Adaptive Training Systems for Human-Robot Interaction

by

Emily K. Jensen

B.S. and B.A., Case Western Reserve University, 2018

M.S., University of Colorado Boulder, 2022

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

Institute of Cognitive Science

2024

Committee Members:

Bradley Hayes and Sriram Sankaranarayanan, Chairs

Alessandro Roncone

Leanne Hirshfield

Peter Foltz

Jensen, Emily K. (Ph.D., Computer Science and Cognitive Science)

Adaptive Training Systems for Human-Robot Interaction

Thesis directed by Profs. Bradley Hayes and Sriram Sankaranarayanan

This dissertation explores the development of intelligent tutoring systems tailored for training in human-robot interaction (HRI) tasks. This research stems from the increasing need for a workforce capable of collaborating with automated systems. The work presented here addresses this need by proposing Adaptive Training Systems (ATS) that can upskill and reskill the workforce.

I first provide an overview of the current state of adaptive training systems and lay out a theoretical framework for their future development. I discuss automated assessment methods, elements of formative feedback, and existing training systems for psychomotor skills. The theoretical framework is built upon the foundations of Intelligent Tutoring Systems (ITS), highlighting the unique challenges and considerations necessary for adapting these systems to HRI tasks.

I then present empirical work focused on developing methods for automated assessment and feedback. This work focuses on two domains: advanced vehicle drivers and drone pilots. For advanced vehicle drivers, I introduce risk field models to symbolically model personalized driving behavior parameters, predicting lapses in situational awareness and likely trajectories of future behavior. For drone pilots, I employ temporal logic task specifications and robustness metrics to evaluate performance, generating multimodal formative feedback to enhance learning outcomes.

The final section reflects on the presented work and discusses future research directions. I revisit the proposed theoretical framework, assessing its applicability and suggesting improvements for designing more effective ATS. The dissertation concludes by reviewing the key contributions, emphasizing the potential of these adaptive systems to revolutionize training for complex, psychomotor tasks in dynamic environments. Overall, the dissertation underscores the importance of learner-centered, theory-grounded, and modular systems that can adapt to the evolving skill requirements of the workforce, ensuring continuous improvement and skill refinement.

Dedication

To those who don't fit into the traditional boxes.

It may take some work to find, but there is a space that is perfect for you.

Acknowledgements

It truly took a village to get me to the finish line.

My eternal thanks go to Drs. Bradley Hayes and Sriram Sankaranarayanan for taking me on as a student halfway through my degree. You gave me the space I needed to develop into an independent researcher and pushed me to expand my methodological toolbox. You taught me how to get out of my head and actually make things happen.

I am also grateful to the CPS Frontiers team spanning UT Austin, Purdue, UNM, and Penn State. You have selflessly shared your time, advice, code, and encouragement. I hope to continue working together in the future. My particular thanks go to Dr. Meeko Oishi; you have been an invaluable mentor and have advocated for me more than I deserve.

I could not have done this without my family. A huge thanks go to my mom for proofreading all my papers, my dad for reminding me that learning can be fun, and my sister for supplying me with an endless supply of pet photos. You continue to remind me that there is more to life than what appears in this dissertation.

I don't have nearly enough space to thank my friends for the joy they have brought to my life. Here's to the happy hours, crafternoons, niche memes, social lunches, and conference travel buddies. You have helped me more than you know.

And finally, thank you to my health support team. I have learned that mental, emotional, spiritual, and physical wellness are intertwined and will always be a work in progress.

Contents

Tables

Table

Figures

Figure

Chapter 1

Introduction

The continued development of advanced modeling tools such as foundation models provides an exciting opportunity to adapt and augment current methods of teaching and learning [22]. These tools allow us to more deeply interact with knowledge bases, integrate disparate data sources, and personalize information delivery. In particular, educators have quickly adopted machine learning and artificial intelligence tools to inform their teaching and develop personalized experiences for learners.

1.1 Context

1.1.1 Learner Outcome Modeling

An important area of this research uses activity data to predict student success, with the goal of informing instructors where to focus their efforts or delivering automated support. Previous research on predicting student performance has focused on outcomes at a variety of levels. Most broadly, some researchers have attempted to predict drop-out from MOOCs [35, 88] or a summative measure of course performance such as course grade or final exam score [2, 5, 49, 105, 141, 161] (for a review, see [112]). In contrast, other lines of research have predicted student performance on individual assessment items [25, 28, 126, 128, 127, 129, 151], programming problems [50, 102], or entire assessments [40, 49].

Previous approaches have used a variety of features and methods to predict student success. One active line of research uses Bayesian Knowledge Tracing [32] and related methods [40] to model student mastery of specific concepts and skills based on performance on individual problems [14, 126, 128, 127, 129, 183]. Other research has predicted student performance using features external to the immediate learning environment. These include student grades [49], prior performance within the course [28, 50], historical course and instructor information [49], and student demographics [49].

Finally, a growing body of work has focused on predicting student performance using information from the immediate learning environment. Some work has found success using simple counts of activities on the learning platform [2, 89, 141, 161] or engineered interaction patterns and sequences [2, 25, 28, 49, 50, 102]. For example, [141] predicted performance on items using measures of prior performance on specific concepts, current knowledge, item difficulty level, and engineered activity features. In [74], we combine data-driven machine learning modeling with Item Response Theory to predict formative quiz scores using student activity logs leading up to the quiz. This work was published at the 2021 Learning Analytics and Knowledge Conference.

1.1.2 Affect Modeling

In addition to predicting outcomes such as grades, recent research emphasizes the important role student affect and emotions play in the learning process. This work aims to both identify learner emotional states and provides targeted interventions [39]. Common methods include physiological sensors such as heart rate or recorded facial expressions while others rely on sensor-free approaches meant to preserve learner privacy. Baker and Ocumpaugh review work in sensor-free affect detection in educational software and discuss methods for collecting ground truth labels [13]. Additionally, [71] reviews a representative collection of sensor-free affect detection models developed in authentic classroom environments. In [75], we investigate how affective models generalize between different groups of students and discuss the merits of personalized versus generalized affect detection models. This work was published at the 2019 Educational Data Mining Conference.

1.1.3 Teacher Feedback

Research has also considered how to support teachers as they continue to learn and grow in their profession. In the absence of frequent, effective professional development opportunities, scholars have investigated how to deliver automated feedback to teachers so they can self-reflect on their classroom effectiveness.

Previous work has used audio recordings to model classroom discourse at several different levels. The coarsest level is activity classification, which attempts to classify classroom recordings according to broad categories such as discussion or individual seatwork. Using the Language Environment Analysis (LENA) [56] system, Wang et al. [176] used an analysis of turn taking dynamics to identify general classroom activities from audio. Similarly, Donnelly et al. [42] segmented audio into 60-second windows and labeled each with the dominant classroom activity. They then trained models to identify general classroom activities using utterance timing, language, and acoustic features.

Recent work focuses on analysis of classroom discourse at the individual utterance level. For example, Donnelly et al. [43] expands the work in [21] to specifically identify teacher questions. In this work, they transcribed classroom speech using ASR and predicted teacher questions using acoustic, linguistic, and context features. Stone et al. [166] used similar features as well as specific words and phrases (n-grams) to improve utterance-level modeling of several discourse variables such as content-specific questions and instructional statements. Suresh and colleagues have also successfully used deep learning methods to detect specific dialogic strategies in middle school mathematics classrooms [169, 168].

Other work has focused on analysis of classroom discourse at the class session level rather than individual utterances. This approach has been promising when predicting the prevalence of infrequent discourse strategies such as open-ended questions [83]. For example, Olney et al. [121] used word, part of speech, syntactic, and other discourse structure to directly predict the proportion of open-ended questions for a class session. They also showed that this model outperformed models that aggregated predictions at the utterance level. Building off of the work in [121], Cook et al. [31] found that models trained on words and phrases performed similarly to those trained on predefined part of speech and discourse structure [30] features for predicting open-ended questions. Importantly, combining the predictions from these models yielded improved performance over the individual models.

Using automated models to provide teacher feedback is an area still in the beginning stages of research. Dashboards are a popular method of giving student feedback; however, there is little work analyzing potential benefits to teachers and how this may improve student learning [68, 109, 110, 146, 180]. There are a few notable examples of automated teacher feedback systems. Holstein and colleagues have developed real-time systems that can inform and guide teachers during live class sessions. In [68], they introduce Lumilo, which pairs smart glasses with an Intelligent Tutoring System; this system alerts teachers when students need help that the tutoring system cannot provide. Additionally, Poskin et al. developed a smartphone application TeachFX which models the proportion of teacher talk using classroom audio recordings. Finally, Aslan et al. developed a real-time system to alert teachers of student disengagement [6]. However, none of these systems provide teachers with automatic discourse feedback. In [72], we designed an economical system for teachers to record their own classroom data. We then developed dialog strategy detection models in [72, 78] and explored several methods of presenting feedback to teachers. This work was published at the 2020 CHI Conference on Human Factors in Computing Systems and the 2021 Learning Analytics and Knowledge Conference.

1.1.4 Learning in the Workplace

In addition to the classroom setting, advanced computing capabilities are poised to transform training in critical industrial and other workplace settings. Future industrial development will depend on collaboration between humans and automated systems. While some fear losing jobs to automation, experts argue there will be a need for highly-skilled human-automation teams that can adapt to customer-specific tasks [19, 115]. Humans in these collaborative teams must be able to understand how the autonomous system works, how to manage it, and how to adapt when maintenance is needed or other technical issues arise [61]. A recent report estimates that one third of job requirements will require technological skills that are not yet considered crucial [155], meaning that employees will need to continually adapt as technological innovation continues.

Some of the most significant barriers to achieving this industrial development are the ability to scale up capacity as well as upskill and reskill the current workforce [116]. Experts estimate that 50% of existing employees will need to be retrained or upskilled by 2025 to keep up with technological advancement, placing significant pressure on both employers and employees to meet these demands [155, 99]. With the recent developments of artificial intelligence capabilities, researchers are considering how to improve and automate this crucial training.

Training systems and programs for developing industrial skills are a promising opportunity to expand the workforce. For example, sub-baccalaureate training programs and stackable certifications may allow disadvantaged workers to access the training needed to enter highly-skilled industrial sectors [4]. In order to achieve this goal, training programs will need to focus on transferable skills and present interfaces that are "customizable, individualized, and on-demand" to address the needs of each unique learner [70].

Current training methods such as individualized instruction and pre-recorded modules cannot scale up to meet this need to upskill. They also ignore the fact that many employees enter training with skills that can be transferred to a new task. Intelligent Tutoring Systems (ITS) are designed to meet just these demands in classrooms by developing personalized models of students and building on knowledge the student has already mastered. Although previous work discusses applying these approaches outside the classroom [154], existing approaches for training physical tasks has not been systematically researched and integrated with learning theory. I discuss the current state of automated training methods in Chapter 2.

1.2 Thesis Statement

Recent work has developed a strong foundation for personalized learning and training systems. However, the field is critically missing formative feedback immediately directed at the learner's previous actions. This element of training is crucial for guiding motivation and strategies throughout the learning process. Additionally, the work presented above and in Chapter 2 incur significant data needs and rely on domain-specific modeling methods. In this dissertation, I develop an approach to automated, formative, natural language feedback for human-robot interaction (HRI) tasks using assessment derived from formal task specifications. I demonstrate this approach in a driving domain and a drone teleoperation domain.

1.3 Document Outline

This dissertation is organized into three main sections. In the first section, I discuss the current state of Adaptive Training Systems (ATS) and develop a theoretical framework that can be used for future systematic development. Chapter 2 includes an overview of automated assessment methods, elements of formative feedback, and current training systems for psychomotor skills. In Chapter 3, I discuss how ATS build on the foundations of ITS; although these research areas share many characteristics, ATS encounter unique challenges that must be addressed through careful system design and selection of computational tools. I present the main contribution of this dissertation, which is a theoretical framework for developing ATS.

The second section of this document presents my empirical work developing methods for automated assessment and feedback. Chapter 4 introduces the two domains of interest and connects the studies with the proposed theoretical framework. In Chapter 5, I introduce *risk field models* as a method for symbolically modeling personalized driving behavior parameters. This method can also produce likely trajectories of the driver's future behavior and predict lapses in situation awareness. The work in this chapter was published at the 2022 Intelligent Transportation Systems Conference [77] and the 2022 Cyber-Physical Human-Systems workshop [76]. Chapter 6 similarly uses symbolic logic in the form of temporal logic task specifications and evaluates quadrotor drone piloting performance using a robustness metric. We show that this method can provide nuanced understandings of operator performance. Using this evaluation method, we generate automated, multimodal formative feedback to help pilots improve their performance. Components of this work have been published as a late-breaking report at the 2023 Human-Robot Interaction Conference [73] and as a workshop paper at the 2024 Human-Robot Interaction Conference [79].

The final section of this document reflects on the work presented here and discusses how to move the research forward. In Chapter 7, I revisit the proposed framework and propose future directions for designing ATS. Finally, Chapter 8 concludes the dissertation by reviewing the key contributions.

Chapter 2

Background

This work focuses on training for complex psychomotor tasks. Psychomotor tasks require the coordination of physical (grasping, teleoperating) and cognitive (planning, decision making) elements to successfully complete the task [120]. I will adopt the definition of *complex task* given by [179]:

(1) Completing the task requires the examinee to undergo multiple, non–trivial, domain-relevant steps and/or cognitive processes.

(2) Multiple elements, or features, of each task performance are captured and considered in the determination of summaries of ability and/or diagnostic feedback.

(3) There is a high degree of potential variability in the data vectors for each task, reflecting relatively unconstrained work product production.

(4) The evaluation of the adequacy of task solutions requires the task features to be considered as an interdependent set, for which assumptions of conditional independence typically do not hold.

2.1 Automated Assessment for Complex Psychomotor Tasks

Adaptive Training Systems (ATS) must be able to automatically assess performance before providing feedback. This is especially difficult for complex psychomotor tasks because successful performance depends on a variety of factors.

Assessment is also highly dependent on the task domain; as such, previous work in automated assessment has developed specialized methods for the specific domain. For example, Rauter et al. analyzed performance on a rowing task by comparing the velocity profile of the rowing stroke against expert performance [143]. Other studies similarly compare performance to expert trajectories as a benchmark for successful performance [156, 36]. Surgical robotics studies have used physiological metrics such as smoothness, motion amplitudes, and muscular activation [177] in addition to response time for unanticipated events [182]. A recent study evaluated performance in human-robot teaming using number of collisions, number of re-grasps, and total task time [138]. The metrics presented here are largely *outcome-based*, meaning they provide an overall indication of task success, but lack a nuanced description of the learner's *process* of completing the task.

Additionally, the methods used to assess performance are not typically generalizable to task variants or new domains. Recent examples of automated assessment (or more simply, error detection) include domains such as table tennis [104], martial arts [48, 47, 135], piano [113], medical first response [136], industrial production [64], and surgery [149]. Many of these methods rely on neural network classifiers, which require significant data to train and do not provide explanations for their predictions.

To define tasks for robots, previous work has focused on the use of skill primitives, which are atomic actions that may be combined and sequenced depending on the target task. For example, [80] defines manipulation primitives which are defined by parameterized twist and wrench trajectories. Other work uses these skill primitives in directed graphs [171] or a relational assembly model [114]. From an human-robot interaction (HRI) perspective, robotic skill primitives can be used in interfaces to allow human users to quickly define different tasks [165] or easily transfer tasks between robotic systems [137]. These approaches are generally evaluated based on the performance of the robotic system; that is, whether the task is successfully completed and other performance metrics such as time-to-completion. We explore the concept of task primitives more in Chapter 6.

2.2 Formative Feedback

Providing feedback is key to improving a learner's performance. Summative feedback provides a general summary of performance after the learning program is completed [164]. While useful for providing an overview of performance, learners are left to self-regulate their practice in the absence of other feedback. On the other hand, formative feedback is provided during the learning process to help guide future learning [158]. This type of feedback is given more frequently and focuses on encouragement. Based on recent reviews in the educational technology literature, we identified the following elements of effective formative feedback:

- *Reflection*: feedback gives detailed information about the task, process, and encourages the learner to self-reflect [134, 103, 15, 66].
- *Motivation*: feedback expresses confidence in the learner's abilities [134, 103, 65].
- *Timely*: feedback is directly connected to the learner's recent actions [103, 65].
- *Actionable*: feedback provides specific guidance for improvement that is related to the assessment criteria [103, 65, 66].
- *Manageable*: feedback is detailed but not overwhelming to interpret [65, 66].

These elements of feedback have been shown to support learning outcomes and are positively perceived by students in classroom learning settings. One goal of the study in Chapter 6 is to evaluate whether this theory of effective feedback improves task performance in a complex psychomotor task domain.

2.3 Training for Psychomotor Tasks

Recent work in training humans to work effectively with automated and robotic systems is siloed into specific application domains, limiting possible insights about training more broadly. If we consider training motor skills more broadly, many approaches focus on augmenting sensory input to provide control-level feedback during a task [160]. Some examples include visualizing the predicted future trajectory of a drone [178] or generating a haptic response to bias the operator to an ideal course [1, 148].

Recent work in developing end-to-end ATS for complex psychomotor tasks has focused on individual domains such as surgery, sports, marksmanship, karate, driving, aircraft maintenance, and additive manufacturing [96, 189, 154].

Several works have discussed how pedagogically-informed feedback strategies may be implemented in training systems. For example, Korhonen et al. [92] and Pérez-Ramírez et al. [139] discuss how theories such as embodied cognition can be implemented into virtual reality learning environments. Other work proposes inserting erroneous solutions to encourage critical thinking [26] or using adaptive epistemic feedback for training [100]. However, none of these studies have implemented and evaluated the effectiveness of these theories.

Training systems that provide performance feedback tend to rely on prerecorded responses or templates for reacting to failure modes [37, 123, 36] or display statistical summaries of key performance outcomes [189, 156, 143]. An ultrasound placement study generated a visual comparison between the learner's placement and orientation compared to an expert [156]. These studies indicate an opportunity to investigate the use of generated natural-language text for providing formative feedback to learners. In the study presented in Chapter 6, we provide statistical summaries as a baseline condition and compare learner performance to generated text containing the identified elements of effective formative feedback.

Generating new training examples is out of scope for the work presented here. However, some researchers have begun to explore how to define a curriculum for effective automated training [98].

Chapter 3

Theoretical Framework

3.1 Introduction

In Chapter 1, I discussed the need for adaptive systems that can train the future workforce for human-robot interaction (HRI) tasks and reviewed previous work in this area in Chapter 2. The literature on automated training for complex, psychomotor tasks is relatively new and is lacking a methodology guided by theories of learning.

Intelligent Tutoring Systems (ITS) research is a mature field and provides well-established approaches to automated assessment, feedback, and task selection and generation. However, ITS are limited by their use in static, well-defined domains such as algebra or introductory computer programming. We need to adjust the approach and methods of these system to apply to HRI domains.

In this chapter, I introduce the core contribution of this dissertation by describing a framework for Adaptive Training Systems (ATS) based on the foundations of ITS. Figure 3.1 summarizes the key components and relationships of the framework. The framework is defined by an iterative cycle that is triggered when the learner attempts the provided task to demonstrate their skill. After the task is complete, the system then assesses the learner's performance, updating the internal record of the learner's skill, and generates specific feedback. Finally, the system provides the learner with a new practice task and the cycle repeats.

3.2 Background on Intelligent Tutoring Systems

Modern ITS developed from a desire to automate educational goals [33]. As computing tools developed, researchers developed increasingly advanced systems such as Programmed Instruction (mid 1960's) and Intelligent Computer-Assisted Instruction (1980's) [159]. The major elements of ITS have been presented since 1973 [63] and have not considerably changed over the years. Figure 3.2 gives a representative overview of these elements. Each element of the figure presents its own series of research questions and design considerations. Many ITS have been developed with the intent of improving a subset of these elements. For example, the Assistments platform [144] was initially designed to address both assessment and assisting students; it is still used in current studies continuing to refine other aspects of the platform. In this section, I provide more background on key elements of Figure 3.2, focusing on pedagogical theory and how it may or may not align with automated methods. An excellent overview of many of these elements can be found in [132] and [133].

Domain modeling is used to define the scope of knowledge targeted by an ITS. This is similar to setting concrete learning objectives for a course [38]. A common method of representing the domain is the Knowledge-Learning-Instruction Framework [87]. In this framework, experts define *knowledge components* as a fundamental unit of knowledge that can be inferred from performance on a set of tasks (tasks often correspond to items in a formal assessment). A Q-matrix is then a mapping between knowledge components and related tasks. That is, $Q_{tk} \in \{0,1\}$ represents whether task t is associated with knowledge component k. A common approach to modeling knowledge components and their structure (such as a hierarchy of prerequisites) is dynamic Bayesian networks, which can include uncertainty in estimated parameters [132]. Domain modeling is usually done by experts in the target area, but some automated approaches exist for discovering latent knowledge components [18]. Further discussion is outside the scope of this thesis; in the work I present later, domain modeling is completed by the research team or, in the case where we use existing tools, has been done by system designers.

A learner's knowledge and skills are inherently unobservable. To estimate what the learner knows, we must represent the structure and dynamics of knowledge (Learner Modeling) and infer the learner's current state given their performance on a task (Assessment).

Learner models represent the learner's current knowledge and skills. Using the Knowledge-Learning-Instruction framework, a learner model corresponds to the probabilities that the learner has mastered each knowledge component. A common learner model in ITS is Bayesian knowledge tracing, where the student is represented as a Hidden Markov Model [32, 132]. This model assumes a knowledge component is either known or unknown; students may transition between these at discrete time steps (see Figure 3.3). The relevant parameters are:

- correctness of the observed answer (c)
- probability the learner mastered the knowledge component (θ)
- probability of initially knowing the knowledge component (P_i)
- probability of learning the knowledge component at the current time step (P_l)
- probability of slipping and answering incorrectly even if the knowledge component is known (P_s)
- probability of guessing correctly even if the knowledge component is not known (P_g)

The learner parameters (P_i, P_l, P_s, and P_g) are global values applied to all learners. At each time step, the model uses Bayes rule to update the new estimate of the learner's skill (θ) based on the correctness of their answer. There are many variations on the basic Bayesian knowledge tracing model, such as models that allow individual student parameters [183] and varied item difficulty [128].

Another popular learner model is Performance Factors Analysis [130], which is a class of methods using a logistic function to update the learner's estimated abilities and predict correctness on a given item. This method is easily extended to include additional parameters such as item difficulty parameters. The above methods represent simplistic models of the learning process, but have been shown to be effective in a variety of contexts. Recent work has also investigated the use of machine learning methods such as neural networks [102]; these methods may be desirable in some cases due to their lack of assumptions, but are less interpretable for users. The work presented later in this thesis intentionally uses methods that allow interpretability.

Assessment is paired with a learner model to estimate whether the learner has mastered a particular concept. Using the Knowledge-Learning-Instruction framework, performance on tasks is mapped directly to knowledge components. For example, if a learner performs poorly on task t, it is unlikely that they have mastered corresponding knowledge component k. ITS use a variety of heuristics to determine if a learner has reached mastery [133]. For methods such as Bayesian knowledge tracing and Performance Factors Analysis discussed above, performance estimates are included as an estimated parameter in the model. This value can be used with an empirically determined cutoff to signal mastery; for example, if the performance estimate θ is greater than 0.8 (out of 1), then the system may move the learner on to the next concept. Other mastery heuristics that do not make assumptions about learning are:

- Consecutive correct: answering N items correctly in a row means the learner has mastered the concept
- Moving average: answering N items correctly out of the last K attempts means the learner has mastered the concept
- Exponential moving average: same as moving average but recent attempts are weighted higher

Evidence-Centered Design [107] provides another approach to assessment seen less frequently in the ITS literature. This approach designs the system around assessment (see Figure 3.4 using input from both domain experts and system developers). Using this approach, the development team starts by identifying key evidence that should be used to assess mastery. Training examples and assessment activities are then developed to elicit this evidence. This approach is specifically designed for assessment in complex tasks, where a learner's attempt gives more information than just correctness. Similar to this approach, in Chapter 6 we present a framework for assessment in HRI contexts; we define primitive tasks that compose larger tasks and develop a nuanced assessment system using signal temporal logic specifications.

An instructional policy is used to move the learner towards mastery. These policies inform what types of tasks we assign to learners and when we introduce material (this is the curriculum). Instructional policy is an exciting opportunity to operationalize established pedagogical principles into an automated learning context. Some examples of instructional policies include:

- Deliberate Practice [52]: refines performance through feedback and focused repetition
- Zone of Proximal Development [173]: scaffolds learning by maintaining a difficulty level that the learner can achieve with assistance
- Spaced Repetition [7]: administers new or difficult concepts more frequently than mastered or easy concepts
- Retrieval Practice [147]: promotes active retrieval of knowledge (usually through low-stakes assessment) rather than simply reviewing information. It is especially effective when paired with feedback.

Domain modeling and development of knowledge components give some required structure to a curriculum; for example, some skills must be mastered before moving on to new concepts. New modeling approaches have the opportunity to adapt problem presentation and ordering. Recent work in reinforcement learning has framed curriculum generation as an optimization problem attempting to maximize final performance [118].

Providing feedback about a learner's performance is another type of instructional policy. Summative feedback provides a general summary of performance after the learning program is completed. These are often associated with stressful, high-stakes assessments such as course final exams. On the other hand, formative feedback is provided during the learning process to help guide future learning [158]. This type of feedback is given more frequently and focuses on encouragement. I discuss this more in Chapter 2.2.

A learner is much more than a computer that takes in data and applies it to a task; humans are affected by their motivation, emotions, and strategies. A growing body of work is investigating how the learner develops over time and interacts with the training systems themselves. Achievement emotions (e.g., frustration, boredom, enjoyment) are well-studied for their impact on learning outcomes in the classroom (see [131] for an overview). This theory is starting to be integrated into ITS with the hope of building interventions and adjusting instruction based on a learner's current emotional state. Other work investigates how learners develop strategies to regulate their own learning [9] and reflect on their performance [146].

3.3 Machine Learning for Classroom Skills

The framework presented in this chapter is informed by my previous work developing automated assessment and feedback tools in traditional classroom learning domains. In these studies, I investigate how machine learning methods can be used to understand the learner's internal states, with the intention of providing personalized feedback and interventions. I give an overview of these studies below and discuss the main lessons learned.

In [75], we developed models of students' affective states using activity trace data. The activity trace data included counts of actions in an online algebra learning environment such as playing videos, posting on the discussion board, and answering quiz questions. Using data from 69,174 students, we trained Bayesian Ridge Regression models to predict ratings to surveys targeting affective states such as frustration, pride, and boredom. The core contribution of this study is evaluating the benefits of using personalized models compared to a generic model trained on the entire dataset. To do this, we clustered students into groups based on their demographic features and usage patterns on the algebra platform and trained the affect models on data from those groups of students. We found that these models had a small increase in predictive performance compared to the general models, but the difference was not likely to be meaningful when integrated into a larger recommendation system. One key lesson from this study is that more data does not always yield better models. After conducting simulations training the models at different sample sizes, we found that increasing the dataset beyond 2,000 samples does not appreciably improve the personalized models over the general model. However, models using less than 2,000 samples were not stable and may not be reliable.

The work in [74] used the same online algebra domain. In this study, we developed models to predict students' scores on a formative algebra quiz using their activity data preceding the quiz. Using data from 32,685 students, we trained Random Forest Regression models to predict student scores on three-question quizzes. We compared these machine learning models to Item-Response Theory models, which are theoretically-grounded models of student learning and assessment. The key finding from this study is that features relating to more active learning strategies (such as reviewing incorrect questions on a previous quiz) were predictive of higher quiz scores, which align with ICAP [29] and retrieval practice [147] theories of learning.

Another project focused on developing speech models to support teachers' professional development. In [72], we equipped 16 English Language Arts teachers with microphones to record their classroom speech. The teachers recorded 91.7 hours of classroom data. Experts coded a subset of the speech data, yielding nearly 17,000 utterances. With this data, we developed an automated pipeline using signal processing, feature extraction, and Random Forest models to process the speech data and identify key elements of dialogic speech that have been shown to engage and challenge students [157]. One practical finding of the study was that teachers were able to self-record good quality data for automated analysis, even in noisy and unstructured classroom environments. We also found that our models achieved comparable performance to human experts in identifying dialogic speech and were robust to speech transcription errors.

In a follow-up study using the same dataset, we investigated how deep learning methods impact the predictive accuracy and feedback effectiveness of the speech models [78]. We compared the Random Forest models from the previous work with a fine-tuned BERT language model using simulations of different dataset sizes. We found that the deep learning (BERT) models performed better for larger dataset sizes, but the Random Forest models were more accurate when using smaller datasets and when identifying dialogic speech with lower incidence rates. When presenting feedback at different levels of granularity, we also found that the deep learning models performed better, although it is not clear whether this difference would appreciably benefit teachers. This study served to highlight different design choices one could make while designing an automated feedback system based on the availability of data resources and computing capabilities.

Several key methodological themes emerge from these studies. First, the automated methods presented here depend on relatively large datasets. To train the models, we used over 70,000 affect surveys, 210,000 algebra quizzes, and 16,000 teacher utterances. Datasets of this scale will likely not be available for robotics training, since tasks often require specialized equipment or take more time to complete. Second, the domains we considered had well-defined labels, making them suitable for supervised learning methods. For example, we used self-report data as labels for affect and utterance-level expert codes for classroom discourse. In complex psychomotor tasks such as teleoperation, skill assessment needs more nuance; a learner may be able to move an object to the desired location, but their performance also depends on how many times they crashed into other equipment and how much power they used to complete the task. Finally, the set of possible assessment outcomes is fixed and not trivially changed. If we wanted to model a new affective state, we would need to collect a new dataset of survey responses and train a new model. The next section discusses the need to adapt the training framework to address these drawbacks so it can be effective for training in more complex domains.

3.4 Adapting Intelligent Tutoring Systems for Human-Robot Interaction

ITS and ATS share several key components we can exploit for future development. First, learners need to demonstrate mastery over a set of desired criteria. For ITS, these criteria may be a set of knowledge components for a specific unit (e.g., adding fractions with different denominators). The ATS I discuss here include a combination of cognitive and physical skills, such as planning an efficient flight path. Given the performance criteria, both systems also assess learner mastery according to some domain model of ideal performance. ITS may use both experienced classroom instructors and pedagogical researchers as domain experts for developing these models; the pool of experts for developing HRI curricula will be much smaller, especially when teaching platform-specific skills. Both ITS and ATS will need to provide interventions to move the learner towards mastery. This may include just-in-time feedback during the task or more comprehensive analysis after the task is complete (this is pictured in Figure 3.1). Finally, the system needs to decide what the learner should work on next. ITS frequently include a bank of practice tasks as well as a robust knowledge graph of how the learner should proceed through the curriculum. This structure is less clear for ATS, which may have more skill components that can be combined in a variety of ways.

Despite these similarities, there are critical differences that impact the design of ATS compared to ITS. The types of performance criteria are inherently different between the two types of systems: ATS need to assess more than cognitive skills. This requires a different approach to assessment that focuses on the nuance of performance rather than simply assessing whether the performance was correct or incorrect. I discuss this more in Section 6.1.2. Second, ATS need to be able to cover a wider range of possible training scenarios that vary over environmental conditions, physical task layout, and task constraints. This makes it unlikely that ATS can depend on a fixed set of task templates or predefined questions. We will need different methods to generate tasks that target a specific skill for the learner to practice. Finally, it is unclear if or how learning processes differ between cognitive and psychomotor tasks. There is little work exploring how previous theories of learning transfer between these domains. This means that current principles of feedback may not apply in ATS domains; at the very least, providing feedback will be more complex for psychomotor skills.

Other practical differences surface when developing ATS. First, the role of the expert may be much more limited in an ATS setting, since the targeted HRI platforms are much more specialized compared to traditional academic settings. This means we need to empower others to be able to develop these systems, such as through the design of intermediate interfaces that transform natural language instruction into formal task specifications. Additionally, the capabilities of robots and autonomous systems are quickly evolving. As platforms become obsolete and new platforms come online, we need to be able to transfer learner models between systems to reduce redundant training. Finally, there currently is no standardized curriculum for psychomotor tasks. While classroom learning domains often have structured learning paths that build on previous knowledge to master more complex tasks, designers of ATS are forced to develop a curriculum in addition to their desired technical innovations.

These differences between ITS and ATS objectives present several methodological challenges, particularly for assessment. As discussed above, the typical data-driven pipeline often depends on acquiring large datasets to develop predictive models. For HRI domains, more data may not be the best approach. Similar to the principles in Evidence-Centered Design [107], developers will need to think carefully about which data streams will be useful for assessment. Additionally, developing methods that do not require prior learning will alleviate the need to collect more data for training. Another key consideration is that assessment and decision-making methods will need to adapt as the tasks change. As robotic systems develop new capabilities, their human collaborators will need to adapt to new sets of required skills. In particular, machine learning assessment methods likely will not be able to flexibly adapt to these changes since they are specifically trained to assess one type of task. Work in model transfer and generalization may solve this challenge in the future. Finally, ATS need to be able to differentiate between different aspects of performance. While some work in ITS assessment includes the concept of partial credit [175], most related work in assessment focuses on identifying why the learner got the problem incorrect rather than identifying nuances in performance outcomes.

Based on these challenges, I propose the following principles and best practices that should govern the development of ATS:

- Systems should be *learner-centered.* Learner modeling and interventions should focus on personalization and adapting to the learner's current skill level rather than following a fixed procedure.

- Systems should *start with theory-grounded methods.* While not all established pedagogical theory will transfer to psychomotor tasks, using these approaches as a starting point will allow for more systematic development and experimentation to determine the limits of these theories.

- Systems should be *modular.* This will allow designers to quickly prototype new components that target specific aspects of the training cycle while minimizing impact on other parts of the system.

- Systems should *emphasize real-world skills.* To benefit learners, practice tasks need to simulate key characteristics and challenges they will find in real life. This does not mean visualizations need to be photo-realistic; rather, simulations can abstract away unnecessary detail while focusing on developing realistic system dynamics and environmental conditions that may impact their performance in the field.

- Systems should focus on *continuous improvement and skill refinement.* Learning should be a cyclical process where learners repeatedly practice skills with increasing complexity.

The framework in the next section shows one method of integrating these design guidelines.

3.5 Proposed Framework for Adaptive Training Systems

The objective of this framework is to develop a dynamic and adaptive system designed to enhance training in human-robot and autonomy interaction tasks. This framework aims to systematically guide learners through an iterative process that is personalized to the learner's evolving skill level. By integrating theory-driven policies with flexible computational assessment methods, this framework seeks to address the unique challenges posed by the rapidly evolving nature of robotic and autonomous systems, ultimately improving skill acquisition and making training more accessible. While this framework can be applied to traditional academic domains, it is meant to expand the scope of previous ITS frameworks to encompass complex psychomotor tasks.

The training cycle begins with a *task attempt*, where the learner tries to complete the given HRI task. The initial task may be the full target task or a simpler version of the target task designed to assess the learner's current skill level. While the learner attempts the task, the system records activity traces that document the learner's actions. Activity traces may include control inputs, trajectories of relevant task objects, and event records such as collisions and reaching waypoints. These traces are stored in an activity history knowledge base and can be queried in later parts of the learning cycle.

After the task attempt is completed, the system completes an *assessment* of the learner's skill. Using the assessment criteria and activity history, the system analyzes the learner's performance and updates the learner's estimated skills in the learner model knowledge base. The assessment module is flexible and can accommodate a variety of assessment methods and learner models. Chapters 5 and 6 introduce two possible configurations.

The system then generates *feedback* based on the results of the assessment. First, the pedagogical policy knowledge base queries the learner model for the current estimated skill level. The pedagogical policy can be determined based on a specified learning theory framework or data-driven methods. Given the estimated skill level, the pedagogical policy determines the feedback content and the feedback delivery. The feedback content consists of the information to be conveyed to the learner. The feedback delivery involves the presentation of the information, such as feedback modality and formality.

After delivering feedback, the system completes the learning cycle with *task generation*. Here, the curriculum policy knowledge base queries both the learner model and pedagogical policy model to determine what the learner should practice for the next cycle. Like the pedagogical and learner policies, the curriculum policy can be based on a variety of established learning theories or policies learned from data. The curriculum policy determines what skills to target, how difficult the task should be, and the task assessment criteria. The system then generates a new practice task that meets these criteria and presents the task to the user, starting the learning cycle again.

In this chapter, I present a model of after-interaction assessment and feedback, where each module processes sequentially after the learning attempt is complete. This framework can be adapted to occur in real time, where the assessment, feedback, and task-generation modules happen just-in-time while the learner is attempting the task. This approach can use predictions of the learner's future actions to proactively adjust the training content on the fly.

3.6 Conclusion

In this chapter, I introduced the central contribution of this dissertation, a framework for ATS for HRI tasks. This framework addresses the limitations of ITS; namely, ITS are designed for static and well-defined cognitive tasks. The framework presented in this chapter is a step towards developing training systems for complex, psychomotor tasks whose required skills change over time.

In the remainder of this dissertation, I discuss the application of the proposed framework in two case psychomotor domains: driving in an autonomous vehicle and piloting a quadcopter drone. Chapter 5 investigates the use of risk field models as an assessment component of the training framework. Chapter 6 investigates the use of formal task specifications for assessment and generative artificial intelligence for creating personalized formative feedback.

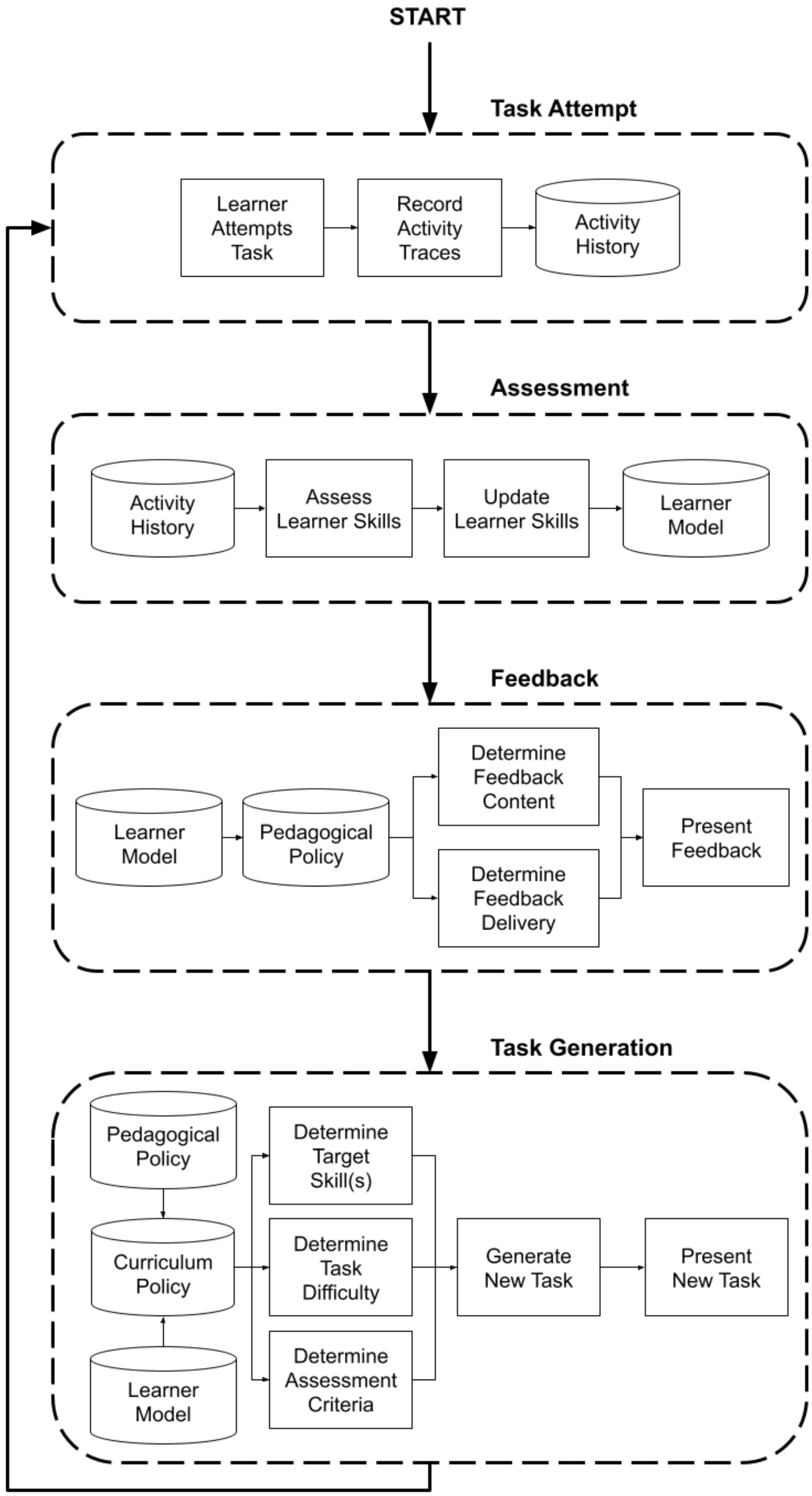


Figure 3.1: Diagram of the proposed ATS framework. Solid boxes represent actions and decisions, cylinders represent knowledge bases, and dashed boxes represent the core modules of the framework. Light arrows represent the flow of data within a module and heavy arrows represent the flow of the iterative learning cycle.

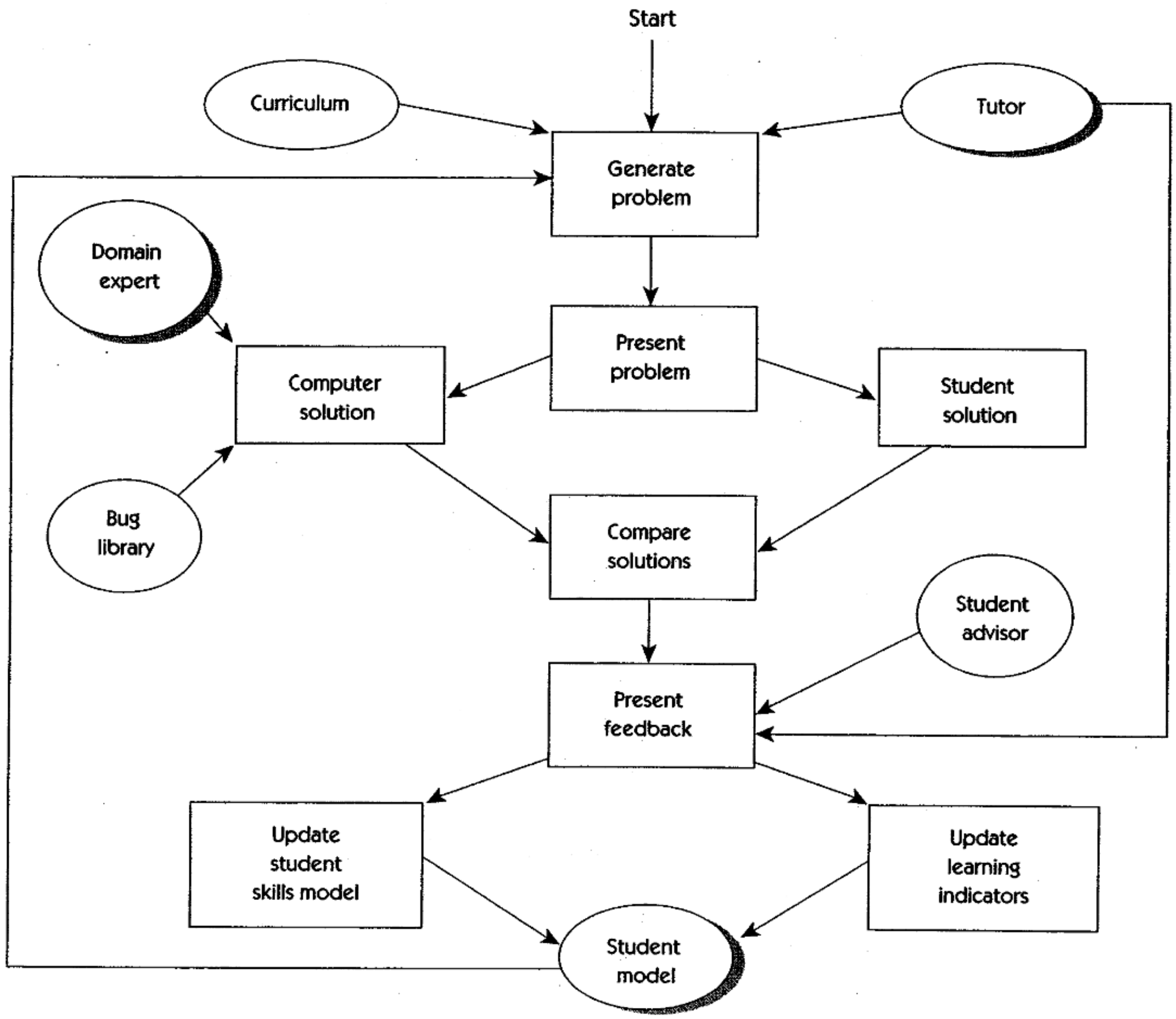
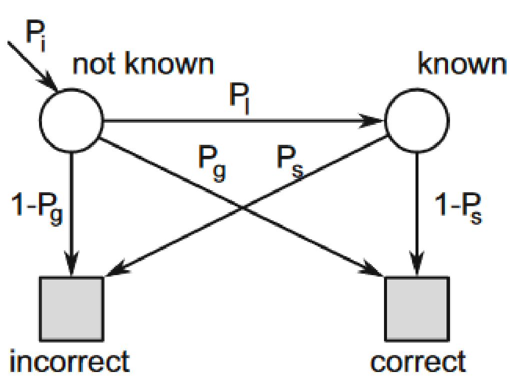


Figure 3.2: Overview of the elements of an ITS, from [159]



update equation:

if $c = 1$: $\theta' := \frac{\theta(1-P_s)}{\theta(1-P_s)+(1-\theta)P_g}$

if $c = 0$: $\theta' := \frac{\theta P_s}{\theta P_s+(1-\theta)(1-P_g)}$

$\theta := \theta' + (1-\theta')P_l$

prediction equation:

$P_{correct} = P_g \cdot \theta + (1 - P_s) \cdot (1 - \theta)$

Figure 3.3: Depiction of structure and equations for Bayesian Knowledge Tracing from [132].

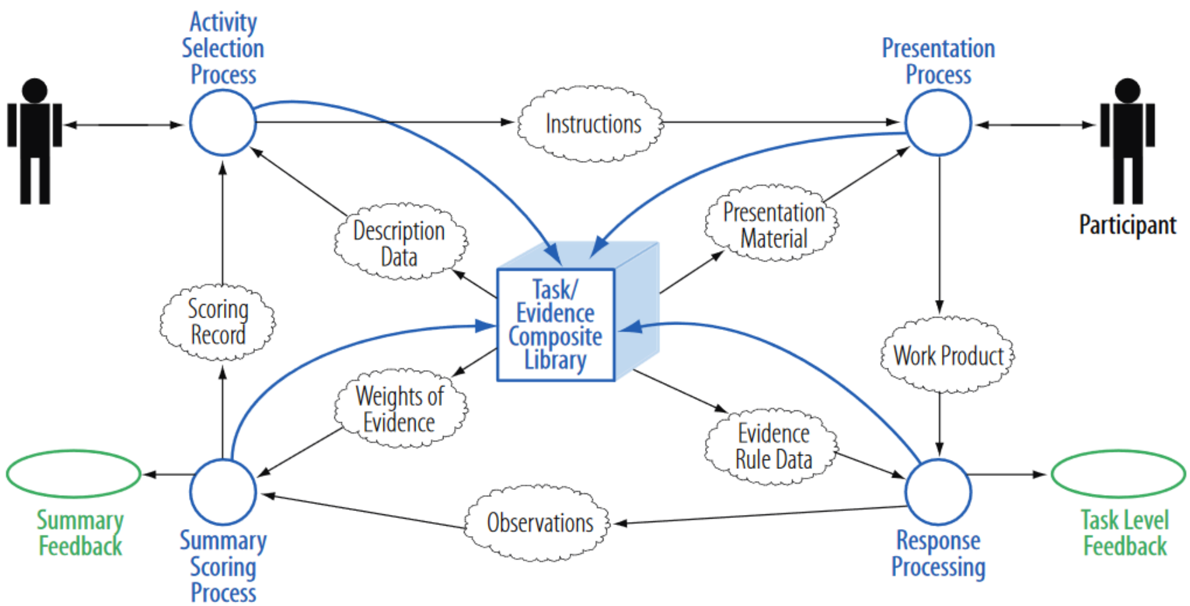


Figure 3.4: Overview of the Assessment Cycle, from [108]

Chapter 4

Research Setup

The remainder of this dissertation will discuss how we applied elements of the theoretical framework from Chapter 3 to two psychomotor domains. I first discuss the motivation for choosing each domain and then introduce how the work in each domain fits into the theoretical framework and the learning context.

4.1 Domain Selection

Psychomotor tasks are interesting to study because they are difficult to learn and are challenging to computationally assess. In this dissertation, I present work in two different psychomotor domains. The first domain considers drivers in an advanced vehicle simulator. Driving is a common task that most people have some experience with. In this task, drivers are given several competing objectives that are not always possible to maintain at the same time. Chapter 5 investigates how we can model learner behavior as they balance these objectives.

The driving domain we present here is a multi-objective optimization task where the driver follows a path to a goal state. In Chapter 6, we introduce a multi-objective optimization task with multiple phases; this is a drone landing task. The task requires pilots to maintain multiple concurrent safety constraints and accomplish two sequential phases. Most people do not have experience piloting drones, giving us more insight into the learning process.

Both domains presented here are safety critical, where persons and property are at risk in the event of a crash. In addition, both domains are generalizable to several different contexts; one might drive a vehicle for personal use or for delivering shipments while drones may be used for construction inspection tasks or search-and-rescue. Additionally, both driving and piloting require multiple control inputs, broadening the potential behavior space.

4.2 Modeling Learner Behavior in a Multi-Objective, Single Phase Optimization Task

4.2.1 Learning Context

In Chapter 5, we study drivers in a simulated vehicle with advanced driving features. In this domain, drivers navigated a nighttime course with tight corners and limited visibility. The drivers had multiple competing objectives: avoid hitting obstacles and maintain safe driving behaviors (e.g., staying in the center of lane, avoiding sharp turns). During the drive, participants encountered an obstacle in the road they needed to avoid hitting while maintaining the original objectives as much as possible. Each drive included four obstacles and drivers completed the drive at least three times.

4.2.2 Theoretical Framework Components

Task Attempt. We consider each encounter with an obstacle to be a task attempt. The activity traces include the vehicle's position and velocity, distance from the obstacle, distance from the center of the lane, and the driver's control inputs (steering angle and braking force). The activity history knowledge base records these activity traces as a time series over the course of each task attempt.

Assessment. Our work in this domain focuses primarily on the assessment module. Using the activity history, we developed a risk field model that learns driver-specific behavior parameters and saves them to the learner model. These parameters represent the driver's relative adherence to the competing driving objectives. Using this method, we can flexibly add or remove behavior parameters to simulate different levels of situation awareness and can accurately model the driver's future trajectory over long horizons.

Feedback. The work we present in this dissertation does not explicitly manipulate feedback for this domain. Participants only receive haptic feedback when they crash into the obstacle or run off the road, similar to what one would feel in a collision with another object. Because of this, we did not define a pedagogical policy for the system.

Task Generation. We do not change the driving task between attempts. This is because our goal was to combine repeated attempts at the same task to see how they differ. Since we did not generate new tasks, we did not define a curriculum policy for the system.

4.3 Generating Formative Control Feedback in a Multi-Objective, Multiple Phase Task Using Automated Assessment

4.3.1 Learning Context

In Chapter 6, we study pilots in two simulated quadcopter drone environments. In the first study (Section 6.1.2), pilots completed a smooth vertical takeoff, hovered in a floating ring for five seconds, and safely landed the drone in the original starting position using a dual-joystick controller. Participants completed this task five times. In the second study (Section 6.3), pilots landed a drone on a narrow landing platform. The pilots in this scenario had multiple objectives. Before landing, they needed to avoid crashing in the narrow flight area. To successfully land, they needed to keep the drone at a slow speed and shallow angle. Participants completed this task 20 times.

4.3.2 Theoretical Framework Components

Task Attempt. We consider each landing to be a task attempt. The activity traces include the drone's position, velocity, and tilt angle along with learner keyboard inputs to control the drone. The activity history knowledge base records these activity traces as a time series for each task attempt.

Assessment. Using the recorded activity history, we use temporal logic specifications to assess landing performance for several task components. Using the robustness metric, we evaluated how well the task attempt adhered to the specifications. The learner model stores the evaluated performance values at each time step of each task attempt.

Feedback. Using the robustness values stored in the learner model, the pedagogical policy selects which task component needs the most improvement. The system then creates natural language and image feedback using a generative AI prompt template that uses elements of effective formative feedback.

Task Generation. We do not change the landing task between attempts. This allowed us to focus on validating the assessment and feedback modules. We thus did not define a curriculum policy for the system.

Chapter 5

Learner Modeling in a Multi-Objective, Single Phase Optimization Task

In this chapter we explore an approach to learner assessment in the context of human drivers using vehicles with advanced driving capabilities. This work was originally published in [77, 76].

5.1 Introduction

We consider the problem of systematically modeling human control actions inside an intelligent transportation system. Ideally, such a model would enable interpretable explanations of why human drivers make certain control decisions in a given situation. Moreover, a model of driver decisions should be able to capture the variation in human driving behavior and emulate qualitatively different driving behaviors. Such models of human drivers can be quite helpful in developing autonomous vehicles that behave in a predictable manner and are able to operate on roads with human-driven vehicles [17, 167]. In particular, we focus on modeling driver situational awareness.

Situational awareness, as the name implies, refers to the perception by an agent of different aspects of their operating environment as well as knowledge of how these would affect their goals and overall performance (Cf. [51]) [1] . For instance, an agent driving a car may possess situational awareness of other cars that are in close proximity, so that they are aware of the positions, headings and velocities of these cars as well as whether a future collision with any of these cars may be imminent. Inferring the (lack of) situational awareness of an agent during task performance is a challenging problem.

[1] The terms "situational awareness" and "situation awareness" are used interchangeably. We will exclusively use the former here.

In this chapter, we first consider probabilistic models of human actions by building upon the concept of **artificial risk fields**. Such risk fields map states of the system to non-negative risk values, wherein larger risk values imply the state is close to a violation. The choice of a control action from a given state by the human operator follows from the risk model in a simple way: the probability that a given control action is chosen is proportional to the exponential of the risk at the state that is reached at a fixed **preview time** by applying that action. We develop this idea in the context of human control of a car wherein the human operator is tasked with driving the car safely along a road while staying in the designated lane, and at the same time, avoiding obstacles placed on the road. We first show how a family of possible risk functions can be formulated for such tasks, wherein each risk function is obtained by instantiating some unknown parameters to a specific values. We demonstrate how the risk function can yield a probability distribution over possible choices of control input that a human operator may select from a given state, assuming a fixed preview time. We also consider the problem of inferring risk functions from actual human operator data. In particular, we show that deriving maximum likelihood risk function parameters for a class of "additive" risk functions reduces to a convex optimization problem that can be solved to global optimum.

We evaluate the proposed framework on data collected from human drivers inside a simulated driving environment, wherein the humans are tasked to drive the vehicle along a fixed course while avoiding obstacles placed along the vehicle's path. Using data from six different drivers with up to four trials around the course for each driver, we show that our approach can fit parameters for risk models in each case. We explore the interpretation of these parameters showing how they predict qualitatively different behaviors. Next, we evaluate the ability of our model to predict future trajectories that are close to the ground truth trajectories. Here, we show that our model can provide very accurate predictions with errors that lie within a few meters for predicting the position 20 seconds out into the future. However, at the same time our model is less accurate for predicting how drivers accelerate or decelerate over different portions of their driving tasks.

The main contributions of the proposed framework are: (a) We formalize the risk field-based approach that has been proposed by many researchers in the past [142, 59, 81, 90]. A key contribution lies in formalizing the driver model based on a risk field as a stochastic model and providing approaches to discovering model parameters from naturalistic data. (b) We instantiate our framework to a driving simulator-based study of human operators driving a vehicle around in a simulated course with obstacles. (c) Our empirical evaluation shows that risk field-based approach can provide reasonable predictions of future trajectories. (d) Finally, we systematically vary risk field parameters to generate distinct driver behaviors.

Additionally, we extend this work to infer key aspects of the driver's situational awareness on the fly. Our approach relies on passive observations of the vehicle state and control inputs. For instance, our framework can predict the likelihood that the driver is aware of an obstacle in front of them by matching the driver's actions against two different hypotheses: one where the driver is **unaware** of the obstacle and the other where the driver is aware. We use the underlying probabilistic model to predict the likelihood of the driver's currently observed control inputs under each hypothesis. Therefore, using Bayes rule, we can then predict the probability that the driver is unaware of the obstacle in front of them, or more precisely, if the driver's actions are consistent with someone who is unaware under the assumed probabilistic model. We show that our approach can extend to other aspects such as ascribing a spatial position to the vehicle that is most compatible with the driver's current choice of control inputs under the assumed probabilistic model. Such a position could inform us about the driver's likely mental model of the vehicle state given their actions.

Our approach can be quite useful in many practical applications in Human Cyber-Physical Systems. Originating in the aviation domain [94], there has long been a focus on understanding operator behavior in uncertain or dynamic environments. In particular, pilots as well as other vehicle drivers need to be able to detect potentially dangerous situations so that they may react in a timely manner. For vehicle drivers, unsafe situations may arise due to a variety of factors such as fatigue during a long drive, a pedestrian suddenly entering the road, or when the autonomous vehicle fails to identify a stopped emergency vehicle [23].

The motivation for this work is to model driver situational awareness with the ultimate goal of providing interventions through shared controllers or user interfaces. In this chapter, we predict whether or not a driver has detected an imminent obstacle in the simulated driving task. We use a risk field modeling approach to infer the driver's mental state and their estimate of the distance to an upcoming obstacle. We show that this approach can distinguish between trials where drivers successfully avoid an obstacle and trials where drivers collide with the obstacle.

5.2 Related Work

Munir et al. [117] discuss the main challenges facing feedback control with human-in-the-loop; in particular, they discuss the need for developing systematic models of human behavior. Previous approaches to modeling driver behavior rely on cognitive models of human information processing. Salvucci and Gray [153] exploit the tendency of a driver's gaze to fixate on a near and far point. Subsequent work by Salvucci [152] used models of human declarative and procedural knowledge in the ACT-R cognitive architecture [3] to simulate steering angle and lateral position for navigating curves. Our work also models human operator control choices in a systematic manner. The key differences are two-fold: our model predicts a distribution over possible control inputs rather than a fixed prediction based on the state. Also, unlike the works mentioned above, we do not aim to model the mental processes that underlie the driver's decision making.

Other work captures driver behaviors in a qualitative manner. For example, Zhang et al [186] characterized drivers as novices or experts using a pattern recognizer on steering inputs. Similarly, Filev et al [54] used a rule-based system to classify drivers as cautious or aggressive based on the variation in their braking and acceleration behaviors. Finally, Wang et al [174] used k-means clustering to identify key characteristics of long-term driving behaviors such as prudence, stability, conflict proneness, and skillfulness. These approaches aim to develop driver profiles. Our approach can be interpreted similarly by examining the relative weights of the risk model components; we can additionally apply artificial risk fields as a generative model of future behavior.

Recent methods in modeling operator behavior are based on navigating "interaction fields" in the task environment [81]. Foundational work by Gibson et al [59] hypothesizes that humans navigate a "field of safe travel" by evaluating possible paths based on subjective experience and objective physical limitations. In the recent work of Kolekar et al [91], participants in a driving simulator were asked to react to obstacles placed at varying positions relative to their vehicle. Based on recorded reactions, the authors constructed a "driver's risk field" surrounding the vehicle. In a subsequent work [90], they then quantified a driver's perceived risk as the product of their risk field and the cost of certain events (colliding with obstacles). This leads to a controller which generates human-like behavior in a variety of scenarios when set to maintain risk under a certain threshold.

The motivation for developing a risk field framework is similar to the work of Kolekar et al [91, 90] in that we seek an interpretable and generative model of driving behavior grounded in theories of human reasoning and decision making. Our approach differs from the above work in several important respects. First, we define a risk field as a characteristic of the task environment and control inputs selected by the operator. The operator then stochastically navigates this risk space with the goal of minimizing risk. A second distinction is that because the risk fields presented here are defined in the task environment, they extend to other scenarios besides driving.

Our approach is closely related to **inverse reinforcement learning** where the vehicle model and operator's actions are captured by a Markov Decision Process (MDP) model with unknown reward functions. The goal is to infer these unknown rewards either through solving an optimization problem [119, 188] or through Bayesian methods [140]. There has been a long history of using inverse reinforcement to explain the actions of human operators inside a known environment [12]. The recent work of Ozkan et al studies how inverse reinforcement learning can be used to learn a driver model that is able to predict lead vehicle following behaviors of human drivers in a 3D driving simulation environment [124]. Our approach bears many similarities to inverse reinforcement learning: for instance, we can view risk fields as a (negative) "reward" function that the driver is minimizing. However, some key differences exist: we explicitly consider a "preview time" that the operator looks ahead into the future. This allows us to keep our risk functions simple since they apply to the state that is reached at some time in the future. Inverse reinforcement learning approaches compute rewards/risks that apply to the current state. This means that they have to consider more complicated functions than we do. As a result of our setup, we also have the benefit of solving a convex optimization problem and thus guarantee that we can compute the most likely model.

Rather than modeling the driver's risk perception and control choice, data-driven approaches such as Kim et al [84] and Long et al [181] train recurrent neural networks that input numerous features such as the vehicle's past trajectories and from its surrounding environment to predict the future trajectories of the vehicle. While these approaches are promising as predictors of future trajectories, they require a larger volume of data to reliably train and test a recurrent neural network. It is often challenging to interpret variations between drivers or in general understand these models once they are trained. Nevertheless, data driven approaches have proven more versatile and capable of handling many more situations than our approach in this chapter. Our work is currently aimed towards more narrowly defined settings although we hope to generalize it in future iterations of our framework to handle a richer variety of driving scenarios.

When dealing with driver safety and situational awareness, recent work has focused on takeover requests when an autonomous vehicle detects a possible collision or dangerous situation [86, 170, 125].

A few recent papers have developed models to predict situational awareness. In a review focused on situational awareness for connected cars, [60] discuss previous modeling approaches applied to the core stages of situational awareness: perception, comprehension, projection, and management. More recently, researchers have used advances in artificial intelligence to improve real-time prediction of situational awareness using eye tracking data [187, 67]. In each case, the operational definition of situational awareness varies, ranging from a composite of avoidance behaviors in a takeover situation [187] to a function of eye fixations in key areas such as vehicle instruments [67].

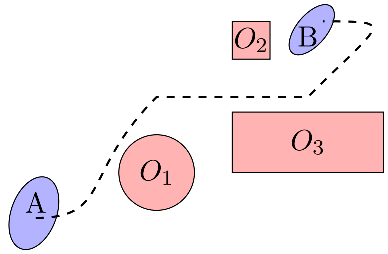


Figure 5.1: Schematic diagram of the overall task setup showing initial set of states A, target set B, obstacles O_i and optionally a reference path.

5.3 Artificial Risk Fields

We describe a general approach to defining human operator behavior using artificial risk fields. Subsequently, we will apply the framework to a driving task in Section 5.4.

5.3.1 Problem Formulation

We consider a single vehicle inside a known environment of **states** $\mathbf{x} \in X$. The human driver's task is to control the vehicle from a starting configuration A to a goal configuration B. Additionally, we designate a set of **obstacles** $O_1, \ldots, O_m$; each obstacle $O_i \subseteq X$ represents an unsafe configuration. It may also be natural to specify a "desired" path π that connects the start to the end state, that the vehicle should stay close to (Figure 5.1).

The vehicle is modeled by its **dynamics**: $\frac{d\mathbf{x}}{dt} = f(\mathbf{x}(t), \mathbf{u}(t))$, wherein $\mathbf{x}(t)$ models the state at some time t and $\mathbf{u}(t) \in U$ models the action (control input) at time t and U is the set of actions available to the human driver. The function f is assumed to be a fixed and known state update function. Our approach makes some key assumptions about the behavior of the human operator:

- The operator knows the state $\mathbf{x}$, or at least, those state variables involved in choosing the control.

- The operator model is Markovian — i.e, the probability distribution depends on the current state $\mathbf{x}$ and not necessarily on the path taken to reach the state.

Whereas the assumptions above are somewhat restrictive, we note that our goal is to build a model that predicts the operator's decision making rather than capture the mental processes involved in the decision making.

The overall goal of this chapter is to predict what control actions are likely to be chosen by the human operator at a given state. That is, we seek to model the probability distribution of $\mathbb{P}(\mathbf{u}|\mathbf{x})$ that an action $\mathbf{u}$ that is chosen by the human operator at state $\mathbf{x}$. Our model makes the following assumptions about the operator's control selection strategy:

- Each state $\mathbf{x} \in X$ is associated with a non-negative **risk value** $\mathsf{risk}(\mathbf{x}; \mathbf{p})$ which provides an aggregate numerical score, wherein $\mathbf{p}$ denotes a set of parameters that may be specific to an individual operator at a given time. The higher the risk score associated with a state, the closer it is to being a property violation such as entering a forbidden obstacle region or deviating too far from a desired path.

- The operator plans ahead to some "preview" time $\delta_p > 0$ into the future.

- The operator's decision making balances two factors: the risk of the future state that would be reached if a particular control were chosen against the magnitude of the control input. Thus, the operator would prefer not to apply extreme values of brakes/acceleration or steering inputs while at the same time they would prefer to stay away from obstacles and close to the center of their designated lane.

We will first describe each component of the model starting with the risk function. Next, we will describe how the overall probability distribution is defined.

5.3.2 Risk Function

The risk may be defined by many factors including the proximity of the state to various obstacles and the deviation of the state from the desired path π. The risk function $\mathsf{risk}(\mathbf{x}; \mathbf{p})$ is given by:

$$\sum_{j=1}^{m} p_j \mathsf{obstacleRisk}(\mathbf{x}, O_j) + p_{m+1} \mathsf{deviationRisk}(\mathbf{x}, \pi)\,,$$

wherein $\mathsf{obstacleRisk}(\mathbf{x}, O_j)$ is a function that measures the risk connected with the state $\mathbf{x}$ being inside (or close to) the obstacle O_j, and $\mathsf{deviationRisk}(\mathbf{x}, \pi)$ measures the risk arising from the state $\mathbf{x}$ being far way the reference path π (if one is given in the problem formulation). In general, any function that ensures that the risk is monotonically decreasing as one moves away from the obstacle can be chosen. Similarly, $\mathsf{deviationRisk}(\mathbf{x}, \pi)$ will be 0 if $\mathbf{x}$ lies on the reference path, and increases monotonically as the distance from the state $\mathbf{x}$ to the reference path π increases.

Finally, we note that the parameters $\mathbf{p} : (p_1, \ldots, p_{m+1})$ are non-negative weights that model the relative weightages associated with avoiding various obstacles and being close to the reference path. The choice of these parameters will affect the nature of the risk function. In Section 5.5, we demonstrate how parameters for a risk model are chosen given observed experimental data.

5.3.3 Overall Operator Model

The next component of the risk model concerns the assumption of a preview time. Let $\mathbf{x}$ be a current state and $\mathbf{u}$ be a control action under consideration. We assume that the operator computes the state $\mathbf{x}'$ at some fixed time $\delta_p > 0$ in the future. In other words, let $\mathbf{x}'(\mathbf{u}, \delta_p)$ be the state that results at time $t + \delta_p$ if the control action $\mathbf{u}$ were chosen at time t and held constant. Also, we associate a non-negative cost to each control action $\mathbf{u}$ denoted by $\mathsf{cost}(\mathbf{u}; \mathbf{q})$. Once again, the cost model can be parameterized by a set of unknown parameters $\mathbf{q}$ that will be estimated from experimental data.

The operator model we formulate assumes that

$$\mathbb{P}(\mathbf{u}|\mathbf{x}) \propto \exp(-\mathsf{risk}(\mathbf{x}'(\mathbf{u}, \delta_p); \mathbf{p}) - \mathsf{cost}(\mathbf{u}; \mathbf{q}))\,.$$

Suppose the set of possible actions U is a finite set $\{\mathbf{u}_1, \ldots, \mathbf{u}_N\}$, then we write the exact expression as Eq. (5.1). The denominator normalizes the probability over all actions. For continuous set of control actions, we can replace the summation by an integral over the set U. Doing so, we obtain the following expression for $\mathbb{P}(\mathbf{u}|\mathbf{x})$:

$$\frac{\exp(-\mathsf{risk}(\mathbf{x}'(\mathbf{u},\delta_p);\mathbf{p}) - \mathsf{cost}(\mathbf{u};\mathbf{q}))}{\sum_{j=1}^{N}\exp(-\mathsf{risk}(\mathbf{x}'(\mathbf{u}_j,\delta_p);\mathbf{p}) - \mathsf{cost}(\mathbf{u}_j;\mathbf{q}))}\,. \tag{5.1}$$

The operator model implicitly assumes that (a) the operator can forecast a future state $\mathbf{x}'(\mathbf{u})$ some time δ_p in the future as a result of a control input $\mathbf{u}$; and (b) chooses control actions which yield future states with lower risk + cost values preferentially over those with higher risk + cost values.

5.4 Driving Task

We describe the driving task that will be the central case-study to motivate our work and develop a risk field model specific to the task.

5.4.1 Task Description

The driving task is performed in a medium-fidelity driving simulation environment developed by the National Advanced Driving Simulator (NADS miniSim) at Purdue University [172]. The system includes three high resolution monitors for displaying the driving environment and a smaller monitor for the vehicle dashboard display. The user controls a steering wheel and foot pedals for acceleration and braking as in a standard automobile (Figure 5.2, left).

Driving Scenario. The driving scenario consists of driving the simulated vehicle at night time on a two lane city highway with four obstacles placed along the route. Illumination using street lights was present. The overall simulated driving course distance was roughly 4.8 km (3 miles). To increase the difficulty of the task, participants were asked to drive one handed with their non dominant hand. There were no oncoming, leading, or trailing vehicles. The obstacles were placed so that they were visible only after the participant rounded the curve (Figure 5.2, right).

The objectives for the human driver are as follows:

- The operator must practice safe driving by keeping within their lane and minimizing deviations. They must never exit the paved road.

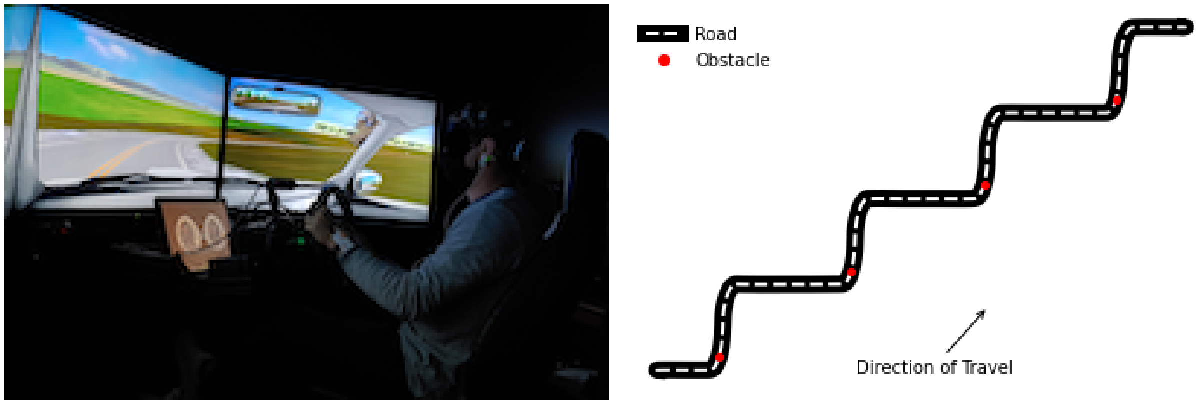


Figure 5.2: (Left) Picture of the NADS miniSim setup showing a participant driving along a course (daytime simulation), (Right) plot of the centerline of the simulated course showing obstacle placement as red circles.

- Obstacles (a tire) placed in the operator's lane are to be avoided.
- Vehicle speed is to be maintained as close as possible to 45 mph ($\approx$ 20 m/s) at all times.

Participants. The study was conducted with six participants (3 male, 3 female) with a mean age of 21.33 years (SD = 0.82). Participants were all undergraduate students at Purdue University, and were all engineering senior undergraduate students. On average, the participants had 4.2 years of driving experience, with all of them reporting having driven 10K or more miles per year, on average. The participants were allowed to practice driving the vehicle on the simulator using a daytime practice course that involved an open highway.

Data Collection. Each participant drove the course over three (or in one case, four) separate trials, yielding nineteen separate trials for the six participants, in total. Data collected includes the position, velocity, heading angle, steering wheel position, accelerator/brake pedal positions sampled at 60 Hz.

5.4.2 Risk Field Formulation

We will now derive risk models for the human driving task. First, we will describe a simple unicycle model for the vehicle's dynamics. This model is appropriate since effects such as cornering over tight turns, wheel slip and skids are not important for the speed and road conditions that were simulated in the study. The state of the vehicle is described by $\mathbf{x} : (x, y, v, \psi)$, wherein x, y denote the position in a fixed coordinate frame, v describes the velocity of the vehicle and ψ is the heading angle. The control inputs are u_1: the acceleration (or deceleration) and u_2: the turning rate. The dynamics are described by the ODEs:

$$\left.\begin{array}{llll} \dot{x} & = v\cos(\psi) & \dot{y} & = v\sin(\psi) \\ \dot{v} & = u_1 & \dot{\psi} & = u_2 \end{array}\right\} \tag{5.2}$$

We define the function $\mathsf{ptLineDistance}((x, y), C)$ as the Euclidean distance from a given position (x, y) to the nearest point in the center-line C.

Similarly, we are given a list of obstacle positions $O : [(x_{o,1}, y_{o,1}), \cdots (x_{o,4}, y_{o,4})]$. Each obstacle has a fixed diameter $d_o = 0.3$ meters. We define the function $\mathsf{obstacleDistance}((x, y), O)$ as the Euclidean distance from a given position (x, y) to the obstacle that will be encountered next in the vehicle's direction of travel.

The overall risk for a given state $\mathbf{x} : (x, y, v, \psi)$ and control $\mathbf{u}$ is given by:

$$\mathsf{risk}(\mathbf{x}) : \begin{cases} \mathbf{A} \cdot \mathsf{ptLineDistance}((x, y), C)^2 + \\ \mathbf{B} \cdot \exp\left(-\frac{\mathsf{obstacleDistance}((x,y),O)^2}{d_o^2}\right) + \\ \mathbf{C} \cdot (v - v_{tgt})^2 \end{cases} \tag{5.3}$$

and the cost of the control input is given by $\mathsf{cost}(\mathbf{u})$:

$$\mathsf{cost}(\mathbf{u}) : \ \mathbf{D} \cdot u_1^2 + \mathbf{E} \cdot u_2^2 \tag{5.4}$$

Here $\mathbf{A}, \ldots, \mathbf{E} \geq 0$ are unknown parameters whose values will determine the actual tradeoffs that the driver makes while staying in their lane and avoiding the obstacles during the execution of the task.

We consider control inputs $u_1 \in \{-1, -0.9, \cdots, 0.9, 1\}$ (units are $\mathtt{m/s^2}$) and $u_2 \in \{-0.5, -0.45, \cdots, 0.45, 0.5\}$ (units are $\mathtt{radians/s}$), yielding 400 discrete choices for (u_1, u_2). For a given state $\mathbf{x}$, the probability of control inputs (u_1, u_2) being chosen $\mathbb{P}((u_1, u_2) \mid \mathbf{x})$ is described

Data: $\mathsf{risk}, \mathsf{cost}$: risk/cost functions, $\mathbf{x}_0$: Initial State, δ_p: preview time, δ: time step, n_s: number of simulation steps, $U : \{\mathbf{u}_1, \ldots, \mathbf{u}_N\}$ all control inputs
Result: Sample Trajectory: $\mathbf{x}(0), \ldots, \mathbf{x}(n_s\delta)$

```
x(0) ← x_0;
for s ← 1, ··· , n_s do
    for each u_j ∈ U do
        /* Simulate until the preview time.                         */
        x'_j ← next(x(δ(s − 1)), u_j, δ_p);
        /* Calculate Risk.                                          */
        p(u_j) ← exp(−risk(x'_j; A, B, C) − cost(u_j; D, E));
    end
    sample u ∈ U with probability p(u)/ Σ_{k=1}^{N} p(u_k);
    x(δs) ← next(x, u, δ) ;                           /* State for δs */
end
```

Algorithm 1: Algorithm for sampling a trajectory given risk and cost functions, initial states.

once again by Eq. (5.1). Here we define the risk and costs by Eqs. (5.3) and (5.4). The next state $\mathbf{x}'(\mathbf{u}, \delta_p)$ is obtained by simulating the ODE in Eq. (5.2).

Algorithm 1 shows the overall algorithm for sampling a trajectory from the risk model.

5.5 Model Fitting

5.5.1 Maximum Likelihood Estimation

In this section, we consider how to infer a risk field given data in the form of states $\mathbf{x}(t)$ and controls $\mathbf{u}(t)$. We will assume that the risks and costs are **additive** over component functions as follows:

$$\mathsf{risk}(\mathbf{x}; \mathbf{p}) : \sum_{j=1}^{m} p_j f_j(\mathbf{x}), \quad \mathsf{cost}(\mathbf{u}; \mathbf{q}) : \sum_{i=1}^{l} q_i g_i(\mathbf{u})\,. \tag{5.5}$$

Note however, that we do not assume much for functions f_j, g_i other than that they are non-negative and well-defined over the relevant values of $\mathbf{x}, \mathbf{u}$. The parameters for risk and cost functions are collected as a vector $(p_1, \ldots, p_m, q_1, \ldots, q_l)$. Assuming that the controls are chosen from a finite set $U : \{\mathbf{u}_1, \ldots, \mathbf{u}_N\}$, fixing δ_p to be the preview time and $\mathsf{next}(\mathbf{x}, \mathbf{u}, \delta_p)$ being the state reached starting from current state $\mathbf{x}$ if control $\mathbf{u}$ is applied for time δ_p. Recall that the model chooses a control input $\mathbf{u}$ for a state $\mathbf{x}$ in proportion to the risk and cost according to Eq. (5.1).

Let us assume that we are given driving data of the form $(\mathbf{x}(t_i), \mathbf{u}(t_i))$ consisting of states and controls applied at various times t_i for $i = 1, \ldots, M$. Our goal is to find risk parameters $\mathbf{p}, \mathbf{q}$

$$\log \mathbb{P}(\mathbf{u}|\mathbf{x}) = \begin{array}{l} -\sum_{j=1}^{m} p_j f_j(\mathbf{x}'(\mathbf{u})) - \sum_{i=1}^{l} q_i g_i(\mathbf{u}) \\ -\log\left(\sum_{k=1}^{N} \exp\left(-\sum_{j=1}^{m} p_j f_j(\mathbf{x}'(\mathbf{u}_k)) - \sum_{i=1}^{l} q_i g_i(\mathbf{u}_k)\right)\right) \end{array} . \tag{5.6}$$

for Eq. (5.5) that maximizes the overall log-likelihood $\mathcal{L}(\mathbf{p}, \mathbf{q}) : \sum_{i=1}^{M} \log \mathbb{P}(\mathbf{u}(t_i)|\mathbf{x}(t_i))$, wherein $\mathbb{P}(\mathbf{u}(t_i)|\mathbf{x}(t_i))$ is as given in Eq. (5.1).

Note that if the risk and cost models are additive as in Eq. (5.5), then the overall log-likelihood $\mathcal{L}(\mathbf{p}, \mathbf{q})$ is a **concave function** for a fixed value of δ_p. This means that we can solve the maximization problem of a concave function (or alternatively minimization of a convex function) to obtain a global optimum using standard off-the-shelf convex optimization tools[24].

Theorem 1. *If the risk and cost models are additive as in Eq.* (5.5)*, then the overall log-likelihood* $\mathcal{L}(\mathbf{p}, \mathbf{q})$ *is a **concave function** for a fixed value of* δ_p*.*

Proof. The proof consists in observing that $\log \mathbb{P}(\mathbf{u}|\mathbf{x})$ is concave function of $\mathbf{p}, \mathbf{q}$. Let $\mathbf{x}'(\mathbf{u})$ denote the value of $\mathsf{next}(\mathbf{x}, \mathbf{u}, \delta_p)$. Expanding Eq. (5.1) using the form of the risk model in (5.5), we obtain an expression for $\log \mathbb{P}(\mathbf{u}|\mathbf{x})$ in Eq. (5.6)

Since $\mathbf{x}, \mathbf{u}$ are given to us in the data, the terms $f_j(\mathbf{x}'(\mathbf{u}))$ and $g_i(\mathbf{u})$ are all fixed constants. Thus, as a function of $\mathbf{p}, \mathbf{q}$, we note that $\log \mathbb{P}(\mathbf{u}|\mathbf{x})$ is the difference of a linear function over $\mathbf{p}, \mathbf{q}$ and the log-sum-exp of linear function over $\mathbf{p}, \mathbf{q}$. This is a difference of a concave function and a convex function, which is itself concave.The overall likelihood is the sum of concave functions, and is concave. □

5.5.2 Fitting Parameters From Obstacle Avoidance Data

In this section, we report on the application of the maximum likelihood minimization approach to the data obtained from six human drivers in the NADS vehicle simulator, as described in Section 5.4.

We recall that each participant drove along a road with obstacles placed at periodic intervals. In particular, each "trial" by a participant involved four encounters with the obstacle. We will fit the risk model parameters using the data from each obstacle, using the `scipy.optimize` module

Quantile	A	B	C	D	E
5%	0.248	0.000	0.000	0.000	14.233
50%	0.544	16.349	0.000	1.416	40.782
95%	0.939	110.864	0.025	11.827	99.543

Table 5.1: Distribution of parameters fit around one obstacle, using the best preview time δ_p. There were 76 total fit models.

for various values of $\delta_p \in \{0.6, 0.8, 1.0, 1.2\}$ seconds. The risk functions used are described in Section 5.4.2 and in particular Eqs. (5.3) and (5.4). This yielded 19 trials $\times$ 4 obstacles $=$ 76 fit models for each of the four δ_p values.

For each obstacle encounter, we selected the preview time δ_p which maximizes the likelihood of the data. Of the four δ_p values considered, 94.7% of fitted models achieved a maximum likelihood using $\delta_p = 1.2$ seconds. Thus, for a car driven at $20m/s$, the preview distance is 24 meters.

Table 5.1 shows the distribution of fit parameters as the median value as well as extremely low (5th percentile) and extremely high (95th percentile) values. We see that each parameter takes on a different range of values, with the parameter C (associated with staying close to the target velocity of $20m/s$) varying very little ($0 - 0.025$), whereas B (associated with the weightage placed on obstacle avoidance) encompasses a wide range ($0 - 110.86$).

5.6 Evaluating Driver Models

We first provide a preliminary analysis to evaluate the accuracy of our method for predicting driver trajectories. Of the 19 initial recorded course trials, we removed any trial where the driver collided with an obstacle, yielding 17 successful trials. For each successful trial, we fit risk field parameters using the formulation in Section 5.4.2 and the driver data from the first two obstacles in the trial. Using these parameters, we used Algorithm 1 to generate 100 trajectories for the held out data of the last two obstacles in the trial. We used a preview time $\delta_p = 1.2$ for all of the trajectories based on the analysis from Section 5.5.2.

To define a single trajectory for comparison with the actual driver behavior, we took the

	1s	2s	5s	10s	20s
min	0.002	0.051	0.002	0.180	0.104
median	0.064	0.234	0.493	1.114	0.764
max	0.245	0.944	1.655	2.373	1.482

Table 5.2: Deviation (meters) of generated trajectory from actual human trajectory, across all successful course trials. Results are reported at different times from the starting position (seconds)

median x and y value over the 100 trajectories at each time point. We then defined the divergence of the generated trajectory as the distance from the median point to the line created by the human trajectory. Table 5.2 shows the minimum, median, and maximum divergence from the human trajectory at times $t \in \{1, 2, 5, 10, 20\}$ seconds from the initial position.

Table 5.2 shows that, like one would intuitively expect, the deviation from the actual human trajectory increases over time, except between 10 and 20 seconds. The change in deviations may be a function of the course characteristics (e.g., rounding a turn) and also show that our model can self-correct based on the high-level priorities defined in the risk field model. Additionally, these results show that the model is able to predict the future position 20 seconds ahead with an error of less than 3 meters. This is promising, given that the lane width in the driving task was 3 meters.

Figure 5.3 shows sample (x, y) trajectories predicted by our model and velocities over time for three separate initial conditions drawn from the actual driver data. We also plot the actual "ground-truth" data for each of these situations. It is interesting to see that the simulated (x, y) trajectories are viable trajectories that keep close to the center line while avoiding obstacles. In the bottom row of Figure 5.3, we see that the predicted velocity deviates from the true human driver velocity by as much as 4 m/s, especially in cases where the driver accelerates swiftly. The mean absolute difference in predicted velocity versus actual velocities are 0.1 m/s for predicting 1 seconds out into the future, 1.3 m/s for 10 second prediction horizons and 2.5 m/s for 20 second horizon. However, we also observe that our model has the tendency to under-estimate the actual velocity around sharp turns: it is likely that the driver allows the vehicle to move towards the edge of their lane to reduce steering effort and allow themselves to accelerate. We conclude that the participants do not prioritize the instruction to maintain their velocity around 20 m/s, while focusing more on maintaining their lane position and avoiding obstacles. Modeling their choice of velocities requires considerations that are subtly different from their perceived risk such as their self-confidence.

5.7 Characterizing Driver Behavior

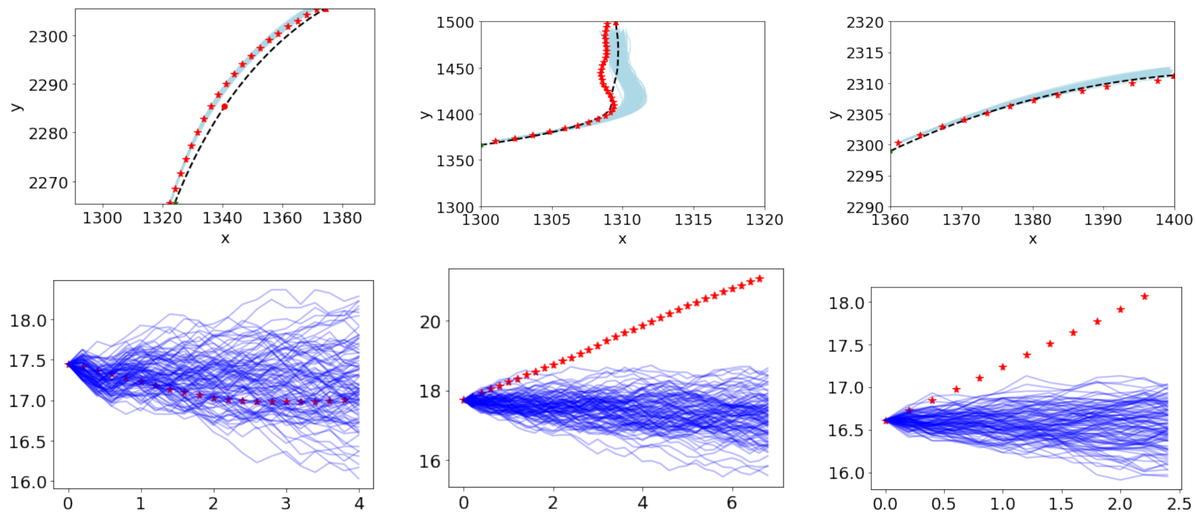


Figure 5.3: (Top Row) Sample (x, y) trajectories generated by the risk model against ground truth shown by red stars with centerline shown as a dashed black line and obstacle shown as red circle. **Warning:** x and y axes are drawn to different scales. (Bottom Row) Corresponding velocity (m/s) values over time against ground truth.

Showing the overall accuracy of our model, we reach the main research question, do the risk model parameters account for different types of obstacle avoidance behavior? To answer this, we will visualize generated trajectories using different parameter configurations. For each condition, we used Algorithm 1 to generate 20 trajectories around the course segment for the first obstacle. Our baseline comparison uses the median value for each parameter when calculating the risk field. To simulate the condition using the low and high values of a parameter, we used the 5th and 95th percentiles of the parameter, respectively, leaving the remainder of the parameters at their median level (see Table 5.1). We used a preview time of $\delta_p = 1.2$ as in the previous section.

Figure 5.4 shows the differences as we vary parameters from the baseline. In 5.4a we see that high values for A lead to trajectories very close to the centerline while low values for A stray farther from the centerline. While 5.4b shows more consistent trajectories between conditions, higher values for B leave a higher margin when passing the obstacle compared to the lower value condition.

Since parameter D impacts user controls, we visualize the acceleration under the different conditions rather than the physical position. In Figure 5.4c we see that high values of D lead to much more consistent acceleration compared to the low values. For parameter E, in 5.4d low value trajectories show a sharper decrease in steering rate after 5 time steps compared to the baseline and high level trajectories.

Overall, we conclude that varying the risk model parameters has the expected change in the trajectories. For instance, increasing B causes the trajectories to clear the obstacle with a much larger safety margin. Increasing A on the other hand has the opposite effect of bringing trajectories closer to the centerline.

5.8 Modeling Situational Awareness

In this section, we will use the previously described driver model to reason about the possible situational awareness of the driver. Situational awareness refers to perception of key aspects of the environment that will be critical for decision making on the part of the driver. Specifically, we will capture the probability that the driver's action indicate that they are aware of the obstacle in front of them. Similarly, we will use the driver's action to ascribe a "mental estimate" of the distance to the obstacle.

Consider a vehicle state $\mathbf{x} : (x, y, v, \psi)$ with an obstacle O at some distance $\mathsf{dist}((x, y), O)$ from the vehicle. Let us consider two alternative mental states: AWARE: the driver is **aware** of the obstacle in front of them, versus UNAWARE: the driver is **unaware** of the obstacle in front of them. The key difference lies in the **perceived** risk in these states. If the driver is unaware of an obstacle the risk model will not include the term associated with the obstacle, or in other words $\mathrm{dist}((x, y), O)$ is taken to be ∞ in Eq. (5.3). Let $\mathsf{risk}_{\mathrm{un}}(\mathbf{x})$ denote the risk associated with state

$\mathbf{x}$ assuming that the driver is unaware of the obstacle. This is equivalent to setting the distance $\mathsf{dist}((x, y), O) = \infty$ (or alternatively, $B = 0$) in Eq. (5.3). Note that when the driver is aware of the obstacle, $\mathsf{risk}(\mathbf{x})$ according to Eq. (5.3) will continue to model the risk associated with a state $\mathbf{x}$.

Thus, we define the probability:

$$\mathbb{P}(\mathbf{u}|\mathbf{x}, \text{UNAWARE}) \propto \exp\left(-\mathsf{risk}_{\mathrm{un}}(\mathsf{next}(\mathbf{x}, \mathbf{u}, \delta)) - \mathsf{cost}(\mathbf{u})\right) \tag{5.7}$$

whereas the probability of control choice when the driver is aware is given by Eq. (5.1), recalled and simplified below:

$$\mathbb{P}(\mathbf{u}|\mathbf{x}, \text{AWARE}) \propto \exp\left(-\mathsf{risk}(\mathsf{next}(\mathbf{x}, \mathbf{u}, \delta)) - \mathsf{cost}(\mathbf{u})\right) \tag{5.8}$$

The difference lies in the use of risk function as opposed to the risk_{un} function. Suppose we have a **prior belief** that the driver is **unaware** of the obstacle with probability p_U, then by Bayes rule, we obtain the following expression for $\mathbb{P}(\text{UNAWARE}|\mathbf{u}, \mathbf{x})$:

$$\frac{\mathbb{P}(\mathbf{u}|\mathbf{x}, \text{UNAWARE}) \times p_U}{\mathbb{P}(\mathbf{u}|\mathbf{x}, \text{UNAWARE}) \times p_U + \mathbb{P}(\mathbf{u}|\mathbf{x}, \text{AWARE}) \times (1 - p_U)}\,. \tag{5.9}$$

This allows us to provide a **recursive** estimate of the probability that the driver remains unaware of the obstacle in front of them. We initialize the probability of being unaware to some suitable starting value eg., $p_U = 0.5$. At each step, we obtain a state $\mathbf{x}$ and a control input $\mathbf{u}$ from the data. We use this to update the posterior probability according to Eq. (5.9). This provides us the prior distribution for the next time step. Often however, when p_U is close to 0 or 1, the recursive process stops evolving when new data is available. To avoid this, we use an "ϵ-transition" wherein the posterior value of p_U is updated as $p'_U = (1-\epsilon)p_U + \frac{\epsilon}{2}$ to yield a prior value for the next time step. We set $\epsilon = 0.05$ for our experiments.

Thus, we can obtain an estimate of the probability that the user is unaware of the obstacle at each time step. Next, we can refine our analysis to ask other questions about the situational awareness of the driver. For instance, we can use the risk model to infer the driver's likely estimate of their own position $(\hat{x}, \hat{y})$. To do so, we set up a prior distribution over likely positions $\pi(\hat{x}, \hat{y})$.

Typically such a prior is specified as a uniform distribution over positions that are within some distance of their true position. Given the vehicle state $\mathbf{x} : (x, y, v, \psi)$, let $\hat{\mathbf{x}}$ denote the state $(\hat{x}, \hat{y}, v, \psi)$. Furthermore, for simplicity let us consider a finite set of hypothesized mental model positions $\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_K$. Our risk model allows us to evaluate $\mathbb{P}(\mathbf{u}|\hat{\mathbf{x}}_j)$ for a given control input $\mathbf{u}$ and position $\hat{\mathbf{x}}_j$. Once again using Bayes rule we obtain:

$$\mathbb{P}(\hat{\mathbf{x}}_j|\mathbf{u}) = \frac{\mathbb{P}(\mathbf{u}|\hat{\mathbf{x}}_j) \times \pi(x_j, y_j)}{\sum_{k=1}^{K} \mathbb{P}(\mathbf{u}|\hat{\mathbf{x}}_k) \times \pi(x_k, y_k)}. \tag{5.10}$$

5.9 Situational Awareness Results

Figure 5.5 shows some of the results obtained by our approach on actual encounters of various drivers with different obstacles in the course. First, our risk model parameters $A - E$ are simply fixed to the mean values shown in Table 5.1. Next, we plot the probability $\mathbb{P}(\text{UNAWARE}|\mathbf{x}, \mathbf{u})$, having initialized it to 0.5 at the very beginning of each obstacle encounter. Figure 5.5 shows four different scenarios labeled **(a)-(d)**. Scenario **(a)** represents the vehicle colliding with the obstacle. Notice that the probability that the user is unaware of the obstacle rapidly rises from 0.5 to 1.0, about 1 second prior to the collision. We contrast that with Fig. 5.5 **(b)** wherein the obstacle is successfully avoided. As expected, the estimated probability rapidly falls from 0.5 to below 0.1 nearly 1 second prior to the vehicle passing the obstacle. Fig. 5.5 **(c)** also shows a successful obstacle avoidance that is achieved by deviating from the center line much closer to the obstacle when compared to Fig. 5.5 **(b)**. As expected, we note that the probability that the user is unaware falls rapidly but also rises back up. Finally, Fig. 5.5 **(d)** shows a situation where the driver approaches very close to the obstacle without necessarily colliding with it. Our approach estimates that the probability of being unaware of the obstacle rises rapidly.

Figure 5.6 plots the average of the driver's own estimate of their position $(\hat{x}, \hat{y})$ as inferred by comparing the chosen control input against the risk model versus the actual ground truth position. Figure 5.6(b)-(d) show cases where the obstacle is avoided whereas Figure 5.6(a) shows the case when collision with obstacle occurs. As expected, for the cases when a collision is avoided successfully, the estimated positions seem to coincide with the actual positions. A marked difference is observed in Figure 5.6(a) where a collision occurs. We interpret this result to mean that the driver's behavior in this case does not match what one would expect from the risk model. As a result, the $(\hat{x}, \hat{y})$ position wherein the driver's control inputs would make "most sense" are farther away from the vehicle's current position. However, for Figures 5.6(b)-(d), this is less true – in general, the decisions made by the driver seem more or less consistent with what would be expected at that position under the assumed risk model.

5.10 Discussion and Future Work

In this chapter, we have presented an approach to model control choices of the human driver by quantifying the risk and showing how the risk model can be inferred from data. We have also demonstrated our approach on actual human driving data from a medium-fidelity simulation environment showing that our models can accurately predict future positions and generate qualitatively different driving behaviors. In particular, we show that deviation of generated trajectories from the human trajectory remains relatively stable over time periods up to 20 seconds into the future.

The main area for improvement is that our model currently does not capture how human operators control the velocity. We plan to improve this aspect of our model in our future work. For example, we can consider more complex representations of risk and cost beyond the simple quadratic model presented here. Additionally, a driver's choice of velocity may depend on other factors such as their confidence in driving or the overall level of risk of the current situation. The fact that our models had very small values for the C parameter that measures velocity deviations from intended target indicates that human driver behavior during the task may have been influenced by factors different from risk. While our model was defined to maintain a predefined velocity as stated in the task instructions, we observed that the drivers themselves did not adhere to this requirement.

The main result of this chapter shows that by using this risk model framework with simple models for risk and control cost, we are able to generate distinct driver behaviors such as obstacle avoidance and keeping to the center of the lane. Using real driver data collected in a simulation environment, we have also shown that we can extract unique parameters that characterize individual driver behavior. Future work should investigate how accurately these models track more complex human behavior over time. Additionally, this model can be used as part of a predictive run-time monitoring system, where the goal is to predict impending violations of safety property (i.e., colliding with an obstacle) ahead of time. This system could be integrated into future driver safety interfaces and be used to study potential handover protocols with autonomous driving subsystems.

At a general level, this framework can be used to model a variety of scenarios and adapted to test hypotheses about human operator behavior. As discussed above, we noted that the participants in the study did not maintain the target velocity given in the task instructions. When fitting the risk model parameters, this behavior was indicated by the fact that the fitted values for parameter C were heavily skewed to 0, indicating no effect. Future work can systematically test different forms of the risk function to see which is a better fit for human behavior. This framework may also be adapted in an attempt to infer the human's true reward function during the driving task.

We also presented an extension of the risk field framework that constructs a probabilistic model of human decision making in dynamic environments wherein our extension allows such models to reason about key situational awareness properties of the user. The approach often produces results that are consistent with the ground truth data.

Key limitations of our data collection methodology include the limited number of participants and the straightforward nature of the driving task in our initial study. Some of these limitations are being addressed by collaborators in ongoing studies at the time of writing that will explore a larger pool of participants and more dynamic driving scenarios involving traffic patterns, wind, visibility restrictions, moving obstacles on the road and construction.

The data collected did not include ground truth data about the actual situational awareness of the drivers. Note that ground truth data about situational awareness is hard to collect, especially since we are interested in detecting the lack of situational awareness. In the future, we propose to correlate our approach with indirect measures such as gaze tracking data or more direct user reports of their ongoing situational awareness in the future.

This chapter contributes to the key results of the thesis by exploring a novel framework to model the driver's internal states such as individual risk behaviors and situational awareness. Compared to machine-learning based learner modeling methods, the proposed risk framework provides several benefits: (1) we can learn accurate models using substantially less data, on the order of tens of obstacle approaches per person; (2) model components are interpretable and easy to interchange; and (3) we can systematically test theories of learners' internal states and decision processes by comparing different structures of the risk model framework. Future work will explore how this framework can apply to a variety of interaction tasks. For example, we plan to address dynamic scenarios involving multiple agents and include more complex task requirements in our framework.

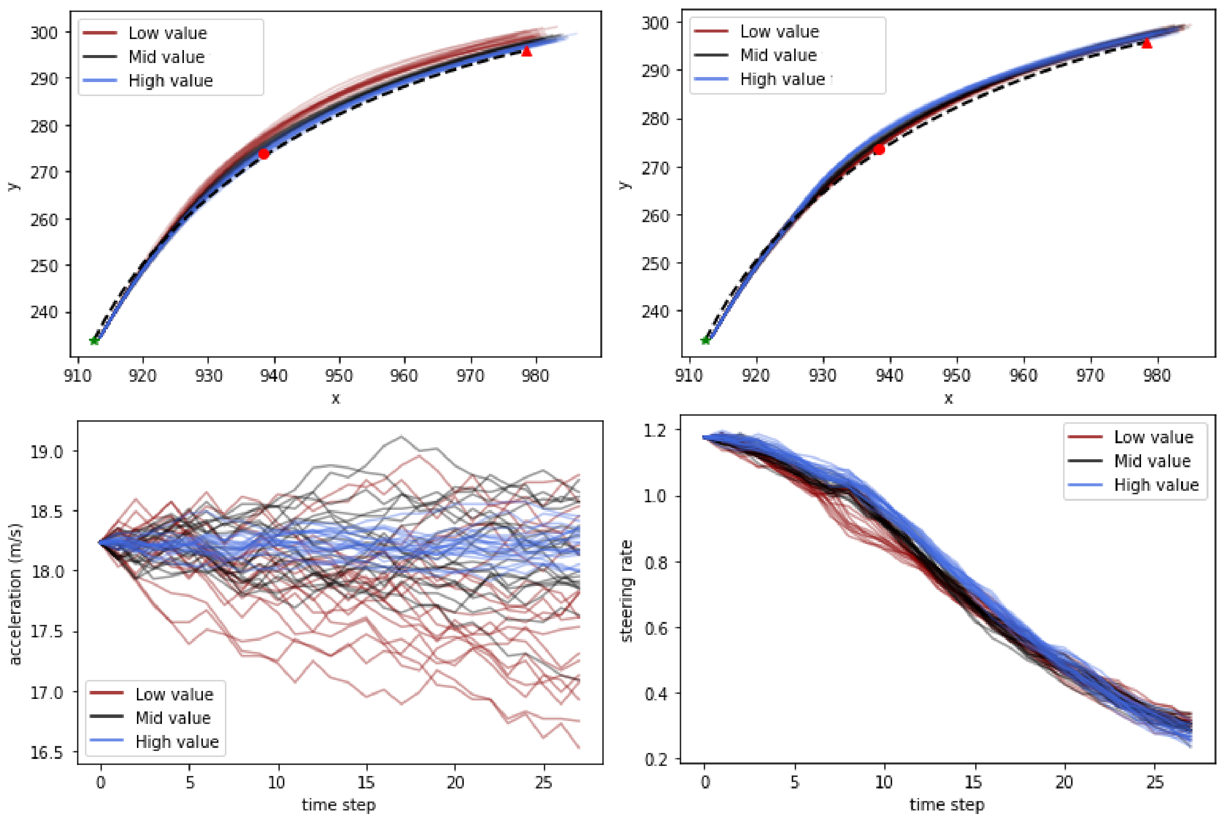
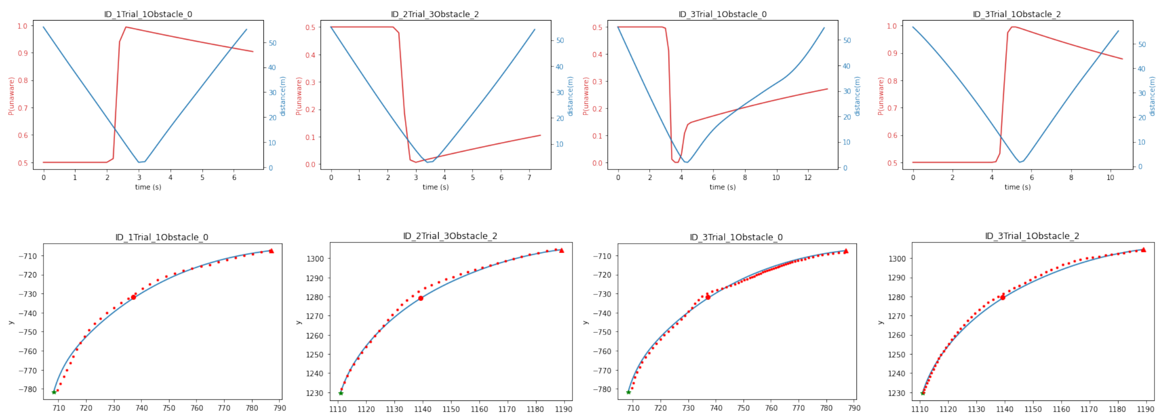


Figure 5.4: Simulated trajectories using extreme low (5^{th} percentile) and high values (95^{th} percentile) for the parameters. From left to right, top to bottom; (a) parameter A: center line deviation risk weightage; (b) parameter B: obstacle avoidance risk weightage; (c) parameter D: cost for acceleration; (d) parameter E: cost for turning rate control.



(a) **(b)** **(c)** **(d)**

Figure 5.5: **Top Row:** Combined plot of probability $\mathbb{P}(\textsc{unaware}|\mathbf{x}, \mathbf{u})$ that the driver is unaware of the obstacle in front (red) and distance from obstacle (blue); **Bottom Row:** Plot of vehicle trajectory (shown as dotted red line) against the center line shown in solid blue and obstacle shown as a filled red circle. **(a)-(d)** represent four selected scenarios each involving a different participant, trial and obstacle in the course.

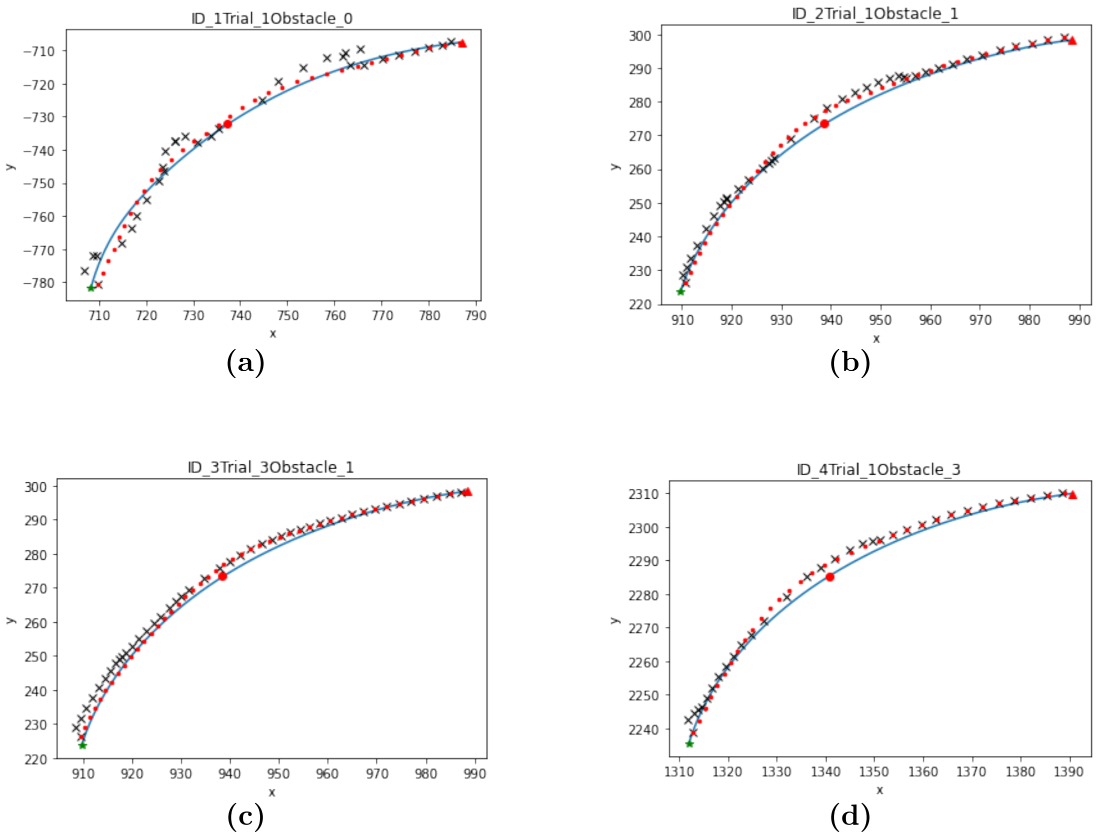


Figure 5.6: Mean over estimated driver positions shown as black x versus actual positions shown using red dots. The center line is shown in blue and obstacle is shown as a bright red circle.

Chapter 6

Generating Formative Control Feedback in a Multi-Objective, Multiple Phase Task Using Automated Assessment

The work in Section 6.1 was originally published in [73] and the work in Section 6.2 was originally presented in [79]. These sections describe preliminary work in automated assessment in feedback. I then present a full experimental study incorporating these ideas in Section 6.3.

6.1 A Framework for Automated Assessment for Multi-Objective and Multi-Phase Tasks

Skill is a highly valued attribute for numerous human endeavors. Thus, its measurement is of great importance. The need for humans to co-operate with an autonomous system to skillfully complete safety-critical tasks is common across diverse domains such as surgery, planetary exploration, and visual inspection using drones.

The problem of measuring skill is well-known to be extremely hard. Existing approaches use examinations of an operator by qualified judges who provide numerical scores. This process is often effort intensive, subject to bias and hard to automate. As a result, the numerical scores for different operators who undergo different exams judged by a different panel of judges are hard to compare against each other. The same problem arises when it comes to judging the same operator at different points in time to measure their learning progress. Thus, we need a framework for quantifying skill that is based on simple principles that can be applied uniformly in an unbiased manner. Also, the idea of a single number representing skill level lacks **nuance**. For instance, a particular operator's performance may exhibit better safety margins for the desired specifications while sacrificing on time efficiency. This sort of nuance is often absent in a single score.

In this section, we attempt to formulate such a framework using a combination of ideas from logic and control theory. Our proposed framework first identifies **primitive tasks** that a skillful operator needs to demonstrate. Next, we use signal temporal logic (STL) to specify these tasks in an unambiguous manner [93, 45] and define **skill** as a vector that measures the demonstrated performance of a task. Next, we reflect on the various aspects that characterize a skillful performance.

When considering teleoperation tasks such as piloting a drone, we arrive at several dimensions along which we can evaluate performance:

Robustness: Performance of the required task that is **clearly** correct (compared to **barely** correct) under nominal conditions. This aspect is especially important for safety-critical tasks.

Efficiency: Performance that minimizes time and energy.

Resiliency: Performance under varying environmental conditions for the teleoperated system.

Readiness: Performance under variations in the **human factors**: eg., at different times of day or different levels of comfort.

I provide a more rigorous background in skill assessment in Chapter 2. In this section, we design a framework that seeks to evaluate each of these aspects of skillful performance in the context of teleoperating a drone. We formalize notions of robustness and efficiency while demonstrating how these notions allow us to evaluate human teleoperation of a drone in a simulation environment.

We first describe primitive tasks specified using temporal logic. Next, we will use this in our framework to measure skill levels.

6.1.1 Primitive Tasks and Skills

Our proposed framework for measuring skillfulness starts from a domain-specific knowledge of what tasks are to be performed by the operator. For instance, consider the job of teleoperating a drone to perform an inspection of an oil rig. To perform the overall inspection successfully, the operator must be able to perform numerous atomic, **primitive tasks** such as taking off, maneuvering, hovering and landing. The operator will need to be skilled in performing these primitive tasks in an appropriate manner to complete the overall task at hand. Although the set of all tasks that an operator can be called to perform can be a forbiddingly large, we can enumerate a relatively smaller number of primitives that can be sequenced and combined to form complex tasks.

For drone teleoperation, we define several (but not all possible) primitive tasks and the required drone input controls in Table 6.1. These tasks can be sequenced and combined to form complex tasks; for example, the operator may take off, perform a circular trajectory around a landmark while also keeping the camera trained on it, and land near a charging station.

Table 6.1: Selection of primitive tasks for drone teleoperation

Primitive Task	Controls Required
Angled takeoff	upward throttle + pitch or roll
Angled landing	downward throttle + pitch or roll
Straight angled line	fixed pitch + roll
Curving line	changing pitch + roll
Perspective change	yaw
Hover in place	throttle
Maintain altitude	hover in place + pitch, roll, or yaw

Next, we need to carefully specify each of these primitives formally. Temporal logics were originally proposed for this purpose in the field of computer-aided verification of hardware and software systems [11, 101] and subsequently adopted to robotics as a means for specifying complex robotic tasks [150, 20, 85, 106, 95]. The primitive tasks can be easily specified using a suitable temporal logic. We propose to use metric/STL which include ability to specify real-time constraints as part of the logic [93, 45]. Let $(x(t), y(t), z(t))$ represent the position of a drone with y axis pointing vertically up, and $(v_x(t), v_y(t), v_z(t))$ represent velocities along the respective axes. We will omit other state variables that describe attitude and control inputs $u(t)$ for simplicity.

Hover in place for some time: Let $[y_{\min}, y_{\max}]$ denote the desired range of altitude and $[-\epsilon, \epsilon]$ denote the desired limits on the velocity for some $\epsilon > 0$. Let T be the minimum amount of time we require the UAV to hover in place. In STL, we specify the desired task as follows:

$$\textsc{Eventually}\left(\text{Always}_{[0,T]}\left(y \in [y_{\min}, y_{\max}] \ \wedge \ |v_y| \leq \epsilon\right)\right) .$$

Note that constraints on the attitude can exist but are omitted for simplicity. This formula requires us to find a time window of at least T seconds where the y, v_y are within desired bounds.

Vertical Takeoff: We specify a vertical takeoff from the ground $y = 0$ to some altitude range $H \pm \epsilon$ and a level velocity limit $\delta > 0$ within T seconds. We specify this as:

$$(v_y \geq 0) \ \textsc{until}_{[0,T]} \ (|y - H| \leq \epsilon \ \wedge \ |v_y| \leq \delta) \ .$$

The property specifies that the UAV rise up from the ground until it achieves an altitude in the range $H \pm \epsilon$ with the vertical velocity in the range $\pm\delta$. A vertical landing can be similarly specified.

Temporal Logics provide two advantages: (a) unambiguous specification language that is close to natural language and can be efficiently monitored in real-time [16]; and (b) the ability to measure compliance in terms of distance between the operator's trajectory and a desired specification. The latter property is called robustness and will be explained in the subsequent section.

6.1.2 Measuring Skill

Robustness: We measure robustness of a task performance as a numerical distance between the actual performance and the desired task specification. Consider four trajectories $T_1 - T_4$ in Figure 6.1. The overall task specification is to avoid region R_1 and reach R_2. We note that trajectories T_1 and T_3 both achieve this task. However, T_1 is seen to be more "robust" in achieving the specification than T_3 since it avoids R_1 with a larger margin. At the same time T_2 and T_4 violate the specification. However, a small perturbation of T_4 could have potentially caused it to reach the region R_2 and satisfy the specification. As a result, T_4 is a "less severe" violation than T_2.

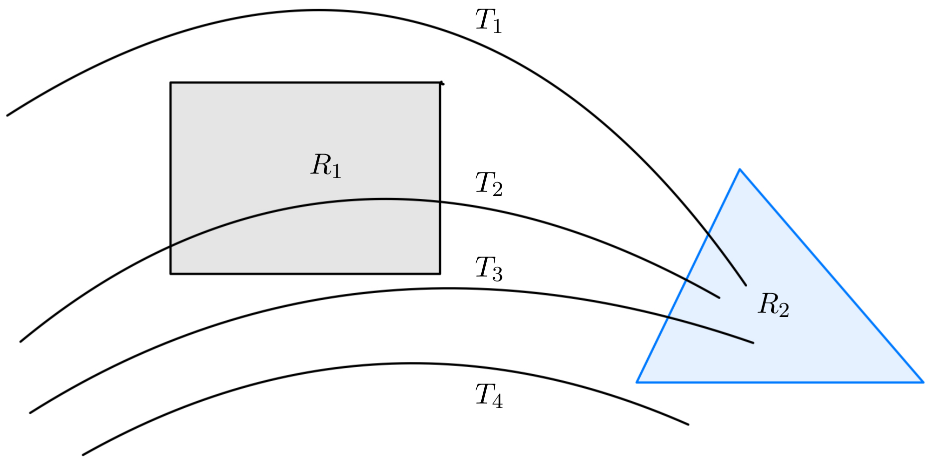


Figure 6.1: Illustration of robustness for various trajectories $T_1 - T_4$ for a drone. The desired task specification is to avoid the shaded region R_1 and reach region R_2.

The task in Figure 6.1 is specified in STL:

$$\underbrace{\textsc{Always}(\neg R_1)}_{\text{avoid } R_1} \wedge \underbrace{\textsc{Eventually}(R_2)}_{\text{Reach } R_2}$$

The previous work of Fainekos et al [53] and Donze et al [46] allow us to systematically compute robustness values with respect to the STL specification. This robustness has a positive value for trajectories that satisfy this property and negative values for violating the trajectory. Robustness measures the diameter of the smallest "tube" around a trajectory such that all trajectories that stay inside this tube have the same outcome (satisfaction or violation) as the original trajectory. Thus, T_1's robustness will yield a large positive value, whereas T_3's robustness will be positive but smaller. Likewise, T_2's robustness will be a negative value with a large magnitude whereas T_4's robustness is also negative but with a small magnitude.

Efficiency: Efficiency is concerned with minimizing use of resources. We can consider resources such as: (a) **Time efficiency:** how much time is taken by the operator to complete the task? (b) **Resource efficiency:** How much energy is expended by the operator? Often, control designers express the resource efficiency as a function over the states of the trajectory and the control action of the human operator. While this is domain specific, it is easy to express and evaluate systematically. (c) **Control Variation**: is the applied control jerky or smooth? This can be measured by computing the total variation distance over the operator inputs. Depending on the domain, there may be many types of efficiency, each contributing to a different skill dimension.

Resilience: Resilience pertains to the correct execution of a task under varying environmental conditions. In a teleoperation setting these apply to the environment surrounding the remote system. Thus, for a teleoperated drone environmental conditions manifest in many ways including wind, sensor malfunctions, and damage to the drone. To measure resilience, we propose repeated execution of a task under unanticipated off-nominal conditions and measuring how robustness of the resulting trajectories vary with changing environmental conditions.

Readiness: Readiness depends on the context surrounding the operator themselves. We posit that skilled operators exhibit readiness against changing contexts that may include their physical comfort, or biological state such as time since last meal or sleepiness. This aspect of skill is the hardest to measure systematically since it is often undesirable to subject humans knowingly to adverse physical conditions.

6.1.3 Evaluation

We conducted an initial evaluation of the proposed skill assessment approach using data collected from a convenience sample of five individuals. Using a drone piloting simulation implemented in Unity (see Figure 6.2) and an Xbox controller, the target task was to take off vertically to reach the floating target, hover within the floating target for five consecutive seconds, and land vertically to reach the landing pad. Note that if the drone strays from the target area during the hover segment, the timer is reset. Each person recorded two attempts of the specified task, yielding 10 different trajectories.

For each trajectory, we evaluated skill along the Robustness and Efficiency dimensions. We discuss possible elicitation of Resiliency and Readiness in the next section.

Robustness: The specifications of takeoff and hover are as specified in Section 6.1.1. The

Table 6.2: Robustness and efficiency measures for each recorded trajectory. The best values in each column are bolded and the worst values are italicized. RO = Robustness, TE = Time Efficiency, and CE = Control Efficiency

		Takeoff			Hover			Land		
Part.	Trial	RO	TE	CE	RO	TE	CE	RO	TE	CE
1	1	-4.99	-5.90	-1.79e-06	0.25	-40.32	**-1.92e-06**	-0.82	-5.38	-5.74e-06
	2	-2.54	-6.11	-1.74e-06	*0.09*	-5.94	-2.91e-06	*-2.19*	*-6.08*	-2.60e-06
2	1	*-7.14*	-6.34	-1.62e-06	0.44	*-44.72*	-4.82e-06	-1.55	-3.53	-6.00e-06
	2	-3.72	-7.60	-2.62e-06	0.12	-15.66	-4.89e-06	-1.59	-4.42	-6.58e-06
3	1	-1.66	*-12.59*	-2.51e-06	0.38	-22.40	-9.72e-06	-1.31	-3.23	-3.03e-06
	2	-7.12	-8.70	**-0.85e-06**	0.25	-7.74	-6.62e-06	**-0.65**	**-2.58**	**-2.07e-06**
4	1	-1.63	-7.95	-1.97e-06	0.71	**-5.22**	-6.84e-06	-1.46	-3.85	*-6.96e-06*
	2	-0.51	-10.33	-1.97e-06	0.93	-7.10	-6.81e-06	-0.13	-4.44	-4.46e-06
5	1	**-0.00**	-9.60	*-5.67e-06*	**1.08**	-6.06	*-11.55e-06*	-1.85	-4.69	-5.65e-06
	2	-4.84	**-5.42**	-2.56e-06	0.95	-13.92	-8.15e-06	-1.32	-2.73	-6.01e-06

specification for landing is similar to takeoff but specifies a negative v_y until the drone reaches a minimum altitude with an appropriately small vertical velocity. We implemented a robustness computation engine as a simplified version of the tool TaLiRo [58]. We calculated separate robustness scores for each of the three task segments.

Efficiency: We measured time efficiency as the time required to complete each segment of the task, where longer times are considered less efficient. For the takeoff and landing segments, the minimum time (and thus most time efficient) to complete the task is essentially 0 seconds. For the hover task, time efficiency is the time required to complete the hover outside of the minimum 5 seconds (if the drone never leaves the target area).

We measured control variation using mean variation distance. For each time t, we measured the distance between the control input vector $(roll, pitch, yaw, throttle)$ and the control input vector at time $t + 1$, scaling by the size of the time step. Smoother control actions yield less change and smaller distances between the control input vectors. To remove redundancy with the time efficiency measure, we computed the mean control variation over the given segment. Efficiency is coded as a negative number, so larger (negative) values represent lower efficiency; this is done to align with intuition that higher numbers are better scores.

6.1.4 Results and Discussion

We present the robustness and efficiency measures for each recorded segment in Table 6.2. One immediate finding is that all robustness estimates for the takeoff and landing segments are negative, which means that none of the recorded trajectories met the desired specifications. In particular, users flew the drone too fast at the end of the takeoff segment, overshooting the desired ending location. For the landing segment, participants often missed the task change and hovered too long before landing using a high speed. All robustness estimates for the hover segments were positive by design, since the simulator would not trigger the next segment until the drone successfully hovered at the target for the desired length of time (5 seconds).

The control efficiency values are very small. This is likely due to the fact that the simulator task only required use of one control input (throttle) and thus the overall control input needed was very small. However, we do see relative differences in control efficiency between the recorded segments, with the hover task showing overall less control efficiency than the takeoff or landing segments.

We arrive at more interesting findings by investigating the relationship between the different skill measures. First, notice in Table 6.2 that Participant 5 in Trial 1 achieved the best robustness and the worst control efficiency in the takeoff and hover segments (see Figure 6.3a). This could be caused by the user “feathering” the controls in order to correct their trajectory. On the other hand, Participant 3 in Trial 2 achieved the best robustness, time efficiency, and control efficiency in the landing segment (see Figure 6.3b). This shows that these dimensions of skill are not necessarily correlated, and may reflect individual user operating styles. Note that we cannot draw any generalized examples here due to our small and non-representative sample of operators; we merely attempt to highlight the nuance afforded by measuring skill along multiple dimensions.

Defining skill along multiple dimensions provides system designers with important decisions as they develop products and interventions. For a specific domain, which aspect of skill is most important? Perhaps the given task is so safety-critical that robustness is the only dimension that matters. Other tasks may depend more on resource efficiency. While we propose several possible dimensions of skill, dimension reduction techniques applied to a larger volume of data may indicate a smaller set of latent dimensions that measure skill.

A limitation of this preliminary analysis is that our data did not allow us to measure the skill dimensions of Resiliency and Readiness. Future work can specifically test these dimensions in a controlled user study by varying conditions in the simulator (e.g., wind, time of day) or the user's conditions (e.g., before/after meals, using distractor tasks). Future work can also extend the proposed framework to describe more complicated tasks that require concurrent primitive skills. Following additional user studies, we can also calculate the distribution of these skill values across a larger population as well as plotting learning curves to see how users improve in various skill dimensions over repeated trials. We plan to also incorporate user confidence ratings of their performance and structured interviews to investigate how users experience skill development. For example, we may see a change from effortful to automatic control such as in [82] or other more qualitative stages of development as users acclimate the control into their own body perception [10, 55].

6.2 Using Large Language Models to Enable Automated Formative Feedback

Automated feedback is a promising approach to scale up training for human-robot interaction (HRI) tasks. By pairing domain knowledge representations with effective assessment, automated feedback systems can identify a learner's current strengths and weaknesses and suggest future actions that will help the learner master the target task. In this section, I discuss how Large Language Models (LLMs) can be used as a tool for providing automated feedback for learning HRI tasks alongside illustrative examples.

Representing knowledge and assessing someone's ability in an HRI task is difficult, due to complex objectives and high variability in human performance. In Section 6.1, we begin to address this question by breaking down HRI tasks into objective primitives that can be combined sequentially and concurrently (e.g., maintain slow speed and reach waypoints). We then show that STL specifications, paired with a robustness metric, are a useful tool for assessing performance along each primitive. These formal methods allow designers to precisely represent ideal trajectories. This formulation admits explainability, as one can identify and elaborate upon specific objectives that learners did not accomplish.

We claim that *LLMs can be paired with formal analysis methods to provide accessible, relevant feedback for HRI tasks.* While logic specifications are useful for defining and assessing a task, these representations are not easily interpreted by non-experts. Luckily, LLMs are adept at generating easy-to-understand text that explains difficult concepts. By integrating task assessment outcomes and other contextual information into a prompt, we can effectively synthesize a useful set of recommendations for the learner to improve their performance (refer to Figure 3.1 as a reminder of the central framework of this dissertation).

I discuss broader approaches to training for psychomotor tasks in Chapter 2. These approaches to feedback fall short in several respects: (1) just-in-time feedback does not promote intentional reflection from the learner on how to improve their performance, (2) feedback is not integrated with an established training curriculum, and (3) feedback does not adapt to a learner's learning trajectory over time.

LLMs are a promising technology poised to tackle these challenges in generating feedback. To address challenge (1), we can develop feedback templates that include elements of effective formative feedback. Formative feedback is an established approach in education that focuses on motivating learners, having them reflect on their performance, and providing a manageable amount of feedback they can use on their next attempt. LLMs can also be integrated with broader training systems that document domain knowledge and skill structures, addressing challenge (2). Finally, LLMs can access historical records of the learner's performance to understand repeated mistakes and opportunities for growth. The greatest strength of LLMs is generating friendly, approachable text, making it ideal for providing feedback that learners perceive positively.

Feedback from LLMs is easy to iterate on and integrate into existing technical workflows. For example, if a task involves multiple robots, we can quickly modify a template prompt for feedback by including a short description of each robot's dynamics or the specific part of the task it is used for. If we need a new feedback format to test a new learning theory, we can swap out the part of the prompt that tells the model how to frame its response. Changing the feedback presentation is also straightforward; the model can act as an intermediate interface between the task and feedback, generating appropriate low-level code to display on a virtual reality headset or to be spoken by a social robot.

The social implications of the proposed approach are also worth considering. As more people are required to interact with robots, training needs to be scalable and personalized to each learner. LLMs can help us reach this goal by making training a friendlier and more appealing process. When paired with commercially available products like virtual reality headsets, training for robotic systems can be accessible to a more diverse group of learners and enable them to train for technical jobs. Additionally, automated feedback generation lessens the burden on human instructors, who will not need to provide as much direct oversight during training.

As with any new technology that is not fully understood, there are many questions to consider before integrating feedback from LLMs into high-stakes or high-impact HRI domains.

The first consideration is anticipating and handling unexpected outputs from the feedback system. It is widely known that LLMs can produce incorrect and harmful responses due to the stochastic nature of the model, biases in the training data, and nuances in system prompts. An automated feedback system should have internal processes to moderate potential harmful outputs, which can be built into the prompt. For example, Tree-of-Thought prompting [69] can be used to emulate experts giving multiple feedback variations and having them reach a consensus based on internal knowledge (recent performance, historical errors, possible emotional states). This approach allows LLMs to recognize and discard inaccurate or poorly phrased feedback. Additional research can collaborate with natural language processing efforts to develop safety alignment when training new models [41].

The second consideration is implementing the feedback system in a robust and sustainable manner. A feedback system requires a thorough yet flexible knowledge representation of the target domain. Using principles from participatory design [163], system creators and domain experts can work together to identify key learning outcomes and assessment criteria. These core learning concepts can then be codified in a formal framework such the Knowledge-Learning-Instruction Framework [87], which associates each practice item with one or more knowledge components. Finally, feedback systems should operationalize theory-driven intervention strategies for providing feedback, such as the Zone of Proximal Development [173] or Deliberate Practice [52]. This would bring more rigor to HRI studies while also contributing to interdisciplinary discourse on training for real-world domains.

The study presented in the remainder of this chapter presents novel work starting to explore how natural language feedback can be used in an adaptive training system (ATS) for a quadrotor landing task.

6.3 Automated Assessment and Feedback Study

6.3.1 Contribution and Research Questions

The contribution of this section is **a flexible and validated framework for automatically assessing performance in multi-objective tasks and generating personalized formative feedback**. We accomplish this by pairing robustness measures of formal task specifications with natural language feedback generated from pedagogically-grounded templates. In this study, we compare groups that received summary metrics of their performance, automatically generated text feedback, and text feedback paired with an annotated figure showing their trajectory. We evaluate the system using the following research questions (RQs).

- *RQ 1.* Do participants perceive the elements of formative feedback differently between score-based, semantic, or multimodal presentations?
- *RQ 2.* What factors predict the perception of formative feedback?
- *RQ 3.* How does automated formative feedback affect participants' learning trajectory?

6.3.2 Quadrotor Landing Task

In this experiment, participants completed a simulated quadrotor landing task. In this task, participants used keyboard inputs to adjust the quadrotor's throttle (vertical force) and attitude (rotation for horizontal force). To achieve a safe landing, the quadrotor must reach the landing pad with a speed less than 15 m/s and a rotation angle within $\pm 5°$. We labeled a landing attempt as unsafe if the drone reached the landing pad but did not satisfy the required speed or angle constraints. All other landing attempts were crashes. We refer to a landing attempt as a *trial*. Each trial was capped at 120 seconds. Figure 6.4 shows the initial configuration of the task participants completed. The initial position of the drone and the landing pad did not change between trials. Yuh et al.'s work provides more details on the dynamics of the quadrotor and design of the simulator [184].

6.3.3 Participants

We recruited participants using the Prolific platform. All participants were United States residents. 177 participants completed the study. Of these participants, 16 restarted the study due to technical issues. To minimize confounding learning effects, we excluded six participants that completed more than five trials before restarting. We excluded another four participants who did not provide a good faith effort in the experiment, as measured by never using the horizontal input controls and crashing the quadrotor on each trial. This resulted in a final dataset of 167 participants.

Participants ranged in age from 18 to 74 years, with a median age of 35 years. Their reported gender identities were 73 Men (44%), 87 Women (52%) and 7 Non-binary individuals (4%). 97% of participants reported no prior experience flying drones or have flown a drone a few times. Participants reported a range of video game experience, with 30 not playing video games (18%), 46 playing monthly (27.5%), 40 playing weekly (24%) and 51 playing daily (30.5%).

6.3.4 Experiment Design and Procedure

We conducted a between-subjects study with three experimental conditions. In the baseline condition, participants received summary statistics such as their landing outcome and an overall score of their performance, replicating prior work [185]. In the second condition, participants received AI-generated text feedback, described in Section 6.3.6. In the third condition, participants received AI-generated text feedback along with an annotated image of their trajectory, which highlighted an area of their trajectory to focus on improving. In our final dataset, 55 participants were in the baseline condition, 56 participants were in the text feedback condition, and 56 participants were in the multimodal feedback condition.

After consenting to participate in the study and reading the instructions for the task, participants completed the quadrotor landing task. After each trial, participants received feedback on their performance depending on their experimental condition. Participants then rated the feedback they received and completed the landing task again. After completing the task 20 times, participants completed a brief demographic questionnaire and rated their overall perception of the feedback they received. On average, participants completed the experiment in 29.12 minutes (SD = 10.24 minutes). They spent an average of 28.12 seconds (SD = 11.08 seconds) reviewing and rating their feedback on each trial.

6.3.5 Automated Assessment

The system automatically assessed landing performance using a previously validated framework [73]. For each component of the task, we defined a specification using STL [44], a formalism for specifying complex temporal tasks. For the quadrotor landing task, the specifications focused on the safety and landing behaviors. The specifications are shown in Table 6.3. Robustness values are a quantitative score that describes how well the trajectory of the quadrotor meets the given specification; large positive values indicate better compliance (e.g., staying far away from the edge of the simulation window) while large negative values indicate stronger violations (e.g., extreme landing angle) [44, 73].

Table 6.3: Overview of specifications for quadrotor landing task with range of possible robustness values for the individual components. Note that the specific values for s_i and l_i depend on the size of the simulation window and the quadrotor.

Description	Specification	Robustness Range
Avoid left edge	$s_1 = x > 0$	[0, 1210]
Avoid right edge	$s_2 = x < 1250$	[0, 1210]
Avoid bottom edge	$s_3 = y > 0$	[0, 575]
Avoid top edge	$s_4 = y < 600$	[0, 575]
Avoid left land edge	$l_1 = x > 650$	[-650, 560]
Avoid right land edge	$l_2 = x < 850$	[-360, 850]
Slow landing speed	$l_3 = \|\|v\|\| < 15$	[-17, 15]
Shallow landing angle	$l_4 = \|\phi\| < 5$	[-24, 5]
Safety component	$S = \wedge_{i=1}^{4} s_i$	
Landing component	$L = \wedge_{i=1}^{4} l_i$	
Complete task in 120s	$S \text{ UNTIL}_{[0,120]} L$	

To keep the feedback manageable, we used a heuristic for selecting the top area of improvement the participant should focus on for the next trial. The safety components were given the highest priority; if the quadrotor crashed into any of the sides of the simulation window (indicated by $s_i = 0$), this was selected as the area of improvement. If the quadrotor landed unsafely, either landing speed or angle was chosen as the area of improvement (l_3 or $l_4 < 0$). For successful landings, the area of improvement was selected as overall efficiency if the trial was longer than a predetermined length or otherwise defaulted to smoothness.

6.3.6 Formative Feedback Design

Participants received formative feedback based on the context generated from the automated assessment in Section 6.3.5 and natural language generated from a prompt incorporating the elements of formative feedback discussed in Section 2.2. The prompt included a description of the target task, the identified area of improvement, the generated image of the trajectory, and an explanation of what each element of the feedback should contain. We used GPT-4V [122] to generate the text feedback.

The visual feedback consisted of an image of the landing trajectory with a superimposed circle to highlight a specific area of improvement along the trajectory. We identified the location of the circle using the area of improvement heuristic described in Section 6.3.5. In the event of a crash, we placed the circle on the location where the quadrotor crashed. If the quadrotor landed unsafely, we placed the circle at the point in the last 50 steps in the trajectory that had the worst robustness for landing speed or landing angle. For a safe landing, we placed the circle at the point in the trajectory with the highest combined control inputs.

Figure 6.5 shows an example of each type of feedback. We generated the full set of text and image feedback regardless of condition so participants waited the same amount of time between trials.

6.3.7 Measures

Subjective Measures. After each trial, participants rated the feedback they received. The purpose of the survey items was to understand how the generated feedback aligned with the desired dimensions of formative feedback described in Section 2.2. Table 6.4 summarizes the survey items participants completed after each trial. After completing the experiment, participants completed an exit survey that recorded their gender identity, age, experience flying drones, and video game experience. Participants also rated how helpful the feedback was overall ("The feedback I received helped me perform better on the task"; 1 = Strongly disagree, 5 = Strongly agree) and provided a text response discussing how the feedback influenced their piloting strategy over time.

Objective Measures. We recorded the trajectory for each trial. The trajectory data included the quadrotor's x and y position and velocity, the quadrotor's rotation, and the participant's control throttle and attitude inputs. For each time step, we also calculated the trajectory's robustness according to the specifications in Section 6.3.5. We recorded both trajectory and robustness data at 50 Hz.

Table 6.4: Summary of the survey items participants completed after receiving feedback for a given trial. Participants rated their feedback using these items for each of the 20 trials in the experiment.

Feedback Dimension	Survey Item	Response Options
Motivation	"The feedback motivated me to do better in future trials."	1 = Strongly disagree, 5 = Strongly agree
Manageable	"How much information did the feedback give?"	1 = Much too little, 5 = Much too much
Actionable	"The feedback suggestions were actionable."	1 = Strongly disagree, 5 = Strongly agree
Timely	"How often was the feedback presented?"	1 = Much too infrequent, 5 = Much too often
Reflection	"The feedback prompted me to reflect on my performance."	1 = Strongly disagree, 5 = Strongly agree

6.3.8 Data Analysis

RQs 1 and 2 ask how participants perceived the feedback they received. The variables of interest were the subjective measures on each feedback dimension shown in Table 6.4 and the overall rating of feedback helpfulness, which yielded ordinal values. We found little discrimination between the extreme values of the Likert scales (Strongly Agree vs. Agree and Strongly Disagree vs. Disagree) so we collapsed these measures to a three-point scale (Disagree, Neutral, Agree) for analysis.

To answer RQ 1, we used the Kruskal-Wallis H-test to test for differences in feedback ratings between groups. As mentioned above, participants rated each dimension of feedback after every trial. To create independent samples, we aggregated survey responses for each participant across trials by calculating the most common response for each item. We found that participant ratings do not change much over time, which suggests that this method of aggregation provides an overall rating of each dimension of feedback.

We used ordered logistic regression models to answer RQ 2. The outcome variables were each participant's overall rating for each feedback dimension and their overall rating of the feedback's helpfulness. The independent variables included participant demographics, total number of safe landings, average trial time, and average time spent reviewing feedback. We also performed a trial-wise analysis of the feedback ratings, using trial time, feedback time, trial number, and landing outcome as predictors. The coefficients of these models (β) represent log-odds; we also report odds-ratios as OR to aid with interpretation.

For RQ3, we considered several metrics of learning trajectory. We first evaluated mastery of the quadrotor landing task by calculating how many participants in each condition achieved at least one safe landing across the 20 trials. We used Fisher's Exact Test to test for differences between feedback conditions. We also considered how much participants improved in the task over time. We measured this by calculating how many more safe landings each participant achieved in the second half of the trials compared to the first half. To compare differences between feedback conditions, we used an independent-samples t-test.

6.3.9 Results

RQ 1 asks if participants in different feedback conditions perceived the dimensions of formative feedback differently. There is a statistically significant difference in ratings along the manageable ($H(2) = 18.0,\ p < 0.001$) and actionable ($H(2) = 18.1,\ p < 0.001$) dimensions. Post-hoc Dunn's test with Bonferroni corrections reveals a significant difference in ratings along the manageable dimension between the baseline and text feedback conditions ($p < 0.001$) and between the baseline and multimodal feedback conditions ($p = 0.005$). There is also a significant difference in ratings along the actionable dimension between the baseline and text feedback conditions ($p < 0.001$) and between the baseline and multimodal feedback conditions ($p < 0.001$). There are no differences in ratings between the non-baseline feedback conditions. There are no significant differences in ratings for the motivation, timely, or reflection dimensions between the feedback groups.

A closer investigation of the distributions of survey responses shows that while participants in all conditions rate the manageability of the feedback as the right amount of information (50-60% of participants in each condition), more participants rate the generated feedback conditions as providing too much information (32-41%). Participants in the baseline condition are more likely to rate the feedback as having not enough information (29%). More participants receiving generated feedback agree that the feedback was actionable (66-70% of participants), while only 35% of participants in the baseline condition agree that the feedback was actionable. Figure 6.6 shows the distributions of ratings for the manageable and actionable feedback dimensions.

Although there are no significant differences between groups, participants as a whole generally find the feedback to be motivational (58-64% agree) and prompting reflection (64-70% agree). Participants also report similar ratings for the timely dimension, with 25-39% reporting that feedback was delivered too often.

There is no significant difference between groups regarding where they found the feedback helpful to improving their performance. The majority of participants in the generated feedback conditions agree that the feedback benefited their performance (61-66%) and 45% of participants in the baseline condition agree that the feedback helped their performance on the task.

RQ 2 asked what factors predict the perception of formative feedback. We first fit an ordered logistic regression model to predict the rating of whether the feedback helped improve performance. While none of the demographic variables are significant predictors, four of the five aggregate measures of formative feedback are significant ($p < 0.05$). Participants with higher 'motivation' ($\beta = +0.97,\ OR = 2.36$) and higher 'reflection' ($\beta = +0.76,\ OR = 2.14$) responses are more likely to rate the feedback as more helpful. Participants with higher 'timely' responses are more likely to rate the feedback as less helpful ($\beta = -1.12,\ OR = 0.33$), since higher 'timely' ratings correspond to the perception that the feedback was given too often. Those with higher 'manageable' ratings rate the feedback as more helpful ($\beta = +1.10,\ OR = 3.01$), which is interesting because higher manageable ratings mean the feedback contained too much information.

There are few variables that predict overall ratings for the elements of formative feedback. Participants with more experience with drones were more likely to rate the manageable dimension as having not enough information ($\beta = -0.80,\ OR = 0.45$). Those who achieved more safe landings are more likely to rate the feedback as occurring too often ($\beta = +0.09,\ OR = 1.10$). Older participants rate the feedback as promoting more reflection ($\beta = +0.06,\ OR = 1.06$). Finally, participants that spent more time completing the experiment rate the feedback as promoting less reflection, although this difference is small ($\beta = -0.001,\ OR = 1.00$).

Table 6.5 summarizes which variables predicted feedback ratings at the trial level. Motivation, reflection, and actionable ratings increase with longer trial times and more time spent reviewing feedback. Participants view feedback more negatively as the experiment progresses, with higher trial numbers corresponding to lower motivation, reflection, and actionable ratings. Participants also rate the feedback as containing too much information and occurring too often when they achieve more successful landings.

Table 6.5: Predictors of trial-wise feedback ratings for each dimension. We report only significant ($p < 0.05$) coefficients as β with corresponding odds-ratios.

Predictor	Motivation (β, OR)	Manageable (β, OR)	Timely (β, OR)	Reflection (β, OR)	Actionable (β, OR)
Trial Time (s)	+0.005, 1.00		-0.01, 0.99	+0.01, 1.01	+0.004, 1.00
Feedback Time (s)	+0.01, 1.01		+0.01, 1.01	+0.01, 1.01	+0.01, 1.01
Trial Number	-0.03, 0.97		+0.02, 1.02	-0.03, 0.97	-0.02, 0.98
Type of Landing		+0.15, 1.17	+0.33, 1.40		

RQ 3 asks how learning trajectories differed between groups. Fisher's Exact Test shows a significant difference in the number of people who failed to achieve a safe landing in any of the trials between the multimodal feedback condition and the baseline condition ($p = 0.04$). Only two participants fail to achieve a safe landing in the multimodal feedback condition (3.6%), compared to eight participants in the baseline condition (14.5%). There are no differences between the multimodal and text feedback conditions or the text and baseline feedback conditions.

All groups improved their performance at the task, as demonstrated by fewer crashes and more safe landings in the second half of the trials (see Table 6.6). Participants in the multimodal feedback condition show a larger increase in safe landings ($M = 2.4$ more safe landings, $SD = 2.3$) compared to the text feedback condition ($M = 1.4$, $SD = 2.7$), $t(110) = 2.2$, $p = 0.03$. There are no significant differences between the baseline condition and the other feedback conditions.

Table 6.6: Average (SD) number of landings for each feedback condition, calculated for the first and second half of the trials and for the whole experiment. Improvement is the average (SD) number more landings in the second half of trials.

	Safe Landings				Safe and Unsafe Landings			
Condition	Trials 1-10	Trials 11-20	Improvement	All Trials	Trials 1-10	Trials 11-20	Improvement	All Trials
Baseline	3.42 (2.85)	5.40 (3.28)	1.98 (2.18)	8.82 (5.75)	7.04 (2.24)	8.53 (1.93)	1.49 (1.92)	15.56 (3.71)
Text	3.05 (2.96)	4.41 (3.07)	1.36 (2.65)	7.46 (5.42)	6.55 (2.61)	8.20 (2.02)	1.64 (2.19)	14.75 (4.17)
Multimodal	2.75 (2.46)	5.12 (2.52)	2.38 (2.33)	7.88 (4.39)	6.93 (2.21)	8.89 (1.27)	1.96 (1.93)	15.82 (3.06)

6.3.10 Main Findings

In this chapter, we developed an end-to-end training system that assesses performance and provides actionable feedback on a quadrotor landing task with no human intervention. The system uses temporal logic task specifications and task demonstrations to assess performance and provide formative feedback to the learner. We found significant differences in how manageable and actionable participants in the different conditions perceived the feedback. Importantly, we found several differences in learning outcomes between conditions. **Participants receiving multimodal feedback were more likely to safely land the quadrotor and showed greater improvement in safe landings in the second half of trials** compared to other feedback conditions.

Overall, participants in all conditions had favorable views of the feedback they received. In particular, participants in the formative feedback conditions mentioned how the feedback impacted their motivation and self-confidence. One participant in the text feedback condition noted, "Overall the encouragement was genuinely nice to receive, and helped to give motivation in completing the task and wanting to do well." Participants in the text and multimodal feedback conditions also reported the feedback felt personalized to their own skills and struggles. Another participant receiving text feedback reported, "The feedback actually felt tailored to me, and not just the same stock answer every time."

Participants in the baseline condition showed surprisingly positive perceptions of their feedback. The written responses indicated that participants were motivated by wanting to figure out how to improve their performance score. One participant noted, "I tried to tell which criteria affected the score more, and how." Although we did not specifically design the baseline condition to be motivational and engaging, this result is in line with work showing that feedback can be intrinsic to the learner [27]. Future works can investigate how to integrate feedback with principles of self-regulated and gamified learning.

However, we found that participants naturally differentiated between performance data and formative feedback. In particular, participants in the baseline condition pushed back against labelling the data summary as feedback. Participants in this condition noted, "The feedback did not help much with strategizing, but it did make me want to get better scores." and "The feedback didn't seem like feedback, because there was no suggestions. The feedback ... was just the numbers that we scored." These findings show that both the content and the delivery of feedback matters. Many of the studies discussed in Chapter 2 implemented feedback similar to our baseline condition in the form of summary metrics. This feedback may be effective by providing learners with more information about their performance, but this depends on the learner to be able to interpret and devise new strategies based on their data. Truly formative feedback should help the learner interpret their data and act on it in future practice.

Participants receiving multimodal feedback were more likely to achieve a safe landing compared to the baseline condition. This could be due to how the multimodal feedback was personalized to address the learner's greatest area of improvement, while the summary statistics in the baseline condition remained the same regardless of performance. For example, one participant in the multimodal feedback condition said, "The feedback helped tremendously by showing the exact location the unnecessary movements where at." Additionally, the format of the formative feedback may have encouraged participants to experiment with new control strategies. Another participant noted, "The feedback helped me feel more confident in the adjustments I was making and to try new approaches."

Finally, participants in the multimodal feedback condition improved more than the text feedback condition by achieving more safe landings in the second half of the experiment compared to the first half. This may be due to additional information provided by the annotated trajectory in the multimodal feedback condition. With the annotated trajectory, participants can pair general ideas presented in the text with a concrete emphasis on a particular area highlighted on the trajectory. In both conditions, participants noted that the feedback did not give specific enough strategies to improve performance. One participant in the text feedback condition noted, "It asked me to consider how changing it "might" be more effective but not exactly how (try using the W key more often to keep the drone up longer, for example)."

These observations highlight an area to improve the feedback prompt template. When designing the feedback, we prompted the model to use actionable suggestions related to the throttle and rotation of the quadrotor. While these terms are specific to quadrotors and other aircraft, they did not tell the learner exactly what to do (e.g. what buttons to push and how) to perform better in this particular simulation environment. This suggests feedback can be actionable on several levels, depending on the complexity of the task one is learning.

6.3.11 Emerging Themes

The results from this study illuminate a need to consider how to adapt feedback beyond the most recent trial. For example, the approach presented here does not consider persistent skill gaps that appear over several trials. One participant in the text feedback condition wrote, "I wish the feedback generated built on the performance in prior trials so the feedback could say *you've improved!* instead of *you need to be better at the same thing... even though you actually did improve compared to the last trial.*"

Additionally, we can use different feedback strategies depending on the overall task performance; high-performing individuals may only need to reinforce their successful control strategies while novices may need more structured and specific feedback presented in this study. Several participants noted frustration when receiving feedback after a successful landing. One participant in the text feedback condition noted, "It is a little discouraging to finally make a successful landing, and then get *a yeah, you did it - but you should focus on doing it better.*"

Future work should also consider how to schedule feedback over time. Several participants reported ignoring the feedback as the trial progressed, especially if they were consistently performing well on the task. A participant receiving multimodal feedback wrote, "After finding the fastest way of landing the drone, I did not follow any more suggestions." Additionally, participants also reported needing time to independently explore the dynamics of the task before receiving performance feedback. A participant in the text feedback condition reported, "It would benefit me to go straight into the next trial so I can continue to make small adjustments back to back... Half of learning is trial and error."

Recent work discusses how prompt-based generative feedback methods are ideal for quickly prototyping and testing feedback templates [79]. Future works can investigate using simple rules to determine what feedback template to generate. How to adjust the timing of formative feedback based on the number of attempts and performance outcomes remains an open question.

As automated training systems continue to develop, it is important to consider their place among other workplace programs. It is likely we will need to balance automated approaches with more traditional one-on-one training [70]; in addition to learning technical skills, training programs will need to consider the social aspects of learning such as developing a community of practice within an organization [97]. As required workplace skills and knowledge continue to develop over time, training systems will need to both provide initial background knowledge and additional support to help workers remain up-to-date [111].

6.3.12 Study Limitations

This work is limited in several ways. First, the quadrotor landing task did not change between trials. This means that when participants found a control strategy that yielded a successful result, they tended to repeat the same strategy. Future works may wish to randomize the starting point of the drone in the simulation to provide more insight about if participants are learning strategies that transfer to other landing scenarios.

The other main limitation of this study is the online nature of the data collection. While the Prolific platform allowed us to quickly recruit a large sample of participants, we were not able to observe nuanced reactions to the feedback they received. Future work can pair crowd-sourced methods with in-person studies to understand how participants choose to integrate feedback into their learning process.

6.4 Conclusions

In this chapter, we motivated and then developed an ATS for a simulated quadrotor landing task. The system first assesses performance based on temporal logic specifications, which require no prior data to learn and can be flexibly adapted to new tasks and situations. Using these assessment results, we automatically generated multimodal feedback adhering to principles of effective formative feedback. While participants in all conditions reported finding the feedback engaging and motivating, they differed in their ratings of how actionable and manageable the feedback was. Since the goal of a training system is to help learners master a new task, we also considered learning differences between conditions. We found that participants receiving multimodal feedback were more likely to achieve a safe landing. They also improved more over the course of the experiment by increasing their safe landings more compared to other feedback conditions. Based on these results, we identified future opportunities to further adapt feedback over time and consider the learner's affective state when delivering feedback. While future work in psychomotor task training will continue to depend on domain-specific methods and knowledge, we encourage researchers to align their methods with established pedagogical theories of learning and feedback.

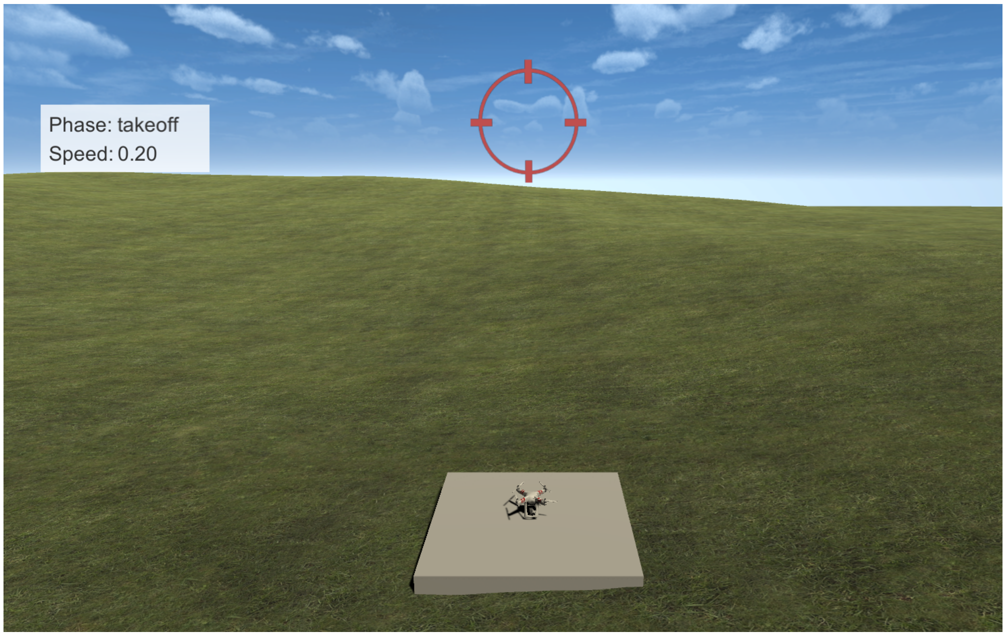
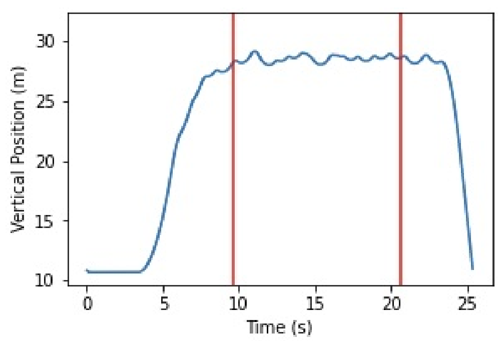
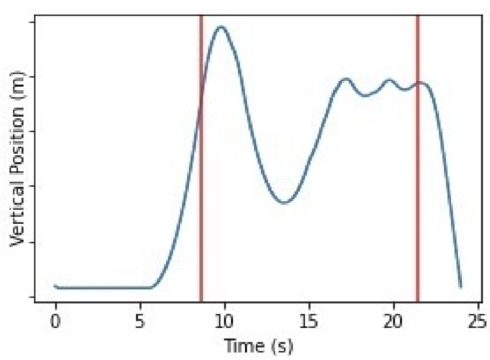


Figure 6.2: Unity-based Drone piloting simulator.



(a) Participant 5, Trial 1



(b) Participant 3, Trial 2

Figure 6.3: Example user trajectories. The vertical position (in meters) is plotted over time (in seconds). The red vertical lines indicate the transition between task segments.

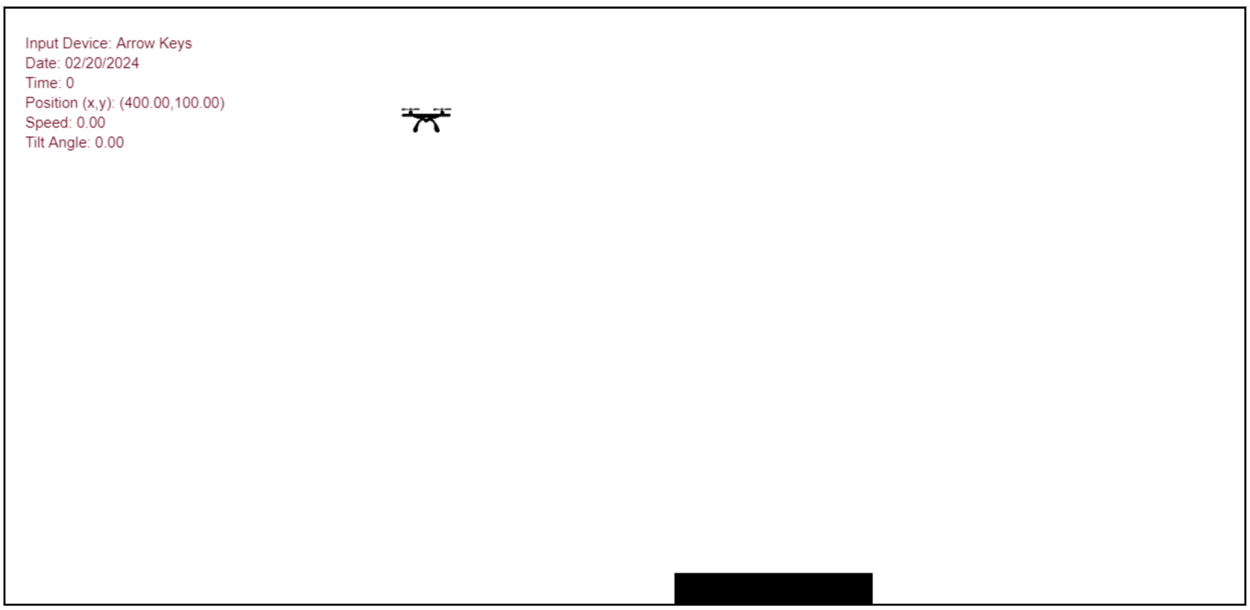


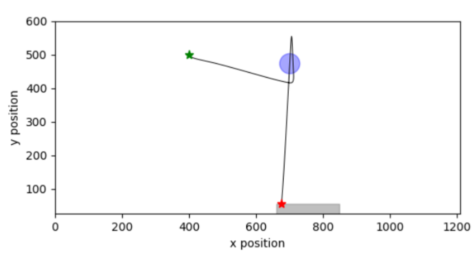
Figure 6.4: Screenshot of the starting configuration of the quadrotor landing task.

Trial No.	Total Time (seconds)	Final Velocity (m/s)	Final Angle (°)	Landing	Score
1	4.22	21.4	9.724	Unsafe	552.6

(a) Baseline feedback condition

Firstly, you've done a good job at keeping the drone within the safe flight area throughout your flight path, which is crucial for avoiding any possible incidents. However, the area you can most improve in is the landing speed. Specifically, your approach speed on landing needs to be more controlled; it's essential to ensure the drone is not moving too quickly as it touches down on the landing pad. To improve on this, focus on reducing thrust a little earlier as you approach the landing area, allowing the drone to decelerate more smoothly. As you reflect on your flight, consider at what point during your descent you could start easing off the thrust to achieve a gentler landing. I am confident that with this adjustment, you'll master the precision required for a smooth and safe touchdown, enhancing your piloting skills significantly. Keep up the determined effort; you're on the right track to completing the target task with excellence.

(b) Text feedback condition

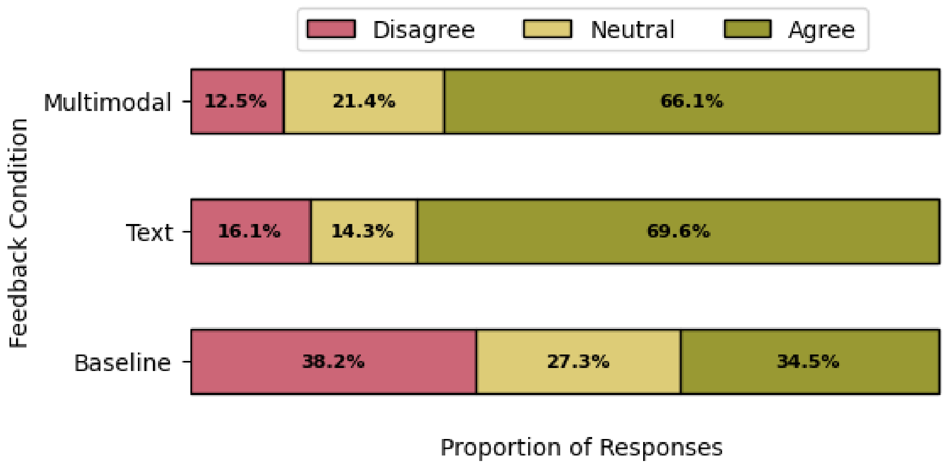


Great job maintaining the drone within the safe flying zone throughout your trajectory; you've shown that you're capable of effectively controlling the drone within the set boundaries. However, there's a significant opportunity for improvement in the way you manage your landing speed. For your next attempt, focus on gradually reducing thrust as you approach the landing pad to ensure the drone descends gently and its speed is well below the target upon touchdown. Reflect on how you can initiate the descent earlier and with less abrupt changes in altitude, which will help you achieve a smoother landing. I'm confident in your ability to adapt and apply these strategies, as the control you've already demonstrated suggests that with a little fine-tuning, you'll be able to nail the perfect landing speed. Keep up the good work and continue to practice these adjustments for an even better performance next time.

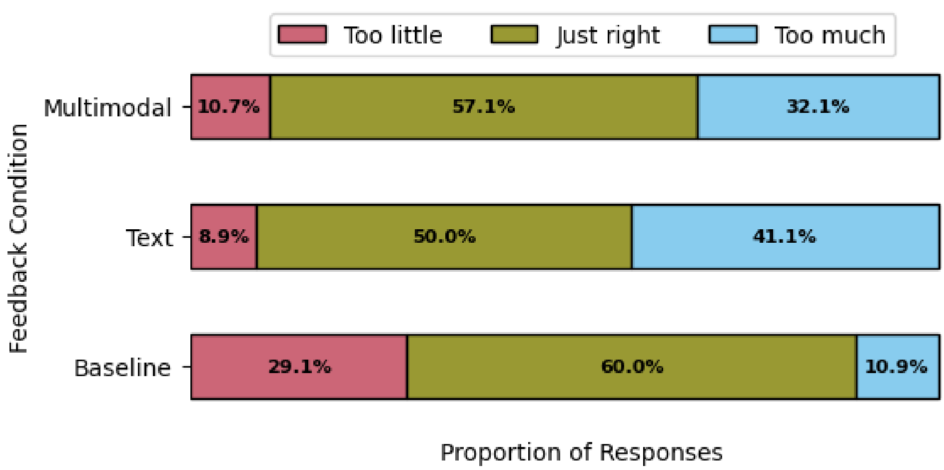
The trajectory above has been highlighted to show which part of your performance you can focus on improving.

(c) Multimodal feedback condition

Figure 6.5: Examples of the three feedback conditions used in the experiment.



(a) Distribution of responses for actionable feedback dimension: "The feedback suggestions were actionable."



(b) Distribution of responses for manageable feedback dimension: "How much information did the feedback give?"

Figure 6.6: Distributions of responses for feedback dimensions that were significantly different between groups. We show the responses collapsed to a three-point likert scale.

Chapter 7

Discussion

7.1 Assessing the Adaptive Training Systems Framework

In Chapter 3 I introduced a framework for developing Adaptive Training Systems (ATS) based on previous work in Intelligent Tutoring Systems (ITS). The framework incorporates four key modules (task attempt, assessment, feedback, and task generation) into an iterative cycle that adapts to the learner's demonstrated skill level. Chapters 5 and 6 demonstrate the utility of symbolic methods for automatic assessment and integrate this assessment with Large Language Model-generated natural language feedback.

The main strengths of the work presented here are that the methods used are explainable and highly modular. The risk models in Chapter 5 provide personalized parameters that describe driver behavior and their relative adherence to the task requirements. For new task requirements or new domains, it is simple to swap out individual risk model parameters without interrupting the rest of the assessment pipeline. Additionally, the robustness metric in Chapter 6 is simple to interpret and new task specifications can readily be incorporated into the assessment module. The assessment and feedback methods used here are largely independent as well; we can generate feedback using information from the risk models or use a different feedback mechanism without impacting the assessment methods. This approach allows system designers to select what methods work best for their specific domain.

The work in this dissertation is limited in several aspects. The first limitation is our evaluation platforms. In both the driving and piloting domains, we assessed our methods in a simulated environment. Simulation provides many benefits, such as controlling external variables and allowing more potential data streams. However, this approach loses some task fidelity and may limit the generalizability of the training to real scenarios. The second limitation is the difficulty of obtaining ground-truth estimates for psychomotor skills. Similar to assessing knowledge in ITS, there are a variety of reasons a learner may perform poorly at a task; they may not understand the requirements of the task, they may not understand the dynamics of the controls, or they do not have the motor control to adequately complete the task. We need to continue developing assessment methods that can distinguish between failure modes in order to provide accurate and effective feedback.

7.2 Revisiting Principles and Best Practices

In addition to those introduced in Section 3.4, I propose two new principles that are meant to improve the scalability and accessibility of ATS. First, ATS should have a *low barrier to entry.* In particular, training systems should consider using affordable off-the-shelf platforms when possible. Technologies such as virtual reality headsets are affordable to many consumers and can provide a realistic training environment when paired with common controllers such as video game joysticks. Using affordable and easily accessible technologies will allow more people to access training resources and can provide career mobility.

Second, ATS should have *development interfaces for non-technical experts.* As ATS become more prevalent and integrated into industrial training pipelines, the role of system designer is likely to fall to task experts without extensive computational knowledge. For example, an expert drone pilot specializing in search and rescue operations will be able to outline key learning objectives and develop training task examples; it is unlikely they will be versed in the convex optimization and logical task specification methods we use in this dissertation. Recent work has begun to explore how to iteratively develop temporal logic task specifications using natural language [57, 34]. Interfaces like these remove unnecessary implementation challenges and empower domain experts to actively engage in the design process.

7.3 Future Work

The studies presented in this dissertation highlight opportunities to improve ATS in the future. Most importantly, ATS need to consider long-term system adaptability. In this work, we do not consider how a driver's risk model changes over time or how drone pilots have different feedback needs based on their experience. For example, in Chapter 6 we found that pilots tended to ignore formative feedback once they achieved a certain level of confidence in the landing task. To address this, the pedagogical and curriculum policies in ATS should include monitoring over several task attempts and even over several training sessions. New methods are needed to computationally identify persistent skill gaps and adjust the training process accordingly.

Future work should also consider a more holistic model of the learner. For instance, the work here does not directly model or consider the affective experience of the learner. While we did design the formative feedback in Chapter 6 to include encouragement and positive support, this fixed feedback template does not directly acknowledge the frustration pilots may feel when they fail the task. Similarly, our driver assessment approach in Chapter 5 does not consider how changes in environmental conditions (e.g., time of day, room temperature) may impact driving preferences and behaviors. These factors will also change over time, underscoring the need for flexible and adaptive long-term modeling approaches.

Finally, work in adaptive task generation is a largely unexplored topic in ATS. For ITS, some work has been done to procedurally generate mathematics problems and their corresponding solutions [62]. For psychomotor tasks, we can take inspiration from procedural content generation in video games; work in this area has focused on generating new game levels that are guaranteed to be playable and adapt to a player's skill level [145]. We can pair these methods with the domain model of the training task to create training tasks that target a specific skill [8, 162].

Chapter 8

Conclusion

The rise of advanced machine learning and artificial intelligence methods presents an exciting opportunity to personalize classroom learning and workforce training. The work presented in this dissertation was motivated by the need for timely formative feedback that adapts to the learner's immediate actions. Specifically, this dissertation addresses this gap by pairing automated assessment from signal temporal logic (STL) task specifications with natural language feedback from a generative language model.

Using a driving domain and a drone teleoperation domain, we showed that STL specifications provide a flexible, generalizable, and interpretable framework for defining and assessing human-robot interaction (HRI) tasks. Compared to previous methods, the approach presented in this dissertation does not require extensive data to train and can easily be adapted to new domains and tasks. By pairing this assessment with a generative language model, we were able to personalize feedback for learners using principles of effective formative feedback. I discuss findings from the individual studies below in more detail.

In this Chapter 3, I proposed a framework for developing Adaptive Training Systems (ATS) to support HRI tasks. This framework builds on foundational work in Intelligent Tutoring Systems (ITS) and acknowledges the training challenges unique to HRI contexts. Namely, the automated modules for assessment, feedback, and task generation need to be flexible enough to transfer to new tasks with minimal training data and should embrace the nuance in performance that comes with evaluating psychomotor tasks. Along with the framework, I discuss best practices and guidelines for designing ATS.

I then demonstrated how these principles can be applied to two different domains targeting psychomotor skills. In Chapter 5, we focused on the assessment module of the training framework. We defined multiple competing objectives in a driving simulator task using risk field models. The risk field model learned individual parameters for how well each driver adhered to the task objectives. We were then able to reliably predict the driver's future driving trajectory. Additionally, we could estimate the driver's situation awareness by predicting whether they were aware of an upcoming obstacle.

The work in this chapter provides a first glimpse of the utility for using STL as an assessment tool for HRI tasks. Using task specifications, we can monitor learner performance either over a subset of specific task components or the task as a whole. Additionally, the risk framework presented in this chapter provides a personalized, generative model of learner behavior that can be integrated into future feedback and intervention systems.

In Chapter 6, we developed automated assessment and feedback modules in a quadrotor drone piloting domain. We again defined task objectives using STL specifications, which are easy to define and simple to programmatically evaluate. Using these specifications, we calculated how robustly the drone pilot adhered to the task objectives. The feedback module selected the skill area the pilot needed the most improvement in and generated text and image feedback using principles of formative feedback. We found the drone pilots positively perceived the generated feedback and their performance improved more than pilots receiving other forms of feedback.

The main contribution of this chapter is an end-to-end training system that provides personalized formative feedback on a drone landing task using automated assessment from STL task specifications. This framework enables us to systematically test different feedback mechanisms to understand their efficacy in training. For example, the feedback presented in this chapter used formative feedback, which emphasizes motivation; however, future work can compare how well this feedback compares to other types of scaffolding or different social presentations.

The work in these two domains shows how portions of the framework can work together to accurately model the learner and provide useful performance feedback. The methods used in these two domains emphasize the design principles I introduced in Chapter 3. For example, the feedback presented to pilots was based on pedagogical theory of formative feedback. In addition, the assessment and feedback methods presented here are highly modular; it would be a very simple task to swap out feedback templates or introduce a new assessment metric to replace the robustness values. Future work can continue to refine these principles and introduce alternate methods as new computational approaches emerge. By adopting a systematic approach to design, we have the chance to revolutionize how we train the future workforce and confidently adapt to technologies that do not yet exist.

Bibliography

[1] Agrawal, S. K., Chen, X., Ragonesi, C., and Galloway, J. C. Training toddlers seated on mobile robots to steer using force-feedback joystick. IEEE Transactions on Haptics 5, 4 (2012), 376–383.

[2] Agudo-Peregrina, A. F., Iglesias-Pradas, S., Conde-González, M. A., and Hernández-García, A. Can we predict success from log data in VLEs? Classification of interactions for learning analytics and their relation with performance in VLE-supported F2F and online learning. Computers in Human Behavior 31 (Feb. 2014), 542–550.

[3] Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., and Qin, Y. An Integrated Theory of the Mind. Psychological Review 111, 4 (2004), 1036–1060.

[4] Andrew, M., Marler, T., Lastunen, J., Acheson-Field, H., and Popper, S. An Analysis of Education and Training Programs in Advanced Manufacturing Using Robotics. RAND Corporation, Pittsburgh, Pennsylvania, 2020.

[5] Ashenafi, M. M., Riccardi, G., and Ronchetti, M. Predicting students' final exam scores from their course activities. In 2015 IEEE Frontiers in Education Conference (FIE) (Oct. 2015), IEEE, pp. 1–9.

[6] Aslan, S., Alyuz, N., Tanriover, C., Mete, S. E., Okur, E., D'Mello, S. K., and Esme, A. A. Investigating the Impact of a Real-time, Multimodal Student Engagement Analytics Technology in Authentic Classrooms. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19 (New York, New York, USA, 2019), ACM Press, pp. 1–12.

[7] Ausubel, D. P., and Youssef, M. The Effect of Spaced Repetition on Meaningful Retention. The Journal of General Psychology 73, 1 (Oct. 1965), 147–150.

[8] Azad, S., Saldanha, C., Gan, C.-H., and Riedl, M. Mixed Reality Meets Procedural Content Generation in Video Games. Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment 12, 2 (2016), 22–26.

[9] Azevedo, R., Johnson, A., Chauncey, A., and Burkett, C. Self-regulated Learning with MetaTutor: Advancing the Science of Learning with MetaCognitive Tools BT - New Science of Learning: Cognition, Computers and Collaboration in Education. In New Science of Learning: Cognition, Computers and Collaboration in Education, M. S. Khine and I. M. Saleh, Eds. Springer New York, New York, NY, 2010, pp. 225–247.

[10] Baber, C., Cengiz, T., and Parekh, M. Tool use as distributed cognition: how tools help, hinder and define manual skill. Frontiers in Psychology 5 (2014).

[11] Baier, C., and Katoen, J.-P. Principles of Model Checking. MIT Press, 2008.

[12] Baker, C. L., Saxe, R., and Tenenbaum, J. B. Action understanding as inverse planning. Cognition 113 (2009), 329–349.

[13] Baker, R. S., and Ocumpaugh, J. Interaction-Based Affect Detection in Educational Software. In The Oxford Handbook of Affective Computing, R. A. Calvo, S. K. D'Mello, J. Gratch, and A. Kappas, Eds. Oxford University Press, Jan. 2015.

[14] Baker, R. S. J., Corbett, A. T., and Aleven, V. More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. In Intelligent Tutoring Systems (Berlin, Heidelberg, 2008), B. P. Woolf, E. Aïmeur, R. Nkambou, and S. Lajoie, Eds., Springer Berlin Heidelberg, pp. 406–415.

[15] Banihashem, S. K., Noroozi, O., Van Ginkel, S., Macfadyen, L. P., and Biemans, H. J. A systematic review of the role of learning analytics in enhancing feedback practices in higher education. Educational Research Review 37 (Nov. 2022), 100489.

[16] Bartocci, E., Deshmukh, J., Donzé, A., Fainekos, G., Maler, O., Ničković, D., and Sankaranarayanan, S. Specification-Based Monitoring of Cyber-Physical Systems: A Survey on Theory, Tools and Applications. In Lectures on Runtime Verification: Introductory and Advanced Topics. Springer, Feb. 2018, pp. 135–175.

[17] Basu, C., Yang, Q., Hungerman, D., Singhal, M., and Dragan, A. D. Do You Want Your Autonomous Car To Drive Like You? In Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction (New York, NY, USA, 3 2017), ACM, pp. 417–425.

[18] Beheshti, B., Desmarais, M. C., and Naceur, R. Methods to find the number of latent skills. In Proceedings of the 5th International Conference on Educational Data Mining (2012).

[19] Bell, D. A. Employment skills for the robot age. Robotica 3, 2 (Apr. 1985), 93–95.

[20] Bhatia, A., Kavraki, L. E., and Vardi, M. Y. Sampling-based motion planning with temporal goals. In 2010 IEEE International Conference on Robotics and Automation. IEEE, May 2010, pp. 2689–2696.

[21] Blanchard, N., Donnelly, P. J., Olney, A. M., Samei, B., Ward, B., Sun, X., Kelly, S., Nystrand, M., and D'Mello, S. K. Automatic Detection of Teacher Questions from Audio in Live Classrooms. In Proceedings of the 9th International Conference on Educational Data Mining (EDM 2016) (2016), International Educational Data Mining Society.

[22] Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy,

J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P. W., Krass, M., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X. L., Li, X., Ma, T., Malik, A., Manning, C. D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J. C., Nilforoshan, H., Nyarko, J., Ogut, G., Orr, L., Papadimitriou, I., Park, J. S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y., Ruiz, C., Ryan, J., Ré, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K., Tamkin, A., Taori, R., Thomas, A. W., Tramèr, F., Wang, R. E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S. M., Yasunaga, M., You, J., Zaharia, M., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K., and Liang, P. On the opportunities and risks of foundation models, 2022.

[23] Boudette, N. E., and Chokshi, N. U.s. will investigate tesla's autopilot system over crashes with emergency vehicles. The New York Times (August 2021).

[24] Boyd, S., and Vandenberghe, S. Convex Optimization. Cambridge University Press, 2004.

[25] Brinton, C. G., and Chiang, M. MOOC performance prediction via clickstream data and social learning networks. Proceedings - IEEE INFOCOM 26 (2015), 2299–2307.

[26] Buche, C., Querrec, R., De Loor, P., and Chevaillier, P. MASCARET: pedagogical multi-agents systems for virtual environment for training. In Proceedings. 2003 International Conference on Cyberworlds (Singapore, 2003), IEEE Comput. Soc, pp. 423–430.

[27] Butler, D. L., and Winne, P. H. Feedback and self-regulated learning: A theoretical synthesis. Review of Educational Research 65, 3 (1995), 245–281.

[28] Chaturvedi, R., and Ezeife, C. I. Predicting Student Performance in an ITS Using Task-Driven Features. In 2017 IEEE International Conference on Computer and Information Technology (CIT) (Aug. 2017), IEEE, pp. 168–175.

[29] Chi, M. T. H., and Wylie, R. The ICAP Framework: Linking Cognitive Engagement to Active Learning Outcomes. Educational Psychologist 49, 4 (Oct. 2014), 219–243.

[30] Church, K. W., and Hanks, P. Word Association Norms, Mutual Information, and Lexicography. Computational Linguistics 16, 1 (1990), 76–83.

[31] Cook, C., Olney, A. M., Kelly, S., and D'Mello, S. K. An Open Vocabulary Approach for Estimating Teacher Use of Authentic Questions in Classroom Discourse. In Proceedings of the 11th International Conference on Educational Data Mining (2018).

[32] Corbett, A. T., and Anderson, J. R. Knowledge-tracing: Modeling the acquisition of procedural knowledge. User Modeling and User Adapted Interaction 4 (1995), 253–278.

[33] Corbett, A. T., Koedinger, K. R., and Anderson, J. R. Chapter 37 - Intelligent Tutoring Systems. In Handbook of Human-Computer Interaction (Second Edition), M. G.

Helander, T. K. Landauer, and P. V. Prabhu, Eds. North-Holland, Amsterdam, Jan. 1997, pp. 849–874.

[34] Cosler, M., Hahn, C., Mendoza, D., Schmitt, F., and Trippel, C. nl2spec: Interactively translating unstructured natural language to temporal logics with large language models. In Computer Aided Verification (Cham, 2023), C. Enea and A. Lal, Eds., Springer Nature Switzerland, pp. 383–396.

[35] Dalipi, F., Imran, A. S., and Kastrati, Z. Mooc dropout prediction using machine learning techniques: Review and research challenges. In 2018 IEEE Global Engineering Education Conference (EDUCON) (2018), pp. 1007–1014.

[36] Davaris, M., Wijewickrema, S., Zhou, Y., Piromchai, P., Bailey, J., Kennedy, G., and O'Leary, S. The Importance of Automated Real-Time Performance Feedback in Virtual Reality Temporal Bone Surgery Training. In Proceedings of the 2019 Artificial Intelligence in Education Conference (Cham, 2019), S. Isotani, E. Millán, A. Ogan, P. Hastings, B. McLaren, and R. Luckin, Eds., Springer International Publishing, pp. 96–109.

[37] Di Mitri, D., Schneider, J., and Drachsler, H. Keep Me in the Loop: Real-Time Feedback with Multimodal Data. Int J Artif Intell Educ 32, 4 (Dec. 2022), 1093–1118.

[38] Diamond, R. Clarifying instructional goals and objectives. Designing and assessing courses and curricula: A practical guide (1998).

[39] D'Mello, S. K., Blanchard, N., Baker, R. S., Ocumpaugh, J., and Brawner, K. I Feel Your Pain: A Selective Review of Affect-Sensitive Instructional Strategies. In Design Recommendations for Intelligent Tutoring Systems -Volume 2 Instructional Management, R. A. Sottilare, A. C. Graesser, X. Hu, and B. S. Goldberg, Eds. U.S. Army Research Laboratory, 2014, pp. 35–48.

[40] Doan, T.-N., and Sahebi, S. Rank-Based Tensor Factorization for Student Performance Prediction. In The 12th International Conference on Educational Data Mining (2019), C. F. Lynch, A. Merceron, M. Desmarais, and R. Nkambou, Eds., International Educational Data Mining Society, pp. 288–293.

[41] Dong, Z., Zhou, Z., Yang, C., Shao, J., and Qiao, Y. Attacks, defenses and evaluations for llm conversation safety: A survey, 2024.

[42] Donnelly, P. J., Blanchard, N., Samei, B., Olney, A. M., Sun, X., Kelly, S., Nystrand, M., and D'Mello, S. K. Automatic Teacher Modeling from Live Classroom Audio. In Proceedings of the 24th Conference on User Modeling, Adaptation, and Personalization (UMAP 2016) (2016), ACM Press.

[43] Donnelly, P. J., Kelly, S., Blanchard, N., Nystrand, M., Olney, A. M., and D'Mello, S. K. Words matter: Automatic detection of teacher questions in live classroom discourse using linguistics, acoustics, and context. ACM International Conference Proceeding Series (2017), 218–227.

[44] Donzé, A., and Maler, O. Robust satisfaction of temporal logic over real-valued signals. In FORMATS, vol. 6246 of Lecture Notes in Computer Science. Springer, 2010, pp. 92–106.

[45] Donzé, A. On Signal Temporal Logic. In Runtime Verification. Springer, 2013, pp. 382–383.

[46] Donzé, A., and Maler, O. Robust Satisfaction of Temporal Logic over Real-Valued Signals. In Formal Modeling and Analysis of Timed Systems. Springer, Berlin, Germany, 2010, pp. 92–106.

[47] Echeverria, J., and C. Santos, O. Kumitron: Artificial intelligence system to monitor karate fights that synchronize aerial images with physiological and inertial signals. In Companion Proceedings of the 26th International Conference on Intelligent User Interfaces (New York, NY, USA, 2021), IUI '21 Companion, Association for Computing Machinery, p. 37–39.

[48] Echeverria, J., and Santos, O. C. Toward modeling psychomotor performance in karate combats using computer vision pose estimation. Sensors 21, 24 (2021).

[49] Elbadrawy, A., Polyzou, A., Ren, Z., Sweeney, M., Karypis, G., and Rangwala, H. Predicting Student Performance Using Personalized Analytics. Computer 49, 4 (2016), 61–69.

[50] Emerson, A., Smith, A., Smith, C., Rodríguez, F. J., Min, W., Wiebe, E., Mott, B., Boyer, K. E., and Lester, J. Predicting Early and Often: Predictive Student Modeling for Block-Based Programming Environments. In The 12th International Conference on Educational Data Mining (2019), C. F. Lynch, A. Merceron, M. Desmarais, and R. Nkambou, Eds., International Educational Data Mining Society, pp. 39–48.

[51] Endsley, M. R. Toward a theory of situation awareness in dynamic systems. Human Factors 37, 1 (1995), 32–64.

[52] Ericsson, K. A., Krampe, R. T., and Tesch-Römer, C. The Role of Deliberate Practice in the Acquisition of Expert Performance. Psychological Review 100, 3 (1993), 363–406.

[53] Fainekos, G. E., and Pappas, G. J. Robustness of temporal logic specifications for continuous-time signals. Theoret. Comput. Sci. 410, 42 (Sept. 2009), 4262–4291.

[54] Filev, D., Lu, J., Prakah-Asante, K., and Tseng, F. Real-time driving behavior identification based on driver-in-the-loop vehicle dynamics and control. In IEEE International Conference on Systems, Man and Cybernetics (2009), IEEE, pp. 2020–2025.

[55] Fischer, K., and Zheng, Y. Conceptions of Development. Psychology Press, 2002, ch. The Development of Dynamic Skill Theory.

[56] Ford, M., Baer, C. T., Xu, D., Yapanel, U., and Gray, S. The LENA Language Environment Analysis System. Tech. rep., LENA Foundation, Boulder, CO, 2008.

[57] Gavran, I., Darulova, E., and Majumdar, R. Interactive synthesis of temporal specifications from examples and natural language. Proc. ACM Program. Lang. 4, OOPSLA (November 2020).

[58] Georgios Fainekos et al. TALIRO-TOOLS, November 2022.

[59] Gibson, J. J., and Crooks, L. E. A theoretical field-analysis of automobile-driving. The American Journal of Psychology 51, 3 (1938), 453–471.

[60] Golestan, K., Soua, R., Karray, F., and Kamel, M. S. Situation awareness within the context of connected cars: A comprehensive review and recent trends. Information Fusion 29 (2016), 68–83.

[61] Gregory, R. A., and Ward, T. Work Force Characteristics in a Robot Driven Construction Industry. In Proceedings of the 17th IAARC/CIB/IEEE/IFAC/IFR International Symposium on Automation and Robotics in Construction (Taipei, Taiwan, Sept. 2000), International Association for Automation and Robotics in Construction (IAARC), pp. 1–5.

[62] Gulwani, S. Example-based learning in computer-aided STEM education. Communications of the ACM 57, 8 (Aug. 2014), 70–80.

[63] Hartley, J. R., and Sleeman, D. H. Towards more intelligent teaching systems. International Journal of Man-Machine Studies 5, 2 (1973), 215–236.

[64] Haslgrübler, M., Gollan, B., Thomay, C., Ferscha, A., and Heftberger, J. Towards skill recognition using eye-hand coordination in industrial production. In Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments (New York, NY, USA, 2019), PETRA '19, Association for Computing Machinery, p. 11–20.

[65] Hatziapostolou, T., and Paraskakis, I. Enhancing the Impact of Formative Feedback on Student Learning Through an Online Feedback System. Electronic Journal of e-Learning 8, 2 (2010), 111 – 122.

[66] Henderson, M., Phillips, M., Ryan, T., Boud, D., Dawson, P., Molloy, E., and Mahoney, P. Conditions that enable effective feedback. Higher Education Research & Development 38, 7 (Nov. 2019), 1401–1416.

[67] Hofbauer, M., Kuhn, C. B., Püttner, L., Petrovic, G., and Steinbach, E. Measuring driver situation awareness using region-of-interest prediction and eye tracking. In 2020 IEEE International Symposium on Multimedia (ISM) (2020), pp. 91–95.

[68] Holstein, K., McLaren, B. M., and Aleven, V. Student learning benefits of a mixed-reality teacher awareness tool in AI-enhanced classrooms. In Artificial Intelligence in Education (2018), pp. 154–168.

[69] Hulbert, D. Using tree-of-thought prompting to boost chatgpt's reasoning. `https://github.com/dave1010/tree-of-thought-prompting`, May 2023.

[70] Hutson, J., and Ceballos, J. Rethinking Education in the Age of AI: The Importance of Developing Durable Skills in the Industry 4.0. Journal of Information Economics 1, 2 (July 2023), 26–35.

[71] Hutt, S., Grafsgaard, J. F., and D'Mello, S. K. Time to Scale : Generalizable Affect Detection for Tens of Thousands of Students across an Entire School year. In 2019 CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019) (2019), ACM Press.

[72] Jensen, E., Dale, M., Donnelly, P. J., Stone, C., Kelly, S., Godley, A., and D'Mello, S. K. Toward Automated Feedback on Teacher Discourse to Enhance Teacher

Learning. In 2020 CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2020) (2020), ACM Press.

[73] Jensen, E., Hayes, B., and Sankaranarayanan, S. More Than a Number: A Multi-dimensional Framework For Automatically Assessing Human Teleoperation Skill. In Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction (Stockholm Sweden, Mar. 2023), ACM, pp. 653–657.

[74] Jensen, E., Hunkins, N. C., D'Mello, S. K., Hutt, S., and Huggins-Manley, A. C. What You Do Predicts How You Do: Prospectively Modeling Student Quiz Performance Using Activity Features in an Online Learning Environment. In LAK21: 11th International Learning Analytics and Knowledge Conference (LAK21) (2021), ACM.

[75] Jensen, E., Hutt, S., and D'Mello, S. K. Generalizability of Sensor-Free Affect Detection Models in a Longitudinal Dataset of Tens of Thousands of Students. In The 12th International Conference on Educational Data Mining (2019), M. Desmarais, C. F. Lynch, A. Merceron, and R. Nkambou, Eds., pp. 324–329.

[76] Jensen, E., Luster, M., Pitts, B., and Sankaranarayanan, S. Using Artificial Potential Fields to Model Driver Situational Awareness. IFAC-PapersOnLine 55, 41 (2022), 148–153.

[77] Jensen, E., Luster, M., Yoon, H., Pitts, B., and Sankaranarayanan, S. Mathematical Models of Human Drivers Using Artificial Risk Fields. In Intelligent Transportation Systems Conference (October 2022).

[78] Jensen, E., Pugh, S. L., and D'Mello, S. K. A Deep Transfer Learning Approach to Modeling Teacher Discourse in the Classroom. In LAK21: 11th International Learning Analytics and Knowledge Conference (LAK21) (2021), ACM.

[79] Jensen, E., Sankaranarayanan, S., and Hayes, B. Large language models enable automated formative feedback in human-robot interaction tasks. In Human-Large Language Model Interaction workshop at the 2024 ACM/IEEE International Conference on Human-Robot Interaction (2024).

[80] Johannsmeier, L., Gerchow, M., and Haddadin, S. A framework for robot manipulation: Skill formalism, meta learning and adaptive control. In 2019 International Conference on Robotics and Automation (ICRA) (2019), pp. 5844–5850.

[81] Kadar, E. E., and Shaw, R. E. Toward an ecological field theory of perceptual control of locomotion. Ecological psychology 12, 2 (2000), 141–180.

[82] Kahneman, D. Thinking, fast and slow. Farrar, Staus and Giroux, New York, NY, USA, 2011.

[83] Kelly, S., Olney, A. M., Donnelly, P. J., Nystrand, M., and D'Mello, S. K. Automatically Measuring Question Authenticity in Real-World Classrooms. Educational Researcher 47, 7 (2018), 451–464.

[84] Kim, B., Kang, C. M., Kim, J., Lee, S. H., Chung, C. C., and Choi, J. W. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network.

In 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC) (2017), pp. 399–404.

[85] Kim, J., Muise, C., Shah, A., Agarwal, S., and Shah, J. Bayesian inference of linear temporal logic specifications for contrastive explanations. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19 (7 2019), International Joint Conferences on Artificial Intelligence Organization, pp. 5591–5598.

[86] Ko, S., Kutchek, K., Zhang, Y., and Jeon, M. Effects of Non-Speech Auditory Cues on Control Transition Behaviors in Semi-Automated Vehicles: Empirical Study, Modeling, and Validation Effects of Non-Speech Auditory Cues on Control Transition Behaviors in. International Journal of Human–Computer Interaction (2021), 1–16.

[87] Koedinger, K. R., Corbett, A. T., and Perfetti, C. The Knowledge-Learning-Instruction Framework: Bridging the Science-Practice Chasm to Enhance Robust Student Learning. Cognitive Science 36, 5 (July 2012), 757–798.

[88] Koedinger, K. R., Kim, J., Jia, J. Z., McLaughlin, E. A., and Bier, N. L. Learning is not a spectator sport: Doing is better than watching for learning from a mooc. In Proceedings of the Second (2015) ACM Conference on Learning @ Scale (New York, NY, USA, 2015), L@S '15, Association for Computing Machinery, p. 111–120.

[89] Koedinger, K. R., McLaughlin, E. A., Kim, J., Jia, J. Z., and Bier, N. L. Learning is not a spectator sport: Doing is better than watching for learning from a MOOC. L@S 2015 - 2nd ACM Conference on Learning at Scale (2015), 111–120.

[90] Kolekar, S., de Winter, J., and Abbink, D. Human-like driving behaviour emerges from a risk-based driver model. Nature Communications 11, 1 (2020).

[91] Kolekar, S., Winter, J. D., and Abbink, D. Which parts of the road guide obstacle avoidance? Quantifying the driver's risk field. Applied Ergonomics 89, July (2020).

[92] Korhonen, T., Lindqvist, T., Laine, J., and Hakkarainen, K. Training Hard Skills in Virtual Reality: Developing a Theoretical Framework for AI-Based Immersive Learning. In AI in Learning: Designing the Future, H. Niemi, R. D. Pea, and Y. Lu, Eds. Springer International Publishing, Cham, 2023, pp. 195–213.

[93] Koymans, R. Specifying real-time properties with metric temporal logic. Real-Time Syst. 2, 4 (Nov. 1990), 255–299.

[94] Krems, J. F., and Baumann, M. R. K. Driving and situation awareness: A cognitive model of memory-update processes. In Human Centered Design (Berlin, Heidelberg, 2009), M. Kurosu, Ed., Springer Berlin Heidelberg, pp. 986–994.

[95] Kress-Gazit, H., Fainekos, G. E., and Pappas, G. J. Temporal-Logic-Based Reactive Mission and Motion Planning. IEEE Trans. Rob. 25, 6 (Sept. 2009), 1370–1381.

[96] Laine, J., Lindqvist, T., Korhonen, T., and Hakkarainen, K. Systematic Review of Intelligent Tutoring Systems for Hard Skills Training in Virtual Reality Environments. International Journal of Technology in Education and Science 6, 2 (May 2022), 178–203.

[97] Leon, R. D. Employees' reskilling and upskilling for industry 5.0: Selecting the best professional development programmes. Technology in Society 75 (Nov. 2023), 102393.

[98] Lesgold, A. Toward a Theory of Curriculum for Use in Designing Intelligent Instructional Systems. Springer US, New York, NY, 1988, pp. 114–137.

[99] Li, L. Reskilling and Upskilling the Future-ready Workforce for Industry 4.0 and Beyond. Inf Syst Front 2022 (July 2022).

[100] Luengo, V., and Mufti-Alchawafa, D. Target the controls during the problem solving activity, a process to produce adapted epistemic feedbacks in ill- defined domains. In Formative Feedback in Interactive Learning Environments (FFILE) Conference (Memphis, USA, 2013), Springer, p. 8.

[101] Manna, Z., and Pnueli, A. The Temporal Logic of Reactive and Concurrent Systems. Springer, New York, NY, USA, 1992.

[102] Mao, Y., Lin, C., and Chi, M. Deep Learning vs. Bayesian Knowledge Tracing: Student Models for Interventions. Journal of Educational Data Mining 10, 2 (2018), 28–54.

[103] Martinez, C., Serra, R., Sundaramoorthy, P., Booij, T., Vertegaal, C., Bounik, Z., Van Hastenberg, K., and Bentum, M. Content-Focused Formative Feedback Combining Achievement, Qualitative and Learning Analytics Data. Education Sciences 13, 10 (Oct. 2023), 1014.

[104] Mat Sanusi, K. A., Mitri, D. D., Limbu, B., and Klemke, R. Table tennis tutor: Forehand strokes classification based on multimodal data and neural networks. Sensors 21, 9 (2021).

[105] Meier, Y., Xu, J., Atan, O., and van der Schaar, M. Predicting Grades. IEEE Transactions on Signal Processing 64, 4 (Feb. 2016), 959–972.

[106] Menghi, C., Tsigkanos, C., Berger, T., Pelliccione, P., and Ghezzi, C. Property specification patterns for robotic missions. In ICSE '18: Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings. Association for Computing Machinery, New York, NY, USA, May 2018, pp. 434–435.

[107] Mislevy, R. J. Evidence-Centered Design for Simulation-Based Assessment. Military Medicine 178, 10S (Oct. 2013), 107–114.

[108] Mislevy, R. J., Behrens, J. T., Bennett, R. E., Demark, S. F., Frezzo, D. C., Levy, R., Robinson, D. H., Rutstein, D. W., Shute, V. J., Stanley, K., and Winters, F. I. On the Roles of External Knowledge Representations in Assessment Design. Journal of Technology, Learning, and Assessment 8, 2 (Jan. 2010).

[109] Molenaar, I., and Knoop-van Campen, C. Teacher Dashboards in Practice: Usage and Impact. In 12th European Conference on Technology Enhanced Learning, EC-TEL 2017 (Tallinn, Estonia, 2017), E. Lavoué, H. Drachsler, K. Verbert, J. Broisin, and M. Pérez-Sanagustín, Eds., Springer International Publishing, pp. 125–138.

[110] Molenaar, I., and Knoop-van Campen, C. A. N. How Teachers Make Dashboard Information Actionable. IEEE Transactions on Learning Technologies 12, 3 (July 2019), 347–355.

[111] Morandini, S., Fraboni, F., De Angelis, M., Puzzo, G., Diusino, D., and Pietrantoni, L. The Impact of Artificial Intelligence on Workers' Skills: Upskilling and Reskilling in Organisations. Informing Science: The International Journal of an Emerging Transdiscipline 26 (Feb. 2023), 039–068.

[112] Moreno-Marcos, P. M., Alario-Hoyos, C., Muñoz-Merino, P. J., and Kloos, C. D. Prediction in MOOCs: A Review and Future Research Directions. IEEE Transactions on Learning Technologies 12, 3 (2019), 384–401.

[113] Moringen, A., Rüttgers, S., Zintgraf, L., Friedman, J., and Ritter, H. Optimizing piano practice with a utility-based scaffold, 2021.

[114] Mosemann, H., and Wahl, F. Automatic decomposition of planned assembly sequences into skill primitives. IEEE Transactions on Robotics and Automation 17, 5 (2001), 709–718.

[115] Moya, A., Bastida, L., Aguirrezabal, P., Pantano, M., and Abril-Jiménez, P. Augmented Reality for Supporting Workers in Human–Robot Collaboration. MTI 7, 4 (Apr. 2023), 40.

[116] Mukherjee, A. A., Raj, A., and Aggarwal, S. Identification of barriers and their mitigation strategies for industry 5.0 implementation in emerging economies. International Journal of Production Economics 257 (2023), 108770.

[117] Munir, S., Stankovic, J. A., Liang, C. J. M., and Lin, S. Cyber physical system challenges for human-in-the-loop control. In 8th International Workshop on Feedback Computing (2013).

[118] Narvekar, S., and Stone, P. Learning Curriculum Policies for Reinforcement Learning. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (Richland, SC, May 2019), AAMAS '19, International Foundation for Autonomous Agents and Multiagent Systems, pp. 25–33.

[119] Ng, A. Y., and Russell, S. Algorithms for inverse reinforcement learning. In Proc. International Conf. on Machine Learning (ICML) (2000), Morgan Kaufmann, pp. 663–670.

[120] Nicholls, D., Sweet, L., and Hyett, J. Psychomotor skills in medical ultrasound imaging. Journal of Ultrasound in Medicine 33, 8 (2014), 1349–1352.

[121] Olney, A. M., Samei, B., Donnelly, P. J., and D'Mello, S. K. Assessing the Dialogic Properties of Classroom Discourse: Proportion Models for Imbalanced Classes. In Proceedings of the 10th International Conference on Educational Data Mining (2017), pp. 162–167.

[122] OpenAI. GPT-4V(ision) technical work and authors, 2023.

[123] Ortiz, A. C. Capturing, Modelling, Analyzing and providing Feedback in Martial Arts with Artificial Intelligence to support Psychomotor Learning Activities. Master's thesis, Universidad Nacional de Educación a Distancia, 2020.

[124] Ozkan, M. F., Rocque, A. J., and Ma, Y. Inverse reinforcement learning based stochastic driver behavior learning. CoRR (2021).

[125] Pakdamanian, E., Sheng, S., Baee, S., Heo, S., Kraus, S., and Feng, L. DeepTake: Prediction of Driver Takeover Behavior using Multimodal Data. In CHI Conference on Human Factors in Computing Systems (CHI '21) (January 2021).

[126] Pardos, Z. A., Bergner, Y., Seaton, D. T., and Pritchard, D. E. Adapting Bayesian knowledge tracing to a massive open online course in edX. Proceedings of the 6th International Conference on Educational Data Mining, EDM 2013 (2013).

[127] Pardos, Z. A., Gowda, S. M., Baker, R. S., and Heffernan, N. T. The sum is greater than the parts: ensembling models of student knowledge in educational software. ACM SIGKDD Explorations Newsletter 13, 2 (May 2012), 37.

[128] Pardos, Z. A., and Heffernan, N. T. KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. In International Conference on User Modeling, Adaptation, and Personalization (Berlin, Heidelberg, 2011), J. A. Konstan, R. Conejo, J. L. Marzo, and N. Oliver, Eds., Springer Berlin Heidelberg, pp. 243–254.

[129] Pardos, Z. A., Heffernan, N. T., Anderson, B. S., and Heffernan, C. L. Using Fine-Grained Skill Models to Fit Student Performance with Bayesian Networks. CRC Press, Oct. 2010.

[130] Pavlik, P. I., Cen, H., and Koedinger, K. R. Performance Factors Analysis – A New Alternative to Knowledge Tracing. In Proceedings of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems That Care: From Knowledge Representation to Affective Modelling (NLD, 2009), IOS Press, pp. 531–538.

[131] Pekrun, R. Emotion and Achievement During Adolescence. Child Development Perspectives 11, 3 (2017), 215–221.

[132] Pelánek, R. Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. User Modeling and User-Adapted Interaction 27 (2017), 313–350.

[133] Pelánek, R., and Řihák, J. Analysis and design of mastery learning criteria. New Review of Hypermedia and Multimedia 24, 3 (July 2018), 133–159.

[134] Pishchukhina, O., and Allen, A. Supporting learning in large classes: online formative assessment and automated feedback. In 2021 30th Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE) (Prague, Czech Republic, Sept. 2021), IEEE, pp. 1–4.

[135] Portaz, M., Corbi, A., Casas-Ortiz, A., and Santos, O. C. Exploring raw data transformations on inertial sensor data to model user expertise when learning psychomotor skills. User Model User-Adap Inter 2024 (Apr. 2024).

[136] Pretolesi, D., Zechner, O., Guirao, D. G., Schrom-Feiertag, H., and Tscheligi, M. AI-Supported XR Training: Personalizing Medical First Responder Training. In AI Technologies and Virtual Reality (Singapore, 2024), K. Nakamatsu, S. Patnaik, and R. Kountchev, Eds., Springer Nature Singapore, pp. 343–356.

[137] Profanter, S., Breitkreuz, A., Rickert, M., and Knoll, A. A hardware-agnostic opc ua skill model for robot manipulators and tools. In 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA) (2019), pp. 1061–1068.

[138] Pérez-D'Arpino, C., Khurshid, R. P., and Shah, J. A. Experimental Assessment of Human-Robot Teaming for Multi-Step Remote Manipulation with Expert Operators. J. Hum.-Robot Interact. 2023 (Oct. 2023), 3618258.

[139] Pérez-Ramírez, M., Ontiveros-Hernández, N. J., Ochoa-Ortíz, C. A., Hernández-Aguilar, J. A., and Zayas-Pérez, B. E. Intelligent Tutoring Systems based on Virtual Reality for the Electrical Domain. RCS 122, 1 (Dec. 2016), 163–174.

[140] Ramachandran, D., and Amir, E. Bayesian inverse reinforcement learning. In AAAI (01 2007), pp. 2586–2591.

[141] Ramos, C., and Yudko, E. "Hits" (not "Discussion Posts") predict student success in online courses: A double cross-validation study. Computers and Education 50, 4 (2008), 1174–1182.

[142] Rasekhipour, Y. Prioritized Obstacle Avoidance in Motion Planning of Autonomous Vehicles. PhD thesis, University of Waterloo, 2017.

[143] Rauter, G., Gerig, N., Sigrist, R., Riener, R., and Wolf, P. When a robot teaches humans: Automated feedback selection accelerates motor learning. Sci. Robot. 4, 27 (Feb. 2019).

[144] Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N. T., Koedinger, K. R., Junker, B., Ritter, S., Knight, A., Mercado, E., Turner, T. E., Upalekar, R., Walonoski, J. A., Macasek, M. A., Aniszczyk, C., Choksey, S., Livak, T., and Rasmussen, K. The Assistment project: Blending assessment and assisting. In Proceedings of the 12th Annual Conference on Artificial Intelligence in Education (Amsterdam, 2005), C. Looi, G. McCalla, B. Bredeweg, and J. Breuker, Eds., ISO Press, pp. 555–562.

[145] Risi, S., and Togelius, J. Increasing generality in machine learning through procedural content generation. Nature Machine Intelligence 2, 8 (2020), 428–436.

[146] Rodriguez-Triana, M., Prieto, L., Vozniuk, A., Boroujeni, M., Schwendimann, B., Holzer, A., and Gillet, D. Monitoring, Awareness and Reflection in Blended Technology Enhanced Learning: A Systematic Review. IJTEL 9, 2-3 (2017), 126–150.

[147] Roediger, H. L., and Butler, A. C. The critical role of retrieval practice in long-term retention. Trends in Cognitive Sciences 15, 1 (2011), 20–27.

[148] Rognon, C., Ramachandran, V., Wu, A. R., Ijspeert, A. J., and Floreano, D. Haptic feedback perception and learning with cable-driven guidance in exosuit teleoperation of a simulated drone. IEEE Transactions on Haptics 12, 3 (2019), 375–385.

[149] Rosen, J., Sinanan, M., and Hannaford, B. Objective Assessment of Surgical Skills. In Surgical Robotics: Systems Applications and Visions, J. Rosen, B. Hannaford, and R. M. Satava, Eds. Springer US, Boston, MA, 2011, pp. 619–649.

[150] Sadigh, D., and Kapoor, A. Safe control under uncertainty with probabilistic signal temporal logic. In Proceedings of Robotics: Science and Systems (AnnArbor, Michigan, June 2016).

[151] Sahebi, S., and Brusilovsky, P. Student performance prediction by discovering inter-activity relations. In Proceedings of the 11th International Conference on Educational Data Mining, EDM 2018 (2018).

[152] Salvucci, D. D. Modeling driver behavior in a cognitive architecture. Human Factors 48, 2 (2006), 362–380.

[153] Salvucci, D. D., and Gray, R. A two-point visual control model of steering. Perception 33, 10 (2004), 1233–1248.

[154] Santos, O. C. Training the Body: The Potential of AIED to Support Personalized Motor Skills Learning. Int J Artif Intell Educ 26, 2 (June 2016), 730–755.

[155] Schwab, K., and Zahidi, S. The Future of Jobs Report 2020. Tech. rep., World Economic Forum, Oct. 2020.

[156] Sheehan, F. H., McConnaughey, S., Freeman, R., and Zierler, R. E. Formative Assessment of Performance in Diagnostic Ultrasound Using Simulation and Quantitative and Objective Metrics. Military Medicine 184, Supplement_1 (Mar. 2019), 386–391.

[157] Shernoff, D. J. Optimal Learning Environments to Promote Student Engagement. Springer New York, New York, NY, 2013.

[158] Shute, V. J. Focus on Formative Feedback. Review of Educational Research 78, 1 (2008), 153–189.

[159] Shute, V. J., and Psotka, J. Intelligent tutoring systems: Past, present, and future. Scholastic, 1994, ch. 19, pp. 570–600.

[160] Sigrist, R., Rauter, G., Riener, R., and Wolf, P. Augmented visual, auditory, haptic, and multimodal feedback in motor learning: A review. Psychon Bull Rev 20, 1 (Feb. 2013), 21–53.

[161] Sinha, T., and Cassell, J. Connecting the Dots : Predicting Student Grade Sequences from Bursty MOOC Interactions over Time. In Proceedings of the Second (2015) ACM Conference on Learning @ Scale (New York, NY, USA, 2015), Association for Computing Machinery, pp. 249–252.

[162] Smith, G. M. Expressive Design Tools: Procedural Content Generation for Game Designers. Doctoral, University of California Santa Cruz, 2012.

[163] Spinuzzi, C. The methodology of participatory design. Technical communication 52, 2 (2005), 163–174.

[164] Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A., and Goodrich, M. Common metrics for human-robot interaction. In Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction (Salt Lake City Utah USA, Mar. 2006), ACM, pp. 33–40.

[165] Steinmetz, F., Wollschläger, A., and Weitschat, R. Razer—a hri for visual task-level programming and intuitive skill parameterization. IEEE Robotics and Automation Letters 3, 3 (2018), 1362–1369.

[166] Stone, C., Donnelly, P. J., Dale, M., Capello, S., Kelly, S., Godley, A., and D'Mello, S. K. Utterance-level Modeling of Indicators of Engaging Classroom Discourse. In The 12th International Conference on Educational Data Mining (Montreal, Canada, 2019), C. F. Lynch, A. Merceron, M. MDesmarais, and R. Nkambou, Eds., pp. 420–425.

[167] Sun, L., Zhan, W., Tomizuka, M., and Dragan, A. D. Courteous Autonomous Cars. In International Conference on Intelligent Robots (IROS) 2018 (8 2018).

[168] Suresh, A., Sumner, T., Huang, I., Jacobs, J., Foland, B., and Ward, W. Using deep learning to automatically detect talk moves in teachers'mathematics lessons. Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018 (2018), 5445–5447.

[169] Suresh, A., Sumner, T., Jacobs, J., Foland, B., and Ward, W. Automating Analysis and Feedback to Improve Mathematics' Teachers' Classroom Discourse. In Proceedings of the Ninth Symposium on Educational Advances in Artificial Intelligence (EAAI) (2019).

[170] Tabrez, A., Luebbers, M. B., and Hayes, B. Automated Failure-Mode Clustering and Labeling for Informed Car-To-Driver Handover in Autonomous Vehicles. In HRI'20Workshop on Assessing, Explaining, and Conveying Robot Proficiency for Human- Robot Teaming (2020), pp. 13–15.

[171] Thomas, U., Finkemeyer, B., Kroger, T., and Wahl, F. Error-tolerant execution of complex robot tasks based on skill primitives. In 2003 IEEE International Conference on Robotics and Automation (2003), vol. 3, pp. 3069–3075 vol.3.

[172] University of Iowa. National advanced driving simulator (nads) minisim. `https://nads.uiowa.edu/minisim`.

[173] Vygotsky, L. S. Mind in Society: The Development of Higher Psychological Processes. Harvard University Press, Cambridge, MA, 1978.

[174] Wang, J., Lu, M., and Li, K. Characterization of longitudinal driving behavior by measurable parameters. Transportation Research Record, 2185 (2010), 15–23.

[175] Wang, Y., and Heffernan, N. Extending Knowledge Tracing to Allow Partial Credit: Using Continuous versus Binary Nodes. In Artificial Intelligence in Education (Berlin, Heidelberg, 2013), H. C. Lane, K. Yacef, J. Mostow, and P. Pavlik, Eds., Springer Berlin Heidelberg, pp. 181–188.

[176] Wang, Z., Miller, K. F., and Cortina, K. S. Using the LENA in Teacher Training: Promoting Student Involvement through Automated Feedback. Unterrichtswissenshaft 4 (2013), 290–305.

[177] Wang, Z., Reed, I., and Fey, A. M. Toward Intuitive Teleoperation in Surgery: Human-Centric Evaluation of Teleoperation Algorithms for Robotic Needle Steering. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (Brisbane, QLD, May 2018), IEEE, pp. 5799–5806.

[178] Wilde, M., Chan, M., and Kish, B. Predictive human-machine interface for teleoperation of air and space vehicles over time delay. In 2020 IEEE Aerospace Conference (2020), pp. 1–14.

[179] Williamson, D. M., Mislevy, R. J., and Bejar, I. I., Eds. Automated Scoring of Complex Tasks in Computer-Based Testing. Lawrence Erlbaum, Mahwah, New Jersey, 2006.

[180] Xhakaj, F., Aleven, V., and McLaren, B. M. Effects of a Teacher Dashboard for an Intelligent Tutoring System on Teacher Knowledge, Lesson Planning, Lessons and Student Learning. In EC-TEL. Springer, 2017, pp. 315–329.

[181] Xin, L., Wang, P., Chan, C.-Y., Chen, J., Li, S. E., and Cheng, B. Intention-aware long horizon trajectory prediction of surrounding vehicles using dual lstm networks. In 2018 21st International Conference on Intelligent Transportation Systems (ITSC) (2018), pp. 1441–1446.

[182] Yang, J., Barragan, J. A., Farrow, J. M., Sundaram, C. P., Wachs, J. P., and Yu, D. An Adaptive Human-Robotic Interaction Architecture for Augmenting Surgery Performance Using Real-Time Workload Sensing—Demonstration of a Semi-autonomous Suction Tool. Hum Factors 66, 4 (Apr. 2024), 1081–1102.

[183] Yudelson, M. V., Koedinger, K. R., and Gordon, G. J. Individualized Bayesian Knowledge Tracing Models BT - Artificial Intelligence in Education. In Artificial Intelligence in Education (Berlin, Heidelberg, 2013), H. C. Lane, K. Yacef, J. Mostow, and P. Pavlik, Eds., Springer Berlin Heidelberg, pp. 171–180.

[184] Yuh, M. S., Rabb, E., Thorpe, A., and Jain, N. Using Reward Shaping to Train Cognitive–based Control Policies for Intelligent Tutoring Systems. In 2024 American Control Conference (ACC) (Toronto, ON, 2024), IEEE.

[185] Yuh, M. S.-T., Ortiz, K. R., Sommer-Kohrt, K. S., Oishi, M., and Jain, N. Classification of human learning stages via kernel distribution embeddings. IEEE Open Journal of Control Systems 3 (2024), 102–117.

[186] Zhang, Y., Lin, W. C., and Chin, Y.-K. S. A pattern-recognition approach for driving skill characterization. IEEE Transactions on Intelligent Transportation Systems 11, 4 (2010), 905–916.

[187] Zhou, F., Yang, X. J., and de Winter, J. C. F. Using eye-tracking data to predict situation awareness in real time during takeover transitions in conditionally automated driving. IEEE Transactions on Intelligent Transportation Systems 23, 3 (2022), 2284–2295.

[188] Ziebart, B. D., Maas, A. L., Bagnell, J. A., Dey, A. K., et al. Maximum entropy inverse reinforcement learning. In Aaai (2008), vol. 8, Chicago, IL, USA, pp. 1433–1438.

[189] Zotov, V., and Kramkowski, E. Moving-Target Intelligent Tutoring System for Marksmanship Training. Int J Artif Intell Educ 33, 4 (Dec. 2023), 817–842.