

Robot Operating System

Chapter 5

Mobile Robot FK & Implementation

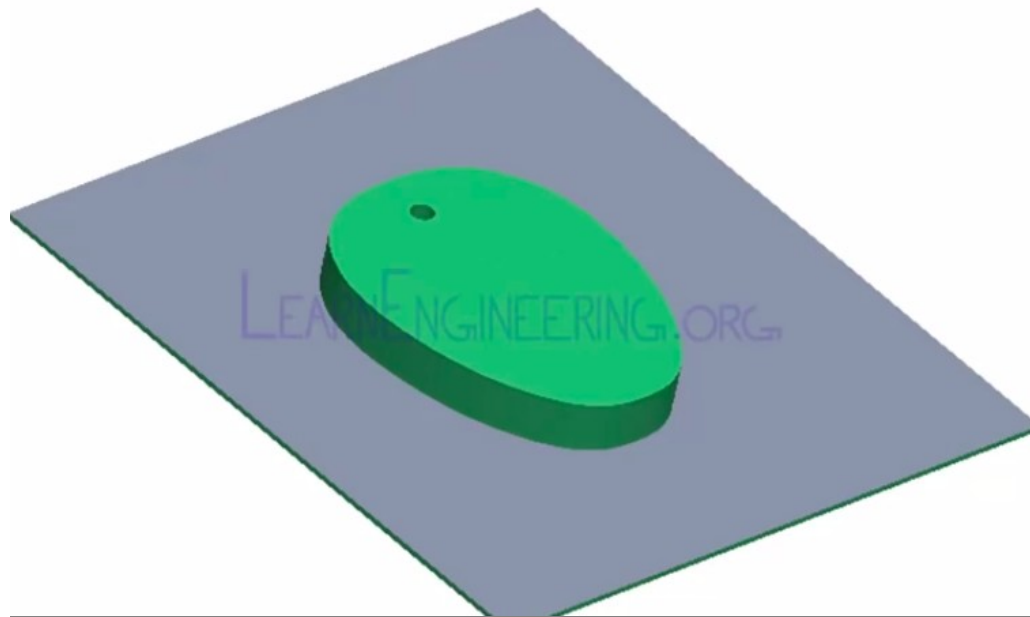
Pyae Soan Aung
RHCSA
Rom Robotics Co.ltd

Mobile Robot Kinematics

Outline

- **Degrees of Freedom**
- **Holonomic / Non-Holonomic**
- **Ominidirectional Drive / Mecanum Drive**
- **Differential Drive**
- **Instantaneous Center**
- **Mecanum Drive Kinematic (Skip)**
- **Differential Drive Forward Kinematic & How to Implement**

Degrees of Freedom



3 D.O.F

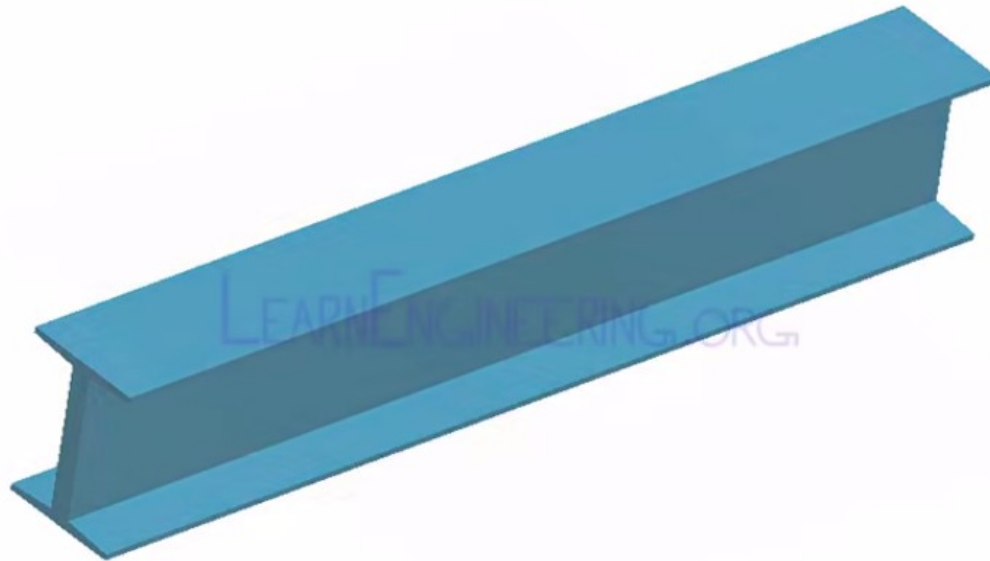
2 TRANSLATIONAL X Y

1 ROTATIONAL θ_z

Source from internet

In a planar mechanisms, all of the relative motions of the rigid bodies are in one plane or in parallel planes. If there is any relative motion that is not in the same plane or in parallel planes, the mechanism is called the spatial mechanism. In other words, planar mechanisms are essentially two dimensional while spatial mechanisms are three dimensional.

Degrees of Freedom



6 D.O.F

3 TRANSLATIONAL X Y Z

3 ROTATIONAL θ_x θ_y θ_z

Grubler's formula

$$\text{dof} = \underbrace{m(N-1)}_{\text{rigid body freedoms}} - \underbrace{\sum_{i=1}^J c_i}_{\text{joint constraints}}$$

$$= m(N-1) - \sum_{i=1}^J (m - f_i)$$

Dof = freedom of body

N = # of body (including ground)

J = # of Joints

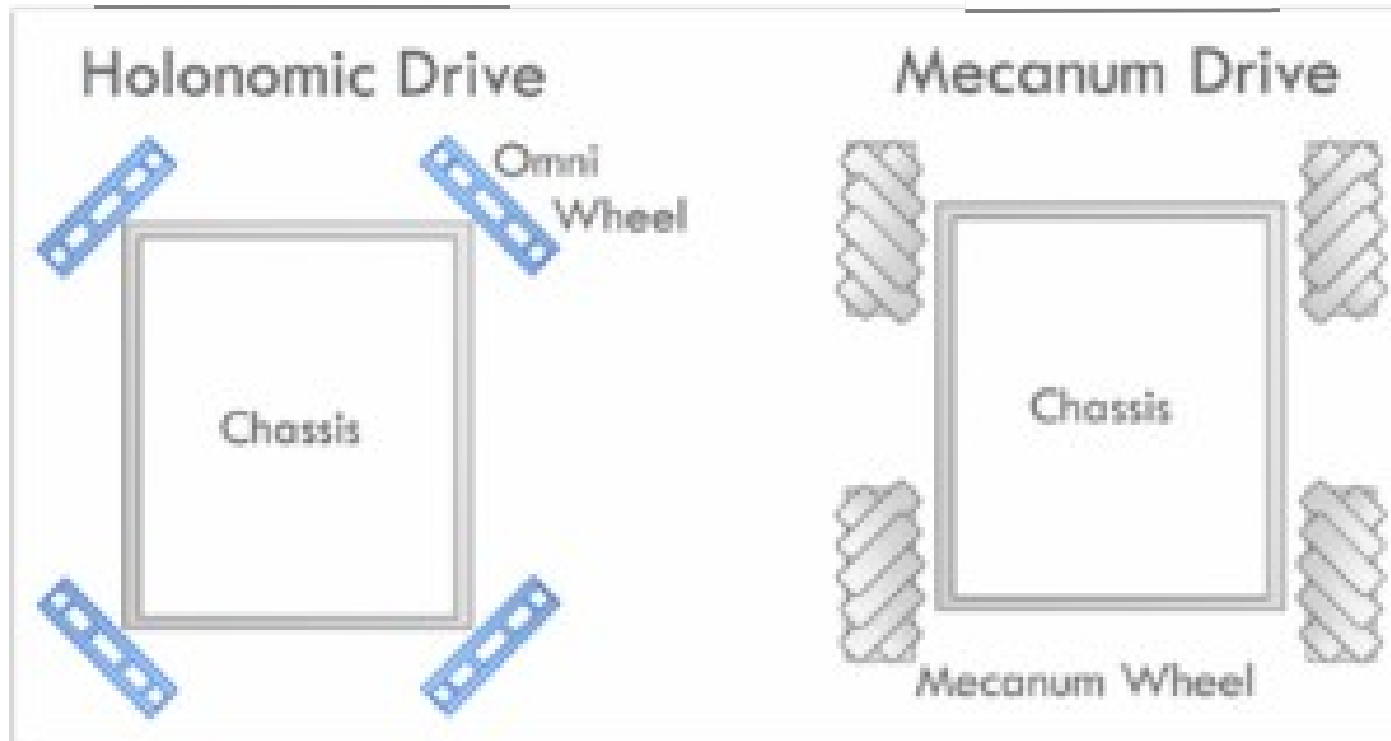
M = 6 for spatial body, 3 for planar

Joint type	Dof f	C in 2 pl rig	C in 2 sp rig
Revolute(R)	1	2	5
Prismatic(P)	1	2	5
Helical(H)	1	N/A	5
Cylindrical(C)	2	N/A	4
Universal(U)	2	N/A	4
Spherical(S)	3	N/A	3

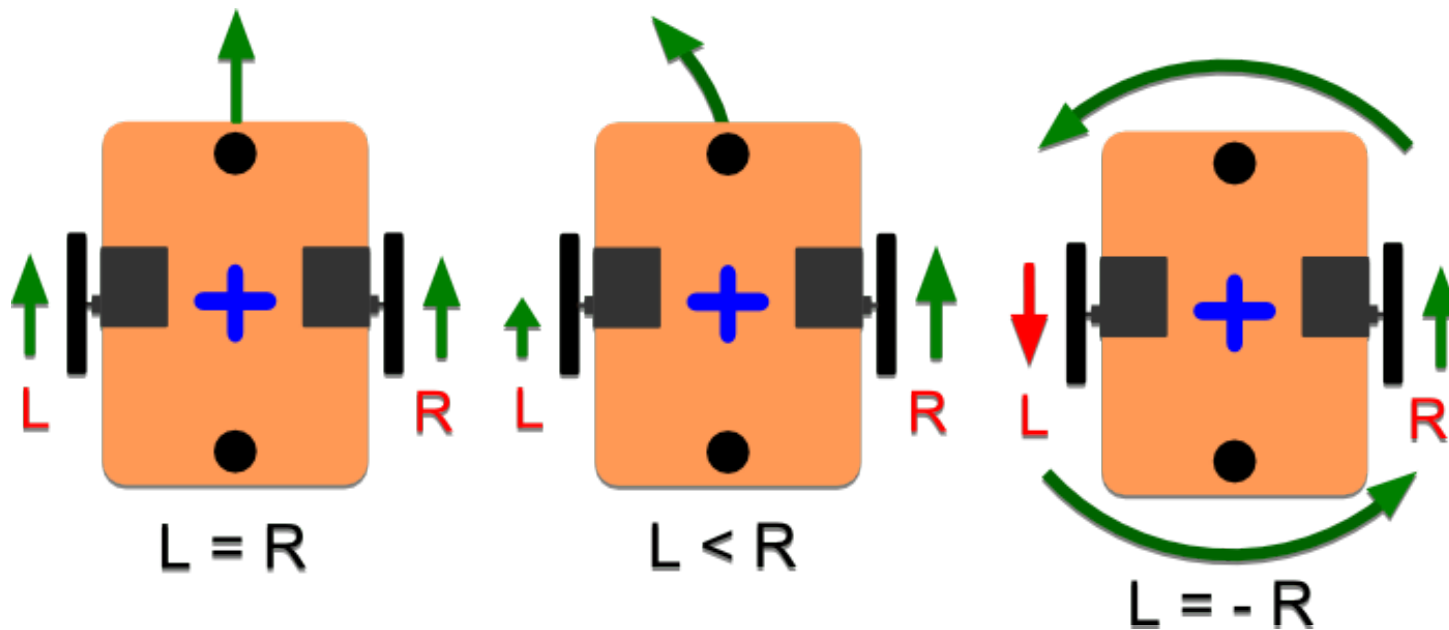
Holonomic / NonHolonomic

- **Holonomic / NonHolonomic** ဆိုတာ **control** လုပ်လို့ရတဲ့ **degree of freedom** နဲ့ **total degree of freedom** တို့ရဲ့ **relation** ကို ခေါ်တာပါ။
- **Control** လုပ်လို့ရတဲ့ **dof** အရေအတွက်ဟာ **Total dof** အရေအတွက်ထက် နည်းတယ်ဆိုရင် **NonHolonomic** လို့မှတ်ယူနိုင်ပါတယ်။ **mobile robot** တွေ ၊ အိမ်စီးကားတွေကို **NonHolonomic** လို့ခေါ်နိုင်ပါတယ်။ **control** လုပ်လို့ရနိုင်တာက **speed** ရယ် **steering** ရယ် နှစ်ခုရှိပြီး **x,y,theta** အတိုင်း **position** ကို ရရှိမှာဖြစ်ပါတယ်။
- **Control** လုပ်လို့ရတဲ့ **dof = Total dof** ဆိုရင်တော့ **Holonomic**
- **Control** လုပ်လို့ရတဲ့ **dof** က **Total dof** ထက်များရင်တော့ **Redundant drive** လို့ခေါ်ပါတယ်။ ဥပမာ **servo** မော်တာ ၇လုံး စလုံးတပ်ထားတဲ့ **robot arm** တွေပါ။

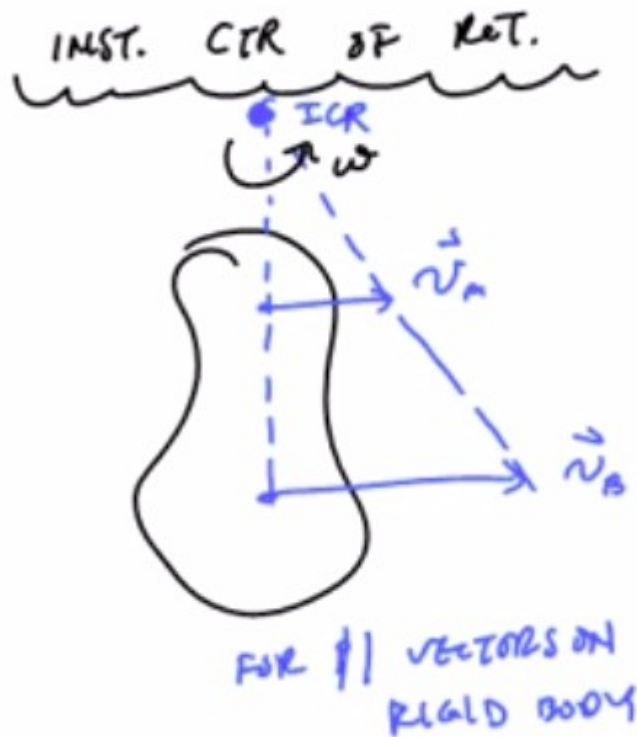
Ominidirectional Drive / Mecanum Drive



Differential Drive

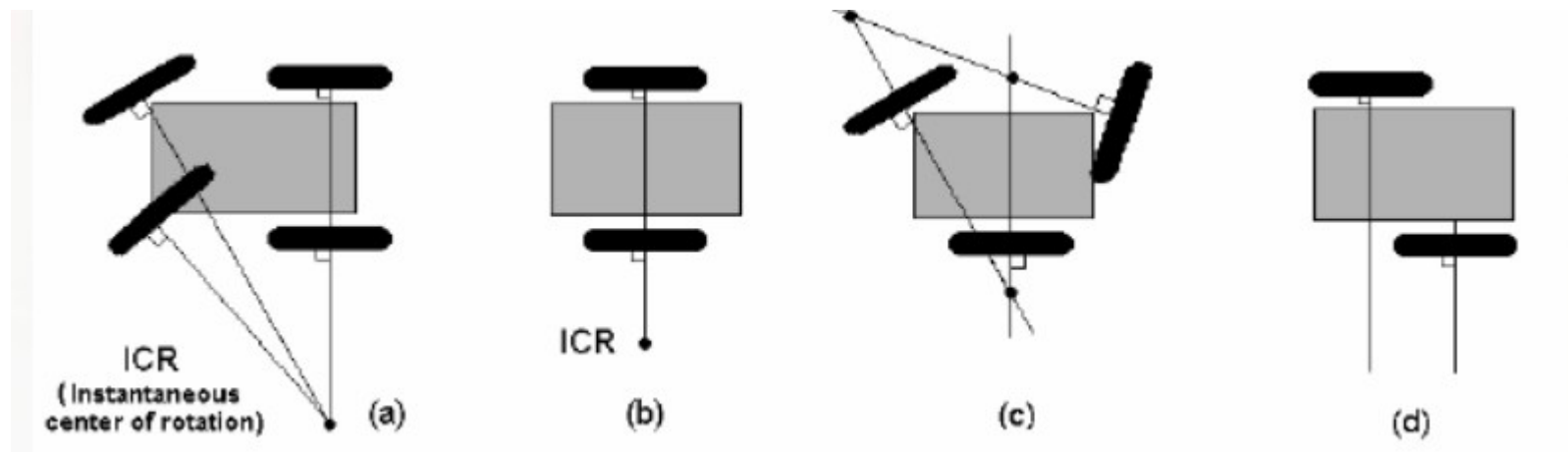


Instantaneous Center



Rigid body တစ်ခုမှာ **magnitude** မတူတဲ့ **Force vector** နှစ်ခုဟာ ပုံပါအတိုင်းသက်ရောက်ရင် **Instantaneous Center** ကို ဗဟိုပြုပြီး **motion** ဖြစ်ပါတယ်။
Perpendicular to Vector အတိုင်း **line** ဆွဲပြီး **Instantaneous Center** ကိုတွေ့နိုင်ပါတယ်။

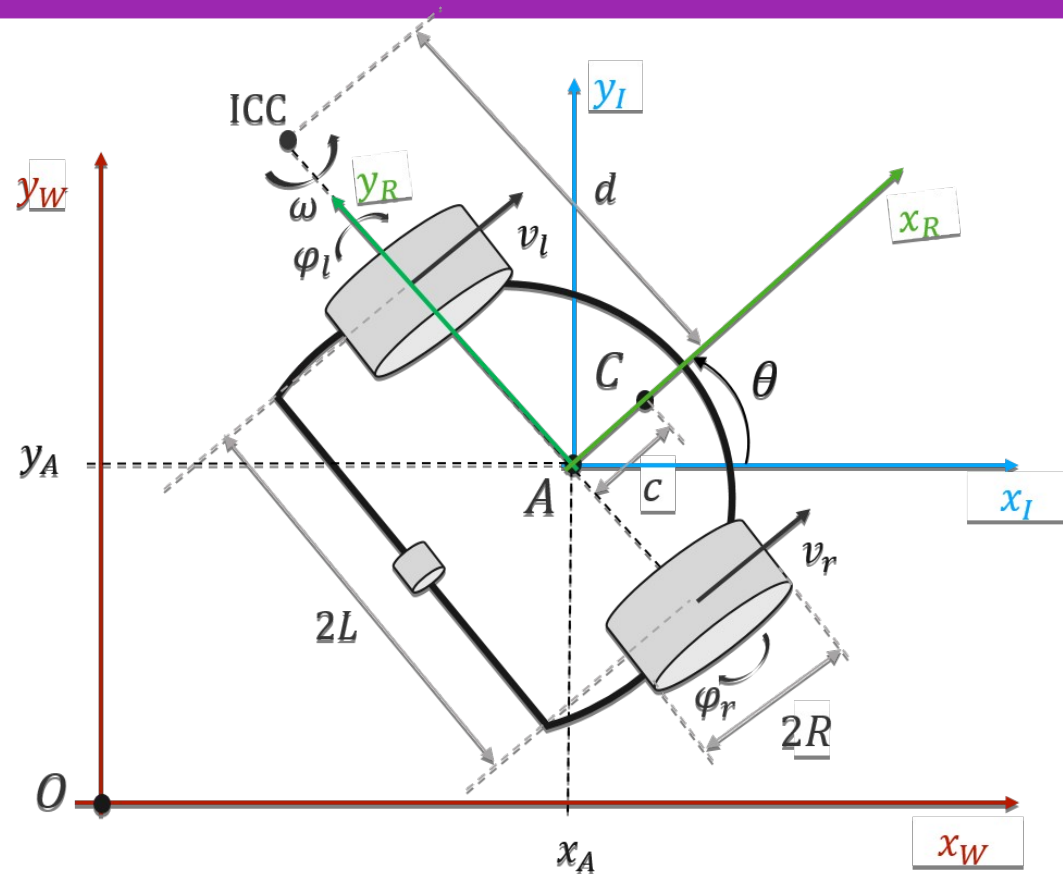
Instantaneous Center



Instantaneous Center အရေအတွက် ၁ခု မက ရှိနိုင်ပြီး အမျိုးအစား ၃ မျိုးခွဲနိုင်တယ်။

- **fixed**
- **permanent**
- **Neither fixed nor permanent**

Instantaneous Center



Differential drive mobile robot တွေမှာ ဘီးရဲ့ rotate ဖြစ်တဲ့ Y-axis အတိုင်း ဆက်ဆွဲပြီး
Robot ICC Point ကိုရရှိပါတယ်။

Robot ဟာ အဲ့ဒီ **ICC Point** ကို ဗဟိုပြုပြီး **Motion** ဖြစ်ပါတယ်။

Robot ရဲ့ **Position** ကိုတော့ **reference frame** တစ်ခုကနေတိုင်းတာပါတယ်။

Differential Drive Kinematic

Kinematic ဘာလဲ?

- Study of Motion ဖြစ်ပြီး Force ကိုထည့်သွင်းမတွက်ချက်ပါဘူး။ (No Dynamic)

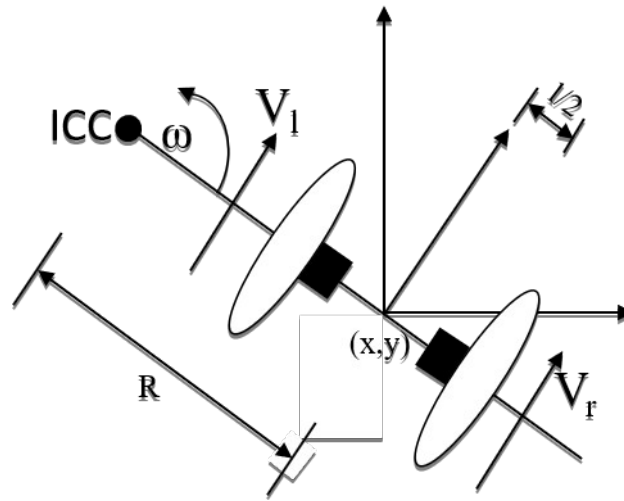
ဥပမာ Forward Kinematic

Mobile robot ရဲ့ Kinematic model ဆိုတာက wheel speed ဘယ်လောက် မှာ robot ရဲ့ position ဘယ်လောက်လဲတွက်ချက်တာမျိုး
Robot arm တွေဆိုရင် joint value ဘယ်လောက်မှာ eef position ဘယ်လောက်လဲ?

ဥပမာ Inverse Kinematic

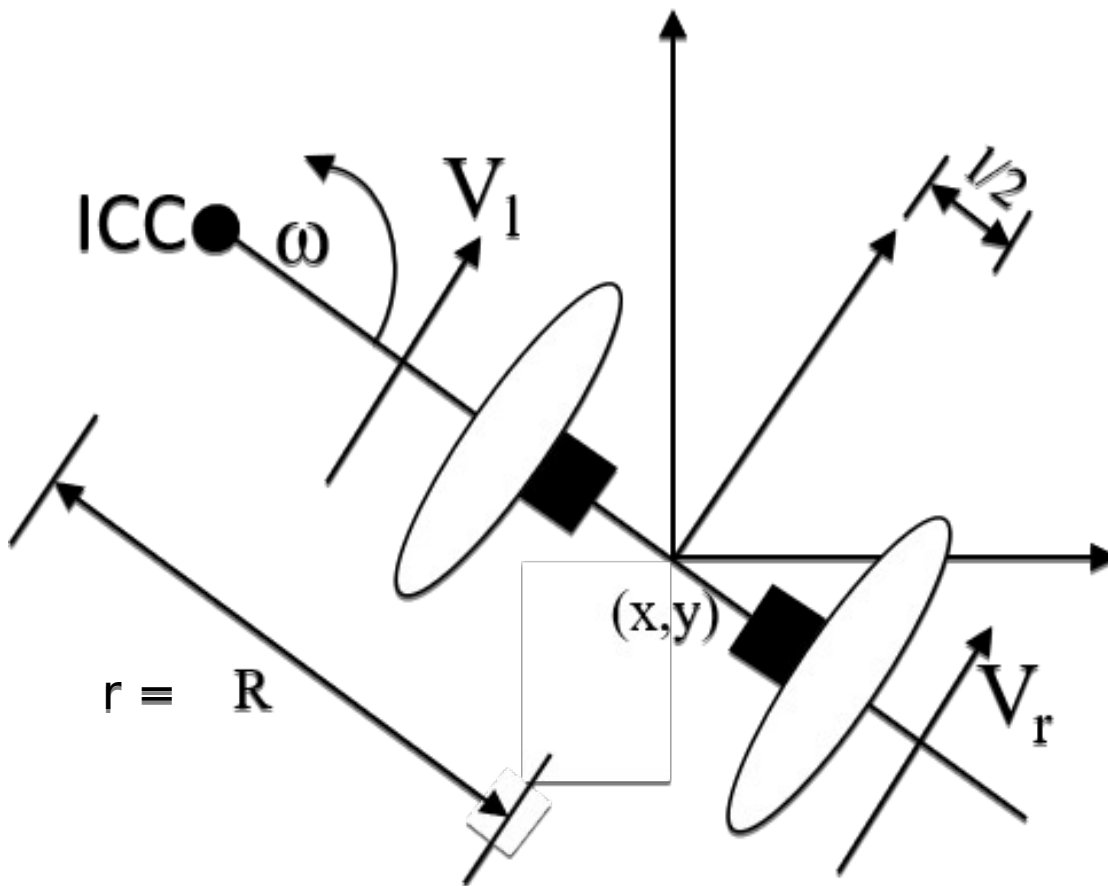
Mobile robot ရဲ့ Kinematic model ဆိုတာက robot ရဲ့ position ဘယ်လောက် မှာ wheel speed ဘယ်လောက်လဲတွက်ချက်တာမျိုး
Robot arm တွေဆိုရင် eef position ဘယ်လောက်မှာ joint value ဘယ်လောက်လဲ?

Differential Drive Kinematic Forward Kinematic



- ကျွန်တော်တို့ရဲ့ Mobile Robot မှာ 3 dof (x, y, yaw) ရှိပြီး x, y က position ၊ Yaw က orientation ဖြစ်ပါတယ်။
- Mobile Robot တွေမှာ ဘီးတစ်ခုစီဟာ သူတို့ရဲ့ Y-axis မှာ Rolling ဖြစ်ပြီး robot အတွက် ICC ရှိပါတယ်။
- Differential Drive တွေဟာ ICC ကို Center ပြုပြီး Forward , Backward သွားပါတယ်။
- ကျွန်တို့အခါ ဘီးနှစ်ခုရဲ့ Center ကို ဗဟိုပြုပြီး Spin ဖြစ်ပါတယ်။ ဒါကို ICC နေရာပြောင်းခြင်း သို့မဟုတ် ICC နဲ့ Center အကွာအဝေး $R = 0$ ဖြစ်ခြင်းလို့ယူဆနိုင်ပါတယ်။

Differential Drive Kinematic Forward Kinematic



$$\text{Velocity}_{\text{linear}} = V = 2\pi r/T$$

$$\text{Velocity}_{\text{angular}} = \omega = 2\pi/T$$

$$V = r\omega$$

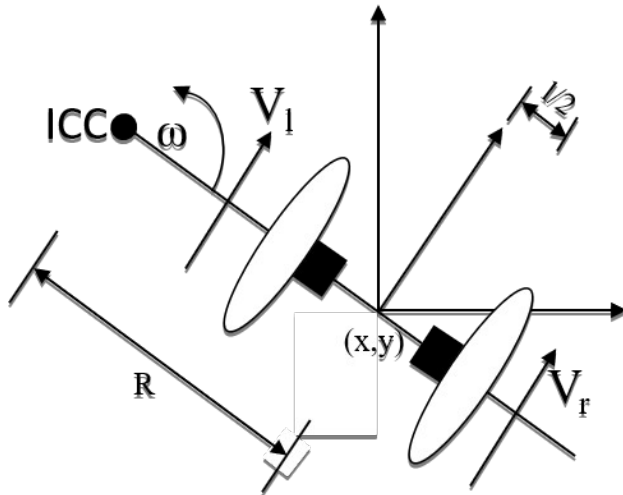
$$V_{\text{LEFT}} = \omega (R - L/2)$$

$$V_{\text{RIGHT}} = \omega (R + L/2)$$

$$R = L / 2 (V_l + V_r / V_r - V_l)$$

$$\omega = (V_r - V_l) / L$$

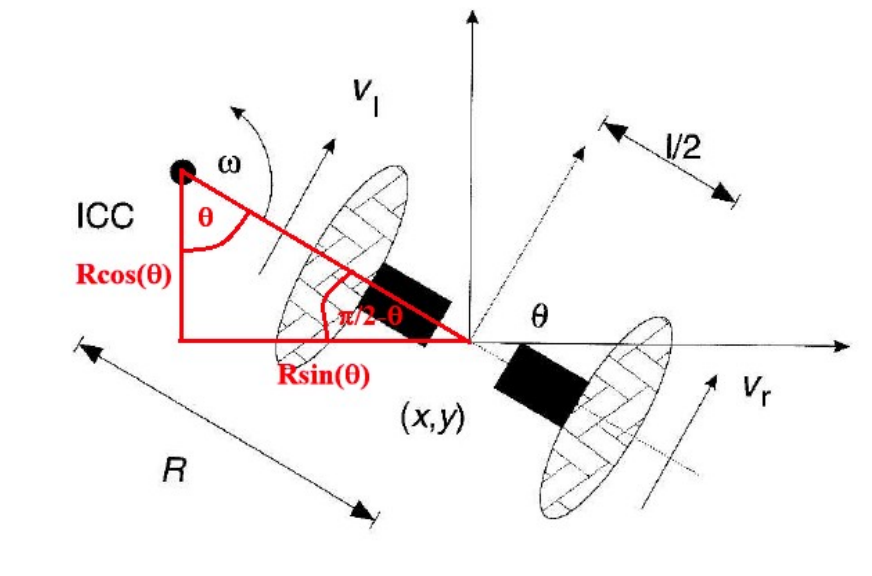
Differential Drive Kinematic Forward Kinematic



V_{left} နဲ့ V_{right} ကို တန်ဖိုးအမျိုးမျိုးပေးခြင်းဖြင့် ကွဲပြားတဲ့ လမ်းကြောင်းတွေရရှိလာပါမယ်။

- ၁) V_{left} နဲ့ V_{right} တူမယ်ဆိုရင် robot ဟာ straight line အတိုင်းသွားမယ်။ R တန်ဖိုးအဆုံးမရှိဖြစ်ပြီး ω တန်ဖိုး 0 ဖြစ်ပါမယ်။
- ၂) $V_{left} = -V_{right}$ ဆိုရင် R တန်ဖိုး 0 ဖြစ်ပြီး robot ဟာ သူ့နေရာမှာဘဲ spin ဖြစ်နေမယ်။
- ၃) V_{left} နဲ့ V_{right} တခုခုသာ 0 ဖြစ်မယ်ဆိုရင် R တန်ဖိုးဟာ $L/2$ ဖြစ်ပြီး Velocity 0 ဖြစ်နေတဲ့ဘက်ကို ကွေ့မှာဖြစ်ပါတယ်။

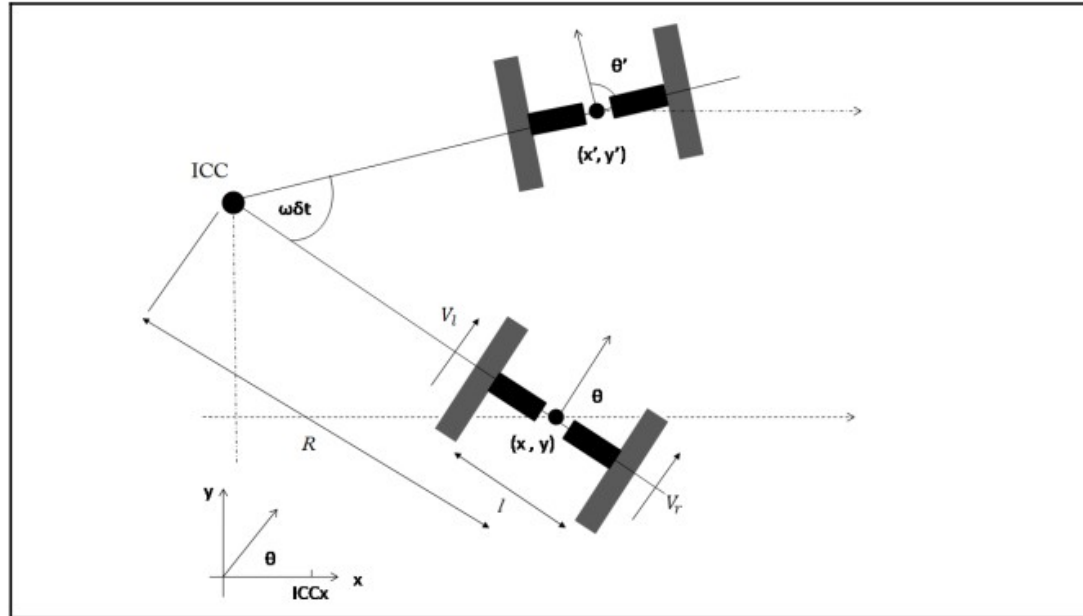
Differential Drive Kinematic Forward Kinematic



$$\begin{aligned} \text{ICCX} &= x - R * \sin(\theta) \\ \text{ICCY} &= y + R * \cos(\theta) \end{aligned}$$

Differential Drive Kinematic Forward Kinematic

Velocity Input

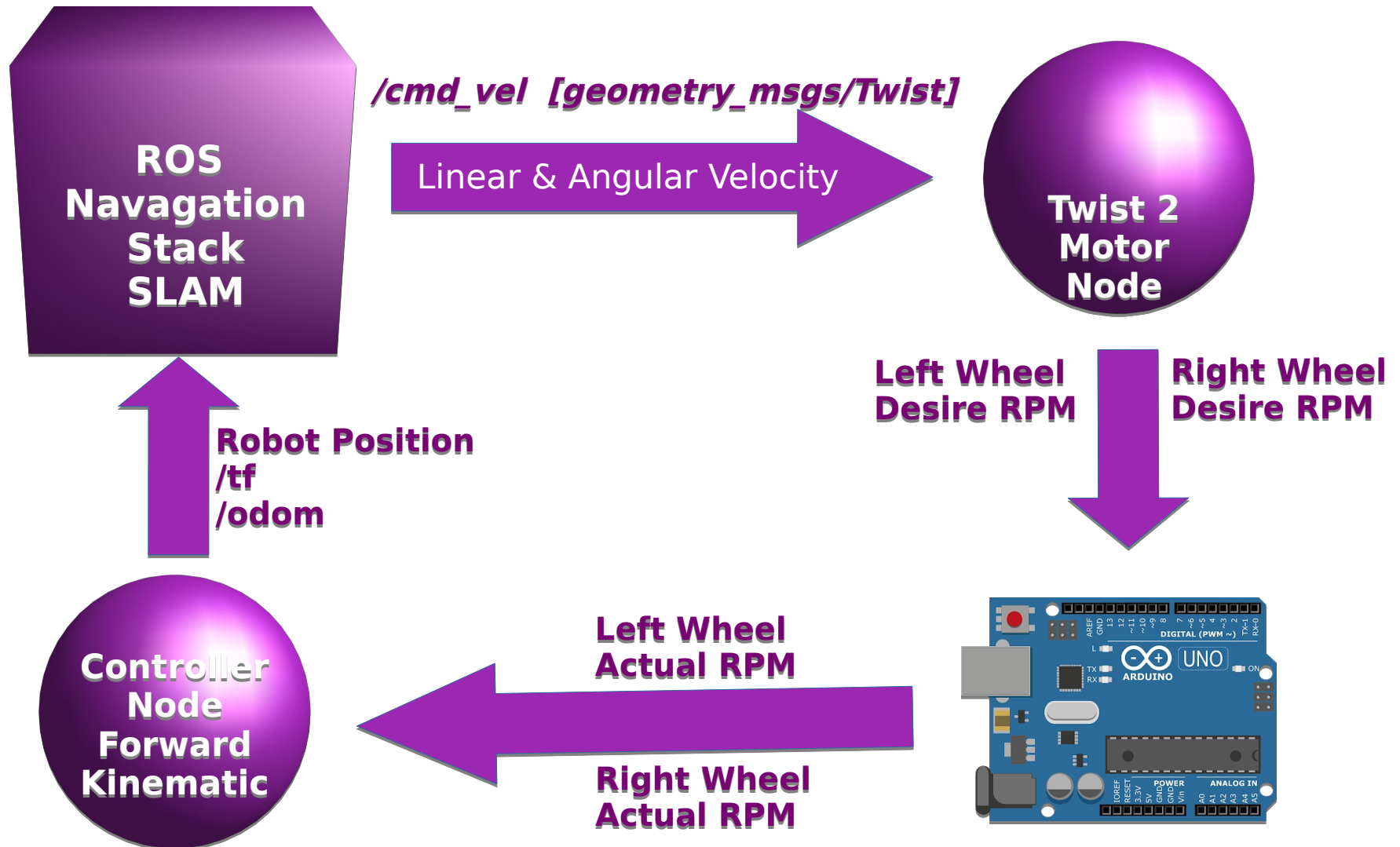


Position Output

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega \delta t) & -\sin(\omega \delta t) & 0 \\ \sin(\omega \delta t) & \cos(\omega \delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega \delta t \end{bmatrix}$$

$$\begin{aligned} X' &= x \cdot \cos(\omega \cdot \delta t) - y \cdot \sin(\omega \cdot \delta t) \\ Y' &= x \cdot \sin(\omega \cdot \delta t) + y \cdot \cos(\omega \cdot \delta t) \\ \theta' &= \omega \cdot \delta t \end{aligned}$$

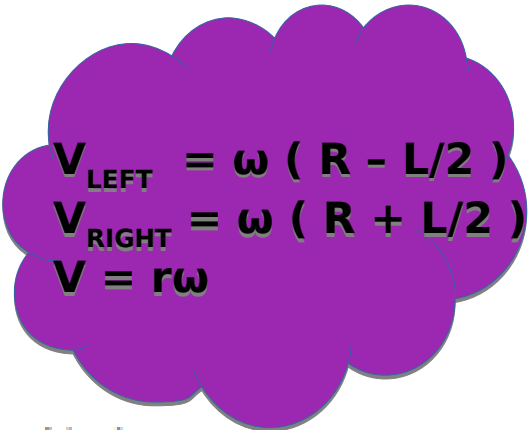
How to Implement?



How to Implement?

Robot Velocity to Wheel Velocity

```
void twistCallback( const geometry_msgs::Twist& cmd_msg) {  
    double v = cmd_msg.linear.x;  
    double omega = cmd_msg.angular.z;  
  
    // Velocity to left velocity and right velocity(wheel velocity)  
    double V_right = 1.0 * v + (omega * track_width / 2);  
    double V_left  = 1.0 * v - (omega * track_width / 2);  
  
    // ms to rpm  
    V_right = V_right * 60 / (pi * wheel_diameter);  
    V_left  = V_left  * 60 / (pi * wheel_diameter);  
}
```


$$\begin{aligned}V_{\text{LEFT}} &= \omega (R - L/2) \\V_{\text{RIGHT}} &= \omega (R + L/2) \\V &= r\omega\end{aligned}$$

How to Implement?

Forward Kinematic

Average_RPM = (left_rpm+right_rpm) / 2

Average_Velocity = (Average_RPM* π *wheel_diameter) / 60

Linear_Displacement = Average_Velocity * δt

$$\omega = (V_{\text{RIGHT}} - V_{\text{LEFT}}) / L$$

```
linear_displacement = (right_act_rpm+left_act_rpm)*dt*wheel_diameter*pi/(60*2);
```

```
//angular_velocity = (right_act_rpm-left_act_rpm)/track_width;
```

```
//angular_displacement = angular_velocity*dt*wheel_diameter*pi/60;
```

```
angular_displacement = (right_act_rpm-left_act_rpm)*dt*wheel_diameter*pi/(60*track_width);
```

rev/min.m x 2 π m/1rev x 1 min/60s

```
x = cos(dth) * linear_displacement;
```

```
y = -sin(dth) * linear_displacement;
```

```
delta_x = (cos(theta) * x - sin(theta) * y);
```

```
delta_y = (sin(theta) * x + cos(theta) * y);
```

```
delta_theta = dth;
```

```
x_pos += delta_x;
```

```
y_pos += delta_y;
```

```
theta += delta_theta;
```

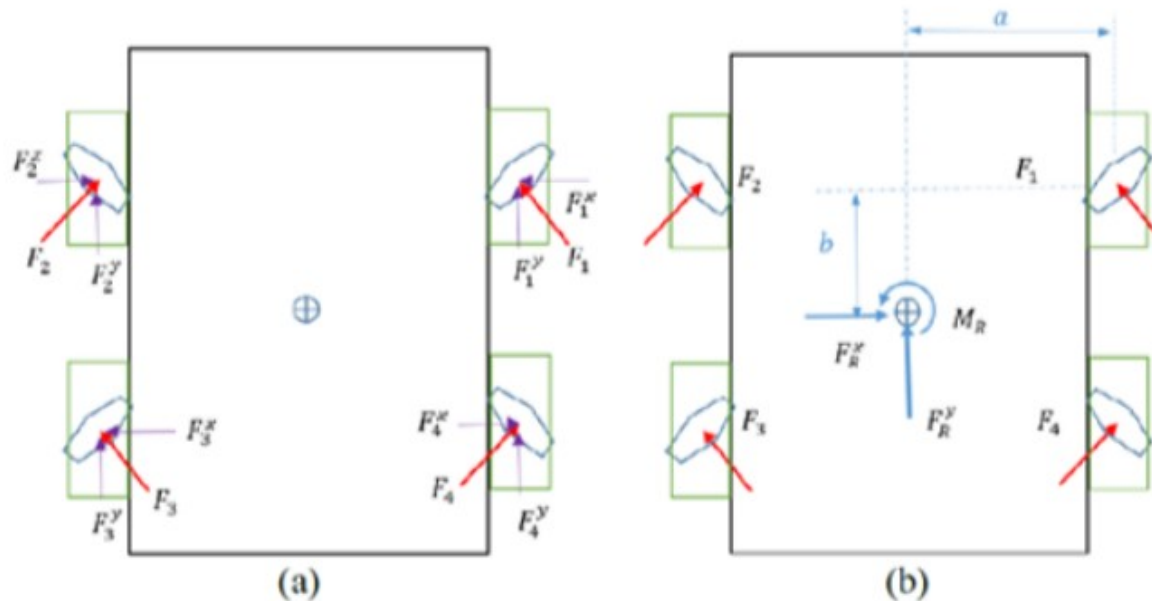
$$\begin{aligned} X' &= x.\cos(\omega.\delta t) - y.\sin(\omega.\delta t) \\ Y' &= x.\sin(\omega.\delta t) + y.\cos(\omega.\delta t) \\ \theta' &= \omega.\delta t \end{aligned}$$

Differential Drive Inverse Kinematic

To be Continue.

Mecanum Drive Kinematic

Kinematic for Mecanum Drive



Skip

ပုံကတော့ mecanum wheel မှာတပ်ဆင်ထားတဲ့ rollers မှာသက်ရောက်နေတဲ့ force ကိုပြထားတာဖြစ်ပါတယ်။ wheel 1 နှင့် 3 က right-handed ဖြစ်ပြီး 2 နှင့် 4 ကတော့ left-handed ဖြစ်နေတာကို ပုံမှာတွေ့နိုင်ပါတယ်။ ပုံ (a) အရ robot's center ၏ output ဖြစ်တဲ့ force နှင့် moment က wheel တစ်ခုစီကထုတ်ပေးတဲ့ force နှင့်အောက်ပါတိုင်း တူညီနေပါတယ်။

$$F_R^x = -F_1^x + F_2^x - F_3^x + F_4^x \quad (1)$$

$$F_R^y = F_1^y + F_2^y + F_3^y + F_4^y$$

Thank you!

