

一种改进的补偿因子区域不变 CS 算法 及 Simulink 模型建模

宋栩潮¹, 周 鹏¹, 张振华², 肖忠源²

(1 中国石油大学(华东)海洋与空间信息学院 青岛 266580;

2 北京遥测技术研究所 北京 100076)

摘要: 相位补偿因子、距离向补偿因子和方位向补偿因子的产生是合成孔径雷达实时 CS (Chirp Scaling, 调频变标) 算法中的关键。针对一种现有的补偿因子区域不变 CS 算法进行了改进, 对于更新步长内各单元的统一补偿因子, 采用该区域内所有单元频率平均值对应的补偿因子, 代替原来使用的第一个单元对应的补偿因子, 使得最终的补偿结果相比于原有算法更加均匀, 计算量与原有算法相当。经过 Matlab 仿真验证, 改进算法的成像指标得到提升。为便于生成 FPGA 代码, 采用 Simulink 工具搭建了用于生成改进算法中三种补偿因子的模型。将 Simulink 模型输出的代码加载到 Vivado 软件中, 对 Vivado 输出的补偿因子与由 Matlab 输出的精确补偿因子进行了对比, 精度满足要求, 验证了所搭建 Simulink 模型的准确性。

关键词: 合成孔径雷达; 实时成像; 调频变标算法; Matlab/Simulink; FPGA

中图分类号: TN957.52; V443+2 文献标志码: A 文章编号: 2095-1000(2023)01-0099-12

DOI: 10.12347/j.ycyk.20220530001

引用格式: 宋栩潮, 周鹏, 张振华, 等. 一种改进的补偿因子区域不变 CS 算法及 Simulink 模型建模[J]. 遥测遥控, 2023, 44(1): 99-110.

An improved chirp scaling algorithm featured by invariance of compensation factors in a region and its modeling by Simulink

SONG Xuchao¹, ZHOU Peng¹, ZHANG Zhenhua², XIAO Zhongyuan²

(1. College of Oceanography and Space Informatics, China University of Petroleum, Qingdao 266580, China;

2. Beijing Research Institute of Telemetry, Beijing 100076, China)

Abstract: The generation of the phase compensation factor, the range compensation factor and the azimuth compensation factor are the key in the real-time CS (chirp scaling) algorithm of synthetic aperture radar. An existing CS algorithm featured by the invariance of compensation factors in a region is improved. For the unified compensation factor in units inside the update step size, the compensation factor corresponding to the average frequency of all units in a region is used to replace the compensation factor corresponding to the first unit, which makes the final compensation result using the improved algorithm is more uniform than that of the original algorithm. The computation load of the improved algorithm is equivalent to that of the original algorithm. Through Matlab simulation, the imaging performance of the improved algorithm is improved. In order to facilitate the generation of FPGA code, the Simulink tool is used to build a model for generating three compensation factors used in the improved algorithm. The output code of Simulink model was loaded into Vivado software. The compensation factor outputted by Vivado is compared with the accurate compensation factor outputted by Matlab. The output generated by Vivado meets the accuracy requirement, which verifies the correctness of the built Simulink model.

Keywords: Synthetic aperture radar; Real-time imaging; Chirp scaling algorithm; Matlab/Simulink; FPGA

Citation: SONG Xuchao, ZHOU Peng, ZHANG Zhenhua, et al. An improved chirp scaling algorithm featured by invariance of compensation factors in a region and its modeling by Simulink [J]. Journal of Telemetry, Tracking and Command, 2023, 44(1): 99-110.

基金项目: 国家自然科学基金重点项目 (61931025); 国家自然科学基金项目 (61971455); 山东省自然科学基金项目 (ZR2019MF004)

通信作者: 周鹏 (zhoupeng@upc.edu.cn)

收稿日期: 2022-05-30 修回日期: 2022-09-12

引言

传统实时 SAR (Synthetic Aperture Radar, 合成孔径雷达) 成像系统架构大都采用 FPGA (Field- Programmable Gate Array, 现场可编程门阵列) 与 DSP (Digital Signal Processor, 数字信号处理器) 相结合的架构^[1-3]。其中, FPGA 主要用于实现数据采集和预处理, DSP 主要用于实现 SAR 成像算法^[4]。近年来, 随着 FPGA 器件的快速发展, 以及其在星上特殊环境下具有优于 DSP 的抗单粒子翻转能力, 采用 FPGA 逐步替代 DSP 来实现 SAR 成像算法, 成为星上 SAR 实时成像系统的重要研究方向之一^[5-7]。

然而, 算法设计人员进行 FPGA 程序开发时常会受到以下限制^[8-11]: ① 算法设计人员通常不熟悉硬件描述语言, 导致开发周期长; ② FPGA 的并行程序设计与基于 Matlab 或 DSP 的串行程序有很大的差异。而利用 Simulink 工具实现 FPGA 快速开发的方式则较好地解决了上述问题^[12]。算法设计人员利用 Matlab 验证算法的正确性后, 可利用 Simulink 搭建算法模型并自动转换为 FPGA 代码, 大大降低了 FPGA 开发的难度、提高了开发效率。另外, 当算法需要改进时, 只需要在 Simulink 模型对应的模块上进行修改, 通过自动生成硬件语言即可实现算法的改进, 灵活性较强。

CS 算法是合成孔径雷达实时成像处理中常采用的一种算法。在经典 CS 算法的处理过程中, 涉及到相位补偿、距离向补偿和方位向补偿这三种补偿因子的计算。这些补偿因子涉及到正余弦和开根号等超越运算, 采用 FPGA 实现这些计算需要较长的运算时间和很大的存储空间。若对于数据矩阵中的每条数据都要更新补偿因子, 则严重影响 SAR 成像处理的实时性。针对上述问题, 文献[13]提出一种具有补偿因子区域不变特点的 CS 算法, 其特点是区域内的各条数据采用固定的补偿因子。该方法在少量损失了三种补偿因子精度的情况下大幅减少了计算量, 提高了处理效率。但在该方法中, 区域内的统一补偿因子采用的是区域内第一条数据对应的补偿因子, 会造成相位补偿误差不均匀的问题。

针对上述问题, 本文对文献[13]中的方法进行改进, 将区域内的统一补偿因子改用该区域内所有单元频率平均值对应的补偿因子进行代替,

使得采用统一补偿因子造成的相位误差在区域内更加均匀。显然, 改进算法与原有算法的计算量相当。但由于改进后算法对应的相位误差在区域内更加均匀, 使得改进算法的成像质量得到提升。为此, 首先开展了 Matlab 仿真工作, 验证了改进后算法相比于原有算法在成像指标方面的提升。接下来, 搭建了用于生成改进算法中三种补偿因子的 Simulink 模型。之后, 将 Simulink 模型输出的代码加载到 Vivado 软件中, 通过对比 Vivado 软件输出的补偿因子和 Matlab 软件输出的精确补偿因子, 验证了所搭建 Simulink 模型的高精度。

1 补偿因子区域不变 CS 算法的改进及 Matlab 验证

1.1 经典 CS 算法原理

雷达发射信号一般为 LFM (Linear Frequency Modulated, 线性调频) 信号。对去载频后的点目标回波信号进行方位向 FFT (Fast Fourier Transform, 快速傅里叶变换), 可得信号在距离多普勒域的表达式为^[14]:

$$S(f_n, \tau) = A \times \exp \left(-j\pi K_s(f_n, \tau) \left[\tau - \frac{2R(f_n, \tau)}{c} \right]^2 \right) \times \exp \left(-j \frac{4\pi}{\lambda} r \sqrt{1 - \left(\frac{\lambda f_n}{2v} \right)^2} \right) \quad (1)$$

$$R(f_n, \tau) = \frac{r}{\sqrt{1 - \left(\frac{\lambda f_n}{2v} \right)^2}} = r[1 + C_s(f)] \quad (2)$$

$$C_s(f_n) = \frac{1}{\sqrt{1 - \left(\frac{\lambda f_n}{2v} \right)^2}} - 1 \quad (3)$$

式中, τ 表示距离向快时间, f_n 表示方位向频率, r 表示雷达与点目标的距离, A 表示调幅项, 包括目标后向散射系数、距离向天线方向图和方位向天线方向图等因素的影响, K_s 表示距离信号多普勒域调频率因子, c 表示光速, λ 表示雷达发射脉冲信号波长, v 表示平台飞行速度, C_s 表示距离弯曲因子。

为使在不同距离上的移动轨迹都与在参考距离上的移动轨迹一致, 需对式(1)乘以如下 CS 相位补偿因子:

$$\phi_1(f_\eta, \tau, r_{\text{ref}}) = \exp\left(-j\pi K_s(f_\eta, r_{\text{ref}}) \times C_s(f_\eta) \times \left[\frac{\tau - 2R(f_\eta, r_{\text{ref}})}{c}\right]^2\right) \quad (4)$$

式中, r_{ref} 表示参考距离。

对乘以相位补偿因子后的信号进行距离向 FFT, 将信号转换至二维频域, 然后将信号二维频域表达式乘以距离向补偿因子, 完成距离压缩、二次距离压缩和距离徙动校正。距离向补偿因子的表达式为:

$$\phi_2(f_\eta, f_r; r_{\text{ref}}) = \exp\left(-j\pi \frac{f_r^2}{K_s(f_\eta, r_{\text{ref}})[1 + C_s(f_\eta)]}\right) \times \exp\left(j \frac{4\pi r_{\text{ref}}}{c} f_r C_s(f_\eta)\right) \quad (5)$$

式中, f_r 表示距离向频率。

将二维频域信号与式(5)相乘后进行距离向 IFFT, 再乘以方位向补偿因子后完成方位压缩和残余相位补偿。方位向补偿因子的表达式为:

$$\phi_3(f_\eta, \tau) = \exp\left(j \frac{4\pi}{\lambda} r \sqrt{1 - \left(\frac{\lambda f_\eta}{2v}\right)^2}\right) \times \exp\left(j\theta_\Delta(f_\eta, r)\right) \times \exp\left(-j \frac{2\pi}{\lambda} c\tau\right) \quad (6)$$

其中,

$$\theta_\Delta = \frac{4\pi}{c^2} K_s(f_\eta, r_{\text{ref}})[1 + C_s(f_\eta)] C_s(f_\eta) (r - r_{\text{ref}})^2 \quad (7)$$

再将乘以方位向补偿因子后所得的信号进行方位向 IFFT (Inverse Fast Fourier Transform, 快速傅里叶逆变换), 即可得到目标图像。

为方便在 FPGA 上实时生成三种相位因子, 需要对 f_η 、 f_r 进行量化处理。将 f_η 等分为 N_η 份, 将 f_r 等分为 N_r 份 (N_η 和 N_r 分别表示方位向和距离向点数), 即

$$f_\eta = f_{\text{PRF}} \times \left(-\frac{1}{2} + \frac{n_\eta}{N_\eta}\right), n_\eta = 0, 1, 2, \dots, N_\eta - 1 \quad (8)$$

$$f_r = f_s \times \left(-\frac{1}{2} + \frac{n_r}{N_r}\right), n_r = 0, 1, 2, \dots, N_r - 1 \quad (9)$$

式中, f_s 表示雷达信号采样频率, f_{PRF} 表示脉冲重复频率。

CS 算法的成像质量与式(4)、式(5)和式(6)表示的三种补偿因子的精度有直接关系, 因此在实际

应用中, 补偿因子计算模块的精度非常重要^[15]。

1.2 补偿因子区域不变 CS 算法的原理

CS 算法成像过程中含有三种补偿因子的计算, 且每种补偿因子都需要与对应的数据矩阵进行复乘。同时 CS 成像算法中补偿因子的计算比较复杂, 包括正余弦和开根号等多种超越运算, 采用 FPGA 实现需要较长的运算时间和很大的存储空间, 对 SAR 成像处理系统的实时性产生较大影响。

针对上述问题, 文献[13]中提出一种具有补偿因子区域不变特点的 CS 算法, 即选定一个更新步长 $\Delta n = 2^n$ (n 为正整数), 每个更新步长内只计算第一个距离向或方位向的补偿因子, 将后几个距离向或方位向的补偿因子用其替换。该文中指出: Δn 越大, 在区域内采用统一的补偿因子带来的误差也越大; 当 Δn 不超过 16 时, 三种误差均不超过 6×10^{-5} 。该算法使得三种补偿因子的计算量大幅降低, 从而有效减少补偿因子的计算时间及其占用的存储器资源。该算法的原理如图 1 所示。

对于补偿因子区域不变 CS 算法, 在一个更新步长内, 其统一补偿因子满足如下特点:

$$\phi_1(n + \Delta n - 1), \phi_1(n + \Delta n - 2) \dots, \phi_1(n + 1) = \phi_1(n) \quad (10)$$

$$\phi_2(n + \Delta n - 1), \phi_2(n + \Delta n - 2) \dots, \phi_2(n + 1) = \phi_2(n) \quad (11)$$

$$\phi_3(n + \Delta n - 1), \phi_3(n + \Delta n - 2) \dots, \phi_3(n + 1) = \phi_3(n) \quad (12)$$

1.3 改进的补偿因子区域不变 CS 算法

在 1.2 节所述的现有算法中, 对于更新步长内各单元的统一补偿因子, 使用的是第一个单元对应的补偿因子。在该算法中, 对于更新步长内首行/列的相位误差为 0, 但后续每个距离向和方位向单元对应的误差会逐渐增加, 导致补偿后的相位误差在区域内分布不均匀。

针对上述问题, 本文对文献[13]中的算法进行了改进, 将一个更新步长内的统一补偿因子, 采用区域内所有单元频率平均值对应的补偿因子进行代替。相比于改进前的算法, 在更新步长对应的区域内, 中间单元对应的误差最小, 第一个和最后一个单元对应的误差最大, 补偿结果的相位误差在区域内更加均匀。

对于改进后的补偿因子区域不变 CS 算法, 在一个更新步长内, 其统一补偿因子对应的频率值满足如下特点:

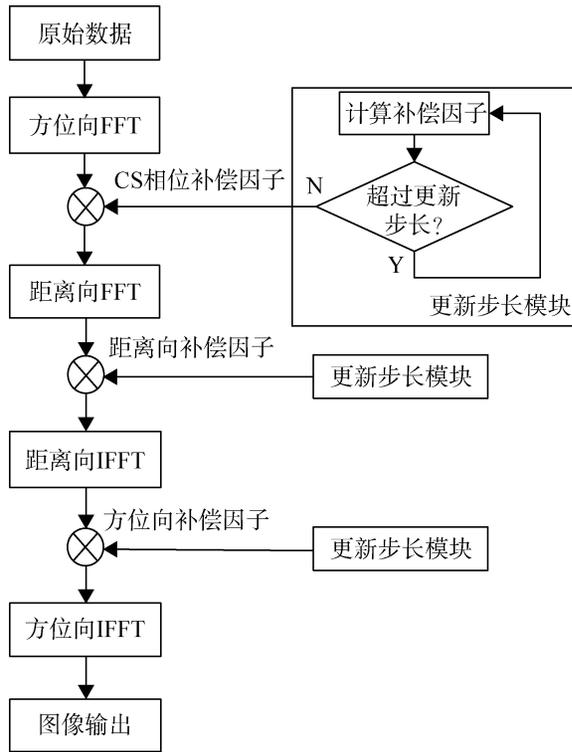


图1 补偿因子区域不变CS算法的处理流程图

Fig. 1 Flow chart of a chirp scaling algorithm featured by invariance of compensation factors in a region

$$f_n(n+m) = \frac{f_n(n+\Delta n-1) + f_n(n+\Delta n-2) + \dots + f_n(n)}{\Delta n} = f_{\text{PRF}} \times \left(-\frac{1}{2} + \frac{2n+\Delta n-1}{2N_n} \right), m=0, 1, 2, \dots, \Delta n-1 \quad (13)$$

$$f_r(n+m) = \frac{f_r(n+\Delta n-1) + f_r(n+\Delta n-2) + \dots + f_r(n)}{\Delta n} = f_s \times \left(-\frac{1}{2} + \frac{2n+\Delta n-1}{2N_r} \right), m=0, 1, 2, \dots, \Delta n-1 \quad (14)$$

在FPGA实现CS成像算法的过程中, 会频繁地从存储器中读取提前存储好的补偿因子, 影响处理效率。经典CS算法在进行补偿因子的复乘操作时, 每进行一次复乘都需要给存储器发送地址来读取该地址下的数据, 而改进后的补偿因子区域不变CS算法只需要每隔一个更新步长取出一次数据即可, 大大降低了进行复乘操作时的存储器读写频率, 同时节约了存储器资源, 相比于经典CS算法更易于FPGA实现。在实际应用时, 只要在成像之前完成所有补偿因子的计算并存入存储

器中即可, 不会占用成像时间。改进算法相比于文献[13]中的算法, 计算量相当。由于最终的补偿结果相比于文献[7]中算法的补偿结果更加均匀, 成像质量将有一定提升。下面对改进算法中3种补偿因子的相位误差进行量化分析。

1.4 改进算法中3种补偿因子的相位误差分析

式(4)~式(6)中的 K_s 和 C_s 仅随方位向频率变化, 而雷达有效速度 v 和方位向频率 f_n 在距离向不发生变化, 因此 v 和方位向频率 f_n 不受更新步长的影响; 而 τ 和 r 是随着距离单元变化的, 即:

$$\tau = \frac{2R_0}{c} + \frac{n-1}{f_s} \quad (15)$$

$$r = \frac{c\tau}{2} = R_0 + \frac{c(n-1)}{2f_s} \quad (16)$$

式中, R_0 表示最近斜距。在一个更新步长内, 相位因子的最大相位误差为:

$$\Delta\varphi = \varphi(n) - \varphi\left(n + \frac{2\Delta n-1}{2}\right) \quad (17)$$

下面对三种补偿因子的相位误差进行量化分析。

1.4.1 CS相位补偿因子的相位误差

由式(4)、式(15)可得, 第 n 个距离单元内的CS相位补偿因子的相位可以表示为:

$$\varphi_1(n) = -\pi K_s C_s \left[\frac{2R_0}{c} + \frac{n-1}{f_s} - \frac{2r(1+C_s)}{c} \right]^2 \quad (18)$$

在一个更新步长内, 对应的最大相位误差 $\Delta\varphi_1$ 为:

$$\Delta\varphi_1 = \pi K_s C_s \frac{2\Delta n-1}{2f_s} \left[\frac{4R_0-4r(1+C_s)}{c} + \frac{2n+\Delta n-\frac{5}{2}}{f_s} \right] \quad (19)$$

经推导后可得, 原始算法对应的最大相位误差 $\Delta\varphi'_1$ 为:

$$\Delta\varphi'_1 = \pi K_s C_s \frac{\Delta n}{f_s} \left[\frac{4R_0-4r(1+C_s)}{c} + \frac{2n+\Delta n-2}{f_s} \right] \quad (20)$$

下面通过仿真, 对两种算法的最大相位误差进行比较。仿真参数设置为: 雷达发射信号频率为 5.3×10^9 Hz, 场景中心斜距为2000 m, 雷达有效速度为150 m/s, 距离向调频率为 2×10^{13} Hz/s, 距离向采样率为 6×10^7 Hz, 方位向采样率为200 Hz, 距离向采样点数为640, 方位向采样点数为2048。两种算法最大相位误差随更新步长变化

的仿真结果见表1。可以看出,改进算法的最大相位误差小于原始算法的最大相位误差。

表1 CS相位因子最大相位误差随更新步长的变化

Table 1 Variation of maximum phase error of chirp scaling phase factor with the update step size

更新步长	2	4	8	16
改进算法的最大相位误差(rad)	$3.109\ 0 \times 10^{-4}$	$9.477\ 3 \times 10^{-4}$	0.002 2	0.004 7
原始算法的最大相位误差(rad)	$6.218\ 1 \times 10^{-4}$	0.001 9	0.004 4	0.009 4

1.4.2 距离向补偿因子的相位误差

由式(5)可得,第 n 个方位单元内的距离向补偿因子的相位可以表示为:

$$\varphi_2(n) = \frac{4\pi r_{\text{ref}}}{c} f_{\tau} C_s(n) \frac{\pi f_{\tau}^2}{K_s(n)[1+C_s(n)]} \quad (21)$$

在一个更新步长内,对应的最大相位误差 $\Delta\varphi_2$ 为:

$$\Delta\varphi_2 = \frac{4\pi r_{\text{ref}}}{c} f_{\tau} \left[C_s(n) - C_s\left(n + \Delta n - \frac{1}{2}\right) \right] - \left\{ \frac{\pi f_{\tau}^2}{K_s(n)[1+C_s(n)]} - \frac{\pi f_{\tau}^2}{K_s\left(n + \Delta n - \frac{1}{2}\right)[1+C_s\left(n + \Delta n - \frac{1}{2}\right)]} \right\} \quad (22)$$

经推导后可得,原始算法对应的最大相位误差 $\Delta\varphi'_2$ 为:

$$\Delta\varphi'_2 = \frac{4\pi r_{\text{ref}}}{c} f_{\tau} [C_s(n) - C_s(n + \Delta n)] - \left\{ \frac{\pi f_{\tau}^2}{K_s(n)[1+C_s(n)]} - \frac{\pi f_{\tau}^2}{K_s(n + \Delta n)[1+C_s(n + \Delta n)]} \right\} \quad (23)$$

利用与前文相同的参数进行了仿真,两种算法最大相位误差随更新步长变化的仿真结果见表2。可以看出,改进算法的最大相位误差小于原始算法的最大相位误差。

1.4.3 方位向补偿因子的相位误差

由式(6)、式(16)可得,第 n 个距离单元内的方位向补偿因子的相位可以表示为:

表2 距离向补偿因子最大相位误差随更新步长的变化

Table 2 Variation of the maximum phase error of the range compensation factor with the update step size

更新步长	2	4	8	16
改进算法的最大相位误差(rad)	0.004	40.013	20.030	90.066
原始算法的最大相位误差(rad)	0.008	80.026	50.061	60.131

$$\varphi_3(n) = \frac{4\pi}{c^2} K_s C_s (1 + C_s) (r - r_{\text{ref}})^2 - \frac{2\pi}{\lambda} c \tau \left[1 - \sqrt{1 - \left(\frac{\lambda f_{\eta}}{2v} \right)^2} \right] \quad (24)$$

$$\text{设 } \theta_1 = \frac{4\pi}{c^2} K_s C_s (1 + C_s), \quad \theta_2 = \frac{2\pi}{\lambda} c \left[1 - \sqrt{1 - \left(\frac{\lambda f_{\eta}}{2v} \right)^2} \right]$$

则在一个更新步长内,最大相位误差 $\Delta\varphi_3$ 可以表示为:

$$\Delta\varphi_3 = \theta_1 \frac{c(1-2\Delta n)}{4f_s} \left[2R_0 + \frac{c(2n+\Delta n - \frac{5}{2})}{2f_s} - 2r_{\text{ref}} \right] - \theta_2 \frac{1-2\Delta n}{2f_s} \quad (25)$$

经推导后可得,原始算法对应的最大相位误差 $\Delta\varphi'_3$ 为:

$$\Delta\varphi'_3 = -\theta_1 \frac{c\Delta n}{2f_s} \left[2R_0 + \frac{c(2n+\Delta n - 2)}{2f_s} - 2r_{\text{ref}} \right] + \theta_2 \frac{\Delta n}{f_s} \quad (26)$$

利用与前文相同的参数进行了仿真,两种算法最大相位误差随更新步长变化的仿真结果见表3。可以看出,改进算法的最大相位误差小于原始算法的最大相位误差。

表3 方位向因子最大相位误差随更新步长的变化

Table 3 Variation of maximum phase error of the azimuth compensation factor with the update step size

更新步长	2	4	8	16
改进算法的最大相位误差(rad)	$3.163\ 4 \times 10^{-4}$	$9.314\ 9 \times 10^{-4}$	0.002 2	0.004 6
原始算法的最大相位误差(rad)	$6.326\ 8 \times 10^{-4}$	0.001 9	0.004 3	0.009 3

1.5 三种CS算法的Matlab验证

对经典CS算法、补偿因子区域不变CS算法和

改进的补偿因子区域不变CS算法，采用 Matlab 语言编制程序，分别对点目标仿真回波和实测数据进行了成像处理。其中，对点目标的成像指标采用距离向和方位向峰值旁瓣比、积分旁瓣比和分辨率进行比较，对实测数据的成像结果采用均方根误差进行比较。由于FPGA本身进行浮点数运算会存在误差，更新步长的值不宜偏大，因此本文中假设更新步长等于4。

1.5.1 点目标仿真成像

首先，以更新步长等于4的情况为例，分别采用经典CS算法、补偿因子区域不变CS算法以及改进的补偿因子区域不变CS算法，对点目标仿真回波数据进行了成像处理，得到详细的成像结果及

成像指标。在处理原始数据时，在方位向加矩形窗以对天线双程方向图的数据范围进行限制。仿真参数设置如下：发射信号脉宽为 2.5×10^{-6} s，雷达发射信号频率为 5.3×10^9 Hz，场景中心斜距为2 000 m，雷达有效速度为150 m/s，距离向调频率为 2×10^{13} Hz/s，距离向采样率为 6×10^7 Hz，方位向采样率为200 Hz，距离向采样点数为640，方位向采样点数为2 048，波束方向为正侧视。成像过程中三种算法均未进行加窗处理。图2、图3和图4分别给出了三种算法的点目标成像结果、距离向剖面图和方位向剖面图。观察图2、图3和图4中的剖面图与成像结果图可以看出，三种算法对点目标的成像质量都很高。

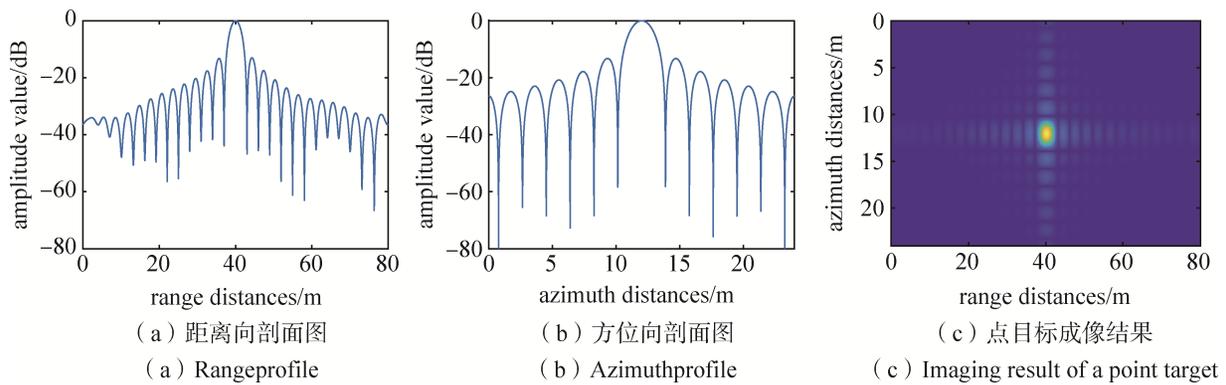


图2 经典CS算法的点目标成像结果

Fig. 2 Imaging results of a point target by the traditional CS algorithm

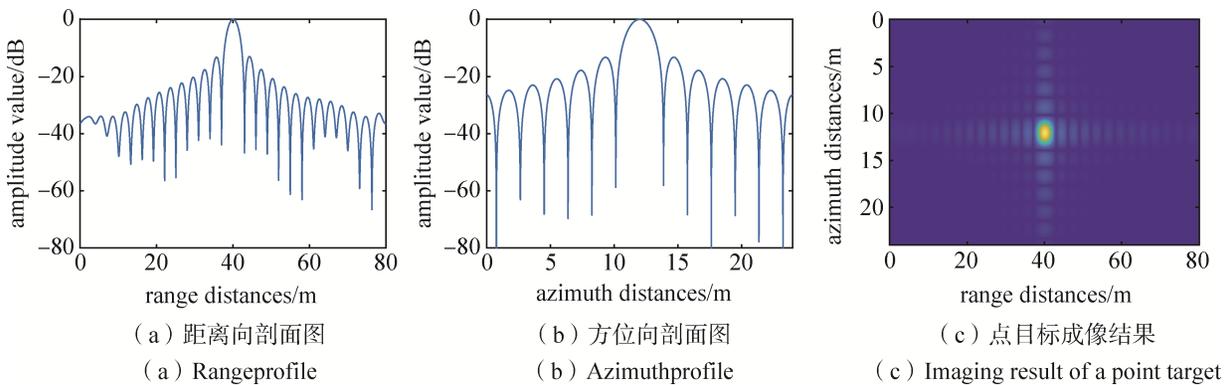


图3 补偿因子区域不变CS算法的点目标成像结果

Fig. 3 Imaging results of a point target by the CS algorithm featured by invariance of compensation factors in a region

以便于对三种算法的性能进行定量比较，分别计算了距离向和方位向的峰值旁瓣比、积分旁瓣比和分辨率，列于表4和表5。

为了比较改进前、后两种算法在更新步长较大情况下的成像质量，分别采用改进前、后两种

补偿因子区域不变CS算法，在更新步长等于8、16的情况下对点目标进行了成像处理，成像结果如图5和图6所示，成像指标列于表6和表7。仿真参数设置与更新步长等于4的情况一致。

对比图2、图3、图4和表4、表5中数据可发

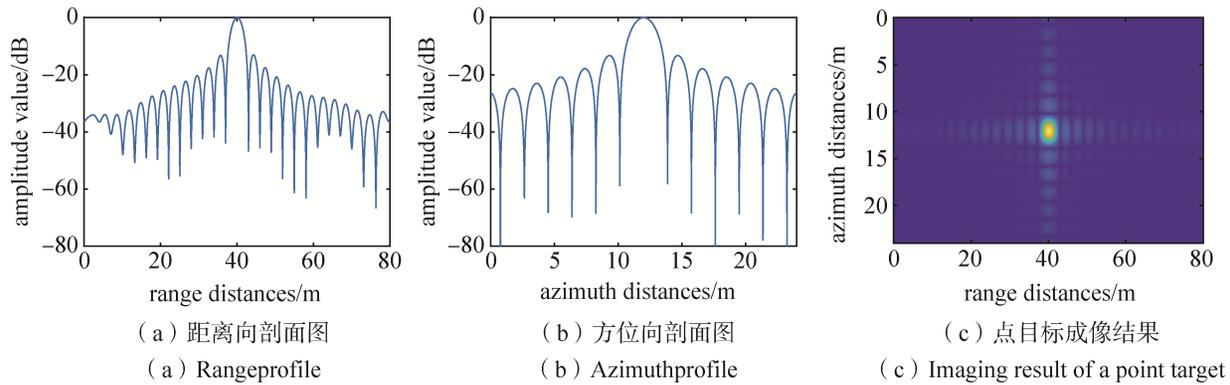


图 4 改进的补偿因子区域不变 CS 算法的点目标成像结果

Fig. 4 Imaging results of a point target by the improved CS algorithm featured by invariance of compensation factors in a region

表 4 点目标仿真结果的距离向成像性能指标

Table 4 Imaging performance indexes of a point target's result in range direction

算法名称	峰值旁瓣比(dB)	积分旁瓣比(dB)	距离向分辨率(m)
经典 CS 算法	-13.858 4	-10.888 6	2.667 5
补偿因子区域不变 CS 算法	-13.241 6	-10.333 1	2.693 9
改进的补偿因子区域不变 CS 算法	-13.265 3	-10.351 7	2.692 9

表 5 点目标仿真结果的方位向成像性能指标

Table 5 Imaging performance indexes of a point target's result in azimuth direction

算法名称	峰值旁瓣比(dB)	积分旁瓣比(dB)	方位向分辨率(m)
经典 CS 算法	-13.775 4	-11.489 7	1.663 0
补偿因子区域不变 CS 算法	-13.554 9	-10.827 1	1.674 5
改进的补偿因子区域不变 CS 算法	-13.563 2	-10.836 0	1.674 1

现：三种算法的成像质量较为接近。其中，经典 CS 算法的成像指标最佳。但需说明的是，尽管经典 CS 算法的指标最佳，但与另两种算法的指标相比提升有限，而其付出的代价是运算复杂度高。相比于补偿因子区域不变 CS 算法，改进后算法的成像性能指标有提升。

对比表 6、表 7 中数据可以看出，随着更新步长的增大，改进后算法的成像性能指标相比改进前的指标，其提升会更加明显，进一步体现了在数据量大、硬件资源有限进而需要选取更大的更新步长的情况下，改进后算法相比于原始算法有更大优越性。

1.5.2 实测数据成像

实测数据采用一组来源于 RADARSAT-1 卫星的某海岸线周边区域的回波数据，令更新步长等于 4，得到经典 CS 算法、补偿因子区域不变 CS 算法和改进的补偿因子区域不变 CS 算法的成像结果

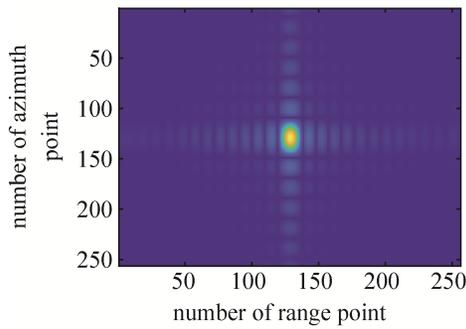
如图 7 所示。

以经典 CS 算法的成像结果为基准值，分别以补偿因子区域不变 CS 算法和改进的补偿因子区域不变 CS 算法的成像结果作为对比值，计算出两种算法的均方根误差指标。从表 8 数据可以看出，相比于改进前算法，改进后算法的成像性能指标得到提升。

2 改进算法中补偿因子计算模块的结构设计和 Simulink 模型

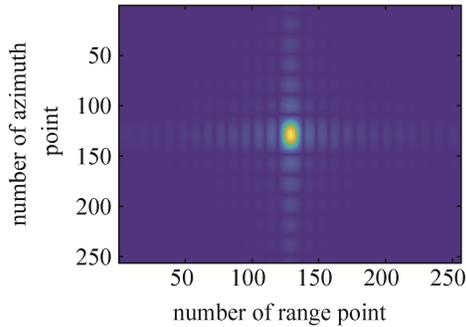
2.1 补偿因子计算模块的结构设计

本文采用乒乓工作机制对改进算法中用于计算补偿因子的模块进行了设计，其信号实时处理能力是单个补偿因子模块的两倍。当有数据输入时，该结构采用了乒乓处理原理，将数据分别输入通道 1 和通道 2 中，在输出端对处理结果进行乒乓接收。图 8 和图 9 分别给出了所设计的补偿因子



(a) 补偿因子区域不变CS算法点目标成像结果

(a) Imaging results of a point target by the CS algorithm featured by invariance of compensation factors in a region



(b) 改进的补偿因子区域不变CS算法点目标成像结果

(b) Imaging results of a point target by the improved CS algorithm featured by invariance of compensation factors in a region

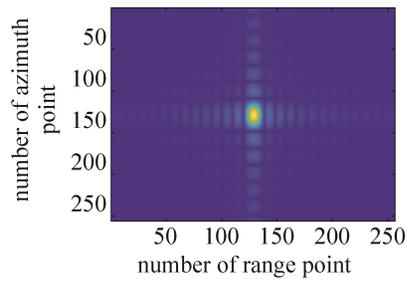
图 5 改进前后两种补偿因子区域不变CS算法的点目标成像结果($\Delta n=8$)

Fig. 5 Imaging results of point targets with original and improved compensation factor region invariant CS algorithm($\Delta n=8$)

计算模块的乒乓结构示意图和时序图。

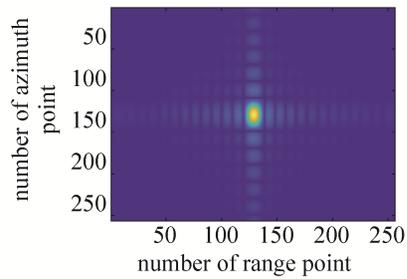
2.2 补偿因子计算模块的 Simulink 模型搭建

所搭建的三种补偿因子计算模块的 Simulink 模型分别如图 10(a)~图 10(c)所示。为了节约硬件资源, Simulink 模型中所有输入、输出和中间变量的数据格式均采用 32 bit 的单精度浮点数, 数据格式



(a) 补偿因子区域不变CS算法点目标成像结果

(a) Imaging results of a point target by the CS algorithm featured by invariance of compensation factors in a region



(b) 改进的补偿因子区域不变CS算法点目标成像结果

(b) Imaging results of a point target by the improved CS algorithm featured by invariance of compensation factors in a region

图 6 改进前后两种补偿因子区域不变CS算法的点目标成像结果($\Delta n=16$)

Fig. 6 Imaging results of point targets with original and improved compensation factor region invariant CS algorithm($\Delta n=16$)

中有 7 位有效数字。

需注意的是, 在补偿因子的计算过程中, 某几个中间变量相乘后的值会非常大, 使得整数部分占有较多的有效数字个数, 从而导致产生较大的计算误差。因此, 为减小上述误差来源, 需对模块中的部分组成进行专门的设计。下面以方位向补偿因子计算模块的部分组成为例进行说明。

表 6 点目标仿真结果的成像性能指标($\Delta n=8$)

Table 6 Imaging performance performance of a point target's result($\Delta n=8$)

算法名称	峰值旁瓣比(dB)		积分旁瓣比(dB)		分辨率(m)	
	距离向	方位向	距离向	方位向	距离向	方位向
补偿因子区域不变CS算法	-13.235 8	-13.535 0	-10.318 4	-10.811 4	2.695 5	1.675 3
改进的补偿因子区域不变CS算法	-13.262 4	-13.555 7	-10.346 2	-10.831 6	2.674 2	1.674 4

表7 点目标仿真结果的成像性能指标($\Delta n=16$)
Table 7 Imaging performance of a point target's result($\Delta n=16$)

算法名称	峰值旁瓣比(dB)		积分旁瓣比(dB)		分辨率(m)	
	距离向	方位向	距离向	方位向	距离向	方位向
补偿因子区域不变CS算法	-13.220 6	-13.511 2	-10.304 7	-10.792 3	2.696 4	1.676 2
改进的补偿因子区域不变CS算法	-13.261 2	-13.550 8	-10.344 1	-10.828 2	2.694 7	1.674 8

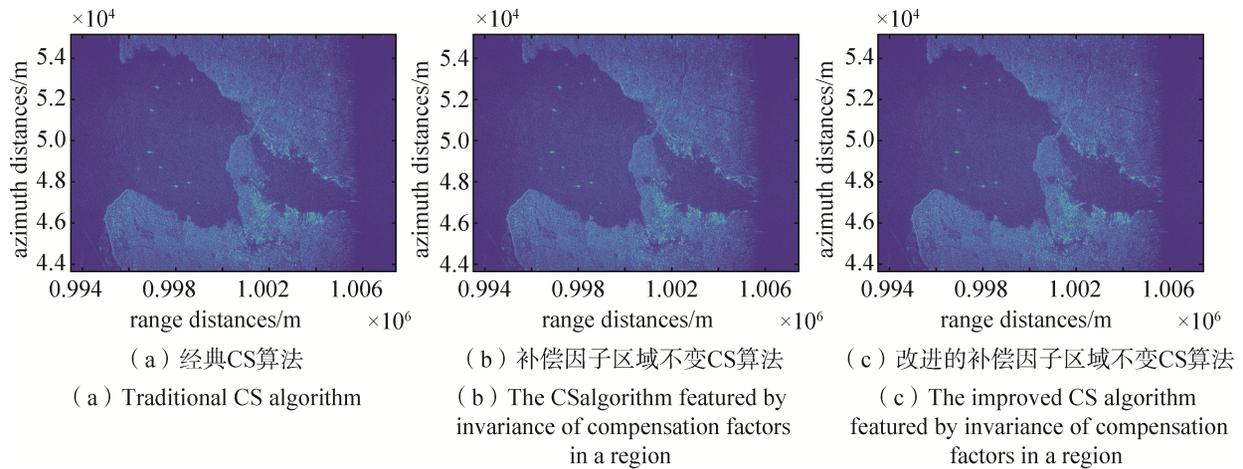


图7 3种算法对RADARSAT-1原始数据的成像结果

Fig.7 Imaging results of three algorithms on raw data of RADARSAT-1

表8 补偿因子区域不变CS算法和改进的补偿因子区域不变CS算法的均方根误差

Table 8 Mean root squares of the previous and the improved CS algorithms featured by invariance of compensation factors in a region

算法名称	均方根误差
补偿因子区域不变CS算法	0.548 3
改进的补偿因子区域不变CS算法	0.426 1

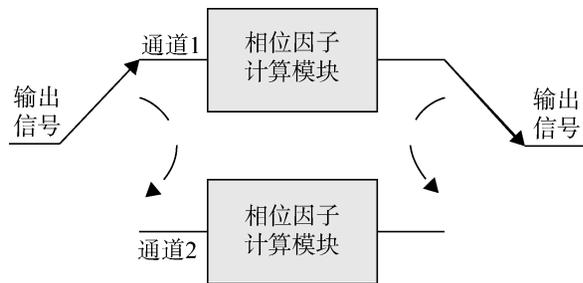


图8 补偿因子计算模块的结构示意图

Fig. 8 Structure diagram of the designed module for calculating compensation factor

在方位向补偿因子计算模块中,先提出了 2π 作为指数项的公因子,使得最终补偿因子的计算结果仅取决于输入数据的小数部分,再对计算过

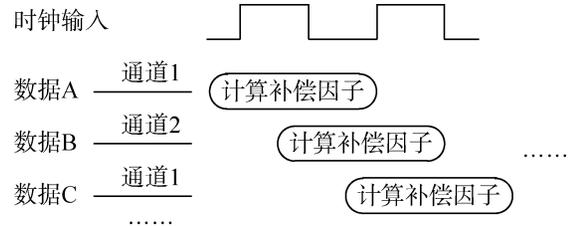


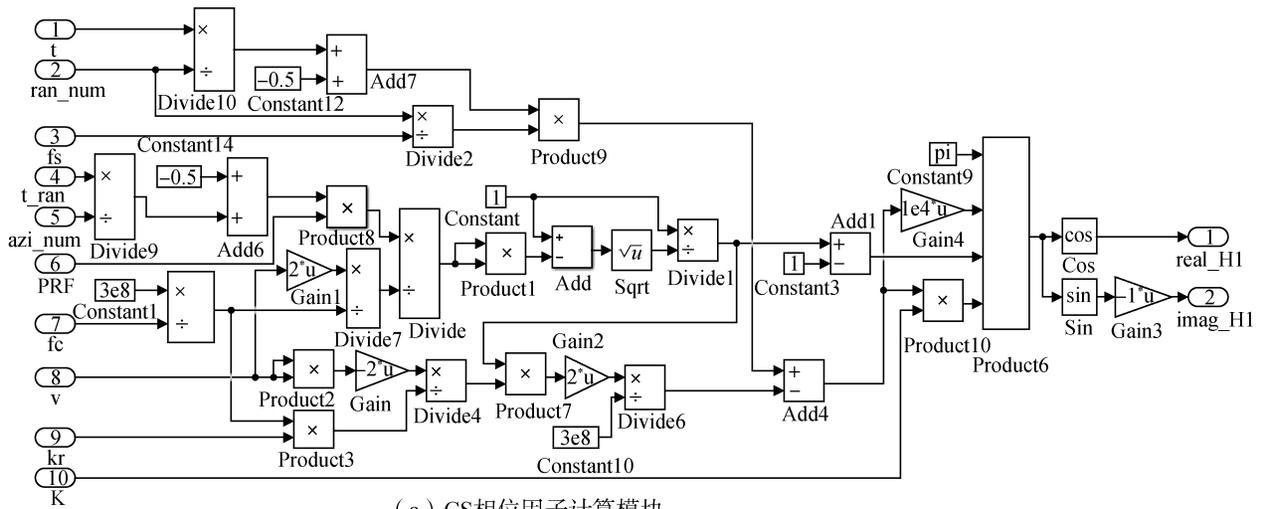
图9 补偿因子计算模块的时序图

Fig. 9 Timing diagram of the designed module for calculating compensation factor

程中两个带有小数的中间变量相乘的项采用如图11所示的子模块代替。图11中,fix表示向0靠拢取整, a 、 b 表示输入的被乘数, c 表示乘数。通过该子模块的专门设计,可使得两个较大的含小数中间变量相乘后,整数部分占有的有效数字个数减少,从而提高最终计算结果的精度。

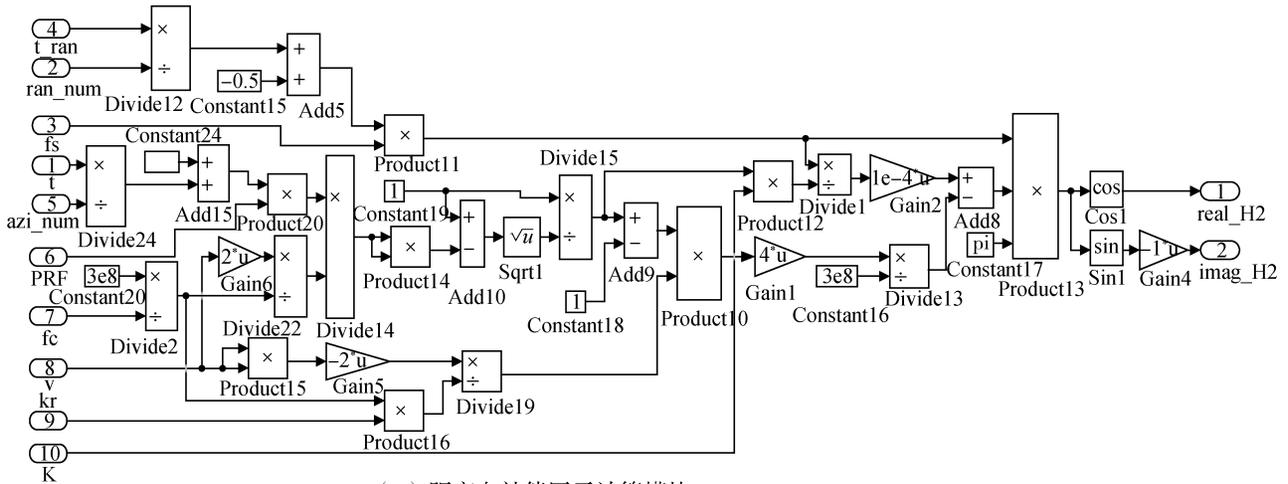
3 实验结果

仿真参数设置如下:脉冲重复频率为1 400 Hz,信号采样频率为 3×10^7 Hz,发射信号中心频率为 10^9 Hz,发射信号距离向调频斜率为 2×10^{12} Hz/s,雷达有效运动速度为1 000 m/s,距离向点数为1 024,方位向点数为1 024,方位向信号的多普勒



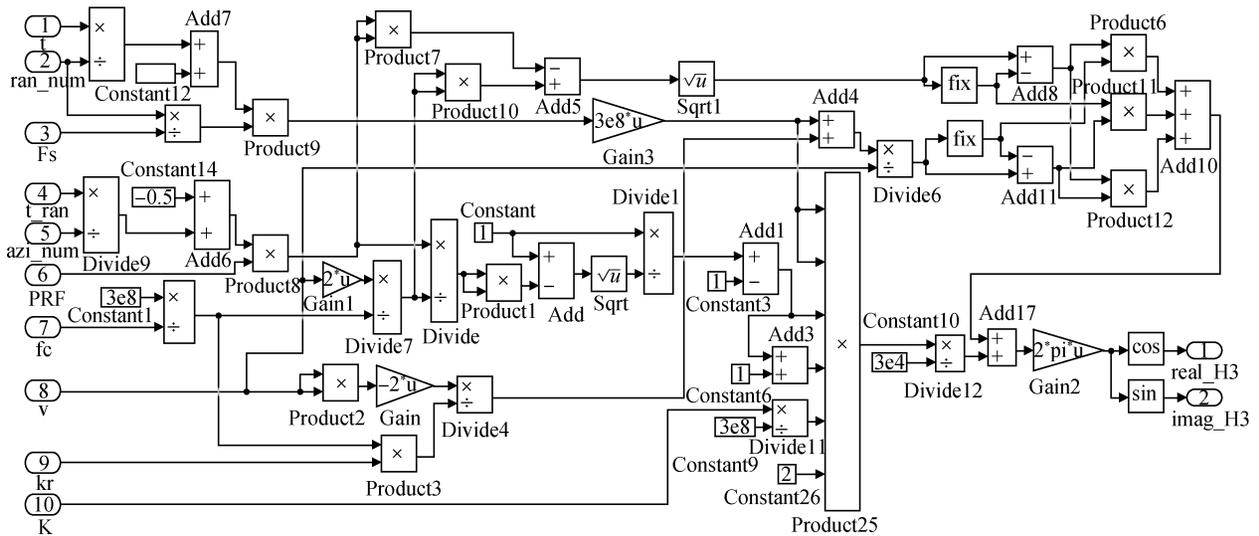
(a) CS相位因子计算模块

(a) Module for calculating phase factor of chirp scaling processing



(b) 距离向补偿因子计算模块

(b) Module for calculating compensation factor in range direction



(c) 方位向补偿因子计算模块

(c) Module for calculating compensation factor in range direction

图 10 三种补偿因子计算模块的 Simulink 模型图

Fig. 10 Three Simulink models for calculating compensation factors

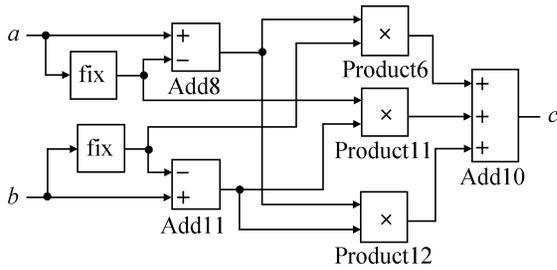


图 11 用于计算小数部分的子模块

Fig. 11 Sub module for calculating decimal part

调频率为-8 357 Hz/s。

根据上述仿真参数，执行如图 8 所示的三个 Simulink 模块并输出 FPGA 代码。将 Simulink 模型

输出的代码加载到 Vivado 软件中进行执行。图 12 给出了 Vivado 软件输出的三种补偿因子的部分结果。

利用 Vivado 软件将补偿因子的计算结果存入 txt 文件，然后由 Matlab 读取并将结果转换为十进制小数存入 .mat 文件中，每个补偿因子的实部和虚部分别构成一个矢量。编写改进的补偿因子区域不变 CS 算法的 Matlab 程序，输出正确的补偿因子仿真结果，每个补偿因子的实部和虚部分别构成一个矢量。将 Vivado 输出的补偿因子与由 Matlab 输出的精确的补偿因子进行相减，图 13 给出了三种补偿因子实部和虚部的误差。

clk_150m	0									
real_H1[31:0]	3f7fa4c	3f7fe40d	3f7fe7e8	3f7feb5a	3f7fee69	3f7ff11e	3f7ff380	3f7ff595	3f7ff765	3f7ff8f6
imag_H1[31:0]	bc580966	bcef3788	bcde1975	bccd9de7	bcdbc4cd	bcae8e3e	bc9ffa43	bc9208e4	bc84ba32	bc701c38
real_H2[31:0]	bcd32fe2	bcab32af	bcafb26d	bcb3b22f	bcb831e6	bcbc719e	bcc0f14e	bcc570fb	bcca309e	bccce7047
imag_H2[31:0]	bf7fea38	bf7ff1af	bf7ff0ec	bf7ff03a	bf7fef6d	bf7feea8	bf7fedd1	bf7fecf6	bf7fec09	bf7feb2f
real_H3[31:0]	3f5f47d2	3e864e28	bf7f9ac9	3ebbd2bb	3f440904	bf59d203	be69aa54	3f7ee65e	becdd862	bf3d9db2
imag_H3[31:0]	befa7434	3f770913	bd638e32	bf6e270a	3f24a3ba	3f068009	bf793ef4	3dbda932	3f6a662a	bf2bfe56

图 12 Vivado 软件的部分仿真结果

Fig. 12 Partial simulation results by Vivado software

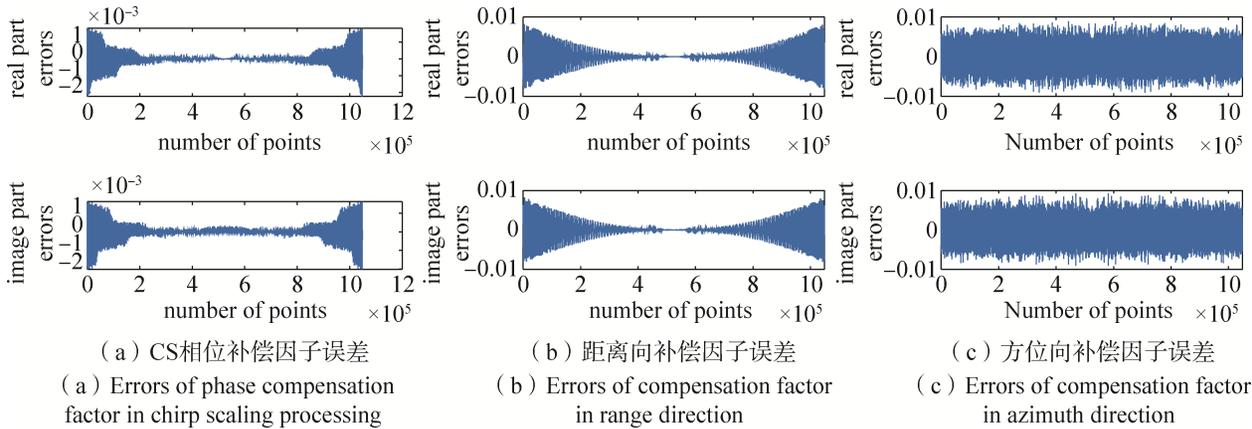


图 13 Vivado 输出结果相比于 Matlab 输出结果的误差

Fig. 13 Error of Vivado output result compared with Matlab output result

从图 13 可以看出，CS 相位补偿因子的误差在 $\pm(2 \times 10^{-3})$ 以内，距离向和方位向补偿因子的误差在 ± 0.01 以内，误差均较小，验证了本文所搭建 Simulink 模型的准确性。其中，CS 相位补偿因子和距离向补偿因子的误差曲线图具有明显对称性，且越靠近成像场景中心的补偿因子对应的误差越小，体现了本文所提出改进算法的特点。

4 结束语

对一种具有补偿因子区域不变特点的 CS 算法进行了改进，对于更新步长内各单元的统一补偿因子，采用该区域内所有单元频率平均值对应的补偿因子代替原来使用的第一个单元对应的补偿因子，使得采用统一补偿因子造成的相位误差在区域内更加均匀。原算法与改进后的算法计算量

相当。经过Matlab仿真实验验证,改进算法的成像指标得到提升,且更新步长越大,改进算法的性能提升越明显。为提高FPGA代码的生成效率,采用Simulink工具搭建了用于生成改进算法中的三种补偿因子模型。将Simulink模型输出的代码加载到Vivado软件中,通过对比Vivado输出的补偿因子与由Matlab输出的精确补偿因子,验证了所搭建Simulink模型的高精度。本文所研究的利用Simulink建模生成FPGA代码的方法具有较好的通用性,可推广应用至SAR实时成像或其他电子信息领域,提高FPGA代码的生成效率。

参考文献

- [1] 李翱,于勇,褚超,等.基于FPGA的高效脉冲压缩技术工程应用研究[J].遥测遥控,2017,38(2):13-21.
LI Ao, YU Yong, CHU Chao, et al. Engineering application research of efficient pulse compression technology based on FPGA[J]. Journal of Telemetry, Tracking and Command, 2017, 38(2): 13-21.
- [2] 黄太.基于FPGA的视频SAR实时成像处理技术研究[D].成都:电子科技大学,2022.
- [3] 陈皓.基于FPGA+DSP的斜视SAR实时处理技术[D].西安:西安电子科技大学,2012.
- [4] 于健.基于FPGA和DSP的SAR成像信号处理板设计[D].西安:西安电子科技大学,2009.
- [5] LIU Jin, CHEN Liang, LIU Ying, et al. Design of spaceborne SAR imaging processing and fast verification based on FPGA[C]//International Radar Conference. IET, 2013.
- [6] GAO Lining, LONG Teng. Spaceborne digital signal processing system design based on FPGA[C]//2008 Congress on Image and Signal Processing. IEEE, 2008, 577-581.
- [7] 杨涛,赵启钊,孙光才,等.基于多处理板联合SAR成像的分布式计算模拟系统研究[J].海洋科学,2021,45(5):54-61.
YANG Tao, ZHAO Qizhao, SUN Guangcai, et al. Distributed computing simulation system based on a multi-processing board combined with SAR imaging[J]. Marine Sciences, 2021, 45(5): 54-61.
- [8] 张佳琪,文腾飞,李凉海,等.基于Matlab的信号处理模块的FPGA快速开发[J].计算机与网络,2020,46(11):60-63.
ZHANG Jiaqi, WEN Tengfei, LI Lianghai, et al. Rapid development of FPGA for signal processing module based on Matlab[J]. Computer & Network, 2020, 46(11): 60-63.
- [9] 韩增玉.前视SAR成像处理方法及硬件实现[D].成都:电子科技大学,2021.
- [10] 郑云伟.基于FPGA的视频SAR成像处理系统设计与实现[D].成都:电子科技大学,2021.
- [11] 王旭.星载SAR实时成像技术研究[D].西安:西安电子科技大学,2020.
- [12] 张佳琪.小卫星SAR实时成像技术研究[D].北京:中国运载火箭技术研究院,2019.
- [13] 刘小宁.星上SAR实时成像处理关键技术研究[D].北京:北京理工大学,2016.
- [14] 皮亦鸣,杨建宇,付毓生,等.合成孔径雷达成像原理[M].成都:电子科技大学出版社,2007.
- [15] 李志华.基于FPGA的微型SAR实时成像处理研究[D].西安:西安电子科技大学,2014.

[作者简介]

宋栩潮 1998年生,硕士研究生,主要研究方向为雷达信号处理。

周鹏 1976年生,博士,副教授,主要研究方向为SAR成像处理和海洋遥感应用。

张振华 1977年生,博士,研究员,主要研究方向为雷达信号处理和SAR成像。

肖忠源 1984年生,博士,高级工程师,主要研究方向为雷达系统设计和SAR成像。

(本文编辑:傅杰)