

MICHEL LEGLISE

**Étude et réalisation d'un modèle de simulation  
du trafic urbain**

*Revue française d'informatique et de recherche opérationnelle. Série verte*, tome 5, n° V1 (1971), p. 61-83

[http://www.numdam.org/item?id=RO\\_1971\\_\\_5\\_1\\_61\\_0](http://www.numdam.org/item?id=RO_1971__5_1_61_0)

© AFCET, 1971, tous droits réservés.

L'accès aux archives de la revue « Revue française d'informatique et de recherche opérationnelle. Série verte » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme  
Numérisation de documents anciens mathématiques  
<http://www.numdam.org/>

## ETUDE ET REALISATION D'UN MODELE DE SIMULATION DU TRAFIC URBAIN

par M. Michel LEGLISE <sup>(1)</sup>

---

*Résumé. — Le modèle de simulation du trafic urbain présenté dans cet article est un modèle numérique microscopique (simulation véhicule par véhicule, tous les véhicules étant identifiés). Cet outil doit permettre de dégager certains algorithmes de commande susceptibles d'améliorer les conditions de circulation dans un réseau.*

*Il est présenté ici l'étude générale du modèle, c'est-à-dire la manière dont sont représentés la voie et les véhicules.*

*Puis on dégage les lois régissant le comportement des véhicules en section courante (loi de poursuite) aux carrefours, et sur le réseau lui-même (détermination des trajets).*

*On traite ensuite de la génération aux entrées, de la réalisation pratique du modèle et de son organisation générale.*

*Les résultats obtenus sur un réseau particulier sont alors présentés.*

*La conclusion dégage enfin les travaux en cours sur la validation et l'optimisation du modèle.*

### I. INTRODUCTION

Il paraît, à l'heure actuelle, pratiquement impossible d'effectuer une approche mathématique du phénomène trafic.

Le recours à la simulation semble alors un bon moyen de continuer les recherches théoriques et pratiques, à condition évidemment que cette simulation soit validée.

De plus, cet outil permet en particulier :

- une approche pratique de la « solution » par optimisation,
- un traitement du problème complexe du réseau,
- des changements de réseau à volonté par simple changement de données.

Nous avons choisi de faire une simulation numérique microscopique (véhicule par véhicule) pour mieux comprendre les phénomènes locaux du point de vue théorique et pour permettre une validation par tranches.

Le but initial est de pouvoir simuler un petit réseau de 5 à 6 carrefours et de dégager certains algorithmes de commande susceptibles d'améliorer les conditions de circulation dans ce réseau, suivant différents critères d'optimisation.

---

(1) Laboratoire d'informatique de la Faculté des Sciences de Toulouse.

Il ne sera présenté ici que la première partie de ce travail (étude et réalisation du modèle de simulation), les travaux de validation et d'optimisation étant en cours et non encore achevés.

## 2. ETUDE DU MODELE

Nous disposons d'un ordinateur IBM 7044 de 32 K mots de 36 bits de mémoire centrale, et de temps de cycle  $2,5 \mu s$ . Il était donc nécessaire d'avoir un modèle assez rapide pour ne pas avoir des temps d'exécution prohibitifs, et prenant assez peu de place en mémoire.

Nous avons adopté un modèle de représentation des véhicules par mémoires (2 mémoires sont nécessaires à la représentation complète d'un véhicule). Quant à la voie, elle est représentée par un ensemble de listes. Chaque véhicule (2 mémoires) représente un élément de la liste.

Nous allons expliciter plus en détail les représentations de la voie et des véhicules.

### 2.1. Représentation de la voie

Le réseau à simuler est divisé en tronçons. Un tronçon peut être défini comme une file à sens unique terminée par un feu.

La figure suivante montre la disposition des tronçons sur un exemple précis.

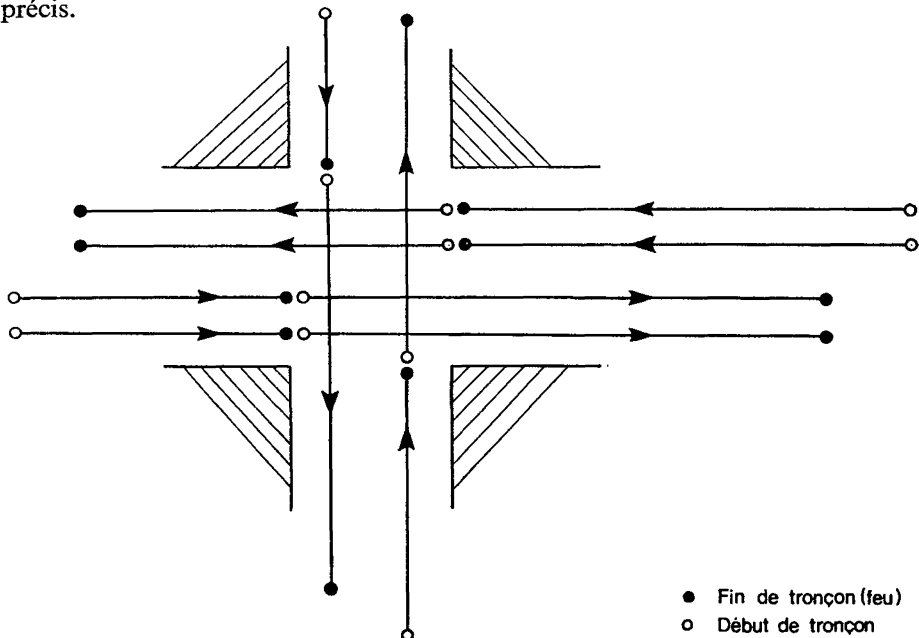


Figure 1

Douze tronçons sont nécessaires pour cet exemple. Un feu se trouve donc obligatoirement à la fin d'un tronçon.

Le début d'un tronçon est, soit une sortie de feu, soit une entrée de réseau.

Physiquement, en machine, un tronçon est représenté par une liste de cellules (une cellule est composée de deux mémoires); dans chaque cellule un véhicule peut prendre place.

## 2.2. Représentation des véhicules

Chaque véhicule est représenté par deux mots mémoire pouvant entrer dans une cellule de la liste tronçon.

### 2.2.1. Représentation interne d'une cellule

Nous avons besoin de 10 paramètres pour représenter un véhicule. Il a donc été nécessaire de découper les deux mémoires de la cellule en zones, chaque zone représentant un de ces paramètres.

La première mémoire contient des paramètres généraux du véhicule, la deuxième est découpée en zones représentant les données de l'avancement du véhicule.

#### — PREMIÈRE MÉMOIRE

- Zone 1. DE : destination, c'est-à-dire n° de sortie du véhicule.
- Zone 2. VD : vitesse désirée du véhicule (en m/s).
- Zone 3. HI : heure d'introduction du véhicule dans le réseau.
- Zone 4. IN : n° de l'entrée où est apparu le véhicule.
- Zone 5. VM
- Zone 6. TO } index de servitude.
- Zone 7. AR : comptabilité du nombre d'arrêts.

#### — DEUXIÈME MÉMOIRE

- Zone 1. PO : position du véhicule dans le tronçon par rapport à l'origine (en dm).
- Zone 2. VR : vitesse réelle du véhicule (en cm/s).
- Zone 3. DS : dernier avancement effectué par le véhicule (en dm).

L'implantation effective dans les deux mémoires se résume sur la figure suivante :

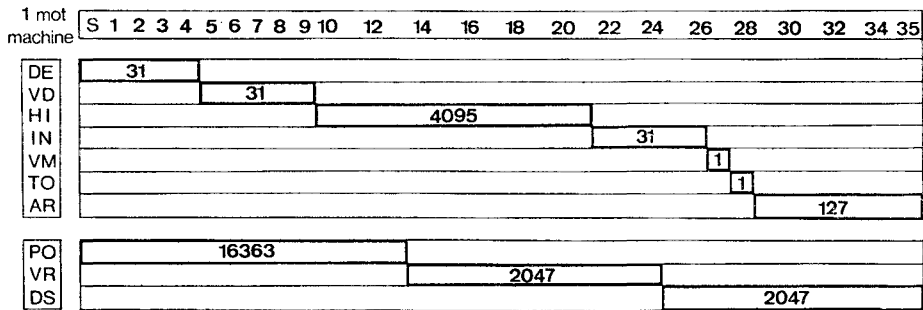


Figure 2

Les chiffres indiqués dans les différentes zones indiquent le nombre maximum que l'on peut introduire dans chaque zone. L'écriture ou la lecture dans les différentes zones se fait à l'aide d'une macro-instruction écrite en MAP, et qui exécute tous les contrôles de non-débordement nécessaires.

### 2.2.2. Représentation générale de la mémoire tronçon-véhicule

Pour des raisons de place, nous avons limité le nombre de segments à 80, et le nombre de cellules par segment à 70.

Nous avons donc besoin de 11 200 mémoires pouvant contenir 5 600 véhicules.

Le réseau est donc représenté par un tableau à 3 dimensions fictives :  $i$ ,  $j$  et  $k$ .

$i$  représente le n° du tronçon.

$j$  représente le n° de la cellule dans le tronçon.

$k$  représente le n° du mot dans la cellule.

### 2.2.3. Repérage d'un véhicule

Il faut pouvoir repérer un véhicule à l'intérieur d'un tronçon donné.

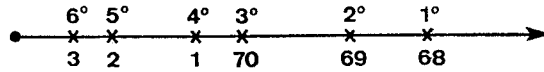
Pour cela, à chaque tronçon, sont associés deux paramètres :

- le nombre de véhicules dans le tronçon,
- l'indice du premier véhicule dans le tronçon.

Ces deux valeurs tournent modulo 70. Ainsi l'indice du 1<sup>er</sup> véhicule n'est

pas forcément inférieur à celui du dernier (tableaux circulaires). Voici un exemple de configuration possible :

ordre des véhicules



indice des véhicules dans le tronçon.

indice du 1<sup>er</sup> = 68                      nombre = 6.

Dans ce cas, sur les 70 cellules du tronçon, 6 seulement sont utilisées.

## 2.2.4. Organisation de l'avancement des véhicules

La position du véhicule dans la voie par rapport à l'origine est elle-même incluse dans l'élément véhicule de la liste. Ainsi, lorsque le véhicule se déplace dans le tronçon, son adresse physique en machine ne change pas, seule la zone « position » de sa mémoire est modifiée.

Il ne faut effectivement transférer le véhicule à une autre adresse que lorsqu'il change de tronçon.

On peut donc dire, qu'au lieu de déplacer les véhicules sur la voie, on fait glisser la voie, en la déformant, sous les véhicules, ce qui entraîne un gain considérable en temps d'exécution.

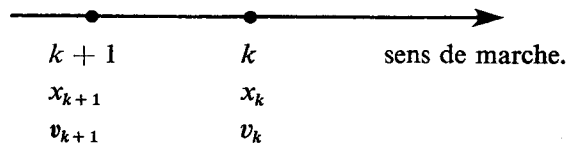
## 2.3. Comportement des véhicules

Ils obéissent évidemment aux règles élémentaires du code de la route, et ne peuvent, dans l'état actuel, se dépasser.

### 2.3.1. Comportement en section courante

Les véhicules obéissent, quelle que soit leur position dans le réseau (section courante ou carrefours) à une loi de poursuite élaborée par MM. G. Authie et M. Léglise (réf. 13). Nous ne reviendrons pas sur le détail de cette loi, mais nous rappellerons certains points nécessaires à la compréhension de cet exposé.

#### 2.3.1.1. Loi de Poursuite



Le mouvement du véhicule suiveur est basé sur une équation différentielle aux différences qui contient trois paramètres ayant une signification physique. Dans le modèle envisagé, est introduit un terme qui ne joue un rôle qu'en

n° V-1, 1971.

régime transitoire et qui apporte une information sur la nature de ce régime (accélération ou décélération). En effet, la vitesse du véhicule suiveur est exprimée, non en fonction de la distance réelle du véhicule qui le précède, mais d'une distance fictive, qui est la distance réelle modifiée par un terme correctif. De plus, l'accélération et la décélération des véhicules sont limitées.

Par ailleurs, en régime permanent, la loi est telle que la relation débit-concentration soit vérifiée.

On utilise les notations suivantes :

$h_0$  : espacement pare choc Av à pare choc Av des véhicules à l'arrêt.

$k$  : rang du véhicule.

$t$  : temps.

$x_k$  : position du véhicule  $k$ .

$v_k$  : vitesse du véhicule  $k$ .

$\gamma_k$  : accélération du véhicule  $k$ .

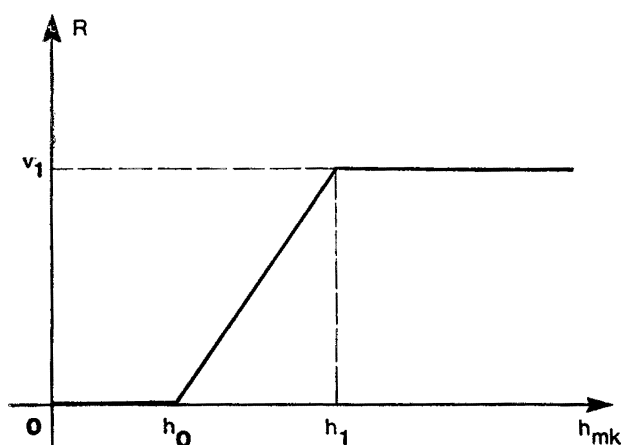
$\Delta$  : temps de réaction du système conducteur-véhicule.

$v_1$  : vitesse maximum des véhicules.

La loi est la suivante :

$$\begin{cases} v_{k+1}(t) = R[h_{mk}(t - \Delta)] \\ h_{mk}(t) = x_k(t) - x_{k+1}(t) + \alpha(v_k(t) - v_{k+1}(t)) \\ \gamma_k(t) \in [a, b] \end{cases}$$

La fonction  $R$  est la suivante :



$$h_0 \leq h_{mk} \leq h_1 \quad R = \frac{v_1}{h_1 - h_0} [h_{mk} - h_0]$$

$$h_1 < h_{mk} \quad R = v_1$$

$$h_0 > h_{mk} \quad R = 0$$

Figure 3

$T$  a les dimensions d'un temps, et représente l'espacement temporel entre deux véhicules en régime permanent, quelle que soit la vitesse.

$\alpha$  a les dimensions d'un temps. Ce coefficient entraîne une hystérésis du système.

Les détails de l'intégration sont consignés dans la référence [12].

#### 2.3.1.2. *Détermination du véhicule que l'on doit suivre*

La loi de poursuite est applicable à tous les cas, mais il est nécessaire de pouvoir déterminer le véhicule que l'on suit.

Si l'on se trouve dans un peloton, c'est évidemment le véhicule précédent.

Mais il faut examiner la question de plus près pour le premier véhicule d'un tronçon.

Si le feu est vert, le premier véhicule doit chercher quel est le prochain segment dans lequel il va, et suit le véhicule immédiatement devant lui dans ce segment.

Si le feu est rouge, on place un véhicule fictif arrêté au droit du feu, et le véhicule survenu s'arrêtera ainsi au feu.

### 2.3.2. **Comportement des véhicules aux carrefours**

#### 2.3.2.1. *Structure de l'intersection*

Nous avons déjà dit que chaque segment était terminé par un feu. Pour que toutes les voies existantes soient réellement représentées au carrefour, les segments débutent donc à la suite du feu précédent.

La figure 1 montre l'exemple d'une intersection.

#### 2.3.2.2. *Réactions aux feux*

C'est toujours le premier véhicule d'un tronçon qui se trouve confronté avec le feu :

- feu vert : voie libre,
- feu rouge : celui-ci est considéré comme un véhicule fictif arrêté,
- feu orange : on calcule combien il reste de secondes avant le rouge. On examine alors si le véhicule peut passer dans le temps qui lui reste avant le rouge. Sinon, il s'arrête, à moins que sa distance minimum d'arrêt (en freinant au maximum) soit trop grande auquel cas il « brûlera » le feu rouge.



### 2.3.2.3. *Changements de direction*

Un certain nombre de variables sont nécessaires pour définir les changements de direction :

$i$  variable de destination qui indique ce que va faire le véhicule en sortant du tronçon

$i = 1$  il ira tout droit,

$i = 2$  il tournera à droite,

$i = 3$  il tournera à gauche;

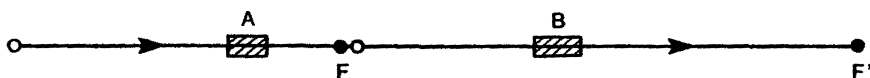
$j$  variable de position qui indique ce que fait le véhicule à cet instant précis :

$j = 1$  il est en zone de tourne à gauche,

$j = 0$  partout ailleurs.

### 2.3.2.4. *Manœuvres « va tout droit »*

Il ne se pose pas de problème pour ces manœuvres.



Le véhicule  $A$  entreprenant cette manœuvre sait qu'il va tout droit ( $i = 1$ ). Il est forcément le premier de son tronçon. Il opère comme suit :

— il teste le feu. Si celui-ci est rouge ou orange, il réagit comme décrit précédemment. Si celui-ci est vert, il recherche le dernier du segment dans lequel il va, soit le véhicule  $B$ , et le suit (loi de poursuite). Si  $B$  n'existe pas, il considère qu'il est en régime libre.

### 2.3.2.5. *Manœuvres « tourne à droite »*

Considérons la figure 4.

Le segment dans lequel l'on tourne à droite a son origine au feu précédent, c'est-à-dire bien à gauche du segment d'où l'on vient. On fait donc partir un véhicule qui tourne à droite d'un point nommé « point d'injection » (INJECT) pour tenir compte du fait que la distance entre 0 et INJECT n'est pas effectivement parcourue.

Par ailleurs, on définit sur le tronçon une abscisse de limite de ralentissement (LIMRAL), limite à partir de laquelle le véhicule doit commencer à ralentir avant de s'engager dans sa manœuvre de tourne à droite.

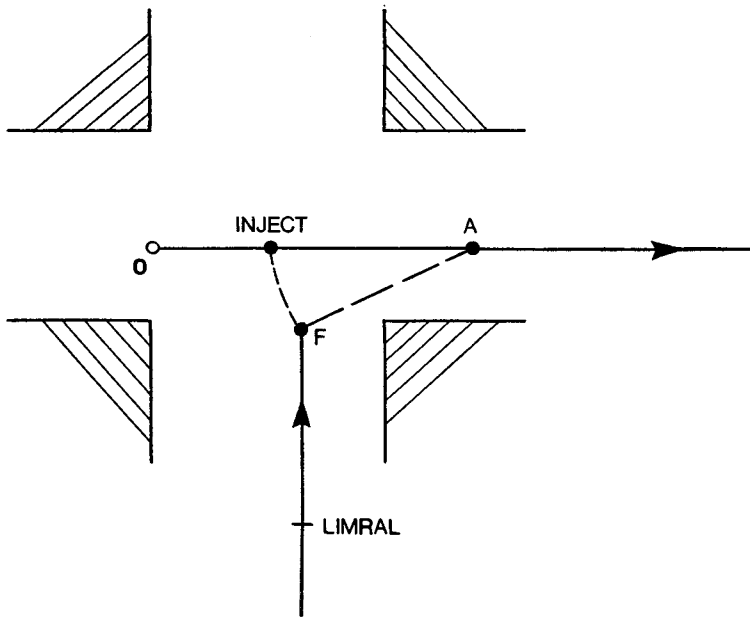
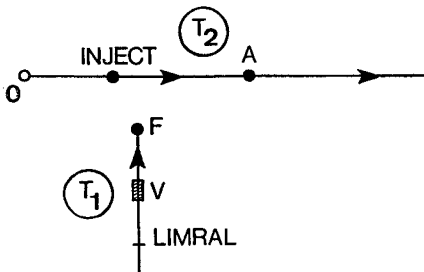


Figure 4

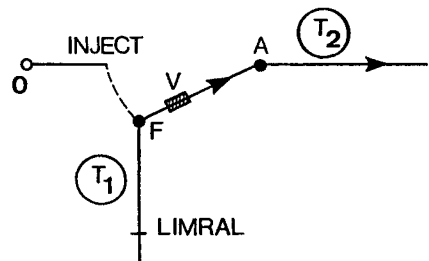
Le processus utilisé est un processus d'aiguillage schématisé sur la figure 5.

Le déroulement des opérations est le suivant (fig. 6) :

— le véhicule est en  $V_1$  sur le tronçon  $T_1$ . On sait qu'il veut tourner à droite en fin de tronçon ( $i = 2$ ) mais puisqu'il se trouve entre 0 et LIMRAL ( $T_1$ ) il n'y a pas lieu de modifier son mouvement;



Le véhicule  $V$  se présente en fin de tronçon pour tourner à droite.  
Configuration normale.



Le véhicule  $V$  est en train de tourner à droite.  
L'aiguillage se positionne pour le recevoir.

Figure 5

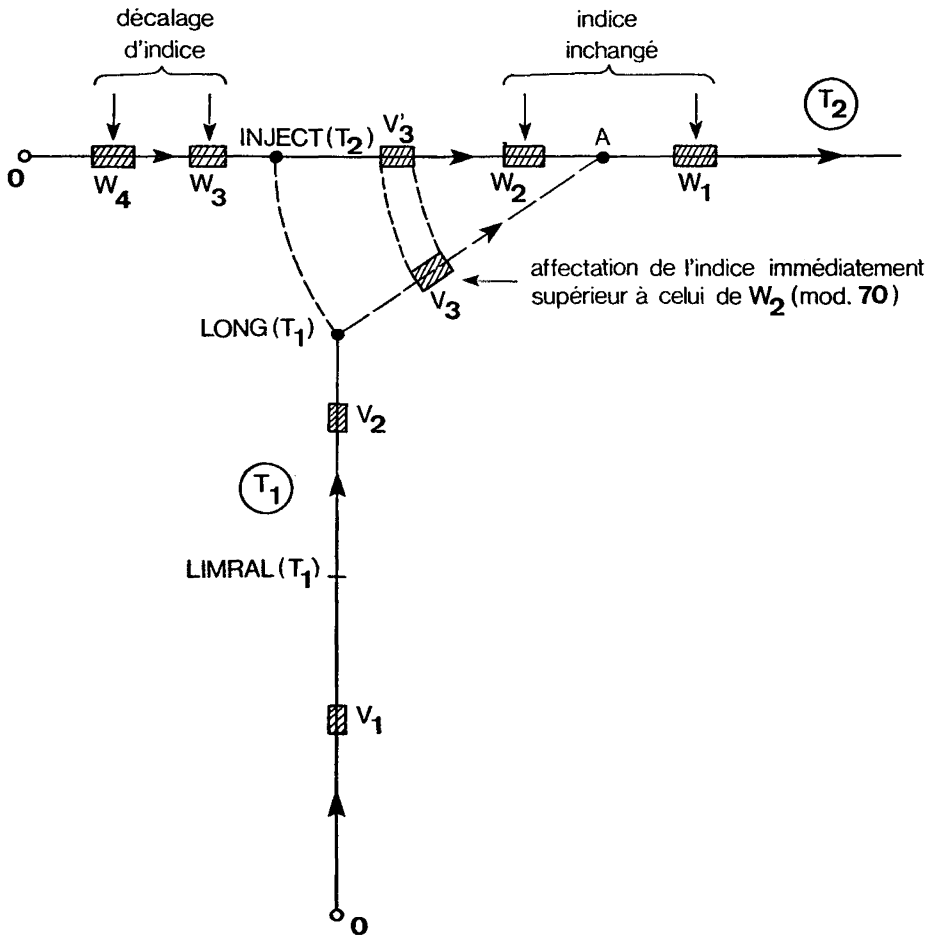


Figure 6

— le véhicule est en  $V_2$  sur  $T_1$ . Il a dépassé la limite de ralentissement. On le ralentit artificiellement en faisant décroître progressivement sa vitesse désirée.

Pendant tout ce mouvement, le véhicule a évidemment suivi le premier véhicule du tronçon  $T_2$  dont l'abscisse est supérieure à  $INJECT(T_2)$ . Dans ce cas, il s'agit de  $W_2$ .

— Le véhicule se trouve en  $V_3$  dans  $T_2$ .

Il a donc été rentré dans le tronçon  $T_2$  à partir de  $INJECT(T_2)$  et non à partir de zéro, et se trouve donc placé derrière  $W_2$  puisqu'il le suit. Mais il convient d'affecter à  $V_3$  un indice dans la liste des véhicules du segment  $T_2$ .

Il suit  $W_2$  et prendra donc l'indice immédiatement supérieur à celui de  $W_2$  (mod. 70). Les véhicules  $W_3$  et  $W_4$  qui suivent  $V_3$  verront leurs indices augmenter d'une unité (mod. 70) pour respecter la succession des véhicules.

$V_3$  étant alors en place dans le segment  $T_2$ , l'aiguillage pourra reprendre sa position initiale. On voit que ce système permet de respecter les règles de priorité à droite, même si le carrefour n'est pas vidé lorsque le feu de  $T_1$  passe au vert.

#### 2.3.2.6. Manœuvres de « tourne à gauche »

Le principe de base est sensiblement le même que pour les tourne à droite. On utilise encore un système d'aiguillage en faisant tourner le début du segment qui reçoit le tourne à gauche autour d'un point fictif  $A$  de manière que le zéro du tronçon vienne en  $F$ , feu du tronçon d'où l'on vient.

Ainsi les véhicules sont rentrés dans le segment  $T_2$  à l'abscisse zéro, ce qui permet d'aménager une zone de stockage dans la manœuvre de tourne à gauche. Cette zone est d'ailleurs représentative de la réalité. Par ailleurs, le véhicule désirant tourner à gauche ralentit lui aussi à partir de LIMRAL (fig. 7).

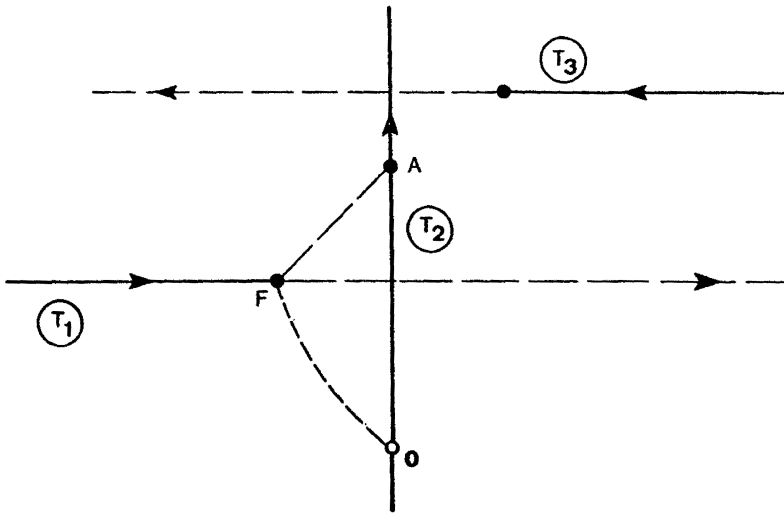


Figure 7

Il peut y avoir conflit avec le segment  $T_3$ . C'est pourquoi sur le segment est définie une abscisse de limite de conflit, LIMCFL, où les véhicules venant de tourner à gauche doivent s'arrêter si le conflit ne leur est pas favorable.

Examinons en détail ce qui se passe dans une manœuvre de tourne à gauche (fig. 8).

Deux cas peuvent se présenter :

— 1<sup>er</sup> cas : il n'y a pas de véhicule dans  $T_2$  entre les abscisses zéro et LIMCFL;

— le véhicule est en  $V_1$ . On sait qu'il va tourner à gauche ( $i = 3$ ) mais il n'a pas encore atteint l'abscisse LIMRAL, il n'y a pas lieu de modifier son mouvement;

— le véhicule est en  $V_2$ . Il a déjà entrepris son ralentissement puisqu'il a dépassé l'abscisse LIMRAL ( $T_1$ );

— le véhicule est en  $V_3$ . L'aiguillage s'est mis en position pour recevoir le véhicule. Il y a lieu alors d'examiner le conflit avec le véhicule  $W_1$  de  $T_3$  s'il existe. On calcule le temps  $t_1$  que doit mettre  $V_3$  pour dégager complètement le carrefour (c'est-à-dire atteindre le point  $D$ ), et le temps  $t_2$  que doit mettre  $W_1$  pour atteindre le segment  $T_2$ , et suivant le signe de la différence  $t_1 - t_2$  on décide que  $V_3$  peut passer ou bien qu'il doit s'arrêter à LIMCFL ( $T_2$ ) (véhicule fictif arrêté à cet endroit-là).

— 2<sup>e</sup> cas. Il y a déjà un véhicule  $V$  dans  $T_2$  entre 0 et LIMCFL ( $T_2$ ).

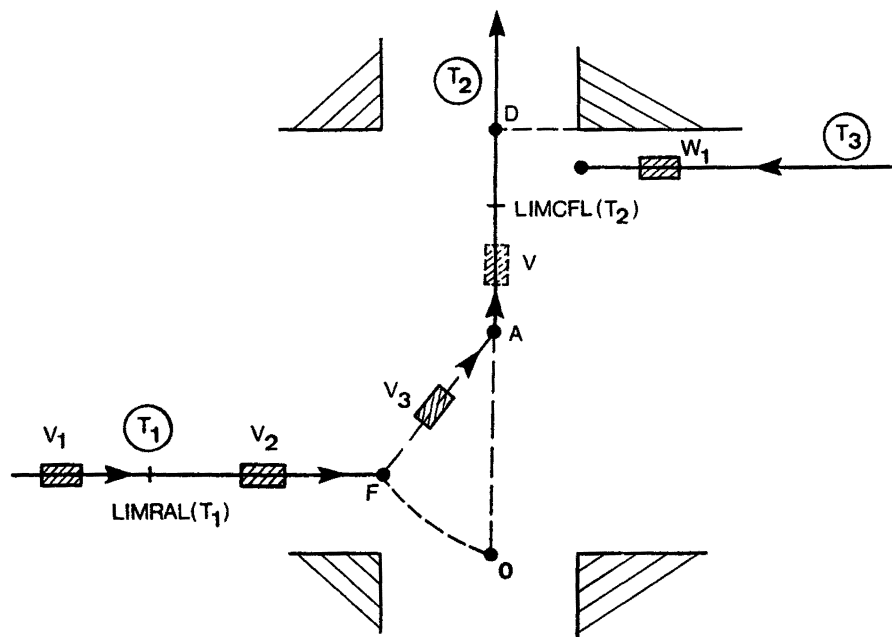


Figure 8

— Le véhicule est en position  $V_1$  ou  $V_2$  : rien de changé par rapport au cas précédent.

— Le véhicule est en  $V_3$ .

Il suit alors tout simplement le véhicule  $V$ , sans s'occuper du conflit, et vient s'arrêter derrière lui si le conflit de  $V$  est défavorable. Bien entendu, en même temps  $V$  s'occupe du conflit pour son propre compte, et dès que  $V$  aura dépassé LIMCFL ( $T_2$ ),  $V_3$  deviendra le premier après LIMCFL ( $T_2$ ) et nous serons ramenés au cas précédent.

### 2.3.3. Comportement des véhicules sur le réseau

C'est évidemment la variable  $i$  présentée ci-dessus qui indique dans quel sens va s'effectuer la manœuvre. Mais il convient de savoir d'où est tirée cette variable. Pour cela, on utilise une matrice de trajet. Un véhicule, dès qu'il est entré dans le réseau a un trajet parfaitement prédéterminé, on sait où il sortira et quel trajet il empruntera.

La destination du véhicule est consignée dans une partie de sa mémoire. C'est un nombre compris entre zéro et 31.

On porte en colonnes les différents segments, et en ligne les destinations possibles.

	1	2	3	...	$k$	...	$n$
1					.		.
2					.		.
3					.		.
.					.		.
.					.		.
$j$	.....				$N$		.
.							.
.							.
$n'$	.....						.

A l'intersection de la ligne  $j$  et de la colonne  $k$  se trouve un certain nombre  $N$ .

Cinq cas peuvent se présenter (le nombre de segments est limité à 80) :

—  $N$  est compris entre 1 et 80.

Un véhicule de destination  $j$  quittant le segment  $k$  doit entrer dans le segment  $N$  en allant tout droit ( $i = 1$ ).

—  $N = 0$ .

Le véhicule de destination  $j$  quittant le segment  $k$  doit sortir du réseau, ce qui revient à dire que  $i$  est une sortie.

—  $N$  est compris entre — 80 et — 1.

Un véhicule de destination  $j$  quittant le segment  $k$  doit entrer dans le segment —  $N$  en tournant à droite ( $i = 2$ ).

—  $N$  est compris entre 101 et 180.

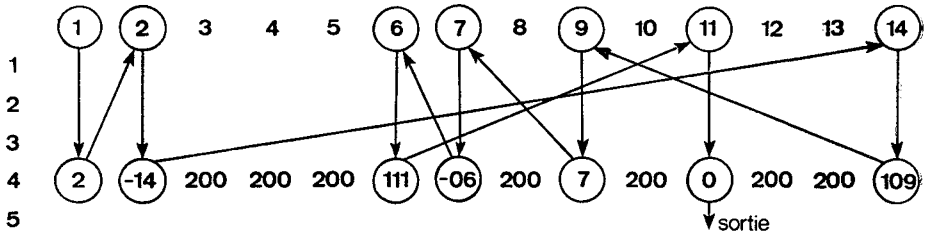
Un véhicule de destination  $j$  quittant le segment  $k$  doit entrer dans le segment  $N - 100$  en tournant à gauche ( $i = 3$ ).

—  $N = 200$  trajet non utilisé.

Ainsi, dans cette matrice, se trouvent des renseignements de deux types :

- le segment où l'on doit entrer,
- la manœuvre à effectuer pour y entrer.

La matrice ci-dessous montre un exemple de cheminement d'un véhicule.



Le trajet défini pour un véhicule de destination 4 se trouvant initialement dans le segment 1 est donc le suivant :

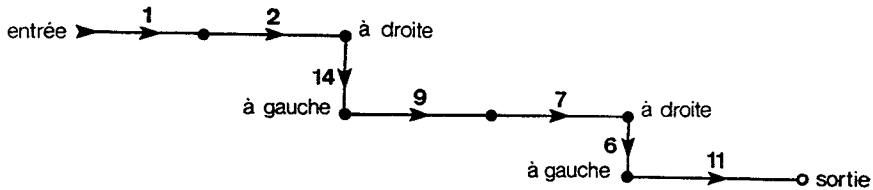
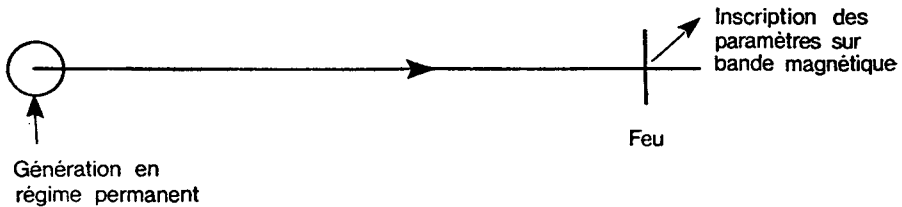


Figure 9

## 2.4. Génération des véhicules aux entrées du réseau

Les véhicules à entrer sur le réseau sont lus sur une bande magnétique d'entrée.

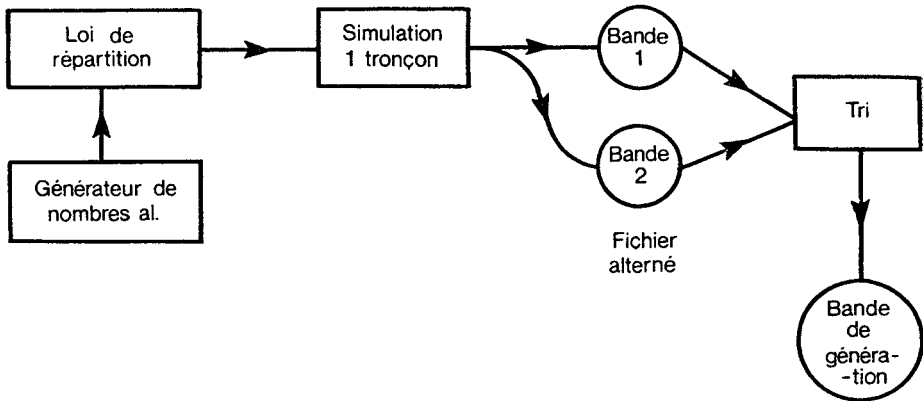
Pour créer cette bande, on procède de la manière suivante :



On génère les véhicules à l'entrée d'une longue voie. Cette génération est telle qu'elle respecte le débit imposé, et que les espacements entre véhicules ne puissent être inférieurs à  $2\Delta$  ( $\Delta$  = temps de réaction conducteur-véhicule).

On simule l'avancement des véhicules sur la voie. Au bout de la voie, se trouve un feu, et on récupère à la sortie de ce feu les paramètres de tous les véhicules quand ils se présentent. Ces paramètres sont inscrits sur bande magnétique.

Mais s'il y a  $n$  entrées sur le réseau, il faut pouvoir lire à un temps  $t$  donné les  $n$  entrées successivement. Or, l'on est obligé de simuler les  $n$  entrées séparément. Il faudra donc faire un tri sur bande pour créer la bande définitive servant d'entrée à la simulation proprement dite.



Grâce à cette méthode, on ne se place plus dans un réseau isolé, mais dans un complexe plus vaste, entouré de feux d'entrée.

### 3. REALISATION DU MODELE

#### 3.1. Organisation de la simulation

Soit  $\Delta$  le temps de réaction introduit dans la loi de poursuite. La révision de la situation se fait tous les  $\Delta$  c'est-à-dire que chaque véhicule présent dans le réseau est avancé toutes les  $\Delta$  secondes.

Il faut aussi faire un balayage dans l'espace, c'est-à-dire savoir dans quel ordre on va traiter les tronçons. Pour éviter de faire énormément de tâtonnements et de retours en arrière, il a été décidé, au risque de faire quelques erreurs dans des cas particuliers, de traiter les tronçons séquentiellement et toujours dans le même ordre.

L'ordre choisi est l'ordre décroissant des n<sup>os</sup> de segments.

On numérotera donc autant que possible les tronçons dans l'ordre croissant dans le sens de la marche.



Il est toujours possible cependant, qu'au moment d'un changement de direction, un véhicule passe d'un segment à un autre de numéro inférieur. Quand on traitera ce segment de numéro inférieur, le véhicule ne sera pas avancé de nouveau grâce à la présence d'un pointeur indiquant que le véhicule a déjà été avancé.

Quant à l'avancement, nous avons vu qu'il se réduisait la plupart du temps à des calculs d'indices puisqu'il n'y avait que très peu de transferts mémoires.

### **3.2. Déroulement de la simulation**

La simulation se déroule en quatre phases principales :

- lecture des données, contrôles, changements d'unité, réécriture des paramètres,
- remplissage du réseau, de manière à avoir une population initiale. Impression de l'état de cette population initiale,
- simulation proprement dite. Cette opération s'arrête soit après aboutissement du calcul, soit après saturation physique en mémoire d'une liste tronçon,
- analyse des résultats. Sortie des tableaux correspondants.

Ce déroulement est conforme à l'organigramme général de la figure 10.

### **3.3. Traitement des résultats**

Le programme d'analyse et d'interprétation des résultats peut évidemment être modifié suivant les paramètres que l'on veut obtenir. Il est même nécessaire de savoir exactement la validité des résultats que l'on obtient. Ainsi par exemple, une longueur de queue peut être microscopiquement définie de plusieurs façons. Nous présentons simplement ici la prise des informations au cours de la simulation et les traitements de ces informations tels qu'ils sont conçus à l'heure actuelle.

#### **3.3.1. Prise des informations au cours de la simulation**

Il est nécessaire de collecter des résultats au cours de la simulation de manière à pouvoir les traiter par la suite, une fois la simulation terminée.

Les opérations suivantes sont effectuées :

- en début de travail, on détermine le nombre de véhicules saturant physiquement un segment.

Au cours de la simulation, l'existence d'un segment saturé entraîne un message sur l'imprimante indiquant à quelle heure se produit cette saturation.

Par ailleurs, chaque fois qu'un véhicule entre dans un segment, on contrôle chaque fois que le nombre de véhicules présents dans le segment est  $\leq 70$ .

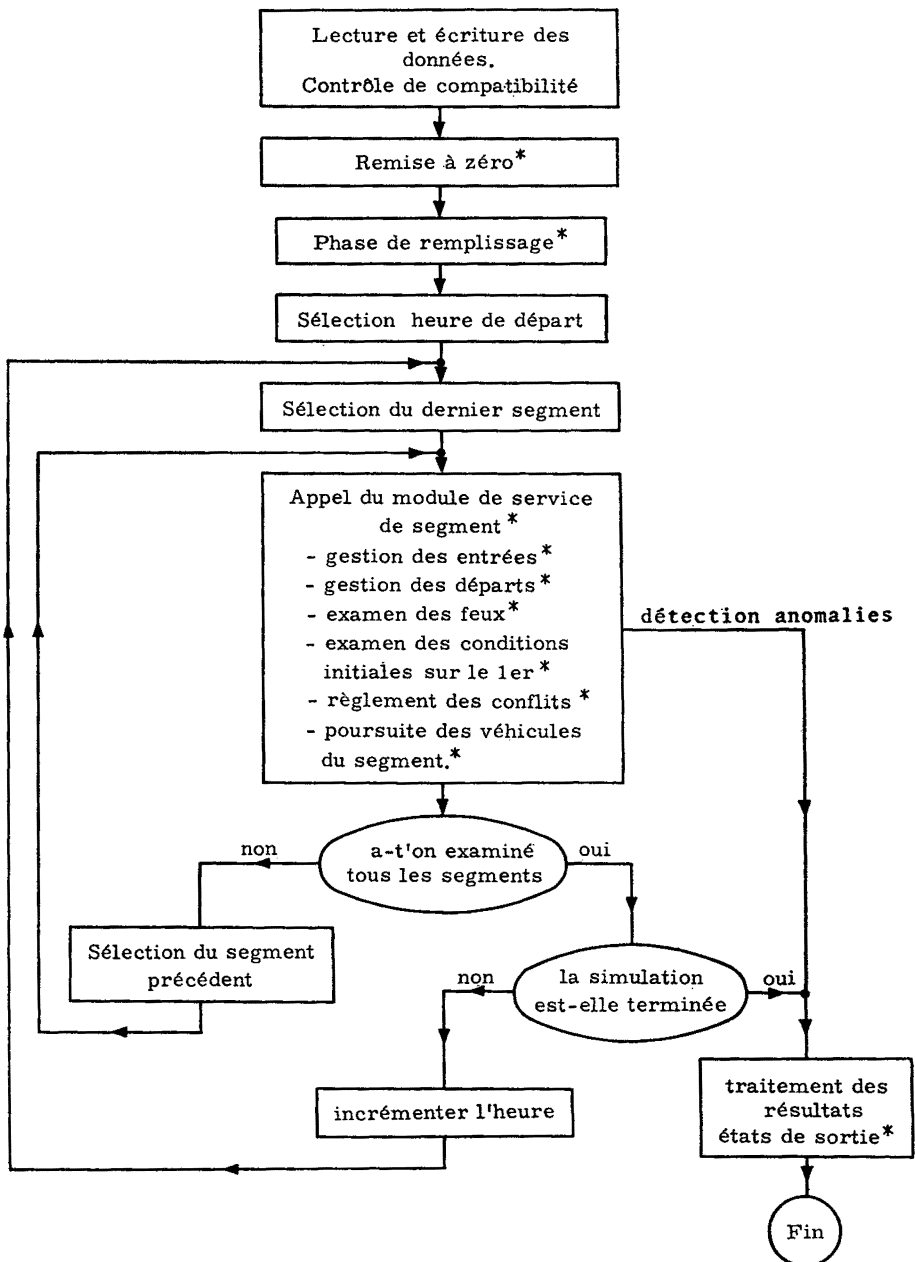


Figure 10

La présence d'un \* indique un sous-programme.

Sinon, il y a arrêt de la simulation, traitement des résultats obtenus, et fin du travail.

— Lorsque la phase de remplissage est terminée, on sort sur imprimante la population initiale sur chaque segment, c'est-à-dire le nombre de véhicules existants dans le segment en fin de remplissage.

— Tout au cours de la simulation, on incrémente deux tableaux : l'un du nombre d'entrées, l'autre du nombre de sorties, chaque fois qu'il y a entrée ou sortie d'un segment ou du réseau.

— Chaque fois qu'un véhicule sort du réseau, on consigne sur bande les mémoires représentant le véhicule pour traitement ultérieur.

— Chaque fois que dans un segment un véhicule est arrêté (avancement nul) on incrémente une mémoire représentant le temps perdu dans le segment.

— De même, on comptabilise pour chaque véhicule le nombre d'arrêts de celui-ci.

### 3.3.2. Traitement des informations

Le programme de traitement intervient lorsque la simulation est terminée (voir fig. 10).

On commence par imprimer la population finale dans chaque tronçon, et le nombre d'entrées et de sorties par tronçon, ainsi que les débits d'entrée et sortie par tronçon, on a ainsi la tendance générale du trafic.

Le traitement proprement dit des résultats s'articule en deux phases :

— traitement par tronçons.

Celui-ci permet d'imprimer :

- le temps total perdu par tronçon,
- la queue instantanée moyenne dans le tronçon,
- le temps moyen perdu par véhicule dans un tronçon,
- le nombre d'arrêts moyens par tronçon.

— traitement par trajets.

Celui-ci ne concerne que les véhicules ayant emprunté un trajet dans sa totalité, c'est-à-dire déjà sortis du réseau en fin de simulation.

On imprime sur les trajets effectivement utilisés :

- la longueur de chaque trajet,
- le nombre de véhicules ayant emprunté ce trajet,
- le temps de parcours total des véhicules sur le trajet,
- le temps de parcours moyen d'un véhicule sur le trajet,
- le nombre d'arrêts total sur le trajet,
- le nombre d'arrêts par véhicule sur le trajet,
- la vitesse moyenne d'un véhicule sur le trajet.

### 3.4. Programmation

Celle-ci a été faite pour l'IBM 7044 de la faculté des Sciences de Toulouse. Le langage employé est le FORTRAN IV sauf le sous-programme de lecture/écriture dans les zones d'une mémoire, qui se présente sous la forme d'une macro-instruction en MAP.

Le modèle est constitué :

- d'un programme principal jouant le rôle de moniteur,
- de 12 sous-programmes de service, de conception modulaire,
- d'une macro-instruction de lecture/écriture dans deux mots de mémoire,
- de 6 fonctions de service.

Dans l'état actuel du modèle, celui-ci occupe en mémoire centrale environ 24 000 mots.

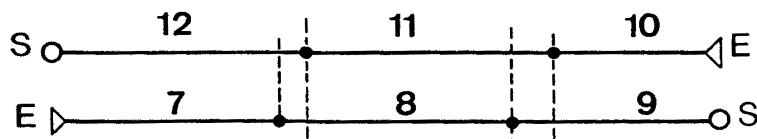
Pour un réseau comportant 18 tronçons, 6 entrées et 6 sorties, comprenant en moyenne 200 véhicules pour simuler 10 mn de temps réel, le sous-programme de poursuite des véhicules est appelé 92 000 fois, et il faut 5 mn 21 s de temps machine.

## 4. RÉSULTATS DE LA SIMULATION

Nous présentons ici quelques listings de résultats obtenus sur le réseau précédent. Le listing des données ne présentant que peu d'intérêt, nous ne le faisons pas figurer.

Nous donnons simplement les résultats par segments (fig. 11) et par trajets (fig. 12), ainsi que la visualisation des diagrammes espace temps des véhicules (fig. 13) que l'on peut demander en option.

Sur la figure 13 se trouvent représentés les segments 7, 8 et 9 de gauche à droite, et 10, 11 et 12 de droite à gauche. Les chiffres de gauche indiquent le temps en secondes. La présence de points indique un croisement de deux véhicules.



Les apparitions et disparitions de véhicules aux carrefours sont évidemment le résultat des changements de direction (disparition ou apparition dans le segment visualisé).

## REPARTITION PAR SEGMENT

SEG	POP.	T. PERDU TOTAL	QUEUE MOYENNE	T. PERDU PAR VEH.	NBRE ARR TOTAL	N. ARR PAR VEH
1	7	524.0	1.0	8.2	19	0.30
2	19	3569.0	7.1	71.4	75	1.50
3	11	118.0	0.2	1.2	20	0.20
4	36	3816.0	7.6	53.7	93	1.31
5	1	202.0	0.4	7.8	10	0.38
6	1	268.0	0.5	4.1	28	0.43
7	12	654.0	1.3	3.5	32	0.42
8	1	308.0	0.6	7.3	20	0.48
9	8	106.0	0.2	1.9	17	0.31
10	1	580.0	1.2	6.9	33	0.39
11	30	2406.0	4.8	34.9	72	1.04
12	1	34.0	0.1	1.0	7	0.21
13	48	1107.0	2.2	8.1	72	0.53
14	8	703.0	1.4	16.7	33	0.79
15	6	126.0	0.3	2.5	14	0.27
16	44	2423.0	4.8	21.3	109	0.96
17	2	159.0	0.3	2.8	11	0.20
18	1	23.0	0.0	0.5	5	0.11

POPULATION TOTALE 237

NOMBRE D'ENTREES 508

NOMBRE DE SORTIES 323

APPELS A SUIVI 74504

Figure 11

## ETAT DE SORTIE

TRAJET	LDNG TOT	NBRE VEH	T. PAR. TOT.	T. PAR. MOY.	NBRE ARR TOT	NBRE ARR MOY	VIT. MOY.
E 1/S 1	950	18	3093.	171.83	26	1.44	19.90
E 1/S 3	1000	12	2370.	197.50	21	1.75	18.23
E 1/S 4	600	12	1702.	141.83	17	1.42	15.23
E 1/S 5	1100	6	1483.	247.17	14	2.33	16.02
E 1/S 6	850	10	1495.	149.50	13	1.30	20.47
E 2/S 2	950	4	800.	200.00	6	1.50	17.10
E 2/S 3	750	4	579.	144.75	6	1.50	18.65
E 2/S 4	750	10	1646.	164.60	18	1.80	16.40
E 2/S 5	850	12	2018.	168.17	14	1.17	18.20
E 2/S 6	700	51	7347.	144.06	53	1.04	17.49
E 3/S 1	750	6	568.	94.67	4	0.67	28.52
E 3/S 2	600	17	1499.	88.18	14	0.82	24.50
E 3/S 3	800	2	373.	186.50	3	1.50	15.44
E 3/S 5	900	19	3292.	173.26	37	1.95	18.70
E 3/S 6	650	16	1352.	84.50	12	0.75	27.69
E 4/S 1	750	8	2026.	253.25	20	2.50	10.66
E 4/S 2	1000	13	1777.	136.69	19	1.46	26.34
E 4/S 4	800	2	201.	100.50	2	1.00	28.66
E 4/S 5	500	24	1918.	79.92	24	1.00	22.52
E 4/S 6	650	4	790.	197.50	8	2.00	11.85
E 5/S 1	700	3	444.	143.00	4	1.33	17.03
E 5/S 2	850	5	971.	194.20	10	2.00	15.76
E 5/S 3	650	6	948.	158.00	10	1.67	14.81
E 5/S 4	650	12	2266.	188.83	26	2.17	12.39
E 5/S 5	750	3	453.	151.00	5	1.67	17.88
E 6/S 1	850	11	2300.	209.09	19	1.73	14.63
E 6/S 2	1100	6	726.	121.00	5	0.33	32.73
E 6/S 3	500	8	583.	72.88	7	0.33	24.70
E 6/S 4	900	11	1144.	104.00	9	0.82	31.15
E 6/S 6	750	8	1425.	178.13	13	1.63	15.16

Figure 12

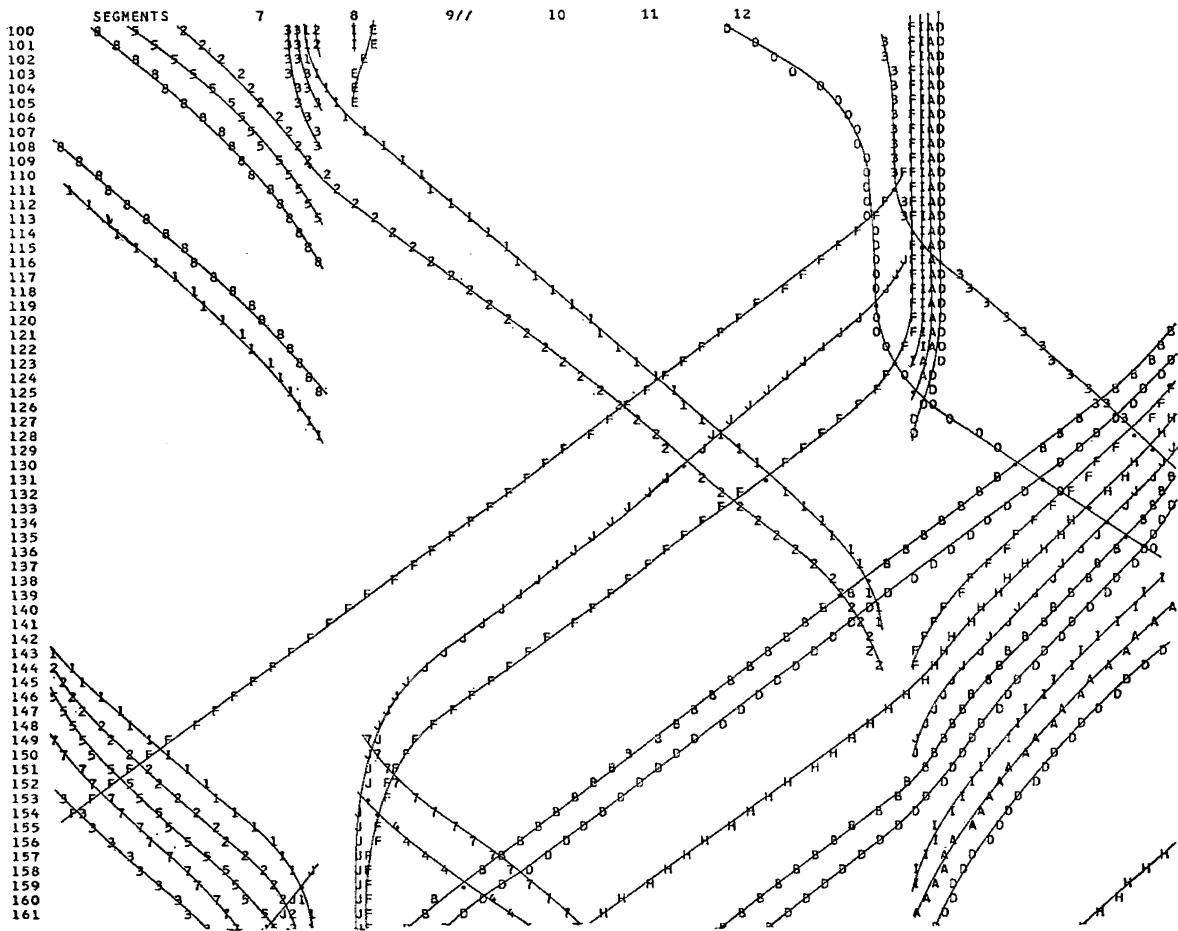


Figure 13

## 5. CONCLUSION

Le modèle présenté ne pourra être utile à nos recherches — le but est de trouver des algorithmes de commande d'un petit réseau — que s'il est validé.

La phase de validation est d'ores et déjà commencée, puisqu'à l'aide de photographies aériennes d'artères toulousaines prises à des intervalles de 1 seconde, nous sommes en train de rechercher les paramètres fondamentaux de la loi de poursuite. Un dépouillement plus complet de ces photos permettra par la suite de valider le comportement aux carrefours. Le modèle devra vraisemblablement être retouché dans certains détails au vu des mesures, mais nous avons bon espoir d'obtenir ainsi un outil correct pour nos recherches en

optimisation, qui nous permettra de mieux comprendre les phénomènes de trafic.

Nous tenons à remercier ici MM. Noailles et Genthon qui ont contribué à la définition générale de ce modèle.

#### BIBLIOGRAPHIE

- [1] D. L. GERLOUGH, « Simulation of freeway traffic by an electronic computer », *Proceedings of HRB*, 1956, n° 35, pp. 543-547.
- [2] D. L. GERLOUGH, « Simulation of traffic flow. An introduction to traffic flow theory », *HRB Special Report*, 79, pp. 97-118.
- [3] H. H. GOODE, C. H. POLLMAR et J. B. WRIGHT, « The use of a digital computer to model a signalized intersection », *Proceedings of HRB*, vol. 35, 1956, pp. 548-557.
- [4] H. H. KATZ, Final Report study of the traffic signal system in a portion of the district of Columbia, *TRW California*, Nov. 1962.
- [5] J. H. KELL, « Analyzing vehicular delay at intersections through simulation », *HRB Bulletin*, n° 356, 1962, pp. 28-39.
- [6] R. M. LEWIS et H. L. MICHAEL, « Simulation of traffic flow to obtain volume warrants for intersection control », *HRB*, 1953, pp. 1 à 43.
- [7] R. SAGEN, Traffic Simulation. Paper presented at the IVth International Symposium on the theory of traffic flow Karlsruhe University. Germany, 15-20 June 1965.
- [8] M. C. STARK, « Computer simulation of street traffic », *National Bureau of standards Technical*, note n° 119.
- [9] M. C. STARK, « Computer simulation of nine blocks of a city street », *HRB Bulletin*, n° 356, 1962, pp. 40-47.
- [10] D. L. GERLOUGH, « Traffic Inputs for simulation on a digital computer », *Proceedings of HRB*, 1959, n° 38, pp. 480-492.
- [11] B. D. GREENSHIELDS et F. M. WEIDA, « Statistics with application to Highway Traffic Analysis », *The Eno Foundation for Highway Traffic control*, 1952.
- [12] M. LEGLISE, Étude du contrôle et de la conduite centralisée de la circulation en zone urbaine, avec application à la ville de Toulouse. *Compte rendu d'une recherche financée par la DGRST. Action concertée Automatisation*, contrat 68-01-310, mai 1969 (annexes V, VI, VII, VIII).
- [13] G. AUTHIE et M. LEGLISE, Étude d'un modèle de poursuite entre véhicules. *Journées d'étude sur la régulation du trafic sur un réseau urbain*. Toulouse, 13-14 mars 1969 (Laas).
- [14] R. W. CONWAY, B. M. JONHSON et W. L. MAXWELL, « Some problems of digital systems simulation », *Management Science*, October, 6, 1959, pp. 92-110.
- [15] D. C. GAZIS, R. HERMAN et A. A. MARADUDIN, « The problem of the Amber Signal light in traffic flow », *Ops. Res.*, 8, 1960, pp. 112-132.
- [16] P. H. OLSON et R. W. ROTHERY, « Driver Response to the Amber Phase of traffic signals », *Ops. Res.*, 9, 1961, pp. 650-663.
- [17] M. BILLOTTE, M. FAUCHEUX et M. SAHAROVITCH, Simulation de la circulation sur un axe routier à feux adaptatifs. *Actes des journées d'Étude sur la régulation du trafic sur un réseau urbain*. Toulouse, 13-14 mars 1969, Laas-Toulouse.