A Julia Implementation of Gensys sourcecode

Gilberto Boaretto & Daniel Coutinho

October, 2019

1 Code

Here is the code

```
function gensys(G0,G1,Psi,Pi)
    decomp_1 = schur(G0,G1)
    gen_eigen = abs.(decomp_1.beta ./ decomp_1.alpha)
    ordschur!(decomp_1, gen_eigen .< 1)
    n = size(G0,1)
    ns = findfirst(sort(gen_eigen) .> 1) -1 #finding the number of stable roots: find
first unstable root
    nu = n - ns
    S11 = decomp_1.S[1:ns,1:ns]
    S12 = decomp_1.S[1:ns,(ns+1):n]
    S22 = decomp_1.S[(ns+1):n,(ns+1):n]

    T11 = decomp_1.T[1:ns,1:ns]
    T12 = decomp_1.T[1:ns,(ns+1):n]
    T22 = decomp_1.T[(ns+1):n,(ns+1):n]

    Qt = decomp_1.Q'

    Q1 = Qt[1:ns,:]
    Q2 = Qt[(ns+1):n,:]

    Q2Pi = Q2*Pi #This is equation 2.25 in p. 46 Miao (2014)

    m = size(Q2Pi,2)

    svd_Q2Pi = svd(Q2Pi)
    r = size(svd_Q2Pi.S)[1] #S is a vector

    #Checking existence and uniqueness see p. 46-47, Miao (2014)
    if m > r
        eu = [1;0]
        @warn "No Unique Solution"
    elseif m < r
        eu = [0;0]
        @warn "No solution"
    else
        eu = [1;1]
        @info "Unique and Stable Solution"
        U1 = svd_Q2Pi.U[:,1:r]
        Xi = Q1*Pi*svd_Q2Pi.V*inv(Diagonal(svd_Q2Pi.S))*U1' #bottom of p 46
        Aux1 = S12-Xi*S22

        larg1 = size(Aux1,2)
        Aux2 = [S11 Aux1;zeros(larg1,size(S11,2)) Matrix(I,larg1,larg1)]
        larg2 = size(Aux2,2)
        larg2 = larg2 - size(T11,1)

        ##Matrices on top of p. 46

        Theta1 = [T11 T12-Xi*T22;zeros(larg2,n)]
        Theta1 = decomp_1.Z*inv(Aux2)*Theta1*decomp_1.Z'
        Theta2 = [Q1 - Xi*Q2;zeros(larg2,n)]
        Theta2 = decomp_1.Z*inv(Aux2)*Theta2*Psi
        Theta3 = zeros(n,n)
        ans = Sims(Theta1,Theta2,Theta3,eu)
        return ans#Theta1,Theta2,eu
    end
end
```

```
gensys (generic function with 1 method)
```