NeTrainSim V0.1.1 Beta Manual

Development:

Ahmed Aredah, M.Sc.
Ph.D. student, Dept. of Civil and Environmental Engineering, Virginia Tech
M.Sc. Student, Dept. of Computer Science | Engineering, Virginia Tech
Graduate Research Assistant at Virginia Tech Transportation Institute
E-mail: AhmedAredah@VT.EDU

Karim Fadhloun, Ph.D.
Research Associate at Virginia Tech Transportation Institute
E-mail: Kfadhloun@vtti.vt.edu

Hesham A. Rakha, Ph.D. P.Eng., F.IEEE
Samuel Reynolds Pritchard Professor of Engineering, Charles E. Via, Jr. Dept. of Civil and Environmental Engineering
Courtesy Professor, Bradley Department of Electrical and Computer Engineering
Director, Center for Sustainable Mobility at the Virginia Tech Transportation Institute
Fellow of Asia Pacific Artificial Intelligence Association
Fellow of the American Society of Civil Engineers
Fellow of the Canadian Academy of Engineering
Fellow of IEEE
E-mail: hrakha@vt.edu

Center for Sustainable Mobility
Virginia Tech Transportation Institute

2023

Table of Contents

Table of Figures

Table of Tables

Simulator Description

The Network Multi-Train Simulator (NeTrainSim) is a software application designed to simulate the movement and interactions of multiple trains within a graph-based network, where points are connected by links and signals are placed only at the intersection of these links. The simulator analyzes the network and trains paths to identify potential collisions and resolves them in a first-come, first-served manner by adjusting the network signals state. Additionally, the simulator takes into account train following, train dynamics, and external resistance.

The simulator operates on a time-driven algorithm utilizing a time step as the base for the calculations. It is recommended to use a small-time step to minimize errors in the simulation results. The algorithm progresses through each time step, updating the positions and states of the trains and analyzing their interactions within the network. At the end of the simulation, a summary file is generated with the key information about the trains' performance. The summary file includes details such as travel time, traveled distance, and consumed energy.

The simulator handles different train consists, each with its own locomotive characteristics and power type. The model accommodates various energy sources, including diesel, biodiesel, diesel-hybrid, biodiesel-hybrid, and electric trains.

Prerequisites

There is no prerequisite required to run NeTrainSim as the installer has all the required 3rd party packages.

** To edit the code, the Qt V6 must be installed on your machine along with the MSVC compiler and visual studio 2019 from Microsoft.

Simulator Schema Description

To efficiently handle the simulation, NeTrainSim is divided into several modules:

1. Network Module:
   The Network Module handles the calculations related to the network and defines its structure. It establishes the connections between points, links, and signals within the network. This module is responsible for ensuring the integrity and proper functioning of the network during the simulation.
2. Train Dynamics Module:
   The Train Dynamics Module defines the characteristics of each train, including its physical properties and movement dynamics. It incorporates realistic train behavior, such as acceleration, deceleration, and braking, to simulate the trains' movements within the network.
3. Energy Module:
   The Energy Module is responsible for calculating the energy consumption of each train during the simulation. It considers the different energy sources available. The module accounts for the specific characteristics of each train consist and estimates the amount of energy consumed based on the simulated operations.
4. Simulator:
   The simulator module is responsible for the management of all trains on the network. It ensures the integrity of the simulator as a whole. It is also responsible for writing the output files.

By utilizing these modules, the NeTrainSim offers a comprehensive simulation environment for analyzing the behavior of multiple trains on a network. Its modular structure allows for the efficient handling of network calculations, train dynamics, and energy consumption calculations, providing accurate and detailed simulation results.

To define a network in the simulator you need to define nodes and links files (Figure 1). An alignment/train track is represented as links in the simulator (Appendix 2). A node is a connection point between two links (Appendix 1). Links could be separated by a node when links have different grades and curvatures, or when there is an intersection between the links. Signals are important, especially in a collision zone. The simulator uses these signals to allow passing only the first train arriving at the node while stopping any other opposite-direction trains.

To define a train, you need a train characteristics file, this file (Appendix 3) contains all the simulated trains with their characteristics and paths. The trains’ paths should be defined as a vector instead of a single value. A train path is a set of nodes’ IDs where that train will be passing. If in the links file, there is a stopping node, the node, in this case, is considered a stopping station where the train comes to a complete stop and accelerates again from zero speed. If only one train is moving on the network, the signals are not important in that case because all of them will always be green all the time anyways. If, on the other hand, many other trains are on the network, then we have two scenarios. The first scenario is when there is no conflict between the trains and accordingly the trains move freely without any restrictions. In contrast, if the trains' paths intersect at any point, this intersection becomes a conflict point and the simulator deals with it by signals.

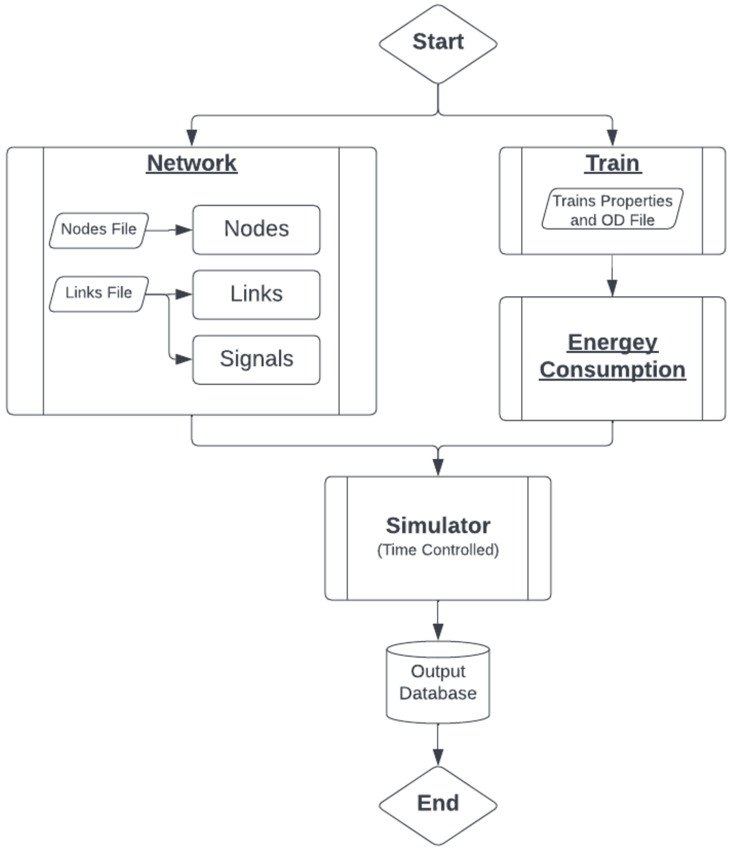


Figure 1: Simulator Schema

You can run the simulator at a specific time step. A time step is in seconds, and it is the time step that a train takes to calculate the next position it will be at with a certain speed and acceleration. In addition to that, the energy consumption of the train is calculated and reported. At each time step, the simulator will report its coordinate on the network, how much it traveled on its path, and its instantaneous current speed, acceleration, and energy consumption. The units used in the simulator output are in the metric system; time in seconds, distance in meters, speed in m/s, and acceleration in m/s^2. Finally, there is a simulation summary file that gives a quick look at the simulation's total actual and simulated times, total distance traveled, and cumulative energy consumption in kW.h.

Considerations and Limitations

The network links are assumed to be linear. This is because the simulator uses vectors to reduce the calculation time. Vectorization requires the links to be linear instead of curves since curves are composed of millions of approximated vectors.

In this simulator, trains are allowed to span as many links as they could and therefore every car or locomotive may have a different grade, curvature, and maximum allowed speed. The train is not allowed to speed up more than the lowest free flow speed of any of the train-spanned links. Further, trains are forced to reduce their speed before entering a link that has a free flow speed less than the train's current speed. In contrast, it is assumed that the train drives by the maximum acceleration that its locomotives can pull by.

If any two trains have a conflict point, then there must be signals at the point of conflict. These signals should be provided in the link file that generates the network along with the nodes file.

Code Structure

The simulator (Figure 2) starts by setting where is the stopping stations from the link files. If no stopping stations are provided, the default stopping station is set to the last station of the train's path. The main driving point of the simulator is checking whether all trains reached their destination or not. If all trains reached the destination, the simulator finishes and gives a summary file along with an optional trajectory file. If any train did not reach the destination, the simulator only runs still-not-reached-destination trains until they reach their destinations. Additionally, because the simulator is time-driven, each train will move forward at each time step, and hence, when a train reaches a group signal earlier, it will have the right of way and all other trains in the opposite direction will have to stop.

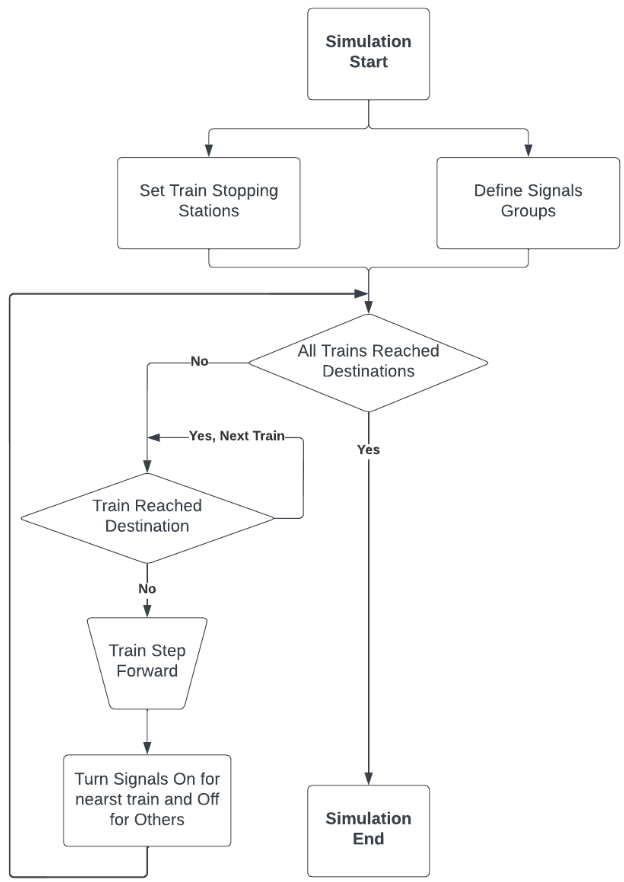


Figure 2: Simulation Flow Chart

At each time step (Figure 3), from the links each car and locomotive are at, the simulator retrieves each one's grade, curvature, and max free-flow speed. The simulator uses all the grades and curvature of all cars and locomotives to calculate an accurate resistance force. In addition, the simulator sets the maximum speed that the train can go at by the minimum speed of all links maximum free-flow speed each car or locomotive is at. This is to ensure that the train does not exceed the maximum free-flow speed on any link where part of the train is on. At the same time, the simulator (Network Class) calculates how long the entire train has as a distance from the front tip of the entire train to the next stopping station, reduced-speed point, or other trains on the network. This is to ensure that the train will reduce its speed accordingly if required.

All the gathered information is passed to the train dynamics class to calculate how much acceleration is required and calculates the train speed accordingly. This speed is used to calculate the jump distance ahead. The distance is added to the train's cumulative traveled distance along its path. Lastly, the energy consumption of the train is calculated based on the train characteristics as stated in (Wang, Ghanem, Rakha, & Du, 2021). At this point, the simulator does the same calculations for the next train until all trains are one step forward.

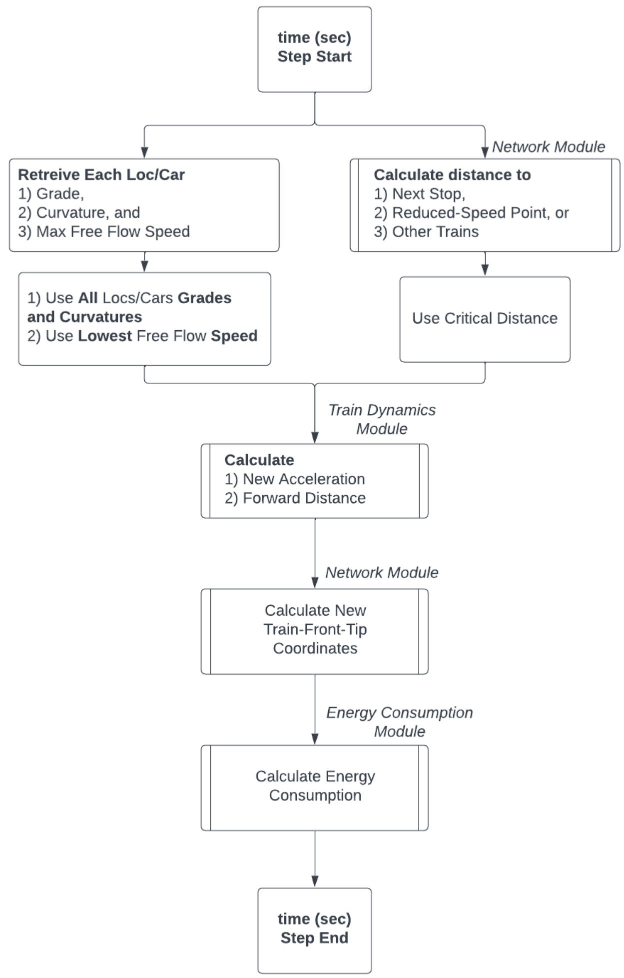


Figure 3: Time Step Simulation Calculations

Signals and Conflict Resolution

Signals are the main control points for the trains to either continue on their path or stop. These signals are not distributed randomly, but specifically placed at nodes within the network. However, it's important to note that although they are situated at nodes, each signal is oriented toward a specific direction corresponding to the link direction. This ensures that the signals are effective and efficient in managing train movements (Figure 4).

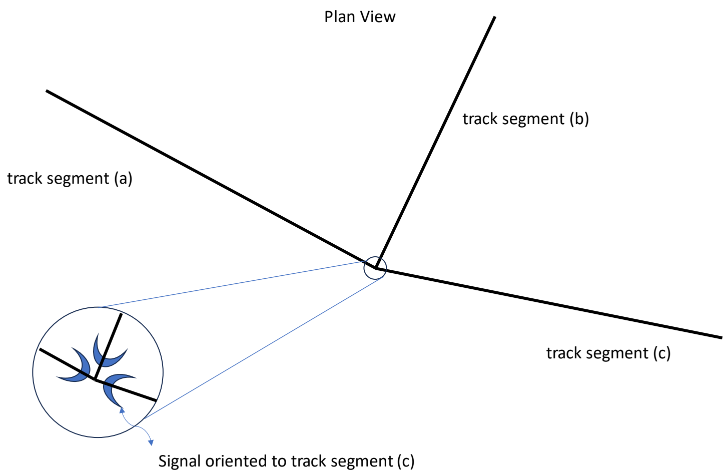


Figure 4: Signals Definition and location

These signals are combined into groups, each managed by a controller that operates on the FIFO (First In First Out) concept. This means that each controller governs which train has the priority to pass and which should stop, based on the order of their arrival at the signal node.

Conflicts in trains' paths are determined before the simulation starts. A conflict zone occurs when there's a two-way track connecting two nodes with no other link connecting these nodes. A conflict zone can also appear if the train length is larger than the total length of the links. In such cases, all the links spanned by the train are considered a conflict zone.

Two conflict scenarios can appear in the simulator, each managed by the controller using the FIFO method (Figure 5). In the first scenario (case 1), when two opposing trains approach each other on a single two-way link, the controller gives priority to the first train that reaches the conflict zone, directing the second train to stop and yield.

The second scenario (case 2) arises when there are multiple links connecting two points with two opposing trains, but the total length of these links is less than the trains' lengths. Here again, the controller prioritizes the first train that arrives at the zone, signaling the other train to wait its turn. An extension to this case (case 3) occurs when the distance between any two nodes along the paths of the two trains is less than the trains' lengths. The controller ensures the first train occupies that region, while the other waits for its turn.

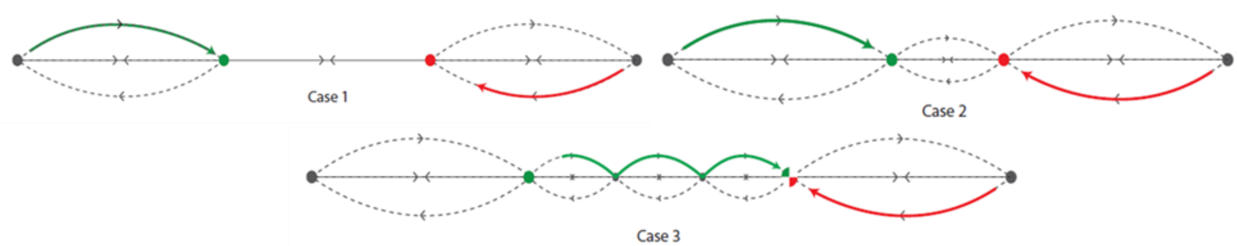


Figure 5: Conflict Zone Resolution Strategy

Running the Simulator

The simulator has 2 methods to run; 1) Command Line Interface (CMD) and 2) GUI.

1. Command Line Interface (CMD)

To run the simulator through the CMD, follow the steps below:

1. Open a shell/command line window,
2. Navigate to the installation path using

```
cd "C:\Program Files\NeTrainSim"
```

if you changed the default installation path, make sure to replace `C:\Program Files\NeTrainSim` with your installation path.

3. To request help from NeTrainSim, type the following in the shell

```
NeTrainSim -h
```

```
-h, --help, -?                         Display this help message.
-v, --version                          Displays version information.
-n, --nodes <nodesFile>                [Required] the nodes filename.
-l, --links <linksFile>                [Required] the links filename.
-t, --trains <trainsFile>              [Required] the trains filename.
-o, --output <outputLocation>          [Optional] the output folder address.
                                      Default is
                                      'C:\Users\<USERNAME>\Documents\NeTrain
                                      Sim\'.
-s, --summary <summaryFilename>        [Optional] the summary filename.
                                      Default is
                                      'trainSummary_timeStamp.txt'.
-a, --all <summarizeAllTrains>         [Optional] bool to show summary of
                                      all trains in the summary file.
                                      Default is 'false'.
-e, --export <exportTrajectoryOptions> [Optional] bool to export
                                      instantaneous trajectory.
                                      Default is 'false'.
-i, --insta <instaTrajectoryFile>      [Optional] the instantaneous
                                      trajectory filename.
                                      Default is
                                      'trainTrajectory_timeStamp.csv'.
-p, --timeStep <simulatorTimeStep>     [Optional] the simulator time step.
                                      Default is '1.0'.
```

Figure 6: Run Simulator through Command Line Interface

Example of a minimum-flag command:

```
NeTrainSim.exe -n "path\to\nodes\file" -l "path\to\links\file" -t
"path\to\trains\file"
```

2. Graphical User Interface (GUI)

The simulator can run from the GUI if you run the "NeTrainSimGUI.exe". The main menu will appear as in Figure 7. The Nodes, links, and trains files can either be created through the GUI or manually and provided to the simulator as a .dat file.

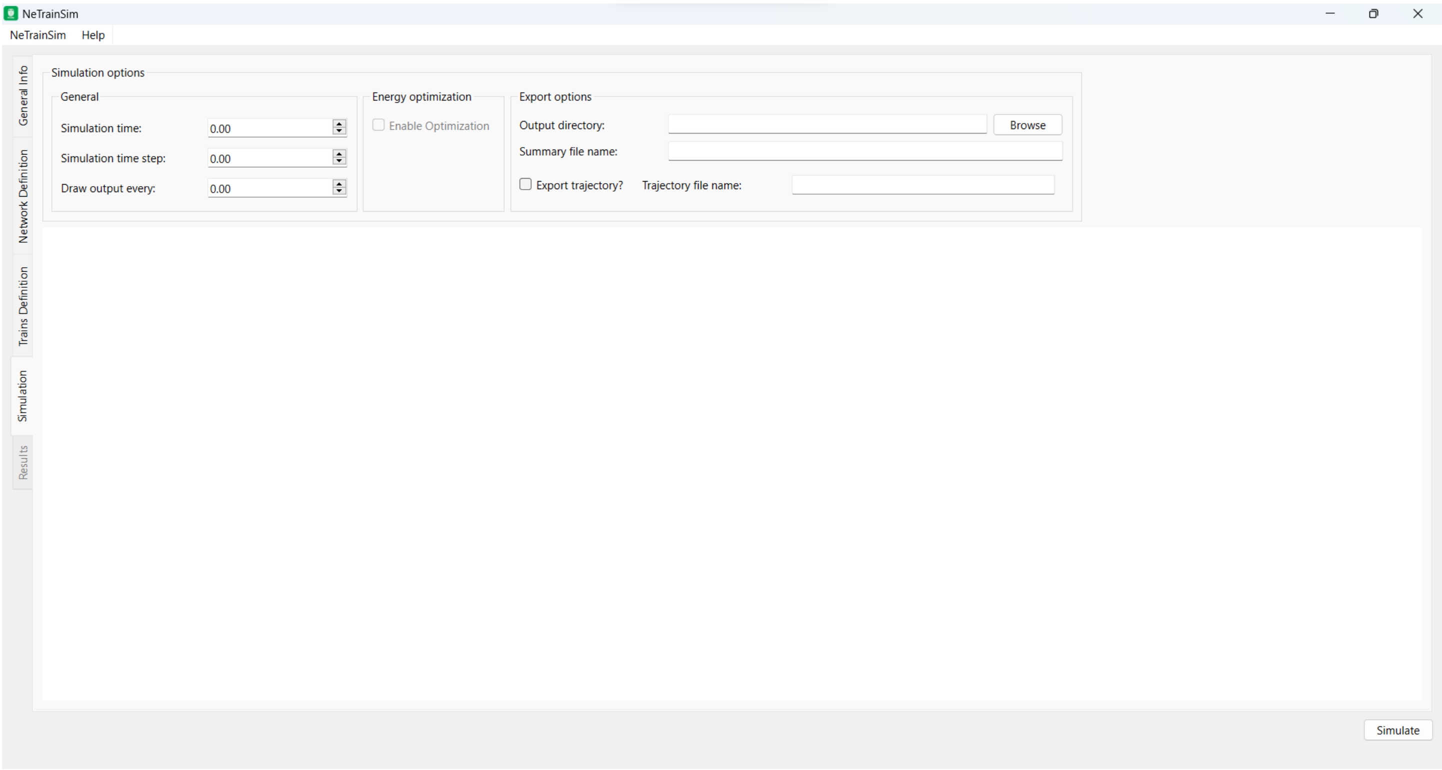
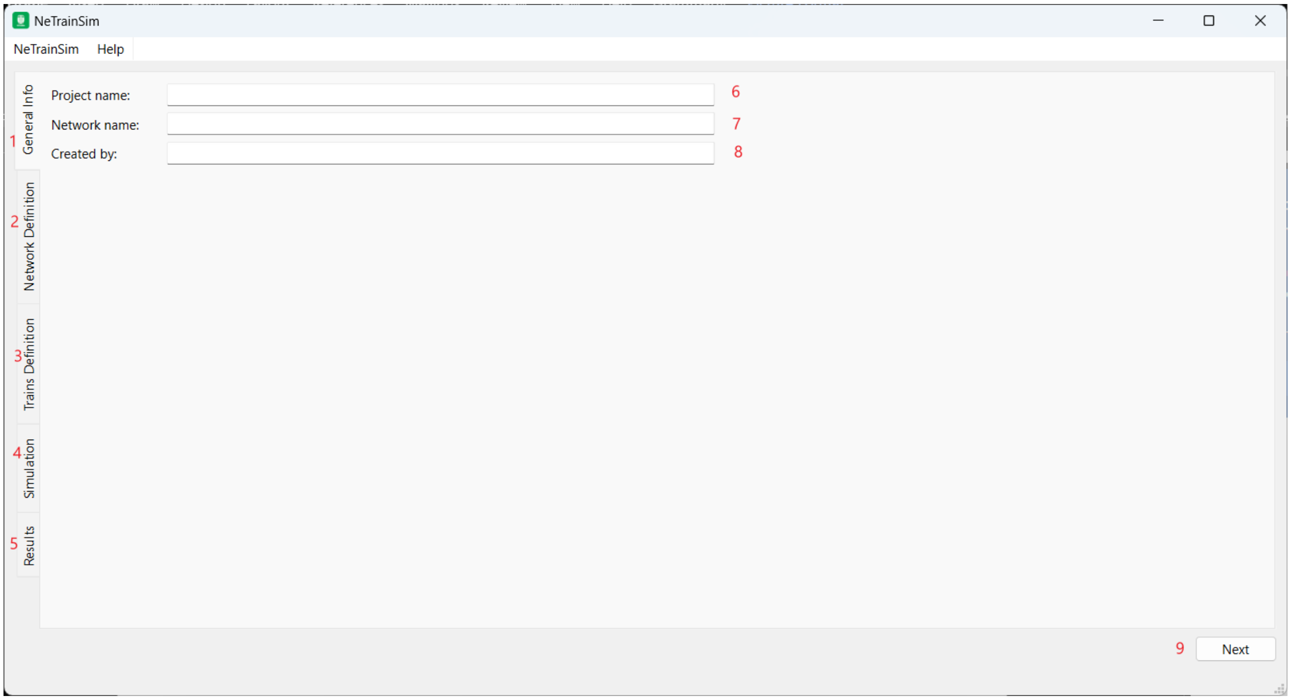
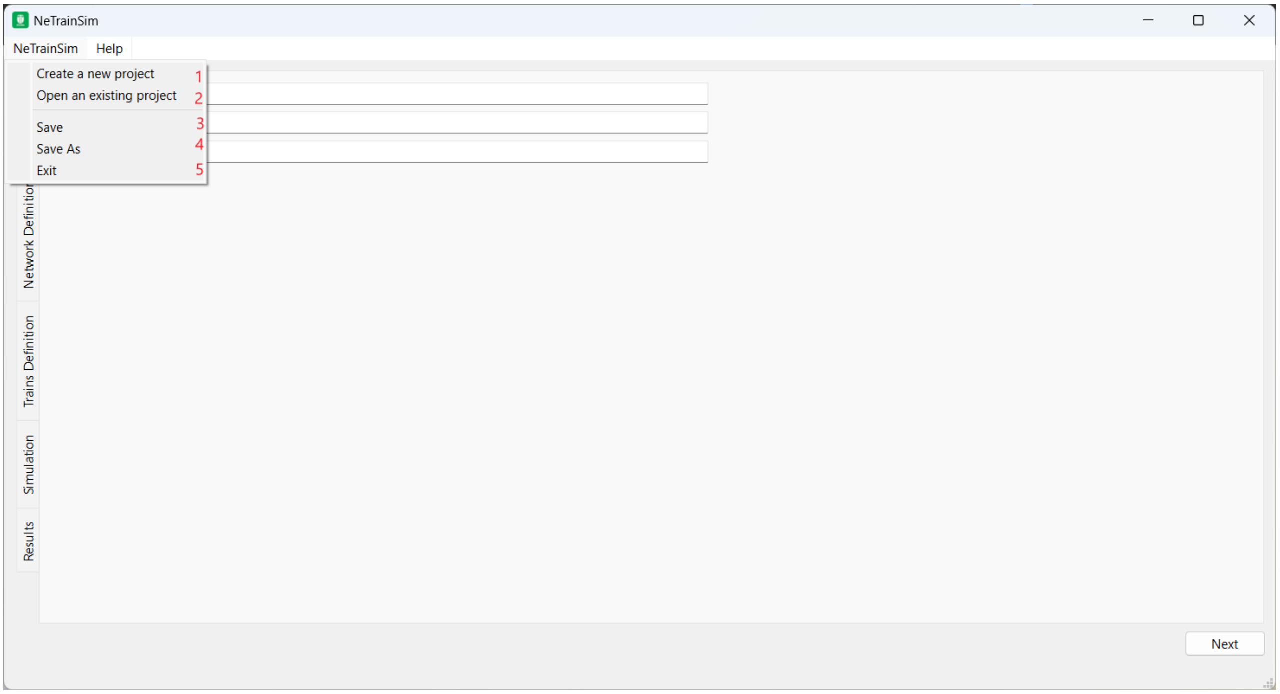


Figure 7: Simulator Main Window

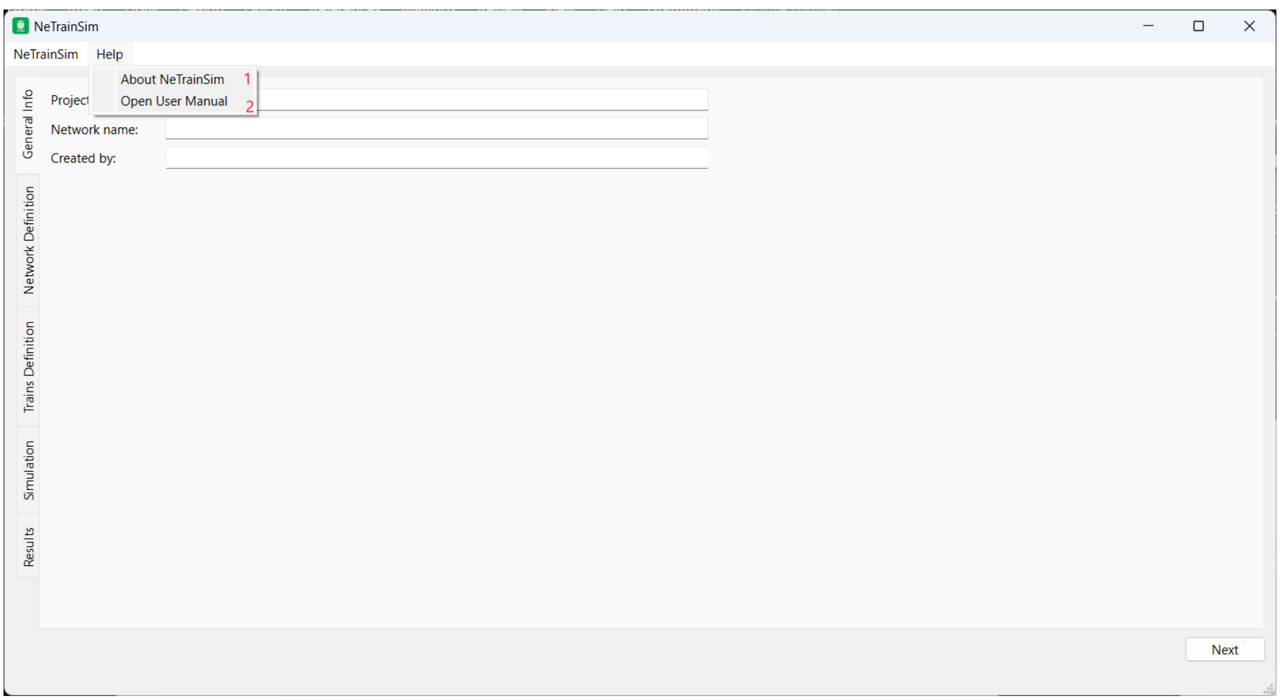
GUI Description



1. **General Information**: The information related to the project/simulation.
2. **Netowrk definition**: Tracks lengths, grades, curvature, etc.
3. **Trains Definition**: Define the trains on the network; start time, locomotives and, cars characteristics.
4. **Simulation**: Run Simulation.
5. **Results**: Visualize the results of the simulation.
6. **Project Name**: the name of the project. This will appear in the report.
7. **Network Name**: the name of the network. This will appear in the report.
8. **Created by**: the author of the simulation run.
9. **Next**: Go to next page

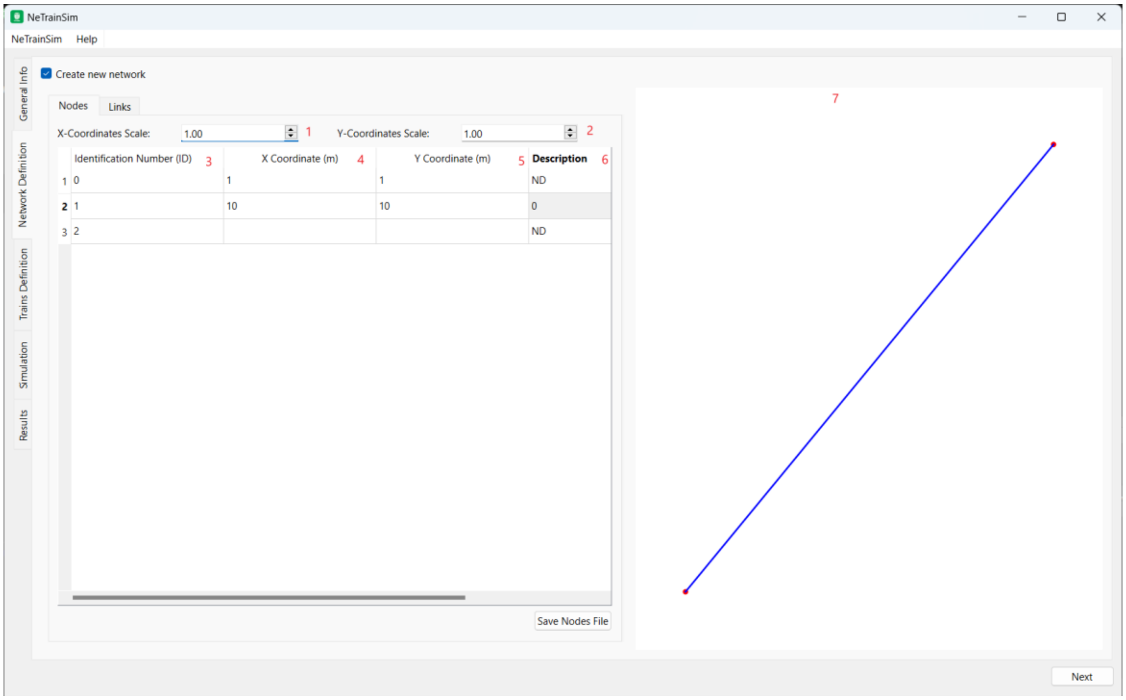


1. Create a new project and clear the current project data from the form
2. Open an early-saved project. This does not save the simulation. It only loads the input parameters.
3. Save the current project on your hard drive.
4. Save as the current project on your hard drive.
5. Exist the simulator window.



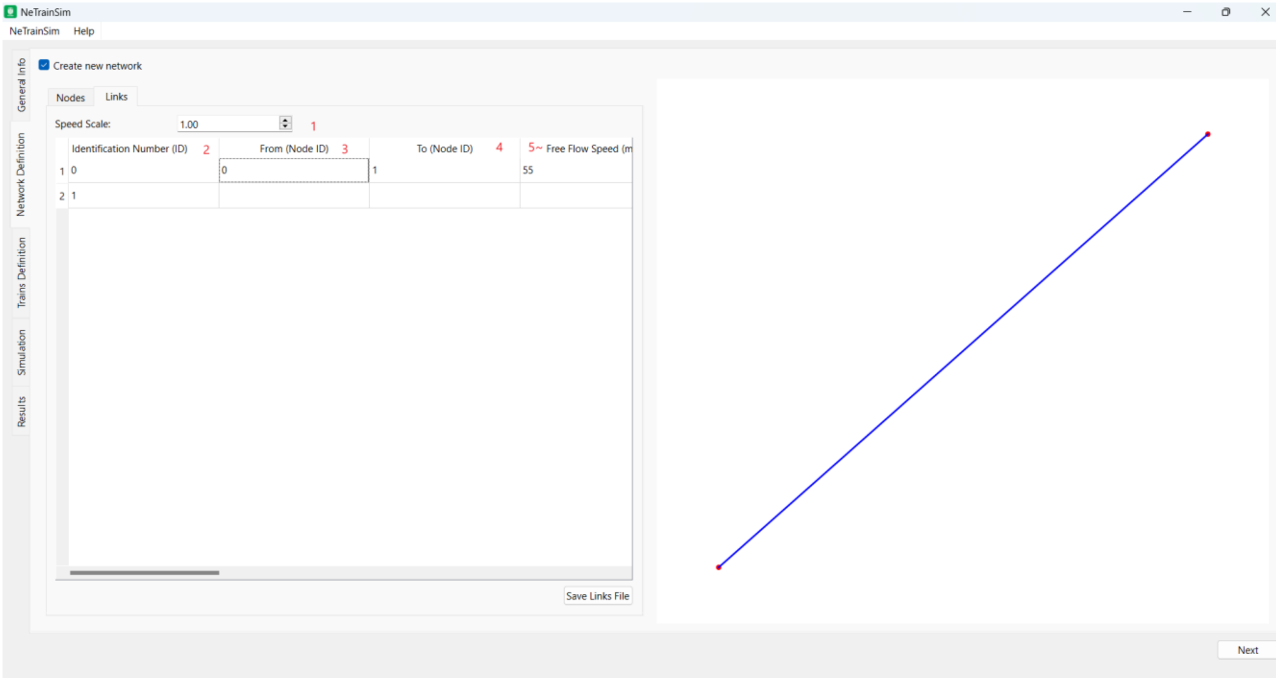
1. Open the about window.
2. Open the user manual in the default system pdf viewer.

1. **Create new network:** if not checked, you can browse for an already created network.
2. **Browse Nodes File:** Browse for the nodes DAT file from your hard drive.
3. **Browse Links File:** Browse for the links DAT file from your hard drive.
4. **Drawing Area:** This area shows how the network looks like.
5. **Next:** Go to Next Page.

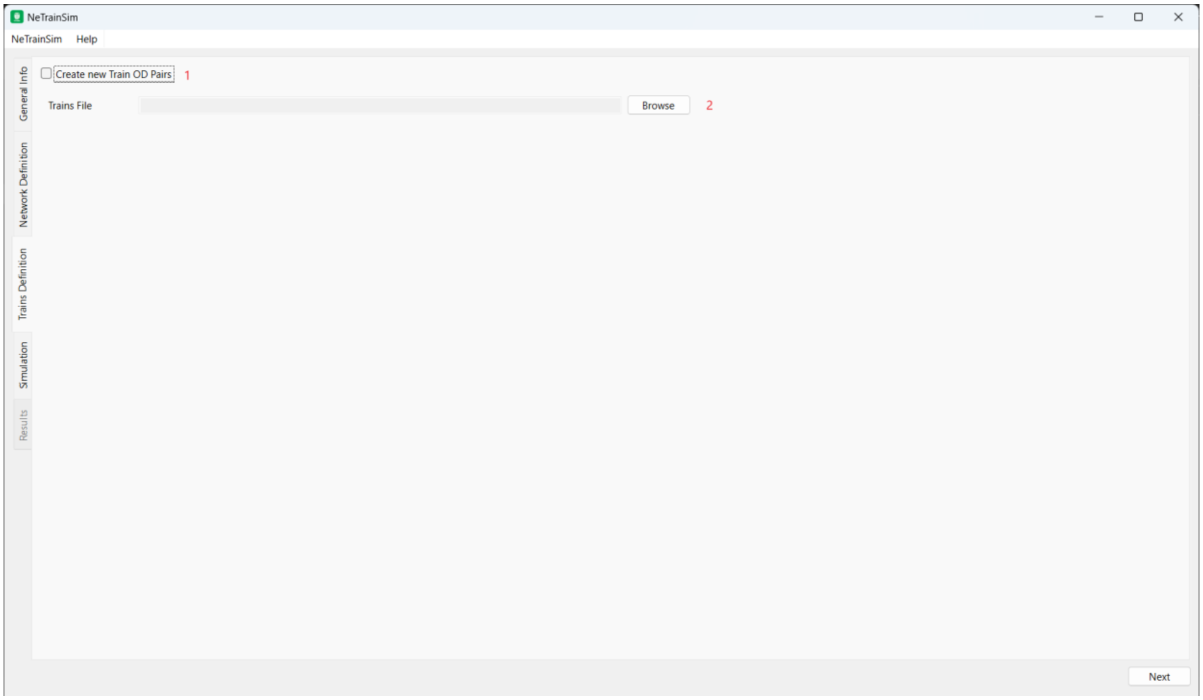


If a new network should be created, then the create new network check box should be ticked. A tab widget will appear as shown above, In the Nodes tab, you can create the network nodes and similarly you can create the network links in the links tab.

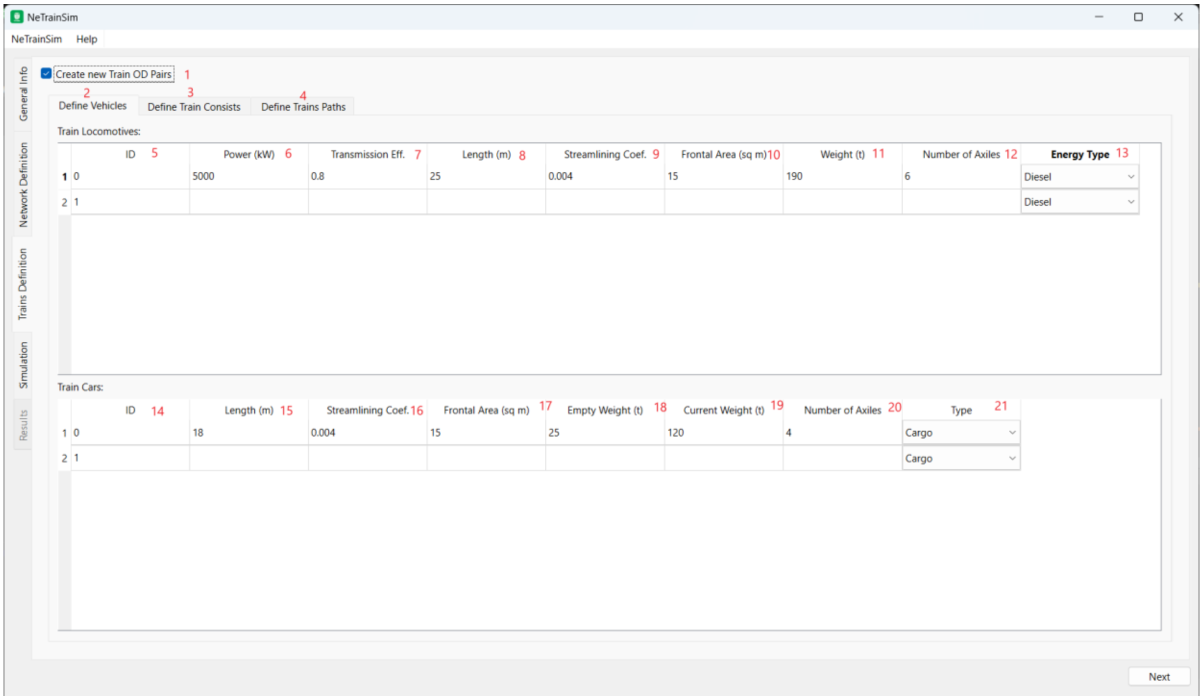
1. **X Coordinate Scale:** Is the scale applied for the x coordinates of the defined nodes.
2. **Y Coordinate Scale:** Is the scale applied for the y coordinates of the defined nodes
3. **Identification Number ID:** Is a unique ID the node should have. The link definition will need this ID to define which nodes are used for the links as start and end points.
4. **X-Coordinate:** Is the X-coordinate of each node. This should be in meters as the simulator uses the metric system in calculations. If your units are different from the metric system. You may add the coordinate as is and adjust the x-coordinate scale to convert from your unit to meters.
5. **Y-Coordinate:** Is the Y-coordinate of each node. This should be in meters as the simulator uses the metric system in calculations. If your units are different from the metric system. You may add the coordinate as is and adjust the y-coordinate scale to convert from your unit to meters.
6. **Description**: if the node has any description, you may add it here. This is an optional input to the simulator.



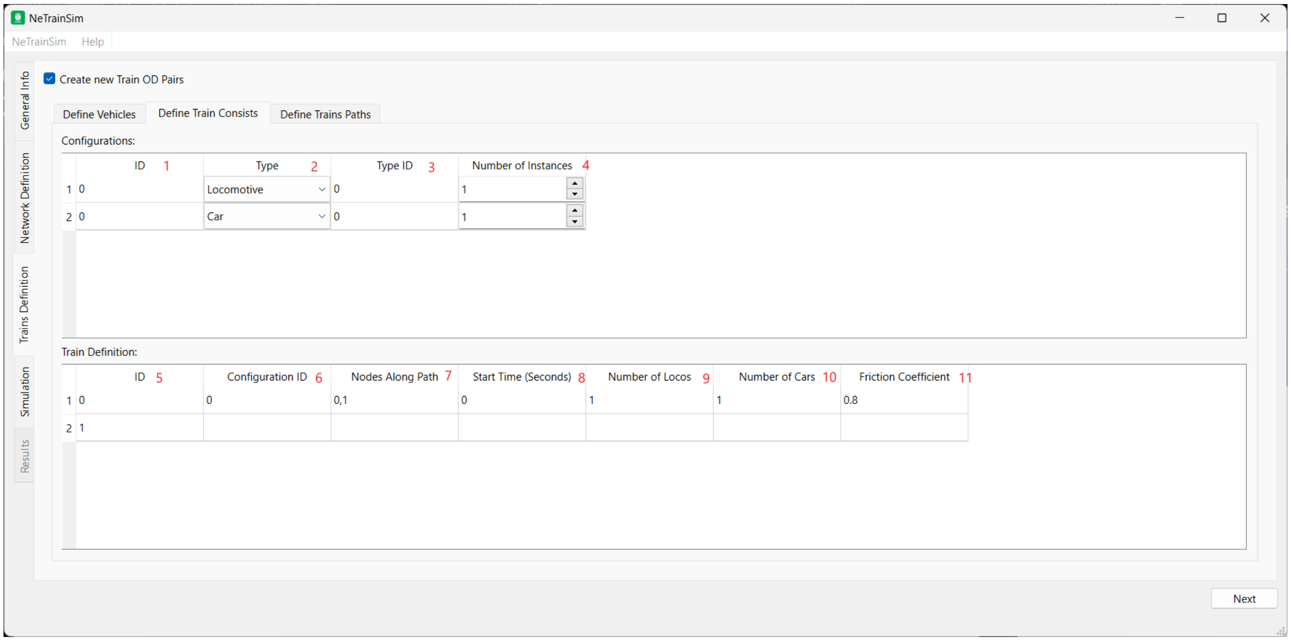
1. **Speed Scale:** Is the scale applied for the freeflow speed defined for the link.
2. **Identification Number ID:** Is a unique ID the link should have.
3. **From (Node ID):** Is the Node the link should start from. Select the Node ID as appropiate from the Nodes Table.
4. **To (Node ID):** Is the Node the link should end at. Select the Node ID as appropiate from the Nodes Table.
5. **Freeflow speed**: Is the max speed the train can achieve on this link.
6. **Traffic signal Number**: Is the ID of the signal that should be placed at the end of the link. If the link is one directional, the signal will be placed at the end node. If the link is two-directional, the signal will be placed at the two ends of the link.
7. **Grade (Percentage)**: is the grade in percentage of the link. It is from the From-To Node direction.
8. **Curvature (Percentage)**: is the percentage of curvature of the link.
9. **Is Two Directional**: select if the link should be two directional, otherwise, it will be one directional.
10. **Speed variation factor**: is the flactuation in the freeflow speed on the link.
11. **Has Catenary**: select if the link has catenary and electric trains can recharge or consume energy from the catenary.
12. **Region**: Add the region this link is considered part of. This could be a state name if the network crosses multiple states.



1. **Create new train:** Define new trains in the GUI.
2. **Browse trains file:** Browse for the trains file on the hard desk.



1. **Create new train:** If ticked, the user can define new trains in the GUI, if not, the user must browse for the file from the hard drive.
2. **Define Vehicles:** The tab where the user can define locomotives and cars.
3. **Define Train Consists:** the tab where the user can define the train consists and train characteristics.
4. **Define Trains Path:** the tabe where the user can define the start and end point of the train.
5. **Locomotive ID:** the unique identifier for the locomotive. This can only be numeric.
6. **Power (KW):** the tractive power of the locomotive in KW
7. **Transmission Effeciency:** the transimission effeciency of the locomotive. This depends on the locomotive type but in general the range is 0.95 to 0.8
8. **Length (m)**: the length of this locomotive in meters.
9. **Streamline Coefficient**: the aerodynamics streamline coeffecient divided by 10,000.0. a typical value for the leading locomotive is 24.0/10,000.0 = 0.0024 and 0.0055 for the trailing locomotive
10. **Frontal Area** (sq. m): the frontal area of the locomotive in square meters. This area contributes to the air resistance.
11. **Weight (ton):** the weight of the locomotive in metric tons.
12. **Number of axles:** the number of axles of the locomotive. A typical value is 4 or 6.
13. **Energy type:** the locomotive type. This could be Diesel, Electric, Biodiesel, diesel-hybrid, hydrogen-tender, or biodiesel hybrid.
14. **Car ID:** the unique identifier of the car.
15. **Length** (m): the length of the car.
16. **Streamline Coefficient**: the aerodynamics streamline coeffecient divided by 10,000.0. a typical value for the car is 5.0/10,000.0 = 0.005
17. **Frontal Area** (sq. m): the frontal area of the car in square meters. This area contributes to the air resistance.
18. **Empty weight (ton)**: the empty weight of the car in tons.
19. **Weight (ton):** the gross weight of the car in metric tons.
20. **Type**: The car can be of type Cargo, or a tender of diesel, biodiesel, hydrogen or a battery.

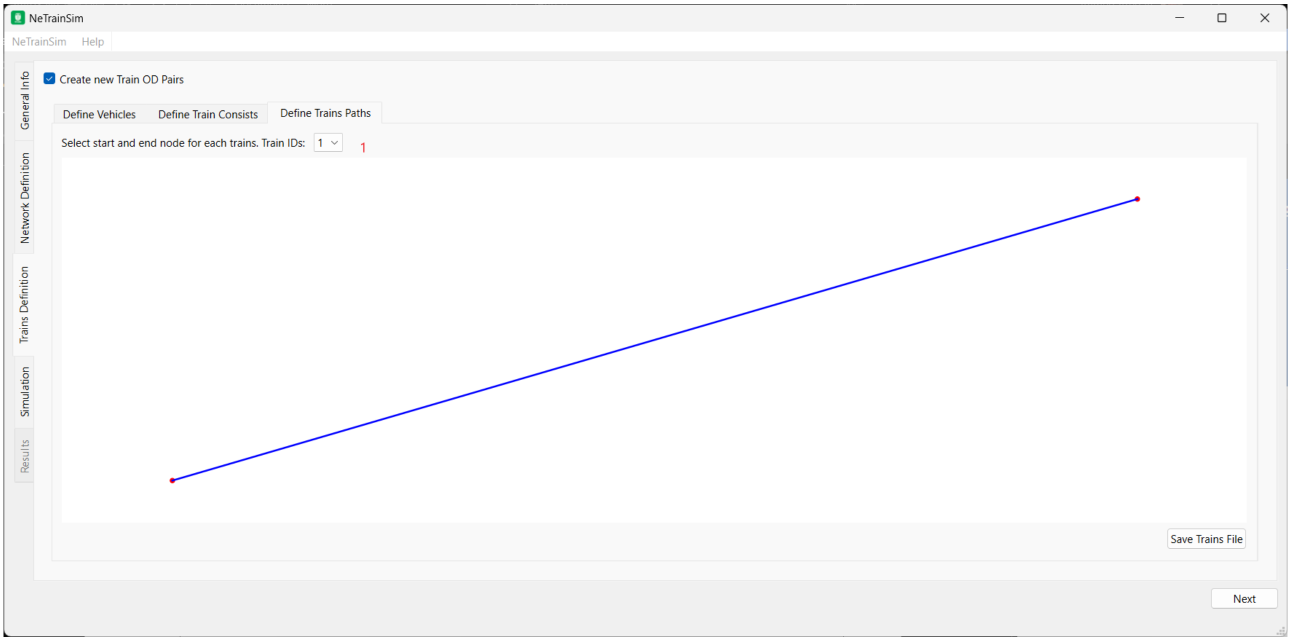


1. **Configuration/Consist ID:** The unique identifier of the train consist. This can be numeric or text. You may have multiple rows with the same ID. One ID represents a consist which may have locomotives and railcars.
2. **Type:** the type of the vehicle you are adding to the consist.
3. **Type ID:** the ID for either the locomotive or car (depends on the type in 2) and it comes from the "Define Vehicles" tab.
4. **Number of instances:** the repetition of the vehicle.
5. **Train ID**: the train unique identifier.
6. **Configuration ID**: the consist ID from the above table (**Configuration/Consist table).**
7. **Nodes Along Path:** you may either add only the start and end node IDs of the trains path or the whole path node IDs. If you provided two node IDs only, the simulator will find the shortest path between the nodes and assign the node IDs automatically. This can be left for now and populated with data from the next tab.
8. **Start Time (Seconds)**: the time the train will enter the network. Note that the current version of the simulator does not optimize the time the trains should enter the network/ depart the station.
9. **Number of Locos**: the total number of locomotives in the train.
10. **Number of cars**: the total number of cars in the train.
11. **Friction Coeffecient**: the fricition coefficient of the tracks.

The process for creating a train in the simulator is outlined as follows:

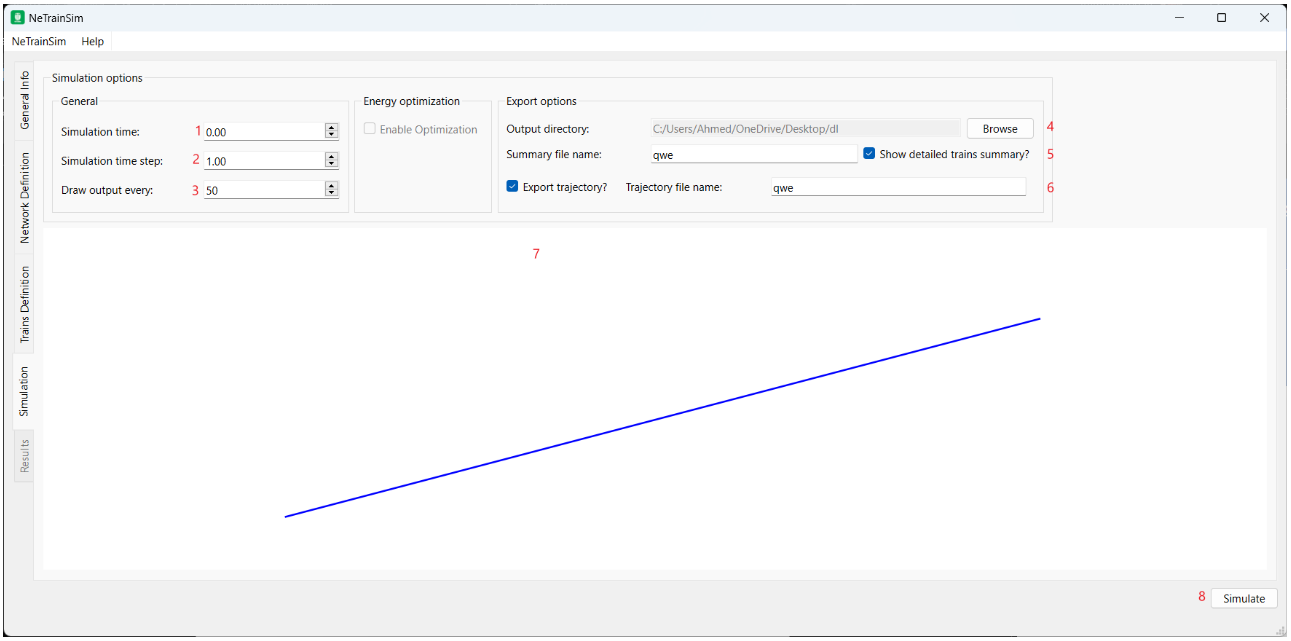
A. The simulator includes separate tables for locomotives and cars. These tables contain the definitions and characteristics for each.
B. A configuration table is available to customize how each train is assembled. This includes specifications like the number of cars and locomotives per train.
C. The details pertaining to the train itself, such as its type or category, are managed through the trains table.
D. Finally, the simulator features a tool for defining the start and end nodes for each train's route. This tool works by selecting nodes: the first node selected (via a left click) establishes the start point for the train's journey, while the second node chosen sets the end point. It's important to note that the end node must be a different node from the starting one.

By working through these steps, users can accurately design and introduce new trains into the simulator.



In this tab, you can define the trains path by either selecting the first and last node of the path or all the nodes that the train should pass by. Selecting a node is done by left clicking by the mouse on the node. Deselecting the node is done by right-clicking on the node by the mouse. Panning and zooming are also enabled in the view similar to the rest of plot views. To start selecting the nodes, the user needs to select which train the user is defining its path.

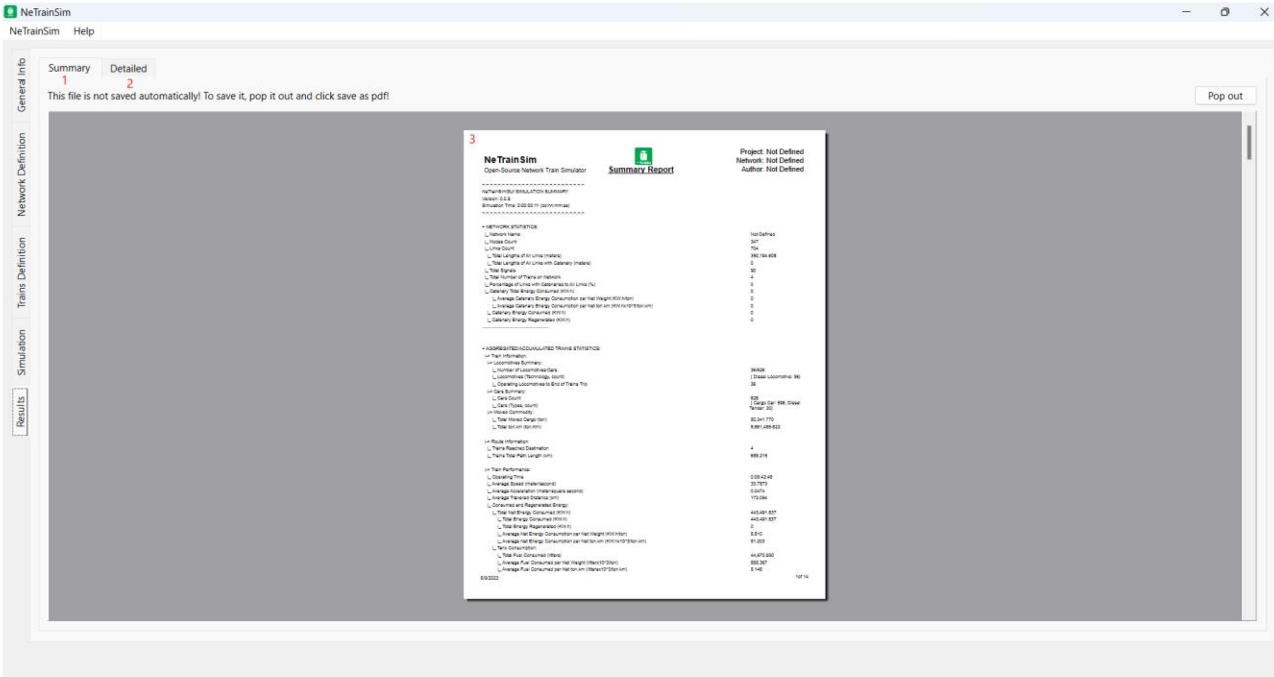
1. Train ID: the train the user is defining its path.



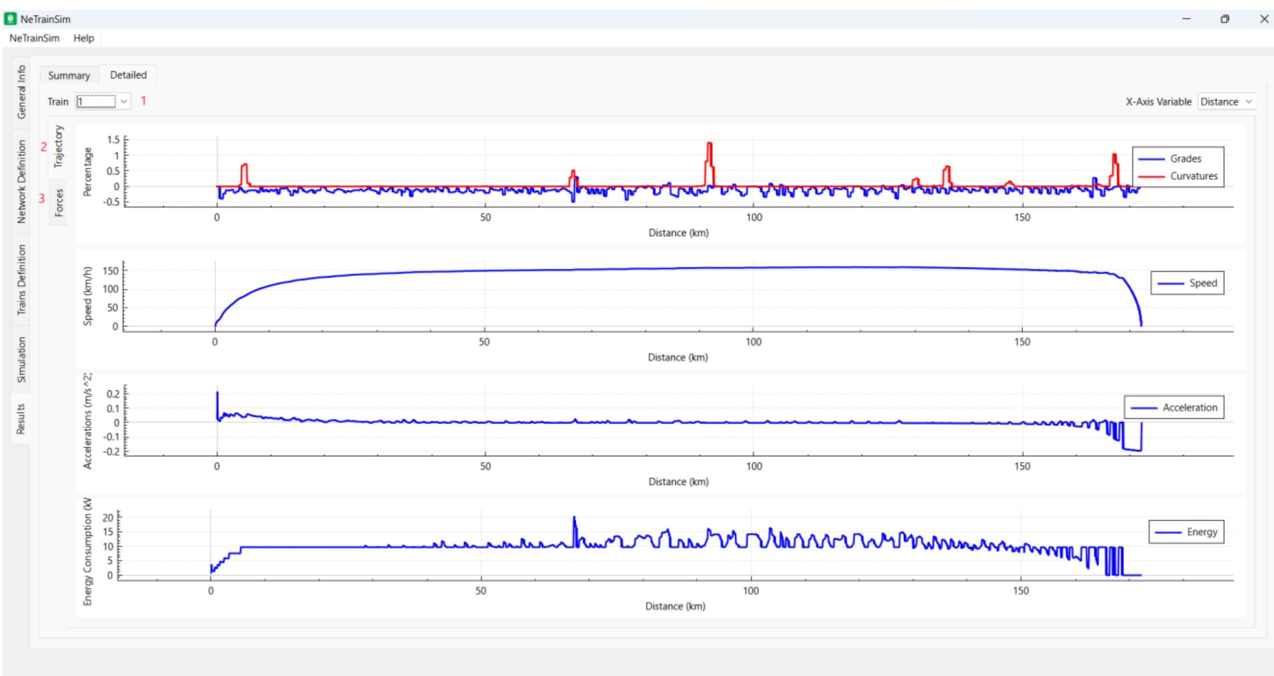
2. **Simulation End time**: If the value is modified to anything other than 0, the simulator will only simulate the movement of the trains until the specified end time of the simulation. Consequently, there is a possibility that the trains may not reach their intended destinations. To ensure the simulation continues until the trains reach their destinations, please keep the value as 0.
3. **Simulation time step**: The time step of the simulation is measured in seconds. Increasing the time step will result in a faster simulation, but it may compromise the accuracy of the results.
4. **Draw frequency**: If the value is set to 0, the trains will not be displayed on the plot view of the simulator. However, if the value is greater than 0, the plot will be updated at the specified frequency. To ensure smooth simulation performance, it is advisable to choose a higher value that does not hinder the speed of the simulator.
5. **Output directory**: the directory where the summary and trajectory files are written.
6. **Summary filename**: the name of thesummary file. The file has a a txt file extension.

    Show detailed trains summary: the default is the summary will be a network summary. Check this If the user prefers a detailed summary for each train.
7. **Export Trajectory :** Check if the user prevfers to have a step by step trajectory. The forces and trajectory information tab will not be visible if this is not checked.

    Trajector filename: the name of the trajectory file. The file extension is csv.



8. **Summary:** Shows the summary report of the entire network. Also shows a detailed sumaary of each train in the network is the 'Show Detailed Train Summary" checkbox is checked.
9. **Detailed:** shows the detailed trajectory of each train in the network. This tab is visible only if the trajectory file is exported.
10. **Report view:** the main report view.



11. **Train:** Select the train ID from the dropdown menu to visualize the trajectory and forces details.
12. **Trajectory:** This tab shows the trajectory speed, acceleration, and energy consumption of the selected train.
13. **Forces:** This tab shows the trajectory tractive forces, resistance, and net tractive forces.

Where to Know More

You may refer to any of the following papers for the scientific background of the NeTrainSim:

1. Aredah, A.; Fadhloun, K.; Rakha, H.; List, G. NeTrainSim: A Network Freight Train Simulator for Estimating Energy/Fuel Consumption. Preprints.org 2022, 2022080518. https://doi.org/10.20944/preprints202208.0518.v1

2. Aredah, Ahmed S. and Fadhloun, Karim and Rakha, Hesham and List, George and Hegazi, Mohamed and Hoffrichter, Andreas, Netrainsim: A Network Freight Train Simulator for Estimating Energy/Fuel Consumption. Available at SSRN: https://ssrn.com/abstract=4377164 or http://dx.doi.org/10.2139/ssrn.4377164

Appendix 1 (Nodes File)

The structure of the nodes file is akin to that of the INTEGRATION traffic simulator, designed for straightforward data transfer from platforms such as Excel. Values on the same line are separated by indentation.

In this system, nodes primarily serve as coordinate records, having no direct influence over the movement of trains. Instead, the locomotion of trains is managed through a network of links and signals.

Significantly, signals are exclusively placed at nodes. If a signal needs to be positioned at a specific location, a node must be present or established at that location. Moreover, each entry in the nodes dataset results in the creation of a new node within the simulator, corresponding to a unique point in the network.

However, it's important to note that nodes are connected by links, establishing the accessible routes for train movement. Any node that isn't connected by links is inaccessible, rendering it isolated and unvisited by any train movement.

Therefore, in addition to accurately mapping and recording nodes, careful construction and representation of links connecting these nodes is paramount. This ensures the creation of an effective and realistic simulation of train traffic, where signals placed at nodes guide train movement along the interconnected network of links.

Table 1: Nodes File Definition and Schema

Line No	Variable Name	Description
1	File Description	File description. It could be used for quality control
2	Number of nodes	Number of nodes in this file
	X-value scale	For the scale of the coordinate x-value, you can use different units other than the metric system, but you must change this value to convert from your units to the metric system (meter).
	Y-value scale	For the scale of the coordinate y-value, you can use different units other than the metric system, but you must change this value to convert from your units to the metric system (meters)
3	Node ID	A unique ID for this Node. It should be numeric and starts from 1 to inf
	X	The x value of the Node Coordinate
	Y	The y value of the Node Coordinate
	Description (Optional)	Node Description

Appendix 2 (Links File)

The structure of the links file echoes that of the INTEGRATION traffic simulator and is designed for easy data transfer from applications like Excel, with values on the same line separated by indentation.

In this model, each link signifies a single track segment. Therefore, every entry in the links dataset results in the creation of a new link or track segment within the system. This is particularly important when there are multiple tracks between two nodes, as each track should be represented by a unique link in the model.

The directionality of a link is determined by the 'From node' to the 'To node' parameters. For one-way links, the direction is established from the 'From node' towards the 'To node', accurately mirroring the flow of trains on these specific track segments.

Moreover, the presence of links connecting nodes is crucial for the movement of trains. Nodes without links connecting them are inaccessible from any direction, making links indispensable for train mobility across the network.

Interestingly, while the dataset may contain specific lengths for each link, these lengths are not used directly by the simulator. Instead, the simulator recalculates the length of each link based on the coordinates of the start and end nodes ('From node' and 'To node'). Nevertheless, it's recommended to include these lengths in the dataset, even though they are not directly used, for completeness and potential reference.

Thus, careful mapping and accurate representation of these links is vital for the effective management and precise control of train traffic within the network.

Table 2: Links File Definition and Schema

Line No	Variable Name	Description
1	File description	File Description. It could be used for quality control
2	Number of links	Number of links in this file
	Length scale	For the scale of the length, you can use different units other than the metric system, but you must change this value to convert from your units to the metric system (meters)
	Free speed scale	Free flow speed scale, you can use different units other than the metric system, but you must change this value to convert from your units to the metric system (m/s)
3	Link ID	A unique ID for this link. It should be numeric and starts from 1 to inf
	From node	The first node ID of this link
	To node	The last node ID of this link
	Length	The length of this link
	Free speed	The max speed a train can drive by on this link
	Signal Number	The Single ID at the end of this link; if it is 2 directional links, a signal will be added at the two ends. The signal IDs are unique for every signal.
	Signal Placed at	The location of the signal to be placed at. If the link is 2 way (check direction variable) and you would like to add the signal at only a specific end, write the node number of the link here. This node number should be equal to either the ‘From node’ or ‘To Node’.
	Grade	The grade of the link in percentage. It does not include the percentage sign
	Curvature	The curvature of the link
	Direction	The direction of the link, if one way, insert 1, if two ways, insert 2
	Speed variation	The speed variation of the link is not yet implemented in this version
	Has Catenary	0: If the link does not have a catenary. 1: O.W. The catenary provides electricity/power to only suitable locomotives. Diesel-Electric and BioDiesel-Electric locomotives do not use catenaries.
	Region (Optional)	The region this link belongs to (Optional)

Appendix 3 (Trains File)

The structure of the trains file echoes that of the INTEGRATION traffic simulator, designed for straightforward data transfer from applications like Excel. Values on the same line are separated by indentation.

The mechanism for positioning locomotives within trains in the simulator operates as follows:

a. If only a single locomotive is provided, it is placed at the front of the train, acting as the leader.
b. In instances where less than seven locomotives are provided, they are situated at the beginning and the middle of the train, with a priority given to the start of the train.
c. If more than seven locomotives are provided, they are evenly distributed along the train, positioned at the beginning, middle, and end. In this arrangement, precedence is given to the start of the train, followed by the middle, and finally the end.

These placement rules offer flexibility in how trains are configured, enabling a variety of train setups to be modeled accurately within the simulator.

Table 3: Trains File Definition and Schema

Line No	Separator to previous	Variable Name	Description
1	New line	File description	The file Description. IT could be used for quality control
2	New line	Number of trains	The number of trains in this file
3	Tab	Train ID	A unique numeric ID for each train. It could be of the type "String"
		Train path by node ID *	The train Path as a list of network nodes ID (it should be separated by commas and no spacings)
		Train start time	Time offset (measured from simulation start time) for the train to enter the network.
		Transmission efficiency	The train transmission efficiency
		Number of locomotives	The number of locomotives in the train
		Number of cars	The number of cars
		Friction (mu)	The friction coefficient
		Define Locomotive-Type Groups **	
		Type count	The number of locomotives in this group
	Comma	Max power (KW)	The max power of locomotives in this group
		Number of loc axles	The number of axles in locomotives in this group
		Locomotive K value	The streamlining coefficient for the locomotives in this group
		Locomotives frontal area	The typical frontal area of this group of locomotives
		Locomotives length	The typical length of this group of locomotives
		Locomotives weight	The typical weight of this group of locomotives
		Locomotives type	The typical type of locomotives in this group (integer number from the following list): 0: diesel, 1: Electric, 2: Biodiesel, 3: Diesel-Electric, 4: Diesel Hybrid, 5: Hydrogen Hybrid, 6: Biodiesel Hybrid.
	Tab	**Define Car-Type Groups ****	
		Type count	The number of cars in this group
	Comma	Number of car axles	The number of axles in cars in this group
		Car K value	The streamlining coefficient for the cars in this group
		Car frontal area	The typical frontal area of this group of cars
		Car length	The typical length of this group of cars

Total car weight	The typical full weight of this group of cars
Car weight	The typical empty weight of this group of cars
Car Type	The typical car type of this group of cars (integer number from the following list): 0: Cargo Car, 1: Diesel Tank, 2: Battery Tender, 3: Hydrogen Tender, 4: Biodiesel Tank

* Train Path is a vector.
Vector example: 1,2,3,4,5
** Create as many groups as the train has. Groups are separated by semi-comma (;) separators.