THE UNIVERSITY OF MICHIGAN
INDUSTRY PROGRAM OF THE COLLEGE OF ENGINEERING

THE USE OF A DIGITAL COMPUTER TO MODEL A SIGNALIZED INTERSECTION

Harry H. Goode
Carl H. Pollmar
Jesse B. Wright

This paper was presented at the January, 1956, meeting of the Highway Research Board of the National Research Council -- National Academy of Sciences in Washington, D. C.

ACKNOWLEDGEMENT

We wish to express our appreciation to the authors and the Engineering Research Institute for permission to give this paper limited distribution under the Industry Program of the College of Engineering.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THE USE OF A DIGITAL COMPUTER TO MODEL A SIGNALIZED INTERSECTION

1. Introduction

In 1952*, staff members of the Willow Run Research Center of the University of Michigan developed the following method for the study of automobile "ride" characteristics: A test car, with a light on its rear hubcap was driven slowly, at night, over a test road. Successive photographs of the light provided a functional representation of the test road. Accelerations were recorded at several points on the body of the car as it moved at various speeds over the test road. Next, the equations which governed the motion of the car in the vertical plane were set up on an electromechanical analog computer and with the test road function reproduced in a function generator, the corresponding computed accelerations were recorded with the computer car model "running" at the same speeds over the computer "test road". Since agreement between real and simulated acceleration values was not perfect, more factors were added to the equations to account for effects at first omitted as negligible. This work finally produced computed accelerations which agreed with test values within a pencil's width on the recorder. Armed with this "computer model", changes were made in various factors such as shock absorber characteristics and points of suspension and in each case the computer predicted the outcome of the changes before any expensive physical changes were made. The Chrysler Corporation is now designing ride into its cars with this tool.**

---

* Jeska, R. D., "A Comparison of Real and Simulated Automobile Suspension Systems", University of Michigan, Engineering Research Institute, February, 1953, Contract No. DA-20-018-ORD 12087 (Unclassified).

** Huebner, G. J., Jr., Chrysler Engineering Division, Address to Association for Computing Machinery Meeting, University of Michigan, 1954.

The method used is general and this paper is concerned with its application to the Traffic problem. The steps are represented in Figures 1.1 and 1.2 for the Traffic Process. In these diagrams, "independent quantities" would be the lighting cycle, the probability of turns, the average number of cars per hour arriving, etc. The "traffic process" would be the functioning of the intersection, a group of intersections, a network or a highway. The dependent quantities would be average delay, time thru a region, or congestion, etc. The remainder of the figure on modelling (1.1) is self evident. Prediction (1.2) would take the form of predicting the result of a change in lighting cycle, lane width, parking regulations, etc., by incorporating the change in the computer model and if the result is satisfactory, using it in the traffic process.

The application of these simulation methods to a process as complicated and discrete as vehicular traffic flow is not without precedent. The Bell Telephone Laboratories even without a digital computer successfully used such modelling techniques in predicting the performance of the number 5 crossbar, a new type of switching equipment, prior to its introduction in operating exchanges. The model was known as "The No. 5 Crossbar Throwdown". Our problem is of similar nature. The results of the present paper demonstrate further the feasibility of the development of the computer model as a powerful tool for traffic analysis.

2. Analysis, Simulation and Trial

We digress for a moment to discuss the relative merits of analysis, simulation and trial (and error). Table I summarizes the results of such considerations.

Table I

Criterion	Analysis	Simulation	Trial
Cost	Least	Medium	Most
Time	Least	Medium	Most
Reproducibility	Most	Medium	Least
Realism	Least	Medium	Most
Generality of results (if real)	Most	Medium	Least

Attacks on the traffic problem have been made in the past with the tools of both analysis and trial. By analysis we mean writing a mathematical expression to represent the traffic process and then manipulating it mathematically to determine values to be used in changing to better traffic conditions. Trial involves a change in the real traffic situation, and subsequent correction if it turns out badly. Simulation is actually a combination of both methods but allows attack on the most complicated of processes, which analysis does not, and on the other hand, does not affect traffic until a solution has been reached, which trial methods do. Simulation, it will be noted in Figure 2, is almost always midway between analysis and trial. But as the situation being studied becomes more complex, the differences between methods in terms of cost, time, etc. become more pronounced until finally simulation is the only feasible method.

There is one aspect of the attack on a problem in which simulation is better than both analysis and trial. When a suggested solution is being examined, it is in the light of some measurable criterion which determines its usefulness. Sometimes, as is the case with traffic, several criteria may be used (time thru region; flow across boundary; or congestion as measured by average delay), or different criteria may be used at various times.

In analysis we may use only those criteria which are mathematically tractable (e.g. least squares, but not maximum absolute deviation), and in trial, we choose only one criterion because even one is costly to measure. In simulation, we may select any criterion and as many as we like, measuring them continually if necessary.

In the traffic process, the goals achievable by simulation are clearcut and offer a pronounced payoff. The worst aspect of our lack of knowledge of the traffic process today is the absence of a reasonable estimate of the theoretically achievable improvement in traffic flow by traffic control methods. Estimates derived from data on speeds and number of cars per hour are so far from the practical situation as to be useless. Clearly we need a method for estimating this theoretical upper bound. For if we had it, we would know that either (a) only a small improvement is possible with control methods and we may save all the money we are investing in control methods for use in radical solutions like the dispersal of cities, moving sidewalks, etc., or (b) a sizeable improvement is possible. In case (b) we may use the same tool that gave the theoretical upper bound, the model, to determine necessary changes. But further than that, the logic governing automatic traffic control systems, such as those which automatically meter traffic flow in order to control lights, must be studied. The model offers the means for such a study. Moreover, it is possible that future traffic control will require car control. This may be done, for example, by directly controlling the movement of each car or by a broadcast which informs every driver, perhaps by means of a cathode ray tube display, of the traffic conditions for blocks around him. Car control will have to be investigated and its techniques developed and evaluated without interfering with actual traffic. The computer model is an ideal instrument for use in such research.

The simulation model is not another means for accomplishing what we can do today, but is a tool for solving a problem which cannot be solved today. To see this, we quickly review present methods of traffic design. A bad situation in city traffic is brought to the attention of the traffic engineer by a death or perhaps a string of accidents at an intersection. He examines the characteristics of the intersection in terms of the light cycle, the number of lanes, the type of area, etc. By changing some factor which is controllable, he then seeks to increase capacity (cars per hour) to a value greater than the number per hour observed at the intersection. To aid him in his choice he uses some reference such as the Highway Capacity Manual. And frequently he gets a reasonable solution to the intersection problem. But none of the traffic engineers consulted claimed to be able to predict the effect of a change at a given intersection on adjacent intersections. On the other hand, all agreed the major traffic problem in cities concerns sets of intersections, or regions, not single intersections. If this were not so, the automatic controller would be the final solution. But as in so many problems, optimizing at a point does not imply optimization over the whole region.

3. A Computer Model for Simulating Traffic

The traffic model discussed here was set up on a digital computer, the MIDAC, at the University of Michigan. It is intended as a demonstration of the proposed technique. No experimental verification of its applicability could be undertaken with the means available, but it contains enough of the real traffic situation to make it interesting.

In developing the computer model, the fundamental unit to be simulated was first chosen. It consisted of a street intersection and the lanes approaching it. Such a combination was called a cross-block and is illustrated in Figures 3.1 and 3.2. The cross-block can be considered a "building block" from which the network of streets of a city can, in many cases, be constructed. A simple combination of two cross-blocks is shown in Figure 3.3.

After the general characteristics of the cross block had been decided upon, a method was devised for representing the position of every car in the cross-block at any particular instant of time. This was extended in the light of the operations available on MIDAC to a method for simulating traffic flow. This method requires the computer to determine the motion of and to move each car in the cross-block individually.

Once the overall picture had been sketched the detailed development of the rules according to which the computer determines the movement of the cars, their introduction into the lanes, and so on were developed. The formulation of these rules concluded the first phase in the construction of the computer model. The final phase, the writing of the computer program to realize the model on MIDAC, was then carried out.

3.1. General Characteristics of the Cross-Block

The first step in the construction of the model is to define a "typical" cross-block to be simulated. The abstract and idealized cross-block which constitutes the computer model will be an approximation of this "typical" one.

The cross-block upon which the MIDAC model is based has streets 22 feet wide and lanes in the neighborhood of 400 feet long. The vehicles travelling through it are assumed to average about 18 feet in length, being in any case more than 11 feet and less than 22 feet long. They travel in the lanes at 30 mph when unobstructed, pass through the intersection under the control of a three phase traffic light, and turn right, left or go straight ahead according to the "desires of the driver". Their position in the block is given by the position of the mid-point of the front bumper.

To avoid complications cars are not allowed to pass one another and interference from parked cars and pedestrians is assumed to be negligible.

Instead of conceiving of a lane as a two dimensional strip, it can clearly be thought of as a line, and the position of a car in the lane can be thought of as corresponding to the position of a point on the line.

In the idealized cross-block to be represented in the computer a further idealization takes place. The line representing a lane is replaced by a sequence of 40 points. A car in the lane must be thought of as being at one of these 40 points. As it moves down the lane, it "jumps" from point to point.

In our model there are four lanes. With each lane are associated four paths lying within the intersection, one followed by right turning cars, one by cars going straight ahead and two by cars turning left. These are called ρ, α, λ, and $\bar{\lambda}$ respectively. They are also considered to be sets of points and are shown for a single lane in Figure 3.2. The end point of λ is of special importance and is called the <u>Left Turn Zone</u>.

Cars move down the idealized lanes and paths by "jumping" from one point to the next. When a car is moving, it "jumps", and thereby covers the distance between two adjacent points, every quarter second.

The distances between points are different for the lanes and the several intersection paths. Consequently, the simulated speeds vary. The distances and corresponding speeds are:

	Length	Number of points	Distance between points	Speed
Lane	429 ft	40	11 ft	30.0 mph
α	33 ft	4	8.25 ft	22.5 mph
ρ	18.4 ft	5	3.67 ft	10.0 mph
λ	19.8 ft	9	2.20 ft	6.0 mph
$\bar{\lambda}$	18 ft	6	3.00 ft	8.0 mph

3.2. Representation of a Distribution of Cars in a Cross-Block

This idealized cross-block can be represented in the computer. To each lane and to each intersection path there corresponds a register, twenty in all since there are four lanes and sixteen paths. The points of the lane or intersection path are associated one to one with some digit positions of the corresponding register (not all digit positions need be used).

To represent the distribution of cars in a lane at a particular instant of time, it is only necessary to specify the presence or absence of a car for each point of the lane. This is done by having ones in the digit positions corresponding to points at which there is a car and zeros otherwise. For any particular instant of time, t, the set of such distributions for all lanes and intersection paths is called the <u>distribution</u> for t.

As an example of these ideas, consider the first 10 points of a lane. Label the points at which there is a car by C. Then the following scheme shows the points of the lane, the correspondence of these points with the digit positions of the associated register, and the distribution of zeros and ones corresponding to the distribution of cars in the lane:

```
               C   C       C   C   . . .
     Lane:     x x x x x x x x x x . . .
               ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
 Register:     1 0 1 0 0 0 1 0 1 0 . . .
```

This form of representation is particularly suitable for MIDAC because numbers are stored in its registers in binary form (i.e. the digits are either 0 or 1 instead of 0, 1, . . . 9 as in decimal numbers). Each register contains 44 digit positions, more than enough for our lanes which require 40 digits.

3.3. Simulation of Traffic Flow in a Cross-Block

In a moving picture, motion is simulated by observing a sequence of pictures taken in temporal succession, the time between successive pictures being very small. The motion of traffic in a cross-block can be similarly conceived as a temporal succession of car distributions taken (in the case of our model) at one quarter second intervals. The computer simulates the flow of traffic by constructing a sequence of car distributions for successive moments of time.

Consider again the first ten points of a lane. Let the binary number representing the distribution of cars in this lane be stored in register, R. Then the contents of R at successive intervals might be:

At time t : 1010001010 . . .
At time t + 1/4 : 1010010010 . . .
At time t + 2/4 : 1010100010 . . .
At time t + 3/4 : 1010100100 . . .

If the cars are considered to be numbered from left to right then car 1 and car 2 remain stationary; car 3 moves up until it is right behind car 2; and car 4 after remaining stationary moves up until only two points separate it from car 3. Because of the restriction imposed on car length (greater than 11 and less than 22 feet) cars in a lane are separated by at least one point (i.e. there must be at least one zero between two ones).

Although it is not essential for understanding the MIDAC model, some insight into the working of the construction process may be gained by seeing how a car is moved in a lane or intersection path, how the computer can tell if there is a car at a point, and how it can choose among different courses of action.

A car (i.e. a one) is moved from one point to the next by addition. Addition in MIDAC is binary, since the numbers are in binary form. It is defined by:

Binary Addition Table

	0	1
0	0	1
1	1	10

In order to get the distribution for t + 1/4 from that for t, we proceed as follows:

Distribution for t	:	1 0 1 0 0	0 1	0 1 0
Shift Number	:	+0 0 0 0 0	0 1	0 0 0
Distribution for t + 1/4	:	1 0 1 0 0	1 0	0 1 0

A MIDAC operation used in deciding if there is a car at a particular point (i.e. a one at the corresponding digit position) is "logical extract". It isolates any specified digit of a register by putting it into some other register, E. E is then studied by the "comparison operation". For example, if the digit position under examination contains a one, the logical extract operation will put a 1 in E; if it contains a zero, it will put a zero in E. In the construction of a car distribution, the examination of a digit position may lead to two separate courses of action:

1. if there is a car there - to the determination of and the carrying out of the car movement.
2. if there is no car - to the examination of the next digit position.

The choice is made by the comparison operation, which compares the number zero to the number in E. If the latter is greater than zero, then (1) is followed. Otherwise (2) is carried out. This operation is used wherever choices must be made, e. g. between the steps to be taken if the light is red or green.

3.4. The Determination of Successive Car Distributions

The construction of a car distribution employs procedures for entering cars in the lanes, for determining the direction of cars entering the intersection, for cycling the traffic signal, and for taking account of "traffic conditions".

In the MIDAC model, cars enter the four lanes at point 40. They are "generated" by a process which makes use of pseudo random numbers. At the end of each quarter second interval a random number subroutine generates a random number R (between 0 and 1) for lane N. R is then compared to a number M. If $R < M$ a car is generated for N. Clearly by changing the value of M, which is a parameter of the program, the "average number of cars per hour" entering the lane may be controlled. For example, if $M = 0$, R cannot be less than M and so no cars would be generated; on the other hand if $M = 1$, R would always be less than M so a car would be generated every quarter second. For the small intervals of time used, the cars generated in this fashion will have approximately a Poisson distribution.

In order to avoid "piling cars on top of one another", which would occur whenever cars are generated at successive quarter second intervals, cars are first put in a register, B. No attempt is made to take account of the position of cars placed in B. Its contents merely indicate the number of cars waiting to enter the lane. As space becomes available, the cars are moved from B into N.

A similar procedure is carried out for the other three lanes.

When the cars leave the end points of the paths in the intersection, they are dropped from consideration.

If several cross-blocks were to be fitted together, the picture would be somewhat different. Cars would be generated in the manner described above at the free ends of lanes, i.e., those ends not attached to an intersection path - Figure 3.3, W_1, N_1, S_1, N_2, E_2, and S_2. However, they would not be so generated at E_1 and W_2. Cars leaving cross-block I at point A, for example, would become the inputs for lane W of cross-block II and would only be dropped from consideration when they left free ends of paths.

At an actual intersection an observer of traffic cannot tell which way a particular car will turn. However, he may know the probabilities for a right turn, left turn, or for going straight ahead. This characteristic of traffic is simulated by associating a turn register with each lane. It can be thought of as representing the turn indicator of the car nearest the intersection in that lane. If the turn register contains a 1 then the car nearest the intersection will turn right. A zero indicates straight ahead and a two a turn left. After the car has made its turn the turn register must be set to indicate the next turn. Suppose the probability for a right turn is .3 and that for a left turn, .2. Then to determine what the next turn is to be, a random number is generated. If it lies between 0 and .3, the next turn is to the right; between .3 and .8 it is straight ahead; and between .8 and 1.0 it is left. Since the numbers are equally likely throughout the 0 - 1 range, cars will turn with the desired frequency in each direction. These probabilities are parameters for the program and as such may be varied at will.

The three phase traffic light is simulated in the computer by a light register. The register contains a zero if the light is red, a one if the light is green, and a two if the light is amber. The duration of each phase of the light is controlled by counting the number of quarter seconds during which it has been continuously in that phase and changing to the next phase when the counter has reached a certain specified value. The duration of red and green for the N and S lanes are parameters of the program and may be set to anything desired. The duration of amber, however, is fixed at three seconds (12 quarter seconds), which is long enough to enable all cars entering the intersection on green to pass through before the light turns red.

The effect of the turn register, the light, and the traffic can be seen by considering the manner in which the computer moves the "cars" and thereby determines the next distribution in the cross-block. Behind the specific rules the computer obeys are the following general principles: (1) Cars approaching the intersection give the right of way to cars which are in the intersection but not in the left turn zone and (2) cars in the left turn zone, give right of way to cars which will cross their paths.

The cars on α, ρ, $\overline{\lambda}$ are first considered and are moved up one point. This can be done without any consideration of the light or traffic because they are not allowed to enter these paths until the way is clear for their complete negotiation.

A car approaching the Left Turn Zone is likewise automatically moved up. If there is a car, C, in the left turn zone, the light is checked. If it is red, C completes the turn. If it is green or amber, it examines the right and straight ahead paths of the opposing lane (e.g. S if C is turning left from N). If they contain a car, C remains where it is. If they are empty, the traffic in the opposing lane is examined. If there isn't any car within 55 feet of the intersection (i.e. if there are zeros corresponding to the first six points) the car continues its left turn. If there is a car within 55 feet, the turn register is examined. If it indicates that the nearest car is to turn left, C completes the turn otherwise it remains in the left turn zone.

If the car is at point one (i.e. is about to enter the intersection) the light is checked. If it is red, the car remains at point one. If it is green, the computer examines the turn register. If a right turn is indicated, the digit corresponding to point one is made zero and the digit corresponding to the first point of the right turn path is made 1, (i. e. the car turns right). If a left turn is specified, the left turn path is examined. If it is empty the car proceeds, otherwise it remains at point one. If the turn indicator shows straight ahead, the car proceeds.

A car a point two of a lane always moves to point one and a car at point three advances unless there is a car at point one or a car entering the intersection from this lane. These facts are determined by checking to see if zeros or ones are associated with the points of these intersection paths.

Cars further back in the lanes follow rules designed to maintain a distance of at least 55 feet between the front bumpers of moving cars. This was deemed a reasonable minimum distance for cars travelling at thirty miles per hour. A moving car will, of course, approach a stopped car until there is a distance of 22 feet between their front bumpers.

4. Study of the Model

The MIDAC under the control of this program simulates the flow of traffic. Cars enter and move down the lanes. They pass through the intersection under the control of the traffic light, turning right, left, or moving straight ahead, before leaving the area of observation. The observer operating the MIDAC, however, knows nothing of these things.

In order to study the flow of traffic provided by the model additional programs called "study routines" are employed. Study routines may be used to gather statistics and to make calculations regarding the flow of traffic in the model. In fact, study routines may be written to investigate the traffic from any point of view desired.

A routine to compute the "average delay" at an intersection has been written and illustrates the nature and use of such routines.

The average delay for cars in a given lane is the average actual time needed to go from the far end of the lane (point 40) through the intersection less the minimum time for negotiating this same course. The routine used calculates an approximation of this average delay. The average time needed to pass through the lane and intersection is approximated by counting the ones (i.e. the cars) in the lane and its associated intersection paths every quarter second, accumulating this count and dividing by the number of cars leaving the lane's intersection paths.

The average delay was computed under many different conditions. The average number of cars arriving per hour from each direction, the duration of the light cycle, the fraction of the cycle the light was green (thus determining all other light phase durations), the fraction of cars going right, left, and straight ahead all were varied from run to run. The values of the variables chosen and the results obtained are presented in Figures 4.1 and Table I.

Although it has not been done, a study routine could be written which would display the traffic moving in a cross-block on a cathode ray tube in a manner similar to television. The outline of the cross-block could be permanently recorded on the face of the tube and the traffic would appear moving within this outline as if viewed from high above.

This form of presentation was actually used on MIDSAC, another Michigan Computer, where a pool game, with 15 balls and a cue ball, was simulated. The boundaries of the pool table were recorded on the face of the cathode ray tube, the balls were represented by circles of light, which moved within these boundaries, and the cue stick appeared in the form of a shaft of light. In order to make a shot the player would set the cue stick to the desired angle by turning a knob and would then throw a switch. MIDSAC would thereupon calculate the position of all balls in real time taking account of elasticity, acceleration, cushion angle, etc. - everything but "English" - and the cathode ray tube display would show the balls moving around this "pool table". A similar technique would work with traffic.

5. Critique and Prognosis

It should be emphasized that realism is not of interest for its own sake. It is necessary only insofar as it provides answers from the model which are applicable to the real situation. Thus, in Physics, the wave or particle model may be used as required although it is difficult to conceive of both being correct, at the same time. The decision to add complexity to a model should depend not on any improvement in its realism but only upon whether the change produces answers which are in closer agreement with measurements.

The methods outlined here are not cheap in absolute cost. Digital machine staff and digital machines are expensive, but the potential power of the tool and the foreseeable contribution are so great that by comparison the costs are negligible, even if in the hundreds of thousands.

Development of the method will require a large initial investment. Further sizeable expenditures will undoubtedly be needed as traffic workers develop the use of more powerful techniques, which realize to an ever increasing extent the potentialities of the model method.

The ratio of machine to model real time for the MIDAC model is 3.2 to 1, i.e., to simulate one minute of real time takes 3.2 minutes of computer time. Under these conditions, if 50 intersections were handled, and assuming no increase due to interconnections, the ratio would go to 160 to 1. This would seem almost intolerable if it were not for the fact that it can be much improved. This can be accomplished, for example, by using a faster computer, more sophisticated techniques of simulation, or equipment especially designed to handle traffic simulation.

Although all digital computers are fast, they cover a wide range of speeds. the NORC computer, which is 100 times as fast as MIDAC, would cut the ratio back to 1.6 to 1, and even faster computers can be built if required.

Simulation techniques are foreseen at this writing which would reduce the ratio by a large factor. For example, if enough were learned about intersections, it would be possible to associate a function with each of the four directions in which traffic leaves the intersection. Each of these functions would have four variables representing the number of cars backed up in each of the four lanes. The values of each function would be the number of cars leaving the cross-block in the associated direction. These values would be determined in such a way that their time distribution is similar to that encountered in real traffic. A single evaluation of these functions would give the result of a light cycle computation. The investigation required to develop this technique is, of course, long and expensive.

In going from 1 to 50 intersections, it was necessary to multiply 3.2 by 50, because the computer could work on only one intersection (actually only one point of a lane or intersection path) at a time. If small digital computers designed for intersection simulation were provided for each intersection and then were hooked together to simulate a traffic network, all 50 intersections could be considered at the same time. Some consideration has been given to this approach by UCLA in the report.***

A program to develop the model method might begin with an attempt to simulate traffic in a small relatively isolated community, which bottle

---

***D. L. Trautman, Harold Davis, Jack Heilfron, Er-Chun Ho, Arnold Rosenbloom, "Analysis and Simulation of Vehicular Traffic Flow", Institute of Transportation and Traffic Engineering, University of California, Los Angeles, December, 1954.

necks the flow of traffic to and from work in a large city. After a successful simulation has been achieved and its efficiency tested, the method might be adapted to the simulation of a region of a city with less detailed representation serving for the region than for an intersection. In the possibility that such regions can be simulated lies the hope for large metropolitan areas. After this had been accomplished the traffic engineer could begin to study the flow of traffic between regions. Finally, since regions are not independent, the investigation of traffic might be aided by the development of models in which regions are joined together.

It is possible to imagine a time in the future when every large city will have its modelling team and computer which the traffic engineer will use as a tool to try out his projected changes, assuring himself of a smooth transition to a new optimum use of his city's traffic facilities. The small communities, unable to afford a large machine, would band together perhaps under state auspices, to use a single large machine. Since the model, once developed, can be changed with relatively little effort, even small communities will be able to develop the technique without having a machine standing by.

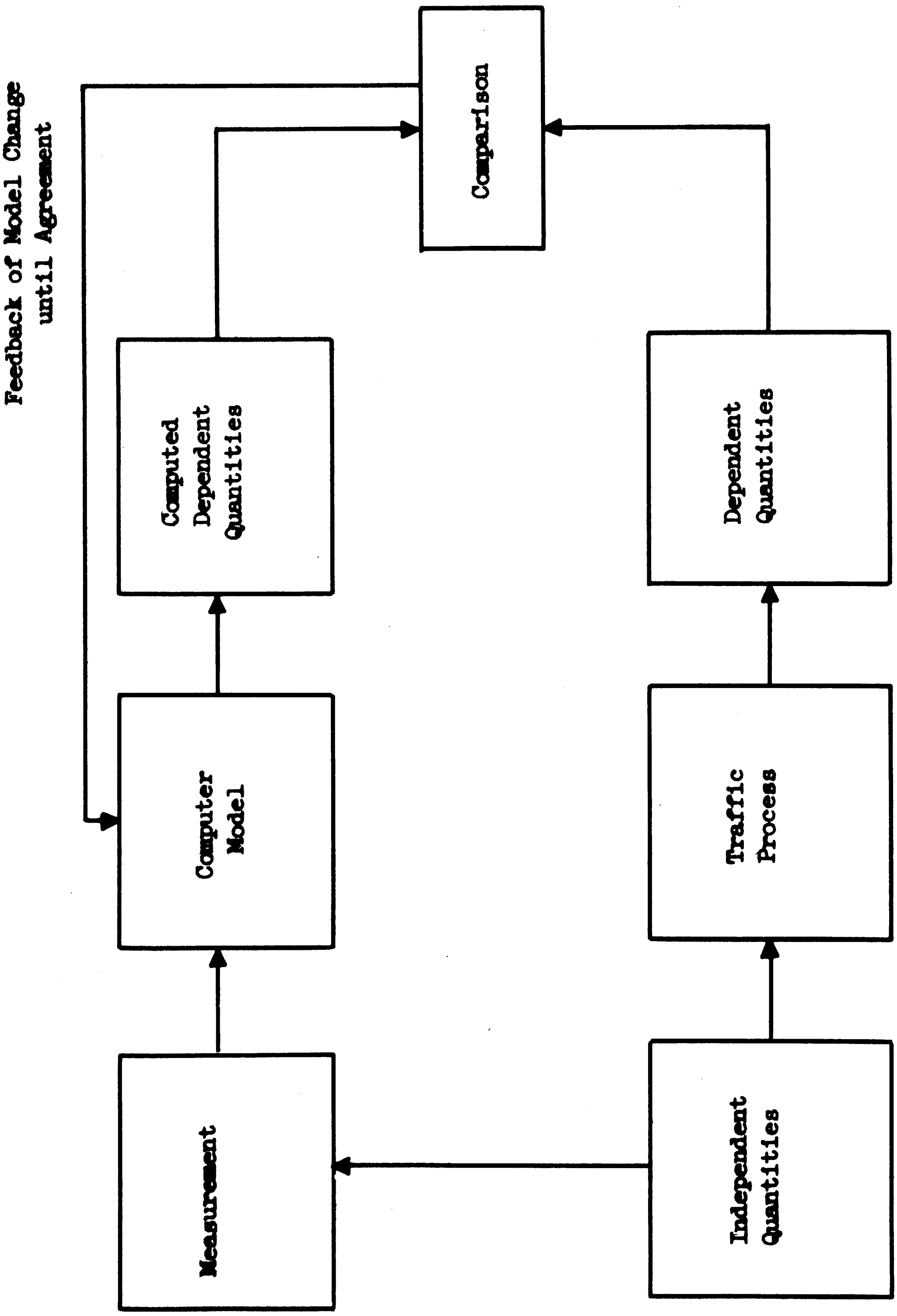

Figure 1.1. Modelling

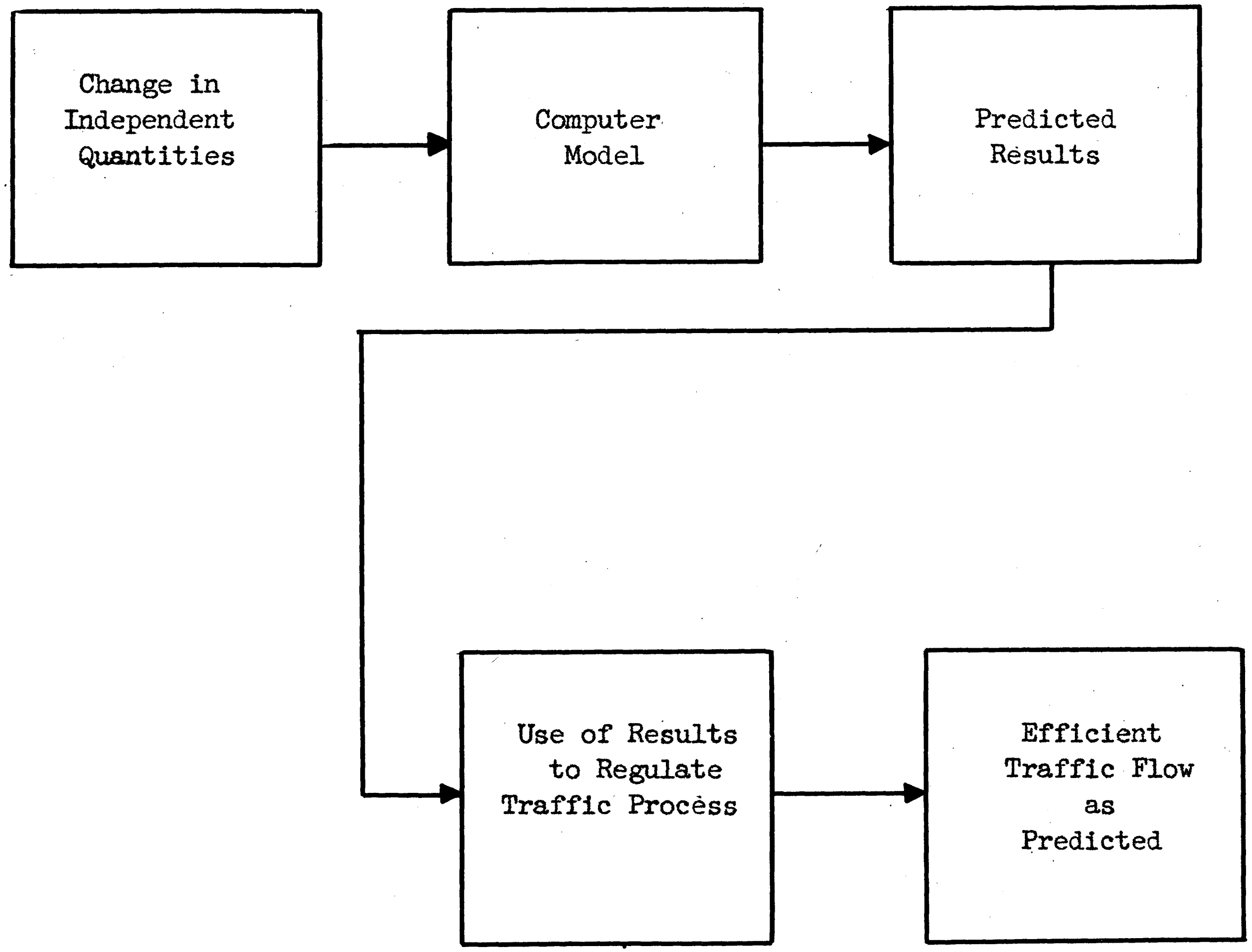

Figure 1.2. Prediction

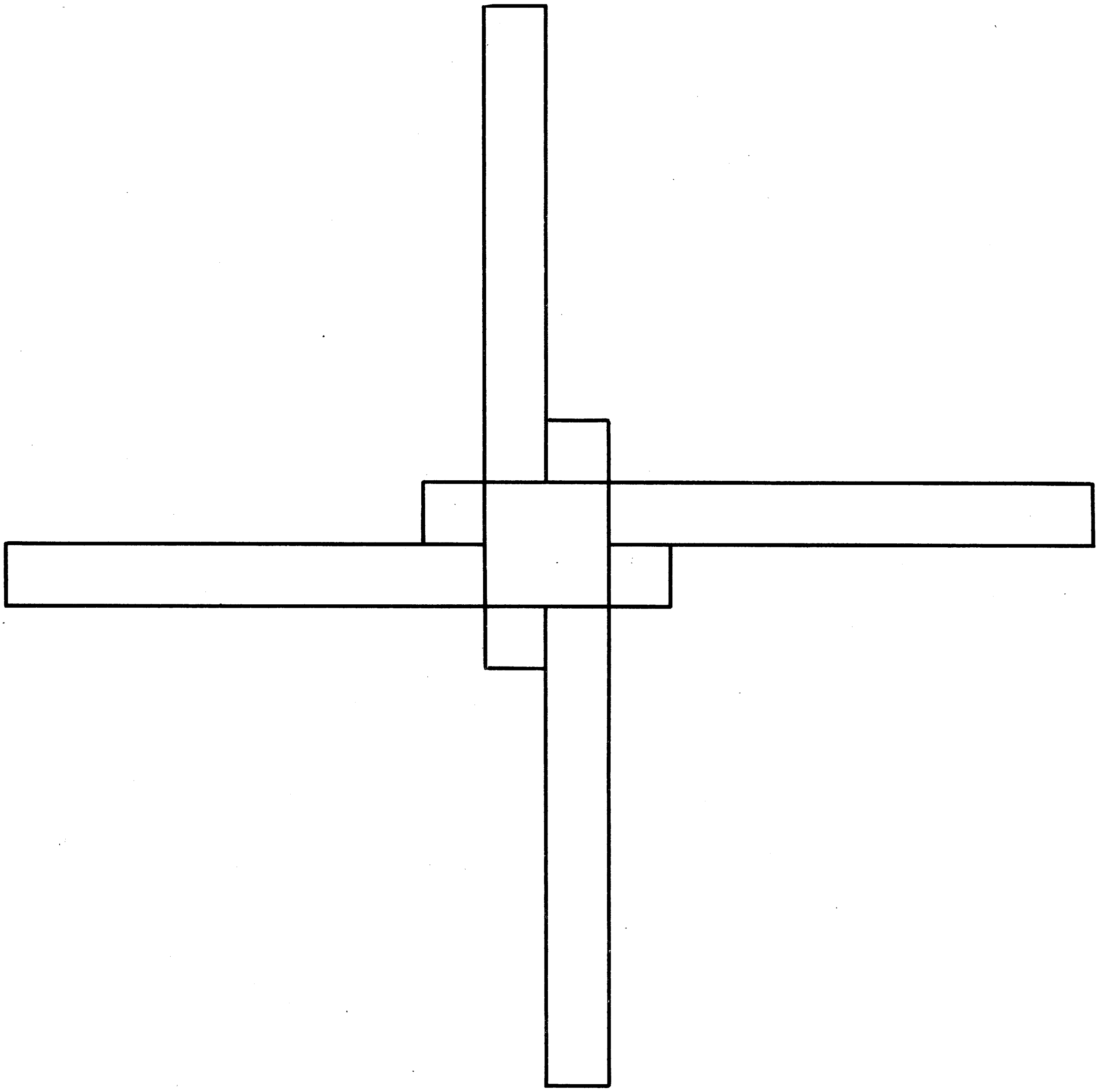

Figure 3.1. Simple Cross-Block

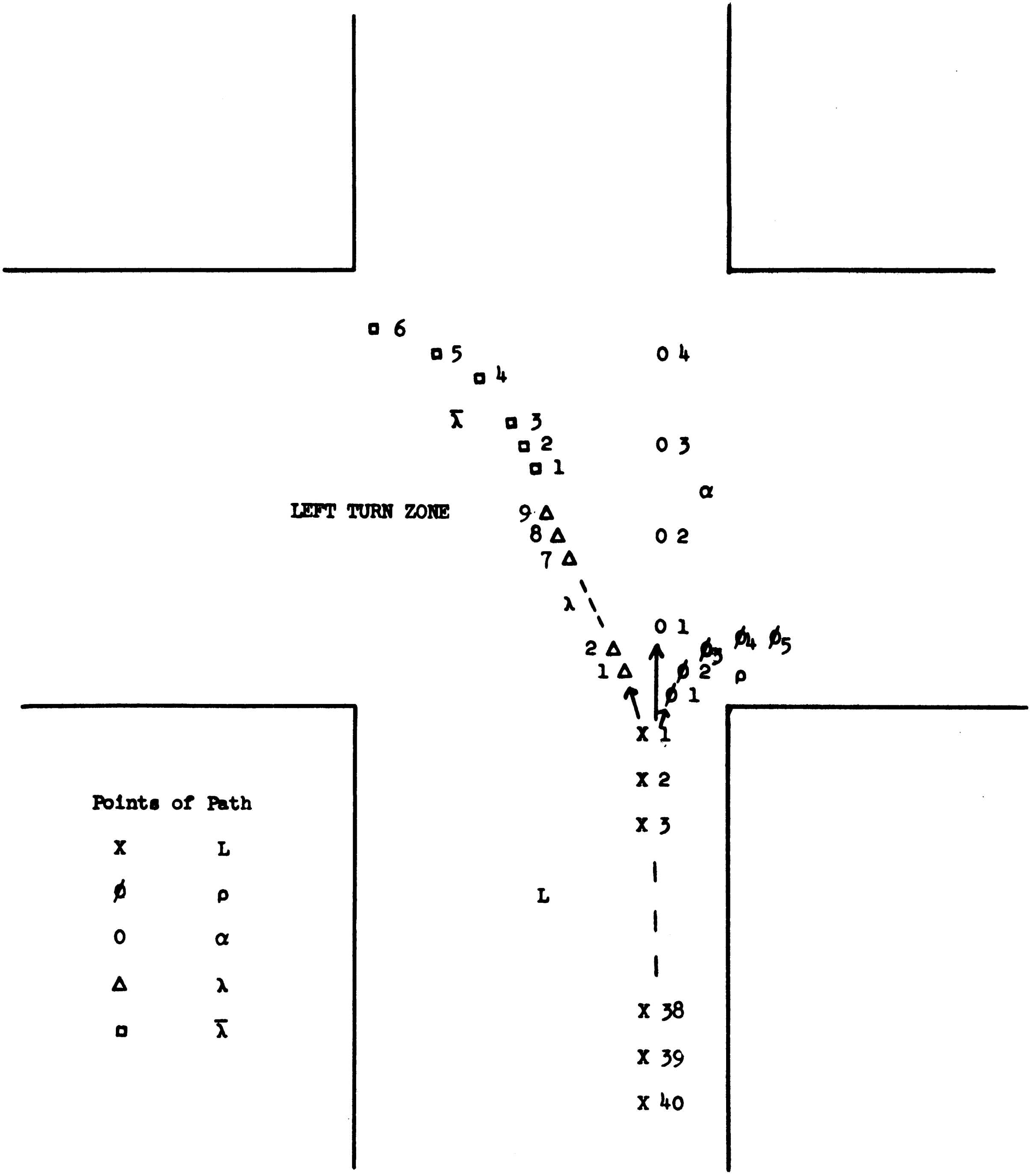

Figure 3.2. Cross-Block, Position of Cars

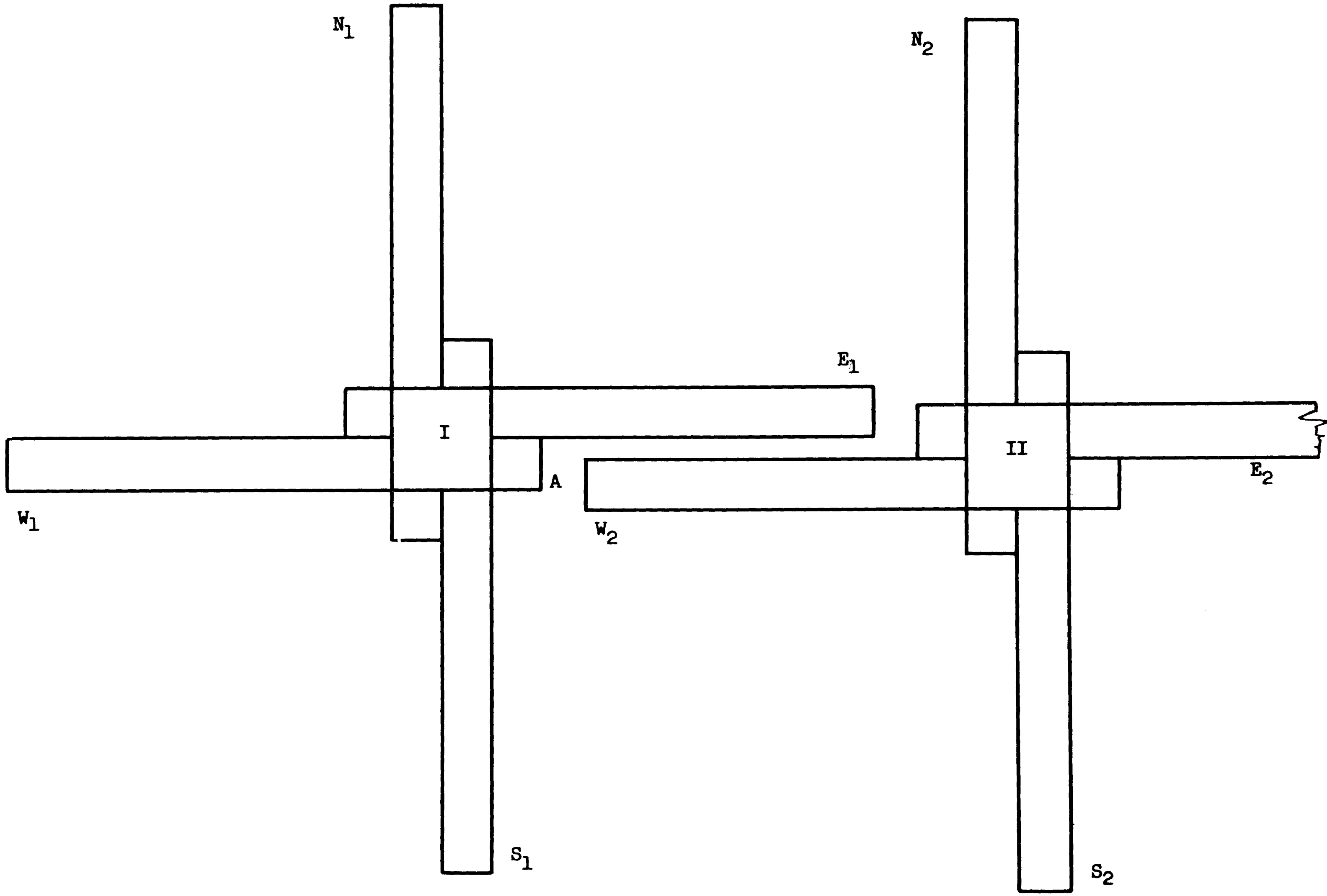

Figure 3.3. Combination of Two Cross-Blocks

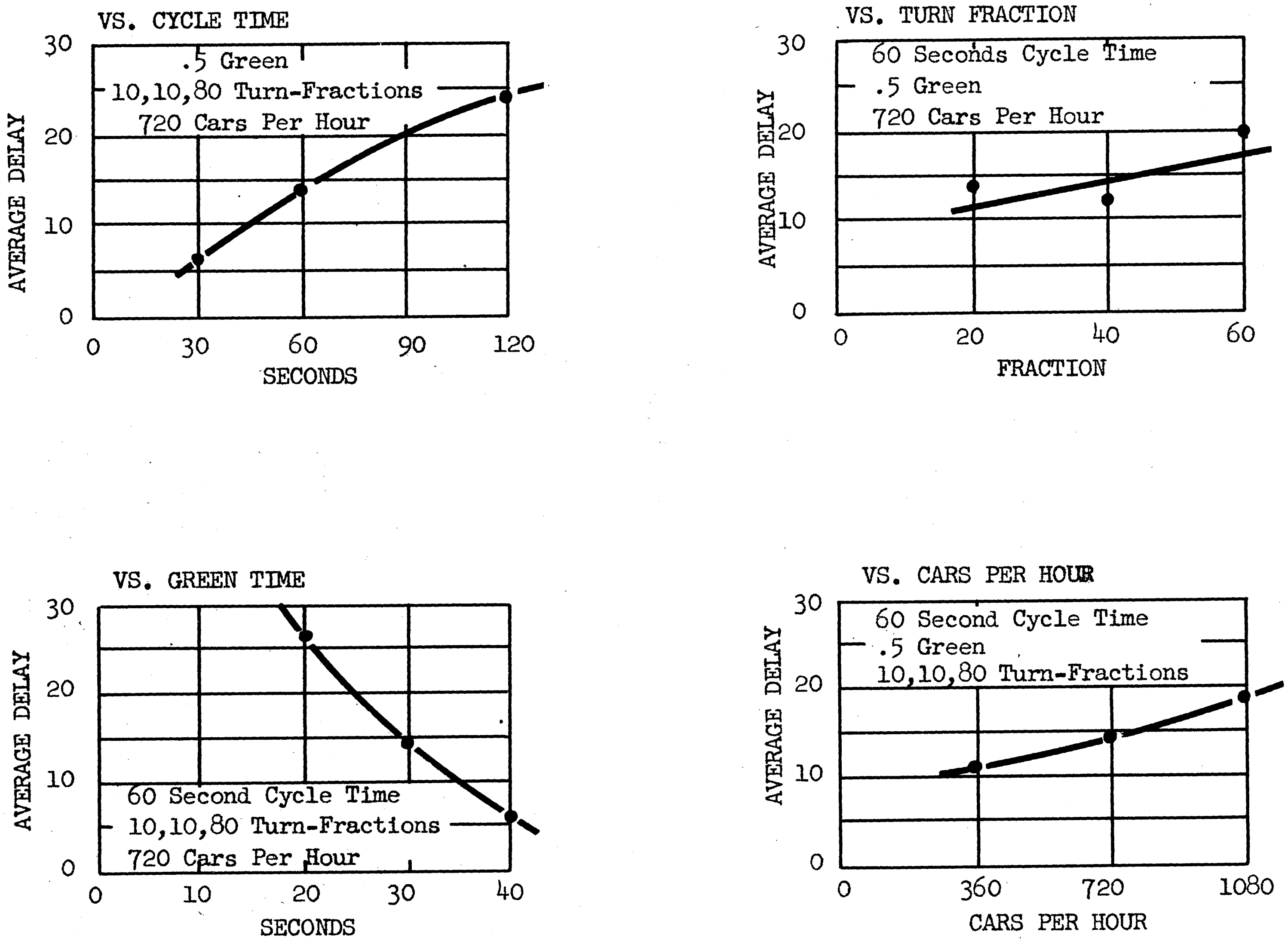

Figure 4.1. Average Delays

Number	Theoretical Rate of Car Flow Cars/Hour	Right Turn Fraction	Left Turn Fraction	Cycle Length (Seconds)	Green Time (Seconds)	Aver. Delay (Seconds/Car)
1	360	10	10	60	20	19.6
2					30	11.2
3					40	5.4
4	720	2	82	60	30	30.4
5		8	28	60	30	13.3
6		10	10	30	12.5	10.5
7					15.0	7.0
8				60	20	26.5
9					30	14.2
10					40	5.4
11				120	60	24.6
12						
13		20	20	60	30	12.7
14		30	30	60	30	19.7
15	1080	10	10	60	20	51.9
16					30	18.8
17					30	18.5
18					40	7.7

Table II. Result of Runs