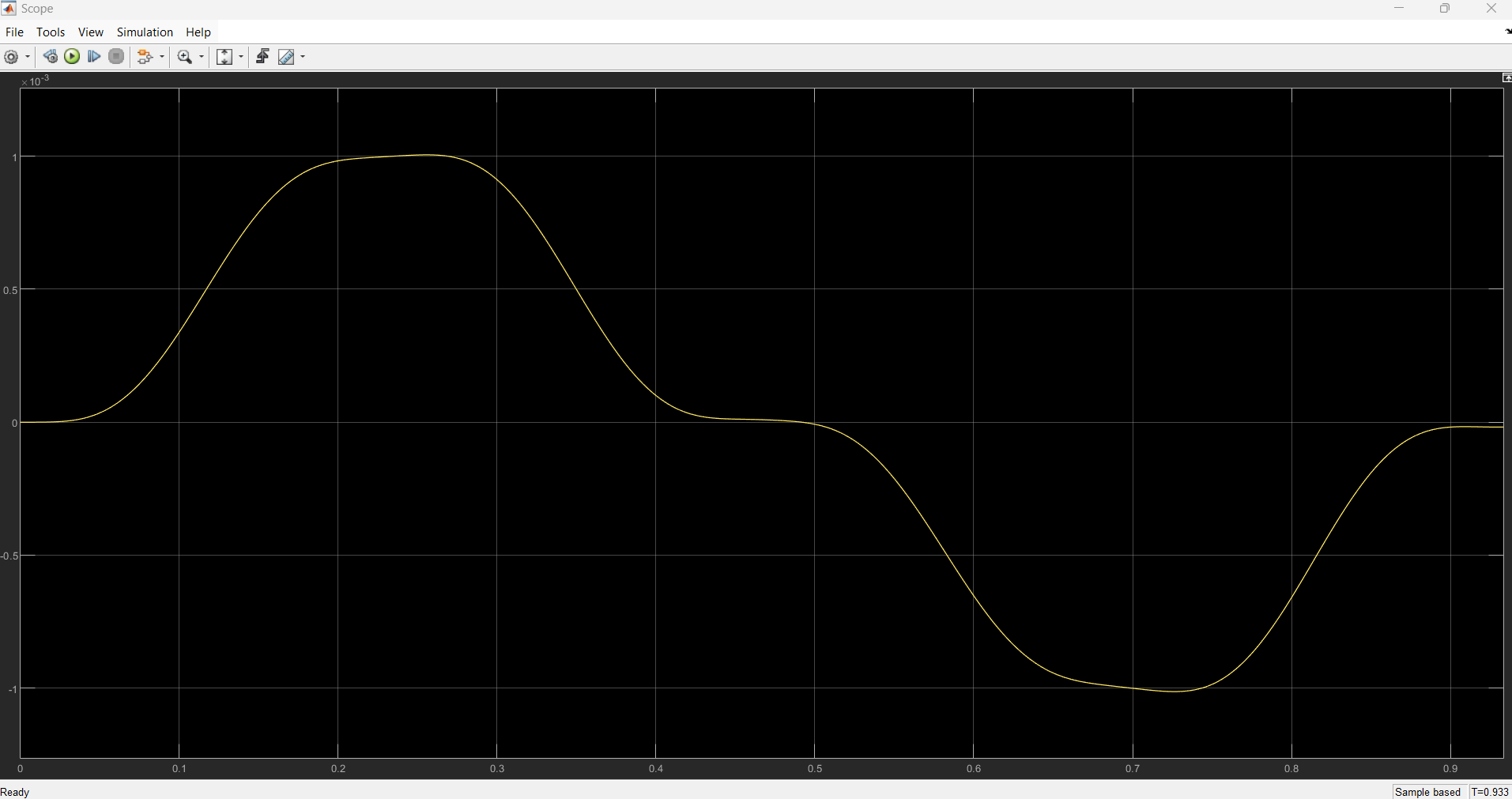
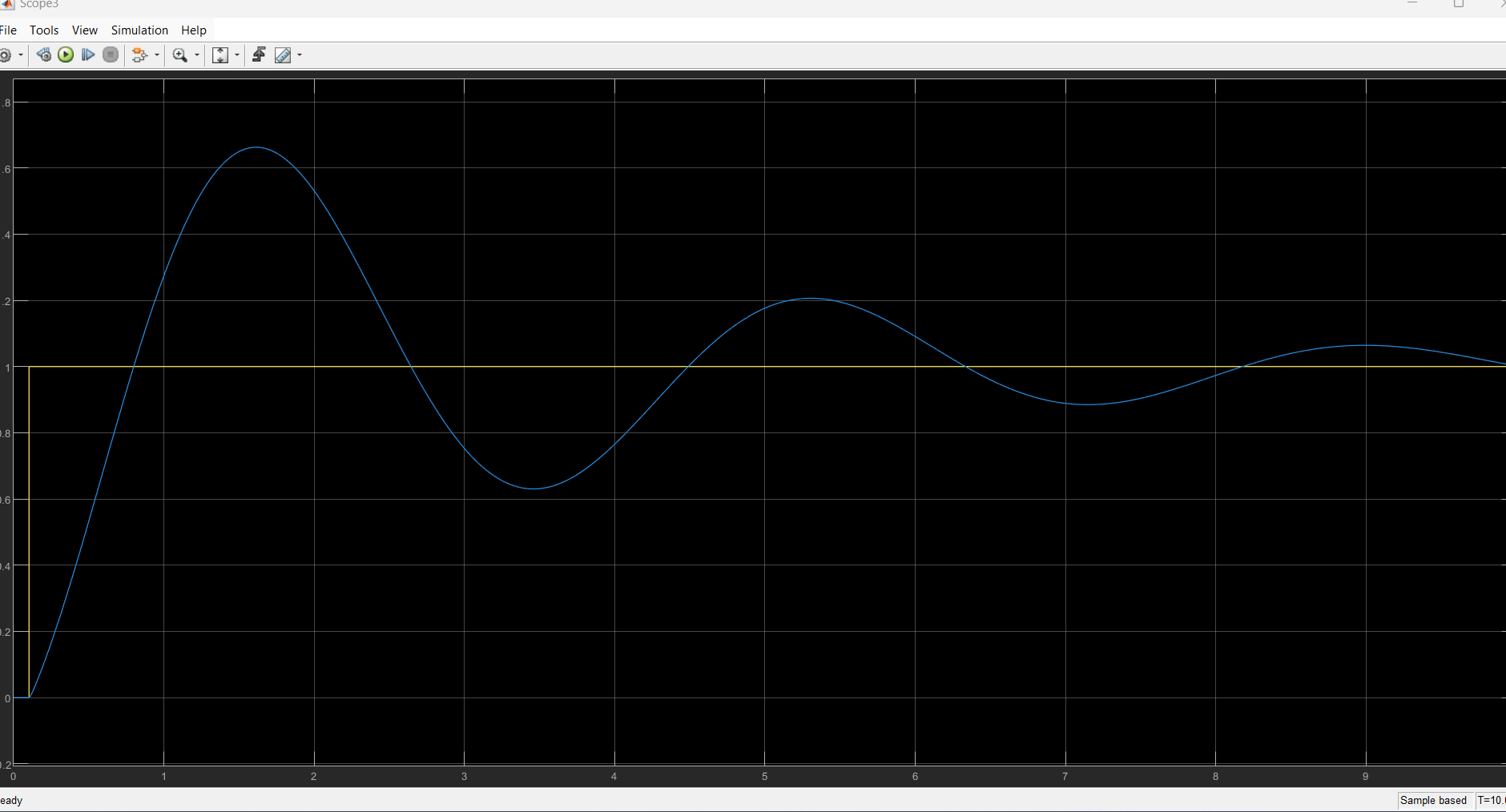
kNL = 0;



The below is for the linear system msd with only step input and interestingly tuning looks terribly had.

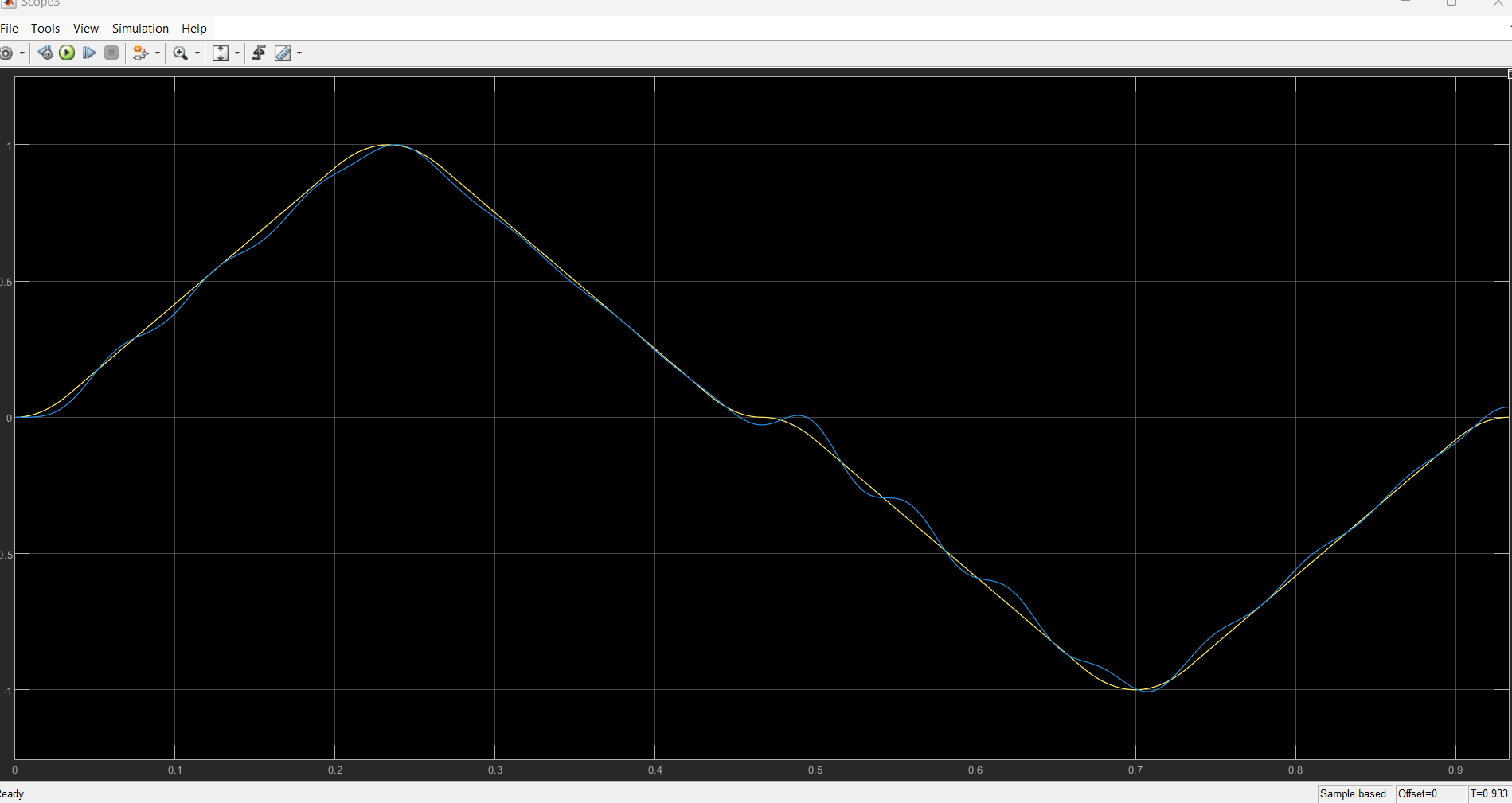
so it taught me that dynamics of the main system is very important.

Mass and spring coefficients are dramatically important: I just made the mass 100 times smaller and sring coeff 10 times smaller and look. Nice tracking



Another finding, if there is no NLY, tuning in this specific case terrible. So honestly, NLY helps in traj. Tracking. So NLY helps indeed for ref track.

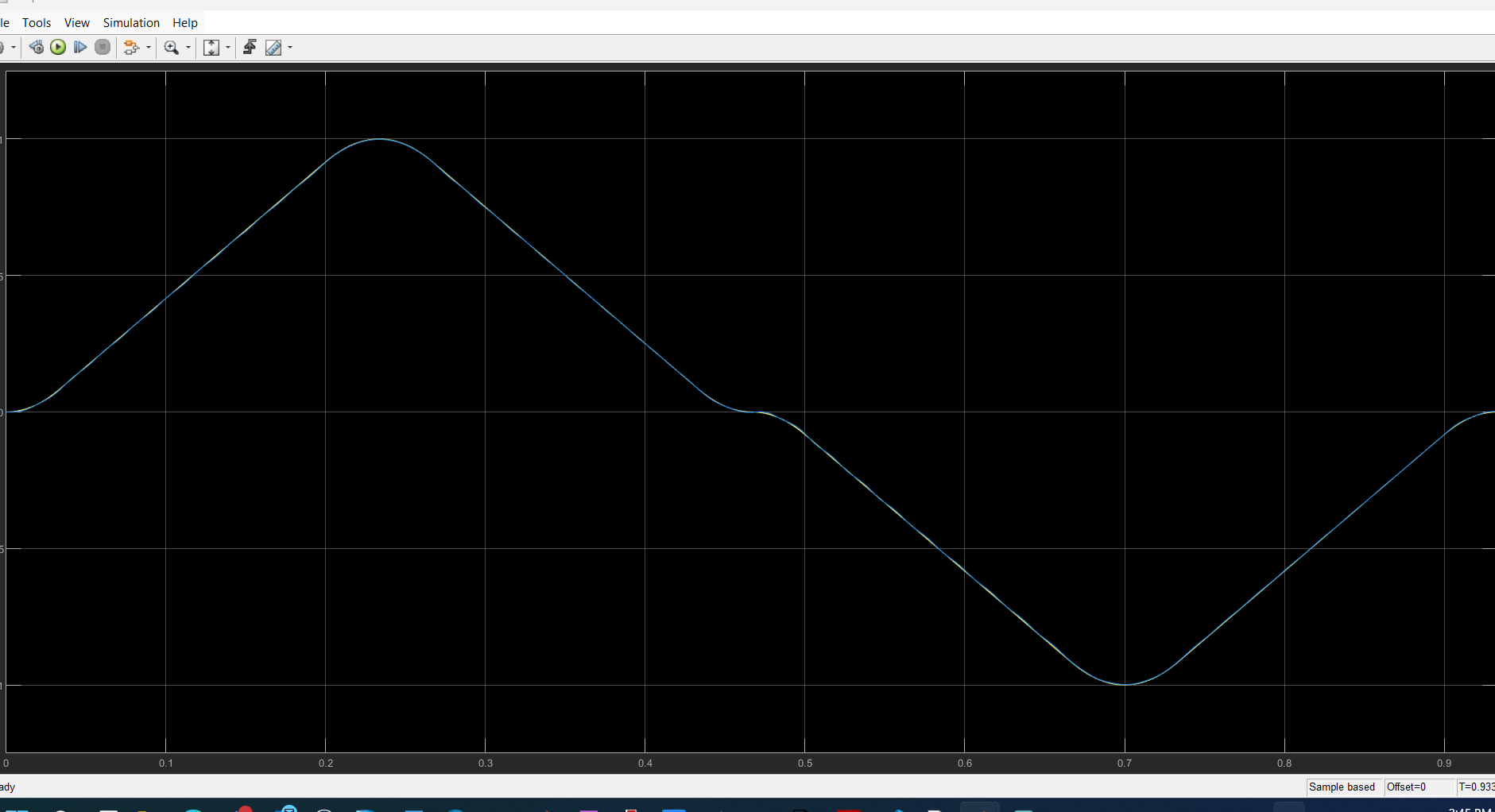
The best I could get is:



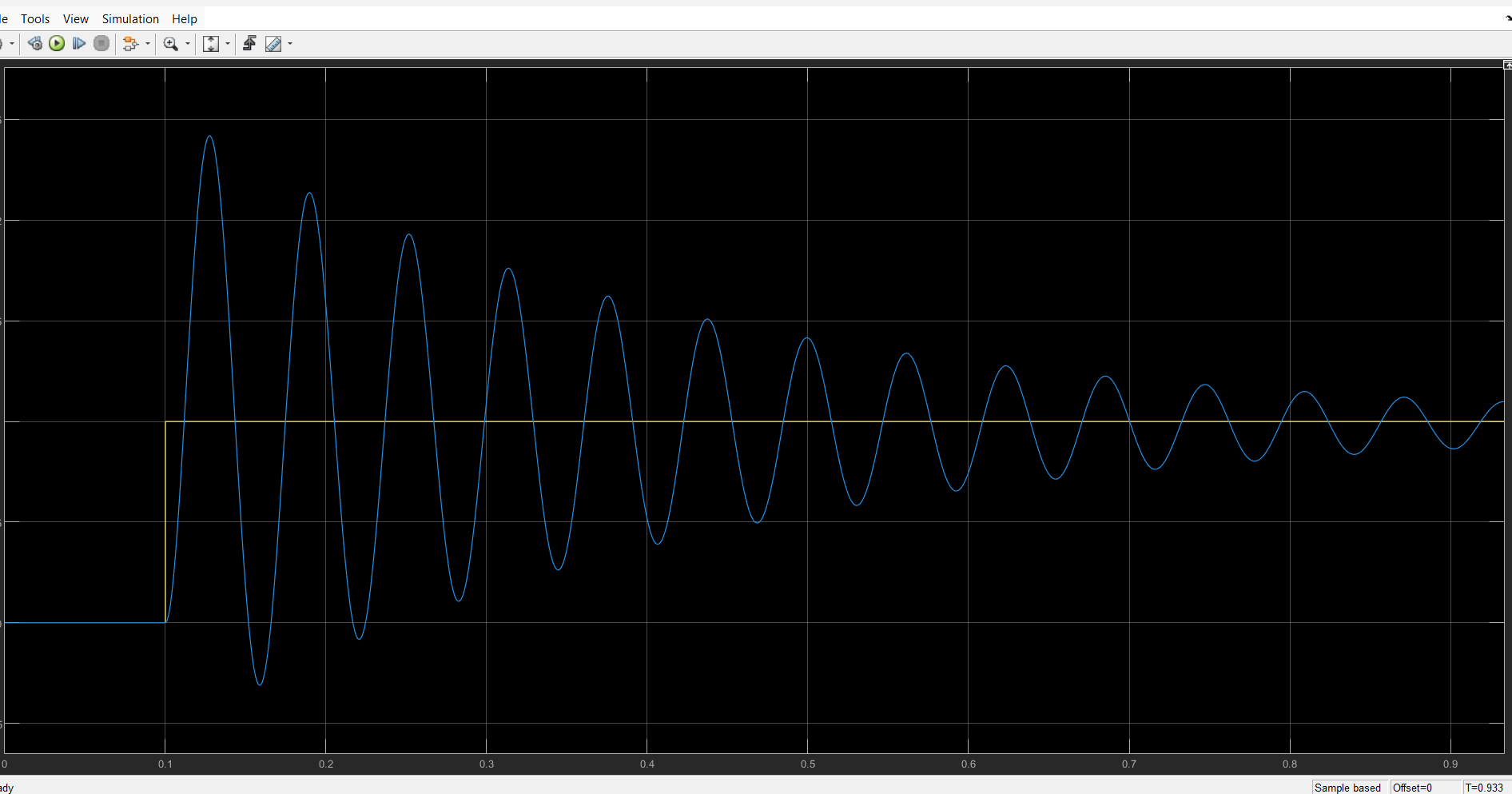
Best ref tracking for above with nly is: kp=100, ki=1, kd=0

So pid cannot do betther than this even with GA. So adopt NL ctrl laws.

Is also afcn of knl for knl=100; kp=100; for knl=1000; kp=1000 check the below:

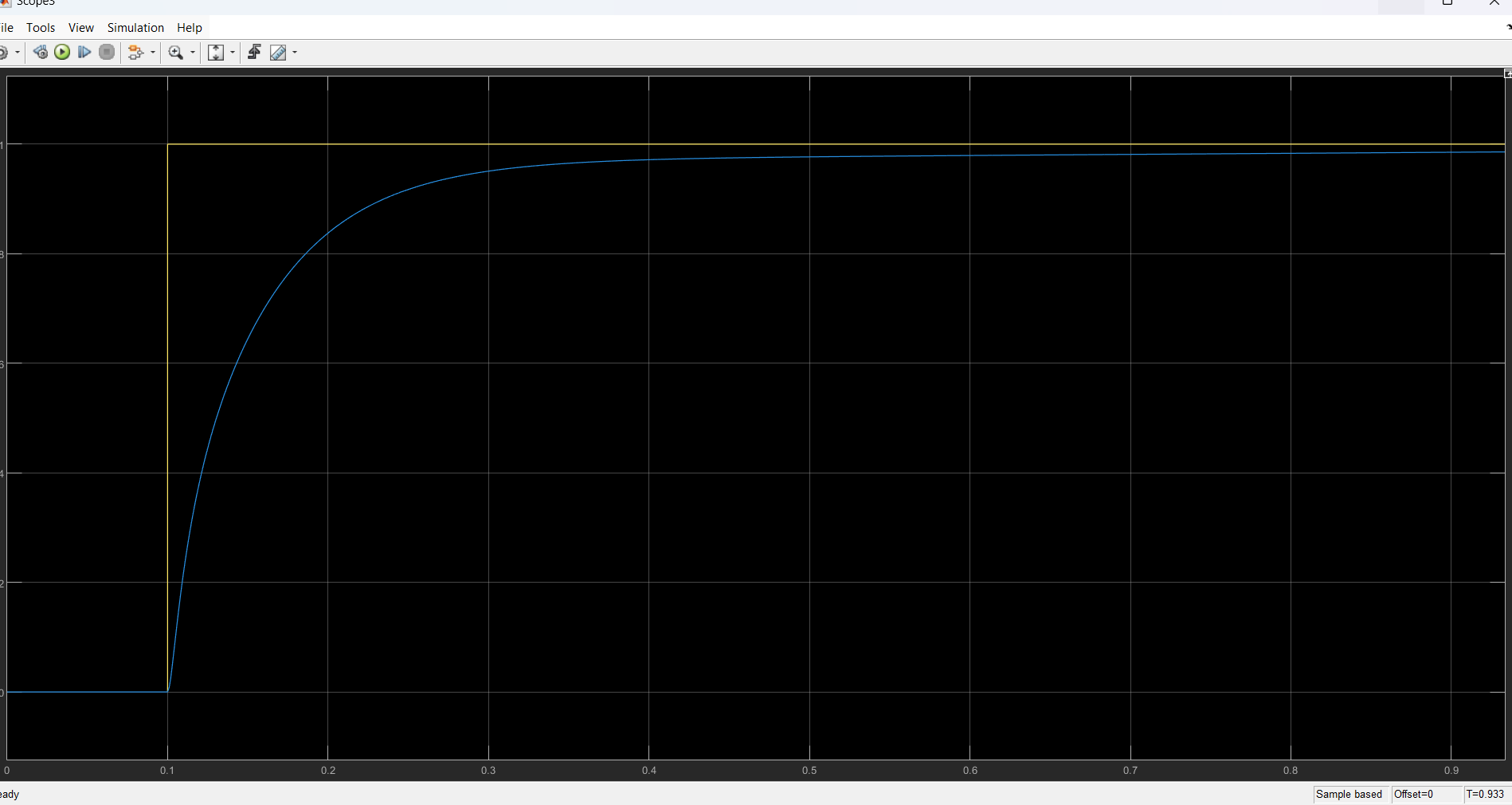


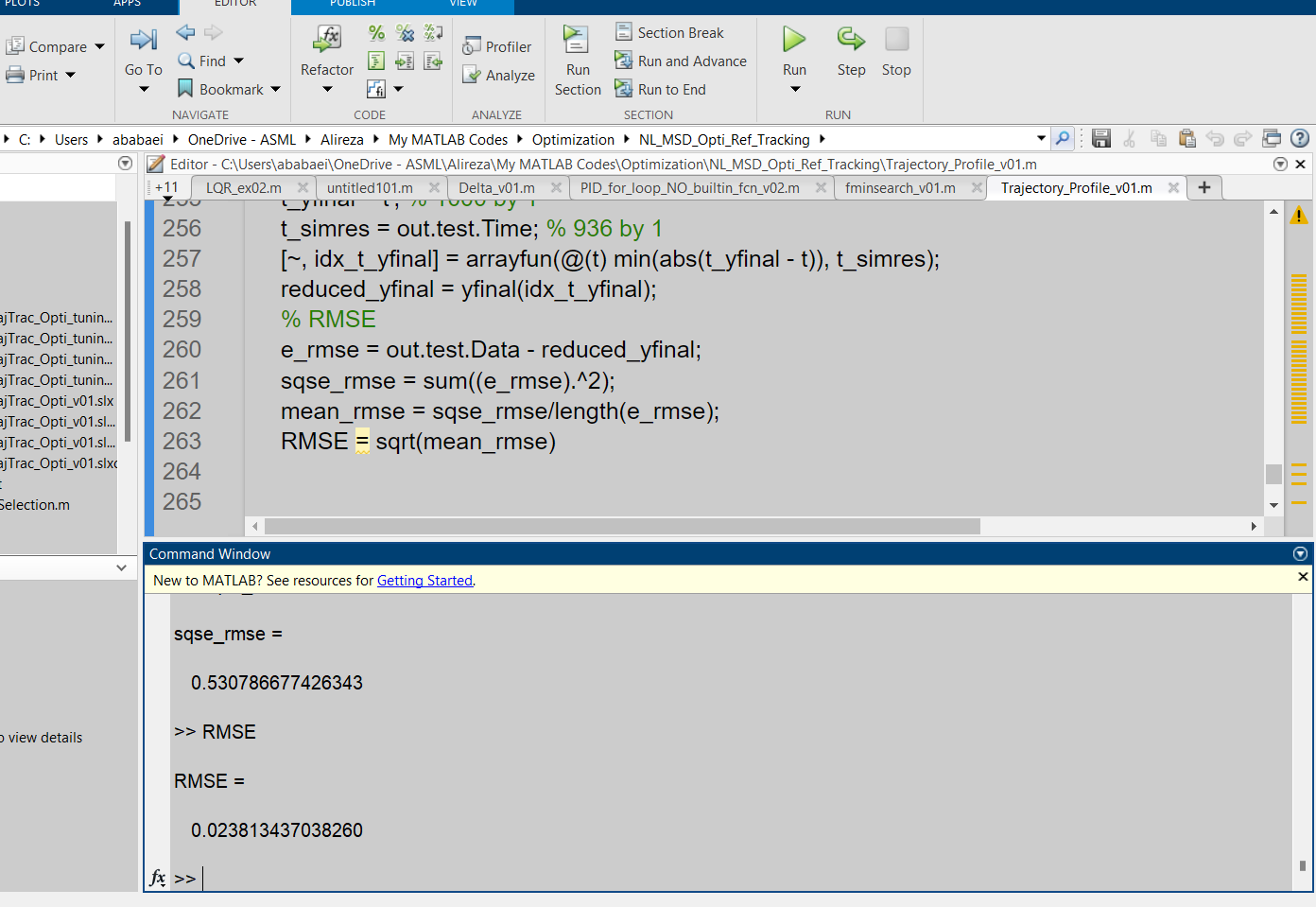
Though you might have said that linear pid works for NL system; but pay attention for that specific ref. traj. Yes

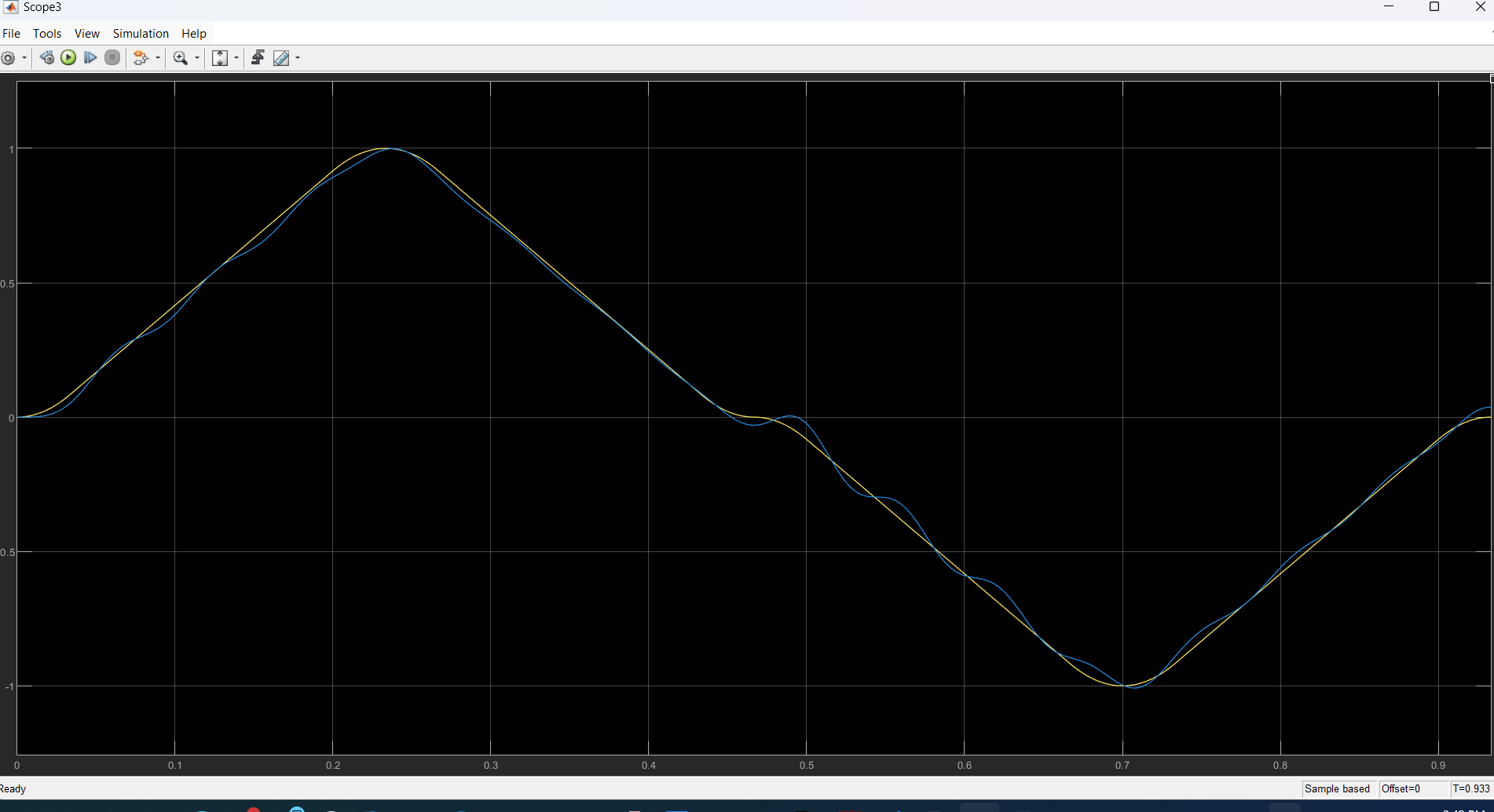


Check above for step. Also note that for very shirt time interval; it is very hard for a lightly-damped sys (zeta = 0.01) to track the ref without oscillations. This is the best ctrl with pid you could have. For better perf. Check NL ctrl laws. Of course, if you increase zeta iw till help dramatically. Particularly lightly-damped systems is very tricky to do nice ref tracking for a very short time interval as it naturally takes a long time for the transients to finish.

Following is with 100 times stronger damping:

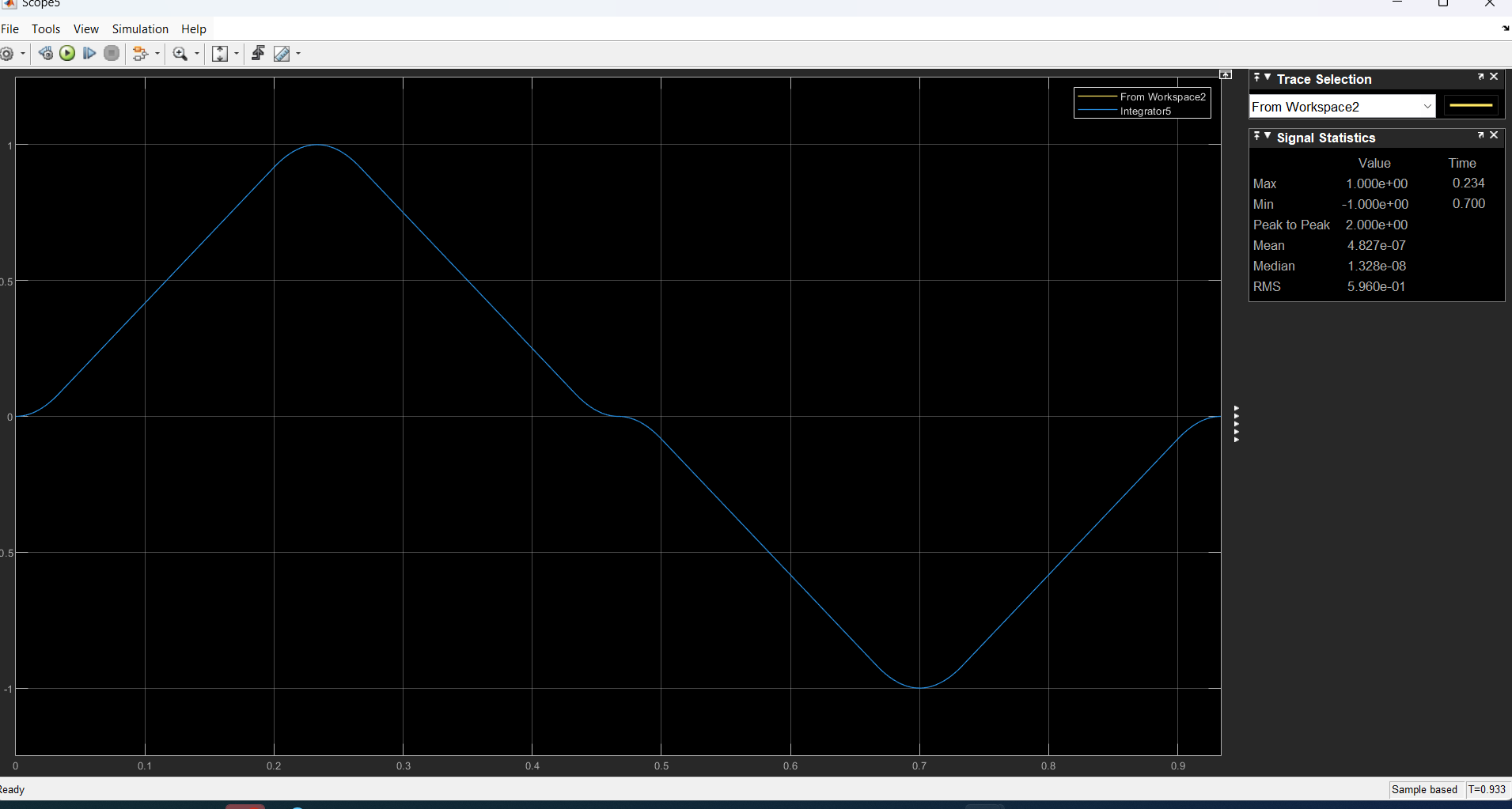
so system internal dynamics REALLY MATTER. Reference profile trajectory MATTERS, given time interval for tracking really MATTERS. Nonlinearity really MATTERS.



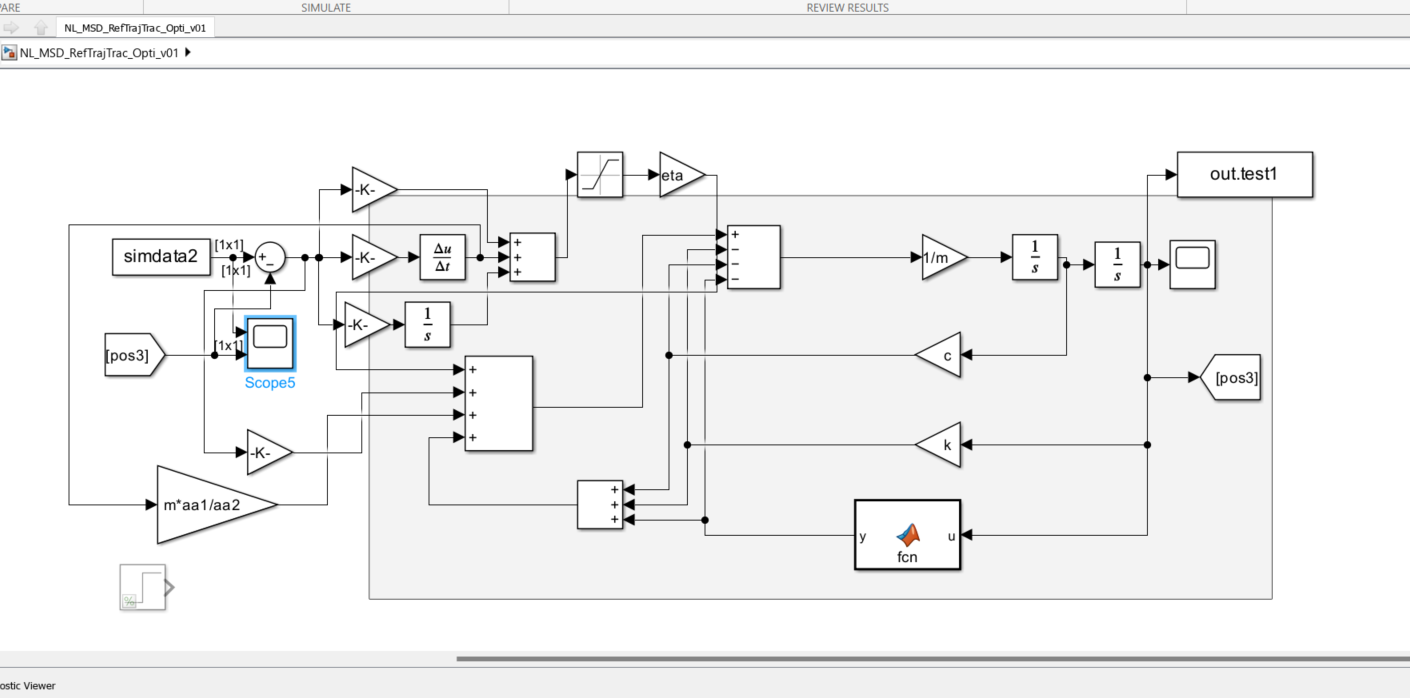


Goal with NL ctrl laws to see if there any improvements in the above which is for PID with NL dynamics.

Check the bellow: my own SMC works like this:

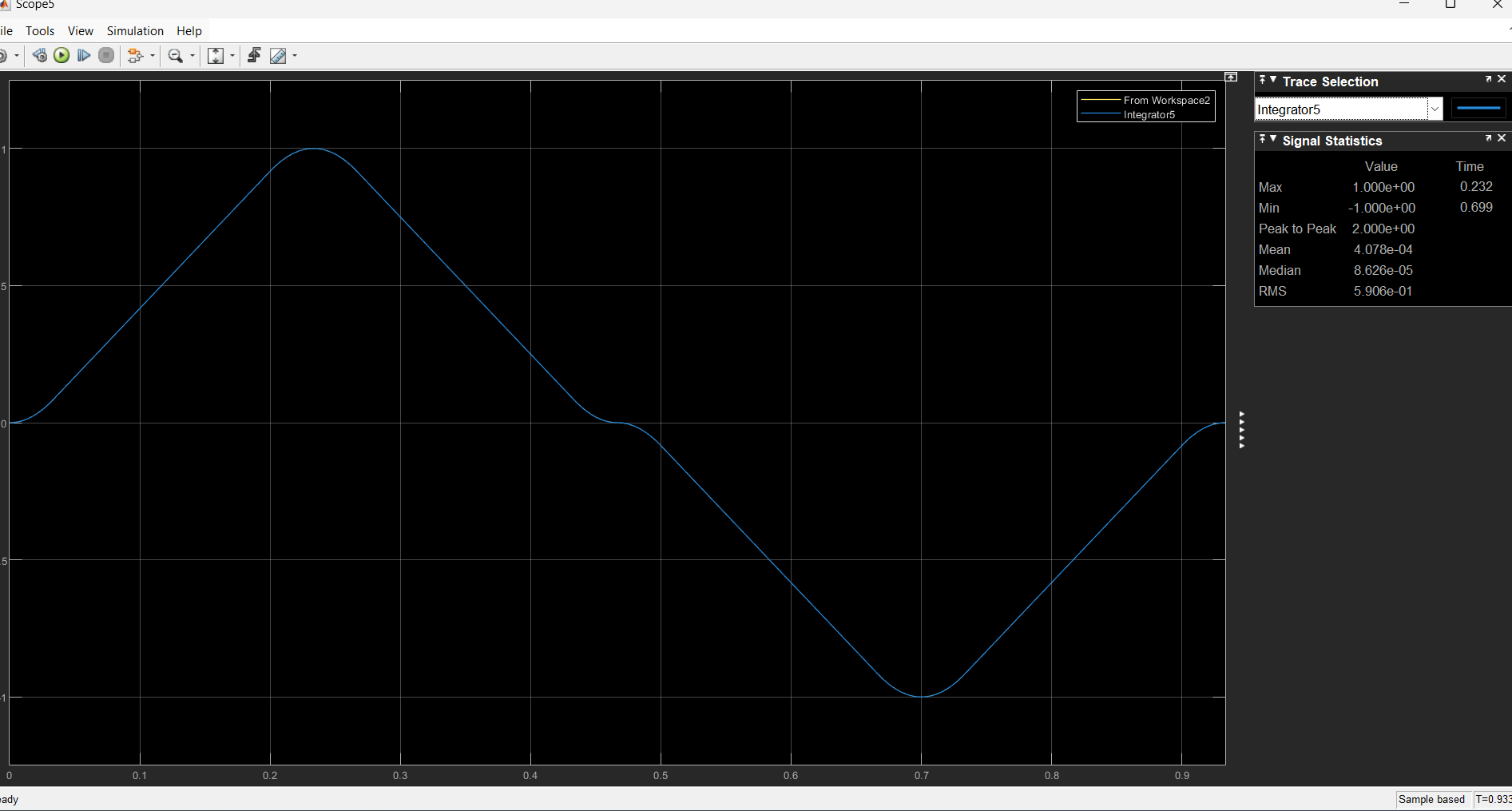


Amazing!!! –

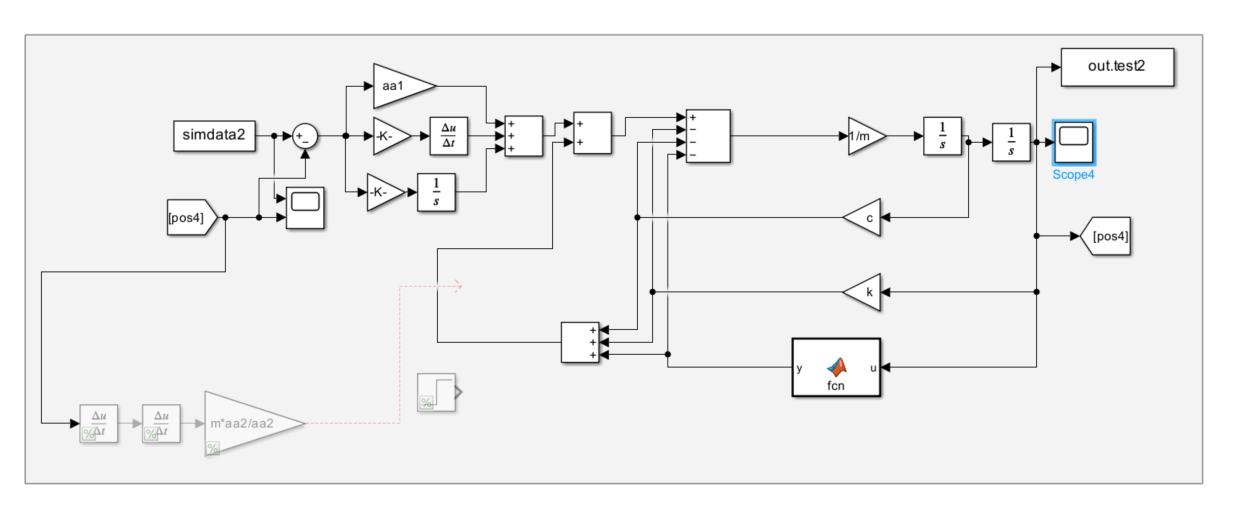


Eta=1e6; aa1=1e4; aa2=1; aa3=1e-12.

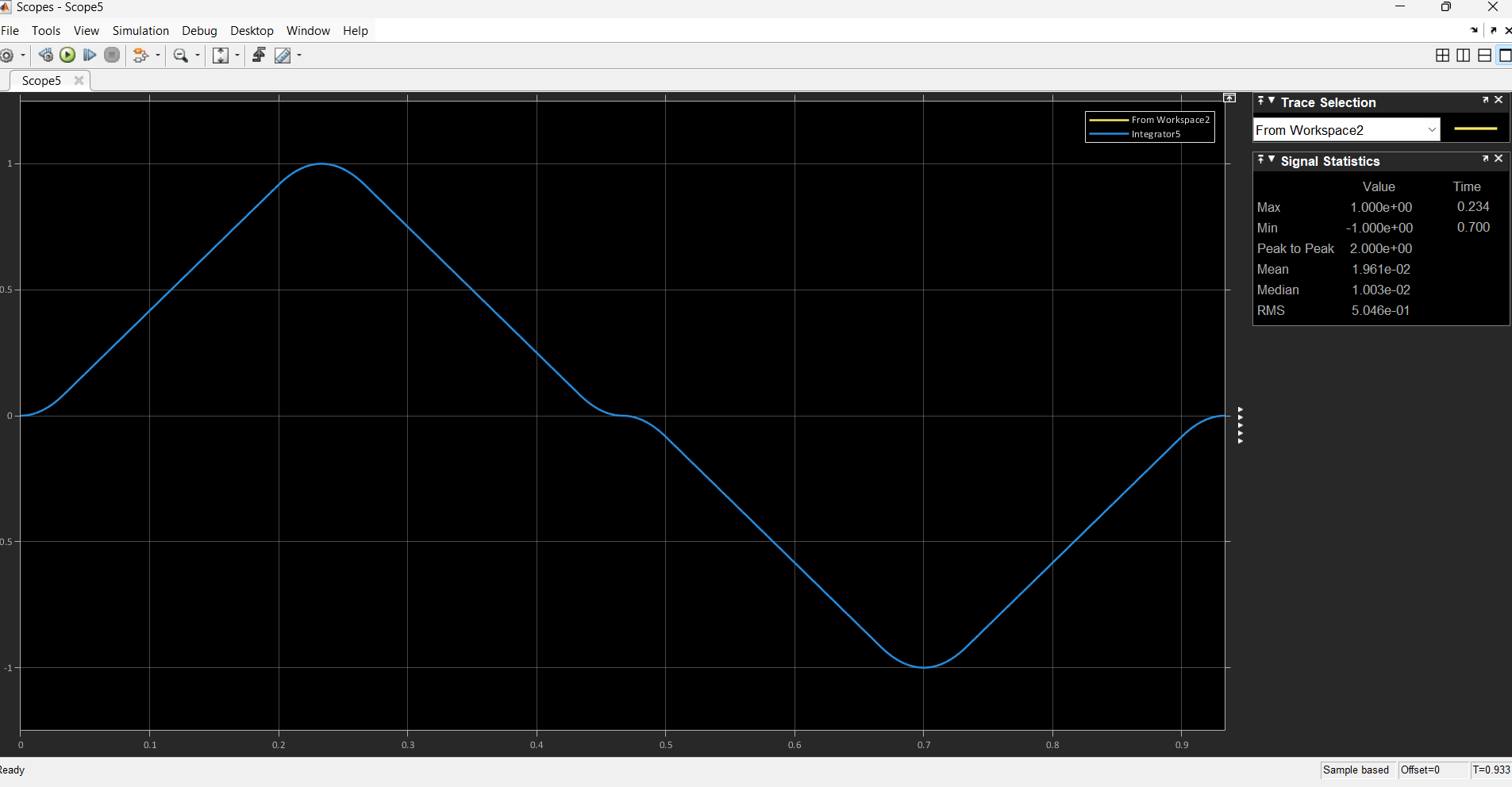
Bellow is with FeedBack Linearization:



Amazingly good

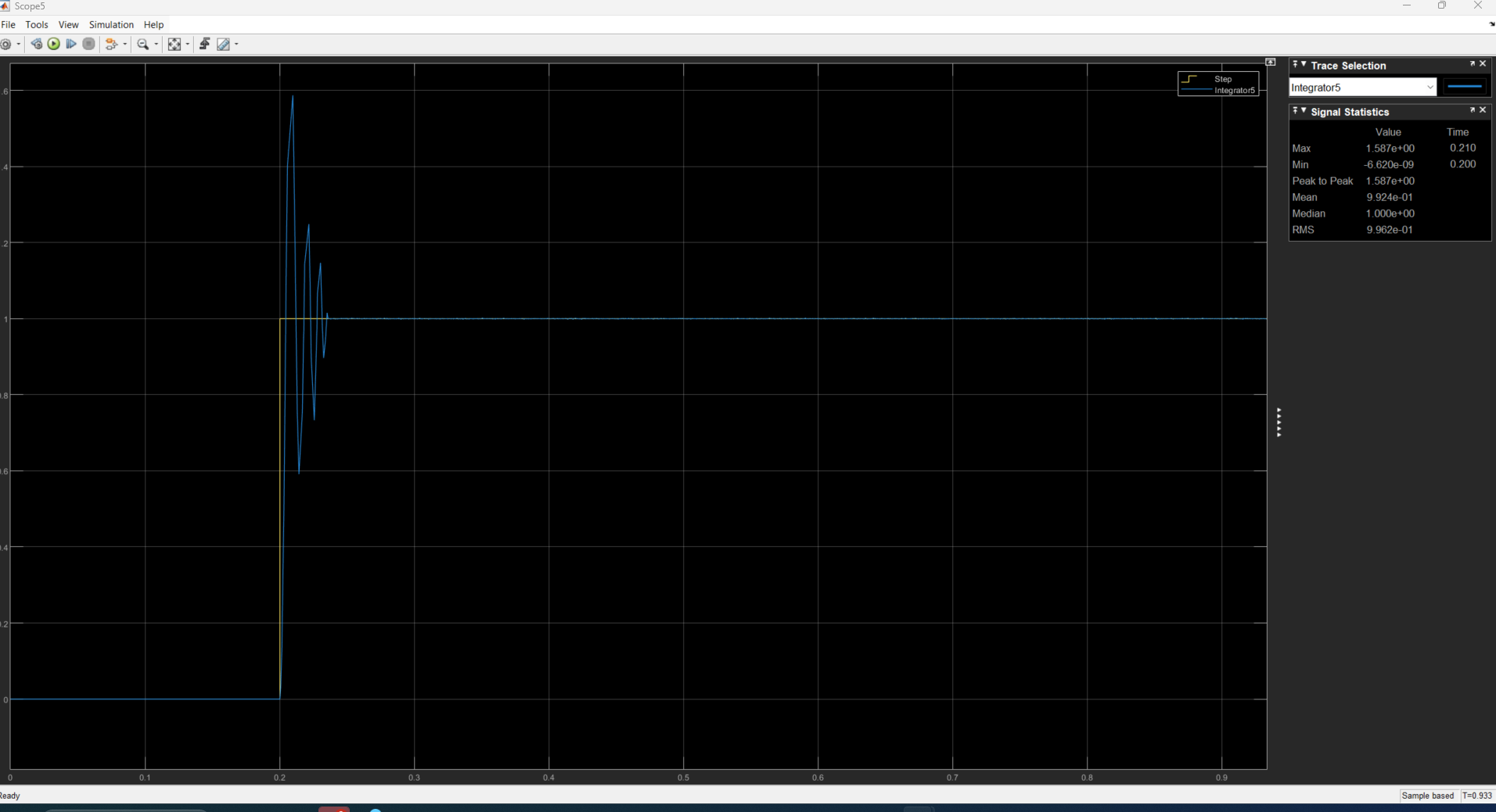


Pure SMC result is below:

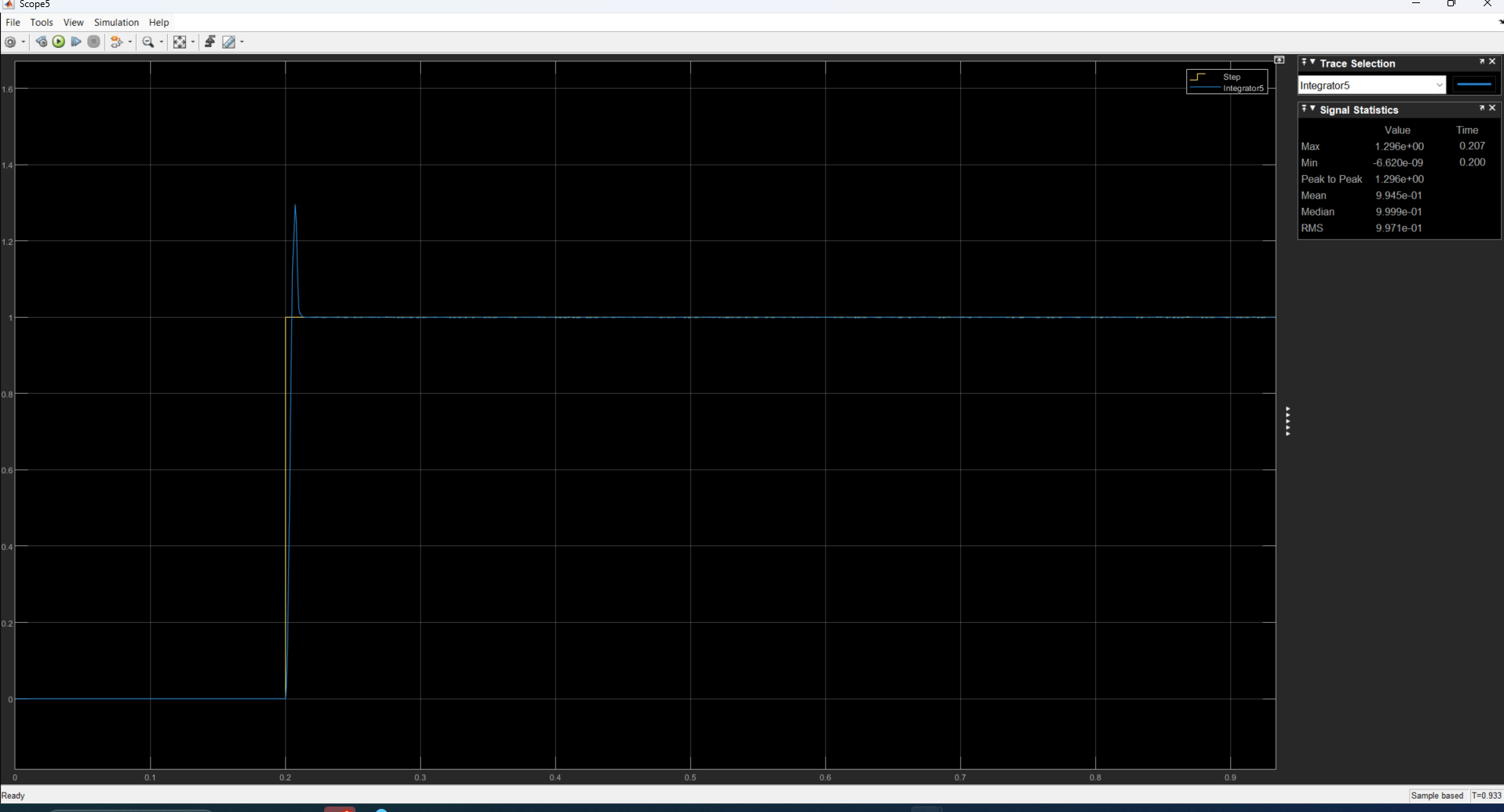


Step input:

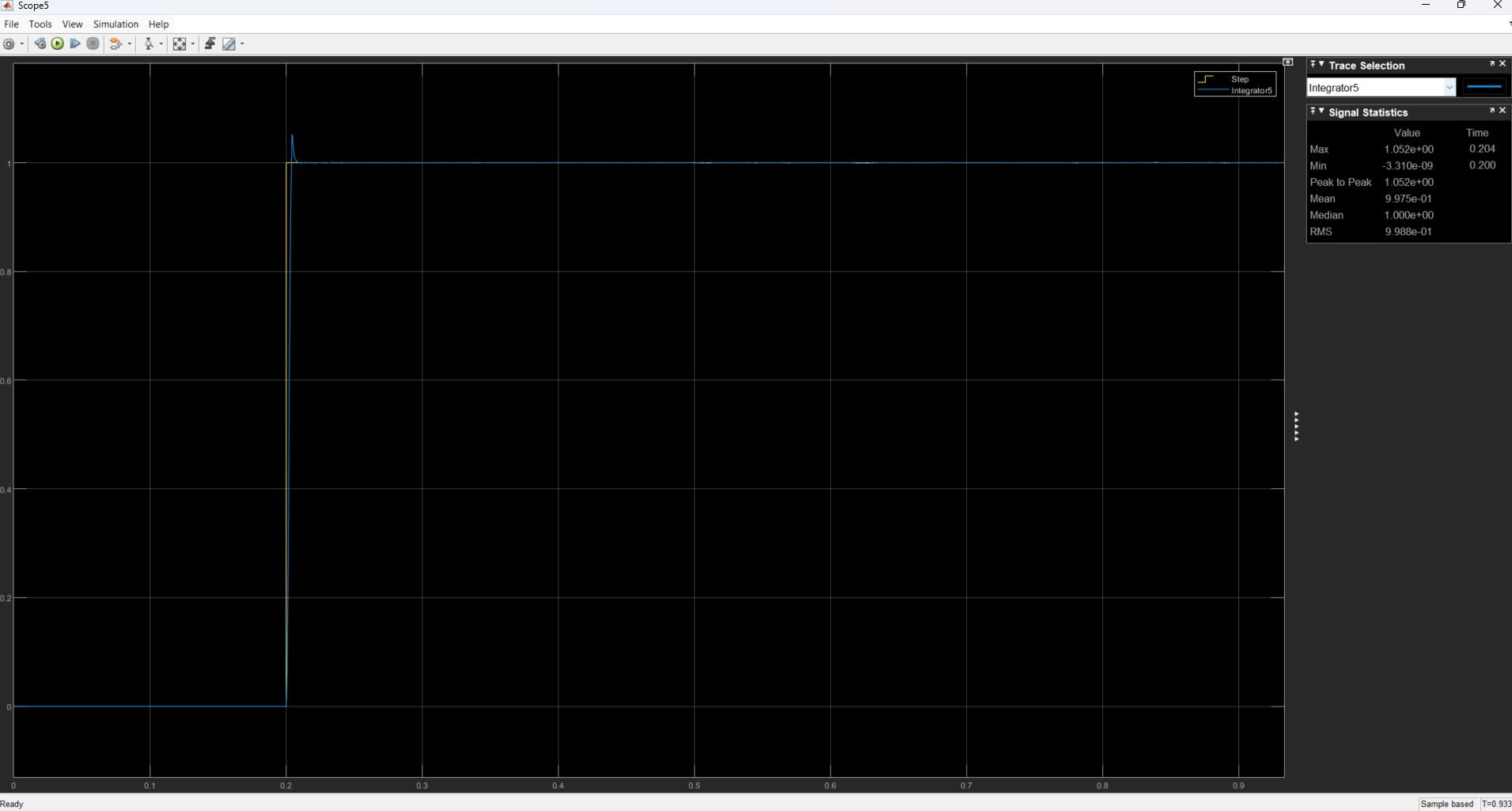
Below is: lambda = 5000 and eta = 1e5



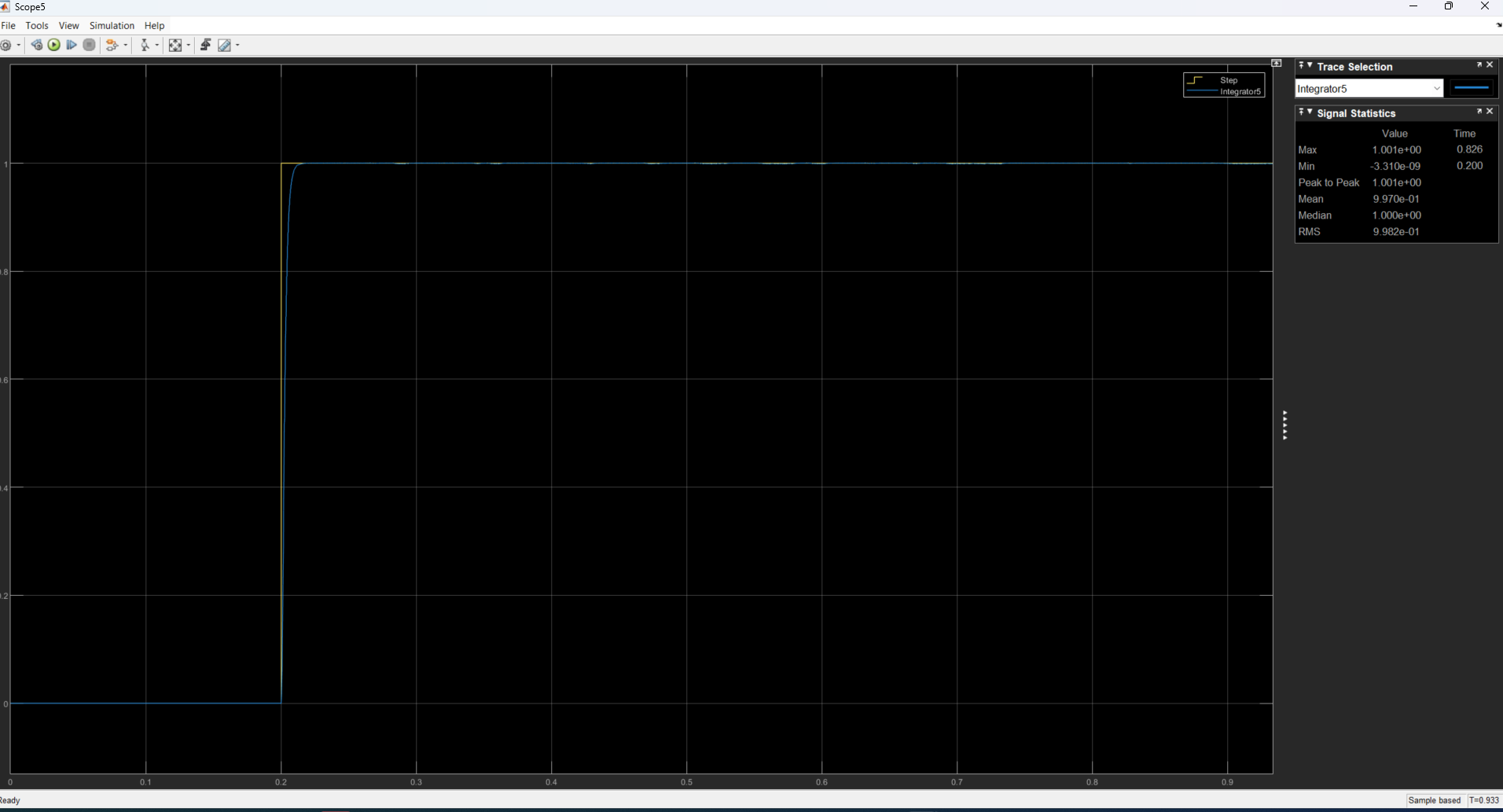
lambda = 1000 and eta = 1e5



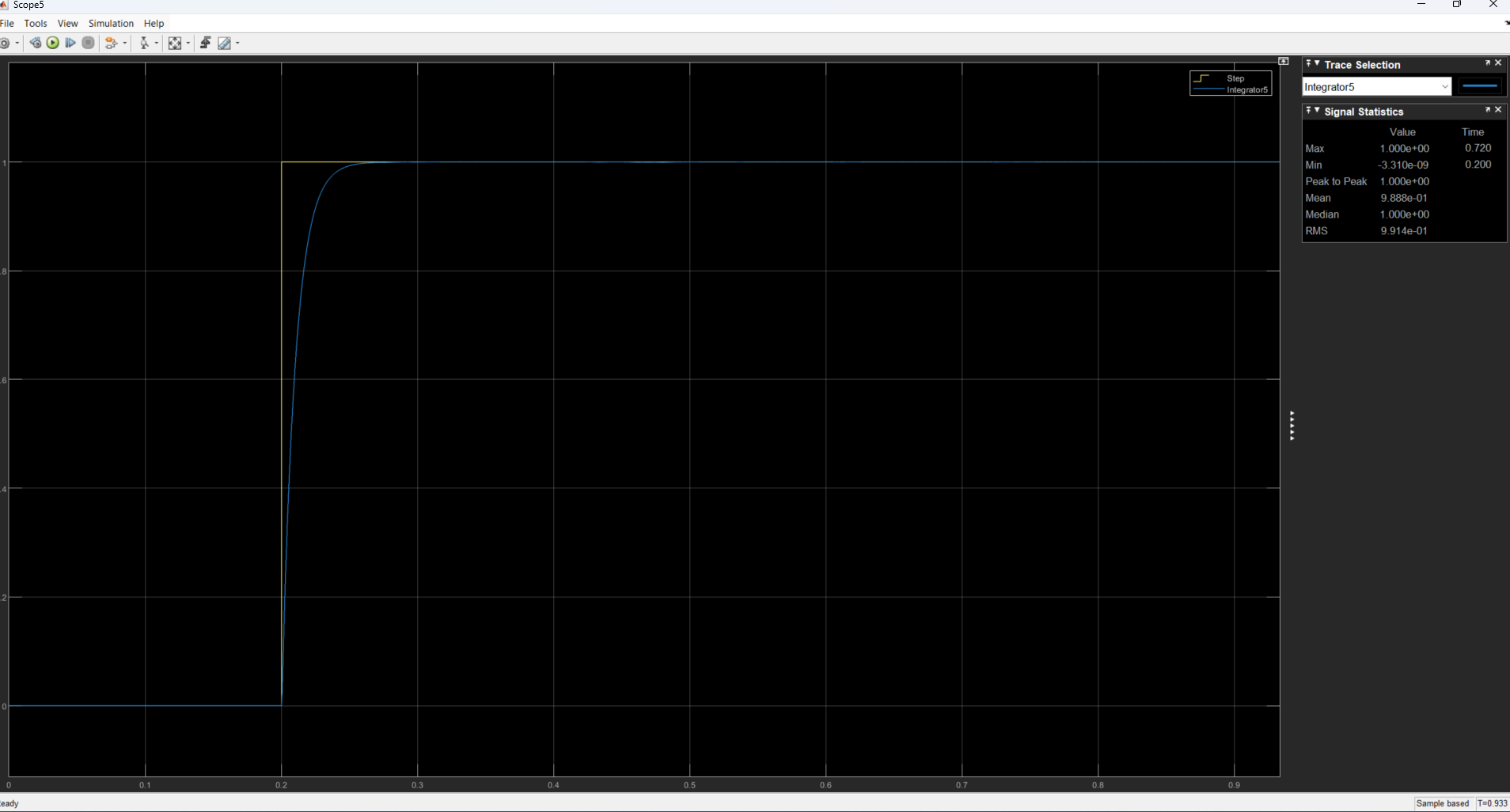
Eta=2e5, lambda=1e3



Lambda = 500; eta = 2e5



Lambda = 1e2; eta = 2e5



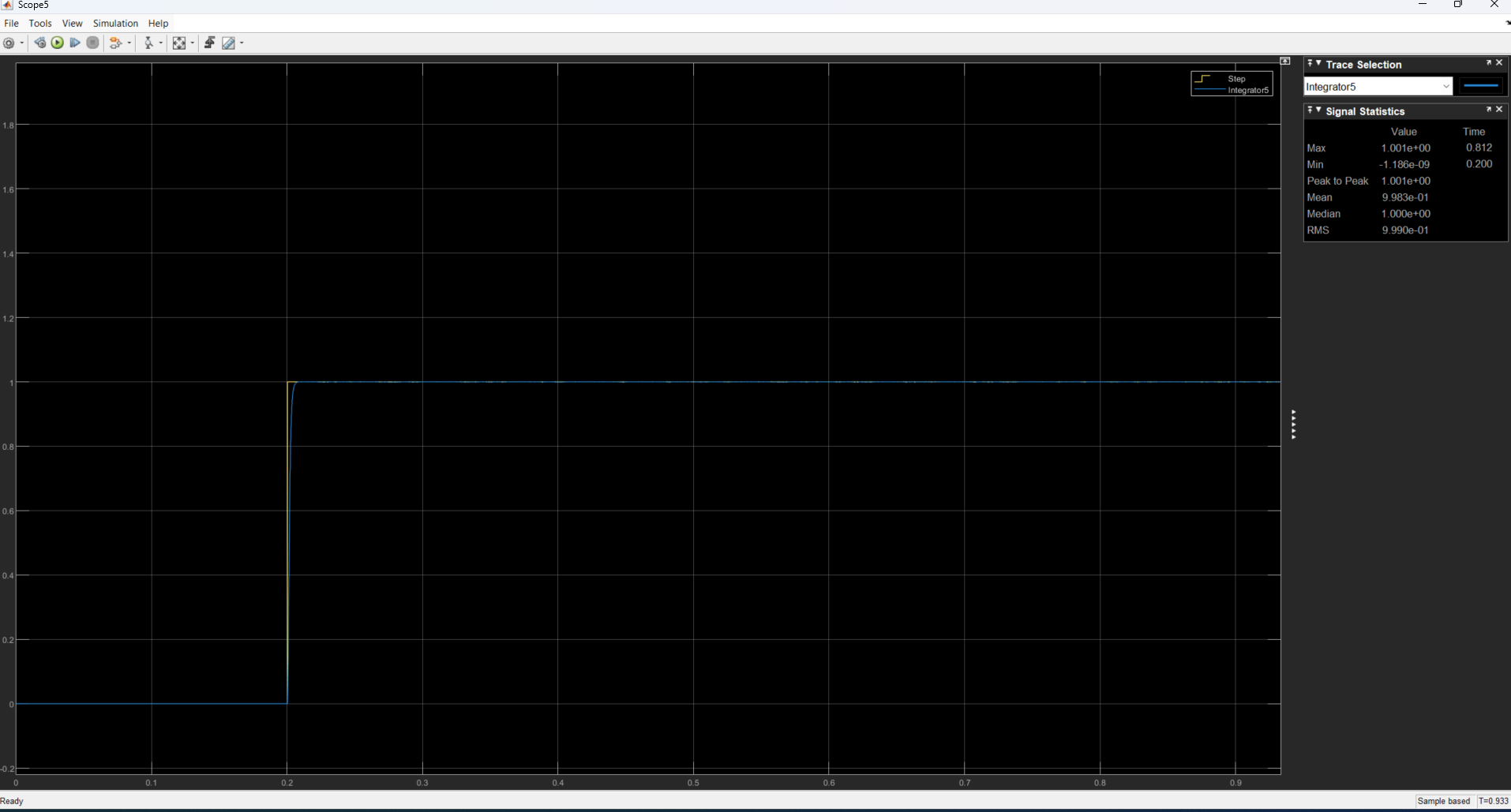
So; decreasing lambda leads to slightly slow system (rise time and peak times are both larger/greater) and with less maximum peak overshoot. But grater values of lambda leads to large maximum peak overshoot.

Below is with: lambda = 1e4; eta = 2e5



So increasing lambda leads also to oscillatory response. Also it shows that increasing the eta led to no maximum peak overshoots and all peaks are gone though small delay is resulted.

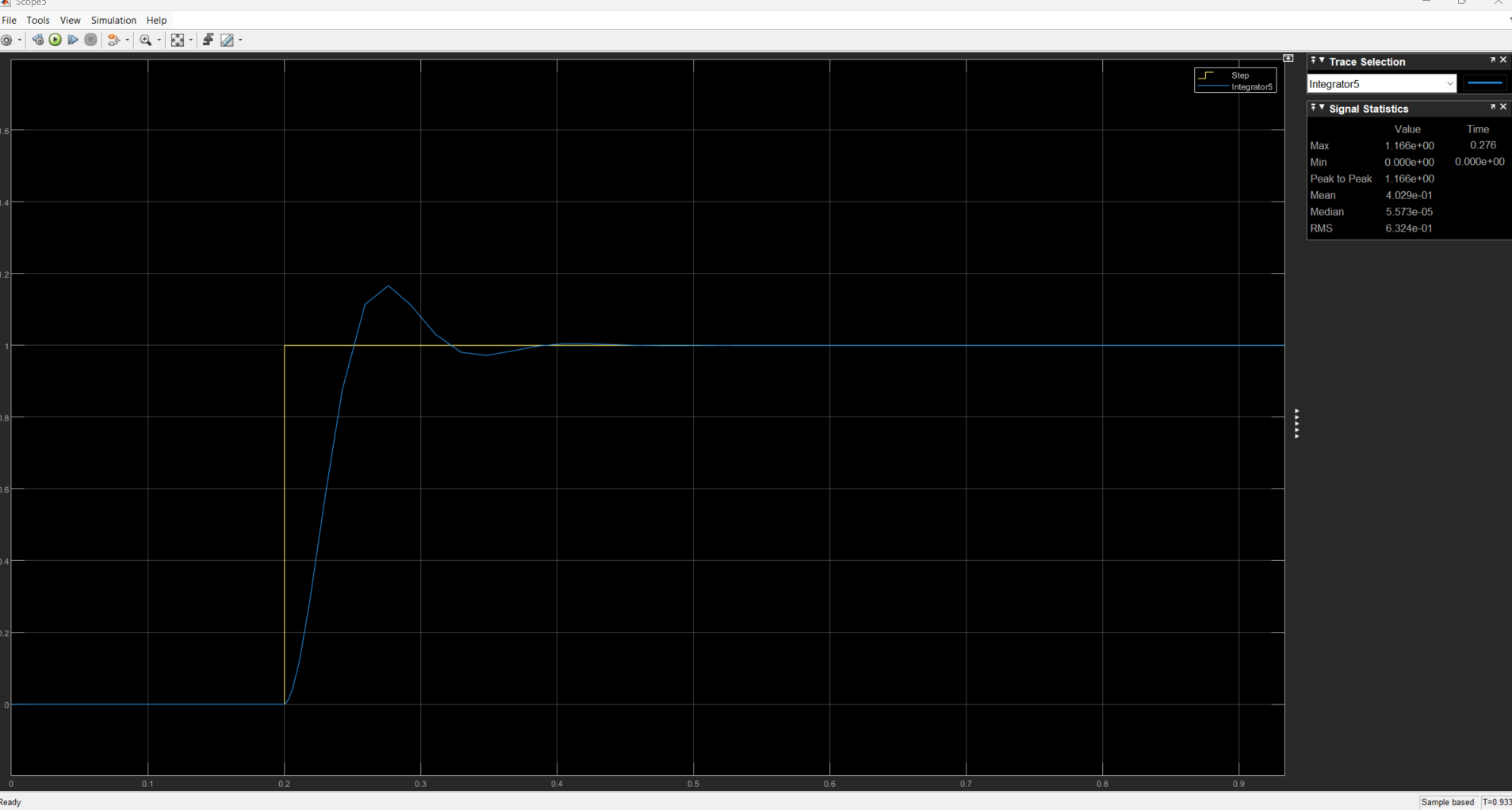
Below is: lambda = 1e3 and eta = 5e5



The above is pure SMC with step. Now FBLin with pure step:

Step for pure FBLin

Kp=2000, ki=5; kd=40



Now below is smc+FBLin for nonlinear mass spring damper

Kp=1e3, ki=1, kd=1

