2016年1月14日

プラットホームでの酔歩シミュレーション

中央大学理工学部物理学科
香取研究室
江田和志

目的

- 研究室のテーマであるランダムウォークとは、日本語で酔歩と呼ばれることがある。一定時間にランダムな方向に動くものを数学的にモデル化したものが、あたかも酔っ払いが千鳥足をするときの動きのようであることからこう呼ばれるのであるが、では文字通り酔っ払いの動きを、私の最寄り駅のプラットホームで再現するとどうなってしまうのか、検証してみようと思った。

目的②

- 一、二次元での酔歩では再帰確率が１なのでいずれは元いた場所に帰ってこられるのだが、おそらく狭いプラットホームでは帰ってくる前に転落してしまうだろう。そこで転落しないためにはどうすればよいかも加えて考えてみようと思った。

酔歩と拡散

- シミュレーションの前に、プラットホームで酔っ払いが縦横無尽に徘徊しながらも転落しないでいられることがいかに困難であるかをイメージしやすいように、酔歩運動と拡散は物理的に等価であることを一次元の酔歩を元に説明しておく。

まず、数直線上の原点に酔った人がいるとする。その人は歩数に関係なく常に左右それぞれに等しく０．５の確率で移動する。歩幅は一定とする。

ｎ歩歩いた後で右へm歩分離れた位置にいる確率を$p(m,n)$とする。

このとき$(n+1)$歩進んだ時にmにくるためには

n歩目に$(m-1)$にいて$(n+1)$歩目に右

もしくは

n歩目に$(m+1)$にいて$(n+1)$歩目に左

に進むことになるからその確率は

$$p(m,n+1)=0.5p(m-1,n)+0.5p(m+1,n)\cdot\cdot\cdot①$$

より一般化する目的でΔtの間に$\Delta x(>0)$だけ左右共に０．５の確率で動く物体を考える。

このとき時間tの間に位置xに来る確率を$p(x,t)$とすると①の式は

$$p(x,t+\Delta t)=0.5p(x-\Delta x,t)+0.5p(x+\Delta x,t)\cdot\cdot\cdot②$$

この式を展開すると

$$\frac{\partial p(x,t)}{\partial t}=\frac{(\Delta x^2)}{2\Delta t}\frac{\partial p^2(x,t)}{\partial x^2}\cdot\cdot\cdot③$$

(この展開の途中式は次のページ)

②式の展開

②の左辺をΔｔについて一次までTaylor展開

$$p(x, t + \Delta t) \cong p(x, t) + \frac{\partial p(x, t)}{\partial t}\Delta t$$

右辺をΔxについて二次までTaylor展開

$$\frac{1}{2}p(x + \Delta x, t) \cong \frac{1}{2}p(x, t) + \frac{1}{2}\frac{\partial p(x, t)}{\partial x}\Delta x + \frac{1}{4}\frac{\partial^2 p(x, t)}{\partial x^2}\Delta x^2$$

$$\frac{1}{2}p(x - \Delta x, t) \cong \frac{1}{2}p(x, t) - \frac{1}{2}\frac{\partial p(x, t)}{\partial x}\Delta x + \frac{1}{4}\frac{\partial^2 p(x, t)}{\partial x^2}\Delta x^2$$

これらを②式に代入しまとめ、両辺をΔｔで割ると③式が導ける。

Δtの間の動きの平均二乗変位を$\langle\Delta(X_i)^2\rangle$とすると

$$\langle\Delta(X_i)^2\rangle = 0.5(\Delta x^2) + 0.5(\Delta x^2) = \Delta x^2$$

であるから、時間tの間に位置$X(t) = x$まで移動した場合の平均二乗変位は

$$\langle\{X(t) - X(0)\}^2\rangle = \frac{t}{\Delta t}\Delta x^2 = 2\frac{\Delta x^2}{2\Delta t}t$$

上の式の$\frac{\Delta x^2}{2\Delta t}$を拡散定数Dとみなすと、変位がガウス分布を持つときの平均二乗変位の式と等しくなり、③は拡散方程式となる。

9

以上より、酔歩は拡散過程と等価であるといえる。

となると、方向感覚を失い、一歩ごとの記憶も常にリセットされ続ける状態に陥った酔っ払いの動きは、水にたらしたインクのように広がっていくのか。

ただしそれを前後左右の動きを取り入れて細長いプラットホームの上で行うと、電車の向きと平行なライン(以後、 x 方向とする)と端から端まで行き渡る前にあっという間に電車の向きに垂直なライン(以後、 y 方向とする)から転落してしまうだろう、と仮定しておく。

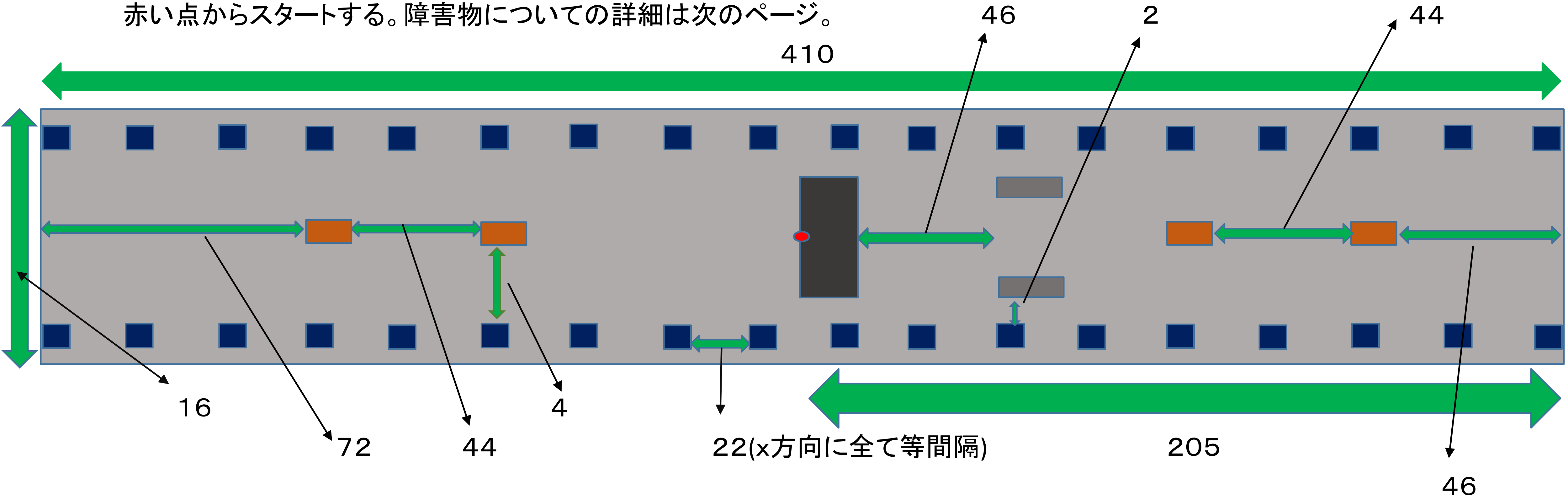
シミュレーション　条件

- 私が毎日使用している最寄り駅のプラットホームの図(かなり簡略しているが)を示した。数値は実際の距離ではなく、歩幅０．５mの人間が歩いた場合の歩数である。
- スタート地点から一秒につき一歩、前後左右に等確率で進む。
- 障害物(ベンチや階段、柱、休憩所の壁など)に当たったら一秒静止
- 転落したら(枠から外れたら)終了
- これを１００回繰り返す。

シミュレーション①　　図

この図がシミュレーションに使うプラットホームである。実際私が毎日利用している最寄り駅がモデルである。(シミュレーションしやすいようにかなり簡易化している)数値は歩数である。

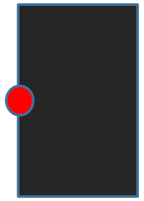
赤い点からスタートする。障害物についての詳細は次のページ。



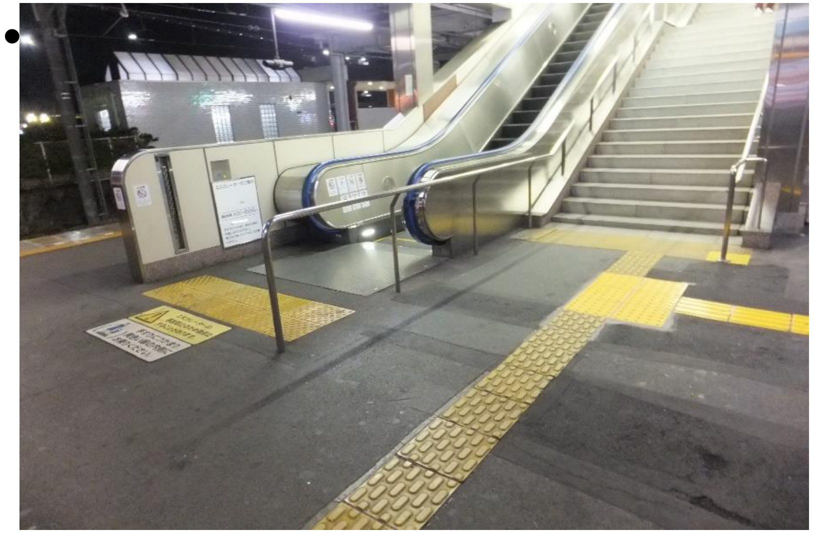
・・・柱(x方向に２２歩、y方向に１０歩の間隔でホーム内に３６個
一辺が2歩分の正方形、転落地点からの距離は１歩分)

・・・ベンチ(x・・４歩、y・・２歩分の大きさ、ホーム内に４個
x軸に対称的に存在)

・・・休憩所(双壁。それぞれx・・１０歩、y・・１歩分。壁と壁の間y隔は４歩分。x軸に対称に存在。ホーム内に１個)



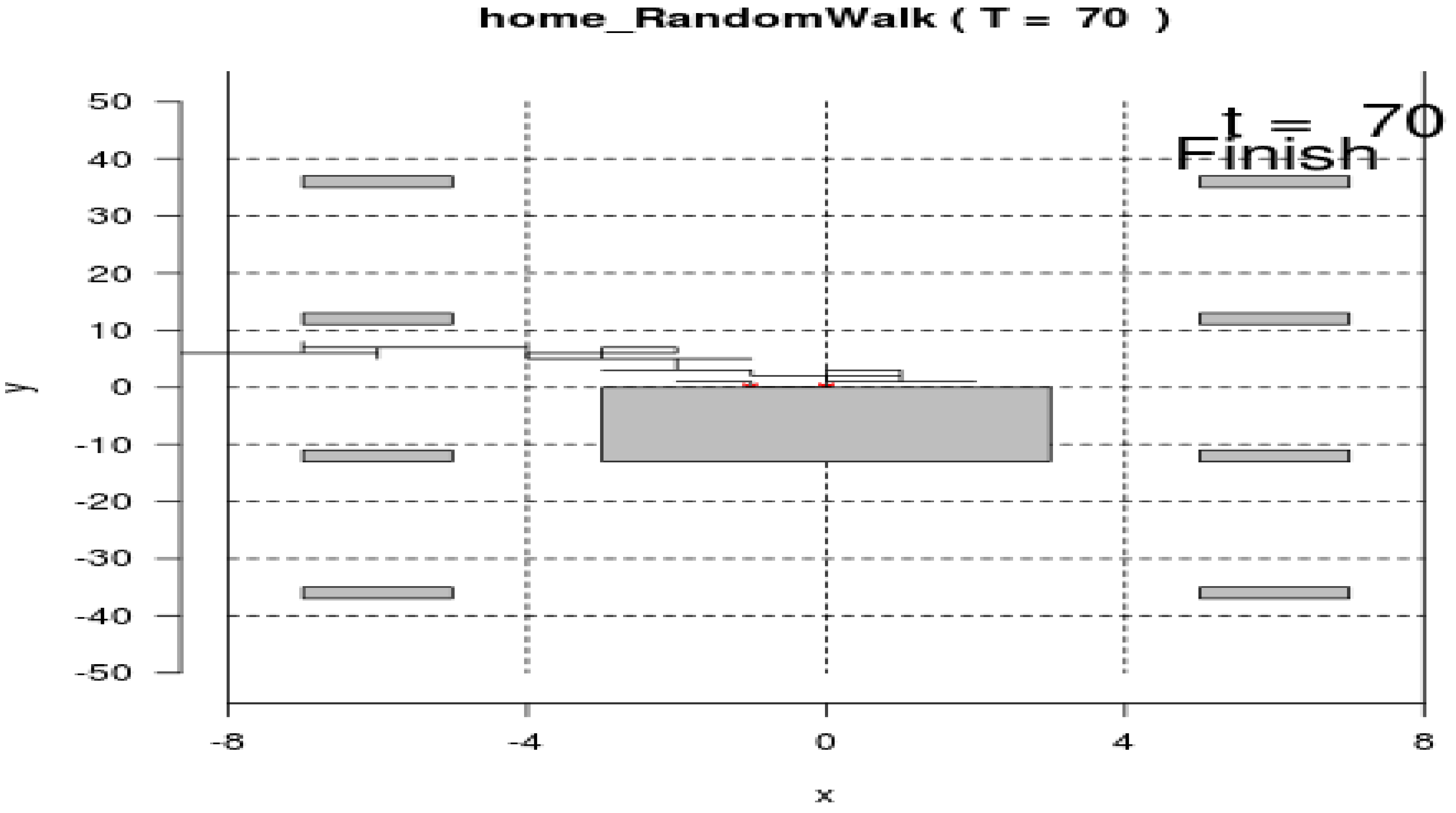
・・・階段(x・・１３歩分、y・・６歩分の大きさ、x軸に対称に存在、赤い地点がスタート場所でその場所はちょうどホームの中心地点となっている。)

-  階段

 ベンチ

-  休憩所

試しに一度シミュレーションした結果次のようになる。(ホームのｘ方向の長さを大幅に縮小している。また、縦と横を入れ替えているためｘ軸とｙ軸も入れ替わっている。)

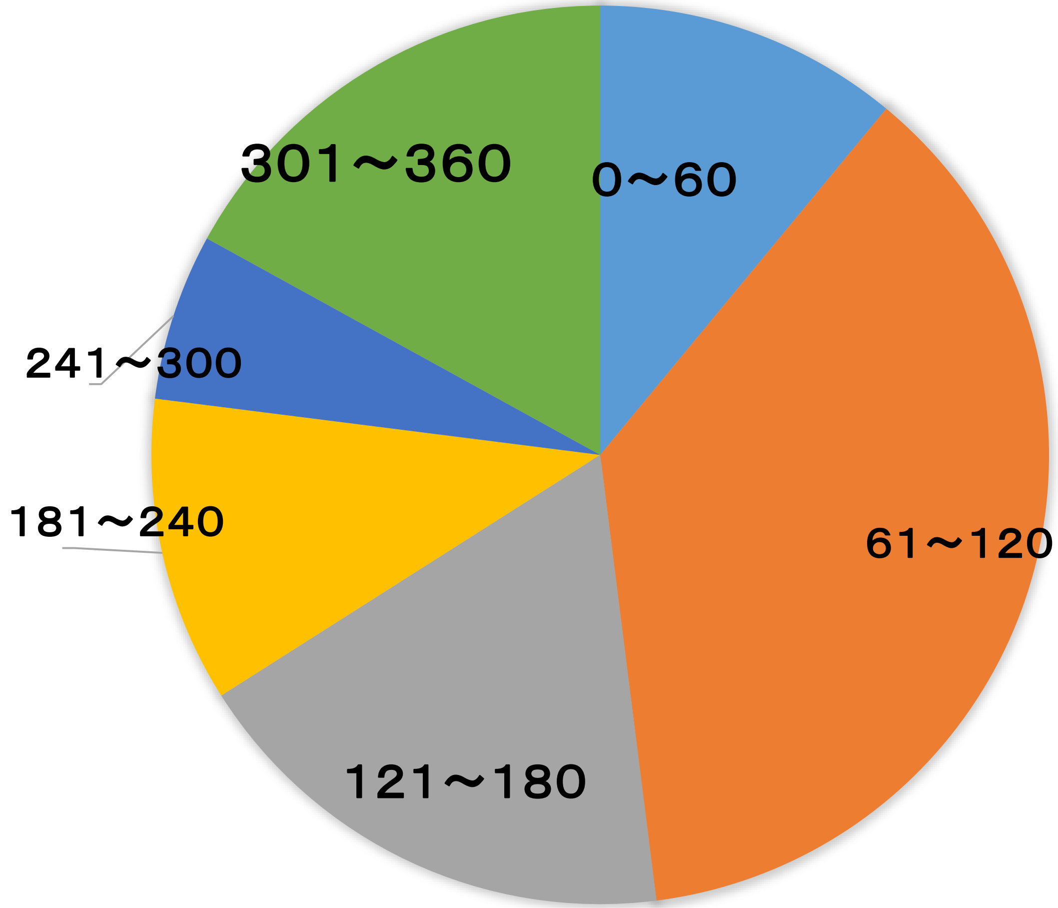

home_RandomWalk (T = 70)
t = 70
Finish
50
40
30
20
10
0
-10
-20
-30
-40
-50
y
-8
-4
0
4
8
x

では同じ試行を１００回繰り返してみる。結果は表と円グラフにまとめた。

シミュレーション① 結果

回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)
1	57	11	518	21	180	31	221	41	54	51	59	61	288	71	193	81	651	91	227
2	67	12	312	22	252	32	158	42	60	52	118	62	101	72	399	82	533	92	110
3	67	13	190	23	148	33	90	43	93	53	81	63	163	73	367	83	155	93	178
4	65	14	154	24	21	34	460	44	273	54	110	64	148	74	42	84	159	94	357
5	65	15	116	25	229	35	80	45	180	55	61	65	102	75	171	85	181	95	123
6	339	16	100	26	220	36	70	46	165	56	274	66	206	76	174	86	94	96	356
7	37	17	59	27	460	37	88	47	167	57	96	67	369	77	133	87	115	97	108
8	80	18	27	28	86	38	128	48	19	58	92	68	60	78	65	88	397	98	29
9	112	19	104	29	237	39	603	49	95	59	277	69	161	79	111	89	214	99	76
10	432	20	270	30	87	40	895	50	152	60	455	70	151	80	227	90	40	100	119

シミュレーション①



時間(秒)	確率(%)	確率の足し合わせ
～60	11	11
61～120	37	48
121～180	18	66
181～240	11	77
241～300	6	83
301～	17	100

最小値　19
最大値　895
平均値　186.18

リアルな駅の構造に近い状態でシミュレーションしてみた結果、８３%の確率で５分以内に転落してしまうことがわかった。

ちなみにモデルとなっている駅の実際の時刻表を見る限り、酔っ払いが増えるであろう２０時以降は終電まで大抵５分以内に上下線どちらかがやってくるが、５分以上上下線ともに電車が来ない状態が１８回ある。

つまり上下線どちらに乗っても構わないとしても運が悪い(その１８回のうち、前の電車が出発したと同時にスタート地点にやってくる)と少なくとも８３%の人間は危険な目にあってしまう。

電車の本数を増やすとしても、およそ半分の確率で２分以内に転落しているのであまり効果は期待できないだろう。

シミュレーション①の課題

まず障害物のリアルな位置関係をできる限り崩さずシミュレーションしてみたが、ほぼかすりもせず転落してしまうことが分かった。特にｘ方向に進まな過ぎて、ｘ方向に遠い位置にある障害物がまるで意味をなしてくれなかった。

ちなみに、障害物の有効な置き場を調べるために障害物なしのプラットホームでシミュレーションすると平均持続時間は１６３秒間だった。そしてその１６３秒間、ｘ方向の平均二乗変位はたったの９.０１だった。

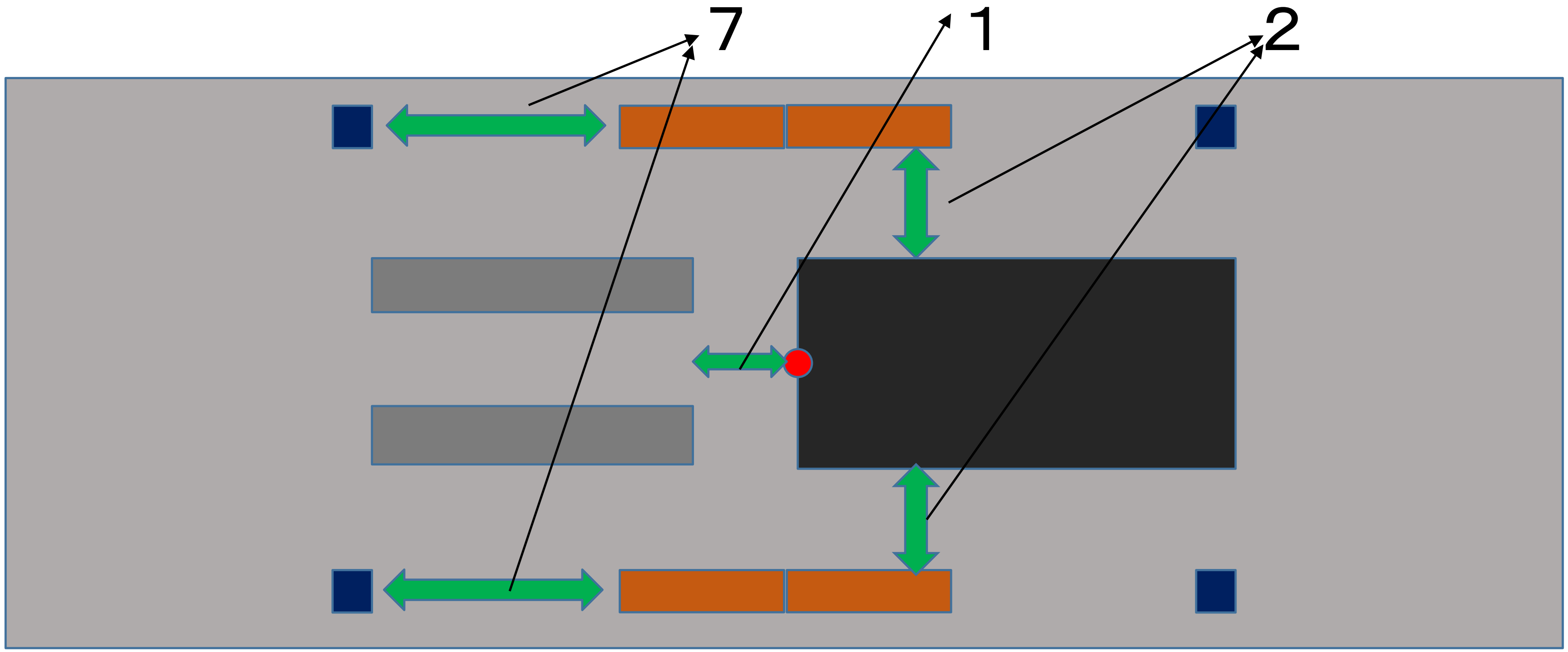
X方向への平均二乗変位

回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)
1	8. 8	11	10. 0	21	9. 0	31	9. 4	41	8. 6	51	8. 6	61	9. 6	71	10. 8	81	8. 6	91	9. 2
2	9. 0	12	8. 6	22	9. 1	32	8. 9	42	9. 4	52	9. 0	62	8. 7	72	9. 1	82	8. 6	92	9. 3
3	9. 1	13	8. 9	23	8. 5	33	9. 1	43	10. 2	53	9. 5	63	9. 5	73	9. 3	83	9. 1	93	11. 0
4	9. 0	14	9. 9	24	9. 1	34	8. 4	44	9. 1	54	8. 0	64	9. 6	74	9. 2	84	9. 0	94	8. 9
5	8. 2	15	8. 4	25	8. 9	35	8. 6	45	7. 9	55	9. 0	65	7. 6	75	8. 4	85	8. 2	95	9. 6
6	8. 5	16	9. 4	26	9. 9	36	9. 1	46	8. 6	56	8. 7	66	7. 8	76	9. 8	86	9. 0	96	8. 6
7	9. 1	17	8. 5	27	8. 9	37	8. 5	47	8. 9	57	9. 3	67	9. 0	77	7. 8	87	7. 6	97	8. 5
8	8. 8	18	8. 6	28	8. 7	38	8. 9	48	9. 5	58	10. 7	68	9. 1	78	9. 4	88	9. 5	98	9. 1
9	9. 5	19	9. 0	29	9. 4	39	8. 2	49	9. 6	59	8. 6	69	9. 1	79	8. 9	89	9. 0	99	9. 1
10	9. 2	20	9. 7	30	9. 5	40	10. 0	50	8. 3	60	8. 5	70	8. 3	80	9. 4	90	9. 1	100	9. 4

ということで、多少強引ではあるが酔っ払いを救うために全ての障害物をこの９．０１の範囲(しかもスタートする向き)に設置してやり直してみる。

シミュレーション② 図

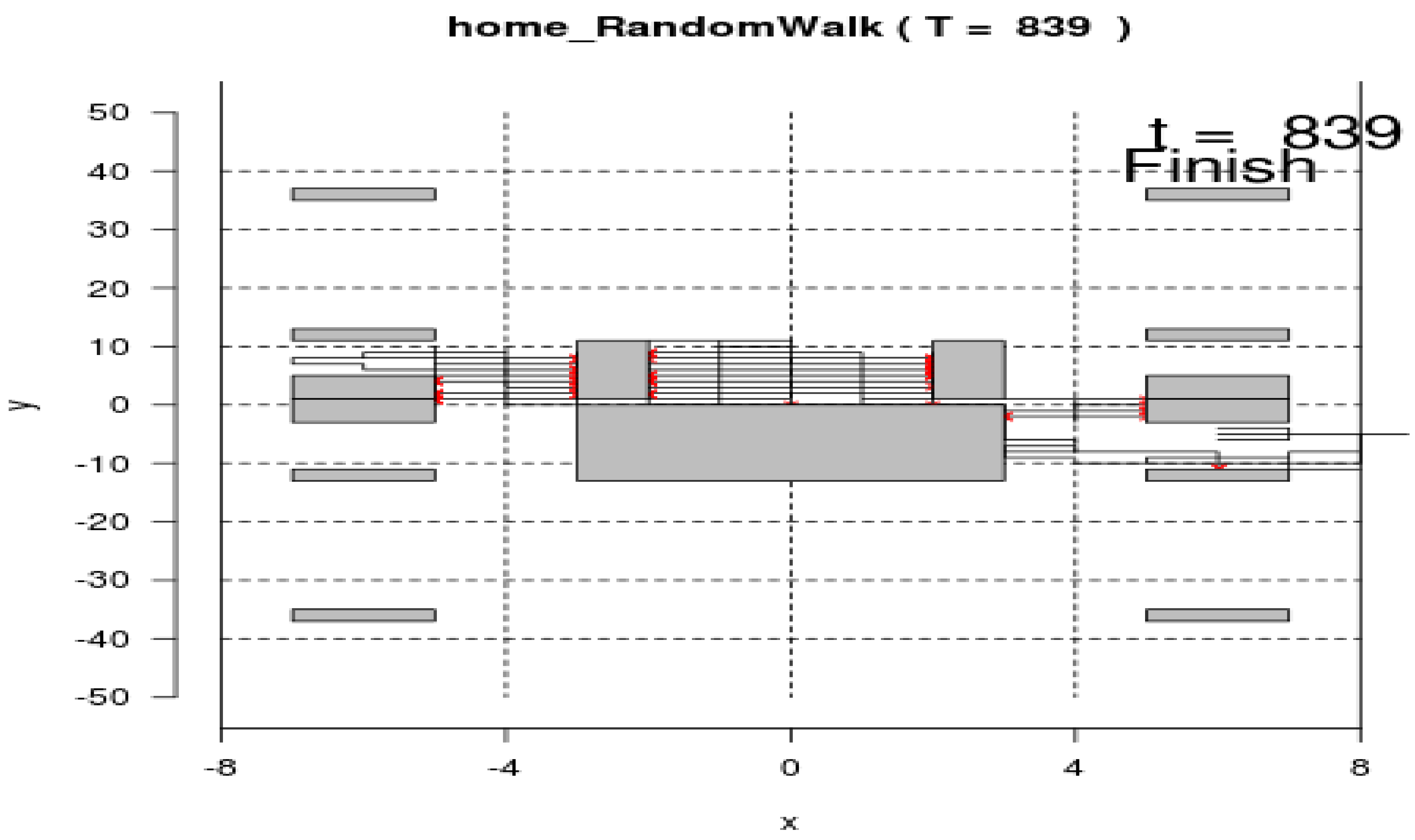
強引と言われるのを覚悟し、ベンチ、休憩所を全て元の位置から取り外し、平均二乗変位の結果を元にスタート地点付近に設置し直した。



この図はスタート地点からx方向にプラスマイナス20歩分のみに省略している。動かしたのはベンチ、休憩所のみ。もちろん障害物それぞれの大きさは全くかわっていない。

この図で試しに一度シミュレーションすると次のようになる。

(またｘ軸とｙ軸を入れ替えて表示)



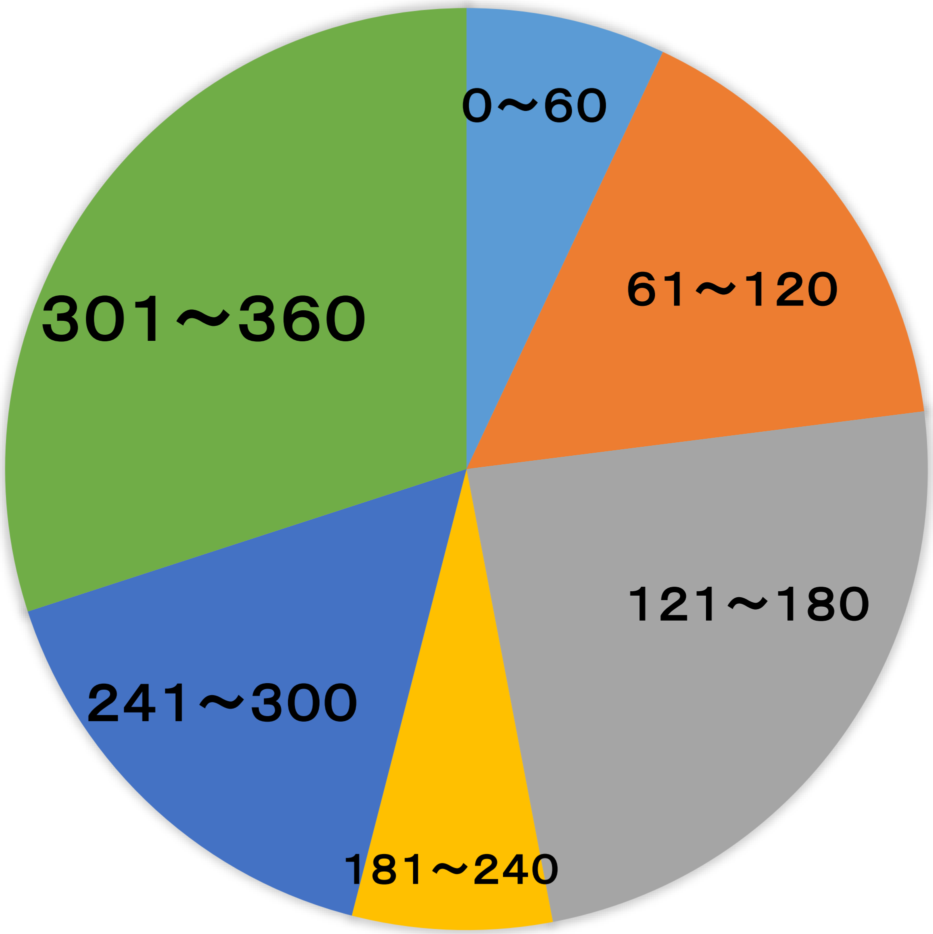
home_RandomWalk (T = 839)
t = 839
Finish
50
40
30
20
10
0
-10
-20
-30
-40
-50
y
-8
-4
0
4
8
x

同じことを１００回試行し、シミュレーション①と同様に表と円グラフにまとめた。

シミュレーション② 結果

回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)	回数	時間(秒)
1	108	11	156	21	28	31	55	41	164	51	647	61	81	71	265	81	149	91	274
2	103	12	751	22	115	32	156	42	73	52	45	62	266	72	78	82	238	92	685
3	417	13	96	23	63	33	359	43	104	53	911	63	48	73	184	83	185	93	567
4	542	14	459	24	63	34	131	44	516	54	116	64	165	74	235	84	177	94	464
5	893	15	168	25	70	35	165	45	474	55	240	65	37	75	265	85	294	95	151
6	83	16	262	26	154	36	300	46	270	56	118	66	128	76	157	86	367	96	124
7	168	17	149	27	404	37	705	47	137	57	407	67	296	77	600	87	407	97	547
8	498	18	272	28	64	38	124	48	264	58	247	68	845	78	317	88	166	98	581
9	246	19	311	29	106	39	427	49	143	59	291	69	208	79	133	89	422	99	59
10	216	20	282	30	128	40	485	50	59	60	268	70	893	80	218	90	520	100	133

シミュレーション②



時間(秒)	確率(%)	確率の足し合わせ
～60	7	7
61～120	16	23
121～180	24	47
181～240	7	54
241～300	16	70
301～	30	100

最小値　28
最大値　911
平均値　277.63

転落するまでに通るであろう位置に障害物を持ってくるだけでも５分以上転落しない確率が３０％に増えた。何より２分もたず転落してしまう確率が２３%にまで減少しているため、電車の本数を増やすことによる効果が大いに期待できるだろう。

まとめ

拡散と同じような動きをする最悪な酔っ払いを救うにはホームの幅とほぼ同じ長さの範囲内にホーム上の置物を集中させるのが手っ取り早いことが分かった。

もちろん階段の目の前の休憩所や真横のベンチをすり抜けて１分もたず転落してしまうパターンもわずかにあったが、完全に事故を防ぐためには障害物を増やすか転落防止用の自動ドアを設置するしかないだろう。

この研究を進めるにあたり、適切な助言をしてくださった香取眞理教授や、共に議論してくれた研究室のメンバーに感謝します。

参考文献　　「ブラウン運動」米沢冨美子

シミュレーション協力　白木俊平(香取研究室)

駅のモデル　東武東上線　高坂駅　(埼玉県東松山市)