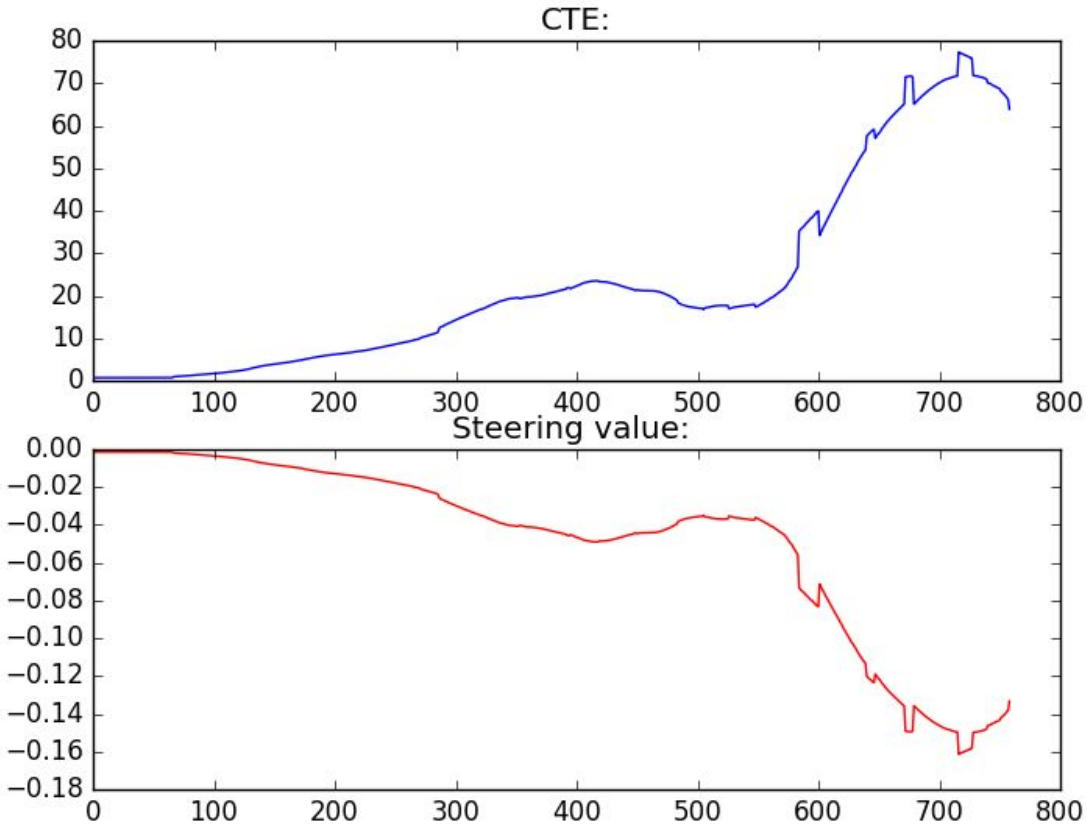
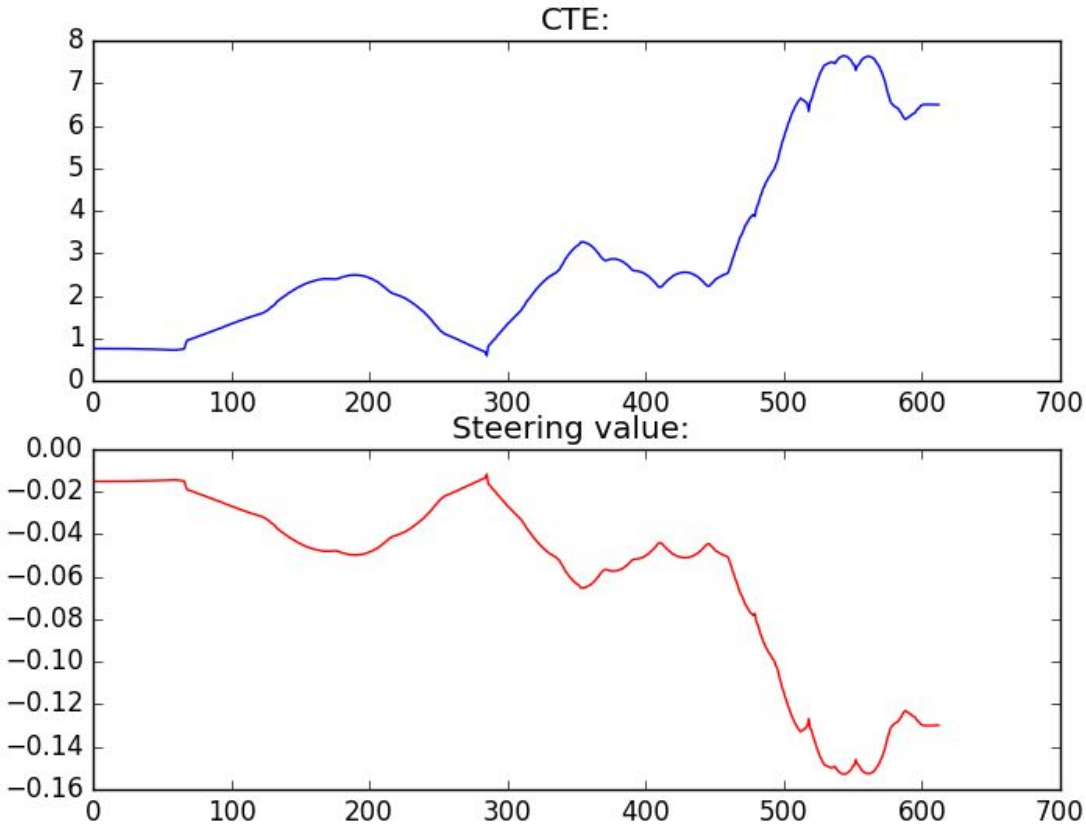
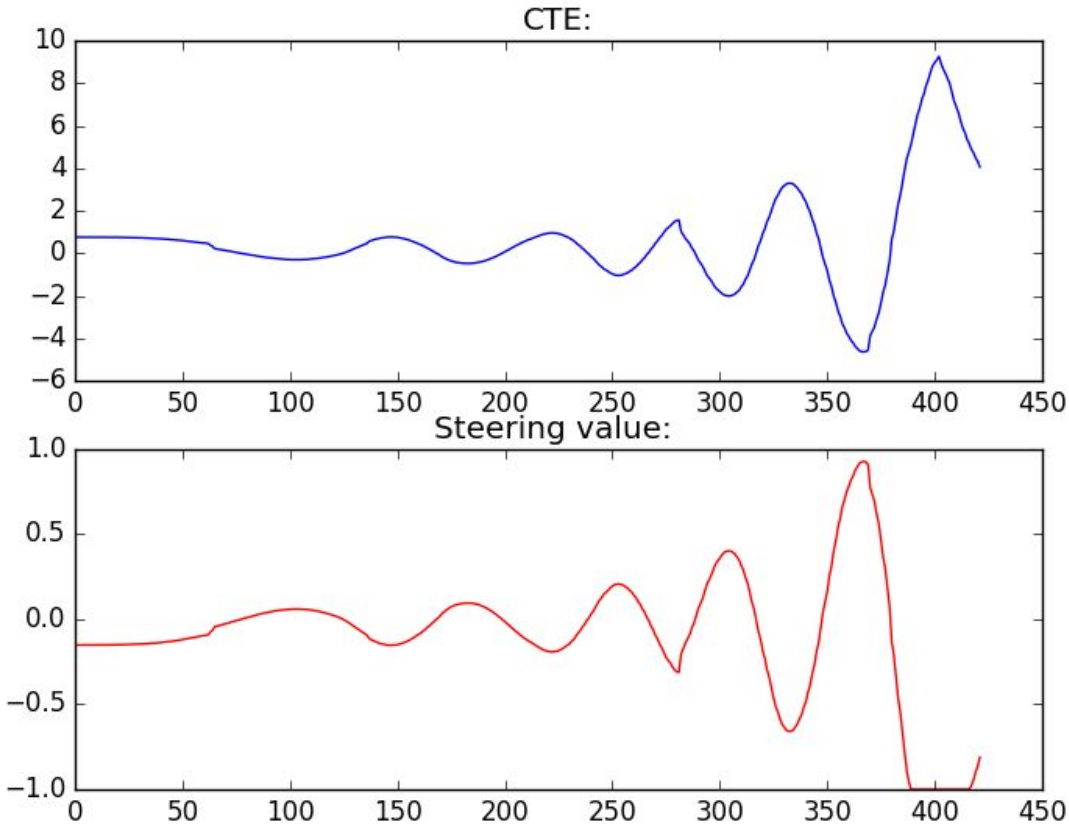
To tune parameters I used python code from lesson to create robot_simulator.py. Script gave me good starting for further optimization.
When considering only P controller, simulator outputs [0.002, 0.0, 0.0] as optimum values of parameters. These values, when used inside PID controller did not produce good result.



After increasing value of Kd to 0.02 I get

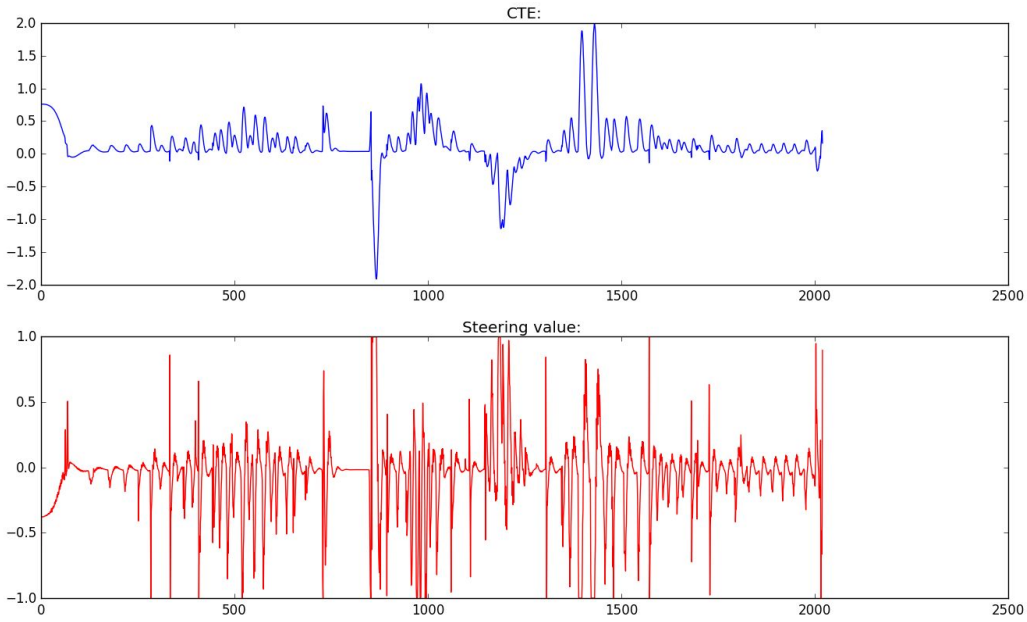


After increasing Kt to 0.2 I get similar behaviour as demonstrated in lectures



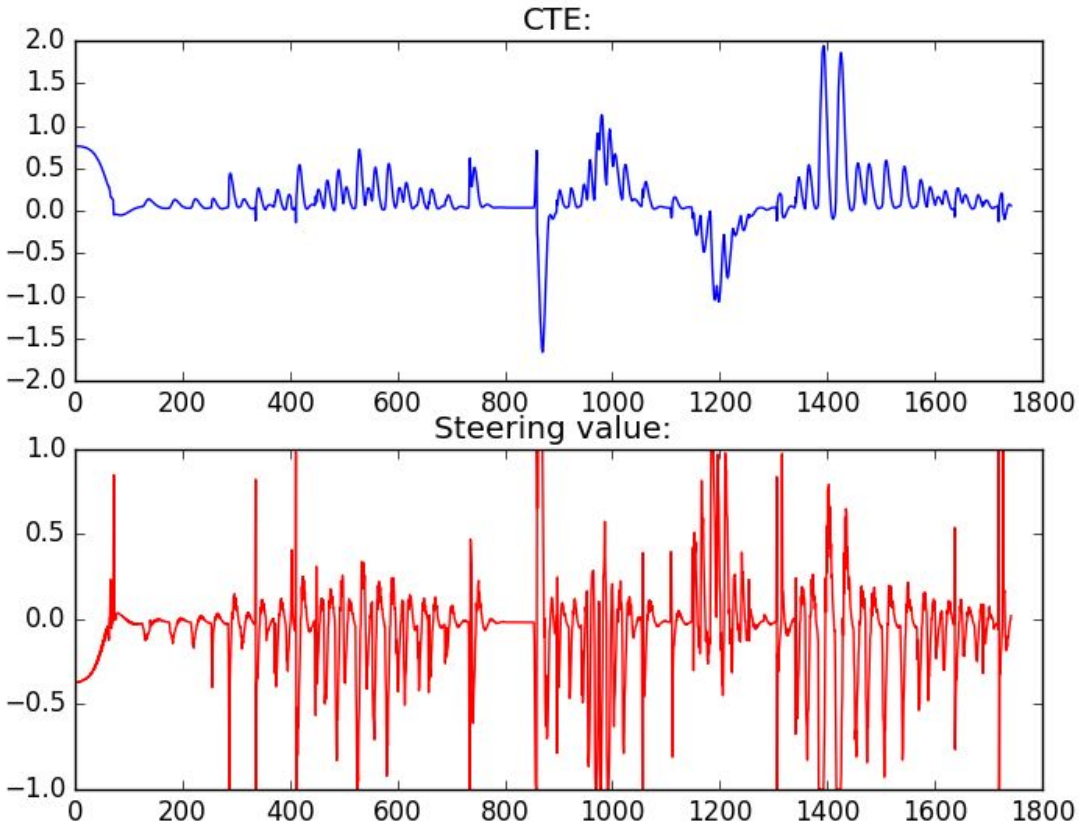
Obviously, P controller overshoots when trying to steer, this produces positive feedback loop and car steers of the track.

Using PD controller in simulator outputs values [0.49720621312975055, 0.0, 5.134407401149259] ([Kp, Ki, Kd] respectively). Trying these parameter values produces controller that successfully finished one circle on the track.



Car is still little erratic in its steering.

Optimizing PID controller in simulator outputs [0.4884067975842662, 1.274358677148635e-12, 5.096416801295116]. Parameter Ki is dependent on each individual vehicle so I set it to 0.0. Using these parameters seem to improve control of the car compared to PD controller while car finished circle on the track.



Using these values as starting point I tuned parameters even further using total square error after 1700 calls of PID controller.

Kp	Ki	Kd	Sum square errors
0.4884067975842662	0.0	5.096416801295116	232.803
0.4884067975842662	1.0e-5	5.096416801295116	238.367
0.4884067975842662	-1.0e-5	5.096416801295116	231.012
0.4884067975842662	-2.0e-5	5.096416801295116	234.421
0.4884067975842662	-5.0e-6	5.096416801295116	234.333

It seems that Ki parameter does not influence error much. Therefore,I fixed Ki on -1.0e-5, and continued experimenting with other parameters.

Kp	Kd	Sum square errors
0.49	5.096416801295116	235.072
0.485	5.096416801295116	234.098
0.4875	5.096416801295116	217.618
0.4875	5.09	277.717
0.488	5.096416801295116	222.666
0.48775	5.096416801295116	235.945
0.4875	**5.1**	**209.001**
0.4875	5.2	232.201