

1. Page 14:

Pure Pursuit Control for WMR

- Concept:
 - Modify the angular velocity to let the center achieve a point on path

$$\alpha = \arctan\left(\frac{y - y_g}{x - x_g}\right) - \theta$$

$$\omega = \frac{2 \sin(\alpha)}{L_d} \quad \leftarrow \quad \frac{2v \sin(\alpha)}{L_d}$$

$L_d = (kv + L_{fc})$, where k, L_{fc} are parameters.

1. Set a distance **Ld**.
2. Find the nearest point on the path.
3. Search the following point until the distance of the point larger than or equal to **Ld**.

14

2. Page 14:

Pure Pursuit Control for WMR

- Concept:
 - Modify the angular velocity to let the center achieve a point on path

$$\alpha = \arctan\left(\frac{y - y_g}{x - x_g}\right) - \theta \quad \leftarrow \quad \left(\frac{y_g - y}{x_g - x}\right) - \theta$$

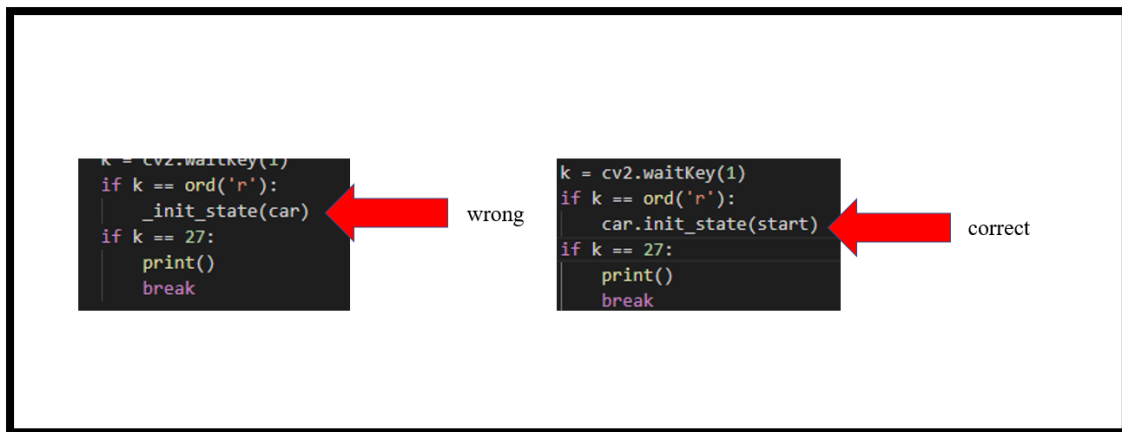
$$\omega = \frac{2 \sin(\alpha)}{L_d}$$

$L_d = (kv + L_{fc})$, where k, L_{fc} are parameters.

1. Set a distance **Ld**.
2. Find the nearest point on the path.
3. Search the following point until the distance of the point larger than or equal to **Ld**.

14

3. initial function in all of the code: **initial function usage** in some file is wrong



Start may like this: (x, y, yaw)

```
start = (50, 300, 0)
```

4. Page 16

Stanley Control


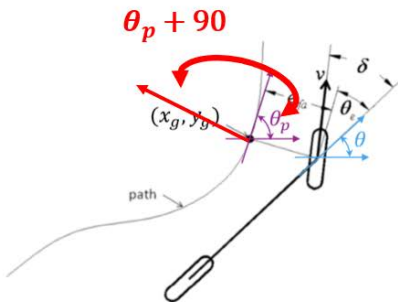
- Concept:
 - Exponential stability for front wheel feedback
- Some Implementation Details

$$\theta_e = \theta_p - \theta$$

$$\dot{e} = v_f \sin(\delta - \theta_e)$$

$$\delta = \arctan\left(-\frac{ke}{v_f}\right) + \theta_e$$

$$e = \begin{bmatrix} x - x_g \\ y - y_g \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta_p) \\ \sin(\theta_p) \end{bmatrix} \Rightarrow \theta_p + 90$$

16

5. Before use the **yaw**, you need to transform the yaw's **degree to radian**

After calculate the angle you need to transform back

Something like this:

```
alpha = np.arctan2(target[1]-y, target[0]-x) - np.deg2rad(yaw)
next_w = np.rad2deg(2*v*np.sin(alpha) / Ld)
```