

用窗口卷积实现反卷积的算法研究

赵永平, 许 鹏, 杨 硕

(哈尔滨工业大学自动化测试与控制系, 黑龙江哈尔滨 150001)

摘 要: 本文提出了一种用窗口卷积实现反卷积的新算法, 这种算法解决了用圆周卷积计算反卷积时出现的病态问题. 对用圆周卷积、线性卷积和窗口卷积实现反卷积三种方法进行误差、计算量与存储空间的分析 and 对比. 结果表明, 用窗口卷积实现反卷积运算的误差、计算量和存储空间都是最小的.

关键词: 窗口卷积; 卷积; 反卷积; 误差; 病态问题

中图分类号: TN911.72 **文献标识码:** A **文章编号:** 0372-2112 (2005) 08-1533-04

Study on an Algorithm of Deconvolution Via Window Convolution

ZHAO Yong-ping, XU Peng, YANG Shuo

(Department of Automatic Measurement and Control, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China)

Abstract: A new algorithm of deconvolution via window convolution operation is proposed, and the algorithm can solve the ill-posed problem occurring in deconvolution via circular convolution. In addition, the estimation error, the calculating amounts and the occupied memory of the three algorithms of deconvolution that include circular convolution, linear convolution and window convolution are also discussed and compared. The results show that factors of the new algorithm are the smallest.

Key words: window convolution; convolution; deconvolution; error; ill-posed problem

1 引言

在采用时域测量方式时, 经常会遇到反卷积运算. 对于数字信号处理中系统识别和信号恢复问题可以用反卷积方法求解. 当测量一个系统、或某个网络、或者某一个器件的冲击响应时, 需要完成如下的反卷积运算:

$$h(n) = y(n) (1/ \ast) x(n) \quad (1)$$

用符号“ $1/ \ast$ ”表示反卷积运算, $x(n)$ 为系统输入激励信号, $y(n)$ 为系统输出响应, $h(n)$ 为单位样值响应. 当重建系统的激励信号时, 需要完成如下的反卷积运算:

$$x(n) = y(n) (1/ \ast) h(n) \quad (2)$$

在数学形式上, 式(1)和式(2)具有完全相同的形式, 所以分析其中一个所得结论同样也适用于另一个. 分别计算序列 $x(n)$ 、 $y(n)$ 和 $h(n)$ 的频谱 $X(k)$ 、 $Y(k)$ 和 $H(k)$, 得到频域反卷积运算表达式

$$\begin{aligned} h(n) &= \text{IFFT}[H(k)] = \text{IFFT}[Y(k)/X(k)] \\ x(n) &= \text{IFFT}[X(k)] = \text{IFFT}[Y(k)/H(k)] \end{aligned} \quad (3)$$

理论分析表明, 如果数据精确, 或者说不存在测量误差, 则由式(1)~(3)确定的反卷积的解是唯一的且是稳定的^[1].

但是, 如果 $X(k)$ 或 $H(k)$ 中有零点, 那么 $Y(k) = X(k)H(k) = 0$. 此时将出现的不确定状态, 针对这个问题人们已经做过很多有益的研究工作^[2,3]. 由于存在测量误差和噪声的干扰, 就可能使这些点上的 $H(k)$ 或 $X(k)$ 的值发散. 用卷积

运算实现反卷积的方法直观上可以避免 $X(k)$ 或 $H(k)$ 的零点^[4].

当一个线性时不变系统作用一个样值函数 $A\delta(n)$ 时, 其输出(零状态)响应为 $Ah(n)$, 即

$$Ah(n) = A\delta(n) \ast h(n) \quad (4)$$

其中, $\delta(n)$ 为单位样值函数, $h(n)$ 为单位样值响应. 由式(4)表明, 如果使输入 $x(n)$ 变换成 $A\delta(n)$, 根据输出 $y(n)$ 就可以得到单位样值响应 $h(n)$. 用卷积计算反卷积的方法就是基于这一思想.

2 用圆周卷积和线性卷积计算反卷积

2.1 圆周卷积计算反卷积

这一方法和原理在文献[4]中已详细论述, 这里只给出必要的结果. 设输入序列 $x(n)$ 长度为 N_x , 单位冲击响应序列 $h(n)$ 长度为 N_h , 输出序列 $y(n)$ 长度为 N_y , 且三者满足关系式: $N_y = N_x + N_h - 1$. 设 $p = \lceil \log_2 N_y \rceil$, 且 $N = 2^p$, 符号 $\lceil a \rceil$ 表示对数 a 向 $+\infty$ 方向取整. 因为

$$y(n) = h(n) \ast x(n) \quad n = 0, 1, 2, \dots, N_y - 1 \quad (5)$$

所以可将序列 $x(n)$ 、 $h(n)$ 和 $y(n)$ 补零使其长度都为 N . 由圆周卷积定义知

$$y(n) = h(n) \odot x(n) = \sum_{m=0}^{N-1} h(m) x((n-m)_N) R_N(n) \quad (6)$$

进行 p 次圆周卷积运算, 得到

收稿日期: 2004-08-09; 修回日期: 2005-03-16

$$y_p(n) = h(n) \odot x_p(n) = h(n) \odot A\delta(0) \quad (7)$$

求得单位样值响应 $h(n)$:

$$h(n) = y_p(n) / A \quad (8)$$

2.2 线性卷积计算反卷积

设 $p = \lceil \log_2 N_h \rceil$, 且 $N = 2^p$. 进行 p 次线性卷积运算得到序列 $x_p(n)$ 的每 N 个元素中仅含一个非零元素, 其长度为 $N(N_x - 1) + 1$. 序列 $y_p(n)$ 的长度为 $(N - 1)(N_x - 1) + N_y$, N_y 为序列 $y(n)$ 的长度. 而且 $y_p(n) / x_p(n)$ 将包含至少 $N_x - 1$ 组 $h(n)$ 的解. 随着 N 的增加, 序列 $x_p(n)$ 与 $y_p(n)$ 的长度急剧增加, 不但增加了计算量, 而且占用了大量的存储空间.

3 用窗口式卷积运算实现反卷积

设想如果在整个反卷积运算过程中, 始终保持一个 $h(n)$ 序列长度的计算, 就可以大大的减少计算量, 而且节省了存储空间. 因此提出了改进算法, 即用窗口式卷积计算反卷积的方法. 设 $p = \lceil \log_2 N_h \rceil$, 即 $N = 2^p$.

由 $y(n) = h(n) * x(n)$ 可得,

$$y(n) R_{N_h}(n) = (h(n) R_{N_h}(n)) * (x(n) R_{N_h}(n)) \quad (9)$$

因此,

$$h(n) = h(n) R_{N_h}(n) = (y(n) R_{N_h}(n)) (1 / *) (x(n) R_{N_h}(n))$$

$$n = 0, 1, 2, \dots, N_h - 1 \quad (10)$$

上式说明用窗口式卷积计算反卷积的方法是可行的.

设定序列

$$x^*(n) = (-1)^n x(n) R_{N_h}(n) \quad (11)$$

$$x_1(n) = (x^*(n) * x(n)) R_{N_h}(n) = (-1)^n x_1(n) \quad (12)$$

式(12)表明, 当 n 为奇数时, 则 $x_1(n) \equiv 0$. 经过一次线性卷积运算, 得到的序列 $x_1(n)$ 中连续两个数中有一项不为零. 序列 $y_1(n)$ 等于

$$y_1(n) = (y(n) * x^*(n)) R_{N_h}(n) \quad (13)$$

再构造序列

$$x_1^*(n) = (-1)^{n/2} x_1(n) R_{N_h}(n) \quad (14)$$

以此类推, 序列 $x_i(n)$ 中连续的 2^i 个数中有一项不为零, 得到序列 $x_i^*(n) = (-1)^{n/2^i} x_i(n) R_{N_h}(n)$.

当经过 p 次上述的运算, 得到序列 $x_p(n)$ 等于

$$x_p(n) = (x(n) * x^*(n) * x_1^*(n) \dots x_{p-1}^*(n)) R_{N_h}(n) \quad (15)$$

并且序列 $x_p(n)$ 只有第一项 $x_p(0)$ 不为零, 序列长度仍为 N_h , 即

$$x_p(n) = A\delta(n) \quad (16)$$

同理可以得到序列 $y_p(n)$ 为

$$y_p(n) = (y(n) * y^*(n) * y_1^*(n) \dots y_{p-1}^*(n)) R_{N_h}(n) \quad (17)$$

并且序列 $y_p(n)$ 的长度仍为 N_h .

求得单位样值响应 $h(n)$, 即

$$h(n) = \frac{1}{A} \cdot y_p(n) \quad (18)$$

在计算时应当注意的是, 当 $N_x < N_h$ 时, 必须给序列 $x(n)$ 补零, 使得 $N_x = N_h$, 这样最终才能解得正确的序列

$h(n)$; 而当 $N_x > N_h$ 时, 可以将序列 $x(n)$ 截断, 使得 $N_x = N_h$. 而且可以看出在改进算法的计算过程中, 序列长度始终保持为, 这样不但减少了计算量, 而且节省了存储空间.

4 误差分析

由于测量误差或噪声会对各种算法的反卷积运算产生很大的影响^[5,6,7]. 所以有必要对用卷积计算反卷积这种方法的误差作以分析.

4.1 圆周卷积计算反卷积的误差分析

由 $x(n)$ 、 $y(n)$ 经过 $p_1 = \lceil \log_2 N_y \rceil$ 步圆周卷积计算得 $x_{p_1}(n)$ 、 $y_{p_1}(n)$, 设 $N_1 = 2^{p_1}$. 设 $x(n)$ 的相对误差为 e_x , $y(n)$ 的相对误差为 e_y . 由文献[8]的结果可得 $h(n)$ 的相对误差为

$$e_h = (2N_1 - 1) |e_x| + |e_y| \quad (19)$$

4.2 窗口式卷积计算反卷积的误差分析

由 $x(n)$ 、 $y(n)$ 经过 $p_2 = \lceil \log_2 N_h \rceil$ 步线性卷积计算得 $x_{p_2}(n)$ 、 $y_{p_2}(n)$, 设 $N_2 = 2^{p_2}$. 设 $x(n)$ 的相对误差为 e_x , $y(n)$ 的相对误差为 e_y . 由

$$x_{p_2}(n) = x(n) * x^*(n) * x_1^*(n) * \dots * x_{p_2-1}^*(n) \quad (20)$$

$$y_{p_2}(n) = y(n) * x^*(n) * x_1^*(n) * \dots * x_{p_2-1}^*(n) \quad (21)$$

得 $x_{p_2}(n)$ 的相对误差为 $2^{p_2} e_x$, $y_{p_2}(n)$ 的相对误差为 $(2^{p_2} - 1) \cdot e_x + e_y$. 再由 $h(n) = y_{p_2}(n) / x_{p_2}(n)$, 得 $h(n)$ 的相对误差为

$$e_h = (2^{p_2+1} - 1) |e_x| + |e_y| = (2N_2 - 1) |e_x| + |e_y| \quad (22)$$

同理若 $x(n)$ 、 $y(n)$ 经过 $p_2 = \lceil \log_2 N_h \rceil$ 步线性卷积计算可得 $x'_{p_2}(n)$ 、 $y'_{p_2}(n)$, 最终得到 $h(n)$ 的相对误差亦为 $e_h = (2N_2 - 1) |e_x| + |e_y|$.

对三种算法的误差进行比较: 因为 $N_y = N_x + N_y - 1$, 所以 $N_y > N_h$. 可得 $p_1 > p_2$, 相对误差 $e_y > e_h$. 对这三种算法的误差分析可知, 用改进的窗口式卷积和线性卷积实现反卷积算法的相对误差较小.

5 三种反卷积方法的计算量分析及比较

在实际计算反卷积运算时, 要考虑算法的计算量和计算过程中占用存储空间的大小. 算法的计算量直接影响计算时间和使用效率, 在用卷积计算反卷积的算法中主要的计算量是乘法计算. 设 $p_1 = \lceil \log_2 N_y \rceil$, $N_1 = 2^{p_1}$, $p_2 = \lceil \log_2 N_h \rceil$, $N_2 = 2^{p_2}$.

5.1 圆周卷积计算反卷积的计算量分析

由 $x(n)$ 、 $y(n)$ 经过 $p_1 = \lceil \log_2 N_y \rceil$ 次圆周卷积计算得到 $x_{p_1}(n)$ 、 $y_{p_1}(n)$. 由 $x(n)$ 计算 $x_1(n)$ 乘法运算 N_1^2 次, 由 $y(n)$ 计算 $y_1(n)$ 乘法运算 N_1^2 次. 同理可得, 由 $x_{p_1-1}(n)$ 计算 $x_{p_1}(n)$ 乘法运算 N_1^2 次; 由 $y_{p_1-1}(n)$ 计算 $y_{p_1}(n)$ 乘法运算 N_1^2 次.

经过分析可得由初始序列 $x(n)$ 和 $y(n)$ 经过圆周卷积计算最终得到反卷积结果 $h(n)$, 要经过 $2N_1^2 \log_2 N_1$ 次乘法运算. 在用圆周卷积计算反卷积的过程中, 序列长度保持为 $N_1 = 2^{p_1}$.

5.2 线性卷积计算反卷积的计算量分析

由 $x(n)$ 、 $y(n)$ 经过 $p_2 = \lceil \log_2 N_h \rceil$ 次线性卷积计算得到

$x_{p_2}(n), y_{p_2}(n)$. 由 $x(n)$ 计算 $x_1(n)$ 乘法运算次数 N_x^2 . 由 $y(n)$ 计算 $y_1(n)$ 乘法运算次数 $N_x \cdot N_y$. 同理可得, $x_{p_2}(n) = x_{p_2-1}(n) * x_{p_2-1}^*(n)$, 由 $x_{p_2-1}(n)$ 计算 $x_{p_2}(n)$ 乘法运算次数 $\left[\frac{N_2}{2}(N_x - 1) + 1\right]^2 \cdot y_{p_2}(n) = y_{p_2-1}(n) * x_{p_2-1}^*(n)$ 序列 $y_{p_2-1}(n)$, 由 $y_{p_2-1}(n)$ 计算 $y_{p_2}(n)$ 乘法运算次数 $\left[\left(\frac{N_2}{2} - 1\right)(N_x - 1) + N_y\right] \left[\frac{N_2}{2}(N_x - 1) + 1\right]$.

可得由初始序列 $x(n)$ 和 $y(n)$ 经过线性卷积计算最终得到反卷积结果 $h(n)$, 要经过 $\frac{2}{3} N_x^2 N_y^2$ 量级次乘法运算. 在用线性卷积计算反卷积的过程中, 序列长度不断增大. 序列 $y_{p_2}(n)$ 长度为 $(N_2 - 1)(N_x - 1) + N_y$, 序列 $x_{p_2}(n)$ 长度为 $N_2(N_x - 1) + 1$, 占用了大量的存储空间.

5.3 窗口式卷积计算反卷积的计算量分析

由 $x(n), y(n)$ 经过 $p_2 = \lceil \log_2 N_h \rceil$ 次线性卷积计算得到 $x_{p_2}(n), y_{p_2}(n)$. 由 $x(n)$ 计算 $x_1(n)$ 乘法运算次数 $\frac{N_h(N_h+1)}{2}$; 由 $y(n)$ 计算 $y_1(n)$ 乘法运算次数 $\frac{N_h(N_h+1)}{2}$. 同理可得, $x_{p_2}(n) = (x_{p_2-1}(n) * x_{p_2-1}^*(n)) R_{N_h}(n)$ 由 $x_{p_2-1}(n)$ 计算 $x_{p_2}(n)$ 乘法运算 $\frac{N_h(N_h+1)}{2}$ 次; $y_{p_2}(n) = (y_{p_2-1}(n) * x_{p_2-1}^*(n)) R_{N_h}(n)$ 由 $y_{p_2-1}(n)$ 计算 $y_{p_2}(n)$ 乘法运算 $\frac{N_h(N_h+1)}{2}$ 次.

可得由初始序列 $x(n)$ 和 $y(n)$ 经过窗口式卷积计算最终得到反卷积结果 $h(n)$, 要经过 $N_h(N_h+1) \cdot \lceil \log_2 N_h \rceil$ 次乘法运算. 在用窗口式卷积计算反卷积的过程中, 序列长度保持为 N_h .

对三种反卷积方法进行比较, 如表 1 所示. 由比较可知三种计算反卷积的算法中, 用窗口式卷积计算反卷积算法的计算量及计算过程中占用的存储空间都是最小的.

表 1 三种反卷积方法的比较

计算反卷积方法	计算量	序列存储空间
圆周卷积	$2N_1^2 \log_2 N_1$	N_1
线性卷积	$\frac{2}{3}(N_x - 1)^2 N_y^2$	$(N_2 - 1)(N_x - 1) + N_y$
窗口式卷积	$N_h(N_h + 1) \log_2 N_2$	N_h

6 用改进算法解决圆周卷积计算反卷积的病态问题

用圆周卷积运算实现反卷积时, 会出现 $x_p(n) = [0, 0, \dots, 0]$. 这样就无法求得 $h(n) = y_p(n) / x_p(n)$. 经过分析, 可知出现这种问题是由于 $x_{p-1}(0) = A, x_{p-1}(N/2) = \pm A, x_{p-1}^*(0) = A, x_{p-1}^*(N/2) = \mp A$ 造成的^[8]. 而用线性卷积或改进的窗口式卷积计算反卷积时, 都可以解决这种病态问题. 但是用线性卷积计算反卷积时不但计算量大, 而且占用了大量的存储空间, 所以用改进的窗口式卷积计算反卷积的方法更实用. 下面举例来说明用改进算法解决这类问题.

例: 已知系统的输入和输出分别为 $x(n) = [1, 1, 1, 1, 0, 0, 0, 0], y(n) = [1, 7, 4, 3, 3, -3, 0, 1]$, 求系统的单位抽样响应 $h(n)$.

解: (1) 根据圆周卷积计算反卷积的原理可得此例中 $N = 8, p = 3$. 计算过程在文献[9]中已论述, 结论如下:

$$x_3(n) = [0, 0, 0, 0, 0, 0, 0, 0]$$

从此例可以看出, 由于 $x_3(n) = [0, 0, 0, 0, 0, 0, 0, 0]$, 产生所谓的病态问题, 不能得到 $h(n)$.

(2) 用改进的窗口式卷积计算反卷积的方法可以解决这种病态问题. 此例中, $N_x = 4, N_y = 8, N_h = 5$, 于是可得 $N = 8$, 即 $p = 3$. 根据窗口卷积计算反卷积的方法原理计算过程如下:

$$x(n) = [1, 1, 1, 1, 0] \quad x^*(n) = [1, -1, 1, -1, 0]$$

$$x_1(n) = x(n) * x^*(n) = [1, 0, 1, 0, -1]$$

$$y_1(n) = (y(n) * x^*(n)) R_{N_h}(n) = [1, 6, -2, 5, -3]$$

$$x_1^*(n) = [1, 0, -1, 0, -1]$$

$$x_2(n) = x_1(n) * x_1^*(n) = [1, 0, 0, 0, -3]$$

$$y_2(n) = (y_1(n) * x_1^*(n)) R_{N_h}(n) = [1, 6, -3, -1, -2]$$

$$x_2^*(n) = [1, 0, 0, 0, 3]$$

$$x_3(n) = x_2(n) * x_2^*(n) = [1, 0, 0, 0, 0]$$

$$y_3(n) = (y_2(n) * x_2^*(n)) R_{N_h}(n) = [1, 6, -3, -1, 1]$$

$$h(n) = y_3(n) / x_3(n) = [1, 6, -3, -1, 1]$$

7 结论

由于用圆周卷积和线性卷积运算实现反卷积的方法存在计算量大, 占用存储空间大, 存在病态问题等缺点, 本文提出了一种用窗口卷积运算实现反卷积的方法. 对此方法进行误差分析, 计算量分析及运算过程中占用存储空间的分析, 并和原方法作比较可知窗口卷积的误差、计算量和占用存储空间都有减小. 通过实例说明用窗口卷积运算实现反卷积的运算过程中不存在用圆周卷积运算实现反卷积时出现的病态问题.

参考文献:

- [1] S M Riad. The deconvolution problem: an overview[J]. Proc IEEE, 1986, 74(1): 82 - 85.
- [2] 舒勤, 张有正. $X[k]$ 有零点的卷积反演的 DFT 算法[J]. 电子学报, 1990, 18(3): 83 - 89.
SHU Q, ZHANG Y Z. The DFT algorithms of deconvolution for $X[k]$ having zeros[J]. Acta Electronica Sinica, 1990, 18(3): 83 - 89.
- [3] 舒勤, 张有正. 再论 $X[k]$ 有零点的卷积反演的 DFT 算法[J]. 电子学报, 1992, 20(12): 79 - 81.
SHU Q, ZHANG Y Z. Rediscover the DFT algorithm of deconvolution for $X[k]$ having zeros[J]. Acta Electronica Sinica, 1992, 20(12): 79 - 81.
- [4] 孔凡年. 时域卷积反演的新方法[J]. 电子学报, 1985, 13(4): 9 - 14.
KONG F N. A new method of deconvolution in the time domain[J]. Acta Electronica Sinica, 1985, 13(4): 9 - 14.

- [5] A Bennia ,S M Riad. An optimization technique for iterative frequency-domain deconvolution[J]. IEEE Trans Instr Meas ,1990 ,IM39(2) :358 - 362.
- [6] T Dhaene ,L Martens. Extended bennia-riad criterion for iterative frequency-domain deconvolution[J]. IEEE Trans Instr Meas ,1994 ,IM42(2) :176 - 180.
- [7] 刘明亮,白瑞林. 测量误差对反卷积影响的分析及其滤波器的设计思想[J]. 电子测量与仪器学报,1994,8(3) :44 - 49.
LIU M L ,BAI R L. The analysis of measuring error for deconvolution effect and the design thinking of its filter [J]. Journal of Electronic Measurement and Instrument ,1994 ,8(3) :44 - 49.
- [8] 刘明亮,蔡永泉,等. 用卷积运算实现反卷积[J]. 电子学报,2000,28(5) :111 - 112.
LIU M L ,CAI Y Q ,et al. Deconvolution via convolution operation[J]. Acta Electronica Sinica ,2000 ,28(5) :111 - 112.

作者简介:



赵永平 男,1964 年 1 月生于黑龙江哈尔滨,哈尔滨工业大学自动化测试与控制系副教授,主要研究方向为汽车电子技术、信息工程. E-mail :zhaoy2590 @hit. edu. cn.



许 鹏 男,1980 年 9 月生于黑龙江哈尔滨,哈尔滨工业大学自动化测试与控制系硕士研究生,主要研究方向为信息工程、多功能传感技术. E-mail :noonmoonwing @yahoo. com. cn.