Dbw Node Implementation:

Summary:

Dbw node deals with feedback controller loop that is considered as last component in the hierarchical decision-making process as referred in Survey paper Survey of Motion Planning and Control Techniques for self-driving urban vehicles and hierarchical decision-making process involves

1. Route Planning (input: Road Network data and User specified destination)
2. Behavior Layer (input: perceived agents, obstacles and signage)
3. Motion Planning(Estimated pose and collision free space).
4. Local feedback Control (estimate of vehicle state)

Output of above process lead to – ***Steering, throttle and brake commands***

In current project, Route Planning is already provided as input to project and loaded using ***waypoint_loader.py***. Behavior and Motion planning are provided by ***waypoint_updater.py*** and ***tl_detector.py***. Local feedback Control implemented in ***pure_pursuit.cpp***, ***dbw_node.py*** and ***twist_controller.py*** involves two sub components

1. Path Stabilizer
2. Controller that can adjust for errors in Path stabilizer.

As per current industry survey, Path Stabilization for a kinematic model has many standard techniques like

1. Pure pursuit – used by three vehicles in DARPA Urban challenge
2. Rear Wheel position based feedback
3. Front Wheel position based feedback – used Stanford in DARPA Grand challenge in 2005 (Stanley)

From above survey, results show trajectory generated by Rear Wheel and Front Wheel position based feedback show very less error whenever CURVATURE is present and there is considerable error in PURE PURSUIT(page#20 of above survey paper) and therefore, it is more critical to have good CONTROLLER mechanism that can minimize the error.

Our team has implemented PID controller for VELOCITY error correction that output THROTTLE and BRAKE commands. We used YAW CONTROLLER(provided by Udacity) that calculates STEERING and apply PID controller to correct STEERING error and also generates output as STEERING command.

Implementation:

It was decided to use below components:

1. To continue with Pure pursuit algorithm that is used for path stabilization.
2. Use PID to correct Velocity and Steering errors that will generate Steering, Brake and Throttle.
3. Develop components that will help in fine tuning

   a. Path monitoring tool that an show base_waypoints and real-time update of plot with current car’s path.
   b. Develop twiddle to fine tune PID gain tuning.
4. Velocity PID controller will be based on (Required Linear Velocity – Current Linear Velocity)
5. Steering PID controller will be based on (steering report’s current steering – estimated current steering)

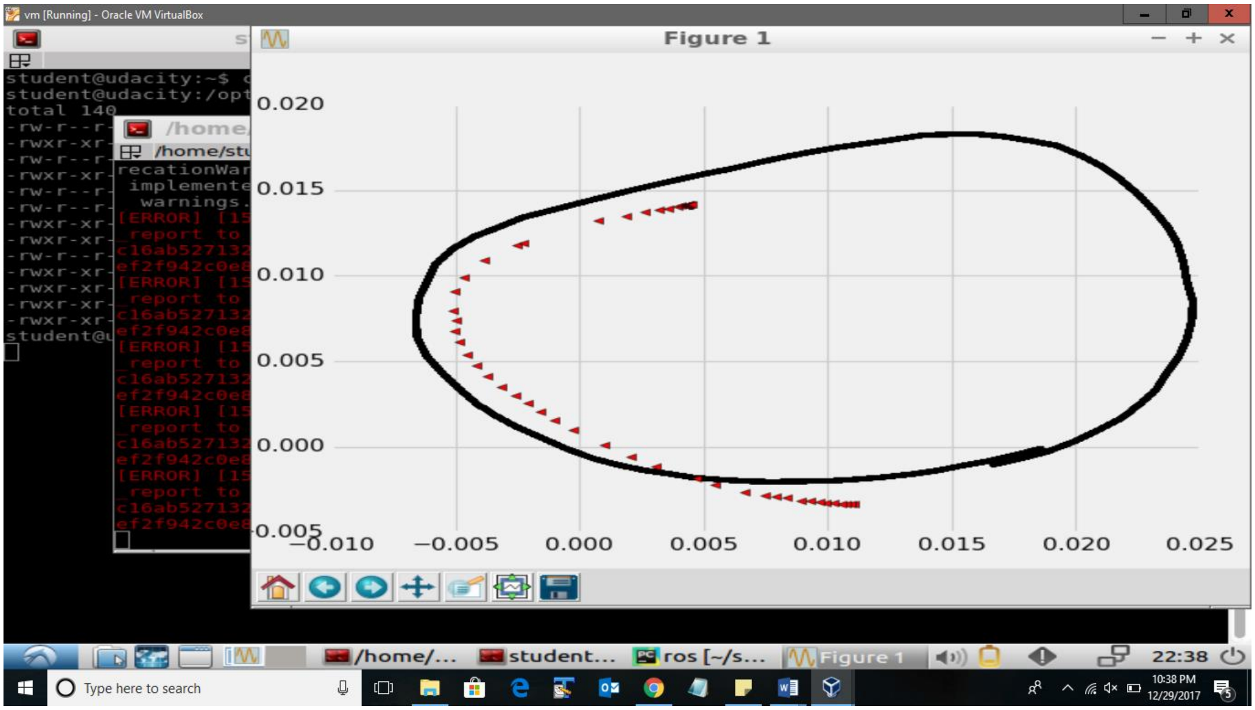
Challenges:

We faced many challenges while implementing the controller and below are few mentioned

1. Collecting metrics that can verify closeness of car current path and base_waypoints helps developers to identify if they are proceeding in correct direction for PID controller development because current controllers implemented as VELOCITY and STEERING controllers that contribute THROTTLE, BRAKE and STEERING instead of Car’s POSITION on the planned trajectory. Therefore, implemented Plot.py that will generate a plot and does below two steps
   a. Generate path on the plot based on base_waypoints
   b. Mark cars travelled path overlapping with base_waypoints that can help in visualizing entire path.

   This feature can be enable by including following line in launch files

   <node pkg=”twist_controller” type=”plot.py” name=”plot_node”/>



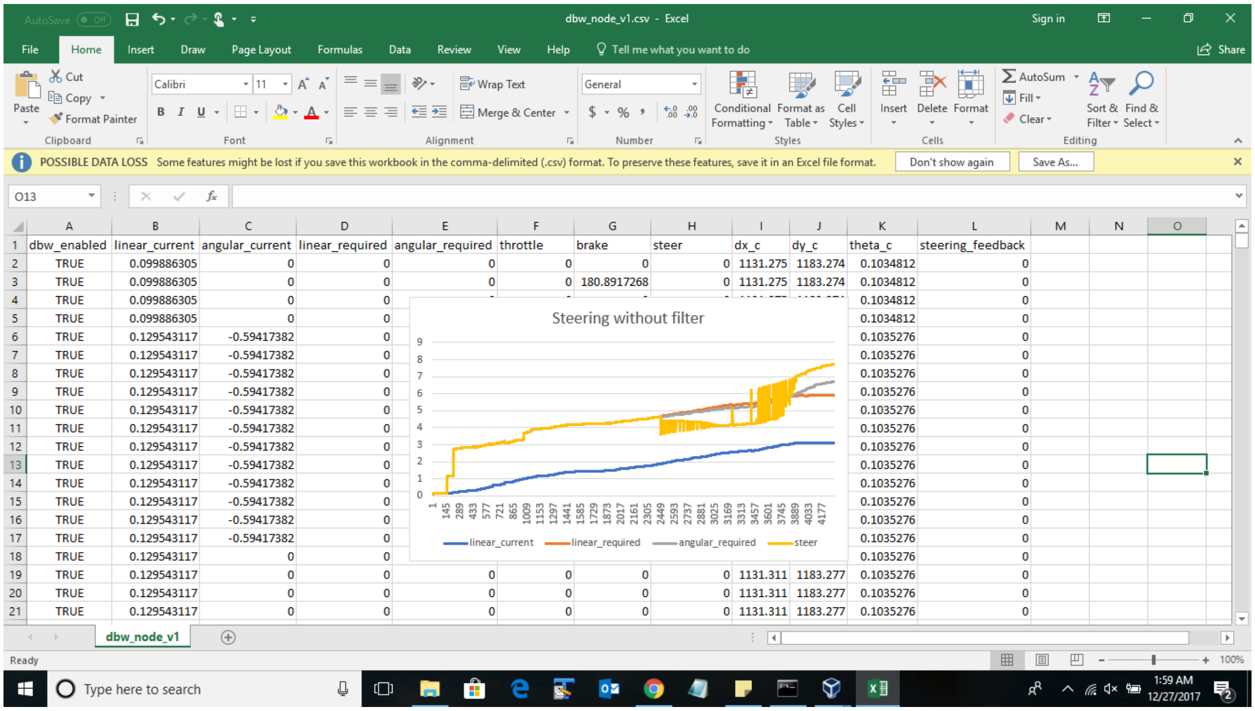
```
student@udacity:~$ c
student@udacity:/opt
total 140
-rw-r--r-
-rwxr-xr-
-rw-r--r-
-rwxr-xr-
-rw-r--r-
-rw-r--r-
-rwxr-xr-
-rwxr-xr-
-rw-r--r-
-rwxr-xr-
-rwxr-xr-
-rwxr-xr-
-rwxr-xr-
student@u
```

```
/home/stu
recationWar
 implemente
  warnings.
[ERROR] [15
 report to
c16ab527132
ef2f942c0e8
[ERROR] [15
 report to
c16ab527132
ef2f942c0e8
[ERROR] [15
 report to
c16ab527132
ef2f942c0e8
[ERROR] [15
 report to
c16ab527132
ef2f942c0e8
[ERROR] [15
 report to
c16ab527132
ef2f942c0e8
```

2. Tuning PID controller gains. There are two PID controller(velocity pid controller and steering pid controller). Yaw controller calculates steering, however, steering PID controller has been put upon YAW controller to adjust for errors. Below are controller details
    a. To calculate Throttle and brake, PID is placed on the velocity error(**Required Linear velocity – Current Linear velocity**). PID controller is used to identify velocity error correction. If correction is +ve it is directly used as **throttle** otherwise used as **brake**.
    b. To control Steering, Yaw Controller outputs Steering value that is calculated based on current linear velocity, required linear and angular velocity. Using **/vehicle/steering_report's current steering** and **Calculated Steering value**, difference will be calculated and used as input to PID controller to adjust for errors.

NOTE: *There seems like some problem with /vehicle/steering_report because it is always generating ZERO steering value instead of current steering value*.

Step wise activities performed in tuning PID controller has been captured on plot and shown below.



POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

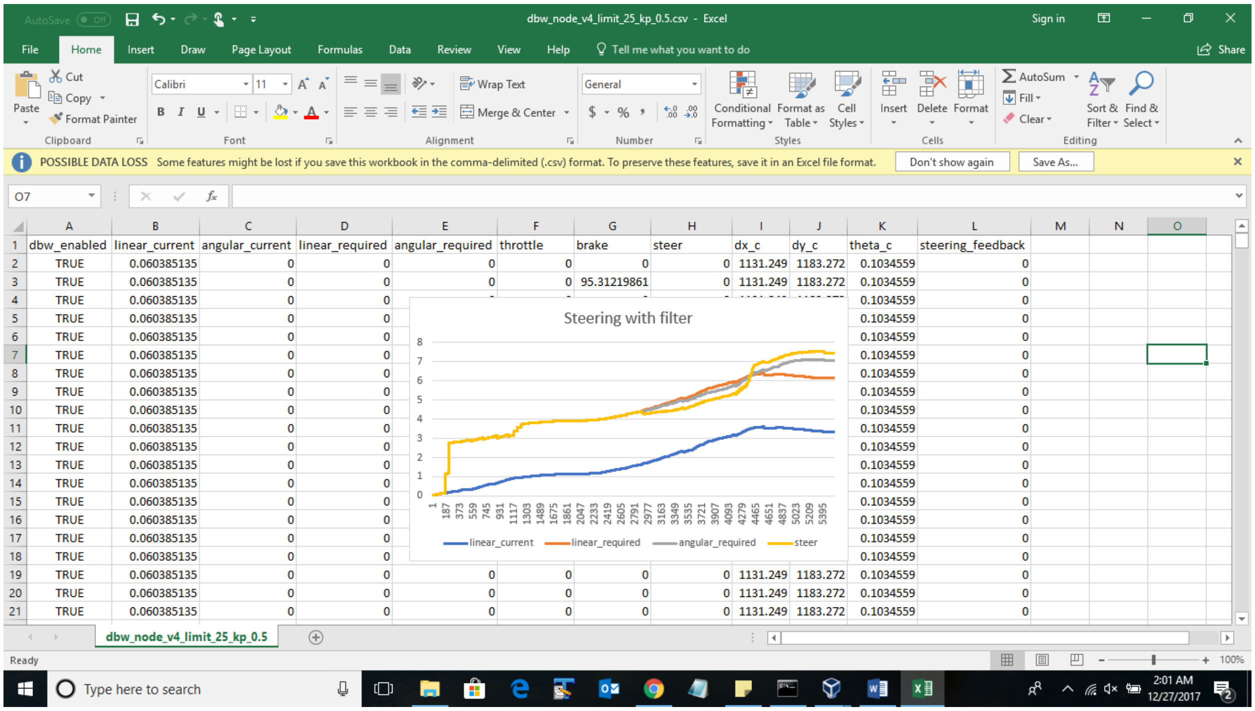
	dbw_enabled	linear_current	angular_current	linear_required	angular_required	throttle	brake	steer	dx_c	dy_c	theta_c	steering_feedback
2	TRUE	0.099886305	0	0	0	0	0	0	1131.275	1183.274	0.1034812	0
3	TRUE	0.099886305	0	0	0	0	180.8917268	0	1131.275	1183.274	0.1034812	0
4	TRUE	0.099886305	0	0							0.1034812	0
5	TRUE	0.099886305	0	0							0.1034812	0
6	TRUE	0.129543117	-0.59417382	0							0.1035276	0
7	TRUE	0.129543117	-0.59417382	0							0.1035276	0
8	TRUE	0.129543117	-0.59417382	0							0.1035276	0
9	TRUE	0.129543117	-0.59417382	0							0.1035276	0
10	TRUE	0.129543117	-0.59417382	0							0.1035276	0
11	TRUE	0.129543117	-0.59417382	0							0.1035276	0
12	TRUE	0.129543117	-0.59417382	0							0.1035276	0
13	TRUE	0.129543117	-0.59417382	0							0.1035276	0
14	TRUE	0.129543117	-0.59417382	0							0.1035276	0
15	TRUE	0.129543117	-0.59417382	0							0.1035276	0
16	TRUE	0.129543117	-0.59417382	0							0.1035276	0
17	TRUE	0.129543117	-0.59417382	0							0.1035276	0
18	TRUE	0.129543117	0	0							0.1035276	0
19	TRUE	0.129543117	0	0	0	0	0	0	1131.311	1183.277	0.1035276	0
20	TRUE	0.129543117	0	0	0	0	0	0	1131.311	1183.277	0.1035276	0
21	TRUE	0.129543117	0	0	0	0	0	0	1131.311	1183.277	0.1035276	0

Above is initial setup. With PID controller for both velocity and steering.

I have setup

gains velocity = (kp, ki, kd) = (1.0,0.0, 0.0)

gains steering = (kp, ki, kd) = (1.0,0.0, 0.0)



dbw_node_v4_limit_25_kp_0.5.csv - Excel

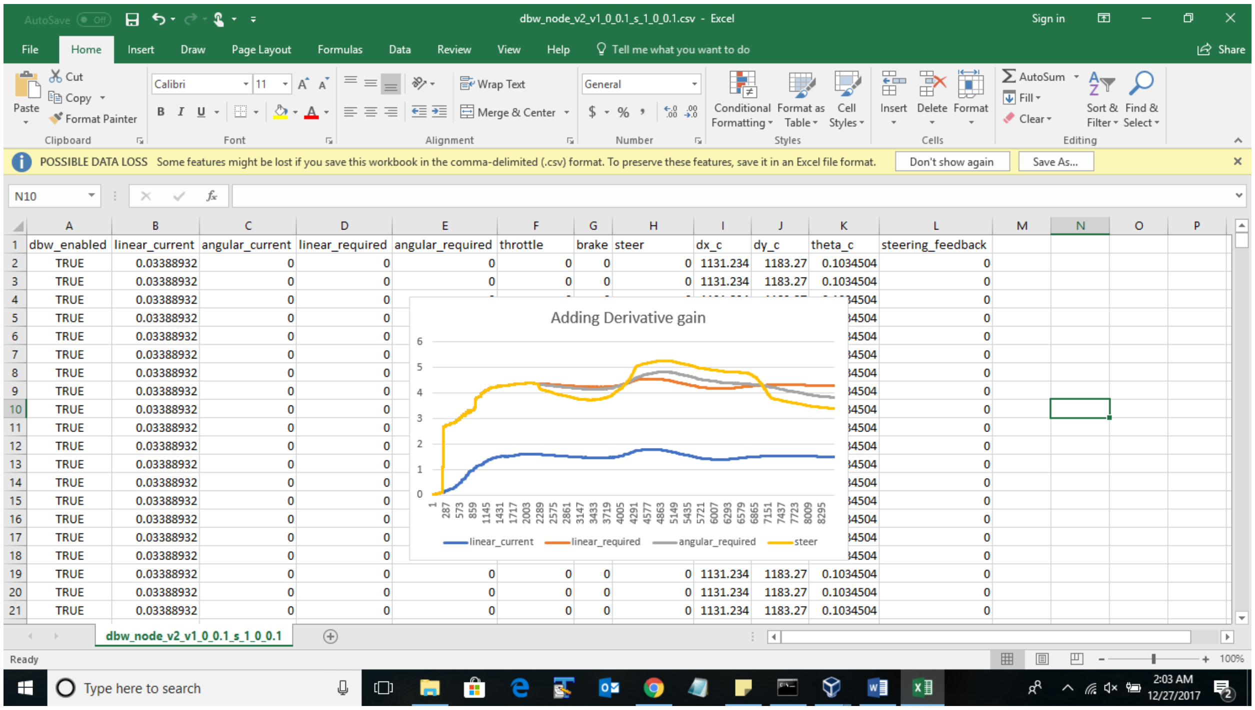
POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

O7

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	dbw_enabled	linear_current	angular_current	linear_required	angular_required	throttle	brake	steer	dx_c	dy_c	theta_c	steering_feedback			
2	TRUE	0.060385135	0	0	0	0	0	0	1131.249	1183.272	0.1034559	0			
3	TRUE	0.060385135	0	0	0	0	95.31219861	0	1131.249	1183.272	0.1034559	0			
4	TRUE	0.060385135	0	0							0.1034559	0			
5	TRUE	0.060385135	0	0							0.1034559	0			
6	TRUE	0.060385135	0	0							0.1034559	0			
7	TRUE	0.060385135	0	0							0.1034559	0			
8	TRUE	0.060385135	0	0							0.1034559	0			
9	TRUE	0.060385135	0	0							0.1034559	0			
10	TRUE	0.060385135	0	0							0.1034559	0			
11	TRUE	0.060385135	0	0							0.1034559	0			
12	TRUE	0.060385135	0	0							0.1034559	0			
13	TRUE	0.060385135	0	0							0.1034559	0			
14	TRUE	0.060385135	0	0							0.1034559	0			
15	TRUE	0.060385135	0	0							0.1034559	0			
16	TRUE	0.060385135	0	0							0.1034559	0			
17	TRUE	0.060385135	0	0							0.1034559	0			
18	TRUE	0.060385135	0	0							0.1034559	0			
19	TRUE	0.060385135	0	0	0	0	0	0	1131.249	1183.272	0.1034559	0			
20	TRUE	0.060385135	0	0	0	0	0	0	1131.249	1183.272	0.1034559	0			
21	TRUE	0.060385135	0	0	0	0	0	0	1131.249	1183.272	0.1034559	0			

Steering with filter

In above setup, I have restricted steering to -25 degree to 23 degrees. Also, I have added filter after Steering PID.



dbw_node_v2_v1_0_0.1_s_1_0_0.1.csv - Excel

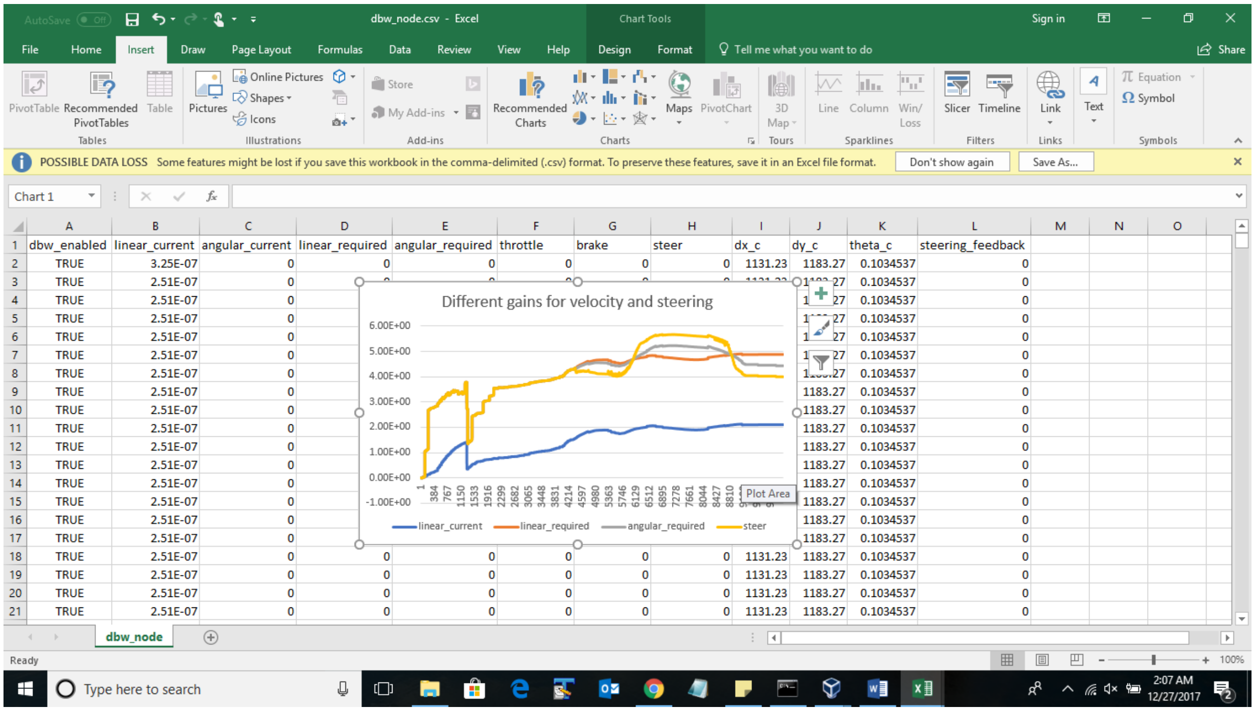
POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

N10

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	dbw_enabled	linear_current	angular_current	linear_required	angular_required	throttle	brake	steer	dx_c	dy_c	theta_c	steering_feedback				
2	TRUE	0.03388932	0	0	0	0	0	0	1131.234	1183.27	0.1034504	0				
3	TRUE	0.03388932	0	0	0	0	0	0	1131.234	1183.27	0.1034504	0				
4	TRUE	0.03388932	0	0							34504	0				
5	TRUE	0.03388932	0	0							34504	0				
6	TRUE	0.03388932	0	0							34504	0				
7	TRUE	0.03388932	0	0							34504	0				
8	TRUE	0.03388932	0	0							34504	0				
9	TRUE	0.03388932	0	0							34504	0				
10	TRUE	0.03388932	0	0							34504	0				
11	TRUE	0.03388932	0	0							34504	0				
12	TRUE	0.03388932	0	0							34504	0				
13	TRUE	0.03388932	0	0							34504	0				
14	TRUE	0.03388932	0	0							34504	0				
15	TRUE	0.03388932	0	0							34504	0				
16	TRUE	0.03388932	0	0							34504	0				
17	TRUE	0.03388932	0	0							34504	0				
18	TRUE	0.03388932	0	0							34504	0				
19	TRUE	0.03388932	0	0	0	0	0	0	1131.234	1183.27	0.1034504	0				
20	TRUE	0.03388932	0	0	0	0	0	0	1131.234	1183.27	0.1034504	0				
21	TRUE	0.03388932	0	0	0	0	0	0	1131.234	1183.27	0.1034504	0				

Adding Derivative gain

I have added derivative gain as 0.1



POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format. Don't show again Save As...

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	dbw_enabled	linear_current	angular_current	linear_required	angular_required	throttle	brake	steer	dx_c	dy_c	theta_c	steering_feedback			
2	TRUE	3.25E-07	0	0	0	0	0	0	1131.23	1183.27	0.1034537	0			
3	TRUE	2.51E-07	0							1183.27	0.1034537	0			
4	TRUE	2.51E-07	0							1183.27	0.1034537	0			
5	TRUE	2.51E-07	0							1183.27	0.1034537	0			
6	TRUE	2.51E-07	0							1183.27	0.1034537	0			
7	TRUE	2.51E-07	0							1183.27	0.1034537	0			
8	TRUE	2.51E-07	0							1183.27	0.1034537	0			
9	TRUE	2.51E-07	0							1183.27	0.1034537	0			
10	TRUE	2.51E-07	0							1183.27	0.1034537	0			
11	TRUE	2.51E-07	0							1183.27	0.1034537	0			
12	TRUE	2.51E-07	0							1183.27	0.1034537	0			
13	TRUE	2.51E-07	0							1183.27	0.1034537	0			
14	TRUE	2.51E-07	0							1183.27	0.1034537	0			
15	TRUE	2.51E-07	0							1183.27	0.1034537	0			
16	TRUE	2.51E-07	0							1183.27	0.1034537	0			
17	TRUE	2.51E-07	0							1183.27	0.1034537	0			
18	TRUE	2.51E-07	0	0	0	0	0	0	1131.23	1183.27	0.1034537	0			
19	TRUE	2.51E-07	0	0	0	0	0	0	1131.23	1183.27	0.1034537	0			
20	TRUE	2.51E-07	0	0	0	0	0	0	1131.23	1183.27	0.1034537	0			
21	TRUE	2.51E-07	0	0	0	0	0	0	1131.23	1183.27	0.1034537	0			

I have setup different gains for velocity and steering

gains velocity = (kp, ki, kd) = (1.0,0.01, 6.0)

gains steering = (kp, ki, kd) = (0.65,0.0, 6.0)

From above picture, we can see there is smoothness, however, still swing in steering.

3. From above steps, it is clear that car is still swinging and need some tuning on PID gains. Therefore, implemented Twiddle algorithm(one for Velocity PID and other for Steering PID). These Twiddle algorithms can be enabled by enabling flag in twist_controller.py

    twiddle_on = True
    vel_twiddle_on = True.

After running few steps on the simulator PID gain can fine tuned. Once user realizes PID gains are well tunes, above flags can be set to False.

Initially tuning was focused on Velocity PID and it is observed that velocity is very low **0.00008.** Therefore to start tuning, at this point, Velocity PID gain(Kp, Ki, Kd) set to (1,0,0). By setting Kp=1 and other value as ZERO will apply Velocity error as Throttle/Brake without any additional adjustments from derivative component(Kd) and integral component(Ki).

Steering PID gain(Kp, Ki, Kd) set to (0,0,0) along with best_error = 999.0 and this was allowed to run through twiddle algorithm. It is little difficult to implement it on simulator as it is not easy to restart the simulator very often to reset the best_error. Therefore, after every cycle of Twiddle algorithm, Gain values(Kp, Ki, Kd) were retained and best_error was reset back to very high value.

Improvements required:

1. Twiddle algorithms must be reviewed and need some adjustment that will help in tuning PID gains.
2. Enabling and Disable Twiddle algorithm to fine tune gain should be parameterized in launch file.
3. /vehicle/steering_report need fix that will give correct "current steering value in simulator"
4. Pure pursuit algorithm for trajectory stabilizes is one of the oldest, however, might not be best algorithm. It would be good idea to implement front/back wheel position based feedback loop.