

Title	照明列車向け高速光空間通信システムにおける快適操作を実現するユーザーインターフェースの検討
Sub Title	Comfortable user interface technology for high-speed ground-to-train communication system using free-space optics technology
Author	小川, 豊弘(Ogawa, Toyohiro) 春山, 真一郎(Haruyama, Shinichiro)
Publisher	慶應義塾大学大学院システムデザイン・マネジメント研究科
Publication year	2009
Jtitle	
JaLC DOI	
Abstract	
Notes	修士学位論文. 2009年度システムデザイン・マネジメント学 第6号
Genre	Thesis or Dissertation
URL	<a href="https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40002001-00002009-0008">https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40002001-00002009-0008</a>

慶應義塾大学学術情報リポジトリ(KOARA)に掲載されているコンテンツの著作権は、それぞれの著作者、学会または出版社/発行者に帰属し、その権利は著作権法によって保護されています。引用にあたっては、著作権法を遵守してご利用ください。

The copyrights of content available on the Keio Associated Repository of Academic resources (KOARA) belong to the respective authors, academic societies, or publishers/issuers, and these rights are protected by the Japanese Copyright Act. When quoting the content, please follow the Japanese copyright act.

修士論文

2009 年度

照明列車向け高速光空間通信システム  
における快適操作を実現する  
ユーザーインターフェースの検討

小川 豊弘

(学籍番号：80833091)

指導教員 教授 春山 真一郎

2010 年 3 月

慶應義塾大学大学院システムデザイン・マネジメント研究科  
システムデザイン・マネジメント専攻

# 論 文 要 旨

学籍番号	80833091	氏 名	小川 豊弘
論文題目： 照明列車向け高速光空間通信システムにおける 快適操作を実現するユーザーインターフェースの検討			
(内容の要旨) 照明列車向け高速光空間通信システム実現に伴う列車内における PC 快適操作を実現するためのユーザーインターフェースの検討を行っている。照明列車向け高速光空間通信システムの現実化に伴い、列車内におけるパソコンの使用率が上昇することが予測される。また、今後座席シートへのモニター設備の充実化も予想される。しかし、列車内の振動に伴いモニターへの注視は乗り心地を低下させ、結果として乗り物よいを誘発させてしまう恐れがある。 本研究では、照明列車向け高速光空間通信システム実現に向けて、列車利用者に快適にサービスを利用してもらうための問題改善や品質向上を目指すために新幹線内で重要と考える要素をアンケートにより抽出した。結果として、新幹線内の振動が重要な要素であることが確認できた。そこで、列車内での PC 作業時の最適なテーブル位置、及び今後座席シートに付属モニターが設備充実化も視野に入れ、列車内振動とモニター注視との乗り心地の関連や乗り物酔い防止方法を検討していこうと考えた。そのため、本研究では、酔いを乗り物及び PC 作業に起因するものに限定している。 列車内振動の問題点としては大きく分けて三種あることを把握した。具体的には一つ目に振り子式車両によるもの。二つ目に作業台及び座面に起因するもの。三つ目に PC 作業による視覚情報に起因するものである。ここで、大きな問題としては、作業台及び座面に起因するもの、PC 作業による視覚情報に起因するものである。しかし、これらの問題点の改善には車両そのものの改善や列車内設備の改善と、大規模でコストや時間の観点から効率的ではない。そこで私がこの新幹線列車内の PC 作業に付随して発生する酔いの提言方法として提案する方法は作業台における振動に付随して振動とは逆の方向(上下・左右)にウィンドウを振動させることである。ウィンドウ振動の具体的方法として、加速度センサーを用いた方法を検討した。この方法を適用すればコストや時間の観点で優位性があると考えられる。 加速度センサーは本来三軸構造から成り立っているが、そのうち二軸を利用することで可能となる。加速度センサーの二軸を PC のウィンドウの左右・上下に対応するようウィンドウに設置し、加速度センサーの検出値から距離を算出し、ウィンドウのピクセル値に変換し、それに対応させウィンドウを動かすことが可能となる。結果として、視覚を動かすことなくモニターを注視することができる。また、動画閲覧時にも、本来動画を追う視覚変動に付随して振動が発生することにより、酔いを発生させるが、ウィンドウを振動させることで動画本来の視覚変動に収縮させることができ、列車内振動を低減できると考えられる。また、振り子式車両の改善や作業台の変更・改善に比較するとウィンドウを振動に反応させ振動させることはコスト・時間の面からも早期に実現することができ、効果としても十分に期待できる。			

## SUMMARY OF MASTER'S DISSERTATION

Student Identification Number	80833091	Name	Toyohiro Ogawa
Title			
Comfortable User Interface Technology for High-Speed Ground-to-Train Communication System Using Free-Space Optics Technology			
Abstract			
<p>It is forecast that the usage rate of the personal computer in the train rises along with realizing the high-speed ground-to-train communication system using free-space optics technology. Moreover, the expectation will leave the enhancement of the monitor equipment to the seat in the future. However, the gazing to the monitor decreases riding comfortable, and might cause that it is as a result by the vehicle along with the vibration in the train.</p> <p>It was able to be confirmed that the vibration in Shinkansen was an important factor by the questionnaire. Then, an attached monitor contemplated the equipment enhancement to the seat in the future the best table position to work PC in the train and it was thought that the method of preventing the relation and motion sickness of the riding comfort of the vibration in the train and the monitor gaze was examined. Therefore, intoxication has been limited to the one that originates in the vehicle and the PC working in the present study.</p> <p>Three kinds were understood to be roughly separately as a problem of the vibration in the train. It is concretely the one in first with the pendulum type vehicle. It is the one in second that originates in the worktable and the bearing surface. It is the one in third that originates in the visual information by the PC work. However, it is not efficient in the improvement of these problems from the viewpoint of the cost and time. The method with an acceleration sensor was examined. It is thought that it is a viewpoint of the cost and time and there is domination if this method is applied.</p> <p>Two axes of the acceleration sensor are set up in the window so that the window of PC may be right and left and correspond up and down. And the distance is calculated from the detection value of the acceleration sensor, it converts into the pixel value of the window, they are made to correspond to it, and it becomes possible to move the window. Consequently, it is possible the gazing at the monitor without moving the sight. It is possible to shrink to a sight change animation original by vibrating the window, and it is thought that the vibration in the train can be decreased.</p>			

# 目次

あらまし	4
第1章	
はじめに	6
第2章	
高速光空間システムに伴う列車内環境の把握	
2.1 列車内環境の調査とその方法	8
2.2 調査結果	20
2.3 問題点	31
2.4 列車内 PC 操作が人体に及ぼす影響	35
第3章	
列車内振動に対する検討と改善方法の提言	
3.1 列車内振動	39
3.2 振動と乗り心地及び酔いの関係	42
3.3 振動に対する対策の現状	46
3.4 振動に対する対策方法の提案	51
第4章	
結論	54
謝辞	
参考文献	
付録	

## あらまし

情報端末機器の発展に伴い現代ではあらゆる場所でインターネット接続環境が整備されている。家庭内やオフィス内などには高速通信網がすでに整備されている。徒歩などの移動中においても、携帯電話やホットスポットなどを利用してインターネット接続が容易になり、同時に、インターネット接続環境が整備されていることを付加価値としたビジネスまで誕生している。このように現代ではインターネット接続環境はなくてはならないインフラストラクチャーとして整備され、ユビキタス社会の実現へ向かっている。同時に、このインフラストラクチャー普及に伴い高速光空間通信システムを用いて、新たに新幹線等の高速移動列車内においても整備が普及の一途を辿っている。また一方で、このようなインフラストラクチャー普及に伴い携帯型パソコン(本稿では以下PCと略記)が急速に普及し、様々な場所でPCを使用する機会が多くなっている。

しかし、このようなインフラストラクチャーの整備の進行とPCニーズの普及の一方で利用者のPC環境(騒音, 振動, 温度, 照明, 空気清浄度等)の整備は不十分な状況にある。特に新幹線等の高速移動列車内では人体へ与えるPC環境(騒音, 振動, 温度, 照明, 空気清浄度等)の諸要因が多いため、PC利用の快適な空間を実現することは極めて困難な状況である。

そこで本研究では、インターネット接続環境のインフラストラクチャーが整備されつつある中で、よりよいPC使用環境の実現を目指すためのユーザーインターフェースの検討を行う。そのためにPCを利用する様々な場所の中で、快適性に影響する諸要因(騒音, 振動, 温度, 照明, 空気清浄度等)が最も多い高速移動する新幹線等の列車内に着目した。具体的にはまず、ユーザーニーズの調査を行うことによって、新幹線等の高速移動列車内におけるPC利用のニーズの把握を行った。その調査の結果から実際のユーザーがPC使用時に人体に与える不快感要素の抽出を行い、PC快適利用実現に向けた問題点を把握した。人体に与える不快感要素は多々あり、当然使用環境によっても種類や強弱の違いから単一要素として捉え解決することは困難である。そのため、人体に与える不快感要素の中から最もPCの快適な利用に影響し、なおかつ人体への不快感要素が強く、新幹線等の高速移動列車外でも十分に起こりうる要素である振動に着目し

た。

従来の研究では振動が及ぼすパソコン作業に及ぼす影響や振動が及ぼす酔いに関するものが数少なく。なおかつこのような先行研究では単一周波数の影響についての分析が中心であり高速列車走行時のような複合振動を考慮したものはほとんどない。また、ISO規格では乗り物のような振動環境下での作業について十分に検証されていない。

そのため、インターネット接続環境であるインフラストラクチャーの普及に付随して発生するPC環境の問題点を改善することが必要となってくるため、ユーザー調査を前提として列車走行時の振動環境下におけるPC作業を対象として具体的な改善方法・ユーザーインターフェースの検討を行った。

# 第1章

## はじめに

背景として、ユビキタス社会システムの発展に伴いインターネット接続環境としてのインフラストラクチャーの整備が進行している。具体的には家庭内やオフィス内などには高速通信網がすでに整備されています。徒歩などの移動中においても、携帯電話やホットスポットなどを利用して、インターネット接続が容易になり、ユビキタス社会の実現が現実のものとなりつつある。しかしながら、航空機や新幹線など、出張や旅行に利用する中・遠距離移動において通信環境は決して品質の良いものではない。

しかし、すでにインフラストラクチャー普及に伴い高速光空間通信システムを用いて、新たに新幹線等の高速移動列車内においても整備が普及の一途を辿っている。このようなインフラストラクチャーの整備の進行は我々の生活に密着し、これからの情報化社会に必要不可欠なものである。

また、同時にインターネット接続環境の整備に付随して起こることがある。それが、情報端末機器の普及である。具体的には携帯電話や携帯型パソコン(本稿では以下PCと略記)などがあげられる。そのため現代では携帯電話の普及率は約90%であり、PCの普及率も増加の一途を辿っている。

そこでひとつの問題点が生じてくる。それがインターネット接続環境のインフラストラクチャー普及に伴う情報端末機器の使用環境(この環境とは快適性に影響する諸要因、すなわち、騒音、振動、温度、照明、空気清浄等。以下同じ。)である。携帯電話に関しては片手で操作できることや使用場所が多様であることから使用環境の影響を受けにくい。しかし、PCに関してはそうではない。PCはある固定した場所でないと快適に使用することができないため使用環境の影響を受けやすい、また、外部でPCを使用してインターネット接続



を行うことはビジネスなど必要不可欠な作業に用いられる場合が多い。総務省調査によると PC を用いてインターネット接続を行うユーザーが 77.4%にものぼることが明らかにされている。しかし、現状では PC 使用環境に対する対策が図られていないことが現状である。このようにインターネット接続環境のインフラストラクチャー整備が進む一方で PC 使用環境の整備が進んでいない。

そこで、ここからは使用環境の影響を受けやすい PC に着目する。PC 使用環境の対策が進まない理由がある。それは PC 作業の快適性に及ぼす諸要因が多いことである。要因としては騒音、振動、温度、照明、空気清浄等があるが、いずれの要因も使用場所や使用時間など外部要因が多く個人の対策では解決することが不可能な要因が多い。つまり、これは PC 使用環境の整備が必要であることも表している。

そのため、本論文では高速列車内における PC 快適操作を実現することを目的としたインターフェースの検討を示す。高速移動列車をあげた理由としては PC 使用環境の外部要因が多いことで問題点が把握しやすく、PC 操作に影響を及ぼす要因の抽出が的確に行えること。また、新たに新幹線等の高速移動列車内においても整備が普及の一途を辿っていることから、今後使用環境の整備が必要不可欠であるためである。

本論文は次のように構成される。第 2 章では高速光空間システムに伴う列車内環境の把握として列車内環境の調査とその方法、調査結果、そこから生じる問題点を示す。第 3 章では列車内振動に対する検討と改善方法の提言として列車内振動の定義や影響、現状の対策、それに対する振動防止の検討を示し、第 4 章で検討の実験結果及び評価を示す。第 5 章が結論である。

## 第2章

### 高速光空間システムに伴う列車内環境の把握

列車内におけるパソコンの使用率が上昇することが予測される。また、今後座席シートへのモニター設備の充実化も予想される。しかし、列車内の振動に伴いモニターへの注視は乗り心地を低下させ、結果として乗り物よいを誘発させてしまう恐れがある。この実験理由としては、超高速光無線列車通信システム実現に向けて、列車利用者に快適にサービスを利用してもらうための問題改善や品質向上を目指すためにお客様が新幹線内で重要と考える要素をアンケートにより抽出した。

#### 2.1 列車内環境の調査とその方法

アンケート調査の目的は現行新幹線における列車内において利用者が重要視するニーズの把握(この場合のニーズとは PC 利用に関するものではなく、新幹線車両内のすべてのサービスを示す)である。また同時に、現在新幹線の列車内において携帯電話や PHS、PC を用いてインターネットを利用する利用者のニーズとその現状の利用状況の把握である。調査対象者は東海道新幹線を用いて東京地域～大阪地域～博多地域を月1回以上利用している利用者に対して行った。この理由としては、より日常的に新幹線を利用する顧客のニーズをアンケート結果に求めることで普段の新幹線内での利用者の意思を反映させるためである。

アンケートの調査内容としては大きく分けて二つある。

(1)実態把握として現行の交通機関の選択状況、交通機関選択要因の調査、移動時におけるパソコン、携帯電話等の利用状況、移動中

のデータ通信サービスに関する意見・要望

(2)列車内ブロードバンドサービスでの想定ニーズ調査としてブロードバンド回線を用いたサービス例を用いてその利用意向を問う事、移動効率（所要時間）、コスト以外の要因（居住性、乗り心地、マイレージサービス等）の中で通信サービスが占める相対的な重要度評価である。

また、今回のアンケート調査は照明列車向け高速光空間通信システム実現プロジェクトが鉄道総合技術研究所との共同研究であることから JR 東海登録ユーザーのスクリーニングにより東海道新幹線を用いて東京地域～大阪地域～博多地域を月 1 回以上利用している利用者を絞り込み Web 調査による方法により行った。調査時期は 2009 年 2 月である。実際のサンプル内約としては東京地区⇔大阪地区間（目的地は東京、神奈川、千葉、埼玉、大阪、京都、兵庫、奈良、和歌山）利用者 722 名、東京地区⇔岡山、広島、福岡地区間（目的地は、東京、神奈川、千葉、埼玉）利用者 309 名の合計 1031 名から回答を得た。このアンケート調査の内容を以下に示す。

## アンケート調査票の内容

1, あなたの本アンケート依頼文に記載されている区間名をお答えください。

- ①東京-大阪
- ②東京-岡山
- ③東京-広島
- ④東京-福岡

2, あなたが【1, 選択肢内容表示】間を移動する際の状況についてお答えください。

2, 1【1, 選択肢内容表示】間の移動手段について以下の中から実態に最も近いものをひとつ選んで下さい。

- ①常に新幹線
- ②常に航空機
- ③場合によるが新幹線が多い
- ④場合によるが飛行機が多い
- ⑤新幹線と飛行機が半々くらい

2, 2 あなたが【1, 選択肢内容表示】間を、出張等、仕事の用事で移動する場合についてお答えください。このような場面での交通機関選択にあたって、以下の各項目をどの程度重視しますか。項目ごとに該当するものをひとつ選んで下さい。

	重視する	やや重視する	あまり重視しない	重視しない
運賃・料金	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
所要時間	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
移動空間の居住性、乗り心地	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
拠点駅、空港等へのアクセスの良さ	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
移動中の通信サービス(携帯電話、インターネットサービス)	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
予約・予約変更のしやすさ	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
喫煙可能かどうか	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
高頻度利用者向けサービス(マイレッジサービス等)	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

2,3 あなたが【1, 選択肢内容表示】間を、家族旅行、帰省等のプライベートな用事で移動する場合についてお答えください。このような場面での交通機関選択にあたって、以下の各項目をどの程度重視しますか。項目ごとに該当するものをひとつ選んで下さい。

	重視する	やや重視する	あまり重視しない	重視しない
運賃・料金	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
所要時間	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
移動空間の居住性、乗り心地	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
拠点駅、空港等へのアクセスの良さ	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
移動中の通信サービス(携帯電話、インターネットサービス)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
予約・予約変更のしやすさ	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
喫煙可能であるかどうか	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
高頻度利用者向けサービス(マイレッジサービス等)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2,4 あなたは【1, 選択肢内容表示】間を仕事で移動する場合それぞれの項目は新幹線を利用するにあたってどの程度魅力的ですか。それぞれあてはまるものをお選び下さい。※仕事で新幹線を利用することがない方は、イメージでお答えください。

	魅力的である	やや魅力的である	あまり魅力的でない	魅力的でない
運賃・料金の割引き(10%程度)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
所要時間の短縮(10分)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
専用会議スペースの提供(有料)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
各座席での高速インターネットサービスの提供(有料)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
共用ビジネススペースでの高速インターネットサービスの提供(有料)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
高頻度旅行者向けサービスの充実(航空会社とほぼ同様のマイレッジサービス)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
完全禁煙化	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2, 5(1, 1 の回答が②④⑤の方にお尋ねします)

今後以下のようなサービスが実現した場合に、【1, 選択肢内容表示】間の移動における新幹線利用頻度の変化について該当するものをひとつ選んで下さい。

	全面的に新幹線 に変更する	現在より新幹線 を選択する機会 が多くなる	新幹線を選択す る機会は現状と 変わらない
運賃・料金の割引(10%程度)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
所要時間の短縮(10分)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
専用会議スペースの提供(有料)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
各座席での高速インターネットサービスの提供(有料)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
共用ビジネススペースでの高速インターネットサービスの提供(有料)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
高頻度旅行者向けサービスの充実(航空会社とほぼ同様のマイレージサービス)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
完全禁煙化	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

3, 出張, 旅行等で移動中におけるパーソナルコンピュータ(以下、PC)およびインターネットサービスの利用について以下の問いにお答えください。

3, 1 あなたは以下のものを用いて, 新幹線列車内において, 仕事目的で以下のものを利用しますか?

※項目[2, PC]はネットワークに接続せず, PC 単体で使用する場合を含みます。

	常に利用する	時々利用する	ほとんど利用しない	まったく利用しない
携帯電話もしくはPHS端末を用いて(i-mode等)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PCに携帯電話やPHS等のデータ通信カードをつけて	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



3,5 新幹線車内でご自分の座席の周辺の方が PC を使用している場合にどのように感じますか。あてはまるものを以下からひとつ選んで下さい。

- ①特に気にならない
- ②ほとんど気にならない
- ③少し気になる
- ④とても気になる
- ⑤その他【

】

4, 将来, 新幹線列車内で, 高速インターネットサービスが実現された場合を想定して以下の問いにお答えください。

4,1 自宅やオフィスなどで一般的に使用されている比較的高速なインターネットサービス(動画の閲覧や Skype 等の音声通話も可能)が列車内で提供された場合, あなたはこのサービスを使用したいと思いますか。以下からひとつ選んで下さい。なお, サービスを利用する場合にはご自身の PC を車内に持参し, ネットワークに接続するものとします。

- ①利用したい
- ②条件次第では利用したい
- ③利用しない



4, 2(4, 1 で①あるいは②と回答した方にお訊ねします)

利用する場合にどのような用途に用いますか。あてはまるものをすべて選んで下さい。

- ①メールの送受信(Web メールを含む)
- ②各種予約システム(交通機関、宿泊等)の利用
- ③ブログへの書き込み, Web ページの製作
- ④掲示板の読み書き
- ⑤音声通話サービス, テレビ会議等(たとえば Skype の利用)
- ⑥旅行先, 出張先に関する情報収集(観光情報, 道案内情報等)
- ⑦会社のネットワークへの接続
- ⑧動画の閲覧, アップロード(Youtube 等の動画共有サービス等)
- ⑨インターネットバンキング
- ⑩オンラインゲーム
- ⑪株, 為替等のオンライントレード
- ⑫ニュース, 天気予報等の一般的な Web ページの閲覧
- ⑬その他【 】

4, 3(4, 1 で②と回答した方にお訊ねします)

利用するための条件は何ですか。あてはまるものをすべて選んで下さい。

- ①利用料金がかからない(無料である)ならば
- ②利用料金が1回の乗車あたり( )円以下であれば
- ③安定した接続が可能ならば
- ④会社のネットワークと接続可能ならば
- ⑤セキュリティ対策がしっかりしているならば
- ⑥PCのレンタルサービスもしくは共用PCがあれば(自分のPCを持ち歩く必要がなければ)
- ⑦その他【 】

4, 4(4, 1 で③と回答した方にお訊ねします)

利用しない理由は何ですか。あてはまるものをすべて選んで下さい。

- ①ネット接続の必要性を感じないから
- ②通信速度がまだまだ遅いから
- ③PC を持ち歩きたくないから
- ④セキュリティが心配だから
- ⑤会社からネット接続を禁止されているから
- ⑥その他【

】

4, 5 新幹線列車内で備え付けの端末(シートバックもしくは専用スペース)を用いて高速なインターネットサービスができるとしたら利用したいと思いませんか。

以下からひとつ選んで下さい。

- ①利用したい
- ②条件次第では利用したい
- ③利用しない

4, 6(4, 5 で①および②と回答した方にお訊ねします)

新幹線車内で利用する場合にどのような用途に用いますか。あてはまるものをすべて選んで下さい。

- ①メールの送受信(Web メールを含む)
- ②各種予約システム(交通機関、宿泊等)の利用
- ③ブログへの書き込み, Web ページの製作
- ④掲示板への書き込み
- ⑤音声通話サービス, テレビ会議等(たとえば Skype の利用)
- ⑥旅行先, 出張先に関する情報収集(観光情報, 道案内情報等)
- ⑦会社のネットワークへの接続
- ⑧動画の閲覧, アップロード(Youtube 等の動画共有サービス等)
- ⑨インターネットバンキング
- ⑩オンラインゲーム
- ⑪株, 為替等のオンライントレード
- ⑫ニュース, 天気予報等の一般的な Web ページの閲覧
- ⑬その他【

】

4, 7(4, 5 で②と回答した方にお訊ねします)

利用するための条件は何ですか。あてはまるものをすべて選んで下さい。

- ①利用料金がかからない(無料である)ならば
- ②セキュリティ対策がしっかりしているならば
- ③安定した接続が可能ならば
- ④会社のネットワークと接続可能ならば
- ⑤現状のブロードバンドサービスよりもさらに高速なサービスが提供されるならば
- ⑥仕事以外の用途に限定して
- ⑦その他【 】

4, 8(4, 5 で③と回答した方にお訊ねします)

利用しない理由は何ですか。あてはまるものをすべて選んで下さい。

- ①ネット接続の必要性を感じないから
- ②(会社の PC の持ち出し禁止等の理由で)自分が普段使っている PC が使えないから
- ③セキュリティが心配だから
- ④(会社の規則等で)不特定多数が利用する端末では業務利用が禁止(制限)されているから
- ⑤移動中の時間は他の活動(たとえば、読書、睡眠等)で使いたいから
- ⑥その他【 】

4, 9 新幹線列車内で高速インターネットサービスが実現した場合、利用料金の支払い方法はどのような形態が望ましいですか。以下からひとつ選んで下さい。

- ①切符購入時に同時に支払う
- ②列車内で現金、プリペイドカード、ICカード等を用いて支払う
- ③事前にこのサービスの会員登録をし、会費として事前に支払う
- ④無線 LAN サービスの会社にローミング機能を申し込み、支払う
- ⑤その他【 】





## 2.2 調査結果

アンケート調査の回答者の属性を図1～図4に示した。回答者の男女比率は図1に掲げるように男性88.6%、女性11.4%であった。また、アンケート回答者の年齢構成比としては図2に掲げるように30歳～34歳が14.7%、35歳～39歳が19.2%、40歳～44歳が20.5%、45歳～49歳が19.1%、50歳～54歳が12.2%と、30歳～54歳で全体の85.7%を占めていた。また、アンケート回答者の職業種別構成比では図4に掲げる通り会社員(事務系)が28.8%、会社員(技術系)が38.2%、会社員(その他)が16.0%であった。アンケート回答者の男女構成比と年齢構成比及び職業種別構成比の結果からわかるように、97.4%が社会人として働いている方であり、同時に年齢も73.5%が30～49歳であることから東海道新幹線をビジネスの際の移動手段として利用していることが把握される。

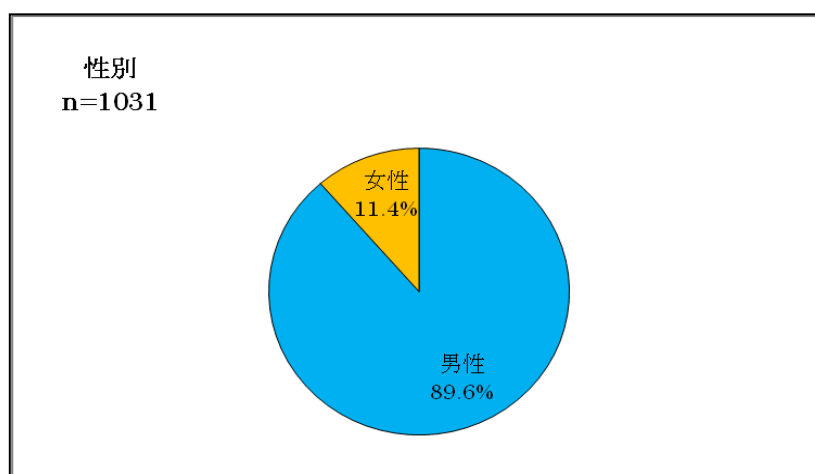


図1：性別構成比

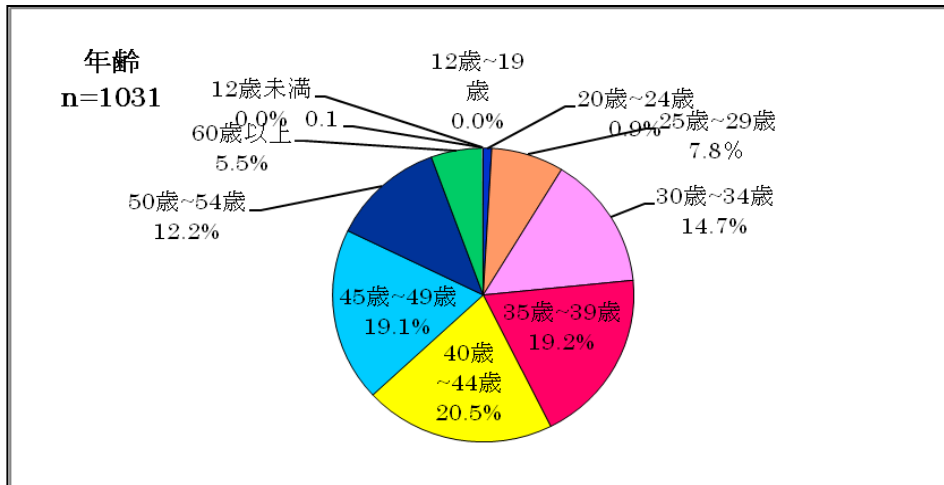


図 2：年齢構成比

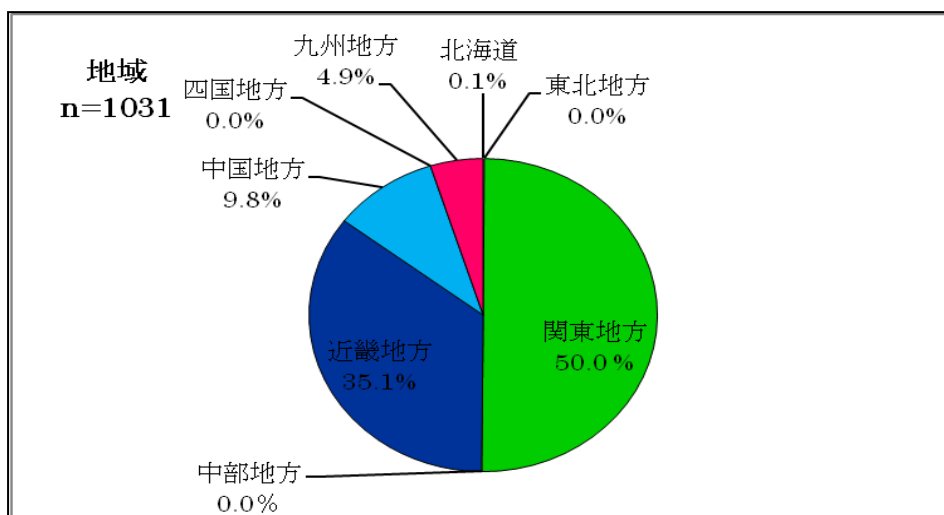


図 3:利用場所構成比

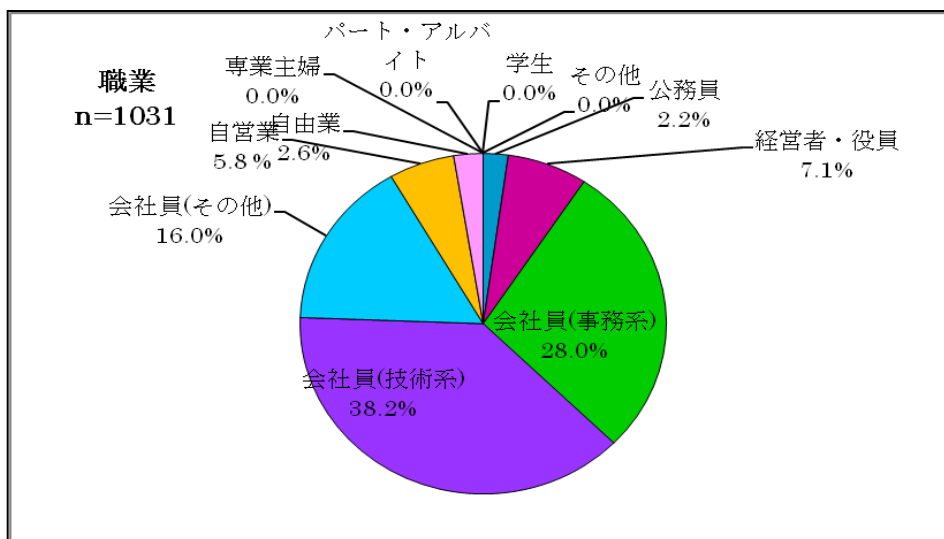


図 4:職業別構成比

ここからはアンケート調査票に記載したアンケート結果を示していく。今回のアンケート調査対象となった新幹線利用者の区間利用率は図5に示したとおり、東京エリア⇄関西エリアが70.0%、東京エリア⇄岡山県が8.3%、東京エリア⇄広島県が10.4%、東京エリア⇄福岡県が11.3%となった。この新幹線利用者が使う国内で使用する移動手段を示したものが図6である。新幹線と航空機の差はほとんどなく、拮抗した状況にある。このような利用者が重視する移動手段の優位性を示したものが図7である。この結果から導き出される列車利用者の重要度としては拠点駅・空港等へのアクセスの良さが94%、所要時間が92.3%、移動空間の居住性・乗り心地が86%、予約・変更のしやすさが83.8%、運賃・料金が80.1%、高頻度利用者向けサービスが56.9%、移動中の通信サービス（携帯電話、インターネットサービス）が41.6%、喫煙可能であるかどうか21.1%となった。

やはり新幹線の利用者は駅・空港等へのアクセスの良さや所要時間、予約・変更のしやすさからわかるように移動や乗降車に関わる時間を第一に考え、次に運賃・料金や高頻度利用者向けサービスからわかるように料金面、そして第三に移動空間の居住性、乗り心地を考え、第四に移動中の通信サービス（携帯電話、インターネットサービス）、第五に喫煙可能であるかを重要視していると推測できる。このことからインターネット通信サービス（携帯電話、インターネットサービス）の列車内インフラストラクチャーとして実現化が必要である。

しかし、駅・空港等へのアクセスの良さや所要時間、予約・変更のしやすさである移動や乗降者に関わる時間、運賃・料金や高頻度利用者向けサービスである料金面は鉄道会社としてのあるべき姿を反映しているが、それに対して、インターネット接続環境への期待が利用者の約半数が重視する傾向にあることはインターネット接続環境整備への期待が高まっていることが顕著に表れている。また、図8ではプライベートな用事で移動する際の重要事項を示しているが、ここで興味深い点がある。それが図7と図8である。両者を比較すると、出張や仕事の用事で移動する際の方がプライベートな用事で移動する場合と比較してより移動中の通信サービスに対するニーズが高いことがわかる。この結果からも、東海道新幹線の利用者



が図 4 に示した通り 80%以上の利用者が勤務中での利用であることから、新たな付加価値サービスとして移動中の通信サービスの需要が高まっていることが把握できる。

図 9 では新幹線利用にあたってのサービス需要を示したものであるが、各座席での高速インターネットサービスの提供・共用ビジネススペースでの高速インターネットサービスの提供の需要は運賃・料金の割引(10%程度)や所要時間の短縮(10分程度)とほぼ同じ需要を示している。これは上記で述べた図 7 や図 8 での需要の差とは異なっている。つまり、新幹線車内の付加価値として求められている需要と新たな付加価値として求められている需要の比率が異なるということである。新幹線車内の付加価値として求められている需要では鉄道本来のインフラストラクチャーとしての機能である運賃・料金、所要時間、移動空間の居住性・乗り心地や拠点駅・空港等へのアクセスの良さへの需要が高いのに対し、新たな付加価値として求められている需要では鉄道本来のインフラストラクチャーとしての機能である運賃・料金、所要時間、移動空間の居住性・乗り心地や拠点駅・空港等へのアクセスの良さへの需要と同様に各座席での高速インターネットサービスの提供や共用ビジネススペースでの高速インターネットサービスの共有というインターネット接続環境という新たなインフラストラクチャーの需要が高まっている結果となった。これは図 10 でも同様の結果が表れている。

図 10 では図 9 でのサービスの結果、新幹線利用頻度の変化を調査したものであるが、各座席での高速インターネットサービスの提供や共用ビジネススペースでの高速インターネットサービスの提供は運賃・料金の割引(10%程度)や所要時間の短縮(10分程度)に比べるとやや低いものの図 9 と同様の結果を示している。以下に利用者が交通機関で移動する際の状況についてアンケートした結果を図 5～図 10 に示す。

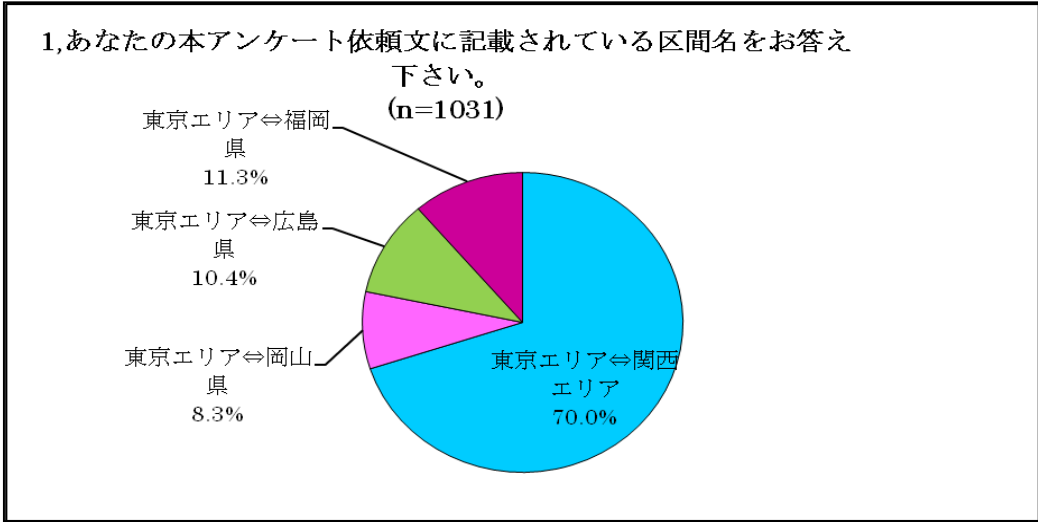


図 5

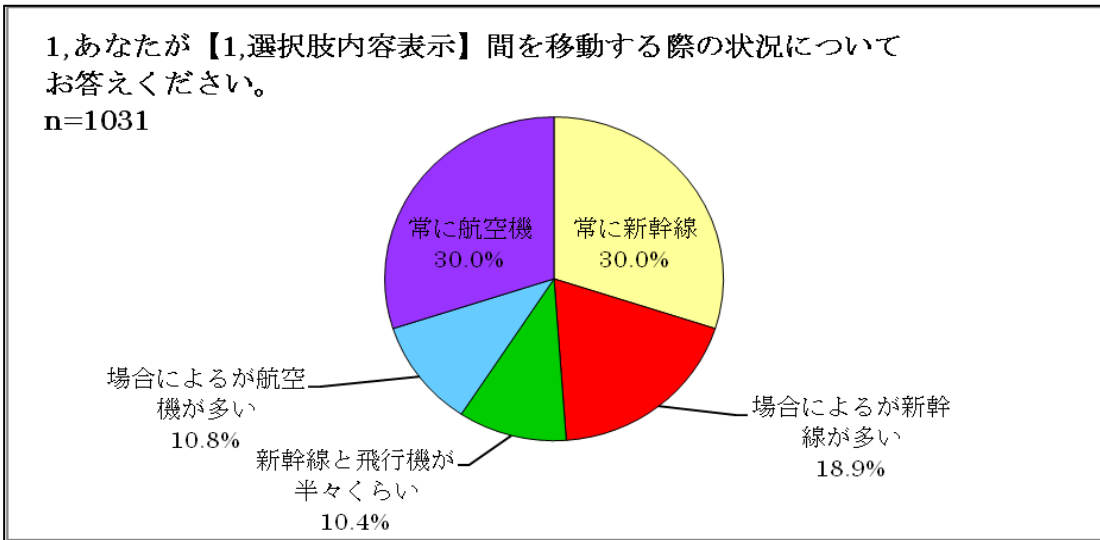


図 6

2,2あなたが【1の選択肢内容表示】間を、出張等、仕事の用事で移動する場合についてお答え下さい。  
 このような場面での交通機関選択にあたって、以下の各項目をどの程度重視しますか。(それぞれひとつずつ)

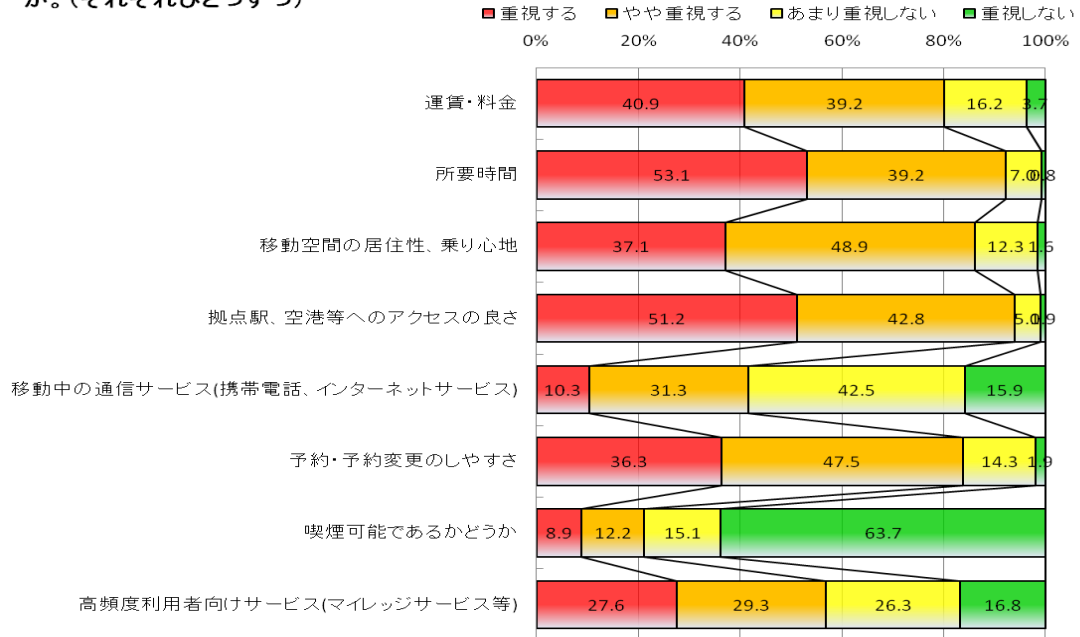


図 7

2,3あなたが【1の選択肢内容表示】間を、家族旅行、帰省等のプライベートな用事で移動する場合についてお答え下さい。このような場面での交通機関選択にあたって、以下の各項目をどの程度重視しますか。(それぞれひとつずつ)

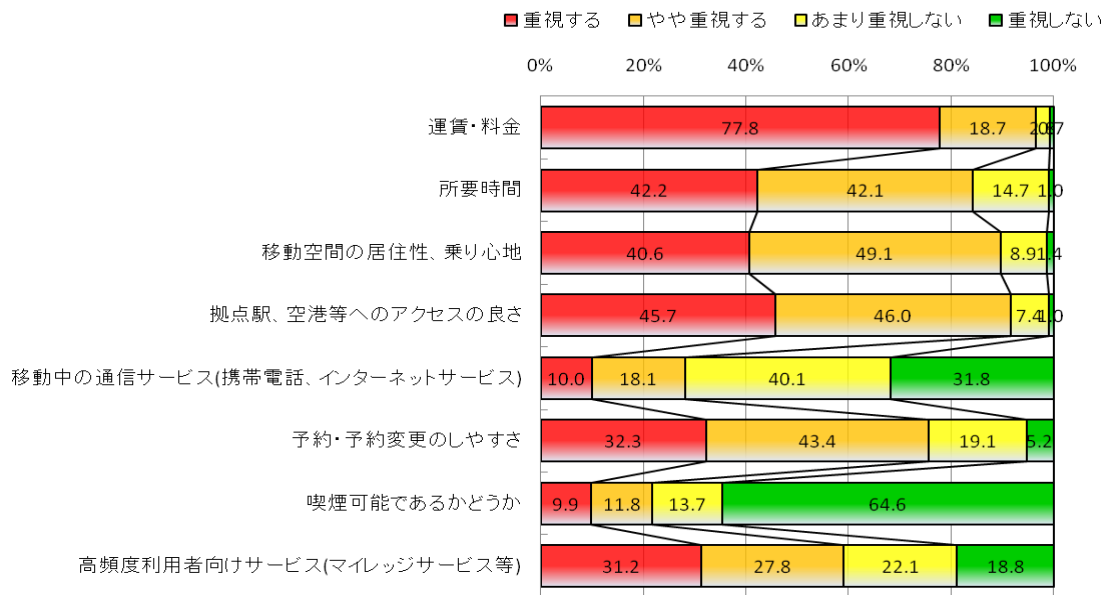


図 8

2,4あなたは【Q1の選択肢内容】間を仕事で移動する場合、それぞれの項目は新幹線を利用するにあたってどの程度魅力的ですか？それぞれあてはまるものをお選びください。

※仕事で新幹線を利用することがない方は、イメージでお答えください。

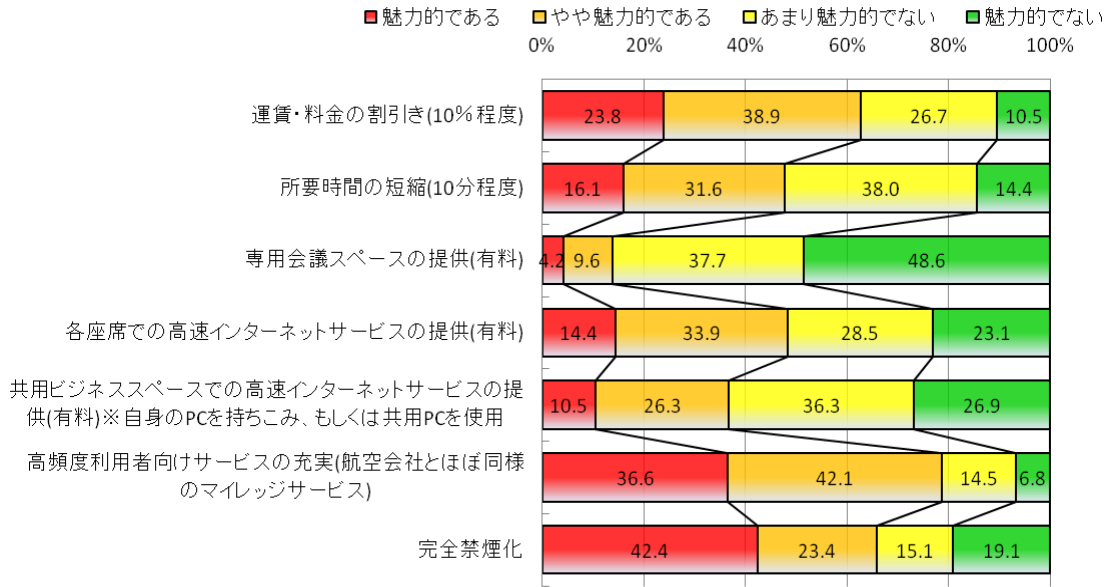


図 9

2,5あなたは【Q1の選択肢内容】間を仕事で移動する場合、それぞれの項目により新幹線利用頻度は変化しますか？それぞれあてはまるものをお選びください。

※仕事で新幹線を利用することがない方は、イメージでお答えください。

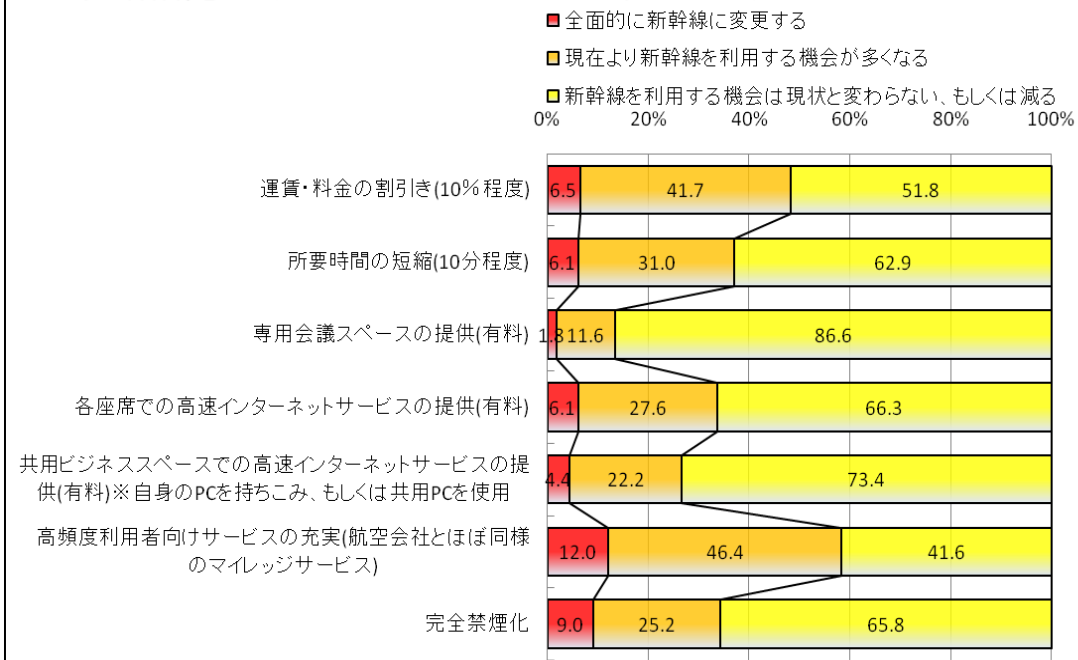


図 10

ここからは東海道新幹線における利用者の割合が業務に際しての利用が多いことより、仕事中や出張に際しての移動手段として利用する利用者に焦点をあてた移動中における PC の利用状況についてのアンケート結果を示していく。

図 11 では現状の新幹線列車内における携帯電話もしくは PHS 端末、PC の利用頻度を表している。携帯電話もしくは PHS はほとんどの新幹線利用者が利用していることが把握できる。しかし、同時に半数以上の新幹線利用者が PC を列車内で利用している。これに関連するアンケート結果として図 12 に新幹線列車内でのインターネット利用度合いを示した。この結果から新幹線列車内における PC でのインターネット使用率がほぼ半数にわたり、図 10 のグラフと照らし合わせると、新幹線列車ないでの PC 使用者の約 8 割の方が PC の使用に付随してインターネットを使用していることが明らかになった。しかし、この列車内での PC の使用とインターネットの使用に関する利用者の満足度を表したものが図 13 と図 14 である。図 13 では新幹線列車内での携帯電話もしくは PHS 端末を用いてインターネットを利用する際の満足度を表しており、図 14 では新幹線列車内での PC を用いたインターネットを利用する際の満足度を表しているが、利用者ともに半数以上の方が不満傾向を示していることが把握できる。これは上記で取り上げた新たな付加価値として求められている需要では鉄道本来のインフラストラクチャーとしての機能である運賃・料金、所要時間、移動空間の居住性・乗り心地や拠点駅・空港等へのアクセスの良さへの需要と同様に各座席での高速インターネットサービスの提供や共用ビジネススペースでの高速インターネットサービスの共有というインターネット接続環境という新たなインフラストラクチャーの需要が高まっている結果となっているにも関わらず、その現状に新幹線利用者が満足していない。しかし、これは新幹線の列車内付加価値として新たな顧客獲得に十分な優位性を示せるものであることが証明できる。また、現状や新たなインターネット接続環境の整備に伴って新幹線利用者の列車内での PC 利用者が今以上に増加する。しかし、上記のアンケート結果でも述べたように新幹線利用者は社内での乗り心地や快適性を重要視している。そのため、新幹線列車内での PC 利用に対する周囲の快適性についての結果

が図 15 である。これは新幹線列車内で自分の周囲の方が PC を使用していることに対する不快感のアンケート結果を示したものであるが、ほとんどの新幹線利用者は PC への不快感がないことが把握できる。以下に仕事中や出張に際しての移動手段として利用する利用者に焦点をあてた移動中における PC の利用状況についてのアンケート結果を図 11～図 15 に示す。

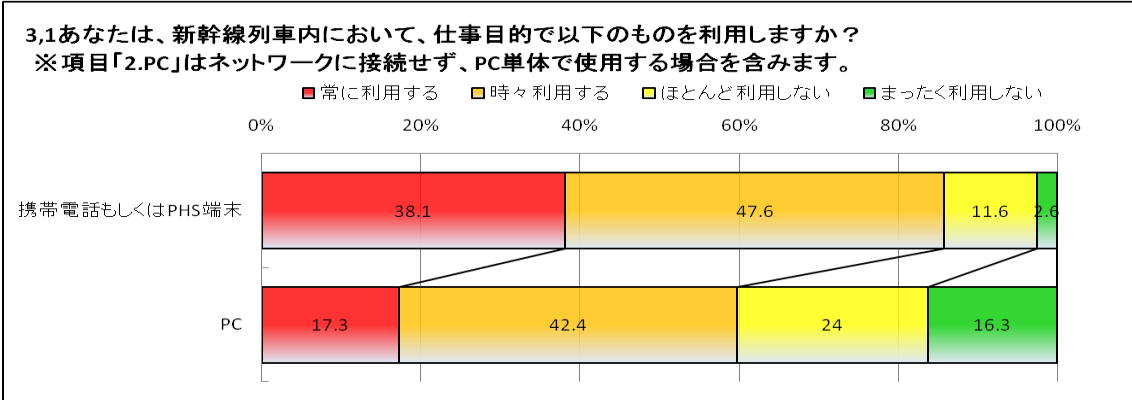


図 11

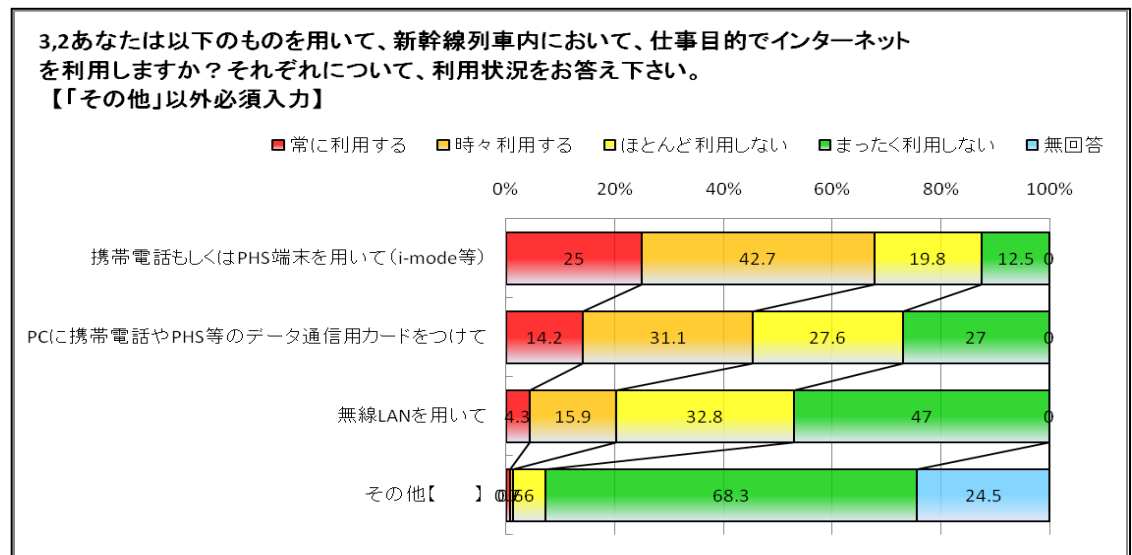


図 12

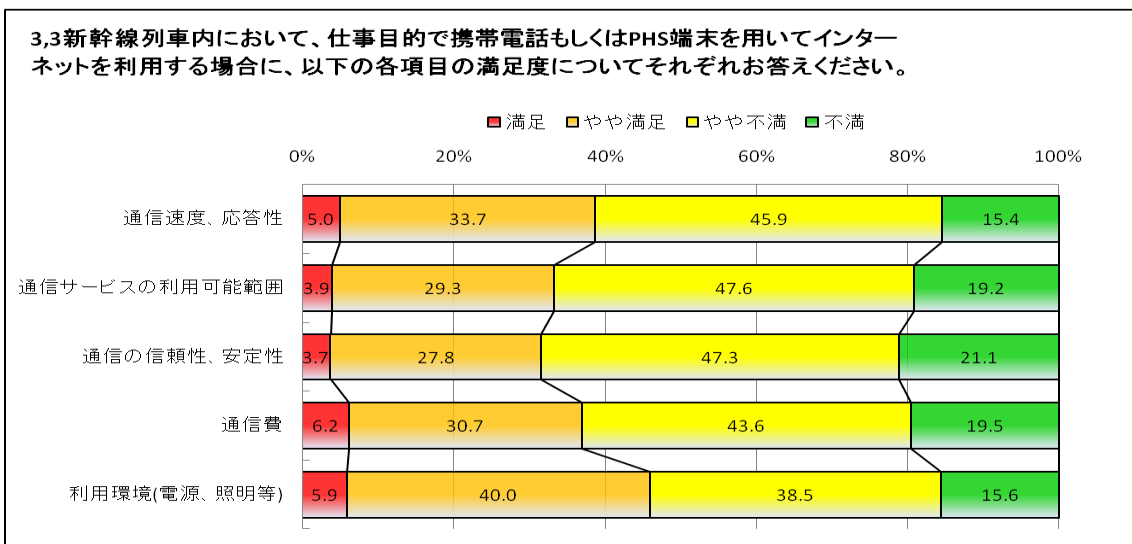


図 13

3,4新幹線列車内において、仕事目的でPCを用いてインターネットを利用する場合に、以下の各項目の満足度についてそれぞれお答えください。

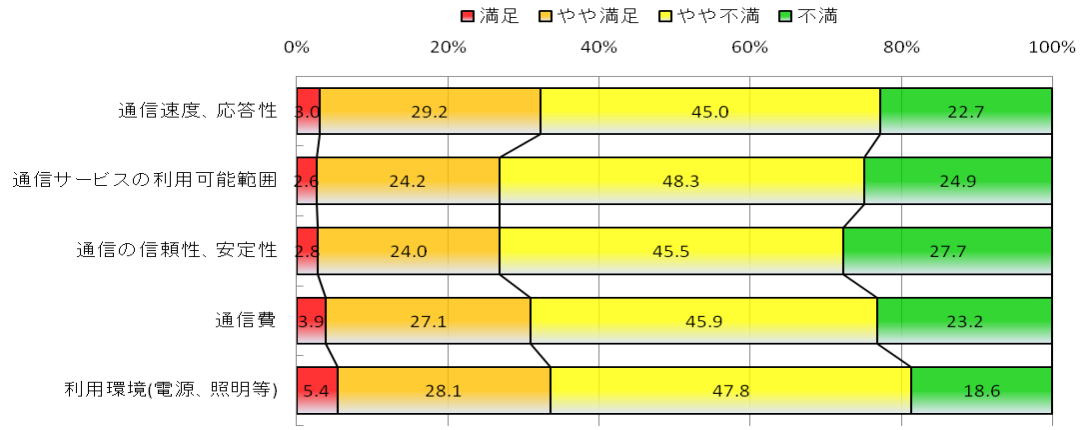


図 14

3,5新幹線車内でご自分の座席の周辺の方がPCを使用している場合にどのように感じますか。あてはまるものを以下からひとつ選んで下さい。

n=722

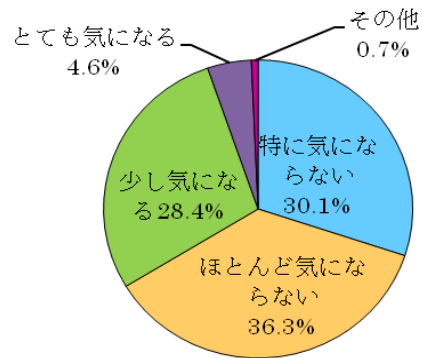


図 15



## 2.3 問題点

ここからは二章で行ったアンケートとその結果から、照明列車向け高速光空間通信システムに実現に向けた問題点の把握を行っていく。上記のアンケート結果でも述べたように列車利用者の列車内空間の重要性として拠点駅・空港等へのアクセスの良さが 94%，所要時間が 92.3%，移動空間の居住性・乗り心地が 86%，予約・変更のしやすさが 83.8%，運賃・料金が 80.1%、高頻度利用者向けサービスが 56.9%，移動中の通信サービス(携帯電話、インターネットサービス)が 41.6%，喫煙可能であるかどうか 21.1%となっている。駅・空港等へのアクセスの良さや所要時間，予約・変更のしやすさである移動や乗降者に関わる時間，運賃・料金や高頻度利用者向けサービスである料金面は鉄道本来のインフラストラクチャーとしての需要であるのに対して，付加サービスとしてのインターネット接続環境への期待が利用者の約半数にも及んでいることがインターネット接続環境整備への期待が高まっていることが顕著に表れているといえる。また，新幹線列車内でのサービスの需要としても各座席での高速インターネットサービスの提供・共用ビジネススペースでの高速インターネットサービスの提供の需要は運賃・料金の割引(10%程度)や所要時間の短縮(10分程度)とほぼ同じ需要を示している。つまり，新幹線列車内の付加価値として求められている需要と新たな付加価値として求められている需要の比率が異なるということである。新幹線列車内の付加価値として求められている需要では鉄道本来のインフラストラクチャーとしての機能である運賃・料金、所要時間、移動空間の居住性・乗り心地や拠点駅・空港等へのアクセスの良さへの需要が高いのに対し，新たな付加価値として求められている需要では鉄道本来のインフラストラクチャーとしての機能である運賃・料金，所要時間，移動空間の居住性・乗り心地や拠点駅・空港等へのアクセスの良さへの需要と同様に各座席での高速インターネットサービスの提供や共用ビジネススペースでの高速インターネットサービスの共用というインターネット接続環境という新

たなインフラストラクチャーへの需要が高まっている。また、現状での列車内でのインターネット環境への満足度としては利用者の半数以上が不満傾向を示していた。

以上のような結果を総称すると、列車利用者の比率が圧倒的に社会人でありビジネスマンの確率が非常に高いことが予想され、列車利用者の重要度としては拠点駅・空港等へのアクセスの良さが 94%、所要時間が 92.3%、移動空間の居住性・乗り心地が 86%、予約・変更のしやすさが 83.8%、運賃・料金が 80.1%、高頻度利用者向けサービスが 56.9%、移動中の通信サービス（携帯電話、インターネットサービス）が 41.6%、喫煙可能であるかどうかは 21.1%となった。この結果を総称するとやはり新幹線の利用者は駅・空港等へのアクセスの良さや所要時間、予約・変更のしやすさからわかるように移動や乗降車に関わる時間を第一に考え、次に運賃・料金や高頻度利用者向けサービスからわかるように料金面、そして第三に移動空間の居住性、乗り心地を考え、第四に移動中の通信サービス（携帯電話、インターネットサービス）、第五に喫煙可能であるかを重要視していると推測できる。このことからインターネット通信サービス（携帯電話、インターネットサービス）の列車内インフラとして実現化が望まれていることが推測できる。

また、現状の列車内でのインターネット接続の利用度を示したアンケート結果を図 2 に示した。インターネットに接続する利用者はほとんど携帯電話もしくは PHS 端末を利用し、次に PC に携帯電話や PHS 等のデータ通信カードをつけて利用している。だが、PC でのインターネット接続利用者の列車内通信における満足度は低いことが確認されている。そこで、本論文では列車内でのインターネット接続環境の普及とともに、列車内での PC 利用頻度の増加が見込まれる状況において、列車利用者の要望が高かった移動空間の居住性・乗り心地に焦点を当てた照明列車向け高速光空間通信システムにおける快適操作を実現するユーザーインターフェースの検討を行った。

ここで、三章から焦点を当てていく移動空間の乗り心地と通信サービスの関連性の現状を述べることにする。現在携帯用パソコンの普及と「乗り物の中でも利用したい」といったニーズの高まりに伴い、一部の鉄道や飛行機ではすでにパソコン利用者を意識したアコ

モーションを導入し始めている。通信サービス実用化に伴い列車内においてもパソコンの使用率が向上することが予想できる。パソコンの使いやすさを評価するには、「振動環境における視覚的表示と手作業のためのデザインガイド」や VDT 構成要素の個人への適合性についての検討事例、視覚表示や作業環境の要件を定めた規格（IS09241 シリーズ）などが参考になる。しかし、ISO 規格では列車のような振動環境下での作業について十分に検証していないし、振動環境を扱った先行研究においても単一周波数の影響についての分析が中心であり、列車走行時のような複合振動を考慮したものはほとんどない。また、鉄道総研報告でも過去において「振動が社内でのパソコン作業に及ぼす影響」でしか報告されておらず、この中においても体格差による影響やテーブル高さの最適値においては検証されておらず、姿勢に伴う座席形状や材質についても検討されていない。鉄道総研報告の中での「列車振動がパソコン作業に及ぼす影響と最適なテーブル位置の検討」の中においても列車設備の問題点について検討している。この中で列車利用者の座席におけるテーブルの位置調節や大きさ等に対する大きさ等に対する不満が高いことを提言している。高速移動列車（新幹線）と地上間の高速通信実現に向けて現状の利用者が重要視するニーズの把握、また同時に現在新幹線内において携帯電話や PHS、パソコンを用いてインターネットを利用する利用者のニーズとその現状の利用状況の把握を行った。その結果、高速移動列車（新幹線）と地上間の高速通信のニーズが十分にあることが検証できた。但し、通信サービスにあたってのパソコン利用の増加や現状のパソコン利用に対する列車内におけるテーブル位置や座席形状・材料等の設備が十分検討されていない。また、パソコンに関するディスプレイに注目していたが、飛行機のシート座席のように新幹線列車内においてもシート座席にディスプレイが装着されることも視野に入れ、パソコン及び座席モニター双方に関する振動と作業・乗り心地に対する検討を行い、テーブル位置や座席形状・材料等の列車内設備の検討が課題である。そこで三章から列車内でのインターネット接続環境の普及とともに、列車内での PC 利用頻度の増加が見込まれる状況において、列車利用者の要望が高かった移動空間の居住性・乗り心地に焦点を当てた照明列車向

け高速光空間通信システムにおける快適操作を実現するユーザーインターフェースの検討を行うこととする。

## 2.4 列車内 PC 操作が人体に及ぼす影響

上記のアンケート結果から新幹線列車利用者の列車内における PC 使用の需要とともに、インターネット接続への需要が高いことが把握でき、今後もこの需要の増加が見込まれることが把握できた。また、同時に PC 利用時の列車内の快適性・居住性の需要も高いことから、列車内の PC 利用に伴う環境の整備が重要であると考えられる。新幹線列車内における PC 利用に伴い、PC 利用が人体に与える不快感要素として映像酔いが考えられる。TV 画面やスクリーン上の映像が不規則に激しく揺れるのを見続けていると、軽く酔ったような感覚に陥ることがある。映像酔いの前兆的な感覚の一種であり、気分が悪い（不快感）とか目が回りそうだ（眩暈感）とを感じるのも、映像酔いの兆候の自覚的な現れの一つである。この兆候は、乗り物酔いを典型とする動揺病の初期症状と似ているため、映像酔いも動揺病の一種に概括されているが、乗り物酔いはその名のごとく乗り物の揺れが外的主要因であるのに対して、映像酔いは視覚系からの情報のみによって誘発されるという点に特徴がある。乗り物酔いの解説では、「各種の乗り物が発する振動が原因で、体の内耳にある三半規管が体のバランスを取れなくなって引き起こす身体の諸症状。」とある。映像酔いに関連して映像情報工学の領域では、近年、高臨場感映像によってもたらされる動揺病と類似の症状が、多数の研究者によって報告されている。これらの酔いは「VE 酔い」あるいは「シミュレータ酔い」と呼ばれ、宇宙飛行士の訓練用の模擬装置や、アミューズメント用の体感ゲーム等の仮想環境 (Virtual Environments) 等の状況下での酔い発生に関する研究が行われている。また、生体は視覚系・前庭系・大性感覚系などからの感覚情報を統合的に処理して安定した空間を通常は認識しており、その感覚情報パターンに基づく外部空間モデルの形成はすでに神経回路網に記憶されているのであるが、それと不一致な外部環境の情報パターンが入力されて矛盾が生じると空間認識に混乱がおり、交換神経系を興奮させて酔いの症状を発生させるという。このような映像酔いが人体に与える影響について

ての先行研究である「映像視聴と生体への影響・森田寿哉、原澤賢充」では SSQ を用いた主観評価実験と、光トポグラフィによる視覚運動の検出実験を紹介しており、酔いの生じやすい映像のグローバルモーションの特徴の一部を明らかにし、また、ディスプレイの大きさのような視聴環境によっても酔いの程度が変わることを示している。また、映像酔いの防止として「サウンドは映像酔いを抑止できるのか・長嶋洋一」や「色情報が映像酔いに与える影響についての考察・瀧口智史」では場面の明るさや色数、きめの細やかさなども映像を特徴付ける重要な要素と捉え、色相、輝度および彩度をそれぞれ個別に設定できるシステムを作成し、色情報が映像酔いに与える影響や従来手法と異なる新しい映像酔い心理学実験手法、映像の予測に役立つような変化のサウンドを加えることで映像酔いが低減できるかどうかという視点で研究を行っているが、色の影響はあるものの、明確な傾向は見出せていなく、また、映像に対応して適切なサウンドを生成提供する必要があることの知見を得られたに留まっており、実用化には及んでいない。しかし、シミュレータ酔いが発症する主要因に視覚系の情報と前庭系や体性感覚系の情報との間の不一致があることはほぼ明白であることから、これまで視点移動のある映像を、静止させた状態の被験者あるいは映像の動きと同調的に動く状態の被験者に観視させるなどの操作を通して、MSQ(Motion sickness Questionnaire)などの主観的評価尺度や、呼吸曲線、心電波計、皮膚電気活動、重心動揺、胃電図など他覚的な生理的指標を用いて、外部要因と内部状態との関係が検討されてきた。他方、心理学の領域において、視覚系運動情報が誘発する自己運動の知覚(self-motion perception)は、誘導運動の一類型として、Dunker(1929)以来、古くからの関心事であった。この類型の誘導運動は、移動する視対象と静止する観察者との間で生じる錯覚的な自己運動知覚であり、その後、自己運動知覚に与える視覚的運動刺激の性質が他側面から追及されてきた。このように、PC 操作に付随する PC モニターの注視が新幹線列車利用者の酔いの発生を高めることが十分に予測される。そのため、PC モニターの注視に付随して発生する映像酔いへの対策が必要である。また、新幹線列車内での PC 操作に付随する酔い発生の原因は PC モニター注視によるものだけではな

い。それが、振動である。「身体および視野の振動が快適さに与える影響・長野茂人、河合俊岳、増田修、河合敦夫、井須尚紀」では、列車や自動車、航空機などの乗り物に乗って一直線に前進している場合でも、乗り物は揺れてロールやピッチ回転が複合した複雑な非周期的振動が生じることから、身体や視野に加わる非周期的振動において、振動の回転軸方向の変動や身体―視野振動の時間差が乗り心地にどのように影響するかを検討している。また、「乗り物酔いの実用的評価法・松本互平、藤井智弘」でも、国際振動基準 ISO2631-1(1997)の一部に記載されている動揺病に関する評価の考え方と特徴について述べ、それに基づいた乗り物酔いの推定法と評価法について検討を行っている。乗り物酔いは人体の平衡感覚をつかさどる三半規管や耳石などの内耳前庭器官の加速度刺激が強すぎたとき、その働きに破綻をきたして発症するものと考えられてきた。しかし、視覚による映像だけで発症する「映像酔い」や宇宙飛行士の大半がかかる「宇宙酔い」など単に加速度刺激だけでは説明できない乗り物酔いの現象が現れ、脳の知覚や記憶に深く関係する複雑なメカニズムであることがわかってきた。一方、遊園地のスリルライドから受ける衝動加速度やスピードは短時間であれば、快適感や達成感につながる。人体が過大な振動環境に暴露されたときの心理的応答は時代とともに変化し、また、現象の理解を難しくしている。動揺(0.1~0.5Hz)と振動(0.5Hz~80Hz)に対する心理的反応は受容器の違いはあれ、快適感や不快感の表れ方において同様な性質をもつ。乗り物を始めとする新しい機械や構造物の出現にともない、その計画や設計において、人間の作業環境や居住性におよぼす振動の影響を評価する方法が必要になっている。全身振動に対する国際基準 ISO2631 は 1974 年に発行されたが、動揺病に対する試案 ISO2631-3 はその一部として 1985 年に始めて表れている。評価方法は基本的には振動に対するものと同一であった。その後、その現象を支配する物理量や人体の許容量に関する研究成果をもとに 1997 年に ISO2631 の全体が改定されたのに伴い、動揺病に対する基準も新しくなった。乗り物酔いによりあらわれる症状は、船酔いに関する統計的観察によれば、頭重、欠伸、頭痛、冷汗、便意、胃部膨満感、顔面蒼白、悪心、ふらつき、嘔吐の発症順序になる。このほか、めまい、胸苦

しき、無力感などの症状を分類に含めることもある。これらの症状は、すべてこの順序で起こるとは限らないし、起こらないものもある。また、発症頻度では、悪心、嘔吐、生唾、頭重、冷汗などが多いことが報告されている。このように、PC 操作に付随する映像酔いだけでなく、それに伴う列車内の振動の影響も考えられ、その抑制方法の検討が必要であると考えられる。そのため、列車内の PC 作業が人体に及ぼす影響として PC 注視による映像酔いと列車内振動の二つの項目を挙げ、この二項目を抑制することによって新幹線列車内で快適な PC 操作が実現できると考えた。そのため、PC 注視による映像酔いと列車内振動の防止策の提案が新幹線列車内での PC 作業に付随する酔い防止につながるという仮説を立て、そのユーザーインターフェースの検討を行った。次章では新幹線列車内の振動原因とその人体への影響を述べる。



## 第3章

# 列車内振動に対する検討と改善方法の提言

### 3.1 列車内振動

上記のアンケート結果からも把握できるように近年ではPCが急速に普及しており、乗り物内でも利用したいというニーズ、特に列車内でのニーズが高まっていることが把握できた。そこで、まず列車内振動について述べる前にPC操作による酔いの影響を述べていく。

「VDT構成要素の個人への適合性に関する研究」ではPC操作が人体に影響する要素に関する先行研究であるが、ワークステーションの主要な要素(椅子、作業台、ディスプレイなど)と作業者との関係が別個に固有の身体測定点と対応して定義されていることからワークステーションと作業者との物理的な関係を主とした寸法と姿勢上の適合性を考え、複数の条件での総合的な実験からVDT構成要素と作業との物理的な関係の正式化を行っている。

また、「VE酔い研究および関連分野における研究の現状」では酔いの発生において、視覚の影響が大きい、なくても発生する。逆に視覚のみでも発生するし、視覚情報により軽減されることもあり、身体への動揺と視覚情報との間に不一致がある場合などで酔いが発生することを示唆している。また、乗り物酔いに関しては振動方向や周波数など物理的な環境要因、あるいは生体の振動伝達特性および感度との関連については一部で生理データによる酔いの判定の試みがあるが、定性的にも不明確な状態であることを示唆している。以上のように、PC操作環境に限った酔いの現象として、酔いの症状はほとんどの場合、内省的なものであり、現状では被験者と観察者の主観評価を症状ごとに点数化して合計点を出す方式が用いられるのが一般的であり、あくまでも相対的に評価する方法が一般的であることが把握できる。以下に酔いの程度の定量化は何種類かの方法があり、参考として以下に示す。ただし、いずれも自覚症状や主観

評価などの内省報告か嘔吐を基準にしており，精度の高い客観的定量化は難しいのが現状である。

- (1)MSI(Motion Sickness Incidence)
- (2)Graybiels scale
- (3)SSQ(Simulation Sickness Question-naire)

ここからは酔いを乗り物に対する症状と限定し，列車内振動についての総論を述べていく。列車は比較的酔いにくい乗り物とされているが，振り子式車両などにおける酔いが話題となっている。現状の新幹線列車では振り子式車両が多く採用されており，比較的振動を減少させる上高速移動する列車での速度を緩めることなく曲がる効果を発揮し，新幹線列車の高速移動を大きく担う役割を果たしている。コンピューター制御による車体傾斜技術が著しく向上している。

この車両が人体への酔いに影響を与えていることについての先行研究として，低周波振動が列車よいに及ぼす影響がある。この先行研究では車内快適性の評価を車内の快適性に影響する諸要因として(振動，騒音，温度，空気清浄度)の各々について7段階の評価尺度を設け，その満足度の評価を求めている。また，酔いの程度の主観評価としても船酔いに関する先行研究においてフィールド調査での信頼性や妥当性が高く評価されている Illness Rating(IR)と呼ばれる下記の4段階尺度(①まったく問題ない，②やや気分が悪い，③かなり気分が悪い，④きわめて気分が悪い)を採用している。そこから，ISO2631-1規格に示されるMSDV指標(motion sickness dose value)を用いて酔いの適用可能性を研究している。これに関する結果は3-2振動と乗り心地及び酔いの関係で述べていく。

新幹線列車に及ぼす酔いとしては振り子式車両の導入によるもの，また，酔いの症状は視覚の影響が大きいが，なくても発生する。逆に視覚のみでも発生するし，視覚情報により軽減されることもあり，身体への動揺と視覚情報との間に不一致がある場合などで酔いが発生する。また，乗り物酔いに関しては振動方向や周波数など物理的な環境要因，あるいは生体の振動伝達特性および感度との関連につ

いては一部で生理データによる酔いの判定の試みがあるが，定性的にも不明確な状態であることを示唆している。

## 3.2 振動と乗り心地及び酔いの関係

この章では振動と乗り心地及び酔いの関係について述べていくが、この場合の乗り心地とはPC作業中に対する列車内振動との相関について言及していく。そのため、新幹線列車と人体への振動の影響ではなく、あくまで列車内振動が列車内でのPC作業中に及ぼす影響について限定し言及していく。

この列車内の振動がPC作業への影響についての先行研究として、「振動が車内でのパソコン作業に及ぼす影響」がある。この研究では、携帯型パソコンの急速な普及に伴い列車内での使用ニーズの高まりから、一部の鉄道や飛行機ではすでにPC利用者を意識したアコモデーションを導入していることから、PCの使いやすさの評価について言及し、PC作業の車内設備の問題点や改善要望を把握し、列車内のテーブル位置に着目することでPC作業に適切な位置の検証を行っている。

PCの使いやすさを評価するには「振動環境における視覚的表示と手作業のためのデザインガイド」やVDT構成要素の個人への適合性についての検討事例、視覚表示や作業環境の要件を定めた規格(ISO9241シリーズ)などが参考になる。しかし、ISO規格では乗り物のような振動環境下での作業について十分に検証していないし、振動環境を扱った先行研究においても、単一周波数の影響についての分析が中心で、列車走行時のような複合振動を考慮したものはほとんどない。

この「振動が車内でのパソコン作業に及ぼす影響」では列車走行時の振動環境下におけるPC作業を対象に、具体的な設備改善に役立てるための検討を行っている。そのため列車内でのPC利用者に対してアンケート調査を実施し、現在の車内設備の問題点や改善要望を把握し、調査結果から、列車内でのPC作業時の最適なテーブル位置を検証するための実験を実施している。

結論としては付属テーブルの位置調整や大きさ・強度などに対する不満やテーブルの奥行に余裕がない場合は、座面中央からテーブ

ル手前端までの最適な距離として 40%の被験者が 80mm±5mm に設定すること、パソコンの手前側にテーブル設置空間が確保できると、テーブルに対する満足度が向上し、また作業成績が高まり生理負担も軽減されたことから、テーブルの奥行は重要な要素であること、テーブルの前後位置の最適値は個人差が大きく、体格が比較的揃っていたとしても 80%が満足するには±75mm 範囲のスライド調節が必要であったことが把握されている。いずれにしろ、この先行研究ではテーブル位置という列車内設備の改善を通して列車内振動がPC作業に及ぼす影響を軽減する方法を検討している。

また、上記でも述べた「低周波振動が列車酔いに及ぼす影響」では、振動が人体に及ぼす影響として、前後振動、左右振動が人体に及ぼす影響が相関性として高いことを示しており、上下振動に関する相関関係が極めて低いことを示している。(図 16)また、振子式車両が非振子式車両よりも人体の酔いに及ぼす影響が高く(図 17)、乗車時間が長くなるにつれ、気分の悪い乗客の割合が微増し、振動の時間累積値を用いた評価指標の開発が有効であること、酔いが個人の主観的要素が占める割合が高いこと、乗車目的、同行人数、当該列車の利用経験などの要因が酔いの発現に及ぼす影響は認められないことを示している。また、酔いの発現に最も影響するのは 0.25～0.32Hz の低周波左右振動であることを示唆している。

「振動環境における視覚的表示と手作業のためのデザインガイド」では全身振動、ディスプレイ振動、及び全身とディスプレイの同時振動の異なる観察条件の影響が観察距離及び他の変量に依存し、典型的な視準されていない飛行機のディスプレイと観察者の同時的振動が最小の作業現象を生じさせ、また、静止した観察者で、振動しているディスプレイでは、最大の減少となることが推測されている。

以上のように列車内振動が列車内でのPC作業中に及ぼす影響としては左右振動が最も大きく人体に作用し、次いで、前後運動、上下運動に関しては人体に影響をほとんど及ぼすことなく、考慮するに乏しいことが明らかになった。また、PC作業には作業に要する設備、列車内でいえば作業台の位置関係、座面の高さやテーブル位置との距離である。

しかし、同時にこれらの要素の標準化としては個人の性別や体格差による範囲が広いため設備が固定されている状況下での実現は不可能であることが把握できる。具体的にテーブルの前後位置の最適値は個人差が大きく、体格が比較的揃っていたとしても80%が満足するには±75mm 範囲のスライド調節が必要であったことであるが、±75mm の範囲を実現するにはスライド調節が不可欠でありこれでは、PC 作業の影響を低減するには対象となる人体が少なすぎてしまう結果をもたらすことになる。

ここで、テーブル位置に関する先行研究として「VDT 作業台の最適高さの研究」がある。この先行研究では VDT(Visual Display Terminal)の普及が顕著であることからその作業姿勢を決定する机(以下、作業台という)についての研究が十分になされていないことから、VDT 作業に不可欠なキーボードのキー操作、および CRT(Cathode Ray Tube)ディスプレイ画面の注視作業に見合った作業台の高さ、ならびに椅子の高さについて検討を加え、作業台高さ設定法の提案を行っている。これは作業者の負担を軽減する目的で作業構造の最適化をはかることに主眼をおいている。

VDT 作業台の高さ設定法に従い、身体寸法値各部位(座位)のパーセントイル値に基づくキーボード、および CRT ディスプレイ置台の寸法範囲を算出・設定し、実際における作業台使用にほぼ等しい結果を得ているが、これにも人体の体格差による影響を広く覆うことはできていない。

また、VDT 作業台はキーボード置台、CRT ディスプレイ置台個別の調整が必要であり、その調整も簡便なものが必要とされている。その適切な VDT 作業台の差尺は  $1/3$  座高-(4~6)cm としている。

振動と乗り心地及び酔いの関係としては人体の位置に対する作業台の位置関係、人体の位置を決める座面の位置関係、列車からの振動、さらに、PC 作業に関わる視覚の影響が大きく作用していくことが把握できる。

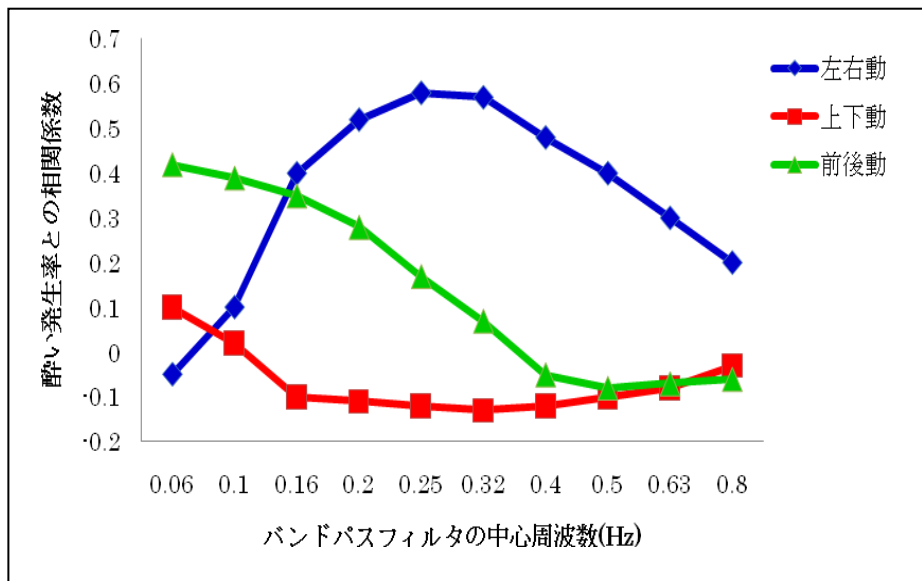


図 16: 低周波振動と酔い発生率(MR)との関係  
 (「低周波振動が列車酔いに及ぼす影響」より引用)

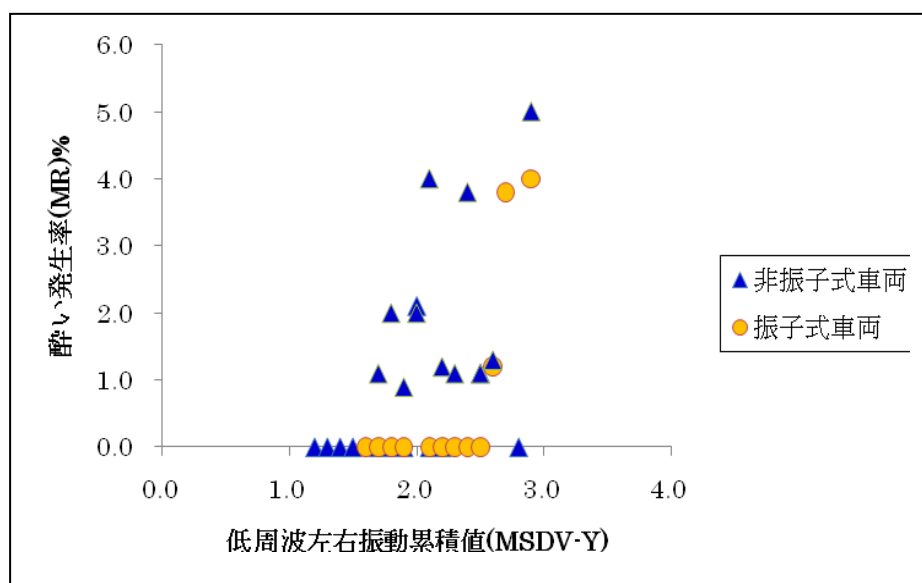


図 17: 振子式車両と非振子式車両の酔い発生率の相関  
 (「低周波振動が列車酔いに及ぼす影響」より引用)

### 3.3 振動に対する対策の現状

列車内振動に対する現状としては大きく分けて三種ある。一つ目に振り子式車両によるもの。二つ目に作業台及び座面に起因するもの。三つ目にPC作業による視覚情報に起因するものである。

まず、一つ目の振り子式車両によるものから説明する。列車は比較的酔いにくい乗り物とされているが、振り子式車両などにおける酔いが話題となっている。コンピューター制御による車体傾斜技術が著しく向上したことに起因し、現状の新幹線列車では振り子式車両が多く採用されており、新幹線列車の高速移動を大きく担う役割を果たすことによって、比較的振動を減少させる上高速移動する列車での速度を緩めることなく曲がる効果を発揮している。この車両が人体への酔いに影響を与えていることに関しての先行研究として、上記でもあげた「低周波振動が列車よいに及ぼす影響」がある。

この研究では、車内快適性の評価を車内の快適性に影響する諸要因として(振動、騒音、温度、空気清浄度)の各々について7段階の評価尺度を設け、その満足度の評価を求めている。また、酔いの程度の主観評価としても船酔いに関する先行研究においてフィールド調査での信頼性や妥当性が高く評価されている Illness Rating(IR)と呼ばれる下記の4段階尺度(①まったく問題ない、②やや気分が悪い、③かなり気分が悪い、④きわめて気分が悪い)を採用している。そこから、ISO2631-1 規格に示される MSDV 指標(motion sickness dose value)を用いている。そこから用いられた結果として、左右振動が大きく人体の酔いに対して影響を与えていることが述べられている。上下振動は左右振動まではいかないまでも多少の酔いを引き起こす要因を持っており、上下運動に関してはほとんど人体の酔いへの影響がないことが把握できる。また、人体の酔いの発現に最も影響するのは0.25~0.32Hzの低周波左右振動である。

一つ目の振り子式車両による影響に関する研究の人体への酔いへの影響が示唆されているものの、現状把握だけであり、対策が全くされていないことが現状である。



次に二つ目の作業台及び座面に起因するものである。「振動車内でのパソコン作業に及ぼす影響」において、付属テーブルの位置調整や大きさ・強度などに対する不満やテーブルの奥行に余裕がない場合は、座面中央からテーブル手前端までの最適な距離として40%の被験者が80mm±5mmに設定すること、パソコンの手前側にテーブル設置空間が確保できると、テーブルに対する満足度が向上し、また作業成績が高まり生理負担も軽減されたことから、テーブルの奥行は重要な要素であること、テーブルの前後位置の最適値は個人差が大きく、体格が比較的揃っていたとしても80%が満足するには±75mm範囲のスライド調節が必要であったことが把握されている。

また、「VDT作業台の最適高さの研究」ではVDT(Visual Display Terminal)の普及が顕著であることからその作業姿勢を決定する机(以下、作業台という)についての研究が十分になされていないことから、VDT作業に不可欠なキーボードのキー操作、およびCRT(Cathode Ray Tube)ディスプレイ画面の注視作業に見合った作業台の高さ、ならびに椅子の高さについて検討を加え、作業台高さ設定法の提案を行っている。これは作業者の負担を軽減する目的で作業構造の最適化をはかることに主眼をおいている。VDT作業台の高さ設定法に従い、身体寸法値各部位(座位)のパーセンタイル値に基づくキーボード、およびCRTディスプレイ置台の寸法範囲を算出・設定し、実際における作業台使用にほぼ等しい結果を得ているが、これにも人体の体格差による影響を広く覆うことはできていない。また、VDT作業台はキーボード置台、CRTディスプレイ置台個別の調整が必要であり、その調整も簡便なものが必要とされている。その適切なVDT作業台の差尺は $1/3$ 座高-(4~6)cmとしている。これらの研究において作業台に関する人体との位置関係についての酔いに対する影響が低い位置の検討はされている。しかし、人体の性別や年齢、個人差の問題によって人体の体格差を広く満たすことのできる作業台は不可能に近い。また、作業台をスライド式にさせるには列車内すべての座席に対応させることとなり、これを新幹線列車内に対応させることはコストや時間の面から非現実的であると考えられる。

最後に三つ目のPC作業による視覚情報に起因するものである。「VE酔いおよび関連分野における研究の」現状では酔いの原因の一つと

して視覚をあげている。当然視覚以外の原因にも言及されているが、酔いの発生において視覚の影響は大きく、視覚のみでも酔いが発生する十分な要因になりうることを示唆している。特に身体への動揺と視覚情報との間に不一致がある場合に酔いが発生することをあげている。

また、広範囲の視野に情報を与える動きのある映像は(自己運動知覚：自分が動いているような錯覚)を生じ平衡の感覚に影響することや、広範囲の視野に情報を与える動きのある映像は平衡感覚に影響し、映像による視覚情報と生体感覚情報のミスマッチにより不快・酔いを生じさせることがあることや実空間でのある物体(例えば椅子など)の大きさと映像で提示されたその物体の大きさとの比率が著しく異なる映像は違和感や酔いを生じることが示唆されている。また、動作物体を負うことによる視覚の微動が平衡感覚を麻痺させることで酔いを発生させると述べられている。これはPCの作業上不可避であり、特に列車内でのPC作業はキーボードタイピングでの視覚を集中させた内容である、また、動画閲覧による視覚への影響が大きい場合が十分に考えられる。しかし、どの先行研究においても原因は特定できてもそれに対する対応について検討されたものはなく、未だ十分に検討されていない課題となっている。

また、「手ぶれ映像に含まれる映像酔いを引き起こしやすい振動成分の解析」では手ぶれ映像に含まれる振動について、映像酔いを引き起こしやすい振動の周波数と振幅を調査することで、映像酔いの改善につなげる検討結果を示している。具体的には、映像を視聴している際に引き起こされる吐き気、めまい、頭痛、眼精疲労などの症状を指しており、体の揺れではなく、運動する視覚刺激に対する原因追究を示している。特に、手持ちカメラで撮影された手ぶれを多く含む映像は、強い酔いを引き起こす可能性があることから、手の細かい震え、撮影しながらの歩行、急激なカメラモーションの多用などにより、様々な方向、周波数、振幅を含んでいる。この「手ぶれ映像に含まれる映像酔いを引き起こしやすい振動成分の解析」ではこのような手ぶれ映像に含まれる振動について、映像酔いを引き起こしやすい映像酔いの程度の主観評価を示している。

しかし、この「手ぶれ映像に含まれる映像酔いを引き起こしやすい

振動成分の解析」では映像自体の乱れによる人体の影響, すなわち, 酔いの発生を示したものである。したがって, 本稿の列車内振動の影響による人体への影響, すなわち, 酔いの発生とは発生源に決定的な違いがみられる。それは, 列車内振動がディスプレイ自体の映像に関しては全くぶれを想定しておらず, 列車運行に付随する振動を発生源としたぶれであるのに対して, 「手ぶれ映像に含まれる映像酔いを引き起こしやすい振動成分の解析」では映像自体のぶれに対する検討であり, 映像外の外部要因については全く考慮していない点で大きく異なっている。そのため, 酔い発生に対する対策であるという点では合致しているが, 酔いの発生原因が大きく相違する点でその対策の有用性も全く異なってくる。以上のように過去の先行研究においては, 列車内での PC 操作に対する酔いの発生原因の調査は行われているものの, その対策として明確に効果を発揮する研究がおこなわれていないことが現状である。また, 日本国特許庁に登録されているヘッドマウントディスプレイ装置では頭部を動かしても, 目で見ている画像があたかも, 所定の位置に固定されているように見える装置の開発がされているが, これにも問題点がある。ただし眼球の輻輳角と焦点距離に差が出るため, 逆に眼精疲労の原因となることが危惧されている。以下に, このヘッドマウントディスプレイ装置による酔い防止映像の体型を示す。(図 18)



### 3.4 振動に対する対策方法の提案

列車内振動に対する現状としては大きく分けて三種あると述べた。具体的には一つ目に振子式車両によるもの。二つ目に作業台及び座面に起因するもの。三つ目にPC作業による視覚情報に起因するものである。しかし3.3でこれらの原因に対する対策の現状はどれも新幹線列車内に対して適用することはコストの面からも時間の面からも不可能に近いと考えられる。

そこで私がこの新幹線列車内のPC作業に付随して発生する酔いの提言方法として提案する方法は作業台における振動に付随して振動とは逆の方向(上下・左右)にウィンドウを振動させることである。

3-3で述べたように、酔いの発生において視覚の影響は大きく、視覚のみでも酔いが発生する十分な要因になりうることを示唆している。特に身体への動揺と視覚情報との間に不一致がある場合に酔いが発生することをあげている。また、広範囲の視野に情報を与える動きのある映像は(自己運動知覚：自分が動いているような錯覚)を生じ平衡の感覚に影響することや、広範囲の視野に情報を与える動きのある映像は平衡感覚に影響し、映像による視覚情報と生体感覚情報のミスマッチにより不快・酔いを生じさせることがあることや実空間でのある物体(例えば椅子など)の大きさと映像で提示されたその物体の大きさとの比率が著しく異なる映像は違和感や酔いを生じることが示唆されている。

また、動作物体を追うことによる視覚の微動が平衡感覚を麻痺させることで酔いを発生させると述べられている。これはPCの作業上不可避であり、特に列車内でのPC作業はキーボードタイピングでの視覚を集中させた内容である、また、動画閲覧による視覚への影響が大きい場合が十分に考えられる。しかし、どの先行研究においても原因は特定できてもそれに対する対応について検討されたものはなく、未だ十分に検討されていない課題となっている。

この方法はウィンドウを作業台の振動と逆方向に際して振動させることで、列車振動時にも作業者が注視しているモニターの一部と

視覚との位置がずれることなく、視覚を動かすことなくモニターを注視することができる。また、動画閲覧時にも、本来動画を追う視覚変動に付随して振動が発生することにより、酔いを発生させるが、ウィンドウを振動させることで動画本来の視覚変動に収縮させることができ、列車内振動を低減できると考えられる。また、振子式車両の改善や作業台の変更・改善と比較するとウィンドウを振動に反応させ振動させることはコスト・時間の面からも早期に実現することができ、効果としても十分に期待できる。

ウィンドウ振動の具体的方法として、加速度センサーを用いた方法を検討した。加速度センサーは本来三軸構造から成り立っているが、そのうち二軸を利用することで可能となる。加速度センサーの二軸をPCのウィンドウの左右・上下に対応するようウィンドウに設置し、加速度センサーの検出値から距離を算出し、ウィンドウのピクセル値に変換し、それに対応させウィンドウを動かすことが可能となる。

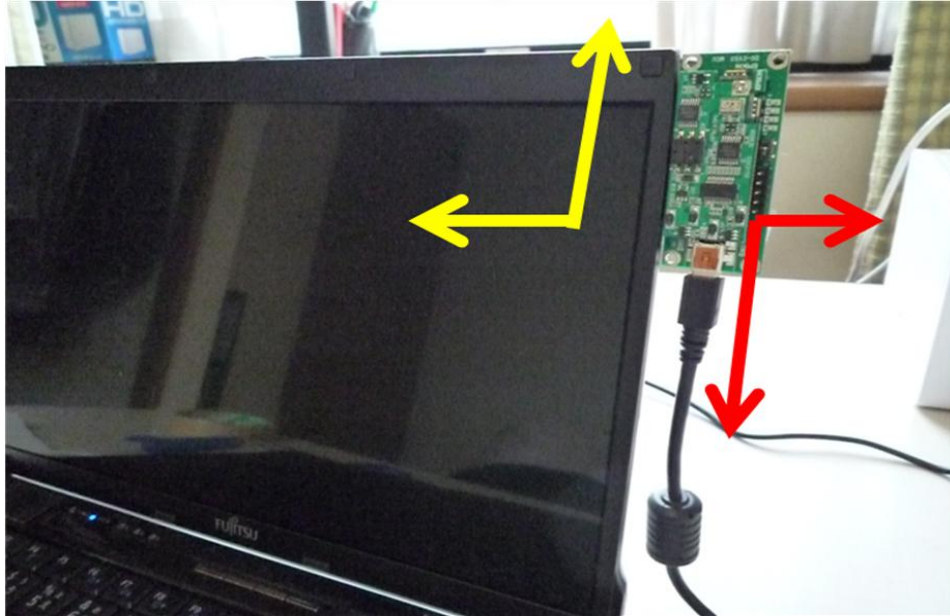


図 19:イメージ

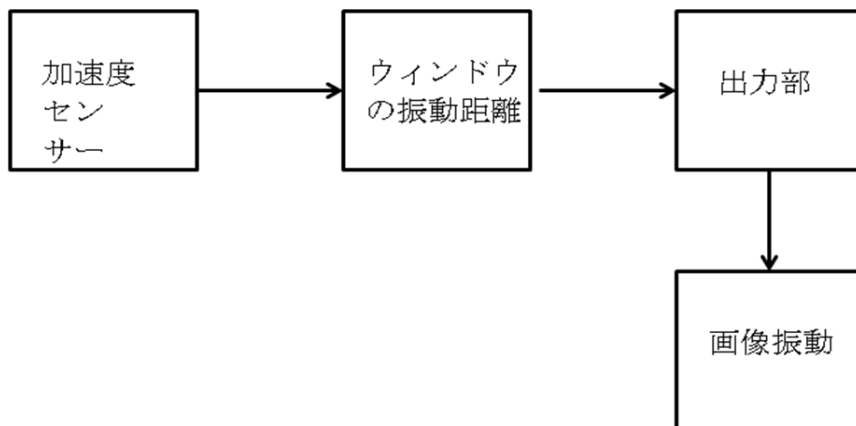


図 20:画面振動防止ソフトウェアの振動出力図

## 第4章

### 結論

振子式車両によるもの。作業台及び座面に起因するもの。PC 作業による視覚情報に起因するもの。これらの原因に対する対策の現状はどれも新幹線列車内に対して適用することはコストの面からも時間の面からも不可能に近い。

そこで私がこの新幹線列車内の PC 作業に付随して発生する酔いの提言方法として提案する方法は作業台における振動に付随して振動とは逆の方向(上下・左右)にウィンドウを振動させることである。

ウィンドウ振動の具体的方法として、加速度センサーを用いた方法を検討した。加速度センサーは本来三軸構造から成り立っているが、そのうち二軸を利用することで可能となる。加速度センサーの二軸を PC のウィンドウの左右・上下に対応するようウィンドウに設置し、加速度センサーの検出値から距離を算出し、ウィンドウのピクセル値に変換し、それに対応させウィンドウを動かすことが可能となる。

結果として、視覚を動かすことなくモニターを注視することができる。また、動画閲覧時にも、本来動画を追う視覚変動に付随して振動が発生することにより、酔いを発生させるが、ウィンドウを振動させることで動画本来の視覚変動に収縮させることができ、列車内振動を低減できると考えられる。また、振子式車両の改善や作業台の変更・改善に比較するとウィンドウを振動に反応させ振動させることはコスト・時間の面からも早期に実現することができ、効果としても十分に期待できる。また、画面振動防止ソフトウェアの動作確認を行うことができた。

列車内の PC 作業が人体に及ぼす影響として PC 注視による映像酔いと列車内振動の二つの項目を挙げ、この二項目を抑制することによって新幹線列車内で快適な PC 操作が実現できると考えた。そのため、PC 注視による映像酔いと列車内振動の防止策の提案が新幹線列



車内での PC 作業に付随する酔い防止につながるという仮説を立て、そのユーザーインターフェースの検討・実装を行った。本研究が今後新幹線列車内における PC 操作に伴う酔い発生の防止策構築の一端を担う手助けになれば幸いである。

## 謝辞

この度、私の修士論文作成にあたり本研究室の春山真一郎教授には、数々のご指導、ご助言を賜りました。深く感謝致します。そればかりか、社会人としての礼儀、持つべき知識、また社会生活での人と人との繋がり大切さなど数々の事を教えて頂きました。

また、手配・進行、論理的思考や論文の書き方まで丁寧に教えて頂きました。そして、研究以外にも普段の生活、コミュニケーションの中で先生方から学ばせて頂いたことは卒業した後の人生において必ず自分を救っていつてくれることと思います。本論文は、先生の深甚ご指導を賜り完成することができました。ここに心より御礼申しあげます。

浦邊秀樹氏・田中稔也氏には、その豊富な知識と斬新な視点からレベルの高いアドバイスを頂きました。また、浦邊秀樹氏の実験に対する姿勢には何度も勇気付けられました。それとともに、ひとつの目標に向かって純粋に突き進む人柄にいつか自分もなれるよう努力していきたいと思いました。

森谷英一郎氏・正源司智博氏には、修士論文作成での授業の上での鋭い指摘・意見を頂いたり、私の拙い質問にも快く答えて頂いたり大変お世話になりました。充実した学生生活を送れたのは人生の諸先輩方のお陰だと思っています。

本研究で勉学に励み、皆様と苦楽を共にしたこの2年間、私自身大きく成長することができたと感じています。今後において、本研究室で過ごしたということがいつか大きな財産になると信じています。

以上をもって、謝辞とさせていただきます。

## 参考文献

- [1]白戸 宏明, 中川 千鶴, 鈴木 浩明: “車内快適性シミュレータの開発と活用法”, RTRI REPORT Vol. 18, No. 2, 2004. 2
- [2]高井 秀之, 矢澤 英治: “車両動揺推定システムの開発と軌道整備への適用”, 鉄道総研報告, Vol. 8, No. 11, pp37~42, 1994
- [3]鈴木 浩明, 白戸 宏明, 田中 綾乃, 手塚 和彦, 仲川 滋: “車体傾斜車両の緩和曲線走行時の乗り心地評価法”, 鉄道総研報告, Vol. 13, No11, pp33~38. 1999
- [4]鈴木 浩明, 白戸 宏明, 田中 綾乃, 手塚 和彦, 高井 秀之: “曲線区間の乗り心地の許容限度”, 鉄道総研報告, Vol. 14, No12, pp19~24, 2000
- [5]中川 千鶴, 大須賀 美恵子: “VE 酔い研究および関連分野における研究の現状”, TVRSJ Vol. 3, No2, 1998
- [6]中川 千鶴, 鈴木 浩明: “振動が車内でのパソコン作業に及ぼす影響”, 鉄道総研報告, Vol. 19, No1, 2005. 1
- [7]Moseley, M. J., Griffin, M. J., 松本 忠雄: “振動環境における視覚的表示と手作業のためのデザインガイド 第1部 視覚的表示”, 労働科学, 64(9), pp422~434, 1988
- [8] Moseley, M. J., Griffin, M. J., 松本 忠雄: “振動環境における視覚的表示と手作業のためのデザインガイド 第2部 手作業”, 労働科学, 64(9), pp435~447, 1988
- [9]戸上 英憲, 野呂 影勇: “VDT 作業台の最適高さの研究”, 人間工学, 23(3), pp155~162, 1987

- [10]平沢 尚毅, 野呂 影勇, 戸上 英憲: “VDT 構成要素の個人への適合性に関する研究”, 人間工学, 26(2), pp87~93, 1990
- [11]鈴木 浩明, 白戸 宏明, 手塚 和彦: “低周波振動が列車酔いに及ぼす影響”, 鉄道総研報告, 18(2), pp9~14, 2004
- [12]柿崎 祐一, 他: “両側性転移”, 心理学実験始動研究会, 実験テスト=心理学の基礎, 実習編, 培風館, pp49~51, 1985
- [13]小原 二郎: “暮らしの中の人間工学”, 実教出版, 139~141, 1971
- [14]鈴木 浩明, 白戸 宏明, 中川 千鶴, 大野 央人: “乗り物酔いの評価に関する研究の現状と課題”, 鉄道総研報告, Vol. 12, No11, pp. 1~6, 1998
- [15]鈴木 浩明, 白戸 宏明, 手塚 和彦: “列車内における乗り物酔いに影響する振動特性”, 人間工学, Vol. 39, No. 6, 2003
- [16]鈴木 浩明: “鉄道車両の振動乗り心地評価のための実験方法”, 騒音制御, Vol. 26, No6, pp364~368, 2002
- [17]中川 千鶴, 鈴木 浩明: “生理的手法による乗り物酔い発症評価の検討”, 鉄道総研報告, Vol. 14. No12, pp25~30, 2000
- [18]鈴木 浩明, 手塚 和彦, 高井 秀之: “鉄道車両の乗り心地評価法と国際標準化”, 鉄道総研報告, Vol. 16, No. 1, pp5~10, 2002
- [19]井上 哲理: “眼球運動と調節の視覚特性”, テレビジョン学会誌, 50-4, p423-428, 1996
- [20]狩野 千鶴: “自己運動知覚と視覚系運動情報”, 心理学評論, 34-2, p240~255, 1991
- [21]佐藤 隆夫: “両眼立体視と運動立体視の統合過程に関する検討”, Human Interface N&R, 12, p223~230, 1997

- [22]中村 学,村田 浩之,緒方 誠人,森本 一人,黒川 隆夫:“現実・仮想融合空間における視知覚の統合化”,Human Interface N&R, 12, p151~156, 1997
- [23]小原 二郎,大内 一雄,寺門 弘道:“差尺に関する研究”,人間工学, 3(2), 159~165, 1967
- [24]大西 徳明:“VDT 作業による筋疲労”,労働の科学, 37(5)p17~22, 1985
- [25]酒井 一博:“オフィスの人間工学”,労働の科学, 34(3), 10~17, 1979
- [26]戸上 英憲,野呂 影勇:“作業姿勢調査に基づく VDT 作業台の高さの提案”,人間工学, 21, 156~157, 1985
- [27]松田 隆夫,大中 悠紀子:“「映像酔い」の自覚的評価とその誘発要因”,立命館人間科学研究, 第 9 号, 2005. 3
- [28]長嶋 洋一:“サウンドは映像酔いを抑止できるのか”
- [29]瀧口 智史:“色情報が映像酔いに与える影響についての考案”
- [30]長野 茂人,河合 俊岳,増田 修,河合 敦夫,井須 尚紀:“身体および視野の振動が快適さに与える影響”,第 8 回情報科学技術フォーラム, FIT2009
- [31]松本 互平,藤井 智弘,“乗り物酔いの実用的評価法”,Mem. School. B. O. S. T. Kinki University No. 12:37~47(2003)



```

pstParam->vpClient );
    }

    while ( *pstParam->bpLoop ) {}

    return iRet;
}

```

## ファイル名 (AccSample.cpp)

```

#include "stdafx.h"
#include "AccSample.h"

static int acc_command_start( HANDLE &hComm );
static int acc_send( HANDLE &hComm, const char *command );
static int acc_receive( HANDLE &hComm, char *buff, int buffSize );

int acc_open( HANDLE &hComm, const _TCHAR *comName )
{
    int iRet = ACC_NG;

    HANDLE handle = ::CreateFile( comName,

    GENERIC_READ |
    0,
    NULL,
    OPEN_EXISTING,
    FILE_ATTRIBUTE_NORMAL,
    NULL);

    if ( handle != INVALID_HANDLE_VALUE ) {
        DCB dcb;

        ::ZeroMemory(&dcb, sizeof(DCB));
        dcb.DCBlength = sizeof(DCB);
        dcb.BaudRate = 230400;
        dcb.ByteSize = 8;
        dcb.Parity = NOPARITY;
        dcb.StopBits = ONESTOPBIT;
        BOOL bRet = ::SetCommState( handle, &dcb );
        if ( bRet ) {
            iRet = acc_command_start( handle );
        }

        if ( iRet == ACC_OK ) {
            hComm = handle;
        } else {
            CloseHandle( handle );
        }
    }

    return iRet;
}

int acc_sampling( HANDLE &hComm, int &x, int &y, int &z )
{
    int iRet = ACC_NG;
    char buff[20];
    memset( (void *)buff, 0, sizeof( buff ) );
    iRet = acc_receive( hComm, buff, sizeof( buff ) );
    if ( iRet == ACC_OK ) {
        if ( buff[18] == 0x0f && buff[19] == 0x0d ) {
            short *data = (short *)&buff[6];

```

```

        x = (int)(( (*data & 0xFF00) >> 8) | (*data << 8) ) & 0xFFFF;
        data++;
        y = (int)(( (*data & 0xFF00) >> 8) | (*data << 8) ) & 0xFFFF;
        data++;
        z = (int)(( (*data & 0xFF00) >> 8) | (*data << 8) ) & 0xFFFF;

        iRet = ACC_CONTINUE;
    } else {
        iRet = ACC_NG;
    }
}
return iRet;
}

```

```

void acc_close( HANDLE &hComm )
{
    const char *ACC_COMMAND[] = {
        "s¥r",
        "r¥r",
        "¥r",
        NULL,
    };

    ::PurgeComm( hComm, PURGE_TXABORT | PURGE_RXABORT );

    for ( int i = 0; ACC_COMMAND[i] != NULL; i++ ) {
        Sleep(1000);
        acc_send( hComm, ACC_COMMAND[i] );
    }

    CloseHandle( hComm );
}

```

```

static int acc_command_start( HANDLE &hComm )
{
    int iRet = ACC_NG;
    const char *ACC_COMMAND[] = {
        "¥r",
        "¥r",
        "MO05¥r",
        "S¥r",
        NULL,
    };

    for ( int i = 0; ACC_COMMAND[i] != NULL; i++ ) {
        iRet = acc_send( hComm, ACC_COMMAND[i] );
        if ( iRet != ACC_OK ) {
            break;
        }
    }

    return iRet;
}

```

```

static int acc_send( HANDLE &hComm, const char *command )
{
    int iRet = ACC_NG;
    DWORD iSize = strlen( command );
    BOOL bRet = ::WriteFile( hComm,
                                command,
                                iSize,
                                &iSize,

```



```

NULL );
    if ( bRet ) {
        char buff[4];
        memset( (void *)buff, 0, sizeof( buff ) );
        iRet = acc_receive( hComm, buff, sizeof( buff ) - 1 );
        if ( iRet == ACC_OK ) {
            if ( buff[0] != '0' || buff[1] != 'K' || buff[2] != 'r' ) {
                iRet = ACC_NG;
            }
        }
    }

    return iRet;
}

static int acc_receive( HANDLE &hComm, char *buff, int buffSize )
{
    #if 0
    #else
        int iRet = ACC_NG;
        DWORD lSize = 0;
        int readLen = 0;
        while( readLen < buffSize ) {
            BOOL bRet = ::ReadFile( hComm, &buff[readLen], buffSize - readLen, &lSize, NULL );
            if ( !bRet ) {
                break;
            }
            readLen += lSize;
        }
        if ( readLen == buffSize ) {
            iRet = ACC_OK;
        }
    #endif

    return iRet;
}

```

## ファイル名 (CAccIF.cpp)

```

#include "stdafx.h"
#include "CAccIF.h"
#include "AccReadThread.h"
#include "wmid.h"

enum{
    ACC_STATUS_BASIC = 0,
    ACC_STATUS_DISTANCE,
};

#define Acc_GETX(s , p) ((s - p) * 0.000752)
#define Acc_GETY(s , p) ((s - p) * 0.000749)
#define ACC_SAMPLE_NUM (50)
static ACC_READ_THREAD_PARAM g_AccThreadParam;

CAccIF::CAccIF(HWND hWnd)
{
    mhWnd = hWnd;
    mhDialog = NULL;
    mAccSampleThreadHandle = NULL;
    mError = FALSE;
    mFirstAveragePixel = TRUE;
}

```

```

CAccIF::~CAccIF(void)
{
    if ( mShake ) {
        mStopShake();
    }
    mCloseAcc();
}

void CAccIF::mToggleShake(void)
{
    if ( mShake ) {
        mStopShake();
    } else {
        mStartShake();
    }
}

BOOL CAccIF::mOpenAcc(LPTSTR filePath )
{
    BOOL bRet = FALSE;

    mShake = FALSE;
    mAccSampleThreadLoop = TRUE;

    int confResult = cfr_readConfigFile( (const char *)filePath );
    if ( confResult != CFR_OK ) {
        cfr_setDefault();
    }
    mConfig = cfr_getConfig();

    memset( (void *)&g_AccThreadParam, 0, sizeof( g_AccThreadParam ) );
    g_AccThreadParam.bpLoop = &mAccSampleThreadLoop;
    g_AccThreadParam.bpNotice = &mShake;
    g_AccThreadParam.comName = mConfig->caAccDeviceName;
    g_AccThreadParam.noticeProc = &CAccIF::mAccNoticeProc;
    g_AccThreadParam.vpClient = (void *)this;

    mAccSampleThreadHandle = (HANDLE)_beginthreadex(NULL,
        0,
        accReadThreadMain,
        (void *)&g_AccThreadParam,
        0,
        &mAccSampleThreadID);
    if ( mAccSampleThreadHandle ) {
        bRet = TRUE;
    }

    return bRet;
}

void CAccIF::mErrorAcc( int error )
{
    LPCSTR message;
    switch ( error ) {
    case ACC_THREAD_ERROR_OPEN:
        message = "Acc Thread Open Error.¥nCheck Com Name.";
        break;
    case ACC_THREAD_ERROR_SAMPLE:
        message = "Acc Sampling Error.";

```

```

        break;
    default:
        message = "Acc Thread Unknown Error.";
        break;
    }

    if ( !mError ) {
        mError = TRUE;
        ::MessageBox( NULL, (LPCSTR)message, "ACC THREAD ERROR", MB_ICONSTOP );
    }
}

void CAccIF::setDialogWnd( HWND dialog )
{
    mhDialog = dialog;
}

void CAccIF::mCloseAcc(void)
{
    if ( mAccSampleThreadHandle ) {
        mAccSampleThreadLoop = FALSE;
        WaitForSingleObject( mAccSampleThreadHandle, INFINITE );
        CloseHandle( mAccSampleThreadHandle );
        mAccSampleThreadHandle = NULL;
    }
}

void CAccIF::mStartShake(void)
{
    if ( !mShake ) {
        mShake = TRUE;
    }
}

void CAccIF::mStopShake(void)
{
    if ( mShake ) {
        mShake = FALSE;
    }
}

void CAccIF::mAccNoticeProc( int iKind, void *vpServer, void *vpClient )
{
    CAccIF *This = reinterpret_cast<CAccIF *>(vpClient);
    switch ( iKind ) {
        case ACC_THREAD_KIND_NOTICE_ACC:
            {
                ACC_READ_RESULT *result = (ACC_READ_RESULT *)vpServer;
                This->mAccCalc( result->x, result->y, result->z );
            }
            break;
        case ACC_THREAD_KIND_ERROR:
            This->mAccThreadError( (int)vpServer );
            break;
        default:
            break;
    }
}

void CAccIF::mAccCalc( int x, int y, int z )
{
    double SpeedX = 0;
    double SpeedY = 0;
    double DistanceX = 0;
    double DistanceY = 0;
}

```

```

int PixelX = 0;
int PixelY = 0;
int mAccumPixelX = 0;
int mAccumPixelY = 0;

if(mStatus == ACC_STATUS_BASIC) {
    if(mFirst) {
        mFirstValueX=x;
        mFirstValueY=y;
        mAccumX = 0;
        mAccumY = 0;
        mAveragex=x;
        mAveragey=y;
        mSampleCounter=1;
        mFirst=FALSE;
    } else {
        int SaX = mFirstValueX - x;
        int SaY = mFirstValueY - y;
        if(SaX > -mConfig->iAccMove &&
           SaX < mConfig->iAccMove &&
           SaY > -mConfig->iAccMove &&
           SaY < mConfig->iAccMove ) {
            mAveragex += x;
            mAveragey += y;
            mAveragex /= 2;
            mAveragey /= 2;
            mAccumX += x;
            mAccumY += y;
            mSampleCounter++;
            if (mSampleCounter > mConfig->iBasicSampleNum) {
                mAveragex = mAccumX / mConfig->iBasicSampleNum;
                mAveragey = mAccumY / mConfig->iBasicSampleNum;
                mBasicX = mAveragex;
                mBasicY = mAveragey;
                mFirst = TRUE;
                mPrevSpeedX = 0;
                mPrevSpeedY = 0;
                mPrevDistanceX = 0;
                mPrevDistanceY = 0;
                mPrevX = mBasicX;
                mPrevY = mBasicY;
                mPrevPixelX = 0;
                mPrevPixelY = 0;
                mStatus = ACC_STATUS_DISTANCE;
            }
        } else {
            mFirst = TRUE;
        }
    }
} else {
    int Sa2X = abs( x - mBasicX );
    int Sa2Y = abs( y - mBasicY );
    if( Sa2X < mConfig->iAccMove && Sa2Y < mConfig->iAccMove ){
        double accX = Acc_GETX(x , mBasicX);
        double accY = Acc_GETY(y , mBasicY);
        SpeedX = accX / ACC_SAMPLE_NUM + mPrevSpeedX * SLOWDOWNRATIO;
        SpeedY = accY / ACC_SAMPLE_NUM + mPrevSpeedY * SLOWDOWNRATIO;
        DistanceX = SpeedX / ACC_SAMPLE_NUM + mPrevDistanceX * CENTERPULLRATIO;
        DistanceY = SpeedY / ACC_SAMPLE_NUM + mPrevDistanceY * CENTERPULLRATIO;
        PixelX = (int)((double)(DistanceX * (mConfig->iDispSizeX /
mConfig->dMonitorSizeX)));
        PixelY = (int)((double)(DistanceY * (mConfig->iDispSizeY /
mConfig->dMonitorSizeY)));
        if ( mFirstAveragePixel ) {
            mAveragePixelX = PixelX;

```

```

        mAveragePixelY = PixelY;
        mFirstAveragePixel = FALSE;
    }

    mAccumPixelX += PixelX;
    mAccumPixelY += PixelY;
    mPixelSampleCounter++;
    if (mPixelSampleCounter > mConfig->iPixelSampleNum) {
        mAveragePixelX = mAccumPixelX / mConfig->iPixelSampleNum;
        mAveragePixelY = mAccumPixelY / mConfig->iPixelSampleNum;
        mPixelSampleCounter = 0;
        mAccumPixelX = 0;
        mAccumPixelY = 0;
    }

    mAveragePixelX = ( mAveragePixelX + PixelX ) / 10 ;
    mAveragePixelY = ( mAveragePixelY + PixelY ) / 10 ;

    if ( mConfig->bWindowMove ) {
        mMoveWindows( mAveragePixelX - mPrevPixelX, mAveragePixelY -
mPrevPixelY);
    }

    mPrevSpeedX = SpeedX;
    mPrevSpeedY = SpeedY;
    mPrevDistanceX = DistanceX;
    mPrevDistanceY = DistanceY;
    mPrevPixelX = mAveragePixelX;
    mPrevPixelY = mAveragePixelY;

    } else {
        mStatus = ACC_STATUS_BASIC;
    }
}

double *data = (double *)calloc( ACC_MONITOR_MAX, sizeof( double ) );
if ( data != NULL ) {
    data[ACC_MONITOR_ACC_X] = x;
    data[ACC_MONITOR_ACC_Y] = y;
    data[ACC_MONITOR_ACC_Z] = z;
    data[ACC_MONITOR_SPEED_X] = SpeedX;
    data[ACC_MONITOR_SPEED_Y] = SpeedY;
    data[ACC_MONITOR_DISTANCE_X] = DistanceX;
    data[ACC_MONITOR_DISTANCE_Y] = DistanceY;
    data[ACC_MONITOR_PIXEL_X] = PixelX;
    data[ACC_MONITOR_PIXEL_Y] = PixelY;
    PostMessage( mhDialog, WM_APP_MODIFY, NULL, (LPARAM) data );
} else {
    mAccThreadError( ACC_THREAD_ERROR_SAMPLE );
}
mPrevAccX = x;
mPrevAccY = y;
mPrevAccZ = z;
}

void CAccIF::mAccThreadError( int error )
{
    mStopShake();
    PostMessage( mhWnd, WM_APP_THREAD_ERROR, NULL, (LPARAM) error );
}

void CAccIF::mMoveWindows( int x, int y )
{
    HWND lhwnd = GetTopWindow(GetDesktopWindow());
    while ( lhwnd ) {

```

```

        if ( IsWindowVisible(lhwnd) ) {
            RECT rect;
            GetWindowRect( lhwnd, &rect );
            if ( !MoveWindow( lhwnd,
                rect.left + x,
                rect.top + y,
                rect.right - rect.left,
                rect.bottom - rect.top, TRUE ) )
        }
        break;
    }
    lhwnd = GetNextWindow( lhwnd, GW_HWNDNEXT );
}
}

```

## ファイル名 (configRead. cpp)

```

#include "stdafx.h"

enum {
    CFR_MAX_ONE_LINE_SIZE = 256,
    CFR_MAX_CONFIG_VALUE_SIZE = 64,
};

#define CFR_FLAG_ON "ON"
#define CFR_FLAG_OFF "OFF"

#define CFR_DEFAULT_ACC_DEVICE_NAME "com5"
enum {
    CFR_DEFAULT_DISP_SIZE_X = 1024,
    CFR_DEFAULT_DISP_SIZE_Y = 768,
    CFR_DEFAULT_MONITOR_SIZE_X = 20,
    CFR_DEFAULT_MONITOR_SIZE_Y = 16,
    CFR_DEFAULT_ACC_MOVE = 0xF000,
    CFR_DEFAULT_BASIC_SAMPLE_NUM = 50,
    CFR_DEFAULT_PIXEL_SAMPLE_NUM = 1000,
};

typedef struct {
    char *cpConfigName;
    int (*func)(const char *i_cpStr);
} ST_CFR_INDEX_TBL;

static int cfr_parseCongig( FILE *i_fpConfig );
static int cfr_findConfigKey( const char *i_cpStr );
static int cfr_getConfigValue( char *o_cpValue, const char *i_ccpStr, int i_iValueBuffSize );
static int cfr_getAccDeviceName( const char *i_ccpStr );
static int cfr_getWindowMove( const char *i_ccpStr );
static int cfr_getDispSize( const char *i_ccpStr );
static int cfr_getMonitorSize( const char *i_ccpStr );
static int cfr_getAccMove( const char *i_ccpStr );
static int cfr_getBasicSampleNum( const char *i_ccpStr );

static ST_CONFIG g_stConfig;

static const ST_CFR_INDEX_TBL g_cstConfigIndex[] = {
    { "ACC_DEVICE_NAME", cfr_getAccDeviceName,
    },
    { "WINDOW_MOVE", cfr_getWindowMove,
    },
    { "DISP_SIZE", cfr_getDispSize,
    }
}

```

```

},
    { "MONITOR_SIZE",                                     cfr_getMonitorSize,
    },
    { "ACC_MOVE",                                       cfr_getAccMove,
    },
    { "BASIC_SAMPLE_NUM",                               cfr_getBasicSampleNum,
    },
    { NULL,                                             NULL,
    },
};

```

```
int cfr_readConfigFile( const char *i_ccpConfigFileName )
```

```

{
    int iRet = CFR_NG;
    FILE *fpConfig;

    if ( i_ccpConfigFileName != NULL ) {
        memset( (void *)&g_stConfig, 0, sizeof( ST_CONFIG ) );
        fpConfig = fopen( i_ccpConfigFileName, "r" );
        if ( fpConfig != NULL ) {
            iRet = cfr_parseCongig( fpConfig );
            fclose( fpConfig );
        } else {
            perror( "file open" );
        }
    }

    return iRet;
}

```

```
void cfr_setDefault( void )
```

```

{
    memset( (void *)&g_stConfig, 0, sizeof( ST_CONFIG ) );
    strncpy( g_stConfig.caAccDeviceName, CFR_DEFAULT_ACC_DEVICE_NAME, sizeof(
(g_stConfig.caAccDeviceName) ) );
    g_stConfig.bWindowMove = false;
    g_stConfig.iDispSizeX = CFR_DEFAULT_DISP_SIZE_X;
    g_stConfig.iDispSizeY = CFR_DEFAULT_DISP_SIZE_Y;
    g_stConfig.dMonitorSizeX = CFR_DEFAULT_MONITOR_SIZE_X;
    g_stConfig.dMonitorSizeY = CFR_DEFAULT_MONITOR_SIZE_Y;
    g_stConfig.iAccMove = CFR_DEFAULT_ACC_MOVE;
    g_stConfig.iBasicSampleNum = CFR_DEFAULT_BASIC_SAMPLE_NUM;
    g_stConfig.iPixelSampleNum = CFR_DEFAULT_PIXEL_SAMPLE_NUM;
}

```

```
ST_CONFIG *cfr_getConfig( void )
```

```

{
    return &g_stConfig;
}

```

```
static int cfr_parseCongig( FILE *i_fpConfig )
```

```

{
    char caBuff[CFR_MAX_ONE_LINE_SIZE];
    char *cpSearch = NULL;
    char *cpGetResult = NULL;
    int iRet = CFR_NG;

    if ( i_fpConfig != NULL ) {
        while ( CFR_TRUE ) {
            memset( (void *)caBuff, 0, sizeof( caBuff ) );
            cpGetResult = fgets( caBuff, sizeof( caBuff ), i_fpConfig );
            if ( cpGetResult == NULL ) {
                if ( !ferror( i_fpConfig ) ) {
                    iRet = CFR_OK;
                }
            }
        }
    }
}

```

```

    }
    break;
} else {
    cpSearch = strchr( caBuff, '¥n' );
    if ( cpSearch != NULL ) {
        *cpSearch = '¥0';
    }

    cpSearch = strchr( caBuff, '#' );
    if ( cpSearch != NULL ) {
        *cpSearch = '¥0';
    }

    if ( strlen( caBuff ) > 0 ) {
        cfr_findConfigKey( caBuff );
    }
}

}

return iRet;
}

static int cfr_findConfigKey( const char *i_ccpStr )
{
    int iRet = CFR_NG;
    int i;
    int iConfigNameLength;

    if ( i_ccpStr != NULL ) {
        iRet = CFR_OK;
        for( i = 0; g_cstConfigIndex[i].cpConfigName != NULL; i++ ) {
            iConfigNameLength = strlen( g_cstConfigIndex[i].cpConfigName );
            if( strncmp( i_ccpStr, g_cstConfigIndex[i].cpConfigName, iConfigNameLength )
== 0 ) {
                iRet = g_cstConfigIndex[i].func( &i_ccpStr[iConfigNameLength] );
                break;
            }
        }
    }

    return iRet;
}

static int cfr_getConfigValue( char *o_cpValue, const char *i_ccpStr, int i_iValueBuffSize )
{
    int iRet = CFR_NG;
    int i, j;

    if ( (o_cpValue != NULL) && (i_ccpStr != NULL) ) {
        for ( i = 0; i_ccpStr[i] != '¥0'; i++ ) {
            if ( i_ccpStr[i] == '=' ) {
                i++;
                break;
            }
        }

        memset( (void *)o_cpValue, 0, i_iValueBuffSize );
        for ( j = 0; i_ccpStr[i] != '¥0'; i++ ) {
            if ( (i_ccpStr[i] != ' ') && (i_ccpStr[i] != '¥t') ) {
                o_cpValue[j] = i_ccpStr[i];
                j++;
            }
        }
        iRet = CFR_OK;
    }
}

```



```

    }

    return iRet;
}

static int cfr_getAccDeviceName( const char *i_ccpStr )
{
    int iRet = CFR_NG;
    char caValue[CFR_MAX_CONFIG_VALUE_SIZE];

    if ( i_ccpStr != NULL ) {
        iRet = cfr_getConfigValue( caValue, i_ccpStr, sizeof( caValue ) );
        if ( iRet == CFR_OK ) {
            strncpy( g_stConfig.caAccDeviceName, caValue,
sizeof( g_stConfig.caAccDeviceName ) );
        }
    }

    return iRet;
}

static int cfr_getWindowMove( const char *i_ccpStr )
{
    int iRet = CFR_NG;
    int iMove;
    char caValue[CFR_MAX_CONFIG_VALUE_SIZE];

    if ( i_ccpStr != NULL ) {
        iRet = cfr_getConfigValue( caValue, i_ccpStr, sizeof( caValue ) );
        if ( iRet == CFR_OK ) {
            iMove = atoi( caValue );
            if ( iMove ) {
                g_stConfig.bWindowMove = true;
            } else {
                g_stConfig.bWindowMove = false;
            }
        }
    }

    return iRet;
}

static int cfr_getDispSize( const char *i_ccpStr )
{
    int iRet = CFR_NG;
    char caValue[CFR_MAX_CONFIG_VALUE_SIZE];

    if ( i_ccpStr != NULL ) {
        iRet = cfr_getConfigValue( caValue, i_ccpStr, sizeof( caValue ) );
        if ( iRet == CFR_OK ) {
            sscanf( caValue, "%d,%d", &g_stConfig.iDispSizeX,
&g_stConfig.iDispSizeY );
        }
    }

    return iRet;
}

static int cfr_getMonitorSize( const char *i_ccpStr )
{
    int iRet = CFR_NG;
    char caValue[CFR_MAX_CONFIG_VALUE_SIZE];

```

```

        if ( i_ccpStr != NULL ) {
            iRet = cfr_getConfigValue( caValue, i_ccpStr, sizeof( caValue ) );
            if ( iRet == CFR_OK ) {
                sscanf( caValue, "%lf,%lf", &g_stConfig.dMonitorSizeX,
&g_stConfig.dMonitorSizeY );
            }
        }

        return iRet;
    }

static int cfr_getAccMove( const char *i_ccpStr )
{
    int iRet = CFR_NG;
    char caValue[CFR_MAX_CONFIG_VALUE_SIZE];

    if ( i_ccpStr != NULL ) {
        iRet = cfr_getConfigValue( caValue, i_ccpStr, sizeof( caValue ) );
        if ( iRet == CFR_OK ) {
            g_stConfig.iAccMove = atoi( caValue );
        }
    }

    return iRet;
}

static int cfr_getBasicSampleNum( const char *i_ccpStr )
{
    int iRet = CFR_NG;
    char caValue[CFR_MAX_CONFIG_VALUE_SIZE];

    if ( i_ccpStr != NULL ) {
        iRet = cfr_getConfigValue( caValue, i_ccpStr, sizeof( caValue ) );
        if ( iRet == CFR_OK ) {
            g_stConfig.iBasicSampleNum = atoi( caValue );
        }
    }

    return iRet;
}
}

```

## ファイル名 (Shake.cpp)

```

#include "stdafx.h"
#include "Shake.h"
#include "CAccIF.h"
#include "wmid.h"

#define MAX_LOADSTRING 100
#define SHAKE_MUTEX_NAME _T("T_T_APPLICATION_7368616B-74616E616B61-746F7368697961")

HINSTANCE hInst;
TCHAR szTitle[MAX_LOADSTRING];
TCHAR szWindowClass[MAX_LOADSTRING];
CAccIF *cpAccIF;
NOTIFYICONDATA taskTray;
HANDLE hMutex;
LPTSTR g_filePath;

ATOM MyRegisterClass(HINSTANCE hInstance);
BOOL InitInstance(HINSTANCE, int);
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);

```

```

INT_PTR CALLBACK    About(HWND, UINT, WPARAM, LPARAM);
INT_PTR CALLBACK    ShakeMonitor(HWND, UINT, WPARAM, LPARAM);
BOOL SetTrayIcon(HWND hWnd);
BOOL DelTrayIcon(void);
void OnTaskTray( HWND hWnd, WPARAM wparam, LPARAM lparam );

int APIENTRY _tWinMain(HINSTANCE hInstance,
                      HINSTANCE hPrevInstance,
                      LPTSTR lpCmdLine,
                      int nCmdShow)
{
    UNREFERENCED_PARAMETER(hPrevInstance);
    UNREFERENCED_PARAMETER(lpCmdLine);

    hMutex = CreateMutex(          NULL,
                                FALSE,
                                SHAKE_MUTEX_NAME );

    if(!(hMutex && GetLastError() != ERROR_ALREADY_EXISTS))
    {
        return FALSE;
    }
    g_filePath = lpCmdLine;

    MSG msg;
    HACCEL hAccelTable;

    LoadString(hInstance, IDS_APP_TITLE, szTitle, MAX_LOADSTRING);
    LoadString(hInstance, IDC_SHAKE, szWindowClass, MAX_LOADSTRING);
    MyRegisterClass(hInstance);

    if (!InitInstance (hInstance, nCmdShow))
    {
        ReleaseMutex( hMutex );
        CloseHandle( hMutex );
        return FALSE;
    }

    hAccelTable = LoadAccelerators(hInstance, MAKEINTRESOURCE(IDC_SHAKE));

    while (GetMessage(&msg, NULL, 0, 0))
    {
        if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg))
        {
            TranslateMessage(&msg);
            DispatchMessage(&msg);
        }
    }

    return (int) msg.wParam;
}

ATOM MyRegisterClass(HINSTANCE hInstance)
{
    WNDCLASSEX wcex;

    wcex.cbSize = sizeof(WNDCLASSEX);

    wcex.style          = CS_HREDRAW | CS_VREDRAW;
    wcex.lpfnWndProc    = WndProc;
    wcex.cbClsExtra     = 0;
    wcex.cbWndExtra     = 0;
    wcex.hInstance     = hInstance;
    wcex.hIcon          = LoadIcon(hInstance, MAKEINTRESOURCE(IDI_SHAKE));
    wcex.hCursor        = LoadCursor(NULL, IDC_ARROW);

```

```

    wcx.hbrBackground = (HBRUSH) (COLOR_WINDOW+1);
    wcx.lpszMenuName = MAKEINTRESOURCE (IDC_SHAKE);
    wcx.lpszClassName = szWindowClass;
    wcx.hIconSm = LoadIcon(wcx.hInstance, MAKEINTRESOURCE (IDI_SMALL));

    return RegisterClassEx(&wcx);
}

BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)
{
    hInst = hInstance;

    HWND hWnd = CreateWindowEx(    WS_EX_TOOLWINDOW,

                                   szTitle,
                                   NULL,
                                   WS_POPUP | WS_VISIBLE,
                                   0,
                                   0,
                                   1,
                                   1,
                                   NULL,
                                   (HMENU) NULL,
                                   hInst,
                                   NULL);

    if (!hWnd)
    {
        return FALSE;
    }

    ShowWindow(hWnd, SW_HIDE);

    cpAccIF = new CAccIF( hWnd );
    if ( !cpAccIF->mOpenAcc(g_filePath) ) {
        delete cpAccIF;
        return FALSE;
    }

    SetTrayIcon(hWnd);

    return TRUE;
}

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    int wmId, wmEvent;
    PAINTSTRUCT ps;
    HDC hdc;

    switch (message)
    {
    case WM_COMMAND:
        wmId = LOWORD(wParam);
        wmEvent = HIWORD(wParam);
        switch (wmId)
        {
        case IDM_ABOUT:
            DialogBox(hInst, MAKEINTRESOURCE (IDD_ABOUTBOX), hWnd, About);
            break;
        case IDM_EXIT:
            DestroyWindow(hWnd);
            break;
        default:
            return DefWindowProc(hWnd, message, wParam, lParam);
        }
    }
}

```

```

        }
        break;
    case WM_PAINT:
        hdc = BeginPaint(hWnd, &ps);
        EndPaint(hWnd, &ps);
        break;
    case WM_DESTROY:
        DelTrayIcon();
        delete cpAccIF;
        ReleaseMutex( hMutex );
        CloseHandle( hMutex );
        PostQuitMessage(0);
        break;
    case WM_APP_TASKTRAY:
        OnTaskTray( hWnd, wParam, lParam );
        break;
    case WM_APP_THREAD_ERROR:
        cpAccIF->mErrorAcc( (int)lParam );
        DestroyWindow(hWnd);
        break;
    default:
        return DefWindowProc(hWnd, message, wParam, lParam);
    }
    return 0;
}

INT_PTR CALLBACK About(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)
{
    UNREFERENCED_PARAMETER(lParam);
    switch (message)
    {
    case WM_INITDIALOG:
        return (INT_PTR)TRUE;

    case WM_COMMAND:
        if (LOWORD(wParam) == IDOK || LOWORD(wParam) == IDCANCEL)
        {
            EndDialog(hDlg, LOWORD(wParam));
            return (INT_PTR)TRUE;
        }
        break;
    }
    return (INT_PTR)FALSE;
}

BOOL SetTrayIcon(HWND hWnd)
{
    taskTray.cbSize          = sizeof(NOTIFYICONDATA);
    taskTray.uID             = 1;
    taskTray.hWnd            = hWnd;
    taskTray.uFlags          = NIF_MESSAGE | NIF_ICON | NIF_TIP;
    taskTray.uVersion        = NOTIFYICON_VERSION;
    taskTray.hIcon           = LoadIcon(hInst, MAKEINTRESOURCE(IDI_SHAKE));
    taskTray.uCallbackMessage = WM_APP_TASKTRAY;
    _tcsncpy_s(taskTray.szTip, _tcslen("Shake") + 1, "Shake");

    return Shell_NotifyIcon(NIM_ADD, &taskTray);
}

BOOL DelTrayIcon(void)
{
    return Shell_NotifyIcon(NIM_DELETE, &taskTray);
}

void OnTaskTray( HWND hWnd, WPARAM wParam, LPARAM lParam )

```

```

{
    switch (lparam) {
    case WM_RBUTTONDOWN:
        {
            POINT point;
            GetCursorPos( &point );
            HMENU hMenu = CreatePopupMenu();
            if(hMenu) {
                AppendMenu(hMenu, MFT_STRING, IDM_ABOUT, _T("About"));
                AppendMenu(hMenu, MFT_SEPARATOR, NULL, NULL);
                AppendMenu(hMenu, MFT_STRING, IDM_EXIT, _T("Quit"));
                SetForegroundWindow(hWnd);
                TrackPopupMenu(hMenu,
                    TPM_RIGHTALIGN | TPM_RIGHTBUTTON,
                    TPM_BOTTOMALIGN |
                    point.x,
                    point.y,
                    0,
                    hWnd,
                    NULL);
                DestroyMenu(hMenu);
            }
        }
        break;
    case WM_LBUTTONDOWN:
        DialogBox( hInst, MAKEINTRESOURCE(IDD_DIALOG_MONITOR), hWnd, ShakeMonitor );
        break;
    }
}

```

```

INT_PTR CALLBACK ShakeMonitor(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)

```

```

{
    static const int MONITOR_EDIT_IDS[ACC_MONITOR_MAX] = {
        IDC_EDIT_ACC_X,
        IDC_EDIT_ACC_Y,
        IDC_EDIT_ACC_Z,
        IDC_EDIT_SPEED_X,
        IDC_EDIT_SPEED_Y,
        IDC_EDIT_DIS_X,
        IDC_EDIT_DIS_Y,
        IDC_EDIT_PIX_X,
        IDC_EDIT_PIX_Y,
    };
    switch (message) {
    case WM_INITDIALOG:
        cpAccIF->setDialogWnd( hDlg );
        return (INT_PTR)TRUE;
    case WM_COMMAND:
        if ( LOWORD(wParam) == IDC_BUTTON_START ) {
            EnableWindow( GetDlgItem( hDlg, IDC_BUTTON_START ), FALSE );
            EnableWindow( GetDlgItem( hDlg, IDC_BUTTON_STOP ), TRUE );
            cpAccIF->mToggleShake();
        } else if ( LOWORD(wParam) == IDC_BUTTON_STOP ) {
            EnableWindow( GetDlgItem( hDlg, IDC_BUTTON_START ), TRUE );
            EnableWindow( GetDlgItem( hDlg, IDC_BUTTON_STOP ), FALSE );
            cpAccIF->mToggleShake();
        } else if ( LOWORD(wParam) == IDC_BUTTON_QUIT ) {
            DestroyWindow( GetParent( hDlg ) );
        } else {
            return (INT_PTR)FALSE;
        }
        return (INT_PTR)TRUE;
    case WM_APP_MODIFY:
        {
            double *value = (double *)lParam;

```

```

        for ( int i = 0; i < ACC_MONITOR_MAX; i++ ) {
            char edit[64];
            memset( edit, 0, sizeof(edit) );
            if ( i >= ACC_MONITOR_ACC_X && i <= ACC_MONITOR_ACC_Z ) {
                sprintf( edit, "0x%X", (int)value[i] );
            } else {
                sprintf( edit, "%lf", value[i] );
            }
            SetWindowText( GetDlgItem( hDlg, MONITOR_EDIT_IDS[i] ), edit );
        }
        free( value );
        return (INT_PTR)TRUE;
    }
case WM_KEYDOWN:
    DestroyWindow( GetParent( hDlg ) );
    return (INT_PTR)TRUE;
}

return (INT_PTR)FALSE;
}
}

```

ファイル名 (stdafx.cpp)

```
#include "stdafx.h"
```

- ヘッダーファイル

ファイル名 (AccReadThread.h)

```
#ifndef __ACC_READ_THREAD_H
```

```
#define __ACC_READ_THREAD_H
```

```
enum {
    ACC_THREAD_KIND_NOTICE_ACC = 0,
    ACC_THREAD_KIND_ERROR,
};
```

```
enum {
    ACC_THREAD_ERROR_OPEN = 0,
    ACC_THREAD_ERROR_SAMPLE,
};
```

```
typedef struct {
    int x;
    int y;
    int z;
} ACC_READ_RESULT;
```

```
typedef struct {
    BOOL *bpNotice;
    BOOL *bpLoop;
    void (*noticeProc)( int iKind, void *vpServer, void *vpClient );
    _TCHAR *comName;
    void *vpClient;
} ACC_READ_THREAD_PARAM;
```

```
extern unsigned __stdcall accReadThreadMain( void *param );
```

```
#endif
```

## ファイル名 (AccSample.h)

```
#ifndef __ACC_SAMPLE_H
#define __ACC_SAMPLE_H

enum {
    ACC_ARGV_APP_NAME = 0,
    ACC_ARGV_COM_NAME,
    ACC_ARGV_MAX,
};

enum {
    ACC_OK = 0,
    ACC_NG,
    ACC_CONTINUE,
};

extern int acc_open( HANDLE &hComm, const _TCHAR *comName );
extern int acc_sampling( HANDLE &hComm, int &x, int &y, int &z );
extern void acc_close( HANDLE &hComm );

#endif
```

## ファイル名 (CAccIF.h)

```
#pragma once

typedef struct {
    int x;
    int y;
} TACC_MOVE_POS;

enum {
    ACC_MONITOR_ACC_X = 0,
    ACC_MONITOR_ACC_Y,
    ACC_MONITOR_ACC_Z,
    ACC_MONITOR_SPEED_X,
    ACC_MONITOR_SPEED_Y,
    ACC_MONITOR_DISTANCE_X,
    ACC_MONITOR_DISTANCE_Y,
    ACC_MONITOR_PIXEL_X,
    ACC_MONITOR_PIXEL_Y,
    ACC_MONITOR_MAX,
};

class CAccIF
{
public:
    CAccIF( HWND hWnd );
    virtual ~CAccIF( void );
    void mToggleShake( void );
    BOOL mOpenAcc( LPTSTR filePath );
    void mErrorAcc( int error );
    void setDialogWnd( HWND dialog );

private:
    HWND mhWnd;
    HWND mhDialog;
    ST_CONFIG *mConfig;
    void mCloseAcc( void );
    void mStartShake( void );
    void mStopShake( void );
};
```





```

enum {
    CFR_FILE_PATH_MAX = 256,
    CFR_DEVICE_NAME_MAX = 256,
};

typedef struct {
    char caAccDeviceName[CFR_DEVICE_NAME_MAX];
    bool bWindowMove;
    int iDispSizeX;
    int iDispSizeY;
    int iAccMove;
    int iBasicSampleNum;
    int iPixelSampleNum;
    double dMonitorSizeX;
    double dMonitorSizeY;
} ST_CONFIG;

extern int cfr_readConfigFile( const char *i_ccpConfigFileName );
extern void cfr_setDefault( void );
extern ST_CONFIG *cfr_getConfig( void );

#endif

```

## ファイル名 (Resource.h)

```

#ifndef IDC_STATIC
#define IDC_STATIC (-1)
#endif

#define IDD_ABOUTBOX 103
#define IDM_ABOUT 104
#define IDM_EXIT 105
#define IDI_SHAKE 107
#define IDI_SMALL 108
#define IDC_SHAKE 109
#define IDD_DIALOG_MONITOR 111
#define IDC_BUTTON_START 1003
#define IDC_BUTTON_STOP 40000
#define IDS_APP_TITLE 40000
#define IDC_BUTTON_QUIT 40001
#define IDC_EDIT_ACC_X 40003
#define IDC_STATIC_ACC_X 40005
#define IDC_STATIC_ACC_Y 40006
#define IDC_STATIC_ACC_Z 40008
#define IDC_STATIC_ACC 40010
#define IDC_STATIC_SPEED 40016
#define IDC_STATIC_SPEED_X 40017
#define IDC_STATIC_SPEED_Y 40018
#define IDC_EDIT_SPEED_X 40019
#define IDC_EDIT_SPEED_Y 40020
#define IDC_EDIT_ACC_Y 40021
#define IDC_EDIT_ACC_Z 40022
#define IDC_STATIC_DIS 40028
#define IDC_STATIC_DIS_X 40029
#define IDC_STATIC_DIS_Y 40030
#define IDC_EDIT_DIS_X 40031
#define IDC_EDIT_DIS_Y 40032
#define IDC_STATIC_PIX 40038
#define IDC_STATIC_PIX_X 40039
#define IDC_STATIC_PIX_Y 40040
#define IDC_EDIT_PIX_X 40041
#define IDC_EDIT_PIX_Y 40042

```

## ファイル名 (Shake.h)

```
#pragma once
```

```
#include "resource.h"
```

## ファイル名 (stdafx.h)

```
#pragma once
```

```
#include "targetver.h"
```

```
#define WIN32_LEAN_AND_MEAN
```

```
#include <windows.h>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <malloc.h>
```

```
#include <memory.h>
```

```
#include <tchar.h>
```

```
#include <time.h>
```

```
#include <shellapi.h>
```

```
#include <process.h>
```

```
#include "configRead.h"
```

## ファイル名 (targetver.h)

```
#pragma once
```

```
#ifndef WINVER
```

```
#define WINVER 0x0600
```

```
#endif
```

```
#ifndef _WIN32_WINNT
```

```
#define _WIN32_WINNT 0x0600
```

```
#endif
```

```
#ifndef _WIN32_WINDOWS
```

```
#define _WIN32_WINDOWS 0x0410
```

```
#endif
```

```
#ifndef _WIN32_IE
```

```
#define _WIN32_IE 0x0700
```

```
#endif
```

## ファイル名 (wmid.h)

```
#ifndef __WMID_H
```

```
#define __WMID_H
```

```
enum {
```

```
    WM_APP_THREAD_ERROR = WM_APP + 1,
```

```
    WM_APP_TASKTRAY,
```

```
    WM_APP_MODIFY,
```

```
};
```

```
#endif
```