Final Report

EC ENGR 3, Spring 2025, June 10

Lab Section 1F

By Shaan Mistry and Ashley Torres

Question 1: Plan the Development Tests

Our development began by writing code to read raw sensor values each loop iteration. Initially, we debated whether to average five sensor readings or use just one. We started with averaging to reduce the impact of random noise. The steering correction was calculated by taking a weighted sum of the normalized sensor values to find an error, then multiplying it by a Kp value of 0.012. This correction was subtracted from the right motor speed and added to the left motor speed. We broke the track into segments and tested only the relevant parts for each development phase. This saved time and allowed us to quickly identify what was or wasn't working before testing the entire loop. We prioritized smooth turns, straight paths, and then progressively moved to more complex features like the "esses," the 180-degree turnaround, and the split.

When first tested, the car was unable to follow the path consistently. We introduced a Kd term (starting at 0.06), which improved response time and reduced overcorrection, but the car still lagged during sharp turns. To improve reaction speed, we removed the averaging process and instead based corrections on each single reading. To avoid misinterpretation of phantom crosspieces as track lines, we added logic to ignore a single frame where all sensors read normalized values over 900. However, if this occurred more than once in a row, we treated it as a real trigger. This helped the car recognize significant visual cues.

We noticed that during sharp turns, the robot needed one motor to spin backward. So, we implemented logic to check if motor speeds were negative and reverse direction using the dir pins. To sharpen response when this reversal occurred, we multiplied the steering correction by 1.5 if the direction changed, allowing the car to get into turns faster and handle “jump” segments more confidently.

Later, we refined our turnaround logic. When the car saw a streak of high readings across multiple loops for the first time, it performed a 180-degree turn. If it saw a similar streak again later in the run, it stopped entirely, indicating it had completed its return loop. To ensure reliability near the split and arch, we customized the weighting system to have heavier inner weights on the left side during the outbound trip, and then the same on the right side during return. After extensive testing, we finalized Kp and Kd at 0.0355 and 0.1775, respectively.

Question 2: Conduct the Tests

One development test problem we had difficulty solving was determining the optimal weight distribution before and after the 180 degree turning point. Initially, the car did not consistently ignore the second black line on the right of the split and the arch. We adjusted the weights array on the second half of the distribution, slightly increasing them so the left-side sensors would be more valued than the right. This bias was intended to help the car to follow the left line. On the track, it showed some minor improvements in these detections, but the results overall were the same. We continued incrementally adjusting the weights, focusing primarily on the inner-left sensors while keeping the outer sensors relatively stable. This was to avoid disrupting performance on other sections, such as the jump, where edge sensor readings were key.

Eventually, the car reliably followed only the left path, and reached the turning point successfully.

On the return path, we realized that we would need to include a separate constant weight array to reverse the bias on the way back and favor the right-side sensors. Since the path layout was now mirrored on the way back from the car's perspective, we needed to flip the logic in the code. We repeated the procedure of slightly incrementing and testing, ensuring it would follow the path properly while maintaining stability. While this logic successfully followed the split, it was still slightly off during the curve on the arch, adjusting too late. Despite attempting to increment one side specifically, the results were the same. We attempted to adjust the two middle sensors, allowing the car to react better to the line. After some adjustments, this change resulted in a successful navigation of the arch in both directions.

Question 3: Analyze Your Data

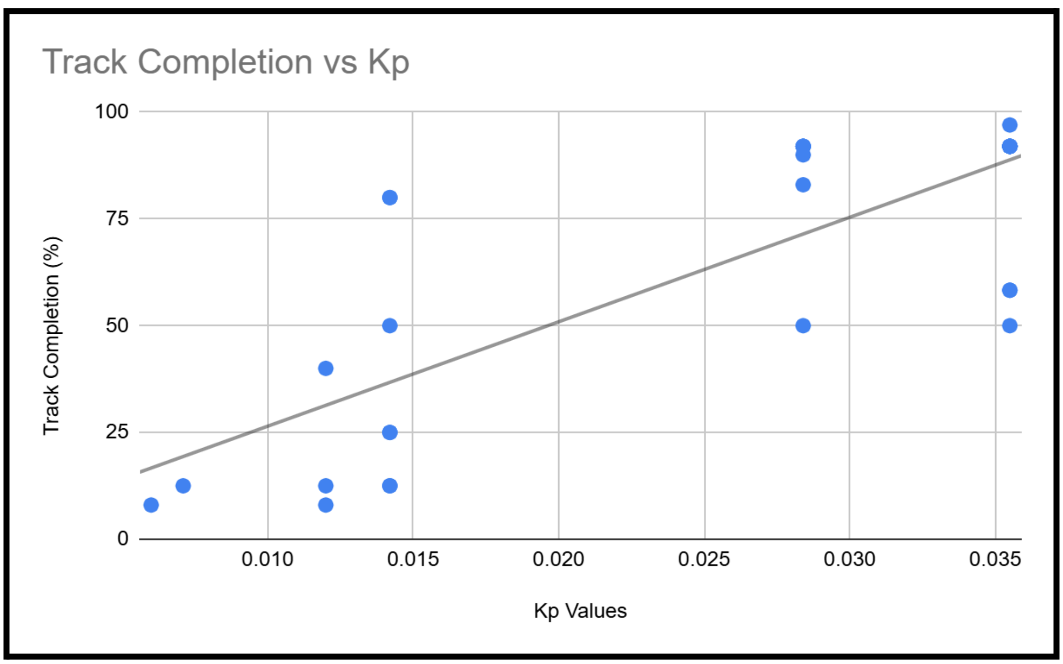

Figure 1: *Track completion percentage vs. Kp*. Higher Kp values generally improved track completion, suggesting better responsiveness and correction accuracy during turns.

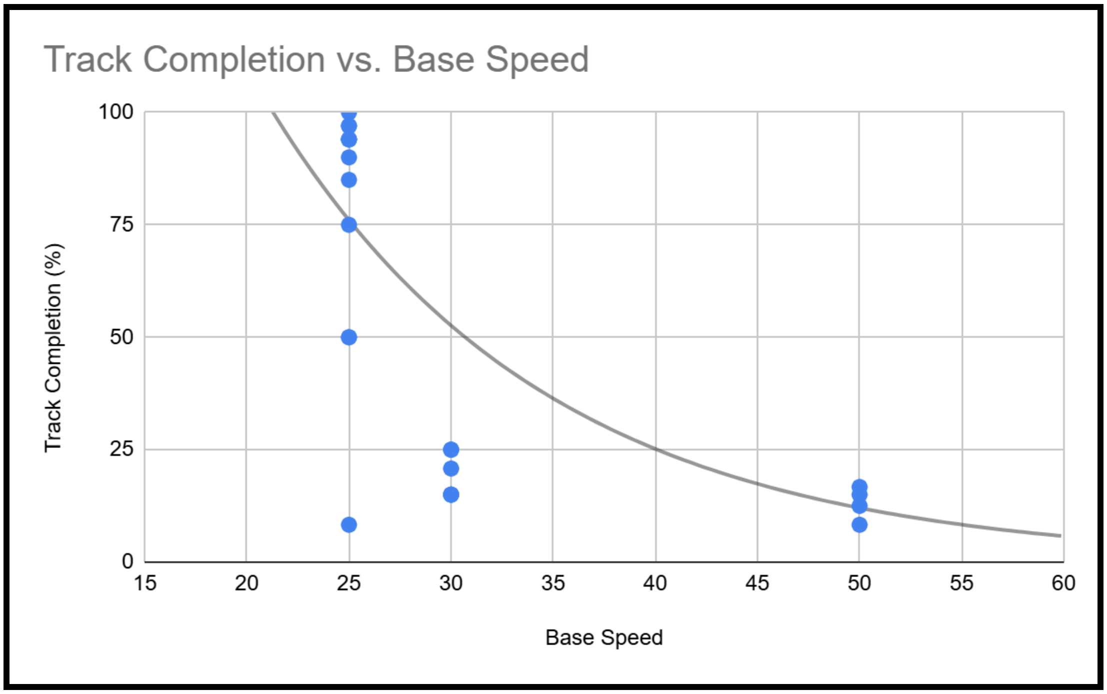

Figure 2: *Track completion percentage vs. base motor speed.* As base speed increases, the percentage of the track completed generally decreases with an inverse relationship.

Question 4: Interpret Your Analyses

Figure 1 illustrates how the various Kp values affected track completion percentage across different trial runs. Initially, we started with very small Kp values to test like 0.012, and it resulted in minimal success, with most track completion rates mostly being below 50%. The car struggled to follow turns accurately, with delayed and weak corrections that caused the car to drift off the path. However, as we increased Kp, the car responded more assertively to derivations from the path, and subsequently the average track completion rates improved to 50-75% overall. The steep rise in performance from 0.015 to 0.030 suggests that Kp is a key factor in correcting errors faster. Our final Kp value of 0.0355 resulted in the most consistent success, putting it at a track completion percentage of around 97%. This indicates that proportional control was critical for accurate navigation through curves and turns. With this said, the part of the track that still gave us trouble was the arch, specifically on the way to the turnaround. This suggests that although Kp was crucial to drastically improving the car's control on the path, additional adjustments such as refining Kd and the sensor weighting are necessary to handle sharper transitions of the track like the arch.

Figure 2 shows an inverse relationship between base motor speed and track completion percentage. At lower base speeds such as 25 and 30, the robot consistently completed most or all of the track, suggesting that slower speeds gave the car more time to process sensor input and make precise corrections. Slower base speeds made sharper turns appear much smoother during test runs. On the other hand, as base speeds increase to above 30, the track completion rates decline exponentially. At a base speed of 50, the trends show that performance was significantly worse as most tests completed less than 25% of the track. This trend exposes a key limitation, in that higher speeds amplify the effects of error and reduce the car's ability to compensate in time. Obstacles that required precise timing, such as the jump, were much less reliable. The car's momentum at higher speeds often causes the car to overshoot turns and subtle curves in the path, especially in complex regions like the split and arch. These results show the importance of balancing speed and car stability. For our final configuration, we used a base speed of 25, as we felt it had produced the best test runs. Of course, the results for testing base speeds were affected by the development of our code logic at the time. Still, it was clear that slowing the car down allowed us to highlight the exact navigation and decision-making of the car better, while also ensuring higher consistency.

Question 5: Appendix

9 May 2025

	Speed	Kp	KD	Start pos	Batt. volt	(5 = most 1 = least) Weaving ~~batt~~	WTS	Result
1	70	0.012	NA	0	7.52	NA	15 14 12 8	Kept going forward - off track at mark 1
2	50	0.012	NA	0	7.52	NA	15 14 12 8	moved forward cont. off track at mark 1
3	25	0.006	NA	1	7.52	NA	15 14 12 8	wheels changed speed, no longer at mark 1 just forward
4	50	0.012	NA	1	7.52	5	15 14 12 8	follows path, still goes off track marked but changes wheel speed

16 May 2025

	Speed	Kp	KD	Start pos	Batt. volt	Weaving	WTS	Result
5	50	0.012	0.06	2	7.52	2	15 14 12 8	completed curved path, continued after end mark 6
6	50	0.012	0.06	2	7.52	2	15 14 12 8	↑ failed before mark 2, on the "esses"
7	30	0.0071	0.0355	2	7.52	2		= 6 result
8 = 7								
9 = 8								
10	30	0.0142	0.071	2	7.52	2	15 14 12 8	fails halfway through "esses" piece (before mark 2)
11	30	0.0142	0.071	2	7.52	2	15 14 12 8	didn't do split, went in middle of it, but completed the rest. (mark 3 fail)
12	30	0.0142	0.071	1	7.52	2	15 14 12 8	went through middle of split, but works on arch, completes the rest. (mark 3 failure)

May 23, 2025

Speed	kp	kd	Start pos	Battery Volt ~~Weaving~~	1-least 5-most Weaving	Wts	Result
(1) 25	0.0142	0.071	2	7.82	2	15 14 12 8	took right path on both arch ; split first time, did not turn around at end. (mark 6)
(2) = (1)						20 18 16 10 8 12 14 15	spun around at "esses" weights may have affected it (before mark 2)
(3) = (2)					3	18 16 14 9 8 12 14 15	completed track (still didn't completely do arch correctly, mark 4)
(4) = (2)						18 16 14 9 8 12 14 15	
(5) = (4)							
(6) = (5)							
(7) = (3)							completed track (~~?~~ arch unsuccesful, mark 4)
(8)	0.0284	0.071	2	7.82	2	20 18 16 12 5 8 10 15	completed track, more noticeable turn on arch (mark 4)

May 30, 2025

Speed	kp	kd	Start pos	Battery Volt	Weaving	Wts	Result
(9) 25 ~~100~~	0.0284	0.071	2	7.65	2	15 10 6 3 14 18 20 21	on the return didn't follow track (reached mark 6)
(10) 25	0.0284	0.071	2	7.65	2	15 10 8 5 12 16 18 20	Completed track, doesn't react quick enough on arch (mark 5)
(11) 25	0.0284	0.071	2	7.65	2	15 10 8 8 12 16 18 20	completed track, successful arch only on way back. (4)
(12) 25	0.0284	0.071	2	7.65	2	15 10 8 10 12 16 18 20	completed track, successful only on way back of arch, didn't stop at the end. (missed mark 12)
(13) = (12)						Turn before: 15 10 8 10 13 16 18 20 After: 15 10 8 10 12 16 18 20	
(14) = (13)							
(15) 25	0.0284	0.071	2	7.65	2	Before turn: 15 10 8 10 13 16 18 20 After turn 20 18 16 12 10 8 10 15	completed track, stopped at end. worked only on way back of arch

so at mark 4 unsuccessful

2

May 30, 2025

	Speed	Kp	Kd	Start pos	Battery Volt	Weaving (1 = least, 5 = most)	wts	Results
16	25	0.0284	0.071	#2	7.65	1	Before turn: 15 10 8 11 14 16 18 20 After turn: 20 18 16 12 10 8 10 15	completes track, still only works on way back (arch) ([illegible]) (marks [illegible])
17	25	0.0355	0.071	#2	7.65	2	Before turn: 15 10 8 11 14 16 18 20 After turn: 20 18 16 14 ~~14~~ 8 10 15	only works on way back on arch (mark 4 unsuccessful) 11/12 marks total

May 31

	Speed	Kp	Kd	Start pos	Battery Volt	Weaving	wts	Results
18	25	0.0355	0.071	#2	7.67	3	~~before~~ 15 10 8 12 14 Same ~~[illegible]~~ 16 18 20 uses for before & after	went off jump on way back (before mark 8)
19 = 18						2	~~[illegible]~~ 15 10 8 11 14 16 18 20	~~[illegible]~~
20	25	0.0355	0.071	#2	7.67	2	15 10 8 11 14 16 18 20	Completed track arch only on way back (unsuccessful on mark 4) but reached mark 12
21	25	0.0355	0.071	#2	7.67	2	15 10 8 11 14 16 18 20	Completed track arch only on way back (11/12 marks)
22	25	0.0355	0.071	#2	7.67	2	15 10 8 11 14 16 18 20	Stopped at jump (before mark 8)
23	25	0.0355	0.071	#2	7.67	1	15 10 8 11 14 16 18 20	completed track, arch only on way back (reached mark 12)
24	25	0.0355	0.071	#2	7.67	2	15 10 8 10 14 16 18 20	turned around at split (3/12 marks)
25	25	0.0355	0.071	#2	7.67	2	15 10 8 10 14 16 18 20	Completed track, arch only on way back (11/12 marks)