

## 2面L型立体表示システムにおける 視点とオブジェクト位置間の最適制御

渡邊優太<sup>†1</sup> 坂下善彦<sup>†2</sup>

近年、バーチャルリアリティ(VR)を見るディスプレイ装置として、利用者の周囲を囲んだ没入型ディスプレイ(Immersive Projection Display)が存在する。従来までの没入型ディスプレイでは、スクリーン数が3面から6面のもので大多数を占めており大規模な装置であったが、2面のディスプレイ(L型ディスプレイ)でも没入感が得ることが可能である見通しを得た。しかし、L型ディスプレイは正面と床面にスクリーンを設置してあるため、仮想空間内での視野が狭い。また、画角(視野角)が広がるに連れ仮想空間内のオブジェクトが歪み、正しい形状を認識することが不可能である。そこで本研究では利用者に取り付けたセンサー類の観測データを用いて、仮想空間内の視点位置(View point)とオブジェクト位置間の最適制御を行うことにより、オブジェクトの形状が歪まない限定された空間内でオブジェクトの左右情報が得られるシステムの検討を行った。

### Optimal control between the object position and view point in the 2 screen's L-type stereoscopic display system

YUTA WATANABE<sup>†1</sup> YOSHIHIKO SAKASHITA<sup>†2</sup>

#### 1. はじめに

近年、立体映像技術はTVやゲーム機などで利用されている。また、バーチャルリアリティ(VR)では運転シミュレーション等を疑似体験することが可能である。このVRのディスプレイ装置として没入感が得られ、VR空間を体験することが可能となる没入型ディスプレイ(Immersive Display)がある[1]。この没入型ディスプレイは主に情報の可視化(Visualization)や3次元CAD等で作成されたモデルの評価に利用される。現在では没入型ディスプレイは広く利用されるようになり、その背景として高性能なビデオカード(GPU)やプロジェクターの低価格化によって普及していると考えられる[2]。しかし、従来までの没入型ディスプレイは大規模であるため、広大な場所を必要とした。そこでより小規模に行うことが可能かつ、没入感を得られるディスプレイ装置として考案したのがL型ディスプレイである[3]。

このL型ディスプレイは正面と床面にスクリーンが設置されており、2面で構成されている。そのため仮想空間内のオブジェクトを正面と上面からの情報を得ることが可能ではあるが、オブジェクトの側面の情報を得ることはデバイスを用いて操作を行わない限り不可能である。そこで本研究ではマイコンボード(Arduino)に加速度センサー(ADXL345)とジャイロセンサー(ITG-3200)を搭載し、その観測値によって利用者の状態を検出することにより、デバ

イス等から操作を行わず利用者の動作によって直感的に視点変更を行うシステムの開発を行った。しかし、このシステムには問題点が存在し、画角(視野角)が狭いことによる問題がある。

利用者の動作によって視点変更が行えるシステムと述べたが、実際にオブジェクトを側面情報が欲しい場合、仮想空間内での視点位置(View Point)はオブジェクトの横まで移動しなければならない。その為、我々はL型ディスプレイの横に移動しなければならない。しかし、画角が狭いため画角が広がるに連れスクリーンに投影されたモデルの形状が歪んでしまう為、正しい情報を得ることが不可能である。そのため利用者はL型ディスプレイの正面に立ちコンテンツを見なければならない。また、センサーの特性で静止状態であっても時間と共に変動(ドリフト)してしまう現象がある。この現象はセンサーの種類に関係なく存在し、加速度センサーとジャイロセンサーのドリフトを考慮しなければならない。また、加速度センサーは変位と旋回角、ジャイロセンサーは旋回角の推定が可能となる。しかし、加速度センサーで変位を推定する場合には正確な位置情報が得ることが困難である。

本研究では利用者がL型ディスプレイの正面方向に立ち、オブジェクトの前後左右情報を得られなければならない。そこで視点とオブジェクト間を最適制御することにより、オブジェクトが歪まない範囲内での適切な移動量を得るための実験を行った。また、センサーの特性であるドリフトをソフトウェア側から抑制する為の手法について調査・検討し、その手法を実装した場合の利用者状態の推定を行った。

<sup>†1</sup> 湘南工科大学 工学研究科 電気情報工学専攻  
<sup>†2</sup> 湘南工科大学 工学部 情報工学科  
DICOM02013

## 2. 立体表示

立体表示を行うための手法はいくつか存在するが本研究で用いる手法は両眼視差方式で行う[4]。我々人間は図1のように右目と左目から得られる情報はそれぞれ位置が違う為、物体を立体として捉えることが可能である。

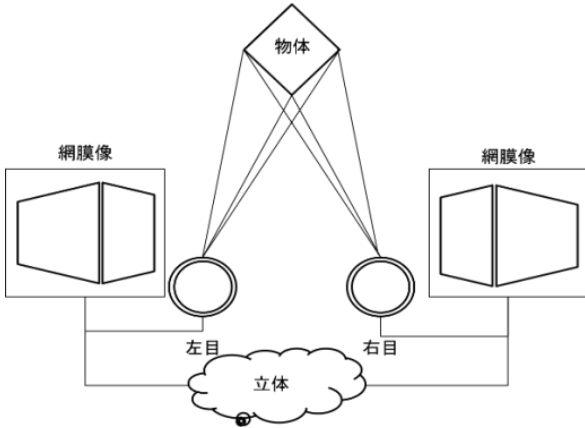


図1 両眼視差方式

本研究で用いる手法はこの原理を利用したアクティブシャッターメガネ方式で行う。この方式は右目用の画像と左目用の画像を高速に切り替えて描画を行う。そして現在のディスプレイに描画されている画像とは反対の目をシャッターメガネによって電子的に閉じることによって平面のディスプレイ装置で奥行きを感じさせることが可能となる。この方式を用いて複数のスクリーン上に立体映像を描画させることによって、没入感を得られる。しかし、それには高速に切り替えが可能となるプロジェクターとリアルタイムレンダリング可能なGPUが必要となる。

## 3. システム構成

本研究で開発を行うシステムの全体像を図2に示す。まず、利用者に取り付け利用者の状態を検知するセンサー(図中のS)から現在の観測値を計算機に送信(シリアル通信)する。計算機は受信した情報をもとに利用者の状態を推定し、その推定した情報を仮想空間(OmegaSpace)に送信(UDP)することによって仮想空間内の視点位置の制御を行う。そして、各プロジェクター(P)それぞれ対応した映像が描画される。このシーケンスを繰り返す。

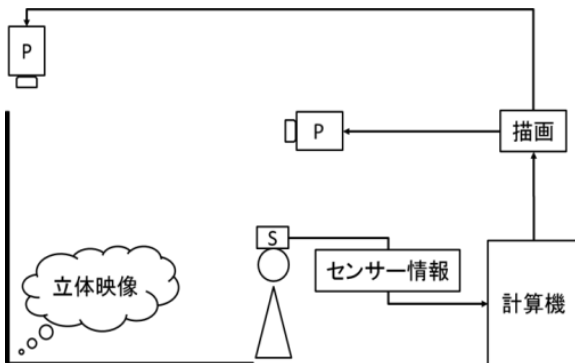


図2 システムの全体像

本研究で使用する個々のハードウェアの仕様を示す。まずは計算機の主なハードウェア構成を表1に示す。本研究では立体映像(アクティブシャッターメガネ方式)を各スクリーンに投影を行うため、リフレッシュレートが120Hz以上で描画可能なGPUとプロジェクターが必要となる為、表1に示した機材が必要となる。また、本研究で使用するOmegaSpace(ソリッドレイ研究所)ではOpenGLでオブジェクトを描画している為、GPUはOpenGLに最適化されているQuadro 5000を採用した。

表1 ハードウェア構成

Processor	Intel(R) Xeon(R) CPU@ 2.40GHz
RAM	12288MB
GPU	NVIDIA Quadro 5000
プロジェクター	三菱 LVP-XD600

センサー情報を取得するために使用するマイコンボード(Arduino)の主な仕様を表2に示す。こちらのArduinoはI<sup>2</sup>C(Inter-Integrated Circuit)により各種センサーと通信を行い、計算機側で実行中のソフトウェアに加速度・角速度の情報を送信する。本来であればArduino側で利用者の状態の推定を行うことが好ましい。しかし、利用者状態を算出する過程でメモリ資源を多く使用する為、Arduinoの場合では必要最低限のメモリ容量を満たしていない。そこで推定を算出する処理は計算機で行った。

表2 Arduino の仕様

動作電圧	5V
SRAM	2KB
Flash メモリ	32KB
クロック周波数	16MHz

加速度・ジャイロセンサーの仕様をそれぞれ表3及び表4に示す。本研究ではI<sup>2</sup>Cによって情報を取得し、重力加速度(1g=9.80665)の最大観測値は±2gとなる[5]。換算係数(Scale factor)は1bitあたりの重力加速度を示しており、最小(Min)で3.3mg/LSB、最大(Max)で4.3mg/LSBとなる。これにより3軸それぞれ最大出力は異なっており、それぞれの軸を地面方向(1g フィールド)に向けた場合の出力が1gの感度となり、この計測結果は6章で示す。ジャイロセンサーの最大出力は±2000deg/secとなっている[6]。加速度・ジャイロセンサーは伴に人の動きを検知するには十分な性能となっている。

表3 加速度センサーの仕様

出力	デジタル		
計測方向	3軸(x,y,z)		
最大出力	±2g(最大で±16g)		
感度	Min	Typ <sup>2</sup>	Max
	232LSB/g	256LSB/g	282LSB/g
換算係数	3.3mg/LSB	3.9mg/LSB	4.3mg/LSB

表 4 ジャイロセンサーの仕様

出力	デジタル
計測方向	3 軸(x,y,z)
最大出力	±2000deg/sec
感度	14.375LSB/(deg/sec)

## 4. 推定手法

### 4.1 原理

加速度センサーとジャイロセンサーから得られる加速度・角速度のデータからどのようにして利用者の状態の推定を行うのか述べる。まず、加速度 $a$ は速度 $v$ の時間変化量の事である為、(1)式のように時間 $t$ で積分することにより速度へ変換が可能となる。

$$v(t) = \int a(t)dt \quad (1)$$

速度 $v$ も変位 $x$ の時間変化量の事であるため(2)式のように時間 $t$ で積分することにより変位へ変換が可能となる。加速度から変位を推定する場合、2 階積分することにより初期位置からの移動距離を推定することが可能である[7]。

$$x(t) = \int v(t)dt \quad (2)$$

また、加速度センサーから得られる加速度情報により旋回角を推定することも可能となる。図 3 のように X 軸が重力方向へ推移し、X 軸の出力が  $1g$  となる場合では旋回角が  $\theta = 90^\circ$  となる。これにより加速度センサーでも旋回角の推定は可能である。しかし、重力の影響を受けない軸の場合では旋回角の推定を行うことは不可能である為、ジャイロセンサーを用いて旋回角の推定を行う。

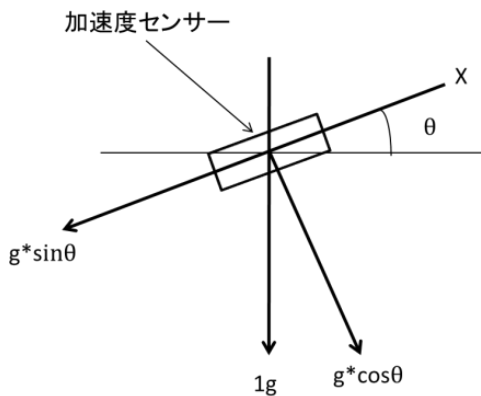


図 3 加速度センサーによる旋回角の推定

角速度 $\omega$ の場合、(3)式のように時間 $t$ で積分することにより利用者の旋回角 $\theta$ を推定することが可能となる。

$$\theta(t) = \int \omega(t)dt \quad (3)$$

しかし、本研究ではセンサー出力によって推定を行う為、単純に時間積分するだけでは精度の良い利用者の状態を推定することは不可能である。本研究では台形公式(台形則)を用いて積分を行う。2 次関数を用いたシンプソン法で算

出を行うと一般的には台形公式より優れているが、センサーからの観測値はドリフト成分を含んでいるため誤った値に近似を行うことになる。その為、台形公式を用いて近似の方が誤差の算出が遅くなる。

### 4.2 センサー特性

本研究ではセンサーを用いて推定する場合、センサー特性を考慮しなければならない。4.1 節で述べた台形公式を用いて算出する前にバイアス値を与えなければならない。これはセンサーの物理特性で現在の観測値が過去の計測値(動作)に依存してしまう(ヒステリシス)。このヒステリシスが原因でバイアス(偏り)があるためバイアス値を与える。バイアス値は(4)式で算出した結果をバイアス値としている。 $x$ は出力される観測値であり、 $bs$ はバッファサイズとなる。観測値 $x$ はひとつ飛ばしてバッファに格納される。これは過去の計測値に依存するためひとつ飛ばして行っている。(4)式を用いて算出したバイアス値は受信される観測値にバイアス値を減算することにより、偏り分を差し引くことが可能となる。

$$bias = \frac{x_0 + x_2 + x_4 \dots x_{bs-2} + x_{bs}}{bs} \quad (4)$$

### 4.3 推定方法

4.2 節で述べたバイアス値を加えた数式は(5)式となる。式中にある $S_p$ はセンサーの最大観測値の事を指し、観測値は  $int$  型で送信されるため  $int$  の最大値である  $32767(2^{16})$  で  $S_p$  を割ることにより、1bit(LSB)あたりの重みをつけることが可能となる。そして台形公式を用いて現在の観測値 $p_i$ と過去の観測値 $p_{i-1}$ にそれぞれバイアス値 $bias$ を引いてサンプリング周期である  $Int$  を乗算した値が瞬間の変位と旋回角の変化量となり、その値の累積が利用者の現在の状態となる。

$$\int_{p_{i-1}}^{p_i} f(t)dt \approx \frac{S_p}{32767} * \frac{p_{i-1} + p_i - bias * 2}{2} * Int \quad (5)$$

## 5. システム仕様

### 5.1 入出力

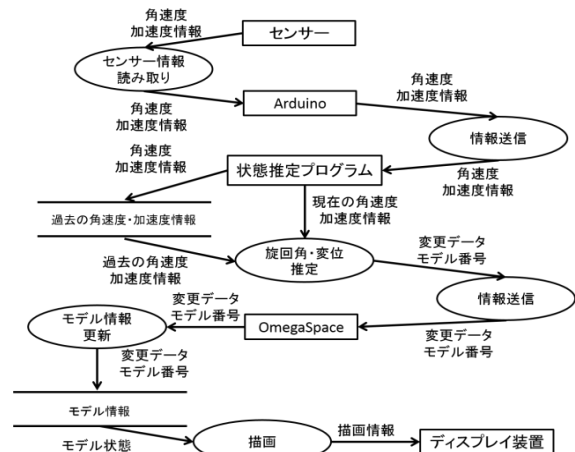


図 4 システムの入出力

本研究で開発するシステム全体の入出力(I/O)は図4のようになる。図中に OmegaSpace とは VR 空間を構築する際に使用したソフトウェアである。また、状態推定プログラムは、Arduino からシリアル通信で送信される観測値を受信し、利用者の状態を推定してから UDP/IP で OmegaSpace に情報を送信する機能を備えている。

本研究で VR 空間を構築するために使用した OmegaSpace は自作したプログラムからオブジェクトを制御がすることが可能となる。それを利用してオブジェクトが持つ Translate(3次元空間での座標位置)と Rotate(3軸の旋回角)により、VR 空間上でのカメラ制御を行う。

## 5.2 通信仕様

本研究で使用する通信は I<sup>2</sup>C、UDP、シリアル通信となっている。そこで各通信の通信仕様を示す。I<sup>2</sup>C はマスター(Arduino)とスレーブ(ADXL345・ITG-3200)が存在し、Arduino が一方的に制御を行う通信となっている。こちらを用いてセンサー内部のレジスタから観測値を読み取っている。各レジスタを表5に示す。I<sup>2</sup>Cの場合ではデータ(16bit)の送信を2回に分けて行う。その為、レジスタアドレスは各軸に2つ存在する。

表5 I<sup>2</sup>Cの仕様

センサー	レジスタアドレス	
	ADXL345	I <sup>2</sup> C Address
X 軸		0x32~0x33
Y 軸		0x34~0x35
Z 軸		0x35~0x37
ITG-3200	I <sup>2</sup> C Address	0x68
	X 軸	0x1D~0x1E
	Y 軸	0x1F~0x20
	Z 軸	0x21~0x22

UDP についての主な仕様を表6に示す。UDP で通信する為に Windows SDK というプラットフォームに含まれる ws2\_32.lib を使用する。表中の送信先とは OmegaSpace であるためそのポート番号は 8888 となる。OmegaSpace の場合では UDP か RS232C で制御が可能となっている。UDP か RS232C の2通りしかない理由としてはリアルタイムでレンダリングを行う為、TCP ではデータに対する確認応答があるので信頼性の高いプロトコルではあるが、その分 UDP と比べると低速である為である。

表6 UDPの仕様

プラットフォーム	Windows SDK(Version7.1)
ライブラリ	ws2_32.lib
IP	IPv4
送信先ポート番号	8888

次にシリアル通信の主な仕様を表7に示す。ボーレートは 115200(230400bps)となっており、Arduino と PC 間の通

信ではこのボーレートが最大となっている。シリアル通信で Arduino から送信される加速度と角速度の観測値は文字列としてシリアルポートへ送信される。そして状態推定プログラムがシリアルポートを読み込み”改行”を受信すると、文字列を閉じ状態を推定して OmegaSpace に情報を送信する。そして新たな観測値がシリアルポートに入ると上記の処理を行い、このサイクルを繰り返し行っていく。

表7 シリアル通信の仕様

接続方法	USB 接続(A-B Type)
Baudrate	115200

## 6. 制御系

### 6.1 最適制御

本研究の核心となる最適制御について述べる。まず、問題点はいくつかあり、1章で述べた画角の狭さがあるため移動範囲に制限がある。そこでL型ディスプレイの正面に立ち、固定された位置ではあるが左右情報の得られる手法を考案した。

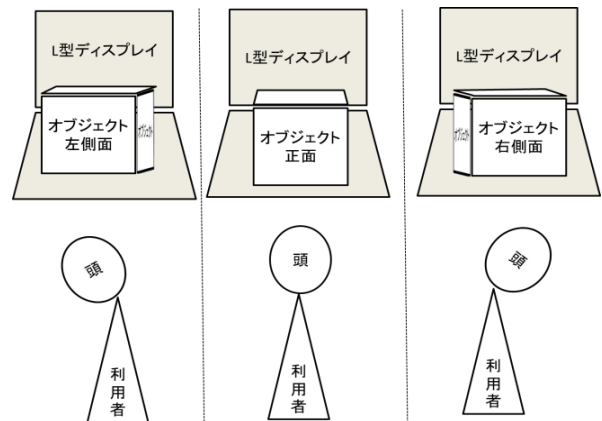


図6 左右情報の取得手法

その手法は4.1節で述べた加速度センサーによる旋回角の推定を利用した。頭に取り付けたセンサーが傾くことにより、利用者は移動せずに視点の移動が可能となる。これは利用者の横方向に向いている軸(Y軸)が地面方向に傾くことにより、重力によって加速度が上昇する。これを利用して、ある閾値を超えた場合に側面情報が得られる位置へと移動する。これによって利用者は頭を見たい情報の方向へ頭を傾けることにより、利用者の見たい情報方向からの視点でオブジェクト見ることが可能となる。

まず、3章で述べたように加速度センサーの場合3軸それぞれの感度が異なっている為、それぞれの軸の感度がどれほどのものか計測しなくてはならない。その計測方法は水平な3脚の上にセンサー本体を固定して、3軸それぞれを地面方向に向けて加速度を取得すると、3軸の1gあたりの感度を計測することが可能となる。それを計測した結果を表7に示す。この値は3軸それぞれの値を1万個集め、その平均した値となっており、この感度を元に軸の閾値の設定を行う。

表 7 1g フィールド上の感度

	X 軸	Y 軸	Z 軸
感度(1g)	266.5711	274.9117	269.9096
感度(-1g)	-255.225	-253.428	-231.979

## 6.2 ドリフト制御

1 章で述べたドリフトについて考慮しなければならない。このドリフトは様々な要因から発生し、動作しているセンサーの内部が温度変化によって電圧が変動して発生する温度ドリフトやオフセットによるオフセット・ドリフトが存在する。オフセット・ドリフトは 4.2 節で述べたバイアス値を利用して抑えることが可能となる。しかし、温度ドリフトはこの処理では全てを除去することができない。そのためこのドリフト成分を抑えるための手法をいくつか考えられる。

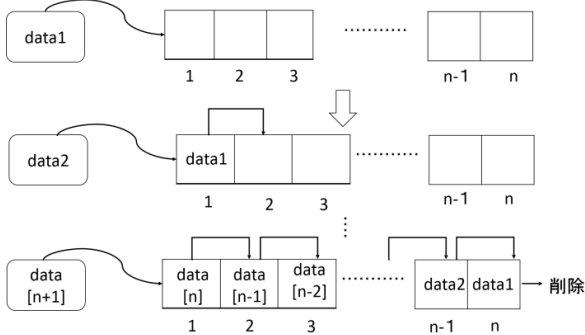


図 7 平滑化処理

その 1 つは平滑化(スムージング)を行うことである。この平滑化は図 7 のような処理となる。初めは図中にある  $n$  個分のセンサーからのデータを集める。一度集めたらそのデータ群の平均を算出する。そして算出した値が現在の観測値となり、最も古いデータは削除してまた新たなデータを加え平均を算出する。これを繰り返すことによりドリフト成分を抑え込むことが可能となる。しかし、抑え込むことは出来るがドリフト成分を完全に排除することは出来ない。この平滑化処理ではドリフト成分を他の含まない値に近づけ、目立たないように平らにしている。その為、根本的なドリフトの解決方法とはならず、累積値を初期化する処理を行わなければならない。初期化することにより、初期化後は精度の高い値を算出することが可能である。また、要素数  $n$  をより多く集めた場合ではよりドリフト成分を抑えることが可能となるが、有効データの算出には時間がかかる。

図 8 に平滑化を行った場合に推定した旋回角(度数法)を示す。こちらはセンサーのバイアス値を加味してから、(4)式を用いて 5 分間建物に固定して得られた観測値から推定を行った旋回角となっている。Arduino 側では 10ms ごとに情報を読み取り送信を行い、そのデータから算出するバイアス値は 1024 個のデータから算出し、平滑化処理では 1000 個のデータで行った。

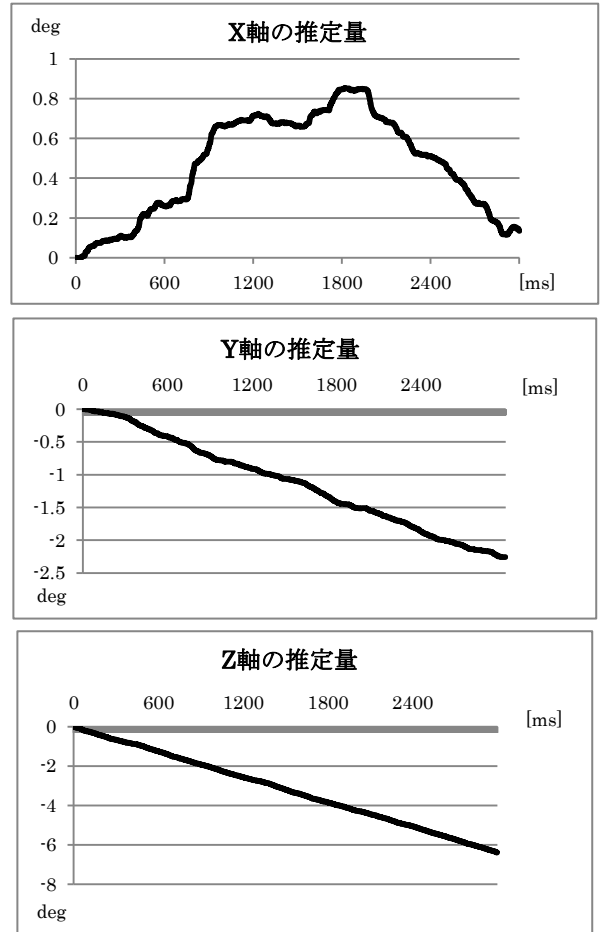


図 8 固定した状態での推定値

この推定結果を見る限りでは本来、旋回角は 0 度であるはずの旋回角が特に Z 軸では 5 分後には 6 度以上の誤差を生じている。この推定結果は微小な値であっても推定結果を累積しているため、微小な値が積み重なって 5 分後にはとても大きな誤差になってしまう。また、図 8 ではセンサー本体を固定した状態で行っており、実際に利用者に取り付けて計測を行った場合では推定値の発散はより早くなると考えられる。

## 7. 実験

### 7.1 実験概要

今回の実験では被験者それぞれに L 型ディスプレイの正面に立ち、左右それぞれの視点からコンテンツの形状を確認してもらった。センサー本体はヘッドホンに固定し、利用者はヘッドホンを装着した状態でバイアス値を取得する。また、6.1 節で述べた閾値は傾斜角  $30^\circ$  (0.5g) とし、被験者それぞれに本研究で開発したシステムを利用し、システムの問題点や今後の方針について考察した。

### 7.2 実験結果

本システムを被験者に利用してもらった結果、閾値に関しては被験者にアンケートとったところ無理なく視点変更が行えるという回答であった。しかし、時間が経過するに連れ、思っている方向に向かないという被験者全員から

の回答であった。そこで図8と同様の条件で行い、利用者にセンサーを装着した状態でバイアスを算出し、静止状態での旋回角の推定を行う実験を新たに行った。その推定量を図9に示す。その実験の結果、各軸の推定量は全て5分後には100度以上の誤差が発生する結果となった。この実験ではセンサーを取り付けているのは利用者の頭部であるため、完全な静止状態ではないが100度以上の誤差は異常である。この結果からバイアスが正確に間引くことができていないと考えられる。

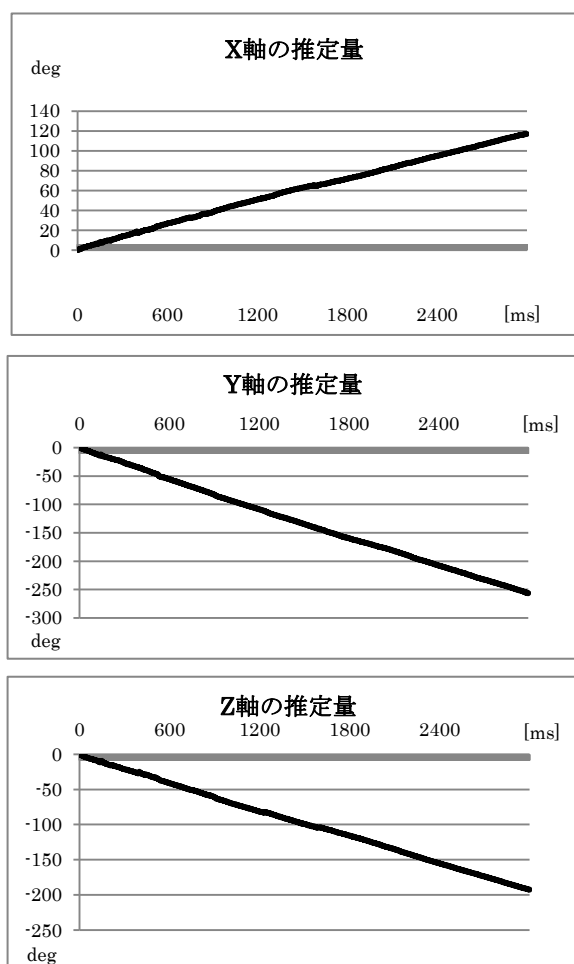


図9 利用者に取付け状態での推定量

バイアス値を取得する場合、利用者(センサー)は静止状態でなくてはならない。しかし、利用者は静止状態ではあるが、微小ながら変化がある為に正しいバイアス値が算出することが不可能である。それにより、偏りを除く際にバイアスが除去しきれない、または除き過ぎてしまうと考えられる。そしてサンプリング周期ごとにバイアス分の誤差が発生し、発散する結果になると考えられる。その為、図9ではドリフトがあるため線形ではないが、線形に近い形で推定値の誤差が発生していると考えられる。

## 8. まとめ

本研究では視点とオブジェクトの最適制御を行い、直感的な操作でオブジェクトの側面を捉えることが可能となっ

た。また、平滑化処理やバイアス値を算出し、ドリフトやヒステリシスによるバイアスを考慮したシステムとなっている。しかし、実際に利用者に取り付けて行う場合にはドリフトやバイアスの影響を受け、時間経過すると共に推定値が大きく発散する結果となった。

このシステムは立体映像を投影している為、長い時間見ていると目や脳に疲労感がある[8]。その為、長い時間見ることが考えられないが、5分程度での推定結果が最大で250°以上の誤差がある。その為、現状のシステムではドリフトやバイアスの誤差を抑えない限り、実用的なシステムではない。

今後はソフトウェア側からドリフトやバイアスによる誤差を抑える為の手法を考案する。そして、その手法を実装しなければならない。

## 参考文献

- 1) 小木哲郎, “没入型ディスプレイの特徴と応用の展開”, ヒューマンインタフェース学会論文誌, Vol.1, No.4(1999)pp.43-49.
- 2) 小木哲郎, “没入型ディスプレイとインタラクション技術”, 計算工学 13(4), pp.1917-1920, 2008-10-3.
- 3) 渡邊優太, 佐藤康明, 坂下善彦, “2面L型ディスプレイによる立体表示システムの試作”, 情報処理学会 第75回全国大会
- 4) 森尚平, 一刈良介, 柴田史久, 木村朝子, 田村秀征, “実世界の撮影現場で3D映画演出を事前検討できるMR-PreVizシステムの開発”, 日本バーチャルリアリティ学会論文誌 17(3), 231-240, 2012-09-30
- 5) ANALOG DEVICES, “ADXL345”, <[http://www.analog.com/static/imported-files/data\\_sheets/ADXL345.pdf](http://www.analog.com/static/imported-files/data_sheets/ADXL345.pdf)>
- 6) InvenSense, “ITG-3200 Product Specification”, <<https://www.sparkfun.com/datasheets/Sensors/Gyro/PS-ITG-3200-00-01.4.pdf>>
- 7) 西形利一, “速度計測アプリケーションの可能性”, トランジスタ技術 2007年12月 pp133-140
- 8) 矢野澄夫, “立体映像が視覚機能に及ぼす影響”, 映像情報メディア学会技術報告 35(15), 21-26, 2011-03-07