

# 基于滑动窗口最长公共子序列 Wi-Fi 指纹定位算法

张明洋<sup>1</sup>, 陈 剑<sup>1,2</sup>, 闻英友<sup>1,2</sup>, 赵 宏<sup>1,2</sup>

(1. 东北大学 信息科学与工程学院, 辽宁 沈阳 110819; 2. 东北大学 医学影像计算教育部重点实验室, 辽宁 沈阳 110819)

**摘 要:** 针对基于 Wi-Fi 瞬时指纹定位算法中由于 RSS 信号的时变特性引起的 Wi-Fi 定位精度差问题, 提出了一种基于滑动窗口最长公共子序列指纹定位算法. 该算法将时间序列的 RSS 信号指纹转化为基于滑动窗口的数据模型, 增加了指纹特征信息, 提高比对准确性. 通过计算请求定位数据与样本的最长公共子序列来获得样本点的相似性, 解决由于窗口伸缩或滑动窗口中个别采样点无信号引起的比对不准确问题, 从而提高了定位的精确性和鲁棒性. 实验结果表明, 所提定位算法的结果明显优于瞬时指纹定位算法.

**关 键 词:** 室内定位; 指纹; 滑动窗口; 时间序列; 最长公共子序列

中图分类号: TP 393.0

文献标志码: A

文章编号: 1005-3026(2014)10-1390-05

## Wi-Fi Fingerprint Localization Algorithm Based on Sliding Window Combined with Longest Common Subsequence

ZHANG Ming-yang<sup>1</sup>, CHEN Jian<sup>1,2</sup>, WEN Ying-you<sup>1,2</sup>, ZHAO Hong<sup>1,2</sup>

(1. School of Information Science & Engineering, Northeastern University, Shenyang 110819, China; 2. Key Laboratory of Medical Image Computing of Ministry of Education, Northeastern University, Shenyang 110819, China. Corresponding author: ZHANG Ming-yang, E-mail: zmyzyc1988@126.com)

**Abstract:** To reduce the negative effect in the Wi-Fi fingerprint localization algorithm caused by the fluctuation of the received signal strength (RSS), a Wi-Fi fingerprint localization algorithm based on sliding window combined with the longest common subsequence was proposed. First, the time sequence RSS fingerprints were converted to the sliding window data model to increase the fingerprint characteristic information and improve the matching accuracy. And then, the requesting location data and the longest common subsequence were calculated to get the similarity of sampling points, which could solve the problem caused by the window scaling or the individual sampling point without signal in the sliding window, thereby the localization accuracy and robustness were improved. The results showed that the proposed localization algorithm was superior to the instantaneous fingerprints localization algorithm.

**Key words:** indoor localization; fingerprint; sliding window; time series; longest common subsequence

近年来,随着位置服务和位置感知计算等移动互联网增值业务的发展和普及,基于 Wi-Fi 网络的定位技术以其开放接入、低成本的特点受到重视<sup>[1]</sup>. 该定位技术利用部署在 Wi-Fi 网络环境中的无线接入点(access point, AP)或 Wi-Fi 终端的信号强度,就可以对移动终端进行定位,而其中基于 Wi-Fi 的指纹定位技术以其设备部署便利的优势,尤其受到学术界的关注<sup>[2]</sup>.

在室内环境中,由于建筑物布局、多径、墙面反射吸收、温湿度变化、人员出入等复杂因素对 Wi-Fi 信号传播衰减造成的影响,导致信号强度衰减量与收发设备之间距离并不满足路径损耗渐变模型<sup>[3]</sup>,因此使用基于测距的三边或三角定位算法难以获得较理想的定位效果. 而基于 Wi-Fi 指纹比对的定位技术则利用复杂环境引起的 Wi-Fi 信号的特异性来构建指纹信息. 因为 Wi-Fi 信

收稿日期: 2014-04-23

基金项目: 国家自然科学基金资助项目(60903159, 61173153); 沈阳市科技计划项目(1091176-1-00); 中央高校基本科研业务专项资金资助项目(N110318001, N100218001).

作者简介: 张明洋(1989-),男,辽宁辽阳人,东北大学博士研究生; 赵 宏(1954-),男,河北河间人,东北大学教授.

号的特异性对地形和传播时的障碍物具有依赖性,呈现出针对环境的特殊特征,相对于路径损耗渐变模型,由此方式构造出的特征指纹库进行指纹定位可获得更好的定位结果。

室内 Wi-Fi 信号除了不满足路径损耗渐变模型之外,Wi-Fi 信号本身还具有较大的时变性和随机性<sup>[4]</sup>,移动终端在同一地点观测到的实时信号值也并不总能够与该点的指纹样本所匹配,由此引起个别测试点的跳动从而影响总体定位效果.针对该问题现有的解决方案大多采取定位后修正的方式,如基于高斯滤波方法<sup>[5]</sup>、卡尔曼滤波方法<sup>[6]</sup>等通过过滤的方式实现对定位结果的修正,然而该种方式的缺陷是只适用于连续定位过程.更好的方式是在定位过程中避免信号时变性引起的定位跳动,然而基于该种思想的解决方案相对较少。

基于在定位过程中避免定位跳动的思想,针对现有基于瞬时采样指纹比对定位方法的不足,本文提出了一种基于滑动窗口最长公共子序列的定位算法,它通过将时间序列采样指纹作为滑动窗口数据,充分利用时间序列指纹特征,并将滑动窗口数据进行最长公共子序列比对,从而消除时间伸缩对指纹比对的影响,进而在定位过程中抑制了结果跳变,提高了总体定位效果。

# 1 指纹定位算法

通常进行的指纹定位算法<sup>[7]</sup>为瞬时指纹数据定位,一般可以划分为指纹样本的采集处理和指纹数据的比对定位两个步骤.在指纹采集处理过程中,通过 Wi-Fi 信号采集终端定位环境中的指纹信号数据,通常为 AP 信号强度,通过对采集到的信号强度数据  $\mathbf{R} = \{r_1, r_2, \dots, r_m\}$ ,采样点位置  $\mathbf{X} = \{x_1, x_2, \dots, x_k\}$  进行映射形成一条指纹样本记录  $s_i = \{\mathbf{R}_i, \mathbf{X}_i\}$ ,采集过程通过对定位环境中预先设置的采样点进行采样,并将采样记录收集整理,进行包括剔除错误点、信号异常点、采样冗余点等预处理后,形成用于指纹比对的样本库  $\mathbf{S} = \{s_1, s_2, \dots, s_n\}$ ,从而实现了将每一组指纹信息映射到定位空间中一个特定的位置.对于指纹比对的定位过程,定位请求终端以同样形式采集定位环境中的瞬时指纹信号生成请求数据  $\mathbf{R}_q = \{r_{q1}, r_{q2}, \dots, r_{qm}\}$ ,采用相应的比对算法与样本指纹数据进行比对,此处通常采用欧式距离进行比对,如公式(1)所示,而后从  $d_i$  中选取若干个最佳结果对应的实际坐标位置,进而得出用户的实际位置。

$$d_i = \sqrt{(\mathbf{R}_i - \mathbf{R}_q)^T (\mathbf{R}_i - \mathbf{R}_q)}. \tag{1}$$

# 2 基于滑动窗口指纹数据模型

基于滑动窗口的采样指纹模型的采集方式与瞬时指纹采样方式不同,通常的瞬时指纹采样是在规划的定位环境中选取若干采样点,尽量保证采样点在定位环境中均匀分布、没有遗漏点,而后静止于每个采样点逐个对其进行一段时间的采样.而基于滑动窗口的指纹采样方式是在规划的定位环境中选取若干线路,而后沿着每条路径以大致匀速行走若干次,并沿途相隔均等时间记下若干点的信号强度信息  $\mathbf{R}_t = \{r_1, r_2, \dots, r_m\}$  和坐标位置信息  $\mathbf{X}_t = \{x_1, x_2, \dots, x_k\}$ ,而后针对时间  $t$  设置滑动窗口的大小  $k$ ,建立基于滑动窗口大小的信号指纹组  $\mathbf{A}_i = \{\mathbf{R}_{t-k+1,i}, \mathbf{R}_{t-k+2,i}, \dots, \mathbf{R}_{t,i}\}$ ,并依据滑动窗口的大小建立指纹数据样本记录  $s_i = \{\mathbf{A}_i, \mathbf{X}_{t,i}\}$ ,如图 1 所示,图中滑动窗口设置为 3 的样本记录情况.而后针对采样线路上各点逐个按照滑动窗口模型建立用于指纹比对的数据样本库  $\mathbf{S} = \{s_1, s_2, \dots, s_n\}$ ,该样本库中每条指纹记录包括  $m$  个信号发射终端在时间窗口为  $k$  时间内的全部指纹数据.其指纹信息量相对于瞬时指纹采样的每条记录是其  $k$  倍,且其反映了信号发射终端信号强度随位置变化的规律,针对请求定位指纹数据亦采用适用于该种模型的格式数据  $\mathbf{A}_q = \{\mathbf{R}_{t-k+1,q}, \mathbf{R}_{t-k+2,q}, \dots, \mathbf{R}_{t,q}\}$ 。

由于本文所设计的数据采样模型主要是应用于典型的公共场所定位环境,例如:商场、超市、楼宇等,针对存在的特殊应用环境可以针对具体的情况进行处理,如路径较多的开放区域可考虑在不影响定位精度的情况下选取其中若干条路径作为数据采集路径。

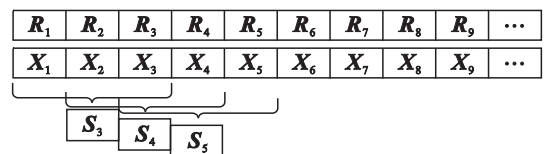


图 1 滑动窗口数据模型  
Fig. 1 Sliding window data model

# 3 最长公共子序列定位算法

本文的请求定位数据与指纹库样本数据均采用基于滑动窗口的指纹数据模型,若样本数据和定位请求终端数据均是以往常规速度行进采集,使

用指纹比对算法进行指纹比对,基于多窗口的指纹比对定位效果相比于瞬时指纹比对定位效果要好很多.然而,当定位请求终端的行进速度改变时,由于样本数据还是常规速度数据,因此请求定位窗口中的数据不再和样本窗口中的数据相对应,相当于对滑动窗口中数据进行了拉伸或是收缩,从而导致窗口中比对的指纹不再对应,进而导致定位结果不再准确,在实际应用中无法对所有变速数据进行样本的采集,因此,本文为解决滑动窗口伸缩的问题引入了最长公共子序列的比对方式.

最长公共子序列(longest common subsequence, LCSS)比对算法<sup>[8]</sup>,主要用于字符串数据的相似性计算,目前已应用于分子 DNA 匹配<sup>[9]</sup>、流数据匹配<sup>[10]</sup>等,针对传统一维的标称类型数据,具体匹配过程如公式(2)所示:

$$\begin{cases} \text{LCSS}(A_i, B_j) = 0, \\ \text{if } i = 0 \text{ or } j = 0; \\ 1 + \text{LCSS}(A_{i-1}, B_{j-1}), \text{ if } F_{\text{equal}}(a_i, b_j) = 1; \\ \text{MAX}(\text{LCSS}(A_{i-1}, B_j), \text{LCSS}(A_i, B_{j-1})), \\ \text{if } F_{\text{equal}}(a_i, b_j) \neq 1. \end{cases} \quad (2)$$

式中:LCSS( $A, B$ )为计算  $A$  与  $B$  的最长公共子序列的函数; $A_i = \{a_1, a_2, \cdots, a_i\}$ ,  $B_j = \{b_1, b_2, \cdots, b_j\}$  为数据集; $F_{\text{equal}}(a, b)$  为相等公式.

然而,本文所使用的基于滑动窗口模型的数据均为数值型多维数据,无法应用传统最长公共子序列比对算法中的相等公式.鉴于此,本文对相等公式进行了修改,扩展了最长公共子序列算法,以适用于滑动窗口指纹数据,修改的相等公式( $F_{\text{equal}}(\mathbf{R}_{t,i}, \mathbf{R}_{t,q})$ )计算过程如下:

- 1) 计算  $d_i = \sqrt{(\mathbf{R}_{t,i} - \mathbf{R}_{t,q})^T (\mathbf{R}_{t,i} - \mathbf{R}_{t,q})}$ ;

2) 若  $d_i \leq C$ ,则认为  $\mathbf{R}_{t,i}$  与  $\mathbf{R}_{t,q}$  相等;

3) 否则,认为  $\mathbf{R}_{t,i}$  与  $\mathbf{R}_{t,q}$  不相等.

其中: $\mathbf{R}_{t,i}$  为指纹样本记录中每个信号强度信息值集; $\mathbf{R}_{t,q}$  为请求定位的指纹数据信号强度信息值集; $C$  为设置的一个阈值,作为相等的判断.

另外,本文对最长公共子序列传统算法的返回值以及 MAX 函数等加以修改以适用于滑动窗口指纹数据比对,传统算法返回结果通常为匹配的数据个数  $l_i$ 、匹配的  $A_i$  中元素集  $\alpha_i$ 、匹配的  $B_j$  中元素集  $\beta_j$ ,本文为其添加由  $F_{\text{equal}}(\mathbf{R}_{t,i}, \mathbf{R}_{t,q})$  计算所得的  $d_i$ ,且同样经过每次递归调用对  $d_i$  累加,而对 MAX 值判断的修改是取  $d_i/l_i$  较小值对应的数据作为 MAX 选取值.通过最长公共子序列比对后选取样本指纹库中 LCSS 值最大的前若干个样本记录.若  $l_i \neq 0$ ,则计算其  $d_i/l_i$  值,选择

最小的若干值而后取均值作为最终定位结果;若  $l_i = 0$ ,则计算滑动窗口中样本记录与定位请求指纹的欧氏距离,选取其中最小的若干值取平均作为定位结果.

需要注意的是,若数据样本库过大,LCSS 比对过程会产生较高的计算复杂度,对此,可以采取将采集数据在 LCSS 比对前利用一些分类算法<sup>[11]</sup>进行分类后再比对,以降低比对计算量.

## 4 实验测试

本实验测试环境如图 2 所示,实验环境面积大小为  $25\text{ m} \times 15\text{ m}$ ,其中上下左右的 4 个“米”处为无线 AP 所在位置,其中的实心点为实验采样点.本实验对环境进行了多次采样,其中包括正常速度、快 1~8 倍速度、慢 1~8 倍速度数据采样,并将各类采样数据分别分为两部分:一部分作为采样样本,另一部分作为测试数据.本文实验部分主要包括三部分:第一部分是对窗口及选取最终参考点平均个数的测试;第二部分是对比采用滑动窗口最长公共子序列与仅采用滑动窗口欧式距离比对在快、慢 1~8 倍速度时各自的定位结果;第三部分是测试  $C$  值的选取对定位结果的影响.

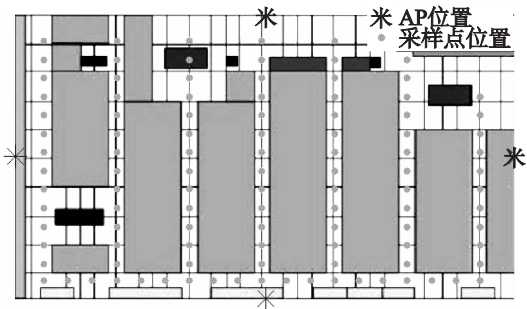


图 2 测试环境采点图  
Fig. 2 Points diagram of test environment

4.1 窗口及选取最终参考点平均个数的测试  
窗口及选取最终参考点平均个数的测试在于验证滑动窗口指纹定位效果要优于瞬时指纹比对的定位效果,而对于相同窗口大小,算法选取用于计算平均值的结果个数  $K$  的大小影响定位效果.如图 3 所示,本实验共测试 1~8 个滑动窗口,每个窗口测试 1~5 个  $K$  值选取,图 3 中,每相邻的 5 个点为一个窗口的 1~5 个  $K$  值的测试平均误差效果,由图 3 可以看出测试窗口在 5(图 3 中横坐标 21 点~25 点)之前定位平均误差急速下降,而在 5 之后测试平均误差下降不再明显.而对于  $K$  值瞬时指纹比对(窗口大小为 1 时)的定位随  $K$

值变化呈现无规律变化,但对于窗口大小不为 1 的定位结果随  $K$  值的增加,平均误差变大.

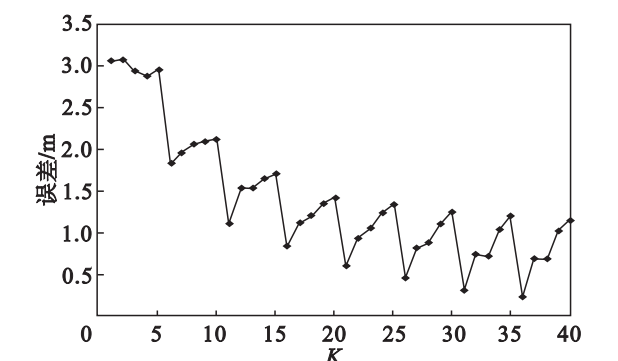


图 3 不同窗口及平均值下的测试误差  
Fig. 3 Test error of different window and mean value

#### 4.2 变速滑动窗口欧氏距离与 LCSS 算法测试

本实验通过将滑动窗口数据分别代入欧氏距离指纹比对算法和最长公共子序列算法,计算定位平均误差,如图 4 所示. 其中 0 点左侧为从慢 8 倍到慢 1 倍的测试数据代入正常速度样本所获得的定位平均误差,0 点右侧为从快 1 倍到快 8 倍的测试数据代入正常速度样本所获得的定位平均误差. 从图中可以看到最长公共子序列比对的定位算法因为能够收集窗口中的特征相似数据而不受比对数据双方中一方的速度变化的影响,因而能够获得较好的定位效果,其针对变速数据的测试平均误差均稳定在 3 m 以内. 而采用欧氏距离比对因其只能对滑动窗口内数据按先后对应顺序进行比对,因而无法克服由于比对数据中单方的速度变化带来的比对特征错位的问题,因而其定位平均误差较大,大多已经超过了 3 m,甚至有的已经超过了 4 m.

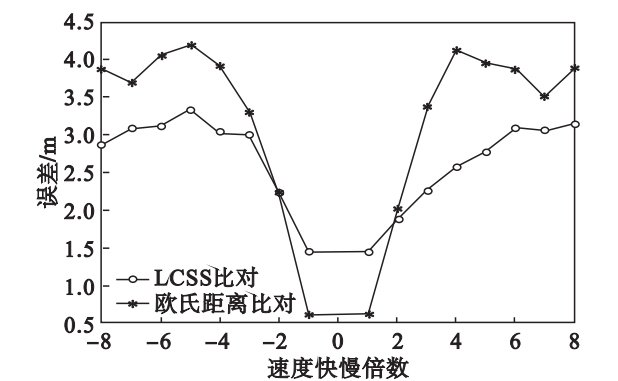


图 4 不同速率下的测试误差  
Fig. 4 Test error of different speed

#### 4.3 选取 $C$ 值的定位结果的测试

本文的定位算法中,阈值  $C$  的选取对定位效果会有影响,当阈值  $C$  过小时,相等公式的判断

条件成立的可能性会减小,从而导致最长公共子序列比对过程失效,根据算法执行过程,进而直接进行欧式距离比对,从而影响定位精度;而当阈值  $C$  过大时,相等公式的判断条件成立的可能性会变大,从而导致滑动窗口内数据的比对变成按顺序依次对应比对,进而算法退化为欧氏距离的比对,影响定位精度. 对此,本文针对当前实验环境下,对相等公式的阈值  $C$  的选取进行了实验测试. 如图 5 所示,针对 1~8 倍不同的速度选取 3 到 24 且值之间间距为 3,通过 8 条曲线数据的分析,可以直观地看到当阈值  $C$  为 9~12 之间的值时,各条曲线都趋向于达到一组较小的平均误差值,即是该算法针对该实验环境的最小平均误差,即是在最大 8 倍变速的情况下,当阈值  $C$  取 9~12 之间的值时,该定位算法均可获得最小定位误差.

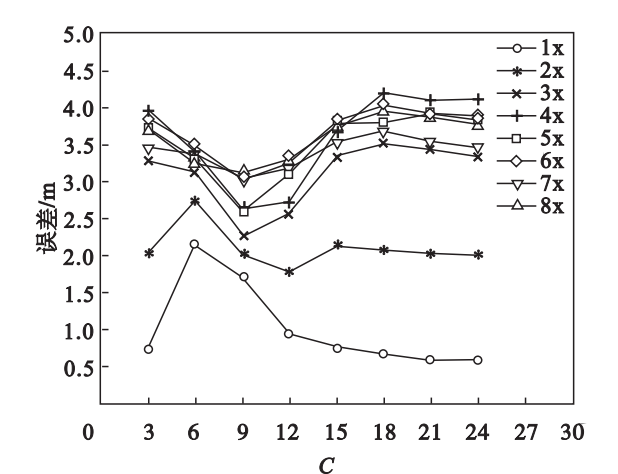


图 5 不同  $C$  值下的测试误差  
Fig. 5 Test error of different  $C$ -value

#### 4.4 定位测试分析

本文提出的定位算法是对原始指纹定位算法中指纹比对方式做出优化,修改比对的指纹数据模型为滑动窗口指纹数据模型,引入了最长公共子序列比对的方式,从而提高了定位精度. 而针对一些诸如环境因素造成的恶劣情况下,本算法的定位精度可能会下降,而退化的下限为原始指纹定位算法. 因此其中的一些参数设置是本文所提出算法能够达到较好定位效果的关键. 由于该算法主要是基于多窗口指纹进行定位,其中若将滑动窗口值设置为 1,且算法的相等公式  $F_{equal}$  中的阈值  $C$  设置为一个较大值,则该算法退化为原始指纹比对算法. 通过对 4.1~4.3 节的实验分析,可知既能够满足高精度定位需求又不至于影响定位速度的适用于日常生活环境的滑动窗口设置为 4~6,阈值  $C$  选取 9~12 较为合适.



5 结 论

基于指纹的定位算法中由于指纹信号特征的时变特性引起的定位精度问题是指纹定位算法普遍存在的问题,本文通过建立基于时间序列的滑动窗口模型指纹数据,控制时变特性引起的定位精度问题,并通过最长公共子序列算法解决由于滑动窗口造成的时间伸缩问题,由此提高了定位算法的整体定位精度.实验验证了本文算法的总体定位精度较瞬时指纹定位算法有一定提高.

参考文献:

[ 1 ] Wu K S, Xiao J, Yi Y W, *et al.* CSI-based indoor localization [ J ]. *IEEE Transactions on Parallel and Distributed Systems*, 2013, 24 ( 7 ) : 1300 – 1309.

[ 2 ] Niu J W, Lu B H, Cheng L, *et al.* ZiLoc: energy efficient WiFi fingerprint-based localization with low-power radio [ C ] // *Wireless Communications and Networking Conference (WCNC)*. Shanghai, 2013 : 4558 – 4563.

[ 3 ] Bahl P, Padmanabhan V N. RADAR: an in-building RF-based user location and tracking system [ C ] // *International Conference on Computer Communications*. Tel Aviv, 2000, 775 – 784.

[ 4 ] 赵方,罗海勇,马严,等.基于公共信标集的高精度射频指

纹定位算法 [ J ]. *计算机研究与发展*, 2012, 49 ( 2 ) : 243 – 252.

( Zhao Fang, Luo Hai-yong, Ma Yan, *et al.* An accurate fingerprinting localization algorithm based on common beacons [ J ]. *Journal of Computer Research and Development*, 2012, 49 ( 2 ) : 243 – 252.

[ 5 ] Duvallet F, Tew A D. WiFi position estimation in industrial environments using Gaussian processes [ C ] // *Intelligent Robots and Systems*. Nice, 2008 : 2216 – 2221.

[ 6 ] Yim J, Jeong S, Gwon K, *et al.* Improvement of Kalman filters for WLAN based indoor tracking [ J ]. *Expert Systems with Applications*, 2009, 37 ( 6 ) : 426 – 433.

[ 7 ] Gezici S. A survey on wireless position estimation [ J ]. *Wireless Personal Communications*, 2008, 44 ( 3 ) : 263 – 282.

[ 8 ] Bergroth L, Hakonen H, Raita T. A survey of longest common subsequence algorithms [ C ] // *String Processing and Information Retrieval*. Curuna, 2000 : 39 – 48.

[ 9 ] Parvinnia E, Taheri M, Ziarati K. An improved longest common subsequence algorithm for reducing memory complexity in global alignment of DNA sequences [ C ] // *Biomedical Engineering and Informatics*. Sanya, 2008 : 57 – 61.

[ 10 ] Liben-Nowell D, Vee E, Zhu A. Finding longest increasing and common subsequences in streaming data [ J ]. *Journal of Combinatorial Optimization*, 2006, 11 ( 2 ) : 155 – 175.

[ 11 ] Harrington P. *Machine learning in action* [ M ]. New York: Manning Publications, 2012 : 1 – 148.

(上接第 1389 页)

参考文献:

[ 1 ] Wang T S, Wang X Y, Wang M J. A simple criterion for impulsive chaotic synchronization [ J ]. *Communications in Nonlinear Science and Numerical Simulation*, 2011, 16 ( 3 ) : 1464 – 1468.

[ 2 ] Nian F Z, Wang X Y, Li M, *et al.* A universal projective synchronization of general autonomous chaotic system [ J ]. *Pramana-Journal of Physics*, 2012, 79 ( 6 ) : 1375 – 1383.

[ 3 ] Lin D, Wang X Y, Nian F Z, *et al.* Dynamic fuzzy neural networks modeling and adaptive backstepping tracking control of uncertain chaotic systems [ J ]. *Neurocomputing*, 2010, 73 ( 16/17/18 ) : 2873 – 2881.

[ 4 ] Wang X Y, Luan D P. A novel image encryption algorithm using chaos and reversible cellular automata [ J ]. *Communications in Nonlinear Science and Numerical Simulation*, 2013, 18 ( 11 ) : 3075 – 3085.

[ 5 ] Bamieh B, Pearson J, Francis B, *et al.* A lifting technique for linear periodic systems [ J ]. *Systems and Control Letters*, 1991, 17 ( 2 ) : 79 – 88.

[ 6 ] Hu L S, Lam J, Cao Y Y, *et al.* A LMI approach to robust  $H_2$  sampled-data control for linear uncertain systems [ J ]. *IEEE Transactions on Systems, Man, and Cybernetics: B*, 2003, 33 ( 1 ) : 149 – 155.

[ 7 ] Fridman E, Seuret A, Richard J. Robust sampled-data

stabilization of linear systems: an input delay approach [ J ]. *Automatic*, 2004, 40 ( 8 ) : 1441 – 1446.

[ 8 ] Lu J G, Hill D J. Global asymptotical synchronization of chaotic Lur'e systems using sampled data: a linear matrix inequality approach [ J ]. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2008, 55 ( 6 ) : 586 – 590.

[ 9 ] Zhang C K, He Y, Wu M. Improved global asymptotical synchronization of chaotic Lur'e systems with sampled-data control [ J ]. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2009, 56 ( 4 ) : 320 – 324.

[ 10 ] 吴忠强,邝钰,汤辉.连续及采样异构混沌同步控制 [ J ]. *控制工程*, 2010, 17 ( 2 ) : 142 – 145.

( Wu Zhong-qiang, Kuang Yu, Tang Hui. Continuous-time and sampled-data synchronization control of chaotic with diverse structures [ J ]. *Control Engineering of China*, 2010, 17 ( 2 ) : 142 – 145. )

[ 11 ] Kuo H H, Hou Y Y, Yan J J, *et al.* Reliable synchronization of nonlinear chaotic systems [ J ]. *Mathematics and Computers Simulation*, 2009, 79 ( 5 ) : 1627 – 1635.

[ 12 ] Zhang Y, Sun J. Impulsive robust fault-tolerant feedback control for chaotic Lur'e systems [ J ]. *Chaos, Solitons and Fractals*, 2009, 39 ( 3 ) : 1440 – 1446.

[ 13 ] Ma D Z, Zhang H G, Wang Z S, *et al.* Fault tolerant synchronization of chaotic systems based on T-S fuzzy model with fuzzy sampled-data controller [ J ]. *China Physics: B*, 2010, 19 ( 5 ) : 1 – 11.