Discrete Time Convolution

11/8/2022

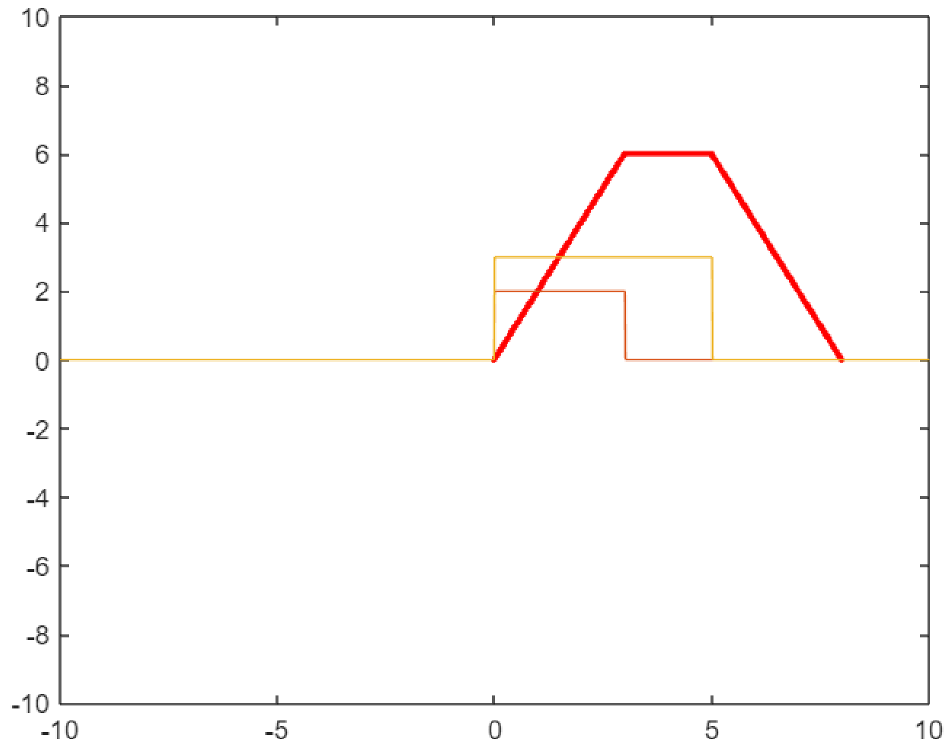
Experimenting with the Discrete Time Convolution, inspired by my Signal Processing Course.

```
clear
clc
fs = 100
```

```
fs = 100
```

```
t = (-100+1/fs:1/fs:100-1/fs);
U = @(t) 2 .* (t >= 0 & t <= 3);
V = @(t) 3 .* (t >= 0 & t <= 5);
w = DTC(U,V,t,fs)
```

```
UU = 1×19999
     0     0     0     0     0     0     0     0     0     0     0     0     0⋯
VV = 1×19999
     0     0     0     0     0     0     0     0     0     0     0     0     0⋯
toff = 1×801
         0    0.0100    0.0200    0.0300    0.0400    0.0500    0.0600    0.0700⋯
```



```
w = 1×801
    0.0600    0.1200    0.1800    0.2400    0.3000    0.3600    0.4200    0.4800⋯
```

```
function WW = DTC(u,v,t,fs)
uvec = [u(t)];
```

```
vvec = [v(t)];
offU = offset(uvec,t);
offV = offset(vvec,t);
offUfunc = @(t) u(t-(-(offU)));
offVfunc = @(t) v(t-(-(offV)));
UU = [offUfunc(t)]
VV = [offVfunc(t)]
WW = DTConv(UU, VV) .* 1/fs;
    toff = (offU + offV) : 1/fs : 1./fs * length(WW) - 1/fs + (offU + offV)
    clc
    figure
    plot(toff,WW ./ 3,'r.'),hold on
    plot(t,uvec,t,vvec),hold off
    axis([-10, 10, -10, 10])
 function tmem = offset(vec,t)
        indmem = 1;
        for ind = 1:length(vec)
            if vec(ind) ~= 0
                indmem = ind;
                break
            end
        end

        tmem = t(indmem);
 end
    function neuW = DTConv(u,v)
        uu = [u, zeros(1,length([u,v]) - length(u) - 1)];
        vv = [v, zeros(1,length([u,v]) - length(v) - 1)];
        w = zeros(1,length([u,v]) - 1);
        for k = (1:length([u,v]) - 1)
            s = 0;
            for j = (1:k)
                s = s + uu(j) * vv(k - j + 1);
            end
            w(k) = s;
        end
        neuW = [];
        neuW = w(w ~= 0);
    end

end
```