



**Politecnico
di Torino**

Politecnico di Torino

Mechanical Engineering Master's Degree

Academic Year: 2024/2025

Graduation Period: October 2025

Artificial Intelligence in Mechanical Design

A User-Interaction Framework

Academic Supervisor:

Prof. Nicolò Zampieri

Company Supervisor:

Eng. Marco Borlo

Candidate:

Daniele Spano

S331732

Contents

PART I INTRODUCTION TO RAILWAY BOGIES

<i>Rail Transport</i>	5
<i>Trains</i>	8
<i>Railway Vehicles</i>	9
<i>Railway Bogies</i>	13

PART II INTRODUCTION TO THE PROJECT

<i>Foreword</i>	22
<i>Comparison of different design procedures</i>	23
<i>Implementation of GUIDE</i>	28

PART III AI IMPLEMENTATION

<i>Requirements and constraints</i>	31
<i>Conventional AI implementations</i>	34
<i>The G-Map Approach</i>	38
<i>AI Implementation in the Project</i>	45

PART IV USER INTERFACE

<i>Requirements and implementation</i>	58
<i>Interface and Controls</i>	63

PART V STATIC ANALYSIS

<i>Introduction</i>	69
<i>The Finite Elements Method</i>	71
<i>FEM in structural static analysis</i>	73
<i>Beam elements</i>	76
<i>FEA implementation in the project</i>	79
<i>Primary suspension modeling</i>	83
<i>Modeling of nonlinearities</i>	87

<i>Analysis outputs</i>	92
<i>Results and observations</i>	97
<i>A case study: bogie pitch analysis</i>	101

Abstract

The objective of this thesis is to explore the integration of AI technology in the preliminary design of mechanical systems and components and to evaluate its capabilities and performance.

A theoretical-only analysis would most likely fail to properly showcase the possibilities that the introduction of AI in the design process could offer. The realization of a practical working example has therefore been deemed necessary.

Unfortunately, an extremely large variety of mechanical systems exists and developing a framework capable of handling all of them would be impossible, from a practical perspective at least. As a consequence, the approach that has been adopted is to:

1. develop an interactive AI design framework from a general perspective, that can be adapted to as many mechanical systems as possible
2. realize a practical implementation to showcase and test its capabilities and shortcomings.

The underlying assumption, which will be a common theme of the project, is that concepts and tools that are developed in the practical realization can be relatively easily extended to other mechanical systems with minor adjustments.

The design of a railway bogie has been chosen for this project: it has been deemed to be complex enough to test a wide range of functionalities and models, while avoiding excessive intricacies.

Railway bogies and their role in the broader context of railway transportation will be therefore introduced first.

The requirements of such an implementation will then be discussed as a basis for its development. As the introduction of AI in mechanical design is a relatively new and unexplored field, a completely new operating framework has been developed from the ground up, known as the *GUIDE framework*. Its main features will be discussed and compared to other more typical design procedures.

The adopted AI model too represents a novelty and has been conceived from scratch for this project. Its working principles, and the underlying mathematical concepts, will be therefore presented.

The realization of the user interface and the implementation of analysis tools will be discussed and, as a conclusion, results and examples will be presented.

Part I

Introduction to Railway Bogies

Rail Transport

Rail transport includes an extremely vast collection of vehicles and infrastructures; their main common feature is the adoption of wheeled vehicles running in tracks, which guide their trajectory¹.

To this day, rail based transportation systems represent one of the most widespread solution for rapid and efficient movement of passengers and freight. Its applications span a wide spectrum of applications² and operating environments³.

While exact figures are difficult to evaluate, it is estimated that approximately 1.2 millions km of railways are currently operated globally, transporting 50-60 billions passengers and roughly 15 billions of cargo per year.

While it is true that the broad definition of *rail transport* includes a wide range of solutions, only categories that are relevant for the understanding of railway bogies will be hereby considered⁴.

¹ This is to be intended in a broad sense: while the most common form of tracks is represented by two parallel metallic rails, other options have been developed. Magnetic tracks employed to guide and propel maglevs and the single tracks in the case of monorails are examples of alternative layouts.

² e.g. from short urban commuter routes to long-distance and even inter-continental freight transport.

³ e.g. flat plains, mountains, underground networks, etc.

⁴ A detailed description of all rail transport solution would in fact be far beyond the scope of this introduction.



Figure 1: Examples of rail transport systems: (from top to bottom, left to right)

High-speed rail: ETR500 Frecciarossa operated by Trenitalia

Monorail: typically adopted for short routes; nowadays this solution is rarely employed

Metro: trains that typically operate underground in urban environments

Maglev: magnetic levitation is employed to reduce friction and allow for extremely high speeds

Rack railways: commonly adopted solution in steep and mountainous terrains

Funiculars: employed in similar contexts to rack railways

Solutions adopted in rail transport can significantly vary due to the extremely wide range of environment in which they might operate and their design objectives.

Diversity among rail based systems and vehicles is furthermore enhanced by their typical longevity: in a railway system, vehicles and infrastructures designed and realized some decades apart may coexist.

History and development

The origins of rail based transportation trace back to ancient Greece and Rome: carts running on grooves or wooden tracks were typically employed in mines to transport extracted ores. This remained the case until the beginning of the 19th century, albeit with some improvements: wagonways with wooden or iron rails, often horse-drawn, were common in European mines.

The Industrial Revolution marked a significant turning point: the introduction of steam engines and the increased need for fast transportation of freight and passengers led to the development of early railways in Britain, where mining wagonways were adapted⁵ and improved with iron rails and steam-powered locomotives. This solution quickly proved its effectiveness and, inspired by Britain's success, railways spread rapidly across Europe and North America.

Railways soon started to be employed in urban transportation too: the world's first underground railway opened in London in 1863, using steam trains to connect central areas with surrounding districts; Paris followed with the opening of the Métro in 1900, a fully electric underground system designed for efficient urban commuting.

Around the same time, trams became widespread in cities across Europe and North America. Initially horse-drawn and later electrified, trams offered a flexible and affordable solution for short-distance urban travel and played a key role in shaping early public transportation networks⁶. Technological advancements throughout the 19th century, such as steel rails, standardized gauges, air brakes, and telegraphic signaling, vastly improved safety and efficiency.

In the early 20th century, steam locomotives gradually gave way to diesel and electric engines, which offered greater power, reliability, and lower operational costs.

In the mid-20th century, cars and road transportation became increasingly widespread, especially in North America. Only in the late 20th century rail saw a resurgence, driven by growing concerns over congestion, fossil fuel dependency, and environmental sustainability. High-speed rail networks emerged, beginning with Japan's Shinkansen in 1964, followed by France's TGV, and later similar systems in Germany, Spain, Italy, China, and other countries.

Despite more than two centuries of evolution, the fundamental concept of railway transport⁷ has remained largely unchanged. The main differences lie in technological advancements, such as the transition from steam to electric and high-speed trains, the development of automated systems, and improvements in safety, energy efficiency, and passenger comfort.

⁵ as they were designed with the goal of transporting mining products, they had to be properly adapted to passenger and freight transport

France, for example, began constructing its first lines in the late 1820s, and by the mid-19th century, nations like Germany, Belgium, and Italy had launched their own rail systems. Rail networks soon expanded across colonies in Asia, Africa, and South America, often driven by imperial powers seeking to improve goods transportation and territorial control.

⁶ it is in fact worth considering that modern road vehicles had not been developed yet: trams represented one of the fastest way to move across the city, as the main alternatives were walking or animal-driven carriages. This is the case for early metros too.

After World War II, many European cities were rebuilt with the growing demand for automobile commuting in mind; however, most of them retained much of their historic urban layout, characterized by dense centers and mixed-use neighborhoods.

In contrast, many cities in the United States underwent extensive redevelopment, often adopting a car-centric approach that fundamentally altered their urban structure. Entire downtown areas were redesigned to prioritize highways, parking, and suburban expansion, leading to urban sprawl and a strong reliance on personal vehicles. This transformation continues to pose significant challenges today, as implementing efficient public transportation systems is far more difficult in environments that were not designed to support them.

⁷ i.e. vehicles running on guided tracks to move people and goods efficiently.

Advantages and disadvantages

From its inception, rail transportation has seen a rapid and widespread adoption due to its many advantages:

- Low friction between tracks and wheels: with respect to other means of transportation, less energy is dissipated and therefore power requirements are lower⁸. It is worth mentioning that, at least in some circumstances, this can represent an issue too, as force transfer between wheels and tracks can be impaired⁹.
- High load capacity: a single train can transport large amounts of passengers and cargo¹⁰.
- High energy efficiency: due to the low friction and the possibility of employing electrical traction, the efficiency of trains is considerably higher with respect to other transport systems.
- Low operating costs: while rail transport is usually very capital intensive, it is true that operating costs¹¹ are low.
- High speed transport: if a proper infrastructure is designed and realized, high-speed trains can allow for very fast transportation of passengers. High-speed railways are in fact a viable alternative to air transport, with comparable overall travel time¹².
- Eco-friendliness: the use of electrical propulsion allows for the use of renewable energy for traction. Even if the energy source is not renewable, though, electric propulsion systems avoid the need to release pollutants in their surrounding environments. Railways are therefore a valid solution for urban environments too: some examples are urban railways, metros and trams.

Its main disadvantages are typically caused by the need to properly construct railway tracks and infrastructure:

- Lack of flexibility: due to its nature, tracks are usually almost impossible to modify¹³: once a path has been defined, it cannot be changed. As railways are usually designed to last for long periods of time¹⁴, their surroundings could significantly change, altering their operation capabilities¹⁵.
- Influence of terrain and orography: rail transport has strict requirements in terms of slopes, curvature radii and in general track geometry. In harsh terrains¹⁶ the construction and operation of railway infrastructure can be very difficult and expensive.
- Investment cost: due to the resources necessary to build railway infrastructure, connecting areas that provide low passenger or freight traffic can often be economically disadvantageous. State subsidies are often needed to build and maintain railways in relatively remote areas that would otherwise be unprofitable.

⁸ at least with respect to transported load.

⁹ As an example, braking requires significantly more time and space than it would on an equivalent road vehicle. Another relevant consequence is a significant sensitivity to slope, the operating range of which is considerably restricted.

¹⁰ significantly more than many other means of transportation allow, assuming equal resources consumption.

¹¹ per transported passenger/cargo. As an example, road transportation is significantly more expensive in relative terms.

¹² While airplanes are considerably faster than trains, they imply additional time requirements not depending on the travel itself: airport-city connection, security controls, taxing to the runways. On the other hand, high-speed trains allow for quick boarding and main stations are usually located in the city center.

¹³ in many cases, the easiest solution is to build a new track from scratch.

¹⁴ sometimes in the order of hundreds of years.

¹⁵ A common example is represented by lines that have been built in the early days of railway development: train dimensions, loads and speeds were significantly smaller. Furthermore, as building and landscaping technology was much inferior in capabilities, older routes are typically more constrained to the orography of the terrain. As modern rail transport requires increased performance, many of these routes have been replaced by modern alternatives. Building new infrastructure is typically easier than adjusting and modernizing existing ones.

¹⁶ e.g. mountainous areas with steep changes in altitude.

Trains

Trains, which are collections of railway vehicles, represent the most widely adopted solution for railway transport. This approach allows for modularity, as each railway vehicle represents a functional unit. The layout, number and type of vehicles constituting the train depends on its operational objectives and conditions. A broad classification can be considered depending on various factors:

- **Hauling:** trains usually transport passengers or freight. Some mixed solutions exist¹⁷ but they may present compatibility issues
- **Weight:** a distinction is made between heavy and light rails. Heavy rails usually operate long routes transporting large amounts of goods and passengers. This is the usual option for regional, national and international transport. Light rails are mostly found in urban environments¹⁸ or in local transportation¹⁹. They usually are made up of a reduced number of vehicles²⁰.
- **Speed:** train speed can vary significantly: high-speed trains can often surpass 300 km/h by themselves, with track characteristics being their main limiting factor. Metros, trams and other urban trains are, on the other hand, usually limited to top speeds between 50 km/h and 100 km/h²¹.
- **Range of operation:** as previously discussed, trains can be destined to urban, short haul, long distance or even trans- or inter-continental transportation²². Requirements can therefore vary significantly²³.

¹⁷ known as 'mixed trains'.

¹⁸ e.g. in the form of trams, metros or urban railway systems.

¹⁹ e.g. metropolitan areas of large cities.

²⁰ In fact, trams and metros often do not exceed 3, with commuter trains being slightly longer. Clearly these are only reference values: many light rail solution exists and layouts can vary significantly.

²¹ usually even lower for trams.

²² e.g. Trans-Siberian railway, railways connecting the east and west coast of the USA, to some degree European railway corridors, etc.

²³ While passengers can stay in commuter trains for some minutes, some routes can take whole days/weeks to be traveled along.



Figure 2: A freight train and an high-speed passenger train operating on the same railway system. This is a common occurrence as railway infrastructure is typically shared between trains with different purposes.

Railway Vehicles

Usually trains (or *convoys*) are made up of many railway vehicles (also known as *rolling stock*). These can be broadly categorized as:

- **Motored rolling stock:** capable of autonomous motion/propulsion
- **Trailing rolling stock:** not capable of self-propulsion and requiring connection to other vehicles for movement
- **Electric Multiple Unit (EMU):** multiple-unit train. Electric traction motors are incorporated into one or more carriages, so no locomotive is necessary

Motored rolling stock

Motored vehicles are responsible for the vehicle motion. They are divided into *locomotives* and *railcars*.

Railcars

Railcars are self-propelled trains consisting of a single carriage/car with a driver's cab at one or both ends. Due to their relatively small size and capacity, they are usually employed on local/com-muter routes. They can also be connected to other vehicles to form small convoys, usually less than 5-6 units long.

Locomotives

Locomotives are vehicles able to haul a convoy. The traction power source can vary and the locomotive's structure, layout and components vary with it; the most common power sources are:

- **Electrical:** power is provided by an electricity source external to the locomotive²⁴. This is nowadays the preferred solution, as electrical propulsion is cleaner, less noisy and can also provide better dynamical characteristics. The worldwide trend is, in fact, towards a progressive electrification of rail lines.

While infrastructural requirements are more complex²⁵ and costly, operating costs are lower, higher speeds can be reached and, in general, reliability is improved.

Another relevant issue is that, across different countries, the electrification characteristics²⁶ can significantly vary: if no standardization effort is made, this may cause incompatibilities²⁷.



Figure 3: Ale501-Le220-Ale502 train, commonly known as 'Minuetto'; these employ railcars. They are also classified as Electric Multiple Units: often the distinction between trains containing railcars and EMUs is not clear cut.

²⁴ Various solutions exist for the connection between the power lines and the train: overhead line + pantograph, third rail, etc.

²⁵ due to the additional electric transmission lines.

²⁶ e.g. operating voltage, AC or DC lines, etc.

²⁷ It is worth mentioning that even single countries may have different electrification standards depending on the line. As an example, Italy adopts 25 kV AC for high-speed trains, while standard lines are powered at 3 kV DC).

- **Diesel:** power is generated by a Diesel engine. This is considered a relatively outdated technology when compared to electric locomotives. While being inferior in terms of performance with respect to electrified lines, Diesel locomotives are still employed on:
 - short or local lines transporting low passengers counts, for which the electrification of the line may not be worth the investment
 - lines across areas with different electrification standards, to avoid compatibility issues
 - difficult environmental characteristics or remote areas, where the installation of power lines could be problematic²⁸.
- **Mixed electric + diesel:** various solutions exist. The most common approaches are:
 - The Diesel engine is connected to a generator that powers electric motors. While still maintaining the drawbacks of Diesel propulsion, traction is delivered by electric motors that have better dynamic characteristics when compared to Diesel
 - The locomotive can both produce power by running its Diesel engine or by absorbing electrical power from external sources. This solution is typically adopted on lines that are not fully electrified/are currently under electrification.
- **Steam:** steam locomotives were the first to be employed, but they are no longer used

²⁸ This is typical of freight trains traveling across vast uninhabited and inhospitable regions.

The absence of electrification avoids the need to maintain and repair power lines, guaranteeing autonomy even in the event of adverse and extreme external conditions. In remote areas, damage to the electrical line may cause a train to remain immobilized until the connection is restored. This situation is not only inefficient but also poses significant safety risks.

The operation of electric motors is inherently complex, as it requires precise regulation of both thermal and electrical variables.

Their full power is often not exploited to prevent excessive thermal losses and overheating. Effective cooling and thermal management are therefore relevant to improve exploitation of the motor capabilities

Since speed and torque depend on input voltage and current characteristics (such as amplitude, frequency, and phase), modern power electronics are fundamental to enable accurate control. Before such technologies were available, diesel engines were preferred for their relative simplicity, but their use has declined with the advancement of control systems.



Figure 4: Examples of locomotives (from left to right, top to bottom):

Gruppo 685 'Regina' steam locomotive, employed from 1912 to 1962

E402 electric locomotive, employed on Intercity routes in Italy from the '90 to 2019

TRAXX electric locomotive; these represent one of the most recent locomotive designs

D145 diesel locomotive, most commonly employed as switch engine

The design, performance and capabilities of locomotives significantly depend on their duty and expected operating environment²⁹. Some locomotives, known as switch engines, are used in depots/stations to move other railway vehicles³⁰.

Due to their role, the performance of locomotives is very relevant and it affects the behavior and capabilities of the whole train.

²⁹ To provide some examples, they could be designed to reach high speeds, to have a high carrying capacity, to work on steep inclines or rough terrains, etc.

³⁰ usually to assemble/disassemble trains.

Trailing rolling stock

Trailing vehicles are not capable of autonomous motion and must therefore be connected to locomotives or railcars. They are employed to contain and transport goods and passengers. The trailing rolling stock nomenclature is quite diverse and many terms can be used to refer to the same vehicle in different English-speaking countries. Different terms will therefore be presented if necessary.

Passenger cars

Passenger cars³¹, as the name suggests, are dedicated to the transportation of passengers. They are present in a variable number in trains, usually depending on the forecast passenger traffic. Due to their role, they must be designed to guarantee passenger comfort: they usually offer seats³², toilets, air conditioning, baggage areas and other amenities.

An extremely wide range of designs exists for carriages, usually heavily influenced by their expected operations³³. Dining and sleeping cars are usually present in long routes, while local routes are usually served by simpler and less furnished passenger cars. A gangway connection is typically³⁴ present at the ends of the car to allow passage and communication between carriages. This connection is realized in such a way that passengers are not exposed to the outside or to dangers while passing from one carriage to another.

As they are designed to transport passengers, their dynamical characteristics should be properly evaluated too. In particular, the comfort of the occupants should be guaranteed by not exposing them to high and sudden accelerations or to the effects of track irregularities. Suspension systems must therefore be carefully designed and have a significant effect.

Freight cars

Freight cars are designed for the transportation of goods. Due to the many types of goods that can be transported, the number of freight cars layouts far surpasses the already large passenger cars' one. Some examples are³⁵:

- *General purpose cars*, both open and close
- *Controlled temperature cars*: typically used in a refrigerating role to transport food and other temperature sensitive goods. It can serve an heating purpose too³⁶.
- *Hopper cars*: very common for the transportation of raw materials, especially bulk goods³⁷.
- *Tank cars*: dedicated to the transportation of liquid goods, typically oil and its derivatives³⁸. Some variants of tank cars exist for the transportation of fine powders or similar materials³⁹.

³¹ also known as passenger *coaches* and passenger *carriages*.

³² although in some cases passengers are expected to stand for the whole duration of the trip; this is most common in commuter transport and it is not the case for long distance travels.

³³ Some very specific and exotic applications exist. To provide an example, in Russia prisoner cars are used to transport prisoners from courts to prisons and vice-versa. In general, it is not trivial to provide an exhaustive list, due to the vast amount of options.

³⁴ if not always

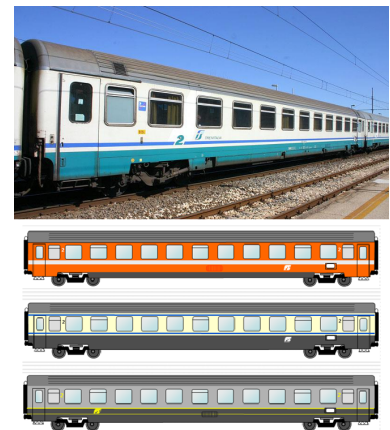


Figure 5: UIC-Z passengers cars have seen a widespread adoption in Italian railways

³⁵ not at all representing an exhaustive list.

³⁶ as an example, to avoid freezing of materials that are sensitive to low temperatures in winter or cold external environments.

³⁷ e.g. coal, mining products, etc.

³⁸ gasoline, petrol and other fuels. They could also be used for liquid alimentary goods, such as milk, yet this is a rarer occurrence.

³⁹ e.g. sand, but even cereals

- *Container cars*: simple cars, as they only work as a support for containers
- *Ultra-low cars*: employed to transport goods that have a significant vertical extension



Figure 6: Examples of freight cars:
Left: standard open freight car
Right: hopper freight car

Trailers

Trailers are smaller than freight cars and are usually employed for the transportation of goods whose size does not require a freight car.

Electric Multiple Units (EMU)

In EMU, traction is distributed along the convoy, instead of being localized in one or two locomotives. This solution is often adopted in commuter and suburban transport, yet some intercity EMUs exist too. A notable exception are high-speed trains: the EMU layout is commonly adopted for these, to provide a more homogeneous power distribution and delivery along the train. Except for a few examples, they are mostly used for passenger transportation. EMUs are composed of:

- A **power car**, to draw power from external electrical infrastructure. They usually are provided with the necessary power electronics and electrical systems to convert, adapt and distribute electrical power to the whole train
- **Motors cars** contain the electric traction motors
- **Driving cars**, usually one at each end, allow to control the train
- **Trailer cars** do not provide traction and are comparable to passenger cars in standard trains

Usually EMUs are permanently or, more frequently, semi-permanently coupled.



Figure 7: Trailers are typically adopted when whole freight cars are not required



Figure 8: An ETR 425 train; these EMU are widely adopted by Trenitalia and Trenord to serve short-range routes



Figure 9: An ETR 1000 Frecciarossa train; these represent an example of high-speed EMU

Railway Bogies

While the overall layout of railway vehicles can vary significantly, they are typically composed of two primary subsystems: the *main body* and *bogies*.

Main body

The main body, also known as car body, forms the central structure that carries passengers or cargo. It mostly serves a structural function and its features strongly depend on their expected operations:

- *Passenger car* bodies are typically built with a focus on comfort, safety, and accessibility. They often feature windows, seating arrangements, restrooms, and climate control systems⁴⁰. These bodies also house essential systems like air conditioning units, lighting, and sometimes onboard entertainment or communication equipment. Additionally, passenger cars are designed to carry and distribute vital utilities such as electricity for lighting and outlets, compressed air for braking systems, and data cables for operational controls.
- *Freight car* bodies are primarily engineered for durability and load efficiency, with less emphasis on comfort. Their shape depends on the type of cargo they are destined to and can vary significantly. While they generally lack systems like air conditioning or internal lighting, cargo cars still require mechanisms to distribute braking power and, in some cases, they may also incorporate electrical systems for monitoring and communication.

⁴⁰ in general, they are designed to be thermally controllable. This is not only limited to air conditioning systems: as an example, proper thermal insulation is required too and must be properly accounted for in the body design.

Bogies

Bogies⁴¹ are wheel-carrying subframes mounted beneath the main body of the vehicle. They typically consist of two or more wheelsets⁴², a suspension system, and sometimes braking and traction equipment. Their primary functions are:

- supporting the weight of the vehicle
- guiding the vehicle along the tracks
- absorbing track irregularities
- allowing for smooth cornering and improving ride stability.

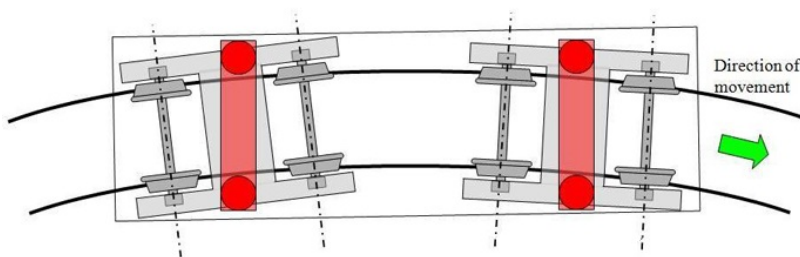
⁴¹ also known as trucks in North American English.

⁴² combination of an axle and two (or, unusually, more) wheels.

Bogies are used in place of directly connecting wheelsets to the car body for several important reasons:

- Since the bogie can swivel independently, the vehicle's ability to navigate curves is improved. If the axles are directly connected to the body, the length of the vehicle is limited by the radius of the curves it is expected to encounter during operations. Introducing bogies eliminates, at least theoretically⁴³, this limitation. Lateral forces and stresses on the wheels are lower too, reducing wear and fatigue phenomena on these components
- Bogies distribute more evenly loads across the axles, enhancing stability and allowing the vehicle to carry heavier cargo or an increased amount of passengers
- The inclusion of suspension systems can significantly improve the overall dynamic performance of the vehicle. As shocks and irregularities are absorbed, passenger comfort is improved and stresses on the vehicle's structure are reduced. As forces exchanged between the wheels and the rails are reduced, tracks are less stressed too, improving their operating life
- Bogies allow for a modular approach: they can be substituted, removed or in general modified without altering the car body itself. This is also useful from a maintenance perspective⁴⁴.

Due to the numerous advantages of their implementation, bogies have been adopted from the early days of railway transportation. The first locomotives⁴⁵ typically had rigid frames with fixed wheelsets directly attached to the body. These early designs worked well for short, relatively straight tracks and lighter loads. As railways expanded and trains were required to run longer routes and to face sharper curves and uneven tracks, bogies started to be introduced and quickly became the standard as their advantages became clear.



Jacobs bogies

Jacobs bogies are a type of railway bogie that is shared between two adjacent carriages, effectively linking them together. This approach is commonly found on articulated trains⁴⁶ and tramway vehicles.

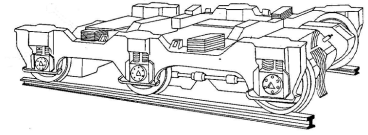


Figure 10: Locomotive bogie: three axles are adopted as locomotives require heavier loads

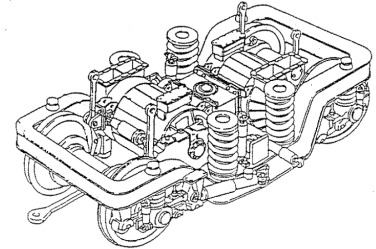


Figure 11: ETR500 bogie: two axles are sufficient to its passenger transport purpose; the suspension system is significantly more complex to increase ride comfort

⁴³ clearly other constraints exist on the vehicle length.

⁴⁴ If a bogie has to be maintained, it can be removed from the vehicle and taken to another place for maintenance, while being replaced with another bogie.

⁴⁵ such as George Stephenson's Locomotion No. 1 (1825) and Rocket (1829).

Figure 12: The introduction of bogies allow the train to better follow the tracks. As the wheel-track interaction is smoother, contact forces and stresses are reduced. The mechanical link (red) between the bogie and the main body allows for free rotation while transmitting traction.

⁴⁶ The most notable example are high-speed trains: Jacobs bogies are typically adopted due to their excellent dynamic performance.

The main advantages of Jacobs bogies are:

- Reduction in the total number of required bogies⁴⁷. Vehicle weight and maintenance requirements are therefore reduced⁴⁸.
- Improved dynamical performance: ride stability and passenger comfort are enhanced, particularly at high speeds or around sharp curves⁴⁹.
- Safety is increased, as the semi-permanent and shared nature of the connection between vehicles reduces the risk of car separation and jackknifing in the event of a derailment.

The main disadvantages of their adoption are:

- More complex designs may be necessary, as they require two connection systems⁵⁰.
- Reduced flexibility, as the vehicles are semi-permanently coupled and can only be separated in appositely equipped workshops⁵¹.

Bogies features and characteristics

Depending on their purpose and operating requirements, bogies can present different features and characteristics, also allowing for a broad classification of their layouts.

Layouts chosen for wheels and their respective axles and traction systems are known as *wheel arrangements* and significantly influenced by regulations and overall vehicle requirements⁵². The vehicle type is relevant too: a locomotive for a high-speed train clearly requires a different layout with respect to a local transport passenger car.

Different notation schemes exist to describe wheel arrangements. They usually refer to the whole vehicle⁵³. One of the most widespread notation systems is the UIC⁵⁴ classification⁵⁵. While this notation is commonly adopted to describe whole vehicles, it is suitable to describe single bogies too.

The UIC/AAR notation

In the context of UIC/AAR notation, each wheel arrangement can be described by an alphanumeric string, following a well-defined convention:

- Each bogie of the vehicle is separated by a '-'
- The string represents the bogie disposition from front to rear
- Numbers represent unpowered axles in a bogie
- Letters represent powered axles in a bogie
- If axles are independently powered, a 'o' is added to the letter indicating their amount



Figure 13: Example of a Jacobs bogie

⁴⁷ Conventional passenger cars typically employ two bogies. If bogies are shared between two carriages, two half bogies are required for each. The amount of bogies is therefore approximately halved.

⁴⁸ This can be an issue too: loads are distributed on a reduced amount of bogies; loads per axle can therefore be significantly higher. Jacobs bogies are therefore better suited for light-weight applications or passenger trains.

⁴⁹ This is the main reason for their adoption in high-speed trains.

⁵⁰ One for each vehicle they are connected to.

⁵¹ This solution is therefore commonly adopted when the composition of the train is not often change; this is therefore a viable option for high-speed trains.

⁵² A simple example: maximum loads per axle are regulated. The higher the load the vehicle has to carry, the higher the number of axles to distribute its weight.

⁵³ therefore including multiple bogies.

⁵⁴ International Union of Railway; the acronym derives from its French name *Union Internationale des Chemins de fer*.

⁵⁵ in North America, a simplified version, known as AAR, is adopted.

A progressive alphabetic order is adopted for powered axles: A = one powered axle, B = two powered axles, and so on.

Examples of wheel arrangements

A large variety of wheel arrangements has been developed through the long history of railway transport, each typically designed for specific operating requirements. Some examples are:

- **2-B:** two bogies; the front one is unpowered, while the rear one provides traction. The powered axles are connected by the transmission system to the same motor, therefore not being independent from each other. This solution is not symmetric with respect to travel direction; such asymmetric solutions are usually adopted for vehicles that have a preferred direction of motion, such as locomotives.
 - **A1A-A1A:** two bogies with the same wheel arrangement. Two powered axles connected to the same motor, one at the front and one at the rear of the bogie, and a central unpowered axle. Three-axles solutions are more stable and usually adopted in high loads or high speed designs. Compared to two-axles layouts, they are more stable and allow for a better weight distribution, at the cost of reduced flexibility in turns. The idler wheels can be smaller than the powered ones.
 - **B-B:** this configuration is nowadays popular for high-speed/low-weight designs, as intermodal trains or high-speed railways.
 - **Bo-Bo-Bo:** three bogies; each bogies has two axles and each of these is powered by an independent motor. This solution has been extensively adopted on Italian and Japanese railways. Significant play should usually be provided to the center bogie to allow for proper behavior in curved sections of the track.
 - **C-C:** usually two identical bogies (not only in terms of wheel arrangement but as overall design). Each bogie has three powered axles, powered by the same motor. This layout is usually found in high-weight/low-speed vehicles; a typical example is represented by freight cars. The Co-Co configuration is sometimes adopted too.
- As previously discussed, three-axles bogies allow for better stability and weight distribution but reduce flexibility and speed.
- **D-D:** Two bogies, each comprising four powered axles. This solution has been adopted in the past for locomotives, but has fallen out of favor: it is usually significantly more complex and difficult to manage than two- and three-axles layouts, without significantly improving their performance.

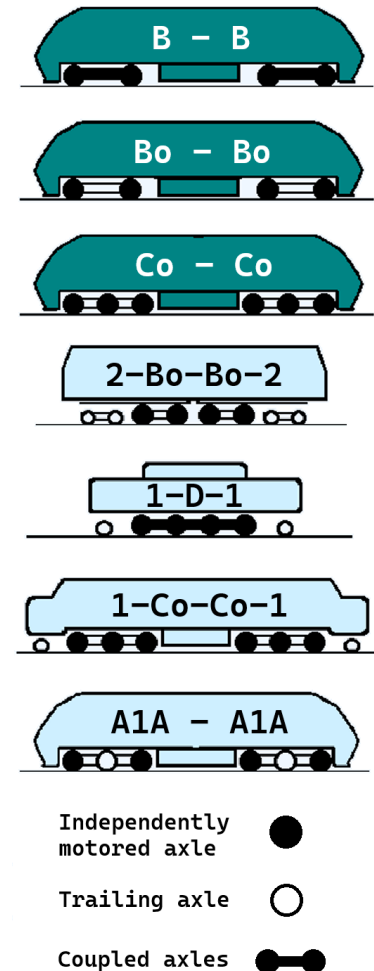


Figure 14: Examples of wheel arrangements; some of these are briefly discussed.

The classification of bogies is strongly connected to wheel arrangements. The most relevant bogie features are in fact:

- **Traction systems** or lack thereof: as previously mentioned, bogies can provide traction ⁵⁶ or just have a structural support function. In motorized bogies, axles can be independent or linked to the same motor; as shown in examples, not all axles necessarily provide traction even if overall the bogies does.
- **Number of axles** and overall **wheel arrangement**: most commonly, bogies adopt either two- or three-axles configuration. Four-axles configuration are rare and are usually not as effective, and therefore avoided.

⁵⁶ motorized bogies

- **Steering:** the orientation of axles with respect to the bogie can either be fixed or (at least partially) unconstrained. In the latter case, the bogie is able to adapt to curves it encounters and can travel through these without significant limitations. This solution provides better performance but it is more complex, both from a design, production and maintenance perspective.

If the axles are instead fixed, though, a minimum curve radius exists to allow for proper operation, determined by the geometry of the bogie. Despite this limitation, this solution is cheaper and simpler.

Bogies components and architecture

Since the design process of bogies is the objective of this project, understanding its components and architecture is most relevant. The most significant and common elements will therefore hereby be presented.

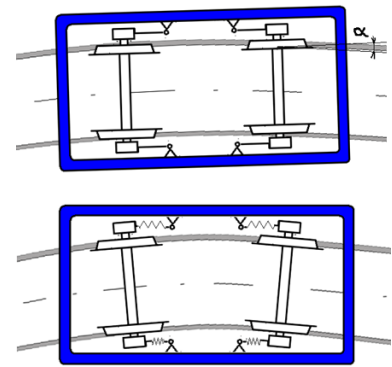
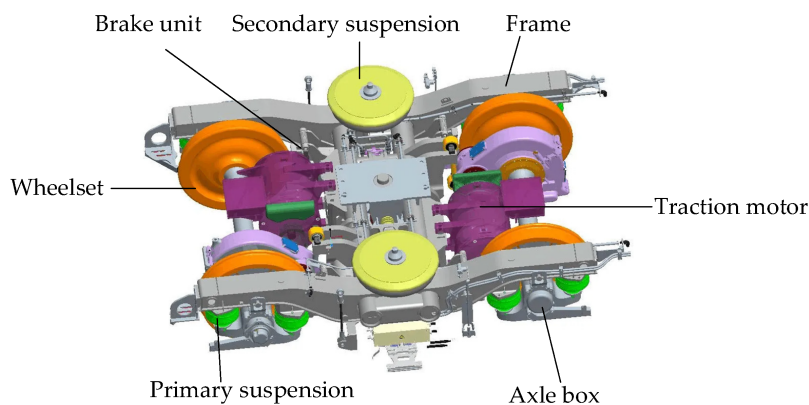


Figure 15: Non-steering (up) and steering (down) bogies. The steering bogie is able to follow the tracks more easily.

Figure 16: Typical bogie structure and main components.

The frame provides structural support to the bogie and is connected to the car body by the secondary suspensions.

Primary suspensions and their respective mechanical supports and connections provide the connection between the wheelsets and the frame.

Traction motors, if present, can either be directly connected to the wheelset or provide power through transmission systems.

Motors could even not be part of the bogie and transfer their output power through more complex transmission systems.

Bogie frame

The bogie frame serves a structural function, representing the supporting element of the bogie itself. Due to its role, it must be designed to properly withstand the loads and stresses encountered during operations.

In this project, its layout and geometry represents the main focus of the design activity.

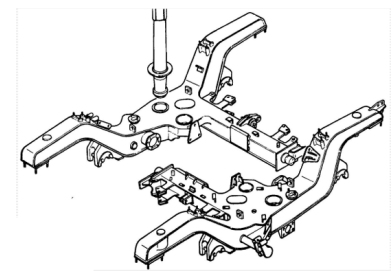


Figure 17: Bogie frame adopted on ETR450. Its two-part structure is highlighted.

Wheelsets

Wheelsets are pairs of wheels rigidly mounted on an axle. Their individual components are:

- **Wheels** represent the interface between the vehicle and the ground. They usually are designed and realized with a conical shaped thread and a flange. Many geometrically well-defined wheel profiles exist. The thread is the part of the wheel that is in contact with the track in normal operations, while the flange prevents derailment under exceptional circumstances⁵⁷. The conical shape of the thread allows for easier travel through curves. Contact between wheels and track is a complex phenomenon to model and some aspects of it are not fully understood or well-described to this day.
- **Axle:** supported by bearings, it allows for connection between the wheels and the bogie⁵⁸. As it is subject to all loads that are transferred from the vehicle to the ground and vice-versa, it must be properly designed not only from a static perspective, but considering fatigue too⁵⁹.
- **Axle journals:** wheels are usually not at the extremities of the wheelset⁶⁰. Typically, the axle extends beyond the wheels to allow for connection to bearings and thus to the non-rotating part of the bogie/vehicle.

The distance between wheels is known as gauge. To be compatible, wheelsets must be designed for the gauge of the track they are destined to.

Gauge compatibility issues

From an historical perspective, the development of railway networks has been driven by national or even local priorities, often with little regard for future international integration. Each nation developed its own standards and even across a single nation, different systems have been implemented.

In some cases, this has been a deliberate choice: the use of railway vehicles for the military invasion of a neighbouring country was made very difficult, if not impossible, by the different standards.

Unfortunately, nowadays this fragmentation represents a significant obstacle to the international transportation of passenger and goods between nations. While other standardization differences have been overcome either by later uniformation or by technological adaptation, the difference in gauges between countries represent to this day a very difficult discrepancy to solve. While solutions have been proposed and designed, no definitive remedy has been found.

⁵⁷ e.g. excessive displacement of the wheel due to extremely sharp corners.

⁵⁸ or directly to the vehicle if bogies are not employed.

⁵⁹ Due to the rotational nature of their operating conditions, axles are usually exposed to periodic and alternating stresses, which are very significant in the context of fatigue.



Figure 18: Example of wheelset; in this case, brake discs are present too.

Italy, as an example, continues to operate some local networks, such as the Circumvesuviana, that employ a much narrower gauge and are therefore not compatible with the national railway systems.

Electric power delivery, signaling and similar compatibility issues can typically be solved by later infrastructure and rolling stock updates. As an example, the electric power input can usually be adjusted by the use of properly chosen power electronics.

This is not true in the case of track gauge: a more profound restructuring would be necessary.

To avoid a complete infrastructure renovation, solutions such as wheelsets that are able to change their gauge have been proposed, yet they have seen very limited adoption.

Bogie - Wheelset connection

While a wide variety of solution have been devised to realize the mechanical link between the bogie frame and the wheelset, usually two main components are employed:

- An **axlebox** is present at each extremity of the axle. It contains bearings, thus transferring the vehicle weight and other loads to the wheelset.
- **Primary suspensions** represent the first layer of suspension system in the overall vehicle-ground interaction. They influence the dynamical behavior of the bogie and their objective is to absorb shocks and track irregularities. Coil springs are typically adopted due to their compactness, yet other solutions exist.

In some cases, the axlebox is connected to the bogie frame by the primary suspensions, with no direct connection; in other designs⁶¹ both the axlebox and the springs are connected to the frame.

Bogie - Body connection

As in the case of the bogie - wheelset connection, many solutions exist. That said, the most typical approach⁶² is to employ a structural component⁶³ and a dynamic component⁶⁴. The two main elements therefore are:

- A **traction link** is responsible for the transfer of traction forces between the bogie and the body⁶⁵. This should not be a rigid connection, though, as the bogie should be capable of independent motion, especially on the vertical axis⁶⁶.
- **Secondary suspensions** are realized through a wide variety of solutions. In fact, many types of spring can be employed⁶⁷:
 - *Coil springs*
 - *Leaf springs*: made of metal 'leaves', they can provide some damping too, due to the friction between leaves. They are usually larger than coil spring⁶⁸ and must be correctly maintained and lubricated to achieve good performance.
 - *Air/pneumatic springs*: the elastic properties of air⁶⁹ are employed. This type of springs are usually employed in combination with other systems and are remarkably non-linear in their behavior. They can provide superior ride comfort and other dynamical advantages but are usually more complex to design and require dedicated pneumatic circuits.
 - *Rubber elements*: as in the case of air springs, they are usually combined to other devices⁷⁰. Due to their viscoelastic behavior, their dynamic response is usually complex to evaluate. Rubber elements usually provide both an elastic response and damping/energy dissipation.

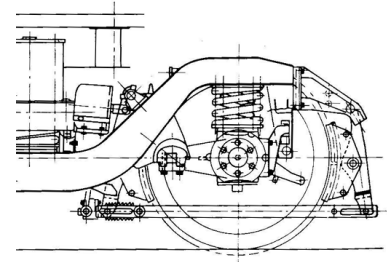


Figure 19: Example of primary suspensions. The axlebox and the coil spring are visible.

⁶¹ this is the case for the suspension design that will be considered in this project.

⁶² as seen for bogie - wheelset connection.

⁶³ the axlebox in the previous case.

⁶⁴ springs/primary suspensions in the previous case.

⁶⁵ these can act in both directions: in the case of a non-powered vehicle, traction is provided by the connection to other vehicles; traction is therefore transferred to the bogie by the body. On the other hand, during braking or in the case of motorized bogies, traction/braking forces are transferred from the bogie to the body.

⁶⁶ a rigid connection would defy the whole purpose of bogies: axles would basically be connected and fixed to the body.

⁶⁷ in the case of primary suspensions, dimensional constraints are usually more significant and coil spring are preferred due to their reduced size. This is not the case for secondary suspensions.

⁶⁸ to achieve the same results.

⁶⁹ or other gases.

⁷⁰ e.g. rubber elements in coil springs.

Dampers

Dampers are part of both primary and secondary suspension systems. They influence the dynamical behaviour of the vehicle and are commonly employed to reduce the amplitude of vibrations and to absorb shocks. The dynamical characteristics of railway vehicles and thus the effect of dampers will be explored in more detail in later sections.

Other components

Brakes and traction motors are a common occurrence in railway bogies too⁷¹.

Braking is typically realized by the addition of brake discs to the wheelset. The static component of the braking system is usually connected to the bogie frame.

While this is a commonly adopted solution, many options exist and can significantly vary in their realization.

Traction motors, when present, can transfer power to the wheels in different ways, depending on the bogie's wheel arrangement and its functionalities. Some possible solutions are:

1. The traction motor is directly connected to the wheelsets. This approach has the advantage of requiring a comparatively simple transmission, at the cost of exposing the motor to vibrations and shocks due to the interaction with the track
2. A single traction motor provides power to more than one axles through a transmission system. This setup is typically more complex than the previous solution.
3. The power unit could even not be part of the bogie, being in the car body instead. This is a common occurrence in locomotives. This approach allows for the use of large motors⁷² at the cost of considerably more complex transmissions.

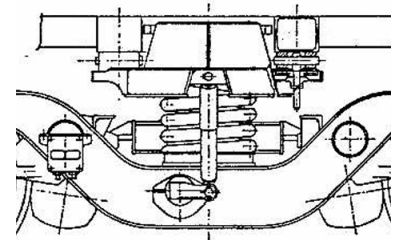


Figure 20: Coil spring based secondary suspension. Vertical dampers are visible too.

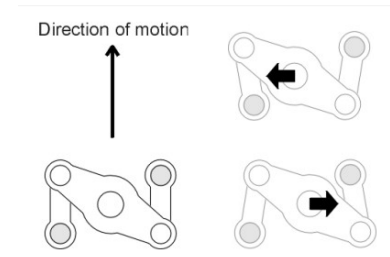


Figure 21: Watt's linkage: this is a commonly adopted solution to realize a mechanical connection between bogie and body able to transfer traction loads while still allowing for relative movement. Dampers are commonly introduced in the link too, with the goal of smoothing oscillations and vibrations that may be transferred to the main body from the bogie.

⁷¹ Both braking and traction systems represent broad topics worthy of extended analysis and study. As they are not considered in the project, a complete description of their working principles and implementation would exceed the scope of this analysis of railway bogies.

⁷² this may be necessary if traction power requirements are significant.

Part II

Introduction to the Project

Foreword

The overall objective of this project is to demonstrate the validity and feasibility of the integration of generative AI in the preliminary phase of the design process. While this objective could be reached from a theoretical perspective only, a demonstrative practical implementation has been deemed necessary to properly showcase the main features of a concrete realization of this approach.

The evident challenge in providing a demonstrative example is represented by the extremely large variety of systems and component that could be the objective of design processes. In fact, even among the same type of systems, characteristics and designs could considerably vary⁷³.

The focus of this project is therefore not on exhaustively analyzing all possible implementations of the concept, but on providing a working example, showcasing the overall workflow and features of the design tool. The development of the latter has therefore been carried on with a general scope in mind, being conceptualized to be adaptable to different design objectives⁷⁴.

As mentioned, the development of a tool to aid in the preliminary phase of railway bogie design has been considered as a valid option. Railway bogie are complex systems that involve a wide range of phenomena to be properly modeled and features to be implemented. Some examples are:

- Generation of complex tridimensional structures
- Non-linearities and articulated suspension systems
- Static analysis under different loading conditions
- Complex dynamic behavior

The preliminary design of railway bogie has therefore been considered as representative of many structures and components that may be subject of a design process. Techniques and approaches that are developed in the context of bogie design could often be easily adapted to other mechanical systems⁷⁵.

In line with the overall theme of this project, the solution that has been developed for the integration of generative AI in the design process will be discussed from a theoretical perspective first ⁷⁶; its practical implementation will then be presented.

⁷³ As previously discussed, this is the case for railway bogies too: many design parameters (e.g. the number of axles, suspension types, size, materials, etc.) could be varied.

⁷⁴ Clearly, adjustments are necessary, as designing different system could require vastly different approaches. Nevertheless, developing a flexible framework allows to reduce both time and effort that have to be spent to adapt the concept.

⁷⁵ This is a common theme for the whole project. While it is not possible to devise strategies valid for every system, some general considerations, guidelines and insights can be provided.

Due to their relatively complex nature, railway bogie allow to showcase how to model and implement characteristic features of many mechanical systems.

⁷⁶ As mentioned, the objective is to provide a general approach. The theoretical discussion allows therefore to present the adopted solutions from a wide perspective. A practical example allows then to clarify how such a general approach can be adapted to much more specific solutions.

Comparison of different design procedures

To properly describe the advantages and features of the AI-based approach that will be developed in this project, it can be useful to compare it to more conventional design procedures.

Traditional design

In traditional design, the use of computers and other calculation devices is limited to the analysis and verification of candidate solutions: the conceptual development and design definition⁷⁷ is carried out by a human user. Computers therefore provide a numerical aid to the conceptual/projectual phase⁷⁸ and in the numerical analysis of the resulting design⁷⁹, yet they do not actively contribute to the conceptualization and generation of the structure/component, which is left to the designer. The typical workflow for this approach is:

1. Determination of requirements and constraints
2. Design of a possible solution
3. Analysis of the solution
4. Correction of the design (if needed) considering analysis results
5. Final verification and validation

Generative optimization

The adoption of optimization in design generation includes a vast collection of techniques and approaches to find the best performing solution for the design of a system or component. The scope of optimization can significantly vary: in some cases a single parameter may have to be determined; in other cases, the whole layout of a system may have to be determined.

In spite of their differences, some features are typically adopted by most optimization algorithms:

- An iterative process is adopted. Each iteration refines the current solution until a satisfactory result is achieved.
- A cost/fitness function is introduced: it provides a numerical description of the design performance.
- The objective of the algorithm is to, iteration by iteration, maximize or minimize the fitness/cost function.

Mechanical design is a complex process and, depending on its objective, adopted procedures can significantly vary. The objective of this section is therefore not to present every design technique but to provide a reference benchmark.

The approaches to design that are presented are described from a broad perspective, but variations and differences can be found in practical applications. In fact, even in this project, the GUIDE approach is not implemented exactly as described.

⁷⁷ sizing, choice of materials and layout, choice of components/subsystems (e.g. suspension systems in bogie design)

⁷⁸ e.g. CAD software

⁷⁹ e.g. FEM analysis tools

1. Requirements can be provided by the customer, decided by the designer or depend on engineering constraints.
2. The solution must be compatible with the requirements. The design itself can range significantly in its scope (e.g. from sizing only to the conceptualization of the whole design from scratch).
3. Depending on the scope of the design process, the type of analysis to be performed can significantly vary. In general, though, its objective is to evaluate the performance/fitness of the proposed solution.
4. This phase represents the core of the iterative nature of the design process: the design is corrected and the process is repeated until a satisfactory result is achieved.
5. If a final design proposal is obtained, it still has to be tested and verified (e.g. with physical realization and testing). Issues emerging in this phase can lead to further correction to the design.

The definition of a valid cost function is extremely important to obtain proper results, and in some cases the conversion from more abstract concepts to mathematical requirements could not be straightforward. If the fitness/cost function is not properly representative of the desired design features, results could significantly deviate from expectations.

If only a few parameters are to be determined, optimization typically involves relatively simple numerical techniques to maximize or minimize the fitness/cost function. These solutions are typically adopted when the overall layout of the design is known and only some of its characteristic parameters are to be determined.

If more complex systems are considered, such algorithms become insufficient. As the number of parameters increase, evaluating each configuration becomes impossible. Other solutions, such as genetic algorithms, are therefore adopted. This approach could be referred to as *generative optimization*: the algorithm is not only searching for an optimal solution but generating a viable layout for the whole design.

Considering the objectives of this project, generative optimization is a better reference for comparison⁸⁰. Its typical⁸¹ workflow is:

1. Determination of requirements and constraints; they also have to be expressed in a mathematical form to be handled by the algorithm
2. Algorithmic generation of a candidate solution; remarkably, as the solution is generally devised by the software, a procedure to generate it must be coded. This could be difficult to realize for complex systems, as designing a viable option may not be trivial and require human experience and knowledge
3. Evaluation of the cost function for the solution⁸²
4. Feedback: in the case of iterative optimization, the results of the previous step are then used to adjust and improve the previously generated solutions. The process is then repeated until a satisfactory result is obtained.
5. Final verification and validation⁸³

Remarkably, the workflow is not significantly different from traditional design. The main difference between the two is that optimization is typically autonomously performed by software and does not require considerable user intervention.

The main advantage of this approach is that it can be fully automated: the user can provide the constraints and objectives and a final design is obtained. If the software implementation is well-realized, it can also be much faster than alternative methods. Furthermore, optimization algorithms can, in the same interval, explore a considerably larger set of configurations with respect to a human user.

Typically, research and designer experience are required to properly define the cost/fitness function in the case of complex optimizations (involving many parameters).

If a large amount of parameters are to be determined, the cost function should take in account many characteristics of the final design, often in opposition one to another (e.g. a lighter structure can be beneficial, but its structural resistance may decrease; stability in a bogie may come at the cost of decreased performance in curves, etc.).

Furthermore, it may not be easy to define from a numerical perspective what a 'good performance' is. (To provide a practical example, considering the static analysis of a structure, it may not be easy to define from a numerical perspective what a desirable solution is: the structure may have small displacements yet high stresses, displacement could be present where they should not, it may be difficult to define what a 'good' deformed shape is, etc.).

⁸⁰ As it involves mostly automated generation of designs, it provides a more useful benchmark to evaluate the solution that is developed in this project with respect to a fully human-based approach such as traditional design.

⁸¹ The actual implementation may vary on a case-by-case basis

⁸² or the solutions: many possible options may be generated and evaluated at the same time, then compared and selected. As an example, this is the case when genetic algorithms are employed.

⁸³ Depending on the objectives of the design process, the verification phase can either be automatically performed by the software or carried out by the user.

If a numerical problem is considered (e.g. only a numerical model of the system is studied and developed), the verification process can be performed by the software and it is often already included in it.

If physical systems are designed, often further confirmation and validation is required by the realization of prototypes and their consequent testing. The involvement of human users is therefore required.

As previously mentioned, the main drawback of generative optimization is the proper definition of a cost function: while in some cases a straight-forward mathematical relation could be obtained, in other circumstances other factors that are not purely numerical in nature could be relevant.

In general, a typical issue with fully automated generation is the dependence of some design choices on designer experience or, more generally, non strictly numerical considerations. These may be difficult to precisely translate to numerical definitions and may lead to unsatisfactory results⁸⁴.

The GUIDE framework

The *Generative User-Interactive Design Environment* (GUIDE) framework introduces the intervention of an AI capable of structure generation in the design process.

It has been developed from the ground up⁸⁵ for this project and demonstrating its features and capabilities is in fact the main objective of the latter. In fact, the complex objectives of the latter and the restrictive constraints in the AI implementation have required the conceptualization of the GUIDE approach and of its AI component. Its features can be better understood by examining the workflow that is adopted:

1. Requirements and constraints are understood by the AI and implemented in the designs it produces.
Parameters of the layout can be preliminary set or left to be determined during the design phase. In this phase, initial conditions for the AI training can be imposed too⁸⁶.
2. Generation of a candidate solution by the AI: it generates a solution that is coherent with the constraints that have been imposed
3. Human interaction: the user can interact with the design that is proposed by the AI. The latter is then able to learn from the user's actions. The user can:
 - Modify the structure and its free parameters
 - Perform analysis on the design⁸⁷
 - Provide feedback on the proposed design⁸⁸.

The user can flag the proposed solution as desirable or invalid. The AI is updated to prefer and avoid such solutions respectively.

If the current design is deemed to be valid enough, it can be saved and stored for later use or for exportation. In the same way, a state in the training process of the AI can be saved for later use⁸⁹.
4. After learning from the user feedback, the updated AI generates a new design that is again presented to the user, repeating the procedure from 2.

⁸⁴ As an example, considerations regarding the manufacturing and the cost of components are often relevant. The ease of manufacturing a certain structure/system is not easily mathematically defined and it is often evaluated by the judgment and experience of the designer.

Business choices, aesthetic considerations, product user experience and other similar factors may be difficult to describe numerically too.

⁸⁵ No similar generative tools have been found to be freely available.

Certainly proprietary solutions may have been developed, yet these are not publicly available and may be integrated in non open source commercial software.

The GUIDE approach therefore represents a completely independent solution and has been developed without the use of any external software or by adopting pre-existing techniques.

⁸⁶ By default, the AI is initialized with no preset training.

To avoid re-training the model from scratch when unnecessary, previously saved training configurations can be loaded and set as starting conditions. Due to the flexibility of the system, it is also possible to appositely design and load starting conditions to achieve specific configurations without having to train the AI for it.

⁸⁷ in this project, static analysis will be considered, as it represents one of the the most common occurrence. Other analysis tools could be implemented if necessary on a case-by-base basis.

⁸⁸ either after having edited it or directly as proposed by the AI.

⁸⁹ this can be considered similar to a 'checkpoint' in the training process, to be saved for later use.

Despite the apparent similarity to generative optimization, the workflow is not as straight-forward and significant conceptual differences exist:

- While the optimization process is fully automated, the GUIDE approach requires the intervention of a human user.
While this can be considered a disadvantage in terms of time requirements⁹⁰, it can be advantageous when human experience and knowledge are relevant or when the design is too generic to be obtained by optimization⁹¹.
- AI can take some time to train⁹² but once this is done, the trained AI can be saved and later reused to explore similar layouts. To save time on the training, if a broad concept for the layout has already been conceived, it can be preliminary added to start the generation from similar layouts.
- Differently from optimization, the AI is also able to understand broader concepts that are not strictly numerical in nature⁹³.
- The goal of optimizations is to produce a final output design; this is not the case for the GUIDE approach: as the name suggests, its goal is to support and guide the user in its own creative process. In particular:
 - The generation process does not have a predefined end: the user can iterate and explore layouts, save ones that are considered good and go back to previous setups if desired.
In general, the GUIDE approach is not limited to a 'linear' process and it is more similar to the organic nature of conventional design
 - The integrated analysis tools allow the user to easily and quickly modify the structure and analyze different layouts.
While results are clearly preliminary and not necessarily coincident with the final design, it will be demonstrated in later sections that they are quite accurate and representative; this is particularly useful to avoid from the beginning solutions that are not satisfactory⁹⁴.
 - Designers can have biases towards solutions that they know or have already implemented. The AI can generate solutions that are coherent with what desired by the user, while introducing a certain degree of randomness and autonomously developed insight.
This could therefore enhance the creativity of the user by exploring solutions that would not have been considered otherwise, while providing reasonable and valid designs.
 - With respect to optimization, the focus of the GUIDE approach is mostly on preliminary design⁹⁵. In fact, optimization can be later performed on designs produced by GUIDE. The two approaches are therefore not antithetical but, if properly implemented, they can complement each other.

⁹⁰ usually optimization can produce a valid result in a smaller amount of time.

⁹¹ Railway bogies are a practical and relevant example:

1. An extremely vast set of possible bogie designs exists. While optimization could be implemented to find the best dimensions (and similar parameters) for an already defined layout, its application to explore all possible 3D structure configurations could be significantly more complex.

2. The bogie design process has to consider a large collection of features such as materials, costs, manufacturing processes and maintenance. As an example, a solution could perform very well from a theoretical perspective yet be very difficult or costly to manufacture/-maintain.

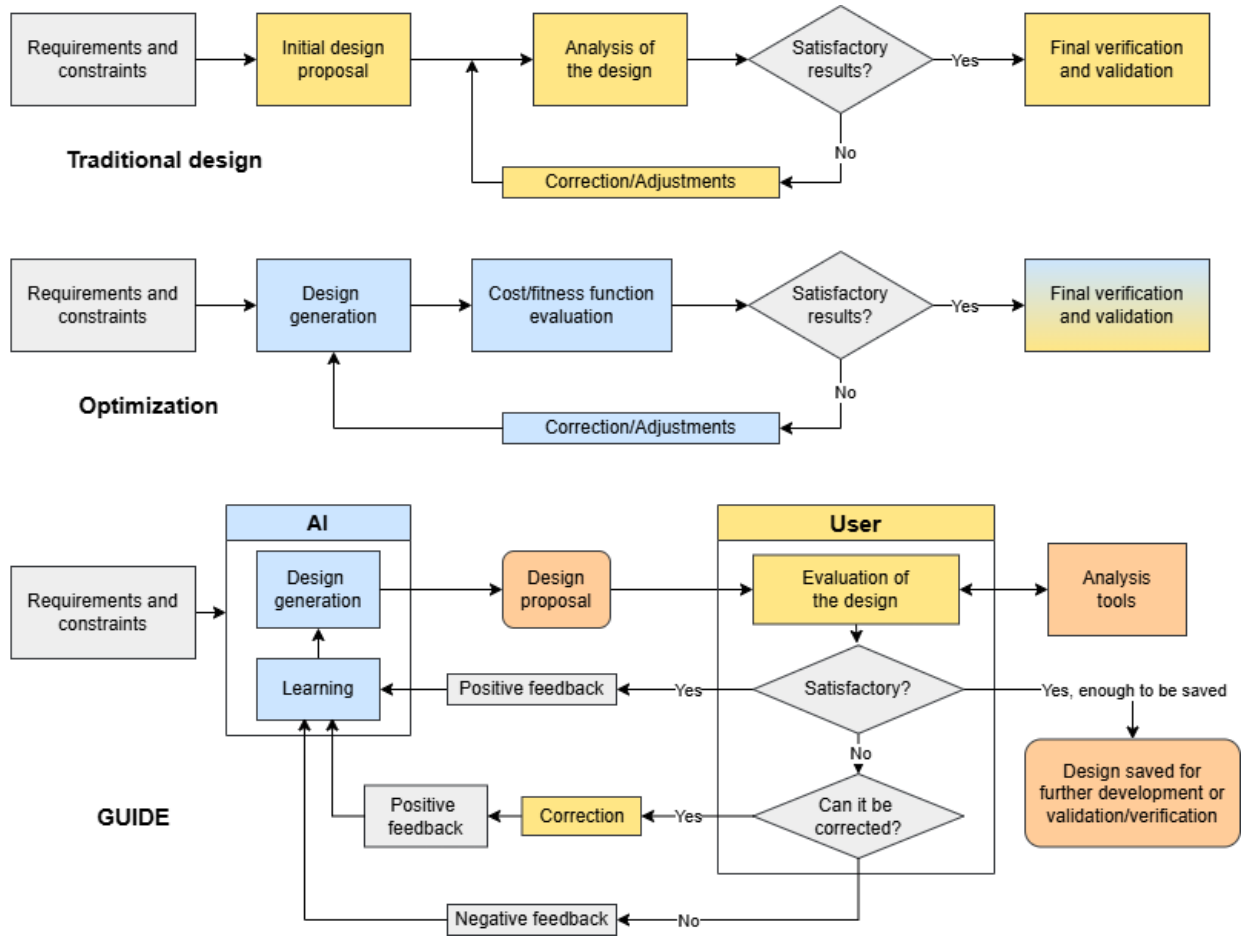
While this is true, the numerical component of the design process is not ignored and it is fact fully part of the process: the user can employ the integrated analysis tools to evaluate the performance of the design from a quantitative perspective.

⁹² usually more than performing an optimization.

⁹³ e.g. which shapes are preferred for the structure, where to place certain elements with respect to others, etc.

⁹⁴ The later in the design process corrections are applied, the more costly and damaging they are. As an example, if an accurate CAD model is realized of the structure, only to find out that it does not perform as expected, significant resources are wasted. Being able to preliminary analyze designs in an early phase can save time and resources.

⁹⁵ i.e. defining a preliminary layout to be then refined.



Comparison between the flow diagrams of the discussed approaches to the design process.

Notably, traditional design and generative optimization broadly share their workflow structure. In fact, in both cases designs are progressively and iteratively corrected considering the results of one or more evaluations of the performance of the design⁹⁶

The GUIDE framework relies on iterations too, yet the cyclic nature of the process is due to the continuous interaction between the user and the AI. Furthermore, more complex dynamics are introduced with respect to a simple closed loop. As an example, the user decides when to stop the iterations, designs can be saved and previous configurations loaded.

While the goal of optimization is to provide one or more final designs, the objective of GUIDE is to assist the user in its creative process. No limit on saved resulting designs exists and, at least theoretically, the user can explore as many solution as desired. As the name suggest, the objective of this approach is to *guide* the designer in the development of a valid layout, in place of directly providing a final output.

Light blue: tasks performed by software or AI

Orange: tasks performed by the user

As mentioned, in generative optimization, the verification/validation phase can be both carried out by software or users

⁹⁶ This is a very broad concept. Typically in traditional design the evaluation is performed by numerical analysis, while in optimization results of such analyses are converted to a cost function. These are however broad generalization and implementation may significantly vary.

Implementation of GUIDE

To implement the GUIDE approach in a comprehensive design tool, the following features have to be developed:

- A **generative AI** capable of structure generation and learning from user inputs
- A **user interface**, to display designs generated by the AI and allow the user to interact with these. It should also allow to upload and save data
- **Analysis tools** to allow the user to assess the performance of the proposed designs

Application to railway bogie design

The parameters to be determined in the bogie design process vary on a case-by-case basis⁹⁷. Nevertheless, to present a practical example, it is necessary to determine a set of fixed parameters and design variables.

In fact, considering all possible parameters that define a bogie and assuming these to be either fixed or free, the number of possible combinations to be accounted for is extremely large. Implementing and showcasing all of these would be unnecessarily time consuming and useless⁹⁸.

In light of these assessments, the approach that has been chosen is to:

1. Preliminarily determine which parameters to fix and which to consider as design variables. A 'sample project' is therefore defined: it represents a possible scenario so that a realistic example could be presented.
2. Implement the design tool to be as flexible as possible, so that parameters can be easily switched from being fixed to free.
If new constraints are introduced or, on the contrary, some previously fixed parameters are freed, the tool can easily adapt to the new setup.

While a single combination of fixed parameters and design variables has been chosen to be presented, this is a matter of clarity and clearness in the exposition and not an inherent limitation of the tool. In fact, all fixed parameters could easily be unconstrained and vice-versa.

⁹⁷ To provide some examples, the length/height and other dimensions of the bogie could be specifically set and requested by the customer or left to the designer to be decided.

Some variables can be constrained by engineering requirements too: as an example, a bogie that has to withstand high loads could be forced to be designed with three axles, as two would not be enough to satisfy regulations. The gauge too is pre-determined by compatibility requirements.

⁹⁸ This is even more relevant when considering the demonstrative nature of this implementation.

The selection of design variables has been carried out with the ultimate goal of avoiding unnecessary complexity while still providing a challenging design objective to showcase the capabilities of the GUIDE approach.

in the project, parameters can be divided in three broad categories:

- **Completely fixed parameters:** this category includes all parameters that would require a considerable restructuring of the software to be modified. Some examples are:

- the *number of axles*, which is set at two. While three and even four axles bogies do exist, they are typically more complex in terms of design. A two-axles bogie has been deemed more flexible in terms of implementation and demonstration of different available features.

While the GUIDE approach could be adapted to accommodate for more axles, this is avoided as it would far exceed the scope of this project.

- layout and geometrical parameters of the primary suspensions. As it will be later discussed, proper modeling of suspension systems is not trivial and significant effort would be required to develop a database of primary suspensions layouts, far exceeding the time constraints of this project⁹⁹.

- **Preliminarily tunable parameters:** this category includes parameters that are set at the beginning of the design process and cannot be altered in following phases.

In general, these parameters can be easily set and are fixed during iterations simply to avoid an excessive number of free variable and reduce the size of the configuration space. Examples include:

- Primary and secondary suspension stiffnesses
- Wheelbase, gauge and other relevant axles dimensions
- Bogie size limits
- Geometrical and material properties of structural elements

- **Free parameters:** in the context of this project, parameters that can be altered are mostly relative to the position and shape of the bogie structure.

In particular, the objective of the AI generation process and of the user editing is to devise a viable bogie frame, once the other parameters have been set.

⁹⁹ As discussed in more detail in following chapters, the primary suspensions layout has been chosen to be quite complex in nature and present many relevant features common to most suspension systems.

As the addition of other possible suspension systems is more a matter of time and effort than it is of software programming, it has been preferred to prioritize more relevant features and topics.

Some of the preliminarily tunable parameters can be determined by the use of the dynamic analysis tool included in the project. Its working principles will be later discussed.

Quantities that can be evaluated include:

- Primary suspension vertical and lateral stiffness
- Secondary suspension vertical stiffness
- Primary and secondary suspension vertical damping
- Wheelbase

Other tunable parameters can be either imposed by customer requirements or determined by realistic and compatible values found in literature.

Parameters such as gauge can be set in the context of the project but their value is typically fixed for tracks compatibility.

Part III

AI Implementation

Requirements and constraints

To properly understand the approach that has been chosen to be implemented, it is necessary to first examine in which contexts it has been developed to work in, the resulting limitations and its requirements.

To identify the objectives and the constraints of the implementation it is first off worth examining what the ideal scenario would be and then evaluating its feasibility, correcting and adapting until a feasible implementation is devised.

The ideal AI

Considering the design process described in the previous chapter, an ideal AI would, given a set of requirements, generate a final, already tested and verified, design. This would require the AI to:

1. Understand requirements, which typically are provided in the form of text, tables and other data collection. Proper language and input interpretation tools would therefore be necessary.
2. Generate valid solutions, compatible with constraints and satisfying the objectives of the design. Realistically, solutions would have to be iteratively refined to improve their performance, until a final candidate is obtained.
3. Validate and test the final solution. If the validation and verification phase is not passed, the procedure should be repeated from the generation phase.

Clearly, an AI capable of performing all these task would far surpass currently available capabilities, especially in the context of this project.

While the ideal AI represents an unreachable concept, it represents an useful benchmark: by identifying the problems that arise in its implementation, it is possible to conceptualize a more realistic and practically realizable generative AI approach that can fulfill as many as possible of the desired tasks.

This is in fact the approach that has been adopted in the project: an initially defined set of requirements has been compared to actually implementable solutions to devise a feasible generative AI tool for the project.

The main issues that have been identified when examining the ideal AI implementation are:

- **Text comprehension in the requirements assessment:** usually requirements are not presented by the buyer as a simple list of numbers but in a much more complex textual form, that requires knowledge and experience in its corresponding field to be properly understood.

While modern language model can effectively understand technical documents, the chance of misunderstandings/misreadings occurring is not negligible: given the large amount of information, it is easy to presume that some errors may occur during the interpretation process. Unfortunately, even a single wrong input could significantly damage or impair the whole design process.

In the project, the software has been developed assuming that requirements and constraints for the design have already been extracted and available. The data input interface has been conceived with compatibility in mind, though, to allow for connection to a requirements interpretation system, if eventually developed.

- **Lack of training data:** even considering small AI models, with few inputs and outputs parameters, a considerably large dataset is required¹⁰⁰.

This is in open contrast with reality: the number of detailed designs and their corresponding requirements that are available for training is extremely small.

As a consequence, even training an extremely simple conventional AI model would prove to be unfeasible.

- **Complexity of the design process:** summing up the design process in a purely computational and numerical systems is extremely challenging, if not impossible at all: as mentioned, many design choices are not only based on numerical results but on human experience, practical limitations and other non-numerical factors.

In general, the large amount of design parameters can easily surpass the capabilities of possible generative AI implementations.

- **Performance evaluation:** the AI should be able to evaluate the performance of the design. As discussed, determining a valid cost/fitness function is typically a complex task.

Furthermore, for complex systems physics simulation would be required. As it will be apparent in later sections, the conceptualization and implementation of such models can be quite complex and only provide approximate results.

- **Validation process:** the objective of validation is to assess the compliance of the design to a wide range of requirements. To do so, a well-defined and detailed model is often required¹⁰¹.

The realization of such a detailed and accurate model is in open contrast with the other identified issues and is therefore to be considered unfeasible.

A possible solution would be to implement some form of data verification of the data extracted by the language model.

While technically possible, this would be in contrast of the overall objective of automating the design process: data could be either verified by a human user or an automated system.

In the first case, the overall goal of automating the design process is considerably contradicted; in the second case, the issue of reliability in automated data verification persists.

The implementation of a language model would anyway represent more of an exercise in programming and linguistic than in engineering. Given the time constraints of this project, its implementation has therefore been avoided.

Nevertheless, it can be realistically assumed that, given a larger time frame, such a system could be realized and adopted.

¹⁰⁰ in the order of thousands/ten of thousands data points.

Considering the example of a bogie, the number of free parameters and variable is extremely large and can be barely handled.

The geometry of the frame and other components is rich in details and often small adjustments can significantly alter its validity.

Furthermore, many subsystems are present in bogies and they should properly accounted for, such as traction motors, electrical and pneumatic connections, brakes, and so on.

Implementing an AI capable of handling this vast configuration space would therefore be practically unfeasible, especially considering the extremely reduced available dataset.

¹⁰¹ An example is the evaluation of stresses in a structure: the presence of notches can significantly affect the stress distribution. It is therefore difficult to properly evaluate stresses without a detailed model of the component.

Realistic practical options

In light of the presented issues, the scope of the AI has been reduced to be more feasible, considering the time constraints of the project too. The practical application of the AI has therefore been realized to be able to:

- Start its design process from already obtained constraints¹⁰².
- Work with a simplified mechanical model: in the case of this project, the frame of the bogie has been represented as a set of beams. In general, other simplified shapes can be chosen. In some cases this approach could not be valid, but it is often the case that, at least in a preliminary design phase, a reduced model is enough to understand the overall behavior of the systems and to identify solutions that would not be viable¹⁰³.
- Constantly interact with the user and learn from its action, to compensate for the lack of training data. Generation is therefore contemporaneous to learning, at least if so desired
- Start from known configurations, to avoid learning from the ground up each time the tool is used. In particular, the AI should be able both to start from a previously learned configuration or from an appositely designed configuration¹⁰⁴. Clearly, it should then be possible to save and store configurations for future use

Possible implementations will now be discussed, to assess how these requirements could be satisfied.

¹⁰² from the buyer

¹⁰³ to avoid further investing time in these

¹⁰⁴ a practical example in bogie design: the AI should both be able to start from what it has learned from a previous session. At the same time, if the overall shape of the bogie is already known and only minor adjustments are desired, the system should be able to start from an apposite layout provided by the user, to obtain coherent results from the beginning, with no waste of time.

Conventional AI implementations

As many AI types, each with their own features, are in existence, a possible implementation of these has been considered. Their main characteristics and working principles will be described, to demonstrate why they have not been deemed able to satisfy the expected requirements.

A new AI concept, capable to learn to generate mechanical systems, has therefore had to be conceived and realized.

Deep Neural Networks

Deep Neural Networks (DNN) are a class of machine learning models that have seen a widespread adoption in the contexts of data and image classification, object detection and speech-to-text conversion.

Their main feature is their multilayer structure, composed of interconnected artificial neurons. Each layer transforms its input data through a series of linear and non-linear operations.

The fundamental building block of a DNN is the artificial neuron, which performs a weighted summation of its inputs followed by a non-linear activation function, such as ReLU, sigmoid, or tanh, to introduce non-linearity into the model.

The architecture typically includes an input layer, several hidden layers, and an output layer.

Deep neural networks are typically adopted for input classification. A common example is detecting the content of an image.

During training, the model is provided with a collection of image-label pairs. Labels describe the content of the image and are considered to be certainly accurate (ground-truth).

During each iteration of the training process, the model tries to guess the correct label for the image. A correction is applied to the network, reducing the mismatch between the expected and actual output. As this process is repeated, outputs that are produced by the model become closer to the correct ones.

Typical examples of image classification are handwritten text interpretation (each symbol is treated as an image) and object classification (e.g. in self-driving vehicles).

Due to their excellent pattern recognition capabilities, DNNs can also be employed to analyze signals and detect relevant or recurring features that would otherwise be extremely difficult to identify for a human user.

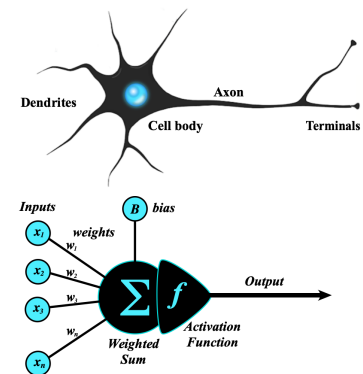


Figure 22: The fundamental functional unit of DNNs is the artificial neuron; the latter is designed to mimic the behavior of biological neurons.

Inputs are summed and the result is then processed by the activation function, which introduces the non-linear component of the model.

In the context of machine learning, the presence of non-linear elements is necessary to obtain proper results.

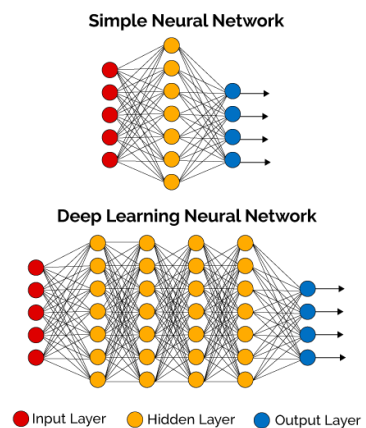


Figure 23: Simple and deep neural networks share the same overall working principle. The main difference between the two is the number of neurons and layers.

Clearly, as the number of hidden layers is increased, the model becomes more complex and therefore more capable of learning; on the other hand, training time and computational effort are increased.

Deep Neural Networks, why not?

The adoption of DNNs in this project is unfortunately unfeasible due to multiple reasons:

- Deep Neural Networks require a vast dataset of inputs-outputs correspondences. This would require a large collection of requirements lists and their respective outputs. In the case of railway bogies, as an example, such collections of data do not exist.
- Many designs could be obtained from the same set of requirements: the definition of one-to-one input-output correspondences is therefore almost impossible.
This is in stark contrast with the operating principle of DNNs. In fact, they are mostly designed to learn to associate inputs to labels and are therefore more well-suited for categorization tasks. On the contrary, they are not, generally at least, well suited for generation.
- Real-time learning would be impossible, as once deep neural networks are trained, they cannot be modified.
- Output type: the output of DNNs usually consist in the selection of an option among a set of available ones. Adapting the nature of these outputs to the generation of 3D structures is very difficult, if not impossible.
- In general, DNNs are synthetic in nature: they produce a single output from a large collection of inputs. As generative processes usually require the opposite process to happen (i.e. the generation of a large number of outputs from a small set of requirements), DNN are not well suited for their realization.

Reinforcement Learning

Reinforcement learning (RL) is a paradigm of machine learning in which an agent learns to make sequential decisions by interacting with an environment in order to maximize a cumulative reward signal.

Unlike supervised learning¹⁰⁵, where ground-truth labels guide the learning process, RL relies on delayed and often sparse feedback.

The *agent*, which is the subject of the training process, learns on a trial-and-error basis: it proposes a possible sequence of actions and the performance of the latter when interacting with the environment is evaluated by a corresponding reward.

The agent then iteratively tries to maximize the reward by improving its decision-making. A proper definition of the reward function is therefore extremely important to achieve satisfying results.

From a broad mathematical perspective, RL involves the association of a sequence of actions to states in a state space. The state space represents the possible configurations of the system and the surrounding environment¹⁰⁶. The objective of the training is to associate to each state a proper course of action¹⁰⁷.

To provide a practical example, even a simple neural network with a few inputs and outputs may require some thousands of data points to be properly trained. Furthermore, only part of the dataset is employed for training, as data for the validation and testing process is required too.

Considering the extremely large number of parameters to be set in the design of a railway bogie, such a dataset clearly does not exist.

Due to the large amount of DNN variants that have been developed, it is realistic to assume that these issues do not necessarily apply to all of them.

Nevertheless, the introduction of more complex approaches has been deemed unnecessarily time consuming with respect to the AI workframe that has been developed for this project.

¹⁰⁵ Deep Neural Networks represent an example of supervised learning: during training, the correct or expected output to be provided is known and well-defined. This is not the case for RL: if the correct action to be performed in a certain state was already known, it would just be implemented with no need for training.

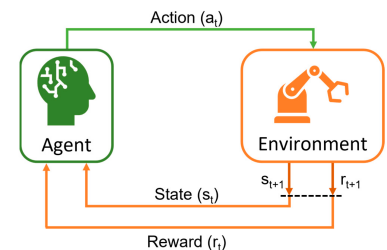


Figure 24: Schematic representation of the working principle of RL

¹⁰⁶ Due to its combinatorial nature, even a small set of variables could produce a vast state space. This can significantly increase training time.

¹⁰⁷ in the case of extremely large state spaces, some interpolation may be performed to account for missing state-action correspondences. This approach should be carefully implemented, as it may lead to improper performance.

RL is particularly well-suited for problems characterized by sequential decision-making under uncertainty, with long-term dependencies and potentially unknown dynamics. Examples are robotic control, game playing and autonomous driving.

Autonomous driving represent a typical context in which RL can be implemented. The state space is represented by the current value of variables that are both intrinsic (such as engine speed and torque, available power, steering angle, etc.) and external (characteristics of the surrounding environments, such as road type, presence of objects and signs, etc.) with respect to the vehicle.

The objective of the training process is to obtain a model that can reliably associate an action (e.g. increasing or reducing the speed, turning, etc.) to the current state.

Reinforcement Learning, why not?

While reinforcement learning is a typical approach for on-line/real-time training. Unfortunately, this approach cannot be employed due to various reasons:

- Reinforcement learning is best suited for processes in which a new state has to be determined from a previous one and a set of external inputs. Unfortunately, this characteristic is not well-suited for generation of structures.
- Training data-set: to provide enough data samples to the AI, it would be necessary to run a very large number of iterations even for a simple model. In fact, imagining a possible implementation, a human user would have to provide a rating to layouts generated by the AI. This should be repeated for thousands of times, even for a simple model.
- Lack of flexibility: again, once the training is completed, it is very difficult to modify the capabilities of the AI. This solutions does not therefore allow for easily starting from a predefined or saved layout and in general it hinders the model customizability.

As in the case of DNNs, output compatibility issues arise when considering RL. In fact, in both cases the nature of their outputs does not allow for easy conversion to 3D structural elements.

Considering the objectives of the project, the user would have to evaluate proposed designs, maybe with the aid of numerical tools, and provide a feedback on their performance. This procedure would represent the agent interaction with the environment.

This approach is not only very time consuming, but quite imprecise: well-defined evaluation criteria should be implemented; they should also be representative of design goals to obtain valid results. Clearly, the additional degree of complexity that would be involved renders the adoption of this approach unfeasible.

Generative AI

Differently from deep neural networks and reinforcement learning, generative models are designed to learn the underlying distribution of training data in order to synthesize new, realistic samples. A large variety of models have been developed, each typically targeting one or a few output types¹⁰⁸.

Some examples of generative AI include:

- **Generative Adversarial Networks (GANs)** consist of a generator and a discriminator that are trained in a minimax game. The generator produces artificial samples from random noise, while the discriminator attempts to distinguish real samples from generated ones. Over time, the generator learns to produce highly realistic data. GANs are particularly effective for image generation tasks but are notoriously difficult to train due to instability and mode collapse.

¹⁰⁸ Generative models can be employed in the production of text, music, images, videos and other media. More specialized models can provide specific outputs such as software code and other specific applications. Typically, different outputs require dedicated approaches, which are usually not compatible. Nevertheless, some overlap may exist: as an example, models adopted in text generation can be adapted to generic time series.

- **Variational Autoencoders (VAE):** input data is compressed into a lower-dimensional space (the so-called *latent space*) and then reconstructed. The model learns to generate new data by sampling from this space. VAEs are useful for tasks where it's important to explore variations of data or understand its underlying structure.
- **Autoregressive models**, such as those used in modern language generation (e.g., GPT), generate content one element at a time (like words in a sentence) by predicting the next part based on what has already been produced. These models have shown remarkable performance in generating coherent text and realistic audio, though they can be slow when producing long sequences.
- **Diffusion models** have recently gained attention for their high-quality results, especially in image generation. They work by learning how to reverse a process that gradually adds noise to data. Starting from pure noise, the model learns to recover clean data in a series of steps, often producing remarkably detailed and realistic outputs.

Generative models, why not?

While these approaches are conceived with generation as their ultimate goal, they still present relevant issues in the context of structure generation:

- **Output conversion:** as these models are typically designed for a specific output format, it may be extremely difficult if not impossible to convert it to an unconventional output type.
As the generation of 3D structures represent an unusual objective for AI generation, existing and available models are typically not well suited for it.
- **Model complexity:** generative AI models are notoriously very complex and require extensive development to reliably produce valid outputs.
- **Training data:** as for DNNs and RL, available dataset are far smaller than what would be necessary to obtain valid results, even for a reduced and simple structure.

From a general perspective, generative AI is well known for its high degree of complexity, typically vastly surpassing DNNs and RL.

While the adoption of a generative AI model would be theoretically possible, its practical implementation would require a disproportionate effort with respect to obtained results.

Furthermore, the insufficiency of available training data is even more relevant, as such complex models require extremely large collections of reference samples.

Due to the complexity of the models, computational effort is significantly increased too.

The G-Map Approach

The Generative Maps (G-Map) approach has been conceptualized and developed for the specific requirements of this project and represent its main element of innovation, as it has been conceived from the ground up. It represents the AI component of the GUIDE framework.

To maintain generality, the G-Map approach will first be presented from an abstract and theoretical perspective. As it is conceived to be applied in a vast range of contexts, one of its requirements is in fact to be as general as possible in its scope.

Since many concepts and their role may not be easily understandable from a theoretical-only point of view, the implementation that has been realized in the project will be therefore be discussed too.

Fundamental concepts and definitions

The understanding of the working principle and the main features of the G-Map requires an introduction to the mathematical objects that are adopted.

Their definitions and descriptions will hereby be provided, complemented by practical examples.

Object: an object $O(T)$ of type T is defined as a structured collection of variables, attributes or other objects, such that two objects $O_1(T)$ and $O_2(T)$ present the same structure and require the same arguments, even if with different values.

Point: a n -dimensional point $\mathbb{P}^n(X_n)$ is defined as an ordered n -tuple X^n of coordinates that uniquely describe a position in a n -dimensional space¹⁰⁹.

In the project: a three-dimensional cartesian reference frame is chosen for the bogie structure and the position tuple X^3 is defined to be $X^3 = (x_p, y_p, z_p)$, where x_p, y_p, z_p are the cartesian coordinates of the point.

Node: a n -dimensional node $\mathbb{N}^n(\mathbb{P}_n, [Att])$ is an object whose position is defined by the point $\mathbb{P}^n$ and characteristics¹¹⁰ by its attributes set $[Att]$. It represents an extension of the concept of point, allowing to describe not only the latter's position but its expected behavior in the operating environment.

As discussed, due to the very specific requirements of three-dimensional structure generation and the reduced training data size, the introduction of a new and innovative approach has been necessary.

G-Map and the GUIDE framework therefore represent a new solution, with all the corresponding benefits (e.g. contribution to innovation) and issues (e.g. lack of comparison and benchmarks, absence of reference material, increased development effort, etc.).

As the G-Map approach is adopted in the GUIDE framework, mathematical concepts and features have been developed to guarantee compatibility with the latter.

As it will later be discussed, adopted solutions guarantee easy conversion from AI generated designs to user interactive representations and static analysis models.

¹⁰⁹ This is basically the conventional definition of an n -dimensional point. In this project, cartesian coordinates are adopted; this is not mandatory and, depending on a case-by-case basis, different options could be chosen.

¹¹⁰ in the context of the G-Map and GUIDE workframe.

In the project: three-dimensional nodes are adopted. A three-dimensional node $\mathbb{N}^3$ is described by:

- its position, defined by the point $P^3 \in \mathbb{N}^3$
- a node ID $id(\mathbb{N}^3) \in [Att]$; node IDs allow for identification. They are extremely important for proper operation and their role will become apparent when describing the working principles of the G-Map approach
- a node type $t(\mathbb{N}^3) \in [Att]$. The node type is interpreted in different phases of the GUIDE procedure and provides information about the expected behavior of the point. Some examples:
 - *AI generation*: during the generation of a candidate structure by the AI, node type allows to determine which coordinates of the node are free and which are pre-determined or fixed. The position of a node may be completely free, completely constrained or free only along some directions. Furthermore, node type allows to identify the role of the node in the structure: as an example, the number of secondary suspension nodes is constrained and identifying them is necessary to determine their number
 - *User interaction*: during user interaction, node type allows to determine how a node can be edited. As in the case of AI generation, the coordinates of the node may be fixed along some directions and free along others. Furthermore, some node types may be added or removed by the user, while others are not deletable or cannot be inserted.
 - *Structural analysis*: since the node type also indicates the role of the node in the structure, it defines its required constraints, applied loads and other mechanical characteristics in the context structural analysis.

Edge: an n -dimensional edge $\mathbb{E}^n(\mathbb{N}_1^n, \mathbb{N}_2^n, [Att])$ is defined by two node objects $\mathbb{N}_1^n$ and $\mathbb{N}_2^n$ and an attributes set $[Att]$.

Differently from their geometrical counterparts, they are not extended uni-dimensional objects but are merely structures defined by two points and their attributes: the overall software framework is responsible for their geometrical and structural interpretation.

Different implementations could therefore produce different geometrical and mechanical representation of the same edge object¹¹¹.

In the project: edges are geometrically defined by two three-dimensional nodes $\mathbb{N}_1^3, \mathbb{N}_2^3$ and an attribute set $[Att]$

From a geometrical perspective, the edge object is interpreted as a segment with its extremes defined by the position of $\mathbb{N}_1$ and $\mathbb{N}_2$. This representation is adopted to graphically visualize the edge and its position.

The geometrical interpretation is expanded to include the edge ID from the $[Att]$ list during AI generation and user interaction. it is in fact extremely important to correctly identify each edge to properly edit the structure.

In the context of the mathematical description of G-Map objects, apices are employed to indicate the dimensionality of the corresponding element.

As they can add unnecessary clutter to formulas and expressions, their use will typically avoided if not required; this is typically the case if the dimensionality of the object in exam is either well defined or can be easily assessed by context.

From a theoretical perspective, $\mathbb{E}^n(\mathbb{N}_1, \mathbb{N}_2, [Att]) \neq \mathbb{E}^n(\mathbb{N}_2, \mathbb{N}_1, [Att])$. Edges can therefore be considered oriented. That said, in many practical application (such as this project), this distinction is unnecessary and not considered.

¹¹¹ While this is true in theory, in practice the concrete realization of the edge object usually is the conventional geometrical edge.

The main advantage of this representation of edge objects is that they can edited in one phase of the procedure (e.g. their geometry is modified during user interaction) and the changes are instantly readable by other sections of the code (e.g. the static analysis tool). While providing different interpretation of it, different phases do therefore access the same elements database.

From a structural and mechanical point of view (e.g. in the context of static and dynamic analysis), edges can be interpreted as

- FEM beam elements:
 - The geometrical extremes of the beam are provided by the position of N_1 and N_2
 - Material (e.g. elastic modulus, Poisson ratio, density, etc.) and geometrical (e.g. section area, section inertia moduli, etc.) properties are included in the attributes list $[Att]$
 - the node ID is still relevant to properly reconstruct the structure
- Dynamical components, such as springs and dampers. As in the case of beam elements, their extremes are specified by the position of N_1 and N_2 . Their geometrical and dynamic characteristics are specified in the attributes list.

The edge ID, contained in the edge attributes list, is also employed to properly reconstruct the structure during analysis.

In general, edge objects allow to freely specify their type. The latter then influences their behaviour during various phases of the GUIDE iterations.

- During AI generation and user interaction, some edge types may not be movable or editable. As an example, element in suspension systems may not be altered in their shape and location.
- As discussed, the mechanical properties and behaviour can be determined by the edge type.

Workspace: the n -dimensional workspace W^n is the *finite* set of possible coordinates X^n that could define the position of a point P^n .

The G-Map approach does in fact require the workspace to be discretized and the set of possible coordinates to be finite. As the discretization can vary on a case-by-case basis¹¹², the numerical definition depends on the specific context.

In the project: as cartesian coordinates are adopted in the project, the workspace is defined as:

$$W = \{X_n = (x, y, z) \mid (x, y, z) = g \cdot (a, b, c)\}$$

such that

$$\begin{cases} a, b, c \in \mathbb{Z} \\ g \in \mathbb{R} \wedge g > 0 \\ (x_{min}, y_{min}, z_{min}) \leq (x, y, z) \leq (x_{max}, y_{max}, z_{max}) \end{cases}$$

where:

j_{min} and j_{max} are respectively the minimum and maximum value that the coordinate j can assume in the workspace

g is the grid resolution/size

As later described in more detail, the approach that is adopted for this project is to model the bogie structure as a collection of uni-dimensional structural and dynamical elements. The bogie frame is therefore represented as a collection of connected beam elements; this represents an example of a *wireframe* structure.

The wireframe approach has been chosen for this project as it represents a reasonable approximation of a railway bogie frame. As later section will demonstrate, significant simplifications of the modeling process can be achieved without a meaningful loss in accuracy.

This is even more true when considering that the objective is only to provide a preliminary analysis, as much of the details is missing in such an early stage of design.

With respect to other solutions, modeling the structure as a set of nodes connected by edges also allows for much easier manipulation of the structure geometry, while also facilitating the AI generation process.

While this approach has been chosen due to the nature of this project, other types of elements can be used. In general there is no significant restriction on the nature of elements to be used (e.g. plate elements, 3D-shaped elements, etc.) if not the implementability of these in the analysis tools (in particular FEM analysis).

To provide a practical example, a plate object could be defined by extending the concept of edge object to accommodate four nodes.

¹¹² e.g. depending on the dimensionality and type of the coordinate space.

A smaller grid resolution g allows for increased control on the structure, at the cost of heavier computational cost and training time.

In the project, $g = 10$ cm has been chosen: this has been considered to be a reasonable compromise between fine structure customization and computational effort. In fact, it is reasonable to assume that in a preliminary design phase, extremely fine position control is unnecessary.

As no direction is preferred, the grid resolution is set to be uniform along all axes.

Body: a body object $\mathbb{B}(\{\mathbb{E}\}, \{\mathbb{N}\}, [In])$ is a collection of a set of edges $\{\mathbb{E}\}$, a set of nodes $\{\mathbb{N}\}$ and an incidence matrix¹¹³ $[In]$ representing their connections.

In the project: the body is considered as the collection of all edges and nodes that are not fixed. This is not mandatory and this choice is due to increased ease of programming.
The body object $\mathbb{B}$ does therefore exclude all pre-defined bogie components, such as suspension systems.

Map: a map $\mathbb{M}$ is a function associating an object or value Y to each element of a finite input set $\bar{X}$.

First order maps associate to each input a so-called *probability factor* P . They are typically adopted to select the value of a parameter from a finite set of possible values, each with its own probability to be selected. Probability factors are representative of the likelihood of a value to be selected¹¹⁴.

Higher order maps associate a lower order map to each input. They allow for more complex structures, examples of which will be later discussed.

The input set $\bar{X}$ can contain one-dimensional or multi-dimensional candidate values that could be selected¹¹⁵.

Probability factors and input selection

As mentioned, probability factors are not constrained and can assume any real value. The guiding principle of their application in value selection is: the higher the probability factor, the higher the probability of the corresponding input to be chosen.

The procedure that is adopted can be described as follows:

1. A value is to be determined for a variable u . The corresponding discretized set $\bar{X}$ of possible values for u is considered.
2. A reference subset $X_r \in \bar{X}$ is considered.
3. A candidate value $x_c \in X_r$ is randomly chosen. The corresponding probability factor $y_c = \mathbb{M}(x_c)$ is evaluated.
4. Probability factors are computed for the whole X_r set, obtaining the probability factors set $\{Y_r\} = \mathbb{M}(X_r)$.

The maximum and minimum values of Y_r are evaluated:

$$y_{max} = \max(Y_r) \quad y_{min} = \min(Y_r)$$

5. A score $s(x_c)$ for the candidate input x_c is evaluated as:

$$s(x_c) = \frac{y_c - y_{min}}{y_{max} - y_{min}}$$

By definition, $s(x_c) \in [0, 1]$

6. A random real number $r \in [0, 1]$ is generated.
If $s \geq r$ the candidate value x_c is accepted and the variable is set to $u = x_c$
If $s < r$ the candidate value is rejected and the selection process is repeated from 3. until a value for u is selected.

¹¹³ The incidence matrix will be later treated in more detail.

As edge and node objects carry their own information, the body object also contains all attributes of its components. In general, it condenses all information about the structure to be generated, edited and analyzed.

¹¹⁴ Probability factors are different from the statistical concept of probability. In fact, they can assume every real value and are not constrained to the $[0, 1]$ interval.

¹¹⁵ As an example, a value could be needed for a 1D variable, such as the length of an element.
On the contrary, 2D or 3D coordinates for a point could be the objective of the selection process, representing multivariate inputs.

If all inputs are to be considered, $X_r = \bar{X}$. In some cases, though, only some elements of X may be available for selection.

As an example, a map could be defined representing the probability of point (i, j) to be chosen from a set X of coordinates. It may be the case that a point has to be chosen with a pre-set value for its i or j coordinate.

The reference set is therefore the subset $x \in X$ that respects this constraint.

A function can be applied to the score $s(x_c)$ if needed.

As an example, by applying the square root function $s'(x_c) = \sqrt{s(x_c)}$, scores can be artificially increased, especially if they already are close to 1. The probability of a candidate to be selected is therefore increased.

On the other hand, squaring the score $s'(x_c) = s^2(x_c)$ decreases the likelihood of a candidate to be selected and biases the selection process towards high scoring candidates

In general, the introduction of such auxiliary function allows for further customization of the selection process.

It is advisable yet not mandatory to select score functions $f(s)$ such that $f([0, 1]) = [0, 1]$.

The concept of map and its application in value selection may not be intuitive from a theoretical-only perspective. Examples are therefore presented to clarify their nature and role in the G-Map approach.

Example 1: a structure S is composed of one-dimensional elements and their lengths is constrained to be in the $[2, 14]$ cm interval. The interval is discretized with a resolution $g = 2$ cm.

The set of possible lengths is therefore:

$$\bar{X} = \{2, 4, 6, 8, 10, 12, 14\} \text{ cm}$$

A first order map $\mathbb{M}$ is defined such that:

$$Y = \mathbb{M}(\bar{X}) = \{-1, 3, 2, 5, -2, 3, 4\}$$

This map is therefore valid for any element length selection in the generation of S .

If an element has to be generated, such that its length l must respect the constraint $3 \text{ cm} < l < 11 \text{ cm}$, a coherent reference set X_r and its corresponding probability factors Y_r must be considered:

$$X_r = \{4, 6, 8, 10\} \text{ cm} \implies Y_r = \mathbb{M}(X_r) = \{3, 2, 5, -3\}$$

Assuming, for the sake of example, $x_c = 6 \text{ cm}$ to be chosen as a candidate for the value of l . Relevant values can be determined to be:

$$y_c = \mathbb{M}(x_c) = 3$$

$$y_{\min} = \min(Y_r) = -3 \quad y_{\max} = \max(Y_r) = 5$$

The score $s(x_c)$ can be evaluated to be:

$$s(x_c) = \frac{y_c - y_{\min}}{y_{\max} - y_{\min}} = \frac{3 - (-3)}{5 - (-3)} = 0.75$$

A random real number $r \in [0, 1]$ is then generated.

If $s(x_c) \geq r \implies l = x_c$

If this is not the case, the iteration is repeated with a new candidate.

Example 2: the position p of an element in a 2D discretized workspace W has to be determined. The workspace is defined as:

$$W = [0 : 1 : 10] \times [0 : 1 : 10]$$

A map $\mathbb{M}$ is defined for the whole workspace, expressing the probability of each point to be selected. The inputs space $\bar{X}$ of the map is therefore correspondent to the whole workspace W .

Assuming, though, that some other constraints exist, such that only points of the form $p = (2, y)$ can be selected, the reference set X_r is a subset of the input space:

$$X_r = [2] \times [0 : 1 : 10]$$

Once the reference set has been determined, the same procedure as in *Example 1* is followed. Scores can be determined for any $(x, y) \in X_r$ coordinate pair, allowing to chose a position p for the element from possible options.

This example demonstrates how inputs do not necessarily have to be single-valued: they can represent coordinates and other more complex mathematical objects.

While the distinction between $\bar{X}$ and X_r may seem unnecessary, it is quite useful when dealing with practical applications of maps.

As an example, a map may implemented for a whole workspace, yet some elements might be insertable in a restricted set of positions.

It would be unnecessarily complex to appositely develop and update a map for any of these circumstances; it is in fact much easier and more compact to just implement one and consider subsets of its input space if necessary.

To preserve the statistical properties and coherence of the process, a candidate could be chosen many times in a row. In practice, this is unlikely due to the typically large size of the input set.

The notation

$$W = [x_{\min} : g_x : x_{\max}] \times [y_{\min} : g_y : y_{\max}]$$

defines the workspace as the set of points such that:

$$\begin{cases} x_{\min} \leq x \leq x_{\max} \\ y_{\min} \leq y \leq y_{\max} \\ x = x_{\min} + g_x \cdot n_x \\ y = y_{\min} + g_y \cdot n_y \\ n_x, n_y \in \mathbb{N} \end{cases}$$

Reference frame: as the G-Map approach is adopted in the context of structure generation, a reference frame is needed to properly define the position and shape of its constituting elements.

As the GUIDE framework could involve supplementary analysis tools, the same reference frame can be adopted for these too¹¹⁶.

In the project: the same reference frame is adopted both for AI generation, user interaction and structural analysis. Its main axes are defined to be:

- *x-axis*: longitudinal axis of the bogie, parallel to the ground. It represents the symmetry axis of the bogie. x increases from front to back.
- *y-axis*: it is defined to be coherent with the x and z axes. It represents the lateral coordinate and, looking from the back of the bogie to the front, it is positive on its right side, negative on the left.
- *z-axis*: vertical axis of the bogie; $z = 0$ at ground height and it increases as the height does.
The z coordinate is assumed to always be positive, as negative values would be below ground level.

¹¹⁶ This is advisable from a ease of programming and understandability perspective, as multiple reference frames may cause unnecessary confusion. This is in fact the approach that is adopted for this project.

G-Map application in the GUIDE framework

As discussed, the G-Map approach as been conceived to be integrated in the broader GUIDE framework. Maps represent the core of the learning and generation process; as a consequence, they are updated and employed during most phases of the GUIDE procedure.

During the **AI generation** phase, maps guide the design choices that are necessary for the realization of a candidate layout solution.

The design process typically requires parameters to be chosen¹¹⁷. Maps guide the choice of such variables by employing the information that they have acquired during training.

The feedback provided during the **user interaction** phase is employed to update the maps with new information. At each iteration, results become more coherent with user expectations and, after enough training, they can closely match the concepts and ideas that guide the user design process.

Maps are therefore able to dynamically store information acquired during training; this feature represents the core of the AI learning process.

Due to their structure, they also allow for quick **export**, external **storage** and **data input**. In fact, from a numerical and software perspective they consist of simple numerical tables or multilayered arrays. Saving the maps that describe a certain snapshot of the AI training allows it to later restart from that exact configuration.

Furthermore, maps can be designed to provide the AI with a desired starting configuration.

¹¹⁷ such as dimensions of relevant elements and their position, mechanical, geometrical and dynamic properties, etc.

The implications of this approach are significant:

- As maps are refined iteration by iteration, results tend to converge to solutions that are most likely to be acceptable for the user. Due to their probabilistic nature, though, maps still allow for unusual or unexpected solutions to be proposed, albeit with reduced frequency. This additional element of randomness can stimulate the creative process by introducing solutions that would not have been considered, while not straying too far from expected and reasonable layouts. More generally, it is not even required for the user to have any expectation about the solution to be designed, if not for unavoidable constraints and requirement. In fact, as the user-AI interaction progresses, new concepts and layout may emerge, as creative elements are introduced by the AI and further refined by the user if deemed to be satisfactory.
- Maps can be purposefully designed and loaded into the AI to obtain a desired starting configuration. As an example, some parameters' values or element locations could be prioritized¹¹⁸. Significant time resources can be therefore saved by avoiding unnecessary training¹¹⁹.
- Generally speaking, maps are extremely flexible concepts that can be adapted to a wide range of applications. As mentioned, they allow to save, load, create and store AI configurations.

As the practical implementation of maps to achieve such objectives can vary on a case-by-case basis and represent a quite complex topic, its details will be treated in the following chapter.

¹¹⁸ Some practical examples are:

To provide an overall starting shape to the bogie frame, it is enough to introduce a point map where points that lie on and around the shape have much higher probabilities.

To prioritize some position of the secondary suspensions, it is enough to introduce a secondary suspension map where probabilities around the desired positions have been purposefully increased

To increase the probability of having a number of nodes in a certain range, the node number map can appropriately be modified.

¹¹⁹ As an example, if a certain overall shape is desired for the bogie, it is not necessary to train the AI to produce such shapes: it is enough to introduce a starting snapshot for the training that already prioritizes solutions that are more similar to the desired layout.

If not otherwise specified (no pre-made or saved maps are loaded), at the start of the training process maps have uniform probability factors: their value is the same for all candidate input.

As a consequence, when the AI is totally untrained, no configuration/input is preferred with respect to another. This is desirable, as in an untrained condition, the AI should not have any pre-imposed bias or expectation.

AI Implementation in the Project

As it is designed to be as general and adaptable as possible, the working principles of the G-Map approach are quite generic. To better understand how such an AI could be implemented, the project itself will be presented as an example.

Objects that are employed in the project have already been discussed in the previous chapter, with the exception of maps¹²⁰. The maps that are implemented will therefore hereby discussed, to provide a practical example in a more complex context.

The realization of the generation and learning process will then be analyzed, by presenting how they are realized in the project. While some differences may exist depending on the objective of the AI, broad and general concepts are presented that are likely to be applicable to most GUIDE implementations.

Generative Maps

As different types of parameters have to be set, the structure of maps that are adopted in their generation can significantly vary. Maps adopted in the project can be divided in three main categories.

First order maps: these maps are employed to chose the value of a variable. Examples include:

- *Transversal elements map:* transversal elements are nodes such that their $y = 0$ ¹²¹.
Such points must clearly exist, as the two lateral halves of the bogie must be connected. The transversal elements map associates a probability factor to a predetermined set of possible amounts of such transversal elements¹²².
- *Node number map:* the goal of the node number map is to provide information about the likelihood of a certain number of nodes to be present in the structure. A set of possible node numbers is defined¹²³; to each, a probability factor is assigned¹²⁴.
- *Edge number map:* conceptually the same as the node number map, with the main difference that it accounts for the number of edges¹²⁵.
- *Mechanical link map:* the position of the mechanical link is constrained by default to lie on the longitudinal axis of the bogie. The mechanical link maps takes as inputs the possible x -coordinates of the mechanical link and associates a probability factor to each.

¹²⁰ As it will become apparent in this chapter, an extensive description of implemented maps is quite lengthy. It has therefore been chosen to separately describe them and their role in the project.

¹²¹ i.e. they lie on the longitudinal symmetry plane of the bogie.

¹²² In the project, it is assumed that no more than 10 transversal elements are present. This is a quite significant overestimation, as typically their number is not greater than 3-4; nonetheless, 10 is chosen to leave the AI free to explore unlikely but possible solutions. Furthermore, at least one transversal element is present. The inputs for the map are therefore the integer numbers from 1 to 10 (extremes included). To each one, a probability factor is assigned, representative of how likely the bogie is to have that number of transversal elements.

¹²³ clearly, these values must be integers.

¹²⁴ As it will be later discussed, the generation process actually takes place on half the bogie and is then expanded to the other half. Furthermore, not all nodes are considered in this count. The process will be later discussed in more detail.

¹²⁵ The same considerations apply as in the case of the node number map. The details of the generation process will be later presented.

Two-dimensional maps: these maps associate a probability factor to a finite set of two-valued variables¹²⁶. They are defined as two dimensional as they are memorized as a two-dimensional table¹²⁷. The most relevant two-dimensional maps that are implemented in the project are:

- *Points map:* the point map assigns a probability factor P to each (x, y) pair of the workspace¹²⁸. This probability factor is then employed during generation to assess which (x, y) coordinates are most likely to be occupied by generated points. This is a two-dimensional map, as a pair of values is associated with a probability factor.
- *Secondary suspensions map:* this map has the same structure and features of the points map; in fact it could be considered a points map that is dedicated to secondary suspensions only. It allows to determine where secondary suspensions are more likely to be placed.

Two dimensional nested maps: these are by far the most complex maps that are adopted in the project. They associate an uni-dimensional map to each element of a finite set of two-valued variables¹²⁹. Examples of this category in the project are:

- *Height map:* the height map associates each (x, y) pair with a map that represents how likely a point that has coordinates (x, y) is to be found at a certain z-coordinate.

The two-dimensional component of the map is therefore represented by all possible (x, y) pairs of the workspace; a map having as inputs the possible z-coordinates of the workspace is then associated with each pair.

- *Edge direction map:* the edge map associates each (x, y) coordinate pair with a map representing the probability of edges connected to that point to have a certain orientation: the inputs of the secondary map are angles¹³⁰.

When evaluating the probability of an edge of being in a certain direction, it may not necessarily be aligned with these predetermined angles. To obtain a probability factor for it, interpolation between the two adjacent values is performed.

Design Generation

During the generation phase, the previously described maps are employed to guide design choices.

In the project, the symmetry of the bogie is exploited in the generation process: only half of the bogie (with respect to the longitudinal symmetry plane) is generated; the remaining half is then reconstructed once the generation process has been concluded.

This solution allows for reduced complexity and computational effort.

¹²⁶ these are typically two dimensional coordinates, yet they may describe other features. As an example, they could represent the number of elements of two types (e.g. edges and nodes) and their possible combinations.

¹²⁷ in general, they are similar to a discrete input two-variable function.

¹²⁸ as the workspace is discretized, only a finite number of (x, y) pairs exists.

¹²⁹ the difference with respect to two-dimensional maps is that they associate a map to each value in place of a single-valued probability factor.

¹³⁰ in particular, the round angle is uniformly divided in eight 45° sectors.

The symmetry also explains the relevance of considering transversal nodes: as they represent the connection between the two halves, they represent boundary nodes during the generation process. A proper definition of their position and number is therefore relevant to obtain satisfactory results.

The same approach is implemented for the user interaction too: the user only edits one half of the bogie and the other half is modified as a consequence.

Another relevant feature of the generation process is the layout is designed in two-dimensions first and then converted to a three-dimensional shape. As mentioned, the overall shape of a railway bogie is much more determined by its longitudinal and lateral components than the vertical one. Point coordinates on the x - and y -axis are determined first; the z component is then generated.

While such considerations may not be valid for many structures and component, they have been implemented to showcase how properly adapting the overall GUIDE system on a case-by-case base and exploiting feature of the object of the design procedure can significantly improve results.

To accommodate for this approach, different workspaces are defined in the project:

- W includes all points (x, y, z) that are valid coordinates for characterizing features (such as extremes in the case of edges or position in the case of nodes)
- W_2 only accounts for the (x, y) coordinates. It is therefore a two dimensional reduction of W , ignoring the z coordinate.
- $W_2^+ = \{(x, y) \in W_2 \mid y \geq 0\}$
This is the workspace that is employed during the first phase of the generation process. Points are then extended by symmetry to W_2 and then tridimensionalized to W .

The procedure adopted for the design generation is:

1. All elements whose positions and features are predetermined and fixed are generated first¹³¹.

In the project: the wheelset and primary suspension geometry is assumed to be fixed. These elements are therefore generated first; the rest of the bogie is then constructed around them.

2. Semi-constrained elements are generated. This category includes all elements that are partially constrained¹³².

In the project: examples of semi-constrained elements that are implemented are:

- Secondary suspensions: while their position is not fixed, it also cannot be freely chosen. In fact, some relevant constraints are:
 - their number is by default fixed
 - they should not lie on the longitudinal axis ($y = 0$) as pairs of secondary suspensions would then be coincident. This is anyway not advisable from a design perspective as it represents a mechanically undesirable configuration.

The (x, y) coordinates of secondary suspensions is determined by the use of the secondary suspension map.

To respect the previously mentioned constraints, the input set $\bar{X} = \{(x, y) \in W_2^+ \mid y > 0\}$ of the secondary suspensions map does not include points on the longitudinal axes.

The reference sample $X_r = \bar{X}$ as no additional constraints are present.

A relevant advantage of exploiting symmetries during generation and editing is the reduction in computation time and effort.

While in this project generation time is not too significant, as a relatively small number of element has to be generated, this may not be the case for more complex systems and structures.

As a consequence of this approach, the expected lateral symmetry of the bogie is also already built-in and does not have to be imposed or worse learned by the AI during the training process.

¹³¹ This approach avoids the need to ap-
positely impose their position, geome-
try and features in a later phase. Train-
ing the AI to already include such con-
straints would be even worse, as results
are not guaranteed and the required
training time could be significant.

¹³² Typical partial constraints involve
limitations in the position that an el-
ement can assume, in the number of el-
ements that can be generated or other
intrinsic properties.

It is worth mentioning that, dur-
ing point generation, selected points
should not coincide. While this may
seem obvious in theory, a proper imple-
mentation is needed to guarantee that
the same point cannot be selected more
than one time.

Project examples demonstrate how of-
ten partial constrained may present as
a collection of limitations of various
nature on the features of an element.
Other characteristics of the latter are
free to be modified or selected, though.

In the conversion process from two- to three-dimensional coordinates, the z coordinate of the secondary suspension can be either:

- (a) be set to a default value
- (b) determined by employing the height map
- (c) determined by an appositely implemented secondary suspension height map
- Transversal nodes: these nodes are partially constrained, as they lie on the longitudinal axis. Their y -coordinate is therefore constrained to be $y = 0$.

The transversal elements map is employed to select their number.

To determine their position, the points map is considered. While its input set $X = \mathbb{W}_2^+$, the reference set is considered to be $X_r = \{(x, y) \in \mathbb{W}_2^+ \mid y = 0\}$ to respect the imposed constraints.

3. Free reference elements are generated. Reference elements control the geometry of the structure and their position is edited during the user interaction phase to change the shape and layout of the structure.

The most common option is represented by simple nodes¹³³. They can either be the extremes of geometrical elements¹³⁴ or control the structure by a more complex approach¹³⁵.

While nodes represent the most straight-forward approach, if needed, more complex solutions may be implemented. This is however typically not advisable if not strictly necessary, as the user interaction could be made significantly more complex.

Option *a*) is the simpler one yet is typically effective enough if an approximate value for the height of the secondary suspension is known. This is usually the case if an overall concept for the desired layout is already defined.

Option *b*) represents a good compromise: it does not excessively increase the complexity of the secondary suspensions generation process while allowing for more refined control and learning

Option *c*) is by far the most complex and its implementation should be well justified, as the addition of a secondary suspension height map can require substantial effort. It should be implemented only if results obtained with simpler solutions are not effective enough.

In the project: nodes are chosen as reference elements.

Their number is selected employing the node number map. Only half of the nodes are actually generated in this phase, as symmetry is exploited. The inputs of the node number map are therefore half of the total number of free nodes.

Furthermore, only free nodes are considered, not including those contained in primary and secondary suspension. This is reasonable as the number of the latter types is fixed.

Transversal nodes are not considered too, as their number is defined according to their own number map.

The position of the nodes is selected by the points map. It is worth recalling that in this phase only their (x, y) coordinates are selected, as three-dimensionalization is performed in a later moment.

¹³³ this is in fact the case in the project.

¹³⁴ e.g. beam, plate and other three-dimensional elements.

¹³⁵ e.g. control points of Bezier curves.

4. Structural elements are generated in relation to reference elements; they represent the elements that actually define the structure and its mechanical properties.

As discussed, in this project the structure is mechanically defined as a collection of beam elements, yet other element types can be adopted if necessary.

In the project: the edge generation process is performed as follows:

- (a) A candidate edge is randomly generated, connecting two of the nodes.

If so desired by the user, it is possible to artificially set this process to be biased. As an example, edges connecting constrained and semi-constrained elements could be prioritized; this option has been not implemented as default, though, to allow the AI to freely explore layouts without pre-imposed conditioning.

- (b) A fitness score is evaluated for the edge. The two main contribution to the score evaluation are:

- **Edge direction map:** probability factors for the extremes of the edge are obtained from the edge direction map. These are then converted to probabilities considering the whole edge direction map as reference sample: the maximum and minimum value that are considered during the conversion process are the maximum and minimum values of probability factors for the whole map.

Two values are obtained, one for each extreme. Each of these is representative of the likelihood of an edge being connected to the extreme at that specific orientation. The mean of the two probabilities is then considered to evaluate an edge direction score.

Unfortunately, this approach can be quite heavy from a training perspective. In fact, if many points are present in the workspace, preferred orientation for each have to be obtained by a significant data collection that may require extensive training.

To solve this issue, a second generation mode is available: the simplified edge generation process does not consider the orientation of the edges and only expresses how likely an edge is to be connected to the considered extremes. This significantly reduces training time.

- **Intersection score:** intersections in the xy plane are detected. In fact, while not impossible, it is uncommon to have superimposing elements with respect to the longitudinal-axial plane. A parameter p_i is introduced to describe the likelihood of such intersection; it can be either set by the user or left unchanged to the default value, which testing has proven to produce valid results.

An adjusted score is evaluated by combining the edge direction and intersection scores through an appositely defined function.

- (c) The adjusted score is compared to a randomly generated number $r \in [0, 1]$ and accepted or rejected.

If the candidate edge has been rejected, the procedure is repeated until an edge is accepted and generated.

The choice between these solutions is mainly determined by the objectives of the user. The simplified approach allows for faster training, at the cost of reduced learning capabilities.

This is in general true not only for AIs but in general for mathematical models: the higher the number of parameters, the most tunable the model is; at the same time, the tuning process (that in the case of AI is the training itself) requires more significant effort. It is worth mentioning that, from a practical perspective, the simplified approach does not provide significantly worse results and it is valid (if not the suggested one) if no highly specific requirements are imposed.

Even if two edges do not intersect from a three-dimensional perspective, they may be seen crossing each other when viewed from the top. Such intersection are detected during the edge detection process.

In practical terms, typically the bogie is not structured on different 'layers' in the vertical direction. It is therefore uncommon for edges to be seen intersecting when the bogie is seen from above. The likelihood of such intersection is mathematically defined by p_i . This parameter is not analytically derived: it can be set by trial and error on a case-by-case basis.

In practice, the better option is to not alter the default value that has been obtained by extensive testing and has proven to generate valid results.

5. The structural element generation is repeated until a pre-determined stopping condition has been reached.

If no stopping condition was implemented, step 4. would be indefinitely repeated and no proper structure would be generated.

The choice of such stopping criteria significantly depends on the type of structure to be generate; that said, in general they express minimum requisites for the layout to be acceptable.

Proper selection and design of stopping criteria allows to avoid and preliminary discard solutions that would definitely not be viable, saving time during training by avoiding the user feedback process.

In the project: to assess when to stop iterations, two conditions are evaluated:

- **Connectedness of the structure:** a mandatory requirement for the bogie is that its frame must be fully connected. No edges or collection of edges that are not connected to the rest of the structure may exist.

If this criterion is not fulfilled, the generation process cannot stop, independently from other factors.

- **Number of edges:** while a solution may be valid from a connectedness perspective, it may not contain a satisfactory number of edges.

At each iteration of the edge addition process, the connectedness criterion is checked. If it is satisfied, the current number of edges present in the design is compared to the edge number map. A score is then computed for it and the selection process is performed.

If the latter is passed, edge generation is stopped and the obtained result is then allowed to progress to the successive phase.

If this is not the case, the edge generation and addition process is repeated.

6. Further structure editing: once the structural elements have been defined, additional elements or modification may have to be introduced.

In the project: the result of the previously described steps is a mostly bi-dimensional collection of nodes and edges. Furthermore, only half of the bogie frame is defined. Additional processing is therefore necessary to obtain the final design:

- Nodes that are not already provided with a z coordinates are three-dimensionalized.

As the z-coordinate of fixed nodes, such as those defining the primary suspensions, is set, the three-dimensionalization process does not apply to them.

For the remaining nodes, the height map is employed to determine missing z-coordinates. Refining this process is also part of AI training.

The result of this operation is a three dimensional lateral half of the bogie.

While not generating the free nodes and edges directly as three-dimensional objects may seem counterintuitive, this approach significantly improves the generation process.

As discussed, the shape of the bogie is mostly determined by its longitudinal and lateral components. Determining its shape first and expanding it in a later phase has proven to provide better and more coherent results.

It is worth mentioning that the heights map is generated by default with some pre-imposed constraints: as an example, heights that would conflict with other elements (such as wheels or axles) are not available for selection.

- The missing half on the xy plane is reconstructed by symmetry. Symmetrical nodes and edges can be obtained by simply negating the y coordinate of their positions.

The result of these final adjustments is a complete candidate design that can be presented to the user

7. The final design can be then interpreted by the user interface. In fact, due to the object-oriented approach of the G-Map environment, the objects that define the structure can be simply transmitted to the user interface; the latter is then able to provide its own interpretation and display the layout to the user, while maintaining the underlying mathematical structure.

Mathematical concepts in structure generation

Some tasks that are quite simple from a human perspective, are actually not trivial to be implemented from a mathematical and programming perspective.

Their practical realization can provide some insight into strategies that can be more generally adopted when dealing with the generation of structures.

Edge intersection detection: in general, intersection problems are known for their non-trivial nature. The solution of such problems is common for collision detection in virtual environments, such as computer graphics.

While from a human perspective detecting if two objects intersect is considered an easy task, even for simple constructs as edges this is not computationally as trivial. The procedure that is implemented in the project to detect if one edge $\mathbb{E}_1$ has any intersection is:

1. Another edge $\mathbb{E}_2$ is selected
2. Both $\mathbb{E}_1$ and $\mathbb{E}_2$ are converted to straight lines of the form $y = ax + b$. Lines t_1 and t_2 are respectively defined.
3. The intersection between t_1 and t_2 is calculated. If no intersection exists, the process is stopped.
4. The envelope of the two edges is considered, to determine if the intersection point lies on any of them. Only if the intersection point is part of both edges, it is recorded and added to a list of intersection
5. The process is repeated from 1. to 4. until each edge of the structure has been checked.

Structure connectedness: determining if the structure is connected is even more counterintuitively much more complex from a computational perspective than expected.

The problem of connectedness is a topological one; the structure can therefore be interpreted as a graph defining its topology. Nodes and edges in the structure are considered to be nodes and edges of a graph. The connectedness of this graph, and consequently of the structure, can be then determined.

The *adjacency matrix* A of a graph G containing N nodes is defined to be a $N \times N$ matrix such that:

$$A_{ij} = \begin{cases} 1 & \text{if node } i \text{ and } j \text{ are connected by an edge} \\ 0 & \text{otherwise} \end{cases}$$

The node numbers i and j are defined by their node ID. As an example:

$$A_{ij} = A_{ji} = 1 \implies \mathbb{N}_1 \mid id(\mathbb{N}_1) = i \text{ is connected to } \mathbb{N}_2 \mid id(\mathbb{N}_2) = j.$$

This is in fact one of the reason for the necessity of implementing such identification system.

For non-oriented graphs (this is the case for most structures), the adjacency matrix is by its nature symmetric.

The *degree matrix* D of a graph G containing N nodes is defined to be a diagonal $N \times N$ matrix such that:

$$D_{ii} = \sum_{k=1}^N A_{ik} = \sum_{k=1}^N A_{ki}$$

The *Laplacian matrix* L of a graph G containing N nodes is defined to be:

$$L(G) = D(G) - A(G)$$

As both D and A are symmetric, L is symmetric too. Its eigenvalues are therefore positive and real.

It can be demonstrated that the graph is connected if exactly one of its eigenvalues (no more than one) is zero.

The adjacency matrix A is initialized (i.e. created with all its entries equal to zero) once all nodes have been generated. Their number will in fact not change during edge generation.

As an edge is added to the graph, the adjacency matrix A is constructed and the D and L matrices from it. The eigenvalues of the L matrix are then calculated to check for connectedness.

Once the graph has been determined to be connected, this procedure is not performed anymore as the further addition of edges cannot disconnect the graph.

Evaluating the eigenvalues of a matrix can become significantly intensive from a computational perspective as the size of the matrix increases. Typical algorithms adopted for their calculation present computational costs in the order of $O(k \cdot n^2)$ or $O(k \cdot n^3)$.

While in this project the number of nodes is typically reduced, this could be an issue for structures involving more than hundreds or thousands of nodes.

In such cases it is advisable to adopt other connectedness checking algorithms, such as Breadth-First Search and Depth-First Search, whose cost is linearly dependent on the number of nodes. This approach has not been implemented in the project as incidence and adjacency matrices have to be evaluated anyway and the number of nodes is relatively small.

AI Learning

The learning process is realized by updating the generative maps based on the inputs provided by the user. As the user accepts or refuses a design proposal¹³⁶, relevant information is extracted from the model and maps are consequently updated.

¹³⁶ possibly after having modified it.

Mathematical definitions and concepts

The introduction of some definitions is necessary to properly describe and understand the map updating process at the core of G-Map AI learning.

Norm: in mathematics, a *norm* is a function $N(X)$ from a real or complex vector space X to the non-negative real numbers such that:

- $N(x + y) \leq N(x) + N(y) \quad \forall x, y \in X$
- $N(k \cdot x) = |k| \cdot N(x) \quad x \in X, k \text{ is a scalar}$
- $N(x) = 0 \implies x = 0 \quad x \in X$

Norms represent a generalization and extension of the concept of distance to the origin. In fact, the Euclidean distance fulfills all the above mentioned requirements and represents the most commonly adopted example of norm.

Grid neighborhood: considering a discretized n -dimensional workspace W_n operating under a norm N , the grid neighborhood $G(p, r, W_n)$ of a point $p \in W_n$ is defined to be:

$$G(p, r, W_n) = \{q \in W_n \mid N(p - q) \leq r\}$$

where r is the *radius* of the neighborhood

Hollow grid neighborhood: an hollow grid neighborhood of *thickness* t is defined to be:

$$H(p, r, W_n, t) = G(p, r, W_n) \setminus G(p, r - t, W_n)$$

Update coefficients function: the update coefficients function $u = f_u(r, \mathbb{M})$ of a map $\mathbb{M}$ associates an update coefficient u to a given neighborhood radius r .

Ordering function: an ordering function $O(X)$ maps an n -dimensional input space X of a map $\mathbb{M}$ to a vector space V on which a norm can be defined.

Updating process

The map updating process represents the core of AI learning in the G-Map approach. Each map is typically representative of some feature of the structure to be designed¹³⁷; information about such features is extracted from the model that has been edited, accepted or rejected by the user and maps are consequently updated.

If a design is accepted, it is assumed that its features are considered to be desirable by the user. The probability factors of the corresponding maps are therefore updated to increase the likelihood of such features to manifest.

If a design is rejected, its features are assumed to not be desirable. Two approaches may be adopted¹³⁸:

- The map is updated so that the likelihood of the undesired feature to occur in proposed designs is decreases¹³⁹
- The map is left unchanged¹⁴⁰

The update process typically does not involve only one input from the input space but neighboring elements too¹⁴¹. The underlying assumption is that if the probability of a certain input is increased, it is reasonable to assume that the probability of similar inputs should be increased too.

even if this is the case, some form of weighting is usually advisable: if an input is selected for update, the further other inputs are, the lower the collateral effect on these should be¹⁴².

This weighting effect is provided by the update coefficients function: it expresses how the update of an entry influences its surroundings as a function of the neighborhood radius.

More rigorously, the update procedure can be described as follows:

1. The value (or values) $x \in \bar{X}(\mathbb{M}(u))$ that have chosen for the variable u that map $\mathbb{M}(u)$ describes are evaluated from the design that has been accepted or rejected by the user¹⁴³.
2. A vector space $V = O(X(\mathbb{M}(u)))$ is defined. A norm N is pre-determined and intrinsically programmed for each map and its corresponding vector space.

In practical terms, the ordering function allow to provide a measure of 'how close' two map inputs are.

In the case of uni-dimensional input spaces, often the input is represented and stored as a list. The list index of an input can therefore be considered as its position and nodes with closer indices are considered to be closer.

Two- and three-dimensional input spaces are typically representative of coordinate spaces; their conversion to a vector space is therefore simply the association to the coordinates they represent.

Due to the wide variety of features that maps can represent, no general rule exist. That said, some recurrent themes, such as the above described, represent typical examples that can be found in practice.

¹³⁷ Examples are its number of edges, nodes or other characterizing elements and their respective position.

¹³⁸ In fact, in the project some maps are updated following one approach, others adopting the other solution. This choice depends on the type of map and the feature it is representative of.

¹³⁹ Typically, yet not necessarily, in this case updates are the opposite to those in the case of an accepted design.

¹⁴⁰ In some cases, avoiding excessive penalization of undesired features may be more productive.

¹⁴¹ Thus the necessity for the definition of 'closeness' between inputs in the input space.

¹⁴² To provide a practical example: if a certain position is chosen for secondary suspension by the user, it is reasonable to assume that similar locations should be viable options too. Neighboring points should therefore be updated too. That said, they should not be updated with the same weight as the selected one, especially if they are relatively further from it.

¹⁴³ As an example, if the map describes the expected position of an element, the position of the latter in the model is extrapolated.

As the input space $X(\mathbb{M}(u))$ is by definition discretized, the resulting vector space V is discretized too and represents a valid workspace to define grid neighborhoods.

3. The probability factor $P(p \in X) = \mathbb{M}(p)$ is updated as follows:

$$O(p) \in H(O(x), r, V, t) \implies P'(p) = P(p) + f_u(r, \mathbb{M})$$

where $P'(p)$ is the updated value for $\mathbb{M}(u)$. Typical update functions are such that

$$\begin{cases} |f_u(r_0, \mathbb{M})| \geq |f_u(r_0 + \Delta r, \mathbb{M})|, & \Delta r > 0 \\ r > r_{max} \implies f_u(r_0, \mathbb{M}) = 0 \end{cases}$$

By imposing such conditions, the updating effect diminishes the further from the value extrapolated from the model and is null above the maximum radius r_{max} .

Map updating in the project

In the project, all maps are updated following the previously described procedure. Some general properties are set for all maps and subordinate elements:

- The thickness t of all hollow neighborhoods is set to be $t = 1$. Each hollow neighborhood in the update process is therefore one unit thick.
- The norm that is adopted for all maps is the maximum or infinity norm, defined to be:

$$N(x) = \lim_{p \rightarrow \infty} \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}} = \max(|x_i|)$$

While different maps require different approaches, common themes can be observed, especially among maps of the same type. The distinction in categories will therefore be maintained when discussing their updating process too.

Uni-dimensional maps: in the context of uni-dimensional maps, the concept of maximum norm just reduces to the absolute value of the difference between the input values: $N(x - y) = N(y - x) = |x - y|$

As in the project the inputs space for uni-dimensional maps is in the form of an array, the index $idx(x)$ of an input $x \in X$ is adopted for the conversion to vector space: $O(x \in X) = idx(x)$

It follows that:

$$O(y) \in H(O(x), r, O(X), t = 1) \implies idx(y) = idx(x) \pm r$$

Two $\mathbb{M}(x)$ values at most¹⁴⁴ are therefore updated for each hollow neighborhood.

If the update coefficient function in the case of accepted design is represented as $f_u^+(r, \mathbb{M})$, its rejected design counterpart is typically either $f_u^-(r, \mathbb{M}) = -f_u^+(r, \mathbb{M})$ or $f_u^-(r, \mathbb{M}) = 0$

¹⁴⁴ In fact, defining the minimum and maximum indices of the input array to be i_{min} and i_{max} respectively, it must necessarily be that:

$$i_{min} \leq idx(y) \leq i_{max}$$

Depending on the input x and on the radius r , some y values may be out of bound for the array and therefore be undefined or non-existing.

- **Transversal elements map:** the number n_t of transversal elements in the model is counted. n_t is then considered as the center of the neighborhood to be updated. If the design is accepted, the probability of the number of transversal elements N_t to be $N_t = n_t$ or $N_t \sim n_t$ is increased. If the design is rejected, the opposite is true.
- **Node number map:** the number n_f of free nodes present in the model are counted. The same procedure that is adopted for transversal elements map is adopted.
- **Edge number map:** same as the node number map, with the only difference that the number of edges n_e is evaluated and its corresponding map updated.

Two-dimensional maps: in the case of two-dimensional maps, the maximum norm adopted in the definition on neighborhoods implies that the latter are squares of side length $l = 2r + 1$.

Hollow neighborhoods $H(p, r, O(X), t = 1)$ are, from a geometrical perspective, the difference between two concentric squares of side-length $l_1 = 2r + 1$ and $l_2 = 2r - 1$; they therefore contain $8(r - 1)$ elements.

Due to the mathematical generality of the update procedure described in the previous section, minimal adaptations are needed for its application to two-dimensional maps.

- **Points map:** the points map is updated for each free point. If a design is accepted, the probability factors of its free nodes and nearby points is increased. The opposite is true if the design is rejected.
- **Secondary suspensions map:** the same approach of the points map is employed, considering secondary suspensions instead

Two-dimensional nested maps: due to the increased complexity of their structure, a different update procedure is required for two-dimensional nested maps.

Inputs of the form $\mathbb{X} = \{q = (i, j), k\}$ are required for the update of two-dimensional maps. The tuple (i, j) is required for indexing and k is the update parameter.

For a certain update radius r_{upd} , the set of first order maps to be updated in a two-dimensional nested map $\mathbb{M}^2$ is defined to be:

$$M_{upd} = \mathbb{M}^2(G(q, r_{upd}, \bar{X}(\mathbb{M}^2)))$$

All maps $\mathbb{M} \in M_{upd}$ are then updated as one-dimensional maps¹⁴⁵ with k as their updated input¹⁴⁶.

Typical radii for one-dimensional maps are relatively small (at least in the project) as the input space is quite small in size.

To provide a practical example, the size of the transversal node map is around ten inputs. A neighborhood radius $r = 2$ would imply that approximately half of the inputs are involved in the update process; this may be excessive unless the update function is properly tuned to reduce the update effect at the extremities.

Fixed nodes are not considered in the points map update procedure: as their position does not change, they would alter the statistics of the node location selection.

Typically, two-dimensional maps updates involve a larger number of inputs with respect to lower order maps. Neighborhood radii are usually larger too, to extend the influence of the update to a larger subset of the input space.

¹⁴⁵ following the previously described procedure.

¹⁴⁶ To reduce complexity, in the project all maps $\mathbb{M} \in M_{upd}$ undergo the same update. This has proven to significantly reduce complexity with no perceivable reduction in performance.

- **Height map:** for each free point $p = (x_p, y_p, z_p)$, the input is defined to be $\mathbb{X} = \{q = (x_p, y_p), z_p\}$.

In practical terms, for each free point $p = (x_p, y_p, z_p)$ the probability of the height of a point z to be $z = z_p$ or $z \sim z_p$ is increased or decreased for points such that their (x, y) coordinates are close or equal to (x_p, y_p) if the design is respectively accepted or rejected.

The extent of this effect depends on the update radius r_{upd} chosen for the map.

- **Edge direction map:** for each edge extreme $e = (x_e, y_e, z_e)$, the update input is defined to be $\mathbb{X} = \{q = (x_e, y_e), \theta\}$.

The tuple q represent the position of e when flattened on the xy plane. The angle θ is representative of the orientation of the edge with respect to e .

Maps $\mathbb{M}(x, y)_\theta \in M_{upd}$ are representative of the probability of an edge with one of its extremes at coordinates (x, y) to assume an orientation θ .

As orientation θ is continuous, it is necessary to perform some kind of interpolation to adapt it to the discretized nature of the map input space; this is valid both during generation and the update process. Different approaches can be adopted; some examples are:

- The input $\theta_X \in X(\mathbb{M}_\theta)$ closer to θ is updated while other inputs are not altered.
This solution is simpler yet it may offer a sub-optimal performance
- Some form of norm N for the input space $X(\mathbb{M}_\theta)$ is defined and other inputs $\theta' \in X(\mathbb{M}_\theta)$ are updated depending on $N(\theta' - \theta)$.
The implementation of this solution is more complex can be avoided if not necessary for proper performance.

Once all maps have been updated, the iteration loop can restart from the generation phase.

As maps mutate, results typically differ iteration by iteration¹⁴⁷. As the training progresses, though, generated layouts tend to converge to desirable outputs and practically reasonable designs.

¹⁴⁷ This is especially true in the iteration, when the AI is not trained: major correction are in fact introduced. As the training process progresses, correction should become less extreme as the focus of the training shifts to minor refinement of learned knowledge.

Part IV

User Interface

Requirements and implementation

The GUIDE approach considerably relies on user interaction; the UI¹⁴⁸ should therefore allow for a wide range of operating capabilities and features to properly fulfill its requirements.

To realize a proper UI implementation it is therefore first of all necessary to assess its operating requirements and then devise solutions for their practical implementation.

Structure manipulation

The UI should allow the user to easily manipulate three-dimensional structures. As mentioned, AI generation adopts so-called *reference elements* to define the geometry of structural components. These reference elements are manipulated by the user to edit the design proposed by the AI.

Structure manipulation may also imply the ability by the user to modify the mechanical properties of the structural elements¹⁴⁹.

In the project: nodes represent the control elements and edges the structural ones; the overall layout is therefore edited by controlling the position of the nodes. As nodes represent the extremes of edges, corresponding edges are consequently edited.

In the project, the user interface is required to:

- Move free nodes in all three dimensions. The edges that connect them should change their extremes accordingly
- Add or delete nodes. In the latter case, the system should be able to recognize which edges are connected to the node and delete these too
- Add or delete edges
- Move nodes that have special characteristics while preserving their characteristics and constraints

Mathematical concepts in structure editing

While the above described objectives may seem straight-forward, they are in fact non-trivial in their implementation

The tool is not only required to know the position of the nodes, but their type and the topology of the structure too. Furthermore, as the user modifies the structure, such information should be updated in real time. To achieve this goal, different mathematical constructs are introduced.

¹⁴⁸ User Interface

To maintain as much as possible a general perspective, requirements will be considered to be valid to as many possible design objectives as possible. Clearly, the practical example implementation is instead limited in scope to railway bogies.

¹⁴⁹ The control of mechanical properties is limited in this project and they are typically set by default; nevertheless, in other contexts this could be a relevant feature.

While these requirements are introduced for the manipulation of a node-edge structure, they are however easily adaptable to other contexts. In fact, even if different structural elements may require different manipulation approaches, the overall guiding principles are the same.

The same is true for solutions that are adopted for their practical implementation: they can be easily adapted and applied to other structural elements and geometrical/physical models.

To provide a practical example: if during editing the movement of type of node is constrained on one of its axes, this limitation should be respected. The UI (and the software in general) should not only limit its motion to expected axes but preserve the nature of the node: even if the latter has changed its position, information about its type should be preserved.

Attributes system: the adoption of identifiers and attributes for nodes and edges allows to uniquely determine their identity and track their properties and attributes as they are moved and edited. Some relevant examples are:

- **Node ID:** IDs are employed to identify a node or an edge and to determine their respective connections to each other.
- **Type:** both edges and nodes may have a type descriptor that defines its constraints and characteristics.
Due to the definition of node and edge objects, their position-defining component can be modified without altering its attributes. The latter are therefore maintained even after shape modification.
- **Other object properties:** additional properties may be connected to an element. As discussed, as an example, the mechanical properties of structural element might be defined or definable by the user.

Incidence matrix: recalling the structure of edges to be $\mathbb{E}(\mathbb{N}_1, \mathbb{N}_2, [Att])$, the incidence matrix B of a N -nodes, M -edges graph G is an $N \times M$ matrix such that:

$$\mathbb{E}(\mathbb{N}_1, \mathbb{N}_2, [Att]) \implies \begin{cases} B_{ij} = 1 \\ B_{kj} = -1 \\ i = id(\mathbb{N}_1), k = id(\mathbb{N}_2), j = id(\mathbb{E}) \end{cases}$$

$$B_{nm} = 0 \implies \nexists \mathbb{E} \mid id(\mathbb{N}_{1,2} \in \mathbb{E}) = n \wedge id(\mathbb{E}) = m$$

The incidence matrix uniquely describes the topology of the graph. A biunivocal correspondence exists between node-edge arrangements and their respective adjacency matrix.

The incidence matrix of the structure is constructed at the end of the generation process. Since the topology of the structure changes as nodes and edges are added or deleted, the B matrix must correspondingly be updated.

In the project, node types determine:

- The order in which they are generated and their constraints for their introduction
- The constraints on their editing
- Their mechanical function and the corresponding effects on static and dynamic analysis (e.g. the loads to be applied)

The type of the edge object can be set to allow it to represent:

- A structural edge element
- A spring
- A damper

This implementation of the incidence matrix assumes the graph to be oriented.

This is typical of applications in which the verse and general directionality of edges is relevant, as in flow networks or electrical circuits.

If this is the case, the sign of matrix entries can be employed to express the direction of the edge.

In this project, this is not the case as from a structural and physical perspective edges have no orientation.

Nevertheless, edge directionality is still adopted to easily determine which of the extremes of the edge to edit if the corresponding node is modified.

The reason for the choice of this approach is therefore not physical but computational and mathematical in nature.

The general procedure for reference and structural elements editing is¹⁵⁰:

1. The user selects a reference or structural element.
This step is not always required: as an example, the addition of new element may not require other elements to be selected.
2. The inputs provided by the user through the keyboard, mouse or other input devices is converted to an editing command for the selected element or to the addition of a new element.
3. The applicability to the element or to the workspace of the required input is verified¹⁵¹.
4. The position or other properties of the element are updated accordingly to what has previously determined.
5. Information about the structure is updated accordingly to the new properties of the edited or added element¹⁵².
6. The graphical representation of the structure is updated.

¹⁵⁰ clearly some minor variations may exist depending on the context, yet this procedure is considered to be broadly representative of most implementations.

¹⁵¹ e.g. if the input is a movement along one axis and the selected element is a node, it must be verified if the node type and characteristics allow for such type of movement.

If an element is to be added, it is necessary to verify if it can actually be added in that position.

¹⁵² As an example, the incidence matrix requires to be updated.

In the project: the user interface allows for the editing, addition and deletion of both edges and nodes.

Node editing: as in this implementation nodes are employed to control the position and shape of structural elements (edges), their movement allows to edit the shape of the design. The actions that can be performed on nodes are:

- **Node movement:** if a node has been selected and the user provides a movement input, a check is performed to assess if such movement would be compatible with the node type. Nodes determine the shape of edges of which they are the extremes. If a node is moved, edges connected to it must be update accordingly.

The incidence matrix allows to determine which edges are connected to each node and which of the two edge extremes it represents.

- **Node addition:** the addition of a new node does not require any node to be selected. The user decides the (x, y) coordinates for the new nodes on the graphical interface, while the z coordinate is determined by default during initialization and can later be edited. Node attributes are also defined by default; in terms of node type, they are set to be free nodes.

A relevant attribute to be generated is the node ID: it must be coherent with existing ones and avoid any superimposition. To preserve topological coherence too, the incidence matrix must be consequently updated introducing a new row for the new node. At creation, new nodes are not connected to any edges; new connecting edges can be added by the user as desired.

- **Node deletion:** if a node has been selected, it can be deleted. The elimination of a node also implies the deletion of all edges connected to it, which have to be determined through the incidence matrix. The latter has then to be updated considering the IDs of the deleted nodes and edges.

Edge editing: the approach adopted for the editing of edges is similar to what has been discussed for nodes.

- **Edge movement:** the movement of edges and the modification of their shape is commonly realized by the editing of the nodes at its extremes.
- **Edge addition:** edges can be added between already existing nodes. A new edge object is created with default attributes and a new edge ID coherent with existing ones. A new column is added to the incidence matrix to accommodate for the new edge.
- **Edge deletion:** the removal of an edge does not automatically delete the nodes at its extremes. The only necessary update to the incidence matrix is therefore the deletion of the corresponding column.

As mentioned, after the whole structure has been updated with the new information, the graphical representation is updated too.

In practice, the whole update process (both mathematical and graphical) is very fast and it is barely perceptible by the user. The interface allows therefore for smooth operation.

Graphical visualization

The UI should allow to effectively graphically represent proposed designs to allow for proper editing. This may not be easy for complex 3D structures: as it will be discussed, bogies too require two interactive windows for proper visualization and manipulation.

Despite the significant dependence of the actual implementation on the nature of the design to be realized, some broadly valid requirements are:

- The user should be able to easily determine sizes, distances and relative positions of elements of the structure
- Elements should be selectable and editable with reduced effort
- The graphical representation should provide a complete representation of the whole structure; as few as possible elements should be out of sight or difficult to interpret.

In the project: two windows allow for graphical visualization. One of them provides a schematic 2D representation of the model and allows for structure manipulation; the other provides a more realistic, albeit stylized, representation of the model.

Due to the extreme variety of 3D systems that could be implemented, it is difficult, if not impossible, to devise a single type of visualization layout capable of handling such a large set of geometries and structures.

The approach adopted in this project is therefore broadly representative of how such a graphical interface could be implemented and has the goal of providing overall guidelines, yet implementations have to be properly adapted on a case by case basis.

Analysis

The UI should allow the user to request analyses of the structure. Required capabilities are therefore not limited to starting the analysis, but allowing the user to choose the analysis mode¹⁵³, introducing parameters for the simulation¹⁵⁴ and selecting which results to display from the available ones¹⁵⁵.

In the project: the user can request a static analysis of the design. Different load configurations can be selected by the user to evaluate the behavior of the bogie under different operating conditions. Results are then presented to the user in a easily readable format. Furthermore, some structural parameters are tunable, while other require a more complex intervention on the software.

¹⁵³ e.g. different types of static analysis may be available, choosing between static and dynamic analysis and in general selecting one type of analysis from the available ones.

¹⁵⁴ e.g. spring stiffnesses and diameters, geometrical properties, damping coefficients, etc.

¹⁵⁵ in fact, a quite large set of data maybe available while the user may only need to focus on a specific variable.

¹⁵⁶ In theory, the user could reject the corrected design too. This is unlikely, though, as after being modified and adapted by the user it is typically more coherent with user expectations.

That said, the user could reject it too. In fact, the user could deliberately produce undesirable layouts and reject them to train the AI to avoid such solutions.

¹⁵⁷ this is typical of the first iteration starting from an untrained AI setup: solutions provided by the AI may be chaotic or in general not at all viable and it may be not worth the time correcting these. Furthermore, it is an occasion for the AI to learn which solutions to avoid.

Feedback on AI proposed designs

At each iteration of AI generation, the user should be able to:

- Directly accept the solution provided by the AI, if it is deemed to be already satisfactory
- Correct/modify the solution provided by the AI and then accept it¹⁵⁶
- Reject a solution if it is deemed to be excessively far from correctness¹⁵⁷

The AI should then be updated: if the solution has been deemed valid it should learn from it; if it has been rejected, the AI should learn to avoid such outputs.

The procedure for the learning process is the one described in the *AI learning* section.

In the project: the user can accept, reject or even just skip a proposed or corrected design. In the latter case, no effect on the training is introduced.

The learning process adopted in the project is the same as considered in the *AI Learning* section.

Saving and loading data

If the user is satisfied with an output or a certain training configuration, it should be possible to save these in an external file for later use.

As the objective is the design of systems/structures, the user should be able to obtain saved outputs from which to continue the design process: such files should therefore contain the geometrical description of the system for later use and development.

The user should also be able to load previous configuration of the AI or to start from appositely defined ones.

Maps allow for easy exportation of AI training states. As from a practical computational perspective they are represented by tables or more complex array structures of numbers, it is enough to export the underlying numerical data in an appositely defined format. Data can then be read back and imported during later use.

The export of desirable designs represent a more complex matter and in this project is left in a semi-open ended state. In fact, the format to be adopted and the parameters to save can significantly depend on the objectives of the user.

One possible solution is to export all edges and other structural elements by simply listing their coordinates and relevant features in a simple text file.

More complex exportation procedures, e.g. to CAD software, require appropriately designed user interfaces and have to be conceived and realized on a case-by-case basis.

Interface and Controls

The graphical visualization of three-dimensional structures and the interaction with the latter requires some careful consideration to avoid typical issues that may arise.

The representation of a 3D object on a device screen requires a 3D-to-2D conversion; in the dimension reduction process, some information is inevitably lost.

In fact, the most typical approaches¹⁵⁸ to perform such a conversion require a significant subset, if not the entirety, of the resulting 2D points to be associated with an infinite number of 3D points. As a consequence, part of the information about the 3D location of the point is lost ¹⁵⁹.

Undesirable consequences of these observations are:

- Visualizing the whole structure at once may be difficult, especially in the case of complex design with a large number of intersection and overlapping elements.

Allowing the user to rotate and, in general, modify interactively the view of the structure can partially alleviate the issue; it is also true that this approach may not be sufficient for structures with complex internal geometries¹⁶⁰ that may require more complex visualization methods.

- Even if some form of perspective is introduced to facilitate depth perception, assessing relative distances and dimensions may prove problematic¹⁶¹.
- Due to dimensionality reduction, nodes that have different position may appear to be coincident in the projection; while this is an inconvenience in the context of visualization, it becomes very problematic when interacting with the structure: selecting a point in the two-dimensional projection is equivalent to selecting an infinite amount of points in the corresponding 3D space.

While these issues can be mitigated by the introduction of appropriately designed selection systems, the degree of complexity is significantly increased.

Due to the above described complexities and in general the intricacies of proper visualization and interaction, user interaction has to be implemented on a case-by-case basis, taking into account the relevant characteristics and defining features of the structure to be designed.

User interaction in the context of static and dynamic analysis will be described in detail in the dedicated section, as it requires contextualization for proper understanding.

In this chapter, the focus will mainly be on structure visualization, interaction and modification.

¹⁵⁸ These are also the ones that are best suited and actually implementable in graphical visualization.

Exceptions may exist yet they are limit cases and represent more mathematical curiosities than actually implementable solutions.

¹⁵⁹ One typical occurrence in orthogonal projection is that information about the position along one of the coordinate axis is lost. Other projections introduce different types of indeterminacy, yet in general a degree of freedom is added to the location of a point.

¹⁶⁰ that cannot be easily visualized from an exterior view; this is not the case for railway bogies, yet it represents a relatively common occurrence in mechanical design.

¹⁶¹ Coherently with most technical drawing guidelines, perspective is avoided. While it may be useful in some context, the graphical distortion that it introduces can impair the evaluation of sizes and distances.

As mentioned when describing the AI generation process, a relevant feature of railway bogies is that they are prevalently extended on their longitudinal and lateral axes, with relatively small development on the vertical direction.

The approach that has been developed for this project is to employ two projections to display and interact with the model.

- A 2D dimensional top orthogonal projection allows for easy editing and determination of relevant dimensions and positions; this is defined to be the *interaction window*, as it allows the user to edit the design.
- The *3D rendering window* provides the user with a more refined 3D rendering of the structure.

As the structure is edited in the interaction windows, changes are also displayed in real-time in the 3D rendering window.

To improve clarity, the same color scheme is adopted for both visualizations:

- Blue: primary suspensions springs
- Orange: secondary suspensions
- Red: mechanical link / selected nodes
- Gray: general structural elements

3D rendering window

The 3D rendering window provides the user with a 3D graphical rendition of the bogie. This visualization allows the designer to obtain a clearer picture of the current design and the modifications that are applied to it.

This window does not allow the user to modify the structure: as mentioned, the editing of the model is significantly dependent on node selection which is difficult to properly implement in a three-dimensional environment.

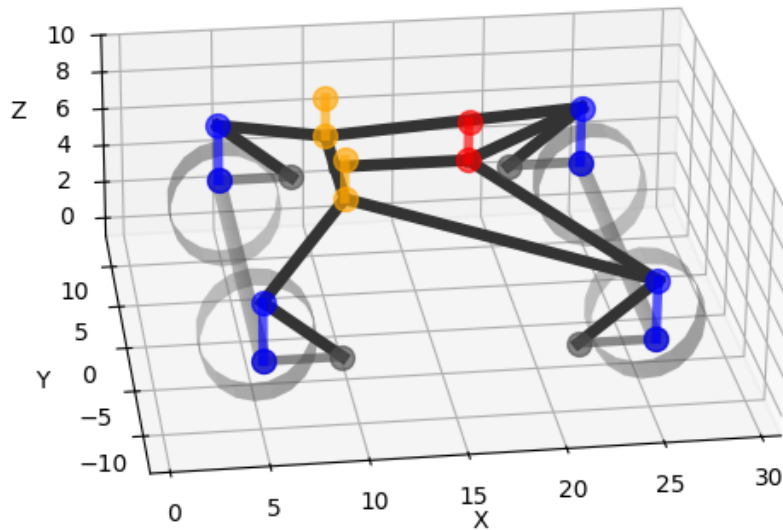
The goal of the rendering window is therefore not to allow for design modification but to provide an interactive view of the model: the user can move, rotate and zoom on it to better assess the overall shape and layout of the design¹⁶².

As previously described, a common issue in 2D projection is that elements might hide other elements that lie behind them¹⁶³. This is partially solved by implementing so-called *partial transparency*: springs and other suspension elements are only in part opaque and allow to partially see behind them. The same is true, even if to a slightly lesser degree, for other structural elements. This allows to see the whole structure at once while maintaining some sense of depth and layers¹⁶⁴.

¹⁶² As mentioned, this approach may be insufficient for more complex components and systems, as it visualizes its exterior; this solution is more than enough for bogies, though, as their overall shape is typically far from this level of complexity.

¹⁶³ with respect to the view point.

¹⁶⁴ This solution may prove ineffective for components that involve many overlapping layers, as the representation may soon become unreadable.



Interaction window

The interaction window allows the user to modify the structure and to access other features and controls. Its most relevant features are:

- The model is shown as a 2D orthogonal projection from the top. This might seem a significant limitation to the user interaction, yet on the contrary it facilitates it.

As mentioned, the height component is typically smaller and more constrained when compared to other dimensions. Its omission in the 2D orthogonal projection does not represent a significant issue for the user, though, as vertical dimensions can be determined from the 3D rendering window.

The 2D projection approach allows the user to quickly determine and modify the shape and layout of the bogie. Practical testing has shown model editing to be quicker and less cluttered when carried out in such a 2D environment¹⁶⁵.

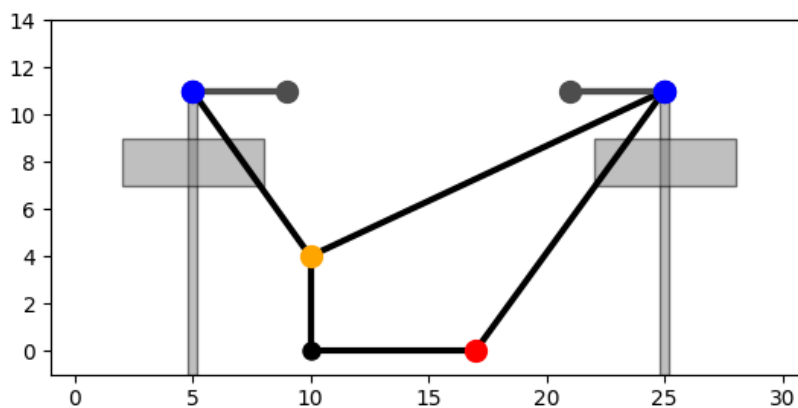


Figure 25: 3D rendering window representing an example model.

An orthogonal projection is adopted to avoid size and position distortions typical of perspective.

Exact geometries and type of secondary suspensions and of the mechanical link is ignored: they are represented only in terms of their connection to the bogie frame. As it will be later discussed, this solution has no negative effects while greatly simplifying design and avoiding unnecessary clutter.

Notably, a lighter shade of gray is adopted for the arm element of primary suspensions, while the aforementioned color code is adopted for other elements.

Reference axes are indicated, while axes units are not as they are set by the user and assumed to be well-known and defined. The default unit for axes is 100 mm.

¹⁶⁵ As previously discussed, 3D interaction presents many challenges. Furthermore, 3D visualization may appear more cluttered and less interpretable.

The determination of sizes and dimensions, which are very relevant in the design phase is more difficult too.

Simplifying the projection allows for easier interaction; the loss of the vertical component can be easily compensated by the 3D rendering window.

Figure 26: Corresponding interaction window of the model represented in Figure 25.

The view is fixed and height information is lost; on the other hand, ease of selection and manipulation is greatly improved and dimensions can more easily be assessed.

Axes direction are not indicated as they are fixed and well-defined; the same reasoning as in Figure 25 is valid for axes units.

Most notably, only one lateral half of the bogie is represented.

- Only one half of the model is displayed: the bogie is assumed to be symmetric with respect to its longitudinal axis. One lateral side only is displayed, with three main advantages:
 - The visualization is more compact
 - If the user desired to move one node, its corresponding symmetric¹⁶⁶ one should be moved too.
If this was performed by the user, it would represent a significant and unnecessary increase in time and effort requirements. Even if an automated solution could be adopted¹⁶⁷ this has been deemed to be an unnecessary addition of complexity¹⁶⁸.
 - Coherence with AI generation: as the same approach is adopted during AI generation¹⁶⁹, this solution allows for easier interfacing between different software subsystem¹⁷⁰.

This choice is therefore not dependent on any visualization limitation¹⁷¹.

- The editing process should be coherent with the workspace discretization. The movement of nodes should therefore be limited only to points that are included in the workspace.
The minimum resolution of the movement of a point on an axis is therefore represented by the grid resolution on that same axis.

In the project: a uniform grid resolution $g_{x,y,z} = 100 \text{ mm} = 1 \text{ dm}$ is adopted along all axes.

While it is true that a finer grid allows for more precise control, it also implies increased computational, memory and training requirements. In fact, as the grid resolution decreases in size, the number of map elements is significantly increased, requiring more iterations to obtain valid results and more memory to store data.

At each movement key input the node position is moved along the corresponding axis and direction by a grid unit. As a consequence, its minimum displacement along each axis is 10 cm.

- Axes allow the user to effectively determine positions and dimensions. The default axis unit is decimeters and it is applied to the 3D rendering view too¹⁷². While this choice may seem unusual, it is worth considering that:
 - The grid unit size $g_{x,y,z} = 1 \text{ dm}$. Axes therefore represent a positional and dimensional reference both in terms of physical length and grid units
 - The most commonly adopted unit in bogie design¹⁷³ is the millimeter.
Bogie dimensions are typically in the order of meters: numbers on the axes would therefore be in the order of thousands if millimeters were to be adopted.
Introducing decimeters as axis units removes unnecessary zeros and non-significant digits from axes labels, providing a less cluttered and more readable interface.

¹⁶⁶ with respect to the longitudinal axis.

¹⁶⁷ i.e. as one node is selected and edited, the corresponding action is performed on the symmetric node too.

¹⁶⁸ While this is not the case in the project, in systems with large amounts of nodes this simplification can significantly reduce computational time and effort.

¹⁶⁹ only half of the bogie is generated, while the remaining part is obtained by symmetry.

¹⁷⁰ this is mostly a coding and mathematical matter; it is not easily explainable out of the context of underlying software code.

¹⁷¹ In fact, the full layout is displayed in the rendering window. Graphical presentation is therefore not an issue; this is even more true in a simpler 2D environment.

The grid resolution adopted in the project has been deemed a reasonable compromise between precise editing and computational cost.

Nevertheless, if required its value can easily be adjusted, as the latter is not a consequence of any constraint and has been chosen only because of its practical convenience.

¹⁷² Even if the 3D rendering window has a lesser role in size and position determination, it is still relevant to have meaningful reference axes. This is especially true when considering its role in assessing vertical position.

¹⁷³ and in general in most engineering applications.

Controls and interaction

The controls that are available to the user are:

- **Node editing:** the user can select a node by *left-clicking* on it. The selected node, if any, is indicated to the user by changing its color to red.

Once a node has been selected it can be:

- moved, by pressing the *arrow keys* for x- and y-axis movement and the *page up/down keys* to control its height. Each press of the key moves the point by one grid unit. To move by more than a grid unit it is either possible to press the key repeatedly or just to keep the key pressed.
- deleted, by pressing the *'delete'* key. As a node is deleted, all edges connected to it are deleted too.

Furthermore, a node can be added by *right-clicking* on an empty area of the workspace. The position of the node is automatically adjusted to the nearest available point of the workspace.

The z-coordinate of the point is set to a default value and can be later adjusted.

- **Edge editing:** edges can be either added or deleted. To do so:
 1. One node is selected by left-clicking on it
 2. Once the node has been selected, the other node is right-clicked on
 3. If an edge between the two nodes already exists, it is deleted; if that is not the case, an edge between them is added.

- **AI feedback:** as mentioned, the user can either accept or reject a design.

If the user is satisfied with the current design, (either as proposed by the AI editing or after editing) the latter can be accepted by pressing *'enter'*.

Following the same logic, designs can be refused by pressing the *'r'* key and skipped by the *'i'* key.

In all cases, the AI registers the current configuration and learns from it as described in previous chapters; it then proposes a new layout.

- **Analysis request:** as static analysis tools have been implemented in this project, the user can access these functionalities by pressing the *'a'* key.

The analysis tool has its own user interface, designed for mode selection and user data input.

- **Data memorization:** the user may desire to save relevant information for later use. In particular:

- the current AI state of training can be saved by pressing the *'z'* key
- the current design can be saved by pressing the *'s'* key

In both cases, data is stored in a dedicated folder.

AI training data is saved in a proprietary format of the tool. Saved designs can be exported to be available for later use.

- **Other functionalities:** other accessory functionalities are present. As an example, pressing the *'k'* key closes the tool and stops operations.

As for previous characteristics of the implementation, the choice of axis unit is not due to any technical limitation in fact, the interface can be set to millimeter or other standards with no significant adjustments required. The objective of this choice is in fact limited to improving user experience and readability.

A further proof of the validity of the chosen grid resolution is that node movement is quite smooth and the discretization of the underlying grid is barely perceptible.

Newly introduced nodes have to respect the workspace discretization too. Their position is therefore snapped to the underlying grid as they are created.

As the bogie does not typically extend much in the vertical direction, the editing of the height of newly introduced points does not imply a significant effort by the user.

Part V

Static Analysis

Introduction

The GUIDE approach includes the concept of numerical analysis of the proposed design. The user should be able to assess the performance of the current layout both immediately after its generation and during the editing process; this is in fact necessary to guide the corrections to the model when needed, and to identify the best candidate solutions when providing feedback.

The design process of most mechanical structures requires some form of static analysis. In fact, while some components may not be exposed to time-dependent phenomena¹⁷⁴, some degree of static evaluation is commonly required to evaluate the response to loads that are encountered during operation.

A static analysis tool has been implemented to showcase how it could be integrated in the GUIDE approach. The main features of this implementation are:

- As the bogie structure is generated and edited as a collection of nodes and edges, the same approach is adopted for its static analysis: edges object are interpreted as beam elements, while other structural elements are treated as part of their respective category¹⁷⁵.

The node and edge attributes system allows to easily retrieve information about both the geometrical and mechanical properties of all elements.

- The focus of the analysis is on displacements¹⁷⁶. The deformed configuration is computed through Finite Element Analysis (FEA) and relevant quantities are evaluated from it¹⁷⁷; this process thus requires further elaboration of the obtained displacements and deformations.

While the loads on some elements, such as springs and torsional elastic joints, can be computed as a function of displacements, in general the computation of stresses is avoided: to properly evaluate the stress distribution an accurate model of the structure is necessary. In fact, notch effects, welded joints and other inhomogeneities or localized features can have a very relevant effect. Due to the preliminary design focus of the tool, this has been deemed unreasonable.

This limitation is not too impactful, though: in the preliminary phase of bogie design, the focus is often on evaluating displacements in components such as suspension systems. Furthermore,

¹⁷⁴ such as dynamic loads/displacements, fatigue or more complex phenomena such as creep and wear.

¹⁷⁵ Some examples are springs or load carrying nodes, such as the connection to secondary suspensions.

¹⁷⁶ this is to be interpreted in the generalized sense: displacements include both nodal translations and rotations.

¹⁷⁷ outputs of the analysis and their significance in the context of this project will be presented in following chapters

while the exact stress distribution is unknown, its overall behavior can be at least somewhat assessed. In fact, while local effects cannot be model, the global trend of the stress distribution is often related to the displacements¹⁷⁸.

- The primary suspension system is, by its nature, non-linear both from a static and dynamic perspective. A proper approach to adapt FEA, which is oriented to linear system, to model non-linear components had therefore to be developed and implemented.

Modeling a complex structure, such as the frame of a bogie and its other components, as simple beam elements may seem an excessive simplification with respect to the sophistication of a finished design.

In spite of such expectations, actual practical testing and analytical consideration have proven this relatively simplified approach to be more than enough with respect to the degree of accuracy that could be expected in a preliminary design phase.

In fact, more sophisticated approaches would require a much more significant programming and computational effort; furthermore, a more detailed knowledge of the design would be necessary and a proper integration of non-linear components modeling would have to be implemented in a more complex framework.

Some considerations that prove the validity of the beam elements approach are:

- Deformations in the frame of the bogie are significantly smaller than those on the suspension systems. While asserting that the frame could be modeled as a rigid body may be too extreme of a claim, its displacements are surely much smaller than those on the suspension systems.

A more complex model for the bogie frame would likely not produce significant differences in suspensions displacement, at the cost of increased complexity.

- Tests with more complex models have been carried out. As an example, an algorithm has been developed to transform the wireframe structure into a solid 3D mesh, which is then processed by 3D elements FEA.

This approach has been proven to be far more computationally intensive and to require far longer elaboration times. Furthermore, complex mathematical tools, such as 3D triangulation, mesh restructuring and normals analysis are required.

Despite these vastly increased resources requirements, obtained results do not show any particular improvement. In fact, deviations from results obtained by FEA are negligible when compared to the uncertainties and approximations that are implicit in the model.

- The approach has been validated by comparison with results obtained by accredited software, such as *Simpack*. Results have been shown to not differ significantly to what is obtained through more complex commercially available models.

Even if some results cannot be directly compared, their overall coherence with verified results and expected values proves them to be satisfactory reliable and valid.

¹⁷⁸ As an example, a member undergoing significant displacements and rotations at its extremes is more likely to be undergoing higher stresses.

This is not necessarily true: a relevant counterexample is represented a component that is rigidly translated while maintaining its shape. Its overall displacement could be high, yet internal stresses may be negligible as relative displacements between different regions of the element are negligible too. It would be therefore more precise to say that stresses can be related to relative displacement and not absolute ones. That said, it is generally possible for an experienced user to relatively easily determine, at least approximately, what the stress distribution may look like by observing structural displacements.

In the preliminary phase of bogie design, the focus is often on the suspension systems displacements; in fact, the latter are necessary to avoid interferences and other geometrical issues.

While assessing possible excessive loads on the bogie frame, this is typically not considered too much of an issue in such a preliminary phase, as stresses can be evaluated only as a more complete model is available.

Bogie frame sizing is also quite well characterized and typical dimensions that are chosen for the structure are already guaranteeing some basic level of structural resistance to expected loads.

While exact results are obviously unknown, some expectation about plausible values and ranges for them exists. While results respecting such expectations does not imply their correctness, it allows to detect potential issues and provides some degree of confirmation.

Furthermore, some mathematical and physical relations are implicit in the modeling (e.g. lateral symmetry: the same results should be obtained for the two lateral halves of the bogie). If some of the results are verified by accredited software, the coherence of other calculated values with these can be evaluated and allows for further confirmation of the validity of the model.

The Finite Elements Method

Partial Differential Equations (PDE) are commonly adopted in physics and engineering to model real-life systems. In some cases they exactly describe the phenomena to be studied¹⁷⁹, while in other contexts they emerge from approximate or simplified modeling of a more complex system¹⁸⁰.

The Finite Element Method (FEM) is a widespread technique for the solution of partial PDEs in many fields of physical modeling. Typical applications are:

- Heat transfer
- Fluid flow¹⁸¹
- Electromagnetism¹⁸²
- Structural analysis, the object of the implementation in this project

FEM allows to obtain a numerical solution of 2D and 3D PDE problems. In most cases, such solution does not coincide with the exact one; if proper criteria are respected, though, the approximate solution can be made arbitrarily close to its exact correspondent at the cost of increased computational effort¹⁸³. Implementations in a higher number of dimension exist but usually are very niche applications, typically in research contexts

While the specific implementation depends on the field of application, some common features define the Finite Elements Method:

1. The domain to be studied is discretized in a set of elements of finite size¹⁸⁴ that could assume different shapes and characteristics depending on the dimensionality¹⁸⁵ and the requirements imposed by the problem¹⁸⁶. The collection of these elements is known as *mesh*; its properties can significantly influence the results and its proper realization and generation is extremely relevant for accuracy and stability.
2. In the case of steady state models¹⁸⁷ The mathematical problem is transformed from a system of PDEs to a system of algebraic equations.

In the case of time-dependent model, the procedure is more complex, as PDEs are converted to Ordinary Differential Equations (ODE) first; ODEs then are further solved by converting them to a system of algebraic equation. A large number of methods for such conversions has been devised.

¹⁷⁹ e.g. Maxwell equations.

¹⁸⁰ as an example, simplified or approximated forms of more complex PDEs are often adopted to model fluid flow. Simpler models can also be adopted to reduce computational effort, as in some cases calculation times can become impractically long.

¹⁸¹ The analysis of heat transfer and fluid flow are often combined (e.g. fluids running in heat exchangers).

¹⁸² e.g. solution of the Maxwell equations, which are PDEs.

¹⁸³ The topic of numerical approximation is really broad and its discussion would far exceed the scope of this brief introduction to FEM.

Despite differences between each implementation and its objectives and physics, a general guiding principle is that the higher the level of required accuracy, the more significant the computational effort.

Excessive accuracy is typically not required, as beyond a certain degree of precision other approximations and uncertainties become more relevant (e.g. implicit assumptions regarding the model and simplifications of its features). In general, finding the right balance between accuracy and computational effort is a matter of user experience and knowledge and a vast literature exists about this topic.

¹⁸⁴ from which the name *Finite Elements Method*.

¹⁸⁵ 2D or 3D

¹⁸⁶ a wide variety of elements types exists: triangular, square, tetrahedral, plate, brick, beam and so on. Their choice is relevant for the proper solution of the problem and the validity of the obtained results.

¹⁸⁷ i.e. with no time evolution and dependence; time is therefore not considered as a variable.

In both cases, the conversion to algebraic equations allows to obtain a numerical solution, at the cost of introducing some degree of approximation. If the underlying mathematical model is linear, these equations are linear too.

3. As PDEs require a set of boundary conditions, these must be imposed too to the model. The nature of such boundary conditions significantly depend on the physical nature of the problem; in general, though, they constrain some parameters of a subset of the mesh elements and are necessary to obtain a unique solution.

Finite Elements Analysis (FEA) is widely adopted in many branches of engineering and represents the most common technique to realistically simulate many physical phenomena. Often it is employed both in the design and validation/verification process. In the latter case, its role is extremely important, as it can be adopted to verify compliance to standards ¹⁸⁸

¹⁸⁸ notably, simulations are quite strictly regulated and in the context of safety standards compliance evaluation.

In this context, FEA allows to simulate damage or extreme events involving the object of the compliance evaluation; the performance of the latter in such conditions can then be evaluated.

To provide an example, in the case of passenger vehicles, such as cars or trains, safety assessments are strictly regulated not only in real-life testing but in computer simulations too, to ensure proper survivability of the vehicle.

Advantages and disadvantages of FEA

The main advantages of Finite Element Analysis are:

- Representation of complex geometries: articulated shapes can be easily represented by their discretization
- Multiphysics simulation: different physical models can be implemented at the same time. Different material, with their respective properties) can be implemented too.

A typical example is represented by the simulation of heat transfer in fluids: both flow and heat transfer simulation are required. Furthermore, this type of analysis typically requires both solid and fluids to be present and modeled at the same time. This can easily be achieved by FEM analysis.

- Multiscale simulation: the mesh refinement can be varied in different areas of the model. A larger mesh can be adopted to evaluate the global behavior of areas exhibiting more homogeneous/large scale features, while it can be refined to capture local/small scale effects.
- Generally speaking, FEM allows to obtain a numerical solution for PDEs that would otherwise not be analytically solvable. In fact, in most cases closed form solutions do not exist or would be practically impossible to evaluate. In some cases, even if a closed form analytical solution could be calculated, it is still more practical and faster to calculate FEM numerical solution.

Its main drawbacks are:

- Complex models may require significant computational effort; this is even more significant if time dependent solutions are required.
- The obtained solution is a numerical approximation of the exact one. This is usually not too much of an issue, as the accuracy of the solution can be arbitrarily increased.
- Accuracy - Computational effort trade off: to obtain better results, more refined meshes and models are necessary, increasing the computational requirements.

FEM in structural static analysis

In the context of static analysis, the time evolution of the systems is not considered, as loads and the system's response to these is assumed to not vary in time.

FEA can be employed to evaluate and predict displacements¹⁸⁹, strains and stresses in solid structures subject to static loads. As the problem is time-invariant, it can be approximately reduced to a system of linear equations and then solved to obtain the displacements. The latter then allow to compute strains and stresses.

The typical workflow of static structural FEM analysis is as follows:

1. The structure is discretized by subdivision in elements. Many element types have been developed and can be implemented depending on the nature and complexity of the structure to be modeled. They can be uni-dimensional¹⁹⁰, bi-dimensional¹⁹¹ or tridimensional¹⁹².

More complex elements may allow for more specific and detailed applications, at the cost of increased computational effort.

In the project: as mentioned, beam elements are adopted to model the bogie and its components.

The adoption of more complex elements has been avoided as the computational and programming effort would have been significantly increase with no relevant advantages in terms of performance. In fact, the adoption of some element types, such as plates, not only would not have improved the simulation but impaired it, as they are not well suited to represent the shape of commonly adopted bogie elements.

Beam elements have demonstrated to provide reliable and reasonably correct results while avoiding unnecessary complication and have therefore been chosen for the project.

2. Properties are associated to each element. These describe the geometry of the element and its mechanical properties; they are needed to properly evaluate its behavior.

In the project: as discussed, edge objects allow to store relevant attributes that are needed for static analysis. As beam elements are adopted, the features that have to be defined are:

- Material properties: elastic modulus, density, Poisson ratio, etc.
- Geometrical properties: length, area, section moments of inertia, etc.

¹⁸⁹ displacements are to be intended in their generalized sense: deviation from a reference position. This, as an example, includes rotations too.

¹⁹⁰ e.g. beams

¹⁹¹ e.g. plates, triangular elements

¹⁹² ex. brick and tetrahedral elements

3. Each element type is well defined from a mathematical and physical perspective. Typically, the shape of the elements is defined by the position of its nodes; degrees of freedom and constraints of the latter depend on the element type and its mathematical description.

External forces and torques can be related to nodal generalized displacements. It is in fact this mathematical relation that allows to evaluate displacements by solving a linear system of equations.

A nodal *generalized displacement vector* u is defined for each element. Each entry in u represents the value of the generalized displacement along the corresponding degree of freedom. Displacements are *generalized* as they represent a generic deviation from a reference position on its corresponding degree of freedom. They can therefore represent translations, rotations or more complex expressions and properties.

As displacements are generalized, the same is true for forces acting on the elements: their generalized form also includes momenta and other types of external inputs. The *generalized forces vector* F usually, if not always, has the same dimensions of the displacement vector u and it is representative of the external inputs acting on the element.

The *stiffness matrix* K mathematically correlates the displacements and forces vectors by the relation

$$Ku = F$$

The stiffness matrix depends on the properties of the elements and on the element type. It is often derived by assuming an approximate but realistic displacement field over the element using shape functions; then the principle of virtual works or the direct stiffness method can be applied to evaluate its entries.

K, u and F are defined to be *local*, as they refer to the local frame of reference of the element and ignore its position and orientation in the structure.

4. Supports: As the solution to PDEs requires boundary conditions, the structure must be properly constrained. The most common constraints for node degrees of freedom are:

- Fixed: the displacement with respect to the degree of freedom is set to be null¹⁹³.
- Free: the generalized displacement is free to assume any value
- Spring constrained: the generalized displacement is not fixed, but a restoring force¹⁹⁴/torque¹⁹⁵ proportional and opposite to it is applied.

The same node can have different constraints on different degrees of freedom. Many combinations can therefore arise, especially as the dimensionality of the problem is increased.

In two dimensions, the number of possible combinations of fixed/free d.o.f is quite small. In fact, only three d.o.f are present, two translational d.o.f and one rotational d.o.f.

¹⁹³ In the case of translational d.o.f this is equivalent to fixing the position of the node on the d.o.f. axis; in the case of rotational d.o.f, this is equivalent to not allowing the node to rotate along the d.o.f axis.

¹⁹⁴ in the case of translational d.o.f

¹⁹⁵ in the case of rotational d.o.f

Other types of two dimensional supports exist, such as double-pendulums and double double-pendulums. They are however quite rarely found in most applications.

Ignoring spring-based constraints, four main combinations are therefore usually found:

- Free node: the node is not constrained on any d.o.f.
- Roller support: one translational d.o.f is constrained, while the other and the rotational d.o.f are unconstrained
- Pinned support: the two translational d.o.f are constrained, while rotation is free
- Fixed support: all d.o.f are fixed

In three dimensions, the number of available combinations increases drastically. In fact, six d.o.f are present: assuming three options to be available for each (fixed, free, spring constrained), the number of possible combinations is $3^6 = 729$.

Due to the large number of combinations, only a few are classified; in fact, in many cases this would be unnecessary or unuseful.

It is worth mentioning that these idealized constraints often are just approximations of physical reality. Typically, the interaction between a structure and its surroundings involves the combination of many complex phenomena, such as friction, non-linear contact behavior, and so on. While more complex models for supports exist, they usually are not worth the additional computational and conceptual effort. In general, properly choosing how to properly model supports and constraint is not easy and requires some experience in the field; at the same time, proper selection of constraints is very relevant to obtain valid results.

5. Global stiffness matrix assembly: as the reference frame of the element does not necessarily coincide with that of the structure as a whole, to properly solve the system it is necessary to convert each local stiffness matrix, force vector and displacement vectors in their global form-

The local stiffness matrices are combined, by applying the proper rotation matrices, to form the global stiffness matrix K_g . The same is true for generalized displacements and forces: global vectors u_g and F_g are obtained. These describe respectively all nodal displacements and loads in the reference frame of the structure.

All displacements can then be calculated by solving the system

$$K_g u_g = F_g$$

Only part of the displacement may be calculated if needed. In some cases (e.g. in large systems, containing many elements) this solution can significantly reduce computational costs.

The mathematical derivation of the global matrix and the necessary rotation matrices are quite lengthy to be described and will be not be presented for brevity.

6. The system is then solved: displacements are obtained. Usually, for each element type, a relation between nodal displacements and strain/stresses can be determined. It is therefore possible to evaluate stresses and deformation in the structure¹⁹⁶.

In general, resulting displacement can be further elaborated to evaluate relevant quantities. This is in fact the case in the project: as it will be discussed, many insights on the behavior under load of the system can be derived from its displacements.

¹⁹⁶ As discussed, stresses evaluated by this technique do not account for local effect and can be misleading if an accurate model of the structure is available.

Beam elements

Beam elements represent the simplest 3D element in structural FEA. They allow to model components that are significantly more extended along one dimension than the others.

In spite of their relative simplicity, they support the modeling of axial, torsional and bending stresses. When combined in a wire-frame/truss layout, they can approximate quite well many complex structures and shapes, even if these are not implicitly beam-like in nature¹⁹⁷.

Various formulation for beam elements have been devised. The most common are *Euler-Bernoulli* and *Timoshenko* beam elements.

The **Euler-Bernoulli** beam model assumes that, under deformation, plane sections maintain their shape and remain perpendicular to the beam's axis. This approach does not account for shear stresses and it is typically applicable in the case of slender beams.

The **Timoshenko** beam model does account for shear stresses and the effects of rotational inertia. Its application is required for thick beams or in the case of high shear stresses.

The Euler-Bernoulli model represents the limit case of the Timoshenko beam as its slenderness tends to infinity.

The Timoshenko model is more accurate, at the cost of increased mathematical and computational complexity. Its adoption is therefore typically avoided if not strictly necessary. This is in fact the case in this project: Euler-Bernoulli elements have been adopted.

Beam elements are typically defined by two nodes, one at each extreme. As three-dimensional beams are considered, each node has six degrees of freedom: one translation and rotation for each axis.

As the displacement vector u represents all displacements in the element, it has to account for all degrees of freedom of the two nodes. It is most commonly expressed as a 12×1 column vector.

The same reasoning can be applied to external loads: forces and torques applied to each node can be decomposed in their components along each axis. Non-nodal forces, such as distributed loads, are converted to nodal forces that well approximate them.

As each node accounts for six degrees of freedom, the generalized forces vector F is typically expressed as a 12×1 column vector too.

A full derivation of equations describing beam elements will be avoided, as it would be unnecessarily lengthy without significantly contributing to the understanding of the analysis process.

¹⁹⁷ A complex three-dimensional shape can be triangulated and the resulting simplices' edges can be converted to beam elements.

While in such cases tetrahedral elements may be preferable, this can still be a valid approach and it is sometimes adopted.

Testing has been conducted to evaluate differences in results between the two approaches. The implementation of the Timoshenko model has proven to be significantly more complex without a significant increase in accuracy. Time requirements are also increased.

In fact, the differences in results are almost negligible with respect to the degree of uncertainty and approximation that is implicit in the bogie model.

The adoption of the Euler-Bernoulli beam has therefore been preferred for its simplicity and efficiency.

Stiffness matrices and displacements calculation

While a complete form for the characteristic expression $Ku = F$ exists, the axial, bending and torsional contribution are typically decomposed.

Assuming the x -axis to be the longitudinal axis of the beam and the y and z axes to be perpendicular to it, the following expressions can be derived:

Axial component: the axial behavior of beam elements is described by the reduced form

$$K_{ax}u_{ax} = F_{ax}$$

with:

$$u_{ax} = \begin{Bmatrix} u_{x,1} \\ u_{x,2} \end{Bmatrix} \quad F_{ax} = \begin{Bmatrix} F_{x,1} \\ F_{x,2} \end{Bmatrix} \quad K_{ax} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

The displacement vector represents the axial displacement of the nodes, while the force vector sums up regular axial nodal forces and equivalent forces representing non-nodal contributions.

Torsional component: the torsional behavior of the beam is described by the rotations along the x -axis of the nodes. The formulation is very similar to the axial case:

$$K_{tors}u_{tors} = F_{tors}$$

with:

$$u_{ax} = \begin{Bmatrix} \phi_{x,1} \\ \phi_{x,2} \end{Bmatrix} \quad F_{ax} = \begin{Bmatrix} M_{x,1} \\ M_{x,2} \end{Bmatrix} \quad K_{ax} = \frac{GJ}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Bending contribution: bending is further decomposed in its xy and xz plane components. The mathematical description is the same, with the exception of the different variables to be considered and possibly, depending on the adopted sign convention, a change in sign of some elements of the stiffness matrix.

In the case of bending on the xy planes:

$$K_{bend}u_{xy} = F_{xy}$$

with:

$$u_{xy} = \begin{Bmatrix} u_{y,1} & \phi_{z,1} & u_{y,2} & \phi_{z,2} \end{Bmatrix}^T \quad F_{xy} = \begin{Bmatrix} F_{y,1} & M_{z,1} & F_{y,2} & M_{z,2} \end{Bmatrix}^T$$

$$K_{bend} = \frac{EI}{L^3} \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix}$$

Stiffness matrices and they respective forces and displacements vectors are combined to obtained the complete form. The complete 12×12 stiffness matrix and the corresponding vectors are then appropriately transformed to be integrated in the global stiffness matrix and to allow for a solution in the reference plane of the structure.

By applying appropriate matrix transformations, elemental strains and stresses can be calculated from nodal displacements.

Properties of the beam that are considered are:

A: cross-sectional area

L: length

E: elastic modulus

G: shear modulus

I: second moment of area

J: polar moment of inertia

All these parameters are included in the attribute set of edge objects and are required for proper modeling.

Their values are set by default to reasonable estimated based on industry standards yet they can be modified or set by the user if desired.

The derivation of the hereby presented expression is quite lengthy and far beyond the scope of this introduction to the static analysis techniques that are adopted in this project.

The derivation of strain and stresses is also not straightforward and, as it is not performed in the project, it will not be presented to avoid unnecessary digressions.

The decomposition into axial, torsional and bending contributions is possible due to the assumption, characteristic of Euler-Bernoulli beams, of their independence.

In real beam this only approximately true: as an example, axial loads can modify quite significantly the bending stiffness; in general, interactions exist between loads and stresses of different nature.

These effects are only partially accounted for in this project, as they require significantly complexity to be added. Testing has proven their effect to be negligible anyway: as mentioned the behavior of the bogie frame is close to that of a rigid body; such effects are therefore of little relevance.

Modeling of complex structures requires beams to be joined at one of their extremes, known as *common node*. Different types of connection can be realized:

- **Rigid connection:** the two beams share the same position and rotation of the common node. The rotation is constrained between the two and in the global reference frame they have the same nodal rotations.
- **Hinged connection:** the two beams share the same position of the common node, but rotation is not constrained on one or more axes. In the global reference, the two beams share the location of the common node, but the nodal rotations may be different. Ideal hinged connections transfer forces but not torques between beams.
- **Spring connection:** the extremes of the two beams do not share either the position and the rotation of the common node. A restoring force, proportional and opposite to their relative displacement (both translational and rotational), is applied. This is a less common solution, sometimes used to model elastic joints or similar components.

Different beam connections can often be introduced on different axes. This is true in the project too and will be discussed when analyzing the primary suspension system.

As an example, the relative rotation of the beams can be constrained on one of the axes and free on others.

In other cases, torsional or linear spring connection can be adopted to simulate elastic elements between beams.

In general, many solutions can be adopted, with the goal of obtaining as close of a model as possible to reality. As previously discussed, though, constraints often represent approximations of physical connection and their proper selection is sometimes not straightforward.

FEA implementation in the project

In the project, Finite Element Analysis has been chosen to perform the static analysis of proposed designs. This approach has been chosen as it represents the most widespread solution to evaluate displacements in a structure under static loads.

While related to displacements, quantities that are to be evaluated in the context of static analysis are typically more complex: they require further examination and elaboration of calculated displacements to obtain more insightful results.

As an example, the behavior of characteristics parameters as a function of loading conditions can be evaluated and graphically presented to the user.

The implementation of the above mentioned analysis tool has required the introduction of additional capabilities with respect to conventional FEA. In fact, the primary suspension type and its mathematical model are non-linear in nature. A proper technique to model non-linear mechanical behavior by the use of an inherently linear approach, such as FEM, had therefore to be devised.

Considering the requirement for a static analysis tool, the AI itself has been design not only for easy compatibility with the graphical interface but to allow for smooth conversion to the FEM environment.

Objectives

In spite of what may be expected, the main focus of the static analysis process is not on bogie frame displacements. In fact, as previously discussed, the frame is significantly stiffer than suspension systems¹⁹⁸.

The behavior of the frame under load is not ignored though: displacements of all bogie elements are presented to the user if required. That said, practical experimentation has shown that deformations¹⁹⁹ in the frame are not particularly insightful and relevant²⁰⁰ when compared to suspension behavior.

A deeper analysis of the bogie design process does in fact show that, in a preliminary phase, the determination of suspension characteristics is more relevant than frame design. Suspension systems do in fact determine many of the features of the bogie, including the shape and layout of the bogie frame. It is therefore sensible to consider its evaluation as a starting point for the design process.

¹⁹⁸ to provide a numerical reference, displacement in the frame are in the order of fractions of a *mm*; spring elongations and other deformations/displacements in the suspension system are easily in the order of tens of *cm*.

¹⁹⁹ While frame displacement may not be relevant, stresses are: in fact, stress verification of the frame is extremely important in assessing the validity of the design. As mentioned though, no stress analysis is performed; the focus of the analysis is therefore shifted to the suspension system, which typically is more displacement dependent.

²⁰⁰ In some limit cases deformation can perceptibly vary along the frame; such results, though, are typically obtained when considering very unrealistic configurations, such as cantilever beams spanning the whole length of the bogie.

FEM model generation

Designs provided by the AI generation and user interaction process need to be converted to a form that allows for FEM analysis. Considering the need for static analysis, objects that are adopted to implement the GUIDE procedure are conceived to be easily convertible to a beam element based structure. Elements and subsystems present in the bogie design are converted and interpreted as follows:

- **Frame:** the conversion from geometrical edge objects to beam elements is performed assuming beam-to-beam connections to be fixed²⁰¹.
 1. Beams nodes are defined to have the same position as edge objects extremes
 2. Material and geometrical properties are imported from edge objects' attributes set and adopted for the corresponding beam element
 3. If two edge objects are geometrically connected by a node, the corresponding beams are considered to be linked by a fixed connection
- **Wheelsets:** as only the bogie frame and suspension systems are considered in the static analysis, whole wheelsets are not accounted for in the FEM model.
 Their presence in the bogie is not ignored, though: the extremes of the wheelset are considered as the points to which the primary suspension are attached. Such points are therefore considered to be the interface between the bogie structure and the ground²⁰².
- **Primary suspensions:** the modeling of the primary suspension system that has been chosen for this project is a complex topic and will be later discussed in detail. Its behavior is considerably non-linear and requires proper analysis methods to be implemented. As they only provide a dynamic contribution, dampers are ignored in the context of static analysis.
- **Secondary suspensions:** as in the case of wheelsets, secondary suspensions are not directly model; their role in the system is represented by assuming their connection point²⁰³ with the bogie to transmit vertical loads to the frame.
 As a consequence, during analysis all vertical loads are considered to be applied to the frame-secondary suspensions connection point²⁰⁴.
- **Mechanical link:** the mechanical link between the bogie frame and the vehicle main body is assumed to transmit all horizontal traction loads. The latter are therefore applied during analysis at the node representing the mechanical link connection to the frame.
- **Reference frame:** to improve compatibility, the structure global reference frame is the same that is adopted for AI generation and user editing.

²⁰¹ this approach might imply a loss of generality: in theory, it could be possible to adopt flexible of free connections along some axes between beams. The implementation of such possibilities has been determined to be both very time consuming and often unuseful from a user perspective; fixed connections have therefore been assumed by default.

²⁰² Assuming wheelset extremes to represent the connection to the ground may seem a gross approximation. It is worth mentioning, though, that wheelset deflections can be assumed to be negligible; interaction forces between the bogie and the ground can be therefore considered to be transmitted directly to the wheelset-primary suspension interface without significant loss in accuracy, while greatly simplifying the simulation process.

²⁰³ this is coherent with how secondary suspensions are represented both in the AI generation and user editing phase: the suspension system itself is not considered, as it may significantly vary in nature; its connection point with the frame is generated and edited instead.

²⁰⁴ While secondary suspension do deform under load, the position of their frame connection point only depends on the load that is transferred by them. Static loads from the main body are therefore assumed to be directly transferred by secondary suspensions to the bogie frame.

As it will later discussed, while secondary suspensions properties can be ignored in the context of static analysis, this is not the case when evaluating the dynamic behavior of the bogie.

Supports and constraints

The bogie structure as a whole should be properly constrained with respect to all its degrees of freedom²⁰⁵. As previously mentioned, the connection points between the axles and the primary suspensions have been considered as supports: four support points are therefore defined; to guarantee symmetry, the same constraints have been applied to all of them.

Constraints that are adopted at support points are:

- **x-axis translation:** in theory, wheels and, as a consequence, axles should be free to move along the x-axis; the latter does in fact represent the expected direction of motion along the tracks. Axles extremities, which have been defined to represent supports, should therefore not be constrained on the x-axis.

Unfortunately, if no constraint were to be applied in this direction, the system would become unstable from a numerical perspective; either invalid solutions or no results at all would be produced.

To avoid this issue, very soft spring constraints have been added on the x-axis on the support points. This solution allows the structure to be numerically constrained while maintaining sufficient freedom of movement along the x-axis.

- **x-axis rotation:** rotation about the x-axis is theoretically mildly constrained, as the extremities of the axles can present small rotations due to bending.

In practice, it has been found out that differences in results between fixed and free x-axis rotations are negligible; fixing the degree of freedom, though, improves numerical stability. This latter solution has therefore been implemented.

- **y-axis translation:** translation along the y-axis is fixed. While in practice small displacements could be present, testing has proven them to be negligible with respect to expected accuracies; on the other hand, such displacements typically involve non-linear phenomena and are not easily modeled.

- **y-axis rotation:** rotation along the y-axis is fixed. This choice may seem counterintuitive: the axlebox is in fact free to rotate about the end of the axle. The reasons for this choice will be presented as the modeling of primary suspensions is described as proper contextualization is needed.

- **z-axis translation:** from a theoretical point of view, the support is fixed with respect to z-axis translations.

In the practical implementation, this is realized by introducing an extremely stiff spring. This solution allows to determine vertical loads on the spring by evaluating its displacement at the free end. The stiffness of the artificially introduced springs is so high that its effect on the numerical solution is completely negligible.

- **z-axis rotation:** the rotation of the supports is assumed to be fixed with respect to the z-axis. This can be realistically assumed true as the wheel-track contact significantly constrains the z-rotation at the extremities of the axle. While phenomena such as bending of the axle and displacements in the bearings may present, they have been determined to be negligible.

²⁰⁵ three translations and three rotations, as it is three-dimensional.

Testing has proven that the addition of soft spring constraints has a negligible effect on obtained results, with the added benefit of vastly increasing numerical stability.

Spring that are adopted for this approach are many order of magnitude less stiff than elastic and structural elements present in the bogie.

The difference with respect to fully fixed supports on the x-axis is negligible too. This is relevant as the simulation of horizontal loads requires more stringent constraints on the x-axis to avoid numerical instability.

When considering translations along the x- and y- axes and rotations about the z-axis at support points, effects due to axle bending, clearances in bearings and wheel-track contact may be present. These effects are in general very difficult to model as supports.

The most commonly adopted approach is to verify that imposing a fixed constraints does not significantly alter results. This is the case in this project too: analysis has shown these nonlinearities to produce negligible effects when compared to implicit approximations present in the model and numerical accuracy of the solution.

As these examples show and as it has been previously mentioned, the choice of constraints is often not easy, as real physical systems typically do not exactly follow mathematical supports models.

A compromise is usually to be found between ease of implementation from a mathematical/computational perspective and realistic modeling of the underlying physical system. User experience is often needed to properly balance such requirements and to determine a degree of approximation that is compatible with the objectives of the analysis.

The selected constraints at support points allow to constrain the structure as a whole. In fact, to allow for a valid numerical solution, the model itself should be properly constrained.

In other words, imagining the structure to be a rigid body, its movement along each of its degrees of freedom should be either fixed or defined by a mathematical relation²⁰⁶; a stable and converging numerical analysis is possible only if this condition is fulfilled²⁰⁷.

From a global perspective, the structure as a whole is limited in its movement along reference frame axes as a consequence of local constraints imposed at supports.

- **x-axis translation:** as previously mentioned, soft springs limit the movement of the bogie as a whole on the x-axis. While bogie movement along the x-axis is therefore not fixed, it requires loads that are significant enough to cause a displacement in the virtual springs, thus avoiding numerical singularities.
- **x-axis rotation:** fixed, as supports are, practically speaking, fixed on their vertical axis
- **y-axis translation:** fixed, as supports are fixed on the y-axis
- **y-axis rotation:** fixed, same as x-axis rotation
- **z-axis translation:** fixed, as supports are vertically constrained
- **z-axis rotation:** fixed, as supports are fixed on the y-axis and constrained on the x-axis

While these constraints may seem restrictive, they are coherent with practical expectations: as the bogie is assumed to stand on tracks, its movement can be assumed to be impeded along all directions, except that of the tracks.

External loads

The objective of static analysis is to assess the response of the bogie to different time invariant loading conditions.

Vertical loads are assumed to be transmitted to the bogie frame secondary suspensions. Their application point is therefore determined to be the attachment point of the latter to the bogie frame.

Horizontal loads are assumed to be transferred to the structure by the mechanical link. This is often the case, as the latter is usually responsible for the transmission of traction loads from the bogie to the vehicle body and vice-versa.

As it will later be discussed, different combinations of vertical and horizontal loads can be applied to simulate different operating conditions.

²⁰⁶ if no constraint is present and movement along an axis is free, results are typically undefined and numerical singularities arise during the solution process.

²⁰⁷ Conventional FEA relies on the solution of a linear system. The non-linear solution approach that is adopted in this project does instead require an iterative solution; stability and convergence are therefore much more relevant and the correct implementation of boundary conditions is of great importance.

With respect to the tracks, lateral movements (y-axis translation) and rotations with respect to the vertical axis (z-axis rotation) can be considered to be fixed. While some sliding may occur at the wheel-track interface, external loads (both forces and torques) inducing movement along these directions are far more likely to deform the bogie structure than to cause significant global displacements.

The bogie as a whole is, in theory at least, not constrained on the vertical direction. In fact, wheel-track contact is only granted by the vehicle self-weight. That said, the latter is so significant that, from a practical perspective, wheels can be considered fixed to the track in their vertical axis movement. As a consequence, x- and y- axis rotations and z-axis translations are prevented from occurring.

Loads that are encountered during operation are typically not constant in time and be applied along a wide variety of directions. Nonetheless, evaluating the response of the bogie under standardized loading conditions, assumed to be reasonably representative of real operational loads, can provide insight into the bogie behavior and allow for comparison between different designs.

Primary suspension modeling

Many types of primary suspension exists and, as this section will demonstrate, their modeling can be quite complex and a detailed knowledge of its physical behavior may be needed to obtain accurate results.

Considering the scope and time-frame of this project, the implementation of a large library of primary suspensions has been deemed both unreasonable and unnecessary.

The approach that has been adopted is to select one suspensions type to be modeled, assumed to be representative of the largest possible variety of components and physics. This model has then been not only implemented and simulated, but thoroughly analyzed and verified.

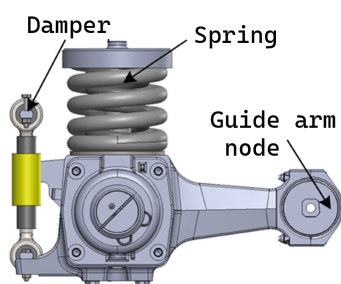
The objective is therefore not to exhaustively implement and simulate every type of primary suspension, but to provide a relevant example that can act as a guideline for similar implementations.

The axlebox-arm suspension system

The axlebox-arm suspension system is a commonly adopted solution for primary suspensions. It features a spring acting in the vertical direction and a guiding arm²⁰⁸ providing horizontal support²⁰⁹.

An annular elastic element²¹⁰ is typically inserted at the guide arm node. By design, its effect on the vertical direction is limited, as the spring stiffness and load capacity is significantly higher; its main role is improving stability during curve insertion.

Dampers are typically present and parallel to the spring. Due to their dynamic-only nature, they will not be considered in static analysis.



The axlebox-arm suspension system that has been chosen for primary suspensions is considered to be one of the most complex in terms of modeling among those that see common practical applications.

The underlying assumption is that, if modeling strategies and solutions can be developed for it, the implementation of simpler suspension system would be at worst of equal complexity, if not easier.

²⁰⁸ comprising the axlebox

²⁰⁹ as it will be discussed, in operating conditions this element typically is not completely horizontal

²¹⁰ usually made out of rubber

Figure 27: Main components of the axlebox-arm suspension system (left) and example of its realization in a railway bogie.

Two axlebox-arm assemblies are present for each axle, one per each side. A total of four is therefore necessary in the case of a two axles bogie, as those considered in this project.

Main features of axlebox-arm suspensions

To properly model the complex behavior of axlebox-arm suspensions, some of its most relevant must be considered:

- **Preload in operating conditions:** as the suspension system is assembled, it is typically preloaded²¹¹.

Axlebox-arm suspensions are defined to be at *assembly condition* if the guide arm is horizontal. The external vertical load under which this condition is achieved is the *assembly load*.

A typical approach adopted in bogie design is to assume normal operating conditions²¹² to be coincident with assembly condition. Except for specific circumstances, spring and suspension displacements are therefore commonly evaluated with respect to assembly condition.

External vertical loads can be either expressed in absolute terms or with respect to the assembly loads; as it will later be discussed, the choice between the two depends on the objective of the analysis in which they are considered.

- **Complex geometrical interaction between components:** the modeling of springs, especially in the context of dynamic analysis, often relies on the assumption of small displacements around an equilibrium point.

Unfortunately, this simplification is not applicable in the context of axlebox-arm systems: springs deformation are typically close or in the same order of magnitude as the overall dimension of the suspension. The shape of the suspension as a whole can therefore be significantly altered depending on external loads.

Considerable non-linear effects are therefore introduced: as the loads varies, the layout of the suspension is altered and, as a consequence, its response to the load too²¹³.

- **Non-standard spring operating conditions:** while coil springs are often modeled as acting only on their main axis, this approach is insufficient to properly model their behavior in the context of axlebox-arm suspensions.

Mechanical systems exhibiting significant shape modification under load do in fact induce spring reactions that are not aligned with its main axis²¹⁴.

The spring does not only exhibit compression on its main axis, but radial displacements and rotations at its ends. While these deformation are on a smaller order of magnitude with respect to the main vertical deformation, they induce a response by the spring.

Mathematical models have been developed to evaluate the radial and torsional stiffness of coil springs; they are adopted in this project to estimate and simulate the spring response and its effect on the suspension system.

²¹¹ This preload can be either externally imposed or provided by the self-weight of the bogie and of the vehicle. In practice, both are usually present.

²¹² While operating conditions can significantly vary (number of passenger or amount of cargo to be transported, external environment, etc.), *normal operating conditions* are considered the most likely or most representative and are often adopted as a reference during the design process.

To provide a practical example:

Assuming an assembly load of 70kN, if the total external load amounts to 90kN the relative load is 20kN.

If the external load is decreased to 60 kN, the relative load is -10kN. The spring is still under compression, though, as the total load is positive. In fact, springs in axlebox-arm suspensions do always work under compression.

The same reasoning can be applied to spring elongation: while, if the load is reduced, the spring is longer than at assembly condition, it still is compressed and much shorter than its free length.

²¹³ Non-linearities do in fact represent one of the reasons for the choice of axlebox-arm suspensions in this project: their modeling implies the simulation of non-linear effects, allowing to showcase how they could be implemented in the overall GUIDE approach, showcasing the flexibility of the latter.

While not representative of all structures that may be object of a design process, demonstrating that non-linear effects can be modeled in this work-frame allows to significantly extend the set of designs that can be handled by the GUIDE approach.

²¹⁴ This effects can typically be ignored in the case of small displacements; in the context of axlebox-arm suspensions, though, they do have a significant effect on their system stiffness and reaction to external loads.

Mathematical modeling of coil springs

Coil springs manifest different reactions to external loads, depending on their nature and direction. Stiffnesses can be therefore defined to describe the spring behavior under different loading conditions.

Axial stiffness

The axial stiffness represent the reaction of the spring to external loads acting on its main axis. If non-linearities are ignored, this is the stiffness that is usually considered in coil spring modeling.

It can be evaluated from geometrical and material parameters of the spring as follows:

$$k_z = \frac{Gd^4}{8D^3N} \quad F_z = k_z \cdot s_z$$

Radial and flexural stiffness

The radial stiffness of the spring expresses its reaction to radial loads. In particular, the relation between radial displacement and force is expressed by:

$$F_q = k_q \cdot s_q$$

The spring also produces a torque proportional to the difference in rotation angles $\Delta\theta$ of the free ends of the spring:

$$T_f = k_f \cdot \Delta\theta$$

The torque T_f is considered to be applied at the free ends of the node; the direction of the torque is assumed to be restoring: it acts to realign the spring to its undeformed main axis.

k_f is the flexural stiffness of the spring.

The Krettek-Sobczak model

The evaluation of radial and flexural stiffness is not trivial, as it not only depends on geometrical and material properties, but on the applied load too: as the axial load increases, the spring becomes more stiff both with respect to bending and radial loads.

The Krettek-Sobczak model provides a procedure to evaluate both bending and radial stiffness; it is in fact adopted in this project too.

Two auxiliary functions are defined:

$$B(s_z) = \frac{1+\nu}{4+2\nu} \cdot D^2 \cdot H \cdot k_z \quad S(s_z) = 2(1+\nu) \cdot k_z \cdot H$$

$$k_h(s_z) = \sqrt{\frac{F_z}{B(s_z)} \cdot \left(1 + \frac{F_z}{S(s_z)}\right)}$$

Theoretical lateral and flexural stiffnesses are calculated as:

$$k_{qh} = \frac{F_z}{\frac{2}{k_h} \left(1 + \frac{F_z}{H}\right) \cdot \tan\left(\frac{H \cdot k_h}{2}\right) - H}$$

$$k_{fh} = k_{qh} \cdot \frac{B}{F_z} \cdot \left(1 + \frac{F_z}{S} - \frac{H \cdot k_h}{\tan(H \cdot k_h)}\right)$$

Finally, effective radial and flexural stiffness can be evaluated to be:

$$k_l = a_1 \cdot k_{qh} \quad a_1 = 1.9619 \frac{s_z}{H_0} + 0.674$$

$$k_f = a_3 \cdot k_{fh} \quad a_1 = 0.8945 \frac{s_z}{H_0} + 0.669$$

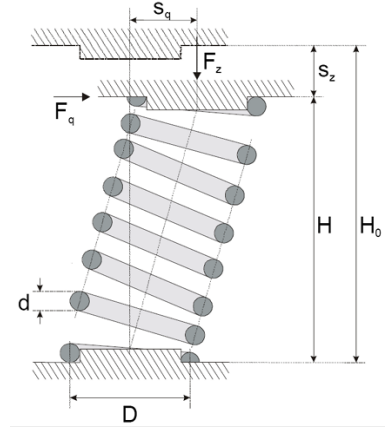


Figure 28: Spring subject to axial and non-axial loads and displacements.

The spring is compressed on its main axis due to vertical loads.

Radial loads also induce a corresponding displacement. The relation between the two is provided by the radial stiffness.

The spring may also be subject to bending. The flexural stiffness allows to describe its reaction to bending loads.

Relevant parameters and quantities are:

H : deformed axial length

H_0 : spring free length

D : mean spring diameter

d : coil diameter

N : number of active coils

F_z : axial load

s_z : axial displacement

F_q : radial load

s_q : radial displacement

G : shear module

ν : Poisson ratio

The increase in bending and radial stiffness under load can be intuitively explained:

Considering bending, as the spring is shortened it becomes harder to bend, as its length/thickness ratio is decreased.

As coils are more packed, their resistance to radial stresses is increased to as their lateral movement is more and more impeded.

Modeling of the suspension components

Each component of the suspension system has to be mathematically model. This conversion is realized as follows:

- The guide arm is modeled as a beam. Its dimensions and geometrical properties are fixed and are set to be coherent to commonly adopted layouts in real-life bogies; they are different from these of other structural elements.
Both arm extremities are considered to be hinged connections²¹⁵.
- Springs are modeled by a specifically introduced element type. Its basic behavior is that of a axial-only spring, yet lateral and bending stiffness are introduced too.
The realization of such a complex and non-linear element is non-trivial and requires the adoption of an unconventional FEM approach²¹⁶.
- The elastic joint, connecting the arm to the frame, is modeled as applying a restoring torque to the arm, proportional to the rotation of its extreme with respect to the body.
- The structural member of the suspension that is part of the bogie is modeled as a beam connecting the free end of the spring to the guide arm node. This element is by default considered to be part of the bogie frame too; it is therefore shared between the latter and the primary suspension system.

²¹⁵ On the frame side, the arm is free to rotate, except for the presence of the elastic elements. The latter is separately modeled, though, and the frame side can therefore considered to be hinged. As discussed, the wheelset extreme is counterintuitively modeled as a fixed support. This solution is chosen as its connection to the arm is assumed to be hinged and it therefore does not constrain its rotation. On the other hand, adopting a fixed support allows to maintain the rotation of the spring end connected to it to be fixed; this is close to real working conditions of the spring.

²¹⁶ Details of its implementation are the object of later chapters.

Modeling of nonlinearities

The Finite Elements Method is well suited for the analysis of linear systems; in fact, displacements are typically obtained by reducing the problem to a system of linear equations. As discussed, the axlebox-arm suspension is characteristically nonlinear; issues therefore arise when adopting a conventional FEM approach:

- **Springs unidirectionality:** even if their final deformed configuration is not parallel to their undeformed axis, springs are assumed to only exert their reaction on their original axis.

While this effect can often be neglected in many systems characterized by small radial/bending displacements, in axlebox-arm suspensions the orientation of the spring can significantly vary.

As discussed, non-linear springs exert radial forces and torques at their extremes. Most conventional FEM springs implementation do not allow to accurately model this behavior.

- **Non-invariant model:** one of the core assumptions of conventional FEM modeling is that the mechanical properties of the system do not change in their deformed configuration.

Even in the case of a non-linear system such as the axlebox-arm suspension, beam elements do maintain their material and section properties.

This assumption does not hold true for springs, though: as mentioned, the effective stiffness of the spring can vary under load, both due to its intrinsic nonlinear behavior and the change in geometry of the whole suspension. The properties of the system are therefore *non-invariant*.

Linear vs. non-linear differential equations

The considerable difference both in solution complexity and system behavior between a linear and non-linear model can be illustrated by considering the free response simple undamped mass-spring SDOF system.

The linear, and most studied, model adopted in the study of this system leads to the time-invariant differential equation:

$$m\ddot{x} + kx = 0 \implies \ddot{x} + \omega_n^2 x = 0 ; \quad \omega_n = \sqrt{\frac{k}{m}}$$

The general solution can be analytically determined to be:

$$A \cdot \sin(\omega_n t + \phi) \quad A \text{ and } \phi \text{ depend on initial conditions}$$

Many physical system are modeled by differential equations. In particular, time-invariant differential equations are typically adopted: the properties of the system are assumed to not vary in time. As an example, in a mass-spring system, both the mass and the spring stiffness are assumed to be constant.

The solution of time-dependent parameters or non-linear differential equations is much more complex and can typically only be performed with the adoption of numerical methods.

Another relevant issue with the modeling of non-linear system is the interaction between components. As an example, loads alter the spring stiffness; displacements are altered as a consequence, modifying applied loads and so on.

Such complex interactions can typically be handled only by iterative methods.

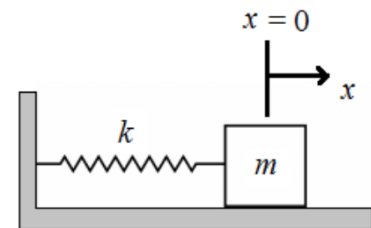


Figure 29: A typical mass-spring system.

The system is at equilibrium when its displacement $x = 0$.

The spring exerts a restoring force in the opposite direction with respect to its displacement $F = -k(x, t) \cdot x$.

While the stiffness $k(x, t)$ can both vary depending both on mass displacement and time, it is typically assumed to be constant to simplify its dynamical modeling.

A possible example of non-linear spring behavior is represented by stiffening under load: the more the spring is compressed, the more its stiffness increases; the opposite is true too: as the spring is elongated its stiffness reduced.

This phenomenon can be modeled by the differential equation:

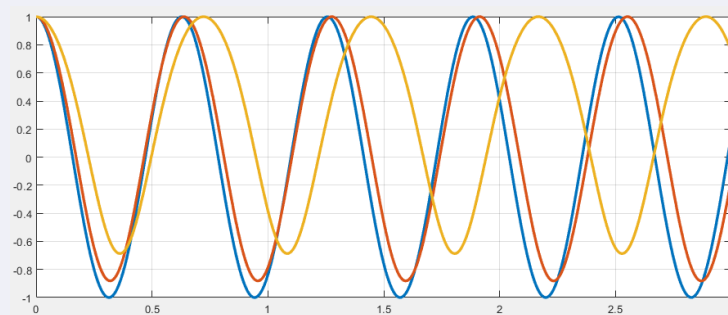
$$m\ddot{x} + k_0(1 - \alpha x)x = 0$$

α is the stiffening coefficient: the higher its value, the stronger the stiffening effect.

k_0 is the stiffness of the spring at $x = 0$.

To allow for comparison between the two cases, numerical simulations can be performed assuming:

$$m = 1 \text{ kg} \quad k = 100 \text{ N/m} \quad x(0) = 1 \text{ m} \quad \dot{x}(0) = 0 \text{ m/s}$$

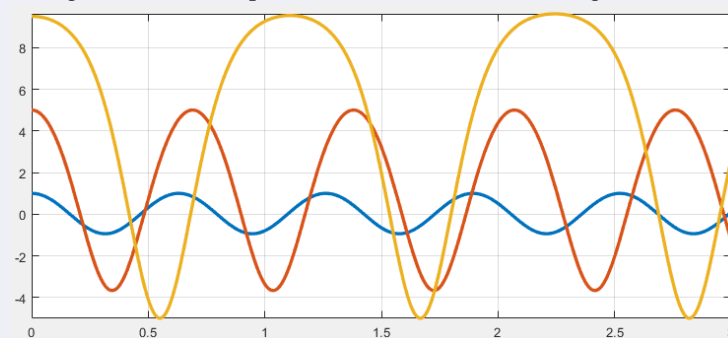


By setting the stiffening coefficient α to be $\alpha = 0$ (*blue*), $\alpha = 0.2$ (*red*) and $\alpha = 0.6$ (*yellow*), it can be observed how both the overall shape and the oscillation frequency are influenced.

In particular, the stronger the non-linear effect, the more results depart from the sinusoidal linear solution.

Another characteristic of non-linear systems is dependence on initial conditions: in the case of a linear system, if the initial position is changed, the amplitude of the oscillations is simply scaled proportionally; the overall shape of the solution, though, is maintained. This is a consequence of the linear nature of the system.

This is not the case when considering non-linear systems. This can be demonstrated by setting $\alpha = 0.1$ and evaluating the free response from different starting conditions.



As the amplitude is increased, the shape of the oscillations is significantly distorted and their frequency is almost halved.

[illegible]

Figure 30: The analytical solution to the case $\alpha = 0.1$.

While in this case it can be evaluated, it is clearly unreasonably complex and cannot be practically employed.

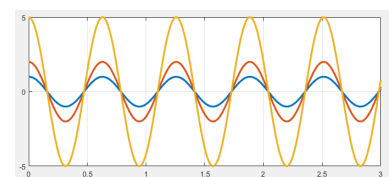


Figure 31: If a linear system is considered, different initial position only affect the amplitude and not the overall shape of solutions.

It is worth mentioning that truly linear systems are rarely found in real applications. Typically, linearizations can be performed to simplify analysis; such approximation can often produce accurate results, yet in some cases (as in this project) they may be insufficient.

Modeling of nonlinear springs

While it is true that some deviation from their undeformed axis may occur, to properly operate it is necessary for coil springs to remain at least somewhat close to it. In fact, the main focus of their design is their axial behaviour and radial/flexural reaction represent secondary effects.

If operating conditions deviate too much from the expected ones, the spring could stop operating properly and start to manifest abnormal behaviors.

These considerations are valid in the contexts of axlebox-arm suspensions too: the spring is assumed to deviate from its undeformed configuration but by a small angle²¹⁷. Lateral displacements too are typically some order of magnitude smaller than vertical ones. While their effect is not negligible, the spring can be considered to still mostly operate on the vertical axis.

Consequently with these observations and coherently with the procedure that is commonly followed in the design of axlebox-arm suspensions, the spring axial stiffness is evaluated first²¹⁸.

Unfortunately, radial and flexural stiffnesses cannot be directly evaluated from the axial stiffness, as they depend on the applied load. They are therefore determined at each iteration of the solution process. The previously described Krettek-Sobczak model is adopted to estimate their values.

Nonlinear FEM: an iterative approach

As mentioned, the unconventional nature of the model to be simulated requires the development of a specific solution for its analysis. As it is often the case for non-linear systems, an iterative approach has to be defined.

As springs are responsible for the non-linear nature of the system, the iterative process focuses on these. The bogie frame, on the other end, is modeled as composed of conventional linear elements. The iterative solution is implemented as follows:

1. **Mode selection:** different loading conditions can be applied to the bogie model²¹⁹. While true operating loads can significantly vary, three static analysis modes²²⁰ have been implemented, considering to be representative and to provide valid insight into the behavior of the bogie.
 - *Vertical load only:* a variable vertical load is applied to the secondary suspensions-frame connection point
 - *Horizontal load only:* a variable horizontal load is applied to the mechanical link
 - *Vertical and horizontal loads:* a fixed horizontal load is applied, while the vertical one is variable.

²¹⁷ typically in the order of few of degrees if not fractions of degree.

²¹⁸ The estimation of a valid axial stiffness is performed by the user. The latter can chose a value based on personal experience, reference projects and available literature.

Another option, though, is to employ the dynamic analysis tool that is included in the project.

²¹⁹ As previously described, vertical and horizontal loads can be applied.

²²⁰ the user selects loading conditions at the start of the analysis process, depending on the objectives of the latter.

Reasons for the adoption of these loading conditions will become apparent as the simulation process is described in more detail and the phenomenon of bogie pitch is explored.

2. **Load set evaluation:** in all analysis modes, only one load at a time is varied. This solution allows for easier result interpretation by the user: simultaneously evaluating the effect of multiple varying loads would in fact be unnecessary complex and produce results of dubious readability.

Clearly, simulating each possible magnitude that the load could assume is unfeasible. A set of loads to be simulated has therefore to be defined. To do so, a minimum and maximum value are set, to prescribe the extremes of the range. A step is then set for the discretization of the interval.

A proper discretization is relevant for good performance: if the discretization step is too small, the excessive number of loads to be tested can significantly increase analysis times; on the other hand, if too few are considered, results may lose in resolution and allow only for gross estimates. A compromise has therefore to be found on a case-by-case basis.

3. **Simulation hyperparameters:** hyperparameters are variables that are set for the whole analysis process, before it is performed. They therefore do not change during iterations or the analysis as a whole. Some examples are:

- *Structural parameters:* this category has been thoroughly discussed and involves all quantities that are necessary to define the mechanical properties of the structure, such as material and geometrical properties²²¹.

Reasonably, these are not expected to change during the analysis.

- *Simulation parameters:* these define the iterative process or other global features of the analysis procedure. Examples are: maximum number of iterations, convergence criteria and tolerances, relaxation factors, etc.

- Other *global properties* of the system, such as the assembly load.

4. **Iterative solution:** a solution must be calculated for each load in the loading conditions set.

If the behavior of the model was linear, this would not be the case, as it would be enough to solve it for a reference value and proportionally scale it to obtain displacements for different loads²²².

As demonstrated in the non-linear spring-mass example, this is not valid for non-linear systems and solutions must be individually evaluated.

To provide some practical examples, in the project:

The minimum load can be set to the minimum expected value during operations. This can be assumed to be when the train is completely empty.

The maximum load can be set to slightly exceed, to guarantee a safety margin, the full load condition.

The step has been set to 1000 kg. This has been determined by testing to represent a valid compromise between accuracy and computational effort.

²²¹ Structural parameters do not only include beam attributes: as an example, spring parameters such as wire diameter and number of active coils are considered too.

In general, all variables that influence the mechanical behavior of the system are included.

²²² as a practical example, if a certain elements manifests a displacement of 1 mm under 1000 N, if the system is linear its displacement under 2000 N (i.e. doubling the load) would be 2 mm (i.e. doubling the displacement too).

Non-linear iterative solution procedure

For each load, a valid solution must respect a set of conditions:

- | | | |
|----|-------------------------------|---------------------------------|
| 1. | $F_z = k_z \cdot \delta z$ | axial spring reaction - z axis |
| 2. | $F_x = k_q \cdot \delta x$ | radial spring reaction - x axis |
| 3. | $F_y = k_q \cdot \delta y$ | radial spring reaction - y axis |
| 4. | $M = k_f \cdot \delta \theta$ | flexural spring reaction |

$\delta x, \delta y, \delta z$: displacements on the x, y and z axis respectively of the spring free end
 $\delta \theta$ difference in y-axis rotation between the extremes of the spring

F_x, F_y, F_z : spring reactions at the free end
 M : reaction torque at spring extremes

To model the spring, a conventional FEM spring element is adopted as the baseline. Such element provides a reaction parallel to the undeformed axis of the spring which, in this case, is the vertical one. Condition 1. is therefore satisfied by default.

The radial reaction of the spring is modeled by virtual forces added at the free end of the node F_x and F_y .

The flexural stiffness is added to the torsional stiffness of the elastic joint. It can in fact be proven that this approach is mechanically equivalent to applying a restoring torque to the ends of the spring directly.

For each loading condition, the following procedure is adopted:

1. As a preliminary step, the system is solved ignoring all non-linear contributions. While this solution is not exact, it provides a useful starting point.
2. The *expected* radial and flexural stiffnesses k_q and k_θ are calculated by the Krettek-Sobczak model. These represent the values that the vertical load on the spring would imply.
3. The expected radial forces are calculated as:

$$\Phi_x = k_q \cdot \delta x \quad \Phi_y = k_q \cdot \delta y$$

4. Expected radial forces are compared to virtual nodal forces at iteration n . In fact, at convergence:

$$\Phi_x^n = F_x^n \quad \wedge \quad \Phi_y^n = F_y^n$$

5. Nodal forces are corrected:

$$F_x^{n+1} = F_x^n + \alpha(\Phi_x^n - F_x^n) \quad F_y^{n+1} = F_y^n + \alpha(\Phi_y^n - F_y^n)$$

During the first iteration, no nodal forces are applied:

$$F_x^1 = F_y^1 = 0$$

6. The total stiffness of the elastic joint is defined to be:

$$k_{ej}^{n+1} = k_{0,ej} + k_\theta^n$$

where k_0 is the fixed stiffness of the elastic element and k_θ^n is the expected flexural stiffness at iteration n

7. The system is solved following the conventional FEM approach. The main difference with respect to the latter is the addition of the virtual forces.
8. The procedure is repeated from 2. until stopping conditions are reached. If valid results are obtained, displacements and other results are selected to be associated to the corresponding loading condition.

As iterations progress, condition 2. and 3. become closer and closer to be exactly respected.

Condition 4. is fulfilled as a consequence: coherence of radial stiffness with the vertical load causes the flexural behavior to be consistent with it too.

In the context of iterative solutions, the introduction of a relaxation factor α is commonly adopted to improve stability and guarantee convergence.

Proper tuning of its value is relevant to obtain proper results: if it is too high, stability could be compromised; if it is too low, convergence may be too slow.

In the project, the numerical solution has proven to be quite robust and relaxation factor up to 1 can often be reached. That said, the default value has been set to 0.5, as it has proven to be a good compromise between convergence speed and reliable results

A large variety of solutions can be adopted to determine when to stop iterations.

In the project, convergence is determined by comparing solutions obtained at consecutive iterations: the difference ϵ between the two is evaluated.

In the project, ϵ is a 4×1 vector containing radial and flexural stiffnesses for the front and rear axes.

Given a tolerance τ different convergence criteria can then be applied. Some examples are:

- $mean(\epsilon) \leq \tau$
- $\left[\sum_{i=1}^N (\epsilon_i)^p \right]^{\frac{1}{p}} \leq \tau$
- $max(\epsilon) \leq \tau$

The latter of these approaches has been chosen as it is the more restrictive condition. That said, in the project results have shown to not vary significantly depending on the convergence criterion.

Analysis outputs

The solution obtained by the analysis process is expressed in the form of nodal displacements. In fact, as previously discussed, the solution of structures by Finite Element Analysis involves the determination of generalized displacement resulting from a known loading condition.

While nodal displacements can be useful to the designer, it is often the case that other derivative quantities are more useful to evaluate the static behavior of the system.

This is the case for bogies too: as discussed, displacements in the bogie frame are typically not very useful to the designer in a preliminary phase; on the other hand, a characterization of the primary suspension system can provide significant insight and guide design choices.

As the numerical solution of the structure is concluded, displacements are collected. The latter are then mathematically manipulated to evaluate relevant quantities as a function of the loading conditions. Results are then displayed to the user in the form of plots.

Preliminary results collection

Some quantities are employed in the evaluation of more than one output quantity. As determining their values is typically straightforward, they are calculated in the preliminary data collection phase to be later employed in the evaluation of more complex quantities.

Examples of quantities that are evaluated in the preliminary phase are:

- **Loads on the supports/axles:** as discussed, extremely stiff virtual springs are introduced to constrain z-axis displacement at supports.

Vertical loads F_z transferred to the axles can be therefore evaluated from the known stiffness of the virtual spring k_v and the calculated z-axis displacement at the support δz_{supp} .

$$F_z = -k_v \cdot \delta z_{supp}$$

- **Torque on elastic joints:** the torque acting on the elastic joint T_{ej} can be evaluated by considering its torsional stiffness k_{ej} and the calculated relative rotation between its extremes $\delta\theta$

$$T_{ej} = k_{ej} \cdot \delta\theta$$

As the virtual spring is much stiffer than other structure elements, its effect on results is negligible.

This is a commonly adopted approach to determine reaction loads at supports in the context of structural FEA.

The evaluation of torque acting on elastic joint allows to determine if the latter are excessively stressed and to estimate their contribution to the non-linear behaviour of the suspension.

- **Spring free end displacements:** while all displacements are recorded, a special treatment is reserved to those at the free end of the spring.
They are stored in specifically introduced data structures for easy access.
- **Free length of the spring:** in the project, the free length H_0 of the spring is not set by the user.
As the bogie design process is performed at assembly conditions, the length h_0 under preload is known: the free length has therefore to be determined from it.
Simulations are automatically performed to determine a value H_0 guaranteeing the spring length to be h_0 at assembly load.

To determine the free length of the spring, a first guess of its value is obtained from the product of its axial stiffness and the assembly load.

To account for non-linearities, an iterative solution process is then performed.

Outputs definition and evaluation

Many plots and graphical renditions can be provided by the analysis tool. As often they are not all necessary at the same time, the static analysis UI allows the user to select which one to visualize.

Definitions and terminology

To properly understand the quantities that are evaluated as outputs, it is first of all necessary to provide some definitions.

- **Relative axle loads** are loads at supports with respect to the assembly condition. R_r and R_f are the vertical reaction at supports at the rear and front axle respectively.

$$R_f = R_r = \alpha \quad \text{at assembly conditions}$$

- **Absolute axle loads** are reactions at supports expressed in absolute terms. The assembly load α is therefore considered too. Total axle loads at the rear and front axles A_r and A_f are expressed as:

$$\begin{aligned} A_f &= R_f + \alpha & A_r &= R_r + \alpha \\ A_f &= A_r = \alpha & & \text{at assembly conditions} \end{aligned}$$

- The **relative secondary suspension load** L , often simply referred to as *relative load*, is the load transmitted by each secondary suspension to the frame.

The **total relative load** L_{tot} considers the sum of all relative secondary suspensions loads. If N_s secondary suspensions are present:

$$\begin{aligned} L_{tot} &= N_s \cdot L \\ L_{tot} &= L = 0 & & \text{at assembly conditions} \end{aligned}$$

- The **absolute secondary suspension load** Λ , often simply referred to as *absolute load*, is the load transmitted by each secondary suspension to the frame when including the assembly load.
The **total relative load** Λ_{tot} considers the sum of all absolute secondary suspensions loads. If N_a axles are present:

$$\Lambda = L + 2 \cdot \frac{N_a}{N_s} \cdot \alpha \implies \Lambda_{tot} = N_s \cdot \Lambda = N_s L + 2N_a \alpha$$

In general, vertical absolute loads are assumed to be positive in the downward direction.

In fact, vertical loads acting on bogies are due to its own weight and that of the vehicle body and other components.

It is extremely unusual, if not impossible, for the bogie to be subject to external vertical loads in the upwards direction.

Coherently with this assumption, relative loads are defined to be:

$$\begin{cases} = 0 & \text{if } \Lambda = \Lambda_0 \\ > 0 & \text{if } \Lambda > \Lambda_0 \\ < 0 & \text{if } \Lambda < \Lambda_0 \end{cases}$$

Where Λ_0 is the absolute load at assembly condition.

In the context of bogie design, loads are typically expressed in terms of kilograms kg in place of Newtons N .

The conversion can be easily performed by considering the gravity acceleration g as a proportionality constant.

While somewhat incoherent from a dimensional perspective, this approach is typically adopted as it simplifies estimates and in general allows for smoother mathematical manipulation during design.

Analysis outputs that are available to the user are:

- **3D displacement visualization:** the structure is represented in an interactive 3D interface, similar to the graphical rendering window adopted for user interaction. Some of its relevant features are:
 - The graphical rendering can be rotated and manipulated by the user
 - Displacements and spring elongations are represented both by representing the deformed configuration of the structure and color coding²²³.
The undeformed configuration is represented too, albeit with lesser graphical weight, to provide a reference.
 - Spring elongation is color coded too with a different color palette. In general, compression is represented by increasingly dark red, tension by increasingly dark blue and near zero elongation by white.
 - External vertical and horizontal loads are represented by red arrows with their tip at the application points.
Support reaction, or equivalently axles loads, are represented by blue arrows with their extremity at support points.
The proportional²²⁴ nature of the arrow visualization allows for an easy first estimate of load magnitude and balance.

As the analysis is performed on a large set of loads, it is not possible to represent each configuration. Due to the non linearity of the system, results under different loads are not the scaled version of each other too.

While this is true, typically the difference is barely perceptible in the visual representation and it is mostly detectable only by mathematical processing. Displacements under the most heavy loading conditions are therefore displayed.

This solution allows the designer to observe the bogie under the most critical conditions and determine extremes for its displacements.

- **Spring elongation:** the elongation of the spring is plotted with respect to relative secondary suspension load $\mathbb{L}$. Two definitions for spring elongation can be adopted:
 - **Relative elongation** $\epsilon = \delta z$ is defined with respect to the assembly load condition.
 - **Absolute elongation** is the overall elongation with respect to the free length and it is defined to be:

$$E = H_0 - h_0 + \epsilon$$

As it can be easily determined, absolute and relative elongation only differ by a constant term.

²²³ Different color palettes are adopted for frame and spring deformations.

Colors representing frame displacements are adapted to the range of displacements and a colorbar is provided as a reference.

Spring elongation is represented by a blue-white-red color spectrum. Compression is represented by increasingly dark red, tension by increasingly dark blue and near zero elongation by white.

As precise data regarding spring behavior is provided as output, the focus of their representation is not on providing exact values, but a general first glance impression of how they are operating.

²²⁴ The sizing of arrows is performed by fixing the length of those representing external vertical loads. The latter do in fact typically determine the scale and magnitude of corresponding reaction forces.

Arrows representing horizontal and reaction forces are scaled accordingly to maintain proportion.

As radial displacements are present at the spring free end, the deformed length $l(L)$ of the spring is different from its height on the vertical direction $h(L)$.

In the project, the elongation of the spring δz is measured on its vertical axis only, yet from a practical perspective the difference between $l(L)$ and:

$$\delta z = h(L) - H_0 \approx l(L) - H_0$$

This solution is preferred as the vertical displacement is necessary to evaluate other quantities too.

Relative elongation is preferred in the context of analysis output as it provides better readability and it is better suited for user interpretation. It is in fact more convenient from a practical perspective to express displacements with respect to the assembly reference condition.

As it will be later discussed, absolute elongation is employed in the evaluation of other output quantities.

To reflect the common practice adopted in the industry, in the output plot elongations are expressed in millimeters *mm* and loads in kilograms *kg*.

- **Elastic joint torque:** as discussed, the torque acting on elastic joints is evaluated in the preliminary phase of the analysis.

As both the torsional deformation and the reaction torque of the elastic joint are null at assembly conditions, torque is plotted with respect to relative load L .

By default, torque is expressed in $N \cdot m$, yet its unit can be set to $kg \cdot mm^{225}$ if preferred by the user.

As in previous cases, the relative load is expressed in kilograms.

- **Load balance ratio:** in the context of bogie design, the evaluation of axle loads in absolute terms does not typically provide significant insight. In fact, axle loads vary as external loads change.

Load balance ratios λ_r and λ_f are evaluated for the rear and front suspension respectively; they are representative of how loads are distributed between axles and provide better insight on the effect of external loads on bogie attitude.

$$\lambda_r = \frac{A_r}{|A_r| + |A_f|} \quad \lambda_f = \frac{A_f}{|A_r| + |A_f|}$$

In practical applications²²⁶ the following relation holds true:

$$\lambda_r + \lambda_f = 1$$

Due to their definition, load balance ratios are adimensional. They are plotted with respect to the relative load L .

- **Pitch angle:** the pitch angle β is representative of the global rotation of the bogie along the y -axis and it is in fact defined to be coherent with the latter.

While frame elements can present different rotations, a global approximation can be defined by considering displacements at the free end of the front and rear spring:

$$\beta = \text{atan} \left(\frac{\delta z_f - \delta z_r}{L_0 + \delta x_f - \delta x_r} \right) \implies \beta = 0 \text{ at assembly conditions}$$

As its reference is the assembly condition, it is plotted with respect to relative loads. It is expressed in degrees for readability, but it can be set to radians if required.

As the elastic behavior of the joint is assumed to be linear, the relative rotational displacement at its extremities and the reaction torque are directly proportional.

To avoid the unnecessary representation of both, torque has been chosen as it is more representative of stresses withstood by the elastic joint.

²²⁵ this unit is typically adopted in the context of railway vehicles design.

²²⁶ In the evaluation of load balance ratios, loads are considered positive if applied to supports in the negative z direction.

As a consequence of their definitions:

$$\begin{cases} A_r > 0 \wedge A_f > 0 \implies \lambda_r + \lambda_f = 1 \\ A_r < 0 \wedge A_f < 0 \implies \lambda_r + \lambda_f = -1 \\ A_i < 0 \wedge A_j > 0 \implies \lambda_i + \lambda_j = \frac{A_j - |A_i|}{A_j + |A_i|} \end{cases}$$

From a practical point of view, only the first case is to be considered. The presence of negative absolute loads would in fact not be compatible with proper bogie operation.

- **Specific stiffness:** the specific stiffness σ is defined to be:

$$\sigma(\delta z) = \frac{A}{E} = \frac{R + \alpha}{H_0 - h_0 + \delta z}$$

As the load on the suspension is varied, its stiffness varies too due to non-linear effects. While the axial stiffness remains constant, the specific stiffness provides information on the vertical behavior of the spring when introduced in the context of the axlebox-arm suspension system.

The specific stiffness is also known as *system stiffness*. In fact, it is representative of the vertical stiffness of primary suspensions as a whole.

Following the commonly adopted practice in bogie design, specific stiffness is expressed in kg/mm .

It is by default represented with respect to the absolute load, as the latter is more coherent with its definition. Nevertheless, the user can set it to be presented with respect to relative load if preferred²²⁷.

- **Global stiffness:** the global stiffness σ_γ is defined to be:

$$\sigma_\gamma(\delta z) = \frac{\Lambda_{tot}}{E} = \frac{N_s \cdot L + 2N_a \cdot \alpha}{H_0 - h_0 + \delta z}$$

While the specific stiffness focuses on the suspension system only, the global stiffness describes its behavior in the context of the bogie by considering the total absolute load Λ acting on it.

As in the case of its specific counterpart, it is expressed in kg/mm and can be presented both as a function of relative and absolute loads. As the total absolute load Λ_{tot} is implicit in its definition, this is the option that is adopted as default.

If the suspension system was linear, the specific stiffness would be constant, as no dependence on load would manifest.

In spite of this, its value would not exactly match that of the spring axial stiffness itself, as the elastic joint provides a contribution too. The latter is typically quite small, yet large enough to absorb a detectable fraction of the load.

To provide a numerical reference, the elastic joint typically absorbs $\sim 5 - 10\%$ of the axle load.

²²⁷ From a practical perspective, absolute and relative loads only differ by a constant terms. The choice between the two is only a matter of readability and does not alter the overall shape of the plotted results.

As it will be demonstrated by practical examples, the specific stiffness is an intrinsic property of the primary suspensions and does not depend on the surrounding bogie frame. It is therefore more useful in the context of suspension system design.

The global stiffness, on the other hand, provides insight in the behavior of the bogie as a whole. It describes the reaction of the suspension as external loads are modified.

Results and observations

To provide a practical demonstration of the capabilities of the modeling that has been implemented, testing on a reference bogie has been performed.

The main objective is to evaluate the coherence with expectations and the validity of results that are obtained by the static analysis process. As a consequence, the non-linear behavior of the primary suspension system can be observed and evaluated too.

In the preliminary phase of testing, vertical loads only have been considered. As it will be later discussed in more detail, horizontal loads introduce additional effects on the results, that may complicate the interpretation of the latter²²⁸.

²²⁸ While this may be useful in the actual design process, the addition of unnecessary complication has been deemed unsuitable for the preliminary testing phase.

Horizontal loads are not excluded from testing, yet as they require an appropriate introduction they will be discussed in the next chapter.

Bogie 01

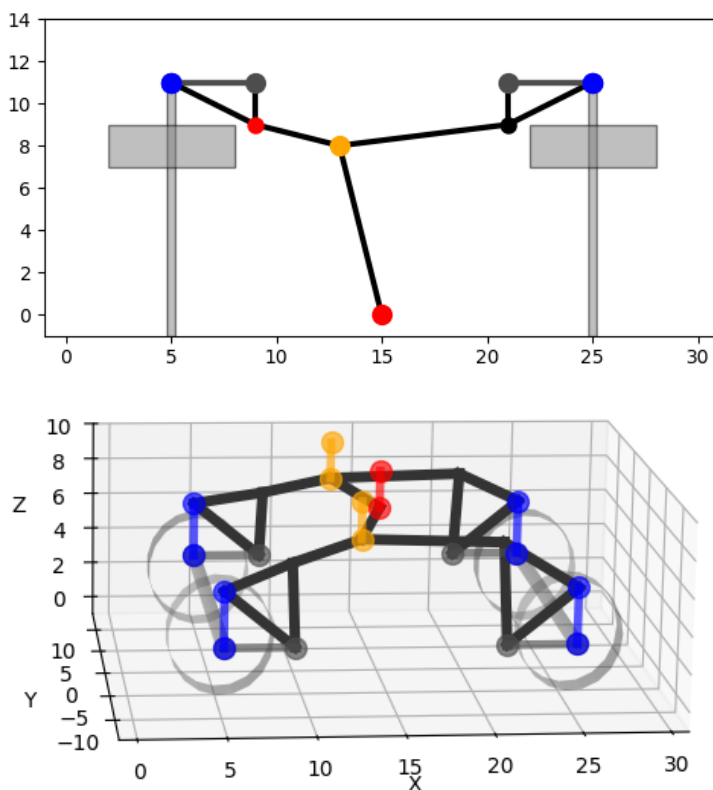


Figure 32: Relevant features of the Bogie 01 model:

Wheelbase: 2000 mm
Gauge: 1450 mm
Spring length at assembly: 30 mm
Guide arm length: 40 mm
Assembly load: 7012.5 kg
Spring stiffness: 954 N/mm
Elastic joint stiffness: 90 Nm/°

Primary suspension parameters are selected to reflect those of a well-characterized and known suspension system for which reference data exists. This solution has allowed for comparisons to check the validity of the implementation.

Notably, the bogie design is asymmetric. It is assumed to predominantly operate in the positive-to-negative x-axis direction.

Loading condition:

Vertical only loads, applied at the secondary suspension attachment point.

Load range: $L = [-7500, 11500]$ kg

Load discretization step: 1000 kg

Bogie 01 is chosen as the design to be tested in this demonstration. It represents a possible configuration that could be obtained during the design process. It does not therefore represent a finished layout, but a model that has to be improved by the designer.

A relevant feature of *Bogie 01* is its longitudinal asymmetry. While the shape is almost symmetric, slight differences exist between its front and rear.

While most bogies are designed to be longitudinally symmetric²²⁹, this is not a mandatory requirements and asymmetric solutions exist. A slight longitudinal asymmetry has therefore been introduced to test the effect on balance and to verify the effective evaluation of load distribution and primary suspension behavior.

Results of the analysis process are hereby presented and commented:

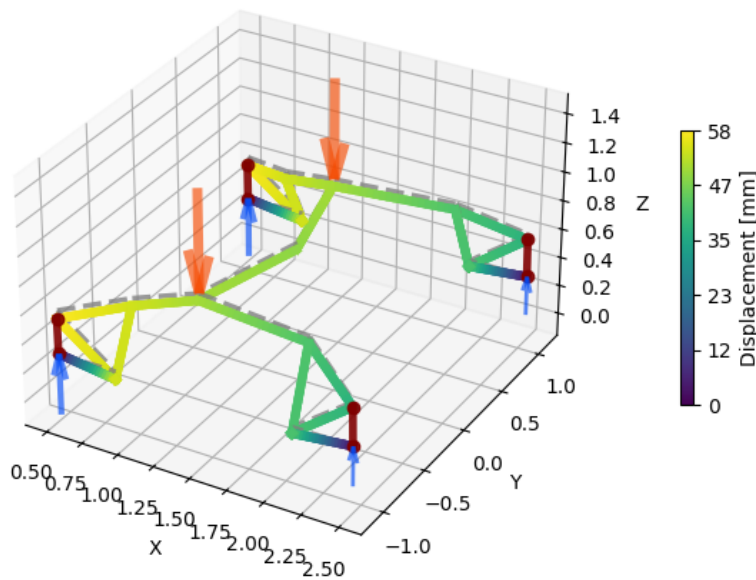


Figure 33: **Graphical representation of displacements**

External loads and support reactions are represented. The slight load unbalance towards the front of the bogie can already be assessed.

Both springs are compressed, as expected. If this was not the case, balance issue may arise.

The structure as a whole does not seem to present particular structural criticalities and can be considered compatible with operations.

As previously mentioned, the frame as a whole maintains its shape almost unvaried (even in the maximum load case that is represented in the figure) while most of the displacement occurs in the suspension system. This is expected and desirable.

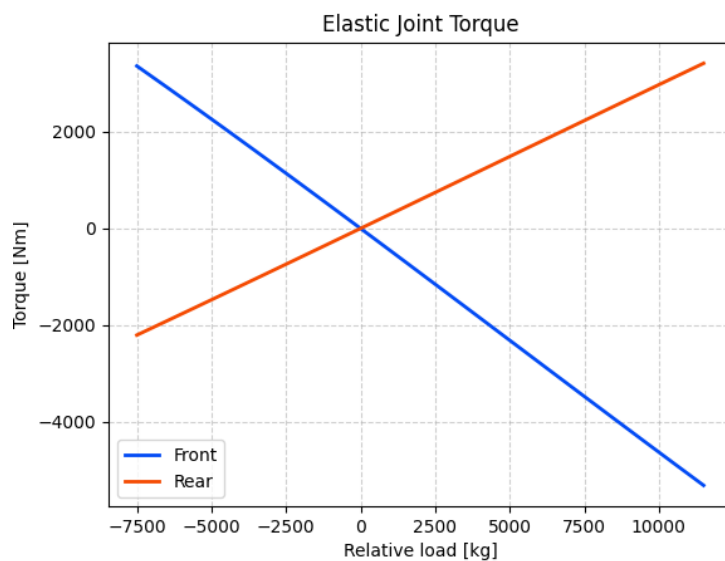


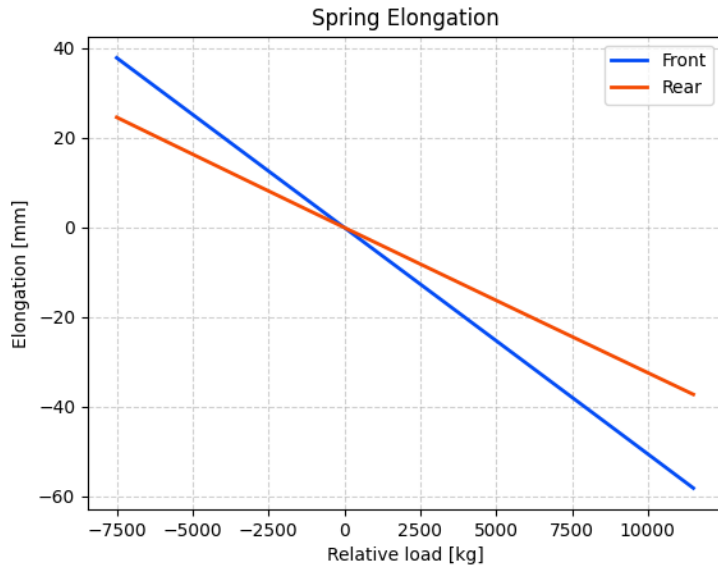
Figure 34: **Elastic joint torque**

The torques acting on the elastic joints, and their relative rotational displacements as a consequence, present an almost linear behavior. In fact, from a graphical-only perspective it would be very difficult to detect any non-linearity.

Notably, front and rear elastic joints withstand torques in opposite direction; this is to be expected due to their position and role in the system.

As expected, both torques are null at assembly load. Notably, front elastic joints are more loaded.

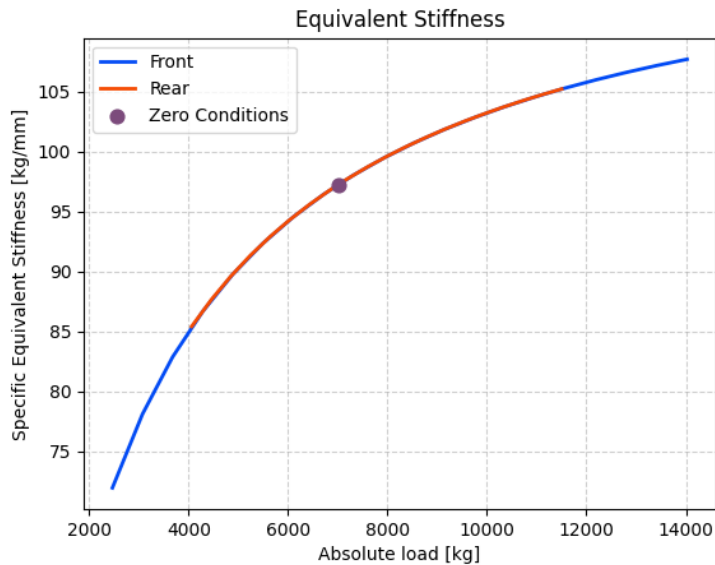
The main insight of the analysis of the elastic joint torque is to evaluate operating ranges and measure maximum stresses.

Figure 35: **Spring elongation**

As in the case of elastic joint torques, non-linearities are barely detectable. The spring elongation plot main purpose is therefore to evaluate operating ranges and observe relevant features of the spring behavior.

As an example, the different load balance between the front and the rear can be easily detected. The front spring deflections is ~ 1.5 times greater in magnitude than the rear one.

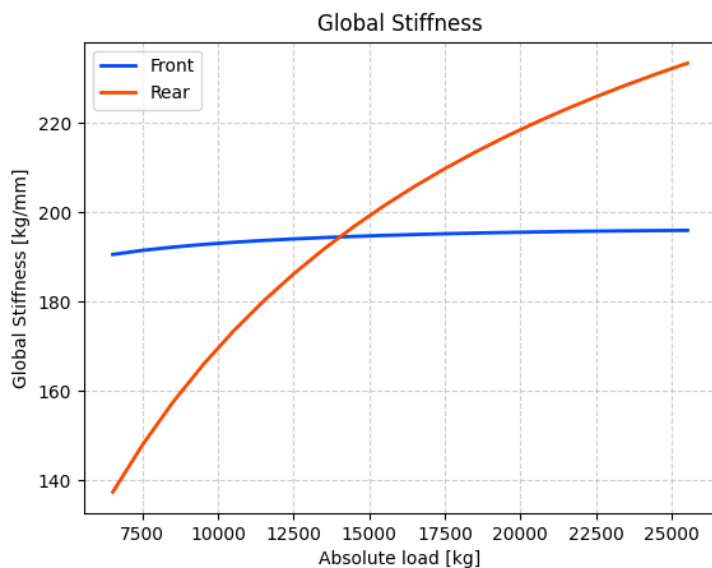
Notably, at assembly conditions spring displacement are null. This is coherent with the adopted convention. As the load is increased, both springs are compressed; if the opposite is true, their length is increased, yet still lesser than the free length.

Figure 36: **Specific stiffness**

As mentioned, the specific stiffness can be considered an intrinsic property of the suspension system. This is confirmed by presented results: the behaviors of the front and rear suspensions are coincident, despite their difference in loading.

The only effect of the latter is that different ranges of loads are explored. As the front suspension encounters a wide range of stresses, its specific stiffness can be evaluated in a larger number of conditions.

Notably, at assembly load the specific stiffness is coincident with the axial stiffness of the spring. A remarkable stiffening behavior can be observed; it becomes even more significant as the load is decreased, causing a sharp loss of stiffness.

Figure 37: **Global stiffness**

The analysis of global stiffness results is not straight-forward and requires user experience as knowledge of the examined design.

In the case of *Bogie 01* it can be observed that the behavior of the front suspension remains quite unaltered with respect to secondary suspension total absolute loads.

The rear suspension systems presents a remarkable stiffening: as the load increases, the resistance it opposes to external load increases significantly. On the other hand, under low-load conditions, its effect reduces, possibly causing significant unbalances.

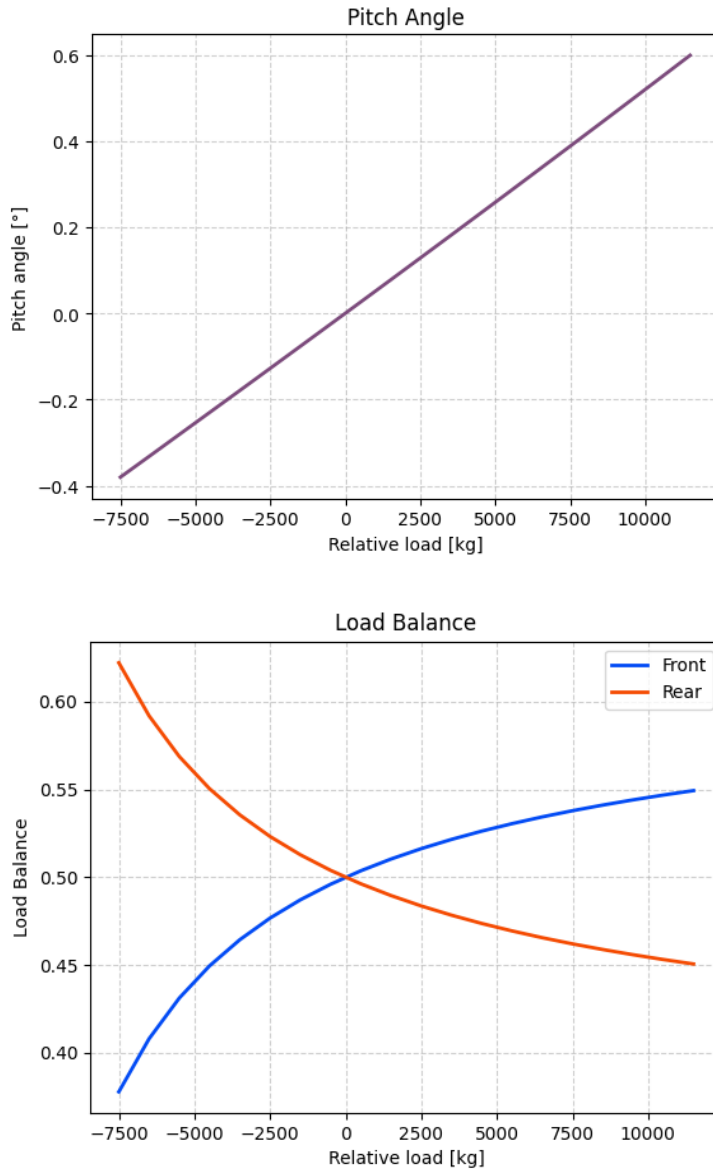


Figure 38: Pitch angle

The behavior of the pitch angle is coherent with expectations:

At assembly load, the angle is null as per its definition. As the load is increased, the bogie pitches more and more forward. The opposite is true if unloaded.

The pitch angle is typically small and in many cases its absolute value is lower than one degree. Understandably, this is to be expected to guarantee proper bogie operation.

While being quite linear, some deviation from a perfectly linear shape can be observed. In fact, in more unbalanced bogies, this feature becomes even more apparent.

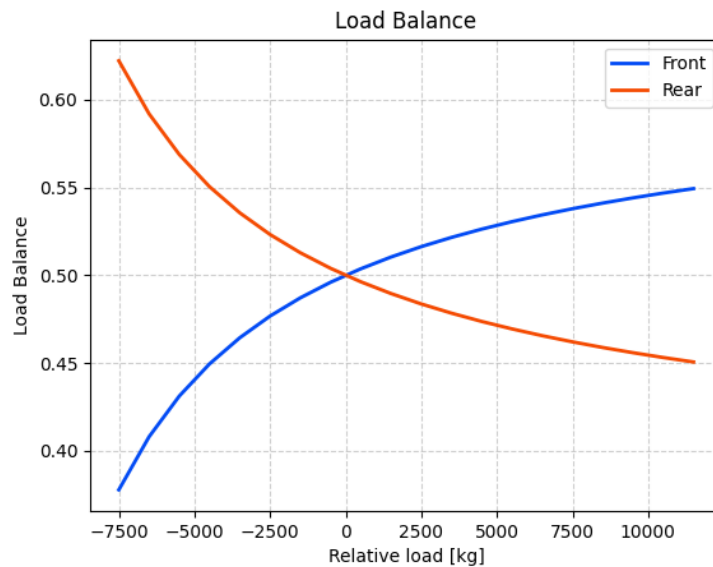


Figure 39: Load balance

The load distribution is remarkably non linear and strongly depends on the applied load.

It is worth mentioning that *Bogie 01* presents a relatively small unbalance in its load distribution. For more unbalanced designs, the behavior can be even more far from an uniform load distribution.

As the external load is increased, the front becomes more loaded than the rear. Below assembly load, the opposite is true.

Remarkably, the effect of lowering the applied load is significantly more unbalancing than the opposite. This can be intuitively explained by considering that, as external loads are increased, the system becomes more stiff and therefore more stable.

Practical operating loads

The range of external loads that is typically considered in the static analysis of the bogie is very large. The purpose of this choice is to evaluate the system under all possible operating condition.

From a practical perspective, though, loads than are realistically encountered during operation lie on a smaller range.

To allow for proper comparison between different designs, in this demonstration the operating range has been assumed to be $L = [-5000; 7500]$ kg.

Relative loads lower than $L = -5000$ kg do in fact manifest an evident loss in stiffness and stability. Operation in such conditions would therefore not be recommendable.

Relative loads higher than $L = 7500$ kg have been deemed to be excessively high for realistic applications.

A case study: bogie pitch analysis

Up to this point, only vertical loads have been considered in the static analysis. While this is a useful benchmark for the design process, this is not necessarily representative of operating conditions.

Horizontal loads are typically transmitted through the bogie-body mechanical link during operations, both in the case of motor and trailing bogies:

- In *non-motorized bogies*, a force must be transmitted from the vehicle body to the bogie to force the latter to move and follow the rest of the vehicle.
- In *motorized bogies*, the tractive force must be transmitted from the bogie to the vehicle body. As traction is generated at the bogie, the proper transfer of the corresponding horizontal load is necessary for vehicle and train propulsion.

In both cases, if a brake system is installed on the bogie²³⁰ braking loads are transferred too.

As horizontal loads are applied, a torque on the y -axis is generated. This phenomenon can significantly alter the attitude of the bogie and its pitch angle.

This effect is particularly evident in symmetric bogies, in which vertical loads alone are not able to generate any y -axis torque. In this case, under vertical load, the pitch angle is fixed at 0° . As horizontal loads are introduced, though, significant alterations of the pitch angle are observed.

Even in the case of asymmetric bogies, such as *Bogie 1*, horizontal loads can further contribute to load unbalances and often have a considerable effect. In fact, even if the bogie is not symmetrical, torques generated by vertical loads alone should be relatively small²³¹.

Bogie pitch under traction and braking loads should be carefully accounted for in the design process. In fact, significant load unbalance between axles can arise, considerably impacting the behavior of the bogie and the vehicle.

If one of the axles is too unloaded, the lack of normal force in the wheel-track contact can significantly decrease the ability to transfer tractive power by friction.

On the other hand, excessive loads can damage components, induce excessive stresses and, in general, affect the performance of the axle.

The main difference between horizontal loads in motorized and non-motorized bogies is the direction of the transmitted force.

As it can be intuitively understood, in the first case the bogies push the body in the direction of motion. In the opposite case, the body pulls the bogie, inverting the direction of the load.

²³⁰ this is the case in many vehicles and bogie arrangements.

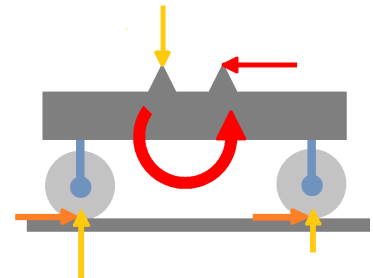


Figure 40: Schematic representation of the effect of the introduction of horizontal loads.

While vertical loads alone can produce y -axis torque, horizontal loads typically provide a significant contribution to it and measurably alter the bogie attitude.

²³¹ This effect can be observed in the results of the previous simulation. Even if some load unbalance is present, the pitch angle is typically only fractions of degree and rotations are relatively contained.

Testing and results

To properly assess the effect of horizontal loads, *Bogie 01* is again tested: as its response to vertical loads has already been determined, a comparison can be performed to evaluate the effect of horizontal loads.

The bogie is considered to not be motorized. The horizontal load acting on it is therefore exerted along the direction of motion.

Notably, in such conditions the additional horizontal load is expected to worsen the unbalance observed in the previous analysis.

The horizontal load has been set at $H = 3500 \text{ kgf} = 34335 \text{ N}$.

This value has been estimated both by available literature and calculation from bogie parameters. This is considered to be a good approximation of real operating conditions²³².

²³² The assumption of traction and braking loads to be horizontal is in itself an approximation.

In general, properly modeling and correctly estimating mechanical interaction of such complexity is not a trivial objective and requires detailed information about the model. The traction load itself can significantly vary during operation due to changes in wheel friction, inclines, power and speed requirements, etc.

The value that has been chosen for this analysis has therefore to be intended as indicative and not as an immutable reference.

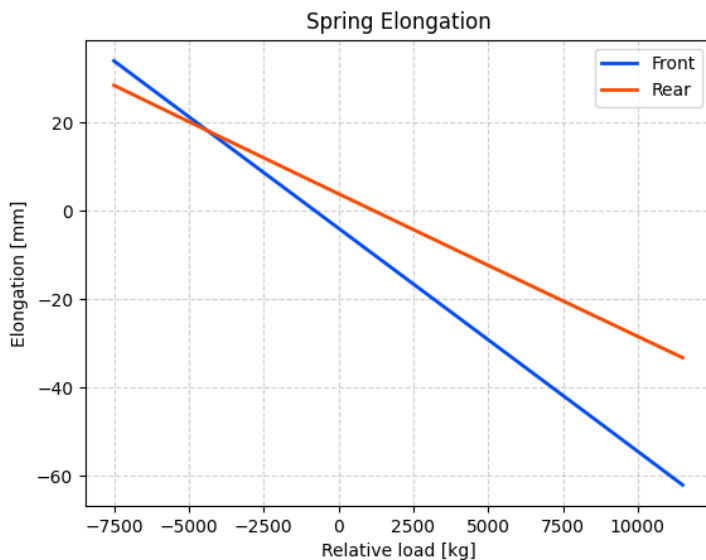


Figure 41: **Spring elongation**

A significant shift is observed in the trend of spring elongations.

Springs elongation at assembly load are not null anymore, as even in this case compensation for the horizontal load must be provided.

In general, a slight increase in the front load and a corresponding decrease in the rear load can be observed. The difference in elongation is therefore more pronounced.

Notably, the two suspension system present the same elongation near the lower extreme of the operating range.

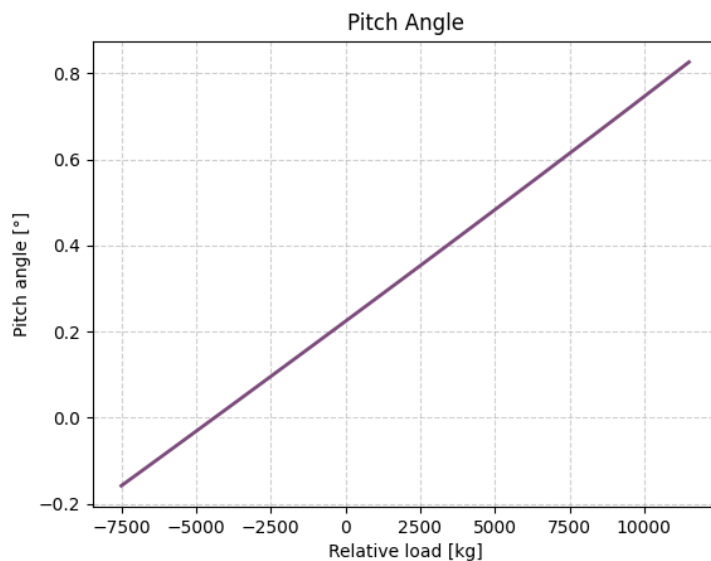


Figure 42: **Pitch angle**

While presenting a similar trend and shape to the vertical only case, the pitch angle is considerably altered. In fact, in most of the operating range the pitch is positive, implying a front pitch attitude.

The maximum pitch is increased too and surpasses 0.8° , with respect to 0.6° in the vertical only case.

In general, the pitch curve maintains its shape almost unvaried and is mostly shifted upwards by the introduction of the horizontal load.

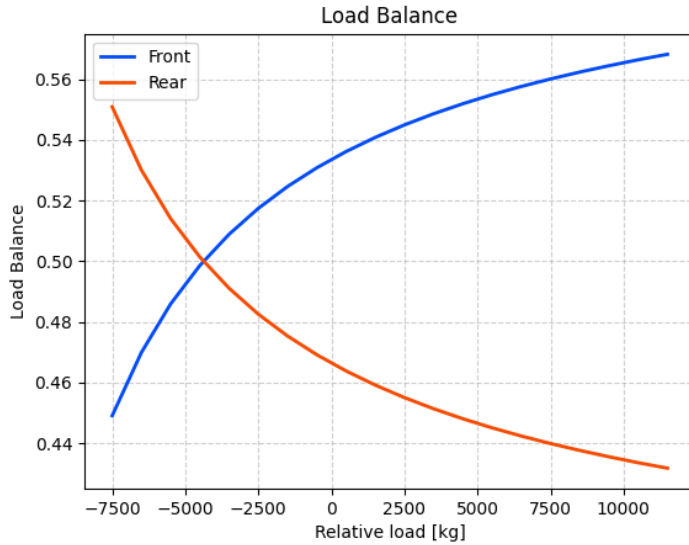


Figure 43: Load balance

The analysis of load balance provides significant insight in how the behavior of the bogie is altered by the introduction of horizontal loads.

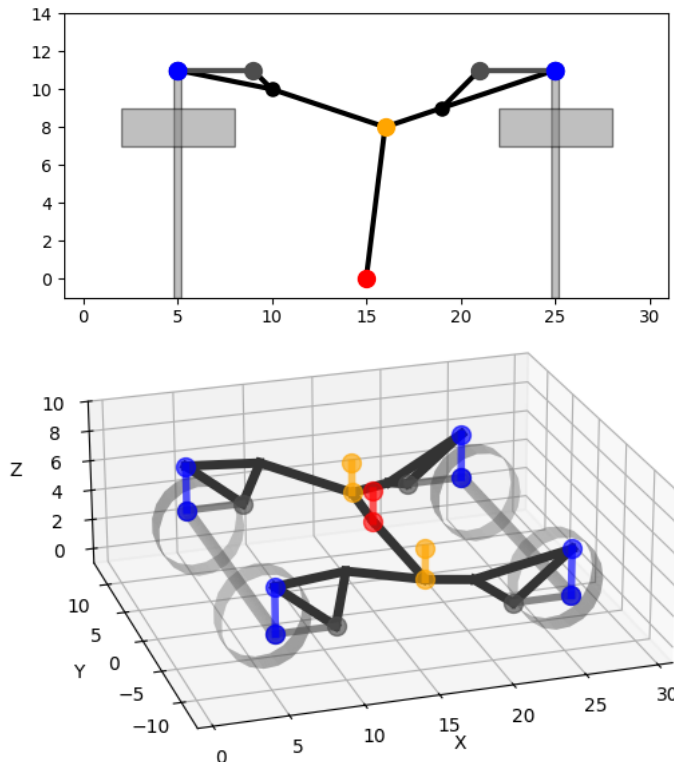
A first very apparent feature of this operating condition is that loads are balanced only at the lower extreme of the practical operating range ($L = -5000 \text{ kgf}$).

It can be therefore assumed that the front axle would realistically always be more loaded than the rear. This has relevant implication on long-term operation and maintenance of the suspension system.

Furthermore, the difference between the front and rear load is increased, leading to worst performance and distribution of tractive power.

Bogie 02

In the GUIDE approach, at each iterations designs are refined with the objective of improving the coherence with requirements and user expectation. To demonstrate how this could be realized, previously presented results have been analyzed and employed to improve the performance of *Bogie 01*. A new design, *Bogie 02*, has been consequently produced.

Figure 44: Graphical visualization of *Bogie 02*

Notably, its design is not radically different from *Bogie 01*. In fact, its topology is preserved and only minor geometrical corrections are implemented.

The most relevant upgrades are:

- The height of the mechanical link connection is significantly lowered. Horizontal forces have therefore a reduced available arm to generate torque and their effect is therefore decreased. As a consequence of this modification, the bogie as a whole is reduced in height. Typically, this solution improves stability too.
- Secondary suspension attachment points are shifted backwards. As a consequence, positive vertical loads cause a negative y -axis torque, partly balancing the effect of horizontal loads.
- Frame connections to primary suspension are more vertically aligned, decreasing lateral loads on the latter.

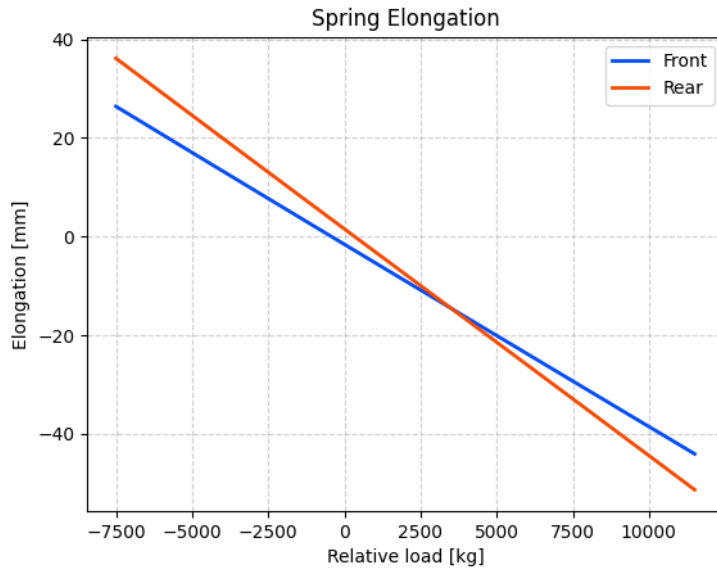


Figure 45: Spring elongation

With respect to *Bogie 01*, spring elongation is significantly more similar between the two suspensions. Operating conditions are therefore expected not to differ significantly between the front and rear axles.

The maximum spring elongation is reduced too, as the load is more uniformly distributed.

All these factors not only contribute to improve the stability of the bogie but reduce the effects of fatigue: lower maximum loads imply reduced stresses while similar loading conditions on the two suspensions allow for more uniform wear and, as a consequence, maintenance.

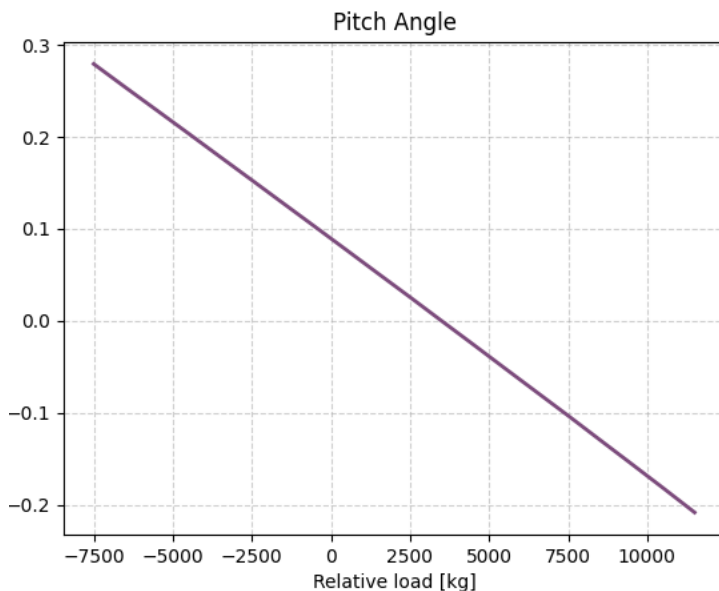


Figure 46: Pitch angle

The range of the pitch angle is significantly reduced. In fact, in operating conditions it is expected to be comprised between 0.15 and -0.1 . This represents a significant improvement with respect to *Bogie 01*.

Notably, in this configuration the pitch angle trend is inverted: it decreases (i.e. the bogie pitches toward the rear) as loads are increased. This slight rear pitch tendency is introduced to increase stability and compensate for possible increases in traction load.

In general, the range in which the pitch angle varies is considerably reduced and, when compared to *Bogie 01*, the trend appears to be much flatter.

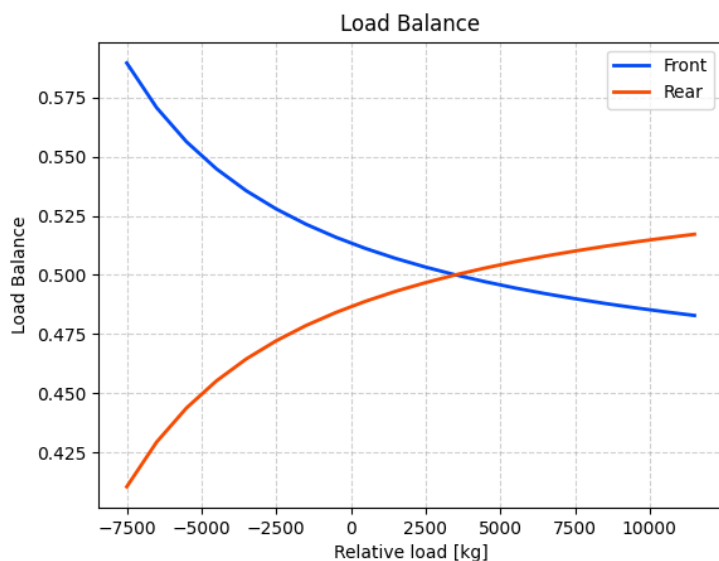


Figure 47: Load balance

The load balance is significantly improved, especially in load ranges that are more likely to be encountered in operations.

In fact, above $L = -2500 \text{ kgf}$ the load distribution is considerably uniform, with load balance ratios in the range $\lambda_r = 0.47 \div 0.52$ and $\lambda_f = 0.51 \div 0.49$.

The load balance is especially improved near assembly conditions, where most operations are supposed to occur. In this case, ranges are even more restrictive.

These improvements came at the cost of worsened performance at very low loads. The latter, though, are assumed to not be encountered in operations and can therefore be sacrificed to improve the overall performance of the bogie.

Conclusions

The objective of this work was to showcase the feasibility of introducing artificial intelligence in the design process. This goal has been successfully achieved through the development of an AI-User interaction framework and, consequently, a functional software prototype. The latter has demonstrated that such an approach is both possible and effective. The results obtained are satisfactory and highlight the potential of AI to support and enhance the early stages of design.

Nevertheless, due to restrictive time constraints, it was not possible to carry out practical testing in the field. As a result, parameter optimization and further fine-tuning of the software could not be performed. In addition, some more complex features that had been envisioned at the outset were not implemented within the available timeframe.

Despite these limitations, the project can be considered successful. The developed framework represents a valid and solid base for future upgrades.

From a practical perspective, with further time and effort the software prototype could be enriched with additional functionalities, refined through real-world testing, and ultimately established as a valuable tool to support designers in their work.

From a more general perspective, the underlying theoretical and mathematical foundation has proven to be valid and possibly adaptable to other systems and structures to be designed.

Bibliography

- [1] F. di Majo, *Costruzione di Materiale Ferroviario*
- [2] N. Bosso, *Slides from the course "Progettazione Meccanica e Trazione Elettrica di Veicoli su Rotaia"*
- [3] G. Genta, *Vibration of Structures and Machines*
- [4] G. Peyrè, *The Mathematics of Artificial Intelligence*
- [5] S. S. R. Anjaneyulu, *AI-Based Optimized Design of Structural Frames: A Guide to Structural Design Automation*

Images Sources

- [1] Wikimedia Commons Archive, <https://commons.wikimedia.org>:
Figures 1, 2, 3, 8, 9, 13, 18, 21, 24, 27
- [2] Slides from the course "Progettazione Meccanica e Trazione Elettrica di Veicoli su Rotaia":
Figures 4, 5, 6, 7, 10, 11, 14, 15, 17, 19, 20, 28
- [3] ResearchGate, <https://researchgate.net>:
Figures 12, 16

Not previously mentioned figures are to be considered original production by the author or (*Figures 22, 23, 24, 29*) significant re-elaboration of pre-existing material.

Acknowledgments

I would like to express my sincere gratitude to all those who supported me throughout the development of this thesis.

I would like to thank Marco Borlo and Pierangelo Farina, who welcomed me to Blue Engineering and introduced me to its work environment, providing the foundation for my experience. Their prompt responses and readiness to assist, both at the start of my training and throughout the subsequent phases, were extremely helpful in allowing me to smoothly integrate into the company and successfully carry out my work.

I extend my thanks to Emilio Sapochetti, who facilitated my connection with the company and enabled me to pursue this project in a professional context.

More in general, I would like to thank all those at Blue Engineering who generously shared their time and expertise, assisting me with practical insights and helpful feedback during the development of this project.

I would like to thank to my academic tutor too, Prof. Nicolò Zampieri, for their support, feedback, and guidance throughout the preparation of this thesis.

Finally, I wish to thank my family, friends and loved ones for their unwavering support, patience, and encouragement throughout this journey. Their presence truly made this experience possible.