PID Controller

The implementation basically followed Sabastian's Algorithm.

Steer = -tau_p * cte - tau_d * diff_cte - tau_i * int_cte

Cte: cross track error
Diff_cte: difference of CTEs between two samples
Int_cte: integral of CTE errors
tau_p, tau_d, and tau_i are three parameters

"-tau_p * cte" directly compensates CTE in proportion.
"- tau_d * diff_cte" makes it smoothly approaching the target line.
"- tau_i * int_cte" eliminates bias of mechanic issue.

ZieglerNichols method provides a method to calculate tau_p, tau_d, tau_i.

I recorded a series of trials and errors under Images directory.

"tau_p" was chosen to 0.2
When tau_d is zero, it couldn't complete track with S shape driving.
"tau_d = 0.1" is speedy. People might get sea sick easily.
"tau_d = 0.2" is more comfortable than 0.1.
When tau_d gets larger, it slows the car. When it over-componstates, car shifts to right or left with unnecessary drag force.

tau_d	recording
0.00	D_00.mov
0.10	D_010.mov
0.20	D_020.mov
0.25	D_025.mov
0.30	D_030.mov
0.40	D_040.mov
0.50	D_050.mov
0.60	D_060.mov
1.0	D_100.mov
1.5	D_150.mov

The car has a small bias, neglect-able.
I_00.mov shows recording with tau_i equal to zero.
I_05.mov shows recording with tau_i equal to 0.5, which couldn't complete the track.