Prediction Decision Making - Railway Simulator

Prof. Narayanaswamy N S
Hari Charan Korrapati CS20B086

January 31, 2023

1 State space definition

At any instance (a discrete time interval), the state or configuration of the railway network can be expressed in terms of its state variables which include the following

static variables:
- **train data:** train no, train length, timetable information, max speed, etc
- **track data:** distance between stations and elevation above sea level
- **station data:** no of platforms, location

dynamic variables:
- velocity
- location i.e distance from source (or distance from a particular station in the train route)

control variables:
- **signal lights data:** location, and color (red/green/amber) of each signal.

The static varibles comprise of constant data that does not change over the period of many simulations and predictions unless the Indian railways decide to revise time tables, speed limits set on trains etc. So, we store them in a database (will be designed by Yuvan) in such a way that our desired queries will be answered quickly (like fetching neighboring station etc).

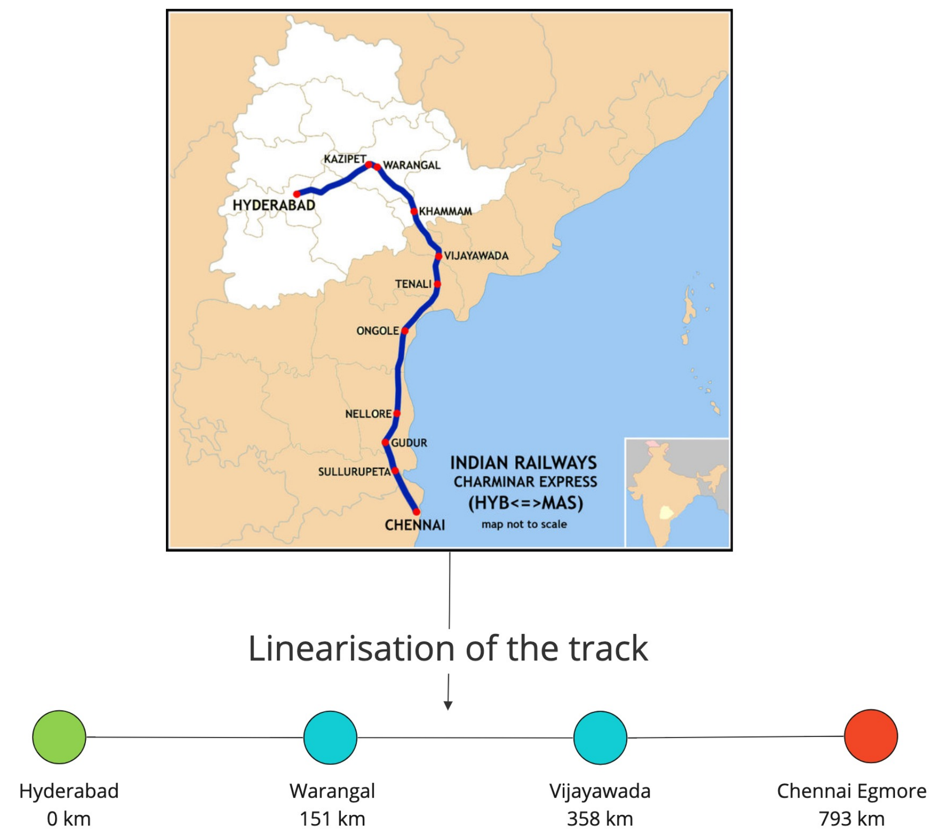
The dynamic and control variables velocity, location (distance from source) and signal colours are the ones that are being rapidly accessed and updated at every discrete time interval, so we might use a NoSQL key-value representation or a hash table to store them for easy access and update.

Here, the velocity of each train is the only variable that is independent i.e can be controlled by the controller. The location of the train is a dependent variable that changes according to the changes in speed. The means through which we control the velocity are signals.

2 Linearizing the tracks

Mathematically we view the tracks as straight lines. Essentially, the track between adjacent stations is considered as a straight line and the distance (not displacement) is considered. We are just ignoring the turns on the track for now. Maybe, we'll have to account for it later as it brings up some speed constraints and might add up a significant amount of time (like 5 minutes) in between each station. We are trying to see the network as a graph where edge weights represent distances.

For example, one possible track between Hyderabad and Madras can be viewed this way



3 Expressing location

The location of a train is represented in terms of the distance it has traveled from the last node (station) it has visited and the station towards which it is going.

Location of stations as well can be represented like this. We can take a few root stations and deduce the locations of remaining stations similarly. Also, we assume that station has sufficient platforms for trains to cross. This is a safe assumption because usually every station has >= 2 platforms.

To do: From the train routes (scrapped from the internet and railway API) fetch the neighboring stations for each station and their corresponding distances between them and store it as an adjacency list for convenience. (data structure used frequently to store graphs)

Also, fetch the signal location information. usual distribution of signals is clustered near the stations and sparse in between stations, we cannot assume uniform distribution.

To be discussed: should we consider all trains as identical or make a few classifications like, goods train vs passenger train (super fast, normal and slow - passenger) ?

4 Model Predictive Control

We have finished modelling the state space of the system. The network simulator will simulate the system and to control it we use a strategy called MPC. Model Predictive Control (MPC) is a control strategy that uses a mathematical model of a process to predict future behavior and optimize control actions. It is usually used in production plants to achieve a desired set point. Here the production plant is the entire railway network and the desired set point is to minimize overall delays.

The basic idea behind MPC is to use a model of the process to predict how the process will behave over a certain time horizon. This model is then used to optimize a set of control actions that will be applied to the process in order to achieve a desired setpoint. The control actions are chosen to minimize a cost function that represents the deviation of the process from the desired setpoint.

prediction horizon:

The number of discrete time intervals (steps) the control systems looks into the future to predict an optimal action. It is usually represented by k. The value of k needs to be decided based on the computational constraints. Also, too large k is counter productive because it will simply deviate from the actual thing that will happen (as its a prediction, not determinism)

control horizon:

The number of steps the control system acts into the future. Usually, control horizon is just 1-2 steps (much less than prediction horizon).

cost function:

MPC allows a quadratic cost function. so we can take advantage of this. cost is proportional to square of delay. This makes more sense because a train arriving on time and another arriving more than an hour late is not ok. It is better to have both trains coming 30mins late (just a hypothetical case).

The cost needs to be calculated not only when a train is at the station but at each discrete interval. We define expected time of arrivals at each discrete and interval and calculate delays against it. This can be divided by evenly distributing time between 2 stations in its scedule.

For example, train T is sceduled at station A at 10:00 AM and B at 11:00AM (station that comes after A). if the distance is 30km, then ideally the train should be going at a constant speed of 30 kmph, we calculate delays at each point accordingly.

Constraints: The only variable that is controlled by MPC is speed of each train. We assume that change in speed is almost instantaneous (maybe takes ¡30s). The constraint on speed change is it can be only ± 20 or ± 30 at each signal (exact number to be decided). Train going from 80 to 0 or 0 to 80 at one signal is not allowed. This constraint can be mathematically represented in MPC. It can also be represented as a percentage change like +- 10 percent. MPC allows this.

Time interval: We have to decide upon the smallest discerete time interval value. Like after how many time units will we look at tht next state. Practically it can be 5-10 mins.

5 Conditions for MPC:

The basic algorithm is a **Linear time-invariant MPC**. For this we need the following criteria to be met:

- **linear system**, i.e the output variables are linearly dependent on the input variables
- **Linear constraints**
- **Quadratic cost function**

6 Predicting next state

Traditional MPC requires a linear system to predict the future states. So, we can use a linear machine learning model like Linear regression to calculate delays. If other non linear techniques like Decision trees, Random forest are used, then we have to use linearisation techiniques and use advanced MPC algorithms.

7 Predicting delays

"past repeats itself"

Railway networks constitute a heavily intertwined system as numerous train vehicles share a limited infrastructure of tracks. A single train running behind its schedule is likely to hinder other trains using the same tracks leading to cascading consequences. Due to this high degree of inter-dependency in railway operations, initial delays often propagate to other trains in the railway network

data driven vs event driven approaches. It is ideal to choose data driven

multi-variate linear regression (suitable for MPC)
decision trees
random forest
nueral network ?
support vector machine

problem of overfitting

factors:
- current delay of each train
- delay at the origin
- weather conditions (breaking down into regions)
- day of the week
- delay of the previous days (for ex last 7 days)
- delay on the same date previous year (and neighbouring dates)
- delay of surrounding trains (running in similar routes and time)

challenges to be discussed with Ravi sir

what algorithm are we using to predict ?

should we use a linear algoritihm or a non-linear one and linearize it for the sake of MPC ? which one is efficient

8 Advanced MPC techniques:

What if the system is non-linear ?

we have to use either of the following

adaptive MPC

gain sceduled MPC

Non linear MPC

9 Interaction with network simulator

can the simulator account for weather conditions as rains / fogg etc do actually have a significant impact of railway delays.

can the simulator account for errors in wait times at stations like maintainance etc.

10 Data collection:

what kind of data is required and how do we obtain it ?

Train details - Indian railway's API (we can use python requests module)

Train delays - Indian railways website

weather data - OpenWeatherMap API

11 Implementing MPC

MatLab has some excellent libraries to work with MPC, but very little documentation is available on that. Have to find for sources and study them.

12 discrete distance intervals or time intervals ?

13 thoughts on research papers

14 Challenges:

- Tracks can be unidirectional or bidirectional, How do we represent this mathematically in MPC

- How do we facilitate crossing of trains near stations in MPC

- Should we consider discrete time intervals or discrete distance intervals

- How do we tackle asynchronous nature of the network

- How to solve MPC for multiple trains

15 work done so far:

- Finised modelling the state space of the system.

- Finished reading "A review of train delay prediction approaches" reseach paper that discusses vairious event driven and data driven techniques to predict delays along with their accuracy.

- Finished reading "Prediction of Train Delay in Indian Railways through Machine Learning Techniques" a paper on predicting train delays specifically for indian railways.

- Learnt non linear supervised learning algorithms that are more accurate than linear regression like Decision trees, Random forest.

- Skimmed through a few scholarly articles, but weren't really helpful, either they had similar information or were unrelated.

16 To do:

- Some papers have suggested advanced algorithms like Ada-Boost, XG-boost for predicting delays, learn them and try to see if they can be linearised to use for MPC.

- Can we use nueral networks? Most papers showed that neural networks gave the most accurate results, but it is a non linear system. Think if it can be used, discuss with Ravi sir.

- mathematically express the cost function

- Model the linear system for MPC.

- Come up with an exhaustive list of factors that affect the prediction of delays. (few already listed for reference). A very large number of factors are allowed for multi-variate linear regression.