Answer sheet Processing lab 1

Instructions: Fill out your answers below. Make a PDF of the complete file, and upload that **PDF** on Blackboard.

Student 1 (Full name & student number): Lena Feiler 1751611

Student 2 (Full name & student number): As discussed via Email, my partner has decided to drop out of this course and I will therefore be submitting the assignment individually.

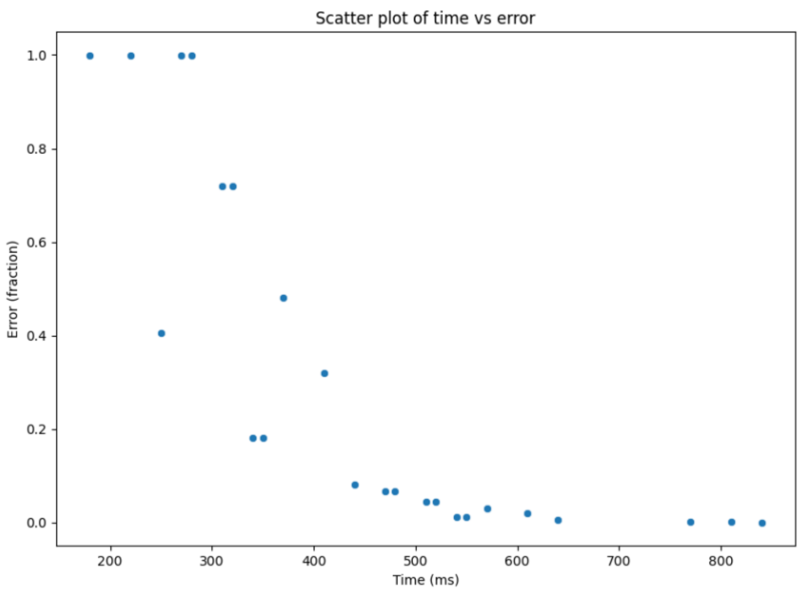
Blackboard question 1

1A: *For example 3: what is the predicted task time for the model that does all steps as fastman in milliseconds?*
Answer: 180 ms

1B: *For example 3: What is the predicted task time for a model that has the fastest perception, takes average time for cognitive steps, but is the slowest in motor execution?*
Answer: 340 ms

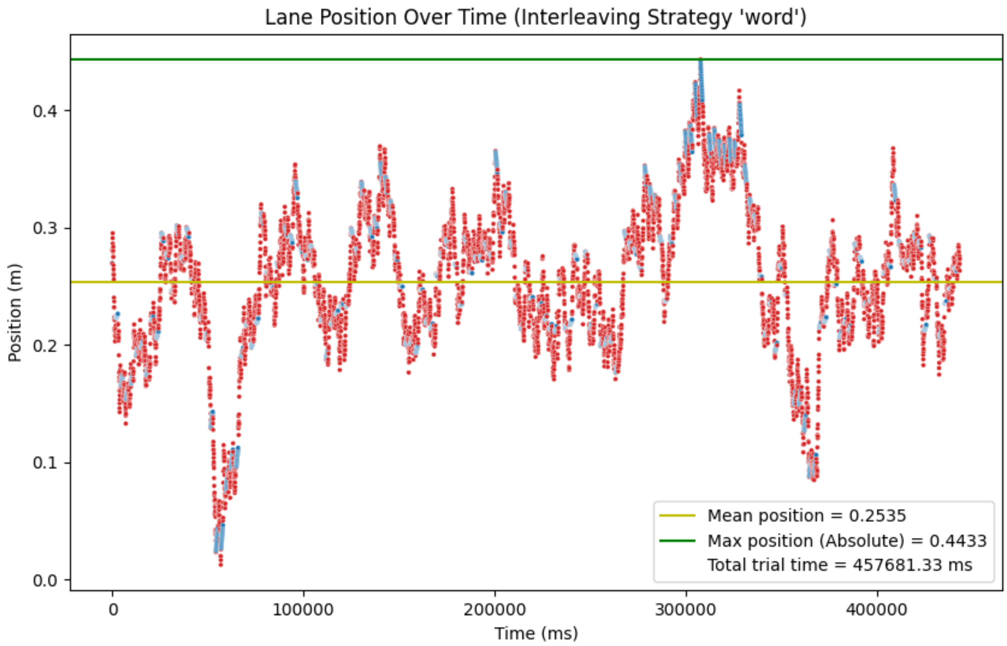
1C: *For example 4: What is the predicted slowest time that we might observe in this experiment?*
Answer: 220 ms

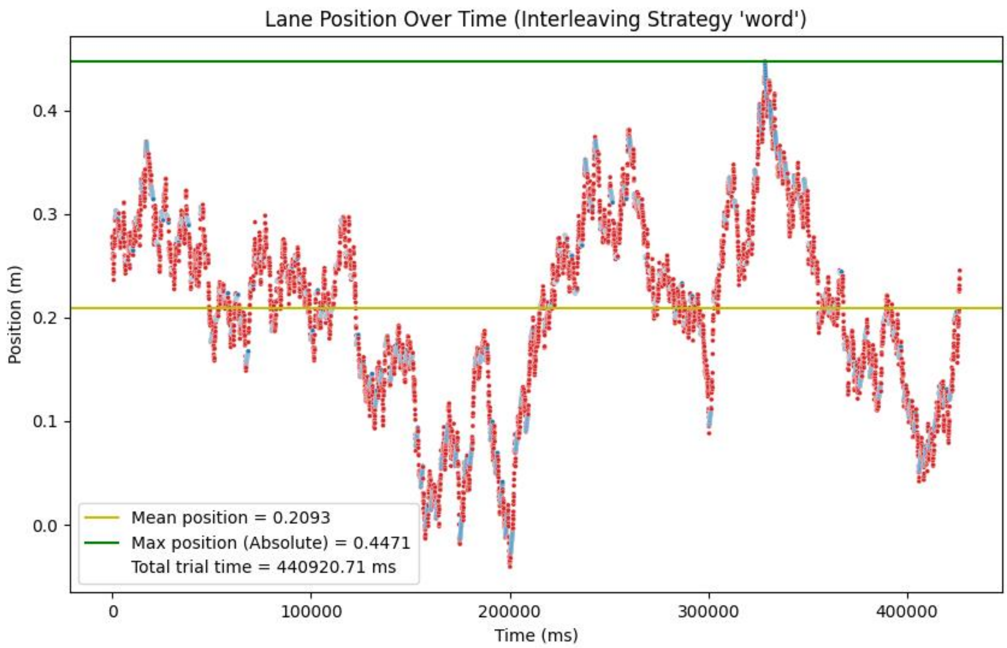
1D: *(paste a picture or screenshot below): For example 5: add a picture or screenshot of the scatterplot (as an image), Make sure to clearly label the x- and y-axis (give name and measurement unit, e.g. "time (ms)") and to use an appropriate range of values on each axis.*

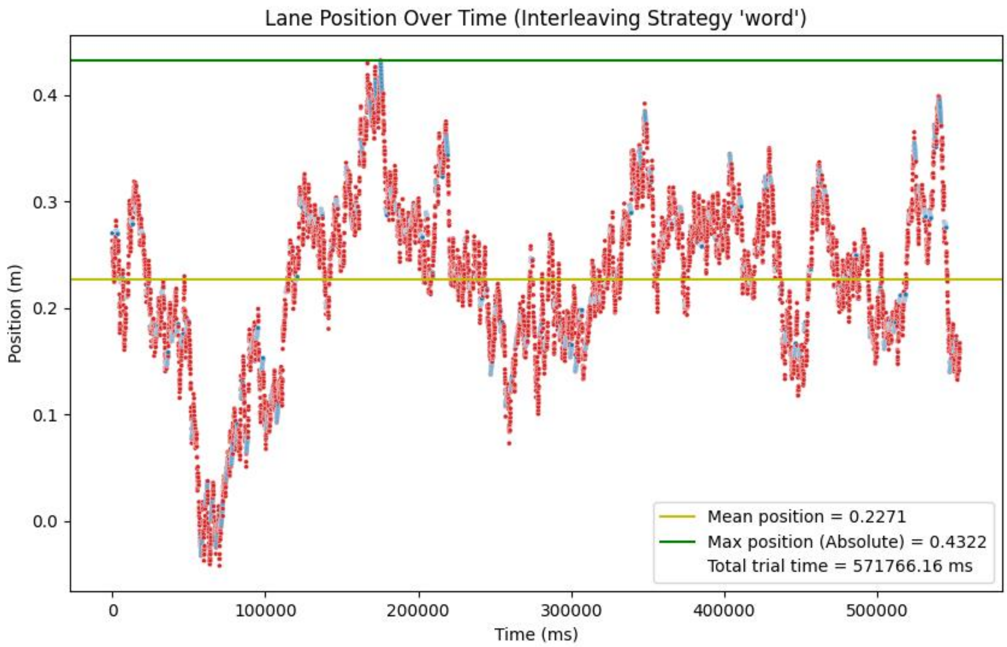


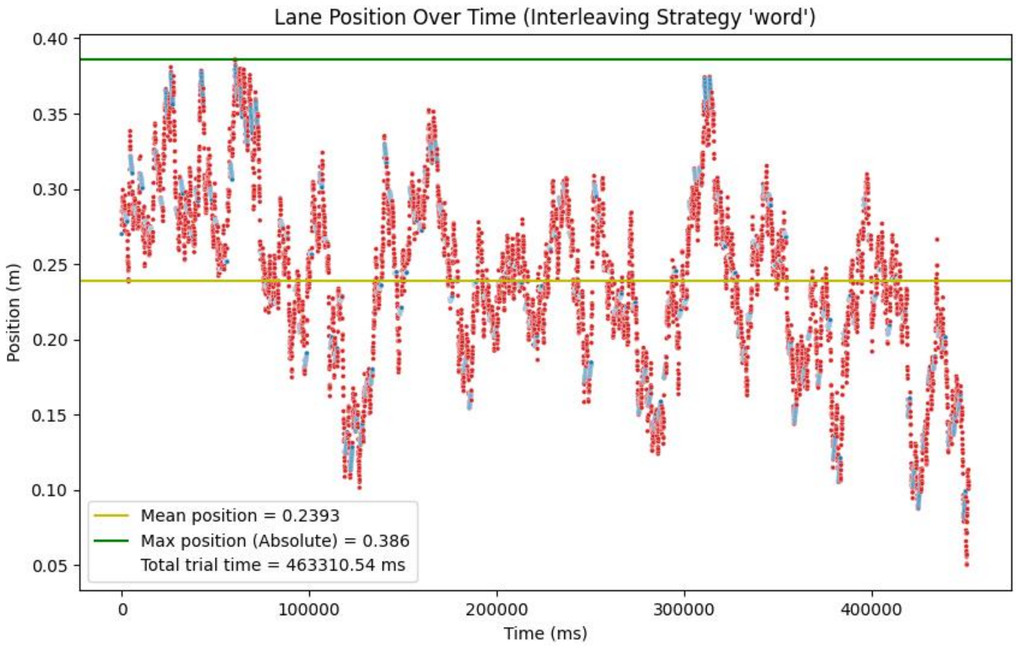
Blackboard question 2:

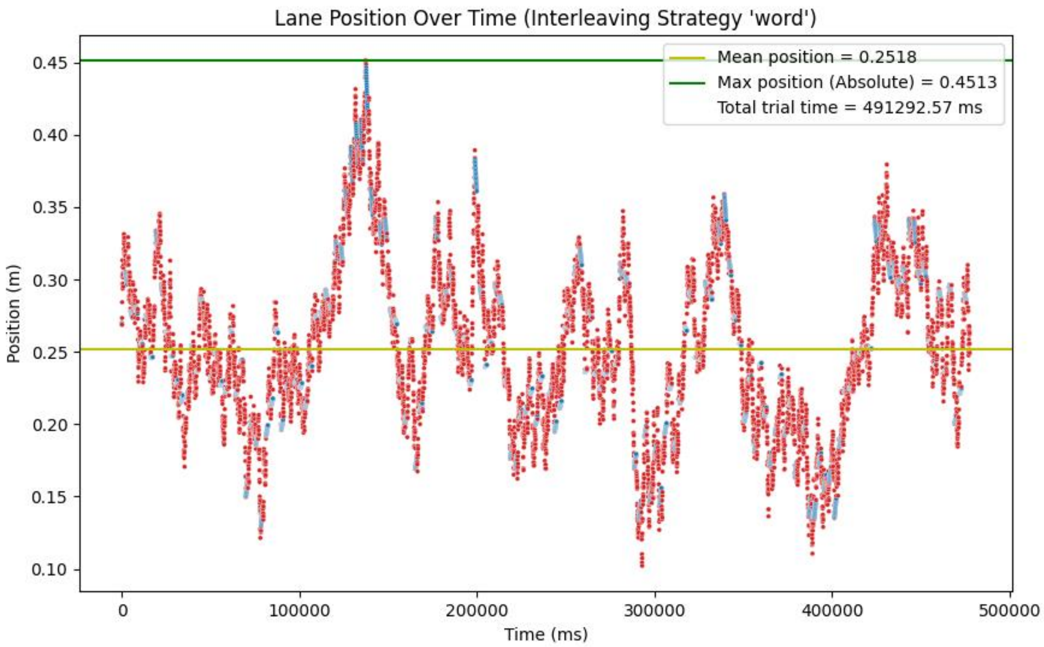
2A: Copy screenshots or pictures of 10 different plots below.

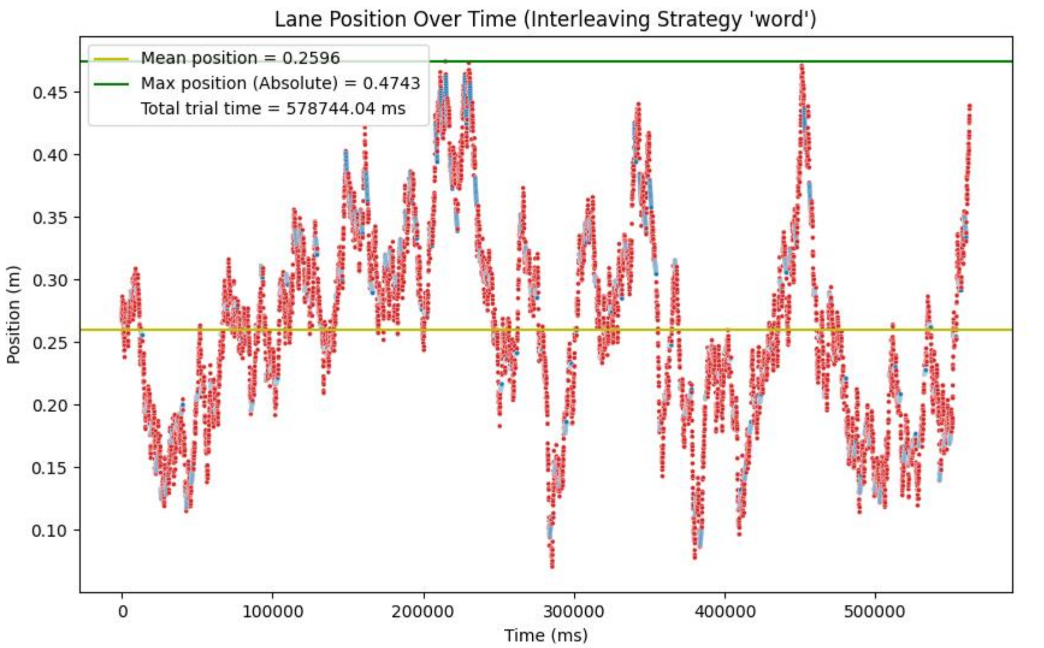


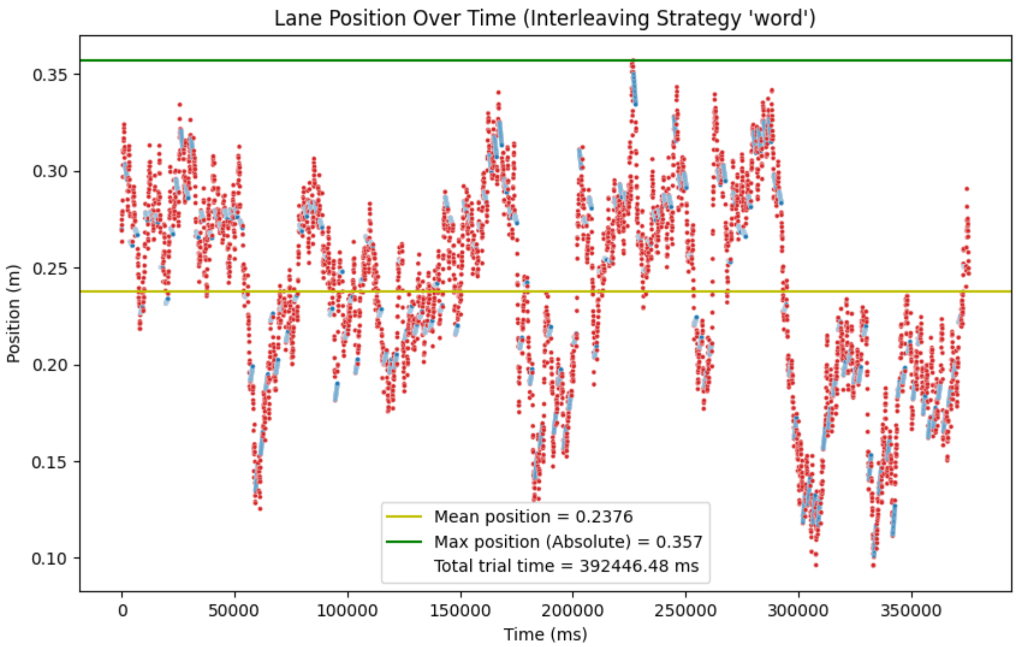


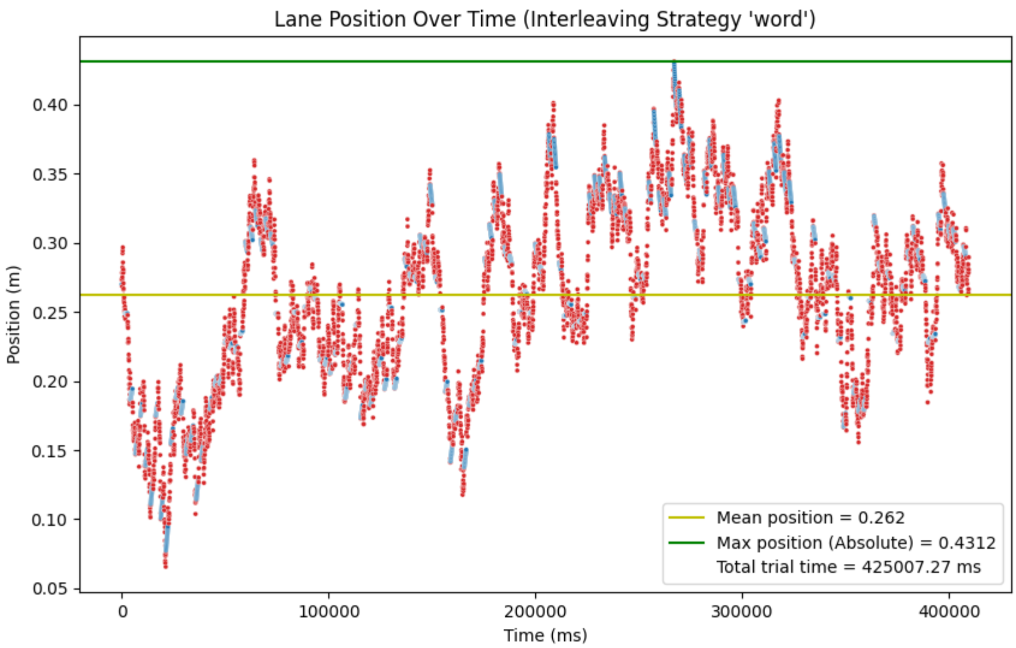

Lane Position Over Time (Interleaving Strategy 'word')
Position (m)
Time (ms)
Mean position = 0.2271
Max position (Absolute) = 0.4322
Total trial time = 571766.16 ms
0.4
0.3
0.2
0.1
0.0
0
100000
200000
300000
400000
500000

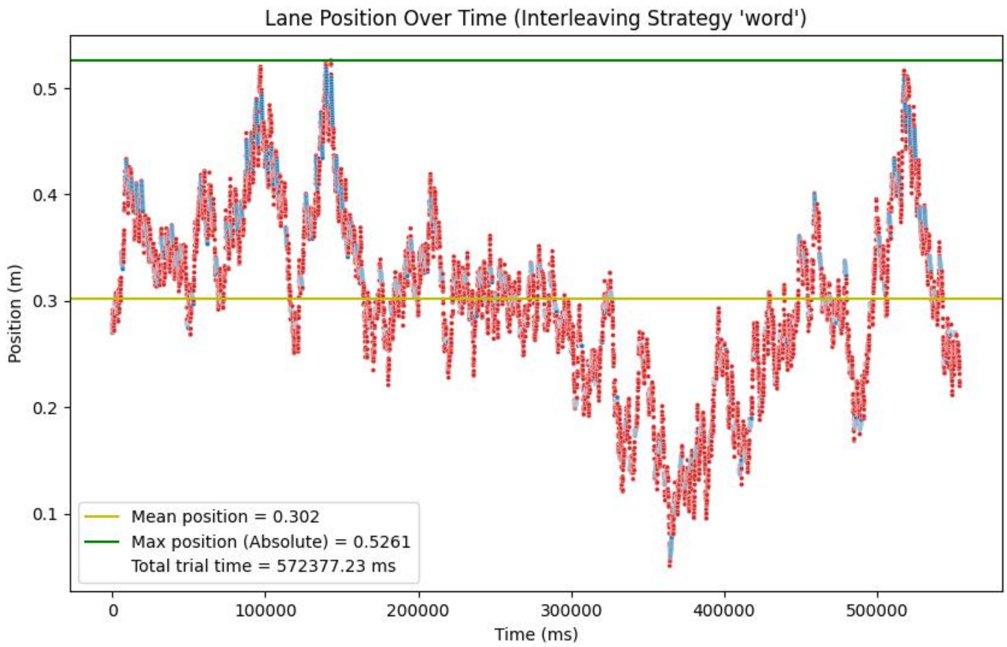

Lane Position Over Time (Interleaving Strategy 'word')
Position (m)
Time (ms)
Mean position = 0.2393
Max position (Absolute) = 0.386
Total trial time = 463310.54 ms
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0
100000
200000
300000
400000

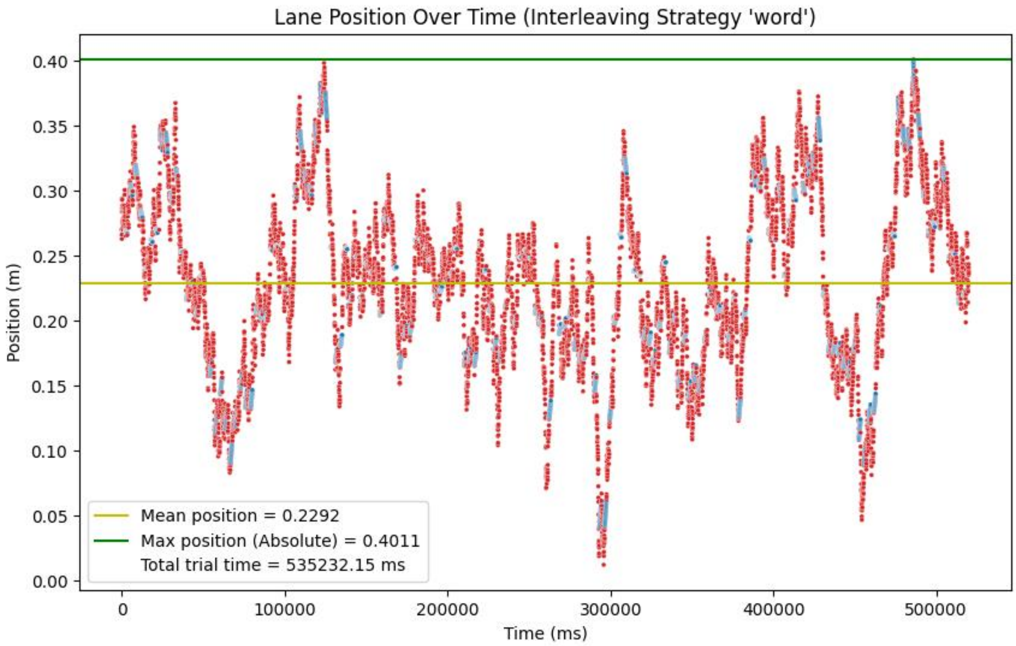

Lane Position Over Time (Interleaving Strategy 'word')
Mean position = 0.2518
Max position (Absolute) = 0.4513
Total trial time = 491292.57 ms
Position (m)
Time (ms)
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0
100000
200000
300000
400000
500000


Lane Position Over Time (Interleaving Strategy 'word')
Mean position = 0.2596
Max position (Absolute) = 0.4743
Total trial time = 578744.04 ms
Position (m)
Time (ms)
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0
100000
200000
300000
400000
500000

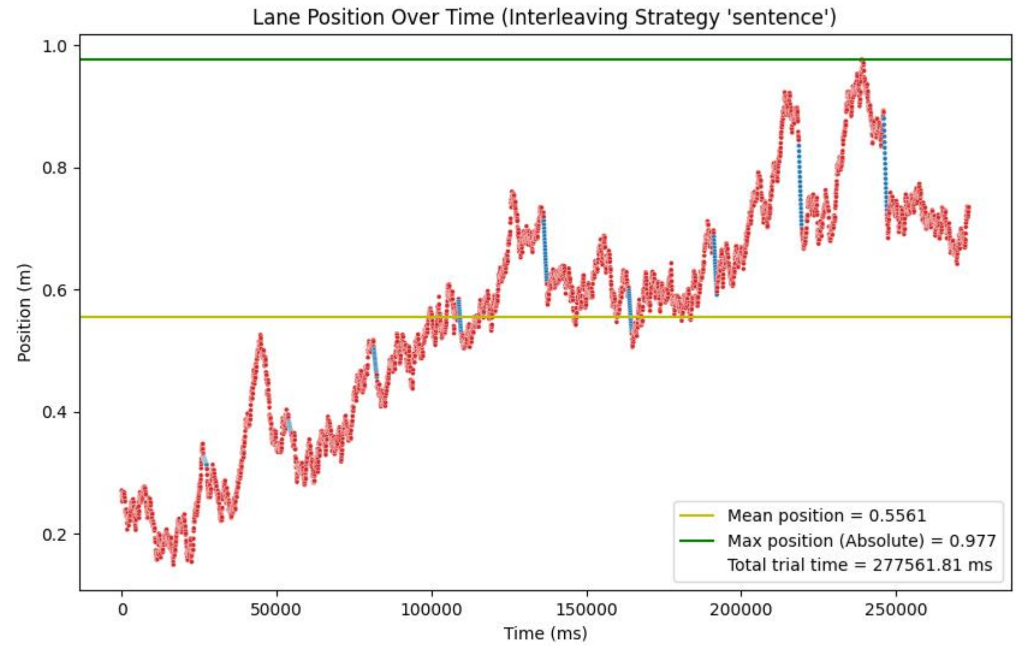

Lane Position Over Time (Interleaving Strategy 'word')
Position (m)
Time (ms)
Mean position = 0.2376
Max position (Absolute) = 0.357
Total trial time = 392446.48 ms

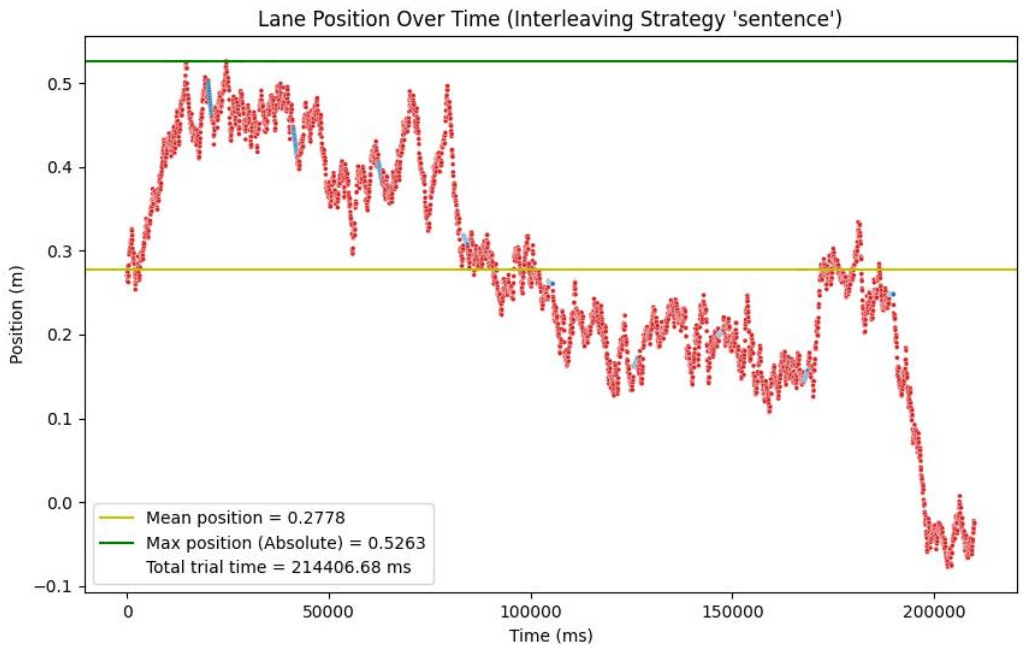

Lane Position Over Time (Interleaving Strategy 'word')
Position (m)
Time (ms)
Mean position = 0.262
Max position (Absolute) = 0.4312
Total trial time = 425007.27 ms

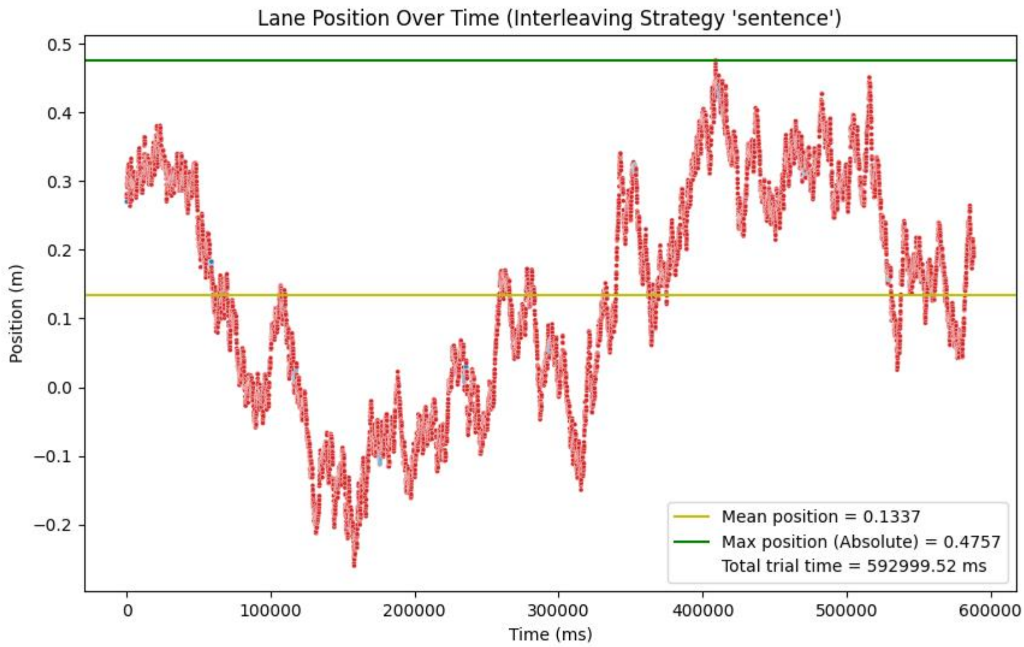

Lane Position Over Time (Interleaving Strategy 'word')
Position (m)
Time (ms)
Mean position = 0.302
Max position (Absolute) = 0.5261
Total trial time = 572377.23 ms
0.1
0.2
0.3
0.4
0.5
0
100000
200000
300000
400000
500000

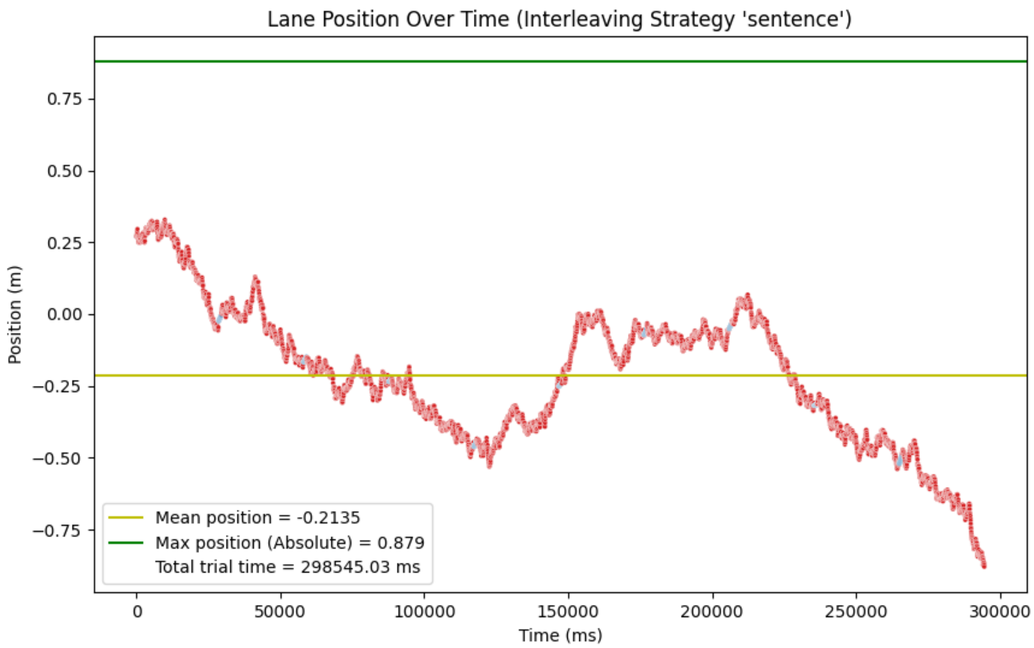

Lane Position Over Time (Interleaving Strategy 'word')
Position (m)
Time (ms)
Mean position = 0.2292
Max position (Absolute) = 0.4011
Total trial time = 535232.15 ms
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0
100000
200000
300000
400000
500000

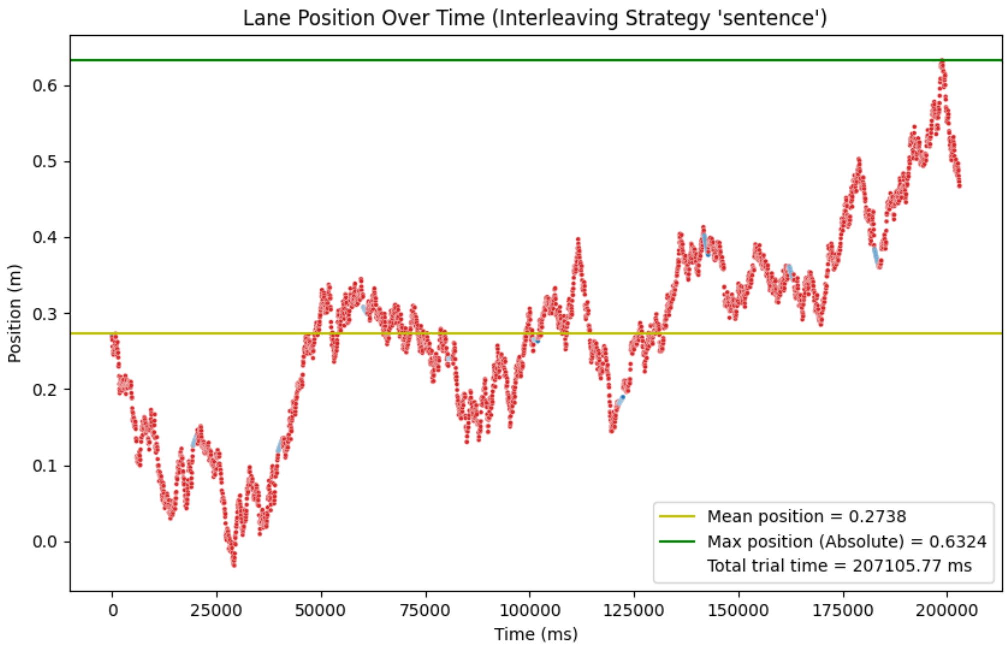
2B: Copy screenshots or pictures of 10 different plots below.

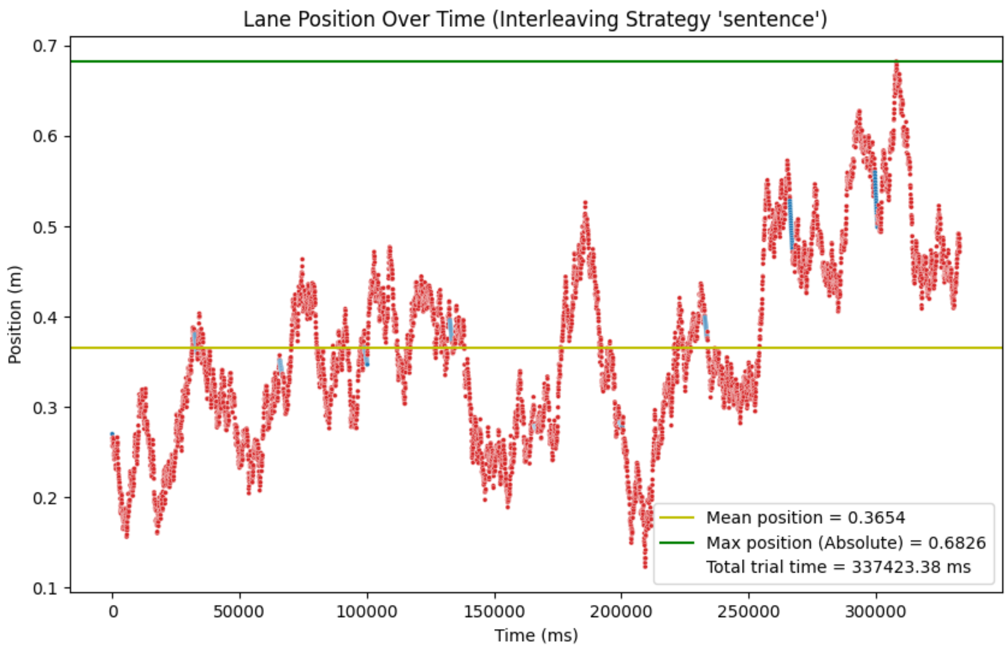


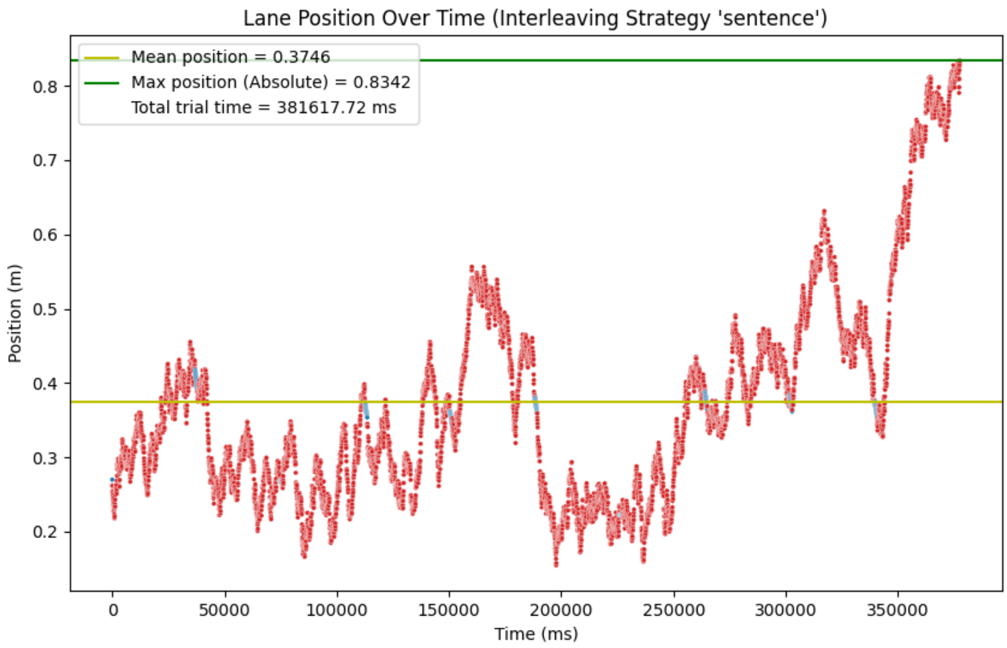


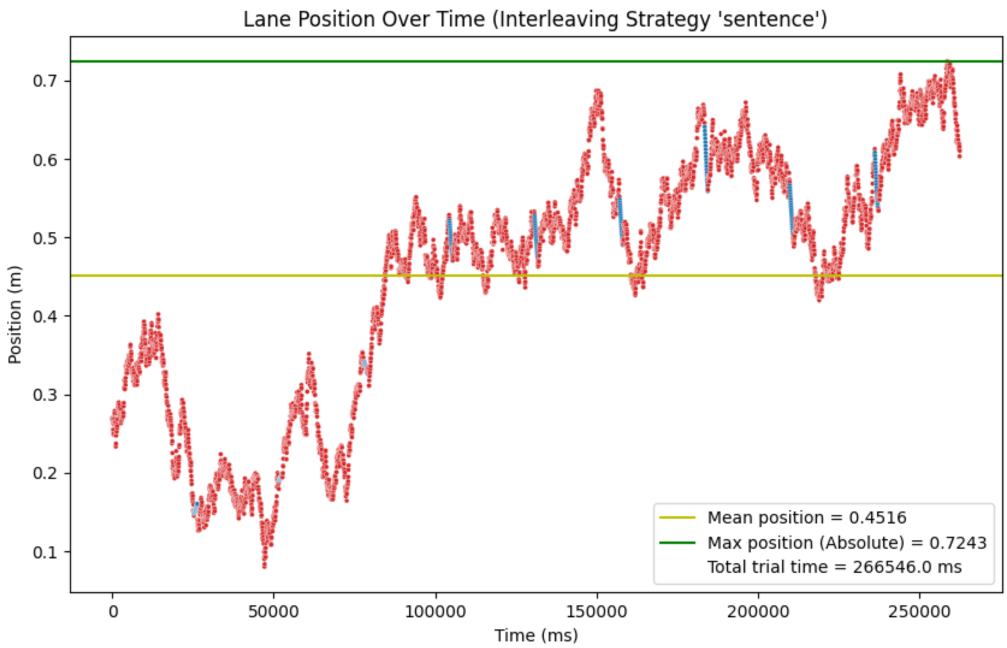

Lane Position Over Time (Interleaving Strategy 'sentence')
Position (m)
Time (ms)
Mean position = 0.1337
Max position (Absolute) = 0.4757
Total trial time = 592999.52 ms

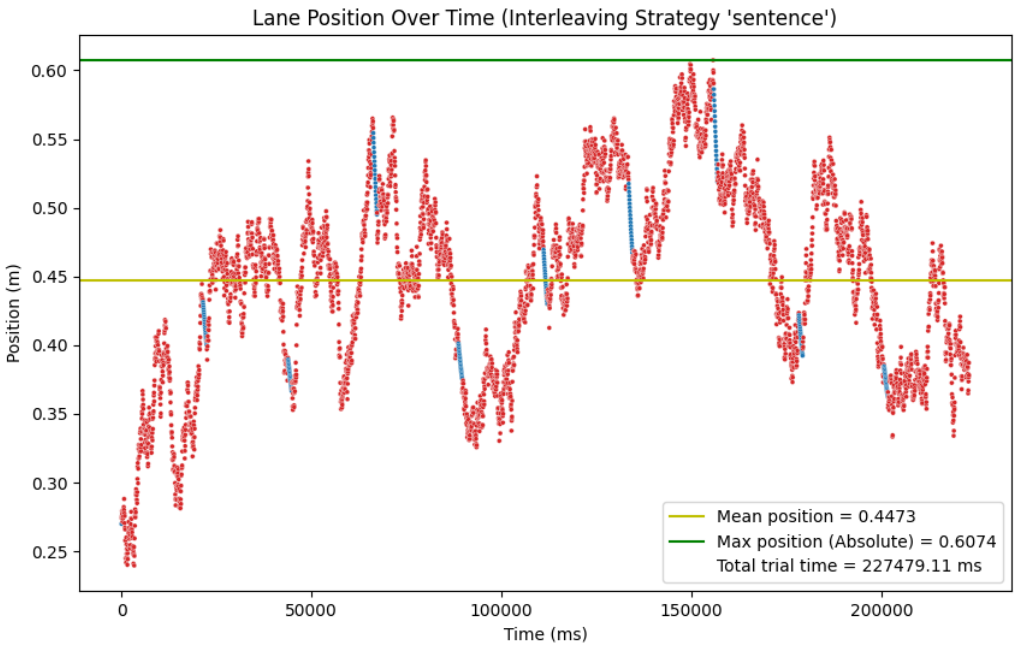

Lane Position Over Time (Interleaving Strategy 'sentence')
Position (m)
Time (ms)
Mean position = -0.2135
Max position (Absolute) = 0.879
Total trial time = 298545.03 ms

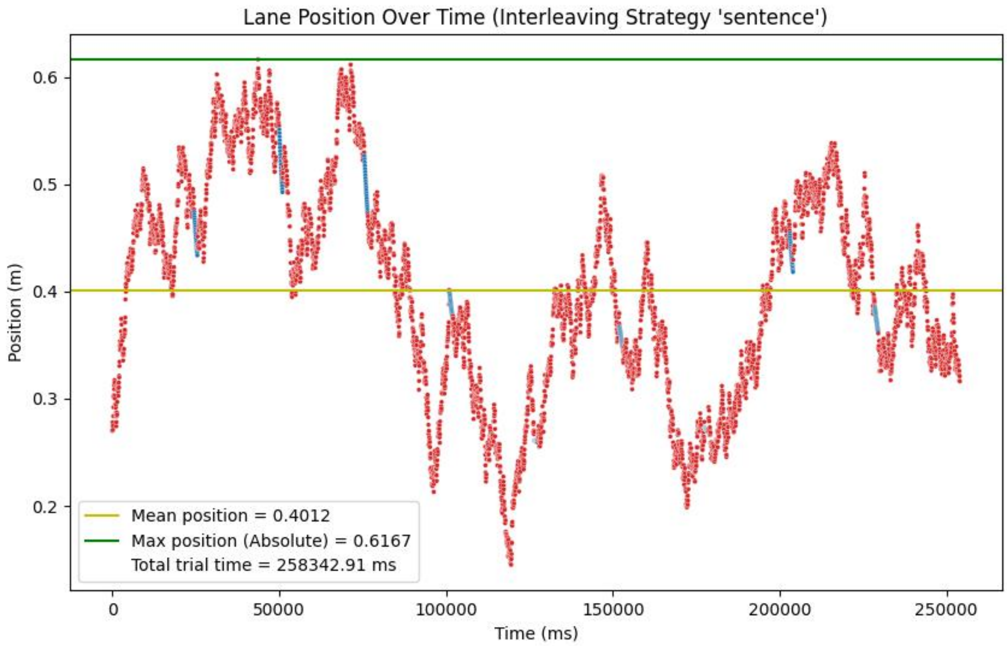

Lane Position Over Time (Interleaving Strategy 'sentence')
Position (m)
Time (ms)
Mean position = 0.2738
Max position (Absolute) = 0.6324
Total trial time = 207105.77 ms


Lane Position Over Time (Interleaving Strategy 'sentence')
Position (m)
Time (ms)
Mean position = 0.3654
Max position (Absolute) = 0.6826
Total trial time = 337423.38 ms


Lane Position Over Time (Interleaving Strategy 'sentence')
Mean position = 0.3746
Max position (Absolute) = 0.8342
Total trial time = 381617.72 ms
Position (m)
Time (ms)


Lane Position Over Time (Interleaving Strategy 'sentence')
Mean position = 0.4516
Max position (Absolute) = 0.7243
Total trial time = 266546.0 ms
Position (m)
Time (ms)





2C. Answer:

(i) The main difference between the two strategies is that the driver corrects the car much more frequently when steering after every word, as opposed to after every sentence. In the majority of the plots, this difference is clearly visible: the "sentence" strategy shows greater deviation from the starting line over longer periods of time. This occurs because, with the "word" strategy, the car is corrected

more often. Even with stronger drifts, these corrections tend to redirect the car back to the starting position, resulting in a characteristic wave-like pattern.
In contrast, the "sentence" strategy involves less frequent active steering, which reduces the impact of corrections on the car's path. As a result, this strategy has higher maximum deviations—up to 0.99 meters—whereas the "word" strategy typically results in maximum deviations of less than half a meter (0.3-0.5m). Furthermore, the mean position in the lane shows this larger car motion. For the word strategy model this average deviation remains close to the starting line (with values varying between 0.2 and 0.3), while those of the sentence model vary much more (mean positions from -0.2 to 0.55).
When looking at the different plots, as previously said, the word strategy tends to follow a wave-like pattern, waving left and right in the lane while remaining somewhat close to the starting line. The sentence strategy on the other hand, tends to both wave back and forth but also sometimes follow a certain direction. This pattern is largely dependent on the random drift data, since the driver is not able to correct the motion much.
However, it is important to note that the patterns observed in the plots depend heavily on the randomness inherent in each run, particularly in the drifting behavior. Therefore, both strategies can produce runs with varying degrees of performance/ deviation.

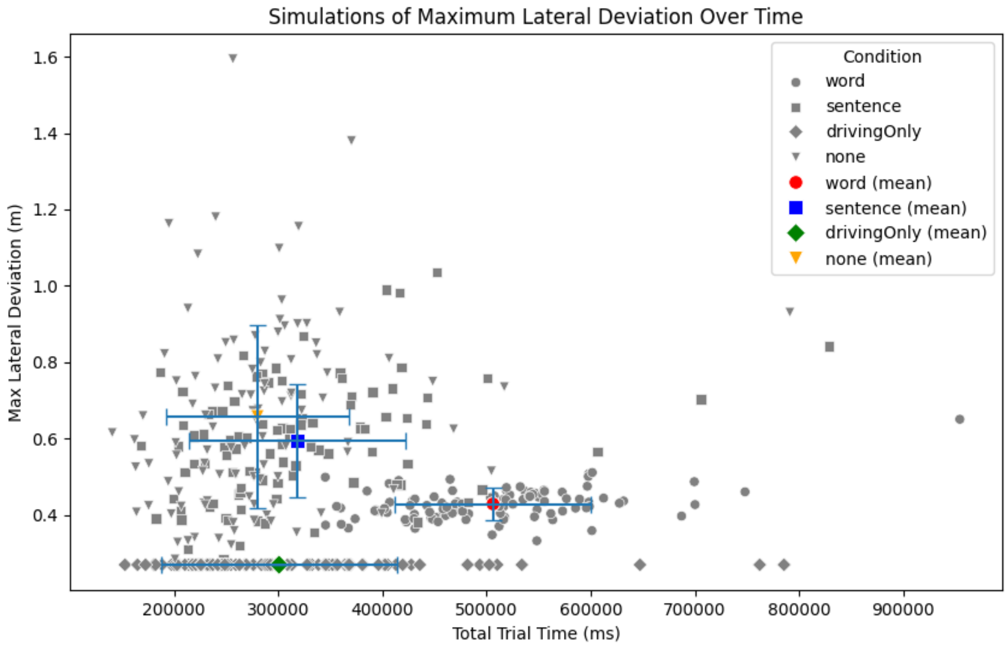
(ii) For each strategy, while the parameters, and thus the number of words or sentences that the driver needs to type, remain the same, there is a few causes for variability between different runs in our current model:

- **Time per word:** In each independent run the time required to type a word is determined by sampling a typing speed (in words per minute) from a normal distribution. This variability impacts both the total time and the amount that the car drifts while typing a word or sentence. If the typing speed is larger, the car has more time to deviate from its previous position, while the adjustment time (active steering) remains the same. Hence, if the time per word is larger, it is likely that the plot will show stronger deviations compared to runs with faster typing speed.
- **Drifting (passive/ non-active steering):** The drifting motion of the car, which occurs between active steering corrections, is also sampled from a normal distribution. Thus, both the magnitude and direction of the drift vary each run, introducing further variability in the final results.

This randomness in our model is the cause of the differences in each run. The combination of variability in typing speed (time per word) and random drifting between words leads to differences in the car's movement across runs. The typing speed affects the time available for drifting, while the random drifting introduces additional variability.

Blackboard question 3:

3A: Copy screenshot or picture of your plot below.



3B. Answer:

From the plot in 3A, one can observe that the model that interleaves after every word has a lower average maximum lateral deviation (0.4m compared to 0.6), as well as a smaller standard deviation of the maximum lateral deviation. This is likely because the driver actively steers more often in the word strategy, giving them the opportunity to correct the car's direction more quickly after each word. By steering more often, the car stays closer to the center of the lane, reducing the chance of drifting too far off course.

The extra time spent retrieving a word and steering between typing each word adds up, explaining the larger total time when compared to the sentence strategy. Since it takes additional time to steer between typing each word, this is the time difference when comparing it to the sentence model. However, the standard deviation of total time is similar for both models since the variation in duration largely depends on the time spent typing each word, which is sampled from the same normal distribution for both models.

Due to the smaller average lateral deviation, the word-strategy model would be the recommended approach when writing an email while driving. Altough it might require more time, it would increase safety by reducing the likelihood of a significant deviation from the lane which could cause the car to either drive off the road or crash into an object or another vehicle, potentially harming the driver themself and others on the road.

3C. Answer:

The "switch cost" described in the assignment would impact the four strategies differently. To start, for "driving only" there would be no impact, as the driver never reaches for their phone and thus will never have to apply this switch cost.

Similarily for the "none" strategy this additional cost would have no large impact. For this strategy, the driver would only have to each for their phone once at the very beginning of the task. Hence, the switch cost would be added as a constant to each simulation run, increasing their duration. Essentially, the entire distribution of trial times would shift to the right by the amount of the switch cost, but there would be no changes in the fluctuations of individual trial durations.

In the "word" strategy, the switch cost would cause a larger disruption, since the driver would have to reach for the phone for every word. This frequent movement would also drastically increase the total duration required to complete the task. The added switch cost would introduce greater variability in the car motion since it increases the time for drifting. Therefore, the task would become slower and higher maximum deviations from the starting line would be possible.

For the sentence strategy, the situation is similar. The driver now has to reach for the phone before each sentence, overall much less frequent than in the word strategy, resulting in additional drifting time. Since this motion is less frequent it would result in a smaller increase in total duration compared to the previous model and thus introduce less additional variability in the car motions. Therefore, the interleaving word strategy will be most affected by this additional switch cost and “driving only” the least.

Bonus question:

If you complete a bonus question (optional), please answer the questions of the assignment below.

A. A model that interleaves after every couple of words (instead of after every word, or after every sentence). There are various ways to implement this (e.g., have a fixed number of words, or a variable number of words typed each time). Argue why you chose what you chose. (Difficulty estimate: medium)

1. What scenario did you model (interleaving after X words, or error making)

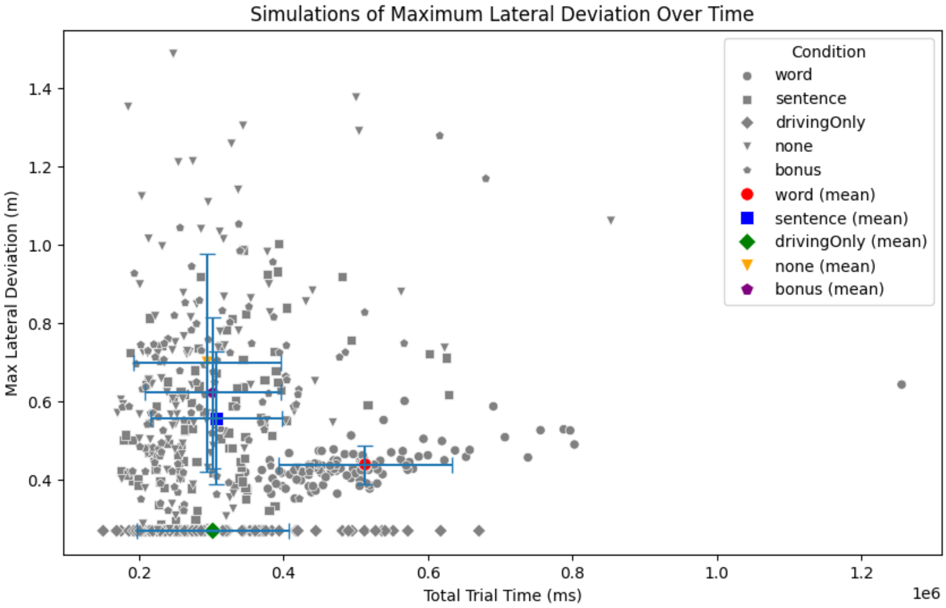
The “interleaving after X words” option was selected. In the implementation, a variable number of words, where the user can type up to 2 sentences at a time, is used (instead of a typeing a set amount of words each time), in an attempt to keep the scenario more realistic and (in my opinion) drivers are unlikely to ignore the car they are driving for very long periods at a time. For each word batch that is typed, the actual size is a random number between 1 and this maximum number (or the number of words left in the email when there is less than 2 sentences left to write). Furthermore, it is assumed that the driver will write at least one word each time. During the period of writing the car drifts and after each batch is finished the driver actively steers the car. The experimental parameters are kept the same as in exercise 2.

The model assumes that the driver has to think about what they type each time they begin a new batch. Thus, at the beginning of each batch either the word or sentence retrieval (if a new sentence is started) time is added to the trial time. Additionally, within one word batch, of a new sentence is started it also made sense that the driver would have to think about this. Therefore, the sentence retrival time is added to the total time whenever a new sentence is encountered within a word batch.

2. Submit a screenshot or PDF of the critical part of your code

```python
def runTrial(nrWordsPerSentence=17, nrSentences=10, nrSteeringMovementsWhenSteering=4, interleaving="word"):
    global timePerWord
    resetParameters()
    locPos = [startingPositionInLane]
    trialTime = 0
    locColor = ["blue"]

    # Sample words per minute from normal distribution
    s = max(0, np.random.normal(wordsPerMinuteMean, wordsPerMinuteSD))
    timePerWord = 1 / (s / 60) * 1000

    if interleaving == "bonus":
        numberOfSteeringDrifts = math.floor(
            nrSteeringMovementsWhenSteering * steeringUpdateTime / timeStepPerDriftUpdate)

        words_left = nrWordsPerSentence * nrSentences  # total words to process

        while words_left > 0:
            # determine number of words to read in current batch; assume we read at least one word everytime
            words_to_read = random.choice(range(1, min(words_left, nrWordsPerSentence * 2))) if words_left > 1 else 1

            trialTime += timePerWord * words_to_read

            # add retrieval time for sentence or word depending on or current position in sentence
            if words_left % nrWordsPerSentence == 0:
                trialTime += retrievalTimeSentence
            else:
                trialTime += retrievalTimeWord

            # add sentence retrieval time if a new sentence start is encountered in our current word batch
            for w in range(words_left - words_to_read, words_left):
                if w % nrWordsPerSentence == 0:
                    trialTime += retrievalTimeSentence

            # simulate the car drifts while typing
            drift_steps = math.floor(timePerWord * words_to_read / timeStepPerDriftUpdate)
            for _ in range(drift_steps):
                locPos.append((locPos[-1] + vehicleUpdateNotSteering() * 0.05))
                locColor.append("red")

            # actively steer after one batch of words
            for _ in range(numberOfSteeringDrifts):
                locPos.append(locPos[-1] + vehicleUpdateActiveSteering(locPos[-1]) * 0.05)
                locColor.append("blue")

            words_left = words_left - words_to_read  # update remaining word count in email
```

3. Submit a plot of what performance now looks like. The plot should be in the style of Blackboard question 3, but adjusted to the scenario at hand.



4. Explain how and why performance of the model has (not) changed compared to the strategies that you modelled before.

The performance needs to be analysed with the initial assumptions in mind. Since the number of words the driver can type at a time was limited to two sentences, and a random number between one and this number is selected, one can expect the average length of word batches to be similar to one sentence. This can also be seen in the plot as the average of the bonus startegy is closest to that of the interleaving sentence strategy.

It should be noted that if the used maximum batch size was selected differently, this average value would change, and with more freedom in batch size it could be expected to reduce the total time but also increase the maximum lateral deviation, since it is likely to encounter at least one large batch during which the car drifts. A more restricted version, on the other hand, where the user would only type small word batches, would be expected to have the opposite effect and become more similar to the word strategy.

In terms of maximum lateral deviance, the implemented model remains similar to the sentence model but with a slightly higher deviation. This makes sense since longer word batches are likely to result in higher deviations from the starting line. The trial average duration of both models is very similar. While the bonus strategy model does theoretically have at least the same trial time as the sentence model (the sum of all words being typed, the sentence retrieval times and additionally some word retrieval times) the averages shown in this plot are recorded over indpendent runs and hence show slight deviations based on the random sample drawn for the time per word.

Finally, it should be noted that different simulation runs will have different outcomes and the additional randomness in the interleaving bonus model will cause some deviation. Thus, in another simulation it could also happen that this model has a higher duration than the sentence model or also a smaller maximum deviation. We can however expect these values to remain close to the sentence model when we use our assumption of maximum word batch size.