移动机器人导论 HW1

> 任云帆
>
> 17自动化1班

题目要求：

设计基于极坐标的控制器实现叉车定点控制。

叉车运动学

$$\dot{\xi}_R = \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{bmatrix} = r\dot{\phi} \begin{bmatrix} \sin\beta \\ 0 \\ -\frac{\cos\beta}{L_2} \end{bmatrix}$$

基于极坐标的控制器设计

叉车系统的输入为

$$u = \begin{bmatrix} v_1 \\ b \end{bmatrix} \in \mathbb{R}^2$$

分别表示主动轮的速度和角度,根据运动学模型可以将输入表示为机器人的速度v_1和角速度w_1。

因此控制器的输入为平面笛卡尔坐标系位姿$[x_c, y_c, \theta_c]^T$，输出为机器人机身坐标系下速度集合角速度$[v1, w_1]^T$。误差表示为

$$e_c = \begin{bmatrix} x - x_{tar} \\ y - y_{tar} \\ \theta - \theta_{tar} \end{bmatrix}$$

根据笛卡尔坐标到极坐标的转换关系

$$\begin{cases} \rho = \sqrt{\Delta x^2 + \Delta y^2} \\ \alpha = \arctan 2(\Delta y, \Delta x) \\ \beta = -\theta - \alpha \end{cases}$$

求导得

$$\begin{cases} \dot{\rho} = \frac{\dot{\Delta x}\Delta x + \dot{\Delta y}\Delta y}{\rho} \\ \dot{\beta} = -\frac{\Delta x}{\rho}(\frac{\dot{y}}{\Delta x} - \frac{\dot{x}\Delta y}{\Delta x^2}) \\ \dot{\alpha} = -\dot{\beta} - \dot{\theta} \end{cases}$$
$$\Longrightarrow \begin{bmatrix} \dot{\rho} \\ \dot{\beta} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} -\cos\alpha & 0 \\ -\frac{\sin\alpha}{\rho} & 0 \\ \frac{\sin\alpha}{\rho} & -1 \end{bmatrix}$$

设计线性控制器

$$v_1 = k_\rho \rho$$
$$v_2 = k_\alpha \alpha + k_\beta \beta$$

代入闭环误差方程得

$$\begin{bmatrix} \dot{\rho} \\ \dot{\beta} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} -k_\rho \rho \cos\alpha \\ -k_\rho \sin\alpha \\ k_\rho \sin\alpha - k_\alpha \alpha - k_\beta \beta \end{bmatrix}$$

设计李雅普诺夫函数

$$V_1 = \frac{1}{2}\rho^2 + \frac{1}{2}\alpha^2 + \frac{1}{2}\beta^2$$
$$\dot{V}_1 = \rho\dot{\rho} + \alpha\dot{\alpha} + \beta\dot{\beta}$$

在$\alpha = 0$进行线性化，得到

$$\begin{bmatrix} \dot{\rho} \\ \dot{\beta} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} -k_\alpha & 0 & 0 \\ 0 & -(k_\alpha - k_\rho) & k_\beta \\ 0 & -k_\rho & 0 \end{bmatrix} \begin{bmatrix} \rho \\ \alpha \\ \beta \end{bmatrix}$$

$$V = -k_\alpha \rho^2 - (k_\alpha - k_\rho)\alpha\beta + k_\beta \beta^2 - k_\rho \alpha\beta$$

当参数满足

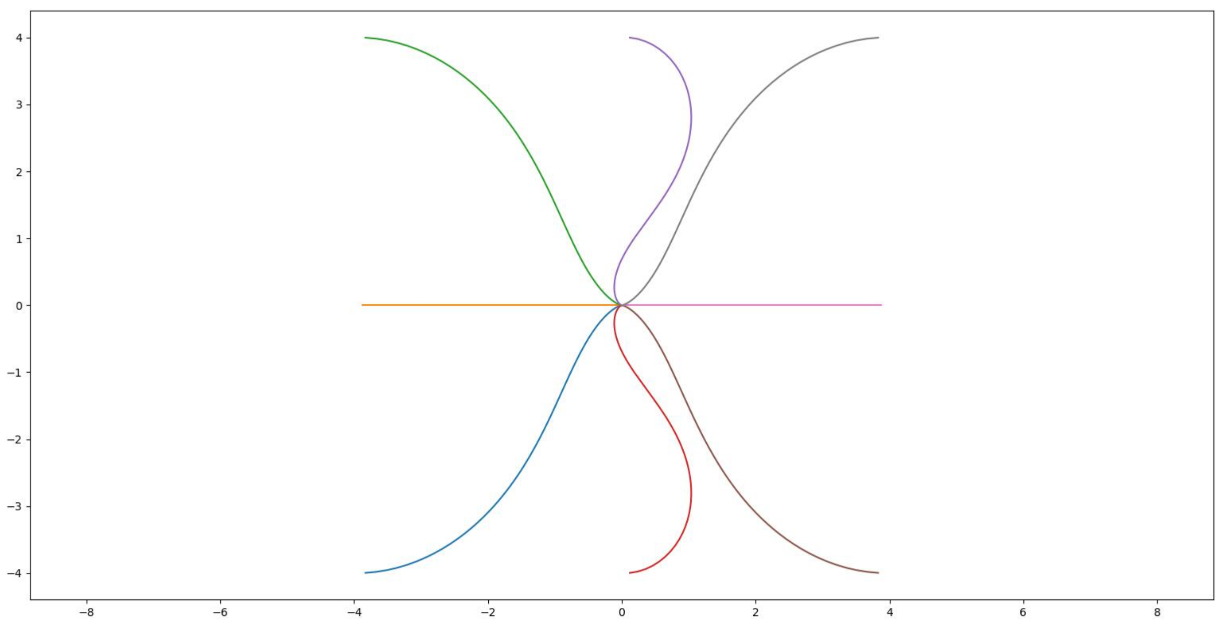
$$k_\beta \leq 0$$
$$k_\rho \geq 0$$
$$k_\alpha - k_\rho \geq 0$$

是系统局部稳定。

实验

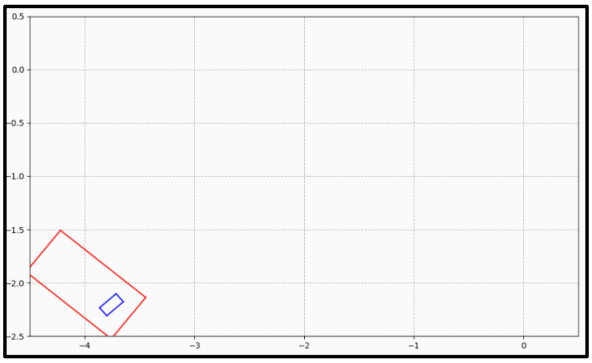
通过Python实现上述控制器，并使用Matplotlib进行仿真实验。代码见Github

实验1 八个方向出发

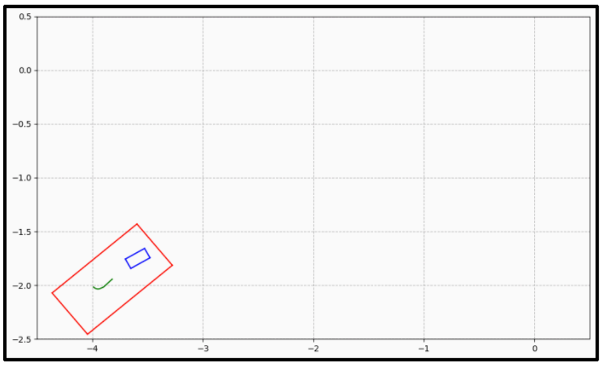


实验2 动画仿真

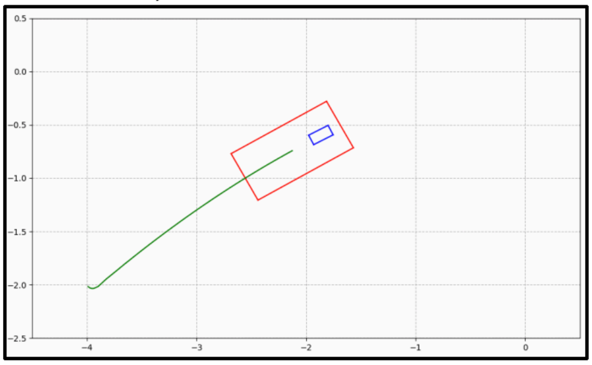
GIF动画见[Github]



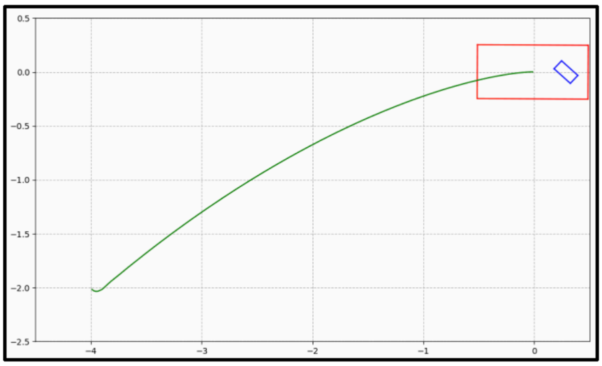
1)出发角度为 -60 °



2)



3)



4)目标角度为0 °