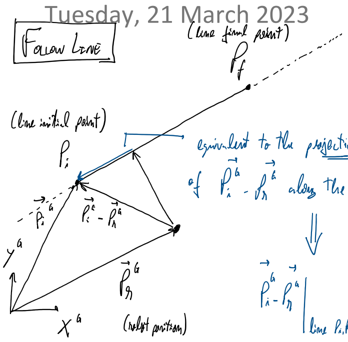
pileec2223_move_robot_followline

Tuesday, 21 March 2023 14:13

Follow Line



Line: $\vec{x} = \vec{P}_i + t\,n$, where t moves along the line, $\vec{x}$ is a point along the line

$\Downarrow$

$$\text{distance}\left(\vec{x}, \vec{P}_r^G\right) = \text{distance}\left(\vec{P}_i + t\,n, \vec{P}_r^G\right)$$

equivalent to the projection of $\vec{P}_i^G - \vec{P}_r^G$ along the direction of the line

$\Downarrow$

$$\vec{P}_i^G - \vec{P}_r^G \Big|_{\text{line } P_i P_f} = \left(\vec{P}_i^G - \vec{P}_r^G\right) \bullet \left(\frac{\vec{P}_i - \vec{P}_f}{\|\vec{P}_i - \vec{P}_f\|}\right)$$

" unit vector along the direction of the line P_i, P_f

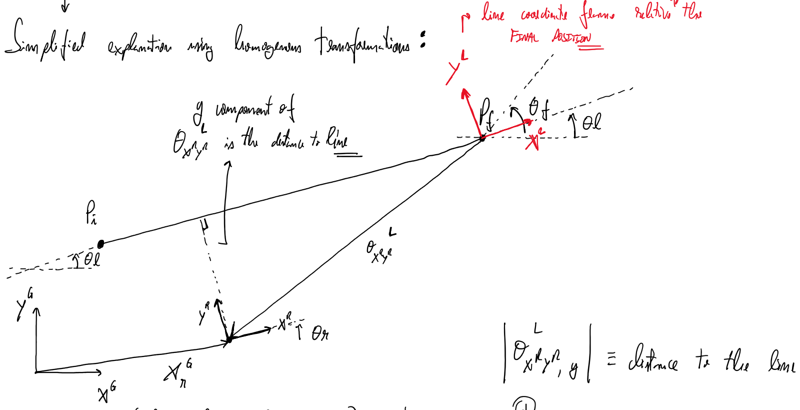
dot product

$$\left(\vec{a} \bullet \vec{b} = \|\vec{a}\| \cdot \|\vec{b}\| \cdot \cos\theta_{\vec{a},\vec{b}}\right)$$

$\Downarrow$

$$\text{distance to line} = \left\| \left(\vec{P}_i^G - \vec{P}_r^G\right) - \left(P_i^G - P_r^G\right) \bullet \left(\frac{\vec{P}_i - \vec{P}_f}{\|\vec{P}_i - \vec{P}_f\|}\right) \right\|$$

Simplified explanation using homogeneous transformations:



$$\left| O_{x^R y^R, y}^L \right| \equiv \text{distance to the line}$$

$$\theta_l = \operatorname{atan2}\left(P_{f,y} - P_{i,y},\ P_{f,x} - P_{i,x}\right)$$

$$O_{x^R y^R}^R = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad R_R^G = R(\theta_r) \qquad T_R^G$$

$$H_R^G = \begin{bmatrix} c\theta_r & -s\theta_r & X_{r,x}^G \\ s\theta_r & c\theta_r & X_{r,y}^G \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Longrightarrow O_{x^R y^R}^L = H_G^L \cdot H_R^G \cdot O_{x^R y^R}^R = H_G^L \cdot H_R^G \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$H_L^G = \begin{bmatrix} c\theta_l & -s\theta_l & P_{f,x} \\ s\theta_l & c\theta_l & P_{f,y} \\ 0 & 0 & 1 \end{bmatrix} \longrightarrow H_G^L = \begin{bmatrix} R^{-1} & -R^{-1} \cdot T \\ 0_{1\times 2} & 1 \end{bmatrix} = \begin{bmatrix} c\theta_l & s\theta_l & -c\theta_l \cdot P_{f,x} - s\theta_l \cdot P_{f,y} \\ -s\theta_l & c\theta_l & s\theta_l \cdot P_{f,x} - c\theta_l \cdot P_{f,y} \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_L^G = R(\theta_l) \qquad T_L^G$$

Current implementation in RDF:

$\theta_{aux} = G_{DIST} \cdot \text{side line} \cdot \text{dist2line}$ → you may add here a derivative gain for dist2line

// relative to line

$v_l = \|v_{nom}\| \cdot c\,\theta_{aux}$ → if $\theta_{aux} = 0$, $\begin{cases} v_l = \|v_{nom}\| \\ v_{nl} = 0 \end{cases}$ ⇒ exactly what we want!

$v_{nl} = \|v_{nom}\| \cdot s\,\theta_{aux}$

// robot velocities

$v = v_l \cdot c_{\theta_l - \theta_r} - v_{nl} \cdot s_{\theta_l - \theta_r}$

$v_n = v_l \cdot s_{\theta_l - \theta_r} + v_{nl} \cdot c_{\theta_l - \theta_r}$