

# 追従モデルを用いた交差点を含む交通流シミュレーション

Traffic flow simulation in an intersection with car-following model

発表者：丸目 新也

指導教官：坪井 一洋

Design of classes for traffic flow simulation in an intersection is presented, according to the object-oriented modeling. The behavior of right turning car in the intersection is modeled with the car-following model. The designed model is implemented by Java language and simulation in several conditions is performed.

## 1. はじめに

一般道において交通事故は様々な状況で発生しており、その一つに交差点内での右折車と対向車との事故があげられる。事故原因には、様々なことが考えられるが、右折車が対向車の速度と距離を誤認することが大きな原因と考える。そこで、事故を回避できる具体的な対向車の位置を求める。

個々の車両を一つの粒子として扱う追従モデルを使って交差点を含む交通流シミュレーションを行い、事故を回避するためにモデル化した挙動を用いて、事故を回避できるかを確認する。本研究ではオブジェクト指向モデリング<sup>[1]</sup>を利用した設計を行い、車両一台を一つのオブジェクトとして扱う。また、実装には Java 言語を用いる。

## 2. 追従モデル

本研究で用いる交通流のモデルは、モデルとして簡単であり、現実的なシミュレーションに応用しやすいことから、追従モデルとする。

追従モデルは、「車両の加速度は先行車の走行状態とそれに対する運転者の反応感度によって決まる」というものであり、個々の車両の状態は運動方程式により記述される。

最も単純な追従モデルは「加速度 = 感応度 × 相対速度」と考えることであり、このときの各車両の運動方程式は式(1)となる。

$$\frac{d^2 x_i}{dt^2}(t) = I_i \left\{ \frac{dx_{i-1}}{dt}(t) - \frac{dx_i}{dt}(t) \right\} \quad (1)$$

ここで、 $I_i$  は  $i$  番目の車両の感応度 (反応の強さを表す正定数で単位は [1/s]) である。式(1)より、微小時間  $t[s]$  後の  $i$  番目の車両の速度、位置をルンゲクッタ法<sup>[3]</sup>を用いて求める。

## 3. オブジェクト指向モデリング

オブジェクト指向モデリングを用いて設計した交差点を含む交通流シミュレーションのためのクラス図を図 1 に示す。交差点を含む交通流シミュレーションを 6 個のクラスを用いて実現し、これらのクラス間には図 1 の関連があると考えた。

TrafficApplet クラスは、シミュレーション全体を管理するクラスである。Car クラスは車両の挙動をモデル化したクラスであり、車両の位置や速度の計算等はこのクラスで行われる。直進走行時は追従モデルを用い、その他の挙動についてモデル化を考えた。特に、右折時の挙動については次節で概略を示す。Road クラスは、直線道路を表すクラスであり、プログラム中では、双方向リスト<sup>[4]</sup>を用いて表す。リストを用いる理由は、配列を用いる場合に比べて編集を高速に行え、多くの車両を扱うことが容易になるためである。RoadinCross クラスは、交差点内の道路を表すクラスであり、Road クラスを継承し、右左折する車両がリスト付け替え地点を通過したことをチェックする内部メソッドを追加した。CrossPoint クラスは、交差点を表すクラスであり、右折車や左折車を現在の道路から他の道路へ移動させることが主な役割である。Signal クラスは、信号を表すクラスである。

## 4. 右折時の挙動の概略

交差点内での事故原因の一つに、右折車が対向車の速度と交差点までの距離の判断を間違えることがあげられる。ここで、追従モデルは、前車の挙動に対する自車の挙動のみをモデル化しているため、対向車の影響を考えた挙動のモデル化をしなければ、シミュレーションにおいて、交差点で右折する際に衝突が起こる可能性がある。衝突を防ぐに

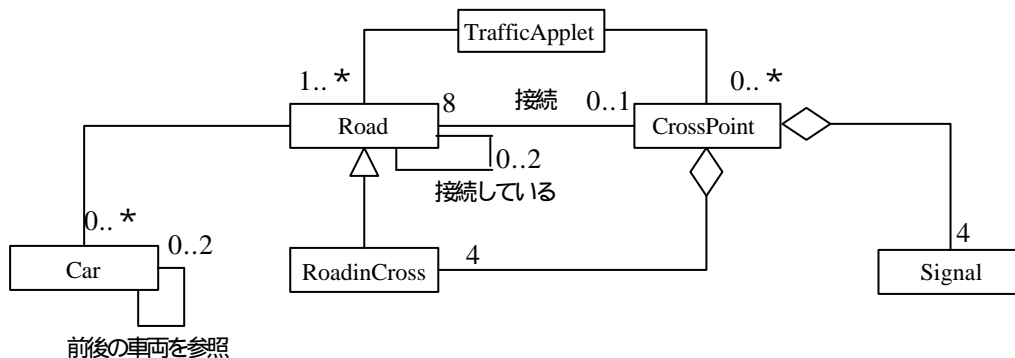


図 1 交差点を含む交通流シミュレーションのためのクラス図

は、右折車が挙動を開始する時点での対向車の速度と位置を確認し、対向車が交差点から安全な距離より離れているかを判断することで可能と考えた。対向車の安全な距離とは、右折の挙動を終えるまでに対向車が交差点に進入しない距離と考える。ここで、右折の挙動を終えるには、「右折挙動準備時間」、「交差点内走行時間」、「直線走行時間」を合計した時間がかかる。したがって、合計した時間と対向車の速度の積を対向車の安全な距離とした。

得られた式に具体的なパラメータ値を代入した対向車の安全な距離  $x_d[m]$  は次の通りである。ここで、パラメータ値として、道路幅を 6m、最高速度を 16.6m/s(約 60km/h)、曲がる時の交差点内での最高速度を 5.55m/s(約 20km/h)とした。

$v_{init} \geq v_{cmax}$  のとき

$$x_d = \begin{cases} v_{t0} \left( 2.3 + \frac{6v_{init}}{5.55 + v_{init}} \right) & (v_{t0} \geq v_{cmax}) \\ v_{cmax} \left( 2.3 + \frac{6v_{init}}{5.55 + v_{init}} \right) & (v_{t0} < v_{cmax}) \end{cases}$$

$0 < v_{init} < v_{cmax}$  のとき

$$x_d = \begin{cases} 8.9v_{t0} & (v_{t0} \geq v_{cmax}) \\ 8.9v_{cmax} & (v_{t0} < v_{cmax}) \end{cases}$$

$v_{init} = 0$  のとき

$$x_d = \begin{cases} 5.9v_{t0} & (v_{t0} \geq v_{cmax}) \\ 5.9v_{cmax} & (v_{t0} < v_{cmax}) \end{cases}$$

ここで、曲がる挙動開始時の自転車の速度を  $v_{init}$  [m/s]、対向車の速度を  $v_{t0}$  [m/s] とする。

## 5. モデル化した挙動の確認

前節でモデル化した挙動を Car クラスに実装し、シミュレーションを用いて交差点内での右折車と対向車との事故を回避できるか確認する。車両の初期配置を図 2 に示す。交差点付近での車両挙動の時間変化を図 3 に示す。

$t=16.8[s]$ の時点で対向車の位置を前節で求めた式で確認し、対向車が危険な位置にいるため信号で停止しようと減速を開始する。その後、 $t=26.0[s]$ までは、右折するには対向車が危険な位置にいるため信号地点で停止している。 $t=27.5[s]$ では、対向車がいなくなり、右折する際に危険がないために、右折を開始している。そして、 $t=30.0[s]$ では対向車と衝突することなく右折を終了し、直進走行している様子が分かる。

## 6. まとめと今後の課題

オブジェクト指向モデリングを用いて交差点を含む交通流シミュレーションのためのクラス設計を行い、それに基づいて Java プログラムを実装した。自転車の速度と対向車の速度の関係から、自転車が曲がる際の対向車の危険な位置を特定し、衝突を回避するモデルを作成できた。

今後の課題は、車両の挙動の見直し、交差点が複数存在する場合のシミュレーション、より現実的なパラメータの導入などがあげられる。

## 参考文献

- [1] 磯田定宏：オブジェクト指向モデリング、コロナ社、(1998).
- [2] 佐々木綱 他：交通工学、国民科学社、(1992).
- [3] 三井田惇朗、須田宇宙：数値計算法、森北出版、(1991).
- [4] Robert Lafore：Java で学ぶアルゴリズムとデータ構造、ソフトバンク、(1999).

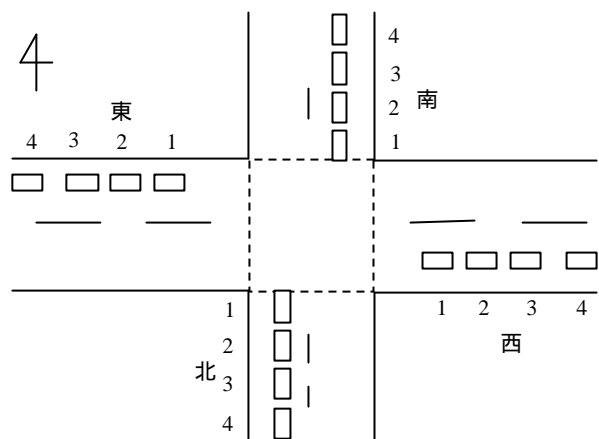


図 2 車両の初期配置

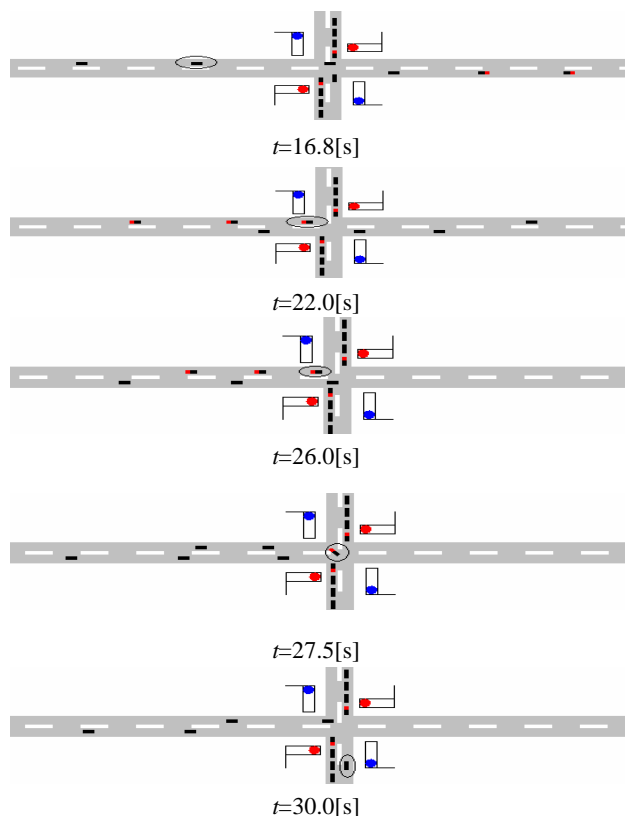


図 3 交差点付近での車両挙動