pressions for Average Delay. Transportation Science, Vol. 12, 1978, pp. 29-47.
6. F.V. Webster. Traffic Signal Settings. Road Research Technical Paper 39. Her Majesty's Stationary Office, London, England, 1958.
7. Road Research Laboratory. Research on Road Traffic. Her Majesty's Stationary Office, London, England, 1965.
8. D.I. Robertson and P. Gower. User Guide to TRANSYT Version 6. Supplementary Report 255. Transport and Road Research Laboratory, Crowthorne, Berkshire, England, 1977.
9. D.I. Robertson. Traffic Models and Optimum Strategies of Control--A Review. Proc., International Symposium on Traffic Control Systems, edited by W.S. Homburger and L. Steinman, Vol. 1, Berkeley, Calif., 1979, pp. 262-288.
10. R.M. Kimber and E.M. Hollis. Traffic Queues and Delays at Road Junctions. Laboratory Report 909. Transport and Road Research Laboratory, Crowthorne, Berkshire, England, 1979.
11. R. Akcelik. Traffic Signals: Capacity and Timing Analysis. Research Report 123. Australian Road Research Board, Victoria, 1981.
12. R. Akcelik. Time-Dependent Expressions for Delay, Stop Rate and Queue Length at Traffic Signals. Internal Report AIR 367-1. Australian Road Research Board, Numawading, Victoria, 1980.
13. W.R. Reilly, C.C. Gardner, and J.H. Kell. A Technique for Measurement of Delay at Intersections. Report FHWA-RD-76-137. FHWA, U.S. Department of Transportation, 1976.

Publication of this paper sponsored by Committee on Traffic Flow Theory and Characteristics.

Automated Collection of Vehicular Delay Data at Intersections

JAY F. LEGERE and A. ESSAM RADWAN

ABSTRACT

Most current methods used to estimate vehicular delay at intersections involve some form of manual data collection. These methods rely on statistical techniques (such as multiple linear regression) to improve the accuracy of the delay estimates. In addition, most require significant data collection and reduction efforts. The theory, design, operation, and evaluation of a microprocessor-based system for the collection of vehicular delay data at intersections is presented. The principle of the automated system, including definitions of pertinent variables and equations, is discussed. An overview of the system design, including a description of the vehicle detection scheme and the microprocessor's recognition of vehicle arrivals and departures, is presented. There is a discussion of the system software as well as a description of the data collection and reduction processes. The system performance was evaluated both in the laboratory and through analysis of data collected in the field. Recommendations for further development of the device are presented.

Since the Arab oil embargo of 1973, the United States has become increasingly concerned with its energy supply and with ways in which that supply can be conserved. One area of particular interest is the conservation of energy within the transportation sector. About 40 percent of this nation's petroleum consumption is attributable to passenger travel by automobile.

Much traffic engineering research has been done on delay and fuel consumption at signalized intersections simply because they are considered the locations where most delay and excess fuel consumption occur. Unfortunately, the most accurate methods of data collection and analysis have proven to be extremely time consuming and costly.

Estimates of intersection delay are used in numerous applications, some of which are validation and calibration of computer simulation models, estimates of road-user costs, before-and-after studies, comparisons of the efficiency of various types of intersection control, and comparison of specific signal timing and phasing.

The theory, design, implementation, and evaluation of a microprocessor-based system for the collection of vehicular delay data at intersections are presented here. The primary application of the system is for the collection of data at intersections that are under some form of signalized control. The system is also applicable to any intersection or, in general, to any section of highway for which values of average travel time and delay are desired. Details of the hardware specifications, the software routines, and the assembly language program for this application are fully documented elsewhere (1).

PRINCIPLE OF AUTOMATED DATA COLLECTION METHOD

A report prepared for the FHWA by JHK & Associates (2) presents the results of a research project on defining and measuring delays at intersections. Four basic methodologies were identified for use in estimating delay: point sample, input-output, path trace, and modeling. The input-output method has proven to be convenient and reliable when used to measure approach delay. However, the data reduction process is extremely tedious and time consuming. If the input-output method could be automated by the use of a microprocessor-based system, the problem of data reduction would be eliminated and the accuracy of the data collected would be improved.

The input-output method requires

1. Definition of an approach delay section. The downstream end or exit point of the section is located just beyond the stop line. The entrance to the section is located at an arbitrary point upstream such that the length of the section includes all delay associated with the signal.
2. Determination of a sample interval for the data collection process. The boundaries of the approach delay section were defined by two sets of detectors and a sample interval of 1 sec was chosen to increase the accuracy of the data collected.

Figure 1 shows a simplified diagram of the system setup. Each set of detectors is controlled by its own microprocessor. For each sample interval in the data collection period, each microprocessor counts and records the number of wheels that pass over its corresponding set of detectors. The method of wheel counting is a new concept that resulted as a by-product of the detection scheme chosen. It was possible that a new source of error would be introduced with this method. However, it was believed that the accuracy gained by decreasing the sample interval would result in a net improvement in the accuracy of the data collected.

The number of wheels crossing the upstream detectors for any sample interval i is denoted by NWI_i (number of wheels in for sample interval i). Likewise, the number of wheels crossing the downstream detectors is denoted by NWO_i (number of wheels out for sample interval i). The number of wheels on the approach delay section for any sample interval i, $NWADS_i$, is given by

$$NWADS_i = NWADS_{i-1} + NWI_i - NWO_i$$

where

$NWADS_i$ = number of wheels on the approach delay section for the previous sample interval.

For the first sample interval (i = 1) the number of wheels on the approach delay section for the previous interval must be determined by a manual count.

Total travel time for all wheels traversing the approach delay section in a 15-min data collection period is given by

$$TTT = \sum_{i=1}^{n} (NWADS_i)(\Delta t)$$

where

$NWADS_i$ = number of wheels on the approach delay section for sample interval i,
Δt = length of the sample interval (seconds), and
n = number of sample intervals in the data collection period.

Dividing the total travel time (in wheel-seconds as a result of the multiplication) by the number of wheels entering the system yields the average travel time per wheel (in seconds).

To estimate average approach delay, an estimate of the approach free flow travel time is required. This estimate may be obtained by two methods: (a) averaging a sample of travel times for unimpeded vehicles over the approach delay section or, (b) for types of control other than traffic signals, calculating a free flow travel time based on upstream or downstream free flow travel speeds. Subtracting the estimated free flow travel time from the average travel time yields an estimate of the average approach delay.

Also shown in Figure 1 is a communications line connecting the two processors. This line is used to pass control information from the "master" processor

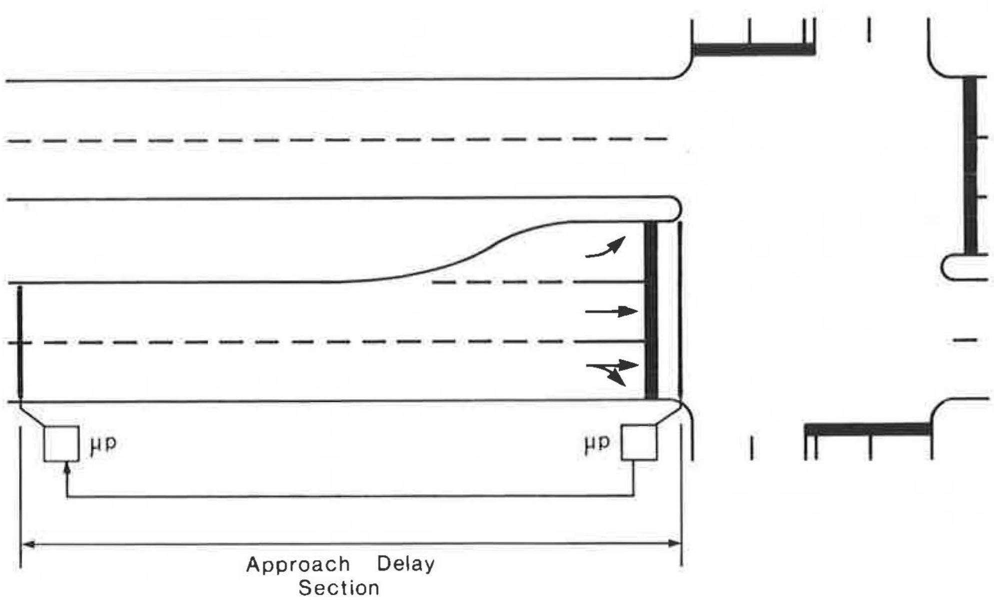

FIGURE 1 Microprocessor system setup.

at the stop line to the "slave" processor. The control information is used to synchronize the two processors at the beginning of a data collection period.

MICROPROCESSOR SYSTEM DESIGN OVERVIEW

Vehicle Detection Scheme

Two forms of vehicle detection were considered for this application. The most common form, the loop detector, was examined as well as a more temporary form of detection, the electrical tapeswitch. Both forms of detection were analyzed in several configurations to determine which would most nearly provide the accuracy required by this application.

Loop Detectors

The principle of loop detection (disturbance of a magnetic field by a heavy metal object) makes it difficult to accurately define the detection area of the loop. For this reason a standard detector configuration presents two possible sources of error that are shown in Figure 2.

In Figure 2(a), vehicles A and B pass over the loop detector with a headway sufficiently small to cause a continuous disturbance within the loop. This situation would probably not occur at the upstream detectors. However, at the stop line, where speeds can be low due to departure from a queue or execution of a turning movement, this case could occur frequently.

In Figure 2(b), a vehicle performs a lane change between two loop detectors. The question here is whether the vehicle generates a single count, two counts (one on each detector), or no count at all. This case may not occur at the stop line, but it might occur at the upstream detectors. In addition to the inaccuracy considerations, the cost of loop detection made it an extremely unattractive alternative for this application.

Electrical Tapeswitches

Electrical tapeswitches were preferred to pneumatic pressure tubes because the interface between the microprocessor and the tapeswitches could be more easily developed.

The first tapeswitch configuration scheme analyzed is shown in Figure 3. The figure shows two tapeswitches, one switch covering both lanes of traffic and the other covering only one lane. The possible sources of error for this configuration include (a) axle counting as opposed to vehicle counting (multiaxled vehicles); (b) for angled vehicles, four counts (one for each wheel) instead of two counts; and (c) coincident closure of a switch by two adjacent vehicles.

To eliminate the first two sources of error, an attempt was made to find an angle at which a tapeswitch could be mounted to ensure a count for each wheel. This second configuration is shown in Figure 4. However, the problem of coincident closure would remain if adjacent vehicles were slightly staggered as shown in the figure.

It became obvious that, to alleviate that third source of error, shorter tapeswitches would be necessary. A configuration using 2-ft tapeswitches placed end-to-end was investigated. This configuration is shown in Figure 5. With the shorter switches, each closure would consistently represent a vehicle's wheel rather than its axle. Because of this, the counting of angled vehicles would present no problem. Also, it would be impossible for two adjacent vehicles to actuate the same switch. After considering all possible sizes, speeds, and combinations of vehicles, it was decided that the most cost-effective and accurate detection scheme would be that shown in Figure 5.

Switch Scanning and Closure Detection

After the switch configuration was chosen, it was necessary to determine how the switches were to be

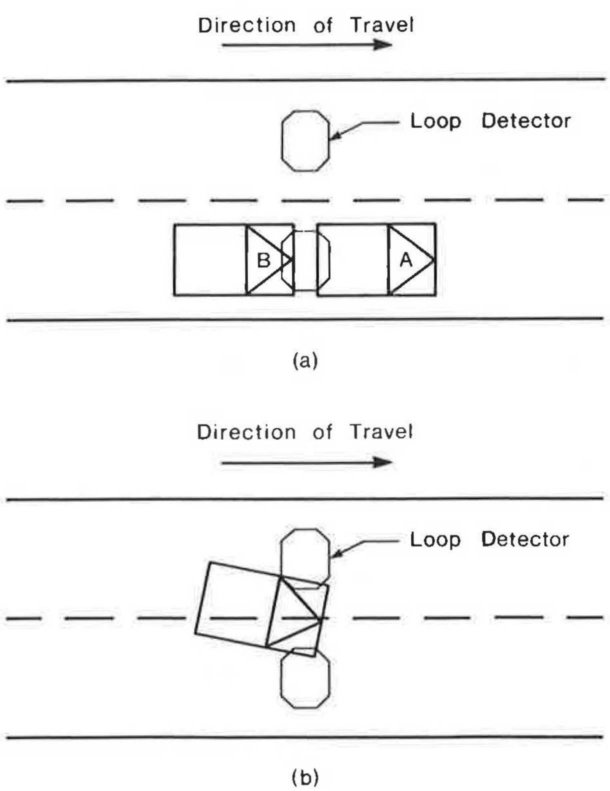

FIGURE 2 Sources of error in loop detection.

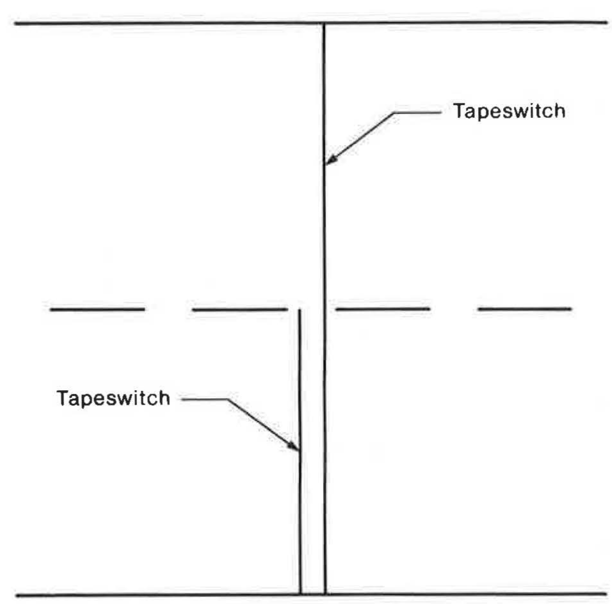

FIGURE 3 First electrical tapeswitch configuration.

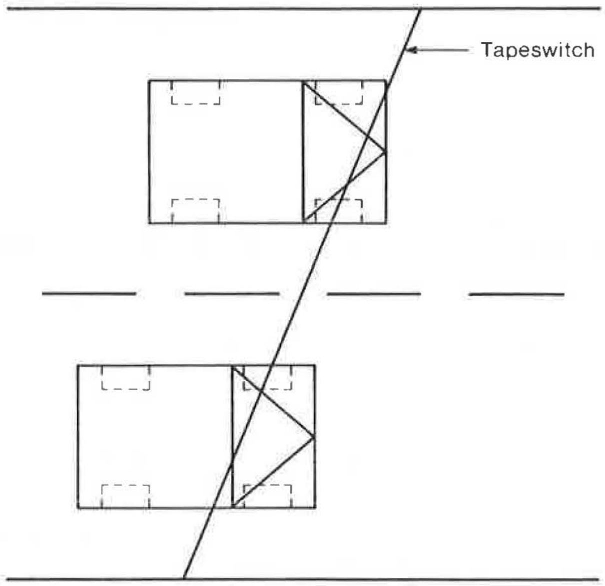

FIGURE 4 Second electrical tapeswitch configuration.

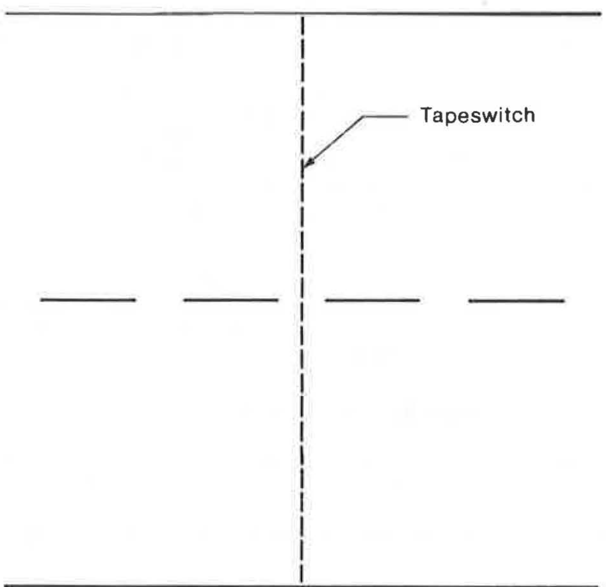

FIGURE 5 Series of 2-ft tapeswitches.

scanned by the microprocessor so as not to miss the passage of any vehicle wheels. A tapeswitch interface circuit was designed to provide a logic 0 level to the microprocessor when the switch was open and a logic 1 level when closed. With this interface, a typical logic-level diagram for the passage of a vehicle's wheel over a tapeswitch would be as shown in Figure 6(a), and a vehicle passage (two wheels) would be as shown in Figure 6(b). The time lapse depicted in the figure may be only a fraction of a second. Because of this, the microprocessor must scan the switches quickly enough to detect the switch closures caused by both the front and rear wheels.

Switch-Status Flags

Each tapeswitch has two switch-status flags assigned in the microprocessor's main memory. The first flag is set when the switch is hit (logic 0 to 1 transition shown in Figure 6) by a vehicle's wheel and the second flag is set when the switch has been cleared (logic 1 to 0 transition) by a wheel passing completely over it. The logical process required to set these flags is as follows: (a) Determine if the switch has previously been hit. (b) If the switch has not been hit, determine the current status of the switch (open or closed). If the switch is still open, do nothing. If the switch is now closed, set the switch hit flag. (c) If the switch has been hit, determine the current status of the switch. If the switch is still closed, do nothing. If the switch is now open, set the switch cleared flag. (d) Repeat the process for each switch.

The switch-status flags are used by another software routine that logs or counts the passage of each vehicle's wheels and resets the flags when the wheel has passed completely over the switches.

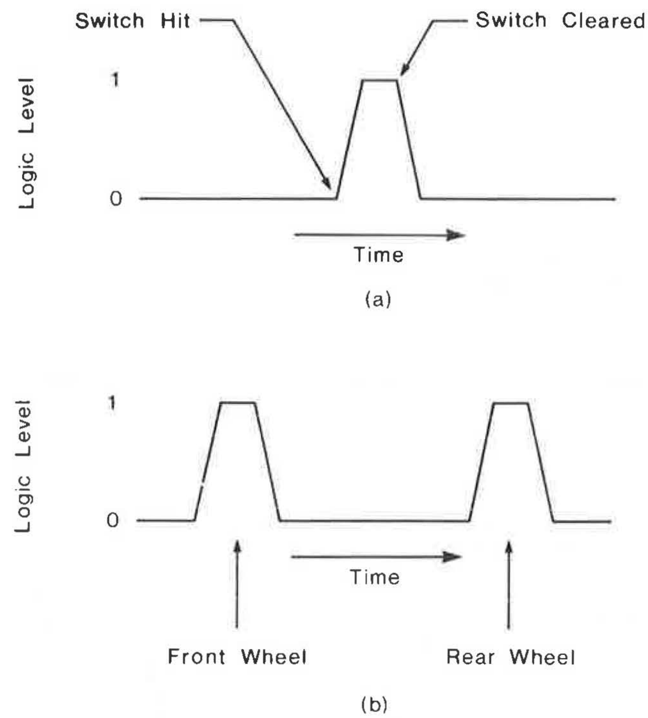

FIGURE 6 Tapeswitch logic-level diagrams.

Flag-Check Interval

The switch-scanning routine described in the previous section is executed repeatedly by the microprocessor. However, it was also necessary to check the switch-status flags at regular intervals to determine the presence of any vehicle wheels to be counted. A flag-check interval was chosen based on calculations of front and rear wheel passage time. Assuming a minimum of 5 ft between a vehicle's front and rear wheels, maximum vehicle passage speeds were calculated for various flag-check intervals. Some of the intervals and their corresponding maximum speeds are

Flag-Check Interval (msec)	Maximum Speed (mph)
10	341
20	170
25	136
30	114
40	85
50	68

The 25-msec flag-check interval was selected because it provided a safe (not likely to be exceeded) maximum speed and because 1 sec is evenly divisible by this interval. A flag-check interval of 25 msec implies that the tapeswitch status flags are examined 40 times per second. Each time the flags are examined, the wheel count is incremented if it is determined that a wheel has passed over the tapeswitch. This wheel count is stored in memory and reset to zero every second.

Switch Patterns and Wheel Counting

After the flag-check interval was selected, it was necessary to develop a method for recognizing the possible switch closure patterns and a method for counting vehicle wheels. It was recognized that, with the detection scheme used, it was likely that a vehicle's wheels would not always pass directly over the middle of a switch. In fact, four possible switch closure patterns were identified. These patterns are shown in Figure 7 and are (a) A wheel passes directly over the middle of a switch. (b) A wheel passes directly between two adjacent switches. (c) A vehicle with a wheelbase of less than 4 ft has its left wheel pass directly between two adjacent switches and its right wheel over a third switch. (d) A vehicle with a wheelbase approximately equal to 4 ft has its left wheel pass directly between two adjacent switches and its right wheel pass directly between another pair of adjacent switches.

These patterns result in the closure of one, two, three, or four switches in a row and each possible pattern was recognized by a software routine and handled as follows: Scan the array of switches sequentially from one side of the roadway to the other, and, when a switch pattern is recognized, check the pattern to be sure that each switch in the pattern has been cleared. If not, ignore the pattern until all switches have been cleared. If all switches in the pattern have been cleared, increment the wheel count by 1 for one or two switches in a row, or by 2 for three or four switches in a row, then reset the appropriate switch-status flags.

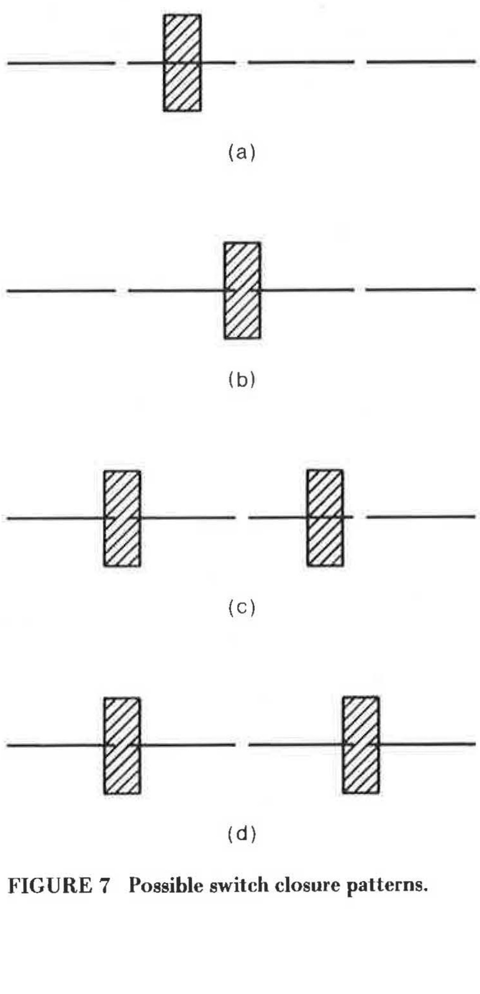

FIGURE 7 Possible switch closure patterns.

SYSTEM HARDWARE AND SOFTWARE

The automated data collection system is based on the Intel 8085 microprocessor. The foundation of the system is Intel's SDK-85 system design kit. This kit provides all the necessary components to build a complete 8085-based microcomputer system (3,4).

The electrical tapeswitches used with the data collection system were connected to the microprocessor input ports through the tapeswitch interface circuitry. Each tapeswitch is made up of two metal contacts separated by thin plastic spacers. At one end of the switch, two lead-in wires are spot soldered, one to each metal contact. To protect the sensitive ends of the tapeswitches from the impact of vehicle wheel passages, small pieces of U-shaped steel channel were used to bridge the wheels over switches. Each tapeswitch is actuated by applying approximately 20 pounds pressure at any point along its length. The switches are catalog number 170-IS Temporary Roadway Instrumentation Switches manufactured by the Tapeswitch Corporation of America (5).

The software that controls the data collection system performs four basic functions:

1. Initialization;
2. System synchronization and start-up;
3. Switch scanning and status-flag update; and
4. Status-flag check, wheel counting, and data storage.

The initialization process is essentially the same for the two microprocessors. The direction of data flow for all input-output (I/O) ports is defined and all memory locations, flags, and pointers are set up.

The master processor and slave processor operate independently during initialization and data collection. However, synchronization of the two processors is required before data collection begins to ensure that both operate together as a system. After its initialization routine, the slave processor remains in a waiting loop until a key is pressed on the keypad of the master processor. When this key is pressed, both processors enter a short start-up routine after which data collection begins.

The switch-scanning routine interacts directly (through the I/O ports) with the electrical tapeswitches. The routine scans each switch being hit (logic 0 to 1 transition) and cleared (logic 1 to 0 transition) and updates the switch-status flags accordingly. The data update routine is executed once every 25 msec (the flag-check interval). This routine uses the switch-status flags to determine the passage of vehicle wheels over a switch or set of switches. When a wheel has completely passed over a switch, the wheel count is incremented. This count is stored in memory and is reset every second.

SYSTEM EVALUATION

Data collected from nine 15-min test periods were used to evaluate the performance of the microprocessor system. This evaluation involved three steps:

1. For each data collection period, true average travel time valves were determined as follows: (a) Each microprocessor was used to display elapsed time on a light-emitting diode (LED) display. Each time a vehicle crossed the upstream or downstream set of detectors, its arrival or departure time was recorded. (b) For each vehicle, travel time was determined by subtracting that vehicle's arrival time from its departure time. (c) These travel time values were averaged to obtain the true average travel time.
2. Data collected by the microprocessor system were used to calculate average travel time estimates. The percent difference (or percent error) was calculated between the microprocessor system values and the true values.
3. The arrival and departure times for each vehicle in each data collection period were used to simulate the conditions that would be seen by a field observer. The input-output method was simulated using a 15-sec sample interval, and the average travel time values obtained were compared with the true average travel time values. After the accuracy of the field-observer method had been evaluated, it was possible to compare both field data collection methods to determine which was the more accurate.

A FORTRAN program was developed to accept the microprocessor-collected data and to perform the calculations necessary to transform this data into values of

1. Average travel time (true) in seconds,
2. Average travel time (microprocessor system) in seconds,
3. Average travel time (field observer) in seconds,
4. Percent error between the true values and the microprocessor system values, and
5. Percent error between the true values and the field-observer values.

In addition, average approach delay values were calculated using a free flow travel time that was taken as the average travel time of 50 unimpeded vehicles.

To compare the accuracy of the microprocessor system with that of a field observer, it was necessary to simulate the operation of a field observer with the data analysis program. A simple input-output method was simulated using a 15-sec sample interval. For every sample interval, the observer determined the number of vehicles crossing the upstream and downstream detectors. The number of vehicles on the approach delay section was then calculated for each sample interval. The summation of these values was used to obtain the total travel time in vehicle-seconds and the average travel time was determined by dividing the total travel time by the number of vehicles arriving during the data collection period. The microprocessor data were reduced in a similar manner except that wheel counts, rather than vehicle counts, were used. Also, a sample interval of 1 sec was used to provide accuracy greater than that of the field-observer method.

The next step was to determine the accuracy of the microprocessor system and the field-observer method with respect to the true travel time values. This was done by calculating the percent error of each method. A positive error indicated that the measured value was greater than the true value and a negative error indicated that the measured value was less than the true value.

Finally, the desired system output, average approach delay, was calculated for each data period by subtracting the free flow travel time from the average travel time.

Average Travel Time and Delay

Data were collected on the southbound approach to the intersection of Progress Street and Giles Road in Blacksburg, Virginia. This intersection handles very low traffic volumes and is under pretimed signal control. The low volumes at this location were desirable for system testing. Data were collected in 15-min periods beginning at 11:00 a.m. and ending at 6:00 p.m. In this time period, nine sets of data were collected.

Table 1 gives the true average travel time values for each data collection period as well as the values measured by the two field measurement techniques. A free flow travel time for the study approach was determined by averaging the travel times of 50 vehicles that passed through the intersection without stopping. This free flow travel time was subtracted from the average travel time values to obtain estimates of average approach delay. The average approach delay values are given in Table 2.

Percent Error and Sources of Error

Field-Observer Method

Table 3 gives the true average delay values, the field observer values, and the corresponding percent error values for each data collection period. The errors for this method range from -1.52 to 14.68 percent. In addition, the errors are both positive and negative indicating both overestimation and underestimation of the true average values.

The primary source of error with the field-observer method is the length of the sample interval. This interval must be long enough to accommo-

TABLE 1 Average Travel Time Values

	Travel Time (sec)		
Period	True	Microprocessor	Field-Observer
1	33.0	31.5	32.5
2	19.9	19.9	21.4
3	30.6	30.4	27.0
4	33.6	33.2	31.9
5	27.3	26.9	31.3
6	18.9	18.7	20.0
7	20.4	20.3	19.5
8	22.3	21.7	20.4
9	13.8	13.5	15.0

TABLE 2 Average Delay Values

	Delay (sec)		
Period	True Average Approach	Microprocessor Average Approach	Field-Observer Average Approach
1	23.1	21.6	22.6
2	9.9	9.9	11.5
3	20.7	20.5	17.1
4	23.7	23.3	22.0
5	17.3	17.0	21.3
6	9.0	8.8	10.1
7	10.5	10.3	9.6
8	12.4	11.8	10.4
9	3.9	3.6	5.1

TABLE 3 Percent Error Comparison, Field-Observer Method

Period	No. of Vehicles	True Travel Time (sec)	Field-Observer Travel Time (sec)	Error (%)
1	6	33.0	32.5	-1.52
2	7	19.9	21.4	7.91
3	10	30.6	27.0	-11.76
4	8	33.6	31.9	-5.20
5	12	27.3	31.3	14.68
6	18	18.9	20.0	5.88
7	10	20.4	19.5	-4.41
8	14	22.3	20.4	-8.65
9	15	13.8	15.0	8.70

TABLE 4 Percent Error Comparison, Microprocessor Method

Period	No. of Vehicles	True Travel Time (sec)	Microprocessor Travel Time (sec)	Error (%)
1	6	33.0	31.5	-4.42
2	7	19.9	19.9	0.00
3	10	30.6	30.4	-0.65
4	8	33.6	33.2	-1.30
5	12	27.3	26.9	-1.22
6	18	18.9	18.7	-1.10
7	10	20.4	20.3	-0.74
8	14	22.3	21.7	-2.52
9	15	13.8	13.5	-2.29

date the limitations of a human observer. However, as the length of the interval increases, the amount of error associated with the data also increases. It should be noted that the percent error for this method will decrease as the observed travel times increase. However, the error can be either positive or negative, which makes it difficult to apply a correction factor to the results.

Microprocessor System

Table 4 gives the true average delay values, the microprocessor system values, and the corresponding percent error values for each data collection period. The automated system errors range from -4.42 to 0.00 percent. The improvement in accuracy due to shortening of the sample interval is obvious. However, a new source of error has been introduced with the microprocessor system. This new error is due to wheel counting.

The error due to wheel counting is significant only when the travel time of a vehicle's front wheels differs greatly from the travel time of its rear wheels. When a red signal is encountered by a vehicle, it is possible for the front wheels of the vehicle to pass over the detectors before the vehicle comes to a complete stop. In this case, the measured travel time of the vehicle's front wheels will be less than the travel time of the vehicle causing the average travel time for the four wheels to be less than the true travel time of the vehicle. This explains the negative percent error values in Table 4. The fact that the percent error values for the microprocessor system are always either zero or negatives suggests that a correction factor could easily be applied to the these results.

CONCLUSIONS AND RECOMMENDATIONS

The automated data collection system developed in this research was proven to be theoretically sound. However, a problem was encountered whenever the device was taken into the field for data collection. Close examination of the software, including a detailed analysis of the time-dependent routines, revealed no problems. Similarly, in extensive laboratory testing using a function generator to simulate switch closures, the system displayed correct, predictable results.

For these reasons, it was concluded that the performance of the tapeswitches was probably not as "clean" as presumed in the design. A detailed analysis of the mechanical and electrical characteristics of the tapeswitches was beyond the scope of this research and is therefore recommended for further research.

As designed, the system cannot tolerate so-called "ghost" vehicles that may appear or disappear from entry or exit points between the two sets of detectors. It would be possible to develop additional hardware and software capable of handling intermediate entry and exit points within the approach delay section.

The possibility of applying a correction factor to the microprocessor data was alluded to previously. A large quantity of data would be required to identify a factor that could be correlated to the error introduced by wheel counting.

Finally, the feasibility of developing a more convenient detection scheme, or an improved version employing electrical tapeswitches, should be investigated. A single strip containing the separate 2-ft tapeswitches would significantly reduce the cost of setup and removal.

REFERENCES

1. J.F. Legere. Automated Collection of Vehicular Delay Data at Intersections. M.S. thesis. Virginia Polytechnic Institute and State University, Blacksburg, 1983.
2. C.C. Gardner, J.H. Kell, and W.R. Reilly. A Technique for Measurement of Delay at Intersections, Vol. 1. JHK & Associates, Alexandria, Va., Sept. 1976.
3. SDK-85 System Design Kit User's Manual. Intel Corporation, Santa Clara, Calif., 1978.
4. MCS-80/85 Family User's Manual. Intel Corporation, Santa Clara, Calif., Oct. 1979.
5. Ribbon Switches. Industrial Catalog C-8. Tapeswitch Corporation of America, Farmingdale, N.Y., 1981.

Publication of this paper sponsored by Committee on Traffic Flow Theory and Characteristics.