

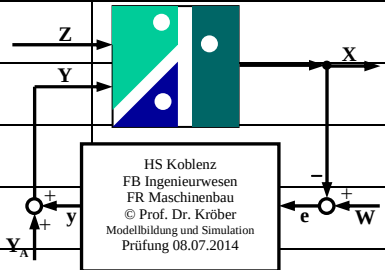
Modellbildung und Simulation SS 2014
 Prof. Dr. W. Kröber

Zur Bewertung der Aufgaben muss der gesamte Lösungsweg ersichtlich sein.

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

Note : _____

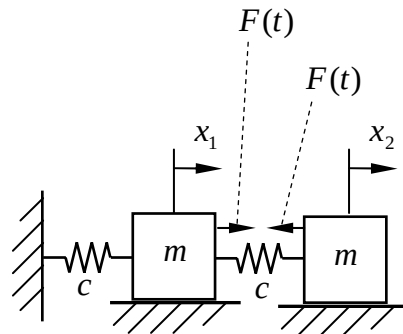
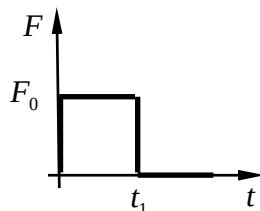
Aufgabe	erreichte Punkte
1	
2	
3	
4	
5	
6	
Summe	



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*

Das abgebildete Schwingungssystem hat die beiden Freiheitsgrade x_1 und x_2 . Die Anfangsauslenkungen sowie die Anfangsgeschwindigkeiten sind gleich Null. Durch einen Aktuator wirkt zwischen den beiden Massen eine Kraft $F = F(t)$ mit dem folgenden Zeitverlauf:



Die Differentialgleichungen zur Beschreibung des Bewegungsablaufes lauten:

$$m \cdot \ddot{x}_1 = -c \cdot x_1 - c \cdot (x_1 - x_2) + F$$

$$m \cdot \ddot{x}_2 = -c \cdot (x_2 - x_1) - F$$

Die Daten sollen in MATLAB vorgegeben werden. Die weitere Lösung soll komplett in SIMULINK erfolgen. Die Ausgabe vom zeitlichen Verlauf von x_1 und x_2 erfolgt graphisch in einem Scope-Block. Die Simulationsdauer soll 2 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.: $m = 4$ kg; $c = 200$ N/m; $F_0 = 20$ N; $t_1 = 0,25$ s

Aufgabe 2 (18P)

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

Ein Ball fällt aus einer Anfangshöhe y_0 auf eine Unterlage. Nach einem ersten Kontakt mit dem Boden hebt der Ball wieder ab.

Die Kräftebilanz an dem Ball ergibt:

$$m \cdot \ddot{y} = -m \cdot g + F_{Bod}$$

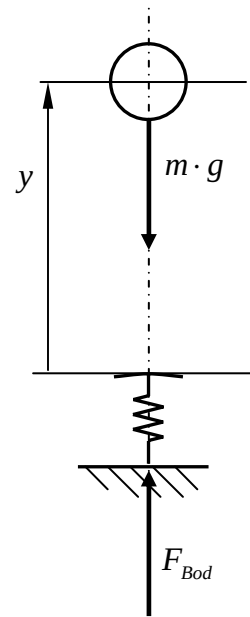
Für den Bodenkontakt wird folgendes nichtlineare Kraft-Weg-Gesetz zugrunde gelegt:

$$y > 0: F_{Bod} = 0$$

$$y \leq 0: F_{Bod} = k \cdot (-y)^{1,5}$$

Eventuell vorhandene Reibungs- und Dämpfungskräfte werden nicht berücksichtigt. Die Daten sind so bemessen, dass im betrachteten Zeitraum nur ein Bodenkontakt stattfindet.

Die Daten sollen in MATLAB vorgegeben werden. Die Simulation erfolgt in SIMULINK. Der graphische Verlauf $y(t)$ soll wieder in MATLAB erfolgen. Zusätzlich sollen in MATLAB die Anzahl der Zeitschritte ermittelt werden, in denen die Bodenkraft größer als Null ist.



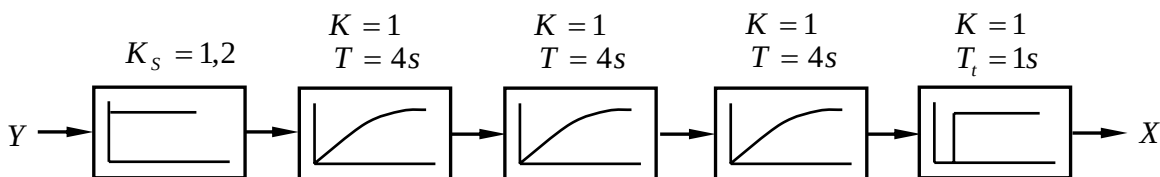
Die Simulationsdauer soll 1 Sekunde betragen, der feste Zeitschritt $\Delta t = 0,1$ ms. Das vorgeschlagene Lösungsverfahren zur Lösung der Differentialgleichungen (ode3 Bogacki-Shampine) soll so belassen werden.

Geg.: $y_0 = 0,5$ m; $m = 0,8$ kg; $g = 9,81$ m/s²; $k = 1000$ N/m^{1,5}

Aufgabe 3 (18P)

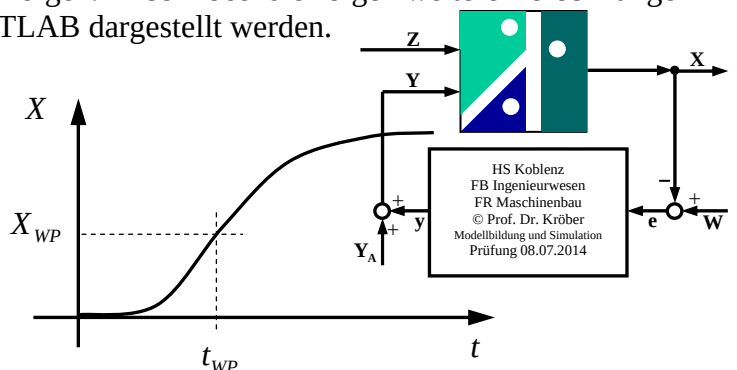
Ergebnisdateien → *nnnvvv3.m* und *nnnvvv3_sl.slx*

Von der abgebildeten Regelstrecke wird im REG-Labor die Sprungfunktion aufgenommen. Die dort durchgeführte Auswertung soll hier teilweise nachvollzogen werden.



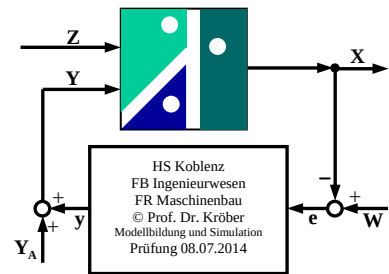
Der Verlauf von $Y(t)$ ist eine Sprungfunktion. Konkret: $Y=0$ für $t < 0$ und $Y=1$ für $t \geq 0$. Nach einer Parametervorgabe in MATLAB (alle Zahlenwerte von 1 müssen nicht aus MATLAB nach SIMULINK übergeben werden) soll die Lösung in SIMULINK erfolgen. Anschließend erfolgen weitere Berechnungen in MATLAB. Der Verlauf $X(t)$ soll ebenfalls in MATLAB dargestellt werden.

Der Verlauf von $X(t)$ ist eine Funktion mit einem Wendepunkt. In dem Wendepunkt ist die Steigung am größten. Bei welcher Zeit t_{WP} liegt der Wendepunkt vor und wie groß ist der Funktionswert X_{WP} an diesem Wendepunkt?



Verwenden Sie als Steigung (z.B.) $\dot{X} \approx \frac{X_{i+1} - X_{i-1}}{2 \cdot \Delta t}$

Die Simulationsdauer soll 40 Sekunden betragen, der feste Zeitschritt $\Delta t = 0,01$ Sekunden. Das vorgeschlagene Lösungsverfahren zur Lösung der Differentialgleichungen (ode3 Bogacki-Shampine) soll so belassen werden.

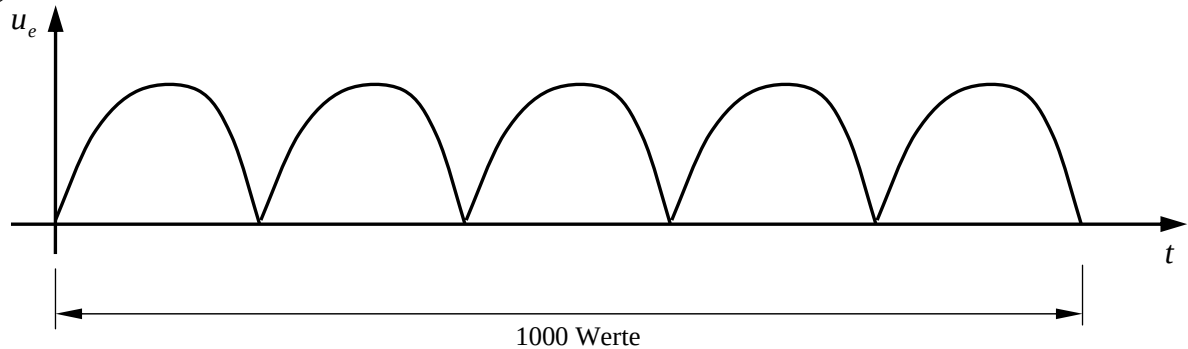


Aufgabe 4 (16P)

Ergebnisdatei → *nnnvvv4.xlsm*

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

Auf einen Tiefpassfilter (einfaches RC-Glied) wirkt eine gleichgerichtete Sinusfunktion als Eingangsgröße. Betrachtet werden soll der Fall, dass die Eingangsgröße 5 volle Halbschwingungen durchläuft (siehe Skizze).



Formeln und Hilfestellungen zur Lösung:

$$u_e = \hat{u}_e \cdot |\sin(\omega \cdot t)| \quad \text{wobei} \quad \hat{u}_e = 2V \quad \text{und} \quad \sin(\omega \cdot t) = \sin\left(i \cdot \frac{2 \cdot \pi}{1000} \cdot 2,5\right)$$

Für den Tiefpassfilter gilt die Rekursionsgleichung: $u_{ai+1} = \left(1 - \frac{\Delta t}{RC}\right) \cdot u_{ai} + \left(\frac{\Delta t}{RC}\right) \cdot u_{ei}$

Weitere Daten: $R = 10 \text{ k}\Omega$; $C = 10 \mu\text{F}$; $\Delta t = 1 \text{ ms}$; $u_a(t = 0) = 0$

Im Excel-Sheet sollen in der ersten Spalte die Zeit, in Spalte 2 die Eingangsspannung $u_e(t)$ und in der 3. Spalte die Ausgangsspannung $u_a(t)$ stehen. 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 (Spannungssignale in Funktion der Zeit).

Aufgabe 6 (16P)

Ergebnisdatei → *nnnvvv6.vi*

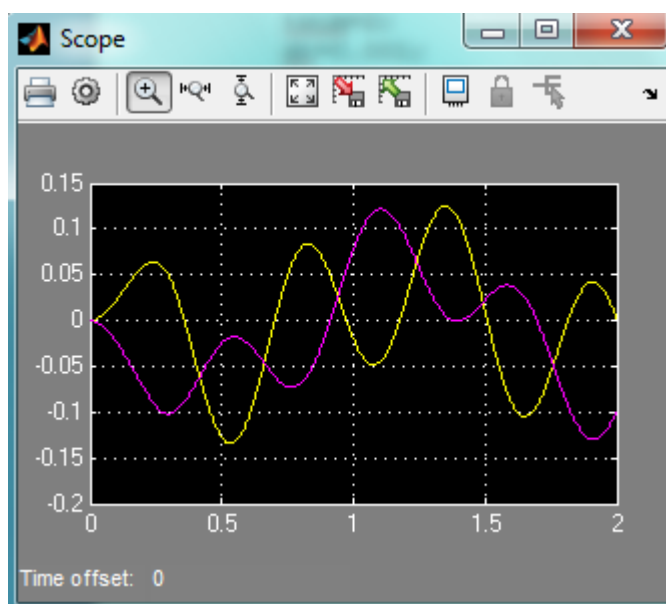
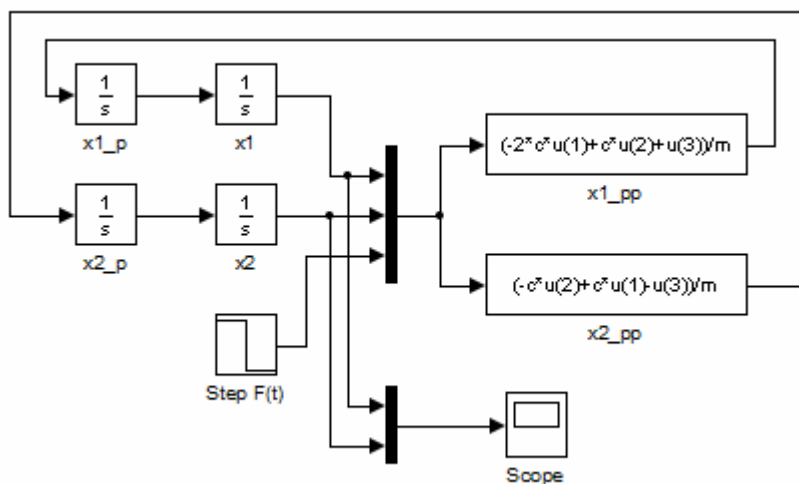
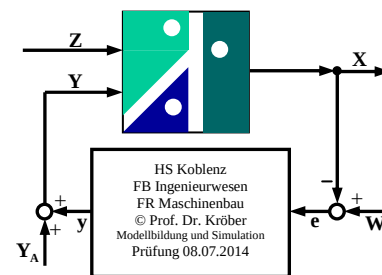
Mit dem USB6008 ist durch das Steckmodul ein Fotowiderstand angeschlossen. Mit dem in der Vorlesung erarbeiteten Programmbeispiel wird die Leuchtstärke [in Lux] in Funktion der Zeit angezeigt.

Das Programm ist dahingehend zu ergänzen: Es sollen zwei zusätzliche numerische Ausgabefenster eingefügt werden. In einem Ausgabefenster wird die maximale Leuchtstärke seit Programmstart angezeigt und in dem zweiten Fenster die minimale Leuchtstärke seit Programmstart.

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

Prüfung 08.07.2014 / Lösung zur Aufgabe 1

```
clear;clc;
m=4;
c=200;
F0=20;
t1=0.25;
tsim=2;
dt=0.001;
sim('krowoll1_sl'); % oder auch direkt
```

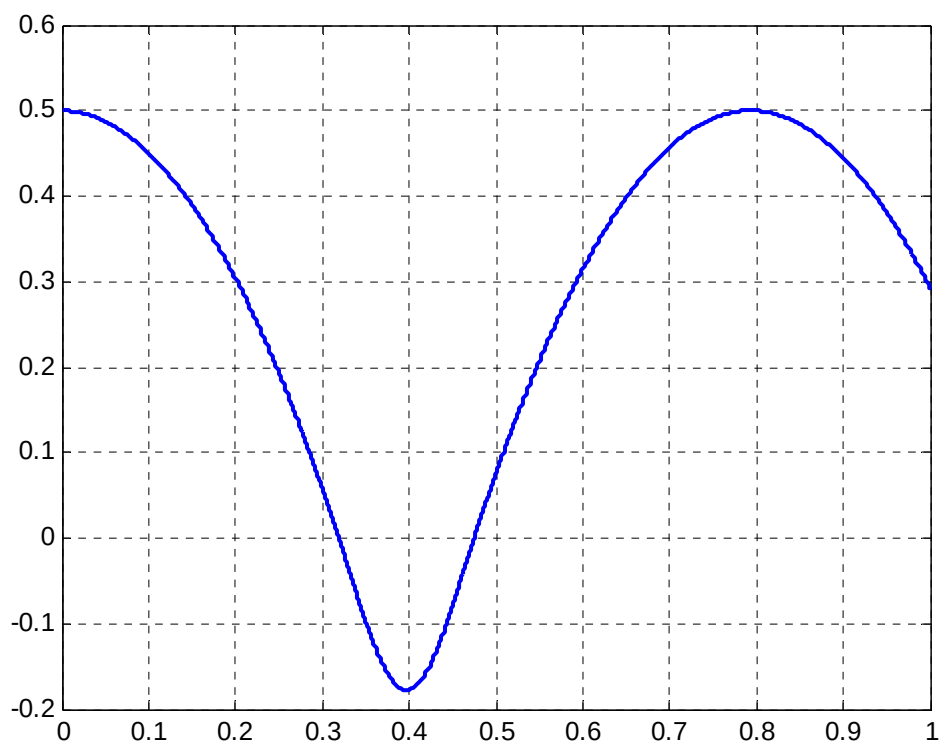
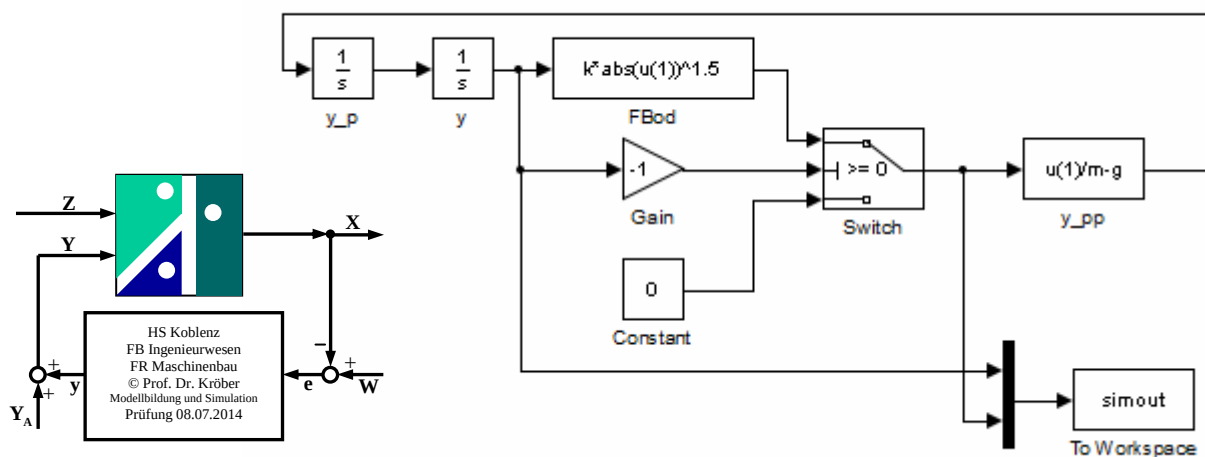


Prüfung 08.07.2014 / Lösung zur Aufgabe 2

```
clear;clc;
y0=0.5;
m=0.8;
g=9.81;
k=1000;
tsim=1;
dt=0.0001;
sim('krowol2_sl');
t=simout.time(:,1);
y=simout.signals.values(:,1);
Fbod=simout.signals.values(:,2);
%subplot(2,1,1);
%plot(t,y);grid;
plot(t,y,'LineWidth',2);grid;
%subplot(2,1,2);
%plot(t,Fbod);grid;

nw=0;
for kk=1:10001
    if Fbod(kk)>0
        nw=nw+1;
    end
end
```

Workspace	
Name ▲	Value
Fbod	10001x1 double
dt	1.0000e-04
g	9.8100
k	1000
kk	10001
m	0.8000
nw	1558
simout	1x1 struct
t	10001x1 double
tout	1000x1 double
tsim	1
y	10001x1 double
y0	0.5000

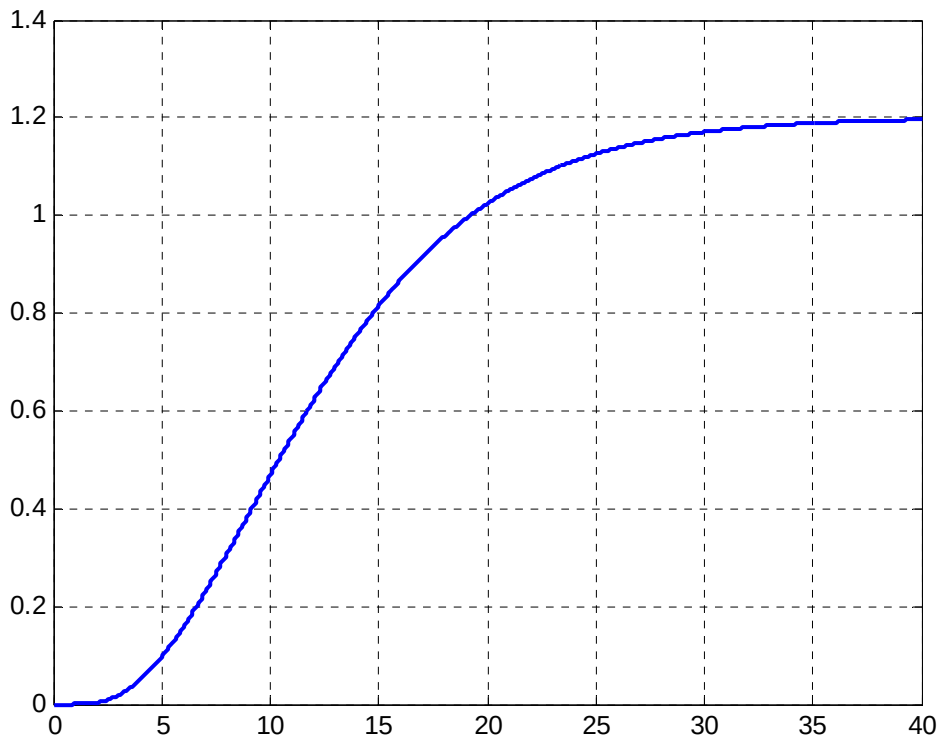
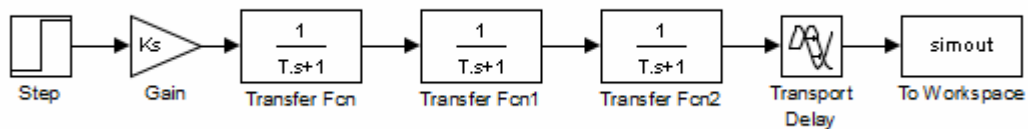
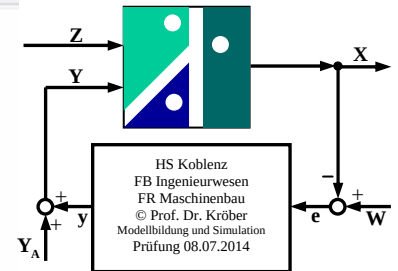


Prüfung 08.07.2014 / Lösung zur Aufgabe 3

```
clear;clc;
Ks=1.2;
T=4;
Tt=1;
tsim=40;
dt=0.01;
sim('krowol3_sl');
t=simout.time(:,1);
X=simout.signals.values(:,1);
plot(t,X,'LineWidth',2);grid;

steigmax=0;
kk_twp=0;
for kk=2:4000
    steig=(X(kk+1)-X(kk-1))/(2*dt);
    if steig>steigmax
        steigmax=steig;
        kk_twp=kk;
    end
end
twp=kk_twp*dt;
Xwp=X(kk_twp);
```

Name	Value
Ks	1.2000
T	4
Tt	1
X	4001x1 double
Xwp	0.3880
dt	0.0100
kk	4000
kk_twp	901
simout	1x1 struct
steig	8.3290e-04
steigmax	0.0812
t	4001x1 double
tout	1000x1 double
tsim	40
twp	9.0100



```

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

  Dim ue(1000), ua(1000) As Double

  uemax = 2
  R = 10000
  C = 0.00001
  dt = 0.001
  Pi = 4 * Atn(1)

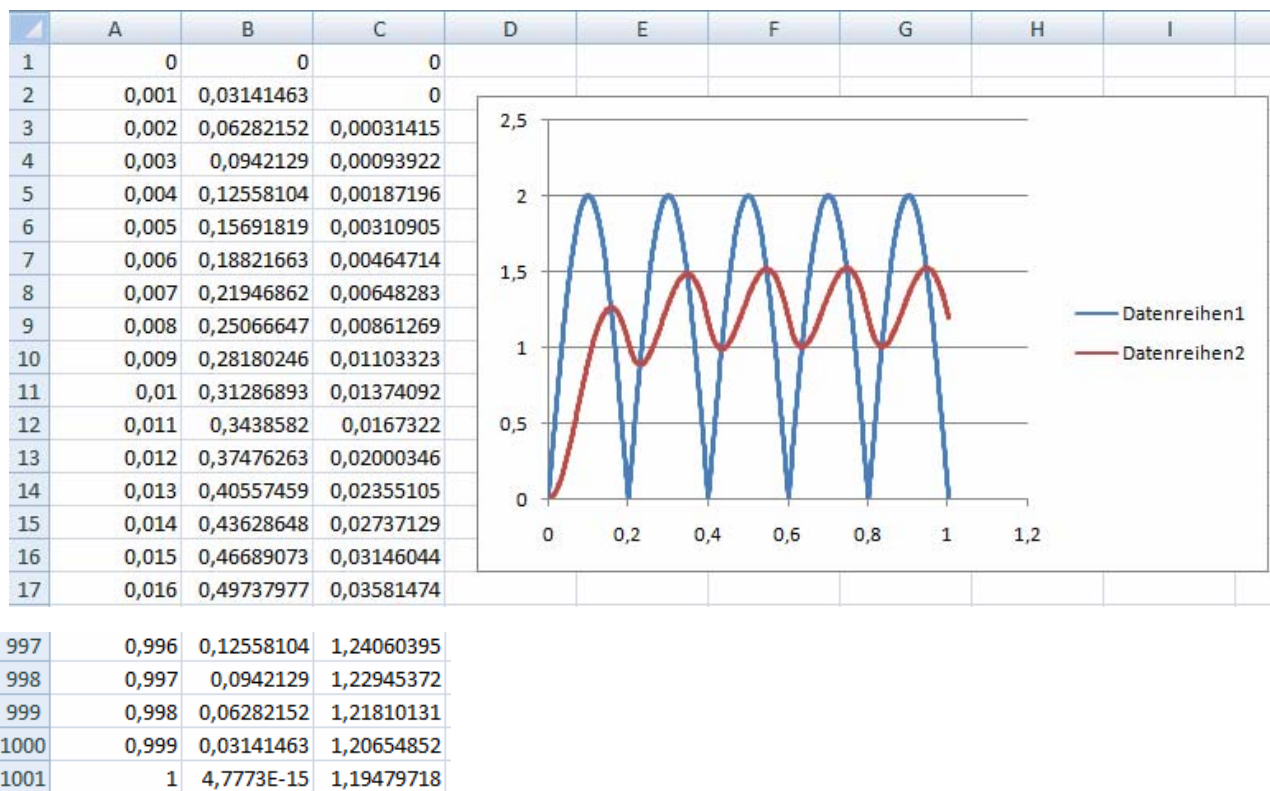
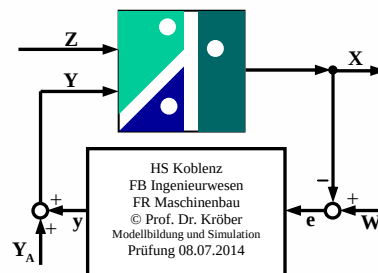
  For i = 0 To 1000
    ue(i) = uemax * Abs(Sin(i * 2 * Pi * 2.5 / 1000))
  Next i

  ua(0) = 0
  For i = 0 To 999
    ua(i + 1) = (1 - dt / (R * C)) * ua(i) + dt / (R * C) * ue(i)
  Next i

  For i = 0 To 1000
    Cells(i + 1, 1) = i * dt
    Cells(i + 1, 2) = ue(i)
    Cells(i + 1, 3) = ua(i)
  Next i

End Sub

```



```

float t[1001],ue[1001],ua[1001];
float uemax=2;
float R=10000;
float C=0.00001;
float dt=0.001;
float mypi=4*atan(1.);
int i;

for(i=0;i<1001;i++)
{
t[i]=i*dt;
ue[i] = uemax * abs(sin(i * 2 * mypi * 2.5 / 1000));
}

ua[0]=0;
for(i=0;i<1000;i++)
{
ua[i+1] = (1-dt/(R*C))*ua[i]+dt/(R*C)*ue[i];
}
    
```

