Computer modelling of the automotive energy requirements for internal combustion engine and battery electric-powered vehicles

J.R. Bumby, B.Sc., Ph.D., C.Eng., M.I.E.E., P.H. Clarke, B.Sc., Ph.D., C.Eng., F.Inst.E., and I. Forster, B.Sc.

Indexing terms: *Measurement and measuring, Instrumentation and measuring science, Computer simulation*

Abstract: In the paper the road vehicle simulation package Janus, developed in the Engineering Department at Durham University, is described. Janus is a flexible simulation package that allows internal combustion engine vehicles, electric vehicles and hybrid vehicles to be simulated, and their performance and energy consumption evaluated over standard driving cycles. The simulation techniques used in these programs are described and the simulation program shown to produce results comparable with experimental data.

List of symbols

a = vehicle acceleration, m/s^2
A = vehicle projected frontal area, m^2
$BMEP$ = brake mean effective pressure, kPa
cc = engine cubic capacity, cm^3
C_d = aerodynamic drag coefficient
C_D = discharged ampere-hours. A h
C_r = coefficient of rolling resistance
C_R = engine compression ratio
C_1, C_2, C_3 = motor loss constants
d = distance travelled, km
E = EMF, V
E_D = battery energy density, kJ/kg
E_s = battery open-circuit terminal voltage when fully charged, V
E_t = battery terminal voltage, V
f_T = cold engine fuel flow factor
g = acceleration due to gravity, 9.81 m/s^2
I_a = armature current, A
I_b = battery current, A
I_f = field current
I_i = battery discharge current
I_w = wheel inertia, kg m^2
k_1 = DC motor EMF constant
k_2 = DC motor torque constant
K = polarisation resistance, Ω
n = rotational speed, rev/min
n = Peukert index
P_{aL} = DC machine armature conduction loss, W
P_{bL} = DC machine brush loss, W
P_{cL} = DC machine core loss, W
P_{fL} = DC machine field resistance loss, W
P_i = input power, W
P_{mL} = DC machine mechanical loss, W
P_{sL} = DC machine stray loss, W
P_{Di} = battery power density, W/kg
P_{GBL} = gearbox losses, W
Q = battery ampere-hour capacity, A h
r_D = wheel rolling radius, m
R = ohmic resistance, Ω
SOC = battery state of charge
t = time
T = ambient temperature, °C
T_a = airgap torque, N m
T_{CB} = engine compression braking torque, N m
T_E = tractive effort at the road wheels, N
V = vehicle velocity, m/s
V_a = armature voltage, V
V_b = battery voltage, V
V_w = head (or tail wind) velocity m/s
W = vehicle mass, kg
W' = modified vehicle accelerative mass, kg
Δt = time interval
η_c = chopper efficiency
η_{cb} = battery charge efficiency
η_{gb} = gearbox partial efficiency
τ_{ci} = battery discharge time at constant current I_i, h
τ_{pi} = battery discharge time at constant power density P_{Di}, h
ρ = air density, kg/m^3
ϕ = hill severity, degrees
ϕ_f = field flux, Wb

Suffixes

max = maximum
a = armature
f = field
i = discharge rate
5 = 5 hour discharge rate
k = step number
pu = per unit

NB cc denotes engine cubic capacity, 1 cc ≡ 1 cm^3

Paper 4048A (S1), first received 1st November 1984 and in revised form 1st May 1985

The authors are with the Department of Engineering, University of Durham, Science Laboratories, South Road, Durham DH1 3LE, United Kingdom

1 Introduction

In 1976 the Department of Industry commissioned The International Research and Development Co. Ltd., Newcastle upon Tyne, to produce a worldwide survey of hybrid electric vehicles [1–3]. This report, prepared by A.J. Mitcham and J.R. Bumby, described numerous types of hybrid vehicles and their operating philosophies. It was evident that hybrid vehicles, and hybrid electric vehicles in particular, could be designed to meet a number of different operating objectives. However, as to which was the best design to use to meet a particular operating objective was much less obvious. Consequently, the report recommended that initially the performance potential of different hybrid vehicle designs should be examined using computer simulation.

Stimulated by this recommendation, this paper describes a general road vehicle simulation program for evaluating the performance and energy efficiency of internal combustion engined vehicles, battery electric vehicles or hybrid electric vehicles. This simulation package was conceived as a user-friendly interactive program, capable of evaluating the performance of a vehicle that is characteristic of a particular vehicle group, e.g. small car, medium car, light commercial van etc. The program was given the name Janus. However, during the development of Janus, it became evident that additional software options must be included to allow the user to define precisely the vehicle to be simulated, thereby extending the program to specific vehicle studies. Indeed, it is this option that is used in program verification.

Since the publication of the Mitcham and Bumby report, the use of computer simulation has played an increasingly important role in evaluating the potential of new vehicle technologies, and in the design of specific vehicles [4–6]. A number of simulations have been written specifically for one type of vehicle [7–10], while others, such as ELVEC [11–12] and HEAVY [13–14], are general road vehicle simulation packages. The majority of this published work has been in, and from, the USA, and this is particularly true of the two general packages ELVEC and HEAVY, both of which are powerful design tools. One of these, HEAVY, has been mounted on the SERC computing network, but, at the present time, cannot be run interactively. In its conception, Janus was designed to be an interactive simulation package directly applicable to European vehicles.

In this paper, the characteristics of Janus are described in detail and used to demonstrate how a particular vehicle can be quickly and easily assembled from a standard subroutine library. Initially the fundamental equations describing the vehicle dynamics are reviewed. In the final part of the paper, the use of Janus in simulating the performance of both electric and internal combustion engined road vehicles is demonstrated. In the case of the electric vehicle, this is achieved by simulating the General Electric ETV-1 electric car [15]; while when considering the internal combustion engined vehicle its usefulness in examining new vehicle concepts is demonstrated by investigating the influence of 'fuel off at idle'.

2 Vehicle dynamics

To provide the necessary propulsion power, any vehicle drive train must be able to provide sufficient tractive effort at the road wheels to overcome aerodynamic drag, rolling resistance and hill gradient effects, while still providing the necessary vehicle acceleration. Consequently, at any particular velocity and acceleration, the net tractive effort required at the road wheels can be expressed as the algebraic sum of these components, i.e.

$$T_E = T_d + T_r + T_g + T_a \text{ N} \tag{1}$$

where the respective components of tractive effort are

$$T_d = 1/2 \rho C_d A (V + V_w)^2 \text{ N} \tag{2}$$

$$T_r = C_r W g \text{ N} \tag{3}$$

$$T_g = W g \sin \phi \text{ N} \tag{4}$$

$$T_a = W a \text{ N} \tag{5}$$

and

C_d = drag coefficient
A = vehicle frontal area, m^2
C_r = coefficient of rolling resistance
W = vehicle mass, kg
ϕ = hill severity and percentage grade = 100 tan ϕ, %
V = vehicle velocity, m/s
V_w = head wind velocity, m/s
a = vehicle linear acceleration, m/s^2
ρ = density of air = 1.226 kg/m^3 (at 15°C and 10^5 Pa (1 bar) ambient conditions)

In eqn. 3 the coefficient of rolling resistance is dependent on the type of tyre used, the tyre pressure and the vehicle speed. However, it is difficult to quantify reliably for vehicle simulation studies, and a number of authors have used a quadratic, velocity-dependent equation for rolling resistance [7, 9, 11, 16], while others assume the coefficient of rolling resistance to be a constant [17]. In general, the variation in the coefficient of rolling resistance with velocity is small up to about 90 km/h, with the major changes occurring at speeds significantly greater than this. Indeed, in tests on the General Electric ETV-1 electric car the energy required to overcome rolling resistance was observed to decrease with speed [15], possibly due to increased tyre heating at higher speeds. To allow for simulation flexibility the coefficient of rolling resistance used in Janus contains both a constant and a velocity-dependent term, both of which can be defined by the user.

The accelerative tractive effort in eqn. 5 relates solely to the linear acceleration of the vehicle and takes no account of the rotational inertia of the road wheels and engine. The inertia of the road wheels can increase the effective vehicle weight by about 2% during acceleration, and is included in eqn. 5 by using an effective accelerative weight W' given by

$$W' = W + \frac{I_w}{r_D^2} \text{ kg} \tag{6}$$

where
I_w = inertia of the road wheels, kg m^2
r_D = rolling radius of the wheel, m

The engine inertia is not included directly in the tractive effort but as a local energy demand within the engine algorithm itself. It is interesting to note that, if referred to the wheels through the overall gear ratio, the engine inertia can increase the effective accelerative weight by about 20% in first gear and 2% in top gear.

3 Janus—its structure and operation

A fundamental feature of Janus is the use of separate Fortran subroutines to represent individual vehicle components. The necessary subroutines can then be easily assembled into a master program to simulate a particular type of vehicle. This approach is adopted as many of the components required by the conventional internal combustion engined vehicle are encountered in an electric vehicle while components used in both these appear in the hybrid electric vehicle. Each component block is written to ANSI Fortran VII standard. The master program can also include additional Fortran statements if required. Such versatility is of prime importance, as it allows the user to increase the degree of output information to suit his own particular requirements, or to restructure the control commands when simulating a hybrid vehicle.

The advantages of this block structure approach can be demonstrated by considering the layout of the internal combustion engined vehicle shown schematically in Fig. 1.

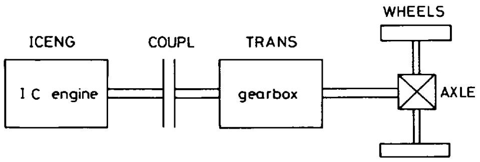

Fig. 1 *Conventional internal combustion engine vehicle drive train*

The software blocks representing this vehicle are shown in the flow diagram of Fig. 2, and it is only the components within the dotted frame that are assembled by the user. To simplify program writing, the names of the software blocks have been selected to be similar to those of the components they represent. For operational flexibility each subroutine is divided into three sections termed, respectively, the initial section, dynamic section and output section. The initial and dynamic sections are further divided into two subsections. During the initialisation phase, each of the component subroutines is entered in turn, and the vehicle power train and driving cycle parameters specified. Unfortunately, in some subroutines information may be required from components upstream of the one currently being accessed, and as such will have yet to be defined. To overcome this problem a two pass system is used. On the first pass full details of the individual components are specified, while on the second pass any calculations requiring information from an upstream component are completed. Such a system is necessary, for example, when automatic weight generation is used. In such circumstances, the weight of the gearbox and final drive depends on the specified engine torque and power, one of the last components to be defined.

Having established the initial operating conditions the simulation enters the dynamic section, the main computational part of the program. In this part of the program, the fuel efficiency of the vehicle and all the component efficiencies are calculated. For the majority of driving cycles, such as the ECE-15 urban cycle shown in Fig. 3, the vehicle velocity is explicitly defined as a function of time,

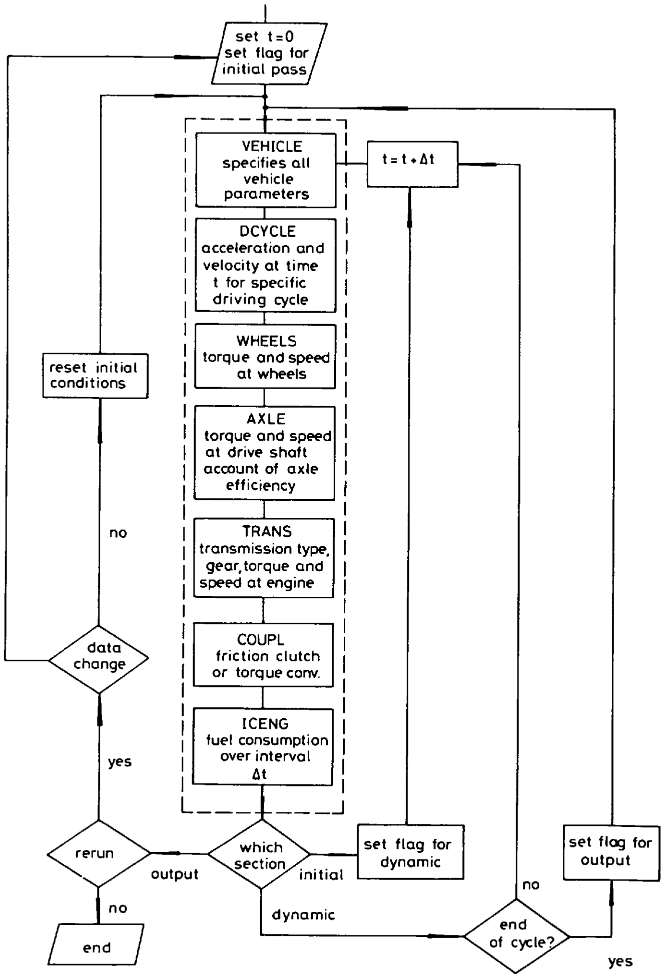

Fig. 2 *Flowchart for a conventional internal combustion engine*

enabling the program to step through the driving cycle at one second intervals (default value) calculating vehicle

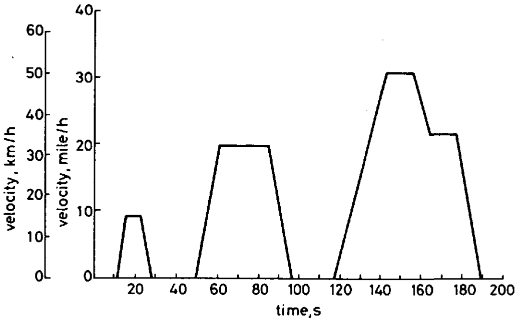

Fig. 3 *ECE-15 urban driving cycle*

velocity and acceleration directly from the driving-cycle data. The tractive effort at the road wheels is calculated at each time instant using eqn. 1 and converted into a torque and rotational speed demand in the subroutine WHEELS. This torque requirement is then reflected back through each of the drive train components to the engine output shaft. At each stage, account is taken of the gear ratio and instantaneous loss within each of the transmission components. Fuel usage is now obtained from the engine fuel map by assuming the engine load to be constant over the step interval. By sequentially repeating this process, the total fuel used over the driving cycle can be found.

Such a direct calculation method is adequate for the majority of situations, as the performance of each drive train component can be considered to be independent of the other components. Normally the component efficiency will depend only on the torque and speed it is required to transmit at that time instant. However, in some cases, the performance and behaviour of two or more of the drive train components may be strongly dependent. For example, in an electric vehicle, the behaviour of both the electronic controller and the traction motor depends on the battery terminal voltage. In turn, the battery terminal voltage depends on load current. In such instances, it may be necessary to iterate around the vehicle drive train until a stable operating condition is achieved, when component efficiencies can then be calculated. To accommodate this, a two-pass system is used at each time step. On the first pass, all the vehicle variables such as torques, speeds etc. are calculated from the road wheels to the energy source, using an iterative process to establish the stable operating condition. Once the stable operating condition has been reached the second dynamic pass is started, in which the fuel used during the last time step and the energy loss in each of the drive-train components is recorded. The time is then updated and the process repeated until the driving cycle is completed.

Once the driving cycle is completed, the simulation enters the output section, where full details of the vehicle, driving cycle and the individual drive-train components is given with graphical presentation of time-varying quantities, engine fuel maps etc. Besides displaying component efficiencies, losses and the overall vehicle fuel economy, the percentage of the total cycle time spent in each area of the engine fuel map is also given. Such fuel map information is invaluable, particularly when detailed studies on the effect of the vehicle component sizing and control on fuel efficiencies are being undertaken. Alternatively, if a detailed component breakdown is not required, a reduced output facility may be selected which simply details the vehicle range and/or fuel consumption when operated continuously over a particular driving cycle.

Once a simulation run has been completed, the controlling software allows further runs to be conducted over a different driving cycle. Alternatively, modifications may be made to the individual power-train components and/or the vehicle parameters.

The different component subroutines available within Janus are listed in Table 1 and, in many cases, contain

Table 1: Component subroutine names

Component	Simulation name
Vehicle definition	VEHICLE
Driving cycle	DCYCLE
Wheels	WHEELS
Final drive	AXLE
Transmission	TRANS
Clutch or torque convertor	COUPLE
Internal combustion engine	ICENG
Series DC motor	DCSER
Separately excited DC motor	DCSHUNT
DC switched reluctance motor	DCREL
AC inductance motor	ACINDUC
DC generator	DCGEN
Field chopper	FCHOPR
Armature chopper	ACHOPR
Traction battery	BATTERY
'Gearing' for connecting two prime movers	DRIVE
Battery switching	BATSWCH
Summing block	SUM
Torque splitting module (hybrids)	TORQSPLT
Power splitting module (hybrids)	POWSPLT
Vehicle controllers	VEHCONT

performance details and simulation algorithms on a number of related items. For example, the tree structure of Fig. 4 describes the specific components available within the transmission module, any of which can be selected from within the program.

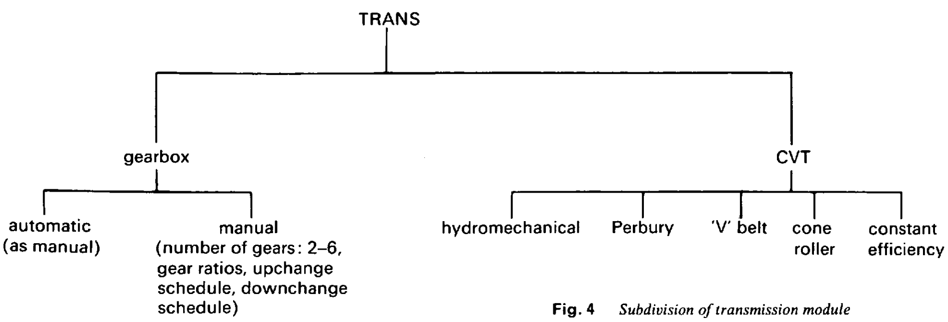

Fig. 4 *Subdivision of transmission module*

4 Component simulation

4.1 Transmission

The transmission subroutine (TRANS) simulates the behaviour of the vehicle transmission system and includes models of manual and automatic, fixed ratio, gearboxes and a variety of continuously variable transmissions (CVTs). The fixed ratio gearbox model is based on an efficiency algorithm that predicts the gearbox efficiency, depending on the transmitted torque and speed. The algorithm also allows the number of gears, gear ratios and the gear change schedule to be specified. Both the up-change and the down-change schedule can be specified as a function of vehicle speed or, alternatively, the ECE-15 gear change schedule may be used. As the time taken during the gear change is small compared with the total cycle time, the simulation assumes instantaneous gear changes.

The efficiency of a gearbox depends on the power transmitted, the operational speed, the gear ratio and the gear profile and will vary with different gearbox designs. Although efficiency is high at normal full-load operating conditions, typically in excess of 96%, at part load, particularly high-speed part-load operation, efficiency drops and can significantly effect the vehicle fuel economy.

Gearbox losses can be broadly divided into two components, a no-load loss due to gears churning in the gearbox oil and a load friction loss due to the transfer of load between the gears. This last component of loss is also dependent on the gear profile. Such losses are approximated in Janus by the power loss expression

$$P_{GBL} = (1 - \eta_{GB})P_i + 1.14 \times 10^{-8} n^v \text{ W} \tag{7}$$

where P_i is the gearbox input power and η_{GB} is the partial efficiency relating to the friction loss. The second velocity-dependent term is associated particularly with the churning loss. The friction loss component is dependent on gear ratio and in the simulation is changed such that nonmeshing gears, i.e. gear ratio of 1 : 1, have a higher efficiency than the meshing gears. Typically, η_{GB} is taken as 0.99 for nonmeshing gears and 0.98 for meshing gears. For an automatic gearbox, a single value of 0.97 is assumed.

At high rotational speeds, particularly at low load, the value of the churning loss speed index v becomes critical as gearbox efficiency is dominated by this loss term. To obtain gearbox efficiency values typical of those quoted in the literature [17, 22], a speed index of 2.1 is assumed.

The transmission is coupled to the engine by the COUPLE routine, which contains models for both a torque convertor and a friction clutch.

Efficiency curves similar to those described by eqn. 7 are also used in the final drive model. However, as the final drive may be constructed from either bevel or spur gears, a different friction loss component is required for each gear type, bevel gears having a higher frictional loss. The final drive simulation block also contains efficiency values representative of both chain and belt drives, as these have been used previously in electric vehicles [15, 20].

4.2 Internal combustion engine

Accurate engine maps giving the specific fuel consumption at different loads and speeds are essential if fuel consumption over urban driving cycles is to be accurately computed. Such fuel maps are commonly presented, either in a power/speed, brake-mean-effective-pressure (BMEP)/speed, or torque/speed form. An example of the latter is shown in Fig. 5 for a typical 1850 cc engine. For computational purposes the fuel map is divided into twenty BMEP and speed increments and stored as a 20 × 20 two-dimensional array. At each time interval through the cycle,

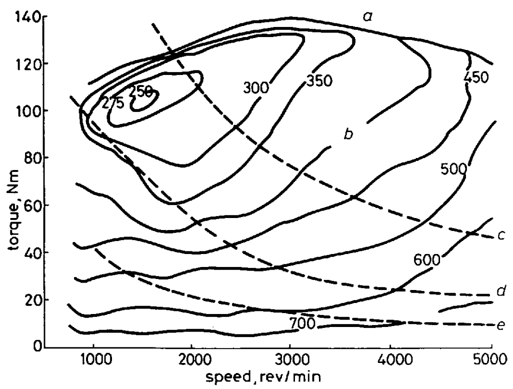

Fig. 5 *Engine fuel map for a typical 1850 cc engine*
See Reference 18
a Maximum torque curve
b Constant SFC 400g/kWh
c Power for constant 115 km/h
d Power for constant 80 km/h
e Power for constant 50 km/h

the fuel consumption is calculated for the particular load and engine speed by linearly interpolating between the four specific fuel values adjacent to the operating point. The position within the fuel map array is then recorded. At the end of the cycle simulation, the total fuel used is calculated and the fuel usage information presented graphically in the form of a fuel usage map. This map displays that percentage of the cycle time the engine spent in different parts of the operating region. As engine efficiency maps are obtained from steady-state load tests to obtain net engine torque during an acceleration interval, the engine inertia torque must be added algebraically to the output torque. As the engine inertia torque is proportional to the engine angular acceleration, this is readily achieved, except when a gear change takes place. As a gear change is assumed to occur instantaneously, a step change occurs in the engine speed which can lead to very high, false, inertia torques. In general, engine inertia has only a small influence on the vehicle fuel economy; thereby, allowing engine inertia effects to be neglected during the computational time step, the gear change takes place.

Experience has shown that the use of BMEP/normalised speed fuel maps are to be preferred to power maps, as, with the latter, significant errors can be introduced at low-torque low-speed conditions; conditions that are common in urban driving cycles. This arises because under such conditions the power demand is low, possibly less than 10% of maximum power, so that if equal power increments are used in the storage process, accurate specific fuel values corresponding particularly to the lower left-hand corner of Fig. 5 can be lost. By normalising the engine speed, the fuel maps can be 'stretched' to either a different engine cubic capacity, and hence power output, or to a different maximum engine speed other than their design value. Although stretching of the engine maps over too large an engine range, and particularly over a different compression ratio range, is not recommended, it does provide substantial additional flexibility with negligible additional programming complexity.

At some stage during the driving cycle, it is possible for the demand torque to be greater than the full throttle

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase® Smarter legal research.