Emerson Hall

Ramana M. Pidaparti

23 October 2025

Integration with Simulation

Introduction

The goal for Week 9 was to integrate the trained model into the UAV simulation and test how well it could perform real time trajectory optimization. The main objective was to see if the UAV could avoid both static and moving obstacles while still reaching the goal efficiently. This week built on the earlier path planning work by combining global A* planning, local path following, and dynamic obstacle avoidance into a single system. The process started with simple fixed environments and gradually moved to randomized worlds with moving obstacles. Four videos were created to show how the UAV handled each scenario, including cases where it succeeded and cases where it still struggled. In the final stage of testing, a wind model was introduced to evaluate how the UAV would perform under realistic environmental disturbances. The simulation included a steady background wind combined with mild random gusts that changed over time. The goal was to see if the UAV could still reach the target while maintaining a smooth and stable path. This addition made the system more realistic and tested its ability to adapt to unpredictable external forces.

Method

The simulation was tested in three main stages, each represented by a different version of the Week 9 code. Every version added improvements or adjustments that made the UAV more adaptive and consistent in how it moved through the environment. All the codes are shown in GitHub.

The first simulation used the file simulationW9firstsuccess.m. This test focused on getting the UAV to fly around basic static objects such as a blue wall and a gray box, along with one moving obstacle that shifted side to side. The control system used a finite state machine that switched between different behaviors like climbing, moving laterally, or descending, depending on where the UAV was relative to obstacles. Features like the "lip penalty" and "corner-pop" helped the UAV avoid skimming too close to walls and getting stuck near corners.
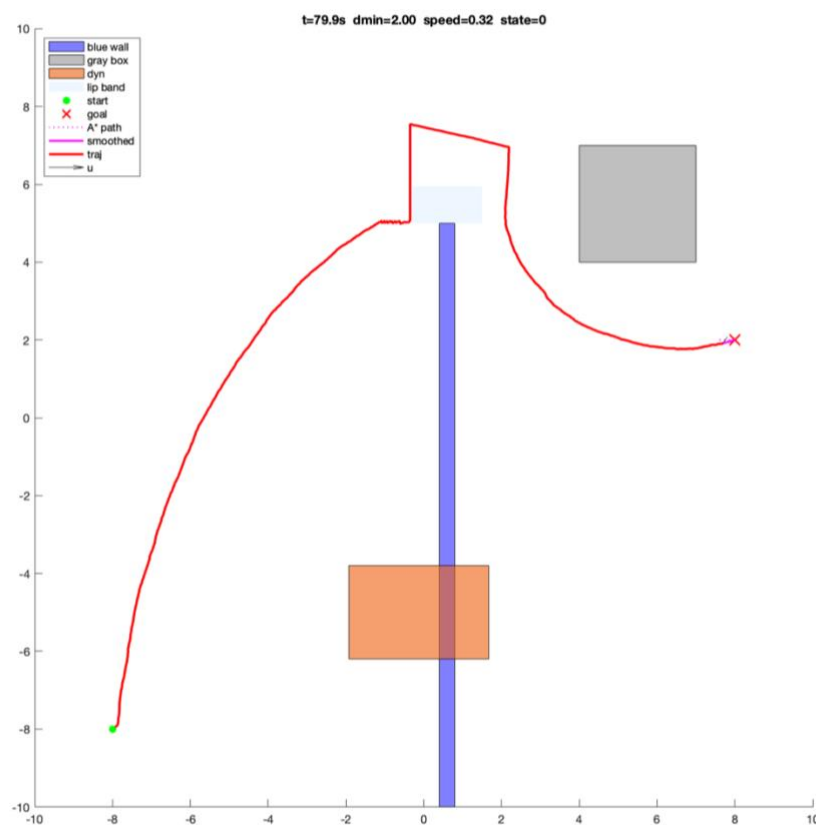

Figure 1: 1ˢᵗ Simulation Results

The second simulation, which used the file rand1stSimW9.m, introduced randomization into the environment. Each run generated new obstacle positions and sizes, making the test less predictable. While this version added a new unstick state to help the UAV recover when it got trapped, it did not perform consistently. In some runs, the UAV successfully avoided obstacles

and reached the goal, but in others, it became stuck on the side of objects or lost its path entirely. The random layouts made it much harder for the UAV to navigate smoothly, and sometimes it failed to recover from collisions. Even though it did not always work, this simulation helped reveal weaknesses in the control logic and path smoothing process that needed to be fixed in the final version.
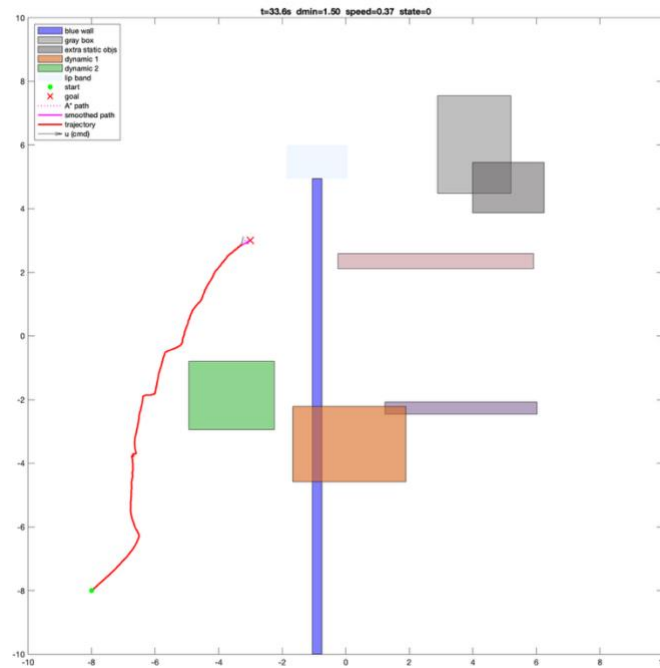

Figure 2: 2ⁿᵈ Simulation Results

The final and most successful version, W9successfulrandom.m, combined all of the key improvements from earlier tests but used a more stable and efficient control approach. Instead of relying only on discrete state changes the UAV used a real-time A* planner that updated every second to reflect the current environment. After each replan, the path was refined using a string-pull smoothing algorithm, which removed unnecessary waypoints while checking for clearance with a thick safety mask.
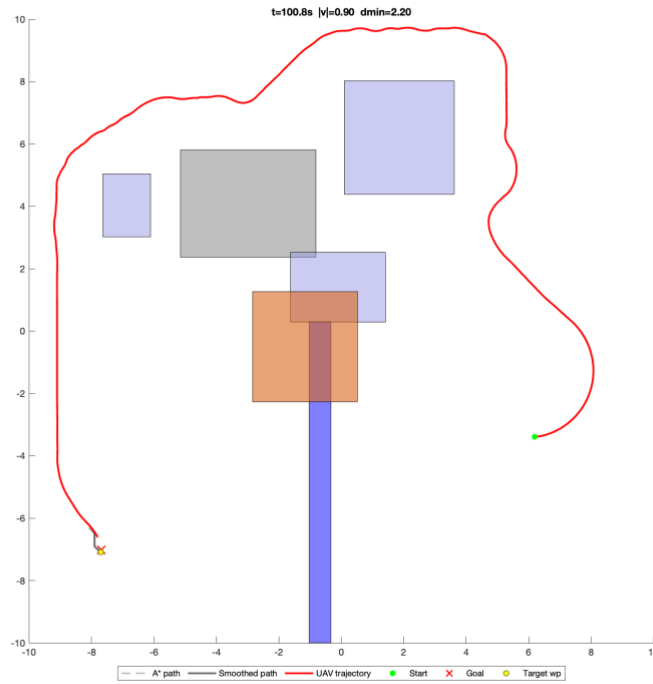
Figure 3: Final Success

For local control, the final version introduced a Frenet-frame path-following method. This method calculates the UAV's lateral error from the path and applies a correction to keep it centered, rather than steering directly at the next waypoint. This makes the motion smoother and more realistic. A repulsion vector derived from the gradient of the distance map was also added so that the UAV automatically veer away from obstacles without having to replan immediately. A heading rate limit prevented sudden turns, while an emergency braking system used short-range ray casts to slow the UAV when it approached close objects. Together, these features gave the UAV the ability to move fluidly around obstacles, even when the environment changed dynamically.

After completing the main simulations, wind dynamics were added to the final version to study how the UAV could compensate for external disturbances. The wind was modeled as a

constant vector with a random direction and magnitude, plus small gusts that varied smoothly over time. To handle this, the controller was modified so that the UAV computed both its desired ground velocity and the corresponding air velocity needed to achieve that motion despite the wind. The wind vector was subtracted from the desired ground velocity, effectively allowing the UAV to "crab" into the wind to stay on course. A minimum ground speed guard was also added to prevent the UAV from being stalled by strong headwinds, ensuring that it continued moving forward even when the wind opposed its motion. These adjustments allowed the UAV to remain stable, reach the goal, and maintain smooth movement even in windy conditions.
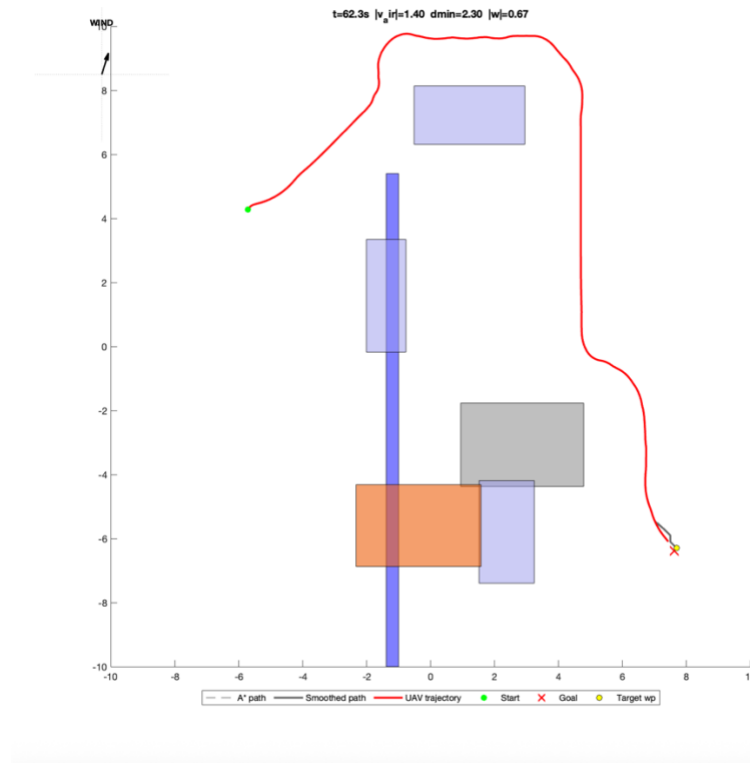

Figure 4: Final UAV with Wind

Each simulation frame was drawn and saved to a video file showing the UAV's progress in real time. The videos clearly showed improvements in stability, collision avoidance, and adaptability with each iteration of the system.

Results

   The results from this week showed clear progress, even though not every simulation worked perfectly. The first simulation performed well in simple scenes, with the UAV successfully flying around walls and a moving box. The second simulation with randomized environments was inconsistent. Sometimes the UAV navigated correctly, but other times it got stuck or could not find a valid path. Despite the mixed results, it helped identify the causes of instability, such as problems with path smoothing and obstacle sampling.

   The final version was the strongest and most reliable. The UAV reacted smoothly to both static and moving obstacles, adjusted its speed automatically, and reached the goal in most trials. It was able to adapt to random worlds without crashing, although there were still small issues with getting too close to walls or pausing briefly when navigating tight areas. The Frenet control and repulsion forces worked together to keep the UAV on track while still maintaining a realistic motion pattern.

Emerson Hall

Ramana M. Pidaparti

23 October 2025

<div align="center">Integration with Simulation</div>

Introduction

   The goal for Week 9 was to integrate the trained model into the UAV simulation and test how well it could perform real time trajectory optimization. The main objective was to see if the UAV could avoid both static and moving obstacles while still reaching the goal efficiently. This week built on the earlier path planning work by combining global A* planning, local path

following, and dynamic obstacle avoidance into a single system. The process started with simple

fixed environments and gradually moved to randomized worlds with moving obstacles. Four

videos were created to show how the UAV handled each scenario, including cases where it

succeeded and cases where it still struggled. In the final stage of testing, a wind model was

introduced to evaluate how the UAV would perform under realistic environmental disturbances.

The simulation included a steady background wind combined with mild random gusts that

changed over time. The goal was to see if the UAV could still reach the target while maintaining

a smooth and stable path. This addition made the system more realistic and tested its ability to

adapt to unpredictable external forces.

Method

The simulation was tested in three main stages, each represented by a different version of

the Week 9 code. Every version added improvements or adjustments that made the UAV more

adaptive and consistent in how it moved through the environment. All the codes are shown in

GitHub.

The first simulation used the file simulationW9firstsuccess.m. This test focused on

getting the UAV to fly around basic static objects such as a blue wall and a gray box, along with

one moving obstacle that shifted side to side. The control system used a finite state machine that

switched between different behaviors like climbing, moving laterally, or descending, depending

on where the UAV was relative to obstacles. Features like the "lip penalty" and "corner-pop"

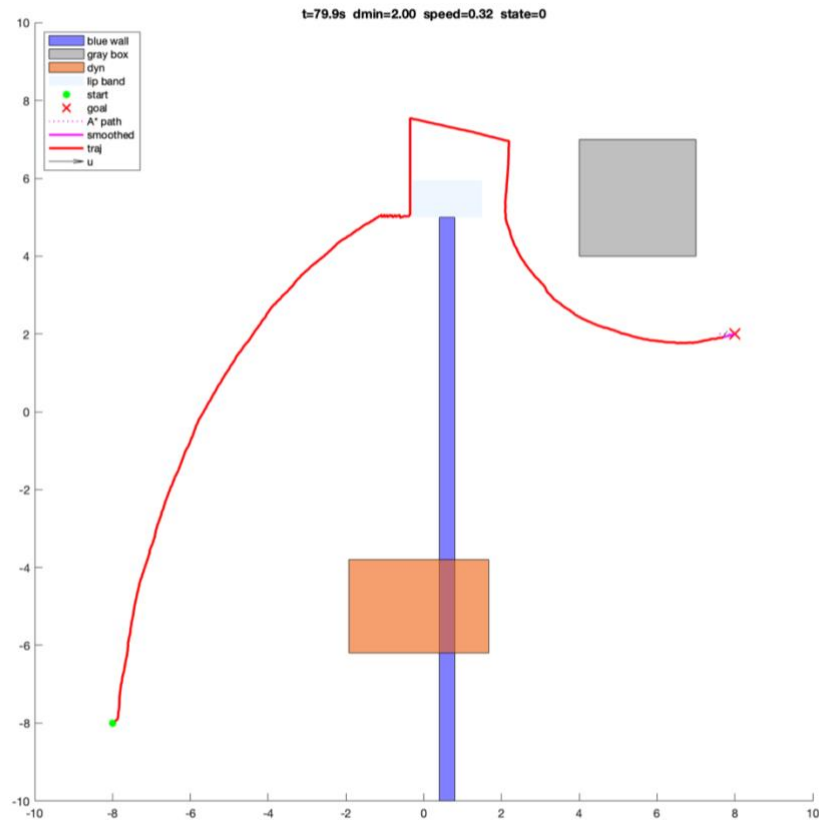helped the UAV avoid skimming too close to walls and getting stuck near corners.

Figure 1: 1ˢᵗ Simulation Results

The second simulation, which used the file rand1stSimW9.m, introduced randomization into the environment. Each run generated new obstacle positions and sizes, making the test less predictable. While this version added a new unstick state to help the UAV recover when it got trapped, it did not perform consistently. In some runs, the UAV successfully avoided obstacles and reached the goal, but in others, it became stuck on the side of objects or lost its path entirely. The random layouts made it much harder for the UAV to navigate smoothly, and sometimes it failed to recover from collisions. Even though it did not always work, this simulation helped reveal weaknesses in the control logic and path smoothing process that needed to be fixed in the final version.
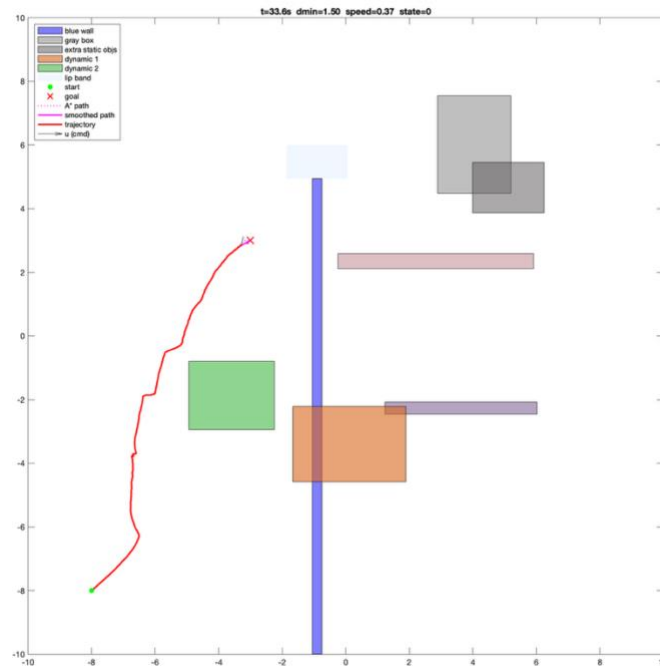
Figure 2: 2nd Simulation Results

The final and most successful version, W9successfulrandom.m, combined all of the key improvements from earlier tests but used a more stable and efficient control approach. Instead of relying only on discrete state changes the UAV used a real-time A* planner that updated every second to reflect the current environment. After each replan, the path was refined using a string-pull smoothing algorithm, which removed unnecessary waypoints while checking for clearance with a thick safety mask.
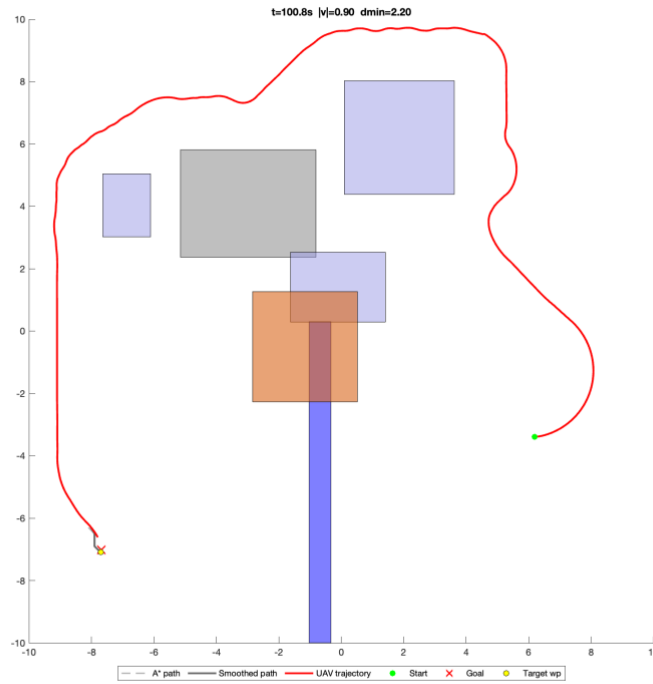
Figure 3: Final Success

For local control, the final version introduced a Frenet-frame path-following method. This method calculates the UAV's lateral error from the path and applies a correction to keep it centered, rather than steering directly at the next waypoint. This makes the motion smoother and more realistic. A repulsion vector derived from the gradient of the distance map was also added so that the UAV automatically veer away from obstacles without having to replan immediately. A heading rate limit prevented sudden turns, while an emergency braking system used short-range ray casts to slow the UAV when it approached close objects. Together, these features gave the UAV the ability to move fluidly around obstacles, even when the environment changed dynamically.

After completing the main simulations, wind dynamics were added to the final version to study how the UAV could compensate for external disturbances. The wind was modeled as a

constant vector with a random direction and magnitude, plus small gusts that varied smoothly over time. To handle this, the controller was modified so that the UAV computed both its desired ground velocity and the corresponding air velocity needed to achieve that motion despite the wind. The wind vector was subtracted from the desired ground velocity, effectively allowing the UAV to "crab" into the wind to stay on course. A minimum ground speed guard was also added to prevent the UAV from being stalled by strong headwinds, ensuring that it continued moving forward even when the wind opposed its motion. These adjustments allowed the UAV to remain stable, reach the goal, and maintain smooth movement even in windy conditions.

Figure 4: Final UAV with Wind

Each simulation frame was drawn and saved to a video file showing the UAV's progress in real time. The videos clearly showed improvements in stability, collision avoidance, and adaptability with each iteration of the system.

Results

The results from this week showed clear progress, even though not every simulation worked perfectly. The first simulation performed well in simple scenes, with the UAV successfully flying around walls and a moving box. The second simulation with randomized environments was inconsistent. Sometimes the UAV navigated correctly, but other times it got stuck or could not find a valid path. Despite the mixed results, it helped identify the causes of instability, such as problems with path smoothing and obstacle sampling.

The final version was the strongest and most reliable. The UAV reacted smoothly to both static and moving obstacles, adjusted its speed automatically, and reached the goal in most trials. It was able to adapt to random worlds without crashing, although there were still small issues with getting too close to walls or pausing briefly when navigating tight areas. The Frenet control and repulsion forces worked together to keep the UAV on track while still maintaining a realistic motion pattern.

After completing the main simulations, wind dynamics were added to the final version to study how the UAV could compensate for external disturbances. The wind was modeled as a constant vector with a random direction and magnitude, plus small gusts that varied smoothly over time. To handle this, the controller was modified so that the UAV computed both its desired ground velocity and the corresponding air velocity needed to achieve that motion despite the wind. The wind vector was subtracted from the desired ground velocity, effectively allowing the UAV to "crab" into the wind to stay on course. A minimum ground speed guard was also added to prevent the UAV from being stalled by strong headwinds, ensuring that it continued moving

forward even when the wind opposed its motion. These adjustments allowed the UAV to remain stable, reach the goal, and maintain smooth movement even in windy conditions.
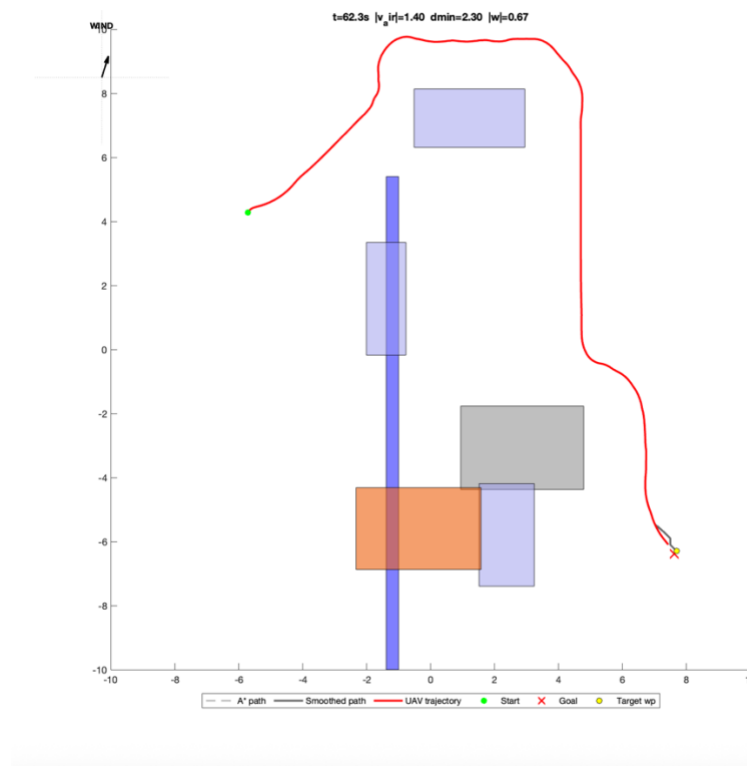
Discussion

Week 9 focused on combining everything developed in the previous weeks into a single working system. The global A* planner gave the UAV an efficient path, while the Frenet-based controller handled smooth local adjustments. The repulsion and braking systems added a level of safety that allowed the UAV to react to unexpected obstacles without completely stopping. The introduction of randomization made the test more realistic, since the UAV had to plan in new environments each time.

The main limitation during this week was with the second simulation, which was unstable when the randomized maps were too tight or when the UAV spawned near an obstacle. The path smoothing method also caused the UAV to cut too close to objects, which led to occasional collisions. Increasing the number of collision checks and adding predictive sampling for moving objects could help reduce this problem in the future. The system could also be improved by refining how the UAV handles corners and adding a more controlled slowdown near the goal.

Conclusion

Week 9 successfully integrated the trained model and control logic into a real-time MATLAB simulation. The UAV demonstrated the ability to plan, follow, and adapt its trajectory around both static and moving obstacles. While the second simulation was inconsistent and highlighted problem areas, it helped guide improvements for the final version. The last simulation, which used A* replanning and Frenet-based control, was able to complete multiple successful runs with smooth and collision-free movement.

Overall, this week showed major progress in making the UAV behave more realistically and independently. The four videos recorded this week include two that show the UAV avoiding stationary objects and two that show it successfully dodging a moving obstacle. Even though there are still small issues with smoothing and getting stuck near edges, the final version of the system represents a solid step toward reliable real-time UAV trajectory optimization. The final addition of wind dynamics further demonstrated the robustness of the UAV control system. By computing air-relative velocity commands that counteracted the wind, the UAV maintained its intended ground path and avoided being blown off course. The system successfully balanced smooth path-following, obstacle avoidance, and wind compensation, showing that it could operate reliably even under varying environmental conditions. This marks a key step toward realistic and resilient UAV trajectory optimization in simulation.

Discussion

Week 9 focused on combining everything developed in the previous weeks into a single working system. The global A* planner gave the UAV an efficient path, while the Frenet-based controller handled smooth local adjustments. The repulsion and braking systems added a level of safety that allowed the UAV to react to unexpected obstacles without completely stopping. The introduction of randomization made the test more realistic, since the UAV had to plan in new environments each time.

The main limitation during this week was with the second simulation, which was unstable when the randomized maps were too tight or when the UAV spawned near an obstacle. The path smoothing method also caused the UAV to cut too close to objects, which led to occasional collisions. Increasing the number of collision checks and adding predictive sampling for moving

Hall 15

objects could help reduce this problem in the future. The system could also be improved by refining how the UAV handles corners and adding a more controlled slowdown near the goal.

Conclusion

Week 9 successfully integrated the trained model and control logic into a real-time MATLAB simulation. The UAV demonstrated the ability to plan, follow, and adapt its trajectory around both static and moving obstacles. While the second simulation was inconsistent and highlighted problem areas, it helped guide improvements for the final version. The last simulation, which used A* replanning and Frenet-based control, was able to complete multiple successful runs with smooth and collision-free movement.

Overall, this week showed major progress in making the UAV behave more realistically and independently. The four videos recorded this week include two that show the UAV avoiding stationary objects and two that show it successfully dodging a moving obstacle. Even though there are still small issues with smoothing and getting stuck near edges, the final version of the system represents a solid step toward reliable real-time UAV trajectory optimization. The final addition of wind dynamics further demonstrated the robustness of the UAV control system. By computing air-relative velocity commands that counteracted the wind, the UAV maintained its intended ground path and avoided being blown off course. The system successfully balanced smooth path-following, obstacle avoidance, and wind compensation, showing that it could operate reliably even under varying environmental conditions. This marks a key step toward realistic and resilient UAV trajectory optimization in simulation.