

平成26年度 特別研究報告書

知的制御によるエレベータの  
利便性と省エネ性能の向上の検討

龍谷大学 工学部 情報メディア学科

T110426 佐伯啓太

指導教員 三好 力 教授

## 内容概要

近年、都市部には超高層マンションが、10階建てのマンションなども地方には次々と建設され、いずれも人気が高い。マンションの多くはエントランスや自室の施錠においてセキュリティ認証をおこなうことで、マンション共用部にも関係者以外の侵入を防いでいる。また、マンションには必ずエレベータが存在し、各階のエレベータ前に設置されたボタンを押すことでその階へ呼び出すことが出来る。しかし、認証機能とエレベータが連動するシステムは少ない。本研究では、エントランスや自室での認証があった場合、エレベータと連動して認証のあった階へ予めエレベータを呼び出すシステムを考え、エレベータ待ち時間の減少などの利便性とエレベータの移動距離による省エネ性能の向上について検討した。

# 目次

謝辞.....	1
第1章 はじめに.....	1
第2章 基本事項と既存技術.....	2
2.1 エレベータの駆動方式[1].....	2
2.1.1 ロープ式.....	2
2.1.2 油圧式.....	5
2.1.3 その他の駆動方式.....	7
2.1.4 リニアモーター式の将来.....	7
2.2 電力中央研究所報告 [2] .....	7
2.3 丸紅 ホームギャラリー[3].....	7
2.4 ITセキュリティ[4][5].....	8
2.4.1 ITセキュリティとは.....	8
2.4.2 ITセキュリティ対策.....	8
第3章 提案手法.....	9
第4章 実験と評価.....	10
4.1 実験目的.....	10
4.2 実験環境.....	10
4.3 実験内容.....	10
4.4 実験結果.....	11
4.5 考察.....	17
第5章 まとめ.....	18

謝辞

参考文献

付録

## 第1章 はじめに

近年、都心部には超高層マンションが次々と建設され、いずれも人気が高い。超高層マンションだけでなく、10階建てのマンションなども地方には次々と建設されていっている。それぞれのマンションには必ずエレベータが存在します。マンションは一軒家とは違い、一つのビルで複数の家庭があり、上層階に住んでいる人は出かける際にはエレベータを使うことが必須となっている。従来のエレベータはエレベータの入り口付近に[上]、[下]のボタンがあり、そのボタンが押されるとエレベータはボタンが押された階へ移動する。

セキュリティとITが融合したITセキュリティは今後のセキュリティの主流になっていくと考えられる。ITセキュリティとエレベータが連動したマンションはまだ少ないが徐々に増えてきている。マンションにおいてITセキュリティの主な使用場面はマンションのエントランスにある。マンションのエントランスにて住居者以外の侵入を制限するためにICカードをかざすことで認証を行い、その後エントランスの解錠とともにエレベータを呼び寄せるセキュリティだ。呼び寄せたあとエレベータにてICカードを再びかざすことでエレベータが利用可能になり、そうすることで住居者以外の人利用を制限することが出来、安心と快適を実現している。

しかし、まだまだ改善すべき点はいくつかある。その改善出来るポイントの一つとして待ち時間の削減がある。ここでいう待ち時間とは、エレベータの利用者が[上][下]ボタンを押してからエレベータに乗るまでの時間を指す。つまりエレベータが来るまでの待ち時間を短縮できないか考えた。1階からエレベータに乗るときはエントランスにてICカードで解錠とともにエレベータの呼び出しが可能なのですでに待ち時間は削減されている。ITセキュリティにより、住戸玄関もICカードで施錠、解錠が出来るので、住戸玄関からもエレベータを呼び出せないか考えた。従来のシステムと提案したシステムとで待ち時間の評価を比較して、提案したシステムが効果的かどうか検討する研究を行う。

## 第2章 基本事項と既存技術

### 2.1 エレベータの駆動方式[1]

上下に移動させる駆動方式はロープ式が主流。1984年頃に建てられたマンションでは油圧式も見られる。ロープ式の初期段階は屋上に機械室があったが、2000年以降は20回程度以下の建物では機械室無しのタイプが普及している。少数派の方式としてリニア式、ベルト式、水圧式、巻胴式がある。

#### 2.1.1 ロープ式

かつてエレベータは蒸気機関や燃焼機関などで直接駆動されていましたが、現在ではほとんどが電動モーターによって駆動されている。また駆動方式は「ロープ式」が主流で、釣合おもりを使用した「トランクシオン式」と、巻胴(ドラム)にロープを巻き付ける「巻胴式」にわけることができる。

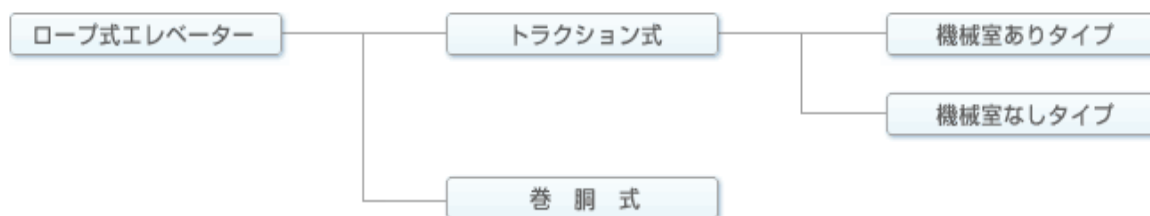


図1: 駆動方式

トランクシオン式の機械室ありタイプは、「かご」と「釣合おもり」の重量をバランスさせ、上部に取り付けた巻上機で効率よく駆動する、エレベータのもっとも基本的なタイプだ。システム構成も簡単で、低層ビルから超高層ビルまであらゆるシーンで活躍している。

ロープ式の機械室なしタイプは、機械室が不要になり、建築上部の突出物がないので北側斜線制限・日撮規制の影響がなく、また、建築上部に荷重がかからず昇降路を自由に設計できるタイプだ。巻上機の設置場所は、巻上機を上部に設けるタイプと下部に設けるタイプがある。また調速機が作動して、エレベータ制御が行われた場合の復帰操作は必ず手動で行われている。

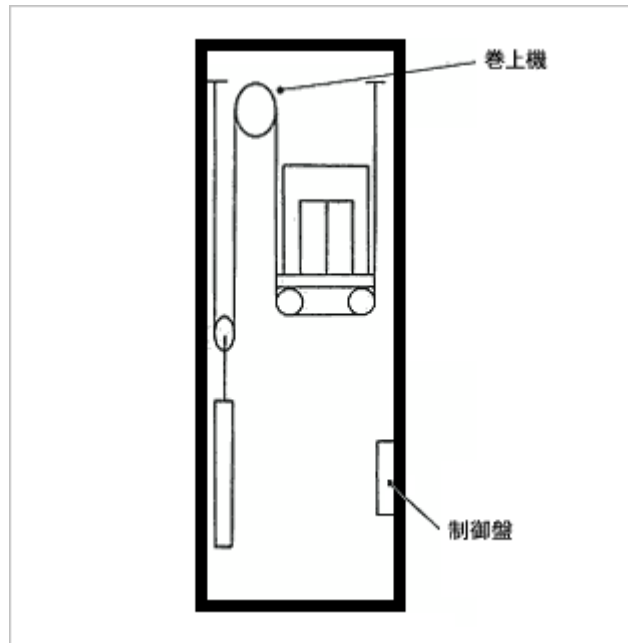


図2: 巻上機が上に設置されたエレベータの概念図

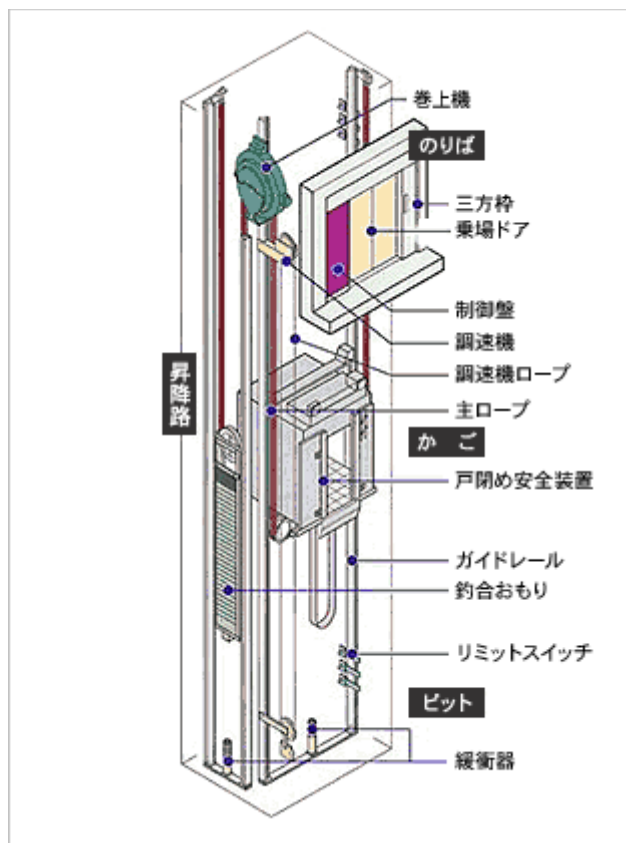


図3: 巻上機が上に設置されたエレベータの構造図

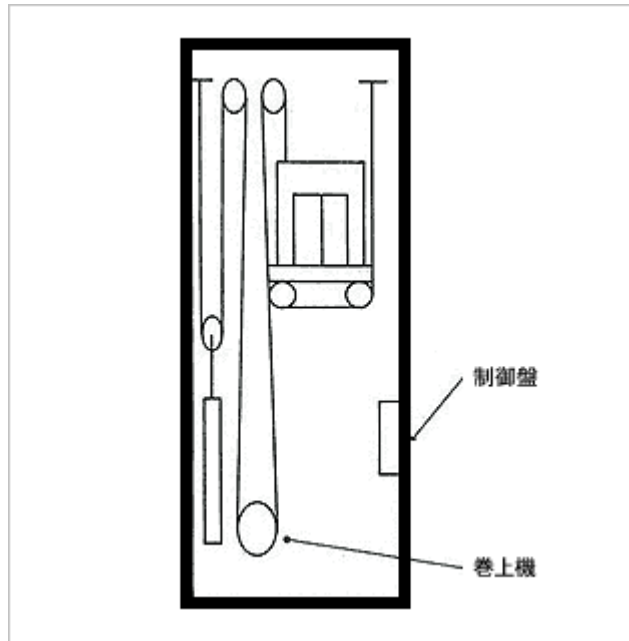


図4: 巻上機が下に設置されたエレベータの概念図

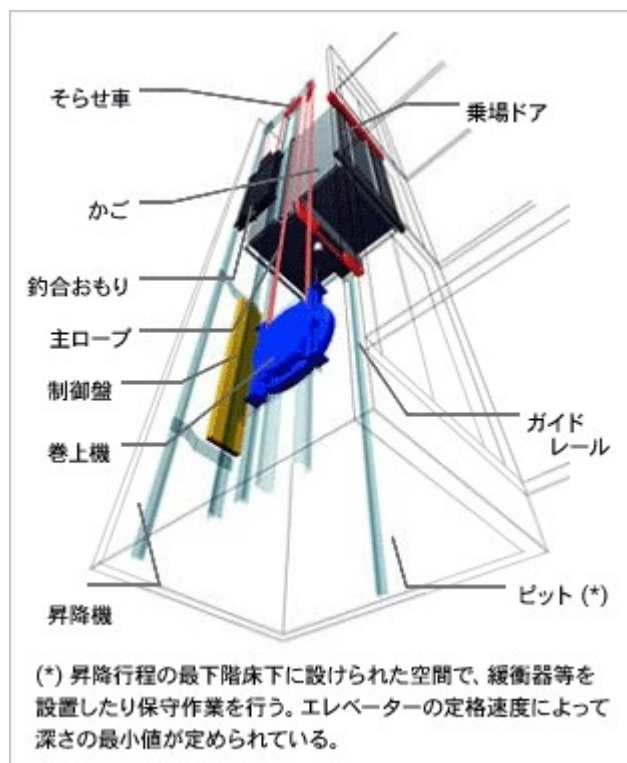


図5: 巻上機が下に設置されたエレベータの構造図

## 2.1.2 油圧式

低層用エレベータには電動ポンプで油圧を制御し、その圧力でかごを昇降させるものがある。この油圧式エレベータの駆動方式は、「直接式」「関節式」「パンタグラフ式」に分類することができる。

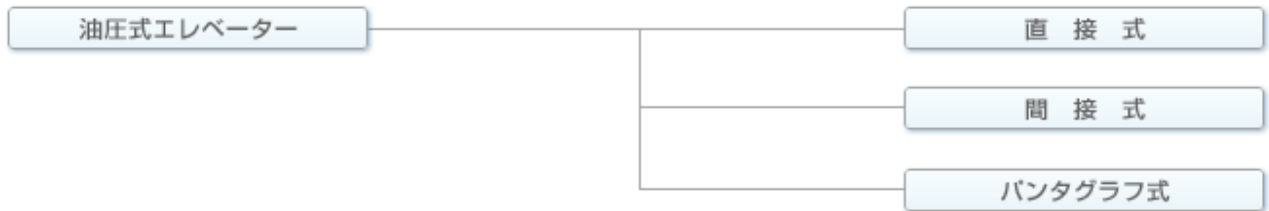


図6:油圧式エレベータの駆動方式

- 直接式  
油圧シリンダー内のプランジャー(上下する部分)にエレベータのかごを直結し昇降を行う。

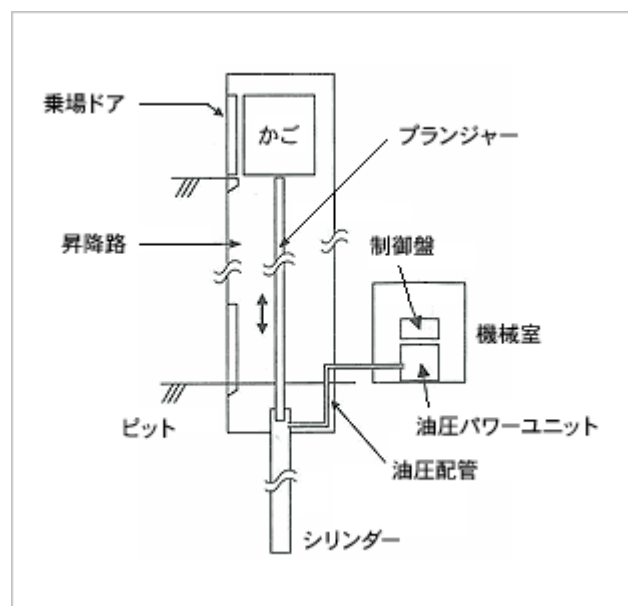


図7:直接式の概念図



- 関節式

ブランジャーの動きをロープや鎖を介して間接的にエレベータのかごに伝え、昇降を行う。また、调速機が作動して、エレベータ制御が行われた場合の復帰操作は必ず手動で行う。

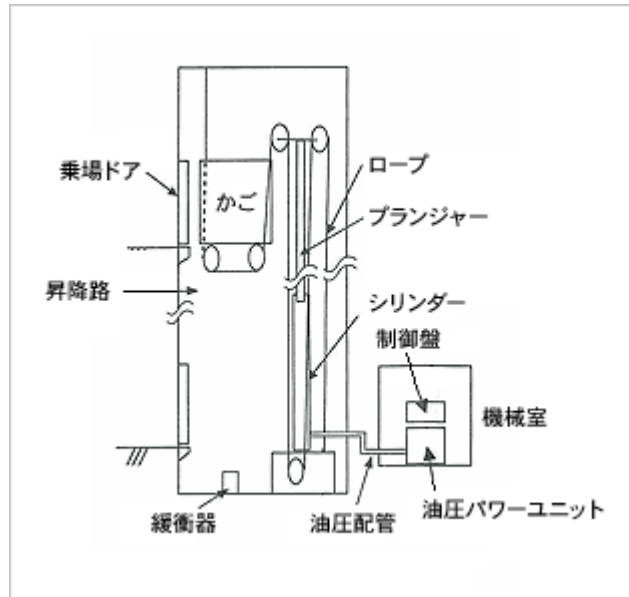


図8: 関節式の概念図

- パンタグラフ式

アームと油圧ジャッキにより、アーム頂部にとりつけたかごの昇降を行う。

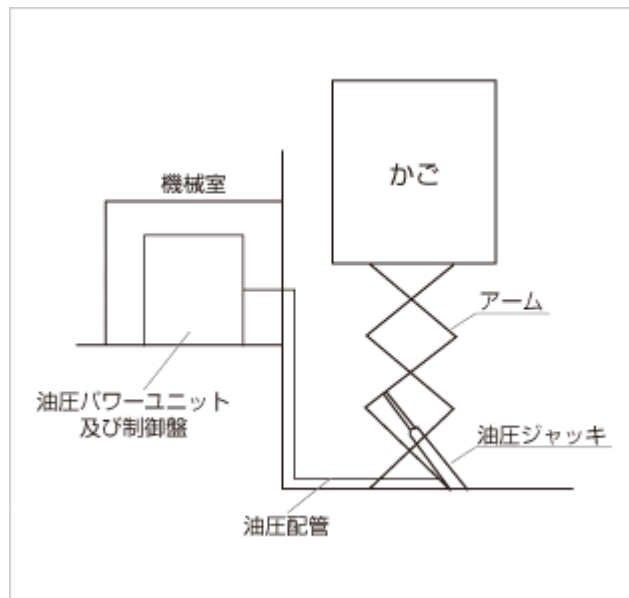


図9: パンタグラフ式

### 2.1.3 その他の駆動方式

快適性能の向上や動力効率の追求、また環境対応などを目指して研究が進められている新しい駆動方式。そのいくつかは、すでに実現されている。リニアモーター式は回転運動を直接運動に置き換えるリニアモーターを利用したエレベータだ。一次側を釣合おもりに内蔵し、二次側を昇降路の全長に伸ばすことで、巻上機を設置する必要がなくなった。低層用エレベータなどの使用が見られる油圧式ですが、よりコンパクトに設置でき、また環境にも配慮した「水」で駆動するエレベータだ。

### 2.1.4 リニアモーター式の将来

現在のリニアモーター式は「究極の機械室なしタイプ」だが、次世代のリニアモーター式エレベータとして、釣合おもりに内蔵されている一次側を、かごに内蔵する研究が進められている。この方式が実現すれば、昇降路内に伸ばした二次側にそって一次側の「かご」が移動することになるので、ロープが不要になるほか、水平・垂直・カーブなど、かごの移動方向も制約がなくなり、エレベータの概念が根底から覆ることになる。

## 2.2 電力中央研究所報告 [2]

モデルビルに対してエレベータの消費電力、利用者数、待ち時間を調査し、エレベータの台数を変更することによってどのように変化があるのかを評価している。モデルビルは6階建て、従業員130人にエレベータを2台備えるオフィスビルにおいて1台のみ運転した場合と2台とも運転した場合とで計測している。主な成果としては、計測したビルにおいてエレベータの運転台数を2台から1台に変更すると電力消費量の削減量は0.7%となり、待ち時間は55分/日(53%)増加した。

## 2.3 丸紅 ホームギャラリー[3]

マンションのセキュリティにおいてICカードによる玄関施錠など、ITセキュリティシステムを採用したマンションの導入をすでに始めている。そのICカードの主な使用場面はエントランス、エレベータホール、住戸玄関などだ。エントランスでは住居者以外の侵入を制限するためにICカードをかざすことで住居者の認証を行える。認証後エントランスの施錠とともに、エレベータを1階まで呼び寄せる最新のセキュリティを搭載している。エレベータホールではエレベータホールでICカードをかざすことにより、エレベータホールのボタンを押すことが可能になる。次にエレベータ内でもう一度かざし、利用階のボタンを選ぶ。こうすることで住居者以外の利用を制限する。住戸玄関では住戸の玄関扉を鍵ではなくICカードで施錠・解錠ができる。リバーシブルディングキーと併用できるので、さらに安心・便利に使うことが出来るようになっている。

## 2.4 ITセキュリティ[4][5]

### 2.4.1 ITセキュリティとは

情報の機密性、完全性、可用性の以下の三つを維持することであると定義されている。

- 機密性  
その情報にアクセスすることが認められた者だけが、その情報にアクセスすることが出来る状態を確保すること、認可されていない個人やエンティティには情報を非公開にして使用させないこと。
- 完全性  
その情報が破壊や改ざんまたは消去されていないで完全な状態を確保すること。
- 可用性  
その情報にアクセスすることを認められた個人やエンティティが、必要な時に速やかにその情報や資産にアクセスできる状態を確保すること。

### 2.4.2 ITセキュリティ対策

主にウイルス・不正アクセスや脆弱性に関する対策情報、制御システムの情報セキュリティ対策等がある。制御システムは、製造業を含むさまざまな産業領域で利用されている他、大規模な石油科学プラントの制御や、電力システム監視制御、ダムや水供給システム監視制御など国民生活の基板サービスを提供する重要なシステムとして利用されている。その一方で制御システムに関するソフトウェアに脆弱性が発見されるという事案も散見され始めている。JPCERT/CCでは、プロセス監視・制御システムのセキュリティに関し、国内の脆弱性関連情報調整機関として対策の促進に資する活動を進めている。

### 第3章 提案手法

通常、エレベータはボタンを押すことによって呼び出す。そして、乗車してから自分が行きたい階のボタンを押して[閉]のボタンを押す。ドアが閉まってからエレベータが自分が行きたい階へと移動し、着くとドアが開きます。この一連の流れを IC カードで操作することが出来る。

上述でも上げたようにマンションのエントランスからは IC カードによるセキュリティの解錠とともにエレベータの呼び出しが可能だ。この場合上りは待ち時間の削減が可能となる。そして、エレベータホールにて再度 IC カードを使用し、エレベータを使うことが可能だ。こうすることで住居者以外の人からの侵入を防ぐことが出来る。

さらにこの IC カードで住戸玄関の施錠、解錠も出来る。そこで、待ち時間に重きを置くと、改善できる点がいくつかある。まず一つはエントランスにて IC カードで解錠すると同時にエレベータが1階へ来るが、行き先指定をエントランスで解錠と同時に行うことでエレベータに乗ればその階へのボタンを押さずに済む。そして、もう一つはその IC カードで住戸玄関の施錠、解錠が出来るが、出かける際エレベータを使うのは必須となるので施錠とともにエレベータの呼び出しを可能にする。

以下の機能を有するエレベータシステムを提案し、通常のマansion、IT セキュリティのマansionでシミュレーション実験を行い、その待ち時間を比較する。

- エントランスで解錠とともにエレベータを1階に呼ぶ機能
- エントランスで解錠とともにエレベータに住戸階を指定する機能
- 住戸施錠とともにエレベータに住戸階に呼ぶ機能
- 住戸施錠とともにエレベータに1階を指定する機能
- 指定階ボタンの長押し等で指定をキャンセルする機能

## 第4章 実験と評価

### 4.1 実験目的

従来のエレベータと提案するシステムとでどちらが待ち時間の削減が出来るのかと検証する。

### 4.2 実験環境

モデルマンションの構造を以下のように設定した。

- 10階建て
- 1フロア4部屋
- 部屋からエレベータまでの時間は近い順に4秒、8秒、12秒、16秒
- エレベータの最大速度を分速40m、分速50m、分速60mの3通り
- どの場合も加速度は同じ

### 4.3 実験内容

検証対象となる時間は8時から20時とし、その間に200回エレベータが使われるとして、ランダムにキューを作成した。それを100日分計測し、待ち時間と移動時間の平均を取った。

比較対象としては、エレベータが1階で待機する場合とエレベータを使用した人が降りた階にて待機する場合の2パターン、ITセキュリティとエレベータが連動していない場合、ITセキュリティとエレベータが連動している既存技術、提案システムの3パターンの計6パターンで検討する。ここでいう待ち時間とは、部屋からエレベータまでの時間、エレベータを待つ時間、エレベータに乗っている時間、エレベータからエントランスまでの時間の総和とする。

検証パターンを表1にまとめ以下に示す。

表1:各パターンの詳細

	待機する場所	ITセキュリティとの連動
通常1	1階	エントランスなし、住戸玄関なし
通常2	降りた階	エントランスなし、住戸玄関なし
既存1	1階	エントランスあり、住戸玄関なし
既存2	降りた階	エントランスあり、住戸玄関なし
提案1	1階	エントランスあり、住戸玄関あり
提案2	降りた階	エントランスあり、住戸玄関あり

#### 4.4 実験結果

分速 40 m の各パターンでの待ち時間の平均と標準偏差を表 2 にまとめ以下に示す。また分速 40m の各パターンでの待ち時間の平均を図 10 に示す。図 10 を見ると待ち時間は提案 2 が最良である事がわかる。

表 2:分速 40m でのエレベータの待ち時間の平均と標準偏差

	平均	標準偏差
通常1	34.688	0.026
通常2	34.412	0.042
既存1	32.315	0.062
既存2	32.079	0.054
提案1	30.096	0.025
提案2	29.840	0.014

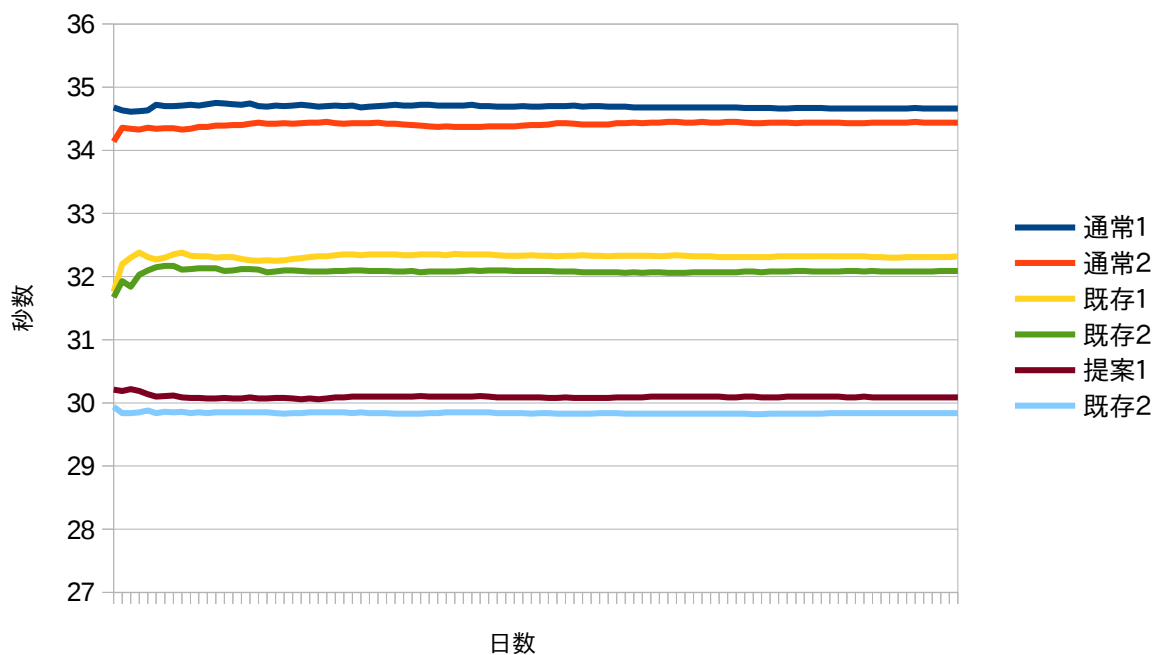


図 10:分速 40m でのエレベータの待ち時間

分速 40 mの各パターンでのエレベータの移動時間の平均と標準偏差を表 3 にまとめ以下に示す。分速 40m の各パターンのエレベータの移動時間の平均値を図 11 に示す。図 11 を見ると移動時間は提案 2 が最良である事がわかる。

表 3:分速 40m でのエレベータの移動時間の平均と標準偏差

	平均	標準偏差
通常1	1729.782	0.555
通常2	1699.724	0.791
既存1	1738.134	0.828
既存2	1697.196	0.376
提案1	1739.312	0.807
提案2	1695.739	1.293

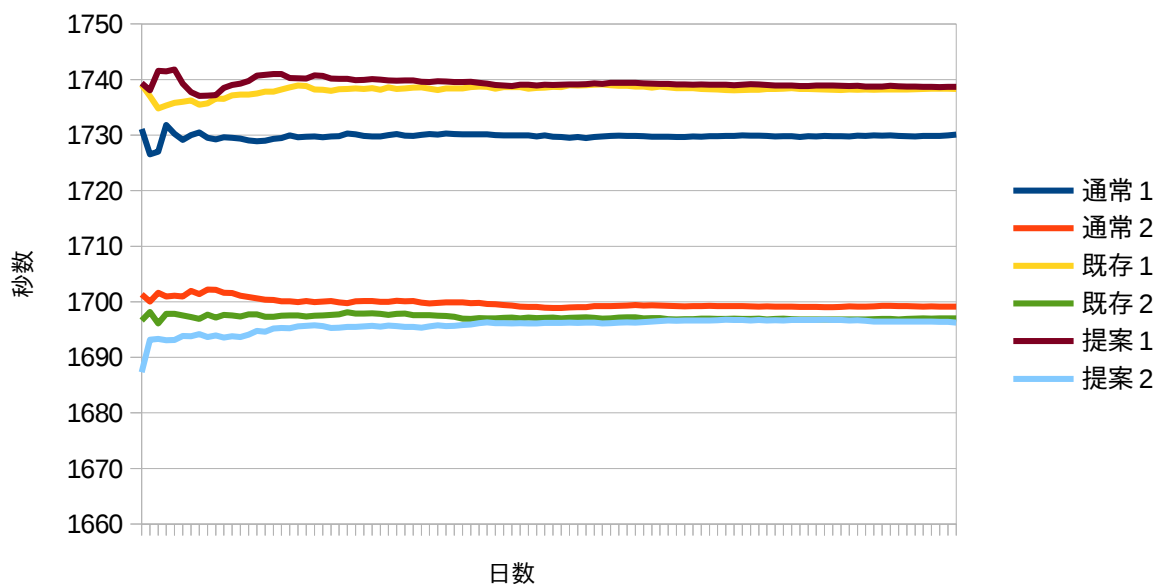


図 11:分速 40m でのエレベータの移動時間

分速 50 mの各パターンでの待ち時間の平均と標準偏差を表 4 にまとめ以下に示す。また分速 50m の各パターンでの待ち時間の平均を図 12 に示す。図 12 を見ると待ち時間は提案 2 が最良である事がわかる。

表 4:分速 50m でのエレベータの待ち時間の平均と標準偏差

	平均	標準偏差
通常1	34.536	0.053
通常2	34.451	0.097
既存1	32.369	0.039
既存2	32.001	0.058
提案1	30.279	0.011
提案2	29.748	0.016

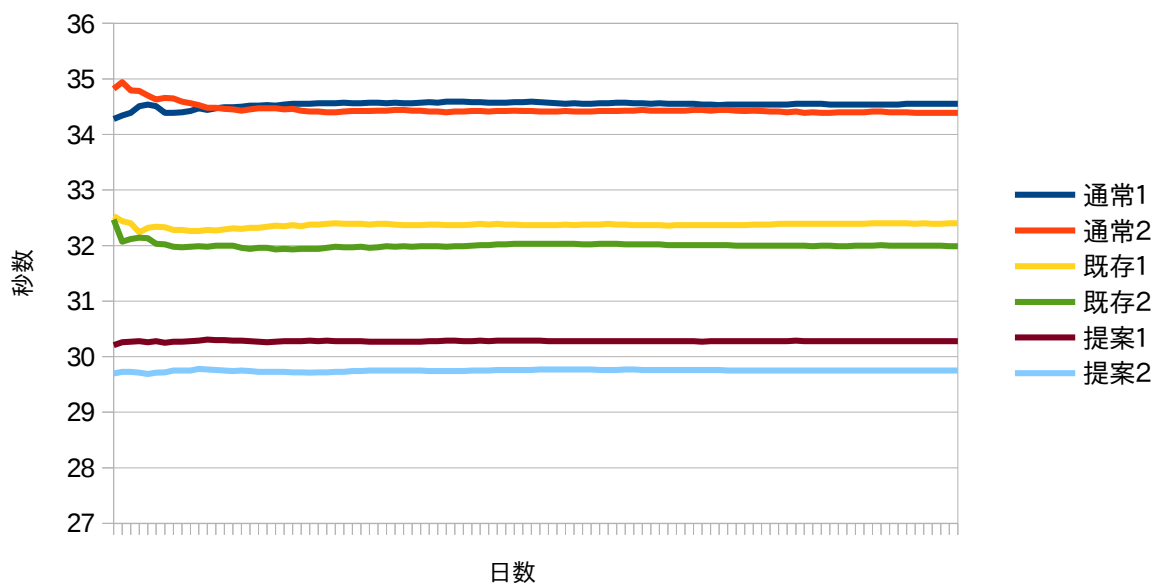


図 12:分速 50m でのエレベータの待ち時間



分速 50 mの各パターンでの待ち時間の平均と標準偏差を表 5 にまとめ以下に示す。また分速 50m の各パターンでの待ち時間の平均を図 13 を示す。図 13 を見ると待ち時間は提案 2 が最良である事がわかる。

表 5:分速 50m でのエレベータの移動時間の平均と標準偏差

	平均	標準偏差
通常1	1704.123	1.100
通常2	1678.930	0.555
既存1	1710.610	1.274
既存2	1674.926	0.329
提案1	1710.861	0.800
提案2	1675.039	0.578

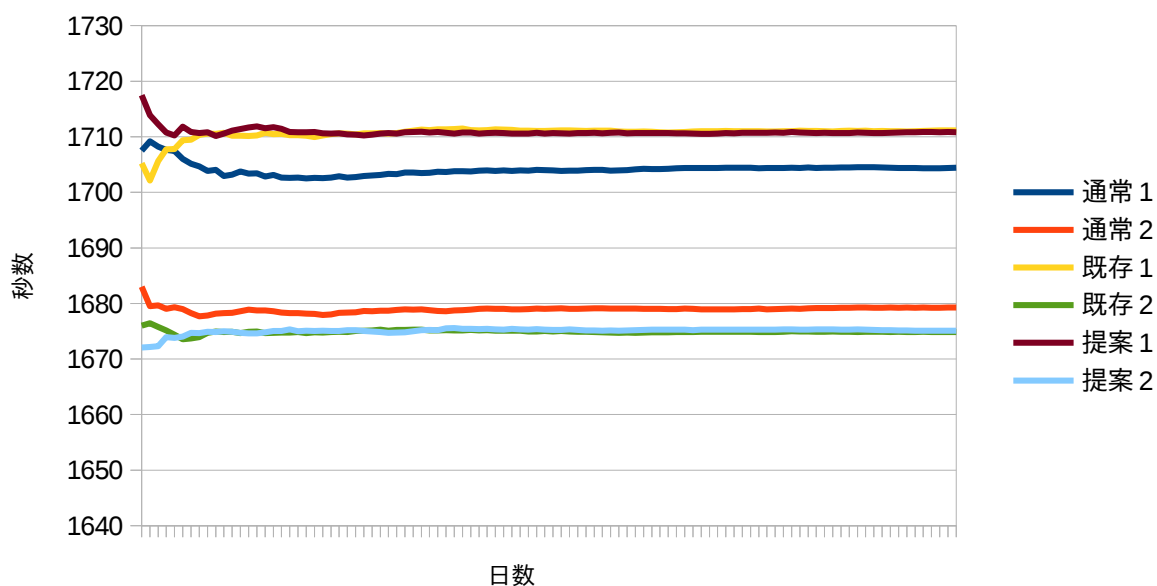


図 13:分速 50m でのエレベータの移動時間

分速 60 m の各パターンでの待ち時間の平均と標準偏差を表 6 にまとめ以下に示す。また分速 60m の各パターンでの待ち時間の平均を図 14 に示す。図 14 を見ると待ち時間は提案 2 が最良である事がわかる。

表 6:分速 60m でのエレベータの待ち時間の平均と標準偏差

	平均	標準偏差
通常1	34.560	0.040
通常2	34.341	0.087
既存1	32.247	0.059
既存2	31.953	0.027
提案1	29.938	0.021
提案2	29.716	0.012

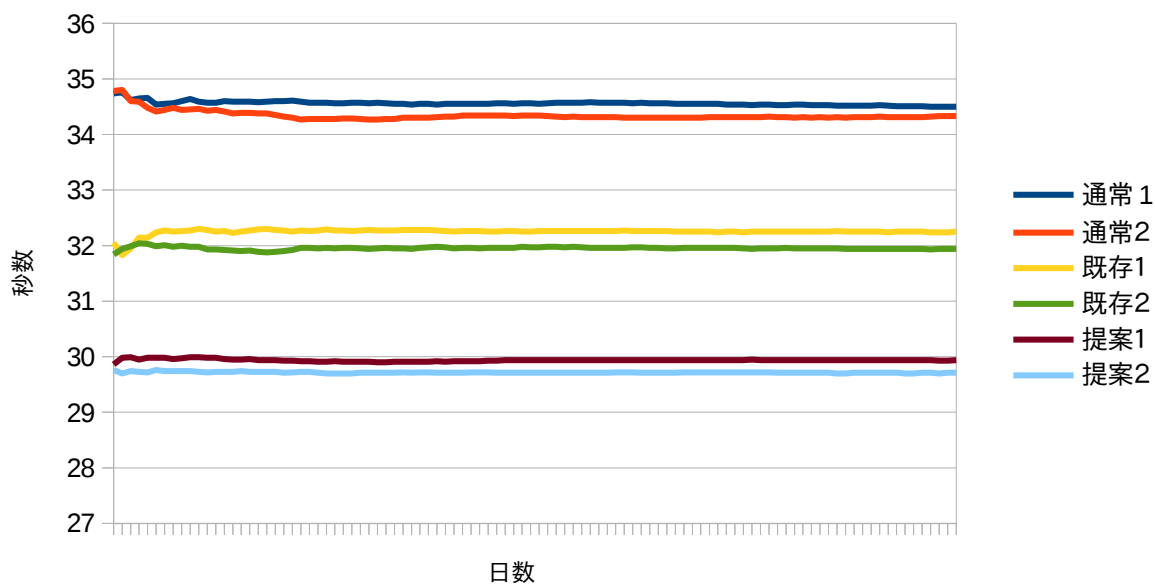


図 14:分速 60m でのエレベータの待ち時間

分速 60 mの各パターンでの待ち時間の平均と標準偏差を表 7にまとめ以下に示す。また分速 60m の各パターンでの待ち時間の平均を図 15 を示す。図 15 を見ると待ち時間は提案 2 が最良である事がわかる。

表 7:分速 60m でのエレベータの移動時間の平均と標準偏差

	平均	標準偏差
通常1	1687.070	0.465
通常2	1665.356	0.699
既存1	1692.316	0.287
既存2	1661.651	0.497
提案1	1692.188	0.414
提案2	1661.257	0.699

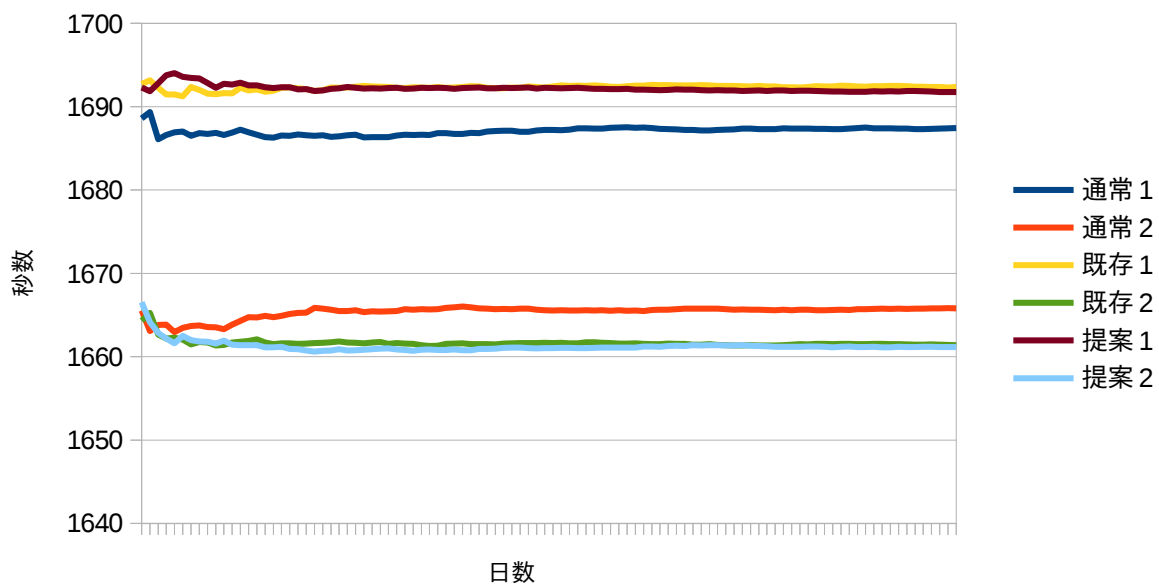


図 15:分速 60m でのエレベータの移動時間

## 4.5 考察

図 10、図 12、図 14、を見ると、どの速度、どのパターンでもエレベータが 1 階で待機しているよりエレベータを使用した人が降りた階で待機しているほうが平均 1%待ち時間が削減されていることがわかった。また、通常システムのエレベータと提案システムのエレベータとでは約 13%の削減、既存システムのエレベータと提案システムのエレベータとでは約 7%削減されていることがわかった。このことから提案したシステムが待ち時間を削減出来ていると考えられる。

また、図 11、図 13、図 15 を見ると、エレベータの移動時間は、エレベータが 1 階に待機している場合より、降りた階で待機したほうがどの速度、どの手法でも約 2%~4%削減されていることがわかった。移動時間の削減は消費電力の削減であるので、1 階で待機している場合より降りた階で待機している場合のほうが消費電力を削減できると言える。

表 2 から表 7 をみると標準偏差が 0~1.3 となっているので数値にばらつきがないことがわかる。このことからそれぞれの数値の平均値が妥当な数値であることがわかる。

結果的には IC カードによる IT セキュリティを導入した方が待ち時間の削減にはなったこのことからマンションのエントランスと住戸玄関に IC カードによる IT セキュリティを導入することによって待ち時間の削減は可能と考えた。しかし、IC カードを提示するだけでエントランスを通ると住民以外が後ろにいた場合そのまま通ることが考えられる。また、今回乗り合わせを考慮していないので考慮するともう少し待ち時間の検証結果が変化すると考えられる。乗り合わせを考える場合も検証できればより正確な結果が得られたと考える。

## 第5章 まとめ

本論文では IC カードによる IT セキュリティとエレベータを連動するシステムを提案した。提案システムの性能を評価するために通常のエレベータシステムと既存システムと提案システムとで比較実験を行った。その結果、通常のエレベータシステムと提案システムとでは 7%、既存システムと提案システムでは 13% 待ち時間が削減されている結果となった。消費電力も通常のエレベータシステムと提案システムとでは約 4%、既存システムと提案システムとでは約 2% 削減することが出来た。このことによりマンションの IT セキュリティとエレベータが連動した提案システムが有効であることが判明した。

本実験では、いつ使われたかを乱数で決定しているため時間帯による使用密度が考慮されていない。よく使うと思われる 7 時から 8 時半、17 時半から 20 時の使用密度を上げ、それ以外の時間帯の密度を下げることでより正確な評価が出来ると考えられる。

## 謝辞

本研究を進めるにあたり、ご指導を頂いた卒業論文指導教員の三好力教授に感謝致します。また、日常の議論を通じて多くの知識や示唆を頂いた三好研究室の皆様にも感謝致します。

## 参考文献

- [1]一般社団法人日本エレベーター協会  
URL: [www.m-elekyo.or.jp](http://www.m-elekyo.or.jp)
- [2]電力中央研究所  
「エレベータの運転台数変更による省エネルギー効果と利用者便益の変化に関する定量的分析」(2006.7)
- [3]丸紅 ホームギャラリー  
URL: [www.marubeni-sumai.com/club/column/sec\\_01.html](http://www.marubeni-sumai.com/club/column/sec_01.html)
- [4]情報セキュリティ検定試験一財団法人 全日本情報学習振興協会  
URL: [www.joho-gakushu.or.jp](http://www.joho-gakushu.or.jp)
- [5]JRCERT コーディネーションセンター  
URL: [www.jpccert.or.jp](http://www.jpccert.or.jp)

## 付録

```
//通常 1
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#define Number 200
#define SKD 40 //速度を分速 40m、分速 50m、分速 50m で
計測する。

void swap(int *x, int *y){
    int temp = *x;
    *x = *y;
    *y = temp;
}

void sort(int data[],int num){
    int o = num -1;
    while(o>=0){
        int p,q;
        for(p=1,q=-1;p<=o;p++){
            if(data[p-1] > data[p]){
                q = p - 1;

                swap(&data[p], &data[q]);
            }
            o = q;
        }
    }

    int main(){
        FILE *fq;
        int a,e,f,i,m,n,p;
        double d,g,ga,gb,gt,h,q,r,qr,sum;
        int
        b[Number],c[Number],j[Number],k[Number],l[Number];
        int unixtime[Number];
        srand((unsigned int)time(NULL));

        for(i=0;i<Number;i++){
            unixtime[i] = rand()%43200 + 28800;
        }

        char ss[15];
        sprintf(ss,"pt1-%d.txt",SKD);
        fq=fopen(ss,"a");

        int Time;
        for(Time=1;Time<101;Time++){
            for(a=0;a<Number;a++){
                p = rand()%2+1;
                if(p==1){
                    b[a] = rand()%9 + 2;
                    c[a]=2*(b[a]-1);
                    d = 0.1 + c[a]*0.2;
                    sum = sum + d;
                    e = rand()%4 + 1;
                    if(e == 1)f = 4;
                    else if(e == 2)f = 8;
                    else if(e == 3)f = 12;
                    else if(e == 4)f = 16;
                    g = 4*2 + (3*c[a] - 2.4)/SKD;
                    ga = ga + g;
                    printf("%2d 階分の移動時間:%2.2f ",c[a],g);
                    h = f + g + 16;
                    q = q + h;
                    sort(unixtime, Number);
                    j[a] = unixtime[a]/3600;
                    k[a] = (unixtime[a] - j[a]*3600)/60;
                    l[a] = unixtime[a] - (j[a]*3600 + k[a]*60);
                }

                else if(p==2){
                    b[a] = rand()%8+2;
                    c[a]=2*(b[a]-1);
                    d = 0.1 + c[a]*0.2;
                    sum = sum + d;
                    e = rand()%4 + 1;
                    if(e == 1)f = 4;
                    else if(e == 2)f = 8;
                    else if(e == 3)f = 12;
                    else if(e == 4)f = 16;
                    g = 4*2 + (3*c[a] - 2.4)/SKD;
                    gb = gb + g;
                    printf("%2d 階分の移動時間:%2.2f ",c[a],g);
                    h = f + g + 16;
                    r = r + h;
                    sort(unixtime, Number);
                    j[a] = unixtime[a]/3600;
                    k[a] = (unixtime[a] - j[a]*3600)/60;
                    l[a] = unixtime[a] - (j[a]*3600 + k[a]*60);
                }

                qr = q + r;
            }
        }
    }
}
```



```

    gt = ga + gb;
}
fprintf(fq,"%2.2f %2.2f\n",(qr/Number)/Time,gt/Time);
printf("待ち時間の合計:%2.2f ",qr/Time);
printf("待ち時間の平均:%2.2f ",(qr/Number)/Time);
}
fclose(fq);
return 0;
}

//通常 2
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#define Number 200
#define SKD 40 //速度を分速 40m、分速 50m、分速 50m で
計測する。

void swap(int *x, int *y){
    int temp = *x;
    *x = *y;
    *y = temp;
}

void sort(int data[],int num){
    int o = num -1;
    while(o>=0){
        int p,q;
        for(p=1,q=-1;p<=o;p++){
            if(data[p-1] > data[p]){
                q = p - 1;

                swap(&data[p], &data[q]);
            }
            o = q;
        }
    }

    int main(){
        FILE *fq;
        int a,e,f,i,m,n,p,s;
        double d,g,ga,gb,gt,h,q,r,qr,sum;
        int b[Number],c,j[Number],k[Number],l[Number];
        int unixtime[Number];
        srand((unsigned int)time(NULL));

        s=1;
        for(i=0;i<Number;i++){
            unixtime[i] = rand()%43200 + 28800;
        }

        char ss[15];
        sprintf(ss,"pt1-%d.txt",SKD);
        fq=fopen(ss,"a");

        int Time;
        for(Time=1;Time<101;Time++){
            for(a=0;a<Number;a++){
                p = rand()%2+1;
                if(p==1){
                    b[a] = rand()%9 + 2;
                    if(b[a]>s)c=(b[a]-s)+(b[a]-1);
                    else c=(s-b[a])+(b[a]-1);
                    s = 1;
                    d = 0.1 + c*0.2;
                    sum = sum + d;
                    e = rand()%4 + 1;
                    if(e == 1)f = 4;
                    else if(e == 2)f = 8;
                    else if(e == 3)f = 12;
                    else if(e == 4)f = 16;
                    g = 4*2 + (3*c - 2.4)/SKD;
                    ga = ga + g;
                    printf("%2d 階分の移動時間:%2.2f ",c,g);
                    h = f + g + 16;
                    q = q + h;
                    sort(unixtime, Number);
                    j[a] = unixtime[a]/3600;
                    k[a] = (unixtime[a] - j[a]*3600)/60;
                    l[a] = unixtime[a] - (j[a]*3600 + k[a]*60);
                }

                else if(p==2){
                    b[a] = rand()%8+2;
                    if(s==1)c=(b[a]-1);
                    if(s!=1 && b[a]>s)c=(b[a]-1)+(b[a]-s);
                    if(s!=1 && b[a]<s)c=(b[a]-1)+(s-b[a]);
                    s = b[a];
                    d = 0.1 + c*0.2;
                    sum = sum + d;
                    e = rand()%4 + 1;
                    if(e == 1)f = 4;
                    else if(e == 2)f = 8;

```

```

else if(e == 3)f = 12;
else if(e == 4)f = 16;
g = 4*2 + (3*c - 2.4)/SKD;
gb = gb + g;
printf("%2d 階分の移動時間:%2.2f ",c,g);
h = f + g + 16;
r = r + h;
sort(unixtime, Number);
j[a] = unixtime[a]/3600;
k[a] = (unixtime[a] - j[a]*3600)/60;
l[a] = unixtime[a] - (j[a]*3600 + k[a]*60);
}
qr = q + r;
gt = ga + gb;
}
fprintf(fq,"%2.2f %2.2f¥n",(qr/Number)/Time,gt/Time);
printf("待ち時間の合計:%2.2f ",qr/Time);
printf("待ち時間の平均:%2.2f ",(qr/Number)/Time);
}
fclose(fq);
return 0;
}

//既存 1
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#define Number 200
#define SKD 40 //速度を分速 40m、分速 50m、分速 50m で
計測する。

void swap(int *x, int *y){
    int temp = *x;
    *x = *y;
    *y = temp;
}

void sort(int data[],int num){
    int o = num -1;
    while(o>=0){
        int p,q;
        for(p=1,q=-1;p<=o;p++){
            if(data[p-1] > data[p]){
                q = p - 1;
                swap(&data[p], &data[q]);
            }
        }
        o = q;
    }
}

int main(){
    FILE *fp;
    int a,e,f,i,m,n,p;
    double d,g,ga,gb,gt,h,q,r,qr,sum;
    int
b[Number],c[Number],j[Number],k[Number],l[Number];
    int unixtime[Number];
    srand((unsigned int)time(NULL));

    for(i=0;i<Number;i++){
        unixtime[i] = rand()%43200 + 28800;
    }

    char ss[15];
    sprintf(ss,"pt1-%d.txt",SKD);
    fq=fopen(ss,"a");
    int Time;

    for(Time = 1;Time < 101;Time++){
        for(a=0;a<Number;a++){
            p = rand()%2+1;
            if(p==1){
                b[a] = rand()%9 + 2;
                c[a]=2*(b[a]-1);
                d=0.1 + c[a]*0.2;
                sum = sum + d;
                e = rand()%4 + 1;
                g = 4*2 + (3*c[a] - 2.4)/SKD;
                ga = ga + g;
                printf("%2d 階分の移動時間:%2.2f ",c[a],g);
                if(e == 1){
                    f = 4;
                }
                if(e == 2){
                    f = 8;
                }
                if(e == 3){

```

```

        f = 12;
    }
    if(e == 4){
        f = 16;
    }
    h = f + g + 16;
    q = q + h;
    sort(unixtime, Number);
    j[a] = unixtime[a]/3600;
    k[a] = (unixtime[a] - j[a]*3600)/60;
    l[a] = unixtime[a] - (j[a]*3600 + k[a]*60);
}
if(p==2){
b[a] = rand()%9 + 2;
c[a]=2*(b[a]-1);
d = 0.1 + c[a]*0.2;
sum = sum + d;
e = rand()%4 + 1;
g = 4*2 + (3*c[a] - 2.4)/SKD;
gb = gb + g;
printf("%2d 階分の移動時間:%2.2f ",c[a],g);
if(e == 1){
    if(g < 4)f = 4 - g;
    else f = 4;
}
if(e == 2){
    if(g < 8)f = 8 - g;
    else f = 8;
}
}
}
if(e == 3){
    if(g < 12)f = 12 - g;
    else f = 12;
}
if(e == 4){
    if(g < 16)f = 16 - g;
    else f = 16;
}
h = f + g + 16;
r = r + h;
sort(unixtime, Number);
j[a] = unixtime[a]/3600;
k[a] = (unixtime[a] - j[a]*3600)/60;
l[a] = unixtime[a] - (j[a]*3600 + k[a]*60);
}
qr = r + q;
gt = ga + gb;
}
fprintf(fp,"%2.2f %2.2f¥n",(qr/Number)/Time,gt/Time);
printf("待ち時間の合計:%2.2f 秒 ",qr/Time);
printf("待ち時間の平均:%2.2f ",(qr/Number)/Time);
}
fclose(fp);
return 0;
}
//既存 2
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#define Number 200

```

```

#define SKD 40 //速度を分速 40m、分速 50m、分速 50m で
計測する。
void swap(int *x, int *y){
    int temp = *x;
    *x = *y;
    *y = temp;
}

void sort(int data[],int num){
    int o = num -1;
    while(o>=0){
        int p,q;
        for(p=1,q=-1;p<=o;p++){
            if(data[p-1] > data[p]){
                q = p - 1;

                swap(&data[p], &data[q]);
            }
            o = q;
        }
    }

    int main(){
        FILE *fq;
        int a,e,f,i,m,n,p;
        double d,g,ga,gb,gt,h,q,r,qr,sum;
        int b[Number],c,j[Number],k[Number],l[Number],s;
        int unixtime[Number];
        srand((unsigned int)time(NULL));
        s = 1;
        for(i=1;i<Number;i++){
            unixtime[i] = rand()%43200 + 28800;
        }

        char ss[15];
        sprintf(ss,"pt1-%d.txt",SKD);
        fq=fopen(ss,"a");

        int Time;
        for(Time=1;Time<101;Time++){
            for(a=1;a<Number;a++){
                p = rand()%2+1;
                b[a]=rand()%9+2;
                if(p==1){
                    if(b[a]>s)c=(b[a] - s)+ (b[a] - 1);

                    else c=(s-b[a])+(b[a]-1);

                    s = 1;

                    d =0.1 + c*0.2;
                    sum = sum + d;

                    e = rand()%4 + 1;
                    g = 4*2 + (3*c - 2.4)/SKD;

                    ga = ga + g;

                    printf("%2d 階分の移動時間:%2.2f ",c,g);

                    if(e == 1){
                        if(g < 4)f = 4 - g;

                        else f = 4;
                    }

                    if(e == 2){
                        if(g < 8)f = 8 - g;

                        else f = 8;
                    }

                    if(e == 3){
                        if(g < 12)f = 12 - g;

                        else f = 12;
                    }

                    if(e == 4){
                        if(g < 16)f = 16 - g;

                        else f = 16;
                    }
                }
            }
        }
    }

```

```

}
h = f + g + 16;

q = q + h;

sort(unixtime, Number);

j[a] = unixtime[a]/3600;

k[a] = (unixtime[a] - j[a]*3600)/60;

l[a] = unixtime[a] - (j[a]*3600 + k[a]*60);
}
if(p==2){
    if(s==1)c=(b[a] - 1);

if(s!=1 && b[a]>s)c=(b[a]-1)+(b[a]-s);

if(s!=1 && b[a]<s)c=(b[a]-1)+(s-b[a]);

s = b[a];

d =0.1 + c*0.2;

sum = sum + d;

e = rand()%4 + 1;

g = 4*2 + (3*c - 2.4)/SKD;

gb = gb + g;

printf("%2d 階分の移動時間:%2.2f ",c,g);

if(e == 1){

    f = 4;

}

if(e == 2){

    f = 8;

}

if(e == 3){

    f = 12;

}

if(e == 4){

    f = 16;

}

h = f + g + 16;
    r = r + h;

sort(unixtime, Number);

j[a] = unixtime[a]/3600;

k[a] = (unixtime[a] - j[a]*3600)/60;

l[a] = unixtime[a] - (j[a]*3600 + k[a]*60);
}
    qr = r + q;
    gt = ga + gb;
}
fprintf(fq,"%2.2f %2.2f\n",(qr/Number)/Time,gt/Time);
printf("待ち時間の合計:%2.2f ",qr/Time);
printf("待ち時間の平均:%2.2f ",(qr/Number)/Time);
}
fclose(fq);
return 0;
}

//提案2
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#define Number 200
#define SKD 40 //速度を分速 40m、分速 50m、分速 50m で
計測する。

void swap(int *x, int *y){
    int temp = *x;
    *x = *y;
    *y = temp;
}

```

```

}
void sort(int data[],int num){
    int o = num -1;
    while(o>=0){
        int p,q;
        for(p=1,q=-1;p<=o;p++){
            if(data[p-1] > data[p]){
                q = p - 1;
            }
            swap(&data[p], &data[q]);
        }
        o = q;
    }
}

int main(){
    FILE *fp;
    int a,e,f,i,m,n,p;
    double d,g,ga,gb,gt,h,q,r,qr,sum;
    int
b[Number],c[Number],j[Number],k[Number],l[Number];
    int unixtime[Number];
    srand((unsigned int)time(NULL));

    for(i=0;i<Number;i++){
        unixtime[i] = rand()%43200 + 28800;
    }

    char ss[15];
    sprintf(ss,"pt1-%d.txt",SKD);
    fq=fopen(ss,"a");

    int Time;
    for(Time = 1;Time < 101;Time++){
        for(a=0;a<Number;a++){
            p = rand()%2+1;
            if(p==1){
                b[a] = rand()%9 + 2;

                c[a]=2*(b[a]-1);

                d =0.1 + c[a]*0.2;
                sum = sum + d;
            }
            else {
                e = rand()%4 + 1;
                g = 4*2 + (3*c[a] - 2.4)/SKD;

                ga = ga + g;
                printf("%2d 階分の移動時間:%2.2f ",c[a],g);
                if(e == 1){
                    if(g < 4)f = 4 - g;
                    else f = 4;
                }
                if(e == 2){
                    if(g < 8)f = 8 - g;
                    else f = 8;
                }
                if(e == 3){
                    if(g < 12)f = 12 - g;
                }
                else f = 12;
            }
        }
        if(e == 4){
            if(g < 16)f = 16 - g;
            else f = 16;
        }
        h = f + g + 16;
        q = q +h;
        sort(unixtime, Number);
        j[a] = unixtime[a]/3600;
        k[a] = (unixtime[a] - j[a]*3600)/60;
        l[a] = unixtime[a] - (j[a]*3600 + k[a]*60);
    }
    if(p==2){
        b[a] = rand()%9 + 2;

        c[a]=2*(b[a]-1)

        d =0.1 + c[a]*0.2;

        sum = sum + d;

        e = rand()%4 + 1;

        g = 4*2 + (3*c[a] - 2.4)/SKD;
    }
}

```

```

gb = gb + g;
printf("%2d 階分の移動時間:%2.2f ",c[a],g);

if(e == 1){

if(g < 4)f = 4 - g;

else f = 4;

}

if(e == 2){

if(g < 8)f = 8 - g;

else f = 8;

}

if(e == 3){

if(g < 12)f = 12 - g;

else f = 12;

}

if(e == 4){

if(g < 16)f = 16 - g;

else f = 16;

}

h = f + g + 16;

r = r + h;

sort(unixtime, Number);

j[a] = unixtime[a]/3600;

k[a] = (unixtime[a] - j[a]*3600)/60;

l[a] = unixtime[a] - (j[a]*3600 + k[a]*60);

```

```

}

qr = r + q;

gt = ga + gb;
}
fprintf(fp,"%2.2f %2.2f¥n",(qr/Number)/Time,gt/Time);
printf("待ち時間の合計:%2.2f秒 ",qr/Time);
printf("待ち時間の平均:%2.2f ",(qr/Number)/Time);
}
fclose(fp);
return 0;
}

//提案2
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#define Number 200
#define SKD 40 //速度を分速 40m、分速 50m、分速 50mで
計測する。

void swap(int *x, int *y){
int temp = *x;
*x = *y;
*y = temp;
}

void sort(int data[],int num){
int o = num -1;
while(o>=0){
int p,q;
for(p=1,q=-1;p<=o;p++){
if(data[p-1] > data[p]){

q = p - 1;

swap(&data[p], &data[q]);
}
o = q;
}
}

int main(){
FILE *fq;

```

```

int a,e,f,i,m,n,p;
double d,g,ga,gb,gt,h,q,r,qr,sum;
int b[Number],c,j[Number],k[Number],l[Number],s;
int unixtime[Number];
srand((unsigned int)time(NULL));
s = 1;
for(i=1;i<Number;i++){
    unixtime[i] = rand()%43200 + 28800;
}

char ss[15];
sprintf(ss,"pt1-%d.txt",SKD);
fq=fopen(ss,"a");

int Time;
for(Time=1;Time<101;Time++){
for(a=1;a<Number;a++){
    p = rand()%2+1;
    b[a]=rand()%9+2;
    if(p==1){
        if(b[a]>s)c=(b[a] - s)+ (b[a] - 1);
        else c=(s-b[a])+ (b[a]-1);
        s = 1;
        d =0.1 + c*0.2;
        sum = sum + d;
        e = rand()%4 + 1;
        g = 4*2 + (3*c - 2.4)/SKD;
        ga = ga + g;
        printf("%2d 階分の移動時間:%2.2f ",c,g);
        if(e == 1){
            if(g < 4)f = 4 - g;

            else f = 4;
        }
        if(e == 2){
            if(g < 8)f = 8 - g;

            else f = 8;
        }
        if(e == 3){
            if(g < 12)f = 12 - g;

            else f = 12;
        }
        if(e == 4){
            if(g < 16)f = 16 - g;

            else f = 16;
        }
        h = f + g + 16;
        q = q + h;

        sort(unixtime, Number);
        j[a] = unixtime[a]/3600;
        k[a] = (unixtime[a] - j[a]*3600)/60;
        l[a] = unixtime[a] - (j[a]*3600 + k[a]*60);
    }

    if(p==2){
        if(s==1)c=(b[a] - 1);
        if(s!=1 && b[a]>s)c=(b[a]-1)+(b[a]-s);
        if(s!=1 && b[a]<s)c=(b[a]-1)+(s-b[a]);
        s = b[a];
        d =0.1 + c*0.2;
        sum = sum + d;
        e = rand()%4 + 1;
        g = 4*2 + (3*c - 2.4)/SKD;
        gb = gb + g;
        printf("%2d 階分の移動時間:%2.2f ",c,g);

        if(e == 1){
            if(g < 4)f = 4 - g;

            else f = 4;
        }
        if(e == 2){
            if(g < 8)f = 8 - g;

            else f = 8;
        }
        if(e == 3){
            if(g < 12)f = 12 - g;

            else f = 12;
        }
        if(e == 4){
            if(g < 16)f = 16 - g;

            else f = 16;
        }
        h = f + g + 16;
        q = q + h;

        sort(unixtime, Number);
        j[a] = unixtime[a]/3600;
        k[a] = (unixtime[a] - j[a]*3600)/60;
        l[a] = unixtime[a] - (j[a]*3600 + k[a]*60);
    }
}

```



```

if(g < 16)f = 16 - g;

else f = 16;
    }
    h = f + g + 16;
    r = r + h;
    sort(unixtime, Number);
    j[a] = unixtime[a]/3600;
    k[a] = (unixtime[a] - j[a]*3600)/60;
    l[a] = unixtime[a] - (j[a]*3600 + k[a]*60);
    }
    qr = r + q;
    gt = ga + gb;
    }
    fprintf(fq,"%2.2f %2.2f¥n",(qr/Number)/Time,gt/Time);
    printf("待ち時間の合計:%2.2f ",qr/Time);
    printf("待ち時間の平均:%2.2f ",(qr/Number)/Time);
    }
    fclose(fq);
    return 0;
}

```