

无线网络环境下移动机器人轨迹跟踪的认知控制分析与设计

徐君鹏, 尹逊和[†]

(北京交通大学 电子信息工程学院, 北京 100044)

摘要: 现有的网络控制系统领域中的研究通常先对网络环境的时延和丢包特性进行理想性假设, 然后设计对应的控制算法. 然而, 由于无线网络环境具有复杂的时延和丢包特性, 这些假设在无线网络控制系统(WNCS)运行过程中很难得到满足. 为了在不对网络环境的时延和丢包特性进行理想性假设的前提下设计控制系统, 本文受认知控制思想的启发, 在控制系统中加入了认知控制器. 认知控制器在感知-作用循环中学习产生认知作用的策略, 调节无线网络的媒体接入控制(MAC)层的重传次数上限和物理控制器的命令序列长度, 使控制系统可以主动地适应所处的无线网络环境. 本文以全向轮移动机器人为被控对象, 对使用认知控制器的WNCS和使用固定配置的WNCS的仿真结果进行比较. 仿真结果表明使用认知控制器调节MAC层的重传次数上限和物理控制器的命令序列长度, 可以提高WNCS对网络环境的时延和丢包的承受能力.

关键词: 网络控制系统; 模型预测控制; Q学习; 认知控制; 感知-作用循环

引用格式: 徐君鹏, 尹逊和. 无线网络环境下移动机器人轨迹跟踪的认知控制分析与设计. 控制理论与应用, 2020, 37(12): 2511 – 2524

DOI: 10.7641/CTA.2020.90869

Analysis and design of cognitive control for trajectory tracking of mobile robot in wireless network environment

XU Jun-peng, YIN Xun-he[†]

(School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China)

Abstract: Idealized assumptions about the character of delay and packet loss of network environment are first made in existing research in the domain of networked control system, and then a corresponding control algorithm is designed based on these assumptions. However, these assumptions hardly meet the realities during the operation of wireless networked control system (WNCS) due to the complex character of delay and packet loss in wireless network environment. In order to design the control system without making idealized assumptions of delay and packet loss in wireless network environment, inspired by the idea of cognitive control, a cognitive controller is added into the control system. The cognitive controller learns strategies of taking cognitive action in perception-action cycle, and the upper limit of the retry number of media access control(MAC) layer in wireless network and the length of command sequence of the physical controller are adjusted by the cognitive action of the cognitive controller. In this way, the control system can actively adapt to the wireless network environment. In this paper, an omnidirectional wheel mobile robot is selected as the plant. The simulation results of WNCS using cognitive controller and WNCS using fixed configuration are compared. Simulation results show that the cognitive controller can improve the ability of WNCS to overcome delay and packet loss in the network environment by adjusting the retry limit of the MAC layer and the length of command sequence of the physical controller.

Key words: networked control systems; model predictive control; Q learning; cognitive control; perception-action cycle

Citation: XU Junpeng, YIN Xunhe. Analysis and design of cognitive control for trajectory tracking of mobile robot in wireless network environment. *Control Theory & Applications*, 2020, 37(12): 2511 – 2524

1 引言

网络控制系统是由各个智能节点(传感器节点、控制器节点和执行器节点)通过共享的通信网络连接构

成的分布式控制系统. 与传统的点对点控制系统相比, 通信网络的引入降低了控制系统的维护成本, 提高了控制系统的灵活性. 因此网络控制系统广泛地应用在

收稿日期: 2019-10-17; 录用日期: 2020-07-31.

[†]通信作者. E-mail: xhyin@bjtu.edu.cn.

本文责任编辑: 黄攀峰.

中央高校基本科研业务费专项资金项目(2018YJS022)资助.

Supported by the Fundamental Research Funds for the Central Universities (2018YJS022).

工业控制、智能电网、汽车等实际系统中^[1-3]。通信网络为控制系统带来便利的同时也引入了网络诱导时延、数据包丢失、数据包乱序等问题。这些问题降低了控制系统的性能,甚至让控制系统不稳定。

对于通信网络带来的时延、丢包等问题,现有的文献已经提出了很多方法。文献[4]针对网络环境的丢包问题,使用改进的模型预测控制算法来控制焦炭炉的燃烧室压力,但是没有考虑网络时延对控制系统的影响。文献[5]假设网络的连续数据包丢失数量小于预先设定的上界,设计预测控制器来处理网络的丢包问题。文献[6]假设无线网络的时延小于设定的采样周期,设计模糊滑模控制器控制机器人跟踪给定的参考轨迹。文献[7]假设网络时延为常数,设计了自适应模糊控制器来控制非线性的被控对象。文献[8]假设网络时延小于设定的上界,使用切换控制方法设计了控制器,但是没有考虑网络的丢包问题。文献[9-10]把网络的丢包和数据包乱序考虑为往返时延(round trip time),假设网络的往返时延小于一个设定的上界,使用预测控制方法对往返时延进行补偿。

上述文献首先对网络环境的时延和丢包特性进行理想性假设,然后设计对应的控制算法,所以能够较好地控制被控对象。然而无线网络环境的时延和丢包特性具有不确定性,在控制系统运行过程中难以保证所做的假设得到满足。在2012年,Simon Haykin提出了工程意义下的认知控制思想^[11]。他认为传统的控制方法在结构化的环境(structured environment)中,可以很好地完成控制任务,但是在非结构化的环境(unstructured environment)中,传统的控制方法的性能不够好,而认知控制能够更好地在非结构化的环境中控制被控对象^[11]。

在本文中,如果网络的时延和丢包特性满足预先设定的理想性假设,例如,网络的时延总是小于一个采样周期或者总是小于设定的上界、网络连续丢包数量一定小于提前设定的上界,那么称这样的网络环境为结构化的网络环境;如果网络的时延和丢包特性不满足预先设定的理想性假设,例如,网络环境的时延和丢包特性是未知的,无法设置一个合理的上界来说明网络的时延和丢包特性,那么这样的网络环境为非结构化的网络环境。

到目前为止,工程领域的认知控制处于刚刚起步的阶段。文献[12-14]使用认知控制方法调节雷达发射的电波的波形参数,提高了雷达追踪目标的精度。文献[15-16]把认知控制思想应用到基于通信的列车控制系统中,提高了列车的控制性能,降低了列车的能耗。文献[17]以认知车为应用背景,概述了环境认知、驾驶员认知及驾驶员的刺激-反应过程、决策-动作过程的建模方法,突出了人在控制系统中的作用。文献[18]把无人机与环境的交互过程建模为脉冲微分

包含(impulse differential inclusions)模型,导出了认知无人机-环境系统的防碰撞稳定性条件,验证了该条件的有效性。文献[19]提出了基于认知博弈制导的无人机防碰撞方法,并通过仿真验证了所提方法的有效性。

与通常的设计控制算法的思路不同,本文是在没有预先对网络环境的时延和丢包特性进行理想性假设的条件下进行控制算法设计的。借鉴Simon Haykin提出的认知控制思想,本文定义了无线网络环境的状态(请见第5部分),并使用网络的状态表示信息间隙(information gap);然后在无线网络控制系统(wireless networked control system, WNCS)中加入了认知控制器,用认知控制器来调节无线网络的媒体接入控制(media access control, MAC)层的重传次数上限和物理控制器的命令序列长度。目的是通过调节MAC层的重传次数上限来减小WNCS的信息间隙,降低非结构化的网络环境的时延和丢包对控制系统的影响。通过调节物理控制器的命令序列长度(请见第6部分),来提高控制系统对非结构化的网络环境的时延和丢包的承受能力。

与现有的认知控制相关的工作^[15-16]相比,本文的创新点为:在无线网络控制系统中应用认知控制思想,把模型预测控制器(本文中称为物理控制器)的命令序列长度的调节和无线网络MAC层重传次数上限参数的调节加入到基于认知控制的WNCS的设计中,提高了WNCS对网络环境的时延和丢包的承受能力。

本文的结构如下:第2部分介绍了认知控制思想的基本内容;第3部分概括地描述了本文设计的WNCS的工作过程,它的工作细节在第4部分到第6部分给出;第4部分介绍了全向轮移动机器人的动态特性数学模型及其工作过程;第5部分描述了来自于无线网络环境的感知过程;第6部分展示了WNCS的控制部分;第7部分分析了基于认知控制思想设计的WNCS的仿真结果;第8部分对本文进行了总结。

2 认知控制思想简介

认知控制起源于神经科学和心理学^[20]。在2012年,Simon Haykin提出了工程意义下的认知控制思想^[11]。在文献[11]中,Simon Haykin指出当传统的控制系统在非结构化的或者高度不确定的环境中运行时,人类对控制系统的调节是必不可少的。如果把“认知能力”整合到控制系统中,那么这个“认知能力”可以替代或者减少人类对控制系统的干预。因此,在控制系统中加入认知控制器,代替人类对控制系统的调节作用,会使得在非结构化的环境中运行的控制系统的性能更好。

与传统的以控制被控对象的状态为目标的控制算法不同,认知控制思想的目标是通过向系统有向信息流的调整,来减小由具体任务定义的信息间隙^[11]。通

过减小信息间隙, 提高在非结构化的环境中运行的控制系统的性能. 认知控制系统的结构框图如图1所示. 认知控制系统由感知部分、控制部分、工作记忆(working memory)、系统和环境组成^[11]. 为避免混淆, 认知控制系统中的“系统”在本文中称为广义的被控对象. 值得注意的是, 图1中的认知控制系统是一个抽象的框架, 其中的箭头及结构可以根据具体的应用进行裁剪^[11]. 图1所示的认知控制系统的工作过程概述如下: 感知部分的感知器联合感知记忆, 对来自于广义的被控对象和环境的感知测量信息进行处理, 然后把处理后的信息反馈给认知控制系统的控制部分. 控制部分中的认知控制器利用感知部分反馈的信息进行学习, 并对环境、广义的被控对象及物理控制器产生认知作用(cognitive action). 物理控制器产生物理作用(physical action)作为广义的被控对象的输入. 广义的被控对象和环境由于认知控制器和物理控制器的激励而产生变化. 此时, 认知控制系统完成了一个感知-作用循环(perception-action cycle). 认知控制器在不断的感知-作用循环中学习产生认知作用的策略.

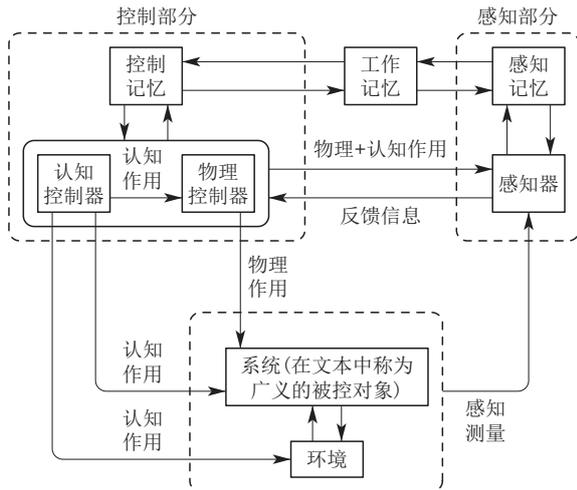


图 1 Simon Haykin提出的认知控制系统框架^[11]
Fig. 1 The cognitive control framework proposed by Simon Haykin^[11]

3 基于认知控制思想设计的WNCS架构

本文把认知控制思想应用到WNCS中, 设计的WNCS如图2所示. 图2中的WNCS由全向轮移动机器人(简称移动机器人, 即图 1 中的“广义的被控对象”)、无线网络环境、感知部分和控制部分组成. WNCS 的工作过程分为 2 个子过程: ① 感知过程; ② 控制过程. 这 2 个子过程构成了WNCS的基本工作过程: 感知-作用循环.

在概述WNCS的感知-作用循环之前, 首先说明WNCS使用的时间戳机制. 设控制系统的当前时刻 k 为传感器发送的数据包的时间戳, $k = 0, \Delta T, 2\Delta T, \dots$, 其中 ΔT 是传感器的采样间隔. 由于无线网络中存在时延和丢包的现象, 到达感知部分的数据包并不

是传感器在当前时刻 k 发送的, 而是在时刻 k 之前的某个触发时刻 k' 发送的, 所以感知部分收到的数据包的时间戳为 k' , 且有 $k' < k$. 同理, 设移动机器人的缓存模块收到的数据包的时间戳为 k'' , 则有 $k'' < k'$. 最后 k, k', k'' 三者的关系为: $k > k' > k''$.

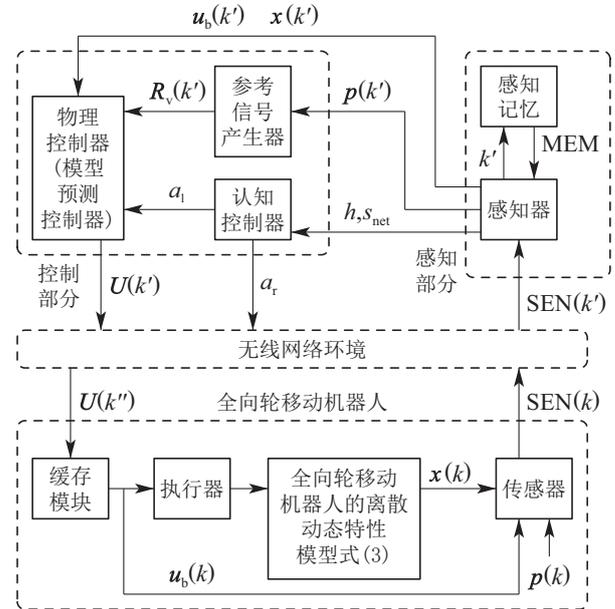


图 2 基于认知控制思想设计的WNCS

Fig. 2 WNCS designed based on the idea of cognitive control

以时间戳机制为基础, WNCS的感知-作用循环过程概述如下: ① 感知过程: 在触发时刻 k , 传感器把 $SEN(k) = \{u_b(k), x(k), p(k)\}$ 送入网络, 其中: $u_b(k)$ 为执行器的输入信号, $x(k)$ 为移动机器人的离散动态模型式(3)中的状态(请见第4部分), $p(k)$ 为移动机器人的位置及姿态. 传感器之所以发送信号 $u_b(k)$, 是因为物理控制器需要利用信号 $u_b(k)$ 和状态 $x(k)$ 计算移动机器人下一时刻的状态(请见第6.3节). 由WNCS的时间戳机制可知, 感知部分收到的数据包的时间戳为 k' , 所以设感知部分收到的数据包中的信息为 $SEN(k')$. 在感知部分收到信息 $SEN(k')$ 后, 它联合感知记忆存储的历史信息MEM, 利用时间戳 k' 得到网络的状态 s_{net} 和信息间隙 h (请见第 5.2 节). 感知器把 $SEN(k')$ 中的移动机器人的状态 $x(k')$ 、执行器在时刻 k' 的输入信号 $u_b(k')$ 送入物理控制器, 把移动机器人的位置及姿态 $p(k')$ 传给参考信号产生器; ② 控制过程: 认知控制器由信息间隙 h 、网络的状态 s_{net} 产生认知作用, 来调节物理控制器的命令序列长度 a_1 、无线网络MAC层的重传次数上限 a_r . 参考信号产生器产生速度参考信号 $R_v(k')$ 并把它传给物理控制器, 物理控制器由信号 $R_v(k'), x(k'), u_b(k'), a_1$ 产生控制命令序列 $U(k')$ 并把它发送到网络中. 由WNCS的时间戳机制可知, 移动机器人的缓存模块收到的数据包的时间戳为 k'' , 所以这里使用 $U(k'')$ 表示移动机器人收到

的信息. 在移动机器人的缓存模块收到 $\mathbf{U}(k'')$ 后, 把其中的控制命令 $\mathbf{u}_b(k)$ 传给执行器, 执行器驱动移动机器人运动.

本文设计的WNCS的工作细节在第4部分到第6部分给出, 其中: 第4部分介绍了移动机器人的动态特性数学模型及其工作细节; 第5部分描述了感知部分的工作细节; 第6部分展示了控制部分的工作细节.

4 全向轮移动机器人

本部分首先介绍全向轮移动机器人的动态特性数学模型, 然后给出全向轮移动机器人的缓存模块及传感器的工作过程.

本文使用文献[21]中的三轮移动机器人作为被控对象. 图3为移动机器人的结构示意图, $v(t)$ 和 $v_n(t)$ 分别表示全向轮移动机器人的前进速度和侧向速度的大小, $w(t)$ 为转动角速度的大小.

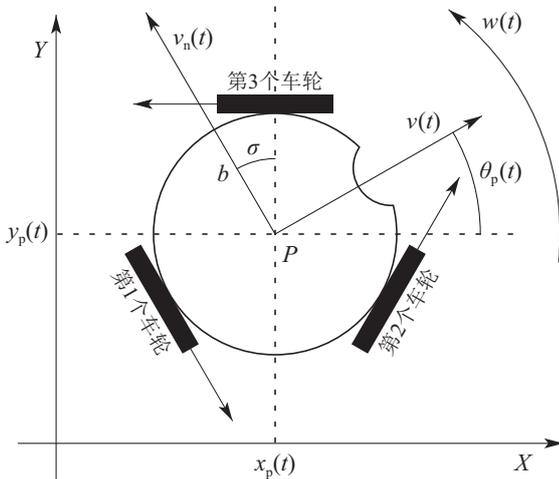


图3 三轮全向轮移动机器人的结构示意图^[21]

Fig. 3 The structure of omnidirectional three wheels mobile robot^[21]

图3中: σ 为移动机器人侧向速度与其中心点 P 到第3个车轮所在射线的夹角, σ 固定为 30° . $x_p(t)$, $y_p(t)$ 表示移动机器人中心点 P 的横坐标和纵坐标, $\theta_p(t)$ 表示移动机器人与地面坐标系 x 轴之间的夹角, 移动机器人的位置及姿态用 $\mathbf{p}(t) = [x_p(t) \ y_p(t) \ \theta_p(t)]^T$ 表示. 全向轮移动机器人的动态特性数学模型由式(1)给出^[21]:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{K}\text{sgn}\mathbf{x}(t), \\ \mathbf{y}(t) = \mathbf{x}(t), \end{cases} \quad (1)$$

其中: $\mathbf{x}(t) = [v(t) \ v_n(t) \ w(t)]^T$, $\mathbf{u}(t) = [u_1(t) \ u_2(t) \ u_3(t)]^T$ 为移动机器人3个马达的输入电压, $\mathbf{y}(t) = \mathbf{x}(t)$ 是移动机器人的模型的输出, $\text{sgn}(\cdot)$ 是符号函数, 矩阵 \mathbf{A} , \mathbf{B} , \mathbf{K} 的具体数值以及移动机器人的各个参数的数值请见文献[21]. 由于移动机器人的动态特性模型式(1)中带有非线性项 $\text{sgn}\mathbf{x}(t)$, 可令 $\mathbf{u}(t) = \mathbf{u}_f(t)$

+ $\mathbf{u}_b(t)$ 进行反馈线性化^[21], 其中 $\mathbf{u}_f(t) = -\mathbf{B}^{-1}\mathbf{K}\text{sgn}\mathbf{x}(t)$. 把 $\mathbf{u}(t)$ 代入式(1)可得

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}_b(t), \\ \mathbf{y}(t) = \mathbf{x}(t), \end{cases} \quad (2)$$

进而可以把式(2)变换为如式(3)所示的离散形式:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}_d\mathbf{x}(k) + \mathbf{B}_d\mathbf{u}_b(k), \\ \mathbf{y}(k) = \mathbf{x}(k), \end{cases} \quad (3)$$

其中: $\mathbf{A}_d = e^{\mathbf{A}\Delta T}$, $\mathbf{B}_d = \int_0^{\Delta T} e^{\mathbf{A}t} dt \mathbf{B}$, ΔT 为采样周期.

移动机器人的缓存模块用新得到的物理控制器的命令序列覆盖原本存储于其中的命令序列. 执行器为时间驱动, 在执行器的触发时刻 k , 设缓存模块中的命令序列为 $\mathbf{U}(k'') = \{\mathbf{u}_b(k''+1|k''), \mathbf{u}_b(k''+2|k''), \dots, \mathbf{u}_b(k''+a_1|k'')\}$, 其中 a_1 是物理控制器的命令序列长度(控制命令的计算过程请见第6.3节). 缓存模块根据以下方法取出控制命令序列 $\mathbf{U}(k'')$ 中的命令: ① 如果 $k''+a_1 < k$, 即 $\mathbf{U}(k'')$ 中不存在时刻 k 的命令, 那么使用零输入策略, 即 $\mathbf{u}_b(k) = \mathbf{0}$; ② 如果 $k''+a_1 \geq k$, 即 $\mathbf{U}(k'')$ 中存在时刻 k 的命令, 则 $\mathbf{u}_b(k) = \mathbf{u}_b(k|k'')$. 缓存模块取出命令 $\mathbf{u}_b(k)$ 后, 把它送给执行器来驱动移动机器人运动. 需要强调的是, 本文设计的WNCS的目标是控制移动机器人跟踪给定的正弦参考轨迹, 缓存模块采取“零输入策略”比“保持策略”具有更好的控制效果. 因为在无线网络环境的时延和丢包比较严重时, 零输入策略会让移动机器人失去动力, 它会停在参考轨迹附近, 而保持策略会继续驱动移动机器人运动. 这会使它严重地偏离正弦参考轨迹, 所以本文采用了零输入策略.

设移动机器人的传感器可以测得其位置及姿态信号 $\mathbf{p}(k) = [x_p(k) \ y_p(k) \ \theta_p(k)]^T$. 在触发时刻 k , 传感器得到执行器的输入信号 $\mathbf{u}_b(k)$ 、移动机器人的状态 $\mathbf{x}(k)$ 及移动机器人的位置及姿态 $\mathbf{p}(k)$ 后, 把信息 $\text{SEN}(k) = \{\mathbf{u}_b(k) \ \mathbf{x}(k) \ \mathbf{p}(k)\}$ 送入网络.

5 来自无线网络环境的感知过程

由于感知部分需要测量无线网络的丢包率及时延, 本部分首先介绍无线网络的丢包率和数据包重传机制的关系, 然后描述感知部分对无线网络环境的感知过程.

5.1 无线网络环境

本文使用的无线网络协议符合IEEE 802.11g标准. 与有线网络相比, 虽然无线网络的丢包现象比较严重, 但是无线网络的MAC层的重传机制可以有效地降低WNCS的丢包率. 因为一旦数据包在MAC层传输失败, MAC层会对丢失的数据包进行重传, 直到数据包传输成功或者重传次数达到设定的上界. 如果MAC

层成功地重传了数据包, 那么对于WNCS而言, 相当于这个数据包没有丢失, 所以MAC层重传数据包的机制会减小无线网络的丢包率。

但是MAC层的重传次数上限不是越高越好. 如果MAC层的重传次数上限过高, 那么MAC层会对丢失的数据包进行大量地重传, 这会增加网络的负担, 并有可能造成网络的拥塞. 如果网络产生了拥塞, 那么网络控制系统将会瘫痪. 大量地重传数据包还会增加数据包的时延, 而这对实时性要求较高的控制系统而言是无价值且不利的。

5.2 感知过程

感知部分的感知器只接收最新的数据包. 假设新得到的数据包中的信息为 $\text{SEN}(k') = \{\mathbf{u}_b(k'), \mathbf{x}(k'), \mathbf{p}(k')\}$. 在接收到数据包后, 感知器由现在的时间 t_c 及时间戳 k' 计算数据包从传感器到感知器的时延 $\tau_{\text{sp}} = t_c - k'$, 并把时间戳 k' 传给感知记忆. 感知记忆中存储的时间戳集合用PM表示. 在感知部分收到数据包后, 感知记忆把时间窗口 N 中的时间戳的集合 $\text{MEM} = \{i | k_c - N + 1 \leq i \leq k_c, i \in \text{PM}\}$ 传给感知器(若 $k_c - N + 1 < 0$, 则 $\text{MEM} = \{i | 0 \leq i \leq k_c, i \in \text{PM}\}$), 其中 k_c 为与现在的时间 t_c 最近且小于 t_c 的传感器触发时刻. 感知器利用MEM由式(4)计算WNCS的丢包率

$$p_{\text{loss}} = \begin{cases} 1 - \frac{\sum_{i=k_c-N+1}^{k_c} I(i)}{N}, & k_c - N + 1 \geq 0, \\ 1 - \frac{\sum_{i=0}^{k_c} I(i)}{k_c + 1}, & k_c - N + 1 < 0, \end{cases} \quad (4)$$

其中 $I(i)$ 用来指示时间戳 i 是否属于MEM; 若属于, 则 $I(i) = 1$, 否则 $I(i) = 0$.

有了估算的WNCS的丢包率 p_{loss} 及时延 τ_{sp} , 就可以对网络环境的状态进行表示. 设网络的状态为 $s_{\text{net}} = f(p_{\text{loss}}, \tau_{\text{sp}})$, 其中函数 f 是一个非减函数, 且有理想的网络状态 $s_{\text{net}}^* = f(0, 0) = 0$, 这里使用函数

$$s_{\text{net}} = f(p_{\text{loss}}, \tau_{\text{sp}}) = \begin{cases} p_{\text{loss}} + \tau_{\text{sp}}, & p_{\text{loss}} + \tau_{\text{sp}} < 1, \\ 1, & p_{\text{loss}} + \tau_{\text{sp}} \geq 1. \end{cases} \quad (5)$$

把 p_{loss} 和 τ_{sp} 映射到范围在0和1之间的网络状态 s_{net} . 一般地, 丢包率越高, 时延越长代表网络的状态越差. 式(5)表示的是时延、丢包到网络状态的映射关系, 意在表示网络的状态越好, 其对应的时延和丢包率越小; 网络的状态越差, 对应的时延和丢包率越大. 如果考虑网络其他的不利因素, 可以根据经验或者其他依据设计式(5), 表达网络的状态和不利因素的关系。

定义 1 信息间隙 h 为感知器得到的通信网络的状态 s_{net} 和理想的网络的状态 s_{net}^* 之间的差, 即 $h = s_{\text{net}} - s_{\text{net}}^*$.

这里定义的信息间隙本质上代表了网络环境的优劣. 信息间隙 h 越大, 它所代表的网络环境越恶劣. 为了提高网络控制系统的控制效果, 根据第5.1节对无线网络环境的分析, 可以通过认知控制器调节MAC层的重传次数上限来减小WNCS的丢包率和时延. 这样就减小了本部分定义的信息间隙, 而认知控制的目标即是减小由具体任务定义的信息间隙. 认知控制器的细节在第6.1节给出。

最后, 感知部分把执行器的输入 $\mathbf{u}_b(k')$ 、移动机器人的状态 $\mathbf{x}(k')$ 传给物理控制器, 把移动机器人的位置及姿态 $\mathbf{p}(k')$ 传给参考信号产生器, 把网络状态 s_{net} 和信息间隙 h 发送给认知控制器。

6 控制过程

本部分首先描述认知控制器的工作过程, 参考信号产生器的工作细节在第6.2节给出. 第6.3节展示了物理控制器的工作过程。

6.1 认知控制器

本文采用Q学习算法作为认知控制器的算法. 认知控制器的输入是感知部分传来的网络的状态 s_{net} 和信息间隙 h , 其中: 信息间隙 h 用来计算认知控制器的回报, 网络的状态 s_{net} 用来确定认知控制器的状态. 认知控制器产生的认知作用为 $a = (a_r, a_l)$.

由式(5)可知, s_{net} 越大表示网络的时延及丢包越严重. 本文把 s_{net} 的取值范围划分为4个区间, 每个区间对应认知控制器的一个状态, 如表1所示. 认知控制器可以根据其状态产生相应的认知作用, 调节WNCS的MAC层重传次数上限和物理控制器的命令序列长度, 提高WNCS对网络的时延和丢包的承受能力。

表 1 网络的状态 s_{net} 所在的区间、网络的时延和丢包的严重程度定性描述、认知控制器的状态的对应关系

Table 1 The relationship between the interval of network state s_{net} , qualitative description of severity of the network delay and packet loss and the state of cognitive controller

网络的状态所在区间	时延及丢包严重程度的定性描述	认知控制器的状态
[0, 0.2)	时延及丢包较轻	s_1
[0.2, 0.4)	时延及丢包适中	s_2
[0.4, 0.6)	时延及丢包较重	s_3
[0.6, 1]	时延及丢包严重	s_4

认知控制器产生的认知作用集合为 $A_c = \{(a_l, a_r) | a_l \in A_L, a_r \in A_R\}$, 其中: $A_R = \{2, 4, 6, 8\}$, $A_L = \{2, 4, 8, 16\}$. 这样设计的原因是: ① Q学习算法的动

作空间应该尽量的小,这有利于认知控制器快速地学习产生认知作用的策略;②对于实时性要求较高的控制系统而言,MAC层的重传次数上限不宜过高;③物理控制器的命令序列长度过长会增加物理控制器的计算负担,过短则难以起到预测的效果.为了减小动作空间的大小,集合 A_R 中的重传次数上限采用偶数次.为了使MAC层的重传次数上限不至于过高,设定最大的重传次数上限为8次.由于物理控制器的命令序列过长会增加物理控制器的计算负担,所以设置它的命令序列长度最长为16.同样为了减小动作空间的大小, A_L 中的物理控制器命令序列长度为2的指数次.

使用Q学习算法实现认知控制器的目的是减小第5.2节定义的信息间隙,所以设计Q学习算法的回报为

$$r = \frac{h' - h}{h + \mu}, \quad (6)$$

其中: h' 表示认知控制器在上一个感知-作用循环中得到的信息间隙; μ 的值为0.1,用来防止当 h 接近0时,回报的绝对值变得很大.这样设计回报的原因是:可以通过回报的符号(即,正:奖励;负:惩罚)使认知控制器产生合适的认知作用.

对于认知控制器在状态 s 做出的每个动作 $a = (a_1, a_r)$,认知控制器可由式(6)得到相应的回报 r 、由表1的对应关系得到状态 s' ,认知控制器利用式(7)^[22]来更新 $Q(s, a)$:

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a' \in A_c} Q(s', a') - Q(s, a)], \quad (7)$$

其中: α 是学习率, γ 是折扣因子, a' 是使 $Q(s', a')$ 的值最大的认知作用.最后,认知控制器由式(8)产生认知作用:

$$a = \arg \max_{a' \in A_c} Q(s, a'), \quad (8)$$

其中 A_c 为认知控制器的认知作用的集合.

6.2 参考信号产生器

顾名思义,参考信号产生器的功能是由移动机器人的位置及姿态 $p(k')$ 产生适合的物理控制器的速度参考信号 $R_v(k')$.

参考信号产生器中预先设定的正弦轨迹参考点如式(9)所示.

$$\begin{cases} \bar{x}_p(n) = n\Delta d, \\ \bar{y}_p(n) = \sin(n\pi\Delta d), \\ \bar{\theta}_p(n) = \arctan \frac{\bar{y}_p(n+1) - \bar{y}_p(n)}{\Delta d}, \end{cases} \quad (9)$$

其中: $\bar{x}_p(n)$ 和 $\bar{y}_p(n)$ 分别为轨迹参考点的横坐标和纵坐标, $\bar{\theta}_p(n)$ 为轨迹参考点对应的移动机器人与世界坐标系 x 轴的夹角, n 表示轨迹参考点的编号, Δd 是一个正常数.可以把上述轨迹参考点表示为向量的形式

$\bar{p}(n) = [\bar{x}_p(n) \ \bar{y}_p(n) \ \bar{\theta}_p(n)]^T$, $n = 1, 2, 3, \dots$. 由于本文所设计的WNCS的任务是控制移动机器人跟踪式(9)所示的参考轨迹,而式(3)中状态 $x(k) = [v(k) \ v_n(k) \ w(k)]^T$ 表示的是移动机器人的速度,所以需要利用式(9)中的轨迹参考点为物理控制器产生适合的速度参考信号.在参考信号产生器首次得到移动机器人的位置及姿态 $p(k')$ 后,它首先判断第1个轨迹参考点 $\bar{p}(1)$ 与移动机器人的距离是否小于阈值 l_s .若小于,则继续判断下一个轨迹参考点 $\bar{p}(2)$ 与移动机器人的距离是否小于阈值 l_s ,直到出现第 n_k' 个轨迹参考点与移动机器人的距离大于等于 l_s .如图4所示,这时参考信号产生器选择这个轨迹参考点及之后的 $N_p - 1$ 个轨迹参考点来计算物理控制器的速度参考信号.这 N_p 个轨迹参考点表示为 $\bar{p}(n_k')$, $\bar{p}(n_k' + 1)$, \dots , $\bar{p}(n_k' + N_p - 1)$.

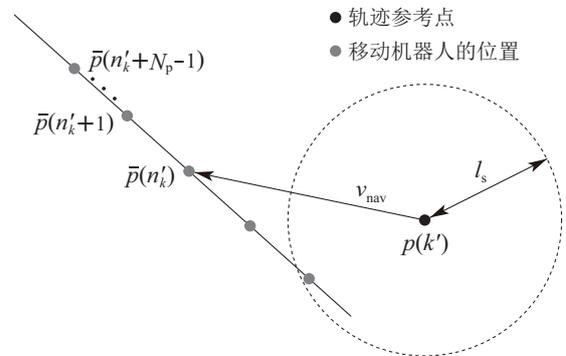


图4 参考信号产生器选择轨迹参考点

Fig. 4 Reference signal generator choosing trajectory reference points

需要注意的是,在下次参考信号产生器选择轨迹参考点时,它是接着从第 n_k' 个轨迹参考点开始判断移动机器人与轨迹参考点的距离是否小于 l_s .按照这种方法选出的轨迹参考点会在移动机器人的前方引导它前进.使用阈值 l_s 是由于网络的时延和丢包现象导致参考信号产生器得到的信息 $p(k')$ 不是实时信息.因为在参考信号产生器得到信息 $p(k')$ 时,移动机器人已经向前走了小段的距离,所以在选择轨迹参考点时,应该留有一定的裕量.在选定 N_p 个轨迹参考点后,参考信号产生器用式(10)和式(11)^[21]来计算物理控制器的参考信号

$$\Phi(k' + j - 1) = \arctan(\bar{y}_p(n_{k'} + j - 1) - y_p(k'), \bar{x}_p(n_{k'} + j - 1) - x_p(k')), \quad (10)$$

$$\begin{bmatrix} \bar{v}(k' + j | k') \\ \bar{v}_n(k' + j | k') \\ \bar{w}(k' + j | k') \end{bmatrix} = R_o \begin{bmatrix} v_{nav} \cos(\Phi(k' + j - 1)) \\ v_{nav} \sin(\Phi(k' + j - 1)) \\ \bar{\theta}_p(n_{k'} + j - 1) - \theta_p(k') \end{bmatrix}, \quad (11)$$

其中: $R_o = \begin{bmatrix} \cos(\theta_p(k')) & \sin(\theta_p(k')) & 0 \\ \sin(\theta_p(k')) & \cos(\theta_p(k')) & 0 \\ 0 & 0 & 1 \end{bmatrix}$ 为正交旋

转矩阵; v_{nav} 表示移动机器人的参考速度的大小, 为 0.5 m/s ^[21]. 移动机器人的参考速度由移动机器人的位置 $\mathbf{p}(k')$ 指向第 $n_k + j - 1$ 个轨迹参考点. 式(10)求出参考速度与世界坐标系 x 轴之间的夹角 $\phi(k' + j - 1)$; 然后式(11)根据 $\phi(k' + j - 1)$ 把参考速度分解为垂直方向上的速度, 其大小为 $v_{\text{nav}} \sin(\phi(k' + j - 1))$, 以及水平方向上的速度, 其大小为 $v_{\text{nav}} \cos(\phi(k' + j - 1))$; 然后用式(11)中的正交旋转矩阵 R_o 求出物理控制器的速度参考信号 $\bar{\mathbf{x}}(k' + j|k') = [\bar{v}(k' + j|k') \quad \bar{v}_n(k' + j|k') \quad \bar{w}(k' + j|k')]^T$, $j = 1, 2, \dots, N_p$. 通过迭代式(10)和式(11) N_p 次, 即可得到物理控制器的速度参考信号 $\mathbf{R}_v(k') = \{\bar{\mathbf{x}}(k' + 1|k'), \bar{\mathbf{x}}(k' + 2|k'), \dots, \bar{\mathbf{x}}(k' + N_p|k')\}$.

6.3 物理控制器

物理控制器中运行的控制算法是模型预测控制算法^[23], 预测模型为式(3). 在得到反馈的状态 $\mathbf{x}(k')$ 和 k' 时刻执行器的输入 $\mathbf{u}_b(k')$ 后, 物理控制器首先利用式(3)求出 $k' + 1$ 时刻的移动机器人的状态 $\mathbf{x}(k' + 1)$. 这是因为移动机器人在 k' 时刻已经输入了 $\mathbf{u}_b(k')$, 计算 k' 时刻的控制命令已经没有意义.

在得到信号 $\mathbf{R}_v(k')$, $\mathbf{x}(k')$, $\mathbf{u}_b(k')$, a_1 后, 物理控制器使用模型预测控制算法计算它的命令序列. 模型预测控制算法中预测时域 (predictive horizon) 的长度 (即参考信号序列的长度) 为 N_p ; 控制时域 (control horizon) 的长度 (即物理控制器命令序列的长度 a_1) 由认知控制器调节. 模型预测控制算法表示为^[23]

$$\begin{aligned} \min_{\mathbf{U}(k')} \sum_{j=1}^{N_p} \|\bar{\mathbf{x}}(k' + j|k') - \mathbf{x}(k' + j|k')\|_Q^2 + \\ \|\mathbf{u}_b(k' + j|k')\|_R^2, \\ \text{s.t.} \\ \begin{cases} \mathbf{x}(k' + j + 1|k') = A_d \mathbf{x}(k' + j|k') + B_d \mathbf{u}_b(k' + j|k'), \\ \|\mathbf{u}_i(k' + j|k')\| \leq 6, j = 1, 2, \dots, a_1, i = 1, 2, 3, \end{cases} \end{aligned} \quad (12)$$

其中: $Q = 100I$, $R = I$. 式(12)是一个二次规划问题, 可以通过“内点法”进行求解. 最后, 物理控制器得到的控制命令序列为 $\mathbf{U}(k') = \{\mathbf{u}_b(k' + 1|k'), \mathbf{u}_b(k' + 2|k'), \dots, \mathbf{u}_b(k' + a_1|k')\}$.

在无线网络的时延和丢包较轻时, 使用较短的物理控制器命令序列可以减小物理控制器的计算负担; 在无线网络的时延和丢包比较严重时, 增加物理控制器的命令序列长度对网络的时延和丢包具有一定的补偿作用. 所以, 物理控制器的命令序列长度需要认知控制器根据网络的状态来进行调节.

7 仿真结果及分析

本部分使用70%, 80%, 90%的MAC层丢包率及相应的网络时延来近似地表示非结构化的无线网络环

境. 在此网络环境下, 对比两种配置的WNCS的仿真结果, 这两种配置为: ① 使用固定的MAC层重传次数上限和固定的物理控制器命令序列长度 (请见第7.1节); ② 使用认知控制器调节MAC层重传次数上限和物理控制器命令序列长度 (请见第7.2节). 通过第7.1节和第7.2节中移动机器人跟踪正弦参考轨迹的控制效果及相对应的网络状态的比较, 展示了在WNCS中加入认知控制器的有效性, 即通过调节MAC层重传次数上限和物理控制器的命令序列长度, 减小了第5.2节定义的信息间隙, 降低了非结构化的网络环境对WNCS的控制性能的影响, 提高了WNCS对非结构化的网络环境的时延和丢包的承受能力.

本文使用Truetime工具箱进行WNCS的仿真. 在仿真中, 移动机器人的初始位置是世界坐标系的原点. 移动机器人与世界坐标系 x 轴的夹角初始值为 0° . 传感器的采样间隔 ΔT 为 0.01 s . 关于移动机器人的参数请见文献[21]. 无线网络的数据率为 54 Mb/s . 感知器估计丢包率的时间窗口 N 为 50 . 参考信号产生器中预先设定的参考轨迹为式(9)所示的正弦参考轨迹, 其中: Δd 为 0.005 m ; 参考信号产生器中的阈值 l_s 为 0.05 m ; 选取的轨迹参考点的个数 N_p 为 20 ; 式(7)中的学习率 α 为 0.1 ; 折扣因子 γ 为 0.9 .

由于运动控制系统要求较高的采样率, 而符合IEEE 802.11g标准的无线网络的数据率可以达到 54 Mb/s , 这为传感器 0.01 s 的采样间隔提供了保障. 在感知部分估计网络的丢包率时, 采用过长的时间窗口会使估计过程变慢; 如果时间窗口过短则导致估计的丢包率不准确, 所以选择 N 为 50 . 由于在参考轨迹上取足够多的轨迹参考点才能让这些参考点逼近连续的参考轨迹, 所以式(9)中 Δd 为 0.005 m . 移动机器人在一个采样周期中前进的距离 $d \approx v_{\text{nav}} \Delta T = 0.005 \text{ m}$. 由第6.2节所述, 这里选择 l_s 为 0.05 m 大约可以预留 10 个采样周期中移动机器人前进距离的裕量, 目的是保持轨迹参考点在移动机器人的前方引导它前进. 如果 l_s 选择的过大, 则在参考轨迹曲率较大的转角处, 移动机器人无法精确地跟踪参考轨迹; N_p 的数值过大会增加模型预测控制算法的计算负担, 又因为在模型预测控制算法中, N_p 需要大于最长的控制时域的长度, 即 a_1 的最大值 16 , 所以这里选择 N_p 为 20 . 在Q学习算法中, 折扣因子 γ 的取值范围是 0 到 1 , γ 越大, 未来的回报的权重越大. 如果选择的学习率 α 的值过大, 会使Q学习算法不稳定^[22], 所以这里选择 γ 为 0.9 , α 为 0.1 .

7.1 使用固定的 a_l , a_r

因为MAC层的重传机制可以降低WNCS的丢包率, 但是重传次数过大会增大网络的时延、负荷, 所以在集合 A_R 中选择 a_r 为 6 ; 物理控制器的命令序列长度过长会增大控制器的计算量, 过短则难以起到预测的

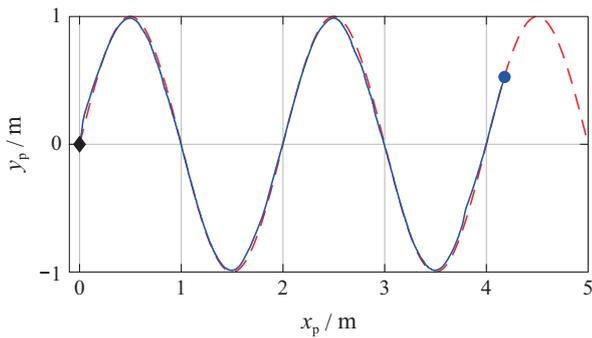
效果,所以在集合 A_L 中选择 a_1 为8.

下面将结合表2和图5-7,按照MAC层丢包率增大的顺序分析WNCS的仿真结果.

表2 MAC层丢包率、WNCS的丢包率、移动机器人与参考轨迹的距离均值的对应关系 ($a_1 = 8, a_r = 6$)

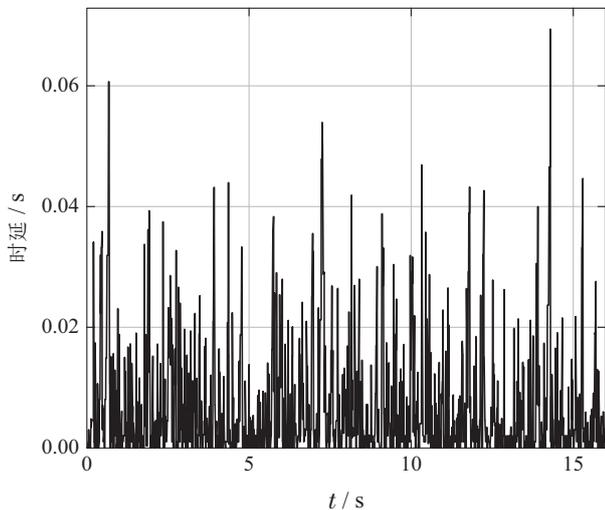
Table 2 The relationship between packet loss probability of MAC layer, packet loss probability of WNCS and average distance between the mobile robot and reference trajectory ($a_1 = 8, a_r = 6$)

MAC层的丢包率/%	WNCS的丢包率/%	移动机器人与参考轨迹的距离均值/m
70	12.59	0.0226
80	46.14	0.0392
90	96.84	—

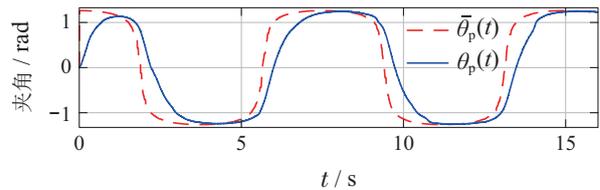
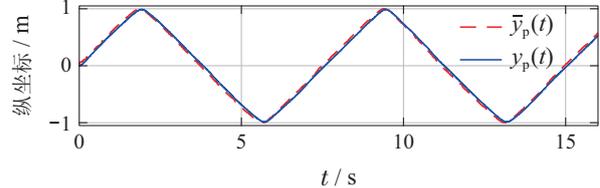
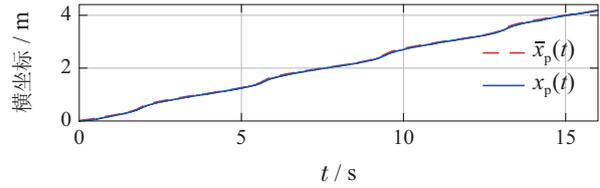


— — 移动机器人的参考轨迹
— — 移动机器人的实际轨迹
◆ 移动机器人的出发点
● 移动机器人的终止点

(a) 移动机器人的轨迹



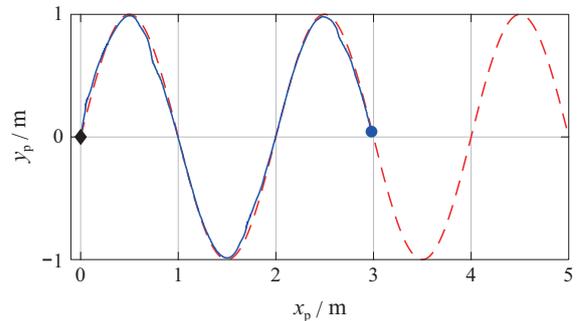
(b) 传感器到感知器的时延



(c) 移动机器人的横坐标、纵坐标及其与x轴的夹角

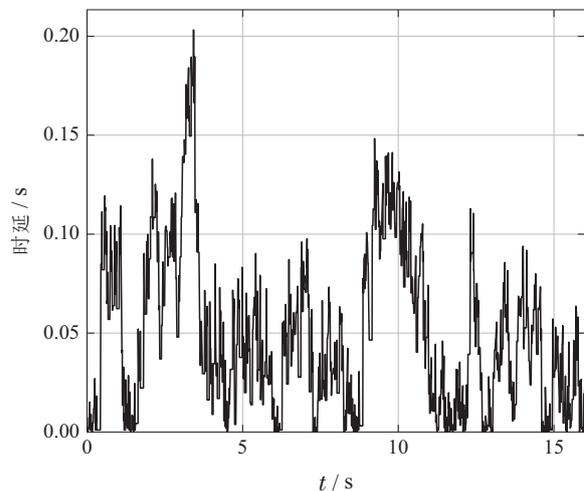
图5 移动机器人跟踪正弦参考轨迹的仿真结果($a_1 = 8, a_r = 6$, MAC层的丢包率设为70%)

Fig. 5 Simulation result that the mobile robot tracking sinusoidal reference trajectory ($a_1 = 8, a_r = 6$, and packet loss probability of the MAC layer is set as 70%)

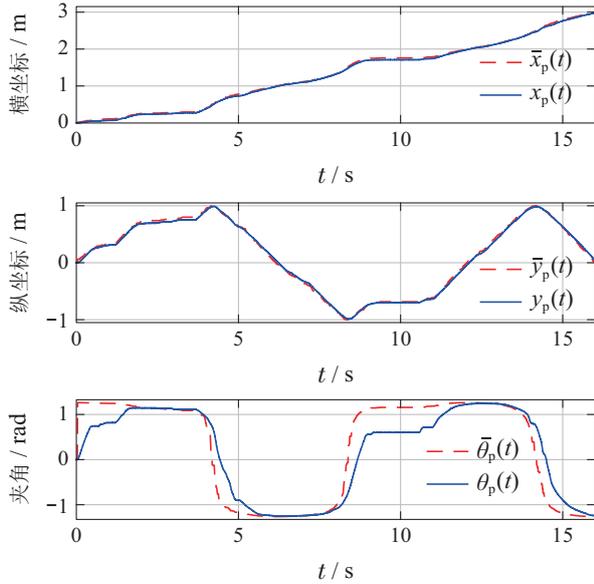


— — 移动机器人的参考轨迹
— — 移动机器人的实际轨迹
◆ 移动机器人的出发点
● 移动机器人的终止点

(a) 移动机器人的轨迹

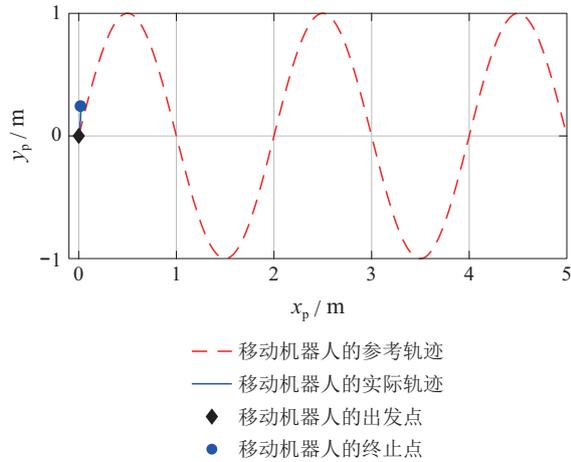


(b) 传感器到感知器的时延

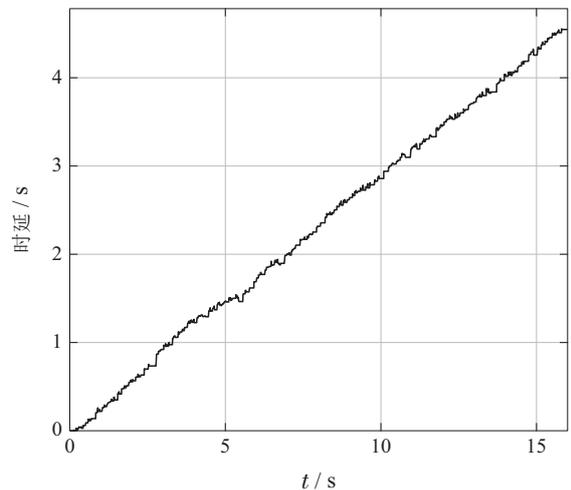


(c) 移动机器人的横坐标、纵坐标及其与x轴的夹角
图 6 移动机器人跟踪正弦参考轨迹的仿真结果($a_1 = 8$, $a_r = 6$, MAC层的丢包率设为80%)

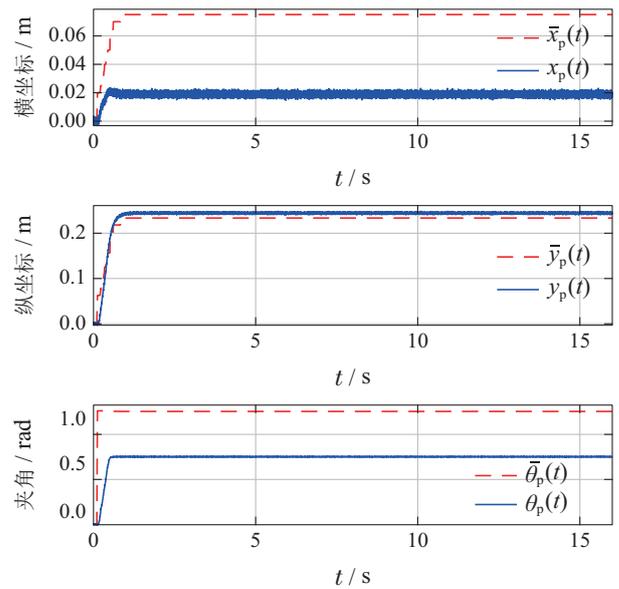
Fig. 6 Simulation result that the mobile robot tracking sinusoidal reference trajectory ($a_1 = 8$, $a_r = 6$, and packet loss probability of the MAC layer is set as 80%)



(a) 移动机器人的轨迹



(b) 传感器到感知器的时延



(c) 移动机器人的横坐标、纵坐标及其与x轴的夹角
图 7 移动机器人跟踪正弦参考轨迹的仿真结果($a_1 = 8$, $a_r = 6$, MAC层的丢包率设为90%)

Fig. 7 Simulation result that the mobile robot tracking sinusoidal reference trajectory ($a_1 = 8$, $a_r = 6$, and packet loss probability of the MAC layer is set as 90%)

从表2可以看出: 在MAC层丢包率为70%时, MAC层的重传机制可以把WNCS的丢包率降至12.59%; 移动机器人与参考轨迹的距离均值为0.0226 m.

在图5(a)中, 移动机器人可以令人满意地跟踪给定的正弦参考轨迹. 在图5(b)中, 传感器到感知器的时延低于0.07 s. 在图5(c)中, 移动机器人的横坐标 $x_p(t)$ 和纵坐标 $y_p(t)$ 可以较好地跟踪它们的参考值 $\bar{x}_p(t)$ 和 $\bar{y}_p(t)$, 没有出现由于时延和丢包使移动机器人停止前进的情况. 在图5(c)的第3个子图中, 在参考轨迹的转角处, $\bar{\theta}_p(t)$ 变化很快, 移动机器人与世界坐标系x轴的夹角 $\theta_p(t)$ 跟踪 $\bar{\theta}_p(t)$ 需要一定的时间. 在图5及之后的仿真图中, 在正弦参考轨迹曲率较大的转角处, 移动机器人与参考轨迹的距离有所增加. 这是由第6.2节中参考信号产生器产生速度参考信号的机制造成的, 即为了防止由于网络的时延或丢包造成移动机器人的轨迹参考点位于其后方, 使用移动机器人前方 $l_s = 0.05$ m的轨迹参考点来引导移动机器人前进. 而这0.05 m的裕量, 导致了在参考轨迹转角处, 移动机器人与参考轨迹的距离的增加.

在MAC层的丢包率为80%时, 由表2可知, 固定为6的MAC层重传次数上限虽然也可以降低WNCS的丢包率, 但是WNCS的丢包率已经达到了46.14%, 移动机器人与参考轨迹的距离均值也相应地增大到0.0392 m. 从图6(a)可以看出, 移动机器人依然可以跟踪参考轨迹. 由图6(b)可知, 大约在1 s附近、2 s到4 s、9 s到11 s, 无线网络的时延增加. 在图6(c)中相应的时

间段, $x_p(t), y_p(t), \theta_p(t)$ 的数值保持不变. 这说明因为无线网络状态变差, 导致移动机器人得不到物理控制器的命令, 使移动机器人停止了前进. 由图6的仿真结果可以看出, 无线网络MAC层丢包率增加到80%, 导致WNCS控制移动机器人跟踪参考轨迹的性能下降, 但WNCS依然可以完成移动机器人的控制任务.

在无线网络的MAC层丢包率为90%时, 由表2可知, WNCS的丢包率达到了96.84%. 这是由于大量的数据包的丢失使MAC层大量地重传丢失的数据包, 最终导致了网络的拥塞. 由图7(a)可以看出, WNCS已经无法控制移动机器人跟踪参考轨迹. 在图7(b)中, 网络的时延呈现出不可接受的线性增长. 图7(c)表明, 移动机器人在大约0.5 s时, 由于网络的拥塞而停止了运动. 所以, 对于MAC层丢包率为90%的情形, MAC层重传次数上限固定为6次、物理控制器命令序列长度固定为8的配置显然是不合适的.

综合分析表2和图5-7的仿真结果, 可以看出在无线网络MAC层的丢包率为70%时, 设置 a_r 固定为6可以显著地降低WNCS的丢包率. 但是在无线网络MAC层的丢包率为80%时, 设置 a_1 为6会导致WNCS对丢失的数据包进行大量地重传, 这加重了无线网络的负担. 长度固定为8的物理控制器命令序列不能有效地克服网络的时延和丢包, 导致移动机器人多次失去动力. MAC层丢包率为90%时, 使用固定配置的WNCS产生了网络拥塞. 所以, 使用固定的MAC层重传次数上限和物理控制器命令序列长度具有局限性.

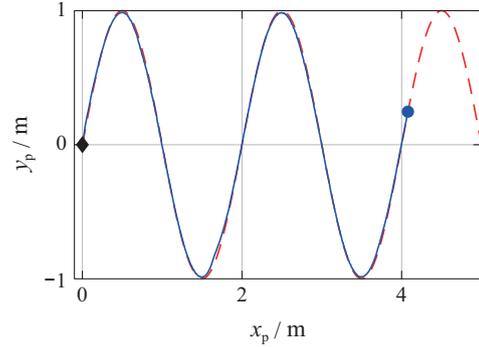
7.2 使用认知控制器调节 a_1, a_r

为了克服WNCS使用固定的MAC层重传次数上限和固定的物理控制器命令序列长度的缺点, 本文在WNCS中加入认知控制器来调节MAC层的重传次数上限和物理控制器的命令序列长度.

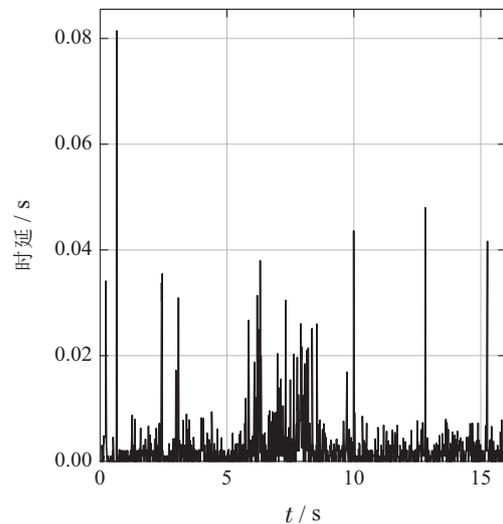
本文依然按照MAC层丢包率增大的顺序, 结合图8-10和表3对WNCS的仿真结果进行分析. 首先, 由表3中的数据可以看出, 在MAC层的丢包率为70%, 80%, 90%情形下, WNCS没有出现网络拥塞的状况. 这说明认知控制器通过调节MAC层重传次数上限和物理控制器命令序列长度避免了网络的拥塞.

在MAC层的丢包率为70%时, 表3中的WNCS的丢包率大于表2中的相应的丢包率. 表3中的移动机器人与参考轨迹的距离均值要比表2中相应的距离均值小0.0085 m. 对比图8(a)和图5(a)、图8(c)和图5(c)可以看出, 两种配置的WNCS控制移动机器人跟踪正弦参考轨迹的控制效果相差很小. 图8(b)中传感器到感知器的时延小于图5(b)中的时延; 图8(d)是认知控制器调节MAC层重传次数上限 a_r 和物理控制器命令序列长度 a_1 的过程. 从图8(d)可以看出, 在认知控制器的调节下, 大部分时间 a_1 的值为4, a_r 的值为4, 可见物理

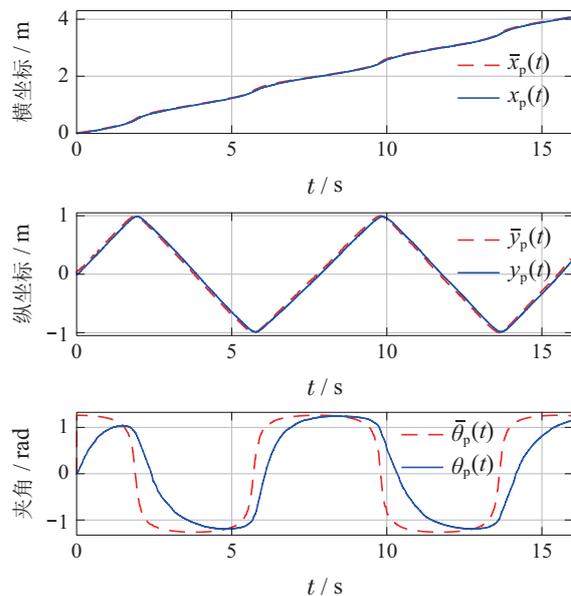
控制器的命令序列长度 a_1 为4足够克服WNCS的70%的MAC层丢包率及对应的时延.



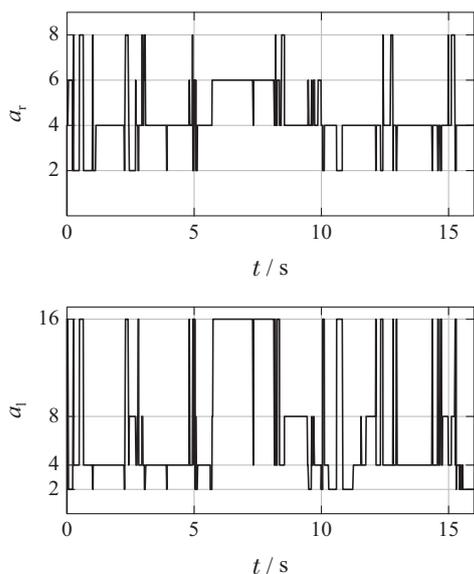
(a) 移动机器人的轨迹



(b) 传感器到感知器的时延



(c) 移动机器人的横坐标、纵坐标及其与x轴的夹角

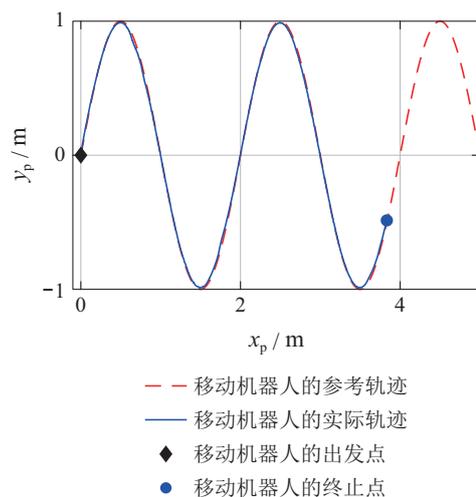


(d) 认知控制器产生的认知作用

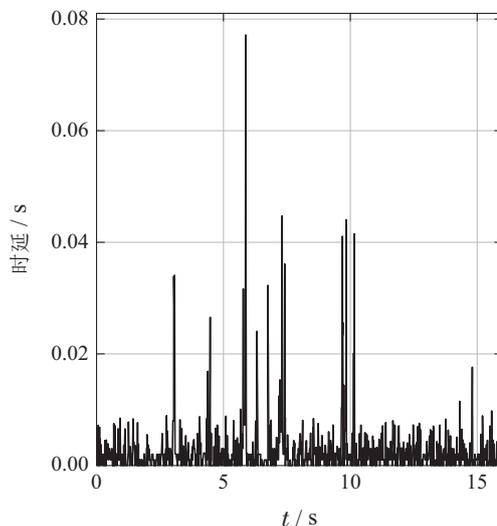
图 8 移动机器人跟踪正弦参考轨迹的仿真结果 (使用认知控制器调节 a_r 和 a_1 , MAC 层丢包率为 70%)

Fig. 8 Simulation result that the mobile robot tracking sinusoidal reference trajectory (a_r and a_1 are adjusted by cognitive controller, and packet loss probability of the MAC layer is set as 70%)

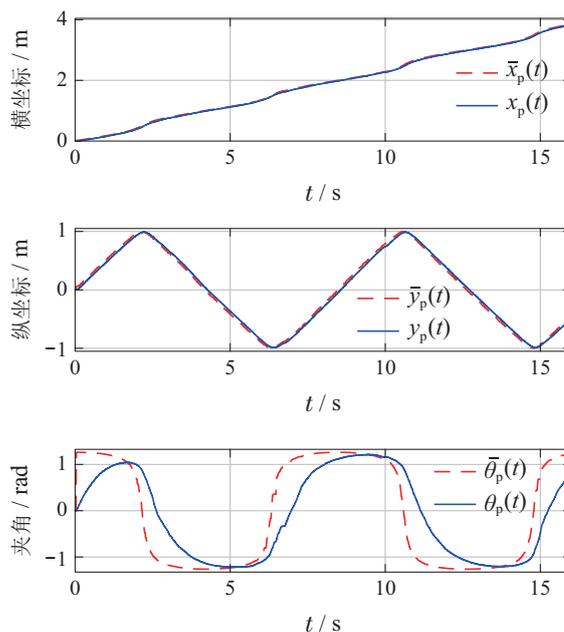
在 MAC 层的丢包率为 80% 时, 表 3 中 WNCS 的丢包率为 35.03%, 而表 2 中对应的 WNCS 的丢包率为 46.14%. 可见使用认知控制器调节 MAC 层的重传次数上限和物理控制器的命令序列长度, 可以使 WNCS 的丢包率降低 11% 左右. 表 3 中的移动机器人与参考轨迹的距离均值比表 2 中相应的值小 0.0255 m. 图 9(a) 和图 6(a) 中, 移动机器人都可以跟踪正弦参考轨迹. 对比图 9(b) 和图 6(b) 可以看出, 图 9(b) 的时延要明显小于图 6(b) 的时延. 从图 9(c) 可以看出, 移动机器人没有出现因为失去动力而停止前进的情况; 而图 6(c) 中, 移动机器人出现了多次失去动力的情形. 由图 9(d) 可知, a_r 大部分时间为 4, a_1 大部分时间为 16. 图 9 和图 6 的仿真结果表明: 对于高达 80% 的 MAC 层丢包率, MAC 层重传大量的数据包不但没有显著地减小 WNCS 的丢包率, 还增加了网络的时延. 这是因为重传大量的数据包会增加网络的负担. 认知控制器选择 a_r 为 4, a_1 为 16, 一方面减小了数据包的重传次数, 这减小了网络的负荷; 另一方面增加了物理控制器的命令序列长度, 这提高了 WNCS 对网络时延和丢包的承受能力. 由仿真结果可以看出, 认知控制器通过调节 MAC 层的重传次数上限和物理控制器的命令序列长度, 改善了网络的状态, 进而减小了第 5.2 节定义的信息间隙, 提高了 WNCS 控制移动机器人跟踪参考轨迹的性能.



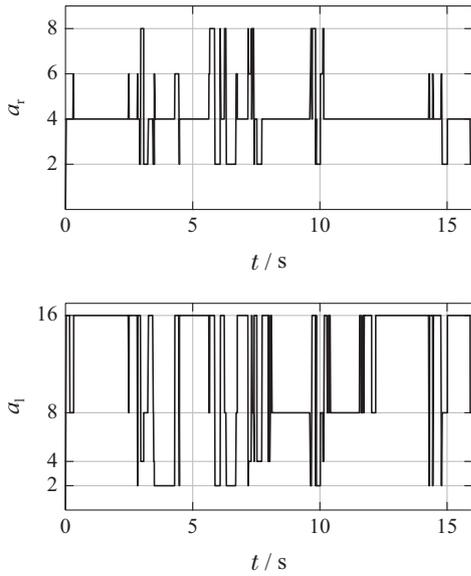
(a) 移动机器人的轨迹



(b) 传感器到感知器的时延



(c) 移动机器人的横坐标、纵坐标及其与 x 轴的夹角



(d) 认知控制器产生的认知作用

图9 移动机器人跟踪正弦参考轨迹的仿真结果 (使用认知控制器调节 a_r 和 a_1 , MAC层丢包率为80%)

Fig. 9 Simulation result that the mobile robot tracking sinusoidal reference trajectory (a_r and a_1 are adjusted by cognitive controller, and packet loss probability of the MAC layer is set as 80%)

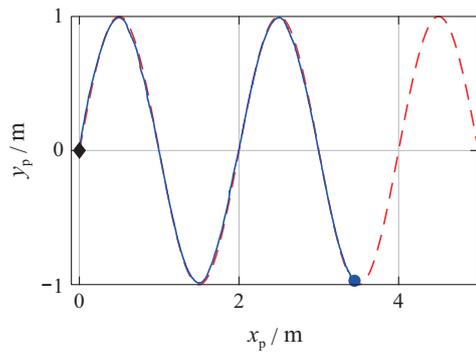
表3 MAC层的丢包率、WNCS的丢包率及移动机器人与参考轨迹的距离均值的对应关系(使用认知控制器)

Table 3 The relationship between packet loss probability of MAC layer, packet loss probability of WNCS and average distance between the mobile robot and reference trajectory (using cognitive controller)

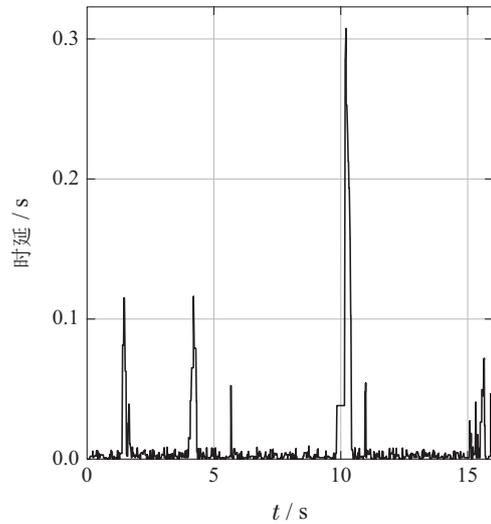
MAC层的丢包率/%	WNCS的丢包率/%	移动机器人与参考轨迹距离均值/m
70	17.93	0.0141
80	35.03	0.0137
90	60.19	0.0176

对于无线网络更加严酷的90%的MAC层丢包率,由第7.1节可知:使用固定配置的WNCS的网络发生了拥塞,WNCS无法控制移动机器人运动.表3中的WNCS的丢包率为60.19%,可见认知控制器的调节作用大幅降低了WNCS的丢包率.图10表明,在WNCS中加入认知控制器可以避免这一问题.从图10(a)和图10(c)可以看出,控制系统依然可以控制移动机器人跟踪参考轨迹.图10(b)中的时延没有出现线性增长.由图10(d)可以看出,在WNCS运行过程中, a_r 的值大部分时间是4, a_1 大部分时间为16.与MAC层丢包率

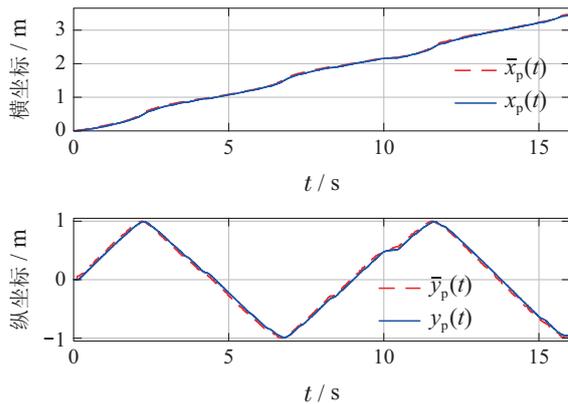
为80%的情形相同,这一方面减小了网络的负荷,另一方面提高了WNCS对时延和丢包的承受能力.图10的仿真结果表明,在MAC层丢包率为90%的情形下,认知控制器调节MAC层重传次数上限和物理控制器命令序列长度,改善了WNCS的无线网络环境,减小了信息间隙,提高了WNCS对无线网络环境的时延和丢包的承受能力.

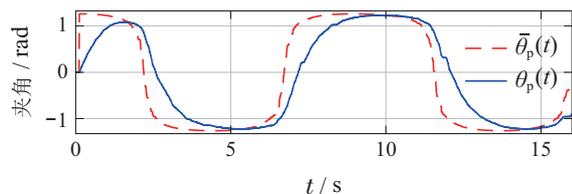
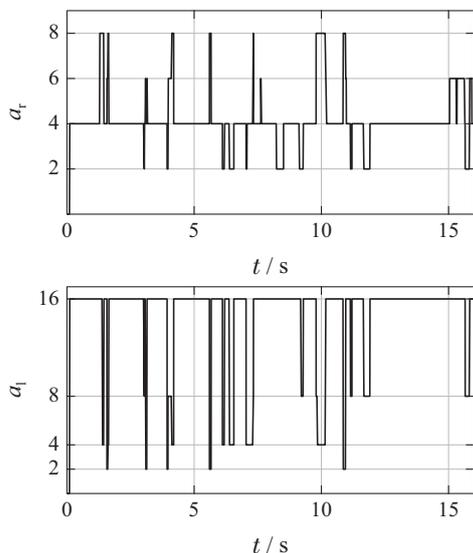


(a) 移动机器人的轨迹



(b) 传感器到传感器的时延



(c) 移动机器人的横坐标、纵坐标及其与 x 轴的夹角

(d) 认知控制器产生的认知作用

图 10 移动机器人跟踪正弦参考轨迹的仿真结果 (使用认知控制器调节 a_r 和 a_l , MAC层丢包率为90%)Fig. 10 Simulation result that the mobile robot tracking sinusoidal reference trajectory (a_r and a_l are adjusted by cognitive controller, and packet loss probability of the MAC layer is set as 90%)

综合第7.1节和第7.2节的仿真结果可以看出: 在WNCS中加入认知控制器调节MAC层的重传次数上限和物理控制器的命令序列长度, 增强了WNCS对网络时延和丢包的承受能力, 降低了WNCS的丢包率和时延, 进而改善了第5.2节定义的网络状态, 减小了信息间隙. 这为WNCS控制移动机器人运动提供了较好的网络环境, 有利于提高WNCS控制移动机器人跟踪参考轨迹的性能.

8 结论

本文受启发于Simon Haykin提出的认知控制思想, 用70%, 80%, 90%的MAC层丢包率及相应的网络时延来近似非结构化的无线网络环境. 按照Simon Haykin提出的认知控制系统框架, 设计了包含认知控制器的WNCS. 通过对比使用固定配置的WNCS的仿真结果和加入认知控制器的WNCS的仿真结果, 得出认知控制器通过调节MAC层的重传次数上限和物理控制器的命令序列长度, 增强了WNCS对无线网络环境的时延和丢包的承受能力, 降低了WNCS的丢包率和时

延, 改善了网络的状态, 减小了本文定义的信息间隙, 提高了WNCS控制移动机器人跟踪给定正弦参考轨迹的性能.

然而, 网络控制系统面临的不仅仅是网络环境的时延和丢包问题. 如何应用认知控制思想来同时处理更多的网络的不理想的特性, 如量化误差、时序错乱、时延及丢包, 是一个挑战性的问题.

参考文献:

- [1] MAHMOUD M S, HAMDAN M M. Fundamental issues in networked control systems. *IEEE/CAA Journal of Automatica Sinica*, 2018, 5(5): 902 – 922.
- [2] ZHANG D, SHI P, WANG Q G, et al. Analysis and synthesis of networked control systems: a survey of recent advances and challenges. *ISA Transactions*, 2017, 66: 376 – 392.
- [3] QIU J B, GAO H J, DING S X. Recent advances on fuzzy-model-based nonlinear networked control systems: a survey. *IEEE Transactions on Industrial Electronics*, 2016, 63(2): 1207 – 1217.
- [4] JIN Q B, WU S, ZHANG R D. Improved constrained model predictive tracking control for networked coke furnace systems over uncertainty and communication loss. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020, 50(5): 1877 – 1884.
- [5] YANG R N, LIU G P, SHI P, et al. Predictive output feedback control for networked control systems. *IEEE Transactions on Industrial Electronics*, 2014, 61(1): 512 – 520.
- [6] HWANG C L, CHANG L J, YU Y S. Network-based fuzzy decentralized sliding-mode control for car-like mobile robots. *IEEE Transactions on Industrial Electronics*, 2007, 54(1): 574 – 585.
- [7] WU C W, LIU J X, JING X J, et al. Adaptive fuzzy control for nonlinear networked control systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017, 47(8): 2420 – 2430.
- [8] CHEN Q X, LIU A D. D-stability and disturbance attenuation properties for networked control systems: switched system approach. *Journal of Systems Engineering and Electronics*, 2016, 27(5): 1108 – 1114.
- [9] PANG Z H, LIU G P, ZHOU D H. Design and performance analysis of incremental networked predictive control systems. *IEEE Transactions on Cybernetics*, 2016, 46(6): 1400 – 1410.
- [10] PANG Z H, LIU G P, ZHOU D H, et al. Data-based predictive control for networked nonlinear systems with network-induced delay and packet dropout. *IEEE Transactions on Industrial Electronics*, 2016, 63(2): 1249 – 1257.
- [11] HAYKIN S, FATEMI M, SETOODEH P, et al. Cognitive control. *Proceedings of the IEEE*, 2012, 100(12): 3156 – 3169.
- [12] FATEMI M, HAYKIN S. Cognitive control: theory and application. *IEEE Access*, 2014, 2: 698 – 710.
- [13] HAYKIN S, AMIRI A, FATEMI M. Cognitive control in cognitive dynamic systems: a new way of thinking inspired by the brain. *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*. Orlando, FL, USA: IEEE, 2014: 1 – 7.
- [14] FENG S, HAYKIN S. Cognitive risk control for transmit-waveform selection in vehicular radar systems. *IEEE Transactions on Vehicular Technology*, 2018, 67(10): 9542 – 9556.
- [15] WANG H W, YU F R, ZHU L, et al. A cognitive control approach to communication-based train control systems. *IEEE Transactions on Intelligent Transportation Systems*, 2015, 16(4): 1676 – 1689.
- [16] SUN W Z, YU F R, TANG T, et al. A cognitive control method for cost-efficient CBTC systems with smart grids. *IEEE Transactions on Intelligent Transportation Systems*, 2017, 18(3): 568 – 582.

- [17] LI Li, WANG Feiyue, ZHENG Nanning. Cognitive vehicle: a new research direction integrating cognitive science and control theory. *Control Theory & Applications*, 2011, 28(2): 137 – 142.
(李力, 王飞跃, 郑南宁. 认知车—结合认知科学和控制理论的新研究方向. 控制理论与应用, 2011, 28(2): 137 – 142.)
- [18] WEI Ruixuan, NI Tian, ZHAO Xiaolin, et al. Collision avoidance stability of the cognitive unmanned aerial vehicle & environment system. *Control Theory & Applications*, 2019, 36(9): 1453 – 1460.
(魏瑞轩, 倪天, 赵晓林, 等. 认知无人机—环境系统的防碰撞稳定性. 控制理论与应用, 2019, 36(9): 1453 – 1460.)
- [19] RU Changjian, WEI Ruixuan, GUO Qing, et al. Guidance control of cognitive game for unmanned aerial vehicle autonomous collision avoidance. *Control Theory & Applications*, 2014, 31(11): 1555 – 1560.
(茹常剑, 魏瑞轩, 郭庆, 等. 面向无人机自主防碰撞的认知博弈制导控制. 控制理论与应用, 2014, 31(11): 1555 – 1560.)
- [20] GARDNER R W, HOLZMAN P S, KLEIN G S, et al. *Cognitive Control: A Study of Individual Consistencies in Cognitive Behavior*. Charlottesville, VA, USA: International Universities Press, 1959: 185 – 186.
- [21] BARRETO J C L S, CONCEICAO A G S, DOREA C E T, et al. Design and implementation of model-predictive control with friction compensation on an omnidirectional mobile robot. *IEEE/ASME Transactions on Mechatronics*, 2014, 19(2): 467 – 476.
- [22] SUTTON R S, BARTO A G. *Reinforcement Learning: An Introduction, Second Edition*. Cambridge, MA, USA: MIT Press, 2018: 131 – 133.
- [23] MACIEJOWSKI J M. *Predictive Control with Constraints*. London, UK: Prentice Hall, 2002: 74 – 97.

作者简介:

徐君鹏 硕士研究生, 目前研究方向为认知控制在网络控制系统中的应用, E-mail: 17120290@bjtu.edu.cn;

尹逊和 教授, 博士生导师, 目前研究方向为网络化控制系统、信息—物理系统、认知控制, E-mail: xhyin@bjtu.edu.cn.