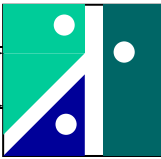


Modellbildung und Simulation WS 11/12
 Prof. Dr. W. Kröber

Zur Bewertung der Aufgaben muss der gesamte Lösungsweg ersichtlich sein.

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

Note : _____

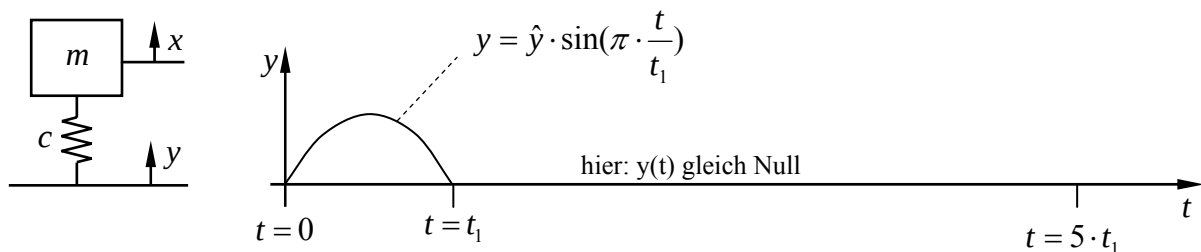
Aufgabe	erreichte Punkte
1	
2	
3	
4	
5	
6	FH Koblenz FB Ingenieurwesen Maschinenbau © Prof. Dr. Kröber Modellbildung und Simulation Prüfung 23.01.2012
Summe	

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.mdl*

Das abgebildete Schwingungssystem beschreibt ein Fahrzeug, welches über eine "einmalige" Bodenwelle fährt. Dämpfungswirkungen werden vernachlässigt. Die Erdbeschleunigung wird nicht berücksichtigt.

$$m \cdot \ddot{x} = -c \cdot (x - y)$$



Berechnen Sie den Verlauf des Schwingweges der Obermasse (Koordinate x) für den Zeitraum $0 \leq t \leq 5 \cdot t_1$! Geben Sie zunächst die Parameter in MATLAB vor. Die weitere Aufgabe kann komplett in Simulink gelöst werden. In dem Scope-Block sollen $y(t)$ sowie $x(t)$ graphisch dargestellt werden. Verwenden Sie als festen Zeitschritt $\Delta t = 1$ Millisekunde. Das vorgeschlagene Lösungsverfahren zur Lösung der Differentialgleichungen (ode3 Bogacki-Shampine) soll so belassen werden.

Zahlenwerte:

$$m = 1000 \text{ kg}; c = 250000 \text{ N/m}; \hat{y} = 0,04 \text{ m}; t_1 = 0,2 \text{ s}; \Delta t = 0,001 \text{ s}$$

Aufgabe 2 (16P) Ergebnisdateien → *nnnvvv2.m* und *nnnvvv2_sl.mdl*

Ein Fahrzeug mit Tempomat (Geschwindigkeitsregelung) fährt zunächst auf einer horizontalen Straße mit der Geschwindigkeit $v_{Soll} = v_0$. Als Fahrwiderstand wird nur der Luftwiderstand berücksichtigt. Dieser Luftwiderstandskraft steigt quadratisch mit der Fahrgeschwindigkeit. Zur Aufrechterhaltung der Geschwindigkeit wirkt an der angetriebenen Achse eine Umfangskraft F_{Umfang} . Der geschilderte Sachverhalt lässt sich durch folgende Gleichung beschreiben:

$$m \cdot \frac{dv}{dt} = F_{Umfang} - m \cdot g \cdot \sin(\alpha) - k \cdot v^2$$

Für den stationären Zustand unmittelbar vor dem Befahren der Steigung gilt:

$$F_{Umfang\ 0} = k \cdot v_0^2$$

Diese Angabe wird als Anfangsbedingung für den später eingesetzten Regler benötigt. Zum Zeitpunkt $t = 0$ befährt das Fahrzeug dann eine Steigung mit dem Winkel α . Durch die Hangabtriebskraft wird die Geschwindigkeit zunächst abnehmen. Zur Aufrechterhaltung der Geschwindigkeit wird hier der Einfachheit halber ein I-Regler eingesetzt, dessen Ausgangsgröße die Umfangskraft variiert. Dies wird durch folgende Gleichung beschrieben:

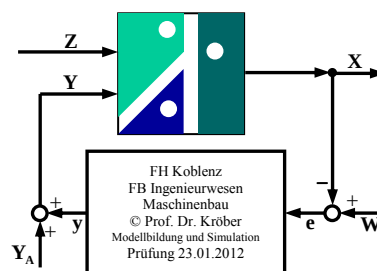
$$F_{Umfang} = K_I \int (v_{Soll} - v) dt$$

Berechnen Sie den Verlauf der Geschwindigkeit des Fahrzeuges für eine Zeit von 300 Sekunden! Geben Sie zunächst die Parameter in MATLAB vor. Die weitere Aufgabe kann in Simulink gelöst werden. Die graphische Darstellung des Geschwindigkeitsverlaufes über der Zeit kann ebenfalls in Simulink erfolgen. Verwenden Sie als festen Zeitschritt $\Delta t = 100$ Millisekunden. Das vorgeschlagene Lösungsverfahren zur Lösung der Differentialgleichungen (ode3 Bogacki-Shampine) soll so belassen werden.

Zahlenwerte:

$$m = 1000 \text{ kg}; g = 9,81 \text{ m/s}^2; \alpha = 3^\circ; v_0 = 25 \text{ m/s}; k = 0,96 \frac{\text{N}}{(\text{m/s})^2}; K_I = 0,8 \frac{\text{N}}{\text{m}}; v_0 = v_{Soll}$$

Anmerkung: Am Ergebnis kann man sehen, dass man in der Praxis doch einen PI-Regler oder PID-Regler einsetzt.



Aufgabe 3 (16P) Ergebnisdateien → *nnnvvv3.m* und *nnnvvv3_sl.mdl*

Bei der für die BUGA 2011 errichteten Seilbahn werden Passagiere zwischen der Talstation und der Bergstation befördert. Untersucht werden soll die Bewegung einer Gondel, wenn die Gondel aus voller Fahrt (der Winkel φ ist zunächst gleich Null) in der Zeit $t_1 = 3$ s mit einer konstanten Verzögerung bis zum Stillstand abgebremst wird. Die Betrachtung erfolgt an einem Punkt über dem Rhein, wo die Seilführung horizontal verläuft.

Aus Wikipedia stammen folgenden Zahlenangaben:

Normalbetrieb $v_0 = 4,5$ m/s

Masse einer Gondel 3,5 Tonnen (Passagiere werden nicht berücksichtigt)

Folgenden Angaben beruhen auf einer Schätzung:

Abstand Schwerpunkt Gondel/Seil $l_s = 5$ m

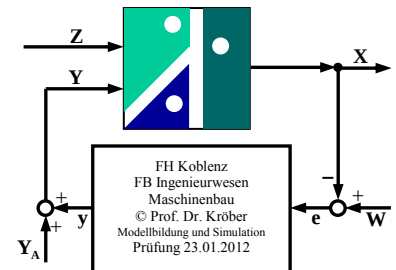
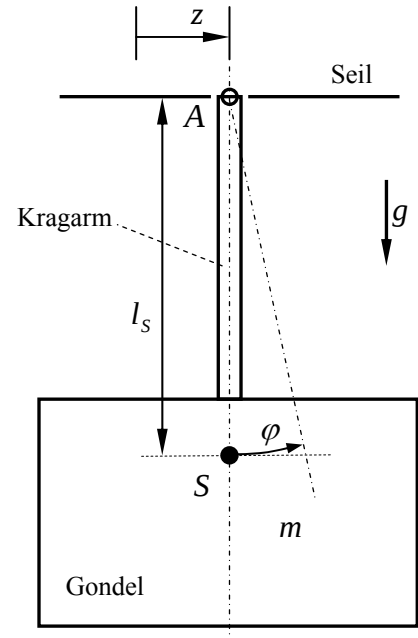
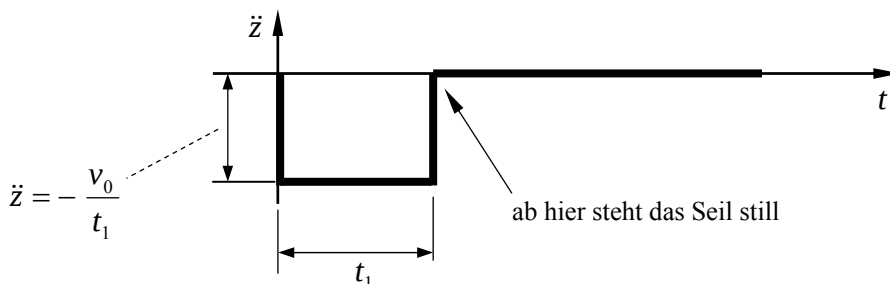
Massenträgheitsmoment $J_A = 97500$ kgm²

Ferner gegeben: $g = 9,81$ m/s²

Aus dem Drallsatz lässt sich folgende Gleichung herleiten:

$$\ddot{\varphi} = - \frac{m \cdot l_s \cdot (\ddot{z} \cdot \cos(\varphi) + g \cdot \sin(\varphi))}{J_A}$$

Vorgegebener Verlauf der Verzögerung:



Geben Sie die Zahlenwerte in Matlab vor und verwenden Sie ein Simulink-Modell zur Lösung. Im Matlab-Programm soll anschließend mit einem plot-Befehl der Verlauf des Winkels φ [im Gradmaß] über der Zeit dargestellt werden. Verwenden Sie als festen Zeitschritt $\Delta t = 10$ Millisekunden. Das vorgeschlagene Lösungsverfahren zur Lösung der Differentialgleichungen (ode3 Bogacki-Shampine) soll so belassen werden. Die Simulationszeit soll 10 Sekunden betragen. Im Workspace soll der maximale Winkelausschlag [im Gradmaß] ersichtlich sein.

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

Ein Fallschirmspringer startet seinen Sprung (ohne Anfangsgeschwindigkeit) in einer Höhe h_0 . Die Luftwiderstandskraft steigt quadratisch mit der Geschwindigkeit. Berechnen Sie den Verlauf der momentanen Flughöhe in Abhängigkeit der Zeit mit einem Excel-Makro. Als Ergebnis soll in dem Excel-Sheet in der ersten Spalte die Zeit und in der zweiten Spalte die momentane Flughöhe stehen. Eine Graphik ist nicht erforderlich. Verwenden Sie als Zeitschritt $\Delta t = 10$ Millisekunden. Die Simulationszeit soll, unabhängig davon, dass sich nach einer gewissen Zeit negative Höhen ergeben, 0s bis 25s betragen. Das Makro soll mit der Tastenkombination „Strg y“ gestartet werden.

Gegebene Zahlenwerte:

$$h_0 = 1000 \text{ m}$$

$$m = 75 \text{ kg}$$

$$g = 9,81 \text{ m/s}^2$$

$$k = 0,2943 \text{ N s}^2/\text{m}^2$$

Aus dem Ansatz $m \cdot \ddot{h} = -m \cdot g + k \cdot \dot{h}^2$

und
$$m \cdot \frac{h_{i+1} + h_{i-1} - 2 \cdot h_i}{\Delta t^2} = -m \cdot g + k \cdot \left(\frac{h_i - h_{i-1}}{\Delta t} \right)^2$$

lässt sich folgende Rekursionsgleichung herleiten:

$$h_{i+1} = 2 \cdot h_i - h_{i-1} + \frac{k}{m} (h_i - h_{i-1})^2 - g \cdot \Delta t^2$$

Tipp: $h(0)=h_0$ und $h(1)=h_1$ setzen, dann mit $h(2)=\dots$ beginnen.

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. Auf dem Frontpanel soll sich nur ein einziges numerisches Anzeigeelement befinden. Im Blockdiagramm befindet sich ein Formelknoten, der einen einzigen Wert ausgibt, der auf dem Frontpanel dann zu sehen ist (keine While-Schleife etc.).

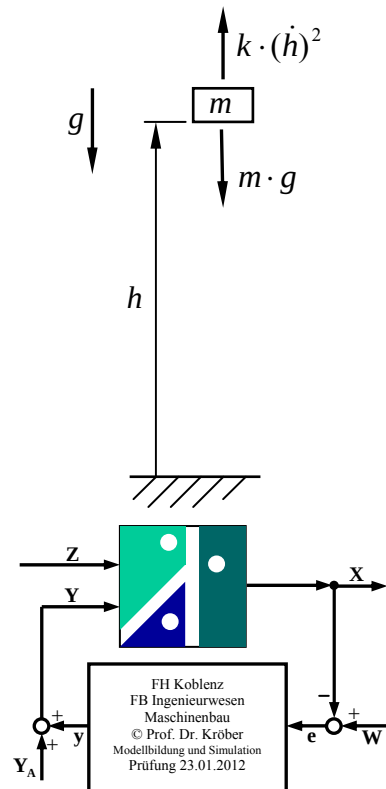
1. Schritt:

Zunächst soll in dem Formelknoten die Höhe h in Abhängigkeit der Zeit berechnet werden. Dabei wird das Array $h[i]$ mit Werten aufgefüllt.

2. Schritt:

Dann soll ermittelt werden, zu welchem Zeitpunkt die Höhe erstmals negativ ist. Dieser Zeitpunkt soll ausgegeben werden. Variiert man dann beispielsweise die Anfangshöhe h , dann soll stets automatisch die entsprechende Zeit ausgegeben werden.

Als Hilfestellung sind die möglichen ersten Zeilen des Programmes im Formelknoten angegeben.



Formelknoten:

```
int i;
float h[2501];
float h0=1000;
float m=75;
float g=9.81;
float k=0.2943;
float dt=0.01;
h[0]=h0;
h[1]=h0;
for(i=1;i<2500;i++)
{
    ...
    h[i+1]= ...
}
...
```

Aufgabe 6 (20P) Ergebnisdatei → *nnnvvv6.vi*

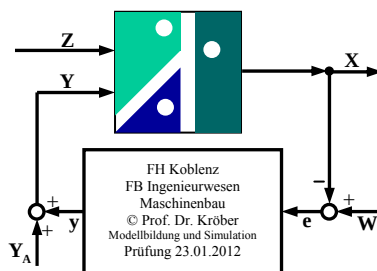
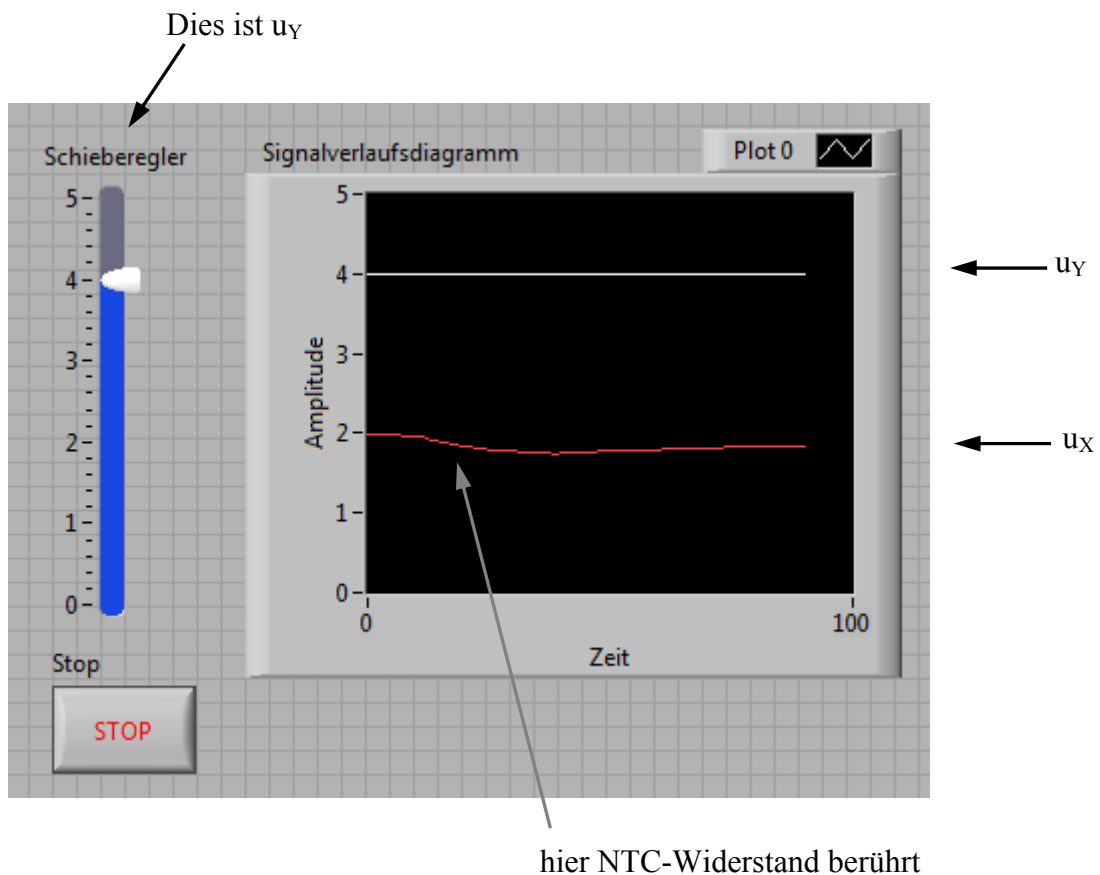
Mit dem USB6008 ist durch das Steckmodul ein Spannungsteiler mit einem NTC-Widerstand angeschlossen. Folgendes Problem ist zu lösen: Im Blockdiagramm wird in einer While-Schleife (versehen mit 100ms Timer) ein Slider (Schieber mit Zeiger) auf dem Frontpanel abgefragt. Auf dem Slider sollen Werte zwischen 0V und 5 V einstellbar sein. Der Wert wird im Folgenden mit u_Y bezeichnet. Der Wert u_Y wird auf dem Ausgang „AnalogOutput1“ ausgegeben.

Ferner wird in der While-Schleife der Spannungswert von „AnalogInput2“ eingelesen. Dieser Wert wird im Folgenden mit u_X bezeichnet. Die Werte u_Y und u_X sollen in einem gemeinsamen Signalverlaufdiagramm auf dem Frontpanel ausgegeben werden.

Hilfestellungen und Motivation:

u_X wird etwa halb so groß wie u_Y sein. Wird der NTC-Widerstand mit dem Finger berührt, wird der Wert für u_X etwas kleiner (infolge der Widerstandsänderung des NTC's).

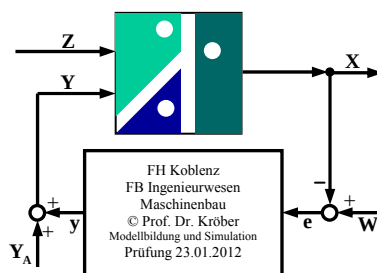
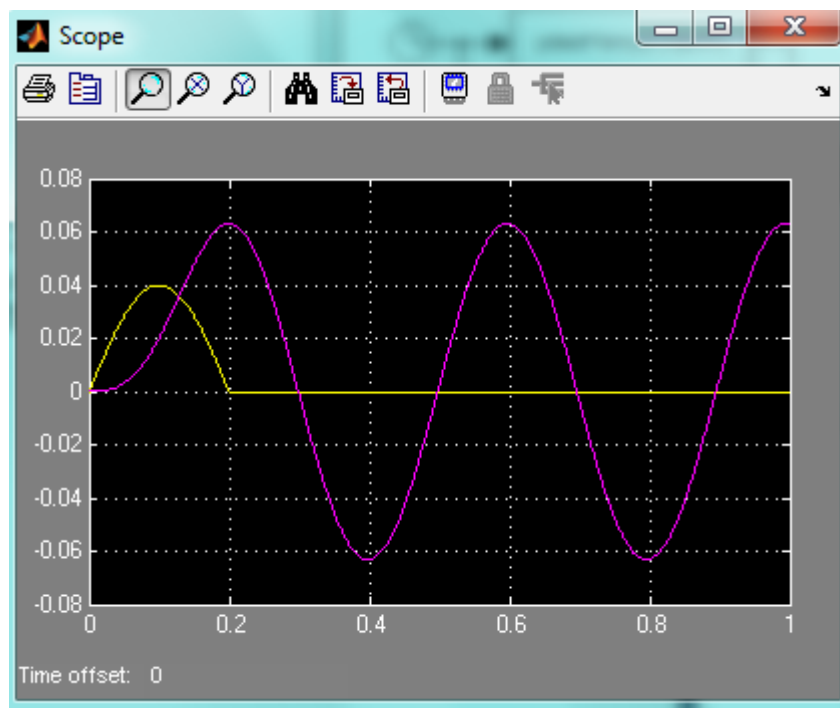
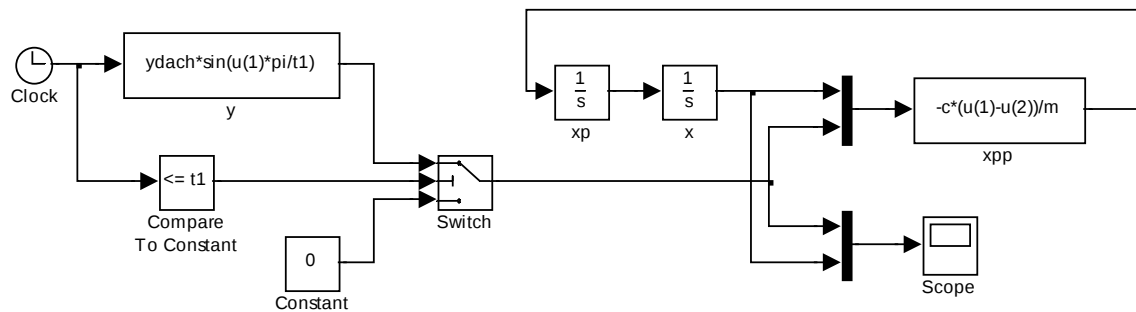
Dies ist eine Abbildung des gewünschten Frontpanels:



Prüfung 23.01.2012 / Lösung zur Aufgabe 1

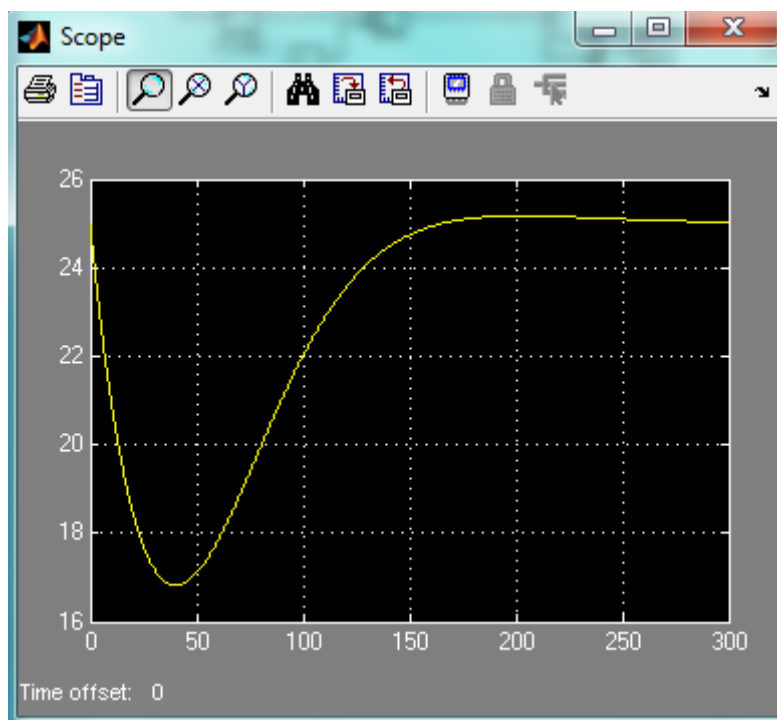
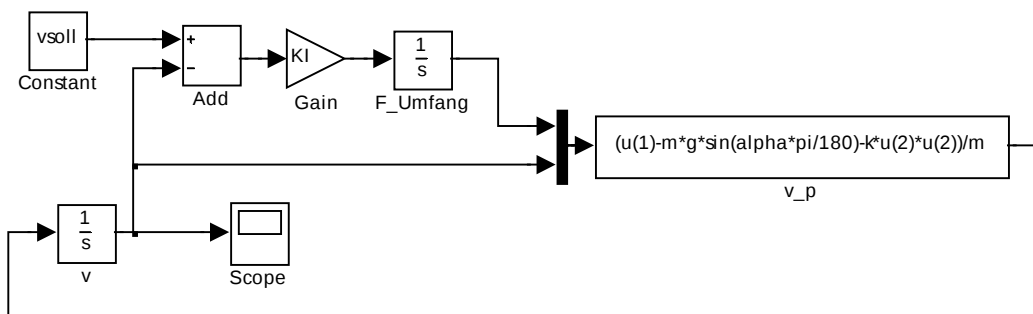
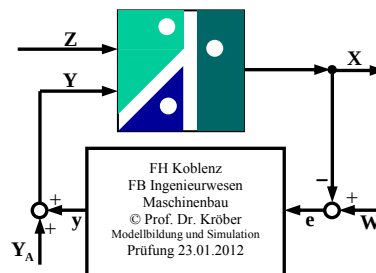
```

clear;clc;
m=1000;
c=250000;
ydach=0.04;
t1=0.2;
dt=0.001;
    
```

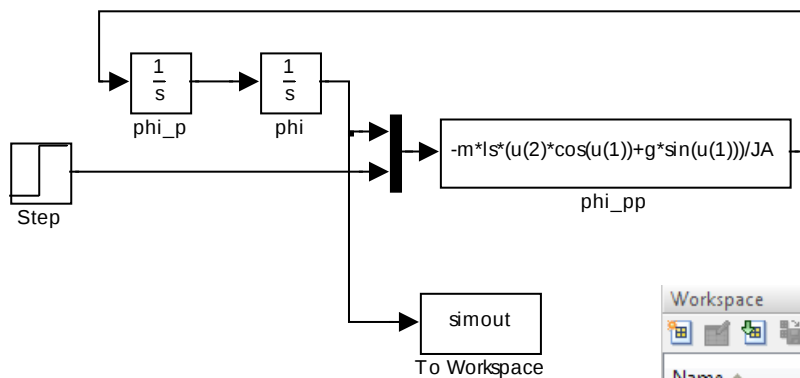
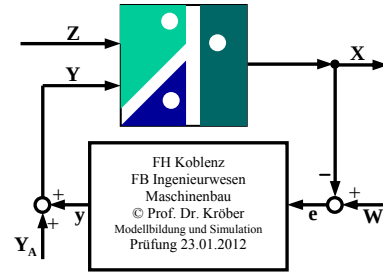


Prüfung 23.01.2012 / Lösung zur Aufgabe 2

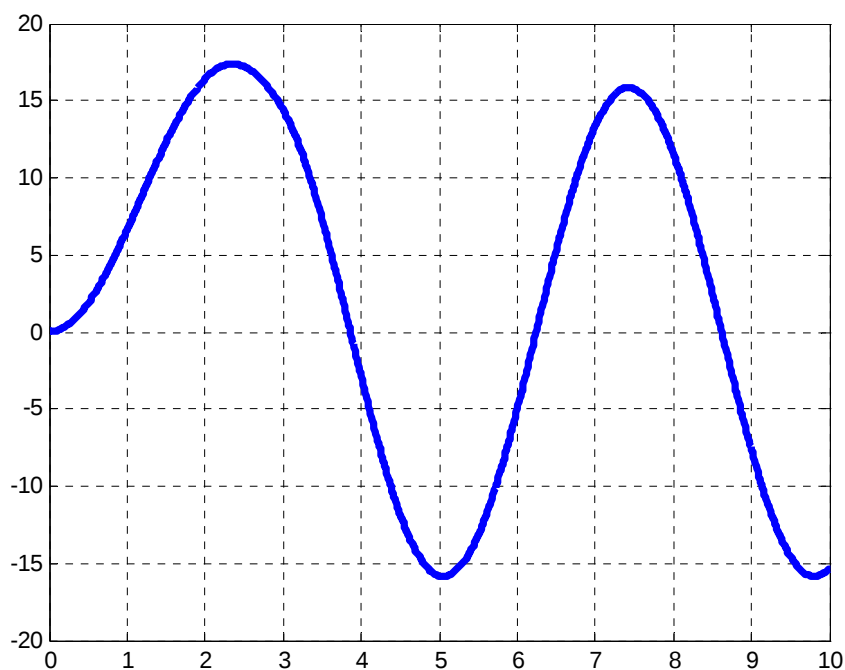
```
clear;clc;
m=1000;
g=9.81;
alpha=3;
v0=25;
vsoll=v0;
k=0.96;
KI=0.8;
tsim=300;
dt=0.1;
```



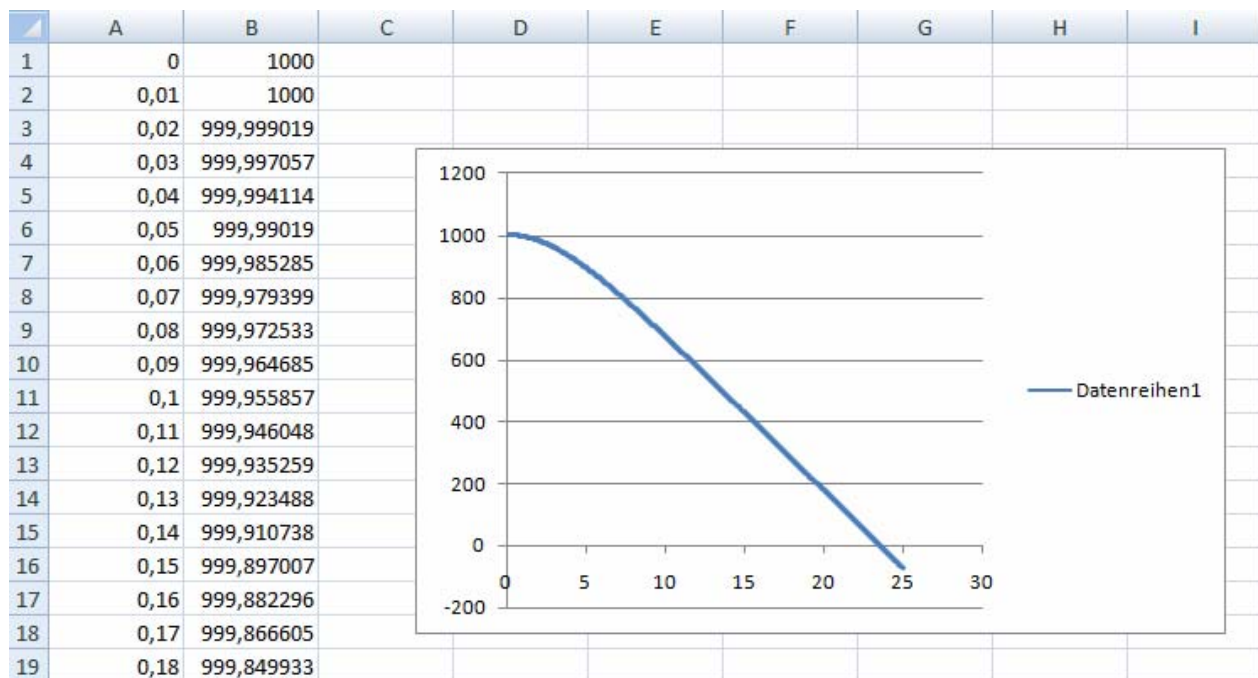
```
clear;clc;
t1=3;
v0=4.5;
m=3500;
ls=5;
JA=97500;
g=9.81;
dt=0.01;
tsim=10;
sim('krowol3_sl');
plot(simout.time,simout.signals.values*180/pi);grid;
max_winkel=max(simout.signals.values*180/pi);
%T0=2*pi*sqrt(JA/(m*g*ls)); %Schwingperiodendauer
```



Workspace			
Name	Value	Min	Max
JA	97500	97500	97500
dt	0.0100	0.0100	0.0100
g	9.8100	9.8100	9.8100
ls	5	5	5
m	3500	3500	3500
max_winkel	17.3869	17.3869	17.3869
simout	<1x1 struct>		
t1	3	3	3
tout	<1000x1 double>	0.0100	10
tsim	10	10	10
v0	4.5000	4.5000	4.5000



Prüfung 23.01.2012 / Lösung zur Aufgabe 4
(Bem.: Graphik nur zur eigenen Kontrolle)



	A	B	C	D	E	F	G	H	I
2494	24,93	-69,7744582							
2495	24,94	-70,2744027							
2496	24,95	-70,7743474							
2497	24,96	-71,2742923							
2498	24,97	-71,7742375							
2499	24,98	-72,2741828							
2500	24,99	-72,7741284							
2501	25	-73,2740742							
2502									

```

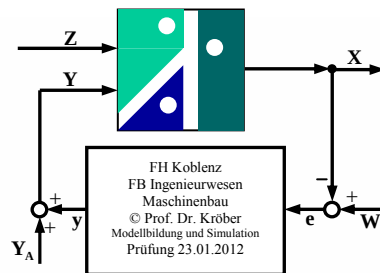
Sub Makrol1()
'
' Makrol Makro
'
' Tastenkombination: Strg+y
'

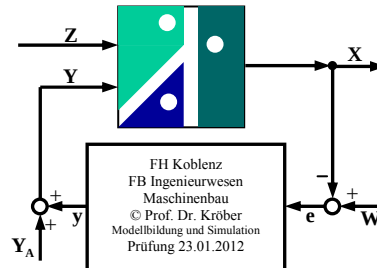
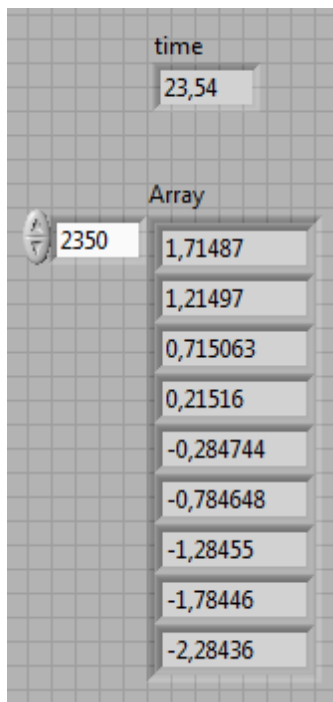
Dim h(2500) As Double
h0 = 1000
m = 75
g = 9.81
k = 0.2943
dt = 0.01
h(0) = h0
h(1) = h0
For i = 1 To 2499
    h(i + 1) = 2 * h(i) - h(i - 1) + k / m * (h(i) - h(i - 1)) * (h(i) - h(i - 1)) - g * dt * dt
    ' alternativ: h(i + 1) = 2 * h(i) - h(i - 1) + k / m * (h(i) - h(i - 1))^2 - g * dt^2
Next i

For i = 0 To 2500
    Cells(i + 1, 1) = i * dt
    Cells(i + 1, 2) = h(i)
Next i

End Sub

```





```
int i;
float h[2501];
float h0=1000;
float m=75;
float g=9.81;
float k=0.2943;
float dt=0.01;
h[0]=h0;
h[1]=h0;

for(i=1;i<2500;i++)
{
    h[i+1]=2*h[i]-h[i-1]+k*(h[i]-h[i-1])*(h[i]-h[i-1])/m-g*dt*dt;
    // alternativ: h[i+1]=2*h[i]-h[i-1]+k*(h[i]-h[i-1])**2/m-g*dt**2;
}

int ineg=0;
for(i=0;i<2501;i++)
{
    if((ineg==0)&&(h[i]<0))
    {
        ineg=i;
    }
}

float time;
time=ineg*dt;
```

