# Advancing Drone Teleoperation through Feedback-Driven AI

**Jasdeep Singh**

A thesis submitted for the degree of

*Bachelor of Science*

Defense Date: May 9, 2025

Thesis Committee:

Dr. Sriram Sankaranarayanan, Department of Computer Science, Advisor

Dr. Majid Zamani, Department of Computer Science

Dr. Juan Restrepo, Department of Applied Mathematics

Department of Computer Science
University of Colorado Boulder

# Contents

# Chapter 1

# Introduction

As robotic systems become increasingly capable, their integration into critical domains such as healthcare, aerospace, disaster response, and transportation is accelerating. In these incredibly high-risk environments, the collaboration between humans and autonomous systems must be not only functional but also seamless, reliable, and intuitive. While the technical advancement of autonomous systems has received considerable attention, (especially in navigation, perception, and decision making) less emphasis has been placed on designing interfaces and interaction paradigms that enable humans to effectively oversee, guide, and trust these systems.

At a broad level, this work is driven by the urgent need to enhance human-robot interaction (HRI) in safety-critical applications. In such settings, the cost of miscommunication or misinterpretation between a human and a robot can be catastrophic. A surgical robot halting unexpectedly, a drone not landing precisely, or an autonomous vehicle making an unclear decision. All of these are reminders that autonomy is only effective as a human's ability to understand and influence it. The research challenge is also shifting. Researchers aren't looking at solely achieving autonomy, but also designing systems that work with people, adapting to their goals, limitations, and cognitive models.

To meet this challenge, the field must incorporate insights from human-centered design, cognitive science, control theory, and machine learning. One especially promising direction is the development of adaptive interfaces that personalize support and guidance to users based on their real-time behavior and inferred mental state. For example, an interface that can gauge a user's confidence or confusion and respond accordingly. Whether thats through automated assistance, trajectory suggestions, or natural language feedback. This can significantly improve performance and user experience.

This research aims to explore exactly this interaction. Between creating intelligent and adaptive interfaces that facilitate human understanding and control of autonomous systems. A particular emphasis is placed on feedback-driven learning, where user behavior is not only guided by the system but also used to continually refine the interaction loop. In doing so, this work contributes to a broader vision of robotics where machines are not just tools, but collaborative partners. Where autonomy is complemented by interpretability and where confidence and clarity are integral to control.

Ultimately, improving the quality of human robot interaction in these settings is not just a matter of convenience or usability, its a necessity. As robots are deployed in these environments where humans lives and complex decisions are at stake, we must ensure that these systems empower their users, rather than overwhelm them. This research takes a step in that direction, combining formal models of behavior with learning based approaches to generate meaningful feedback and assistance. Thus, enabling a more resilient and human aligned future for autonomous systems.

In many safety critical scenarios, full autonomy is neither feasible nor desirable. Instead human operators are tasked with intervening, supervising, or taking control in complex, rapidly evolving situations. This research is motivated by a central question: how can we design feedback mechanisms that help humans learn and perform better in dynamics control tasks such as drone teleoperations? Drone piloting, especially under real time constraints and noisy dynamics, is difficult for non experts to master. These difficulties are amplified when the interface fails to provide interpretable feedback, or when users are left to infer what went wrong after poor performance. In safety-critical applications where human error can lead to mission failure or physical damage, it is essential to develop systems that assist in users in learning more effectively, not just in controlling the system.

Human feedback is critical in motor learning, especially for tasks involving dynamic systems where optimal actions depend on subtle environmental cues and precise timings. Instructors and copilots naturally provide trajectory level corrections and verbal guidance to help novices improve. However in many human robot systems, this kind of responsive and interpretive feedback is missing. This thesis explores how automated systems can replicate aspects of that feedback using a combination of optimal control theory and Large Language Models (LLMs).

The key contribution of this work is a novel framework that provides human users with both optimal trajectory corrections and natural language feedback generated by LLMs. First, given a user's attempt at a drone landing task, we compute a minimally correcting optimal trajectory by solving a linear program (LP). This alternative trajectory maintains the user's intent as much as possible while corrections for suboptimal control decisions. Second, we use the differences between the user's attempt and the optimal trajectory to generate interpretable feedback through an LLM. This feedback is intended to be personalized, context aware, and pedagogically useful. Which allows users to iteratively improve their performance and udnerstanding of the task.

This approach bridges formal methods with interactive learning and human centered AI. By grounding feedback in optimal behavior and translating it into natural language, the systems provides both action level and conceptual guidance. The result is a hybrid interface that goes beyond binary success/failure signals or opaque metrics, helping users build intuition about dynamic control in an interpretable and adaptive manner.

**Thesis Structure**. The remainder of this thesis is organized as follows:

- **Chapter 2** introduces related work in human-robot interaction, feedback-driven learning, and optimal control.
- **Chapter 3** describes the drone simulation environment and user interface for collecting trajectory data.
- **Chapter 4** presents the formulation of the optimal correction framework.
- **Chapter 5** discusses how LLMs are integrated into the feedback loop, including the prompt engineering and data used to generate feedback.
- **Chapter 6** evaluates the framework through user studies, analyzing both quantitative improvements and qualitative user responses.
- **Chapter 7** concludes with reflections on future directions, including broader applications in HRI and adaptive instruction.

Through this work, we take a step toward more intelligent and supportive human robot interfaces. Systems that not only act optimally, but also teach, guide, and adapt to the human in the loop.

# Chapter 2

# Background and Related Work

This chapter provides a comprehensive overview of the foundational concepts and exciting research relevant to this thesis. We explore the landscape of HRI, particularly in teleoperation and dynamic control tasks, and delve into the mechanisms of feedback driven learning. We then review established methods in optimal control for trajectory generation and correction, highlighting the specific advantages of Linear Programming. Finally, we examine the burgeoning role of LLMs in HRI and educational contexts, setting the stage for our integrated framework.

## 2.1 Human Robot Interaction and Teleoperation

THe increasing sophistication of robotic systems has led to their deployment in domains where direct human intervention is either impractical or unsafe, such as deep sea exploration, space missions, and hazardous material handling. In these scenarios, teleoperation becomes a critical mode of interaction. Effective teleoperation demands not only robust robotic capabilities but also seamless and intuitive human robot interfaces. As robots become more autonomous, the paradigm shifts from direct manual control to supervisory control, where humans oversee and intervene when necessary. This transition introduces new challenges in maintaining human situational awareness, trust, and the ability to effectively guide complex systems [4]

In safety critical applications, such as drone teleoperation for inspection or delivery, the consequences of human error or misinterpretation can be severe. Therefore, designing interfaces that facilitate clear communication and enable precise human influence over robotic actions is a paramount. Traditional teleoperation interfaces often provide raw sensor data or limited control options, requiring operators to possess significant expertise to interpret complex dynamics and execute precise maneuvers. This thesis addresses the need for more intelligent interfaces that bridge the gap between novice human operators and the demands of complex dynamic control tasks.

## 2.2 Feedback and Learning in Dynamic Control Tasks

Human motor learning, especially for dynamic tasks, relies heavily on effective feedback. Feedback serves as a crucial information signal, allowing learners to understand discrepancies between their intended actions and actual outcomes, and to adjust their strategies accordingly. However, generic feedback, such as a simple "success" or "failure" signal, often lacks the specificity and context necessary for meaningful learning. Novice operators struggle to infer the underlying causes of their errors from high level outcomes feedback alone. This highlights the need for personalized, context-aware, and actionable feedback that can guide users through the learning process. [2]

Intelligent Tutoring Systems (ITS) have long explored how to provide adaptive guidance in educational settings. These systems aim to mimic human tutors by diagnosing learner difficulties and providing tailored instruction. In dynamic control tasks, an ITS can analyze a learner's performance, identify suboptimal behaviors, and offer targeted advice. Research in this area often focuses on modeling learner cognition and adapting instructional strategies. For instance, approaches like reward shaping have been explored to guide agents towards desired behavior by providing intermediate feedback signals that reinforce correct actions, even if the ultimate goal is not immediately achieved [3]. This concept of breaking down complex learning into manageable, feedback driven steps is central to improving human performance in teleoperation. The challenge lies in generating feedback that is not only accurate but also interpretable and motivating for human users.

## 2.3 Optimal Control for Trajectory Generation and Correction

Optimal control theory provides a mathematical framework for determining the best possible control inputs to achieve a desired system behavior, subject to dynamic constraints and performance objectives. Various methods exist within optimal control, each with its strengths and limitations.

Linear Quadratic Regulation (LQR) is a widely used optimal control technique for linear systems with quadratic cost functions. LQR provides a linear feedback control law that minimizes a weighted sum of state deviations and control effort. While computationally efficient and providing stable control, LQR is primarily designed for unconstrained problems or those with simple state/control bounds that are handled implicitly. For problems with strict, hard constraints, such as physical limits on states (e.g. position boundaries, maximum velocity) or control inputs (e.g., maximum thrust), LQR often requires extensions like

Model Predictive Control (MPC) combined with quadratic programming, which can increase computational complexity.

Other approaches, including those based on Markov Decision Processes (MDPs), are suitable for problems with discrete states and actions, or continuous spaces that can be discretized. While MDPs can handle complex, non linear dynamics and stochasticity, they often suffer from the curse of dimensionality in continuous state action spaces, making them computationally intensive and potentially unable to guarantee strict constraint satisfaction.

In contrast, LP offers a powerful and direct approach for optimal control problems where both the system dynamics and all relevant constraints can be expressed linearly, and the objective function is also linear. The advantages of LP for trajectory generation and correction are significant. The ability of LP to generate feasible, optimal, and interpretable corrections under explicit constraints makes it an ideal candidate for providing actionable guidance to human operators.

## 2.4 Large Language Models in Human Robot Interaction and Education

The rapid advancements in LLMs have opened new frontiers in human computer interaction, particularly in their capacity for natural language understanding and generation. LLMs can process complex textual inputs, synthesize information, and generate coherent, context aware, and human like responses. This capability is increasingly being explored for applications beyond traditional conversational agents, including education and human AI collaboration.

In educational contexts, LLMs hold promise for creating more personalized and adaptive learning experiences. They can interpret learner queries, explain complex concepts, and provide tailored feedback based on a student's performance. Their ability to generate natural language explanations allows for a more intuitive and less technical form of communication, which is particularly beneficial for novice users in complex domains.

More recently, LLMs have begun to be integrated into human in the loop control systems to provide intelligent feedback. This involved leveraging LLMs to translate quantitative performance data and optimal control solutions into qualitative, actionable advice for human operators. Such systems aim to bridge the gap between the precise, mathematical world of control theory and the intuitive, natural language world of human learning. The work by [1] directly explores LLM powered personalized feedback for human in the loop control tasks, demonstrating their potential to enhance human understanding and performance by providing context aware, natural language explanations of deviations from optimal behavior. This aligns directly with the core objective of this thesis.

## 2.5 Bridging the Gap: Integrating Optimal Control and LLM Feedback

This thesis directly builds upon and integrates the advancements in optimal control and LLM capabilities to address a critical need in human robot interaction: providing effective, personalized feedback for dynamic control tasks. Traditional approaches often fall short in translating complex system dynamics and optimal control strategies into human understandable guidance. Our framework explicitly tackles this "expert gap" by combining the rigor of LP based trajectory corrections with the interpretive power of LLMs.

By first computing a minimally correcting optimal trajectory using LP, we establish an objective, feasible benchmark for desired performance. This optimal trajectory serves as a "personalized expert demonstration" that is close enough to the user's actual path to be relatable, yet optimized for critical performance aspects like smoothness and safety. Subsequently, an LLM is employed to analyze the discrepancies between the user's trajectory and this optimal correction, and to generate natural language feedback. This feedback is designed to be clear, relevant, actionable, and encouraging, directly addressing the limitations of generic feedback.

This integrated approach represents a significant step towards enabling interpretable, data driven support for novice operators. It moves beyond simple success/failure signals to provide nuanced insights into control errors and actionable strategies for improvement, ultimately fostering more resilient and human aligned autonomous systems.

# Chapter 3

# Problem Formulation

## 3.1 Overview of Drone Landing Task

The primary objective of the drone landing task is to enable a human operator to land a quadrotor safely on a designated landing pad using real time keyboard based control inputs. This task occurs in a constrained two dimensional environment, simulating basic physical dynamics such as gravity, inertia, and thrust based motion. The environment serves as a controlled platform for studying human in the loop control, providing a realistic proxy for teleoperation in dynamic and safety critical contexts.

This task encapsulates several challenges faced in broader HRI domains. Operators must understand and compensate for the dynamics of a linear system while responding to environmental cues such as the drone's position and velocity. Thus, the drone landing task is not just a mechanical control challenge, it is also a cognitive one, where the user must build an internal model of system behavior and adjust based on feedback from the interface and system responses.

## 3.2 Definition of Success and Failure

To evaluate user performance, we define clear criteria for successful and unsuccessful drone landings. The classification depends on three key components: position, velocity, and orientation at the moment of landing.

- **Landing Pad Criteria:** The drone's final (x, y) position must fall within a predefined rectangular area representing the landing pad. This zone is typically centered at a fixed position on the bottom of the simulation screen.
- **Velocity Constraint:** The drone must land with a low vertical and horizontal velocity. If the

magnitude of the velocity vector at landing exceeds a threshold (e.g. $|v| = \sqrt{v_x^2 + v_y^2} > 15m/s$), the landing is deemed unsafe, as it would represent a hard impact in a physical scenario.

- **Orientation Constraint:** The drone must land upright. If the final pitch angle deviates significantly from vertical alignment (e.g. $|\phi| > 3°$), the landing is considered unstable.
- **Control Limits:** Maximum thrust magnitudes and torque rates are bounded to prevent extreme or unsafe control behavior.

Based on these parameters, we classify outcomes into three categories:

- **Crash:** The drone exits the simulation boundaries or collides with the ground.
- **Unsafe Landing:** The drone lands on the pad area but violates one or more velocity or orientation constraints.
- **Safe Landing**: The drone lands within the pad boundaries and satisfies all dynamic constraints.

These outcome classifications are essential for generating feedback, evaluating performance trends, and training models to identify improvement strategies.

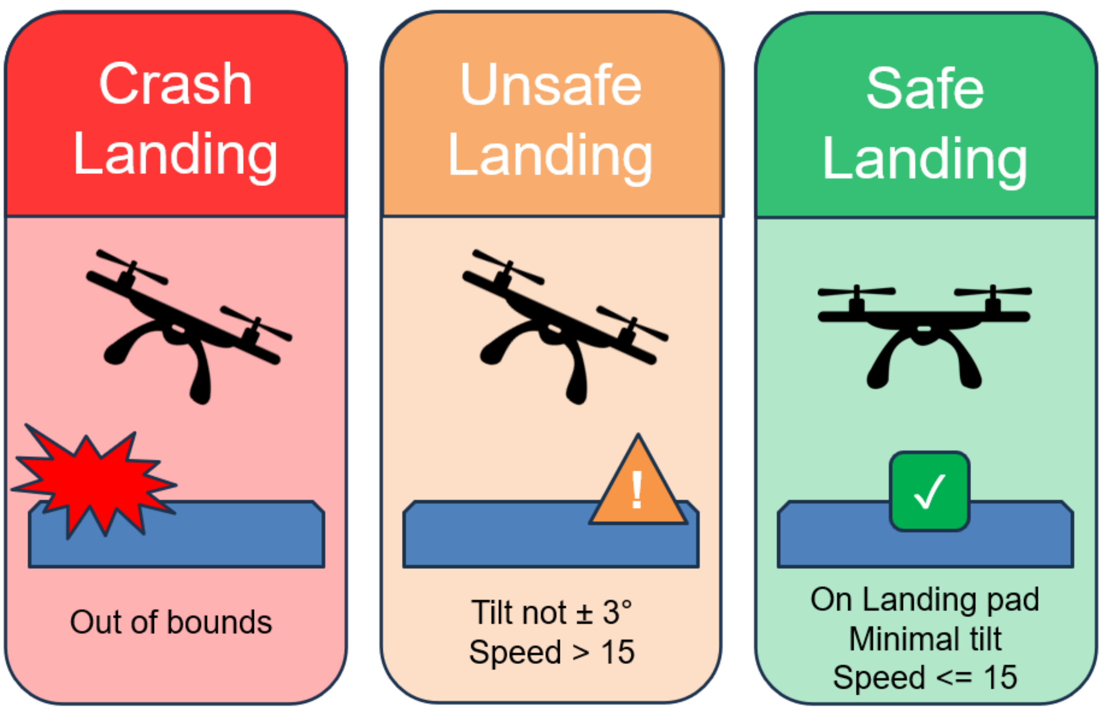



Figure 3.1: Graphic representing different types of Landing

## 3.3 User Control Model

The user interacts with the system via discrete keyboard inputs. The mapping between user actions and system responses is designed to simulate a real world teleoperation interface while maintaining tractability for experimental analysis.

Users control the drone using keys (e.g. arrow keys or "W" and "S"). Each key corresponds to thrust in a specific direction (left right, up and down), or in some setups, torque to adjust drone's pitch and vertical force.

These inputs are translated into force and torque vectors that are applied to the drone's physics model at a fixed sampling rate (e.g. 20 Hz). This rate determines how often the drone's state is updated based on control input.

This control scheme emphasizes the importance of learning not only what inputs to provide but also when to provide them. The timing, sequence, and duration of inputs directly affect the trajectory and final landing condition.

## 3.4 Platform Description

The experimental platform consists of a web based drone simulator, developed with a physics engine and interactive interface. It includes several features to support both the user experience and backend data analysis:

- **Visual Interface:** A 2D side view display presents the drone, the landing pad, and key state indicators (e.g. velocity vectors, orientation angle).
- **Control Scheme** The interface also provides immediate visual feedback on control inputs. The keyboard input system maps each key to discrete control actions, maintaining simplicity and accessibility for participants.

The interface is designed to collect high quality interaction data.

## 3.5 User Data Logging

Robust data collection is central to both feedback generation and performance analysis. Each user trial is logged with the following components:

- **Trajectory States:** Full time series logs of the drone's position, velocity, orientation, control inputs, and key strokes at each timestep.

- **Outcome Data:** Timestamped records of all users. Automatically assigned classification of the trial (crash, unsafe landing, safe landing).
- **Initial Positions:** Sampled from a bounded space to ensure a wide but manageable range of starting states.
- **Environmental Constraints:** Gravity, mass, and feedback gains are fixed across all trials to ensure consistency.

These data enable detailed offline analysis of control behavior, training of predictive models, and generation of personalized feedback. Additionally, they ensure reproducibility of trajectories and ensure each trial presents a novel but learnable challenge, preventing users from relying solely on memorized sequences.

# Chapter 4

# Trajectory Correction via Linear Programming

## 4.1 Motivation

In drone teleoperation tasks, human operators frequently produce suboptimal or unsafe trajectories due to factors such as delayed reaction times, inaccurate control inputs, and insufficient awareness of system constraints. These issues are particularly critical during precise maneuvers like landing, where even small errors in velocity, angle, or position can lead to failure. Rather than discarding the user generated trajectory entirely, we propose a method for minimally correcting it while preserving the user's intent.

Our approach formulates the trajectory correction problem as an LP over a finite time horizon. This method provides several advantages compared to traditional control methods like Linear Quadratic Regulation (LQR) or techniques based on Markov Decision Processes (MDPs) when dealing with systems that have hard constraints. While LQR is efficient for unconstrained linear systems with quadratic costs, it struggles to directly incorporate struct bounds on states, controls, or their rates of change. MDPs can handle complex dynamics and stochasticity but often require discretizations of continuous spaces, which can lead to the curse of dimensionality, and do not inherently guarantee constraint satisfaction or optimality in continuous domains.

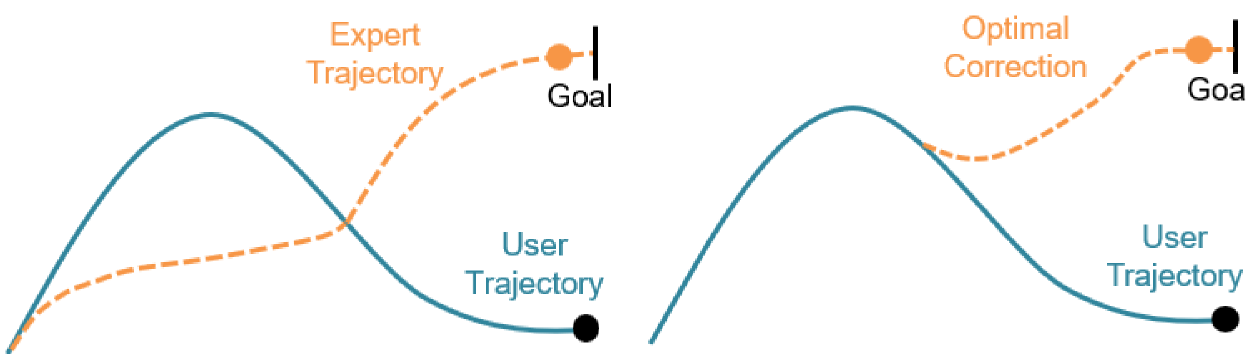
Linear Programming, conversely, is specifically designed to solve optimization problems with linear objectives and linear equality and inequality constraints. This makes it exceptionally well-suited for trajectory correction problems involving linear system dynamics and operational limits that can be expressed linearly. The key benefits of using LP in this context include:

- **Direct Handling of Hard Constraints**: LP solvers can directly enforce linear constraints, ensuring that the corrected trajectory strictly adheres to physical limits, safety requirements, and control limitations.

- **Guaranteed Global Optimality**: For any feasible LP, a globally optimal solution is guaranteed, which is vital for critical applications where finding the best possible correction is necessary.

- **Flexibility in Objective Function**: LP allows for objective functions based on the $L_1$ norm, which can promote sparse control inputs and provide robustness. This contrasts with LQR's quadratic cost, which penalizes larger errors more heavily.

- **Interpretability**: The resulting sparse corrections from an $L_1$ objective can highlight the most critical points where the user's trajectory deviated significantly, offering insights for feedback generation.




(a) Traditional Path Planning (b) Optimal Alternative Trajectory

As illustrated in the comparison between traditional path planning and the concept of an optimal alternative trajectory, traditional systems often rely on generic, precomputed expert trajectories. These can be significantly different from a user's actual trajectory, making them difficult for a human operator to relate to, imitate, or learn from. By formulating the problem as an LP allows us to generate "optimal alternative trajectories" that are computationally derived corrections. These trajectories can serve as personalized expert demonstrations that are close to the user's original path but optimized for safety, smoothness, and efficiency. This proximity to the user's behavior, combined with the optimization for desirable properties, makes the resulting corrections more interpretable and actionable for the human operator when translated into feedback.

## 4.2 System Dynamics

Similar to our drone simulation, we assume the drone operates under discrete time linear dynamics described by:

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t, \quad \text{for } t = 0, 1, \dots, N-1 \tag{4.1}$$

Here, $\mathbf{x}_t \in \mathbb{R}^6$ is the state vector, typically including horizontal and vertical positions, velocities, orientation (angle), and angular velocity:

$$\mathbf{x}_k = \mathbf{x}(k) = \begin{bmatrix} x(k) & y(k) & \phi(k) & \dot{x}(k) & \dot{y}(k) & \dot{\phi(k)} \end{bmatrix}^T \tag{4.2}$$

The control input $\mathbf{u}_t \in \mathbb{R}^2$ includes the thrust and torque inputs:

$$\mathbf{u}_k = \mathbf{u}(k) = \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} = \begin{bmatrix} u_y(k) \\ u_x(k) \end{bmatrix} \tag{4.3}$$

where $u_x(k)$ and $u_y(t)$ represent control forces in the x and y directions at time t.

Matrices $A$ and $B$ are constant and derived from a local linearization or physical modeling of the drone over a discrete time step $\Delta t$:

$$A = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & g\Delta t & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 + K(1)\Delta t & 0 \\ 0 & 0 & K(2)\Delta t & 0 & 0 & 1 + K(3)\Delta t \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{m}\Delta t \\ \frac{1}{I_{xx}\Delta t} & 0 \end{bmatrix} \tag{4.4}$$

Here, $g$ is the acceleration due to gravity, $m$ is the mass of the drone, $I_{xx}$ is the movement of inertia about the pitch axis, and $K(1), K(2), K(3)$ are gains from a feedback controller. The specific values for these parameters are $\Delta t = 0.02$ s, $m = 0.25$ kg, $g = 9.8\text{m/s}^2, I_{xx} = 0.01\text{kg/m}^2$, and $K = [-0.1, -1, -30]$.

## 4.3 Linear Program Based Correction Problem

The goal of the trajectory correction is to generate a corrected trajectory, defined by a sequence of states $(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_N)$ and control inputs $(\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{N-1})$, that minimizes a specified objective function while satisfying the system dynamics and a set of constraints. The decision variables for the LP are the state vectors $\mathbf{x}_t$ and control input vectors $\mathbf{u}_t$ for all relevant time steps.

$$
\begin{aligned}
\textbf{minimize} \quad & \sum_{t=0}^{N-1} (||u_x(t)||_1 + ||u_y(t)||_1) \\
\textbf{subject to} \quad & \mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t, \quad \text{for } t = 0, \dots, N-1 \\
& \mathbf{x}_0 = \mathbf{x}_{\text{start}} \\
& |\phi(N)| \leq 3^\circ \\
& |\dot{x}(N)| \leq 11 \text{ m/s} \\
& |\dot{y}(N)| \leq 11 \text{ m/s} \\
& |x(N) - x_{end}| \leq \text{width of landing pad} \\
& |y(N) - y_{end}| \leq \text{height of landing pad} \\
& |u_x(t+1) - u_x(t)| \leq \Delta u_x \quad \text{for } t = 0, \dots, N-2 \\
& |u_y(t+1) - u_y(t)| \leq \Delta u_y \\
& |u_x(t)| \leq u_{x_{max}} \\
& |u_y(t)| \leq u_{y_{max}}
\end{aligned}
$$

At first we started with an objective function of 0 to check for feasibility. Later on, we moved on to objective functions that minimized the $L_1$ norm of the control inputs. The $L_1$ norm encourages sparsity, resulting in minimal and interpretable changes. We also implemented objective functions that minimized the sum of the squares of the control inputs (minimizing the $L_2$ norm of the control inputs).

The constraints are: the ending position, velocity, and orientation constraints must be met for a safe landing, the control inputs must increase or decrease in increments, and the control inputs must be within a maximum and minimum bound.

The objective function can be linearized and the absolute value constraints can be converted into pairs of linear equality and inequality constraints. By combining the linearized objective function and all the linear equality and inequality constraints, the trajectory correction problem is formulated as a standard LP. The implementation involves modeling the LP using CVXPY and using a commercial solver like Gurobi. A longer horizon $N$ allows for more foresight in planning corrections, potentially leading to smoother and more optimal trajectories over the long run, but significantly increases the size and solution time of the LP.

# Chapter 5

# Feedback Generation via LLMs

This chapter details the methodology for generating natural language feedback for users based on their drone landing performance, leveraging the capabilities of LLMs. The feedback systems is designed to translate the quantitative analysis of user trajectories and optimal corrections, as derived in the previous chapter, into qualitative interpretable, and actionable advice.

## 5.1 Inputs to the LLM

The effectiveness of the LLM generated feedback is directly dependent on the quality and relevance of the input data provided to the model. Following each user trial in the drone simulation, a comprehensive set of data is collected and processed to serve as the basis for feedback generation. The key inputs provided to the LLM include:

- **User Trajectory Data**: The LLM was provided with visual representations of the user's performance, including:
  - A 2D plot of the drone's trajectory, overlaid with images of the drone at regular time intervals (specifically, every 200 time steps). The orientation of the drone image in the plot corresponded to the drone's actual angle at that specific time step. This visual aid helped the LLM understand the drone's path and orientation dynamics.
  - A graph showing the drone's horizontal (x) and vertical (y) velocities over time, synchronized with the time steps shown in the trajectory plot. This graph provided insight into the user's control over the drone's speed in both axes.

  - A graph illustrating the drone's angle over time. This graph was crucial for the LLM to diagnose issues related to maintaining a stable, level orientation, especially during descent and landing.

- **State Violations and Outcome**: Information about whether the trial resulted in a safe landing, unsafe landing, or crash. For unsafe landings or crashes, specific details about the violated landing criteria are included, such as the magnitude of excessive velocity, the degree of tilt beyond the safe limit, or exiting simulation boundaries. This helps the LLM identity the primary issues that led to an unsuccessful outcome.

- **Optimal Corrections and Deviations**: The optimal trajectory starting from different initial states of the user's trajectory. This allows the the LLM to understand where and how the user's actions deviated from an optimal path and which specific maneuvers or states were critical for correction. For instance, the LP solution can indicate that a large horizontal velocity needed to be corrected by a specific thrust input at a particular time.

These inputs, including both the quantitative data and the visual representations, are structured and formatted into a clear prompt for the LLM. The goal is to provide the LLM with sufficient context to understand the user's performance, the objective of the task, the constraints that were violated, and the nature of an optimal solution.

## 5.2 Prompt Design

The design of the prompt is critical for shaping the LLM's output into supportive, actionable, and beginner friendly feedback. The prompt is carefully structured to guide the LLM in analyzing the provided data and generating a response that aligns with the pedagogical goals of the system. The prompt typically includes:

- **Role Playing Instruction**: The LLM is instructed to act as a friendly and encouraging assistant or coach for a novice drone pilot. This sets the tone and ensures the feedback is delivered in a supportive manner.

- **Task Description**: A brief explanation fo the drone landing task an the criteria for a safe landing. This reminds the LLM of the performance objectives.

- **Trial Summary**: A summary of the user's trial outcome and the specific violations that occurred.

- **Analysis of User Trajectory vs. Optimal**: This is a key part of the prompt, where the differences between the user's trajectory/controls and the optimal corrections are described. The visual inputs provided alongside the data likely aided the LLM in performing analyses by offering an intuitive overview of the trial dynamics.

- **Request for Explanation**: The LLM is asked to explain the likely causes of the observed issues in simple, non technical language, relating them to the drone's dynamics. For example, explaining how excessive tilt leads to uncontrolled horizontal movement.

- **Request for Actionable Advice**: The LLM is prompted to provide specific, actionable, suggestions for improvement in future attempts. This advice is grounded in the optimal corrections and the identified user errors. Examples include suggesting smaller control inputs, focusing on maintaining a level orientation, or managing descent rate earlier.

- **Encouragement**: The prompt explicitly requests the LLM to include encouraging remarks to maintain user motivation, acknowledging the difficulty of the task and the value of practice.

The prompt design is iterative and refined based on evaluating the quality and helpfulness of the generated feedback in user studies. The aim is to strike a balance between providing clear diagnosis of issues and offering constructive, easy to understand guidance without overwhelming the user.

# Chapter 6

# Case Studies and Evaluation

This chapter presents the evaluation of the proposed feedback framework through user studies. The primary goal of these studies was to assess the effectiveness of the LP based optimal corrections and the LLM generated natural language feedback in helping users improve their performance in the drone landing task. We detail the structure of the user sessions, the key subjective metrics used for evaluation, and present in depth analyses of the individual user case studied based on these selected trials.

## 6.1 Structure of User Sessions and Trial Selection

The user study involved participants performing multiple trials of the 2D drone landing task using the developed simulation platform. Each participant completed a total of 20 trials. To ensure a diverse range of challenges and prevent users from simply memorizing sequences of controls, the initial conditions (starting position) for each trial were randomized within a predefined bounded space. This randomization ensured that each trial presented a novel albeit solvable control problem, requiring participants to adapt their strategy based on the real time dynamics.

From the entire pool of trials complete by all users, three particularly interesting trials were selected for in depth analysis and feedback generation. These trials were chosen to represent different types of challenges or errors commonly observed during the landing task (e.g. issues with velocity control, angle management, or fine adjustments near landing). This approach allowed for a focused examination of how the feedback system performed in diagnosing and explaining specific types of suboptimal behavior.

For each of these three selected trials, the optimal alternative trajectory was computed using the LP framework described in Chapter 4.

## 6.2 Feedback Presentation and Data Collection

Following the completion of their 20 trials, participants were presented with the data and feedback corresponding to the three selected trials. For each selected trial, the participant was shown:

- An animation of their original trajectory.
- Graphs illustrating their horizontal (x) and vertical (y) velocities over time for that trial.
- A graph showing their drone's angle over time for that trial.
- The newly computed optimal alternative trajectory for that specific trial.
- The natural language feedback message generated by the LLM based on the analysis of their performance in that trial and the optimal correction.

After reviewing the information for each of the three selected trials, participants were asked to complete a questionnaire designed to capture their subjective experience and perception of the feedback's usefulness. The questions included:

**Self Assessment**

- Question 1: Where do you think you went wrong?
- Question 2: What were you trying to do at the moment things went wrong?
- Question 3: How confident were you in your trajectory being seeing the result? (Scale 1-5)
- Question 4: What advice would you give yourself for correcting your drone trajectory?

**Optimal Alternative Trajectories**

- Question 5: Did the alternative trajectory make it clear what you could have done differently? Why or why not?
- Question 6: Did the alternative trajectory seem like the "best" way to correct your path? Why or why not?
- Question 7: Did the alternative trajectory help you understand the LLM feedback better?
- Question 8: Was the combination of the alternative trajectory and text feedback more helpful than either alone?

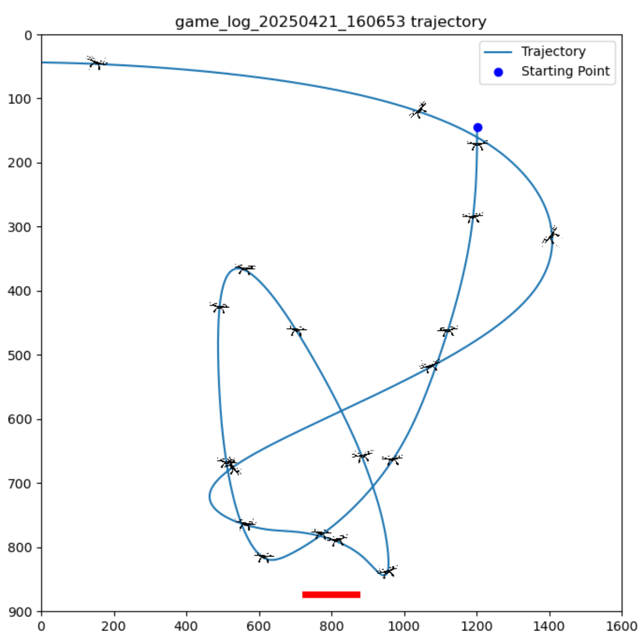
**LLM Questions**

- Question 9: Did the feedback help you understand your mistake? Why or why not?
- Question 10: Was the feedback actionable? Could yo use it to improve your next attempt?
- Question 11: Was feedback understandable in terms of drone controls and your experience?
- Question 12: How useful was the feedback? (Scale 1-5)

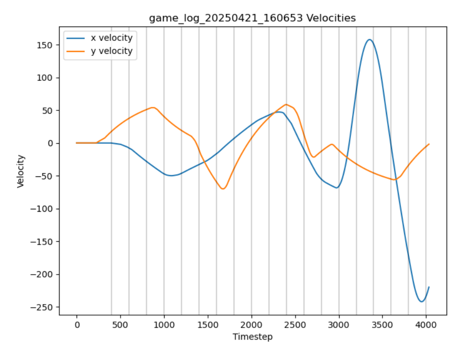
The responses to these questions, both quantitative ratings and qualitative comments, form the basis for the analysis presented in the following sections.

## 6.3 Case Studies

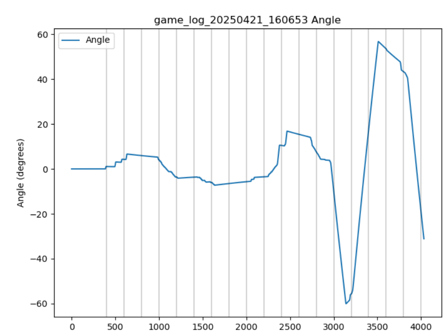
### 6.3.1 Case Study #1




(a) Trajectory




(b) Velocities

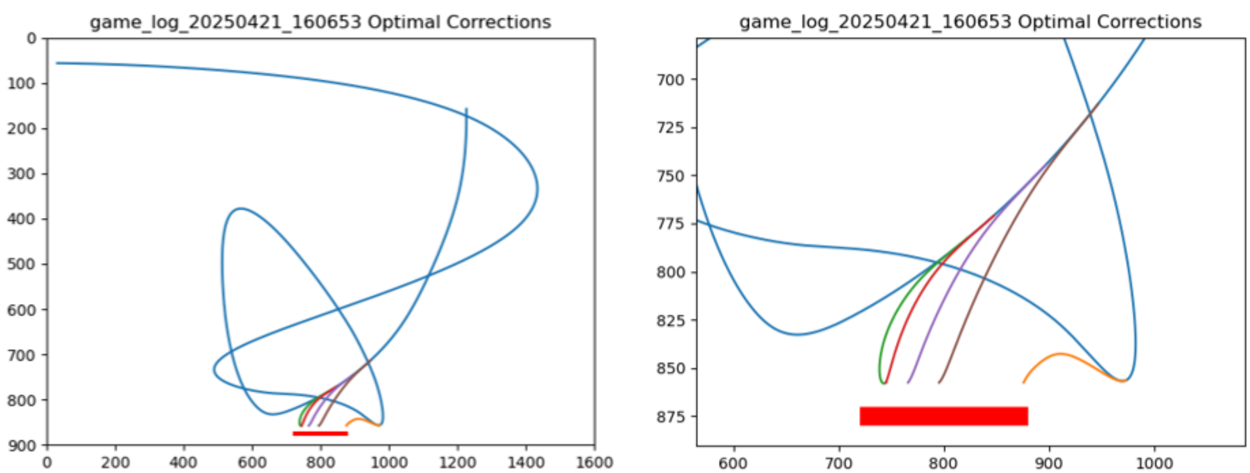



(c) Angle

Figure 6.1: Trajectory, velocities, and angle for game_log_20250421_160653

This case study focuses on a selected trial where the participant, starting from a corner initial condition, struggled with managing the drone's velocity. This starting position required significant control inputs to change direction and slow down the drone's momentum towards the landing pad.

**Analysis of Feedback Effectiveness**: The user's self reflection indicated an awareness of misjudging speed and reacting too slowly, with comments like "I rushed... I need to hedge into a direction and adjust before its too late." Their confidence was low at 2/5. The optimal alternative trajectory was perceived as a "Good starting point" and "useful for some of the paths," providing "ideas I wouldn't have thought of" and complementing the LLM feedback. The LLM feedback was rated highly useful 5/5, described as "Very well written and easy to understand", and "Definitely applicable to future attempt. I just need to slow down". The user noted that the visual path "helped confirmed what went wrong" when combined with the LLM's explanation. This case demonstrates how the combined feedback helped the user identity and conceptualize strategies for addressing underreaction to velocity.

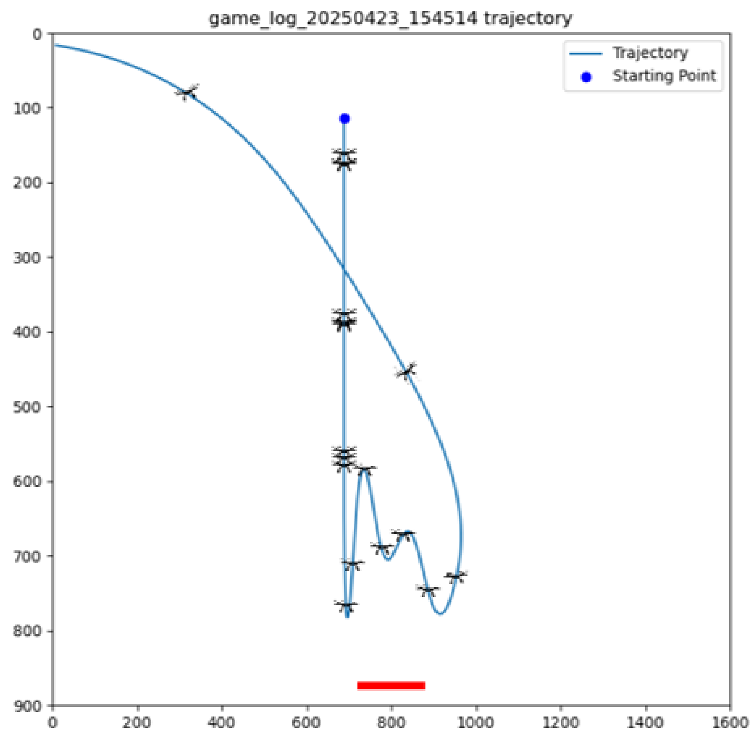



(a) Optimal Trajectories

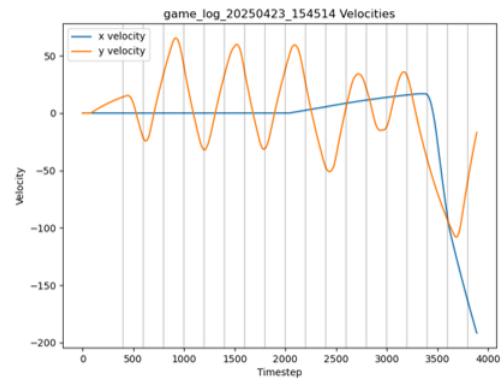
(b) Zoomed-In Optimal Trajectories

Figure 6.2: Optimal trajectories and zoomed-in view for for game_log_20250421_160653

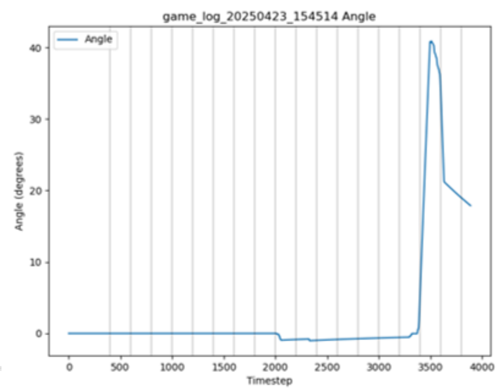
### 6.3.2 Case Study #2




(a) Trajectory




(b) Velocities

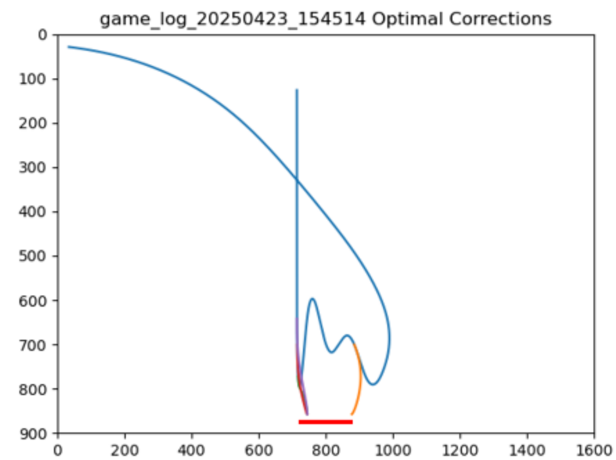



(c) Angle

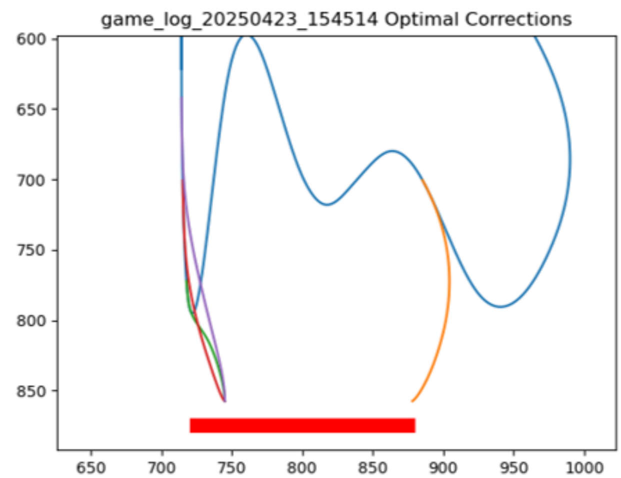
Figure 6.3: Trajectory, velocities, and angle for game_log_20250423_154514

This case study examines a selected trial where the participant exhibited issues with overcompensating during the drone's descent, leading to instability as they approached the landing pad. The trajectory for this trial showed a path that become oscillatory closer to the landing pad.

**Analysis of Feedback Effectiveness**: The user's self assessment revealed an understanding that "Overcompensating near landing caused instability," with a self reflection of "Go as slow as possible... slow and steady is the way to go." Their confidence was relatively high at 4/5. The optimal trajectory was perceived as clear, "I already had the trajectory in mind but couldn't follow it due to overcompensation", and visually helpful, "The graph helped a lot - I'm a visual learner", confirming it seemed like the "best path." The LLM feedback was seen as validating their thought process, actionable ("I can apply it during my next attempt"), and clear ("Presented in layman's terms"). Its usefulness rating was 3/5. This case illustrates how the feedback reinforced existing user understanding and provided actionable strategies for refinement.



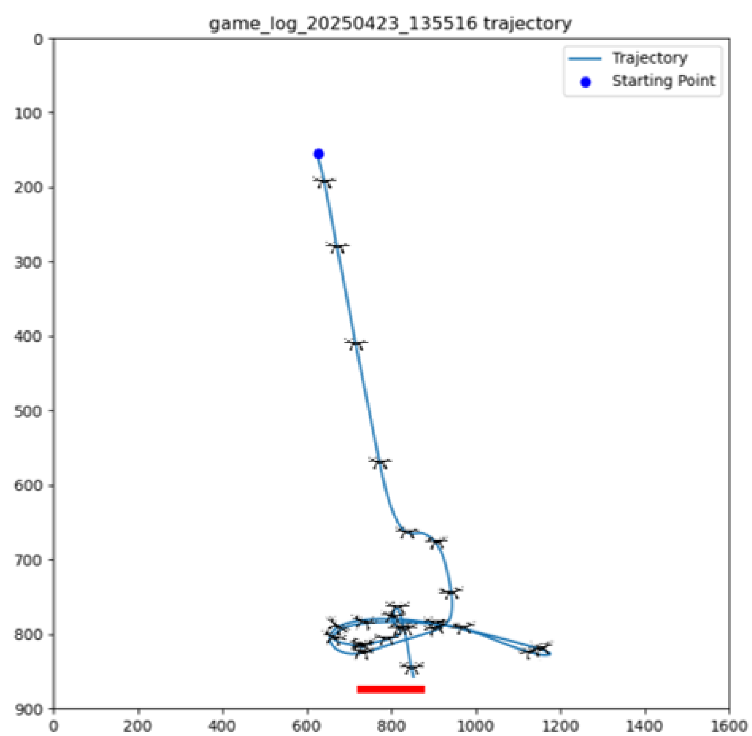

(a) Optimal Trajectories



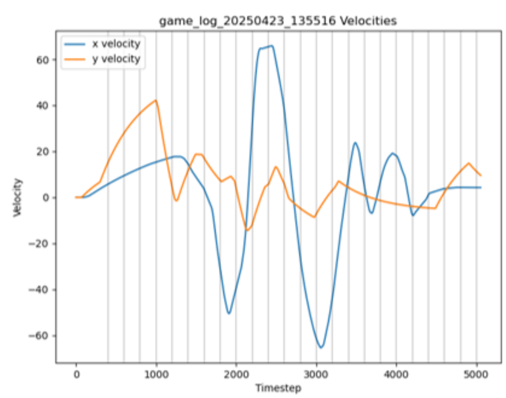

(b) Zoomed-In Optimal Trajectories

Figure 6.4: Optimal trajectories and zoomed-in view for for game_log_20250423_154514

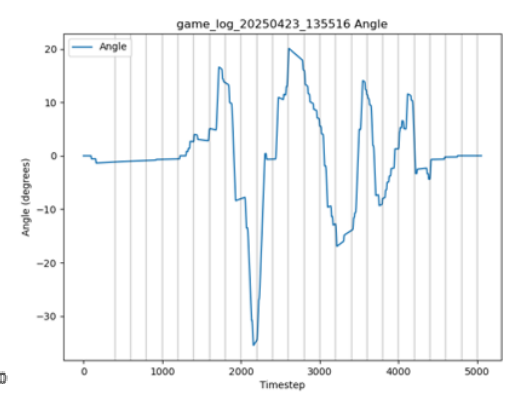
### 6.3.3 Case Study #3




(a) Trajectory




(b) Velocities




(c) Angle

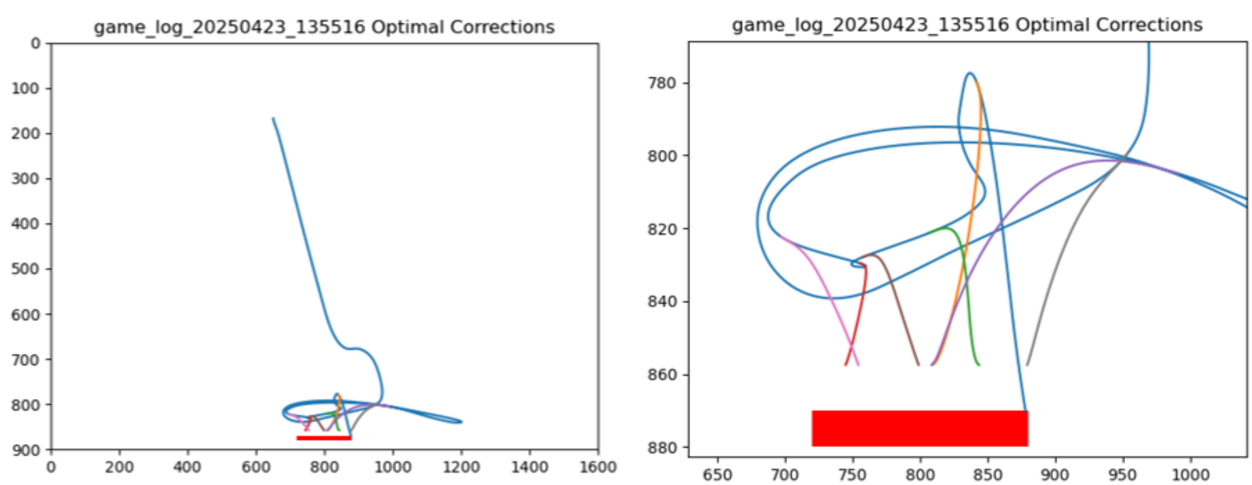
Figure 6.5: Trajectory, velocities, and angle for game_log_20250423_135516

This case study focuses on a selected trial where the participant encountered significant challenges with making fine, micro adjustments as the drone approached the landing pad, leading to instability and an "Unsafe Landing" outcome. The trajectory showed sharp wiggles and quick changes in angle and speed during the descent, indicating the drone was working hard to correct itself.

**Analysis of Feedback Effectiveness**: The user's self assessment identified the main issue as "Crashes occured just before landing due to exiting the safe zone." In hard mode, micro adjustments were hard to control, leading to overcorrection and instability." Their self reflection, "Was trying to make micro adjustments but instead overcommitted," clearly articulated the difficulty with precise control. Their confidence for this trial was moderate at 3/5.

Regarding the optimal alternative trajectories, the user's perception was less clear ("I wasn't really sure what it was trying to show me," and "Not understanding it made it hard to know if it was the best"). However, they acknowledged that the paired feedback was helpful ("both in combination give more context"), suggesting that while the visual alone was ambiguous, it gained value when paired with the LLM's explanation.

The LLM feedback for this trial acknowledged the "Unsafe Landing" and pointed out the "significant changes in your angle and speed' and "sharp wiggles" in the path. The user's evaluation indicated that it was "Reinforced that the models also saw I was struggling near the end". However, it's actionability was rated lower, with the user noting, "I was already learning through practice, feedback might not change much". The usefulness rating was 2/5, the lowest among the three case studies, suggesting that for highly nuanced control issues like micro adjustment instability, the feedback's impact was perceived as less direct or immediately actionable.




(a) Optimal Trajectories

(b) Zoomed-In Optimal Trajectories

Figure 6.6: Optimal trajectories and zoomed-in view for for game_log_20250423_135516

# Chapter 7

# Future Work and Conclusion

This chapter outlines potential avenues for future research building upon the framework developed in this thesis, discusses the implications of this work, and concludes by summarizing the key contributions and their significance for the field of human robot interaction and autonomous systems.

## 7.1 Future Work

The trajectory correction and feedback generation framework presented in this thesis provides a solid foundation for enhancing human teleoperation of dynamic systems. However, several promising directions exist for extending and improving this work:

**Online Trajectory Correction and Real Time Feedback**: The current implementation computes optimal corrections and generates feedback post trial. A significant step forward would be to enable online trajectory correction and provide real time feedback during the teleoperation task. This would require developing more computationally efficient LP solving techniques or exploring alternative real time optimization methods. Real time feedback from the LLM would also necessitate faster inference times and careful consideration of how to deliver information to the operator without causing cognitive overload. The goal is to move towards a system where optimal trajectories are computed and displayed during the trial, and LLM feedback is delivered in game or immediately post trial, adapting based on user progress.

**Personalized Learning Paths Based on Confidence and Performance History**: The current feedback is generated on a trial by trial basis for selected instances. Future work could explore developing personalized learning paths for users. By tracking a user's performance history, confidence levels, and the types of errors they consistently make, the system could tailor the difficulty of trials and the focus of the feedback. For instance, if a user consistently struggles with angle control despite feedback, the simulation could introduce scenarios that specifically challenge this skill, and the LLM could provide more targeted advice or explanations related to orientation dynamics. This aligns with the future work goal of feedback adapting based on the user's learning curve, confidence, and history.

**Expansion to 3D Drones, More Complex Tasks, or Real World Platforms**: This thesis focused on a 2D linear drone landing task. Extending the framework to 3D drone dynamics would introduce additional complexities in modeling, LP formulation, and visualization. Applying the approach to more complex tasks, such as navigating obstacles, performing aerial manipulation, or controlling other types of robotic systems, would require adapting the system dynamics and constraints within the LP framework. Ultimately, validating the framework on real world drone platforms would be crucial, addressing challenges related sensor noise, unmodeled dynamics, and communication latency.

## 7.2 Conclusion

This undergraduate thesis has presented a novel framework for enhancing drone teleoperation through feedback driven AI. The core contributions of this work are threefold:

1. **Development of a New Drone Simulation Environment**: We created a 2D drone simulation environment with accurate linear dynamics and randomized initial conditions, providing a controlled yet challenging platform for studying human in the loop control and collecting high resolution trajectory data.

2. **Formulation and Implementation of LP Based Trajectory Corrections**: We successfully formulated the drone landing trajectory correction problem as LP, capable of computing optimal alternative trajectories that minimize control effort while strictly adhering to system dynamics, landing constraints, and control smoothness limits.

3. **Integration of LLM Generated Natural Language Feedback**: We developed a methodology for using LLMs to translate the analysis of user performance and the computed optimal corrections into personalized, clear, actionable, and supportive natural language feedback delivered to the user.

The significance of this work lies in its potential to enable more interpretable, data driven support for novice operators of complex dynamic systems. By providing feedback that is not only objectively optimal but also understandable and relatable, we can accelerate skill acquisition, improve performance, and enhance user confidence and trust in autonomous systems.

The broad implications of this research extend to the design of human centered autonomous systems across various domains. The approach of using formal methods to derive optimal behavior and then leveraging

LLMs to communicate insights and guidance to human users can be applied to a wide range of human robot collaboration scenarios. This work takes a step towards a future where autonomous systems are not just capable performers but also effective teachers and collaborative partners, empowering humans to interact with and control complex technology more effectively and safely.

# Bibliography

[1] Emily Jensen, Sriram Sankaranarayanan, and Bradley Hayes. Automated assessment and adaptive multimodal formative feedback improves psychomotor skills training outcomes in quadrotor teleoperation. In *Proceedings of the 12th International Conference on Human-Agent Interaction*, HAI '24, page 185–194, New York, NY, USA, 2024. Association for Computing Machinery.

[2] Katherine J. Williams, Madeleine S. Yuh, and Neera Jain. A computational model of coupled human trust and self-confidence dynamics. *J. Hum.-Robot Interact.*, 12(3), June 2023.

[3] Madeleine S. Yuh, Ethan Rabb, Adam Thorpe, and Neera Jain. Using reward shaping to train cognitive-based control policies for intelligent tutoring systems. In *2024 American Control Conference (ACC)*, pages 3223–3230, 2024.

[4] Madeleine Shuhn-Tsuan Yuh, Kendric Ray Ortiz, Kylie Sue Sommer-Kohrt, Meeko Oishi, and Neera Jain. Classification of human learning stages via kernel distribution embeddings. *IEEE Open Journal of Control Systems*, 3:102–117, 2024.