
inverter-chain

Dan White

February 19, 2016

```
In [1]: # this turns on "hold on" mode in the interactive notebook
%config InlineBackend.close_figures=False

import numpy as np
import matplotlib as mpl

mpl.rcParams['axes.grid'] = True
mpl.rcParams['lines.linewidth'] = 3.0

close('all')
```

```
In [2]: close('all')
```

```
def total_delay(Cload, Cin, Cout, Rt, N):
    A = (Cload/Cin)**(1.0/N)
    #print A
    return 0.7 * N * Rt * (Cout + A*Cin)
```

```
def plot_curve(Cload):
    Cin = 1
    Cout = 1
    Rt = 1
    N = arange(1, 10, 0.1)
    Nint = arange(min(N), max(N), 1)

    Nopt = log(Cload / Cin)-1

    plot(N, total_delay(Cload, Cin, Cout, Rt, N), '-')
    plot(Nint, total_delay(Cload, Cin, Cout, Rt, Nint), 'ok')

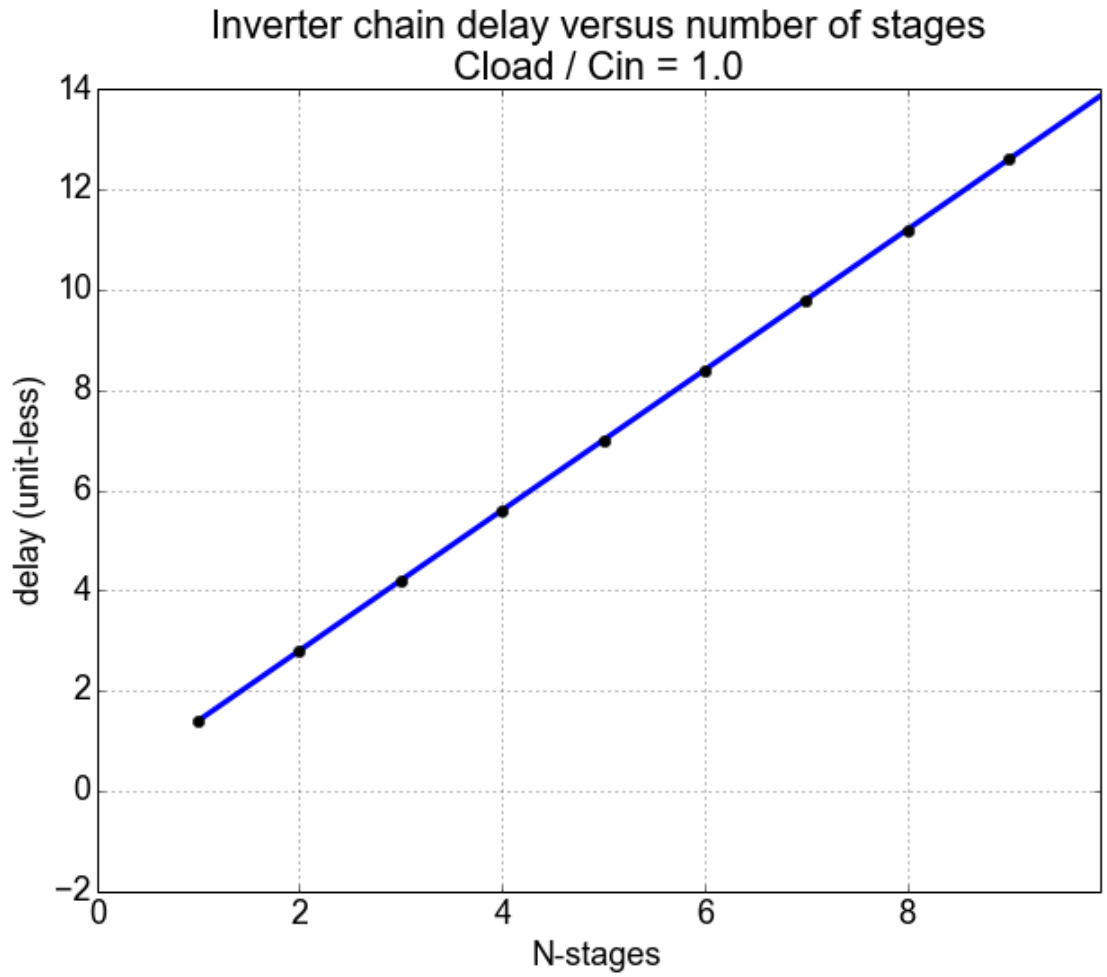
    plot([Nopt, Nopt], [0, total_delay(Cload, Cin, Cout, Rt, Nopt)], '-xr')

    xlabel('N-stages')
    ylabel('delay (unit-less)')
    title('Inverter chain delay versus number of stages\nCload / Cin = %.1f' % Cload)

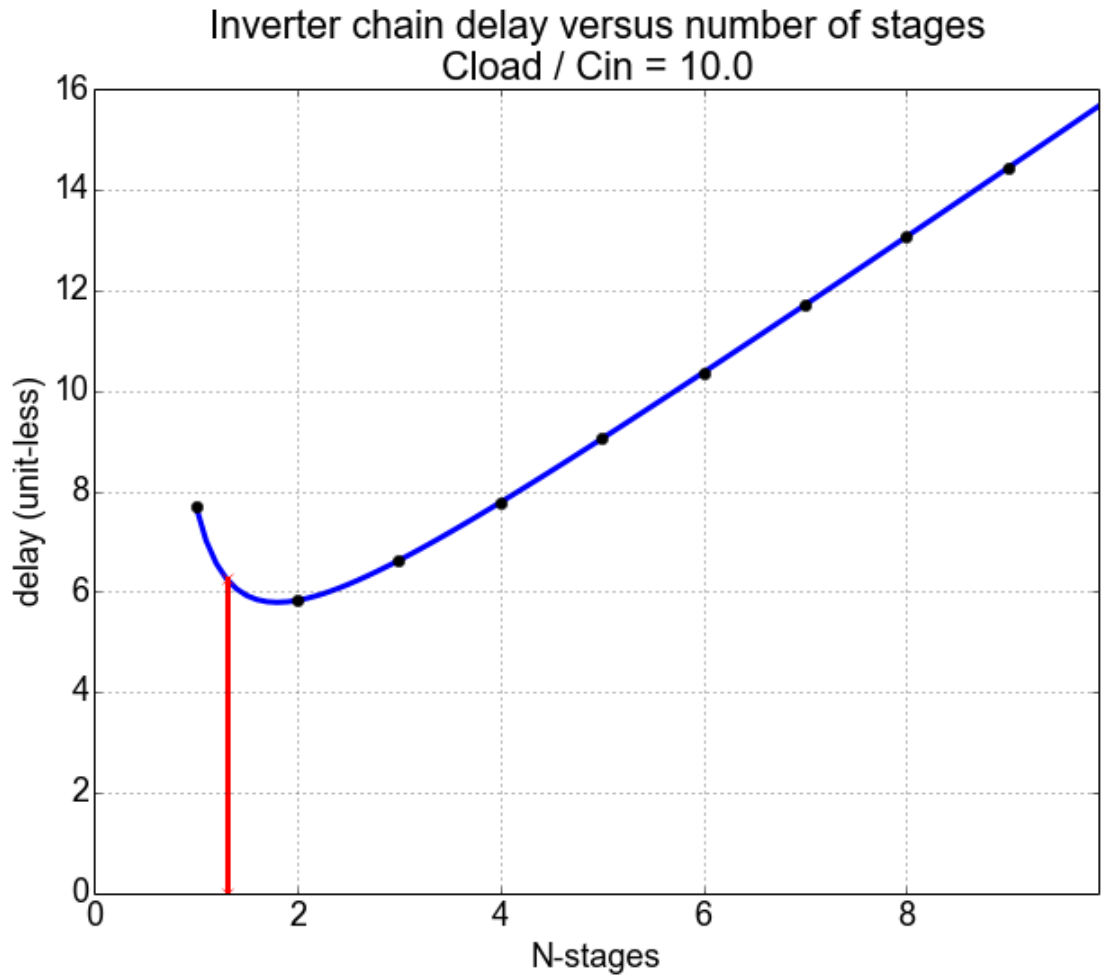
    print 'Nopt: ', Nopt
    print 'delay:', total_delay(Cload, Cin, Cout, Rt, Nopt)

    xlim([0, max(N)])
```

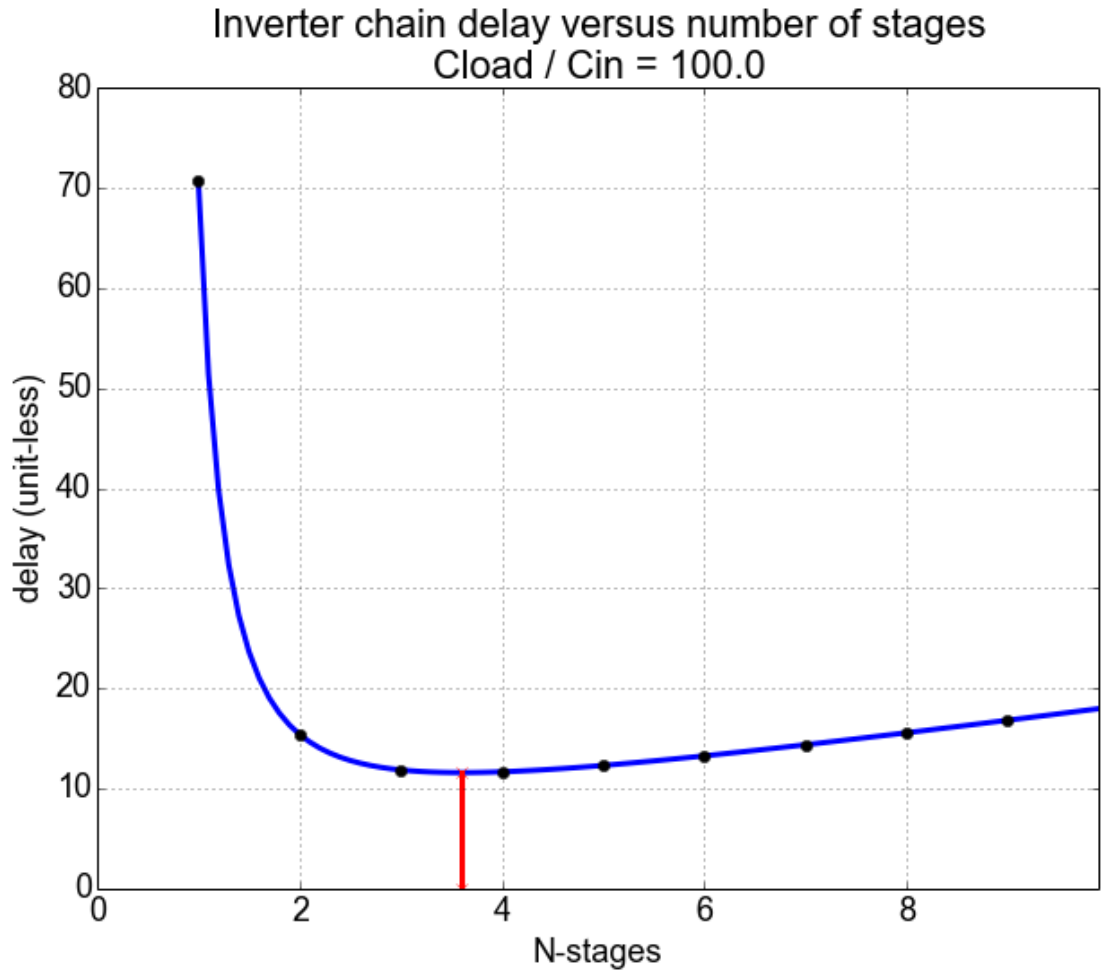
```
In [3]: figure()  
plot_curve(1)  
Nopt: -1.0  
delay: -1.4
```



```
In [4]: figure()  
plot_curve(10)  
Nopt: 1.30258509299  
delay: 6.25263407254
```

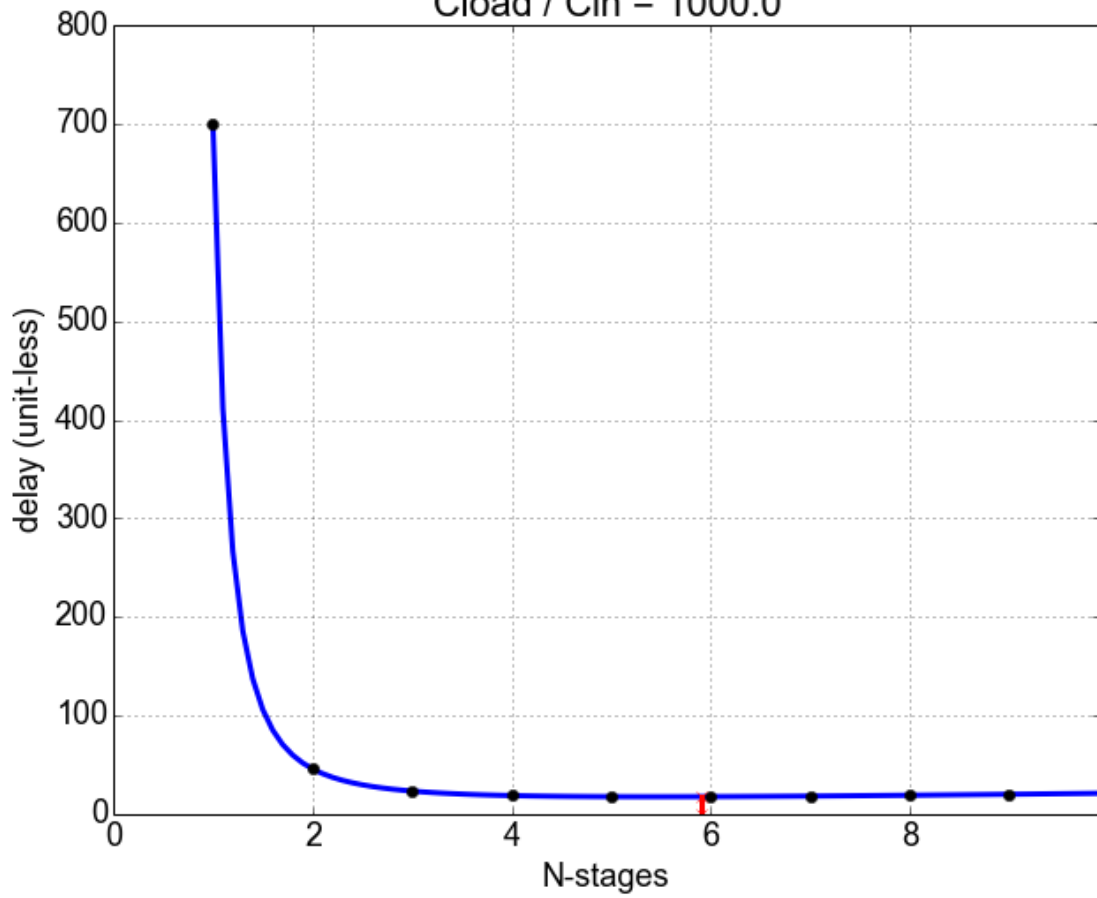


```
In [5]: figure()  
        plot_curve(100)  
        Nopt: 3.60517018599  
        delay: 11.5764132247
```



```
In [6]: figure()  
        plot_curve(1000)  
        Nopt: 5.90775527898  
        delay: 17.4500133762
```

Inverter chain delay versus number of stages
Load / Cin = 1000.0



In []:

In []: