

# チルトコントロールを用いたステアリングタスクにおける ユーザパフォーマンスの評価

下野 弘朗<sup>1</sup> 山中 祥太<sup>1,2</sup> 宮下 芳明<sup>1</sup>

**概要:** 本研究は、デバイスの傾きによってポインタの位置を入力する手法について、そのユーザパフォーマンスを評価したものである。実験では参加者に両手でタブレット型端末を持たせ、直線型・円形の経路に沿ってポインタを通過させた。複数の経路幅・経路長の組み合わせについてそれぞれ実施した結果、そのユーザパフォーマンスがステアリングの法則に適合することが分かった。また、横方向に通過する直線型経路を縦方向にずらした状態でポインタを通過させる実験を行った。その結果、経路幅・経路長が同じであっても経路の位置によって通過時間が異なり、経路が回転軸離れているほど通過時間が長くなることが分かった。ただし、回転軸から経路までの距離と通過時間の関係は必ずしも対称になるわけではないことが観察された。

## Evaluation of the user performance in steering tasks using tilt controlling

HIROAKI SHIMONO<sup>1</sup> SHOTA YAMANAKA<sup>1,2</sup> HOMEI MIYASHITA<sup>1</sup>

**Abstract:** In this study, we evaluate the user performance of tilting a tablet device. In the experiment, participants held a tablet and let the pointer pass through a straight and circular paths. For each combination of multiple route widths and route lengths, we found that the user performance conforms to the steering law. In addition, experiments were conducted to pass the pointer in a state in which the straight type path passing in the lateral direction is shifted in the longitudinal direction. As a result, it was found that the movement time is depending on the position of the path even if the path width / path length is the same, and the movement time becomes longer as the path is away from the rotation axis. However, it was observed that the relationship between the distance from the rotation axis to the path and the movement time is not necessarily symmetrical.

### 1. はじめに

デバイスを傾けることで、その傾き具合に応じたインタラクションを行うシステムがスマートフォンやタブレット端末、ゲーム機器などによく搭載されている。例としてデバイスの向きに応じて画面全体の向きが変化する機能や、表示されるコンテンツが増減する機能などがある。

このようにデバイスの傾きの大きな変化をトリガーとして動作する機能がある一方で、傾きの変化量をオブジェク

トの傾きや位置として入力する手法もある。例えば、テキストの入力に利用することを検討した例 [5]、Web 上のインタラクションに利用する例 [6]、近年ではスマートウォッチへの適用を試みた例 [7][8] などがある。このような入力手法はチルトコントロールと呼ばれている。

チルトコントロールによってオブジェクトの位置を操作する際に、指定された道を壁にぶつからないように移動させるようなシチュエーションがある。例えば、ゲーム内のキャラクターを移動させる場合やドローンの操作をチルトコントロールによって行う場合などでは、コースからはみ出ずにできるだけ速くゴールにたどり着くというようなルールが設けられることがある。こうした細長い経路に

<sup>1</sup> 明治大学  
Meiji University, Nakano, Tokyo, Japan  
<sup>2</sup> 日本学術振興会  
JSPS, Chiyoda, Tokyo, Japan

沿って対象を移動させる操作（以降ステアリングタスク）に対しては、チルトコントロールを用いた場合のユーザパフォーマンスと経路から推定されるタスクの難易度の関係について評価がされていない。

本研究では、チルトコントロールを用いてポインタの位置を入力し、そのステアリングタスクの難易度とユーザパフォーマンスの関係について評価する。実験では直線型・円形の2種類の経路を用いたステアリングタスクを実施し、様々な方向の運動においてステアリングの法則 [4] が成立するか検証する。さらに、チルトコントロールを利用したステアリングタスクを評価するためには、経路の幅と経路の長さだけでなく、経路の位置についても考慮する必要がある。なぜなら、経路を通過するためには通過方向に対して垂直の方向にポインタを固定する必要があるが、そのポインタの位置によってデバイスを固定する角度も変化するからである。デバイスを固定する角度が異なれば、腕にかかる負担も異なる。したがって、経路の位置が画面の中心から離れるほどユーザへの負担が高くなるのではないかと考えられる。そこで、このような影響を調べるために経路を移動させた状態でステアリング実験を行い、経路の移動距離がステアリングタスクのユーザパフォーマンスに及ぼす影響についても調査する。

## 2. 関連研究

MacKenzie[1]らは、チルトコントロールによるポインティング手法においてポインティングタスクのパフォーマンス評価を行っている。実験では、円形のターゲットをポインティングするタスクによってフィッツの法則 [3] が成立することを検証している。Teather[2]らは、チルトコントロールによってポインティングを行う際に、デバイスの傾きをポインタの移動速度に対応させた場合とポインタの位置に対応させた場合のユーザパフォーマンスを比較している。ポインタの位置に対応させた場合のほうが速くポインティング可能であることが示されている。

Fitton[5]らは、チルトコントロールによってテキスト入力を行う手法について、実験参加者が座っている状態と歩行している状態での操作性についての比較や、デバイスを片手で持っている状態と両手で持っている状態での操作性の比較を行っている。歩行している状態では座っている状態よりも入力にかかる時間が長く、エラー率が大きくなることが示されている。またデバイスを両手で持っている状態では片手で持っている状態よりもエラー率が低くなることを示されている。Geronimo[6]らは、チルトコントロールをweb上のインタラクションに取り入れるためのフレームワークと、それを用いたアプリケーションについて検討している。Guo[7]らは、スマートウォッチの操作をチルトコントロールで行うための2種類のインタフェースについて比較している。画面にポインタを表示して位置を入

力するよりも、対象オブジェクトを傾きに直接対応させるほうが選択操作を速く正確に行えることが示されている。Shima[8]らは、スマートウォッチのポインティング操作をチルトコントロールで行う際の操作精度について評価している。x軸方向の精度がy軸方向の精度よりも高いことが示されている。

## 3. ステアリングの法則

ステアリングの法則 [4] はフィッツの法則 [3] から派生した、細長い経路を経路幅からはみ出ないように通過させるタスクに対するパフォーマンスモデルである。この法則はステアリングタスクにおける通過速度と正確さの関係を表しており、通過時間 MT は式 (1) から求まる。

$$MT = a + b \times ID \quad (1)$$

a, b は実験により得られる定数である。ID はステアリングタスクの難易度を示すもので、単位は bits である。直線型経路のステアリングタスクにおける ID は式 (2)、円形経路のステアリングタスクにおける ID は式 (3) で表すことができる。

$$ID = \frac{A}{W} \quad (2)$$

$$ID = \frac{2\pi R}{W} \quad (3)$$

A は経路の長さ、W は経路の幅、R は円形経路の半径を表している。このような異なる条件の経路であってもモデルの一致度が高いだけでなく、様々なデバイスや入力手法を用いた場合においてもこの法則が成立することが調査されている。

## 4. ステアリング実験

### 4.1 装置

実験はタブレット端末である ASUS 社製 Nexus7 (2012 年モデル) を用いて行った。使用したデバイスを図 1 に示す。ディスプレイの解像度は 800 px × 1280 px であった。実験システムは Android SDK を用いて Java で開発した。チルトコントロールの実現にはデバイスに搭載されている加速度センサーを利用した。センサーのサンプリングレートは設定可能な内の最速（約 50 Hz）とした。したがって実験システムは、約 20 ms ごとにデバイスの加速度センサーの値を取得して処理を行う。

### 4.2 チルトコントロールシステムの実装

デバイスに搭載されている加速度センサーから取得したデータを座標系に変換するアルゴリズムについて述べる。Teather[2]らの研究で位置操作として用いられたものを採用し、再実装した。この手法では、デバイスの傾きを画面上の絶対座標に変換した値がポインタの位置となる。デ

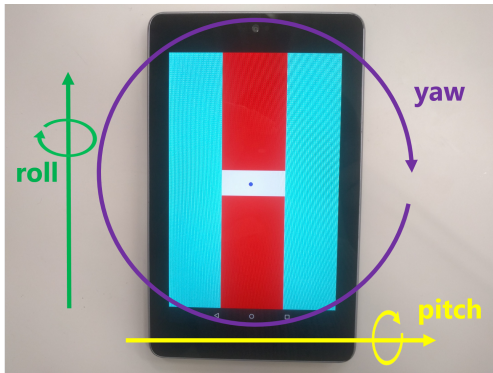


図 1 実験に用いたデバイス

デバイスを傾ければ傾けるほどポインタは初期位置から遠くへ移動し、デバイスを元に戻すとポインタは初期位置に戻る。また、デバイスを固定している状態ではポインタは移動しない。座標の算出方法については、まず得られた pitch と roll から動径と偏角を式 5 のように算出する。

$$\text{tiltMag} = \sqrt{\text{pitch}^2 + \text{roll}^2} \quad (4)$$

$$\text{tiltAngle} = \text{asin}\left(\frac{\text{roll}}{\text{tiltMag}}\right) \quad (5)$$

これらの値から、ディスプレイ上のポインタの位置を xy 座標系で式 (6)~(8) のように算出する。

$$\text{dBall} = \text{tiltMag} \times \text{tiltGain} \quad (6)$$

$$\text{offset} = (\text{dBall} \times \sin(\text{tiltAngle}), \text{dBall} \times \cos(\text{tiltAngle})) \quad (7)$$

$$\text{ballPos} = \text{center} + \text{offset} \quad (8)$$

dBall はポインタの位置の変化量を表す。これを x, y 方向に分解して offset ベクトルとし、それぞれ中心座標に加算することでポインタの位置が求まる。

### 4.3 直線型経路を通過する実験

#### 4.3.1 実験システム

直線型経路を通過するタスクのための実験システムを実装した。その概観を図 2 に示す。画面内のボールの位置をチルトコントロールによって入力することができる。図内の白い領域が経路となっており、ボールの前側が経路の開始地点を通過すると経路に侵入したと判定される。また、ボールが経路に侵入した状態でボールの後ろ側が経路の終了地点を通過すると、経路の通過が完了したと判定され、開始地点と終了地点の色が変化する。ボールが経路に侵入している間にボールが一部でも経路の壁に触れた場合は通過失敗と判定され、ボールの前側が開始地点より前に戻るまで通過の判定を行わないようにした。内藤ら [9] の検証をもとに、ボールに大きさを設定する場合は画面上の経路の幅と経路の長さをボールの大きさ分増減させるようにした。

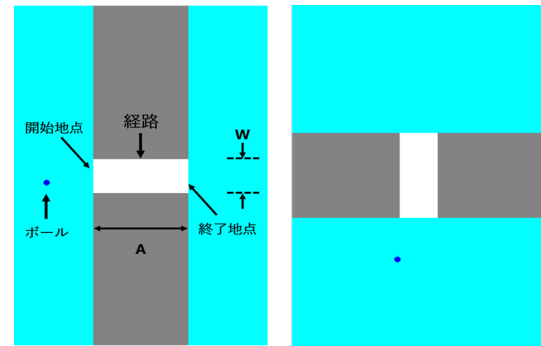


図 2 横方向の直線型経路のステアリング実験システム (左), 縦方向の直線型経路のステアリング実験システム (右)

#### 4.3.2 実験参加者

前述の実験用システムを用いて、直線型経路の通過タスクを行う実験を 8 人のボランティアに実施した。6 人が男性、2 人が女性、平均年齢は 20.25 歳であり、標準偏差は 0.83 歳であった。

#### 4.3.3 実験デザイン

参加者にデバイスを両手で持たせ、直線型経路をはみ出さないようにボールを通過させる実験を行った。経路の長さ A と経路の幅 W の組み合わせ 8 パターンについてそれぞれランダムな順番で行うことを 1 セットとし、繰り返し回数は 4 回とした。このタスクを 4 方向 (上下左右) についてそれぞれの向きの直線型経路に対して行った。ボールの直径を 20 px とし、W と A は以下のように設定した。W と A の値の単位は px である。

$$W : 40, 60, 80, 100$$

$$A : 320, 620$$

実際に表示される経路の幅は 20 px だけさらに大きく、経路の長さは 20 px だけ小さくしている。参加者には経路をできるだけ速く正確に通過するように求め、経路に侵入してから通過するまでの時間を記録した。経路の通過に失敗した場合は開始位置に戻り、そこから再度試行するように指示した。ボールを右から左、左から右へ通過させる場合は、デバイスを pitch の方向に固定しながら roll の方向に傾ける必要があり、ボールを上から下、下から上へ通過させる場合は、デバイスを roll の方向に固定しながら pitch の方向に傾ける必要がある。

### 4.4 円形経路を通過する実験

#### 4.4.1 実験システム

円形経路を通過するタスクのための実験システムを実装した。その概観を図 3 に示す。経路の開始地点と終了地点が同じ位置であり、開始地点よりも左側からであればボールを経路へ侵入させることが可能である。経路の通過完了及び通過失敗の判定は直線型経路を通過する実験で使ったシステムと同様である。

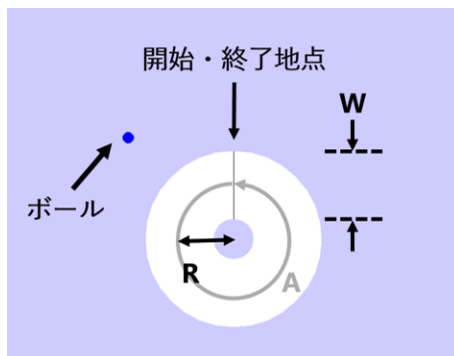


図 3 円形経路のステアリング実験システム

#### 4.4.2 実験参加者

前述の実験用システムを用いて、円形経路の通過タスクを行う実験を6人のボランティアに実施した。5人が男性、1人が女性、平均年齢は20.67歳であり、標準偏差は0.94歳であった。

#### 4.4.3 実験デザイン

参加者にデバイスを両手で持たせ、円形経路をはみ出さないようにボールを通過させる実験を行った。このタスクを2方向（時計回り、反時計回り）についてそれぞれの向きの直線型経路に対して行った。経路の長さAと経路の幅Wの組み合わせ8パターンについてそれぞれランダムな順番で行うことを1セットとし、繰り返し回数は4回とした。ボールの直径を20pxとし、WとAは以下のように設定した。WとAの値の単位はpxである。

W : 40, 60, 80, 100

A : 620, 1220

実際に表示される経路の幅は20pxだけさらに大きく、経路の長さは20pxだけ小さくしている。

経路の半径Rは式9から求めた。

$$R = \frac{A}{2\pi} \quad (9)$$

直線型経路のステアリング実験と同様に、参加者には経路をできるだけ速く正確に通過するように求め、経路に侵入してから通過するまでの時間を記録した。経路の通過に失敗した場合は開始位置から再度試行させた。

#### 4.5 結果

各実験の結果について述べる。直線型経路の通過時間について、IDと通過時間のグラフを図4に、また方向別のエラー率を図5示す。直線型経路の通過時間において、上下左右のすべての方向について  $R^2 > 0.96-0.98$  でステアリングの法則に適合している。円形経路の実験結果について、IDと通過時間のグラフを図6に、また方向別のエラー率を図7示す。円形経路において、時計回り・反時計回りのすべての方向について  $R^2 > 0.98-0.99$  でステアリングの法則に適合している。

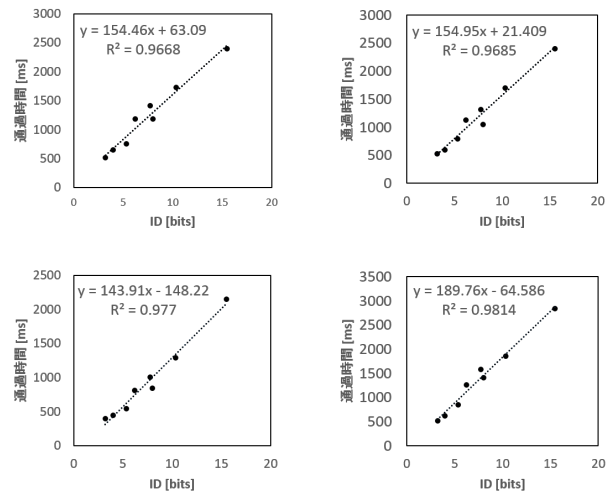


図 4 左右方向の通過時間 (左上), 右左方向の通過時間 (左下), 上下方向の通過時間 (右上), 下上方向の通過時間 (右下)

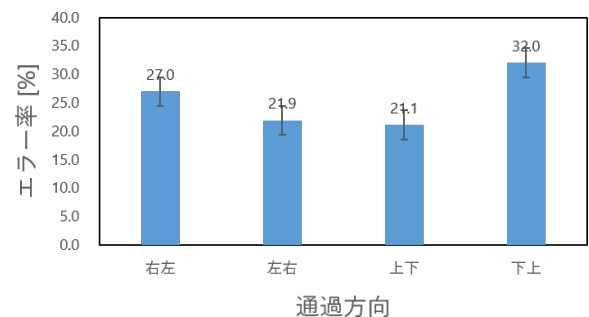


図 5 直線型経路を用いた実験における方向ごとのエラー率

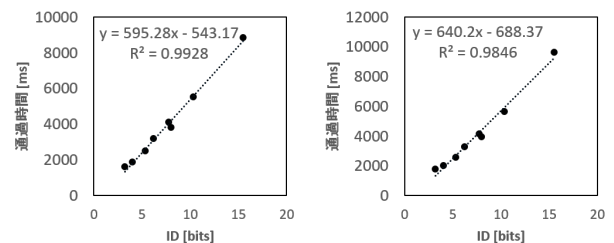


図 6 時計回りの通過時間 (左), 反時計回りの通過時間 (右)

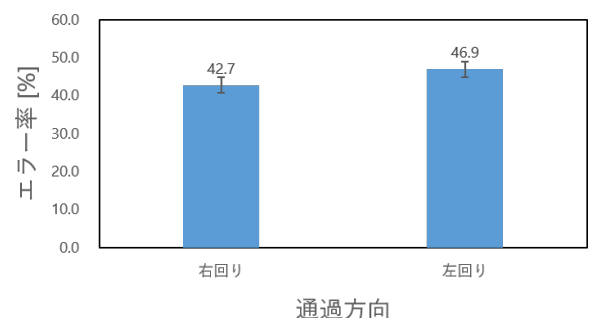


図 7 円形経路を用いた実験における方向ごとのエラー率

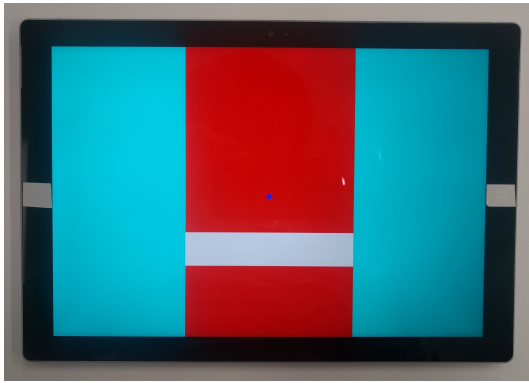


図 8 実験に用いたデバイス

#### 4.6 考察

実験の結果から、直線型経路用いたステアリングタスクについては、そのユーザパフォーマンスがステアリングの法則に適合していることが分かった。円形経路を用いたステアリングタスクについても同様にステアリングの法則に適合していると言える。エラー率については2方向とも平均で40%を超える高い数値であった。その理由としては、ボールに円形経路を通過させるにはなるべく一点を軸としてデバイスを回転させなければならないため、回転軸がぶれやすく難しい操作になるからであると考えられる。

### 5. 位置をずらした経路のステアリング実験

#### 5.1 装置

実験はタブレット PC である Microsoft 社製 Surface Pro 3 を用いて行った。使用したデバイスを図 8 に示す。ディスプレイの解像度は 2160 px × 1440 px であった。実験システムは Windows SDK を用いた UWP (Universal Windows App) として C# で開発した。チルトコントロールの実現にはデバイスに搭載されている加速度センサーを利用した。センサーのサンプリングレートは約 60 Hz とした。したがって実験システムは、約 16 ms ごとにデバイスの加速度センサーの値を取得して処理を行う。

#### 5.2 実験用システム

位置をずらした直線型経路を通過するタスクのための実験システムを実装した。その概観を図 9 に示す。前章の実験システムと同様に画面内のボールの位置をチルトコントロールによって入力することができ、経路への侵入や通過完了などの判定が行われる。経路の幅  $W$  と経路の長さ  $A$  の他に、経路の移動距離  $S$  を設定できるようにした。

#### 5.3 実験参加者

前述の実験システムを用いて、直線型経路の通過タスクを行う実験を 7 人のボランティアに実施した。3 人が男性、4 人が女性、平均年齢は 20.14 歳であり、標準偏差は 0.64 歳であった。

#### 5.4 実験デザイン

参加者にデバイスを両手で持たせ、直線型経路をはみ出さないようにボールを通過させる実験を行った。経路の長さ  $A$  と経路の幅  $W$ 、経路の移動距離  $S$  の組み合わせ 40 パターンについてそれぞれランダムな順番で行うことを 1 セットとし、繰り返し回数は 2 回とした。このタスクを左から右方向の直線型経路に対してのみ行った。ボールの直径を 20 px とし、 $W$ 、 $A$ 、 $S$  は以下のように設定した。 $W$ 、 $A$ 、 $S$  の値の単位は px である。

$W$  : 60, 80, 100, 120

$A$  : 320, 620

$S$  : -400, -200, 0, 200, 400

実際に表示される経路の幅は 20 px だけさらに大きく、経路の長さは 20 px だけ小さくしている。

参加者には経路をできるだけ速く正確に通過するように求め、経路に侵入してから通過するまでの時間を記録した。経路の通過に失敗した場合は開始位置に戻り、そこから再度試行するように指示した。

実験デバイスの重量による影響がないか調べるため、デバイスをそのまま持った場合（以降通常持ち）と上下逆さまに持った場合（以降逆さま持ち）についてそれぞれ同じ実験を行った。端末を上下逆さまに持ったときに参加者から見た経路の通過方向が左右逆になってしまうため、実験システムの経路への侵入・通過完了の判定も同様に逆になるようにした。また、デバイスを持つ手の位置の違いによる影響をなくすため、参加者には親指の位置がデバイスの縦方向中心（図 8 の白いシールの位置）になるように持たせた。

#### 5.5 結果

実験の結果について、ID と通過時間のグラフを図 10 に示す。通常持ちと逆さま持ちの両方の場合について  $R^2 > 0.96$  でステアリングの法則に適合している。また、経路の移動距離ごとにまとめた場合についても  $R^2 > 0.96-0.98$  であった。

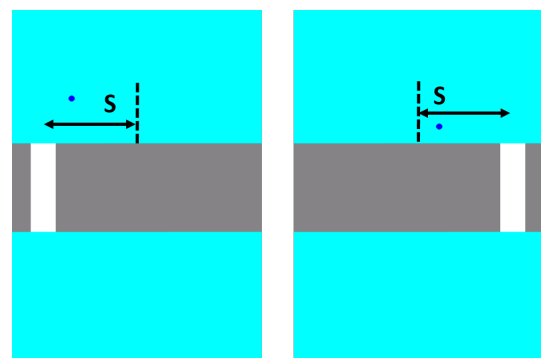


図 9 位置をずらした経路を通過するタスクのための実験システム



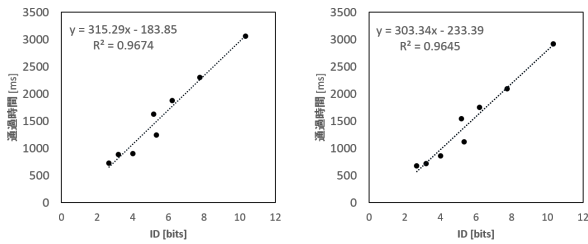


図 10 通常持ちの通過時間 (左) と逆さま持ちの通過時間 (右)

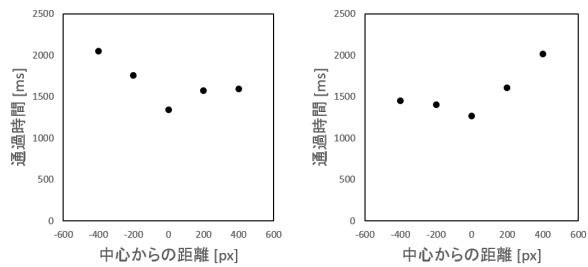


図 11 通常持ちにおける経路の移動距離と通過時間 (右),  
逆さま持ちにおける経路の移動距離と通過時間 (左)

経路の移動距離と通過時間のグラフを 11 に示す。経路の通過時間について分散分析を行った。多重比較には Bonferroni の手法を用いた。通常持ちの場合において経路の移動距離 ( $F_{4, 24} = 25.31, p < 0.01$ ) に主効果が見られた。また、逆さま持ちの場合においても経路の移動距離 ( $F_{4, 24} = 26.97, p < 0.01$ ) に主効果が見られた。

通常持ちの場合について多重比較を行った結果、経路が中心にある場合がその他の全ての場合に比べて有意に通過時間が短く ( $p < 0.05$ )、経路が変位-400 にある場合が他の全ての場合に比べて有意に通過時間が長い ( $p < 0.05$ ) ことが分かった。

逆さま持ちの場合について多重比較を行った結果、経路が変位+400 にある場合が他の全ての場合に比べて有意に通過時間が長く ( $p < 0.05$ )、経路が変位中心にある場合が経路が変位+200 にある場合に比べて有意に通過時間が短い ( $p < 0.05$ ) ことが分かった。

## 5.6 考察

実験の結果から、経路の移動距離が通過時間に影響していることが分かった。正負どちらの方向に対しても経路が中心から移動すればするほど通過時間は長くなっている。ただしその増加量は対称ではなく、デバイスを逆さま持ちにすることで、増加量の大小関係が入れ替わる現象が観察された。

通過時間が長くなる現象については、デバイスを傾げることで力のつり合いがなくなり、固定する際の負担が大きくなることが原因であると考えられる。画面の中心に経路がある場合が最も通過時間が短い、これはデバイスを持つ位置をデバイスの縦方向中心に指定したため、回転軸が

画面中心となったからであると考えられる。また、通過時間の増加量が対称でないことについては、デバイスの重量分布の偏りが原因であると予想される。今回用いたデバイスの縦方向中心に対して糸を巻いて吊るしたところ、通常持ちの場合の手前側に倒れた。これはデバイスの重心が見た目の中心よりも手前側によっているためである。したがって、経路がデバイスの重心に近い位置にあるときによりデバイスを固定しやすかったのではないかと考えられる。

## 6. まとめと課題

本研究では、チルトコントロールを用いたステアリングタスクについてそのユーザパフォーマンスを評価した。結果として、直線型・円形の経路を用いたステアリングタスクのユーザパフォーマンスがステアリングの法則に適合することが示された。ただし、ポインターを固定する方向に沿って経路をずらした状態でステアリングタスクを行うと、同じ幅、長さの経路であっても通過時間が異なることが分かった。今後は、通過時間の変化が実際にはどのような原理で起きているのかについて調査し、チルトコントロールの特徴に合わせたインタフェースデザインを検討したいと考えている。

## 参考文献

- [1] I. Scott MacKenzie and Robert J. Teather. FittsTilt: the application of Fitts' law to tilt-based interaction. NordiCHI '12, pp.568-577 (2012).
- [2] Robert J. Teather and I. Scott MacKenzie. Position vs. velocity control for tilt-based interaction. GI '14, pp.51-58 (2014).
- [3] Paul M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. Journal of Experimental Psychology, Vol.47, No.6, pp.381-391 (1954).
- [4] Johnny Accot and Shumin Zhai. Beyond Fitts' law: models for trajectory-based HCI tasks. CHI '97, pp.295-302 (1997).
- [5] Daniel Fitton, I. Scott MacKenzie, Janet C. Read, and Matthew Horton. Exploring tilt-based text input for mobile devices with teenagers. BCS-HCI '13, pp.1-6 (2013).
- [6] Linda Geronimo, Ersan Aras, and Moira C. Norrie. Tilt-and-Tap: Framework to Support Motion-Based Web Interaction Techniques. ICWE '2015, pp.565-582 (2015).
- [7] Anhong Guo and Tim Paek. Exploring tilt for no-touch, wrist-only interactions on smartwatches. MobileHCI '16, pp.17-28 (2016).
- [8] Keigo Shima, Kazusa Onishi, Ryosuke Takada, Takuya Adachi, Buntarou Shizuki, and Jiro Tanaka. Investigating Accuracy of Tilting Operation on Wrist-worn Devices with Touchscreens. CHI EA '16, pp.2705-2711 (2016).
- [9] 内藤悟史, 北村喜文, 岸野文郎. 2次元直接指示環境におけるステアリングの法則に関する検討. 電子情報通信学会技術研究報告, Vol.102, No.736, pp.1-6 (2003).