



(12)发明专利

(10)授权公告号 CN 110244735 B

(45)授权公告日 2020.08.21

(21)申请号 201910551319.6

CN 107272677 A,2017.10.20

(22)申请日 2019.06.24

CN 109164810 A,2019.01.08

(65)同一申请的已公布的文献号

申请公布号 CN 110244735 A

LIN Weisong等.Adaptive critic motion control design of autonomous wheeled mobile robot by dual heuristic programming.《Automatica》.2008,

(43)申请公布日 2019.09.17

BEGNINI M等.A robust adaptive fuzzy variable structure tracking control for the wheeled mobile robot:simulation and experimental results.《Control Engineering Practice》.2017,

(73)专利权人 安徽农业大学

地址 230036 安徽省合肥市长江西路130号

武星等.视觉导引AGV鲁棒特征识别与精确路径跟踪研究.《农业机械学报》.2016,第47卷(第7期),

(72)发明人 宋彦 张羊阳 姚琦

(74)专利代理机构 安徽省合肥新安专利代理有限责任公司 34101

代理人 陆丽莉

魏爽等.基于GNSS的农机自动导航路径搜索及转向控制.《农业工程学报》.2017,第33卷

(51)Int.Cl.

G05D 1/02(2020.01)

审查员 刘慧

(56)对比文件

CN 105549598 A,2016.05.04

权利要求书4页 说明书10页 附图3页

(54)发明名称

移动机器人跟踪预定轨迹的启发式动态规划控制方法

(57)摘要

本发明公开了一种移动机器人跟踪预定轨迹的启发式动态规划控制方法,其步骤包括:1、建立大地坐标系和机器人坐标系;2、确定移动机器人的位姿信息和预瞄点位姿信息;3、计算预瞄点位置的道路曲率;4、建立移动机器人的马尔可夫决策过程模型;5、基于启发式动态规划算法设计跟踪预定轨迹的控制方法;6、推导启发式动态规划算法中执行器和评价器的在线学习规则。7、判断预瞄点是否为预定轨迹中的最后一位置坐标,若是,则控制机器人减速至预瞄点位置,若否则返回步骤2顺序进行。本发明能够在跟踪不同的预定轨迹时不需要对控制参数进行反复调试就能够获得较好的控制效果,从而能提高移动机器人的环境适应性和自主优化能力。



1. 一种移动机器人跟踪预定轨迹的启发式动态规划控制方法,其特征是按如下步骤进行:

第一步、以移动机器人安装的GPS传感器主天线处为坐标原点,以移动机器人前进方向为x轴,以移动机器人的前进方向逆时针旋转90度方向为y轴,建立机器人坐标系xoy;以大地质心为原点,正东方向为X轴,以正北方向为Y轴,建立大地坐标系XOY;

第二步、控制移动机器人运动,并将大地坐标系下移动机器人的位姿信息记录存储作为预定轨迹;所述位姿信息包括位置坐标和航向角;

第三步、确定当前时刻 t_i 下移动机器人的位姿信息和预瞄点位姿信息,从而建立机器人坐标系下的轨迹跟踪误差状态方程;

步骤3.1、定义 $(X_c(t_i), Y_c(t_i))$ 为移动机器人在大地坐标系XOY中当前时刻 t_i 下的位置坐标,定义航向角 $\phi_c(t_i)$ 为当前时刻 t_i 下移动机器人纵轴线与大地坐标系X轴的夹角,定义 $(X_c(t_i), Y_c(t_i), \phi_c(t_i))$ 为移动机器人的当前时刻 t_i 下的位姿信息;

步骤3.2、设定预瞄距离为L,在预定轨迹中搜索大于预瞄距离L且距离移动机器人最近的位姿信息,并将符合要求的位姿信息定义为预瞄点 O_p 记为 $(X_p(t_i), Y_p(t_i), \phi_p(t_i))$;

步骤3.3、定义 $(x_e(t_i), y_e(t_i), \phi_e(t_i))$ 为当前时刻 t_i 下移动机器人在机器人坐标系xoy中的位姿误差,其中 $x_e(t_i)$ 为纵向偏差, $y_e(t_i)$ 为横向偏差, $\phi_e(t_i)$ 为航向角偏差,则利用式(1)建立轨迹跟踪误差状态方程:

$$\begin{bmatrix} x_e(t_i) \\ y_e(t_i) \\ \phi_e(t_i) \end{bmatrix} = \begin{bmatrix} \cos \phi_c(t_i) & \sin \phi_c(t_i) & 0 \\ -\sin \phi_c(t_i) & \cos \phi_c(t_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_p(t_i) - X_c(t_i) \\ Y_p(t_i) - Y_c(t_i) \\ \phi_p(t_i) - \phi_c(t_i) \end{bmatrix} \quad (1)$$

步骤3.4、利用式(2)得到当前时刻 t_i 下移动机器人与预瞄点 O_p 的横向偏差的变化率 $\dot{y}_e(t_i)$ 和航向角偏差的变化率 $\dot{\phi}_e(t_i)$:

$$\begin{cases} \dot{y}_e(t_i) = -v_y(t_i) - L\omega(t_i) + v_x(t_i)\phi_e(t_i) \\ \dot{\phi}_e(t_i) = v_x(t_i)\rho(t_i) - \omega(t_i) \end{cases} \quad (2)$$

式(2)中, $v_x(t_i)$ 为移动机器人当前时刻 t_i 下的纵向速度, $v_y(t_i)$ 为移动机器人当前时刻 t_i 下的侧向速度, $\omega(t_i)$ 为移动机器人当前时刻 t_i 下的角速度, $\rho(t_i)$ 为当前时刻 t_i 下预瞄点 O_p 处的道路曲率;

第四步、计算当前时刻 t_i 下预瞄点 O_p 处位置的道路曲率 $\rho(t_i)$;

步骤4.1、从所述预定轨迹中选择三个位姿信息,包括以预瞄点 O_p 作为第1个位姿信息、以与移动机器人的位置坐标 $(X_c(t_i), Y_c(t_i))$ 距离相同的两个位姿信息A和B作为第2个和第3个位姿信息,其中位姿信息A与位置坐标 $(X_c(t_i), Y_c(t_i))$ 的距离为 $L-\delta$;位姿信息B与位置坐标 $(X_c(t_i), Y_c(t_i))$ 的距离为 $L+\delta$, δ 为所设定的间隔距离;

步骤4.2、利用式(3)将选取的三个位姿信息进行坐标转换,从而得到当前时刻 t_i 下在机器人坐标系xoy中任意第j个位姿信息与移动机器人的位置坐标 $(X_c(t_i), Y_c(t_i))$ 的距离分别在x轴的投影 $x_{d,j}(t_i)$ 和y轴上的投影 $y_{d,j}(t_i)$ 以及当前时刻 t_i 下第j个位姿信息在机器人坐标系中的横坐标 $x_j(t_i)$ 和纵坐标 $y_j(t_i)$;

$$\begin{cases} x_{d,j}(t_i) = \frac{R_{ave}\pi}{180}(Y_j(t_i) - Y_c(t_i)) \cos \frac{\pi X_j(t_i)}{180} \\ y_{d,j}(t_i) = \frac{R_{ave}\pi}{180}(X_j(t_i) - X_c(t_i)) \\ x_j(t_i) = y_{d,j}(t_i) \cos \phi_c(t_i) + x_{d,j}(t_i) \sin \phi_c(t_i) \\ y_j(t_i) = y_{d,j}(t_i) \sin \phi_c(t_i) - x_{d,j}(t_i) \cos \phi_c(t_i) \end{cases} \quad (3)$$

式(3)中, $X_j(t_i)$, $Y_j(t_i)$ 为选取的预定轨迹上的经纬度坐标, R_{ave} 为地球平均半径, $j=1, 2, 3$;

步骤4.3、使用三次多项式对坐标转换后的三个位姿信息的位置坐标进行曲线拟合, 从而得到当前时刻 t_i 下预瞄点 O_p 处的道路曲率 $\rho(t_i)$;

第五步、建立移动机器人的马尔可夫决策过程模型:

步骤5.1、定义移动机器人在当前时刻 t_i 下的状态向量为 $s(t_i) = (y_e(t_i), \phi_e(t_i), \omega(t_i), \rho(t_i))^T$, 定义移动机器人在当前时刻 t_i 下的控制动作为角速度 $u(t_i)$;

步骤5.2、利用式(4)计算当前时刻 t_i 下状态向量 $s(t_i)$ 的回报 $r(s(t_i))$:

$$r(s(t_i)) = s(t_i)^T Q s(t_i) + (u(t_i) - \omega_p(t_i))^T R (u(t_i) - \omega_p(t_i)) \quad (4)$$

式(4)中, Q 为半正定对角矩阵, R 为一正常数, $\omega_p(t_i)$ 为当前时刻 t_i 下预瞄点 O_p 处期望的角速度, 且 $\omega_p(t_i) = v_x(t_i) \rho(t_i)$;

步骤5.3、利用式(5)定义当前时刻 t_i 下状态向量 $s(t_i)$ 的性能指标函数值 $J(s(t_i))$:

$$J(s(t_i)) = r(s(t_i), u(s(t_i))) + \gamma J(s(t_{i+1})) \quad (5)$$

式(5)中, $r(s(t_i), u(s(t_i)))$ 为当前时刻 t_i 下的立即回报值, 其中 $s(t_i)$ 为当前时刻 t_i 下的状态向量, $u(s(t_i))$ 为当前时刻 t_i 下移动机器人的动作, γ 为折扣因子, $J(s(t_{i+1}))$ 为下一时刻 t_{i+1} 下的性能指标函数, $s(t_{i+1})$ 为下一时刻 t_{i+1} 下的移动机器人状态向量, 并由式(6)得到:

$$s(t_{i+1}) = \begin{bmatrix} y_e(t_i) \\ \phi_e(t_i) \\ \omega(t_i) \\ \rho(t_i) \end{bmatrix} + T_s \times \begin{bmatrix} -L \times \omega(t_i) + v_x(t_i) \times \phi_e(t_i) - v_y(t_i) \\ v_x(t_i) \times \rho(t_i) - \omega(t_i) \\ 0 \\ 0 \end{bmatrix} \quad (6)$$

式(6)中, T_s 为控制周期, 且 $T_s = t_{i+1} - t_i$;

步骤5.4、利用式(7)建立当前时刻 t_i 下最优的控制动作 $u^*(s(t_i))$:

$$u^*(s(t_i)) = \frac{-1}{2R} \gamma \left[\left\{ \frac{\partial s(t_{i+1})}{\partial u(s(t_i))} \right\}^T \frac{\partial J(s(t_{i+1}))}{\partial s(t_{i+1})} \right] + \omega_p(t_i) \quad (7)$$

第六步、利用式(8)计算当前时刻 t_i 下的控制动作 $u(s(t_i))$ 并作为HDP算法中执行器的输出:

$$u(s(t_i)) = \Phi_a(W_a(t_i) \times \sigma_a(V_a(t_i) \times s(t_i) + b_{a1}(t_i)) + b_{a2}(t_i)) \quad (8)$$

式(8)中, $V_a(t_i)$ 为当前时刻 t_i 下神经网络从输入层到隐含层的第一权值, $b_{a1}(t_i)$ 为当前时刻 t_i 下神经网络从输入层到隐含层的第一偏置, $W_a(t_i)$ 为当前时刻 t_i 下隐含层到输出层的第一权值, $b_{a2}(t_i)$ 为当前时刻 t_i 下隐含层到输出层的第一偏置; $\Phi_a(\cdot)$ 表示用于逼近

执行器的神经网络输出层激活函数; $\sigma_a(\cdot)$ 表示用于逼近执行器的神经网络隐含层激活函数;

利用式(9)计算当前时刻 t_i 下状态向量 $s(t_i)$ 的性能指标函数值 $J(s(t_i))$ 并作为HDP算法中评价器的输出:

$$J(s(t_i)) = \Phi_c(W_c(t_i) \times \sigma_c(V_c(t_i) \times s(t_i) + b_{c1}(t_i)) + b_{c2}(t_i)) \quad (9)$$

式(9)中, $V_c(t_i)$ 为当前时刻 t_i 下神经网络从输入层到隐含层的第二权值, $b_{c1}(t_i)$ 为当前时刻 t_i 下神经网络从输入层到隐含层的第二偏置, $W_c(t_i)$ 为当前时刻 t_i 下隐含层到输出层的第二权值, $b_{c2}(t_i)$ 为当前时刻 t_i 下隐含层到输出层的第二偏置; $\Phi_c(\cdot)$ 表示用于逼近评价器的神经网络输出层激活函数; $\sigma_c(\cdot)$ 表示用于逼近评价器的神经网络隐含层激活函数;

第七步、建立HDP算法中执行器网络和评价器网络的在线学习规则;

步骤7.1、执行器网络在线学习规则:

利用式(10)建立执行器误差函数 $E_a(s(t_i))$,并将最小化执行器误差函数 $E_a(s(t_i))$ 作为执行器网络的学习目标:

$$E_a(s(t_i)) = \frac{1}{2}(u(s(t_i)) - u^*(s(t_i)))^2 \quad (10)$$

定义中间变量 $\zeta_d(t_i) = u(s(t_i)) - u^*(s(t_i))$,并利用式(11)-式(14)更新当前时刻 t_i 下执行器网络的权值 $W_a(t_i)$, $V_a(t_i)$ 和偏置 $b_{a1}(t_i)$, $b_{a2}(t_i)$,从而得到下一时刻 t_{i+1} 下的执行器网络的权值 $W_a(t_{i+1})$, $V_a(t_{i+1})$ 和偏置 $b_{a1}(t_{i+1})$, $b_{a2}(t_{i+1})$:

$$W_a(t_{i+1}) = W_a(t_i) - \alpha_a \zeta_d(t_i) \frac{\partial u(s(t_i))}{\partial W_a(t_i)} \quad (11)$$

$$V_a(t_{i+1}) = V_a(t_i) - \alpha_a \zeta_d(t_i) \frac{\partial u(s(t_i))}{\partial V_a(t_i)} \quad (12)$$

$$b_{a1}(t_{i+1}) = b_{a1}(t_i) - \alpha_a \zeta_d(t_i) \frac{\partial u(s(t_i))}{\partial b_{a1}(t_i)} \quad (13)$$

$$b_{a2}(t_{i+1}) = b_{a2}(t_i) - \alpha_a \zeta_d(t_i) \frac{\partial u(s(t_i))}{\partial b_{a2}(t_i)} \quad (14)$$

式(11)-式(14)中, α_a 为执行器网络的学习率;

步骤7.2、评价器网络在线学习规则:

定义中间变量为 $\delta_d(s(t_i)) = J(s(t_i)) - r(s(t_i), u(s(t_i))) - \gamma J(s(t_{i+1}))$,并利用式(15)建立评价器误差函数 $E_c(s(t_i))$,并以最小化误差函数 $E_c(s(t_i))$ 作为评价器的学习目标:

$$E_c(s(t_i)) = \frac{1}{2} \delta_d^2(s(t_i)) \quad (15)$$

利用式(16)-式(19)更新当前时刻 t_i 下评价器网络的权值 $W_c(t_i)$, $V_c(t_i)$ 和偏置 $b_{c1}(t_i)$, $b_{c2}(t_i)$,从而得到下一时刻 t_{i+1} 下的评价器网络的权值 $W_c(t_{i+1})$, $V_c(t_{i+1})$ 和偏置 $b_{c1}(t_{i+1})$, $b_{c2}(t_{i+1})$:

$$W_c(t_{i+1}) = W_c(t_i) - \alpha_c \delta_d(t_i) \frac{\partial J(s(t_i))}{\partial W_c(t_i)} \quad (16)$$

$$V_c(t_{i+1}) = V_c(t_i) - \alpha_c \delta_d(t_i) \frac{\partial J(s(t_i))}{\partial V_c(t_i)} \quad (17)$$

$$b_{c1}(t_{i+1}) = b_{c1}(t_i) - \alpha_c \delta_d(t_i) \frac{\partial J(s(t_i))}{\partial b_{c1}(t_i)} \quad (18)$$

$$b_{c2}(t_{i+1}) = b_{c2}(t_i) - \alpha_c \delta_d(t_i) \frac{\partial J(s(t_i))}{\partial b_{c2}(t_i)} \quad (19)$$

式(16)-式(19)中, α_c 为评价器网络的学习率;

第八步、由执行器网络得到的控制动作 $u(s(t_i))$, 以及当前移动机器人的前进速度, 若所述移动机器人为履带式机器人, 则利用式(20)得到移动机器人两侧电机的转速 $n_L(t_i)$, $n_R(t_i)$; 若所述移动机器人为轮式机器人, 则利用式(21)得到移动机器人前轮转向角:

$$\begin{bmatrix} n_L(t_i) \\ n_R(t_i) \end{bmatrix} = \begin{bmatrix} 1/2\pi r_d & B/4\pi r_d \\ 1/2\pi r_d & -B/4\pi r_d \end{bmatrix} \begin{bmatrix} v_c \\ u(s(t_i)) \end{bmatrix} \quad (20)$$

式(20)中, r_d 为移动机器人的驱动轮半径, B 为履带中心距, v_c 为移动机器人前进速度;

$$\delta(t_i) = \frac{L_z u(s(t_i))}{v_c} \quad (21)$$

式(21)中, $\delta(t_i)$ 为前轮转向角, L_z 为移动机器人轴距;

第九步、判断当前时刻 t_i 下预瞄点 O_p 是否为预定轨迹中的最后一位置坐标, 若是, 则令角速度 $u(s(t_i)) = 0$, 控制移动机器人减速并在达到预瞄点 O_p 时停止运动; 否则, 将 $i+1$ 赋值给 i , 并返回第三步顺序执行。

移动机器人跟踪预定轨迹的启发式动态规划控制方法

技术领域

[0001] 本发明属于移动机器人技术领域,具体涉及一种移动机器人跟踪预定轨迹的控制方法。

背景技术

[0002] 随着移动机器人的广泛应用,复杂的地形和多变的环境对移动机器人的运动控制能力提出了更高的要求,如何提高移动机器人对于不同线形道路的路径跟踪性能成为了研究的热点。

[0003] 移动机器人对于预定轨迹的跟踪多采用“预瞄—跟随”原理,传统的控制方法有纯追踪和PID控制,但是上述方法均忽略了系统的数学模型,难以取得高精度的控制效果。一些学者提出了基于车辆动力学模型或运动学模型的控制方法,上述方法在模型较为精确的条件下,提高了控制精度,但是需要调节和优化的控制参数较多,尤其是当移动机器人的工作环境发生变化时,需要重新选择控制参数,优化控制性能。

[0004] 为了提高移动机器人的环境适应性和自主行为能力,一些学者提出了基于智能控制的方法,如模糊控制和神经网络控制等。但是模糊控制仍然需要将专家经验抽象为模糊控制规则,并调校模糊控制参数。神经网络控制方法需要大量的试验数据训练网络,并且这些数据影响最终的控制效果。

发明内容

[0005] 本发明为现有技术存在的不足之处,提出一种移动机器人跟踪预定轨迹的启发式动态规划控制方法,以期能够在跟踪不同的预定轨迹时不需要对控制参数进行反复调试就能够获得较好的控制效果,从而能提高移动机器人的环境适应性和自主优化能力。

[0006] 本发明为解决技术问题,采用如下技术方案:

[0007] 本发明一种移动机器人跟踪预定轨迹的启发式动态规划控制方法的特点是按如下步骤进行:

[0008] 第一步、以移动机器人安装的GPS传感器主天线处为坐标原点,以移动机器人前进方向为x轴,以移动机器人的前进方向逆时针旋转90度方向为y轴,建立机器人坐标系xoy;以大地质心为原点,正东方向为X轴,以正北方向为Y轴,建立大地坐标系XOY;

[0009] 第二步、控制移动机器人运动,并将大地坐标系下移动机器人的位姿信息记录存储作为预定轨迹;所述位姿信息包括位置坐标和航向角;

[0010] 第三步、确定当前时刻 t_i 下移动机器人的位姿信息和预瞄点位姿信息,从而建立机器人坐标系下的轨迹跟踪误差状态方程;

[0011] 步骤3.1、定义 $(X_c(t_i), Y_c(t_i))$ 为移动机器人在大地坐标系XOY中当前时刻 t_i 下的位置坐标,定义航向角 $\phi_c(t_i)$ 为当前时刻 t_i 下移动机器人纵轴线与大地坐标系X轴的夹角,定义 $(X_c(t_i), Y_c(t_i), \phi_c(t_i))$ 为移动机器人的当前时刻 t_i 下的位姿信息;

[0012] 步骤3.2、设定预瞄距离为L,在预定轨迹中搜索大于预瞄距离L且距离移动机器人

最近的位姿信息,并将符合要求的位姿信息定义为预瞄点 O_p 记为 $(X_p(t_i), Y_p(t_i), \phi_p(t_i))$;

[0013] 步骤3.3、定义 $(x_e(t_i), y_e(t_i), \phi_e(t_i))$ 为当前时刻 t_i 下移动机器人在机器人坐标系 xoy 中的位姿误差,其中 $x_e(t_i)$ 为纵向偏差, $y_e(t_i)$ 为横向偏差, $\phi_e(t_i)$ 为航向角偏差,则利用式(1)建立轨迹跟踪误差状态方程:

$$[0014] \quad \begin{bmatrix} x_e(t_i) \\ y_e(t_i) \\ \phi_e(t_i) \end{bmatrix} = \begin{bmatrix} \cos \phi_c(t_i) & \sin \phi_c(t_i) & 0 \\ -\sin \phi_c(t_i) & \cos \phi_c(t_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_p(t_i) - X_c(t_i) \\ Y_p(t_i) - Y_c(t_i) \\ \phi_p(t_i) - \phi_c(t_i) \end{bmatrix} \quad (1)$$

[0015] 步骤3.4、利用式(2)得到当前时刻 t_i 下移动机器人与预瞄点 O_p 的横向偏差的变化率 $\dot{y}_e(t_i)$ 和航向角偏差的变化率 $\dot{\phi}_e(t_i)$:

$$[0016] \quad \begin{cases} \dot{y}_e(t_i) = -v_y(t_i) - L\omega(t_i) + v_x(t_i)\phi_e(t_i) \\ \dot{\phi}_e(t_i) = v_x(t_i)\rho(t_i) - \omega(t_i) \end{cases} \quad (2)$$

[0017] 式(2)中, $v_x(t_i)$ 为移动机器人当前时刻 t_i 下的纵向速度, $v_y(t_i)$ 为移动机器人当前时刻 t_i 下的侧向速度, $\omega(t_i)$ 为移动机器人当前时刻 t_i 下的角速度, $\rho(t_i)$ 为当前时刻 t_i 下预瞄点 O_p 处的道路曲率;

[0018] 第四步、计算当前时刻 t_i 下预瞄点 O_p 处位置的道路曲率 $\rho(t_i)$;

[0019] 步骤4.1、从所述预定轨迹中选择三个位姿信息,包括以预瞄点 O_p 作为第1个位姿信息、与移动机器人的位置坐标 $(X_c(t_i), Y_c(t_i))$ 距离相同的两个位姿信息A和B作为第2个和第3个位姿信息,其中位姿信息A与位置坐标 $(X_c(t_i), Y_c(t_i))$ 的距离为 $L-\delta$;位姿信息B与位置坐标 $(X_c(t_i), Y_c(t_i))$ 的距离为 $L+\delta$, δ 为所设定的间隔距离;

[0020] 步骤4.2、利用式(3)将选取的三个位姿信息进行坐标转换,从而得到当前时刻 t_i 下在机器人坐标系 xoy 中任意第 j 个位姿信息与移动机器人的位置坐标 $(X_c(t_i), Y_c(t_i))$ 的距离分别在 x 轴的投影 $x_{d,j}(t_i)$ 和 y 轴上的投影 $y_{d,j}(t_i)$ 以及当前时刻 t_i 下第 j 个位姿信息在机器人坐标系中的横坐标 $x_j(t_i)$ 和纵坐标 $y_j(t_i)$:

$$[0021] \quad \begin{cases} x_{d,j}(t_i) = \frac{R_{ave}\pi}{180}(Y_j(t_i) - Y_c(t_i)) \cos \frac{\pi X_j(t_i)}{180} \\ y_{d,j}(t_i) = \frac{R_{ave}\pi}{180}(X_j(t_i) - X_c(t_i)) \\ x_j(t_i) = y_{d,j}(t_i) \cos \phi_c(t_i) + x_{d,j}(t_i) \sin \phi_c(t_i) \\ y_j(t_i) = y_{d,j}(t_i) \sin \phi_c(t_i) - x_{d,j}(t_i) \cos \phi_c(t_i) \end{cases} \quad (3)$$

[0022] 式(3)中, $X_j(t_i), Y_j(t_i)$ 为选取的预定轨迹上的经纬度坐标, R_{ave} 为地球平均半径, $j=1,2,3$;

[0023] 步骤4.3、使用三次多项式对坐标转换后的三个位姿信息的位置坐标进行曲线拟合,从而得到当前时刻 t_i 下预瞄点 O_p 处的道路曲率 $\rho(t_i)$;

[0024] 第五步、建立移动机器人的马尔可夫决策过程模型:

[0025] 步骤5.1、定义移动机器人在当前时刻 t_i 下的状态向量为 $s(t_i) = (y_e(t_i), \phi_e(t_i), \omega(t_i), \rho(t_i))^T$,定义移动机器人在当前时刻 t_i 下的控制动作为角速度 $u(t_i)$;

[0026] 步骤5.2、利用式(4)计算当前时刻 t_i 下状态向量 $s(t_i)$ 的回报 $r(s(t_i))$;

[0027] $r(s(t_i)) = s(t_i)^T Q s(t_i) + (u(t_i) - \omega_p(t_i))^T R (u(t_i) - \omega_p(t_i))$ (4)

[0028] 式(4)中, Q 为半正定对角矩阵, R 为一正常数, $\omega_p(t_i)$ 为当前时刻 t_i 下预瞄点 O_p 处期望的角速度,且 $\omega_p(t_i) = v_x(t_i) \rho(t_i)$;

[0029] 步骤5.3、利用式(5)定义当前时刻 t_i 下状态向量 $s(t_i)$ 的性能指标函数值 $J(s(t_i))$:

[0030] $J(s(t_i)) = r(s(t_i), u(s(t_i))) + \gamma J(s(t_{i+1}))$ (5)

[0031] 式(5)中, $r(s(t_i), u(s(t_i)))$ 为当前时刻 t_i 下的立即回报值,其中 $s(t_i)$ 为当前时刻 t_i 下的状态向量, $u(s(t_i))$ 为当前时刻 t_i 下移动机器人的动作, γ 为折扣因子, $J(s(t_{i+1}))$ 为下一时刻 t_{i+1} 下的性能指标函数, $s(t_{i+1})$ 为下一时刻 t_{i+1} 下的移动机器人状态向量,并由式(6)得到:

[0032]
$$s(t_{i+1}) = \begin{bmatrix} y_e(t_i) \\ \phi_e(t_i) \\ \omega(t_i) \\ \rho(t_i) \end{bmatrix} + T_s \times \begin{bmatrix} -L \times \omega(t_i) + v_x(t_i) \times \phi_e(t_i) - v_y(t_i) \\ v_x(t_i) \times \rho(t_i) - \omega(t_i) \\ 0 \\ 0 \end{bmatrix} \quad (6)$$

[0033] 式(6)中, T_s 为控制周期,且 $T_s = t_{i+1} - t_i$;

[0034] 步骤5.4、利用式(7)建立当前时刻 t_i 下最优的控制动作 $u^*(s(t_i))$:

[0035]
$$u^*(s(t_i)) = \frac{-1}{2R} \gamma \left[\left\{ \frac{\partial s(t_{i+1})}{\partial u(s(t_i))} \right\}^T \frac{\partial J(s(t_{i+1}))}{\partial s(t_{i+1})} \right] + \omega_p(t_i) \quad (7)$$

[0036] 第六步、利用式(8)计算当前时刻 t_i 下的控制动作 $u(s(t_i))$ 并作为HDP算法中执行器的输出:

[0037] $u(s(t_i)) = \Phi_a(W_a(t_i) \times \sigma_a(V_a(t_i) \times s(t_i) + b_{a1}(t_i)) + b_{a2}(t_i))$ (8)

[0038] 式(8)中, $V_a(t_i)$ 为当前时刻 t_i 下神经网络从输入层到隐含层的第一权值, $b_{a1}(t_i)$ 为当前时刻 t_i 下神经网络从输入层到隐含层的第一偏置, $W_a(t_i)$ 为当前时刻 t_i 下隐含层到输出层的第一权值, $b_{a2}(t_i)$ 为当前时刻 t_i 下隐含层到输出层的第一偏置;

[0039] 利用式(9)计算当前时刻 t_i 下状态向量 $s(t_i)$ 的性能指标函数值 $J(s(t_i))$ 并作为HDP算法中评价器的输出:

[0040] $J(s(t_i)) = \Phi_c(W_c(t_i) \times \sigma_c(V_c(t_i) \times s(t_i) + b_{c1}(t_i)) + b_{c2}(t_i))$ (9)

[0041] 式(9)中, $V_c(t_i)$ 为当前时刻 t_i 下神经网络从输入层到隐含层的第二权值, $b_{c1}(t_i)$ 为当前时刻 t_i 下神经网络从输入层到隐含层的第二偏置, $W_c(t_i)$ 为当前时刻 t_i 下隐含层到输出层的第二权值, $b_{c2}(t_i)$ 为当前时刻 t_i 下隐含层到输出层的第二偏置;

[0042] 第七步、建立HDP算法中执行器网络和评价器网络的在线学习规则;

[0043] 步骤7.1、执行器网络在线学习规则:

[0044] 利用式(10)建立执行器误差函数 $E_a(s(t_i))$,并将最小化执行器误差函数 $E_a(s(t_i))$ 作为执行器网络的学习目标:

[0045]
$$E_a(s(t_i)) = \frac{1}{2} (u(s(t_i)) - u^*(s(t_i)))^2 \quad (10)$$

[0046] 定义中间变量 $\zeta_d(t_i) = u(s(t_i)) - u^*(s(t_i))$,并利用式(11)-式(14)更新当前时刻 t_i 下执行器网络的权值 $W_a(t_i)$, $V_a(t_i)$ 和偏置 $b_{a1}(t_i)$, $b_{a2}(t_i)$,从而得到下一时刻 t_{i+1} 下的执

行器网络的权值 $W_a(t_{i+1})$, $V_a(t_{i+1})$ 和偏置 $b_{a1}(t_{i+1})$, $b_{a2}(t_{i+1})$:

$$[0047] \quad W_a(t_{i+1}) = W_a(t_i) - \alpha_a \zeta_d(t_i) \frac{\partial u(s(t_i))}{\partial W_a(t_i)} \quad (11)$$

$$[0048] \quad V_a(t_{i+1}) = V_a(t_i) - \alpha_a \zeta_d(t_i) \frac{\partial u(s(t_i))}{\partial V_a(t_i)} \quad (12)$$

$$[0049] \quad b_{a1}(t_{i+1}) = b_{a1}(t_i) - \alpha_a \zeta_d(t_i) \frac{\partial u(s(t_i))}{\partial b_{a1}(t_i)} \quad (13)$$

$$[0050] \quad b_{a2}(t_{i+1}) = b_{a2}(t_i) - \alpha_a \zeta_d(t_i) \frac{\partial u(s(t_i))}{\partial b_{a2}(t_i)} \quad (14)$$

[0051] 式(11)-式(14)中, α_a 为执行器网络的学习率;

[0052] 步骤7.2、评价器网络在线学习规则:

[0053] 定义中间变量为 $\delta_d(s(t_i)) = J(s(t_i)) - r(s(t_i), u(s(t_i))) - \gamma J(s(t_{i+1}))$, 并利用式(15)建立评价器误差函数 $E_c(s(t_i))$, 并以最小化误差函数 $E_c(s(t_i))$ 作为评价器的学习目标:

$$[0054] \quad E_c(s(t_i)) = \frac{1}{2} \delta_d^2(s(t_i)) \quad (15)$$

[0055] 利用式(16)-式(19)更新当前时刻 t_i 下评价器网络的权值 $W_c(t_i)$, $V_c(t_i)$ 和偏置 $b_{c1}(t_i)$, $b_{c2}(t_i)$, 从而得到下一时刻 t_{i+1} 下的评价器网络的权值 $W_c(t_{i+1})$, $V_c(t_{i+1})$ 和偏置 $b_{c1}(t_{i+1})$, $b_{c2}(t_{i+1})$:

$$[0056] \quad W_c(t_{i+1}) = W_c(t_i) - \alpha_c \delta_d(t_i) \frac{\partial J(s(t_i))}{\partial W_c(t_i)} \quad (16)$$

$$[0057] \quad V_c(t_{i+1}) = V_c(t_i) - \alpha_c \delta_d(t_i) \frac{\partial J(s(t_i))}{\partial V_c(t_i)} \quad (17)$$

$$[0058] \quad b_{c1}(t_{i+1}) = b_{c1}(t_i) - \alpha_c \delta_d(t_i) \frac{\partial J(s(t_i))}{\partial b_{c1}(t_i)} \quad (18)$$

$$[0059] \quad b_{c2}(t_{i+1}) = b_{c2}(t_i) - \alpha_c \delta_d(t_i) \frac{\partial J(s(t_i))}{\partial b_{c2}(t_i)} \quad (19)$$

[0060] 式(16)-式(19)中, α_c 为评价器网络的学习率;

[0061] 第八步、由执行器网络得到的控制动作 $u(s(t_i))$, 以及当前移动机器人的前进速度, 若所述移动机器人为履带式机器人, 则利用式(20)得到移动机器人两侧电机的转速 $n_L(t_i)$, $n_R(t_i)$; 若所述移动机器人为轮式机器人, 则利用式(21)得到移动机器人前轮转向角:

$$[0062] \quad \begin{bmatrix} n_L(t_i) \\ n_R(t_i) \end{bmatrix} = \begin{bmatrix} 1/2\pi r_d & B/4\pi r_d \\ 1/2\pi r_d & -B/4\pi r_d \end{bmatrix} \begin{bmatrix} v_c \\ u(s(t_i)) \end{bmatrix} \quad (20)$$

[0063] 式(20)中, r_d 为移动机器人的驱动轮半径, B 为履带中心距, v_c 为移动机器人前进速度;

$$[0064] \quad \delta(t_i) = \frac{L_z u(s(t_i))}{v_c} \quad (21)$$

[0065] 式(21)中, $\delta(t_i)$ 为前轮转向角, L_z 为移动机器人轴距;

[0066] 第九步、判断当前时刻 t_i 下预瞄点 O_p 是否为预定轨迹中的最后一位置坐标,若是,则令角速度 $u(s(t_i))=0$,控制移动机器人减速并在达到预瞄点 O_p 时停止运动;否则,将 $i+1$ 赋值给 i ,并返回第三步顺序执行。

[0067] 与现有技术相比,本发明的有益效果在于:

[0068] 1、本发明首先建立了移动机器人跟踪预定轨迹控制问题的马尔科夫决策过程(MDP)模型,选取了移动机器人的横向偏差 y_e 、航向角偏差 ϕ_e 、当前角速度 ω_c 和预瞄点处的道路曲率 ρ 作为状态向量,综合误差性能指标和跟随稳定性指标设计了回报函数,然后提出了一种基于HDP算法的跟踪预定轨迹的控制方法,采用了三层前馈神经网络设计评价器和执行器,并推导了网络参数的在线学习规则,最后通过系统试验验证了本发明跟踪预定轨迹的性能,解决了传统的移动机器人跟踪预定轨迹的控制方法需要人工调校参数,缺乏自主优化能力的问题,实现了不需要对控制参数进行反复调试就能够获得较好的控制效果,提高了移动机器人路径跟随控制方法的环境适应性和自主优化能力;

[0069] 2、采用本发明移动机器人跟踪预定轨迹的控制方法,能够利用目标点相对机器人的位置,利用HDP算法确定移动机器人的角速度使得机器人跟踪预定轨迹,功能通用性强,且具有对道路环境自适应性;

[0070] 3、移动机器人对预定轨迹的跟踪能力,是评价移动机器人性能的重要指标之一,也是移动机器人能够安全、有效的完成预定任务的重要保障。采用本发明移动机器人跟踪预定轨迹的控制方法,综合考虑了误差性能指标和跟随稳定性指标,能够在实现移动机器人对预期轨迹的跟踪的同时也保证了移动机器人的安全性和稳定性;

[0071] 4、由于采用本发明方法在控制过程中仅需要目标点相对机器人的坐标,而并不局限于某一种固定的传感器和定位手段,获取目标点相对机器人的坐标,既可以使用GPS系统,也可以采用雷达、光学等传感器,因此本发明方法具有较好的普适性、通用性,能够适用于安装不同传感器的移动机器人;

[0072] 5、本发明基于启发式动态规划算法框架,通过对当前控制动作的评价性反馈信号优化控制动作,在跟随路径线形变化时,该方法不需要对控制参数进行调试就能够获得高精度的控制效果,具有较好的环境适应性和自主优化能力;

[0073] 6、由于本发明采用经典的三层前馈神经网络构建HDP算法中的执行器网络和评价器网络移动机器人跟踪预定轨迹的智能控制方法中的控制算法采用经典的三层前馈神经网络构建HDP算法框架,实现简单,鲁棒性和可靠性较高。

附图说明

[0074] 图1为本发明移动机器人跟踪预定轨迹的控制系统结构图;

[0075] 图2为本发明移动机器人跟踪预定轨迹的示意图;

[0076] 图3为本发明基于HDP算法的跟踪预定轨迹的控制方法结构图;

[0077] 图4为采用本发明移动机器人跟踪预定轨迹的智能控制方法对预定轨迹为直线的效果图;

[0078] 图5为采用本发明移动机器人跟踪预定轨迹的智能控制方法对预定轨迹为钝角转向曲线的效果图;

[0079] 图6为采用本发明移动机器人跟踪预定轨迹的智能控制方法对预定轨迹为锐角转向曲线的效果图。

具体实施方式

[0080] 本实施例以实验室的履带式机器人平台为例,具体说明如何采用本发明方法实现对预定轨迹的跟踪及其功能效果。

[0081] 本实施例中所采用的履带式机器人平台上,其长度为0.8m,宽度为0.57m,重量为44kg,履带中心距B为0.485m,主动轮半径r为0.2m。采用48V20Ah的锂电池供电,两侧的电机功率为400w。装有北斗星通公司提供的产品名称为BDM680的GPS传感器,能够获取机器人所在位置的经纬度、机器人速度信号,可以通过双天线定位提供移动平台的航向角信息,在使用差分定位的情况下,定位精度可以达到0.02m。该机器人的计算平台采用控汇智能公司提供的产品名称为i5-4500U的工控机和正点原子型号为STM32ZET6开发板组成,工控机实现计算控制动作;单片机通过机器人数学模型将动作分解为两侧电机的转速,并给两侧的电机发送指令。

[0082] 本实施例中,采用如图1所示的移动机器人跟踪预定轨迹的控制系统结构,一种移动机器人跟踪预定轨迹的启发式动态规划控制方法,采用等采样周期的计算机控制方式来实现对预定轨迹的跟踪,每个控制周期是按如下步骤进行:

[0083] 第一步、建立机器人坐标系:以机器人安装的GPS传感器主天线处为坐标原点,以机器人前进方向为x轴,以机器人的前进方向逆时针旋转90度方向为y轴,建立机器人坐标系xoy;运动过程中的坐标系如图2中所示,以大地质心为原点,以正东方向为X轴,以正北方向为Y轴,建立大地坐标系XOY;

[0084] 第二步、遥控控制移动机器人运动,将大地坐标系下移动机器人的位姿信息记录存储作为预定轨迹,位姿信息包括位置坐标和航向角;预定轨迹是由n个目标点组成的序列,目标点序列采用一 $n \times 3$ 维的数组描述,每一行的三个元素分别代表轨迹中一点的经纬度和航向角信息。

[0085] 第三步、确定当前时刻 t_i 下移动机器人的位姿信息和预瞄点位姿信息,从而建立机器人坐标系下的轨迹跟踪误差状态方程:

[0086] 步骤3.1、如图2中所示,定义 $(X_c(t_i), Y_c(t_i))$ 为移动机器人在大地坐标系XOY中当前时刻 t_i 下的位置坐标,定义航向角 $\phi_c(t_i)$ 为当前时刻 t_i 下移动机器人纵轴线与大地坐标系X轴的夹角,定义 $(X_c(t_i), Y_c(t_i), \phi_c(t_i))$ 为移动机器人的当前时刻 t_i 下的位姿信息;

[0087] 步骤3.2、设定预瞄距离为L,在预定轨迹中搜索大于预瞄距离L且距离移动机器人最近的位姿信息,并将符合要求的位姿信息定义为预瞄点 O_p 记为 $(X_p(t_i), Y_p(t_i), \phi_p(t_i))$;

[0088] 步骤3.3、定义 $(x_e(t_i), y_e(t_i), \phi_e(t_i))$ 为当前时刻 t_i 下移动机器人在机器人坐标系xoy中的位姿误差,其中 $x_e(t_i)$ 为纵向偏差, $y_e(t_i)$ 为横向偏差, $\phi_e(t_i)$ 为航向角偏差,则利用式(1)建立轨迹跟踪误差状态方程:

$$[0089] \begin{bmatrix} x_e(t_i) \\ y_e(t_i) \\ \phi_e(t_i) \end{bmatrix} = \begin{bmatrix} \cos \phi_c(t_i) & \sin \phi_c(t_i) & 0 \\ -\sin \phi_c(t_i) & \cos \phi_c(t_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_p(t_i) - X_c(t_i) \\ Y_p(t_i) - Y_c(t_i) \\ \phi_p(t_i) - \phi_c(t_i) \end{bmatrix} \quad (1)$$

[0090] 步骤3.4、利用式(2)得到当前时刻 t_i 下移动机器人与预瞄点 O_p 的横向偏差的变化率 $\dot{y}_e(t_i)$ 和航向角偏差的变化率 $\dot{\phi}_e(t_i)$ ：

$$[0091] \quad \begin{cases} \dot{y}_e(t_i) = -v_y(t_i) - L\omega(t_i) + v_x(t_i)\phi_e(t_i) \\ \dot{\phi}_e(t_i) = v_x(t_i)\rho(t_i) - \omega(t_i) \end{cases} \quad (2)$$

[0092] 式(2)中, $v_x(t_i)$ 为移动机器人当前时刻 t_i 下的纵向速度, $v_y(t_i)$ 为移动机器人当前时刻 t_i 下的侧向速度, $\omega(t_i)$ 为移动机器人当前时刻 t_i 下的角速度, $\rho(t_i)$ 为当前时刻 t_i 下预瞄点 O_p 处的道路曲率；

[0093] 第四步、计算当前时刻 t_i 下预瞄点 O_p 处位置的道路曲率 $\rho(t_i)$ ；

[0094] 步骤4.1、从预定轨迹中选择三个位姿信息,包括以预瞄点 O_p 作为第1个位姿信息、与移动机器人的位置坐标 $(X_c(t_i), Y_c(t_i))$ 距离相同的两个位姿信息A和B作为第2个和第3个位姿信息,其中位姿信息A与位置坐标 $(X_c(t_i), Y_c(t_i))$ 的距离为 $L-\delta$ ；位姿信息B与位置坐标 $(X_c(t_i), Y_c(t_i))$ 的距离为 $L+\delta$, δ 为所设定的间隔距离；

[0095] 步骤4.2、利用式(3)将选取的三个位姿信息进行坐标转换,从而得到当前时刻 t_i 下在机器人坐标系 $x_o y$ 中任意第 j 个位姿信息与移动机器人的位置坐标 $(X_c(t_i), Y_c(t_i))$ 的距离分别在 x 轴的投影 $x_{d,j}(t_i)$ 和 y 轴上的投影 $y_{d,j}(t_i)$ 以及当前时刻 t_i 下第 j 个位姿信息在机器人坐标系中的横坐标 $x_j(t_i)$ 和纵坐标 $y_j(t_i)$ ：

$$[0096] \quad \begin{cases} x_{d,j}(t_i) = \frac{R_{ave}\pi}{180}(Y_j(t_i) - Y_c(t_i)) \cos \frac{\pi X_j(t_i)}{180} \\ y_{d,j}(t_i) = \frac{R_{ave}\pi}{180}(X_j(t_i) - X_c(t_i)) \\ x_j(t_i) = y_{d,j}(t_i) \cos \phi_c(t_i) + x_{d,j}(t_i) \sin \phi_c(t_i) \\ y_j(t_i) = y_{d,j}(t_i) \sin \phi_c(t_i) - x_{d,j}(t_i) \cos \phi_c(t_i) \end{cases} \quad (3)$$

[0097] 式(3)中, $X_j(t_i), Y_j(t_i)$ 为选取的预定轨迹上的经纬度坐标, R_{ave} 为地球平均半径, $j=1,2,3$ ；

[0098] 步骤4.3、使用三次多项式对坐标转换后的三个位姿信息的位置坐标进行曲线拟合,得到如式(4)所示的曲线方程：

$$[0099] \quad y = a_3x^3 + a_2x^2 + a_1x \quad (4)$$

[0100] 将转换后的轨迹点分别代入上式计算出 a_3, a_2, a_1 ,则预瞄点 O_p 位置的道路曲率 $\rho(t_i)$ 为： $\rho(t_i) = 3a_3x_1^2(t_i) + 2a_2x_1(t_i) + a_1$ ；

[0101] 第五步、建立移动机器人的马尔可夫决策过程模型：

[0102] 步骤5.1、针对本实施例履带机器人跟踪预定轨迹的问题,定义履带机器人在当前时刻 t_i 下的状态向量为 $s(t_i) = (y_e(t_i), \phi_e(t_i), \omega(t_i), \rho(t_i))^T$,定义履带机器人在当前时刻 t_i 下的控制动作为角速度 $u(t_i)$ ；

[0103] 步骤5.2、跟踪预定轨迹的过程中,不但期望减小横向偏差和航向角偏差,也期望尽量减少控制动作波动,因此利用式(5)计算当前时刻 t_i 下状态向量 $s(t_i)$ 的回报 $r(s(t_i))$ ：

$$[0104] \quad r(s(t_i)) = s(t_i)^T Q s(t_i) + (u(t_i) - \omega_p(t_i))^T R (u(t_i) - \omega_p(t_i)) \quad (5)$$

[0105] 式(5)中, Q 为半正定对角矩阵,本实施例中 $Q = \text{diag}\{2, 1, 0, 0\}$,表征了对横向偏差

和航向角偏差指标的权重;R为一正常数,本实施例中R=5,表征了对当前控制动作补偿的权重; $\omega_p(t_i)$ 为当前时刻 t_i 下预瞄点 O_p 处期望的角速度,且 $\omega_p(t_i) = v_x(t_i) \rho(t_i)$;

[0106] 步骤5.3、利用式(6)定义当前时刻 t_i 下状态向量 $s(t_i)$ 的性能指标函数 $J(s(t_i))$:

$$[0107] \quad J(s(t_i)) = r(s(t_i), u(s(t_i))) + \gamma J(s(t_{i+1})) \quad (6)$$

[0108] 式(6)中, $r(s(t_i), u(s(t_i)))$ 为当前时刻 t_i 下的立即回报值,其中 $s(t_i)$ 为当前时刻 t_i 下的状态向量, $u(s(t_i))$ 为当前时刻 t_i 下移动机器人的动作, γ 为折扣因子,本实施例中 $\gamma = 0.9$, $J(s(t_{i+1}))$ 为下一时刻 t_{i+1} 下的性能指标函数, $s(t_{i+1})$ 为下一时刻 t_{i+1} 下的移动机器人状态向量,并由式(7)得到:

$$[0109] \quad s(t_{i+1}) = \begin{bmatrix} y_e(t_i) \\ \phi_e(t_i) \\ \omega(t_i) \\ \rho(t_i) \end{bmatrix} + T_s \times \begin{bmatrix} -L \times \omega(t_i) + v_x(t_i) \times \phi_e(t_i) - v_y(t_i) \\ v_x(t_i) \times \rho(t_i) - \omega(t_i) \\ 0 \\ 0 \end{bmatrix} \quad (7)$$

[0110] 式(7)中, T_s 为控制周期,且 $T_s = t_{i+1} - t_i$;

[0111] 步骤5.4、利用式(8)建立当前时刻 t_i 下最优的控制动作 $u^*(s(t_i))$:

$$[0112] \quad u^*(s(t_i)) = \frac{-1}{2R} \gamma \left[\left\{ \frac{\partial s(t_{i+1})}{\partial u(s(t_i))} \right\}^T \frac{\partial J(s(t_{i+1}))}{\partial s(t_{i+1})} \right] + \omega_p(t_i) \quad (8)$$

[0113] 第六步、采用HDP算法来求取上式(8)中的最优控制动作,构建如图3所示的控制框架。利用式(9)计算当前时刻 t_i 下的控制动作 $u(s(t_i))$ 并作为HDP算法中执行器的输出:

$$[0114] \quad u(s(t_i)) = \Phi_a(W_a(t_i) \times \sigma_a(V_a(t_i) \times s(t_i) + b_{a1}(t_i)) + b_{a2}(t_i)) \quad (9)$$

[0115] 式(9)中, $V_a(t_i)$ 为当前时刻 t_i 下神经网络从输入层到隐含层的第一权值, $b_{a1}(t_i)$ 为当前时刻 t_i 下神经网络从输入层到隐含层的第一偏置, $W_a(t_i)$ 为当前时刻 t_i 下隐含层到输出层的第一权值, $b_{a2}(t_i)$ 为当前时刻 t_i 下隐含层到输出层的第一偏置;

[0116] 图(3)中评价器1和评价器2的结构及参数均相同,但是分别用于估计当前状态的性能指标函数值以及预测状态的性能指标函数值。

[0117] 利用式(10)计算当前时刻 t_i 下状态向量 $s(t_i)$ 的性能指标函数 $J(s(t_i))$ 并作为HDP算法中评价器的输出:

$$[0118] \quad J(s(t_i)) = \Phi_c(W_c(t_i) \times \sigma_c(V_c(t_i) \times s(t_i) + b_{c1}(t_i)) + b_{c2}(t_i)) \quad (10)$$

[0119] 式(10)中, $V_c(t_i)$ 为当前时刻 t_i 下神经网络从输入层到隐含层的第二权值, $b_{c1}(t_i)$ 为当前时刻 t_i 下神经网络从输入层到隐含层的第二偏置, $W_c(t_i)$ 为当前时刻 t_i 下隐含层到输出层的第二权值, $b_{c2}(t_i)$ 为当前时刻 t_i 下隐含层到输出层的第二偏置;

[0120] 第七步、建立HDP算法中执行器网络和评价器网络的在线学习规则,实质上是各网络权值的在线调整过程,这一过程是通过迭代运算逐步实现的。其中执行器网络和评价器网络的初始权值既可以使用其他的路径跟随控制方法在实际跟踪预定轨迹时使用本步骤推导的学习规则得到,也可以使用其他路径跟随控制方法的离线数据通过本步骤推导的学习规则得到。

[0121] 步骤7.1、执行器网络在线学习规则:

[0122] 执行器网络优化的目标是找到最优控制动作 $u^*(s(t_i))$ 使得目标函数值 $J(s(t_i))$ 最小,利用式(11)建立执行器误差函数 $E_a(s(t_i))$,并将最小化执行器误差函数 $E_a(s(t_i))$ 作

为执行器网络的学习目标：

$$[0123] \quad E_a(s(t_i)) = \frac{1}{2}(u(s(t_i)) - u^*(s(t_i)))^2 \quad (11)$$

[0124] 定义中间变量 $\zeta_d(t_i) = u(s(t_i)) - u^*(s(t_i))$ ，并利用式(12)-式(15)更新当前时刻 t_i 下执行器网络的权值 $W_a(t_i)$ ， $V_a(t_i)$ 和偏置 $b_{a1}(t_i)$ ， $b_{a2}(t_i)$ ，从而得到下一时刻 t_{i+1} 下的执行器网络的权值 $W_a(t_{i+1})$ ， $V_a(t_{i+1})$ 和偏置 $b_{a1}(t_{i+1})$ ， $b_{a2}(t_{i+1})$ ：

$$[0125] \quad W_a(t_{i+1}) = W_a(t_i) - \alpha_a \zeta_d(t_i) \frac{\partial u(s(t_i))}{\partial W_a(t_i)} \quad (12)$$

$$[0126] \quad V_a(t_{i+1}) = V_a(t_i) - \alpha_a \zeta_d(t_i) \frac{\partial u(s(t_i))}{\partial V_a(t_i)} \quad (13)$$

$$[0127] \quad b_{a1}(t_{i+1}) = b_{a1}(t_i) - \alpha_a \zeta_d(t_i) \frac{\partial u(s(t_i))}{\partial b_{a1}(t_i)} \quad (14)$$

$$[0128] \quad b_{a2}(t_{i+1}) = b_{a2}(t_i) - \alpha_a \zeta_d(t_i) \frac{\partial u(s(t_i))}{\partial b_{a2}(t_i)} \quad (15)$$

[0129] 式(12)-式(15)中， α_a 为执行器网络的学习率；

[0130] 步骤7.2、评价器网络在线学习规则：

[0131] 评价器网络的优化目标是对最优性能指标函数值 $J^*(s(t_i))$ 的近似，定义中间变量为 $\delta_d(s(t_i)) = J(s(t_i)) - r(s(t_i), u(s(t_i))) - \gamma J(s(t_{i+1}))$ ，并利用式(16)建立评价器误差函数 $E_c(s(t_i))$ ，并以最小化误差函数 $E_c(s(t_i))$ 作为评价器的学习目标：

$$[0132] \quad E_c(s(t_i)) = \frac{1}{2} \delta_d^2(s(t_i)) \quad (16)$$

[0133] 利用式(17)-式(20)更新当前时刻 t_i 下评价器网络的权值 $W_c(t_i)$ ， $V_c(t_i)$ 和偏置 $b_{c1}(t_i)$ ， $b_{c2}(t_i)$ ，从而得到下一时刻 t_{i+1} 下的评价器网络的权值 $W_c(t_{i+1})$ ， $V_c(t_{i+1})$ 和偏置 $b_{c1}(t_{i+1})$ ， $b_{c2}(t_{i+1})$ ：

$$[0134] \quad W_c(t_{i+1}) = W_c(t_i) - \alpha_c \delta_d(t_i) \frac{\partial J(s(t_i))}{\partial W_c(t_i)} \quad (17)$$

$$[0135] \quad V_c(t_{i+1}) = V_c(t_i) - \alpha_c \delta_d(t_i) \frac{\partial J(s(t_i))}{\partial V_c(t_i)} \quad (18)$$

$$[0136] \quad b_{c1}(t_{i+1}) = b_{c1}(t_i) - \alpha_c \delta_d(t_i) \frac{\partial J(s(t_i))}{\partial b_{c1}(t_i)} \quad (19)$$

$$[0137] \quad b_{c2}(t_{i+1}) = b_{c2}(t_i) - \alpha_c \delta_d(t_i) \frac{\partial J(s(t_i))}{\partial b_{c2}(t_i)} \quad (20)$$

[0138] 式(17)-式(20)中， α_c 为评价器网络的学习率；

[0139] 第八步、由执行器网络得到的控制动作 $u(s(t_i))$ ，以及当前移动机器人的前进速度，若移动机器人为履带式机器人，则采用差速转向的方式，从而利用式(20)得到移动机器人两侧电机的转速 $n_L(t_i)$ ， $n_R(t_i)$ ；若移动机器人为轮式机器人，则采用前轮转向的方式，从而利用式(21)得到移动机器人前轮转向角：

$$[0140] \quad \begin{bmatrix} n_L(t_i) \\ n_R(t_i) \end{bmatrix} = \begin{bmatrix} 1/2\pi r_d & B/4\pi r_d \\ 1/2\pi r_d & -B/4\pi r_d \end{bmatrix} \begin{bmatrix} v_c \\ u(s(t_i)) \end{bmatrix} \quad (20)$$

[0141] 式(20)中, r_d 为移动机器人的驱动轮半径, B 为履带中心距, v_c 为移动机器人前进速度;

$$[0142] \quad \delta(t_i) = \frac{L_z u(s(t_i))}{v_c} \quad (21)$$

[0143] 式(21)中, $\delta(t_i)$ 为前轮转向角, L_z 为移动机器人轴距;

[0144] 本实施例中使用的是履带机器人,其前进速度为 $v_c=0.3\text{m/s}$;则利用式(21)得到

$$\text{移动机器人两侧电机的转速} \begin{bmatrix} n_L(t_i) \\ n_R(t_i) \end{bmatrix} = \begin{bmatrix} 0.8 & 0.193 \\ 0.8 & -0.193 \end{bmatrix} \begin{bmatrix} 0.3 \\ u(s(t_i)) \end{bmatrix};$$

[0145] 第九步、判断当前时刻 t_i 下预瞄点 O_p 是否为预定轨迹中的最后一位置坐标,若是,则令角速度 $u(s(t_i))=0$,控制移动机器人减速并在达到预瞄点 O_p 时停止运动;否则,将 $i+1$ 赋值给 i ,并返回第三步顺序执行。

[0146] 从图4、图5和图6可以看出实际行驶轨迹虚点曲线1和预定轨迹实曲线2几乎重合,说明采用本发明方法有较高的跟踪精度,并且该方法仅通过历史跟随数据对评价器和执行器网络进行了离线训练,变换跟随路径时控制参数未做人工调试,仅通过HDP算法的迭代优化机制对控制网络进行在线优化,说明该方法具有较好的自主优化能力。

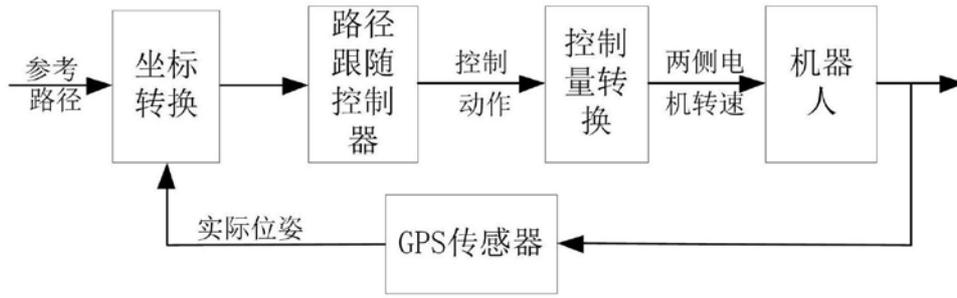


图1

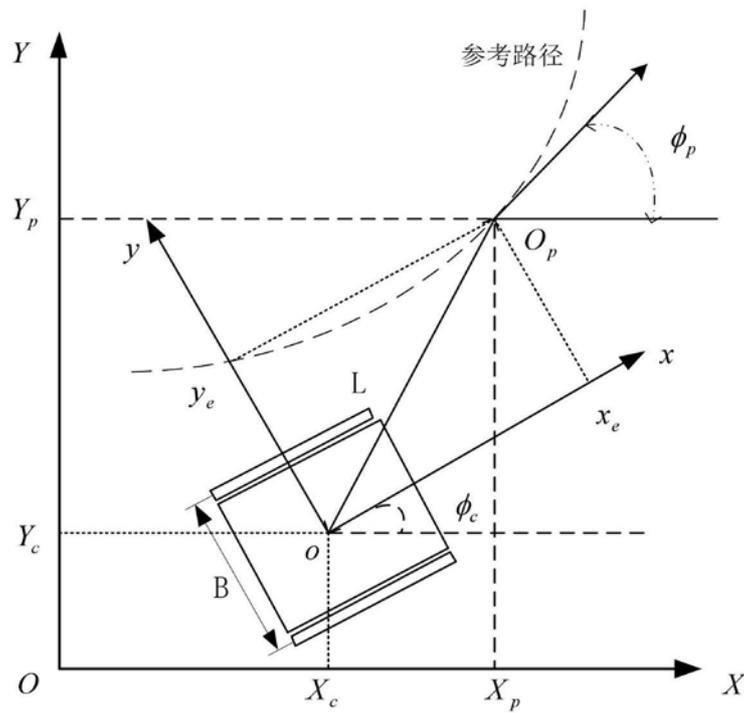


图2

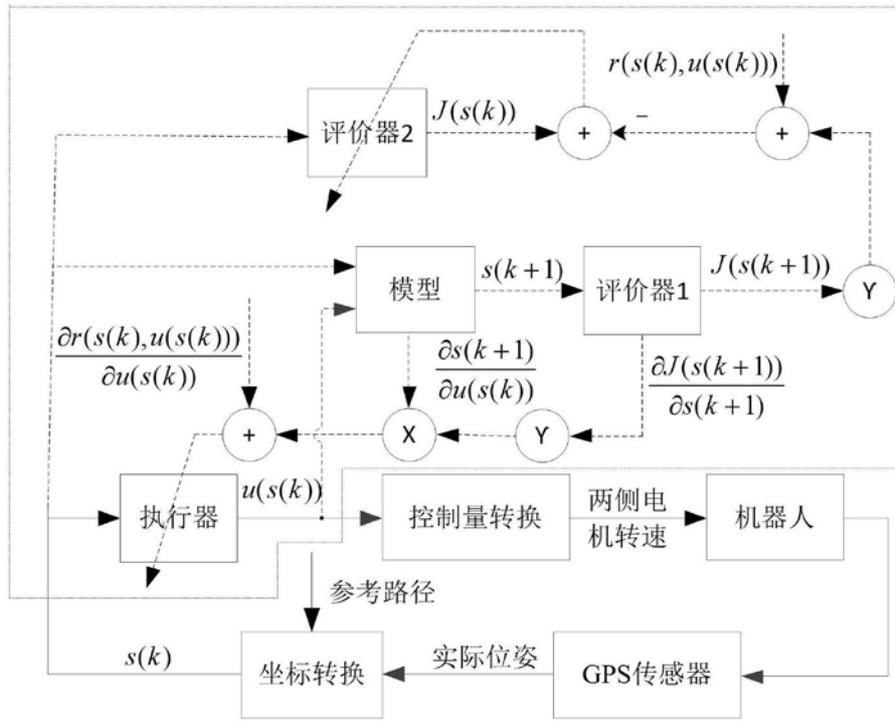


图3

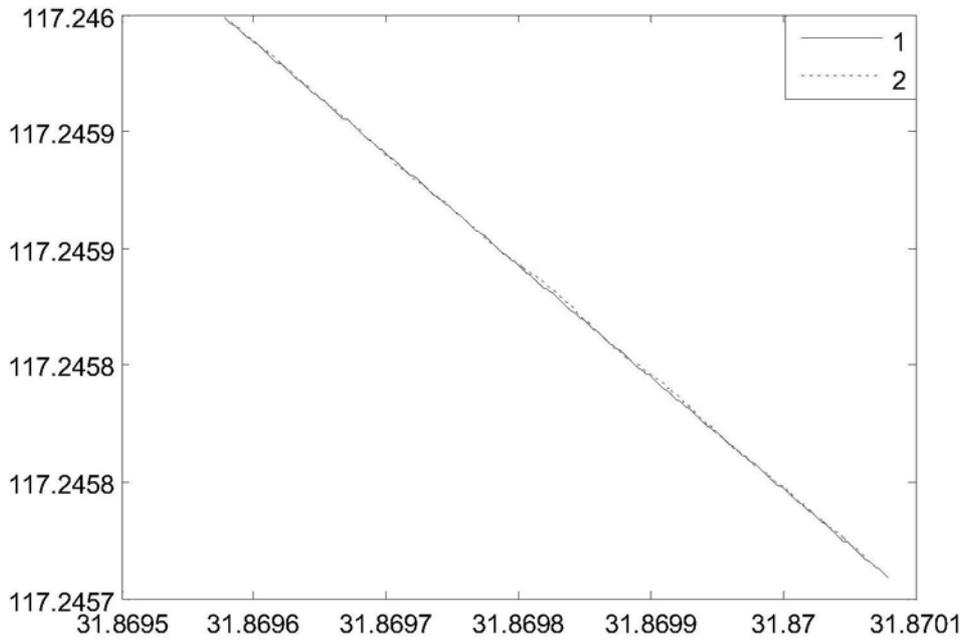


图4

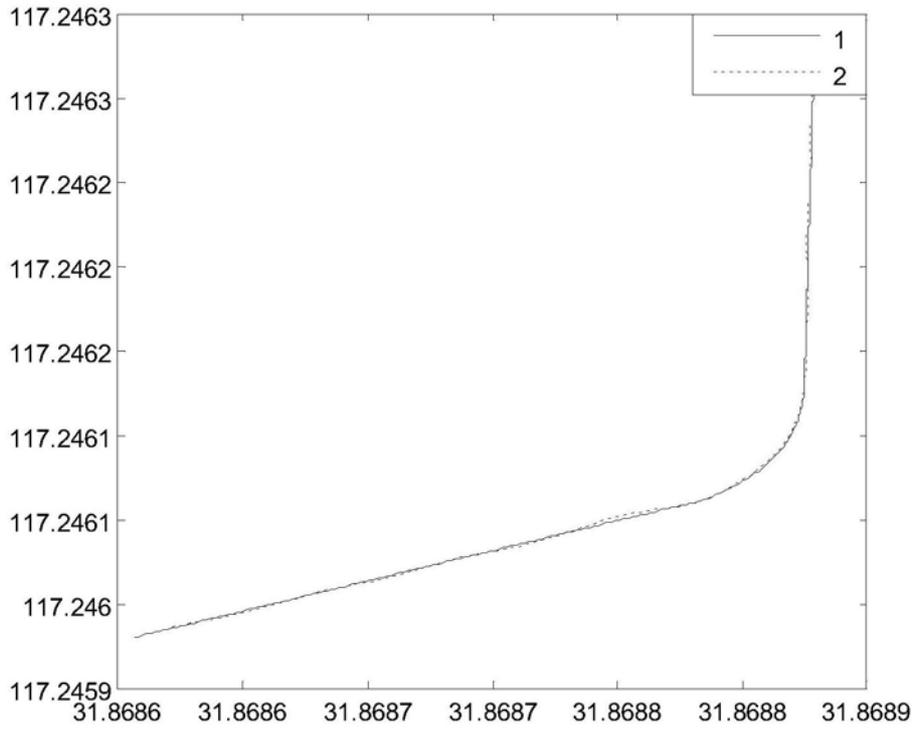


图5

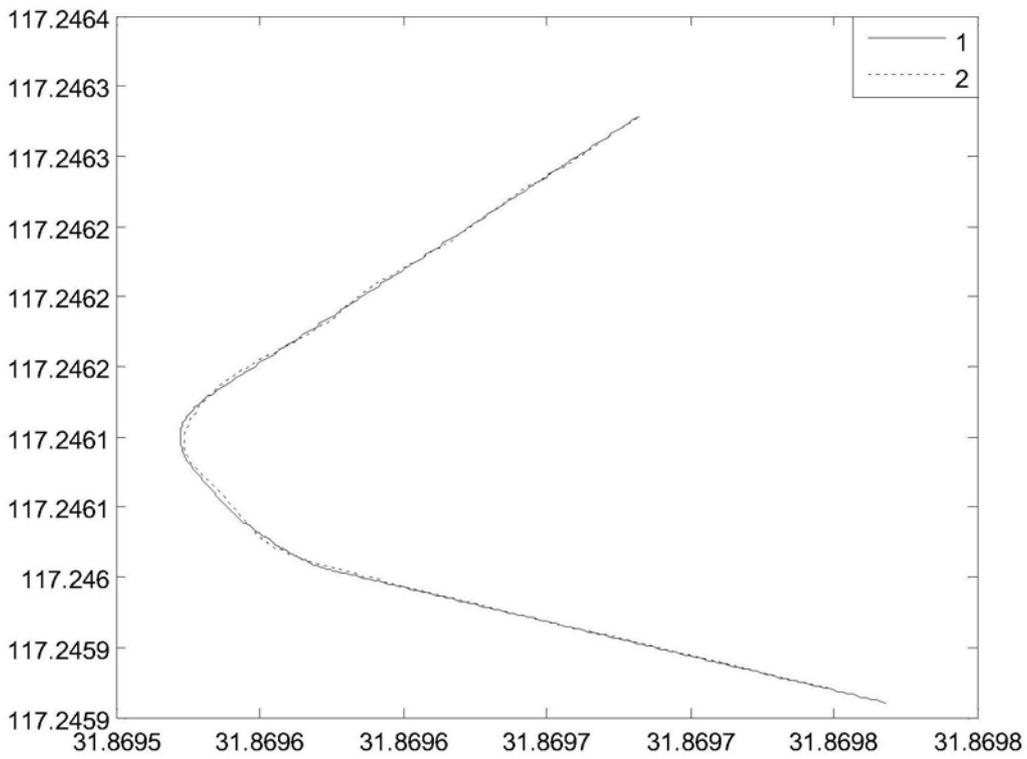


图6