Multi-train simulation of DC rail traction power systems with regenerative braking

M. Turan Söylemez¹ & S. Açıkgöz²
¹Istanbul Technical University, Electrical Engineering Department, Turkey
²Istanbul Ulaşım A.Ş., Istanbul Municipality, Turkey

Abstract

Simulation is an important part of the design and optimisation of DC rail traction power systems. With the help of simulation, it is possible to determine possible problems, reduce the design costs and optimise for several design criteria such as power consumption, passenger flow, and passenger comfort. This paper presents a new simulation tool that can be used for these purposes while discussing several problems that must be tackled in writing a rail traction power simulator. Keywords: multi-train simulation, regenerative braking, DC rail traction systems.

1 Introduction

Computer-based simulation of electric railway traction power systems has been studied since the late 1970s [1-9]. Simulation tools are now considered as a standard part of the design and analysis of traction power systems. Power engineers use simulators to size all kinds of cables to be used, to determine the locations and power ratings of substations, and to try different modifications in the overall system in order to produce a cost effective design.

With the advent of computer technology, the type and scope of the properties expected from a simulator has increased considerably over time. It is actually possible to list the main properties that must be possessed by today’s simulator as follows:

1. Correctness: Simulation results should reflect the real system.
2. Fastness: Simulation results should be produced quickly.
3. **Flexibility:** The simulation tool should allow the parameter changes and changes in the interconnection of subsystems.

4. **Interaction:** Simulation tool should provide a nice interaction for the user. This is achieved with a good user interface.

5. **Standardization:** The tool should use available standards to ease up data manipulation.

6. **Report generation:** As part of a good user interface, the simulation tool should be able to create detailed reports of simulation results.

In this paper, we discuss how to tackle several problems faced by a traction power simulator so as to have the above properties while presenting a newly developed simulation tool called SimuX (or Simulator X [10]) that can simulate multi-line, multi-train DC rail traction power systems with regenerative braking. The organization of the rest of the paper is as follows: In Section 2, the type and kinds of data required for simulation as well as how to alleviate the process of entering this data is discussed. Simulating train movements is discussed in Section 3. Section 4 is on introducing the methods for solving the power network involved in simulation. Some comments on report generation are provided in Section 5, and information on the object oriented programming technology used in developing the simulator is given in Section 6. Section 7 consists of conclusive remarks and possible future work.

## 2 Data input

Data input is often an overlooked aspect of simulation programs. In simulation of traction power systems, there are hundreds, if not thousands, of parameters to be entered by the user in order to have a satisfactory simulation. Therefore, it is essential to have an easy to use interface that allows the input of details, whenever necessary and provide acceptable default values when the details are not available as discussed by Martin [8]. In this section, we attempt to list the kind of data handled by a realistic simulator.

### 2.1 Use of Libraries

Some of the data can be put into libraries so that it is used again and again in different simulations. The following data, for example, can be kept in libraries:

a. **Vehicle data:** This includes all kinds of data to characterise a railway vehicle such as maximum speed, maximum acceleration, comfort rate, dimensions of the vehicle, minimum and maximum operating voltages, maximum voltage allowed on regenerative braking (if it is enabled for the vehicle), auxiliary power and passenger capacity. Characteristics of the motor of the vehicle such as curves for tractive effort vs. speed, and braking effort vs. speed can also be entered as part of the vehicle data. These are usually given as a list of representative points connected using 0th, 1st or 2nd order interpolations. Note that some of these characteristics have distinctive properties and may at the same time depend on the other characteristics of the vehicle. For example, usually the tractive effort curve is flat while the speed...
is less than the base speed (which depends on the line voltage) and then decreases as the speed increases.

b. **Power line types:** The information on different types of power lines (catenary and third rail), especially the resistance and the change of resistance with temperature, can be kept on a separate library.

c. **Rail types:** Similar information for different types of rails can also be entered in a library.

### 2.2 Editing the track

Easy manipulation of the track data allows the user of traction power system simulators to try different combinations of construction to optimise the design, detect possible errors in the input data, and hence speed up the whole design process considerably. The following components are often used in a traction power system simulation:

a. **Lines:** A traction power system consists of one or more lines that may be interconnected through substations and/or jumpers. Typically the length, the types of power line and rail, the conductivity of the rail (to calculate stray currents), gradients and curves on the line, speed limits in both directions, and locations and properties of passenger stations are required for each line to produce a meaningful simulation. Note that outer temperature changes in a day can also be taken into account in more realistic simulations, as this affects all kinds of resistive elements in the electrical network. The types of cables or rails used, the conductivity of rails, and the temperature can vary in different sections of a line (e.g. in tunnels), as well as in different lines.

b. **Transformers (substations):** Nominal operating voltage, properties of the section insulators, the type of earthing strategy (i.e. floating earth, grounded earth, diode earth, or floating earth with rail potential control devices (RPCD) [11]) with proper corresponding parameters, and the properties and connection locations for each feeder is required to define a substation. Note also that a transformer can be flagged as out of service (even in the middle of a simulation) to simulate faulty conditions.

c. **Trains:** A train can be considered as a list of vehicles, with a location, speed and acceleration on a given line. The list of all stations to be stopped should also be determined by the user. Note that a train can be placed anywhere on the track by the user or dispatched by a depot with a predetermined schedule.

d. **Depots:** Depots can be used to dispatch and/or retire trains. Note that it should be possible to determine train schedules with constant and varying headways.

e. **Passenger Stations:** Location of passenger stations and dwell-time statistics are essential for train performance simulation. A realistic simulator should also be able to use passenger flow statistics.

f. **Section Insulators (SI):** These are used to insulate sections of a power line or a running rail. The locations and on/off states of section insulators should be determined by the user.
Jumpers: Jumpers are used to parallel power lines, or running rails of separate tracks to balance voltage differences.

**Rail to ground circuits (RGC):** Some systems may include circuits that connect running rails to ground to increase the safety of the system. These circuits can take the form of a simple resistance, a diode, or a voltage-time dependent relay circuit usually placed at safety critical places such as passenger stations and depots [11-13].

**Short Circuits:** These are usually used to simulate short circuit conditions and can be placed between power line and rail or earth anywhere on the track at any given time.

One important aspect of a good simulation tool is allowing easy editing of simulation data both offline and online (when applicable). With this respect, a visual representation of the track and allowing copy & paste facilities for all the elements considered above brings enormous amount of flexibility to the design process, see Figure 1. It is also important to provide mechanisms to import data prepared in different environments. For this purpose, SimuX uses optional delimited text input areas to retrieve all kinds of data expressed as lists (e.g. gradient and curves of lines, and characteristics of motors). Note that this allows import of data available in spreadsheets and databases.

![Figure 1: Typical main screen for editing a track in SimuX.](image)

### 3 Train movement simulation

A traction power system simulation can be considered in two parts: train movement simulation, and solution of the power network. Although some simulators isolate these two parts, feedback of the power network solution is essential, if the performance limits of the system are to be properly examined as stated by Goodman et al. [4].

In train movement simulation, the trains are assumed to be rigid bodies moving along the track with no slipping or sliding, and obeying the Newton’s third law:

\[
Ma = F - F_R - F_G - F_C
\]
where \( M \) and \( a \) are the mass and acceleration of the train, respectively. In equation (1) \( F \) is the tractive effort applied by the motor, \( F_R \) is the resistance force that depends on the dimensions and the velocity of the train and friction coefficients of the track section, \( F_G \) and \( F_C \) are gradient and curve forces that depend on the gradient and radius of the track at the position of the train (see [14] and [7]). The following algorithm can be used to simulate train movements:

1. Using characteristics of the train (e.g. max. speed and comfort rate), and signalling mechanisms (fixed block or moving block) determine the target acceleration.
2. Find the corresponding required tractive effort using equation (1).
3. Using the motor characteristics and the line voltage determine the maximum tractive effort or maximum braking tractive effort (depending on the required tractive effort) possible with the current speed of the train.
4. Determine the actual acceleration and corresponding electrical power requirement (or regeneration) of the train. Note that in most of vehicles a certain braking deceleration is guaranteed so that if the maximum braking tractive effort is exceeded mechanical brakes are used.
5. Determine the new position of the train to be used in next iteration.

4 Network solution

It is possible to solve traction power network equations using different approaches such as modified load flow approach and direct matrix method as stated by Goodman et al [4]. A direct matrix approach where the network matrix is formulated using nodal analysis is adopted by SimuX. In this approach, the power network is assumed to be formed by resistances and pure voltage or current sources at a given time. Several difficulties including, large dimensions of the problem, the dynamic topology of the network, and nonlinearities brought by regenerative braking have to be tackled in the solution as discussed by Goodman and Siu [15]. Actually, these topics deserve a detailed discussion, as it is important to handle these problems correctly to have a working simulation. Due to space considerations, however, we shall only mention the main problems and some of the approaches to solve them in the following, and leave a more detailed discussion to a future paper.

a. **Large dimensions:** Since there are many elements of the system the resulting network matrix is usually a large positive definite (PD) sparse matrix usually with hundreds of rows. In each time step, a linear equation involving this matrix needs to be solved probably more than once (if trains with regenerative braking exist). Therefore, it is important to use efficient methods to solve the power network. There exist several powerful algorithms such as LU decomposition, Cholesky decomposition, use of diakoptics approach [15], and use of Zollenkopf bifactorization [2, 3] in order to solve such equations. Currently, SimuX uses a modified version of SparseLib, which has been originally developed by K. S. Kundert at the University of California [16]. SparseLib provides a set of procedures to efficiently preorder and solve a PD sparse matrix using an LU factorisation.
It is possible to show that the computational complexity of this approach is linear with respect to the number of nodes in the network (see Figure 2).

b. **Dynamic topology:** Due to moving elements (i.e. trains) the topology of the power traction network changes in time. Therefore effective numbering of nodes may become essential in providing fast solutions as stated by Goodman et al. [4, 15].

c. **Regenerative braking:** Since the operating mode of the rectifier substations can be conducting and non-conducting, and there exist upper limits on the operating voltages of trains, the solution of the power network poses a non-linear problem when regenerating trains exist in the track. Fortunately, the problem can be solved using piecewise linear models assuming constant speed in a given (small) time interval (called sampling time) [15].

![Figure 2: The number of nodes on a typical power network vs. the time to simulate the network for two minutes (with 100ms sampling time) using SimuX on a Pentium III – 1GHz computer.](image)

## 5 Report generation

The data processed by the simulation tool is meaningless if the user cannot visualize and interpret it. This is achieved by efficient report generation. SimuX provides several mechanisms to visualize the output data. The front end of SimuX actually provides an active visualization of the simulation process by showing graphically the locations of all trains and on/off states of substations etc, and providing statistical information about the important parameters of the simulation such as maximum currents observed on all feeders. Besides this, all kinds of data that are required to be recorded are saved using a mechanism called *log*. A *log* can represent the change of a variable over time. It is possible to visualize several logs on separate windows (called scopes: see Figure 3) working as independent threads, as well as to export them to well-known software packages including Matlab, Mathematica and Excel. Note that the scopes provide statistical information on the variable that is logged, and allow the user
examine the data in a detailed manner (e.g. providing zoom-in, zoom-out facilities and permitting the use of different kinds of measurement units for a given variable).

Another mechanism called *message board* (Figure 4) allows the user to follow important events in the simulation using a textual interface. In this mechanism, all the objects that are part of the simulation can dispatch messages when an event related with that object occurs (e.g. when a train stops at a station). It is recommended that the types of messages are put into one of the following three categories: normal (e.g. train leaves a depot), warning (e.g. train is close to the next train), and fatal error (e.g. the line voltage is smaller than the min. operating voltage). The user should be able to filter messages that are produced at a given time interval by a certain object, or with a certain type.

A third mechanism called *info box* allows the user to simply click on any object on the track to get brief information on that object. The overall results of the simulation should be put together and exported to a text file using the well-known industry standard XML.
6 Object-oriented programming

Object-oriented programming (OOP) is a programming approach especially suitable for large programming projects. It is possible to classify all real life objects and model them separately with the help of encapsulation property of OOP. As a result, the programmer can focus on properties and interface of each class individually, and minimize the risk of programming errors. Since there are many objects that interact with each other in railways, simulation of traction power systems forms an ideal application area for object oriented technology.

The hierarchy of some of the important classes of objects used in SimuX is given in Figure 5. Here, LoggingObject class is considered as the base of all other classes. A logging object is supposed to be able to log several variables, produce messages and reports, and graphically display its logs. Two immediate subclasses of the LoggingObject are the Track and TrackLine classes which represent the whole track and a line in the track, respectively. All other classes are considered to be derived from a virtual subclass of the LoggingObject called LocationObject. A location object should be connected to at least one track line at a given location. Some location objects are connected to only a single line (e.g. passenger stations, trains, SIs and short circuits), whereas some can be connected to several lines (e.g. depots, jumpers and substations). Classes of objects that are directly part of the electrical network (e.g. substations, trains, SIs and jumpers) are derived from a virtual class NetworkObject, which is a subclass of the LocationObject. Such a class hierarchy allows enormous flexibility and code reduction by the help of mechanisms such as polymorphism and inheritance provided by OOP. For instance, efficient algorithms can be used to sort and locate location objects. Similarly, all network objects can be inserted in the network, update their node numbers etc. using the same interface.

Figure 5: Class hierarchy used in SimuX.
7 Conclusion

In this paper, several aspects of designing a simulator for traction power systems are discussed, and a new tool for such simulations, in particular, is presented. In general, a good simulator should have the following properties: correctness, fastness, flexibility, interaction, standardization and report generation. In verification of the simulator, we have used Eminönü-Zeytinburnu tramway line and Aksaray-Havalimani LRT line in Istanbul, Turkey as case studies. Our first experiments show that the simulation results are in line with the real world data within a 10% error margin, particularly as far as the maximum currents on feeders, power and energy trends of substations and power consumption per vehicle per kilometre are concerned. The tool has also been started to be used to compare possible power schemes for future lines [17].

Proposed future work includes a more detailed simulation of the signalling system and including new components such as traffic lights and switches.

Acknowledgement

We would like to thank Istanbul Ulaşım A.Ş. for their support and help in providing real world data in the verification of the simulation process.

References


