

基于分组码的改进型防碰撞算法研究*

张学军, 王绪海, 蔡文琦

(南京邮电大学 电子科学与工程学院, 南京 210003)

摘要: 在二叉树防碰撞算法的基础上,应用分组码对标签识别码进行编码的方法,提出了一种基于分组码的改进型防碰撞算法(IABC算法)。通过数学分析,准确地描述了IABC算法识别标签所需的时隙数。仿真结果表明,IABC算法具有较快的识别速度和较高的识别效率。

关键词: 射频识别; 防碰撞算法; 分组码; 改进型树算法

中图分类号: TN92; TP301.6

文献标志码: A

文章编号: 1001-3695(2012)11-4265-04

doi:10.3969/j.issn.1001-3695.2012.11.067

Improved anti-collision algorithm based on block code for RFID system

ZHANG Xue-jun, WANG Xu-hai, CAI Wen-qi

(School of Electronic Science & Engineering, Nanjing University of Posts & Telecommunications, Nanjing 210003, China)

Abstract: Based on the multi-tree anti-collision algorithm, this paper proposed an improved anti-collision algorithm based on block codes(IABC). It used the block code to encode the tag ID. By mathematical analysis, it accurately predicted the total number of time slots required for the tag identification in IABC algorithm. Simulation results show that the IABC algorithm has faster identification speed and higher efficiency.

Key words: RFID(radio frequency identification); anti-collision algorithm; block code; improved tree algorithm

射频识别(radio frequency identification, RFID)是一种利用射频信号通过空间电磁耦合实现无接触式自动识别的技术。RFID技术无须直接接触、光学可视、人工干预即可完成信息输入和处理,具有操作方便快捷、存储数据量大、保密性好、反应时间短、对环境适应性强、识别距离远、穿透能力强、多物体识别、抗污染等优点,现在已广泛应用于工业自动化、商业自动化、交通运输管理、产品证件防伪、防盗等众多领域。RFID技术、传感器技术、纳米技术、智能嵌入技术是实现物联网的四大核心技术。随着物联网技术和应用的不断深入,RFID技术已成为当前研究的热点^[1]。

RFID系统由电子标签、阅读器、数据处理子系统等三个部分^[2]组成。当阅读器的作用范围内存在多个标签,并有两个或以上的标签同时响应阅读器的请求时将会产生碰撞,这种碰撞称为标签碰撞。良好的防碰撞算法可以有效解决标签的碰撞问题,实现标签的快速、高效的识别。本文在二叉树防碰撞算法的基础上,提出了一种基于分组码的改进型防碰撞算法——IABC(improved anti-collision algorithm based on block code)算法。

1 RFID防碰撞算法

RFID防碰撞算法一般有基于空分多址(space division multiple access, SDMA)、频分多址(frequency division multiple access, FDMA)、码分多址(code division multiple access, CDMA)、时分多址(time division multiple access, TDMA)等四种方式^[3]。其中,TDMA方式由于应用简单,容易实现大量标签的

读写,所以被多数防碰撞算法采用。现有防碰撞算法主要分为基于ALOHA算法和基于树算法两种^[4]。

1.1 基于ALOHA的防碰撞算法

纯ALOHA的防碰撞算法是最基本的标签防碰撞算法,是基于TDMA的概率性防碰撞算法^[5,6]。该算法是一种信号随机接入的方法,采用标签先发言的方式,即当标签进入阅读器的作用区域就自动向阅读器发送自身的信息,对同一个标签来说,它发送数据帧的时间是随机的。在标签信息发送过程中如果有其他标签也在发送数据,那么发生信号重叠会导致部分碰撞或者完全碰撞。阅读器检测信号并判断是否碰撞,若发生碰撞,阅读器发送命令让标签停止发送数据,所有标签会随机延迟一段时间再发送数据,由于延迟的随机数不同,再次发生碰撞的概率降低;若没有碰撞,阅读器发送应答信号给标签,标签转入休眠状态。对于无接收功能的标签,标签在收不到阅读器发送的碰撞信息或者应答信息情况下,在检测期间一直重复地发送自己的信息,直到识别过程结束^[7]。

除了纯ALOHA算法,还有时隙ALOHA算法、帧时隙ALOHA算法、动态帧时隙ALOHA算法等几种增强型的ALOHA算法。在上述算法中,标签随机选择算法分配的时隙,可能会出现标签长时间不能被识别的情况,即“标签饿死”现象。

1.2 二进制查询树防碰撞算法

当标签数量较多时,基于时隙ALOHA的防碰撞算法的识别率不能够达到100%,而基于树的防碰撞算法可以实现完全

收稿日期: 2012-03-28; **修回日期:** 2012-05-05 **基金项目:** 国家自然科学基金资助项目(60806027, 61001077); 江苏省高校自然科学基金资助项目(08KJB510015); 南京邮电大学引进人才资助项目(NY208048)

作者简介: 张学军(1969-),男,江苏南通人,副教授,硕士,博士,主要研究方向为无线射频识别技术、通信网络的性能分析、流量控制、QoS理论与技术等(xjzhang@njupt.edu.cn); 王绪海(1987-),男,硕士研究生,主要研究方向为无线射频识别技术; 蔡文琦(1987-),女,硕士研究生,主要研究方向为无线射频识别技术。

识别^[8-12],目前应用较多的是二进制查询树算法。

阅读器发出询问信号,阅读器作用范围内的标签应答,阅读器主动地选择 0 然后是 1。当树在一个节点处分裂时,其步骤为^[4]:

a) 阅读器依次发送查询命令集合 $Q = \{q_i, \dots, q_j\}$ 中的查询命令 $q_i = q_1, \dots, q_x$, 并更新集合 $Q = Q - \{q_i\}$ 。

b) 与该查询命令的二进制序列相匹配的标签都响应。若只有一个标签响应,则识别该标签;若有多于一个的标签响应,说明发生了碰撞,则分别在查询序列后面加 0 和 1 形成两个新的查询命令加入到集合 $Q = Q \cup \{q_1, \dots, q_x, 0, q_1, \dots, q_x, 1\}$, 这样也就把发生碰撞的标签集分成了两个子集;若没有标签响应,则不进行任何操作。

c) 重复步骤 a)b), 直到 Q 为空。

如果碰撞发生在标签的最后一位,基于标签 ID 号唯一性则阅读器认定必有两个标签,即可直接识别而无须再次发送。假定有 N 个标签,则采用树算法成功识别一个标签平均所需的询问次数为

$$I_{BT} = \log(N) / \log(B) + 1$$

在 B 叉树算法中, B 的取值直接影响标签的识别效率。除应用较多的二叉树以外,还有四叉树^[13]、八叉树等。通过分析八叉树每一叉的码结构,将其按照一定方法进行分组,达到将空闲时隙数降为零,碰撞时隙数不变的目的。本文所提出的 IABC 算法就是利用分组码对八叉树每一叉的识别码进行编码的改进型树算法。

2 基于分组码改进型防碰撞算法

在八叉树($B=8$)中,分组长度 $\lambda=3$,其碰撞时隙数比二叉树和四叉树算法少,但其空闲时隙数比二叉树和四叉树算法多。为了减少空闲时隙数,且不增加碰撞时隙数,本文提出了分组码的概念,对八叉树算法进行改进,去除了空闲时隙,提高了识别效率。

2.1 算法描述

IABC 算法主要分为分组操作和识别操作两部分。

1) 分组操作

在分组操作过程中引入分组码的概念。分组规则为:以 0 和 1 的分布情况进行分类,在八叉树中有八个分支,即 000、001、010、011、100、101、110、111。可得到四种类型的分组码(也可以看做是根据标签识别码中 0 和 1 的数量进行分组),如表 1 所示。采用表 1 的编码方式对八叉树的每一种分支进行编码。

2) 识别操作

标签发送分组码,阅读器收到数据后,根据碰撞的位置就可以确定存在的分组码。例如存在标签识别码 111、101、110、001 时,其对应的分组码为 01、001、001、0001。阅读器收到的数据为 $0 \times \times 1$,根据碰撞位可知,存在 01、001、0001 这三种分组码。

当阅读器对标签进行识别操作时,可分为两种情况:

a) 对分组码 1 和 0001 的识别。由表 1 可知,识别码 000 与分组码 1、识别码 111 与分组码 0001 是一一对应的,因此可利用分组码 1 和 111 来分别代替 000 和 111,无须进行二次发送即可识别标签,减少了标签与阅读器的数据通信量。

b) 对分组码 01 和 001 的识别。由表 1 可知,识别码 001、010、100 对应分组码 01,识别码 110、101、011 对应分组码 001,属于多对一的关系,因此分组码 01 和 001 的识别需进行二次发送。当阅读器收到标签二次发送的信息时,通过对属于分组码 01 和 001 的标签识别码中 0、1 分布可知,分组码 01 的标签识别码中有且仅有一个 1,则可直接判断出碰撞位位置即为 1;分组码 001 的标签识别码中有且仅有一个 0,则可直接判断出碰撞位位置即为 0,由碰撞位置直接得出标签识别码,两者均不会产生空闲时隙。

定义分组码的识别顺序为:1、0001、01、001。

2.2 标签操作

将标签的状态分为激活状态、等待状态和睡眠状态三类。在初始化时,所有标签都处于激活状态,或是其分组码与阅读器发送请求序列相同,则处于激活状态;等待状态即标签暂不参与阅读器的通信,直到被激活;睡眠状态即标签已被识别,不再响应阅读器。标签状态转换关系如图 1 所示^[14]。

表 1 标签识别码编码表

标签识别码	分组码
000	1
001,010,100	01
110,101,011	001
111	0001

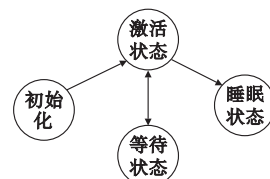


图1 标签状态转换

2.3 算法流程

阅读器向标签进行请求,根据分组码和碰撞位的位置判断具体的标签 ID,其主要步骤如下:

a) 阅读器发送初始化命令 Φ 。

b) 标签响应,并发送第 k 组的分组码。

c) 阅读器接收分组码,并根据碰撞的位置判断存在的分组码,分组码存入堆栈 Q 中。

d) 弹出堆栈 Q 中的分组码。当存在 1 或 0001 分组码时,阅读器发送分组码请求标签响应,如果响应信号没有碰撞,则识别标签,转步骤 f);发生碰撞,转步骤 e)。当存在 01 或 001 分组码时,阅读器将相应的标签置于激活状态后,标签发送其识别码,阅读器根据碰撞位的位置判断标签 ID,如果没有发生碰撞,则识别标签,转步骤 f);发生碰撞,则转步骤 e)。

e) 标签响应后 3 位识别码的分组码,并使 $k = k + 1$,转步骤 c)。

f) 判断堆栈 Q 是否为空,不为空则转步骤 c);为空,识别完毕。

IABC 算法流程如图 2 所示。

2.4 算法举例

下面举例说明 IABC 算法识别标签的步骤。假设有八个待识别标签的 ID 分别为 a:000001,b:111010,c:001110,d:101101,e:110111,f:100000,g:000000,h:100111。当接收到阅读器发送的查询命令后,标签发送 ID 前三位的分组码 1、0001、01、001、001、01、1、01;阅读器接收到 $\times \times \times 1$,根据碰撞位可知,存在分组码 1、01、001、0001。具体识别过程如表 2 所示。

运用八叉树算法识别上述八个标签时,阅读器需要向标签发送 24 次请求。而在 IABC 算法中无空闲时隙,阅读器识别八个标签只需发送 12 次请求,显著提高了标签的识别效率。

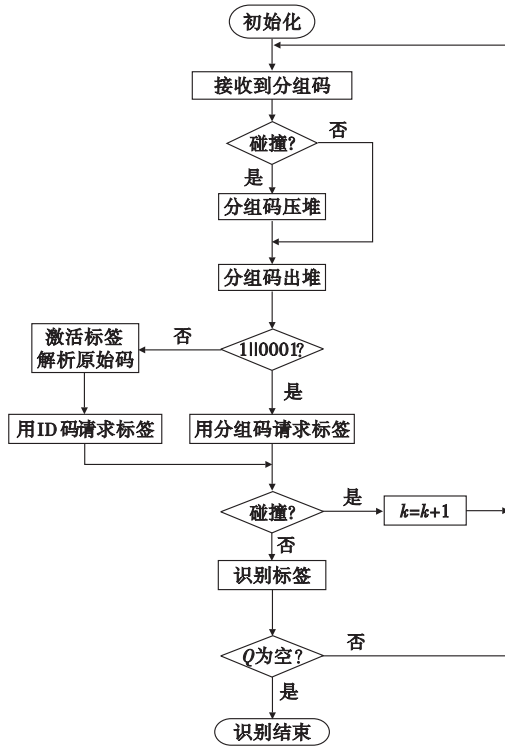


图2 IABC算法流程

表2 算法举例识别过程

序号	阅读器 查询命令	阅读器 接收信息	k	响应 标签	碰撞 信息	堆栈分组码状态 (带下划线的为 k=1 的分组码)
1	Φ	$\times \times \times 1$		全部	碰撞	{ <u>1</u> , 0001, 01, 001}
2	1	$\times 1$	1	a, g	碰撞	{1, 01, <u>0001</u> , 01, 001}
3	1		2	g	识别	{01, <u>0001</u> , 01, 001}
4	01		2	a	识别	{ <u>0001</u> , 01, 001}
5	0001	01	1	b	识别	{ <u>01</u> , 001}
6	01	$\times 0 \times 1$	1	c, f, h	碰撞	{1, <u>0001</u> , 001, <u>001</u> }
7	1		2	f	识别	{ <u>0001</u> , 001, <u>001</u> }
8	0001		2	h	识别	{001, <u>001</u> }
9	001		2	c	识别	{ <u>001</u> }
10	001	00 $\times 1$	1	d, e	碰撞	{ <u>0001</u> , 001}
11	0001		2	e	识别	{ <u>001</u> }
12	001		2	d	识别	Φ

3 性能分析

识别标签所消耗总时隙数是衡量防碰撞算法性能的重要指标。下面通过数学推导来计算 IABC 算法的总时隙数和吞吐率,进而分析算法的性能。

IABC 算法通过引入分组码使空闲时隙数降为 0。设 m 为待识别标签数, L 为搜索树的深度。由文献[16]分析可得八叉树算法的碰撞时隙数、空闲时隙数、识别时隙数和总时隙数(均为期望值)为

$$\overline{c_{TS}}(m) = \frac{1}{8}(\overline{t_{TS}}(m) - 1) \quad (1)$$

$$\overline{z_{TS}}(m) = \frac{7}{8}\overline{t_{TS}}(m) - m + \frac{1}{8} \quad (2)$$

$$\overline{s_{TS}}(m) = \overline{t_{TS}}(m) - m \quad (3)$$

其中:

$$\overline{t_{TS}}(m) = 1 + 8 \sum_{L=0}^{\infty} 8^L [1 - (1 - 8^{-L})^m - m 8^{-L} (1 - 8^{-L})^{m-1}] \quad (4)$$

IABC 算法中去除了空闲时隙,但在 01 和 001 这两组分组

码中会产生二次发送,即分组码 1 和 0001 与原码是一一对应;而分组码 01 和 001 分别有三组原码与之对应,因此后两组需进行二次发送,如图 3 所示。

IABC 算法中分组码的出现是非等概率。其中分组码 1 和 0001 出现的概率为 $\frac{1}{8}$, 分组码 01 和 001 出现的概率为 $\frac{3}{8}$ 。当标签 ID 的分组码为 01 和 001 时需进行二次发送,且是等概率的,均为 $\frac{1}{3}$ 。设 $\overline{t_1}(m)$ 、 $\overline{t_{01}}(m)$ 、 $\overline{t_{001}}(m)$ 、 $\overline{t_{0001}}(m)$ 分别为四种分组码所占的时隙数(均为期望值)。其中:

$$\overline{t_1}(m) = \overline{t_{0001}}(m) = \overline{t_1} \quad (5)$$

$$\overline{t_{01}}(m) = \overline{t_{001}}(m) = \overline{t_2} \quad (6)$$

则 IABC 算法识别标签的总时隙数为

$$\overline{t_{IABC}} = \overline{t_1}(m) + \overline{t_{01}}(m) + \overline{t_{001}}(m) + \overline{t_{0001}}(m) = 2\overline{t_1} + 2\overline{t_2} \quad (7)$$

由图 3 所示的概率分布可知,在 IABC 算法中,分组码 1 和 0001 所占时隙数为

$$\overline{t_1} = \frac{1}{8}(\overline{t_{TS}}(m) - \overline{z_{TS}}(m)) \quad (8)$$

分组码为 01 和 001 时所占时隙数为

$$\overline{t_2} = \frac{3}{8}(\overline{t_{TS}}(m) - \overline{z_{TS}}(m)) + \frac{1}{2}\overline{t_{again}}(m) \quad (9)$$

其中: $\overline{t_{again}}(m)$ 为二次发送的时隙数,将式(8)(9)代入式(7)可得

$$\overline{t_{IABC}}(m) = \frac{1}{4}(\overline{t_{TS}}(m) - \overline{z_{TS}}(m)) + \frac{3}{4}(\overline{t_{TS}}(m) - \overline{z_{TS}}(m)) + \overline{t_{again}}(m) \quad (10)$$

当分组码 01 和 001 并发生碰撞时才需二次发送。标签识别码二次发送的概率为 $\frac{3}{4}$,而八叉树算法中标签发生即为

$\frac{\overline{c_{TS}}(m)}{\overline{t_{TS}}(m)}$ 。所以,在 IABC 算法中,标签识别码二次发送的概率为 $\frac{3}{4} \times \frac{\overline{c_{TS}}(m)}{\overline{t_{TS}}(m)}$,则二次发送所占的时隙数为

$$\overline{t_{again}}(m) = \frac{3}{4}(\overline{t_{TS}}(m) - \overline{z_{TS}}(m)) \frac{\overline{c_{TS}}(m)}{4\overline{t_{TS}}(m)} \quad (11)$$

将式(11)代入式(10)得 IABC 算法的总时隙数为

$$\begin{aligned} \overline{t_{IABC}}(m) &= \frac{1}{4}(\overline{t_{TS}}(m) - \overline{z_{TS}}(m)) + \frac{3}{4}(\overline{t_{TS}}(m) - \overline{z_{TS}}(m)) + \\ &\quad \frac{9}{16}(\overline{t_{TS}}(m) - \overline{z_{TS}}(m)) \frac{\overline{c_{TS}}(m)}{\overline{t_{TS}}(m)} = \\ &\quad (\overline{t_{TS}}(m) - \overline{z_{TS}}(m)) (1 + \frac{9\overline{c_{TS}}(m)}{16\overline{t_{TS}}(m)}) \end{aligned} \quad (12)$$

IABC 算法的吞吐率为

$$S_{IABC} = \frac{m}{\overline{t_{IABC}}(m)} \quad (13)$$

4 仿真分析

图 4 所示为 IABC 算法的总时隙数、碰撞时隙数和空闲时隙数及 IABC 算法与四叉树算法、八叉树算法的碰撞时隙数比较的仿真结果。由图可见,IABC 算法的空闲时隙数为 0,其碰撞时隙数与八叉树算法相同,比四叉树算法少。所以,IABC 算法具有比八叉树更好的识别效率。

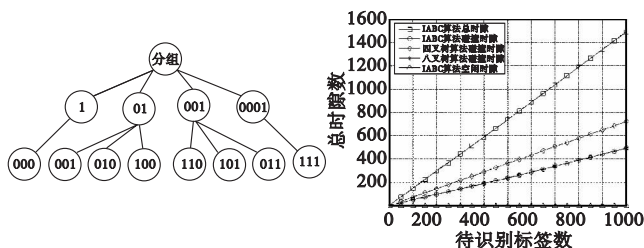


图3 分组码与标签识别码分布图

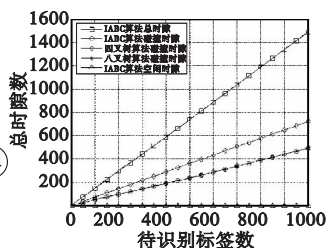


图4 IABC算法性能仿真

图5中对IABC算法、二叉树算法、四叉树算法和八叉树算法识别标签的总时隙进行了比较,可以看出,IABC算法的总时隙数明显要比二叉树、四叉树和八叉树算法少。当待识别标签数达到1000时,IABC算法比二叉树和四叉树算法少用了46.4%的时隙,比八叉树算法少用了62.5%的时隙。IABC算法采用分组码,将空闲时隙数降为零,使得总的时隙数也同时减少,所以IABC算法整体性能都要优于其他三种算法。

图6对IABC算法、二叉树算法、四叉树算法及八叉树算法的吞吐率进行了比较。可以看出,八叉树算法的吞吐率最低,在0.25~0.28;二叉树和四叉树算法的吞吐率基本相同,维持在0.35附近;而IABC算法的吞吐率能达到0.67左右,明显要比其他算法高。

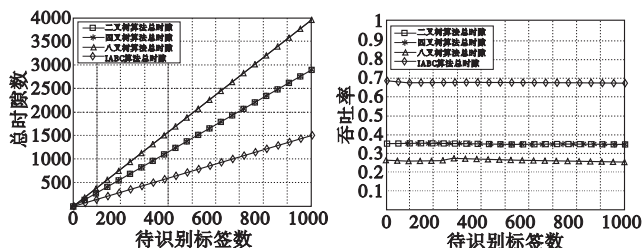


图5 四种算法性能仿真

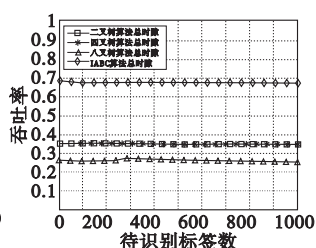


图6 四种算法吞吐率比较

5 结束语

本文采用分组码的方法去除了八叉树算法中的空闲时隙,提出的IABC算法的标签识别总时隙大大减少,识别效率明显提高。但分组码也仅在一层应用,在后续研究工作中将对标签识别码进行多层分类的算法研究,即将分组码再进行分类,或者是采用更有唯一性、更易分类的编码方式,进一步提高标签防撞算法的识别效率。

参考文献:

- [1] 张学军,蔡文琦,王锁萍.改进型自适应多叉树防撞算法研究[J].电子学报,2012,40(1):193-198.
- [2] ALI K, HASSANEIN H, TAHA A E M. RFID anti-collision protocol for dense passive tag environments[C]//Proc of the 32nd IEEE Conference on Local Computer Networks. Washington DC: IEEE Computer Society, 2007: 819-824.
- [3] 朱军,张元,卢小冬.基于分段搜索的多RFID标签抗冲突方法[J].计算机应用研究,2011,28(3):1031-1033.
- [4] 张学军,王娟,王锁萍.基于标签识别码分组的连续识别防撞算法研究[J].电子与信息学报,2011,33(5):1159-1165.
- [5] ZHEN B, KOBAYASHI M, SHIMIZU M. Framed ALOHA for multiple RFID objects identification[J]. IEICE Trans on Communications, 2005, E88-B(3): 991-999.
- [6] LEE S R, JOO S D, LEE C W. An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification[C]//Proc of the 2nd International Conference Mobile and Ubiquitous Systems: Networking and Service. Washington DC: IEEE Computer Society, 2005: 166-174.
- [7] 崔沂峰,陈平,张景元. RFID电子标签防撞算法的研究[J].微计算机信息,2007,23(8-2):233-234.
- [8] CHOI J H, LEE D, JEON H, et al. Enhanced binary search with time-divided responses for efficient RFID tag anti-collision[C]//Proc of IEEE International Conference on Communications. 2007: 3853-3858.
- [9] KIM Y H, KIM S S, LEE S J. Improved 4-ary query tree algorithm for anti-collision in RFID system[C]//Proc of International Conference on Advanced Information Networking and Applications. Washington DC: IEEE Computer Society, 2009: 699-704.
- [10] LAI Yuan-cheng, LIN C C. Two blocking algorithms on adaptive binary splitting: single and pair resolutions for RFID tag identification[J]. IEEE/ACM Trans on Networking, 2009, 17(3): 962-975.
- [11] CHEN Y H, HORNG S J, TUN R S. A novel anti-collision algorithm in RFID systems for identifying passive tags[J]. IEEE Trans on Industrial Informatics, 2010, 6(1): 105-121.
- [12] MYUNG J, LEE W, SHIH T K. An adaptive memory-less protocol for RFID tag collision arbitration[J]. IEEE Trans on Multimedia, 2006, 8(3): 1096-1101.
- [13] HUSH D R, WOOD C. Analysis of tree algorithms for RFID arbitration[C]//Proc of IEEE Symposium on Information Theory. 1998: 107-116.
- [14] ZEIN-ELABDEEN W, SHAABAN E. An enhanced binary tree anti-collision technique for dynamically added tags in RFID systems[C]//Proc of the 2nd International Conference on Computer Engineering and Technology. 2010: V7349-V7353.
- [6] PINNEGAR C, MANSINHA L. The bi-Gaussian S-transform[J]. SIAM Journal on Scientific Computing, 2003, 24(5): 1678-1692.
- [7] 陈学华,贺振华.广义S变换及其时频滤波[J].信号处理,2008,24(1):28-32.
- [8] ROBERT C, PINNEGAR C, MANSINHA C. An method of time-time analysis: the TT-transform digital[J]. Digital Signal Processing, 2003, 13(4): 588-593.
- [9] SIMON C, SCHIMMEL M, DANOBIETIA J. On the TT-transform and its diagonal elements[J]. IEEE Trans on Signal Processing, 2008, 56(11): 5709-5714.
- [10] PINNGAR C R. Time-frequency and time-time filtering with the S-transform and TT-transform[J]. Digital Signal Processing, 2005, 15(10): 604-620.
- [11] FAN Xian-feng, ZUO Ming. Gearbox fault detection using Hilbert and TT-tranform[J]. Key Engineering Materials, 2005, 293(7): 79-86.

(上接第4256页)特点。仿真实验表明,相比S变换滤波与TT变换滤波,该方法滤波后信号信噪比更高,重构信号与原信号误差更小。

参考文献:

- [1] 李桂元. f-k域滤波假频的消除方法[J].石油地球物理勘探, 1994, 29(S1): 86-92.
- [2] 周知进,文泽军,卜英勇.小波降噪在超声回波信号处理中的应用[J].仪器仪表学报,2009,30(2):237-241.
- [3] 刘宝亮,姚慧,郑海荣.小波去噪在瞬时弹性成像估计中的应用[J].计算机仿真,2011,28(8):246-249.
- [4] STOCKWELL R G, MANSINA L, LOWE R P. Localization of the complex spectrum: the S transform[J]. IEEE Trans on Signal Processing, 1996, 44(4): 998-1001.
- [5] STOCKWELL R G. A basis for efficient representation of the S-transform[J]. Digital Signal Processing, 2007, 17(1): 371-393.

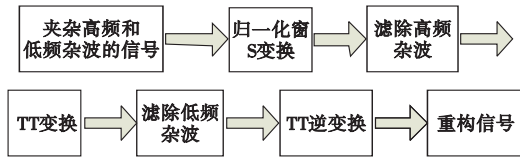


图2 滤波流程

2.1 基于归一化窗S变换的时频滤波

2.1.1 瞬时频率

估计瞬时频率的算法很多,有 Prony 算法、希尔伯特—黄变换算法、相位法、谱峰检测法等。S 变换含有相位信息,可以直接从信号的 S 变换结果计算瞬时频率,大大减少了计算量。对信号进行 S 变换,得到复时频谱 $S(\tau, f)$ 为

$$S(\tau, f) = |A(\tau, f)| \exp[i\beta(\tau, f)] \quad (13)$$

式中: $A(\tau, f)$ 为瞬时振幅, $\beta(\tau, f)$ 为瞬时相位, 而且

$$A(\tau, f) = \sqrt{\{ \operatorname{Re}[S(\tau, f)] \}^2 + \{ \operatorname{Im}[S(\tau, f)] \}^2} \quad (14)$$

$$\beta(\tau, f) = \arctan \left\{ \frac{\operatorname{Im}[S(\tau, f)]}{\operatorname{Re}[S(\tau, f)]} \right\} \quad (15)$$

由相位可以得到瞬时频率为

$$F(\tau, f) = \frac{\partial \beta(\tau, f)}{\partial \tau} \quad (16)$$

2.1.2 二维时频滤波器

二维时频滤波克服了传统一维时频滤波只能在时域或频域滤波的局限性,能更精确地滤波。归一化窗 S 变换时频分布用于非平稳信号的滤波,可以滤除信号在某一时间段和频带内的信号分量。先对原信号作归一化窗 S 变换,获得该信号的时频分布,然后根据特定信号分量在时频平面中的位置(即时频通域)设计时频滤波器。

$$\int_{-\infty}^{+\infty} S_1(\tau, f) d\tau = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} u(t) \frac{|Bf|^{0.5}}{\pi^{0.25}} e^{-\frac{\beta^2 f^2 (\tau-t)^2}{2}} e^{-i2\pi ft} dt d\tau \quad (17)$$

令 $\delta = \tau - t$, 则

$$\int_{-\infty}^{+\infty} \frac{|Bf|^{0.5}}{\pi^{0.25}} e^{-\frac{\beta^2 f^2 \delta^2}{2}} d\delta = \pi^{0.25} \sqrt{\frac{2}{|Bf|}} \quad (18)$$

把式(18)代入式(17)中得到

$$\int_{-\infty}^{+\infty} S_1(\tau, f) d\tau = \int_{-\infty}^{+\infty} u(t) \pi^{0.25} \sqrt{\frac{2}{|Bf|}} e^{-i2\pi ft} dt = \pi^{0.25} \sqrt{\frac{2}{|Bf|}} U(f) \quad (19)$$

即

$$U(f) = \pi^{-0.25} \sqrt{\frac{|Bf|}{2}} \int_{-\infty}^{+\infty} S_1(\tau, f) d\tau$$

所以时频滤波器为

$$u'(t) = \int_{-\infty}^{+\infty} \pi^{-0.25} \sqrt{\frac{|Bf|}{2}} \int_{-\infty}^{+\infty} S_1(\tau, f) H(\tau, f) d\tau \times e^{i2\pi ft} df \quad (20)$$

$$H(n, k) = \begin{cases} 1 & k \in \left[\frac{f_i(n) - B(n)}{2}, \frac{f_i(n) + B(n)}{2} \right] \\ 0 & \text{其他} \end{cases} \quad (21)$$

其中: $H(\tau, f)$ 是时频滤波算子; $f_i(n)$ 为瞬时频率; $B(n)$ 为滤波器带宽, 根据信号能量在时频平面上沿瞬时频率聚集程度来确定。通过对归一化窗 S 变换矩阵二维时频分析, 可以确定不同时间段信号所包含频率成分, 运用时频滤波器把高频杂波和噪声滤除掉。仿真数据: 雷达的脉冲持续时间 $t = 10 \mu\text{s}$, 调制带宽 30 MHz, 雷达目标有效截面 $\text{RCS} = [1 \ 1 \ 1 \ 1 \ 1]$, 采样频率 150 MHz。图 3 中, (a) 是夹杂着低频和低频杂波的雷达回波信号, 信噪比是 0.6 dB; (b) 是信号的归一化窗 S 变换谱; (c) 是滤除高频杂波和噪声后的归一化窗 S 变换谱, 对比 (b) 可

知, 很有效地滤除了高频杂波。

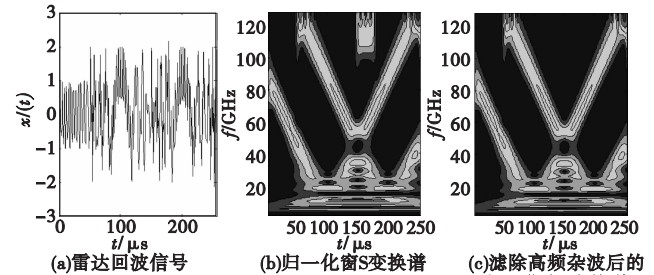


图3 雷达回波变换

2.2 基于 TT 变换的时频滤波

图 4 中, (a) 是基于图 3(c) 的 TT 变换谱; (b) 是提取对角线附近元素后的 TT 变换谱; (c) 是 TT 变换滤波后的归一化窗 S 变换时频谱, 对比图 3(c) 可知, 有效滤除了低频波, 同时保留了高频的有效信号。图 5 中, (a) 是本文方法滤波后重构信号与原信号的误差值; (b) 是单独用 TT 变换滤波后重构信号与原信号的误差值; (c) 是 S 变换时频滤波后重构信号与原信号的误差值。从图 5 可以看出, S 变换与 TT 变换滤波后重构信号与原信号误差的均方差比本文方法大, 也就是说, 本文方法更有效地保留了原信号的特征。表 1 是 S 变换、TT 变换及本文方法具体结果对比, 可知相较于其他两种方法, 本文方法能更好地滤除杂波和保留信号的特征。

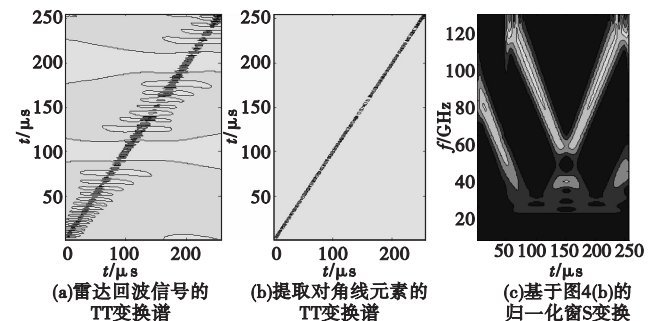


图4 对图3信号的TT变换

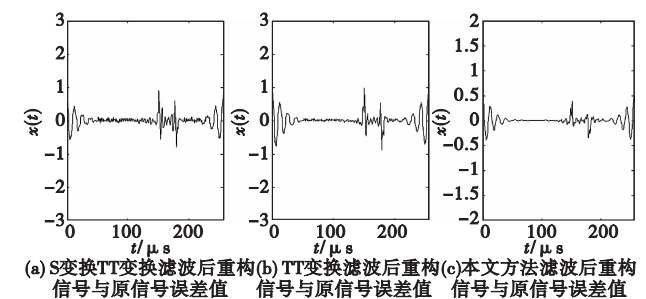


图5 重构信号与原信号误差

表1 滤波结果对比

测试项目	滤波后信噪比/dB	均方误差
TT 变换	1.76	0.135 1
S 变换	1.83	0.101 1
归一化窗 S 变换和 TT 变换	1.92	0.084 4

3 结束语

本文提出一种基于归一化窗 S 变换和 TT 变换的时频滤波方法, 该方法利用归一化窗 S 变换滤除高频杂波, TT 变换滤除低频面波, 克服了单独使用 TT 变换滤波的不足, 又没有使用条件的限制, 同时有适用范围广、计算简便等 (下转第 4268 页)