
Copyright 1998 IEEE
1. INTRODUCTION

With the advances in computer hardware and software development techniques in the past 25 years, digital computer simulation of train movement and traction systems has been widely adopted as a standard computer-aided engineering tool [1] during the design and development stages of existing and new railway systems. Simulators of different approaches and scales are used extensively to investigate various kinds of system studies. Simulation is now proven to be the cheapest means to carry out performance predication and system behaviour characterisation.

When computers were first used to study railway systems, they were mainly employed to perform repetitive but time-consuming computational tasks, such as matrix manipulations for power network solution and exhaustive searches for optimal braking trajectories. With only simple high-level programming languages available at the time, full advantage of the computing hardware could not be taken. Hence, structured simulations of the whole railway system were not very common. Most applications focused on isolated parts of the railway system. It is more appropriate to regard those applications as primarily mechanised calculations rather than simulations.

However, a railway system consists of a number of subsystems, such as train movement, power supply and traction drives, which inevitably contains many complexities and diversities. These subsystems interact frequently with each other while the trains are moving; and they have their special features in different railway systems. To further complicate the simulation requirements, constraints like track geometry, speed restrictions and friction have to be considered, not to mention possible non-linearities and uncertainties in the system.

In order to provide a comprehensive and accurate account of system behaviour through simulation, a large amount of data has to be organised systematically to ensure easy access and efficient representation; the interactions and relationships among the subsystems should be defined explicitly. These requirements call for sophisticated and effective simulation models for each component of the system. The software development techniques available nowadays allow the evolution of such simulation models. Not only can the applicability of the simulators be largely enhanced by advanced software design, maintainability and modularity for easy understanding and further development, and portability for various hardware platforms are also encouraged.

The objective of this paper is to review the development of a number of approaches to simulation models. Attention is, in particular, given to models for train movement, power supply systems and traction drives. These models have been successfully used to enable various ‘what-if’ issues to be resolved effectively in a wide range of applications, such as speed profiles, energy consumption, run times etc.

2. TRAIN MOVEMENT

Train movement is the calculation of the speed and distance profiles when a train is travelling from one point to another according to the limitations imposed by the signalling system and traction equipment characteristics. As the train has to follow the track, the movement is also under the constraints of track geometry and speed restrictions and the calculation becomes position-dependent. The method of arranging the data representing the track geometry and speed restrictions is therefore critical to achieve effective and swift simulation.

2.1 Track-based data representation

In early simulation software designs [2], the data was stored in two-dimensional array with the rows of the array denoting the signalling blocks with fixed block signalling. The track-based data include block identity, gradients, speed restrictions, coasting points and signal aspects etc. within the signalling block. The major advantage of this structure is easy referencing. However, only a single data type (e.g. floating-point numbers) is allowed within the array, which is inflexible for accommodating the diversified nature of data. Besides, the array structure does not offer any representation of track layout except that adjacent rows in the array may depict a sequence of adjacent signalling blocks. When it is necessary to describe the track connections within a complicated railway network, the array structure must be enhanced or additional data structures are needed. Further, the size of the array required varies with application, a ‘supposedly’ large
2.2 Movement calculation

An object-oriented software approach [3] provides the solution for the above deficiencies. The object-oriented concept allows a set of objects in the physical world sharing similar properties and performing similar operations to be grouped together in one class. A class specifies a data structure for its objects and a number of permissible operations. An object’s data can only be accessed and modified by the permissible operations. This protection of data from arbitrary and unintended use and access of data is called encapsulation.

The object-oriented concepts have been applied in railway network modelling [4] [5]. The structure of the network is represented by a number of ‘node’ objects joined by ‘link’ objects. The nodes can be used to represent stations, junctions, points and termini etc. whilst the links are the tracks connecting the above features. They have their own data structures characterising the corresponding objects, and the data structures accept mixed data types for various types of data. Train movement is realised by moving the train from one node to the next through a permitted link, which contains the necessary information for the movement calculation. Creation and initialisation of any new objects from a class can be done at run-time and no reserved memory is needed.

The properties of inheritance and polymorphism for object-oriented systems are the key factors to achieve efficient software coding. Inheritance is an implementation of generalisation as a new class can be defined by the definition of an existing class. It makes the data structure and operations of an existing class (superclass) physically available for re-use by the new class (subclass); hence it enables code sharing. Polymorphism allows modifications of operations in a subclass so that different operations on different objects are possible in a hierarchy classes.

Despite its simplicity, a time-based model usually makes a high computational demand as a significant amount of information has to be produced within each time update. Even though computation effort can be alleviated by the choice of larger time update intervals, the attainable level of details is then compromised. This high computational demand can only be justified in applications where full details of every move of the trains are needed, such as energy consumption and signalling design studies [6] [7].

An event-based model, on the other hand, denotes the progress of train movement by the occurrence of a sequence of pre-defined events, such as arrival at and departure from stations [8] [9] [10]. Since the events are linked to each other according to the interactions among trains through the signalling, power system and other system characteristics, one event, as the consequence of a previous event, will trigger or cause another event to happen. As a result, a chain of events determines the progress of the trains.

Computational effort can be substantially reduced because the calculation of exact details of train movement between pairs of events is skipped. Nevertheless, this apparent advantage may be overshadowed by the fact that the passage of time is irregular and the updates of train movement are not carried out synchronously. It is possible that the processing of an event has to be postponed because the event to trigger it has not occurred in time, or it is processed first on the assumption of certain conditions and it will be re-processed if the assumption is found invalid later. Therefore, great care is needed in the development of event-based models in order to avoid the above drawback. Event-based models find applications mainly in traffic control and timetabling studies [11] [12], in which only the information of timings at certain events are of main concern and quick simulation results are expected.

3. TRACTION POWER SUPPLY AND DRIVE SYSTEMS

In designing new traction power systems, there is a considerable range of alternatives to be contemplated. Traction power engineers would need to analyse and compare the performance of alternative proposals to arrive at a cost-effective design which satisfies the client’s requirements. This calls for software procedures for solving the power network equations repeatedly in order to establish a complete picture of power demand, energy consumption, voltage and current of the feeder stations and trains. Due to the iterative nature required for solving the network equations, efficient algorithms are often needed for the power network simulator to provide accurate results within a reasonable computation time. There are three areas to be considered in developing a traction power network simulator:
• the representation of traction power networks in the solution domain;
• the train position locator; and
• the power network equation solver.

The representation of traction power networks in the solution domain has a direct influence upon the way in which the power network problem is formulated, and hence the train position locator and power network solver organised. There are, by and large, two types of approach to solve the traction power network problem namely:

- modified load flow type approach [14] [15] [16];
- direct matrix method combined with piece-wise linearised circuits approach [1] [2].

### 3.1 Modified load flow approach

For the modified load flow type calculations, the traction power network simulator is quite often separated from the train movement simulator (or train performance calculator). The traction power network simulator, as a stand-alone module, takes in the train movement results, such as train locations and train power demands etc. from data files or intermediate stores. This approach, from a programming viewpoint, provides a much easier interface between the train movement and traction power simulators. It does, however, provide no direct reflections of voltage variations back to the train movement calculations. With modern three-phase drives, the traction drive is less dependent on supply voltage than is the case with DC motors, but all drives do have a designed ‘graceful degradation’ response to reduced traction voltage. Thus, if the performance limits of the total system are to be properly examined, feedback of the power network solution to the traction performance calculations becomes essential.

Another feature of the modified load flow type approach is the combined network solver for AC/DC networks, e.g. 750 V DC to 11/33 kV AC ring. The equations of the two networks are alternatively solved until the final solution of the network is obtained for a given time update period. This particular feature provides the engineers a useful tool to analyse and optimise the complete AC/DC system. In the network equation solver, standard load flow procedures using the conventional algorithms, such as Gauss-Seidel and Newton-Raphson methods, need to be modified to cater for the non-linearities of the traction systems, such as regenerative trains, system non-receptivity, etc. These however apply to the direct matrix approach as well.

### 3.2 Direct matrix method

For the direct matrix approach, the network matrix can be formulated using either mesh analysis or nodal analysis. Comparing the two approaches, the nodal analysis is more suitable for dealing with complex networks, because it is often easier to identify nodes than loops in non-planer networks. From a programming perspective, automatic network set-up procedures and advanced network graph techniques are easier to implement with the use of the nodal approach. Fig. 1 shows the circuit representation of a typical DC traction power network with a branch. It is not difficult to see that the railway traction power network is characterised by sections of ladder type networks infrequently cross-connected. For this characteristic topology, the sparse matrix technique coupled with efficient matrix elimination methods leads to an expeditious network solution, since it does not suffer from the fill-ins that the direct inversion of the coefficient matrix requires.

### 3.3 Static and dynamic ordering

General speaking, the matrix of the traction power network equations is of positive definite (PD), symmetric and sparse type. Many sparse matrix elimination techniques tailored for PD matrices may be used, e.g. $LL^T$ decomposition, Cholesky decomposition etc. For the sparse techniques, the essence is the equations ordering. There are generally two types of ordering method namely, static and dynamic ordering. Typical efficient examples for long and thin ladder type networks include the Cuthill and McKee algorithm [17] and reverse Cuthill and McKee algorithm [18] which make use of the property that the zero elements situated before the first non-zero element on any row always remain zero, and take the advantage of the variation of the matrix bandwidth, also referred to as the matrix envelope.

For dynamic ordering, the minimum degree algorithm provides an efficient alternative for solving the network matrix. The minimum degree ordering scheme is one in which the pivot selection is made in accordance with the way in which the coefficient matrix develops, rather than simply from the structural properties of the original matrix. This is actually a heuristic algorithm for finding an ordering for the coefficient matrix which suffers low fill-ins when it is factored. Therefore, this scheme requires a simulation of the effects on the accumulation of non-zeros of the elimination process. In order to avoid direct elimination with actual values, a symbolic factorisation is usually adopted to obtain the zero and non-zero structures of the factored matrix. As the numerical values of the matrix components are of no significance in this connection, the problem could be studied using a graph approach, instead of using an actual matrix factorisation. The minimum degree algorithm is particularly suitable for solving medium to large networks, e.g. systems with 200 nodes and more. For smaller networks, the minimum degree algorithm
becomes a less efficient option, since a significant portion of the overall processing time will be used in the symbolic factorisation process.

3.4 Traction Drive Models

The basic function of a traction equipment model is to provide tractive effort (TE) output and current/power demands according to the given input parameters for the train movement and power network calculations.

The traction supply voltage with respect to the rolling stock can vary from -30% to +20% of the nominal value (IEC standard for DC). A voltage sensitive drive model is, therefore, essential in achieving accurate electrical and mechanical representations for the conventional DC traction equipment. However, for the modern three phase induction motor drive, the voltage fluctuation on the train pantograph or collecting shoe is less significant with the advanced pre-conditioning front-end technology. The voltage magnitude at the DC link can remain at a fairly constant level. Two kinds of voltage sensitive drive modelling algorithm, namely, the detailed [1] [2] and simplified approaches [19] have been used to represent the traction equipment. When the required information, such as the motor terminal characteristics, winding resistance and reactance, etc. are available, it is often desirable to model a drive using the detailed approach. However, the detailed drive modelling approach requires a deep knowledge of the traction equipment and motor parameters and these are not always easily available. In some circumstances such as at a feasibility study or preliminary engineering stage, it is not always possible to have enough information and time to model a new drive comprehensively. The simplified drive modelling approach based on data fitting and numerical techniques provides a much easier alternative, which only requires the high level information such as tractive effort vs. train speed curves which are generally much easier to obtain.

4. FUTURE DEVELOPMENT

There can be no doubt that computer modelling and simulation of railway systems provides an invaluable tool for engineers to evaluate different ‘what-if’ scenarios, which can lead to the minimisation of project cost and programme overruns. There are a number of areas envisaged to be worthwhile for future developments:

• more integrated environment for different levels of simulation, such as the detailed signalling systems simulator integrated with the traction power network simulator for studying the power demands and current transients for re-start situations after perturbations, in particular for those systems with moving block signalling;

• ability to select relevant aspects of simulation to suit application e.g. to ‘switch on or off’ signalling and power supply separately and possibly dynamically;

• further simplification of input data preparation with graphical data capture where appropriate or standardisation of interfaces to permanent way data bases;

• further widening of choice regarding output format e.g. animated run-time monitoring graphics; export to standard packages such as Excel; export to systems engineering packages for broader trade-off studies and systems evaluation requirements;

• extension to include dynamic passenger flow modelling and integration with traction performance calculations to reflect variable passenger loading;

• further extension of control aspects to include control traffic regulation, on-board trajectory calculation to enable simulator to be used as a test-bed for advanced train control concepts;

• refinement and validation of software models for direct use embedded within central control computers or within train-borne control computer.

5. CONCLUSIONS

Computer simulation of train performance, signalling and traction power analysis is long-established in the engineering design of railway systems. With the increased speed and improved user interfaces characteristic of modern computing systems, this trend will continue with simulation being a routine tool for the design and evaluation of most aspects of the total railway system. The trend will be towards a single integrated package, but with distinct functional subsystems enabling the user to focus on the aspect of current concern but without losing the realism of a total system model and thus preserving knowledge of the system implications.

In parallel and closely linked to the development of these off-line design aid programs, the same algorithms will be incorporated into on-line control software firstly in an advisory role, but eventually fully embedded into signalling and control systems. This will require a carefully structured approach to the question of software validation and clear thinking about the level of reliability required from each part of the software.
REFERENCES


Fig. 1 Simplified branched Traction Power Network.