---

***Example***

We wish to search for the Wiener filter **w** (FIR, length 2) to predict samples.

**Figures 7 and 8.7**: Prediction configuration

We assume samples $u(n)$ are from a 2nd order AR process

$$u(n)+a_1 u(n-1)+a_2 u(n-2)=v(n)$$

with $v(n)$ a white-noise process, zero mean, and variance $\sigma_v^2$ (chosen so that $\sigma_u^2=1$) For experimentation purposes we'll leave $a_1$ and $a_2$ unknown for now.

**Figure** AR process synthesizer

1) Calculate **R** for the 2nd order AR process
2) Calculate **p**
3) Select μ
4) Initialize **w**(0) = **0**
5) Iterate update equation

$$\mathbf{w}(n+1)=\mathbf{w}(n)+\mu[\mathbf{p}-\mathbf{R}\mathbf{w}(n)]$$

6) Examine results

---

1) Calculate **R** for the 2nd order AR process. For the length 2 filter, **w** and 2nd order AR process we have

$$\mathbf{R}=\begin{bmatrix} r(0) & r(1) \\ r^*(1) & r(0) \end{bmatrix}$$

where [see (2.77), (2.88)]

$$r(0)=\sigma_u^2$$

$$r(1)=-\frac{a_1}{1+a_2}\sigma_u^2$$

$$\sigma_u^2=\left(\frac{1+a_2}{1-a_2}\right)\frac{\sigma_v^2}{(1+a_2)^2-a_1^2}$$

As a side note we also have for the eigenvalues of **R**

$$\lambda_1 = \left(1 - \frac{a_1}{1+a_2}\right)\sigma_u^2$$

$$\lambda_1 = \left(1 + \frac{a_1}{1+a_2}\right)\sigma_u^2$$

$$\chi = \frac{1 - a_1 + a_2}{1 + a_1 + a_2}$$

and the eigenvector matrix

$$\mathbf{Q} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

---

2) Calculate **p**

$$\mathbf{p} = \begin{bmatrix} p(0) \\ p(-1) \end{bmatrix}$$

where

$$\begin{aligned} p(-k) &= E[u(n-k)d(n)] \\ &= E[u(n-1-k)u(n)] \\ &= E[u(n)u(n-1-k)] \quad ;u \in \Re \\ &= r(k+1) \end{aligned}.$$

Thus

$$\mathbf{p} = \begin{bmatrix} \rho(1) \\ \rho(2) \end{bmatrix}$$

where (see 2.84)

$$r(2) = \left(-\alpha_2 + \frac{\alpha_1^2}{1+\alpha_2}\right)\sigma_v^2 .$$

---

**Experiment 1**

Choose μ = 0.3. Vary $a_1$ and $a_2$ for different eigenvalue spreads

**Table 8.1**

**Figures 8.9 and 8.10**

**Observations**

1) Typically the trajectory follows a curved path.
2) When eigenvalues are equal, trajectory is a straight line (Figure 8.9a).
3) When eigenvalue spread is high, error surface assumes the shape of a deep valley and trajectory takes on a distinct bend (Figure 8.9d).
4) SD algorithm converges fastest when eigenvalues are equal or **w**(0) is chosen correctly.

---

**Experiment 2**

Fix eigenvalue spread, χ = 10 and choose μ = 0.3 and 1.0.

**Figure 8.12**

**Observations**

When $\mu$ is small, the transient behavior of the SD algorithm is overdamped in that the trajectory follows a continuous path (Figure 8.12a).

When $\mu$ is large, the transient behavior of the SD algorithm is underdamped in that the trajectory exhibits oscillations (Figure 8.12b).

---

**Conclusion**

We conclude that the transient behavior of SD algorithm is highly sensitive to $\mu$ and $\chi$.