

# 自律走行システム特論（第2回）

## Micro Mechatronics (2<sup>nd</sup>)

### □ 自己位置推定（ホイールオドメトリ）

Self-Localization(WheelOdometry)

### □ 解説とプログラムによる実習

Theoreticalexplanations,and the practical training with programming

### □ 自己位置推定（IMU）

Self-Localization(InertialMeasurementUnit: IMU)

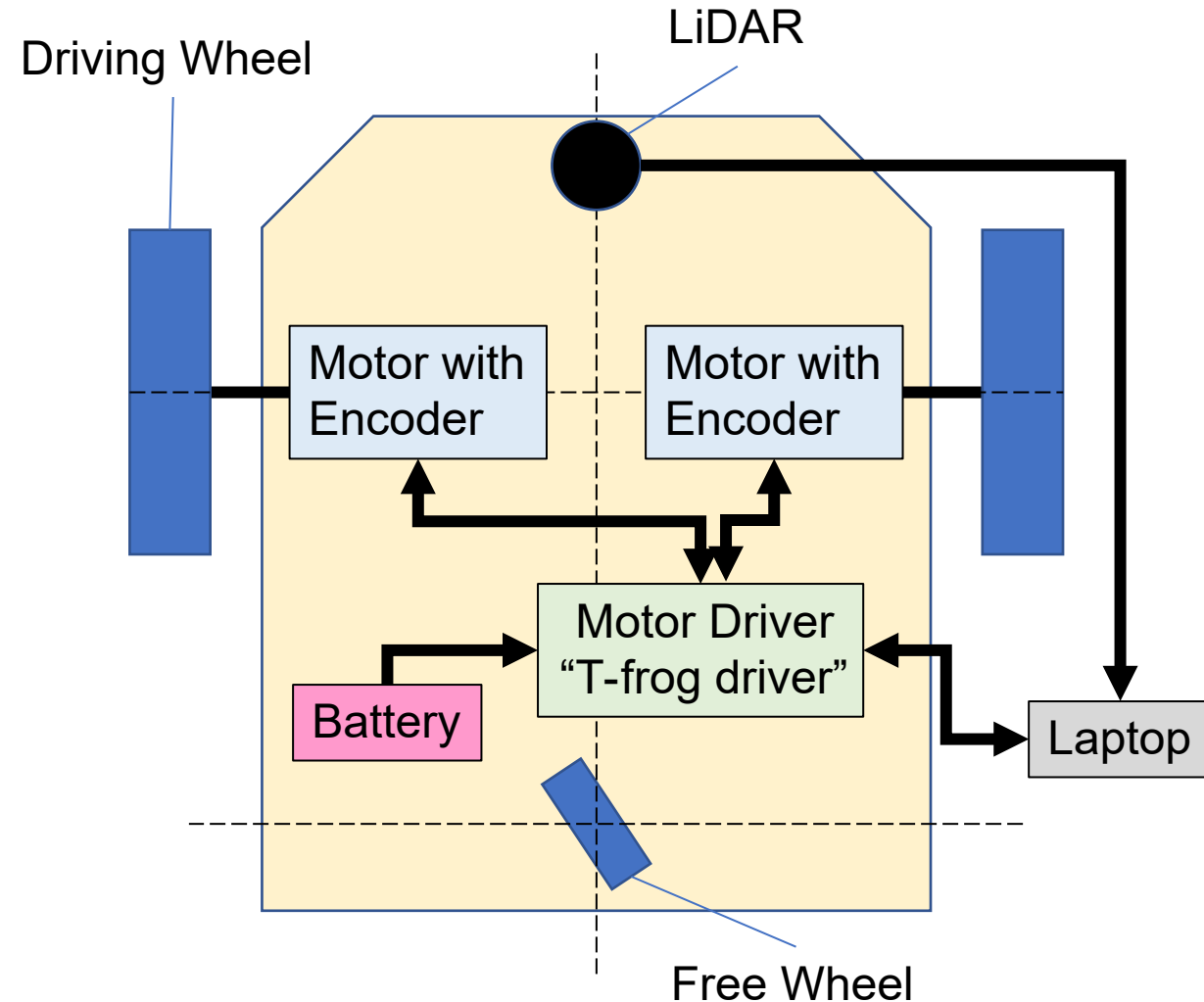
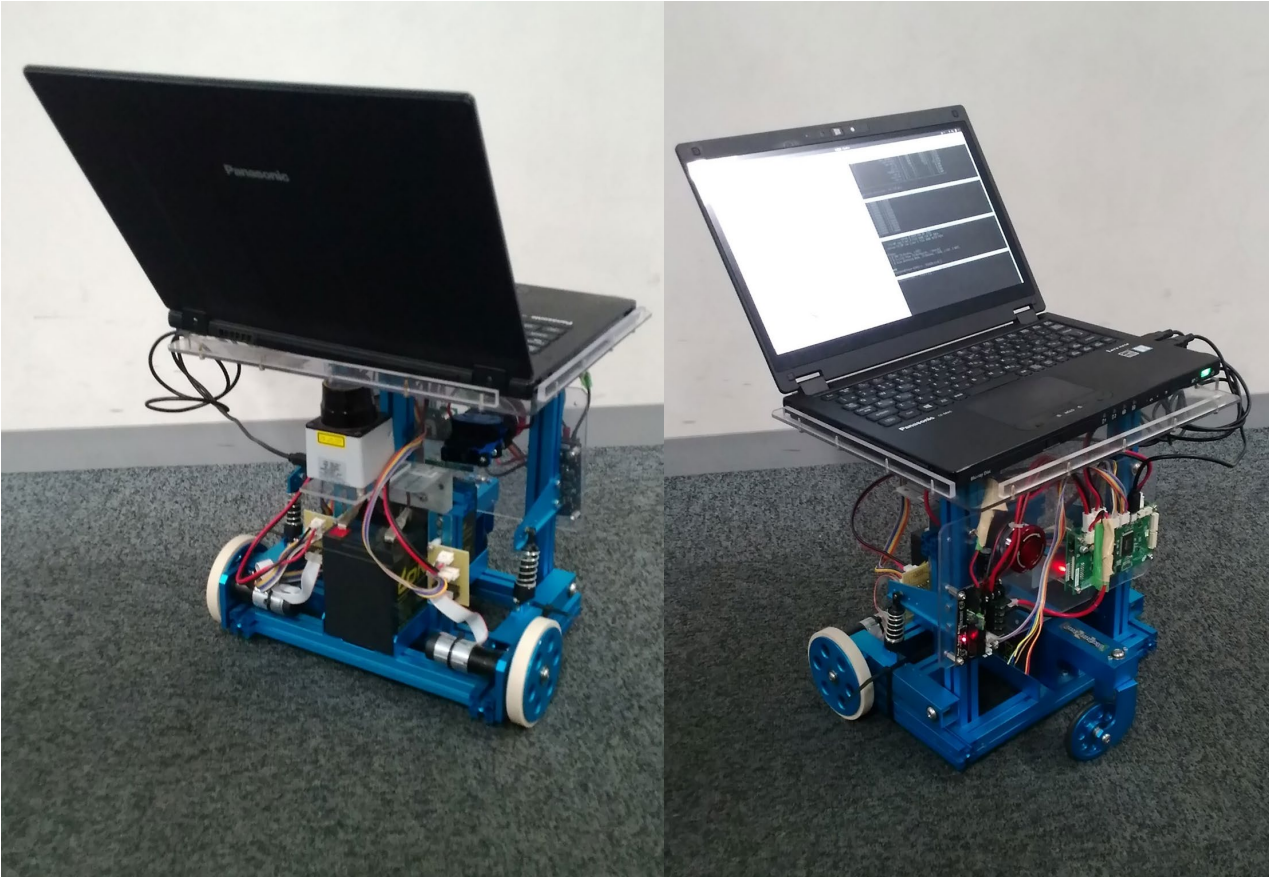
### □ 解説とプログラムによる実習

Theoreticalexplanations,and the practical training with programming.

### □ 課題の確認と考察（ホイールオドメトリ, IMU）

Confirmation and discussion for assignments (WheelOdometry, IMU)

# 教育用自律移動ロボット"Beego" Educational Mobile Robot "Beego"

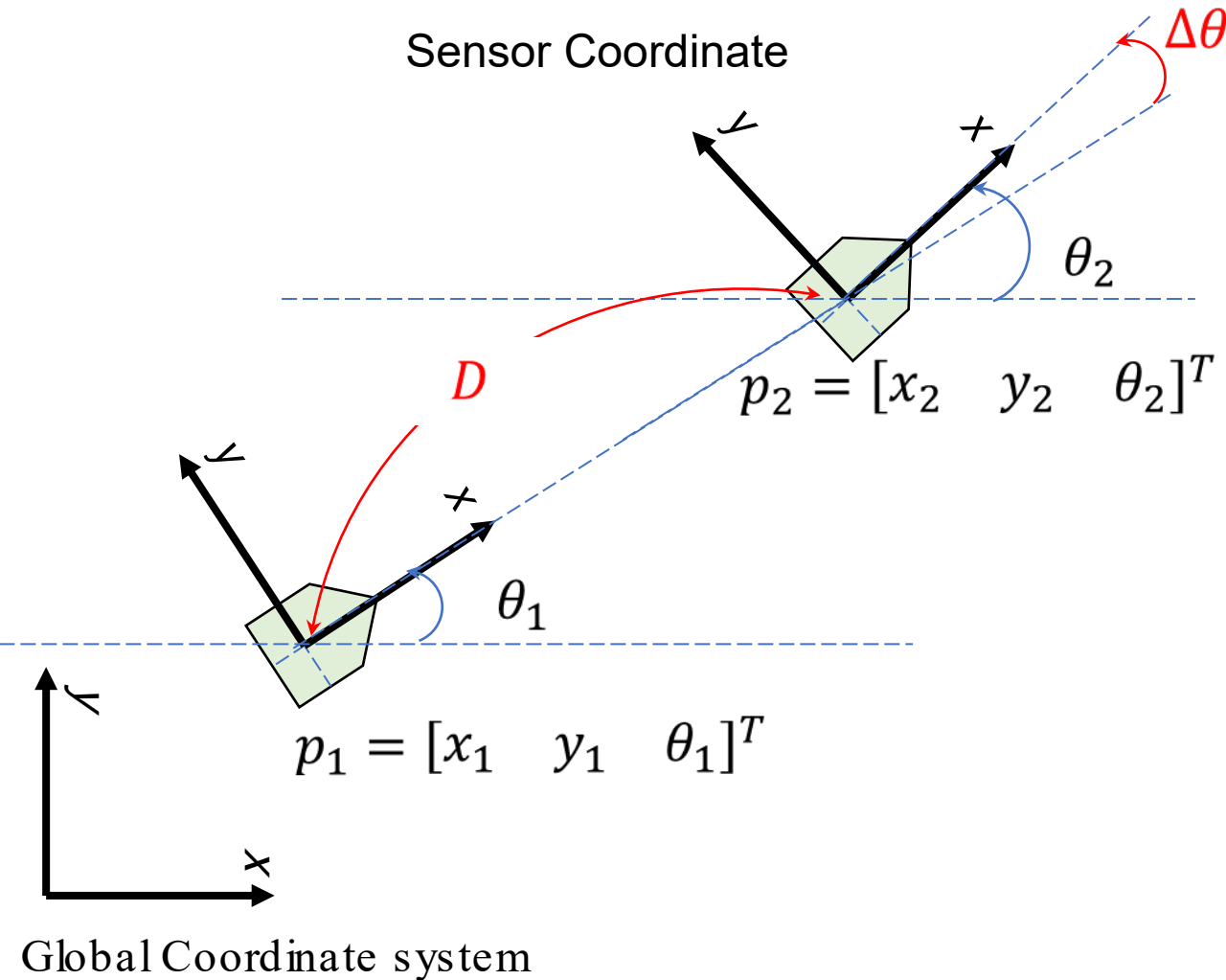


映像あり Movie available

Independent Drive Wheels Mechanism

# ホイールオドメトリの概要 #1

## Overview of Wheel Odometry #1



問1 :  $p_1$ を用いて $p_2$ を求めよ。  
Q1 : Simplify  $p_2$  by using  $p_1$

位置推定 Estimated Position

$$\begin{cases} x_2 = \\ y_2 = \\ \theta_2 \end{cases}$$



# ホイールオドメトリの概要 #2

## Overview of Wheel Odometry #2

移動量 Amount of Movement

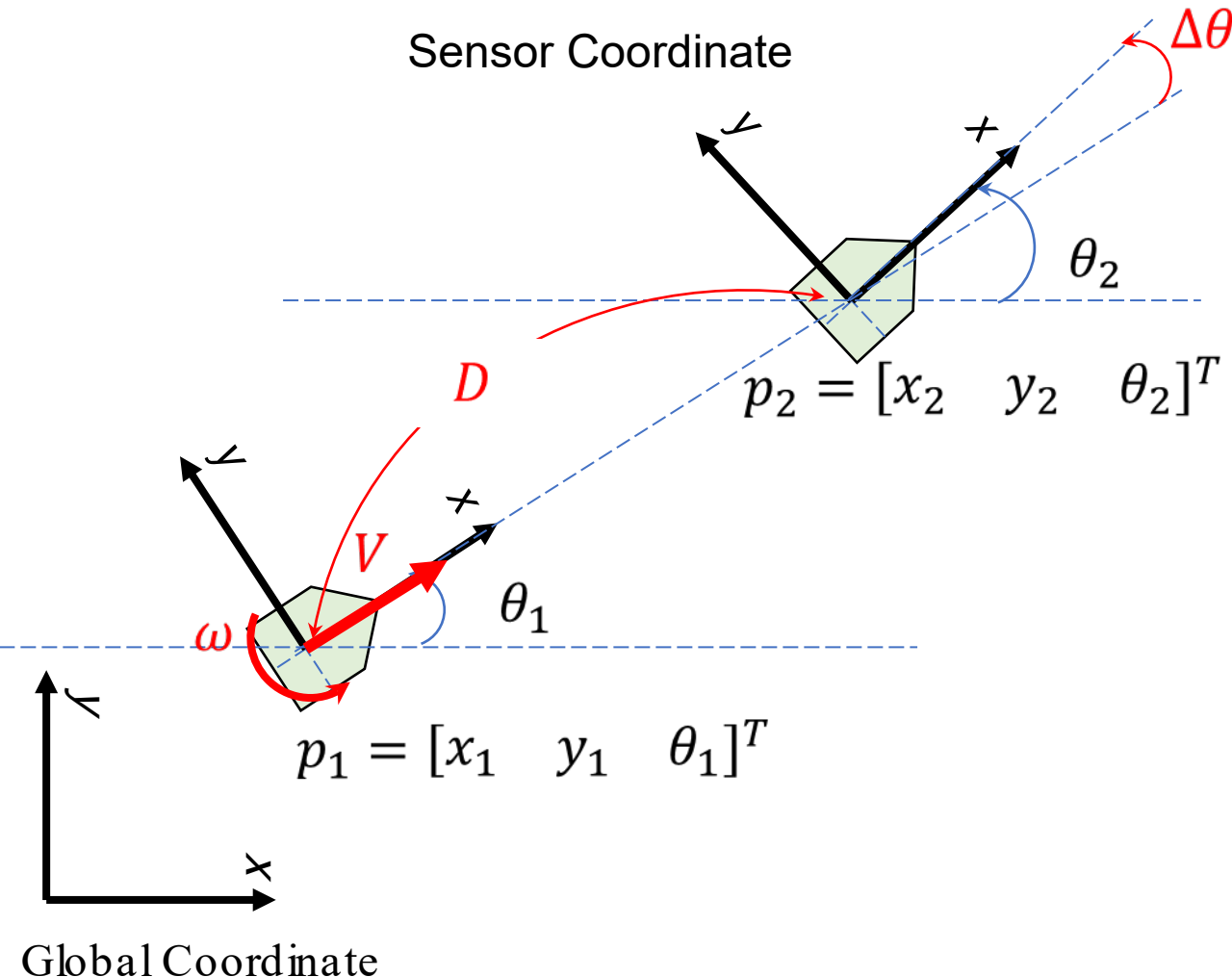
$$\begin{cases} D = V \cdot dt \\ \Delta\theta = \omega \cdot dt \end{cases}$$

位置推定 Estimated Position

$$\begin{cases} x_2 = \cos\theta_1 \cdot D + x_1 \\ y_2 = \sin\theta_1 \cdot D + y_1 \\ \theta_2 = \Delta\theta + \theta_1 \end{cases}$$



$$\begin{cases} x_2 = \cos\theta_1 \cdot V \cdot dt + x_1 \\ y_2 = \sin\theta_1 \cdot V \cdot dt + y_1 \\ \theta_2 = \omega \cdot dt + \theta_1 \end{cases}$$



# ホイールオドメトリの概要 #3

## Overview of Wheel Odometry #3

移動量 Amount of Movement

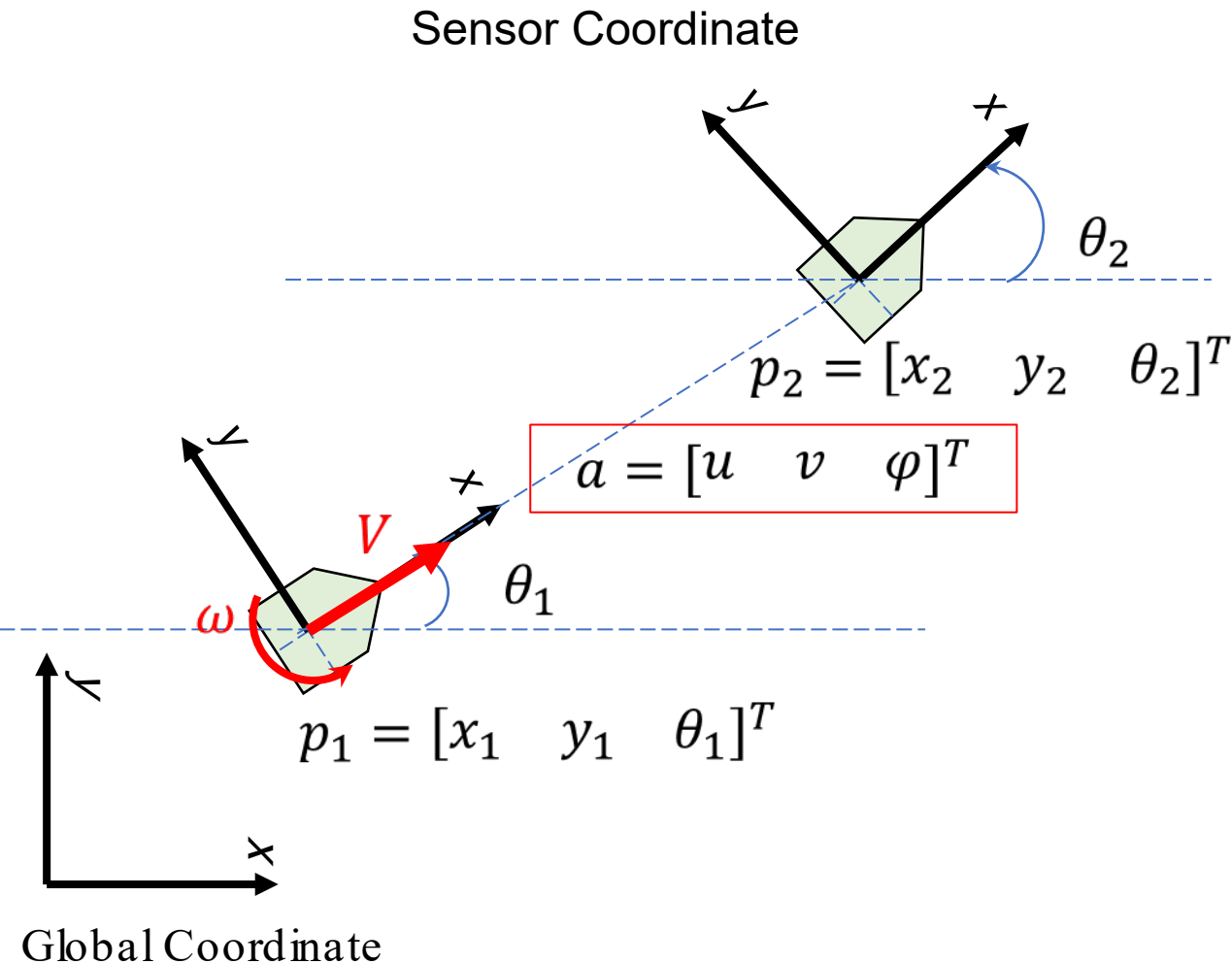
$$\begin{cases} u = V \cdot dt \\ v = 0 \\ \varphi = \omega \cdot dt \end{cases}$$

位置推定 Estimated Position

$$\begin{cases} x_2 = \cos\theta_1 \cdot u + x_1 \\ y_2 = \sin\theta_1 \cdot u + y_1 \\ \theta_2 = \varphi + \theta_1 \end{cases}$$

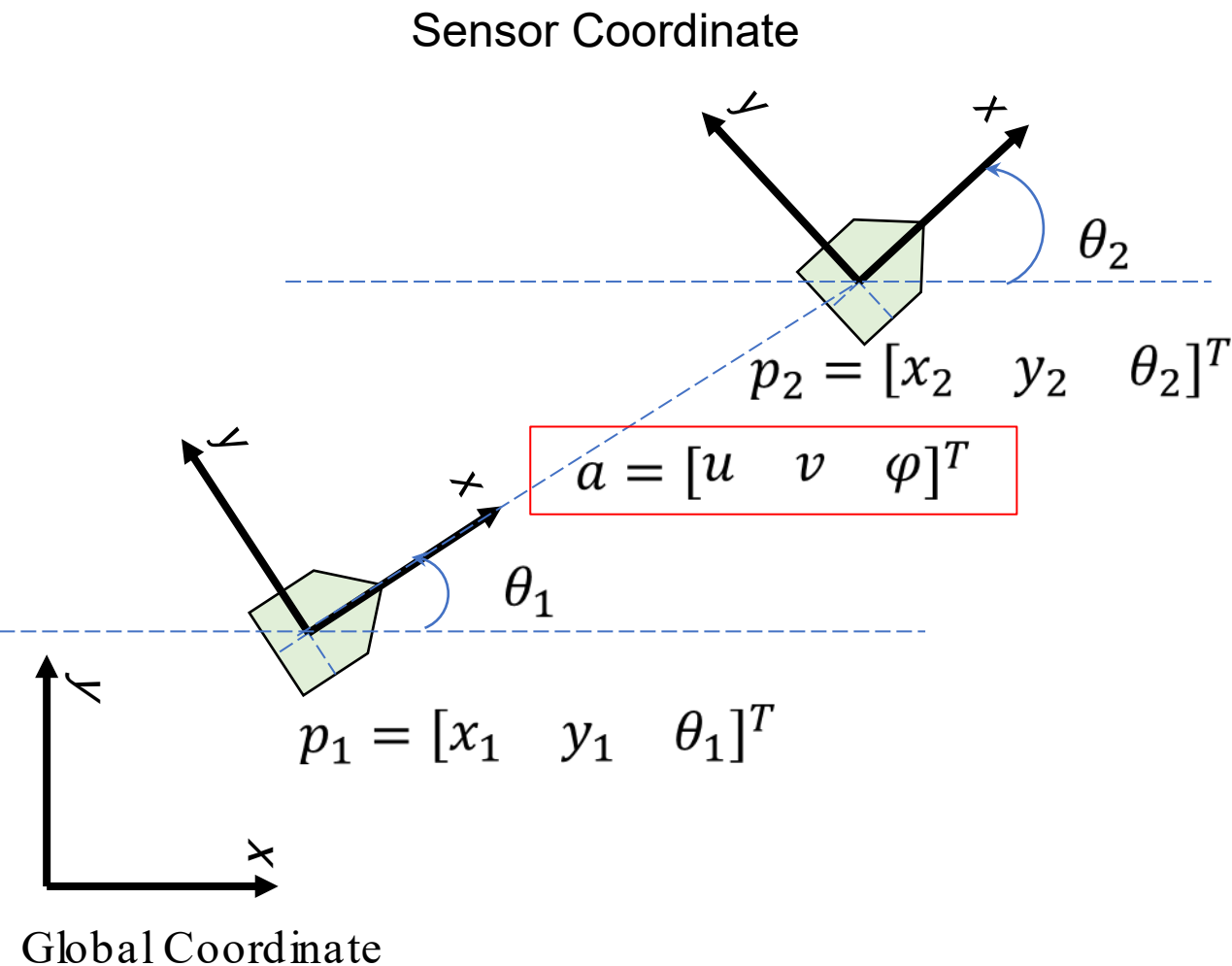


$$\begin{bmatrix} x_2 \\ y_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ \varphi \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \end{bmatrix}$$



# ホイールオドメトリの概要 #4

## Overview of Wheel Odometry #4



位置推定 Estimated Position

$$\begin{bmatrix} x_2 \\ y_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ \varphi \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \end{bmatrix}$$

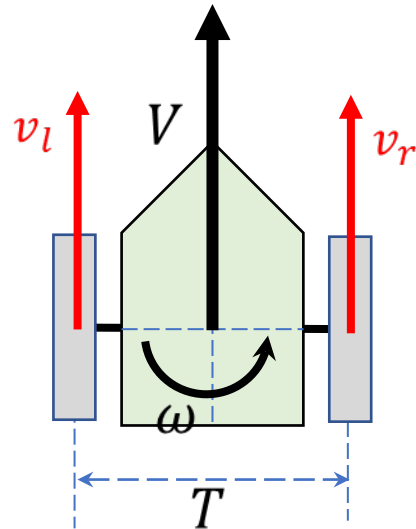
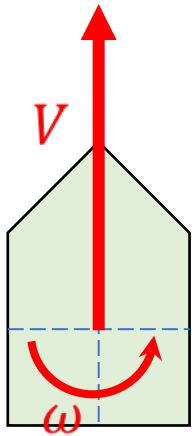
移動量 Amount of Movement

$$\begin{bmatrix} u \\ v \\ \varphi \end{bmatrix} =$$



# ホイールオドメトリの概要 #5

## Overview of Wheel Odometry #5



ロボットの速度と角速度を求める必要あり  
Need to estimate the velocity and angular  
velocity of a mobile robot.

位置推定 Estimated Position

$$\begin{cases} x_2 = \cos\theta_1 \cdot V \cdot dt + x_1 \\ y_2 = \sin\theta_1 \cdot V \cdot dt + y_1 \\ \theta_2 = \omega \cdot dt + \theta_1 \end{cases}$$

ロボットの速度 Velocity of Robot

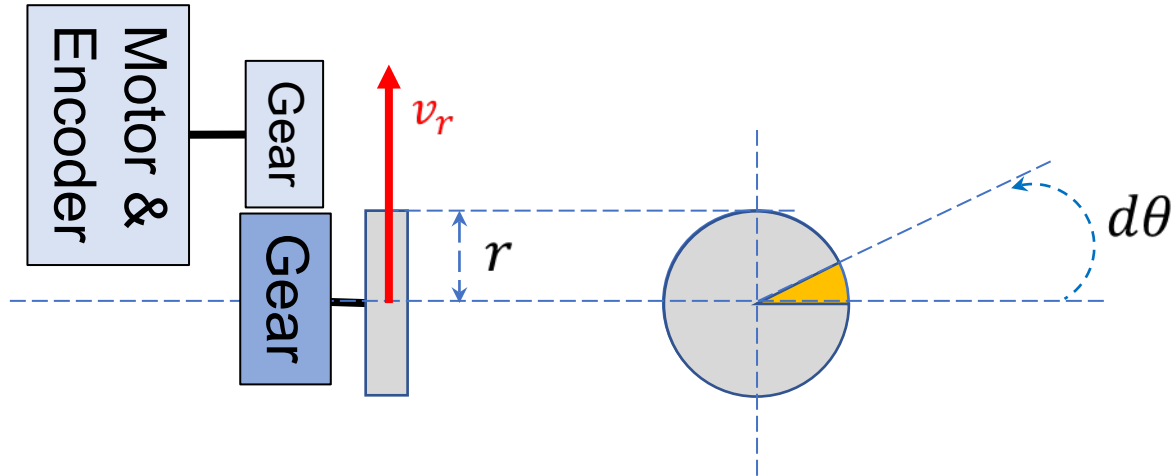
$$V = \frac{v_r + v_l}{2}$$

ロボットの角速度  
Angular Velocity of Robot

$$\omega = \frac{v_r - v_l}{T}$$

# ホイールオドメトリの概要 #6

## Overview of Wheel Odometry #6



エンコーダで車輪の速度を求める  
Calculate the velocity of the wheel by the encoder.

車輪速度 Velocity of Wheel

$$\begin{cases} dC = \frac{dE}{Gear} \\ d\theta = 2\pi \frac{dC}{CRev} \end{cases} \quad \begin{cases} v_r = r\omega_r \\ \omega_r = \frac{d\theta}{dt} \end{cases}$$

$dE$  : サンプルング周期毎のエンコーダのカウント  
Encoder counts per sampling period

$Gear$  : ギア比 Ratio of Gear

$dC$  : サンプルング周期毎の車輪軸上のカウント  
Counter on the wheel axle per sampling period

$CRev$  : 1回転分のカウント  
Num. of counter for one revolution



# エンコーダ Encoder

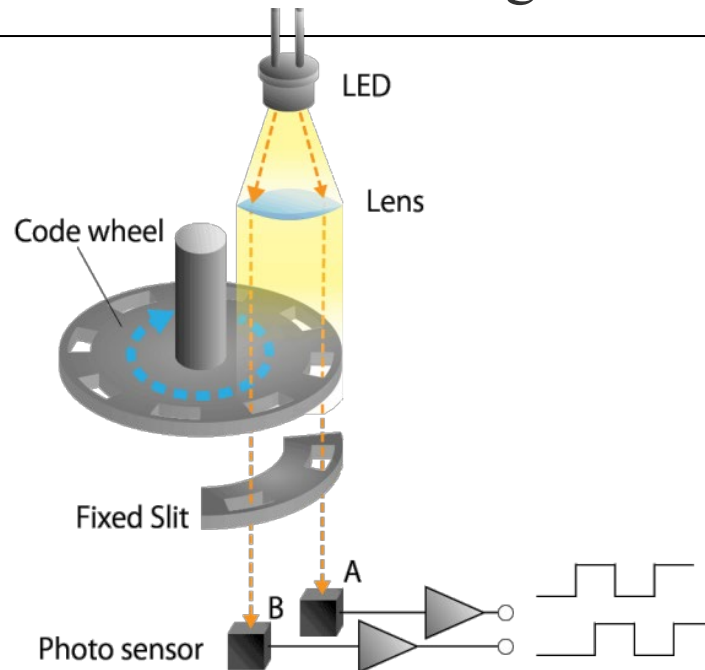
## <種類 Type>

- 機械式（接触式） Mechanical (Contact) type
- 光学式 Optical type
- 磁気式 Magnetic type
- 電磁誘導式 Electromagnetic Induction type



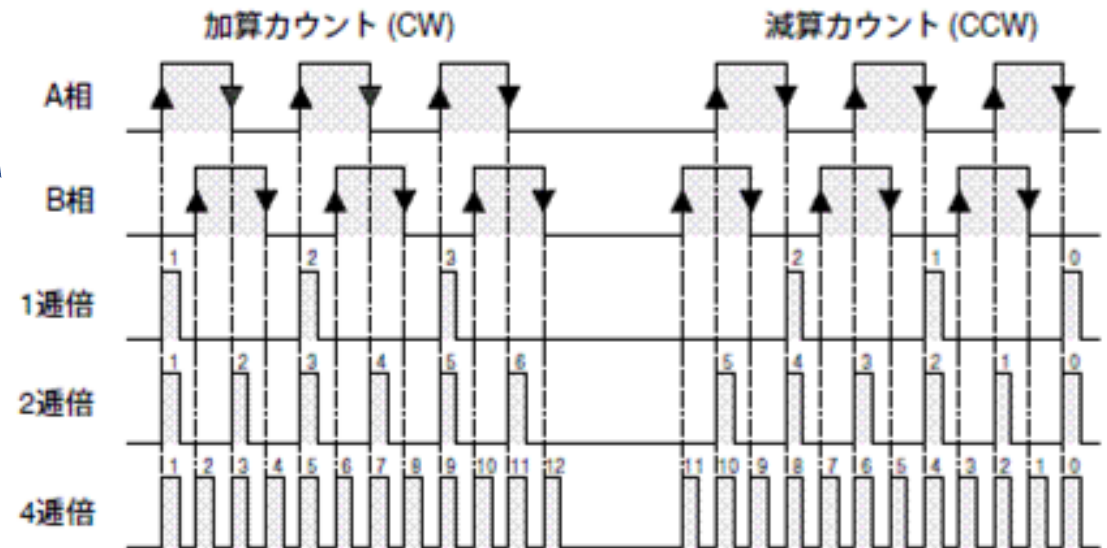
<https://www.monotaro.com/g/00161733/>

時計回り（CW : Clockwise）・反時計回り（  
CCW : Counter Clockwise）に何度回ったか



<https://www.akm.com/jp/ja/technology/technical-tutorial/basic-knowledge-encoder/optical-encoder/>

4-multiplication



<https://www.sk-solution.co.jp/html/qa/qad001c00820090629163814.html>

# プログラムによる実習（ホイールオドメトリ）

## Practical training with programming (Wheel Odometry)

- ホイールオドメトリによる自己位置推定プログラムを作成して提出する

Create and submit a self-positioning program using wheel odometry.

- 推定した位置(x,y)、方位(t-yaw)、速度(t-v)をグラフにして提出する

Submit a graph of your estimated position(x,y), orientation(t-yaw), and velocity(t-v).

- ファイルデータ形式 Data Format in Input file

# Time [s], Encoder Counter1(incremental difference), Encoder Counter2 (incremental difference), PWM1, PWM2

- パラメータ Parameters

車輪半径 Wheelradius : 0.14457758[m], トレッド Tread : -0.3045[m],

ギア比 Ratio of gear : 150,

1回転分のカウント Num. of counter for one revolution : 400

# IMUの紹介 Introduction of Inertial Measurement Unit

← → ↺ 🏠 ⚠ 保護されていない通信 | mems.tamagawa-seiki.com/product/memsimu.html



**tamagawa** 多摩川精機株式会社

PRODUCTS

MOVIES

TECHNOLOGIES

DOWNLOAD

APPLICATIONS

EVENTS

CONTACT

JP | EN

COMPANY

## MEMS IMU

姿勢角精度：0.1度  
各種定数の設定変更機能  
防水ケース仕様（TAG300 Series）

MEMS IMU（小型3軸慣性センサユニット）は、従来の慣性計測装置に比べて低価格ながら、外部GNSS接続可能なタイプに加え、拡張カルマンフィルタによる自律航法タイプもあり、機能性も優れたシリーズです。



AU7684



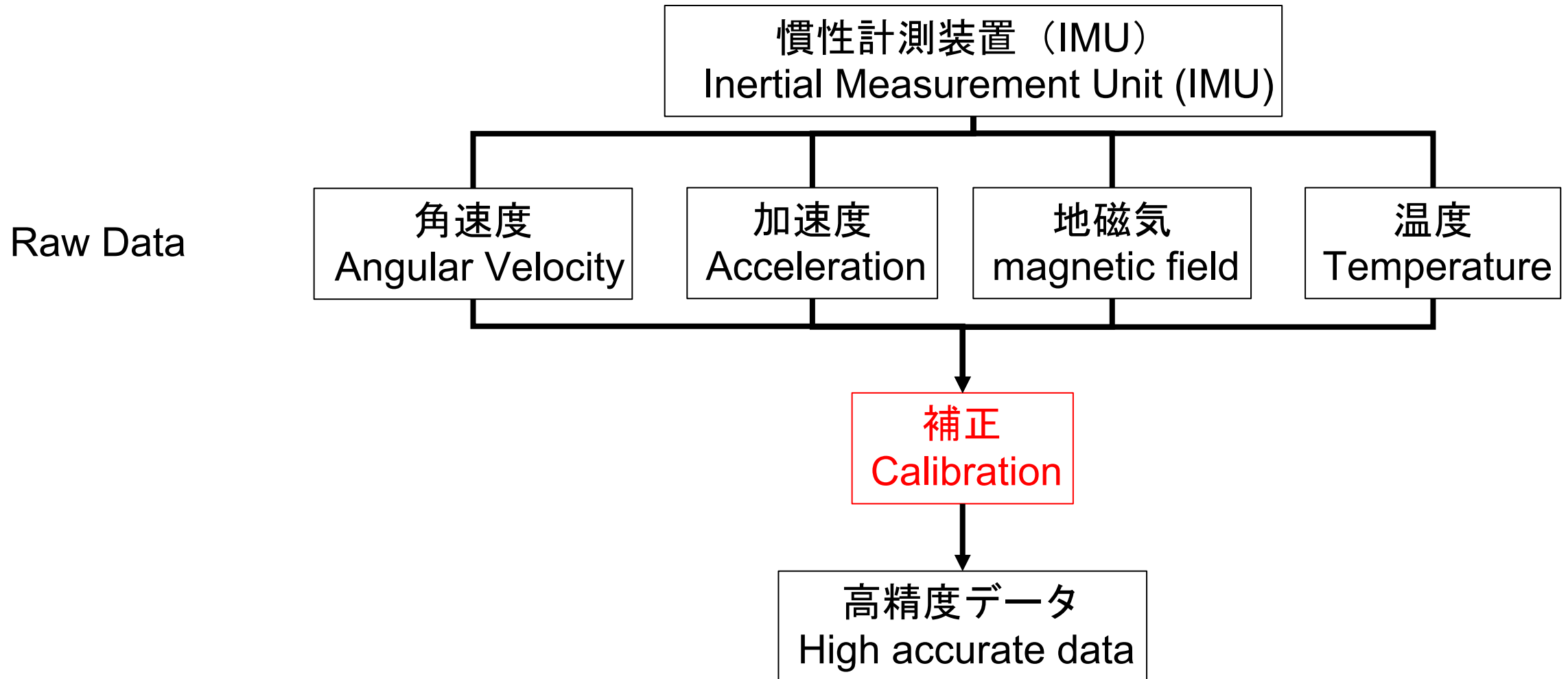
TAG300



TAG289

# IMUの概要

## Overview of IMU (Inertial Measurement Unit)





# ジャイロセンサと加速度センサの種類

## Types of Gyroscope sensor and Acceleration sensor

### ジャイロセンサ Gyroscope Sensor

MEMSジャイロ（振動ジャイロ）  
MEMS Gyroscope (Vibrating Gyroscope)

FOG（光ファイバジャイロ）  
FOG (FibreOptic Gyroscope)

RLG（リングレーザージャイロ）  
RLG (Ring Laser Gyroscope)

### 加速度センサ Acceleration Sensor

周波数変化式  
Frequency Change Type

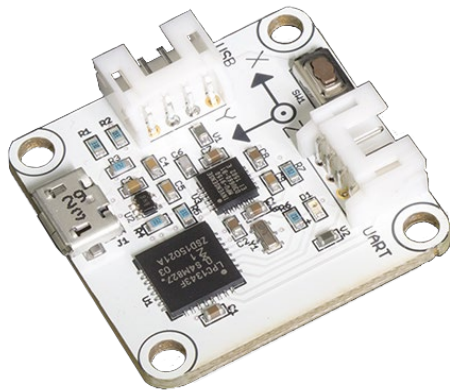
圧電式  
Piezoelectric Type

ピエゾ抵抗式  
Piezoresistive Type

静電容量式  
Capacitive Type

# IMUの紹介 Introduction of Inertial Measurement Unit

## USB出力9軸IMUセンサモジュール RT ROBOT SHOP



<https://www.rt-shop.jp/blog/archives/10921>

	Acceleration	Angular Velocity	Geomagnetism
Range	$\pm 16[g]$	$\pm 2000[\text{deg/sec}]$	$\pm 4800[\mu\text{T}]$
Resolution	16 bit	16 bit	16 bit

## 9DoF Razor IMU M0 SWITCH SCIENCE



<https://www.switch-science.com/catalog/3075/>

# IMUによる自己位置推定

## Self-Localization by using IMU

### <手順 Procedure>

#### □ ジャイロのゼロ点補正

Zero-point correction for a gyroscope sensor

#### □ グローバル座標における角度推定（ジャイロセンサと加速度センサ）

Angle estimation on global coordinates. (Each gyroscope sensor and acceleration sensor)

#### □ 補正フィルタ（相補フィルタ、カルマンフィルタ、LPフィルタなど）

Correctional filter (Complementary filter, Kalman filter, LP filter, etc.)

#### □ 重力加速度の補償

Compensation for gravitational acceleration

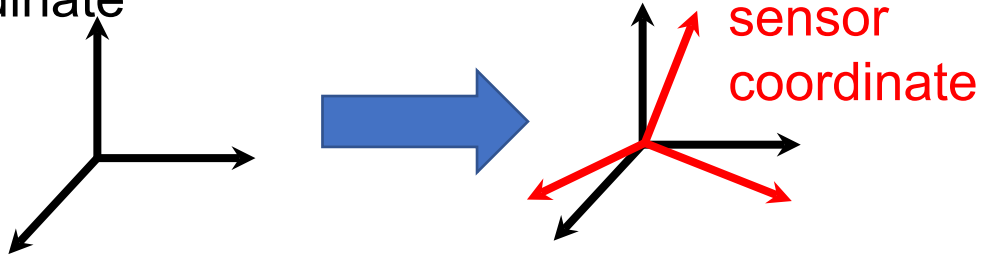
#### □ グローバル座標における加速度の2階積分

Second Order Integration of Acceleration on Global Coordinates

# 加速度の座標変換

## Acceleration Coordinate Transformation

global  
coordinate



<回転行列 rotation matrix>

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

<加速度の座標変換（センサ→グローバル）>  
Coordinate Transformation for Acceleration  
(sensor coordinate to global coordinate)

$$a_{global} = R_z(\psi)R_y(\theta)R_x(\phi)a_{sensor}$$

<加速度の座標変換（グローバル→センサ）>  
Coordinate Transformation for Acceleration  
(global coordinate to sensor coordinate)

$$a_{sensor} = [R_z(\psi)R_y(\theta)R_x(\phi)]^{-1} a_{global}$$

$$a_{sensor} = [R_z(\psi)R_y(\theta)R_x(\phi)]^T a_{global}$$



# 加速度センサから角度推定

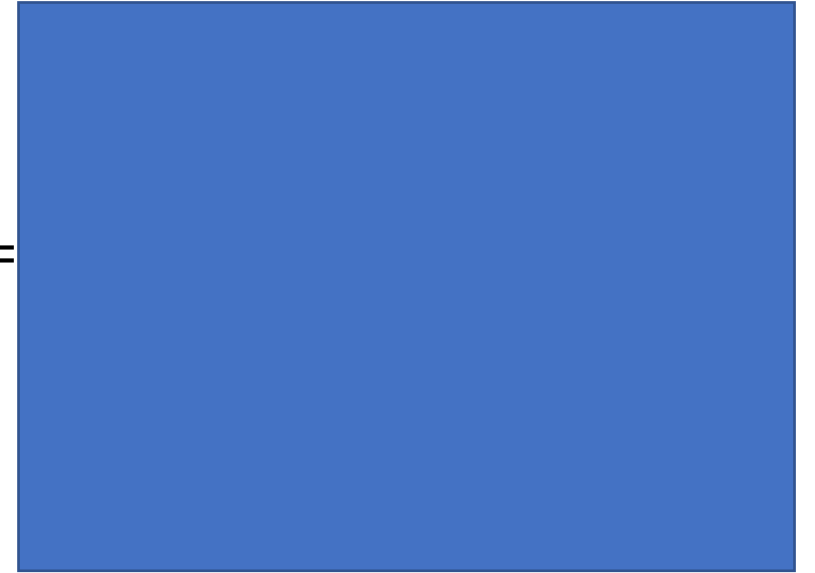
## Angle estimation from acceleration sensor

$$\begin{aligned} a_{sensor} &= [R_z(\psi)R_y(\theta)R_x(\phi)]^T a_{global} \\ &= [R_z(\psi)R_y(\theta)R_x(\phi)]^T \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \\ &= \begin{bmatrix} g\sin\theta \\ -g\cos\theta\sin\phi \\ -g\cos\theta\cos\phi \end{bmatrix} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \end{aligned}$$

重力加速度の補償

Compensation for gravitational acceleration

$$\begin{bmatrix} \phi \\ \theta \end{bmatrix} =$$

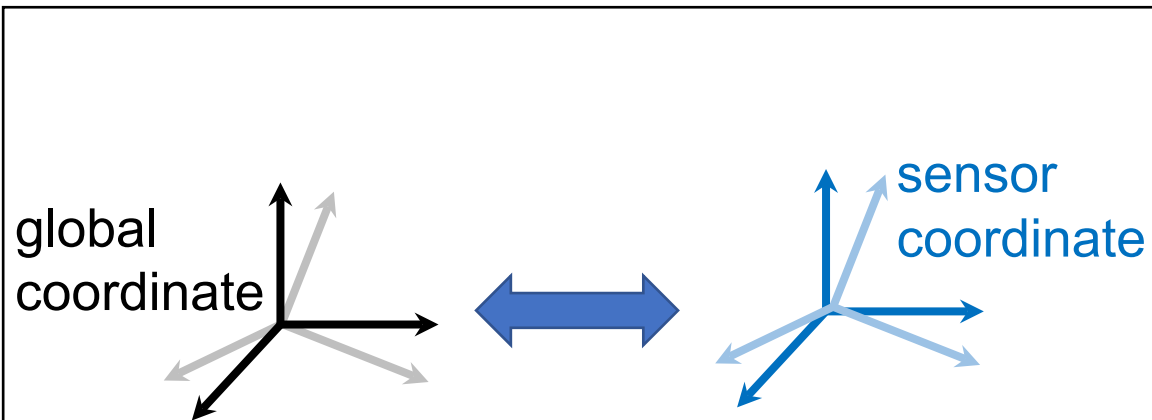


グローバル座標における角度推定

Angle estimation in global coordinates

# 角速度の座標変換

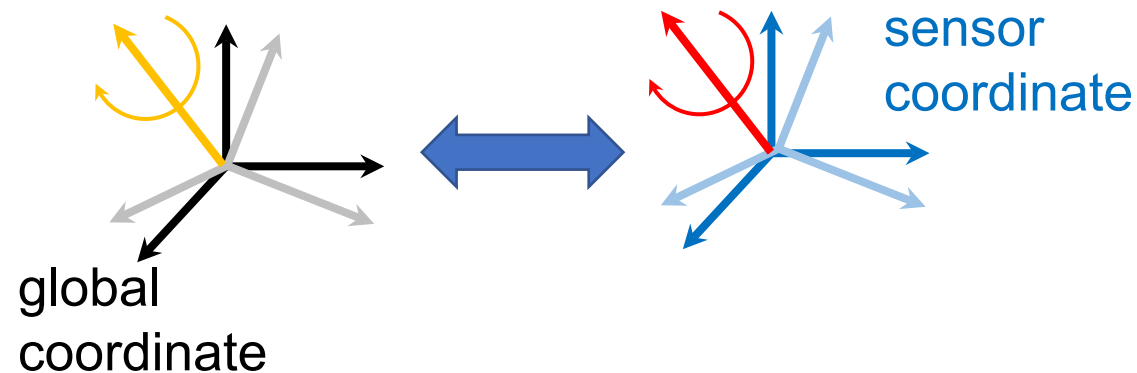
## Coordinate Transformation of Angular Velocity



$$\omega_{global} \neq [R_z(\psi)R_y(\theta)R_x(\phi)] \omega_{sensor}$$

$$\omega_{global} = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^T$$

$$\omega_{sensor} = [\omega_x \quad \omega_y \quad \omega_z]^T$$



$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

グローバル座標における角速度推定  
Angular velocity estimation in global coordinates

# 角速度センサから角度推定

## Angle estimation from angular velocity sensor

$$\begin{bmatrix} \phi(t+1) \\ \theta(t+1) \\ \psi(t+1) \end{bmatrix} = \begin{bmatrix} \phi(t) \\ \theta(t) \\ \psi(t) \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} dt$$

グローバル座標における角度推定  
Angle estimation in global coordinates

# 相補フィルタ Complementary Filter

## ジャイロセンサ Gyroscope Sensor

### <特徴 Features>

- ❑ 角速度を積分して角度を推定  
Integrate the angular velocity to estimate the angle
- ❑ 長所：並進運動の影響を受けない。  
短期的には精度が良い。  
Pros : Unaffected by translational motion. **Good accuracy in the short term**
- ❑ 短所：累積誤差（ジャイロドリフト）  
Cons : Accumulating errors (Gyro Drift)

## 加速度センサ Acceleration sensor

### <特徴 Features>

- ❑ 重力加速度の向きから角度を推定  
Estimate the angle from the direction of the gravitational acceleration
- ❑ 長期的には精度が良い。  
Pros : **Good accuracy in the long term.**
- ❑ 短所：ヨ一角を推定できない。短期的には外乱の影響を受ける。  
Cons: Cannot estimate yaw angle. Affected by disturbances in the short term

センサ融合（ジャイロにハイパスフィルタ、加速度にローパスフィルタをかけて加算）  
Sensor Fusion (Add the gyro with a high-pass filter and the acceleration with a low-pass filter)

# 相補フィルタの適用

## Applying Complementary Filter

$$\begin{aligned}\Theta_{fusion} &= K \cdot \Theta_{gyro} + (1 - K) \cdot \Theta_{accel} \\ &= \Theta_{accel} + K(\Theta_{gyro} - \Theta_{accel})\end{aligned}$$

$$K = 0.999$$

# プログラムによる実習 (IMU) #1

## Practical training with programming (IMU)

- IMUを用いた姿勢推定(roll, pitch, yaw)プログラムを作成して提出する

Create and submit estimated angles (roll, pitch, yaw) program using IMU.

- 推定した姿勢をグラフにして提出する

Submit a graph of your estimated angles.

- ファイルデータ形式 Data Format in Input file

# Time [s], Angular velocity(x) [rad/s], Angular velocity(y), Angular velocity(z),  
Acceleration(x) [g], Acceleration(y), Acceleration(z), other(x), other(y), other(z),  
temperature [degree]

- パラメータ Parameters

....

# プログラムによる実習 (IMU) #2

## Practical training with programming (IMU)

- IMUを用いた自己位置推定プログラムを作成して提出する

Create and submit a self-positioning program using IMU.

- 推定した位置(x,y)、方位(t-yaw)、速度(t-v)をグラフにして提出する

Submit a graph of your estimated position(x,y), orientation(t-yaw), and velocity(t-v).

- ファイルデータ形式 Data Format in Input file

# Time [s], Angular velocity(x) [rad/s], Angular velocity(y), Angular velocity(z),  
Acceleration(x) [g], Acceleration(y), Acceleration(z), other(x), other(y), other(z),  
temperature [degree]

- パラメータ Parameters

....