

基于 FPGA 的目标运动预测实时 kalman 滤波器设计

冯安祥,肖阳辉,王玉良,栗雪峰

(中国科学院沈阳自动化研究所,中国科学院研究生院,辽宁 沈阳 110016)

摘要: 卡尔曼滤波器被广泛应用于目标位置预测领域,但由于算法复杂、计算量大,采用软件实现难以满足现代跟踪系统的实时性要求。本文提出了一种基于 FPGA 的可重配置实时卡尔曼滤波器硬件结构,该结构采用并行流水线技术,可极大提高运算速度。同时,在满足实时性要求的前提下,采用时分复用技术,大量减少了资源占用,节约了成本。

关键词: 卡尔曼滤波器;实时性;FPGA;时分复用

Design of real-time kalman filter for target forecast in FPGA

FENG An Xiang, XIAO Yang Hui, WANG Yu Liang, LI Xiao Feng
(Shenyang Institute of Automation, Chinese Academy of Sciences,
Graduate School of the Chinese Academy of Sciences, Shenyang 110016, China)

Abstract: Kalman filter has been widely used in target location forecast. But the computational time for the algorithms in software is too long to meet today's tracking system needs. This paper presents a architecture for FPGA-based reconfigurable Real-time kalman filter. The operating time of the filter is reduced hugely by the hardware parallel pipelining architecture. At the same time, time division multiplexing(TMD) technology is used to reduce the use of resource and the cost under the premise of meeting real-time requirements.

Key words: Kalman filter; real-time; FPGA; time division multiplexing

卡尔曼滤波是一种线性最小方差估计,算法采用递推形式在时域内滤波,数据存储量小,预测精度高,适用于对多维随机过程进行估计。由于以上优点,它已经成为一最种最重要的估计理论。但是由于卡尔曼滤波的算法复杂,对计算资源的速度和并行性要求比较高,特别是当需要预测多个目标的运动时,当前的计算机软件速度很难满足实时性要求,以至于在某些情况下不得不采用精度较低但相对简单的算法。

FPGA 是一种半定制集成电路,既有硬件逻辑固有的高速、高并行性的特点,又有可编程、设计灵活的优点。它可以在满足高精度的前提下,高速完成复杂的算法,非常适合用来设计应用于高实时性要求领域的 Kalman 滤波器。本文充分利用 FPGA 的上述特点,实现了一种基于 FPGA 的硬件 Kalman 滤波器,在保证高精度的前提下,解决了其在应用过程中的实时性问题。

1 kalman 滤波器计算过程描述

Kalman 滤波的计算过程是一个不断“预测-修正”的过程,求解过程中无需存储大量数据,一旦观测到新的数据就可以计算出新的滤波值,因此非常适合于硬件实

现中的实时性处理。

以一维直线运动为例,系统状态方程和观测方程分别为:

$$X_k = \phi X_{k-1} + \Gamma j_{k-1} \quad (1)$$

$$Z_k = H_k s_k + v_k \quad (2)$$

式(1)中: $X_k = [s_k, v_k, a_k]^T$ 为状态向量, s_k 、 v_k 、 a_k 分别为位移、速度和加速度。

$$\phi = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \text{为状态转移矩阵}; \Gamma = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}, \text{为噪声}$$

输入矩阵; Z_k 为系统观测值; H_k 为系统观测矩阵; T 为系统迭代周期。如果只对物体位移作观测,则

$$H_k = [1 \ 0 \ 0] \quad (3)$$

j_{k-1} 为加加速度,对运动体的跟踪者来说,加加速度 j_{k-1}

是随机量,此处取为系统白噪声,且 $E[j_{kj}^T] = \begin{cases} q & l=k \\ 0 & l \neq k \end{cases}$ 。

v_k 为观测白噪声,且 $E[v_k v_l^T] = \begin{cases} r & l=k \\ 0 & l \neq k \end{cases}$ 。

应用 Kalman 滤波基本滤波方程式如下：

$$\hat{X}_k = \phi \hat{X}_{k-1} + K_k (Z_k - H_k \phi \hat{X}_{k-1}) \quad (4)$$

$$P_{k,k-1} = \phi P_{k-1} \phi^T + q \Gamma \Gamma^T \quad (5)$$

$$P_k = [I - K_k H] P_{k,k-1} \quad (6)$$

$$K_k = P_{k,k-1} H^T [H P_{k,k-1} H^T + R]^{-1} \quad (7)$$

\hat{X}_k 表示 k 时刻系统状态 X_k 的估计值, 则 P_k 为估计误差方差阵, 其物理意义为:

$$P_k = E[\tilde{X}_k \tilde{X}_k^T] = E[(X_k - \hat{X}_k)(X_k - \hat{X}_k)^T] \quad (8)$$

一步预测误差方差阵:

$$P_{k,k-1} = E[(X_k - \phi \tilde{X}_{k-1})(X_k - \phi \tilde{X}_{k-1})^T] \quad (9)$$

P_k 和 $P_{k,k-1}$ 分别为估计值与实际值的误差方差及估计值与一步预测值“ $\phi \tilde{X}_{k-1}$ ”的误差方差。 K_k 为滤波增益矩阵, 其大小反映了观测值 Z_k 将对最终滤波值影响的大小。

以上公式表示的 Kalman 滤波算法方框图如图 1。

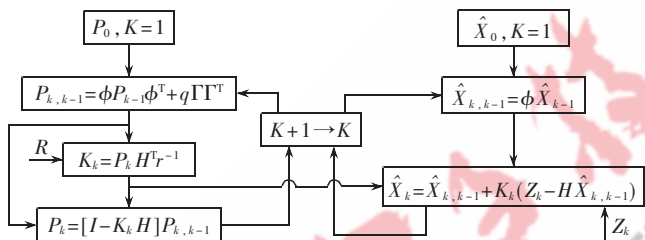


图 1 Kalman 滤波算法方框图

2 基于 FPGA 的 Kalman 滤波算法的实现

如图 1 所示, Kalman 滤波算法由两个回路五个模块组成, 各模块均由矩阵运算完成。矩阵运算经分解可由加法器、减法器、乘法器、除法器及累加器等来实现。下面就这五个模块分别介绍。

2.1 $P_{k,k-1} = \phi P_{k-1} \phi^T + q \Gamma \Gamma^T$ 模块

由式(8)、式(9)可知, P_k 和 $P_{k,k-1}$ 均为对称阵, 且

$$P_{k,k-1} = \begin{bmatrix} PL_{11} & PL_{12} & PL_{13} \\ PL_{21} & PL_{22} & PL_{23} \\ PL_{31} & PL_{32} & PL_{33} \end{bmatrix}。矩阵各元素如下:$$

$$PL_{11} = P_{11} + (P_{12} + P_{22})T + (P_{22} + P_{13})T^2 + P_{23}T^3 + P_{33}T^4/4,$$

$$PL_{12} = P_{12} + (P_{22} + P_{13})T + 3P_{23}T^2/2 + P_{33}T^3/2,$$

$$PL_{13} = P_{13} + P_{23}T + P_{33}T^2/2, PL_{22} = P_{22} + 2P_{23}T + P_{33}T^2,$$

$$PL_{23} = P_{23} + P_{33}T, PL_{33} = P_{33} + qT^2。$$

由 Kalman 滤波理论可知, Q 、 R 和 P_0 均扩大 α 倍, 增益矩阵 K 不变。因此可以在维持增益矩阵不变的情况下调整 Q 、 R 和 P_0 , 从而使误差矩阵保持在一定的数据范围内, 这样可以将浮点运算简化为定点运算。在本文中通过调整, 使误差矩阵中的元素值保持在小于 1 的范围内。该模块的计算非常适合用乘累加器来实现, 如图 2 所示。首先计算出式中与 P 有关的系数(在图 2 中称为

P 系数, 如 P_{11} 、 $P_{12}+P_{22}$ 等)和与 T 有关的系数(在图 2 中称为 T 系数, 如 T 、 T^2 等), 依次输出到乘累加器中并定时读取和复位乘累加器就可得到 PL_{XX} 。大量的加法和乘法运算通过时分复用, 由一个乘累加器来实现, 大大减少了资源的占用。本模块输出了定点 PL 和浮点 PL , 分别作为 $K_k = P_{k,k-1} H^T [H P_{k,k-1} H^T + R]^{-1}$ 模块和 $P_k = [I - K_k H] P_{k,k-1}$ 模块的输入。

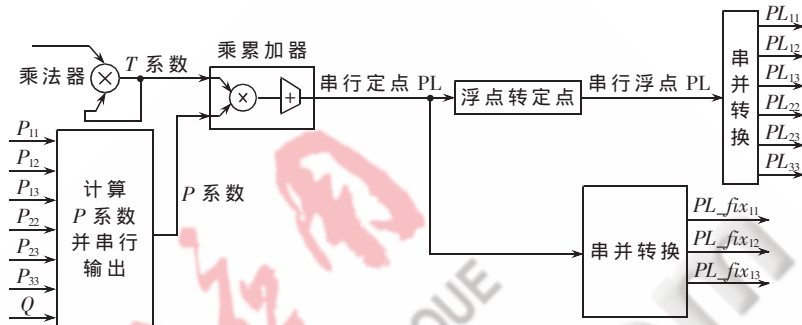


图 2 $P_{k,k-1} = \phi P_{k-1} \phi^T + q \Gamma \Gamma^T$ 模块

2.2 $K_k = P_{k,k-1} H^T [H P_{k,k-1} H^T + R]^{-1}$ 模块

由式(3)、式(7)可得:

$$\begin{bmatrix} K_{11} \\ K_{21} \\ K_{31} \end{bmatrix} = \begin{bmatrix} PL_{11} \\ PL_{21} \\ PL_{31} \end{bmatrix} / (PL_{11} + R) \quad (10)$$

该模块要用到浮点除法器, FPGA 中没有基于芯片内部自带的浮点除法器。如果浮点除法器被调用, 它最后将占用片内的逻辑资源来生成, 因此非常有必要将除法器时分复用, 如图 3。

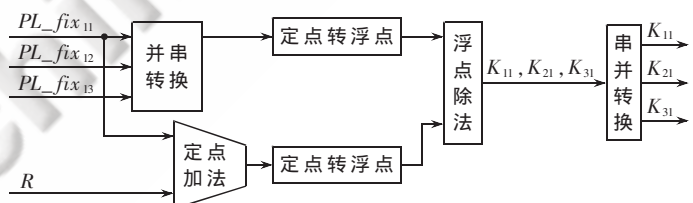


图 3 $K_k = P_{k,k-1} H^T [H P_{k,k-1} H^T + R]^{-1}$ 模块

2.3 $P_k = [I - K_k H] P_{k,k-1}$ 模块

前面已经介绍 P_k 为对称矩阵, 且 $P_k = \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix}$,

其中: $P_{11} = PL_{11} - K_{11} P_{11}$, $P_{12} = PL_{12} - K_{11} P_{12}$, $P_{13} = PL_{13} - K_{11} P_{13}$, $P_{22} = PL_{22} - K_{21} P_{12}$, $P_{23} = PL_{23} - K_{21} P_{13}$, $P_{33} = PL_{33} - K_{31} P_{13}$ 。 P_k 元素的计算都由一次减法和一次乘法完成。通过时分复用可以用一个减法器和一个乘法器来完成所有运算, 如图 4 所示。

2.4 $\hat{X}_k = \phi \hat{X}_{k-1}$ 模块

将式 $\phi = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}$ 代入 $\hat{X}_k = \phi \hat{X}_{k-1}$ 可得:

$$\begin{cases} XL_1 = X_1 + X_2 T + \frac{X_3}{2} T^2 \\ XL_2 = X_2 + X_3 T \\ XL_3 = X_3 \end{cases} \quad (11)$$

由式(11)可知,求 XL_1 需要一个乘累加器,求 XL_2 时需要一个乘加器。乘累加器和乘加器可以用一个乘法器和一个加法器及三个二选一选择器来组成,如图5所示。当 ch 为高电平时该部分为乘累加器,当 ch 为低电平时该部分为乘加器。

2.5 $\hat{X}_k = \hat{X}_{k,k-1} + K_k(Z_k - H\hat{X}_{k,k-1})$ 模块

由式(1)、式(4)可得: $\hat{X}_k = [X_1, X_2, X_3]^T$, 其中: $X_1 = XL_1 + K_{11}(Z_k - XL_1)$; $X_2 = XL_2 + K_{21}(Z_k - XL_1)$; $X_3 = XL_3 + K_{31}(Z_k - XL_1)$, 以上三式都经过了一次减法、一次乘法法和一次加法, 其中

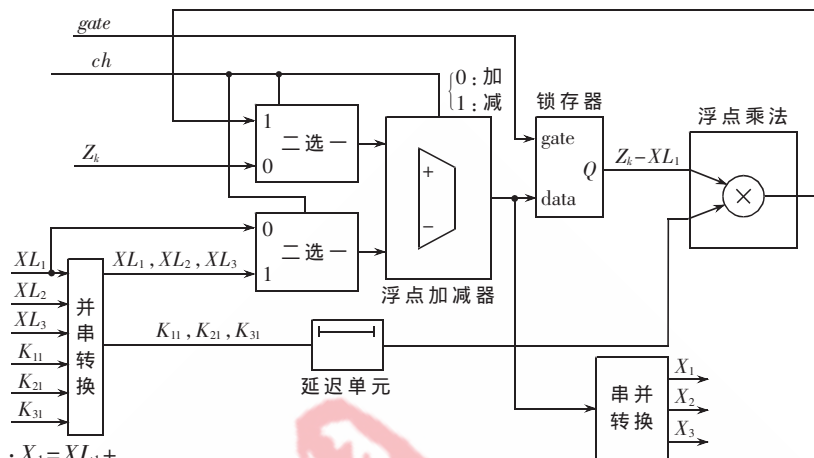


图6 $\hat{X}_k = \hat{X}_{k,k-1} + K_k(Z_k - H\hat{X}_{k,k-1})$ 模块

快,可以用一个一维的滤波器通过时分复用来完成两个方向的滤波运算。本文设计的一维 Kalman 滤波器完成一次迭代需要 135 个时钟,若时钟周期为 20ns,则一次迭代的周期为 2.7 μ s,完全可以满足实时性的要求。此外,Kalman 滤波器不需要存储大量数据,这使得时分复用的资源代价非常低。笔者通过插入 3 组 FIFO 实现了一维滤波器的二维时分复用,如图7所示。

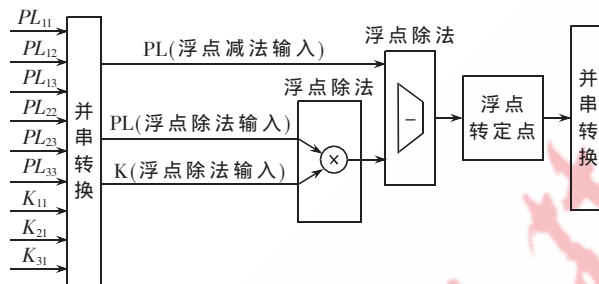


图4 $P_k = [I - K_k H] P_{k,k-1}$ 模块

($Z_k - XL_1$) 在三式中都存在。加减器是 FPGA 自带的 DSP 处理单元,如图6所示,可以通过外部信号 ch 决定其是加法器还是减法器,因此可以先让其执行减法功能,待($Z_k - XL_1$)完成并被锁存后再执行加法器功能,因此该部分也只用了一个加减器和一个乘法器。

3 一维 Kalman 滤波器用于预测目标二维运动

图像中的目标是二维运动的,可以将并行的两个一维 Kalman 滤波器用于目标水平和俯仰方向的运动,其占用的资源是一维滤波器的两倍。若滤波器的速度足够

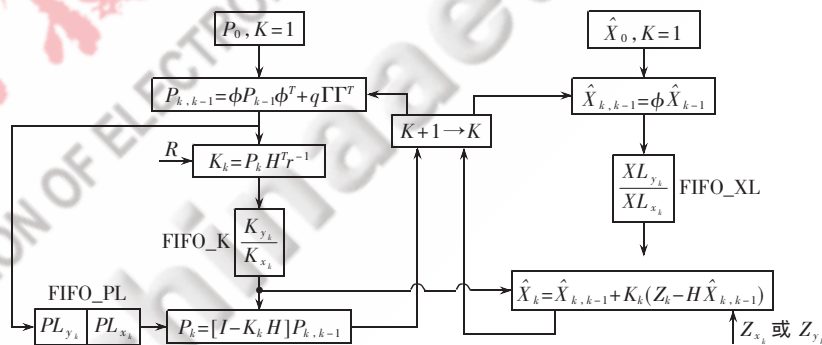


图7 一维滤波器时分复用图

图7和图1的区别是: $P_{k,k-1} = \phi P_{k-1} \phi^T + q \Gamma \Gamma^T$ 、 $K_k = P_{k,k-1} H^T [H P_{k,k-1} H^T + R]^{-1}$ 和 $\hat{X}_{k,k-1} = \phi \hat{X}_{k-1}$ 模块后面分别多了 FIFO_PL、FIFO_K 和 FIFO_XL 三个先入先出寄存器,外部观测量也变为 Z_{x_k} 、 Z_{y_k} 交替输入, X 方向和 Y 方向的滤波迭代也将交替进行。

4 资源占用

本设计中的浮点算术单元均为 IEEE754 标准中的 43 位扩展单精度,非常适合应用在对精度要求较高的视频跟踪中,在 Altera 公司的 Stratix 系列的 EP1S40F780C8 中完全编译后,显示其占用的资源情况如表1。

若应用在对精度要求相对较低的

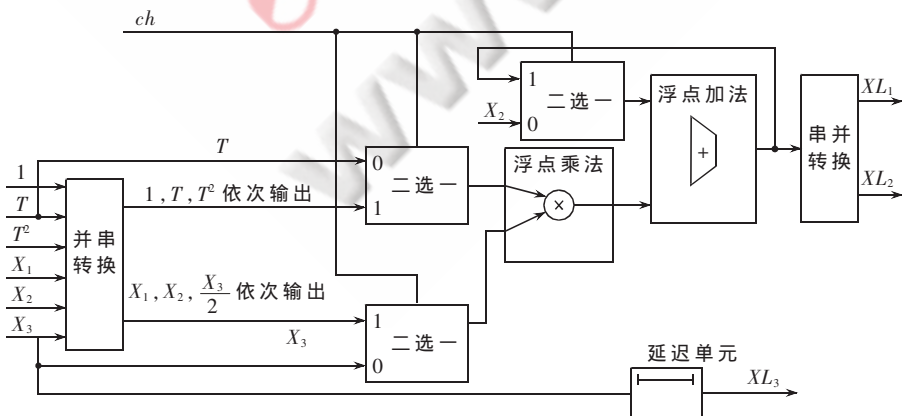


图5 $\hat{X}_{k,k-1} = \phi \hat{X}_{k-1}$ 模块

表 1 43 位精度时的资源消耗

	数量	百分比/%
逻辑单位	15 300	37
内存位数	209 664	6
DSP 块单元	32	29

的环境中,可将数据位宽降低,资源消耗还会大量减少。

现代目标预测系统的发展趋势是:需要被预测的目标越来越多,对精度的要求也越来越高。由于当前软件的速度还不够快,这对基于软件的预测系统来说是一个越来越严峻的考验。由基于 FPGA 的 Kalman 滤波器来代替传统的基于软件的滤波器是解决这一问题的有效途径。试验结果表明,相对于软件滤波器来说,基于 FPGA

的 Kalman 滤波器的速度可以提升三个数量级。速度的提高意味着在保证实时性的前提下,系统可以预测更多的目标,实现更复杂的算法,获得更高的精度。

参考文献

- [1] 付梦印,邓志红,张继伟.Kalman 滤波理论及其在导航系统中的应用.北京:科学出版社,2003.8.
- [2] LEE C.R,SALCIC Z.A Fully_hardware_type Maximum_parallel Architecture for Kalman Tracking Filters in FPGAs. Singapore, 9-12(9).
- [3] SINGER R.A,BEHNKE K.W,Real-time tracking filter evaluation and selection for tactical applications.IEEE Trans. Aersp.Electron.AES-7, 100110, 1971.

(收稿日期:2007-12-24)

