

# Web-based Software-Suite for DC Railway Simulation and Analysis: Architecture, Data Management and Visualization

Islam El-Sayed

LEMUR Group, Electrical Eng. Dept.  
University of Oviedo, Gijón, SPAIN  
corr. author: islam@uniovi.es

Bassam Mohamed

LEMUR Group, Electrical Eng. Dept.  
University of Oviedo, Gijón, SPAIN  
mohamedbassam@uniovi.es

Pablo Arbolea

LEMUR Group, Electrical Eng. Dept.  
University of Oviedo, Gijón, SPAIN  
arboleapablo@uniovi.es

**Abstract**—In this paper, a web-based application for DC Railways networks analysis is presented. The paper provides the guidelines to develop an integrated simulation framework containing different elements like server, databases, visual analytic tools using open-source software. In this case, the proposed application allows to design a DC railway feeding system and analyse the impact of the different agents like vehicles, substations, overhead feeding systems, on-board and wayside energy storage systems, etc. However, the same structure and described tools could be used to develop integrated frameworks applied to other railway or electrical transportation systems.

**Index Terms** – Electrical Railway Systems, Power Flow, DC Traction Networks, Railway, Energy Storage, Simulation Framework, Open-Source Software.

## I. INTRODUCTION

In recent years, railway feeding infrastructures suffer drastic changes triggered by the use of new technologies in the traction networks [1] and the rolling stock [2]. In most of the cases the changes have to do with the installation of new power electronic devices at infrastructure level or at vehicle level [3], the use of energy storage systems installed on-board or wayside [4], [5] or the integration of renewable energy sources in the traction networks [6]. Due to the above-exposed reasons, the authors made big efforts in order to develop tools to model these new technologies and asses and analyse their impact over the distribution systems.

In most of the cases, the tools developed by the researchers are specially conceived to improve the planning or operation of the systems . In many cases, researchers provide very sophisticated mathematical models of the railways feeding systems including or not the interaction with the rolling stock or the transmission and distribution networks. However it is not so common to find academic tools with a good integration between the solving engines and visual analytic modules that allow the designers to obtain an assessment of the analysed case in a fast and simple way, and in many times the results are obtained in raw csv or text files that need to be interpreted using third party software [7].

---

This work was partially supported by the Government of the Principality of Asturias - Foundation for Scientific and Technical Research (FICYT) under grants FC-GRUPIN-IDI/2018/000241 (Laboratory for Electrical Energy Management Unified Research (LEMUR)).

Regarding the existing commercial software with quite advance data driven visualization modules, we need to emphasize that railway feeding infrastructure is a very unique infrastructure with extremely specific features. In addition, the number of companies that develop, plan and operate these networks is quite reduced compared to those developing the same activities in conventional distribution networks. As a consequence, the number of commercial software for this purpose is really limited and these tools do not usually provide access to the mathematical models or the newest technological advances. In addition, many companies develop their own simulation tools [8]. Other solutions are very focused in the rolling-stock but with low emphasis in the distribution systems [9].

Probably one of the most complete software tools in the market is OpenTrack [10], it has many modules that allow the user to calculate all kind of systems from high-speed rails to light rails, metro systems, maglev, mining railways, rack railways...The tool allows to determine the requirements of the infrastructure and calculate energy consumption of the trains among other calculations. In addition, when OpenTrack is combined with OpenPowerNet [11], it can run railway power supply system simulations like the ones studied in the body of this paper. A similar approach is the one offered by [12], in this case, the software offers a cloud based environment for simulation and modelling.

SIGNON developed two software suites called Sinanet and Webanet to solve respectively DC and AC traction systems [13], both software are capable of running power flow analysis. eTraX module [14] from eTap [15] software is a unique solution since allow to run power flow simulations in AC and DC systems, but in this case including the interaction with the distribution network that feed the railway power supply. The software Traferri is only oriented to obtain the power consumption in the railway power substations [16].

This work aims to provide guidelines on how to integrate different modeling, simulation, visualization and analysis tools within a simulation framework so that researchers can implement these types of solutions and get the most out of the data obtained by simulating their models. In this particular case, the authors will use a set of models representing a DC drive system. These models are extensively described in [17], [18].

In this paper we will not focus on their description but on how to integrate them in a web simulation environment using open-source tools.

The structure of this paper is as follows. In section II an overview of the software architecture structure is presented. Section III is focused in describing the front-end components that contain all the interaction between the software and the user. Back-end components containing the server configuration, the interaction with the solver engines and the database structure are described in section IV. A case study using a test network is explained in section VI. Some results are analysed in section V and conclusions are presented in section VII.

## II. APPLICATION STRUCTURE

The proposed software is a web-based application that allows multiple access from different clients. The main advantage of this technology is that the client can use any operating system. The whole application is allocated and served in a Nodejs environment. The application consists of two main parts. One part is the front-end in which all data are introduced, managed and visualized. The other part is the back-end in which all networks data are stored and processed by the algorithm. The whole project is developed using javascript programming language for both front-end and back-end. For the communication between both parts, HTTP protocol is used. Fig. 1 shows the architecture of the web application. The flow of the applications begins from user side when requesting the web application via URL. When main page is rendered, the application asks for credentials to login. The login credentials are sent to server for verification. The server verifies the credentials and sends the result to the front-end either with success or fail. Once login is succeed, user profile is saved in his session for next requests. For all user requests, there some policies applied to check user privileges. In next sections, these two parts and their components are explained.

Fig. 1. Web application architecture

## III. FRONT-END COMPONENTS

The front-end is the side where the user can manage his networks and where the data are visualized and analyzed. For the development of the front-end, Angularjs framework is used. This framework allows to build application for different platforms using the same code. Also it can achieve high speed and scalability. As mentioned in Fig. 1, the front-end developed in Angularjs framework contains some main components like views, templates, controllers, directives and services. The views are the HTML files that contains the markup of the web application. The templates are EJS files (Embedded JavaScript templates) that contains the dynamic contents and data to be rendered in the browser. The template is attached to a javascript code called controller. These controllers provide dynamic data and event handlers for the templates using two-way data binding. Two-way data binding means that any changes in the models inside the controllers will affect the views rendered in browsers and any changes in views will update the models in the controllers. In different templates, some repeated code may be needed. In that case, a directive is used for optimizing the code. A directive can be considered as

a shortcut for the data to be rendered on browser and needed in different parts in the application. A directive is called with an HTML tag passing some parameters to the directive. For example, a directive can be used to show the name of an element in the network such as a substation or a path. In that case, the directive is called passing the type and id of the element. The result of this tag is to see the element name rendered on the screen. Another directive is the one used to select line nodes. This directive is used in different templates to create trains lines, buses lines or also links. As a result of using directives, the code is more optimized and organized. The last component in the front-end is services. As a directive is used to optimize the code for templates, the services has the same role for the controllers. A service contains some functions and variables that can be used in different controllers. Finally, the user can interact with the templates rendered on browser that may contain some directive and get their data from controllers and services.

For the visualization of of different data, D3js library is used. D3js is a powerfull javascript library for Data Driven Documents [19]. Using D3js library, different SVG charts and diagrams can be created customized for the web application.

## IV. BACK-END COMPONENTS

The back-end is the invisible part for the user that has three main functionalities. The first function is to respond http requests coming from front-end. The second one the storing data related to users, nets and nets data. The last one is the power flow algorithms and post processing program. All

### A. Server

The server is the component responsible of handling all requests from different clients to carry out a specific process like managing users, networks or simulation results. The server in this application runs in nodejs environment.

The server is developed using Salilsjs framework based on (Model-View-Controller) design pattern MVC. The Model is the part responsible of communicating with the data to write or to read. The View contains the pages templates made to be sent to the client using the data obtained by Model. The controller is the responsible of handling the client requests, manipulating the obtained data and serve the view pages to the client. Other main components of the server are the routing and policies. The routing module receives each request made by the front-end and redirect it to the corresponding controller and function. The policies are some modules to check the privileges of the client before passing the request to a controller.

### B. Data management

The data are organized in two main categories. The first one is the main database that contains the main data of users and networks. They are saved in SQLite database that contains just two tables (user and net tables) to manage users and networks registers. The main data in this databases are shown in table I. Each register in net table represents the metadata of a network such as the name of the network and its owner and some other data related to the organization of the networks and simulation status. The registers in user table describes each user and his credentials and privileges.

Column	Description
Net table	
id	Unique id for each network
name	Name of the network
tag	Tag to group networks
net_file	Name of input database file
res_file	Name of output database file
db_dir	Name of network directory
simulation	Status of net work simulation
description	Description of the network
user	Owner user of the network
client	invited user to view the network
User table	
id	Unique id for each user
name	Name of the user
username	Login username of the user
password_encrypted	Hashed password
permission	User permissions
email	User email
enabled	User status
password_change	Password status

TABLE I. MAIN DATABASE CONTENTS

The data of each network are organized in a folder in three main database files (net.db, res.db, post.db) containing input data, simulation results and post processing data respectively. The same network folder contains also all profile data files. These files contain (X, T, P) variables that have data about distance, time and power respectively. For Trains models the three variables (X, T, P) variables are required to describe the train profile during its path. For substations only (T, P) variables are required as there is no change in the distance. For buses, only (X, T) variables are required because the bus uses the same power all the time. Organizing networks data inside one folder instead of one large database achieves more performance avoiding queries in big tables. In these sets of variables, X represents the position of the elements (trains in this case), T represent the time and P the Power. There are several type of loads in the system, dynamic loads like the trains are entered as (X,T,P) loads and they change their position with the time. Static loads like EV chargers or other loads in the substations are entered like (T,P) loads, other moving loads like electric busses that do not consume power from the infrastructure when they are not connected are represented using (X,T) files.

### C. External tools

The external tools are the executable files to make some processes with networks files. There are three main executable files to make different processes separately. The first executable is the main solver that analyze the power flow for the whole network in a specific period of time using the schedule and profile data submitted for that network. A deep description of the power flow algorithm as well as the elements included in the simulation can be found in [17]. The second executable is the one which creates the post processing data containing aggregated data of substations and trains for the whole simulation. The third executable is the report generation program that receives a configuration database that describes the report structure and generates a  $\LaTeX$  project including all figures described in the configuration database.

## V. FINAL APPLICATION

The final application contains different pages to make different operations with user profile and network administration.

Profiles	Full control	Manage users	Manage networks	View results
Owner	✓	✓	✓	✓
Administrator	×	✓	✓	✓
Engineer	×	×	✓	✓
Client	×	×	×	✓

TABLE II. PROFILES PRIVILEGES

These pages are explained in next sections.

### A. Profile and administration

In these pages, the user can modify his profile and credentials and make some other operations depending on his profile. There are 4 predefined profiles (Owner, Administrator, Engineer and Client) as mentioned in table II. The owner user is a unique user who has all privileges of the application (managing users, networks and simulations). The user with administrator profile can manage the engineer and client users. The administrator also can create his own networks and copy them with different users. The Engineer user can only manage his networks and copy any of them for other users. The engineer user has no privilege to manage users. The last user profile is the client user who has no privilege to manage neither users nor nets. The administrators or engineers can assign networks to a client user to be able only to view the simulation results.

### B. Networks management and editing

For that purpose there are two panels to deal with networks. The first one is the selection page where the user can manage his own networks. In that page, user networks are listed in a tree view to help to identify main networks and their derived networks. Once the user selects a network from that tree, the network details are shown and more operations are activated. These operations contain the ability to change the net name, tag or description. Also to create a copy of the network for the same user or another user. Another ability also is to create derived networks from this network if it is a principal network and not derived from another one. This last operation creates a number of identical networks as the principal one and also some derived network with some automatic changes in the configurations of network substations.

### C. Simulation and Visualization

In this screen there are divided into six panels as shown in Fig. 2. The main panel in that screen is the marray lines panel. In that panel, all trips of a selected path are shown by lines with horizontal axis as distance and vertical axis as time. This marray lines panel is divided horizontally in segments representing the lines of the selected path. In each segment, all trips passing in this line are drawn to easily analyze the flow of the trains and their effects on the network parameters. This panel contains horizontal and vertical cursors with which a specific time and distance are determined.

As the marray lines panel is long depending on the schedules, an overall view of the whole schedule is shown in another panel highlighting the zone shown in the marray lines panel.

Another panel shows the map of the network highlighting a selected train/bus path. In that map, substations are drawn

with different symbols depending on their types. Also trains and buses are drawn with a color as the path of their trip. The map is animating while moving the cursor in the map lines panel.

After having the simulation results, three data panels are used to plot different electric signals. One panel is used to show signals related to distance. Another one is used to show signals related to time. The last panel is used to draw an animated power flow diagram for the element selected as either a train or a substation.

All these six panels are synchronized together. So, Moving the vertical or horizontal axis, the values in the different data panels are changed depending on the location of these cursors.

#### D. Comparison and report generation

After having simulation results for different networks, it is recommended to have a tool to compare between the results of these networks. In comparison page, the user can easily compare between different signals from different networks with the same topology. Multiple panels can be generated with different comparison between signals. Networks with different topologies can also be compared but only the aggregated data. Panels can be distributed over the screen to create a dashboard that fully fills the user requirements.

Another useful tool is the report layout design tool. In the report generation tool, user can design the layout of a report. Sections, tables and graphs of different types can be generated graphically selecting the networks, elements and signals. After finishing the report layout design, a report configuration database is generated and passed to a tool to generate the latex project.

## VI. CASE STUDY

In this section an example network is introduced going through all steps starting from data entry till getting simulation results. Main properties of this network are mentioned in table III. Full demonstration of creating a network from scratch is explained in the additional material provided with this paper [20].

#### A. Network topology and characteristics

The first step in creating a new network is to set the base power for that network and create base voltages that will be used in the network. Then the substations should be introduced specifying their electrical characteristics and properties. To complete the net topology, lines and paths should be created to be connected between substations. For generating trains some train models should be created and then more instances can be taken from these models. On-board and off-board energy storage systems are created automatically and initially disabled while adding substations and train models. If a substation has an energy storage system, so it should be enabled and configured.

#### B. Profiles and schedules

A profile data file is supported to describe the behaviour of the trains containing where, when and how much power

Nodes	24	Links	4
Trains lines	20	Buses lines	6
Train	62	Buses	28

TABLE III. TEST NETWORK CHARACTERISTICS

they are consuming or generating. Also profile data can be introduced for substations containing when and how much they are consuming. Finally profile data for buses can also be introduced to describe when and where the buses are.

Once all profiles are introduced, they can be used to create different schedules. For the trains, a schedule can be created specifying the train model, profile data, starting time and repeating frequency. The same for substations loads and buses locations.

#### C. Visualization, simulation and analysis

In this part the network input data can be visualized in different panels synchronized together to analyze the train trips on the map. A simulation can be started for the network specifying the starting time and the period of simulation. Once the simulation configurations are set, an http request is sent to the server asking to simulate that network. Once the solver finishes the simulation, the client side starts to load the results data. Once result data are loaded, time and distance signals can be selected to be drawn in the corresponding panels.

#### D. Comparison and Report interface

In these parts, a comparison can be made between different networks comparing either times signals or aggregated data. Another useful part is a tool for generating graphically the layout file of a report containing comparing between different signals and networks.

## VII. CONCLUSION

In the present work the authors presented the guidelines and tools in order to build an integrated simulation framework applied to the simulation of DC railway networks. The authors demonstrated how the use of open-source tools allows the integration of all elements in a web-based environment in which the designer can develop the models, run the simulations and analyze the data in a very simple and agile way.

The proposed structure is quite general and it can be used to integrate whatever other application related to other kind of electrical railway or electrical transportation system. The proposed methodology allows to integrate different kind of simulators in a web based framework accessible remotely and also provide an integrated tool with a visual analytic module able to extract a very deep knowledge of the grid from the stored simulated information.

## REFERENCES

- [1] I. Krastev, P. Tricoli, S. Hillmansen, and M. Chen, "Future of Electric Railways: Advanced Electrification Systems with Static Converters for ac Railways," *IEEE Electrification Magazine*, vol. 4, no. 3, pp. 6–14, 2016.
- [2] J. Feng, W. Q. Chu, Z. Zhang, and Z. Q. Zhu, "Power Electronic Transformer-Based Railway Traction Systems: Challenges and Opportunities," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 5, no. 3, pp. 1237–1253, 2017.

- [3] D. Serrano-Jiménez, L. Abrahamsson, S. Castaño-Solís, and J. Sanz-Feito, "Electrical Railway Power Supply Systems: Current Situation and Future Trends," *International Journal of Electrical Power and Energy Systems*, vol. 92, pp. 181–192, 2017.
- [4] F. Meishner and D. U. Sauer, "Wayside Energy Recovery Systems in DC Urban Railway Grids," *eTransportation*, vol. 1, 2019.
- [5] P. Arboleya, P. Bidaguren, and U. Armendariz, "Energy Is on Board: Energy Storage and Other Alternatives in Modern Light Railways," *IEEE Electrification Magazine*, vol. 4, no. 3, pp. 30–41, 2016.
- [6] J. A. Aguado, A. J. S. Racero, and S. de la Torre, "Optimal Operation of Electric Railways With Renewable Energy and Electric Storage Systems," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 993–1001, 2018.
- [7] A. Pablo, M. Clément, M. Bassam, A. J. Antonio, and D.-L.-T. Sebastián, "A Review of Railway Feeding Infrastructures: Mathematical Models for Planning and Operation," *eTransportation*, vol. (Under Evaluation), 2020.
- [8] Siemens, "PSS - SINCAL," 2020. [Online]. Available: <https://new.siemens.com/global/en/products/energy/services/transmission-distribution-smart-grid/consulting-and-planning/pss-software/pss-sincal.html>
- [9] ESI Group, "SimulationX," 2020. [Online]. Available: <https://www.simulationx.com/industries/simulation-railways.html>
- [10] OpenTrack Railway Technology Ltd., "Opentrack," 2020. [Online]. Available: [www.opentrack.ch](http://www.opentrack.ch)
- [11] Institut für Bahntechnik GmbH, "OpenPowerNet," 2019. [Online]. Available: <https://www.openpowernet.com/>
- [12] SYSTRA, "Railsym X," 2019. [Online]. Available: <https://www.systrausa.com/en/technology/smart-tools/article/railsim-x-c>
- [13] SIGNON Deutschland GmbH, "Sinanet and Webanet Software Packages," 2019. [Online]. Available: [https://www.elbas.de/sinanetwebanet\\_e.html](https://www.elbas.de/sinanetwebanet_e.html)
- [14] Operation Technology Inc., "eTraX - Railway Traction Power Solution," 2020. [Online]. Available: <https://etap.com/solutions/railways>
- [15] —, "eTap - Electrical Power System Analysis Operation Software," 2020. [Online]. Available: <https://etap.com>
- [16] SH Rail, "Traferr - rail electrical traction and power systems simulation," 2020. [Online]. Available: <http://www.rail-simulators.com/traferr-energy-consumption-simulator/>
- [17] P. Arboleya, B. Mohamed, and I. El-Sayed, "Dc railway simulation including controllable power electronic and energy storage devices," *IEEE Transactions on Power Systems*, vol. 33, no. 5, pp. 5319–5329, 2018.
- [18] B. Mohamed, P. Arboleya, and C. González-Morán, "Modified current injection method for power flow analysis in heavy-meshed dc railway networks with nonreversible substations," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 7688–7696, 2017.
- [19] M. Bostock, V. Ogievetsky, and J. Heer, "D data-driven documents," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, Dec 2011.
- [20] I. El-Sayed, *Web application demonstration*, <https://bit.ly/RailsWebApp>, 2020 (accessed June 5th, 2020).