Simulationstechnik

Fachhochschule Heilbronn

Automotive System Engineering

Simulationstechnik

L-3 Matlab .m files

Dr. Tröster
ASE 5

The philosophy of Matlab m files from the simulation point of view

What tools do we have?

Matlab functions

Toolbox functions

Special environment
Simulink

Sometimes it is not enough. You need the possibility to write programs which can include the Matlab and Toolbox functions.

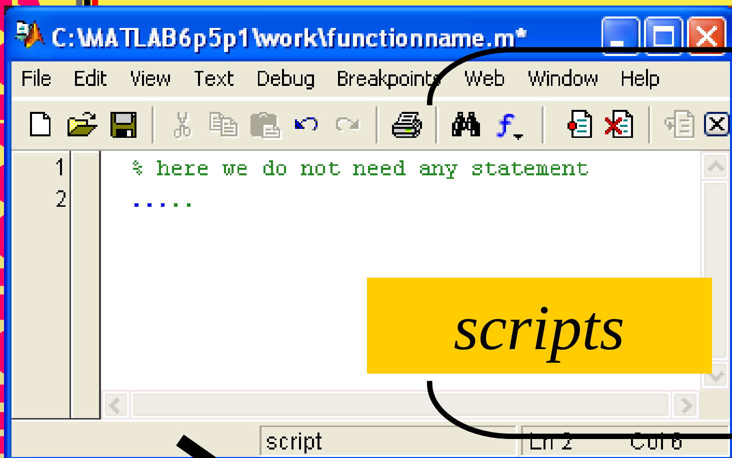
Simulationstechnik

Matlab functions

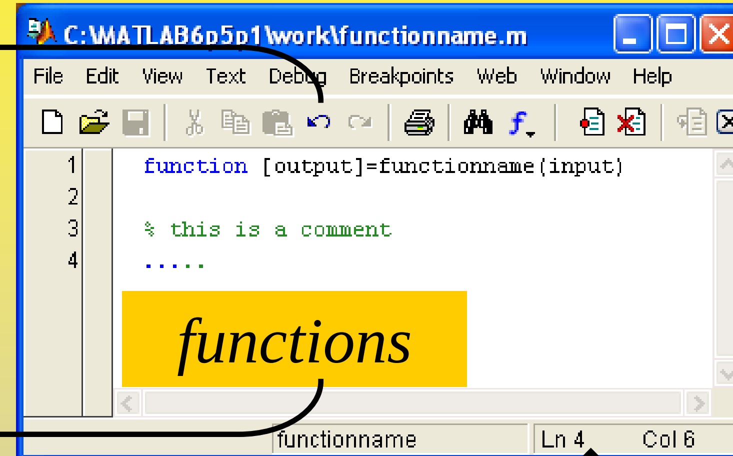
Toolbox functions

Programming

files. m

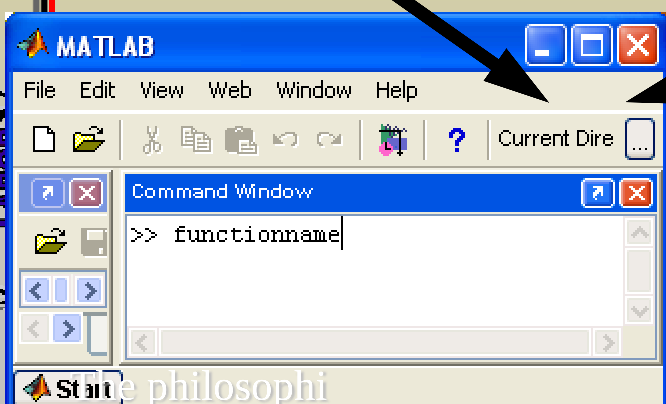


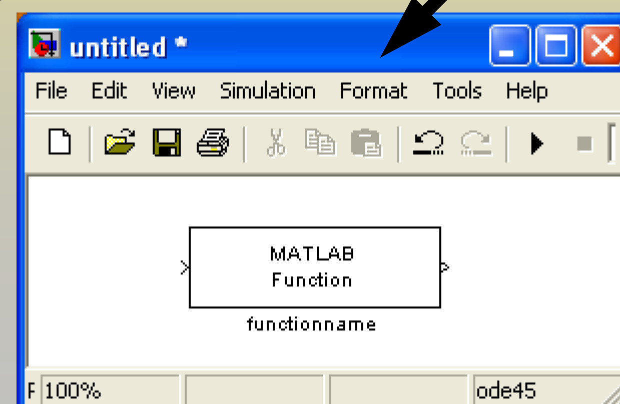
scripts



functions

Where these programs can be use?



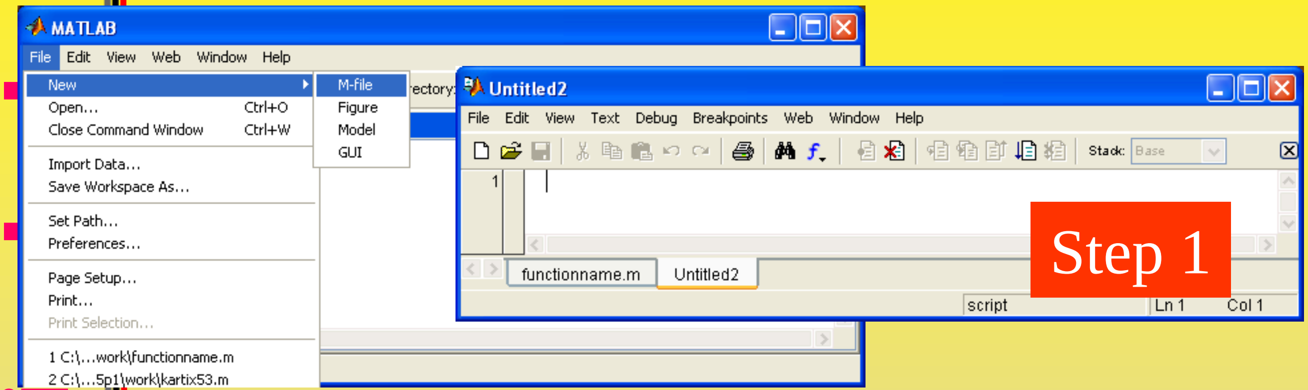


The script files

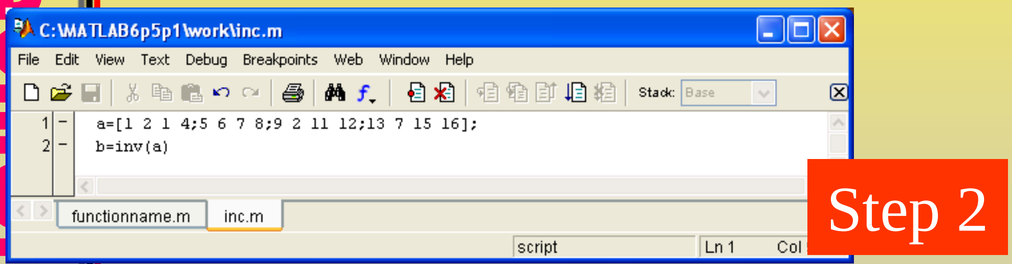
definition

Scripts are the simplest kind of M-file. They are useful for automating blocks of MATLAB commands, such as computations you have to perform repeatedly from the command line. Scripts can operate on existing data in the workspace, or they can create new data on which to operate. Although scripts do not return output arguments, any variables that they create remain in the workspace so you can use them in further computations. In addition, scripts can produce graphical output using commands like plot

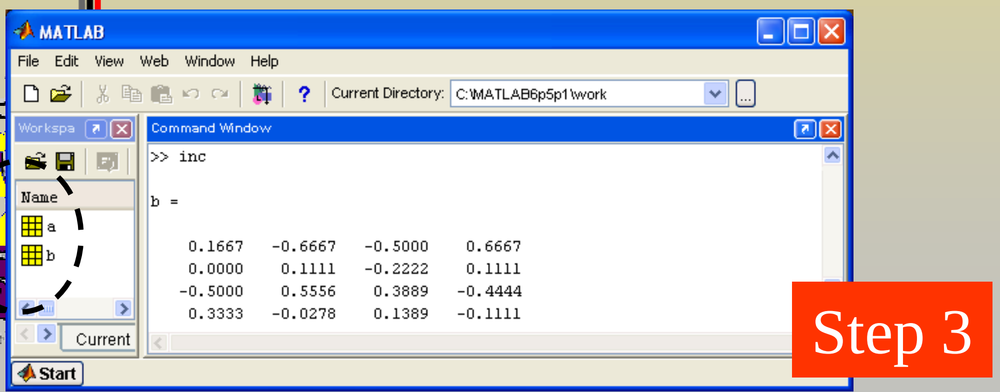
How can I make a script file ?



Open a new notepad



Write yours program and save it.



Run yours program

The function files

definition

M-files can be either scripts or functions. Scripts are simply files containing a sequence of MATLAB statements. Functions make use of their own local variables and accept input arguments.

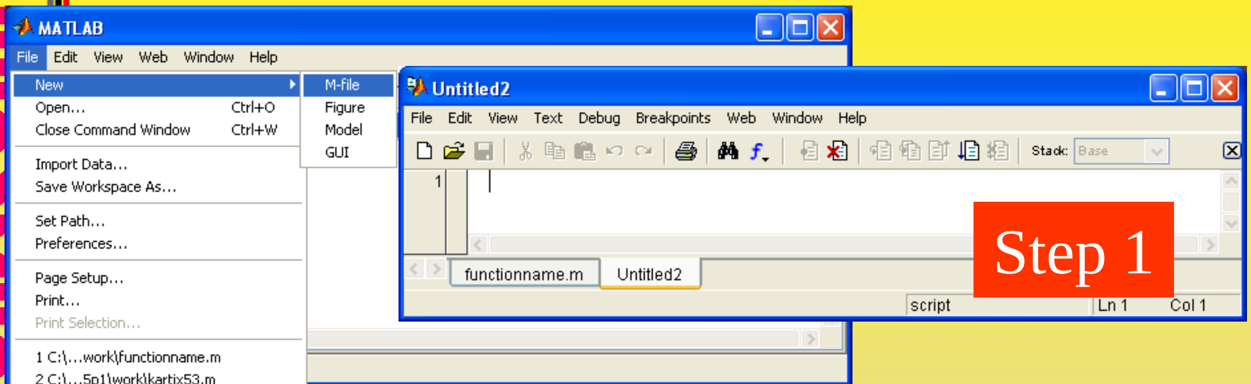
The name of a function, as defined in the first line of the M-file, should be the same as the name of the file without the .m extension.

For example, the existence of a file on disk called stat.m with

```
function [mean,stdev] = stat(x)
```

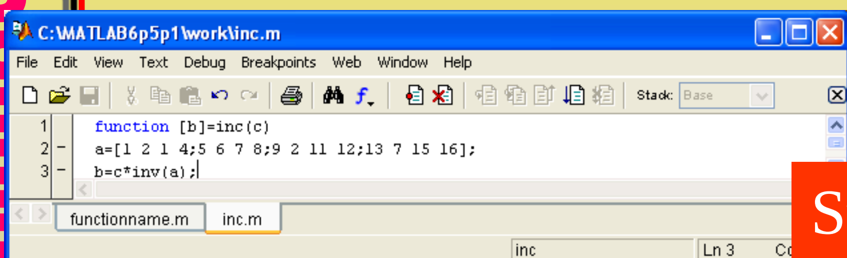
How can I make a function file ?

Simulationstechnik



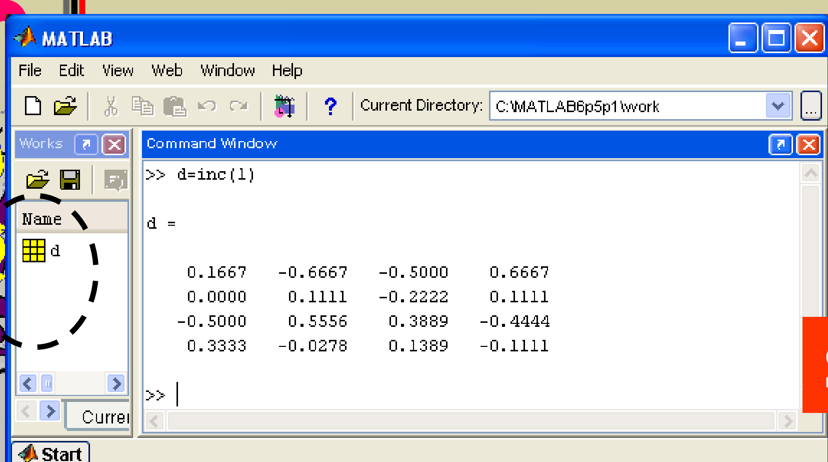
Step 1

Open a new notepad



Step 2

Write your program and save it.

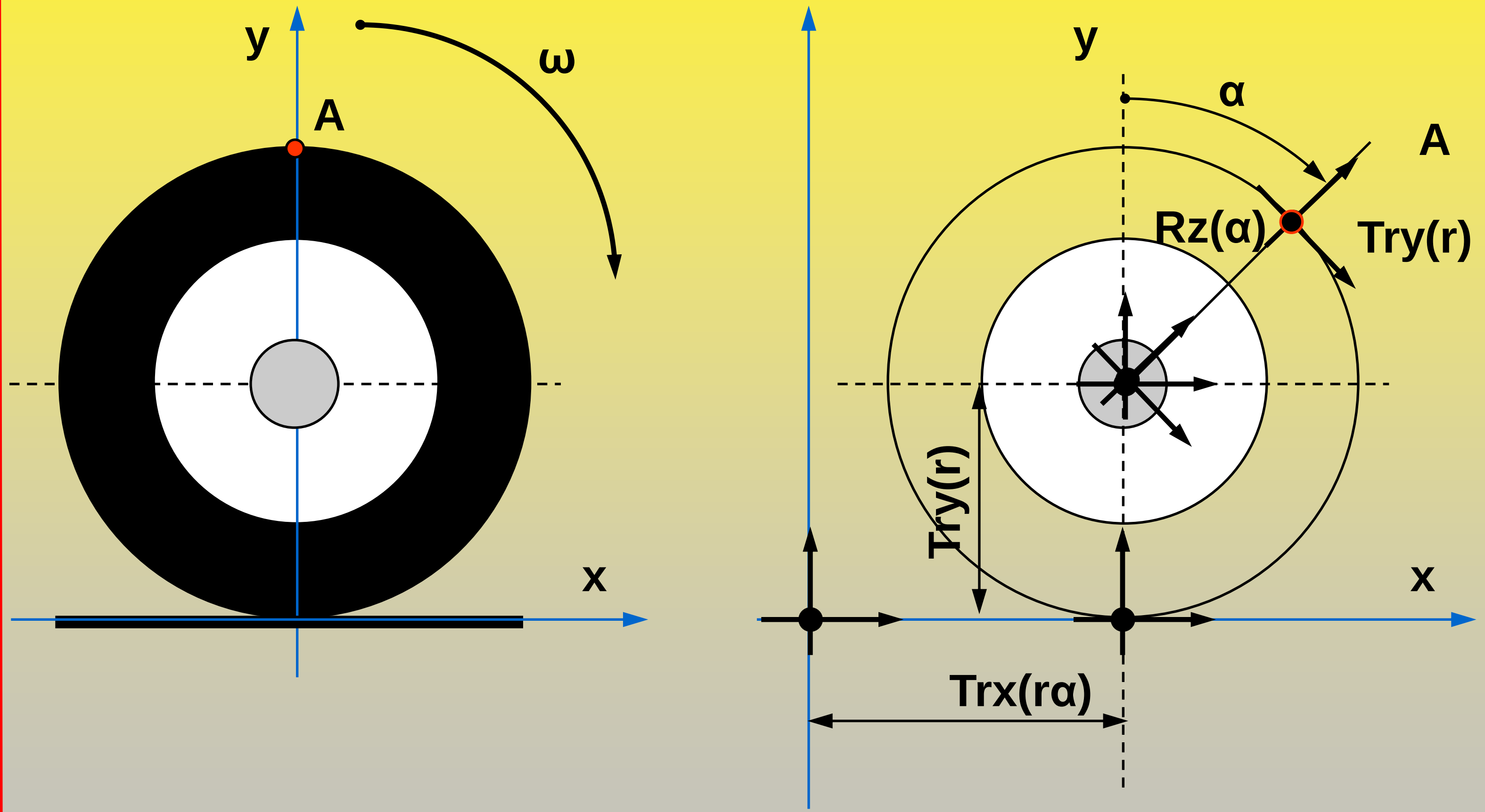


Step 3

Run your program

Drill problem

Plot the trajectory of one point of the wheel during the car locomotion



Simulationstechnik

C:\MATLAB6p5p1\work\functionname.m

```
% here we do not need any statement
r=1;
% the first positon
Txy=[1,zeros(1,3);
    [q(1)*r;r;0],eye(3,3)];
Rz=[1,zeros(1,3);
    zeros(3,1),[cos(q(1)),sin(q(1)),0;-sin(q(1)),cos(q(1)),0;0,0,1]];
Ty=[1,zeros(1,3);
    [0;r;0],eye(3,3)];
P=Txy*Rz*Ty;
poz1=P(2:3,1);
%the second position
Txy=[1,zeros(1,3);
    [q(2)*r;r;0],eye(3,3)];
Rz=[1,zeros(1,3);
    zeros(3,1),[cos(q(2)),sin(q(2)),0;-sin(q(2)),cos(q(2)),0;0,0,1]];
Ty=[1,zeros(1,3);
    [0;r;0],eye(3,3)];
P=Txy*Rz*Ty;
poz2=P(2:3,1);
%the third position
Txy=[1,zeros(1,3);
    [q(3)*r;r;0],eye(3,3)];
Rz=[1,zeros(1,3);
    zeros(3,1),[cos(q(3)),sin(q(3)),0;-sin(q(3)),cos(q(3)),0;0,0,1]];
Ty=[1,zeros(1,3);
    [0;r;0],eye(3,3)];
P=Txy*Rz*Ty;
poz3=P(2:3,1);
%plot the positions
Poz=[poz1,poz2,poz3,poz4];
plot(Poz(1,:),Poz(2,:))
```

script Ln 16 Col 70

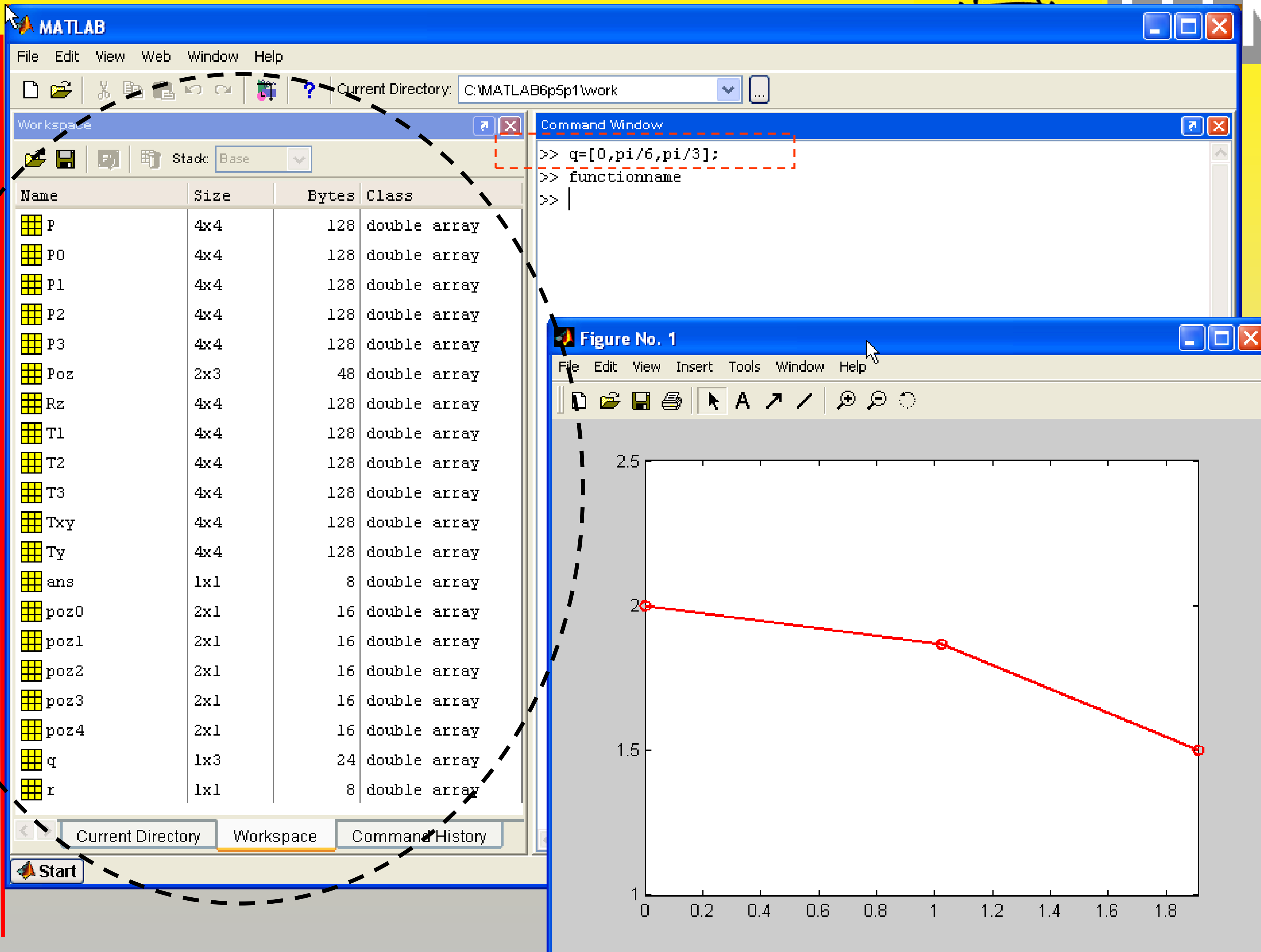
- script

TxyRzTy

X 3

plot

Simulationstechnik



 The MATLAB command window for script slution

Simulationstechnik

C:\MATLAB6p5p1\work\functionname.m*

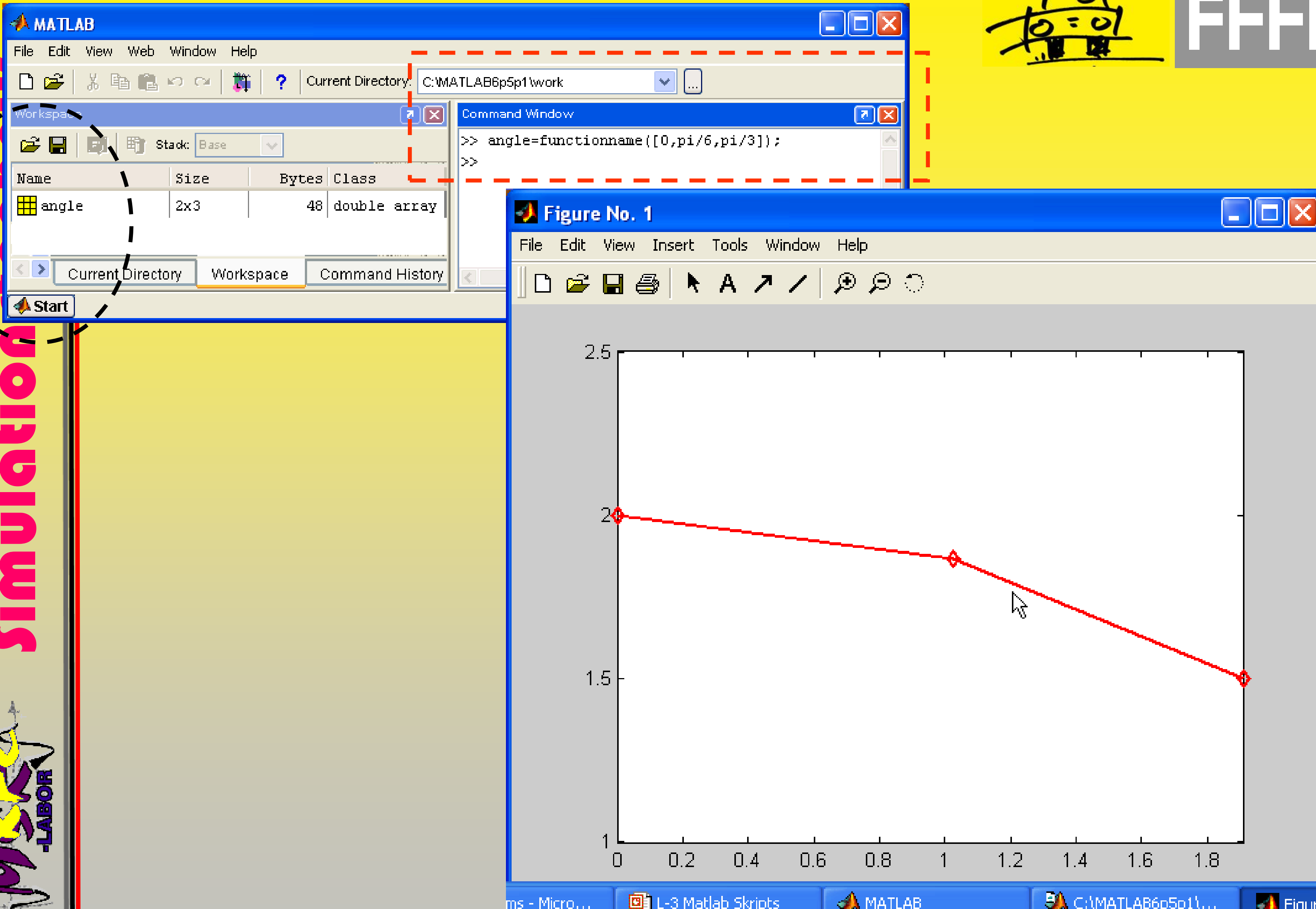
- function

```
function Poz=functionname(q)
r=1;
% the first positon
Txy=[1,zeros(1,3);
    [q(1)*r;r;0],eye(3,3)];
Rz=[1,zeros(1,3);
    zeros(3,1),[cos(q(1)),sin(q(1)),0;-sin(q(1)),cos(q(1)),0;0,0,1]];
Ty=[1,zeros(1,3);
    [0;r;0],eye(3,3)];
P=Txy*Rz*Ty;
poz1=P(2:3,1);
%the second position
Txy=[1,zeros(1,3);
    [q(2)*r;r;0],eye(3,3)];
Rz=[1,zeros(1,3);
    zeros(3,1),[cos(q(2)),sin(q(2)),0;-sin(q(2)),cos(q(2)),0;0,0,1]];
Ty=[1,zeros(1,3);
    [0;r;0],eye(3,3)];
P=Txy*Rz*Ty;
poz2=P(2:3,1);
%the third position
Txy=[1,zeros(1,3);
    [q(3)*r;r;0],eye(3,3)];
Rz=[1,zeros(1,3);
    zeros(3,1),[cos(q(3)),sin(q(3)),0;-sin(q(3)),cos(q(3)),0;0,0,1]];
Ty=[1,zeros(1,3);
    [0;r;0],eye(3,3)];
P=Txy*Rz*Ty;
poz3=P(2:3,1);
%plot the positions
Poz=[poz1,poz2,poz3];
plot(Poz(1,:),Poz(2,:))
```

functionname Ln 1 Col 29

The same

Simulation



 Matlab command window for function solution

Simulationstechnik

Special statements

- for
- while
- break
- error
- return
- if

for

The general format is

```
for variable = expression
    statement
    ...
    statement
end
```

Examples

```
for m = 1:k
    for n = 1:k
        a(m,n) = 1/(m+n -1);
    end
end
```

while

The general format is

```
while expression
    statements
end
```

The statements are executed while the real part of expression has all nonzero elements. Expression is usually of the form

expression rel_op expression

rel_op is ==, <, >, <=, >=, ~=

Examples

```
a = 1;
while (1+a) > 1
    a = a/2;
end
a = a*2
```

if

The general format is

```
if expression
    statements
end
```

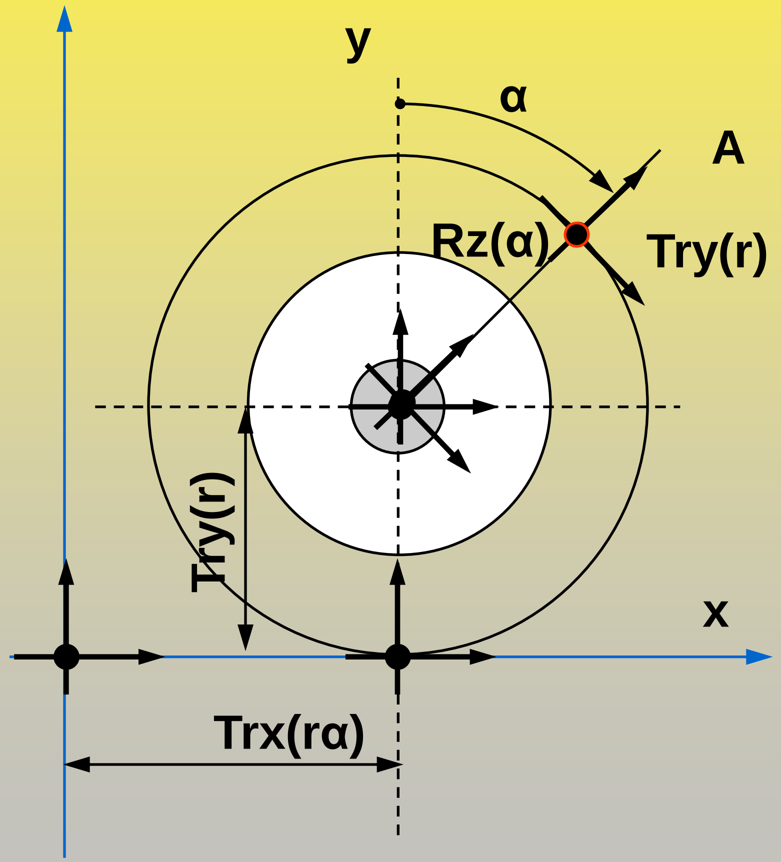
```
if expression1
    statements1
elseif expression2
    statements2
else
    statements3
end
```

Example

```
if ((a >= 1)&(b >= 2))
   c = 1;
else
   c=2
end;
```

Drill problem

Improve yours m files for plotting the trajectory of one point of the wheel during the car locomotion



α=0:0.1:2pi

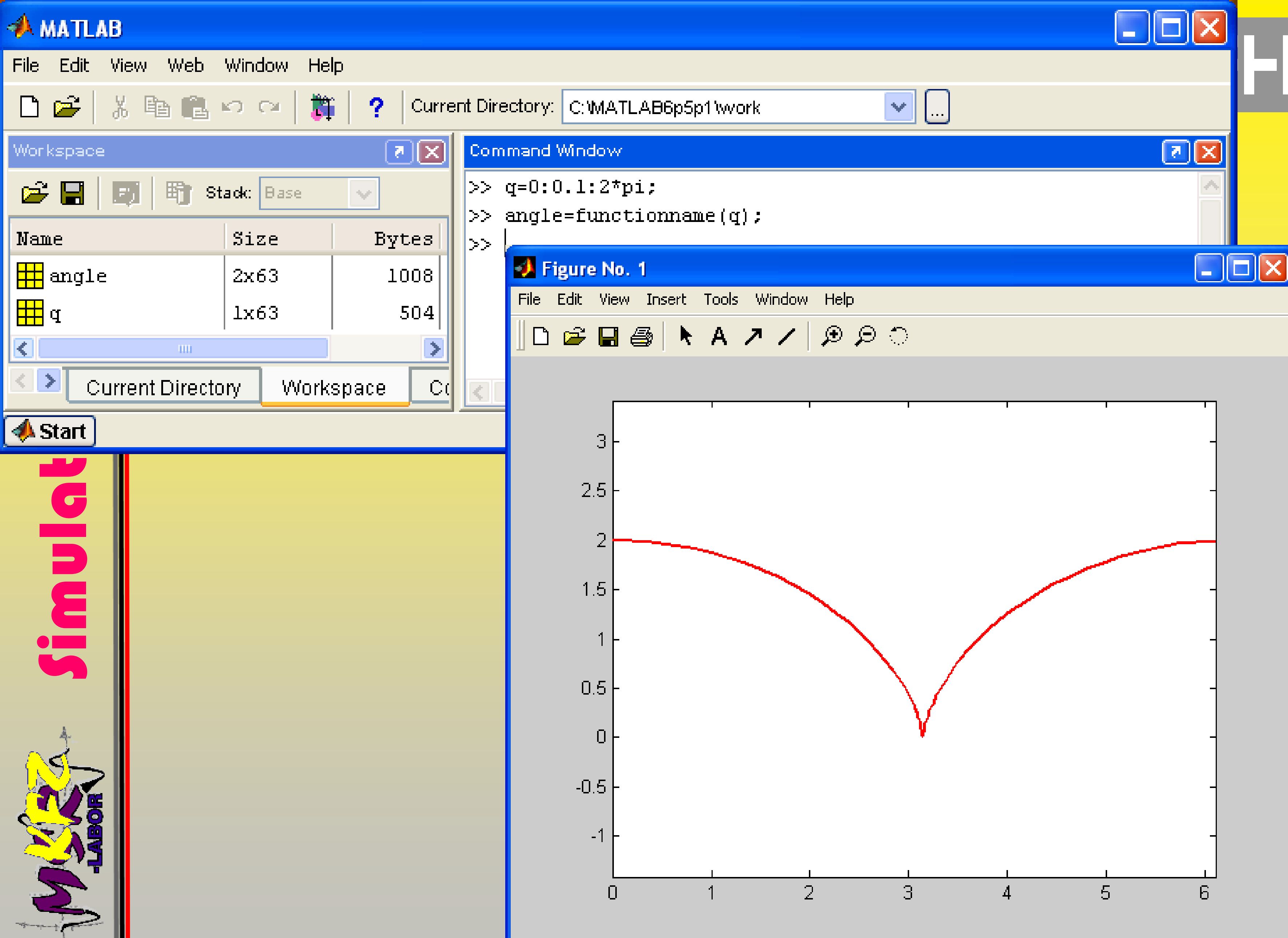
Simulationstechnik

C:\MATLAB6p5p1\work\functionname.m

File Edit View Text Debug Breakpoints Web Window Help

```
function Poz=functionname(q)
r=1;
% the length of the vector q
n=length(q);
% the position of A point
Poz=[];
for i=1:n
    Txy=[1,zeros(1,3);
        [q(i)*r;r;0],eye(3,3)];
    Rz=[1,zeros(1,3);
        zeros(3,1),[cos(q(i)),sin(q(i)),0;-sin(q(i)),cos(q(i)),0;0,0,1]
    Ty=[1,zeros(1,3);
        [0;r;0],eye(3,3)];
    P=Txy*Rz*Ty;
    Poz=[Poz,P(2:3,1)];
end
%the graphical representation
plot(Poz(1,:),Poz(2,:))
```

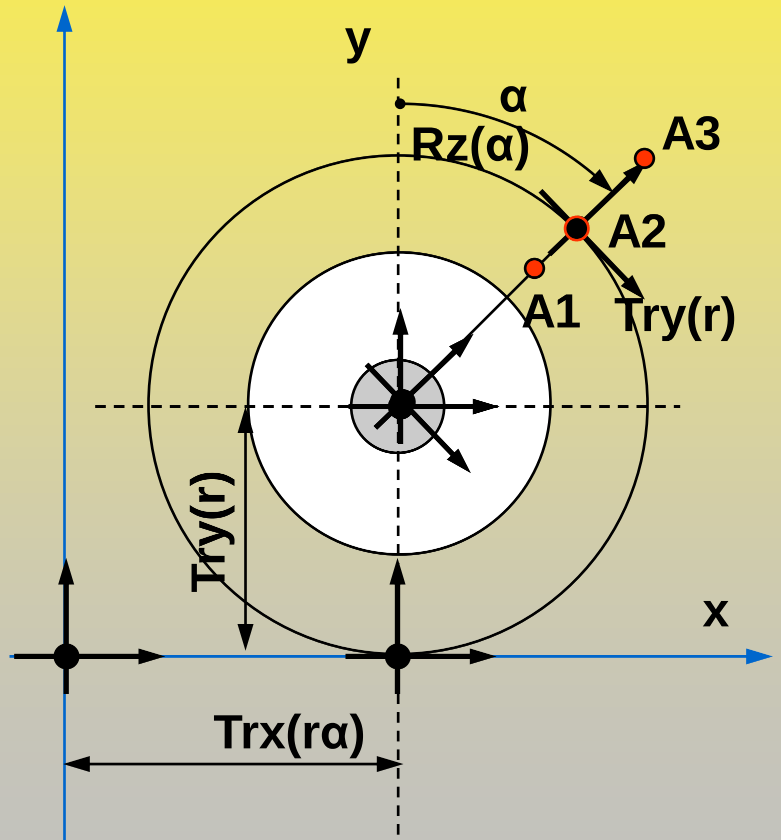
functionname Ln 4 Col 13



 Matlab command window for function solution

Drill problem

Let complicate a little bit the problem and take all the three possibility For the A point position

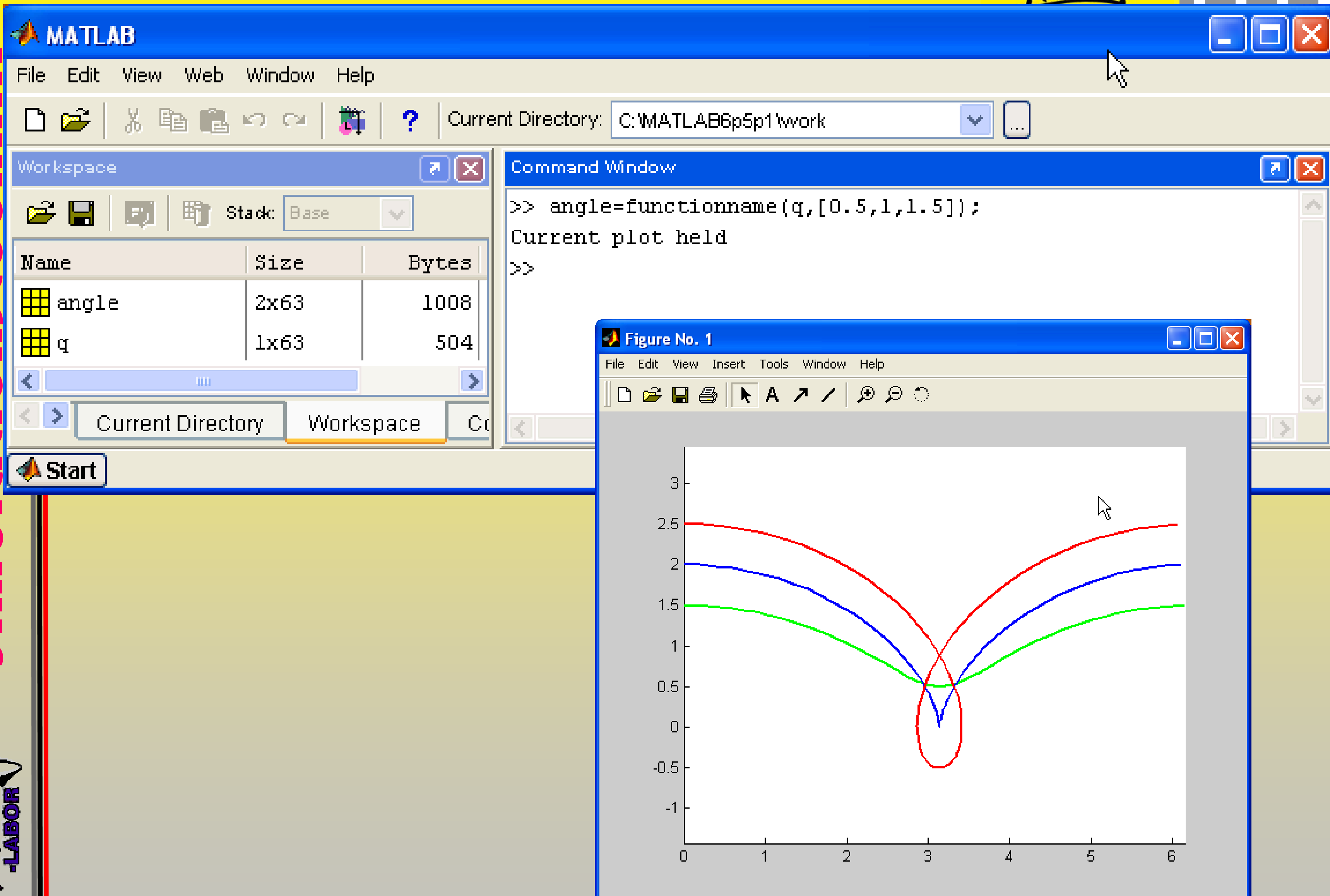


α=0:0.1:2pi
r1= .5
r2=1
r3=2

Simulationstechnik

C:\MATLAB6p5p1\work\functionname.m

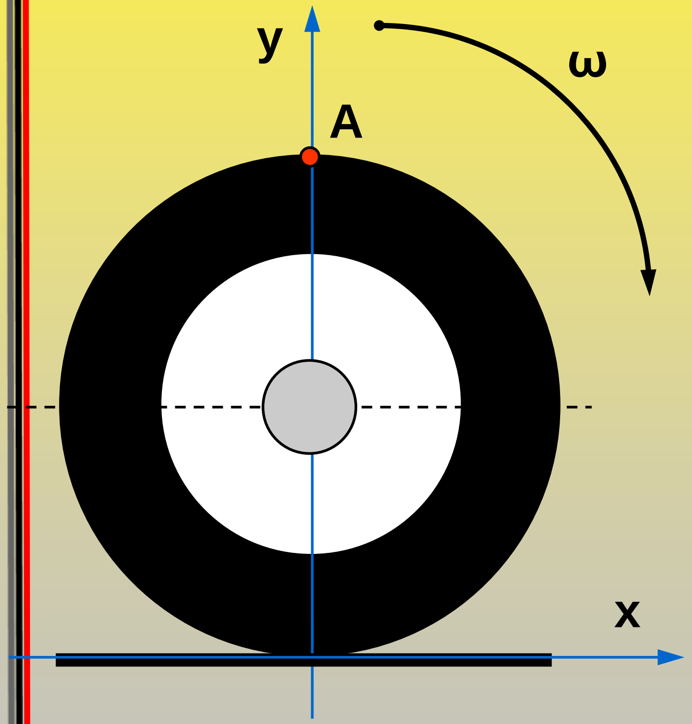
```
function Poz=functionname(q,R)
r=1;
% the length of the vector q
n=length(q);
m=length(R);
% the position of A point
figure
hold
for j=1:m
    Poz=[];
    for i=1:n
        Txy=[1,zeros(1,3);
            [q(i)*r;r;0],eye(3,3)];
        Rz=[1,zeros(1,3);
            zeros(3,1),[cos(q(i)),sin(q(i)),0;-sin(q(i)),cos(q(i)),0;0,0,1]];
        Ty=[1,zeros(1,3);
            [0;R(j);0],eye(3,3)];
        P=Txy*Rz*Ty;
        Poz=[Poz,P(2:3,1)];
    end
    %the graphical representation
    plot(Poz(1,:),Poz(2,:))
    axis('equal')
end
```

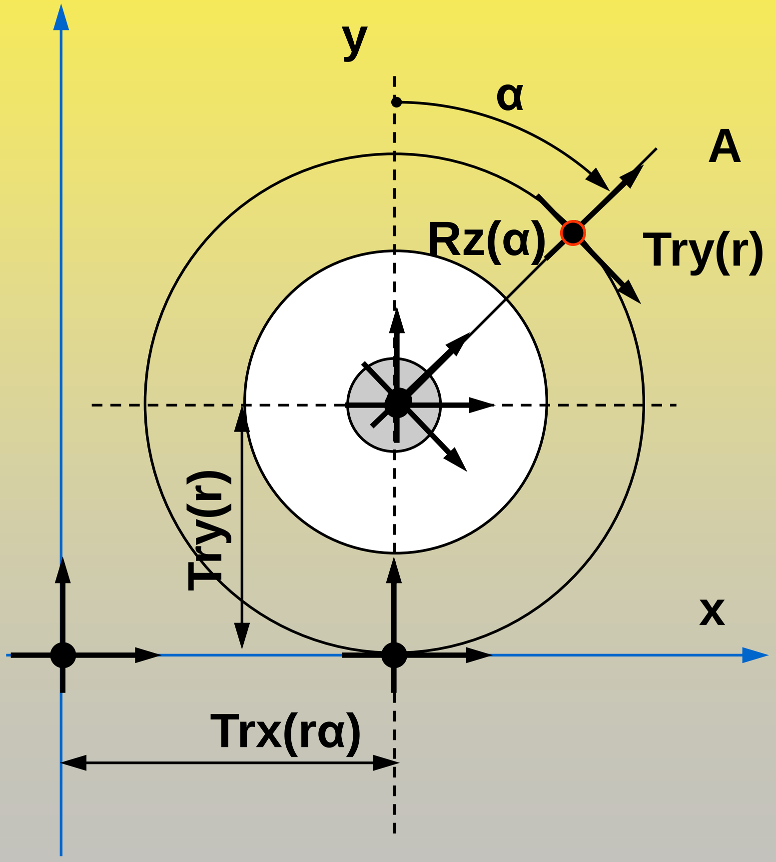

MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5p1\work
Workspace
Stack: Base
Name Size Bytes
angle 2x63 1008
q 1x63 504
Current Directory
Workspace
Command Window
>> angle=functionname(q,[0.5,1,1.5]);
Current plot held
>>
Start
Figure No. 1
File Edit View Insert Tools Window Help
3
2.5
2
1.5
1
0.5
0
-0.5
-1
0 1 2 3 4 5 6

Drill problem

Let transform our problem into a kinematics one

$$\alpha = \int \omega dt$$





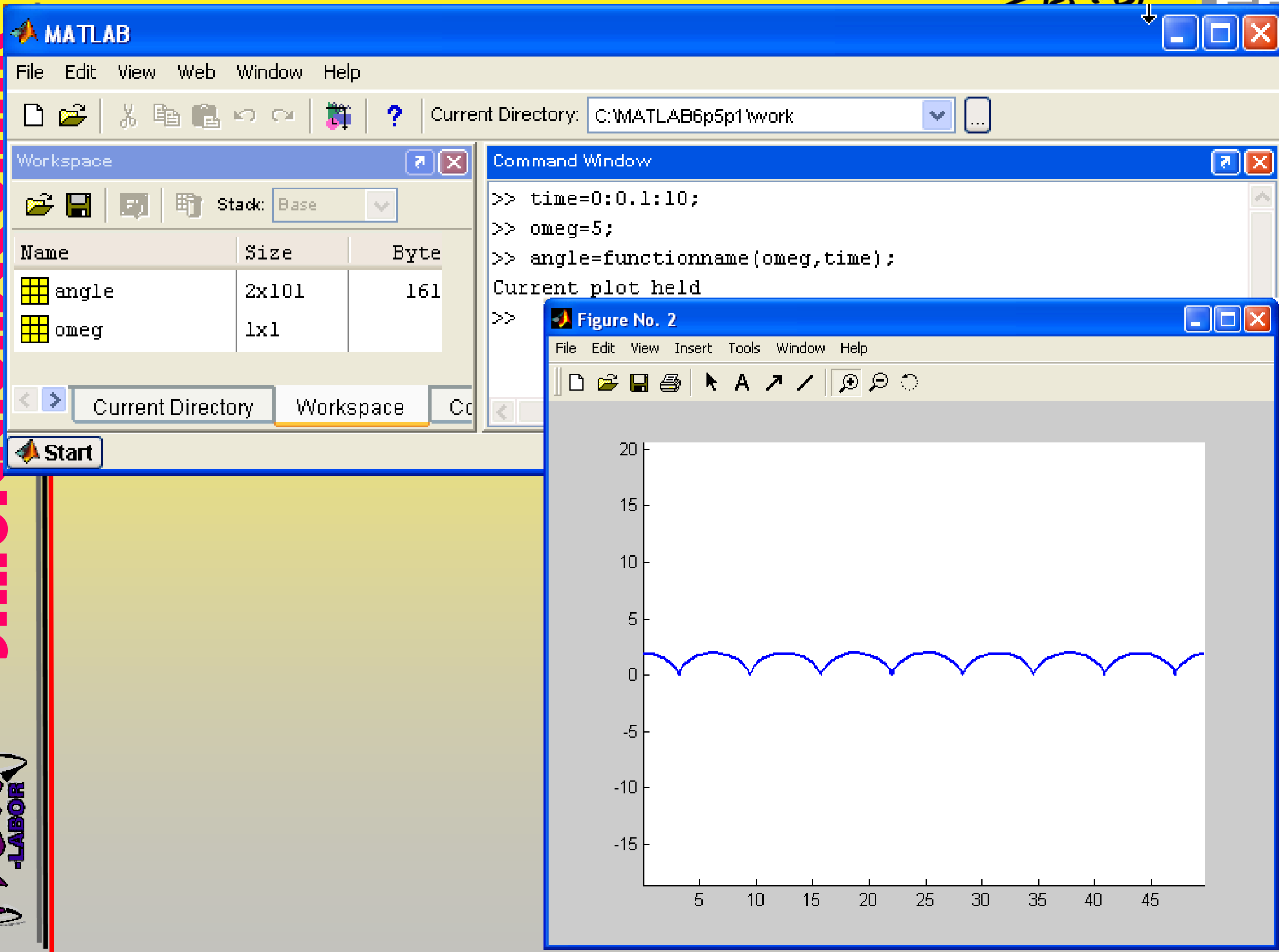
C:\MATLAB6p5p1\work\functionname.m*

File Edit View Text Debug Breakpoints Web Window Help

```
function Poz=functionname(omeg,time)
r=1;
% the length of the vector time
n=length(time);

    figure
    hold
    Poz=[];
    for i=1:n
        % we presume that omeg is constant
        q=omeg*time(i);
        Txy=[1,zeros(1,3);
            [q*r;r;0],eye(3,3)];
        Rz=[1,zeros(1,3);
            zeros(3,1),[cos(q),sin(q),0;-sin(q),cos(q),0;0,0,1]];
        Ty=[1,zeros(1,3);
            [0;r;0],eye(3,3)];
        P=Txy*Rz*Ty;
        Poz=[Poz,P(2:3,1)];
    end
    %the graphical representation
    plot(Poz(1,:),Poz(2,:))
    axis('equal')
```

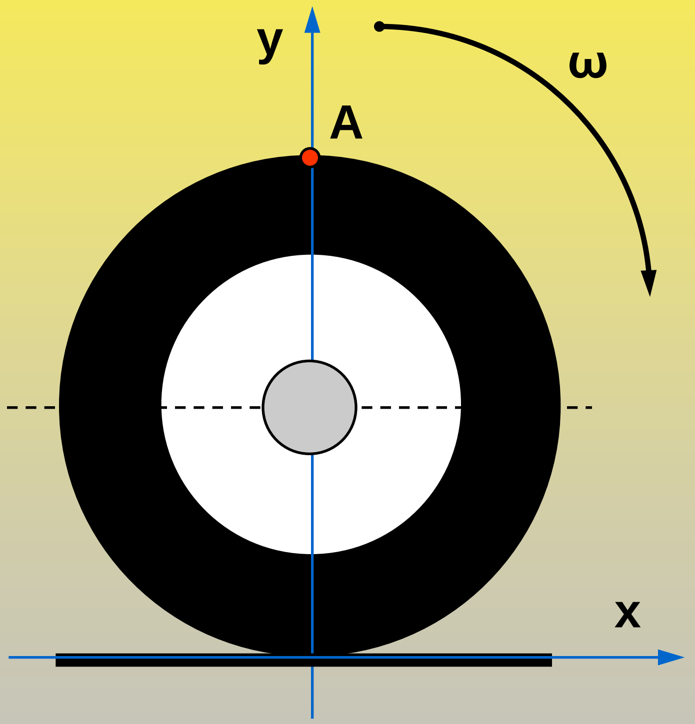
functionname Ln 6 Col 5

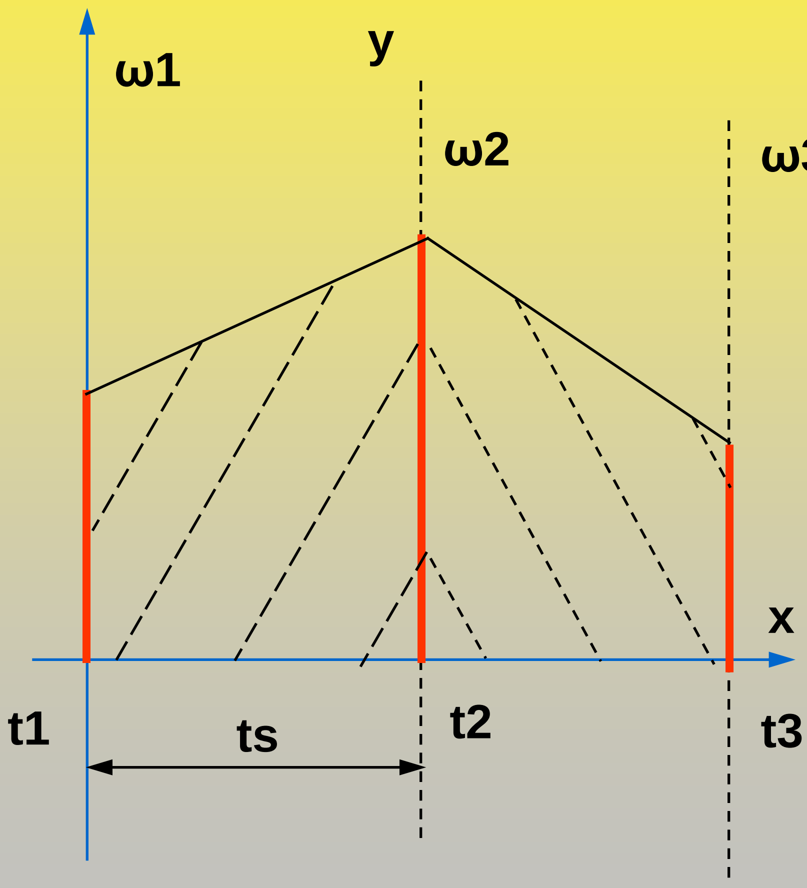
Simulationstechnik



Drill problem

And now let complicate the problem and impose a variable omega





Simulationstechnik

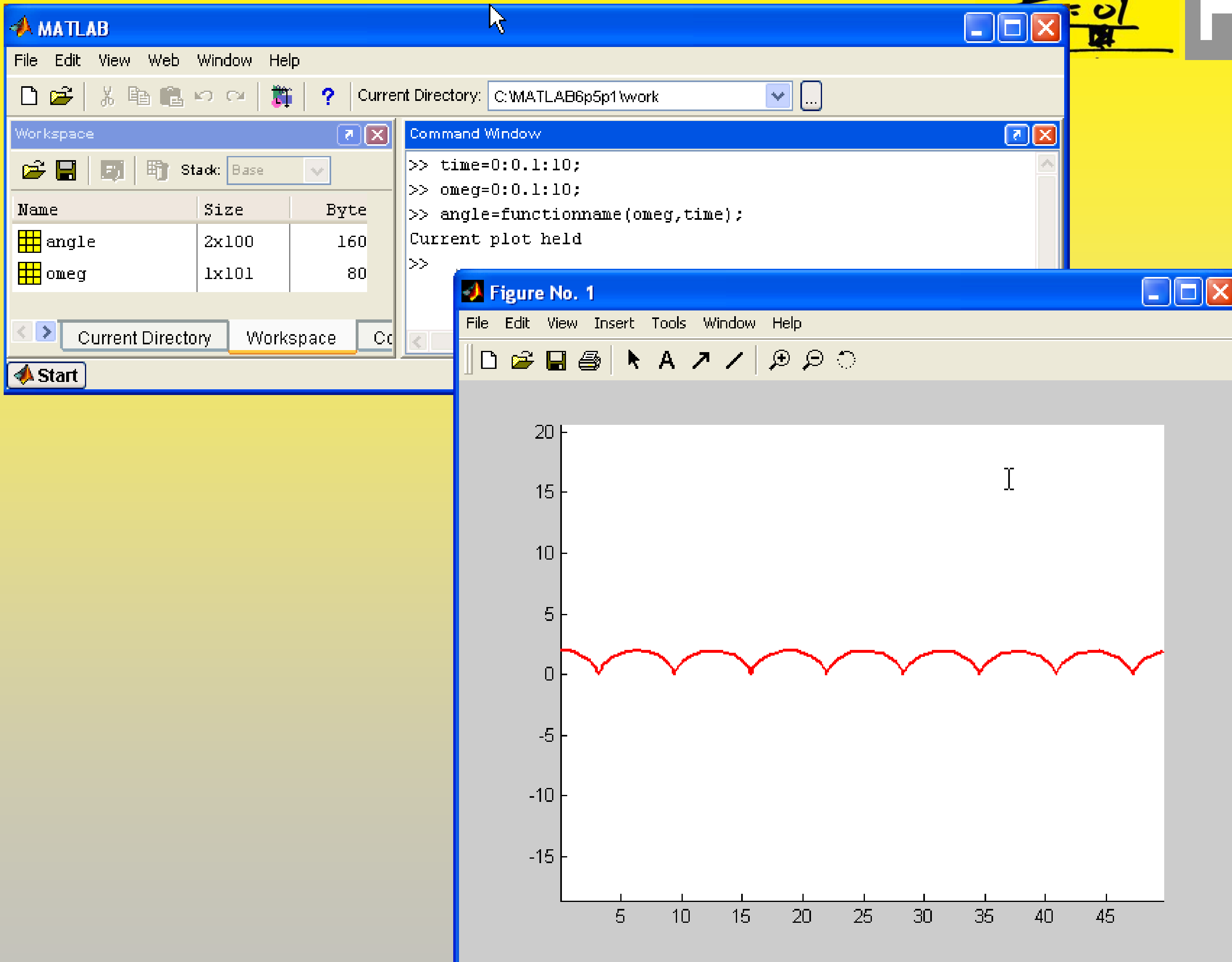
C:\MATLAB6p5p1\work\functionname.m

File Edit View Text Debug Breakpoints Web Window Help

```
function Poz=functionname(omeg,time)
r=1;
% the length of the vector time
n=length(time);

    figure
    hold
    Poz=[];
    q=0;
    for i=1:n-1
        % we integrate the omeg
        q=q+0.5*(omeg(i)+omeg(i+1))*(time(i+1)-time(i));
        Txy=[1,zeros(1,3);
            [q*r;r;0],eye(3,3)];
        Rz=[1,zeros(1,3);
            zeros(3,1),[cos(q),sin(q),0;-sin(q),cos(q),0;0,0,1]];
        Ty=[1,zeros(1,3);
            [0;r;0],eye(3,3)];
        P=Txy*Rz*Ty;
        Poz=[Poz,P(2:3,1)];
    end
    %the graphical representation
    plot(Poz(1,:),Poz(2,:))
    axis('equal')
```

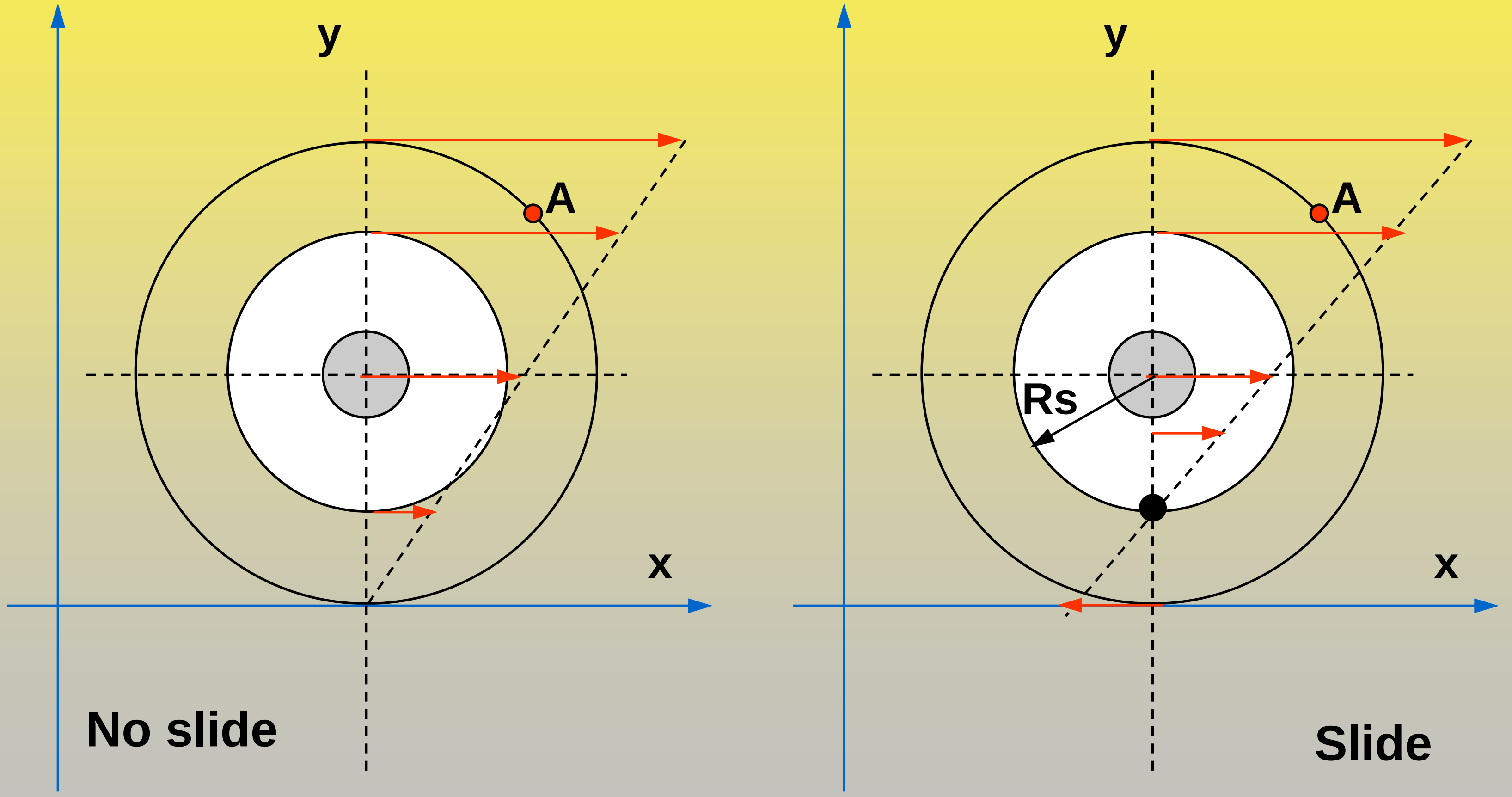
functionname   Ln 11   Col 33

Simulationstechnik



Drill problem

Let simulate now the trajectory when the wheel is slide

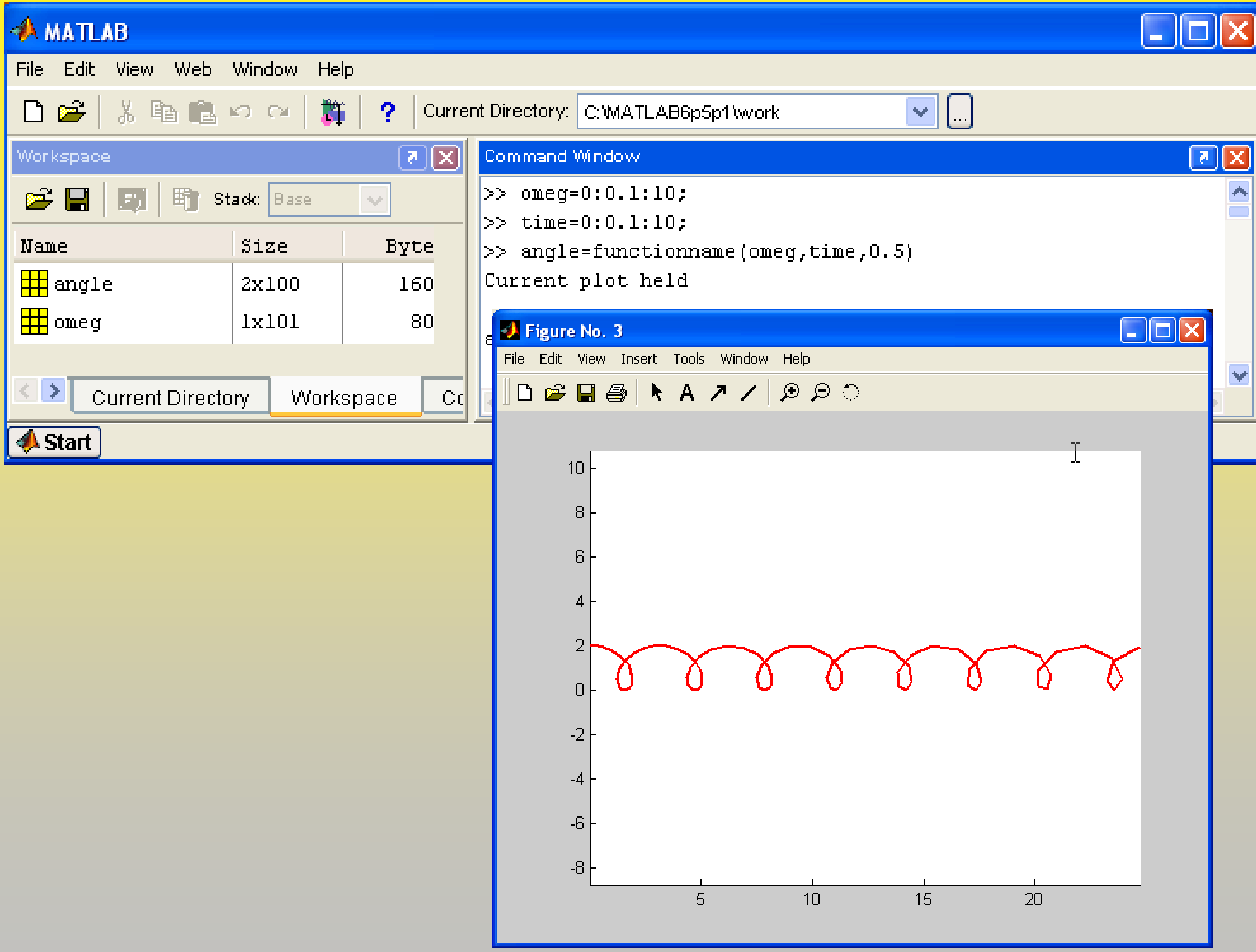


Simulationstechnik

C:\MATLAB6p5p1\work\functionname.m

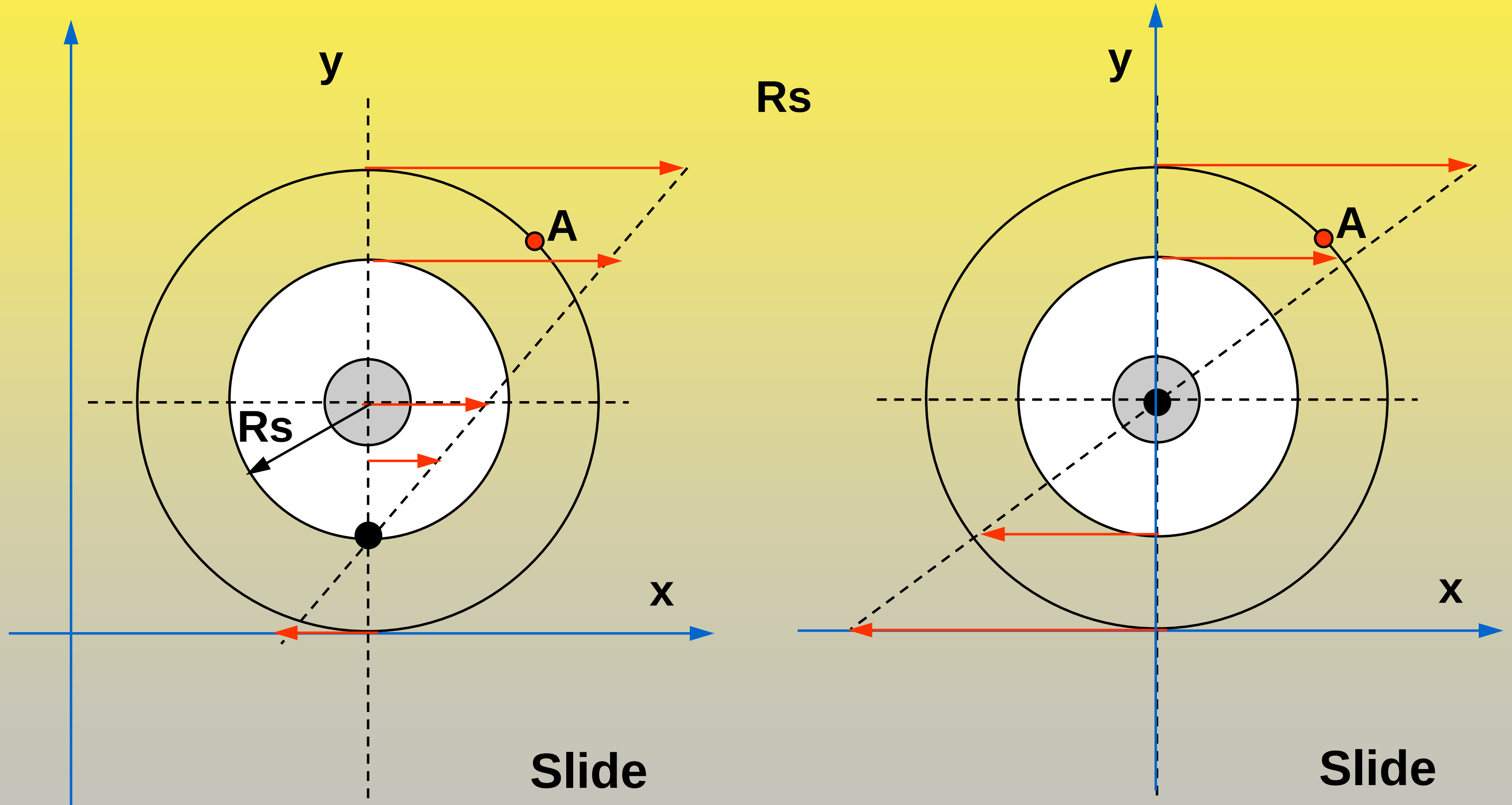
```
function Poz=functionname(omeg,time,Rs)
r=1;
% the length of the vector time
n=length(time);

    figure
    hold
    Poz=[];
    q=0;
    for i=1:n-1
        % we integrate the omeg
        q=q+0.5*(omeg(i)+omeg(i+1))*(time(i+1)-time(i));
        Txy=[1,zeros(1,3);
            [q*Rs;1;0],eye(3,3)];
        Rz=[1,zeros(1,3);
            zeros(3,1),[cos(q),sin(q),0;-sin(q),cos(q),0;0,0,1]];
        Ty=[1,zeros(1,3);
            [0;r;0],eye(3,3)];
        P=Txy*Rz*Ty;
        Poz=[Poz,P(2:3,1)];
    end
    %the graphical representation
    plot(Poz(1,:),Poz(2,:))
    axis('equal')
```

Simulationstechnik



Drill problem

And now verify the simulation by imposing Rs=0



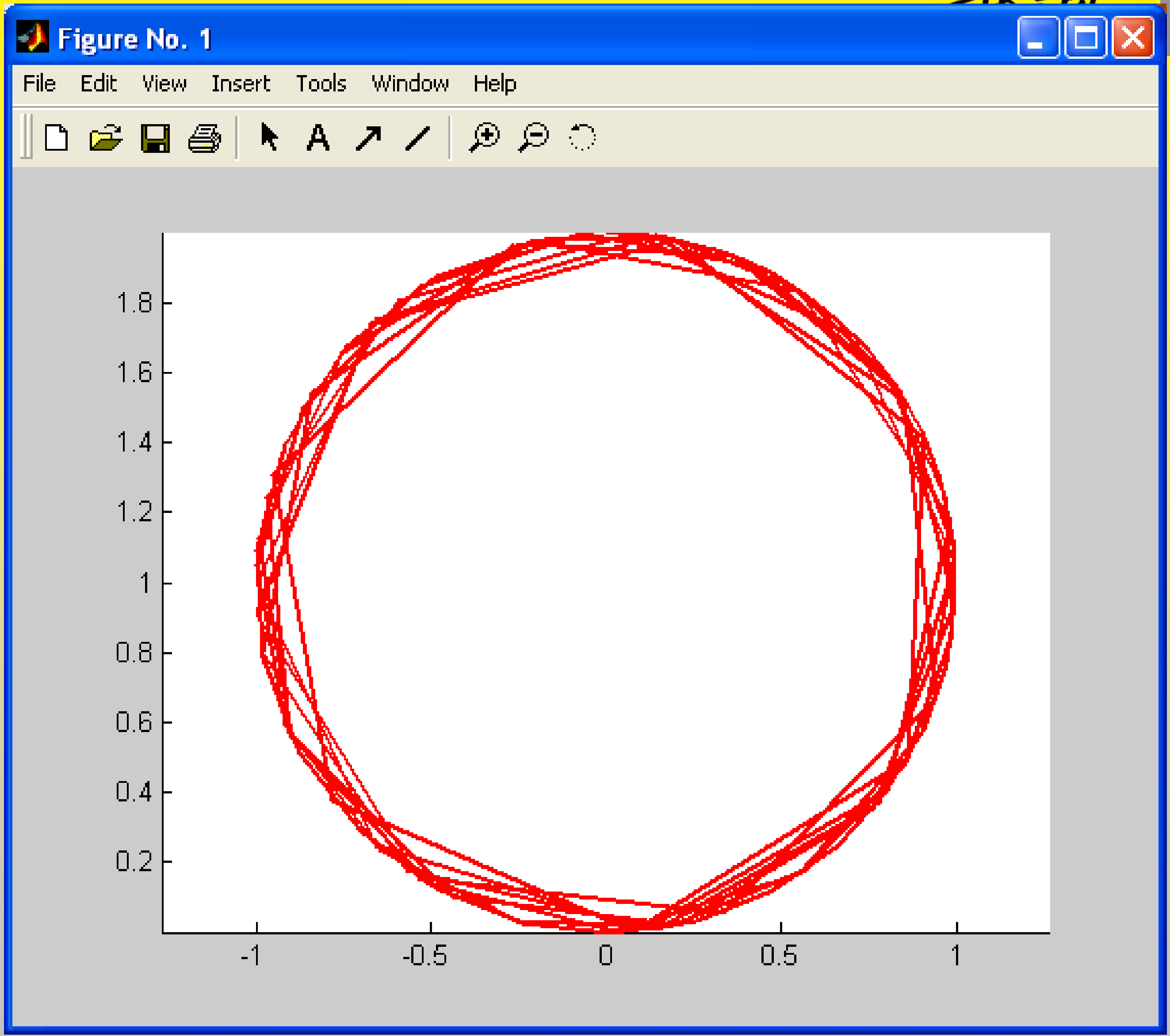
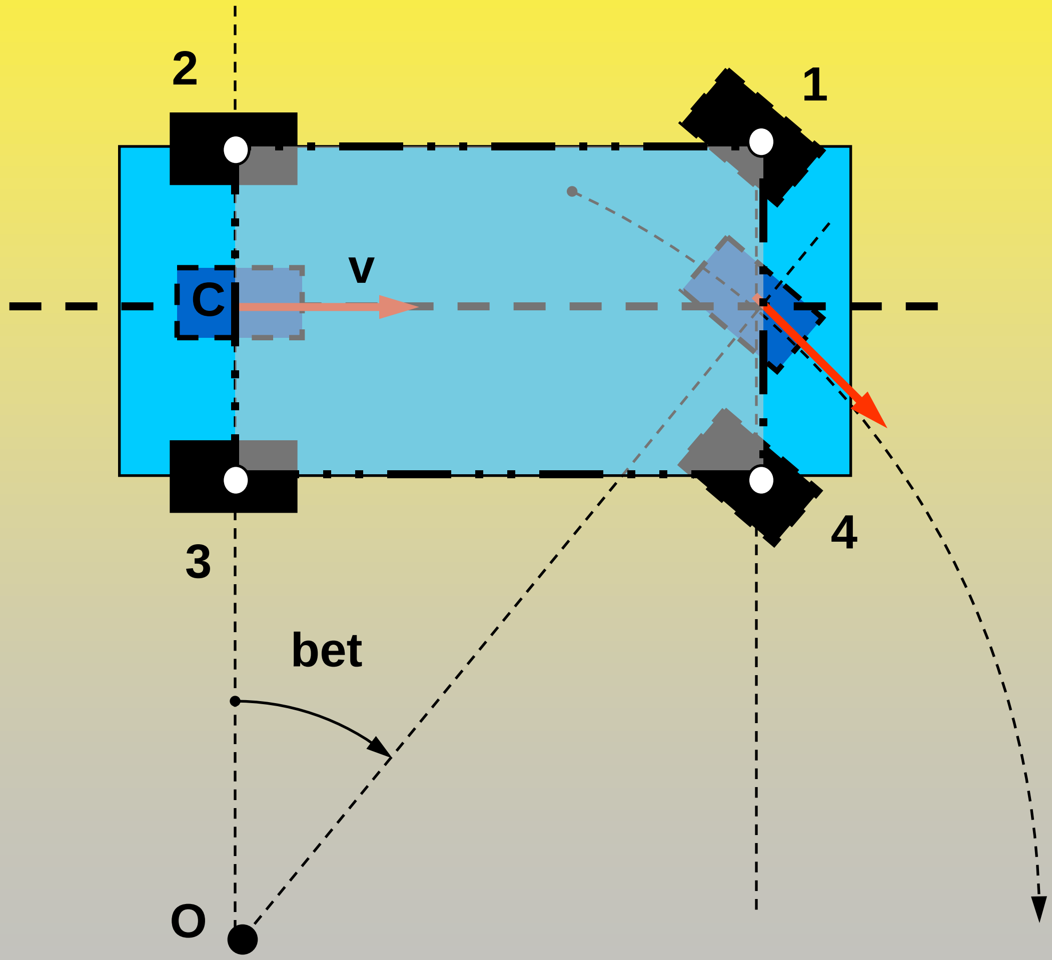
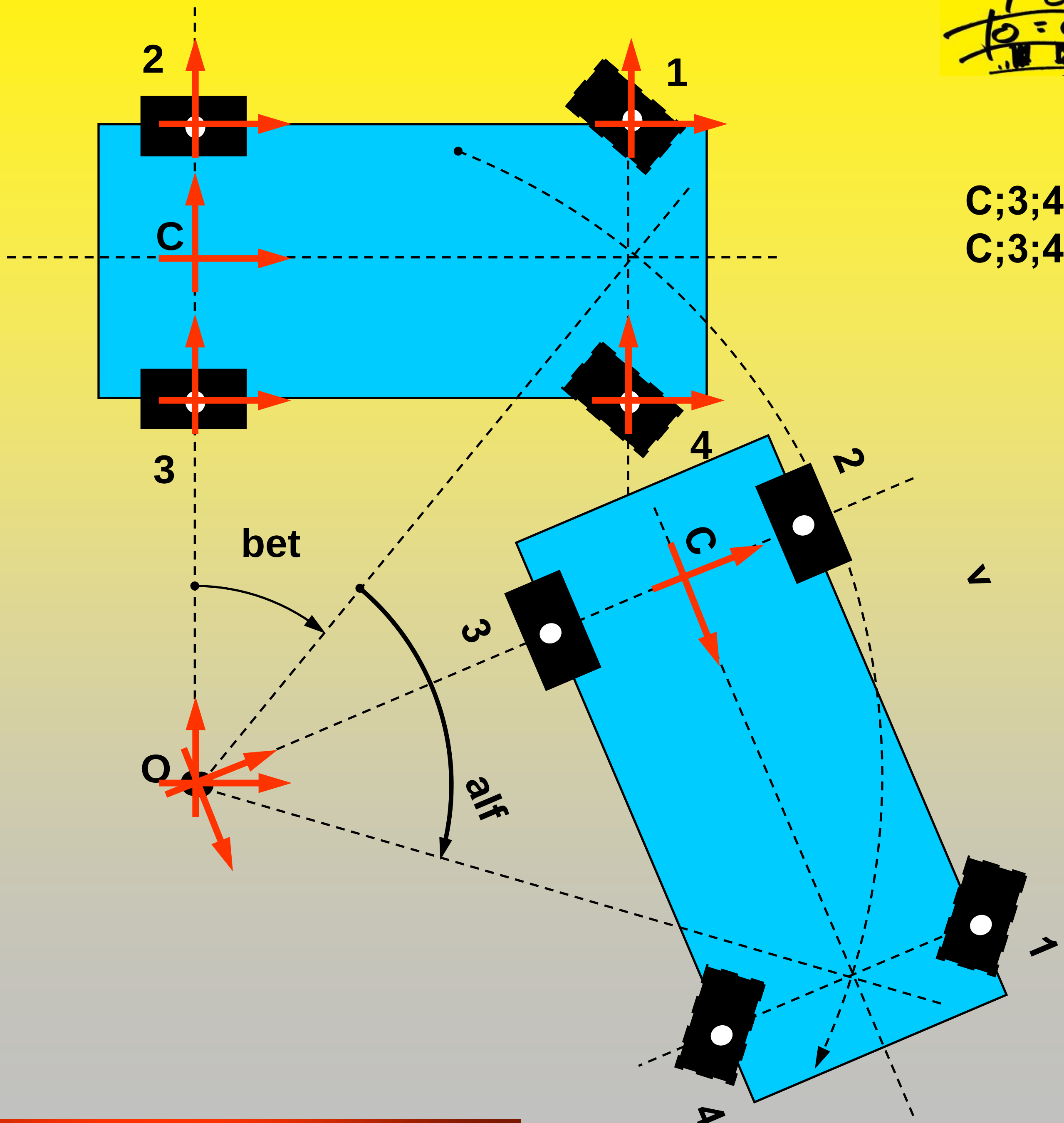


Figure No. 1
File Edit View Insert Tools Window Help
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
-1
-0.5
0
0.5
1

Exercise

Make a simulation of the successively positions for a car which turn in circle. Use for this the kinematical model of the car.





C;3;4;1;2;O;…
C;3;4;1;2;O;…

For the simulation program you can use the following program pieces

C:\MATLAB6p5p1\work\kartix53curs.m*

```matlab

%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%  the car dimensions %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=10;e=5;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%  the initial position  of the car%%%
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Ty3=[1,zeros(1,3);[0;-e/2;0],eye(3,3)];
P3=Ty3;
p3=P3(2:3,1);
Tx4=[1,zeros(1,3);[a;0;0],eye(3,3)];
P4=Ty3*Tx4;
p4=P4(2:3,1);
Ty1=[1,zeros(1,3);[0;e;0],eye(3,3)];
P1=Ty3*Tx4*Ty1;
p1=P1(2:3,1);
Tx2=[1,zeros(1,3);[-a;0;0],eye(3,3)];
P2=Ty3*Tx4*Ty1*Tx2;
p2=P2(2:3,1);
Tyc=[1,zeros(1,3);[0;-e/2;0],eye(3,3)];
C=Ty3*Tx4*Ty1*Tx2*Tyc;
Pct=[p1,p2,p3,p4,p1];
plot(Pct(1,:),Pct(2,:))

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

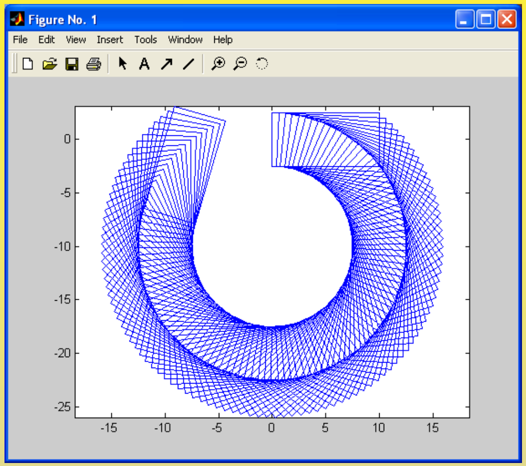
script | Ln 1 | Col 1

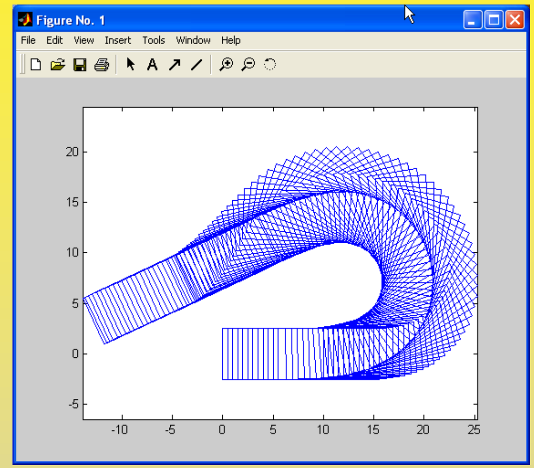
C:\MATLAB6p5p1\work\kartix53curs.m*

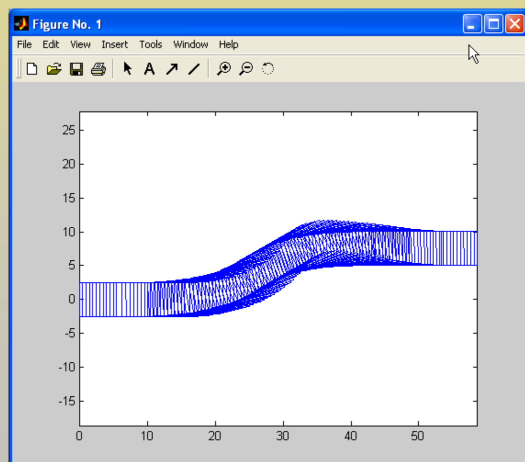
```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%  the rotated position  of the car%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


       %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
       %%%  alf is the angle with which turn the car              %%%
       %%%  ts is the sampling periode between 2 ploted positions  %%%
       %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
       alf=(1/a)*0.5*v*tan(bet)*ts;

       %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
       %%%   Pct The position of points 1,2,3,4   %%%
       %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
       C0=a/tan(bet);
       Tyo=[1,zeros(1,3);[0;C0;0],eye(3,3)];
       Rzb=[1,zeros(1,3);[0;0;0],[cos(alf),-sin(alf),0;sin(alf),cos(alf),0;0,0,1]];
       Ty3=[1,zeros(1,3);[0;-C0-e/2;0],eye(3,3)];
       P3=C*Tyo*Rzb*Ty3;
       p3=P3(2:3,1);
       P4=P3*Tx4;
       p4=P4(2:3,1);
       P1=P4*Ty1;
       p1=P1(2:3,1);
       P2=P1*Tx2;
       p2=P2(2:3,1);
                  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                  %%%  C The position, orientation of C point  %%%
                  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                   C=P2*Tyc;
       Pct=[p1,p2,p3,p4,p1];
       plot(Pct(1,:),Pct(2,:));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

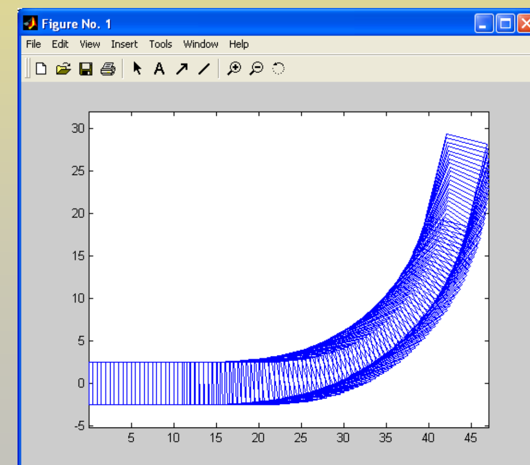
script | Ln 1 | Col 1

Examples of car trajectories









Simulationstechnik