

Test 1:

Test 2:

	Speed	k_p	k_d	Start pos	Battery voltage	<u>Result</u>
T1)	40	0,01				Car failed on milestone #2 $c_p = 0,01$
T2)	40	0,01				

we put it on the straight line part
and it failed every slight turn
but did correct late
Conclusion → we forgot to
remove the serial prints
and delays

T3) 40 0,0 | Failed sharp turn at start immediately

T4) 40 0,0 | When put in straight line part the car navigated the turns somewhat well.
However, Megan told us our car needed to always pick the left path when there were two paths ahead

T5 40 0,0 | Went off at first checkpoint (sharp turn). Problem seems to be that the car doesn't react or reacts late?
Maybe we reduce base speed?

T6 25 0,0 | Was able to navigate more of the sharp turn but still went off.
Ronit's idea: Car goes off into the white completely before completing the turn, making the car go straight.

T7 5 0,0 | Didn't move

T8 10 0,0 | Barely moved, went forward but couldn't do forward on turns

T9 15 0,0

Failed sharp turn but rest
of track was fine (didn't choose
left paths at all),

T10 15 0,05

still couldn't navigate sharp
turn, considering doing something
else,

T11 15 0,05

Test run to make sure nothing
changed from last testing, did
not run the same. Redoing
code from previous runs

T12 20 -0,01

car works like normal (car
goes off Turn 5 but navigates
straightish paths fairly well)
like previous

T13 20 -0,01 $\frac{k_d}{0,05}$

Testing our derivative term
to see its impact on the
car's movement. Helps a bit
with sharp turn and
navigates better.

T14 15 -0,01 -0,05

Turns extreme still → conclusion:
 k_d is too high

T15 15 -0,01 -0,03

Changed derivative logic to
make one wheel turn backward. It
only goes to 0 though. This
is because it is capped
at 0. We are putting an
if statement to make
the wheel go backward

T16 15 -0,0 | -0,03 added conditionals to allow wheels to go backward

T17 25 -0,01 -0,03 sped up the car, especially not navigate turn any longer

T18 20 -0,02 -0,03 Terrible, can't to turn, can't do anything on straight (spins constantly)

T19 15 -0,01 -0,03 Can't turn correctly. But, does very well on great straight lines

T20 25 -0,01 -0,03 - we aren't using change wheel speed anymore (directly analogous to the speed) will speed up changes

Result! S TURN WORKS! straight line is also good

T21 25 -0,01 -0,03 Now trying to detect splits (with a serial print for testing)

↳ Detected splits

Well

T22 Put in split logic, instead of going in middle of split it oscillitated back

and forth

T23 Retested code from before to make sure stuff still worked (S turn navigated properly)

T24 25 -0,0 | -0,03: We tested our detectsplit function on the split with a turn speed of 10. Split not properly navigated

T25 25 -0,0 | -0,03: We tested our detectsplit function on the split with a turn speed of 15. Split

T26 25 -0,0 | -0,03: Test to find what the sensor values look like at the end piece. Middle sensor is weird but the rest are consistent in the 2475-2500 range. Middle sensor is in the 2400-2450 range.

T27 25 -0,0 | -0,03: we left something weird (shouldn't have been there) in the code for when the car did the split correctly. Testing again with that part removed to see if performance is the same.

Result: Split is fine

T28'; 23 -0,01 -0,03'; Test with crossbar end piece code implemented, Result: just turns left all the time

T29'; condition was wrong just fixed it. Still just turns left off track

T30'; Made code return before PID to avoid sharp turn, PID still activated

T31'; Noticed a possible bug in split logic where we reversed the signs. Fixing it made the car go left much better. However, the car goes right again at end of split going, splitIters = 7 in a circle). We tried increasing splitTurn = 20 (from 15) split turn even more

T32'; Tried new split logic. For some reason a endpiece/crosspiece is detected before we even get on the S-turn?? This error is some bug from adding new code

T34' (changed kd to 0,04 from 0,03); Tested old code again to confirm it works

T35'; Changed split logic to rely on "peaks" → split detection is good but still doesn't do / etc

T36, ^{0103 main} Made left turn much sharper on
splits → works on split but
NGT arch
(now 0,015)
T37! Failed arch on the way there, on
the way back the split failed
T38!, We made turnspeed 80 (from 70) to
go left a bit more extremely
T39!, Made turnspeed 100 to see if
the split on the arch is being
detected or not → did not indicate detection
T40!, Slightly changed logic of our split arch
detection function, Before, values
were identified as LOW and HIGH
and we counted the number of
changes from one to the other,
we now use the difference in
values which is more reliable.
Split detection was perfect on
back,

T41', Split worked (detected)

T42', Arch worked (detected)

T43', Arch worked (detected)

T44', Split worked (detected)

→ Experimented with turn speeds to pick split and arch

T45', 40 → not enough for arch

T46', 50 → not enough for arch

T47', 90 → turn was too much for arch

T48', 70 → did arch well

T49', 70 → did arch well again

K_P/K_D shifting because jump now did not work, our jump is completed simply with PID,

T50', K_P=0,01, K_D=-0,03 → went off ESSS jump

TS1', K_P=0,07, K_D=-0,03 → Overshot turn

TS2; $k_p = 0.06$, $k_d = -0.03 \rightarrow$ overshoot jump

* deleted k_d code (Alexiy's suggestion)

TS3; $k_p = -0.06$, $k_d = 0 \rightarrow$ jump overshoot
and esses overshoot

TS4; $k_p = -0.04$, $k_d = 0 \rightarrow$ jump

was completed but Esses still went
off from too much of a turn

TSS; $k_p = -0.0317 \rightarrow$ entire track

specificness ↗ worked! (except
crosspiece which occasionally
bugged out like before. Essentially,
sometimes the car randomly (or
straight lines) detected the crosspiece

was "for fun", but
happened to work
so we'll not
touching it

Testing for crosspiece bug:

After looking through our code we
noticed that we didn't reset our
count for detecting crosspiece. This led

to false positives as the count accumulated (the "defect crosspieces" didn't have to be consecutive)

TS6; $fp = -0.0317$. The crosspiece was detected right and the entire track was traversed correctly.

Testing for reliability/consistency in the car traversing the track.

TS7; Went off EsseS (realized we uploaded wrong code)

TS8; Went off EsseS

TS9; Same conditions as TS8, but it completed the entire track (didn't detect 2nd crosspiece though).

TS10; Almost completed entire track. Failed last EsseS (same as TS9)

T61; Changed -4 weight on outside fc
-5, Failed jump, Failed split, Test
terminated,

T62; Removed serial prints and
every thing worked

T63; ✓ T65; ✓ T67; ✓
T64; ✓ T66; ✓ T68; ✓

Speed UP

T69;
Speed; 25 (from 22) \rightarrow worker

T70;
Speed; 28 (from 25) \rightarrow went off
Eses

T71; Speed; 30, $k_p = 0,06$ \rightarrow worker

T72; Speed; 40, $k_p = -0,08 \rightarrow$ almost
perfect but failed jump on way
back

T73; Speed; 40, $k_p = -0,08$, changed
weights of outside from 5 to 4
 \rightarrow jump on way back failed

T74! speed: 35, kp = -0.07. Worked
flawlessly

T75; Same as T74, Worked flawlessly

Race day plan! first run at 35_{if}
that fails, run at 22 or 25,
since that's basically a guaranteed
complete run.