July 2021

01211433 Homework 2

Figure 1 shows a Scilab/Xcos simulation diagram for output disturbance attenuation at the plant output

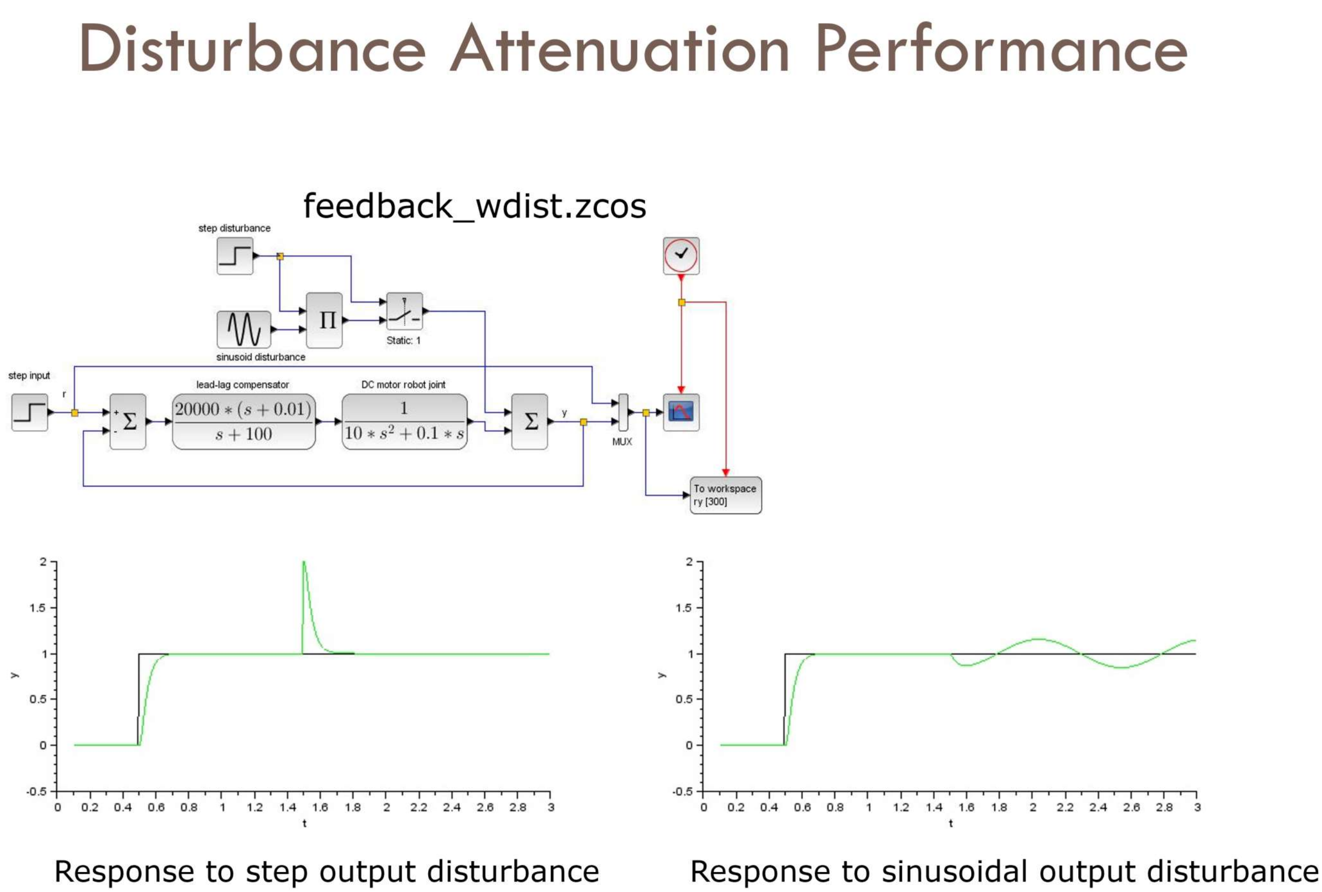


Figure 1 Scilab/Xcos diagram for output disturbance attenuation

Create similar simulation setup in Julia using the same plant and controller as in Figure 1; i.e.,

$$P(s) = \frac{1}{10s^2 + 0.1s}$$

$$C(s) = \frac{20000(s+0.01)}{(s+100)}$$

Since only disturbance attenuation performance is of interest, the step input can be discarded. The expected outputs are

1. For step input disturbance, the output should approach zero at steady state.
2. For low-frequency sinusoid disturbance, say, about 1 rad/s, output should be smaller in magnitude.

Remarks :* For simplicity, the disturbance signal may start at t = 0.

Solution :

If ControlSystems is not installed, uncomment the cell below

```
#using Pkg
#Pkg.add("ControlSystems")
```

Use the packages

```
using ControlSystems   , Plots   ,PlutoUI
```

Create the plant and controller

```
TransferFunction{Continuous, ControlSystems.SisoRational{Float64}}
20000.0s + 200.0
----------------
  1.0s + 100.0

Continuous-time transfer function model
```

```
begin
    # your code
    s = tf("s")
    P = 1/(10s^2+0.1s)
    C = 20000(s+0.01)/(s+100)
end
```

Determine and contstruct the transfer function from output disturbance $d_o(s)$ to plant output $y(s)$

```
TransferFunction{Continuous, ControlSystems.SisoRational{Float64}}
     10.0s^3 + 1000.1s^2 + 10.0s
---------------------------------------
10.0s^3 + 1000.1s^2 + 20010.0s + 200.0

Continuous-time transfer function model
```

```
begin
    # your code
    L = C*P
    S = 1/(1+L)
end
```

Use plot_response() function below for plotting, with arguments

- sys : transfer functtion mapping d_o to y
- u : a vector of disturbance signal d_o of 1. nonzero constant value 2. low frequency sinusoid wave
- t : time vector

```
plot_response (generic function with 2 methods)
```

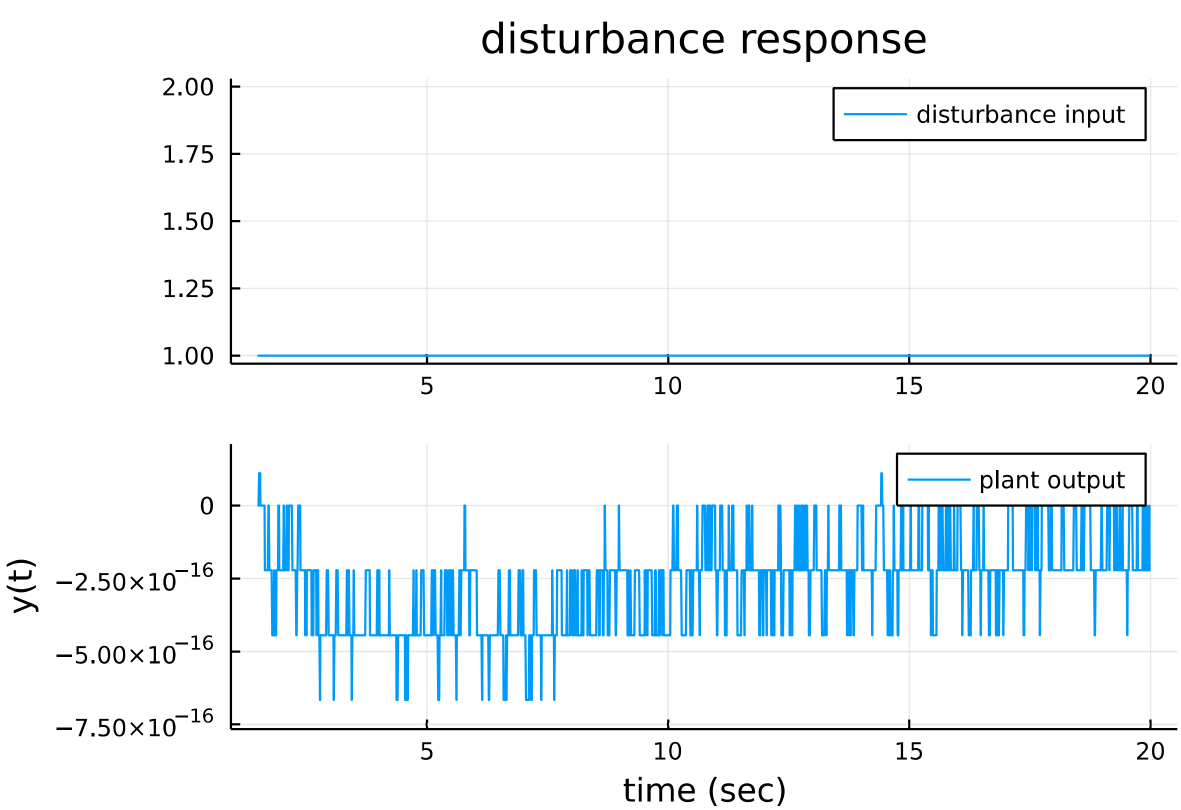
```
# Plotting function. No need to change anything.
function plot_response(sys,u,t, plottitle="disturbance")
    y, tout, x = lsim(sys, u, t,method=:zoh)
    t_idx = 150  # get rid of transient
    tout1 = tout[t_idx:end]
    y1 = y[t_idx:end]
    u1 = u[t_idx:end]

    uplt = plot(tout1,u1, label = plottitle*" input",title=plottitle*" response")
    yplt=plot(tout1,y1, label = "plant output",xlabel="time (sec)",ylabel="y(t)")
    plot(uplt, yplt, layout=(2,1))

end
```

1. Create time vector, and a constant disturbance input $d_o = c$

```
[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1
```

```
begin
    tvec = collect(Float64,0:0.01:20)# your code
    uvec = ones(size(tvec)) # cour code
end
```

Plot step disturbance response



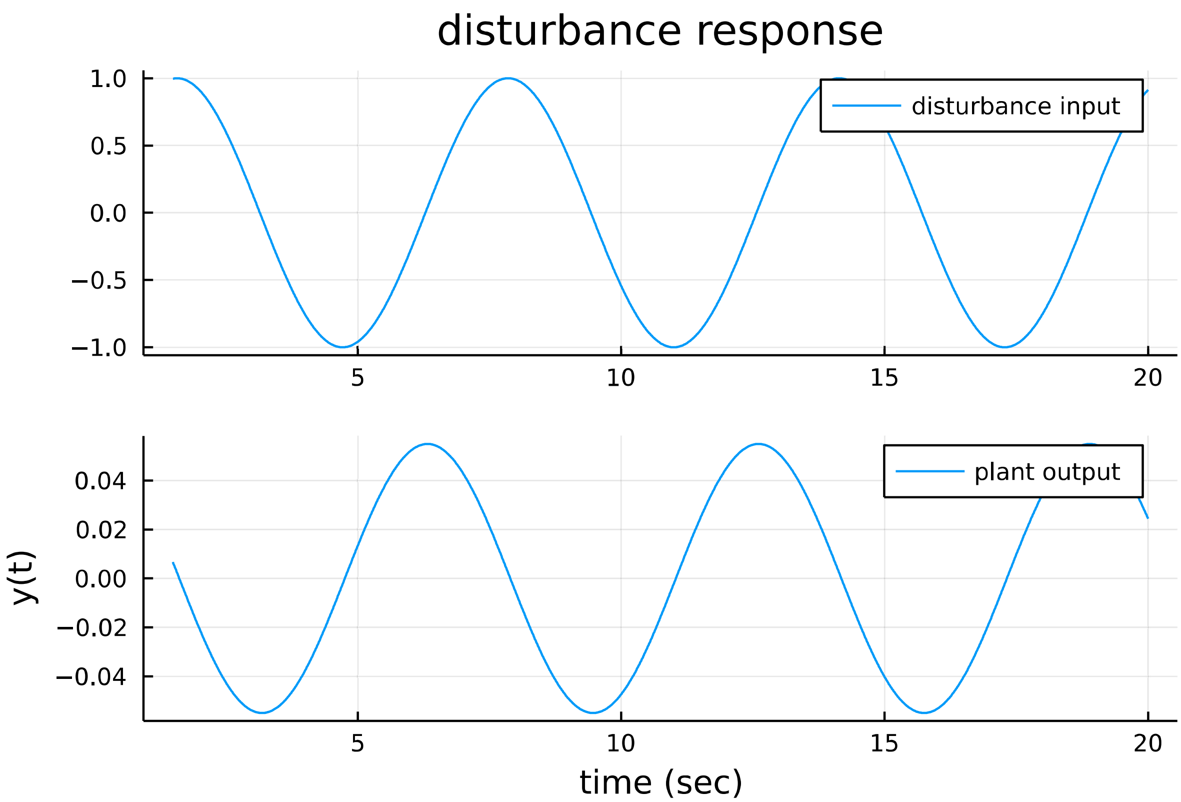

```
plot_response(S,uvec,tvec) # Replace ? with the right closed-loop transfer function
```

2. Use the time vector in 1. to construct a low-frequency sinusoid input vector (say, 1 rad/s)

```
[0.0, 0.00999983, 0.0199987, 0.0299955, 0.0399893, 0.0499792, 0.059964, 0.0699428, 0.07991
```

```
begin
    #w = 1 # rad/s
    uvec2 = uvec2 = sin.(w*tvec) # your code
end
```

Plot sinusoid disturbance response.




```
plot_response(S,uvec2,tvec) # # Replace ? with the right closed-loop transfer function
```

w = 1

```
md"""
w = $(@bind w Slider(1:100,show_value=true,default=1))
"""
```