Geometrical Design of Insulated Rail Joints

Models for Dynamic Performance Evaluation

Leander Engel

Delft University of Technology | Witteveen+Bos

Geometrical Design of Insulated Rail Joints

Models for Dynamic Performance Evaluation

by

Leander Engel

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday October 29, 2024 at 13:45.

Student number:	4295412	
Thesis committee:	Dr. ir. M.J.M.M. Steenbergen,	TU Delft, supervisor
	Dr. ir. K. N. van Dalen,	TU Delft
	Dr. ir. Marieke Bezemer,	Witteveen+Bos

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of Master of Science in Civil Engineering at Delft University of Technology. Over the past year, I have encountered numerous challenges that have helped me grow, not only as a structural engineering student but also as an individual. This journey has taught me much about the subject of engineering, yet even more about the broader responsibilities and skills required of an engineer.

The research presented here was conducted in collaboration with Witteveen+Bos, whose project served as the inspiration for my thesis topic. I am grateful for the opportunity to work within their company, where I always felt welcomed and supported. I would like to express special thanks to my supervisor, Marieke Bezemer, for her guidance. She was consistently available for discussion and provided me with many new and insightful perspectives. Her enthusiasm and expertise enriched my experience.

I also wish to express my appreciation for Karel van Dalen. Though he was less frequently involved, his personal support during key moments was of great importance. He always knew how to lighten the tone of formal meetings without losing focus, which was a refreshing and much-appreciated quality.

My sincere gratitude goes to the chair of my committee, Michael Steenbergen, for his dedicated involvement in this project. His weekly hour-long meetings were crucial in shaping the direction of my thesis. I was consistently impressed by his vast knowledge and efficiency. His attention to precision and the efficiency of his workflow has had a lasting impact on me.

I would also like to warmly thank my family. First, my mother: although our weekly 'meetings' were not about thesis content, they were crucial for emotional support. I could always call her, just as I have been able to do my whole life, and her constant presence has played a significant role in bringing me to this point. Second my father: as a physics teacher, he has inspired me from a young age to pursue this path. No matter where we were, he could always point out an interesting physical phenomenon, which undoubtedly influenced my career choice. Lastly my brother Julian: even though he is younger, I have always felt I could turn to him for conversations. His social skills and ability to connect with people are qualities I admire and continue to learn from. Finally, I am especially grateful for the health and well-being of my family, particularly after the challenges we faced this past year.

Leander Engel
Rotterdam, October 2024

Summary

Insulated rail joints (IRJs) play a crucial role in modern railway systems. They serve the critical function of electrically isolating rail segments through the placement of an insulating material, known as an end plate, between two rail ends. This insulating material is necessary to define track segments, which makes it possible to determine the position of trains within the railway system. Knowing a train's position is key to ensuring efficiency, reliability, and safety. While these joints are highly important, they are also vulnerable. The interruption in rail geometry results in a complex interaction between wheel and rail, giving rise to high dynamic impact forces. Traditional IRJs, or squared IRJs, have the cut between the rail ends orthogonal to the rail. In this thesis, an alternative design with a non-orthogonal junction angle is analyzed.

The primary goal of this thesis is to determine how the junction angle influences both the global wheel-rail interaction and the local contact pressure at the wheel-rail interface. To achieve this, the thesis is split into two parts: (1) the global wheel-rail interaction analysis, which studies the influence of the junction angle on the interaction between the wheel and rail using simplified geometries in a kinematic approach, and (2) a local wheel-rail interface analysis, which studies the effect of the junction angle on an assumed uniform contact pressure between wheel and rail.

The global analysis revealed the possibility of two distinct contact scenarios, depending on lateral wheel position and dip angles greater than zero. In contact scenario 1, the effective geometry and the resulting vertical impulse remained identical to those of squared joints. However, in contact scenario 2, the active geometry of the joint changes, leading to an increase in vertical impulse of the wheel's center of mass. Additionally, the introduction of the junction angle increased the likelihood of less favorable contact conditions for contact scenario 1 and guaranteed less favorable contact conditions for contact scenario 2. The local analysis showed that uniform contact pressure between the wheel and rail increases slightly for non-orthogonal junction angles with dip angles near zero. For small junction angles (resulting in a long cut in the longitudinal direction), outside of the practical range, the rate of change of the contact pressure was greatly reduced.

The study has shown that insulated rail joints with non-orthogonal junction angles within the practical range do not provide significant improvements in dynamic performance compared to traditional squared joints. However, due to the assumptions made in this model, the complexity of the rail geometry was significantly simplified, and material elasticity was not considered. These limitations are expected to affect the contact behavior and could effect the results. This should be investigated further. The second model demonstrates that for junction angles within the practical range, the assumed uniform contact pressure increased slightly. However, for very small junction angles, which result in impractically elongated joints, the rate of change in uniform contact pressure can be greatly reduced.

Contents

List of Figures

List of Tables

1 Introduction

1.1. Background information

Railway systems play a crucial role in efficiently transporting people and goods across regions. Their rapid and large-scale transportation contributes greatly to urban accessibility. Their importance is a result of their high speed, safety, reliability, cost-effectiveness and environmental advantages over road transportation [1]. High speed rail is especially effective for the transportation of people between densely populated urban areas [2]. To maintain the effectiveness of this important transport network, it is essential to ensure the safety and reliability of its components. This is especially challenging in high-speed rail systems where the higher velocities place greater demands on these components. Despite improvements to various system components in the past decades, some remain vulnerable to damage, particularly those associated with imperfections in rail geometry.

Imperfections are an unwanted but unavoidable phenomenon in railway systems. They lead to undesirable effects such as noise, vibration and increased maintenance costs for wheels and tracks. These imperfections occur in various places along the railway track systems, the most common ones being: joints, crossings, switches, turnouts, tunnels and bridges. In response to these undesirable effects, the number of joints in the track has been significantly reduced in recent years, and continuously welded rail (CWR) has become the norm [3]. However, insulated rail joints (IRJs) are still widely used since they fulfil an important role in train signalling as they are required to define track circuits.

A major problem arises when the wheel-rail contact has a spatial discontinuity. A discontinuity occurs, for example, in dipped joints, IRJs and bolted rail joints. These joints have a smaller vertical bending stiffness compared to the rail [4], [5], [6]. This results in a difference in deflection when a train approaches the railway joint. This difference in deflection generates the dip-angle. When a train crosses the joint with a dip-angle, a complex interaction between the train wheels and the joint happens in which large impulse noise peaks [7] and high dynamic forces arise [8]. The forces that arise from this interaction will damage the joint increasing maintenance costs and reducing reliability of the railway system.

In railway infrastructure, IRJs play a critical role in segmenting track circuits for signal and control systems. An electric current is fed to both rails at one side of the segment. When the segment is unoccupied the electric current can be detected on the other end of the section. When a train enters the section, it short-circuits the current in the rails. The standard layout of IRJs typically involves the use of insulating materials between adjoining rail ends to electrically isolate sections of the track. In standard IRJs, this insulating material is placed in the rail, perpendicular to the rolling direction of the train. This means the longitudinal direction of the rail and the insulation material are at an angle of 90 degrees as seen in the last figure in Figure 1.1. However, in past decades the development of insulated rail joints with a non-orthogonal junction angle has gained attention due to their potential to enhance the performance and longevity of IRJs. Unlike the traditional perpendicular IRJs, IRJs with a non-orthogonal junction angle have an angled interface between the rail ends. In Figure 1.1 in the first and second images, two insulated rail joints with non-orthogonal junction angles are shown. According to experiments [5], [9], joints with a non-orthogonal junction angle reduce noise and dynamic impact and improve the structural integrity of the joint.

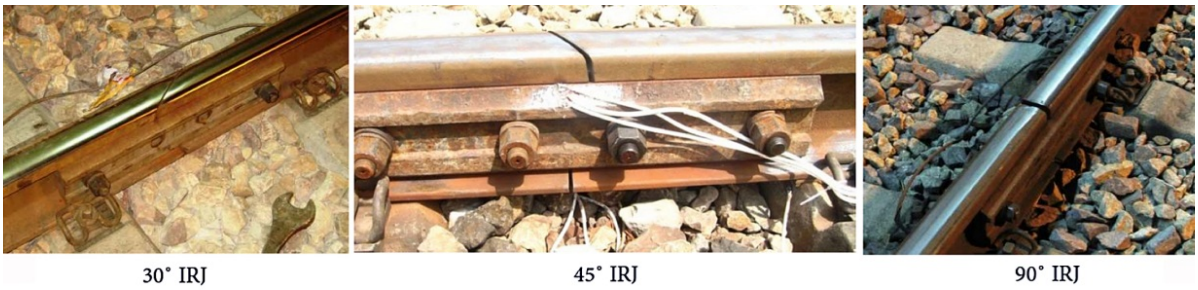


Figure 1.1: Insulated rail joints with 30, 45 and 90 (perpendicular) cuts

1.2. Problem definition

Insulated rail joints (IRJs) are a weak link in modern-day railway systems. A local reduction in stiffness and an interruption in track geometry give rise to high dynamic impact forces. These forces are a cause of quick track deterioration which increases the maintenance required. To minimize dynamic impact forces and extend the service life of IRJs, an alternative design has been developed: the insulated rail joint with a non-orthogonal junction angle. In this design, the end plate (the insulation material between two rail segments) is not positioned perpendicular to the rail's direction as it would be in traditional joint designs, but it is placed diagonally in the rail. In Figure 1.2 a schematic of an IRJ with a non-orthogonal junction angle is presented. While experimental studies have demonstrated that this joint design offers improved structural performance, there is still a lack of comprehensive theoretical research on this topic.

1.3. Aim of the thesis

The aim of this thesis is to investigate the influence of the junction angle of an insulated rail joint on the global and local wheel-rail interaction. This research seeks to develop a better understanding of how varying the junction angle affects the dynamic behaviour of the rail joint and the contact pressure at the wheel-rail interface.

The main objective of this thesis is:

- Analyze the effect of the junction angle on the global and local wheel-rail interaction in insulated rail joints, to enhance theoretical understanding that can aid in design optimization.

The sub-objectives are to:

- Analyze the global dynamic behaviour of the wheel-rail interaction.
- Analyze the local contact pressure at the wheel-rail interface.

1.4. Research question

Research question:

- **What is the influence of the junction angle (θ) on the time-variant wheel-rail interaction in insulated rail joints?**

To assess the influence of the junction angle on the global wheel-rail interaction, it is crucial to describe the trajectory of the wheel's center of mass in time $z(t)$. This time-dependent trajectory reflects the dynamic characteristics of the interaction and provides a basis for determining the magnitude of the resulting impact.

Sub-questions global model:

1. Does a parameter space exist for the junction angle of the rail $(0 < \theta \leq 90)$, a dip angle larger than zero $(\beta > 0)$ and the wheel modeled as an infinitely thin disc, such that the first derivative of the trajectory $(z(t))$ of the moving disc's center of mass is continuously defined in a kinematic approach?
2. If this solution does not exist, which combination of parameters, as mentioned in sub-question 1, results in the smallest variation in the first derivative of the trajectory $z(t)$?
3. What is the influence on the trajectory when the finite width of the rigid wheel is taken into account?

Sub-questions local model:

1. How does the junction angle of an insulated rail joint (θ) affect the magnitude of an assumed uniform contact pressure (p), when an elliptical contact patch moves uniformly over an insulated rail joint with a nominal gap (g)?
2. How does the junction angle (θ) of an insulated rail joint affect the rate of change in uniform contact pressure (p) over time, as an elliptical contact patch moves uniformly over a rail joint with a nominal gap (g)?

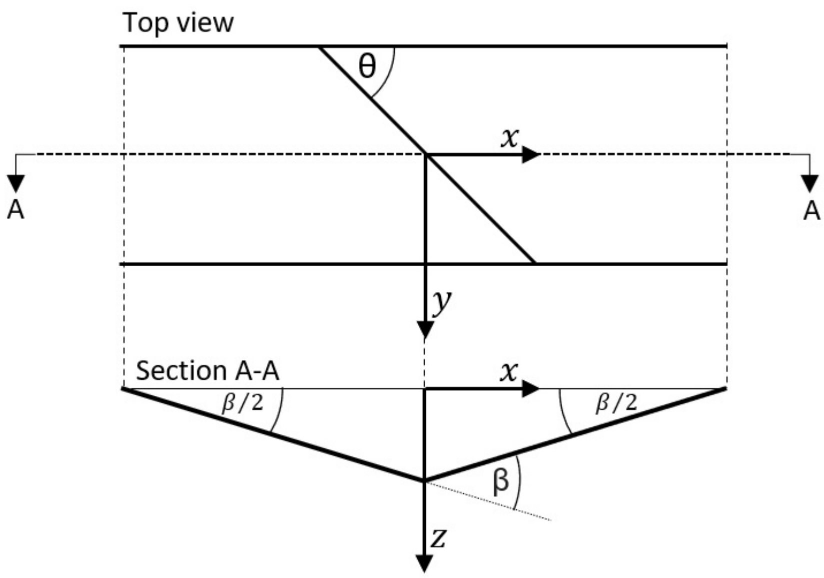


Figure 1.2: Top-view and cross-section of an insulated rail joint, illustrating junction angle θ and dip angle β.

1.5. Structure of the report

This report is divided into seven chapters:

- **Chapter 1: Introduction**
  This chapter provides an overview of the research, presenting background information, problem definition, and the aim of the thesis. It outlines the research questions and the objectives of the study.
- **Chapter 2: Theoretical Background**
  This chapter introduces the theoretical foundations of railway engineering relevant to this thesis.
- **Chapter 3: Global Analysis: Wheel-Rail Interaction**
  Here, the global dynamic behavior of the wheel-rail interaction is explored. The chapter presents the kinematic approach to wheel-rail interaction analysis and introduces the framework and models used for the global analysis.

- **Chapter 4: Results of Global Wheel-Rail Interaction**
  This chapter provides the results of the global analysis, focusing on the dynamic properties of the system, including vertical impulse and the visualization of contact points during double-point contact.
- **Chapter 5: Local Analysis: Wheel-Rail Interface**
  This chapter shifts focus to the local wheel-rail interaction, specifically analyzing contact pressure at the wheel-rail interface.
- **Chapter 6: Results of Local Wheel-Rail Interface Analysis**
  The results of the local contact pressure analysis are presented in this chapter.
- **Chapter 7: Conclusions and Recommendations**
  The final chapter summarizes the findings of the thesis and provides recommendations for future research.

The report also includes an appendix, which contains supporting material such as a literature study and additional calculations.

2
Theoretical background

2.1. Static and dynamic analysis in railway engineering

In railway engineering, the structural integrity, safety and performance of the railway components are dependent on various forces acting on the system. These forces can be categorized into static and dynamic forces which differ in nature and application. To get a broad understanding of how the system performs it is crucial to conduct both static and dynamic analyses.

2.1.1. Static forces

Static forces are the forces that remain constant or vary very slowly over time. In railway engineering they are related to the self-weight of the train, tracks, and other components of the system. The most common example of a static force is the self-weight of the train acting on the track. This load is predictable and remains constant as the train traverses the track.

2.1.2. Dynamic forces

Dynamic forces are the those that vary rapidly over time due to changes in speed, acceleration and the interaction between the train and the track. These forces occur when the train is in motion and are influenced by track irregularities, acceleration, deceleration and aerodynamic effects. Modelling of dynamic forces is generally more complex than static forces and often require a good understanding of the wheel-rail interaction. The dynamic force on the system may be expressed relative to the static force. According to some analyses, the dynamic forces in railway joints can increase the load on the system be a factor 4-6 compared to the static load [7].

2.1.3. Total Force

The total force acting on the railway system is a combination of the static and dynamic force:

$$F_{total} = F_{static} + F_{dynamic}$$

The summation of the forces provides a complete description of the load acting on the railway structure. The static analysis alone is used for many applications but dynamic analysis becomes more and more crucial in modern, high speed rail networks where the dynamic forces become more and more significant.

2.2. Beam on Wrinkler foundation and dipped joint geometry

The combination of rail and underlying soil can be modelled using the Wrinkler foundation model in combination with a beam element. The rail is modelled as a beam element with bending stiffness EI and is placed on a continuous vertical spring with spring constant k, which represents the ballast and subgrade. The model is used to calculate the rail deflection, bending moments, and shear forces in the rail under train loads. The governing equation of the deflection of the beam on elastic foundation is given by:

$$\frac{d^4w(x)}{dx^4}EI + k \cdot w(x) = F(x)$$

where $w(x)$ is the deflection of the beam at position x, k is the stiffness of the subgrade, E is the young's modulus of the beams material, I is the moment of inertia of the beam and $F(x)$ is the applied load as a function of x. In Figure 2.1 the beam on an elastic foundation is shown if no load is applied. For continuous rail, the bending stiffness of the system remains uniform along the track. When a train load is applied, the rail behaves like a beam, resulting in smooth deflection due to its constant bending stiffness. However, this condition changes in the presence of a rail joint. At the joint, the bending stiffness is no longer uniform but experiences a localized reduction. As a result, the deflected shape of the rail near the joint deviates from the smooth profile seen in continuous sections. Instead, a discontinuity, forms at the joint, producing what is commonly referred to as a dipped joint geometry. In Figure 2.2 the model is shown with and without joint. The image also shows the resulting dipped joint geometry. The dipped joint geometry follows from the local displacement field of the system at the location of the joint and is often used in literature to describe the rail's deformation near the joint. In well known paper [10], Jenkins et al. introduced the dipped joint geometry and defined the dip angle which described the gradient of the rail at the point where they connect. Two definitions are commonly used to describe the dip angle: it can either refer to the angle between the horizontal and the rail at the point where the rail ends connect, or the angle between the two rail ends themselves. Various mathematical functions, including linear, quadratic, and sinusoidal functions, have been used to model the shape of the dipped joint. In Figure 2.3 the geometries are shown for a linear and sinusoidal function.

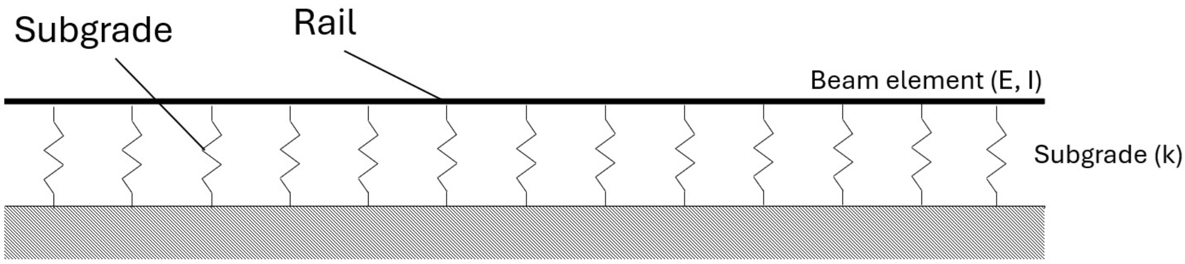


Figure 2.1: Beam element and Wrinkler foundation when no load is applied

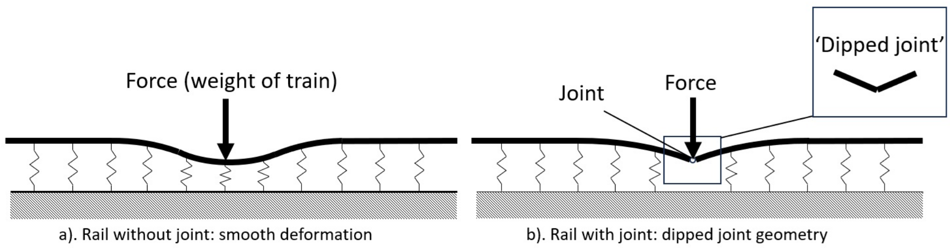


Figure 2.2: Beam element and Wrinkler foundation when load is applied for a). continuous beam element and b). beam element with joint

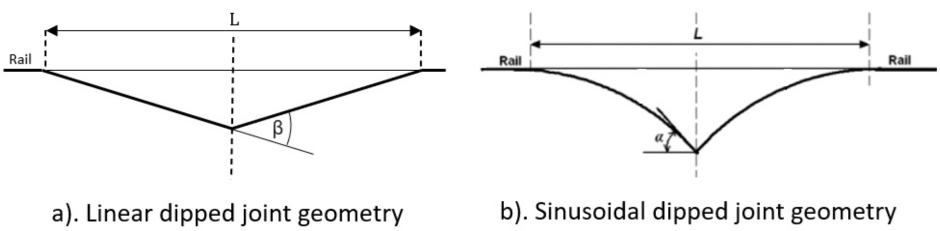


Figure 2.3: Linear and sinusoidal dipped joint geometries

2.3. Classification of wheel-rail contact

In [4] Steenbergen describes the difference between the situation that occurs when a wheel is in single-point contact or multi-point contact with the rail geometry. A perfectly circular wheel running on a perfectly flat rail continuously has a single-point contact in time. The contact patch is reduced to be a point as a result of the assumption that all materials are fully rigid and perfectly shaped. The single-point contact in time may shift to double-point contact when an imperfection in track geometry occurs, see Figure 2.4. The type of contact will only shift when the length scale of the imperfection is smaller than the diameter of the wheel or when the curvature of the vertical rail geometry exceeds the curvature of the wheel rim.

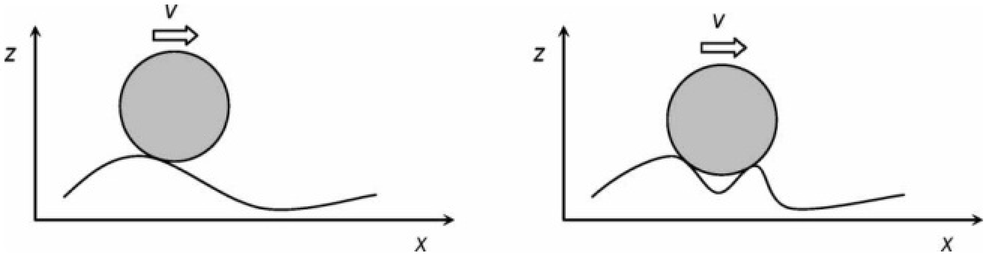


Figure 2.4: Continuous single point contact (left) and transient double-point contact (right) [4]

In the case of single-point contact, the wheel may be modelled as a lumped mass supported by a spring. The spring is used to model the Hertzian wheel-rail contact stiffness. This stiffness is nonlinear but is often assumed to be linear, which is valid under the assumption of small deviations of the load. The irregularities in the track geometry are modelled as excitation's of the wheel-rail contact point in the spring model. For a rail with height $z(x)$ the equation of motion of the wheel on the Hertzian spring becomes the same as for a simple oscillator, see Figure 2.5:

$$m_w \ddot{u}(t) + k_H u(t) = k_H z(t)$$

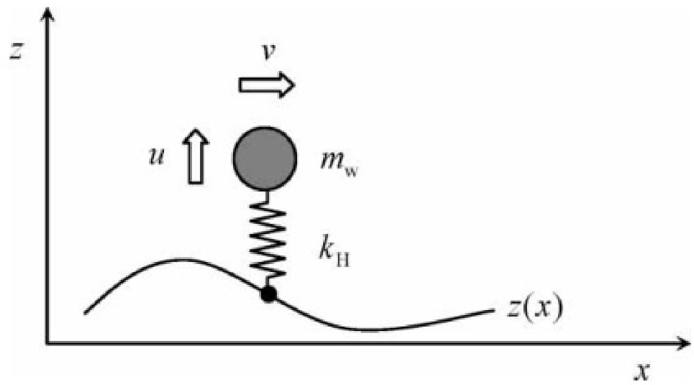


Figure 2.5: Wheel model for single-point contact [4]

In double-point or multi-point contact the kinematic relations between the perfectly shaped wheel and rail geometries have to be described to find the trajectory of the wheel's center of mass during the wheel-rail interaction. The trajectory of the wheel's center of mass depends on the type of vertical geometrical imperfection the wheel traverses. The most common ones are described in [11] and include: smooth irregularity, step-up joint, step-down joint, step-wise joints and wheel flats. In [4] Steenbergen concluded that applying the single-point contact model as described above for these type of track irregularities strongly underestimates the contact forces as it does not describe the occurring mechanisms correctly. In Figure 2.6 an example wheel-rail interaction during two-point contact is shown.

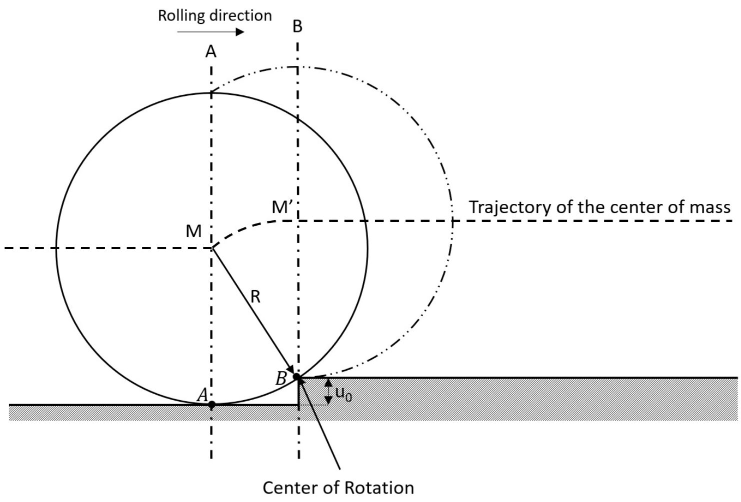


Figure 2.6: Double-point contact when a step in vertical rail geometry occurs.

2.4. Discontinuities in rigid body analysis

To facilitate the analytical analysis of wheel-rail interaction, both the wheel and the rail are considered to be fully rigid, neglecting the elasticity of the materials. This assumption leads to an instantaneous change in the direction of the wheel's center of mass during two-point contact. This instantaneous change in direction corresponds to a singularity in the vertical velocity of the wheel's center of mass, resulting in an undefined time derivative. This derivative is crucial for calculating the vertical impulse force. For a merely kinematic analysis, the impulse force remains undefined. In reality, the elasticity of the contacting bodies will smooth out the interaction, resulting in a continuously defined function for the vertical velocity, and a defined vertical impulse force. In Figure 2.7, an example of the vertical velocity is plotted, with the dotted line illustrating the smoothing effect of material elasticity.

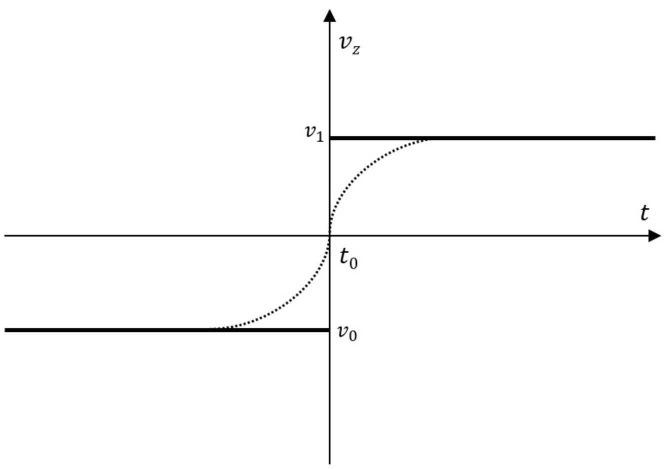


Figure 2.7: Example of discontinuous vertical velocity as a result of rigid bodies. Dotted line illustrates the smoothing effect of elasticity.

3

Global analysis: Wheel-Rail Interaction

In this chapter, the wheel-rail interaction of an insulated rail joint with a non-perpendicular junction angle is analyzed and an analytical wheel-rail interaction model is obtained using a kinematic approach. The primary objective of this analysis is to determine the effect of the junction angle θ on the dynamic performance of an insulated rail joint.

The chapter begins by providing an introduction. Then the objectives, scope, and limitations of the adopted model are discussed. Following this, a road map is outlined, guiding through the steps of the analysis. To illustrate the application of the road map, a well-known problem is solved as a demonstration. The core of the chapter presents the wheel-rail interaction model specific to insulated rail joints with a non-orthogonal junction angle. The results of this analysis will be presented in chapter 4.

3.1. Introduction to kinematic wheel-rail interaction analysis

3.1.1. Background information

A perfectly round wheel traversing a perfectly straight rail experiences single-point contact in time. When the wheel encounters a geometrical imperfection, the contact may shift from single-point-contact to double- or multi-point contact. The kinematic relation between the wheel and rail geometry describes the trajectory of the wheel's center of mass, see section 2.3. In a kinematic approach, the bodies are considered to be fully rigid, neglecting the elasticity in the materials. This leads to a singularity in the time derivative of the velocity. This time derivative is crucial in calculating the impact forces, which means that in a kinematic analysis, the vertical impulse force remains undefined, see section 2.4.

3.1.2. Objectives

The primary objective of this analysis is to address the gap in theoretical knowledge regarding the dynamic performance of insulated rail joints with a non-orthogonal junction angle. To achieve this, a kinematic wheel-rail interaction model is adopted that describes the wheel-rail interaction of a wheel traversing an IRJ with a non-orthogonal junction angle. By examining the effect of the junction angle θ on the dynamic behavior of the IRJ, this study takes the first steps toward establishing a more comprehensive theoretical foundation. Although the adopted model is simplified, it provides an initial understanding of the dynamic performance and can serve as a basis for future research.

3.1.3. Model scope and limitations

The model consists of a wheel, initially modeled as a perfectly shaped infinitely thin disc, and a simplified geometry of a non-orthogonal insulated rail joint. The kinematic interaction between wheel and joint geometry, as the wheel traverses the joint, results in a description of the trajectory of the wheel's center of mass in time. The primary focus is on the trajectory of the wheel's center of mass during the two-point contact phase, as the magnitude of the impact is defined at this specific wheel position.

In this analysis, the geometries of both the wheel and rail are simplified. Initially, the wheel is modeled as an infinitely thin disc, and later as a cylinder with finite thickness. The rail is assumed to be perfectly flat, and the wheel-rail contact assumed as a single point. In reality, the wheel has a conical and more complex shape, while the rail is curved, and the actual contact patch between the two has an elliptical area. Additionally, material properties are not considered, meaning all the bodies are considered fully rigid and geometrically perfectly shaped. Inertia is also neglected. This assumption results in the wheel being 'glued' to the track and not able to break contact because of its velocity.

The analysis conducted in this section has merely theoretical value and is solely used to demonstrate and gain insights in the dynamic behaviour at the wheel-rail contact at insulated rail joints with a non-orthogonal junction angle.

In the horizontal plane (x-y plane), the joint's geometry consists of two rail segments both with length $\frac{L}{2}$ and width 2 b_t (now half the track has thickness b_t) that are separated by a cut that is placed under a junction angle θ to the longitudinal direction of the track. The end plate material is neglected since the stiffness of the material can be neglected in comparison to the stiffness of the rail. In the vertical plane (x-z plane) the model has an assumed linear elastic deformation, defined by the dip angle β as seen in earlier literature. The wheel geometry consists initially of a perfectly round infinitely thin disc with radius R.

3.2. Global wheel-rail interaction: framework and example

3.2.1. Framework

To structure and guide the analysis, a framework has been developed to organize the steps involved. This framework is illustrated in Figure 3.1 and in this subsection, the process is briefly explained. The

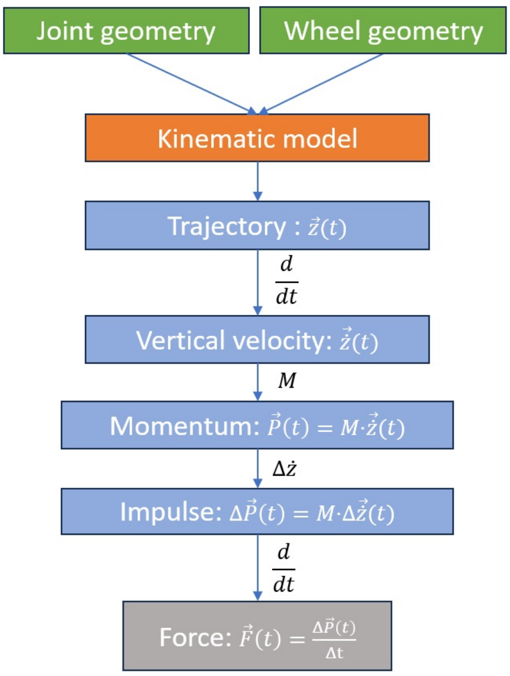


Figure 3.1: Framework for global wheel-rail interaction of an insulated rail joint with non-orthogonal junction angle. The force is not defined due to the rigid body assumption.

joint and wheel geometry both serve as inputs for the kinematic wheel-rail interaction model. The output of the model is the trajectory of the wheel's center of mass as a function of time. By taking the time derivative of the trajectory, the vertical velocity of the wheel's center of mass can be determined. Multiplying this velocity by the mass of the wheel results in the momentum of the wheel. Given that the vertical velocity is a function of time, the change in vertical velocity during double-point contact can be calculated. The difference in vertical velocity before and after double-point contact is now used to determine the magnitude of the impulse during impact. Typically, the time derivative can be taken one more time to determine the impact force. However, assuming rigid bodies leads to a discontinuity in the vertical velocity in time, which coincides with an undefined time derivative of this vertical velocity. Consequently, the force during this interaction is undefined. Since this force remains undefined, the magnitude of the impulse is used as a measurement of the magnitude of impact.

3.2.2. Global wheel-rail interaction example case

Prior to applying the framework to an insulated rail joint with a non-orthogonal junction angle, it will be used to analyze a simpler case: a step in vertical track geometry. This preliminary analysis is intended to present the use of the roadmap on a well known problem.

Joint and Wheel Geometry

In Figure 3.2 a wheel with radius R and center of mass M is shown traversing a step in vertical rail geometry with step height u_0. In reality, the radius of a train wheel typically far exceeds the magnitude of the step in vertical geometry. This characteristic allows for expressions in this analysis to be simplified, as the wheel radius is significantly larger than the size of the imperfection ($R >> u_0$). As the wheel moves along the track, it encounters the scenario illustrated in Figure 3.2. At position A the wheel is in contact with both the vertical step in track geometry and the original surface it was in contact with, resulting in transient double point contact. This causes an instantaneous change in the direction of the wheel's center of mass and shifts the center of rotation from the wheel's geometrical center to the point of contact with the vertical geometry. The wheel then rotates around this new center of rotation until it reaches position B, where its trajectory becomes parallel to the track again.

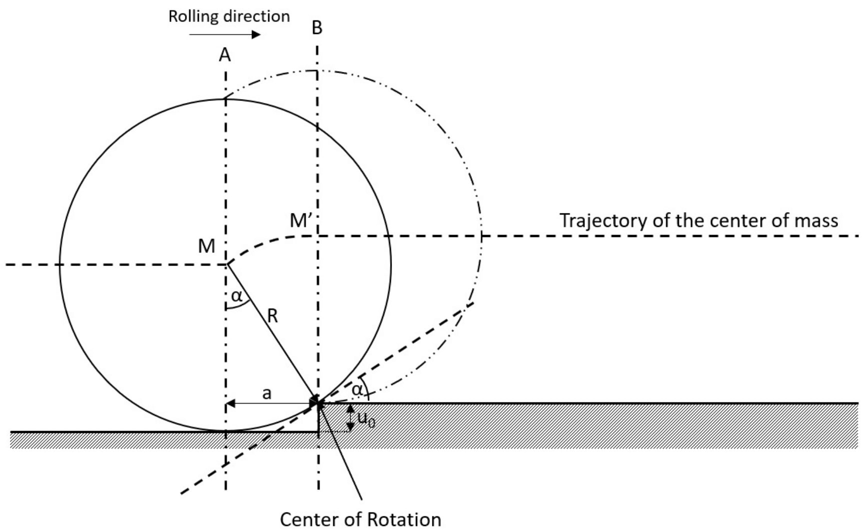


Figure 3.2: Schematic of wheel experiencing a step in vertical geometry

The horizontal distance a between the new center of rotation and the vertical center line of the wheel can be determined using the following trigonometric relations:

$$\cos(\alpha) = \frac{R - u_0}{R}$$

$$\sin(\alpha) = \frac{a}{R} \tag{3.1}$$

$$\cos(\alpha) = \sqrt{1 - \sin^2\alpha} \tag{3.2}$$

Plugging in Equation 3.1 in Equation 3.2 results in an expression for horizontal distance a:

$$a = \pm\sqrt{2Ru_0 - {u_0}^2}, \quad for \quad R >> u_0: \quad a = \pm\sqrt{2Ru_0}$$

Using Equation 3.1 the final result for the vertical component at transient double-point contact at A can then be written as:

$$\sin\alpha = \sqrt{\frac{2u_0}{R}} \tag{3.3}$$

Trajectory

In Figure 3.3 the trajectory for the center of mass M as function of x is shown in a coordinate system. The coordinate system's origin is defined to coincide with the point of instantaneous change in direction to simplify expressions. The point x_0 depicts the moment of two point contact and x_1 depicts the point where the trajectory is merely horizontal again. Between these 2 points the center of mass M follows a circular path around it's new center of rotation as seen in Figure 3.3. α depicts the initial angle of the trajectory right after two point contact. The center of mass M follows this circular path until the direction of the trajectory is merely horizontal again. At this point it will continue it's path parallel to the surface. For the coordinate system defined in Figure 3.3 the location of the new center of rotation can be described using parameters from the geometry. The trajectories illustrated in this analysis are purely theoretical, and their shape is greatly exaggerated. In reality, due to the significant difference in scale between the wheel radius and the step in vertical geometry, the trajectory's shape is far less distinct.

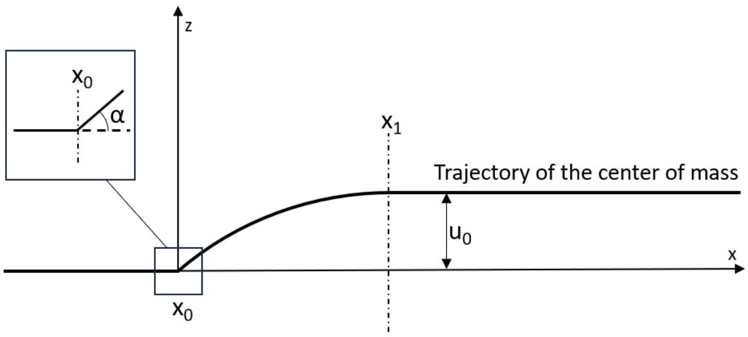


Figure 3.3: Trajectory of wheel's center of mass M as a function of position x

The trajectory of the center of mass M as function of x can be described using the following expressions:

$$z(x) = 0 \quad for \quad x < 0$$

$$z(x) = -R + u_0 + \sqrt{2x\sqrt{u_0(2R-u_0)} + R^2 - 2Ru_0 + u_0^2 - x^2} \quad for \quad 0 \leq x \leq x_1$$

This equation can be slightly simplified using $R >> u_0$:

$$z(x) = -R + u_0 + \sqrt{2x\sqrt{2Ru_0} + R^2 - 2Ru_0 - x^2} \quad for \quad 0 \leq x \leq x_1$$

$$z(x) = u_0 \quad for \quad x > x_1$$

The position of the wheel can be described in the time domain using a linear operator, the train's velocity v. Due to the assumption that the interaction occurs in a zero time interval, the velocity remains constant during the whole interaction between wheel and track:

$$x = vt \tag{3.4}$$

Using Equation 3.4 the trajectory of z can be expressed as a function of time:

$$z(t) = f(x(t))$$

This results in the following expressions for the trajectory in the time domain:

$$z(t) = 0 \quad for \quad t < 0$$

$$z(t) = -R + u_0 + \sqrt{2\sqrt{2Ru_0} \cdot vt + R^2 - 2Ru_0 - v^2t^2} \quad for \quad 0 \leq t \leq t_1$$
$$z(t) = u_0 \quad for \quad t > t_1$$

Due to the linear scaling between v and x, the graph in the time domain appears identical, though with an x-axis scaled with factor v, as seen in Figure 3.4.

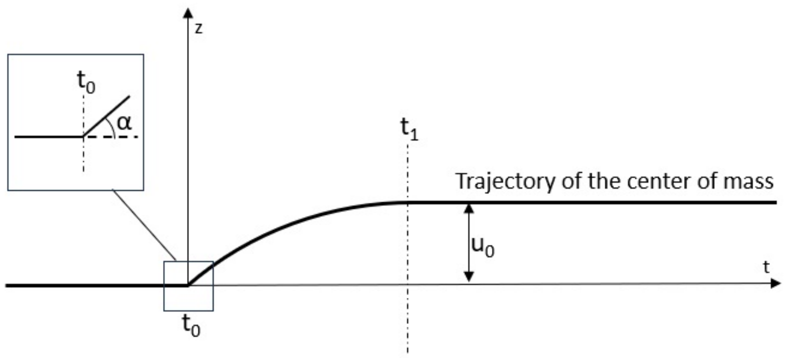


Figure 3.4: Trajectory of the wheel's center of mass M as function of time t

Velocity

Having represented the trajectory in the time domain, the vertical velocity can be determined by taking the time derivative directly, or applying the chain rule to the expressions as function of x to determine the derivative of z with respect to time:

$$\dot{z}(t) = \frac{dz}{dx} \cdot \frac{dx}{dt}$$

This results in the following expressions for the vertical velocity $\dot{z}$ of the center of mass M as a function of time:

$$\dot{z}(t) = 0 \quad for \quad t < 0 \tag{3.5}$$
$$\dot{z}(t) = \frac{(2\sqrt{2Ru_0} - 2vt)v}{2\sqrt{2\sqrt{2Ru_0}vt + R^2 - 2Ru_0 - v^2t^2}} \quad for \quad t_0 < t < t_1$$
$$\dot{z}(t) = 0 \quad for \quad t > t_1$$

The parameter $\xi = \frac{u_0}{R}$ determines the degree of circularity for the trajectory. A smaller ξ corresponds to a trajectory of the wheel's center of mass that more resembles a straight line, while a larger ξ results in a trajectory with a more pronounced circular path and a larger initial angle α. The vertical velocity of the center of mass is non linear for a circular trajectory, but when ξ becomes small enough the vertical velocity of the center of mass along the circular path will seem linear. In Figure 3.5 and Figure 3.6 the trajectories and vertical velocities of the center of mass are shown for various values of ξ. It should be noted that the vertical velocity is defined for every value of t, but is not continuous (jump in vertical velocity at $t = 0$). This discontinuity corresponds to an undefined vertical acceleration (second time derivative of z). These values, though theoretical and greatly exaggerated, are a good illustration for the changes in trajectory and vertical velocity. In practice, the wheel's radius R is typically way larger than u_0. Since dynamic effects are not considered here the train velocity $v = 1$ was used to have the same scaling on the x-axis for the position and the time domain. The chosen values for ξ were chosen to generate easily readable figures. When the center of mass M reaches t_0, and two point contact occurs, the direction of the center of mass changes instantaneous. The vertical velocity of the center of mass at this exact moment $(t = 0)$ can be described as:

$$\dot{z}(t = 0) = v \cdot \frac{\sqrt{2Ru_0}}{\sqrt{R^2 - 2Ru_0}} = v \cdot \frac{\sqrt{2Ru_0}}{\sqrt{R(R - 2u_0)}}$$

For $R >> u_0$:

$$\dot{z}(t = 0) = v \cdot \sqrt{\frac{2u_0}{R}} \tag{3.6}$$

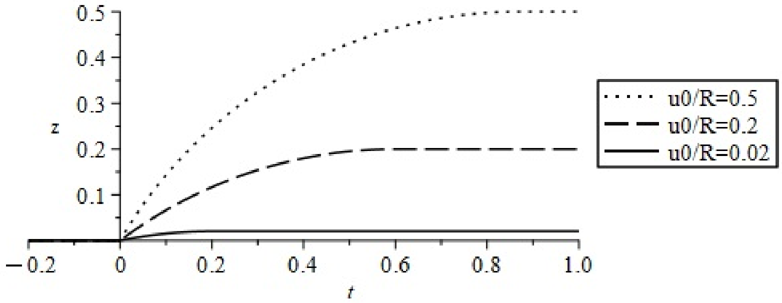


Figure 3.5: Trajectories for various values of ξ

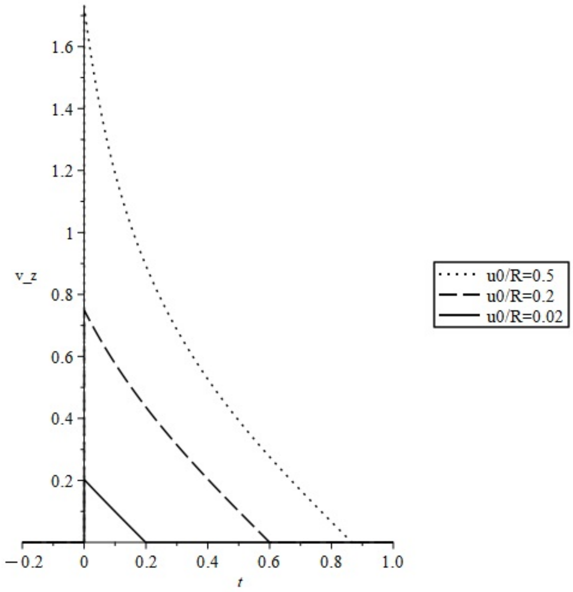


Figure 3.6: Vertical velocities for various values of ξ with discontinuity at $t = 0$

.

As expected, the result from the vertical component solved from the geometry and trigonometric relations (Equation 3.3), when multiplied by v, matches the result obtained using the trajectory of wheel and track (Equation 3.6). While solving the vertical component through geometric relations may be simpler for this case, using trajectory analysis offers a more versatile solution, applicable across a broader spectrum of scenarios. For example, this approach remains viable when measurements along a track are carried out and only the trajectory of the center of mass is known.

Momentum, impulse and force

The vertical momentum of the center of mass is dependent on the vertical velocity of the center of mass and the mass of the wheel. The wheel's momentum while traversing the track can be expressed as:

$$\vec{p}(t) = M \cdot \vec{v}(t)$$

Since the vertical velocities along the path of the center of mass are known, the vertical momentum along the path can determined (the mass doesn't change). From Equation 3.5 and Equation 3.6 can be observed that an instant change in vertical velocity occurs at $t = t_0$. This instant change in vertical velocity, or an instantaneous rotation of the velocity vector, corresponds to a change in vertical momentum, or a vertical impulse Δp_z. For the two-point contact at $t = t_0$ the vertical impulse can be expressed as:

$$\Delta\vec{p}_z = \vec{p}_{z,t_0'} - \vec{p}_{z,t_0} = M \cdot v \cdot \sqrt{\frac{2u_0}{R}}$$

The abrupt change in direction over an infinitesimal time interval produces a velocity that, while defined for every time instant t, is discontinuous, as illustrated in Figure 3.6. This discontinuity in vertical velocity results in an undefined time derivative of the velocity $\frac{dv}{dt}$. Consequently, the force remains undefined, as the force is defined as:

$$\overline{F} = \frac{d\vec{p}}{dt}$$

To address this, the magnitude of the impulse is utilized to quantify the impact.

In Figure 3.7, graphs depicting the trajectory, vertical velocity, momentum, and impulse are presented over time. The horizontal axes are aligned to provide a clear overview of how all concepts relate to each

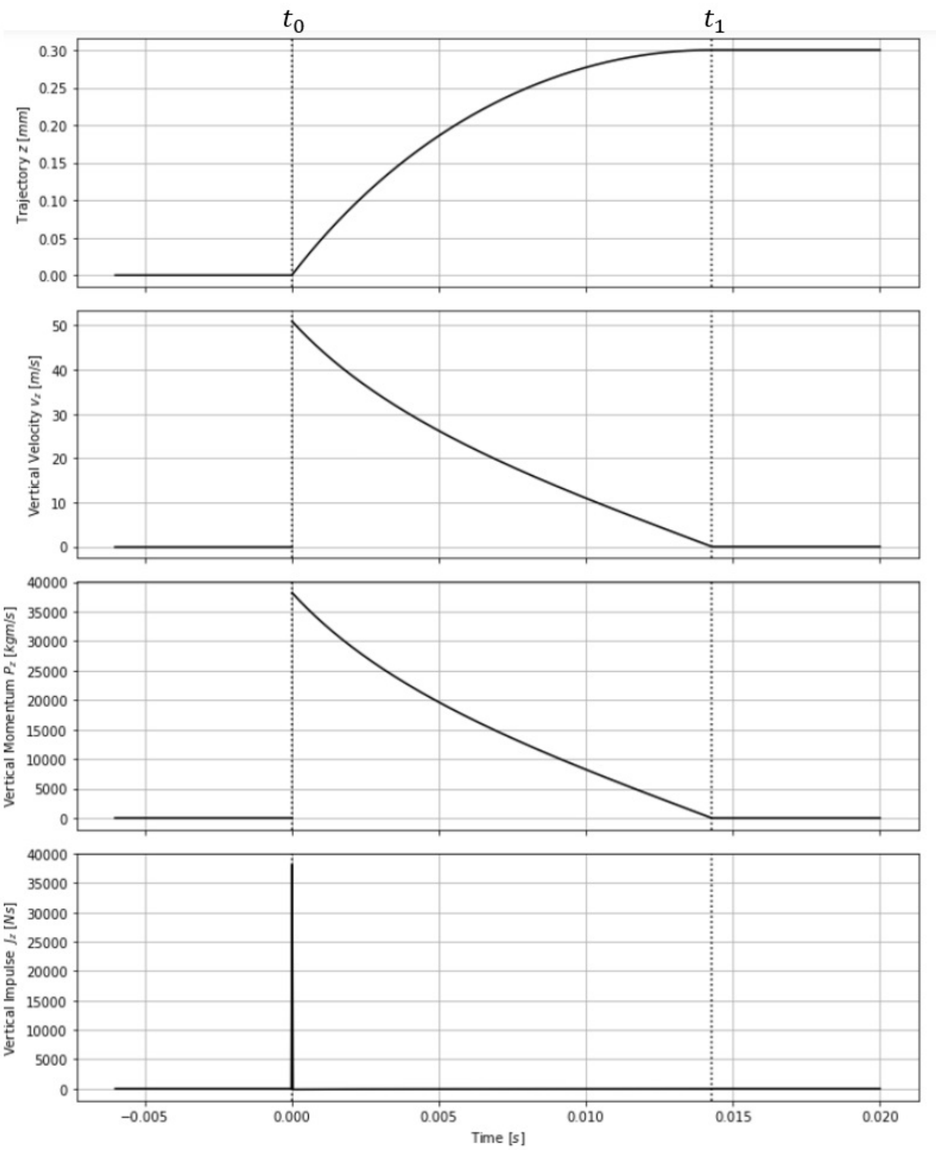


Figure 3.7: Time-Aligned Plots of Trajectory, Vertical Velocity, Vertical momentum and Vertical Impulse of the center of mass of a wheel traversing a step in vertical geometry with $u_0 = 0.3$ m, $R = 1$ m, $M = 750$ kg, $v = 50$ m/s

other. The chosen time interval shows the trajectory of the center of mass of the wheel, starting just before the two-point contact, marked as t_0 and ending when the wheel returns to a merely horizontal position, marked as t_1. The dimensions of the step height u_0 and wheel radius R are selected to create easily readable figures and hold merely theoretical value. In reality, the wheel radius far exceeds the step height. At $t = t_0$, two-point contact occurs, causing an instantaneous change in the direction of the wheel's center of mass. This directional shift in a zero time interval results in an instantaneous change and a discontinuity in the vertical velocity. Given that vertical momentum is the product of vertical velocity and a constant factor (the mass of the wheel's center), the plot of vertical momentum mirrors this behavior, showing a similar discontinuity. The vertical impulse, defined as the change in vertical momentum, shows as a sharp peak at t_0.

The illustrated framework is used to derive the dynamic properties from a wheel and rail geometry for a basic example. In the following sections this framework will be used to analyze a rail joint with a non-orthogonal junction angle.

3.3. Insulated rail joint with a non-orthogonal junction angle

In this section the framework introduced in subsection 3.2.1 is applied to analyze a non-orthogonal rail joint. After introducing the adopted geometries of the wheel and the joint, using the framework, the expressions for the dynamic properties of the joint are derived.

3.3.1. Adopted geometry

To describe the wheel-rail interaction of the wheel as it traverses a non-orthogonal insulated rail joint the dipped joint geometry defined below is adopted. The deflection of the joint is assumed to be linear.

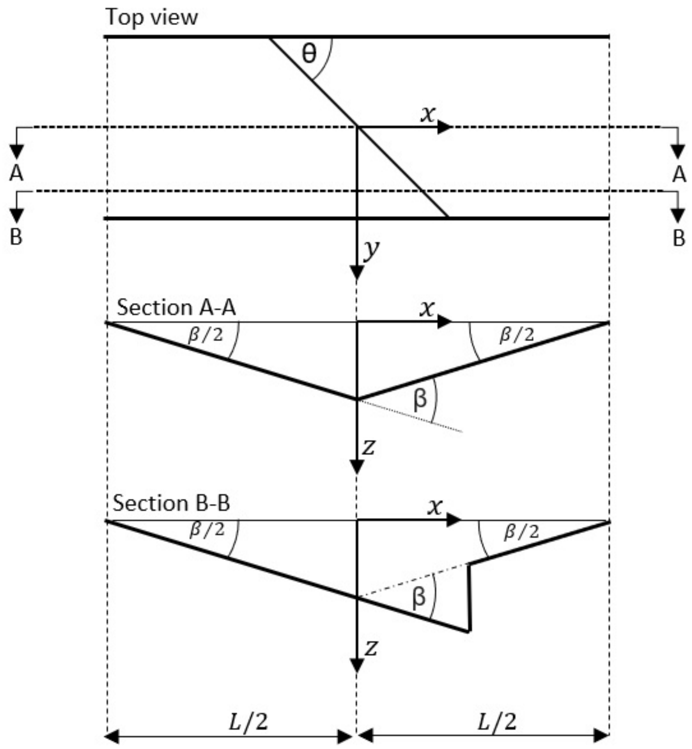


Figure 3.8: Top view and 2 cross-sections of the adopted geometry for the insulated rail joint with a non-orthogonal junction angle.

The diagonal line in the top view depicts the separation between the two rail ends. The global coordinate system is established with its origin at the midpoint of the track for a joint with zero deflection. The angle θ represents the junction angle of the cutoff, while the angle β defines the dip angle, as observed in earlier literature. As a result of the non-orthogonal junction angle, the deflection of the rail's end is dependent on both the x and y coordinates. This results in the following relations between the x and y coordinates for the diagonal cutoff:

$$x_{joint}(y, \theta) = y \cdot \cot \theta \tag{3.7}$$

$$y_{joint}(x, \theta) = x \cdot \tan\theta \tag{3.8}$$

Due to the geometry of the joint, the vertical deflection of the rail ends varies along the cross-section perpendicular to the rail. At the track's centerline $(y = 0)$, the rail head and tail have the same deflection. For every other y-value, the deflection varies between both rail ends, resulting in a vertical mismatch. Due to the placement of the x-y coordinate system in the middle of the track, the deflection of the left rail end at a certain y-coordinate lines up with the deflection of the right rail end for the same negative y-coordinate. Using this characteristic the height of the mismatch Δh between the rail ends can be determined using the following equations:

$$\begin{aligned} w_z(y) &= \sin\left(\frac{\beta}{2}\right) \cdot y \cdot \cot\theta \\ \Delta h(y) &= w_z(y) - w_z(-y) \end{aligned} \tag{3.9}$$

3.3.2. Wheel-rail contact scenarios

The wheel is initially modeled as an infinitely thin disc. The position of the center of this disc in the coordinate system is denoted using x_w, y_w and z_w. The width of the track is denoted as b_t. Due to the diagonal cutoff of the joint, the height difference between the rail ends increases as the position on the track becomes more eccentric. In other words, as y increases, the mismatch height Δh increases. As the disc approaches the rail joint, it is initially rolling down the rail. The locations of the contact points during 2-point contact for a regular dipped joint depend on the radius of the disc R and the dip-angle of the joint β. When the disc is located close enough to the center of the track during 2-point contact, the mismatch of the rail ends will be too small for the disc to encounter. This is when contact scenario 1 occurs. When the disc is positioned more eccentric (y_w increases) the mismatch between rail ends will become large enough for the disc to encounter and 2-point contact will occur in such a way that one of the contact points makes contact the diagonal cutoff. This is when contact scenario 2 will occur. The y-value for which the scenario that occurs changes from scenario 1 to scenario 2 is defined as the transition value and denoted as y_t. This means that depending on the disc's position along the y-axis, 2 ranges for y_w can be distinguished corresponding to the two wheel-rail contact scenarios:

- Contact scenario 1: $0 < y_w < y_t$
- Contact scenario 2 : $y_t < y_w < b_t$

In Figure 3.9 the bandwidths for each scenario are shown on the rail. Due to symmetry in the geometry of the track and the assumption that dynamic effects are not considered (inertia), the contact scenarios will occur symmetrically on both sides of the rail's center line. The maximum value for y_w is limited by the track width b_t. In Figure 3.10 both contact scenarios are depicted in the x-z plane. The trajectory of

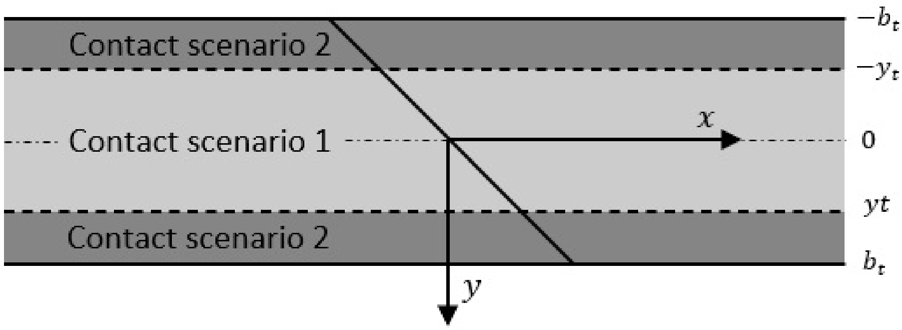


Figure 3.9: Overview of possible values of y_w and the position of the contact scenarios. Note that y_w should always be smaller than the track width b_t

contact scenario 1 follows the dipped joint geometry. Contact scenario 2 initially has the same trajectory as contact scenario 1 but after 2-point contact, the trajectory follows a circular path until the trajectory is parallel to the rail again. Note the horizontal and vertical shift in the position of contact point A for contact scenario 2.

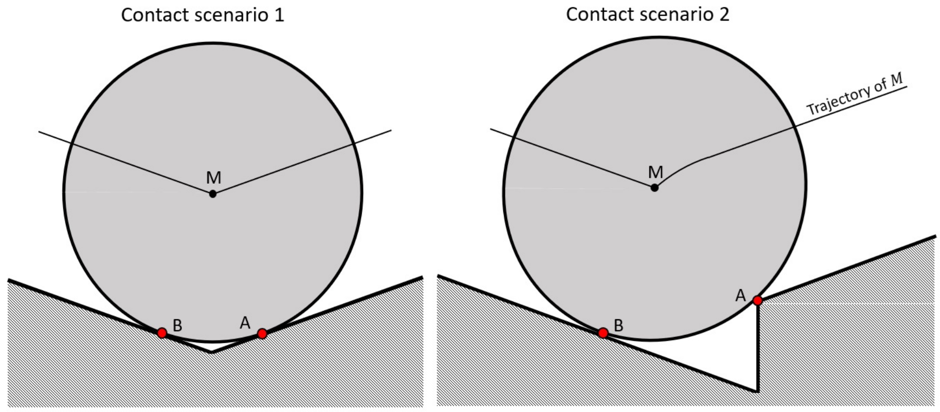


Figure 3.10: Both contact scenarios and their trajectories. Contact scenario 1 follows the shape of the dipped joint geometry, while scenario 2 has a circular trajectory after 2-point-contact

To gain a better understanding of what happens in the x-z plane as the eccentricity of the disc increases Figure 3.11 shows the position of the wheel during 2-point contact for four key values of y_w. Note that y_t denotes the transition value between the two wheel-rail contact scenarios and that x_A and x_B denote the horizontal distance between the contact points A and B to the vertical center line of the wheel. x_{joint} denotes the horizontal distance from the middle of the joint to the diagonal cutoff and Δh denotes the height of the vertical mismatch between the rail ends. The following things should be observed in each situation:

1. The disc is located in the middle of the track $(y_w = 0)$. The height of the vertical mismatch Δh is equal to 0. Contact scenario 1 occurs.
2. The disc is located between the middle of the track and the transition value ($0 < y_w < y_t$). The height of the vertical mismatch Δh is larger than 0 but not large enough for the disc to encounter. Contact scenario 1 occurs.
3. The disc is located at the transition value $(y_w = y_t)$. At this point the vertical mismatch Δh is equal to the vertical position of contact point A. This is the maximum y_w value for contact scenario 1 occurs.
4. The disc is located eccentric on the track beyond the transition value $(y_w > y_t)$ (Note that y_w always has to be smaller than b_t). The vertical mismatch Δh is now large enough for the wheel to encounter. Contact scenario 2 occurs and the center of rotation has shifted to contact point A.

The horizontal distance between the vertical centerline of the wheel and both contact points, denoted as x_A and x_B, remains constant in all cases where contact scenario 1 occurs, and can be directly determined from the wheel and track geometry:

$$x_A = \sin\left(\frac{\beta}{2}\right) \cdot R \tag{3.10}$$

The expression for the transition value y_t follows from the known horizontal distance between the disc's vertical centerline and contact point x_A. As seen in Figure 3.11 the height of the vertical mismatch Δh is 0 when the wheel is placed in the center of the track $(y_w = 0)$. When the eccentricity of the disc on the rail increases (the value of y_w increases), the height of the diagonal cut starts increasing. If the eccentricity of the disc keeps increasing the height of the vertical mismatch Δh will match the height of the contact points. The horizontal distance between the vertical center line of the wheel and the contact point in this position is still equal to Equation 3.10, which results in the following expression:

$$x_{joint} = x_A = \sin\left(\frac{\beta}{2}\right)R$$

Now, using the relationships between the x and y coordinates of the diagonal cut Equation 3.19, and the expression for the known horizontal distance between the disc's vertical center line and contact

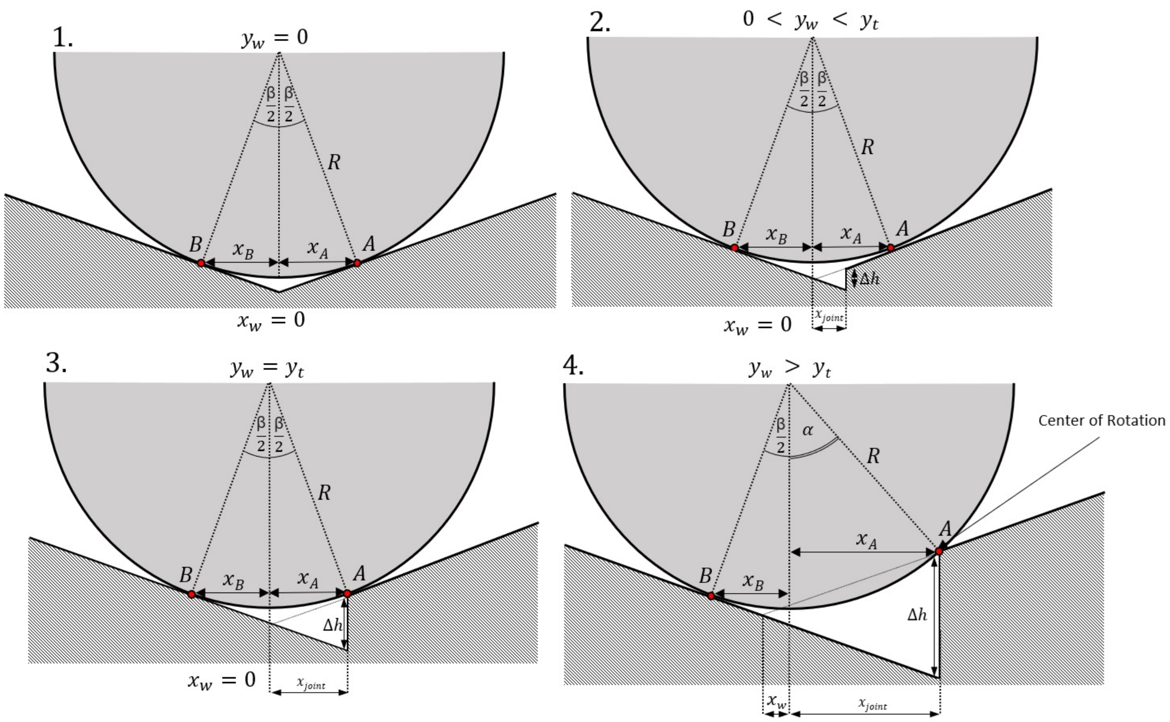


Figure 3.11: Schematic view of wheel-rail interaction in the x-z plane at 4 values of interest when y_w increases

point Equation 3.10 the transition value can be determined:

$$y_t = \sin\frac{\beta}{2}\tan(\theta)R$$

In contact scenario 2, this horizontal distance between the cut and the vertical center line x_A increases, making the derivation of an expression for x_A more complex. The expression for x_A in contact scenario 2 will be handled in the corresponding section subsection 3.3.4.

3.3.3. Contact scenario 1

Trajectory

In Figure 3.12 the trajectory of the disc's center of mass as a function of time is shown in a coordinate system. The origin of the coordinate system is chosen at the time of two-point contact to simplify expressions and the transformation to the time domain is done using Equation 3.4.

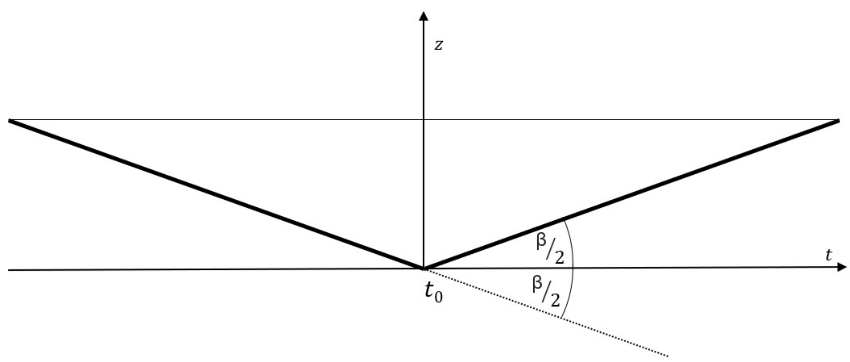


Figure 3.12: Trajectory of the center of mass for mechanism without definitively defined center of rotation

The trajectory in the time domain can be described using the following expressions:

$$z(t) = -\sin\left(\frac{\beta}{2}\right) \cdot v \cdot t \quad for \quad t < t_0$$

$$z(t) = \sin\left(\frac{\beta}{2}\right) \cdot v \cdot t \quad for \quad t \geq t_0$$

Assuming small deflections the expressions can be simplified using small angle theorem:

$$z(t) = -\frac{\beta}{2} \cdot v \cdot t \quad for \quad t < 0$$

$$z(t) = \frac{\beta}{2} \cdot v \cdot t \quad for \quad t \geq 0$$

Vertical velocity

Now that the trajectory in the time domain is known the derivative with respect to time can be taken to obtain the expressions for the vertical velocity along the trajectory:

$$\dot{z}(t) = -\frac{\beta}{2} \cdot v \quad for \quad t < 0 \tag{3.11}$$

$$\dot{z}(t) = \frac{\beta}{2} \cdot v \quad for \quad t \geq 0$$

These functions are not continuously defined and have a singularity at the moment of 2-point contact $(t = t_0)$. This singularity in the vertical velocity is a result of the assumption that the bodies in this analysis are considered fully rigid, see section 2.4. In Figure 3.13 the vertical velocity is plotted as a function of time.

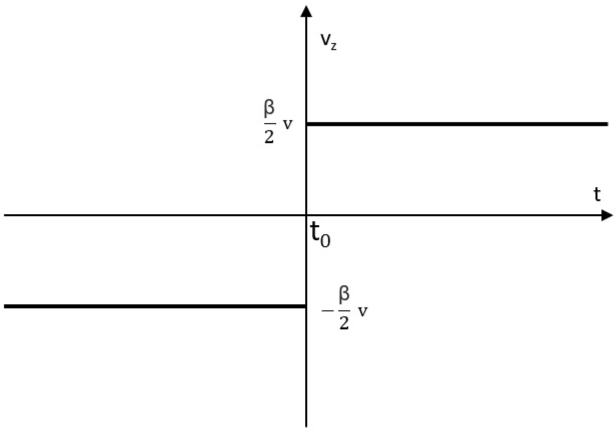


Figure 3.13: Vertical velocity before and after two-point contact for contact scenario 1. Instant change in velocity at $t = t_0$.

Using Equation 3.11 and Equation 3.3.3 the change in vertical velocity at the moment of two-point contact $(t = t_0)$ can be determined:

$$\Delta\dot{z} = \dot{z}^{+} - \dot{z}^{-} = \frac{\beta}{2}v - \left(-\frac{\beta}{2}v\right) = \beta \cdot v \tag{3.12}$$

Vertical Impulse

The instant change in vertical velocity results in an instant change in the momentum of the disc's center of mass. Since a change in momentum is defined as the impulse, Equation 3.12 can be used to define the impulse for contact scenario 1:

$$\Delta p_z = M \cdot \Delta\dot{z} = M \cdot \beta \cdot v \tag{3.13}$$

Given that, as a result of the rigid body assumption, the impact force remains undefined, the impulse will be used as a measurement for the magnitude of impact.

3.3.4. Contact scenario 2

When the eccentricity of the disc on the track exceeds the transition value $(y_w > y_t)$ the height of the vertical mismatch between the rail ends becomes large enough to make contact with the disc. This results in a shift of the wheel's center of rotation to the point of contact on the diagonal cutoff, as seen in Figure 3.10. This situation greatly resembles the example case, see subsection 3.2.2. The key distinctions in this contact scenario are that the initial angle of the trajectory is inclined, and the horizontal distance x_A from the vertical centerline of the wheel to the contact point A is initially unknown, as illustrated in Figure 3.11. For this contact scenario, the horizontal distance x_A is not directly defined from the geometry. To define the trajectory of the disc's center of mass during 2-point contact, knowing the magnitude of x_A is crucial. To find x_A a set of equations following from the rail and disc's geometry have to be solved.

Local coordinate system

To find the horizontal distance between the vertical centerline of the disc and contact point A a local coordinate system is introduced. This local coordinate system is located at the bottom of the wheel at $y_w = y_t$ and its axes are denoted as x', y' and z', as seen in Figure 3.14 and Figure 3.15. The following linear transformations are applied to transform from the global coordinate system to the local coordinate system:

$$x' = x$$

$$y' = y - m, \quad with \quad m = y_t = \sin\left(\frac{\beta}{2}\right) R \tan\theta$$

$$z' = z - n, \quad with \quad n = \frac{L}{2} \sin\left(\frac{\beta}{2}\right)$$

The expression for y_t has been derived in subsection 3.3.2. The expression for n is equal to the vertical distance travelled by the wheel until it makes 2-point contact (note that $y_w = y_t$, so the contact points are located as in contact scenario 1). The vertical distance travelled follows from the horizontal distance travelled by the wheel into the joint and the dip angle. The horizontal distance travelled by the wheel is equal to $\frac{L}{2}$ which leads the expression for n. The expression for y_t has been derived

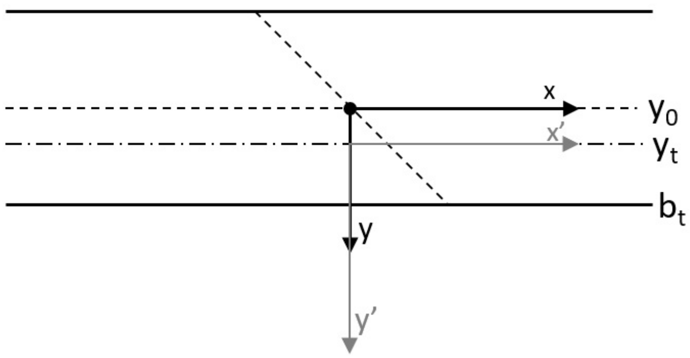


Figure 3.14: Transformation of local axis in x-y plane

in subsection 3.3.2. The expression for n is equal to the vertical distance travelled by the wheel until it makes 2-point contact (note that $y_w = y_t$, so the contact points are located as in contact scenario 1). The vertical distance travelled follows from the horizontal distance travelled by the wheel into the joint and the dip angle. The horizontal distance travelled by the wheel is equal to $\frac{L}{2}$ which leads the expression for n.

Geometry

It should be noted that all values for $y' > 0$ refer to contact scenario 2, since $y' = 0$ coincides with the transition value $y = y_t$ and that the horizontal distance between the vertical center line of the wheel and the contact point A are no longer directly defined from the geometry. To solve the equations for the trajectory, first the expression for x_A should be found, as it directly relates to angle α.

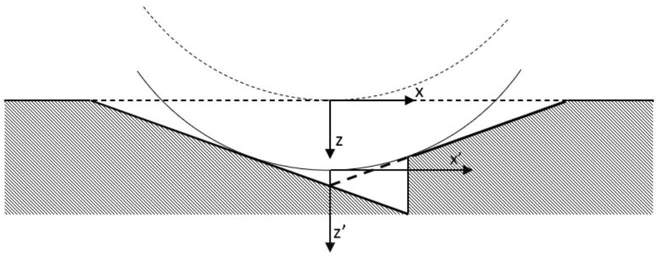


Figure 3.15: Transformation of local axis in x-z plane, situation depicted occurs when $y_w = y_t$

The horizontal distance x_A between the vertical centerline of the disc and the diagonal cutoff can be determined by defining the joint and disc geometry in the local coordinate system. Once these are defined, a set of equations can be solved to find x_A. The diagonal cutoff is modeled as a linear function in 3-dimensional space x', y', z' using the following equations: For the position of the diagonal cutoff in the x'-z' plane:

$$z'_{joint}(x') = -\sin(\frac{\beta}{2})x' + z_0 \tag{3.14}$$

To determine z_0, a point on the line that represents the diagonal cutoff is used where both the x and z-coordinate are defined. This point is defined by the geometry at the transition point between the two mechanism $(y = y_t)$ or $(y' = 0)$, as seen in Figure 3.11 :

$$x'_{joint}(y' = 0) = R\sin(\frac{\beta}{2})\ ;\ z'(y' = 0) = -R(1 - \cos(\frac{\beta}{2})) \tag{3.15}$$

Plugging these values into Equation 3.14 results in an expression for z_0:

$$z_0 = R(\sin(\frac{\beta}{2})^2 + \cos(\frac{\beta}{2}) - 1) \tag{3.16}$$

Now plugging Equation 3.16 into Equation 3.14 leads to the following expression for the position of the diagonal cut in the x'-z' plane:

$$z'_{joint}(x') = -\sin(\frac{\beta}{2})x' + R(\sin(\frac{\beta}{2})^2 + \cos(\frac{\beta}{2}) - 1) \tag{3.17}$$

The relationship between x' and y' for the diagonal cutoff can be determined from the geometry of the track as seen in Figure 3.8. At the transition value of the contact scenarios $(y' = 0)$, the value for x'_{joint} is equal to horizontal distance between the vertical centerline of the wheel and the contact point $(x'_{joint} = x_A)$:

$$x'_{joint} = R\sin(\frac{\beta}{2})\ ,\ y' = 0 \tag{3.18}$$

This results in the following relation between x'_{joint} and y':

$$x'_{joint}(y') = y'\cot\theta + R\sin(\frac{\beta}{2}) \tag{3.19}$$

Substituting Equation 3.19 into Equation 3.17 results in the following expression for the the diagonal cut z_{joint} as a function of y':

$$z'_{joint}(y') = -(y'\cot\theta + \sin\frac{\beta}{2})\sin\frac{\beta}{2} + R(\sin\frac{\beta}{2}^2 + \cos\frac{\beta}{2} - 1) \tag{3.20}$$

For every y'-value the diagonal cutoff has a point corresponding to this y'-value in the x'-z' plane located at $(x'_P,\ z'_P)$. In Figure 3.11 this point is defined as point A. Now that the functions $x'_{joint}(y')$, and $z'_{joint}(y')$ for the diagonal cutoff are known, x_P and z_P can be determined for any $(0 < y' < b_t)$ by filling in Equation 3.19 and Equation 3.17 defining the point of contact on the diagonal cutoff in the x'-z' plane.

Due to the assumption that the wheel is attached to the bogey using a rigid connection the wheel will strictly rotate around its rotational axis. This means the wheel will always remain perfectly straight in the x'-z' plane. The expression for a circle is used to describe the wheel in this plane:

$$(x' - a)^2 + (z' - b)^2 = R^2 \tag{3.21}$$

Parameters a and b are respectively the horizontal and vertical shift of the center point of the wheel and R is the wheel radius. Before making contact with the diagonal cutoff, the wheel will descend along the rail-head at a known angle, namely half the dip angle: $\frac{\beta}{2}$. This means the vertical shift b is directly related to the horizontal shift a and an unknown can be eliminated from Equation 3.21 using the following expression:

$$b = a \cdot \sin\left(\frac{\beta}{2}\right) - R \tag{3.22}$$

Now the horizontal shift a of the wheel can be solved as a function of y' by substituting Equation 3.19, Equation 3.17 and Equation 3.22 into Equation 3.21 and solving for a.Due to the chosen location of the local axis, the expression for a becomes large. A more elegant way would be to define a moving coordinate system along the diagonal cutoff and then proceed as explained above. This will likely result in a more compact expression for a. The maple script to derive the parameters is added to the appendices Appendix B. The horizontal distance x_A between vertical centerline of the wheel and contact point A as a function of y' can now be determined as follows:

$$x_A(y') = x'_{joint}(y') - a(y') \tag{3.23}$$

The horizontal distance from the wheel's vertical centerline to it's point of contact x_A remains constant for all positions of the wheel corresponding to contact scenario 1. Once the wheel is eccentric enough for contact mechanism 2 to occur x_A will increase if the eccentricity of the wheel increases (increase in y_w). In contact scenario 1, x_A corresponds to the angle $\frac{\beta}{2}$. In contact scenario 2, x_A corresponds to the angle α. Since x_A is always larger or equal for contact scenario 2 compared to contact scenario 1, α is always larger or equal than $\frac{\beta}{2}$.

In Figure 3.16 the various parameters discussed in this section are depicted. A top view of half of the rail is shown with the global and local axes. For a certain y-value within the second contact scenario bandwidth, the disc is depicted in the corresponding position in the $x' - z'$ plane. Table 3.1 summarizes these parameters used in this section and gives a short description.

The vertical impulse during impact is determined by the trajectory right before and after 2-point contact occurs. By focusing on a small segment of the trajectory near the point of two-point contact, the circular path of contact scenario 2 can be approximated as linear. With this assumption, both scenarios show a comparable trajectory near the point of impact. The main difference is the angle of the trajectory immediately after impact. This angle is always larger in contact scenario 2 due to the greater horizontal distance from the vertical center line of the wheel to the contact point compared to contact scenario 1. In Figure 3.17 the linearized trajectories are shown with the geometries defining the angles during impact. With the horizontal distance x_A defined for any y'-value within the defined boundaries, the trajectory for contact scenario 2 during two-point contact is fully determined. The value of the outgoing angle (α) can be determined using the now known horizontal distance between the vertical center line of the disc and the diagonal cutoff and is defined as:

$$\sin \alpha = \frac{x_A}{R}$$

Trajectory

In Figure 3.18 the linearized trajectory of contact scenario 2 is shown in a coordinate system that coincides with the instantaneous change in direction during 2-point contact. Since the interaction between the wheel and the joint is assumed to happen in a zero time interval, the train's total velocity v during contact is considered to be constant. As a result, when transferring to the time domain, the horizontal axis of the trajectory is scaled by the constant scalar v. The trajectory of the center of mass M as function of time can be described using the following expressions:

$$z(t) = -\sin\frac{\beta}{2}\,vt \quad for \quad t < t_0$$

Table 3.1: Summary of parameters used in geometry of contact scenario 2

Symbol	Name	Description
a	Wheel shift	Horizontal shift of wheel in local coordinate system, follows from solving the equality between circle equation and position of diagonal cut at a certain y' value
A	Contact point A	Front contact point of the disc during 2-point contact
B	Contact point B	Rear contact point of the disc during 2-point contact
x_A	Horizontal distance contact point A	Denotes the horizontal distance between the wheel vertical center line and contact point A.
x_B	Horizontal distance contact point A	Denotes the horizontal distance between the wheel vertical center line and contact point B.
θ	Junction angle	Denotes the junction angle between the joint and longitudinal direction of the rail. A 90 degree cut corresponds to a 'squared' or 'traditional' IRJ
x'_{joint}	x'-position joint	Position of diagonal cutoff in local coordinate system
z'_{joint}	z'-position joint	Position of joint in z'-direction. The height of the diagonal-cut in local coordinate system

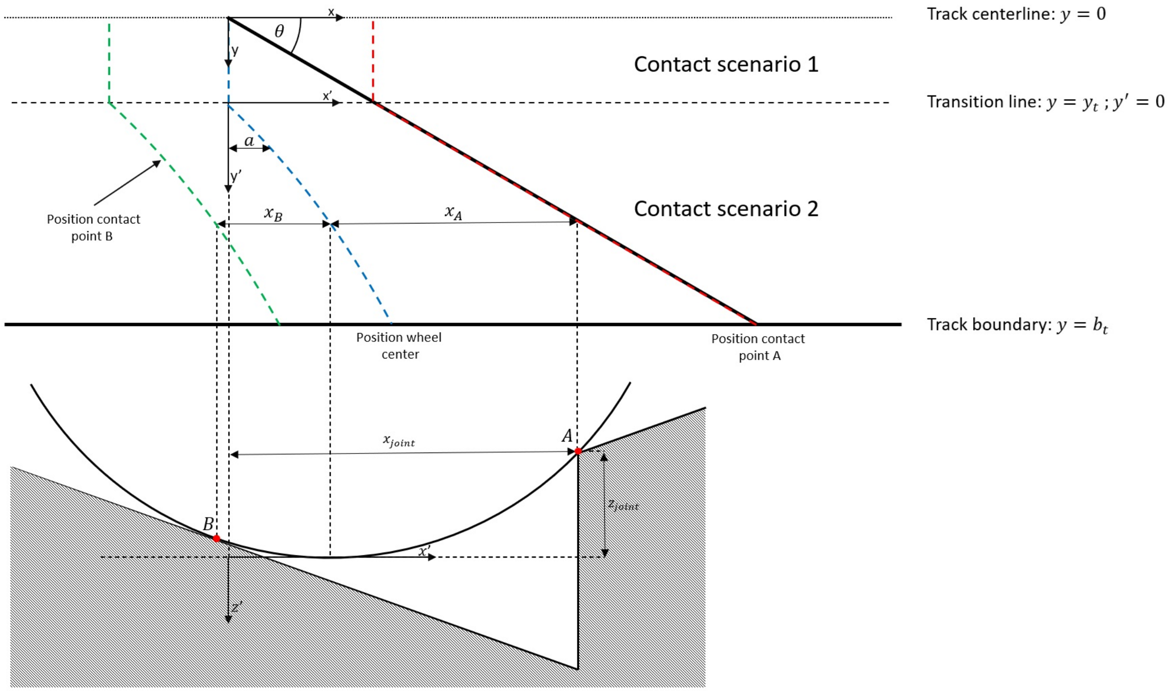


Figure 3.16: Half of the track with an overview of all parameters and the wheel depicted at a position in contact scenario 2.

$$z(t) = \sin\alpha\, vt \quad for \quad t \geq t_0$$

For small deflections, using the small angle theorem:

$$z(t) = -\frac{\beta}{2}\, vt \quad for \quad t < t_0$$

$$z(t) = \alpha\, vt \quad for \quad t \geq t_0$$

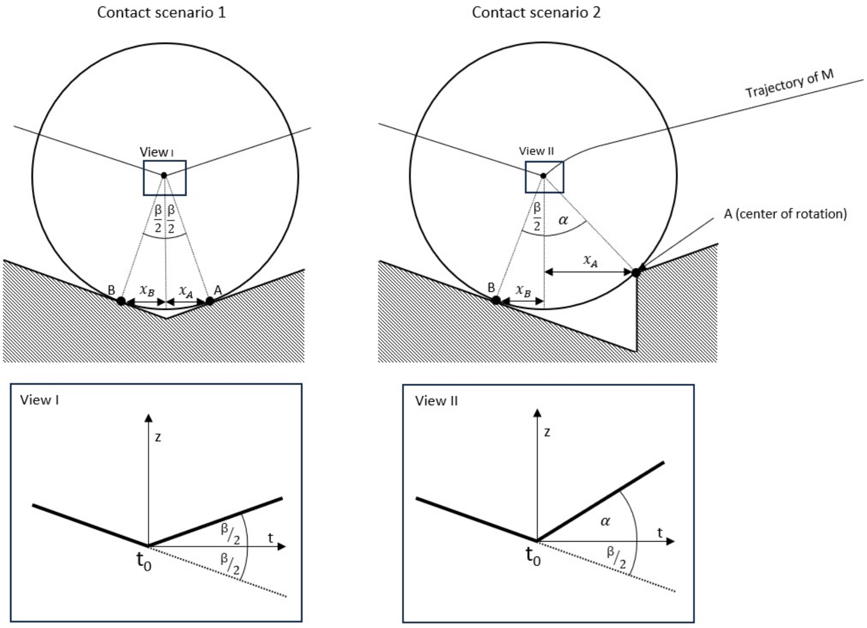


Figure 3.17: Geometries and trajectories of both contact scenarios during double-point contact. View II shows the linearized trajectory of contact scenario 2.

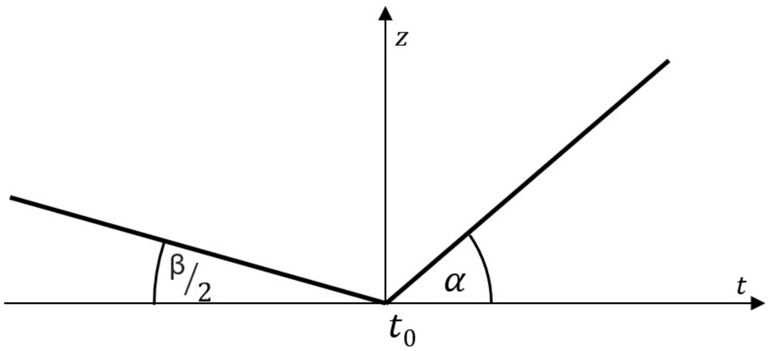


Figure 3.18: Linearized trajectory of the disc's center of mass M for contact scenario 2, immediately before and after 2-point contact

Note that these functions are only valid for values of t close to t_0, since the trajectory has been linearized.

Vertical velocity

Now that the trajectory is known as a function of time the vertical velocity can be determined by taking the time derivative:

$$\dot{z}(t) = -\sin\frac{\beta}{2}\,v \quad for \quad t < t_0 \tag{3.24}$$

$$\dot{z}(t) = \sin\alpha\,v \quad for \quad t = t_0 \tag{3.25}$$

For small deflections, using the small angle theorem:

$$\dot{z}(t) = -\frac{\beta}{2}v \quad for \quad t < t_0 \tag{3.26}$$

$$\dot{z}(t) = \alpha\,v \quad for \quad t = t_0 \tag{3.27}$$

In Figure 3.19 the vertical velocity of the center of mass of the wheel is plotted in time. Again, these functions are non-continuous due to the assumption that all bodies are fully rigid. Using Equation 3.26

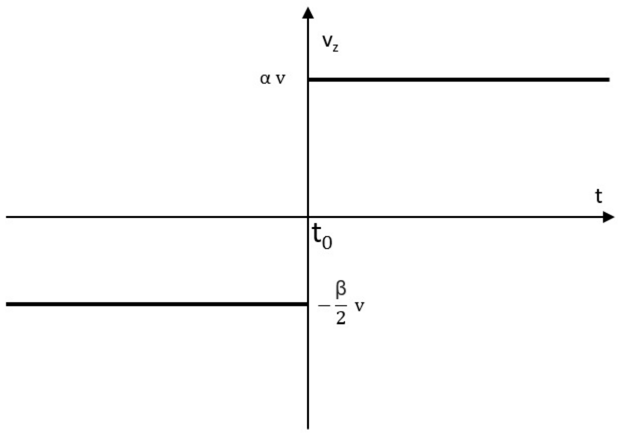


Figure 3.19: Vertical velocity before and after two-point contact for contact scenario 2. Instant change in velocity at $t = t_0$.

and Equation 3.27 the change in vertical velocity at the moment of two-point contact $(t = t_0)$ can be determined:

$$\Delta\dot{z} = \dot{z}^{+} - \dot{z}^{-} = \alpha \cdot v - (-\frac{\beta}{2} \cdot v) = (\alpha + \frac{\beta}{2}) \cdot v \tag{3.28}$$

Vertical impulse

The instant change in vertical velocity results in an instant change in the momentum of the disc's center of mass. Since a change in momentum is defined as the impulse, Equation 3.28 can be used to define the vertical impulse for contact scenario 2:

$$\Delta p_z = M\Delta\dot{z} = M \cdot (\alpha + \frac{\beta}{2}) \cdot v$$

Given that, as a result of the rigid body assumption, the impact force remains undefined, the impulse will be used as a measurement for the magnitude of impact.

3.3.5. Wheel modeled as a cylinder with finite thickness

In the previous sections, the wheel was modeled as a disc. This section presents a qualitative analysis of the wheel modeled as a cylinder. Similar to the disc model, it is assumed that no rotation is possible other than around the central axis. This assumption is justified by the fact that in reality the wheel is attached to a bogey that prevents rotation around any other axis. The cylinder is treated as a series of rigidly connected discs. Unlike the disc, where the thickness b_w approaches zero, the cylinder has a finite thickness b_w. The adopted geometry for the cylinder, represented as a series of connected discs, is shown in Figure 3.20.

The use of a cylinder instead of a disc changes the contact conditions from a contact point to a line contact, which still only holds theoretical value. In reality, the contact patch has en elliptical shape. Depending on the position on the rail, the length of this line contact may vary. For this qualitative analysis, a longer line contact is considered to provide better contact conditions, as it is assumed to lead to more favorable loading conditions for the elliptical contact patch. The actual behavior during contact is complex and would require a more detailed computational analysis. In Figure 3.21 half of the rail is shown with the cylinder positioned in 5 different possible configurations. The image also shows which contact scenario corresponds to each wheel position. Due to symmetry it is unnecessary to provide the other half of the rail.

In configurations 1 through 4, contact scenario 1 occurs because one of the discs in the cylinder remains below the transition value. Due to the assumed rigid connection between the discs, the other discs in the cylinder are constrained to follow the disc that first makes two-point contact. In configuration 5, contact scenario 2 takes place. In configurations 1 and 2, the full width of the line contact is utilized for transferring forces between the cylinder and the rail. In configuration 3, only half of the line contact

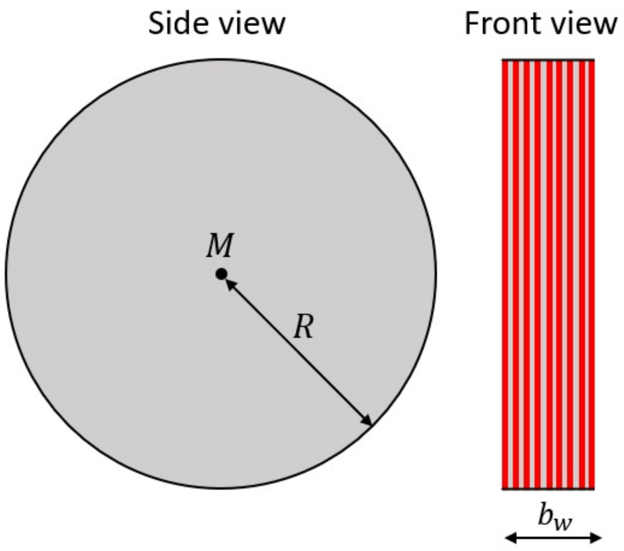


Figure 3.20: Adopted geometry for the cylinder. The discs are depicted in red.

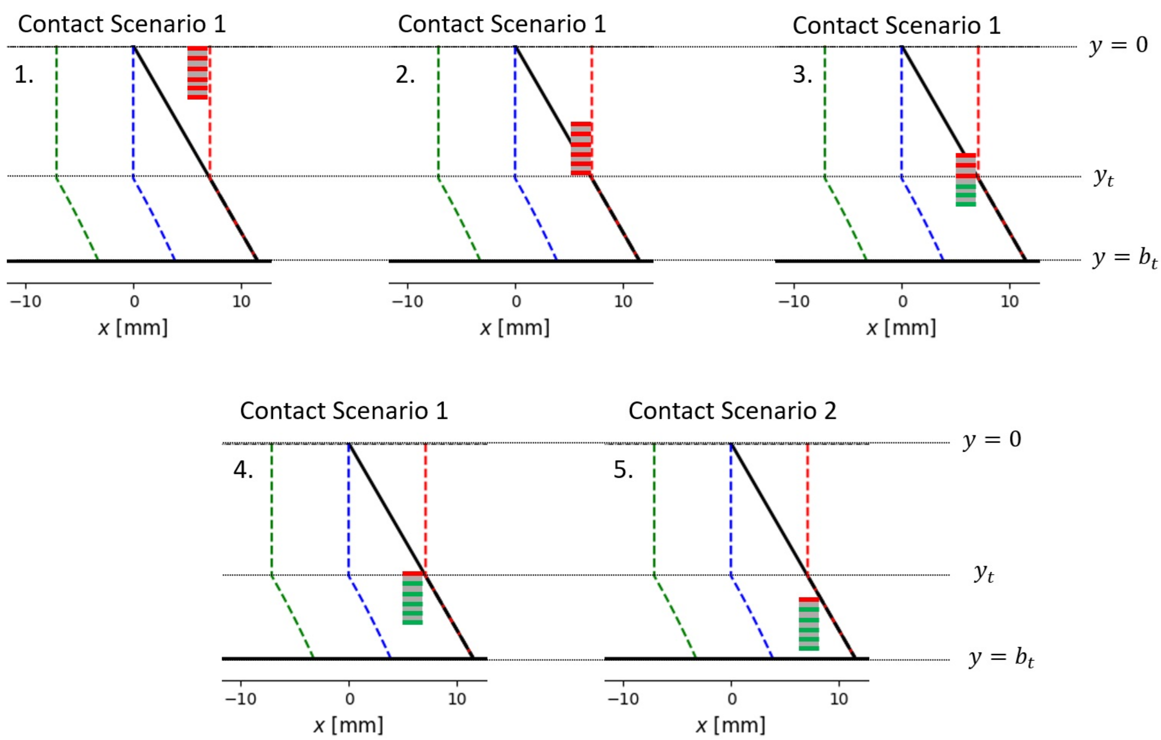


Figure 3.21: Possible contact configurations of a cylinder with finite thickness on a joint with a non-orthogonal junction angle.

width is available to transfer the contact forces. In configurations 4 and 5, the contact reduces to a point contact again, as only a single disc is in contact.

Although this is a qualitative analysis, some theoretical insights can be drawn. When the entire cylinder remains within the range of contact scenario 1, as in configurations 1 and 2, the full width of the contact can be utilized, while benefiting from the smaller impulse of contact scenario 1—making this the ideal configuration. In configurations 3 and 4, the line contact width gradually decreases until it eventually becomes a point contact, but it still operates under the favorable conditions of contact scenario 1. Configuration 5 demonstrates that any position of the cylinder in contact scenario 2 will result in a point contact. This, combined with the larger impulse associated with contact scenario 2, creates the worst contact condition.

4
Results Wheel-Rail Interaction

In chapter 3 , the kinematic behavior of the wheel rail interaction of an insulated rail joint with a junction angle is analyzed. The analysis reveals that two distinct contact scenarios may arise, depending on the wheel's lateral position on the rail. Each contact scenario has its own distinct trajectory for the wheel's center of mass, resulting in different dynamic properties. This section presents the findings from the wheel-rail interaction analysis with the main focus on the influence of the junction angle on the dynamic properties of the joint.

4.1. Wheel and rail parameters

While many dimensions in the railway industry are standardized, there are still numerous models and variations available. Additionally, factors such as wear, fatigue, and other operational conditions can affect the geometry dimensions over time. This results in each parameter having a certain range of variability. In this analysis, specific values were selected for some parameters, while for others, a range of values was considered. The table below Table 4.1 presents the parameters used in the wheel-rail interaction model and their selected values or selected value ranges. The chosen value for each parameter is then briefly explained.

Table 4.1: Parameters and corresponding values or value ranges used for wheel-rail interaction analysis

Symbol	Parameter	Value	Unit
R	Wheel radius	475	mm
M	Wheel's lumped mass	750	kg
v	Train velocity	[10 - 50]	m/s
θ	Junction angle of joint	[30, 45, 60, 90]	degrees
β	Dip angle	[0-0.05]	rad
b_t	Lateral wheel clearance	20	mm
y_w	Lateral position of wheel	[0-20]	mm

The radius and mass of a wheel change throughout its service life due to wear. Initially, the initial wheel's diameter is equal to 975 mm, but over time it decreases to about 875 mm. Naturally, as the diameter of the wheel reduces, the mass reduces as well. For this analysis the diameter is chosen to be 950 mm, which corresponds to a radius of 475 mm. The mass is chosen to be 750 kg which is a value well within the possible range.

The train's minimum velocity considered in this analysis is 10 m/s. Lower velocities will result in relatively low impulses and are not relevant for dynamic analysis. The train's maximum velocity is based on the maximum speed of intercity trains in the Netherlands, which is approximately 160 km/h.

The junction angle of the joint can theoretically vary range from 0 to 90 degrees. An angle of 90 degrees will result in a standard squared joint geometry. As the angle decreases, the joint between the

rail ends becomes more aligned with the longitudinal direction of the track. However, the angle cannot be zero degrees, as this would result in the joint becoming fully oriented in longitudinal axis, making the joint parallel to the rail and preventing it from crossing to the other side. As the angle decreases, the longitudinal length of the joint increases. In practice, an insulating end plate material is placed within the joint. If the longitudinal length of the joint becomes too large, it may become impractical to replace the end plate material in the event of a defect, which is likely to occur over time due to the high and frequent loads applied to the rail as a result of train axle's passing by. For this reason, three practical values for the junction angle θ are selected: 30°, 45°, and 60° to ensure feasibility and manageability. The 90 degree angle, or squared joint geometry, is considered for reference as for this orientation of the joint, only contact scenario 1 occurs.
The chosen range for the dip angle β is determined by using the basic deformation of metallurgical welding in rail and applying a factor for the bolted connection. The dip angle for metallurgical welding ranges practically from 2 to 5 mrad. Applying an assumed factor of 10 for the bolted connection results in the mentioned maximum value above of 50 mrad or 0.05 rad, which is also in line with values from earlier literature.

The width of the rail is determined by extrapolating dimensions the rail head of a 54E1 rail as presented by pro-rail on their website, see Figure 4.1. In reality the rail head is rounded with a radius of 300. For this analysis the linear distance of the R300 part and the connection radii R13 is approximated to be 40 mm, which is used for the dimension of the track width b_t.

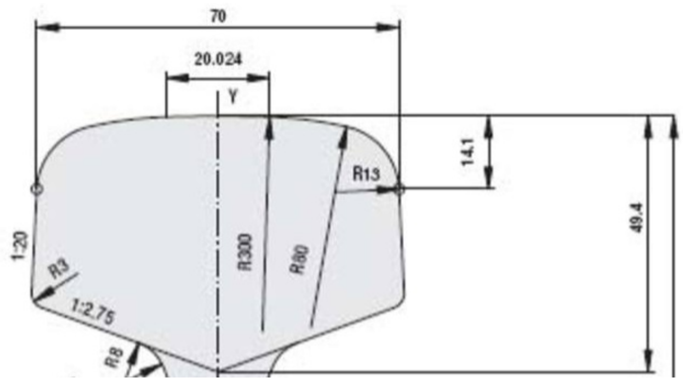


Figure 4.1: Rail head of 54E1 rail. The rolling surface is in reality curved. [12]

4.1.1. Transition value

The transition value, denoted as y_t in Figure 4.2, determines the lateral position on the track where the contact scenario shifts.

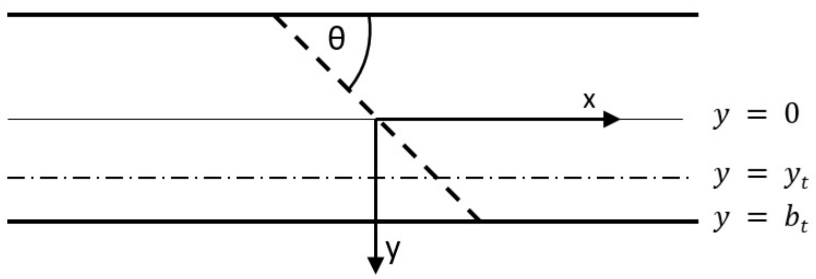


Figure 4.2: Top view of the rail with transition value y_t and track boundary b_t and junction angle θ.

The expression for the transition value follows from the geometry of the joint and is derived in section 3.3:

$$y_t = \sin\left(\frac{\beta}{2}\right)\tan(\theta)R$$

It is important to notice that this value should fall between the outer limits of the rail:

$$-b_t < y_t < b_t$$

In Figure 4.3 the transition value y_t is plotted as a function of the dip angle for the selected junction angles.

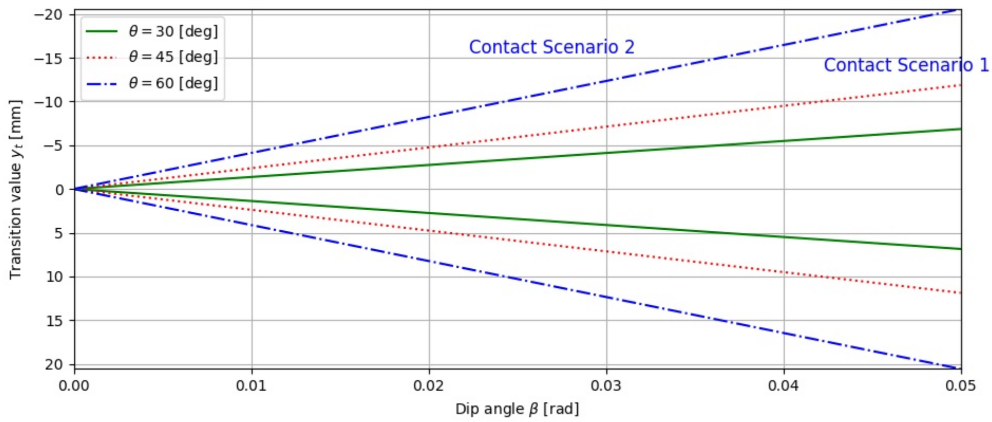


Figure 4.3: Transition value y_t as a function of β for selected values of θ

A smaller junction angle leads to lower values of y_t for the same dip angle. In other words, as the joint aligns more with the track's longitudinal direction, the contact scenario shifts from scenario 1 to scenario 2 for less eccentric wheel positions. Although this may seem counterintuitive, it can be explained by the fact that y_t is proportionally dependent on the x-coordinate of the joint, which determines the joint height, z_{joint}, through the dip angle β. The x_{joint} increases faster for smaller values of θ.

In contact scenario 1, the vertical position of the contact points on the wheel is proportionally dependent on the dip angle. The vertical height of the joint is proportionally dependent on the dip angle, multiplied by the cot θ. This results in an increase of y_t for larger β.
For larger dip angles, the transition value y_t increases. This is a result of the global dipped joint geometry becoming more dominant for larger dip angles.

If, for a certain combination of θ and β, y_t exceeds the track width b_t, contact scenario 2 cannot occur since the transition value lays out of the rail's boundaries. For the selected values, this happens for $\theta = 60$ and dip angles close to the maximum selected value.

If θ approaches 90 degrees, the transition value y_t goes to infinity, meaning contact scenario 2 does not exist for the square joint geometry.

4.2. Dynamic properties

The trajectories in contact scenario 1 and contact scenario 2 are distinct. In contact scenario 1, there is no fixed center of rotation during double-point contact, while in contact scenario 2, the center of rotation shifts to the new point of contact. Even though contact scenario 2 is different, close to the point of impact in time, its trajectory can be considered linear resulting in a trajectory comparable to that of contact scenario 1, as seen in section 3.3. To examine the influence of the junction angle θ on the dynamic properties, the dip angle β and lateral position of the wheel y_w have to be chosen in such a way that contact scenario 2 occurs for all three specified values of the junction angle θ. For this reason, consulting Figure 4.3, the dip angle β and the lateral position of the wheel y_w are set to 0.03 rad and 15 mm, respectively. For contact scenario 1, the trajectory of the wheel's center of mass remains unaffected by its lateral position on the rail, as long as this lateral position doesn't exceed the transition value y_t, see Figure 4.4. In this scenario, the incoming and outgoing angle of the trajectory during impact are equal and only dependent on the dip angle β. In contact scenario 2 the outgoing angle changes due to the interaction with the joint between the rail ends. In Figure 4.5 the dynamic properties of both contact scenario's are compared. The dynamic properties are plotted absolute in time. For contact scenario 1 the point of impact in x-direction always aligns with the point in time $t = 0$. For contact scenario 2, due to the geometry of the joint, the point of impact shifts horizontally depending on the junction angle. The shape of the trajectory in the plots is greatly exaggerated to increase readability, in reality the trajectory appears to be almost horizontal.

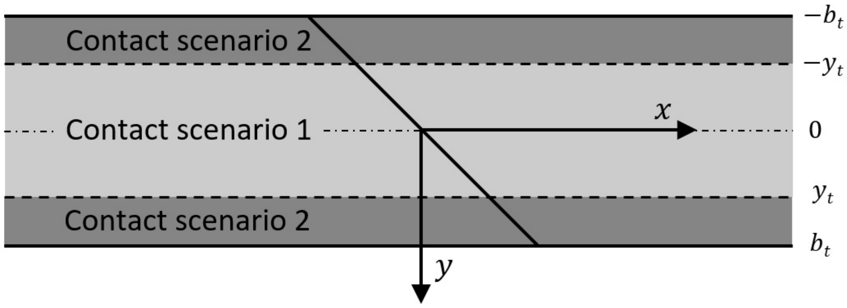


Figure 4.4: Bandwidth of y_c for which contact scenario 1 occurs. The trajectory remains unchanged as long as y_w remains within this bandwidth.

The dynamic performance of contact scenario 1 is better than that of contact scenario 2. Additionally, the dynamic performance decreases as the junction angle becomes smaller.
A smaller junction angle θ leads to larger change in direction during impact. This is a result of longitudinal position of the joint x_{joint} increasing faster for smaller junction angles, which results in larger vertical mismatch between the rail ends at lower y-values. As a result of this, the jump in the vertical velocity, the jump in vertical momentum and the impulse at the moment of impact is greater for smaller junction angles.

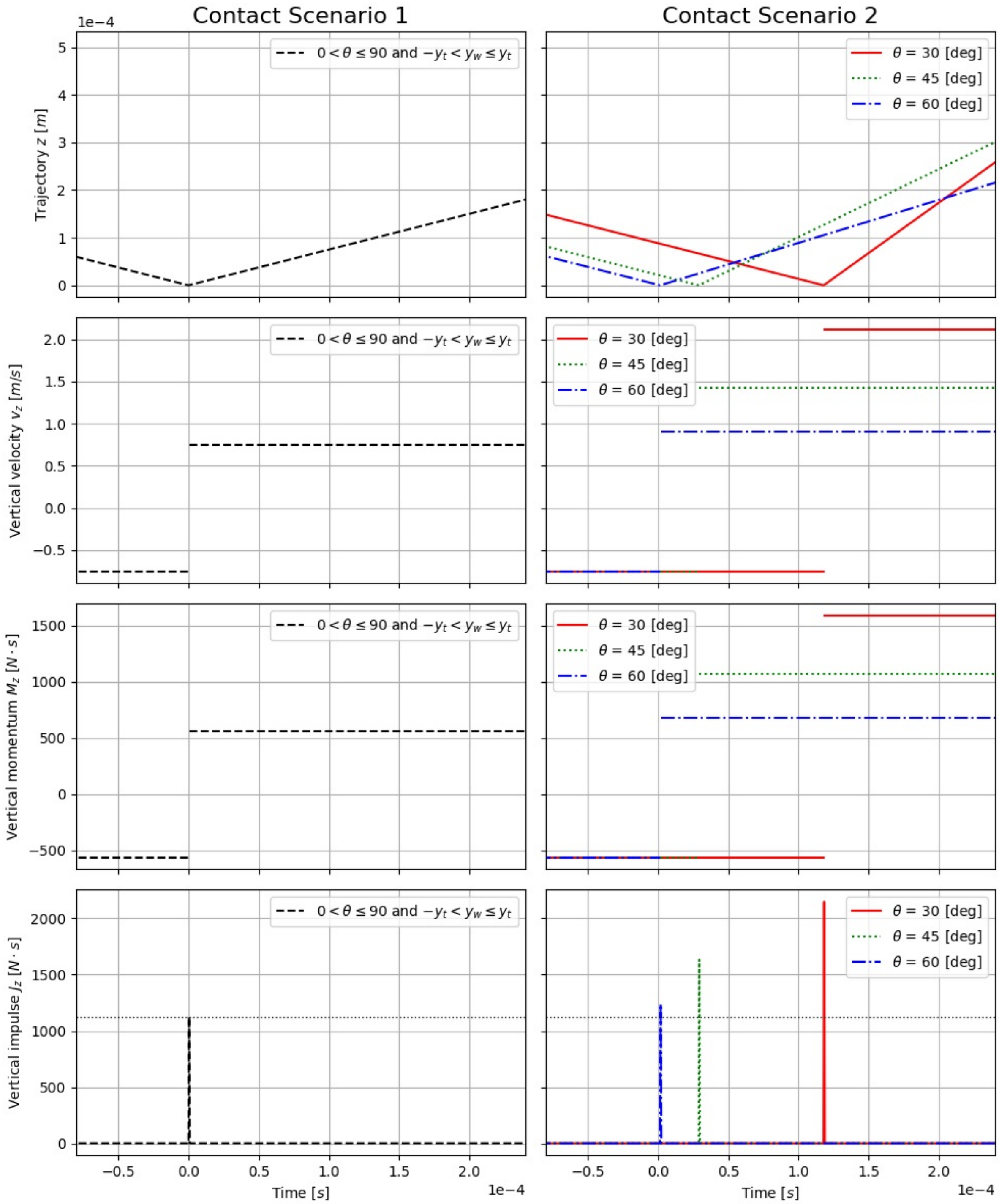


Figure 4.5: Time aligned plots for the trajectory, vertical velocity, vertical momentum and vertical impulse of both contact mechanism. $y_w = 15$ mm $\beta = 0.03$, $R = 475$ mm, $M = 750$ kg, $v = 50$ m/s

4.2.1. Vertical Impulse

Since the vertical impact force is not defined in this analysis due to the absence of material elasticity, the magnitude of the vertical impulse is used as the measurement of impact. For contact scenario 1, the vertical impulse is dependent on the dip angle β, the train velocity v and the mass of the wheel M. For contact scenario 2, the vertical impulse is dependent on the transverse position of the wheel y_w, the position of the transition value y_t, the dip angle β, the train velocity v and the wheel radius R. In Table 4.2 both contact scenario's and the parameters influencing the vertical impulse are shown. In

Table 4.2: Overview of parameters influencing the vertical impulse for each contact scenario

Contact Scenario	Parameters influencing vertical impulse
1	β, M, v
2	y_w, y_t, β, M, v, R

Figure 4.6 the vertical impulse J_z during impact is plotted against the dip angle β. The disc representing the wheel is located 15 mm from the center line of the joint ($y_w = 15$) mm to ensure that contact scenario 2 occurs for all the selected junction angles.

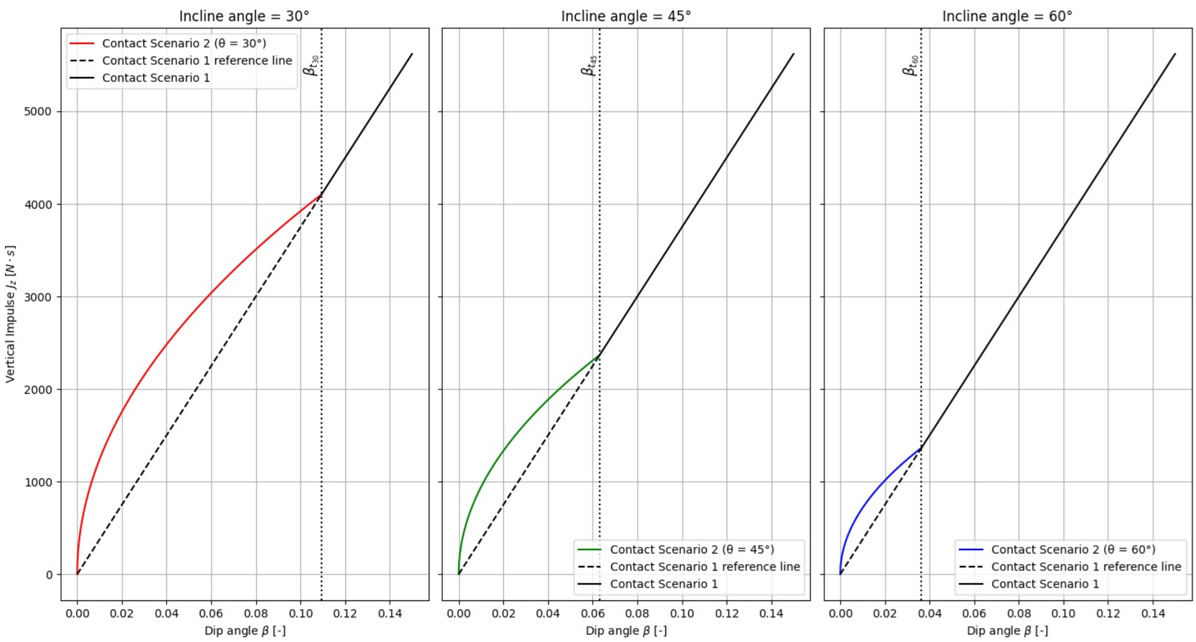


Figure 4.6: Vertical impulse as a function of the dip angle β for selected junction angles. The disc representing the wheel is located at $y_w = 15$ mm, $M = 750$ kg, $R = 475$ mm, $v = 50$ m/s

Due to the chosen eccentric position of the wheel, contact scenario 2 initially occurs for all junction angles. However, as the dip angle increases, the contact scenario transitions from scenario 2 to scenario 1. This shift is caused by the dipped joint geometry of contact scenario 1, which raises the height of the contact points as the dip angle increases, as seen in Figure 3.11.

Velocity

In Figure 4.7 the effect of the velocity on the vertical impulse is illustrated. It can be observed that the relation between velocity and vertical impulse is linear for both contact scenario's. Velocity also has no theoretical effect on which contact mechanism occurs.

4.3. Top View Visualization of Contact Points during Double-Point Contact

In Figure 4.8 four top views of the rail are shown, each presenting the location of the contact points and vertical centerline of the wheel for one of the junction angles. During double-point contact, contact

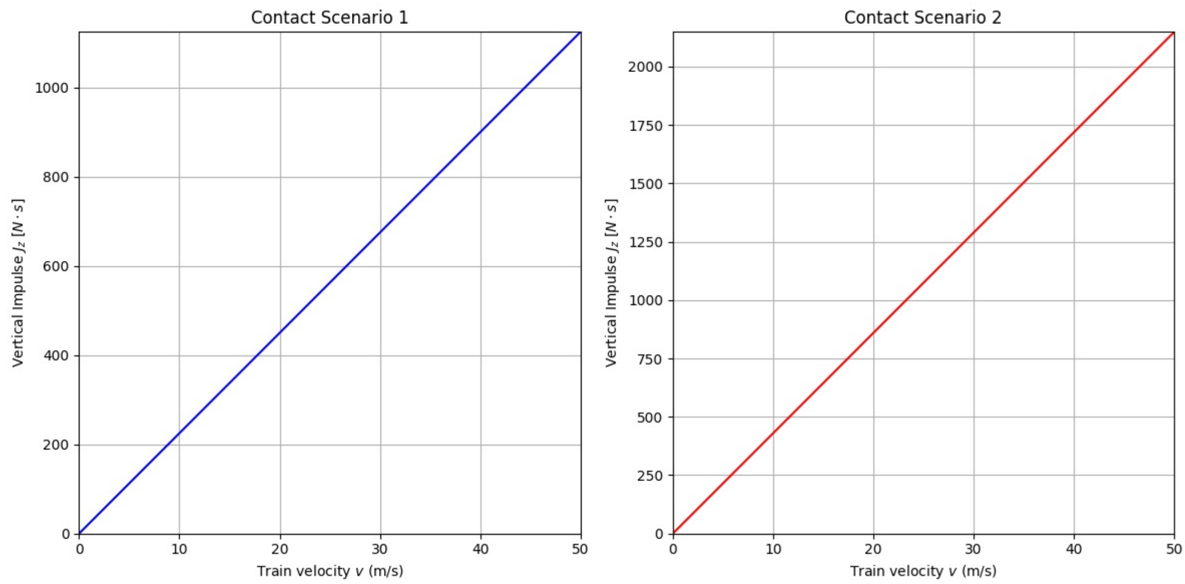


Figure 4.7: Effect of train velocity on vertical impulse

point A is positioned at the cut-off and contact point B is located on the rail end. The bandwidth of contact scenario 1 is marked in dark gray and the bandwidth for contact scenario 2 is marked in light gray.

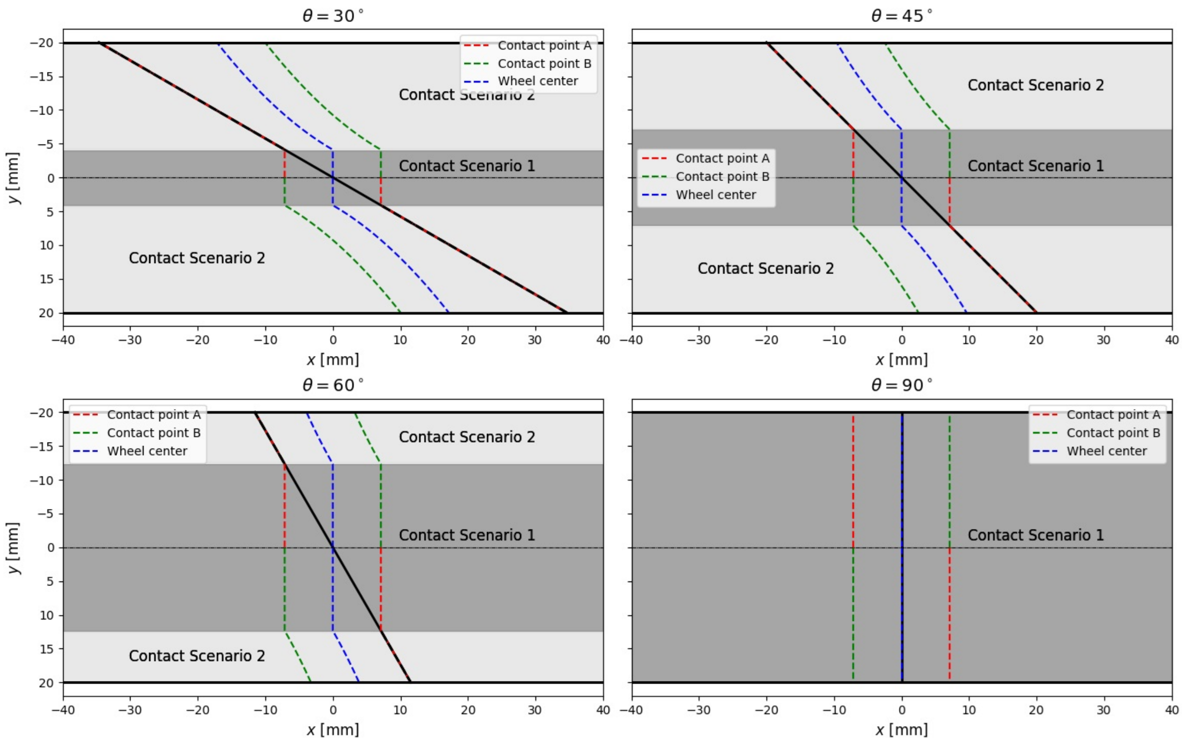


Figure 4.8: Location of the contact point A and B and vertical centerline of the disc when double-point contact occurs for selected junction angles. $\beta = 0.03$ rad $R = 475$ mm, $b_t = 20$ mm

In contact scenario 1, both contact points remain on the rail as the vertical mismatch between the rail ends is not large enough for the wheel to interact with, as seen in Figure 3.11. In contact scenario 2, contact point A is always located on the rail end. As the position of the wheel y_w increases, and the horizontal distance between contact point A and the vertical centerline x_A increases, resulting in a larger angle α, as seen in subsection 3.3.4. For smaller junction angles, the bandwidth of contact scenario 2 increases and for a junction angle of 90 degrees, contact scenario 2 does not occur.

5
Local analysis: the Wheel-Rail Interface

5.1. Introduction to contact pressure analysis

5.1.1. Background

The local interaction between wheel and rail is an important part of railway engineering. At the wheel-rail interface, the mechanical forces from the rolling stock are transferred to the track, making it a critical point of study for understanding complex interactions that influence wear, fatigue and the structural integrity of both the wheel and rail.

The contact pressure refers to the distribution of force over the contact area between the wheel and rail and plays an important role in the wheel-rail interaction. It determines how the stresses are transmitted through the materials which influences things like wear patterns, crack forming and the development of other surface defects.

The wheel-rail contact pressure is complex to study, due to the combination of varying dynamic forces in combination with rolling and sliding motions at the contact patch. The increase in computational power of the past few decades have allowed for more complex numerical computational techniques, such as the finite element method (FEM), to simulate the wheel-rail interaction with greater precision than analytical models. Despite the accuracy of these models, they are time consuming and require significant computational resources to run.

To address this, the analysis simplifies the problem to a quasi-static interaction with a constant normal force. Although this doesn't allow for the same precision as the more complex modelling methods that are available, it does provide initial insights in various key parameters of the interaction between wheel and rail for insulated rail joints with a non-orthogonal junction angle.

5.1.2. Objectives

The primary objective of this analysis is to investigate the influence of the junction angle of an insulated rail joint with a non-orthogonal junction angle on the maximum contact pressure between wheel and rail.

To achieve this, the chapter addresses the following research question:

- How does the junction angle of an insulated rail joint, particularly at very small dip angles $(\beta \approx 0)$, affect the maximum normal contact pressure at the wheel-rail interface?

This investigation aims to provide deeper insights into the impact of variations in the junction angle on the contact area, contact pressure and rate of change of the contact pressure.

5.1.3. Scope

To study the normal contact pressure that occurs during wheel-rail interaction as a wheel crosses a joint, a 2D numerical quasi-static contact model was adopted. It is assumed that the joint experiences negligible elastic displacement and dip under the normal force throughout the interaction. Specifically, the dip angle is considered to be nearly zero $(\beta \approx 0)$, and small deformations are mitigated by the material's elasticity. In practical terms, this implies that the joint's stiffness is nearly equivalent to that of the continuous track, resulting in minimal vertical deformation due to the train's weight.

- Boundaries and limitations
- Relevance to thesis

5.2. Modelling of local contact pressure

5.2.1. Model description

Rail geometry, contact patch geometry and step size

To model the contact pressure during wheel-rail interaction, the rail geometry illustrated in Figure 5.1 was used. A coordinate system is defined with the origin located at the top left corner of the gap. The parameter g represents the absolute distance between two track segments, which is predetermined due to the electrical isolation function of the joint. The parameter θ indicates the junction angle of the track in the xy-plane. When $= 90°$, the joint functions as a squared insulated rail joint. The longitudinal length of the gap g_{long} is the distance between rail-head and -tail in longitudinal direction and is dependent on the nominal gap length g and the junction angle θ. Smaller values of θ result in a larger longitudinal gap between the rail ends:

$$g_{long} = \frac{g}{\sin\theta}$$

Initially, the contact patch was approximated as a rectangle with dimensions (a x b). In the model, this rectangular contact patch is modelled as a polygon with 4 vertices. The horizontal position of the front of the contact patch in the coordinate system is denoted by x_{front}. The value of x_{front} increases with each iteration of the analysis, allowing the contact patch to move horizontally. The positions of the vertices of the rectangular contact patch are defined as follows, starting from the top left vertex going clockwise:

1. $(x_{front} - a, 0)$
2. $(x_{front}, 0)$
3. (x_{front}, b)
4. $(x_{front} - a, b)$

While the actual contact patch is more accurately represented as an oval, simplifying it to a rectangle is useful for an initial analysis. This approach provided an early understanding of the interaction between the contact patch and the joint, offering valuable insights into the system's behavior before complicating the analysis. Just like the contact patch, is the gap between rail-head and -tail modelled as a polygon. It is defined by four vertices spanning a parallelogram. The vertices are labeled P_{tl} (top left), P_{tr} (top right), P_{bl} (bottom left) and P_{br} (bottom right). the positions of these vertices are defined as follows:

1. $P_{tl} = (0, 0)$
2. $P_{tr} = (g_{long}, 0)$
3. $P_{bl} = (\frac{b}{\tan\theta}, b)$
4. $P_{br} = (\frac{b}{\tan\theta} + g_{long}, b)$

By defining the positions of the vertices of the gap based on the dimensions of the contact patch makes it look like the contact patch has the same width of the rail. This is done for modelling convenience and has no effect on the results of the analysis. In reality the width of the rail far exceeds the width of the contact patch. The top right corner of the contact patch is initially located at the origin of the coordinate system to coincide with the top left corner of the gap. In this position $x_{front} = 0$, marking the start of the quasi-static analysis. The contact patch is then translated in positive x-direction for a predetermined number of steps n. A model run is finished when the left edge of the contact patch coincides with the

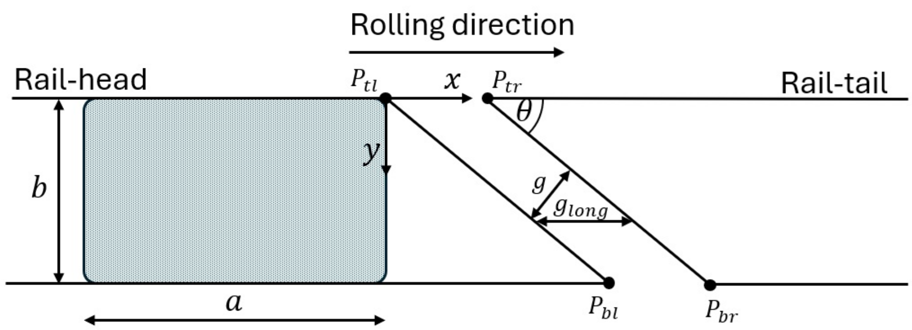


Figure 5.1: Dimensions of contact patch and rail

bottom right corner of the gap, at position $x_{end} = x_{br} + a$. The step size Δx can be determined by dividing the total horizontal distance travelled by the contact patch by number of steps $n - 1$:

$$\Delta x = \frac{x_{end}}{n - 1}$$

In Figure 5.2, the model for an analysis consisting of four steps $(n = 4)$ is presented. Each color corresponds to a different position of the contact patch as it traverses the joint gap, which is depicted in black.

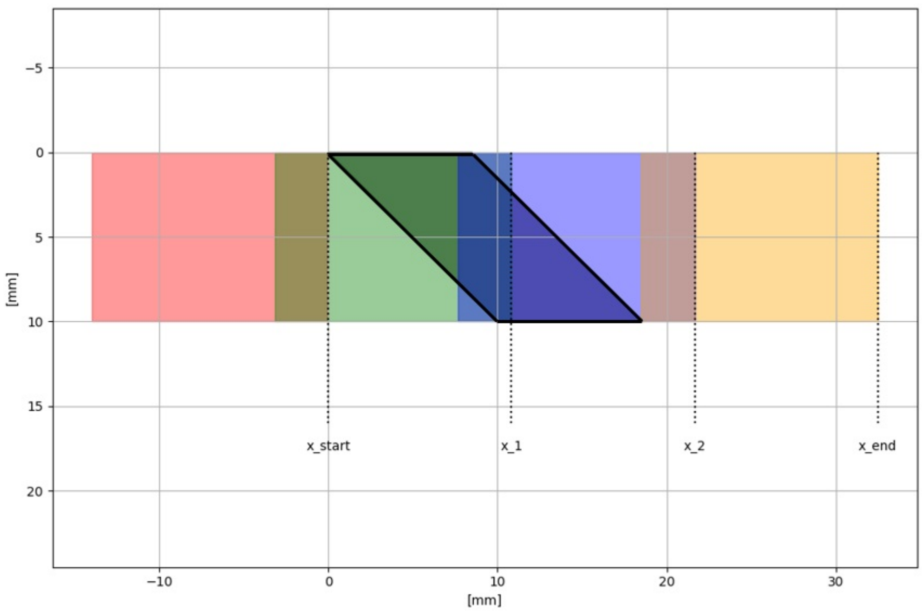


Figure 5.2: Positions of rectangular contact patch for analysis with 4 steps $(n = 4)$ with $a = 14$, $b = 10$, $g = 6$, $\theta = 45°$

Total contact area, reduction area and active contact area

Initially, the entire area of the contact patch is in contact with the rail-head. This total contact area, or initial contact area is denoted as A_0 and is defined as:

$$A_0 = ab \tag{5.1}$$

As the contact patch traverses the gap of the joint the area in contact with the rail is reduced. The part of the area of the contact patch that is in contact with the rail (either rail-head, rail-tail or both) is referred to as the active contact area and is denoted as A_i where i denotes the step number of the analysis. The active contact area is capable of transferring force from the wheel to the rail. The part of the contact patch that is located above the gap of the joint is referred to as the reduction area. This area is not capable of transferring force from the wheel to the rail and is denoted as R_i. The active contact area is defined as:

$$A_i = A_0 - R_i \tag{5.2}$$

When the rectangular polygon, representing the contact patch, traverses in x-direction, it intersects with the parallelogram shaped polygon representing the gap. The result of these two polygons intersecting is a new polygon located 'above' the gap between rail-head and -tail. This polygon represents the reduction area R_n. The vertices of the polygon representing the reduction area can be determined using the following steps:

1. Define linear functions of the line segments of both the contact patch and the gap in the coordinate system.
2. Find points of intersection by setting two equations equal to each other
3. Check if point is located within the boundaries of the gap

Now that the vertices of the polygon representing the reduction area are known the area can be determined using the shoelace formula. The shoelace formula, or Gauss's area formula, returns the area of any polygon connected by straight line segments described in a Cartesian coordinate system. If the x and y values of the intersection points are denoted as x_j and y_j for $j = 1..n$ the reduction area can be calculated using the shoe lace formula:

$$2R_i = \begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} + \begin{vmatrix} x_2 & x_3 \\ y_2 & y_3 \end{vmatrix} + ... + \begin{vmatrix} x_n & x_1 \\ y_n & y_1 \end{vmatrix} \tag{5.3}$$

Now that the total contact area A_0 and the reduction area R_i are known the active area A_i can be determined by substituting Equation 5.1 and Equation 5.3 into Equation 5.2. In Figure 5.3 the relationship between the total contact area, reduction area and active contact area is shown. The areas are normalised by dividing the active contact area and reduction area by the total area. In practise, the size

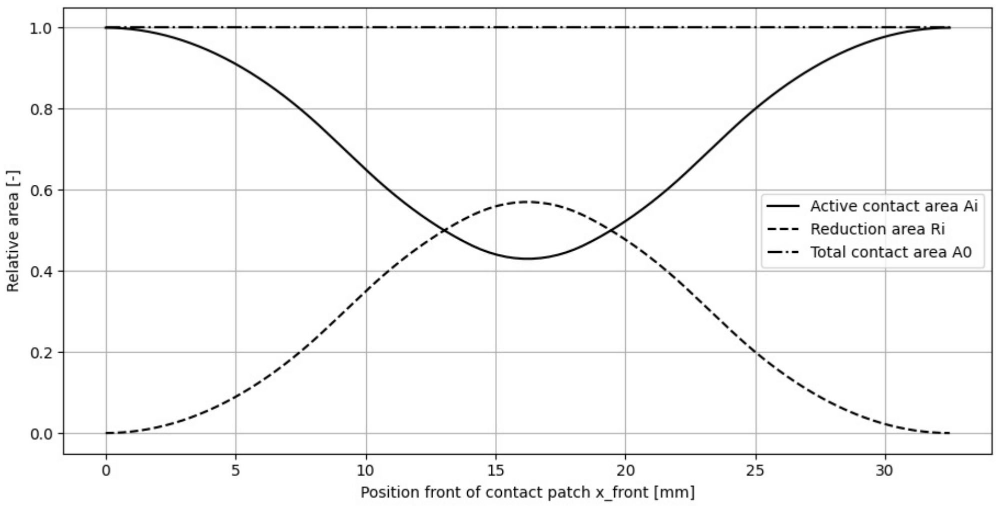


Figure 5.3: Relationship between total contact area, reduction area and active contact area for a rectangular contact patch. $a = 14$, $b = 10$, $g = 6$, $\theta = 45°$ $n = 100$

of the contact patch varies as it crosses a rail gap. As the active contact area reduces, the pressure in the remaining area increases. This increase in stress results in a slight increase of contact area in the remaining active contact area. However, due to the relatively high stiffness of the contacting materials in wheel-rail contact the increase in area is minimal. Since the redistribution of contact area remains small in wheel rail contact it is considered negligible for this analysis. Neglecting deviations of the ideal contact shape is a common way to deal with complex wheel-rail interaction problems and allows for analytical solutions [13].

Uniform contact pressure

To simplify the analysis, it is initially assumed that the contact pressure is uniformly distributed over the active contact area. While this assumption is not entirely realistic it, like the assumption of a rectangular contact patch, serves as a useful starting point for understanding the model. Under this assumption, the uniform contact pressure for each iteration P_i, assuming a constant normal force N acting on the wheel, is defined as:

$$P_i = \frac{N}{A_i}$$

The assumption of a constant normal force acting on the wheel is in line with the assumption of the dip angle being nearly zero since no changes in vertical velocity will occur during the interaction. The normal force's magnitude then merely dependents on the train's weight, which can vary slightly across different train models. According to EN 15528 [14] maximum axle load for railways in the Netherlands is 22.5 tonnes for new rails. Dividing this by 2 (two wheels per axle) gives an approximate normal force of 11.25 tonnes per wheel. For practical purposes a normal force of 100 kN ($\tilde{1}$0 tonnes) was used. The uniform contact stress at the beginning of the analysis corresponds to the position of the contact patch in which the total contact area of the contact patch is active. This initial contact stress is denoted as:

$$P_0 = \frac{N}{A_0}$$

Figure 5.4 shows the absolute contact pressure and the contact pressure relative to the initial contact pressure during joint traversal of a rectangular contact patch.

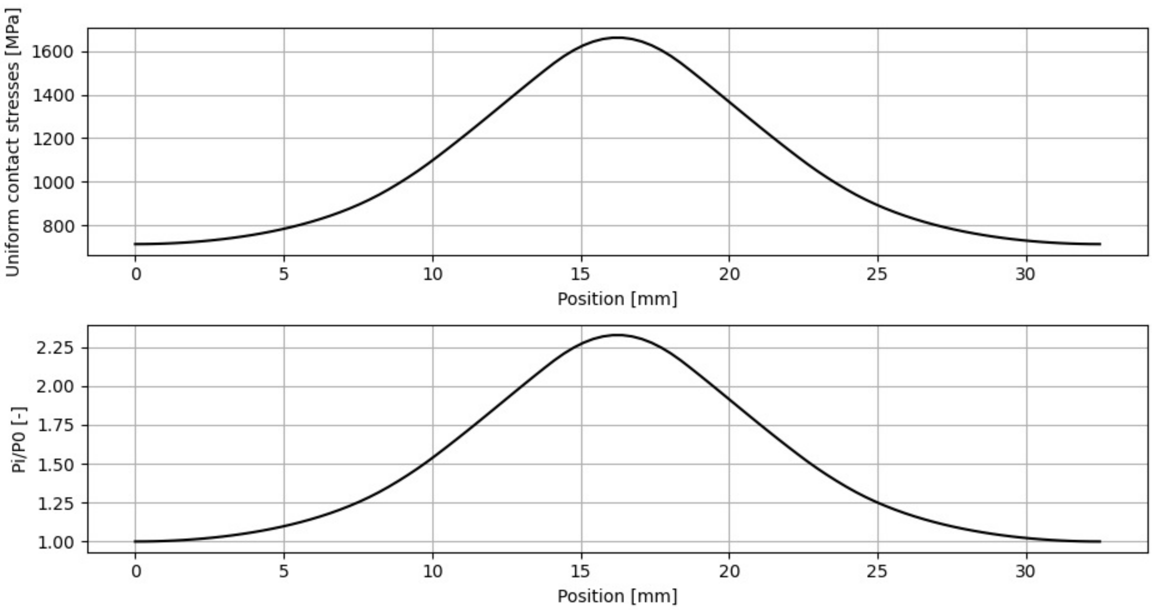


Figure 5.4: Absolute and relative uniform contact pressure for a rectangular contact patch traversing the joint. $a = 14$, $b = 10$, $g = 6$, $\theta = 45°$ $n = 100$

Improved contact patch shape: ellipse shaped polygon

Initially, a rectangular contact patch was used to simplify the building and testing of the model. However, in reality, the contact patch more closely resembles an ellipse [15]. To create a polygon that approximates an ellipse, the position of the vertices in the coordinate system had to be determined. This was achieved using the parametric equations of an ellipse, where a denotes the major axis and b denotes the minor axis of the ellipse Figure 5.5. The x and y coordinates of an ellipse are given by:

$$x(\theta) = \frac{a}{2} \cos\theta$$

$$y(\theta) = \frac{b}{2} \sin\theta$$

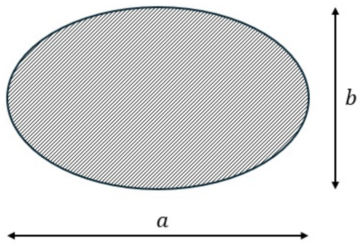


Figure 5.5: Oval contact patch with major axis a and minor axis b

To create a polygon, these equations can be sampled at regular intervals for a specified number of vertices, ranging from 0 to 2π. By connecting these vertices with straight lines, an approximate polygonal shape can be formed. This method allows the contact patch to be modeled with straight line segments which means the Shoelace theorem remains valid. Using more vertices leads to a better approximation of an oval but it also increases the computation time of the model. In Figure 5.6 a polygon approximating an ellipse with 6, 10 and 100 vertices is shown. Note that only polygons with an even amount of vertices result in a shape that is symmetrical around both the x and y axis. For an uneven amount of vertices the shape will only be symmetrical around the x-axis. It is important to note that the

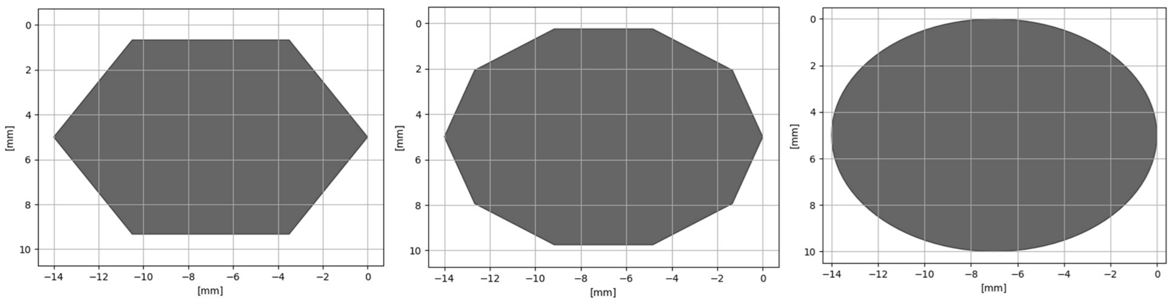


Figure 5.6: Approximation of an ellipse with 6, 10 and 100 vertices. $a = 14$, $b = 10$

total area of the polygon changes with the number of vertices, even when the parameters a and b remain constant. This difference in total area should be considered when analyzing the model's results. To make comparison between different contact shapes possible, it is useful to normalize the active contact areas and contact stresses relative to their initial values, as done in Figure 5.3 and Figure 5.4.

Time Domain

The initial model was established in the position domain, where each step in horizontal translation corresponded to a specific distance. Given that the interaction between the wheel and rail is brief, the train's velocity is assumed to be constant. To transition the model into the time domain, a linear operator v, representing the train's velocity, is applied as follows:

$$x = v \cdot t$$

6
Results Wheel-Rail Interface Analysis

In chapter 5, the uniform contact pressure of a wheel traversing an insulated rail joint with a non-orthogonal junction angle was analyzed. Initially, as simplification, the contact patch was considered as a rectangle. To improve the model, later an elliptical contact patch was considered. The uniform contact pressure is determined for various junction angles using a quasi static model. By assuming a negligible dip angle, the normal force was considered constant during joint traversal. The uniform contact pressure was determined by calculating the contact area between wheel and rail in each step in position or time and dividing the constant normal force by the calculated contact area. This chapter presents the results of this analysis.

6.1. Rail and contact patch parameters

The parameters used in this analysis are presented in Table 6.1.

Table 6.1: Overview of used parameters in wheel-rail interface analysis

Symbol	Parameter	Value	Unit
N	Normal Force	112.5	kN
a	Shape major axis	16	mm
b	Shape minor axis	12	mm
g	Nominal gap	6	mm
θ	Junction angle of joint	[30, 45, 60, 90]	degrees
v	Train velocity	50	m/s
n_steps	Number of steps	1000	-
$n\_vertices$	Number of vertices	100	-

The normal force is determined to be 112.5 kN. According to [14] the axle load in the Netherlands is 225 kN. Dividing this by 2 (two wheels per axle) results in 112.5 kN.

The values of a and b represent the major and minor axes of the ellipse. For an elliptical contact patch, this results in a smaller area compared to a rectangular contact patch. To ensure that the the performance of different shapes on contact pressure can still be compared, the area's and pressures are displayed relative to their initial values. The approximate size of a contact patch in reality is between 1 and 2 cm^2. For the chosen values for a and b, both the rectangular and elliptical contact patch will fall within this range.

The nominal gap size g is determined by the requirement to prevent the flow of current across the joint. According to Dutch standards, the minimum nominal distance between the 2 rail ends should be at least 6 mm. Note that for a cut at a non-orthogonal junction angle between the rail ends the longitudinal distance between the rail ends will exceed 6 mm.

The junction angle θ of the joint can theoretically vary from 0 to 90 degrees. An angle of 90 degrees will result in a standard squared joint geometry. As the angle decreases, the cut between the rail ends becomes more aligned with the longitudinal direction of the track. However, the angle can't be zero degrees, as this would result in the cut becoming fully oriented in longitudinal axis, , making the cut parallel to the rail and preventing it from crossing to the other side. As the angle decreases further, the longitudinal length of the diagonal cut increases significantly. In practice, an insulating end plate material is placed within this cut. If the longitudinal length becomes too large, it may become impractical to replace the end plate material in the event of a defect, which is likely to occur over time due to the high and frequent loads applied to the rail. For this reason three practical values for the junction angle θ are selected.

The train's velocity is based on the maximum speed of intercity trains in the Netherlands, which is approximately 160 km/h.

6.2. Uniform Contact Pressure

The quasi-static uniform contact pressure model is used to determine the relative uniform contact pressure for the selected junction angles, the results are shown in Figure 6.1.

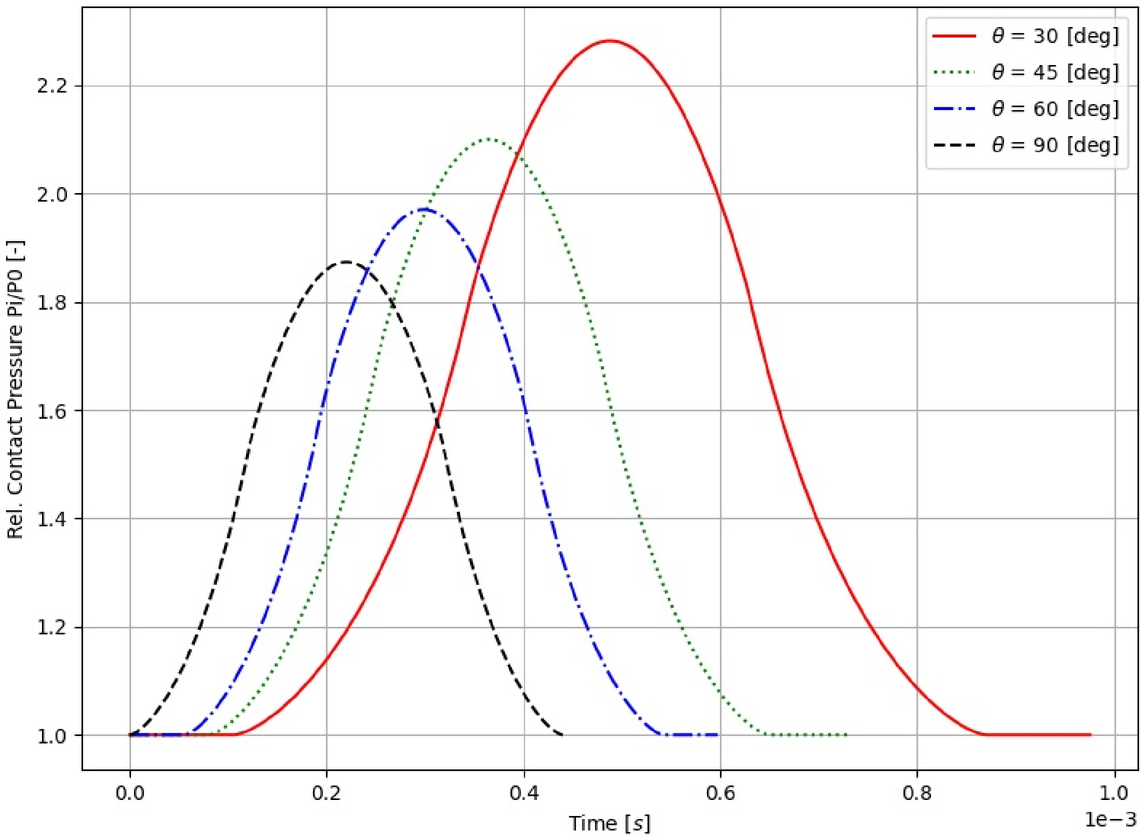


Figure 6.1: Relative contact pressure for selected values of the junction angle.

As the junction angle decreases, resulting in a longer cut along the longitudinal direction of the rail, the longitudinal gap between the rail ends must increase to maintain the minimum required nominal gap length, as shown in section 5.2. This increase in longitudinal gap length leads to a larger total gap area between the rail ends, which in turn causes the uniform contact pressure between the wheel and rail to increase. The horizontal shift between the graphs is a result of the model configuration. The initial position of the contact patch remains the same for every junction angle, but for smaller junction angles the distance between the initial position of the contact patch and the start of the gap increases.

6.3. Rate of change

Given that the uniform contact pressure on each time step is known, the rate of change in this pressure can be determined. A higher rate of change of the stresses results in greater damage to both the wheel and rail. Figure 6.2 illustrates the rate of change in uniform contact pressure for the selected junction angles. The other parameters are as presented in section 6.1.

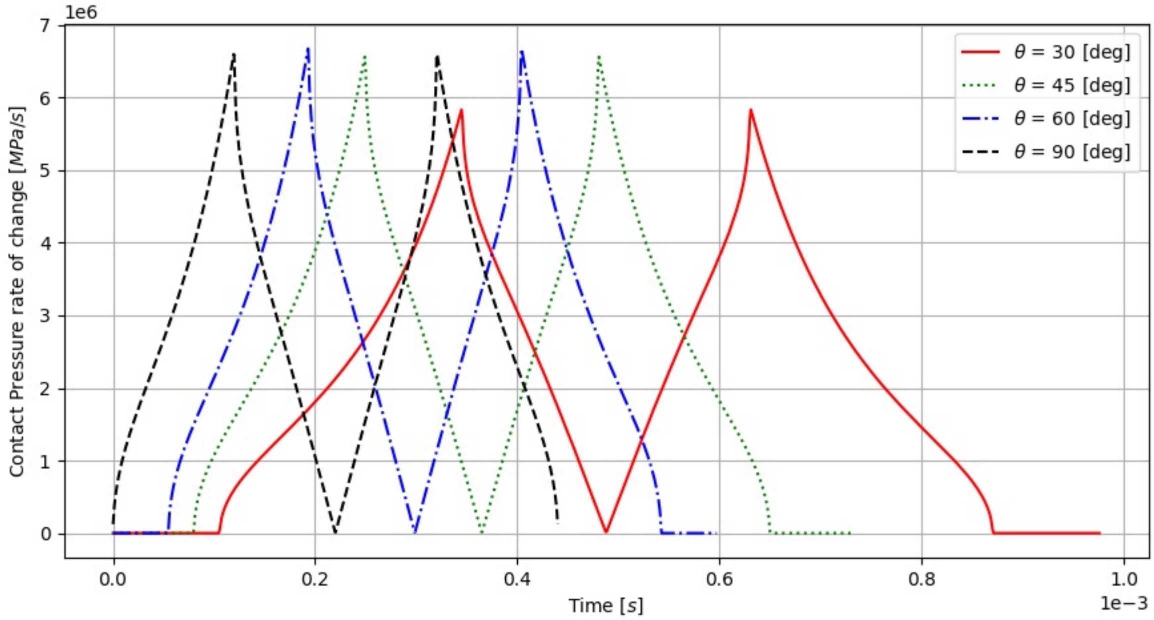


Figure 6.2: Rate of change in uniform contact pressure for an elliptical contact patch crossing the gap between the rail ends at various junction angles.

For junction angles of 90, 60 and 45 degrees, the rate of change shows no significant differences. However, at 30 degrees a significant decrease occurs. To better understand the the graph, Figure 6.3 shows the graph for a 30-degree junction angle with 5 points of interest marked, each explained below:

1. The contact patch leaves the rail head, resulting in an increase of contact pressure.
2. The maximum increase in contact pressure is reached. The contact patch starts making contact with the rail-tail increasing the contact area on this side and reducing the rate of change.
3. The contact patch is positioned exactly in the middle of the gap, resulting in 2 symmetrical contact areas on rail head and tail. The rate of change is 0 in this position. This point coincides with the maximum uniform contact pressure, see Figure 6.1.
4. The peak in the rate of change is reached due to the contact patch losing contact with the rail-head. This configuration is the same as in position 2. Note that the absolute values are plotted, the actual values of this peak are negative.
5. The contact patch is completely located on the rail tail.

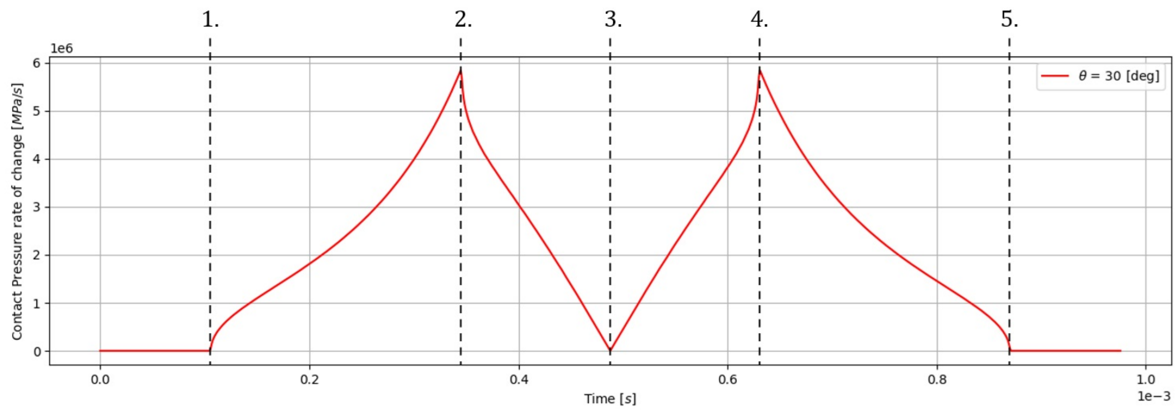


Figure 6.3: Rate of the change of uniform contact pressure for a junction angle of 30 degrees with 5 points of interest marked.

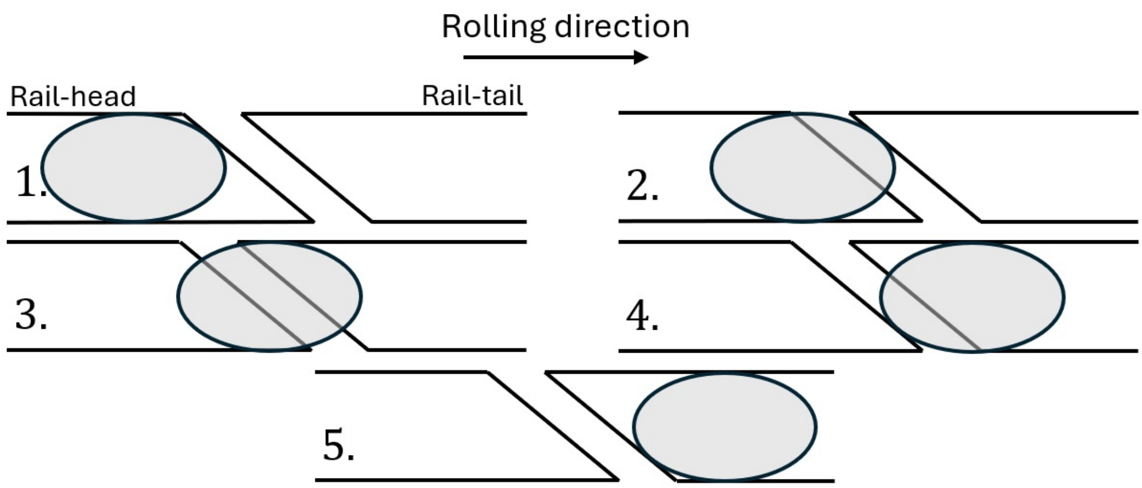


Figure 6.4: Five positions of an elliptical contact patch traversing the gap of a joint with a junction angle. In reality, the rail is wider than the contact patch.

6.4. Junction angles outside of practical range

The notable reduction in rate of change between a junction angle of 45 and 30 degrees suggests that smaller junction angles have a greater effect on the rate of change. This is expected since for smaller junction angles, the longitudinal length of the joint increases significantly, which extends the time of interaction and decreases the rate of change in contact pressure. For these smaller angles, the uniform contact pressure also increases, as discussed in section 6.2. Figure 6.5 shows the rate of change for small junction angles, while Figure 6.6 shows the corresponding uniform contact pressure.

Although a significant decrease in the rate of change and a slight increase in uniform contact pressure at small junction angles might suggest that a longer cut would be an ideal solution, there are practical limitations. Due to the high and frequent dynamic loading as a result of the train axle's passing by, insulated rail joint often get damaged. Replacing the end plate material or doing other repairs to a long joint might be very impractical.

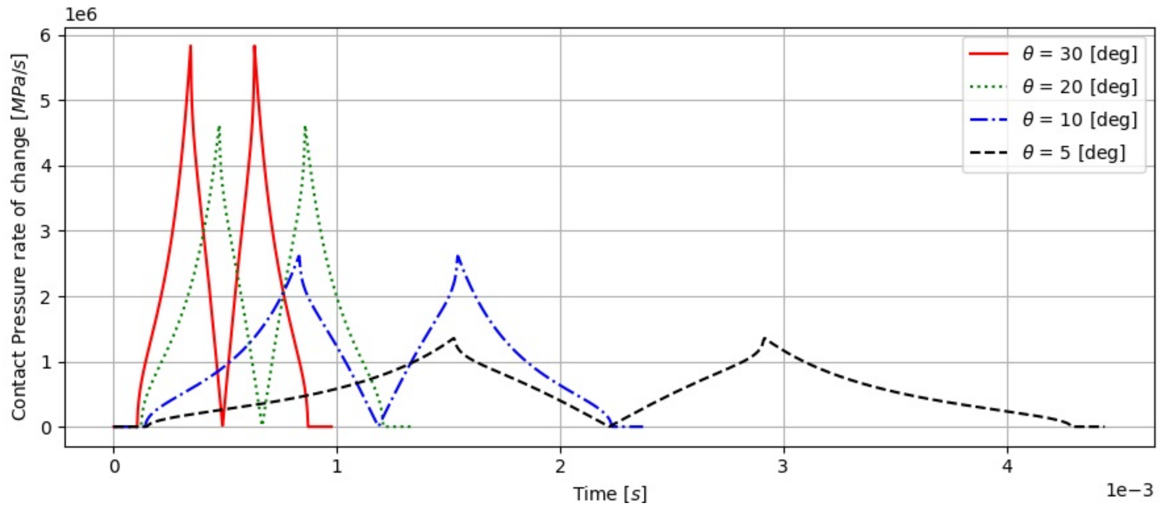


Figure 6.5: Rate of change in uniform contact pressure for small junction angles.

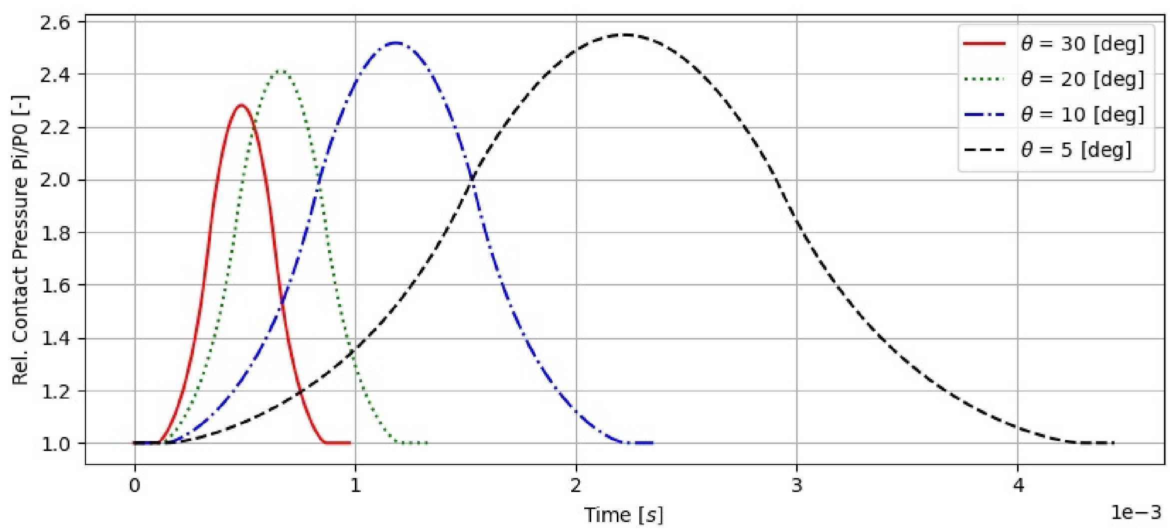


Figure 6.6: Uniform contact pressure for small junction angles.

7

Conclusion and recommendations

The effect of the junction angle on both global wheel-rail interaction and local contact pressure in insulated rail joints has been studied. This chapter presents a conclusion and recommendations of both the global wheel-rail interaction and local wheel-rail interface analysis.

7.1. Conclusion

In a kinematic approach, the global wheel-rail interaction analysis indicated that a diagonal cut for junction angles within a practical range does not improve the dynamic wheel-rail interaction compared to the traditional squared cut joint design. In fact, for eccentric wheel positions on the running surface, the magnitude of the local impact contact force was shown to increase. The analysis showed the possibility of two distinct contact scenarios, depending on lateral wheel position and for dip angles larger than 0. As long as contact scenario 1 occurs, the effective geometry and the resulting vertical impulse remained identical to those of squared joints. However, in contact scenario 2, the effective geometry of the joint changes, leading to an increase in vertical impulse.

When the wheel was considered a cylinder with finite width and the contact between wheel and rail was considered as line contact, the introduction of the junction angle increased the likelihood of less favorable contact conditions for contact scenario 1 and guaranteed less favorable contact conditions for contact scenario 2. For some wheel positions in contact scenario 1, and all wheel positions in contact scenario 2, the contact would reduce to a single point. This could lead to contact conditions that cause significant damage to both the wheel and rail, characterized by a short impulse in time combined with a strongly localized load in space.

The point and line contact assumptions were made to simplify the modeling of the interaction over time. In reality, the contact patch is elliptical, with a 3D Hertzian stress distribution. Moreover, the wheel is conical, and the rail has a curved profile. The precise effect of the simplification of the contact on the exchange of vertical momentum between wheel and rail during impact remains unknown. Although the elasticity of the materials provides a smoothing effect during interaction, its contribution is expected to be only qualitative due to the relatively stiff steel-on-steel contact between the wheel and rail.

In practice, eccentric wheel positions should be avoided for joints with a junction angle smaller than 90 degrees, unless the dip angle can be controlled near zero. However, avoiding eccentric wheel positions near the rail joint may be challenging, as the change in vertical stiffness near rail joints may lead to greater variations in lateral wheel positioning on the rolling surface. Keeping the dip angle close to zero may only be achievable by adding significant material to the joint to match the stiffness of the continuous rail for all passing axles and loads. This could however on its turn lead to a rise in dynamic stiffness because of the added mass.

The local contact pressure analysis showed that a cut at a non orthogonal junction angle increases the relative uniform contact pressure between the wheel and rail for dip angles close to zero. For small junction angles (resulting in a longer cut in the longitudinal direction), outside of the considered practical range, the rate of change of the contact pressure was greatly reduced.

The reduction in the rate of change of the contact pressure becomes more significant at smaller junction angles. At these angles, the horizontal distance of the cut between the rail ends increases substantially, leading to a much longer cut along the longitudinal direction of the rail. This increased interaction time reduces the rate of change in contact stresses. However, for small junction angles, the relative contact pressure barely increases, making it a good theoretical solution, as a cyclic fast increase in local stress over a short time period is expected to increase the susceptibility to damage.

In practice, very small junction angles should be avoided. The long longitudinal cuts make it impractical and time-consuming to replace the end plate material, if necessary. Given the high axle loads, dynamic loading conditions, and frequent cycles characteristic of railway systems, damage to the rail becomes inevitable, making very long diagonal cuts an impractical solution.

7.2. Recommendations

7.2.1. Global Wheel-Rail Interaction Analysis

To improve the accuracy of the global wheel-rail interaction analysis, the description of the geometry of the wheel, rail, and contact patch should be improved. The following recommendations are proposed:

- Incorporate conicity into the cylindrical model of the wheel.
- Add lateral curvature of the rail surface to better reflect real-world conditions.
- Transition from point or line contact models to a Hertzian contact patch for a more accurate representation of the interaction.

Although the assumption of neglecting elasticity is reasonable for stiff materials like steel, adding material elasticity will improve the accuracy of the analysis. Since adding elasticity through analytical methods might be too complex, a finite element method (FEM) should be considered. This would result in a smoothing effect on the wheel-rail interaction, removing the singularity in the vertical wheel velocity, and providing a more accurate definition of the vertical impulse force.

For squared joint geometries, the gap between the rail ends is often neglected, allowing the rail ends to connect directly. For dipped joint geometries, this assumption is justified, as the global geometry is dominant and does not affect the trajectory of the wheel's center of mass. However, due to the different effective geometry in contact scenario 2, including the gap between the rail ends might significantly affect the dynamic performance.

Including inertia could affect the results for contact scenario 2. The vertical mismatch between rail ends in step-down geometries may cause the wheel to lose contact, resulting in a different trajectory. Including inertia in future studies could improve understanding of wheel behavior in these scenarios.

7.2.2. Local Wheel-Rail Interface Analysis

The assumption of uniform contact stress underestimates the maximum contact pressure between the wheel and rail. A more accurate quasi-static analysis could adopt the Hertzian contact model, which better describes the contact pressure distribution between the wheel and rail.

In the wheel-rail interface analysis, the shape and size of the contact patch were predetermined. In reality, the size of the elliptical contact patch between the wheel and rail is not fixed, but constantly changes during interaction. Once the yield stress in the contact patch is reached, the stress will redistribute, resulting in an increase in contact patch area. Including this effect in future analyses will provide a more accurate description of the contact pressures.

References

[1] Kamran Gholamizadeh, Esmaeil Zarei, and Mohammad Yazdi. “Railway Transport and Its Role in the Supply Chains: Overview, Concerns, and Future Direction”. In: *The Palgrave Handbook of Supply Chain Management*. Springer International Publishing, 2022, pp. 1–28. DOI: 10.1007/978-3-030-89822-9_95-1. URL: https://doi.org/10.1007/978-3-030-89822-9_95-1.

[2] Feng Wang et al. “Impact of high-speed rail on population mobility and urbanisation: A case study on Yangtze River Delta urban agglomeration, China”. In: *Transportation Research Part A: Policy and Practice* 127 (Sept. 2019), pp. 99–114. DOI: 10.1016/j.tra.2019.06.018. URL: https://doi.org/10.1016/j.tra.2019.06.018.

[3] Gianluca Megna and Andrea Bracciali. “SMOOTH TRANSITION INSULATED RAIL JOINTS”. In: *2nd International Conference on Contact Mechanics and Wear of Rail/Wheel Systems (CM2022)* (Sept. 2022).

[4] Michaël J. M. M. Steenbergen. “Modelling of wheels and rail discontinuities in dynamic wheel–rail contact analysis”. In: *Vehicle System Dynamics* 44.10 (Oct. 2006), pp. 763–787. DOI: 10.1080/00423110600648535. URL: https://doi.org/10.1080/00423110600648535.

[5] M Dhanasekar and W Bayissa. “Performance of square and inclined insulated rail joints based on field strain measurements”. In: *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* 226.2 (Aug. 2011), pp. 140–154. DOI: 10.1177/0954409711415898. URL: https://doi.org/10.1177/0954409711415898.

[6] Gianluca Megna, Andrea Bracciali, and Nirmal K. Mandal. “Design, wheel-rail interaction and testing of an innovative reinforced smooth transition insulated rail joint”. In: *Wear* 530-531 (Oct. 2023), p. 205038. DOI: 10.1016/j.wear.2023.205038. URL: https://doi.org/10.1016/j.wear.2023.205038.

[7] T.X. Wu and D.J. Thompson. “On the impact noise generation due to a wheel passing over rail joints”. In: *Journal of Sound and Vibration* 267.3 (Oct. 2003), pp. 485–496. DOI: 10.1016/s0022-460x(03)00709-0. URL: https://doi.org/10.1016/s0022-460x(03)00709-0.

[8] H. Askarinejad et al. “Field Measurement of Wheel-Rail Impact Force at Insulated Rail Joint”. In: *Experimental Techniques* 39.5 (Oct. 2012), pp. 61–69. DOI: 10.1111/j.1747-1567.2012.00867.x. URL: https://doi.org/10.1111/j.1747-1567.2012.00867.x.

[9] Shervan Ataei, Saeed Mohammadzadeh, and Amin Miri. “Dynamic Forces at Square and Inclined Rail Joints: Field Experiments”. In: *Journal of Transportation Engineering* 142.9 (Sept. 2016). DOI: 10.1061/(asce)te.1943-5436.0000866. URL: https://doi.org/10.1061/(asce)te.1943-5436.0000866.

[10] H. H. Jenkins et al. “The effect of track and vehicle parameters on wheel/rail vertical dynamic forces”. In: *Railway Engineering Journal* 3.1 (1974), pp. 2–16.

[11] Georges Kouroussis et al. “Railway ground vibrations induced by wheel and rail singular defects”. In: *Vehicle System Dynamics* 53.10 (July 2015), pp. 1500–1519. DOI: 10.1080/00423114.2015.1062116. URL: https://doi.org/10.1080/00423114.2015.1062116.

[12] Railpro. *Spoorstaaf 54E1 R260Mn 12.000mm te bew*. Accessed: 2024-09-10. 2024. URL: https://www.railpro.online/nl_NL/p/spoorstaaf-54e1-r260mn-12-000mm-te-bew/846/.

[13] Andreas Heckmann et al. “The Foundation of the DLR RailwayDynamics Library: the Wheel-Rail-Contact”. In: *Proceedings of the 10th International Modelica Conference, March 10-12, 2014, Lund, Sweden*. Modellica. Linköping University Electronic Press, Mar. 2014. DOI: 10.3384/ecp14096465. URL: http://dx.doi.org/10.3384/ecp14096465.

[14] Netherlands Standardization Institute. *Railway applications - Line categories for managing the interface between load limits of vehicles and infrastructure*. Tech. rep. NEN-EN 15528. Available at: URL (if applicable). Netherlands Standardization Institute, 2008.

[15] Heinrich Hertz. “Über die Berührung fester elastischer Körper”. German. In: *Journal für die reine und angewandte Mathematik* 92 (1882). This work established the basis of what is now known as Hertzian contact theory., pp. 156–171.

[16] Emrecan Soylemez and Korhan Ciloglu. “Influence of Track Variables and Product Design on Insulated Rail Joints”. In: *Transportation Research Record: Journal of the Transportation Research Board* 2545.1 (Jan. 2016), pp. 1–10. ISSN: 2169-4052. DOI: 10.3141/2545-01. URL: http://dx.doi.org/10.3141/2545-01.

[17] A. Jain et al. “Comparative analysis of the dynamic amplifications due to inhomogeneities at railway transition zones”. In: *Civil-Comp Conferences*. Civil-Comp Press. DOI: 10.4203/ccc.1.19.1. URL: https://doi.org/10.4203/ccc.1.19.1.

[18] Zhen Yang et al. “Numerical and experimental study of wheel-rail impact vibration and noise generated at an insulated rail joint”. In: *International Journal of Impact Engineering* 113 (Mar. 2018), pp. 29–39. DOI: 10.1016/j.ijimpeng.2017.11.008. URL: https://doi.org/10.1016/j.ijimpeng.2017.11.008.

[19] Nirmal K Mandal. “Finite element analysis of the mechanical behaviour of insulated rail joints due to impact loadings”. In: *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* 230.3 (Dec. 2014), pp. 759–773. ISSN: 2041-3017. DOI: 10.1177/0954409714561708. URL: http://dx.doi.org/10.1177/0954409714561708.

[20] Nirmal K Mandal, Manicka Dhanasekar, and Yan Quan Sun. “Impact forces at dipped rail joints”. In: *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* 230.1 (June 2014), pp. 271–282. ISSN: 2041-3017. DOI: 10.1177/0954409714537816. URL: http://dx.doi.org/10.1177/0954409714537816.

[21] Meysam Banimahd et al. “Behaviour of train–track interaction in stiffness transitions”. In: *Proceedings of the Institution of Civil Engineers - Transport* 165.3 (Aug. 2012), pp. 205–214. DOI: 10.1680/tran.10.00030. URL: https://doi.org/10.1680/tran.10.00030.

[22] K. Knothe and S. Stichel. *Schienenfahrzeugdynamik*. Berlin: Springer, 2003.

[23] G G Sirata et al. “Study of rail-wheel contact problem by analytical and numerical approaches”. In: *IOP Conference Series: Materials Science and Engineering* 1201.1 (Nov. 2021), p. 012035. ISSN: 1757-899X. DOI: 10.1088/1757-899x/1201/1/012035. URL: http://dx.doi.org/10.1088/1757-899X/1201/1/012035.

[24] Zefeng Wen, Xuesong Jin, and Weihua Zhang. “Contact-impact stress analysis of rail joint region using the dynamic finite element method”. In: *Wear* 258.7–8 (Mar. 2005), pp. 1301–1309. ISSN: 0043-1648. DOI: 10.1016/j.wear.2004.03.040. URL: http://dx.doi.org/10.1016/j.wear.2004.03.040.

[25] Zhen Yang et al. “Numerical study of wheel-rail impact contact solutions at an insulated rail joint”. In: *International Journal of Mechanical Sciences* 138-139 (Apr. 2018), pp. 310–322. DOI: 10.1016/j.ijmecsci.2018.02.025. URL: https://doi.org/10.1016/j.ijmecsci.2018.02.025.

A

Literature study

A.1. Insulated rail joint

Insulated rail joints (IRJs) are vital components in railway systems, playing a critical role in signal control and safety operations. These joints facilitate the connection of two rail segments, with an insulating end post positioned between them to maintain electrical isolation of the adjoining sections. The rail segments are held together by joint bars, which are bolted on either side of the rail. Over the past few decades, two primary design approaches for IRJs have emerged [16]. The first approach incorporates adhesives in addition to mechanical fasteners, leading to what are known as bonded IRJs. This design is the standard in heavy-haul railway applications. The second approach, referred to as non-bonded IRJs, is typically used in scenarios involving lower axle loads. In Figure A.1 a simple schematic side view of an insulated rail joint is presented.

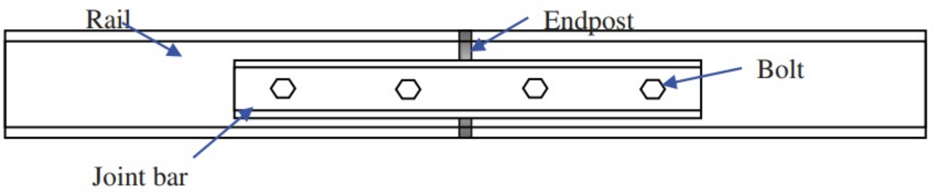


Figure A.1: Schematic side view of inclined rail joint with most important parts [5]

Whether bonded or not bonded, all IRJ's introduce a discontinuity in the rail geometry. These irregularities along longitudinal direction of the track are inherent characteristics of railway transition zones and can be attributed to geometry, configuration of track components, and constitutive properties of the materials [17]. In the case of insulated rail joints, due to the significant stiffness and geometrical discontinuities, it is considered to be one of the weakest parts of railway track structures [18]. The reduced stiffness of components such as fishplates or joint bars, as well as end plate materials like thermoplastic polymers or fiberglass [19], results in greater deflection at the joint compared to continuous rail sections. This local increase in deflection of the rail leads to geometrical spatial discontinuity of the rail also referred to as a dipped joint. In Wu and Thompson [7] the rail joint excitation is examined and it is shown that the gap of a rail joint may be typically between 5-20 mm, with a height difference between both sides of the joint in the range of 0-2 mm. A quadratic formula was developed to estimate the shape of the dipped rail, and its results were compared with experimental measurements, as shown in Figure A.2.

A.1.1. Spatial discontinuities in railway joints

The spatial discontinuity that results from dipping of the rail joint is a source of significant wheel impact [20]. The impact leads high dynamic forces which leads to increased deterioration of the joint.

Jenkins, Lyon and others (British rail) presumably were the first to study the wheel-rail interaction for a dipped joint [10] where they presented formulae for determining P1 and P2 forces. A vertical

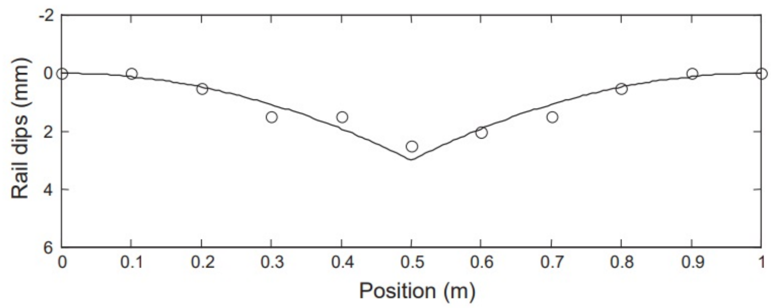


Figure A.2: Dipped rail shape at joint from quadratic function: —— and from measurement: ○ [7]

nonlinear Hertz spring was used to model the wheel-rail contact. A discontinuity in the contact at the dip was not considered.

In Steenbergen [4] a difference is made between modelling a wheel with continuous single-point contact and a wheel with transient double- or multi-point contact which may occur for irregularities with a curvature larger than the wheel circumference. It was shown that the first type of model underestimates the contact forces as the occurring mechanisms are not described correctly.

In [7] Wu and Thompson described the rail joint excitation from measurements and derived a quadratic formula to approximate this excitation. A model was presented which simulates wheel-rail impact in the time and frequency domain. A full wheel-rail interaction was considered instead of a spring model. It was shown that the impact forces increased by 4-6 times compared to the static wheel load depending on the train's velocity. It was also shown that geometry and static wheel load influence the dynamic behaviour of the joint.

In [21] Banimahd and others developed a multi spring model in combination with a train-transition curve was adopted to describe the response of the train in a stiffness transition. The force was calculated using non linear Hertzian contact theory. By modelling the system using a transition curve that excites a spring system the discontinuity was avoided.

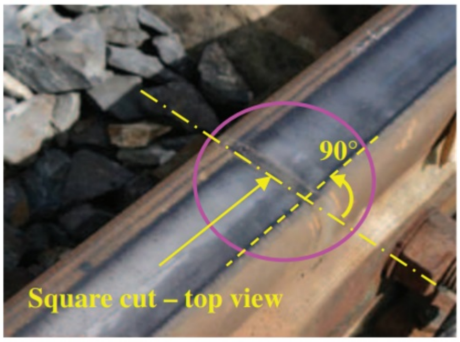
In [20] Mandal, Dhanasekar and Sun reported on wheel impacts caused by permanently dipped rail joints. Instead of a linear or quadratic function for the dipped rail shape they developed a sinusoidal representation. Modelling the spatial discontinuity was avoided by using a spring model which was numerically solved. It was concluded that the impact forces (both P1 and P2) increase with an increase of speed and defect depth (a larger geometrical irregularity).

In [11] Kouroussis and others analyzed the effect of railway track discontinuities on ground vibration generation and propagation. A vehicle/track/soil numerical model was presented. The interaction between vehicle and track was modelled using a multi body model and the ground was modelled using the finite element method. Various formulas for the trajectory of the center of mass of the wheel for common wheel/rail geometric defects were derived such as smooth irregularities, step-up joint, step-down joint, step wise joint and flat spots. The bodies were considered fully rigid.

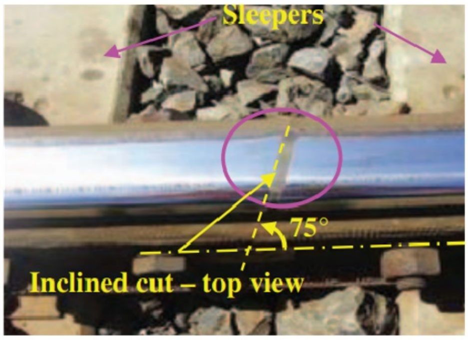
A.2. Inclined insulated rail joints

The inclined insulated rail joint is an insulated rail joint with a different design. Instead of having the end plate material attached perpendicular to the rail the end plate is placed at an angle (the end plate material is still positioned in the vertical plane). The thickness of the end post material has to remain the same to ensure the electrical insulation of the joint. The inclined IRJ is considered as a low-impact joint but there is little to no research evidence to the relative performance of as the vertically cut square and inclined IRJ's [5]. In Figure A.3 a square IRJ and an inclined IRJ at an angle of 75 degrees are shown.

Ataei, Mohammadzadeh, and Miri (2016) conducted a comparative study on the noise, vibration, and vertical displacement of adjacent sleepers for square, 30-degree, and 45-degree cut Insulated Rail Joints (IRJs). This was done through a series of tests on the Tehran-Karaj metro track. Their findings indicated that the acceleration signatures for the 30-degree and 45-degree cut joints were nearly half the magnitude of those observed in square cut joints. In terms of noise performance, the 30-degree cut IRJ showed the best performance, followed by the 45-degree cut, which outperformed the square



(a)



(b)

Figure A.3: Typical designs of square IRJ (a) and inclined IRJ (b) [5]

cut IRJ.

Dhanasekar and Bayissa (2011) discuss the commonly held view that inclined cut Insulated Rail Joints (IRJs) generally perform better than square cut IRJs, but also mention the limited availability of literature on the structural advantages of these two designs. In their study, they analyze the structural response of both IRJ types when subjected to a passing wheel by measuring strain near the rail head. Their findings include strain signatures, revealing that incline IRJs experience higher peak shear strains but lower peak vertical strains compared to square cut IRJs under wheel loading. Furthermore, they propose a hypothesis regarding the comparative performance of the two joint designs. According to this hypothesis, the impact duration on a square cut IRJ is shorter than that of an inclined cut IRJ, leading to higher impact forces on the square cut design.

Megna and Bracciali [3] and Megna Bracciali and Mandal [6] address the failure of traditional insulated rail joints (IRJs), which typically suffer from repeated impacts that cause insulation loss, leading to broken rails and derailments. The papers introduce a new design called ABJ, a joint that replaces the conventional 90° cut with a shallow, tapered cut. This cut is at a 3 degree angle resulting in a very long transition zone between rail head and tail. This design ensures smooth wheel-rail transitions, reducing shock, noise, and ballast deterioration, ultimately extending joint lifespan. The research includes finite element analysis (FEA) and multi-body analysis (MBA) to demonstrate the mechanical advantages of the ABJ over conventional IRJs. It was concluded that dip angle can be eliminated by a long transition zone and attaching a reinforcing joint cover.

A.3. Wheel-Rail Contact Interface

The wheel-rail interface is an important element of railway dynamics. According to Knothe et al. [22] it has three fundamental tasks each associated to a specific contact force component:

1. Load bearing to the vertical force
2. Guiding to the lateral force
3. Traction to the longitudinal force

The contact between wheel and rail in normal direction is very stiff. The deformations are in order of magnitude 10ths of millimeters and the contact area has an approximate size of 1 to 2 cm^2 even when the transmitted vertical loads are very high, in the order of magnitude of 10 tons. This small contact area in combination with large loads results in large normal stresses [13]. In the tangential direction the contact behaviour is ruled by friction and exhibits non-linear behaviour and saturation. These characteristics already express the complexity of the wheel-rail contact problem. In Figure A.4 the various stresses in wheel-rail contact patch are shown as an example of the complexity.

A.3.1. Analytical modelling

Normal contact modelling

When two surfaces come into contact the point or line of contact can become a location of high stress values. Stress, defined as force per unit area, suggests that any load applied to a point contact would

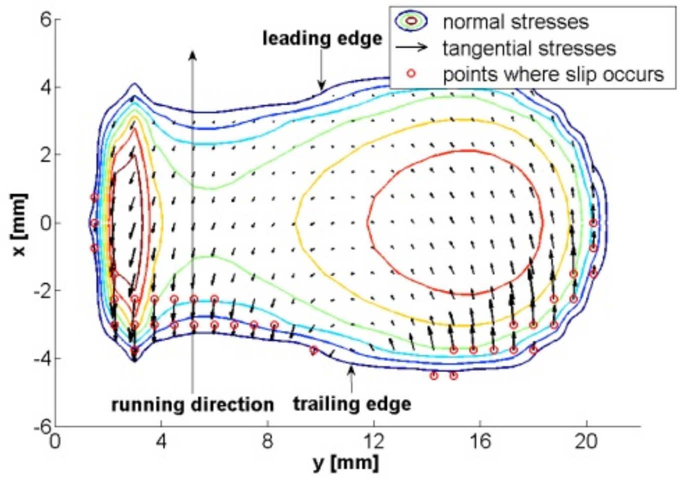


Figure A.4: Example of contact stresses. Calculated according to Kalker's non-elleptical rolling theory [13]

theoretically result in infinite stresses at that point. However, since idealized point or line contacts do not exist in reality, the actual contact area will always have a finite size. When this contact area approaches an IRJ part of the contact patch will lose contact with the rail. This results in a complex interaction between the maximum stresses and the contact area of the contact patch. The behavior of the contact area and the distribution of stress within it have been extensively studied. However, due to the complexity of the phenomena involved, many analytical models rely on approximations or include unknown empirical constants, and do not include imperfections in the rail geometry. In his classical paper Hertz [15] was the first to develop an analytical method to describe the shape of the contact, deformation of the materials and the normal pressures in the contact. The theory was derived based on the fact that elastic deformation between the bodies takes place to provide a larger contact area which results in finite stresses. To do this, two new parameters were introduced, the contact modulus which includes the materials of both contacting bodies and the equivalent radius that includes the geometry of the contacting surfaces. In [23] analytical formulas are given for wheel-rail contact specifically. The contact modulus is defined as:

$$\frac{1}{E^*} = \frac{1 - v_w^2}{E_w} + \frac{1 - v_r^2}{E_r}$$

Where v and E denote the Poisson ratio and elastic modulus respectively. The subscripts w and r respectively denote wheel and rail. The equivalent radii are defined (the conicity of the wheel is ignored since they are relatively small) as:

$$\frac{1}{R_x} = \frac{1}{R_x^{(w)}} = 2A, \quad \frac{1}{R_y} = \frac{1}{R_y^{(w)}} + \frac{1}{R_y^{(r)}} = 2B$$

The contact patch is shaped like an ellipse with major axis a and minor axis $b$and is dependent on the shape and materials of the geometries in contact:

$$a = m\left(\frac{3F_n}{4E^*}\frac{1}{A+B}\right)^{\frac{1}{3}}$$

$$b = n\left(\frac{3F_n}{4E^*}\frac{1}{A+B}\right)^{\frac{1}{3}}$$

In these formulas m and n are coefficients for semi-axis lengths of the ellipse, which were tabulated by Hertz. In Figure A.5 the elliptical contact patch is shown in the wheel-rail interaction. The contact

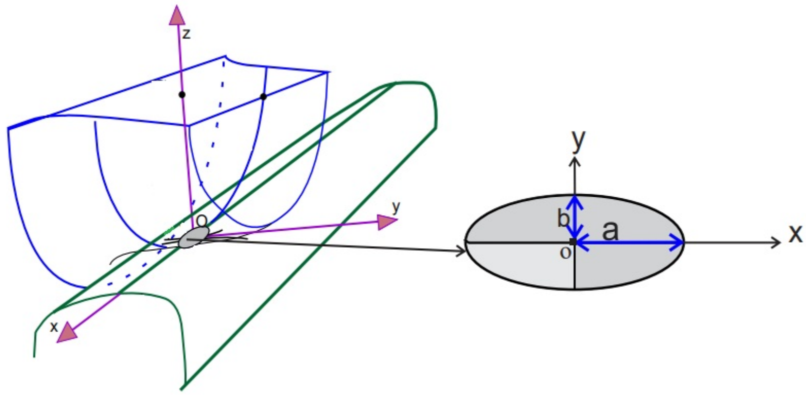


Figure A.5: Elliptical contact patch in wheel-rail interaction

pressure in this contact patch can be expressed as:

$$p = p_0\sqrt{1 - (\frac{x}{a})^2 - (\frac{y}{b})^2}$$

where p_0 is the maximum contact pressure, located at the center of the ellipse and is expressed as:

$$p_0 = \frac{3}{2}\frac{N}{\pi ab}$$

A.4. Numerical modelling

Due to the complexity of the wheel-rail interaction at the contact interface, particularly near geometric rail imperfections, the Finite Element Method (FEM) is frequently employed for analysis. However, FEM-based approaches are computationally intensive and time-consuming. Therefore, when utilizing FEM to simulate wheel-rail contact, it is essential to thoroughly understand the underlying mathematical models and their impact on the performance and accuracy of the simulations.

Wen et al. [24] uses FEM simulations to analyze the dynamic contact-impact behaviour a train wheel and the rail joint region using ANSYS/LS-DYNA software packages. Imperfections like dip angle, rail height mismatch and rail gap are examined. The key focus is on investigating the axle load and train speed on the resulting contact forces, stresses, strains in the rail head of the joint. It is concluded that axle load has a greater impact compared to train velocity on the contact stresses near the rail joint.

In [19] Mandal presents a comprehensive study the dynamic loading effects on IRJ's. The study focuses on the influence of three different end post materials: fiberglass, nylon 66 and polytpolytetrafluoroethylene (PTFE) and their mechanical performance is tested over 2000 cycles of wheel load. The results show that fiberglass performs best in reduction of damage compared to the other materials. A modified Hertzian contact pressure distribution is adopted. This paper shift the focus from mere contact pressure analysis to examining material degradation mechanisms.

Yang et al. [25] deployed a complex FEM model to analyze the transient contact behaviour of the wheel-rail system in the vicinity of an IRJ. Two rail geometries were considered: nominal and measured. For the material models, also two different types were adopted: elastic and elasto-plastic. The contact patch area, contact stress and wave propagation were tracked during the wheel passage over the IRJ. The model successfully models dynamic contact impact-stresses and it was shown that the different material models and rail head geometries play important roles in the contact behaviour during wheel-IRJ impact.

A.5. Concluding remarks

This literature review has highlighted key aspects of modeling insulated rail joints (IRJs), with a focus on the spatial discontinuities and wheel-rail contact interface. One observation is that many existing analytical models simplify the interaction of wheel-rail contact near a joint by using spring systems, which leads to an underestimation of impact forces. This approach, while being more convenient, does not fully capture the complex interactions that occur during wheel-rail contact, especially when a rail joint introduces a geometric irregularity.

Another finding is the lack of literature specifically addressing inclined insulated rail joints. Despite some studies suggesting that inclined joints may offer better performance compared to traditional square-cut IRJs, there is limited research on the structural advantages of these designs. This highlights the need for more comprehensive studies focusing on the theoretical analysis of inclined insulated rail joints.

Finally, numerical modeling has proven to be an effective method for simulating wheel-rail contact pressure. The downside is that these models are highly complex and require a lot of time and effort to develop and run.

While the current state of literature offers good insight into the behavior of insulated rail joints considering wheel-rail interaction and contact stresses, there is not much known about inclined insulated rail joints in particular. Further research in into the working of inclined insulated rail joints may contribute to more reliable and safe infrastructure.

B

Maple Calculations

```
> restart;
```

This file sets up the equations for the joint at a non-orthogonal junction angle and the disc in the local x'y'-plane.
The plots of the joint are upside down since in the rest of the analysis the negative axis is up. Rotating the axis
180 degrees around the x-axis will result in the right configuration. Due to limitations in Maple this is not done in this file.

Import the necessary plot tools:

```
> with(plots):
> with(plottools):
```

Define height of joint in local axis as function of x z(x):

```
> z:= -sin(beta/2) * x + z0;
```

$$z := -\sin\left(\frac{\beta}{2}\right) x + z0 \quad \textbf{(1)}$$

Assign known points in local axis to solve z0:

```
> x:= sin(beta/2)*R:
> eq:= z = -(R-R*cos(beta/2)):
> z0:= solve(eq, z0);
```

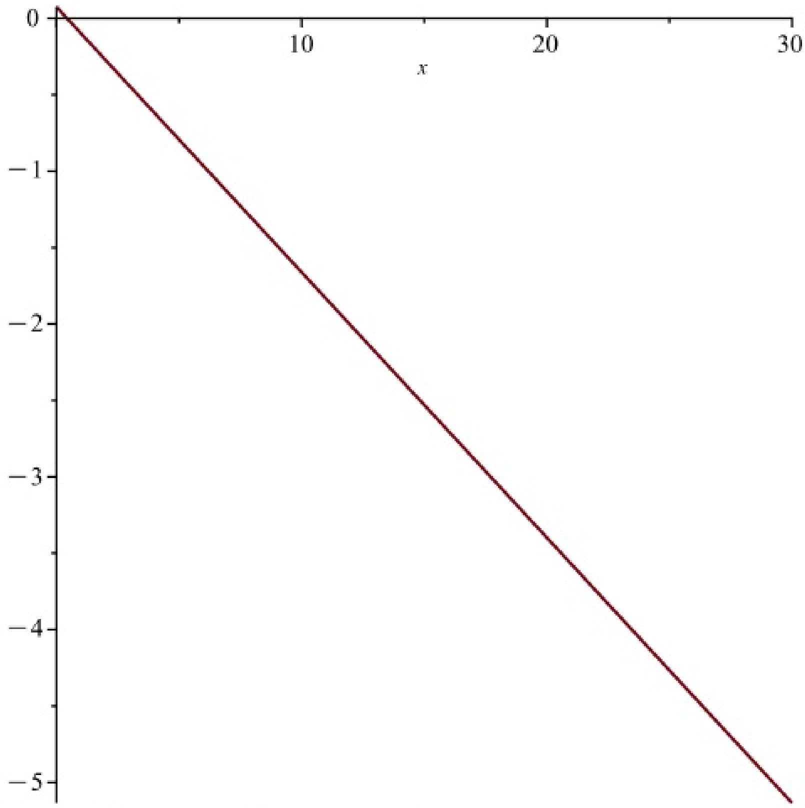
$$z0 := \sin\left(\frac{\beta}{2}\right)^2 R + R\cos\left(\frac{\beta}{2}\right) - R \quad \textbf{(2)}$$

Set x as variable again and plot the vertical height of the joint z as a function of x (z(x)). Parameters are chosen to generate easily readable figures:

```
> x:='x':
> z;
```

$$-\sin\left(\frac{\beta}{2}\right) x + \sin\left(\frac{\beta}{2}\right)^2 R + R\cos\left(\frac{\beta}{2}\right) - R \quad \textbf{(3)}$$

```
> R:= 5: beta:= 20*Pi/180: theta:= 30*Pi/180:
> p:= plot(z, x=0..30); R:= 'R': beta:= 'beta': theta:= 'theta':
```

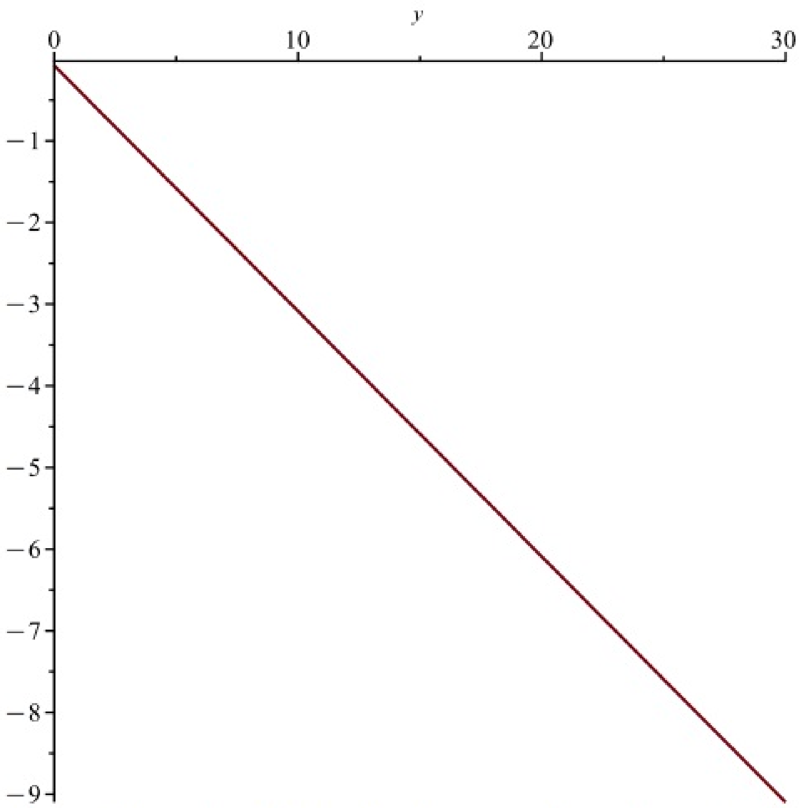


Define the height of the joint z as a function of y (z(y)):

```
> x:= sin(beta/2)*R + y*cot(theta):
> z;
```

$$-\sin\left(\frac{\beta}{2}\right)\left(\sin\left(\frac{\beta}{2}\right)R+y\cot(\theta)\right)+\sin\left(\frac{\beta}{2}\right)^2R+R\cos\left(\frac{\beta}{2}\right)-R \tag{4}$$

```
> R:= 5: beta:= 20*Pi/180: theta:= 30*Pi/180:
> plot(z, y=0..30); R:= 'R': beta:= 'beta': theta:= 'theta':
```



Define the coordinates of a point located on the joint that is dependent on the eccentric wheel position y (P(x_p(y), z_p(y)):

```
> x_p:= x;
```

$$x\_p := \sin\left(\frac{\beta}{2}\right) R + y \cot(\theta) \quad \textbf{(5)}$$

```
> z_p := z;
```

$$z\_p := -\sin\left(\frac{\beta}{2}\right) \left(\sin\left(\frac{\beta}{2}\right) R + y \cot(\theta)\right) + \sin\left(\frac{\beta}{2}\right)^2 R + R \cos\left(\frac{\beta}{2}\right) - R \quad \textbf{(6)}$$

Define the circle function to represent the infinitely thin disc:

```
> C:= (x_c - a)^2 + (z_c - b)^2 - R^2 = 0;
```

$$C := (x\_c - a)^2 + (z\_c - b)^2 - R^2 = 0 \quad \textbf{(7)}$$

The vertical shift of the wheel is related to the horizontal shift, eliminating one unknown:

```
> b:= a*sin(beta/2) - R;
```

$$b := a \sin\left(\frac{\beta}{2}\right) - R \quad \textbf{(8)}$$

Insert the point inside the circle equation and solve for a to find the position of the disc during contact:

```
> x_c:= x_p;
```

$$x\_c := \sin\left(\frac{\beta}{2}\right) R + y \cot(\theta) \qquad \textbf{(9)}$$

```
> z_c := z_p;
```

$$z\_c := -\sin\left(\frac{\beta}{2}\right)\left(\sin\left(\frac{\beta}{2}\right) R + y\cot(\theta)\right) + \sin\left(\frac{\beta}{2}\right)^2 R + R\cos\left(\frac{\beta}{2}\right) - R \qquad \textbf{(10)}$$

```
> C;
```

$$\left(\sin\left(\frac{\beta}{2}\right) R + y\cot(\theta) - a\right)^2 + \left(-\sin\left(\frac{\beta}{2}\right)\left(\sin\left(\frac{\beta}{2}\right) R + y\cot(\theta)\right) + \sin\left(\frac{\beta}{2}\right)^2 R + R\cos\left(\frac{\beta}{2}\right) - a\sin\left(\frac{\beta}{2}\right)\right)^2 - R^2 = 0 \qquad \textbf{(11)}$$

```
> a:= solve(C, a)[1];
```

$$a := -\frac{1}{\left(\cos\left(\frac{\beta}{2}\right)^2 - 2\right)\tan(\theta)}\Bigg(\sin\left(\frac{\beta}{2}\right)\cos\left(\frac{\beta}{2}\right)\tan(\theta)\, R + \sin\left(\frac{\beta}{2}\right)\tan(\theta)\, R$$

$$+ \cos\left(\frac{\beta}{2}\right)^2 y$$

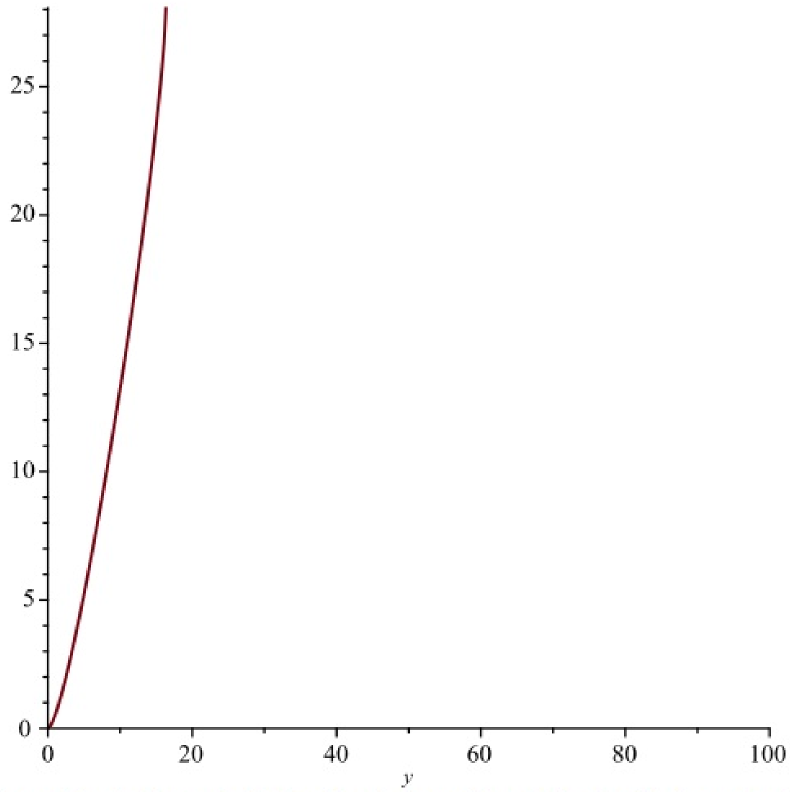
$$- \left(\sin\left(\frac{\beta}{2}\right)^2 \cos\left(\frac{\beta}{2}\right)^2 \tan(\theta)^2 R^2 + 2\sin\left(\frac{\beta}{2}\right)^2 \cos\left(\frac{\beta}{2}\right)\tan(\theta)^2 R^2 + \sin\left(\frac{\beta}{2}\right)^2 \tan(\theta)^2 R^2 + 4\sin\right.$$

$$\left. + 4\cos\left(\frac{\beta}{2}\right)\sin\left(\frac{\beta}{2}\right) R\, y\tan(\theta) - 4\sin\left(\frac{\beta}{2}\right) R\, y\tan(\theta) + 4\cos\left(\frac{\beta}{2}\right)^2 y^2 - 4y^2\right)^{1/2}\Bigg)$$

```
> R:= 5: beta:= 20*Pi/180: theta:= 30*Pi/180:
```

Plot the function of a(y). A qudratic function is expected to the reducing horizontal projection of the circle for larger y:

```
> plot(a, y=0..100);
```



A random disc position is chosen (y=30 [mm]) and the position of the wheel is determined using derived equations. Disc and wheel are plotted and the results match.

```
> x_c:='x_c': z_c:='z_c': R:=500: y:=30:
> evalf(a);
```

$$6.654225166 \tag{13}$$

```
> evalf(b);
```

$$-498.8445059 \tag{14}$$

```
> evalf(C);
```

$$(x\_c - 6.654225166)^2 + (z\_c + 498.8445059)^2 - 250000. = 0. \tag{15}$$

```
> a:= 6.654225166: b:= 498.8445059:
> z_c:= - solve(C, z_c)[2];
```

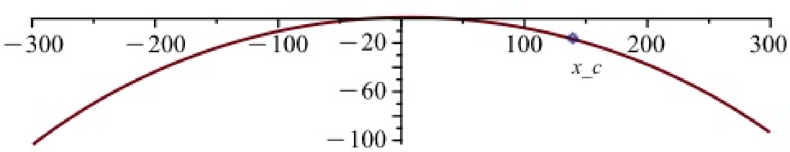
$$z\_c := -498.8445059 + 2.000000000 \times 10^{-9} \sqrt{-2.500000000 \times 10^{17}\, x\_c^2 + 3.327112583 \times 10^{18}\, x\_c + 6.248893032 \times 10^{22}} \tag{16}$$

```
> evalf(z_c);
```

$$-498.8445059 + 2.000000000 \times 10^{-9} \sqrt{-2.500000000 \times 10^{17}\, x\_c^2 + 3.327112583 \times 10^{18}\, x\_c + 6.248893032 \times 10^{22}} \tag{17}$$

```
> p1:=plot(z_c, x_c = -300..300):
> p2:=plot(([[x_p, z_p]], style=POINT)):
```

```
> display(p1, p2);
```



C

Global Wheel-Rail Interaction Plotting and Visualization Code

```
import matplotlib.pyplot as plt
import numpy as np


# Define all functions

# Function for the x_location of the cut-off in x'y' (local coordinate system)
def x_cutoff_loc(y_loc, theta, beta, R):
    x = y_loc * (1 / np.tan(theta)) + R * np.sin(beta / 2)
    return x


# Define the vertical cut-off in x-y plane:
def x_cutoff_glob(y_glob, theta):
    x = y_glob * (1 / np.tan(theta))
    return x


# Transform the cutoff y-coordinates from local axis to global axis:
# Input is a list of y' (local) values that will be transformed to y (global) values:
def transform_to_glob_y(ys, theta, beta, R):
    ys_global = []

    for y in ys:
        ys_global.append(y + R * np.sin(beta / 2) * np.tan(theta))

    return ys_global


# Determine the transition value y_t in global coordinate system
def yt(theta, beta, R):
    y_t = R * np.sin(beta / 2) * np.tan(theta)
    return y_t


```

```python
# Determine the wheel shift when 2-point contact with the cutoff occurs:
def a(y, theta, beta, R):
    a = (-np.tan(theta) * np.sin(beta / 2) * np.cos(beta / 2) * R - np.tan(theta) * np.sin(beta / 2) * R - np.cos(
        beta / 2) ** 2 * y + np.sqrt(np.tan(theta) ** 2 *
                                     np.sin(beta / 2) ** 2 * np.cos(beta / 2) ** 2 * R ** 2 + 2 * np.tan(
        theta) ** 2 * np.sin(beta / 2) ** 2 * np.cos(beta / 2) * R ** 2 + np.tan(theta) ** 2 * np.sin(
        beta / 2) ** 2 * R ** 2
                                     + 4 * np.tan(theta) * np.sin(beta / 2) * np.cos(
        beta / 2) ** 2 * R * y + 4 * np.cos(beta / 2) * np.sin(beta / 2) * R * y * np.tan(theta) - 4 * np.sin(
        beta / 2) * R * y * np.tan(theta)
                                     + 4 * np.cos(beta / 2) ** 2 * y ** 2 - 4 * y ** 2)) / (
                        (np.cos(beta / 2) ** 2 - 2) * np.tan(theta))
    return a


# Determine the horizontal distance between the vertical centerline of the wheel and contact point A
def get_xA(y_loc, theta, beta, R):
    xA = x_cutoff_loc(y_loc, theta, beta, R) - a(y_loc, theta, beta, R)
    return xA


# Define the position of the vertical center line of the wheel in x'y' plane:
def xW(y, theta, beta, R):
    xW = x_cutoff_loc(y, theta, beta, R) - a(y, theta, beta, R) - R * np.sin(beta / 2)
    return xW


# Define the exit angle of the trajectory right after two-point contact:
def get_alpha(xA, R):
    alpha = np.arcsin(xA / R)
    return alpha


# Define the trajectory of contact scenario 1:
def trajectory_1(ts, beta):
    zs = []
    for t in ts:
        if t <= 0:
            zs.append(-np.sin(beta / 2) * v * t)
        if t > 0:
            zs.append(np.sin(beta / 2) * v * t)
    return zs


# Define the trajectory of contact scenario 2:
def trajectory_2(ts, y_loc, theta, beta, R, v):
    xA = get_xA(y_loc, theta, beta, R)
```

```
    alpha = get_alpha(xA, R)

    zs = []

    for t in ts:
        if v * t <= 0:
            zs.append(-np.sin(beta / 2) * v * t)
        if v * t > 0:
            zs.append(np.sin(alpha) * v * t)
    return zs


# Define the vertical velocity of contact scenario 1:
def vertical_velocity_1(ts, beta, v):
    vzs1_before = []
    vzs1_after = []
    for t in ts:
        if v * t <= 0:
            vzs1_before.append(-np.sin(beta / 2) * v)
        if v * t > 0:
            vzs1_after.append(np.sin(beta / 2) * v)
    return vzs1_before, vzs1_after


# Define the vertical velocity of contact scenario 2:
def vertical_velocity_2(ts, y_loc, theta, beta, R, v):
    xA = get_xA(y_loc, theta, beta, R)
    alpha = get_alpha(xA, R)

    vzs2_before = []
    vzs2_after = []
    for t in ts:
        if v * t <= 0:
            vzs2_before.append(-np.sin(beta / 2) * v)
        if v * t > 0:
            vzs2_after.append(np.sin(alpha) * v)
    return vzs2_before, vzs2_after


# Define vertical momentum of contact scenario 1:
def vertical_momentum_1(ts, beta, M, v):
    vs_before = vertical_velocity_1(ts, beta, v)[0]
    vs_after = vertical_velocity_1(ts, beta, v)[1]
    Ms_before = []
    Ms_after = []
    k = 0
    m = 0
    for t in ts:
        if t <= 0:
            Ms_before.append(vs_before[k] * M)
            k += 1
        if t > 0:
            Ms_after.append(vs_after[m] * M)
            m += 1
    return Ms_before, Ms_after
```

```

# Define vertical momentum of contact scenario 2:
def vertical_momentum_2(ts, y, theta, beta, R, M, v):
    vs0 = vertical_velocity_2(ts, y, theta, beta, R, v)[0]
    vs1 = vertical_velocity_2(ts, y, theta, beta, R, v)[1]
    Ms0 = []
    Ms1 = []
    k = 0
    m = 0
    for t in ts:
        if t <= 0:
            Ms0.append(vs0[k] * M)
            k += 1
        if t > 0:
            Ms1.append(vs1[m] * M)
            m += 1
    return Ms0, Ms1


# Define the vertical impulse of contact scenario 1:
def vertica_impulse_1(ts, beta, M, v):
    Ms_before = vertical_momentum_1(ts, beta, M, v)[0]
    Ms_after = vertical_momentum_1(ts, beta, M, v)[1]
    Ms_before.extend(Ms_after)
    Ps = [0]

    for i in range(len(Ms_before) - 1):
        Ps.append(Ms_before[i + 1] - Ms_before[i])
    return Ps

# Define vertical impulse of contact scenario 2:
def vertical_impulse_2(ts, y, theta, beta, R, M, v):
    Ms = vertical_momentum_2(ts, y, theta, beta, R, M, v)[0]
    Ms1 = vertical_momentum_2(ts, y, theta, beta, R, M, v)[1]
    Ms.extend(Ms1)
    Ps = [0]

    for i in range(len(Ms) - 1):
        Ps.append(Ms[i + 1] - Ms[i])
    return Ps


# Define the input parameters for the model
y_glob = 15  # Transverse position of the wheel on the track [mm]
thetas =  [30 * np.pi / 180, 45 * np.pi / 180, 60 * np.pi / 180]  #
    Selected incline angles [degrees]
beta = 0.03 # Dip angle [rad]
R = 475 # Radius of the wheel [mm]
M = 750 # Mass of the wheel [kg]
v = 50 # Train velocity [m/s]
bt = 20 # Width of the track [mm]

# Set up plotting parameters:
ts = np.linspace((-10/1000) / v, (12/1000 / v), 1000)
betas = np.linspace(0, 0.05, 100)

```

```python
# Plot the transition values yt as a function of beta for 3 selected values of theta:
yts = [[], [], []]

for i in range(len(thetas)):
    for beta in betas:
        yts[i].append(yt(thetas[i], beta, R))

# Create inverse values of each list to plot on the mirrored side:
yts_negative = []
for sub_list in yts:
    yts_negative.append([value * -1 for value in sub_list])

# Create the plot for yt
plt.figure()
plt.plot(betas, yts[0], 'g', label=r'Transition value $y_t$ for $\theta=30^\circ$')
plt.plot(betas, yts[1], 'r', linestyle='dotted', label=r'Transition value $y_t$ for $\theta=45^\circ$')
plt.plot(betas, yts[2], 'b', linestyle='dashdot', label=r'Transition value $y_t$ for $\theta=60^\circ$')
plt.plot(betas, yts_negative[0], 'g')
plt.plot(betas, yts_negative[1], 'r', linestyle='dotted')
plt.plot(betas, yts_negative[2], 'b', linestyle='dashdot')
plt.axhline(y = -20, color = 'k', linestyle = '-', linewidth=2)
plt.axhline(y = 20, color = 'k', linestyle = '-',  linewidth=2)
plt.text(0.0532, -20, r'$-b_t$', va='center', ha='right', fontsize=12)
plt.text(0.053, 20, r'$b_t$', va='center', ha='right', fontsize=12)
plt.text(0.027, -15, 'Contact Scenario 2', fontsize=12, color='b', ha='center', va='bottom')
plt.text(0.047, -13, 'Contact Scenario 1', fontsize=12, color='b', ha='center', va='bottom')
plt.xlabel('Dip angle $\\beta$ [rad]')
plt.ylabel('Lateral position on track $y$ [mm]')
plt.legend(loc='best')
plt.grid(True)
plt.gca().invert_yaxis()
plt.margins(x=0)
plt.margins(y=0)

beta = 0.03

# Determine the y_loc and horizontal shift for all three values of theta:
yt_thetas = [yt(theta, beta, R) for theta in thetas]
y_locs = [y_glob - yt for yt in yt_thetas]
a_values = [a(y_locs[i], thetas[i], beta, R)/1000 for i in range(len(thetas))]

# Compute the trajectories for contact scenario 1:
zs_1 = trajectory_1(ts, beta)

# Compute the trajectories for contact scenario 2:
zs_2 = []
for i in range(len(thetas)):
    zs_2.append(trajectory_2(ts, y_locs[i], thetas[i], beta, R, v))

```

```
# Determine the vertical velocities for contact scenario 1 before and after impact for each theta:
vzs1_before = vertical_velocity_1(ts, beta, v)[0]
vzs1_after = vertical_velocity_1(ts, beta, v)[1]

# Determine the velocities for contact scenario 2 before and after impact for each theta:
vzs2_before_30 = vertical_velocity_2(ts, y_locs[0], thetas[0], beta, R, v)[0]
vzs2_after_30 = vertical_velocity_2(ts, y_locs[0], thetas[0], beta, R, v)[1]
vzs2_before_45 = vertical_velocity_2(ts, y_locs[1], thetas[1], beta, R, v)[0]
vzs2_after_45 = vertical_velocity_2(ts, y_locs[1], thetas[1], beta, R, v)[1]
vzs2_before_60 = vertical_velocity_2(ts, y_locs[2], thetas[2], beta, R, v)[0]
vzs2_after_60 = vertical_velocity_2(ts, y_locs[2], thetas[2], beta, R, v)[1]

# Determine the vertical momentum before and after impact for contact scenario 1:
Mzs1_before = vertical_momentum_1(ts, beta, M, v)[0]
Mzs1_after = vertical_momentum_1(ts, beta, M, v)[1]

# Determine the vertical momentum before and after impact for contact scenario 2:
Mzs2_before_30 = vertical_momentum_2(ts, y_locs[0], thetas[0], beta, R, M, v)[0]
Mzs2_after_30 = vertical_momentum_2(ts, y_locs[0], thetas[0], beta, R, M, v)[1]
Mzs2_before_45 = vertical_momentum_2(ts, y_locs[1], thetas[1], beta, R, M, v)[0]
Mzs2_after_45 = vertical_momentum_2(ts, y_locs[1], thetas[1], beta, R, M, v)[1]
Mzs2_before_60 = vertical_momentum_2(ts, y_locs[2], thetas[2], beta, R, M, v)[0]
Mzs2_after_60 = vertical_momentum_2(ts, y_locs[2], thetas[2], beta, R, M, v)[1]

# Determine the vertical impulse for contact scenario 1:
Ps1 = vertica_impulse_1(ts, beta, M, v)

# Determine the vertical impulse for contact cenario 2:
Ps2_30 = vertical_impulse_2(ts, y_locs[0], thetas[0], beta, R, M, v)
Ps2_45 = vertical_impulse_2(ts, y_locs[1], thetas[1], beta, R, M, v)
Ps2_60 = vertical_impulse_2(ts, y_locs[2], thetas[2], beta, R, M, v)

# Split the t-values to before and after impact:
t_before = ts[:len(vzs2_before_30)]
t_after = ts[len(vzs2_before_30):]

# Create a figure with 4 rows and 2 columns (8 subplots in total),
# sharing x-axis in columns and y-axis in rows
fig, axs = plt.subplots(4, 2, figsize=(10, 12), sharex='col', sharey='row')
```

```python

# ---- Column 1: Contact Scenario 1 ----

# Plot 1: Trajectory contact scenario 1
axs[0, 0].plot(ts, zs_1, 'k', label='$0 < \\theta \\leq 90$ and $-y_t < y_w \\leq y_t$', linestyle='dashed')
axs[0, 0].set_ylabel("Trajectory $z$ [$m$]")
#axs[0, 0].set_title('a1). Trajectory (Scenario 1)')
axs[0, 0].set_title('Contact Scenario 1', fontsize=16)
axs[0, 0].ticklabel_format(style='sci', axis='x', scilimits=(0,0))
axs[0, 0].ticklabel_format(style='sci', axis='y', scilimits=(0,0))
axs[0, 0].legend()
axs[0, 0].grid(True)
axs[0, 0].margins(x=0)

# Plot 2: Vertical velocity contact scenario 1
axs[1, 0].plot(t_before, vzs1_before, 'k', linestyle='dashed', label='$0 < \\theta \\leq 90$ and $-y_t < y_w \\leq y_t$')
axs[1, 0].plot(t_after, vzs1_after, 'k', linestyle='dashed')
axs[1, 0].set_ylabel("Vertical velocity $v_z$ [$m/s$]")
#axs[1, 0].set_title('b1). Vertical velocity (Scenario 1)')
axs[1, 0].legend()
axs[1, 0].grid(True)
axs[0, 0].margins(x=0)

# Plot 3: Vertical momentum contact scenario 1
axs[2, 0].plot(t_before, Mzs1_before, 'k', label='$0 < \\theta \\leq 90$ and $-y_t < y_w \\leq y_t$', linestyle='dashed')
axs[2, 0].plot(t_after, Mzs1_after, 'k', linestyle='dashed')
axs[2, 0].set_ylabel("Vertical momentum $M_z$ [$N \\cdot s$]")
#axs[2, 0].set_title('c1). Vertical momentum (Scenario 1)')
axs[2, 0].legend()
axs[2, 0].grid(True)
axs[2, 0].margins(x=0)

# Plot 4: Vertical impulse contact scenario 1
axs[3, 0].plot(ts, Ps1, 'k', label='$0 < \\theta \\leq 90$ and $-y_t < y_w \\leq y_t$', linestyle='dashed')
axs[3, 0].set_xlabel("Time [$s$]")
axs[3, 0].set_ylabel("Vertical impulse $J_z$ [$N \\cdot s$]")
#axs[3, 0].set_title('d1). Vertical impulse (Scenario 1)')
axs[3, 0].axhline(y = max(Ps1), color = 'k', linestyle = ':', linewidth=1)
axs[3, 0].legend()
axs[3, 0].grid(True)
axs[3, 0].margins(x=0)

# ---- Column 2: Contact Scenario 2 ----

labels = ['$\\theta$ = 30 [deg]', '$\\theta$ = 45 [deg]', '$\\theta$ = 60 [deg]']
linestyles = ['solid', 'dotted', 'dashdot']

# Define offsets to plot the wheel position in absolute point in time:
offset_30 = a_values[0] / v
offset_45 = a_values[1] / v
offset_60 = a_values[2] / v
```

```python
offsets = [offset_30, offset_45, offset_60]
colors = ['r', 'g', 'b']

# Plot 1: Trajectory contact scenario 2
for i in range(len(thetas)):
    axs[0, 1].plot(ts + offsets[i], zs_2[i], color=colors[i], linestyle=linestyles[i], label=labels[i])
#axs[0, 1].set_title('a2). Trajectory (Scenario 2)')
axs[0, 1].set_title('Contact Scenario 2', fontsize=16)
axs[0, 1].ticklabel_format(style='sci', axis='x', scilimits=(0,0))
axs[0, 1].legend()
axs[0, 1].grid(True)

# Plot 2: Vertical velocity contact scenario 2
axs[1, 1].plot(t_before + offset_30, vzs2_before_30, 'r', label='$\\theta$ = 30 [deg]')
axs[1, 1].plot(t_after + offset_30, vzs2_after_30, 'r')
axs[1, 1].plot(t_before + offset_45, vzs2_before_45, 'g', linestyle='dotted', label='$\\theta$ = 45 [deg]')
axs[1, 1].plot(t_after + offset_45, vzs2_after_45, 'g', linestyle='dotted')
axs[1, 1].plot(t_before + offset_60, vzs2_before_60, 'b', linestyle='dashdot', label='$\\theta$ = 60 [deg]')
axs[1, 1].plot(t_after + offset_60, vzs2_after_60, 'b', linestyle='dashdot')
#axs[1, 1].set_title('b2). Vertical velocity (Scenario 2)')
axs[1, 1].legend()
axs[1, 1].grid(True)

# Plot 3: Vertical momentum contact scenario 2
axs[2, 1].plot(t_before + offset_30, Mzs2_before_30, 'r', label='$\\theta$ = 30 [deg]')
axs[2, 1].plot(t_after + offset_30, Mzs2_after_30, 'r')
axs[2, 1].plot(t_before + offset_45, Mzs2_before_45, 'g', linestyle='dotted', label='$\\theta$ = 45 [deg]')
axs[2, 1].plot(t_after + offset_45, Mzs2_after_45, 'g', linestyle='dotted')
axs[2, 1].plot(t_before + offset_60, Mzs2_before_60, 'b', linestyle='dashdot', label='$\\theta$ = 60 [deg]')
axs[2, 1].plot(t_after + offset_60, Mzs2_after_60, 'b', linestyle='dashdot')
#axs[2, 1].set_title('c2). Vertical momentum (Scenario 2)')
axs[2, 1].legend()
axs[2, 1].grid(True)

# Plot 4: Vertical impulse contact scenario 2
axs[3, 1].plot(ts + offset_30, Ps2_30, 'r', label='$\\theta$ = 30 [deg]')
axs[3, 1].plot(ts + offset_45, Ps2_45, 'g', linestyle='dotted', label='$\\theta$ = 45 [deg]')
axs[3, 1].plot(ts + offset_60, Ps2_60, 'b', linestyle='dashdot', label='$\\theta$ = 60 [deg]')
axs[3, 1].axhline(y = max(Ps1), color = 'k', linestyle = ':', linewidth=1)
axs[3, 1].set_xlabel("Time [$s$]")
#axs[3, 1].set_title('d2). Vertical impulse (Scenario 2)')
axs[3, 1].legend()
axs[3, 1].grid(True)
```

```


# Defining custom 'xlim' and 'ylim' values.
custom_xlim = ((-4 / 1000) / v, (12 / 1000) / v)

# Setting the values for all axes.
plt.setp(axs, xlim=custom_xlim)

#plt.xlim((-4 / 1000) / v, (12 / 1000) / v)

# Adjust layout for better spacing
plt.tight_layout()

# -------------------------------------------------------------
# Set up the plot for the topviews:
# Define the rail in global coordinate system:
thetas = [30 * np.pi / 180, 45 * np.pi / 180, 60 * np.pi / 180, 90 * np.pi
     / 180] # Add the 90 degree angle to the list
ys = np.linspace(0, bt, 100)
ys_neg = np.linspace(0, -bt, 100)
beta = 0.03

xs_cut_30 = [x_cutoff_glob(y, thetas[0]) for y in ys]
xs_cut_30_neg = [value * -1 for value in xs_cut_30]
xs_cut_45 = [x_cutoff_glob(y, thetas[1]) for y in ys]
xs_cut_45_neg = [value * -1 for value in xs_cut_45]
xs_cut_60 = [x_cutoff_glob(y, thetas[2]) for y in ys]
xs_cut_60_neg = [value * -1 for value in xs_cut_60]
xs_cut_90 = [x_cutoff_glob(y, thetas[3]) for y in ys]
xs_cut_90_neg = [value * -1 for value in xs_cut_90]

# Define the transition values y_t for each theta:
yt_30 = yt(thetas[0], beta, R)
yt_30_neg = -yt_30
yt_45 = yt(thetas[1], beta, R)
yt_45_neg = -yt_45
yt_60 = yt(thetas[2], beta, R)
yt_60_neg = -yt_60
yt_90 = yt(thetas[3], beta, R)
yt_90_neg = -yt_90
# Define the position of the contact points and center of the wheel for
    each theta:
xB = np.sin(beta / 2) * R
xB_neg = -xB

xAs_30, xWs_30, xBs_30 = [], [], []
xAs_45, xWs_45, xBs_45 = [], [], []
xAs_60, xWs_60, xBs_60 = [], [], []
xAs_90, xWs_90, xBs_90 = [], [], []

# Loop through for theta 30 degrees
for y in ys:
    if y <= yt_30:
        xAs_30.append(xB)
        xWs_30.append(0)
        xBs_30.append(-xB)
```

```
    else:
        xAs_30.append(x_cutoff_loc(y - yt_30, thetas[0], beta, R))
        xWs_30.append(xW(y - yt_30, thetas[0], beta, R))
        xBs_30.append(xW(y - yt_30, thetas[0], beta, R) - xB)

# Loop through for theta 45 degrees
for y in ys:
    if y <= yt_45:
        xAs_45.append(xB)
        xWs_45.append(0)
        xBs_45.append(-xB)
    else:
        xAs_45.append(x_cutoff_loc(y - yt_45, thetas[1], beta, R))
        xWs_45.append(xW(y - yt_45, thetas[1], beta, R))
        xBs_45.append(xW(y - yt_45, thetas[1], beta, R) - xB)

# Loop through for theta 60 degrees
for y in ys:
    if y <= yt_60:
        xAs_60.append(xB)
        xWs_60.append(0)
        xBs_60.append(-xB)
    else:
        xAs_60.append(x_cutoff_loc(y - yt_60, thetas[2], beta, R))
        xWs_60.append(xW(y - yt_60, thetas[2], beta, R))
        xBs_60.append(xW(y - yt_60, thetas[2], beta, R) - xB)

# Loop through for theta 90 degrees
for y in ys:
    if y <= yt_90:
        xAs_90.append(xB)
        xWs_90.append(0)
        xBs_90.append(-xB)
    else:
        xAs_90.append(x_cutoff_loc(y - yt_90, thetas[3], beta, R))
        xWs_90.append(xW(y - yt_90, thetas[3], beta, R))
        xBs_90.append(xW(y - yt_90, thetas[3], beta, R) - xB)

xAs_30_neg, xWs_30_neg, xBs_30_neg = [value * -1 for value in xAs_30], [
    value * -1 for value in xWs_30], [value * -1 for value in xBs_30]
xAs_45_neg, xWs_45_neg, xBs_45_neg = [value * -1 for value in xAs_45], [
    value * -1 for value in xWs_45], [value * -1 for value in xBs_45]
xAs_60_neg, xWs_60_neg, xBs_60_neg = [value * -1 for value in xAs_60], [
    value * -1 for value in xWs_60], [value * -1 for value in xBs_60]
xAs_90_neg, xWs_90_neg, xBs_90_neg = [value * -1 for value in xAs_90], [
    value * -1 for value in xWs_90], [value * -1 for value in xBs_90]

# Create subplots: 2 rows, 2 columns, sharing y-axis
fig, axs = plt.subplots(2, 2, figsize=(16, 16), sharey=True)  # Increased
    figure size for better spacing

xmin = -40  # Lower limit for the x-axis
xmax = 40   # Upper limit for the x-axis

# Function to plot on a given subplot axis
```

```python
def plot_lines(ax, xs_cut, xAs, xBs, xWs, title, yt, ys, add_legend=False):
    # Plot the cut line (thick)
    ax.plot(xs_cut, ys, 'k', linewidth=2)  # Thick solid line for the cut

    # Plot xA (dashed) and xB (dashed)
    ax.plot(xAs, ys, 'r--', linewidth=1.5, label=r'Contact point A')  # Dashed line for xA
    ax.plot(xBs, ys, 'g--', linewidth=1.5, label=r'Contact point B')  # Dashed line for xB

    # Where xA overlaps with the cut, make it thick dashed
    overlap = np.array(xs_cut) == np.array(xAs)
    ax.plot(np.array(xs_cut)[overlap], np.array(ys)[overlap], 'k--', linewidth=2)  # Thick dashed line for overlap

    # Plot xW (dotted)
    ax.plot(xWs, ys, 'b--', linewidth=1.5, label=r'Wheel center')  # Dotted line for xW

    # Add plot titles and formatting
    ax.axhline(0, color='k', linestyle='-.', linewidth=0.5)  # Thin line at y = 0
    ax.axhline(-bt, color='k', linewidth=2)  # Thick line at y = -bt
    ax.axhline(bt, color='k', linewidth=2)  # Thick line at y = bt
    ax.set_xlim([xmin, xmax])
    ax.set_title(title, fontsize=14)

    # Only add the legend once per subplot
    if add_legend:
        ax.legend(loc='best')

# Function to plot on a given subplot axis with shading
def plot_lines_with_shading(ax, xs_cut, xAs, xBs, xWs, title, yt, yt_neg, ys, add_legend=False):
    # Plot the cut line (thick)
    ax.plot(xs_cut, ys, 'k', linewidth=2)  # Thick solid line for the cut

    # Plot xA (dashed) and xB (dashed)
    ax.plot(xAs, ys, 'r--', linewidth=1.5, label=r'Contact point A')  # Dashed line for xA
    ax.plot(xBs, ys, 'g--', linewidth=1.5, label=r'Contact point B')  # Dashed line for xB

    # Where xA overlaps with the cut, make it thick dashed
    overlap = np.array(xs_cut) == np.array(xAs)
    ax.plot(np.array(xs_cut)[overlap], np.array(ys)[overlap], 'k--', linewidth=2)  # Thick dashed line for overlap

    # Plot xW (dotted)
    ax.plot(xWs, ys, 'b--', linewidth=1.5, label=r'Wheel center')  # Dotted line for xW

    # Add plot titles and formatting
    ax.axhline(0, color='k', linestyle='-.', linewidth=0.5)  # Thin line at y = 0
```

```python
    ax.axhline(-bt, color='k', linewidth=2)  # Thick line at y = -bt
    ax.axhline(bt, color='k', linewidth=2)  # Thick line at y = bt
    ax.set_xlim([xmin, xmax])
    ax.set_title(title, fontsize=14)

    # Only add the legend once per subplot
    if add_legend:
        ax.legend(loc='best')

    # Fill the region between yt_neg and yt with dark gray
    ax.fill_betweenx(ys, xmin, xmax, where=(ys >= yt_neg) & (ys <= yt), color='gray', alpha=0.7)

    # Fill the regions between -bt and yt_neg, and between yt and bt with lighter gray
    ax.fill_betweenx(ys, xmin, xmax, where=(ys >= -bt) & (ys <= yt_neg), color='lightgray', alpha=0.5)
    ax.fill_betweenx(ys, xmin, xmax, where=(ys >= yt) & (ys <= bt), color='lightgray', alpha=0.5)


# Function to plot on a given subplot axis with shading and text annotations
def plot_lines_with_shading_and_text(ax, xs_cut, xAs, xBs, xWs, title, yt, yt_neg, ys, add_legend=False):
    # Plot the cut line (thick)
    ax.plot(xs_cut, ys, 'k', linewidth=2)  # Thick solid line for the cut

    # Plot xA (dashed) and xB (dashed)
    ax.plot(xAs, ys, 'r--', linewidth=1.5, label=r'Contact point A')  # Dashed line for xA
    ax.plot(xBs, ys, 'g--', linewidth=1.5, label=r'Contact point B')  # Dashed line for xB

    # Where xA overlaps with the cut, make it thick dashed
    overlap = np.array(xs_cut) == np.array(xAs)
    ax.plot(np.array(xs_cut)[overlap], np.array(ys)[overlap], 'k--', linewidth=2)  # Thick dashed line for overlap

    # Plot xW (dotted)
    ax.plot(xWs, ys, 'b--', linewidth=1.5, label=r'Wheel center')  # Dotted line for xW

    # Add plot titles and formatting
    ax.axhline(0, color='k', linestyle='-.', linewidth=0.5)  # Thin line at y = 0
    ax.axhline(-bt, color='k', linewidth=2)  # Thick line at y = -bt
    ax.axhline(bt, color='k', linewidth=2)  # Thick line at y = bt
    ax.set_xlim([xmin, xmax])
    ax.set_title(title, fontsize=14)

    # Only add the legend once per subplot
    if add_legend:
        ax.legend(loc='best')

    # Fill the region between yt_neg and yt with dark gray
```

```python
    ax.fill_betweenx(ys, xmin, xmax, where=(ys >= yt_neg) & (ys <= yt), color='gray', alpha=0.7)

    # Fill the regions between -bt and yt_neg, and between yt and bt with lighter gray
    ax.fill_betweenx(ys, xmin, xmax, where=(ys >= -bt) & (ys <= yt_neg), color='lightgray', alpha=0.5)
    ax.fill_betweenx(ys, xmin, xmax, where=(ys >= yt) & (ys <= bt), color='lightgray', alpha=0.5)

    # Add the text "Contact Scenario 1" for the area between -yt and yt
    ax.text(xmax * 0.5, -bt * 0.08 , "Contact Scenario 1", fontsize=12, color='black', ha='center', va='center')

    # Add the text "Contact Scenario 2" for the area between -bt and -yt
    ax.text(xmax * 0.5, (yt_neg + -bt) / 2, "Contact Scenario 2", fontsize=12, color='black', ha='center', va='center')

    # Add the text "Contact Scenario 2" for the area between yt and bt
    ax.text(-xmax * 0.5, (yt + bt) / 2, "Contact Scenario 2", fontsize=12, color='black', ha='center', va='center')

# Now, update the plotting part for each subplot

# 30 degrees case
plot_lines_with_shading_and_text(axs[0, 0], xs_cut_30_neg, xAs_30_neg, xBs_30_neg, xWs_30_neg, r"$\theta = 30^\circ$", yt_30, yt_30_neg, ys_neg, add_legend=True)
plot_lines_with_shading_and_text(axs[0, 0], xs_cut_30, xAs_30, xBs_30, xWs_30, r"$\theta = 30^\circ$", yt_30, yt_30_neg, ys)

# 45 degrees case
plot_lines_with_shading_and_text(axs[0, 1], xs_cut_45_neg, xAs_45_neg, xBs_45_neg, xWs_45_neg, r"$\theta = 45^\circ$", yt_45, yt_45_neg, ys_neg, add_legend=True)
plot_lines_with_shading_and_text(axs[0, 1], xs_cut_45, xAs_45, xBs_45, xWs_45, r"$\theta = 45^\circ$", yt_45, yt_45_neg, ys)

# 60 degrees case
plot_lines_with_shading_and_text(axs[1, 0], xs_cut_60_neg, xAs_60_neg, xBs_60_neg, xWs_60_neg, r"$\theta = 60^\circ$", yt_60, yt_60_neg, ys_neg, add_legend=True)
plot_lines_with_shading_and_text(axs[1, 0], xs_cut_60, xAs_60, xBs_60, xWs_60, r"$\theta = 60^\circ$", yt_60, yt_60_neg, ys)

# 90 degrees case
plot_lines_with_shading_and_text(axs[1, 1], xs_cut_90_neg, xAs_90_neg, xBs_90_neg, xWs_90_neg, r"$\theta = 90^\circ$", yt_90, yt_90_neg, ys_neg, add_legend=True)
plot_lines_with_shading_and_text(axs[1, 1], xs_cut_90, xAs_90, xBs_90, xWs_90, r"$\theta = 90^\circ$", yt_90, yt_90_neg, ys)

# Add x and y labels with units
for ax in axs.flat:
    ax.set_xlabel(r"$x$ [mm]", fontsize=12)
    ax.set_aspect('equal')
```

```python
axs[0, 0].set_ylabel(r"$y$ [mm]", fontsize=12)  # Set y-label only once
    since they share y-axis
axs[1, 0].set_ylabel(r"$y$ [mm]", fontsize=12)
axs[0, 0].invert_yaxis()

# Adjust layout
plt.tight_layout()


# Determine the maginutude of vertical impulse during contact:
def vertical_impulse_during_contact_1(beta, v, M):
    Mz0 = -np.sin(beta / 2) * v * M
    Mz1 = np.sin(beta / 2) * v * M

    return (Mz1 - Mz0)


def vertical_impulse_during_contact_2(y_loc, theta, beta, R, M, v):
    xA = get_xA(y_loc, theta, beta, R)
    alpha = get_alpha(xA, R)

    Mz0 = -np.sin(beta / 2) * v * M
    Mz1 = np.sin(alpha) * v * M

    return (Mz1 - Mz0)


# Plot the vertical impulse as a function of beta for both contact
    scenario's:
# Constants
y_w = 15  # mm
M = 750   # kg
v = 50    # m/s
thetas = [30 * np.pi / 180, 45 * np.pi / 180, 60 * np.pi / 180]
betas = np.linspace(0, 0.15, 1000)  # Beta range
colors = ['r', 'g', 'b']
linestyles = ['solid', 'solid', 'solid']
intersect_text = [r'$\beta_{t_{30}}$', r'$\beta_{t_{45}}$', r'$\beta_{t_
    {60}}$']
theta_labels = [30, 45, 60]  # Labels for degrees

# Create subplots: 1 row, 3 columns (one for each theta)
fig, axs = plt.subplots(1, 3, figsize=(18, 6), sharey=True)

# Loop over the range of theta indices
for i in range(len(thetas)):
    theta = thetas[i]  # Get the current theta value
    vertical_impulses_1 = []
    vertical_impulses_2 = []

    # Calculate vertical impulses for both contact scenarios
    for beta in betas:
        y_loc = y_w - yt(theta, beta, R)
        vertical_impulses_1.append(vertical_impulse_during_contact_1(beta,
            v, M))
```

```
        vertical_impulses_2.append(vertical_impulse_during_contact_2(y_loc, theta, beta, R, M, v))

    # Find the intersection point where both lines intersect
    difference = np.array(vertical_impulses_1) - np.array(vertical_impulses_2)
    intersection_points = np.argwhere(np.diff(np.sign(difference))).flatten()

    # Exclude the first intersection point if it's at the start
    if len(intersection_points) > 0 and intersection_points[0] == 0:
        intersection_points = intersection_points[1:]

    # Plot on the corresponding subplot
    ax = axs[i]

    if len(intersection_points) > 0:
        x_intersection = betas[intersection_points[0]]

        # Plot Contact Scenario 2 as defined styles up to the intersection
        ax.plot(betas[:intersection_points[0] + 1], vertical_impulses_2[:intersection_points[0] + 1],
                label=f'Contact Scenario 2 ( = {theta_labels[i]}°)',
                color=colors[i], linestyle=linestyles[i])

        # Plot Contact Scenario 1 as dotted line up to the intersection
        ax.plot(betas[:intersection_points[0] + 1], vertical_impulses_1[:intersection_points[0] + 1],
                label=f'Contact Scenario 1 reference line',
                linestyle='--', color='black')

        # Plot Contact Scenario 1 as solid line after the intersection
        ax.plot(betas[intersection_points[0]:], vertical_impulses_1[intersection_points[0]:],
                label=f'Contact Scenario 1',
                linestyle='-', color='black')

        # Add vertical dashed line at the intersection point
        ax.axvline(x=x_intersection, color='black', linestyle=':')

        # Add label for intersection point
        ax.text(x_intersection, 5500, intersect_text[i], fontsize=12, rotation=90,
                verticalalignment='center', horizontalalignment='right')
    else:
        # If no intersection point, plot both scenarios normally
        ax.plot(betas, vertical_impulses_1,
                label=f'Contact Scenario 1 ( = {int(np.degrees(theta))}°)',
                linestyle=':', color='black')
        ax.plot(betas, vertical_impulses_2,
                label=f'Contact Scenario 2 ( = {int(np.degrees(theta))}°)',
                color=colors[i], linestyle=linestyles[i])

    # Set labels and grid for each subplot
```

```
    ax.set_xlabel(r'Dip angle $\beta$ [-]')
    ax.grid(True)

    # Add title with LaTeX syntax for incline angle
    ax.set_title(r'Incline angle = {0}°'.format(theta_labels[i]))

# Add y-axis label to the first subplot
axs[0].set_ylabel(r'Vertical Impulse $J_z$ [$N \cdot s$]')

# Add a legend to each subplot
for ax in axs:
    ax.legend()

# Adjust the layout
plt.tight_layout()

# Plot the vertical impulse as a function of beta for both contact scenario's:
y_w = 15 # mm
R = 475 # mm
M = 750 # kg
beta = 0.03 # [rad]
theta = 30 * np.pi / 180
y_loc = y_w - yt(theta, beta, R) # Determine the distance between yt and the wheel

vs = np.linspace(0, 50, 100)

impulses_v_1 = []
impulses_v_2 = []

for v in vs:
    impulses_v_1.append(vertical_impulse_during_contact_1(beta, v, M))
    impulses_v_2.append(vertical_impulse_during_contact_2(y_loc, theta, beta, R, M, v))

# Creating a subplot with 1 row and 2 columns
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 6))

# First plot: Contact Scenario 1
ax1.plot(vs, impulses_v_1, 'b')
ax1.set_title('Contact Scenario 1')
ax1.set_xlabel(r'Train velocity $v$ (m/s)')
ax1.set_ylabel(r'Vertical Impulse $J_z$ [$N \cdot s$]')
ax1.grid(True)
ax1.margins(x=0)
ax1.margins(y=0)

# Second plot: Contact Scenario 2
ax2.plot(vs, impulses_v_2, 'r')
ax2.set_title('Contact Scenario 2')
ax2.set_xlabel(r'Train velocity $v$ (m/s)')
ax2.set_ylabel(r'Vertical Impulse $J_z$ [$N \cdot s$]')
ax2.grid(True)
ax2.margins(x=0)
ax2.margins(y=0)
```

```
# Adjusting layout for clarity
plt.tight_layout()

# Display the plots
plt.show()
```

D

Local Wheel-Rail Interface Model Code

```python
import matplotlib.pyplot as plt
import numpy as np
from shapely.geometry import Polygon


class PlotData:
    def __init__(self, x_data, y_data, xlabel, ylabel, title):
        self.x_data = x_data
        self.y_data = y_data
        self.xlabel = xlabel
        self.ylabel = ylabel
        self.title = title


def create_parallelogram(b, g, theta):
    g_long = g*(1/np.sin(theta))

    P1 = (0, 0)
    P2 = (b/np.tan(theta), b)
    P3 = (b/np.tan(theta) + g_long, b)
    P4 = (g_long, 0)

    parallelogram = Polygon([P1, P2, P3, P4])
    return parallelogram


def create_rectangle(a, b, x, v, time_domain):
    if time_domain:
        vertices = [((x * v), 0), ((x * v) - a, 0), ((x * v) - a, b), ((x * v), b)]
    else:
        vertices = [(x, 0), (x-a, 0), (x-a, b), (x, b)]
    return Polygon(vertices)


def create_polygon(a, b, num_vertices, x, v, time_domain):
    phis = np.linspace(0, 2 * np.pi, num_vertices + 1)
```

```
    if time_domain:
        vertices = [((a / 2) * np.cos(phi) - (a / 2) + x * v, (b / 2) * np
            .sin(phi) + (b / 2)) for phi in phis]
    else:
        vertices = [((a / 2) * np.cos(phi) - (a / 2) + x, (b / 2) * np.sin
            (phi) + (b / 2)) for phi in phis]
    return Polygon(vertices)


def get_xs(a, b, g, theta, steps, v, time_domain):
    g_long = g * (1 / np.sin(theta))
    start_x = 0
    if time_domain:
        end_x = (b / np.tan(theta) + g_long + a) / v
    else:
        end_x = b / np.tan(theta) + g_long + a

    xs = np.linspace(start_x, end_x, steps)
    return xs


def get_derivative(xs, ys):
    # Get rate of change of a plot using central difference method
    dx = xs[1] - xs[0]
    dydx = np.gradient(ys, dx)
    return dydx


def find_x_local_maxima(x_data, y_data):
    maxima = []

    for i in range(1, len(y_data) - 1):
        if y_data[i-1] < y_data[i] > y_data[i+1]:
            maxima.append(x_data[i])
    return maxima


def find_contact_patches(a, b, g, theta, steps, v, time_domain,
        num_vertices, shape, x_values):
    contact_patches = get_reduction_areas(a, b, g, theta, steps, v,
        time_domain, num_vertices, shape)[0]
    x_values_patches = []
    found_contact_patches = []

    for contact_patch in contact_patches:
        if time_domain:
            x_value_patch = contact_patch.exterior.xy[0][0] / v
            x_values_patches.append(x_value_patch)
        else:
            x_value_patch = contact_patch.exterior.xy[0][0]
            x_values_patches.append(x_value_patch)

    for i, x_value_patch in enumerate(x_values_patches):
        if x_value_patch in x_values:
            found_contact_patches.append(contact_patches[i])
    return found_contact_patches
```

```python


def get_initial_area(a, b, v, time_domain, num_vertices, shape):
    if shape == 'Rectangle':
        return create_rectangle(a, b, 0, v, time_domain).area
    elif shape == 'Polygon':
        return create_polygon(a, b, num_vertices, 0, v, time_domain).area
    else:
        raise ValueError("Shape type not supported")


def get_reduction_areas(a, b, g, theta, steps, v, time_domain,
    num_vertices, shape):
    xs = get_xs(a, b, g, theta, steps, v, time_domain)

    joint = create_parallelogram(b, g, theta)

    contact_patches = []
    reduction_areas = []

    if shape == 'Rectangle':
        def create_contact_patch(x_pos): return create_rectangle(a, b,
            x_pos, v, time_domain)
    elif shape == 'Polygon':
        if num_vertices is None:
            raise ValueError("num_vertices must be provided for polygon 
                shape")

        def create_contact_patch(x_pos): return create_polygon(a, b,
            num_vertices, x_pos, v, time_domain)
    else:
        raise ValueError("Unsupported shape type")

    for x in xs:
        contact_patch = create_contact_patch(x)
        contact_patches.append(contact_patch)
        intersection = joint.intersection(contact_patch)
        reduction_areas.append(intersection.area)

    return contact_patches, reduction_areas


def get_reduction_areas_relative(a, b, g, theta, steps, v, time_domain,
    num_vertices, shape):
    reduction_areas = get_reduction_areas(a, b, g, theta, steps, v,
        time_domain, num_vertices, shape)
    initial_area = get_initial_area(a, b, v, time_domain, num_vertices,
        shape)

    reduction_areas_relative = [reduction_area / initial_area for
        reduction_area in reduction_areas[1]]
    return reduction_areas_relative


def get_contact_areas(a, b, g, theta, steps, v, time_domain, num_vertices,
     shape):
```

```
    A0 = get_initial_area(a, b, v, time_domain, num_vertices, shape)
    reduction_areas = get_reduction_areas(a, b, g, theta, steps, v,
        time_domain, num_vertices, shape)[1]
    contact_areas = [A0 - reduction_area for reduction_area in
        reduction_areas]
    return contact_areas


def get_contact_areas_relative(a, b, g, theta, steps, v, time_domain,
    num_vertices, shape):
    contact_areas = get_contact_areas(a, b, g, theta, steps, v,
        time_domain, num_vertices, shape)
    initial_area = get_initial_area(a, b, v, time_domain, num_vertices,
        shape)
    relative_contact_areas = [(contact_area / initial_area) for
        contact_area in contact_areas]
    return relative_contact_areas


def get_uniform_contact_stresses(N, a, b, g, theta, steps, v, time_domain,
     num_vertices, shape):
    contact_areas = get_contact_areas(a, b, g, theta, steps, v,
        time_domain, num_vertices, shape)
    uniform_contact_stresses = [N / contact_area for contact_area in
        contact_areas]
    return uniform_contact_stresses


def get_uniform_contact_stresses_derivative(N, a, b, g, theta, steps, v,
    time_domain, num_vertices, shape):
    x_data = get_xs(a, b, g, theta, steps, v, time_domain)
    y_data = get_uniform_contact_stresses(N, a, b, g, theta, steps, v,
        time_domain, num_vertices, shape)
    y_data = abs(get_derivative(x_data, y_data))
    return y_data


def get_uniform_contact_stresses_relative(N, a, b, g, theta, steps, v,
    time_domain, num_vertices, shape):
    uniform_contact_stresses = get_uniform_contact_stresses(N, a, b, g,
        theta, steps, v, time_domain, num_vertices, shape)
    initial_contact_stress = uniform_contact_stresses[0]
    relative_uniform_contact_stresses = [uniform_contact_stress /
        initial_contact_stress for uniform_contact_stress in
        uniform_contact_stresses]
    return relative_uniform_contact_stresses


def get_linear_contact_stresses(N, a, b):
    # Calculate the contact stress at point (x, y)
    p_max = 3 * N / (a * b)


def plot_reduction_areas(a, b, g, theta, steps, v, time_domain,
    num_vertices, shape):
    xs = get_xs(a, b, g, theta, steps, v, time_domain)
```

```python
    y_data = get_reduction_areas(a, b, g, theta, steps, v, time_domain, num_vertices, shape)[1]

    if time_domain:
        xlabel = 'Time [s]'
        ylabel = 'Reduction of contact area [mm2]'
        title = f'Reduction of contact area for theta = {round(theta * (180 / np.pi))} [deg]'
    else:
        xlabel = 'Position [mm]'
        ylabel = 'Reduction of contact area [mm2]'
        title = f'Reduction of contact area for theta = {round(theta * (180 / np.pi))} [deg]'
    plot_data = PlotData(xs, y_data, xlabel, ylabel, title)
    return plot_data


def plot_reduction_areas_relative(a, b, g, theta, steps, v, time_domain, num_vertices, shape):
    xs = get_xs(a, b, g, theta, steps, v, time_domain)
    y_data = get_reduction_areas_relative(a, b, g, theta, steps, v, time_domain, num_vertices, shape)

    if time_domain:
        xlabel = 'Time [s]'
        ylabel = 'Ri/A0 [-]'
        title = f'Ratio of reduction area to initial area for theta = {round(theta * (180 / np.pi))} [deg]'
    else:
        xlabel = 'Position [mm]'
        ylabel = 'Ri/A0 [-]'
        title = f'Ratio of reduction area to initial area for theta = {round(theta * (180 / np.pi))} [deg]'
    plot_data = PlotData(xs, y_data, xlabel, ylabel, title)
    return plot_data


def plot_contact_areas(a, b, g, theta, steps, v, time_domain, num_vertices, shape):
    xs = get_xs(a, b, g, theta, steps, v, time_domain)
    y_data = get_contact_areas(a, b, g, theta, steps, v, time_domain, num_vertices, shape)

    if time_domain:
        xlabel = 'Time [s]'
        ylabel = 'Contact area [mm2]'
        title = f'Active contact area for theta =  {round(theta * (180 / np.pi))} [deg]'
    else:
        xlabel = 'Position [mm]'
        ylabel = 'Contact area [mm2]'
        title = f'Active contact area for theta =  {round(theta * (180 / np.pi))} [deg]'
    plot_data = PlotData(xs, y_data, xlabel, ylabel, title)
    return plot_data

```

```python

def plot_contact_areas_relative(a, b, g, theta, steps, v, time_domain, num_vertices, shape):
    xs = get_xs(a, b, g, theta, steps, v, time_domain)
    y_data = get_contact_areas_relative(a, b, g, theta, steps, v, time_domain, num_vertices, shape)

    if time_domain:
        xlabel = 'Time [s]'
        ylabel = 'Ai/A0 [-]'
        title = f'Ratio of active contact area to initial contact area for theta = {round(theta * (180 / np.pi))} [deg]'
    else:
        xlabel = 'Position [mm]'
        ylabel = 'Ai/A0 [-]'
        title = f'Ratio of active contact area to initial contact area for theta = {round(theta * (180 / np.pi))} [deg]'
    plot_data = PlotData(xs, y_data, xlabel, ylabel, title)
    return plot_data


def plot_uniform_contact_stresses(N, a, b, g, theta, steps, v, time_domain, num_vertices, shape):
    xs = get_xs(a, b, g, theta, steps, v, time_domain)
    y_data = get_uniform_contact_stresses(N, a, b, g, theta, steps, v, time_domain, num_vertices, shape)

    if time_domain:
        xlabel = 'Time [s]'
        ylabel = 'Uniform contact stresses [MPa]'
        title = f'Uniform contact stresses for theta = {round(theta * (180 / np.pi))} [deg]'
    else:
        xlabel = 'Position [mm]'
        ylabel = 'Uniform contact stresses [MPa]'
        title = f'Uniform contact stresses for theta = {round(theta * (180 / np.pi))} [deg]'
    plot_data = PlotData(xs, y_data, xlabel, ylabel, title)
    return plot_data


def plot_uniform_contact_stresses_relative(N, a, b, g, theta, steps, v, time_domain, num_vertices, shape):
    xs = get_xs(a, b, g, theta, steps, v, time_domain)
    y_data = get_uniform_contact_stresses_relative(N, a, b, g, theta, steps, v, time_domain, num_vertices, shape)

    if time_domain:
        xlabel = 'Time [s]'
        ylabel = 'Rel. contact pressure Pi/P0 [-]'
        title = f'Ratio of uniform contact stress to initial contact stress for theta = {round(theta * (180 / np.pi))} [deg]'
    else:
        xlabel = 'Position [mm]'
        ylabel = 'Rel. contact pressure Pi/P0 [-]'
```

```
        title = f'Ratio of uniform contact stress to initial contact stress for theta = {round(theta * (180 / np.pi))} [deg]'
    plot_data = PlotData(xs, y_data, xlabel, ylabel, title)
    return plot_data


def plot_max_uniform_contact_stress_vs_theta(N, a, b, g, steps, v, time_domain, num_vertices, shape):
    thetas = np.linspace(1, 90, 100)
    max_stresses = []

    for theta in thetas:
        theta = theta * (np.pi / 180)
        max_stresses.append(max(get_uniform_contact_stresses(N, a, b, g, theta, steps, v, time_domain, num_vertices, shape)))

    x_data = thetas
    y_data = max_stresses
    xlabel = 'Theta [deg]'
    ylabel = 'Stress [MPa]'
    title = 'Max stress during joint traversal as function of theta'
    plot_data = PlotData(x_data, y_data, xlabel, ylabel, title)
    return plot_data


def plot_derivative_uniform_contact_stresses(N, a, b, g, theta, steps, v, time_domain, num_vertices, shape):
    x_data = get_xs(a, b, g, theta, steps, v, time_domain)
    y_data = get_uniform_contact_stresses(N, a, b, g, theta, steps, v, time_domain, num_vertices, shape)
    y_data = get_derivative(x_data, y_data)
    y_data = [abs(y) for y in y_data]

    if time_domain:
        xlabel = 'Time [s]'
        ylabel = 'Rate of change [MPa/s]'
        title = 'Uniform contact stress derived to time'
    else:
        xlabel = 'Position [mm]'
        ylabel = 'Rate of change [MPa/mm]'
        title = 'Uniform contact stress derived to position'
    plot_data = PlotData(x_data, y_data, xlabel, ylabel, title)
    return plot_data


def plot_contact_patch(contact_patch):
    x_contact_patch, y_contact_patch = contact_patch.exterior.xy
    fig, ax = plt.subplots()
    ax.fill(x_contact_patch, y_contact_patch, edgecolor='black', facecolor='black', alpha=0.6)
    ax.invert_yaxis()
    ax.set_xlabel('[mm]')
    ax.set_ylabel('[mm]')
    ax.axis('equal')
    ax.grid(True)

```

```python

def plot_system_static(contact_patch, joint, position=0):
    x_contact_patch, y_contact_patch = contact_patch.exterior.xy
    x_joint, y_joint = joint.exterior.xy

    x_contact_patch = np.array(x_contact_patch)
    x_contact_patch += position

    fig, ax = plt.subplots()
    ax.fill(x_contact_patch, y_contact_patch, edgecolor='red', facecolor='
        None', alpha=0.8)
    ax.fill(x_joint, y_joint, edgecolor='black', facecolor='None')
    ax.invert_yaxis()
    ax.axis('equal')
    ax.grid(True)


def plot_system_dynamic(contact_patches, joint):
    x_joint, y_joint = joint.exterior.xy

    xs_contact_patch, ys_contact_patch = [], []

    #labels = ['x_start', 'x_1', 'x_2', 'x_end']

    for contact_patch in contact_patches:
        x_contact_patch, y_contact_patch = contact_patch.exterior.xy
        xs_contact_patch.append(x_contact_patch)
        ys_contact_patch.append(y_contact_patch)

    fig, ax = plt.subplots()
    ax.fill(x_joint, y_joint, edgecolor='black', facecolor='black', alpha
        =0.5)

    colors = ['red', 'green', 'blue', 'orange', 'purple']
    for i in range(len(xs_contact_patch)):
        color = colors[i % len(colors)]
        ax.fill(xs_contact_patch[i], ys_contact_patch[i], edgecolor=color,
             facecolor=color, alpha=0.4)

        #x_front = min(xs_contact_patch[i])
        #ax.plot([x_front + 14, x_front + 14], [0, max(y_joint) + 6],
            color='black', linestyle='dotted')
        #ax.text(x_front + 14, max(y_joint) + 7, labels[i], rotation=0, va
            ='top', ha='center')

    ax.set_xlabel('[mm]')
    ax.set_ylabel('[mm]')
    ax.invert_yaxis()
    ax.axis('equal')
    ax.grid(True)


def plot_critical_points(N, a, b, g, theta, steps, v, time_domain,
    num_vertices, shape):
    xs = get_xs(a, b, g, theta, steps, v, time_domain)
```

```
    ys = get_uniform_contact_stresses_derivative(N, a, b, g, theta, steps,
         v, time_domain, num_vertices, shape)
    xs_maxima = find_x_local_maxima(xs, ys)
    polygons = find_contact_patches(a, b, g, theta, steps, v, time_domain,
         num_vertices, shape, xs_maxima)

    fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 8), sharex=True)

    ax1.plot(xs, ys)
    ax1.grid(True)
    ax1.set_ylabel('Rate of change [MPa/mm]')

    for polygon in polygons:
        x, y = polygon.exterior.xy
        if time_domain:
            x = [x_val / v for x_val in x]
            y = [y_val / v for y_val in y]
        ax2.fill(x, y, edgecolor='r', facecolor='none')

    joint = create_parallelogram(b, g, theta)
    x_joint, y_joint = joint.exterior.xy

    if time_domain:
        x_joint = [x_val / v for x_val in x_joint]
        y_joint = [y_val / v for y_val in y_joint]

    ax2.fill(x, y, edgecolor='r', facecolor='none')
    ax2.fill(x_joint, y_joint, edgecolor='k', facecolor='none')

    ax2.set_aspect('equal')
    ax2.set_xlabel('x [mm]')
    ax2.set_ylabel('y [mm]')
    ax2.invert_yaxis()

    if time_domain:
        ax1.set_ylabel('Rate of change [MPa/s]')
        ax2.set_xlabel('t [s]')
        y_ticks = ax2.get_yticks()
        ax2.set_yticklabels([f'{y_tick * v:.0f}' for y_tick in y_ticks])

    for x_max in xs_maxima:
        ax1.axvline(x=x_max, color='gray', linestyle='--')
        ax2.axvline(x=x_max, color='gray', linestyle='--')


def main(N, a, b, g, theta, steps, v=0, time_domain=False, num_vertices=
    None, shape='Rectangle', plot_geometry=False, plot_critical=False):
    # Convert from degrees to radians to use in numpy functions
    theta = theta * (np.pi / 180)

    # Check if there are enough vertices to make a reasonable shape
    if shape == 'Polygon' and num_vertices < 4:
        raise ValueError("Number of vertices should be minimum 4")

    # Using self defined plot functions to gather all the plot data using
        self defined PlotData object
```

```
    p1 = plot_contact_areas(a, b, g, theta, steps, v, time_domain, num_vertices, shape)
    p2 = plot_contact_areas_relative(a, b, g, theta, steps, v, time_domain, num_vertices, shape)
    p3 = plot_reduction_areas(a, b, g, theta, steps, v, time_domain, num_vertices, shape)
    p4 = plot_reduction_areas_relative(a, b, g, theta, steps, v, time_domain, num_vertices, shape)
    p5 = plot_uniform_contact_stresses(N, a, b, g, theta, steps, v, time_domain, num_vertices, shape)
    p6 = plot_uniform_contact_stresses_relative(N, a, b, g, theta, steps, v, time_domain, num_vertices, shape)
    p7 = plot_max_uniform_contact_stress_vs_theta(N, a, b, g, steps, v, time_domain, num_vertices, shape)
    p8 = plot_derivative_uniform_contact_stresses(N, a, b, g, theta, steps, v, time_domain, num_vertices, shape)
    plot_list = [p1, p2, p3, p4, p5, p7, p6, p8]

    # Create the subplot figure and fill it with the plot data
    n_cols = 2  # Make sure there is 2 columns
    n_rows = (len(plot_list) + n_cols - 1) // n_cols  # Determine the number of rows needed
    fig, axs = plt.subplots(n_rows, n_cols, figsize=(15, 10))  # Create figure with right amount of subplots

    for i, p in enumerate(plot_list):
        row = i // n_cols
        col = i % n_cols
        ax = axs[row, col] if n_rows > 1 else axs[col]

        ax.plot(p.x_data, p.y_data, color='k')
        ax.set_xlabel(p.xlabel)
        ax.set_ylabel(p.ylabel)
        #ax.set_title(p.title)
        ax.grid(True)

    # Remove the empty subplots if they are there
    for j in range(len(plot_list), n_rows * n_cols):
        fig.delaxes(axs.flatten()[j])

    plt.tight_layout()

    # Plot the geometry at first step and all steps if necessary to visualize the system
    if plot_geometry:
        plot_contact_patch(get_reduction_areas(a, b, g, theta, steps, v, time_domain, num_vertices, shape)[0][0])
        plot_system_static(get_reduction_areas(a, b, g, theta, steps, v, time_domain, num_vertices, shape)[0][0], create_parallelogram(b, g, theta))
        plot_system_dynamic(get_reduction_areas(a, b, g, theta, steps, v, time_domain, num_vertices, shape)[0], create_parallelogram(b, g, theta))

    # if plot_critical:
```

```python
        #plot_critical_points(N, a, b, g, theta, steps, v, time_domain, num_vertices, shape)

    # This part of code creates a plot that describes the relationship between the total area, reduction area and total area
    plt.figure()
    plt.plot(p1.x_data, p2.y_data, color='black', label='Active contact area Ai')
    plt.plot(p3.x_data, p4.y_data, color='black', linestyle='dashed', label='Reduction area Ri')
    plt.plot(p1.x_data, [p2.y_data[i] + p4.y_data[i] for i in range(len(p4.y_data))], color='black', linestyle='dashdot', label='Total contact area A0')
    plt.xlabel('Position front of contact patch x_front [mm]')
    plt.ylabel('Relative area [-]')
    plt.legend(loc='best')
    plt.grid(True)

    plt.tight_layout()
    plt.show()


# The main function plots and tests all the functions and is used to build the file. The create_plots_report function creates the plots for the report.
# main(N=112.5*10**3, a=16, b=12, g=6, theta=45, steps=100, v=50*10**3, time_domain=True, num_vertices=100, shape='Polygon', plot_geometry=True, plot_critical=False)


def create_plots_report(N, a, b, g, theta, steps, v, time_domain=False, num_vertices=None, shape='Rectangle', plot_geometry=True, plot_critical=True):
    thetas = [30 * np.pi / 180, 20 * np.pi / 180, 10 * np.pi / 180, 5 * np.pi / 180]
    contact_pressure_plot_data = []
    contact_pressure_plot_data_relative = []
    rate_of_change_plot_data = []
    for theta in thetas:
        contact_pressure_plot_data.append(plot_uniform_contact_stresses(N, a, b, g, theta, steps, v, time_domain, num_vertices, shape))
        contact_pressure_plot_data_relative.append(
            plot_uniform_contact_stresses_relative(N, a, b, g, theta, steps, v, time_domain, num_vertices, shape))
        rate_of_change_plot_data.append(
            plot_derivative_uniform_contact_stresses(N, a, b, g, theta, steps, v, time_domain, num_vertices, shape))

    linestyles = ['solid', 'dotted', 'dashdot', 'dashed']
    labels = ['$\\theta$ = 30 [deg]', '$\\theta$ = 20 [deg]', '$\\theta$ = 10 [deg]', '$\\theta$ = 5 [deg]']
    colours = ['red', 'green', 'blue', 'black']

    # Plot the contact pressures:
    plt.figure()
    for i, data in enumerate(contact_pressure_plot_data):
```

```
        plt.plot(data.x_data, data.y_data, 'k', linestyle=linestyles[i], label=labels[i])
    plt.xlabel("Time [$s$]")
    plt.ticklabel_format(style='sci', axis='x', scilimits=(0, 0))
    plt.ylabel(r'Uniform Contact Pressure $P$ [$MPa$]')
    plt.legend()
    plt.grid(True)

    # Plot the contact pressures (relative):
    plt.figure()
    for i, data in enumerate(contact_pressure_plot_data_relative):
        plt.plot(data.x_data, data.y_data, color=colours[i], linestyle=linestyles[i], label=labels[i])
    plt.xlabel("Time [$s$]")
    plt.ticklabel_format(style='sci', axis='x', scilimits=(0, 0))
    plt.ylabel(r'Rel. Contact Pressure Pi/P0 [-]')
    plt.legend()
    plt.grid(True)

    # Plot the rate of change:
    plt.figure()
    for i, data in enumerate(rate_of_change_plot_data):
        plt.plot(data.x_data, data.y_data, color=colours[i], linestyle=linestyles[i], label=labels[i])
    plt.xlabel("Time [$s$]")
    plt.ticklabel_format(style='sci', axis='x', scilimits=(0, 0))

    if time_domain:
        plt.ylabel(r'Contact Pressure rate of change [$MPa/s$]')
    else:
        plt.ylabel(r'Contact Pressure rate of change [$MPa/mm$]')
    plt.legend()
    plt.grid(True)
    plt.show()


create_plots_report(N=112.5*10**3, a=16, b=12, g=6, theta=45, steps=1000,
    v=50*10**3, time_domain=True, num_vertices=100, shape='Polygon',
    plot_geometry=True, plot_critical=False)
```