Real-time micro-simulation of urban traffic

Iisakki Kosonen
Helsinki University of Technology, Transportation Engineering
P.O.Box 2100, FIN-02015 HUT, Finland, e-mail: Iisakki.Kosonen@Hut.Fi

Andrzej Bargiela
The Nottingham Trent University, Real-time Telemetry Systems
Butron Street, Nottingham, NG1 4BU, e-mail: andre@doc.ntu.ac.uk

KEYWORDS
Real-time micro-simulation, distributed processing, traffic measurements.

ABSTRACT

The paper presents results of the feasibility study of real-time microscopic simulation of urban traffic (using the HUTSIM micro-simulator) which has been extended to accept the real-time telemetry data provided by the urban traffic control system (SCOOT). The motivation for this study was the desire to overcome the limitation of the off-line traffic simulators for which the microscopic results do not correspond to reality. The heterogeneous computing resources that are utilised for the execution of the simulator and the urban traffic control system meant that there was a need for a suitable distributed computing environment to integrate the software components. A purpose-made Distributed Memory Environment (DIME) software, developed at the Nottingham Trent University, and the on-line version of the HUTSIM software, developed at the Helsinki University of Technology, have been deployed in this study.

After presenting the principles of interfacing the micro-simulator to the real-time traffic measurements the paper discusses the prototype implementation of the system. An important benefit of such an integrated environment is the novel extension of the use of micro-simulation as a tool for implicit measurements of the averaged acceleration rates, which offers a valuable insight into the dynamics of traffic processes.

INTRODUCTION

There is a growing need for the improvement of the efficiency of urban traffic in order to ensure the sustainability of modern cities. It is now recognized that this objective requires not only the improvement of traffic monitoring and management schemes in traffic control centres but also the provision of information services for ordinary road users. The former measure has been widely adopted by many urban traffic control centres during the last decade (deployment of traffic control systems such as SCOOT, SCATS, CARS etc.) and the latter is currently the subject of intensive research and development.

The prerequisite for the development of any traffic telematics application is the availability of real time traffic data. The system reported here makes use of the telemetry data underlying the operation of the SCOOT-urban traffic control system. The SCOOT is an adaptive system optimising the split, cycle and offset times of traffic signals. The optimisation is based on maintaining mezzoscopic models of queues and traffic flows and balancing saturation flows on all approaches to the controlled intersestions. The system provides both the unprocessed on-line detector and signal data and the various higher order measures derived through its traffic model, thus making it an ideal basis for the development of traffic information systems.

The potential for the amelioration of urban traffic through the combined supervisory control and the road-traffic information, has created the need for a flexible computing environment in which various new applications can be fully integrated with the existing traffic control systems without adversely affecting the performance of the original systems. The DIME (Distributed Memory Environment) system, developed at the Nottingham Trent University *(Argile et al 1996)*, allows various distributed traffic telematics applications to communicate with each other through the LAN or WAN networks while maintaining the shared memory logical view of data. The communication harness is based on TCP/IP-protocol and client/server architecture and it is independent of the physical network type. The performance of this system has been verified in extensive on-line tests with the SCOOT system at the Nottingham Traffic Control Centre.

HUTSIM is an object-oriented microscopic urban traffic simulation model developed at Helsinki University of Technology during the nineties *(Kosonen 1996)*. The connectivity to real control systems has been the principal aspect from the very begining of HUTSIM. The simulation model comprises a flexible and interactive object-oriented framework with a detailed rule-based vehicle dynamics, which has been calibrated based on field measurements. The first version of HUTSIM was developed in 1997. As a result of this collaborative project, HUTSIM has now been equipped with an interface to the DIME system opening, as a result, a path for the development of integrated simulation, monitoring and supervisory control applications based on real-time data.

REAL-TIME SIMULATION

The main idea behind the real-time simulation is that of the use of real-time traffic measurements as input data to microscopic simulations. In such a context, various non-measured traffic parameters, that can be deduced from micro-simulations, are deemed to be a good approximation of the reality by virtue of being based on the actual measurements. Although the prototype system presented in this paper makes use of the specific microsimulator (HUTSIM) and the urban traffic control system (SCOOT), the approach is general and is applicable to any traffic control system that supplies the required real-time data.

The basic data obtained through the traffic telemetry systems is the lane-occupancy data from detectors embedded in the road surface. Each detector provides information about the presence or absence of a vehicle in a discrete location. The rest of the information about traffic situation must be derived from the general knowledge of system layout, statistics of the traffic patterns and the estimate of vehicles' dynamics. In a simulation model all these factors are methodically combined. The simulation model provides also an engine for creating hypothetical traffic situations and for deriving higher order measures to be used by traffic information services.

In the absence of the actual traffic data from the detectors, simulation model generates vehicles on a statistical basis. Ideally this should be accurate enough to produce reliable average measures i.e. in off-line simulation mode. However, in real-time operation average measures are not always relevant, so the micro-simulation model is made realistic by replacing the time headway distribution with real-time arrivals. Since the simulation model is to mirror the operation of the actual traffic control system it requires also the real-time signal status data.

The real-time simulation approach postulated here extends significantly the monitoring capabilities of telemetry systems by extrapolating the traffic occurrences in discrete locations through to the simulation of realistic traffic flows in the whole of the network.

USING THE REAL-TIME DATA

The accuracy of real-time simulations clearly depends on the accuracy and the availability of the relevant detector data. Within the limits of the accuracy of the traffic model, the presence of discrepancies between the real and the simulated traffic can be seen as an indication of the potential for the improvement of the telemetry system itself. If the measurement data is inaccurate or incomplete the simulation will highlight this fact by demonstrating the cumulative effect of these errors.

In a real-time microscopic simulation model, individual vehicles are generated according to the lane-occupancy detector data. Vehicle arrivals are recognized from the edges of detector signal i.e. the changes of signal status from passive to active or vice versa. Usually a single vehicle corresponds to the occurrence of a set of pulses. However, some detectors cover two lanes and therefore it is possible that two vehicles travelling side-by-side produce a contiguous group of pulses which hides the presence of the second vehicle (Figure 1). This introduces an error into real-time simulations, which may be significant when traffic volume is high. Another source of discrepancies between the simulations and the reality is the absence of information about parking and minor streets traffic and the inaccuracy of the estimates of the turning movement percentages.

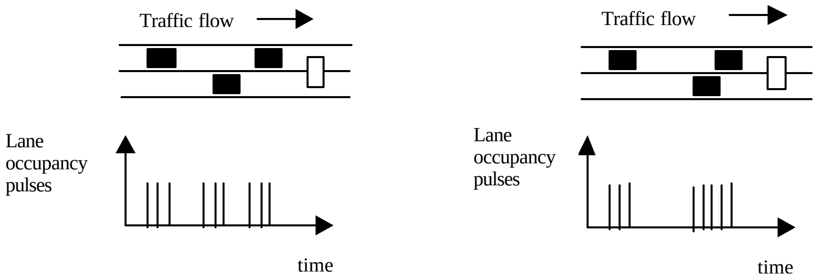

Figure 1. Real-time traffic measurements with 2-lanes inductive loops

Since, unlike in the computer simulations, the telemetry system may 'loose' some vehicles, the lane-occupancy data has to be augmented by some additional measurements that facilitate resetting of the errors in simulations that are due to inaccurate input data. In the SCOOT environment such a resetting of errors can be facilitated with special queue length detectors. When these detectors indicate continuous occupancy over several seconds this is considered to be an indication of a queue extending from the stop-line up to the detector. The SCOOT-model employs this type of detectors in resetting its own queues so that the 'back-of-queue' data reported in the SCOOT messages can be used for cross-referencing purposes by the real-time simulator.

Accepting that the simulated traffic will always differ somewhat from reality and, in particular, because even comparatively small errors can accumulate over long periods of simulation, HUTSIM model facilitates corrective generation and/or removal of vehicles in the links. This type of procedure has been shown to be effective in correcting discrepancies, on a continuous basis, thus preventing the build-up of errors (Figure 2).

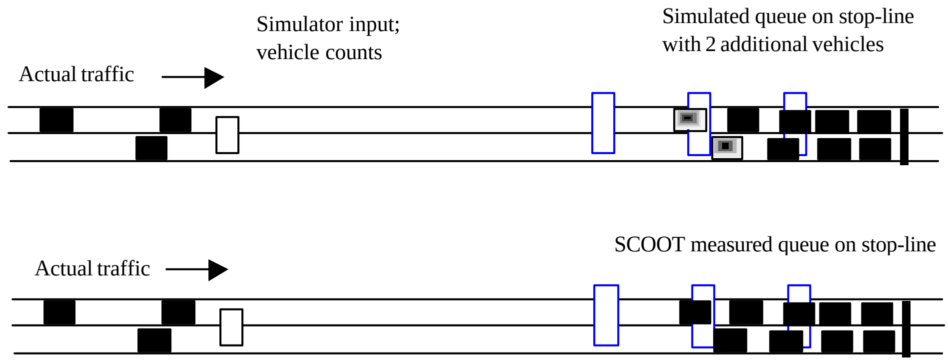

Figure 2. Correction of the systematic error in real-time simulations

Within the context of real-time simulations, detector data (time-stamped vehicle counts) can be used not only to generate the instances of vehicles in specific locations but also to improve the accuracy of the various traffic estimates. For instance if lane-occupancy detectors are deployed at each exit from an intersection, turning movements can be calculated on the 'moving average' basis and be compared with the values used by the simulator. If the corresponding queues in the 'downstream' links indicate a systematic error that is positive in some links and negative in others, then the turning movement coefficients are adjusted in the model. On the other hand, if the errors are all positive or negative in the 'downstream' links, then the discharge flow rates can be adjusted accordingly. When there are lane occupancy detectors positioned immediately after the stop line, their readings could be used for on-line adjustment of the discharge flow rate. Furthermore, if separate detector data is available for individual lanes, this data could be used for tuning the lane change parameters and, when lanes determine the direction of turning, it could be used to adapt turning movements on the basis of the incoming link data.

In a specific case when the lane occupancy data is available together with traffic lights data it becomes possible to infer the dynamics of the vehicles. By comparing time-delays between the stage change and the subsequent groups of pulses recorded by the 'downstream' inductive loops in the actual SCOOT readings and in the HUTSIM simulations (time instances T1, T2, etc. – Figure 3), we derive adjustments to the acceleration parameter in the micro-simulation. In order to instill a degree of robustness into the calculations we process only these lane occupancy readings that fall within the time-window defined by the maximum and minimum acceleration rates. Any reading outside this time-window is discarded as spurious data possibly representing the vehicles that have been parked and have re-joined the traffic flow or the exceptionally slow vehicles that have entered the link during the previous signaling stage. In either case discarding the a-typical data is consistent with our objective of identifying temporal variation of the average drivers' behaviour characteristics.

TECHNICAL FRAMEWORK

A prototype of the real-time simulation system has been constructed using the distributed shared memory environment (DIME) developed at NTU. This enables several pieces of software to execute on networked computers while cooperating in performing the simulation task (Figure 4).

Each process connects to the shared memory manager (SMM), which runs in either UNIX or

Windows-PC environment. The memory manager acts as a server to which multiple clients can connect. Each client can have a read and/or write access to several areas (buffers) of the memory manager. A client can also create a memory area with an exclusive write access so that other clients connect to that area are allowed a read access only. The specification of shared memory data structures and the corresponding access privileges is accomplished through a user-friendly API. Two types of areas are supported in DIME namely buffers for passing messages and arrays for sharing static data.

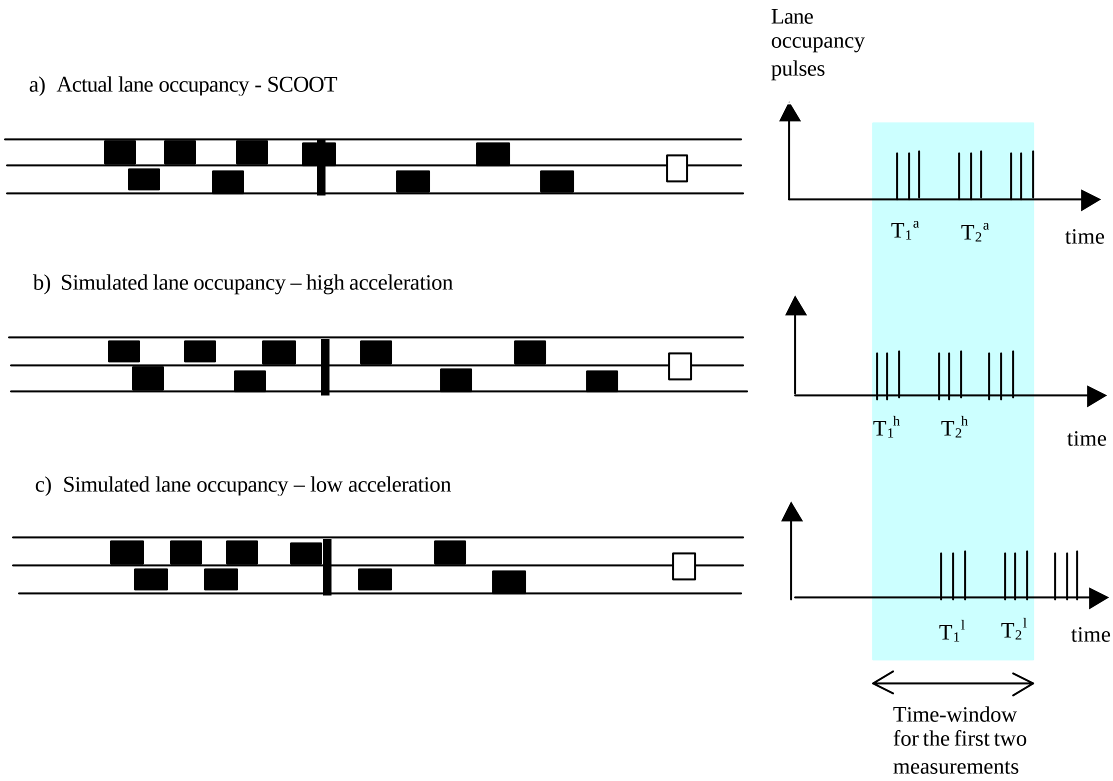

Figure 3. Actual and simulated lane occupancy with their corresponding measurements

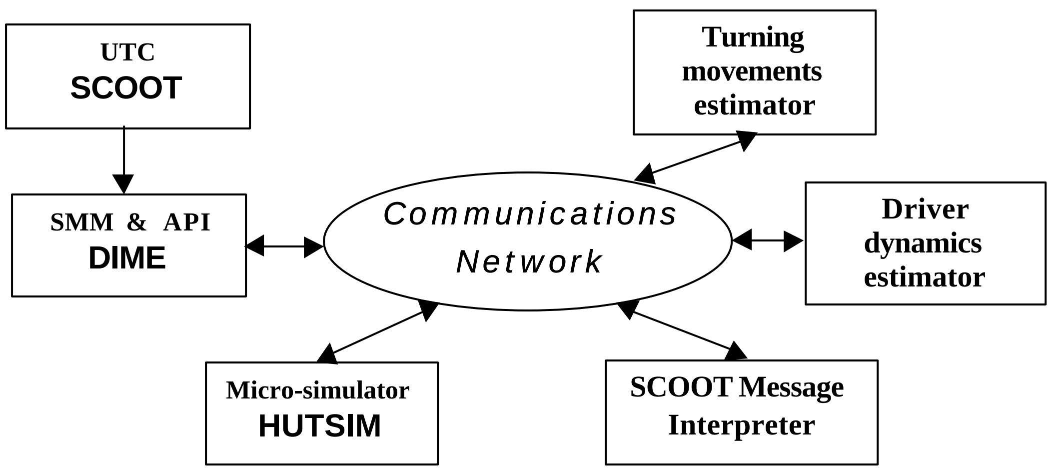

Figure 4. DIME implementation framework for distributed telematics applications.

One of the clients connected to the memory manager is the SCOOT-system which provides real-time telemetry data. The real-time simulation application makes use of two types of messages from the SCOOT system: M19 and M14. The M19-messages supply the detector status data. These messages are generated once per second for each detector and they contain the last four states of the lane-occupancy detector with 0.25 second resolution. The M14 messages are generated every four seconds and supply the link flow and the queue length data together with the signal head status over the last four seconds. Although only the 'vehicle occurence' and the signal head status are mandatory for the simulations, the queue length data from the M14-message is also used for resetting of the cumulative error.

The M19 and M14 buffers of the DIME memory manager are read by the Message Interpreter client which converts them into a suitable format. The interpreter turns the SCOOT detector data (M19) into HUTSIM vehicle arrival messages by identifying changes in the detector status. Changes in the signal heads status are identified in the M14 data and are converted into HUTSIM signal change messages. Because only the changes of the status are recorded the amount of output data is much less than the amount of input data. The output messages are written into another DIME buffer readable by other clients like HUTSIM. The message interpreter successfully isolates the simulation task from the SCOOT specific interface issues.

The turning movement estimation client was originally developed for the predictive macroscopic simulation model (PADSIM) but it could also be used here in the context of real-time micro-simulation to provide adaptively updated turning movement coefficients. The estimation results are stored in a static memory array of DIME since the coefficients are updated only every 20 minutes. The message interpreter reads the array and generates HUTSIM-messages that update the turning percentages of traffic generators of boundary links and route generators of internal links. These messages are stored in the same DIME buffer as the vehicle arrival and signal messages. In the current version however, the turning movement coefficients are kept constant throughout the simulation.

The micro-simulation client, HUTSIM, is operating in an autonomous fashion running with real-time speed and processing external events from the input stream. While in the off-line mode the input stream is provided from an input file, in the case of real-time simulation the input stream is supplied by the message interpreter through DIME. The present implementation involves four types of messages namely the vehicle arrival-, signal change- , turning percentages and acceleration update messages. The output stream, which contains time sequences of the simulated lane occupancy readings, is also channeled to DIME.

The Driver Dynamics Estimator client processes both the simulated (from HUTSIM) and the actual (from SCOOT) lane occupancy readings and performs Kalman filtering of the discrepancies to arrive at the estimate of the acceleration parameter. The estimate is forwarded to DIME for use by HUTSIM in subsequent simulations.

The communication link involves data transfer from the Mansfield test area to the Nottingham Traffic Control Centre and from there to the memory manager run at the Nottingham Trent University. Although the DIME access is very rapid, the local- and the wide area networks are subject to random fluctuations of the communications load, the maximum delay of the whole chain can be several seconds and messages can appear in bursts. Therefore, in the interest of realism the message processing has been based on time stamping. By offsetting the simulation, by a time representing the maximum communications delay, the random delays in communications are prevented from affecting the simulation. In the test system, a 15 seconds offset has been found sufficient under most

SIMULATION OF MANSFIELD TEST AREA

The HUTSIM / DIME real-time simulation system described here has been implemented at the Nottingham Trent University and is connected to the SCOOT-system of the Nottingham Traffic Control Centre. The system typically runs as a fully distributed application with each of the four cooperating tasks executing on a separate computer. However the DIME framework does not place any constraints on the hardware configuration and e.g. HUTSIM and the message interpreter can run on the same PC while the memory manager and the turning movements estimator can execute either on a UNIX workstation or a PC. The HUTSIM / DIME simulation is running successfully and the framework has proven to be reliable.

The performance of the system has been validated through real-time simulations of the Mansfield-south SCOOT area. The test area covered six intersections for which the rush hour traffic tends to be very busy. A detailed HUTSIM-model was constructed including the geometry, lane organisation, detector positions etc. (Figure 5.). The boundary link detectors were replaced with vehicle generators and the internal links were equipped with route generators in the HUTSIM model. All signals and generators were labeled according to the link and detector numbers of SCOOT so that that all messages could be delivered to the correct object.

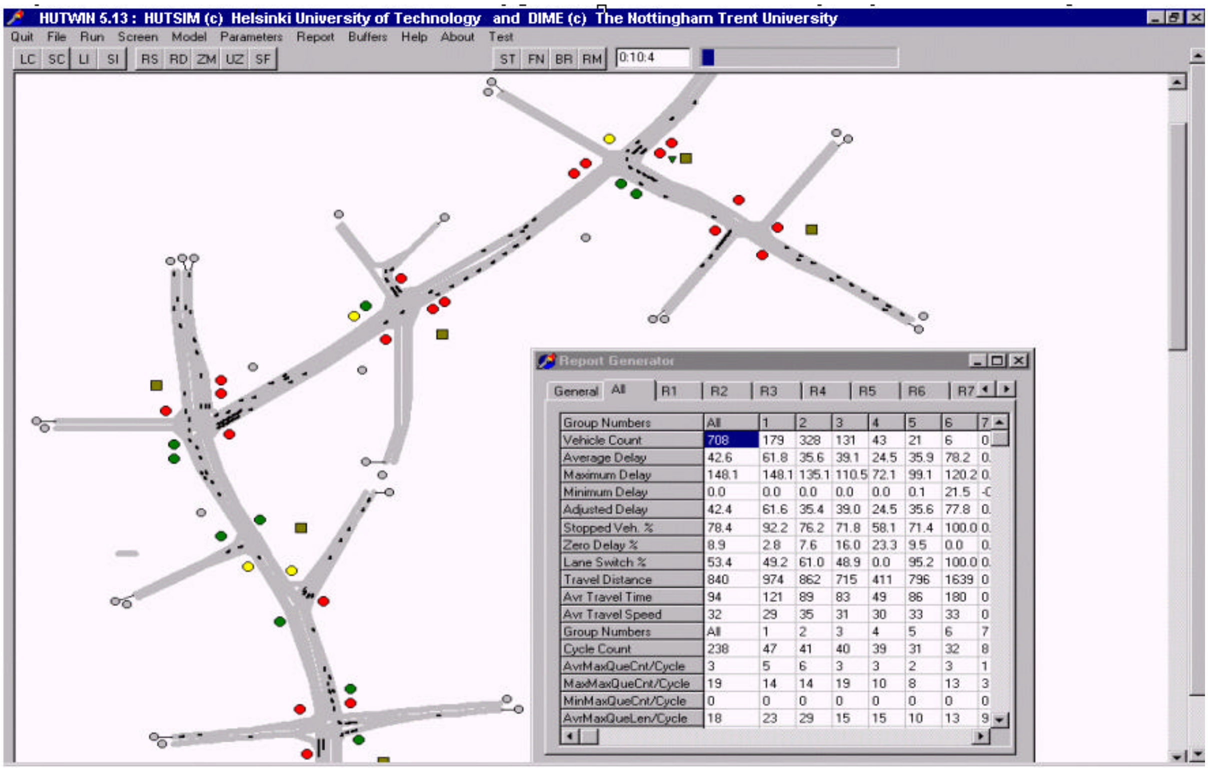

Figure 5. Real-time simulation of Mansfield test area

In order to compare the simulated and the real traffic, a ‘bird’s eye’ video recording of the entire SCOOT area would be most useful. However, such an recording was beyond the means of this project. An alternative video recording of individual approaches to all simulated intersections was also considered uneconomical and exceedingly tedious when attempting to time-synchronise 24 video recordings and the real-time simulations.

As a practical approach we have used therefore the queue measure from the M14-SCOOT message and have compared it with the queue simulated by HUTSIM. The SCOOT queue is expressed in LPU:s (Link Profile Units) and has been found to provide a good estimate of vehicle count. The results of the field validation of the SCOOT queue as a measure of the number of vehicles on the stop-line has been performed for a number of intersections and a representative set of results is given in Figure 6. It can be seen that the majority of queue readings give an unambiguous count and the statistical variability of readings is a realistic reflection of the variability of vehicle types.

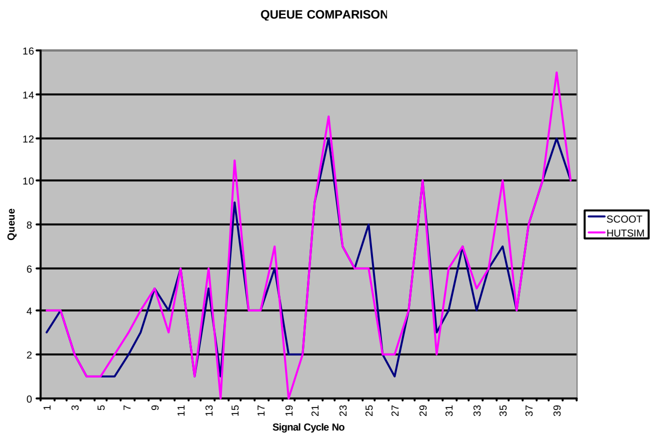

Figure 6. Comparison of the on-line simulated queue with SCOOT data

Preliminary simulations show that the matching of vehicle counts on boundary links was good. In internal links the matching between simulation and field data can be improved by tuning the model parameters. Especially the discharge flow parameters may need to be adjusted properly for each link. The results show that there is also a need for refinement of the calculation of dynamic turning movement coefficients which are adversly affected by the missing data in some of the links. Consequently, static turning movement percentages need to be used in some cases.

Because the number of settings and parameters can be quite large, the manual tuning of the simulator can easily becomes tedious. The preliminary experiences indicate that there is a need for adaptive tuning of the parameters. This is one of the areas to be addressed in future research.

The computational performance of the simulation system executing on a 200MHz Pentium PC has been found to be sufficient for this size of model. For larger models the performance issues may need to be reconsidered. The critical factors are the size of the simulation model and the maximum rate of data communication.

ESTIMATION OF DRIVERS' DYNAMICS

Within the computational framework described in the previous sections the task of estimation of the average value of the acceleration parameter in n discrete locations can be formalised as follows. We describe the drivers' dynamics **D**, as n-dimensional discrete stochastic process $\{\mathbf{x}(k),\ k=0,1,\ldots\}$. The system state **x** represents the discrepancy between the actual and the assumed (in the micro-simulation) values of the acceleration parameters in each of the n locations. The state **x** can only be measured indirectly through some metering system **M** that provides measurements $\{\mathbf{z}(k),\ i=1,2,\ldots\}$ which are m*n-dimensional discrete stochastic processes (Figure 7).

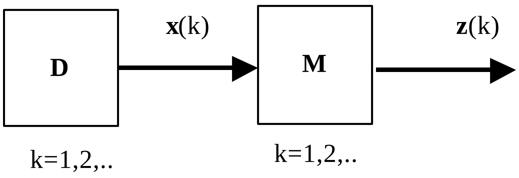

Figure 7. The representation of the discrete dynamic system with state **x** and measurements **z**

The metering system **M** is constructed by combining the real-time simulation with the telemetry data acquisition to provide virtual measurements of discrepancies between the actual and the simulated arrival times of vehicles over the monitored lane occupancy detectors

$$\mathbf{z}(k)=[z_1(k),\ z_2(k),\ ..z_m(k)]^T \text{ with } z_i(k)=T_i^a-T_i^s : \quad i=1,2,..,m*n$$

where, T_i^a is the actual arrival time, T_i^s is the corresponding simulated arrival time (with current acceleration parameters) and i is the measurement index.

The model of the dynamic system considered here is given by the following equations

$$\mathbf{x}(k+1) = \mathbf{F}(k+1,k)\mathbf{x}(k) + \mathbf{G}(k+1,k)\mathbf{w}(k) \qquad (1)$$

$$\mathbf{z}(k+1) = \mathbf{H}(k+1)\mathbf{x}(k+1) + \mathbf{v}(k+1) \qquad (2)$$

where **x** is n-dimensional state vector, **w** is p-dimensional disturbance vector, **z** is m*n-dimensional measurement vector, **v** is m*n-dimensional measurement noise vector and k is the discrete time index. State transition matrix **F**, system noise matrix **G** and measurement noise matrix **H** have corresponding dimensions of n x n, n x p and (m*n) x n.

The processes $\{\mathbf{w}(k),\ k=0,1,\ldots\}$ and $\{\mathbf{v}(k+1), k=0,1,\ldots\}$ are discrete white noise for which

$$E[\mathbf{w}(k)]=0,$$
$$E[\mathbf{v}(k+1)]=0 \qquad \text{for } k=0, 1, 2, \ldots$$

$$E[\mathbf{w}(j)\mathbf{w}^T(k)] = \mathbf{Q}(k)$$
$$E[\mathbf{v}(j+1)\mathbf{v}^T(k+1)] = \mathbf{R}(k+1) \qquad \text{for } j, k = 0, 1, 2, \ldots$$

$$E[\mathbf{v}(j)\mathbf{w}^T(k)] = 0$$

With the above definitions it is possible to derive the following equations for optimal estimation **x** of the system state **x** on the basis of the available measurements **z**. If the time index of the system state and the measurement vector are identical this is referred to as optimal filtration.

$$\mathbf{x}(k+1|k+1) = \mathbf{F}(k+1,k)\ \mathbf{x}(k|k) + \mathbf{K}(k+1)[\mathbf{z}(k+1)-\mathbf{H}(k+1)\mathbf{F}(k+1,k)\mathbf{x}(k|k)] \qquad (3)$$
$$k=0, 1, \ldots, \text{ with } \mathbf{x}(0|0)=\mathbf{0}$$

$$\mathbf{P}(k+1|k)=\mathbf{F}(k+1,k)\mathbf{P}(k|k)\ \mathbf{F}^T(k+1,k)+\mathbf{G}(k+1,k)\mathbf{Q}(k)\mathbf{G}^T(k+1,k) \qquad (4)$$

$$\mathbf{K}(k+1)=\mathbf{P}(k+1|k)\mathbf{H}^T(k+1)[\mathbf{H}(k+1)\mathbf{P}(k+1|k)\mathbf{H}^T(k+1) + \mathbf{R}(k+1)]^{-1} \qquad (5)$$

$$\mathbf{P}(k+1|k+1) = [\mathbf{I} - \mathbf{K}(k+1)\mathbf{H}(k+1)]\mathbf{P}(k+1|k) \qquad (6)$$

k=0, 1, ..., with $\mathbf{P}(0|0) = E[\mathbf{x}(0)\mathbf{x}^T(0)]$ is the initial condition of equation (4)

The difference between the actual and the estimated system state

$$\mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1|\, k+1), \qquad \text{for } k=0, 1, \ldots$$

is a Gauss-Markow process with a zero mean and a covariance matrix defined by equation (6). The solution of the equations (3)-(6) is computationally quite straightforward except for the inverse of the matrix in the equation (5). Even for a moderately small dimensionality of the system state (e.g. n=20) and the number of independent measurements taken for each state variable (e.g. m=5), the size of the matrix to be inverted in equation (5) is m*n=100, which precludes real-time computation on today's 'top-of-the-range' PCs. We have therefore adopted an alternative formulation of the problem as a collection of n independent scalar state estimation problems. Indeed, if the results prove that the estimates of the acceleration rate in various locations are strongly correlated it might be possible to simplify the task even further and to consider a single scalar state estimation.

CONCLUSION

The HUTSIM / DIME system has been implemented and is running successfully on a network of distributed computing nodes. The system demonstrates the principle of real-time simulation and it's benefits. Further research is needed to study in depth the achievable accuracy of real-time simulations. This will involve full scale comparison of field data with simulated measurements. The research is also likely to result in improvements in the telemetry system by suggesting the optimal number, type and location of detectors in support of real-time simulations. The motivation for further research is twofold: to provide a reference data for the various information services and to develop a reliable predictive model of urban traffic in support of future traffic control strategies.

Increasingly, the combination of simulations and the real traffic data is providing a viable route to obtaining greater insights into the dynamics of traffic processes with the consequent possibility of improved operation of traffic and transportation systems. By formulating the task of estimation of drivers' dynamics as an optimal filtration problem we have highlighted the additional relevance of real-time micro-simulations as a tool for obtaining greater insights into traffic system operations.

REFERENCES

Argile A, Peytchev E, Bargiela A, Kosonen I (1996). *DIME: A shared memory environment for distributed simulation*, monitoring and control of urban areas, Proc. ESS'96, Genoa, 1996, ISBN 1-56555-099-4, Vol.1, pp.152-156.

Bargiela A, Berry R (1999), *Enhancing the benefits of UTC through distributed applications*, Traffic Technology International, February 1999, pp.63-66.

Kosonen I, Bargiela A, Claramunt C (1999), *A distributed traffic monitoring and information system*, Journal og GIC and Data Bases, Vol.3, No.1, June 1999.

Kosonen I (1996). *HUTSIM - A Simulation Tool for Traffic Signal Control Planning. Helsinki University of Technology*, Laboratory of Transportation engineering. Licentiate Thesis. 150 p.

Medich J.S. (1969), Stochastic Optimal Linear Estimation and Control, McGraw Hill

Peytchev E, Bargiela A, Gessing R (1996). *A predictive macroscopic city traffic simulation model*, Proc. ESS'96, Genoa, 1996, ISBN 1-56555-099-4, Vol.2, pp.38-42.

Peytchev E, Bargiela A, (1998), *Traffic telematics software environment*, Proc. 10th European Simulation Symposium ESS'98, ISBN 1-56555-147-8, pp.378-382.