

---

# **TransitLab SimMETRO**

**Mojtaba Yousefi**

**Aug 28, 2024**



## CONTENTS:

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Project Overview . . . . .	1
1.2	Features . . . . .	1
1.3	License . . . . .	2
<b>2</b>	<b>Setup Guide</b>	<b>3</b>
<b>3</b>	<b>Getting Started</b>	<b>5</b>
3.1	Quick Start . . . . .	5
3.2	Understanding the Inputs and Config . . . . .	6
<b>4</b>	<b>Usage</b>	<b>7</b>
4.1	Basic Usage . . . . .	7
4.2	Advanced Usage . . . . .	7
<b>5</b>	<b>Configuration</b>	<b>9</b>
5.1	Configuration Options . . . . .	9
5.2	Default Configuration . . . . .	9
<b>6</b>	<b>Architecture</b>	<b>11</b>
6.1	Components Overview . . . . .	11
6.1.1	Infrastructure Module . . . . .	11
6.1.2	Passenger Module . . . . .	15
6.1.3	Simulation Module . . . . .	16
6.1.4	Train Module . . . . .	16
6.2	Data Flow . . . . .	19
<b>7</b>	<b>Development</b>	<b>21</b>
7.1	Development Environment Setup . . . . .	21
7.2	Build Instructions . . . . .	21
7.3	Testing . . . . .	21
<b>8</b>	<b>Contribution Guidelines</b>	<b>23</b>
8.1	How to Contribute . . . . .	23
8.2	Community Guidelines . . . . .	23
<b>9</b>	<b>API Documentation</b>	<b>25</b>
9.1	API Overview . . . . .	25
9.2	Endpoints/Functions . . . . .	25
<b>10</b>	<b>Faqs</b>	<b>27</b>

<b>11 Troubleshooting</b>	<b>29</b>
11.1 Common Issues . . . . .	29
11.2 Getting Help . . . . .	29
<b>12 Changelog</b>	<b>31</b>
12.1 Version History . . . . .	31
<b>13 Appendix</b>	<b>33</b>
13.1 Glossary . . . . .	33
13.2 Further Reading . . . . .	33
<b>14 Indices and tables</b>	<b>35</b>
<b>Bibliography</b>	<b>37</b>
<b>Python Module Index</b>	<b>39</b>
<b>Index</b>	<b>41</b>

## 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, reseachers 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.

## 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.





## 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.

## **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.



## CONFIGURATION

### 5.1 Configuration Options

Document configuration options, environment variables, etc.

### 5.2 Default Configuration

Explain the default setup and how to customize it.



## ARCHITECTURE

### 6.1 Components Overview

#### 6.1.1 Infrastructure Module

```
class transit_lab_simmetro.simulation_engine.infrastructure.__init__.AbstractBlock
    Bases: ABC
    __init__(block_id: str, block_alt_name: str, visible_distance: float, length: float, default_speed_code:
        float, station: Station | None = None)

    abstract current_speed_code(requesting_train: Train) → float
    abstract property is_occupied: bool
    abstract is_occupied_by(train: Train) → bool
    set_slow_zone(speed_limit: float)

class transit_lab_simmetro.simulation_engine.infrastructure.__init__.Block
    Bases: AbstractBlock
    __init__(*args, speed_codes_to_communicate: Dict[str, float] | None = None, **kwargs)
    activate(entering_train: Train) → None
    add_communicated_speed_code(block_id: str, speed_code: float) → None
    add_observer(observer: SignalControlCenter) → None
    block_logger: BlockActivationLogger | None = None
    current_speed_code(requesting_train: Train) → float
    deactivate(exiting_train: Train) → None
    property id_of_last_train: str | None
    property is_occupied: bool
    is_occupied_by(train: Train) → bool
    property last_train_visit_time: float
```

`notify_observers()` → None

`remove_communicated_speed_code(block_id: str)` → 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)` → None

`is_it_clear_to_dispatch()` → 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)` → None

`is_it_clear_to_dispatch()` → bool

property `last_train_visit_time`: float

`ready_to_dispatch(train: Train)` → bool

**class** `transit_lab_simmetro.simulation_engine.infrastructure.__init__.MovingBlock`

Bases: `AbstractBlock`

`__init__(*args, **kwargs)`

`activate(entering_train: Train)` → None

property `civil_speed_limit`: float

`current_speed_code(requesting_train: Train)` → float

property `current_train`: Train | None

`deactivate(exiting_train: Train)` → None

property `is_occupied`: bool

`is_occupied_by(train: Train)` → bool

`next_train(requesting_train: Train)` → Train | None

`register_moving_block_control_center(moving_block_control)` → 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) → None
```

```
get_distance_to_next_train(asking_train: Train) → float
```

```
get_speed_code(asking_train: Train, asking_block: MovingBlock) → 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) → None
```

```
current_speed_code(requesting_train: Train) → float
```

```
set_unsymptomatic() → 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) → List[Station]
```

```
get_all_stops_ahead_which_are_served(block_index: int) → List[Station]
```

```
get_block_by_id(block_id: str) → MovingBlock | Block
```

```
get_block_index_by_id(block_id: str) → int
```

```
get_distance_to_the_next_station(current_block_index: int, current_location_on_block: float) → float
```

```
get_next_station(current_block_index: int) → Station | None
```

```
get_next_train(current_block_index: int) → Train | None
```

```
get_previous_train(current_block_index: int) → Train | None
```

```
get_total_length() → float
```

```
get_total_travelled_distance(current_block_index: int, distance_travelled_in_current_block: float) → float
```

```
is_inspected() → bool
```

```
is_short_turn() → 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) → Block | None

    restore_speed_codes_to_upstream_blocks(block: Block) → None

    send_speed_codes_to_upstream_blocks(block: Block) → None

    update(block: Block) → 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) → SlowZone

    to_dict() → 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) → List[Passenger]

    board_passengers_onto_train(train_capacity: int, served_destinations: List[str]) → List[Passenger]

    generate_and_add_passengers(train: Train, holding_time: float = 0) → None

    get_dwell_time(door_metrics: List[Tuple(int, int, int)]) → 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) → None

    current_speed_code(requesting_train: Train) → float
```

## 6.1.2 Passenger Module

```

class transit_lab_simmetro.simulation_engine.passenger.__init__.ArrivalRate
    Bases: object
    __init__(filename, demand_factor: int = 1)

    get_a_i(current_hour: float, current_weekday: bool, start_station: str, critical_station: str, direction: str)
        → float

    get_all_destination_stops_for_origin(origin_stop: str) → List[str]

    get_all_destination_stops_for_origin_and_direction(origin_stop, direction) → List[str]

    get_all_previous_stops_for_station_and_direction(station: str, direction: str) → List[str]

    get_lambda_bar(current_hour: float, current_weekday: bool, start_station: str, critical_station: str,
        direction: str) → float

    get_lambda_for_station(current_hour: float, current_weekday: bool, station: str, direction: str) → float

    get_p_for_station(current_hour: float, current_weekday: bool, station: str, direction: str) → float

    get_smoothed_rate(current_hour, current_weekday, origin_stop, destination_stop)

    is_southbound_trip(origin_stop, destination_stop)

    sort_stations_by_direction(direction) → List[str]

class transit_lab_simmetro.simulation_engine.passenger.__init__.Passenger
    Bases: object
    __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) → list

```

### 6.1.3 Simulation Module

```
class transit_lab_simmetro.simulation_engine.simulation.simulation.Simulation
    Bases: object
    __init__(schedule: Schedule | BaseSchedule, path: Dict[str, Path], signal_control_center:
        MovingBlockControl | SignalControlCenter, train_speed_regulator: str, time_step: float = 0.5,
        start_hour: float = 5.0, is_weekday: bool = True, total_time: float = 14400)

    get_current_hour() → float
    is_weekday() → bool
    remove_train(train: Train) → None
    run() → None
    simulation_logger: SimulationLogger | None = None

class transit_lab_simmetro.simulation_engine.simulation.simulation.SimulationContext
    Bases: object
    __init__(simulation: Simulation)
```

### 6.1.4 Train Module

```
class transit_lab_simmetro.simulation_engine.train.__init__.DummyTrain
    Bases: Train
    layover_and_turnback() → None
    log() → None
    should_log() → bool
    simulation: Simulation

class transit_lab_simmetro.simulation_engine.train.__init__.DummyTrainDecorator
    Bases: DummyTrain
    __init__(train: Train)
    simulation: Simulation

class transit_lab_simmetro.simulation_engine.train.__init__.Train
    Bases: object
    __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)

    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() → None
property distance_to_next_block: float
property distance_to_next_station: float
distance_traveled_from_the_start_of_block(asking_block: BlockType) → float
property first_block_after_station: BlockType | None
classmethod generate_train_id()
get_all_stops_ahead() → list[Station]
get_next_station() → Station
property location_from_terminal: float
log() → None
property next_block: BlockType
property previous_block: BlockType
set_state_to_dwelling_at_station(station: Station) → None
should_log() → 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() → None
update_block() → None
update_distance_travelled() → float
update_speed() → None

class transit_lab_simmetro.simulation_engine.train.__init__.TrainPassengerManager
    Bases: object
    __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() → 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) → None
```

```
property normal_acceleration: float
```

```
property normal_acceleration_in_fps2
```

```
property normal_decceleration_in_fps2
```

```
property planning_distance: float
```

```
register_train(train: Train) → None
```

```
regulate_acceleration()
```

```
property train: Train
```

```
train_stopped_at_station(station: Station) → None
```

```
update_train_speed() → 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) → None
```

```
property normal_acceleration: float
```

```
property normal_acceleration_in_fps2
```

```
property normal_decceleration_in_fps2
```

```
property planning_distance: float
```

```
register_train(train: Train) → None
```

```
regulate_acceleration()
```

```
property train: Train
```

**train\_stopped\_at\_station**(*station*: Station) → None

**update\_desired\_speed**() → None

**update\_train\_speed**() → None

## 6.2 Data Flow

Explain how data flows through the system.





## 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.



## 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.



## API DOCUMENTATION

### 9.1 API Overview

High-level overview of the API.

### 9.2 Endpoints/Functions

Detailed descriptions of API endpoints/functions, including parameters, request/response formats, and examples.



---

**CHAPTER  
TEN**

---

**FAQS**

Address common questions and issues.





## TROUBLESHOOTING

### 11.1 Common Issues

List common issues and their solutions.

### 11.2 Getting Help

Information on where to ask questions or report issues.



## CHANGELOG

### 12.1 Version History

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



## **13.1 Glossary**

Definitions of terms used in the documentation.

## **13.2 Further Reading**

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



## 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>.



## PYTHON MODULE INDEX

### t

`transit_lab_simmetro.simulation_engine.infrastructure.__init__`,  
11  
`transit_lab_simmetro.simulation_engine.passenger.__init__`,  
15  
`transit_lab_simmetro.simulation_engine.simulation.simulation`,  
16  
`transit_lab_simmetro.simulation_engine.train.__init__`,  
16



## INDEX

## Symbols

[illegible]

## B

**Block** (class in `transit_lab_simmetro.simulation_engine.infrastructure.__init__.Term`  
`sit_lab_simmetro.simulation_engine.infrastructure.__init__.method`), 14

**block\_logger** (`transit_lab_simmetro.simulation_engine.infrastructure.__init__.Block`  
`attribute`), 11

**block\_with\_red\_signals\_in\_sight()** (`transit_lab_simmetro.simulation_engine.train.__init__.Train`  
`method`), 16

**board\_passengers()** (`transit_lab_simmetro.simulation_engine.train.__init__.TrainPassengerManager`  
`method`), 17

**board\_passengers\_based\_on\_destiations\_and\_probability()** (`transit_lab_simmetro.simulation_engine.infrastructure.__init__.Block`  
`method`), 14

**board\_passengers\_onto\_train()** (`transit_lab_simmetro.simulation_engine.infrastructure.__init__.Block`  
`method`), 14

**boarding\_time** (`transit_lab_simmetro.simulation_engine.passenger.__init__.Passenger`  
`property`), 15

**braking\_distance** (`transit_lab_simmetro.simulation_engine.train.__init__.TrainSpeedRegulator`  
`property`), 18

**braking\_distance** (`transit_lab_simmetro.simulation_engine.train.__init__.TrainSpeedRegulator`  
`property`), 18

**braking\_distance\_for\_station** (`transit_lab_simmetro.simulation_engine.train.__init__.TrainSpeedRegulator`  
`property`), 18

## C

**civil\_speed\_limit** (`transit_lab_simmetro.simulation_engine.infrastructure.__init__.Train`  
`property`), 12

**copy()** (`transit_lab_simmetro.simulation_engine.infrastructure.__init__.Train`  
`method`), 13

**current\_block** (`transit_lab_simmetro.simulation_engine.train.__init__.Train`  
`property`), 16

**current\_speed\_code** (`transit_lab_simmetro.simulation_engine.train.__init__.Train`  
`property`), 16

**current\_speed\_code()** (`transit_lab_simmetro.simulation_engine.infrastructure.__init__.AbstractBlock`  
`method`), 11

**current\_speed\_code()** (`transit_lab_simmetro.simulation_engine.infrastructure.__init__.Block`  
`method`), 11

**current\_speed\_code()** (`transit_lab_simmetro.simulation_engine.infrastructure.__init__.MovingBlock`  
`method`), 12

**current\_speed\_code()** (`transit_lab_simmetro.simulation_engine.infrastructure.__init__.OffSymptomaticBlockDecorator`  
`method`), 13

## D

**deactivate()** (`transit_lab_simmetro.simulation_engine.infrastructure.__init__.Block`  
`method`), 11

**deactivate()** (`transit_lab_simmetro.simulation_engine.infrastructure.__init__.Block`  
`method`), 12

**delete()** (`transit_lab_simmetro.simulation_engine.train.__init__.Train`  
`method`), 17

**denied\_boarding()** (`transit_lab_simmetro.simulation_engine.passenger.__init__.Passenger`  
`method`), 15

**DispatchingBlockDecorator** (class in `transit_lab_simmetro.simulation_engine.infrastructure.__init__`), 14

**DispatchingMovingBlockDecorator** (class in `transit_lab_simmetro.simulation_engine.infrastructure.__init__`), 14

**distance\_to\_next\_block** (`transit_lab_simmetro.simulation_engine.train.__init__.Train`  
`property`), 18

**distance\_to\_next\_block** (`transit_lab_simmetro.simulation_engine.train.__init__.Train`  
`property`), 18

**distance\_to\_next\_station** (`transit_lab_simmetro.simulation_engine.train.__init__.Train`  
`property`), 18

**distance\_traveled\_from\_the\_start\_of\_block()** (`transit_lab_simmetro.simulation_engine.train.__init__.Train`  
`method`), 17

**distance\_traveled\_from\_the\_start\_of\_block()** (`transit_lab_simmetro.simulation_engine.train.__init__.Train`  
`method`), 17

**DummyTrain** (class in `transit_lab_simmetro.simulation_engine.train.__init__`), 16

**DummyTrainDecorator** (class in `transit_lab_simmetro.simulation_engine.train.__init__`), 16

**DummyTrainDecorator** (class in `transit_lab_simmetro.simulation_engine.train.__init__`), 16

**entered\_symptomatic\_block()** (`transit_lab_simmetro.simulation_engine.train.__init__.TrainSpeedRegulator`  
`method`), 18

**entered\_symptomatic\_block()** (`transit_lab_simmetro.simulation_engine.train.__init__.TrainSpeedRegulator`  
`method`), 18

**entered\_symptomatic\_block()** (`transit_lab_simmetro.simulation_engine.train.__init__.TrainSpeedRegulator`  
`method`), 18

**entered\_symptomatic\_block()** (`transit_lab_simmetro.simulation_engine.train.__init__.TrainSpeedRegulator`  
`method`), 18

**first\_block\_after\_station** (`transit_lab_simmetro.simulation_engine.train.__init__.Train`  
`property`), 17

**from\_dict()** (`transit_lab_simmetro.simulation_engine.infrastructure.__init__.OffSymptomaticBlockDecorator`  
`method`), 13

**from\_dict()** (`transit_lab_simmetro.simulation_engine.infrastructure.__init__.OffSymptomaticBlockDecorator`  
`method`), 13

## G

G		get_lambda_bar()	(transit_lab_simmetro.simulation_engine.passenger.__init__.ArrivalRule method), 15
generate_and_add_passengers()	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.Station method), 14	get_lambda_for_station()	(transit_lab_simmetro.simulation_engine.passenger.__init__.ArrivalRule method), 15
generate_passengers()	(transit_lab_simmetro.simulation_engine.passenger.__init__.PassengerGenerator method), 15	get_next_station()	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.Path method), 13
generate_train_id()	(transit_lab_simmetro.simulation_engine.train.__init__.Train class method), 17	get_next_station()	(transit_lab_simmetro.simulation_engine.train.__init__.Train method), 17
get_a_i()	(transit_lab_simmetro.simulation_engine.passenger.__init__.ArrivalRule method), 15	get_next_train()	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.Path method), 13
get_all_destination_stops_for_origin()	(transit_lab_simmetro.simulation_engine.passenger.__init__.ArrivalRule method), 15	get_p_for_station()	(transit_lab_simmetro.simulation_engine.passenger.__init__.ArrivalRule method), 15
get_all_destination_stops_for_origin_and_direction()	(transit_lab_simmetro.simulation_engine.passenger.__init__.ArrivalRule method), 15	get_previous_train()	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.Path method), 13
get_all_previous_stops_for_station_and_direction()	(transit_lab_simmetro.simulation_engine.passenger.__init__.ArrivalRule method), 15	get_smoothed_rate()	(transit_lab_simmetro.simulation_engine.passenger.__init__.ArrivalRule method), 15
get_all_stops_ahead()	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.Path method), 13	get_speed_code()	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.MovingBlockControl method), 13
get_all_stops_ahead()	(transit_lab_simmetro.simulation_engine.train.__init__.Train method), 17	get_total_length()	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.Path method), 13
get_all_stops_ahead_which_are_served()	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.Path method), 13	get_total_travelled_distance()	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.Path method), 13
get_block_by_id()	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.Path method), 13	id_of_last_train	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.Block property), 11
get_block_by_id()	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.SignalControlCenter method), 14	is_inspected()	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.Path method), 13
get_block_index_by_id()	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.Path method), 13	is_it_clear_to_dispatch()	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.DispatchingBlockControl method), 12
get_current_hour()	(transit_lab_simmetro.simulation_engine.simulation.simulation.Simulation method), 16	is_it_clear_to_dispatch()	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.DispatchingBlockControl method), 12
get_distance_to_next_train()	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.MovingBlockControl method), 13	is_occupied	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.Path property), 11
get_distance_to_the_next_station()	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.Path method), 13	is_occupied	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.Path property), 11
get_door_metrics()	(transit_lab_simmetro.simulation_engine.train.__init__.Train method), 18	is_occupied	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.Path property), 11
get_dwelling_time()	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.Station method), 14	is_occupied	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.Path property), 11
		is_occupied_by()	(transit_lab_simmetro.simulation_engine.infrastructure.__init__.Path method), 13

`sit_lab_simmetro.simulation_engine.infrastructure.make_off_block_decorator` (transit\_lab\_simmetro.simulation\_engine.infrastructure.\_\_init\_\_.Path method), 11  
`is_occupied_by()` (transit\_lab\_simmetro.simulation\_engine.infrastructure.\_\_init\_\_.Block method), 11  
`is_occupied_by()` (transit\_lab\_simmetro.simulation\_engine.infrastructure.\_\_init\_\_.MovingBlock method), 12  
`is_short_turn()` (transit\_lab\_simmetro.simulation\_engine.infrastructure.\_\_init\_\_.Path method), 13  
`is_short_turned_at_this_station()` (transit\_lab\_simmetro.simulation\_engine.infrastructure.\_\_init\_\_.MovingBlock method), 13  
`is_southbound_trip()` (transit\_lab\_simmetro.simulation\_engine.passenger.\_\_init\_\_.MovingBlockControl method), 15  
`is_weekday()` (transit\_lab\_simmetro.simulation\_engine.simulation.Simulation method), 16

## N

`needed_braking_distance()` (transit\_lab\_simmetro.simulation\_engine.infrastructure.\_\_init\_\_.MovingBlock method), 13  
`next_block()` (transit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.Train property), 17  
`next_train()` (transit\_lab\_simmetro.simulation\_engine.infrastructure.\_\_init\_\_.Block method), 12  
`normal_acceleration` (transit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.TrainSpeedReducer property), 18  
`normal_acceleration` (transit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.DispatchingMovingBlockDecorator property), 18  
`normal_acceleration_in_fps2` (transit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.TrainSpeedReducer property), 18  
`normal_acceleration_in_fps2` (transit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.Station property), 18  
`normal_deceleration_in_fps2` (transit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.TrainSpeedReducer property), 18  
`normal_deceleration_in_fps2` (transit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.Train property), 17  
`log()` (transit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.DummyTrain property), 16  
`log()` (transit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.Train property), 16  
`notify_observers()` (transit\_lab\_simmetro.simulation\_engine.infrastructure.\_\_init\_\_.Block method), 11

## M

`make_dispatching_block()` (transit\_lab\_simmetro.simulation\_engine.infrastructure.\_\_init\_\_.Path method), 13

## O

`OffScanSymptomaticBlockDecorator` (class in transit\_lab\_simmetro.simulation\_engine.infrastructure.\_\_init\_\_), 13





SimulationContext (class in tran- TrainPassengerManager (class in tran-  
 sit\_lab\_simmetro.simulation\_engine.simulation.simulation), sit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_),  
 16 17  
 SlowZone (class in tran- TrainSpeedRegulator (class in tran-  
 sit\_lab\_simmetro.simulation\_engine.infrastructure.\_\_init\_\_), sit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_),  
 14 18  
 sort\_stations\_by\_direction() (tran- TrainSpeedRegulatorCTA (class in tran-  
 sit\_lab\_simmetro.simulation\_engine.passenger.\_\_init\_\_.ArrivalRate, sit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_),  
 method), 15 18  
 sorted\_current\_train\_list (tran- transit\_lab\_simmetro.simulation\_engine.infrastructure.\_\_init\_\_  
 sit\_lab\_simmetro.simulation\_engine.infrastructure.\_\_init\_\_.MovingBlock  
 property), 12 transit\_lab\_simmetro.simulation\_engine.passenger.\_\_init\_\_  
 speed\_in\_fps (transit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.WaitingTime  
 property), 17 transit\_lab\_simmetro.simulation\_engine.simulation.simulation  
 Station (class in tran- module, 16  
 sit\_lab\_simmetro.simulation\_engine.infrastructure.transit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_  
 14 module, 16  
 station\_logger (tran- travel\_time (transit\_lab\_simmetro.simulation\_engine.passenger.\_\_init\_\_  
 sit\_lab\_simmetro.simulation\_engine.infrastructure.\_\_init\_\_.Station), 15  
 attribute), 14

## U

T update() (transit\_lab\_simmetro.simulation\_engine.infrastructure.\_\_init\_\_  
 method), 14  
 Terminal (class in tran- update()) (transit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.Train  
 14 method), 17  
 time\_step (transit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.UpdateBlock) (tran-  
 property), 17 sit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.Train  
 to\_dict() (transit\_lab\_simmetro.simulation\_engine.infrastructure.\_\_init\_\_.SlowZone  
 method), 14 update\_desired\_speed() (tran-  
 total\_passengers (tran- sit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.TrainSpeedRe  
 sit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.TrainPassengerManager  
 property), 18 update\_distance\_travelled() (tran-  
 total\_travelled\_distance (tran- sit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.Train  
 sit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.Train method), 17  
 property), 17 update\_speed() (tran-  
 total\_travelled\_distance\_from\_dispatch (tran- sit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.Train  
 sit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.Train method), 17  
 property), 17 update\_train\_speed() (tran-  
 Train (class in tran- sit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.TrainSpeedRe  
 sit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_), method), 18  
 16 update\_train\_speed() (tran-  
 train (transit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.TrainSpeedRegulator, sit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.TrainSpeedRe  
 property), 18 method), 19  
 train (transit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.TrainSpeedRegulatorCTA  
 property), 18

## W

train\_logger (transit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.Train  
 attribute), 17 waiting\_time (transit\_lab\_simmetro.simulation\_engine.passenger.\_\_init\_\_  
 property), 15  
 train\_stopped\_at\_station() (tran-  
 sit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.TrainSpeedRegulator  
 method), 18  
 train\_stopped\_at\_station() (tran-  
 sit\_lab\_simmetro.simulation\_engine.train.\_\_init\_\_.TrainSpeedRegulatorCTA  
 method), 18