Technik
Informatik & Medien

Hochschule Ulm

University of
Applied Sciences

Computerlab 1

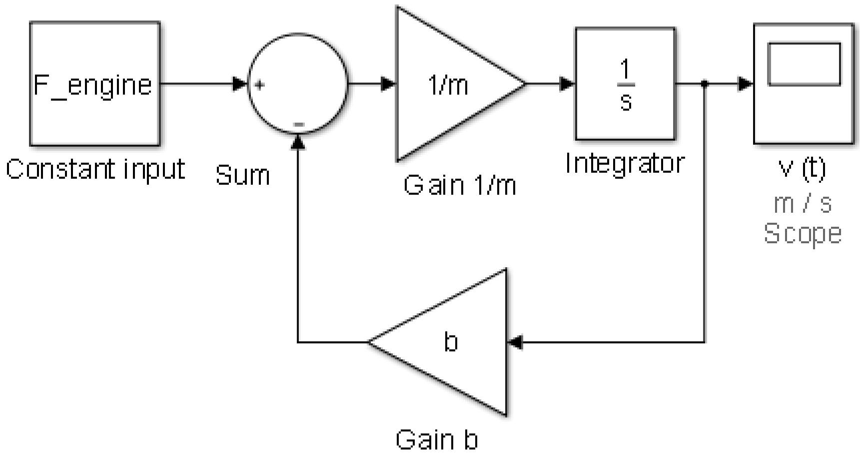
Simulink Model of the Car for Cruise Control

Prof. Dr. A. Beckmann

Computerlab 1:
Simulink Model of the Car for Cruise Control

1 Simulink Model of the Car with Linear Friction Force

In the first model of the car we assume a linear friction force. The final block diagram of the Simulink model of the car, we want to create now, will look like this:



We want to use this model to derive the response of the system from simulations:

a) Response to a constant input (= step input)
b) Response to initial conditions without an input

1.1 Steps to Build the Simulink Model of the Car

Step A: We rewrite the differential equation of the system:

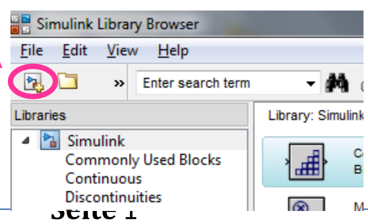
$$m\frac{dv}{dt} = F_{engine} - bv \quad \Rightarrow \quad \frac{dv}{dt} = \left(F_{engine} - bv\right)\frac{1}{m}$$

$$\Rightarrow \quad \boxed{v = \int \frac{dv}{dt}dt = \int \left(F_{engine} - bv\right)\frac{1}{m}dt} \qquad \text{(Eq. 1)}$$

Step B: Install MatLab
See Appendix A of this paper to install MatLab on your computer in the computerpool of the university.

Step C: To **start Simulink** enter in the Matlab command window: >> **simulink**
The Simulink Library Browser will open.

Step D: Click twice on the **icon in the left upper corner** to open a sheet for creating the new model



Step E: Design of the force of the motor (first integral of Eq. 1):

1. Click twice **Simulink** in the left list of the library browser.

2. select **Sources**
   select **Constant** for a constant value of F_{engine}
   and drag and drop it to the work sheet
   Click twice on this icon and **enter F_engine** as constant value;
   enter 0.05 as sample time; **OK**
3. **Math Operations**
   select **Sum** and drag and drop it to the work sheet
   Click twice on the icon and **insert '+-'; OK**

4. **Math Operations**
   select **Gain** and drag and drop it to the work sheet
   Click twice on the icon and **insert 1/m** as gain; **OK**

5. **Continuous**
   select **Integrator** and drag and drop it to the work sheet
   Click twice on the icon and **insert v0** as initial condition; **OK**

6. **Sinks**
   select **Scope** that will show v(t) and drag and drop it to the work sheet
   Click twice on the icon and **rename** it by clicking on scope: **v(t)**

7. **Connect all elements** by selecting icon1 (click once on this first icon),
   while pressing Crtl (or Strg) click on the second icon
   and so on.

Step F: Design of the friction force (2. integral in Eq. 1):

1. **Math Operations**
   select **Gain** and drag and drop it to the work sheet
   click twice on the icon and **insert b** as gain; **OK**

2. To flip the icon we click once on the gain-icon using the right mouse key
   select **Rotate & Flip** in the drop-down-menue
   select **Flip block**

3. Connect the output of the integrator to the input of the new gain-icon.

4. Connect the output of the gain-icon to the minus-input of the sum-icon.

Step G: Save the model using the name **car_linear.slx**
(.slx is added automatically)

Now the simulink model is ready for simulations. So go on reading.

1.2 Simulations and Discussion

We want to study

a) the response to a constant input (step input) and

b) the initial response

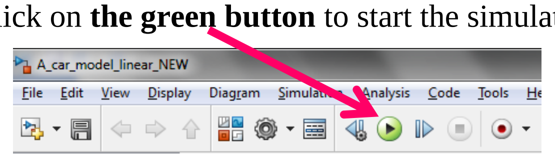
1.2.1 Step Response of the System

1. Define the parameters m, b, F_Engine and v0 in the **matlab command window**:
   >> **m = 1000; b = 50; F_engine = 500; v0 = 0; t_stop = 100;**

2. Press enter and return to the Simulink model.

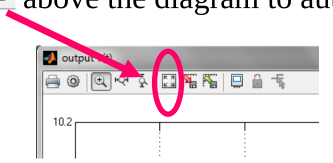
3. Select **Simulation** in the drop-down-menue
   Click on **Model Configuration Parameters**
   Write into the field **Stop time: t_stop**
   OK

4. Click on **the green button** to start the simulation.

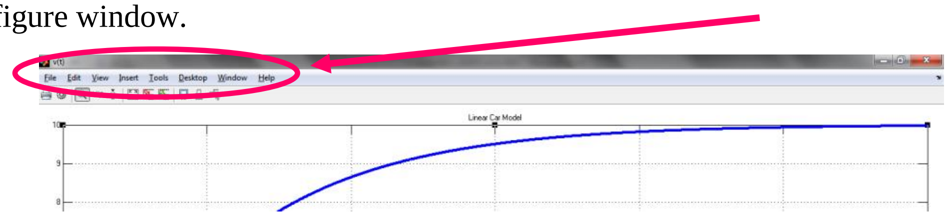


5. After the simulation has stopped, click twice on the scope-icon to open the scope-window to show the diagram.

6. Click on  above the diagram to autoscale the plot.



7. Run the program **scope.m** (see Appendix B) to show the hidden handles of your figure window.

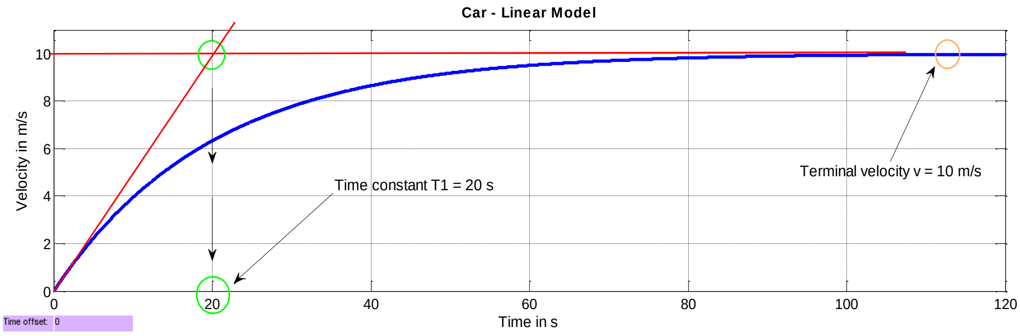


8. Add a title, a label for x-axis and y-axis to the plot.
   Select **Insert**
   Click on **X_Label**
   Now you can write the label for the x-axis **Time in s** and so on.

9. Change the colors, linewidth and size of text so that everything can be read well.
   Select **Edit**
   Click on **Figure Properties…**
   The diagram will be shown in a new window. Click on the item (e.g. the background) in the diagram you want to change.

10. Do not forget to save the model after you have performed all the modifications in your diagram.

11. Draw a line with the slope of the plot at the origin to read out the time constant.
   Select **Insert**
   Click on **Line**; then you can draw the line in the figure window.



For documentation we copy the simulink model and the graph $v(t)$ into a WORD-document and print them out. (See Appendix B for the steps how to do it.)

Derive the equation for the time constant of the system and the equation for the terminal velocity from the differential equation:

$$m\frac{dv}{dt}+bv=F_{engine} \quad \Rightarrow \quad \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_$$

____

____

$\Rightarrow \quad T_1 =$ __________________ $\quad v_t =$ ________________

Determine the values for the time constant and the terminal velocity of the system from the diagram of the simulation and compare the values to the theoretical results you get if you use F_{engine} = 500 N; m = 1000 kg and b = 50 Ns/m.

Theory:	$T_1 =$ ________________	$v_t =$ ________________
Simulation:	$T_1 =$ ________________	$v_t =$ ________________

1.2.2 Initial Response of the System

Start a new simulation to obtain the initial response of the system:

1. Define the parameters m, b, F_Engine and v0 in the matlab command window:
   >> **m = 1000; b = 50; F_engine = 0; v0 = 10; t_stop = 100;**

2. Press enter and return to the Simulink model.

3. Select **Simulation** in the drop-down-menue
   Click on **Model Configuration parameters**
   For **Stop time** we use the variable **t_stop**
   OK

4. Click on the **green button** to start the simulation.

5. After the simulation has stopped, click twice on the scope-icon of the model to open the scope-window and show the diagram.

6. Click on  to autoscale the diagram.

7. Modify the diagram using the hidden handles according to the way you did it for the step response. In case the handles are hidden again, run scope.m to show them.

Print the initial response of the system for documentation.

Derive the time constant of the initial response from the diagram and compare the value of the time constant with the one you derived from the step response.

Initial Response: T_1 = ______________________

Step Response : T_1 = ______________________

Make sure, you copy your program and model to an USB-stick, because the work disc of the university will be cleaned every night.

2 Simulink Car Model with Nonlinear Friction Force

2.1 Design of the model

We want to modify the model of the car to a model with nonlinear friction and perform simulations to get

a) the response to constant input and
b) the initial response

The first step is to save the old linear model using the new name **car_nonlinear.slx** (.slx is added automatically) and modify the friction force in the block diagram according to the following steps.

The differential equation of the nonlinear car model is:

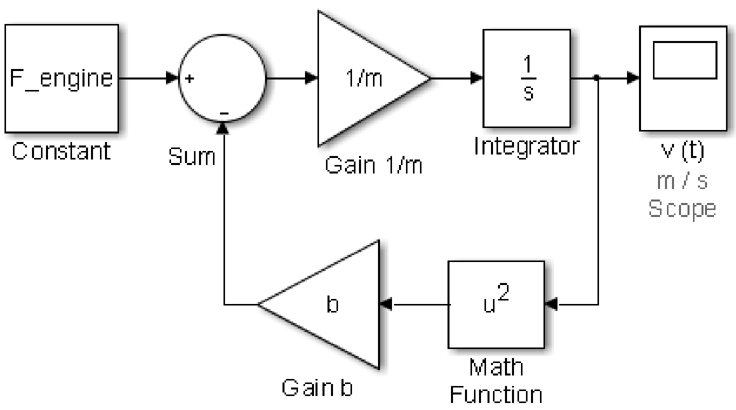
$$m\frac{dv}{dt} = F_{engine} - b_{nl}v^2$$

b_{nl} = coefficient of <u>nonlinear</u> friction

For modelling the nonlinear friction force we use the function element of the simulink library:

Math Operations
- select **Math Function**
  - click twice on the icon
    - select **square** in the drop-down menu of **function**; **OK**
  - click once on the icon using the right mouse key
    - select **Rotate & Flip**
      - select **Flip Block** to turn the icon by 180°

After connecting the icons the nonlinear model of the car will look like this:



Copy the new block diagram into your WORD-document for documentation.

2.2 Simulations and Discussion

Declare the parameters of the car in the MatLab-command window:

a) for step response use: m = 1000; b = 0.96; F_engine = 500; v0 = 0; t_stop = 100;
b) for initial response use: m = 1000; b = 0.96; F_engine = 0; v0 = 10; t_stop = 100;

Perform the simulations on step response and initial response and document your results in your Word-document.

Compare your results to those of the linear model:

a) Step response: ____

____

____

b) Initial response: ____

____

- Calculate the theoretical value of the terminal velocity for step response of the nonlinear model and compare to the result from simulation:

$$m\frac{dv}{dt} + b_{nl} * v^2 = F_{engine} \quad \Rightarrow$$

____

____

____

From theory: $v_t =$ ____________________

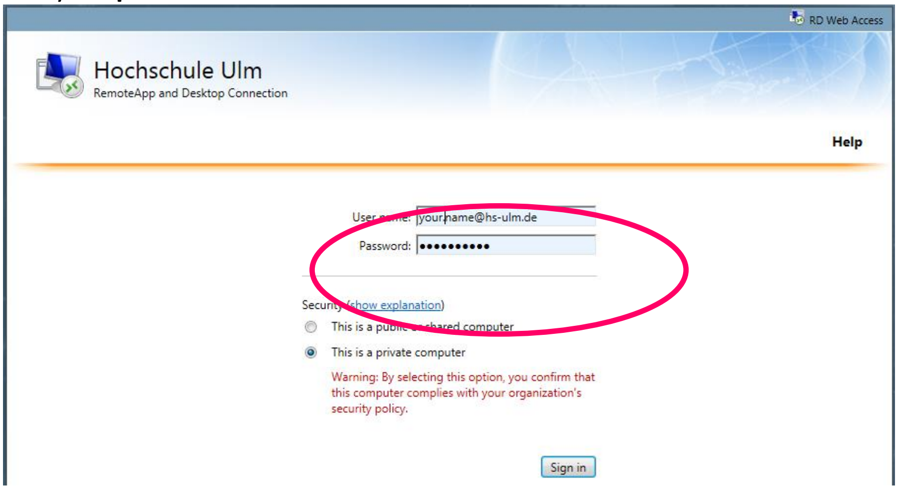
From simulations: $v_t =$ ____________________

3 Appendix A: Steps to Install MatLab on a Computer in the Computerpool of the University of Ulm

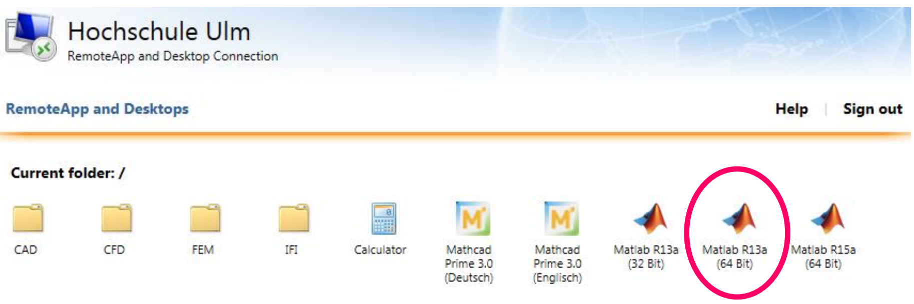
(no access from home without VPN)

1. Use a computer in the computerpool on the campus Hochschule Ulm.

2. Login, open your internet browser and enter
   https://remoteapp.hs-ulm.de/
3. The LOGIN-Screen will open and you have to enter the **email-adress of the University of Ulm** and your **password:**



4. Now the folder for selection of the program will open:



5. Click twice on the icon **MatLab 2013a 64-Bit** to run the software (it will take a while). Sometimes you are asked once more for “Benutzername” which here is your **email-adress of the university of Ulm** and your **password** before you can use MatLab.

6. After closing down, you have to use the remoteapp once again to use MatLab.

Step-by-step tutorials for Matlab and Simulink will help you to learn how to use the software tool:

http://www.mathworks.de/academia/student_center/tutorials/index.html

4 Appendix B: Steps to Transfer the Results into Word

4.1 Steps to Transfer a Simulink Block Diagram into Word

1. Click on **Edit** the menu bar
2. Choose **Copy Current View to clipboard** and select **Bitmap**
3. Change to the WORD-document
4. Choose **Paste** in the edit-menu to insert the block diagram into WORD

4.2 Show hidden Handles in Simulink Scope Diagrams

1. Write the following short Matlab-program **scope.m** in the MatLab-editor:

```
% Program to show the hidden handles
% in the simulink scope window
shh = get(0,'ShowHiddenHandles');
set(0,'ShowHiddenHandles','On')
set(gcf,'menubar','figure')
set(gcf, 'PaperPositionMode', 'auto')
set(gcf,'CloseRequestFcn','closereq')
set(gcf,'DefaultLineClipping','Off')
set(0,'ShowHiddenHandles',shh)
```

2. Run this program from the MatLab-command line (just type in **scope**). Then the toolbar with the hidden handles will be shown in the Simulink scope window.

4.3 Steps to Transfer Simulink Scope Diagrams to Word

a) Transfer from Simulink scope window after showing the hidden toolbar:

1. Run the simulation to get the plot in the Simulink scope window.
2. After running the program scope.m you see a toolbar in the Simulink scope window.
3. Select **Edit** in the toolbar
   Copy Figure
4. Open your Word-document and paste the figure from the clipboard.

b) Transfer via pdf-file:

1. Install a pdf-printer. (e.g. from http://en.pdf24.org/download-pdf-creator.html)
2. Use the printer in the Simulink scope window to create a pdf-file of your diagram and save it on your disk: scope1.pdf
3. Open your Word-document and insert the pdf-file as an object:
   Insert
      Object
         Choose **Adobe Acrobat Document** and press **OK**
            Choose the name of the file to insert: scope1.pdf

4.4 Steps to Transfer the MatLab Diagrams to Word

1. **Open the window with the matlab figure you want to transfer.**

2. **Edit the figure according to your needs:**

   - You can insert something by clicking on **Insert** in the toolbar.
   - You can edit the figure by clicking on **Tools**, choose **Edit plot**, then click on the part of the figure you want to change and perform the changes.

3. **Copy the figure to the Word document:**
   Click on **Edit** in the menu bar of the figure window.
   Choose **Copy Figure**
   Go to your Word document and Paste the figure to Word by clicking **Shift+Insert**.

5 Appendix C: Steps to use MatLab for Initial Response

Initial Response of the Car Model for a Cruise Control System

The MatLab statements to find the initial response function from the differential equation using the **Symbolic Math Toolbox**:

```
% solve differential equation without initial condition
dsolve('m*Dv+b*v=0')

answer:         C10*exp(-(b*t)/m)

% solve differential equation with initial condition
dsolve('m*Dv+b*v=0','v(0)=v0')

answer:         v0*exp(-(b*t)/m)
```

The MatLab statements to plot the initial response for v_o = 50 km/h are:

```
% Car model for cruise control
% Deterimation of the time constant from the plot

clear        % clear all variables of the workspace
close all    % close all figures

% parameters of the car
m = 1000;       % Mass in kg
b = 50;         % friction coefficient in N*s/m
%F_Engine = 500;  % Force of the engine (not needed for initial response)

% initial conditions
v0_kmh = 50;         % initial speed in km/h
v0 = v0_kmh/3.6;     % initial speed in m/s

N = 1000 ;       % number of samples
tau = m/b;       % time constant

t_max = 6*tau;
t = 0:t_max/(N-1):t_max;     % time vector (N samples)

% Calculation of v(t) for the N time samples
v = v0*exp(-t/tau);
v_kmh = v*3.6 ;          % velocity in km/h

% Plot of v(t)
plot(t,v_kmh)
grid
xlabel('Time in s')
ylabel('v in km/h')
title('Car Model: Initial Response')

hold on
```

```
% Plot of the line for determination of the time constant from intercept
% linear regression of v(t) near t = 0s
P = polyfit(t(1:5),v(1:5),1);   % y = a x + c ; P(1) = a; P(2) = c
plot(t,(P(1)*t+P(2))*3.6,'r')   % plot of the straight line
axis([0,t_max,0,v0_kmh])
% end of program
```

6 Appendix D: Linearization of the nonlinear model

If we have Newton's drag instead of Stoke's drag we have to linearize the model.

```
% Linearization of the nonlinear car model for cruise control
% Determination of the friction coefficient of the linear model

clear       % clear all variables of the workspace
close all   % close all figures

% parameters of the drag coefficient of nonlinear air drag
calc_drag = 0
if calc_drag ==1,
    disp('drag coefficient calculated')
    rho_air = 1.204;          % Density of air kg/m3
    A = 2                     % crossectional area in m2 (typical values 1,7
... 2,0 m2)
    c_D = 0.6                 % (typical values: 0,3 ... 0,6)
    c_D_A = c_D * A           % drag area (typical value for a car: c_D_A =
0.79)
    b = rho_air * c_D_A /2    % coefficient of Newton's drag
else
    disp('drag coefficient set to value')
    b =0.96
end %if

% velocity of the car
N = 500;                      % number of samples is N+1
v_max_kmh = 250;              % maximum velocity in km/h
v_max = v_max_kmh / 3.6;      % maximum velocity in m/s
v = 0:v_max/N:v_max;          % velocity in m/s (= vector with N+1 samples)

% Calculation of nonlinear air drag (Newton's drag)
% for the N+1 time samples
F_nonlin = b * v.^2;

% Plot of v(t)
plot(v*3.6,F_nonlin)
grid
xlabel('v in km/h')
ylabel('Air drag in N')
hold on

% Plot of the line for determination
% of the coefficient of linear friction
% from linear regression of F_nonlin(v) for small values of v
% y = m x + b ; P(1) = m; P(2) = b
% we force b = 0 by using zeros in the vectors of v and F_nonlin
P = polyfit([zeros(1,10000*N) v],[zeros(1,10000*N) F_nonlin],1);
%P = polyfit(v,F_nonlin,1);      % here for F_linear = 0 we get v<0
plot(v*3.6,(P(1)*v+P(2)),'r')    % plot of the straight line
axis([0,v_max_kmh,0,max(F_nonlin)])
title(['Linearization: friction coefficient b =   ',num2str(P(1)),' Ns/m'])
%end of program
```