

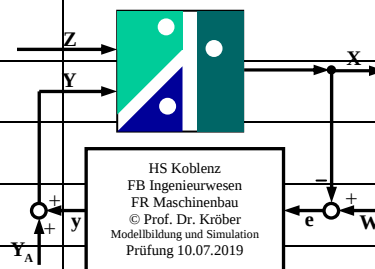
Modellbildung und Simulation SS 2019
 Prof. Dr. W. Kröber

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	

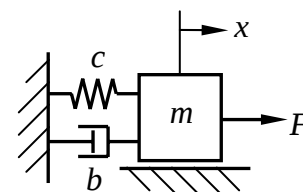
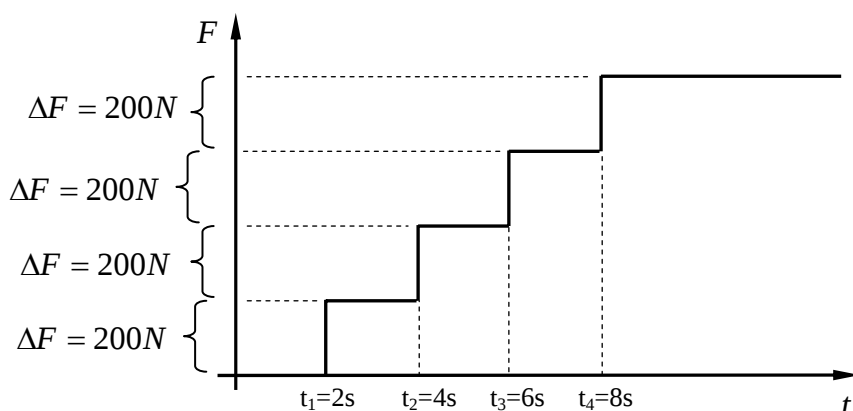


Hinweis zum Abspeichern: Die abzuspeichernden Ergebnisdateien beginnen mit den sechs Buchstaben *nnnvvv*. Dabei steht *nnn* für die ersten drei Buchstaben des Nachnamens und *vvv* für die ersten drei Buchstaben des Vornamens.

Aufgabe 1 (17P)

Ergebnisdateien → *nnnvvv1.m* und *nnnvvv1_sl.slx*

Der zeitliche Verlauf einer Kraft ist in der Abbildung dargestellt. Diese Kraft wirkt auf eine Masse *m*. Die Masse *m* ist mit einer Feder und einem Dämpferelement zur Umgebung abgestützt. Mit einem Simulink-Programm soll der Schwingweg *x* bestimmt werden. Zu Beginn der Untersuchung ist die Masse *m* in Ruhe und die Auslenkung ist gleich Null.



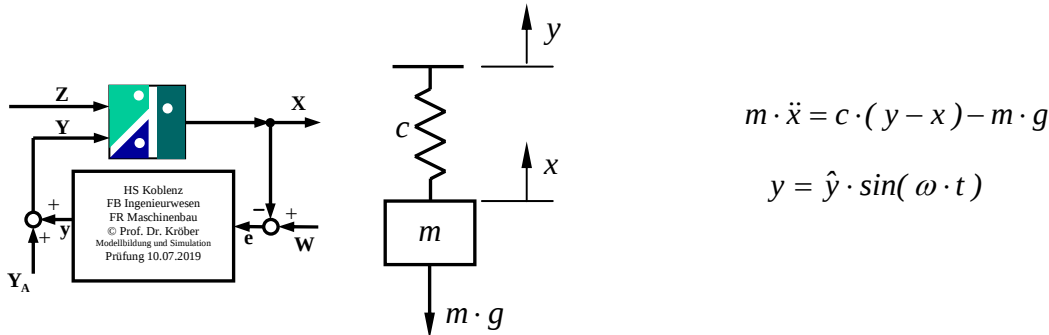
$$m\ddot{x} + b\dot{x} + cx = F$$

Weitere gegebene Daten: $m = 30 \text{ kg}$; $c = 8000 \text{ N/m}$; $b = 150 \text{ N/(m/s)}$; $\Delta t = 0,001 \text{ s}$; $t_{\text{sim}} = 12 \text{ s}$;

Nach Vorgabe der Daten in MATLAB soll die Lösung in Simulink erfolgen. Die graphische Darstellung soll wieder in MATLAB realisiert werden. Die Simulationsdauer und feste Schrittweite werden in MATLAB vorgegeben. Der Plot soll aus zwei Teilen bestehen. Im oberen Teil soll die Kraft *F* in Abhängigkeit der Zeit *t* dargestellt werden, im unteren Teil der Schwingweg *x* in Abhängigkeit der Zeit. Wählen Sie als Lösungsverfahren zur Lösung der Differentialgleichungen "ode3 Bogacki-Shampine".

Aufgabe 2 (17P) Ergebnisdateien → *nnnvvv2.m* und *nnnvvv2_sl.slx*

An einem Aufhängepunkt (Wegkoordinate y) hängt über eine Feder eine Masse m (Wegkoordinate x). Zu Beginn der Betrachtung ist $x = 0$ und $\dot{x} = 0$. Der obere Aufhängepunkt wird sinusförmig bewegt. Die Simulation soll maximal 5 Sekunden laufen. Wenn der Weg x im Rechenverlauf größer ist als $x_{\max} = 0,2 \text{ m}$, soll die Simulation vorher schon stoppen.



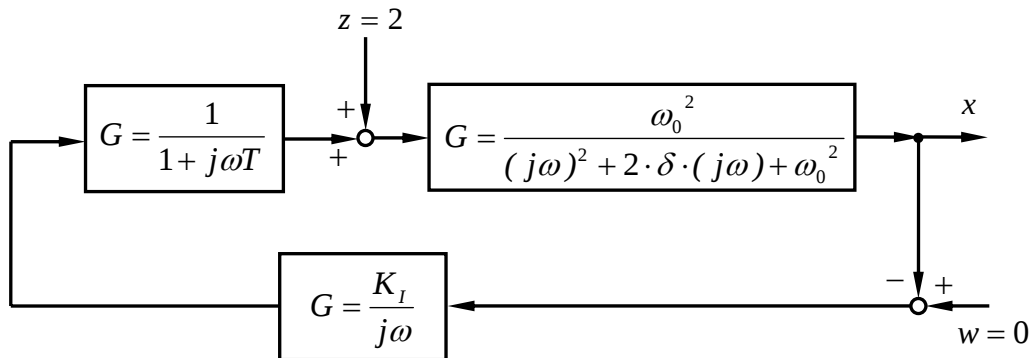
Zahlenwerte:

$m = 0,5 \text{ kg}$; $g = 9,81 \text{ m/s}^2$; $c = 1250 \text{ N/m}$; $\hat{y} = 0,002 \text{ m}$; $\omega = 50 \text{ s}^{-1}$; $\Delta t = 0,001 \text{ s}$; $t_{\text{sim}} = 5 \text{ s}$

Nach Vorgabe der Daten in MATLAB soll die Lösung in Simulink erfolgen. Die graphische Darstellung soll wieder in MATLAB realisiert werden. Die Simulationsdauer und feste Schrittweite werden in MATLAB vorgegeben. In dem Plot soll der Weg x in Abhängigkeit der Zeit dargestellt werden. Wählen Sie als Lösungsverfahren zur Lösung der Differentialgleichungen "ode3 Bogacki-Shampine".

Aufgabe 3 (18P) Ergebnisdateien → *nnnvvv3.m* und *nnnvvv3_sl.slx*

Bei dem folgenden Regelkreis sind zunächst alle Größen gleich Null, nur die Störgröße ist $z = 2$. Ermitteln Sie den Verlauf der Regelgröße x in Abhängigkeit der Zeit.



Weitere gegebene Daten: $T = 2 \text{ s}$; $\omega_0 = 0,3 \text{ s}^{-1}$; $\delta = 0,3 \text{ s}^{-1}$; $K_I = 0,04 \text{ s}^{-1}$; $t_{\text{sim}} = 100 \text{ s}$; $\Delta t = 0,01 \text{ s}$

Die Simulationsdauer und feste Schrittweite werden in MATLAB vorgegeben. Nach Vorgabe aller gegebenen Daten in MATLAB soll die Lösung in Simulink erfolgen. Die graphische Darstellung soll wieder in MATLAB realisiert werden. In der graphischen Darstellung soll x über der Zeit dargestellt werden. Wählen Sie als Lösungsverfahren zur Lösung der Differentialgleichungen "ode3 Bogacki-Shampine".

Aufgabe 4 (16P) Ergebnisdatei → *nnnvvv4.xlsm*

Der Temperaturverlauf in einem Behälter in Abhängigkeit der Zeit kann durch folgende Gleichung beschrieben werden:

$$\vartheta_{\text{Behälter}}(t) = \vartheta_{\text{Behälter}} = a - b \cdot t$$

In den Behälter wird ein Temperatursensor mit der Zeitkonstanten T eingetaucht.

Temperatursensor:

$$\vartheta + T \cdot \frac{d\vartheta}{dt} = \vartheta_{\text{Behälter}} \quad \text{Differenzenausdrücke eingesetzt und umgeformt:} \quad \vartheta_{i+1} = \vartheta_i + (\vartheta_{\text{Behälter}} - \vartheta_i) \cdot \frac{\Delta t}{T}$$

Der Temperatursensor hat zum Zeitpunkt $t = 0$ eine Temperatur von 90°C .

Weitere gegebene Daten: $a = 90^\circ\text{C}$; $b = 0,2^\circ\text{C/s}$; $T = 50\text{s}$; $\Delta t = 0,1\text{s}$

Berechnen Sie den Temperaturverlauf von ϑ in Abhängigkeit der Zeit! Die Zeit läuft von 0 bis 300 Sekunden. Erstellen Sie ein Makro, in dem die zeitlichen Verläufe der beiden Temperaturen berechnet werden. Das Makro soll mit der Tastenkombination „Strg y“ gestartet werden. In Spalte 1 des Excel-Sheets soll die Zeit stehen, in der zweiten Spalte die Werte $\vartheta_{\text{Behälter}}(t)$, in der weiteren Spalte $\vartheta(t)$. Es soll eine Graphik beider Temperaturen über der Zeit erstellt 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 beiden Temperaturen sollen in einem XY-Schrieb ausgegeben werden.

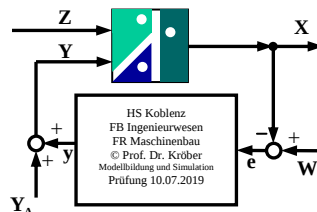
Aufgabe 6 (16P) Ergebnisdatei → *nnnvvv6.vi*

Mit dem USB6008 ist durch das Steckmodul ein NTC-Widerstand zur Bestimmung der Temperatur angeschlossen. Als Ausgangspunkt kann ein vorhandenes Programmbeispiel verwendet werden, bei dem die Temperatur in Funktion der Zeit angezeigt wird.

Das Programm ist dahingehend zu ergänzen: Es soll die momentane Temperatur 2 Sekunden nach dem Programmstart ($\vartheta_{2\text{sec}}$) und 3 Sekunden nach dem Programmstart ($\vartheta_{3\text{sec}}$) programmintern erfasst werden.

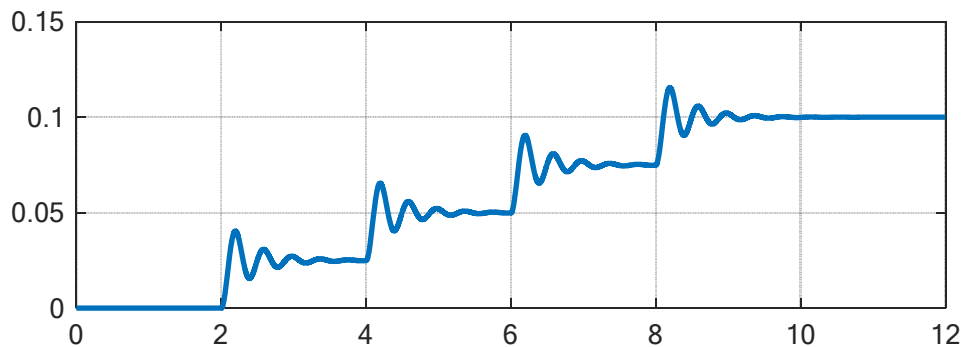
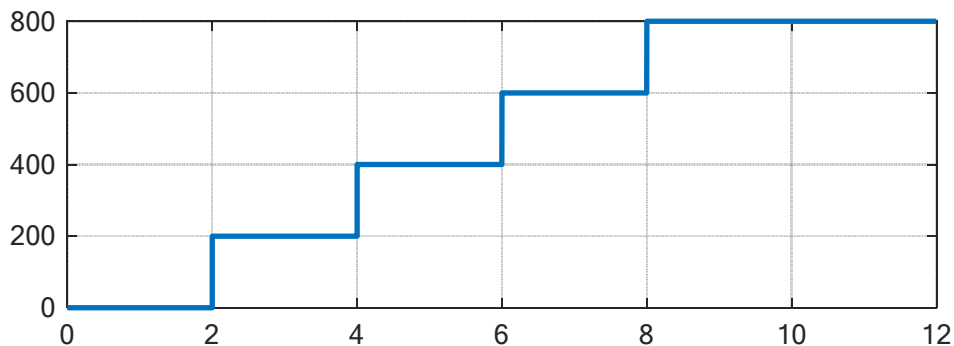
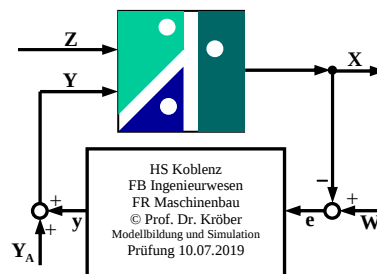
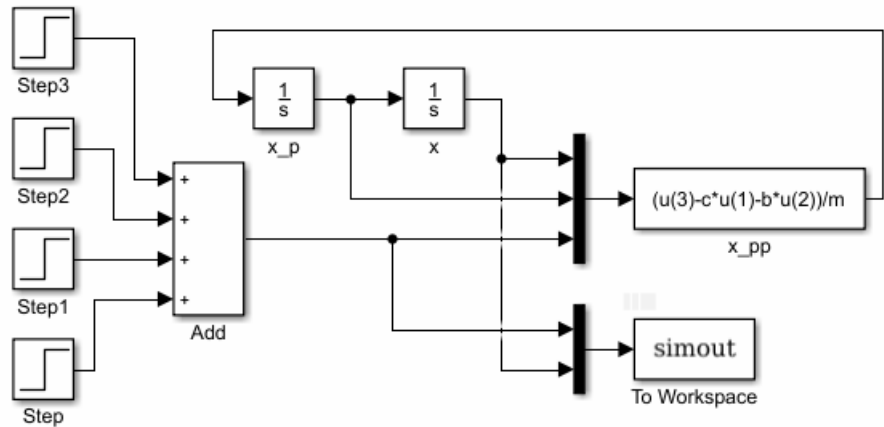
In einem numerischen Ausgabefenster soll die Differenz $\Delta\vartheta = \vartheta_{3\text{sec}} - \vartheta_{2\text{sec}}$ ausgegeben werden. Bevor der Differenzwert gültig ist, soll ein Wert von Null ausgegeben werden.

Sinn: Damit hat man eine Kenngröße, ob und wie stark sich die Temperatur in diesem Zeitraum geändert hat (positiv oder auch negativ).



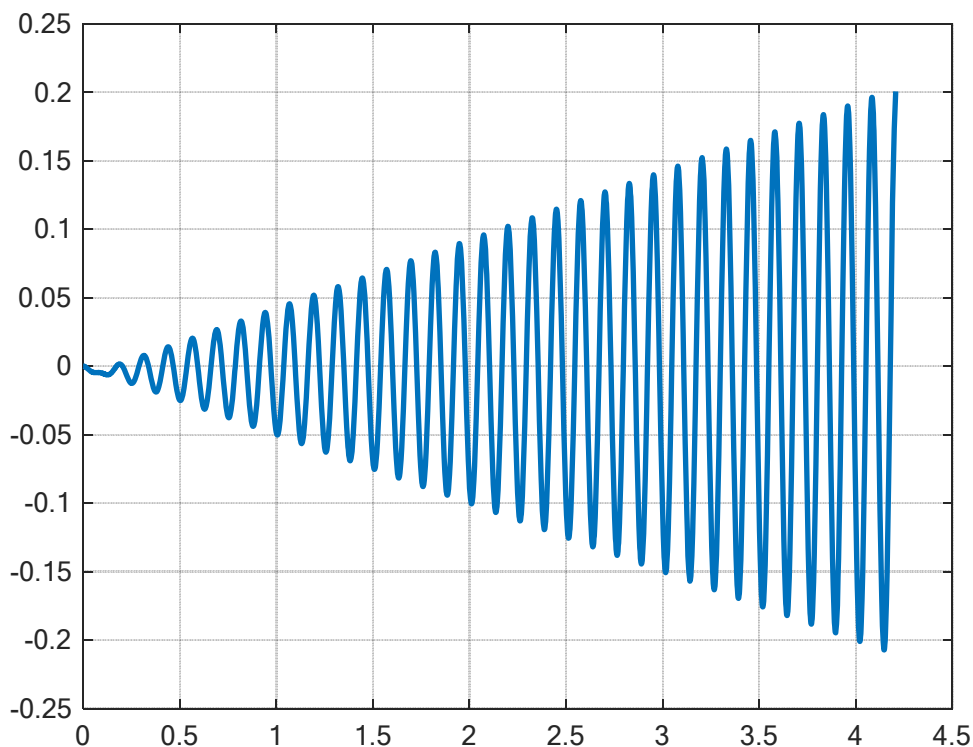
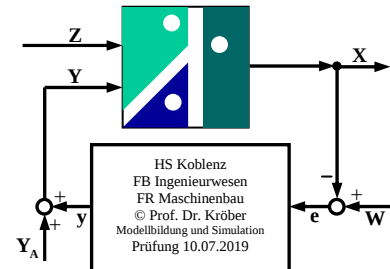
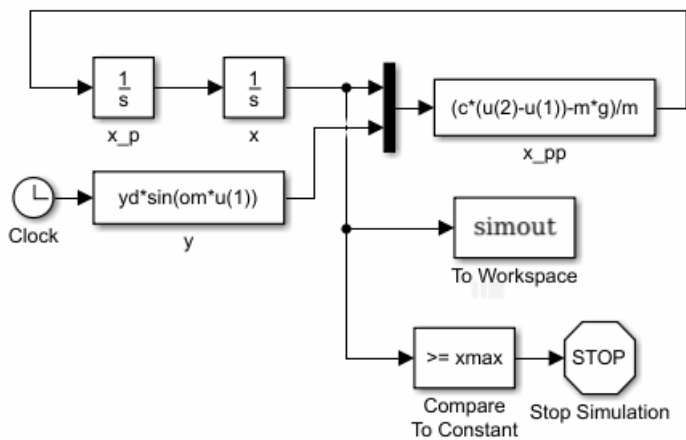
```
clear;clc;
dF=200;
t1=2;
t2=4;
t3=6;
t4=8;
m=30;
c=8000;
b=150;

dt=0.001;
tsim=12;
sim('krowol1_sl');
t=simout.time;
F=simout.signals.values(:,1);
x=simout.signals.values(:,2);
subplot(2,1,1);
plot(t,F,'LineWidth',2);grid;
subplot(2,1,2);
plot(t,x,'LineWidth',2);grid;
```



```
clear;clc;
m=0.5;
g=9.81;
c=1250;
yd=0.002;
om=50;
xmax=0.2;
dt=0.001;
tsim=5;
sim('krowol2_sl');
t=simout.time;
x=simout.signals.values;
plot(t,x,'LineWidth',2);grid;
```

Workspace	
Name ^	Value
c	1250
dt	1.0000e-03
g	9.8100
m	0.5000
om	50
simout	1x1 struct
t	4208x1 double
tout	4208x1 double
tsim	5
x	4208x1 double
xmax	0.2000
yd	0.0020




```

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

    Dim tetb(3000), tet(3000) As Double
    a = 90
    b = 0.2
    dt = 0.1
    T = 50

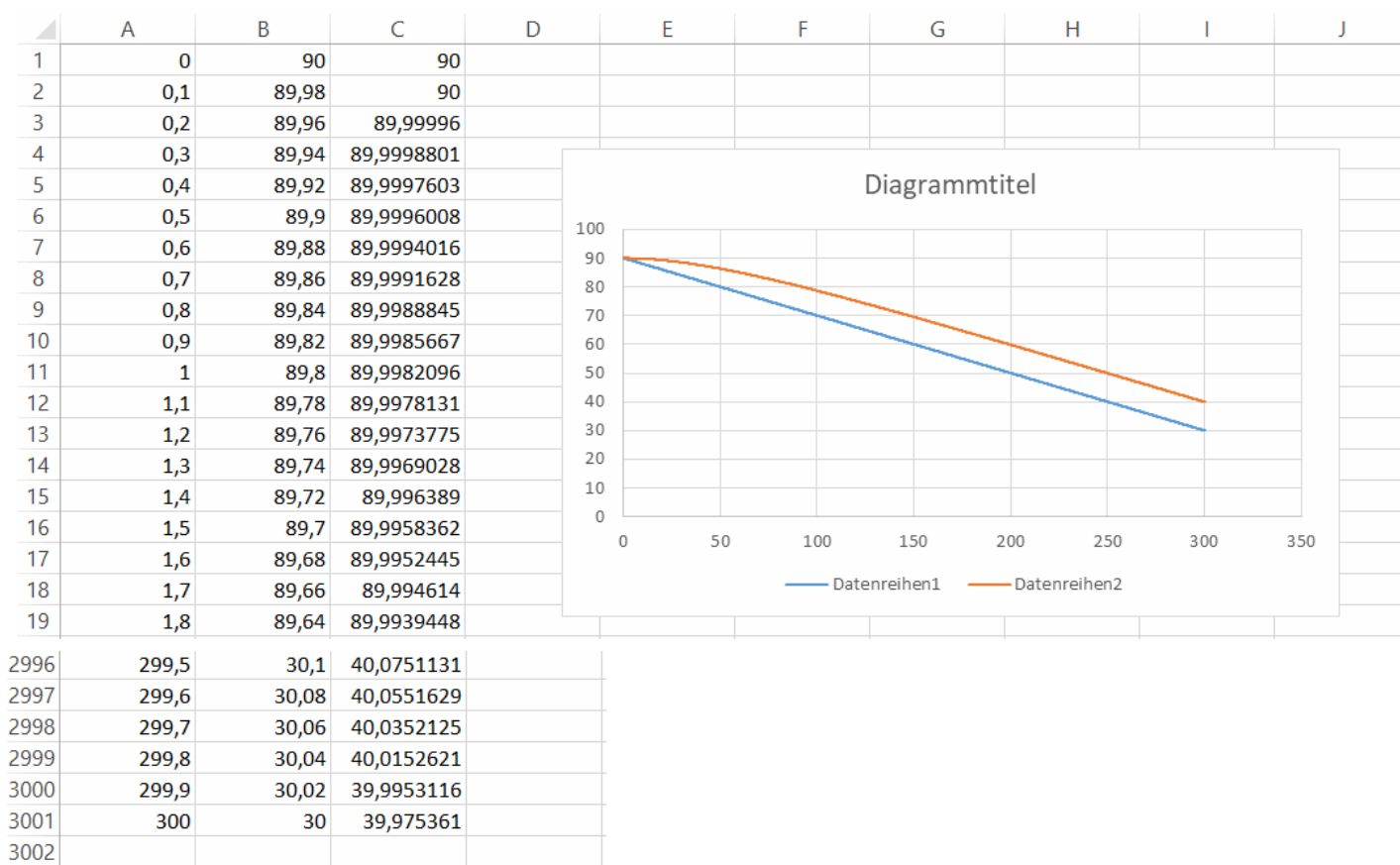
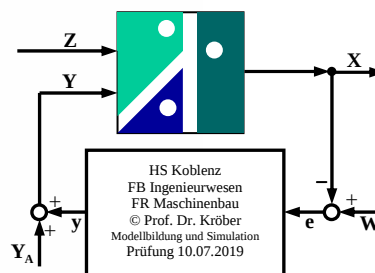
    For i = 0 To 3000
        tetb(i) = a - b * i * dt
    Next i

    tet(0) = 90
    For i = 0 To 2999
        tet(i + 1) = tet(i) + (tetb(i) - tet(i)) * dt / T
    Next i

    For i = 0 To 3000
        Cells(i + 1, 1) = i * dt
        Cells(i + 1, 2) = tetb(i)
        Cells(i + 1, 3) = tet(i)
    Next i

End Sub

```



```
float t[3001],tetb[3001],tet[3001];
float a=90;
float b=0.2;
float dt=0.1;
float T=50;
int i;

for(i=0;i<3001;i++)t[i]=i*dt;

for(i=0;i<3001;i++)tetb[i]=a-b*t[i];

tet[0]=90;
for(i=0;i<3000;i++)
{
    tet[i+1]=tet[i]+(tetb[i]-tet[i])*dt/T;
}
```

