BCA Replication: Corrected Time Domain Targeting Comparison with Frequency Domain

Matthew DeHaven

August 15, 2023

Original Business Cycle Anatomy Figure 15a (Online Appendix)

The Business Cycle Anatomy paper argues that targetting a period of 6 to 32 quarters in the frequency domain is **not** the same as targeting a horizon of 6 to 32 quarters in the time domain. They make this argument largely through comparing the IRFs of shocks targetting different time horizons, finding that the one most closely matching the frequency domian identified shock targets a horizon of 4 quarters. This can be seen in Figure 15a, included below, where the black dotted line is closer to the solid black line than the dashed diamond line.

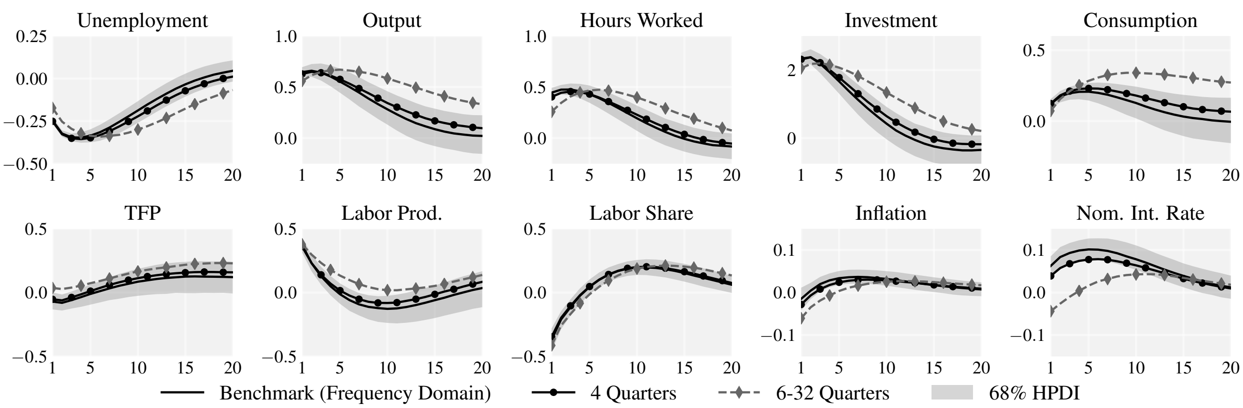

Figure 1: Frequency-Domain vs Time-Domain Identification

Mistargetted Time Domain Horizon

The authors, however, have a coding error that means their time domain targetting does not function as expected. Instead, their code will always target the longest time horizon, rather than the full range. So the dashed diamond line above is actually targetting a horizon of 32 quarters, rather than 6 to 32 quarters.

Coding Error

This coding error can be seen in the Julia code copied below:

```
function functdq(q0,q1,Vtmp1,idx,maxhor;zero_impact=false)
    q0    = q0+1
    q1    = q1+1
    smpl  = (idx[1]-1)*maxhor+1:idx[1]*maxhor
    Itmp  = Vtmp1[smpl,:]
```

```
    n     = size(Vtmp1,2)
    if q1>q0
        V     = zeros(n,n);
        for j=q0:q1
            V0= zeros(n,n); ## These lines should be outside the for loop
            V1= 0           ##
            for k=1:j
                V0 = V0+Itmp[k,:]*Itmp[k,:]'
                V1 = V1+Itmp[k,:]'*Itmp[k,:]
            end
        end
    else
        V0= zeros(n,n)
        V1= 0
        for k=1:q0
            V0 = V0+Itmp[k,:]*Itmp[k,:]'
            V1 = V1+Itmp[k,:]'*Itmp[k,:]
        end
    end
    V               = V0/V1;
    if zero_impact
        D,P         = eig(V[2:n,2:n])
        i           = indmax(abs.(D))
        Q           = vcat(0,P[:,i])
    else
        D,P         = eig(V)
        i           = indmax(abs.(D))
        Q           = P[:,i]
    end
    return Q
end
```

Starting from the end of the function, this code identifies the correct rotation matrix Q that maximizes the FEV contribution at a specified time domain horizon. The matrix Q is found by taken the largest eigen value of the matrix V, which measures the FEV. The matrix V is calculated by summing each FEV for each horizon–the matrices V0. However, at the commented lines it is clear to see that V0 is being overwritten with 0 each time through the loop. Thus only the final value for j is counted in the sum. This is when $j = q1$, the maximum horizon passed in by the user.

How much does it matter?

Using the replicated Bayesian VARs, we can look at how much of a difference targetting just the horizon of 32 matters for the BCA VAR. It turns out that the IRFs are nearly indistinguishable. The differences seen are small enough that the could likely change just given a different random seed and the resulting random variation in Bayesian estimation.

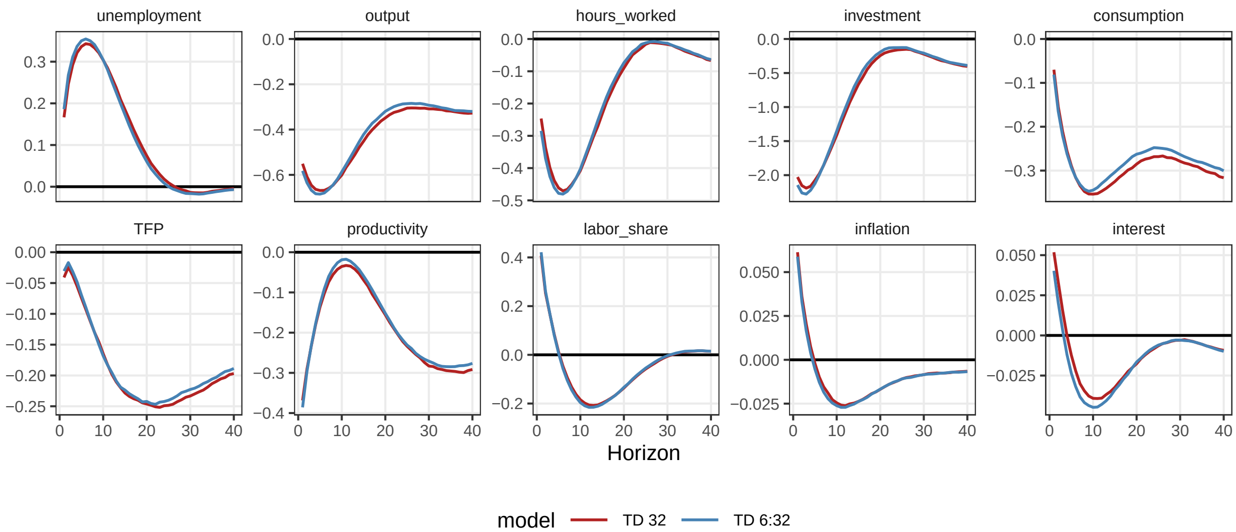

Figure 2: Time Domain Targetting 6:32 vs 32

Corrected Figure

Putting this all together into a corrected version of Figure 15a, we can see that the Time Domain targeting 4 quarters is still extremely close to the actual benchmark FD IRFs, while there is barely a difference between the one that targets a horizon of 32 or of 6:32.

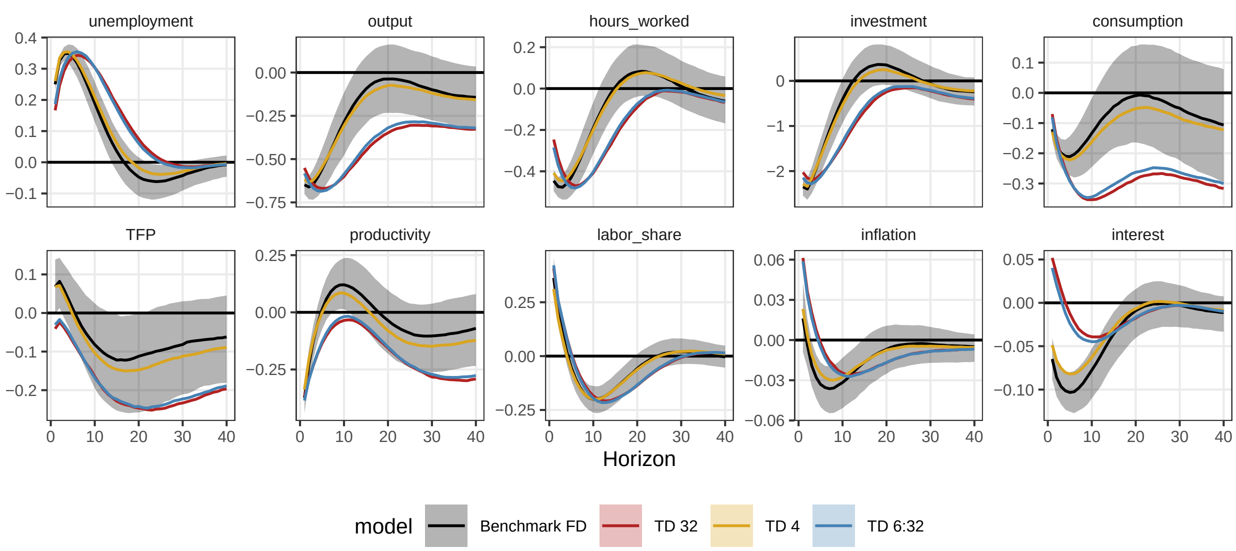

Figure 3: Frequency-Domain vs Time-Domain Identification