6.5 Beating the Eigenvalue Spread – Decorrelation

- We have seen that the eigenvalue spread $\lambda_{max}/\lambda_{min}$ (r) limits the performance. The slow mode (λ_{min}) is r times slower than the fast mode (λ_{max}), but if we try to speed it up by increasing μ, then the fast mode increases J_{ex}.

- Suggestion — if we could work individually with the modes, we could give them individual speed controls μ_i (step size parameter).

$$v_i(n+1) = (1 - \mu_i \lambda_i)\, v_i(n) + \mu_i z_i(n)$$

Specifically $\mu_i = \alpha/\lambda_i$

Then

$$v_i(n+1) = (1-\alpha)\, v_i(n) + \alpha \frac{z_i(n)}{\lambda_i}$$

– Now modes are altered so all have same decay constant

$$n_c = \frac{-1}{\ln(1-\alpha)} \approx \frac{1}{\alpha}$$

just as we did in steepest descent

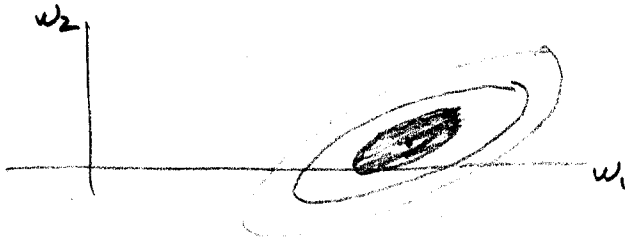
– Also, the perturbing noise variances become

$$E\left[\left|\frac{z_i(n)}{\lambda_i}\right|^2\right] = \frac{J_{min}\lambda_i}{\lambda_i^2} = \frac{J_{min}}{\lambda_i} \qquad \text{(p. 6.4.4)}$$

and the asymptotic jitter variance on mode i:

$$\sigma_{v_i}^2 = \frac{\alpha^2 J_{min}/\lambda_i}{1-(1-\alpha)^2} \approx \frac{\alpha}{2}\frac{J_{min}}{\lambda_i} \qquad \text{(p. 6.4.5)}$$

This means that the blur has same shape as contours



- So the effect on excess mse is

$$J_{ex} = \sum_{i=1}^{M} \lambda_i \sigma_{v_i}^2 = \frac{\alpha}{2} M J_{min}$$

- We have gained independence from eigenvalspread!

$$J_{ex} = \frac{\alpha}{2} M J_{min} \approx \frac{M J_{min}}{2 n_c}$$

For example, if we design for $J_{ex} = 0.1 J_{min}$

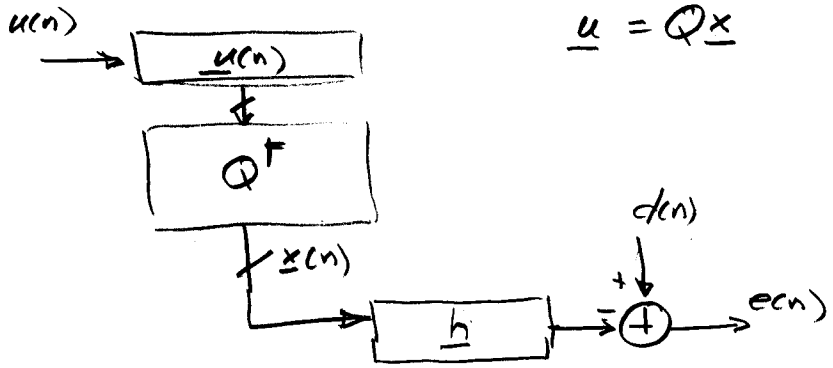
$$n_c = \frac{M}{0.2} = 5M$$

Compare this with original algorithm (p. 6.4.7)

$$5r \leq n_c \leq 5rM \quad \text{where} \quad r = \lambda_{max}/\lambda_{min}$$

- Where's the catch?
  - We don't know R, so we don't know Q, so we can't decorrelate.
  - But there are possibilities for adaptive decorrelation so we'll contrive with decorrelated adaptation, assuming we know R

- Adaptation with explicit decorrelation in eigenspace:



$$\underline{u} = Q\underline{x} \qquad \underline{x} = Q^+\underline{u}$$

Here we have performed transformation first, shift next, unlike previous derivation.

Correlation arrays: $R_x = E[\underline{x}\underline{x}^+] = Q^+ R Q = \Lambda$

$$\underline{p}_x = E[e^*\underline{x}] = Q^+\underline{p}$$

- From p 6.4.2, update is

$$\underline{h}(n+1) = \underline{h}(n) + \mu\, e^*(n)\underline{x}(n) \rightarrow \underline{p}_x - R_x\underline{h}(n) + \underline{\gamma}_x(n)$$

$$= (I - \mu R_x)\underline{h}(n) + \mu\underline{p}_x + \mu\underline{\gamma}_x(n)$$

where $\underline{\gamma}_x(n) = e^*(n)\underline{x}(n) = Q^+ e^*(n)\underline{u}(n) = \underline{v}(n)$ from earlier analysis p 6.4.4

check: $E[\underline{\gamma}_x(n)\underline{\gamma}_x^+(n)] = E[|e(n)|^2 \underline{x}(n)\underline{x}^+(n)] \approx J_{min} R_x$

$$= J_{min}\Lambda = E[\underline{v}\underline{v}^+] \text{ as expected}$$

- Now do the shift. From the iteration above

$$\underline{h}_o = R_x^{-1}\underline{p}_x = Q^+ R^{-1} Q\, Q^+ p = Q^+ R^{-1}\underline{p} = Q^+\underline{w}_o$$

Then $\underline{h}(n) - \underline{h}_o = \underline{v}(n)$ defined as in previous section

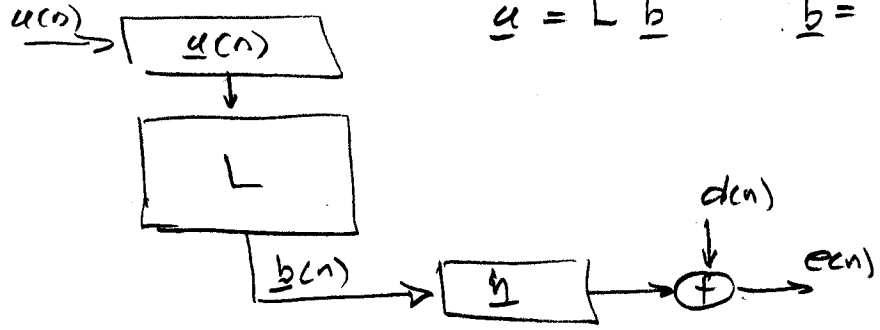
and $\underline{v}(n+1) = (I - \mu\Lambda)\underline{v}(n) + \mu\underline{v}(n)$

- The matrix transformation gives wanted decoupling

$$v_i(n+1) = (1 - \mu\lambda_i)v_i(n) + \mu\, v_i(n)$$

- We can also decorrelate by Gram Schmidt, using prediction error filters

$$\underline{a} = L^+ \underline{b} \qquad \underline{b} = L\,\underline{u}$$



- Correlation arrays

$$R_b = E[\underline{b}\,\underline{b}^+] = L R L^+ = D = \mathrm{diag}[P_0, P_1 \cdots, P_{M-1}]$$

$$\underline{p}_b = E[e^* \underline{b}] = L\,\underline{p}$$

- Iteration:

$$\underline{h}(n+1) = (\underline{I} - \mu R_b)\underline{h}(n) + \mu \underline{p}_b + \mu\, \underline{\gamma}_b(n)$$

where $\underline{\gamma}_b(n) = e^*(n)\underline{b}(n) = L e^*(n)\underline{u}(n) = L\,\underline{\gamma}(n)$

and $E\left[\underline{\gamma}_b \underline{\gamma}_b^+\right] = L\,E[\underline{\gamma}\,\underline{\gamma}^+]\,L^+ = J_{min}\, L R L^+ = J_{min}\, D$

- Now shift:

$$\underline{h}_o = R_b^{-1} p_b = L R^{-1} L^+ L \underline{p} = L R^{-1} \underline{p} = L\,\underline{w}_o$$

Define $\underline{\tilde{v}}(n) = \underline{h}(n) - \underline{h}_o$

and the iteration is

$$\underline{\tilde{v}}(n+1) = (\underline{I} - \mu R_b)\,\underline{\tilde{v}}(n) + \mu\, \underline{\gamma}_b(n)$$

$$= (I - \mu D)\,\underline{\tilde{v}}(n) + \mu\, \underline{\gamma}_b(n)$$

- Decouples to $\tilde{v}_i(n+1) = (1 - \mu P_i)\,\tilde{v}_i(n) + \mu\, \underbrace{\gamma_{b_i}(n)}_{\text{variance } P_i}$

so $$\sigma^2_{\tilde{v}_i} = \frac{\mu^2 J_{min} P_i}{1 - (1 - \mu P_i)^2} \approx \frac{\mu J_{min}}{2}$$

$$J_{ex} = \sum_{i=0}^{M-1} P_i\, \sigma^2_{\tilde{v}_i} = \frac{\mu J_{min}}{2} \sum_{i=0}^{M-1} P_i$$

and since decoupled, we can choose

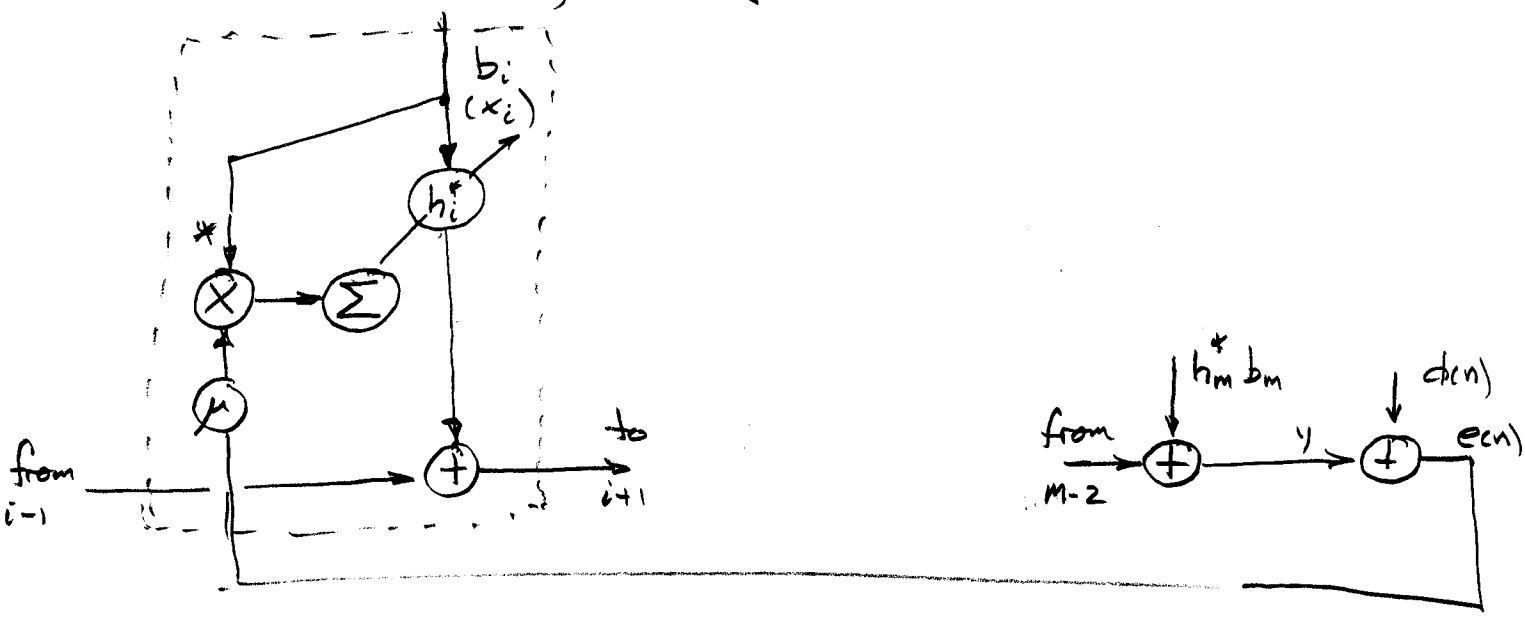
$$\mu_i = \frac{\alpha}{P_i}$$

$$\tilde{v}_i(n+1) = (1-\alpha)\tilde{v}_i(n) + \alpha \frac{\phi_{b_i}(n)}{P_i} \qquad n_c \approx \frac{1}{\alpha}$$

$$\sigma^2_{\tilde{v}_i} \approx \frac{\alpha J_{min}}{2P_i} \qquad J_{ex} = \frac{\alpha J_{min} M}{2}$$

- same advantages as in eigenspace decoupling:
  - independence from eigenvalue ratio
  - uniform convergence speed.

- In both cases, a stage looks like this



- Too bad we don't know R. But what if the lattice filter could be made adaptive? Then adaptive decorrelation.
  - Sets the stage for gradient adaptive lattice