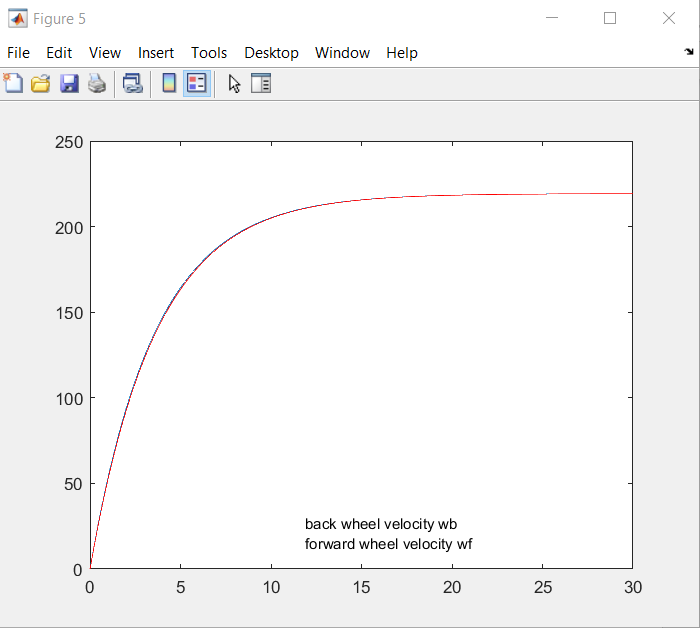
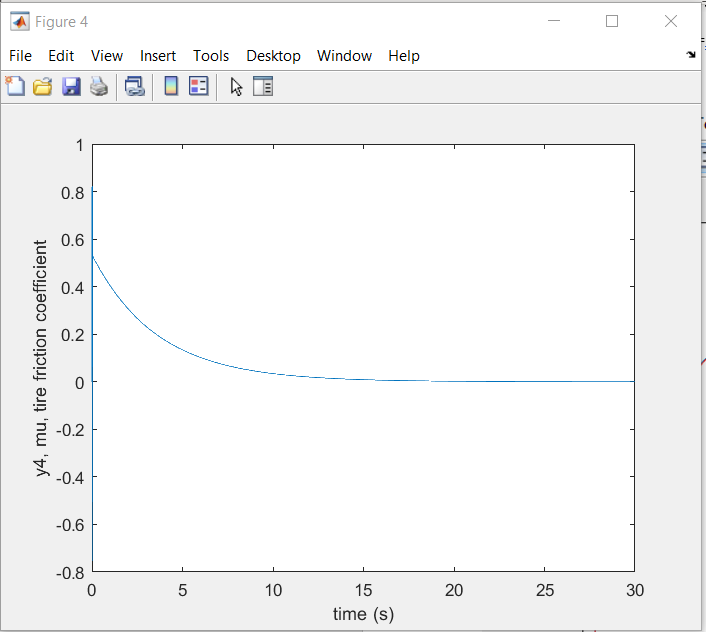
First we try different PID parameters in program simulation(not HIL)

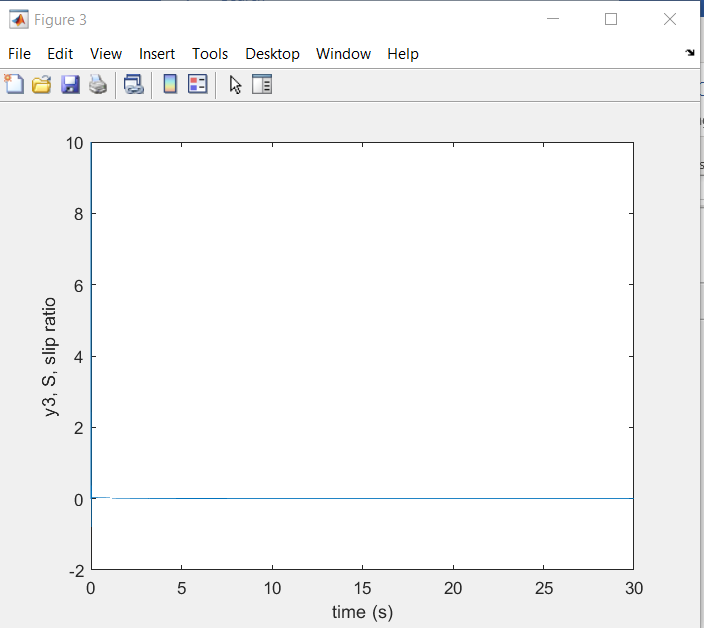
Without PID, the maximus mu value is 0.8

So we chose to control mu to 0.8 and changed the Kp value first

Run1: mu = 0.8, kp = 4, ki=kd=0;







backwheel =

struct with fields:

RiseTime: 8.0514

SettlingTime: 14.3320

SettlingMin: 197.2898

SettlingMax: 219.2073

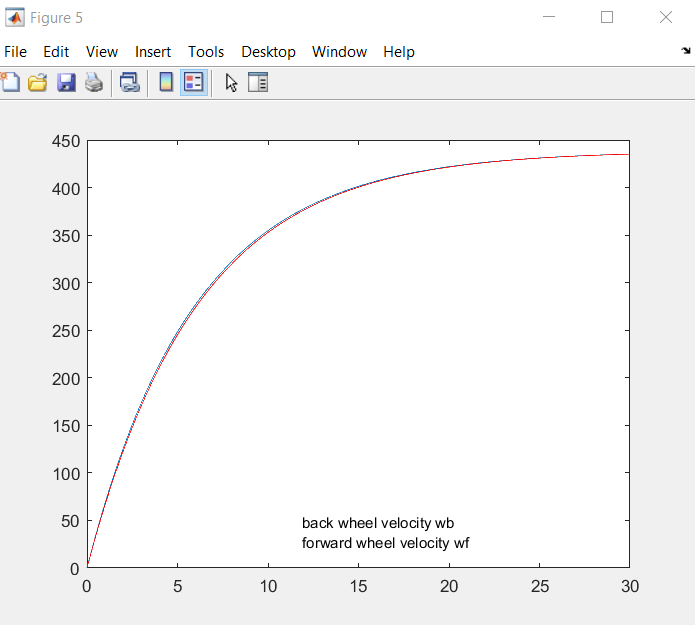
Overshoot: 0

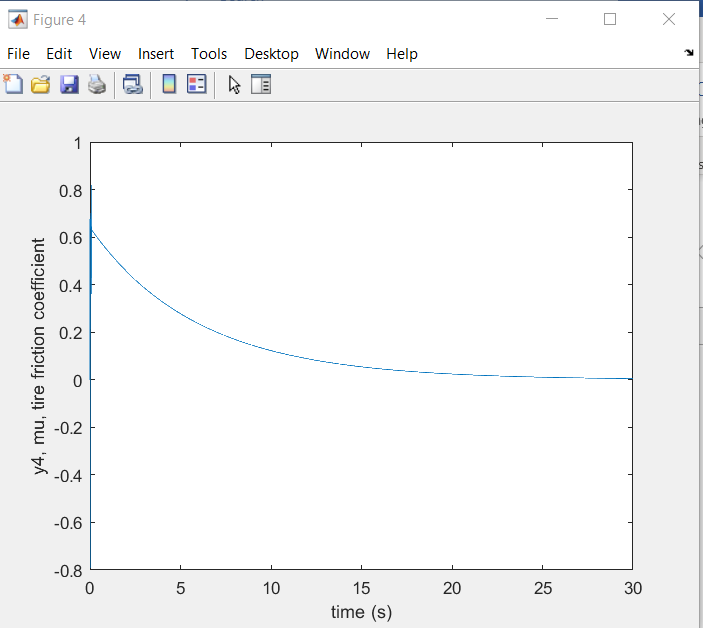
Undershoot: 0

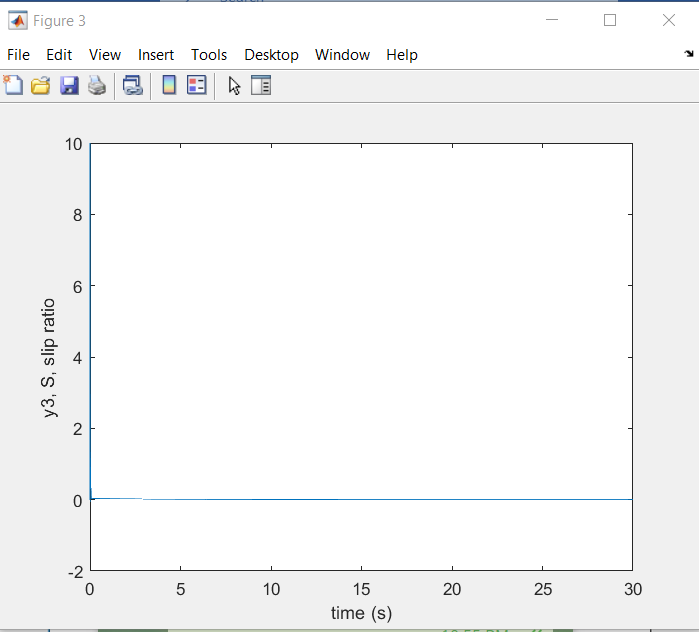
Peak: 219.2073

PeakTime: 29.9990

Run2: mu = 0.8, kp = 8, ki = kd = 0;







backwheel =

struct with fields:

RiseTime: 13.0932

SettlingTime: 22.1190

SettlingMin: 391.6416

SettlingMax: 435.1529

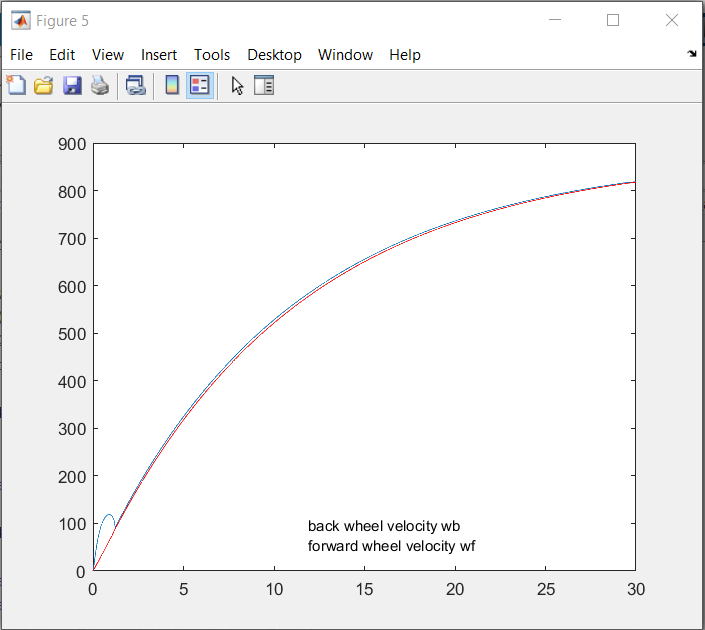
Overshoot: 0

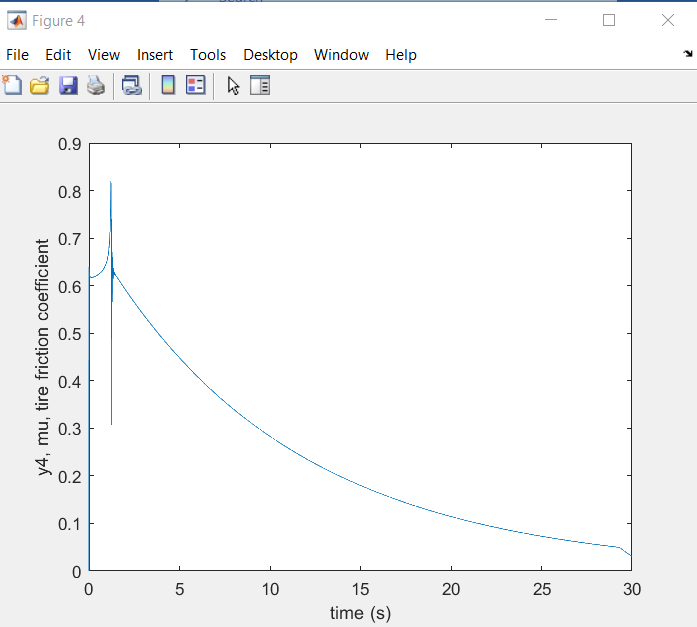
Undershoot: 0

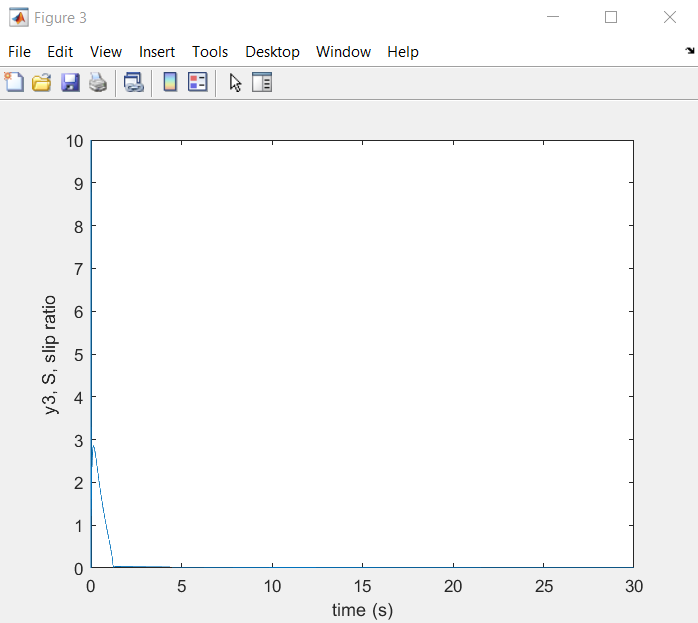
Peak: 435.1529

PeakTime: 29.9990

Run3 : mu = 0.8, kp = 16, ki=kd=0;







backwheel =

struct with fields:

RiseTime: 19.1124

SettlingTime: 27.1780

SettlingMin: 735.5554

SettlingMax: 817.2729

Overshoot: 0

Undershoot: 0

Peak: 817.2729

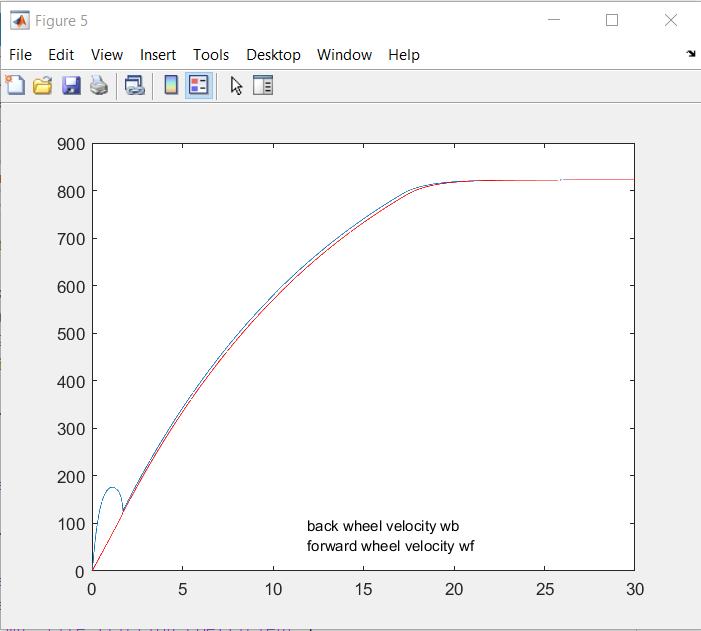
PeakTime: 29.9990

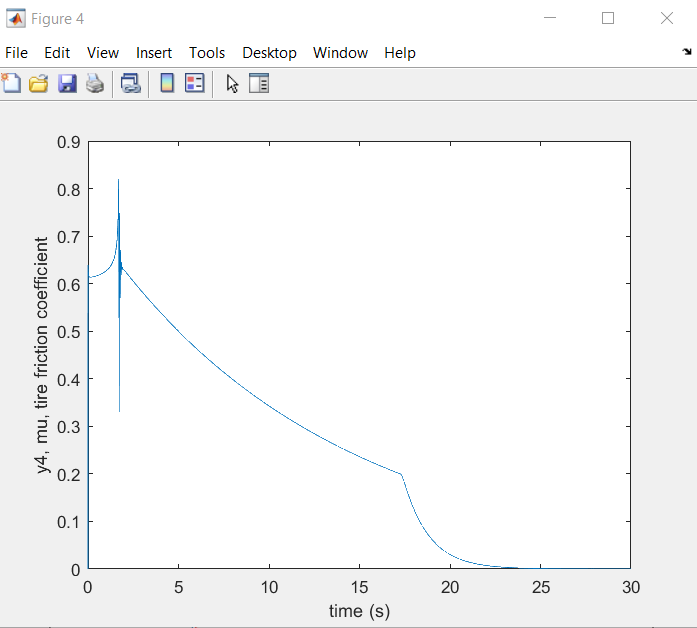
After 3 runs, we find that with mu=0.8 stable, not only can we maintain a good traction control(wheel velocity stay the same and slip ratio return to 0 after launching), but also, kp has some proportional correlation with the final wheel angular velocity: approximately, kp \* 500 = wb.

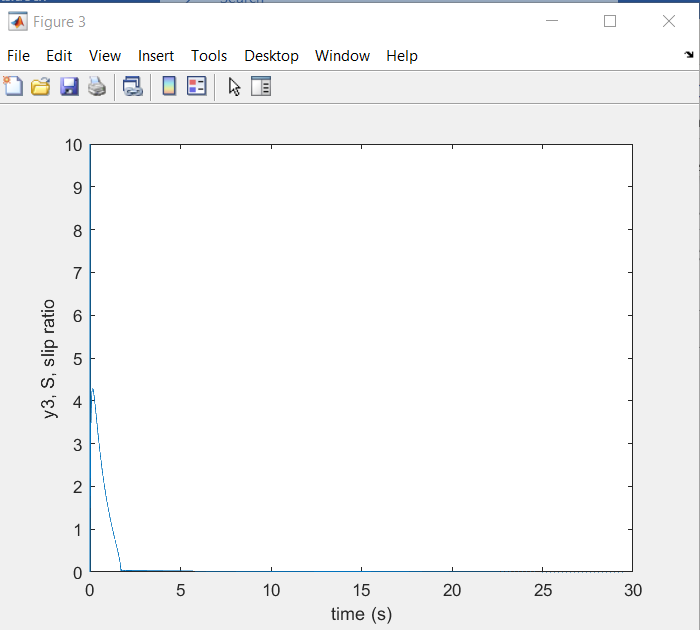
However, we should also notice that with mu = 0.8, the time response, take rise time as example, is 19.11s when accelerating to max speed 800rads/s while without PID, the rise time is 10 s .

Moreover, when we keep increasing kp, something more interesting happened.

Run4: mu = 0.8, kp = 20, ki=kd=0;







backwheel =

struct with fields:

RiseTime: 14.1081

SettlingTime: 18.2865

SettlingMin: 740.0508

SettlingMax: 822.2628

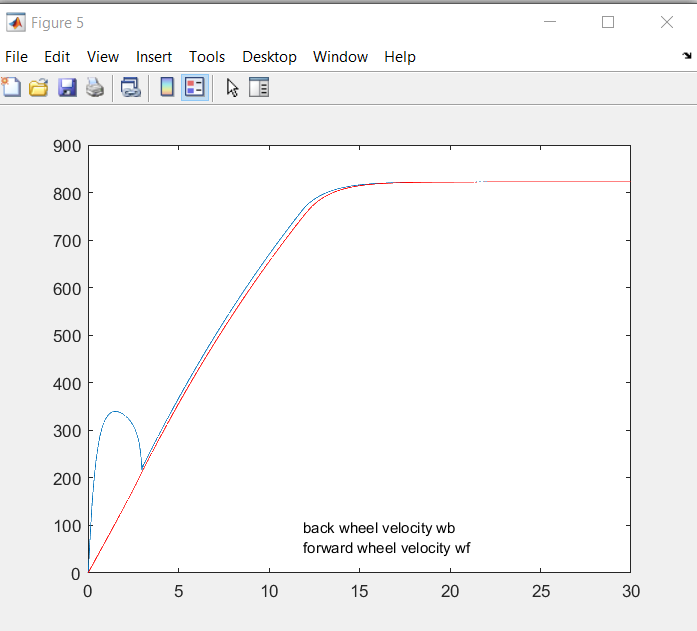
Overshoot: 0

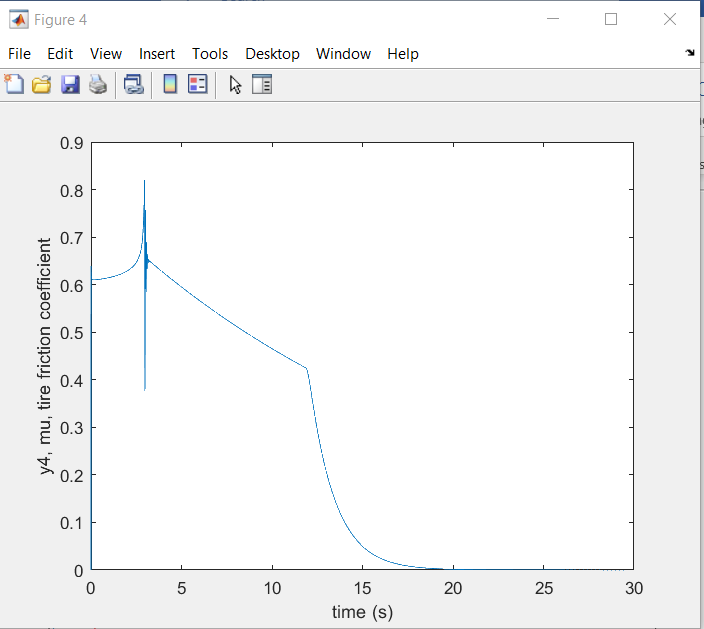
Undershoot: 0

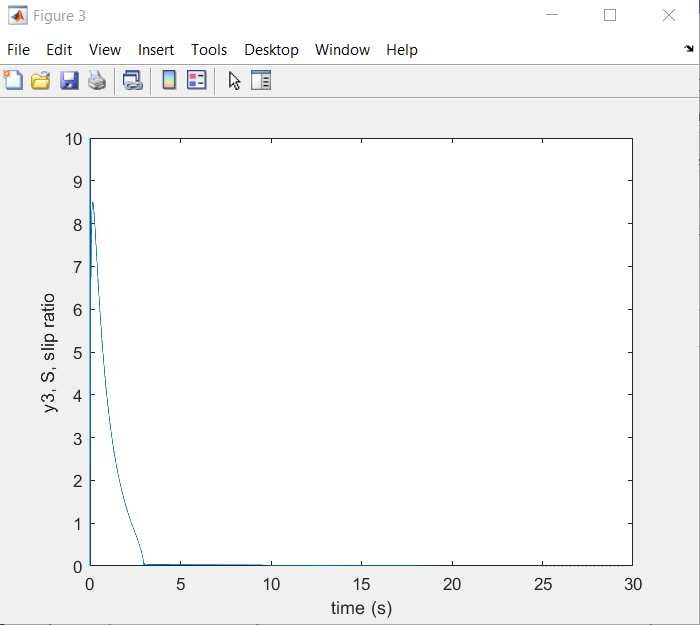
Peak: 822.2628

PeakTime: 29.9990

Run 5 : mu = 0.8, kp =32, ki=kd = 0;







backwheel =

struct with fields:

RiseTime: 10.4979

SettlingTime: 13.9797

SettlingMin: 740.0705

SettlingMax: 822.2667

Overshoot: 0

Undershoot: 0

Peak: 822.2667

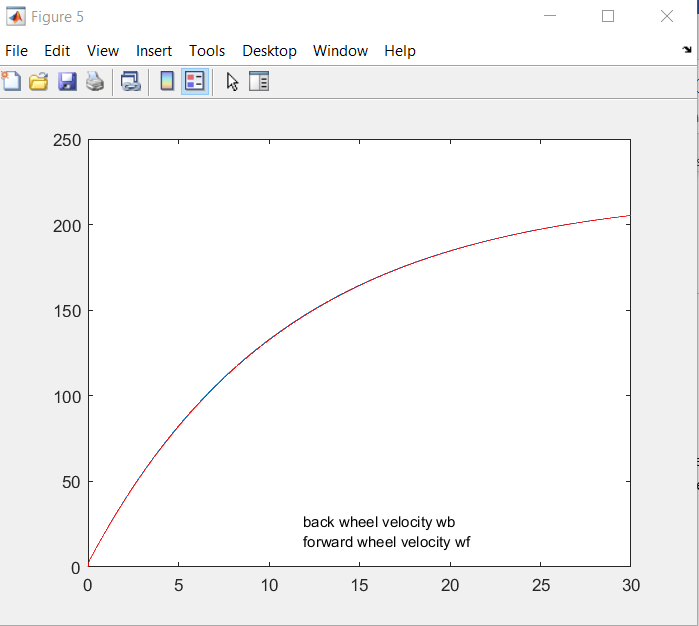
PeakTime: 29.9770

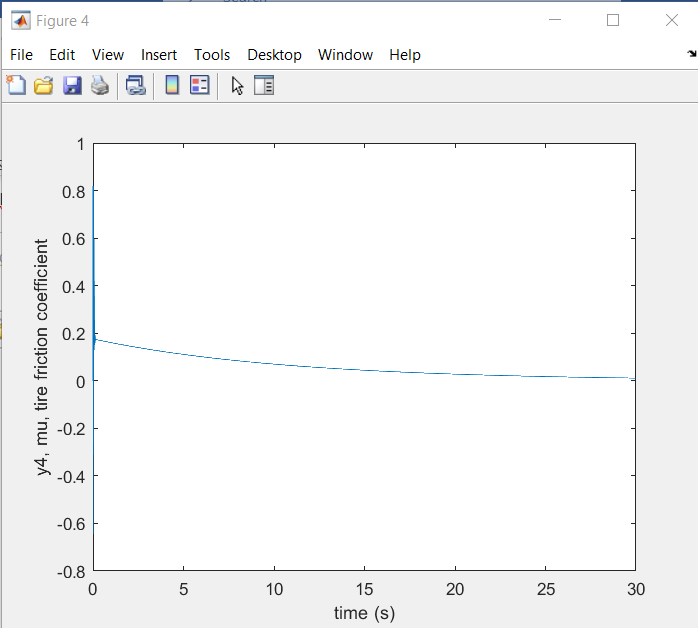
From run4 and run5 we can see as kp > 16, although time response is better(rise time shortened), the traction control became worse, as the difference between wheel velocity increases, which can also be observed in slip ratio(it takes longer to return to 0)

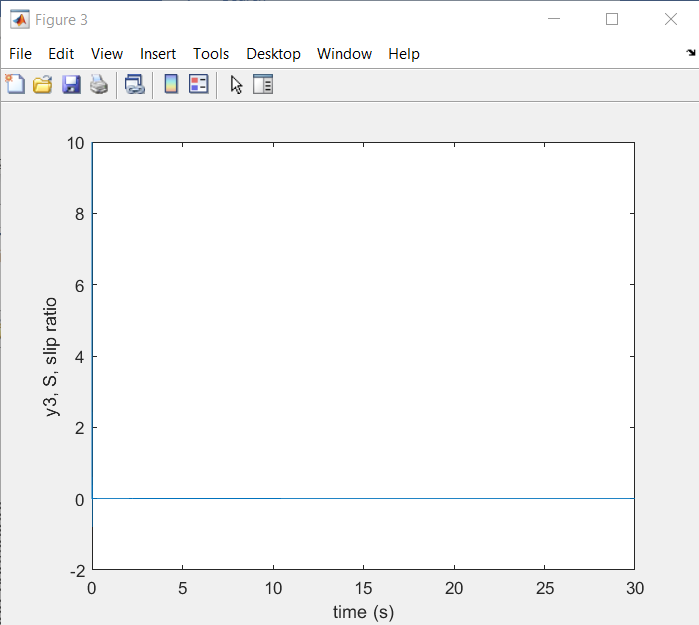
Therefore, we pick kp = 16 to 20 for launch and speed control.

Additionally, we also try to find the relation if we keep kp unchanged and change mu instead.

Run6: kp = 16, mu = 0.2, ki=kd=0;







backwheel =

struct with fields:

RiseTime: 19.1347

SettlingTime: 27.1749

SettlingMin: 184.7800

SettlingMax: 205.3084

Overshoot: 0

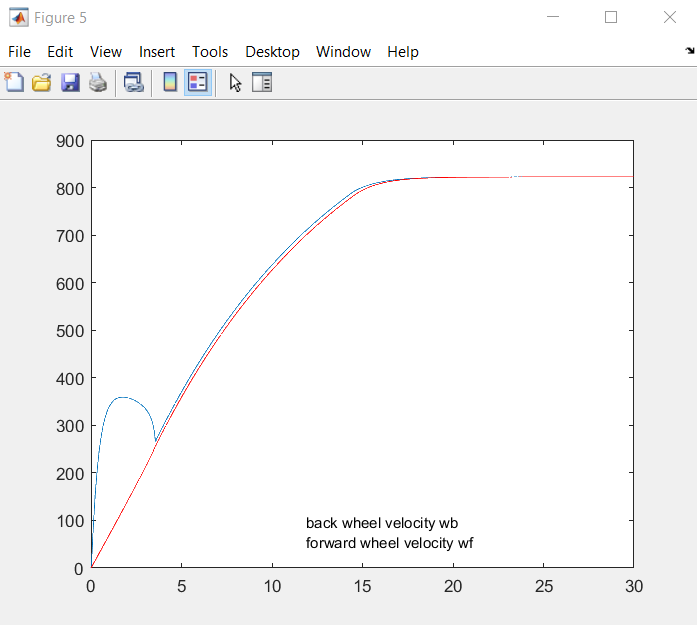
Undershoot: 0

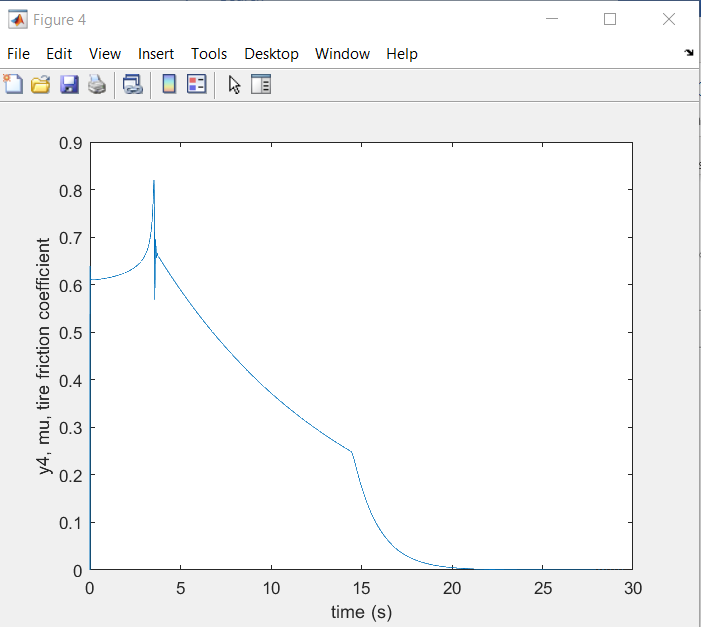
Peak: 205.3084

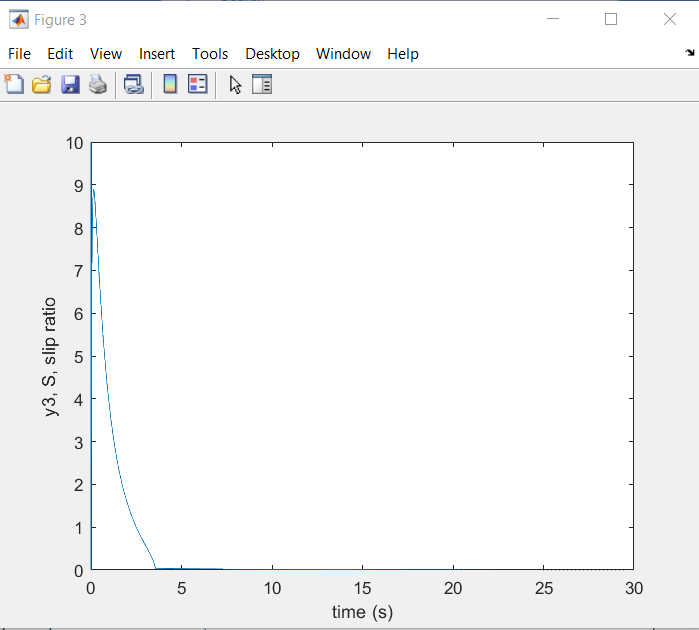
PeakTime: 29.9990

We find that if kp = 16, mu also has some proportional correlation with final wheel speed, approximately, mu = wf / 1000. Therefore in Run6, when mu = 0.2, the final speed is 200rads/s. And as mu increases to 0.8, the final speed increases accordingly to 800 rads/s. Similarly, if mu exceed 0.8, the traction control will be worse while the time response will be better, such as when mu = 1.0:

Run 7: mu = 1.0 , kp = 16, ki=kd=0;







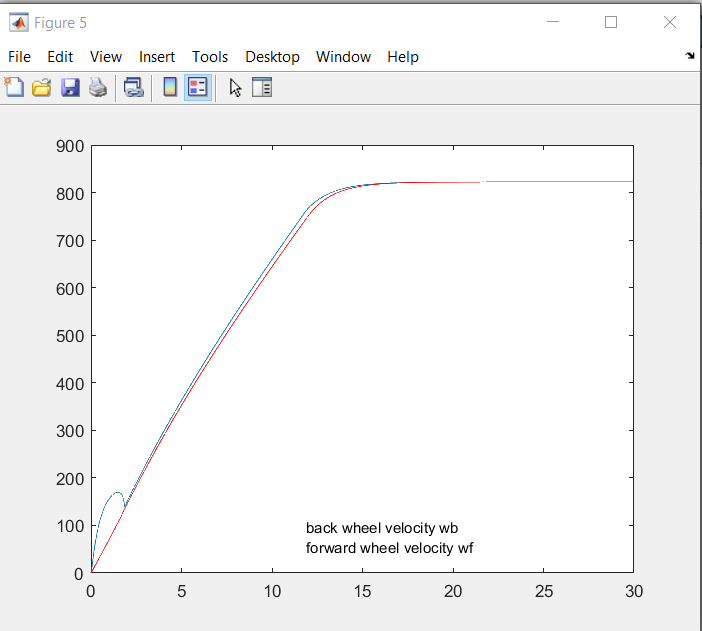
Not surprisingly, the slip ratio is not well-controlled, that’s why the traction control is not so good, although launch control gets better as rise time shortened. And for speed control we find that keeping mu and kp unchanged would be enough for maintaining certain speed.

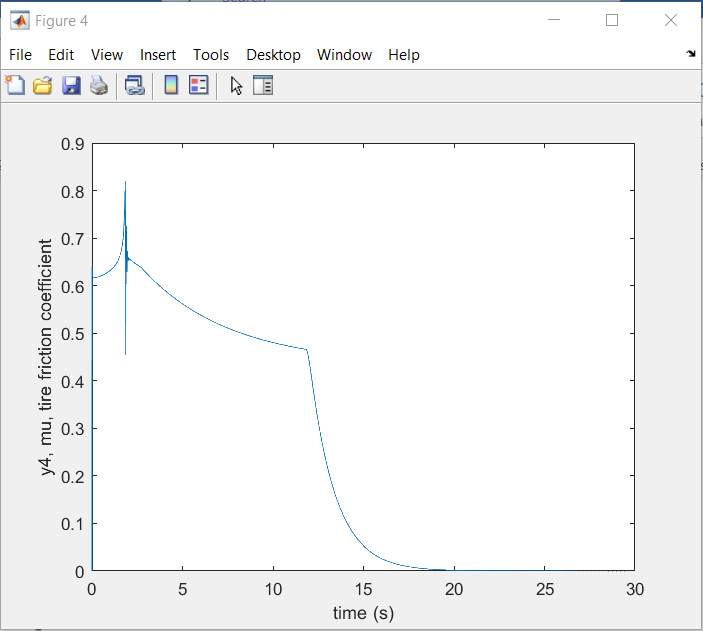
Considering the proportional effect of mu and kp. We decided to choose kp = 20 and mu = 0.8 for launch control, according speed control and traction control. And kp = 16 and mu = 0 to 0.8 to normal speed control along with speed and traction control.

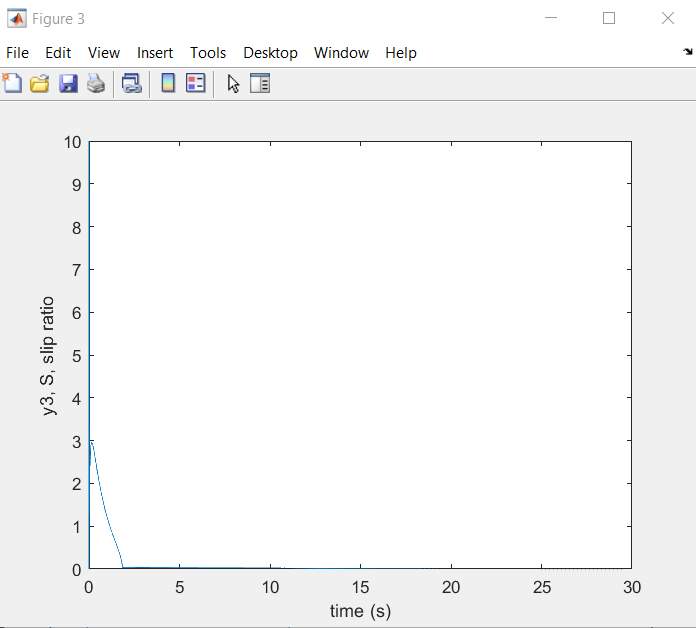
For picking suitable ki, we select from ki = 0 to 14, and we observed that ki has a significant influence on slip ratio/mu control. When ki increases(from 0), the slip ratio will be harder to control. Meanwhile, changing ki will not change the final speed.

Finally, we think ki = 0 to 4 is an acceptable range. The whole process could be shown below in Run8, Run9, Run10.

Run 8 : mu = 0.8, kp = 16, ki=2,kd=0







backwheel =

struct with fields:

RiseTime: 10.6259

SettlingTime: 14.0660

SettlingMin: 740.0491

SettlingMax: 822.2667

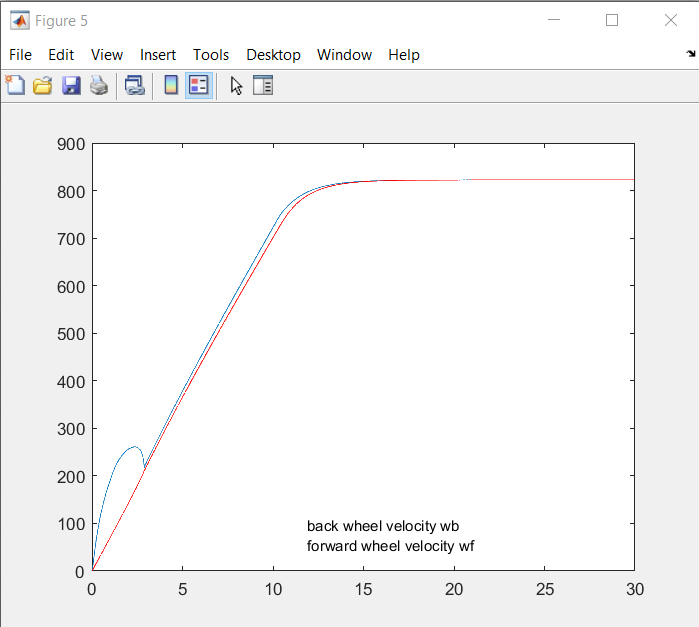
Overshoot: 0

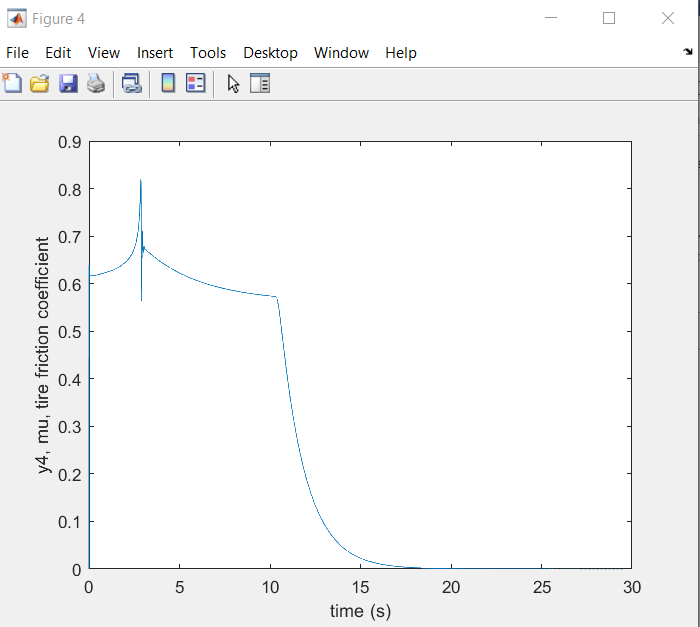
Undershoot: 0

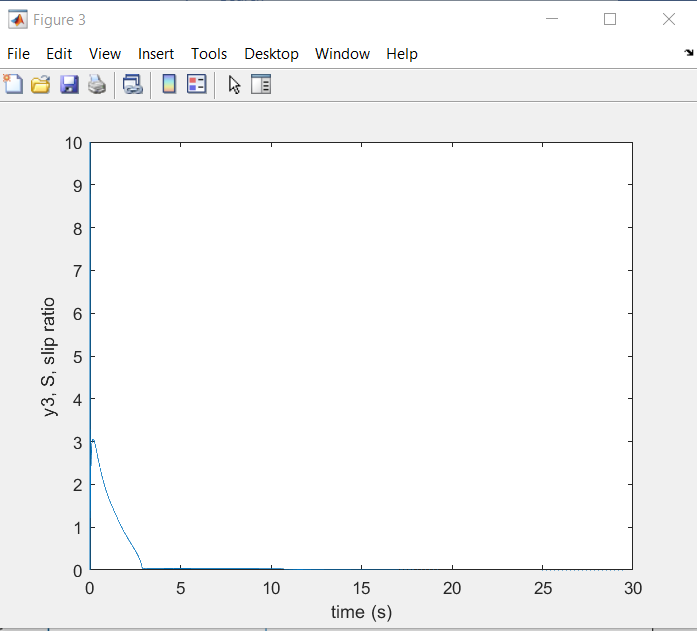
Peak: 822.2667

PeakTime: 29.9940

Run8: mu= 0.8,kp=16,ki=4,kd=0;







backwheel =

struct with fields:

RiseTime: 9.4383

SettlingTime: 12.8728

SettlingMin: 740.0735

SettlingMax: 822.2668

Overshoot: 0

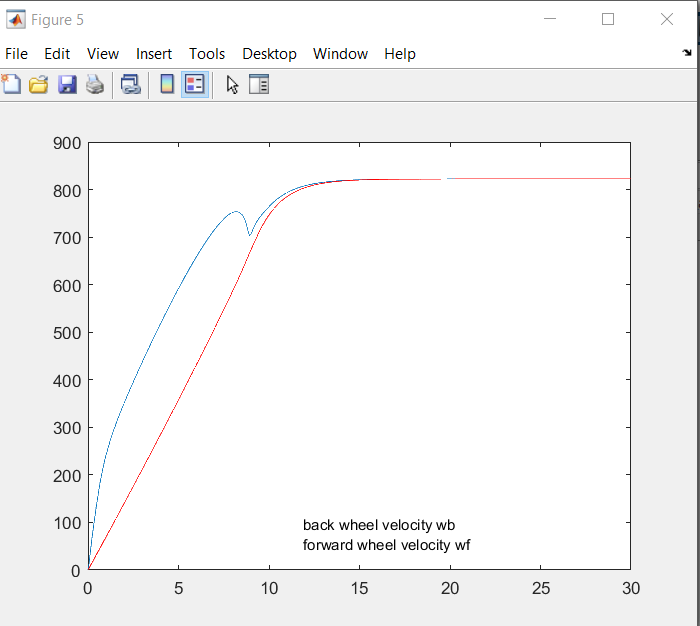
Undershoot: 0

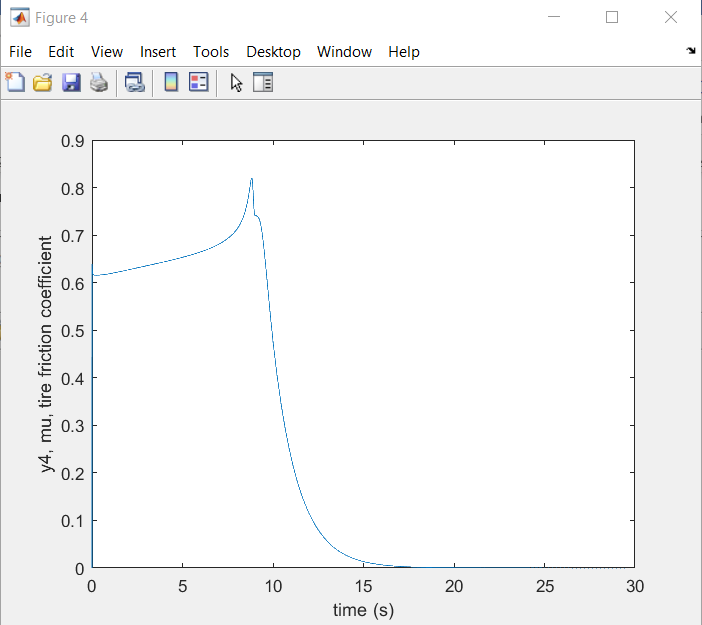
Peak: 822.2668

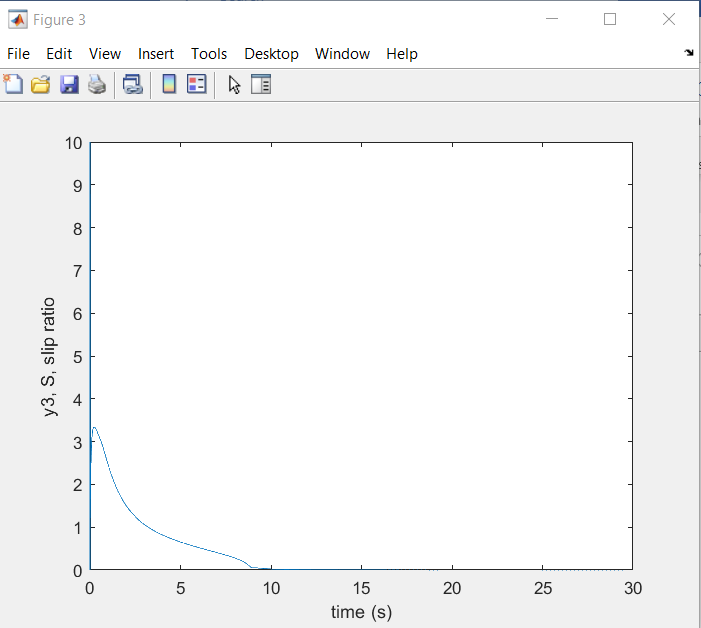
PeakTime: 29.8680

>> plot\_data

Run9: mu= 0.8, kp = 16, ki=8,kd = 0;







backwheel =

struct with fields:

RiseTime: 8.7067

SettlingTime: 12.1451

SettlingMin: 740.0629

SettlingMax: 822.2668

Overshoot: 0

Undershoot: 0

Peak: 822.2668

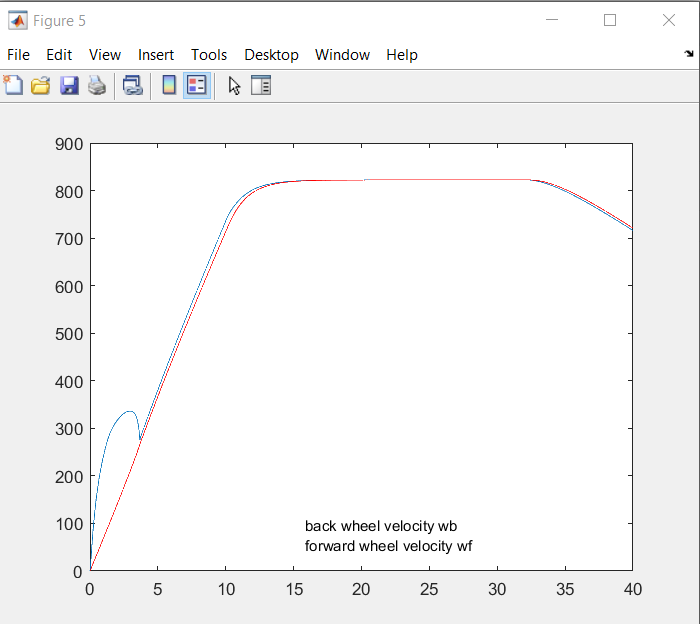
PeakTime: 29.8860

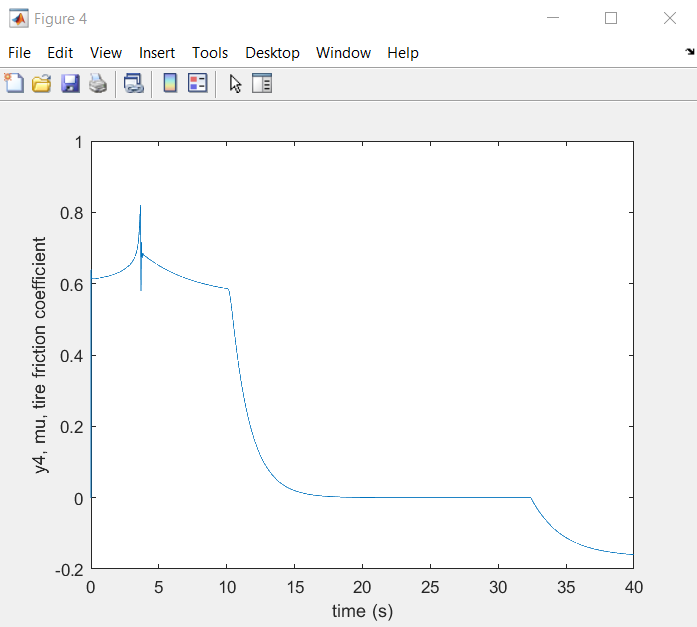
For picking kd, we find out that no matter what kd we chose, it will always break traction control(as if there’s no PID control and slip effect is not controlled at all). So we chose kd = 0.

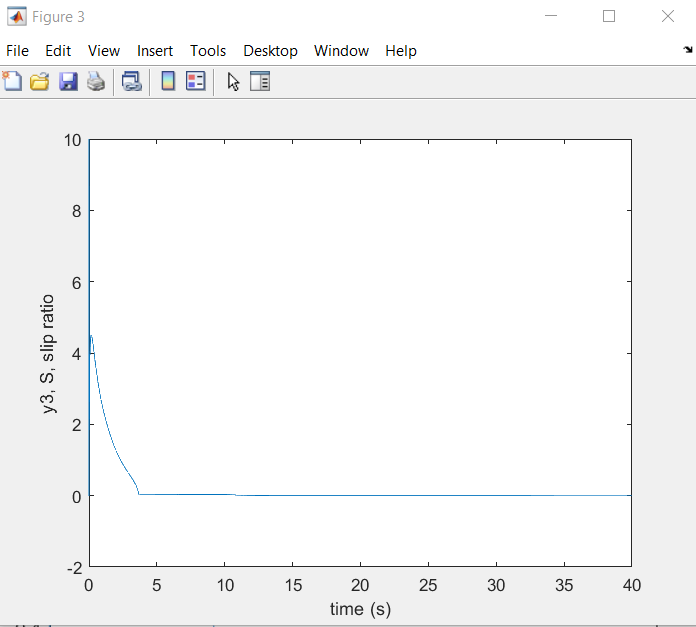
Now we consider brake control. Knowing that controlling mu will somehow affect the final speed, and kp can not be a negative value, we thought maybe we can change mu to a negative value and see if there would be a brake effect.

So we keep kp = 20, ki = 4, kd = 0 , run the car to max speed with mu = 0.8 first and then change mu from -0.2 to -1.2 to see the braking effect.

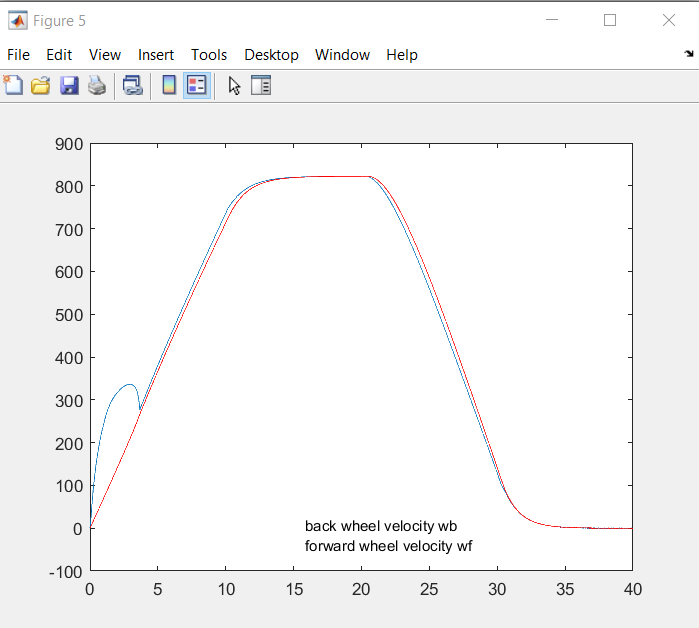
Run 10 mu = -0.2

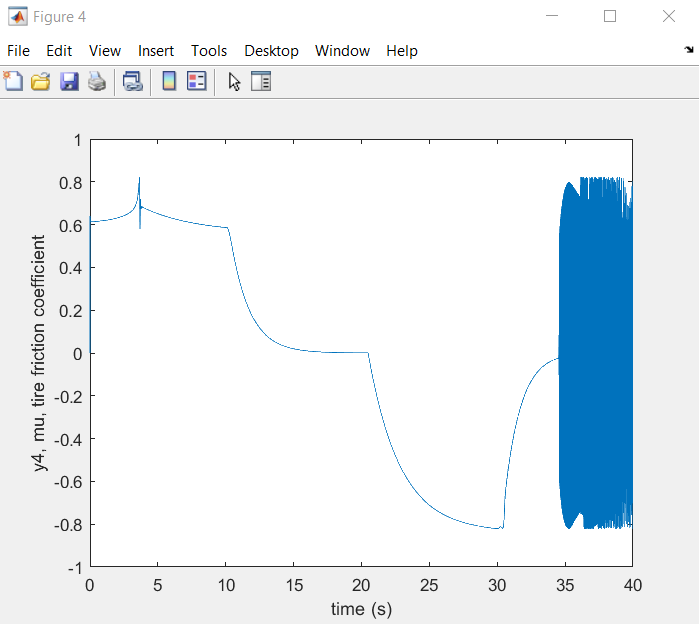


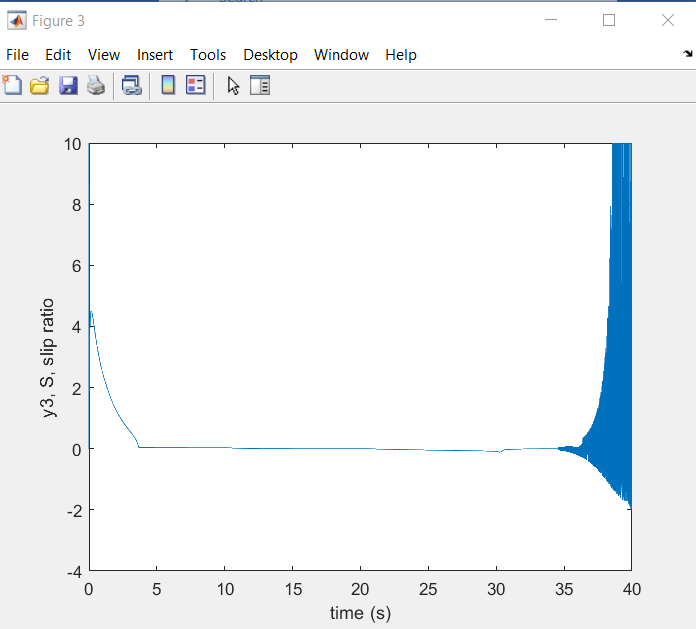




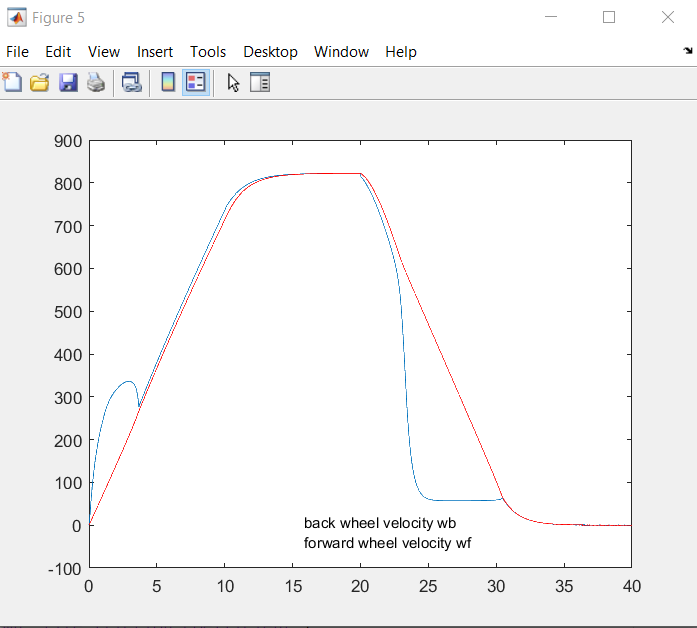
Run11: mu = -1.0

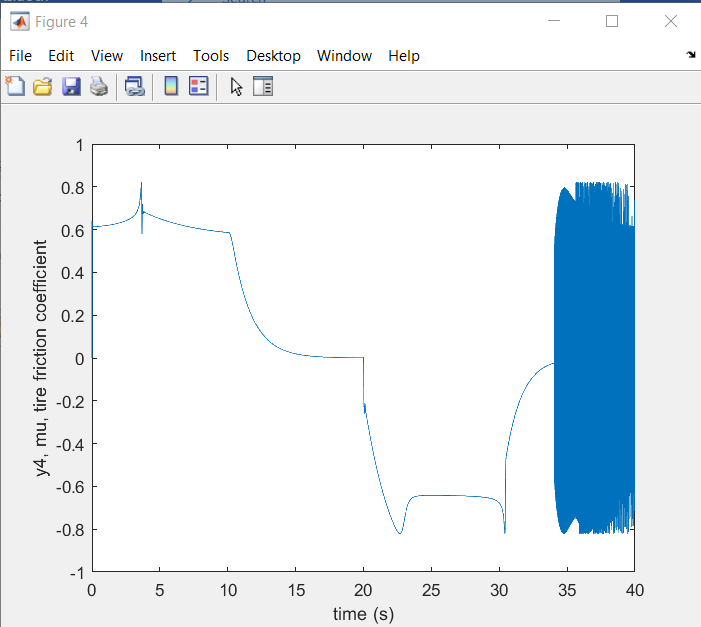


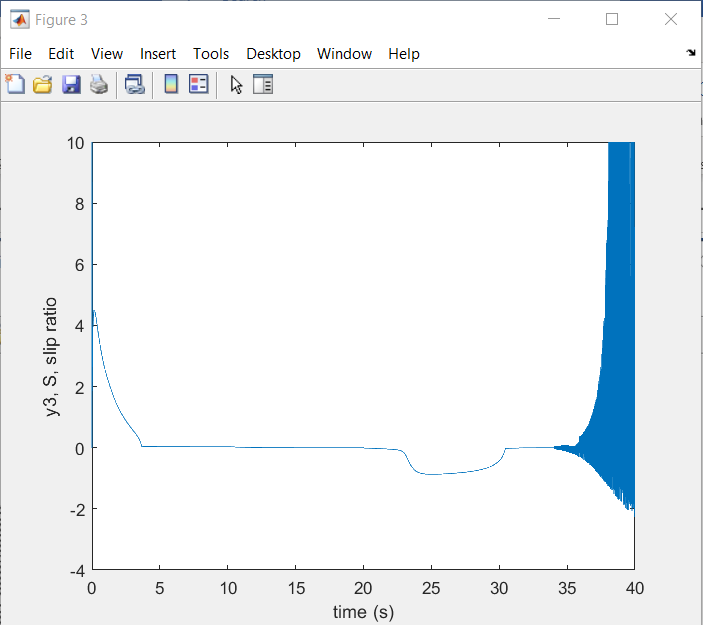




Run12: mu = -1.4

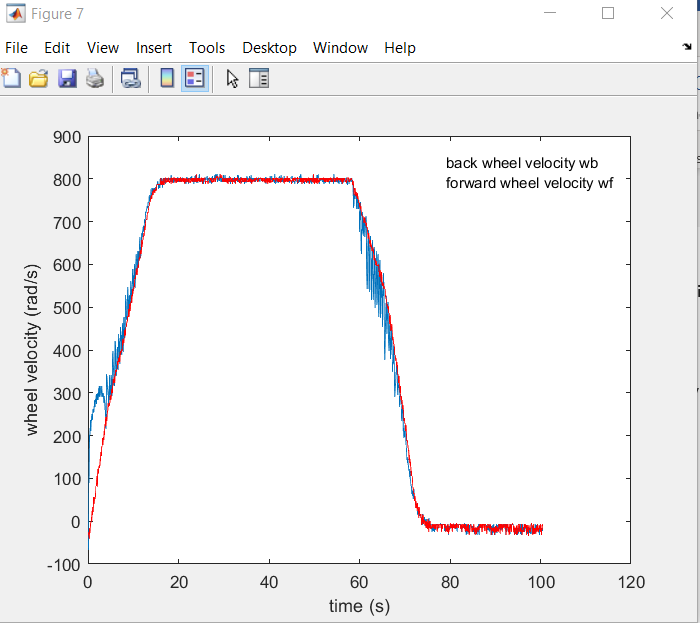


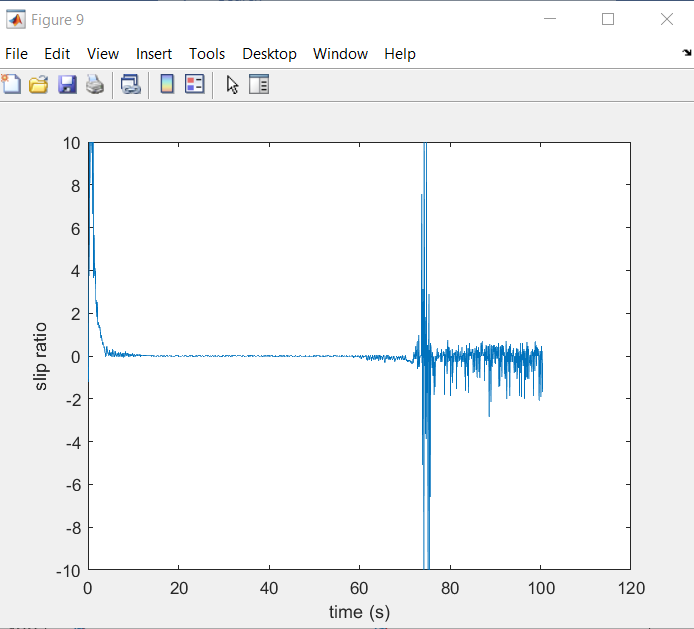


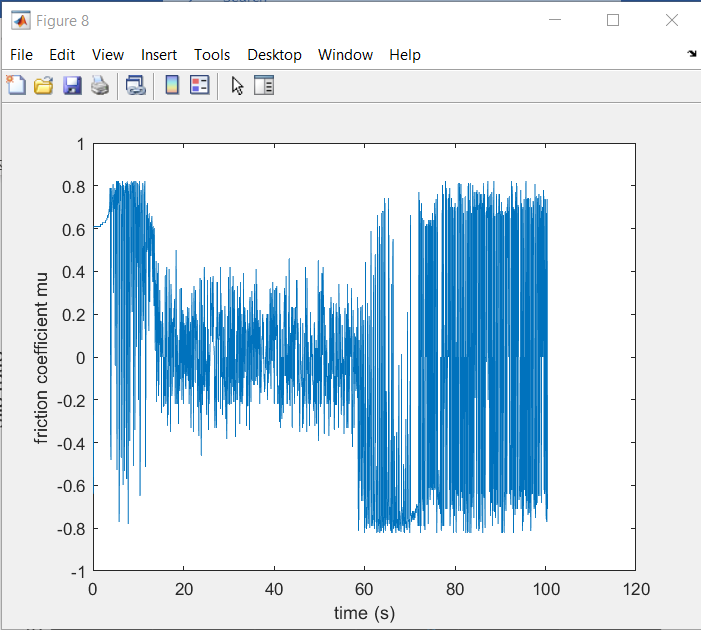


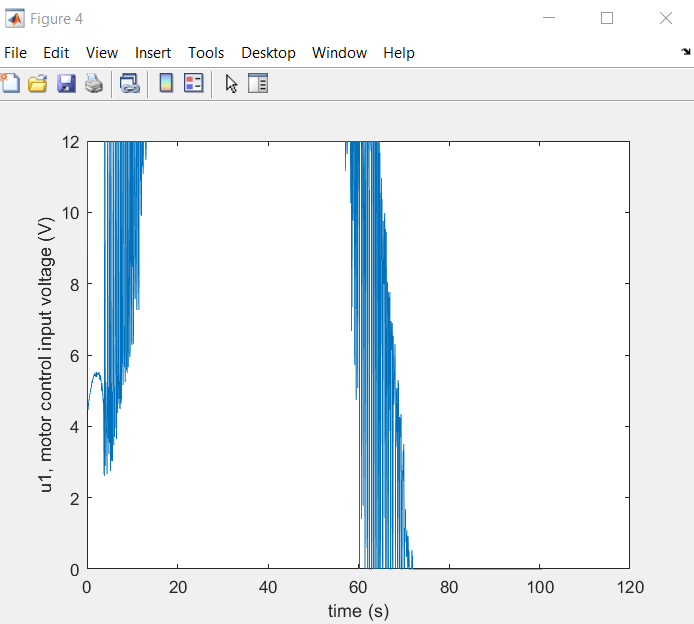
Therefore, we find out that between -0.8 and -1.0, the car can brake with good traction control and time response.

Therefore, we set parameter to kp = 20 ,ki=2,kd = 0 to illustrate full speed launch, cruise and brake(with traction). And the HIL response is shown as below:









To illustrate normal speed launch, cruise and brake, we set kp = 16, ki = 0, kd= 0 and mu =0.4(should set final speed to 400rads/s). And the HIL response is shown as below:

