POLITECNICO
MILANO 1863

Tread Generator 1.0 - User Guide

Developer:
M.Sc. Francesco Fabio Semeraro
francescofabio.semeraro@tum.de

Contents

Chapter 1

Introduction

1.1 What is Tread Generator?

Tread Generator is a GUI-based software that builds tread geometries in a parametric way and it is able to export an .stl file containing the required geometry. It is able to export MRF volumes, since they may be required to simulate lateral grooves.

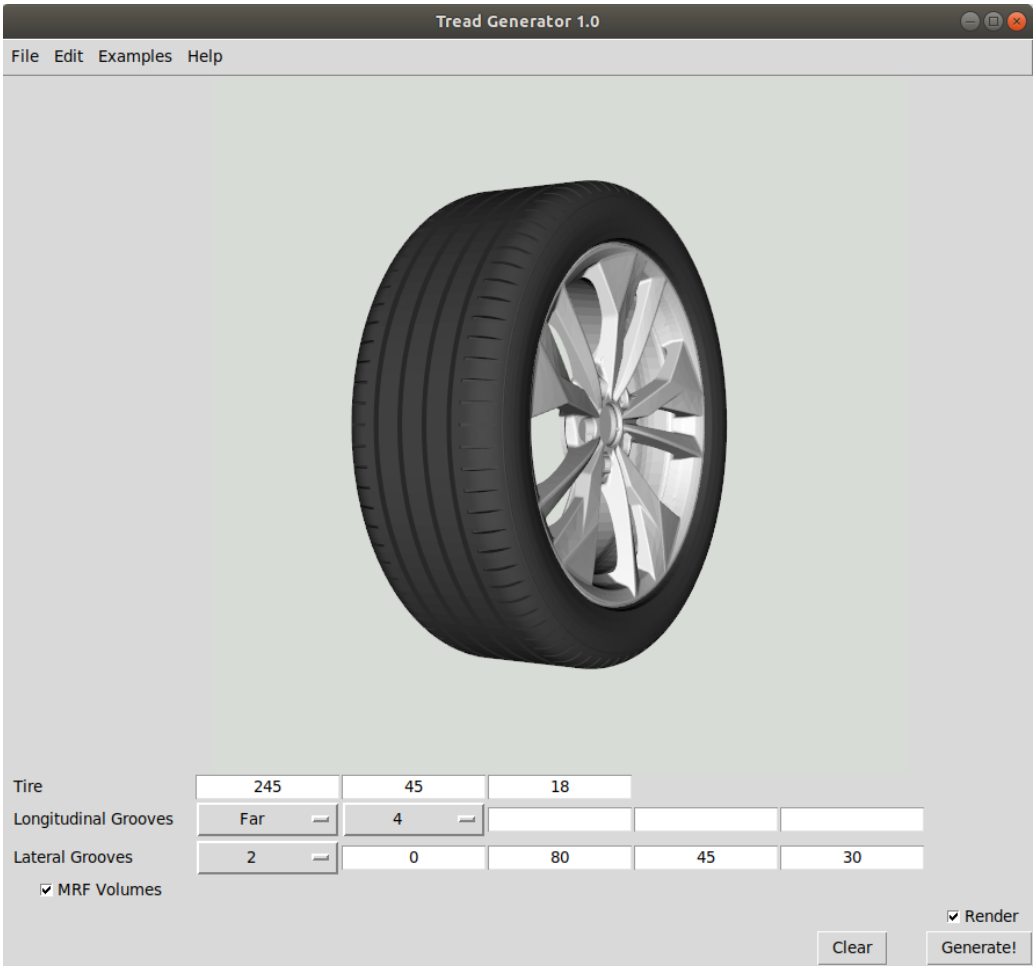

Fig. 1.1: Tread Generator 1.0 - GUI

Chapter 2

Parametric geometry generation

Tread geometry is generated by the entries of (up to) three main elements. The starting point is the common nomenclature used in tire market, that follows the regulations of the European Tyre and Rim Technical Organisation (ETRTO) and it is thought to easily identify tires for their dimension and to allow for a direct match with the rim the car is equipped with. As an example, a complete denomination of a wheel assembly can be:

$$245/45\ R18\ 100V \quad - \quad 7.5J \times 18\ ET29$$

It is composed by two parts, the first set is for tire dimension, the second one refers to the rim. Also, the second one provides very specific information and it is not commonly required for changing only the tire of the vehicle. Thus, only the first part with its three main entries is considered. In the following, the nomenclature is explained in detail:

- 245: Width of the tire expressed in millimeters.
- 45: Aspect ratio, a measure of the height of the shoulder, given as percentage of the width
- R: Construction type of tire internals. R stays for radial, which is the current standard nowadays[1]
- 18: Rim diameter in inches.
- 100: Load index, indicating how much weight that tire can support. Data are provided in tables and 100 stands for a load of 800 kg.
- V: Speed rating, indicating the top speed allowed for that tire. V stands for 240 km/h (67 m/s).

The common nomenclature for tires is adequate for the external size information and for making tire match with the rim easily. Thus, close to the ETRTO standard, there is the need to provide a parametric representation of details for the tread. Thus, an innovative parameterization has been proposed and it follows this scheme:

$$\begin{aligned} LG \rightarrow \quad & [\text{Position}]\_[\#\text{Grooves}]\_[\perp\text{Area}]\_[\text{G}_{Size}]\_[\text{G}_{Position}] \\ QG \rightarrow \quad & [\text{Position}]\_[\text{Inclination}]\_[\#\text{Grooves}]\_[\text{Length}]\_[\text{Width}] \end{aligned}$$

Where, LG stands for longitudinal grooves and QG for lateral grooves.

[1] Radial constructed tires are preferred due to their properties in terms of grip, lower rolling resistance and durability.

By these five entries, several different geometries can be generated, without the need for manually build them in a commercial CAD software. The entries are slightly different between longitudinal and lateral grooves to reach the highest degree of flexibility. Essentially, when a geometry with lateral grooves wants to be generated, it will be necessary to previously characterize also the basic longitudinal groove geometry (this feature will be implemented in the future. In 1.0 version, lateral grooves are generated automatically on a tire with four equally spaced longitudinal grooves).

Explanation of the entries:

- *Position*: Groove global position.
  For longitudinal grooves:
  - C → Central
  - L → Left (or internal)
  - R → Right (or external)
  - F → Far (equally spaced)
  - S → Slick (no grooves)

  For lateral grooves:
  - E → External
  - I → Internal
  - 2 → E + I (on both sides)
- *Inclination*: Angle between the groove axis and the horizontal axis (y), defined in counterclockwise direction.
  For longitudinal grooves there is no inclination, they are all straight by default.
  For lateral grooves any value can be inserted (i.e. +20 or 0).
- *#Grooves*: Number of grooves.
  For longitudinal grooves:
  - 0 → Slick
  - 1 → 1 Groove
  - 2 → 2 Grooves
  - 3 → 3 Grooves
  - 4 → 4 Grooves
  - 5 → 5 Grooves

  For lateral grooves any value can be inserted (i.e. 80).

- $\perp Area$ or $Length$:
  For longitudinal grooves: Frontal area of the grooves in mm^2. The area of each groove is evaluated by $\perp$Area/#grooves and given the groove height (usually 9 mm), the groove width is evaluated.

  For lateral grooves: Length of the grooves in axial direction (for inclined grooves this is the projected length) [in mm]:

  - 45 $\rightarrow$ 45
  - passing $\rightarrow$ lateral grooves connected to longitudinal grooves

- $Width$:
  For longitudinal grooves: evaluated from frontal area and number of grooves (fixed the groove height).

  For lateral grooves: Width of the grooves in tenths of mm:

  - 30 $\rightarrow$ 3 mm
  - 45 $\rightarrow$ 4.5 mm

- G_{Size}:
  For longitudinal grooves: % increment in width for each groove with respect to its reference (evaluated) width.
  For lateral grooves: Not used.

- $G_{Position}$:
  For longitudinal grooves: % increment with respect to the reference gap. Gaps are considered from the outside. Attention: some geometries put some constraints in gap dimension (like *far* geometries). The program is able to detect when a conflict in constraints happens and simply neglects the last increment.
  For lateral grooves: Not used.

Chapter 3

Thread Generator

3.1 Main Folder and start

For version 1.0, the only way to open Tread Generator is by launching its GUI typing in a terminal:

python3 wheel.py

Fig. 3.1 shows the content of the main folder.

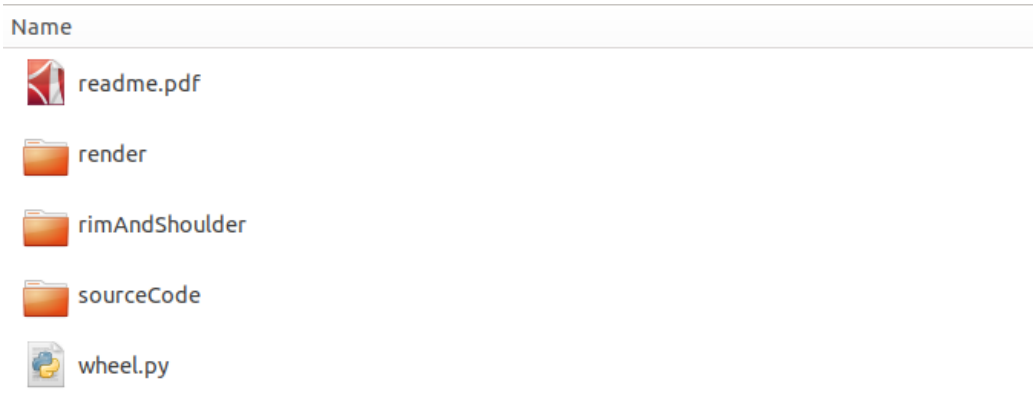

Fig. 3.1: Tread Generator 1.0 - Main folder

The main folder contains all the files required for Tread Generator to work properly. The readme.pdf file is the User Guide. In the folder *sourceCode* are stored the OpenSCAD code used to generate geometries and other python scripts to manage the entries collected by the GUI and perform a render of the .stl that has been generated. The .stl of the required geometry will be found in the main folder at the end of the generations phase, while in folder *rimAndShoulder* there are two .stl that define the rim and the shoulder. They can be changed with the preferred geometry. In the folder *render* are stored the image showed when Tread Generator is launched and the render image generated after the final geometry is ready.

3.2 GUI

When the GUI starts, it appears as showed in fig. 1.1. There are several options available in the main window and they are discussed in the following, starting from the menu in the upper part.

In the upper part there are four drop down menus:

- File: it manages the whole window. Two possibilities are present:
  - New: Close the current window and open a new one.
  - Exit: Close the current window and stop the program.
- Edit: Not assigned to specific functions yet. Inside, there is:
  - Redo: Not assigned to specific functions yet.
- Examples: Provides a fast link to the four most common geometries (Slick, 3_far, 4_far and 2_0_80_45_30 for lateral grooves). By selecting one of the examples, the nomenclature entries will be filled with the require parameters to generate the selected geometry on a 245/45 R18 wheel and display a preview at the end of the generation process.
- Help: Give some support to new users. Inside, there is:
  - Show Nomenclature: It fills each entry with an information about what is required.

The entries required for generating the tire reflect what has been presented previously during the explanation of the nomenclature, so they will not be discussed since their meaning is straightforward.

To generate a tire tread, it is necessary to fill at least the entries for tire size and longitudinal grooves. By clicking on "Generate", the GUI collects the inserted data and passes them to the program that generates the geometry (OpenSCAD). At the end of the process, the geometry will be available as an .stl file in the main folder.

In the GUI there is the possibility to ask for a preview of the generated tread that substitutes the logo/initial picture once the tire is ready and it is activated by the check button "Render". In version 1.0, a rim and a tire shoulder are provided for a tire of dimension 245/18 R18. Thus, if this specific dimension is selected, the newly generated tire tread is then assembled with those component and the entire wheel is displayed in the preview window. Otherwise, only the tire tread is displayed. As a future development, many geometries for tire shoulder and rim can be provided to this program, allowing to switch between them.

One of the possible ways to model tire rotation includes the MRF formulation for lateral grooves. In order to use this model, an additional geometry is required, namely the inside volume of lateral grooves. There is the option to generate and save the .stl for MRF volumes, name "MRF Volumes" and it is available as check button. In version 1.0, it is necessary to have this option checked to build a tire with lateral grooves.

Chapter 4

State of the Project and comments

4.1 Requirements

A brief list of the python modules required to make Tread Generator work: *tkinter, PIL, os, sys, solid, subprocess, numpy, stl, mpl_toolkits,vtkplotlib*. From this list, usually *os, sys, subprocess* come pre-installed with python3. Apart from them, it is required to have a working release of *OpenSCAD* installed.

4.2 Comments

Tread Generator 1.0 has been developed to pave the way for automatic optimization procedures for tire tread design. It is still in an initial development phase even if most of it is stable and properly working. Many ideas could be followed to improve its implementation and capabilities. The most important one is: providing an input file with a list of tires that the program is able to read and understand to generate several .stl files at the same time.

From the development point of view, tire size has been implemented in a satisfactory way. The same can be said for tires with longitudinal grooves. Many combinations have been tested and the implementation appears to be robust and reliable. The presence of the last two parameters that alter groove width and groove gap size gives unlimited freedom in the geometries that can be generated. It must be said that those two parameters should be used appropriately. Indeed, there is no mechanism to check if spurious geometries are generated. This check is left to the user, that has to select reasonable values for those entries and has to check if the generated geometry reflects the input.

4.3 Warnings

The implementation of lateral grooves is still weak and the greatest issue is the definition of the geometry of the groove in a parametric way so that it can be extended to tires with different sizes and give a similar geometry as a result. A great amount of work needs to be done to sort this out, but the remaining part of the implementation is satisfactory and also the export phase of MRF volumes is reliable, yet slow. Indeed, the huge amount of time required to generate lateral grooves ($\sim$10 min required to export both geometry and MRF volumes) is one of the major flaws of this implementation and has to be solved. OpenSCAD seems to struggle with many Boolean operations performed in a sequence to the point that there is some interest in investigating the capabilities of other tools, like Blender.

.

I want to thank you for the interest in Tread Generator. It has been developed mainly in my free time and the hope is that it can be useful to make smoother the simulation process when the attention is on tire treads. If you test it and it gives positive results, I would be glad to receive a feedback. It you try to use it and have problems, please write me and I will work to improve it.