Automating the Go-Around Decision with Markov Decision Processes and Dynamic Policy Switching

Liam Kruse*
Stanford University, Stanford, CA, 94305

Final approach and landing are two of the most hazardous phases of flight, accounting for nearly two-thirds of all aviation accidents. An unstabilized approach, inclement weather, or obstacle incursion on the runway can induce a dangerous landing outcome such as an aircraft runway excursion or controlled flight into terrain. A timely, judicious decision to abort a landing and execute a go-around maneuver is a vital step to ensuring a safe landing outcome. However, studies have found a strong aversion to the go-around maneuver amongst pilots, with as many as 97% of unstabilized approaches resulting in go-around noncompliance. In this work I propose a "guardian angel" oversight system to model the landing sequence and determine whether to continue the landing process or execute a go-around maneuver. I formulate the problem as a Markov decision process (MDP). Such a system could either present advisories to the flight crew or override manual control with a provably safe autopilot system to perform the go-around sequence. Furthermore, I explore the benefits of on-the-fly substitution of different policies computed offline to reflect changing system dynamics.

I. Introduction

Although an aircraft's final approach and landing sequence represent a small fraction of total flight time, these two phases of flight account for almost 65% of all aviation accidents [1]. Adverse weather conditions, runway incursions, or an unstable approach—wherein an aircraft fails to meet speed, descent rate, and flight path criteria—all can result in sub-optimal landing conditions and catastrophic landing outcomes. Runway excursions represent the most common type of aviation accident [1]; however, flights might end with controlled flight into terrain or a collision with a ground-based or airborne obstacle if a flight crew persists in continuing a landing sequence in the face of adverse conditions.

The go-around (or "missed approach") maneuver is a procedure that occurs when a flight crew determines that one or more requirements for a safe landing are not satisfied. It consists of aborting the current landing attempt, rapidly gaining thrust and altitude, and rejoining the airfield traffic pattern until a safe landing can be executed. When executed properly, the go-around is a standard, safe maneuver that at worst serves as a minor inconvenience to passengers and at best averts a catastrophic landing outcome. However, studies have found widespread go-around noncompliance in both commercial and private aviation. For example, a 2017 study that placed ten commercially-rated pilots in a flight simulator found that half of the pilots persisted in erroneous landing decisions [2]. Additional studies have found go-around noncompliance during unstable approaches to be as high as 97% [1]. Rampant go-around noncompliance arises due to several factors. First, researchers have found that rational pilot decision-making is temporarily impaired during the landing sequence due to negative emotional consequences associated with the go-around maneuver [3]. These negative emotional consequences result in plan continuation error, wherein a flight plan is executed despite growing evidence that it is no longer safe [3]. Second, studies have shown that pilots are prone to psychosocial factors such as "get-home-itis" syndrome, wherein pilots become fixated on landing at all costs [3]. Finally, go-around noncompliance arises due to inherent dangers with the maneuver itself. A properly-executed missed approach is not considered to be a dangerous maneuver; however, if aircraft energy is not properly managed during a go-around, a flight can end with a stall and uncontrolled flight into terrain. Studies have found that one in ten go-arounds has a potentially dangerous outcome [1]. Thus, while it is certainly desirable to execute a go-around maneuver if the requirements for a safe landing are not met, it is not advisable to abort a landing unless the risk of landing outweighs the risk of the go-around.

One solution to the go-around noncompliance issue is the introduction of automated stable approach and landing alerting systems [1]. In this work I present a "guardian angel" oversight system that models the landing sequence as a Markov decision process (MDP). This system models the stochastic evolution of an aircraft during a final approach and seeks to balance the risks and rewards of continuing a landing sequence in the face of adverse conditions versus aborting

*Graduate Student Researcher, Department of Aeronautics and Astronautics, lkruse@stanford.edu

the landing. Such a system, if implemented as part of a comprehensive autopilot solution, could issue advisories on whether to continue or abort an approach or could execute a control authority switch to a backup autopilot system if it detected go-around noncompliance from the flight crew.

Unfortunately, MDPs fall victim to the "state-space explosion" problem, wherein exact solutions rapidly become intractable as the problem formulation becomes increasingly complex. Thus, it is desirable to either investigate approximate solvers or decompose an MDP into smaller problems. In this work I explore the possible benefits of decomposing an MDP problem into two smaller problem formulations, with each formulation representing the dynamics of a particular landing scenario. Specifically, I create an MDP to model a landing with a high risk of runway incursion (the "high-traffic" formulation) and an MDP to model a landing with a low risk of runway incursion (the "low-traffic" formulation). I demonstrate that judiciously selecting the "best" problem formulation at every time step and selecting actions according to its associated optimal policy yields a quantifiable performance improvement over action selection according to a fixed policy. Simulations with dynamic policy switching resulted in fewer aircraft crashes when compared to fixed policies. Note that in this work, the "best" problem formulation refers to the MDP that most closely models the true system dynamics at a given time step, i.e., the MDP whose transition function most closely captures the stochastic evolution of the system.

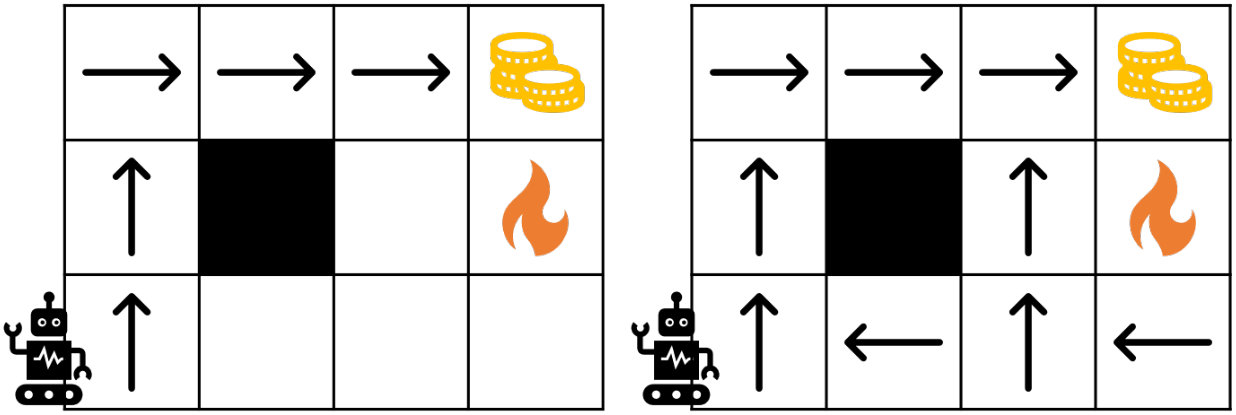
II. Related Work and Preliminaries

Although the go-around noncompliance issue represents one of the most consequential outstanding challenges in aviation, relatively little research has been conducted to develop solutions. Bro explores the utility of modeling the go-around decision with neural networks and attempts to predict aircraft approach outcomes; though this strategy achieves low classification error rates, it is dependent on a large training dataset and can be thrown off by atypical landing scenarios [4]. Kügler and Holzapfel modify the flight control system of the SAGITTA Demonstrator UAV to consider rejected takeoff and go-around decisions [5]. However, their approach only considers fixed controller and flight performance thresholds and is thus incabable of adapting to rapidly-evolving environments. Researchers have developed strategies for autonomous landing that rely on visual localization frameworks [6] and model predictive control [7]; however, these strategies have only been validated on small-scale unmanned aerial vehicles. Baomar and Bentley leverage artificial neural networks to autonomously handle landings and go-arounds in adverse weather conditions [8]. Their solution even generates a go-around flight course after a landing is aborted. Balachandran and Atkins develop a high-level decision-making system to make resilient control override decisions to prevent aircraft loss of control due to in-flight icing [9]. Although they do not tackle the go-around decision specifically, their work could be easily extended to deciding whether or not to abort a landing.

Specifically, Balachandran and Atkins formulate the control authority switching process as an MDP [9]. An MDP is a powerful framework for modeling sequential decision-making problems. Their ability to model stochasticity and represent problems with meaningful yet compact state abstractions makes them a compelling choice to model an aircraft landing process. Challenges with MDP-based strategies include the aforementioned state-space explosion problem and the necessity of having an accurate model of the landing scenario under consideration. Real-world landing scenarios are prone to change as aircraft objectives, constraints, and system dynamics evolve, rendering a static model useless. In this work I explore the benefits of dynamic policy switching as a strategy for handling evolving system dynamics. Recent advancements in MDP state-of-the-art explore other strategies to remedy these MDP shortcomings. Ong and Kochenderfer decompose a large multiagent MDP and fuse their solutions to generate advisories for aircraft to follow [10]; their strategy of decomposing an MDP is a close match to this publication, which seeks to split the go-around problem into multiple problem formulations to improve scalability. Liu and Sukhatme developed time-varying MDPs with stochastic state transitions that vary both spatially and temporally, which represents another solution for handling evolving system dynamics [11]. Li and Li focus on time-varying reward functions rather than time-varying transition models [12]. They study the reward difference between online policies and an optimal, potentially-nonstationary offline policy. Delgado, De Barros, Dias, and Sanner tackle the challenge of solving MDPs with imprecise transition probabilities (MDP-IPs) [13]. They develop algorithms for Stochastic Shortest Path MDP-IPs that are able to solve complex problems by focusing on reachable states.

A. Markov Decision Processes

A Markov decision process (MDP) is a framework for modeling sequential decision problems in stochastic environments. MDPs have been used to solve robotics [14], finance [15], and resource allocation problems [15], among others. Unlike deterministic optimization algorithms, MDPs account for probabilistic transitions within an agents' environment. This stochasticity can arise due to modeling errors, environmental disturbances, or agent actuator failure, and renders a single solution meaningless because an agent cannot be sure that every step of the solution will be followed. Thus, one must find the optimal action in every possible state to maximize the agent's expected cumulative reward. Such a mapping of states to actions is known as a *policy*. Figures 1a and 1b present the difference between an optimal *plan*, which may be found by a deterministic algorithm such as Dijkstra's Algorithm, and an optimal *policy*, which can be found by solving an MDP. These graphics present the canonical *gridworld* example, in which a robot agent must navigate around a barrier and fire obstacle to reach the reward at the top right.



(a) Sample optimal plan.

(b) Sample optimal policy.

Fig. 1 Plan versus policy.

Specifically, an MDP can be represented as the tuple $(\mathcal{S}, \mathcal{A}, T, R, \gamma)$. $\mathcal{S}$ is the set of system states, $\mathcal{A}$ represents the set of actions the agent can take, T is the state transition model $T(s'|s, a)$, and R is the reward function $S \times A \longrightarrow R$. The discount factor γ can be tuned to weight future rewards more or less heavily. This work focuses on discrete MDPs, which assume that the agent exists in discrete states and must choose from a discrete list of actions. However, both the state space and action space can be continuous in the general case.

At every time step, an agent chooses an action to maximize the expected cumulative discounted reward , or *utility* $U(s)$:

$$U(s) = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} R(s, a)\right] \tag{1}$$

where s is the agent's current state. The maximum expected reward is obtained by following the optimal policy π^*, which is the recommended action at every state:

$$\pi^*(s) = \underset{\pi}{\arg\max} U^{\pi}(s) \tag{2}$$

where U^{π} is referred to as the *value function*. The value function can be found by iteratively applying the Bellman Equation:

$$U_{k+1}(s) = \max_{a}\left(R(s, a) + \gamma \sum_{s'} T(s'|s, a) U^*(s')\right) \tag{3}$$

where $R(s, a)$ is the reward for taking action a in state s and $U^*(s')$ is the utility of the next state assuming optimal play from the agent.

III. MDP Formulation for the Go-Around Decision with Dynamic Policy Switching

The scope of this work is to explore the benefits of on-the-fly policy switching rather than model fidelity. Thus, I seek a compact yet meaningful representation of the aircraft landing process to facilitate rapid prototyping. In this section I describe my MDP formulation before presenting experimental results in Section IV.

A. State and Action Spaces

The aircraft landing sequence occurs in continuous time and space; I seek a high-level abstraction with discrete state and action sequences. I define the aircraft state space to be the tuple

$$S = (P, R) \tag{4}$$

where P represents the current phase of flight and R represents the risk of encountering a dangerous runway incursion. Table 1 presents the breakdown of the discretized state space.

Table 1 Discretized State Space

P	Value	R	Value
P_1	Base Leg	R_1	Low Incursion Risk
P_2	Final Approach	R_2	Medium Incursion Risk
P_3	Landing	R_3	High Incursion Risk

Thus, there are a total of $(3)(3) = 9$ total states in the state space. Combining the state variables at a given time step reveals the state of the aircraft. For example, an aircraft in state P_1R_1 is on the base leg of the landing process and faces a low risk of runway incursion, whereas an aircraft at state P_3R_3 is landing and faces a high risk of runway incursion.

At every time step the aircraft can choose to either *Continue* with the landing sequence or to *Go Around* and abort the landing, immediately terminating the sequence. Table 1 displays the discretized action space.

Table 2 Discretized Action Space

A	Action
A_1	Continue
A_2	Go Around

B. Transition Matrix Formulation

I generated transition matrices for both actions by hand. To model different landing scenarios, I create two separate sets of transition matrices. The first set of matrices (the "high-traffic" formulation) models a landing scenario wherein the agent is attempting to land at a busy airfield and faces an elevated risk of a runway incursion. The second set of matrices (the "low-traffic" formulation) models a landing scenario wherein the agent is attempting to land at a relatively empty airfield and faces a decreased risk of a runway incursion.

For both the low-traffic and high-traffic transition models, repeated selection of the *Continue* action moves the agent deterministically from the base leg phase to the final approach phase and eventually to the landing phase. However, the *Risk* state variable stochastically varies between low, medium, and high risks of incursion. The key distinction between the low-traffic and the high-traffic formulation is that the agent has a greater probability of transitioning to a state with a elevated risk of runway incursion when selecting the *Continue* action in the high-traffic formulation.

Selecting the *Go Around* action returns the agent, with high probability, to the state P_1R_1 (base leg phase with low risk of runway incursion). However, the *Go Around* action can also move the agent to the state P_3R_3 (landing phase with high risk of runway incursion) with nonzero probability. This attempts to simulate the tradeoff between the risks and rewards of the go-around maneuver. It should be noted that transition likelihoods are exaggerated for simulation purposes. For example, the actual occurrence rate of runway incursions is extremely small. I deliberately elevate this likelihood to present a compelling toy problem.

To see the complete set of transition matrices, please refer to the Appendix.

C. Reward Function Formulation

The reward function defines the utility that an agent expects to receive from taking action a in state s. I hand-tuned reward functions to imitate plausible real-world utilities and promote realistic agent behavior. For example, I heavily penalized the *Continue* action when the agent is in the final approach or landing phase and faces a high risk of a runway incursion, as such a scenario is likely to end with a collision with a ground-based obstacle. Conversely, I heavily penalized the *Go Around* action when the agent is in the base leg phase and faces low risk of runway incursion; it is ill-advised to conduct a go-around maneuver in this instance, as it inconveniences passengers and nearby aircraft without increasing safety.

To see the complete set of reward functions, please refer to the Appendix.

D. MDP Solution and Policy Extraction

I modeled the high-traffic and low-traffic MDP formulations in Python using `numpy` and the `mdptoolbox`* package. I found the optimal policy for both MDP formulations using 10 iterations of Gauss Seidel value iteration. As expected, the optimal policy for the high-traffic MDP formulation is more "cautious" than the optimal policy for the low-traffic MDP formulation, as it recommends the *Go Around* action for more states. Tables 3 and 4 display the optimal policies for the high-traffic and low-traffic MDP formulations.

Table 3 High-Traffic Policy

State	P_1R_1	P_1R_2	P_1R_3	P_2R_1	P_2R_2	P_2R_3	P_3R_1	P_3R_2	P_3R_3
Action	A_1	A_1	A_1	A_1	A_1	A_2	A_1	A_2	A_2

Table 4 Low-Traffic Policy

State	P_1R_1	P_1R_2	P_1R_3	P_2R_1	P_2R_2	P_2R_3	P_3R_1	P_3R_2	P_3R_3
Action	A_1	A_1	A_1	A_1	A_1	A_1	A_1	A_1	A_2

IV. Experimental Results

My objectives in this work are two-fold. First, I seek to determine whether or not an aircraft should continue with the landing process by evaluating the stochastic state transitions and expected rewards during the landing sequence. Second, I seek to quantify the extent to which dynamically switching between the high-traffic and low-traffic MDP policies to more closely match the system's true dynamics improves planner performance versus assuming a fixed policy. Specifically, I identify the optimal policy at each time step (e.g., if the system is evolving according to the high-traffic MDP formulation, then I select the high-traffic policy) and select the corresponding action for the agent's current state. In this work, I assume omniscient knowledge of the system's transition dynamics, and simply choose the policy that corresponds to the governing MDP model. In reality, a high-level decision-making system such as an oversight partially observable Markov decision process (POMDP) could be used to monitor the agent's environment and select the policy at a given time step. I compare the performance of this on-the-fly policy switching strategy with the baseline performance achieved by only considering the high-traffic or the low-traffic policies.

A. Experimental Mechanics

I developed a simple landing simulator in Python to model a toy landing sequence. The mechanics of the landing sequence with dynamic policy switching are straightforward:

1) The aircraft is randomly assigned to a state in the base leg phase of the landing sequence (P_1R_1, P_1R_2, or P_1R_3) according to a specified probability distribution. For my experiments, I set P(P_1R_1) = 0.6, P(P_1R_2) = 0.3, and P(P_1R_3) = 0.1.
2) The MDP formulation for the current time step is randomly selected; the high-traffic formulation is selected with 50% probability, and the low-traffic formulation is selected with 50% probability.
3) The aircraft selects an action according to the selected MDP's optimal policy. If the *Continue* action is chosen, then the aircraft transitions to a state in the final approach phase of the landing sequence (P_2R_1, P_2R_2, or P_2R_3) according to the probabilities specified in the selected MDP's transition model. If the *Go Around* action is

[*] https://pymdptoolbox.readthedocs.io/en/latest/api/mdptoolbox.html

selected, then the aircraft transitions to either P_1R_1 or P_3R_3 according to the selected MDP's transition model and the simulation ends.

4) If the *Continue* action was selected, then Steps 2 and 3 are repeated, this time with the agent starting in the final approach phase of the landing sequence. If the agent chooses the *Continue* action, then the aircraft transitions to a state in the landing phase of the landing sequence (P_3R_1, P_3R_2, or P_3R_3).
5) If the agent arrives a state in the landing phase of the landing sequence, then it chooses one final action according to the policy of the most recently selected MDP formulation. If it chooses the *Continue* action, then it probabilistically transitions to a state in the landing phase (which includes the state it is currently in). If it chooses the *Go Around* action, then it transitions to either P_1R_1 or P_3R_3.
6) The simulation ends either after the agent chooses an action in the landing phase and completes its transition, or after the agent chooses the *Go Around* action. A simulation ends in a "CRASH" if the aircraft ends the simulation in P_3R_3; otherwise, the aircraft is "SAFE".

The goal is to simultaneously minimize the number of crashes and minimize the number of go-around maneuvers. Figure 2 shows a sample simulation.

Time Step	1	2	3
Start State	P_1R_1	P_2R_2	P_3R_2
MDP Formulation	Low Traffic	High Traffic	High Traffic
Action	Continue	Continue	Go Around
End State	P_2R_2	P_3R_2	P_1R_1
Simulation Outcome			**SAFE**

Fig. 2 Sample Simulation Run.

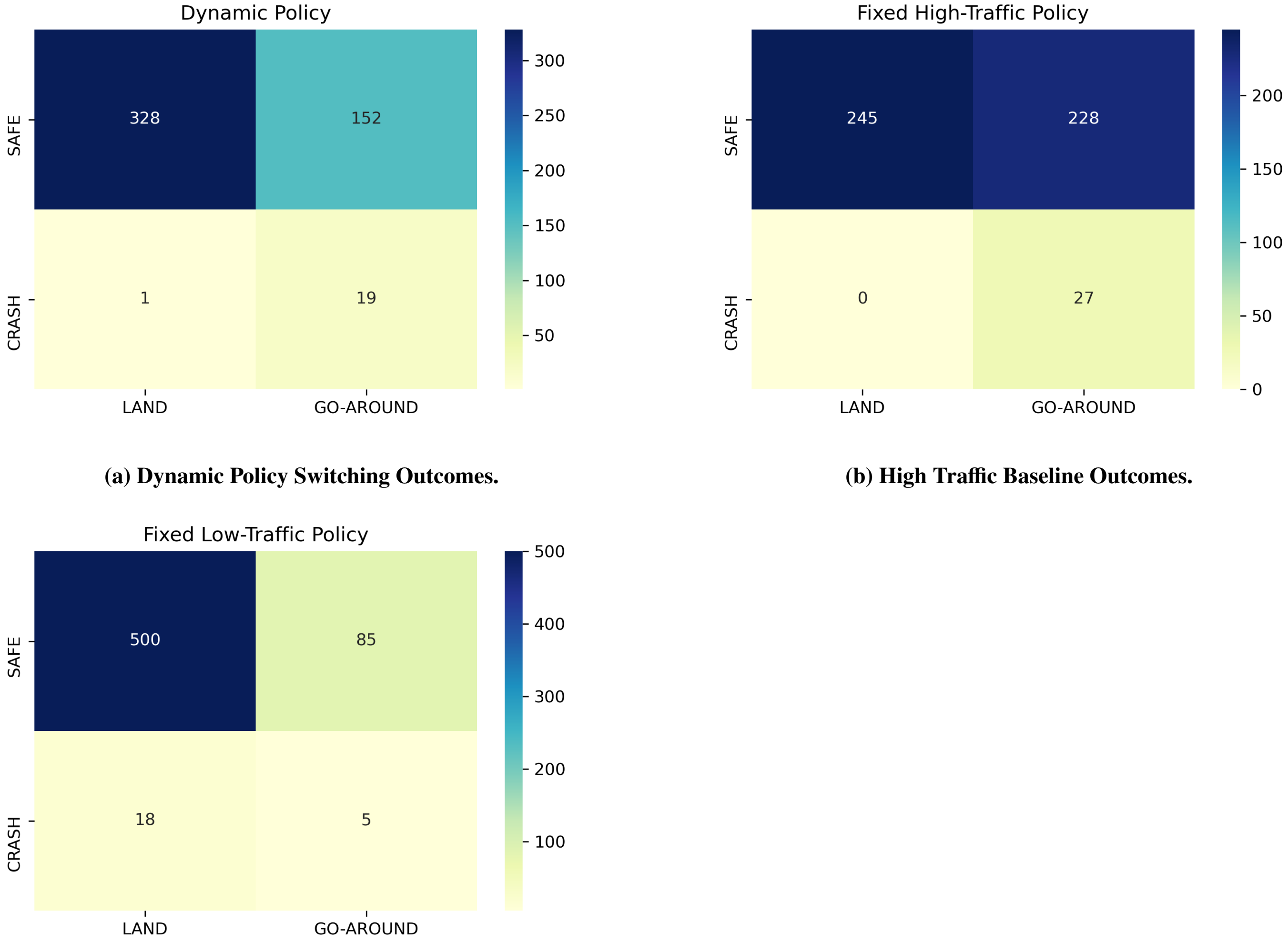
The baseline tests (wherein the true system dynamics evolve but actions are only chosen from a single policy) proceed in much the same way as the dynamic policy switching simulations. The only distinction is that actions are always chosen from the high-traffic optimal policy in the high-traffic baseline test, and from the low-traffic optimal policy in the low-traffic baseline test. The true system dynamics continue to be updated at every time step, however; thus, the aircraft transitions from state to state according to the most recently selected MDP's transition model. The implication is that the aircraft might find itself choosing actions from a cautious policy even though it faces a low risk of runway incursion, or might choose actions from an aggressive policy even though it faces a high risk of runway incursion.

B. Simulations

I ran 500 simulations each for the dynamic policy-switching test, high-traffic baseline test, and the low-traffic baseline test. I used the same starting state for all three procedures for each simulation, and then let each simulation evolve stochastically on its own. As previously stated, each simulation ends when the agent chooses an action from a state in the landing phase, or when the agent chooses the *Go Around* action. If the agent ends the simulation in state P_3R_3, then the trial is considered to be a "CRASH". If the agent ends the simulation in any other state, then the trial was "SAFE". I present the final outcomes for all three procedures in Figure 3, considering the agent's final action and the outcome of its final transition. Figure 4 displays the number of crashes that the agent totals when following each procedure.

C. Results and Discussion

Figure 3 displays the outcomes of all three procedures. Selecting actions while holding the high-traffic policy fixed represents one extreme scenario; we see these results in Figure 3b. Recall that the high-traffic policy is a cautious policy wherein the agent elects to *Go Around* in three of the nine states. Thus, we see that the agent executes the go-around maneuver at a very high rate (slightly over 50% of the time) when choosing actions solely in accordance with the



(a) Dynamic Policy Switching Outcomes.

(b) High Traffic Baseline Outcomes.

(c) Low Traffic Baseline Outcomes.

Fig. 3 Simulation Outcomes.

high-traffic MDP's optimal policy. The aircraft never crashes while attempting to land during the 500 simulations; this is likely due to the fact that it simply chooses not to land during a high percentage of landing sequences. However, it crashes on 27 of its go-around attempts. Furthermore, such a high rate of go-around maneuvers would likely cause a great deal of consternation amongst passengers and airlines alike, especially if many of the aborted landings were not warranted.

Selecting actions while holding the low-traffic policy fixed represents the other extreme scenario; we see these results displayed in Figure 3c. The low-traffic policy represents an aggressive policy wherein the agent only elects to *Go Around* in one of the nine states. Consequently, we see a relatively low go-around rate during this procedure, with the agent only electing to *Go Around* in 18% of the simulations. However, the agent likely should have chosen to *Go Around* more frequently, as it crashes while attempting to land on 18 occasions. When combined with its five crashes during a go-around maneuver, the agent totals 23 crashes when solely choosing actions according to the low-traffic MDP's optimal policy.

The dynamic policy switching procedure represents a tradeoff between these extremes. Flight outcomes when selecting actions according to the optimal policy at every time step are presented in Figure 3a. The agent is not as cautious as when it solely follows the high-traffic policy—it only elects to perform a go-around on about 34% of simulations. However, the agent is not as insistent on landing in the face of elevated runway incursion risk as it is when

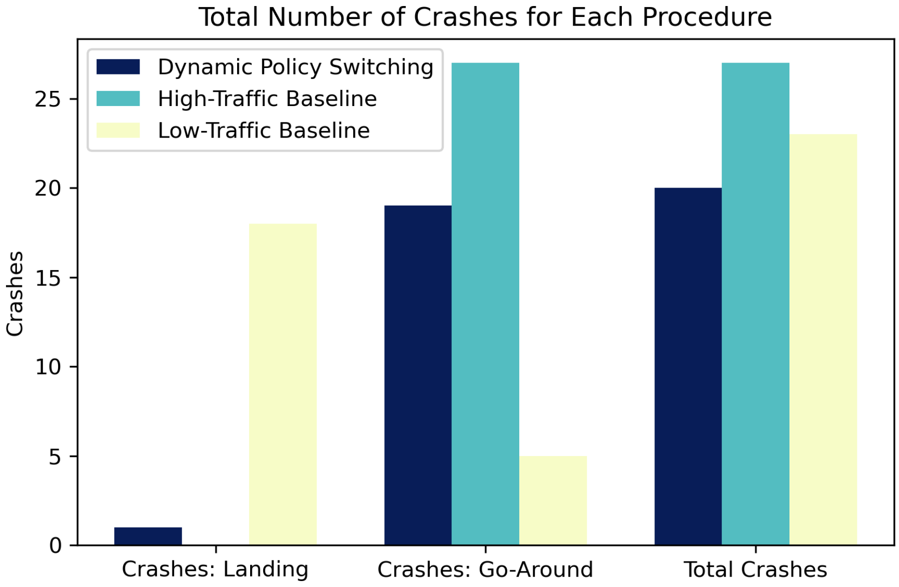


Fig. 4 Crashes Breakdown.

solely following the low-traffic policy. The agent only crashes a single time while attempting to land.

V. Conclusion

In this work I explore the benefits of decomposing a complex MDP problem into multiple problem formulations to represent different system dynamics. I solve both formulations offline to acquire an optimal policy for each formulation. I conduct experiments to quantify the extent to which judiciously selecting the best policy at a given time step can improve performance versus following a fixed policy. In my landing sequence simulator, the aircraft agent crashed on 20 of 500 simulations when performing dynamic policy switching, as opposed to 23 and 27 times when choosing actions according to fixed policies. Additionally, this work serves as a cursory exploration of the go-around noncompliance issue, which is a significant outstanding problem in aviation.

In future work, I intend to develop more complex representations of the aircraft landing sequence. I will explore high-level decision-making systems for determining which policy is "best" at a given time-step, rather than relying on omniscient knowledge to substitute in the best policy. Finally, I will explore more complex models, such as partially observable or time-varying Markov decision processes, to more fully capture the uncertainty inherent in complex decision-making problems.

Appendix

	P_1R_1	P_1R_2	P_1R_3	P_2R_1	P_2R_2	P_2R_3	P_3R_1	P_3R_2	P_3R_3
P_1R_1	0	0	0	0.6	0.3	0.1	0	0	0
P_1R_2	0	0	0	0.3	0.4	0.3	0	0	0
P_1R_3	0	0	0	0.1	0.3	0.6	0	0	0
P_2R_1	0	0	0	0	0	0	0.6	0.3	0.1
P_2R_2	0	0	0	0	0	0	0.3	0.4	0.3
P_2R_3	0	0	0	0	0	0	0.1	0.3	0.6
P_3R_1	0	0	0	0	0	0	0.8	0.2	0
P_3R_2	0	0	0	0	0	0	0.1	0.8	0.1
P_3R_3	0	0	0	0	0	0	0	0.2	0.8

(a) High-Traffic *Continue* Transition Model.

	P_1R_1	P_1R_2	P_1R_3	P_2R_1	P_2R_2	P_2R_3	P_3R_1	P_3R_2	P_3R_3
P_1R_1	0.95	0	0	0	0	0	0	0	0.05
P_1R_2	0.95	0	0	0	0	0	0	0	0.05
P_1R_3	0.95	0	0	0	0	0	0	0	0.05
P_2R_1	0.9	0	0	0	0	0	0	0	0.1
P_2R_2	0.9	0	0	0	0	0	0	0	0.1
P_2R_3	0.9	0	0	0	0	0	0	0	0.1
P_3R_1	0.9	0	0	0	0	0	0	0	0.1
P_3R_2	0.9	0	0	0	0	0	0	0	0.1
P_3R_3	0.9	0	0	0	0	0	0	0	0.1

(b) High-Traffic *Go Around* Transition Model.

Fig. 5 High-Traffic Transition Model.

	P_1R_1	P_1R_2	P_1R_3	P_2R_1	P_2R_2	P_2R_3	P_3R_1	P_3R_2	P_3R_3
P_1R_1	0	0	0	0.75	0.2	0.05	0	0	0
P_1R_2	0	0	0	0.4	0.4	0.2	0	0	0
P_1R_3	0	0	0	0.2	0.4	0.4	0	0	0
P_2R_1	0	0	0	0	0	0	0.75	0.2	0.05
P_2R_2	0	0	0	0	0	0	0.4	0.4	0.2
P_2R_3	0	0	0	0	0	0	0.75	0.2	0.05
P_3R_1	0	0	0	0	0	0	0.8	0.2	0
P_3R_2	0	0	0	0	0	0	0.3	0.6	0.1
P_3R_3	0	0	0	0	0	0	0.1	0.3	0.6

(a) Low-Traffic *Continue* Transition Model.

	P_1R_1	P_1R_2	P_1R_3	P_2R_1	P_2R_2	P_2R_3	P_3R_1	P_3R_2	P_3R_3
P_1R_1	0.95	0	0	0	0	0	0	0	0.05
P_1R_2	0.95	0	0	0	0	0	0	0	0.05
P_1R_3	0.95	0	0	0	0	0	0	0	0.05
P_2R_1	0.9	0	0	0	0	0	0	0	0.1
P_2R_2	0.9	0	0	0	0	0	0	0	0.1
P_2R_3	0.9	0	0	0	0	0	0	0	0.1
P_3R_1	0.9	0	0	0	0	0	0	0	0.1
P_3R_2	0.9	0	0	0	0	0	0	0	0.1
P_3R_3	0.9	0	0	0	0	0	0	0	0.1

(b) Low-Traffic *Go Around* Transition Model.

Fig. 6 Low-Traffic Transition Model.

	P_1R_1	P_1R_2	P_1R_3	P_2R_1	P_2R_2	P_2R_3	P_3R_1	P_3R_2	P_3R_3
P_1R_1	0	-1	-2	0	-1	-2	0	-1	-2
P_1R_2	-1	-3	-5	-1	-3	-5	-1	-3	-5
P_1R_3	-3	-5	-10	-3	-5	-10	-3	-5	-10
P_2R_1	0	-3	-5	0	-3	-5	0	-3	-5
P_2R_2	-3	-5	-10	-3	-5	-10	-3	-5	-10
P_2R_3	-5	-10	-15	-5	-10	-15	-5	-10	-15
P_3R_1	0	-5	-10	0	-5	-10	0	-5	-10
P_3R_2	-5	-10	-20	-5	-10	-20	-5	-10	-20
P_3R_3	-10	-20	-30	-10	-20	-30	-10	-20	-30

(a) *Continue* Reward Function.

	P_1R_1	P_1R_2	P_1R_3	P_2R_1	P_2R_2	P_2R_3	P_3R_1	P_3R_2	P_3R_3
P_1R_1	-30	-25	-20	-30	-25	-20	-30	-25	-20
P_1R_2	-25	-20	-15	-25	-20	-15	-25	-20	-15
P_1R_3	-20	-15	-10	-20	-15	-10	-20	-15	-10
P_2R_1	-30	-25	-20	-30	-25	-20	-30	-25	-20
P_2R_2	-25	-20	-15	-25	-20	-15	-25	-20	-15
P_2R_3	-15	-10	-5	-15	-10	-5	-15	-10	-5
P_3R_1	-30	-25	-20	-30	-25	-20	-30	-25	-20
P_3R_2	-20	-15	-10	-20	-15	-10	-20	-15	-10
P_3R_3	-10	-5	0	-10	-5	0	-10	-5	0

(b) *Go Around* Reward Function.

Fig. 7 Reward Function.

References

[1] Blajev, T., and Curtis, W., "Go-Around Decision-Making and Execution Project: Final Report to Flight Safety Foundation," *Flight Safety Foundation, March*, 2017.

[2] Reynal, M., Rister, F., Scannella, S., Wickens, C., and Hehais, F., "Investigating pilot's decision making when facing an unstabilized approach: an eye-tracking study," *19th International Symposium on Aviation Psychology*, 2017, p. 335.

[3] Causse, M., Dehais, F., Péran, P., Sabatini, U., and Pastor, J., "The effects of emotion on pilot decision-making: A neuroergonomic approach to aviation safety," *Transportation research part C: emerging technologies*, Vol. 33, 2013, pp. 272–281.

[4] Bro, J., "FDM Machine Learning: An investigation into the utility of neural networks as a predictive analytic tool for go around decision making," *Journal of Applied Sciences and Arts*, Vol. 1, No. 3, 2017, p. 3.

[5] Kügler, M. E., and Holzapfel, F., "Online self-monitoring of automatic take-off and landing control of a fixed-wing UAV," *2017 IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, 2017, pp. 2108–2113.

[6] Kong, W., Hu, T., Zhang, D., Shen, L., and Zhang, J., "Localization framework for real-time uav autonomous landing: An on-ground deployed visual approach," *Sensors*, Vol. 17, No. 6, 2017, p. 1437.

[7] Feng, Y., Zhang, C., Baek, S., Rawashdeh, S., and Mohammadi, A., "Autonomous landing of a UAV on a moving platform using model predictive control," *Drones*, Vol. 2, No. 4, 2018, p. 34.

[8] Baomar, H., and Bentley, P. J., "Autonomous landing and go-around of airliners under severe weather conditions using artificial neural networks," *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, IEEE, 2017, pp. 162–167.

[9] Balachandran, S., and Atkins, E. M., "Flight safety assessment and management to prevent loss of control due to in-flight icing," *AIAA Guidance, Navigation, and Control Conference*, 2016, p. 0094.

[10] Ong, H. Y., and Kochenderfer, M. J., "Markov decision process-based distributed conflict resolution for drone air traffic management," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 1, 2017, pp. 69–80.

[11] Liu, L., and Sukhatme, G. S., "A solution to time-varying Markov decision processes," *IEEE Robotics and Automation Letters*, Vol. 3, No. 3, 2018, pp. 1631–1638.

[12] Li, Y., and Li, N., "Online Learning for Markov Decision Processes in Nonstationary Environments: A Dynamic Regret Analysis," *2019 American Control Conference (ACC)*, IEEE, 2019, pp. 1232–1237.

[13] Delgado, K. V., De Barros, L. N., Dias, D. B., and Sanner, S., "Real-time dynamic programming for Markov decision processes with imprecise probabilities," *Artificial Intelligence*, Vol. 230, 2016, pp. 192–223.

[14] Cassandra, A. R., "A survey of POMDP applications," *Working notes of AAAI 1998 fall symposium on planning with partially observable Markov decision processes*, Vol. 1724, 1998.

[15] White, D. J., "A survey of applications of Markov decision processes," *Journal of the operational research society*, Vol. 44, No. 11, 1993, pp. 1073–1096.