

## 6.s02: EECS II - From A Medical Perspective

Ryan Lacey <rlacey@mit.edu>

Collaborator(s): Jorge Perez

March 3, 2014

1. (a) The fundamental period of this DT signal is  $N = 7$ . Let  $\Omega_0 = \frac{2\pi}{7}$

Evaluating the synthesis equation for the period of  $X[k]$  between  $k = -3$  and  $k = 3$

$$x[n] = \frac{1}{2} (e^{j2\Omega_0 n} + e^{-j2\Omega_0 n})$$

$$x[n] = \cos(2\Omega_0 n) = \cos\left(\frac{4\pi}{7}n\right)$$

- (b)  $x[n] = \cos(\Omega n)$

$$\omega = \frac{\Omega}{T_s} = \frac{\frac{4\pi}{7}}{0.1} = \frac{40\pi}{7}$$

$$x[t] = \cos(\omega t) = \cos\left(\frac{40\pi}{7}t\right)$$

- (c) The fundamental period of this DT signal is  $N = 7$ . Let  $\Omega_0 = \frac{2\pi}{7}$

Evaluating the synthesis equation for the period of  $X[k]$  between  $k = -3$  and  $k = 3$

$$x[n] = \frac{1}{2} (e^{j3\Omega_0 n} + e^{-j3\Omega_0 n})$$

$$x[n] = \cos(3\Omega_0 n) = \cos\left(\frac{6\pi}{7}n\right)$$

- (d)  $x[n] = \cos(\Omega n)$

$$\omega = \frac{\Omega}{T_s} = \frac{\frac{6\pi}{7}}{0.15} = \frac{40\pi}{7}$$

$$x[t] = \cos(\omega t) = \cos\left(\frac{40\pi}{7}t\right)$$

2. (a)  $H_d[k] = \frac{I[k]}{Q[k]}$

We want to solve for  $I[k]$

$$I[t] = Q[t] - Q[t-1] \implies I[k] = Q[k] (1 - e^{-j\Omega_0 k})$$

$$\therefore H_d[k] = 1 - e^{-j\Omega_0 k}$$

(b)  $H_a[k] = (H_d[k])^{-1}$

$$H_a[k] = \frac{1}{1 - e^{-j\Omega_0 k}}$$

(c)

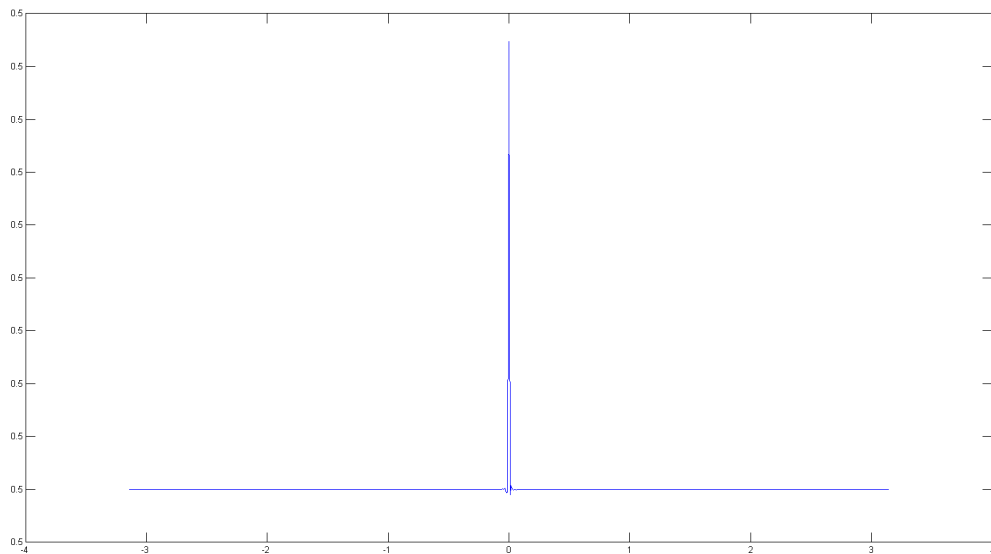


Figure 1:  $H_a$  low-pass filter.

- (d) The shape of the graph shows how we will tightly limit accepted inputs to those of low magnitude. The curve drops very steeply showing little tolerance for noise that could occur at larger values.

$$3. \quad (a) \quad y[n] = \left( \sum_{i=1}^3 \alpha^i \right)^{-1} (\alpha x[n] + \alpha^2 x[n-1] + \alpha^3 x[n-2])$$

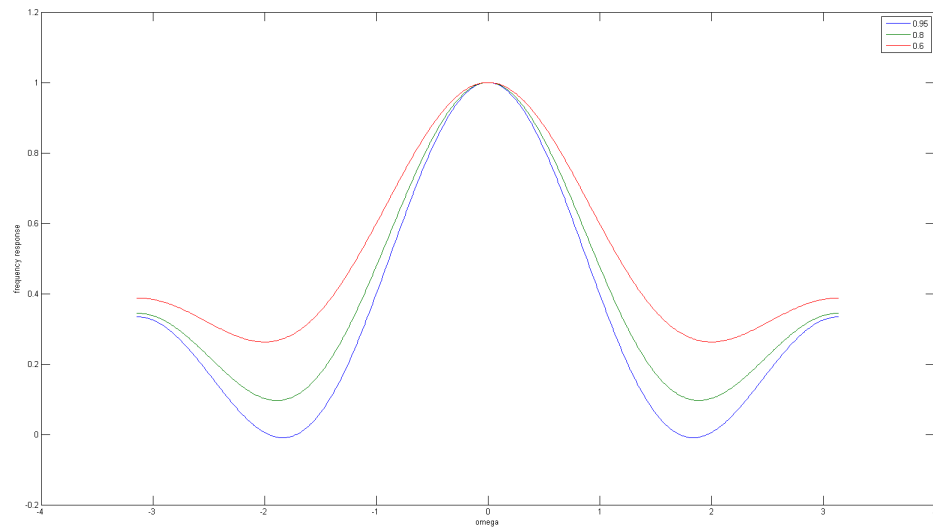
$$y[n] = (0.512) (0.8x[n] + 0.64x[n-1] + 0.512x[n-2])$$

(b) Adding in the appropriate coefficients from Lab 3's prelab yields

$$H[k] = \frac{1}{\sum_{i=1}^3 \alpha^i} (\alpha^1 + \alpha^2 e^{-jk\omega} + \alpha^3 e^{-2jk\omega})$$

$$H[k] = (0.512) (0.8 + 0.64e^{-jk\omega} + 0.512e^{-2jk\omega})$$

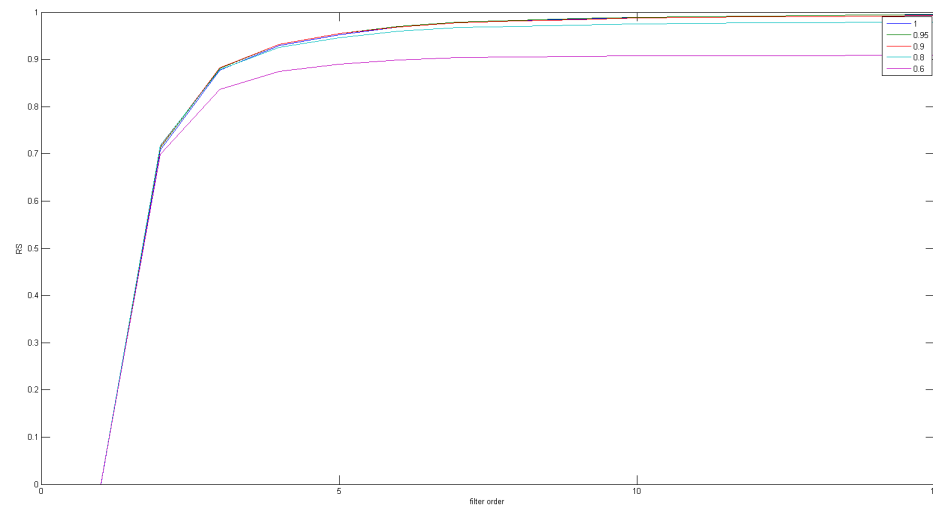
(c)



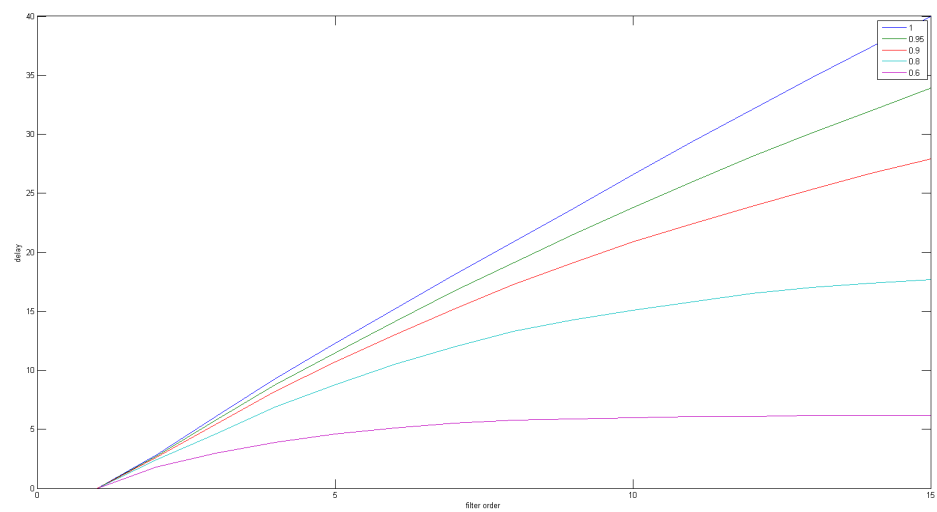
As  $\alpha$  decreases the frequency response gets closer to becoming uniform with respect to time.

(d)

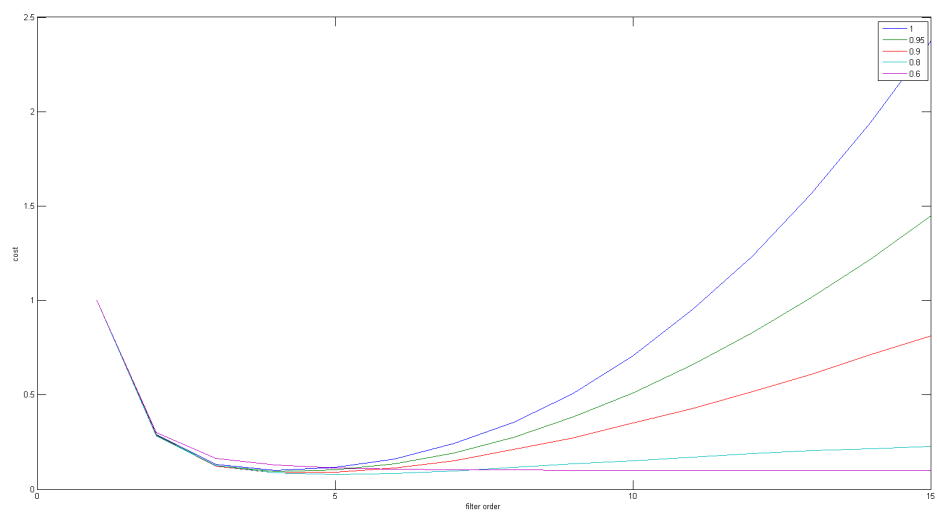
Smoothness



Delay



Cost



---

```

function [ y ] = movingaverage( x, order, alpha )
    y = zeros(1,length(x));
    alphas = alpha.^(1:order)';
    alphasSum = sum(alphas);
    for j = 1:length(x)
        if j >= order
            y(j) = sum(x(j:-1:j-(order-1)) .* alphas) / alphasSum;
        else
            y(j) = sum(x(j:-1:1) .* alphas(1:j)) / alphasSum;
        end
    end
end

% SCRIPT
alphas = [1, 0.95, 0.9, 0.8, 0.6];
RS = zeros(length(alphas), 15);
d = zeros(length(alphas), 15);
cf = zeros(length(alphas), 15);
for j = 1:length(alphas)
    for i = 1:15
        y = movingaverage(x,i,alphas(j));
        RS(j,i) = rs(x,y);
        d(j,i) = delay(t,x,y);
        cf(j,i) = CF(d(j,i),RS(j,i));
    end
end

plot(1:15, RS(1,:), 1:15, RS(2,:), 1:15, RS(3,:), 1:15, RS(4,:), 1:15, RS(5,:));
legend('1', '0.95', '0.9', '0.8', '0.6')
xlabel('filter order')
ylabel('delay')

plot(1:15, d(1,:), 1:15, d(2,:), 1:15, d(3,:), 1:15, d(4,:), 1:15, d(5,:));
legend('1', '0.95', '0.9', '0.8', '0.6')
xlabel('filter order')
ylabel('delay')

plot(1:15, cf(1,:), 1:15, cf(2,:), 1:15, cf(3,:), 1:15, cf(4,:), 1:15, cf(5,:));
legend('1', '0.95', '0.9', '0.8', '0.6')
xlabel('filter order')
ylabel('cost')

```

---

- (e) The contribution from long past events becomes negligible for large  $M$  when  $\alpha = 0.6$  because the coefficient exponential decays very quickly.
- (f) The updated filter is superior to the filter designed in lab 3. The updated filter is less smooth than the filter designed in lab as  $\alpha$  decreases, but the  $M = 5$  updated filter is smoother than the  $M = 4$  lab filter. The updated filter is also less subject to delay than the lab filter due to the decreasing weights for further past events, so for the same order the updated filter has a lesser delay. Thus according to our cost function the updated filter is the better choice.

4. (a)

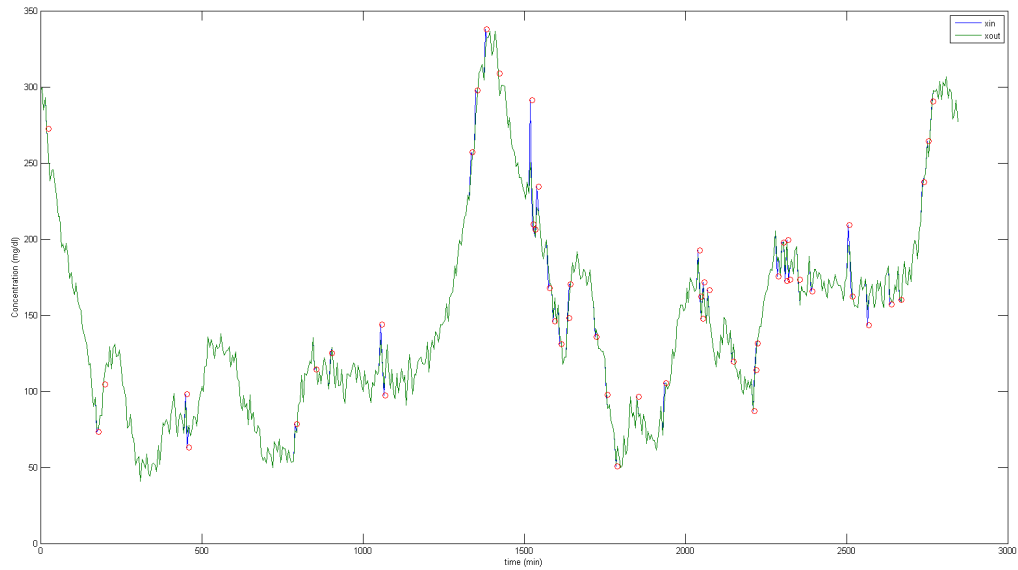


Figure 2: Continuous glucose monitor with overlay of signal with artifacts removed. Points where sensor input was altered is highlighted with red circles.

---

```
function [xout, changes] = cgmprefilter(xin, t)
    len = length(t);
    xout = zeros(len, 1);
    changes = zeros(len, 1);
    xout(1) = xin(1);
    for i = 2:length(t)
        diff = xin(i) - xout(i-1);
        if (abs(diff) > 20)
            changes(i) = 1;
            if (diff > 0)
                xout(i) = xout(i-1) + 20;
            else
                xout(i) = xout(i-1) - 20;
            end
        else
            xout(i) = xin(i);
        end
    end
end
```

---

(b) Input data altered at 52 locations.