

# ITSのためのトラフィックシミュレータの開発

東京大学工学部助教授 藤岡 健彦

東京大学大学院工学系研究科 伏木 匠(発表者)

## 1. はじめに

道路交通のインテリジェント化は、欧米日を中心として活発に研究、開発が行われている。道路交通を取り巻く問題の中でも、特に物流に関しては、小口多頻度の輸送形態の増加、労働力不足や高齢化などの問題が顕著になっており、ITSの技術を大いに活用すべき分野である。現在、新物流システムとして、専用走行路では無人自動運転、一般道路では有人走行を行う Dual Mode Truck(DMT)に関する研究が、建設省土木研究所を中心に行われている。本研究では、DMTのモデル形成から始まり、加減速や追従走行をする様子、また自動的に合流、分岐する様子を再現し、さらには交通流に及ぼした影響を評価検討することを目的としている。また、必要があれば、アニメーションを用い表現することも考える。

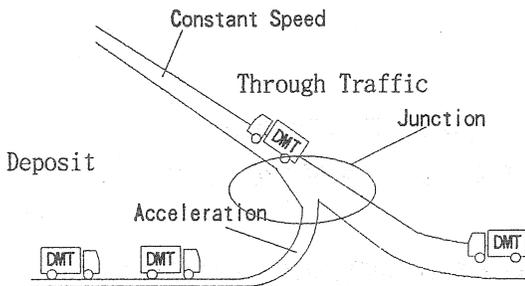


Fig. 1: Image of Advanced and High Efficiency Freight Transport Systems

## 2. シミュレーション方法

### 2.1. 車両モデル

個々の車両を制御するために、車両のモデルを形成する。車両の左右方向には、案内装置により誘導されるので、道路に拘束されるとする(実際にはかなり振動があり問題も多い)。よって前後方向に関しての運動方程式のみをたてる。質量を  $m$ 、回転部分相当質量を  $m_1$ 、速度を  $u$ 、駆動力を  $F_x$ 、走行抵抗を速度の関数  $drag(u)$  とすると運動方程式は以下ようになる。走行抵抗としては、転がり抵抗、空気抵抗、縦勾配を考慮する。

$$(m + m_1)\dot{u} = F_x - drag(u) \quad (1)$$

### 2.2. 制御方針

次に制御方法を検討する。先行する車両が存在する場合は、車頭時間間隔(タイムヘッドウェイ)を一定とした車間距離制御、つまり速度に比例した車間を保つような制御を行う。先行車両が存在しない場合は、ネットワークからの指示による速度、加速度制御を行う。どちらの制御を行うか、すなわち先行車両の存在を認識するか否かは、物理的なもの、あるいはその車間距離で決まる。車間距離制御は、車間距離測定に用いるレーザーレーダの視界に先行車両が存在しない場合、例えば、測定距離の限界(ここでは 150[m]とした)を超えている場合や、コーナーなどで前方の車両が捕らえられない場合は追従走行は不可能である。速度、加速度制御は、先行車両を認識してから通常の制動を行って停止できるだけの余裕があれば速度、加速度制御を安全に行える。この距離を判定距離

$D_{judge}$  と定義して式で表せば以下ようになる。

$$D_{judge} = u_2(1 + \epsilon)T_d + \frac{\{u_2(1 + \epsilon)\}^2}{2\alpha(1 - \epsilon)} \quad (2)$$

ここで  $u_2$  は後続車両の速度、 $T_d$  は空走時間(0.1[s])、 $\alpha$  は標準の減速度(3.5[km/h/s])、 $\epsilon$  は速度や加速度のばらつきである。式では、ばらつきを安全側にとることで、不確実性を簡便に考慮している。

先行車両に対する追従走行を行う際は、車頭時間間隔一定で行うので、その際の目標車間距離を  $D_{head}$  として定義すると以下の式で表される。

$$D_{head} = u_2T_h - L \quad (3)$$

ここで  $T_h$  はタイムヘッドウェイ、 $L$  は車両長さである。

$D_{head}$  は、急制動時に必要な車間距離よりも大きい必要がある。緊急時にブレーキをかける場合、

先行車両が瞬時に停止するとは考えにくい。したがって先行車両が緊急ブレーキを開始したときに緊急信号を発生し、後続車両も緊急ブレーキをかけて衝突を回避すると考える。このために最低必要な距離を安全距離と定義し、 $D_{min}$ として式で表すと以下ようになる。

$$D_{min} = u_2(1+e)T_d + \frac{\{u_2(1+e)\}^2}{2\beta(1-e)} - \frac{\{u_1(1-e)\}^2}{2\beta(1+e)} + \delta \quad (4)$$

ここで  $u_1$  は先行車両の速度、 $\beta$  は緊急ブレーキによる減速度(14.0[km/h/s])、 $\delta$  は停止余裕距離である。

以上をグラフにしたものがFig. 2である。ここではばらつきを5%、タイムヘッドウェイを3[s]としている。グラフは、ある車間距離、速度のときにどの制御がおこなわれるかを示している。グラフで、目標車間距離と安全距離の線が交差していないことから、タイムヘッドウェイ3[s]にとっても充分安全に制御ができると判断した。

制御の方針が定まったので、追従走行に基づく車間距離制御、ネットワーク指示に基づく速度、加速度制御の具体的方法を考える。制御が急に切り替わっても、さほど変化が現れないようにするために、画一的な制御方法を考える。車両自体は、式(1)で示されるようなダイナミクスをもっているため、速度、加速度を自由に変化させることはできない。これに対して、外部からの指令を即座に反映できる仮想のターゲットを発生させ、このターゲットに追従するように制御することによって、車両を制御目標の状態に一致させる。ただしターゲットの発生させ方が、制御方法によって異なる。

まず、速度、加速度制御を行う場合の仮想ターゲット、すなわち先行車両を認識しないときの仮想ターゲットは、Fig. 3のように与えられる。ターゲットの初期位置は、速度、加速度制御に切り替わったときに、もともと有していた位置情報を与える。その後、ターゲットは、道路側から与えられる加速度(Fig. 3の  $A_{desired}$ )に基づいて道路側の与える目標速度(Fig. 3の  $V_{desired}$ )になるまで等加速度運動する。

先行車両を認識するとき、つまり車間距離制御を行うときの仮想ターゲットは、Fig. 4のように与えられる。ターゲットの相対位置は、タイムヘッドウェイが3[s]となる位置を、速度、加速度については先行車両のものを、毎制御サイクルごとに更新する。先行車両の各状態量は車両が備える

レーザレーダの情報から与えられるが、車々間通信のサポートがあれば、より確実に速度、加速度を伝えることができる<sup>1</sup>。

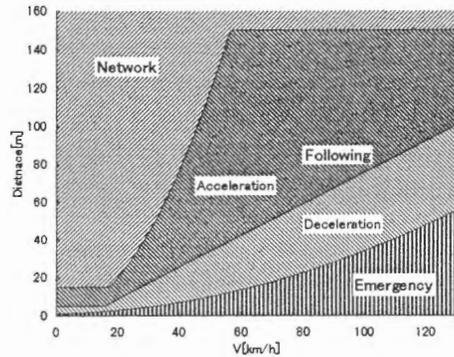


Fig. 2: Control Map according to Inter-Vehicle Distance and Velocity

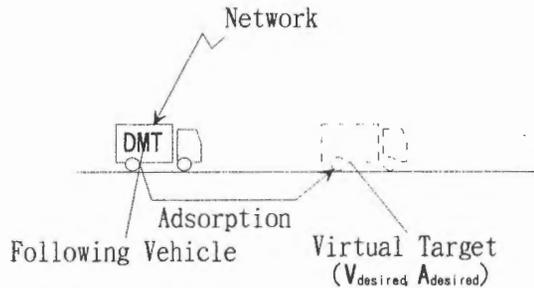


Fig. 3: Virtual Target (without a forward vehicle)

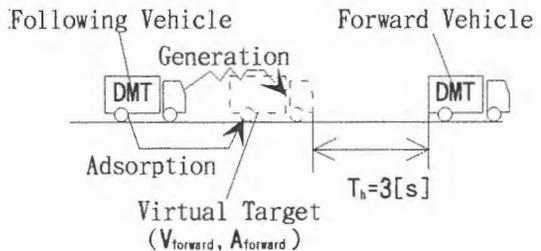


Fig. 4: Virtual Target (with a forward vehicle)

### 2.3. 制御理論

どんな制御方法についても、車両は仮想ターゲットに対して吸着するような制御を行う。車両が吸着するための追従制御理論としては、Sliding制御理論を用いる。

仮想ターゲット、追従車両の進行方向位置を  $x_r$ 、 $x$  とすると車間距離誤差  $\varepsilon$  は以下ようになる。

$$\varepsilon = x - x_T \quad (5)$$

次に  $\varepsilon$  を使い時変スカラー量  $s(t)$  を式のように定義する。

$$s(t) = \dot{\varepsilon} + c_1 \varepsilon + c_2 \int \varepsilon dt \quad (c_1, c_2 > 0) \quad (6)$$

この  $s(t)$  を指数関数的に 0 に収束させると、目標状態に収束する。

$$\dot{s}(t) = -Ks(t) \quad (K > 0) \quad (7)$$

式(5)~(7)を式(1)に代入して  $F_x$  について解き、変形すると以下ようになる。

$$\begin{aligned} F_x &= (m + m_f)(\ddot{x}_T - c_1 \dot{\varepsilon} - c_2 \varepsilon - Ks(t)) + drag(\dot{x}) \\ &= (m + m_f)\{\ddot{x}_T - (c_1 + K)\dot{\varepsilon} - (c_2 + Kc_1)\varepsilon \\ &\quad - Kc_2 \int \varepsilon dt\} + drag(\dot{x}) \end{aligned} \quad (8)$$

(8) 式で  $K' = c_2 + Kc_1$ 、 $T_D = (c_1 + K)/K'$ 、 $T_I = K'/Kc_2$  とおけば、式(9)のように書き換えられる。

$$\begin{aligned} F_x &= (m + m_f)\{\ddot{x}_T - K'(\varepsilon + T_D \dot{\varepsilon} + \frac{1}{T_I} \int \varepsilon dt)\} \\ &\quad + drag(\dot{x}) \end{aligned} \quad (9)$$

以上から求めた駆動力  $F_x$  を制御する。実際は  $F_x$  は直接制御できないが、駆動系から逆算して求めるものとする。

### 3. シミュレーション

以上のようにして定めた制御方針の確認を行う意味で、シミュレーションを行う。まず、制御パラメータ  $K'$ 、 $T_D$ 、 $T_I$  を定めるために、3 秒おきに走行する 12 台の車両が、40[km/h]から 60[km/h]に加速するシミュレーションを行った。このときのターゲットとの誤差、速度変化、車両の軌跡を示したものが、それぞれ Fig. 5、Fig. 6、Fig. 7である。図からは、最初の 1 台のみがネットワーク指示による速度制御、残りは前方の車両をタイムヘッドウェイ 3 秒で追従走行している様子がわかる。Fig. 5、Fig. 6において、安定性、即応性の両面から、速度制御、追従走行のどちらの場合でもうまく収束するように、

制御パラメータを採用したところ、 $K' = 5$ 、 $T_D = 0.5$ 、 $T_I = 1$  となった。また Fig. 7からは、マクロな視点では、遅れや擾乱が、後方にさほど影響していないように思われる。

次に、荷重を 100%から 0%に変化させて同様のシミュレーションを行う。つまり、車両モデルと実車両の質量に差がある場合を想定する。この場合のターゲットとの誤差、速度変化を Fig. 8、Fig. 9に示す。なお制御パラメータは前述のシミュレーションと同じ値を用いる。この場合、1 台目の車両が加速開始、終了時に生じた振動が、後方の車両に伝わり、増幅されている。制御系の安定性により収束するが、変位の振幅自体は小さいが、乗り心地（無人なので、荷物の積み心地の方が適切であろう）に影響する加速度振動を考慮すると、大きな問題である。

今度は縦勾配に対する影響を調べる。条件は、3 秒おきに走行する荷重 100%の車両 12 台が、60[km/h]で一定速度走行しているときに、5%の勾配が 100m 続く場合を考える。制御パラメータは先二つと同じものを用いる。このときの 1、4、7、10 台目のターゲットとの誤差、速度変化を Fig. 10、Fig. 11に示す。ここでも細かくはあるが、振動が発生し、それが後方に次々と伝播している。さらに言えば、後方の車両は、前方の車両に伝わってきたすべての振動と、自ら引き起こす振動を累積することになり、かなり多くの振動が観測される。また、このときも振幅自体は小さいが、加速度振動は大きなものとなっている。

### 4. 結論

新物流システムの一部をモデル化し、その影響を調べるシミュレータの基礎部分の開発は達成された。これにより制御における問題点が明確になった。具体的には、制御パラメータのとり方の基準を、荷重変動が車両の動きに与える影響、勾配が車両の動きに与える影響を考慮してとる必要があることがわかった。また、制御方針にも工夫を加える必要があることがわかった。今後に向けては、合流・分岐制御、停止時のモデルなどを加え、より本来の姿に近づけることが、重要な課題である。

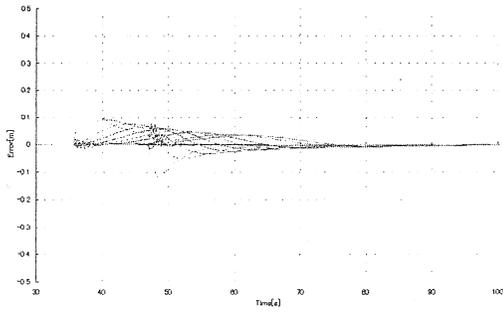


Fig. 5: Difference between Target and Vehicle

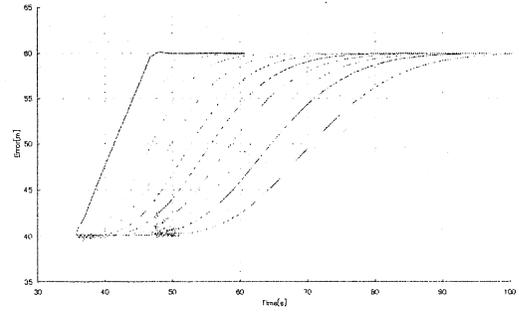


Fig. 9: Change of Velocity (without load)

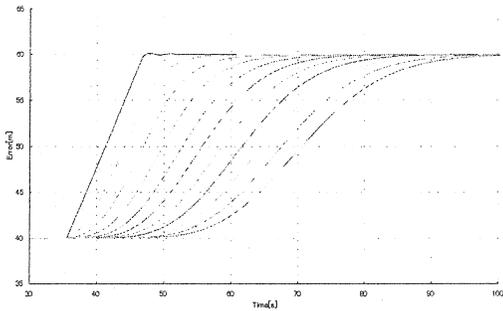


Fig. 6: Change of Velocity

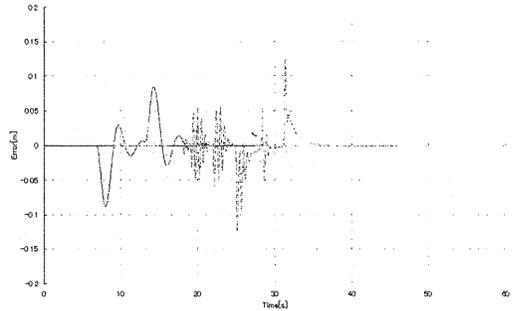


Fig. 10: Difference between Target and Vehicle (at 5% slope)

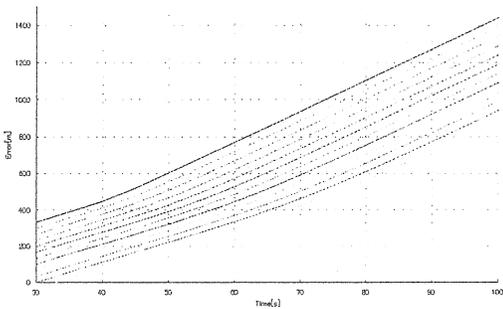


Fig. 7: Trajectory of the Vehicles

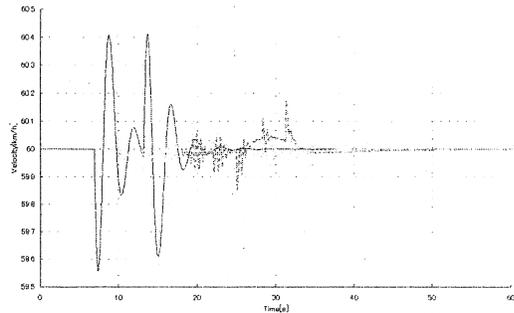


Fig. 11: Change of Velocity (at 5% slope)

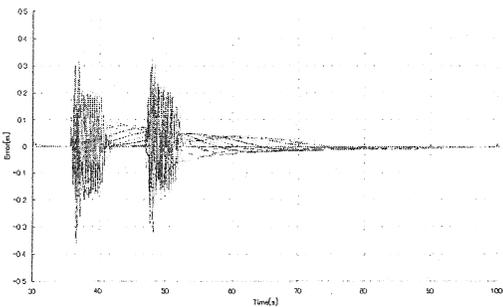


Fig. 8: Difference between Target and Vehicle (without load)

## 参考文献

- [1] 藤岡 健彦, 麻生 誠: 車々間通信を用いた自動車の自動運転における縦方向制御に関する研究, 自動車技術会学術講演会前編集, No.963, (1996), pp.221-224.
- [2] Farokh Eskafi, Delnaz Khorramabadi and Pravin Varaiya, "SmartPath: an Automated Highway System Simulator", University of California