TransitLab SimMETRO

Mojtaba Yousefi

CONTENTS:

1	Introdu 1.1 Pi		1
		3	1
	1.3 L	icense	2
2	Setup G	Guide	3
3	Getting	Started	5
	_		5
	3.2 U	Inderstanding the Inputs and Config	6
4	Usage		7
		ϵ	7
	4.2 A	Advanced Usage	7
5	Configu		9
			9
	3.2 D	Default Configuration	9
6	Archite		_
		Components Overview 1 .1.1 Infrastructure Module 1	
	٠.	.1.2 Passenger Module	
	6.	.1.3 Simulation Module	9
		.1.4 Train Module	
	6.2 D	Oata Flow	5
7	Develop		
		Development Environment Setup	
		Build Instructions	
8		Dution Guidelines Now to Contribute	
		Community Guidelines	-
9	Faqs	3	1
10		eshooting 3	3
		Common Issues	
	10.2 G	Setting Help	3

	Changelog 11.1 Version History	35 35
	Appendix 12.1 Glossary 12.2 Further Reading	
	Indices and tables	39
Bil	bliography	41
Py	thon Module Index	43
Ind	dex	45

ONE

INTRODUCTION

1.1 Project Overview

TransitLab SimMETRO is a sophisticated simulation tool developed upon decades of research (Koutsopoulos and Wang [KW07] and Zhou [Zho22]) and development to address the operational challenges faced by heavy rail systems in major metropolitan areas, particularly during peak periods when demand is highest. This tool provide microscopic simulation model for heavy rail systems, enabling detailed analysis of train behaviors, signaling systems (fixed-block and moving block), and interactions between trains and passengers. By employing TransitLab SimMETRO, researchers can evaluate various operating strategies such as skip-stop, station consolidation, and dwell control to mitigate capacity bottlenecks.

The simulation framework is built using Python in a unix environment, ensuring robust performance and flexibility. The model's accuracy and reliability can be further enhanced through a calibration process using data from Operational Control Systems [WK11]. Numerous visualization tools allow users to analyze simulation outputs.

Numerous case studies (see Zhou *et al.* [ZKS20] and Zhou and Koutsopoulos [ZK22]) demonstrate how MIT Transit-Lab SimMETRO has helped agencies address congestion, improve service reliability, and support long-term planning decisions.

1.2 Features

- 1. Detailed modeling of signal systems, including both fixed-block and moving-block, with precise train movement simulations at the individual vehicle level.
- 2. Accurate dwell time modeling at stations to reflect real-world scenarios.
- 3. Realistic passenger boarding and alighting behaviors.
- 4. Implementation of headway-based dispatching to model stochasticities in rail operations.
- 5. Comprehensive simulation outputs for thorough analysis.
- 6. Multiple visualization tools available for detailed examination of simulation data.
- 7. Customizable train movement models to suit various rail systems.

1.3 License

The project License is yet to be determined. Please contact the development team for more information.

TWO

SETUP GUIDE

This project uses Pipenv for dependency management and packaging. You can learn about the dependencies looking at the Pipfile at the project root directory. Assuming you have a working version of pip, you can install Pipenv using pip:

\$ pip install --user pipenv

Pipenv will take care of installing dependencies and creating the virtual environment. Run the following command from the project root directory (where the Pipfile and Pipfile lock are located):

\$ pipenv install

You can activate the virtual environment using:

\$ pipenv shell

Depending on Python versions available on your system you may get an error like:

```
Error: the specified Python version (3.8) is not available on your system.
```

It is recommended to install the required Python version using pyenv. You can install pyenv using the instructions at https://github.com/pyenv/pyenv?tab=readme-ov-file#installation.

You may need to reactivate your shell after installing pyenv/pipenv to make sure they are added to the PATH.

Once you have pyenv installed, Pipenv will ask you if you like to use pyenv to install the required Python version. You can say yes and Pipenv will install the required Python version in the virtual environment.

THREE

GETTING STARTED

3.1 Quick Start

- 1. Prepare the input files:
 - Ensure you have the following files in the *inputs* directory:
 - infra.json: Infrastructure data
 - slow_zones.json: Slow zone definitions
 - demand/odx_demand.csv: Passenger demand data
 - schedules/empirical_schedule.json: Train schedule data
- 2. Configure the simulation:
 - Open the load-balance/config.yaml file
 - · Adjust the following parameters as needed:
 - $\ simulation.number_of_replications \\$
 - simulation.start_time_of_day
 - simulation.end_time_of_day
 - demand_level
 - station
 - short_turning
 - logger.should_log_trajectories
 - logger.log_interval
- 3. Run the simulation:

```
python -m transit_lab_simmetro.simulation_runner.runner
```

- 4. Check the output:
 - Look for the generated log files in the specified log folder (default: log_folder_path)
 - You'll find CSV and JSON files with various simulation data

3.2 Understanding the Inputs and Config

- 1. Infrastructure (infra.json):
 - Defines blocks, stations, and path information for Northbound and Southbound directions
- 2. Slow Zones (slow_zones.json):
 - Specifies areas with reduced speed limits
- 3. Demand Data (odx_demand.csv):
 - Contains passenger arrival rates at 15-minute intervals for weekdays and non-weekdays for each OD pair
- 4. Schedule Data (empirical_schedule.json):
 - Defines train dispatch times and routes
- 5. Configuration (config.yaml):
 - simulation: Set replication count, start and end times
 - demand_level: Adjust overall passenger demand. Used as a multiplier to uniformly scale demand.
 - station: Specify holding station (e.g., "O-Hare" or "Clark/Lake")
 - short_turning: Set short turning location ("UIC" or "Western")
 - logger: Configure logging options

By adjusting these inputs and configurations, you can simulate various scenarios and analyze the rail system's performance under different conditions.

FOUR

USAGE

4.1 Basic Usage

Explain how to use the project with simple examples.

4.2 Advanced Usage

Dive into more complex use cases and features.

8 Chapter 4. Usage

FIVE

CONFIGURATION

5.1 Configuration Options

Document configuration options, environment variables, etc.

5.2 Default Configuration

Explain the default setup and how to customize it.

SIX

ARCHITECTURE

6.1 Components Overview

6.1.1 Infrastructure Module

Infrastructure components for transit system simulation.

This module provides essential classes and types for modeling the physical infrastructure of a transit system, including blocks, paths, stations, and control systems.

Classes:

Block: Represents a fixed section of track. Terminal: Special block type for end-of-line stations. MovingBlock: Implements moving block signaling system. Path: Defines a sequence of blocks forming a route. Station: Represents a passenger boarding and alighting point. SignalControlCenter: Manages traditional fixed-block signaling. MovingBlockControl: Manages moving block signaling system.

Types:

BlockType: Union type for different block implementations.

The module supports both fixed-block and moving-block signaling systems, allowing for flexible and realistic transit network modeling.

```
add_observer: SignalControlCenter) \rightarrow None
     block_logger: BlockActivationLogger | None = None
     current_speed_code(requesting_train: Train) → float
     deactivate(exiting\_train: Train) \rightarrow None
     property id_of_last_train: str | None
     property is_occupied: bool
     is\_occupied\_by(train: Train) \rightarrow bool
     property last_train_visit_time: float
     notify_observers() \rightarrow None
     remove_communicated_speed_code(block\ id: str) \rightarrow None
     set_slow_zone(speed_limit: float)
class
transit_lab_simmetro.simulation_engine.infrastructure.__init__.DispatchingBlockDecorator
     Bases: Block
     __init__(block: Block, path: Path, dispatch_margin: float = 120, upstream_blocks: List[str] = [])
     activate(entering\_train: Train) \rightarrow None
     is\_it\_clear\_to\_dispatch() \rightarrow bool
     property last_train_visit_time: float
     ready_to_dispatch(train: Train) → bool
class transit_lab_simmetro.simulation_engine.infrastructure.__init__.
DispatchingMovingBlockDecorator
     Bases: MovingBlock
     __init__(block: MovingBlock, path: Path, dispatch_margin: float = 120, upstream_blocks: List[str] = [])
     activate(entering\_train: Train) \rightarrow None
     is_it_clear_to_dispatch() → bool
     property last_train_visit_time: float
     ready_to_dispatch(train: Train) \rightarrow bool
class transit_lab_simmetro.simulation_engine.infrastructure.__init__.MovingBlock
     Bases: AbstractBlock
     __init__(*args, **kwargs)
     activate(entering\_train: Train) \rightarrow None
     property civil_speed_limit: float
     current_speed_code(requesting_train: Train) → float
```

```
property current_train: Train | None
     deactivate(exiting\_train: Train) \rightarrow None
     property is_occupied: bool
     is_occupied_by(train: Train) \rightarrow bool
     next_train(requesting_train: Train) → Train | None
     register\_moving\_block\_control\_center(moving\_block\_control) \rightarrow None
     property sorted_current_train_list: List[Train]
class transit_lab_simmetro.simulation_engine.infrastructure.__init__.MovingBlockControl
     Bases: object
     __init__(blocks: List[MovingBlock], safety_margin: float = 200) \rightarrow None
     get_distance_to_next_train(asking_train: Train) → float
     get\_speed\_code(asking\_train: Train, asking\_block: MovingBlock) \rightarrow float
     needed_braking_distance(asking_train: Train) → float
class transit_lab_simmetro.simulation_engine.infrastructure.__init__.
OffScanSymptomaticBlockDecorator
     Bases: Block
     __init__(block: Block, path: Path, offscan_probability: float = 0.01)
     activate(entering\_train: Train) \rightarrow None
     current_speed_code(requesting_train: Train) → float
     set\_unsymptomatic() \rightarrow None
class transit_lab_simmetro.simulation_engine.infrastructure.__init__.Path
     Bases: object
     __init__(direction: str, blocks: List[MovingBlock | Block], slow_zones: List[SlowZone] = [])
     copy()
     get_all_stops_ahead(block_index: int) \rightarrow List[Station]
     get_all_stops_ahead\_which_are\_served(block\_index: int) \rightarrow List[Station]
     get_block_by_id(block_id: str) \rightarrow MovingBlock \mid Block
     get_block_index_by_id(block_id: str) \rightarrow int
     get\_distance\_to\_the\_next\_station(current\_block\_index: int, current\_location\_on\_block: float) \rightarrow
     get_next_station(current\_block\_index: int) \rightarrow Station \mid None
     get_next_train(current\_block\_index: int) \rightarrow Train \mid None
     get\_previous\_train(current\_block\_index: int) \rightarrow Train \mid None
```

```
get\_total\_length() \rightarrow float
     get_total_travelled_distance(current_block_index: int, distance_travelled_in_current_block: float)
                                          \rightarrow float
     is_inspected() \rightarrow bool
     \textbf{is\_short\_turn()} \rightarrow bool
     is_short_turned_at_this_station(station: Station) → bool
     make_dispatching_block(block id: str, dispatch margin: float = 120, upstream blocks: List[str] = [])
     make_offscan_block(block_index: int, offscan_probability: float)
     path_distance_alignment() → None
     set_block_slow_zone(block_id: str, speed_limit: float)
     short_turn(train: Train)
class transit_lab_simmetro.simulation_engine.infrastructure.__init__.SignalControlCenter
     Bases: object
     __init__(blocks: List/Block])
     get_block_by_id(block\ id:\ str) \rightarrow Block \mid None
     \textbf{restore\_speed\_codes\_to\_upstream\_blocks}(\textit{block}: Block) \rightarrow None
     send\_speed\_codes\_to\_upstream\_blocks(block: Block) \rightarrow None
     update(block: Block) \rightarrow None
class transit_lab_simmetro.simulation_engine.infrastructure.__init__.SlowZone
     Bases: object
     __init__(block_id: str, reduced_speed_limit: float)
     classmethod from_dict(data: dict) \rightarrow SlowZone
     to_dict() \rightarrow dict
class transit_lab_simmetro.simulation_engine.infrastructure.__init__.Station
     Bases: object
     __init__(station_name: str, location_relative_to_block: float, direction: str, arrival_rates: ArrivalRate)
     board_passengers_based_on_destiations_and_probability(train_capacity: int,
                                                                          served destinations: List[str],
                                                                          probability_of_boarding_any_train:
                                                                         float) \rightarrow List[Passenger]
     board_passengers_onto_train(train_capacity: int, served_destinations: List[str]) \rightarrow List[Passenger]
     generate_and_add_passengers(train: Train, holding time: float = 0) \rightarrow None
     get_dwell_time(door_metrics: List[Tuple(int, int, int)]) \rightarrow float
     property last_train_visit_time: float
```

```
simulation: Simulation
station_logger: StationLogger | None = None

class transit_lab_simmetro.simulation_engine.infrastructure.__init__.Terminal
    Bases: Block
    __init__()
    activate(entering_train: Train) \rightarrow None
    current_speed_code(requesting_train: Train) \rightarrow float
```

6.1.2 Passenger Module

This module imports and exposes key components for passenger modeling:

Classes:

Passenger: Represents an individual transit user. ArrivalRate: Models passenger arrival patterns at stations. PassengerGenerator: Creates passenger objects based on arrival rates.

These components form the foundation for simulating passenger behavior within the transit system.

```
class transit_lab_simmetro.simulation_engine.passenger.__init__.ArrivalRate
    Bases: object
```

Calculates the arrival rate and flow probability for a given station, time, and direction in a transit system.

The *get_lambda_for_station* method calculates the total arrival rate for a given station, time, and direction by summing the smoothed arrival rates for all destination stops from that station.

The *get_p_for_station* method calculates the flow probability for a given station, time, and direction by dividing the through-flow rate (sum of arrival rates to all downstream stations) by the total arrival rate at that station.

The get_a_i method calculates the cumulative flow probability from a start station to a critical station in a given direction. This is used to compute the get_lambda_bar method, which calculates the average arrival rate between a start station and a critical station in a given direction.

Calculates the probability (a_i) of passengers arriving at each station between the start and critical stations in a given direction.

The probability is calculated by: 1. Sorting the stations between the start and critical stations in the given direction. 2. Iterating through the stations between the start and critical stations. 3. For each station, calculating the probability (p_k) of passengers passing through that station. 4. Multiplying the probabilities (p_k) to get the overall probability (a_i) .

Parameters

- **current_hour** (*float*) The current hour of the day.
- **current_weekday** (*bool*) Whether the current day is a weekday or not.
- **start_station** (*str*) The start station.
- **critical_station** (*str*) The critical station.
- **direction** (*str*) The direction of travel.

Returns

The probability (a_i) of passengers arriving at each station between the start and critical stations.

Return type

float

$get_all_destination_stops_for_origin(origin_stop: str) \rightarrow List[str]$

Gets all the destination stops that can be reached from the given origin stop.

Parameters

origin_stop (str) – The origin stop to get the destination stops for.

Returns

A list of all the destination stops that can be reached from the given origin stop.

Return type

List[str]

$get_all_destination_stops_for_origin_and_direction(origin_stop, direction) \rightarrow List[str]$

Gets all the destination stops for a given origin stop and direction.

Parameters

- **origin_stop** (*str*) The origin stop to get the destination stops for.
- **direction** (*str*) The direction of travel.

Returns

A list of all the destination stops that can be reached from the given origin stop in the specified direction.

Return type

List[str]

$\textbf{get_all_previous_stops_for_station_and_direction}(\textit{station: str}, \textit{direction: str}) \rightarrow List[str]$

Gets all the previous stops for a given station and direction.

Parameters

- **station** (*str*) The station to get the previous stops for.
- **direction** (*str*) The direction of travel.

Returns

A list of all the previous stops that come before the given station in the specified direction.

Return type

List[str]

```
\begin{tabular}{l} \textbf{get\_lambda\_bar}(\textit{current\_hour: float, current\_weekday: bool, start\_station: str, critical\_station: str, direction: str) $\rightarrow$ float \\ \end{tabular}
```

Calculates the average arrival rate (lambda_bar) for passengers between a start station and a critical station in a given direction.

The arrival rate is calculated by: 1. Sorting the stations between the start and critical stations in the given direction. 2. Calculating the probability (a_i) of passengers arriving at each station between the start and critical stations. 3. Calculating the arrival rate (lambda_i) for each station between the start and critical stations. 4. Summing the product of a_i and lambda_i for each station between the start and critical stations. 5. Adding the arrival rate for the critical station.

Parameters

- **current_hour** (*float*) The current hour of the day.
- **current_weekday** (*bool*) Whether the current day is a weekday or not.
- **start_station** (*str*) The start station.
- **critical_station** (*str*) The critical station.
- **direction** (*str*) The direction of travel.

Returns

The average arrival rate (lambda_bar) between the start and critical stations.

Return type

float

 $get_lambda_for_station(current_hour: float, current_weekday: bool, station: str, direction: str) \rightarrow float$ Calculates the total arrival rate (lambda_i) for a given station and direction.

The total arrival rate is calculated by summing the smoothed arrival rates for all the destination stops that can be reached from the given station in the specified direction.

Parameters

- **current_hour** (*float*) The current hour of the day.
- **current_weekday** (*boo1*) Whether the current day is a weekday or not.
- **station** (str) The station to calculate the total arrival rate for.
- **direction** (*str*) The direction of travel.

Returns

The total arrival rate (lambda_i) for the given station and direction.

Return type

float

 $get_p_for_station(current_hour: float, current_weekday: bool, station: str, direction: str) \rightarrow float$ Calculates the probability (p_k) of passengers passing through a given station in a specific direction.

The probability is calculated by: 1. Getting all the previous stations and next stations for the given station and direction. 2. Calculating the total arrival rate for all the previous stations to the given station, and the next stations. 3. Calculating the "through rate" which is the arrival rate for all the next stations. 4. Returning the ratio of the through rate to the total rate as the probability (p_k) .

Parameters

- **current_hour** (*float*) The current hour of the day.
- **current_weekday** (*boo1*) Whether the current day is a weekday or not.
- **station** (*str*) The station to calculate the probability for.
- **direction** (*str*) The direction of travel.

Returns

The probability (p_k) of passengers passing through the given station in the specified direction.

Return type

float

```
get_smoothed_rate(current_hour, current_weekday, origin_stop, destination_stop)
```

Gets the smoothed arrival rate for a given origin stop, destination stop, current hour, and current weekday.

Parameters

- **current_hour** (*float*) The current hour of the day.
- current_weekday (bool) True if the current day is a weekday, False if it's a weekend.
- **origin_stop** (*str*) The origin stop.
- **destination_stop** (*str*) The destination stop.

Returns

The smoothed arrival rate for the given parameters, or 0 if no matching entry is found.

Return type

float

```
is_southbound_trip(origin_stop, destination_stop)
```

```
sort\_stations\_by\_direction(direction) \rightarrow List[str]
```

```
class transit_lab_simmetro.simulation_engine.passenger.__init__.Passenger
```

Bases: object

Represents a passenger in the simulation. Passengers have various attributes such as arrival time, origin, direction, destination, boarding time, alighting time, waiting time, and travel time. The *Passenger* class also keeps track of the number of times a passenger has been denied boarding.

The *Passenger* class is ordered based on the passenger's arrival time, allowing for easy sorting and comparison of passengers.

The *passenger_logger* attribute is used to log passenger events, such as when a passenger alights from the train. The *set_not_loggable()* method can be used to disable logging for a specific passenger.

```
__init__(arrival_time: float, origin: str, direction: str, destination: str)

property alighting_time

property boarding_time

denied_boarding()

property journey_time

passenger_logger: PassengerLogger | None = None

set_not_loggable()

property travel_time

property waiting_time

class transit_lab_simmetro.simulation_engine.passenger.__init__.PassengerGenerator

Bases: object

__init__(arrival_rate: ArrivalRate)
```

generate_passengers($current_hour$: float, $current_weekday$: bool, $boarding_stop$: str, $alighting_stop$: str, $delta\ t\ in\ seconds$: float) \rightarrow list

Generates a list of passenger arrival times based on the given parameters.

Parameters

- **current_hour** (*float*) The current hour of the day.
- **current_weekday** (*bool*) Whether the current day is a weekday or not.
- **boarding_stop** (*str*) The boarding stop for the passengers.
- **alighting_stop** (*str*) The alighting stop for the passengers.
- **delta_t_in_seconds** (*float*) The time interval in seconds for which to generate passenger arrivals.

Returns

A list of passenger arrival times in seconds.

simulation_logger: SimulationLogger | None = None

Return type

list

6.1.3 Simulation Module

The *Simulation* class is responsible for managing the simulation of a transit system. It handles the scheduling and updating of trains, as well as the overall simulation time and state.

The *SimulationContext* class is a context manager that sets up the necessary simulation-related attributes on the *Train*, *Passenger*, and *Station* classes when entering the context, and cleans up those attributes when exiting the context.

```
class transit_lab_simmetro.simulation_engine.simulation.simulation.Simulation
```

Bases: object

The *Simulation* class is responsible for managing the simulation of a transit system. It handles the scheduling and updating of trains, as well as the overall simulation time and state.

The *Simulation* class initializes with a schedule, path information, a signal control center, a train speed regulator, and various simulation parameters. It creates and manages a list of trains, dispatching new trains as scheduled and updating the existing trains at each time step. The class also provides methods to check the current simulation time and whether it is a weekday.

class transit_lab_simmetro.simulation_engine.simulation.simulation.SimulationContext

Bases: object

A context manager that sets up the simulation context for various simulation entities.

When entering the context, it sets the *simulation* attribute on the *Train*, *DummyTrainDecorator*, *Passenger*, and *Station* classes, and resets the *last id* attributes on the *Train* and *Passenger* classes.

When exiting the context, it sets the *simulation* attribute on the *Train*, *Passenger*, and *Station* classes back to *None*.

__init__(simulation: Simulation)

6.1.4 Train Module

Train module for simulating rail vehicle behavior in a transit system.

This module provides classes and utilities for modeling train operations, including speed regulation, passenger management, and train state handling.

Classes:

Train: Represents a rail vehicle with its physical and operational properties. DummyTrain: A simplified train model for testing and special scenarios. DummyTrainDecorator: Decorator for creating dummy train instances. TrainSpeedRegulator: Manages train speed and acceleration based on signal system. TrainSpeedRegulatorCTA: CTA-specific implementation of speed regulation. TrainPassengerManager: Handles passenger boarding, alighting, and capacity management.

The module integrates various aspects of train operations: - Speed control and acceleration management - Passenger flow and capacity handling - Train state transitions (e.g., dwelling, accelerating, braking) - Support for different signaling systems (fixed-block and moving-block)

Key features: - Realistic acceleration and deceleration profiles - Passenger boarding and alighting simulations - Integration with infrastructure and signaling systems - Support for various operational scenarios (normal running, emergency braking, etc.)

The train module is central to the transit simulation, providing a detailed representation of train behavior and its interaction with passengers and infrastructure.

class transit_lab_simmetro.simulation_engine.train.__init__.DummyTrain

Bases: Train

This class is a decorator for the *Train* class that creates a "dummy" train. The *DummyTrain* class inherits from the *Train* class and overrides some of its methods to provide a simplified implementation for testing or other purposes.

The *log()* method is a no-op, and the *should_log()* method always returns *False*, indicating that this train should not be logged.

The layover_and_turnback() method calls the delete() method, which removes the train from the simulation.

 $layover_and_turnback() \rightarrow None$

 $\log() \rightarrow None$

 $should_log() \rightarrow bool$

simulation: Simulation

class transit_lab_simmetro.simulation_engine.train.__init__.DummyTrainDecorator

Bases: DummyTrain

The *DummyTrainDecorator* class is a decorator for the *Train* class that creates a "dummy" train. It inherits from the *DummyTrain* class and overrides some of its methods to provide a simplified implementation for testing or other purposes.

The __init__ method copies the properties from the *DummyTrain* instance to the *DummyTrainDecorator* instance, sets the *train_id* to a unique identifier, initializes the *state* to *WaitingToBeDispatched*, and registers the train with the *train_speed_regulator*.

The *simulation* attribute is a reference to the *Simulation* object that the *DummyTrain* instance is part of, allowing the *DummyTrain* to interact with the overall simulation, such as removing itself from the simulation when the *layover_and_turnback()* method is called.

```
__init__(train: Train)
```

This method initializes a new instance of the *DummyTrainDecorator* class. It copies the properties from the provided *Train* instance to the *DummyTrainDecorator* instance, sets a unique identifier for the *train_id*, initializes the *state* to *WaitingToBeDispatched*, and registers the train with the *train_speed_regulator*.

simulation: Simulation

class transit_lab_simmetro.simulation_engine.train.__init__.Train

Bases: object

Represents a train in the simulation. The *Train* class handles the train's movement, speed regulation, passenger management, and logging.

The *Train* class has the following key responsibilities: - Manages the train's state and transitions between different states (e.g. waiting to be dispatched, dwelling at a station). - Updates the train's speed and distance travelled based on the current block and speed regulator. - Tracks the train's location and progress along the path. - Interacts with the train's passenger manager to handle passenger boarding and alighting. - Logs the train's status and updates to the simulation logger.

```
__init__(train_speed_regulator: TrainSpeedRegulator | TrainSpeedRegulatorCTA,

train_passenger_manager: TrainPassengerManager, path: Path, starting_block_index: int = 0,

dispatching_time: float | None = None, runid: str | None = None)
```

Initializes a new instance of the *Train* class.

Parameters

- train_speed_regulator (TrainSpeedRegulator / TrainSpeedRegulatorCTA) The speed regulator for the train.
- train_passenger_manager (TrainPassengerManager) The passenger manager for the train.
- path (Path) The path the train will follow.
- **starting_block_index** (*int*, *optional*) The index of the starting block on the path. Defaults to 0.
- **dispatching_time** (*float*, *optional*) The time when the train will be dispatched.
- **runid** (*str*, *optional*) The unique identifier for the train.

train_id

The unique identifier for the train.

Type

str

steps_since_last_log

```
The number of steps since the last log update.
        Type
          int
train_speed_regulator
    The speed regulator for the train.
           TrainSpeedRegulator | TrainSpeedRegulatorCTA
passenger_manager
    The passenger manager for the train.
          TrainPassengerManager
path
    The path the train will follow.
        Type
          Path
has_been_short_turned
    Indicates whether the train has been short-turned.
        Type
          bool
starting_block_index
    The index of the starting block on the path.
        Type
          int
current_block_index
    The index of the current block on the path.
        Type
          int
distance_travelled_in_current_block
    The distance the train has travelled in the current block.
        Type
          float
speed
    The current speed of the train.
        Type
          float
acceleration
    The current acceleration of the train.
        Type
          float
dispatching_time
    The time when the train will be dispatched.
        Type
          float
```

```
state
         The current state of the train.
             Type
               TrainState
     length
         The length of the train.
             Type
               float
     _should_log
         Indicates whether the train should be logged.
             Type
               bool
property acceleration_in_fps2: float
block_with_red_signals_in_sight() → Tuple[float | None, int | None]
property current_block: BlockType
property current_speed_code: float
delete() \rightarrow None
property distance_to_next_block: float
property distance_to_next_station: float
distance\_traveled\_from\_the\_start\_of\_block(asking\_block: BlockType) \rightarrow float
     Calculates the distance traveled by the train from the start of the specified block.
         Parameters
             asking_block (BlockType) - The block for which the distance traveled from the start is to
             be calculated.
         Returns
             The distance traveled by the train from the start of the specified block.
         Return type
             float
property first_block_after_station: BlockType | None
classmethod generate_train_id()
get_all_stops_ahead() \rightarrow list[Station]
get_next_station() \rightarrow Station
property location_from_terminal: float
log() \rightarrow None
property next_block: BlockType
property previous_block: BlockType
set\_state\_to\_dwelling\_at\_station(station: Station) \rightarrow None
```

 $should_log() \rightarrow bool$

```
simulation: Simulation
     property speed_in_fps: float
     property time_step: float
     property total_travelled_distance: float
     property total_travelled_distance_from_dispatch: float
     train_logger: TrainLogger | None = None
     update() \rightarrow None
     update\_block() \rightarrow None
           This method updates the train's position within the current block and moves the train to the next block if
           the distance traveled in the current block exceeds the block's length.
           It first checks if the distance traveled in the current block is greater than or equal to the block's length. If
           so, it subtracts the block's length from the distance traveled in the current block and increments the current
           block index to move the train to the next block.
           It then calculates the position of the train's rear relative to the start of the current block and iterates back-
           wards through the blocks, deactivating the blocks that the train's rear has left behind.
     update\_distance\_travelled() \rightarrow float
     update\_speed() \rightarrow None
class transit_lab_simmetro.simulation_engine.train.__init__.TrainPassengerManager
     Bases: object
     _{\tt init} (train_capacity: int, num_cars: int = 8, num_doors_per_car: int = 2, car_capacity: int = None,
                num\_seats\_per\_door: int = 20)
     alight_all_passengers(current_station, current_time)
     alight_passengers(current_station, current_time)
     board_passengers(passengers, current_time, car_assignment_weights: List[float] = [1, 3, 1, 1, 1, 1, 3, 1])
     get_door_metrics(alight_counts, boarding_counts)
     remaining\_capacity() \rightarrow int
     property total_passengers
class transit_lab_simmetro.simulation_engine.train.__init__.TrainSpeedRegulator
     Bases: object
     __init__(max_acceleration: float, normal_deceleration: float, emergency_deceleration: float)
     property braking_distance: float
     entered_symptomatic\_block(symptomatic\_block) \rightarrow None
     property normal_acceleration: float
```

```
property normal_acceleration_in_fps2
     property normal_decceleration_in_fps2
     property planning_distance: float
     register_train(train: Train) \rightarrow None
     regulate_acceleration()
     property train: Train
     train\_stopped\_at\_station(station: Station) \rightarrow None
     update\_train\_speed() \rightarrow None
class transit_lab_simmetro.simulation_engine.train.__init__.TrainSpeedRegulatorCTA
     Bases: object
     __init__(max_acceleration: float, normal_deceleration: float, emergency_deceleration: float,
               desired\_speed\_range: tuple[float, float] = (0.8, 1.0)
     property braking_distance: float
     property braking_distance_for_station: float
     entered\_symptomatic\_block(symptomatic\_block: OffScanSymptomaticBlockDecorator) \rightarrow None
     property normal_acceleration: float
     property normal_acceleration_in_fps2
     property normal_decceleration_in_fps2
     property planning_distance: float
     register_train(train: Train) \rightarrow None
     regulate_acceleration()
     property train: Train
     train\_stopped\_at\_station(station: Station) \rightarrow None
     update\_desired\_speed() \rightarrow None
     update\_train\_speed() \rightarrow None
```

6.2 Data Flow

Explain how data flows through the system.

6.2. Data Flow 25

SEVEN

DEVELOPMENT

7.1 Development Environment Setup

Guide on setting up the development environment.

7.2 Build Instructions

Explain how to build the project from source.

7.3 Testing

Describe how to run tests.

EIGHT

CONTRIBUTION GUIDELINES

8.1 How to Contribute

Instructions for making contributions, including coding standards and the pull request process.

8.2 Community Guidelines

Code of conduct and how to get involved in the community.

CHAPTER
NINE

FAQS

Address common questions and issues.

32 Chapter 9. Faqs

TEN

TROUBLESHOOTING

10.1 Common Issues

List common issues and their solutions.

10.2 Getting Help

Information on where to ask questions or report issues.

ELEVEN

CHANGELOG

11.1 Version History

List of changes for each version, including new features, bug fixes, and breaking changes.

TWELVE

APPENDIX

12.1 Glossary

Definitions of terms used in the documentation.

12.2 Further Reading

Links to additional resources such as blog posts, tutorials, and papers.

THIRTEEN

INDICES AND TABLES

- genindex
- modindex
- search

BIBLIOGRAPHY

- [KW07] Haris N. Koutsopoulos and Zhigao Wang. Simulation of Urban Rail Operations. *Transportation Research Record*, 2006:84–91, 2007. URL: https://api.semanticscholar.org/CorpusID:110620690.
- [WK11] Zhigao Wang and Haris N. Koutsopoulos. Calibration of urban rail simulation models: a methodology using SPSA algorithm. In *Proceedings of the Winter Simulation Conference*, Wsc '11, 3704–3714. Winter Simulation Conference, 2011.
- [Zho22] Jiali Zhou. *Urban rail simulation and applications in service planning and operations*. PhD thesis, Northeastern University, 2022.
- [ZK22] Jiali Zhou and Haris N. Koutsopoulos. Schedule-based Analysis of Transmission Risk in Public Transportation Systems. *ArXiv*, 2022. URL: https://api.semanticscholar.org/CorpusID:246904579.
- [ZKS20] Jiali Zhou, Haris N. Koutsopoulos, and Saeid Saidi. Evaluation of Subway Bottleneck Mitigation Strategies using Microscopic, Agent-Based Simulation. *Transportation Research Record*, 2674:649–661, 2020. URL: https://api.semanticscholar.org/CorpusID:218922083.

42 Bibliography

PYTHON MODULE INDEX

44 Python Module Index

INDEX

init_	_() (transit_lab_simmetro.simulation_engine.infra striction	_e _();{trans <u>its.lah.esimme</u> tro.simulation_engine.traininitTrain. method), 25
ini+	method), 11 () (transit lab simmetre simulation engine infra should	l log (transit lab_simmetro.simulation_engine.traininitTrain_ ettribute)
1111 (_	method), 11	attribute), 23
init	() (transit_lab_simmetro.simulation_engine.infrastructure	e init DispatchingBlockDecorator
	method), 12	ciiniiz inputerining zito eniz ecciration
init_	_() (transit_lab_simmetro.simulation_engine.infrashstenas	etBliogk DispatelologMovingBlockDecoratoran-
	method), 12	sit_lab_simmetro.simulation_engine.infrastructureinit),
init_	_() (transit_lab_simmetro.simulation_engine.infrastructure	e. <u>11</u> initMovingBlock
	method), 12 acceler	ration(transit_lab_simmetro.simulation_engine.traininitTra
init_	_() (transit_lab_simmetro.simulation_engine.infrastructure	e. <u>attribute</u> M&ingBlockControl
	,,	ration_in_fps2 (tran-
init_	_() (transit_lab_simmetro.simulation_engine.infrastructure	
	method), 13	property), 23
init_		e <u>e ()</u> int <u>ran</u> setulab_simmetro.simulation_engine.infrastructureinit
	method), 13	method), 11
init_		<u>tee ()</u> iht <u>ran</u> StglahCrimmetCommulation_engine.infrastructureinit
	method), 14	method), 12
init_		be <u>()</u> intranstolaborimmetro.simulation_engine.infrastructureinit
	method), 14	method), 12
1n1t_		<u>ee ()</u> int <u>ranstaluh</u> simmetro.simulation_engine.infrastructureinit method), 12
ii-	method), 14	memod), 12 <u>be ()ihtransterlahasi</u> mmetro.simulation_engine.infrastructureinit
1N1 t_	() (transit_tab_simmetro.simutation_engine.injra siruena t method), 15	e <u>e (mathod),</u> 13 method), 13
ini+		method), 15 ผ ล (<u>i) (tAmwit]Rdne</u> simmetro.simulation_engine.infrastructure. <u></u> init
	method), 15	method), 15
init	() (transit_lab_simmetro.simulation_engine.pass edg er <u>con</u>	
	method), 18	sit_lab_simmetro.simulation_engine.infrastructureinitBlock
init	() (transit_lab_simmetro.simulation_engine.passenger	e v
	method), 18 add_obs	server() (tran-
init_		m sitaliah_Simutetrio:s imulation_engine.infrastructureinitBlock
	method), 19	method), 11
init_	_() (transit_lab_simmetro.simulation_engine.simu tdri@ht ii	vallutipassiangarishContext (tran-
	method), 20	sit_lab_simmetro.simulation_engine.traininitTrainPassenge
init_	_() (transit_lab_simmetro.simulation_engine.traininit_	
		_passengers() (tran-
init_	_() (transit_lab_simmetro.simulation_engine.traininit_	Titalla b_simmetro.simulation_engine.traininitTrainPassenge
	method), 21	method), 24
init_	_() (transit_lab_simmetro.simulation_engine.train <mark>al_i</mark> ght_i	
	method), 24	sit_lab_simmetro.simulation_engine.passengerinitPassenge
ini+	() (transit lab simulation angine train init	Thromathy by (Progulator

method), 24

Symbols

```
ArrivalRate
                                         (class
                                                                                       tran- current_speed_code()
                                                                                                                                                                                          (tran-
                sit lab simmetro.simulation engine.passenger. init ), sit lab simmetro.simulation engine.infrastructure. init .Movi
                                                                                                                    method), 12
                                                                                                    current_speed_code()
                                                                                                                                                                                          (tran-
В
                                                                                                                    sit lab simmetro.simulation engine.infrastructure. init .OffSc
                                                                                                                    method), 13
Block
                                (class
                                                               in
                                                                                       tran-
                sit\_lab\_simmetro.simulation\_engine.infrastructur & \verb"urinent") speed\_code()
                                                                                                                                                                                          (tran-
                                                                                                                    sit lab simmetro.simulation engine.infrastructure. init .Term
block_logger(transit_lab_simmetro.simulation_engine.infrastructureethodi), 15Block
                                                                                                    current_train(transit_lab_simmetro.simulation_engine.infrastructure._
                attribute), 12
                                                                                                                    property), 12
block_with_red_signals_in_sight()
                                                                                      (tran-
                sit_lab_simmetro.simulation_engine.train.__init_
                                                                                                       Train
                method), 23
board_passengers()
                                                                                      (tran-
                                                                                                   sit_lab_simmetro.simulation_engine.train.__init__.TrainPasseengerManager
                method), 24
                                                                                                    deactivate()(transit lab simmetro.simulation engine.infrastructure.
board_passengers_based_on_destiations_and_probability (method), 13
                (transit_lab_simmetro.simulation_engine.infrastrutetre_cip(transtatiletb_simmetro.simulation_engine.train.__init__.Train
                method), 14
                                                                                                                    method), 23
board_passengers_onto_train()
                                                                                      (tran- denied_boarding()
                                                                                                                                                                                          (tran-
                sit\_lab\_simmetro.simulation\_engine.infrastructure.\_\_init\_\_. \textbf{Station}\_simmetro.simulation\_engine.passenger.\_\_init\_\_. Passenger.
                                                                                                                    method), 18
boarding\_time (\textit{transit\_lab\_simmetro.simulation\_engine.passenger} \\ \underline{\textbf{passenger}} \\ \underline{\textbf{transit\_lab\_simmetro.simulation\_engine.passenger}} \\ \underline{\textbf{transit\_lab\_sim
                                                                                                                                                                                          (tran-
                property), 18
                                                                                                                    sit_lab_simmetro.simulation_engine.train.__init__.Train
braking_distance
                                                                                      (tran-
                                                                                                                     attribute), 22
                sit_lab_simmetro.simulation_engine.train.__init_DfspinfphedReputation_engine.train.__init_DfspinfphedReputation_engine.train.
                                                                                                                                                                 (class
                property), 24
                                                                                                                    sit lab simmetro.simulation engine.infrastructure. init ),
braking_distance
                                                                                      (tran-
                sit_lab_simmetro.simulation_engine.train.__init_DIspinSpindReproductive for the corator (class in tran-
                property), 25
                                                                                                                     sit_lab_simmetro.simulation_engine.infrastructure.__init__),
braking_distance_for_station
                                                                                      (tran-
                                                                                                                     12
                sit_lab_simmetro.simulation_engine.train.__init_dTrainSpeedRegulator6T64k
                                                                                                                                                                                          (tran-
                property), 25
                                                                                                                    sit lab simmetro.simulation engine.train. init .Train
                                                                                                                    property), 23
C
                                                                                                    distance_to_next_station
                                                                                                                                                                                          (tran-
                                                                                                                    sit_lab_simmetro.simulation_engine.train.__init__.Train
civil_speed_limit
                                                                                      (tran-
                sit_lab_simmetro.simulation_engine.infrastructure.__init___MorengBlock
                                                                                                   distance_traveled_from_the_start_of_block()
                property), 12
copy()(transit_lab_simmetro.simulation_engine.infrastructure.__init_transit_lab_simmetro.simulation_engine.train.__init__.Train
                                                                                                                     method), 23
                method), 13
current_block (transit_lab_simmetro.simulation_engine.tdiatanaa_tranalled_in_current_block
                                                                                                                                                                                          (tran-
                                                                                                                    sit_lab_simmetro.simulation_engine.train.__init__.Train
                property), 23
current_block_index
                                                                                      (tran-
                sit_lab_simmetro.simulation_engine.train.__init_Dummy.Train
                                                                                                                                           (class
                                                                                                                                                                       in
                                                                                                                                                                                           tran-
                                                                                                                     sit lab simmetro.simulation engine.train. init ),
                attribute), 22
current_speed_code
                                                                                      (tran-
                sit_lab_simmetro.simulation_engine.train.__init_DummyTrainDecorator
                                                                                                                                                        (class
                                                                                                                                                                                           tran-
                                                                                                                     sit_lab_simmetro.simulation_engine.train.__init__),
                property), 23
current_speed_code()
                                                                                      (tran-
                sit\_lab\_simmetro.simulation\_engine.infrastructure \_\_init\_\_.AbstractBlock
                method), 11
                                                                                      (tran- entered_symptomatic_block()
current_speed_code()
                                                                                                                                                                                          (tran-
                sit_lab_simmetro.simulation_engine.infrastructure.__init__,Btolab_simmetro.simulation_engine.train.__init__,TrainSpeedRe
                method), 12
                                                                                                                    method), 24
```

entered	_symptomatic_block() (tran- sit_lab_simmetro.simulation_engine.traininit_ method), 25		tance_to_next_train() exid_Respukiton@Tivo.simulation_engine.infro method), 13	(tran- astructureinitMovi
F		get_dis	<pre>tance_to_the_next_station() sit_lab_simmetro.simulation_engine.infra</pre>	(tran- astructure. init .Path
	lock_after_station (tran- sit_lab_simmetro.simulation_engine.traininit_	_g q ‡ _{ai} doo	method), 13	(tran-
	<pre>property), 23 ct() (transit_lab_simmetro.simulation_engine.inj class method), 14</pre>	frastructur get_dwe	- <u>inethod), </u>	(tran-
G			sit_lab_simmetro.simulation_engine.inframethod), 14	istructure. <u> </u>
	e_and_add_passengers() (tran-	get_lam	bda_bar()	(tran-
	e_anu_auu_passengers() (tran- sit_lab_simmetro.simulation_engine.infrastructus method), 14	reinit	sit_lab_simmetro.simulation_engine.pass _Station method), 16	engerinitArrivalR
	o naccongona() (tran	get_lam	bda_for_station()	(tran-
	e_passengers() (ran- sit_lab_simmetro.simulation_engine.passenger method), 18		memoa), 17	
generat	e_train_id() (tran-		t_station() sit_lab_simmetro.simulation_engine.infra	(tran-
	sit_lab_simmetro.simulation_engine.traininit_class method), 23		method), 13	(tran-
	() (transit_lab_simmetro.simulation_engine.passe method), 15	enger <u>. i</u> n	su_tab_simmetro.simutation_engine.trair	`
get_all	_destination_stops_for_origin() (tran-	get nev	method), 23	(tran-
	sit_lab_simmetro.simulation_engine.passenger method), 16		rivatRate sit_lab_simmetro.simulation_engine.infra method), 13	`
get_all	_destination_stops_for_origin_and_dire (transit_lab_simmetro.simulation_engine.passeng method), 16	ction() get_p_f ger:titit_		(tran- engerinitArrivalR
get_all	nrevious stons for station and direct	ion()	method), 17	
	(transit_lab_simmetro.simulation_engine.passengmethod), 16	get_pre gerinit_	su_ido_simmerro.simuiditon_engine.iijre	(tran- istructureinitPath
get_all	_stops_ahead() (tran-	aa+ ama	method), 13	(tuan
	sit_lab_simmetro.simulation_engine.infrastructus method), 13	re. <u> </u>	opnou_rate() -rain -sit_lab_simmetro.simulation_engine.pass -method), 17	(tran- engerinitArrivalR
get_all	_stops_ahead() (tran-	aet sne		(tran-
	sit_lab_simmetro.simulation_engine.traininit_ method), 23	_3.1 Fain F	sit_lab_simmetro.simulation_engine.infra	`
get_all	stons ahead which are served() (tran-		method), 13	
	sit_lab_simmetro.simulation_engine.infrastructus method), 13	re. <u></u> init	su_tao_summero.simutation_engine.tijre	(tran- istructureinitPath
get_blo	ck_by_id() (tran-	got tot	method), 13	(tuan
	sit_lab_simmetro.simulation_engine.infrastructus method), 13	re. <u></u> init	su_ido_summerro.simuiditon_engine.uigre	(tran- astructureinitPath
get_blo	ck_by_id() (tran-		method), 14	
	sit_lab_simmetro.simulation_engine.infrastructumethod), 14			
get_blo	ck_index_by_id() (tran-		n_short_turned	(tran- n init Train
	sit_lab_simmetro.simulation_engine.infrastructus method), 13	reınit	<u>_spa_{tt}qu_simmetro.simulation_engine.trair</u> attribute), 22	<u>_</u> иш11шп
get_cur	rent_hour() (tran-	i	Simulation	
	sit_lab_simmetro.simulation_engine.simulation.s method), 19		<i>Simulation</i> ast_train	(tran-
	<i></i>	-u_U1_I	sit_lab_simmetro.simulation_engine.infra	`

property), 12	lav	over_and_turi	nhack()	(tran-	
is_inspected()	(tran-		**	_engine.trainini	t DummyTrain
sit_lab_simmetro.simulation_engine.ii	`				
method), 14				tion_engine.train	init .Train
<pre>is_it_clear_to_dispatch()</pre>	(tran-	attribute),			<u>-</u>
sit_lab_simmetro.simulation_engine.ir	ıfrastructur d.o c	/ /		· (tran-	
method), 12			0	_engine.trainini	t .Train
<pre>is_it_clear_to_dispatch()</pre>	(tran-	property),		_ 0	
sit_lab_simmetro.simulation_engine.ir	ıfrastructurd. <u>o</u> g			propengine.traini	nitDummyTrai
method), 12		method), 2			
<pre>is_occupied(transit_lab_simmetro.simulation</pre>	_engine.infr bog	rwturansiinlab .	AibwretetBlionk latio	on_engine.traini	nitTrain
property), 11		method), 2		· ·	
<pre>is_occupied(transit_lab_simmetro.simulation</pre>	_engine.infrasti	ructureinit	Block		
property), 12	M				
$\verb"is_occupied" (\textit{transit_lab_simmetro.simulation}) \\$	_engine.infrast	rectiffspaticking	Mogioer tock	(tran-	
property), 13			-	_engine.infrastruct	ure. init .Path
<pre>is_occupied_by()</pre>	(tran-	method), 1	14	_ 0 ,	
$sit_lab_simmetro.simulation_engine.in$	ıfrastructur _{¶ak}	<u>ini6££&b&ka</u> 6#R	Jerk()	(tran-	
method), 11				_engine.infrastruct	ureinitPath
<pre>is_occupied_by()</pre>	(tran-	method), 1		_ 0	
$sit_lab_simmetro.simulation_engine.in$	ıfrastructur _{mod}	inieBlock			
method), 12		transit_lab_	_simmetro.simu	lation_engine.i	nfrastructure.
<pre>is_occupied_by()</pre>	(tran-	11			
sit_lab_simmetro.simulation_engine.ii method), 13	ıfrastructure	initan M ev <u>i</u> naBl 15	<u>⊈</u> ⊀mmetro.simu]	lation_engine.p	assengerini
<pre>is_short_turn()</pre>	(tran-		simmetro simu	lation_engine.s	imulation simu
sit_lab_simmetro.simulation_engine.ir	ıfrastructure	initPath	_DIMMC CI O I DIMO	racion_enginero	Imara cron i Simo
method), 14		transit lab	simmetro.simu	lation_engine.t	rain. init .
<pre>is_short_turned_at_this_station()</pre>	(tran-	20		_ 3	
$sit_lab_simmetro.simulation_engine.in$	ıfrastructur e lov	inigb1 <i>Be4k</i> i	(class	in tran-	
method), 14			`	_engine.infrastruct	ure. init),
<pre>is_southbound_trip()</pre>	(tran-	12		_ 0 ,	
sit_lab_simmetro.simulation_engine.p method), 18	assenger. <u></u> injt _t	-	,	in tran- e_engine.infrastruct	ura init
<pre>is_weekday()(transit_lab_simmetro.simulatio</pre>	n_engine.simul	sti_tab_sti ation.simulation.	Simulation	_engine.iigrasiruci	ureinii),
method), 19	N.I.				
1	N				
J		ded_braking_o		(tran-	
$\verb"journey_time" (transit_lab_simmetro.simulatio$	n_engine.passe	nger. <u>sinitab.</u> Ri	ssauge r.simulation	_engine.infrastruct	ureinitMovi
property), 18		method), 1	13		
1	nex	t_block(<i>transi</i>	t_lab_simmetro.si	mulation_engine.tra	aininitTrain
L		property),	23		
<pre>last_train_visit_time</pre>	(tran- nex	t_train()(<i>trai</i>	nsit_lab_simmetro	.simulation_engine.	.infrastructurei
sit_lab_simmetro.simulation_engine.ir	ıfrastructure	init :B&&& d), 1	13		
property), 12	nor	mal_accelera	tion	(tran-	
last_train_visit_time	(tran-	sit_lab_sii	mmetro.simulation	_engine.trainini	tTrainSpeedRe
$sit_lab_simmetro.simulation_engine.in$					
property), 12	nor	mal_accelera		(tran-	
<pre>last_train_visit_time</pre>	(tran-			_engine.trainini	tTrainSpeedRe
$sit_lab_simmetro.simulation_engine.in$	ıfrastructure	init Dispentsh i	n&MovingBlockDe		
property), 12	nor	mal_accelera	tion_in_fps2	(tran-	
<pre>last_train_visit_time</pre>	(tran-			_engine.trainini	tTrainSpeedRe
$sit_lab_simmetro.simulation_engine.in$	ıfrastructure	init !Stapie nty),	24		
property), 14					

```
normal_acceleration_in_fps2
                                              (tran- ready_to_dispatch()
                                                                                                     (tran-
        method), 12
        property), 25
normal_decceleration_in_fps2
                                              (tran-register_moving_block_control_center() (tran-
        sit_lab_simmetro.simulation_engine.train.__init__.TrainSpexidNetpukitmetro.simulation_engine.infrastructure.__init__.Movi
        property), 25
                                                               method), 13
normal_decceleration_in_fps2
                                              (tran- register_train()
                                                                                                     (tran-
        sit_lab_simmetro.simulation_engine.train.__init__.TrainSpexidNetpukiton@Hib.simulation_engine.train.__init__.TrainSpeedRe
        property), 25
                                                               method), 25
notify_observers()
                                              (tran- register_train()
                                                                                                     (tran-
        sit_lab_simmetro.simulation_engine.infrastructure.__init__sholdb_simmetro.simulation_engine.train.
                                                                                                      _init__.TrainSpeedRe
        method), 12
                                                               method), 25
                                                      regulate_acceleration()
                                                                                                    (tran-
O
                                                               sit_lab_simmetro.simulation_engine.train.__init__.TrainSpeedRe
                                                               method), 25
OffScanSymptomaticBlockDecorator (class in tran-
         sit\_lab\_simmetro.simulation\_engine.infrastructure.eghliate\_acceleration()
                                                                                                     (tran-
                                                               sit_lab_simmetro.simulation_engine.train.__init__.TrainSpeedRe
                                                               method), 25
P
                                                      remaining_capacity()
                                                                                                    (tran-
                                                               sit lab simmetro.simulation engine.train.
                                                                                                      init .TrainPassenge
Passenger
                    (class
                                               tran-
                                                               method), 24
         sit lab simmetro.simulation engine.passenger. init ),
                                                      remove_communicated_speed_code()
                                                               sit_lab_simmetro.simulation_engine.infrastructure.__init__.Block
passenger_logger
                                              (tran-
        sit_lab_simmetro.simulation_engine.passenger.__init__.Passetbed), 12
                                                      remove_train()
                                                                                                     (tran-
        attribute), 18
                                                               sit lab simmetro.simulation engine.simulation.simulation.Simulation
passenger_manager
                                              (tran-
                                                               method), 19
        sit lab simmetro.simulation engine.train.
                                                 init
                                                      .Train
                                                      restore_speed_codes_to_upstream_blocks()
         attribute), 22
                                                               (transit_lab_simmetro.simulation_engine.infrastructure.__init__.
PassengerGenerator
                           (class
                                               tran-
                                                               method), 14
         sit_lab_simmetro.simulation_engine.passenger.__init_
                                                      init__), method), 14
run() (transit_lab_simmetro.simulation_engine.simulation.simulation.Simulation.
Path (class in transit_lab_simmetro.simulation_engine.infrastructure.method), 19
path(transit_lab_simmetro.simulation_engine.train.__init_
         attribute), 22
                                                      send_speed_codes_to_upstream_blocks()
                                                               sit_lab_simmetro.simulation_engine.infrastructure.__init__.Signa
path_distance_alignment()
                                              (tran-
        sit_lab_simmetro.simulation_engine.infrastructure.__init__nPathod), 14
                                                      set_block_slow_zone()
        method), 14
planning_distance
                                                               sit lab simmetro.simulation engine.infrastructure. init .Path
                                              (tran-
         sit_lab_simmetro.simulation_engine.train.__init__.TrainSpendRegd(later
        property), 25
                                                      set_not_loggable()
                                                                                                     (tran-
                                                               sit_lab_simmetro.simulation_engine.passenger.__init__.Passenge
planning_distance
                                              (tran-
                                                init .TrainSpeedthoddal&rCTA
        sit lab simmetro.simulation engine.train.
                                                      set_slow_zone()
                                                                                                     (tran-
        property), 25
previous_block
                                                               sit_lab_simmetro.simulation_engine.infrastructure.__init__.Abstr
                                              (tran-
                                                               method), 11
        sit_lab_simmetro.simulation_engine.train.__init__.Train
                                                      set_slow_zone()
        property), 23
                                                               sit_lab_simmetro.simulation_engine.infrastructure.__init__.Block
R
                                                               method), 12
                                                      set_state_to_dwelling_at_station()
                                                                                                     (tran-
ready_to_dispatch()
                                              (tran-
                                                              _sit_lab_simmetrasionulation_engine.train.__init__.Train
        sit_lab_simmetro.simulation_engine.infrastructure.__init__
                                                               method), 23
        method), 12
                                                      set_unsymptomatic()
                                                                                                     (tran-
                                                               sit_lab_simmetro.simulation_engine.infrastructure.__init__.OffSc
```

method), 13	
short_turn() (transit_lab_simmetro.simulation_engine.infrastructureinitPath in tran-	
$method), 14 \\ sit_lab_simmetro.simulation_engine.infrastructure.__init__), \\ should_log() (transit_lab_simmetro.simulation_engine.train.__init\5DummyTrain$	
method), 20 time_step (transit_lab_simmetro.simulation_engine.traininitTrain should_log() (transit_lab_simmetro.simulation_engine.traininit_propairy), 24	ı
method), 23 to_dict() (transit_lab_simmetro.simulation_engine.infrastructureini	it
SignalControlCenter (class in tran-method), 14	<i>-</i> _
1/	_
sit_lab_simmetro.simulation_engine.traininitTrainPassen Simulation (class in tran- property), 24	ge
sit_lab_simmetro.simulation_engine.simulation.simulation\ranglerapproperty), 24	
sit_lab_simmetro.simulation_engine.traininitTrain simulation(transit_lab_simmetro.simulation_engine.infrastructure.pripjerty,\textstation	
simulation(transit_lab_simmetro.simulation_engine.traininitPuminy_siminetro.simulation_engine.traininitTrain	
attribute), 20 property), 24	
simulation(transit_lab_simmetro.simulation_engine.train_rinitDummyTrainDecorator in tran- attribute), 21	
simulation(transit_lab_simmetro.simulation_engine.traininitTrain	
attribute), 24 train(transit_lab_simmetro.simulation_engine.traininitTrainSpeed	dI
simulation_logger (tran-property), 25	
sit_lab_simmetro.simulation_engine.simulation.simulation_fromsulation_simmetro.simulation_engine.traininitTrainSpeed	dl
property), 25	
SimulationContext (class in tran-train_id(transit_lab_simmetro.simulation_engine.traininitTrain_sit_lab_simmetro.simulation_engine.simulation.simulation),attribute), 21	
train_logger(transit_lab_simmetro.simulation_engine.traininitT	rc
SlowZone (class in tran-attribute), 24	
sit_lab_simmetro.simulation_engine.infrastructure_trainit_speed_regulator (tran-	
sit_lab_simmetro.simulation_engine.traininitTrain	
sort_stations_by_direction() (tran-attribute), 22	
sit_lab_simmetro.simulation_engine.passengerinit_in_Assival Rate(), 22 sit_lab_simmetro.simulation_engine.passengerinit_in_Assival Rate(), 22 method), 18 sit_lab_simmetro.simulation_engine_train_initTrainSpeedle	
su_tao_sintatetro.sintatation_engine.tratits_inta_inta_inta_inta_inta_inta_inta_inta	łе
ait lab ainmetra ainmetria ancies infrastructura init. Marina Plack	
property), 13 sit_lab_simmetro.simulation_engine.traininitTrainSpeedIspeed(transit_lab_simmetro.simulation_engine.traininitTrain method), 25	(e
attribute), 22	
speed_in_fps(transit_lab_simmetro.simulation_engine.traininit_sit_rain_simmetro.simulation_engine.traininit),	
property), 24 24	
starting_block_index (tran- TrainSpeedRegulator (class in tran-	
sit_lab_simmetro.simulation_engine.traininitTrain sit_lab_simmetro.simulation_engine.traininit),	
attribute), 22	
state(transit_lab_simmetro.simulation_engine.traininit_TrainspeedRegulatorCTA (class in tran- attribute), 22	
sit_lab_simmetro.simulation_engine.traininit), Station (class in tran- 25	
sit_lab_simmetro.simulation_engine.infrastructure_rainit_1ab_simmetro.simulation_engine.infrastructure	iı
14 module, 11	11
station_logger (tran- transit lab simmetro simulation engine passenger init	
sit_lab_simmetro.simulation_engine.infrastructureiniodustation_engineiniodustation	_
attribute), 15 transit_lab_simmetro.simulation_engine.simulation.simula	t.
steps_since_last_log (tran- modulo 10	
sit_lab_simmetro.simulation_engine.traininittTrainsit_lab_simmetro.simulation_engine.traininit	
attribute), 21 module. 20	

module, 20

```
travel_time(transit_lab_simmetro.simulation_engine.passenger.__init__.Passenger
         property), 18
U
update() (transit_lab_simmetro.simulation_engine.infrastructure.__init__.SignalControlCenter
         method), 14
update() (transit_lab_simmetro.simulation_engine.train.__init__.Train
         method), 24
update_block()
                                                (tran-
         sit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.Train
         method), 24
update_desired_speed()
                                                (tran-
         sit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.TrainSpeedRegulatorCTA
         method), 25
update_distance_travelled()
                                                (tran-
         sit lab simmetro.simulation engine.train. init .Train
         method), 24
update_speed()
                                                (tran-
         sit_lab_simmetro.simulation_engine.train.__init__.Train
         method), 24
update_train_speed()
                                                (tran-
         sit_lab_simmetro.simulation_engine.train.__init__.TrainSpeedRegulator
         method), 25
update_train_speed()
         sit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.TrainSpeedRegulatorCTA
         method), 25
W
waiting_time(transit_lab_simmetro.simulation_engine.passenger.__init__.Passenger
         property), 18
```