



论文

单状态变量和两级 Kalman 滤波器时间尺度算法

伍贻威*, 龚航, 朱祥维, 欧钢

国防科技大学电子科学与工程学院, 长沙 410073

*联系人, E-mail: Yiwei_Wu_sh@126.com

收稿日期: 2015-11-25; 接受日期: 2016-01-25; 网络出版日期: 2016-05-04

摘要 本文首先提出一种单状态变量 Kalman 滤波器时间尺度算法; 推导了算法的权重和预测值的解析表达式; 在该解析表达式的基础上, 引入虚拟 Kalman 采样间隔; 理论分析和仿真实验都证明: 通过选取不同的虚拟 Kalman 采样间隔, 可以使时间尺度在任何指定的平滑时间的频率稳定度达到最优. 在此基础上, 提出了一种两级 Kalman 滤波器时间尺度算法; 详细描述了算法原理; 算法生成的时间尺度只含有频率随机游走噪声(RWFM), 不含有频率白噪声(WFM); 算法相当于按照 RWFM 的强度进行加权; 生成的时间尺度的中短期频率稳定度更高.

关键词 时间尺度算法, Kalman 滤波器, 虚拟 Kalman 采样间隔, 权重, 频率稳定度

PACS: 06.20.Dk, 06.30.Ft, 93.85.Bc

1 引言

时间尺度算法(Time Scale Algorithms)在守时实验室和全球导航卫星系统(GNSS)中发挥着重要作用. 时间尺度算法^[1-3]的目的是为了综合钟组中所有的原子钟, 生成一个可靠性、频率稳定度更高的、自由的纸面时间. 单台钟和纸面时间都可以视为是一个时间尺度; 没有被驾驭的时间尺度称为自由的时间尺度. 所以说, 使用时间尺度算法是为了生成一个时间尺度, 该时间尺度一般通过它与某一台钟之间的钟差来表示^[2,3]; 算法通常不考虑该时间尺度的频率准确度, 将提升其频率准确度的任务留给驾驭算法来完成^[2].

时间尺度算法通常可以分为两大类: 一类是经

典的加权平均算法^[4-6], 例如: 国际计量局(BIPM)采用的加权平均算法(称为: ALGOS 算法^[4,5])、美国国家标准与技术研究院(NIST)采用的加权平均算法(称为: AT1 算法^[4,6]); 对于任何加权平均算法, 生成的时间尺度都可以用时间尺度基本方程^[2-4]来表示; 另一类是采用 Kalman 滤波器实现的算法^[7,8]. 由于 AT1 算法和 Kalman 滤波器算法所生成的时间尺度是实时的, 比较适用于 GNSS.

20 世纪 80 年代 NIST 首次采用 Kalman 滤波器建立时间尺度^[7,8]. 在一段时期内, 该算法和 AT1 算法同时在 NIST 运行. 该算法的不足在于^[7-9]: (1) 由于系统不完全可观测, 造成估计误差矩阵发散, 表明状态估计的误差不断变大; (2) 实验表明: 对于铯钟和氢钟组成的钟组, 时间尺度 TA 主要由长期频率稳

引用格式: 伍贻威, 龚航, 朱祥维, 等. 单状态和两级 Kalman 滤波器时间尺度算法. 中国科学: 物理学 力学 天文学, 2016, 46: 069502
Wu Y W, Gong H, Zhu X W, et al. One-state and twice Kalman filter time scale algorithms (in Chinese). Sci Sin-Phys Mech Astron, 2016, 46: 069502, doi: 10.1360/SSPMA2015-00590

定度最好的钟决定, 从而没有利用到钟组中短期频率稳定度较好的钟的短期频率稳定度性能. 由于这些不足, NIST 后来放弃了该算法, 只有 AT1 算法一直运行至今. 之后一系列的 Kalman 滤波器算法都是在该算法的基础上改进后得到的, 主要成果有: (1) 文献[9]和[10]分别提出了两种不同的方法来约简估计误差矩阵. 其中, GPS 就是采用文献[10]的方法来生成自由的“GPST”的; (2) 文献[9]认为 NIST 的算法本质上也是加权平均算法, 给出了由 Kalman 增益来表示的权重; (3) 文献[11]指出: 文献[9]提出“约简 Kalman 滤波器算法”(Reduced Kalman Filter Algorithm, 简称为: Kred 算法), 不仅约简了估计误差矩阵, 还提升了时间尺度的中短期频率稳定度; (4) 文献[12]再次对 NIST 算法和 Kred 算法的原理和性能进行了对比分析; (5) 文献[13]将文献[9,10]中两种约简估计误差矩阵的方法组合起来, 提出了一种组合约简方法.

本文作者对上述 Kalman 滤波器算法进行了深入分析和思考. 认为文献[9]的不足在于没有给出由原子钟噪声特性来表示的权重的解析表达式, 而通过对该解析表达式的推导, 可以加深对一系列的 Kalman 滤波器时间尺度算法的本质理解. 所以, 与传统研究思路不同, 本文在文献[9]的基础上, 将算法生成的时间尺度通过时间尺度基本方程^[2-4]来表示, 从算法作为加权平均算法的角度, 而不是状态估计误差的角度(即研究如何约简估计误差)对算法进行分析, 推导了权重和预测值的解析表达式, 从如何优化权重的角度来分析和设计算法.

首先, 本文提出一种单状态变量 Kalman 滤波器时间尺度算法. 其中, 每台钟只有 1 个状态变量, 即每台钟的时差. 在获得了权重的解析表达式的基础上, 引入了虚拟 Kalman 采样间隔, 用于计算过程噪声协方差; 并证明在这种情况下, 权重将反比于平滑时间为虚拟 Kalman 采样间隔的 Allan 方差. 这样, 可以通过选取虚拟 Kalman 采样间隔, 使时间尺度的频率稳定度在任意某一个指定的平滑时间达到最优.

在此基础上, 提出了一种两级 Kalman 滤波器时间尺度算法. 第一级的 Kalman 滤波器用于估计钟差的两个状态. 其中, 第一个状态中包含了频率白噪声(WFM)和频率随机游走噪声(RWFM)两种噪声, 而第二个状态中只包含了 RWFM. 第二级的 Kalman 滤波器实际上是一个单状态变量 Kalman 滤波器时间尺度

算法, 用于生成时间尺度; 其输入为第一级 Kalman 滤波器的第二个状态估计值的“重构时差”(对其的描述详见本文第 4 节). 这样, 算法相当于对只含有 RWFM 的时间序列进行加权, 生成的时间尺度中也只含有 RWFM 这一种噪声, 中短期频率稳定度更高.

本文将首先描述时间尺度基本方程, 原子钟模型, 和 NIST 的 Kalman 滤波器时间尺度算法的基本原理. 在此基础上, 详细介绍本文提出的单状态变量 Kalman 滤波器时间尺度算法和两级 Kalman 滤波器时间尺度算法, 通过严密的数学推导, 给出算法的完整理论框架. 然后, 通过仿真实验来验证理论分析的结论. 最后对全文进行总结和展望.

2 时间尺度与原子钟模型

2.1 时间尺度基本方程

时间尺度基本方程^[2-4]表示为

$$TA(t) = \sum_{i=1}^N \omega_i (x_i(t) - x'_i(t)), \quad (1)$$

其中, $TA(t)$ 是生成的纸面上的时间尺度, N 为钟组中原子钟的数量, ω_i 为第 i 台原子钟的权重, $x_i(t)$ 代表第 i 台原子钟的时差, $x'_i(t)$ 代表第 i 台原子钟的时差的预测值.

式(1)可以这样理解: 对于加权平均时间尺度算法, 权重用于优化时间尺度的频率稳定度, 而预测值则消除了每台钟确定性趋势项的影响. 所以, TA 可以看成是对去除确定性趋势项后只含有噪声分量的原子钟时差的加权平均值.

事实上, 不论是 ALGOS 算法还是 AT1 算法, 其生成的时间尺度都可以进行化简, 最终使用时间尺度基本方程(1)来表示^[4].

2.2 原子钟模型

文献[2,14,15]完整地描述了原子钟随机微分方程(Stochastic Differential Equations, SDEs). 该 SDEs 对 WFM 和 RWFM 进行了建模. 该 SDEs 的闭合解表示为

$$\begin{cases} X_1(t) = x_0 + y_0 \cdot t + 1/2 \cdot d \cdot t^2 + \sigma_1 \cdot W_1(t) \\ \quad + \sigma_2 \cdot \int_0^t W_2(s) ds, \\ X_2(t) = y_0 + d \cdot t + \sigma_2 \cdot W_2(t), \end{cases} \quad (2)$$

其中, X_1 , X_2 分别为两个状态变量, 其中 X_1 代表瞬时

时差, X_2 代表瞬时频差的随机游走部分, 即不含 WFM 的瞬时频差. x_0 和 y_0 分别代表瞬时频差和瞬时频差的初值, d 代表频漂. $W_1(t)$ 和 $W_2(t)$ 分别代表两个独立的维纳过程, 并且有 $W(t) \sim N(0, t)$, 即每个维纳过程服从均值为 0, 方差为时间 t 的正态分布. σ_1 和 σ_2 分别是这两个维纳过程的扩散系数 (Diffusion Coefficients), 用于表明噪声的强度.

式(2)表明: $W_1(t)$ 和 $W_2(t)$ 对时间 t 的积分在状态变量 X_1 上分别表现为 WFM 和 RWFM, 即式(2)的第一个方程中, 最后第二项代表 WFM, 最后一项代表 RWFM.

事实上, NIST 的 Kalman 滤波器时间尺度算法也只对这两种噪声进行建模. 一方面是因为这样的钟差模型具有直观的物理意义, 闪烁噪声很难用 SDEs 进行建模. 另一方面, 大量实验都表明: 铯钟和氢钟中起主导作用的噪声确实为 WFM 和 RWFM.

文献[14]还详细推导了扩散系数与 Allan 方差的关系, 其结论如下:

$$\sigma_y^2(\tau) = \sigma_1^2 / \tau + 1/3 \cdot \sigma_2^2 \cdot \tau, \quad (3)$$

其中, $\sigma_y^2(\tau)$ 代表平滑时间为 τ 时的 Allan 方差, τ 为平滑时间. 式(3)中等号后面第一项的斜率为 -1, 第二项斜率为 1, 这两项分别对应 WFM 和 RWFM, 这说明在对数 Allan 方差图中, WFM 的斜率为 -1, RWFM 的斜率为 1. 在实际运用中, 很容易通过对 Allan 方差拟合得到扩散系数的值. 估计扩散系数会引入估计误差, 这与估计方法有关, 不属于本文研究内容. 本文中, 为了分离研究点, 仿真生成的氢钟和铯钟的扩散系数值都是已知的.

2.3 NIST 的 Kalman 滤波器时间尺度算法的基本原理

该算法的原理可以概括为: 对于一个含有 N 台钟的钟组, 每台钟有 2 个(时差、频差)或 3 个(时差、频差、频漂)状态变量, 所以共有 $2N$ 或 $3N$ 个状态变量. 该 Kalman 滤波器的输入观测量为 $N-1$ 组钟差, 估计得到 $2N$ 或 $3N$ 个状态变量. 其中 N 个时差估计值代表了每台钟与时间尺度 TA 之间的时间差, 于是每台钟的时差减去时差估计值, 得到“校准钟”(The Corrected Clock), 这样共可以得到 N 台“校准钟”, 每台“校准钟”都可以代表时间尺度 TA. 当测量噪声为零时, 由任何一台“校准钟”表示的 TA 都是相同的. 读者可以在阅读完第 3.1 节后再次重读这段文字, 加

深对 NIST 算法原理的理解.

鉴于文献[7,8,12]都对 NIST 算法的原理和性能进行了详细描述, 本文中不展开描述.

3 单状态变量 Kalman 滤波器时间尺度算法

本文提出的单状态变量 Kalman 滤波器时间尺度算法和 NIST 算法的基本原理完全相同, 区别在于 NIST 算法中每台钟有 2 个或 3 个状态变量, 而本文算法中每台钟只有 1 个状态变量, 即原子钟的时差. 通过本节的理论分析和仿真实验, 读者将会看到: 仅仅是这一个改变, 本文算法的性能和 NIST 算法的性能完全不同. 下面将详细对此进行论述.

3.1 算法原理

本节将分 3 个层次描述算法原理: (1) 钟组的状态方程; (2) 钟组的观测方程; (3) 算法步骤与时间尺度 TA 的表达式.

(1) 钟组的状态方程表示为

$$X(k+1) = \Phi \cdot X(k) + J(k). \quad (4)$$

其中, $X(k) = [x_1(k), x_2(k), \dots, x_N(k)]^T$ 是钟组在 k 时刻的时差, $x_i(k) (i=1, 2, \dots, N)$ 是钟组中第 i 台钟的时差. $\Phi = I_{N \times N}$ 是状态转移矩阵, $J(k)$ 是钟组的过程噪声, 表示为 $J(k) = [J_1(k), J_2(k), \dots, J_N(k)]^T$, 其中 $J_i(k) (i=1, 2, \dots, N)$ 是钟组中第 i 台钟的过程噪声. 对比式(2), $J_i(k) (i=1, 2, \dots, N)$ 可以表示为

$$J_i(k) = \sigma_{i,1} (W_{i,1}(t_{k+1}) - W_{i,1}(t_k)) + \sigma_{i,2} \int_{t_k}^{t_{k+1}} (W_{i,2}(t_s) - W_{i,2}(t_k)) ds, \quad (i=1, 2, \dots, N). \quad (5)$$

第 i 台钟的过程状态噪声方差即为式(5)的方差, 表示为^[14]

$$Q_i = \sigma_{i,1}^2 \cdot T + 1/3 \cdot \sigma_{i,2}^2 \cdot T^3. \quad (6)$$

对比式(6)和(3), 得到^[2]

$$Q_i = \sigma_{i,y}^2(T) \cdot T^2, \quad (7)$$

其中, $\sigma_{i,y}^2$ 表示第 i 台钟的 Allan 方差.

于是, 钟组的过程噪声方差表示为

$$Q = \begin{bmatrix} Q_1 & \cdots & 0 \\ \vdots & \ddots & 0 \\ 0 & \cdots & Q_N \end{bmatrix}. \quad (8)$$

(2) 钟组的观测方程表示为

$$\mathbf{Z}(k) = \mathbf{H} \cdot \mathbf{X}(k) + \mathbf{V}(k), \quad (9)$$

其中, $\mathbf{H}$ 表示为

$$\mathbf{H} = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 1 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & -1 \end{bmatrix}_{(N-1) \times N}, \quad (10)$$

$\mathbf{V}$ 是测量噪声, 表示为 $\mathbf{V}(k) = [V_1(k), V_2(k), \dots, V_{N-1}(k)]^T$, 其中 $V_i(k) (i=1, 2, \dots, N-1)$ 是每一组钟差的观测噪声, 它们之间相互独立. $V_i(k)$ 的噪声协方差为 R_i , 所以钟组的测量噪声 $\mathbf{V}$ 的协方差表示为

$$\mathbf{R} = \begin{bmatrix} R_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R_{N-1} \end{bmatrix}. \quad (11)$$

由式(9)和(10), 观测钟差可以表示为

$$\mathbf{Z}(k) = \begin{bmatrix} x_1(k) - x_2(k) + v_1(k) & \cdots \\ x_1(k) - x_N(k) + v_{N-1}(k) \end{bmatrix}^T, \quad (12)$$

其中的每一个量代表在 k 时刻第 1 台钟与第 i 台钟 ($i \neq 1$) 之间的钟差.

(3) 算法步骤与时间尺度 TA 的表达式

Kalman 滤波器通过 $N-1$ 组观测钟差估计出 N 台钟的状态, 记为 $\hat{x}_i(k)$, 其步骤可以由下面 5 个方程来描述.

$$\hat{\mathbf{X}}_{k,k-1} = \boldsymbol{\phi} \cdot \hat{\mathbf{X}}_{k-1,k-1}, \quad (13)$$

$$\mathbf{P}_{k,k-1} = \boldsymbol{\phi} \cdot \mathbf{P}_{k-1,k-1} \cdot \boldsymbol{\phi}^T + \mathbf{Q}, \quad (14)$$

$$\mathbf{K}_k = \mathbf{P}_{k,k-1} \cdot \mathbf{H}^T (\mathbf{H} \cdot \mathbf{P}_{k,k-1} \cdot \mathbf{H}^T + \mathbf{R})^{-1}, \quad (15)$$

$$\hat{\mathbf{X}}_{k,k} = \hat{\mathbf{X}}_{k,k-1} + \mathbf{K}_k \cdot (\mathbf{Z}_k - \mathbf{H} \cdot \hat{\mathbf{X}}_{k,k-1}), \quad (16)$$

$$\mathbf{P}_{k,k} = (\mathbf{I} - \mathbf{K}_k \cdot \mathbf{H}) \cdot \mathbf{P}_{k,k-1}. \quad (17)$$

这 5 个方程中各符号的定义是相关领域内所共知的, 在此不再描述各符号的含义.

$\hat{x}_i(k)$ 的物理意义是第 i 台钟相对于 TA 的时间差, 也就是说, Kalman 估计得到的量是每台钟相对于 TA 的时间差. 把每台钟的时差减去其时差的估计值即为“校准钟”, 于是得到了 N 台“校准钟”. 与 NIST 算法相同, 任何一台“校准钟”都可以被定义为本算法生成的时间尺度

$$\text{TA}(k) = x_i(k) - \hat{x}_i(k), \quad \forall i. \quad (18)$$

单个时间尺度(例如: 式(18)中的 x_i 和 TA)的钟面

读数在实际上是无法获知的. 实际上可以获知的只能是两个时间尺度之间的时间差. 所以, 实际上, TA 通常用它与某台钟之间的时间差来表示, 即

$$\text{TA}(k) - x_j(k) = x_i(k) - x_j(k) - \hat{x}_i(k), \quad \forall i, i \neq j, \quad (19)$$

式中, $x_i - x_j$ 表示第 i 和第 j 台钟之间的时间差, $\hat{x}_i$ 是 Kalman 滤波器估计得到的第 i 台钟相对于 TA 的时间差. 这两个量在实际上是可以获知的, 所以, 由式(19), $\text{TA} - x_j$ 也是在实际上可以获知的.

现在回答一个问题: 为什么很多关于 Kalman 滤波器时间尺度算法的研究(例如: 文献[7-13]等), 都是采用仿真数据, 而不是实测数据进行验证? 原因在于: (1) 实际上只能得到 $\text{TA} - x_j$, 然而 TA 的频率稳定度一般比 x_j 高, 所以无法评价 TA 的频率稳定度在 $\text{TA} - x_j$ 的频率稳定度中的比重; (2) 但是, 仿真时, 可以仿真生成单台钟的钟面读数或单台钟的时差, 通过式(18)可以计算得到 TA 的钟面读数. 因此, 可以研究单台钟、TA 在任意平滑时间的频率稳定度, 以此来分析和改进算法. 综上, 本文中也是采用仿真而不是实测数据来分析和验证算法性能.

3.2 算法的权重与预测值

本节将展示式(18)可以表示成时间尺度基本方程(1), 所以本算法也可以认为是一种加权平均算法.

将式(13)代入式(16), 得到

$$\hat{\mathbf{X}}_{k,k} = \hat{\mathbf{X}}_{k-1,k-1} + \mathbf{K}_k \cdot (\mathbf{Z}_k - \mathbf{H} \cdot \hat{\mathbf{X}}_{k-1,k-1}). \quad (20)$$

式(20)可以写成如下形式:

$$\begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \\ \vdots \\ \hat{x}_N(k) \end{bmatrix} = \begin{bmatrix} \hat{x}_1(k-1) \\ \hat{x}_2(k-1) \\ \vdots \\ \hat{x}_N(k-1) \end{bmatrix} + \begin{bmatrix} K_{11}(k) & K_{12}(k) & \cdots & K_{1(N-1)}(k) \\ K_{21}(k) & K_{22}(k) & \cdots & K_{2(N-1)}(k) \\ \vdots & \vdots & \ddots & \vdots \\ K_{N1}(k) & K_{N2}(k) & \cdots & K_{N(N-1)}(k) \end{bmatrix} \cdot \begin{bmatrix} Z_1(k) \\ Z_2(k) \\ \vdots \\ Z_{N-1}(k) \end{bmatrix} - \begin{bmatrix} \hat{x}_1(k-1) - \hat{x}_2(k-1) \\ \hat{x}_1(k-1) - \hat{x}_3(k-1) \\ \vdots \\ \hat{x}_1(k-1) - \hat{x}_N(k-1) \end{bmatrix}. \quad (21)$$

把式(12)代入式(21), 得到第 1 台“校准钟”表示为

$$\begin{aligned}
x_1(k) - \hat{x}_1(k) = & [1 - K_{11}(k) - K_{12}(k) \cdots - K_{1(N-1)}(k)] \\
& \cdot [x_1(k) - \hat{x}_1(k-1)] \\
& + K_{11}(k) \cdot [x_2(k) - \hat{x}_2(k-1)] \\
& + K_{12}(k) \cdot [x_3(k) - \hat{x}_3(k-1)] + \cdots \\
& + K_{1(N-1)}(k) \cdot [x_N(k) - \hat{x}_N(k-1)] \\
& - \sum_{i=1}^{N-1} K_{1i} v_i(k). \quad (22)
\end{aligned}$$

当观测噪声为零时, 式(22)满足时间尺度基本方程(1), 即

$$TA(k) = x_1(k) - \hat{x}_1(k) = \sum_{j=1}^N \omega_{1,j} \cdot (x_j(k) - x_j'(k)), \quad (23)$$

其中, 权重表示为

$$\omega_{1,1} = 1 - \sum_{j=1}^{N-1} K_{1j}(k), \quad \omega_{1,i} = K_{1(i-1)}(k), \quad (j \neq 1). \quad (24)$$

预测值表示为

$$x_j'(k) = \hat{x}_j(k-1). \quad (25)$$

进一步地, 当测量噪声为零时, 由式(21), 任何一台“校准钟”都满足时间尺度基本方程, 即

$$\begin{aligned}
TA(k) = x_i(k) - \hat{x}_i(k) \\
= \sum_{j=1}^N \omega_{i,j} \cdot [x_j(k) - \hat{x}_j(k-1)]. \quad (26)
\end{aligned}$$

由式(21), 当 $i \neq 1$ 时, 权重表示为

$$\omega_{i,1} = -\sum_{j=1}^{N-1} K_{ij}(k), \quad \omega_{i,i} = 1 + K_{i(i-1)}(k), \quad (27)$$

$$\omega_{i,j} = K_{i(j-1)}(k), \quad (j \neq 1, j \neq i).$$

而预测值的表达式与式(25)相同.

3.3 权重的解析表达式推导

第 3.2 节给出了由 Kalman 增益表达的权重, 但是由该表达式无法解析地分析算法性能. 在本节中, 将进一步推导在观测噪声为零时, 由原子钟噪声特性表达的权重的解析表达式.

本文把式(4)和(9)定义的系统记为系统 1. 按照文献[16]的方法进行验证, 系统 1 是不完全可观测的, 这将导致其估计误差协方差矩阵发散. 这从直观上很容易理解: 该系统需要从 $N-1$ 组观测钟差中估计出 N 台钟的状态, 系统显然是不完全可观测的. 本文另外引入一个完全可观测的系统, 记为系统 2, 其估计误差协方差矩阵收敛. 系统 2 从 $N-1$ 组含观测噪声的观测钟差中估计出 $N-1$ 组观测钟差的状态. 其状

态方程表示为

$$\begin{aligned}
& \begin{bmatrix} x_1(k+1) - x_2(k+1) \\ x_1(k+1) - x_3(k+1) \\ \vdots \\ x_1(k+1) - x_N(k+1) \end{bmatrix} \\
& = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1(k) - x_2(k) \\ x_1(k) - x_3(k) \\ \vdots \\ x_1(k) - x_N(k) \end{bmatrix} + \begin{bmatrix} J_1(k) - J_2(k) \\ J_1(k) - J_3(k) \\ \vdots \\ J_1(k) - J_N(k) \end{bmatrix}. \quad (28)
\end{aligned}$$

其测量方程表示为

$$\begin{aligned}
\mathbf{Z}(k) = & \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1(k) - x_2(k) \\ x_1(k) - x_3(k) \\ \vdots \\ x_1(k) - x_N(k) \end{bmatrix} \\
& + \begin{bmatrix} v_1(k) & & & \\ & v_2(k) & & \\ & & \ddots & \\ & & & v_{N-1}(k) \end{bmatrix}. \quad (29)
\end{aligned}$$

把系统 2 的估计误差协方差矩阵记为 $\tilde{\mathbf{P}}_{k,k}$. 对比式(4)和(28), $\tilde{\mathbf{P}}_{k,k}$ 与系统 1 的估计误差协方差矩阵的关系如式(30)所示

$$\mathbf{H} \cdot \mathbf{P}_{k,k} \cdot \mathbf{H}^T = \tilde{\mathbf{P}}_{k,k}. \quad (30)$$

因为 $\tilde{\mathbf{P}}_{k,k}$ 收敛, 所以 $\mathbf{H} \cdot \mathbf{P}_{k,k} \cdot \mathbf{H}^T$ 收敛(尽管 $\mathbf{P}_{k,k}$ 发散). 把 $\mathbf{H} \cdot \mathbf{P}_{k,k} \cdot \mathbf{H}^T$ 的稳态值记为 $\mathbf{Ps}$.

本文引入系统 2 即为了得出式(30)所示的结论. 下文所有推导将回到系统 1 来完成.

把 $\mathbf{P}_{k,k}'$ 定义为

$$\mathbf{P}_{k,k}' = \mathbf{P}_{k,k} + \mathbf{Q}. \quad (31)$$

因为 $\mathbf{H} \cdot \mathbf{P}_{k,k} \cdot \mathbf{H}^T$ 收敛, 所以 $\mathbf{H} \cdot \mathbf{P}_{k,k}' \cdot \mathbf{H}^T$ 收敛.

$\mathbf{H} \cdot \mathbf{P}_{k,k}' \cdot \mathbf{H}^T$ 的稳态值表示为

$$\begin{aligned}
\mathbf{Ps}' = & \mathbf{Ps} + \mathbf{H} \cdot \mathbf{Q} \cdot \mathbf{H}^T \\
= & \mathbf{Ps} + \begin{bmatrix} Q_1 + Q_2 & Q_1 & \cdots & Q_1 \\ Q_1 & Q_1 + Q_3 & \cdots & Q_1 \\ \vdots & \vdots & \ddots & \vdots \\ Q_1 & Q_1 & \cdots & Q_1 + Q_N \end{bmatrix}. \quad (32)
\end{aligned}$$

由式(15)得到

$$H \cdot K_k = H \cdot P_{k-1,k-1}' \cdot H^T \cdot (H \cdot P_{k-1,k-1}' \cdot H^T + R)^{-1}. \quad (33)$$

因为 $H \cdot P_{k,k}' \cdot H^T$ 收敛, 所以 $H \cdot K_k$ 也收敛. 把 $H \cdot K_k$ 的稳态值记为 HKs . 当观测噪声为零时, 有 $HKs=I$.

由式(14)和(17), 得到

$$P_{k,k} = (I - K_k \cdot H) \cdot (\phi \cdot P_{k-1,k-1} \cdot \phi^T + Q). \quad (34)$$

由式(34)得到

$$H \cdot P_{k,k} = (I - H \cdot K_k) \cdot (H \cdot P_{k-1,k-1} + H \cdot Q). \quad (35)$$

把 $H \cdot P_{k,k}$ 的稳态值记为 HPs . 将 $HKs=I$ 代入式(35), 得到 $H \cdot P_{k,k}$ 收敛于 $HPs=0$. 进一步可以得到, $H \cdot P_{k,k} \cdot H^T$ 收敛于 $Ps=0$. 于是, 由式(32)得到

$$Ps' = \begin{bmatrix} Q_1 + Q_2 & Q_1 & \cdots & Q_1 \\ Q_1 & Q_1 + Q_3 & \cdots & Q_1 \\ \vdots & \vdots & \ddots & \vdots \\ Q_1 & Q_1 & \cdots & Q_1 + Q_N \end{bmatrix}. \quad (36)$$

由 $H \cdot P_{k,k}$ 收敛得到 $H \cdot P_{k,k}'$ 也收敛, 由式(31)得到其稳态值为 $H \cdot Q$.

因为 $H \cdot P_{k,k}' \cdot H^T$ 和 $H \cdot P_{k,k}'$ 收敛, 由式(15)得出 K_k 收敛, 其稳态值为

$$Ks = (HPs')^T \cdot Ps'^{-1} = Q \cdot H^T \cdot Ps'^{-1}. \quad (37)$$

把式(36)代入式(37), 得到

$$Ks = Q \cdot H^T \cdot Ps'^{-1} = \begin{bmatrix} Q_1 & Q_1 & \cdots & Q_1 \\ -Q_2 & 0 & \cdots & 0 \\ 0 & -Q_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -Q_N \end{bmatrix} \cdot \begin{bmatrix} Q_1 + Q_2 & Q_1 & \cdots & Q_1 \\ Q_1 & Q_1 + Q_3 & \cdots & Q_1 \\ \vdots & \vdots & \ddots & \vdots \\ Q_1 & Q_1 & \cdots & Q_1 + Q_N \end{bmatrix}^{-1}. \quad (38)$$

由式(38), 利用 Mathematic 软件, 首先得出 Ks 第一行的表达式

$$Ks_{11} = \frac{1/Q_2}{\sum_{j=1}^N 1/Q_j} Ks_{12} = \frac{1/Q_3}{\sum_{j=1}^N 1/Q_j} \cdots Ks_{1(N-1)} = \frac{1/Q_N}{\sum_{j=1}^N 1/Q_j}. \quad (39)$$

把式(39)代入式(24), 得到

$$\omega_{1,j} = \frac{1/Q_j}{\sum_{j=1}^N 1/Q_j}. \quad (40)$$

然后, 用同样的方法得到 Ks 第 $i(i \neq 1)$ 行的表达式, 并代入式(27)得到

$$\omega_{i,j} = \frac{1/Q_j}{\sum_{j=1}^N 1/Q_j}, (\forall i, i \neq 1). \quad (41)$$

将式(40)和(41)代入式(26), 得到当测量噪声为零, 且 Kalman 增益进入稳态时, TA 表示为

$$\begin{aligned} TA(k) &= x_i(k) - \hat{x}_i(k), (\forall i) \\ &= \sum_{j=1}^N \omega_{i,j} \cdot [x_j(k) - \hat{x}_j(k-1)], (\forall i) \\ &= \sum_{j=1}^N \frac{1/Q_j}{\sum_{j=1}^N 1/Q_j} \cdot [x_j(k) - \hat{x}_j(k-1)]. \end{aligned} \quad (42)$$

进一步地, 把式(7)代入式(42), 得到

$$\begin{aligned} TA(k) &= x_i(k) - \hat{x}_i(k), (\forall i) \\ &= \sum_{j=1}^N \omega_{i,j} \cdot [x_j(k) - \hat{x}_j(k-1)], (\forall i) \\ &= \sum_{j=1}^N \frac{1/\sigma_{j,y}^2(T)}{\sum_{j=1}^N 1/\sigma_{j,y}^2(T)} \cdot [x_j(k) - \hat{x}_j(k-1)]. \end{aligned} \quad (43)$$

式(43)表明: 得到当测量噪声为零, 且 Kalman 增益进入稳态时, 有: (1) TA 可以表示为时间尺度基本方程(1); (2) 由不同“校准钟”表达的 TA 都相等; (3) 权重反比于平滑时间为 Kalman 滤波器采样间隔 T 时的 Allan 方差.

3.4 虚拟 Kalman 采样间隔

本节将在第 3.3 节得到的权重的解析表达式(41)的基础上, 进一步对算法进行优化.

式(43)表明: 本算法只能使 TA 在平滑时间为 T 的频率稳定度达到最优, 而不能保证 TA 在任意指定的平滑时间的频率稳定度最优.

为了进一步优化该算法, 提出了虚拟 Kalman 采样间隔 T_v 的概念, 用于计算得到每台钟的虚拟状态噪声协方差, 表示为

$$Qv_i = \sigma_{i,1}^2 \cdot T_v + 1/3 \cdot \sigma_{i,2}^2 \cdot T_v^3, \forall i(i=1, \cdots, N). \quad (44)$$

按照本文第 3.3 节的推导, 算法其他步骤不变,

TA 将表示为

$$\begin{aligned} \text{TA}(k) &= x_i(k) - \hat{x}_i(k), (\forall i) \\ &= \sum_{j=1}^N \frac{1/Qv_j}{\sum_{j=1}^N 1/Qv_j} \cdot [x_j(k) - \hat{x}_j(k-1)] \\ &= \sum_{j=1}^N \frac{1/\sigma_{j,y}^2(T_v)}{\sum_{j=1}^N 1/\sigma_{j,y}^2(T_v)} \cdot [x_j(k) - \hat{x}_j(k-1)]. \end{aligned} \quad (45)$$

式(45)表明: 在采用虚拟 Kalman 采样间隔 T_v 时, 算法可以使 TA 在平滑时间为 T_v 的频率稳定度达到最优. 因此, 在实际运用中, 可以直接选取一个 T_v 的值, 这样生成的 TA 就在指定的平滑时间 T_v 的频率稳定度达到了最优. 比照第 3.3 节的理论推导, 此时实际的 Kalman 滤波器采样间隔 T 不会对算法性能有任何影响.

3.5 仿真实验

首先, 使用文献[17]的方法仿真生成 2 台氢钟和 1 台铯钟组成钟组(每台钟共 8000 个数据点, 相邻数据点之间的采样间隔为 14400 s). 每台氢钟如式(2)所示的平方扩散系数设为 $\sigma_1^2=5.0 \times 10^{-25} \text{ s}$ 和 $\sigma_2^2=3.0 \times 10^{-35} \text{ s}^{-1}$, 铯钟的平方扩散系数设为 $\sigma_1^2=4.8 \times 10^{-23} \text{ s}$ 和 $\sigma_2^2=1.0 \times 10^{-36} \text{ s}^{-1}$. 这些扩散系数符合典型氢钟和铯钟的指标. 设置观测噪声为零. 为了简化分析, 设每台钟如式(2)所示的确定性分量为 $x_0=y_0=d=0$.

然后, 按照第 3.1 节, 使用本算法生成 TA. 分为两种情况. 第一种情况, 设置虚拟 Kalman 采样间隔 $T_v=5 \text{ d}$; 第二种情况, 设置 $T_v=100 \text{ d}$. 按照本文的分

析, 当观测噪声为零时, 所有“校准钟”都是相等的, 所以不妨把第 1 台“校准钟”作为 TA.

表 1 列出了在这两种情况下, 由式(45)计算得到的每台钟权重的理论值. 图 1 描绘了在第一种情况下, 3 台钟和 TA 的时差(a)、Allan 偏差(Allan 方差的平方根)(b). 图 2 描绘了在第二种情况下, 它们的时差(a)和 Allan 偏差(b).

由表 1、图 1 和 2 可以看出: 在 T_v 较小时, 氢钟的权重较大, 于是 TA 主要由两台氢钟决定. 而在 T_v 较大时, 铯钟的权重较大, 于是 TA 主要由铯钟决定.

综上, 表 1 和图 1-4 验证了本文理论分析的结论, 即: 本算法本质上也是一种加权平均算法; 当测量噪声为零时, 权重反比于每台钟在平滑时间为 T_v 时的 Allan 偏差. 所以, 可以根据实际需求, 灵活地设置 T_v 的值, 使 TA 在该平滑间隔的频率稳定度达到最优.

3.6 算法总结

本算法与 NIST 算法的唯一区别在于 NIST 算法中每台钟有 2 个或 3 个状态变量, 而本算法只有 1 个状态变量. 但是这两种算法性能差异极大. NIST 算法生成的 TA 的频率稳定度主要由长期频率稳定度最好的钟决定, 因此算法可以优化 TA 的长期频率稳定度. 本算法生成的 TA 的权重却反比于平滑时间为虚拟

表 1 两种情况下每台钟的权重

Table 1 The weights of each clock in two cases

	H1	H2	Cs
情况 1	0.4892	0.4892	0.0215
情况 2	0.0817	0.0817	0.8367

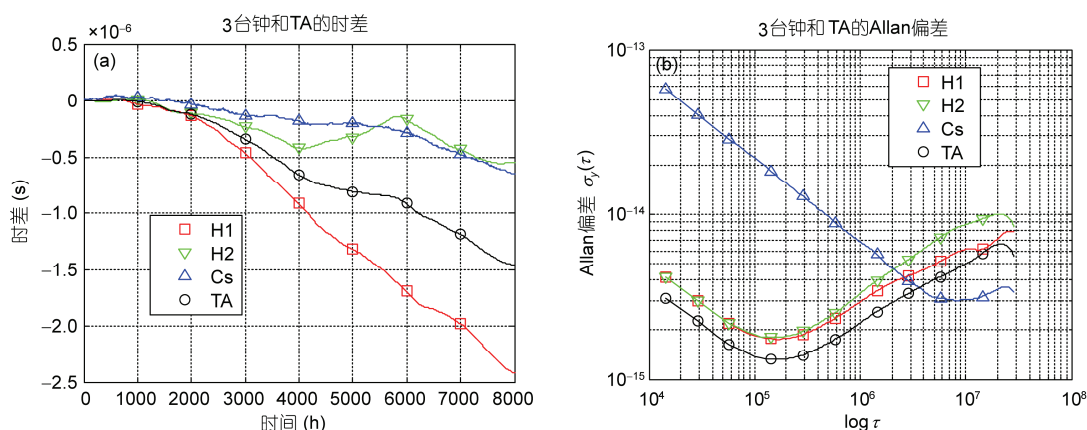


图 1 (网络版彩图)在第一种情况下, 每台钟和 TA 的时差、Allan 偏差. (a) 时差; (b) Allan 偏差

Figure 1 (Color online) Time deviations and Allan deviations of each clock and TA in the first case. (a) Time deviation; (b) Allan deviation.

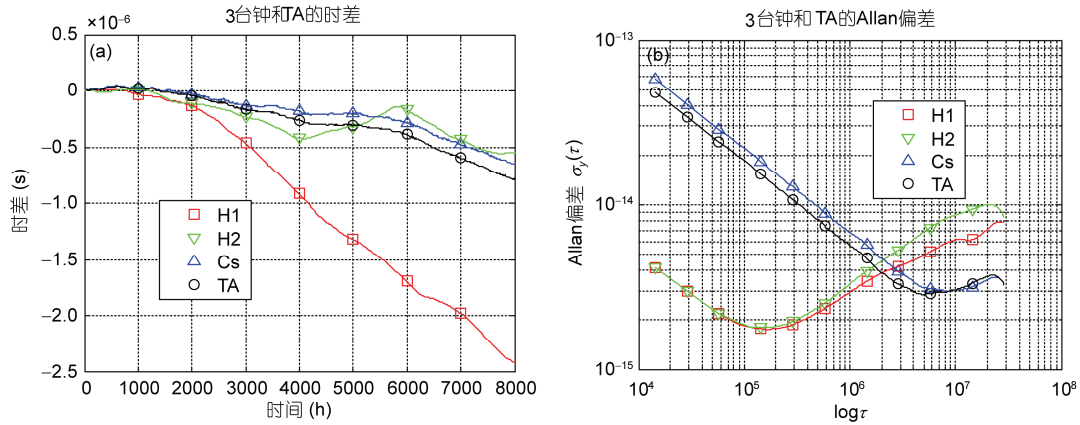


图 2 (网络版彩图)在第二种情况下, 每台钟、TA 的时差和 Allan 偏差. (a) 时差; (b) Allan 偏差

Figure 2 (Color online) Time deviations and Allan deviations of each clock and TA in the second case. (a) Time deviation; (b) Allan deviation.

Kalman 采样间隔 T_v 时的 Allan 方差. 通过改变 T_v 的值, 可以使 TA 在任意指定的平滑时间的频率稳定度达到最优.

本算法和 NIST 算法都可以认为是一种加权平均算法. 如第 3.2 节所示, 通过运行 Kalman 滤波器, 算法自动计算得到了权重和预测值. 然而对于 ALGOS 或是 AT1 等传统加权平均算法, 确定权重和预测值的步骤是比较复杂的(可以参见文献[4]), 而且一般都是先确定权重, 然后再确定预测值; 确定预测值的时候都要使用钟差参数估计和钟差预测算法^[18]. 所以, 本算法相比传统加权平均算法, 实现更为简洁方便.

4 两级 Kalman 滤波器时间尺度算法

本节在第 3 节所描述的单状态变量 Kalman 滤波器时间尺度算法的基础上, 进一步提出一种两级 Kalman 滤波器时间尺度算法. 本算法生成的 TA 相比第 3 节的 TA, 中短期频率稳定度将更高, 因为本算法生成的 TA 中只含有 RWFM, 不含有 WFM. 下面将详细论述该算法.

4.1 算法原理与分析

两级 Kalman 滤波器时间尺度算法的原理如图 3 所示.

第一级 Kalman 滤波器共有 $N-1$ 个, 其输入量分别为 $N-1$ 组钟差观测值(如式(12)所示). 把其中每组钟差分别记为 $Z^{(1,n)}(k)$, 其中上标(1,n)表示第 1 台钟与第 n 台钟之间的钟差, 而第 i 台钟的时差依然如第 3 节表示方法相同, 记为 $x_i(k)$. 如果没有观测噪声, 显

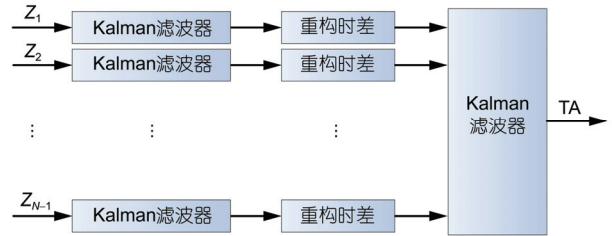


图 3 (网络版彩图)两级 Kalman 滤波器算法原理

Figure 3 (Color online) Principle of the twice Kalman filters time scale algorithm.

然有 $Z^{(1,n)}(k) = x_1(k) - x_n(k)$.

这 $N-1$ 个 Kalman 滤波器分别对这 $N-1$ 组钟差进行状态估计, 得到每组钟差的如式(2)所示的两个状态的估计量, 记为 $\hat{X}_1^{(1,n)}(k)$ 和 $\hat{X}_2^{(1,n)}(k)$. 由式(2)可知, $\hat{X}_1^{(1,n)}(k)$ 时间序列中的噪声为 WFM 和 RWFM, 而 $\hat{X}_2^{(1,n)}(k)$ 时间序列中将只含有 RWFM.

然后, 对时差进行重构.

对于每一组估计值, 令

$$\tilde{X}_1^{(1,n)}(k+1) = \tilde{X}_1^{(1,n)}(k) + \hat{X}_2^{(1,n)}(k) \cdot T. \quad (46)$$

把 $\tilde{X}_1^{(1,n)}(t_k)$ 序列的初始值取为零, 上式可以改写为

$$\tilde{X}_1^{(1,n)}(k) = \sum_{m=0}^{k-1} \hat{X}_2^{(1,n)}(m) \cdot T. \quad (47)$$

于是, 式(47)就是通过估计量 $\hat{X}_2^{(1,n)}(k)$ 进行重构, 得到的第 1 台钟和第 n 台钟之间的“重构时差” $\tilde{X}_1^{(1,n)}(k)$. 显然, “重构时差” $\tilde{X}_1^{(1,n)}(k)$ 中也只含有 RWFM.

把第 i 台钟的“重构钟”的时差记为 $\tilde{x}_i(k)$, 于是有 $\tilde{X}_1^{(1,n)}(k) = \tilde{x}_1(k) - \tilde{x}_n(k)$. 需要说明的是: 单台“重构钟”的时差 $\tilde{x}_i(k)$ 在实际上是无法观测得到的, 实际上能得到的只有两台“重构钟”之间的时间差, 即: “重构时差” $\tilde{X}_1^{(1,n)}(k)$.

因此, 通过第一级 Kalman 滤波器和重构时差这两个环节, 相当于滤除了输入时间序列(即观测钟差)中的 WFM, 得到了只含有 RWFM 的“重构时差”. 能达到这样的效果, 是利用了以下特性: Kalman 滤波器用于状态估计时, 按照原子钟模型(2), 估计得到的第 1 个状态变量中含有 WFM 和 RWFM, 而第 2 个状态变量中只含有 RWFM.

最后, 第二级 Kalman 滤波器是一个本文第 3 节提出的单状态变量 Kalman 滤波器, 用于生成 TA. 该 Kalman 滤波器的输入为 $N-1$ 组“重构时差” $\tilde{X}_1^{(1,n)}(k)$, 输出为 N 台“重构钟”的时差估计值, 记为 $\hat{\tilde{x}}_i(k)$, 它代表了第 i 台“重构钟”的时差 $\tilde{x}_i(k)$ 与 TA 之间的偏差, 于是得到

$$\text{TA}(k) = \tilde{x}_i(k) - \hat{\tilde{x}}_i(k), \forall i. \quad (48)$$

按照第 3 节的分析, 该 Kalman 滤波器相当于对 N 个只含有 RWFM 的“重构钟”进行加权平均, 所以 TA 中也只含有 RWFM. 相比于第 3 节算法生成的 TA, 本算法生成的 TA 的中短期频率稳定度更高.

此时, 设置虚拟 Kalman 采样间隔 T_v 的值对 TA 的影响不大, 因为: (1) 第 3 节的算法中, Kalman 滤波器的输入的噪声为 WFM 和 RWFM, 它们的 Allan 偏差曲线的斜率分别为 -1 和 $+1$, 形状为 V 字型. 这样取不同的 T_v 值时, 不同钟的权重将会不同; (2) 然而本节算法中, 第二级 Kalman 滤波器的输入噪声为 RWFM, 它们的 Allan 偏差曲线的斜率为 $+1$, 不管 T_v 如何改变, 权重将都反比于平方扩散系数 σ_2^2 , 此算法相当于按照 RWFM 的强度进行加权.

4.2 仿真实验

本节中, 采用与第 3.5 节相同的 2 台氢钟和 1 台铯钟组成钟组.

实际应用中的步骤应该为: 首先使用第一级的 Kalman 滤波器, 并对时差进行重构, 获得 $N-1$ 组“重构时差”, 即: H1-H2 和 H1-Cs 的“重构时差”; 然后, 运行如第二级的 Kalman 滤波器, 得到的估计量为每

台“重构钟”与 TA 之间的时间差.

和第 3 节相同, 实际应用中, 单台“重构钟”、TA 的时差是无法获知的, 所能获知的只能是两台“重构钟”之间的时间差, 即“重构时差”, 以及每台“重构钟”与 TA 之间的时间差. 但是, 为了更清晰地展示每台钟对 TA 的影响, 在本节的仿真实验中, 依然假设单台钟的时差是已知的.

所以, 与实际应用中的步骤不同, 在本次仿真中, 首先使用第一级的 3 个 Kalman 滤波器, 分别对 3 台钟进行状态估计, 对时差进行重构, 获得 3 台“重构钟”的时差, 显然也就得到了 2 组“重构时差”. 然后, 运行第二级的 Kalman 滤波器, 其输入为两组“重构时差”, 输出为估计得到的每台“重构钟”与 TA 之间的时间差, 于是也就生成了 TA. 其中, 设置虚拟 Kalman 采样间隔 $T_v=5$ d.

通过仿真得到每台“重构钟”的权重的分别为: 两台氢钟都为 0.0312, 铯钟为 0.9375. 于是可以看出: “重构铯钟”获得的权重更大, 这与理论分析的结果一致, 即权重反比于 σ_2^2 的值. 图 4 描述了每台“重构钟”、TA 的时差(a)和 Allan 偏差(b). 从图 4 中可以看出: TA 中也只含有 RWFM, 它的中短期频率稳定度相比第 3 节生成的 TA 得到了明显的提高.

4.3 算法总结

两级 Kalman 滤波器时间尺度, 相当于先对钟差滤波, 以消除 WFM, 以后的计算方法和单状态变量 Kalman 滤波时间尺度算法相同. 实际上, 任何滤波的目的都是为了滤除高频噪声. WFM 相比 RWFM 是高频噪声, 因此采用其他滤波器同样可以滤除 WFM. 但是, 为了完全滤除 WFM, 而又不至于滤除 RWFM, 滤波器结构、带宽、参数的设计和选取并不容易. 本算法的优势在于充分利用了原子钟模型(2)的特性, 即第 2 个状态变量中只含有 RWFM. 这样, 只使用一个简单的 Kalman 滤波器, 就可以完全滤除 WFM. 算法设计相对比较简单, 并且物理含义清晰.

5 结束语

本文提出了两种时间尺度算法: (1) 单状态变量 Kalman 滤波器时间尺度算法; (2) 两级 Kalman 滤波器时间尺度算法. 其中, 后一种算法是在前一种算法的基础上提出的, 它的第二级的 Kalman 滤波器实际

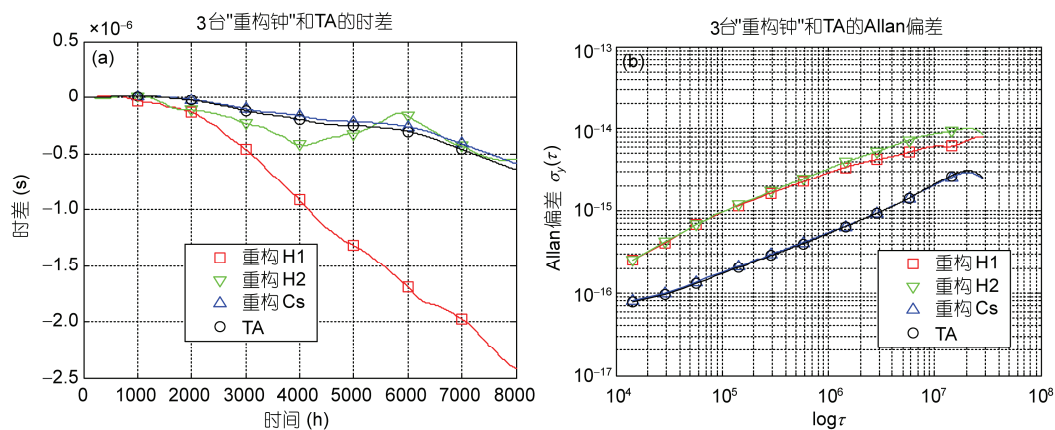


图 4 (网络版彩色)每台“重构钟”、TA 的时差和 Allan 偏差. (a) 时差; (b) Allan 偏差

Figure 4 (Color online) Time deviations and Allan deviations of each clock and TA. (a) Time deviation; (b) Allan deviation.

上就是前一种算法. 本文详细描述了这两种算法的原理, 通过严密的数学推导, 给出算法完整的理论框架, 通过仿真实验验证了理论分析的结论.

下一步, 本文作者期望能借鉴本文的研究思路, 推导 NIST 算法(即在每台钟的状态变量有 2 个或 3

个的情况下)的权重和预测值的解析表达式, 从理论上解释为什么 NIST 的 Kalman 滤波器算法只优化了时间尺度的长期稳定度. 我们认为: 对该问题进行研究可以加深对一系列的 Kalman 滤波器时间尺度算法的本质理解, 以便于更好地设计时间尺度算法.

参考文献

- Peter B W, John A D, Setnam L S. Local representations of UTC in national laboratories. *Metrologia*, 2011, 48: 154–164
- Patrizia T. Statistical and mathematical tools for atomic clocks. *Metrologia*, 2008, 45: 183–192
- Lorenzo G, Patrizia T. Time and Kalman filter. *IEEE Control Syst*, 2010, 30: 44–65
- Tavella P, Thomas C. Comparative study of time scale algorithms. *Metrologia*, 1991, 28: 57–63
- Gianna P, Arias E F. Algorithms for international atomic time. *IEEE Trans Ultra Ferroelect Freq Control*, 2010, 55: 140–150
- Weiss M, Weissert T. AT2, a new time scale algorithm: AT1 plus frequency variance. *Metrologia*, 1991, 28: 65–74
- Richard H J, Peter V T. Estimating time from atomic clocks. *J Res Natl Bureau Stand*, 1983, 88: 17–24
- Weiss M A, Allan D W, Trudi K P. A study of the NBS time scale algorithm. *IEEE Trans Instrum Meas*, 1989, 38: 631–635
- Charles A G. Forming stable timescales from the Jones-Tryon Kalman filter. *Metrologia*, 2003, 40: 335–341
- Brown K R. The theory of the GPS composite clock. ION-GPS-91. In: *Proceedings of the 4th International Technical Meeting of the Satellite Division of the Institute of Navigation*. Albuquerque, 1991. 91: 223–241
- Charles A G. A Kalman filter clock ensemble algorithm that admits measurement noise. *Metrologia*, 2006, 43: 311–321
- Matthias S, Charles A G. Congruence of two Kalman filter composite clocks. *Metrologia*, 2013, 50: 499–508
- Matthias S, Charles A G. Combined covariance reductions for Kalman filter composite clocks. *Metrologia*, 2012, 49: 588–596
- Cristina Z, Patrizia T. The clock model and its relationship with the Allan and related variances. *IEEE Trans Ultra Ferroelect Freq Control*, 2005, 52: 289–296
- Galleani L, Sacerdote L, Tavella P, et al. A mathematical model for the atomic clock error. *Metrologia*, 2003, 40: S257–S264
- 邓自立. 卡尔曼滤波与维纳滤波——现代时间序列分析方法. 哈尔滨: 哈尔滨工业大学出版社, 2001
- Kasdin N J. Discrete simulation of colored noise and stochastic processes and $1/f$ power law noise generation. *Proc IEEE*, 1995, 83: 802–827
- Wu Y W, Zhu X W, Huang Y B, et al. Uncertainty derivation and performance analyses of clock prediction based on mathematical model method. *IEEE Trans Instrum Meas*, 2015, 64: 2792–2801

One-state and twice Kalman filter time scale algorithms

WU YiWei*, GONG Hang, ZHU XiangWei & OU Gang

School of Electronics Science and Technology, National University of Defense Technology, Changsha 410073, China

We first propose a one-state Kalman filter time scale algorithm, and derive the analytical expressions of the weights and the predictions. Based on the analytical expressions, we introduce the virtual Kalman sampling time. The theoretical analyses and the simulations both validate that we can optimize the frequency stability on any one of the certain observation intervals of the time scale by means of choosing a certain virtual Kalman sampling time. Then, based on this algorithm, we propose a twice Kalman filter time scale algorithm, and describe the principle of the algorithm. The forming time scale only involves walk random frequency modulation noise (RWFM), and does not involve white frequency modulation noise (WFM). The weights are in inverse proportion to the intense of RWFM. The short-term and middle-term frequency stability of the time scale is higher.

time scale algorithm, Kalman filter, virtual Kalman sampling time, weight, frequency stability

PACS: 06.20.Dk, 06.30.Ft, 93.85.Bc

doi: 10.1360/SSPMA2015-00590