Understanding the autoregressive filter used for second-stage control

Aditya Sengupta

In his 2023 SPIE paper, Ben uses this filter to separate out the signals going into the SHWFS and FAST:

$$c_{H\{n\}} = \alpha c_{H\{n-1\}} + \alpha(c_n - c_{n-1})$$
$$c_{L\{n\}} = \alpha c_{L\{n-1\}} + (1-\alpha)c_{n-1}$$

where $\alpha = e^{-f_{\text{cutoff}}/f_{\text{loop}}}$. I'd like to convince myself that these do actually define high- and low-pass filters respectively, so I'll calculate their frequency responses.

Since Ben ran these filters in separate terminals, the $c_{H\{n\}}$s and $c_{L\{n\}}$s can't technically see each other, but they do interact since both wavefront sensors read off the coefficients c_n, which get set by the commands from both wavefront sensors. For our purposes here I'll assume $c_n = c_{H\{n\}} + c_{L\{n\}}$; I think this is accurate up to the assumption that both wavefront sensors agree on the overall reconstruction.

Using this, we can rewrite the filter to eliminate c_n:

$$c_{H\{n\}} = \alpha c_{H\{n-1\}} + \alpha\left(c_{H\{n\}} + c_{L\{n\}} - c_{H\{n-1\}} - c_{L\{n-1\}}\right)$$
$$c_{L\{n\}} = \alpha c_{L\{n-1\}} + (1-\alpha)\left(c_{H\{n-1\}} + c_{L\{n-1\}}\right)$$

or

$$c_{H\{n\}} = \frac{\alpha}{1-\alpha}\left(c_{L\{n\}} - c_{L\{n-1\}}\right)$$
$$c_{L\{n\}} = c_{L\{n-1\}} + (1-\alpha)c_{H\{n-1\}}$$

which we can rewrite further so that it's just in terms of a state vector. Let $\vec{c}_n = \begin{bmatrix} c_{H\{n\}} & c_{L\{n\}} \end{bmatrix}^T$, then:

$$\vec{c}_n = \begin{bmatrix} 1 & 0 \\ 1-\alpha & 1 \end{bmatrix} \vec{c}_{n-1} + \vec{x}_n$$

where I've added in driving noise $\vec{x}_n$.

Let's compute a frequency response by allowing $\vec{x}_n = \begin{bmatrix} e^{i\omega_H n} & e^{i\omega_L n} \end{bmatrix}^T$ and setting $\vec{c}_n = F(\omega_H, \omega_L)\vec{x}_n$ where F is a 2x2 matrix (generally we'd use H for this but that's overloaded.)

$$F(\omega_H, \omega_L) \begin{bmatrix} e^{i\omega_H n} \\ e^{i\omega_L n} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1-\alpha & 1 \end{bmatrix} F(\omega_H, \omega_L) \begin{bmatrix} e^{i\omega_H (n-1)} \\ e^{i\omega_L (n-1)} \end{bmatrix} + \begin{bmatrix} e^{i\omega_H n} \\ e^{i\omega_L n} \end{bmatrix}$$

$$F(\omega_H, \omega_L) = \begin{bmatrix} 1 & 0 \\ 1-\alpha & 1 \end{bmatrix} F(\omega_H, \omega_L) \begin{bmatrix} e^{-i\omega_H} & 0 \\ 0 & e^{-i\omega_L} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$F(\omega_H, \omega_L) = \begin{bmatrix} e^{-i\omega_H} & 0 \\ (1-\alpha)e^{-i\omega_H} + e^{-i\omega_L} & e^{-i\omega_L} \end{bmatrix} F(\omega_H, \omega_L) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$F(\omega_H, \omega_L) = \text{inv}\left(\begin{bmatrix} 1 - e^{-i\omega_H} & 0 \\ (\alpha - 1)e^{-i\omega_H} - e^{-i\omega_L} & 1 - e^{-i\omega_L} \end{bmatrix} \right)$$

which gives us the final result,

$$F(\omega_H, \omega_L) = \frac{1}{e^{-i\omega_L} + e^{-i\omega_H} - e^{-i(\omega_L+\omega_H)}} \begin{bmatrix} 1 - e^{-i\omega_L} & 0 \\ (1-\alpha)e^{-i\omega_H} + e^{-i\omega_L} & 1 - e^{-i\omega_H} \end{bmatrix}$$

Let's try some substitutions!

- $F(0,0) = \left[\begin{smallmatrix} 0 & 0 \\ 2-\alpha & 0 \end{smallmatrix}\right]$; a DC offset gets sent from the high-frequency component to the low-frequency component, and then gets removed.
- $F(\omega,0) = \left[\begin{smallmatrix} 0 & 0 \\ (1-\alpha)e^{-i\omega}+1 & 1-e^{-i\omega} \end{smallmatrix}\right]$; a pure high frequency survives only in the low-frequency component? I don't think this is what I want the filter to do.
- $F(0,\omega) = \left[\begin{smallmatrix} 1-e^{-i\omega} & 0 \\ 1-\alpha+e^{-i\omega} & 0 \end{smallmatrix}\right]$; a pure low frequency gets completely ignored? I'm not sure I'm reading these right.

I think what I'd hope this shows me is the filter can take in a signal with a frequency above f_{cutoff} and keep it only in the first component, and one with a frequency below f_{cutoff} and keep it only in the second component. But I don't think that's what this is saying. I might not be interpreting it the right way.