

# 駅構内における歩行者流のシミュレーション ～東京駅北通路における歩行者流制御～

山本 枝布 (沼田 一道 教授, 松浦 隆文 助教)

## 1 はじめに

### 1.1 研究背景と目的

東京駅は新幹線、在来線が多数乗り入れる中央駅であり、その構内を1日に約38万人が移動している[1]。構内は一日中混雑しているが、特に朝のラッシュ時には多方向、大量の人の流れが発生し、スムーズな移動が困難になっている。混雑はスペースの容量を超えて人が通行しようとするからであるが、通路を広げることは極めて難しく非現実的である。しかし、何らかの手段で人の流れを制御することにより混雑が緩和される可能性はある。

本研究では、東京駅の中でも比較的狭く混雑が著しい東京駅北通路を取り上げ、入出場改札機の方向設定を変更することで歩行者の流れが改善される可能性を検討する。しかし、現実に制御パターンを頻繁に変更して実験を行うのは困難である。そこで、歩行者の動きをシミュレートして、制御パターンの評価を行うことを考える。シミュレータの実装方法には様々な選択肢があるが、歩行者の移動規則やその視覚的な表示を考慮して、汎用言語で直接、専用のシミュレータを作成することにした。

本研究の目的は、実測が困難である歩行流動を視覚的に予測・分析できる専用シミュレータを設計・実装し、シミュレーションを通して歩行者のスムーズな移動に寄与する改札機の方向設定パターンを明らかにすることである。

### 1.2 研究方法

東京駅北通路(以下北通路)における歩行者流をシミュレートするために、北通路を二次元セル、歩行者の移動をセルオートマトンにより表現する。すなわち、格子状セルの上を移動する歩行者が近傍セルの状況に応じて各自の動作を決定する離散的計算モデルを考える[2]。改札機については、1台ごとに、入場専用/出場専用の設定を行う(計16ヶ所)。複数の制御パターンについてシミュレーションを行い、移動所要時間、その場に留まる回数を計測して、比較・評価する。

## 2 シミュレーションの対象

扱う対象は、歩行者を「客」、通路を「サービス」とした待ち行列システムに似ているが、本研究で問題とするのは、系外の待ち状態ではなく、系内の客の状態推移である。歩行者は、北通路への入り口である「改札口」及び「ホーム・通路連絡口」に到着し、目的とする「改札口」及び「ホーム・通路連絡口」へ向かって移動し、目的場所でシステムから退去する。

### 2.1 東京駅北通路

東京駅北通路は、東京駅の北側に位置する約78m×5mの通路である。東西に2つの改札口(図1の上が八重洲側、下が丸の内側)があり、南側には他通路との連絡口が3か所、ホームとの連絡口が5か所ある。

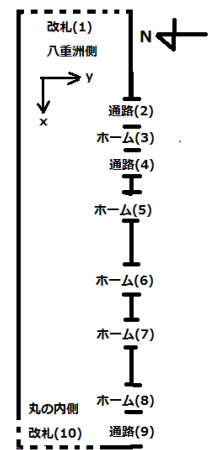


図1: 東京駅北通路モデル

### 2.2 歩行者の到着と退去

ホーム・他通路との連絡口は実状況に合わせて右側通行とする。すなわち、図1の通路に向かって右側から到着し、左側から退去する。各改札口は8個の自動改札機を有するが、実状況に合わせて、半数を出場専用、半数を入場専用を設定する(図2)。歩行者は入場専用改札機に到着し、出場専用改札機から退去する。



図2: 制御パターン例

### 3 シミュレータの概要

2節で述べた北通路と歩行者を、シミュレータで扱うためにモデル化する。北通路を  $155 \times 10$  セル (1つのセルの大きさ  $50\text{cm} \times 50\text{cm}$ ) で表す。セルの縦横のインデックスは図1の左上を原点とし、下方と右方を正方向とする2次元直交座標として扱う。また、北通路への出入り口となる各改札口、連絡口には次のような出入口番号を付与する。すなわち、八重洲改札口=1, 丸の内改札口=10, 通路連絡口=(東側から)2,4,9, ホーム連絡口=3,5,6,7, とする(図1)。北通路(系)の各セルの状態は歩行者が存在するか否かで決まる。その状態は、各歩行者が1単位時間に上下左右のセルに移動するか、現在のセルに留まるかで時々刻々変化していく。このとき、1セルには1人以下の歩行者が存在できるものとする。

シミュレータは時刻0から始まる単位時間刻みの離散的時刻  $t = 0, 1, 2 \dots$  において、①歩行者の到着・通路への進入、②歩行者の通路内の移動、③歩行者の通路からの退去を計算機内で模擬的に実現する。このとき、改札機の方向設定は歩行者の通路への進入、退出セルに反映される。同一の到着系列に対し、改札機の方向設定パターンを変化させ、各歩行者のその場に留まった回数・移動時間を計測する。

#### 3.1 歩行者の到着・通路への進入

各改札口、連絡口にランダムに到着する歩行者はシミュレーション開始時に一括して生成する。具体的には、改札口と他通路との連絡口から発生する歩行者数は、単位時間毎に到着率  $\lambda_1$  のポアソン乱数で与えられ、ホームとの連絡口では、列車の到着時刻から一定時間内に限り到着率  $\lambda_2$  のポアソン乱数に従って歩行者が発生する[3]。発生した歩行者には到着時刻、到着場所、認識番号、退去場所が与えられており、この歩行者1人1人を生起順、出入口番号順に並べたものを  $P\_LIST$  に保存する。このとき、退去場所は到着場所を除いた9か所の改札口及び連絡口の中からランダムに与える。

$P\_LIST$  を基に、各歩行者は該当時刻に到着場所(入口境界部外部)に到着後、系外の仮想的な待ちスペースに入る。この待ちスペースを待ち歩行者リスト  $W\_QUEUE\_s$  ( $s=1 \sim 10, s$  は改札口及び連絡口) とする。 $W\_QUEUE\_s$  は FIFO (First In First Out) のキューである。 $W\_QUEUE\_s$  の先頭の歩行者はいくつか存在する進入セル(入口境界部内側のセル)の中から他の歩行者が存在しない進入セルにランダムに入り、歩行を開始する。しかし、到着場所の進入セルすべてに他の歩行者が存在した場合は系内に入ることが出来ないため  $W\_QUEUE\_s$  で待機する。北通路内に進入した歩行者の認識番号は活動中歩行者集合  $APSET(t)$  に登録される。

#### 3.2 歩行者の通路内の移動

$APSET(t)$  に登録されている歩行者について、単位時間ごとにセルの位置更新を行い、移動を表現する。歩行者の移動は現在位置(現在のセルの位置)の上下左右の4セルの中から退出セル(出口境界部内側のセル)に近い順に優先順位を定め、近傍の歩行者の状態を考慮して移動する。このとき、退出セルはいくつか存在する退出セルの中から現在位置に最も近い1セルに定められており、単位時間ごとに更新を行っている。セルの位置更新の具体的な方法は、現在位置座標と退出セル座標の内積から  $\cos \theta$  を求め、値の大きさによって、上下左右の4セルの中から退出セルに近い順に第1優先セルから第3優先セルまで定める。退出セルから遠ざかる移動を防ぐために第4優先セルは求めない。 $\cos \theta$  は以下の式(2)を用いて求める。

$$\cos \theta = \frac{gy - cy}{\sqrt{(gx - cx)^2 + (gy - cy)^2}} \quad \begin{matrix} gx, gy: \text{退出セルの } x, y \text{ 座標} \\ cx, cy: \text{現在位置の } x, y \text{ 座標} \end{matrix} \quad (1)$$

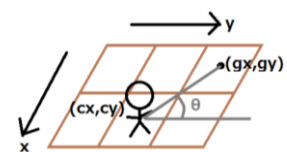


図3:  $\theta$  の場所

次に、優先順にその場所に歩行者が存在するか調べる。存在しなければ、その場所に移動し、歩行者のセルの位置を更新する。また、全てに歩行者が存在したら現在位置に留まり、セルの位置更新は行わない。このとき、評価のためにその場に留まる回数を1回カウントする。ここで、現在位置に留まるのが3回以上であれば、実際には道を譲りあうことを考慮して近傍の歩行者と現在位置の交換を行う。交換を行う近傍の歩行者は、現在位置がモデルの最も北側の場合、退出セルが現在位置の東側であれば東側、西側であれば西側の歩行者とする。また、現在位置がモデルの最も北側以外の場合は北側の歩行者とする。

### 3.3 歩行者の通路からの退去

はじめに、退去方法を述べる。ホーム・他通路への連絡口は、通行区分に従って退出し、改札口へは、実際の歩行者が直前に退出する改札口を選択することを考慮して、改札口の5セル手前の一列を仮の退去場所と設定し、仮の退去場所通過後に最も近い出場専用改札に退出する。

セルの位置更新後、各歩行者に対して北通路からの退去判定を行い、退去可能な歩行者には退去操作、退去不可能な歩行者には退出セルの更新を行う。具体的には、 $APSET(t)$ に登録されている現在位置座標と退出セル座標が等しい歩行者は退去可能であるとし、移動所要時間を調べた後にその歩行者の認識番号を $APSET(t)$ から削除する。退出不可能な歩行者は現在位置から最も近い退去場所のセルを新たな退出セルとして更新する。

## 4 シミュレーション実験

2.1で述べた北通路のモデルを用いてシミュレーションを行う。シミュレーションの時間帯は混雑が発生している平日午前8時から午前8時30分の1800秒について行う。また、1単位時間を1秒とする。対象時間帯を通勤時間とし、主な歩行者は通勤中のサラリーマンと仮定するため、目的地は既知であり、迷うことはないとする。よって、全歩行者の歩行速度は1単位時間に1セル移動とする。到着率を $\lambda_1 = 1/3, \lambda_2 = 18$ と設定し、Delphi 6を使用してシミュレータを作成し、実験を行った。

### 4.1 改札機の方向設定

16か所の改札すべてに入場専用または出場専用のどちらかを割当てたものを1パターン

表 1: 実験パラメータ

	パターン1	パターン2	パターン3(現状)
八重洲側	1 1 1 1 0 0 0 0	0 0 0 0 1 1 1 1	0 0 1 1 1 0 1 0
丸の内側	1 1 1 1 0 0 0 0	0 0 0 0 1 1 1 1	0 1 1 0 0 1 1 0

(例: 図 2)とし、割当てを変更して6パターンの実験を行った。今回は特徴がみられた3パターン(表 1)について記す。表 1では出場専用改札機を1、入場専用改札機を0として示す。また、パターン3は現状の制御パターンである。現状の設定は、駅員さんが歩行者の流れを見て勘を頼りに決めている。

### 4.2 シミュレーションの実施

シミュレーション実行中の画面を図 4に示す。また、1回のシミュレーションにかかった時間は1分30秒である。

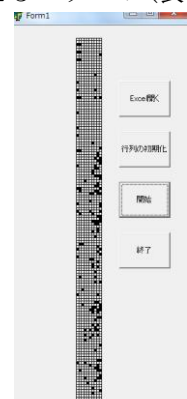


図 4: 実行中の画面

### 4.3 実験結果・考察

表 2より、全歩行者のその場に留まる回数・移動所要時間の合計はパターン3がどちらにおいて最も小さい値をとっているため、効率的な移動を行っていると考えられる。また、パターン1,2を比較すると2つの評価値の大小が一致していない。これは、パターン1は入場専用改札機が連絡口に近い図1の右方に設定されているため、移動線上に他の歩行者が多く存在し、

表 2: 全歩行者の統計

	留まる回数	移動所要時間
パターン1	6390	276395
パターン2	5100	277644
パターン3	4888	275293

留まる回数が多くなっていると考えられる。逆に、パターン2では、移動時間はかかるが、スムーズに移動する歩行者の流れが発生しているため留まる回数が少なくなっていると考えられる。

図5,6,7は到着場所・退去場所の組合せごとに歩行者1人当たりのその場に留まる回数を求め、改札機の方向設定パターンごとに表した図である。この値は、平均値の小数点第3位を四捨五入して求めた。また、縦軸をその場に留まる回数、横軸を到着場所、それぞれの線分を各退去場所とする。

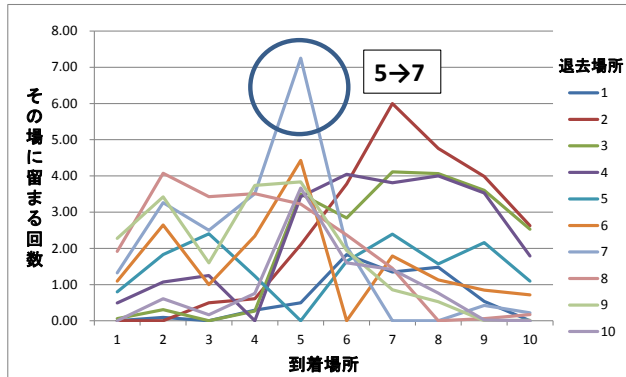


図 5：その場に留まる回数（パターン 1）

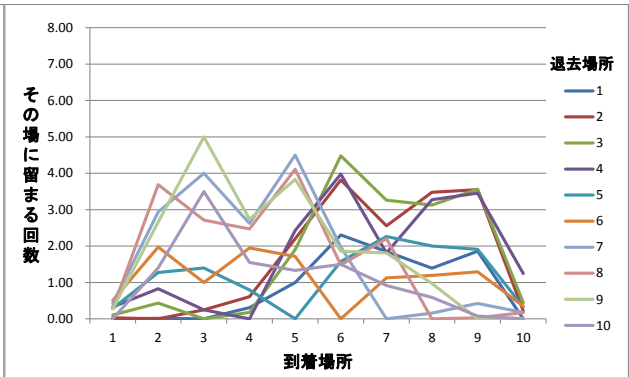


図 6：その場に留まる回数（パターン 2）

図5,7より、ホーム(5)から図1の下方へ移動する歩行者の留まる回数が他と比較して多いため、歩行快適さにバラつきが生じていると読み取れる。これは、7,8,9,10の改札口、連絡口が密集しているためと考えられる。また、図6に関して図5,7と同様の特徴がないことから、歩行快適さのバラつきは小さいと読み取れる。これは、入場専用改札機を連絡口から遠いところに設定しているため、丸の内側改札(10)付近での混雑が比較的緩和されているからと考えられる。よって、現状の制御パターンでは歩行者流全体を見ると効率的な歩行を行っているが、最も快適でない移動をしている歩行者の不満を最小にできていないことが分かった。

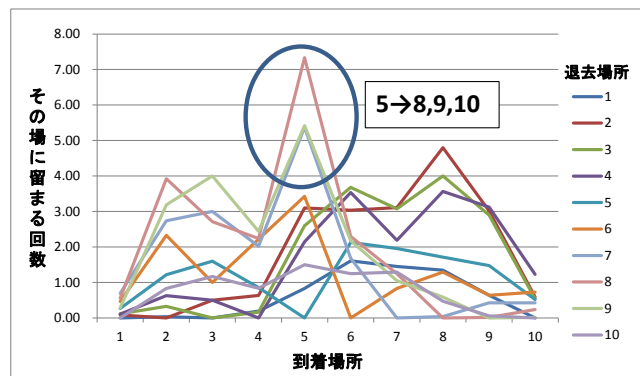


図 7：その場に留まる回数（パターン 3）

## 5 まとめと今後の課題

今回は歩行流動を予測できる専用シミュレータを作り、改札機方向設定を評価した。その結果、現状の改札機方向設定の利点・欠点を評価し、現状の制御パターンは歩行快適さのバラつきを考える上で最適でないことが分かった。そして、歩行快適さのバラつきが現状と比較して小さくなる改札機の方向設定パターンを求めることが出来た。しかし、改札機の方向設定のみで歩行者流を大幅に改善することには限界があると考えられる。今後の課題としては、より改善させるために北通路全体に通行区分を設けることなど、さらに制約を付けることが挙げられる。また、作成したシミュレータをより現実に近づけるため、退去場所の与え方を現状に即することや歩行速度の考え直しも必要になると考えられる。

## 参考文献

- [1] JR 東日本, <http://www.jreast.co.jp>(2011.12.20).
- [2] 大鑄史男, 小野木基裕(2008), セルオートマトン法による避難流動のシミュレーション, 日本オペレーションズ・リサーチ学会和文論文誌, **51**, 94-111.
- [3] 伏見正則(2004), 確率と確率過程, 朝倉書店, 138pp.