

タイトル:「車の渋滞を考える」

1. はじめに

私は田園都市線の青葉台駅から半蔵門線の半蔵門駅の近くにあるオフィスまで通っています。田園都市線は渋谷駅で半蔵門線に接続し、半蔵門線は押上駅で東武鉄道に接続して、神奈川県から東京都を経て、埼玉県さらには栃木県まで4都道府県を繋いでいる長い路線です。

長い路線上で起きる故障や事故が影響して、電車が遅れることが頻繁に起こります。特に朝の通勤時間帯の遅れは尋常ではなく、遅れない日の方が少ないと言えるほどです。駅の直前で止まったりするとイライラは頂点に達しますが、日本人は皆おとなしいですから、体を寄せ合った中でひたすら人生修業だと思って耐えています。

そんな折に決まって同じ口調で「お客様にはご迷惑をお掛けして申し訳ございません。深くお詫びいたします」と駅ごとに、あるいはそれ以上に頻繁に車掌のアナウンスが入りますが、鉄道会社の人には、そのアナウンスが乗客の苛立ちを一層増すことになっていることを推測できないようです。

その不快さの中でいつも考えるのは、1本の線路の上に電車を乗せすぎているのではないのだろうか、すなわち、輸送の限界を超えているのではなかろうかということですが。鉄道会社の専門の方々が検討を重ねながら運転管理システムを改善してきているわけですから改善は限界に達しているのかも知れません。

しかし、1、2年前に運転管理システムに関する新しい試みが効果をあげたこともあると言われています。それは、朝8時台に渋谷駅に到着する上り電車については、急行をなくして準急として、二子多摩川駅から渋谷駅までそれまでの通過駅に全て停車することにしたところ、以前の急行停車駅での乗降車時の混雑が減って、二子多摩川駅から渋谷駅までの所要時間が総合的に数分短くなったそうです。急行に乗車していた乗客の一人としてはその効果を享受しているわけではありませんが。

そんなわけで、電車の運行管理を仮想的に実行して渋滞の苦痛を緩和する方法が他にないか探りたいと思ったのですが、電車には定時管理と、鉄道会社の複雑な運転規則が存在しますから、かなり難しくなりそうです。それでまずは1車線の道路について、車列のモデリング&シミュレーションから始めることにしました。

鉄道に関して長々と書きながら、車の話に安易に結び付けてしまって、ご迷惑をお掛けして申し訳ございません。深くお詫びいたします。

2. 渋滞

渋滞の話に関しては、西成勝裕先生が、新潮選書の“渋滞学”や宝島社新書の“「渋滞」の先頭は何をしているか？”で分かりやすく解説しておられます。車の渋滞は

我々に最もなじみ深い渋滞ですが、これは電車の渋滞に似ていることが直感的に分かります。それだけでなく、社会で起きるさまざまな渋滞ともよく似た現象であることも推測できます。例えばビジネスにおけるサプライチェーン管理とか、人事のキャリアパス管理とか、工場の生産管理などでの渋滞もそうです。車の渋滞における車間距離はいわば“貯まるもの”ですが、サプライチェーン管理でそれに相当するのが在庫量であり、人事管理では同じ資格の社員グループの人数であり、生産管理では仕掛品量になります。そんなわけで、車列のモデルは、電車の運行管理モデルだけでなく、様々な渋滞モデルの原形モデルになりうることを西成先生は指摘しています。

そういえば、ゴールドラットの“ザ・ゴール”の中で、工場責任者のアレックスが息子のボーイスカウト隊の世話をしてハイキングを指揮することになったくだりを思い出しました。その隊列の中で脚の遅い少年ハービーの行進の様子から、自分の管理する生産ラインのボトルネック工程の管理方法に気付く部分です。この本の中で TOC の概念の説明として使われていて、隊列における子供の間隔は、工場の生産ラインの仕掛在庫量に相当すると述べています。

“ザ・ゴール”が発刊された頃ですからもう 10 年近く前になりますが、私も子供のハイキング隊列のモデルを作って行進のシミュレーションをして、それが生産ラインのモデルの構造と同じであることを説明していたことを思い出しました。そのモデルは、この Web の中のモデル公開ページに今もあると思います。

3. サーキットを走行する車列

3.1 マルチ・エージェント・モデル

運転手を含めた車は、決まりにしたがって自律的に動いていくことができるわけですから、西成先生はそれを自己駆動粒子と呼んでいます。このような自己駆動粒子の特性を定義して、それらの相互の拘束条件を規定すれば、モデルは出来上がりです。そのモデルに初期条件を加えてその後の挙動を求めることがシミュレーションになります。

このように複数の自己駆動粒子が相互に影響しあって駆動するモデルを、マルチ・エージェント(型)・モデルと呼んでいます。

日本では、株式会社構造計画研究所がそれ専用のソフトウェア・ツールを販売しています。しかし、適用モデルに限界があるからでしょうか、モデルをC言語で直接構築する人も多いようです。さて、ここでは、Ps Studio でこのモデルを構築してみましょう。

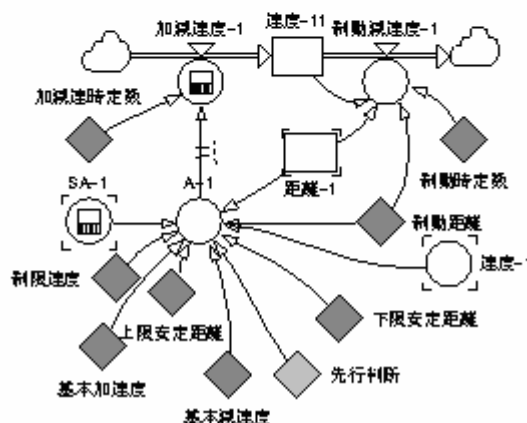
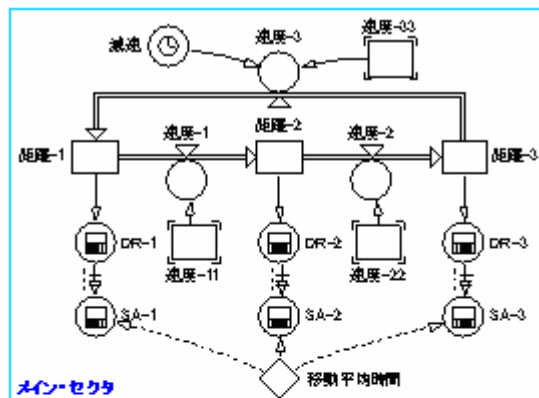
3.2 愚直な方法で作った3台の車列のモデル

まずは、車の台数を3台に限定して、その車がサーキットを回っている状態のモデルを愚直な方法で作ります。車3台の相互の関係(環境あるいは拘束条件)を表したモ

デル図 ”サーキット上の3台の車の相互の関係を表したモデル”を右上図に示します。

車の速度がフローレート(丸)で、車間距離がレベル(四角)です。車は時計回りに走行しています。

車は決まりに従って自律的に動いているわけですが、その1台目の車(エージェント)の特性モデル図 ”車の特性モデル(エージェントの特性)”を右下図に示します。



さて、このサーキットは変なサーキットで、追い抜くことはできず、また、衝突することは許されていません。その特性を文章で書くと以下となります。

エージェントの特性

(1) 速度に対する加減速特性

< 加速 (制限速度 - 5km/h) - 加減速無し (制限速度 + 10km/h) - 減速 >

(2) 前方距離に対する加減速特性

< ① - 上限安定距離 (60m) < ② - 下限安定距離 (40m) < ③ - 制動距離 (5m) - ④ >

① 基本加速度 * 2

② 加減速度 = 0km/hs

③ 前方距離が縮まりつつあるとき

基本減速度 * (下限安定距離 / 前方距離)

③ 前方距離が広がりつつあるとき

基本加速度 * (前方距離 / 下限安定距離) * 先行判断 (-1 ~ 0 ~ 1)

④ 基本減速度 * (下限安定距離 / 前方距離)

④ 緊急停止減速度 = 現在の速度 / 制動時定数 (0.5s)

(3) 基本加減速度

基本加速度 = 1km/hs 基本減速度 = -0.5km/hs

ここで加速度の単位は“km/hs”です。これは、1秒間に幾らの速度(km/h)を加減速できるかの数値を意味しています。

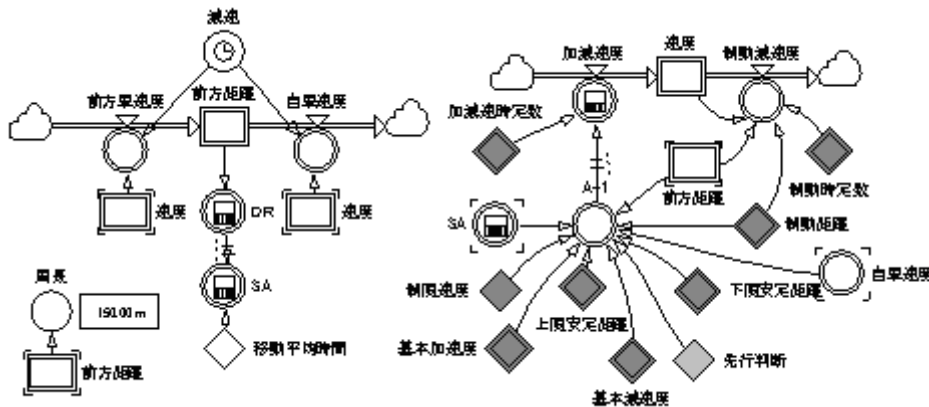
モデルの方程式は、最後の附録に記載しますので、ご参照下さい。

3.3 配列を使って作った3台の車列のモデル

愚直な方法でモデルを作ると、車エージェントの特性モデルはエージェントの個数分が必要になります。また、車の拘束条件を表す環境モデルも全てのエージェントに関する規定を平面的に表した膨大なものになります。

そこで、配列を使ってモデルを表すことで、車エージェントの特性モデルと環境モデルとをそれぞれ1台分だけで済まして下の図”サーキット上を走行する車を配列で表現したモデル”ように表現します。

左側が環境モデルで右側がエージェント・モデルです。車の台数がどんなに多くなってもこのモデルの構造は変わらず、配列の次元が増えるだけです。



3.4 サーキット上の3台の車列の挙動

今回実施したシミュレーションの外生変数を以下の表”サーキット上の3台の外生変数の一覧”に示します。

制限速度	加減速時定数	制動時定数	基本加減速度	基本減速度
30km/h	1sec	0.5sec	0.25km/hs	-0.5km/hs
上限安定距離	下限安定距離	制動距離	先行判断	移動平均時間
60m	40m	5m	-1~-1	1sec
初期速度	初期前方距離	刻み時間	計算時間	
20km/h	50m	0.1sec	3min/1h	

上表の計算時間枠に二つの数字が書かれていますが、3台の場合は3分間で、次に述べる22台の場合は1時間です。

さて、上表の数値について簡単に説明します。制限速度に対して 10km/h まではスピード違反で捕まらないだろうと判断しているドライバーは多いと思います。エージェントもそう考えています。加減速時定数と制動時定数は、共に加減速あるいは制動を掛ける場合の平均遅れ時間を意味します。

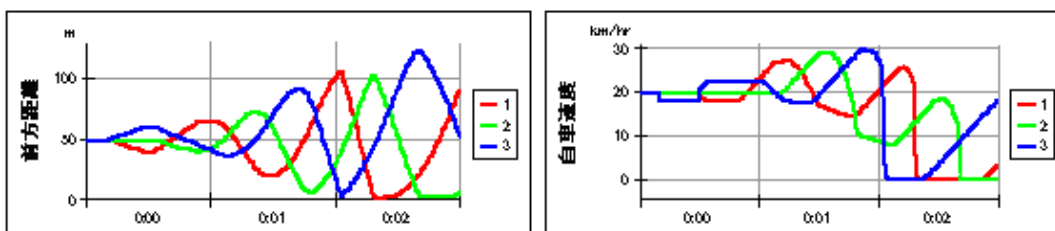
西成先生の前述の著書の中では、渋滞を起こさない条件として前方車間距離は 40m となっています。そこで、40m 以下になったら 40m を保つような運転を当初は考えましたが、これは実際の運転操作とは異なるように思います。この前方車間距離 40m から衝突を意識する 5m までの間では、単に車間距離が 40m 以下だと無条件に減速するのではなく、その間隔が詰まってきているときには減速するが、逆に、開きつつあるときには加減速しないか、または、車間距離に見合った加速をする必要があるようです。実は我々は、車間距離に対して比例制御をかけると共に、車間距離の変化、すなわち微分値に対して微分制御もかけて運転しているようです。

その状況を指定するのが、先行判断です。“-1”は、40m 以下になったら減速する、“0”は、加減速しない、“1”は、前方距離に合わせて加速する、です。“-1”は確かに用心深くて安全かも知れませんが、前方車間距離がやたらと長くなって渋滞発生の原因になります。“1”は、俊敏なドライバーを意味しています。

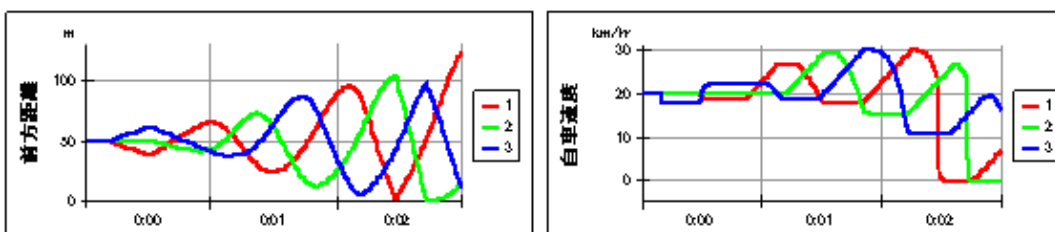
今回取り上げる全てのモデルでは、参加している全ての車が設定条件どおりに運転すれば、初期条件のまま、何時までもその状態を保った運転が続きます。そこで、サーキットの場合はどの車でも構わないのですが、今回の場合には最後の車3台目が初めに 20 秒間、2km/h だけ速度を落としたとして、その外乱が他の車に波及する様子を観察することにします。3台目の車の速度低下を下式で表します。

(STEP(-2<<km/hr>>,STARTTIME+10<<s>>))
+STEP(2<<km/hr>>,STARTTIME+30<<s>>))

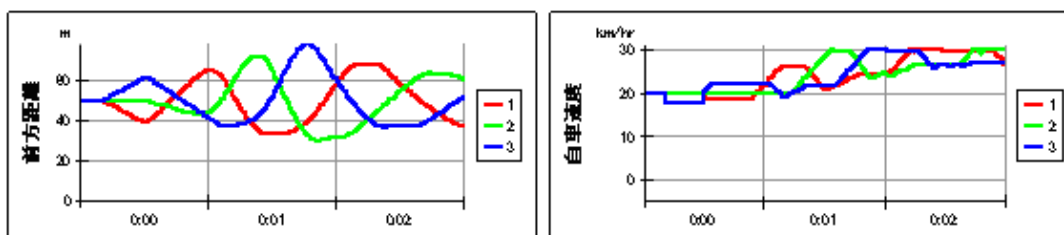
先行判断 = -1 の場合



先行判断 = 0 の場合



先行判断=1の場合



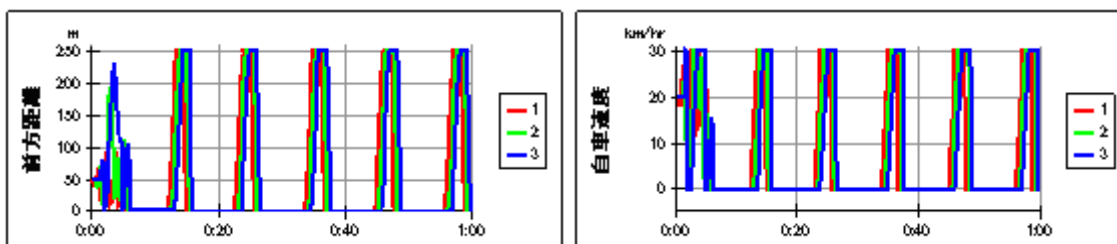
このサーキットは周長が 150m です。上図では、左が前方車間距離で、右が各車の速度です。前方車間距離が 40m から 5m の間まで近づいたときに、その間隔が開きつつあるときにも、ひたすら減速したり、加減速しないで速度をそのまま保った場合には、3 分後には車が停止する状況になっています。しかし、前方の距離に合わせて加速した場合には、それぞれの車の速度は速度制限の 30km/h まで漸近して速くなっています。このような息の詰まるような運転を日曜ドライバーが長時間続けることには無理があり、運転好きなドライバーにのみ可能なことなのかも知れません。

なお、上表では車が3台あるにも拘わらず外生変数の値は同じとしていますが、これらの値は車ごとに設定することももちろん可能です。台数が増えると入力が増え煩雑になりますから、設定には Excel からの読み込みを使うと便利だと思います。

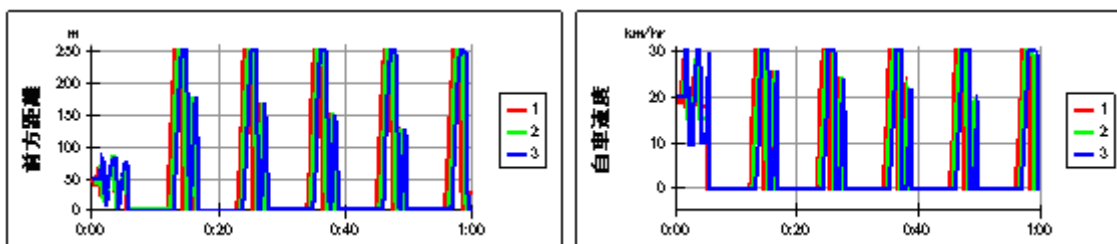
3. 4 サーキット上の 22 台の車列の挙動

西成教授の実験で、22 台の車列のサーキット実験がありましたので、それに近い計算機実験を実施してみました。計算時間は 1 時間です。

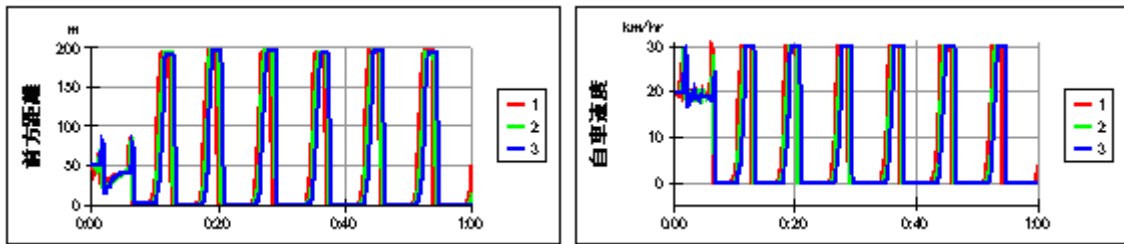
先行判断=-1の場合



先行判断=0の場合



先行判断=1の場合



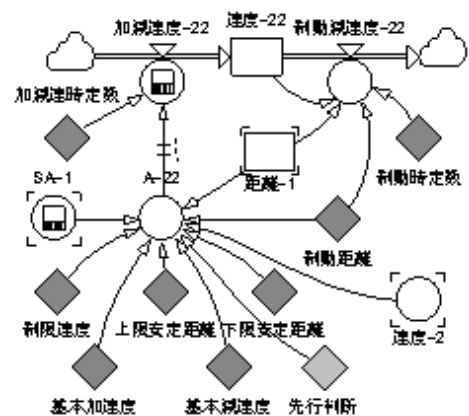
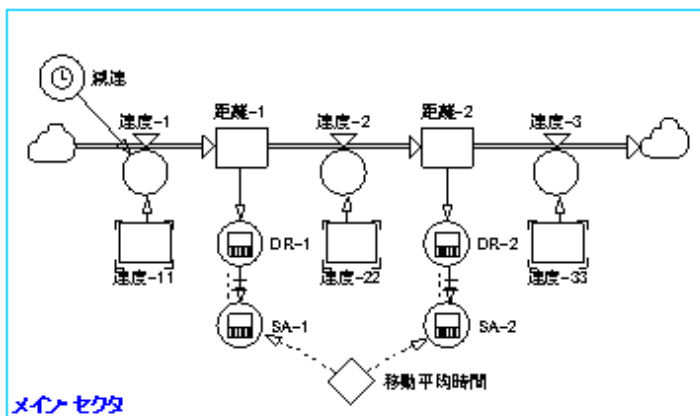
この場合の周長は 1100m です。前方車間距離が 40m から 5m の間まで近づいたときに、その間隔が開きつつあるときにも、ひたすら減速したり、加減速しないで速度をそのまま保ったり、あるいは距離に合わせて加速しても、これだけの台数になると周期的に停車を繰り返すようです。あえて言えば、距離に合わせて加速した場合には、その停車時間の割合が短くなり、最大の前方車間距離が 250m から 200m に減少する程度で、停車しないで走行することはありませんでした。これは朝の田園都市線で、どこかの駅のトラブルがきっかけとなって渋滞が始まった後の様子によく似ています。

4. 直線道路を走行する車列

4.1 直線道路を走行する3台の車列の愚直なモデル

サーキット上のモデルとの違いは車間距離の数です。先頭の車の前方車間距離はないわけですが、前方車間距離が ∞ であるとも考えることができます。車3台の相互の関係(環境あるいは拘束条件)を表したモデルが下左の図”直線道路上の3台の車の相互の関係を表したモデル”です。

車は決まりに従って自律的に動いているわけですが、1台目の車と2台目以降の車の特性モデルは異なります。ここでは2台目の車(エージェント)の特性モデルを下右の図”2台目の車の特性モデル(エージェントの特性)”に示します。



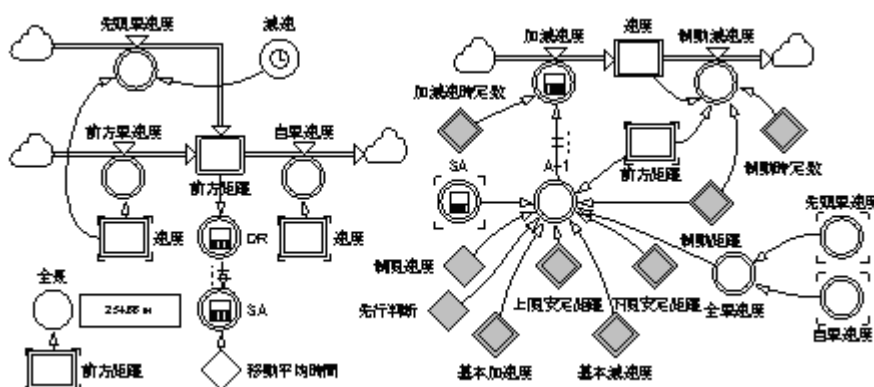
シミュレーションの外生変数を以下の表”直線道路上の3台の外生変数の一覧”に示します。

制限速度	加減速時定数	制動時定数	基本加速度	基本減速度
30km/h	1sec	0.5sec	1km/hs	-0.5km/hs
上限安定距離	下限安定距離	制動距離	先行判断	移動平均時間
60m	40m	5m	-1~1	1sec
初期速度	初期前方距離	刻み時間	計算時間	
30km/h	50m	0.1sec	10min	

4.2 配列を使って作った10台の車列のモデル

直線道路を走行する車列のモデルを、配列を使って作って、車の特性モデルと環境モデルとにより下の図”直線道路を走行する車列を配列で表現したモデル”のように表現します。

左側が環境モデルで右側がエージェント・モデルです。車の台数がどんなに多くなってもこのモデルの構造は変わらず、配列の次元が増えるだけです。ここでは、車の台数が10台、車間距離が9個の場合です。



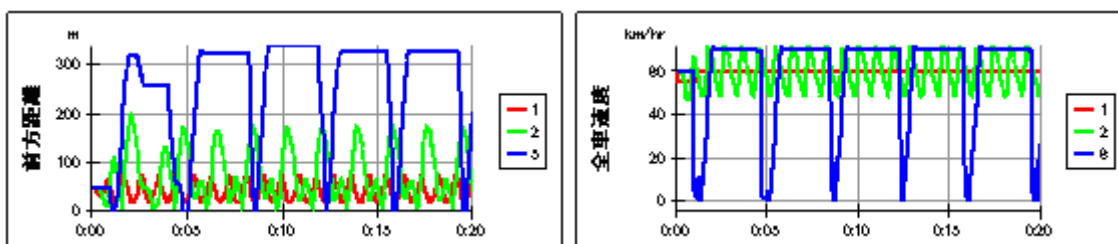
シミュレーションの外生変数を以下の表”直線道路上の10台の外生変数の一覧”に示します。

制限速度	加減速時定数	制動時定数	基本加速度	基本減速度
60km/h	1sec	0.5sec	1km/hs	-0.5km/hs
上限安定距離	下限安定距離	制動距離	先行判断	移動平均時間
60m	40m	5m	-1~1	1sec
初期速度	初期前方距離	刻み時間	計算時間	
60km/h	50m	0.02sec	20min	

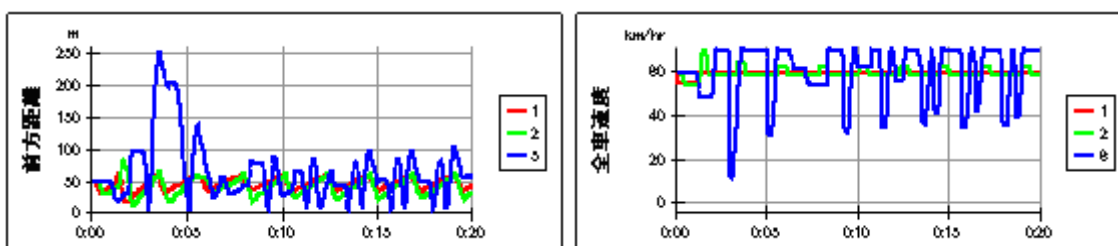
この場合も予行判断を“-1”、“0”、“1”と3種類に変えたシミュレーション結果を以下に示します。先頭車が初期状態で走行し続けると、車列の全ての車は初期状態を保って安定して走行することは、サーキット上の車列の走行の場合と同じです。そこで、先頭車が最初に1分間、5km/hだけ速度が遅くなったとしてシミュレーションを実施します。計算刻み時間は2/100秒で、計算時間は20分間です。

```
(STEP(-5<<km/hr>>,STARTTIME+10<<s>>))
+STEP(5<<km/hr>>,STARTTIME+70<<s>>))
```

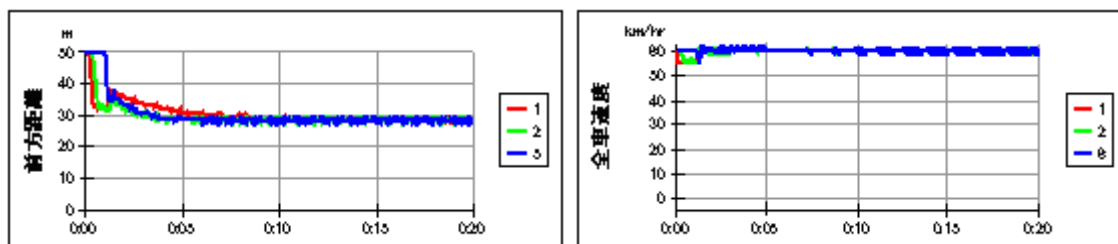
先行判断=-1の場合



先行判断=0の場合



先行判断=1の場合



大変特徴のある挙動が現れています。予行判断=-1の場合には、前の車が40m以下に近づくと減速し、その後、離れ始めても減速したままで、40m以上はなれるとそのときの速度を維持し、さらに60m以上はなれると、やおら加速して追いかける運転です。

1台目は前に車がありませんから、最初の1分間の意識的減速を除けば、一定の速度60km/hで走行しています。2台目の車は1台目の車の減速を端緒にして、速度も車間距離も振動を始めます。その振動が6台目になると停車から70km/hの振幅にまで大きくなります。

これは、サプライチェーンで現れるブルウィップ効果と同じ現象です。この現象から、ブルウィップ効果は、微分制御を加えないことによって起きる現象であることが推測できます。

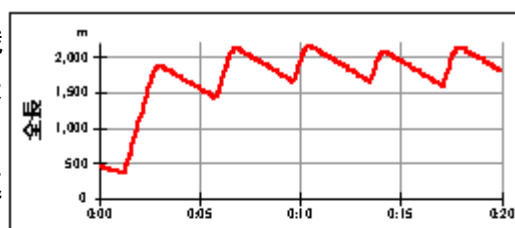
次に、先行判断=0の場合は、前の車が40m以下に近づくと減速し、その後、離れ始めるとそのときの速度を維持し、その車間距離

離が60m以上になると、やおら加速して追いかける運転です。この場合には、速度を意識的に遅くする1台目の直後の2台目はそれほど大きな変動を見せないようですが、車列の半ばの6台目になると、1台目の意識的減速の直後は大きな影響を受けませんが、その後は時間の経過と共にその振動が減衰していることが分かります。

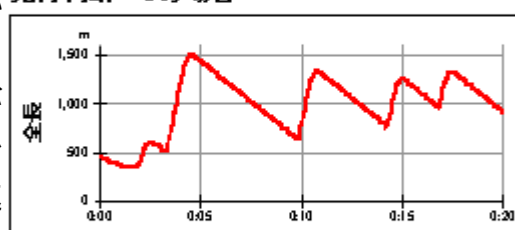
最後に、先行判断=1の場合は、前の車が40m以下に近づくと減速し、その後、離れ始めるとそのときの車間距離に合わせた加速度で加速します。端的には、前の車に付いていく運転です。運転中は神経を張り詰めているでしょうが、速度は制限速度の60km/hが維持され、車間距離は30mにまで短縮されます。

さて、10台の車の1台目から10台目までの車列の長さの変化を右図に示します。

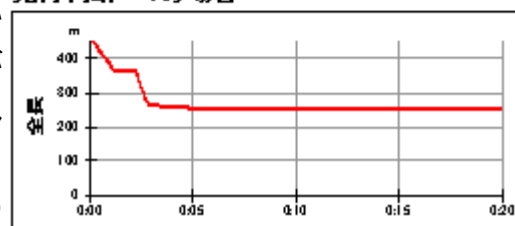
先行判断=-1の場合



先行判断=0の場合



先行判断=1の場合



我々が車を運転する場合には、いろんな運転操作をする人がいるので、今回のように全車が一樣な運転方法を実施する場合とは異なると思いますが、その場合には原則的には外乱が増えてより不安定になるものと思われます。実感する現象として、車列の中ではわけが分からないままに、車が走り始めたり止まったりすることがありますが、その様子がこのシミュレーションで再現されています。車列の長さは、尺取虫のように伸びたり縮んだりして走行しているようです。

5. 車列のモデルを使った今後の課題

車列の基本的なエージェント・モデルを構築しました。このモデルは例えば以下のような状況に関係する分析や問題解決に活用できます。

- (1) 運転手を含めた車の特性がばらついている場合に車列の挙動はどうなるか？
- (2) 運転手の心理学に基づく特性を求めて、このモデルに組み込む。

例えば、異常心理者の車列への影響はどうか？

あるいは地域(例えば、愛媛県と高知県等)における運転状況の違いが何に基づくか？

- (3) 追い越し可能な複線モデルの基本モデルとして採用する。
- (4) ロジスティック・モデルと組み合わせて、道路渋滞の不経済効果を分析する。
- (5) 道路計画・設計を含む交通行政において分析用の基本モデルの一つとする。
- (6) 鉄道やバスの走行モデルの基本モデルとして採用する。
- (7) サプライチェーンなどのマテリアル・フロー・モデルの原型とする。

6. おわりに

モデルはダウンロードもできますし、方程式も附録に記載していますからそれを参考に自ら構築することもできます。このモデルを原型として、独自の課題を解決するために応用モデルを構築してご利用下さい。

このモデルをダウンロードしてお使いになる場合に、Ps Studio のツールをお持ちの方および、フリーの Ps Studio Demo をダウンロードしてお使いになる方は、全く問題ありませんが、フリーの Ps Studio Express をダウンロードしてお使いになる方は、サーキットの 22 台モデルと、直線道路の 10 台モデルは容量の関係で走らすことができませんのでご容赦下さい。

疑問などありましたら、遠慮なくお問い合わせ下さい。

SD閑話－8 了

【附録】

A. 1 車列モデルの方程式

(1) サーキット上の愚直な3台のモデル

Name	Unit	Definition
先行判断	0	
基本減速度	km/s ²	-0.5<<km/(hr*s)>>
基本加速度	km/s ²	0.25<<km/(hr*s)>>
下限安定距離	m	40<<m>>
上限安定距離	m	60<<m>>
制限速度	km/hr	30<<km/hr>>
制動時定数	s	0.5<<s>>
制動距離	m	5<<m>>
移動平均時間	s	1<<s>>
SA-3	m/hr	SLIDINGAVERAGE('DR-3',移動平均時間)

SA-2	m/hr	SLIDINGAVERAGE('DR-2',移動平均時間)
SA-1	m/hr	SLIDINGAVERAGE('DR-1',移動平均時間)
DR-3	m/hr	DERIVN('距離-3')
DR-2	m/hr	DERIVN('距離-2')
DR-1	m/hr	DERIVN('距離-1')
加減速度-3	km/s2	DELAYINF('A-3',加減速時定数)
加減速度-2	km/s2	DELAYINF('A-2',加減速時定数)
加減速度-1	km/s2	DELAYINF('A-1',加減速時定数)
制動減速度-3	km/s2	IF('距離-3'<制動距離 OR '速度-33'<0<<km/hr>>,' 速度-33',0<<km/hr>>)/制動時定数
制動減速度-2	km/s2	IF('距離-2'<制動距離 OR '速度-22'<0<<km/hr>>,' '速度-22',0<<km/hr>>)/制動時定数
制動減速度-1	km/s2	IF('距離-1'<制動距離 OR '速度-11'<0<<km/hr>>,' '速度-11',0<<km/hr>>)/制動時定数
減速	km/hr	(STEP(-2<<km/hr>>,STARTTIME+10<<s>>) +STEP(2<<km/hr>>,STARTTIME+30<<s>>))
周長	m	'距離-1'+ '距離-2'+ '距離-3'
A-3	km/s2	IF('速度-3'>=0<<km/hr>>,(IF('速度-3'>制限速度,基本減速度, IF('距離-3'>=上限安定距離,基本加速度*2, IF('距離-3'<上限安定距離 AND '距離-3'>=下限安定距離,0<<km/(hr*s)>>, IF('距離-3'<下限安定距離 AND '距離-3'>=制動距離 AND 'SA-3'<=0<<km/hr>>, 基本減速度*(下限安定距離/距離-3'), IF('距離-3'<下限安定距離 AND '距離-3'>=制動距離 AND 'SA-3'>0<<km/hr>>, 基本加速度*(距離-3'/下限安定距離)*先行判断,基本減速度*(制動距離/距離-3')))), 0<<km/(hr*s)>>)
速度-33	km/hr	20<<km/hr>>
A-2	km/s2	IF('速度-2'>=0<<km/hr>>,(IF('速度-2'>制限速度,基本減速度, IF('距離-2'>=上限安定距離,基本加速度*2, IF('距離-2'<上限安定距離 AND '距離-2'>=下限安定距離,0<<km/(hr*s)>>, IF('距離-2'<下限安定距離 AND '距離-2'>=制動距離 AND 'SA-2'<=0<<km/hr>>, 基本減速度*(下限安定距離/距離-2'), IF('距離-2'<下限安定距離 AND '距離-2'>=制動距離 AND 'SA-2'>0<<km/hr>>, 基本加速度*(距離-2'/下限安定距離)*先行判断,基本減速度*(制動距離/距離-2')))), 0<<km/(hr*s)>>)
速度-22	km/hr	20<<km/hr>>
加減速時定数	s	1<<s>>

A-1	km/s2	IF('速度-1'>=0<<km/hr>>, (IF('速度-1'>制限速度,基本減速度, IF('距離-1'>=上限安定距離,基本加速度*2, IF('距離-1'<上限安定距離 AND '距離-1'>=下限安定距離,0<<km/(hr*s)>>, IF('距離-1'<下限安定距離 AND '距離-1'>=制動距離 AND 'SA-1'<=0<<m/hr>>, 基本減速度*(下限安定距離/距離-1'), IF('距離-1'<下限安定距離 AND '距離-1'>=制動距離 AND 'SA-1'>0<<m/hr>>, 基本加速度*(距離-1'/下限安定距離)*先行判断, 基本減速度*(制動距離/距離 -1')))))), 0<<km/(hr*s)>>)
速度-11	km/hr	20<<km/hr>>
速度-3	km/hr	MAX(0<<km/hr>>,'速度-33'+減速)
速度-2	km/hr	MAX(0<<km/hr>>,'速度-22')
速度-1	km/hr	MAX(0<<km/hr>>,'速度-11')
距離-3	m	50<<m>>
距離-2	m	50<<m>>
距離-1	m	50<<m>>

(2) サークット上の配列で表現した 22 台のモデル

Name	Dimensions	Unit	Definition
先行判断			0
減速	km/hr		(STEP(-2<<km/hr>>,STARTTIME+10<<s>>) +STEP(2<<km/hr>>,STARTTIME+30<<s>>))
制限速度	km/hr		30<<km/hr>>
移動平均時間	s		1<<s>>
周長	m		ARRSUM(前方距離)
基本減速度	daisu22 km/s2		-0.5<<km/(hr*s)>>
基本加速度	daisu22 km/s2		0.25<<km/(hr*s)>>
下限安定距離	daisu22 m		40<<m>>
上限安定距離	daisu22 m		60<<m>>
制動時定数	daisu22 s		0.5<<s>>
制動距離	daisu22 m		5<<m>>
SA	daisu22 m/hr		FOR(i=daisu22 SLIDINGAVERAGE(DR[i],移動平均時間))
DR	daisu22 m/hr		FOR(i=daisu22 DERIVN(前方距離[i]))
加減速度	daisu22 km/s2		FOR(i=daisu22 DELAYINF('A-1'[i],加減速時定数[i]))
制動減速度	daisu22 km/s2		FOR(i=daisu22 IF(前方距離[i]<制動距離[i] OR 速度[i]<0<<km/hr>>, 速度[i],0<<km/hr>>)/制動時定数[i])

加減速時定数	daisu22	s	1<<s>>
A-1	daisu22	km/s2	FOR(i=daisu22 IF(自車速度[i]>=0<<km/hr>>, (IF(自車速度[i]>制限速度,基本減速度[i], IF(前方距離[i]>=上限安定距離[i],基本加速度[i]*2, IF(前方距離[i]<上限安定距離[i] AND 前方距離[i]>=下限安定距離[i],0<<km/(hr*s)>>, IF(前方距離[i]<下限安定距離[i] AND 前方距離[i]>=制動距離[i] AND SA[i]<=0<<m/hr>>, 基本減速度[i]*(下限安定距離[i]/前方距離[i]), IF(前方距離[i]<下限安定距離[i] AND 前方距離[i]>=制動距離[i] AND SA[i]>0<<m/hr>>, 基本加速度[i]*(前方距離[i]/下限安定距離[i])*先行判断, 基本減速度[i]*(制動距離[i]/前方距離[i]))))))), 0<<km/(hr*s)>>))
速度	daisu22	km/hr	20<<km/hr>>
自車速度	daisu22	km/hr	FOR(i=daisu22 MAX(0<<km/hr>>,速度[i] +IF(i=22,減速,0<<km/hr>>))
前方車速度	daisu22	km/hr	CONCAT(FOR(i=1..1 MAX(0<<km/hr>>,速度[22]+減速)), FOR(i=2..22 MAX(0<<km/hr>>,速度[i-1])))
前方距離	daisu22	m	50<<m>>

(3) 直線道路上の配列で表現した 10 台のモデル

Name	Dimensions	Unit	Definition
先行判断			0
減速		km/hr	(STEP(-5<<km/hr>>,STARTTIME+10<<s>>) +STEP(5<<km/hr>>,STARTTIME+70<<s>>))
制限速度		km/hr	60<<km/hr>>
移動平均時間		s	1<<s>>
全長		m	ARRSUM(前方距離)
全車速度	daisu10	km/hr	CONCAT(FOR(i=1..1 先頭車速度[i]), FOR(i=2..10 自車速度[i-1]))
基本減速度	daisu10	km/s2	-0.5<<km/(hr*s)>> /*FOR(i=daisu NORMAL(1,0.3,0.02*NUMBER(i)))
基本加速度	daisu10	km/s2	1<<km/(hr*s)>> /*FOR(i=daisu NORMAL(1,0.3,0.01*NUMBER(i)))
下限安定距離	daisu10	m	40<<m>>
上限安定距離	daisu10	m	60<<m>>
制動時定数	daisu10	s	0.5<<s>>
制動距離	daisu10	m	5<<m>>
加減速度	daisu10	km/s2	FOR(i=daisu10 DELAYINF('A-1'[i],加減速時定数[i]))

制動減速度	daisu10	km/s2	CONCAT(FOR(i=1..1 0<<km/hr>>/制動時定数[i]), FOR(i=2..10 IF(前方距離[i-1]<制動距離[i] OR 速度[i]<0<<km/hr>>, 速度[i],0<<km/hr>>)/制動時定数[i]))
加減速時定数	daisu10	s	1<<s>>
A-1	daisu10	km/s2	CONCAT(FOR(i=1..1 IF(全車速度[i]>=0<<km/hr>>, IF(全車速度[i]>制限速度+10<<km/hr>>,基本減速度[i], IF(全車速度[i]<制限速度-5<<km/hr>>, 基本加速度[i],0<<km/(hr*s)>>)), 0<<km/(hr*s)>>)), FOR(i=2..10 IF(全車速度[i]>=0<<km/hr>>, (IF(全車速度[i]>制限速度+10<<km/hr>>,基本減速度[i], IF(前方距離[i-1]>=上限安定距離[i],基本加速度[i]*2, IF(前方距離[i-1]<上限安定距離[i] AND 前方距離[i-1]>=下限安定距離[i],0<<km/(hr*s)>>, IF(前方距離[i-1]<下限安定距離[i] AND 前方距離[i-1]>=制動距離[i] AND SA[i-1]<=0<<m/hr>>, 基本減速度[i]*(下限安定距離[i]/前方距離[i-1]), IF(前方距離[i-1]<下限安定距離[i] AND 前方距離[i-1]>=制動距離[i] AND SA[i-1]>0<<m/hr>>, 基本加速度[i]*(前方距離[i-1]/下限安定距離[i])*先行判断, 基本減速度[i]*(下限安定距離[i]/前方距離[i-1]))))))), 0<<km/(hr*s)>>))
速度	daisu10	km/hr	60<<km/hr>>
先頭車速度	shakan10	km/hr	CONCAT(FOR(i=1..1 速度[i]+減速), FOR(2..9 0<<km/hr>>))
SA	shakan10	m/hr	FOR(i=shakan10 SLIDINGAVERAGE(DR[i], 移動平均時間))
DR	shakan10	m/hr	FOR(i=shakan10 DERIVN(前方距離[i]))
自車速度	shakan10	km/hr	FOR(i=shakan10 MAX(0<<km/hr>>,速度[i+1]))
前方車速度	shakan10	km/hr	CONCAT(FOR(i=1..1 0<<km/hr>>), FOR(i=2..9 MAX(0<<km/hr>>,速度[i])))
前方距離	shakan10	m	50<<m>>

A. 2 車列のモデルと Free の SD ツール Ps Studio のダウンロード

説明したモデルは、以下の zip 圧縮ファイルをダウンロードして入手できます。

WinZip 他の適当なツールで解凍した後、自由に操作してお試し下さい。

現在の Ps Studio 8 以前のバージョンをお使いの方にはお詫びしないといけません。いつもは Studio 7 でモデルを作っているのですが、今回うっかりして Studio 8 で作ってしまいました。Studio 7 ではモデルを走らすことができませんので、下方で説明します Demo 版または Express 版をダウンロードしてお試し下さい。

[車列のモデルのダウンロードには、ここをクリックしてください。](#)

Ps Studio 8 の商品版をお持ちでない場合には、下方に示す方法でフリーの Ps Studio 8 Express または Ps Studio Demo をダウンロードしてお使い下さい。Express 版は、適用モデルの大きさが限定されますので、3 台の車列のシミュレーションにしか適用できません。Demo 版は、最上位のツールで全てのモデルのシミュレーションに適用できます。しかし、有効期間が限定されていて、その期間を超えると同じ PC 上に再びインストールできない制限がついていますのでご注意下さい。

評価版 Ps Studio 8 Express のダウンロード方法

SD に関して初心者の方が、SD の実用化の可能性を探ったり、SD の概要を学習するためにお使いいただく Powersim 社が提供する評価版です。

ダウンロードの方法

- ① <http://www.posy.co.jp/PS-download-f.htm> を開く。
- ② 上端の“評価版のダウンロード”をクリック。
- ③ “■評価版: Studio Express のダウンロード”にある空色のボックスをクリックすると、Powersim 社のダウンロードのページに入りますから指示に沿って進んでください。

Studio 8 Express の機能

- ▼再インストール: 繰り返し可能
- ▼機能: 商品版と全く同一機能(具体的には、その時点で最新の Professional 版)
- ▼要素数: 50 以下
- ▼有効期間: Powersim 社がメールでプロダクト・キイを送付した後60日間

評価版 Ps Studio 8 Demo のダウンロード方法

Ps Studio に関して最新の機能を確認して購入の判断にお使いいただくために、Powersim 社が提供する評価版です。

Demo 版は、商用版の最上位の Enterprise 版と同じプログラムですから、最適化とかりスク分析の機能は勿論のこと、SAP BI データベース、Oracle データベースなどにも接続できます。

ただ、制限があります。

- ・使用できる有効期間は30日間です。
- ・一度インストールすると、同じ PC 上に再インストールができません。

ダウンロードの方法

- ① <http://www.posy.co.jp/PS-download-f.htm> を開く。
- ② 上端の“評価版のダウンロード”をクリック。

- ③ “■評価版: Studio Demo のダウンロード”にあるオリーブ色のボックスをクリックすると、Powersim 社のダウンロードのページに入りますから指示に沿って進んでください。