Vehicle Dynamics Simulation Environment
(VDSE)

Christos Patrinos

Georges Sayes

Efstathios Velenis

Efstathios Siampis

2021

Contents

1. Introduction

VDSE (Vehicle Dynamics Simulation Environment) is a tool that predicts the acceleration, braking, cornering and ride behaviour of four-wheeled vehicles with independent suspension. The tool includes a 14 Degrees of Freedom (DoF) sprung/unsprung vehicle model, including three different tyre formulations, a driver model, powertrain model, brakes model and road input. The tool is developed under MATLAB and Simulink and aimed at being user friendly and modular by allowing the user to easily modify or change any of the provided equations and subsystems. Furthermore, the inherent modularity of the model makes it ideal for testing active systems (such as active aerodynamics, 4WS, torque vectoring, ESP, ABS etc), implementing different suspension models (air springs, adaptive damping, hydropneumatic suspension etc), implementing different powertrain models (Hybrid, electric, fuel cell etc) and more. This tool has been created with the goal for it to be fast to run making it suitable to be used as a plant for controller design and running design of experiments or optimisation of a specific subsystem or even the whole vehicle.

2. Vehicle modelling

The vehicle model is composed of 14DoF, six of which are for the chassis and eight for the wheels (2 per wheel). The degrees of freedom of the chassis are longitudinal, lateral vertical, roll, pitch and yaw. The degrees of freedom of the wheels are vertical and wheel spin.

The chassis is considered to be a single solid part with the forces being applied at its centre of gravity. Each wheel is also described as a single solid part linked to the ground through the tyre and to the chassis through the suspension elements. Three different tyre formulations are also available and provide different complexity and fidelity levels.

The suspension kinematics are limited to the definition of a pitch centre, a front and a rear roll centre. All other wheel angles such as camber and toe are not defined. Both roll centres are located on the same lateral position as the centre of gravity and the same longitudinal position as the front and rear axles. Their height is defined by the user. The pitch centre is located at the same longitudinal and lateral position as the centre of gravity and the height is defined by the user. Note that the steering affects the wheel toe angle through a constant steering ratio for the front and rear wheels.

The simple powertrain model includes an engine map (torque vs rpm vs throttle), gear ratios and a simple gear change logic that switch up gear at maximum power and switch down at maximum torque (Clutch has not been modelled). The power is sent to the wheels through three differentials. The front and rear and central differentials are always open, and the central differential can send the power to the front wheels, rear wheels or anything in between. A simple braking model is also provided, which translates the brake pedal position into pressure and then to braking torque through two parameters defined by the user.

3. Abbreviations

m sprung mass [kg]

m_F half front unsprung mass [kg]

m_R half rear unsprung mass [kg]

J_x roll inertia [kg.m^2]

J_y pitch inertia [kg.m^2]

J_z yaw inertia [kg.m^2]

J_{Fx} front wheels spin inertia [kg.m^2]

J_{Rx} rear wheels spin inertia [kg.m^2]

l wheelbase [m]

l_F distance of the front axle to the centre of gravity [m]

l_R distance of the rear axle to the centre of gravity [m]

w_F front track width [m]

w_R rear track width [m]

h_{cog} height of the centre of gravity of sprung mass [m]

h_{uFcog} height of the centre of gravity of front unsprung mass [m]

h_{uRcog} height of the centre of gravity of rear unsprung mass [m]

h_{pc} height of pitch centre [m]

h_{rcF} height of the roll centre at the front axle [m]

h_{rcR} height of the roll centre at the rear axle [m]

h_{rcg} height of the roll centre at the centre of gravity of the sprung mass [m]

r_F unloaded front tyre radius [m]

r_R unloaded rear tyre radius [m]

k_F front spring rate [N/m]

k_R rear spring rate [N/m]

k_{RF} front antiroll bar rate [N/m]

k_{RR} rear antiroll bar rate [N/m]

c_F front damper rate [Ns/m]

c_R rear damper rate [Ns/m]

k_{tF} radial stiffness of front tyres [N/m]

k_{tR} radial stiffness of rear tyres [N/m]

c_{tF} radial damping of front tyres [Ns/m]

c_{tR} radial damping of rear tyres [Ns/m]

c_{lF} front aerodynamic lift coefficient [-]

c_{lR} rear aerodynamic lift coefficient [-]

c_d aerodynamic drag coefficient [-]

A vehicle frontal area [m^2]

ρ air density [kg/m^3]

δ_{ij} wheel toe angle [rad]

V_x $V_y V_z$ longitudinal, lateral and vertical velocity of the CoG [m/s]

φ θ ψ roll, pitch and yaw angles [rad]

F_{ijzu} suspension vertical force [N]

F_{ijz} tyre vertical force [N]

ω_{ij} wheel spin rate [rad/s]

T_{ij} torque applied on the ij tyre [Nm]

V_{ijz} sprung mass velocity [m/s]

s_{ijx} s_{ijy} s_{ij} longitudinal, lateral slip and total slip ratios [-]

μ_{ijx} μ_{ijy} longitudinal and lateral tyre friction coefficients [-]

F_{ijtx} F_{ijty} longitudinal and lateral tyre forces in the wheel frame [N]

F_{ijx} F_{ijy} longitudinal and lateral tyre forces in the vehicle frame [N]

M_{ijtx} M_{ijty} M_{ijtz} x, y, z tyre moments in the tyre frame [Nm]

y_{if} Road vertical excitation at the wheels of the vehicle [m]

i Front (F) or Rear (R)

j Right (R) or Left (L)

4. Equations of Motion

The flow of the modelling starts from the driver and the road inputs. Based on the current vehicle states the driver will give throttle and brake input as well as a steering input. The throttle and brake input comes from a Proportional-Integral-Derivative (PID) controller that maintains the speed of the vehicle to a desired value. The steering input comes from an open-loop steering command that can be chosen by the user. The road input can also be chosen by the user. The vehicle model proposed below is partly based on the works of [1], [2] and [3].

Using the road inputs and the current vehicles states the unsprung masses equations are calculated as follows.

$$F_{ijzu} = -c_f\left(Vz - V_{ijz} + \frac{w_i}{2}\cdot\dot{\varphi} - l_i\cdot\dot{\theta}\right) - k_i\left(z - z_{ij} + \frac{w_i}{2}\cdot\varphi - l_i\cdot\theta\right)$$

$$F_{ijz} = -c_{ti}(\dot{z}_{ij} - \dot{y}_{ij}) - k_{ti}(z_{ij} - y_{ij})$$

$$m\cdot\dot{V}_{ijz} = F_{ijz} - F_{ijzu} + kR_i\left(\varphi - \frac{z_{ij} - z_{ij}}{w_i}\right) - m_i\cdot g$$

The load transfer reacted through the suspension elements is modelled within the roll and pitch equations and the load transfer term that is not reacted by the suspension and transferred through the rigid links is calculated as follows:

$$\begin{aligned}F_{FLz_{overall}} = F_{FLz} &- \left(\frac{2\cdot m_F\cdot h_{uFcog}}{w_F} + \frac{m\cdot h_{rcF}}{w_F}\cdot\frac{l_R}{l_F + l_R}\right)\cdot\left(\dot{v}_y + v_x\cdot\dot{\psi} - v_z\cdot\dot{\varphi}\right)\\ &- \left(\frac{m_F\cdot h_{uFcog} + m_R\cdot h_{uRcog}}{l_F + l_R} + \frac{m\cdot h_{pc}}{l_F + l_R}\cdot\frac{1}{2}\right)\cdot\left(\dot{v}_x + v_z\cdot\dot{\theta} - v_y\cdot\dot{\psi}\right)\end{aligned}$$

$$\begin{aligned}F_{FRz_{overall}} = F_{FRz} &+ \left(\frac{2\cdot m_F\cdot h_{uFcog}}{w_F} + \frac{m\cdot h_{rcF}}{w_F}\cdot\frac{l_R}{l_F + l_R}\right)\cdot\left(\dot{v}_y + v_x\cdot\dot{\psi} - v_z\cdot\dot{\varphi}\right)\\ &- \left(\frac{m_F\cdot h_{uFcog} + m_R\cdot h_{uRcog}}{l_F + l_R} + \frac{m\cdot h_{pc}}{l_F + l_R}\cdot\frac{1}{2}\right)\cdot\left(\dot{v}_x + v_z\cdot\dot{\theta} - v_y\cdot\dot{\psi}\right)\end{aligned}$$

$$\begin{aligned}F_{RLz_{overall}} = F_{RLz} &- \left(\frac{2\cdot m_R\cdot h_{uRcog}}{w_R} + \frac{m\cdot h_{rcR}}{w_R}\cdot\frac{l_F}{l_F + l_R}\right)\cdot\left(\dot{v}_y + v_x\cdot\dot{\psi} - v_z\cdot\dot{\varphi}\right)\\ &+ \left(\frac{m_F\cdot h_{uFcog} + m_R\cdot h_{uRcog}}{l_F + l_R} + \frac{m\cdot h_{pc}}{l_F + l_R}\cdot\frac{1}{2}\right)\cdot\left(\dot{v}_x + v_z\cdot\dot{\theta} - v_y\cdot\dot{\psi}\right)\end{aligned}$$

$$\begin{aligned}F_{RRz_{overall}} = F_{RRz} &+ \left(\frac{2\cdot m_R\cdot h_{uRcog}}{w_R} + \frac{m\cdot h_{rcR}}{w_R}\cdot\frac{l_F}{l_F + l_R}\right)\cdot\left(\dot{v}_y + v_x\cdot\dot{\psi} - v_z\cdot\dot{\varphi}\right)\\ &+ \left(\frac{m_F\cdot h_{uFcog} + m_R\cdot h_{uRcog}}{l_F + l_R} + \frac{m\cdot h_{pc}}{l_F + l_R}\cdot\frac{1}{2}\right)\cdot\left(\dot{v}_x + v_z\cdot\dot{\theta} - v_y\cdot\dot{\psi}\right)\end{aligned}$$

A function is added to make sure the forces don't go negative when tyre lift occurs.

Following that the vehicle slip angle is calculated based on the current vehicle states.

$$\beta = \tan^{-1}\frac{V_y}{V_x}$$

$$V = \sqrt{{V_x}^2 + {V_y}^2}$$

Using the following equations, the wheels longitudinal and lateral velocities are calculated.

$$l_{FL} = \sqrt{{l_F}^2 + (\frac{w_F}{2})^2} \quad l_{FR} = \sqrt{{l_F}^2 + (\frac{w_F}{2})^2} \quad l_{RL} = \sqrt{{l_R}^2 + (\frac{w_R}{2})^2} \quad l_{RR} = \sqrt{{l_R}^2 + (\frac{w_R}{2})^2}$$

$$\gamma_{FL} = \tan^{-1}\frac{w_F/_2}{l_F} \qquad \gamma_{FR} = \tan^{-1}\frac{w_F/_2}{l_F} \qquad \gamma_{RL} = \tan^{-1}\frac{w_R/_2}{l_R} \qquad \gamma_{RR} = \tan^{-1}\frac{w_R/_2}{l_R}$$

$$V = \sqrt{{v_x}^2 + {v_y}^2}$$

$$V_{FLx} = V\cos(\delta_{FL} - \beta) - \dot{\psi} \cdot l_{FL} \cdot \sin(\gamma_{FL} - \delta_{FL})$$

$$V_{FLy} = -V\sin(\delta_{FL} - \beta) + \dot{\psi} \cdot l_{FL} \cdot \cos(\gamma_{FL} - \delta_{FL})$$

$$V_{FRx} = V\cos(\delta_{FR} - \beta) + \dot{\psi} \cdot l_{FR} \cdot \sin(\gamma_{FR} + \delta_{FR})$$

$$V_{FRy} = -V\sin(\delta_{FR} - \beta) + \dot{\psi} \cdot l_{FR} \cdot \cos(\gamma_{FR} + \delta_{FR})$$

$$V_{RLx} = V\cos(\delta_{RL} - \beta) - \dot{\psi} \cdot l_{RL} \cdot \sin(\gamma_{RL} - \delta_{RL})$$

$$V_{RLy} = -V\sin(\delta_{FL} - \beta) - \dot{\psi} \cdot l_{RL} \cdot \cos(\gamma_{RL} - \delta_{RL})$$

$$V_{RRx} = V\cos(\delta_{RR} - \beta) + \dot{\psi} \cdot l_{RR} \cdot \sin(\gamma_{RR} + \delta_{RR})$$

$$V_{RRy} = -V\sin(\delta_{RR} - \beta) - \dot{\psi} \cdot l_{RR} \cdot \cos(\gamma_{RR} + \delta_{RR})$$

The wheel dynamics are then calculated using the following equations:

$$J_{ix} \cdot \dot{\omega}_{ij} = T_{ij} - F_{ijtx} \cdot r_i$$

The wheel rotational velocity, wheel longitudinal and lateral translational velocities as well as the overall normal load applied on the tires is fed into the tyre model calculation which will give us as an output the forces and moments. For more details on the tyre, calculation refer to the tyre modelling section.

The assumption for the longitudinal, lateral and yaw dynamics is that the vehicle (sprung and unsprung mass) behaves as a single unit of mass $(m + 2 \cdot m_F + 2 \cdot m_R)$ with its centre of gravity located at the overall centre of mass of the system.

$$l_{F_{overall}} = \frac{2 \cdot m_F \cdot 0 + 2 \cdot m_R \cdot l + m \cdot l_F}{m_F + m_R + m}$$

$$l_{R_{overall}} = l - l_{F_{overall}}$$

The tyre forces are projected on the vehicle frame as follows:

$$F_{xij} = F_{txij} \cdot cos\,\delta_{ij} - F_{tyij} \cdot sin\,\delta_{ij}$$
$$F_{yij} = F_{tyij} \cdot cos\,\delta_{ij} + F_{txij} \cdot sin\,\delta_{ij}$$

Using the tyre forces and moments and the suspension forces the chassis equations are calculated as follows:

Longitudinal Dynamics

$$
\begin{aligned}
(m + 2 \cdot m_F + 2 \cdot m_R) \cdot \left(\dot{v}_x + v_z \cdot \dot{\theta} - v_y \cdot \dot{\psi}\right) \\
&= (F_{xFL} + F_{xFR} + F_{xRL} + F_{xRR}) - (0.5 \cdot \rho \cdot {v_x}^2 \cdot c_d \cdot A) \\
&+ \frac{M_{zFL} \cdot \cos\delta_{FL} + M_{zFR} \cdot \cos\delta_{FR}}{r_F} + (M_{zRL} \cdot \cos\delta_{RL} + M_{zRR} \cdot \cos\delta_{RR})/r_R
\end{aligned}
$$

Lateral Dynamics:

$$
\begin{aligned}
(m + 2 \cdot m_F + 2 \cdot m_R) \cdot \left(\dot{v}_y + v_x \cdot \dot{\psi} - v_z \cdot \dot{\varphi}\right) \\
&= \left(F_{yFL} + F_{yFR} + F_{yRL} + F_{yRR}\right) + (M_{zFL} \cdot \sin\delta_{FL} + M_{zFR} \cdot \sin\delta_{FR})/r_F \\
&+ (M_{zRL} \cdot \sin\delta_{RL} + M_{zRR} \cdot \sin\delta_{RR})/r_R
\end{aligned}
$$

Yaw Dynamics:

$$
\begin{aligned}
J_z \ddot{\psi} = \frac{w_R}{2}(-F_{xFL} + F_{xFR}) + \frac{w_R}{2}(-F_{xRL} + F_{xRR}) + l_{F_{overall}}\left(F_{yFL} + F_{yFR}\right) \\
+ l_{R_{overall}}\left(-F_{yRL} - F_{yRR}\right) - \left(m_r \cdot l_{R_{overall}} - m_f \cdot l_{F_{overall}}\right) \\
\cdot \left(\dot{v}_y + v_x \cdot \dot{\psi} - v_z \cdot \dot{\varphi}\right) + M_{zFL} + M_{zFR} + M_{zRL} + M_{zRR}
\end{aligned}
$$

For vertical, pitch and yaw dynamics sprung mass and unsprung mass are separated:

The downforce is assumed to be applied on the front and rear of chassis.

Vertical Dynamics:

$$
\begin{aligned}
m\left(\dot{v}_z + v_y \cdot \dot{\varphi} - v_x \cdot \dot{\theta}\right) \\
&= (F_{zuFL} + F_{zuFR} + F_{zuRL} + F_{zuRR}) - m \cdot g + (0.5 \cdot \rho \cdot {v_x}^2 \cdot c_{lF} \cdot A) + (0.5 \cdot \rho \\
&\cdot {v_x}^2 \cdot c_{lR} \cdot A)
\end{aligned}
$$

Roll center height at the CoG:

$$h_{rcg} = (h_{rcF} \cdot l_R + h_{rcR} \cdot l_F)/(l_F + l_R)$$

Roll Dynamics:

$$
\begin{aligned}
(J_x + m \cdot \left(h_{cg} - h_{rcg}\right)^2)\ddot{\varphi} \\
&= -\frac{w_F}{2} \cdot F_{zuFL} + \frac{w_F}{2} \cdot F_{zuFR} - \frac{w_F}{2} \cdot F_{zuRL} + \frac{w_F}{2} \cdot F_{zuRR} \\
&+ kR_F\left(\varphi - \frac{Z_{FL} - Z_{FR}}{\omega_F}\right) + kR_R\left(\varphi - \frac{Z_{RL} - Z_{RR}}{\omega_R}\right) + m \cdot a_y \cdot cos\,\varphi \\
&\cdot \left(h_{cg} - h_{rcg}\right) + m \cdot g \cdot sin\,\varphi \cdot \left(h_{cg} - h_{rcg}\right)
\end{aligned}
$$

Pitch Dynamics:

$$
\left(J_y + m \cdot \left(h_{cg} - h_{pc}\right)^2\right)\ddot{\theta} = l_F \cdot F_{zuFL} + l_F \cdot F_{zuFR} - l_R F_{zuRL} - l_R F_{zuRL} + m \cdot a_x \cdot \cos\theta \cdot \left(h_{cg} - h_{pc}\right) + m \cdot g \cdot \sin\theta \cdot \left(h_{cg} - h_{Pc}\right) + \left(0.5 \cdot \rho \cdot {v_x}^2 \cdot c_{lF} \cdot A\right) \cdot l_R + \left(0.5 \cdot \rho \cdot {v_x}^2 \cdot c_{lR} \cdot A\right) \cdot l_R
$$

5. Tyre formulation

For the Tyre modelling, three different approaches are provided depending on the fidelity and complexity desired.

MFEval

As a first option, the library of MFEval is provided [4]. The library can be downloaded either as a toolbox or as a package in this link:

https://uk.mathworks.com/matlabcentral/fileexchange/63618-mfeval

More information regarding the development, installation and usage of MFeval can be found in this site:

https://mfeval.wordpress.com/

Pacejka 5.2

/!\ For this formulation initial vehicle velocity must be greater than 0

The second tyre formulation consists of a Pacejka 5.2 [5] formulation with the following assumptions:

- All scaling factors are equal to one
- No reduction coefficients due to turnslip
- No relaxation length
- Parameters are not affected by tyre velocity

Simplified Pacejka - BCD Formulation

/!\ For this formulation initial vehicle velocity must be greater than 0

The third tyre formulation provided is a simplified Pacejka representation, using only the B, C and D factors. In this formulation, we firstly calculate the longitudinal and lateral slips using the tyre thread velocities:

$$
s_{ijx} = \frac{\left(V_{ijx} - \omega_{ij} \cdot r_i\right)}{\left|\omega_{ij} \cdot r_i\right|}
$$

$$
s_{ijy} = \frac{\left(1 + s_{ijx}\right) \cdot V_{ijy}}{\left|V_{ijx}\right|}
$$

Based on the friction ellipse concept, the total slip can then be calculated:

$$s_{ij} = \sqrt{{s_{ijx}}^2 + {s_{ijy}}^2}$$

Thereafter, the longitudinal and lateral adhesion can be calculated:

$$\mu_{ijx} = D_{xi} \cdot \sin\left(C_{xi} \cdot \operatorname{atan}\left(B_{xi} \cdot s_{ij}\right)\right)$$

$$\mu_{ijy} = D_{yi} \cdot \sin\left(C_{yi} \cdot \operatorname{atan}\left(B_{yi} \cdot s_{ij}\right)\right)$$

Knowing all these, leads to the calculation of the tire forces:

$$F_{xij} = \frac{-s_{ijx}}{s_{ij}} \cdot F_{zij} \cdot \mu_{ijx}$$

$$F_{yij} = \frac{-s_{ijy}}{s_{ij}} \cdot F_{zij} \cdot \mu_{ijy}$$

6. Parameters, modes and structure of the model

The user can select parameters and enable/disable various features of the vehicle model depending on the desired analysis. This can be done from the "*vehicle_parameters_script.m*"; it is the script where all the vehicle parameters are defined.

The engine map found in the parameter script can be imported from an excel file that should follow the same structure as the provided excel sheet for the example engine map. The engine map data are filled with spline interpolation.

"*enableLongDynamics*" parameter set to 0 allows to bypass the vehicle longitudinal dynamics allowing to test the vehicle is pure cornering conditions.

"*enablePowertrain*" parameter set to 0 allows bypassing the powertrain of the vehicle which can be considered as the driver maintaining the clutch engaged through the manoeuvre.

"*enableDriverControl*" parameter set to 1 enables the controller to maintain a constant speed through the manoeuvre by actuating the throttle or brake.

Depending on the tire model that is selected different parameters are needed. If the selected model is MFEval, a .tir file (6.1 or 5.2) needs to be specified for the front and the rear tires. If the simplified Paceijka 5.2 model is selected either .tir file for the front or the rear tires can be selected, or the user can manually import the important parameters needed for the formulation. Finally, if the BCD formulation is selected, the user needs to specify the different parameters for the front and rear tyres.

The road profile block consists of a function that generates a single sinusoidal bump of height "*bump_height*", length "*bump_length*" and occurs at a time "*bump_time*". The bump will first appear in the front axle followed by the rear axle with a delay that depends on the wheelbase and the vehicle speed. The user can easily replace this block with other road profiles.

The driver block consists of the steering, throttle and braking command.

Throttle and braking commands come from a controller that aims at maintaining the desired speed.

For the steering, only open-loop steering manoeuvres are currently available. The user can choose from four steering profiles:

- A sine steer that starts at 3 seconds with a frequency of 1 rad/s and an amplitude of 30 degrees at the steering wheel.
- A slow ramp steer starting at 1 second with a slope of 10 deg/s the steering wheel
- A step steer starting at 15 seconds with a slope of 1400 deg/s with a maximum step of 700 deg at the steering wheel
- No steering for pure longitudinal or pure ride manoeuvres

Each manoeuvre can be selected by double-clicking on the manual switches.

Other custom manoeuvres can be added by the user.

Manual switches are also available on each wheel to define which wheels are steering and which are not.

The 14-DoF vehicle model block is the block that incorporates the different equations described in the previous section

Finally, the results block includes scopes to visualize the different vehicle states as well as blocks that saves all the simulation data to the Matlab workspace for post-processing.

7. Solver

It is recommended to use a fixed-step solver with a step size of 1 ms. All the available Simulink solvers are robust and work with the model, thus the auto option can be selected.

8. Blender Animation

Setting up Blender (Windows)

It is possible to animate the behaviour of the vehicle using the third-party software of Blender. Blender can be downloaded from the following link:

https://www.blender.org/download/

Firstly, run a simulation with the Simulink model. Note that you need to use a fixed step solver with a step size of 0.1 seconds for accurate results. It is advised however not to go above 0.01 seconds step size to have good results and a good animation.

After the end of the simulation at Simulink, you can generate an .xls file using the provided script *“Generate_XLS.m”.*

If you want to run an animation with multiple vehicles simultaneously, you can run a different scenario then run again the *“Generate_XLS.m”* by replacing the “*sheet*” variable from 1 to the corresponding vehicle (i.e. 2 for the second vehicle, 3 for the third vehicle, 4 for the fourth and so on). Currently, the maximum number of vehicles is 8. The fixed step size and the simulation end time should be the same to all the vehicles that will be in the same animation.

To run the Blender animation, you firstly have to import two python libraries: pandas and xlrd (need to be done only once after the downloading of Blender).

To add the libraries, open the downloaded Blender application. Go to the scripting tab (number 1 on Figure *1*) and write in the console (number 2 on Figure *1*) the following lines:

```
import sys
sys.exec_prefix
```

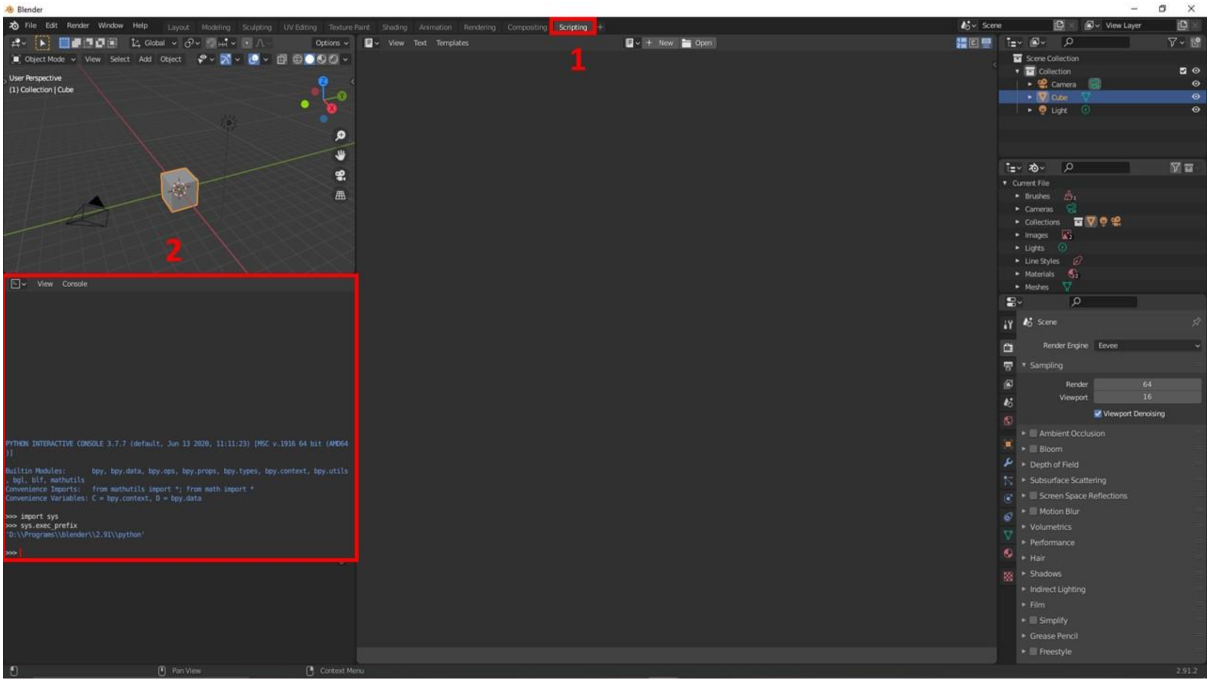

Figure 1: Setup of Blender (Windows)

A path similar to `'C:\\Programs\\blender\\2.91\\python'` should appear. Copy it.

Open widows PowerShell as an administrator (type PowerShell in windows search bar and right click, **run as an administrator**) as in the following Figure.

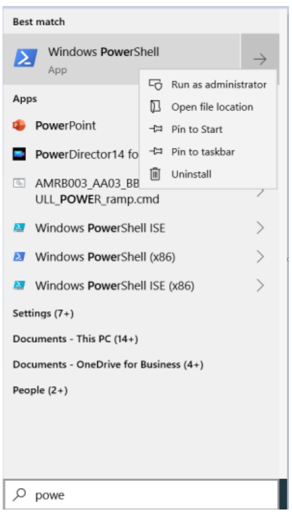

Figure 2: Windows Powershell

In the PowerShell, type cd ‘copied path from blender’. For example:

```
cd ‘C:\\Programs\\blender\\2.91\\python’
```

Then type the following to install pandas library:

```
.\bin\python.exe -m pip install pandas
```

Followed by this line to install xlrd library:

```
.\bin\python.exe -m pip install xlrd
```

You should be able see something similar to the following picture.

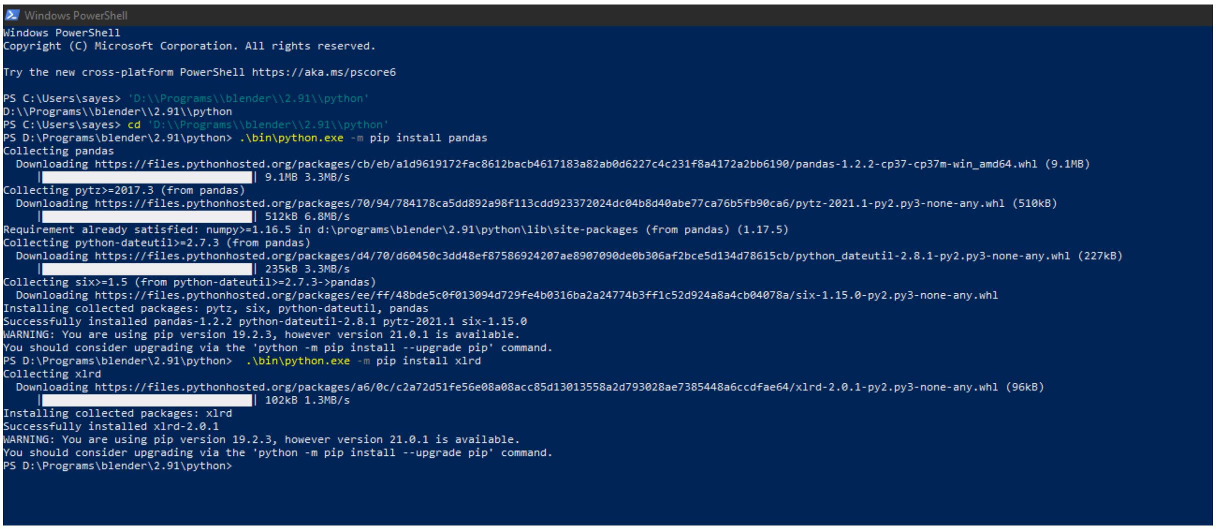

Figure 3: Installation of pandas and xlrd (Windows)

Close PowerShell and close blender.

Run an animation on Blender (Windows)

Double click on the *“VDSE_ANIMATION.blend”* file.

Go to line 16 in the script and change

```
‘C:\\Put\\your\\path\\here\\and\\use\\double\\backslashes.xls’
```

to the path of the generated xls file. In Figure 4, you can see the line and an example of how this path should look like.

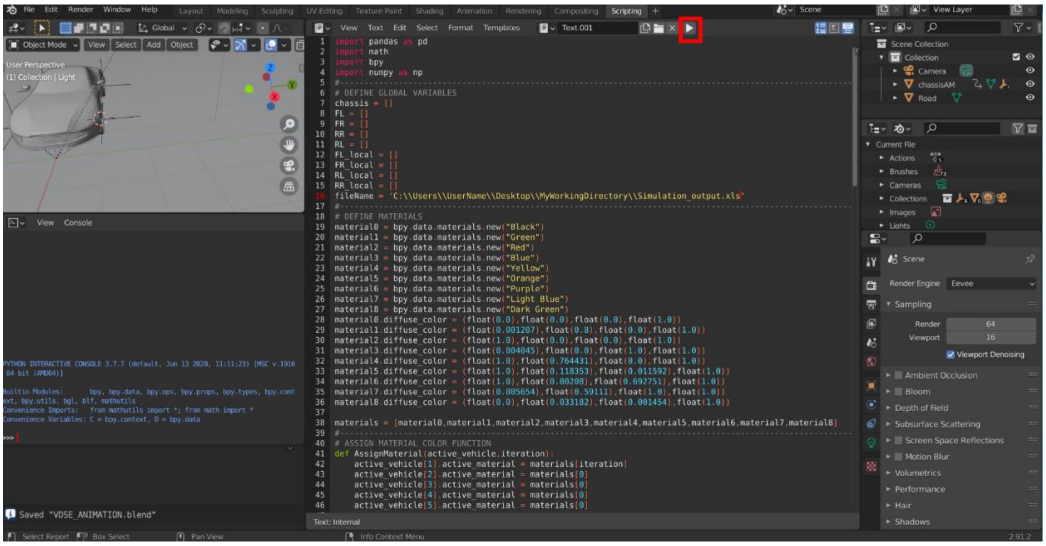

Figure 4: Change the filepath to the Simulation Ouput xls file and run animation (Windows)

Note that you should put either double backslashes in the path or single forward slashes (e.g. 'C:/User/ Desktop/ Simulation_output.xls').

Run the script by pressing the play button on top of the script (see *Figure* 4) or by using the keyboard shortcut *"Alt+P"*.

Go to the layout tab (number 1 in Figure *5*) and you can now visualise the animation. You can choose the render settings (number 2 in Figure *5*). If you have multiple vehicles in the same animation you can choose which vehicle to lock on (number 3 in Figure *5*) or you can choose not to lock on any vehicle by selecting "Camera". You also need to change the end frame of the simulation (number 4 in Figure *5*). This value should be equal to 10 times the end time set on Simulink. The frame rate (number 5 in Figure *5*) should be set to 10 to run the animation in real time. Finally, you can press the play button (number 6 in Figure *5*) to visualise the animation.

For multiple vehicles the colour order is the following: Green, red, blue, yellow, orange, purple, light blue, light green.

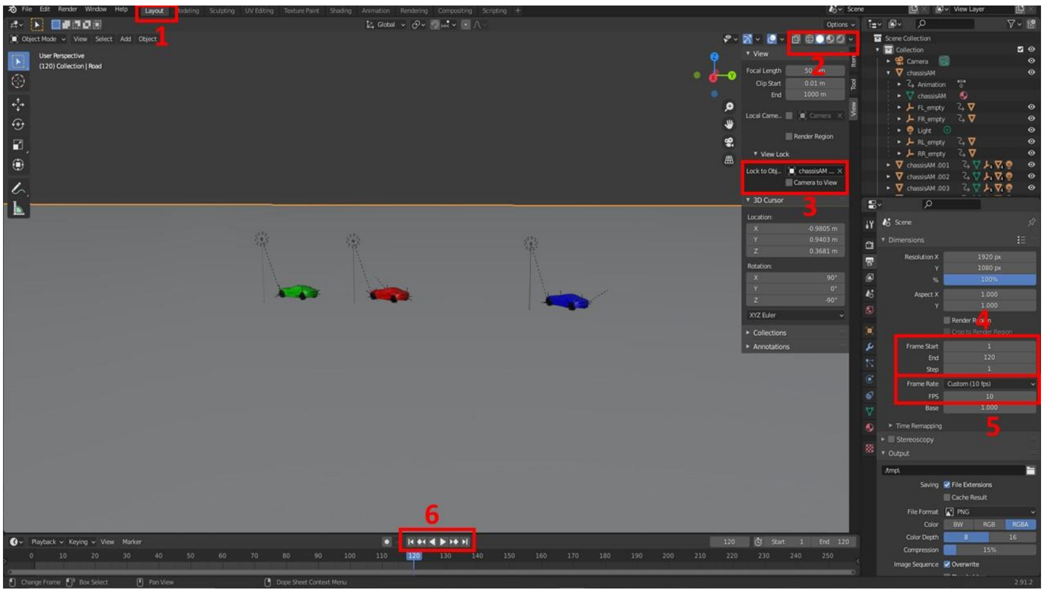

Figure 5: Blender UI (Windows)

Setting up Blender (MacOS)

Download blender from the following link:

https://www.blender.org/download/

Firstly, run a simulation with the Simulink model. Note that you need to use a fixed step solver with a step size of 0.1 seconds for accurate results. It is advised however not to go above 0.01 seconds step size to have good results and a good animation.

After the end of the simulation at Simulink, you can generate an .xls file using the provided script *"Generate_XLS.m"*.

If you want to run an animation with multiple vehicles simultaneously, you can run a different scenario then run again the *"Generate_XLS.m"* by replacing the "*sheet*" variable from 1 to the corresponding vehicle (i.e. 2 for the second vehicle, 3 for the third vehicle, 4 for the fourth and so on). Currently, the maximum number of vehicles is 8. The fixed step size and the simulation end time should be the same to all the vehicles that will be in the same animation.

To run the Blender animation, you firstly have to import two python libraries: pandas and xlrd (need to be done only once after the downloading of Blender).

To add the libraries, open the downloaded Blender application. Go to the scripting tab (number 1 on Figure *6*) and write in the console (number 2 on Figure *6*) the following lines:

```
import sys
sys.exec_prefix
```

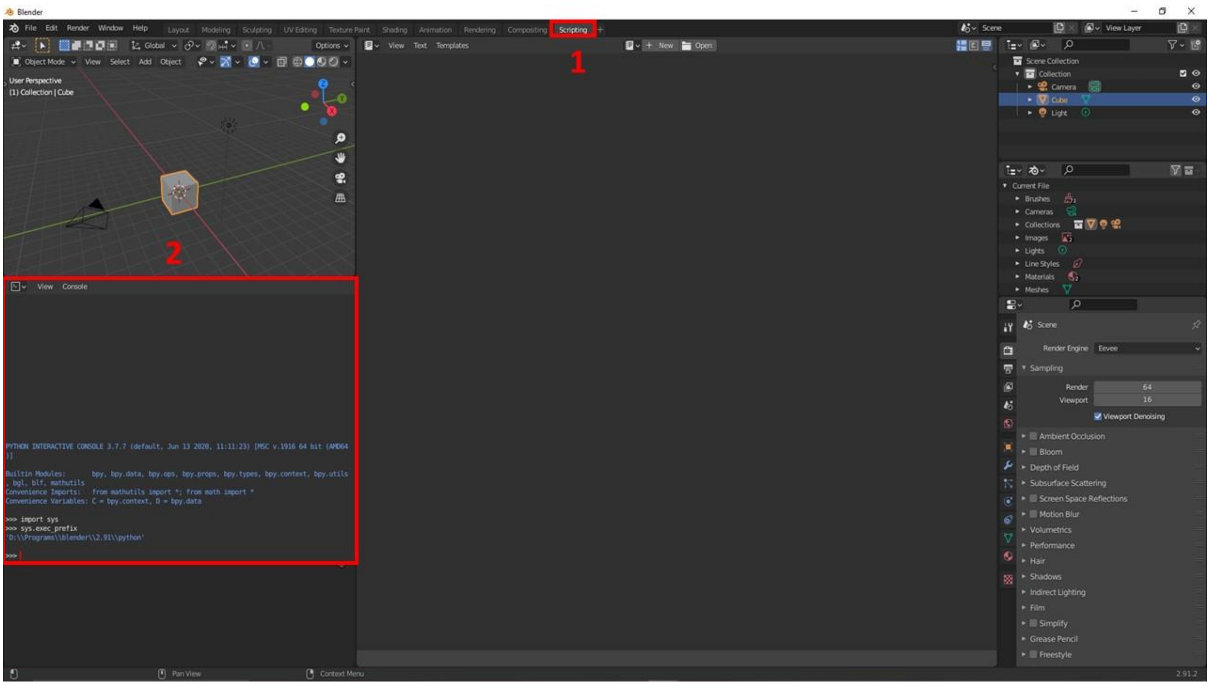

Figure 6: Setup of Blender (MacOS)

A path similar to:

```
'/Applications/Blender.app/Contents/Resources/2.92/python'
```

should appear. Copy it.

Open the terminal from MacOS and run the following commands (using the path that you just copied):

```
cd '/Applications/Blender.app/Contents/Resources/2.92/python'
```

Then type the following to install pandas library:

```
./bin/python3.7m -m pip install pandas
```

Followed by this line to install xlrd library:

```
./bin/python3.7m -m pip install xlrd
```

You should be able see something similar to the following picture.

```
MyMac:python georgiospatrinos$ ./bin/python3.7m -m pip install pandas
Collecting pandas
  Downloading https://files.pythonhosted.org/packages/2d/d2/6473460b1aad0944e8f0d7d618723f4b9d60485554eb1d42c5b193e716ec/pandas-1.2.2-cp37-cp37m-macosx_10_9_x86_64.whl (10.4MB)
     |████████████████████████████████| 10.4MB 5.6MB/s
Requirement already satisfied: numpy>=1.16.5 in ./lib/python3.7/site-packages (from pandas) (1.17.5)
Collecting python-dateutil>=2.7.3 (from pandas)
  Downloading https://files.pythonhosted.org/packages/d4/70/d60450c3dd48ef87586924207ae8907090de0b306af2bce5d134d78615cb/python_dateutil-2.8.1-py2.py3-none-any.whl (227kB)
     |████████████████████████████████| 235kB 6.7MB/s
Collecting pytz>=2017.3 (from pandas)
  Downloading https://files.pythonhosted.org/packages/70/94/784178ca5dd892a98f113cdd923372024dc04b8d40abe77ca76b5fb90ca6/pytz-2021.1-py2.py3-none-any.whl (510kB)
     |████████████████████████████████| 512kB 9.7MB/s
Collecting six>=1.5 (from python-dateutil>=2.7.3->pandas)
  Downloading https://files.pythonhosted.org/packages/ee/ff/48bde5c0f013094d729fe4b0316ba2a24774b3ff1c52d924a8a4cb04078a/six-1.15.0-py2.py3-none-any.whl
Installing collected packages: six, python-dateutil, pytz, pandas
Successfully installed pandas-1.2.2 python-dateutil-2.8.1 pytz-2021.1 six-1.15.0
WARNING: You are using pip version 19.2.3, however version 21.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
MyMac:python georgiospatrinos$ ./bin/python3.7m -m pip install xlrd
Collecting xlrd
  Downloading https://files.pythonhosted.org/packages/a6/0c/c2a72d51fe56e08a08acc85d13013558a2d793028ae7385448a6ccdfae64/xlrd-2.0.1-py2.py3-none-any.whl (96kB)
     |████████████████████████████████| 102kB 1.6MB/s
Installing collected packages: xlrd
Successfully installed xlrd-2.0.1
WARNING: You are using pip version 19.2.3, however version 21.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
MyMac:python georgiospatrinos$
```

Figure 7: Installation of pandas and xlrd (MacOS)

Close the terminal and blender.

Run an animation on Blender (MacOS)

Double click on the *“VDSE_ANIMATION.blend”* file.

Go to line 16 in the script and change

```
'C:\\Put\\your\\path\\here\\and\\use\\double\\backslashes.xls'
```

to the path of the generated xls file. Note that in MacOS, you do not have to put double backslashes but forward single slashes; just copy and paste your Mac filepath. In Figure *8*, you can see an example of such path.

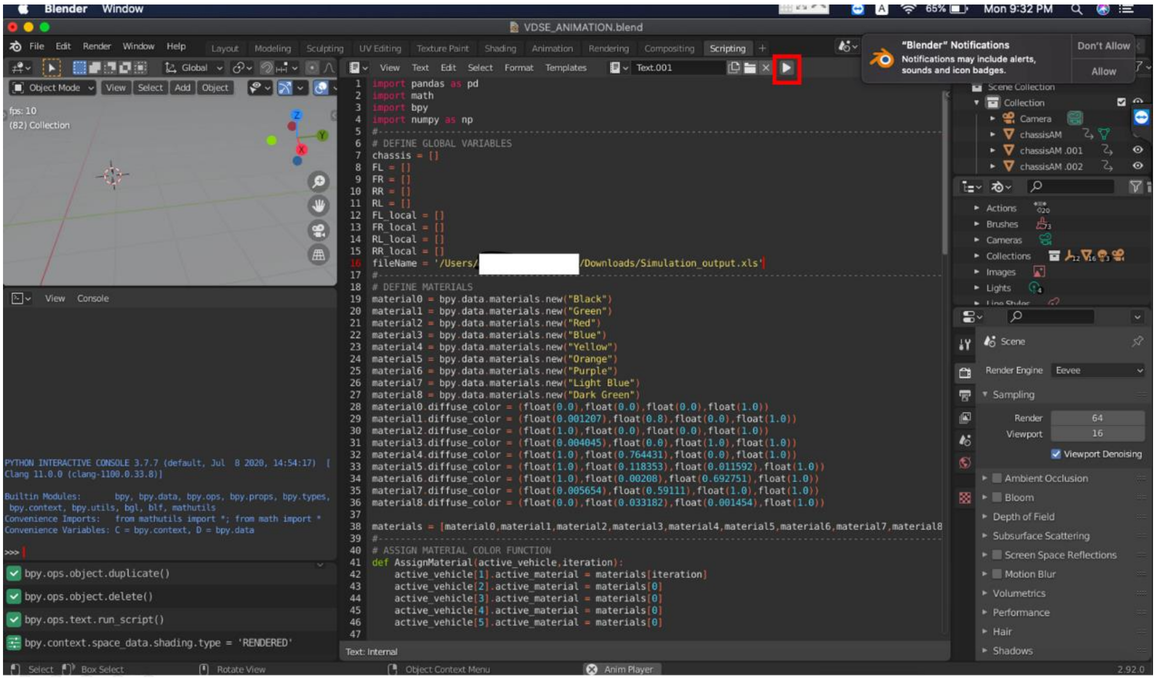

Figure 8: Change the filepath to the Simulation Ouput xls file and run animation (MacOS)

Run the script by pressing the play button on top of the script (see Figure *8*).

Go to the layout tab (number 1 in Figure *9*) and you can now visualise the animation. You can choose the render settings (number 2 in Figure *9*). If you have multiple vehicles in the same animation you can choose which vehicle to lock on (number 3 in Figure *9*) or you can choose not to lock on any vehicle by selecting "Camera". You also need to change the end frame of the simulation (number 4 in Figure *9*). This value should be equal to 10 times the end time set on Simulink. The frame rate (number 5 in Figure *9*) should be set to 10 to run the animation in real time. Finally, you can press the play button (number 6 in Figure *9*) to visualise the animation.

For multiple vehicles the colour order is the following: Green, red, blue, yellow, orange, purple, light blue, light green.

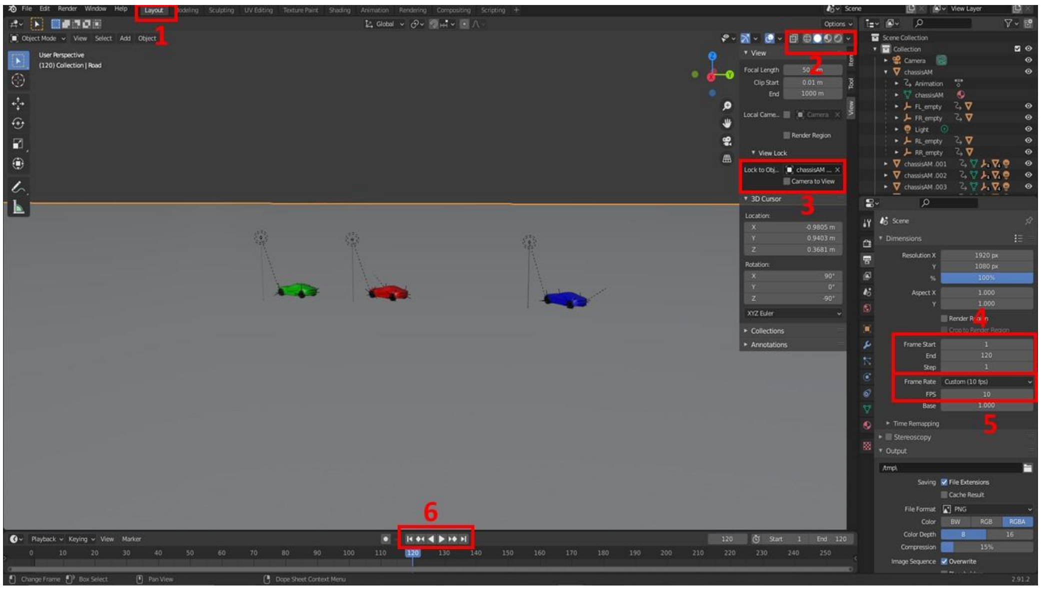

Figure 9: Blender UI (MacOS)

9. Troubleshooting

In this section some of the common issues that can be encountered are presented and how to resolve them.

You will get the following error if you didn't download the MFEVAL library.

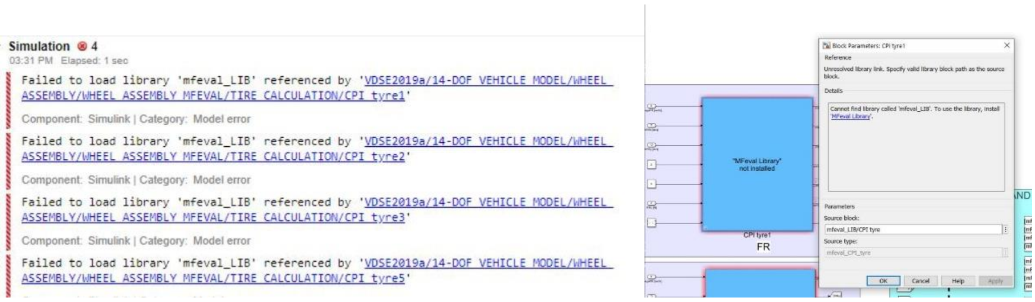

Figure 10: Error MFEVAL library

To download it go to the following link:

https://uk.mathworks.com/matlabcentral/fileexchange/63618-mfeval

Download the toolbox and double click on the downloaded file to install it. Finally, close Simulink, open it again and the error should be removed.

You will get the following error if you didn't specify a tire file.

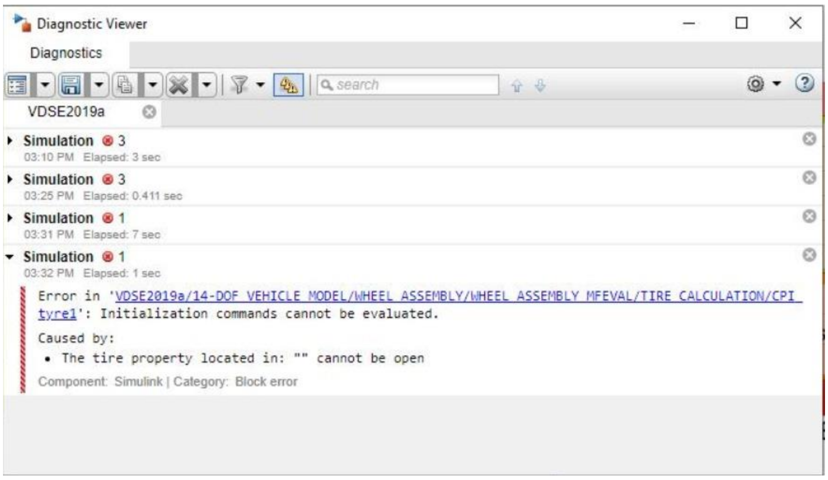

Figure 11: Error with selection of .tir file

Four tire files are provided, select one of them, copy/paste the name of the file in the tire file variables as shown in the following example:

tir_file_rear = 'car235_50R24.tir';

tir_file_front ='car235_50R24.tir';

You will get the error in Figure 12 if you have either defined the file path in the wrong format or the file name is wrong. You may also encounter this error, if your simulation output excel file contains data of simulations (different vehicles) with different end times. For more information, read the instructions at Chapter 8 (Blender Animation).

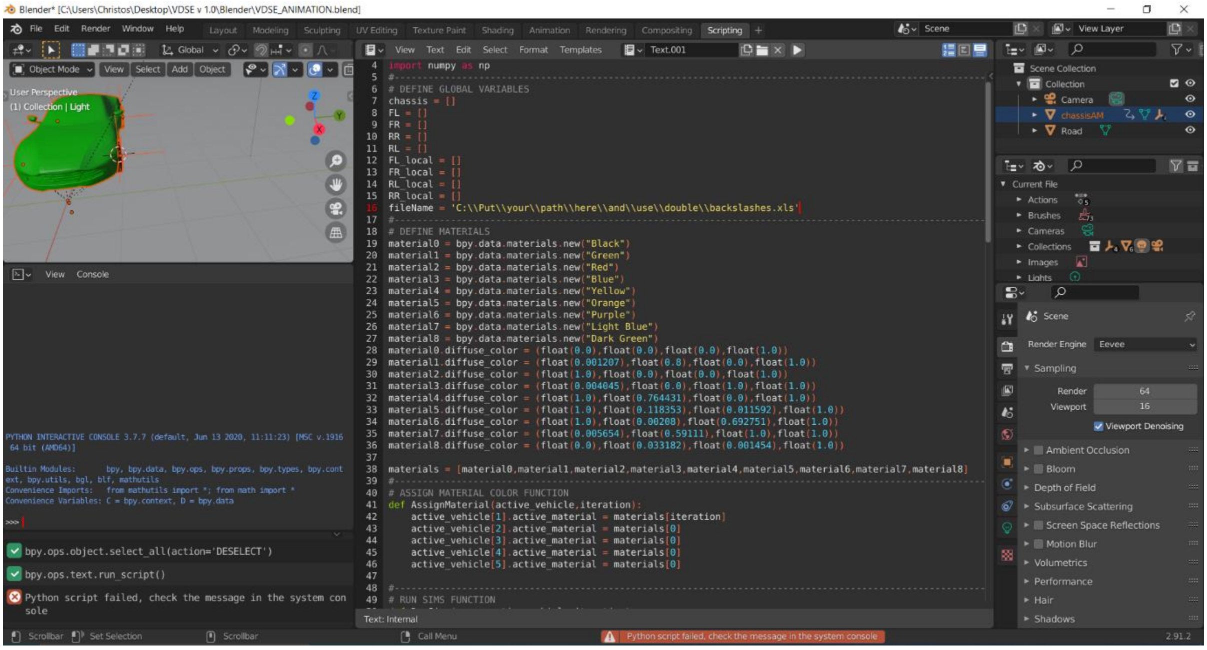

Figure 12: Blender error

10. References

[1] Jazar, R. N. (2008) *Vehicle Dynamics Theory and Application*. 1st edn. New York: Springer.

[2] Shim, T. and Ghike, C. (2007) 'Understanding the limitations of different vehicle models for roll dynamics studies', *Vehicle System Dynamics*, 45(3), pp. 191–216. doi: 10.1080/00423110600882449.

[3] Dixon, J. C. (1996) *Tires, Suspension and Handling*. 2nd edn. London: Society of Automotive Engineers.

[4] Furlan, M. (2020) *MFeval, MATLAB Central File Exchange*. Available at: https://uk.mathworks.com/matlabcentral/fileexchange/63618-mfeval (Accessed: 20 October 2020).

[5] Pacejka, H. B. (2005) *Tyre and Vehicle Dynamics, SAE International*.