



(12)发明专利申请

(10)申请公布号 CN 108681243 A

(43)申请公布日 2018. 10. 19

(21)申请号 201810415570.5

(22)申请日 2018.05.03

(71)申请人 南京理工大学

地址 210094 江苏省南京市孝陵卫200号

(72)发明人 郭健 刘源 袁佳泉 李胜

吴益飞 洪宇

(74)专利代理机构 南京理工大学专利中心

32203

代理人 孟睿

(51)Int.Cl.

G05B 13/04(2006.01)

G05D 1/02(2006.01)

G05D 1/08(2006.01)

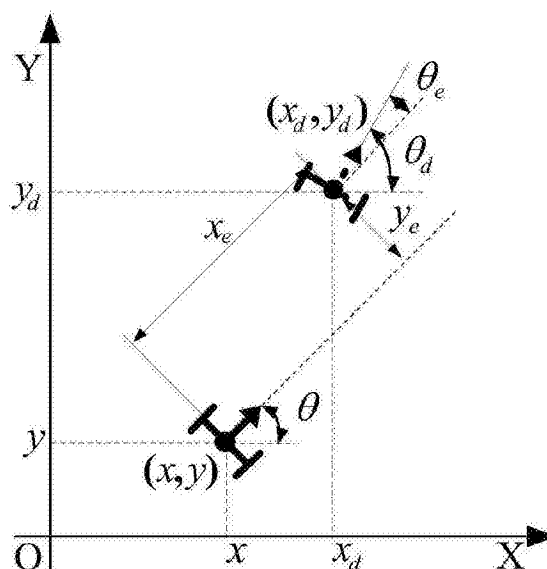
权利要求书2页 说明书6页 附图3页

(54)发明名称

一种机器人轨迹跟踪方法

(57)摘要

本发明涉及一种机器人轨迹跟踪方法。将机器人的离散运动轨迹作为跟踪轨迹,根据跟踪轨迹计算机器人坐标系中的机器人位姿误差,根据本发明设计的控制律计算跟踪速度,根据跟踪速度预估下一时刻的机器人全局位姿,然后计算出预估全局位姿与目标位姿的误差,根据用评价函数计算误差评价价值,寻找评价价值最低的速度作为当前时刻实际的跟踪速度。本发明基于Lyapunov函数构造控制律,能够同时跟踪机器人的位置和姿态;同时,本发明通过引入动态窗口确定参数,提高了跟踪的动态性能,跟踪曲线更贴合、更平滑。



1. 一种机器人轨迹跟踪方法,其特征在于,步骤如下:

步骤1,将机器人的离散运动轨迹作为跟踪轨迹 $(x_d(nT), y_d(nT), \theta_d(nT))$, 其中, T 为采样时间, $x_d(nT)$ 、 $y_d(nT)$ 和 $\theta_d(nT)$ 分别表示第 nT 时刻机器人离散运动轨迹中机器人位姿的横坐标、纵坐标和角度;

步骤2,按照下式计算机器人坐标系中的机器人位姿误差 $(x_e(nT), y_e(nT), \theta_e(nT))$,

$$\begin{cases} x_e(nT) = (x_d(nT) - x(nT)) \cos \theta(nT) + (y_d(nT) - y(nT)) \sin \theta(nT) \\ y_e(nT) = -(x_d(nT) - x(nT)) \sin \theta(nT) + (y_d(nT) - y(nT)) \cos \theta(nT) \\ \theta_e(nT) = \theta_d(nT) - \theta(nT) \end{cases}$$

其中, $x(nT)$ 、 $y(nT)$ 、 $\theta(nT)$ 分别为第 nT 时刻机器人在全局坐标下的横坐标、纵坐标和角度;

步骤3,按照下式所示的控制律,计算跟踪速度 $(v_j(nT), \omega_j(nT))$,

$$\begin{cases} v_j(nT) = v_d(nT) \cos \theta_e(nT) + x_e(nT) - \theta_e(nT) \omega_j(nT) \\ \omega_j(nT) = \omega_d(nT) + v_d(nT) [\alpha_j (y_e(nT) + \theta_e(nT)) + (1 - \alpha_j) \sin \theta_e(nT)] \end{cases};$$

其中, $v_d(nT)$ 和 $\omega_d(nT)$ 分别表示机器人在第 nT 时刻的线速度和角速度; $v_j(nT)$ 、 $\omega_j(nT)$ 分别为,在第 nT 时刻,机器人控制器参数为 α_j 时,控制器给出的线速度和角速度;设参数间隔 i , $0 < i < 1$, 以 i 为间隔遍历;设控制器参数 α , 将第 j 个 α 的取值记作 α_j , 即 $\alpha_1 = i, \alpha_2 = 2i, \dots$, 直到小于1的最大的 α 值;

步骤4,根据跟踪速度 $(v_j(nT), \omega_j(nT))$,按下式预估 $(nT+T)$ 时刻的机器人全局位姿 $(\tilde{x}_{gj}(nT+T), \tilde{y}_{gj}(nT+T), \tilde{\theta}_{gj}(nT+T))$,

$$\begin{cases} \tilde{x}_{gj}(nT+T) = x_g(nT) + v_j(nT)T \cos \theta_g(nT) \\ \tilde{y}_{gj}(nT+T) = y_g(nT) + v_j(nT)T \sin \theta_g(nT) \\ \tilde{\theta}_{gj}(nT+T) = \theta_g(nT) + \omega_j(nT)T \end{cases}$$

其中, $\tilde{x}_{gj}(nT+T)$ 、 $\tilde{y}_{gj}(nT+T)$ 、 $\tilde{\theta}_{gj}(nT+T)$ 分别为机器人在 $(nT+T)$ 时刻应当达到的全局坐标系下的横坐标、纵坐标和角度, $x_g(nT)$ 、 $y_g(nT)$ 、 $\theta_g(nT)$ 分别为 nT 时刻机器人的全局位姿,即机器人在 nT 时刻在全局坐标系下的实际横坐标、纵坐标和角度;

步骤5,按下式计算预估在当前控制器下,在 $(nT+T)$ 时刻机器人预估全局位姿与目标位姿的误差 $(\tilde{x}_{ej}(nT+T), \tilde{y}_{ej}(nT+T), \tilde{\theta}_{ej}(nT+T))$,

$$\begin{cases} \tilde{x}_{ej}(nT) = (x_d(nT) - \tilde{x}_{gj}(nT)) \cos \tilde{\theta}_{gj}(nT) + (y_d(nT) - \tilde{y}_{gj}(nT)) \sin \tilde{\theta}_{gj}(nT) \\ \tilde{y}_{ej}(nT) = -(x_d(nT) - \tilde{x}_{gj}(nT)) \sin \tilde{\theta}_{gj}(nT) + (y_d(nT) - \tilde{y}_{gj}(nT)) \cos \tilde{\theta}_{gj}(nT) \\ \tilde{\theta}_{ej}(nT) = \theta_d(nT) - \tilde{\theta}_{gj}(nT) \end{cases}$$

步骤6,根据误差 $(\tilde{x}_{ej}(nT+T), \tilde{y}_{ej}(nT+T), \tilde{\theta}_{ej}(nT+T))$,用评价函数 $f(v_j(nT), \omega_j(nT))$ 计算其评价值,寻找评价值最低的速度作为 nT 时刻实际的跟踪速度,

$$f(v_j(nT), \omega_j(nT)) = \sqrt{\tilde{x}_{ej}(nT+T)^2 + \tilde{y}_{ej}(nT+T)^2 + \tilde{\theta}_{ej}(nT+T)^2}。$$

2. 如权利要求1所述的机器人轨迹跟踪方法,其特征在于,所述步骤2中,按下式计算第

nT 时刻的线速度 $v_d(nT)$ 和角速度 $\omega_d(nT)$,

$$\begin{cases} v_d(nT) = \frac{\sqrt{(x_d(nT+T) - x_d(nT))^2 + (y_d(nT+T) - y_d(nT))^2}}{T} \\ \omega_d(nT) = \frac{\theta_d(nT+T) - \theta_d(nT)}{T} \end{cases}。$$

3. 如权利要求1所述的机器人轨迹跟踪方法,其特征在于,所述步骤2中,根据连续轨迹 $(x_d(t), y_d(t), \theta_d(t))$ 按下式计算连续速度函数 $(v_d(t), \omega_d(t))$,然后再离散化得到第 nT 时刻的线速度 $v_d(nT)$ 和角速度 $\omega_d(nT)$,

$$\begin{cases} v_d(t) = \sqrt{x_d'(t)^2 + y_d'(t)^2} \\ \omega_d(t) = \theta_d'(t) \end{cases}$$

其中, $x_d(t)'$ 、 $y_d(t)'$ 、 $\theta_d(t)'$ 分别为横坐标函数、纵坐标函数和角度函数对时间的求导。

一种机器人轨迹跟踪方法

技术领域

[0001] 本发明属于机器人导航与控制技术领域,具体涉及一种机器人轨迹跟踪方法。

背景技术

[0002] 机器人在实际运动控制过程中总会产生误差,或者是由于其他原因导致机器人在一定程度上偏离了设定的位置或姿态。如何发布速度指令以使得机器人能够良好地跟踪给定的路径或轨迹是解决误差时需要考虑的重要问题之一。跟踪控制包括路径跟踪和轨迹跟踪,前者跟踪的目标主要通过几何路径的方式表达,后者则通过各状态的时间函数表达。相比路径表达式,轨迹函数不仅包含了路径信息,而且隐含了目标机器人的速度信息,轨迹跟踪问题所研究的内容比路径跟踪问题更为丰富,也更为困难,但其研究的意义也更大。

[0003] 目前的跟踪控制算法一般基于非线性控制理论,如预测控制、模糊控制,或者几何因素,如纯追踪(pure pursuit)算法。常用的控制器包括传统及改进的PID控制器和基于Lyapunov函数构造的控制器。传统的跟踪控制算法大多只能单独跟踪位置或姿态,难以在跟踪位置的同时跟踪机器人的位姿。另外,传统跟踪控制算法的稳定性和动态性能并不理想,不能满足较高要求的跟踪控制。

发明内容

[0004] 本发明的目的在于提供一种机器人轨迹跟踪方法,基于Lyapunov函数构造控制律,能够同时跟踪机器人的位置和姿态;同时,本发明通过引入动态窗口确定参数,提高了跟踪的动态性能,跟踪曲线更贴合、更平滑。

[0005] 为了解决上述技术问题,本发明提供一种机器人轨迹跟踪方法,步骤如下:

[0006] 步骤1,将机器人的离散运动轨迹作为跟踪轨迹 $(x_d(nT), y_d(nT), \theta_d(nT))$,其中, T 为采样时间, $x_d(nT)$ 、 $y_d(nT)$ 和 $\theta_d(nT)$ 分别表示第 nT 时刻机器人离散运动轨迹中机器人位姿的横坐标、纵坐标和角度;

[0007] 步骤2,按照下式计算机器人坐标系中的机器人位姿误差 $(x_e(nT), y_e(nT), \theta_e(nT))$,

$$[0008] \quad \begin{cases} x_e(nT) = (x_d(nT) - x(nT))\cos\theta(nT) + (y_d(nT) - y(nT))\sin\theta(nT) \\ y_e(nT) = -(x_d(nT) - x(nT))\sin\theta(nT) + (y_d(nT) - y(nT))\cos\theta(nT) \\ \theta_e(nT) = \theta_d(nT) - \theta(nT) \end{cases}$$

[0009] 其中, $x(nT)$ 、 $y(nT)$ 、 $\theta(nT)$ 分别为第 nT 时刻机器人在全局坐标下的横坐标、纵坐标和角度;

[0010] 步骤3,按照下式所示的控制律,计算跟踪速度 $(v_j(nT), \omega_j(nT))$,

$$[0011] \quad \begin{cases} v_j(nT) = v_d(nT)\cos\theta_e(nT) + x_e(nT) - \theta_e(nT)\omega_j(nT) \\ \omega_j(nT) = \omega_d(nT) + v_d(nT)[\alpha_j(y_e(nT) + \theta_e(nT)) + (1 - \alpha_j)\sin\theta_e(nT)] \end{cases};$$

[0012] 其中, $v_d(nT)$ 和 $\omega_d(nT)$ 分别表示机器人在第 nT 时刻的线速度和角速度; $v_j(nT)$,

$\omega_j(nT)$ 分别为,在第 nT 时刻,机器人控制器参数为 α_j 时,控制器给出的线速度和角速度;设参数间隔 i , $0 < i < 1$,以 i 为间隔遍历;设控制器参数 α ,将第 j 个 α 的取值记作 α_j ,即 $\alpha_1 = i, \alpha_2 = 2i, \dots$,直到小于1的最大的 α 值;

[0013] 步骤4,根据跟踪速度 $(v_j(nT), w_j(nT))$,按下式预估 $(nT+T)$ 时刻的机器人全局位姿 $(\tilde{x}_{gj}(nT+T), \tilde{y}_{gj}(nT+T), \tilde{\theta}_{gj}(nT+T))$,

$$[0014] \quad \begin{cases} \tilde{x}_{gj}(nT+T) = x_g(nT) + v_j(nT)T \cos \theta_g(nT) \\ \tilde{y}_{gj}(nT+T) = y_g(nT) + v_j(nT)T \sin \theta_g(nT) \\ \tilde{\theta}_{gj}(nT+T) = \theta_g(nT) + \omega_j(nT)T \end{cases}$$

[0015] 其中, $\tilde{x}_{gj}(nT+T), \tilde{y}_{gj}(nT+T), \tilde{\theta}_{gj}(nT+T)$ 分别为机器人在 $(nT+T)$ 时刻应当达到的全局坐标系下的横坐标、纵坐标和角度, $x_g(nT), y_g(nT), \theta_g(nT)$ 分别为 nT 时刻机器人的全局位姿,即机器人在 nT 时刻在全局坐标系下的实际横坐标、纵坐标和角度;

[0016] 步骤5、按下式计算预估在当前控制器下,在 $(nT+T)$ 时刻机器人预估全局位姿与目标位姿的误差 $(\tilde{x}_{ej}(nT+T), \tilde{y}_{ej}(nT+T), \tilde{\theta}_{ej}(nT+T))$,

$$[0017] \quad \begin{cases} \tilde{x}_{ej}(nT) = (x_d(nT) - \tilde{x}_{gj}(nT)) \cos \tilde{\theta}_{gj}(nT) + (y_d(nT) - \tilde{y}_{gj}(nT)) \sin \tilde{\theta}_{gj}(nT) \\ \tilde{y}_{ej}(nT) = -(x_d(nT) - \tilde{x}_{gj}(nT)) \sin \tilde{\theta}_{gj}(nT) + (y_d(nT) - \tilde{y}_{gj}(nT)) \cos \tilde{\theta}_{gj}(nT) \\ \tilde{\theta}_{ej}(nT) = \theta_d(nT) - \tilde{\theta}_{gj}(nT) \end{cases}$$

[0018] 步骤6、根据误差 $(\tilde{x}_{ej}(nT+T), \tilde{y}_{ej}(nT+T), \tilde{\theta}_{ej}(nT+T))$,用评价函数 $f(v_j(nT), \omega_j(nT))$ 计算其评价值,寻找评价值最低的速度作为 nT 时刻实际的跟踪速度,

$$[0019] \quad f(v_j(nT), \omega_j(nT)) = \sqrt{\tilde{x}_{ej}(nT+T)^2 + \tilde{y}_{ej}(nT+T)^2 + \tilde{\theta}_{ej}(nT+T)^2}。$$

[0020] 进一步,所述步骤2中,按下式计算第 nT 时刻的线速度 $v_d(nT)$ 和角速度 $\omega_d(nT)$,

$$[0021] \quad \begin{cases} v_d(nT) = \frac{\sqrt{(x_d(nT+T) - x_d(nT))^2 + (y_d(nT+T) - y_d(nT))^2}}{T} \\ \omega_d(nT) = \frac{\theta_d(nT+T) - \theta_d(nT)}{T} \end{cases}。$$

[0022] 进一步,所述步骤2中,根据连续轨迹 $(x_d(t), y_d(t), \theta_d(t))$ 按下式计算连续速度函数 $(v_d(t), \omega_d(t))$,然后再离散化得到第 nT 时刻的线速度 $v_d(nT)$ 和角速度 $\omega_d(nT)$,

$$[0023] \quad \begin{cases} v_d(t) = \sqrt{x_d'(t)^2 + y_d'(t)^2} \\ \omega_d(t) = \theta_d'(t) \end{cases}$$

[0024] 其中, $x_d(t)', y_d(t)', \theta_d(t)'$ 分别为横坐标函数、纵坐标函数和角度函数对时间的求导。

[0025] 本发明与现有技术相比,其显著优点在于:

[0026] 1、本发明基于Lyapunov函数构造的控制律,保证了轨迹跟踪的稳定性;

[0027] 2、基于轨迹跟踪,本发明能够同时跟踪机器人的位置和姿态;

[0028] 3、引入动态窗口思想确定控制器参数,提高了跟踪的动态性能,跟踪曲线更贴合、

更平滑。

附图说明

- [0029] 图1为本发明机器人轨迹跟踪模型示意图；
 [0030] 图2为本发明机器人轨迹跟踪方法流程图；
 [0031] 图3为本发明机器人轨迹跟踪方法跟踪圆形变速轨迹的效果图；
 [0032] 图4为本发明机器人轨迹跟踪方法跟踪圆形变速轨迹的误差图。

具体实施方式

[0033] 容易理解,依据本发明的技术方案,在不变更本发明的实质精神的情况下,本领域的一般技术人员可以想象出本发明机器人轨迹跟踪方法的多种实施方式。因此,以下具体实施方式和附图仅是对本发明的技术方案的示例性说明,而不应当视为本发明的全部或者视为对本发明技术方案的限制或限定。

[0034] 本发明基本原理:

[0035] 如图1所示,在实际机器人的机器人坐标系中定义位姿误差,其在全局坐标系中的计算公式为:

$$[0036] \quad \begin{cases} x_e = (x_d - x) \cos \theta + (y_d - y) \sin \theta \\ y_e = -(x_d - x) \sin \theta + (y_d - y) \cos \theta \\ \theta_e = \theta_d - \theta \end{cases}$$

[0037] 其中, (x, y, θ) 为机器人在全局坐标中实际位姿的横坐标、纵坐标和角度, (x_d, y_d, θ_d) 为指定轨迹目标位姿的横坐标、纵坐标和角度, (x_e, y_e, θ_e) 表示机器人实际位姿与轨迹目标位姿在机器人坐标系下的误差,即以机器人横纵坐标为原点,角度方向为正方向,两者横纵坐标和角度的差值。可以将轨迹跟踪的问题转换为一个针对系统误差的镇定问题,即设计一个控制器,使得系统在平衡的 $(0, 0, 0)$ 附近稳定,亦即 $\lim_{t \rightarrow \infty} x_e = 0, \lim_{t \rightarrow \infty} y_e = 0, \lim_{t \rightarrow \infty} \theta_e = 0$ 。

$$[0038] \quad \begin{cases} \dot{x}_e = \omega y_e - v + v_d \cos \theta_e \\ \dot{y}_e = -\omega x_e + v_d \sin \theta_e \\ \dot{\theta}_e = \omega_d - \omega \end{cases}$$

[0039] 其中, v 为机器人线速度, ω 为机器人角速度。针对该系统设计Lyapunov函数:

$$[0040] \quad V(x_e, y_e, \theta_e) = \frac{x_e^2}{2} + \frac{(y_e + \theta_e)^2}{2} + (1 - \cos \theta_e)$$

[0041] 易证,当且仅当 $x_e^2 + y_e^2 + \theta_e^2 = 0$ 时, $V = 0$; 对 $\forall x_e^2 + y_e^2 + \theta_e^2 \neq 0, V > 0$; 当 $x_e \rightarrow \infty \mid y_e \rightarrow \infty \mid \theta_e \rightarrow \infty$ 时, $V \rightarrow \infty$ 。

[0042] 如果能设计控制器 $\begin{cases} v = f_v(x_e, y_e, \theta_e) \\ \omega = f_\omega(x_e, y_e, \theta_e) \end{cases}$ 使得 $\dot{V} < 0$, 则系统是稳定的。推导易得 \dot{V} 如式:

$$[0043] \quad \begin{cases} \dot{x}_e = f_\omega(x_e, y_e, \theta_e)y_e - f_v(x_e, y_e, \theta_e) + v_d \cos \theta_e \\ \dot{y}_e = -f_\omega(x_e, y_e, \theta_e)x_e + v_d \sin \theta_e \\ \dot{\theta}_e = \omega_d - f_\omega(x_e, y_e, \theta_e) \end{cases}$$

$$[0044] \quad \dot{V}(x_e, y_e, \theta_e) = x_e(v_d \cos \theta_e - v - \omega \theta_e) + (y_e + \theta_e)(v_d \sin \theta_e + \omega_d - \omega) + (\omega_d - \omega) \sin \theta_e$$

[0045] 设计控制器如式：

$$[0046] \quad \begin{cases} v = v_d \cos \theta_e + x_e - \theta_e \omega \\ \omega = \omega_d + v_d[\alpha(y_e + \theta_e) + (1 - \alpha) \sin \theta_e] \end{cases}$$

[0047] 可得：

$$[0048] \quad \dot{V}(x_e, y_e, \theta_e) = -x_e^2 - v_d \alpha (y_e + \theta_e)^2 - (1 - \alpha) v_d \sin^2 \theta_e$$

[0049] 如果另 $v_d > 0, 0 < \alpha < 1$ 。则有 $\dot{V}(x_e, y_e, \theta_e) < 0$ ，且当 $x_e \rightarrow \infty \mid y_e \rightarrow \infty \mid \theta_e \rightarrow \infty$ 时， $\dot{V} \rightarrow \infty$ 。

此时可证 $\lim_{t \rightarrow \infty} x_e = 0, \lim_{t \rightarrow \infty} y_e = 0, \lim_{t \rightarrow \infty} \theta_e = 0$ 。

[0050] 结合附图，本发明机器人轨迹跟踪方法包括以下步骤：

[0051] 步骤1、给定机器人的离散运动轨迹，或者对机器人的连续运动轨迹离散化，作为机器人的目标跟踪轨迹 $x_d(nT), y_d(nT), \theta_d(nT)$ 。其中 T 为采样时间， $(x_d(nT), y_d(nT), \theta_d(nT))$ 表示在第 n 个序列，即第 nT 时刻离散运动轨迹规定的机器人目标位姿的横坐标、纵坐标和角度；

[0052] 步骤2、由离散运动轨迹计算离散速度函数，或者由连续运动轨迹求导计算连续速度函数，再离散化得到离散速度函数 $v_d(nT), \omega_d(nT)$ ，其中 T 为采样时间， $v_d(nT), \omega_d(nT)$ 表示目标机器人在第 n 个序列，即第 nT 时刻，应该达到的线速度和角速度；

[0053] 步骤3、按下式，在机器人坐标系中计算位姿误差 $(x_e(nT), y_e(nT), \theta_e(nT))$ ，

$$[0054] \quad \begin{cases} x_e(nT) = (x_d(nT) - x(nT)) \cos \theta(nT) + (y_d(nT) - y(nT)) \sin \theta(nT) \\ y_e(nT) = -(x_d(nT) - x(nT)) \sin \theta(nT) + (y_d(nT) - y(nT)) \cos \theta(nT) \\ \theta_e(nT) = \theta_d(nT) - \theta(nT) \end{cases}$$

[0055] 其中， T 为采样时间， $(x(nT), y(nT), \theta(nT))$ 为第 n 个序列，即 nT 时刻机器人的实际位姿， $x(nT), y(nT), \theta(nT)$ 分别为 nT 时刻机器人在全局坐标下的横坐标、纵坐标和角度；

[0056] 步骤4、根据系统性能和实时性要求，选择合适的参数间隔 i ($0 < i < 1$)，以 i 为间隔遍历，选取控制器参数 α ，将第 j 个 α 取值记作 α_j ，即 $\alpha_1 = i, \alpha_2 = 2i, \dots$ ，直到小于1的最大的 α 值，并按下式构建控制律，计算相应的跟踪速度 $v_j(nT), \omega_j(nT)$ ，其中 T 为采样时间， $v_j(nT), \omega_j(nT)$ 表示目标机器人在控制器参数取 α_j 时，在第 n 个 nT 时刻，控制器给出的线速度和角速度：

$$[0057] \quad \begin{cases} v_j(nT) = v_d(nT) \cos \theta_e(nT) + x_e(nT) - \theta_e(nT) \omega_j(nT) \\ \omega_j(nT) = \omega_d(nT) + v_d(nT)[\alpha_j(y_e(nT) + \theta_e(nT)) + (1 - \alpha_j) \sin \theta_e(nT)] \end{cases}$$

[0058] 步骤5、根据跟踪速度 $v_j(nT), \omega_j(nT)$ ，按下式预估相应的下一时刻，即 $(nT+T)$ 时刻的机器人全局位姿 $(\tilde{x}_{gj}(nT+T), \tilde{y}_{gj}(nT+T), \tilde{\theta}_{gj}(nT+T))$ ，即在参数为 α_j 的控制器下，机器人在 $(nT+T)$ 时刻理论达到的全局坐标系下的横坐标、纵坐标和角度：

$$[0059] \quad \begin{cases} \tilde{x}_{gj}(nT+T) = x_g(nT) + v_j(nT)T \cos \theta_g(nT) \\ \tilde{y}_{gj}(nT+T) = y_g(nT) + v_j(nT)T \sin \theta_g(nT) \\ \tilde{\theta}_{gj}(nT+T) = \theta_g(nT) + \omega_j(nT)T \end{cases}$$

[0060] 其中, $(x_g(nT), y_g(nT), \theta_g(nT))$ 为 nT 时刻机器人的全局位姿, 即机器人在 nT 时刻在全局坐标系下的实际的横坐标、纵坐标和角度, T 为采样周期;

[0061] 步骤6、对于根据在参数为 α_j 的控制器下预估的 $(nT+T)$ 时刻全局位姿 $(\tilde{x}_{gj}(nT+T), \tilde{y}_{gj}(nT+T), \tilde{\theta}_{gj}(nT+T))$ 和待跟踪的离散轨迹所指定的 $(nT+T)$ 时刻目标位姿 $x_d(nT+T), y_d(nT+T), \theta_d(nT+T)$, 按下式计算预估在当前控制器下, 在 $(nT+T)$ 时刻机器人理论位姿与目标位姿的误差 $(\tilde{x}_{ej}(nT+T), \tilde{y}_{ej}(nT+T), \tilde{\theta}_{ej}(nT+T))$, 即 $(nT+T)$ 时刻机器人理论横坐标、纵坐标和角度与目标横坐标、纵坐标和角度的差值:

$$[0062] \quad \begin{cases} \tilde{x}_{ej}(nT) = (x_d(nT) - \tilde{x}_{gj}(nT)) \cos \tilde{\theta}_{gj}(nT) + (y_d(nT) - \tilde{y}_{gj}(nT)) \sin \tilde{\theta}_{gj}(nT) \\ \tilde{y}_{ej}(nT) = -(x_d(nT) - \tilde{x}_{gj}(nT)) \sin \tilde{\theta}_{gj}(nT) + (y_d(nT) - \tilde{y}_{gj}(nT)) \cos \tilde{\theta}_{gj}(nT) \\ \tilde{\theta}_{ej}(nT) = \theta_d(nT) - \tilde{\theta}_{gj}(nT) \end{cases}$$

[0063] 此处为方便公式书写, 以上式形式表述, 实际计算时, 将 $n+1$ 代入 n 即可。

[0064] 步骤7、针对预估的在参数为 α_j 的控制器下, $(nT+T)$ 时刻机器人理论位姿与目标位姿的误差 $(\tilde{x}_{ej}(nT+T), \tilde{y}_{ej}(nT+T), \tilde{\theta}_{ej}(nT+T))$, 用评价函数 $f(v_j(nT), \omega_j(nT))$ 计算其评价价值, 寻找评价价值最低的速度 $(v(nT), \omega(nT))$, $s.t. f(v(nT), \omega(nT)) = \min\{f(v_j(nT), \omega_j(nT))\}$ 作为 nT 时刻实际的跟踪速度

$$[0065] \quad f(v_j(nT), \omega_j(nT)) = \sqrt{\tilde{x}_{ej}(nT+T)^2 + \tilde{y}_{ej}(nT+T)^2 + \tilde{\theta}_{ej}(nT+T)^2}$$

[0066] 到下一时刻 $nT+T$, 重复步骤3~7。

[0067] 进一步的, 所述步骤2中由离散运动轨迹 $x_d(nT), y_d(nT), \theta_d(nT)$ 可按下式计算离散速度函数 $v_d(nT), \omega_d(nT)$:

$$[0068] \quad \begin{cases} v_d(nT) = \frac{\sqrt{(x_d(nT+T) - x_d(nT))^2 + (y_d(nT+T) - y_d(nT))^2}}{T} \\ \omega_d(nT) = \frac{\theta_d(nT+T) - \theta_d(nT)}{T} \end{cases}$$

[0069] 进一步的, 所述步骤2中由连续轨迹 $x_d(t), y_d(t), \theta_d(t)$ 可按下式计算连续速度函数 $v_d(t), \omega_d(t)$:

$$[0070] \quad \begin{cases} v_d(t) = \sqrt{x_d'(t)^2 + y_d'(t)^2} \\ \omega_d(t) = \theta_d'(t) \end{cases}$$

[0071] 其中 $x_d(t)', y_d(t)', \theta_d(t)'$ 分别为横坐标函数、纵坐标函数和角度函数对时间的求导。将 nT 代入 t , 即可得到离散速度函数 $v_d(nT), \omega_d(nT)$ 。

[0072] 进一步的, 所述步骤4中的参数间隔 i 越小, 控制器参数 α 的取值情况越多, 耗费的计算时间越长, 但也可能获得更好的跟踪结果。

[0073] 进一步的,所述步骤4中构建的控制律 $\begin{cases} v = f_v(x_e, y_e, \theta_e) \\ \omega = f_\omega(x_e, y_e, \theta_e) \end{cases}$,可构建Lyapunov函数

$V(x_e, y_e, \theta_e) = \frac{x_e^2}{2} + \frac{(y_e + \theta_e)^2}{2} + (1 - \cos \theta_e)$,并易证明该轨迹跟踪系统是稳定的,其中 (x_e, y_e, θ_e) 为机器人在机器人坐标系中的位姿误差。

[0074] 具体的给定目标机器人的轨迹如式:
$$\begin{cases} x_d(t) = \cos(\theta_d(t) - \frac{\pi}{2}) \\ y_d(t) = \sin(\theta_d(t) - \frac{\pi}{2}) \\ \theta_d(t) = t^{\frac{3}{4}} + \frac{\pi}{2} \end{cases}$$

[0075] 则目标机器人的控制输入为: $v_d(t) = \frac{3}{4}t^{\frac{1}{4}}(m/s), \omega_d(t) = \frac{3}{4}t^{\frac{1}{4}}(rad/s)$ 。

[0076] 设离散采样控制周期为 $T_s = 0.02s$,则可得离散轨迹式和控制输入

$v_d(n) = \frac{3}{4}(nT_s)^{\frac{1}{4}}(m/s), \omega_d(n) = \frac{3}{4}(nT_s)^{\frac{1}{4}}(rad/s)$ 。

[0077]
$$\begin{cases} x_d(n) = \cos(\theta_d(nT_s) - \frac{\pi}{2}) \\ y_d(n) = \sin(\theta_d(nT_s) - \frac{\pi}{2}) \\ \theta_d(n) = (nT_s)^{\frac{3}{4}} + \frac{\pi}{2} \end{cases}$$

[0078] 设受控机器人的初始位姿为 $x(0) = 0, y(0) = 0, \theta(0) = 0$ 。按本发明的跟踪方法构造控制器,其中,控制器参数 α 的采样间隔取 $k = 0.1$,仿真时间为 $20s$,不妨将控制器参数 α 取常量值 0.5 作为经典算法加以比较,将动态选取 α 的方法成为改进算法,则实际跟踪效果和误差曲线如图3,图4。

[0079] 由图3、图4可知,即使初始位姿完全不在目标轨迹上,对于变速的圆形轨迹,本发明的轨迹跟踪方法也能较好跟踪。通过引入动态窗口的改进,可以在跟踪时,比不动态选参收获更好的收敛效果。

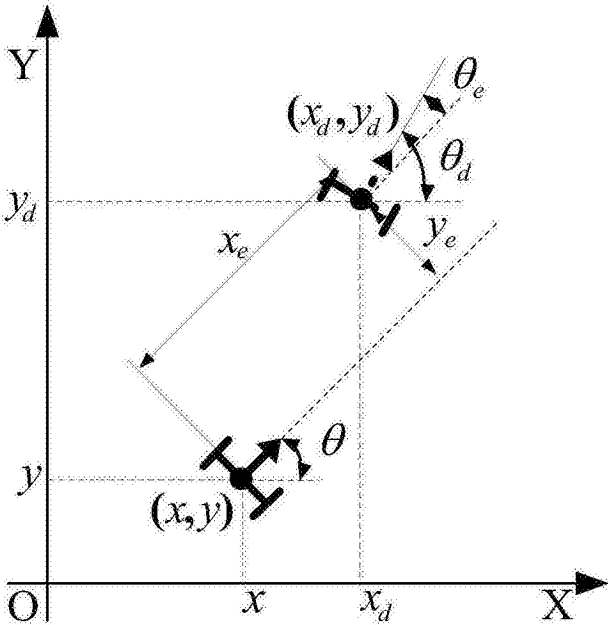


图1

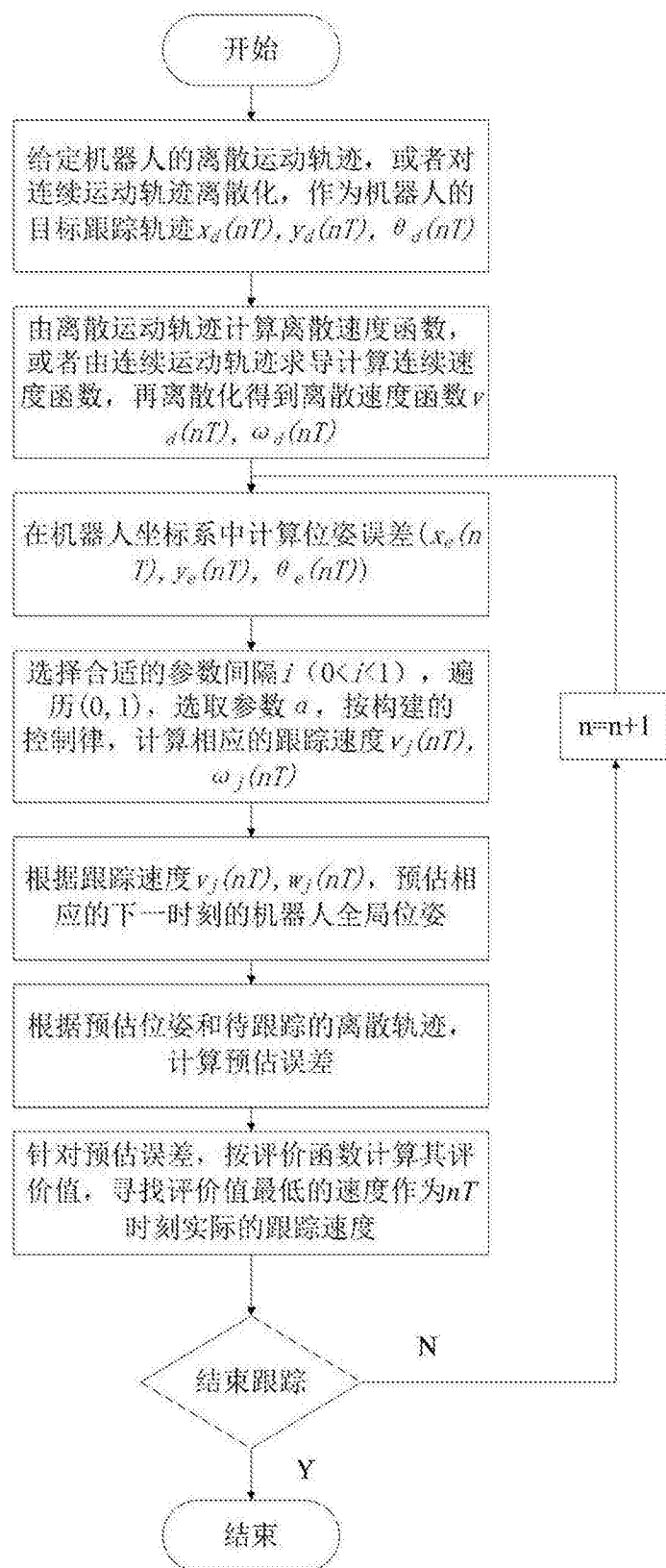


图2

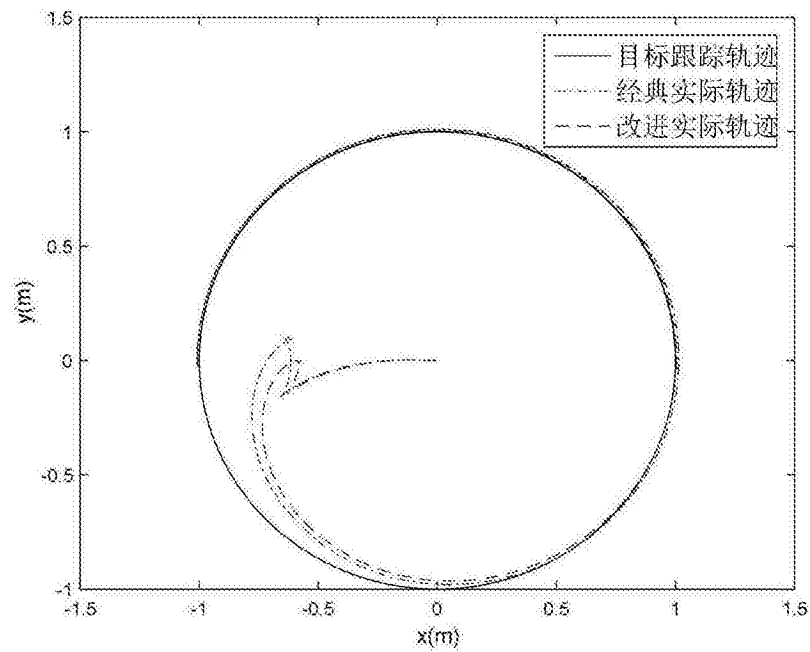


图3

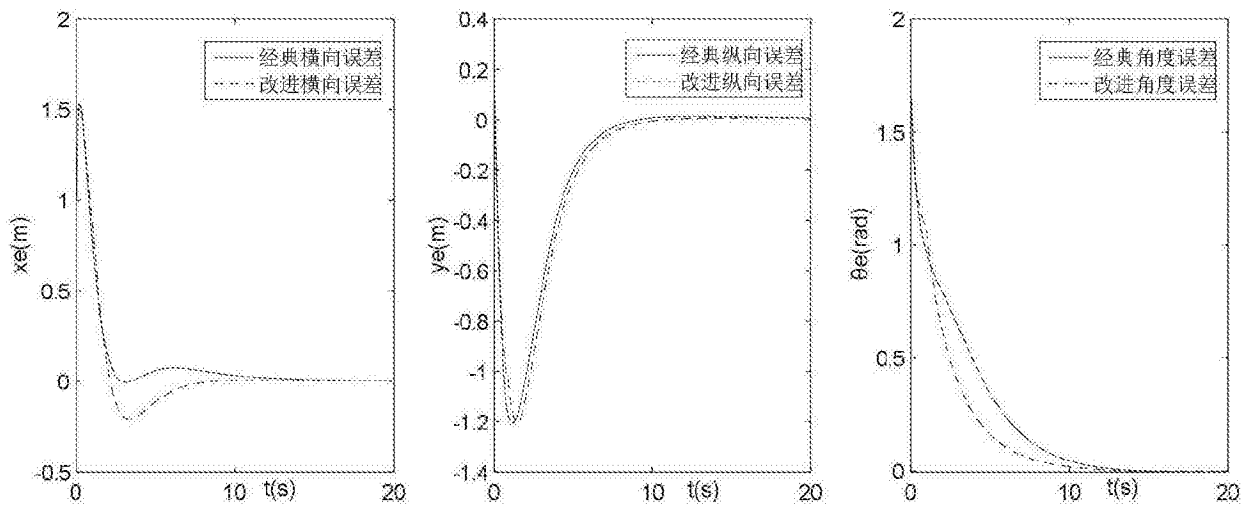


图4