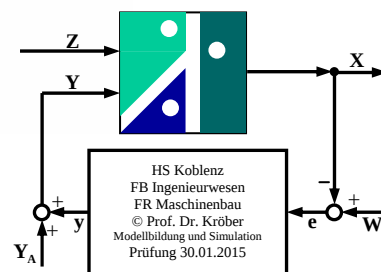
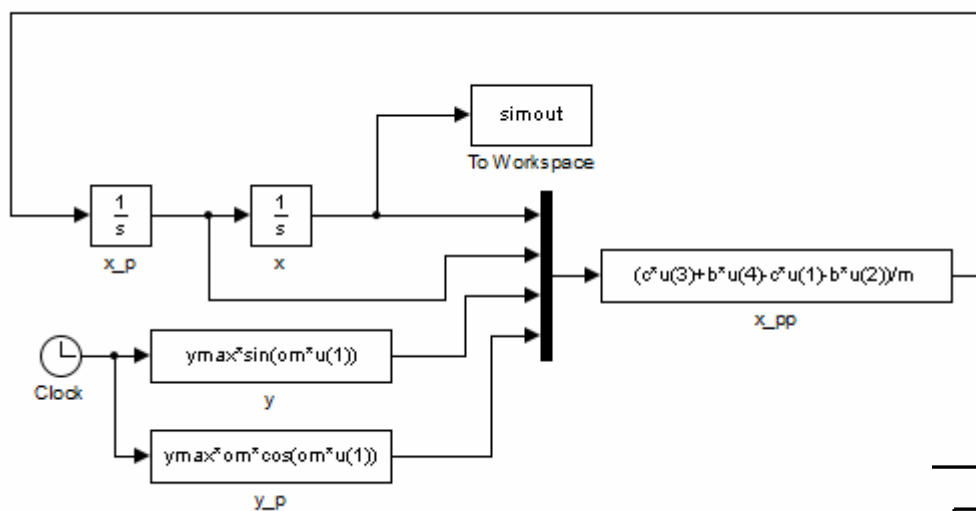
```
clear;clc;
T=0.25;
m=3;
c=7580;
FDach=60;
dt=0.001;
tsim=5;
sim('krowoll1_sl'); % optional
```



Prüfung 30.01.2015 / Lösung zur Aufgabe 2

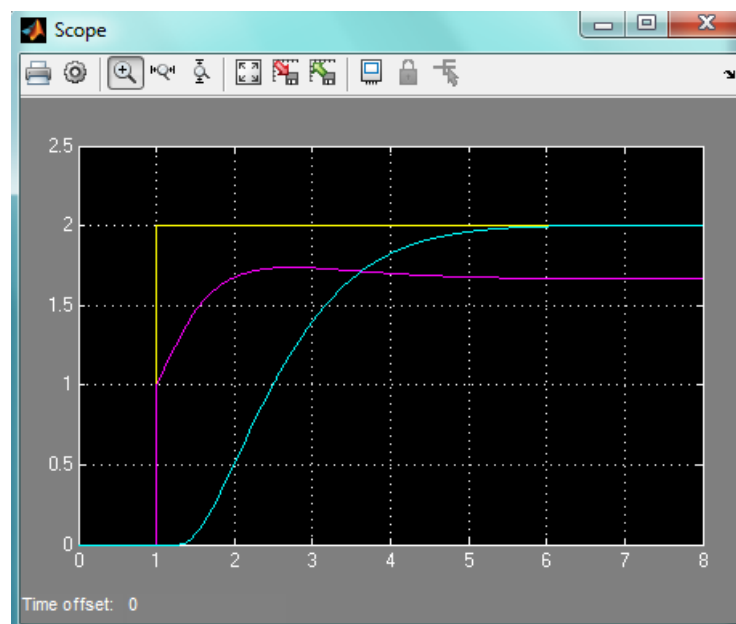
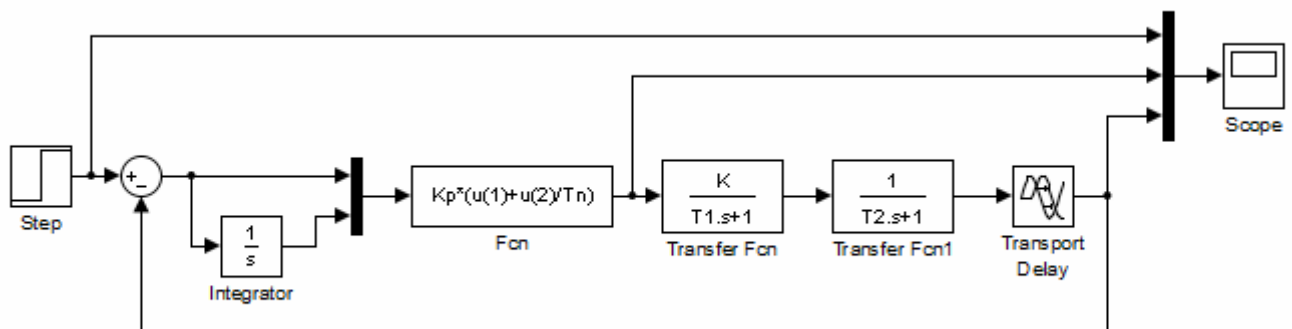
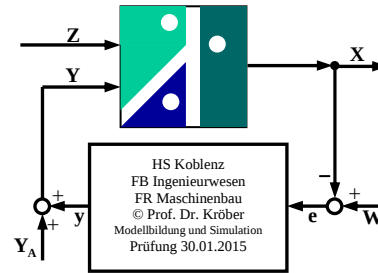
```
clear;clc;
ymax=0.05;
om=11.314;
m=50;
b=120;
c=3200;
dt=0.001;
tsim=10;
sim('krowol2_sl');
t=simout.time;
x=simout.signals.values;
plot(t,x);grid;
x_max=max(x(8001:10001));
x_min=min(x(8001:10001));
```

Workspace			
Name	Value	Min	Max
b	120	120	120
c	3200	3200	3200
dt	1.0000e-03	1.0000...	1.0000...
m	50	50	50
om	11.3140	11.3140	11.3140
simout	1x1 struct		
t	10001x1 double	0	10
tout	1000x1 double	9.0010	10
tsim	10	10	10
x	10001x1 double	-0.0746	0.0721
x_max	0.0500	0.0500	0.0500
x_min	-0.0500	-0.0500	-0.0500
ymax	0.0500	0.0500	0.0500



Prüfung 30.01.2015 / Lösung zur Aufgabe 3

```
clear;clc;
K=1.2;
T1=1;
T2=0.3;
Tt=0.3;
Kp=0.5;
Tn=1;
dt=0.01;
tsim=8;
sim('krowol3_sl'); % optional
```



```

Sub Makrol1()
'
' Makrol Makro
'
' Tastenkombination: Strg+y
'

    Dim tw(1000), tf(1000) As Double
    m = 1.2
    c = 4183
    T = 10
    dt = 0.1

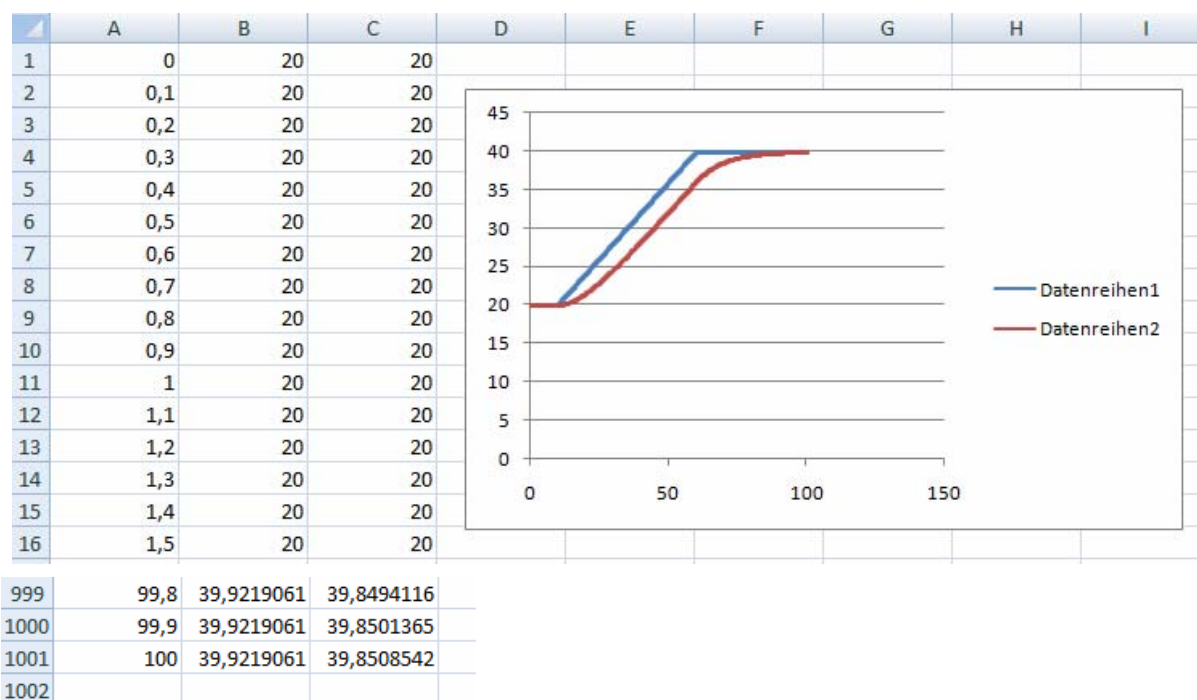
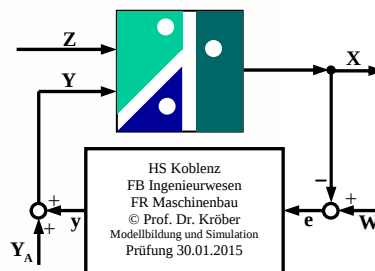
' Wasserbad
    tw(0) = 20
    For i = 0 To 999
        PHeiz = 0
        If (i >= 100) Then PHeiz = 2000
        If (i >= 600) Then PHeiz = 0
' oder If ((i >= 100) And (i < 600)) Then PHeiz = 2000
        tw(i + 1) = tw(i) + PHeiz * dt / (m * c)
    Next i

' Temperaturfühler
    tf(0) = 20
    For i = 0 To 999
        tf(i + 1) = tf(i) * (1 - dt / T) + tw(i) * dt / T
    Next i

' Übergabe
    For i = 0 To 1000
        Cells(i + 1, 1) = i * dt
        Cells(i + 1, 2) = tw(i)
        Cells(i + 1, 3) = tf(i)
    Next i

End Sub

```



```

float t[1001],tw[1001],tf[1001];
float PHeiz;
float m=1.2;
float c=4183;
float T=10;
float dt=0.1;
int i;

// Zeitachse
for(i=0;i<1001;i++)t[i]=i*dt;

// Wasserbad
tw[0]=20;
for(i=0;i<1000;i++)
{
PHeiz=0;
if(i>=100)PHeiz=2000;
if(i>=600)PHeiz=0;
// oder if((i >=100)&&(i<600))PHeiz=2000;
tw[i+1]=tw[i]+PHeiz*dt/(m * c);
}

// Temperaturfühler
tf[0]=20;
for(i=0;i<1000;i++)tf[i+1]=tf[i]*(1-dt/T)+tw[i]*dt/T;
    
```

