A review of road traffic simulation methods using a general purpose digital computer

T. Briggs

School of Mathematics and Computing, Leeds Polytechnic, Calverley Street, Leeds LS1 3HE

This paper reviews ways in which the positions of vehicles, as well as their movement along roads and through junctions, can be represented on a general purpose digital computer. The way in which the system moves forward in simulated time is also considered.

A discussion of whether it is worthwhile to simulate a traffic problem leads on to an outline description of some of the available packages.

(Received September 1977)

1. Introduction

In order to gain some understanding of the behaviour of any complex system, it is helpful to identify and define variables, and then to deduce quantitative relationships between these variables. The relationships may be deduced from theoretical considerations, but in any case, they must be supported by observation.

There are three levels of detail at which relationships may be developed, namely the macroscopic level, the stochastic level and the microscopic level. At the macroscopic level, attention is directed to obtaining mean results. The system may be regarded as a black box since there is no attempt to seek the causation of the values obtained. Even though the behaviour of the system may be very variable, the ultimate recording of average values of dependent variables for fixed settings of independent variables ensures that a deterministic relationship is obtained. The work done on the relationship between traffic speed and flow (Davies, 1963; Thomson, 1967; Wright and Hyde, 1972) falls into this category.

At the stochastic level, observations or theory show that a particular dependent variable is not entirely specified when the values of all the independent variables are fixed and, if the system is observed many times, a pattern becomes discernible of the frequency with which the dependent variable assumes each possible value. This frequency distribution of the dependent variable corresponding to specific values of the independent variables implies that the behaviour of the system is not fully explained; the value of the dependent variable has only partially been accounted for by the independent variables. If we were to use the mean value in place of the frequency distribution we should be reverting to the macroscopic level. There may, of course, be more than one dependent variable. An illustration of the stochastic category is work done on gap acceptance (Drew, 1968, pp. 178-197; Salter, 1968).

The third level of detail is the microscopic level. At this level, detail is so complete that relationships are once more deterministic. All the activities within the system are so precisely understood, or at least imitated, that the random element completely disappears. For example, in a road network within a city where the intersections are controlled by fixed-time signals, the state of the lights at any instant is deterministic (within fine engineering limits) at the microscopic level, whereas the time taken by a specified driver to pass through the network each day would appear to be a stochastic variate, and at the macroscopic level, the *average* total delay to traffic in vehicle-hours per hour (averaged over many similar periods) would have a deterministic relationship with the corresponding *average* traffic flow out of the network. An illustration of a microscopic model is one in which vehicles and drivers are so thoroughly classified that their behaviour is entirely predictable.

1.1 *Computerised models*

The advent of electronic digital computers has made it technically and economically feasible to develop stochastic models (Monte Carlo simulations) or to go even further and develop detailed deterministic models (microsimulations) of complex systems; these models are in the form of computer programs.

In a Monte Carlo simulation of traffic, the basic idea is to use a random number to 'prophesy' how long it will take a particular vehicle to get from one end of a road to the other, whereas, in a microsimulation, the basic idea is to simulate the behaviour of each driver and vehicle with such meticulous attention to detail that virtually nothing is left to chance.

In this paper an account will be given of some of the work which has been done, and some of the methods which have been used, in the digital simulation of road traffic. It is beyond its scope to consider the analogue or hybrid simulation of road traffic, such as is undertaken in the Department of Electrical Engineering at the University of Manchester Institute of Science and Technology (Rumsey and Hartley, 1972; Hartley and Green, 1965; Powner, 1970; Terrell, Neville-Smith and Powner, 1974; Grigg and Hartley, 1975).

2. The simulation of traffic in a road

The earliest recorded computer simulation of the movement of traffic along a road was written by Gerlough (1955). He simulated the movement of traffic on a quarter-mile section of a two-lane one-way road.

There are a number of ways in which traffic and its movement along a road may be represented on a digital computer; some of these will be described in the following sections.

2.1 *Binary digit representation of a vehicle*

The position of a vehicle on a road can be roughly represented by the position of a binary digit within a sequence of words (computer memory cells). The movement of the vehicle is simulated by shifting the bit to right or left as appropriate, and from one word to another as necessary. This was the method which Gerlough used in his pioneering work.

When this method of representation is used, the road is treated as if divided into sections which are about as long as the average car; the binary digit is shifted as many places as the number of section lengths travelled by the vehicle in the time interval chosen.

2.2 *Memorandum representation of a vehicle*

Another scheme, used when computer storage is at a premium, is to pack all the relevant information about a vehicle (such as its position, speed and acceleration) into one word. These data are updated at each time increment as necessary. There is, however, no essential change of principle if each piece of information about a vehicle is stored in a separate word; it requires more computer storage, but data can be manipulated more quickly. Whether the information is stored in one or several words, it has been called the 'Memorandum method' by several authors (Wohl and Martin, 1967; Lewis and Michael, 1963).

When several words are used, the information may be organised such that there is an array of data for each vehicle or, alternatively, such that there is an array for each attribute (such as speed).

When the memorandum method is used, there is no need to treat the road as if divided into sections, since the position of a vehicle can be recorded as a continuous variable (or at any rate, to the nearest foot, if integer arithmetic is used for computational efficiency).

2.3 *Recording the relative positions of vehicles in a lane*

Supposing that there is a word or an array containing the information on each vehicle, there are two basic ways in which changes in the relative positions of vehicles may be recorded.

1. *By sorting*

If one vehicle overtakes another, the data for each of the two vehicles may be interchanged by a simple sorting process. The data for vehicles are then stored in the same order that the vehicles have in the lane. Computationally, this is a time consuming process, particularly if several words are used to store the data for each individual vehicle. It would usually involve considerable computation to move a vehicle from one lane to another; any vehicles which would be behind it would all have to have their arrays of data moved in the computer store, if adjacent vehicles in a lane have corresponding adjacent arrays with no unoccupied words between arrays.

Beilby (1972) has developed a novel simulation program, written in machine code for a 16K computer, which has features akin to both the binary bit and memorandum methods. The relative position of each vehicle is represented physically in the computer store not by a binary bit, but by two words in the equivalent of a large array; the space occupied by the two words represents a total distance of about five feet. The two words are moved about the store in the same way that a binary bit would be moved in that method of physical representation. The second word is the memorandum word into which is packed all the relevant coded data for the vehicle. The most novel feature is that the words are not in the data store, as might be expected, but are in the program store. The first word associated with each vehicle is an unconditional jump instruction; the jump is to a series of subroutines, the first subroutine interprets the information in the memorandum word, the second creates the updated memorandum word, and the third moves the two words to represent the physical movement of the vehicle concerned. All other words in the program store representing the empty road (including intersections) are dummy instructions. Such an arrangement, in which the computer scans through the supposed 'machine code instructions' sequentially is quite efficient. Because there are effectively empty spaces between the physical representations of vehicles the amount of sorting required is minimal; sorting occurs only when a vehicle changes lane, and then only to a limited extent because of the simulated gaps between vehicles.

2. *By list processing*

If an extra piece of information is stored with each vehicle's data, namely a 'pointer' to the vehicle immediately following it, there is no need to move data about the store. In principle, the pointer is the address of some specific piece of information concerning the following vehicle; in practice it may, for example, be the identification number of the following vehicle, which can then be used as an array subscript to yield the desired information. (Equally well, of course, each vehicle may also have a pointer to the vehicle ahead of it.)

If a vehicle is moved to another lane it is only necessary to alter the values of the pointers rather than move data about in the computer store.

Dockerty and Brantigan (1969) have claimed to be the first to apply list processing to traffic simulation; Drew (1968, p. 281), however, referred to an unpublished paper by Sandefur in which the latter described this application of list processing. Lewis (1962), as one of the pioneers, developed a model which partially had list processing features. The road was represented as a 'circular array' long enough to accommodate as many vehicles as may be present. When vehicles left the road, references to them were erased from the array, the pointer to the vehicle at the head of the appropriate lane was changed and the record of the number of vehicles in the lane was reduced by one. The entry of vehicles into the road was similarly catered for. However, overtaking was not generally permitted and vehicles appeared in the array in the same physical order as on the road being simulated.

2.4 *Representation of the movement of vehicles along a road*

There are two general methods of simulating the movement of vehicles along a road. One is to simulate the second-by-second behaviour of each individual vehicle, from a knowledge of car following theory and the like; the other is simply to associate a random variate, representing the time taken to travel along the road, with each vehicle. These two methods will be described in the following sections.

2.4.1 *Microsimulation of the movement of vehicles along a road*

The behaviour of each individual vehicle is simulated; this depends on a detailed knowledge of such factors as:

(*a*) The acceleration and deceleration properties of the vehicles on various gradients (Lewis and Michael, 1963; Mahaffy and Dodson, 1973; Lewis, 1970).

(*b*) Behaviour when following another vehicle but not overtaking (Wohl and Martin, 1967; Constantine and Young, 1967; Herman and Gardels, 1963; Lewis, 1970; Lee and Jones, 1967; Ashton, 1966).

(*c*) Behaviour when overtaking, having regard to vehicles travelling in the opposite direction, and vehicles approaching at a greater speed from behind (Herman and Lam, 1972).

(*d*) Behaviour when starting from rest (such as at traffic signals) (Briggs, 1977).

(*e*) Behaviour when coming to rest (Ham, 1967; Cobbe, 1966).

(*f*) Behaviour when intending to turn at the next intersection (Lewis and Michael, 1963; Home Office, 1968).

(*g*) Drivers' reaction times under various conditions (Johansson and Rumar, 1965).

(*h*) The types of vehicle and the proportion of each on the road.

(*i*) The routes taken by vehicles through the system, and the proportions following each route.

From the earliest days the microsimulation approach has been used, beginning with Gerlough's (1955) simulation mentioned in Section 2. The program has been usually written for a specific problem, and each worker has made his own simplifying assumptions (such as forbidding overtaking). A survey of fifteen such American simulations in the period 1955–65 has been given by Fox and Lehman (1967).

2.4.2 *Simple estimation of total time taken to travel along a road*

Rather than perform a microsimulation, it may be adequate merely to set the journey time of a vehicle between one intersection and another at some constant value, with the possibility of adding a random element to this value if thought desirable. This method was used by Bretschneider and Pavel (1965) when studying the flow in a road network. It was also used by Rumsey and Hartley (1972) when simulating the flow between two intersections on Manchester University's special purpose hybrid computer, developed for traffic simulation.

This method could not be used if there were significant variability in journey times in the road section; in particular it might prove impossible to simulate adequately the correlation between the journey times of adjacent vehicles.

2.5 *The entry of vehicles into the road network*

In all simulations, whether they be microscopic or stochastic, the arrival of vehicles at the periphery of the road network is simulated by sampling inter-arrival times from some probability distribution such as the Borel-Tanner (Borel, 1942; Tanner, 1953; 1961) distribution, or by using directly the raw data obtained on site.

To cover the possibility that a simulated arrival cannot get into the system because it is obstructed by vehicles ahead, one of the following approaches may be used.

(*a*) A backlog list for each entry point may be used to accommodate such vehicles until the road ahead clears. For example, Stark (1962), who divided the road into twelve-foot sections, found it necessary to use this method.

(*b*) If list processing ideas are used to store the vehicles as an ordered list in each particular lane, then vehicles which are obstructed from entering the lane at full speed may be added to the ordered list and admitted at reduced speed. However, some caution must be exercised in using this method since the entry speed will affect the vehicle's journey time.

3. The simulation of traffic movement through intersections

The first reported simulation of an intersection was carried out by Goode (1955). Two two-lane roads crossed one another at the intersection and overtaking was prohibited; control was by three-phase signals.

Again, there are two basic methods of simulating movement through an intersection; these are considered in the two subsections below.

3.1 *Microsimulation of the movement of vehicles through an intersection*

A microsimulation of the movement of vehicles through an intersection having regard to opposing traffic, maximum speed when turning, and so on, may be performed, For example, Lewis (1963) used this method when simulating crossroads where a four-lane and two-lane road intersected.

3.2 *Simple estimation of time taken to pass through an intersection in a given direction*

As with the representation of the movement of vehicles along a road, the simplest method is to assume that the time taken, in this case to pass through an intersection, is some constant value, with the possibility of including a random element if thought desirable. Thus, for example, in simulating traffic movement at a T-junction, Aitken (1963) assumed a constant time to traverse a given path through the intersection.

Thomasson and Wright (1967) obtained empirical evidence at three isolated crossroads, where two two-lane roads crossed one another, that the time taken to traverse an intersection followed a normal distribution for each path through it; they made use of this in their simulation program.

4. Methods of advancing time

There are two wellknown methods of moving forward in time in a simulation; one is to move forward in constant time increments (such as one second), the other is to find the time at which the next event will occur and move forward to this time (Naylor, Balintfy, Burdick and Chu, 1966). In either case, all time dependent variables are then updated.

In the case of traffic simulations there is usually so much activity going on that it would involve a great deal of computation to find which event (such as change in a vehicle's lane) would occur next, and furthermore the interval between events would be very small if there were many vehicles in the system; consequently, fixed time increments are normally used. Gerlough (1964) has suggested that although most of a simulation program may use fixed time increments, some parts of the system could be treated by event-to-event scanning within this framework.

In choosing the size of the fixed time increment, account has to be taken of time intervals used in traffic controller settings, of time headways, of reaction times, of gap acceptance times and so on. One difficulty is that most of these times are on a continuous scale whereas the interval between scans is a fixed unit of time. The interval must not be so large that many events may occur between increments, otherwise much of the realism will be lost unless some very involved computations are performed; on the other hand, the smaller the increment the larger the number of steps (i.e. repetitions of the complex process of updating the model) required to simulate traffic movement over a given period of time, such as a minute.

The logic of a simulation program can be made less complicated if two conditions are imposed. Firstly, inter-event times which are constant (or variable multiples of a constant), such as the intervals between changes of state of traffic signals, should be a multiple of the chosen time increment. Secondly, the minimum headway between vehicles entering the system on any road should exceed the chosen time increment.

The most commonly used time increment giving a compromise between these conflicting requirements is one second, although intervals as small as one-tenth of a second and as large as five seconds (Katz, 1963) have been used by some workers.

5. Criteria for deciding the worth of a traffic simulation

One of the most common measures given for a computer simulation of a traffic problem is the ratio of time simulated to the time actually taken by the central processor to simulate this time. Sometimes the merits of different simulation programs have been compared on the basis of such ratios. However, the measure in itself tells us nothing about the worth of a simulation. Rather than compare times it would be more meaningful to compare costs and benefits. We might reasonably ask such questions as:

(*a*) what do we want to find out about a traffic problem, and what benefit shall we get from knowing?

(*b*) can the knowledge be obtained by simulation, and if so, how much will this cost?

(*c*) is there any other method of obtaining the knowledge, and if so, how much will that cost?

Cotton (1973) has made the point that money spent on computing by traffic engineers might sometimes be better spent directly on road improvements, and that local authorities might well incur hidden costs, if they use a computer, by having qualified people at a loose end because the computer has taken over some of their work. He was not thinking specifically, if at all, about simulation.

The availability of good computing facilities and support staff by no means implies that traffic problems should be tackled by computer simulation. For example, if the problem is simply one of obtaining the optimum signal settings, the methods of Webster and Cobbe (1966) or of Newell (1971) may be adequate. If it is considered important that really good settings are obtained (for specific traffic conditions) then Allsop's method (1971; Taylor and Allsop, 1969) could be used or evolutionary operation (Box, Davies and Swann, 1969) of the signals could be put into practice on site: not only would this be far cheaper usually than a one-off computer program but also there would be no need to investigate the validity of the results.

Generally, the cost of a computer simulation would include the following items:

Data collection (including use of special equipment)
Data preparation (e.g. editing and copying on coding sheets)
Data punching and verifying
Program writing
Program punching
Program debugging
Central processor time } These will take account of the capital cost, expected useful life, and running costs of computer and peripherals
Computer file storage }
Output (including its analysis).

If a general purpose traffic simulation package could be used for a particular problem, then it would not be necessary to spend time and money on computer programming (although a charge might be made for its use) and hence there would be a greater likelihood of the simulation approach being economically feasible. One such package, BUSTLE (= Bradford University Simulation of Traffic Language) (Briggs, 1976a; 1976b) has taken three man-years to develop and in a commercial environment would thus be costed at around £20,000.

6. Traffic simulation packages

In the literature, six references to traffic simulation packages have been found which are claimed to be of reasonably general application. Together with BUSTLE, these packages may be roughly divided into three classes, namely: microsimulation packages, stochastic simulation packages and hybrid assignment/simulation packages.

6.1 *Microsimulation packages*

The first microsimulation package, SIMCAR, was developed by Shumate and Dirksen (1964). They described it as a general traffic simulation language. Nowadays it would be called a package since it was only necessary to supply numeric data. The package was not really general, in that it was only capable of simulating the flow of traffic along a single road with 'intersection stubs' here and there. The road was divided into numbered sixteen-foot sections. Gradients were taken into account in determining accelerations. Each driver was given a 'valor factor' which was a measure of his propensity for taking risks.

In principle, vehicles were said to be updated from rearmost to foremost; since the time increment was one second, this was thought to be equivalent to simulating a driver reaction time of one second. In practice, the vehicles were simply updated in descending order of vehicle identification number (for computational simplicity), so that the most recent arrivals were updated first. Where there was no overtaking, principle and practice agreed.

As already mentioned above, Beilby (1971), at the University of Birmingham, has created a package to perform a microsimulation on a small computer of traffic movement at an intersection or along a road.

BUSTLE, if used as a package, permits microsimulation of traffic behaviour in a road network having fixed-time signals at all intersections. Alternatively, it may be used like SIMON (ICL, 1969) as a simulation language. Its features include facilities for data validation, for execution error detection and diagnosis, and for extension of the program.

The purpose of microsimulation packages is to study and simulate traffic behaviour at a fundamental level. Such packages have a place particularly in academic and research institutions where proposed models of car following, overtaking, and so on, can be studied in depth. However, this does not preclude their use in practical applications such as the optimisation of signal settings.

6.2 *Stochastic packages*

At the University of Birmingham, four officers from the Royal Corps of Signals worked on the development of a stochastic simulation program called COSPARTAN (= Comprehensive Simulation Program for Road Traffic Networks). The initial version was written by Passmore (1968). There was a fixed transit time between nodes. Vehicles first entered a transit queue for the appropriate road, and later entered a lane queue for the appropriate lane, prior to entering an intersection. At the intersection, if there was an acceptable gap in opposing traffic, the vehicle would 'flash over' to the next transit queue on its route.

The transit time in the road just entered was increased to compensate for the underestimate of the time to cross the intersection. Story (1969; 1970) extended the program to handle vehicle-actuated signals and priority intersections. Overtaking was permitted in the transit queue, i.e. before a lane had been selected. The time to cross an intersection was set to a constant value inversely proportional to the corresponding saturation flow rate.

Gadd (1971) tested the accuracy (Tillotson, 1971) of COSPARTAN and calibrated it against real data. Cooke (1971) was concerned to extend the use of the program to the simulation of large traffic networks with heavy traffic. Since such simulations used a lot of computer time, he introduced dumping and restart facilities; this made it possible to request fairly short runs of the program, where previously as much as thirty hours may have been requested. Since the storage space required for a large network was large, data had to be overlaid on to magnetic tape and Cooke introduced a scheme for breaking the network into areas and updating the network area by area. The peripheral transfers of data for one area were completed while the calculations involved in updating another area by thirty time increments were being performed. The time to cross an intersection was assumed to be normally distributed, with a standard deviation of one second and a chosen mean.

Work has also been done at the Road Research Laboratory by Needham and others (1970) to produce a stochastic program called ROSIM. Stochastic (i.e. Monte Carlo) simulation packages are much quicker than microsimulation packages because of the lack of detail. Hence they are excellent for obtaining useful results regarding actual traffic problems where macroscopic (i.e. average) results, rather than a detailed knowledge of the progress of individual vehicles, are required.

6.3 *Hybrid assignment/simulation packages*

In hybrid assignment/simulation packages there is even less detail than in stochastic packages. As far as possible, vehicles are treated in clusters rather than individually. The emphasis is on counting vehicles, rather than considering individual vehicles, although reluctantly this latter approach may be used at intersections where priority rules determine which vehicles may move.

Katz (1963) developed a package called TRANS which appears to be best placed in this category. He was concerned with traffic signal settings in Washington. Giannopoulos (1971) intimated that he too was developing such a package at Imperial College, London. His concern was with environmental problems such as pollution.

7. Conclusion

A number of methods of simulating road traffic now exist. Thus, given access to a computer, it is technologically feasible to write a one-off simulation program to help to solve a specific problem.

Before beginning such a task however, one should consider whether there are heuristic methods (Briggs (1976a) contains several references) or a simulation package which would give adequate results more quickly or more cheaply. A cursory cost/benefit analysis should indicate the most appropriate method in the circumstances.

References

Aitken, J. M. (1963). Simulation of traffic conditions at an uncontrolled T-junction, *Traffic Engineering and Control*, Vol. 5, pp. 354-8.

Allsop, R. E. (1971). Delay-minimizing settings for fixed-time traffic signals at a single road junction, *JIMA*, Vol. 8, pp. 164-185.

Ashton, W. D. (1966). *The Theory of Road Traffic Flow*, Methuen Monograph.

Beilby, M. H. (1972). Road traffic simulation on a small computer, *The Computer Journal*, Vol. 15, pp. 134-7.

Borel, E. (1942). *Calcul des probabilités—Sur l'emploi du théorème de Bernoulli . . . . Académie des Sciences*, Paris, Comptes Rendus, 214, pp. 452-6.

Box, M. J., Davies, D., and Swann, W. H. (1969). ICI Monograph No. 5: *Non-Linear Optimization Techniques*, Oliver and Boyd.

Bretschneider, G. and Pavel, G. (1965). Data processing systems simulate flow of street traffic, *Siemens Review*, Vol. 32, pp. 327-30.

Briggs, T. (1976a). BUSTLE: a road traffic simulation package, M.Sc. thesis (Bradford).

Briggs, T. (1976b). *BUSTLE Users' Manual*, School of Librarianship, Leeds Polytechnic.

Briggs, T. (1977). Time headways on crossing the stop-line after queueing at traffic lights, *Traffic Engineering and Control*, Vol. 18, pp.264-5.

Cobbe, B. M. (1966). *Area Control of Traffic*, West London, Theme V: Area Control of Traffic, Eighth International Study Week in Traffic Engineering, Barcelona 5-10 September 1966. World Touring and Automobile Organisation.

Cooke, B. M. (1971). Traffic Simulation—assessment and development, M.Sc. thesis (Birmingham).

Constantine, T. and Young, A. P. (1967). Traffic Dynamics: Car Following Studies, *Traffic Engineering and Control*, Vol. 8, pp. 551-4.

Cotton, W. F. (1973). The highway engineer, the computer and value for money, *Surveyor*, Vol. 142 (14 Sept. 1973), pp. 28-9.

Davies, E. (1963). *Traffic Engineering Practice*, 1963 edition, Spon.

Dockerty, A. and Brantigan, D. (1969). Computer simulation of town traffic, *Surveyor*, Vol. 133 (9 May 1969), pp. 30-4.

Drew, D. R. (1968). *Traffic Flow Theory and Control*, McGraw-Hill.

Fox, P. and Lehman, F. (1967). Digital Computer Simulation of Automobile Traffic, *Traffic Quarterly*, Vol. 21, pp. 53-66.

Gadd, S. C. L. (1971). The assessment of the accuracy of a traffic simulation program, *Surveyor* (21 May 1971), pp. 21-25 and 30.

Gerlough, D. L. (1964). Simulation of Traffic Flow, *Highway Research Special Report*, Vol. 79, pp. 97-118.

Giannopoulos, G. A. (1971). Towards a comprehensive assessment of traffic management: the use of a simulation model, *Traffic Engineering and Control*, Vol. 12, pp. 105-7.

Grigg, P. J. and Hartley, M. C. (1975). Simulation of traffic flow in a road network, *Traffic Engineering and Control*, Vol. 16, No. 2, pp. 75-77.

Ham, R. (1967). Vehicle Queue Detection using Inductive Loop Detectors, *Traffic Engineering and Control*, Vol. 8, pp. 669-71.

Hartley, M. G. and Green, D. H. (1965). Study of Intersection Problems by Simulation on a Special Purpose Computer, *Traffic Engineering and Control*, Vol. 7, pp. 219-223 and 229.

Herman, R. and Gardels, K. (1963). Vehicular Traffic Flow, *Scientific American*, Vol. 209, No. 6, pp. 35-43.

Herman, R. and Lam, T. (1972). A dilemma in overtaking on two-lane roads, *Traffic Engineering and Control*, Vol. 14, pp. 276-9 and 282.

Home Office. (1968). *Roadcraft: the Police Drivers' Manual*, HMSO.

ICL. (1969). Simulation Language SIMON, Technical Publication 4138.

Johansson, G. and Rumar, K. (1965). *Drivers' Brake-Reaction Times*, Twenty-sixth Report. Department of Psychology, University of Uppsala.

Katz, J. H. (1963). Simulation of a Traffic Network, *CACM*, Vol. 6, pp. 480-6.

Lee, J. and Jones, J. H. (1967). Traffic Dynamics: Visual Angle Car Following Models, *Traffic Engineering and Control*, Vol. 9, pp. 348-50.

Lewis, P. A. (1970). Driver behaviour equations: some limits on their solution due to vehicle dynamics, *Traffic Engineering and Control*, Vol. 12, pp. 414-6 and 420.

Lewis, R. M. and Michael, H. L. (1963). Simulation of Traffic Flow to Obtain Volume Warrants for Intersection Control, *Highway Research Board Record*, No. 15, pp. 1-43.

Mahaffy and Dodson on Road Traffic. (1973). Regulations Service, Vol. 2. Motor Vehicles (Construction and Use) Regulations 1973 (SI 1973, No. 24) Regulations 3, 11, 59. Butterworths, Shaw and Sons.

Naylor, T. H., Balintfy, J. L., Burdick, D. S. and Chu, K. (1966). *Computer Simulation Techniques*, Wiley pp. 126-36.

Needham, R. G. (1970). A simulation program of a road network controlled by traffic responsive systems, Paper 5.7 presented at Premier symposium international sur la régulation du traffic, Versailles 1-5 juin 1970. *IFAC/IFIP préprint no.* 7; pages 5b.13 to 5b.22.

Newell, G. F. (1971). *Applications of Queueing Theory*, Chapman and Hall.

Passmore, J. M. (1968). Digital computer simulation of road traffic networks, M.Sc thesis (Birmingham).

Powner, E. T. (1970). Some Aspects of the Digital Simulation of Multiple Traffic Intersections and Associated Flow Problems, Ph.D thesis (UMIST, May 1970).

Rumsey, A. F. and Hartley, M. G. (1972). Simulation of a pair of intersections, *Traffic Engineering and Control*, Vol. 13, pp. 522-5.

Salter, R. J. (1968). Capacity of Priority Intersections, *Traffic Engineering and Control*, Vol. 10, pp. 134-6 and 140.

Shumate, R. P. and Dirksen, J. R. (1965). A Simulation System for Study of Traffic Flow Behaviour, *Highway Research Record*, Vol. 72, pp. 19-39.

Stark, M. C. (1962). Computer Simulation of Traffic on Nine Blocks of a City Street, *Highway Research Board Bulletin*, Vol. 356, pp. 40-7

Story, C. E. R. (1969). A general purpose simulation of road traffic in a network by digital computer, M.Sc. thesis (Birmingham).

Story, C. E. (1970). Simulation of traffic by digital computer, *Traffic Engineering and Control*, Vol. 11, pp. 464-7.

Tanner, J. C. (1953). A problem of interference between two queues, *Biometrika*, Vol. 40, pp. 58-69.

Tanner, J. C. (1961). A derivation of the Borel distribution, *Biometrika*, Vol. 48, pp. 222-4.

Taylor, J. and Allsop, R. (1969). A new approach to traffic signal calculations, *Surveyor*, Vol. 133 (18 July 1969), pp. 38-40.

Terrell, T. J., Neville-Smith, M. and Powner, E. T. (1974). The UMIST digital traffic simulator, *Traffic Engineering and Control*, Vol. 15, pp. 876-878 and 884.

Thomasson, J. N. and Wright, P. H. (1967). Simulation of Traffic at a Two-Way Stop Intersection, *Traffic Engineering*, pp. 39-45.

THOMSON, J. M. (1967). Speeds and Flows of Traffic in Central London, Part 2. Speed-Flow Relations, *Traffic Engineering and Control*, Vol. 8, pp. 721-5.
TILLOTSON, H. T. (1971). Some aspects of traffic simulation, *Surveyor* (14 May 1971), pp. 20-21.
WEBSTER, F. V. and COBBE, B. M. (1966). *Traffic Signals*, Road Research Technical Paper No. 56, HMSO.
WOHL, M. and MARTIN, B. V. (1967). *Traffic Systems Analysis*, McGraw-Hill.
WRIGHT, C. C. and HYDE, T. (1972). The measurement and interpretation of speed-flow relationships, *Traffic Engineering and Control*, Vol. 13, pp. 507-11 and 525.

Book reviews

Digital Image Processing, by R. C. Gonzalez and P. Wintz, 1977; 431 pages. (*Addison-Wesley Advanced Book Program*, $29.50, $19.50 paper)

There are certain areas in image processing which present difficulties to non-mathematicians. In particular, the various two-dimensional transforms, such as the Fourier and Walsh transforms, and some of the more obscure digital filters may be seen as a complete mystery to those whose mathematical training stopped soon after their 'A' level studies. Professor Wintz and Dr Gonzalez have prepared an excellent text which carries the reader through such difficult areas with the minimum of pain. The expositions are thorough and eminently readable, even enjoyable. There are many worked examples and a wealth of relevant illustrations. Much of the text was originally taught to seniors and graduate students at the University of Tennessee and Purdue University; it would be equally useful for M.Sc. or Ph.D. students in this country as an introductory text and as a text book for reference.

The topics covered include fundamentals of digital images (sampling, quantisation, photography), image transforms, image enhancement and restoration, image encoding and image segmentation and description. A useful feature of the book is an evaluated list of references for further reading with every chapter. This book is highly recommended.

M. J. B. DUFF (London)

Machine Intelligence 8, edited by E. W. Elcock and Donald Michie, 1977; 630 pages. (*Ellis Horwood Limited*, £24)

The Machine Intelligence workshops started at the University of Edinburgh just over ten years ago, and they quickly became an established venue for leading workers in the field. Contributions to the published proceedings were allowed to be long, detailed and civilised enough to include excerpts from poems. The authors responded appropriately with well rounded presentations that included useful surveys and considered discussion rather than scrappy snippets of 'original results'. This eighth workshop was held at the Santa Cruz campus of the University of California; the list of participants has many names familiar from previous workshops and the contributions keep the same flavour, so that it is little changed by a 6,000 mile trip.

There are twenty-seven papers classified into nine sections, and by their nature it is impossible to give a detailed summary of individual papers. It is possible to sense a distinctive trend over the past ten years. In the early days researchers often developed tools for specific limited problems without knowing how these specific techniques might be generalised. Yet the whole motivation for their work was to produce tools to handle a wide range of problems; the only suggested general tool was theorem-proving. Theorem-proving started with techniques established in mathematics and brought the computer in later as a processing device. Now, there is a change of emphasis, the phrase 'representation of knowledge' occurs right through the contributions. It means the entry of knowledge into the computer and its storage and retrieval once there. The significant point is that it is based on a strong sense of the capabilites and characteristics of the computer itself, and this must surely be the starting point for real progress. It is not surprising that the nature of knowledge in programming is examined by several authors. Another emerging trend is a more explicit examination of the process of transferring knowledge between humans and machines; it is just beginning to be recognised that information input and output for a worthwhile problem to be solved on a machine may be a formidable task, perhaps more difficult than the solution process.

This last remark brings out one unsatisfactory feature of these contributions. It has long been recognised by designers of applications data bases in such fields as patent searching and libraries that knowledge transfer between machines and humans can be costly and difficult, yet the authors here seem to be rather unwilling to look outside the work of professional researchers in machine intelligence. There are many other places where the authors seem to be refusing to look at work done by those in specific fields of applications, yet in areas such as satellite image processing complex and subtle techniques have evolved from which these authors can surely learn much. There is only one paper where ideas have been tested in a practical environment, that by Rutovitv on chromosome analysis.

The compact summaries of various points of view presented give this volume a permanent value much above that of typical conference records, and it can be recommended for personal as well as library purchase.

J. J. FLORENTIN (London)

Pattern Recognition, edited by Bruce G. Batchelor, 1978; 485 pages. (*Plenum Press*, $47.40)

This book which is a collection of papers by a number of distinguished workers in the field has the subtitle 'Ideas in Practice'. The contributions give an interesting insight into how far the goal of producing machines which will perform the well known but little understood human facility of recognition of patterns. Very broadly, it is divided into two parts: firstly a collection of ideas, theories and techniques and secondly some down-to-earth applications.

There is a strong emphasis on optical recognition techniques and in an opening chapter, J. R. Ullman presents a detailed survey with a copious bibliography. The theoretical underpinning is presented in chapters by Batchelor and Bell. It becomes clear that pattern recognition has many practical applications, some realisable now and others in the future. These applications include the drilling of printed circuit boards, industrial sensory devices, image analysis of micromolecular structures and applications to cytology. In the acoustic field, chapters are dedicated to vehicle sound recognition and speech recognition.

The book makes fascinating reading and the editor deserves special praise for his introductory comments to each chapter which forge links between the basic ideas and practical implementations.

R. L. GRIMSDALE (Brighton)

Numerical Analysis, by R. F. Churchhouse, 1978; 69 pages. (*Christopher Davies*, £1·50)

This book is based upon a course of eighteen lectures on numerical analysis which was given to first-year students taking Computing at University College, Cardiff. The book is modest in its scope. An introductory chapter is followed by chapters on rounding and errors. Chapter four is devoted to interpolation and includes a discussion of both the methods of Lagrange and Newton. The use of finite differences to detect errors forms the basis for chapter five, whilst chapter six is concerned with numerical integration dealing with the trapezium, midpoint and Simpson rules. The longest, and last, chapter describes iterative methods, including the Newton-Raphson method, for solving nonlinear equations. It is unlikely, because of the limited range of material presented, that the book will appeal to a wider audience than that for which it is specifically prepared.

N. RILEY (Norwich)