

自适应卡尔曼滤波在计算机 继电保护中的应用

华中理工大学 张 哲 陈德树

摘 要

本文基于极大似然估计理论及虚拟噪声补偿技术,提出了关于电压、电流噪声参数的一种在线优化估计方法,据此可构成自适应卡尔曼滤波器,以对保护所需的基频电压、电流分量进行“最优”估计,该算法无需象常规卡尔曼滤波器那样,需事先对电压、电流噪声进行复杂的统计分析以求取滤波算法所需的噪声参数。因此,该算法使用起来将更为灵活、方便,此外,算法的收敛速度较快,噪声参数的估计计算简便,易于在微机保护中加以采用。

一 前 言

为了快速、准确地计算故障暂态量中的基频分量,近十年来,一些基于卡尔曼滤波理论的新算法已引入微机继电保护领域之中,就目前所采用的卡尔曼滤波算法而言,存在的主要困难之一在于如何确定算法所需的噪声统计参数,从理论上来说,这些噪声参数的确定有赖于对不同故障点、不同故障类型及不同故障时刻时的系统故障情况进行统计计算,这通常是十分复杂而又困难的。本文从极大似然估计理论和虚拟噪声补偿技术出发,提出了一种在线优化估计噪声参数的自适应滤波算法,并利用动模实验的采样数据对此算法进行了校验,检验结果表明,该算法收敛较快,实时计算简便,具有一定的实用价值。

二 卡尔曼滤波算法简介

关于卡尔曼滤波算法在计算机继电保护中的应用在许多文献中进行了详细讨论,其基本结论如下,具体推导过程请参见文献[1~3]。电压状态方程:

$$\begin{pmatrix} U_r(k+1) \\ U_l(k+1) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} U_r(k) \\ U_l(k) \end{pmatrix} \quad (1)$$

或写成矩阵形式:

$$U(k+1) = \Phi_U(k+1/k) U(k)$$

电压测量方程:

$$Z_U(k) = [\cos \omega k \Delta T, -\sin \omega k \Delta T] \begin{pmatrix} U_r(k) \\ U_i(k) \end{pmatrix} + N_U(k) \quad (2)$$

或: $Z_U(k) = H_U(k) U(k) + N_U(k)$

式中:

$U_r(k)$ 、 $U_i(k)$ 分别为基频电压的实部和虚部

$Z_U(k)$ 为电压测量(采样)值;

$N_U(k)$ 为电压测量信号中的噪声分量。

在电压测量方程中, 噪声 $N_U(k)$ 可近似认为是具有正态分布的零均值白噪声, 其方差依指数规律衰减, 即:

$$R_{U0}(k) = R_{U0} \cdot \exp(-k\Delta T/T_U)$$

其中 R_{U0} 、 T_U 分别为电压噪声方差的起始值和衰减时间常数。

电流状态方程:

$$\begin{pmatrix} I_r(k+1) \\ I_i(k+1) \\ I_d(k+1) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \exp(-\Delta T/T_d) \end{pmatrix} \begin{pmatrix} I_r(k) \\ I_i(k) \\ I_d(k) \end{pmatrix} \quad (3)$$

或: $I(k+1) = \Phi_I(k+1/k) I(k)$

电流测量方程:

$$Z_I(k) = [\cos \omega k \Delta T, -\sin \omega k \Delta T, 1] \begin{pmatrix} I_r(k) \\ I_i(k) \\ I_d(k) \end{pmatrix} + N_I(k) \quad (4)$$

或: $Z_I(k) = H_I(k) \cdot I(k) + N_I(k)$

式中: —

$I_r(k)$ 、 $I_i(k)$ 分别为基频电流的实部和虚部;

$I_d(k)$ 为电流噪声信号中非周期分量瞬时值;

T_d 为非周期分量衰减时间常数;

$N_I(k)$ 为电流噪声中的白噪声分量。

在电流测量方程中, 噪声 $N_I(k)$ 近似为具有正态分布的零均值白噪声, 其方差为,

$$R_{I0}(k) = R_{I0} \cdot \exp(-k\Delta T/T_I)$$

其中: R_{I0} 、 T_I 分别为电流噪声方差的起始值和衰减时间常数。

基频电压估计方程:

预测估计: $\hat{U}(k+1/k) = \Phi_U(k+1/k) \cdot \hat{V}(k)$

预测估计误差协方差:

$$P_U(k+1/k) = \Phi_U(k+1/k) P_U(k) \Phi_U^T(k+1/k)$$

增益矩阵计算:

$$K_U(k+1) = P_U(k+1/k) H_U^T(k+1) [H_U(k+1) P_U(k+1/k) H_U^T(k+1) + R_U(k+1)]^{-1}$$

预测估计修正:

$$\hat{U}(k+1) = \hat{U}(k+1/k) + K_U(k+1) \cdot [Z_U(k+1) - H_U(k+1) \hat{U}(k+1/k)]$$

误差协方差修正:

$$P_U(k+1) = [1 - K_U(k+1) \cdot H_U(k+1)] P_U(k+1/k)$$

初始条件: 给定电压初始估计向量: $\hat{U}(0)$

初始估计误差协方差矩阵: $P_U(0)$ 。

基频电流分量的估计方程与电压估计方程相似, 这里从略。

从上面所述的电压、电流递推估计方程可知, 当给定电压、电流的初始估计条件后, 递推估计的计算取决于方程中噪声参数 $R_U(k)$ 、 $R_I(k)$ 及非周期电流衰减时间常数 T_d 等的确定。从理论上来说, 要求得这些统计参数需对不同故障点, 不同故障类型及不同故障时刻时系统故障情况进行统计分析, 这通常是十分困难的。因此, 更为简便、灵活的方法是利用测量信号对这些噪声统计参数进行在线优化估计, 构成所谓的自适应式卡尔曼滤波器, 下面将基于极大似然估计理论及虚拟噪声补偿技术, 给出噪声参数的一种在线优化估计算法。

三 电压、电流噪声参数的在线估计

1. 电压噪声参数的估计

电压噪声分量可假设为是具有零均值、正态分布的白噪声, 其方差为:

$$R_U(k) = R_{U0} \cdot \exp(-k\Delta T/T_U)$$

因此, 电压噪声参数的估计将转化成对参数 $R_{U0} \cdot T_U$ 的估计。

设已知观测值集 $S(k) = \{Z_U(k), Z_U(k-1), \dots, Z_U(1)\}$

并假定参数 R_{U0} 、 T_U 的优化估计为 \hat{R}_{U0} 、 \hat{T}_U

选择概率密度函数 J 为优化估计似然函数

$$J = P\{R_{U0}, T_U / S(k)\}$$

则优化估计值 \hat{R}_{U0} 、 \hat{T}_U 将满足:

$$J = P\{\hat{R}_{U0}, \hat{T}_U / S(k)\} \Rightarrow \max$$

由贝叶斯公式有:

$$P\{R_{U0}, T_U/S(k)\} = \frac{P\{R_{U0}, T_U, S(k)\}}{P\{S(k)\}}$$

$$= \frac{P\{S(k)/R_{U0}, T_U\}}{P\{S(k)\}} \cdot P\{R_{U0}, T_U\}$$

由于 $P\{S(k)\}$ 与参数的优化估计无关, 可略去。

此外, 考虑到随机变量 R_{U0} 、 T_U 的相互独立性, 有:

$$P\{R_{U0}, T_U\} = P\{R_{U0}\} \cdot P\{T_U\}$$

由于在一般情况下, 概率密度 $P\{R_{U0}\}$ 、 $P\{T_U\}$ 事先难以确定, 从而给参数的优化估计带来一定困难, 一种简便的处理方法是假设 R_{U0} 、 T_U 服从均匀分布, 即:

$$P\{R_{U0}, T_U\} = P\{R_{U0}\} \cdot P\{T_U\} = \text{常数}$$

则似然函数 $P\{\hat{R}_{U0}, \hat{T}_U/S(k)\} \Rightarrow \max$

等值为: $P\{S(k)/\hat{R}_{U0}, \hat{T}_U\} \Rightarrow \max$

因此, 优化估计的似然函数可选择为:

$$J_1 = P\{S(k)/R_{U0}, T_U\}$$

因为: $P\{S(k)/R_{U0}, T_U\}$

$$= P\{Z_U(k)/S(k-1), R_{U0}, T_U\} \cdot P\{S(k-1)/R_{U0}, T_U\}$$

$$= P\{Z_U(k)/S(k-1), R_{U0}, T_U\} \cdot P\{Z_U(k-1)/S(k-2), R_{U0}, T_U\} \cdots P\{Z_U(1)/R_{U0}, T_U\}$$

根据电压测量方程及考虑到电压噪声为正态分布的白噪声, 有:

$$P\{Z_U(j)/S(j-1), R_{U0}, T_U\}$$

$$= \frac{1}{(2\pi)^{\frac{1}{2}} R_U(j)^{\frac{1}{2}}} \exp\left[-\frac{(Z_U(j) - H_U(j)U(j))^2}{2R_U(j)}\right]$$

从而:

$$J_1 = P\{S(K)/R_{U0}, T_U\}$$

$$= C \cdot \left[\prod_{j=1}^K R_U(j)^{-\frac{1}{2}} \exp\left(-\frac{[Z_U(j) - H_U(j)U(j)]^2}{2R_U(j)}\right) \right]$$

令 $\varepsilon(j) = Z_U(j) - H_U(j)U(j)$

$$b = \exp(-\Delta T/T_U)$$

则 $R_U(j) = R_{U0} \cdot \exp(-j\Delta T/T_U) = R_{U0} b^j$

$$J_1 = P\{S(K)/R_{U0}, T_U\}$$

$$= C \cdot R_{U0}^{-\frac{K}{2}} b^{-\frac{1}{2}} \sum_{j=2}^K j \exp\left(-\frac{1}{2R_{U0}} \sum_{j=1}^K \varepsilon^2(j) b^{2j}\right)$$

$$\ln J_1 = -\frac{K}{2} \ln R_{U0} - \frac{1}{2} \sum_{j=1}^K j \ln b - \frac{1}{2R_{U0}} \sum_{j=1}^K \varepsilon^2(j) b^{2j} + \text{constant}$$

注意到 J_1 与 $\ln J_1$ 具有相同的极值点, 即:

$J_1 \Rightarrow \max$ 等值于 $\ln J_1 \Rightarrow \max$

而:

$$\frac{\partial \ln J_1}{\partial R_{UO}} = -\frac{K}{2} \frac{1}{R_{UO}} + \frac{1}{2 R_{UO}^2} \sum_{j=2}^K \varepsilon^2(j) b^{-j}$$

$$\frac{\partial \ln J_1}{\partial T_U} = \frac{\partial \ln J_1}{\partial b} \cdot \frac{\partial b}{\partial T_U}$$

$$= \left[-\frac{1}{2} \left(\sum_{j=1}^k j \right) \frac{1}{b} + \frac{1}{2 R_{UO}} \sum_{j=1}^k \varepsilon_j^2 b^{-(j+1)} \cdot j \right] \cdot b \frac{\Delta T}{T_U^2}$$

$$\text{令: } \frac{\partial \ln J_1}{\partial R_{UO}} = 0, \quad \frac{\partial \ln J_1}{\partial T_U} = 0$$

则优化估计 \hat{R}_{UO} 、 \hat{T}_U 满足方程组:

$$\hat{R}_{UO} = \frac{1}{k} \sum_{j=1}^k \varepsilon^2(j) \hat{b}^{-j} \quad (5)$$

$$\sum_{j=1}^k [j \cdot \varepsilon^2(j) \cdot \hat{b}^{-j}] = \hat{R}_{UO} \sum_{j=1}^k j \quad (6)$$

式中: $\hat{b} = \exp(-\Delta T / \hat{T}_U)$

$$\sum_{j=1}^k j = \frac{1}{2} k(k+1)$$

∴ 式(6)为:

$$\begin{aligned} \sum_{j=1}^k [j \varepsilon^2(j) \hat{b}^{-j}] &= \frac{k}{2} (k+1) \cdot \hat{R}_{UO} \\ &= \frac{1}{2} (k+1) \cdot \sum_{k=1}^k \varepsilon^2(j) \hat{b}^{-j} \end{aligned}$$

$$\text{即: } \sum_{j=1}^k [\varepsilon^2(j) \hat{b}^{-j} (j - \frac{k+1}{2})] = 0 \quad (7)$$

显然, 方程(7)的解为:

$$\varepsilon^2(j) \hat{b}^{-j} = \text{constant}$$

令: $\varepsilon^2(j) \hat{b}^{-j} = A$

则由方程(5)有:

$$\hat{R}_{UO} = A = \varepsilon^2(j) \hat{b}^{-j}$$

从而: $\hat{b}^j = \varepsilon^2(j) / A = \varepsilon^2(j) / \hat{R}_{UO}$

所以, 电压噪声方差的极大似然估计为:

$$\hat{R}_v(j) = \hat{R}_{v0} \cdot \hat{b}' = \varepsilon^2(j) = [Z_v(j) - H_v(j)V(j)]^2 \quad (8)$$

由于状态向量 $V(j)$ 为未知的待估计量, 在实际计算中可近似用其预测估计 $\hat{U}(j/j-1)$ 代之, 从而得到电压噪声方差的次优估计为:

$$\hat{R}_v(j) = [Z_v(j) - H_v(j)\hat{U}(j/j-1)]^2$$

估计的无偏性检验:

$$\begin{aligned} E(\hat{R}_v(j)) &= E\{[Z_v(j) - H_v(j)\hat{U}(j/j-1)]^2\} \\ &= E\{[Z_v(j) - H_v(j)\hat{U}(j/j-1)][Z_v(j) \\ &\quad - H_v(j)\hat{U}(j/j-1)]^T\} \\ &= H_v(j)P_v(j/j-1)H_v^T(j) + R_v(j) \quad (9) \end{aligned}$$

其中, $R_v(j)$ 为噪声方差的实际值。

由式(9)可知, 上述噪声方差估计为有偏估计, 因此, 实际采用的次优无偏估计为:

$$\begin{aligned} \hat{R}_v(j) &= \varepsilon^2(j) - H_v(j) \cdot P_v(j/j-1) \cdot H_v^T(j) \\ \hat{R}_v(j) &= [Z_v^*(j) - H_v(j)\hat{V}(j/j-1)]^2 \\ &\quad - H_v(j)P_v(j/j-1)H_v^T(j) \quad (10) \end{aligned}$$

2. 电流噪声参数的估计

电流噪声参数除噪声方差 $R_I(k)$ 外, 还包括非周期分量衰减时间常数 T_d , 由于状态转移矩阵 $\Phi_I(k+1/k)$ 中含有未知参数 T_d , 因此, 要同时实现噪声参数 $R_I(k)$ T_d 的在线优化估计一般较为困难。一种简便的处理方法是首先依据经验给出 T_d 的估值, 如近似将 T_d 取为线路的时间常数等。显然, T_d 的人为给定将带来一定误差, 这可通过, 虚拟噪声补偿技术加以解决。

首先, 对电流状态方程(3)及测量方程(4)作一变换。

已知电流噪声中的非周期分量满足:

$$I_d(k+1) = I_d(k) \cdot \exp(-\Delta T/T_d)$$

即: $I_d(k+1) = I_{d0}(k) \exp[-(k+1)\Delta T/T_d]$

式中: I_{d0} 为非周期分量的起始值。

选择 I_{d0} 作为新的状态变量, 则电流状态方程可修改为:

$$\begin{pmatrix} I_r(k+1) \\ I_I(k+1) \\ I_{d0}(k+1) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} I_r(k) \\ I_I(k) \\ I_{d0}(k) \end{pmatrix} \quad (11)$$

或: $\tilde{I}(k+1) = \tilde{\Phi}_I(k+1/k)\tilde{I}(k)$

电流测量方程可修改为:

$$Z_I(k) = [\cos\omega k\Delta T, -\sin\omega k\Delta T, \exp(-k\Delta T/T_d)] \begin{pmatrix} I_1(k) \\ I_2(k) \\ I_{d0}(k) \end{pmatrix} + N_I(k) \quad (12)$$

或： $Z_I(k) = \tilde{H}_I(k) \tilde{I}(k) + N_I(k)$

设：给定的非周期分量的衰减时间常数为 T_{dz} 。

则电流测量方程(12)为：

$$\begin{aligned} Z_I(k) &= [\cos\omega k\Delta T, -\sin\omega k\Delta T, \exp(-k\Delta T/T_{dz})] \tilde{I}(k) \\ &\quad + [0, 0, (\exp(-k\Delta T/T_d) - \exp(-k\Delta T/T_{dz}))] \bar{I}(k) \\ &\quad + N_I(k) \\ &= \bar{H}_I(k) \bar{I}(k) + \Delta\tilde{H}_I(k) \tilde{I}(k) + N_I(k) \end{aligned}$$

即： $Z_I(k) = \bar{H}_I(k) \bar{I}(k) + \tilde{N}_I(k)$

其中：

$\bar{H}_I(k)$ 为给定参数 T_{dz} 所对应的观测矩阵；

$\Delta\tilde{H}_I(k)$ 为采用给定参数 T_{dz} 后所引起的观测矩阵偏差；

$\tilde{N}_I(k)$ 为虚拟噪声，它包括实际暂态电流信号中的白噪声分量和由于模型参数偏差所带来的误差。

$$\begin{aligned} \tilde{N}_I(k) &= \Delta\tilde{H}_I(k) \tilde{I}(k) + N_I(k) \\ &= [\exp(-k\Delta T/T_d) - \exp(-k\Delta T/T_{dz})] \cdot I_{d0}(k) \\ &\quad + N_I(k) \end{aligned}$$

采用虚拟噪声技术后，稳态电流向量的自适应滤波将转化为下述带虚拟噪声的线性系统的自适应滤波。

系统状态方程：

$$\tilde{I}(k+1) = \tilde{\Phi}_I(k+1/k) \tilde{I}(k)$$

测量方程：

$$Z_I(k) = \bar{H}_I(k) \tilde{I}(k) = \tilde{N}_I(k)$$

严格地说，虚拟噪声本身不是白噪声，但若模型参数偏差所引起的“噪声”分量 $\Delta\tilde{H}_I(k)$

$\tilde{I}(k)$ 相对较小时，可近似作为白噪声处理，此时，基频电流向量的自适应滤波与前述电压向量的自适应滤波相同，只需对虚拟噪声方差进行在线估计，根据电压噪声方差的估计方程(10)，不难得到电流虚拟噪声方差的无偏次优估计为：

$$\hat{R}_I(j) = (Z_I(j) - \bar{H}_I(j) \hat{I}(j/j-1))^2$$

$$-\bar{H}_i(j) \tilde{P}_i^-(j/j-1) \bar{H}_i^T(j)$$

其中:

$\hat{\tilde{I}}(j/j-1)$ 为电流向量 $\tilde{I}(j)$ 的预测估计;

$\tilde{P}^-(j/j-1)$ 为预测估计误差协方差矩阵。

四 自适应卡尔曼滤波算法的检测

上述自适应滤波算法利用动模实验数据进行了检验, 模拟系统接线图如图 1 所示。借助单板机进行数据采样, 采样率为 24 点/周, 检测计算包括对不同故障点的单相接地短路, 两相短路, 两相接地短路, 三相短路等进行滤波计算。图 2~图 7 给出了有关的滤波结果, 检测计算结果表明, 该算法具有较快的收敛速度, 对于主要故障类型单相接地短路, 算法一般可在半周波内收敛至真值。然而也必须指出, 算法的收敛速度受故障类型的影响较大, 对于三相短路, 收敛时间可能达到一周波左右。此外, 初始条件的选取对滤波算法的收敛速度也将产生影响, 如何合理地选择初始估计条件需做进一步研究。

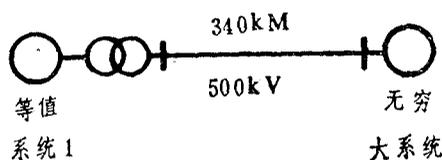


图 1 系统接线图

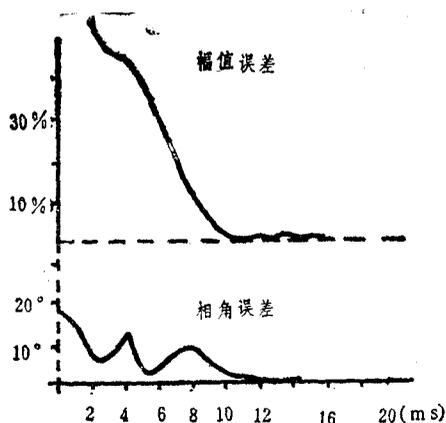


图 2 160kM处A—G故障
电流滤波结果

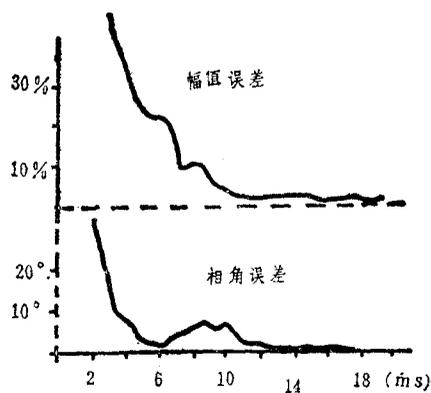


图 3 160kM处A—G故障
电压滤波结果

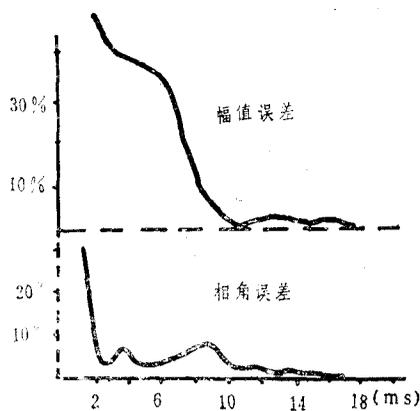


图4 290kM处A—G
故障电流滤波结果

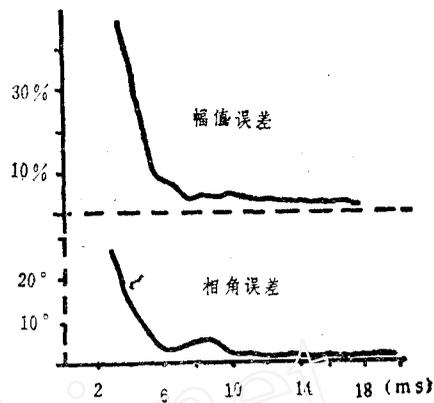


图5 290kM处A—G
故障电压滤波结果

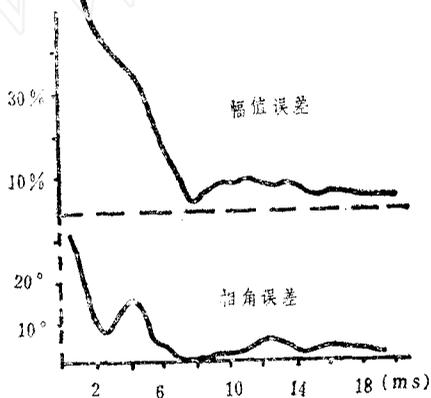


图6 160kM处三相故障
电流滤波结果

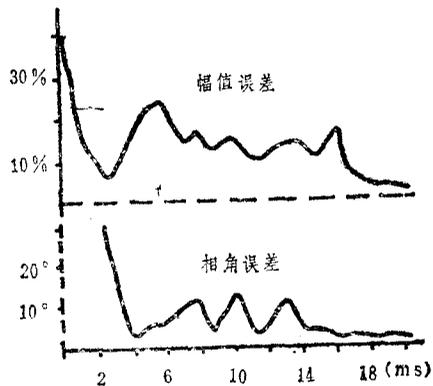


图7 160kM处三相故障
电压滤波结果

参考文献

- [1] Girgis, Adly A. Application of Kalman Filtering in Computer Relaying. IEEE Transactions on Power Apparatus and System, PAS-100, No. 7, 3387-3395
- [2] Girgis, Adly A. A New Kalman Filtering Based Digital Distance Relay. IEEE Transactions on Power Apparatus and System. PAS-101, No. 9, 3471-3480.
- [3] Girgis, Adly A., and R. Grover Brown, Modelling of fault-induced Noise Signal for Computer Relaying Applic-

