

4 輪駆動ロボットを用いた魚釣りシミュレータの製作

2020SC051 永井 翔麻

指導教員：奥村 康行

1. 研究背景

2021 年度の日本の釣り人口は、日本生産本部の調査によると、560 万人である。2012 年度では 810 万人、2015 年度では 720 万人、2018 年度では 620 万人である。[1] 図 1 にグラフを示す。これにより、日本の釣り人口は、年々減少している傾向にあることが分かる。そこで、少しでも多くの人に釣りの楽しさを知ってもらいたいと思ひ、実際の釣りを体験するよりもっと手軽に体験できるシミュレータを製作する事とした。

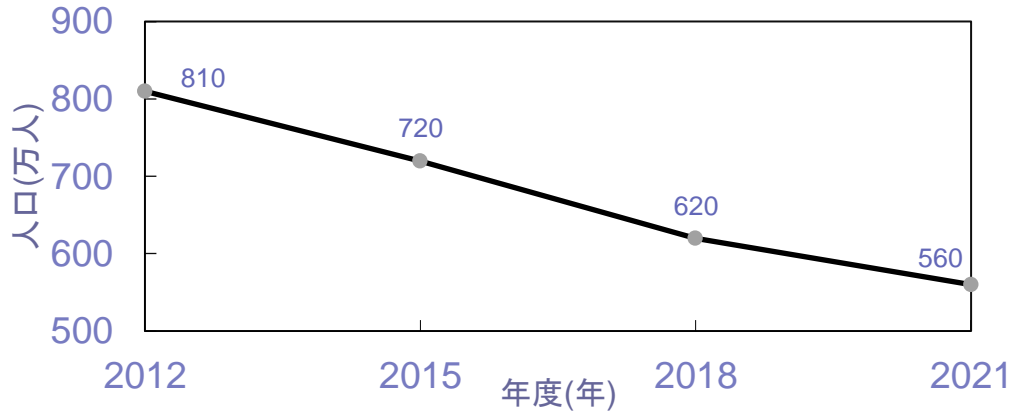


図 1 日本の釣り人口の推移

2. 先行研究

先行研究[2]では、魚の垂直方向の張力を再現するシミュレータを研究していた。Arduino MEGA2560 とサーボモータ、釣り糸を巻き取る為の木製のボビンのような物を用意する。サーボモータとボビンを組み合わせ、ボビンに巻いた釣り糸と釣り竿とを繋ぐ。サーボモータを Arduino MEGA2560 で制御しボビンが回転することで釣り糸を巻き取り、垂直方向の魚の張力を再現するシミュレータを製作していた。先行研究での全体像を図 2 に示す。青で囲われているものがサーボモータ、サーボモータの左側に取り付けられているものがボビンの様な物、右下にあるものが Arduino MEGA2560、黒線は釣り糸である。

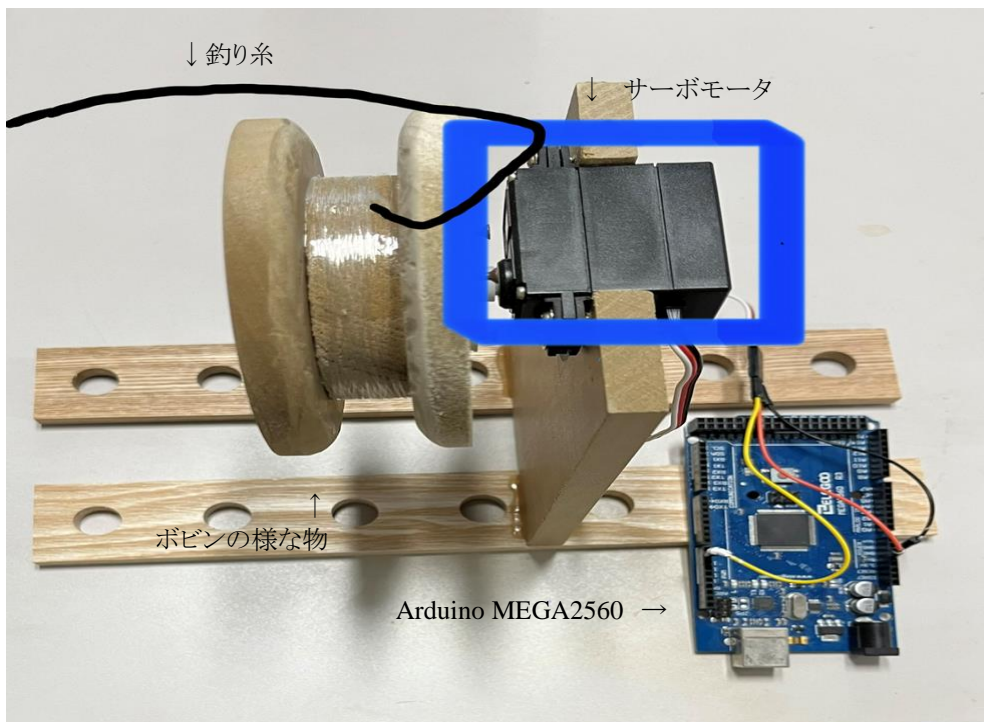


図 2 先行研究の全体像

2.1 先行研究の結果

先行研究の結果として図 3 のような結果を得ていた. 赤線が先行研究のシミュレータが発揮する張力を測定したデータ, 黒線が再現する魚の張力データである. 先行研究の課題点として, 体験者が釣り糸を引っ張る事を考慮する必要はある点. よりリアルな魚の張力を再現するためには, 垂直方向だけでなく, 水平方向の運動も加える必要がある点が述べられていた.

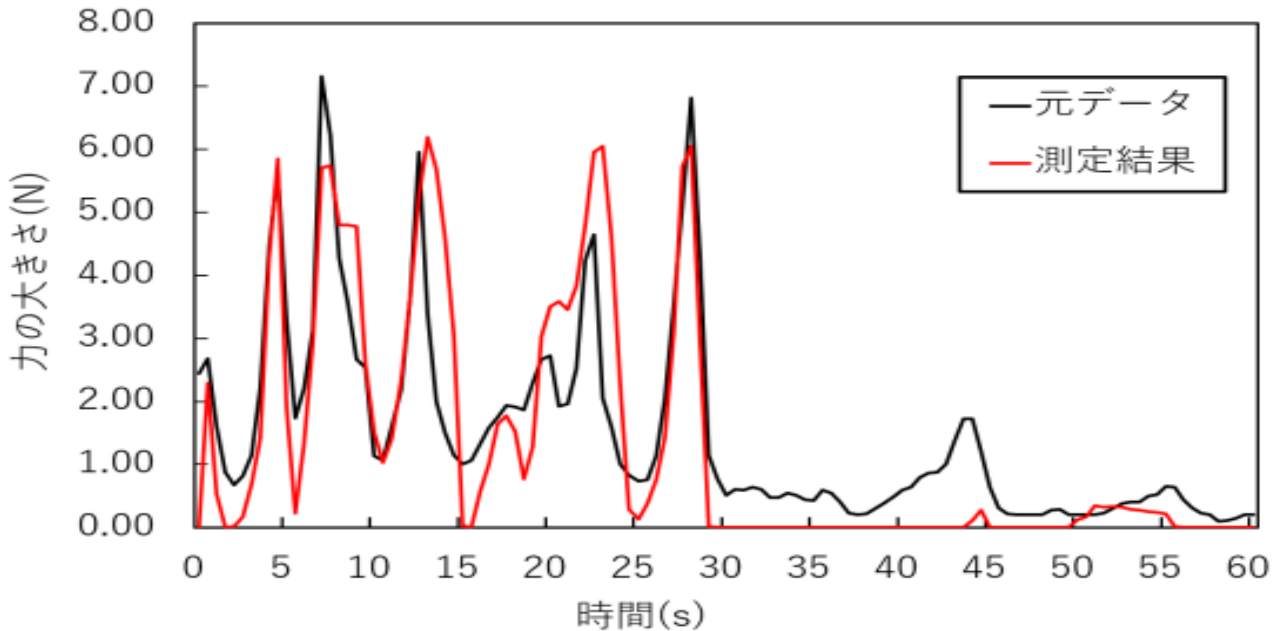


図 3 先行研究の結果

3. 本研究の全体像

本研究は, 先行研究での課題点に挙げられていた水平方向の運動を加えることが目的である. 先行研究で製作された垂直方向の魚の張力を再現するシミュレータに, 水平方向の魚の張力を再現するシミュレータを加えることで, よりリアルな魚の張力を再現しようという研究である.

水平方向の魚の張力を再現する方法として, ロボットに水平方向の魚の動きを再現させる方法を取る. 4 輪ロボット用意し釣り糸をつける. 4 輪ロボットは魚の動きを再現するように動作し, 糸を引っ張ることで水平方向の魚の張力を再現する. 垂直方向の魚の張力は, 4 輪ロボットの上にサーボモータを搭載する. そのサーボモータが回転し, 釣り糸を巻き取ることで再現する. これらの機材を図 4 の様に取り付け, 図 5 の様に 4 輪ロボットと, サーボモータをそれぞれ制御する Arduino2 個とサーボモータ, 各タイヤのモータを接続する. 4 輪ロボットが水平方向, サーボモータが垂直方向にそれぞれ糸を引っ張ることで生まれる合力で, 魚の張力を再現しようという研究である. 下の 4 つのタイヤが付いている直方体が 4 輪ロボット, その上に乗っているものがサーボモータ, サーボモータを制御する Arduino, 右側にあるのが 4 輪ロボットを制御する Arduino である.

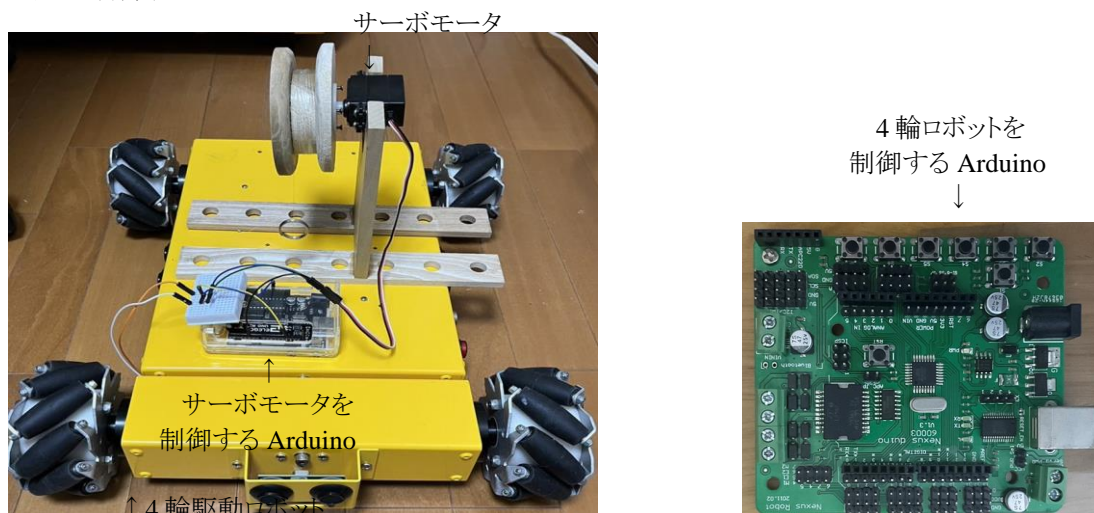


図 4 全体像

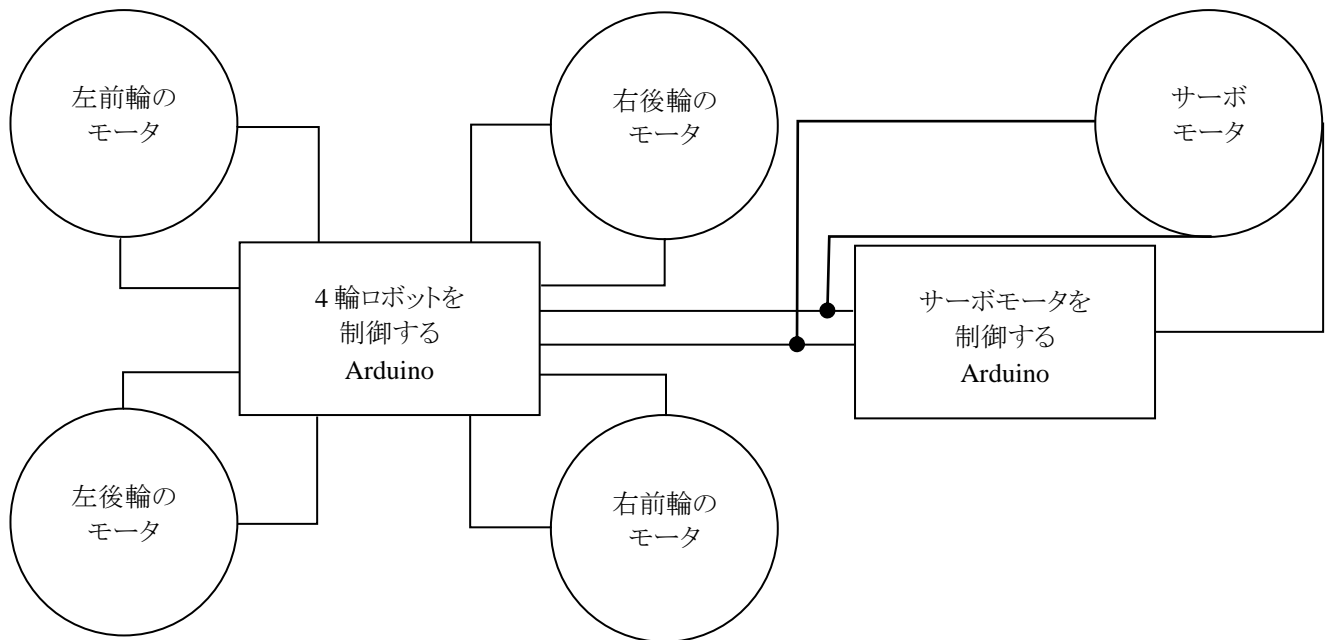


図 5 回路図

4. 魚の張力

参考文献として大浜秀規氏が、マスノスケを釣獲した際の張力データを取っていた[3]が、こちらを参考に製作させて頂く。張力データに関しては、図 6 に示す。0.1 秒毎に魚が糸を引く張力を数値として表にしたデータ、及びそのグラフを 27 匹分頂戴している。

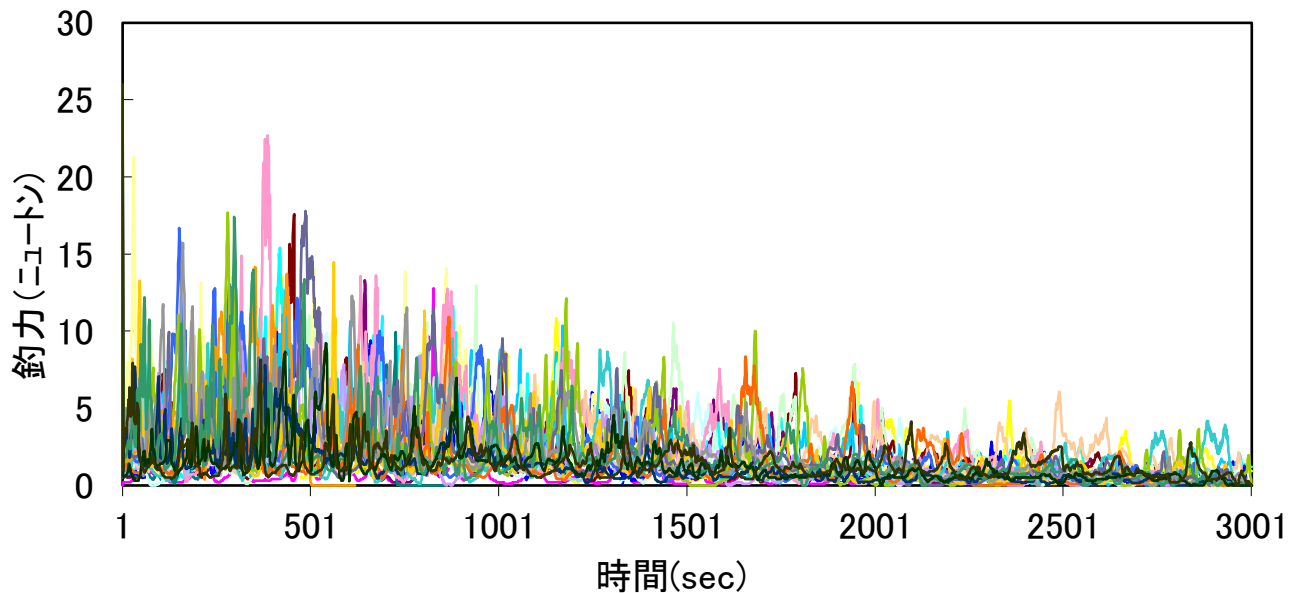


図 6 頂戴したマスノスケ 27 匹分の張力データ

本研究で再現する張力を決定するにあたり、頂戴した 27 匹分のデータの中から一番大きい張力を発揮していた個体の張力を再現する事とした。図 7 にその個体の張力をグラフに示した。図 7 より、再現する張力の最大値は 22N 程であることが見られる。

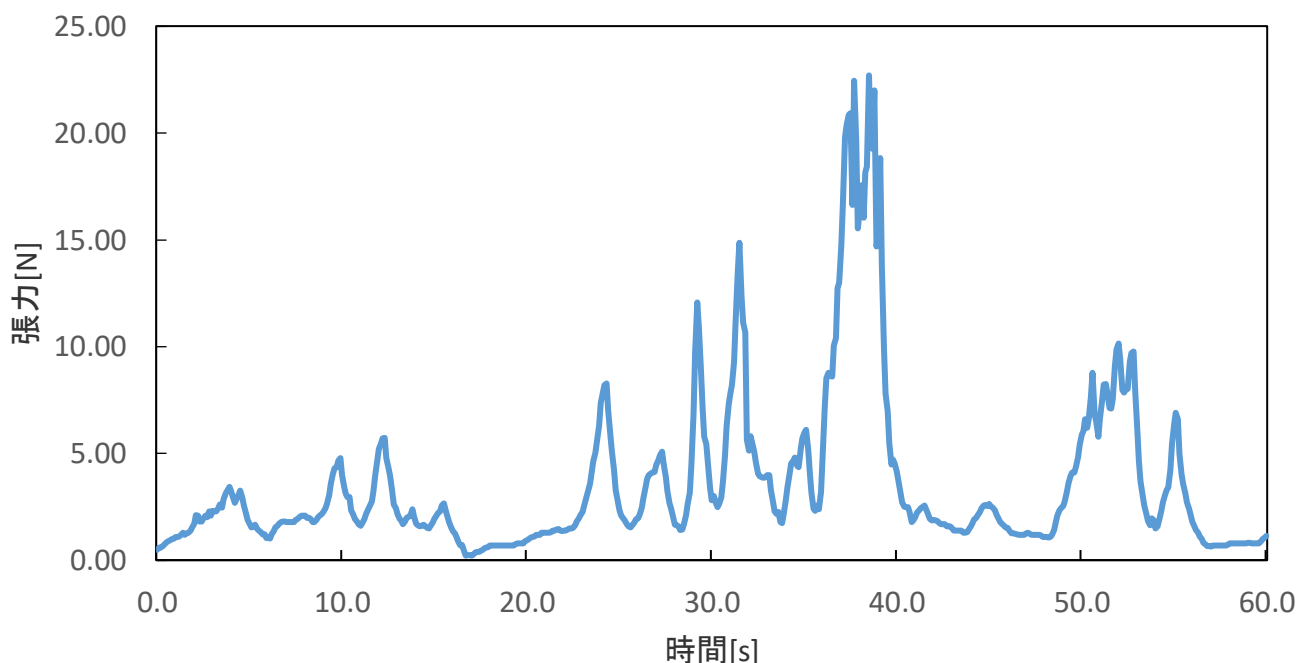


図 7 再現する個体の張力グラフ

5. 本研究の詳細

本研究で利用する機材, 設計方針, 制御方法について述べる.

5.1 利用する 4 輪ロボット・モータの詳細

本実験で利用する 4 輪ロボットは, Arduino で制御することが可能で, メカナムホイールというタイヤを搭載している. 図 4 にあるように, タイヤのホイールに対して斜めにゴムが取り付けられているタイヤの事である. 一般的なタイヤであれば, 各タイヤが回転する方向によって, 前進, 後退, の 2 種類の方向に進むことが出来る. しかし, メカナムホイールの場合には各タイヤが回転する方向によって方向転換なしで前後左右斜めの 8 方向へ進むことが可能となっている. また, Arduino から各モータに数値を入力する事でその数値に対応した速度でタイヤが回転する. 各モータに同じ数値を入力したところ, それぞれのモータが別々の速度で動作したため, 次のように各モータに入力される数値を調整した. 右前輪に入力される値=入力値 \cdot 8, 左前輪に入力される値=入力値 \cdot 28, 右後輪に入力される値=-5, 左後輪に入力される値=入力値 \cdot 23. 入力値に対して, 4 輪ロボットが発揮する張力を測定した結果図 8 のような結果を得られた.

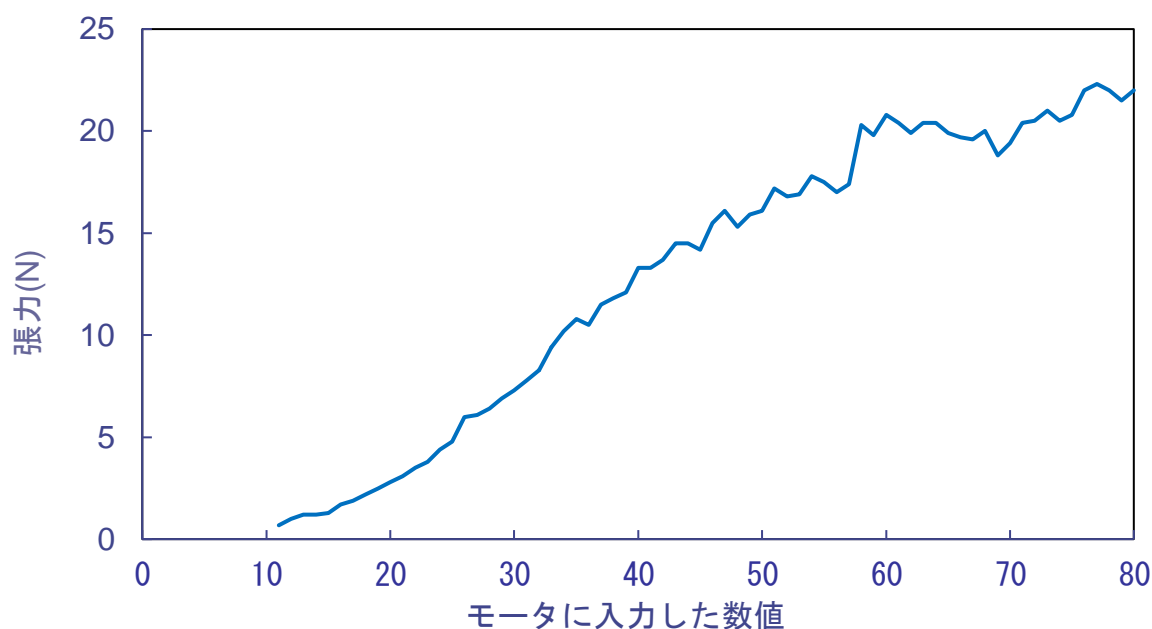


図 8 入力値に対しての4輪ロボットが発揮する張力

張力の測定方法としては測定器を 55cm の高さの台に乗せ固定する。測定器と 4 輪ロボットとを約 100cm 離れた位置に設置し、測定器とシミュレータとを釣り糸で繋ぐ。4 輪ロボットの各タイヤのモータに Arduino から 0 から順に数値を入力し、張力を測定した。

サーボモータは最大トルクが 4.1kg・cm(4.8V)回転可能域は 360° 寸法は 39.5×20×25.6(mm)スピードは 0.27 秒/60° (4.8V)の物を使用する。こちらも Arduino から値を入力する事でその数値に対応した速度でサーボモータが回転する。入力値に対しての張力を測定した結果、図 6 のような結果を得られた。

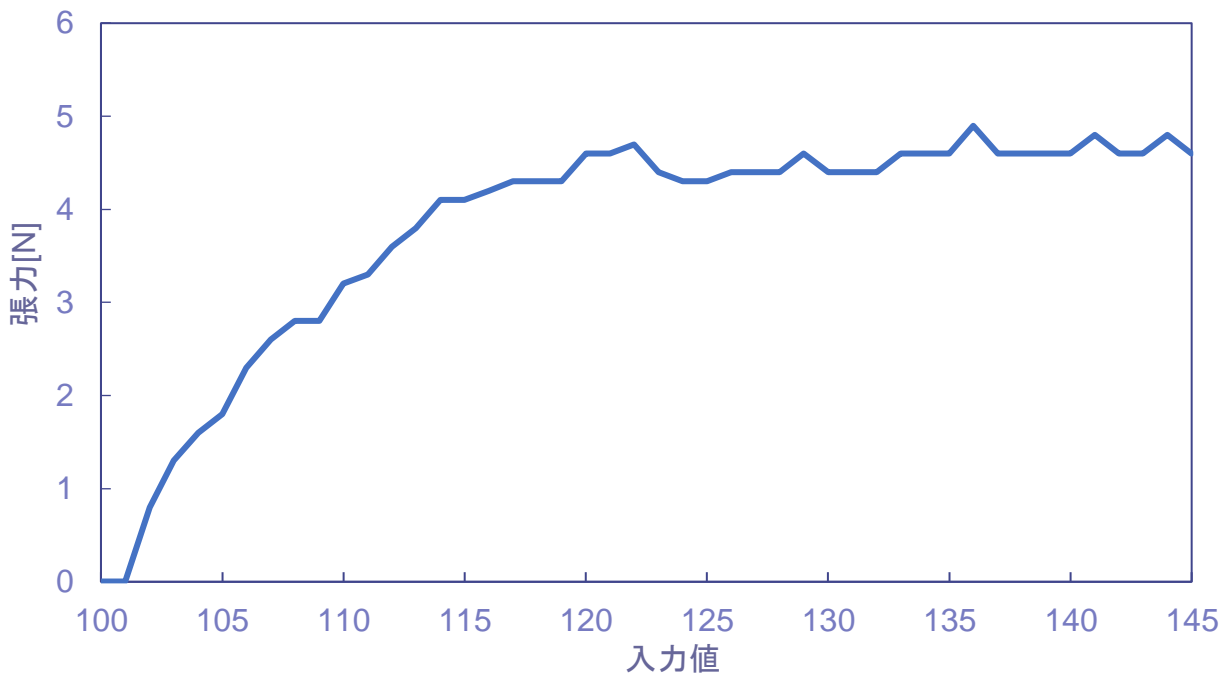


図 9 入力値に対してのサーボモータが発揮する張力

張力の測定方法としては測定器を 55 cmの高さの台に乗せ固定する。測定器とサーボモータとを約 100cm 離れた位置に設置し、測定器とシミュレータとを釣り糸で繋ぐ。サーボモータに Arduino から 0 から順に数値を入力し、張力を測定した。

5.2 設計方針

図 7 に示される張力が先行研究で製作されたサーボモータの張力と 4 輪ロボットの張力との合力となるように設計する。4 輪ロボットは、体験者から離れて行く方向へ円を描くように進むよう設計する。これは、大浜氏に魚が逃げる際どのように逃げる傾向にあるのかと伺ったところ、一方向に逃げ、逃げられないと判断し次は逆方向に逃げようとする結果、円を描くように逃げる傾向にあると仰っていた為である。

5.3 各種機材の制御方法

4 輪ロボットとモータの合力が図 7 に示したグラフに近づくように図 8, 図 9 を参考に入力値を調整してプログラムを組む。4 輪ロボットは 0.5 秒間隔で指定した張力を発揮するよう動作させる。4 輪ロボットを制御する Arduino には次のようなプログラムを書き込む。

```
int FRDPin = 7;
// 右前輪の回転方向を制御する PIN 番号
int FRVPin = 10;
// 右前輪の回転速度を制御する PIN 番号
int FLDPin = 2;
// 左前輪の回転方向を制御する PIN 番号
int FLVPin = 3;
// 左前輪の回転速度を制御する PIN 番号
int BRDPin = 8;
// 右後輪の回転方向を制御する PIN 番号
int BRVPin = 9;
```

```

//右後輪の回転速度を制御する PIN 番号
int BLDPin = 12;
// 左後輪の回転方向を制御する PIN 番号
int BLVPin = 11;
// 左後輪の回転速度を制御する PIN 番号

int D1,D2,D3,D4,D5,D6,D7,D8,D9,D10;
int D11,D12,D13,D14,D15,D16,D17,D18,D19,D20;
int D21,D22,D23,D24,D25,D26,D27,D28,D29,D30;
int D31,D32,D33,D34,D35,D36,D37,D38,D39,D40;
int D41,D42,D43,D44,D45,D46,D47,D48,D49,D50;
int D51,D52,D53,D54,D55,D56,D57,D58,D59,D60;
int D61,D62,D63,D64,D65,D66,D67,D68,D69,D70;
int D71,D72,D73,D74,D75,D76,D77,D78,D79,D80;
int D81,D82,D83,D84,D85,D86,D87,D88,D89,D90;
int D91,D92,D93,D94,D95,D96,D97,D98,D99,D100;
int D101,D102,D103,D104,D105,D106,D107,D108, D109,D110;
int D111,D112,D113,D114,D115,D116, D117,D118,D119,D120;
int D121;
//方向を決める変数を各 0.5 秒で 60 秒分宣言する.

int V1,V2,V3,V4,V5,V6,V7,V8,V9,V10;
int V11,V12,V13,V14,V15,V16,V17,V18,V19,V20;
int V21,V22,V23,V24,V25,V26,V27,V28,V29,V30;
int V31,V32,V33,V34,V35,V36,V37,V38,V39,V40;
int V41,V42,V43,V44,V45,V46,V47,V48,V49,V50;
int V51,V52,V53,V54,V55,V56,V57,V58,V59,V60;
int V61,V62,V63,V64,V65,V66,V67,V68,V69,V70;
int V71,V72,V73,V74,V75,V76,V77,V78,V79,V80;
int V81,V82,V83,V84,V85,V86,V87,V88,V89,V90;
int V91,V92,V93,V94,V95,V96,V97,V98,V99,V100;
int V101,V102,V103,V104,V105,V106,V107,V108,V109,V110;
int V111,V112,V113,V114,V115,V116,V117,V118,V119,V120;
int V121;
//4 輪ロボットのモータが回転する速度を決める変数を各 0.5 秒で 60 秒分宣言する.

void setup() {
  TCCR2B = TCCR2B & 0b11111000 | 0x01;
}

// 前進するプログラム, 以降 front(int 型変数)と記述することで呼び出すことが出来る.
void front(int V){
  analogWrite(FRVPin,V-8);
//右前輪の回転する速度を入力値-8 に決定する.
  digitalWrite(FRDPin,LOW);
//右前輪の回転方向を時計回りに決定する.
  analogWrite(FLVPin,V+28);
//左前輪の回転する速度を入力値+28 に決定する.
  digitalWrite(FLDPin,HIGH);
//左前輪の回転する方向を反時計回りに決定する.
  analogWrite(BRVPin,V-5);
//右後輪の回転する速度を入力値-5 に決定する.
  digitalWrite(BRDPin,LOW);
//右後輪の回転する方向を時計回りに決定する.
  analogWrite(BLVPin,V+23);

```

```
//左後輪の回転する速度を入力値+23に決定する。  
digitalWrite(BLDPin,HIGH);  
左後輪の回転する方向を反時計回りに決定する。  
}
```

```
//右方向に進むプログラム  
void right(int V){  
  analogWrite(FRVPin,V-8);  
  digitalWrite(FRDPin,HIGH);  
  analogWrite(FLVPin,V+25);  
  digitalWrite(FLDPin,HIGH);  
  analogWrite(BRVPin,V-5);  
  digitalWrite(BRDPin,LOW);  
  analogWrite(BLVPin,V+14);  
  digitalWrite(BLDPin,LOW);  
}
```

```
//左方向に進むプログラム  
void left(int V){  
  analogWrite(FRVPin,V-8);  
  digitalWrite(FRDPin,LOW);  
  analogWrite(FLVPin,V+25);  
  digitalWrite(FLDPin,LOW);  
  analogWrite(BRVPin,V-5);  
  digitalWrite(BRDPin,HIGH);  
  analogWrite(BLVPin,V+12);  
  digitalWrite(BLDPin,HIGH);  
}
```

```
//右上方向に進行するプログラム  
void upright(int V){  
  analogWrite(FRVPin,0);  
  digitalWrite(FRDPin,LOW);  
  analogWrite(FLVPin,V+25);  
  digitalWrite(FLDPin,HIGH);  
  analogWrite(BRVPin,V-5);  
  digitalWrite(BRDPin,LOW);  
  analogWrite(BLVPin,0);  
  digitalWrite(BLDPin,HIGH);  
}
```

```
//左上方向に進行するプログラム  
void upleft(int V){  
  analogWrite(FRVPin,V-8);  
  digitalWrite(FRDPin,LOW);  
  analogWrite(FLVPin,0);  
  digitalWrite(FLDPin,LOW);  
  analogWrite(BRVPin,0);  
  digitalWrite(BRDPin,LOW);  
  analogWrite(BLVPin,V+12);  
  digitalWrite(BLDPin,HIGH);  
}
```

```
//後退するプログラム  
void buck(int V){
```

```

analogWrite(FRVPin,V-8);
digitalWrite(FRDPin,HIGH);
analogWrite(FLVPin,V+28);
digitalWrite(FLDPin,LOW);
analogWrite(BRVPin,V-5);
digitalWrite(BRDPin,HIGH);
analogWrite(BLVPin,V+23);
digitalWrite(BLDPin,LOW);
}

```

//進行方向, 速度を決定する値に応じて 0.5 秒ごとに回転方向と回転速度の数値を各タイヤに入力するプログラム.
以降 move(int 型変数 D,int 型変数 V)と記述することで呼び出すことできる.

```

void move(int D,int V){
  if(D==1){
    front(V);
    //D が 1 という数値であった場合前進するプログラムを呼び出す.
  }else if(D==2){
    upright(V);
    //D が 2 という数値であった場合右上方向に進むプログラムを呼び出す.
  }else if(D==3){
    upleft(V);
    //D が 3 という数値であった場合左上方向に進むプログラムを呼び出す.
  }else if(D==4){
    right(V);
    //D が 4 という数値であった場合右方向に進むプログラムを呼び出す.
  }else if(D==5){
    left(V);
    //D が 5 という数値であった場合左方向に進むプログラムを呼び出す.
  }else if(D==6){
    buck(V);
    //D が 6 という数値であった場合後退するプログラムを呼び出す.
  }
  delay(500);
  //0.5 秒待つ
}

```

```

void loop() {
  //各秒数での進行方向を決定する
  D1= 4; D2= 4; D3= 4; D4= 4; D5= 4;
  D6= 4; D7= 4; D8= 2; D9= 2; D10= 2;
  D11= 4; D12= 4; D13= 4; D14= 4; D15= 4;
  D16= 4; D17= 4; D18= 4; D19= 2; D20= 2;
  D21= 2; D22= 2; D23= 2; D24= 2; D25= 1;
  D26= 1; D27= 4; D28= 4; D29= 4; D30= 4;
  D31= 4; D32= 4; D33= 4; D34= 4; D35= 4;
  D36= 4; D37= 4; D38= 4; D39= 5; D40= 5;
  D41= 5; D42= 5; D43= 5; D44= 2; D45= 2;
  D46= 2; D47= 2; D48= 2; D49= 1; D50= 1;
  D51= 1; D52= 2; D53= 2; D54= 2; D55= 2;
  D56= 2; D57= 2; D58= 2; D59= 3; D60= 3;
  D61= 5; D62= 5; D63= 2; D64= 2; D65= 2;
  D66= 2; D67= 4; D68= 4; D69= 4; D70= 4;
  D71= 3; D72= 5; D73= 5; D74= 3; D75= 1;
  D76= 1; D77= 1; D78= 1; D79= 1; D80= 1;
}

```


D81= 4; D82= 4; D83= 4; D84= 4; D85= 5;
D86= 5; D87= 5; D88= 5; D89= 5; D90= 5;
D91= 5; D92= 5; D93= 5; D94= 5; D95= 5;
D96= 5; D97= 5; D98= 5; D99= 5; D100= 5;
D101= 2; D102= 2; D103= 2; D104= 2; D105= 2;
D106= 2; D107= 1; D108= 1; D109= 3; D110= 3;
D111= 3; D112= 6; D113= 6; D114= 6; D115= 6;
D116= 6; D117= 6; D118= 6; D119= 6; D120= 6;
D121= 6;

//各秒数でのタイヤに対する入力値を決定する

V1= 27; V2= 29; V3= 27; V4= 27; V5= 29;
V6= 30; V7= 32; V8= 32; V9= 34; V10= 34;
V11= 29; V12= 27; V13= 27; V14= 29; V15= 30;
V16= 30; V17= 31; V18= 30; V19= 32; V20= 35;
V21= 35; V22= 32; V23= 29; V24= 34; V25= 40;
V26= 38; V27= 32; V28= 31; V29= 29; V30= 29;
V31= 31; V32= 34; V33= 29; V34= 28; V35= 25;
V36= 24; V37= 28; V38= 28; V39= 28; V40= 28;
V41= 28; V42= 27; V43= 27; V44= 27; V45= 27;
V46= 27; V47= 34; V48= 38; V49= 42; V50= 40;
V51= 32; V52= 29; V53= 31; V54= 35; V55= 33;
V56= 35; V57= 29; V58= 29; V59= 41; V60= 41;
V61= 34; V62= 34; V63= 41; V64= 58; V65= 40;
V66= 38; V67= 38; V68= 34; V69= 34; V70= 40;
V71= 41; V72= 33; V73= 40; V74= 42; V75= 56;
V76= 72; V77= 60; V78= 73; V79= 59; V80= 36;
V81= 38; V82= 33; V83= 30; V84= 33; V85= 29;
V86= 29; V87= 29; V88= 29; V89= 29; V90= 30;
V91= 33; V92= 29; V93= 29; V94= 29; V95= 29;
V96= 29; V97= 29; V98= 29; V99= 33; V100= 40;
V101= 42; V102= 41; V103= 41; V104= 41; V105= 43;
V106= 41; V107= 41; V108= 29; V109= 29; V110= 30;
V111= 41; V112= 31; V113= 29; V114= 29; V115= 29;
V116= 29; V117= 29; V118= 29; V119= 29; V120= 29;
V121= 29;

move(D1,V1);
move(D2,V2);
move(D3,V3);
move(D4,V4);
move(D5,V5);
move(D6,V6);
move(D7,V7);
move(D8,V8);
move(D9,V9);
move(D10,V10);
move(D11,V11);
move(D12,V12);
move(D13,V13);
move(D14,V14);
move(D15,V15);
move(D16,V16);
move(D17,V17);
move(D18,V18);

move(D19,V19);
move(D20,V20);
move(D21,V21);
move(D22,V22);
move(D23,V23);
move(D24,V24);
move(D25,V25);
move(D26,V26);
move(D27,V27);
move(D28,V28);
move(D29,V29);
move(D30,V30);
move(D31,V31);
move(D32,V32);
move(D33,V33);
move(D34,V34);
move(D35,V35);
move(D36,V36);
move(D37,V37);
move(D38,V38);
move(D39,V39);
move(D40,V40);
move(D41,V41);
move(D42,V42);
move(D43,V43);
move(D44,V44);
move(D45,V45);
move(D46,V46);
move(D47,V47);
move(D48,V48);
move(D49,V49);
move(D50,V50);
move(D51,V51);
move(D52,V52);
move(D53,V53);
move(D54,V54);
move(D55,V55);
move(D56,V56);
move(D57,V57);
move(D58,V58);
move(D59,V59);
move(D60,V60);
move(D61,V61);
move(D62,V62);
move(D63,V63);
move(D64,V64);
move(D65,V65);
move(D66,V66);
move(D67,V67);
move(D68,V68);
move(D69,V69);
move(D70,V70);
move(D71,V71);
move(D72,V72);
move(D73,V73);

```
move(D74,V74);
move(D75,V75);
move(D76,V76);
move(D77,V77);
move(D78,V78);
move(D79,V79);
move(D80,V80);
move(D81,V81);
move(D82,V82);
move(D83,V83);
move(D84,V84);
move(D85,V85);
move(D86,V86);
move(D87,V87);
move(D88,V88);
move(D89,V89);
move(D90,V90);
move(D91,V91);
move(D92,V92);
move(D93,V93);
move(D94,V94);
move(D95,V95);
move(D96,V96);
move(D97,V97);
move(D98,V98);
move(D99,V99);
move(D100,V100);
move(D101,V101);
move(D102,V102);
move(D103,V103);
move(D104,V104);
move(D105,V105);
move(D106,V106);
move(D107,V107);
move(D108,V108);
move(D109,V109);
move(D110,V110);
move(D111,V111);
move(D112,V112);
move(D113,V113);
move(D114,V114);
move(D115,V115);
move(D116,V116);
move(D117,V117);
move(D118,V118);
move(D119,V119);
move(D120,V120);
move(D121,V121);
}
```

サーボモータは魚が発揮する振動のような張力を再現する為、巻き取りと繰り出しを 0.15 秒間隔で繰り返す。サーボモータを制御する **Arduino** には次のようなプログラムを書き込む。このように動作させる理由は、大浜氏の助言で実際の魚は短時間で細かく振動が伝わってくる。というのを再現する為である。

```
#include <Servo.h>
```

```
//サーボモータのライブラリを読み込む
```

```
Servo servo;
```

```
const int Sv_Pin = 5;
```

```
//サーボモータの回転速度を制御する PIN 番号
```

```
//サーボモータに入力する変数を各 0.5 秒で 60 秒分宣言する。
```

```
int SV1,SV2,SV3,SV4,SV5,SV6,SV7,SV8,SV9,SV10;
```

```
int SV11,SV12,SV13,SV14,SV15,SV16,SV17,SV18,SV19,SV20;
```

```
int SV21,SV22,SV23,SV24,SV25,SV26,SV27,SV28,SV29,SV30;
```

```
int SV31,SV32,SV33,SV34,SV35,SV36,SV37,SV38,SV39,SV40;
```

```
int SV41,SV42,SV43,SV44,SV45,SV46,SV47,SV48,SV49,SV50;
```

```
int SV51,SV52,SV53,SV54,SV55,SV56,SV57,SV58,SV59,SV60;
```

```
int SV61,SV62,SV63,SV64,SV65,SV66,SV67,SV68,SV69,SV70;
```

```
int SV71,SV72,SV73,SV74,SV75,SV76,SV77,SV78,SV79,SV80;
```

```
int SV81,SV82,SV83,SV84,SV85,SV86,SV87,SV88,SV89,SV90;
```

```
int SV91,SV92,SV93,SV94,SV95,SV96,SV97,SV98,SV99,SV100;
```

```
int SV101,SV102,SV103,SV104,SV105,SV106,SV107,SV108,SV109,SV110;
```

```
int SV111,SV112,SV113,SV114,SV115,SV116,SV117,SV118,SV119,SV120
```

```
int SV121;
```

```
void setup(){
```

```
servo.attach(Sv_Pin,500,2400);
```

```
}
```

//入力値に応じた速度で繰り出しと巻き取りを 0.15 秒間隔で繰り返すプログラム。以降 mv と記述することでこのプログラムを呼び出すことが出来る。

```
void mv(int SV){
```

```
servo.write(SV);
```

```
//サーボモータを SV の角度まで回転させる
```

```
delay(150);
```

```
//0.15 秒待つ
```

```
servo.write(SV-25);
```

```
//サーボモータを SV-25 の角度まで回転させる
```

```
delay(150);
```

```
servo.write(SV);
```

```
delay(150);
```

```
servo.write(SV-25);
```

```
delay(150);
```

```
}
```

```
void loop() {
```

```
//各秒数でサーボモータをどの角度まで回転させるか決定する
```

```
SV1=95; SV2=95; SV3=102; SV4=102;
```

```
SV5=102; SV6=102; SV7=102; SV8=102;
```

```
SV9=104; SV10=104; SV11=102; SV12=102;
```

```
SV13=102; SV14=102; SV15=102; SV16=102;
```

```
SV17=102; SV18=102; SV19=102; SV20=102;
```

```
SV21=102; SV22=102; SV23=102; SV24=102;
```

```
SV25=103; SV26=103; SV27=102; SV28=102;
```

```
SV29=102; SV30=102; SV31=102; SV32=102;
```

```
SV33=102; SV34=95; SV35=95; SV36=95;
```

```
SV37=95; SV38=95; SV39=95; SV40=95;
```

```
SV41=95; SV42=102; SV43=102; SV44=102;
```

```
SV45=102; SV46=102; SV47=102; SV48=103;
```

SV49=106; SV50=105; SV51=102; SV52=102;
SV53=102; SV54=102; SV55=104; SV56=102;
SV57=102; SV58=102; SV59=105; SV60=106;
SV61=102; SV62=102; SV63=108; SV64=110;
SV65=103; SV66=102; SV67=102; SV68=102;
SV69=103; SV70=103; SV71=102; SV72=102;
SV73=102; SV74=107; SV75=114; SV76=110;
SV77=109; SV78=120; SV79=106; SV80=109;
SV81=102; SV82=102; SV83=102; SV84=102;
SV85=102; SV86=102; SV87=102; SV88=102;
SV89=102; SV90=102; SV91=102; SV92=102;
SV93=102; SV94=102; SV95=102; SV96=102;
SV97=102; SV98=102; SV99=102; SV100=103;
SV101=103; SV102=108; SV103=106; SV104=108;
SV105=106; SV106=108; SV107=105; SV108=102;
SV109=102; SV110=102; SV111=105; SV112=105;
SV113=102; SV114=102; SV115=102; SV116=102;
SV117=102; SV118=102; SV119=102; SV120=102;
SV121=102;

//mv を呼び出す

mv(SV1);
mv(SV2);
mv(SV3);
mv(SV4);
mv(SV5);
mv(SV6);
mv(SV7);
mv(SV8);
mv(SV9);
mv(SV10);
mv(SV11);
mv(SV12);
mv(SV13);
mv(SV14);
mv(SV15);
mv(SV16);
mv(SV17);
mv(SV18);
mv(SV19);
mv(SV20);
mv(SV21);
mv(SV22);
mv(SV23);
mv(SV24);
mv(SV25);
mv(SV26);
mv(SV27);
mv(SV28);
mv(SV29);
mv(SV30);
mv(SV31);
mv(SV32);
mv(SV33);
mv(SV34);

mv(SV35);
mv(SV36);
mv(SV37);
mv(SV38);
mv(SV39);
mv(SV40);
mv(SV41);
mv(SV42);
mv(SV43);
mv(SV44);
mv(SV45);
mv(SV46);
mv(SV47);
mv(SV48);
mv(SV49);
mv(SV50);
mv(SV51);
mv(SV52);
mv(SV53);
mv(SV54);
mv(SV55);
mv(SV56);
mv(SV57);
mv(SV58);
mv(SV59);
mv(SV60);
mv(SV61);
mv(SV62);
mv(SV63);
mv(SV64);
mv(SV65);
mv(SV66);
mv(SV67);
mv(SV68);
mv(SV69);
mv(SV70);
mv(SV71);
mv(SV72);
mv(SV73);
mv(SV74);
mv(SV75);
mv(SV76);
mv(SV77);
mv(SV78);
mv(SV79);
mv(SV80);
mv(SV81);
mv(SV82);
mv(SV83);
mv(SV84);
mv(SV85);
mv(SV86);
mv(SV87);
mv(SV88);
mv(SV89);

```
mv(SV90);  
mv(SV91);  
mv(SV92);  
mv(SV93);  
mv(SV94);  
mv(SV95);  
mv(SV96);  
mv(SV97);  
mv(SV98);  
mv(SV99);  
mv(SV100);  
mv(SV101);  
mv(SV102);  
mv(SV103);  
mv(SV104);  
mv(SV105);  
mv(SV106);  
mv(SV107);  
mv(SV108);  
mv(SV109);  
mv(SV110);  
mv(SV111);  
mv(SV112);  
mv(SV113);  
mv(SV114);  
mv(SV115);  
mv(SV116);  
mv(SV117);  
mv(SV118);  
mv(SV119);  
mv(SV120);  
mv(SV121);  
}
```

6. 実験概要

シミュレータの張力を測定器で測定する実験、釣り経験者にシミュレータを体験して貰いアンケートによる評価をする実験を行った。それらの実験方法について述べる。

6.1 シミュレータの張力を測定する方法

測定器を約 55cm の高さの台の上に固定する。測定器と作成したシミュレータとを釣り糸で繋ぎ、シミュレータを動作させることで張力を測定する。測定器とシミュレータとは約 100cm 離してシミュレータの動作を開始する。

6.2 シミュレータの体験者による評価方法

体験者に釣り竿を持って貰い、シミュレータと釣り竿とを釣り糸で繋ぐ。マスノスケの張力を再現したシミュレータであると伝え、シミュレータを動作させて張力を感じて貰う。その後アンケートを取る。アンケート内容は以下の通りである。実際の魚にどれだけ近いか。動く速度は実際の魚にどれだけ近いか。水平方向に引っ張られる感覚は実際の魚にどれだけ近いか。垂直方向に引っ張られる感覚は実際の魚にどれだけ近いかの 4 点である。それぞれ 5 点満点で、点数が高い程実際の魚に近いという評価である。

7 実験結果と考察

6 節での 2 つの実験結果、考察を述べる。

7.1 測定器での実験結果

実験結果として図 10 のような結果を得た。

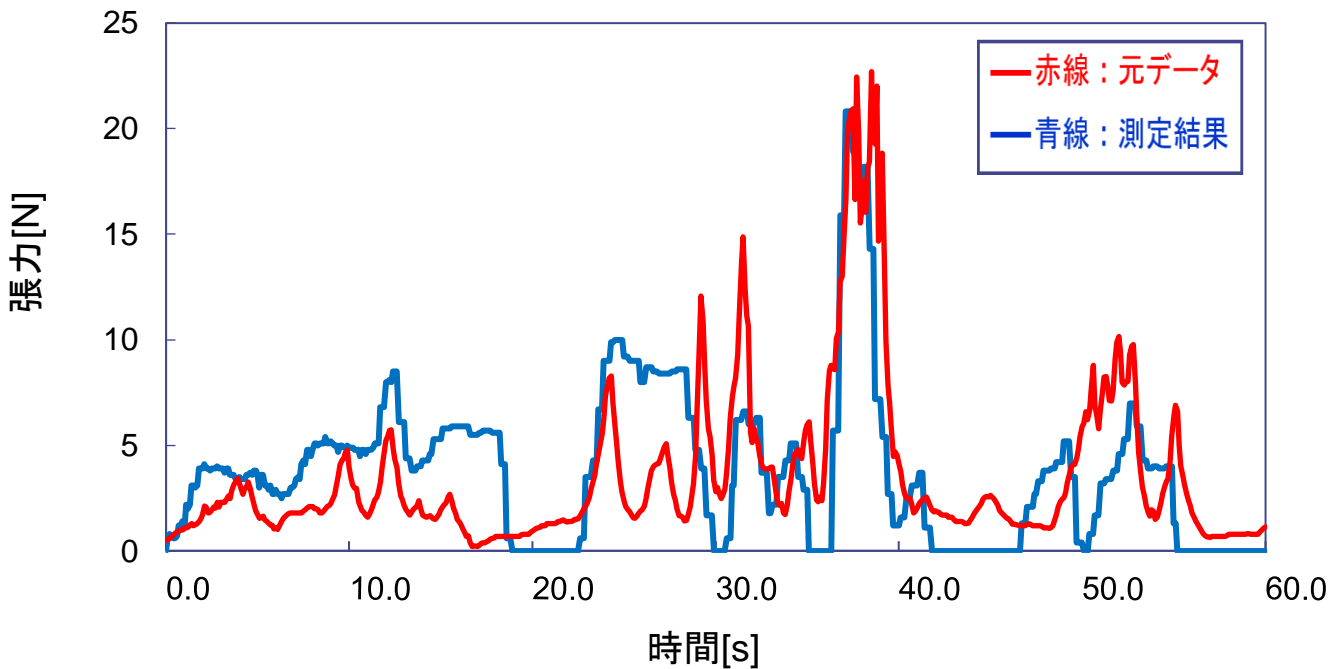


図 10 測定結果

測定結果を見ると、微弱な張力を再現することが出来ていない。細かな張力の変化を再現する為、サーボモータを巻き取りと繰り出しを繰り返すよう設計したが、元データ程の細かな張力変化を再現できていない。固定した測定器に糸が引っ張られる事でシミュレータの向きが変わってしまい、測定器に向かって動く時間があり、その間は糸がたるんでしまい張力が 0N となってしまった。一番大きな山、山が出るタイミングに関しては、おおむね一致していると言える。最後の 5 秒に関しては糸がたるむ方向に進むようプログラムしたため、0N という結果になっている。

7.2 シミュレータの体験者によるアンケート結果

マスノスケを釣獲したことがある人 2 人、マスノスケを釣獲したことがない人 5 人の計 7 人の釣り経験者にアンケートを行った。

実際の魚にどれだけ近いかのアンケート結果を図 11 に、動く速度は魚にどれだけ近いかの結果を図 9 に、水平方向の感覚は魚にどれだけ近いかの結果を図 10 に、垂直方向の感覚は魚にどれだけ近いかの結果を図 11 にそれぞれ示す。各点数の左側の縦棒がマスノスケを釣ったことがある人数を示し、右側の縦棒がマスノスケを釣ったことがない人の点数を示している。

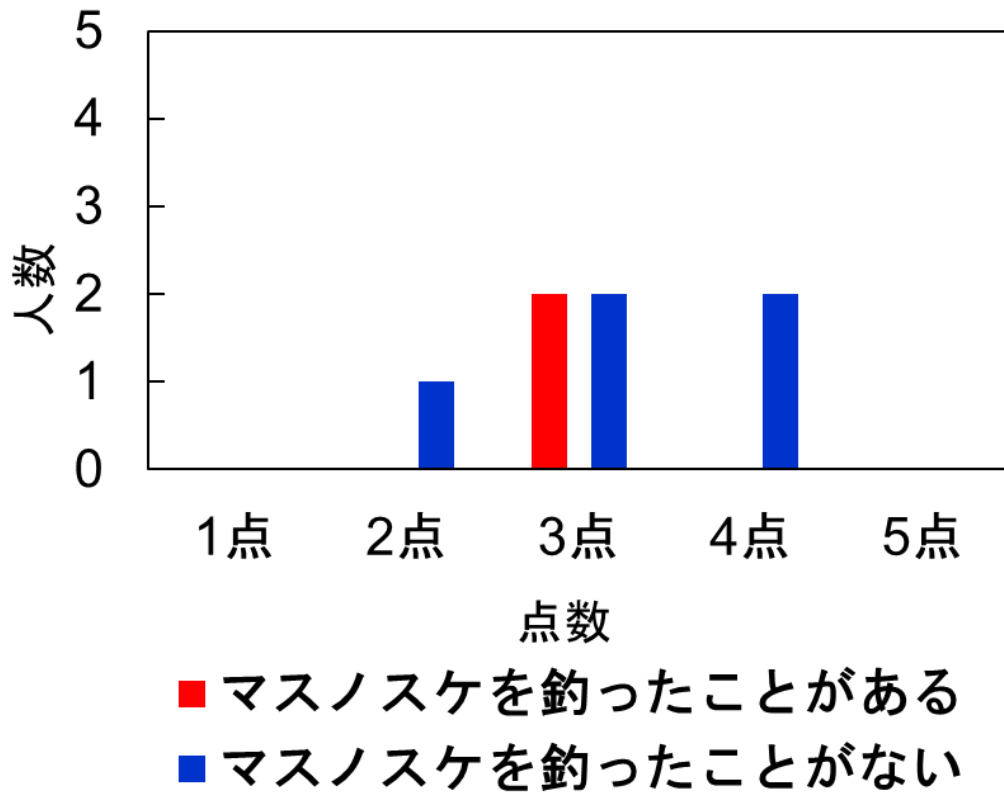


図 11 実際の魚にどれだけ近いかの結果

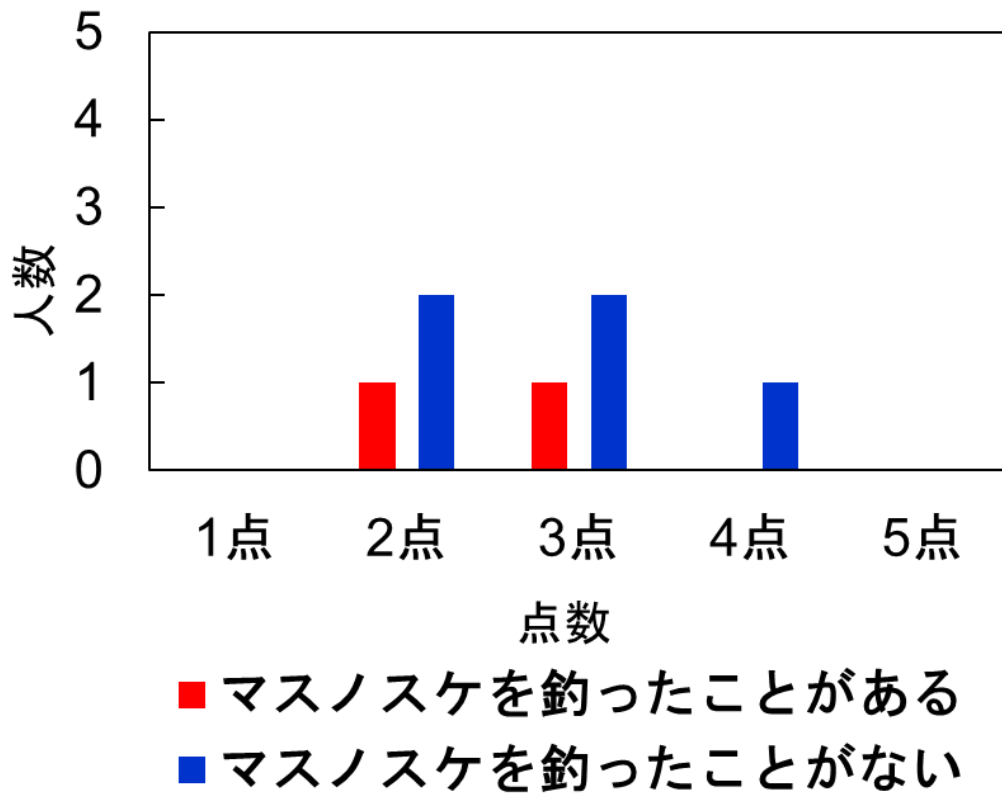


図 12 動く速度は実際の魚にどれだけ近いかの結果

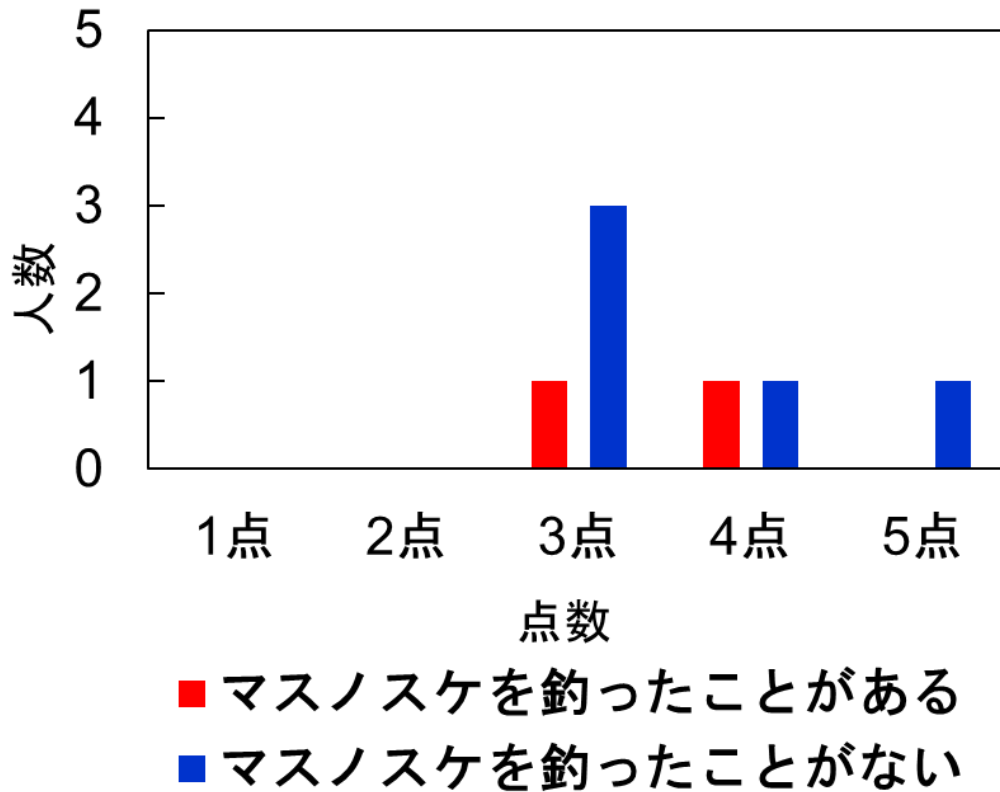


図 13 水平方向の感覚は実際の魚にどれだけ近いかの結果

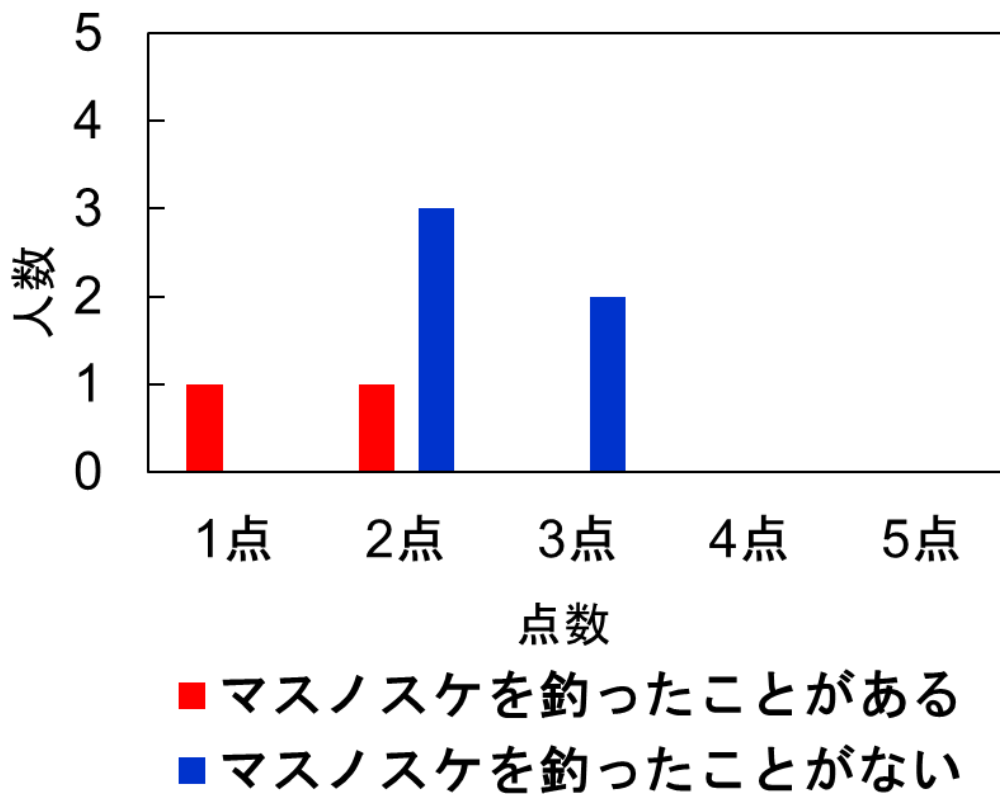


図 14 垂直方向の感覚は実際の魚にどれだけ近いかの結果

実際の魚に対してどれだけ近いかに関しては、平均値が約 3.14 点であり、基準となる 3 点を少し上回る評価を得ることが出来た。マスノスケを釣ったことがある人は 2 人とも 3 点であった。マスノスケを釣ったことがない人は 2 人に 1 人、3 点に 2 人 4 点に 2 人と点数にばらつきが生じてしまった。速度に関してはマスノスケよりも遅いという評価であった。他の魚では、ちょうどよいという意見もあった。平均値は約 2.71 点であり、基準となる 3 点を少し下回ってしまった。水平方向へ引っ張られる感覚に関しては、平均値が約 3.57 点であり、基準となる 3 点を超える評価を得ることができた。垂直方向へ引っ張られる感覚は実際の魚よりも弱いという評価であった。平均値は約 1.71 点と基準となる 3 点を大きく下回ってしまった。

7.3 考察

測定器での実験の際、固定した測定器にシミュレータが引っ張られてしまい、シミュレータの方向が変わってしまうことで糸がたるんで 0Nになってしまう時間があるという結果が得られたが、実際に人に体験して貰う際は、リールで釣り糸を巻き取る行動をするため糸はたるむことはなかった。そのため、人に体験して貰うシミュレータとして影響はないと判断する。

アンケート結果から、3 点を下回ってしまった速度の項目と垂直方向の項目に関しては改善が必要である。速度の項目の点数が低かった原因として 4 輪ロボットの張力変化のタイミングを 0.5 秒間隔にしてしまったことが考えられる。よって 0.5 秒間隔で張力変化をするのではなく、もっと短い感覚で張力を変化させる必要がある。垂直方向の感覚の点数が低かった原因としては、サーボモータを細かな振動を再現する為、巻き取りと繰り出しを繰り返すよう設計した為、垂直方向へ引っ張られる感覚が感じにくかったことが考えられる。よって魚の振動のような細かな張力を再現する別の方法を考える必要がある。

実際に体験して貰った人数が 7 人と少ない人数で終わってしまった為、もっと多くの釣り経験者の方に体験して貰い、評価してもらうことも課題点である

8 終わりに

本研究では、実際の魚の張力を測定したデータを元に垂直方向、水平方向の魚の張力を再現する実験を行った。結果としては、データに対して一致させる事は出来なかったが、実際に人に体験して貰った場合、魚に引っ張られている感覚を 5 点中 3 点程の精度で再現する事が出来ると分かった。また、本研究の目的であった水平方向の張力という事柄に関しては、基準点を 0.57 点を超える結果が得られた。今後の課題としては、速度を速くすること、垂直方向の張力をもっと強くすること、魚の振動のような細かな張力を正確に再現する事が挙げられる。

謝辞

本研究を進める為に貴重なデータを御提供して下さっただけでなく、製作に関しても多くの御助言をして下さった大浜秀規氏に改めて感謝を申し上げます。

参考文献

- [1] 日本生産性本部. レジャー白書 2022. 生産性出版, 2022, 38p, 63p.
- [2] 井嶋 駿太“電子回路を用いた釣り体験システムの製作と評価”, [18sc021.pdf \(nanzan-u.ac.jp\)](https://www.nanzan-u.ac.jp/~18sc021/). 2023 年 6 月参照.
- [3] 大浜秀規, “マスノスケ釣獲時の引き方に関する試験”. [Microsoft Word - 150130 マスノスケ釣力 \(pref.yamanashi.jp\)](https://www.pref.yamanashi.jp/~150130/). 2022 年 11 月参照.