Modelling Transit System Station Occupancy using Numerical Simulation

Aaron W. Langham
Electrical Engineering & Computer Science
alangham@mit.edu

You Xuan Thung
Computational Science & Engineering
thungyx@mit.edu

Edvard Ronglan
Mechanical Engineering
eronglan@mit.edu

***Abstract*—Many urban areas are attempting to shift people's transit behavior away from cars and towards sustainable modes such as public transit. To effectively do so, there is a need for simulation of transit performance under varying conditions so that service can be planned effectively. One important facet of this is modelling the crowdedness of stations under various conditions. This work presents an approach to estimate the amount of crowding at each transit station using knowledge of aggregate passenger flows in and out of the system and transit times between stations.**

I. Introduction & Motivations

According to a report released in 2014 by Moovit, a public transit app, people in major U.S. cities wait approximately 40 minutes each day for public transit [1]. This is a major problem as it takes time away from other important daily activities such as work and personal life. It also leads to unnecessary stress since waiting times often vary across the day [2] and it is therefore difficult to plan the amount of time necessary to get between two places.

A key factor contributing to waiting time is the number of people at a station since it directly affects the time it takes to enter the transportation vehicle [2]. If there are too many people at the station, there can also be a problem where not everyone fits in the transportation vehicle, and some commuters may have to wait for the next vehicle to arrive. Public transit agencies and local governments have an interest in solving this problem in order to retain riders and reduce the number of single-occupancy vehicles on the road. Our goal is to model this problem for a train system and simulate how the average volume of passengers at each station changes in response to passenger flow in and out of stations. Although many transit simulation tools effectively simulate arrivals and departures as discrete events [3], this system focuses on short-term average behavior to pose the system using ordinary differential equations. By using this tool, public transit authorities would be able to judge how to allocate transit capacity during different times, such as peak commute periods and weekends.

II. Problem formulation

A technical description of the problem is given using a dynamical state space framework [4]. This allows the system to be expressed by the differential equation $dx/dt = f(x, p, u)$, where x, $f(\cdot)$, p, and u are given below.

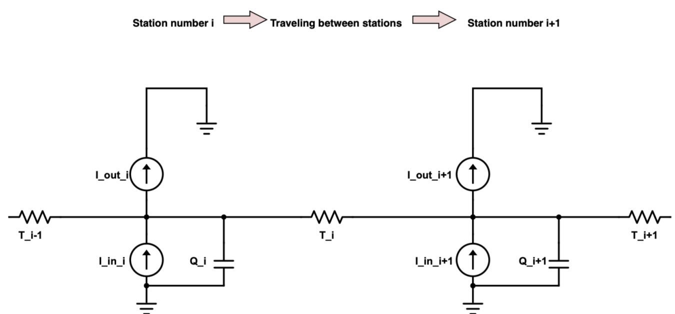

Fig. 1. Analogous circuit model for two nodes of the proposed system.

- A major assumption in this system is that we can use the short-term moving average of the instantaneous number of passengers at each station for the state vector, since this simulation is not designed to model discrete events. This short-term average is obtained by integrating the instantaneous value over some short period and dividing by that period length [5]. The state of the system is given by the vector x, which is composed of the short-term averages of the number of passengers at each station:

$$x = [Q_1, Q_2, \dots, Q_N]^\top. \tag{1}$$

- The parameters of the system are given by the vector p, which contains the transit time between each connected station:

$$p = [T_1, T_2, \dots, T_{N-1}]^\top. \tag{2}$$

  T_i represents the transit time between nodes i and $i+1$ This vector is of length $N-1$, because there is no further connection after station N. We obtained these parameters from Google Maps.
- The sources in the system are given by the vector u, which contains the net passenger flow I into each station from outside the system, i.e. the input flow minus the output flow:

$$u = [I_1, I_2, \dots, I_N]^\top. \tag{3}$$

- The final set of equations that describe the system are given by the following:

$$\begin{aligned}\frac{dx_1}{dt} &= u_1 - I_{max}\tanh\left(\frac{x_1}{p_1 I_{max}}\right)\\ \frac{dx_2}{dt} &= u_2 + I_{max}\tanh\left(\frac{x_1}{p_1 I_{max}}\right)\\ &\quad - I_{max}\tanh\left(\frac{x_2}{p_2 I_{max}}\right)\\ &\vdots\\ \frac{dx_N}{dt} &= u_N + I_{max}\tanh\left(\frac{x_{N-1}}{p_{N-1} I_{max}}\right)\end{aligned} \tag{4}$$

where hyperbolic tangent (tanh) is used to approximate the other nonlinearity in the system: the maximum capacity of a transit line. Since only so many people can fit on a train, this nonlinearity prevents the value of $I_{max}\tanh(x_i/(p_i I_{max}))$ from ever exceeding I_{max}. We derive these equations by assuming that the passenger flow into a station is proportional to the number of people at the previous station and inversely proportional to the travel time between the two stations. I_{max} is approximated as the number of passengers that can fit in one trip multiplied by the number of trips per hour. We estimate this as $18,000$ with information from the Massachusetts Bay Transportation Authority (MBTA) [6].

- The quantities of interest in this system are the short-term averages of the number of people at each transit station. Thus, the output vector y is simply equal to the state vector x:

$$y = x. \tag{5}$$

- For simulation of multiple transit lines, we model each line as a separate system, and include the passenger flow between lines at transfer stations in those stations' u vectors.

III. Fundamental Numerical Methods

The Forward Euler method and the Trapezoidal method (with both fixed and dynamic time-stepping) are implemented to evaluate the time dynamics of the system [4]. The pseudocode for these methods are given in Algorithm 1 and Algorithm 2.

Algorithm 1 Forward Euler Integrator

```
function FORWARDEULER(f(·), x(t_0), p, u, t_0, T, δt)
    l ← 0
    x^l ← x(t_0)
    t_stamps ← [0]
    x = [x(t_0)]
    t ← t_0
    while t < T − δt do
        l ← l + 1
        t_l ← t_{l−1} + δt
        x_l = x_{l−1} + δt ∗ f(x_{l−1}, p, u)
        x ← x + [x_l]
        t_stamps ← t_stamps + [t_l]
    return t_stamps, x
```

Algorithm 2 Trapezoidal Method

```
function TRAPEZOIDAL(f(·), p, u(·), x(t_0), t_0, T, δt)
    l ← 0
    x^l ← x(t_0), x = [x(t_0)]
    t_stamps ← [0], t ← t_0
    while t < T − δt do
        l ← l + 1
        t_l ← t_{l−1} + δt
        t_stamps ← t_stamps + [t_l]
        k ← 0 (Newton iteration index)
        x^{l,k} ← x̂^{l−1} (Use last point as initial guess)
        γ ← x^{l−1} − (Δt/2) f(x^{l−1}, u(t_{l−1}))
        F_trap ← x^{l,k} − (Δt/2) f(x^{l,k}, u(t_l)) − γ
        while Newton not converged do
            J_trap ← I − (Δt/2) J_f(x^{l,k}, u(t_l))
            Solve J_trap Δx = −F_trap
            x^{l,k+1} ← x^{l,k} + Δx
            k ← k + 1
            F_trap ← x^{l,k} − (Δt/2) f(x^{l,k}, u(t_l)) − γ
        if dynamic time-stepping then
            if ||x^l − x^{l−1}||_∞ / δt_l > Max Slope then
                t_l ← t_l − δt, l ← l − 1, δt_l ← 0.8 · δt_l
            else if ||x^l − x^{l−1}||_∞ / δt_l < Min Slope then
                t_l ← t_l − δt_l, l = l − 1, δt_l ← 1.2 · δt_l
        x^l ← x^{l,k}
        x ← x + [x^l]
    return t_stamps, x
```

IV. The Technical Challenge

Simulating our dynamic non-linear system with a smaller time step would yield more accurate results but this comes at the expense of increased computational time. The technical challenge is to explore different ODE integrators and find the balance between accuracy and speed. The system examined in this work was chosen to be the Red Line (in the direction of Braintree to Alewife, not including the Ashmont branch) and the Orange Line (in the direction of Forest Hills to Oak Grove) of the MBTA subway system. Specifically, we considered as inputs the ridership pattern associated with the evening rush hour on a weekday. Using public MBTA rail ridership data [7], values for the input vector u were generated. We conducted multiple simulations with Forward Euler and the Trapezoidal methods using different time steps, with $I_{max} = 18,000$.

We ran Forward Euler using $\Delta t = 10^{-6}$ to get an approximation of the "true" solution, $x_{true}(t)$. We then take $||x(t) - x_{true}(t)||_\infty$ as the measure of accuracy i.e. the maximum deviation in the number of passengers at a station at any point in time. We present our results in Table I.

We find that the Trapezoidal method is stable for the time-steps we experimented with while Forward Euler becomes

Method	Δt	Run-time (s)	Error (No.)
FE	10^{-5}	70.4	0.0229
FE	10^{-4}	13.7	0.229
FE	5×10^{-4}	4.91	0.492
FE	10^{-3}	1.31	2.31
FE	10^{-2}	0.10	24.5
TM	10^{-3}	38.74	0.0786
TM	10^{-2}	8.16	0.561
TM	10^{-1}	0.63	38.9
TM (Adaptive)	10^{-3}	9.98	68.3
TM (Adaptive)	10^{-2}	5.28	72.9
TM (Adaptive)	10^{-1}	2.52	145

TABLE I
COMPARISON OF ODE INTEGRATORS

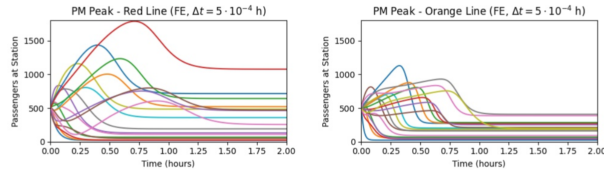

Fig. 2. Simulation of dynamics of x for each station using FE (PM Peak)

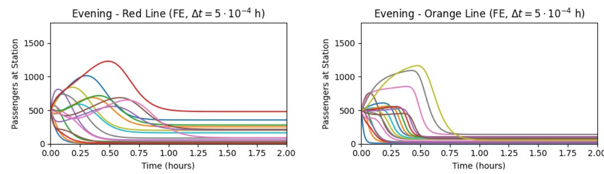

Fig. 3. Simulation of dynamics of x for each station using FE (Evening)

unstable for time steps larger than 10^{-2}. At 10^{-2}, the error is given as 24.5. This is slightly larger than the tolerable error, which we set as ± 10. Since the Trapezoidal method exhibits a quadratic decrease in the maximum error as Δt decreases, we experimented with $\Delta t = 10^{-1}$ for the Trapezoidal method but found that the error, although on the same order as Forward Euler for $\Delta t = 10^{-2}$, is slightly larger. Furthermore, the run-time of 0.63 seconds is much slower than that of Forward Euler at $\Delta t = 10^{-2}$. In an attempt to improve the run-time of the Trapezoidal method, we introduced adaptive time stepping (outlined in Algorithm 2) but found that there is only an improvement in run-time for an initial Δt of 10^{-3}. However, for the range of initial Δt tested, we find that the errors are well above our tolerance.

Trapezoidal method at $\Delta t = 10^{-2}$ achieves comparable performance with Forward Euler at $\Delta t = 10^{-4}$. Comparing their performance suggests a trade-off between run-time and error. However, we find that if we decrease the timestep for Forward Euler just a little more (to 5×10^{-4}), both the run-time and error are better than those of the Trapezoidal schemes at $\Delta t = 10^{-2}$. Therefore, we decided to use the Forward Euler method at $\Delta t = 5 \times 10^{-4}$ for our subsequent simulations.

V. RESULTS

With $\Delta t = 5 \times 10^{-4}$, we simulated the dynamics of our system (Orange Line and truncated Red Line of the MBTA) to observe the crowdedness of stations and identify stations that may be especially overcrowded. The dynamics of simulated passenger levels at each station for the evening peak period (PM peak) is shown in Figure 2. A steady-state visualization of the system is given in Figure 4. As a comparison, we also ran the simulation for evening, off-peak, ridership data; these results are presented in Figure 3 and Figure 5.

To examine the effectiveness of our simulation tool in modeling the effects of policy changes, we re-ran the simulations after changing the frequency of train arrivals. We present the effects of decreasing the frequency of train arrivals from 20 trips per hour to 15 trips per hour on the steady state in Figure 6 and Figure 7.

VI. TECHNICAL DISCUSSION

For the PM peak simulation, stations between the downtown area and Cambridge are quite crowded after an hour. In fact, the stations around Central and Kendall squares are the most crowded after peak hours, which is somewhat in line with our expectations given that this part of Cambridge is densely packed with both residences and offices.

For the evening off-peak simulation, we find that the passenger levels drop compared to the PM peak simulation (from a maximum of around 1500 to a maximum of around 500). This is in line with our expectations since overall usage of the subway in the evening off-peak hours should be lower than in the evening peak hours.

We used a value of 1200 as the maximum number of passengers that can fit in one trip. When we decrease the frequency of trains (from 20 to 15 per hour), we find that the relative crowdedness of stations remains the same but the number of passengers at each station generally increases across the system. From a maximum of around 1000 in PM peak, we now see a maximum of around 1500. In the evening, the maximum number of passengers increases too, albeit more marginally. This aligns with our expectation that increasing train frequency can reduce congestion in stations. While we demonstrate the effectiveness of our simulation in modeling this simplified system, future work would involve scaling it to more complex transportation systems with more connections and transfers, and modeling more sophisticated policies, for example, varying frequency by time of the day or endogenously.

VII. ETHICS & LIMITATIONS

The impact of making changes to public transit is far-reaching and we need to be clear about the potential ramifications of doing so. While we focus on modelling a train system, we recognize that commuters can also take buses, cabs, or drive on their own. Omitting these alternative modes of transportation may mischaracterize commuters' behaviour. In the same vein, given the computational constraints in

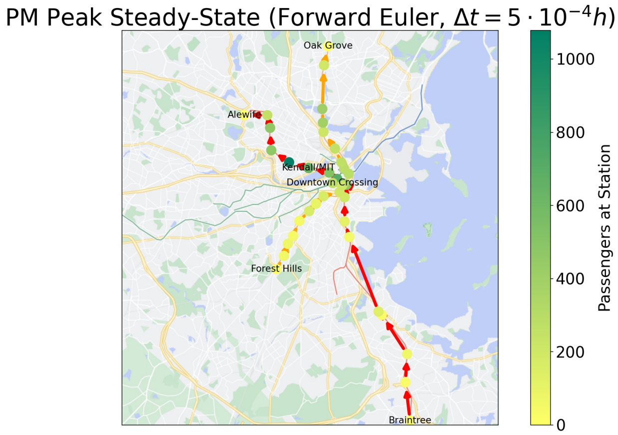

Fig. 4. Steady-state visualization for 20 trips per hour.

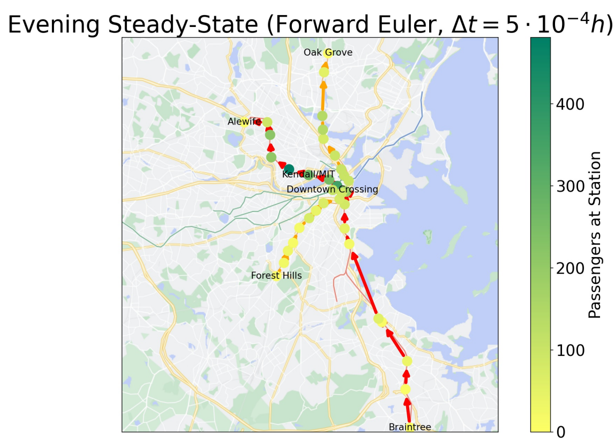

Fig. 5. Steady-state visualization for 20 trips per hour.

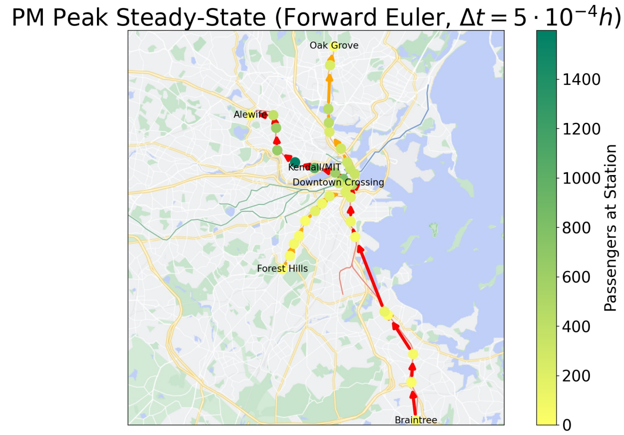

Fig. 6. Steady-state visualization for 15 trips per hour.

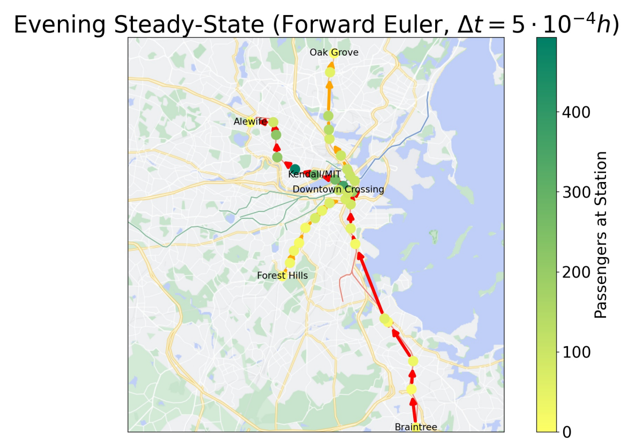

Fig. 7. Steady-state visualization for 15 trips per hour.

simulating a large system with non-linear dynamics within a reasonable amount of time, our simulation involves several approximations and allows errors up to the order of 10^0.

In addition to technical concerns, we also need to consider potential ethical issues in implementation. For one, our model relies on real-world data about the location of people and their travel information to generate reliable simulations. If such information comes from applications which do not ask users for consent directly, then using such information compromises individuals' right to privacy.

Beyond using the simulations to improve transit capacity planning, public transit operators could also use them to dynamically set the price of a ticket by adding surcharges as a function of the number of people at the stations [8]. Dynamic pricing could lower waiting time in peak hours by smoothing demand over time and over multiple stations, as commuters adapt their behavior to the ticket price, and allowing for increased transportation capacity. While this can positively affect the commuters in rush hours, they would also need to pay more compared to commuters in non-rush hours. Furthermore, if prices are too volatile, the uncertainty could lead to commuters in rush hours paying a price they did not expect or cannot afford. A constantly-changing fare may also leave some users stranded if they cannot afford a return trip. Alternatively, if the model estimates a higher number of people at the station than in reality, which would lead to these people paying more than is required to reduce congestion.

Ultimately, transit operators need to be judicious in how they use the information from the simulations, keeping in mind the assumptions involved and the ethical concerns of making changes to the status quo.

VIII. Conclusions

In this study, we simulated the short-term average of passenger flows in the two lines of the MBTA system with Forward Euler and both fixed-time and dynamic time-step Trapezoidal methods. We find that the Forward Euler method achieves higher faster run-times for our desired accuracy. Our simulations serve as a successful proof-of-concept in characterizing the dynamics and steady-state of passenger levels at different times of the day, and demonstrating the effects of transit capacity allocation.

References

[1] Metro Magazine, "U.S. commuters wait approximately 40 mins. a day for public transit", 2014: https://www.metro-magazine.com/10037552/u-s-commuters-wait-approximately-40-mins-a-day-for-public-transit

[2] Mairead Cantwell *et al.*, "Examining the Factors that Impact Public Transport Commuting Satisfaction" Journal of Public Transportation, 12(2): 1-21.

[3] Baichuan Mo *et al.*, "Calibrating Path Choices and Train Capacities for Urban Rail Transit Simulation Models Using Smart Card and Train Movement Data" Journal of Advanced Transportation, 2021: https://www.hindawi.com/journals/jat/2021/5597130/

[4] Luca Daniel, "2.091/6.336/16.910 Introduction to Numerical Simulation" MIT, Graduate Course, Fall 2018.

[5] R. J. Hyndman, "Moving Averages" International Encyclopedia of Statistical Science, 2011: https://doi.org/10.1007/978-3-642-04898-2_380

[6] Massachusetts Bay Transportation Authority, "Red Line Customer Capacity Update", 2016: https://willbrownsberger.com/wp-content/uploads/2016/09/Red-Line-Capacity-Constraints-Sept-19-update-3.pdf

[7] Massachusetts Bay Transportation Authority, "MBTA Rail Ridership by Time Period, Season, Route/Line, and Stop", 2020: https://mbta-massdot.opendata.arcgis.com/datasets/mbta-rail-ridership-by-time-period-season-route-line-and-stop/explore

[8] Metro Magazine, "Dynamic Pricing Can Help Transport Charge Accurately For Services", 2019: https://www.metro-magazine.com/10007197/dynamic-pricing-can-help-transport-charge-accurately-for-services

[9] Eric Jaffe, "Far Beyond Rush Hour: The Incredible Rise of Off-Peak Public Transportation" Bloomberg CityLab, 2014: https://www.bloomberg.com/news/articles/2014-02-06/far-beyond-rush-hour-the-incredible-rise-of-off-peak-public-transportation