小波变换和傅里叶变换的结合

- ❖ 傅里叶变换的局限性：会丢失掉频率的时间信息
- ❖ 解决方法：STFT 的思想--分段进行 FT
  - ■ 先进行 WT 再 FT 或者先 FT 再 WT 都可以，关键是对非平稳信号的时频分布进行判断
  - ■ FT 的去噪：粗略地使用一个阈值进行滤波，这个方法在信噪比偏小的时候会有随机性地产生误判现象
- ❖ 平稳信号的去噪结果
  - ■ 信号：

```
t = np.arange(0,T0,1/fs)
data=np.sin(2*800*np.pi*t)+awgn(np.sin(2*800*np.pi*t),10)
```

图 1 原信号

- ■ 先进行小波变换进行初步滤波，由于带通滤波器并不是十分惊细，选择保留含有 800 赫兹频率的第 3 层以及相邻层第 4 层（这里该如何处理更加合理？第 2 层为什么有的时候保留会让谐波减少，为什么有的时候保留会让谐波增多呢）：

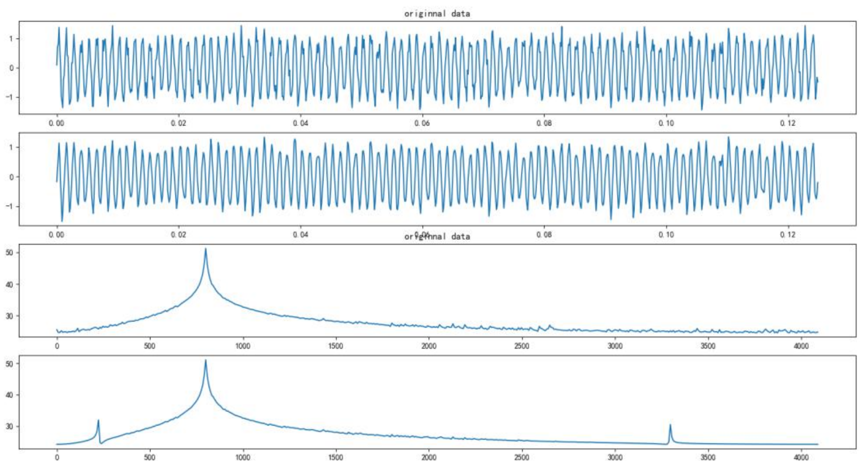


图 2 小波分解与重构后信号

- 可以看到此时有谐波的产生
- 接着使用 FT 来去掉这些谐波，设置阈值为小波重构的数据的傅里叶变换的绝对值最大值的 1/10，即去掉能量差为 100 倍的谐波，这是合理的，将它们置为 10^(-40)~~(这一步是否合理呢？是不是把谐波置为相邻两个正常//。数据的平均值更好）：（不需要，因为其实这些都足够小）~~

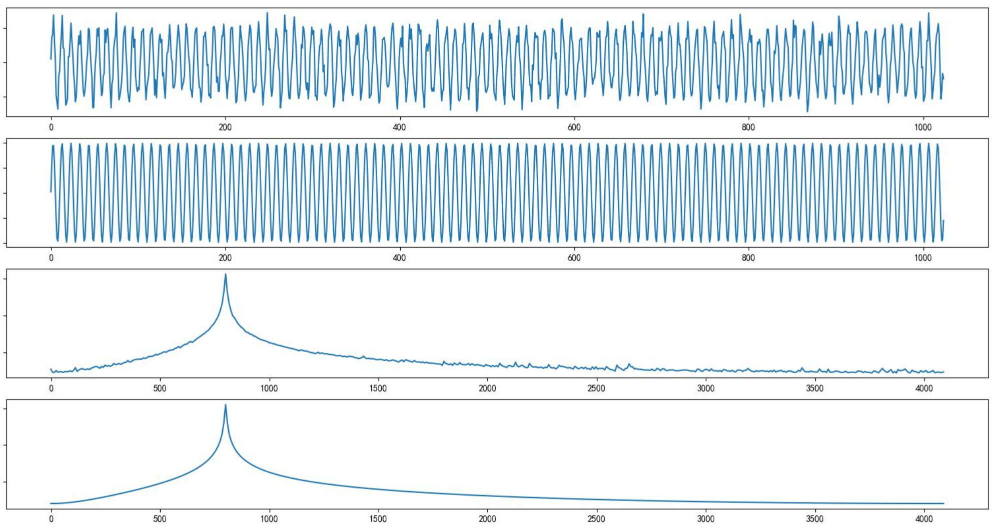


图 3 简单的信号十分好处理

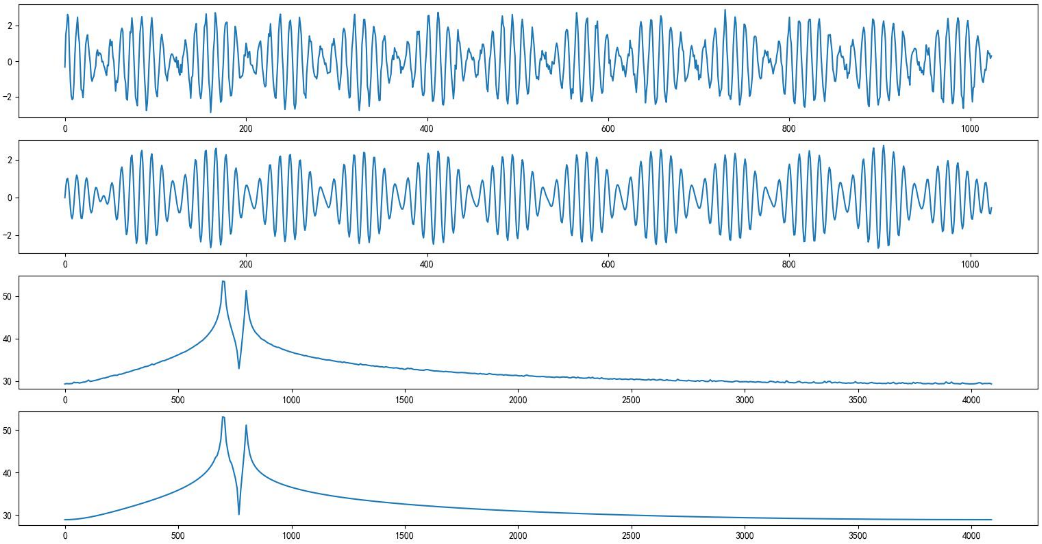


图 4 简单信号

❖ **非平稳信号的去噪结果：**

■ **原信号：**

```
= np.piecewise(t,[t<1,t<0.8,t<0.3],
                [lambda t : np.sin(200*np.pi*t)+awgn(np.sin(200*np.pi*t),10
                 lambda t : np.sin(500*np.pi*t),
                 lambda t :np.sin(200*np.pi*t)+2*np.cos(130*np.pi*t)])
```

■ **思路：在 CWT 中得到的小波系数每一维画图是这样的：**

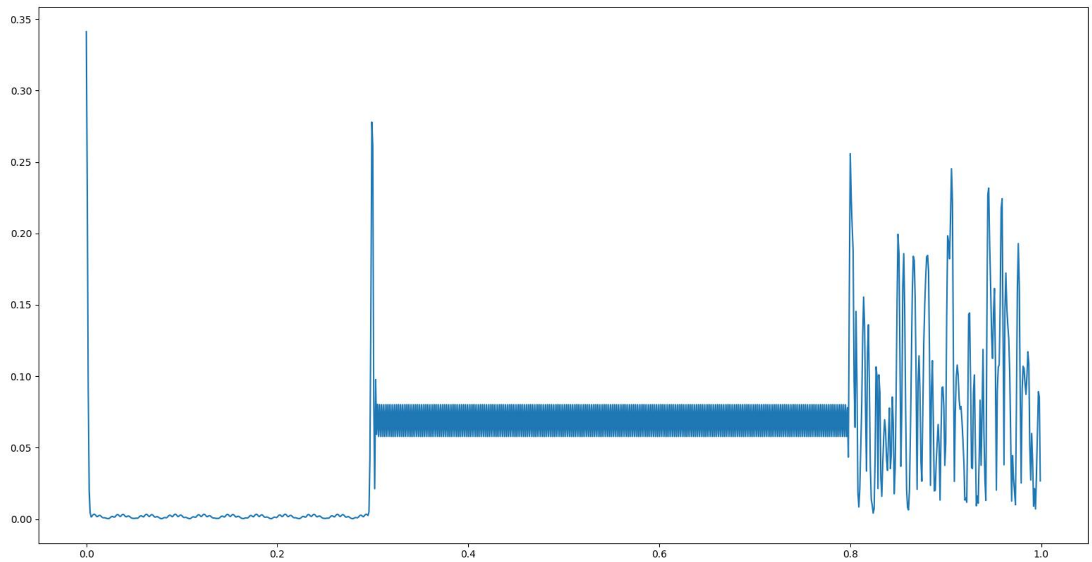
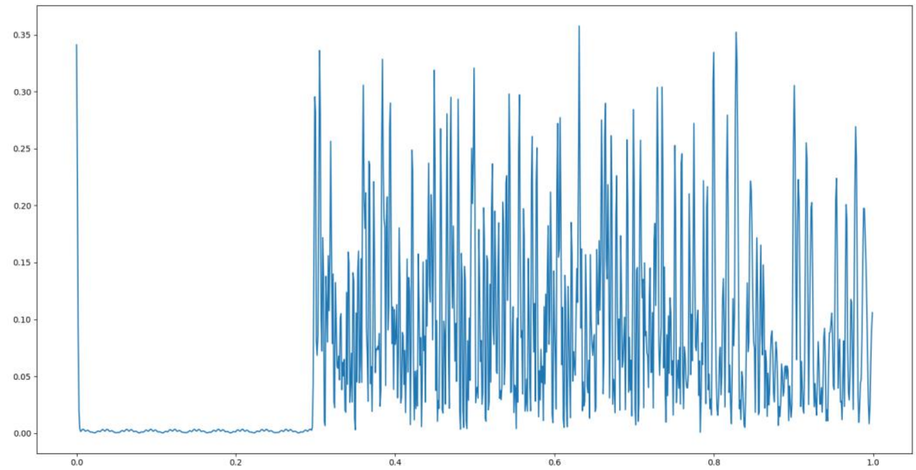


图 5 小波系数

■ 可以看到这个曲线反映了函数的频率特征；但是如果分段函数每一段都有噪声时仍旧不能体现出频率在时间上的分段：



并且如果两段之间没有跳变，暂时想不到可靠的方法辨别出分界点，除非人眼或者结合 gui 的鼠标点击？--经过多次尝试（对不起我对 matplotlib 的一些 gui 应用实在是不熟悉），这个方法并不一定可行：如果整段信号充满噪声，几乎无法分辨出信号的突变点，而且如果信号没有突变，就不能解决任何问题……

■ 补充：上面那个问题，突然想到可以利用模极大值寻找突变点……于是想起了之前没有看懂的模极大值去噪，后面开始尝试--发现并不是按照理论所说 scale 增大噪声对应的李氏指数会减小，因此这个方法似乎不可行

■ 补充 plus:看到有说比较 不同 scale 的相关性，可以找到突变点？后面有时间试一试

■ 结果：如果解决了分段的问题，整个方法应该是效果还行的：

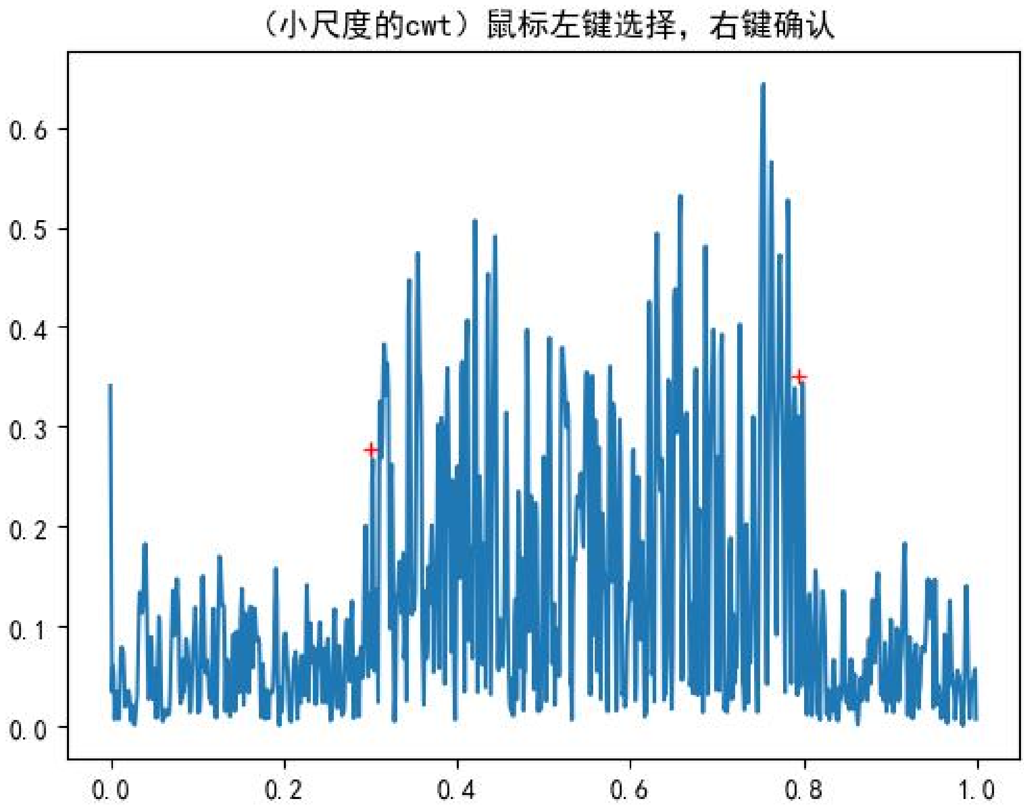


图 6 这里是写错标题了，中键确认

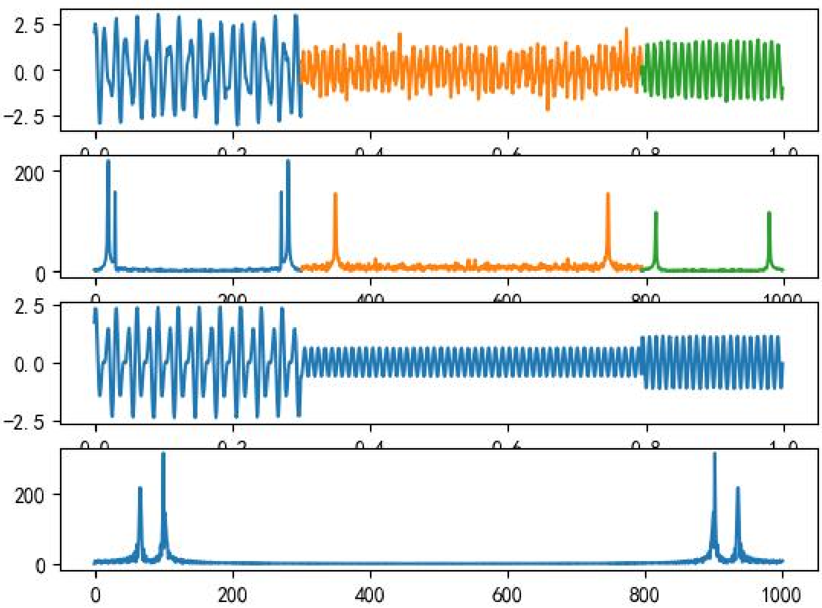


图 7 去噪效果

EMD 模态分解

❖ 特点：具有自适应滤波特性；近似正交性；完备性--由 IMF 和残余量可在较小误差范围内恢复原信号；

❖ **本征模态函数**

❖ 复信号、解析信号的瞬时频率：时间的函数，与希尔伯特-黄变换有关联，是信号在时间上的局部体现

❖ 算法原理：

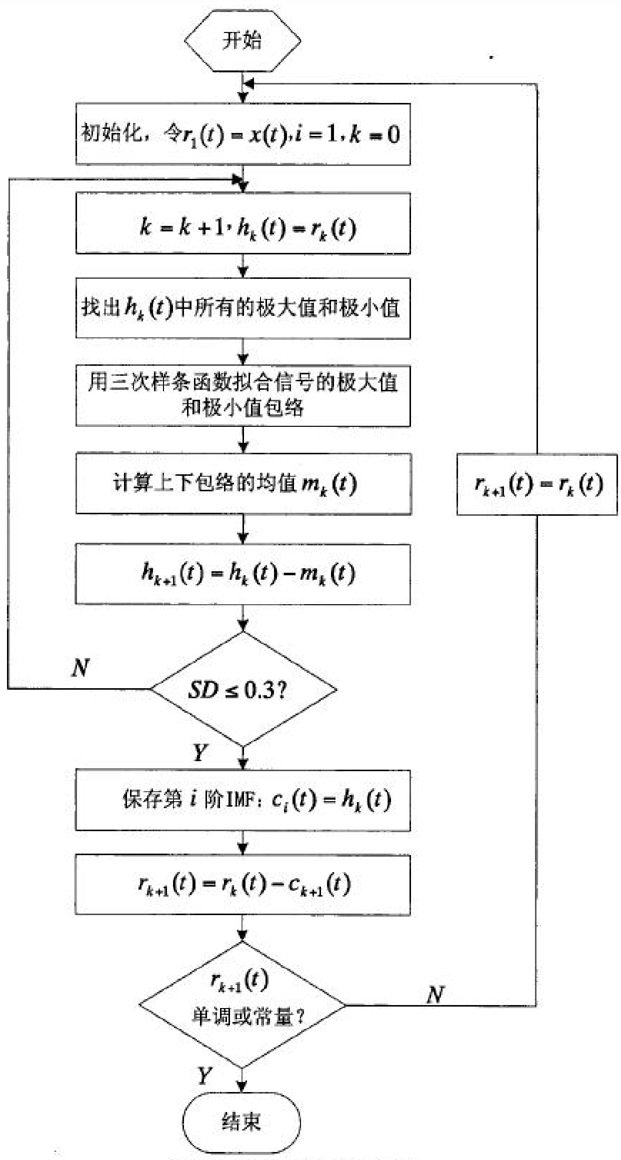


图 2.1 EMD 算法流程图

图 8 EMD 算法流程图

❖ 优点：可以分析非平稳信号，不受到海森堡不确定性原理的约束，可以同时在时域和频域达到较好的分辨率

❖ 缺点：阈值处理比较麻烦-->> EEMD

❖ 效果：

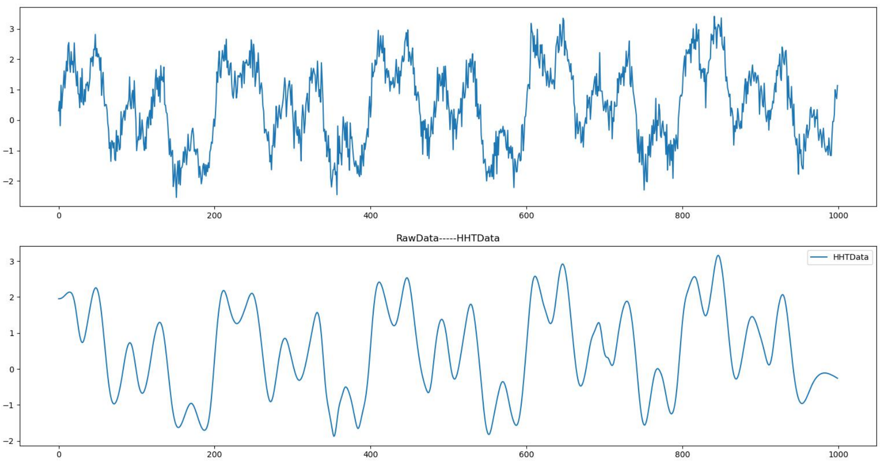


图 9 HHT-平稳信号

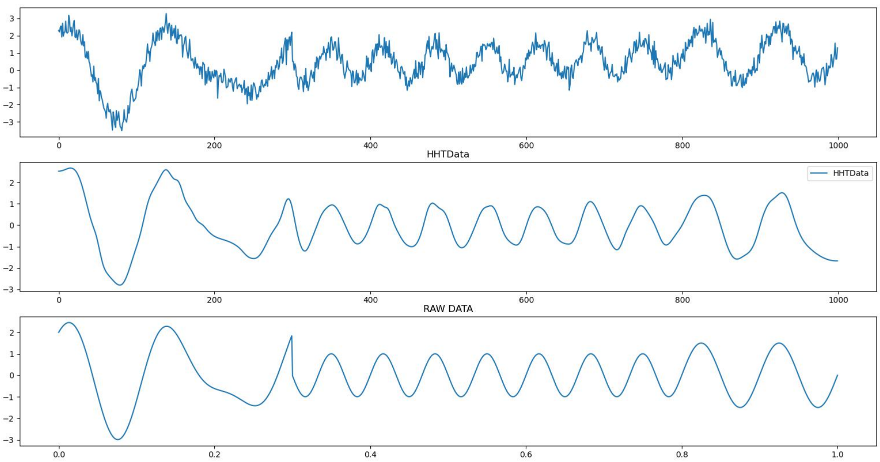


图 10 HHT-非平稳信号

自适应滤波器

❖ 原理：使用误差信号控制系统参数，从而使输出信号更加接近期望信号

❖ 算法：

- 基于维纳滤波的最小均方（LMS）算法

  ◆ 原理：对于一个联合平稳信号 $x(t)=s(t)+v(t)$，通过设置信道参数 h 来达到误差的均方根最小，即 $E(e^2(t))=E\{[s(t+\tau)-x(t)*h(t)]^2\}$。若用 h_{opt} 来表示最优化的 $h(t)$，则有维纳-霍夫方程：$-R_{xs}(t+\tau)+\int_{-\infty}^{\infty}h_{opt}(u)R_{xx}(t-u)du=0(\text{若}h_{opt}\text{为非因果的})$；

  $$-R_{xs}(t+\tau)+\int_{-\infty}^{\infty}h_{opt}(u)R_{xx}(t-u)du=\begin{cases}0,\mathrm{t}\ge 0\\ \text{未知函数，}\ t<0\end{cases}((\text{若}h_{opt}\text{为因果的})$$

- 基于最小二乘法的地推最小二乘 RLS 算法

- 基于卡尔曼滤波的卡尔曼算法

  - 原理：以最小均方差为最佳估计准则，解决一个动态变化的系统的状态跟踪问题

  - 前提：系统的状态方程是线性的；观测方程是线性的；过程噪声符合零均值高斯分布；观测噪声符合零均值高斯分布

❖ 实现结果：

- 维纳滤波器：根据

$$\vec{h}_{opt}=\vec{R}_{xx}^{-1}\vec{R}_{xs},\vec{R}_{xs}=[R_{xs}(0),R_{xs}(1)\ldots R_{xs}(M)]^{T},$$

$$\vec{h}_{opt}=[h_{opt}(0),h_{opt}(1)\ldots h_{opt}(M)]^{T},$$

$$\vec{R}_{xx}=\begin{matrix} R_{xx}(0) & R_{xx}(1) & R_{xx}(2) & \ldots & R_{xx}(M) \\ R_{xx}(1) & R_{xx}(0) & R_{xx}(1) & \ldots & R_{xx}(M-1) \\ \ldots & \ldots & \ldots & .. & \ldots \\ R_{xx}(M) & R_{xx}(M-1) & R_{xx}(M-2) & \ldots & R_{xx}(0) \end{matrix}$$

可以求出最佳的 $h(t)$，含噪声信号和它的卷积就是期望信号。

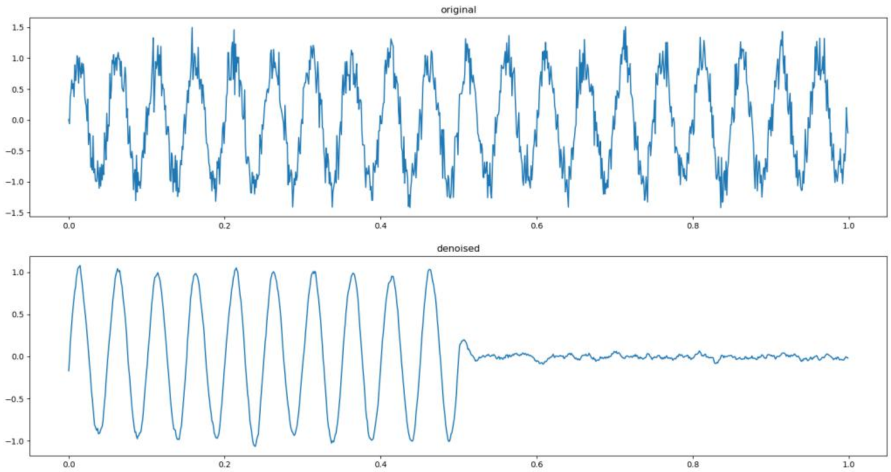


图 11 维纳滤波器·平稳信号

但是维纳滤波只适用于平稳信号

图 12 维纳滤波的效果

- LMS 算法：根据 $$e(n)=d(n)-y(n);$$ $$y(n)=x^{T}(\vec{n})\,w(\vec{n});$$ $$w(\vec{n+1})=w(\vec{n})+2*mu*e(n)*x(\vec{n})$$ 进行迭代，最终会收敛为维纳滤波器：

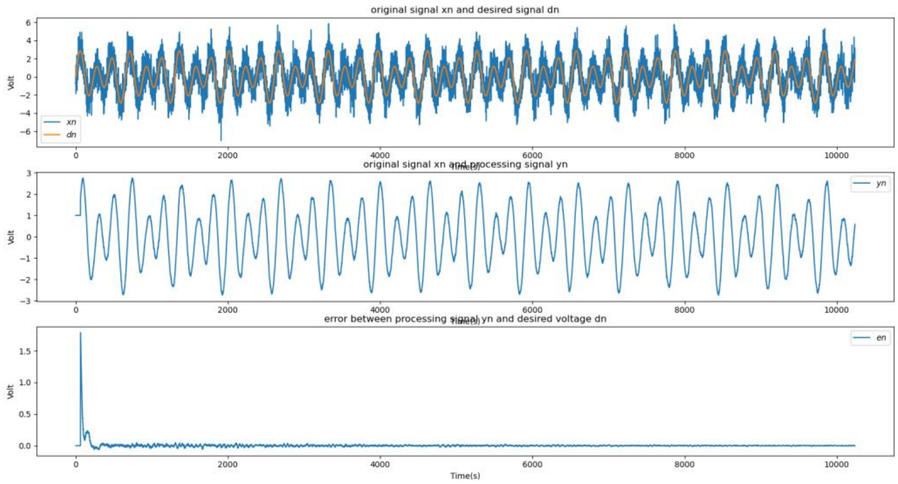


图 13 LMS 算法・平稳信号

■ 卡尔曼滤波器：根据迭代方程：

$$
\begin{aligned}
&\mathrm{Pr}\,ediction:\\
&\overline{x_t} = x_{t-1}\\
&\overline{P_t} = P_{t-1} + R\\
&correction:\\
&K_t = \overline{P_t}(\overline{P_t} + Q)^{-1}\\
&x_t = \overline{x_t} + K_t(Z_t - H_t\overline{x_t})\\
&P_t = (1 - K_t H_t)\overline{P_t}
\end{aligned}
$$

进行迭代，其中 Q 和 R 属于超参数，需要自己调节到合适的值；最终结果：

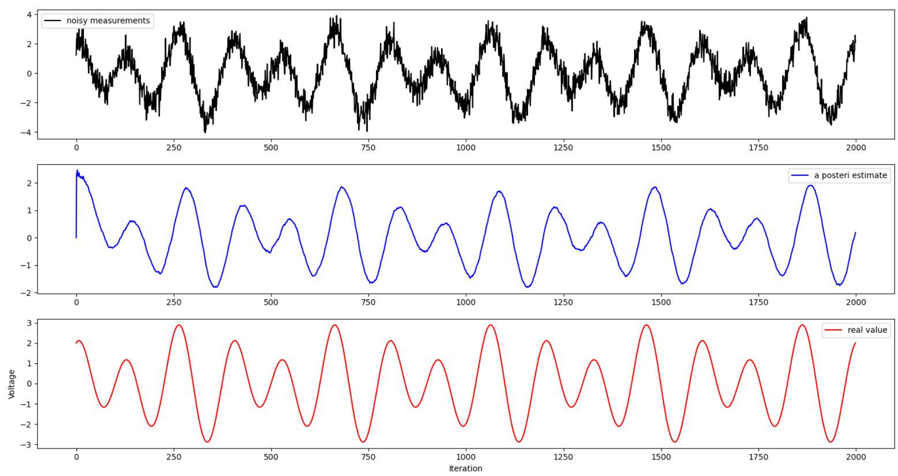


图 14 卡尔曼滤波器・平稳信号

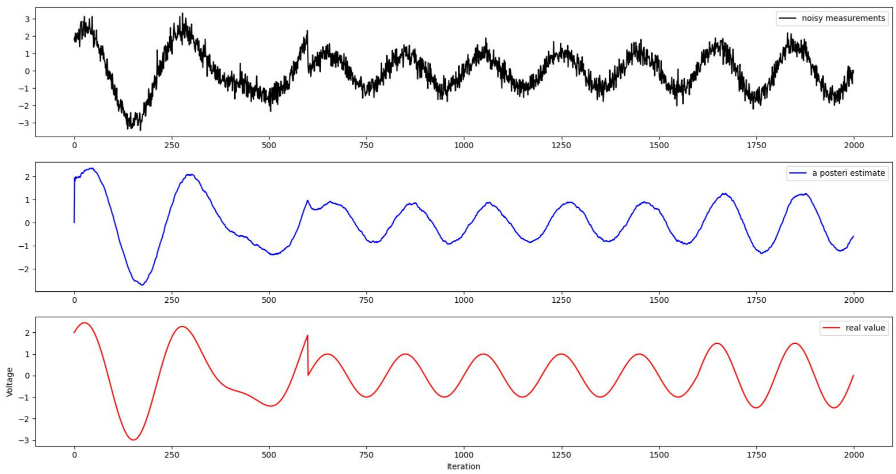


图 15 卡尔曼滤波器·非平稳信号

如图可见，在多次调解 Q 和 R 后，不论是平稳信号还是非平稳信号，都可以有一个较好的滤波效果