EKF: KF假設 linear

EKF是將實際上非線性用泰勒展開

做線性化使其能用KF理論

$x_t = g(u_t, x_{t-1}) + \varepsilon_t$

$z_t = h(x_t) + \delta_t$

1: **Algorithm Extended_Kalman_filter**$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$**:**

2: $\bar{\mu}_t = g(u_t, \mu_{t-1})$

3: $\bar{\Sigma}_t = G_t\ \Sigma_{t-1}\ G_t^T + R_t$

4: $K_t = \bar{\Sigma}_t\ H_t^T (H_t\ \bar{\Sigma}_t\ H_t^T + Q_t)^{-1}$

5: $\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$

6: $\Sigma_t = (I - K_t\ H_t)\ \bar{\Sigma}_t$

7: *return* μ_t, Σ_t

程序 3.3　EKF 算法

	KF	EKF
状态预测（第 2 行）	$\boldsymbol{A}_t \boldsymbol{\mu}_{t-1} + \boldsymbol{B}_t \boldsymbol{u}_t$	$g(\boldsymbol{u}_t, \boldsymbol{\mu}_{t-1})$
测量预测（第 5 行）	$\boldsymbol{C}_t \bar{\boldsymbol{\mu}}_t$	$h(\bar{\boldsymbol{\mu}}_t)$

在mean處做泰勒展開

$$g(\boldsymbol{u}_t, \boldsymbol{x}_{t-1}) \approx g(\boldsymbol{u}_t, \boldsymbol{\mu}_{t-1}) + \underbrace{g'(\boldsymbol{u}_t, \boldsymbol{\mu}_{t-1})}_{=:G_t}(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_{t-1})$$

$$= g(\boldsymbol{u}_t, \boldsymbol{\mu}_{t-1}) + \boldsymbol{G}_t(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_{t-1})$$

$$g'(\boldsymbol{u}_t, \boldsymbol{x}_{t-1}) := \frac{\partial g(\boldsymbol{u}_t, \boldsymbol{x}_{t-1})}{\partial \boldsymbol{x}_{t-1}}$$

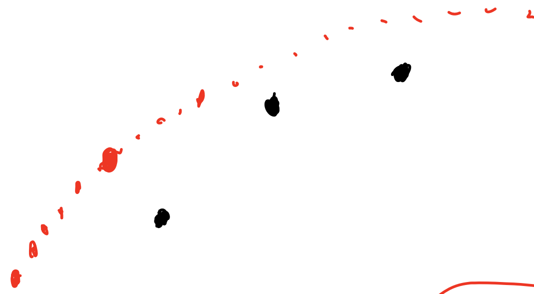
$$h(\boldsymbol{x}_t) \approx h(\bar{\boldsymbol{\mu}}_t) + \underbrace{h'(\bar{\boldsymbol{\mu}}_t)}_{=:\boldsymbol{H}_t}(\boldsymbol{x}_t - \bar{\boldsymbol{\mu}}_t)$$

$$= h(\bar{\boldsymbol{\mu}}_t) + \boldsymbol{H}_t(\boldsymbol{x}_t - \bar{\boldsymbol{\mu}}_t)$$

$$h'(\boldsymbol{x}_t) = \frac{\partial h(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t},$$

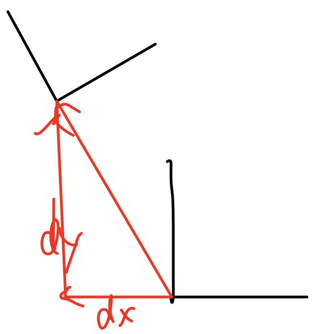
G_t, H_t: Jacobian matrix

會隨每一時刻改變



$X_t = \boxed{X_{t-1} + u}$

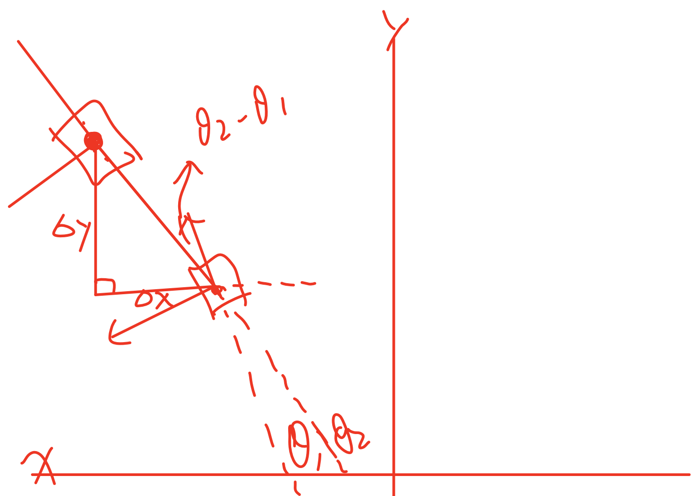
1 0 0

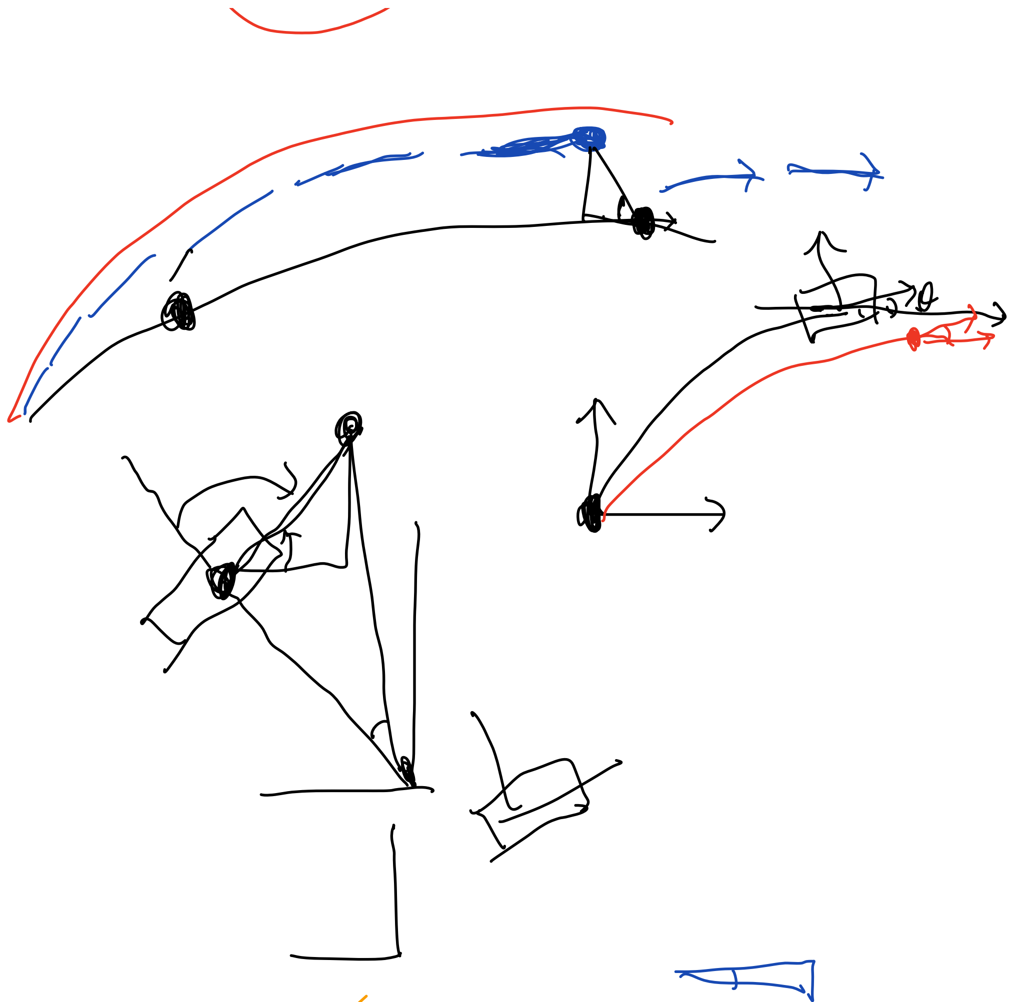


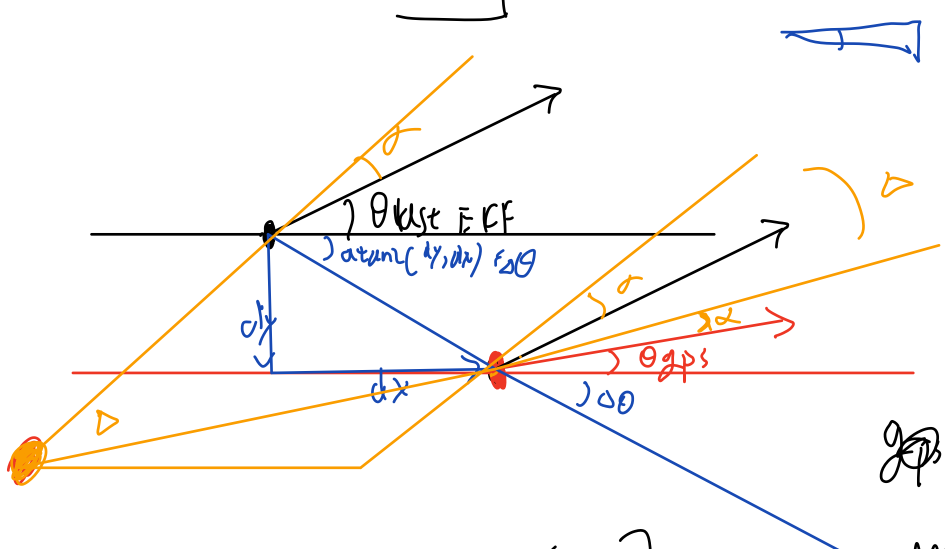
$P_B = P_{B/A} + P_A$

R

$$\begin{matrix} \cos\theta & -\sin\theta & \\ \sin\theta & \cos\theta & \\ 0 & 0 & 1 \end{matrix}$$







$$\begin{bmatrix} gpsx \\ gpsy \\ gpsyaw \end{bmatrix} = \begin{matrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{matrix} \begin{bmatrix} x \\ y \\ yaw \end{bmatrix}$$

gpsYaw
= yaw + atan(y - gpsy,
(x - gpsx)

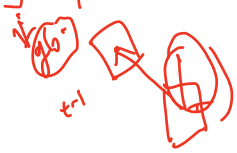
z_t

S_{t-1} $\qquad S_t$

δ

$$S_t = T \cdot S_{t-1}$$

$$\underset{3\times 1}{\begin{bmatrix} x_t \\ y_t \\ \alpha_t \end{bmatrix}} = \underset{3\times 3}{\begin{bmatrix} \; & \; & \; \end{bmatrix}} \underset{\times\, 3\times 3}{\begin{bmatrix} x_{t-1} + \Delta x \\ y_{t-1} + \Delta y \\ \alpha_{t-1} + \Delta \end{bmatrix}}$$



$-\theta$

$$\epsilon_t = \cdots x_t$$

$$\begin{bmatrix} 1 & 0 & 0 & x \\ 0 & & & y \\ 0 & & & z \\ 0 & & & 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & x_{t-1} \\ \sin(\alpha) & \cos(\alpha) & y_{t-1} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ 1 \end{bmatrix}$$

$$x_t = \underline{\cos\alpha\, \Delta x - \sin(\alpha)\, \Delta y + x_{t-1}} \to f_1$$

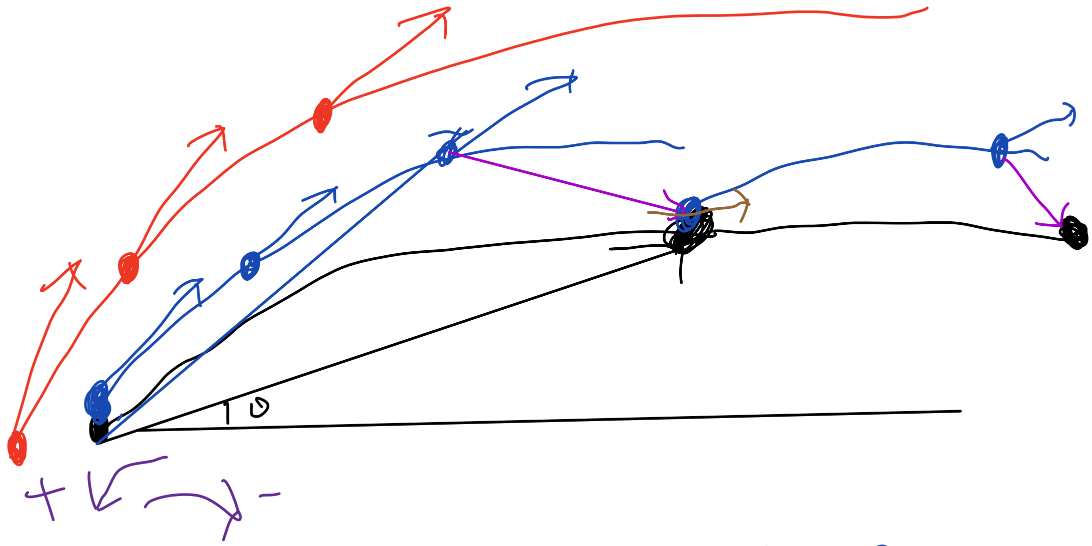
$$y_t = \underline{\sin\alpha\, \Delta x + \cos(\alpha)\, \Delta y + y_{t-1}} \to f_2$$

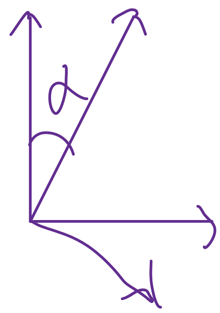
$$\alpha_t = \underline{\alpha_{t-1} + \Delta\alpha} \to f_3$$

$$\alpha = \alpha_{t-1}$$

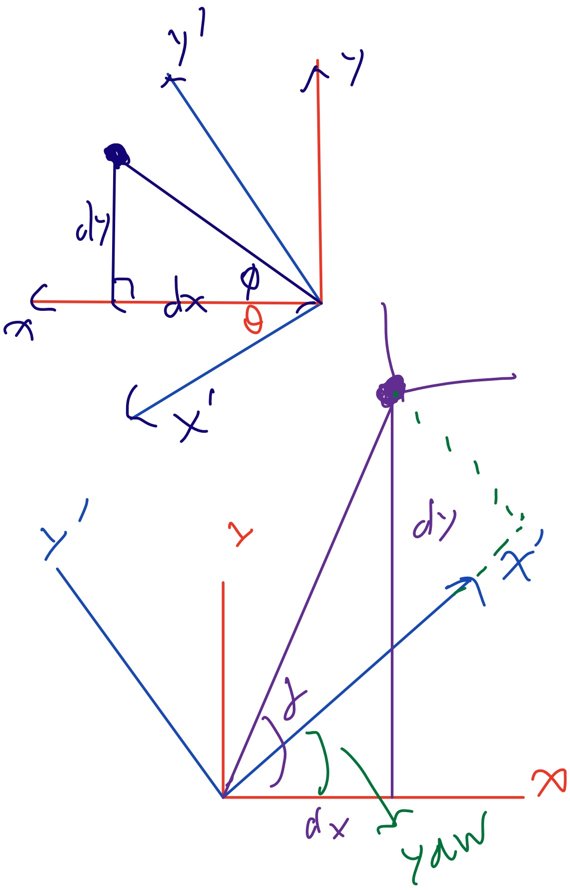
$$\underset{X_t}{\begin{bmatrix} x_t \\ y_t \\ \alpha_t \end{bmatrix}} = \underset{X_{t-1}}{\begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \alpha_{t-1} \end{bmatrix}} + \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \underset{u}{\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\alpha \end{bmatrix}}$$

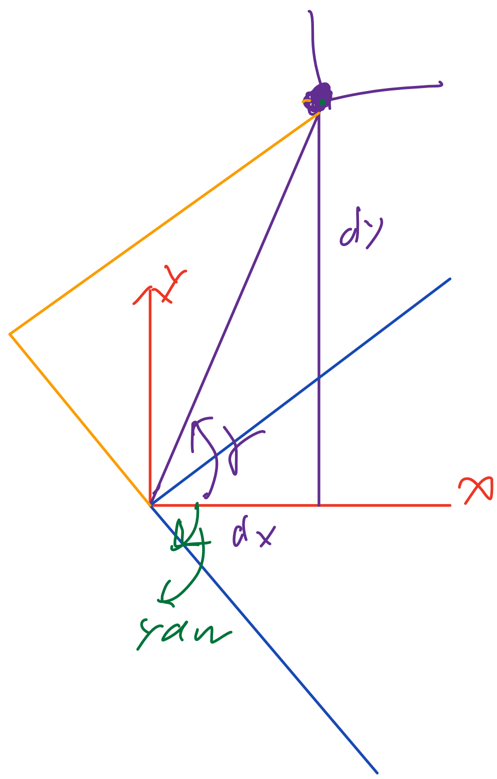
$$G = \begin{bmatrix} \frac{\partial f_1}{\partial x_{t-1}} & \frac{\partial f_1}{\partial y_{t-1}} & \frac{\partial f_1}{\partial \alpha_{t-1}} \\ \frac{\partial f_2}{\partial x_{t-1}} & \frac{\partial f_2}{\partial y_{t-1}} & \frac{\partial f_2}{\partial \alpha_{t-1}} \\ \frac{\partial f_3}{\partial x_{t-1}} & \frac{\partial f_3}{\partial y_{t-1}} & \frac{\partial f_3}{\partial \alpha_{t-1}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -(\sin\alpha)\Delta x - (\cos\alpha)\Delta y \\ 0 & 1 & (\cos\alpha)\Delta x - (\sin\alpha)\Delta y \\ 0 & 0 & 1 \end{bmatrix}$$



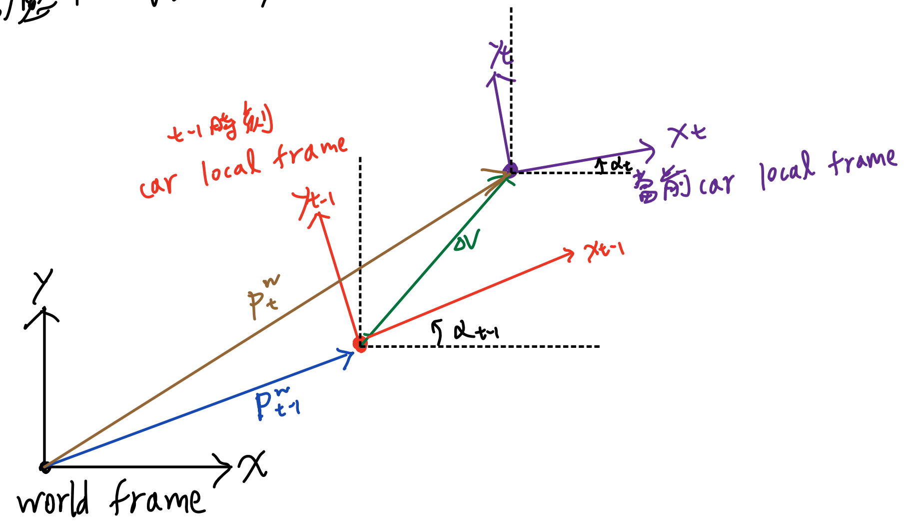


$$\text{gpsyaw} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ & & \end{bmatrix} \begin{array}{l} x \\ y \\ \alpha \\ \text{yaw} \end{array}$$


y'
y
x
dy
dx
ϕ
θ
x'
y'
y
dy
x'
γ
x
dx
yaw


y
dy
γ
x
dx
yaw

考慮俯視下，二維平面



α_{t-1} : t-1時刻, local frame yaw

α_t : t 時刻, local frame yaw

ΔV : 表 local frame t-1 原點相對 local frame t 原點之向量

P_t^w : local frame t 之原點在 world frame 下之表示

P_{t-1}^w : local frame t-1 之原點在 world frame 下之表示

* goal : find transformation of P_w^{t-1} to P_w^t

由向量加法可知， $P_t^w = P_{t-1}^w + \Delta V^w$

ΔV^w 表向量 ΔV 在 world frame 下之表示

其中，選擇 ΔV^{t-1} 可當做兩筆 odometry data 之變化量當作 control

丟入 EKF 之 prediction step 中

ΔV^{t-1} 表 ΔV 在 local frame t-1 下之表示

EKF中， P_w^{t-1} 是上一次儲存之 pose

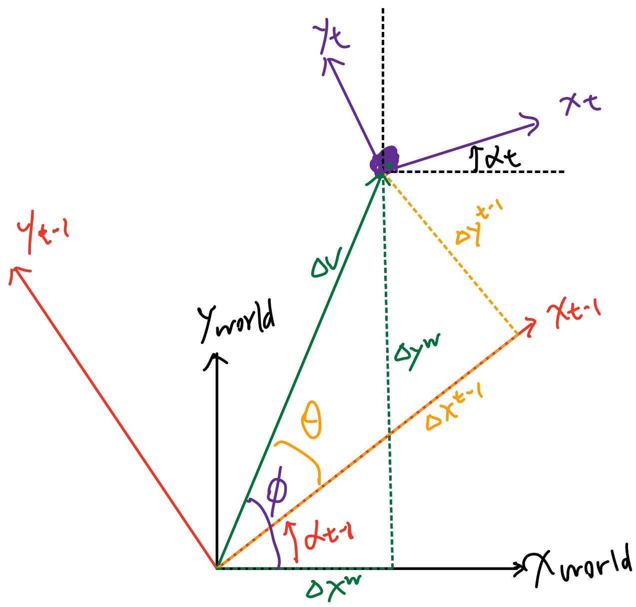
，藉 ΔV^{t-1} control to predict P_t^w (pose at t)

* global change → local change

將原得到的 data 相減，表示的是相差之量在 world frame 下之表示

也就是 ΔV^w，但需轉為相對 t-1 時 local frame 下的改變量再傳入 EKF

以下為原理推導：



α_{t-1} is yaw of local frame t-1 (已知)

將當前 odometry data 和 t-1 時 odometry data 相減

$\Rightarrow$ 得 Δx^w, Δy^w

$\because \phi = atan2(\Delta y^w, \Delta x^w)$

$\therefore \theta = \phi - \alpha_{t-1}$

$$||\Delta V|| = \sqrt{(\Delta x^w)^2 + (\Delta y^w)^2}$$

$$\Rightarrow \begin{array}{l} \Delta x^{t-1} = ||\Delta V|| \times \cos\theta \\ \Delta y^{t-1} = ||\Delta V|| \times \sin\theta \end{array}$$

而 $\Delta yaw = \alpha_t - \alpha_{t-1}$

因此，$u=\begin{bmatrix}\Delta x^{t-1}\\ \Delta y^{t-1}\\ \Delta Yaw\end{bmatrix}$，表時刻t下之pose在local frame t-1下之表示
當作 control 送入EKF

* EKF's motion model

EKF之 state $\mathbf{X}$ 設為 pose，np.array([x, y, yaw])

如一開始之目標，將 $P_t^w = P_{t-1}^w + \Delta V^w$

$$\mathbf{X}_t=\begin{bmatrix}x_t\\ y_t\\ yaw_t\end{bmatrix},\quad \mathbf{X}_{t-1}=\begin{bmatrix}x_{t-1}\\ y_{t-1}\\ yaw_{t-1}\end{bmatrix},\quad P_t^w=\begin{bmatrix}x_t\\ y_t\end{bmatrix},\quad P_{t-1}^w=\begin{bmatrix}x_{t-1}\\ y_{t-1}\end{bmatrix}$$

control：$u=\begin{bmatrix}\Delta x^{t-1}\\ \Delta y^{t-1}\\ \Delta Yaw\end{bmatrix}$　　其中 $\Delta V^{t-1}=\begin{bmatrix}\Delta x^{t-1}\\ \Delta y^{t-1}\end{bmatrix}$，表t時刻下 x_t 和 y_t 在 local frame t-1下之表示

已知 car local frame 和 world frame 是旋轉+平移關係

⇒ 兩座標間轉換可用 transform matrix T_{t-1}^w 表示

$$P_t^w = T_{t-1}^w \Delta V^{t-1},\quad T_{t-1}^w=\begin{bmatrix}R_{t-1}^w & P_{t-1}^w\\ 0^T & 1\end{bmatrix}$$

$\alpha = Yaw_{t-1}$

$$\Rightarrow \begin{bmatrix}x_t\\ y_t\\ 1\end{bmatrix}=\begin{bmatrix}\cos\alpha & -\sin\alpha & x_{t-1}\\ \sin\alpha & \cos\alpha & y_{t-1}\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\Delta x^{t-1}\\ \Delta y^{t-1}\\ 1\end{bmatrix}$$

$$\Rightarrow \begin{aligned} x_t &= x_{t-1} + \cos\alpha\,\Delta x^{t-1} - \sin\alpha\,\Delta y^{t-1} \longrightarrow f_1\\ y_t &= y_{t-1} + \sin\alpha\,\Delta x^{t-1} + \cos\alpha\,\Delta y^{t-1} \longrightarrow f_2\end{aligned}$$

而 Yaw 部分：$Yaw_t = Yaw_{t-1} + \Delta Yaw \longrightarrow f_3$

則 motion model 可整理成：

$$\begin{bmatrix} x_t \\ y_t \\ Yaw_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ Yaw_{t-1} \end{bmatrix} + \begin{bmatrix} \cos\alpha & -\sin\alpha & \\ \sin\alpha & \cos\alpha & \\ 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} \Delta x^{t-1} \\ \Delta y^{t-1} \\ \Delta Yaw \end{bmatrix}}_{= u}$$

此即我的 motion model

接著需要 $\underbrace{\text{Jacobian matrix}}_{\text{存在 self.A}}$ to predict $\underbrace{\text{state covariance}}_{\text{self.S}}$ matrix

$$\text{Jacobian matrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_{t-1}} & \frac{\partial f_1}{\partial y_{t-1}} & \frac{\partial f_1}{\partial \alpha} \\ \frac{\partial f_2}{\partial x_{t-1}} & \frac{\partial f_2}{\partial y_{t-1}} & \frac{\partial f_2}{\partial \alpha} \\ \frac{\partial f_3}{\partial x_{t-1}} & \frac{\partial f_3}{\partial y_{t-1}} & \frac{\partial f_3}{\partial \alpha} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & (-\sin\alpha)\Delta x^{t-1} - (\cos\alpha)\Delta y^{t-1} \\ 0 & 1 & (\cos\alpha)\Delta x^{t-1} - (\sin\alpha)\Delta y^{t-1} \\ 0 & 0 & 1 \end{bmatrix}$$

$\Rightarrow$ state covariance prediction：

$$\text{self.S} = (\text{self.A})(\text{self.S})(\text{self.A})^T + \text{self.R}$$