Modeling Simulation and Optimisation Project

Sobil Dalal
School of Computing
National College of Ireland
Dublin, Ireland
x19148496@student.ncirl.ie

***Abstract*—In this report, a simulation and optimization study on HS2 railway line from London Old Oak Commons to Birmingham Interchange has been described along with system validations using Python programming and SimPY library. The input data is dynamically created using basic formulas of physics like – speed, acceleration, distance, and time. In the HS2 line signaling blocks are equidistant between the station and stations are also considered as a resource; such as only one train can be present at a time. Variability in the dwell time and drive time has been introduced to achieve operational conditions. To solve this problem objects for the network nodes and network have been created to do the simulation and the simulated results have been optimised using linear programming and Monte Carlo.**

***Index Terms*—Simulation, SimPy, Dynamic data creation, Monte Carlo, Linear Programming, Optimisation, Integer Programming**

I. Problem statement

The problem is divided into 2 parts; namely simulation and optimisation which are further detailed below:

- Simulate the operation of HS2 line from London Old Oak Common to Birmingham Interchange Station consisting of equidistant signaling blocks between these stations. Additionally, a single train is allowed at a time inside the signaling block and to verify the simulation incidents have to be added to the network.
- Using simulation, maximise the number of trains with the variable number of signaling blocks, given that the delay time at the Birmingham station is less than or equal to half of inter-arrival time.

II. Assumptions and derivations

A. Given in the question

The following assumptions have been given in question:

- The air resistance effect has been overlooked and a steady acceleration (0.76 m/s^2) and deceleration (0.38 m/s^2 is believed upto maximum speed (86.1 m/s).
- Equal length signaling blocks are present on the track between the 2 stations, one after the other, in a series.
- Energy optimal deceleration (using regenerative braking) is 0.38m/s^2.
- It takes 113,3s to accelerate from 0 to top speed, and the train travels approximately 4,878 m.
- The path to the Birmingham Interchange Station from London Old Oak Common Station is 145 km.
- 5 seconds after the train is out from the block, the block signal is switched back to green.

B. Extra assumptions

The track consists of

- k equidistant signalling blocks
- 2 stations (London and London Old Oak Common and Birmingham Interchange Station)
- 2 depot (London Old Oak Common and Birmingham Interchange Station)

C. Derivations

As per the calculation, the distance required for deceleration from maximum speed is equal to 9,754.22 and thus the maximum number of blocks can be 15 to maintain the safety of the train and passengers.

Distance deceleration:

$$distance_{deceleration} = \frac{Speed^2_{intial} - Speed^2_{final}}{2 * deceleration}$$

$$dD = \frac{86.1^2 - 0^2}{2 * 0.38} = 9,754.22m$$

Time deceleration:

$$time_{deceleration} = \frac{Speedintial - Speed_{final}}{deceleration}$$

$$tD = \frac{86.1 - 0}{0.38} = 226.6s$$

III. Objects and Structure

The train infrastructure has been captured by creating modular code for each individual unit that forms and integral part of the system and has some important characteristic features as shown in Fig. 1 like:

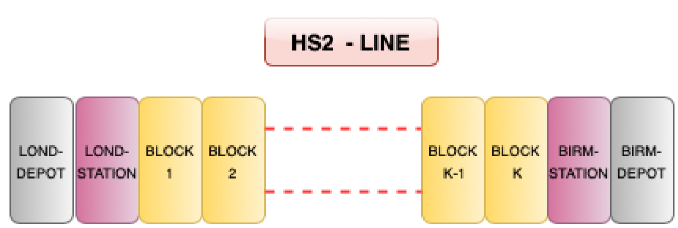

Fig. 1. Structure of HS2 Line

1) Network Nodes:
Network Node forms the base of all communication

through the system and has properties like line, direction, inter-arrival and inter-departure times that define a particular network in whole.

2) Stations:
A station is the starting point and end destination of the track with various signalling blocks spread across in between two end points. It has its own dwell time and drive time which can be randomised using exponential distributions.

3) Block:
A signalling block is similar to a station but is different in terms of dwelling times and processing capabilities. Number of blocks varies for a system but it has only 2 fixed stations.

4) Depot:
A train depot is built for feeding trains into stations to maintain order and control traffic and it works at a fixed frequency of n number of trains per hour.

5) Track:
Tracks are generally used to map the journey of a train when it starts from the depot and goes to the end station while crossing k signalling blocks on the way. It keeps a record of all the times and positions of all the objects/trains currently active in the system by using tracing methods.

6) Network:
Network object is another type of entity that defines the system and in this project has been primarily used to cause system shutdown by causing incidents for validation of the working of the simulation system.

7) Incident:
Incidents can be self- registered by specifying the start time, stop time and location on the track to cause breakdown.

8) Train:
Train entities are called by the track to manage the functioning of the system by maintaining the requests to resources on each arrival and departure. If a resource is occupied in a block, the next train has to wait out until it gets released and even 5 seconds after that to maintain proper flow of objects.

IV. Modeling Entities

A. Data Preparation

Following steps have been taken to generate the dynamic data for the blocks:

- Based upon the number of blocks, the driving time is calculated using function "timeTo" and the Dwell time is generated using a log-normal distribution to introduce variability in the system. The mean and standard deviation used are 3.378 and 0.751 respectively using the reference [1].
- Signal name followed by number of the block has been given for signalling blocks; like B1, B2, etc. while generating data for a particular value of K, where total blocks will b e equal to K.
- Distance, Speed and Time required to travel from stations to signals and in-between signals has been calculated using constant acceleration and deceleration.
- For exploratory data analysis, distributions of dwell time and running times have been plotted using histograms after analysing the mean, standard deviation, min and max values of each as shown in Fig. 2 and 3.

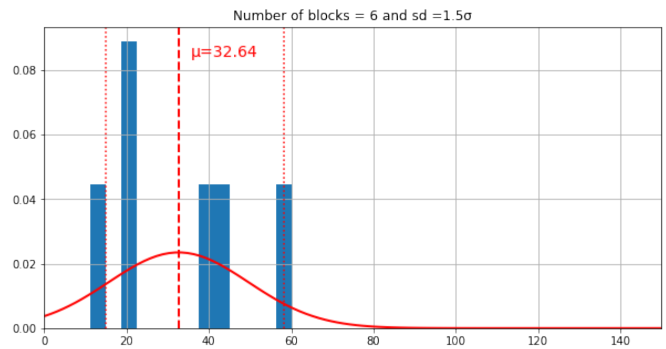

Fig. 2. Blocks dwell time: Histogram

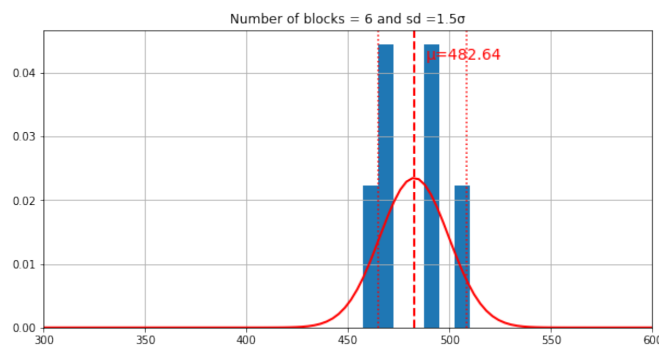

Fig. 3. Blocks running time: Histogram

B. Utilities

- Object oriented paradigm has been followed throughout the project using which various utilities or helper functions have been built to assist in modeling.
- Some of them are dynamic data generation, parsing time, schedule for simulation, scaling of dwell time, peak time, and drive time.

C. Global parameters

- Global parameters are used to make information accessible across different classes which remains constant and frequently accessed for running operations.
- For eg., different dwell and drive times have been used for stations, signals and depots which need to be constantly called by various modeling paradigms.

V. Simulation and Results

Once the objects and basic structure of the system has been finalised, the system needs to be verified by running under different conditions to test all possible ways of system failure if any. In this project two types of simulation runs were performed as following:

A. Normal Simulation

Normal simulation run consists of creating an environment variable from simpy library with a fixed seed value to maintain reproducibility. Then, a network object has been created and start, stop time have been fetched using parameters defined in the dataframe. The system is run from 7:00 am to 10:00am and the network trace is generated to validate the flow of trains from one station to another via signalling blocks that are fixed for this run but changed later in optimisation.

B. Simulation with Incident

A better way to validate the system is to induce delays or breakdown of fixed time and then check the functioning of different components of the system. Therefore, an incident has been registered at 9:00am at P2 signalling block meant to be cleared after half an hour. Then, complete trace over a period of time around the incident was verified to check for ambiguities. In particular, it was checked that the subsequent trains came to a stop and resumed operations after the incident was cleared.

Various simulation runs of the system help in understanding the working of the train network by extracting statistical data from each run. In particular, the inter-arrival times, dwell times, drive times, and inter-departure times have been extracted from network.getTracks() and exploratory data analysis has been done to visualise the results and gain insights about the system. Furthermore, delay time for various blocks and stations has been calculated with the help of simulated data and a distribution of actual delay times has been plotted. Also, the simulation code has been built into a single routine with parameters to change the number of blocks and enable/disable logging traces.

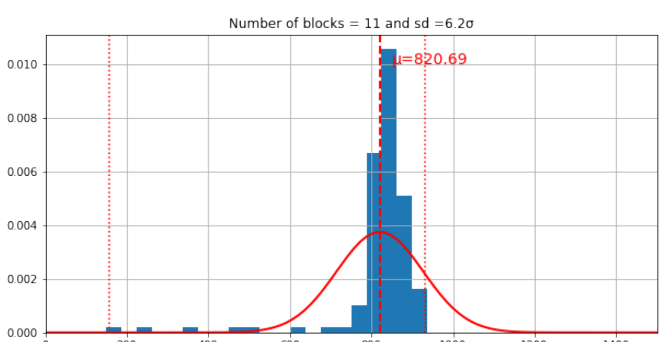

Fig. 4. Delay time distribution for 11 blocks: Histogram

VI. Optimisation and Results

Various optimisation techniques have been used to optimise the results of simulation which have been consolidated as mentioned below:

A. Linear Programming

Linear programming is an optimization technique for a system of linear constraints and a linear objective function. An objective function defines the quantity to be optimized, and the goal of linear programming is to find the values of the variables that maximize or minimize the objective function [1].

- For formulating a linear objective function some coefficients related to the data were needed so ordinary least squares (OLS) regression has been used and different columns of the consolidated data have been passed to the model.
- Objective function is defined using all the coefficients for k,n,delay time and inter-arrival time.
- Bounds are specified and constraints added using LHS and RHS inequalities.
- The results of the optimisation generated a solution of 6 trains and 2 blocks as shown in Fig. 5.

```
Problem:
MAXIMIZE
0.22253257078951172*BIRM_IARR + -0.004950617385175462*Delay_Time + 7.889423960839332
SUBJECT TO
_C1: - 0.5 BIRM_IARR + Delay_Time <= 0

_C2: K >= 2

VARIABLES
260 <= BIRM_IARR <= 1200 Continuous
100 <= Delay_Time <= 600 Continuous
2 <= K <= 15 Integer
```

Fig. 5. Linear programming result

B. Integer Programming

In a general integer linear programming problem, we seek to minimize a linear cost function over all n-dimensional vectors x subject to a set of linear equality and inequality constraints as well as integral restrictions on some or all of the variables in x [2]

- Integer programming using pulp has been used to validate the results further by maximising the objective function based on coefficients generated by the OLS model.
- LpVariables have been defined for all the parameters like delay time, inter-arrival times and number of blocks and a problem has been created.
- The constraint of delay time being less than half of inter-arrival time has been added to the problem.
- The optimal solution of the integer programming problem as described in Fig. 6 has been found to be 8 trains.

C. Monte Carlo

The Monte Carlo method uses a random sampling of information to solve a statistical problem; while a simulation is a way to virtually demonstrate a strategy. Combined, the Monte Carlo simulation enables a user to come up with a bevy of results for a statistical problem with numerous data points sampled repeatedly [3].

[1]https://brilliant.org/wiki/linear-programming/
[2]https://neos-guide.org/content/integer-linear-programming
[3]https://www.investopedia.com/articles/investing/112514/monte-carlo-simulation-basics.asp

```
     con: array([], dtype=float64)
     fun: 6.97472660936872
 message: 'The problem appears infeasible, as the phase one auxiliary problem terminated successfully with a resi
dual of 3.6e+02, greater than the tolerance 1e-12 required for the solution to be considered feasible. Consider i
ncreasing the tolerance to be greater than 3.6e+02. If this tolerance is unnaceptably large, the problem is likel
y infeasible.'
     nit: 0
   slack: array([-359.5])
  status: 2
 success: False
       x: array([  1.,   2., 100., 260.])
```

- Max trains = 6
- Blocks = 2

Fig. 6. Integer programming : Pulp problem

- Monte Carlo uses a different approach than integer programming and linear programming.
- A loss function with variable x,y,z for values of k, delay time and inter-arrival times has been defined using output of OLS model.
- Monte Carlo approach has been formulated in a method which seeks to minimise the values of all the parameters given.
- Upon running with different seed values and n values, the method generated an optimal solution of 9 trains per hour and 11 blocks as shown in Fig. 7.

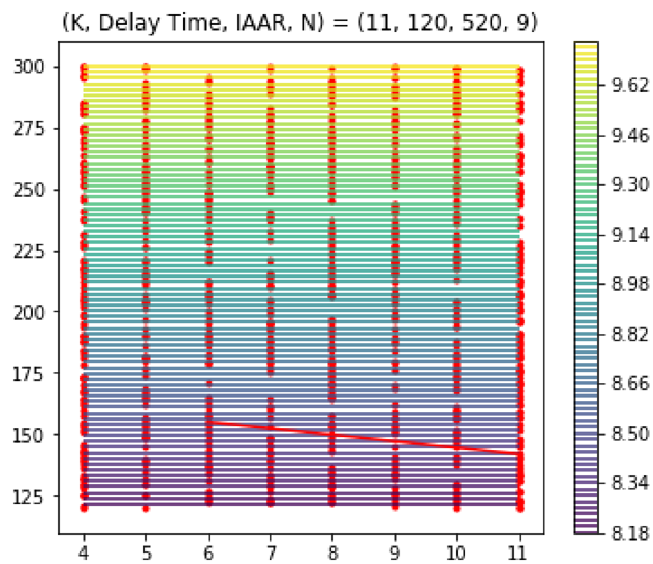

Fig. 7. Monte Carlo Result

VII. CONCLUSION

Overall, from simulation and optimisation models, the best results are generated using Monte Carlo optimisation with maximum trains running per hour as 9 with 11 blocks under the condition that the average delay time should not be higher than half the scheduled time between consecutive trains. Moreover, a simulated schedule of trains running from London Old Oak Commons to Birmingham Interchange along with delay times has been generated as shown in Fig. 8.

[1]

	LOND-Dep	BIRM-Arr	Delay Time
0	07:00:57	08:11:54	00:02:36
1	07:07:45	08:20:15	00:04:10
2	07:14:25	08:28:41	00:05:55
3	07:21:14	08:37:13	00:07:39
4	07:28:00	08:44:55	00:08:35
5	07:34:39	08:53:34	00:10:34
6	07:41:32	09:01:20	00:11:28
7	07:48:18	09:10:41	00:14:02
8	07:55:53	09:18:42	00:14:28
9	08:04:19	09:26:05	00:13:26

Fig. 8. Schedule of trains from London Old Oak Commons to Birmingham Interchange with delay time

REFERENCES

[1] Y. Yang, P. Huang, Q. Peng, J. LI, and C. Wen, "Statistical delay distribution analysis on high-speed railway trains," *Journal of Modern Transportation*, vol. 27, no. 3, pp. 188–197, 2019. [Online]. Available: https://link.springer.com/content/pdf/10.1007/s40534-019-0188-z.pdf