Difference Between Robot Motion Planning, UAV Navigation and Vehicle Driving

Benchun Zhou

School of Automation Science and Electronic Engineering
Beihang University

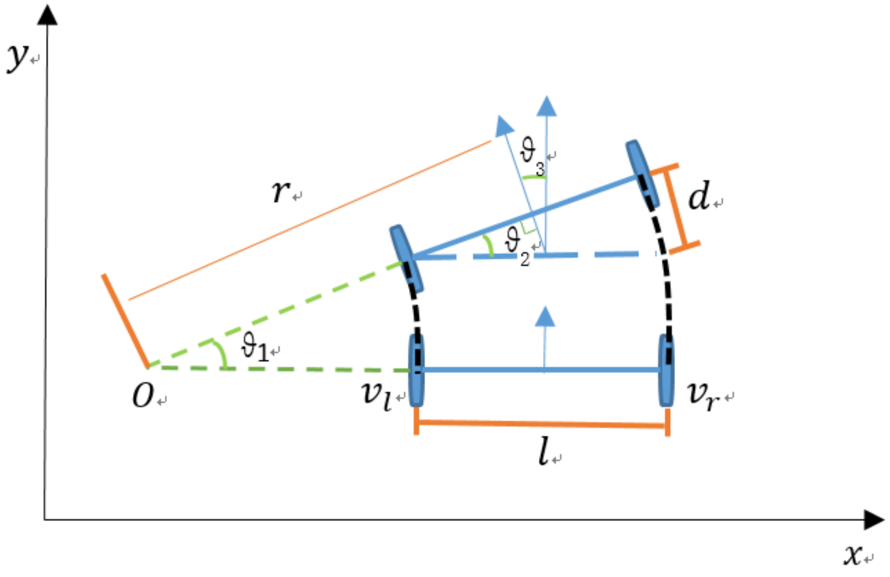
- **Framework**
- Kinematic model
- Action space
- Application

CarSim

Turtlebot

AR drone

1. Kinematic Model



- **Mobile Robot**
- Linear velocity of robot

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

- Geometry

$$\theta_3 = \theta_2 = \theta_1$$

$$\theta_2 \approx \sin(\theta) = \frac{d}{l} = \frac{(v_r - v_l)\Delta t}{l}$$

- Angular velocity

$$\omega = \frac{\theta_1}{\Delta t} = \frac{(v_r - v_l)}{l}$$

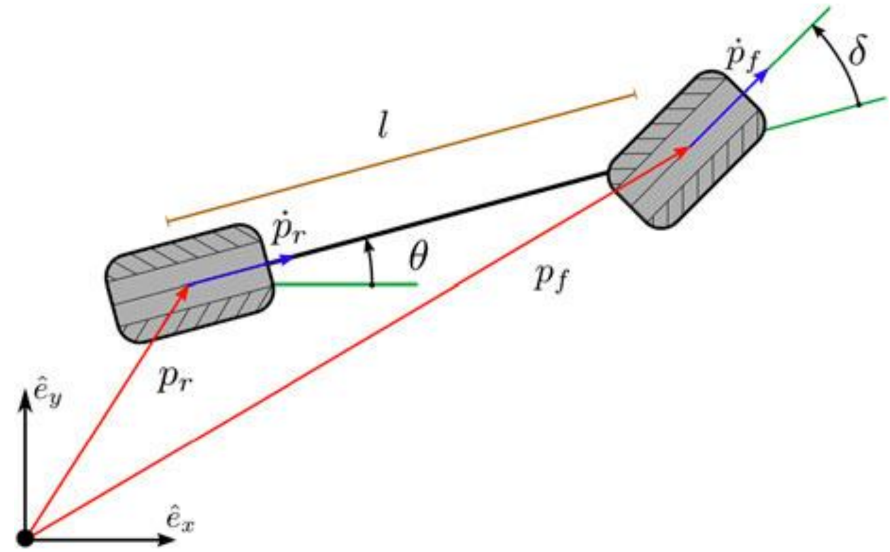
- Kinematic model

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

$$x_{t+1} = x_t + v\Delta t \cos\theta_t$$
$$y_{t+1} = y_t + v\Delta t \sin\theta_t$$
$$\theta_{t+1} = \theta_t + \omega\Delta t$$

1. Kinematic Model

- **Vehicles**



- Velocity of rear wheel (X_r, Y_r)

$$v_r = \dot{X}_r \cos\varphi + \dot{Y}_r \sin\varphi$$

- Constraints of front and rear wheel

$$\begin{cases} \dot{X}_f \sin(\varphi+\delta_f) - \dot{Y}_f \cos(\varphi+\delta_f) = 0 \\ \dot{X}_r \sin\varphi - \dot{Y}_r \cos\varphi = 0 \end{cases}$$

$$\Rightarrow \begin{cases} \dot{X}_r = v_r \cos\varphi \\ \dot{Y}_r = v_r \sin\varphi \end{cases}$$

- Geometry of the front and rear wheel

$$\begin{cases} X_f = X_r + l\cos\varphi \\ Y_f = Y_r + l\sin\varphi \end{cases}$$

$$\Rightarrow \omega = \frac{v_r}{l}\tan\delta$$

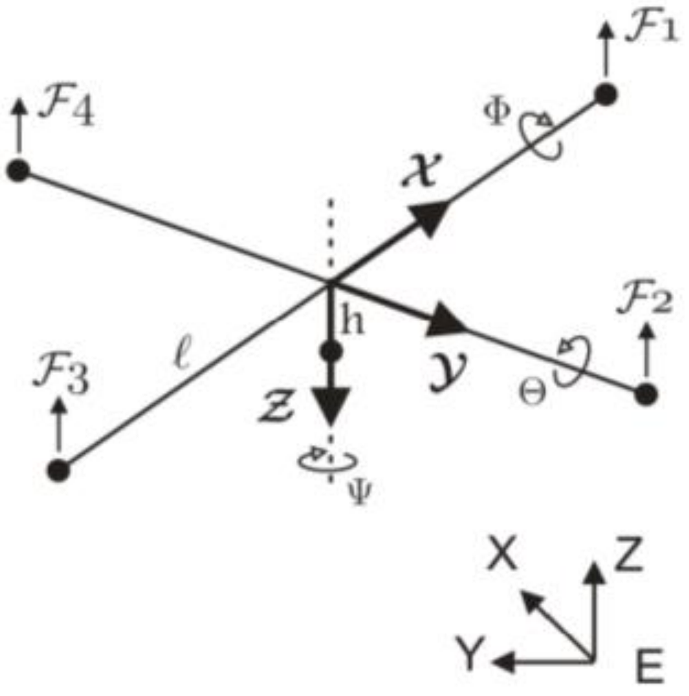
$$\begin{cases} R = v_r/\omega \\ \delta_f = \arctan(l/R) \end{cases}$$

- Kinematic model

$$\begin{bmatrix} \dot{X}_r \\ \dot{Y}_r \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi \\ \sin\varphi \\ \tan\delta/l \end{bmatrix} v_r$$

1. Kinematic Model

- **UAV**



$$
\begin{cases}
\ddot{x} = \dfrac{1}{m}(\cos\psi\sin\theta\cos\varphi + \sin\psi\sin\varphi)U_1 \\
\ddot{y} = \dfrac{1}{m}(\sin\psi\sin\theta\cos\varphi - \sin\psi\sin\varphi)U_1 \\
\ddot{z} = \dfrac{1}{m}(\cos\theta\cos\varphi)U_1 - g \\
\ddot{\varphi} = \dot{\theta}\dot{\psi}\dfrac{I_y - I_z}{I_x} - \dfrac{J}{I_x}\dot{\theta}\Omega + \dfrac{l}{I_x}U_2 \\
\ddot{\theta} = \dot{\varphi}\dot{\psi}\dfrac{I_z - I_x}{I_y} - \dfrac{J}{I_y}\dot{\varphi}\Omega + \dfrac{l}{I_y}U_3 \\
\ddot{\psi} = \dot{\varphi}\dot{\theta}\dfrac{I_x - I_y}{I_z} + \dfrac{l}{I_z}U_4
\end{cases}
$$

2. Action Space

- **Comparison**

- Mobile robot: linear velocity, angular velocity $\begin{bmatrix} v \\ \omega \end{bmatrix}$

- Vehicles: linear velocity, steering angle $\begin{bmatrix} v \\ \delta \end{bmatrix}$

- UAV: force in each motor

$$\begin{cases} U_1 = F_1 + F_2 + F_3 + F_4 \\ U_2 = F_4 - F_2 \\ U_3 = F_1 - F_3 \\ U_4 = F_1 - F_2 + F_3 - F_4 \end{cases}$$

3. Application

- **Comparison**
- Mobile robots: indoor navigation
- Vehicle: urban environment
- UAV: clutter environment

4. Navigation

- **Comparison**
- UAV navigation:
- Mobile robot navigation:
- The UAV needs to process amount of sensors' information in real time in order to fly safely and steady, especially for image processing which greatly increase the computational complexity. So it has become a major challenge a UAV to navigate under constraints of low power consumption and limited computing resources.
- UAV navigation requires a global or local 3D map of the environment; extra dimension means greater computation and storage consumption. So there is a great challenge when a UAV is navigating in a largescale environment for a long time.

THANKS YOU!