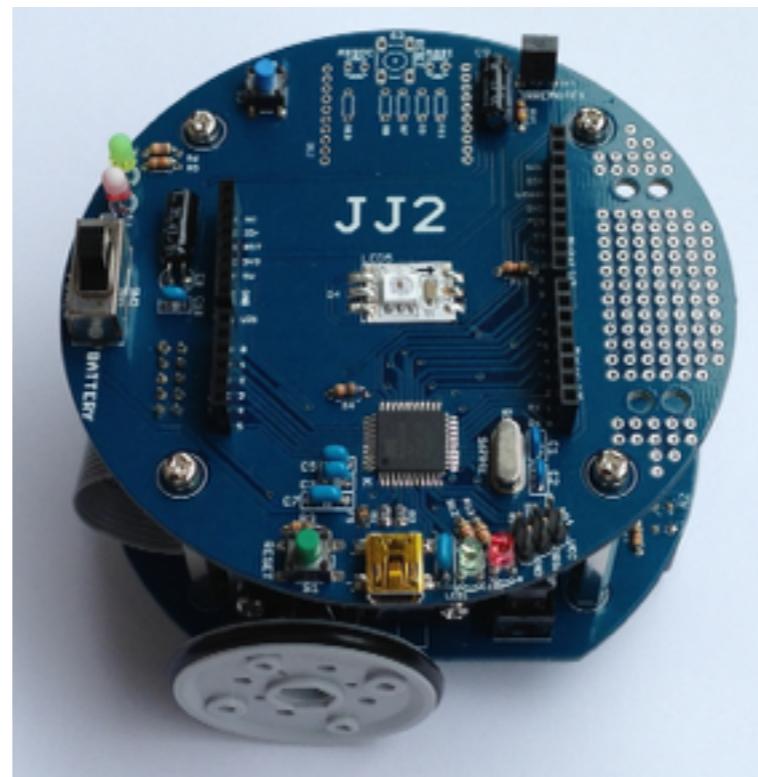


ロボットをつくろう (後編)

第9回 ロボットの走行実験1



<http://cms.db.tokushima-u.ac.jp/DAV/person/S10723/>

川上 博

2014/10/18

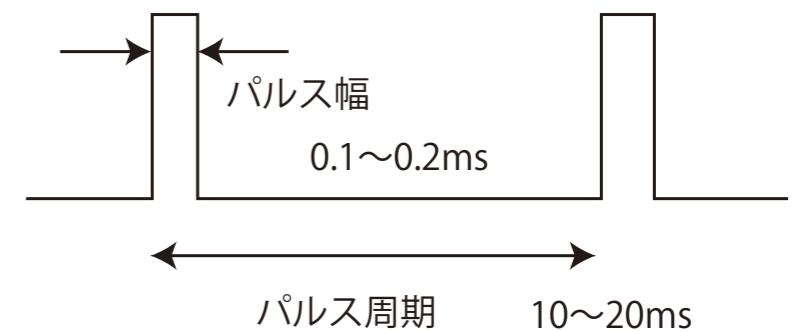
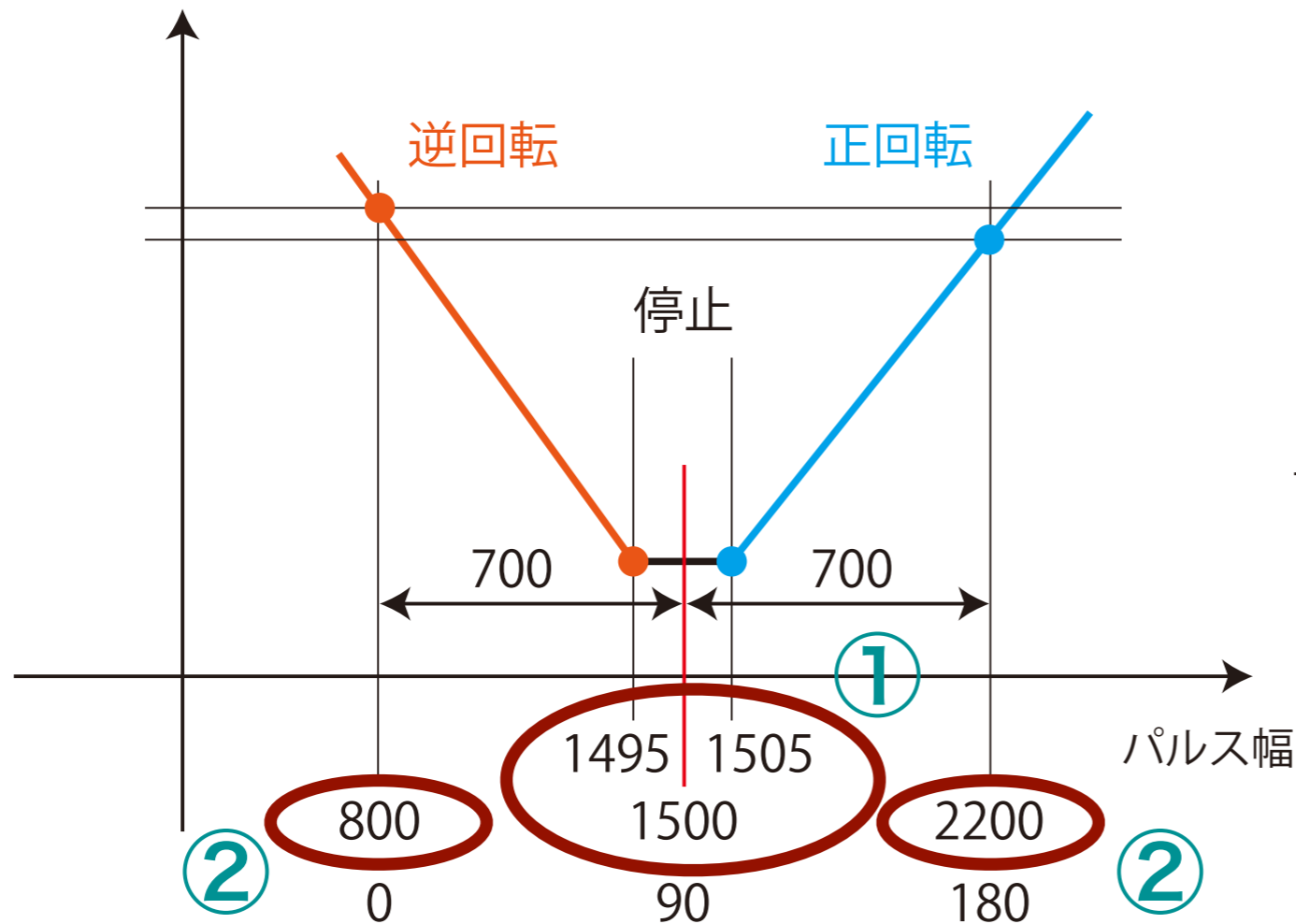
今日のテーマ

- ◎ 前回の復習：サーボモータのキャリブレーション
- ◎ ロボットの走行実験：前進，後退，左折，右折，停止
- ◎ Logo: turtle graphics

サーボモータのキャリブレーション（補正）

- ① モータが停止するパルス幅の最小値と最大値を測定し，中心値を求める

モータの回転速度



- ② 中心値を中心にパルス幅の変化範囲を決める

停止位置を測定するためのスケッチ：901A

```
#include <Servo.h>

const int servoL_Pin = 9;
const int servoR_Pin = 5;

Servo servoL; // left servo
Servo servoR; // right servo

void setup() {
  delay(1000);
  servoL.attach(servoL_Pin);
  // servoR.attach(servoR_Pin);
  Serial.begin(9600);
}

void loop() {
  for (int i=1400; i<1600; i++) {
    servoL.writeMicroseconds(i);
    // servoR.writeMicroseconds(i);
    Serial.println(i);
    delay(600);
  }
}
```

① サーボが停止するパルス幅
(最小値, 最大値) を記録する



② 左が測定できたら, 右を測定する



キャリブレーションのスケッチ：902A

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

```
const int servoL_Pin = 9; // uses timer1
```

```
const int servoR_Pin = 5; // uses timer1
```

```
Servo servoL; // left servo
```

```
Servo servoR; // right servo
```

```
int centerR=1500, centerL=1440, pw=700;
```



- ① サーボが停止するパルス幅（最小値，最大値）から中央値を求める

```
void setup() {
```

```
  delay(1000);
```

```
  servoL.attach(servoL_Pin, centerL-pw, centerL+pw); // D9
```

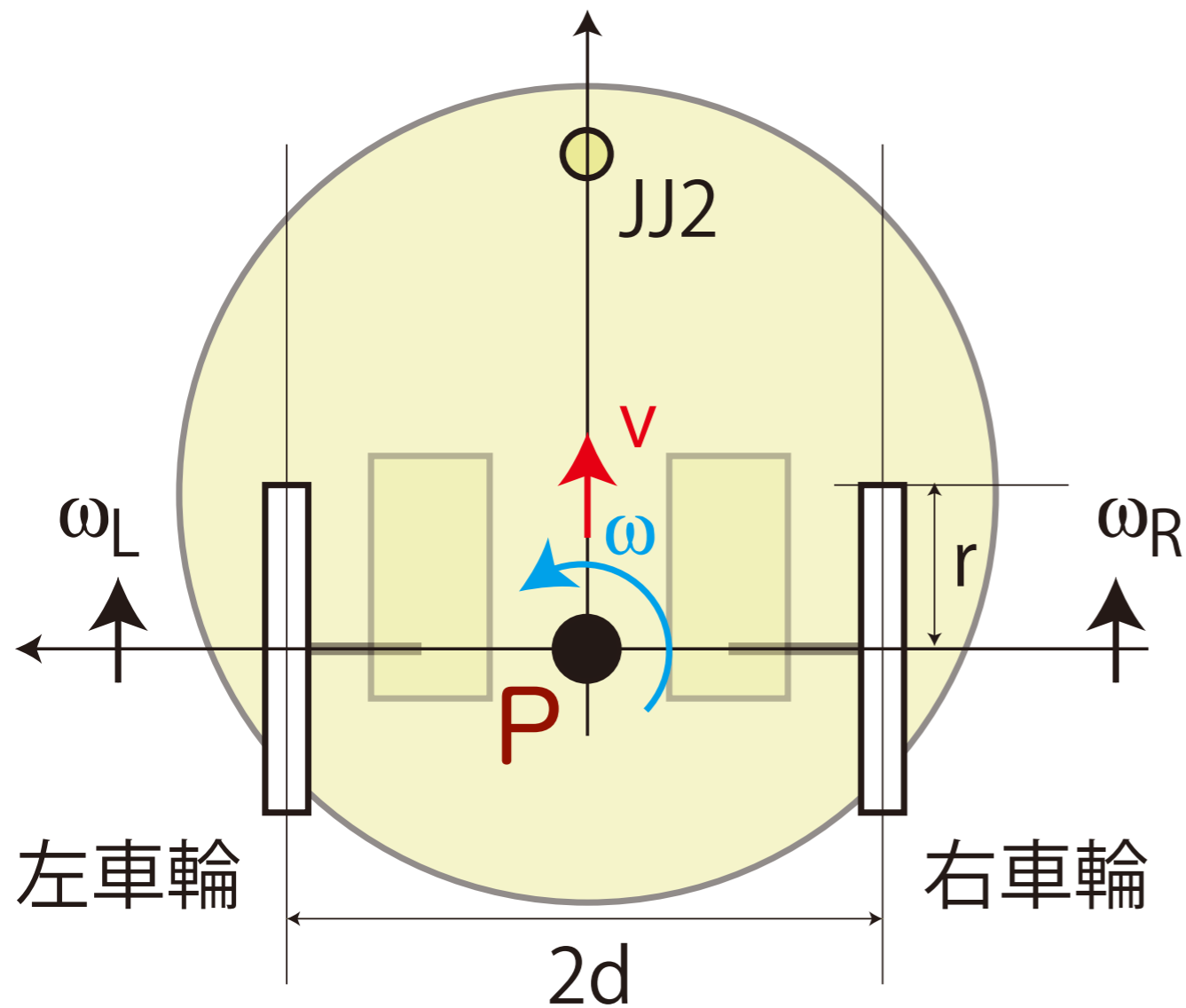
```
  servoR.attach(servoR_Pin, centerR-pw, centerR+pw); // D5
```

```
  // Serial.begin(9600); // open serial port by setting 9600 bps
```

```
}
```



ロボットの動き



ロボットの中心Pの速度：

$$v = \frac{r}{2}(\omega_R + \omega_L)$$

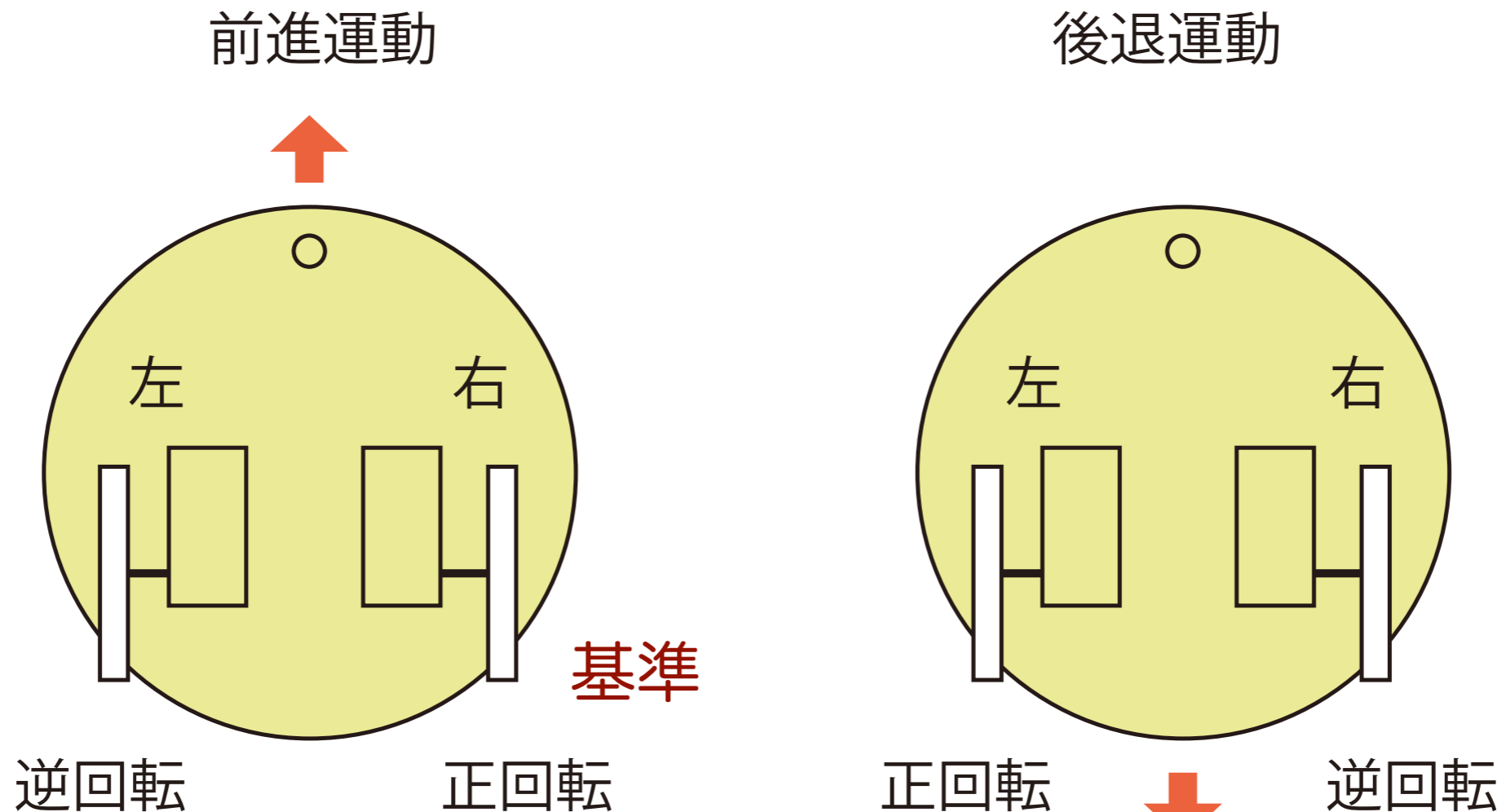
ロボットの旋回角速度：

$$\omega = \frac{r}{2d}(\omega_R - \omega_L)$$

ロボットの旋回半径：

$$\rho = d \frac{\omega_R + \omega_L}{\omega_R - \omega_L}$$

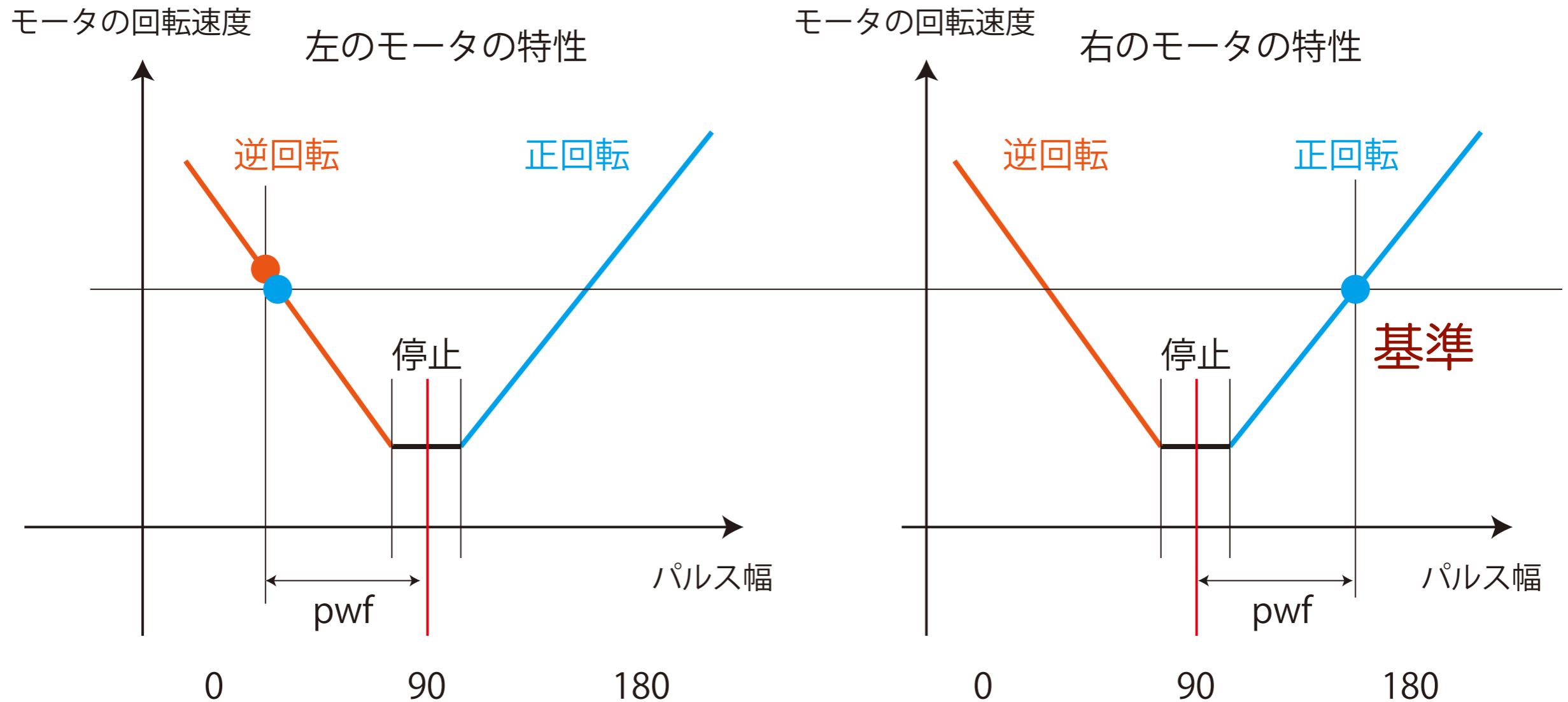
JJ2の運動：前進，後退



```
void fwd(int tau){
  servoL.write(32); //30
  servoR.write(150);
  delay(tau);
}
```

```
void bwd(int tau){
  servoL.write(150);
  servoR.write(27); //30
  delay(tau);
}
```

JJ2の運動：左折，右折

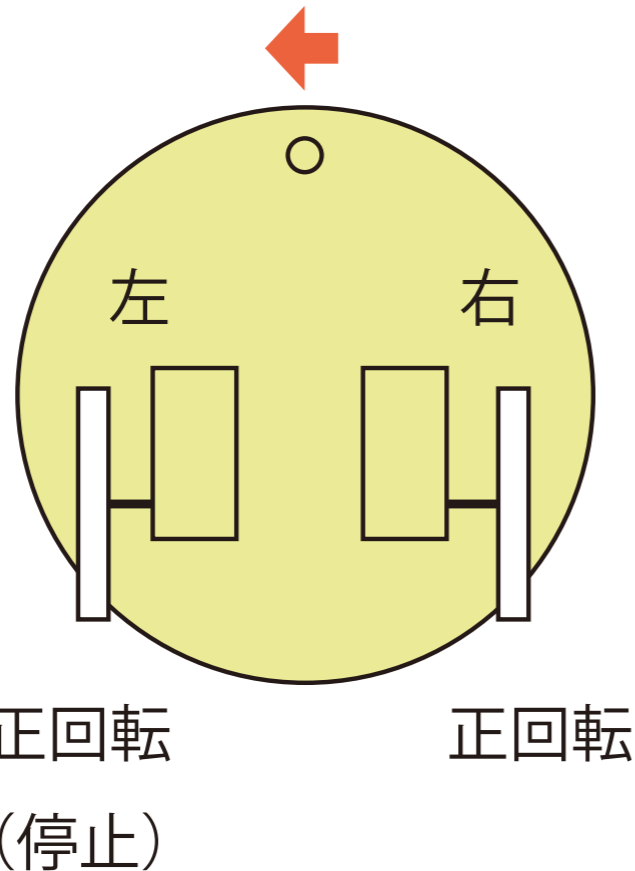


```
void ccw(int tau){
  servoL.write(150);
  servoR.write(150);
  delay(tau);
}
```



JJ2の運動：左折，右折

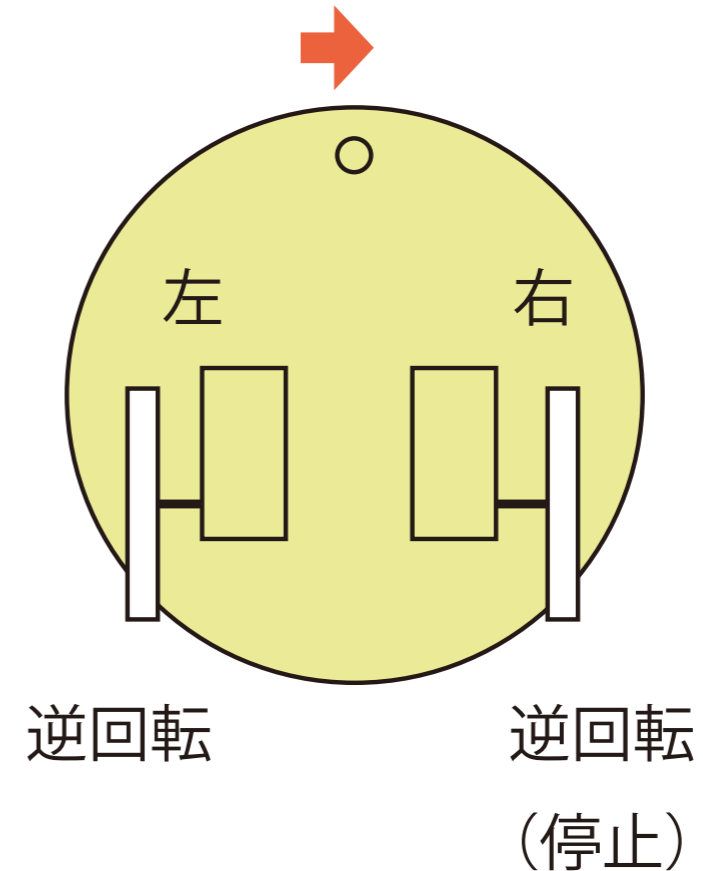
左折運動



```
void ccw(int tau){
  servoL.write(90);
  servoR.write(150);
  delay(tau); ←
}

```

右折運動



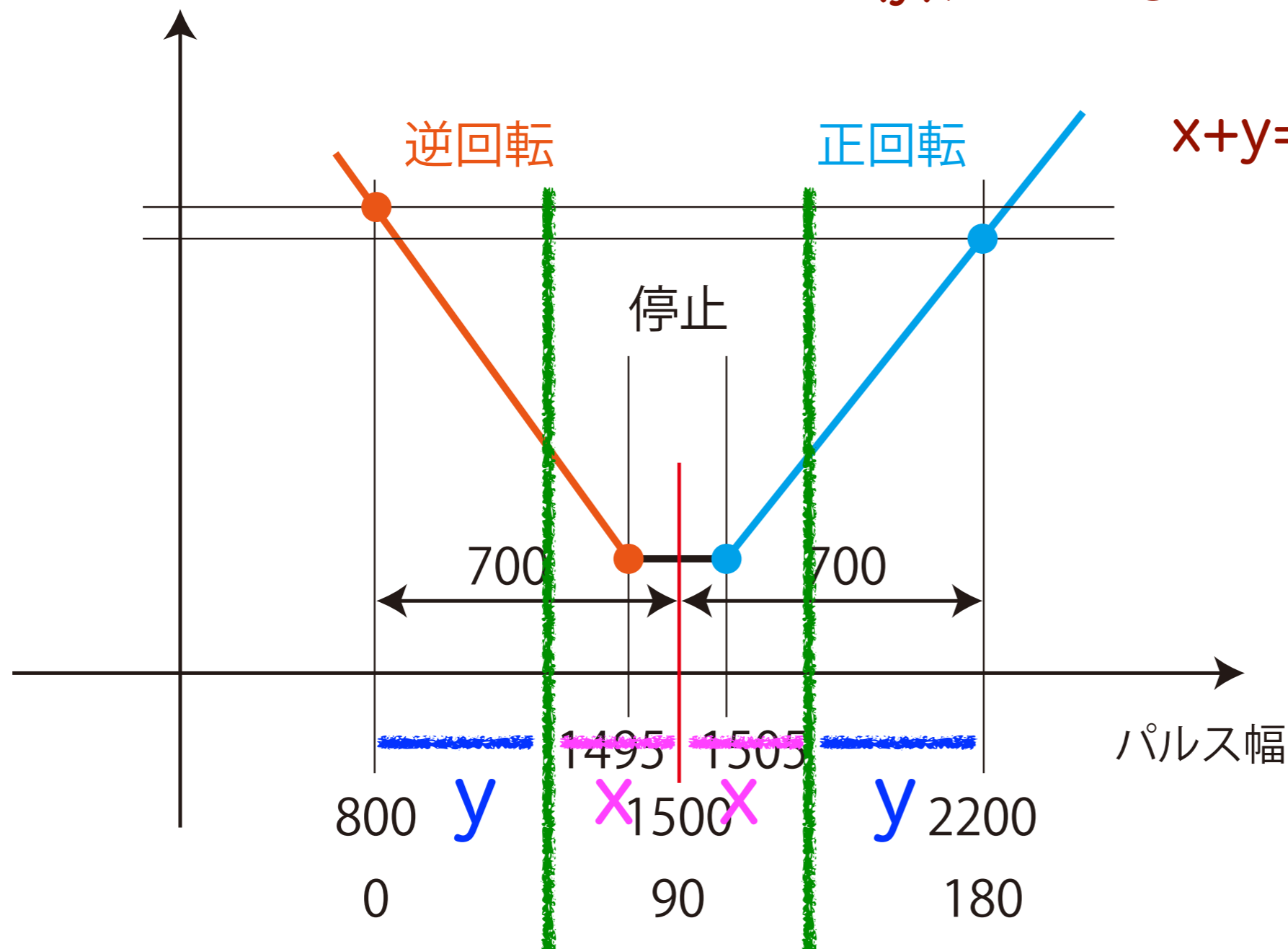
```
void cw(int tau){
  servoL.write(150);
  servoR.write(90);
  delay(tau); ←
}

```

servoR.writeの引数の書き方

- | | |
|------------------------------------|-------------------------------------|
| ① <code>servoR.write(90-x);</code> | ① <code>servoR.write(90+x);</code> |
| ② <code>servoR.write(y);</code> | ② <code>servoR.write(180-y);</code> |

モータの回転速度



$x+y=90$ の関係がある

速度 x 時間 = 距離

```
void fwd(int tau){  
    servoL.write(30);  
    servoR.write(150);  
    delay(tau);  
}
```

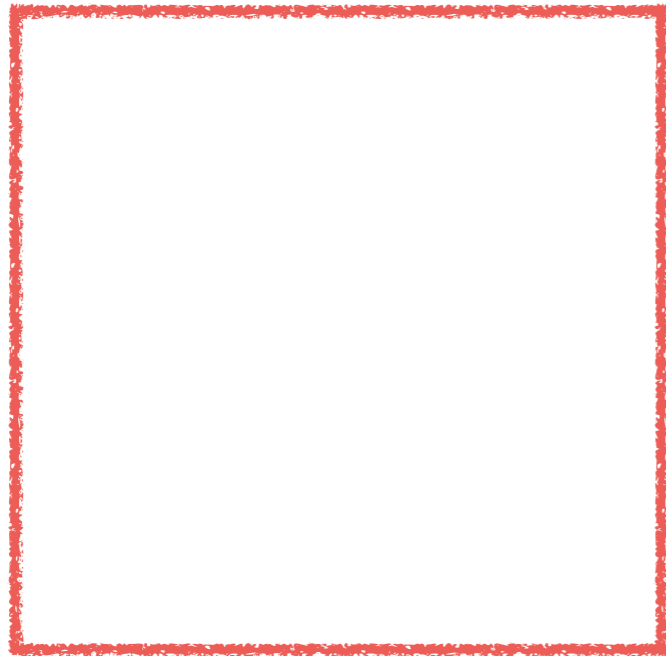
時間 tau を調節して距離を得る

```
void rot(int tau){  
    servoL.write(150);  
    servoR.write(150);  
    delay(tau);  
}
```

時間 tau を調節して回転角度を得る

Logo : turtle graphics

「10cm進んで90度回転する」を4回繰り返して描かれる図形



```
for(int i=0; i<4;i++){  
    fwd(tau1);  
    rot(tau2);  
}
```

turtle00 : Processingの動くパソコンで実行してください

<http://el.media.mit.edu/logo-foundation/index.html>