POLITECNICO DI MILANO

AUTONOMOUS VEHICLES

S. Arrigoni

POLITECNICO
MILANO 1863

Introduction to
Requirements for the course

Today:Just Matlab!

- Install (tested on **2022a**) with academic license (better)
  https://www.software.polimi.it/software-download/studenti/matlab/
- Use it online from Virtual desktop (just for today..)
  https://www.ict.polimi.it/?page_id=485

From next time

- Windows users: dual boot **Ubuntu 20.04** (recommended)
  https://www.ubuntu-it.org/download
  or better: follow the instructions below
- Windows + iOS: Virtual machine
  Vmware? https://www.vmware.com/products/desktop-hypervisor.html

Pay attention: you need at least 80Gb of HD space

ROS

Logo of ROS Noetic Ninjemys

Follow instructions available at:
https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/

Software requirements

ROS Noetic

Kinetic | Melodic | **Noetic** | Dashing | Foxy | Windows

3. 1. 2. Install ROS on Remote PC

Open the terminal with `Ctrl`+`Alt`+`T` and enter below commands one at a time.
In order to check the details of the easy installation script, please refer to the script file.

```
$ sudo apt update
$ sudo apt upgrade
$ wget https://raw.githubusercontent.com/ROBOTIS-GIT/robotis_tools/master/install_ros_noetic.sh
$ chmod 755 ./install_ros_noetic.sh
$ bash ./install_ros_noetic.sh
```

Software requirements

- Try in a terminal to type:

```
raibuntu@RaiBuntu66:~/catkin_ws$ roscore
... logging to /home/raibuntu/.ros/log/97cf9a90-3cec-11ed-accc-272f0f9d8953/roslaunch-RaiBuntu66-17705.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://localhost:33145/
ros_comm version 1.15.14


SUMMARY
========

PARAMETERS
 * /rosdistro: noetic
 * /rosversion: 1.15.14

NODES
```

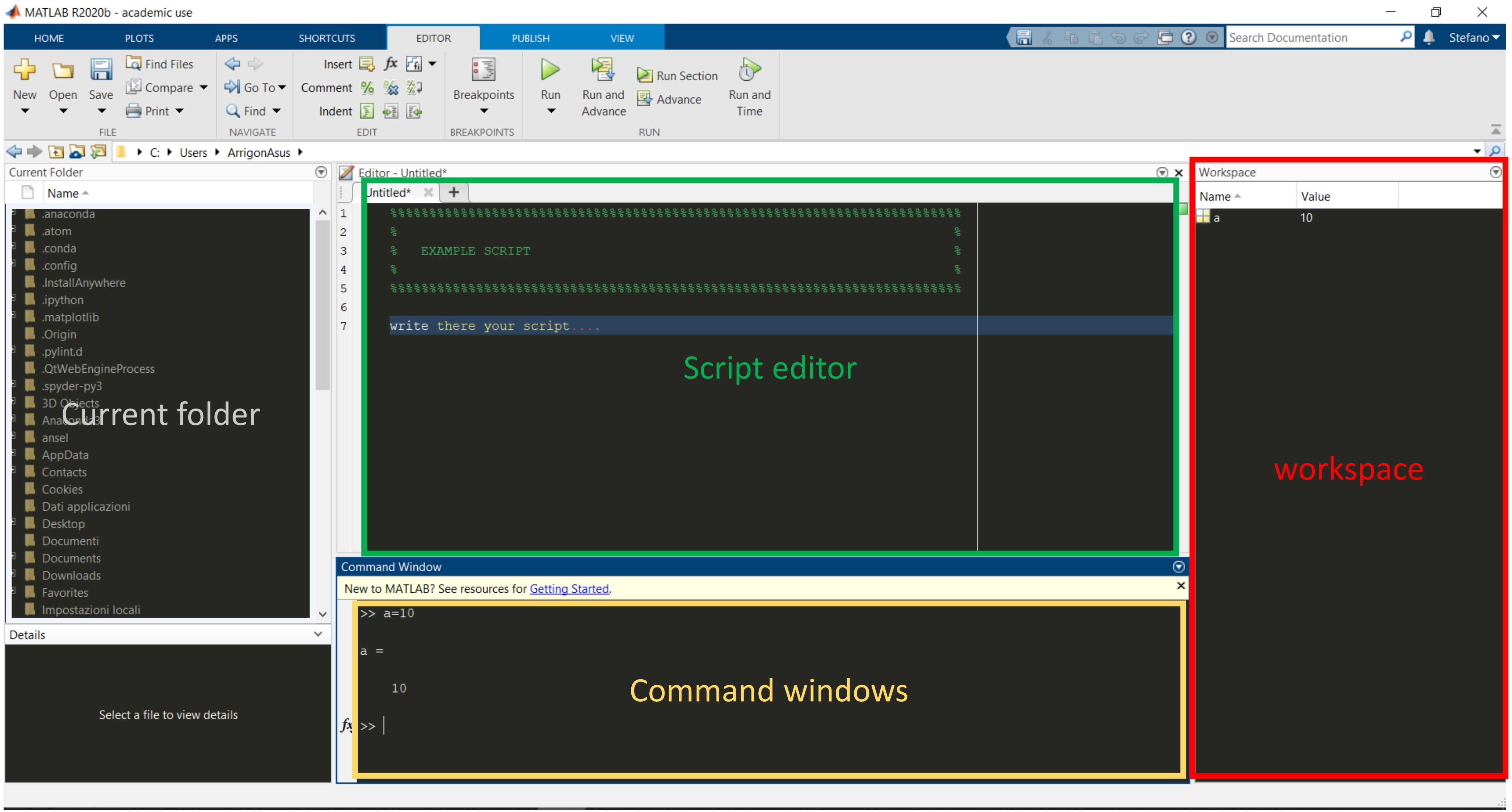
- Next time…

MATLAB®

MATLAB is a programming platform based on its own programming language

Useful for:

- Analyze data
- Develop algorithms (textual language)
- Create models and applications


MATLAB R2020b - academic use
HOME
PLOTS
APPS
SHORTCUTS
EDITOR
PUBLISH
VIEW
Search Documentation
Stefano
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
C: Users ArrigonAsus
Current Folder
Name
.anaconda
.atom
.conda
.config
.InstallAnywhere
.ipython
.matplotlib
.Origin
.pylint.d
.QtWebEngineProcess
.spyder-py3
3D Objects
ansel
AppData
Contacts
Cookies
Dati applicazioni
Desktop
Documenti
Documents
Downloads
Favorites
Impostazioni locali
Current folder
Details
Select a file to view details
Editor - Untitled*
Untitled*
% EXAMPLE SCRIPT %
write there your script....
Script editor
Command Window
New to MATLAB? See resources for Getting Started.
>> a=10
a =
10
>>
Command windows
Workspace
Name
Value
a
10
workspace

We will use it for:

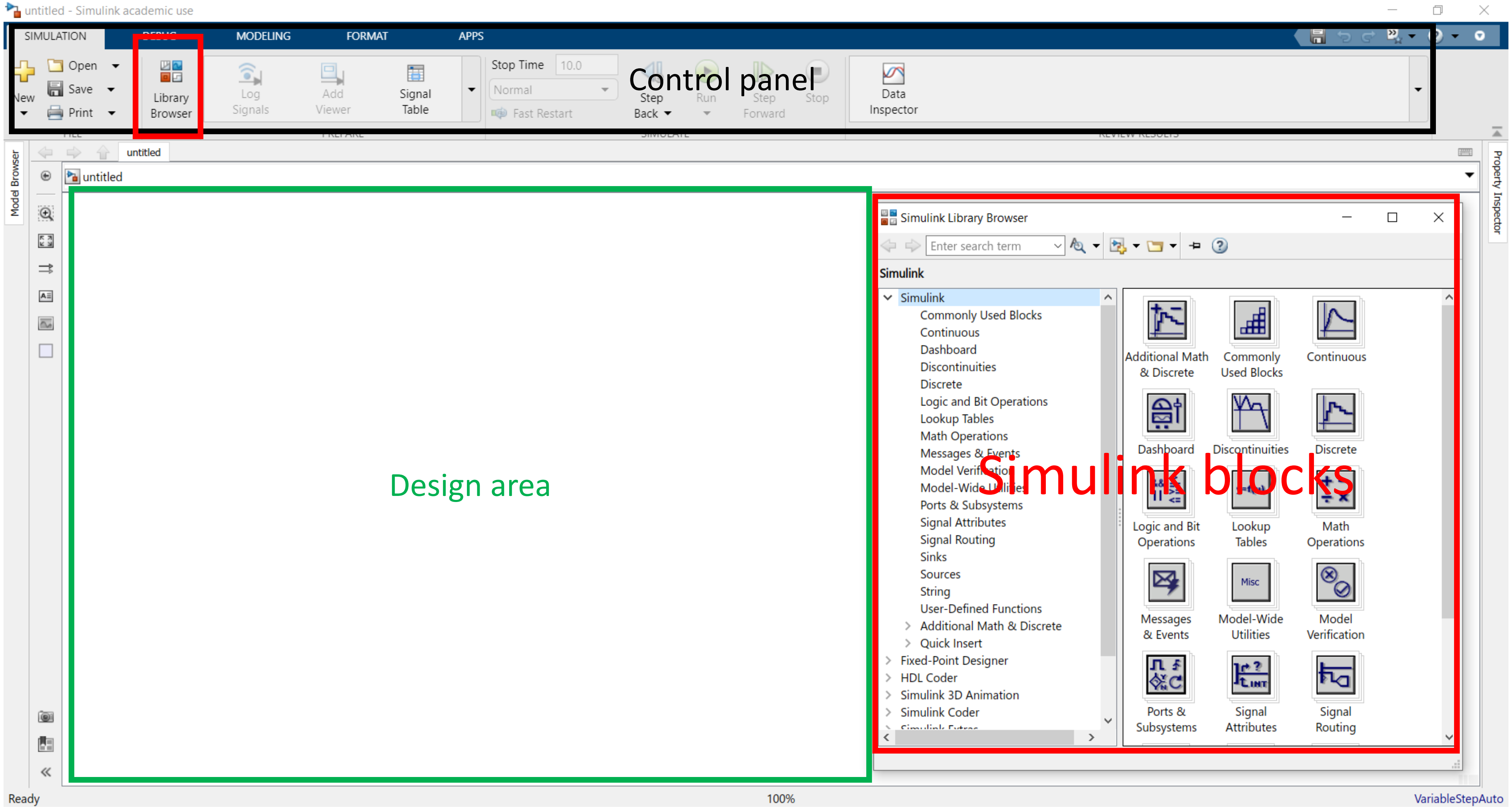
- Stand-alone applications
- Data analysis and figure generation
- Functions and algorithms (models, control algorithms, …)

SIMULINK®

Simulink is a block diagram environment for multidomain simulation and Model-Based Design

Why MATLAB is not enough?

- Useful to simulate complex systems composed by interconnected logical blocks
- Develop algorithms or model in a graphical language
- Make use of other tools and Add-on (as ROS Toolbox)



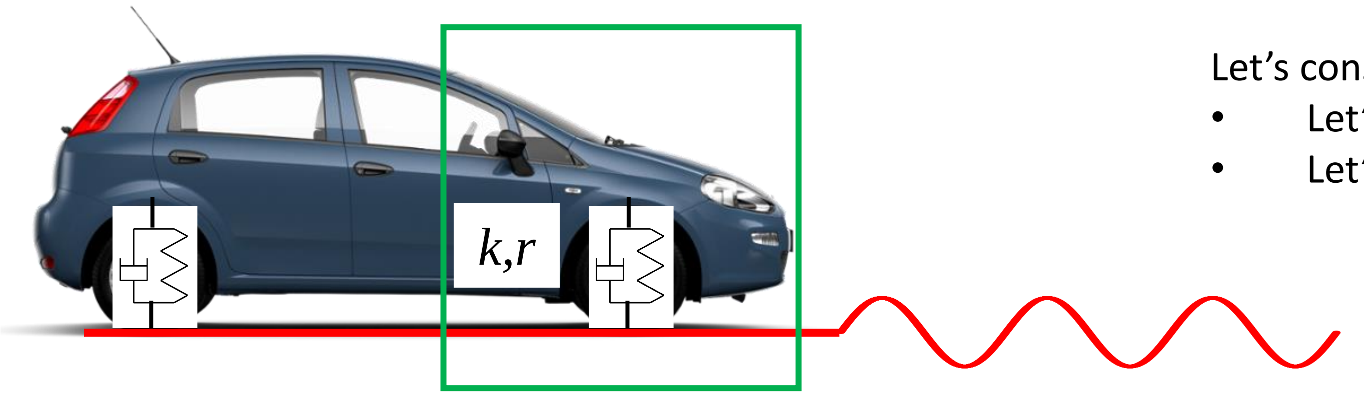
untitled - Simulink academic use
SIMULATION
MODELING
FORMAT
APPS
Open
Save
Print
New
Library Browser
Log Signals
Add Viewer
Signal Table
Stop Time
10.0
Normal
Fast Restart
Control panel
Step Back
Run
Step Forward
Stop
Data Inspector
Model Browser
Property Inspector
untitled
Design area
Simulink Library Browser
Enter search term
Simulink
Commonly Used Blocks
Continuous
Dashboard
Discontinuities
Discrete
Logic and Bit Operations
Lookup Tables
Math Operations
Messages & Events
Ports & Subsystems
Signal Attributes
Signal Routing
Sinks
Sources
String
User-Defined Functions
Additional Math & Discrete
Quick Insert
Fixed-Point Designer
HDL Coder
Simulink 3D Animation
Simulink Coder
Simulink blocks
Additional Math & Discrete
Commonly Used Blocks
Continuous
Dashboard
Discontinuities
Discrete
Logic and Bit Operations
Lookup Tables
Math Operations
Messages & Events
Model-Wide Utilities
Model Verification
Ports & Subsystems
Signal Attributes
Signal Routing
Ready
100%
VariableStepAuto

We will use it for:

- Simulate time evolution of our system
- Incrementally incorporate modeling blocks (models, control algorithms, environment, …)
- Design simple algorithms by graphical language
- Connect to our Robot (virtual – real)

Example:
vehicular suspension

GOAL: simulate and plot time evolution of vehicular suspension



Asphalt irregularities (sine wave)

Let's consider ¼ car:

- Let's write mechanical equation in diagram language
- Let's plot the quantities

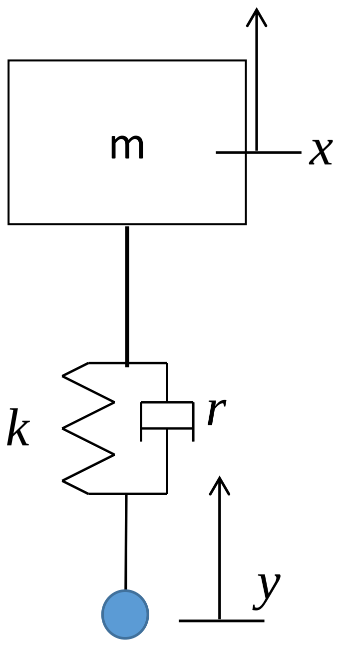
Let's define:

- Mechanical equation of the system
- **m,k,r** parameters of the model and external forces from data
- initial conditions of the system

M = 2000 **kg**
Δl_{st} = **0,1 m**
h $\in [0.5; 1]$

λ = **5 m**
v = **50** km/h

y0 = 0,1 m
$v0$ = 0 m/s
$x0$ = 0 m



Mechanical equations

Dyn equation

$$m\ddot{x} + mg + F_{el} + F_{dis} = 0$$

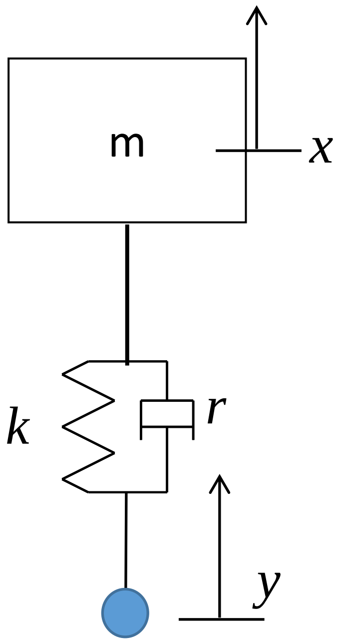
$$F_{el} = k(x - y) \qquad\qquad F_{dis} = r(\dot{x} - \dot{y})$$

$$\Rightarrow \quad m\ddot{x} + r\dot{x} + kx + mg = ky + k\dot{y} \qquad\qquad y(t) = y_0 \sin(\Omega t)$$

Static equilibrium position

$$x_0 = -\frac{mg}{k} \qquad\qquad x = x_0 + \tilde{x}$$

$$m\ddot{\tilde{x}} + r\dot{\tilde{x}} + k\tilde{x} = r\dot{y} + ky \quad \Rightarrow \quad \boxed{\ddot{\tilde{x}} + \frac{r}{m}\dot{\tilde{x}} + \frac{k}{m}\tilde{x} = \frac{r}{m}\dot{y} + \frac{k}{m}y}$$



Model parameters and initial conditions

```
%% define quantities
M = 2000; % [kg]
m = M/4; % [kg] 1/4 mass                          → m
v = 50/3.6; % [m/s] vehicle's speed
% define k based on max Dl static allowed (x0)
Dl_max = 0.1;
k = m*9.81/Dl_max;                                → k
w = sqrt(k/m); % natural frequency
% define r
h = 0.5; % [0.5 - 1]
r = h*2*m*w;                                      → r

%% define road roughness
lambda = 5; %[m]
Omega = 2*pi/lambda*v; % roughness frequency      → Ω

%% define initial conditions
v0 = 0; % [m]
x0 = 0; % [m]
```

Mechanical equations

$$\ddot{\tilde{x}} = -\frac{r}{m}\dot{\tilde{x}} - \frac{k}{m}\tilde{x} + \frac{r}{m}\dot{y} + \frac{k}{m}y$$

A *II Differential equation*

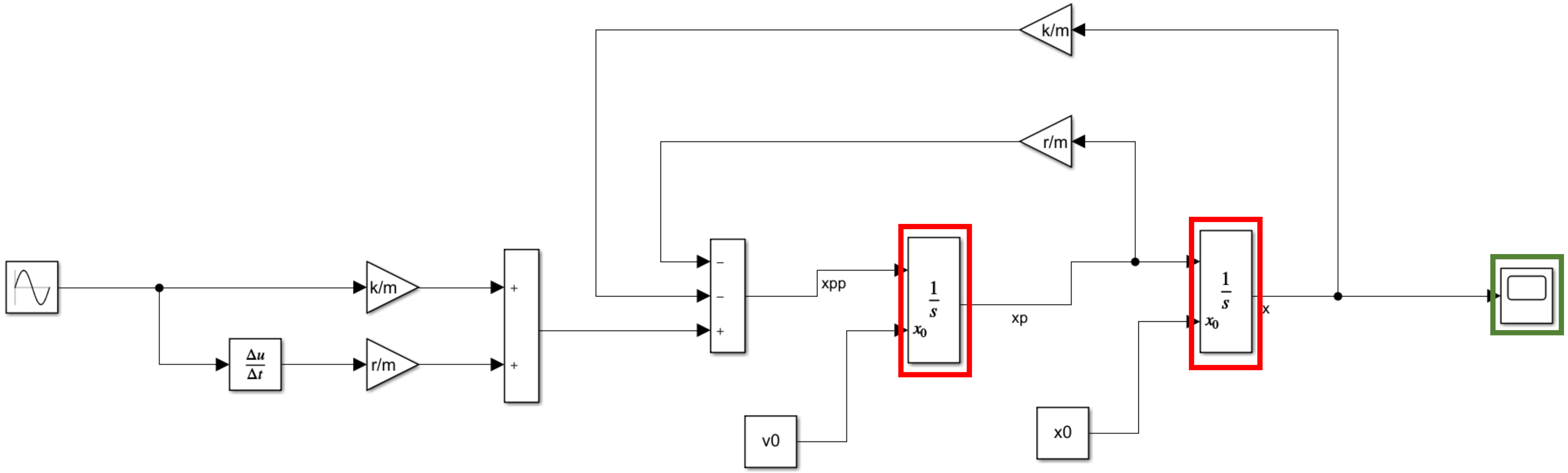
$$\begin{cases} \dot{z}_2 = -\frac{r}{m}z_2 - \frac{k}{m}z_1 + \frac{r}{m}\dot{y} + \frac{k}{m}y \\ \dot{z}_1 = z_2 \end{cases} \quad \begin{cases} y = u_1 \\ \dot{y} = u_2 \end{cases}$$

$$\Rightarrow \quad \dot{\bar{z}} = f(\bar{z}, \bar{u})$$

B *State space*

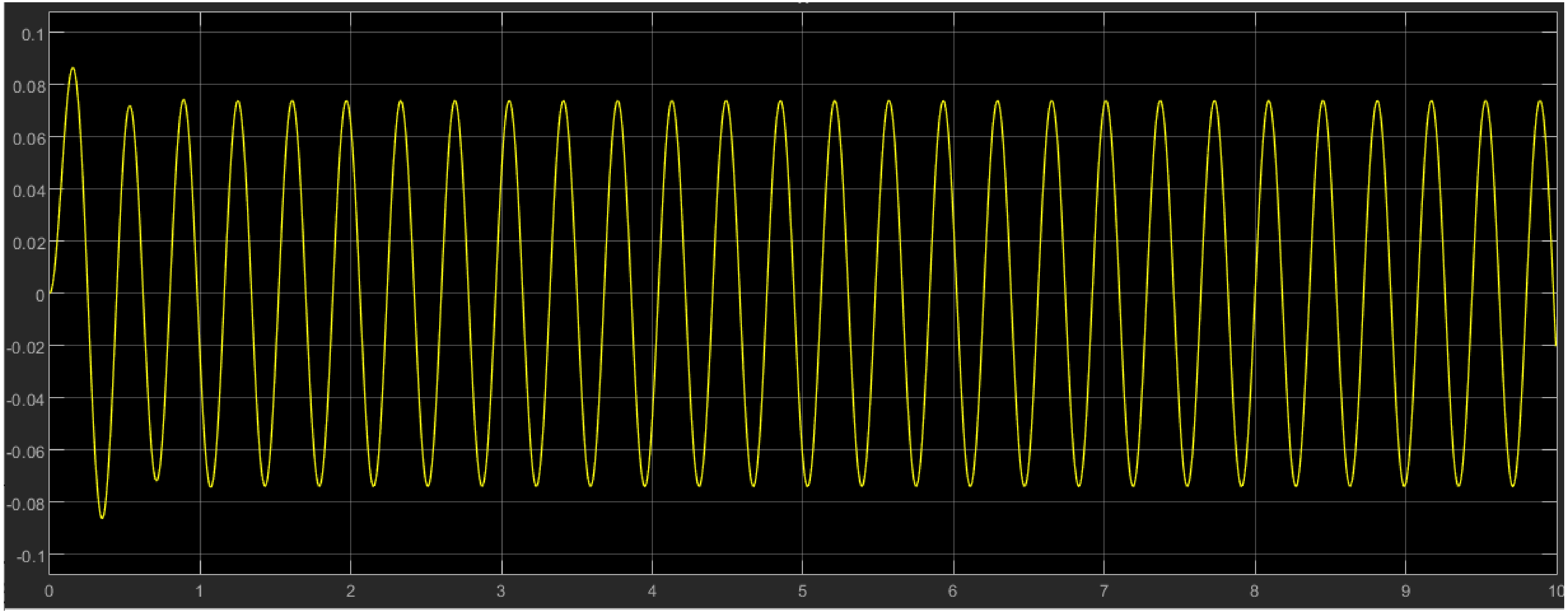
Simulation of the system

CASE A: use of Simulink



Simulation of the system

CASE A: use of Simulink



Simulation of the system

CASE B: use of MATLAB

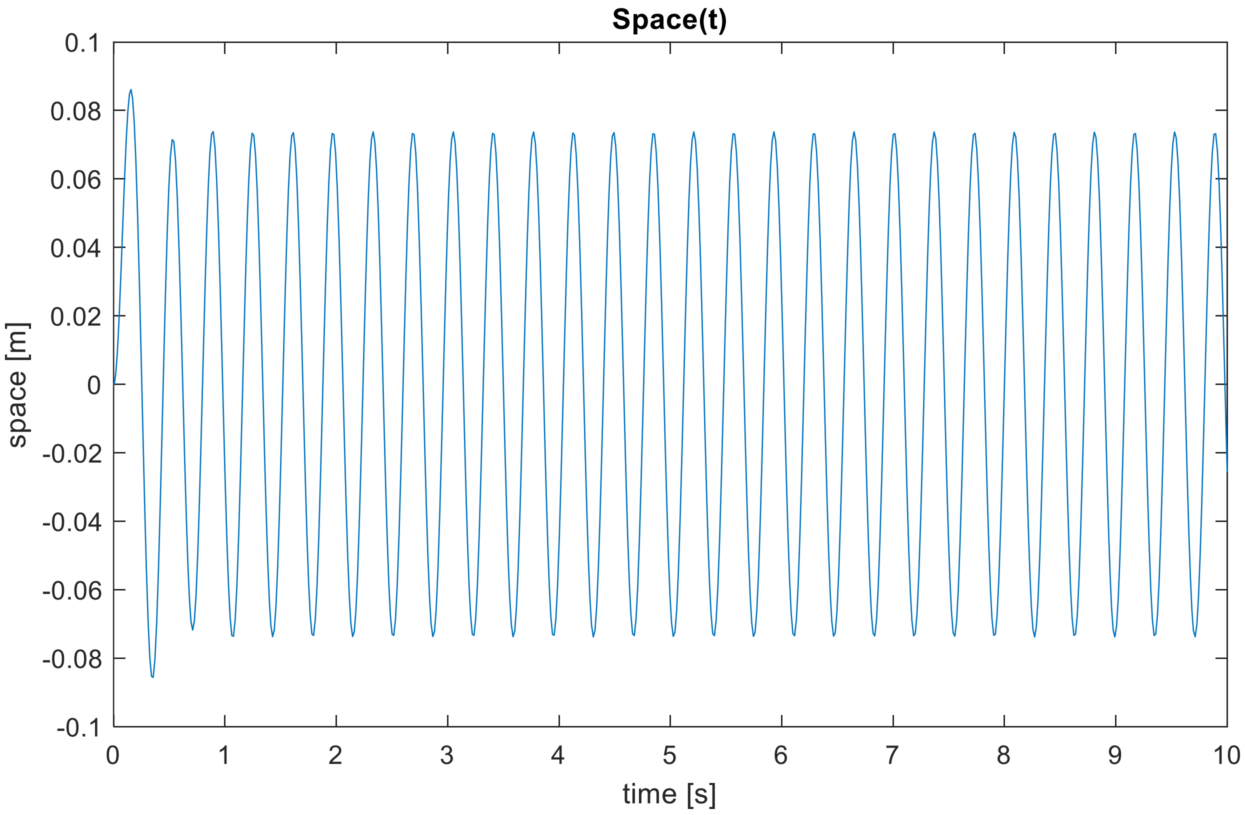
```
%% initialize data (load)
vibration_init_data;

%% define iteration loop
t0=0;
tf=10;
tspan = [t0 tf]; % time interval

z0=[0;0]; % init cond space (z1) - vel(z2)

[t,z] = ode45('veh_model',tspan,z0);

%% plot graphs

figure()
plot(t,z(:,1))
xlabel('time [s]')
ylabel('space [m]')
title('Space(t)')
```

Simulation of the system

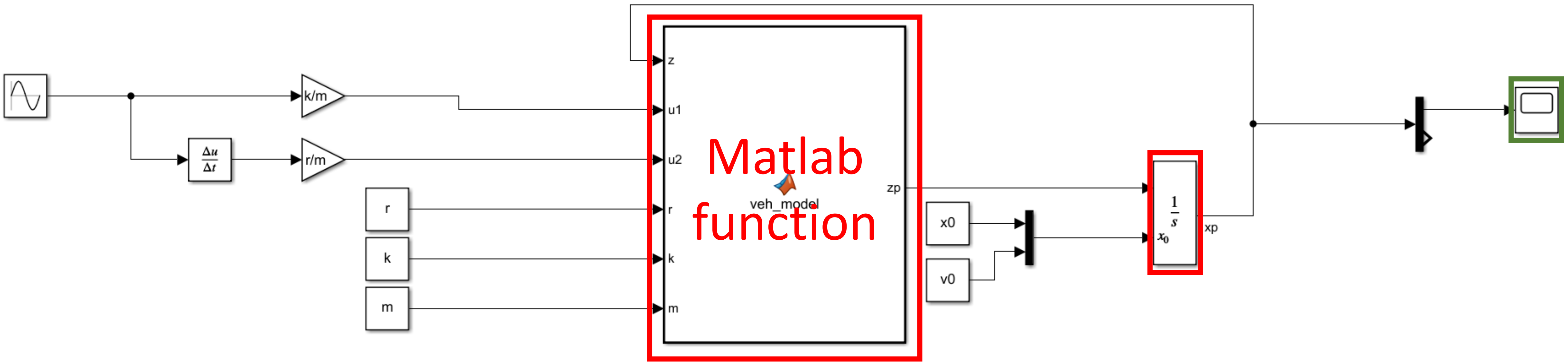
CASE B: use of MATLAB

```
function [zp] = veh_model(t,z)
global system force
zp=zeros(2,1);
u(1)= force.y0*sin(force.Omega*t);
u(2)= force.y0*force.Omega*cos(force.Omega*t);

zp(1) = z(2);
zp(2) = -system.r/system.m*z(2)-system.k/system.m*z(1)+...
    system.r/system.m*u(2)+system.k/system.m*u(1);
```

CASE B: use of MATLAB



CASE A+B: use of Simulink



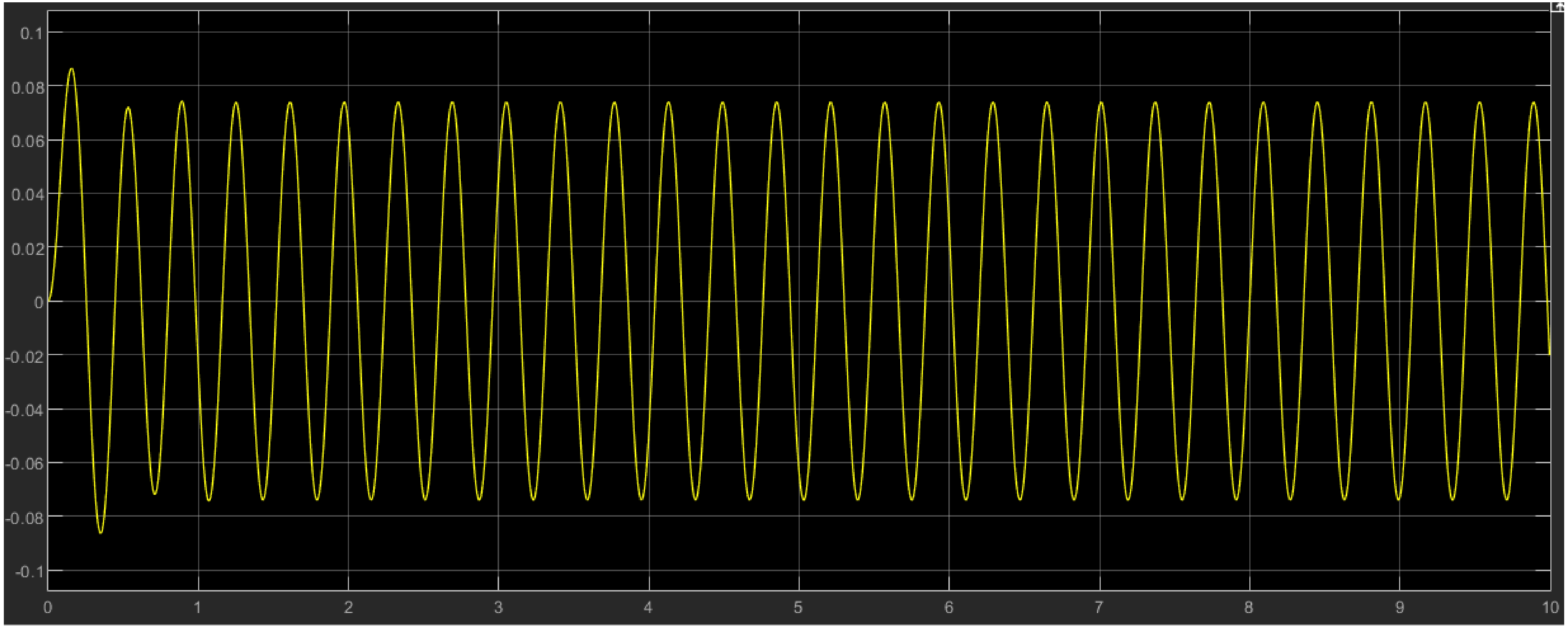
Simulation of the system

CASE A+B: use of Simulink

```
%====================================================================%
%                                                                    %
%   vehicle model for simulink                                       %
%                                                                    %
%====================================================================%

function zp  = veh_model(z,u1,u2,r,k,m)
zp=zeros(2,1);



%u(1)= force.y0*sin(force.Omega*t);
%u(2)= force.y0*force.Omega*cos(force.Omega*t);
zp(1) = z(2);
zp(2) = -r/m*z(2)-k/m*z(1)+u1+u2;
```

Simulation of the system

CASE A+B: use of Simulink



That’s it for today…

See you next time!

S. Arrigoni

POLITECNICO
MILANO 1863