INTERNSHIP REPORT

Automation of the post-processing procedure of numerical modal tyre vibration analyses

Dimitri Siepman

Version 1.3

January 13, 2016

Contents

Abstract

Driving comfort and noise emissions of a car have become increasingly important over the last decades. Over these years the main culprit has been recognized to be the tyre-road contact, which is considered one of the most dominant sources of exterior noise of a vehicle. As a consequence tyre manufacturers such as Apollo Vredestein have increased their efforts on improving these comfort and noise characteristics of their tyres.

Due to cost effectiveness and the ever increasing capabilities of finite element method (FEM) simulations these simulations are used more and more during the early stages of the tyre development process. This assignment will focus on the so-called modal analysis simulations, which are performed to analyze the dynamic behaviour of a tyre and give more insight in the tyres dynamic behaviour. Ultimately this should allow to make better educated design decisions during the development phase of a new tyre, without having to actually conduct extensive tests on real prototype tyres.

The modal analysis simulations are performed using Abaqus FEA, which is a finite element analysis software suite. This assignment will focus mainly on the low frequency range of the tyre vibration spectrum, which is generally considered to be until frequencies of 400 Hz. At the moment of my internship at Apollo Vredestein most of the tasks concerning the modal analysis simulations required many manual user inputs. Since these tasks often involve repetitive steps an overall necessity arouse to improve and automate parts or even the entire simulation procedure. My contribution to improve the simulation procedure is to automate major parts of the post-processing phase of the procedure.

Prior to the development of the automation tooling, first several Abaqus simulations had to be run of different tyres to get an overall feeling of modal analysis procedures. Due to incomplete and imperfect material models no real conclusions could be drawn from the obtained results. However in the end the results of the run modal analyses have been used to test and validate the functionality of the developed automation tooling.

To realize automation of the post-processing procedure a Python script, a MATLAB graphical user interface application and an user manual have been created. With the newly automated post-processing procedure, firstly the Python-script has to be used to *extract* the simulation data

from the Abaqus simulation results files. Next, this extracted data can be evaluated efficiently within the stand-alone MATLAB application, which is called the *Tyre Vibration Toolbox*. Finally, the written user manual is specically made to give a concise and practical description on how to perform the automated post-processing procedure.

With the tooling users are now able to quickly extract and evaluate data of different simulations in a time-efficient manner with less chance of error. Beside that, extras such as easy export of graphs and the mode shape recognition enhance the post-processing procedure of tyre modal analyses.

Chapter 1

Introduction

The last decades, road traffic noise has become one of the main environmental noise problems in urban areas. But even two thousand years ago, during the Roman Empire era, people already complained about road traffic noise. That time, instead from motorized vehicles the noise came from horse-drawn carriages riding on uneven cobblestone roads, which led to serious amount of annoyance. Nowadays, due to the urbanization and ever increasing amount of motor vehicles in the entire world this annoyance has increased drastically.

Apart from this annoyance it universally known that this so-called environmental noise, also known as sound pollution, has an adverse effect on public health. This is also recognized by the World Health Organization (WHO), who list insomnia, high blood pressure, ischemic heart disease and hearing impairment as the harmful health effects that can be caused by traffic noise. Other negative effects like sleep disturbance, psychophysiological stress and negative effect learning capabilities can also occur. All these effects can be translated into financials terms, in Europe alone it is estimated that the cost associated with noise pollution is between 0.2% and 2% of the gross domestic product (GDP) of the entire European Union (EU). This translates to a financial loss of a staggering 24 to 240 billion Euro in 2007 in the EU alone [5].

Therefore over the last decades regulations have been implemented to limit the amount of noise emitted from traffic. One more recent example of these kind of regulations is the introduction tyre label in 2009 by the EU, as shown in figure 1.1. Since November 2012, tyre manufacturers have to specify the fuel efficiency, wet grip and noise tyre classifications on every tyre sold in the EU market, raising more awareness at tyre buying customers. Consequently, due to these regulations tyre manufacturers have been increasingly active in noise reduction as generated by the tyres. Tyre and car manufacturers often conduct so-called *noise, vibration and harshness* (NVH) studies in order to the vibration characteristics of tyres and motor vehicles. While vibrations and noise are readily measurable, harshness is not, and therefore is treated as a subjective quality, 'measured' using human subjective impressions of the noise and vibrations. Within NVH studies two different types can be characterized; *interior* and *exterior* NVH. The former

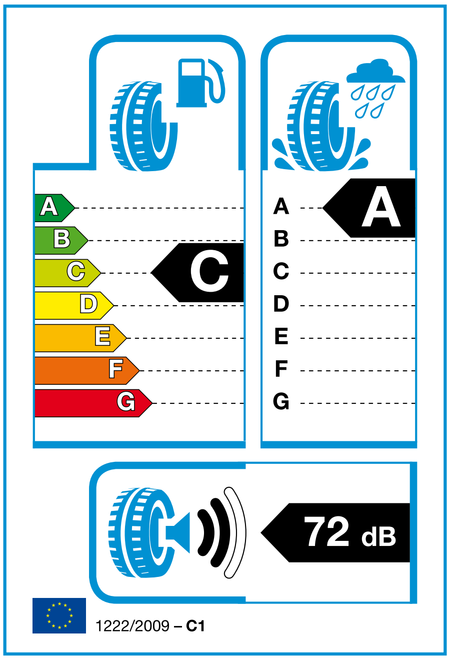


Figure 1.1: Example of EU tyre label.

focuses on the noise and vibrations experienced by the within the motor vehicle, whereas the latter mainly deals with the amount of noise radiated externally by the vehicle.

Since tyres do play a significant part in the overall dynamics of a car, many tyre companies have become original equipment manufacturers (OEM) of car specific tyres. These specifically developed tyres are designed and fine-tuned in such a way that together with the car they have a good balance between noise, rolling resistance and handling (dry & wet).

Apollo Vredestein, the company where I have done my internship, is part of Apollo Tyres from India. Vredestein, a Dutch tyre manufacturer by origin, was founded in 1909, and acquired in 2009 by Apollo Tyres. Today, still a bulk of the production takes place in the factory in Enschede. In January 2013, Apollo Tyres opened its global R&D Centre in Enschede, which houses most of the research, design and testing of all the brands that Apollo sells worldwide, including the still existing Vredestein brand. The current ambition of Apollo Tyres is to become one of the top-ten global tyre manufacturers. To achieve this Apollo Tyres will focus on the OEM industry to increase its brand awareness globally, but to also improve their in-house tyre development capabilities and procedures. One of the other current big projects within Apollo Tyre is the *platform* project, which was initiated at the start of July 2015. The main goal of this project is to create a standard 'template' of a tyre design, entailing an overall standard design procedure of a tyre, which in the future has to accommodate for new tyre design projects.

For tyre noise and vibrations, the use of finite element method (FEM) simulations and boundary element method (BEM) simulations have become more advanced and capable over past years, and are used widespread in the industry. Apollo has also been actively using FEM/BEM software packages to support research and design projects. For their simulations they primarily use SIMULIA Abaqus FEA and LMS Virtual.Lab. Currently both these software packages are only used by software specific experts within the company. Since these experts are preoccupied with a lot of different simulation tasks and research work of their own, a limited amount of simulations can be requested by people who are not familiar with these software packages. Another restriction is the amount of licenses that are available, making it important to wisely plan and run simulations to be as cost and time efficient as possible.

Another yet important part of vibration FEM/BEM analyses is the data *post-processing* of the run simulations. Often the acquired data has to be analysed and compared with other data, at Apollo currently this is mostly done purely manually. Finally by analysing and comparing the data a conclusion has to be drawn, answering the question(s) which initiated the simulation in the first place. This post-processing step can become quite tedious some times as it often involves repetitive tasks, making it also prone to human error. Lastly, this post-processing step also often requires the same FEM/BEM experts as mentioned earlier. For the reason that using the simulation-software and drawing the conclusions often requires a certain amount of expertise, increasing the experts workload even further.

Therefore one of the goals of the platform project is to develop tooling that support several aspects of tyre vibration simulations, which mostly will entail automation of repetitive tasks. Consequently this should lead to a more accessible simulation/analysis procedures, which should be able to be performed by people who are less familiar with FEM/BEM analyses and the required software. The aim of this assignment is to automate and partly standardize the low frequency tyre vibration analyses (50 to 300 Hz), performed in the SIMULIA Abaqus FEA FEM simulation software. These analyses study the dynamic properties of a tyre using the *modal analysis* procedure, which analyses the dynamic response of a tyre during excitation.

1.1 Outline

Chapter 2 will give a basic overview of tyre technology as a whole, covering topics like tyre configurations, tyre components, used materials, et cetera.

Chapter 3 describes all the noise generating excitation mechanisms concerning tyre-road noise and how these waves propagate through the tyre and rim.

Chapter 4 explains the difference between the low and high frequency analysis categories, and how these simulations are set-up and performed.

Chapter 5 covers the parts of the simulation procedure that will be automated, and how this has been actually realized.

Chapter 6 shows the results of the run simulations and also describes the tests and validations that have been performed to assess the made automation tooling.

Chapter 7 concludes the entire report with the conclusions and recommendations.

Chapter 2

Basics of tyre technology

What many people do not realize is the sheer complexity that is involved with the design and production of tyres. The fact is that tyres are responsible for the vehicle's direct contact with the road, hence making them of paramount importance for the handling of the car and the safety of the passengers. To accommodate for all the different properties that a tyre must have, a tyre is constructed of many different materials and layers, all having their specific purpose. This chapter will elaborate briefly upon several of these subjects to give some context to this study.

2.1 Tyre construction

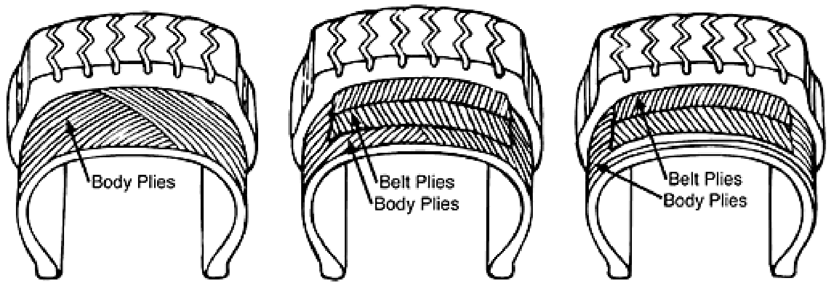
There are many different components to a tyre's construction and a lot of different ways how to construct a tyre. This section will cover some of the common tyre configurations and all the different tyre components and materials.

2.1.1 Tyre configuration

Mainly three different base-types of tyres can be distinguished by their specific carcass orientation, graphic representations of these different configurations can be seen in figure 2.1.

Diagonal bias tyres

The diagonal bias tyre is still used in some applications due to its good sidewall properties, being more resilient against punctures at the sidewall. Therefore this type of tyre is used in rough terrain situations such as in mines and forests. Beside this, due to its simple construction, the manufacturing process is relatively easy and cheap. The main disadvantage however is that this tyre is prone to heat build-up in the side wall. This is caused by shear between the body plies, which also causes increased wear.



(a) Diagonal bias **(b)** Belted bias **(c)** Radial

Figure 2.1: Different types of tyre construction.

Belted bias tyres

A variation of the diagonal bias tyre is obtained by adding a belt in the tread direction at the circumference, and is called the belted bias tyre. This addition of the belt stiffens and strengthens the tread area and restricts the expansion of the body carcass in radial direction, which stabilizes the tyre as a whole and improving the overall handling of a vehicle. The issue with the heat build-up, as mentioned in the previous tyre type, still remain unchanged. Adding the extra components also increases the costs by a significant amount.

Radial tyres

This configuration has body ply cords that lie radially from bead to bead, oriented perpendicular to the tyres centreline. In addition, this configuration of tyre has belts laid around its circumference, under the tyres tread. These belt are oriented with a slight angular offset to the centreline of the tire and are there to add strength and stability in that region. Currently this is the most wide-spread used tyre configuration in passenger vehicles, and will be the only tyre that will be analysed within this report.

2.1.2 Tyre components

As said, a tyre consist of many different components. This section will cover all the general components related to radial tyres. An exploded view of the cross-section can be seen in figure 2.2, every numbered component is briefly explained in the following list:

① **Tread:** The tread is the outer surface that directly makes contact with the road and therefore has to provide the necessary traction for driving, breaking and cornering, whilst having acceptable rolling and wear resistance. The pattern that is molded into the tread to channel water out of the footprint while driving on a wet road surface. Beside this, the pattern also has to designed in such a way that it wears evenly and to minimize tread pattern noise.

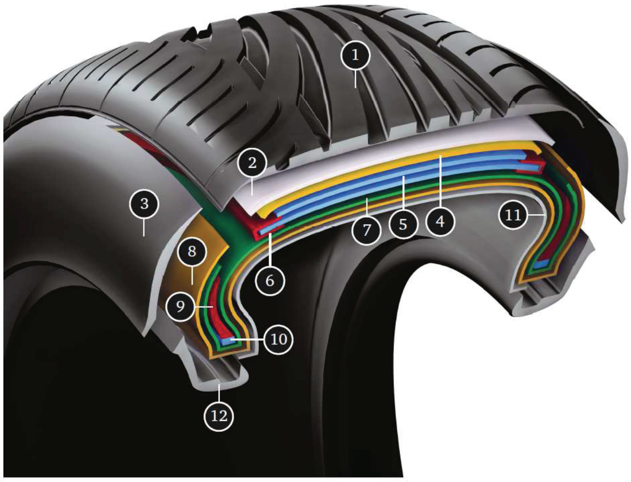


Figure 2.2: Exploded view of a radial tyre's cross section.

② **Undertread:** The undertread is an extra layer that is placed directly under the tread, providing better adhesion of the tread and the cap ply and covering the ends of the cut belts.

③ **Sidewall:** Tyre sidewall protects the tyre – or to be more specific, the carcass layer – from abrasion, impact and flex fatigue, and is made from a rubber compound that is resistant to cracking, UV radiation, oxygen, ozone and heat.

④ **Cap-ply:** Layer made of rubber coated nylon that provides extra circumferential stability by restricting radial expansion. Especially for higher speed rated tyres, since there are higher centrifugal forces at play.

⑤ **Steel belt (two):** The belts are made of steel cords that are embedded into a rubber layer, and are commonly put into a cross-ply configuration around the tyre's circumference using two belts. These layers are there to stabilize the shape of the tyre in circumferential direction and give the tyre some penetration resistance.

⑥ **Belt rim tape:** Belt rim tape is used to cover the steel belt edges to protect other components as these belt edges can be sharp.

⑦ **Carcass layer (1):** The carcass layer is made of rubber coated cords that lie as defined by the tyre's configuration, as described in section 2.1.1, and wrap around both bead bundles on either sides. These layers provide the strength the contain the air pressure, carry the load and have a sufficient amount of sidewall impact resistance. Due to the constant fluctuating load that these layers undergo, a high fatigue resistance is required. Tyres that have to carry a large loads, for example truck tyres, usually have more than one carcass layer and have cords made of steel. Tyres that carry normal loads often have one or two carcass layers, which are made of either nylon, polyester or aramid depending on the application purpose of the tyre.

(8) **Carcass layer (2):** Same as previous description.

(9) **Bead filler strip (apex):** This often very stiff and hard material fills the void above the bead bundle and in-between the carcass layer(s). This way irregularities are prevented with the carcass layer turnup, and it is also made sure that no air gets trapped above the bead bundles. Varying the apex height and material properties does significantly affect the handling characteristics of the tyre.

(10) **Bead:** The bead wires are made of brassed steel wires that are coated with rubber and then wound into bundles. The brass coating enhances the adherence with the rubber. The main function of the bead bundles is to hold the tyre on the rim and to withstand the high dynamic forces that are passed via the rim onto the tyre and vice versa.

(11) **Inner liner:** The inner liner is the inner surface layer that primarily has to have a low air permeability. This layer will keep the inflation gas within the tyre, since tyres are nowadays most commonly tubeless, i.e. with no inner tyre tube.

(12) **Gum strip (gum chaffer):** This rubber tyre component makes contact with the tyre rim and therefore has to be resistant against chafing and provide an airtight seal between the rim and tyre.

2.1.3 Tyre materials

A modern pneumatic tyre is constructed of many different materials. One of the most characterizing material of a modern tyre is rubber. A tyre is constructed using many different rubber compounds, which all have special characteristics, tailored to their specific use case. Most of the other materials are commonly used as reinforcement materials, which are added for structural strength and stability in the sidewall and in the tread of the tyre.

Rubber compounds

The rubber compounds are the main structural materials that hold the entire tyre together by encapsulating all the reinforcement materials. This encapsulation also protects these reinforcement from abrasion and deterioration. The rubber compound primarily consists of the rubber polymer, and is often enhanced with several different additives. To improve the mechanical properties and decrease the stickiness, the rubber compound is vulcanised. This chemical process, called vulcanisation, forms cross-links between polymer chains by heating the compound up, and via the addition of sulphur or other curatives. During the manufacturing process of a tyre, the uncured sticky tyre is often referred to as the *green tyre*.

To give a general idea, here are some of the ingredients of which rubber tyre compounds can consist of:

- The **polymers** form the backbone of the compound and consist of natural or synthetic rubber.

- **Fillers** enhance the compound by strengthening the molecular structure. Carbon black is an example commonly used filler and has the benefits of being cheap and bonding well to the rubber polymers. More recently silica has been found to be also be a very good filler material, giving tyres better rolling resistance properties when used correctly.
- To improve the processing characteristic of rubber compounds **softeners** are used. Examples of softeners are petroleum oils, resins and waxes. Another function of softeners is to further improve tack/stickiness of unvulcanised rubber compounds during tyre manufacturing.
- To protect tyres from heat, oxygen and ozone, **antidegradents**, such as waxes, antioxidants and anti-ozonants, are added to rubber compounds.
- **Curatives** are added to aid and improve the vulcanization or curing step of the rubber compound. To achieve the desired properties sulphur and other accelerators and activators are added to obtain a rubber compound with the desired properties.

The type of ingredients added to the compound and the quantity thereof allows great customization of the final rubber compound properties. Therefore, as said earlier, a tyre consists of several different compounds, each specifically tailored to its particular use. The difficulty with compounding lies within the fact that adjusting the rubber compound properties often affects other performance parameters of the material as well. For example, a tread compound with good dry traction and handling on dry roads might have a high rolling resistance. Hence compounding is an art of finding a good balance between different performance indicators of a rubber compound. Another dimension to this is that the compound also has to be cost-competitive and processable during manufacturing.

Reinforcement materials

The type of reinforcement materials used differ from tyre to tyre, depending on the tyre's application and configuration. Common fibre materials are brass coated steel wires, nylon and polyester. Less common, but sometimes used for specific cases are fibre materials such as rayon and aramid.

2.2 Tyre performances

Within vehicles tyres are one of key parts that allows it to handle as it is supposed to, as tyres transmit the forces from the vehicle to the road. Flexibility of the tyre allows it to form a tyre-road contact patch to effectively transmit the forces which occur during driving. Due to the relatively small size of the contact patch, forces acting upon this contact patch are significantly large, especially during certain driving manoeuvres such as cornering and breaking. At the same time the air pressure within the tyre carries the radial load, which mostly consists of the sheer weight of the vehicle.

The main function of the tyre can be split up in several different sub-functions. The way a tyre performs with the specific sub-functions can be seen as the tyre's performance as a whole. These are the following tyre performances as described in [1]:

- Safety: Braking, high speed and hydroplaning performance
- Handling: Handling and traction performance
- Economy: Rolling resistance and wear performance
- Environment: Weight, compound ingredients, noise performance
- Customer: Price, design, mechanical and acoustical comfort

These tyre performances are used intensively during the development procedure of a tyre. Depending on its application, the tyre performances are optimized to meet the specifications as good as possible. Since some of the performances affect other performances in a negative way trade-offs have to be made. The procedure of balancing the tyre performances to finally obtain a good mix of different performances is very complex and time consuming. For example a trade-off between hydroplaning and noise is known to be apparent, since by adding more grooves within the tread water capacity is increased, hence improving the hydroplaning performance. Consequently the increased amount of grooves led to more contact pressure variations at the footprint, increasing the wear and noise.

Tyre manufacturers always have, and likely will, make intensive use of prototyping and testing tyre concepts to assess the tyre performances. Nowadays however an ever increasing amount of tyre performance assessment is done using simulation tooling, making certain development steps more efficient, hence reducing the development costs and potentially the development time of a tyre.

2.3 Tyre dimensions & coding

Of each tyre there are a lot of different size variants due to the huge variety of rim sizes. To have proper tyre and rim dimensioning, two major worldwide organisations are active that influence the national tyre standards, being the European Tyre and Rim Technical Organisation (ETRTO) and the American Tire and Rim Association (TRA). The ISO Metric tyre code of a tyre engineered to ETRTO standards is build-up as follows (code from left to right):

- **3-digit number:** Nominal tyre width in [mm].
- **/**: Separator.
- **2-digit number:** Aspect ratio of the tyre, which stands for the sidewall height in percentage of the tyre's nominal width.

- **B, D or R**: Type of tyre construction (section 2.1.1), where the letters stand for bias, diagonal and radial respectively.
- **2-digit number:** Rim diameter in [inches].
- **2- or 3-digit number:** Load rating of the tyre.
- **1- or 2-digit/letter:** Speed rating of the tyre.

An example of a tyre code label is presented here:

```
205/70R16 90H
```

which is the code for a 16 inch radial tyre with a nominal width of 205 mm, an aspect ratio of 70%, meaning a sidewall height of about 143.5 mm, a load index of 600 kg and a speed rating of 210 km/h. Beside this standard tyre code there are a lot of other additional tyre markings that specify if it is a winter tyre, where the tyre is made, if it is tubeless, et cetera.

Chapter 3

Physics of tyre-road noise

Over the years all sorts of different tyre-road noise generating mechanisms have been recognized. Sandberg and Ejsmont [8] give a good overview in their comprehensive tyre reference book published in 2002. More recently Bekke [1] has published his dissertation report, which contains his interpretation of all the tyre-road noise and vibration phenomena. The overview of Bekke will be presented briefly in this chapter to give the reader a general feeling for all the noise and vibration generating mechanisms. To describe the physics of tyre-road noise Bekke introduces a function network of tyre-road noise, which is shown in figure 3.1. Within this overview each block represents a part of the tyre-road noise physics and the arrows describe the physical quantities which link each block to form the entire network. The following sections will describe each of physical blocks briefly.

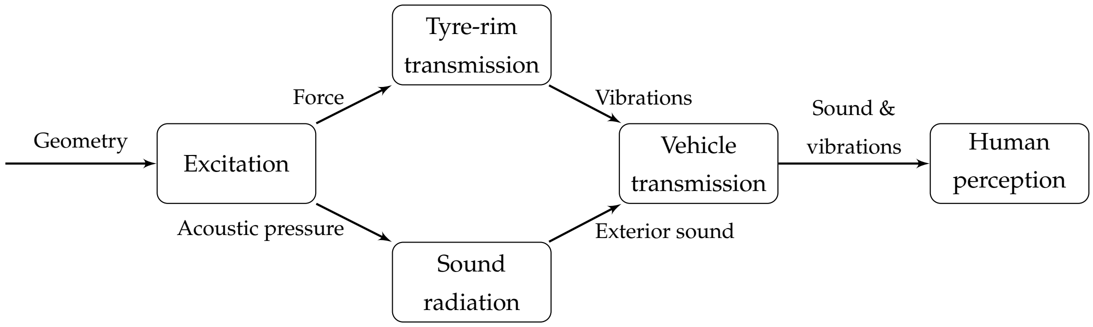


Figure 3.1: Basic function network of tyre-road noise as introduced by Bekke [1].

3.1 Structural excitation

With tyre-road noise and vibrations two excitation categories are recognized: Structural and aerodynamical. The former will be treated in this section and the latter in the next section.

During driving the tyres roll according to the speed, during this rolling tyres deform continuously as a new part of the tyre gets into contact with the road constantly. Several noise and vibration generating mechanisms are induced by this continuous deformation of a tyre, all of which briefly explained in the following list:

Road roughness excitation: When the tyre is rolling over a rough road surface small local pressure variations occur within the contact-patch; a phenomena often called **road impact**. These variations excite local random oscillations of the tyre's structure, whereof the most dominant frequency correlates with the actual road roughness on which the tyre rolls.

Tread pattern excitation: In a similar manner to that of *road roughness excitation*, the tread can also be seen as a geometrical roughness since it consists of tread blocks and grooves. Hence there will also be pressure variations occurring locally at these tread blocks, deforming the tyre's construction continuously as a whole. The main difference however is that the tread pattern repeats itself every revolution, making the dominant frequency rather deterministic as the tyre tread pattern noise has a certain tonalness to it. To prevent having a very dominant tonal noise tread patterns are often designed to have a certain *pitch sequence*, which typically means 2 to 5 different pitch sizes are used to form the entire tyre tread. This way the noise emission is spread over a wider frequency spectrum instead of a single frequency.

Tyre non-uniformity excitation: Often the manufacturing process of a tyre requires splices to form all the different layers, adding practically inherent non-uniformity of the tyre at the splice. During the vulcanisation procedure the green tyre[1] also often does not exactly match the vulcanisation mold, which causes a non-uniform distribution of the rubber. Consequently these non-uniformity lead to a mass imbalance of the tyre, which whilst rolling will become a source of structural excitation and therefore noise and vibration.

Stick-slip excitation: This excitation phenomena is very dominant at higher frequencies (> 1 kHz), especially when the tangential forces on the tyre are high, for example when breaking or steering with high intensity. The basic principle of the stick-slip is that tread blocks of the tyre *stick* due to enough friction force, where after it *slips* as soon as the forces overcome this friction force. During this slip of the tyre block all the accumulated potential energy is released it excites the tyre construction and tread blocks causing them to vibrate at their eigenfrequencies.

Stick-snap excitation: On very clean roads tyres can become sticky, meaning that tyre blocks momentarily adhere (*stick*) to the road where after they are released again (*snap*). During the stick the tyre block will be stretched a little, storing a small amount of potential energy. After the tyre block is released from the surface, the block will vibrate back to its original position. In practice, this mechanism is not very important since there is always some dirt on the roads which reduces the tyres adhesion to the road. Tyre tests in laboratories do however stick significantly to the clean drums, hence generating more vibration than would realistically be the apparent.

[1] The name *green tyre* refers to an uncured tyre, being a tyre that has yet to be vulcanized.

Air pumping by groove deformation: Even though this phenomena is not thoroughly investigated and understood the basic principle will be briefly explained: During the deformation of the tread blocks and grooves locally the air gets sequentially compressed, and at the trailing edge of the contact patch the air gets decompressed again.

This concludes the list with all the structural excitations as described in Bekke's report [1].

3.2 Aerodynamical excitation

The aerodynamical is the second excitation category and contains all the phenomena which are induced by aerodynamical means. The following list gives a brief overview of these phenomena:

Air turbulence: The air surrounding a rotation tyre and rim is put into motion and turbulent airflow occurs leading to airborne vibration, i.e. noise. This mechanism is very significant at high driving speeds and much less apparent at lower speeds [8].

Air pumping in road cavities: Similar to the air pumping by groove deformation this phenomena also generates by pressurizing air within road cavities of non-absorbing roads. In particular, tread blocks close off small road cavities, compressing the air within these sealed cavities by the curvature of the tread block. After full contact is lost the pressurized air gets released again, consequently generating the sound propagating waves, which are in turn perceivable as noise.

3.3 Tyre-rim transmission

The structural excitation phenomena induce structural vibration waves that propagate through the tyre in all directions. These structural vibrations of the tyre generate *exterior* noise. Transmission of these vibrations through the rim to the car itself leads to structure borne *interior* noise and vibrations, which are mostly perceivable by the car driver and passengers. This section will mainly focus on the tyre vibrations, and will briefly cover rim vibrations and the acoustical air cavity mode.

3.3.1 Tyre vibrations

To get a better understanding of the physics behind tyre vibrations, firstly the *free non-rotating* tyres are discussed, where after the *loaded* tyres are covered.

Free non-rotating tyres

With tyre vibrations three different structural waves can be distinguished that propagate along the tyres circumference: *bending* waves, *longitudinal* waves and *rotational* waves. Figure 3.2 shows a graphical illustration of these three waves. To understand how resonances within a

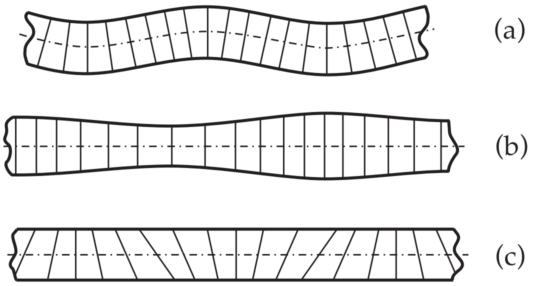


Figure 3.2: The three different wave types: bending wave (a), longitudinal wave (b) and rotational wave (c).

tyre occur a simple 1D ring is considered with a point source excitation with a frequency of ω [Hz]. From this source point waves propagate in both circumferential directions, the velocity of which is non-linearly dependant on the excitation frequency at the source point. Therefore the wave numbers and wavelengths are also non-linearly dependant on the frequency.

Resonance modes occur when the wavelength λ equals the total circumferential length $2\pi R$ of a tyre divided by an integer m. The circumferential wave and mode numbers can be described with the following two equations

$$m\lambda = 2\pi R \quad \text{and} \quad k_\phi = m/R \tag{3.1}$$

where:
m is the circumferential mode number;
λ the wavelength in m;
R the radius of the tyre in m;
k_ϕ the circumferential wave number in m^{-1}.

In reality structural waves propagate in all directions of a tyre and not just the circumferential direction. Two main directions can be distinguished and a combination of both:

Radial mode: Waves travelling in circumferential direction displace the tread in *radial* direction of the tyre, therefore these kind of modes are called radial.

Axial mode: Here the waves propagate in axial direction, which is along the tyre cross-section. These waves get reflected by the 'rigid' rim and travel back, resulting a standing wave pattern.

Combination mode: The modal behaviour of a tyre also show combinations of the two aforementioned directions, meaning that these modes contain axial and radial waves.

Nowadays to study the vibration behaviour often FEA software is used to perform a modal analysis. One of the main steps within these analyses is the extraction of the eigenfrequencies and eigenvalues using an eigensolver. To properly and consistently name the mode shapes the labelling convention of Kindt is adopted [5], which is presented in section 4.4.3.

For unloaded tyres it is important to note that most resonances are double poles due to their rotational symmetry. Therefore there are eigenfrequencies where two modes exist, but with a different phase. For loaded tyres these double modes are split into two separate ones with a slightly different frequency.

Propagation of waves

Rubber, being the most characterizing and important materials of a tyre, has many unique properties, one of which is its viscoelastic behaviour. Due to this viscoelasticity, damping is also inherently present due to the viscous part of the behaviour. Especially at higher frequencies where the loss modulus factor E" is higher, damping caused by viscoelasticity is also more prevalent. For the propagation of the waves this translates into the following:

- Low frequency range (< 400 Hz) waves show little decay, allowing the waves to form the global resonance mode shapes. Within this range Brinkmeier has shown that belt bending modes are dominant [2].

- High frequency range (> 400 Hz) waves are damped much more, hence little to no mode shapes can be formed as waves damp out quickly. Instead of global mode shapes local resonances occur close or at the source of excitation. The 400 to 800 Hz range show mostly side wall modes, whereas everything above 800 Hz appear to be mostly local modes [2].

According to Brinkmeier this also translates the modal density, or in other words the amount of resonances, above 400 Hz becomes significantly large, hence the computation time also will increase dramatically.

Frequency response functions

Another useful way to get an insight of the tyre-rim transmission is by measuring the vibrational response that is caused by a known excitation, which is known as the transfer function. One often use form is the excitation to response in frequency domain, named the *frequency response function* (FRF). To obtain the FRF a frequency sweep can be performed in the frequency range of interest, in FEA this done using a steady state dynamic analysis. The transfer function in the time domain is known as the *impulse response function* (IRF), one way to obtain this function is by performing a transient analysis within a FEA program. To transform the IRF to a FRF a fast Fourier transform (FFT) can be performed. Six different FRF types can be distinguished depending on the way the transfer function is measured, an overview can be seen in table 3.1. Within the table it becomes clear that the way the transfer function is determined classifies the type of FRF.

Standard	Description	Inverse	Description
u/F	Dynamic compliance	F/u	Dynamic stiffness
$\dot{u}/F$	Mobility	$F/\dot{u}$	Impedance
$\ddot{u}/F$	Inertance or Accelerance	$F/\ddot{u}$	Apparent mass

Table 3.1: Different types of frequency response functions.

Loaded and rotating tyre

By loading a tyre with a force and putting it into contact with a road surface the tyre's rotational symmetry is lost due to the deformed contact patch. Therefore the double poles that are normally present with unloaded tyres are split into two single ones. Since these are still related to the same mode of the unloaded tyre the amount of wavelengths in the radial and lateral direction remain the same. Comparing the frequencies between the unloaded and loaded tyres show a frequency difference of around 3 – 10% [1].

Upon rotation of a tyre the structural waves of the tyre in both the circumferential directions travel at different speeds. When observing these waves from the fixed reference system (vehicle) the direction and velocity of these waves alter continuously while the tyres rotate. Due to the *Dopplershift* the observed waves from the fixed reference system have different frequencies. For example when looking at a (3,0) radial mode shape, the waves of the non-rotating free tyre in both direction have a frequency of 131 Hz. Whereas the rotating unloaded tyre at a velocity of 60 km/h waves having frequencies of 102 Hz and 171 Hz [1] are observed in either direction, which is a significant difference. Therefore conducting simulations with rotating tyres are meaningful as these give better insight in the real-world dynamic behaviour of a tyre.

3.3.2 Acoustical cavity mode

The cavity inside the tyre-wheel rim assembly is known to contribute to tyre noise generation [1, 8]. This mechanism seems to contribute more on interior noise generation than it does on exterior noise, at least when measured with A-weighting [8]. In some cases this resonance gets amplified when this frequency lies close to the structural resonance frequency of the rim, leading to a further increase of noise generated by the torus cavity mechanism. The resonance frequency of a inflated tyre (air) usually lies between 220 and 280 Hz.

3.3.3 Rim vibrations & tyre-rim transfer

The rim on which the tyre is mounted evidently also has influence on vibrations and how these vibrations are transmitted to the vehicle. Since in this study the rim will always be considered as a rigid surface the influence of the rim is of no relevance to this study, hence this will not be explained further.

Chapter 4

Tyre vibration analysis procedures

To make better design decisions during the design process of a tyre all kinds of different finite element method (FEM) simulations are conducted within Apollo. The software package that is used the most is *Abaqus* from SIMULIA, with Abaqus several tyre behaviours are predicted, tyre vibrations being one of which. As stated in section 3.3.1 tyre vibrations are generally categorized into two categories, namely low frequency behaviour, normally until 300 to 500 Hz; and high frequency behaviour, being conventionally above 500 Hz. For both categories different simulation approaches are used to predict this behaviour, this chapter will elaborate upon the procedures of these approaches and how they are set-up.

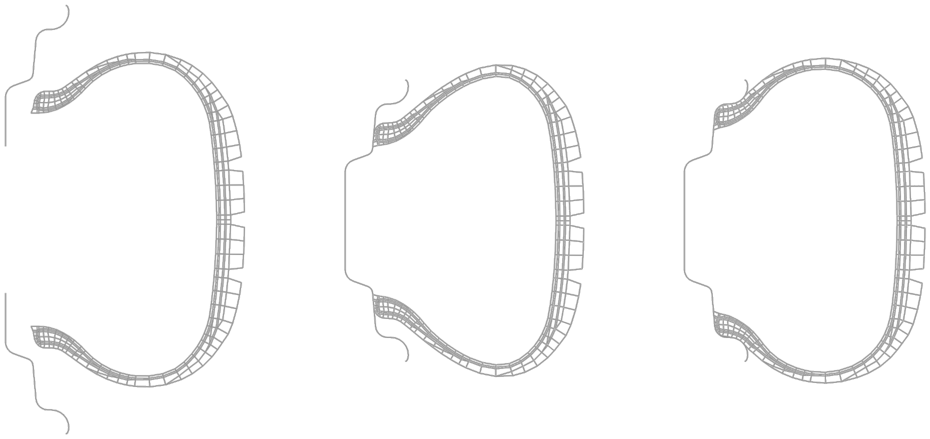
Simulations of either low or high frequency dynamic tyre behaviour within Abaqus consists of several subsequent steps that are required to simulate parts of its static and dynamic characteristics. This chapter will describe the complete simulation procedures that will be done in this study; from the start, the geometric model, to the final results. Within this procedure three major phases can be distinguished, being the pre-processing, the processing itself and the post-processing. The following sections will elaborate upon all the different phases, but firstly the two categories will be explained.

4.1 Categories of tyre vibration analyses

Although this study mainly focuses on the low frequency behaviour, the high frequency behaviour will also be covered in this chapter. The low frequency behaviour within this study is analysed with so-called *modal analysis simulations*. For tyre simulations these modal analyses study the dynamic response of a tyre by measuring and analysing the dynamic response of a tyre during excitation. The high frequency behaviour of a tyre in this study will be analysed using a *transient analysis method*. This analysis type excites the tyre for a short period of time, during this period of time the nonlinear transient dynamic behaviour is measured and analysed.

4.1.1 Low frequency modal analysis

The modal analysis procedure within Abaqus consists of two simulation parts, being the *2D-axisymmetric simulation part* and the *3D-revolved simulation part*. These two simulation parts are run subsequently in that order – that is from the 2D-model to the 3D-model. These simulation parts consist of several simulation steps, all having a specific purpose that ultimately leads to data which can be used to analyse the low frequency behaviour of a tyre. An overview of the entire procedure can be seen in figure 4.2, the following two paragraphs will briefly cover all of the steps as displayed in the aforementioned figure.



(a) Undeformed, rim still split **(b)** Tyre mounted on rim **(c)** Tyre fully inflated

Figure 4.1: The three 2D-axisymmetric simulation steps, with the rim cross-sections on the left.

2D-axisymmetric simulation

The subdivision of the simulation in a 2D and a 3D part is computationally efficient because it uses the axisymmetric nature of the tyre to firstly simulate a part in the 2D-space. After that 2D-simulation the results can be used to create a 3D tyre model. In order to do so two steps are performed as can be seen in figure 4.2. These steps are solved iteratively using the Abaqus/Standard solver, which makes use of a implicit solution technique. The two steps that are performed in this simulation part are as follows:

Rim mounting step: This step simulates the mounting of a tyre into the rim. In the simulation this is done by initially dividing the rim into two separate rigid body parts, and positioning them with a specific offset from their final position. A graphical representation of this can be seen in figure 4.1. During this simulation step the rim parts will be translated to their correct position using boundary conditions. During this translation the surface of the rim(s) will collide with the surface of the tyre, which will deform the tyre by a significant amount, as can be seen in figure 4.1b.

Inflation step: Directly after a successful rim mounting procedure the inflation step is executed. During this step a distributed load is applied at the inner surface of the tyres body, which represents the specified inflation pressure of the tyre. During this step the tyre should mount itself correctly in the rim and will expand a little around its entire circumference, as can be seen in figure 4.1c.

3D-revolved simulation

After the 2D-axisymmetric simulation has run successfully, the solution of this fully mounted and inflated tyre is ready to be used in the 3D-revolved simulation part. The expansion from the 2D to the 3D model is done by revolving the 2D-axisymmetric deformed state solution around the rotational symmetry axis, this finally yields an entire 3D tyre model. Secondly the road surface also has to be added to the model; this is done by adding it as a rigid road surface. After the 3D-models and boundary conditions are all prepared the following steps will be performed:

Equilibrium step: After the conversion to the 3D-model the static equilibrium of the entire tyre has to be ensured. Therefore this simulation step re-applies all the boundary conditions to the 3D-model and verifies if the current state of deformation is correct.

Footprint (displacement controlled) step: Before applying the actual load to the tyre, the tyre is put into contact with the road using a predefined displacement. The reason why this step is performed is because it is known by practice that the solution-procedure has a better convergence rate than it would have by directly loading it with a force.

Footprint (force controlled) step: This step finally loads the road surface with the specified loading force. Since the rim is fixed in space, the tyre deformation will change accordingly so that it gets to its static equilibrium.

Road fix: This step imposes boundary conditions that fixes the road surface in space. In theory this step could also be included into the frequency extraction step. Past experience in a validation study of Apollo showed that the road surface does move slightly when combining both steps into one, hence the separate *road fix* step.

Frequency extraction step: Now that the tyre is in its deformed state this step will use an eigensolver algorithm to obtain all the eigenfrequencies and eigenvalues within the specified frequency range.

Steady state dynamic analysis step: This step harmonically excites the tyre with 4% of its rated loading force with specific frequencies, and finally captures its resulting force-response. These specific frequency points are commonly determined using the eigenfrequencies that have been found during the frequency extraction step.

The two last steps of this procedure can be performed in different ways, using different solution methods. The solution methods available will be explained in section 4.2.3.

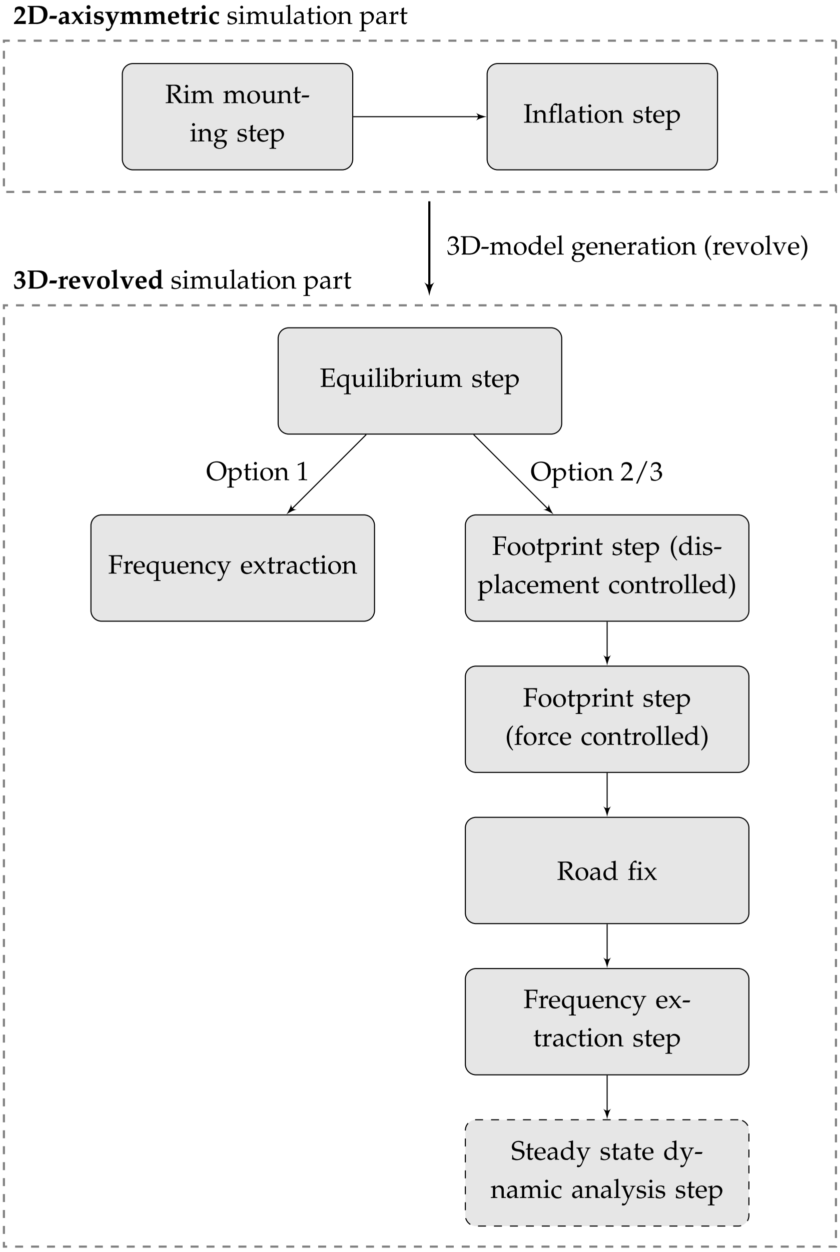

2D-axisymmetric simulation part
Rim mounting step
Inflation step
3D-model generation (revolve)
3D-revolved simulation part
Equilibrium step
Option 1
Option 2/3
Frequency extraction
Footprint step (displacement controlled)
Footprint step (force controlled)
Road fix
Frequency extraction step
Steady state dynamic analysis step

Figure 4.2: Overview of low frequency analysis procedure of a tyre with the different options visualized.

4.1.2 High frequency transient analysis

Since the high frequency transient analysis study is not the main focus of this assignment the procedure will only be briefly covered in this section. Transient analyses of a tyre are generally performed by performing a rolling analysis. Similar to the modal analysis, firstly a 2D-axisymmetric and a 3D-revolved step are performed to get a full 3D model that is already in contact with the road. After that two transient analysis specific steps will be executed to obtain the final transient analysis results. Here follows an overview of all the steps involved:

1. 2D-axisymmetric step;

2. 3D-revolved step;

3. Steady-state rolling step;

4. Transient dynamic analysis step.

The first two steps are partly similar to the ones as explained in the low frequency modal analysis. The only difference is that the frequency extraction and steady state dynamic step are not performed. Hence the simulation is run up until the *footprint force controlled step*, as can be seen in figure 4.2. The latter of the two steps, 3 and 4, are uniquely set-up for this transient analysis and will be briefly covered here:

Steady-state rolling step: Prerequisite to the transient dynamic analysis step is to have a rolling tyre, which is obtained with the steady-state rolling analysis step. The main purpose of this step is to find the *free-rolling solution* by matching the angular velocity of the tyre with the velocity of the road surface (translation) or of the drum (angular). If these velocities are correctly matched, zero torque will be apparent at the axle. The main difficulty lies within the fact that these velocities cannot be readily matched because the exact deformation of the tyre at the footprint is not known beforehand. Consequently leading to a slight difference in radii, leading to an unknown circumferential length. Therefore to successfully determine the free-roll velocity an indirect iterative procedure is used, which is solved with the implicit Abaqus/Standard solver.

Transient dynamic rolling step: When the free-roll solution has been found the results can be imported into an Abaqus/Explicit simulation to analyse its transient response. Beside that the existing road surface or drum have to be added or modified to include some sort of irregular geometry, such as a cleat. Running the explicit transient dynamic rolling simulation will yield the time-force response of the forces acting on the tyre.

Finally the obtained force data in the time-domain, defined as the impulse response function (IRF), will have to get post-processed by transforming this IRF data to the frequency domain using a fast Fourier transform. This will finally yield the frequency response function as described in section 3.3.1.

4.2 Pre-processing

The pre-processing phase of the simulation procedure contains all the preparation of the data that is required to run the analysis. The end result of this phase is in the form of an Abaqus text-based input file, which contains all of the data, setting and simulation steps, ready to be run. This input file can be prepared in two ways:

- Within Abaqus/CAE, the guided Graphical User Interface (GUI) of Abaqus, where CAE stands for **C**omplete **A**baqus **E**nvironment, an obvious backronym of Computer Aided Engineering.
- Completely text-based, meaning that every data set, setting and simulation step is inserted manually. In reality often another existing input file is used as a template to form a new input file.

The latter one is mostly used within Apollo for its transparency and simplicity, and is also the one best suited for automation.

4.2.1 Geometry generation

The basis of every simulation is the geometry of the structure on which the analysis will be performed. For tyre simulations performed in this study, the starting geometry data is a 2D CAD drawing of the tyre's cross-section. Since a discretized geometry is required, a mesh generation of the geometry has to be performed. This can be done with the build-in meshing tool of Abaqus itself or with a third-party software like Altair HyperMesh.

The 2D CAD drawing contains all the different components as the ones described in section 2.1.2, with all the inner reinforcement layers. To properly mesh the tyre's cross-section, the geometry is divided in three layers: the outer geometry layer, the inner components layer and the reinforcement layer. All these layers together will finally yield the meshed tyre cross-section which will be used for the simulation.

Mesh distribution

To ensure a well distributed and symmetric mesh with a proper aspect ratio, a pre-division of the geometry is inputted to the meshing software. These pre-division lines serve as meshing guidelines which are used to form the final mesh. For meshing Altair HyperMesh is generally used within Apollo. This CAE mesh tool has specifically designed add-ons to mesh a 2D cross-section tyre structure.

Rotational symmetry

Not regarding a complex 3D-tread, a tyre is rotationally symmetric around its centre axis. Hence by creating a 2D cross-section of the tyre containing all its internal structural components and revolving it around its centre axis generates a 3D-model of an entire tyre. Since 3D-tread simulations are much more complicated and time consuming, and not very useful for low frequency modal tyre analyses, these type of simulations will not be conducted during this study.

Required segments for revolution

The revolution of the tyre around the symmetry axis is done in a discrete manner, meaning that a finite amount of *segments* are used to form the 3D tyre model. The more segments are used the more accurate the model becomes, however this comes at a cost, namely computation time. Since there are several subsequent steps involved with the analysis procedures the computation time increases drastically with each additional segment. Hence there is a compromise to be made between computation time and accuracy.

For modal analysis one of the most decisive thing is the ability to properly describe vibrations of higher orders mode shapes. This means that enough elements should be present to have sufficient nodes and elements to geometrically be able to form said mode shapes. Therefore it can be stated when not enough segments are used higher amount wavelengths become indistinguishable, meaning that these higher wavelengths cannot be correctly identified. In signal processing there is a very resembling effect that can occur, there it is called *aliasing*.

In practice using an insufficient amount of segments leads indeed to wrong identifications some these higher mode shapes. More importantly it also introduces so-called *junk* modes, which basically are wrongly identified modeshapes which occur due to numeric coarseness of the mesh. To get an idea what actually happens when using different amounts of segments, some preliminary simulations have been run using the practice simulation files, table 4.1 shows the end results. This table clearly shows that indeed while increasing the amount of segments used, the computation time drastically increases. One other interesting thing is the lower the amount of segments used, the higher the number of eigenfrequencies recognized. This confirms that when an insufficient amount of segments are used these so-called junk modes are identified wrongly.

Number of segments	Number of eigenfrequencies	Computation time [sec]
40	71	642
80	44	517
120	42	1190
240	42	2399

Table 4.1: Simulations run with different four different amount of segments. For the eigenfrequency extraction the Lanczos eigensolver is used with a frequency range between 15 and 250 Hz (N.B. This simulation is run without the steady-state dynamic analysis step).

To properly set the amount of segments while being computationally efficient a sort of trail-and-error process has to be performed. This process should be accompanied with the widely known rule of thumb of six to ten elements per wave length and the highest wave number that occurs until within the specified frequency range. Hence by simply multiplying the highest wave number with 6 to 10 yields an estimate of the amount of segments that are required.

4.2.2 Material models

For tyre simulations either hyperelastic or viscoelastic materials models are used. The following sections will briefly elaborate upon these two models and also the damping will be covered.

Hyperelastic models

Initial design iterations mostly use linear secant moduli of the material for 5-10% strain-levels. For improved stability and accuracy the nonlinear hyperelastic models have to be used, which are based on experimental values, obtained by material characterization testing, such as uniaxial, planar or equibiaxial tension tests.

Due to very nonlinear behaviour of rubber nonlinear material models have to be implemented to be able to approximate the real world behaviour of rubber. There are several models which can describe the behaviour, varying from rather simple models to moderately complex. Each of these models have their benefits depending on the amount of test data available and the amount of nominal strain that will occur during simulation.

Over the years several different material models that have been developed, polynomial models either based on strain invariants or on stretches and statistical mechanics. Here is a short overview of these models with a brief overview [3, 6]:

- Neo-Hookean: simplest form which is always stable – based on one strain invariant (I_1). Fits only the low strain area and is inaccurate for higher strains.

- Mooney-Rivlin: widely used form which can be unstable – based on two strain invariants (I_1 and I_2).

- Yeoh: Higher nonlinear form which is always stable – based on one invariant (I_1). Gives a better fit got higher strains.

- Marlow: Recently developed material model which is dependant on the first strain invariant.

- Ogden: based on stretches rather than strain invariants – can be unstable.

Viscoelastic models

The viscoelastic behaviour of rubber comprise of time-dependent behaviour such as creep, stress relaxation and hysteresis. Since these are dependent on loading, the temperature and deformation rate, implementation of the nonlinear models of these phenomena are therefore very complicated. Therefore linear viscoelastic models are usually implemented, one commonly used model and also the one mostly used by Apollo is the so-called Prony series (equation 4.1).

$$E(t) = E_\infty + \sum_{i=1}^{N-1} E_i e^{-t/\tau_i} \quad \text{for} \quad N = 1, 2, ... \tag{4.1}$$

Where:
E_∞ is the long term modulus [N/m^2];
E_i and τ_i are the material dependant coefficients.

To determine the Prony series coefficients a number of test are to be conducted to obtain a good fit of the material for the range in which the simulation will be done. These coefficients then can be used within Abaqus to simulate the rubber compound specific material behaviour.

Damping

Damping dissipates energy, therefore causing the amplitude of free vibration to decrease over time, and limiting amplitudes of vibration that is induced by harmonic loading when its frequency coincides with structures natural frequencies [9]. The damping can be inherently in a structure, and/or added deliberately. Different types of damping exist, which are categorized as follows:

- *Viscous damping* force is proportional to the velocity of the structure.
- *Hysteresis damping* is damping that is inherent to the material.
- *Coulomb damping* is damping that is dependant on dry friction, e.g. slippage in joints.
- *Radiation damping* is the kind of damping when energy is loss to a particular 'unbounded' medium such as soil.

Of the aforementioned physical damping mechanisms, viscous damping is the only type that is easy to represent in dynamic equations. Luckily, the relative force that is generated by damping in dynamic structural problems is usually only about 10% or less compared to the other forces [9, p. 389]. There are two general ways to introduce viscous damping to a dynamic problem, being *proportional damping* or *modal damping*. Only the former type of damping will be elaborated further as this is the one that is going to be implemented in this study.

Proportional damping

This damping device, also known as Rayleigh damping, defines a global damping matrix as a linear combination of the global mass and stiffness matrix, which is expressed with this equation:

$$[\mathbf{C}] = \alpha[\mathbf{M}] + \beta[\mathbf{K}] \tag{4.2}$$

Where:
[**C**] is the global damping matrix;
[**M**] the global mass matrix;
[**K**] the global stiffness matrix;
α and β are the proportional mass and stiffness Rayleigh damping coefficients respectively.

The damping is frequency dependant due to the damping's matrix dependency on the global mass and stiffness matrix. Beside this, it is important to note that the $\alpha[\mathbf{M}]$ contributes the most to the damping at the lower modes, while $\beta[\mathbf{K}]$ contributes the more heavily to the higher frequency modes.

4.2.3 Step definition

This section will describe the different options for the eigenfrequency extraction step and the steady-state dynamic step.

Eigenfrequency extraction step

For the frequency extraction there can be chosen between two different eigensolvers, namely the Lanczos eigensolver or the automatic multi-level structuring (AMS) eigensolver. The main advantage of AMS is that it should have a performance advantage over the Lanczos eigensolver, especially when a large amount of eigenmodes have to be extracted of a system with many degrees of freedom. Since this performance improvement is not really significant for low frequency tyre analyses, the Lanczos eigensolver is chosen for the frequency extraction step.

Besides, the AMS eigensolver requires more Abaqus license capacity than the Lanczos eigensolver when less than 8 CPU cores are used. Abaqus license capacity is defined with the amount of license tokens a Abaqus simulation job uses, an overview of the amount of license tokens required can be seen in table 4.2.

Steady-state dynamic step

The steady-state dynamic (SSD) analysis is used to extract the dynamic linearized response of a system to a harmonic excitation, which is done with a linear perturbation procedure. The system is excited with a force amplitude of 4% of the tyre's rated force. The SSD analysis can be performed in three different ways:

	Number of CPU cores					
Abaqus product	1	2	4	8	12	16
Abaqus/Standard	5	6	8	12	14	16
Abaqus/Standard (with AMS)	10	10	10	12	14	16

Table 4.2: Number of licence tokens required to run a Abaqus analysis job, which depends on the type of analysis run and the number of CPU cores used to run the job.

- Direct: direct calculation of the response in terms of all the degrees of freedom of the model.
- Subspace-based: calculation done on a projection of the dynamic equations onto a subspace of modes.
- Mode-based: here the calculation is based on the system's eigenfrequencies and modes.

Beside the steady-state solution procedure, the type of frequency interval must be chosen, where the interval states at which frequencies the output has to be calculated. These are the two available options:

- `EIGENFREQUENCY`: intervals determined with previously determined values for the eigenfrequencies. The `FREQUENCY SCALE` is usually set to linear, for which the range has to be defined, and the amount of steps between the intervals (if the step in Hz is very small some step may be skipped).
- `RANGE`: user defined intervals as defined by the `FREQUENCY SCALE`, either linear or logarithmic.

Due to the significant damping that is present in the model the direct-solution steady-state dynamic method is used. This is done since this model is the most accurate of these three methods with the conditions that these tyre analyses have. The only downside is that this method does take more time compared to the other methods.

4.3 Processing

After the simulation jobs have been properly set-up the actual simulations performed to obtain the results. This part of the procedure is called the *processing* phase. Simulations are initiated through the command line interface (CLI), within Windows this done in Command Prompt and in Linux with Terminal. To initiate a job the following rule of syntax has to be executed while in the folder where the actual job-files are located:

```
abq6131 job=<filename 1> oldjob=<filename 2> int cpus=<CPU cores>
```

Where:
`<filename 1>`: Is the job input filename without the file-extension;
`<filename 2>`: Is the job input filename of the prerequisite job that has been run already, also without the file-extension;
`<CPU cores>`: Are the amount of CPU cores that the script is allowed to use, inputted as an integer.

Once the job has been initiated, Abaqus will start running the simulation in the background and output some status messages in the CLI. Once completed Abaqus will output whether or not the simulation has run successfully. Either way, several files will be created in the folder which hold information or data. These are the following files of interest:

`.odb`-file: This is the Abaqus database file where all the analysis results are stored, hence this is the main file that will be used to post-process the data.

`.dat`-file: This data text file contains all the information of the model, including some tabular overviews of the results. Any errors or warnings that show up during the processing of the input data will also show up here.

`.msg`-file: This message text file show all the information concerning the progress of the simulation itself. Any errors or warnings that occur during the simulation are also put in the message file.

`.sta`-file: The status text file shows the incremental progress of the simulation and is update every time an increment of the analysis is completed during the simulation. This makes this file very useful to monitor the simulation progress during execution.

Other files: The remaining files are mostly auxiliary files and files that are for example needed in a subsequent simulation part is to be performed. This is for example the case when using the 2D-axisymmetric results to create the 3D-model for the 3D-revolved simulation part.

To be able to queue jobs so that they run subsequently one after the other simple queue-scripts can be written. This makes it easier to run multiple jobs outside of work hours, hence using the license tokens more efficiently.

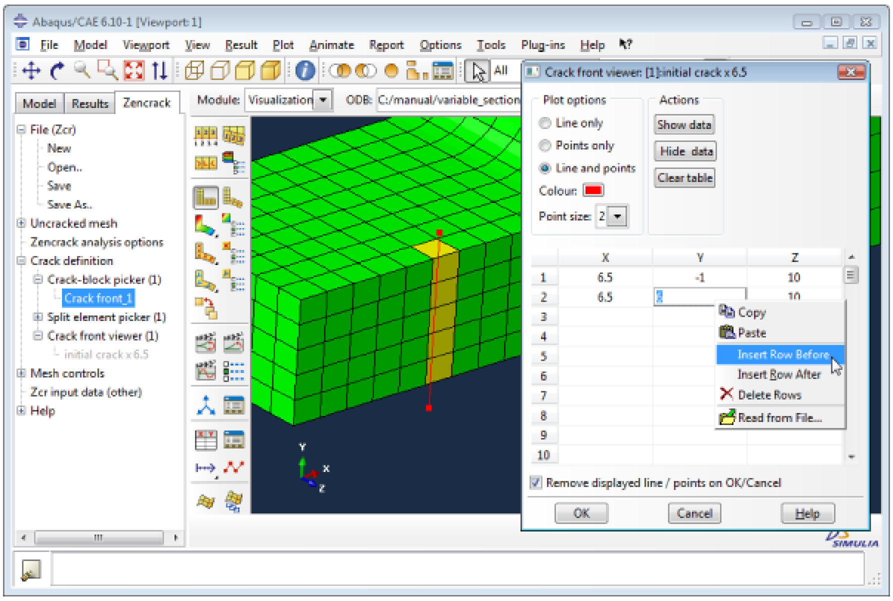
4.4 Post-processing

The last thing remaining in the simulation procedure after the model has been solved is the post-processing phase. It first begins with thoroughly checking whether any problems have occurred during the solution process. For Abaqus this means checking the `.sta`-file and `.msg`-file (section 4.3) for any issues and warning or error messages. Secondly the model has to be checked visually whether elements, contact nodes/patches or the displacements and loads do not show any abnormal behaviour. When the analysis solution has been verified, the result quantities of interest can be examined. For further explanation purposes this step will be referred to as the *model verification* step.

Having large and/or multiple sets of data often makes it inherently difficult to analyse and compare these data sets. Therefore after simulation analyses the data is often post-processed to obtain more tangible results. This, though not exclusively, is often one of the following things:

- Further processing of the data, meaning that the data is manipulated to make it more readable. For example this could simply mean calculating averages, focusing on specific ranges, filtering the data with certain filters, et cetera.
- Graphically visualizing the data with graphs or diagrams, which often is a much more readable presentation of the data if done correctly.

This part of the post-processing will be referred to as the *model data evaluation* step. Abaqus/-CAE has post-processing built-in and is known as Abaqus/Viewer, visually the program looks like the image in figure 4.3. This allows the user to process, visualize and compare the data within the Abaqus/Viewer GUI program. Within this environment the *model verification step* is also performed, as explained in the first paragraph of this section. To assess the results from either the low frequency modal analysis or high frequency transient analysis several key results have to be analysed, the following sections will elaborate further on this.



Abaqus/CAE 6.10-1 [Viewport: 1]

Module: Visualization

ODB: C:/manual/variable_section

Crack front viewer: [1]:initial crack x 6.5

Plot options

○ Line only

○ Points only

● Line and points

Colour: 🟥

Point size: 2

	X	Y	Z
1	6.5	-1	10
2	6.5		10
3			
4			
5			
6			
7			
8			
9			
10			

☑ Remove displayed line / points on OK/Cancel

Figure 4.3: Abaqus/Viewer (CREATE OWN IMAGE).

4.4.1 Low frequency modal analysis

For the low frequency modal analysis the key results of the 2D-axisymmetric and 3-revolved simulation parts have to be analysed and compared to assess how the tyre performs. Predominantly these are steps that are already required by the procedure as currently defined in Apollo.

2D Axisymmetric (rim mounting and inflation)

To evaluate the whether the rim has been correctly fitted on the rim and properly inflated some checks are required. These are the post-processing steps that have to be done to do the assessment after the 2D-axisymmetric analysis:

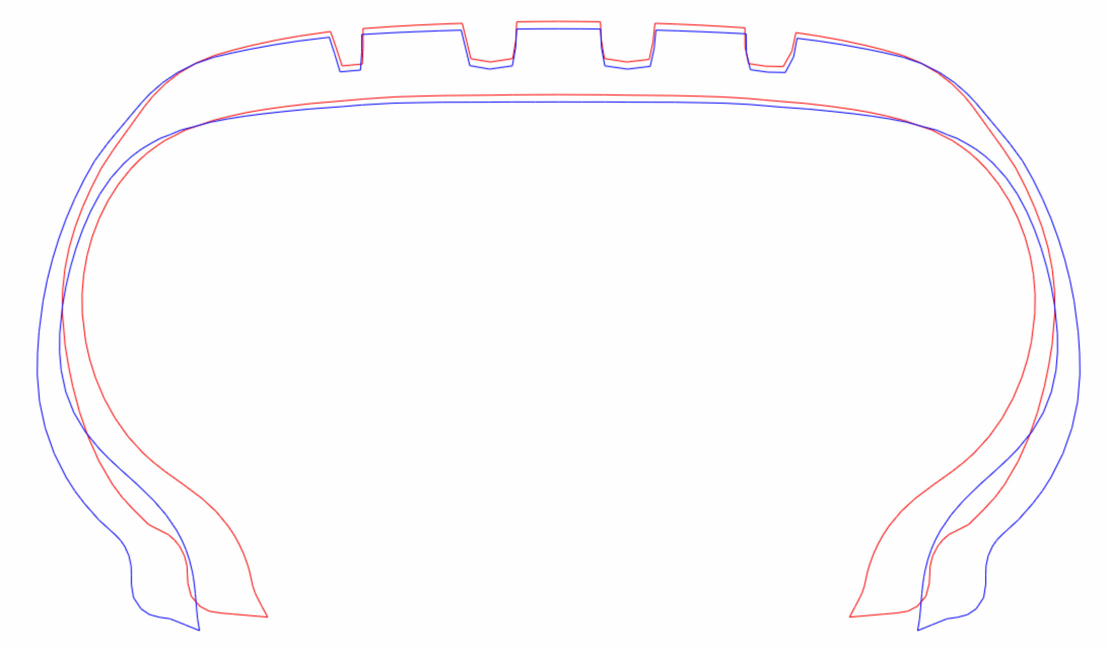


Figure 4.4: Mold vs inflated contour (Apollo Alnac 4G 195/60R15).

- Mold versus inflated contour: This checks how the tyre has deformed after inflation and whether the requirements of shape and deformation have been met. To do so firstly an image showing both the before and after inflation cross-section has to be visually inspected, an example of which can be seen in figure 4.4. Secondly the actual numeric values of the width and outer diameter of the fully inflated tyre have to be obtained.
- Rim-tyre fit: To check the tyre-rim fit after inflation, the stress distribution of the tyre-rim contact can be analysed.

3D Footprint

After the revolved 3D tyre model simulation has been run with both the displacement and force controlled footprint loading, the final footprint has to be analysed. The following steps have to be done:

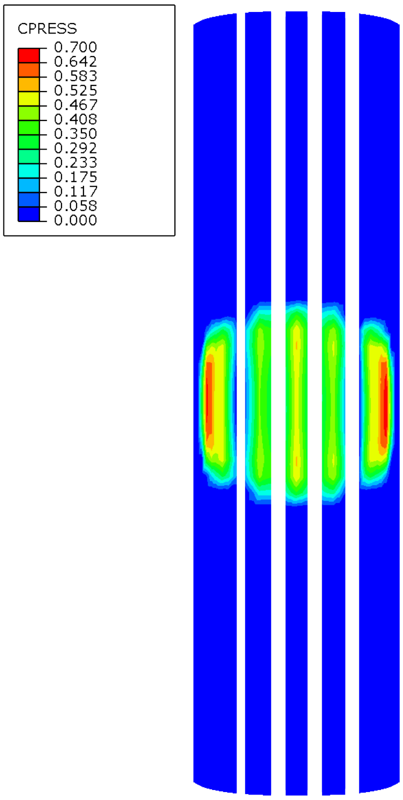


Figure 4.5: Footprint (Apollo Alnac 4G 195/60R15).

- Footprint contact shape: a 2-dimensional plot of the stress distribution of the contact patch of the tyre with the road. The quality of the footprint has to be evaluated by investigating the shape of the footprint and the actual stress distribution within the contact patch. This way it is checked if the tyre's construction has proper contact with the road. Figure 4.5 shows an example of such an image, for clarity the road surface and part of the tyre model has been hidden, only the tread element set is shown.

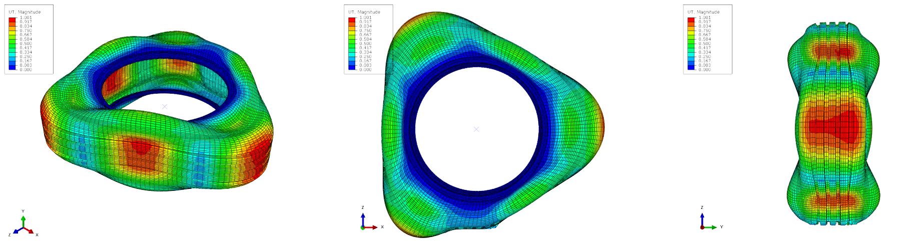
- Footprint dimensions: The length and width of the footprint has to be approximated with a node-to-node measurement. Besides this, the contact area also has to be obtained from the output database (logged under the value `CAREA`, which stands for contact area). These values are for comparative purposes to see the differences between different tyre designs and set-ups.

Eigenfrequency extraction

Of the eigenfrequency step the following steps are to be performed to analyse the results:

- 2D-plot of the mode numbers plotted against their eigenfrequencies. When plotted together with other simulations of tyres of similar type and size but with slight design differences this plot shows how the tyres overall stiffness compare to each other. A tyre with a higher overall stiffness is very likely to show a steeper increase in the frequency values of the mode shapes.

- A qualitative visual inspection and comparison of the mode shapes of interest. For this the resultant displacement data of the entire tyre is visualized per mode shape, of course the actual frequencies of these mode shape themselves are also taken into account. An example of a mode shape can be seen in figure 4.6, which shows the mode shape in three different views. Often these mode shapes are also given unique labels with which the nature of the shape can quickly be identified, section 4.4.3 will describe the labelling convention which will be used in this study.



(a) Isometric view

(b) Side view

(c) Front view

Figure 4.6: Mode shape 11 of the Apollo Alnac 4G 195/60R15 tyre, which is a radial mode shape with wavenumber 3.

Steady-state dynamic analysis

Finally from the steady-state dynamic (SSD) analysis step the following things have to be checked:

- To see at which frequencies the tyre is likely to generate and transmit unwanted vertical forces to the vehicle the frequency response function (FRF) data is investigated, which are obtained with the SSD analysis. These values give an estimation of the drive comfort as perceived by the driver and passengers of a car and are therefore important to assess. Especially the peaks in the FRF data plots are points of interest as these show increased response in force at that specific frequency.

- To compare different tyres with each other root mean square[1] (RMS) values are calculated of frequency ranges of interest. At glance these RMS values give insight which tyre's net results show a higher force response in that frequency range, making it easier to compare tyres with each other. For representation, the RMS-data can be shown in a bar chart.

[1] To get a better insight on how tyres compare to each other based on the FRF-data from the SSD analysis, the **root mean square (RMS)** values of specific frequency ranges are calculated. RMS, also known as the quadratic mean, is a statistical measure defined as the square root of the mean of the squares of a sample [4].

4.4.2 High frequency transient analysis

Since this study mainly focuses on the low frequency modal analysis and I did not have the time to get familiar with the transient analysis procedure a very brief and superficial description of the transient analysis post-processing part will be given.

As described in section 4.1.2 the first two simulation steps are similar to the ones as performed in the modal analysis. To generally the same data could be requested and evaluated, except for the modal data of course since these steps are not performed with transient analyses. This leaves the last two steps:

Steady-state rolling step: From this step the resultant torque on the rim halves could be checked at *free-roll* velocity to verify if indeed the correct free-roll solution has been found.

Transient dynamic analysis step: Having obtained the impulse response function (IRF) time domain data from the simulation this data first has to be verified if it shows the expected behaviour. Next, this IRF data has to be transformed to the frequency domain using the fast Fourier transform (FFT). After this is done the obtained FRF data can be post-processes similarly to that of the *steady-state dynamic analysis* data. This means investigating the FRF-data visually and calculating the RMS-values from this data.

4.4.3 Mode shape labelling convention

Tyre mode labelling has never been really standardized, hence leading to a lot of different tyre mode labelling conventions throughout the literature. This study will use the tyre labelling convention as proposed by Wheeler et al. [7] and an extension thereof, namely the one as introduced by Kindt [5, p. 63]. Wheeler's unambiguous tyre mode labelling convention for *unloaded* tyres consists of two numbers and is formatted as (n,a). In this notation, the indices represent:

n = circumferential index;

a = belt cross-sectional index.

Here the circumferential index (n) is the amount of bending wavelengths of the belt in the radial direction of the tyre. While the belt cross-sectional index (a) stands for the amount of bending wavelengths in the lateral direction. Figure 4.7 shows visual representations of these indices. It must be noted that this notation does not unambiguously encapsulate the mode shapes where the belt translates or rotates rigidly. For these mode shapes labels such as *axial* and *torsional* are used to specify their rigid nature.

Because of the axisymmetric nature of an unloaded tyre most of the mode shapes appear twice, which is caused by double system poles. When a tyre is loaded however it is no longer axisymmetric, leading to a split of these double poles into two single poles. Nonetheless these modes are still related to the same mode of the unloaded tyre, thus having the same amount of wavelengths in the radial and lateral direction. Kindt's extension to Wheeler's convention are

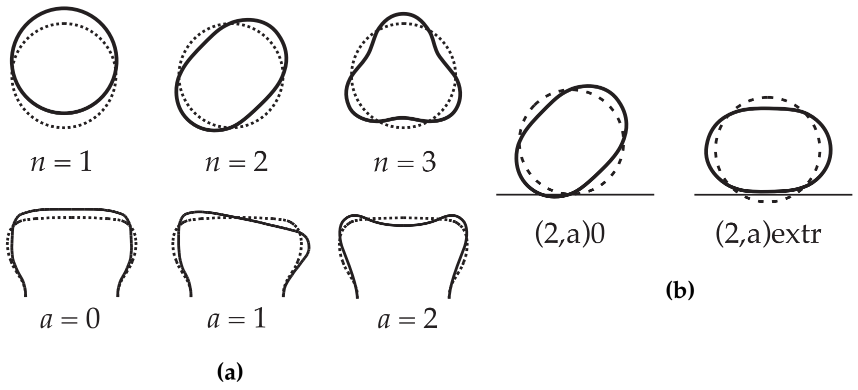


Figure 4.7: Visual representation of the mode shape labelling conventions indices (a) as proposed by Wheeler et al. and additions (b) as introduced by Kindt (illustration courtesy of Peter Kindt, obtained from his doctoral dissertation [5].).

additional suffix labels which distinguished these to single poles from each other. These are the suffix labels as introduced by Kindt:

0 This first additional mode shape identifier is used when the radial displacement at the circumference/tread is zero in the middle of the tyre-road contact (left image in figure 4.7b).

extr This label is added when the displacement at the middle of the tyre-road contact is at an extremum (right image in figure 4.7b).

Lastly there are two special cases of mode shapes of which the above described convention is ambiguous. These special cases are for the (1,0) and (1,1) modes in which the belt rotates or translates rigidly. Therefore the following to labels have been introduced by Kindt:

vert This label added when for the (1,1) and (1,0) mode the belt translates along and rotates about the vertical tyre axis, respectively.

hor Instead of rotating about or translating along the vertical axis, this label indicates that it is about/along the horizontal axis for the (1,1) and (1,0) mode.

This concludes Wheeler's version and Kindt's extension of the mode shape labelling convention.

Chapter 5

Automation of analysis procedure

As mentioned in the introduction of this report, the main goal of this assignment is to automate and partly standardize low frequency tyre vibration analyses, which are performed in Abaqus. Chapter 4 gives an overview of this procedure and what it is comprised of. Currently within Apollo (mid 2015) the *platform* project has been kicked off, which has as main goal to create a standard 'template' of a tyre design. This template consists of a standardized guided design procedure of a tyre, accompanied with the required tooling to develop new tyre designs more efficiently.

For the pre-processing, processing parts and data management of the simulation procedures there are already plans and ideas to automate this, which are:

Athena Framework: This open source application framework will likely form the basis of the database in which all the data from run simulations will be stored.

LMS Virtual.Lab: This software package will realize automation on the pre-processing and processing part of the simulation procedure. Basically Virtual.Lab allows a user to do a simulation request for a specific tyre, where after Virtual.Lab will automatically generate the job and run it when possible.

A short disclaimer on the automation plans within Apollo: Since I am not very familiar with the exact specifics of the *platform* project, and the plans concerning the automation of the pre-processing and processing part within LMS.Virtual lab the above made statements may not be entirely correct.

This leaves the post-processing phase of the analysis procedure uncovered, which coincidentally has become the main focus of this assignment. The post-processing can roughly be divided into two steps, being the *model verification* step and the *model data evaluation* step, as have been introduced in section 4.4. The model verification step mostly consists of several checks that are mostly visual and therefore have to be performed by the user. This means that this step cannot be readily automated as these checks are very dependent on the model itself. The *model data* *evaluation* part of the post-processing procedure however can be readily automated, since output parameters which have to be evaluated are controllable within the model definition without any difficulty.

Currently most of the evaluative post-processing steps are performed within the Abaqus/CAE GUI application and mostly consist of many repetitive tasks, which can become very tedious. Hence there is a great incentive to improve the evaluative post-processing procedures, which will be mostly addresses within this assignment.

5.1 Automatization overview

Abaqus/CAE has build-in scripting language that allows the users to automate almost everything that is performable by users in the Abaqus/CAE GUI application itself. The scripting language used is **Python**, a general-purpose high-level programming language. This basically allows almost the entire post-processing part to be automated within Abaqus, since most of the graphs required can be generated in Abaqus itself.

The main disadvantage to this however is that the use of Abaqus/CAE requires license tokens, and these are limited. An alternative way is to use third-party application to do part of the post-processing steps instead of using Abaqus/CAE. When deciding to do part of the post-processing steps outside Abaqus/CAE, the *raw*-data and images have to be obtained from the Abaqus result database file (`.odb`-file), which still would requires tokens. It can be concluded however that by choosing to externally do part of the post-processing, much less intensive use is made of the Abaqus license tokens. To summarize, this means that the *model data evaluation* is split into two separate parts, being the **data extraction** part were the data is extracted from Abaqus results database file and the **evaluative post-processing** part where the data is further processed to actually evaluate the data.

Of course there are countless programming languages and software packages in which part of the evaluative post-processing can be performed in. Since a great majority of these are not free or require too much experience to properly develop. Therefore only the most readily available and accessible will be examined, which are considered in this case to be either MATLAB or NumPy.

MATLAB

Compared to NumPy, employees at Apollo are generally more familiar with MATLAB and since it is already used within Apollo, MATLAB is a good option which mostly accommodates all of the needs for post-processing adequately. There is also a graphical user interface environment available within MATLAB, combined with the MATLAB Compiler this allows a standalone application to be created. Compared to NumPy, MATLAB has the following advantages:

- Generally easier to use, some people at Apollo are already familiar with MATLAB and it is easier to learn and understand due to a less steep learning curve.
- Hassle free installation when compared to the NumPy installation procedure.

NumPy

Numpy is an open source extension of the Python programming language, which adds support for multi-dimensional arrays and matrices, combined with a large library of mathematical functions. In a sense it does resembles quite a bit to MATLAB's functionality, although it does not have a dedicated editor application environment like MATLAB has. Compared to MATLAB, NumPy has the following main advantages:

- Free to use open source extension, hence no license required.
- Possibility to integrate directly into Abaqus data extraction automation script since both use Python code.
- Possibly more capabilities due to many libraries and extensions.
- Programming language with both better higher and lower level programming capabilities, meaning better object-oriented programming and faster performance.

MATLAB vs NumPy

As discussed in the preceding two paragraphs both methods have their pros and cons. After considering both it is finally concluded that MATLAB is the best choice of the two. One of the key reasons for this is that MATLAB is already used within Apollo, this allows other people at Apollo in the future to use and alter the application with much less effort. Secondly the MATLAB software deployment is much more straightforward, this is a great plus for MATLAB especially when compared to the installation of the Python runtime packages, which requires a lot of manual installations and adequate user experience with the CLI environment.

5.2 Python scripts

The data extraction part still requires the use of Abaqus/CAE since the data has to be read from results database (`.odb`) file. The only way to do this is with Python as automation scripting language, as this is the default scripting language used within Abaqus. Beside the data extraction script an additional Python script has been written, which is an Abaqus job queue script.

5.2.1 Queue script

The queue script is written to improve the way Abaqus simulation jobs are queued for execution. The script has been written in Python and has to following capabilities:

- Partial automization of the queue process;
- Preliminary check if all the queued jobs are correctly inputted and if all the input-files exist;
- Confirmation query if to run all the jobs are shown in the overview.

The main goal of this extra automation script was to improve the way Abaqus simulation jobs are queued, which currently is done with simple python scripts that have to be set-up completely manually. Appendix C will give an explanation on how to set-up a simulation batch queue within the `queue_script.py` file and how to execute it through a command line interface (CLI) accompanied with an example.

The two main advantages of the automation script is the easier queue script set-up and the simulation job conformation query. The latter one prevents user input errors as the user can immediately check whether he or she has queued the jobs properly. Mistakes made can mean that the Abaqus license tokens are used less efficiently.

5.2.2 Data extraction

To accommodate for an easy, fast and consistent way of acquiring the data from the simulation results database file a data extraction script was made. The key features of this script are as follows:

- Effortless automatic extraction of multiple data files in one run.
- Flexibility and compatibility of the script to accommodate for most of the simulation set-ups.
- Ability to run it without the Abaqus/CAE GUI application active, meaning that is run on the background without any visual feedback. Since it does not have to render anything for the GUI the processing speed is increased significantly.

The Python script is executed through the CLI and outputs all the data into a sub-directory which the script makes by itself. The data output of the queue script is as described in section 4.4, to summarize this an overview of the visual and numeric data is given here below.

Visual data

All the visual data images are stored as `PNG`-images, which are:

- Mold vs inflated contour image;
- Mode shape images (isometric, front and side view);
- Footprint contact pressure image.

Most of the images come with a legend for the displayed quantity.

Numeric data

The numeric data is stored in tab-separated `txt`-files, being:

- Contact are values;
- Eigenfrequency values;
- Frequency response force values;
- Coordinate locations of the (un)loaded tyre's symmetry nodes;
- Displacements of the tyre's symmetry nodes of every mode shape.

All the numeric data columns have a header item which describes what data is stored in that column.

Data extraction workflow

Prior to running the script through the CLI, the user must check several input and output settings and set specific run options to get the desired output of the required result database files. To get an idea on what is possible and how to use it, please refer to appendix section B.3. This appendix will give a much more practical explanation of the data extraction script.

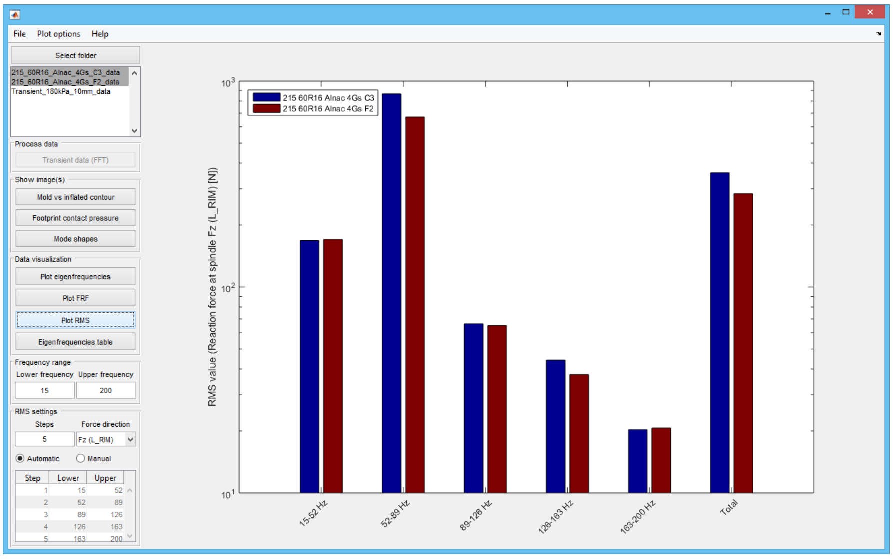
5.3 MATLAB GUI application

As stated the **evaluative post-processing** part has been automated with MATLAB and its GUI capabilities. The general idea behind this is to make the evaluative post-processing part of the data easier by combining everything in one application. Within this application the user is able to easily analyse one specific simulation, or compare multiple simulations with each other. Using the MATLAB Compiler a standalone application is compiled, meaning that users do not require to have an active MATLAB installation and license on their computer. The only requirement to run such a standalone MATLAB application is having the MATLAB run-time environment files installed on the computer at hand.

One of the main goals during the development was to create a robust and intuitive application, which fulfils all of the evaluative post-processing needs of modal analyses. Beside this, this application also had to work properly with different window and screen sizes, meaning that the application itself has to scale properly. The following sections cover all the features of the application itself, section B.4 of the user manual practically describes on how to use the program. The final application has been named the **Tyre Vibration Toolbox**.

5.3.1 Features

All of the added post-processing features are dictated by the modal post-processing steps as described in section 4.4.1. Additionally extra functionality has been added such as multiple data select and export capabilities, which are added to enhance the practical use of the application. Figure 5.1 shows an image of the final Tyre Vibration Toolbox application to get a general idea of what it looks like.



215_60R16_Alnac_4Gs_C3_data
215_60R16_Alnac_4Gs_F2_data
Transient_180kPa_10mm_data

Frequency range
Lower frequency: 15
Upper frequency: 200

RMS settings
Steps: 5
Force direction: Fz (L_RIM)
● Automatic ○ Manual

Step	Lower	Upper
1	15	52
2	52	89
3	89	126
4	126	163
5	163	200

Figure 5.1: Tyre Vibration Toolbox with the RMS plot active.

Multiple data select

One of the key aspects of the tool is the ability to select multiple data sets at once, allowing the user to quickly and efficiently asses different tyres with each other. For example when selecting multiple data sets and letting the application automatically calculate and plot these RMS-values gives a quick insight how these different tyres perform in the specified frequency ranges.

RMS values calculation

The application automatically calculates the root mean square (RMS) values of the force FRF data, using the specified frequency ranges. These frequency ranges can be modified by the user, this can be done in two ways:

- Semi-automatic: Here the user can only choose the amount of frequency range steps. Using the set amount of steps, the entire frequency available frequency domain is divided over the amount of steps.
- Manual: With the manual way the user can set the frequency ranges entire manual, this means that the available frequency domain can be divided non-uniform ranges.

Changes made within the RMS settings are directly reflected in the data, meaning that the graphs are updated after every change.

Data visualization

The data visualization feature allows several graphical representations of the data to be generated in the application. These are data visualizations that can be performed:

- The eigenfrequency data can be displayed in a line chart where the mode number is on the horizontal axis and the frequency on the vertical axis.
- The force FRF data can be displayed in a line chart as well. Here the frequency is on the horizontal axis and the force on the vertical axis. Additionally the RMS ranges are also visually added to the FRF data plot, this allows the user to visually check the set ranges.
- The RMS values are visualized using a bar chart representation, an example of which can be seen in figure 5.1.

Image viewers

Within the application it is also possible view the available images of the simulation data sets. There are three different image types, these are:

- *Footprint contact pressure image*: An example of this image can be seen the previous chapter, figure 4.5. Within the application it is possible to view two footprint images side-by-side to be able to adequately compare multiple footprints.
- *Mold vs inflated contour image*: The mold vs inflated contour image example can be seen in figure 4.4. Similar to the footprint image these images can be viewed side-by-side for comparison sake.
- *Mode shape images*: Due to the multitude of mode shapes per simulation, and the three different mode shape views per mode shape, there are a lot of available mode shape images per data set. To efficiently compare the mode shapes with each other a two-by-two image viewer has been created which allows the user to compare 4 different simulations with each other at once. Within the image viewer the user can change which mode shape has to be displayed and with what view.

Other features

Other than the main features, there are some miscellaneous features that have been added:

- *Export capabilities*: This allows the user to export images and numeric data of the graphs that have been generated within the application. For example the user can export the RMS bar chart as an image for presentation purposes.
- *Import options*: The import options have been added to enable some flexibility on how the data is imported into the toolbox.
- *Plot options*: Allows the user to turn change several options concerning the graphs that are active at that moment.

- *Eigenfrequencies table*: Beside the graphical data visualization also a tabular overview of the eigenfrequencies is available.
- *Mode shape recognition*: This extra feature automatically recognizes the mode shapes and adds them to the eigenfrequency table. A more detailed explanation is given in the following section B.A.

This concludes all the features of the Tyre Vibration Toolbox. More practical information can be found the user manual of the application, located in appendix B.4.

5.4 Mode shape recognition

As said, the mode shape recognition is an algorithm that automatically labels the mode shapes, and is primarily developed as a *proof of concept*. The labels used are a simplified version of Kindt's labelling convention, as explained in section 4.4.3. The input of the algorithm consists of coordinate and displacement data in the x, y and z-direction from the symmetry node-set from the simulation model, these nodes are highlighted with red dots in figure 5.2. This data consists of

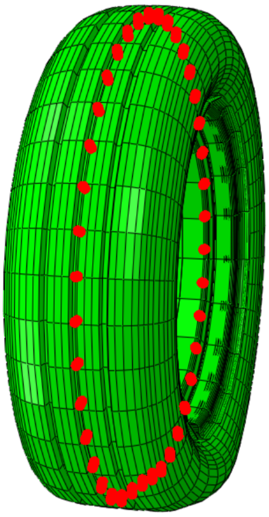


Figure 5.2: Symmetry nodes location on a 3D tyre model.

- the reference coordinate locations of a tyre showing no mode shape (coordinate-data);
- and the relative displacements in all three directions per mode shape (displacement-data).

These coordinate and displacement data sets of two mode shapes are visualized in figure 5.3 and 5.4. The yellow arrows in these figure indicate the direction and magnitude of the displacements of the nodes. For clarity the length of the arrows have been magnified. Looking at both

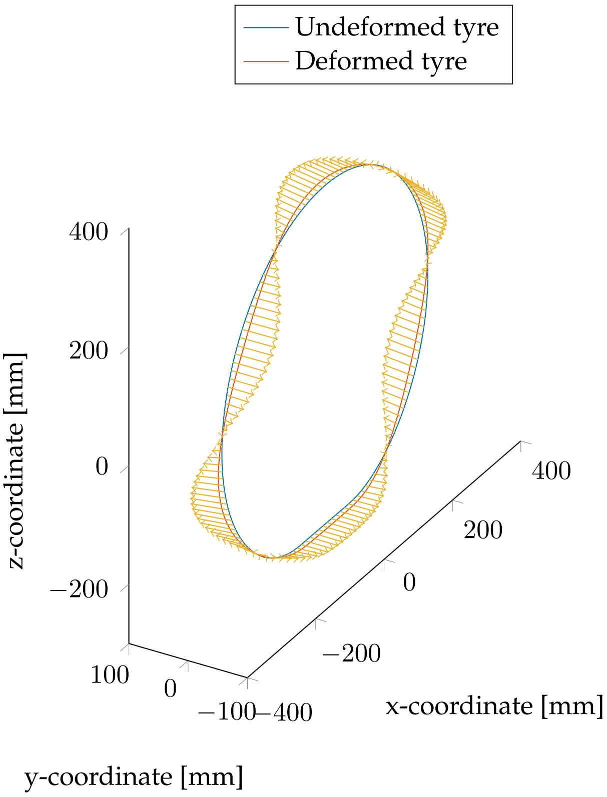


Figure 5.3: Visual representation of the coordinate and displacement data of a lateral (3,1) mode shape, which is in this case mode number 14 of a generic tyre. The yellow arrow indicate the direction of the displacements.

figures a distinct harmonic pattern can be recognized, as of course is to be expected due to the nature of eigenmodes. By determining what the direction is of these harmonic waves and what their specific wave number is the mode shapes should be recognizable. To do this automatically within the algorithm a fast Fourier transform (FFT) will be used on the displacement data. By performing a FFT the power spectrum density data should show peaks at the most dominant wavenumber of the displacement directions. This peak data can then be used by the algorithm to recognize all of the mode shapes. Beside this processed displacement data also the following additional information is generated using the following processing steps within the algorithm:

- The resultant radial displacement is calculated by using the displacements in x and z-direction. This is done to easily determine radial mode shapes as these show dominant displacements in the radial direction.
- The angles per node between the displacement direction and radial axis that goes from the centre of the tyre to the node are calculated. Then from these angles the mean angle value $\phi_{mode,mean}$ is calculated from all the absolute values of these angles. This mean value is used to determine whether or not a mode shape is a torsional mode shape.

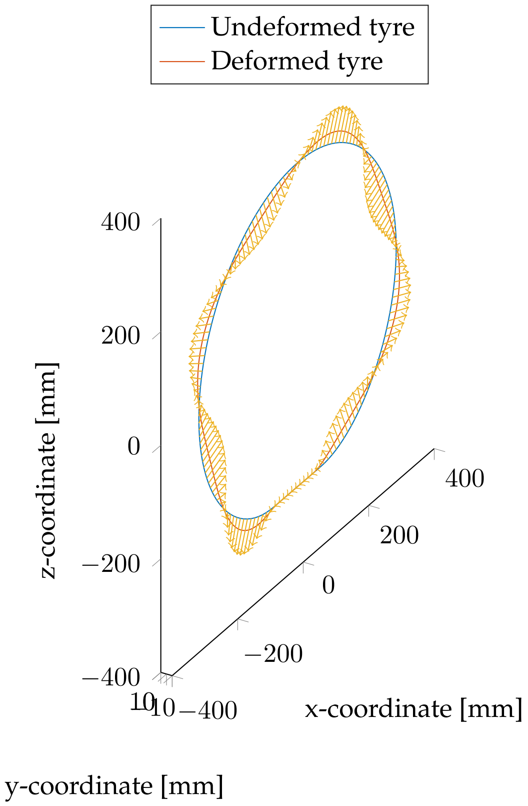


Figure 5.4: Visual representation of the coordinate and displacement data of radial (4,0) mode shape, which is in this case mode number 15 of a generic tyre. The yellow arrow indicate the direction of the displacements.

- To see which is the most dominant displacement the root mean square values of the displacements in all three directions and of the resultant radial displacement are calculated. For example, the root mean square of the displacement u_x is calculated with

$$u_{x,rms} = \sqrt{\frac{1}{n}(u_x, 1^2 + u_x, 2^2 + ... + u_x, n^2)} \tag{5.1}$$

All this processed displacement data is then used by the developed algorithm to recognize all the mode shapes.

Fast Fourier transform & Peak data

The FFT is performed using the build-in MATLAB `fft`-function on the displacement data as shown in figures 5.3 and 5.4. To visualize this displacement data better the upper graphs of figures 5.5 and 5.6 can be observed. These graphs show the displacement data of modes 14 and 15 in all three direction and the resultant radial displacement u_{xy} or u_{13}, plotted against the normalized position on the tyre's circumference. By actually performing the FFT on these displacement data, the lower graphs of figures 5.5 and 5.6 are obtained. From this FFT data

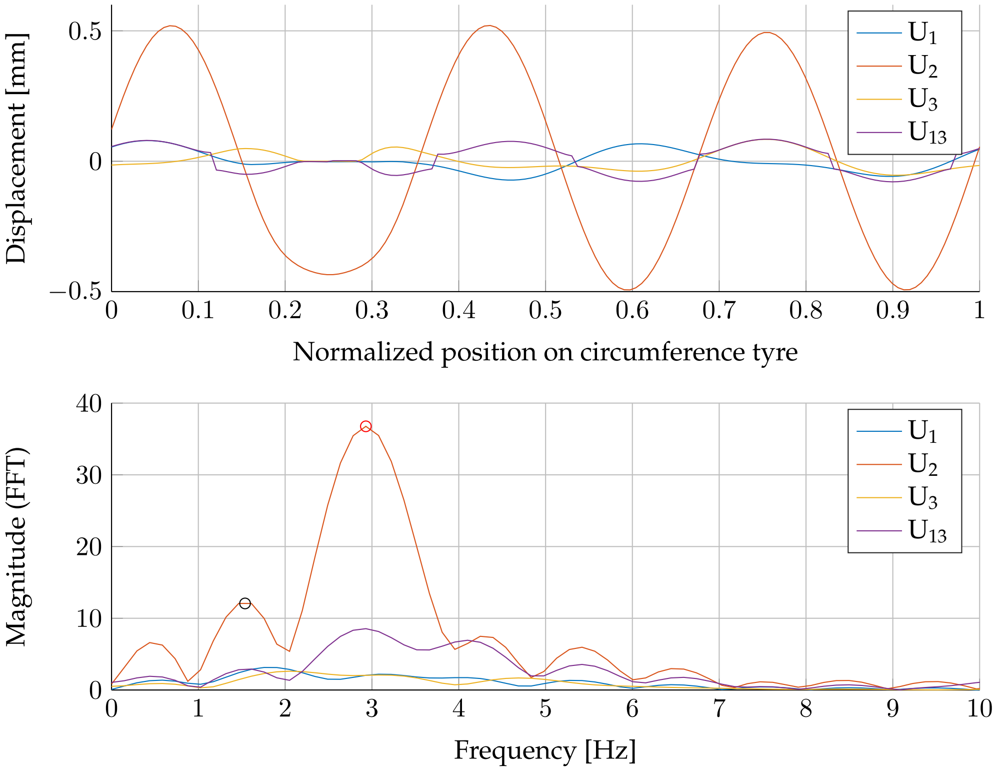


Figure 5.5: Raw and processed data from mode number 14. Upper graph displays all the displacements plotted against the normalized position on the tyre's circumference. The lower graph shows the results from the FFT of every displacement data set, where the peak marked with the red circle is the highest apparent peak (P_1) and the black circle the second highest peak (P_2).

the highest and second highest peaks are localized, which are in turn used by the algorithm to determine which type of mode shape it is and with what wave number. In the lower graphs of figures 5.5 and 5.6 the two peaks are found using a build-in MATLAB `findpeaks`-function. In the following sections the highest peak of the mode in question will be referred to as P_1 and the second highest as P_2.

5.4.1 Algorithm

Using all the processed displacement data the algorithm recognizes with a series of conditional steps what type of eigenmode it is and with what wavenumber. To recognize the mode shapes it is assumed that wavenumbers of both the radial and lateral mode types are always equal or higher than the previous mode. The first conditional step of the algorithm sequence is to determine whether it is a radial or lateral mode, which is

IF $u_{xz,rms}/u_{y,rms} > 4$, then it is a *radial* or *torsional* mode, go to ①.

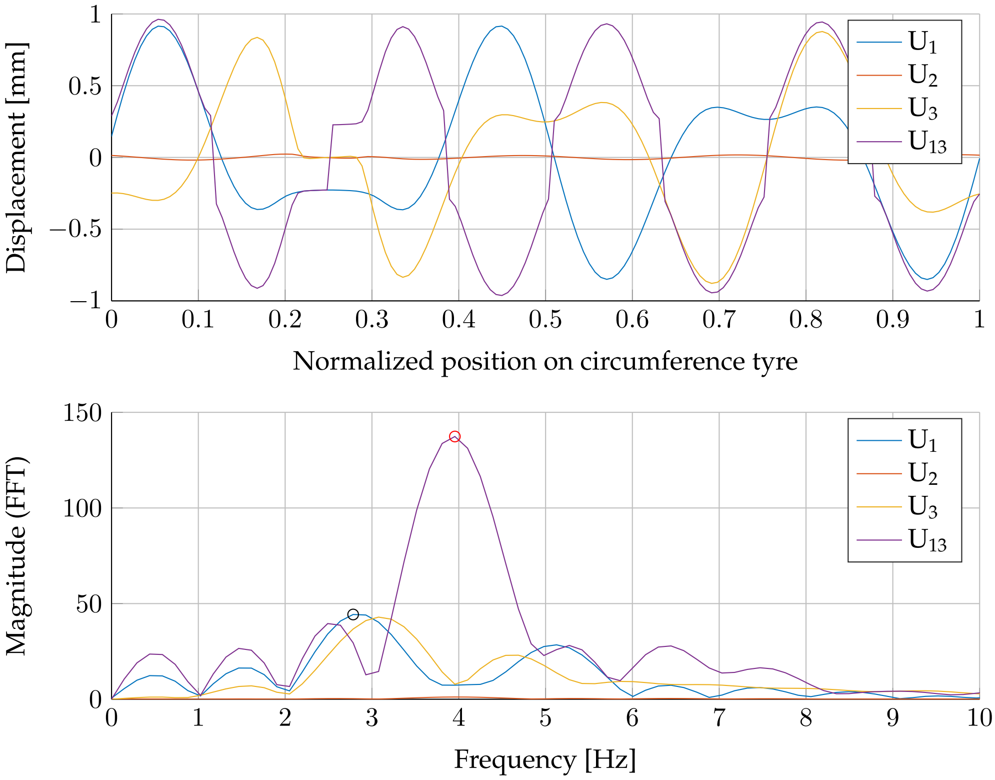


Figure 5.6: Raw and processed data from mode number 15. Upper graph displays all the displacements plotted against the normalized position on the tyre's circumference. The lower graph shows the results from the FFT of every displacement data set, where the peak marked with the red circle is the highest apparent peak (P_1) and the black circle the second highest peak (P_2).

ELSE it is a *lateral* or *axial* mode, then go to ②.

The factor 4 at the if-statement is fitted so that there is a significant enough difference between the two displacement RMS values from u_{xz} and u_y. The following steps depend on the what the aforementioned conditional statement found, either option ① or ②, this will be sub-divided in the two following paragraphs.

① Radial or torsional

Firstly it is checked if it is a torsional mode, this is done by using the mean angle value of the displacement direction.

IF $80 < \phi_{mode,mean} < 90$, then it is considered a *torsional* mode. N.B. the angle range between 80 and 90 degrees is found empirically.

ELSE it is a *radial* mode.

So if the considered mode number is not a torsional mode then it either is a radial mode or an unkown/junk mode. These are the following conditional statements that are performed:

> **IF** the highest peak P_1 of the resultant displacement u_{xz} is equal or larger than the previous mode wavenumber, then the mode is ($P_{1,r}$,0), where $P_{1,r}$ is the rounded value of P_1.
>
> **ELSEIF** the second highest peak P_2 of the resultant displacement u_{xz} is equal or larger than the previous mode wavenumber, then the mode is ($P_{2,r}$,0).
>
> **ELSE** the algorithm is unable to recognize the mode at hand, and the mode is labelled as *unknown*.

② Lateral or axial

Firstly the algorithm always checks whether or not it is an axial mode shape. Since the axial mode shape only shows a significant displacement in the y-direction, which is most dominantly non-harmonic. Therefore the highest peak P_1 will be apparent close to zero, hence the following conditional statement is checked:

> **IF** $P_1 < 0.5$ (or its rounded value is equal to zero), then the mode is considered an axial mode shape.
>
> **ELSE** the mode is either a lateral mode, or is unrecognizable.

So if the considered mode number is not an axial mode then it either is a lateral mode or an unkown/junk mode. Analogues to the radial mode part, the same conditional steps are performed for the lateral mode shapes.

> **IF** the highest peak P_1 of the displacement u_y is equal or larger than the previous mode wavenumber, then the mode is ($P_{1,r}$,0), where $P_{1,r}$ is the rounded value of P_1.
>
> **ELSEIF** the second highest peak P_2 of the displacement u_y is equal or larger than the previous mode wavenumber, then the mode is ($P_{2,r}$,0).
>
> **ELSE** the algorithm is unable to recognize the mode at hand, and the mode is labelled as *unknown*.

5.4.2 Limitations

As mentioned in the introduction of this section, the mode shape recognition script was primarily developed to test its feasibility and usability, hence making this algorithm as is primarily a proof of concept. The actual algorithm is developed in a trail-and-error kind of way, mostly using empirical found data for the conditional statements. Due to the lack of time, and also data in lateral direction of the tyre, the current state of the algorithm is very primitive, with known limitations. The first and most important limitation is the inability to distinguish any lateral mode shape with more than one bending wavelengths in the lateral direction of the tyre. The main reason why the current algorithm is incapable of recognizing these is the lack of nodal displacement data in the lateral direction. Consequently this algorithm can be come inaccurate at the transition point where these second order lateral bending waves start to become apparent. Secondly the algorithm is also not able to distinguish between the different double mode shapes with the same wavelength numbers, as described in section 4.4.3.

Of course there are a lot more shortcomings of this algorithm, but of course this is only natural due to the fact that it is a proof of concept. Much more elaborate work can be done to improve this idea for a mode shape recognition algorithm, especially when a more complete displacement data set is available.

Chapter 6

Results & validation

To get familiar with Abaqus and the post-processing that is involved, several simulations have performed. Modal analysis simulations are the Abaqus simulations that have been primarily run. Additionally, to also gain insight in transient analyses several of these type of simulations have been conducted as well. All the simulations have been run using three different tyre models, being a generic practice tyre model, the existing Apollo Alnac 4G tyre and the still being developed Apollo Alnac 4Gs tyre. Early on during the Abaqus learning phase of my assignment an issue was recognized with the material models, this will be discussed in the first section of this chapter.

Some of the results of the aforementioned tyre models and analysis types will be presented in this chapter to give the reader an idea of what simulations have been run, and what their results look like. The other additional results is added in appendix A, where mostly the graphs of the results have been added. The simulation results data is also required to validate the functionality of the automation tooling and its final results.

6.1 Material models limitations

During the early stages of the simulations an issue was recognized with the actual apparent damping in the results. The force frequency response function of the steady-state dynamic analysis showed abnormal high forces at the eigenfrequencies. This was accountable to the fact that the material data that is used for the simulations was incomplete, as it did not have the proper damping incorporated. Since this data would not be available in the coming months, a substitute proportional damping has been added to the material models, based on other material data that is similar to the material at hand. Proportional damping, or Rayleigh damping, is discussed in section 4.2.2.

Additionally to incorporate more damping *viscoelastic* material models are used instead of hyperelastic models as it also has the supplementary viscoelastic damping. Therefore, since the damping is not based on the actual materials that are used in the simulated tyres the results obtained from the simulation are not likely to match the tyre's real behaviour, but more-or-less resemble the behaviour. Therefore for basic functionality testing the data is usable enough, allowing me to analyse and compare the data with the made automation tooling.

Since the actual simulations are not the focus of my assignment this chapter will only cover a small selection of the run modal analysis simulations. The transient analysis results that are shown in this chapter are not run by myself and do not represent any specific purpose other than to test the functionality of the *Tyre Vibration Toolbox*.

6.2 Modal analysis results

Since the main focus of this assignment are the modal analysis simulations, these type of simulations have been run predominantly. Firstly for practice purposes, simulations were run on a generic tyre model, which also was used to investigate several different simulation settings and post-processing methods. Secondly, to test the created automation tooling multiple simulations have been run on variations of the Apollo Alnac 4G and Alnac 4Gs tyres have been run.

6.2.1 Generic tyre

To start a generic tyre model has been used to perform several simulations to practice with Abaqus and to investigate the following simulation settings such as the eigensolvers and the required amount of segments to use for 3D-revolve. The results from these simulations will not presented in this report as these are already covered in chapter 4. To test the post-processing

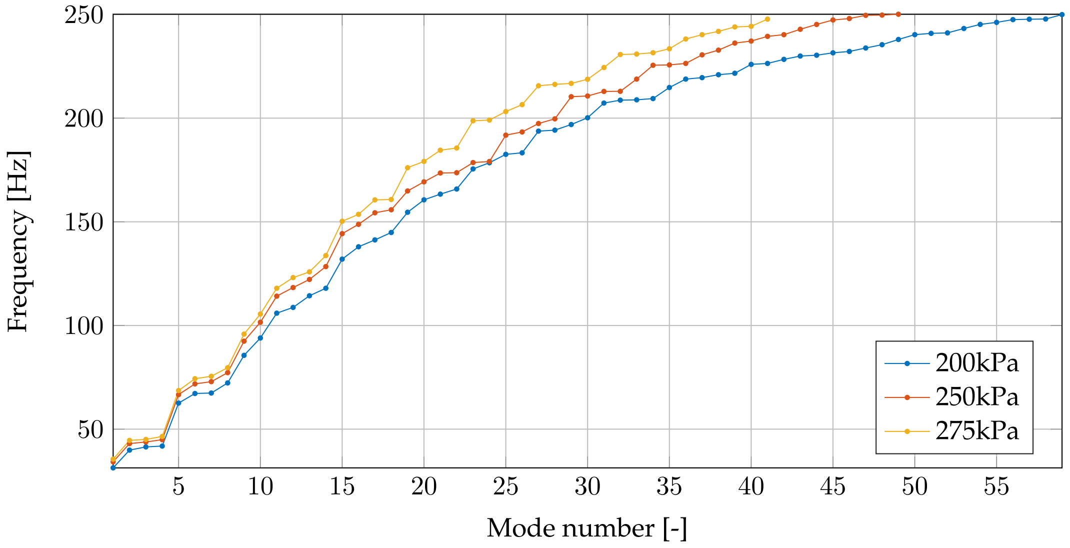


Figure 6.1: Mode numbers vs eigenfrequencies of the generic tyre inflation pressure variants.

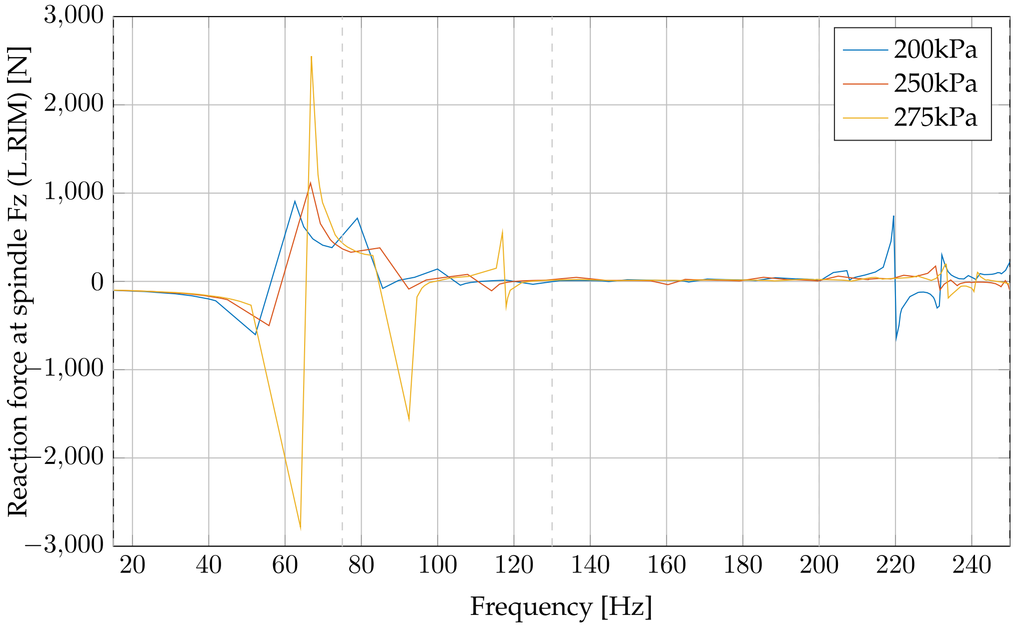


Figure 6.2: Force frequency response function plot of the generic tyre inflation pressure variants.

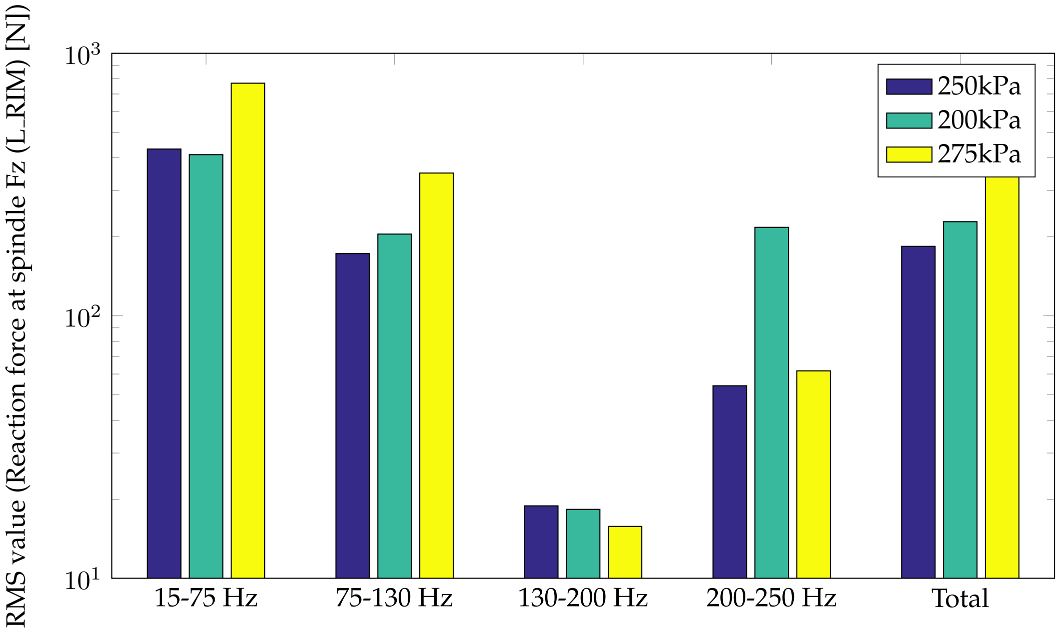


Figure 6.3: Root mean square values of the generic tyre inflation pressure variants.

with the root mean square values (RMS) the generic tyre model has also been used to perform simulations with different inflation pressures. The expectation was that increasing the inflation pressure also increases the overall magnitude of the force response, hence leading to less comfortable driving experience. Looking at the figures 6.1, 6.2 and 6.3 the following is observed:

- A higher inflation pressure means a stiffer tyre, the mode number vs eigenfrequency show the expected trend which is a steeper curve for the higher inflation pressure. The same is apparent at the force frequency response function since the peaks of the higher inflation pressure shift to the left.
- Below 200 Hz in the force frequency response function graph in figure 6.2 shows as expected higher response with higher inflation pressure. Above 200 Hz however this is not the case, since the lowest inflation pressure of 200 kPa overall shows higher peaks. Of course this is also reflected when looking at the RMS bar chart in figure 6.3, where the 200 kPa inflation pressure simulation has the highest RMS value at the 200 – 250 Hz frequency range. The minor discrepancy in the 130 – 200 Hz range is neglectable since the difference is exaggerated by the log scale of the Y-axis.

6.2.2 Apollo Alnac 4G tyre (195/60R15)

Following the generic tyre, simulations of the Apollo Alnac 4G tyre (195/60R15) have been performed with two construction variations:

- 1-Up construction: This tyre construction has only one carcass layer.
- 2-Up construction: This tyre uses two carcass layers for its construction.

Multiple simulations have been run of both the tyre construction variations, where in each simulation one of the base component materials has been changed to an alternative one. Therefore the results, as can be seen in appendix A.1, do show very slight differences between one another. Since the differences are so little no conclusion can be drawn as this is also not the focus of this assignment.

6.2.3 Apollo Alnac 4Gs tyre (215/60R16)

Three undisclosed tyre variants of the Apollo Alnac 4Gs 215/60R16 tyre have been analysed using the modal tyre analysis procedure. Since no specifics of the tyre variants are know the results are analysed as is, and the differences between these three tyre variants are not discussed in much detail. The used mesh and material data are encrypted files, hence the specifics of these files are unknown.

Table 6.1 gives an overview of the defining settings that have been used for the modal analysis of the C3, F2 and F5 construction variation of the Alnac 4Gs tyre. Initially the simulations were run using a 15 – 300 Hz frequency range, however since some of simulations had to be run due

Setting	Quantity or option
Material model	Viscoelastic with proportional damping added
Inflation pressure	230 kPa
Segments for 3D-model revolve	200
Footprint load	4316.4 N
Frequency range	15 – 200 Hz
Eigensolver	Lanczos
Steady-state dynamic	Eigenfrequency interval with 3 evaluation points

Table 6.1: Used Abaqus simulation settings for the Apollo Alnac 4Gs tyre simulations.

to a slight alteration the frequency range was also reduced to 15 – 200 Hz. The following three images have been generated using the application:

- The mode number vs eigenfrequencies can be seen in figure 6.4, where the F2 simulation is still from the simulation with the larger frequency range. As can be seen in this figure the differences are very minor.
- The force **f**requency **r**esponse **f**unction (FRF) data is depicted in figure 6.5. Here the F2 data above 200 Hz is trimmed away in the toolbox. Here the differences between the three different tyre constructions are more apparent.
- Figure 6.6 shows the root mean square (RMS) values of the FRF data. The entire frequency range is split into 4 frequency ranges, of which the first is a bit larger than the rest of the frequency ranges. The overall RMS value shows that the C3 variant is likely the most uncomfortable, especially due to its behaviour at 80 – 120 Hz as corresponds with its peak at 90 Hz in figure 6.5.

Based on this data it could be concluded that either the F2 or F5 are likely the best choices when solely looking at the comfort level at the 80 – 120 Hz frequency range.

6.3 Transient analysis results

To develop and test the transient analysis module of the toolbox simulations were run of all three Apollo Alnac 4Gs tyre variants. These transient analysis simulations are performed by letting the Alnac 4Gs tyre variants roll on a drum with a cleat as described in section 4.1.2. The time-domain, transformed frequency-domain and RMS values plots can be seen in appendix A.2.1, figures A.7, A.8 and A.9 respectively.

In hindsight the transient results are not comparable to those of the modal analysis as described in section 6.2.3. On these results alone no real conclusions can be drawn, however the following things can be observed:

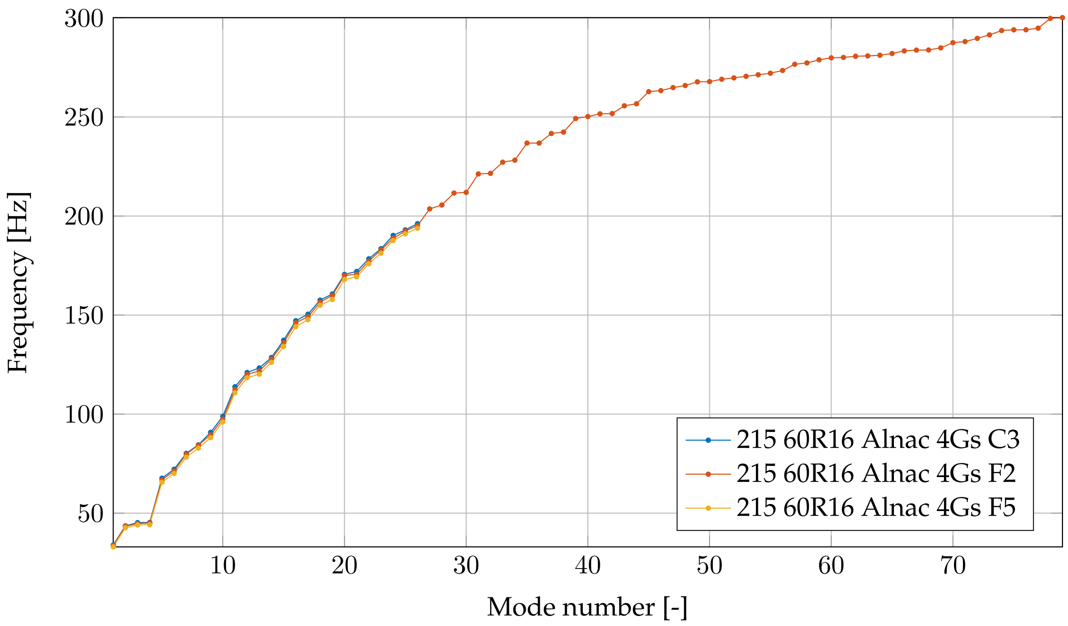


Figure 6.4: Mode numbers vs eigenfrequencies of the three Apollo Alnac 4Gs tyre variants.

- The first part of the force impulse response function (IRF) time-domain data in A.7 clearly shows rolling of the tyre before impact with the cleat until roughly $4 \cdot 10^{-2}$ seconds. In this first part minor oscillations can be observed around the equilibrium of approximately $-4.2 \cdot 10^{3}$ N which is the static tyre loading force. After the cleat impact certain natural frequencies are excited resulting in a large oscillation with an amplitude of around $4.5 \cdot 10^{3}$ N.

- The previously discussed oscillations are reflected in the transformed frequency data as can be seen in figure A.8. The most dominant oscillation appears to lay around 70 Hz, which corresponds with the large oscillation seen in the time-domain data. The other peaks are less recognizable in the time-domain data. This force frequency response data is also translated to root mean square values as can be seen in figure A.9.

Due to time constraints no further meaningful transient analysis simulations have been performed.

6.4 Testing & validation

During the development stage of the data extraction script and toolbox has been tested continuously by myself to ensure that the features work accordingly and the tooling is robust enough. At the end of the development stage of the tooling also has been tested by my supervisor to see if all of the tooling worked without errors and to see if it functionality met his expectations. Since the testing part is really hard to document properly it will not be done in this report. All

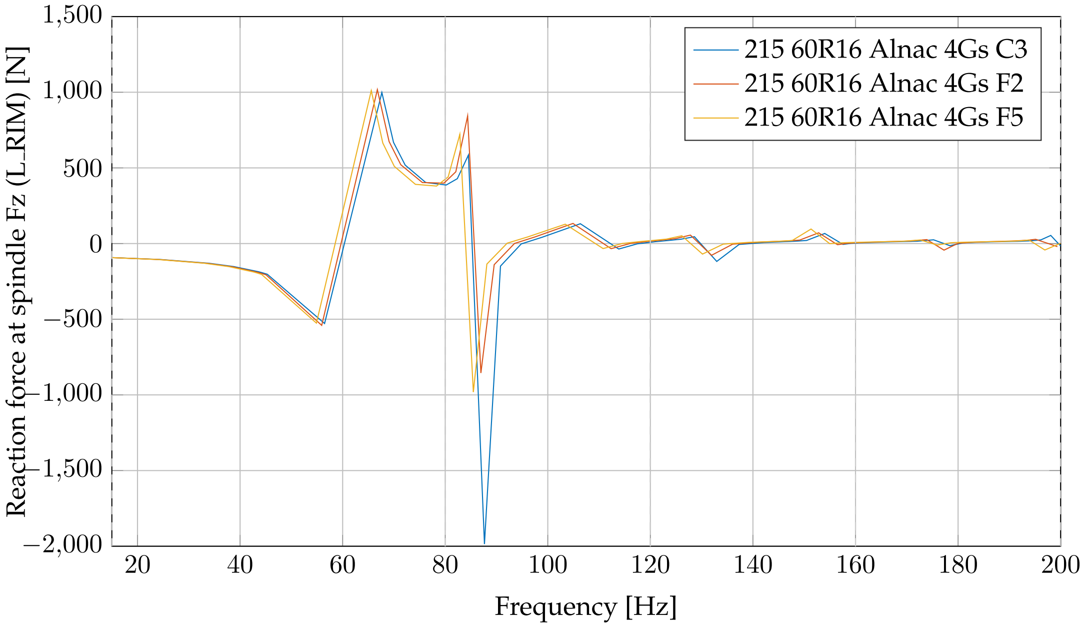


Figure 6.5: Force frequency response function plot of the three Apollo Alnac 4Gs tyre variants.

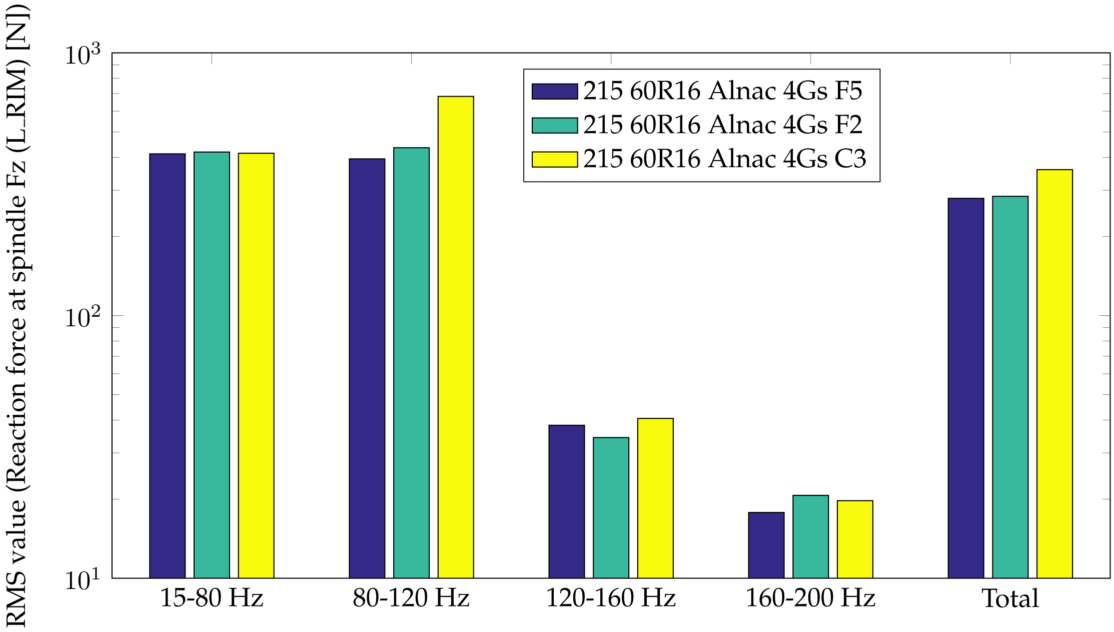


Figure 6.6: Root mean square values of the three Apollo Alnac 4Gs tyre variants.

the features as described in this report and the user manual (appendix B) have all been tested and should working accordingly.

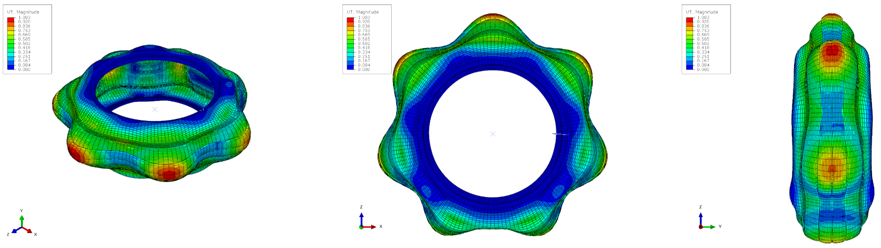
One thing that can be done though is the validation of the mode shape recognition script. As this recognition algorithm is more-or-less a proof of concept the script will be validated using one simulation. Of course during the development of the script other simulation data has been tested as well.

6.4.1 Mode shape recognition

To validate the mode shape recognition script a separate modal analysis tyre simulation has been performed, which had the code name PB12_046A. To validate the mode shapes, beside the mode shape recognition script, a visual mode shape cognition has been performed by myself by analysing the visual images of the mode shapes. The overview of the results can be seen in table 6.2, where the results from an unloaded and loaded tyre can be seen. In this table the red highlighted cells indicate a discrepancy between the mode shape as recognized by the algorithm and visually. The following things can be observed:

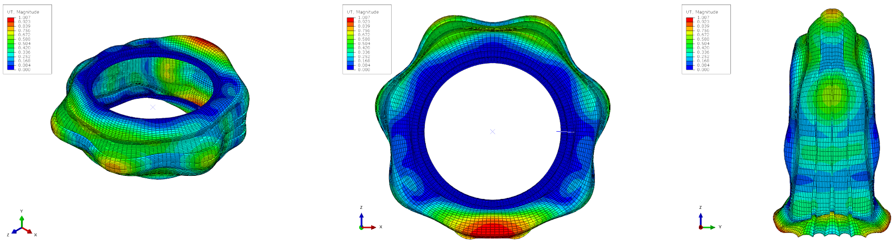
- With the unloaded tyre the torsional mode is not recognized properly. Likely this is caused because the recognition script is developed based on loaded tyre instead of unloaded tyres, hence this wrong recognition. In a possible future version this can be fixed easily by fine-tuning the torsional recognition part of the algorithm accordingly.
- For the unloaded tyres modes 24 and 27 are recognized as (7,0) modes, while these are in reality junk modes. The reason why junk modes can not always be properly recognized is that only the displacement data of the nodes on the symmetry line of the tyre are at hand, and the lack of displacement data in the lateral direction of the tyre. Therefore not enough information is available to distinguish between the proper mode shapes and the junk ones. Figures 6.7 and 6.8 show the actual displacement images of mode shape 24 and 27 respectively. These figures confirm that these mode shapes do show abnormal mode shape behaviour.

The reason why the PB12_046A tyre has been used for this validation step is because there was also experimental data available from this tyre. After consideration it was concluded that this experimental data was not really useful for the validation of the mode shape recognition script. Therefore it was left out of this report as it does not fit within the scope of this assignment.



(a) Isometric view **(b)** Side view **(c)** Front view

Figure 6.7: Mode shape 24 of the loaded PB12_046A tyre.



(a) Isometric view **(b)** Side view **(c)** Front view

Figure 6.8: Mode shape 27 of the loaded PB12_046A tyre.

	Unloaded tyre			Loaded tyre (670 kg)		
#	Frequency [Hz]	Algorithm	Visual	Frequency [Hz]	Algorithm	Visual
1	59,0	**axial**		55.41	**axial**	
2	77,8	(1,1)		73.58	**tors**	
3	77,8	(1,1)		74.13	(1,1)	
4	77,9	(0,0)	tors	78.55	(1,1)	
5	89,8	(1,0)		90.11	(1,0)	
6	89,8	(1,0)		93.23	(1,0)	
7	108,9	(2,0)		110.06	(2,0)	
8	108,9	(2,0)		118.06	(2,0)	
9	128,2	(2,1)		123.65	(2,1)	
10	128,2	(2,1)		130.79	(2,1)	
11	129,3	(3,0)		132.26	(3,0)	
12	129,3	(3,0)		141.70	(3,0)	
13	150,5	(4,0)		155.54	(4,0)	
14	150,5	(4,0)		166.55	(4,0)	
15	173,7	(3,1)		170.2	(3,1)	
16	173,7	(3,1)		180.7	(3,1)	
17	175,7	(5,0)		182.7	(5,0)	
18	175,7	(5,0)		195.1	(5,0)	
19	204,7	(4,1)		203.7	(4,1)	
20	204,7	(4,1)		212.6	(4,1)	
21	205,3	(6,0)		214.1	(6,0)	
22	205,3	(6,0)		223.4	(6,0)	
23	227,2	**junk**		229.9	(5,1)	
24	229,5	(5,1)		236.8	(7,0)	junk
25	229,5	(5,1)		237.1	(5,1)	
26	237,7	(7,0)		247.6	(7,0)	
27	237,7	(7,0)		249.4	(7,0)	junk

Table 6.2: Mode shape recognition results of a unloaded and loaded PB12_046A tyre. When a mode shape as recognized by the algorithm does not correspond with the visually recognized mode shape the table cell is marked red, and the correct mode shape is described in the next column.

Chapter 7

Conclusions & recommendations

As is stated in the introduction the main goal of this internship assignment was to automate and partly standardize the modal analysis procedure of a tyre. With the creation of the *data extraction* script and the *Tyre Vibration Toolbox* MATLAB application, it can be said that the automation part has been successfully realized. Users are now able to efficiently post-process data from either modal or transient tyre analysis with the created tooling, without any extensive manual tasks. Additions such as the mode shape recognition, export capabilities and other miscellaneous features are the first steps towards a more extensive application. The mode shape recognition algorithm is primarily meant as a proof of concept, meaning that the current state of this algorithm is very primitive. To improve this algorithm further, firstly mode nodal displacement is required in the lateral direction of the tyre. Using this additional lateral data the algorithm should be improved further, potentially leading to better recognition capabilities of the algorithm, including the lateral mode shape with wave numbers higher than one and distinguish between mode shapes using Kindt's additional mode shape identifier labels.

For future use within Apollo, both the automation script and application can be used as pilot applications for the *platform* project. If required both the script and application can be extended or altered further to accommodate for the additional required functionality. This should be possible since the within Apollo the Python and MATLAB scripting languages are known and already used in other tooling.

Due to the issues with the material models and the limited available time, the actual numerical modal analyses have not been very extensively and successfully performed. Even though the results have been successfully processed in the Tyre Vibration Toolbox no real useful conclusions have been drawn from these results. Therefore more work and simulations are required with the proper material models to be able to draw useful conclusions, and to really showcase the capabilities of the toolbox.

Appendix A

Additional results

The following sections will show the results of additionally run modal and transient analysis simulations of several different tyres and/or configurations thereof.

A.1 Modal analysis results

A.1.1 195/60R15 Alnac 4G variants 1 Up variant

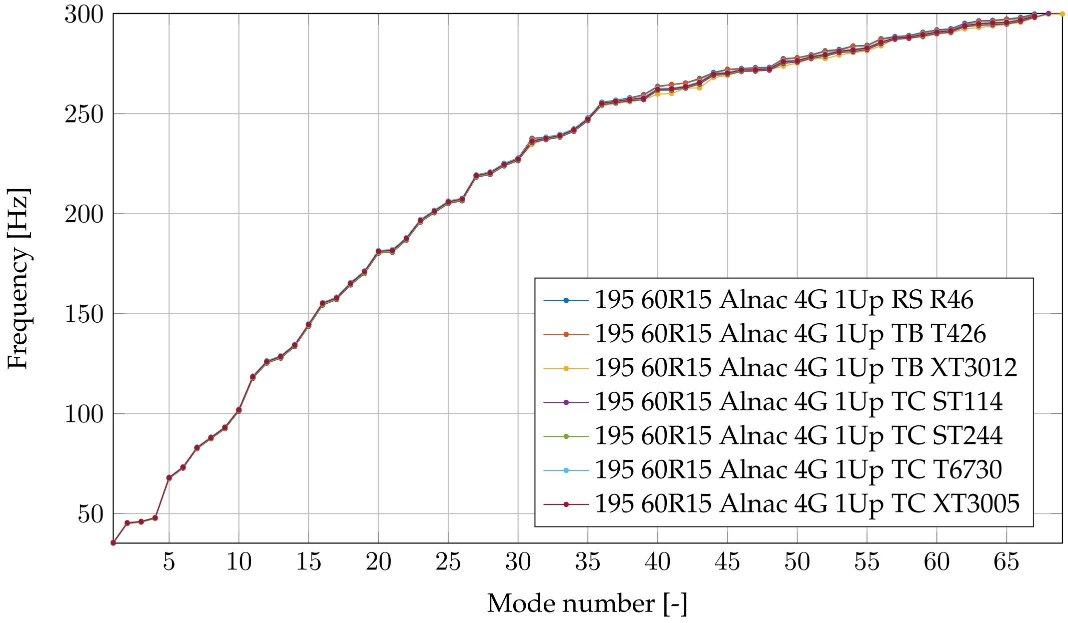


Figure A.1: Mode numbers vs eigenfrequencies of the Apollo Alnac 4G 1 Up tyre variants.

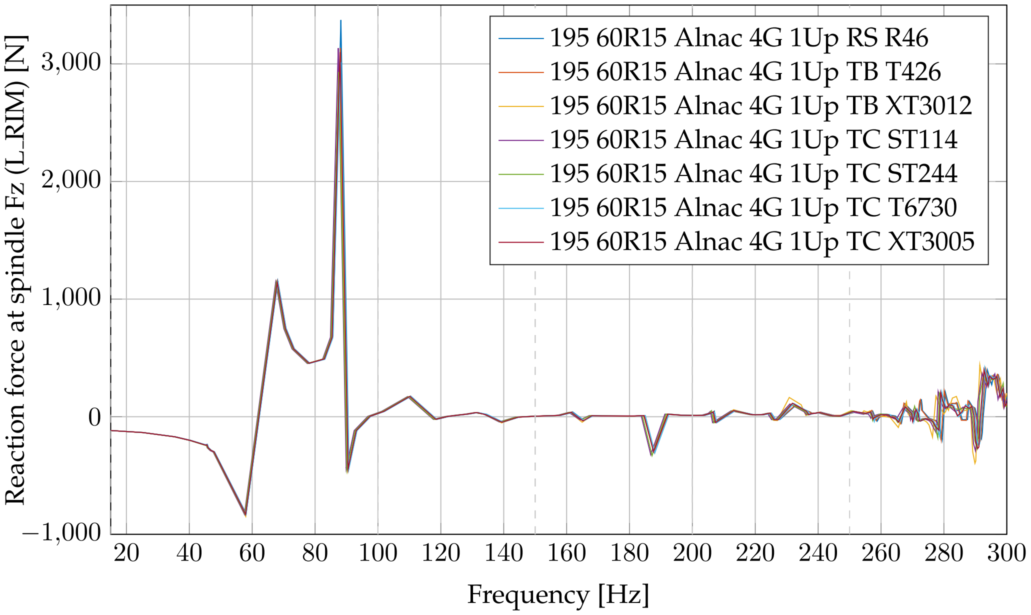


Figure A.2: Force frequency response function plot of the Apollo Alnac 4G 1 Up tyre variants.

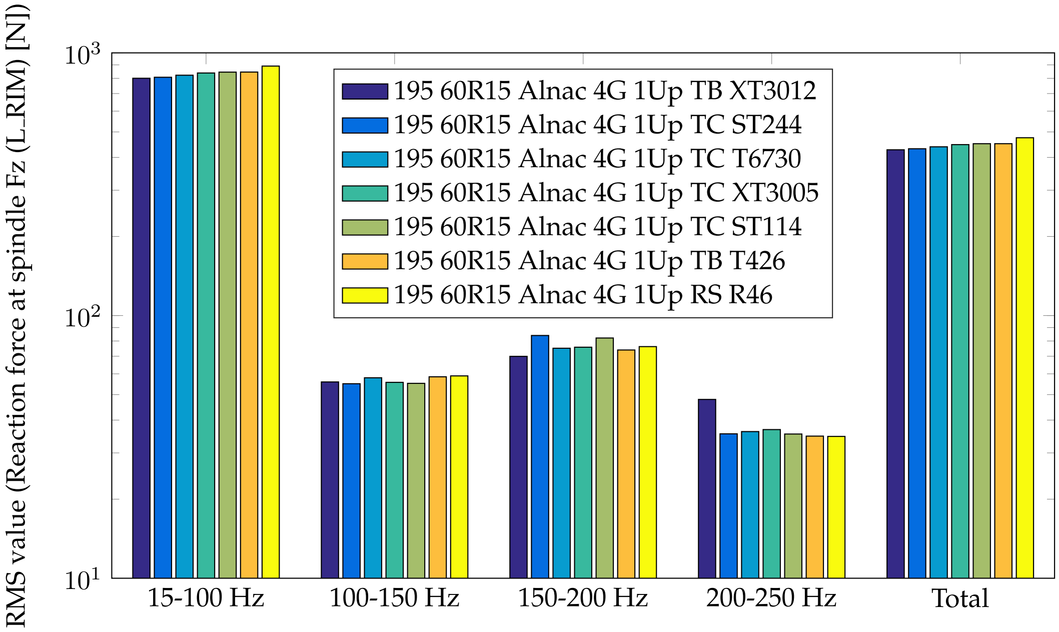


Figure A.3: Root mean square values of the Apollo Alnac 4G 1 Up tyre variants.

A.1.2 2 Up variant

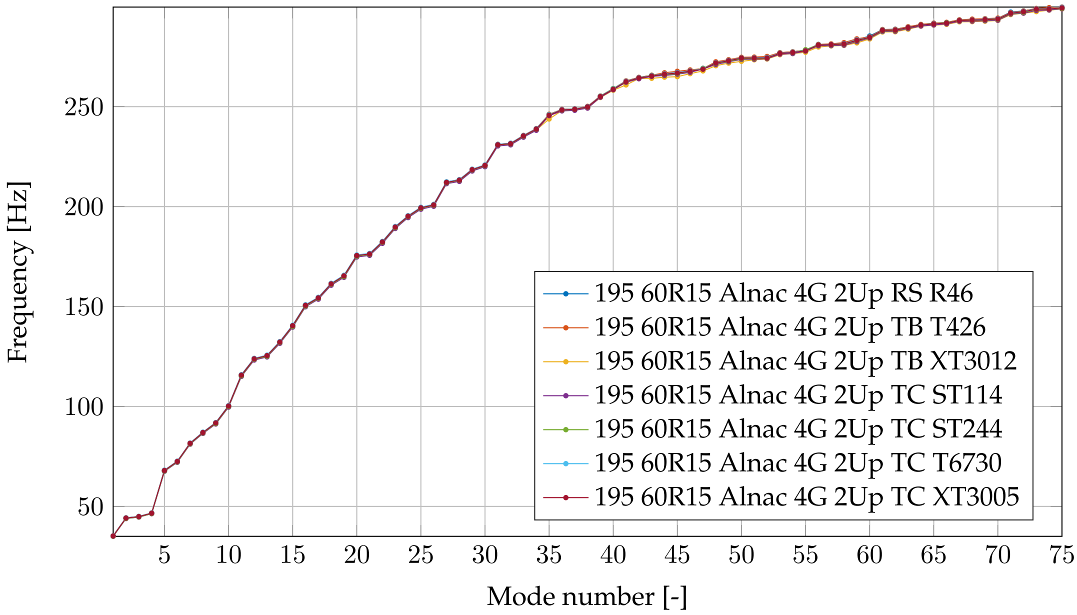


Figure A.4: Mode numbers vs eigenfrequencies of the Apollo Alnac 4G 2 Up tyre variants.

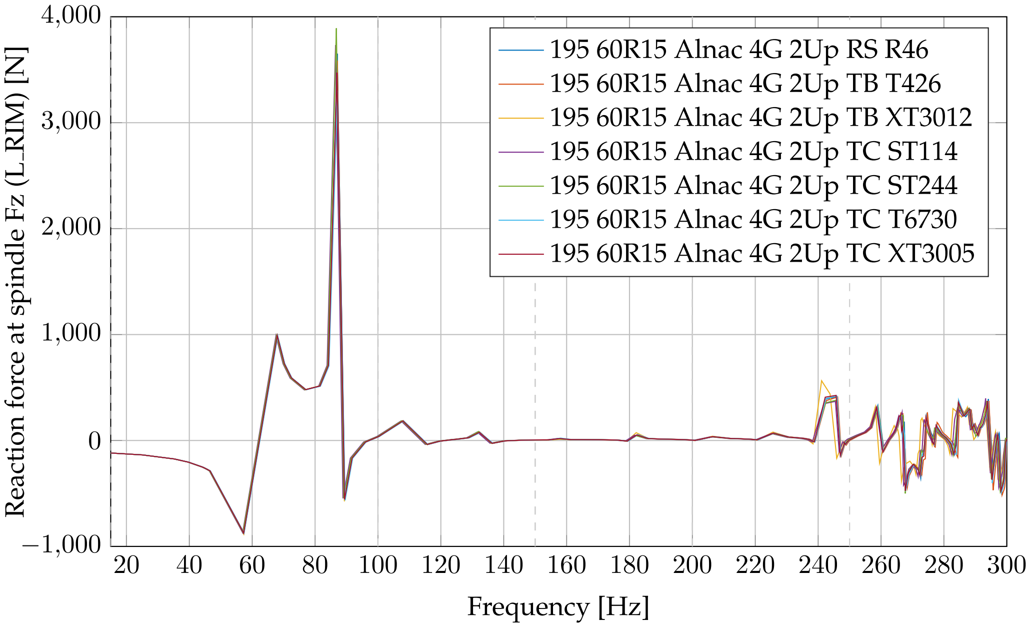


Figure A.5: Force frequency response function plot of the Apollo Alnac 4G 2 Up tyre variants.

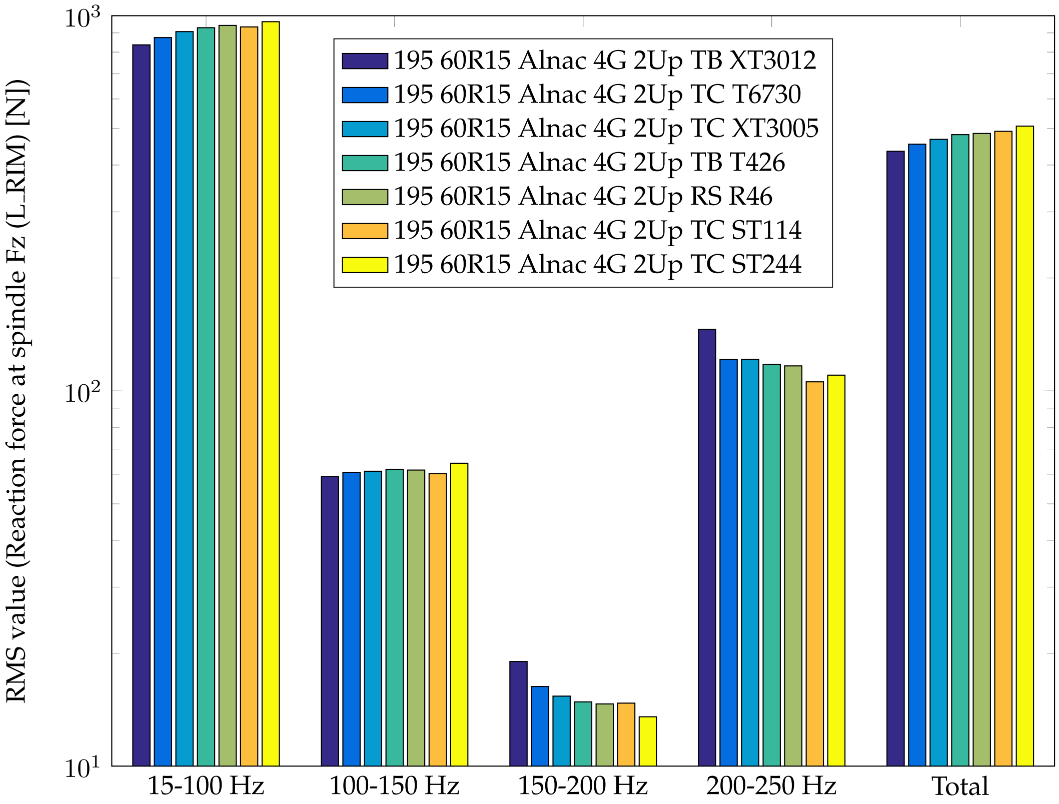


Figure A.6: Root mean square values of the Apollo Alnac 4G 2 Up tyre variants.

A.2 Transient analysis results

A.2.1 Apollo Alnac 4Gs tyre (215/60R16)

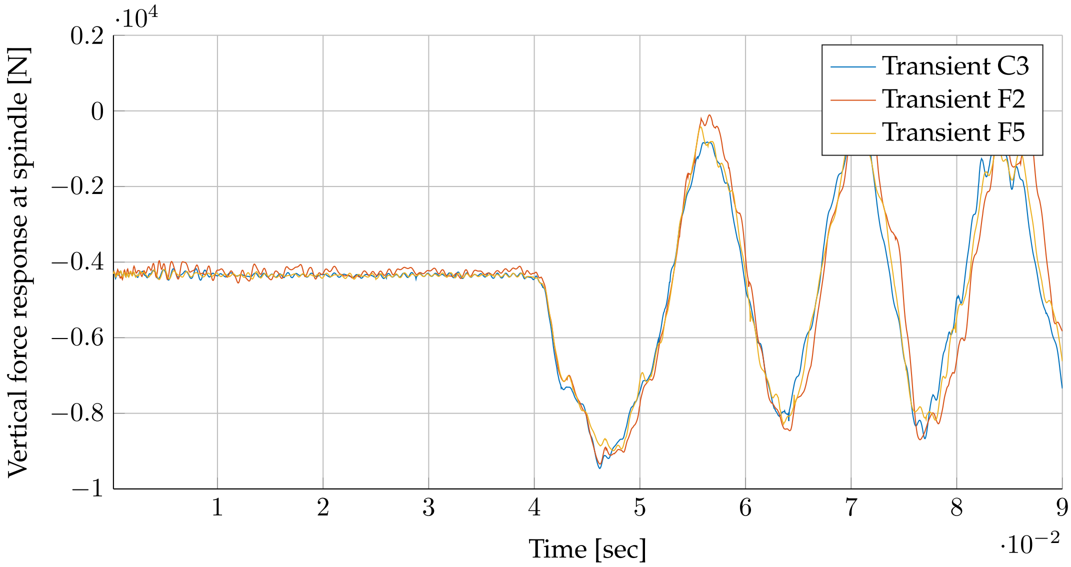


Figure A.7: Force impulse response function (IRF) in the time-domain of the three Apollo Alnac 4Gs tyre varients (F2, F5 and C3).

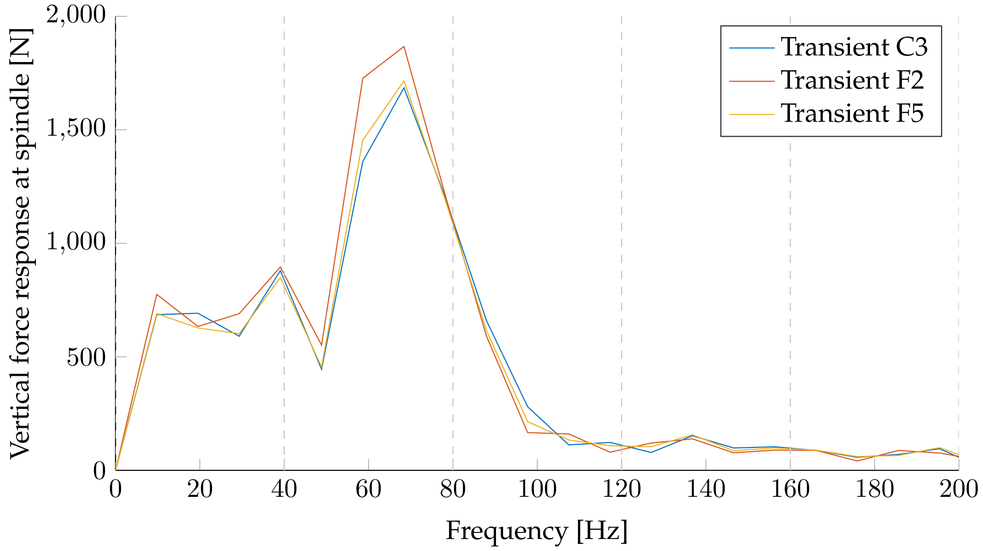


Figure A.8: The force frequency response (FRF) data of the three Apollo Alnac 4Gs tyre varients, obtained by transforming the IRF-data using a fast Fourier transform.

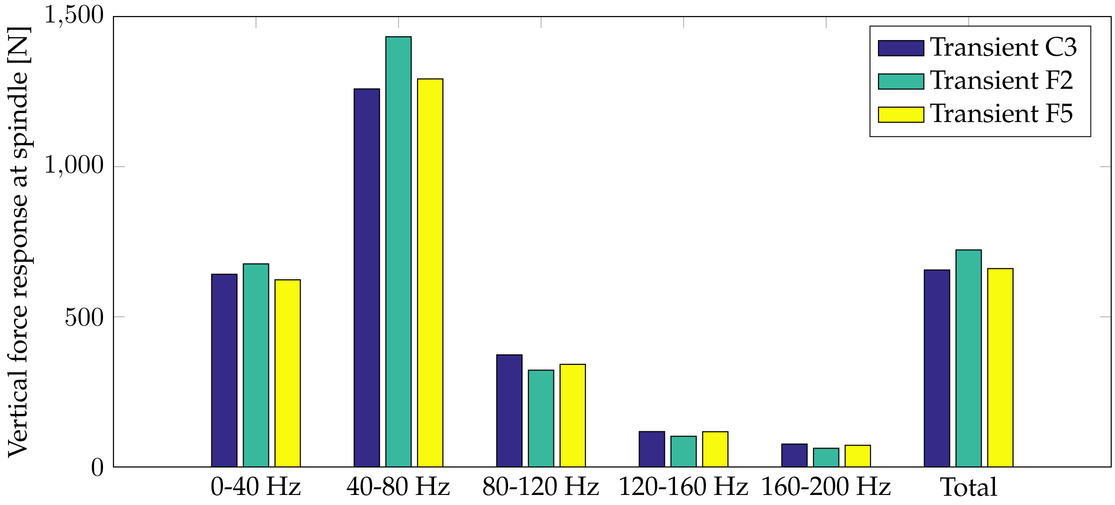


Figure A.9: Root mean square values of the FRF-data from figure A.8 of the three Apollo Alnac 4Gs tyre varients (F2, F5 and C3).

Appendix B

Toolbox manual

This user manual is written to guide the user through the semi-automated procedure to assess the vibrational behaviour of a tyre. The guide will cover two tyre vibration analyses to capture both the low and high frequency response spectra of the tyre, primarily acquired through finite element analysis (FEA) software. One of main purposes of this manual is to be able to quickly and easily compare tyre designs variants in the early design-stages of a tyre. This ultimately allows the user to make educated design decisions based on acquired insight.

B.1 Introduction

Nowadays more and more tyre vibration analyses are performed to be able to improve the a tyre's vibration and noise characteristics. Principally tyre vibration can be classified in two major categories: the *lower* frequency behaviour and the *higher* frequency behaviour. When doing analyses with a finite element method (FEM) package such as Abaqus, the former is generally done with the means of a so-called steady-state dynamic (SSD) analysis, solved in an implicit manner. Whereas the latter is done using a transient analysis, which is solved using an explicit solver. To get a better understanding of both types of solvers, and their respective purpose, both will be briefly explained in the next two paragraphs.

A **low frequency modal analysis** is done to get insight into the vibrational *structural* dynamic behaviour of a tyre, generally up until a frequency of around 300 to 500 Hz. The differentiating aspects of this specific type of simulation is the SSD step, as defined by *modal analysis*; being the study of dynamic properties under vibrational excitation. This step determines the force response of a structure by harmonically exciting it at a specified force amplitude. To make the SSD analysis more efficient it is pre-empted with an eigenfrequency extraction procedure. By using the extracted eigenfrequencies as input, the SSD steps focuses around these frequencies. Since these frequencies represent structurally critical points, within the proximity of these points, the frequency force reponse will show peaks there, hence the focus around these points.

A **high frequency analysis** is performed to get an idea on how tyres respond in the in the higher frequency vibrational spectrum, defined as 500 Hz and up. This dynamic behaviour is assessed using the aforementioned transient analyses. Fundamentally, a transient analysis is done using an impulse, exciting a tyre for a short time, i.e. transiently, and capturing its dynamic time-response. For tyres this is mostly done using a steady-state rolling analysis, followed by transient dynamic analysis, which normally has an irregularity[1,2] in road surface or drum where the tyre will roll on. The other way of performing a transient analysis is by performing a transient impact with for example a hammer-like device or with a pendulum, this however will not be covered. As outcome, the transient analysis data is a time-force response, representing the vertical force acting on the axle. To be able to properly examine and compare the transient analysis results, the time-data has to be transformed to the frequency-domain. This makes the data much more insightful, as it will show peaks at the problematic frequencies. The specifics of transforming time-data to the frequency domain will be elaborated in a later stage of this user manual.

B.1.1 Outline

This documentation will focus on how to obtain and analyse data from Abaqus-simulations using custom built automated procedures. Overall the process is two-part, namely a data extraction and a data analysis part. The former is an Abaqus Python-script executed through command-line interface (CLI), and the latter is a MATLAB stand-alone graphical user interface (GUI) application.

This guide will consist of the following sections:

- Background information of all types of Abaqus simulation procedures that are performed to asses the lower and higher frequency behaviour of tyres.
- Overview of what the data extraction script is able to do and a guide on how to use it, finally illustrated with several examples.
- Step-by-step tutorial on how to use the GUI application to analyse data and compare different tyre variants with each other. The application's capabilities will be demonstrated during the course of this tutorial.

B.2 Background information

To prevent the simulation procedures from becoming a black box procedure - especially for people unfamiliar with Abaqus - this section will briefly cover the different simulation steps involved. The following two paragraphs will elaborate on these simulation steps for both the

[1] This irregularity is often a cleat, being a rectangular part that sticks out of the road or drum surface.

[2] In more advanced simulations, an actual road with irregularities can be modelled and be part of the excitation.

low & high frequency analyses. Several available options per simulation will also be elaborated here.

B.2.1 Low frequency modal analysis

As mentioned earlier, the low frequency analysis is mainly distinguished by its SSD analysis step. Prior to the SSD analysis, several prerequisite steps must be done, an overview of these simulations steps can be seen in figure B.1. This diagram shows that the Abaqus simulation for the low frequency behaviour of a tyre consist of two seperate parts, a 2D and 3D simulation part. To properly explain the difference between these two simulation parts, firstly one must understand the rotational symmetry of a tyre and how this is used.

Rotational symmetry

Not regarding a complex 3D-tread, a tyre is rotationally symmetric around its center axis. Hence by creating a 2D cross-section of the tyre containing all its internal structural components and revolving it around its centre axis generates a 3D-model of an entire tyre. In short, rotational symmetry is often referred to as *axisymmetric*, hence the name **2D-axisymmetric**. The model generated by revolving the results from the 2D-axisymmetric simulation around its center axis is referred to as the **3D-revolved** (or 3D-footprint). In a later stage of this user manual the following abbreviations will be used:

> Simulation abbreviations
>
> Within the automation procedures 2D-axisymmetric and 3D-revolved simulation parts are commonly abbreviated with `AXI` and `REV` respectively. Simulation-files of the specific type are always suffixed with these two abbreviations.

To get a better understanding of the simulation procedure both simulation parts will be briefly discussed in the next two paragraphs.

2D-axisymmetric steps

The 2D-axisymmetric part of the simulation, as can be seen in figure B.1, mainly consists of two important steps:

Rim mounting step: This step mounts the tyre in the rim by bringing the two rim halfs together; figure B.2a and B.2b show an example of this.

Inflation step: After the tyre is correctly mounted, a distributed pressure is applied on the inner surface, simulating the inflation of a tyre. By doing this the tyre gets displaced outward fully fixed within the rim, as can be seen in figure B.2c.

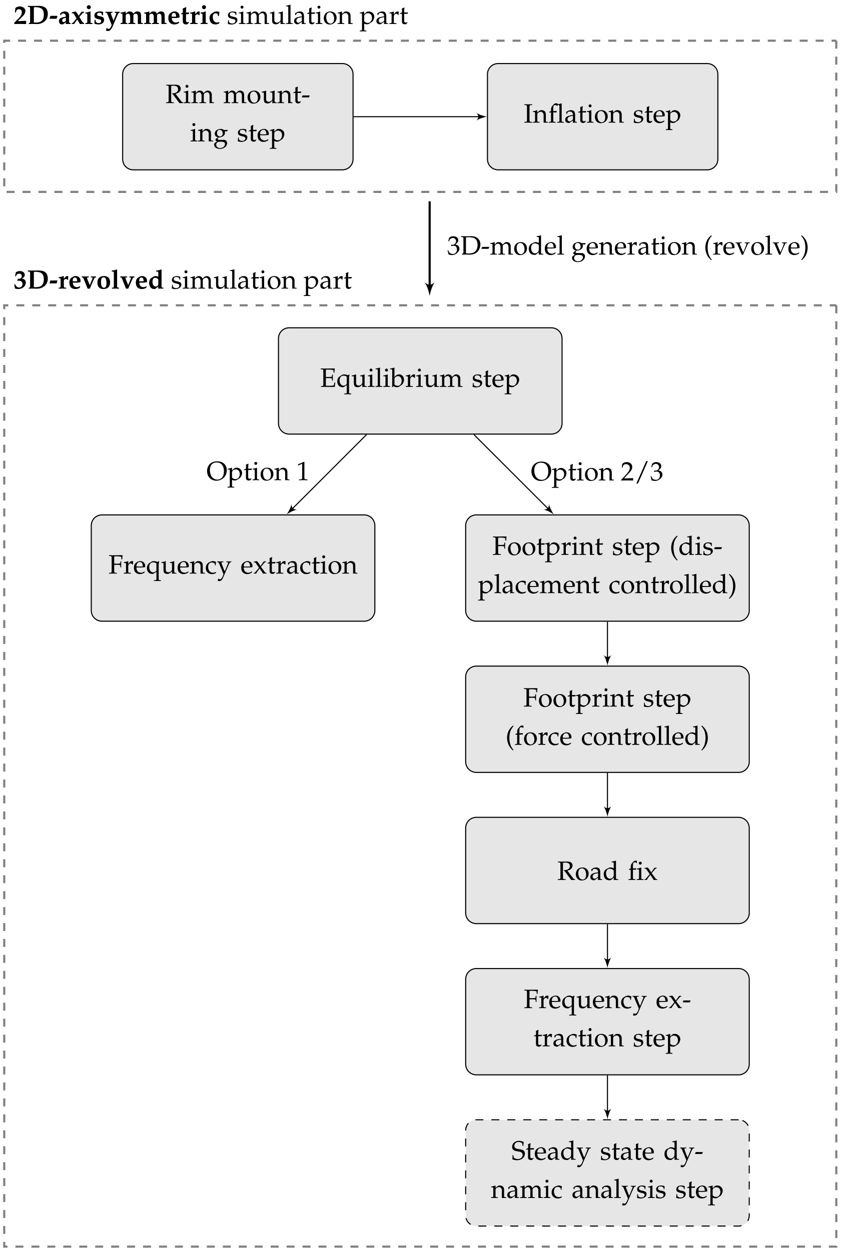
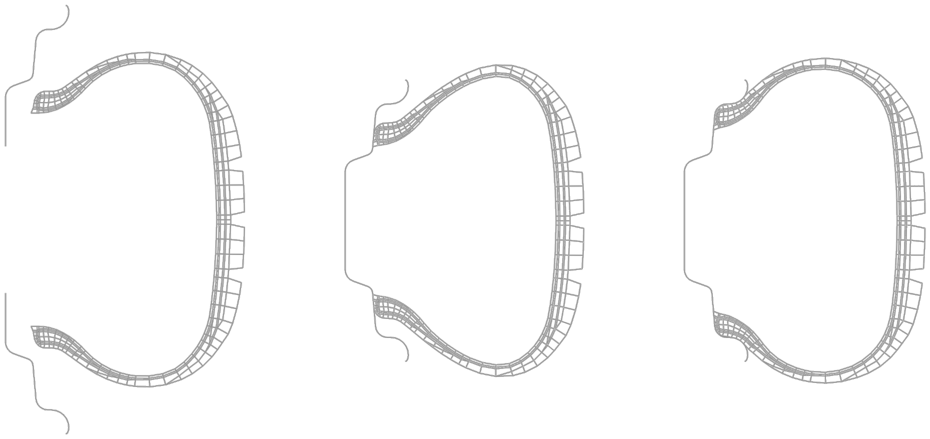

2D-axisymmetric simulation part
Rim mounting step
Inflation step
3D-model generation (revolve)
3D-revolved simulation part
Equilibrium step
Option 1
Option 2/3
Frequency extraction
Footprint step (displacement controlled)
Footprint step (force controlled)
Road fix
Frequency extraction step
Steady state dynamic analysis step

Figure B.1: Overview of low frequency analysis procedure of a tyre with the different options visualized.

After the simulation has run succesfully, the solution of a fully mounted and inflated tyre is ready to be used in the 3D-revolved simulation part.



(a) Undeformed, rim still split **(b)** Tyre mounted on rim **(c)** Tyre fully inflated

Figure B.2: The three 2D simulation steps, with the rim cross-sections on the left.

3D-revolved steps

After the inflated tyre solution has been revolved around its centre axis, the 3D-revolved model is ready for the next steps:

Equilibrium step: Prior to any other step, the equilibrium of the generated 3D-model is ensured by performing this step.

Footprint (displacement controlled) step: Before applying the actual load to the tyre, the tyre is put into contact with the road using a predefined displacement. The reason why this step is performed is because it is known by practice that the solution-procedure has a better convergence rate than it would have by directly loading it with a force.

Footprint (force controlled) step: This step loads the road surface with the specified loading force. Since the rim is fixed in space, the tyre deformation will change accordingly so that it gets to its static equilibrium.

Road fix: This step imposes boundary conditions that fixes the road surface in space. In theory this step could also be included into the frequency extraction step. Past experience however showed that the road surface does move slightly when combining both steps into one, hence the seperate *road fix* step.

Frequency extraction step: Now that the tyre is in its deformed state this step will use an eigenvalue algorithm to obtain all the eigenfrequencies and eigenvalues within the specified frequency range.

Steady state dynamic analysis step: This step harmonically excites the tyre at specific frequency and captures its force-response. These specific frequency points are commonly determined using the eigenfrequencies that have been found during the frequency extraction step.

Simulation options

For modal analysis, as said earlier, the simulation steps as depicted in figure B.1 are of use to properly analyse the tyre's low frequency behaviour. As can be seen in this figure, the user does not have to execute all the simulation steps, as long as is that the 2D-axisymmetric simulation is successfully perfomed with both the rim mounting and inflation steps. In the 3D-revolved part of the simulation, several steps can be applied or skipped leading to three different options:

Option 1: Free tyre, with the eigenfrequency extraction, but without road contact and the SSD analysis part.

Option 2: Loaded tyre, with the eigenfrequency extraction, but the SSD analysis part.

Option 3: Loaded tyre with all the simulation steps, thus including the SSD analysis part for the force response data.

How these options are set when running the data extraction script will be explained in section B.3.1.

Output of simulations

After the simulations have been run successfully two `.odb`-files will have been generated per tyre variant. These files are Abaqus database-files which contain all the requested data from the simulations and can be used to post-process the tyre simulation results. The data extraction of these `.odb`-files is elaborated upon in section B.3.

B.2.2 High frequency transient analysis

A transient analysis of a tyre with the rolling tyre method is done in the following steps:

1. 2D-axisymmetric step;
2. 3D-footprint step;
3. Steady-state rolling step;
4. Transient dynamic analysis step.

The first two steps are partly similar to the ones as explained in the low frequency modal anlysis, up until the *footprint force controlled step*. The latter two will be briefly covered in the following two paragraphs.

Steady-state rolling step

Prerequisite to the transient dynamic analysis step is the steady-state rolling analysis. The main purpose of this step is to find the *free-rolling solution* by matching the angular velocity of the tyre with the velocity of the road surface (translation) or of the drum (angular). If these velocities are correctly matched, zero torque will be apparent at the axle. The main difficulty lies within the fact that these velocities cannot be readily matched because the exact deformation of the tyre at the footprint is not known beforehand. Consequently leading to a slight difference in radii, leading to an unknown circumferential length. Therefore to successfully determine the free-roll velocity an indirect iterative procedure is used.

Transient dynamic rolling step

When the free-roll solution has been found the results can be imported into the Abaqus/Explicit solver, together with a road surface or a drum, including some sort of irregularities, e.g. a cleat. Running the explicit transient dynamic rolling simulation will yield the time-force response of the forces acting on the tyre.

Output of simulation

The extraction of the transient data that is needed for the post-processing in the GUI-application is not automated. A step-by-step procedure on how to do this manually is described in appendix B.B.

B.3 Data extraction script

To accommodate an easy, fast and consistent acquisition of the simulation output-data a data extraction script was made. This script is written with the Abaqus scripting capabilities with use of the Python scripting language. The script is executed through the CLI. Prior to running the script, the user must first

- check several input and output settings;
- and set specific run options to get the desired output.

To get a clear insight on what has to be done and what is possible, first the input/output options are discussed. Hereafter the actual running of the script is explained, and finally all the possible outputs are summarised.

B.3.1 Input/output options

There are two ways in which the user can input options; within the data-extraction script (in-file options) and during invocation of the script (invoke options), both of which are discussed in the following paragraphs.

In-file options

The user-modifiable options that are set internally within the `data_extraction`-script are located at the parts defined with the headers *Run options* and *Step and part names*. Here is an overview of these parameters:

- Run option (`runOption`: this is the option that specifies which simulation steps have been performed, as mentioned earlier. The user can chose between one of the three inputs, being the numeric value of either `1`, `2` or `3`.
- Mode shape numeric data extraction (`modeShapeNumeric`): The option allows the user to turn on or off the coordinate displacement data of the mode shapes. This information is needed for the mode shape recognition feature within the GUI application. This feature basically identifies every mode shape's shape, the exact workings of which will be explained in section B.4.5 and appendix B.A. To turn this extraction *on* or *off* fill in `True` or `False`, with the first letter capitalized
- A comprehensive list of all the step names, part names and import/export prefixes and suffixes. This allows great compatibility when using the data-extraction script. For example, all the names as defined in the simulation's input-files can be adjusted accordingly within this list to make the extraction procedure work. For more information please refer to the list itself within the data-extraction script, all of the list entries should be self-explanatory.

Please note that these options have to be set correctly in order the script to work, otherwise Abaqus will throw an error and the extraction process will be terminated prior to completion. If this is the case please read the error description carefully and try to trace the source of the issue.

Invoke options

When invoking the script some of the options are set by passing them as arguments while others purely lie within the choice of syntax. To show how these options are set, firstly the layout of the invoke-syntax is shown, whereafter each part is explained with more detail:

```
<abaqus version> viewer <gui>=<script> -- <folder> <modes>
```

where:

`<abaqus version>`: Specifies which version of Abaqus to use, in this case version 6.13-1, hence `abq6131`.

`<gui>`: Either `script` or `noGUI`, respectively running the script with and without the Abaqus/Viewer GUI active.

`<script name>`: The name and location of the data extraction script, most likely `data_extraction.py`.

`<folder>`: Input argument one, being the folder in which the `.odb`-files are located. Here, three different input options are available:

- A folder containing `.odb`-file pairs (AXI and REV files);
- A folder containing sub-folders in which one `.odb`-pair is located;
- The string `current`, this searches in the current folder for both of the situations mention above.

Please note that folder input arguments cannot contain any spaces within its name, .replace spaces with underscores (_).
`<modes>`: Input argument two, specifing the mode shape images have to be extracted, either the mode number until which images have to be extracted or the string `all`, meaning that all available mode shape images have to be acquired.

To get a proper understanding on how to run the data-extraction script, two examples will be shown in section B.3.1.

B.3.2 Data extraction example

After having run several simulations, the directory structure could look something like figure B.3. Lets assume for this example that all the simulation steps have been performed, i.e. a loaded tyre including the SSD step. Prior to the data extraction run the user has to check if the correct run options and names have been set within the data extraction script. After this is done the user can invoke the data extraction part.

```
..
├── Tyre_A
│   ├── Tyre A variant #1_AXI.odb
│   ├── Tyre A variant #1_REV.odb
│   ├── Tyre A variant #2_AXI.odb
│   └── Tyre A variant #2_REV.odb
├── Tyre_B
│   ├── Tyre B variant #1_AXI.odb
│   └── Tyre B variant #1_REV.odb
└── data_extraction.py
```

Figure B.3: Example directory structure.

Example 1
Data extraction of all the variants of Tyre A, with no Abaqus/viewer GUI active and requesting the first 10 mode shape images:

```
> abq6131 viewer noGUI=data_extraction.py -- Tyre_A 10
```

Example 2
Data extraction of variant #1 of Tyre B, with Abaqus/viewer active and requesting all the mode shapes available:

```
> abq6131 viewer script=data_extraction.py -- current all
```

As can be seen the `current` folder-option has been utilized in this case. This perfectly illustrates that when invoking this command, only the data of Tyre B is extracted, as the folder of Tyre A contains multiple variants and the current-option only extracts data from folders containing one `.odb`-pair. Of course, in this case it is also possible to simply use `Tyre_B` instead of `current` as folder-argument.

B.3.3 Data extraction output

All the output generated by the script are put into a folder with the name of the simulation with the suffix `_data`. Depending on the folder-input during the script invocation, this data-folder is either located inside the folder containing the `.odb`-files or outside. These are the two distinctions, with the folder-input being a folder containing

- one or more `.odb`-pairs: data-folder gets placed alongside the `.odb`-files;
- sub-folders containing one pair of `.odb`-files: data-folder gets placed outside the folder containing the `.odb`-files.

The output of the data-extraction script is dependant on what options have been set by the user.

Visual data

Several images are generated during the extraction process, all of these are saved with the `PNG`-image format. These are the images that are generated with the data extraction process:

- Mold vs inflated contour image;
- Mode shape images, in three views, being the isometric, side and front;
- Footprint contact pressure image.

Numeric data

All the numeric data is saved with the *Repost XY Data*-option within Abaqus/viewer to a `txt`-file.

- Contact area values;
- Eigenfrequency values;

- Frequency response force values of both rim sides in all three directions;
- Coordinate locations of a (un)loaded tyre's symmetry nodes;
- Displacements of the tyre's symmetry nodes of every mode shape.

B.4 Post processing: Tyre Vibration Toolbox

The post-processing GUI application has been created with MATLAB 2015a, and is compiled to a stand-alone and royalty free application using the MATLAB Compiler, which can be used without having a MATLAB license on your PC. This section will cover most of the capabilities the program has and where to pay extra attention.

Prior to the first use, the MATLAB run-time environment has to be installed, this is covered in the next paragraph. Following that the capabilities of the application are discussed in a tutorial-like way.

B.4.1 Getting started

MATLAB run-time installation

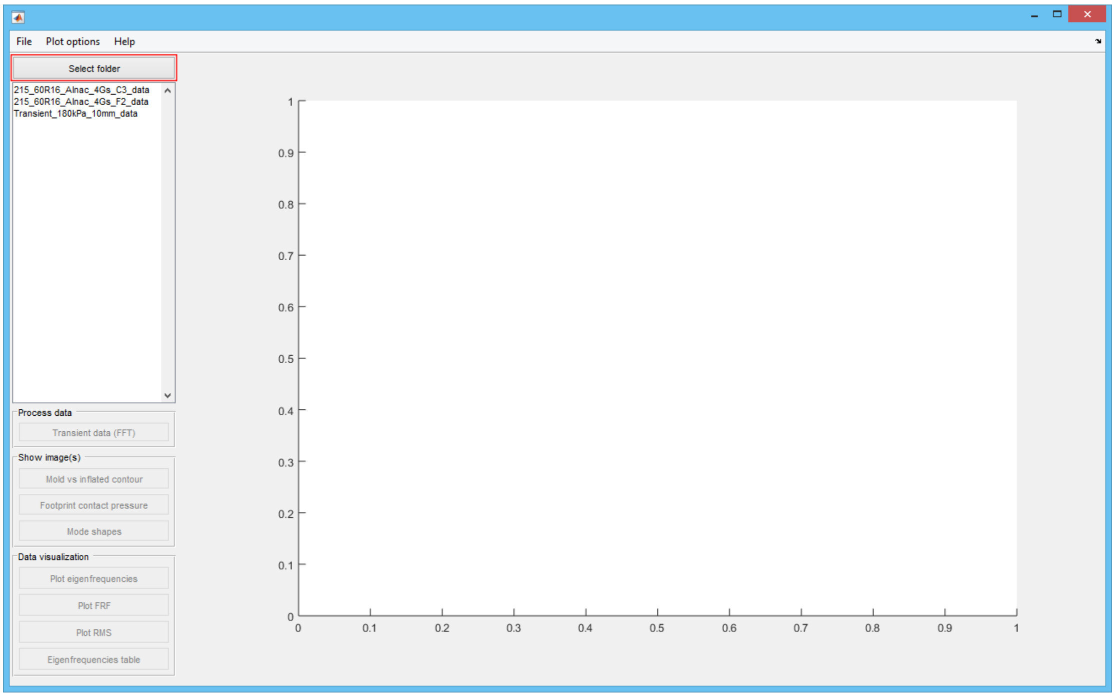
Before using the application the first time, the corresponding run-time installer, namely `MATLAB_2015a_runtime_inst` (located in the `\Installer\` folder), must be installed. N.B. even if you already have the MATLAB run-time environment installed on your PC from another application, it is advised to still run this installation to make sure the latest version is installed.

Starting the application

To start the GUI application the `TyreVibrationToolbox.exe` file must be executed (located in the `\Application\` folder). When starting the application it opens up at the main window, as shown in figure B.4. Most of the controls are located within the left panel, which become active when usable. The rest of the controls are located within the top menus and pop-up windows. These controls will be discussed throughout the following paragraphs.

B.4.2 Selecting folders

To analyse simulation data the user has to select one or multiple data-folders in the so-called listbox. This listbox contains all the `_data`-folders that are located in the current work folder, or more commonly known as the current working directory (CWD). To get started, first a folder has to be selected by pressing *Select folder* (indicated with the red box in the figure), or by going to the menu item *File > Open folder* (CTRL+O). This opens a select folder window where the CWD can be changed. This new CWD has to contain `_data`-folders, otherwise the program will give you an error message.



215_60R16_Alnac_4Gs_C3_data
215_60R16_Alnac_4Gs_F2_data
Transient_180kPa_10mm_data

Figure B.4: Starting screen of the Tyre Vibration Toolbox.

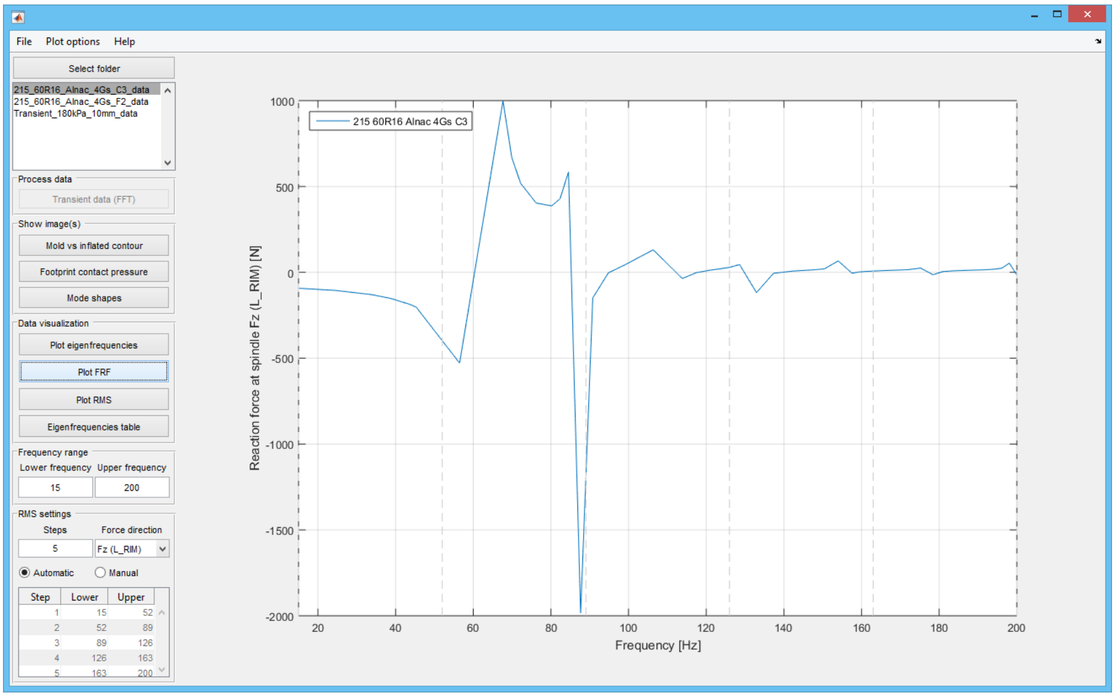
Once a CWD has been selected containing `_data`-folders, only these folders will show up in the listbox. All other files and folders not suffixed with `_data` are not shown. In this example, as can be seen in figure B.4, three data-folders are visible, namely two variants of the 215/60R16 Alnac 4Gs and one unkown tyre of which a transient analysis is done.

B.4.3 Selecting & plotting data

When selecting one of the Alnac 4Gs tyres, see figure B.5, certain buttons on the left panel become active, corresponding with the data available within the selected folder. When pressing *Plot FRF*, the frequency-force response function (FRF) becomes active on the right side of the application. Beside this, extra controls become visible on the left panel, namely the frequency range and the RMS settings, here a short explaination on RMS:

Root mean square values

To get a better insight on how tyres compare to each other based on the FRF-data from the SSD analysis, the **root mean square (RMS)** values of specific frequency ranges are calculated. RMS, also known as the quadratic mean, is a statistical measure defined as the square root of the mean of the squares of a sample [B.1].



215_60R16_Alnac_4Gs_C3_data
215_60R16_Alnac_4Gs_F2_data
Transient_180kPa_10mm_data

Frequency range

Lower frequency	Upper frequency
15	200

RMS settings

Steps	Force direction
5	Fz (L_RIM)

● Automatic ○ Manual

Step	Lower	Upper
1	15	52
2	52	89
3	89	126
4	126	163
5	163	200

Figure B.5: FRF Plot.

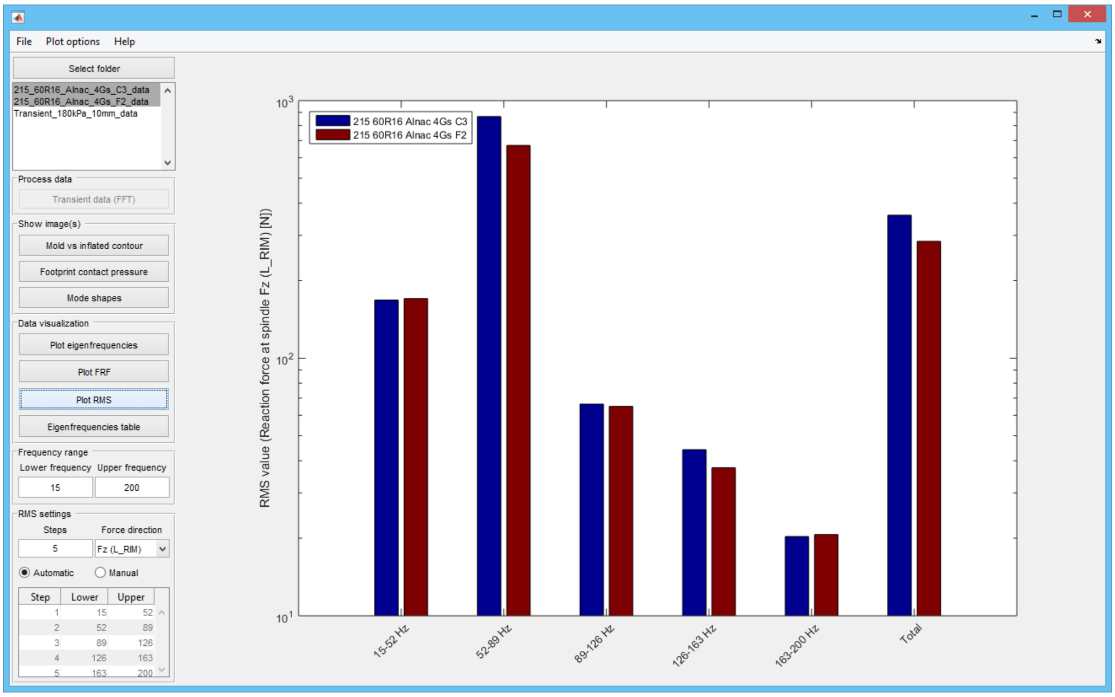
Next to the FRF-plot in figure B.5, dashed lines are also visible in the plot. These lines divide the entire frequency range in smaller ones, as defined by the options in the *RMS settings*-panel (lower left corner of the application). This is an overview of the options available:

- Automatic: this option automatically divides the entire set/available frequency range in N steps. The amount of steps to use can be set in the *Steps*-field.
- Manual: this option allows the user to manually set the frequency ranges within the table that becomes editable as soon as the manual option is activated.

Beside the settings that influence the frequency ranges there is also the option to change the force direction of the frequency-force response plot, this is done by selecting one of the other available forces in the *force direction* drop-down menu.

These RMS values can be plotted. To illustrate this, an additional data-folder is selected in the listbox. When pressing the *Plot RMS*-button a RMS bar chart is plotted, seen as the one in figure B.6. On the left panel nothing will change compared to the case where the FRF-plot was active. One of the features that has become available within the *Plot options*-menu list is the *Sort RMS values*, which - as the name implies - sorts the RMS values based on the total RMS-values.

In a similar way a mode number vs eigenfrequency plot can also be generated by pressing the *Show eigenfrequencies*-button.



215_60R16_Alnac_4Gs_C3_data
215_60R16_Alnac_4Gs_F2_data
Transient_180kPa_10mm_data

Frequency range

Lower frequency: 15
Upper frequency: 200

RMS settings

Steps: 5
Force direction: Fz (L_RIM)

● Automatic ○ Manual

Step	Lower	Upper
1	15	52
2	52	89
3	89	126
4	126	163
5	163	200

Figure B.6: RMS Plot.

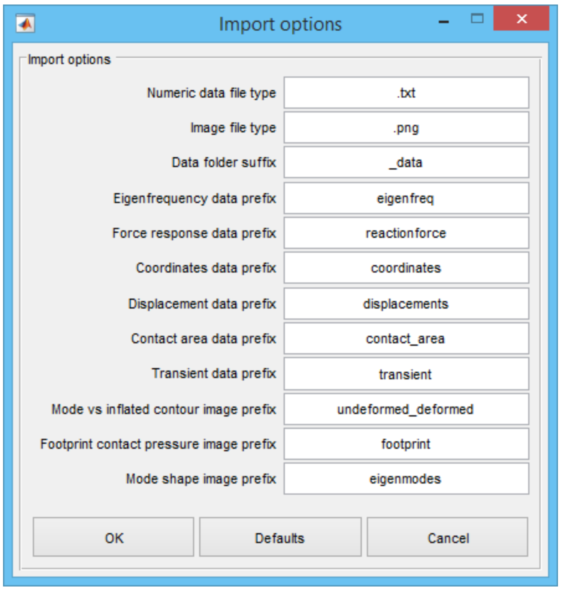
Plot options

The *Plot options*-menu item, as mentioned earlier for the sort functionality for the RMS-plot, also contains several other options that can be used for different plots. Here is a summary of the different options:

- *Y-Axis > Log scale*: Toggles the log-scale on the y-axis on or off.
- *Show legend*: Toggles the legend on or off.
- *Legend position*: A list of all the possible legend locations, with the one active marked as checked.

Data import options

To allow the user to have some flexibility on how data is imported into the program, import options have been added. The import options can be accessed through the file-menu: *File > Import options*, within this window the user can modify certain file types, suffixes and prefixes, see figure B.7. When pressing the button *Defaults*, the default import options are filled in again. To accept the applied changes simply press the *OK*-button.



Numeric data file type	.txt
Image file type	.png
Data folder suffix	_data
Eigenfrequency data prefix	eigenfreq
Force response data prefix	reactionforce
Coordinates data prefix	coordinates
Displacement data prefix	displacements
Contact area data prefix	contact_area
Transient data prefix	transient
Mode vs inflated contour image prefix	undeformed_deformed
Footprint contact pressure image prefix	footprint
Mode shape image prefix	eigenmodes

Figure B.7: Import options, containing all the file type and data/image prefixes and suffixes.

B.4.4 Export figures & data

Using the export capabilities of the application allows the user to save the current active plot as an image, or to export numeric data to an Excel-file. To do so, navigate to *File* > *Export*, this opens a window such as shown in figure B.8. These are the different export possibilities:

- Export current figure: Two file types can be chosen, namely the `PNG`-format and `PDF`-format.

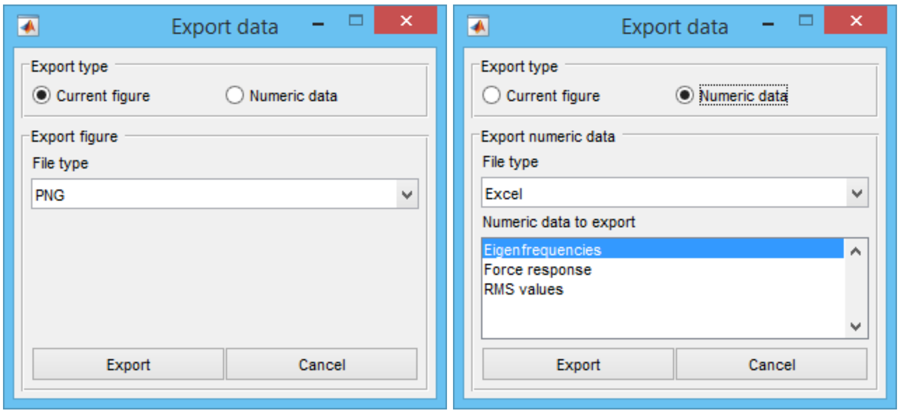
- Export numeric data: Only one file type is available, being an Excel Workbook in `.xlsx`-format. As can be seen in figure B.8b, the user has the ability to select the numeric data which has to be exported.

When pressing the *Export*-button a save file window will pop-up, in which the user can set the file-name and choose where the file has to be saved.

B.4.5 Eigenfrequencies table

To get an overview of all the eigenfrequencies press the *Eigenfrequencies table*-button. This opens a new window containing a table. Within this table all the eigenfrequencies with their corresponding mode number are listed. Figure B.9 shows an example of such a table.

One extra feature of the application is the **mode shape recognition**. Pressing the *Recognize mode shapes*-button in the table-window automatically identifies the mode shapes by running a recognition algorithm. Once it has run successfully, identifiers will be added to the table. These mode shape identifiers are a simplified version of Kindt's mode shape labelling convention



Export type: ● Current figure ○ Numeric data
Export figure
File type: PNG

Export type: ○ Current figure ● Numeric data
Export numeric data
File type: Excel
Numeric data to export: Eigenfrequencies, Force response, RMS values

(a) Figure

(b) Numeric

Figure B.8: Export window.

[B.2] and are explained in appendix B.A.2. Besides, Appendix B.A also shortly explains what the algorithm requires for input to work.

B.4.6 Image viewer

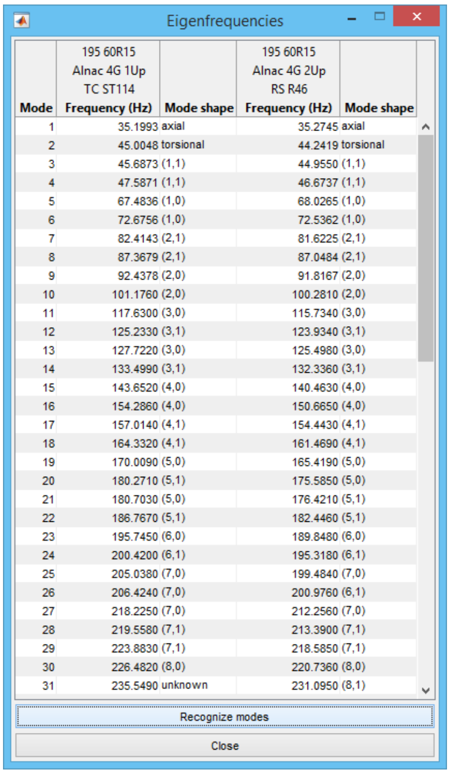
If images are available in the selected data-folder, the corresponding buttons in the *Show image(s)*-panel become active. There are three different image types, of which the image viewer windows have the following options:

- *Footprint contact pressure images*: Ability to show one image, or two side-by-side when having more that one data-folder selected. Within the window there are drop-down menus which allows the user to toggle between the different data-folder images. Lastly also the footprint contact area in the lower left corner of the figure(s) is mentioned. An example of a side-by-side image viewer window can be seen in figure B.10a.

- *Mold vs inflated contour image*: Analog to the previously described image-viewer.

- *Mode shape images*: Compared to the previous two, this version has some extended capabilities. Here is an overview of these additional capabilities with an explanation on how to use them:

  - Instead of just being able to show two images side-by-side, the mode shape image viewer is capable of showing 4 mode shape images.
  - All the available mode shapes can be viewed by going through the drop-down list top-right of the mode shape image.
  - Per mode shape, three different views are available. These views consist of a isometric, front and side view, and can be selected in the drop-down menu on the left side of the mode number drop-down list.



	195 60R15 Alnac 4G 1Up TC ST114		195 60R15 Alnac 4G 2Up RS R46	
Mode	Frequency (Hz)	Mode shape	Frequency (Hz)	Mode shape
1	35.1993	axial	35.2745	axial
2	45.0048	torsional	44.2419	torsional
3	45.6873	(1,1)	44.9550	(1,1)
4	47.5871	(1,1)	46.6737	(1,1)
5	67.4836	(1,0)	68.0265	(1,0)
6	72.6756	(1,0)	72.5362	(1,0)
7	82.4143	(2,1)	81.6225	(2,1)
8	87.3679	(2,1)	87.0484	(2,1)
9	92.4378	(2,0)	91.8167	(2,0)
10	101.1760	(2,0)	100.2810	(2,0)
11	117.6300	(3,0)	115.7340	(3,0)
12	125.2330	(3,1)	123.9340	(3,1)
13	127.7220	(3,0)	125.4980	(3,0)
14	133.4990	(3,1)	132.3360	(3,1)
15	143.6520	(4,0)	140.4630	(4,0)
16	154.2860	(4,0)	150.6650	(4,0)
17	157.0140	(4,1)	154.4430	(4,1)
18	164.3320	(4,1)	161.4690	(4,1)
19	170.0090	(5,0)	165.4190	(5,0)
20	180.2710	(5,1)	175.5850	(5,0)
21	180.7030	(5,0)	176.4210	(5,1)
22	186.7670	(5,1)	182.4460	(5,1)
23	195.7450	(6,0)	189.8480	(6,0)
24	200.4200	(6,1)	195.3180	(6,1)
25	205.0380	(7,0)	199.4840	(7,0)
26	206.4240	(7,0)	200.9760	(6,1)
27	218.2250	(7,0)	212.2560	(7,0)
28	219.5580	(7,1)	213.3900	(7,1)
29	223.8830	(7,1)	218.5850	(7,1)
30	226.4820	(8,0)	220.7360	(8,0)
31	235.5490	unknown	231.0950	(8,1)

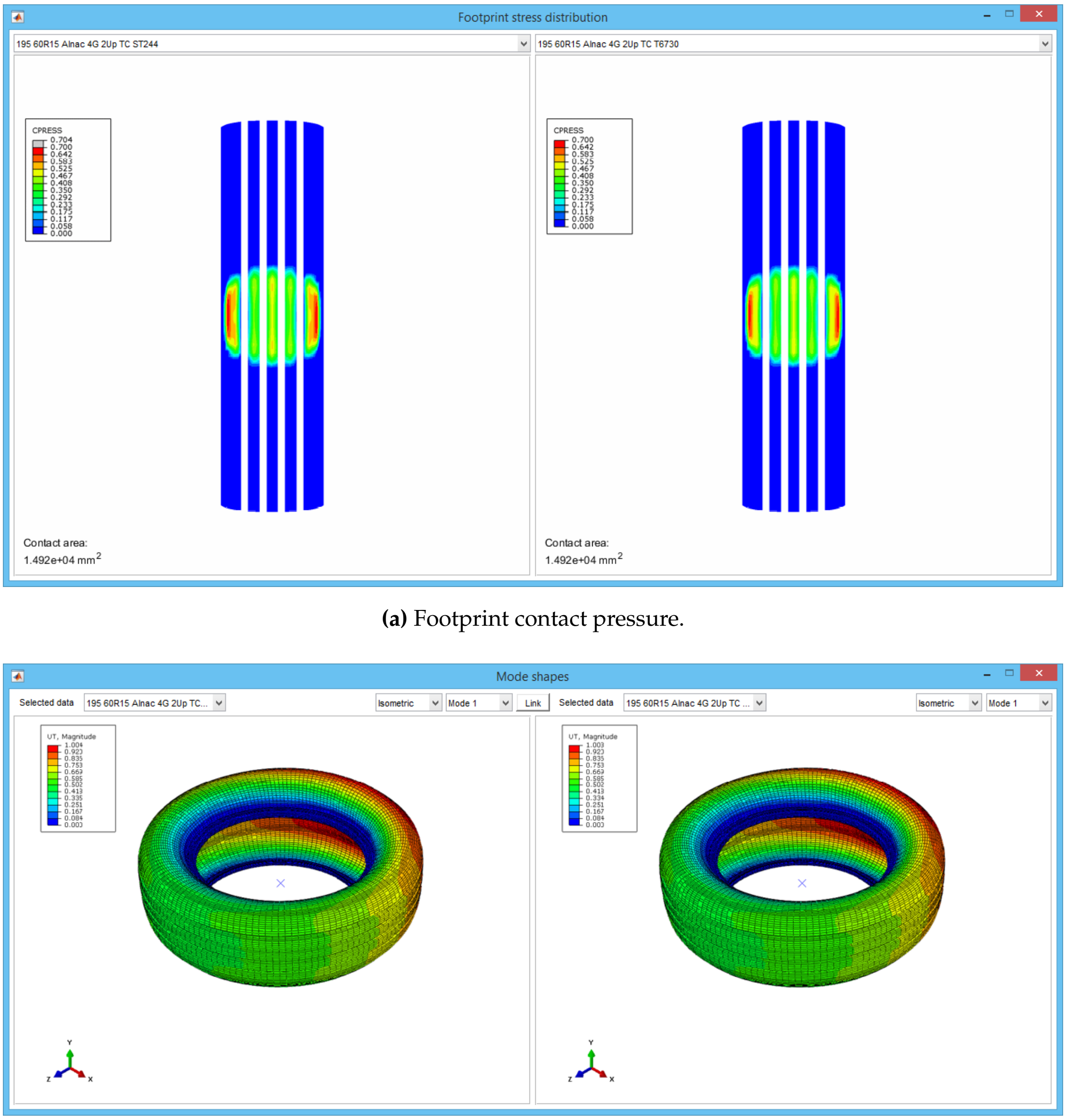
Figure B.9: Eigenfrequeny table including the recognized mode shapes.

- Finally the ability to link the views has been added. This mode is activated and deactivated by pressing the *Link*-button in the middle of the top header. By having this Link-mode active, every change in mode number or view is synchronised between all the images.

B.4.7 Transient data processing

For the processing of transient data a separate module was created. Within this module the user can post-processes the transient data, and analyse the data visually.

As input the transient analysis only requires one file, namely a `.txt`-file that contains the reaction force in vertical direction per time step. Appendix B.B explains how to obtain this data from a transient analysis `.odb`-file. The prefix of this file is set to `transient` by default, and is changeable within the import-options. In a similar fashion, the transient data-file has to be located in a separate data-folder.



Footprint stress distribution

195 60R15 Alnac 4G 2Up TC ST244

195 60R15 Alnac 4G 2Up TC T6730

Contact area:
1.492e+04 mm^2

Contact area:
1.492e+04 mm^2

(a) Footprint contact pressure.

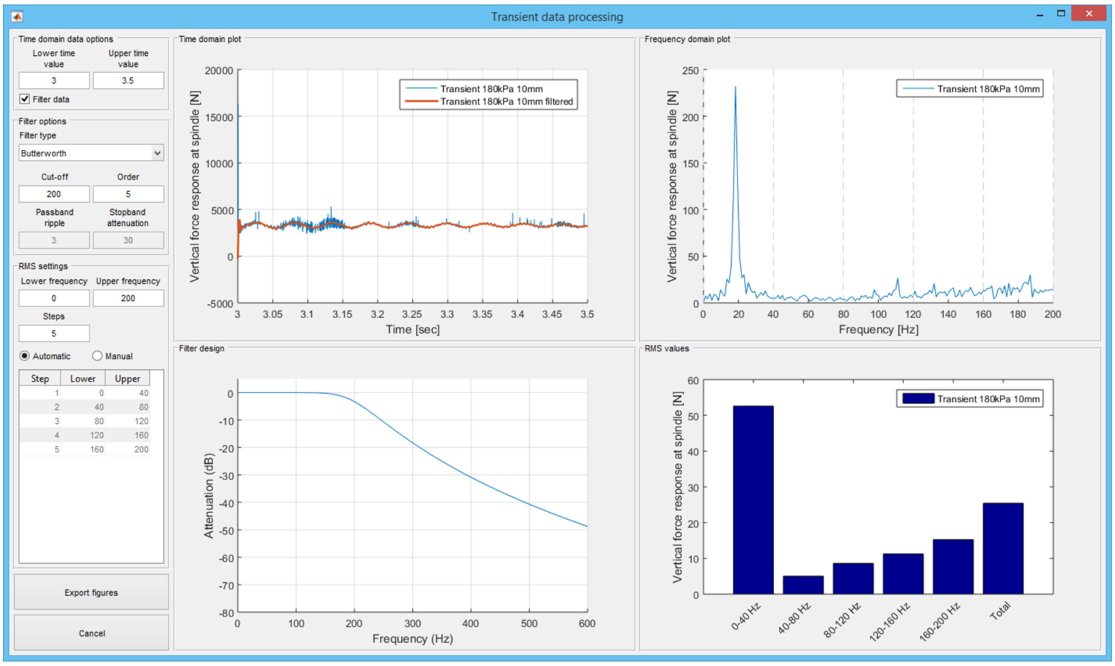
Mode shapes

Selected data: 195 60R15 Alnac 4G 2Up TC... | Isometric | Mode 1 | Link

Selected data: 195 60R15 Alnac 4G 2Up TC ... | Isometric | Mode 1

(b) Mode shapes

Figure B.10: Image viewers.



Transient data processing

Time domain data options

Lower time value: 3

Upper time value: 3.5

☑ Filter data

Filter options

Filter type: Butterworth

Cut-off: 200

Order: 5

Passband ripple: 3

Stopband attenuation: 30

RMS settings

Lower frequency: 0

Upper frequency: 200

Steps: 5

● Automatic ○ Manual

Step	Lower	Upper
1	0	40
2	40	80
3	80	120
4	120	160
5	160	200

Figure B.11: Transient post-processing window.

Getting started

To open the transient data processing module, first data-folders containing transient-data have to be selected in the main window. Once this is done, and the transient-data file has been found by the application, the *Transient data (FFT)*-button becomes active. Pressing this button opens the module that has a similar lay-out compared to the main application window. One of the major differences is that instead of showing only one figure, it shows four figures at the same time, see figure B.11. These figures can be observed:

- Raw time-force data, located in the top-left panel, named as *Time domain plot*. Within this figure the filtered data can also be observed if a filter is active.

- Filter design, located in the bottom-left panel. Visual representation of the selected filter, according to the settings as set in the *Filter options*-panel.

- Transformed frequency-force data, located in the top-right panel, named as *Frequency domain plot*. This figure shows the processed data, more on this later.

- RMS-value bar chart, located in the bottom-right panel. Similar to the low-frequency modal analysis RMS-values, these values get calculated from the frequency-force response data. The settings of which can be set in the *RMS settings*-panel on the left-side of the application.

Data processing settings

As has become clear already from the previous paragraph, the left-side of the application contains all the transient data post-processing options. Any alteration done within one of these options gets applied real-time to the relevant figures on the right. The following options are available:

- Time data can be cropped by altering the *active* time range within the *Time domain data options*-panel.
- Data can be filtered with one of the four available filters, the filter can be set on or off by (un)checking the *Filter data*-checkbox. The available filters are: Butterworth, Chebyshev type I, Chebychev type II and elliptic. Every filter can be fine-tuned with their corresponding settings, such as cutt-off frequency, filter order, passband ripple and stopband attenuation. When applicable these options become activated, otherwise they are deactivated.
- The RMS setting are similar to those explained in section B.4.3, and hence will not be explained here again.

Finally, the independant figures and numeric data can also be exported by pressing the *Export figures*-button in the bottom-left corner of the application. This shows a window similar to the one of the main application window, as described section B.4.4. The only difference here is that the user has to select the figure that has to be exported instead the current figure, the difference is self-explanatory.

B.A Mode shape recognition

As said, the mode shape recognition is an algorithm that automatically labels the mode shapes. The labels used are a simplified version of Kindt's labelling convention, which is explained in the next paragraph.

The input of the algorithm consists of coordinate and displacement data from the `SYMMETRY` node-set, in the x, y and z-direction. This data consists of

- the reference coordinate locations of a tyre showing no mode shape (coordinate-data);
- and the relative displacements in all three directions per mode shape (displacement-data).

The coordinate and displacement data-files are prefixed by default with *coordinates* and *displacements* respectively. The prefixes can be altered in the import settings.

B.A.1 Request output from Abaqus

To get request the data from the Abaqus simulation, the following output-request code has to be added to the input-file of the frequency extraction step:

```
*OUTPUT, FIELD
*NODE OUTPUT
 U, COORD
```

This adds the coordinate and displacements outputs to the `.odb`-file.

B.A.2 Mode shape labelling convention

The label consists of two numbers and is formatted as (n,a). In this notation, the indices represent:

n = circumferential index;

a = belt cross-sectional index.

To elaborate further, the circumferential index (n) is the amount of bending wavelengths in the radial direction of the tyre. While the belt cross-sectional index (a) stands for the amount of bending wavelengths in the lateral direction. Figure B.12 shows visual representations of these indices. It must be noted that this representation does not encapsulate all the mode shapes. Firstly, the rigid body motions of the belt can not be indicated with the label convention. Instead, labels such as *axial* and *torsional* are used.

Secondly, nearly all of the mode shapes appear double, but at a different orientation. The is due to the axisymmetric nature of the tyre. To capture this, Kindt introduced extra identifiers which capture these modes. Due to the complexity of recognising these different mode shapes these

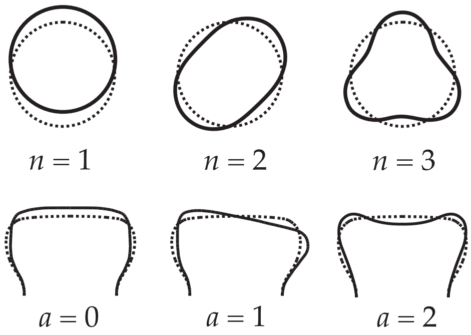


Figure B.12: Visual representation of the mode shape labelling convention indices (illustration courtesy of Peter Kindt, obtained from his doctoral dissertation [B.2]).

extra identifiers have been left out. Hence making this a *simplified* version of Kindt's labelling convention.

Limitations

Firstly it must be noted that the algorithm is incapable of recognising any mode shape with a lateral bending wave of higher than one ($a > 1$). This is due to the complexity of the shape. Consequently this algorithm can be come inaccurate at the transition point where these second order lateral bending waves start to become apparent. After a specific point, any mode shape which cannot be recognised with the algorithm is labelled with *unknown*.

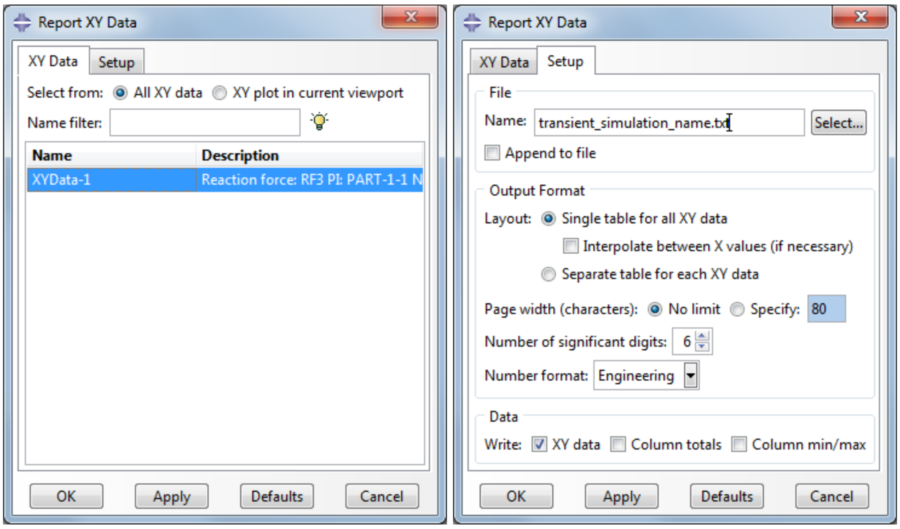
B.B Transient data extraction

To be able to analyse the transient results, firstly the required data has to be extracted from the `.odb`-files. Since this procedure has not been automated the user will have to manually perform some steps to do so, this step-by-step procedure will be explained in the following section.

B.B.1 Step-by-step guide

To get started, firstly the `.odb`-file of a transient simulation has to be opened. This is done either by starting Abaqus/viewer first and opening it through there or by double clicking the `.odb`-file in the file-explorer. Once the correct database-file has been opened the next steps can be followed:

1. In the upper file menu bar navigate to: *Result > History Output...*, this should open the History Output window;
2. Next, select the reaction force of the node set *road* in vertical direction, which is the 3-direction in this case: the list item should look something like `Reaction force: RF3`



Select from: ● All XY data ○ XY plot in current viewport

Name filter:

Name	Description
XYData-1	Reaction force: RF3 PI: PART-1-1 N

File

Name: transient_simulation_name.txt

□ Append to file

Output Format

Layout: ● Single table for all XY data

□ Interpolate between X values (if necessary)

○ Separate table for each XY data

Page width (characters): ● No limit ○ Specify: 80

Number of significant digits: 6

Number format: Engineering

Data

Write: ☑ XY data □ Column totals □ Column min/max

(a) XY data tab

(b) Setup tab

Figure B.13: Report XY data window.

... in NSET ROAD. Now press *Save as...* in the lower left corner, this should bring up a new window, namely *Save XY Data As*.

3. In this next window the name of the XY-data can be changed if the user desires, but this is not required as long as the user remembers the default name it is given. Finally *OK* can be pressed to save the XY-data internally within Abaqus/Viewer.

4. Now, within the upper file menu, navigate to *Report* > *XY...*. This opens the Report XY Data window.

5. Select the earlier saved XY-data (figure B.13a), and afterwards open the tab *Setup*.

6. Here the file-name has to be changed to a name that preferably starts with `transient_` and ends with the `.txt`-extension, as both these are the default import-parameters of the toolbox (figure B.13b). Finally, prior before saving, the *Append to file*-option has to be deselected, whereafter *OK* can be pressed to save the file.

Please note that the file is saved from the location where Abaqus/Viewer is launched from. When launching it with the short-cut within for example the start menu, the file gets saved in the `Temp` folder (for Windows PCs that is most likely at `C:\Temp\`). Otherwise the file gets saved in the folder in which the `.odb`-file is located.

References

[B.1] Daintith, J. (2009). A dictionary of physics. Oxford New York: Oxford University Press.

[B.2] Kindt, P. (2009). *Structure-Borne Tyre/Road Noise due to Road Surface Discontinuities*. Leuven.

Appendix C

Queue script manual

C.1 Introduction

The queue script is written to improve the way Abaqus simulation jobs are queued for execution. The script has been written in Python and has to following capabilities:

- Partial automization of the queue process;
- Preliminary check if all the queued jobs are correctly inputted and if all the input-files exist;
- Confirmation query if to run all the jobs are shown in the overview.

First, an explaination will be given on how to set-up a simulation batch queue within the `queue_script.py` and secondly how to execute it through a command line interface (CLI) accompanied with an example.

C.2 Set-up

Within the queue-script the part *User input* contains all the input-parameters that the script has, namely:

- `ABQ`: The Abaqus command with its correct version with which to run the simulations, for example `abq6131`.
- `flags`: The jobs execution flags which are appended to every job that will be run, for example `int memory=40`. N.B. the flag `int`, which stands for interactive, is required to allow to queue-script to properly function.
- `NCORE`: Amount of CPU cores to use for the 3D-footprint simulation (`REV`), normally either 4 or 8 CPU cores. Please note that the 2D-axisymmetric simulation is always run with the use of only one CPU core.

- `AXISuffix` and `REVSuffix`: Both the suffixes use for the 2D-axisymmetric simulation (`AXI`) and the 3D-footprint simulation (`REV`) respectively.
- `jobs`: Array that defines which jobs the run, containing a multiple of 4 entries. Its exact format will be explained in the next paragraph.

C.2.1 Input-format of `jobs`

The jobs array's format is as follows:

```
jobs = [ '<folder>', '<job name>', <Run AXI>, <Run REV> ]
```

Where:

- `<folder>`: specifies the folder in which the input-files are located, in string format. By inputting the string `current` the current directory is used.
- `<job name>`: defines specifically which job to run. Insert the job-name in string format without the `AXI` & `REV` suffix. Beside specifically defining the job also the string `all` can be input, this way the script automatically find all the input-files within the specified folder and checks if both the corresponding input-files are found when both the `AXI` & `REV` simulations are set to run.
- `<run AXI>`: boolean-parameter specifing whether or not to run the 2D-axisymmetric simulation (`AXI`), hence either the input `True` or `False` respectively.
- `<run REV>`: boolean-parameter specifing whether or not to run the 3D-footprint simulation (`REV`), hence either the input `True` or `False` respectively.

Please note that all string have to be enclosed with quatation marks and that the `True` or `False` booleans need to have their first letter as a capital letter.

C.3 Script execution

To execute the queue script, simply the following rule of code has to be typed in the CLI

```
> abq6131 python queue_script.py
```

This executes the script and outputs directly to the CLI where the overview of jobs to run will show up, whereafter the user is asked whether to continue or not. This query is as follows:

```
Do you want to continue to run all the shown jobs? [Y/n]
```

Since the `Y` is a capital letter this means that this is the default decision, hence simply pressing enter is interpreted as a yes. When the user does not want the queue script to queue the jobs simply type in `n` and press enter, this aborts the script.

C.3.1 Examples

Here are some examples of `jobs`-arrays, as defined in the `queue_script.py`. First an example of a single job of a 175/65R14 dualply variant, where only the 2D-axisymmetric simulation is run:

```
jobs = [ '175_65R14', '175_65R14_Dualply', True, False ]
```

An example output of the queue-script to the CLI by running it with this `jobs`-array is shown in listing C.1, here the entire procedure and output can be observed.

```
------------------------------------------------------------
 ABAQUS JOB QUEUE SCRIPT
------------------------------------------------------------

 PRELIMINARY CHECK
  Performing a preliminary check if all the specified
  folders and input-files exist.

  |- Folder 175_65R14 found
  |-- File 175_65R14_Dualply_AXI.inp found

 The following jobs can be queued up to run:
  1: 175_65R14_Dualply_AXI.inp file(s) in folder 175_65R14

 Do you want to continue to run all the shown jobs? [Y/n] y

------------------------------------------------------------
 Running the 175_65R14_Dualply_AXI.inp job with 1 CPU core
------------------------------------------------------------

[Abaqus simulation output]

Run time: .. seconds

------------------------------------------------------------
 END OF QUEUE SCRIPT
------------------------------------------------------------
```

Listing C.1: Complete output to the CLI when letting the queue-script to actually run the displayed job.

As a second and final example a run of a multiple jobs run is shown, here both the 2D-axisymmetric and 3D-footprint simulations will be specified to run:

```
jobs = [ '175_65R14', '175_65R14_Flatbelt', True, True,
         '195/60R15_Alnac_4G', 'all', True, True ]
```

Bibliography

[1] D.A. Bekke. *Engineering tools for interior tyre tread pattern noise*. Enschede, first edition, 2014.

[2] Maik Brinkmeier and Udo Nackenhorst. An approach for large-scale gyroscopic eigenvalue problems with application to high-frequency response of rolling tires. *Computational Mechanics*, 41:503–515, 2008.

[3] Dassault Systèmes Simulia Corp. Abaqus 6.13 Documentation. 2013.

[4] J. Daintith. A dictionary of physics. 2009.

[5] P. Kindt. *Structure-Borne Tyre / Road Noise due to Road Surface Discontinuities*. Number June. 2009.

[6] R.S. Marlow. A General First-Invariant Hyperelastic Constitutive Model. *Constitutive Models for Rubber III*, pages 157–160, 2003.

[7] H.R. Dorfi R.L. Wheeler and B.B. Keum. Vibration modes of radial tires: Measurement, prediction, and categorization under different boundary and operating conditions. 2005.

[8] U. Sandberg and J.A. Ejsmont. *Tyre, Road Noise: Reference Book*. Informex, 2002.

[9] Robert D. Cook; David S. Malkus; Michael E. Plesha; Robet J. Witt. *Concepts And Applications of Finite Element Analysis*. Wiley, Madison, 4th edition, 2002.