

引用格式:王义平,王佳辉,薛雅丽.基于流水线高斯粒子滤波的无人机姿态估计算法及FPGA实现[J].电光与控制,2019,26(2):66-70,75.
WANG Y P, WANG J H, XUE Y L. UAV attitude estimation algorithm and its FPGA implementation based on pipeline Gaussian particle filter[J]. Electronics Optics & Control, 2019, 26(2):66-70, 75.

基于流水线高斯粒子滤波的无人机姿态估计算法及FPGA实现

王义平, 王佳辉, 薛雅丽
(南京航空航天大学, 南京 210016)

摘要:采用高斯粒子滤波算法进行姿态估计算法设计,将四元数离散方程作为状态方程。算法由采样调节粒子、采样粒子、权值计算、均值协方差计算和Cholesky 5个模块组成。通过采用非标准化权值计算四元数“平均”值和协方差阵,并且改写协方差阵计算公式,实现流水线高斯粒子滤波算法。同时提出了并行化设计方案,利用FPGA剩余资源进一步优化运行速率。给出的简化粒子滤波算法与高斯粒子滤波算法设计不仅可用于无人机姿态估计,对于其他非线性估计问题及应用亦适用。仿真结果表明了本设计的可行性和有效性。

关键词:无人机;高斯粒子滤波;姿态估计;FPGA

中图分类号:V279 文献标志码:A doi:10.3969/j.issn.1671-637X.2019.02.014

UAV Attitude Estimation Algorithm and Its FPGA Implementation Based on Pipeline Gaussian Particle Filter

WANG Yi-ping, WANG Jia-hui, XUE Ya-li
(Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: An attitude estimation algorithm is designed by using Gaussian Particle Filtering (GPF) algorithm. The quaternion discrete equation is taken as the state equation. The algorithm consists of five modules: sampling conditioning particle, sampling particle, weight calculation, mean covariance calculation and Cholesky. The “average” value and covariance matrix of the quaternion are calculated by using non-standardized weights, and the calculation formula of the covariance matrix is rewritten. In this way, the pipeline Gaussian particle filter algorithm is implemented. At the same time, a parallel design scheme is proposed, which can further optimize the running speed by using the remaining resources of FPGA. The simplified particle filter algorithm and Gaussian particle filter algorithm presented here can not only be used in UAV attitude estimation, but also be applied to other nonlinear estimation problems and applications. The simulation results have proved the feasibility and effectiveness of this design.

Key words: UAV; Gaussian particle filter; attitude estimation; FPGA

0 引言

高斯粒子滤波算法(GPF)是应用高斯滤波框架的粒子滤波方法,其假设被估计状态量的概率密度函数近似多维高斯分布,利用粒子滤波技术来求解高斯分布的相关参数并得到最终结果。因此,GPF在时间上递推传播的是概率密度的均值和协方差。简单来说,GPF是一类高斯滤波器,其使用基于蒙特卡罗的方法获得相关密度的均值和协方差估计,并且这些估计在

时间上递推更新^[1],GPF用重要性采样方法更新状态的均值和协方差矩阵,免去了重采样步骤,在硬件上运行时,可有效提高算法的运行速率。对于通常的非线性非高斯状态估计问题,通用粒子滤波算法具有较好的估计精度。但对于较为特殊的噪声为高斯的系统,可以采用高斯粒子滤波算法(GPF)提高系统的实时性,由于GPF不需要重采样过程,非常适合并行实现。文献[2]介绍了针对非线性系统的估计问题,GPF算法具有精度高、鲁棒性好的优点,且不受附加高斯噪声的限制,易于超大规模集成电路实现。由此将GPF用于无人机姿态估计,并通过SysGen在FPGA上实现算法,仿真验证GPF算法的可行性。本文给出的高斯粒子滤波算法设计不局限于在无人机姿态估计上的应

收稿日期:2018-03-06

修回日期:2018-03-20

作者简介:王义平(1987—),男,安徽芜湖人,硕士,研究方向为非线性滤波。

用,对于其他非线性估计问题及应用都具有参考价值
和实际意义。

1 基于四元数的姿态速度变换模型

由于四元数具有维度较低、非奇异等优点,无人机
姿态估计通常采用四元数^[3]作为姿态描述参数。四元
数是一个由1个实数单位和3个虚数单位*i, j, k*构成
的含有4个元的数,定义为

$$\mathbf{A} = (\lambda_0, \lambda_1, \lambda_2, \lambda_3) = \lambda_0 + \lambda_1 i + \lambda_2 j + \lambda_3 k \quad (1)$$

本文采用三角形式

$$\mathbf{A} = \cos(\sigma/2) + n \sin(\sigma/2) \quad (2)$$

来描述一个坐标系或一个矢量相对于某一坐标的旋转
关系。设 $\mathbf{r}^g = [0 \ X_g \ Y_g \ Z_g]^T$, $\mathbf{r}^f = [0 \ X_f \ Y_f \ Z_f]^T$, 则
 \mathbf{r}^f 和 \mathbf{r}^g 的转换关系(即四元数转动公式)为

$$\begin{cases} \mathbf{r}^g = \mathbf{A} \circ \mathbf{r}^f \circ \mathbf{A}^{-1} \\ \mathbf{r}^f = \mathbf{A}^{-1} \circ \mathbf{r}^g \circ \mathbf{A} \end{cases} \quad (3)$$

式中, $\mathbf{A} = [\lambda_0 \ \lambda_1 \ \lambda_2 \ \lambda_3]^T = \cos(\sigma/2) + n \sin(\sigma/2)$
为规范化四元数(σ 为两坐标系之间的旋转角度), 因此
 $\mathbf{A}^{-1} = \cos(\sigma/2) - n \sin(\sigma/2)$ 。

下面给出文献[4]介绍的四元数运动学微分方程

$$\mathbf{Q}(\dot{\mathbf{A}}) = \frac{1}{2} \mathbf{M}(\mathbf{A}) \mathbf{Q}(\omega_{gf}^f) = \frac{1}{2} \mathbf{M}^*(\omega_{gf}^f) \mathbf{Q}(\mathbf{A}) \quad (4)$$

式中: $\mathbf{Q}(\cdot)$ 表示四元数形式; $\mathbf{M}(\cdot)$ 表示矩阵形式; $\mathbf{M}^*(\cdot)$
表示矩阵的共轭转置。

通常离散形式能够更简便地应用滤波方法,将上
式离散化可得离散形式

$$\mathbf{Q}(\mathbf{A}_k) - \mathbf{Q}(\mathbf{A}_{k-1}) = \frac{\Delta t}{2} \mathbf{M}^*(\omega_{gf}^f) \mathbf{Q}(\mathbf{A}_{k-1}) \quad (5)$$

故四元数离散状态更新方程可表示为

$$\mathbf{Q}(\mathbf{A}_k) = \left(\mathbf{I}_{4 \times 4} + \frac{\Delta t}{2} \mathbf{M}^*(\omega_{gf}^f) \right) \mathbf{Q}(\mathbf{A}_{k-1}) \quad (6)$$

速度更新由比力方程给出,具体形式为

$$\dot{\mathbf{v}}^c = \mathbf{f}^c + \mathbf{g}^c \quad (7)$$

式中: \mathbf{v}^c 为导航坐标系下的速度向量; \mathbf{f}^c 为捷联加速度
计的输出,经方向余弦矩阵 \mathbf{C}_g^f 转换后得到; \mathbf{g}^c 为当地
重力加速度。同样地,将比力方程改写为离散形式

$$\mathbf{v}_k^c - \mathbf{v}_{k-1}^c = (\mathbf{C}(\mathbf{A}_{k-1})_f^g (\mathbf{a}_{k-1}) + \mathbf{g}^c) \Delta t \quad (8)$$

式中: $\mathbf{C}(\mathbf{A}_{k-1})_f^g$ 为四元数形式的机体到导航系的坐
标转换矩阵; \mathbf{a}_{k-1} 为捷联加速度计输出的比力。

四元数离散运动学方程,将其作为采样策略,且为
了书写简便和易于理解进行适当改写,有

$$\mathbf{q}_k^{(i)} = \left(\mathbf{I}_{4 \times 4} + \frac{\Delta t}{2} \mathbf{\Omega}(\mathbf{f} \omega_{k-1}^{(i)}) \right) \mathbf{q}_{k-1}^{(i)} + w_q^{(i)} \quad (9)$$

$${}^g \mathbf{v}_k^{(i)} = {}^g \mathbf{v}_{k-1}^{(i)} + (\mathbf{C}(\mathbf{q}_{k-1}^{(i)})_f^g (\mathbf{a}_{k-1}) + w_a^{(i)}) + \mathbf{g} \Delta t + w_v^{(i)} \quad (10)$$

式中: $\mathbf{I}_{4 \times 4}$ 为单位矩阵; $\mathbf{C}(\mathbf{q}_{k-1}^{(i)})_f^g$ 为四元数描述的机体
系到地理系的姿态转移矩阵;矩阵 $\mathbf{\Omega}(\mathbf{f} \omega_{k-1}^{(i)})$ 表示为

$$\mathbf{\Omega}(\mathbf{f} \omega_{k-1}^{(i)}) = \begin{bmatrix} 0 & -f \omega_{xk-1}^{(i)} & -f \omega_{yk-1}^{(i)} & -f \omega_{zk-1}^{(i)} \\ f \omega_{xk-1}^{(i)} & 0 & f \omega_{zk-1}^{(i)} & -f \omega_{yk-1}^{(i)} \\ f \omega_{yk-1}^{(i)} & -f \omega_{zk-1}^{(i)} & 0 & f \omega_{xk-1}^{(i)} \\ f \omega_{zk-1}^{(i)} & f \omega_{yk-1}^{(i)} & -f \omega_{xk-1}^{(i)} & 0 \end{bmatrix} \quad (11)$$

也可将其简化表达为

$$\mathbf{\Omega}(\mathbf{f} \omega_{k-1}^{(i)}) = \begin{bmatrix} 0 & -(\mathbf{f} \omega_{k-1}^{(i)})^T \\ \mathbf{f} \omega_{k-1}^{(i)} & -[\mathbf{f} \omega_{k-1}^{(i)}]_{\times} \end{bmatrix} \quad (12)$$

其中: $\mathbf{f} \omega_{k-1}^{(i)} = \mathbf{f} \omega_{k-1} + w_{\omega}^{(i)}$; $[\cdot]_{\times}$ 表示叉乘矩阵。滤波
器输入为机体系下的机体系到地理系的三轴角速度
 $\mathbf{f} \omega_{k-1}$ 和比力 \mathbf{a}_{k-1} , 向量 \mathbf{g} 表示地理系下当地的重力加
速度,由于 UAV 是在有限的运行范围内工作,因而可
以假设为常值。 $w_{\omega}^{(i)}$, $w_q^{(i)}$, $w_a^{(i)}$ 和 $w_v^{(i)}$ 通过提议分布采
样获得,表征预测步骤中的不确定性。预测过程中的
不确定性通常包含两种情况。

1) 滤波器输入的不确定性: $\mathbf{f} \omega_{k-1}$ 和 \mathbf{a}_{k-1} 通常带
有噪声,且噪声通常服从 $w_a^{(i)} \sim N(0, \sigma_a^2)$ 和 $w_{\omega}^{(i)} \sim N(0, \sigma_{\omega}^2)$ 。

2) 数值积分误差和小角度近似误差:式(9)中对
四元数更新方程进行的简化和近似会引入误差,通常
假设这些误差也满足高斯分布,有 $w_q^{(i)} \sim N(0, \sigma_q^2)$ 和
 $w_v^{(i)} \sim N(0, \sigma_v^2)$ 。

2 高斯粒子滤波算法

本节不再给出 GPF 的基本理论,可参考文献[5],
通用高斯粒子滤波算法伪码如下所示。

输入:观测 y_n 和先验估计 μ_{n-1} 和 C_{n-1} 。

开始:基于先验分布获得均值 μ_0 和协方差 Σ_0 。

方法:

GPF-时间更新算法。

1) 从 $N(x_{n-1}; \mu_{n-1}, \Sigma_{n-1})$ 中采样获取调节粒子 $\{x_{n-1}^{(m)}\}_{m=1}^M$ 。

2) 通过 $p(x_n | x_{n-1} = x_{n-1}^{(m)})$ 更新采样粒子 $\{x_n^{(m)}\}_{m=1}^M$ 。

3) ① 通过 $\tilde{w}_n^{(m)} = p(y_n | x_n^{(m)})$ 计算权值;

② 通过 $w_n^{(m)} = \tilde{w}_n^{(m)} / \sum_{m=1}^M \tilde{w}_n^{(m)}$ 标准化权值。

4) 均值和协方差估计算式为

$$\textcircled{1} \mu_n = \sum_{m=1}^M w_n^{(m)} x_n^{(m)};$$

$$\textcircled{2} \Sigma_n = \sum_{m=1}^M w_n^{(m)} (x_n^{(m)} - \mu_n)(x_n^{(m)} - \mu_n)^T。$$

5) 计算 Σ_n 的 Cholesky 分解 $\Sigma_n = C_n C_n^T$ 。

将高斯粒子滤波用于四元数姿态估计,需要解决
四元数的协方差计算问题以及 Cholesky 分解问题。对
于四元数协方差计算可由文献[6]得到,即

$$\Sigma_k = E \{ (\mathbf{q}_k^{(i)} - \mathbf{q}_k) (\mathbf{q}_k^{(i)} - \mathbf{q}_k)^T \} \quad (13)$$

并且文献[7]证明了对于非标准化四元数 $\mathbf{q}_k^{(i)}$, 同
样可以计算协方差阵 Σ_k 而不失一般性。由于真实四

元数 \mathbf{q}_k 无法得到, 此处用单个粒子的四元数状态与估计四元数来表示四元数的协方差矩阵为

$$\Sigma_k = \sum_{k=1}^M w_k^{(m)} ((\mathbf{q}_k^{(i)} - \hat{\mathbf{q}}_k)(\mathbf{q}_k^{(i)} - \hat{\mathbf{q}}_k)^T) \quad (14)$$

对于协方差阵的 Cholesky 分解, 通常利用平方根法求解矩阵的 Cholesky 分解。假设 $\mathbf{A} = \mathbf{L}\mathbf{L}^T$, 则对于 4×4 方阵 \mathbf{A} , 有 Cholesky 分解 \mathbf{L} 形如

$$\mathbf{L} = \begin{bmatrix} \sqrt{a_{11}} & 0 & 0 & 0 \\ a_{21}/l_{11} & \sqrt{a_{22}-l_{21}^2} & 0 & 0 \\ a_{31}/l_{11} & (a_{32}-l_{31}l_{21})/l_{22} & \sqrt{a_{33}-l_{31}^2-l_{32}^2} & 0 \\ a_{41}/l_{11} & (a_{42}-l_{41}l_{21})/l_{22} & (a_{43}-l_{41}l_{31}-l_{42}l_{32})/l_{33} & \sqrt{a_{44}-l_{41}^2-l_{42}^2-l_{43}^2} \end{bmatrix} \quad (15)$$

3 基于高斯粒子滤波的姿态估计算法

3.1 算法设计

文献[6]已经给出了简化粒子滤波算法用于无人机姿态估计的具体算法流程, 并设计了其 FPGA 实现, 验证了有效性和实时性。本节在 2.1 节给出的通用高斯粒子滤波和文献[6]给出的简化粒子滤波姿态估计算法的基础上, 设计了高斯粒子滤波算法, 并且对其实时性进行了分析。具体算法伪码如下。

输入: 粒子 $\{x_{k-1}^{(j)}\}_{j=1}^N$, IMU 输出 $f\boldsymbol{\omega}_{k-1}^{(i)}$ 和 $f\mathbf{a}_{k-1}$, GPS 输出 $g\tilde{\mathbf{v}}_k$,

Σ_{k-1} 的 Cholesky 分解 $\mathbf{C}_{k-1}; \Sigma_{k-1} = \mathbf{C}_{k-1}\mathbf{C}_{k-1}^T$ 。

输出姿态和速度估计 $\hat{\mathbf{q}}_k$ 和 $g\hat{\mathbf{v}}_k$ 。

1) 采样调节粒子: $\mathbf{q}_{k-1}^{(i)} = \hat{\mathbf{q}}_{k-1} + \mathbf{C}_{k-1} w_c^{(i)}$ 。

2) 采样步骤:
$$\begin{cases} \mathbf{q}_k^{(i)} = \left(\mathbf{I}_{4 \times 4} + \frac{\Delta t}{2} \boldsymbol{\Omega}(f\boldsymbol{\omega}_{k-1}^{(i)}) \right) \mathbf{q}_{k-1}^{(i)} + w_q^{(i)} \\ g\mathbf{v}_k^{(i)} = g\mathbf{v}_{k-1}^{(i)} + (\mathbf{C}(\mathbf{q}_{k-1}^{(i)})^{\#}(f\mathbf{a}_{k-1} + w_a^{(i)}) + \mathbf{g})\Delta t + w_v^{(i)} \end{cases}$$
。

3) ① 权值计算: $w_k^{(i)} = \frac{w_{k-1}^{(i)}}{\sigma_{gv}\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\|g\tilde{\mathbf{v}}_k - g\mathbf{v}_k^{(i)}\|}{\sigma_{gv}}\right)^2\right]$ 。

② 权值标准化: $w_k^{(i)} = w_k^{(i)} / W_k$ 。

4) ① 状态估计:
$$\begin{cases} g\hat{\mathbf{v}}_k = \sum_{i=1}^N w_k^{(i)} g\mathbf{v}_k^{(i)} \\ \mathbf{q}_k = \frac{1}{\sum_{i=1}^N w_k^{(i)} \|\mathbf{q}_k^{(i)}\|} \sum_{i=1}^N w_k^{(i)} \mathbf{q}_k^{(i)} \end{cases}$$
。

② 协方差阵计算: $\Sigma_k = \sum_{k=1}^M w_k^{(m)} ((\mathbf{q}_k^{(i)} - \hat{\mathbf{q}}_k)(\mathbf{q}_k^{(i)} - \hat{\mathbf{q}}_k)^T)$ 。

5) 计算 Σ_k 的 Cholesky 分解 $\Sigma_k = \mathbf{C}_k \mathbf{C}_k^T$ 。

3.2 流水线高斯粒子滤波算法

3.1 节的伪码中步骤 1) ~ 3) 的①之间没有依赖关系, 能够简单地实现流水线运算, 步骤 3) 的②中, 权值标准化需要计算得到所有粒子的权值和, 因而无法与之前的步骤融合, 但可以通过使用非标准化的权值进行均值和协方差计算, 但显然均值和协方差阵必须通过权值和进行标准化, 将这一步放在最后实现。步骤 4)

的②需要状态估计值 $\hat{\mathbf{q}}_k$ 进行计算, 而 $\hat{\mathbf{q}}_k$ 需要等待 N 个粒子执行步骤 4) 的①完成后才能够得到。为了将协方差阵计算与上述步骤融合, 对协方差阵进行修改, 则此时可以将步骤 4) 的②协方差阵计算修改为

$$\Sigma_k = \frac{1}{W_k} \sum_{k=1}^M w_k^{(m)} \mathbf{q}_k^{(i)} (\mathbf{q}_k^{(i)})^T - \hat{\mathbf{q}}_k (\hat{\mathbf{q}}_k)^T \quad (16)$$

右边项 $\hat{\mathbf{q}}_k^{-1} (\hat{\mathbf{q}}_k^{-1})^T$ 是不变的, 可以在循环外计算。

因而流水线步骤 4) 的②能够与之前 3 个步骤融合为一个循环, 协方差矩阵计算时由于采用非标准化的权值, 因而与状态估计值一样需要在累加完成后进行标准化, 流水线的高斯粒子滤波姿态估计算法伪码如下所示。

输入: 粒子 $\{x_{k-1}^{(j)}\}_{j=1}^N$, IMU 输出 $f\boldsymbol{\omega}_{k-1}^{(i)}$ 和 $f\mathbf{a}_{k-1}$, GPS 输出 $g\tilde{\mathbf{v}}_k, \Sigma_{k-1}$

的 Cholesky 分解 $\mathbf{C}_{k-1}; \Sigma_{k-1} = \mathbf{C}_{k-1}\mathbf{C}_{k-1}^T$ 。

输出: 姿态和速度估计 $\hat{\mathbf{q}}_k$ 和 $g\hat{\mathbf{v}}_k$ 。

for $j = 1:N$

$m = r(j)$

1) 采样调节粒子:

$$\mathbf{q}_{k-1}^{(i)} = \hat{\mathbf{q}}_{k-1} + \mathbf{C}_{k-1} \tilde{w}_c^{(i)}$$

2) 采样步骤:

$$\begin{cases} \mathbf{q}_k^{(i)} = \left(\mathbf{I}_{4 \times 4} + \frac{\Delta t}{2} \boldsymbol{\Omega}(f\boldsymbol{\omega}_{k-1}^{(i)}) \right) \mathbf{q}_{k-1}^{(i)} + w_q^{(i)} \\ g\mathbf{v}_k^{(i)} = g\mathbf{v}_{k-1}^{(i)} + (\mathbf{C}(\mathbf{q}_{k-1}^{(i)})^{\#}(f\mathbf{a}_{k-1} + w_a^{(i)}) + \mathbf{g})\Delta t + w_v^{(i)} \end{cases}$$
。

3) ① 权值计算:

$$w_k^{(i)} = \frac{w_{k-1}^{(i)}}{\sigma_{gv}\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\|g\tilde{\mathbf{v}}_k - g\mathbf{v}_k^{(i)}\|}{\sigma_{gv}}\right)^2\right]$$

② 权值累加和计算: $W_k = W_{k-1} + w_k^{(i)}$ 。

4) ① 状态估计累加:
$$\begin{cases} g\hat{\mathbf{v}}_k = g\hat{\mathbf{v}}_{k-1} + w_k^{(i)} g\mathbf{v}_k^{(i)} \\ \hat{\mathbf{q}}_k = \hat{\mathbf{q}}_{k-1} + w_k^{(i)} \mathbf{q}_k^{(i)} \end{cases}$$
。

② 协方差阵累加: $\Sigma_k = \Sigma_{k-1} + w_k^{(m)} \mathbf{q}_k^{(m)} (\mathbf{q}_k^{(m)})^T$ 。

end

$$5) \text{ 状态调整: } \begin{cases} \hat{\mathbf{q}}_k = \frac{\hat{\mathbf{q}}_k}{\|\hat{\mathbf{q}}_k\|} \\ g\hat{\mathbf{v}}_k = \frac{g\hat{\mathbf{v}}_k}{W_k} \\ \Sigma_k = \frac{\Sigma_k}{W_k} \end{cases}$$

6) 计算协方差阵: $\Sigma_k = \frac{1}{W_k} \sum_{k=1}^M w_k^{(m)} \mathbf{q}_k^{(i)} (\mathbf{q}_k^{(i)})^T - \hat{\mathbf{q}}_k (\hat{\mathbf{q}}_k)^T$ 。

7) 计算协方差阵 Σ_k 的 Cholesky 分解 $\mathbf{C}_k; \Sigma_k = \mathbf{C}_k \mathbf{C}_k^T$ 。

3.3 时序分析

图 1 给出了一次高斯粒子滤波的时序分布, 其中, $L_1 \sim L_5$ 表示各个模块对应自身的时延, 对比 SPF 可以看到, 不同于 SPF $2N + L_{\text{sum}}$ 个时延, 高斯粒子滤波只有 $N + L_{\text{sum}}$ 个时延, 虽然高斯粒子滤波算法实现所需的复杂运算较多, 自身模块时延较大, 但当粒子数较多时, GPF 的实时性会好于 PF。

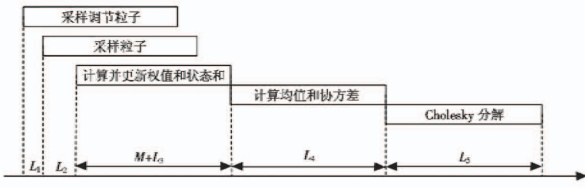


图 1 高斯粒子滤波时序图

Fig. 1 Timing diagram of Gaussian particle filter

3.4 基于 System Generator 的高斯粒子滤波算法实现

System Generator (SysGen) 是 Xilinx 提供的一款 FPGA 开发工具, 可以让使用者在 Simulink 环境下完成 FPGA 设计。本文将高斯粒子滤波模块化分解为如下几个模块: 控制模块、采样生成调节粒子模块、采样粒子模块、权值计算模块、均值和协方差计算模块以及 Cholesky 分解模块。本节主要介绍采样生成调节粒子模块、均值和协方差计算模块以及 Cholesky 分解模块的设计, 粒子滤波模块化流程如图 2 所示。

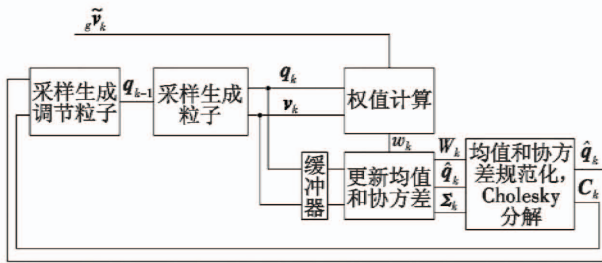


图 2 高斯粒子滤波算法模块化示意图

Fig. 2 Modular diagram of Gaussian particle filter algorithm

1) 采样生成调节粒子模块: 此模块中, 利用由均值

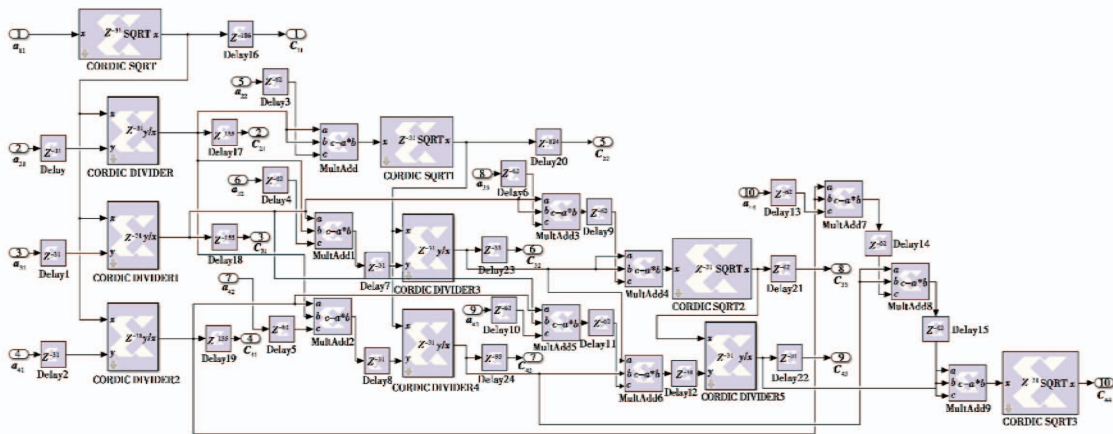


图 4 Cholesky 分解模块实现

Fig. 4 Implementation of Cholesky decomposition module

3) 均值和协方差计算模块: 此模块计算得到均值 \hat{q}_k 和协方差矩阵 Σ_n 。此模块需要 $N_s + N_s(N_s + 1)/2 = (N_s^2 + 3N_s)/2$ 个乘法器, 其中, N_s 表示模型维度, 对于四元数姿态估计模型 $N_s = 4$, 需要 14 个乘法器, 且 14 个输出均需要累加操作, 所以同时需要 14 个累加器。

协方差计算模块得到的均值 \hat{q}_k 和协方差阵分解 C_n 产生调节粒子, 矩阵 C_n 为 4×4 的上三角矩阵, 所有乘法器都是流水线运行的, 它们在 $M + L_1$ 个时钟周期内流水线生成 M 个调节粒子。此模块需要 4 个随机数产生器 (RNGs), 这些随机数产生器产生满足高斯分布的随机数, 产生调节粒子模块具体实现如图 3 所示。

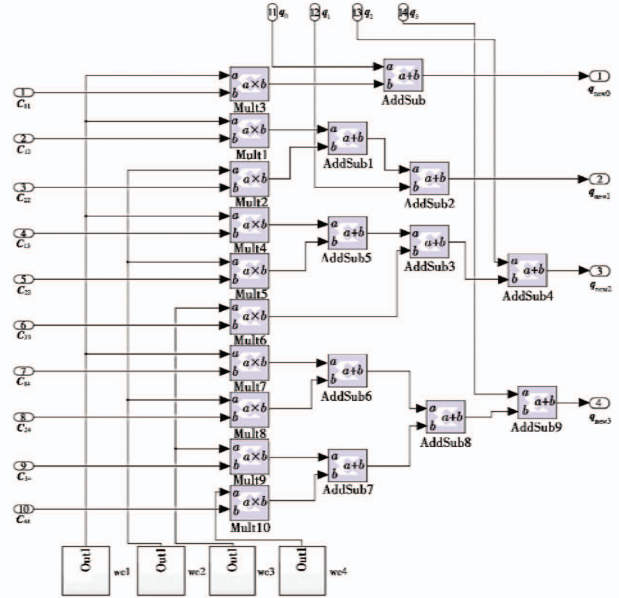


图 3 采样生成调节粒子模块

Fig. 3 Sample generation and adjustment particle module

2) 采样粒子模块, 权值计算模块: 除了粒子不需要存储在存储器中之外, 由粒子生成和权值计算步骤处理的设计与文献[7]中 SPF 的处理相同。这些步骤的设计已经给出。

4) Cholesky 分解模块如图 4 所示, 通过开方和除法运算实现式(15)所给出的方程。其中有 10 个输入, 10 个输出, 且包含 6 个除法运算, 4 个开方运算, 模块共有 217 个时延。除法运算和开方运算均由 CORDIC 算法实现, 这里直接采用 SysGen 提供的参考模块

CORDIC DIVIDER 和 CORDIC SQRT。表1给出了综合实现后的资源利用率。

表1 改进 GPF 的姿态估计算法 ZYNQ-7000 资源利用情况

Table 1 ZYNQ-7000 resource utilization of improved GPF pose estimation algorithm

资源	占用	总数	利用率/%
Slice LUTs	10 764	53 200	20.23
Slice Registers	16 767	106 400	15.75
Block RAMs	6	140	4.29
DSPs	89	220	40.45

运行一个周期的改进 GPF 姿态估计算法: $T_{SPF} = (2048 + 16 + 57 + 217) T_{clk} = 2338 T_{clk}$, 即需要 2338 个时钟周期。在 ZYNQ-7000 使用 100 MHz 晶振的情况下, 能获得的最大采样速率为 42 kHz。虽然资源利用率在 20% 左右, 但仍存在剩余, 如采用并行设计方案利用剩余资源, 能够进一步提升速率。

3.5 仿真分析

为了验证本文在 FPGA 上设计的粒子滤波姿态估计算法, 采用现有基于 Matlab 的飞行器航迹产生模型生成无人机飞行航迹, 利用 IMU 仿真输出陀螺仪加速度计数据和 GPS 仿真输出数据预存入 RAM 中, 验证该算法设计实现的有效性。仿真中, 采用东北天坐标系作为导航坐标系, 右前上坐标系作为机体坐标系, 俯仰角以抬头为正, 滚转角以右倾为正, 方位角以北偏东为正。假设无人机做机动飞行, 其飞行轨迹中含有爬升、变速、平飞和转弯等各种飞行状态, 其飞行航迹如图 5 所示。飞机的初始位置为 (20°N, 110°E)、飞行高度为 500 m, 飞机的初始速度为 0 m/s, 航向正东。设惯导系统的等效陀螺漂移为 0.1 (°)/h, 等效加速度零偏为 $10^{-4} g$; 陀螺一阶马尔可夫过程相关时间为 3600 s, 加速度零偏一阶马尔可夫过程相关时间为 1800 s, 选取状态方程的离散化周期为 1 s。

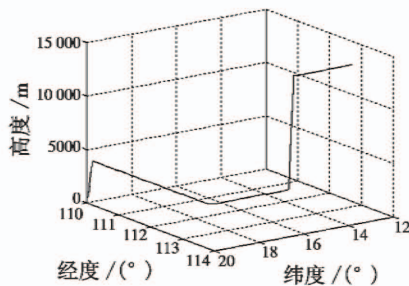


图5 无人机飞行航迹仿真图

Fig. 5 UAV flight path simulation

数值方面均由 24 位定点数表示, 其中: 四元数由 1 位符号位、3 位整数位和 20 位小数位组成; 地理系速度由 1 位符号位、6 位整数位和 17 位小数位组成; 权值由 1 位整数位和 23 位小数位组成。仿真结果如图 6 所示。

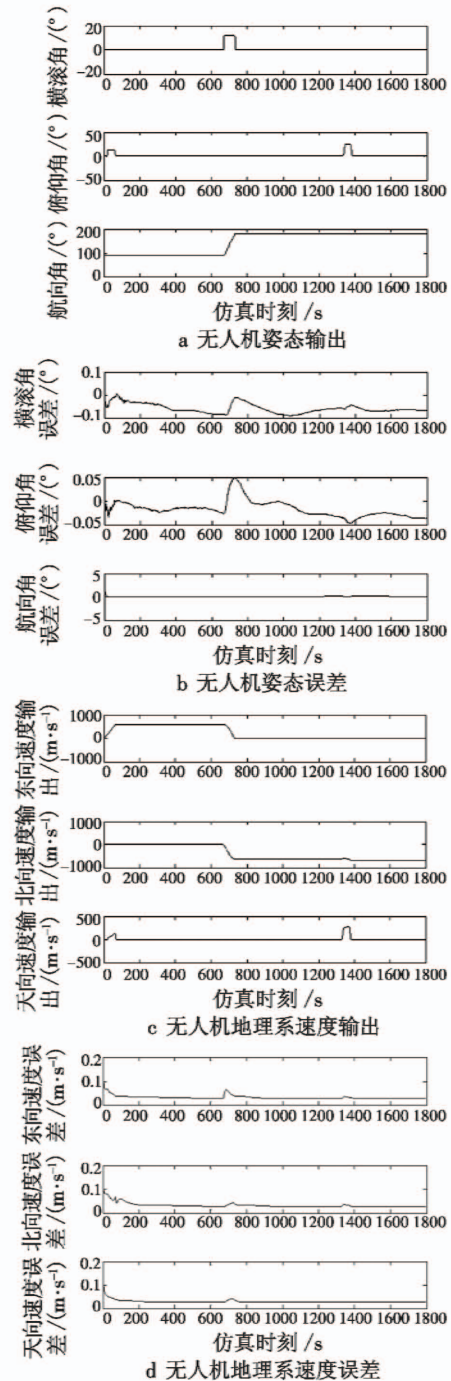


图6 基于 FPGA 平台的高斯粒子滤波结果

Fig. 6 FPGA-based Gaussian particle filter

上述仿真结果表明, 基于 SysGen 所设计的无人机姿态估计算法在 FPGA 上实现是可行且有效的, 由于定点数的设置以及 FPGA 平台上相关求解的近似, 仿真具有一定的误差。

4 结束语

本文采用高斯粒子滤波算法进行姿态估计算法设计, 解决重采样算法设计十分复杂且不利于并行实现

(下转第 75 页)

明,本文提出的算法简单成熟,流程正确有效,使去斜步进频合成可应用于实际雷达系统,且非常适合于实时处理,便于工程实现,可用于宽带/超宽带信号的产生与处理。

参考文献

- [1] 保铮,邢孟道,王彤. 雷达成像技术[M]. 北京:电子工业出版社,2005.
- [2] 龙腾,毛二可,何佩琨. 调频步进雷达信号分析与处理[J]. 电子学报,1998,26(12):84-88.
- [3] 李旸,龙腾. 步进频雷达目标去冗余算法[J]. 电子学报,2000,28(6):60-63.
- [4] 毛二可,龙腾,韩月秋. 频率步进雷达数字信号处理[J]. 航空学报,2001,22(s):16-24.
- [5] 贺志毅,郝祖全,汤斌. 一种新体制雷达合成宽带可行性分析[J]. 电子学报,2003,31(6):940-943.
- [6] LORD R T. Aspects of stepped-frequency processing for low frequency systems[D]. Cape Town:University of Cape Town,2000.
- [7] LEVANAN N, MOZESON E. Nullifying ACF grating lobes in stepped-frequency train of LFM pulses [J]. IEEE Transactions on Aerospace and Electronics Systems,2003,39(2):694-703.
- [8] JING W, YANG L, TENG L. Processing method base on synthetic spectrum using stepped-frequency chirps [C]//

- IET International Radar Conference,2009:1-4.
- [9] ZHANG L, GAO Z Z. Stepped-frequency waveform processing for moving target based on genetic-CLEAN algorithm [C]//The 2nd International Conference on Signal Processing Systems,2010:18-21.
- [10] ZHAO X N, LIU H B, LIU Q B, et al. Parameters design and signal processing algorithm research in clutter based on the stepped frequency waveforms [C]//The 11th International Conference on Signal Processing,2012:1900-1904.
- [11] ZHAO L Z, FU X J, WU J T, et al. A novel range profiles splicing approach in stepped-frequency radars [C]//IET International Radar Conference,2013:1-5.
- [12] NGUYEN L. Signal processing techniques for stepped frequency ultra-wideband radar [J]. Proceedings of SPIE,2014. doi:10.1117/12.2053026.
- [13] 江碧涛,张云华,姜景山. 宽带调频步进信号的全去斜处理方法 [J]. 测试技术学报,2008,22(3):225-230.
- [14] 张忠传,牟善祥,徐光进,等. 基于 Stretch 处理的毫米波调频步进雷达信号处理技术研究 [J]. 南京理工大学学报:自然科学版,2010,34(1):121-125.
- [15] 金胜,朱天林,邓振森,等. 基于去斜的步进频合成高分辨距离像新方法 [J]. 系统工程与电子技术,2012,34(6):1126-1130.

(上接第70页)

的问题。通过 SysGen 在 FPGA 上实现算法,仿真验证了 GPF 算法是可行的,且对比 PF 具有更好的实时性。将四元数离散方程作为状态方程,算法分为采样调节粒子、采样粒子、权值计算、均值协方差计算和 Cholesky 分解 5 个模块,通过采用非标准化的权值计算四元数平均值和协方差阵,并且改写协方差阵计算公式,实现流水线高斯粒子滤波算法,仿真结果表明了本设计的有效性,且提出了并行化设计方案,利用 FPGA 剩余资源进一步提升运行速率。但上述实现均采用模拟航迹进行仿真,对于真实无人机姿态估计的具体实现及工程应用(例如,真实 IMU 产生的数据导入 FPGA,产生的姿态估计结果用于机载控制系统)仍需要大量的板级调试工作。

参考文献

- [1] HONG S, DJURIĆ P M, BOLIĆ M. Simplifying physical realization of Gaussian particle filters with block-level pipeline control [J]. EURASIP Journal on Applied Signal Pro-

- cessing,2005,4:575-587.
- [2] 王宁,王从庆. 高斯粒子滤波器及其在非线形估计中的应用 [J]. 南京航空航天大学学报,2006,38(s):132-135.
- [3] OSHMAN Y, CARMI A. Attitude estimation from vector observations using a genetic algorithm embedded quaternion particle filter [J]. Journal of Guidance, Control & Dynamics,2006,29(4):879-891.
- [4] STEVENS B L, LEWIS F L, JOHNSON E N. Aircraft control and simulation:dynamics, controls design, and autonomous systems [M]. 3rd ed. Hoboken:John Wiley & Sons,2015.
- [5] 周翟和. 惯性组合导航系统中线性/非线性混合粒子滤波技术研究 [D]. 南京:南京航空航天大学,2010.
- [6] 王佳辉,王义平,薛雅丽. 基于 System Generator 的简化粒子滤波算法设计及硬件实现 [J]. 光电与控制,2018,25(5):100-105.
- [7] MARKLEY F L. Attitude estimation or quaternion estimation [J]. Journal of the Astronautical Sciences,2004,52(1/2):221-238.