Transactions on the Built Environment vol 18, © 1996 WIT Press, www.witpress.com, ISSN 1743-3509

Energy-saving automatic optimisation of train speed commands using direct search techniques

F. de Cuadra, A. Fernández, J. de Juan, M.A. Herrero
Instituto de Investigación Tecnológica (IIT-UPCO),
Alberto Aguilera 23, Madrid 28015, Spain

Abstract

This paper describes two computer tools that have been developed for the automatic optimisation of train speed commands. Given a target time for the trip between two consecutive stations, energy consumption is minimised. Two driving strategies have been considered, one based on coasting-remotor cycles and the other in space-related traction commands. Direct search techniques have been applied, using simulation processes that include detailed models of the train and the line.

1 Introduction

The work described in this paper is related to the co-operation held between the research institute IIT and Dimetronic S.A. in the area of modelling, simulation and regulation of railway systems, see Cuadra[1], Fernández[2,3].
One of the problems addressed in this line of work is how to make an optimal use of the ATO equipment (Automatic Train Operation) in a trip between two consecutive platforms. There are two levels of decision in this problem:

- Which is the optimal time of the trip, considering traffic regulation criteria
- Which are the optimal ATO commands for that optimal time, considering energy-saving criteria

These two decisions should be made on-line, requiring a huge amount of computing effort proportional to the number of trains. Fernández[2] describes how computing effort can be dramatically decreased by the use of a reduced set of pre-calculated trips, each one specifically designed for a target time.
In the case of regulation of metro lines, a careful design of these pre-calculated trips is very important, mainly because of two reasons:

Transactions on the Built Environment vol 18, © 1996 WIT Press, www.witpress.com, ISSN 1743-3509

- There is usually a high traffic density, and therefore energy consumption becomes a relevant aspect in the daily operation of the system
- Trips are rather short, so ATO commands can be efficiently tailored for the track gradients and the ATP (Automatic Train Protection) speed limits

The design of the ATO commands can be done with the help of a CAD tool with detailed models of the track and the trains, such as the one described in Cuadra[1]. However, the resulting trial-and-error process is time-consuming, and in some cases good solutions are not easy to find.

This paper is focused on the development of two computer tools that perform the automatic optimisation of the ATO commands for a required target time. Each tool is devoted to a particular ATO strategy, and the different definition of the ATO commands leads in each case to a different optimisation problem. Direct-search techniques have been used in both problems, in order to allow the use of detailed models. Mathematical optimisation techniques restrict the level of detail that can be used in the train and line models, as in Jiaxin[4].

The paper is organised as follows. In section 2, the two optimisation problems are stated. In section 3, the general optimisation methodology applied to both problems is introduced. Section 4 and section 5 describe how the general methodology is applied to each problem. Finally, section 6 compares the results of the computer tools over a set of four test cases.

2 Statement of the problems

Both optimisation problems can be stated as follows. Given:

- A detailed description of the train, including mass, air resistance factors, torque-speed characteristics, current characteristics and the available acceleration commands (see Cuadra[1]),
- A detailed description of the line, including gradients, tunnels, curves and ATP speed limits (see Cuadra[1]),
- A departure station. Next station is assumed to be the destination,
- A target trip delay with respect to the maximum-speed trip,
- ATO operation constraints,
- User-defined optimisation parameters,

find the optimum set of ATO commands, such that the target trip delay is met and the energy consumption is minimum.

Different ATO strategies lead to different optimisation problems. The two problems addressed in this paper are described below.

2.1 Coasting-remotor cycles

In this application, actually used in metro lines, a set of ATO commands consist of at most three coasting-remotor commands plus a braking parabola.

Each coasting-remotor command is defined by:

- Initial space for the coasting-remotor command to be applied.
- Coasting speed. When reached, traction is turned off until remotor speed.
- Remotor speed. When reached, traction is turned on until coasting speed.

Some operation constraints are:
- Minimum length of the coasting-remotor command
- Minimum step between different initial spaces (precision)
- Minimum step between different speeds (precision)

2.2 Space-related traction commands

This is a more general driving policy, prepared to be adapted to any ATO equipment. A set of ATO commands consists of a list of traction commands plus a braking parabola. Each traction command defines:
- Type of traction (acceleration, braking, coasting, regulation)
- Desired value of the acceleration (only valid for acceleration and braking)
- Length of the command (it can be expressed either in space or in time)

Some operation constraints are:
- Minimum length of each command (in space or time)
- Minimum step for the length of a command (precision)

3 A general methodology for direct-search optimisation: SMO

SMO (Structured Multi-attribute Optimisation) is a general methodology to approach complex optimisation problems by direct search techniques. Such problems may be characterised by several properties:
- The number of independent variables is not only large, but variable.
- The independent variables are of diverse nature, not only mathematically (continuous or discrete) but also in their relative importance.
- There are multiple attributes (objectives) to be optimised simultaneously.
- The models used to evaluate solutions include tables, curves and/or simulation procedures. This fact suggests the use of direct-search methods.
- There is uncertainty in the models that affects the decision process.

SMO is basically a generalisation of direct-search optimisation methods. The idea is to combine the human designers ability to exploit the natural structure of problems, with the computing ability of machines, evaluating considerably more alternative solutions than human designers do.

SMO has been successfully applied to a great variety of complex engineering problems: design of electrical machines Cuadra[5,6], allocation of 3_D objects in containers Cuadra[7] and control of railway systems Fernández[2,3].

The major steps suggested by the methodology to approach a problem are:

1) Define the <u>attributes</u> that will be used to evaluate each solution (including in some cases the violation of specific constraints).
2) Define a procedure in charge of computing the attributes for each solution, including most of the equality constraints (<u>evaluation function</u>).
3) Analyse the inputs to the evaluation function. Select which ones are the <u>independent variables</u>. The rest of the inputs are considered <u>parameters</u>.
4) Define a systematic <u>search for candidate solutions</u>. The two basic types of procedures are <u>local</u> search and <u>tree</u> search. Although the resulting procedure can be inefficient, trying to define it helps in steps (6) and (7).

5) Define multi-attribute selection criteria in order to reject "dominated" solutions. The result of the optimisation process should be a list of non-dominated candidate solutions, among which a choice can be made.

The direct-search multi-attribute optimisation problem is already stated. Unfortunately, complex problems usually require a huge amount of time to obtain acceptable results. The methodology suggests the following steps:

6) Define several optimisation phases to find the solution of the problem. Successive optimisation phases increase the level of detail of both the evaluation function and the independent variables. At the same time, the degrees of freedom are systematically reduced from phase to phase, and the solutions of a given phase are used as initial points in the next one.
7) Decompose each optimisation phase into partial optimisations. Each one should be a simple and well-behaved optimisation problem by itself. Try to exploit the internal structure of the original problem.

These two last steps resemble the way engineers approach design: several phases of increasing detail, together with the definition of bounded sub-problems (subsystems) that are ordered to more specialised design teams.

4 Problem I. Coasting-remotor cycles.

The solution to this problem is fully described in Juan[8]. A short overview of the project is included below.

4.1 Application of the SMO methodology

The major steps of the general methodology are applied as follows:

1) The attributes are energy consumption and the deviation from the required trip time (both to be minimised).
2) The evaluation function is the same procedure used in Cuadra[1] to simulate the movement of a train.
3) The independent variables are the coasting-remotor commands and the braking parabola. The parameters are all the data about train and line.
4) The search for candidate solutions is a local search performed in the 10-D space defined by the three coasting-remotor commands (with three independent variables each) plus the braking parabola.
5) The multi-attribute problem is transformed into a mono-attribute one by reducing time deviation to a penalty.
6) Two optimisation phases plus an intermediate process have been defined. The first phase scans the space of search, selecting good initial points for the second one. The intermediate process de-saturates these points in order to avoid "flat-optima". Some independent variables are kept constant in the second phase.
7) Two simple partial optimisations have been defined at the first phase. The "master" process searches for the optimal number of coasting-remotor commands, while the "slave" finds the optimal initial spaces, speeds and braking parabola for a fixed number of commands.

4.2 Optimisation algorithm

The optimisation problem is ill-conditioned, since the behaviour of the train is not always sensitive to changes in the independent variables. This is why two optimisation phases and an intermediate process have been implemented.

In the first phase the space of search is scanned, points evaluated and selected with multi-attribute criteria (energy/time, figure 1). The number of commands and the braking parabola of each point are kept constant after this phase.

A de-saturation intermediate process is needed because each solution is evaluated and selected by the behaviour of the train, but sometimes that behaviour is not directly related to the value of the independent variables. The de-saturation process consists in changing the value of the independent variables until the behaviour of the train is sensitive.

The final phase performs local searches from the set of initial points. The local search process is a variation of the Hooke & Jeeves algorithm, Hooke[9]. It is a hill-climbing algorithm made of exploration+advance steps on the space of search, in which the size of the steps is reduced when a local optimum is found.

4.3 Implementation

The implementation has been adapted to an actual ATO application. It finds four optimal trips for different time delays, sharing the same scanning phase.

The tool uses the same simulation routines as in Cuadra[1], the same databases and the same graphical user interface to access the databases. Figure 1 shows the graphical information displayed on-line during the two optimisation phases.

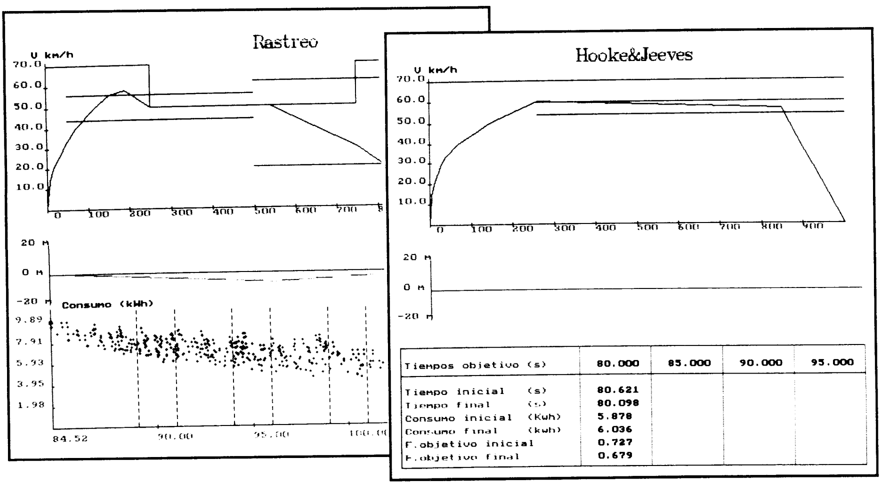

Figure 1. Graphical display in the two phases

The software has been designed using the ANA methodology. It has been implemented in ANSI-C language, on a PC platform, using the Borland C++ environment. It solves an average problem in about 20 seconds on a 486 PC.

5 Problem II. Space-related traction commands

This is a more general approach to ATO, that can be adapted to any kind of train equipment. It is assumed to be implemented as space-related commands to increase the robustness of the solutions. However, time-related commands are internally used in the optimisation process. The solution to this problem is described in Herrero[10]. A short overview of the project is included below.

5.1 Application of the SMO methodology

The major steps of the general methodology are applied as follows:

1) The attributes are energy consumption and the deviation from the required trip time (both to be minimised).
2) The evaluation function is dynamically computed after every new traction command, including some simulation routines from Cuadra[1]. The speed limits imposed by ATP systems are considered in the function.
3) The independent variables are the list of traction commands and the braking parabola. The parameters are all the data about train and line.
4) The search for candidate solutions is a depth-first tree search in which nodes are dynamic train states (space, time, speed, energy consumption).
5) The multi-attribute problem is transformed into a mono-attribute one by reducing time deviation to a penalty.
6) A user-defined number of optimisation phases can be used in the process. Each phase is characterised by a particular precision and minimum length of the traction commands. The optimum trajectory found in a given phase is used to bound the shape of the next-phase train trajectories.

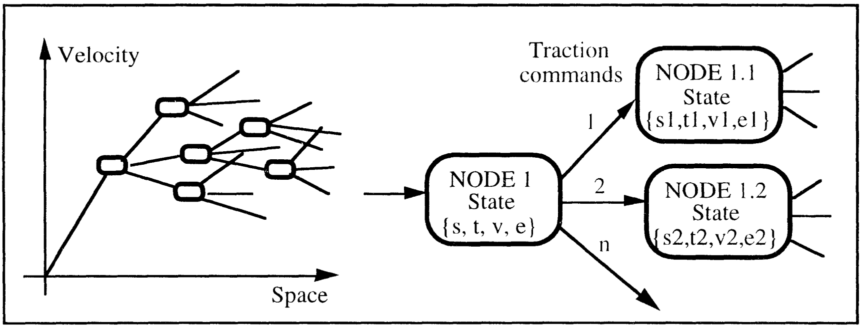

Figure 2. Tree-search representation

5.2 Optimisation algorithm

The optimisation algorithm is a depth-first tree search. Figure 2 shows how a node of the tree represents a dynamic state of the train (space, time, speed and energy consumption), and how the expansion of a node generates child nodes applying different traction commands to the evaluation function.

Transactions on the Built Environment vol 18, © 1996 WIT Press, www.witpress.com, ISSN 1743-3509

A list of arcs of the tree defines a train movement trajectory. When a trajectory meets a precalculated final braking parabola, the braking process is followed until the end of the trip, generating a "leaf" node. Leaf nodes represent full trajectories that are candidates to replace the current optimum.
Three types of strategies have been applied in order to reduce combinatorial explosion, and therefore computing time:

- User-defined parameters to reduce the size of the tree. Precision is the first one, defined as a minimum time step. Moreover, the minimum length of the traction commands, imposed to increase trip comfort, also reduces the tree.
- Heuristics that allow an early pruning of branches. A depth-first search reduces the need for memory resources and allows an early finding of an acceptable optimum. Incomplete trajectories can then be compared with the optimum, and rejected for some reason. A coasting-until-braking command provides, for instance, a lower bound for energy consumption, while an acceleration-until-braking command provides a lower bound for time delay.
- User-defined optimisation phases. The trajectories in the first phase are unbounded, while the trajectories in the rest of the phases are limited by upper and lower bounds. These bounds are calculated as maximum deviations from the previous-phase optimal trajectory.

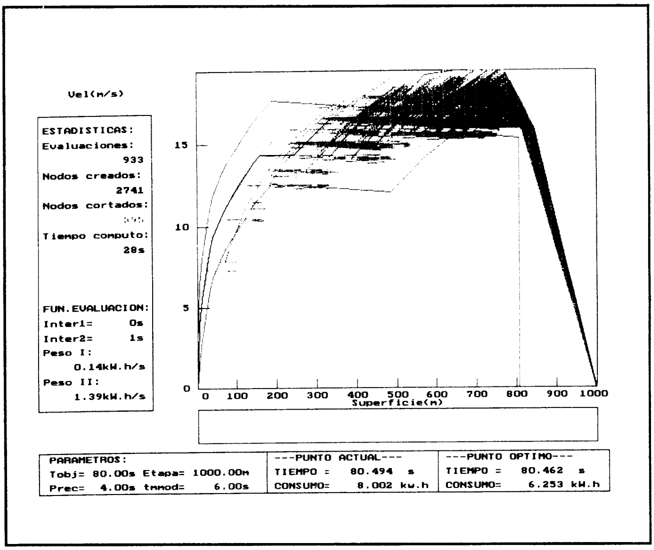

Figure 3. Graphical display in one optimisation phase

5.3 Implementation

The tool uses some simulation routines from Cuadra[1], the same databases and the same graphical user interface to access the databases. Figure 3 shows the on-line graphical information displayed during one optimisation phase.

The software has been developed as in Problem I: ANA methodology, ANSI-C language, PC platform and Borland C++ environment.
Concerning computing time, the tool solves an average problem in about 2 minutes, running on a 486 compatible PC. However, this time is highly dependent on the user-defined parameters and the on-line graphical display.

6 Comparative results

The differences in the software and in the ATO strategies are emphasised by proposing the same problems to both tools. Simple test cases have been selected in order to have predictable results, although the power of these computer tools is more appreciable in complex cases, both in time-saving and in the quality of the solutions.
The same train model has been used in all the test cases, derived from the data of an actual train. The same trip length has been selected (1000 m). The same time delay with respect to the maximum-speed trip has been required (5 s). Four test cases have been defined, as a result of combining:

- Two gradient profiles, downhill-uphill and the opposite.
- Two ATP maximum speed profiles, fast-slow-fast and the opposite.

6.1 Coasting-remotor cycles

Figure 4 shows the results for the four test cases given by the first tool. The train trajectories are conditioned by the absence of regulation in the ATO strategy. The computing time is quite acceptable (about 20 seconds).

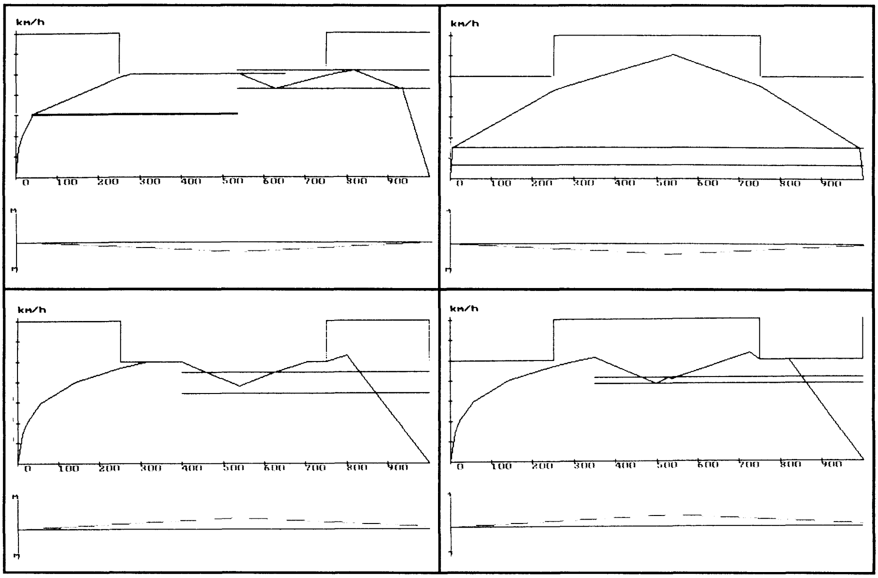

Figure 4. Trajectories defined by coasting-remotor cycles

6.2 Space-related traction commands

Figure 5 shows the results for the four test cases given by the second tool. It must be pointed out that train trajectories may sometimes be conditioned by the user-defined parameters. The computing time is longer (about 3 minutes), but compared with the results of the second tool, energy-saving is about 5% less. An additional advantage is that comfort-related constraints are easy to be implemented in the search process (like forbidden acceleration-braking transitions).

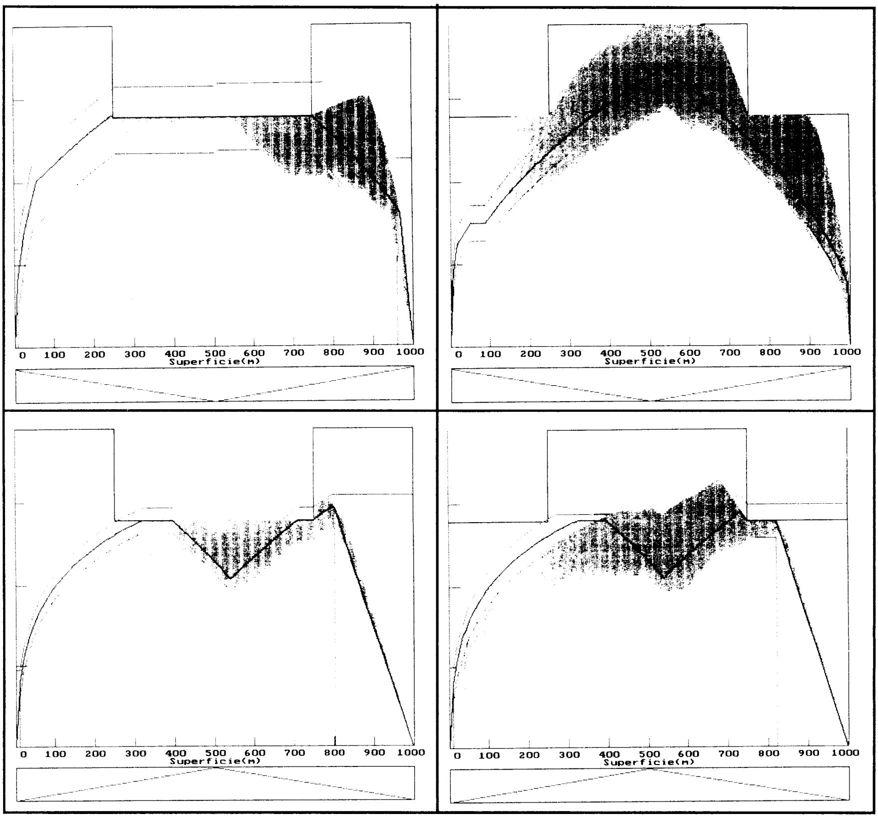

Figure 5. Trajectories defined by traction commands

7 Conclusions

The importance of the ATO strategy in the optimisation process and in the results is remarkable. The definition of detailed models for the movement of the train is critical, especially motor efficiency models and the capability of energy recovering in the braking process. The second tool is more general in the sense that it can be adapted to any driving strategy. For instance, coasting-

remotor cycles allow the definition of regulation in some applications, and this feature would require changes in the simulator of the first tool.
Next step is the development of a flexible and user-friendly commercial tool. A possible approach is to state the problem with space-related traction commands and then translate the result to any other ATO type of commands.

References

1. de Cuadra, F., Fernández, A. & Granados, J.C. *"Train simulation and headway calculations: an approach based on parametrised continuous curves"*. Computers in Railways IV - Vol. I - Railway Design and Management, pp 411-418. Ed. T.K.S. Murthy, B. Mellit, C.A. Brebbia, G, Sciutto, S. Sone. Computational Mechanics Publications, Southampton, 1994.
2. Fernández, A., de Cuadra, F. & Montes, F. *"Traffic regulation and simulation. A predictive adaptive control system"*. Computers in Railways IV - Vol. I - Railway Design and Management, pp 419-428. Ed. T.K.S. Murthy, B. Mellit, C.A. Brebbia, G, Sciutto, S. Sone. Computational Mechanics Publications, Southampton, 1994.
3. Fernández, A., de Cuadra, F. & García. A. *"SIRO: an optimal regulation system in an integrated control centre for metro lines"*. COMPRAIL 96, Berlin, Germany, 21-23 August 1996.
4. Jiaxin, C. & Howlett, P. *"A note on the calculation of optimal strategies for the minimisation of fuel consumption in the control of trains"*. IEEE Transactions on Automatic Control, November 1993, Vol. 38, no. 11, pp. 1730-1734.
5. de Cuadra, F., Pérez Arriaga, J.I. & Lang, J.H., *"The application of knowledge engineering to the computer-aided design of optimal electric drives"*. Proceedings of the International Conference on Electrical Machines, Cambridge (MA) USA, August 1990.
6. de Cuadra, F., Conejo, A., Pérez Arriaga & J.I. "Structured Multi-attribute Optimisation: concepts and applications". Third Stockholm Optimisation Days, Stockholm, Sweden, June 1992. Technical paper IIT-93-042.
7. de Cuadra, F., León, J.J. & Solari, G. *"Application of knowledge engineering to a complex optimisation problem: the allocation of payloads in the HERMES cargo bay"*. Proceedings of the Workshop on Artificial Intelligence and Knowledge-based Systems for Space, ESTEC, Noordwijk, The Netherlands, May 1991.
8. de Juan, J. *"Desarrollo de una herramienta automática para la optimización de la conducción de trenes por vectores de deriva"*. Proyecto Fin de Carrera, ICAI, Universidad Pontificia Comillas, 1995.
9 Hooke, R. & Jeeves, T.A. *"Direct Search. Solution of numerical and statistical problems"*. Journal of the A.C.M., 1961, pp. 212-229.
10 Herrero, M.A. *"Desarrollo de una herramienta para la optimización automática de la conducción continua de un tren"*. Proyecto Fin de Carrera, ICAI, Universidad Pontificia Comillas, 1995.