# MIT RailSim

Mojtaba Yousefi, Jonah Stadtmauer

Apr 21, 2024

# CONTENTS:

**1 Introduction** **1**
1.1 Project Overview . . . . . 1
1.2 Features . . . . . 1
1.3 License . . . . . 2

**2 Setup Guide** **3**

**3 Getting Started** **5**
3.1 Installation Guide . . . . . 5
3.2 Quick Start . . . . . 5

**4 Usage** **7**
4.1 Basic Usage . . . . . 7
4.2 Advanced Usage . . . . . 7

**5 Configuration** **9**
5.1 Configuration Options . . . . . 9
5.2 Default Configuration . . . . . 9

**6 Architecture** **11**
6.1 Components Overview . . . . . 11
6.2 Data Flow . . . . . 11

**7 Development** **13**
7.1 Development Environment Setup . . . . . 13
7.2 Build Instructions . . . . . 13
7.3 Testing . . . . . 13

**8 Contribution Guidelines** **15**
8.1 How to Contribute . . . . . 15
8.2 Community Guidelines . . . . . 15

**9 API Documentation** **17**
9.1 API Overview . . . . . 17
9.2 Endpoints/Functions . . . . . 17

**10 Faqs** **19**

**11 Troubleshooting** **21**
11.1 Common Issues . . . . . 21
11.2 Getting Help . . . . . 21

12 Changelog 23
12.1 Version History . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

13 Appendix 25
13.1 Glossary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25
13.2 Further Reading . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

14 Indices and tables 27

Bibliography 29

CHAPTER

ONE

# INTRODUCTION

## 1.1 Project Overview

MIT RailSim 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 MIT RailSim, 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 RailSim 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.

CHAPTER

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.

CHAPTER

THREE

# GETTING STARTED

## 3.1 Installation Guide

Provide step-by-step instructions for installing the project, including prerequisites.

## 3.2 Quick Start

A quick guide to getting a simple example up and running.

CHAPTER

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.

CHAPTER

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.

CHAPTER

SIX

# ARCHITECTURE

## 6.1 Components Overview

Describe the main components of the project.

## 6.2 Data Flow

Explain how data flows through the system.

CHAPTER

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.

CHAPTER

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

# 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.

CHAPTER

ELEVEN

# TROUBLESHOOTING

## 11.1 Common Issues

List common issues and their solutions.

## 11.2 Getting Help

Information on where to ask questions or report issues.

CHAPTER

TWELVE

# CHANGELOG

## 12.1 Version History

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

CHAPTER

THIRTEEN

# APPENDIX

## 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.

# CHAPTER FOURTEEN

# 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.