Control Algorithm and Theoretical Analysis of a Grade-Crossed Intersection in a Computer-Controlled Vehicle System

MASAHITO KURIHARA, MEMBER, IEEE, IKUO KAJI, YOSHIO HAMAMATSU, MEMBER, IEEE, AND KATSUHIRO NAKADA

Abstract—**A control algorithm and a probability-theoretical traffic analysis of a grade-crossed intersection of the computer-controlled vehicle system (CVS), a computer-controlled personal rapid transit (PRT) system developed in Japan, are presented. The control algorithm is simple but realistic and optimal in the sense that delay is minimized under the following constraints: the first-come first-served rule, the maneuver length limitation, and the collision-freeness condition. The analysis is completely based on this detailed control algorithm; random arrival of vehicles is assumed, and the behavior of the algorithm is formulated as a simple Markov chain known as a one-dimensional random walk. The exact and explicit expressions for some performance measures such as delay, abort rate, and throughput are obtained.**

I. INTRODUCTION

THE MOVEMENT of people and goods in densely populated urban areas is becoming increasingly difficult. Existing transportation systems are already operating at or near their capacity, and it is clear that novel systems will be needed to cope with predicted future demands. Up to now, various kinds of automated rapid transit systems have been proposed [1]–[9]. The computer-controlled vehicle system (CVS) [9]–[14] developed in Japan[1] is one of these systems and is classified as a personal rapid transit (PRT) [1], [2] system. Its main features are

1) on-demand and non-stop between origin and destination,
2) nearly door-to-door service,
3) personal (four seats) and private,
4) automatic operation, and
5) mass transportation.

The CVS is controlled by the cell-following (or moving target) and quasi-synchronous scheme [9]–[12], [15]–[19]. The guideway image is created in the computer, and hypothetical cells are moving along the guideway image in accordance with a predetermined position–time profile. Vehicles are usually required to follow the hypothetical cells currently assigned to them, but at intersections the cells they should follow may be changed in order to avoid collision at merge points. This results in deceleration and delay of the vehicles, and the possibility of needing to reroute (abort) en route vehicles [18], [19], [25]. Since the effective system capacity will be dependent on intersection operating characteristics (e.g., delay, abort rate, etc.), it is vitally important to the entire system that intersection control algorithms be most efficient and that their traffic (or queuing) characteristics be analyzed quantitatively.

To do this type of analysis, there are two typical approaches: computer simulation and probability theory. Though computer simulation [19]–[23] provides the means of monitoring many facets of performance of a sophisticated process, it requires time and money. The theoretical method [24], [25] can yield speedy, inexpensive, and rigorous results. Thangavelu [24] analyzed a grade-separated intersection of a PRT highway network by using conventional queuing theory (M/D/1 and G/M/1), assuming that the vehicles accelerate and decelerate (i.e., maneuver) only in the off lines (i.e., turn ramps). Hamamatsu [25] analyzed, by Markov chain techniques, a grade-separated intersection, where a low-speed way in a local-area network merges into a highway, assuming that the vehicles maneuver in the main line. But these studies seem unsatisfactory to the authors, because the analysis is not based on real algorithms but on conventional queuing models which result in inaccurate results. Apart from these disadvantages of the previous studies, the most essential point that motivated our study is that all of the intersections that have

Manuscript received August 14, 1984; revised January 3, 1985.

M. Kurihara and I. Kaji are with the Department of Electrical Engineering, Faculty of Engineering, Hokkaido University, Sapporo, 060, Japan.

Y. Hamamatsu is with the Department of Electronic Engineering, Faculty of Engineering, Tamagawa University, Tokyo, 194, Japan.

K. Nakada is with the Department of Information Science and Communication Technology, Faculty of Engineering, Tamagawa University, Tokyo, 194, Japan.

[1]The CVS was developed by a group of eight Japanese corporations—Toyo Kogyo Co., Ltd.; Mitsubishi Heavy Industries, Ltd.; Sumitomo Electric Industries, Ltd.; Nippon Electric Co., Ltd.; Nippon Steel Corporation; Fujitsu, Ltd.; Hitachi, Ltd.; and Tokyo Shibaura Electric Co., Ltd.

0018-9472/85/0500-0376$01.00

been studied so far by simulation or theory are "grade-separated," whereas the CVS in Japan adopts "grade-crossed" intersections.[2]

The CVS network has a hierarchy with two levels: a high-speed network and a low-speed network. A high-speed network covers a wide area which includes many towns and cities, and connects them with about 60-km/h-speedways. It provides macroscopic interurban transportation. A low-speed network densely covers a local area usually included in a city area. It provides microscopic, nearly door-to-door transportation by 40-km/h-speedways. When a vehicle travels from some point P in some city A to another point Q in another city B, it usually travels at first in a low-speed network from origin P to an interchange in city A, then in a high-speed network from city A to city B, and finally again in a low-speed network to the destination Q. Note that the CVS network must have three types of intersections (crossings): those consisting of 1) two high-speed ways, 2) a high-speed and a low-speed way, and 3) two low-speed ways. The first type is used in a high-speed network. The second provides the interface between two levels of the network. These two types are usually grade-separated. The last type is used in a low-speed network and is intended to be grade-crossed. But so far no paper has been published concerning the control and traffic analysis of grade-crossed intersections.

In this paper we present a control algorithm and a theoretical analysis of a grade-crossed intersection. The algorithm is simple but realistic and optimal in the sense that delay is minimized under the following constraints: the first-come first-served rule, the maneuver length limitation, and the collision-freeness condition. The analysis is completely based on this detailed control algorithm, and the exact and explicit expressions for some performance measures are presented.

II. The Model

A. The Intersection and the Vehicles

The configuration of the grade-crossed intersection relevant to this study is illustrated in Fig. 1. Two main lines cross at C. The ramps D1–M2 and D2–M1 enable vehicles to diverge from one main line and merge into another. The checkpoints (CP1 and CP2) and the maneuver zones (Z1 and Z2) are assumed to be in the prediverge zones (upstream of the diverge points).

Since our concern is with the cell-following (or moving target) control, it is assumed that hypothetical cells are moving on the main lines and the ramps with the time intervals between their heads constant at Δt. (The head or downstream boundary of each cell is called a "moving target" or "target.") A target is numbered k and denoted by MT#k, if it will pass (or has passed) a checkpoint at time $k\Delta t$ (k is an integer). Thus Fig. 1 shows a possible situation of the intersection at time $10\Delta t$; the moving targets on CP1 and CP2 are numbered 10. Note that the same-numbered targets meet together at the cross point and the merge points.

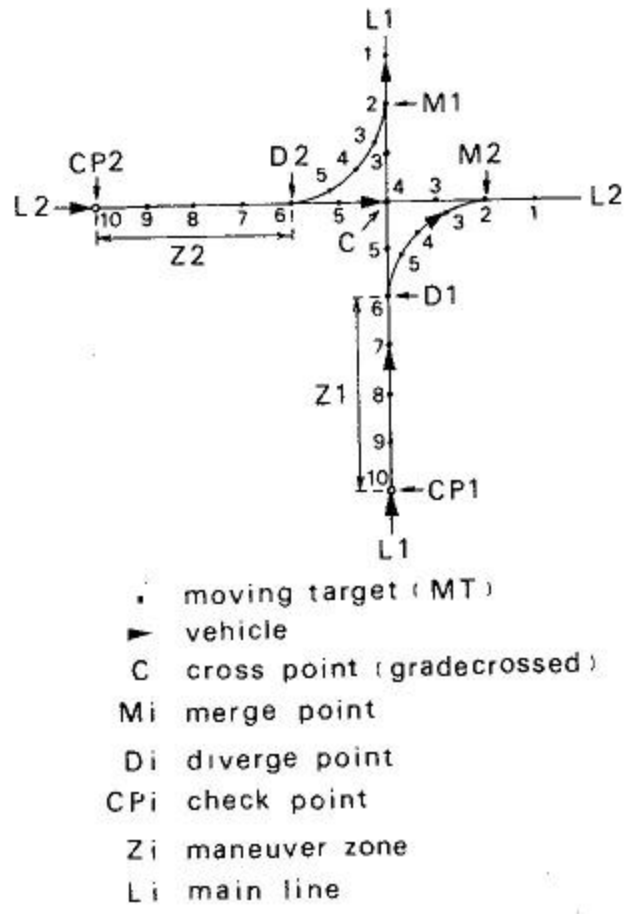

Fig. 1. Schematic representation of a CVS grade-crossed intersection. ●: moving target (MT); ▶: vehicle; C: cross point (gradecrossed); Mi: merge point; Di: diverge point; CPi: checkpoint; Zi: maneuver zone; Li: main line.

Vehicles arriving at the checkpoints are classified into two types: vehicles wishing to go straight (S-type), and vehicles wishing to diverge (D-type). Let p, q, and r be the probabilities associated with the content of each cell that arrives at CP1; p is the probability that the cell contains an S-type vehicle, q is the probability that the cell contains a D-type vehicle, and $r(=1-p-q)$ is the probability that the cell is empty. In the same manner we define p', q', and r' associated with the content of each cell that arrives at CP2.

Except in the maneuver zones, vehicles follow the targets assigned to them. In the maneuver zones, however, vehicles are allowed to maneuver from one cell to another to change the targets they should follow. To illustrate, let us consider the situation in Fig. 1, where vehicles are both on CP1 and CP2. If both vehicles on the check points are S-type, then without maneuvering they would collide together at the cross point. If one is D-type and the other is S-type, then again they would collide together at a merge point. Such a situation is called a "conflict." Maneuvering provides an effective way of resolving conflicts. In the following subsection we will discuss it in detail.

B. Control Algorithm

In this paper we allow vehicles to maneuver forward as well as backward, and assume that maneuvering can occur even when there is no conflict between vehicles. The mechanism based on our assumption is as follows.

[2] The word "grade" in the terms "grade-separated" and "grade-crossed" means the level of each guideway associated with the distance (height) above the ground. A grade-separated intersection is an intersection with two different grades of guideways, including an overpass or underpass. A grade-crossed intersection is an intersection with two equal grades of guideways (thus the guideways intersect.)

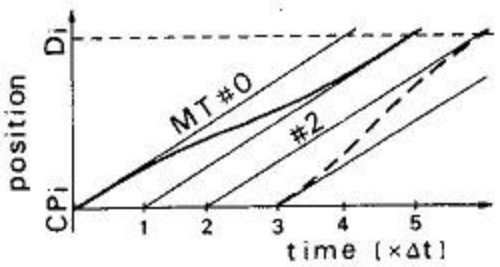

Fig. 2. Maneuver profiles.

When the vehicle on MT#k arrives at the checkpoint at time $k\Delta t$, it receives a command to move to a new cell, say MT#m. (The command is given by the computer which governs the control of the intersection. The real-time algorithm to calculate the value of m is given later.) When the command is received, we say that the vehicle has just been "assigned" MT#m.

The vehicle assigned MT#m attains its new target (i.e., completes maneuvering from MT#k to MT#m) up to the time it reaches the diverge point, by accelerating and/or decelerating in the maneuver zone. Fig. 2 shows typical trajectories for maneuvering vehicles: from MT#0 to MT#1 (the solid line), and from MT#3 to MT#2 (the broken line). The maneuver length is $(m - k)$ cells backward if $k < m$, and $(k - m)$ cells forward if $k > m$. Thus the solid line in Fig. 2 is a profile of maneuvering one cell backward, and the broken line is that of maneuvering one cell forward.

Let f and b be the upper limits of the maneuver length; that is, vehicles are not allowed to maneuver more than f cells forward or more than b cells backward. Thus m must satisfy the inequality

$$k - f \leqslant m \leqslant k + b.$$

The values of f and b depend on the maneuver zone length, the maximum/minimum vehicle speeds, the maximum acceleration, the maximum jerk, etc. (The minimum speed is needed to avoid a traffic jam and ensure non-stop transportation.)

In the remainder of this subsection we describe an algorithm to determine the value of m, i.e., a real-time algorithm that assigns vehicles appropriate targets when they arrive at the checkpoints. Our strategy is to assign each vehicle the most advanced target (i.e., the target with the smallest number) under the following three constraints

1) the "first-come first-served" rule—the target that should be assigned must have a greater number than that of any target assigned earlier,
2) the maneuver length limitation—the maneuver length must not be greater than f forward and b backward, and
3) the collision-freeness condition—a collision at the cross point or at the merge points must be avoided.

To describe the algorithm precisely, let us introduce the variables n, k, and x. The n indicates the most advanced cell under the constraints 1) and 2) above; the arriving vehicle is most likely to be assigned MT#n, except in the conflict case, where the vehicle may be assigned MT#$(n + 1)$, as we will see later. The k is the number of the arriving target (MT#k). The variable k is increased one by one every Δt time unit. The x is the variable to keep the value of $n - k$. If the vehicle on MT#k is assigned MT#n, then it will maneuver x cells backward. (The statement "x cells backward" is interpreted as "$(-x)$ cells forward" if $x < 0$.) We will see later that x satisfies

$$-f \leqslant x \leqslant b \tag{1}$$

and that x is chosen as the state of the stochastic process to be analyzed in this work.

Now let us describe the algorithm of cell assignment. We assume that the initial conditions are $k = 0$, $n = -f$, and $x = -f$. When MT#k arrives at the checkpoint at time $k\Delta t$, the algorithm behaves as in Table I, depending on what type of vehicle (i.e., S-type, D-type, or no vehicle) is on each checkpoint. In Table I

$$\lambda = pp' + pq' + qp' \tag{2}$$

and

$$\mu = rr' \tag{3}$$

are introduced for simplification.

Case 1, Relaxation: If there are no vehicles on the checkpoints, assignment is not performed, so MT#n can be assigned to the next-coming vehicle; MT#n is still the most advanced cell to be assigned. Therefore, the algorithm keeps the value of n as it is and lets $k \leftarrow k + 1$ and $x \leftarrow x - 1$ if $-f < x \leqslant b$.

If $x = -f$, then the above procedure is illegal because x would have the value $-f - 1$, which contradicts (1) or the maneuver length limitation. Therefore, we must give up assigning MT#n to any vehicle; we are obliged to waste MT#n without using it effectively. The next-coming vehicle must be assigned a target numbered $(n + 1)$ or more. Thus if $x = -f$, let $k \leftarrow k + 1$, $n \leftarrow n + 1$, and keep x as it is.

Case 2, Non-conflict: Consider the case in which there is just one vehicle or there are two D-type vehicles on the checkpoints. The vehicle(s) can be assigned MT#n because no collision occurs. Since the most advanced cell to be assigned next should be replaced by MT#$(n + 1)$, let $k \leftarrow k + 1$, $n \leftarrow n + 1$ and keep x as it is.

Case 3, Conflict: In the case where there are two vehicles including at least one S-type vehicle, they will collide together if they are assigned the same-numbered targets. Therefore, we assign one of them MT#n and the other MT#$(n + 1)$. Since the next-coming vehicle should be assigned a target numbered $(n + 2)$ or more, let $k \leftarrow k + 1$, $n \leftarrow n + 2$, and $x \leftarrow x + 1$.

Note that the above procedure is valid only if $-f \leqslant x < b$; if $x = b$, the procedure causes $x \leftarrow b + 1$, which contradicts the maneuver length limitation. Therefore, when $x = b$ we must assign MT#n to both vehicles and force them to diverge (to avoid a collision) even if they wish to go straight. Then, let $k \leftarrow k + 1$, $n \leftarrow n + 1$, and keep x as it is $(x = b)$.

Here we add a comment to case 3. We stated that if $-f \leqslant x < b$, then one of the vehicles is assigned MT#n and the other is assigned MT#$(n + 1)$. However, it is not

TABLE I
INTERSECTION CONTROL ALGORITHM

Vehicles on CP1	Vehicles on CP2	Classification	Probability		Control
O	O	Relaxation	rr'	μ	If $-f<x<=b$, $k\leftarrow k+1$, $n\leftarrow n$, $x\leftarrow x-1$. If $x=-f$, $k\leftarrow k+1$, $n\leftarrow n+1$, $x\leftarrow x$
S	O	Non-conflict	pr'	$1-\lambda-\mu$	Assign the vehicle(s) MT#n, and $k\leftarrow k+1$, $n\leftarrow n+1$, $x\leftarrow x$
D	O		qr'		
O	S		rp'		
O	D		rq'		
D	D		qq'		
S	D	Conflict	pq'	λ	If $-f<=x<b$, assign MT#n to one of the vehicles and MT#(n+1) to the other. Then $k\leftarrow k+1$, $n\leftarrow n+2$, $x\leftarrow x+1$. If $x=b$ then assign MT#n to both vehicles and force them to diverge even if they are wishing to go straight. Then $k\leftarrow k+1$, $n\leftarrow n+1$, $x\leftarrow x$.
D	S		qp'		
S	S		pp'		

S : wishing to go straight
D : wishing to diverge
O : no vehicle exists

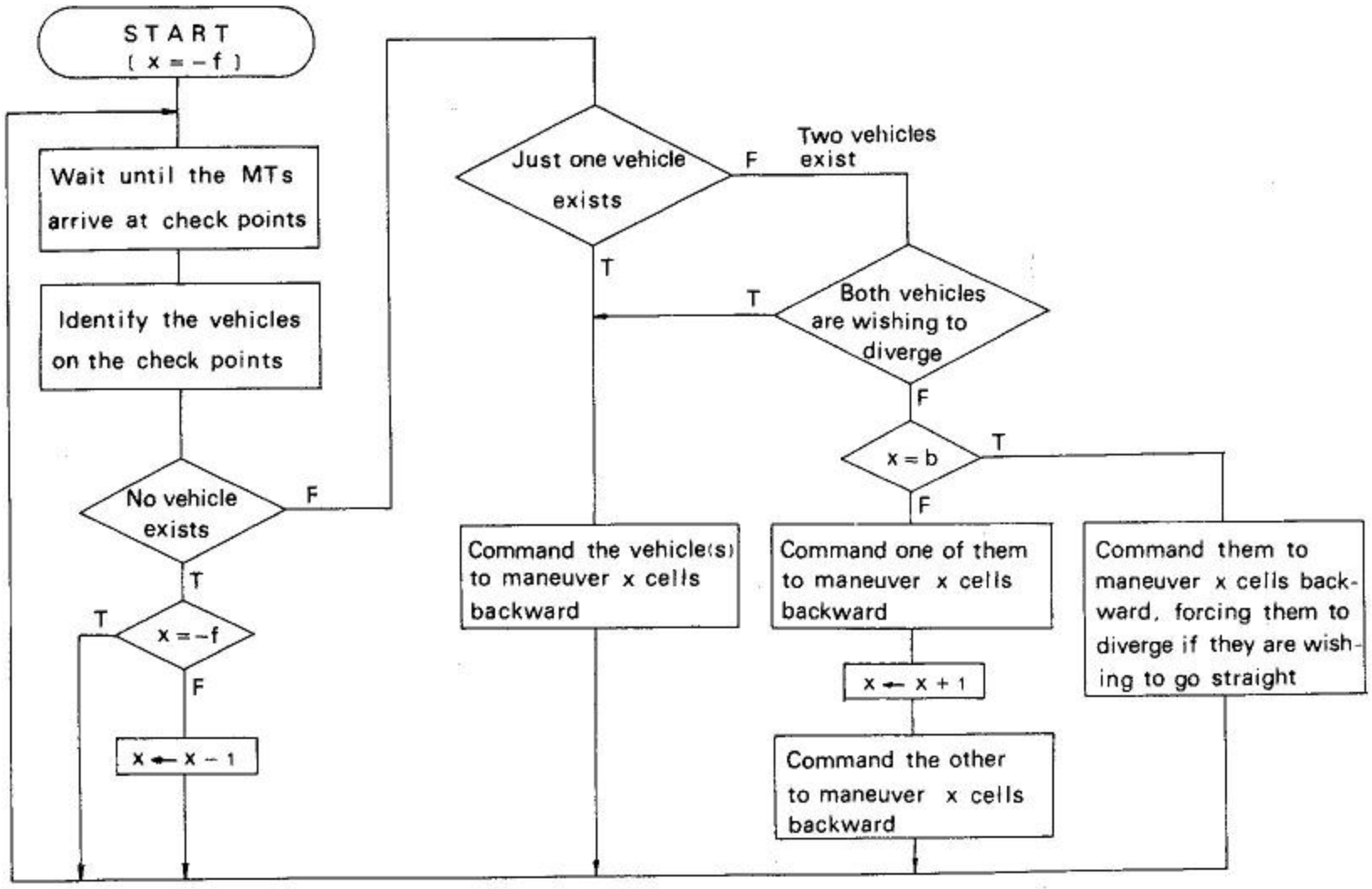

Fig. 3. Control algorithm.

important in this work which target is assigned to each vehicle. For convenience, let it be decided at random with each probability 0.5.

The flowchart in Fig. 3 shows the practical implementation of the algorithm. Note that assigning MT#n implies giving a command to maneuver x cells backward, and that the use of variables n and k is optional and only the one state variable x is needed in the implementation.

C. Illustrative Example

Assume that $f = 1, b = 2$ and that the vehicles arrive as shown in Fig. 4. In the following we illustrate how the control algorithm works.

The initial conditions (the conditions just before time $0\Delta t$) are $k = 0$ and $n = x = -1$. At time $0\Delta t$, an S-type vehicle (denoted by S) and a D-type vehicle (denoted by

input

MT No.	0	1	2	3	4	5	6	7	8	9
main line 1	S	D	D	S	S	O	S	O	O	O
main line 2	D	D	S	S	D	O	O	O	O	S

Fig. 4. Input string of vehicles.

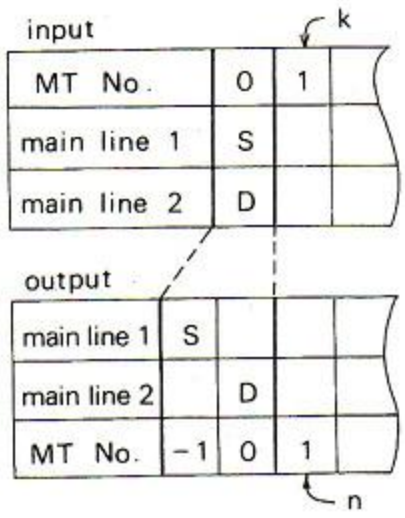

Fig. 5. The situation at time just before $1\Delta t$.

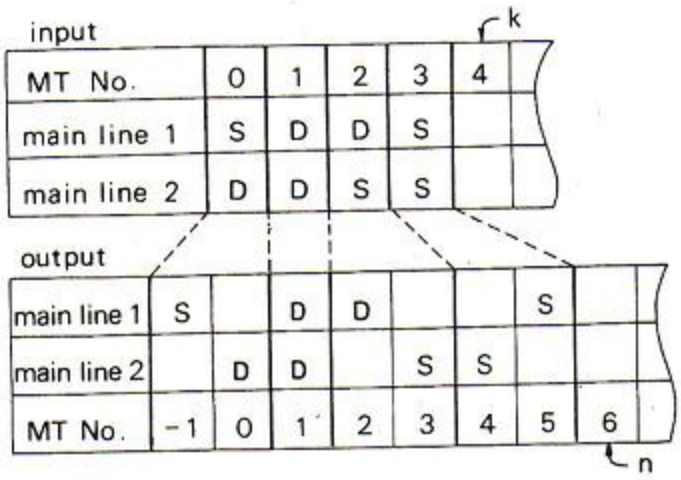

Fig. 6. The situation at time just before $4\Delta t$.

D) arrive at CP1 and CP2, respectively. As a result, S is assigned MT#(−1) and D is assigned MT#0, and the variables k, n, and x keep the values $k = 1$, $n = 1$, and $x = 0$, respectively, until the time just before $1\Delta t$ (Fig. 5).

At time $1\Delta t$, two D-type vehicles arrive, and both are assigned MT#1. At time $2\Delta t$, a D-type vehicle and an S-type vehicle are assigned MT#2 and MT#3, respectively. At time $3\Delta t$, two S-type vehicles are assigned MT#4 and MT#5. At this point, $k = 4$, $n = 6$, and $x = 2$. Figure 6 illustrates this situation.

At time $4\Delta t$, two vehicles, S and D, arrive and the algorithm assigns MT#6 to both of them, forcing vehicle S to diverge (denoted by $\hat{D}$ in Fig. 7). Continuing in this way we can see that the situation at time 10 Δt is $k = 10$, $n = 9$, and $x = 1$, as shown in Fig. 7.

Figure 8 illustrates the trajectories of the vehicles. The solid lines and broken lines represent the trajectories of the vehicles on main line 1 and 2, respectively. The symbol d denotes a D-type vehicle, while the symbol $\hat{d}$ denotes an S-type vehicle forced to diverge in spite of its wish to go straight.

D. Discussion

Let us discuss the validity of the location of maneuver zones. In a big intersection such as a grade-separated one

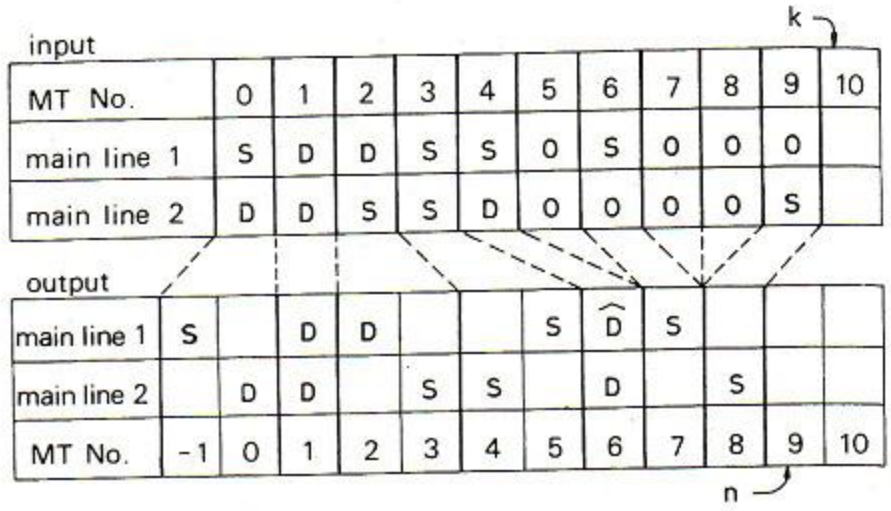

Fig. 7. The situation at time just before $10\Delta t$.

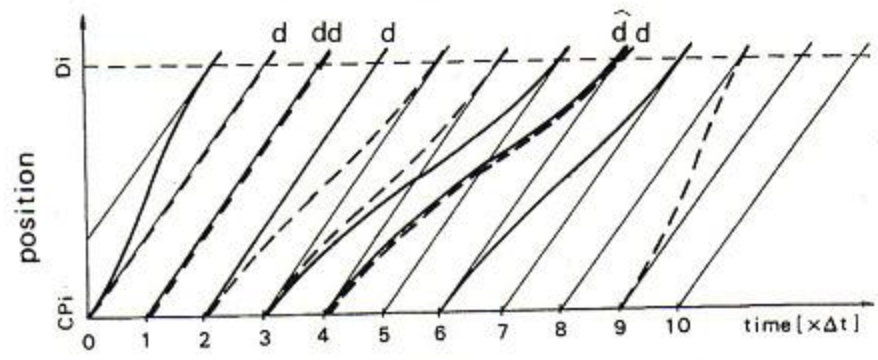

Fig. 8. Trajectories of the vehicles.

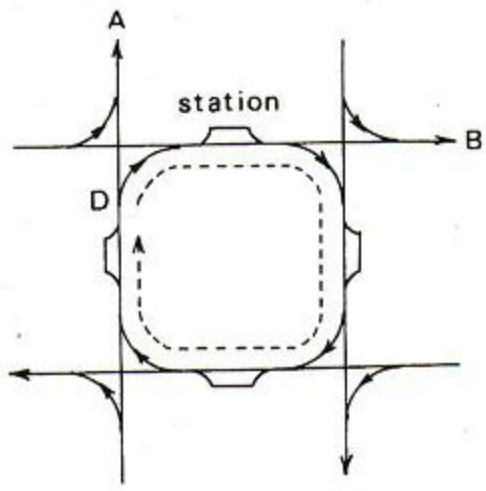

Fig. 9. A part of a CVS network.

in a high-speed network, maneuver zones can be located within the ramp areas (D1–M2, D1–M1, etc.) instead of the prediverge areas (CP1–D1, CP2–D2), and it is reported [21], [22] that the former alternative (ramp maneuvering) can provide better performance. However, our concerns in this work are not with big grade-separated intersections but with small grade-crossed ones which are densely distributed in a low-speed network covering an urban area. In such networks the distances between the diverge points and the merge points (D1–M2, D2–M1), and the distances between the diverge points and the cross point (D1–C, D2–C), are too short to provide enough length for maneuvering. Therefore, it is a valid assumption that the maneuver zones are in the prediverge areas.

Let us discuss an aborting characteristic of our model. In a conventional model of a quasi-synchronous-controlled grade-separated intersection [18], [19] going straight is always allowed, while diverging is sometimes forbidden. On the other hand, in our grade-crossed intersection diverging is always allowed, while going straight is sometimes forbidden. Consider the typical CVS network in Fig. 9. In our model, if a vehicle wishing to go straight toward A is forced to diverge at D, it can return to D by running along the route shown by the broken line. Of course it can select

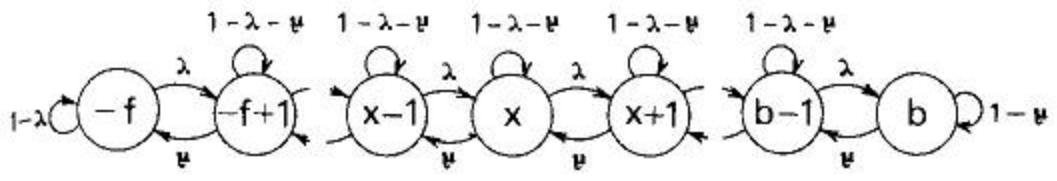

Fig. 10. Transition diagram.

another route instead, but at worst it can return to D at the lowest cost. This is also true when a vehicle cannot enter an off-line station (owing to no empty berths in the station) and is obliged to pass beside the station; it can return near the station without traveling too long.

III. Analysis

A. Markov Chain

The control algorithm discussed in the previous section defines a finite-state automaton if we choose x as its state and the collection of (00), (S0), $\cdots$, and (SS) (shown in Table I) as its input set. If we observe the value of x every Δt time unit, the resultant sequence defines the finite-state discrete-time Markov chain [26] with its transition probabilities (denoted by $p_{i,j}$'s) as follows:

$$p_{x,x-1} = \mu, \qquad -f < x \leqslant b \tag{4}$$

$$p_{x,x} = \begin{cases} 1-\lambda, & x = -f \\ 1-\lambda-\mu, & -f < x < b \\ 1-\mu, & x = b \end{cases} \tag{5}$$

$$p_{x,x+1} = \lambda, \qquad -f \leqslant x < b. \tag{6}$$

This stochastic process is a version of the one-dimensional random walk [26]. The transition diagram is shown in Fig. 10.

B. The Stationary Distribution and the Expected Value of x

Let $\{\pi_i | -f \leqslant i \leqslant b\}$ be the stationary distribution of x; that is, π_i is the probability that $x = i$ in the steady state. It is evident from Fig. 10 that $\{\pi_i\}$ satisfies the following equations [26]:

$$\begin{aligned} \pi_{-f} &= \pi_{-f}(1-\lambda) + \pi_{-f+1}\mu \\ \pi_i &= \pi_{i-1}\lambda + \pi_i(1-\lambda-\mu) + \pi_{i+1}\mu, \qquad -f < i < b \\ \pi_b &= \pi_{b-1}\lambda + \pi_b(1-\mu) \\ \pi_{-f} + \pi_{-f+1} + \cdots + \pi_b &= 1. \end{aligned} \tag{7}$$

These yield

$$\pi_i = \begin{cases} \dfrac{(1-\rho)\rho^{f+i}}{1-\rho^{L+1}}, & \rho \neq 1 \\ 1/(L+1), & \rho = 1, \qquad -f \leqslant i \leqslant b \end{cases} \tag{8}$$

and

$$\bar{x} = \sum_{i=-f}^{b} i\pi_i = \begin{cases} \dfrac{\rho}{1-\rho} \cdot \dfrac{1-(L+1)\rho^L + L\rho^{L+1}}{1-\rho^{L+1}}, & \rho \neq 1 \\ L/2 - f, & \rho = 1 \end{cases} \tag{9}$$

where

$$\rho = \lambda/\mu \text{(the traffic intensity)}, \tag{10}$$

$$L = f + b \text{(the degree of maneuverability)}, \tag{11}$$

and $\bar{x}$ is the expected value of x.

The $\{\pi_i\}$ is a shifted geometric distribution with a finite domain; ρ and L define its form, and f defines the shift. ρ and L also define

$$\pi_b = \begin{cases} \dfrac{(1-\rho)\rho^L}{1-\rho^{L+1}}, & \rho \neq 1 \\ 1/(L+1), & \rho = 1 \end{cases} \tag{12}$$

which is an important factor of the abort rate (described later).

C. Delay

If a vehicle maneuvers δ cells backward, then the time consumed by the vehicle in the maneuver zone will be $\delta\Delta t$ longer than that in the nonmaneuver case (i.e., the case where the vehicle moves without maneuvering). In other words, the maneuver length ($= \delta$) is equivalent to the time delay measured by Δt. (If $\delta < 0$, we interpret it as the maneuvering of $(-\delta)$ cells forward and $(-\delta)$ as the time gain.)

Though we have already obtained the distribution of x, it is not equivalent to the distribution of the delay; if $x = i$, then the delay is either $i\Delta t$ or $(i+1)\Delta t$. In the following we calculate the distribution of the delay based on the distribution of x.

First, let us calculate $\sigma_i = \Pr[\delta_S = i]$, where δ_S is the random variable which represents the delay of an S-type vehicle on main line 1, and σ_i is its stationary distribution. The S-type vehicle is delayed by $i\Delta t$ in the following three situations, given that the vehicle is on CP1:

a) the case where $x = i$ and no conflict arises;
b) the case where $x = i$ and the conflict arises, but this vehicle is assigned MT#i rather than MT#$(i+1)$; and
c) the case where $x = i-1$ and the conflict arises, and this vehicle is assigned MT#i rather than MT#$(i-1)$.

Since in the conflict cases b) and c) this vehicle is assigned MT#i with probability 0.5 (owing to the assumption in II-B), we obtain

$$\sigma_i = \left[(p'+q')/2\right]\pi_{i-1} + \left[(p'+q')/2 + r'\right]\pi_i.$$

However, this discussion is valid only if $-f < i < b$. If $i = -f$, we should omit case c). If $i = b$, we should omit cases a) and b), and add a new case:

d) the case in which $x = b$.

Thus we obtain the exact expression for σ_i

$$\sigma_i = \begin{cases} [1-(p'+q')/2]\pi_{-f}, & i=-f \\ [(p'+q')/2]\pi_{i-1}+[1-(p'+q')/2]\pi_i, & -f<i<b \\ [(p'+q')/2]\pi_{b-1}+\pi_b, & i=b. \end{cases} \tag{13}$$

Note that $1-(p'+q')/2=(p'+q')/2+r'$.

From (13) we can calculate the expected delay of an S-type vehicle on main line 1:

$$\bar{\delta}_S = \sum_{i=-f}^{b} i\sigma_i = \bar{x}+[(p'+q')/2](1-\pi_b). \tag{14}$$

In a similar way we obtain

$$\tau_i = \begin{cases} (1-p'/2)\pi_{-f}, & i=-f \\ (p'/2)\pi_{i-1}+(1-p'/2)\pi_i, & -f<i<b \\ (p'/2)\pi_{b-1}+\pi_b, & i=b \end{cases} \tag{15}$$

and

$$\bar{\delta}_D = \bar{x}+(p'/2)(1-\pi_b) \tag{16}$$

where τ_i is the distribution of the delay of a D-type vehicle on main line 1, and $\bar{\delta}_D$ is the expected delay of it. Furthermore, we can obtain the quantities associated with the delay on main line 2

σ_i' = the distribution of the delay of an S-type vehicle on main line 2,
$\bar{\delta}_S'$ = the expected delay of it,
τ_i' = the distribution of the delay of a D-type vehicle on main line 2, and
$\bar{\delta}_D'$ = the expected delay of it

though the results are not shown here. It is easily seen that the expected delays satisfy

$$\bar{x} \leqslant \bar{\delta}_S, \bar{\delta}_D, \bar{\delta}_S', \bar{\delta}_D' \leqslant \bar{x}+0.5. \tag{17}$$

This means that the difference between these delays is at most 0.5 (Δt). Therefore, in many practical situations we need not evaluate all of them. The average delay defined here is sufficient

$$\bar{\delta} = \frac{p\bar{\delta}_S+q\bar{\delta}_D+p'\bar{\delta}_S'+q'\bar{\delta}_D'}{p+q+p'+q'} = \bar{x}+\frac{\lambda(1-\pi_b)}{p+q+p'+q'} \tag{18}$$

D. Abort Rate

Let ϵ be the abort rate on main line 1, i.e., the probability that an S-type vehicle on main line 1 is forced to diverge. Then we have

$$\begin{aligned} \epsilon &= \Pr[x=b, \text{ there is a vehicle on CP2} \mid \text{an } S\text{-type vehicle is on CP1}] \\ &= \pi_b(p'+q')p/p \\ &= \pi_b(p'+q'). \end{aligned} \tag{19}$$

Similarly we can calculate ϵ' or the abort rate on main line 2

$$\epsilon' = \pi_b(p+q). \tag{20}$$

E. Throughput

Let θ and θ' be the throughputs on main line 1 and 2, respectively, i.e., the expected numbers of vehicles per Δt time unit that will pass CP1 and CP2, respectively, without abortion. Then we have

$$\theta = (1-\epsilon)p+q \tag{21}$$

and

$$\theta' = (1-\epsilon')p'+q'. \tag{22}$$

F. Unrestricted Case

Up to this point we have assumed implicitly that $b<\infty$. In this subsection we assume that $b=\infty$, or that the backward maneuver length is not limited.

When $b=\infty$ the stationary distribution of x does not necessarily exist. The necessary and sufficient condition for its existence [26] is $\lambda<\mu$. In other words, the expected value of x is finite if and only if the following inequality holds:

$$p+q+p'+q'-qq'<1. \tag{23}$$

If (23) holds, we can obtain several measures:

$$\pi_i = (1-\rho)\rho^{f+i}, \quad i \geqslant -f \tag{24}$$

$$\bar{x} = \rho/(1-\rho)-f \tag{25}$$

$$\bar{\delta} = \bar{x}+\lambda/(p+q+p'+q') \tag{26}$$

$$\epsilon = \epsilon' = 0 \tag{27}$$

$$\theta = p+q, \quad \theta' = p'+q'. \tag{28}$$

G. Numerical Examples

Let κ and κ' be the inflow cell occupancies on main lines 1 and 2, respectively, and let β and β' be the diverge-seeker proportions on main lines 1 and 2, respectively. Then

$$\kappa = p+q, \quad \kappa' = p'+q' \tag{29}$$

and

$$\beta = q/(p+q), \quad \beta' = q'/(p'+q'). \tag{30}$$

In the following numerical examples we adopt κ, κ', β, and β' (rather than p, q, r, p', q', and r') as the parameters of the model by substituting

$$\begin{aligned} p &= (1-\beta)\kappa, & q &= \beta\kappa, & r &= 1-\kappa \\ p' &= (1-\beta')\kappa', & q' &= \beta'\kappa', & r' &= 1-\kappa' \end{aligned} \tag{31}$$

into (2), (3), and (18)–(22).

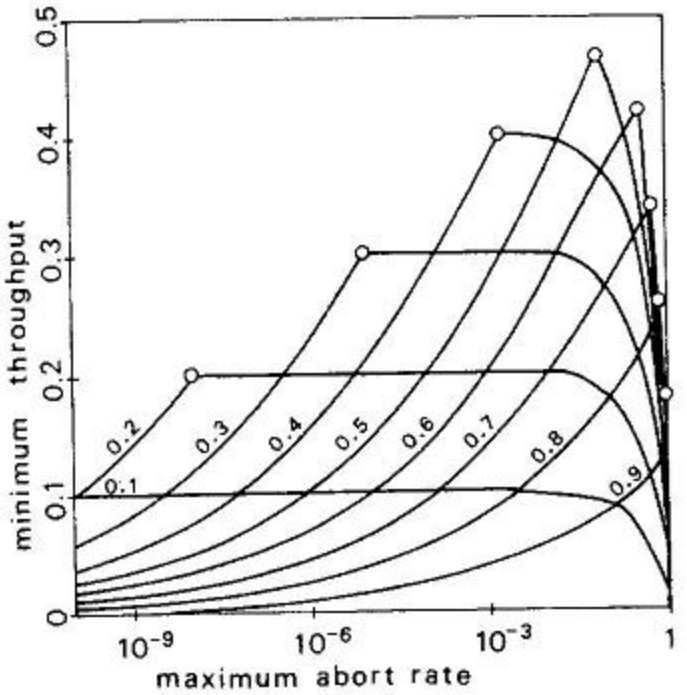

Fig. 11. Minimum throughput versus maximum abort rate (1).

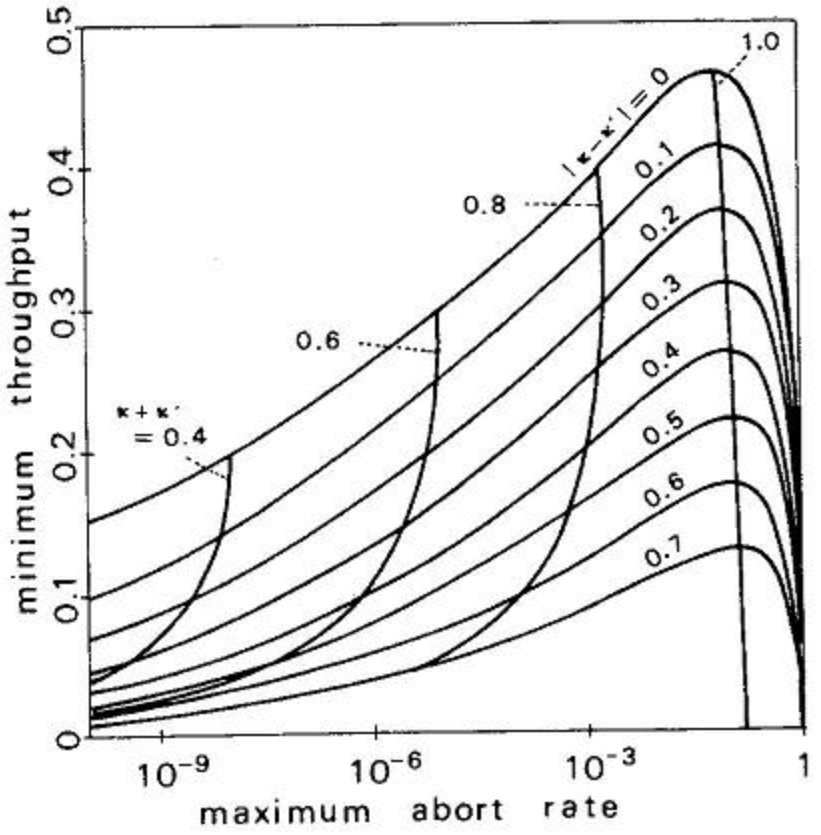

Fig. 12. Minimum throughput versus maximum abort rate (2).

Figs. 11 and 12 show the variation of the minimum throughput ($= \min[\theta, \theta']$) and the maximum abort rate ($= \max[\epsilon, \epsilon']$) with κ and κ'. It is assumed that $L = 6 (f = 0, b = 6)$ and $\beta = \beta' = 0.1$. In Fig. 11 the lines are obtained by changing the value of one κ or κ' while fixing the value of the other; the fixed values are shown on the sides of the related lines. For example, if $\kappa = 0.3$ and $\kappa' = 0.5$, then the minimum throughput and the maximum abort rate are 0.3 and 10^{-3}, respectively. The small circles in the figure indicate the points at which $\kappa = \kappa'$. We see from the figure that if κ is increased while keeping κ' fixed, then the maximum abort rate increases. On the other hand, the minimum throughput increases just until $\kappa = \kappa'$, then it keeps the constant value and finally decreases.

In Fig. 12 the lines are obtained by changing the values of κ and κ' such that their sum or their difference is constant. We see that the maximum abort rate depends mainly on $\kappa + \kappa'$ if the difference between κ and κ' is not so great.

Fig. 13 shows the variation of the average throughput ($= [\theta + \theta']/2$) and the average abort rate ($= [p\epsilon + p'\epsilon']/[p + p']$) in the same way as in Fig. 12. If $\kappa + \kappa'$ is

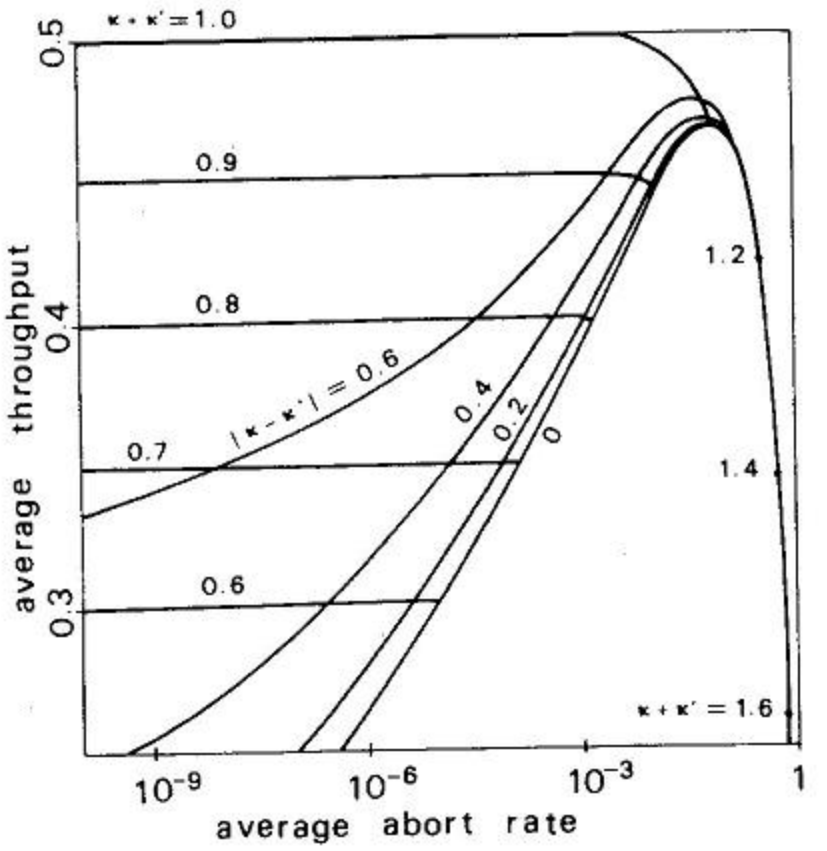

Fig. 13. Average throughput versus average abort rate.

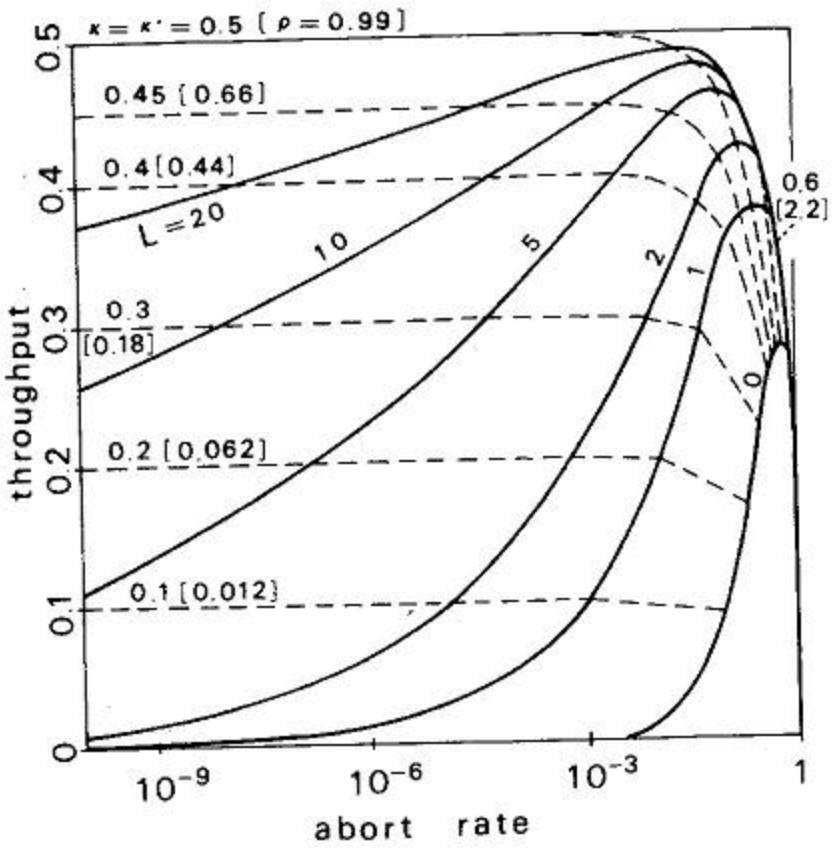

Fig. 14. Throughput versus abort rate.

greater than approximately 1.2, then the average throughput and the average abort rate do not depend on the difference between κ and κ'. We see from Fig. 12 that the smaller difference between κ and κ' makes the minimum throughput higher and the maximum abort rate lower; on the other hand, from Fig. 13, the smaller difference makes the average throughput lower and the average abort rate higher. Therefore, we should say that load-balance (the smaller difference between κ and κ') contributes to the improvement of the "worst" performances (the "minimum" throughput and the "maximum" abort rate) at the cost of the improvement of the "average" performances (the "average" throughput and the "average" abort rate).

Fig. 14 shows the variation of the throughput $\theta (= \theta')$ and the abort rate $\epsilon (= \epsilon')$ with κ and L. It is assumed that $\kappa' = \kappa$, $\beta = \beta' = 0.1$, and $f = 0$. Figure 15 shows the variation of the average delay $\bar{\delta}$ and the abort rate $\epsilon (= \epsilon')$ in the same way as Fig. 14. With an increase in the degree

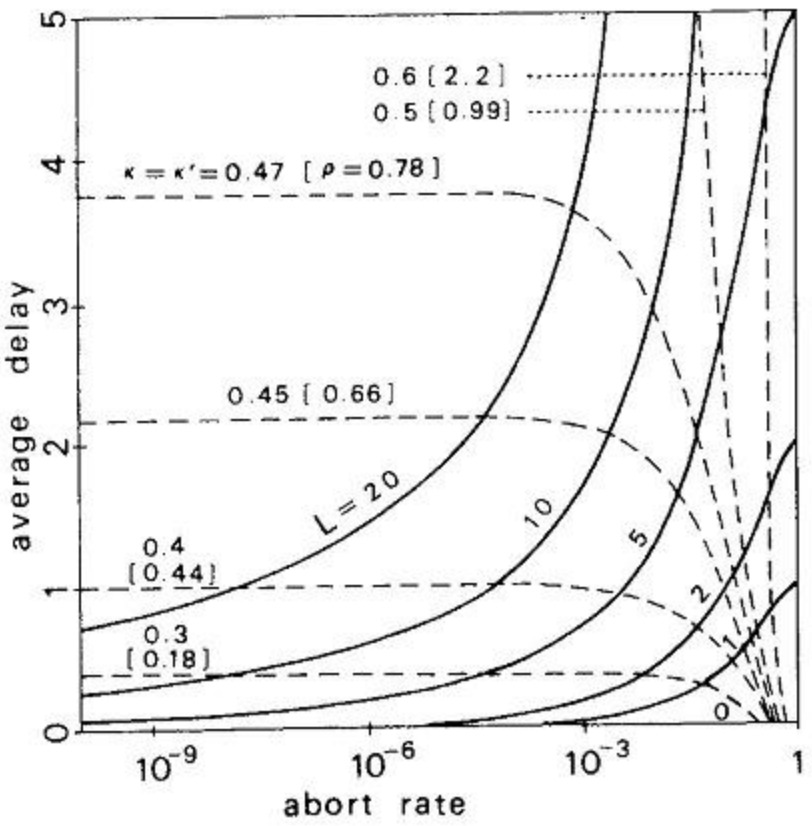

Fig. 15. Average delay versus abort rate.

TABLE II
UPPER LIMIT OF THE INFLOW CELL OCCUPANCY AND THE AVERAGE DELAY

	ε <= 1%		ε <= 0.1%	
L	κ(=θ)	$\bar{\delta}$	κ(=θ)	$\bar{\delta}$
0	0.01	0	0.001	0
1	0.20	0.15	0.10	0.01
2	0.30	0.35	0.20	0.15
5	0.42	1.2	0.35	0.70
10	0.47	3.0	0.43	1.48

of maneuverability ($= L$) the abort rate and the throughput improve at the cost of the increase in the delay.

Table II shows the upper limit of the inflow cell occupancy and the average delay, where L and the upper limit of ϵ are chosen as the parameters; the other conditions are kept the same as in Figs. 14 and 15. For example, when $L = 5$ we see that if at most one vehicle arrives in about $3\Delta t$ time units (i.e., $\kappa \leqslant 0.35$) then the abort rate is less than 0.1 percent.

IV. SUMMARY

We have presented the model and control algorithm of CVS "grade-crossed" intersections. Notable features of them are summarized as follows:

1) simplicity—the algorithm is quite simple, so it can be implemented at low cost by microprocessor or hardware;
2) optimality—the algorithm is optimal in the sense that it assigns each vehicle the most advanced target (thus delay is minimized) subject to the following three constraints: the first-come first-served rule, the maneuver length limitation, and the collision-freeness condition; and
3) analyzability—assuming random arrival, we have been able to formulate the behavior of the algorithm as a simple Markov chain and obtain the exact and explicit expressions for delay, abort rate, and throughput.

REFERENCES

[1] U.S. Congress, OTA, *Automated Guideway Transit*. June 1975.
[2] DOT, UMTA, Request for proposal, Rep. No. DOT-UT-60055, April 1976.
[3] TRW Systems Group, "Proposal for a study of synchronous longitudinal guidance as applied to intercity automated highway network," TRW Proposal 06818.007, prepared by R. K. Boyd *et al.*, submitted to the Office of High-Speed Ground Transportation, U.S. Dep. Trans., June 1968.
[4] W. W. Siefert, "Transportation—1993," *IEEE Spectrum*, vol. 5, pp. 53–57, 1968.
[5] Y. Shirai and Y. Istuhara, "Teito rapid transit authority's automatic train operation," *Proc. IEEE*, vol. 56, pp. 605–615, 1968.
[6] T. E. Parkinson, "Automation in urban transport," *Electronics and Power*, vol. 14, pp. 392–396, 1968.
[7] R. E. Fenton *et al.*, "The electronic highway," *IEEE Spectrum*, vol. 6, pp. 60–66, 1969.
[8] U.S. Department of Housing and Urban Development, *Tomorrow's Transportation*. Washington, D.C., 1968.
[9] Japan Society for the Promotion of Machine Industry, CVS Technical Report, Sept. 1977.
[10] T. Ishii *et al.*, "The control system of CVS using the two-target tracking scheme," in *Proc. Int. Conf. Personal Rapid Transit*, May 1977.
[11] _____, "Computer-controlled minicar system in Expo '70: An experiment in a new personal urban transportation system," *IEEE Trans. Veh. Technol.*, vol. VT-21, pp. 77–91, Aug. 1972.
[12] _____, "On the control system of CVS," in *Proc. 9th Symp. Use of Cybernetics on the Railway*, Nov. 1972, pp. 540–544.
[13] K. Nakada *et al.*, "An analysis of overflows of passengers in a model of CVS station with Markov renewal theory," *Electron. Commun. in Japan*, vol. 63-A, no. 10, pp. 52–62, Oct. 1980.
[14] M. Kurihara *et al.*, "An analysis of a CVS multiberth station model via Markov renewal theory," *Electron. Commun. in Japan*, vol. 64-A, no. 9, pp. 11–20, Sept. 1981.
[15] D. F. Wilkie, "A moving cell control scheme for automated transit systems," *Transport. Sci.*, vol. 4, Nov. 1970.
[16] R. K. Boyd *et al.*, "How to run an automated transportation system," *IEEE Trans. Syst., Man., Cybern.*, vol. SMC-2, no. 3, June 1972.
[17] A. F. Rumsey *et al.*, "Digital-computer control of vehicles in an automated transportation system," *Proc. IEE*, vol. 120, no. 10, pp. 1267–1272, Oct. 1973.
[18] A. V. Munson *et al.*, "Quasi-synchronous control of high-capacity PRT networks," Nat. Conf. Personal Rapid Transit, 1971.
[19] F. Rubin, "Routing algorithm for urban rapid transit," *Transp. Res.*, vol. 9, pp. 215–223, 1975.
[20] T. Dooley and A. S. Priver, "Computer models for AGT system operation studies," *IEEE Trans. Veh. Technol.*, vol. VT-28, no. 1, Feb. 1979.
[21] F. J. McGinley, "An intersection control strategy for a short-headway P.R.T. network," *Transp. Plan. & Technol.*, vol. 3, pp. 45–53, 1975.
[22] _____, "A survey of quasi-synchronous P.R.T. interchange control algorithm," *Transp. Plan. & Technol.*, vol. 3, pp. 233–245, 1977.
[23] _____, "A new algorithm for the control of a PRT intersection," Dep. Civil Eng., Monash Univ., Res. Rep., no. 1, Feb. 1975.
[24] K. Thangavelu *et al.*, "Design analysis of stations and intersections of high-capacity personal rapid transit network," *Transp. Res. Rec.*, no. 522, pp. 10–27, 1974.
[25] Y. Hamamatsu *et al.*, "Modeling and analysis of an intersection on AGT," *Electron. Commun. in Japan*, vol. 64-A, no. 10, pp. 18–27, 1981.
[26] S. Karlin, *A First Course in Stochastic Processes*. New York, London: Academic, 1971.