A REAL TIME EXPERT CONTROL SYSTEM FOR HELICOPTER AUTOROTATION

A Thesis

by

ZACHARY NOLAN SUNBERG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Jonathan Rogers
Committee Members,	Suman Chakravorty
	Reza Langari
	Raktim Bhattacharya
	Sivakumar Rathinam
Department Head,	Rodney Bowersox

May 2013

Major Subject: Aerospace Engineering

ABSTRACT

Autorotation maneuvers are required to perform a safe landing of a helicopter in cases of engine loss in a single engine vehicle and transmission or tail rotor malfunction. The rise of autonomous helicopter technology, and the pilot skill required to manually perform an autorotation, motivate the need for new autonomous autorotation control laws. Previous approaches to automatic control for this maneuver have relied on control law optimization based on a high-fidelity model of the helicopter, or have attempted to match recorded trajectories flown by an expert human pilot. In this paper, a new expert control system is proposed. The term "expert control system" is used because the system is intended to mimic the actions that a human pilot might take, does not require any iterative learning, model prediction, or optimization at runtime, and is based on an inference system that involves fuzzy logic, PID, and other conventional control techniques. The multi-stage control law drives the helicopter to a near-optimal steady-state descent and uses an estimate of the time to impact to safely flare and land the helicopter in the vast majority of flight conditions. The control law is validated using a full 6-degree-of-freedom simulation of both a full-size attack helicopter and a small hobby-class helicopter. The proposed control design is highly flexible and may be used to perform fully autonomous autorotation or to provide guidance to pilots during manual autorotation maneuvers.

DEDICATION

To my father, for teaching me how to think well, and to my mother, for helping me all the times when thinking like an engineer is not the answer.

ACKNOWLEDGEMENTS

I would like to first thank my advisor, Dr. Jonathan Rogers, for making this work possible. He has provided every resource I have needed, and created a strong community and great place for me to work in his lab. His instruction, advice, patience, and guidance were essential to my education.

I would also like to thank the members of my committee: Dr. Bhattacharya, for giving me an important learning experience through one of my first research projects as an undergraduate student; Dr. Chakravorty, for instilling in me a desire for rigor in my research through his class instruction and for giving me the opportunities that have made my further graduate education possible; Dr. Langari, for introducing me to concepts that made the controller described in this report possible; and Dr. Rathinam, for graciously serving on my committee and giving feedback and advice on my thesis.

The other members of the HUSL Lab, Nate, Lee, Bob, Trevor, and Brad have taught me a great deal and made my time here enjoyable. Brad Taylor deserves special thanks for helping to develop the helicopter simulation software used in this research (indeed, he derived all of the really difficult equations needed for that code).

Finally, I would like to thank the National Science Foundation for making my graduate education possible through their generous funding. This material is based upon work supported by Grant No. 1252521 specifically.

NOMENCLATURE

`FAST_COL_INCREASE`	Collective adjustment rate for rapid adjustments during the flare and landing phases (See Table 2.1)
h	Altitude above ground level ($-z$)
$[I_c]_B$	Moment of inertia matrix about the center of mass resolved in the body frame
I_R	Main rotor moment of inertia about shaft
`K_COL`	Rotor collective gain for flare and landing phases (See Table 2.1)
`K_D_SS`	Gain on rotor speed time derivative for collective control during steady-state descent (See Table 2.1)
KE	Kinetic energy
`K_P_SS`	Gain on rotor speed for collective control during steady-state descent (See Table 2.1)
`LANDING_MAX_ANGLE`	Maximum cap on roll and pitch angles during the landing phase (See Table 2.1)
m	Helicopter mass
p, q, r	Helicopter angular rates in body frame
`PRE_FLARE_MAX_ANGLE`	Maximum cap on roll and pitch angle during the pre-flare phase (See Table 2.1)
R	Main rotor radius
r	Radial distance from rotor hub

RPM_AUTO	Desired main rotor rotation rate for the steady state descent phase (See Table 2.1)
TAU	Rotor collective adjustment time constant tuning parameter for flare and landing phases (See Table 2.1)
TOUCHDOWN_MAX_ANGLE	Maximum cap on roll and pitch angles during the touchdown phase (See Table 2.1)
TOUCHDOWN_COL_DECREASE	Constant collective pitch rate during touchdown phase (See Table 2.1)
TTI_F	Desired time to impact during the flare phase (See Table 2.3)
TTI_F_MAX	Maximum cap on the desired time to impact during the flare phase (See Table 2.1)
TTI_L	Desired time to impact during the landing phase (See Tables 2.1 and 2.3)
$TTLE$	Time to landing entry (See Table 2.3)
U_AUTO	Desired forward speed for the steady state descent phase (See Table 2.1)
u_{des}	Desired forward velocity
U_TOUCHDOWN	Desired forward velocity at touchdown (See Table 2.1)
u, v, w	Helicopter velocity components in the body frame
x, y, z	Helicopter position components in a North-East-down coordinate system
β	Blade flapping angle
β_0, β_{1c}, β_{1s}	First harmonic blade flapping states (See Section 3.3)

η	Combined roll and pitch angle $\eta = \sqrt{\phi^2 + \theta^2}$
η_{max}	Maximum limit on combined roll and pitch angle
θ_0	Main rotor collective blade pitch
θ_{1c}	Main rotor lateral cyclic blade pitch
θ_{1s}	Main rotor longitudinal cyclic blade pitch
θ_{tr}	Tail rotor collective blade pitch
λ_i	Main rotor induced inflow
λ_0, λ_c, λ_s	Component states of induced inflow distribution (See Equation 3.7)
μ_{small}	Membership function defining fuzzy set of small velocities
ϕ, θ, ψ	Roll, pitch, and yaw Euler angles
ψ_{MR}	Main rotor azimuth angle
Ω	Main rotor rotation rate

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Helicopters are unique in their ability to provide efficient, flexible, and reliable vertical takeoff and landing flight. However, because vertical lift is derived directly from main rotor thrust (rather than indirectly through a wing as in fixed-wing aircraft), they are much less forgiving than conventional aircraft in the event of power loss. When a single engine helicopter encounters engine failure, or when any helicopter suffers a catastrophic transmission or tail rotor failure, a pilot must perform autorotation to bring the helicopter to a safe landing. In the autorotation maneuver, the engine does not provide power to the main rotor. Instead, the pilot uses the air flowing through the rotor to maintain main rotor kinetic energy, enabling some measure of control of the aircraft and allowing the pilot to slow the helicopter before landing to minimize total velocity at impact.

An autorotative descent consists of entry, a steady state descent towards a suitable landing site, and a flaring maneuver to dramatically reduce kinetic energy immediately before landing. If entry into the autorotation is delayed or if the maneuver is otherwise executed poorly, the rotor rotation speed may drop to a level that is too low for proper control, provides insufficient energy for the flare, or leads to excessive blade flapping. These considerations result in a set of restricted height and velocity combinations known as the "dead man's curve," typically plotted on a so-called Height-Velocity diagram, from which a successful autorotation is unlikely [1]. For this reason, these conditions are largely avoided by pilots of single engine rotorcraft. As single engine autonomous rotorcraft of all sizes become more prevalent, automatic control laws for autorotation that protect expensive equipment and possibly human passengers in cases of engine failure must be developed. These control laws may also

be able to help human pilots autorotate safely from well within the "avoid" region of the H-V diagram by providing real-time guidance in an advanced avionics system.

The autorotation maneuver requires significant pilot skill and historically has not been carried out by automatic control systems. An early study of the autorotation problem from an automatic control perspective is Johnson's 1977 study on optimal autorotation descent [2]. In this work, optimal descent trajectories and control laws are calculated, but the results were intended to be used for vehicle design rather than a real-time feedback controller. In the 1980s, Lee et al improved on this initial analysis and suggested that the flight envelope for safe autorotation could be significantly enlarged using automatic control [3], and indeed a patent for such a system using iterative optimization techniques was filed in 2011 [4].

Recent work has explored the autorotation problem for small, unmanned helicopters in the 5-150 lb range. This class of unmanned vehicle could be used for observation and search and rescue support, among other potential missions. Hazawa et al. created a traditional PI controller for landing a small helicopter [5]. Recently, two other concepts for autorotation controllers on small helicopters have been proposed. Abbeel et al. created a reinforcement learning controller that attempts to match trajectories recorded from autorotative descents by an expert pilot. This is the approach that has been most successfully validated [6]. Dalamagkidis et al. used a one-dimensional vertical autorotation model and receding horizon neural-network optimization to create a controller which was tested in simulation using a one-dimensional model of a small helicopter. This controller was also tested for full-size helicopters using both the one-dimensional model and X-Plane simulation software [7]. It should be noted that the full size aircraft simulated in this work was modified with a High Energy Rotor System (HERS) which increases autorotational performance so that the low speed avoid region of the H-V diagram is significantly

reduced or eliminated even without automatic control [8]. Some researchers have focused on specific phases of a descent in autorotation. For example, Yomchinda et al. have proposed a flight path planning algorithm for the steady-state descent phase [9]. Tierney and Langelaan have presented a method for calculating the set of flare entry points from which a helicopter can safely land [10]. Others have focused on ways to reduce pilot workload during autorotation using simple control laws that communicate to a pilot upon approaching a dangerous state during autorotation [11].

Though there has been considerable research in this area, previous approaches have limitations. For example, Dalamagkidis's approach is focused on axial vertical descent, a problem which has very limited practical motivation since forward flight is usually desirable to reduce the descent rate or maneuver to a landing site. The approach taken by Abbeel et al. requires training data from an experienced pilot, and it is likely that independent sets of training data would be needed for different airframes and flight regimes. Dalamagkidis, Bachelder et al., and Yomchinda et al. propose algorithms which involve iterative optimization of flight trajectories [4, 7, 9]. It is possible that these optimization algorithms may be slow to converge or may not even yield a usable solution under certain conditions. Iterative optimization also requires more computational capability than a classical feedback control law. Furthermore, these schemes may encounter problems if the computer model of the vehicle is inaccurate. For example, even a high fidelity model could be compromised if the actual takeoff weight is significantly different from the estimated weight programmed into the computer. In such a case, the algorithm will yield a control input solution that is optimal only for the model implemented in the computer, and this may not result in a successful landing when applied to the physical vehicle. The limitations described above are especially problematic during the critical flare phase of flight. In the words of Tierney and Langelaan "Computing a safe, feasible

flare trajectory in real time is extremely difficult because of the high dimensionality of the problem, the limited computational resources likely to be available, and the likelihood of external disturbances such as gusts." [10].

The research presented here considers the possibility that a suboptimal controller that mimics the actions of a human pilot performing an autorotation is the most practical approach. The proposed control law is a nonlinear mapping between measured states and control outputs that does not require any iterative calculation or prediction using a complex model. It is also easily scalable to any single main rotor helicopter, from a micro-air-vehicle to a full-size utility helicopter. The controller could be implemented as a closed-loop system on a fully-autonomous vehicle, or may provide guidance to a human pilot as described by Keller et al. and others [4, 11]. The approach described here does not yet address the problem of finding and navigating to a suitable landing site, however, the steady-state descent controller is designed deliberately to accommodate such maneuvers and could readily be combined with a path planning algorithm such as that proposed by Yomchinda et al. [9].

This thesis presents the new control law and simulation results which provide initial validation. In Chapter 2, the control architecture is described. A brief description of the high fidelity simulation model used to validate the control law is presented in Chapter 3. Finally, simulation results for both a full sized single engine Bell AH-1G Cobra helicopter and a small remote control Align TREX 600 helicopter are given in Chapter 4.

2. CONTROL LAW ARCHITECTURE

The term "expert system" has been used to describe a wide range of control systems and other structures. The use of the term here is similar to its use by Livchitz et al. applied their controller for unmanned landing of small fixed wing aircraft [12]. There are two primary characteristics of both the new autorotation controller and the one described by Livchitz that lead to the title "expert control system". First, the controllers are designed to imitate the thought process that an expert human pilot might take. Second, the control system already contains "expert" knowledge at flight time in the form of a deterministic mapping from vehicle and controller states to specific control outputs, that is, there is no iterative learning, high-fidelity model prediction, or optimization at runtime. The new autorotation controller differs from many other expert systems including Livchitz's in that it does not use a single method such as fuzzy logic to apply the expert knowledge contained in it. Instead, the new controller uses a combination of several structures including crisp logic, fuzzy logic, and linear proportional-integral-derivative (PID) control.

The autorotation maneuver is divided into five fuzzily-defined regions based on altitude and the projected time to impact assuming constant velocity. These regions represent the phases of the autorotation maneuver that the pilot would progress through and are described in Section 2.1. For each of these regions a low level control law is defined to guide the dynamic system to the desired state.

The control law does require the definition of several tuning parameters. The most notable requirement is the definition of the fuzzy boundaries of the flight phases. There are also several other parameters that are specific to each flight phase that must be defined. Throughout this text, the parameters that must be specified or tuned

are indicated by the use of a fixed-width teletype font with underscores, e.g. `U_AUTO`. These parameters are listed together with a brief description of each and values used in simulation in Table 2.1 on page 7. They are referenced and discussed more thoroughly in Sections 2.2 through 2.6. It should be noted that these parameters need not require precise values, and can be tuned based on intuition, flight experience, test data, and simulation to achieve acceptable results with reasonable effort.

The control inputs to the helicopter consist of the main rotor collective pitch, θ_0, the longitudinal cyclic pitch, θ_{1s}, the lateral cyclic pitch, $\theta_1 c$, and the tail rotor collective pitch, θ_{tr}. Horizontal velocity, sideward velocity, and yaw control of the helicopter are handled by a standard flight controller which will be referred to as the velocity tracking controller. The autorotation controller recommends a desired near-optimal forward speed to the helicopter velocity tracking controller which tracks these commands through longitudinal and lateral cyclic inputs. In addition, a maximum cap on the pitch and roll angles, η_{max} is imposed to prevent drastic maneuvers in certain conditions such as in close proximity to the ground. This velocity tracking controller can be based on any control approach from a neural network to simple PID. In most cases, this control system should be the same controller that is used in normal powered flight since that controller will have been extensively tuned to provide optimal tracking performance and will have been proven to be robust. An additional outer-loop control block may be added to handle path planning to a suitable landing site during the steady-state descent phase. For the simulations presented here, a simple multi-layered PD controller was implemented for velocity and yaw angle tracking through the θ_{1s}, θ_{1c}, and θ_{tr} control channels (see Section 2.7).

The autorotation controller directly handles the main rotor collective, θ_0, since it is the critical control input affecting the rotor rotational speed, Ω, which must be carefully managed during autorotation. Since the desirable set point of the main

rotor collective is highly dependent upon the helicopter mass and other parameters which may be unknown at flight time, each sub-controller actually recommends an *adjustment* to θ_0 and observes the results to seek a suitable trim value for θ_0 in much the same way that a human pilot would make adjustments. Thus, the outputs of the autorotation control law within each flight phase are a main rotor collective pitch derivative, $\dot{\theta_0}$, a desired forward velocity, u_{des}, and maximum pitch and roll angle, η_{max}, sent to the velocity tracking controller. This control scheme is illustrated in Figure 2.1

Table 2.1: Controller Parameters. This list is complete except for fuzzy definitions of the flight phases and the estimates of the helicopter's physical parameters required to estimate kinetic energy. Note the "roundness" of the values selected for the AH-1; none have more than two significant figures. This is because these parameters are approximate and do not require precise tuning for good performance.

Parameter	*Value for AH-1G*	*Value for TREX 600*
Definition		
`U_AUTO`	100 ft/s	10 ft/s
Desired forward speed for the steady state descent phase. This should be near the recommended autorotation speed published in the helicopter manual, which should be near the forward speed for minimum required power.		
`RPM_AUTO`	34 rad/s	180 rad/s
Desired main rotor rotation rate for the steady state descent phase. This is often near the normal operating RPM.		
`K_D_SS`	0.03 s^{-1}	0.003 s^{-1}
Gain on rotor speed time derivative for collective control during steady-state descent		
`K_P_SS`	0.01 (unitless)	0.001 (unitless)
Gain on rotor speed for collective control during steady-state descent		
`PRE_FLARE_MAX_ANGLE`	10°	10°
Maximum cap on roll and pitch angle during the pre-flare phase		

Table 2.1 (continued)

Parameter	*Value for AH-1G*	*Value for TREX 600*
Definition		
`U_TOUCHDOWN`	10 ft/s	1 ft/s
Desired forward velocity at touchdown		
`TTI_L`	1.5 s	1.0 s
Desired time to impact during the landing phase		
`TTI_F_MAX`	6 s	5 s
Maximum cap on the desired time to impact during the flare phase		
`K_COL`	6×10^{-4} rad·s^2/ft	3.1×10^{-4} rad·s^2/ft
Rotor collective gain for flare and landing phases		
`TAU`	0.05 s	0.05 s
Rotor collective adjustment time constant tuning parameter for flare and landing phases		
`FAST_COL_INCREASE`	15 °/s	10 °/s
Collective adjustment rate for rapid adjustments during the flare and landing phases		
`LANDING_MAX_ANGLE`	8°	10°
Maximum cap on roll and pitch angles during the landing phase		
`TOUCHDOWN_MAX_ANGLE`	1°	3°
Maximum cap on roll and pitch angles during the touchdown phase		
`TOUCHDOWN_COL_DECREASE`	−1 °/s	−1 °/s
Constant collective pitch rate during touchdown phase		

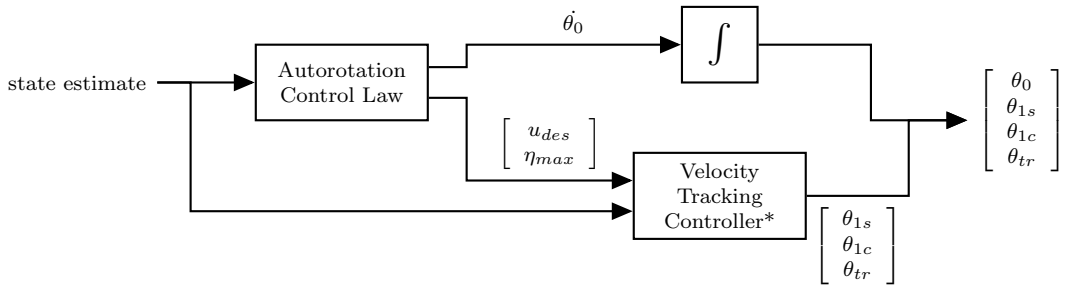


Figure 2.1: High Level Control Architecture. The autorotation control law determines θ_0 while the velocity tracking controller determines the cyclic and tail rotor commands with input from the autorotation control law. *The velocity tracking controller could be replaced with a landing site seeking controller.

2.1 Descent Phases

The autorotation descent is divided into five phases based on the altitude, h, and the predicted time to impact assuming constant vertical velocity, $TTI_{\ddot{h}=0} \equiv -h/\dot{h}$. This is similar to the approach for autorotation taken by Abbeel [6], and Livchitz et al. also use a fuzzy partition of flight phases in their landing controller for small fixed wing aircraft [12]. The names of the control phases are steady state descent, pre-flare, flare, landing, and touchdown. Figure 2.2 shows the definition of the phases using transition values for the Bell AH-1G Cobra that was used for simulation validation. The altitude and time to impact domains are divided into a strict fuzzy partition using trapezoidal membership functions. The values of the transition boundaries for the AH-1G are given in Table 2.2. Note that there is an "OR" relationship between the altitude and time to impact phase definitions; i.e. the controller will begin to advance to the Flare phase if it is below the Flare upper boundary altitude or if the predicted time to impact is less than the upper boundary time to impact. Also, the controller is implemented so that it progresses through the phases sequentially; i.e. once the controller is in the flare phase, it cannot return to the pre-flare phase, even if the altitude increases. This is observed in the simulation results in Figure 4.4 where

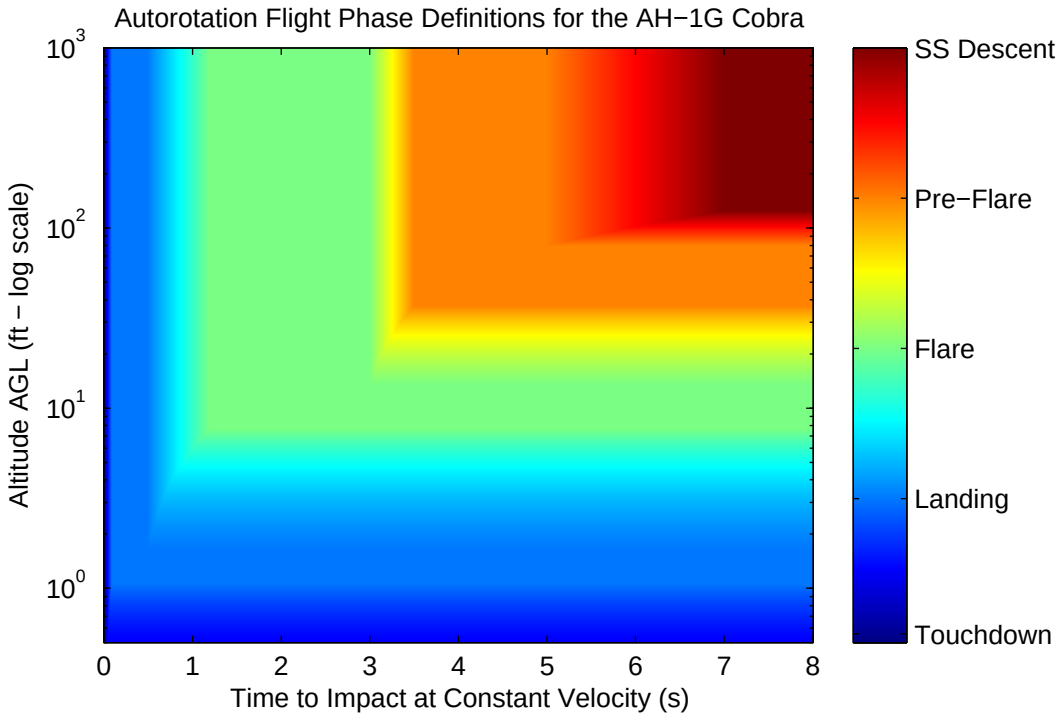


Figure 2.2: Descent Phase Diagram for the Bell AH-1G Controller. A logarithmic scale is used on the altitude axis because the Landing and Touchdown phases are defined at very low altitudes, and the gradual transitions are more apparent when a log scale is used. Note that since the helicopter slows drastically at lower altitudes, the phases are more evenly distributed over the time to impact domain, and thus a log scale is not used.

the distribution of authority between the control laws corresponding to different phases remains constant for a short time as the helicopter ceases to progress forward through the phases. This prohibition on backwards progression through the phases means that there is not necessarily a unique mapping from the physical state of the helicopter at a given time to a control output. Instead, the control output also depends on the internal controller state or equivalently the time history of the helicopter physical state. In the following subsections, the control laws for each phase are described.

Table 2.2: Flight Phase Fuzzy Transitions for the AH-1G Cobra Helicopter. Since trapezoidal membership functions are used, trasitions are linear.

Transition	Altitude Range (ft)	Time to Impact Range (s)
Steady State Descent to Pre-Flare	100 to 150	5 to 7
Pre-Flare to Flare	20 to 50	3 to 3.5
Flare to Landing	3 to 12	0.5 to 1.2
Landing to Touchdown	0 to 2	0 to 0.1

2.2 Steady State Descent

In the steady state descent phase, the controller seeks to maintain a constant rotor rotation rate near the normal operating rate while the helicopter maneuvers to a suitable landing site. Since a path planning algorithm that computes feasible paths to a landing site is outside the scope of this paper, the controller currently simply tries to match a forward speed `U_AUTO` specified by the control law designer. The following equations define the control output in this phase:

$$\begin{aligned} u_{des} &= \texttt{U\_AUTO} \\ \eta_{max} &= \text{limited only by velocity tracking controller} \\ \dot{\theta_0} &= \texttt{K\_D\_SS}\,\dot{\Omega} + \texttt{K\_P\_SS}(\Omega - \texttt{RPM\_AUTO}) \end{aligned} \tag{2.1}$$

The steady state descent forward speed, `U_AUTO`, should ideally be the forward speed at which the required power in steady-state forward flight is minimized [1]. Generally this is near the recommended forward speed for autorotation given in the flight manual for a manned helicopter. Operating near this speed will ensure a slow descent rate. `RPM_AUTO` can be set to the normal operating rotor rotation rate, or an increased value if more energy is desired for flare as long as the value is not above structural limits. The derivative of collective pitch, $\dot{\theta_0}$, is governed by a simple PD controller which drives collective toward a value corresponding to trimmed

autorotation. This is equivalent to governing θ_0 itself using a PI controller. The gains `K_D_SS` and `K_P_SS` can be chosen to yield reasonable response using simplified or high-fidelity dynamic flight simulation. In general, if gains are chosen appropriately, this control law will be stable in the normal operating region where the steady state rotor rotation rate will decrease if θ_0 increases, that is $\partial\Omega_{SS}/\partial\theta_0 > 0$. For some model size aerobatic helicopters, there is a region of large negative θ_0 where a decrease in collective will decrease the steady state rotation rate, that is $\partial\Omega_{SS}/\partial\theta_0 < 0$. In this region, the controller will fail. For this reason care must be taken to make sure that `K_D_SS` is large enough so that θ_0 does not overshoot the target value corresponding to `RPM_AUTO` by too large a margin, or an additional control constraint must be introduced to prevent this.

The steady state control law is designed to maintain an appropriate rotor rotational rate regardless of the forward speed of the helicopter or maneuvers used to reach a safe landing site. This makes it suitable for use with a pilot controlling the cyclic pitch or integration with an algorithm designed to find a suitable landing site. Simulation tests have shown that the controller is able to adjust θ_0 to suit a range of steady state forward speeds.

2.3 Pre-Flare

During the pre-flare phase, the controller attempts to bring the helicopter state into the subspace that will likely result in a successful flare. The pre-flare controller is identical to the steady state descent controller except that it instructs the velocity tracking controller to limit its maneuvers to a small roll or pitch angle (η) so that it is not attempting drastic maneuvers when entering the Flare phase. The following

equations define control output in this phase:

$$\begin{aligned} u_{des} &= \texttt{U\_AUTO} \\ \eta_{max} &= \texttt{PRE\_FLARE\_MAX\_ANGLE} \\ \dot{\theta_0} &= \texttt{K\_D\_SS}\,\dot{\Omega} + \texttt{K\_P\_SS}(\Omega - \texttt{RPM\_AUTO}) \end{aligned} \tag{2.2}$$

2.4 Flare

The Flare phase is the most critical part of the autorotation maneuver and proper timing is vital. The goal of the flare phase is to reduce the vertical and horizontal velocities to values suitable for safe entry into the landing phase. The velocity tracking controller is instructed to bring the helicopter to the small translational velocity value desired for landing, `U_TOUCHDOWN`, as fast as possible and is allowed to use any maneuver (η is not limited) to do so.

$$\begin{aligned} u_{des} &= \texttt{U\_TOUCHDOWN} \\ \eta_{max} &= \text{limited only by velocity tracking controller} \end{aligned} \tag{2.3}$$

The remaining task for the collective controller is to determine and track a vertical trajectory that will cause the helicopter to enter the landing phase at the same time that the speed tracking controller reaches the desired speed. Determining a feasible flare trajectory is noted as a challenge in the literature. For example, Abbeel et al. assert that "helicopter dynamics are very complex and this makes it difficult (if not impossible) to hand-specify a trajectory that (even approximately) obeys the dynamics" [6]. Their response to this challenge is to use data from actual autorotations performed by a human pilot to determine a feasible trajectory. While this

Table 2.3: Time to impact variables and parameters used in the flare control law. Values for the variables during a sample simulation are shown in Figure 4.5

Variable:	$TTI_{\ddot{h}=0}$ (horizontal axis in Fig. 2.2)
Physical Meaning:	Estimated **T**ime **T**o **I**mpact assuming speed remains constant
Source:	Calculated based on measured helicopter state
Use:	Determining which phase the helicopter is in
Variable:	`TTI_L`
Physical Meaning:	*Desired* **T**ime **T**o **I**mpact during the **L**anding phase
Source:	Tunable control law parameter
Use:	Landing phase control law and determining TTI_F
Variable:	$TTLE$
Physical Meaning:	*Desired* **T**ime **T**o **L**anding phase **E**ntry
Source:	Determined by algorithm in Blocks 2.1-2.4
Use:	Determines TTI_F
Variable:	TTI_F
Physical Meaning:	*Desired* **T**ime **T**o **I**mpact during the **F**lare phase
Source:	$TTI_F \equiv \texttt{TTI\_L} + TTLE$
Use:	Flare phase control law

strategy is remarkably successful, it requires the capture of training data and the associated data reduction and analysis for each specific vehicle under consideration.

Instead of attempting to directly specify a feasible trajectory in the space of the helicopter's physical state, we transform the problem to a new domain that lends itself to a heuristic reasoning approach to generating a trajectory. The new domain used for the predictive heuristic reasoning will be referred to as the time to impact domain. Four important variables on the time to impact domain are used by the controller. The meaning, source, and use of each are briefly described in Table 2.3.

With the introduction of the time to impact domain, the tasks of the flare phase controller are to a) determine a suitable value for $TTLE$ (defined in Table 2.3) using

simple heuristic rules (described in Sections 2.4.1 and 2.4.2) and b) apply control inputs to the helicopter that will put it on a trajectory to enter the landing phase approximately $TTLE$ seconds in the future (described in Section 2.4.3).

The logical development of the algorithm that estimates $TTLE$ follows a sequence that is rather different from the implementation of the algorithm, so two separate descriptions are given here. First, a less formal discussion of the reasoning that forms the basis for the algorithm is given in Section 2.4.1, then Blocks 2.1-2.4 show the actual implementation of the algorithm in Section 2.4.2.

2.4.1 Logical development of the algorithm that determines $TTLE$

This section presents the reasoning that serves as the foundation for the $TTLE$ algorithm. The description begins with the basic notion of how $TTLE$ is determined, and then describes the constraints required to make the value of $TTLE$ suitable for all conditions.

The time required to decelerate horizontally is estimated by dividing the difference between the current forward speed, $\dot{x}$, and the desired speed at landing phase entry, `U_TOUCHDOWN`, by the horizontal acceleration, $\ddot{x}$. Similarly, the time required to decelerate vertically is the difference between the current vertical velocity, $\dot{h}$, and the desired value at landing entry, $\dot{h}_{LE}$, divided by the vertical acceleration, $\ddot{h}$. The exact implementation of these two calculations is shown in Section 2.4.2 in Blocks 2.2 and 2.3. $\dot{h}_{LE}$ is automatically inferred from other parameters according to the relationship $\dot{h}_{LE} \equiv h_{LE}/\texttt{TTI\_L}$ where h_{LE} is defined as the altitude midway through the transition between flare and landing phases. Thus, the basis for the heuristic determination of $TTLE$ is the following equation:

$$\overline{TTLE} = \max\left(\frac{\dot{h}_{LE} - \dot{h}}{\ddot{h}}, \frac{\texttt{U\_TOUCHDOWN} - \dot{x}}{\ddot{x}}\right) \tag{2.4}$$

$\overline{TTLE}$ is an estimate of the amount of time it will take to slow to an acceptable speed in both the vertical and horizontal directions. The actual value of $TTLE$ is related to $\overline{TTLE}$, but there several additional rules and constraints in place to handle cases in which $\overline{TTLE}$ is not an accurate estimate of the time required to decelerate.

First, if either $\ddot{h}$ or $\ddot{x}$ has the wrong sign, it means that the physical state of the helicopter is moving away from the desired state. If Equation 2.4 were used directly in this case, $TTLE$ would be negative and thus would not accurately reflect the amount of time required to decelerate. In this case, $TTLE$ is set to a maximum value, $TTLE_{max}$. This value is based on user-selectable parameters (see Equation 2.6 for definition). Also, if the accelerations have the correct sign but are small enough to make $TTLE$ very large, $TTLE$ is capped at $TTLE_{max}$. These rules are shown implemented in Blocks 2.2 and 2.3 on page 18. They enforce the following constraints:

$$\begin{aligned} TTLE &\geq 0 \\ TTLE &\leq TTLE_{max} \end{aligned} \tag{2.5}$$

Second, According to Equation 2.4, if the actual helicopter velocity is near the desired velocity, but the acceleration is small, the value of $TTLE$ will still be large. This is not the desired behavior. When the helicopter has reached a velocity near the desired velocity, it should enter the landing phase regardless of the acceleration. In order to produce this behavior, a fuzzy set of small velocities is defined. When both of the components of velocity lie within this set, $TTLE$ is set to zero (see Blocks 2.2 and 2.3 on page 18 for clarification on how this is implemented). The set is defined by the membership μ_{small}, which is a trapezoidal membership with a support of $(-6 \text{ ft/s}, 6 \text{ ft/s})$ and shoulders at ± 2 ft/s.

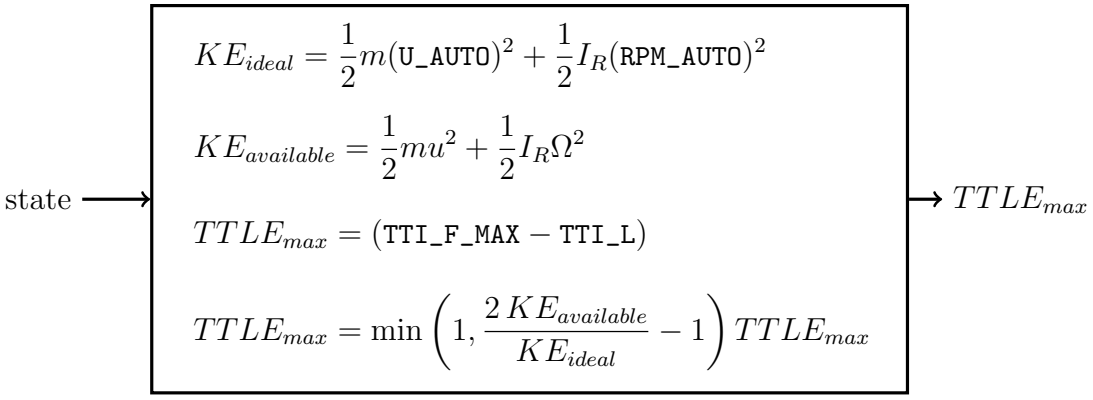
Finally, the amount of energy available to the helicopter must be taken into account. When the helicopter is autorotating from an initial state within the "avoid" region of the H-V curve, the rotor speed and forward velocity may be too low to allow a normal flare to take place. Instead, the helicopter will be forced to rapidly increase collective very late in the descent and land with whatever horizontal velocity it has. In other words, landing with a small vertical velocity is the highest priority; landing with a low horizontal velocity is a secondary consideration. The kinetic energy available to control the descent, $KE_{available}$, is estimated by summing the kinetic energy due to forward velocity and rotor rotation. This is compared to the kinetic energy that would be available for control in a nearly ideal autorotation that matches the desired controller parameters, KE_{ideal}. $TTLE_{max}$ is adjusted so that if $KE_{available} \leq 1/2\, KE_{ideal}$, $TTLE$ is zero.

$$TTLE_{max} \equiv \min\left(1, \frac{2\, KE_{available}}{KE_{ideal}} - 1\right)(\texttt{TTI\_F\_MAX} - \texttt{TTI\_L}) \tag{2.6}$$

A more detailed implementation description of this rule is given in Block 2.1 on page 18.

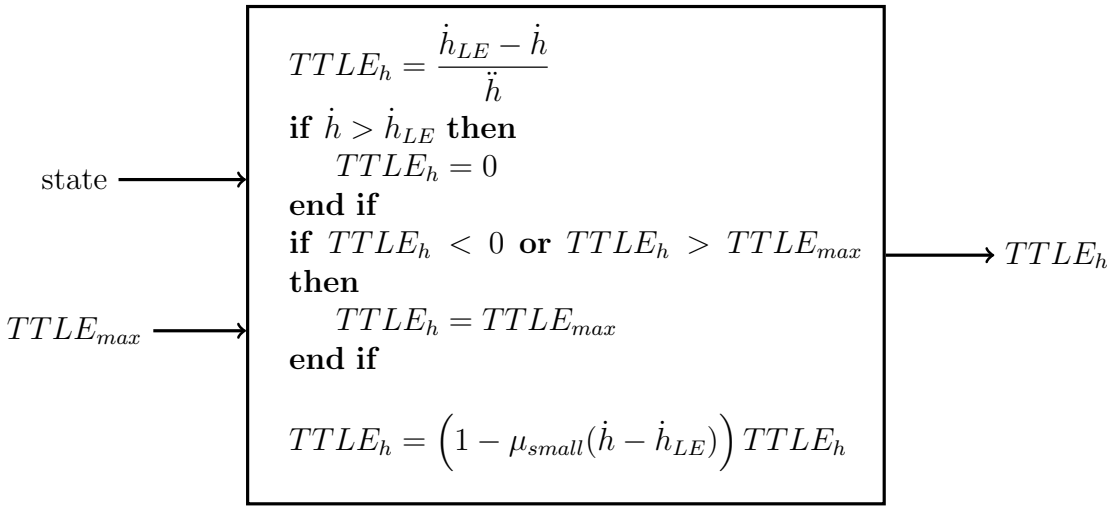
2.4.2 Implementation of the $TTLE$ estimator

The following blocks show the implementation of the $TTLE$ estimator based on the rationale in the previous section. Each block shows a small portion of the algorithm. The statements within the block are executed in order to produce the output. The complete algorithm is formed when the inputs and outputs are linked together. First, Block 2.1 shows the calculation of the energy based limit on $TTLE$, then Blocks 2.2 and 2.3 show calculation of the vertical and horizontal components

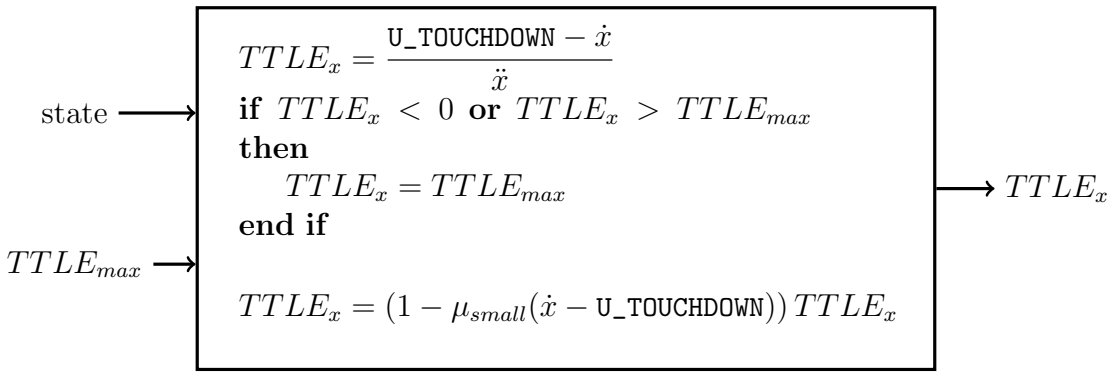


Block 2.1: Energy adjustment of the maximum limit on $TTLE$

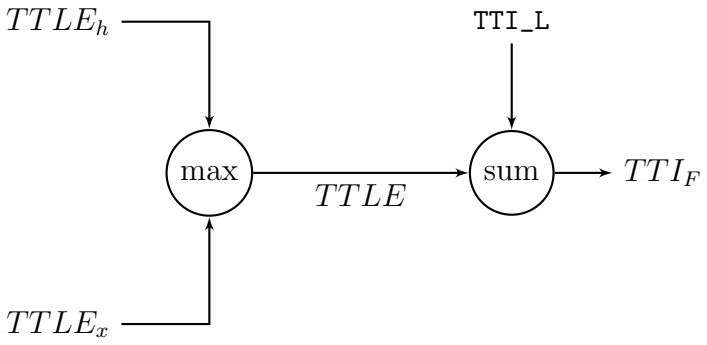
of $TTLE$, and finally Block 2.4 shows the combination of these values to produce $TTLE$ and TTI_F.



Block 2.2: The vertical speed contribution to $TTLE$



Block 2.3: The horizontal speed contribution to $TTLE$



Block 2.4: Calculation of $TTLE$ and TTI_F

2.4.3 Flare trajectory tracking

Now that the desired time to impact required for deceleration, TTI_F, has been calculated, a vertical trajectory can be generated and tracked. If the Helicopter is modeled as a point mass and maintains a constant vertical acceleration $\ddot{h}(t)$ starting at time t, then the altitude at time $t + TTI_F$ will be

$$h(t + TTI_F) = h(t) + \dot{h}(t)TTI_F + \frac{1}{2}\ddot{h}(t)TTI_F^2 \tag{2.7}$$

Under some conditions, solving this expression with $h(t + TTI_F) = 0$ will yield a vertical trajectory for impact at time $t + TTI_F$. Given the altitude h and altitude rate $\dot{h}$, if the condition

$$TTI_F \leq \frac{-2h}{\dot{h}} \tag{2.8}$$

holds, then the constant acceleration

$$\ddot{h}_{des} = -\frac{2}{TTI_F^2}h - \frac{2}{TTI_F}\dot{h} \tag{2.9}$$

will bring the helicopter to the ground at time $t + TTI_F$. If the condition in (2.8) does not hold, then there is no constant acceleration that will cause an impact at time $t + TTI_F$.

The value of the collective pitch corresponding to $\ddot{h}_{des}$ is unknown and highly dependent upon difficult-to-observe components of the state of the helicopter such as inflow. However, in order to define a simple control law, consider the crude linear approximation

$$\ddot{h} = \frac{\theta_0}{K_{\theta_0}} \tag{2.10}$$

The controller should drive θ_0 towards the (unknown) value required to produce $\ddot{h}_{des}$, so the control law

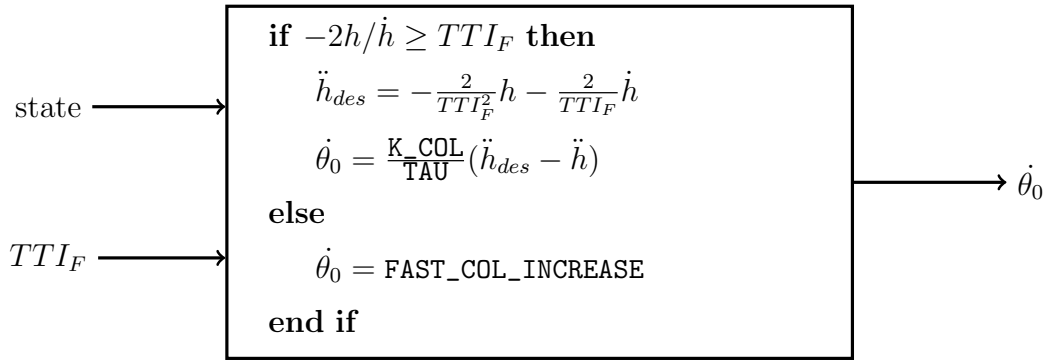
$$\dot{\theta_0} = \frac{\texttt{K\_COL}}{\texttt{TAU}}(\ddot{h}_{des} - \ddot{h}) \tag{2.11}$$

is adopted, where `K_COL` is chosen to be near the value of K_{θ_0}. This control law is analogous to the action that a human pilot would take when controlling a system where the trim control value is unknown. The control is adjusted in the direction of the desired result until the desired result is achieved. `K_COL` and `TAU` are redundant control parameters, but both are very useful for understanding the system and tuning the controller. Since K_{θ_0} can be crudely approximated using (2.10) and blade element theory, `K_COL` can be fixed at `K_COL` $\approx K_{\theta_0}$, and `TAU`, which represents the time constant of the system described in Equations 2.10 and 2.11, can be used to adjust the speed of the control response.[1]

If the condition in Equation 2.8 is not met, Equation 2.9 calculates the acceleration required to cross the ground plane on the *upward* stroke of the parabola defined in Equation 2.7. If this trajectory were followed, the helicopter would impact the ground at a time significantly before $t + TTI_F$. In order to prevent this, if the condition in Equation 2.8 is violated, the Flare controller commands a large upward adjustment (`FAST_COL_INCREASE`) of the collective pitch. This is analogous to a human pilot realizing that the magnitude of the vertical velocity is too large and, in response, rapidly increasing the collective pitch until the velocity has reached a manageable value.

[1]The reader may notice that Equations 2.9, 2.10, and 2.11 form a third order linear system describing the evolution of h (ignoring the condition in Equation 2.8). For positive values of `TAU`, the eigenvalues of this system are in the left half plane and two are complex. Thus, the system would naturally oscillate about $h = 0$, driving the helicopter *through* the ground. This hints at the importance of the condition in Equation 2.8, but since the condition is in place and TTI_F is not constant, the linear analysis offers little further insight.

Block 2.5 shows the implementation of the portion of the control law described in this section. Combining this block with Blocks 2.1-2.4 and Equation 2.3 creates a complete description of the flare control law.



Block 2.5: Flare collective control law

2.4.4 Concluding remarks regarding the flare controller

The preceding description of the flare control law is lengthy, and, at first reading, may seem complicated. However, the basic approach is simple. The flare controller must first estimate how long it will take for the velocity tracking controller to complete the flare while also taking energy constraints into accounts. Then the flare controller must determine a collective command sequence to bring the helicopter to the landing phase in approximately that amount of time. The specific low-level methods outlined here are not the only potential solutions; using a different but approximately functionally equivalent method for calculating $TTLE$ or $\dot{\theta}_0$ would likely result in another acceptable control law. The high-level concept of using approximate reasoning in the time to impact domain just as a human pilot might do is the most important feature of the control law and its fundamental innovation.

2.5 Landing

In the Landing phase, the controller seeks to bring the helicopter to the ground gently with nearly level attitude. The control law is similar to the Flare phase control law (it is derived from Equations 2.9 and 2.11), except that the desired time to impact remains constant, equal to the controller parameter `TTI_L`. For the full sized helicopter used for simulation, a suitable value for `TTI_L` proved to be 1.5 s.

$$\begin{aligned} u_{des} &= \texttt{U\_TOUCHDOWN} \\ \eta_{max} &= \texttt{LANDING\_MAX\_ANGLE} \\ \dot{\theta}_0 &= \begin{cases} \frac{\texttt{K\_COL}}{\texttt{TAU}} \left(-\frac{2(h+\dot{h}\texttt{TTI\_L})}{\texttt{TTI\_L}^2} - \ddot{h} \right) & \textbf{if } -\frac{2h}{\dot{h}} \geq \texttt{TTI\_L} \\ \texttt{FAST\_COL\_INCREASE} & \textbf{else} \end{cases} \end{aligned} \tag{2.12}$$

2.6 Touchdown

The touchdown phase brings the helicopter to rest on the ground by decreasing the collective slowly and attempting to maintain a level orientation. For large helicopters, limits on the control inputs may need to be implemented in this phase to keep the blades from impacting the empennage after touchdown due to the very low rotational rate of the rotor. Note that these limits have not been included here but could be easily implemented and would be highly dependent on the vehicle under consideration.The following equations describe control parameters during this phase:

$$\begin{aligned} u_{des} &= \texttt{U\_TOUCHDOWN} \\ \eta_{max} &= \texttt{TOUCHDOWN\_MAX\_ANGLE} \\ \dot{\theta}_0 &= \texttt{TOUCHCOWN\_COL\_DECREASE} \end{aligned} \tag{2.13}$$

2.7 A Velocity Tracking Controller Reference Implementation

A simple velocity tracking controller was implemented as a reference example and for the sake of conducting simulations. In a commercial or military implementation, the velocity tracking controller would likely be the controller designed for powered flight of the particular helicopter or may be replaced by a more complex controller designed to automatically find a suitable landing site. This reference controller uses a simple two-tiered proportional derivative (PD) architecture shown below in Figure 2.3. The outer loop recommends an orientation ($[\phi_{cmd}, \theta_{cmd}, \psi_{cmd}]^T$) based on u_{des} and the current helicopter velocity. The inner loop attempts to match this orientation using the cyclic and tail rotor controls ($[\theta_{1s}, \theta_{1c}, \theta_{tr}]^T$).

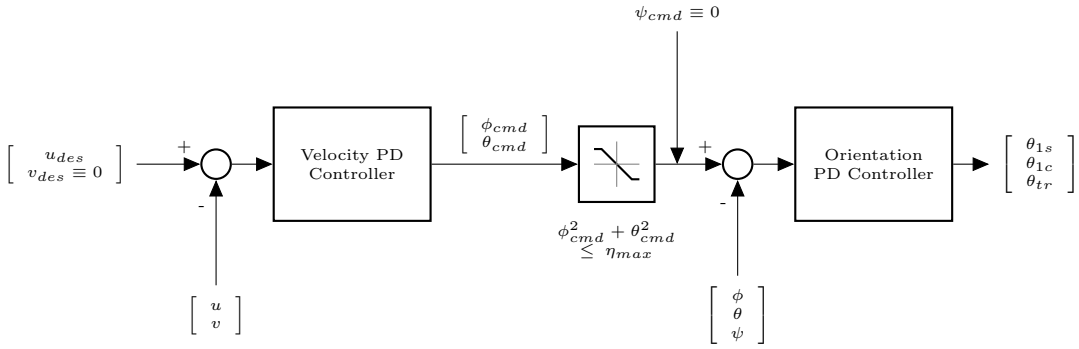


Figure 2.3: Reference Velocity Tracking Controller

3. SIMULATION MODEL

A high-fidelity helicopter simulation model was created in order to validate the proposed control law. This simulation software was also used for flight dynamics research in Reference [13], and thus received limited flight test validation through that project. Fuselage, horizontal stabilizer, vertical stabilizer, and tail rotor forces and moments are computed based on the ARMCOP model developed by Talbot and Chen [14, 15, 16]. The main rotor model, however, provides higher fidelity than that used in ARMCOP, incorporating dynamic inflow, ground effect, and blade stall. Each of the following subsections describe a component of the helicopter model. Detailed descriptions of each model component are beyond the scope of this paper but can be found in the cited references.

3.1 Tail Rotor, Fuselage, Stabilizers

The fuselage, horizontal and vertical stabilizer, and tail rotor components of the helicopter model are implemented as described in Reference [14]. The tail rotor uses Newton-Raphson iteration to calculate uniform tail rotor inflow. Its blade flapping model is quasi-steady state. That is, the blade flapping dynamic equations (Equation 3.2 on page 27) are solved for steady state ($\ddot{\beta} = \dot{\beta} = 0$) at the current conditions because the flapping dynamics for the tail rotor are fast enough to be neglected for control and handling qualities research [14]. Other components have rudimentary aerodynamic models which introduce forces and moments affecting the motion of the helicopter.

3.2 Forces and Moments Generated by the Main Rotor

The forces and moments generated by the main rotor are calculated using a numerical blade element approach. The main rotor blade is divided into 15 blade elements and 2D aerodynamic analysis is performed for each. The velocity of the air due to the motion of the helicopter and the induced inflow is calculated at each blade element. Based on this velocity, the forces on the blade element are calculated using a lift and drag coefficient look up table for the specific airfoil under consideration [17]. This calculation for a representative blade is carried out at 30 rotational stations evenly distributed over a complete revolution. The results are summed and appropriately normalized according to the number of blades and rotation stations. This numerical calculation is used to obtain the aerodynamic forces exerted by the entire rotor on the vehicle. Blade loads determined by these calculations, resolved as the main rotor torque, are used to determine the rotor rotation rate derivative, $\dot{\Omega}$, when the engine is not powering the vehicle. In addition to the forces and moments exerted on the helicopter, these calculations determine the vector of aerodynamic force and moment coefficients, $\mathbf{C}$, needed in the dynamic inflow model.

Since the airfoil look up table contains wind tunnel data for angles of attack through 180°, simple stall behavior is also implicitly included in the model. That is, if the blade has a very large angle of attack, as is the case for the retreating blade in forward flight, the lift and drag coefficients extracted from the table will reflect stall. There is, however, no Reynolds number correction, and small-scale unsteady aerodynamic effects are not modeled, so the stall behavior is not completely accurate. Compressibility effects are ignored.

3.3 Blade Flapping

First harmonic flapping is assumed and higher-harmonic flapping dynamics are neglected for these control law studies. First harmonic blade flapping states β_0, β_{1s}, β_{1c}, and their time derivatives are integrated into the model as states. The flapping angle at any point in the rotation is given by

$$\beta(\psi_{MR}) = \beta_0 - \beta_{1c}\cos(\psi_{MR}) - \beta_{1s}\sin(\psi_{MR}) \tag{3.1}$$

where a positive β represents downward flapping.

The differential equation that governs flapping is given by

$$\ddot{\beta} + \omega_N^2\beta = M_F \tag{3.2}$$

where

$$\omega_N = \Omega\sqrt{\frac{I_B + \frac{meR}{2}}{I_B}} \tag{3.3}$$

In Equation 3.3, m is the blade mass, R is the blade radius, e is the flap hinge offset, and I_B represents the blade flap-wise inertia. M_F in Equation 3.2 includes all aerodynamic loads calculated through blade element theory and inertial moments as described in References [15] and [16], which are too lengthy to be provided here. The flapping differential equation is solved using a harmonic balancing approach in which a first-harmonic solution is assumed and harmonic coefficients β_0, β_{1s}, and β_{1c}

are extracted through the following projection operation:

$$\int_0^{2\pi} \left(\ddot{\beta} + \omega_N^2 \beta - M_F \right) \, d\psi_{MR} = 0 \tag{3.4}$$

$$\int_0^{2\pi} \left(\ddot{\beta} + \omega_N^2 \beta - M_F \right) \cos\psi_{MR} \, d\psi_{MR} = 0 \tag{3.5}$$

$$\int_0^{2\pi} \left(\ddot{\beta} + \omega_N^2 \beta - M_F \right) \sin\psi_{MR} \, d\psi_{MR} = 0 \tag{3.6}$$

Solution of Equations 3.4-3.6 yields second order differential equations for each of the three flapping states β_0, β_{1s}, and β_{1c}. The forces that are used in the flapping equations are calculated using simplified analytical expressions for the flapping moments on the blades that are independent from the blade element force calculations described in Section 3.2

3.4 Dynamic Inflow

The dynamic inflow model used in this work is described by Peters and HaQuang [18]. It is based on the Pitt and Peters model [19], which has been validated using experimental data and is widely used [20]. The model has three states, λ_0, λ_s, and λ_c which describe the induced inflow ratio distribution over the rotor disk according to the equation

$$\lambda_i(r, \psi_{MR}) = \lambda_0 + \lambda_s \frac{r}{R} \sin\psi_{MR} + \lambda_c \frac{r}{R} \cos\psi_{MR} \tag{3.7}$$

These states evolve according to the dynamic equation

$$[M] \begin{bmatrix} \dot{\lambda}_0 \\ \dot{\lambda}_s \\ \dot{\lambda}_c \end{bmatrix} + [\hat{L}]^{-1} \begin{bmatrix} \lambda_0 \\ \lambda_s \\ \lambda_c \end{bmatrix} = \mathbf{C} \tag{3.8}$$

where $\mathbf{C}$ is a vector of force and moment coefficients calculated using the blade element approach described in Section 3.2 above, $[\hat{L}]$ is a matrix dependent on the sideslip angle and wake angle, and $[M]$ is a mass term based on the mass of air near the rotor. Additional details regarding this model can be found in Reference [18].

3.5 Ground Effect

A simple ground effect correction is applied to the dynamic inflow model when the rotor is near the ground. Equation 3.8 shows that when the inflow has reached a steady state (i.e. $\dot{\lambda} = \mathbf{0}$),

$$\mathbf{C} = [\hat{L}]^{-1}\lambda_{\mathbf{ss}} \tag{3.9}$$

where $\lambda_{\mathbf{ss}}$ is the vector of the inflow states at steady state. We assume that in ground effect the steady state inflow can be modeled by

$$\lambda_{\mathbf{ssIGE}} = \left(1 - \frac{\Delta w}{w_0}\right)\lambda_{\mathbf{ss}} \tag{3.10}$$

where $\Delta w/w_0$ is a correction term for ground effect in forward flight described by Heyson [21]. This is applied in the dynamic inflow model by adjusting $\mathbf{C}$ so that λ tends towards $\lambda_{\mathbf{ssIGE}}$. At steady state,

$$\mathbf{C_{IGE}} = [\hat{L}]^{-1}\lambda_{\mathbf{ssIGE}} \tag{3.11}$$

$$= [\hat{L}]^{-1}\left(1 - \frac{\Delta w}{w_0}\right)\lambda_{\mathbf{ssIGE}} \tag{3.12}$$

$$= [\hat{L}]^{-1}[\hat{L}]\left(1 - \frac{\Delta w}{w_0}\right)\mathbf{C} \tag{3.13}$$

$$\mathbf{C_{IGE}} = \left(1 - \frac{\Delta w}{w_0}\right)\mathbf{C} \tag{3.14}$$

In the dynamic equation (3.8), $\mathbf{C}$ is adjusted using Equation 3.14 when the main rotor is within two rotor diameters of the ground:

$$[M]\begin{bmatrix}\dot{\lambda}_0\\ \dot{\lambda}_s\\ \dot{\lambda}_c\end{bmatrix} + [\hat{L}]^{-1}\begin{bmatrix}\lambda_0\\ \lambda_s\\ \lambda_c\end{bmatrix} = \left(1 - \frac{\Delta w}{w_0}\right)\mathbf{C} \tag{3.15}$$

The values for $\Delta w/w_0$ are taken from a lookup table based on Figure 3.1, taken from Reference [21]. This data is based on theory that has been experimentally verified. The data is indexed based on the height above ground and the wake angle determined from the inflow state and velocity of the helicopter.

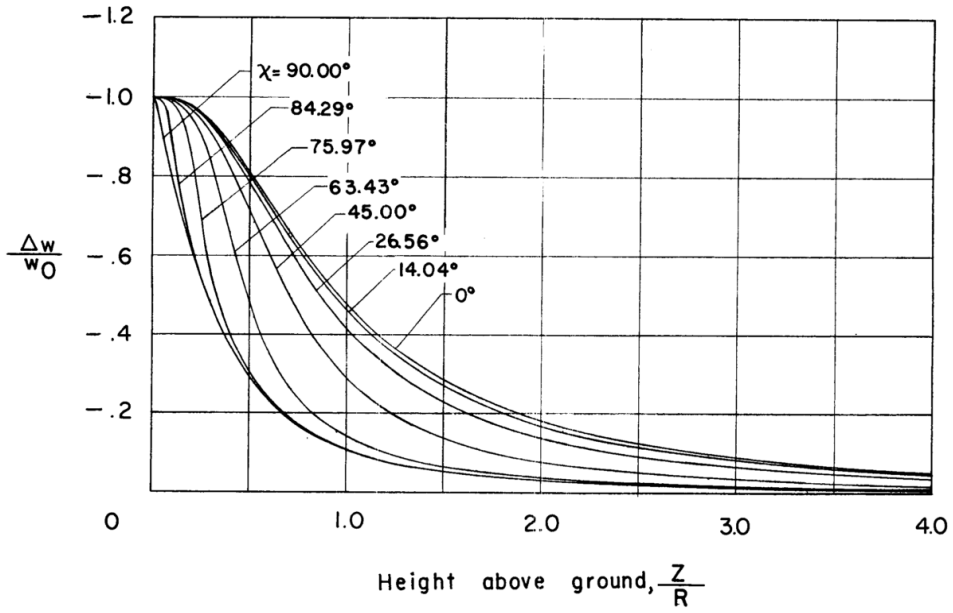


Figure 3.1: Ground Effect Correction (Taken from Reference [21])

3.6 Actuators

The simulated control actuators are limited to a maximum rate and have maximum and minimum stops. Therefore, the actual control value differs from the com-

manded control value depending on how fast changes are applied. The actuators respond as quickly as possible without exceeding the maximum rate. The behavior for a control input updated at 20 Hz and an actuator limited to 40 °/s response is illustrated in Figure 3.2.

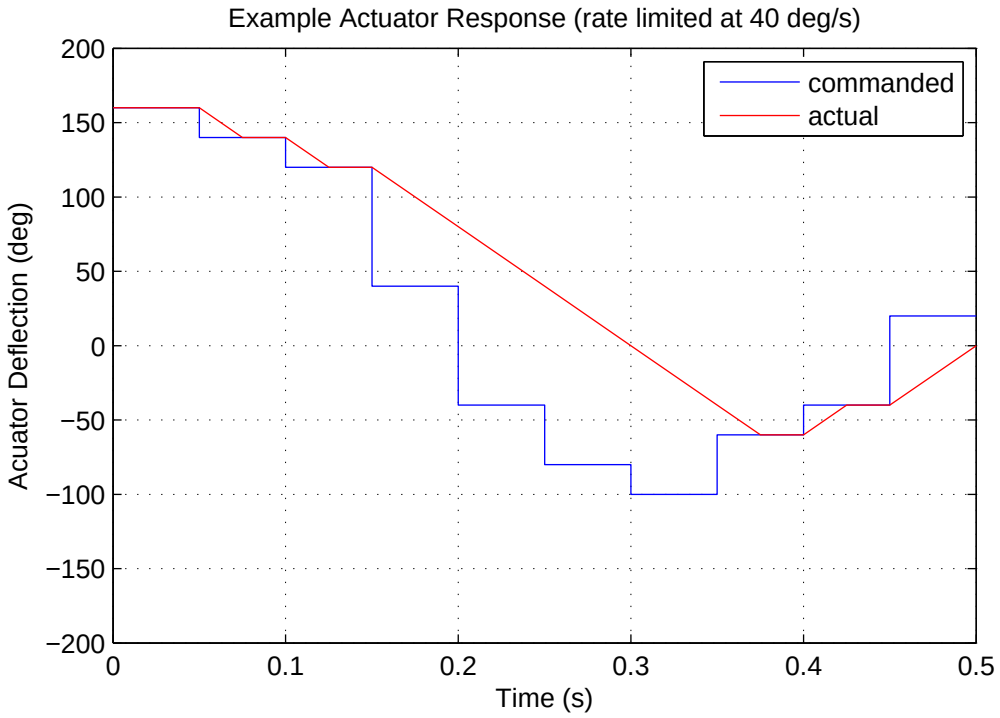


Figure 3.2: Sample Actuator Response

The simulation software models each control (θ_0, θ_{1s}, θ_{1c}, and θ_{tr}) as if it had its own dedicated actuator, so the complex rate and limit interactions between the actuators connected to the swash plate are not modeled. For complete accuracy, these would need to be modeled on a vehicle-specific basis, but, in general, actuator lag is included in the model through this rate-limiting scheme.

3.7 Simulation Overview

The summed forces (F_x, F_y, and F_z) and moments (M_x, M_y, and M_z) generated by the model components described in Sections 3.1 and 3.2 are used to drive the rigid body equations of motion which govern the position of the mass center (x, y, and z) and the orientation of the body with respect to an inertial frame. Orientation is represented by a 3-2-1 Euler angle rotation sequence through ψ, θ, and ϕ. The velocities and angular rates in the body frame are u, v, w, p, q, and r. The rigid body equations of motion below use the shorthand notation $c_{(\cdot)}$ for $\cos(\cdot)$ and $s_{(\cdot)}$ for $\sin(\cdot)$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{3.16}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{c_\theta} \begin{bmatrix} c_\theta & s_\phi s_\theta & c_\phi s_\theta \\ 0 & c_\phi c_\theta & -s_\phi c_\theta \\ 0 & s_\phi & c_\phi \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.17}$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} - \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{3.18}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = [I_c]_B^{-1} \left\{ \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} - \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} [I_c]_B \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right\} \tag{3.19}$$

The complete system state including the states of all components contains the following 22 elements:

$$\text{state} = \left[x, y, z, \phi, \theta, \psi, u, v, w, p, q, r, \Omega, \beta_0, \beta_{1s}, \beta_{1c}, \dot{\beta_0}, \dot{\beta_{1s}}, \dot{\beta_{1c}}, \lambda_0, \lambda_s, \lambda_c\right] \tag{3.20}$$

This state is integrated using the fourth order Runga-Kutta method with a timestep of 0.001 seconds.

4. SIMULATION RESULTS

A large number of Monte-Carlo simulations were run to provide preliminary validation of the controller. Similar tests were conducted for both the AH-1G full size attack helicopter (Section 4.1) and the small Align TREX 600 remote control model helicopter (Section 4.2).

4.1 Bell AH-1G Cobra Simulations

The model used in these tests is based on the Bell AH-1G Cobra attack helicopter. Most of the model parameters were obtained from Reference [14]. Table 4.1 lists some of the important model parameters. Recall that Table 2.1 lists the controller parameters used for these tests.

Figures 4.1 and 4.2 show the time histories of the states of a helicopter performing an autorotation descent from an altitude of 350 ft and a forward speed of 50 knots. This initial state is near the edge of the avoid region of the H-V diagram, but the controller handles the maneuver well, bringing the vehicle to a safe landing. A 1 second delay between engine shutoff and the point at which the autorotation controller takes over from the normal flight controller is simulated, potentially representing the actual time it would take to confirm power loss and initiate the autorotation controller.

There are a variety of notable features in these plots. First, note the immediate drop in rotor rotation rate Ω before the autorotation controller takes effect followed by the return of Ω to a value slightly higher than the normal operating value during steady state descent. Next, note that the vehicle achieves the desired forward speed for minimum descent rate, given as 100 ft/s for this helicopter. Also note the decrease

Table 4.1: Bell AH-1G Model Parameters Pertaining to Autorotation

Parameter	Symbol	Value
Helicopter gross weight	W	8300 lb
Number of main rotor blades	N_b	2 (Teetering)
Main rotor blade chord	c	2.25 ft
Main rotor radius	R	22 ft
Main rotor blade moment of inertia	I_B	2770 slug ft^2
Main rotor height above ground (water line)	WL_{MR}	12.73 ft
Main rotor normal operating speed	Ω_{normal}	32.88 rad/s
Main rotor blade airfoil used for simulation		NACA 0012
Actuator max rate	$\dot{\delta}_{max}$	40 deg/s
Controller update rate		20 Hz

in the induced velocity (λ_0) as the helicopter approaches the ground due to ground effect. Finally, note that at landing all velocities and orientation angles are small indicating a safe touchdown.

Figures 4.3-4.5 show some important controller internal states and outputs. Figure 4.3 shows the control outputs for the sample autorotation. Note the sharp peaks in θ_0 near the end of the dataset. These peaks indicate violations of the $TTI_F \leq -2h/\dot{h}$ condition in the control law. When this occurs, the controller rapidly increases θ_0 as described in Section 2.4. Though these peaks appear dramatic, the amplitude is less than 2° for the largest, and the frequency is not more than 2 Hz.

Monte Carlo simulations were conducted in and around the "avoid" region of the H-V diagram to demonstrate that the controller is able to recover from difficult initial conditions and significantly increase the envelope of safe flight. One important factor in determining the likelihood of a successful autorotation is the time between engine, transmission, or tail rotor failure and the beginning of autorotation-friendly

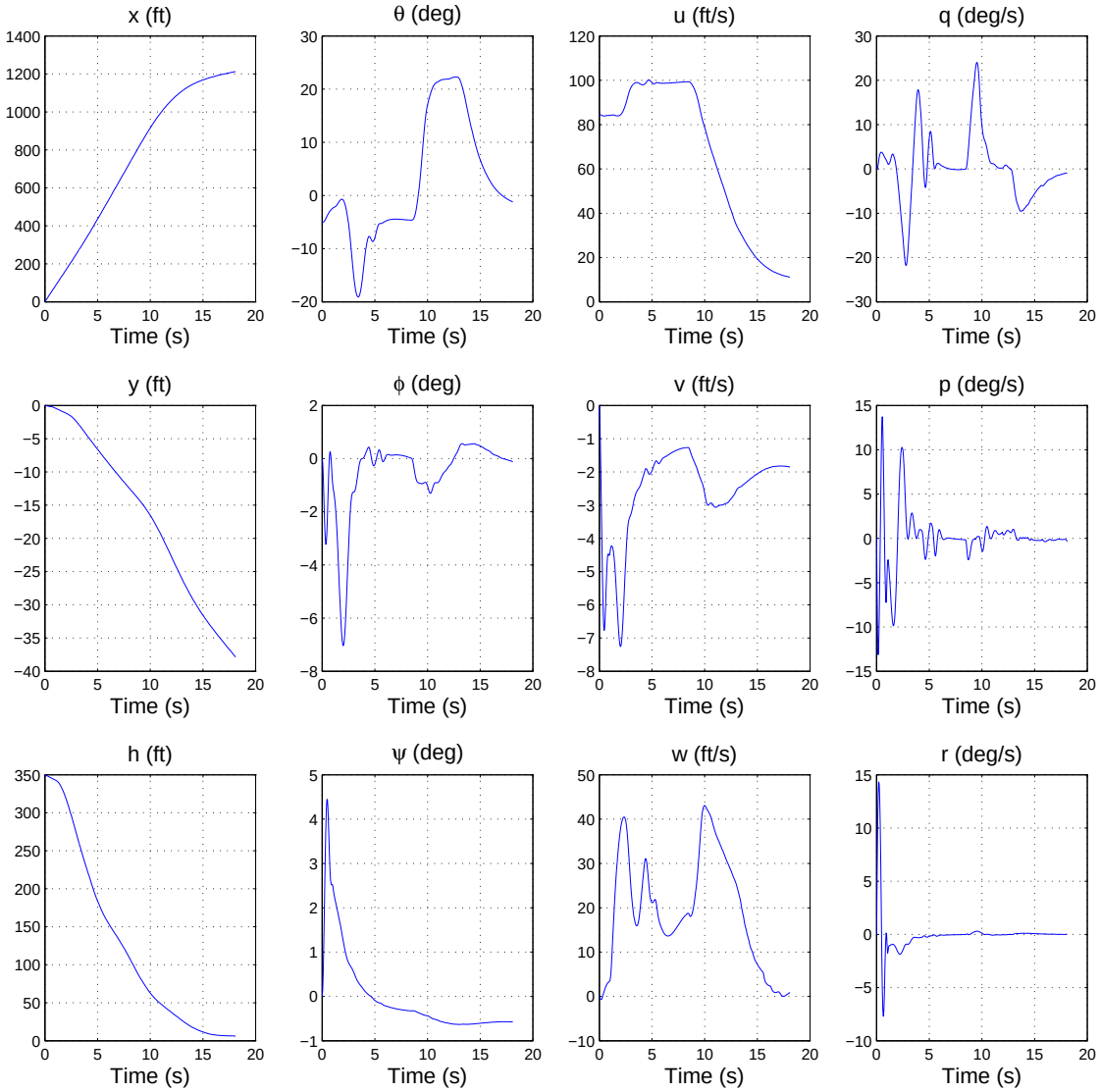


Figure 4.1: Helicopter Position, Orientation, Velocity, and Angular Rate State Histories for a Sample AH-1G Cobra Autorotation Simulation (see Figure 4.2 for main rotor states)

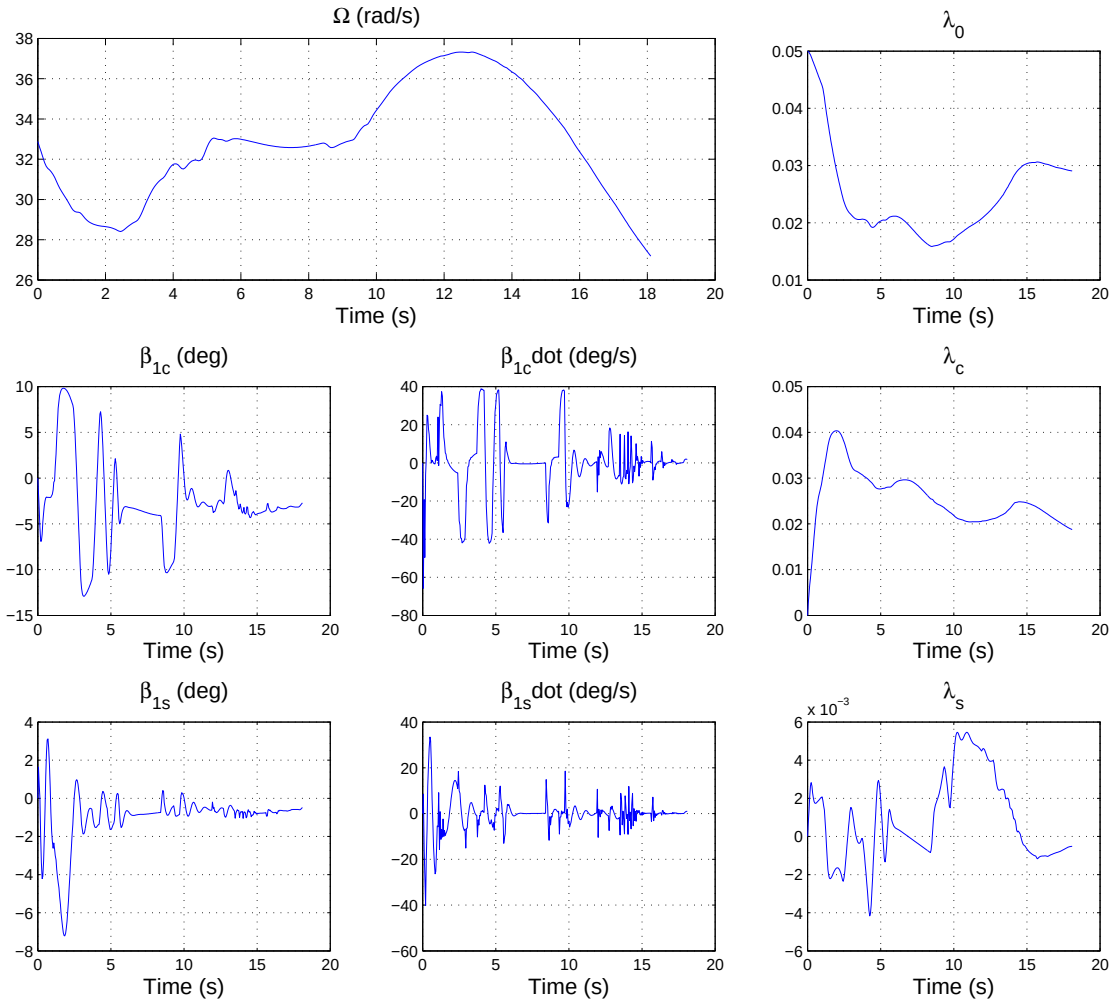


Figure 4.2: Helicopter Main Rotor State Histories for a Sample AH-1G Cobra Autorotation Simulation

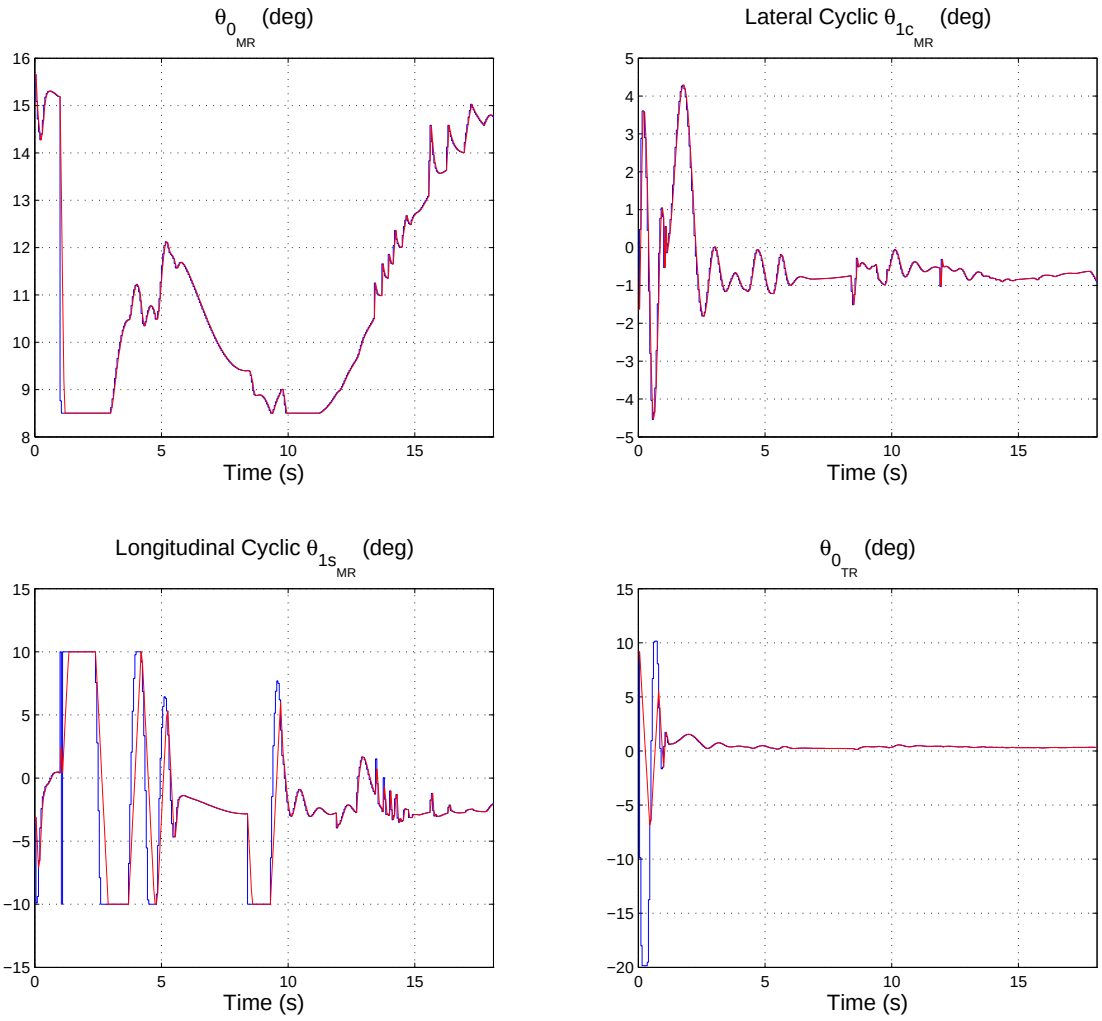


Figure 4.3: Control Histories for a Sample AH-1G Autorotation Simulation. Note that there are two lines in each of the plots. The blue line that leads is the commanded control position; the red line that lags slightly at some points is the actual actuator position.

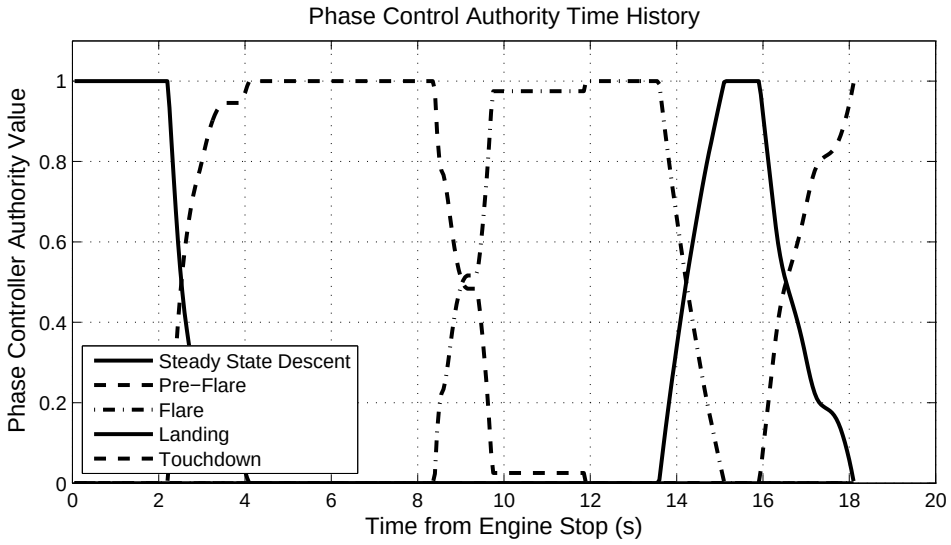


Figure 4.4: Phase Control Authority Time History. This plot shows which phase controllers have authority during different portions of the landing. A value of 1 denotes complete authority

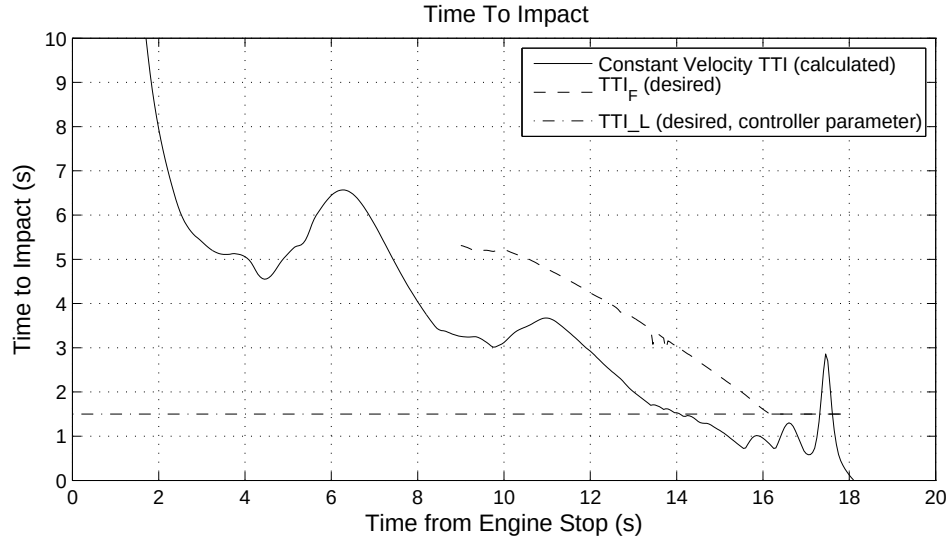


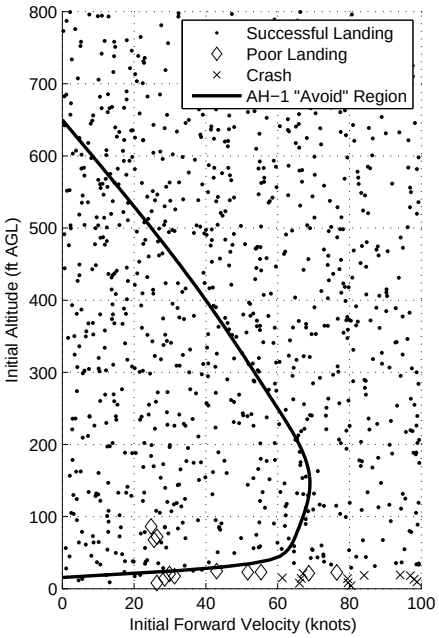
Figure 4.5: Controller Internal Time to Impact Variables for a Sample AH-1G Autorotation. This plot shows the values of several internal controller states. Note that $TTLE$ can be read off the plot as the difference between TTI_F and `TTI_L`

maneuvers by the pilot or control system. In an emergency, even an autonomous system might require some time to detect the failure and hand off control to the autorotation control law. Human pilots are typically expected to react to an emergency in 1-2 seconds depending on pilot workload [1], so simulations are conducted assuming immediate handoff, a handoff delayed by 1 second, and a handoff delayed by 2 seconds.

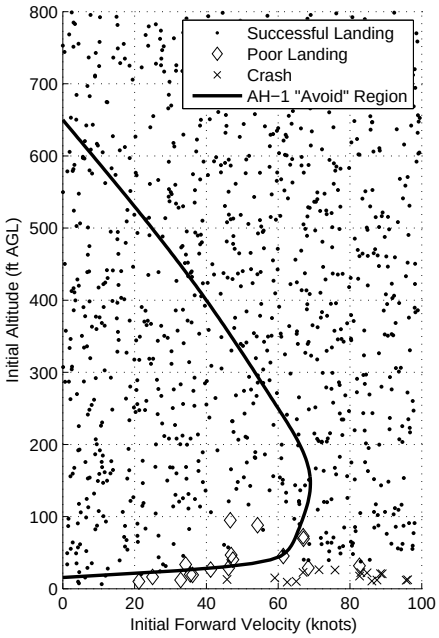
Figure 4.6a shows the results of 1000 simulated autorotation landings with an immediate handoff. Each solid dot represents a successful landing from the indicated position. A diamond indicates a landing that would likely result in damage to the vehicle, but equipment or passengers would not be in serious danger. An $\times$ indicates a crash. The specific thresholds for each of these categories are listed in Table 4.2. The low-speed "avoid" region of the H-V diagram for the Cobra helicopter is also marked. This curve is taken from Reference [22]. Note that the controller is able to perform a safe autorotation in nearly all cases, although some landings are not ideal. Figure 4.6b shows the results of 1000 simulated autorotations with a handoff delayed by 1 second, and Figure 4.6c shows the results for a handoff delayed by 2 seconds.

It is also interesting to consider a case in which vehicle is overweight, resulting in degraded autorotation performance due to the increase in disk loading. Figure 4.6d shows the results of 1000 simulated autorotations for an AH-1G with weight increased to 9000 lb with a handoff delay of 1 second. The control law and all of its parameters are identical to those used in previous tests.

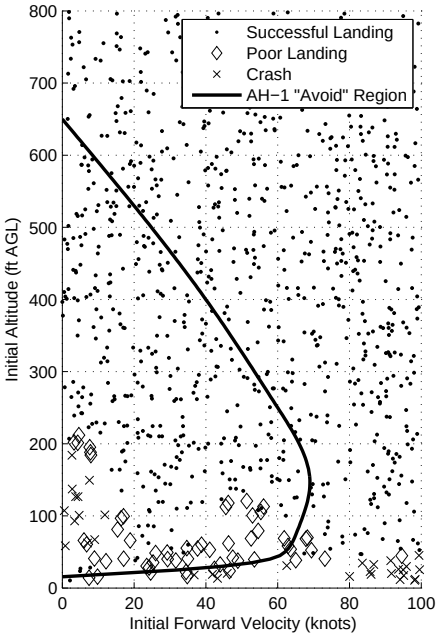
In all tests, the controller generally has difficulty at low altitudes and high speeds. This is typically a dangerous region of the height-velocity envelope since there is usually insufficient time to slow the helicopter to a safe speed before ground impact, and insufficient altitude to establish a steady state descent. This pathology may warrant



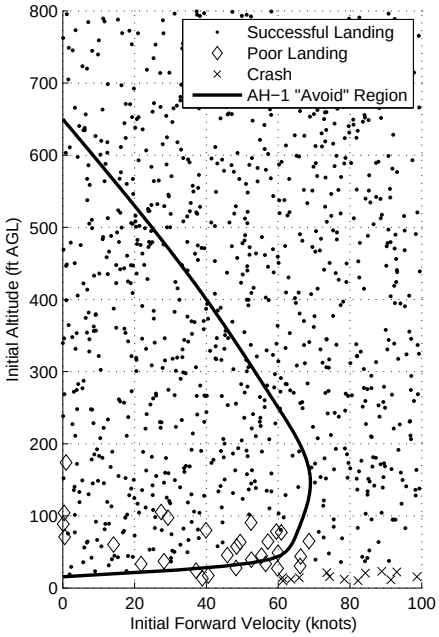
(a) Immediate handoff



(b) 1 sec handoff delay



(c) 2 sec handoff delay



(d) 1 sec delay, overweight

Figure 4.6: AH-1G Monte Carlo Simulation Results

Table 4.2: Conditions for Good and Poor Landings for the AH-1G. Simulations that do not meet these criteria are considered crashes. Conditions are applied to the absolute value of the parameter unless otherwise noted.

Parameter	*Condition for Good Landing*	*Condition for Poor Landing*
Roll angle, ϕ	$< 10^\circ$	$< 20^\circ$
Pitch angle, θ	$< 12^\circ$	$< 20^\circ$
Forward Speed, $\dot{x}$	< 30 knots	< 60 knots
Lateral Speed, $\dot{y}$	< 7 ft/s	< 10 ft/s
Vertical Speed, $\dot{z}$	< 5 ft/s	< 12 ft/s
Roll Rate, p	< 20 °/s	< 40 °/s
Pitch Rate, q	-30 °/s $< q < 20$ °/s	-50 °/s $< q < 40$ °/s
Yaw Rate, r	< 20 °/s	< 40 °/s

adjustment to the control law in the future to better handle this case. Overall, the Monte Carlo simulations presented here clearly demonstrate that the new control law holds the potential to significantly expand the safe H-V envelope when compared to a human pilot.

4.2 Align TREX 600 Simulations

The controller has also been applied to a model of the Align TREX 600 hobby-class helicopter to demonstrate its scalability. This simulation model has the same structure as the AH-1G model with different parameters, which are listed in Table 4.3. The rotor hub on the TREX Helicopter is semi-rigid meaning that the blades have pitch and lead-lag, but not flapping degrees of freedom. Blade flapping for a helicopter with this configuration can be modeled using an equivalent offset and hub spring [1]. Recent work [13] has suggested that a hinge offset of zero and reasonably stiff hub spring best match the dynamics of this specific helicopter.

Figures 4.7-4.9 show the state and control histories of a sample autorotation for the small helicopter. This simulation shows behavior that is similar in many ways

Table 4.3: Align TREX 600 Model Parameters Pertaining to Autorotation

Parameter	*Symbol*	*Value*
Helicopter gross weight	W	8.15 lb
Number of main rotor blades	N_b	2 (fixed hub)
Main rotor blade chord	c	0.1771 ft
Main rotor radius	R	2.208 ft
Main rotor blade moment of inertia	I_B	0.01357 slug ft^2
Main rotor height above ground (water line)	WL_{MR}	1.5 ft
Main rotor normal operating speed	Ω_{normal}	162 rad/s
Main rotor blade airfoil used for simulation		NACA 0012
Actuator max rate	$\dot{\delta}_{max}$	100 deg/s
Controller update rate		20 Hz

to the behavior of the larger helicopter. Figure 4.10 shows the histories of some internal time-to-impact controller states. The handoff delay is 3 seconds. This delay can clearly be seen in the θ_0 history in Figure 4.9 as the control law attempts to maintain altitude at the beginning of the autorotation.

The control histories show some high frequency oscillations in θ_{1s}. These are due to the stiff response of the helicopter caused by the rigid hub. There are also oscillations or peaks in the collective control. These are due to the $TTI \leq -2h/\dot{h}$ condition in the flare control law and interaction with the dependence of $TTLE$ (and therefore TTI_F) on vertical speed shown in Block 2.2. The oscillations in TTI_F due to this interaction are clearly visible in Figure 4.10. It should be noted that the amplitude of these oscillations is small at less than 1°.

Monte Carlo simulations were run on the TREX 600 model to demonstrate the controller's effectiveness in a wide variety of initial conditions. Landings are classified according to the criteria listed in Table 4.4. The results are presented in Figure 4.11. When the handoff delay is small, it is clear that the controller is able to handle a

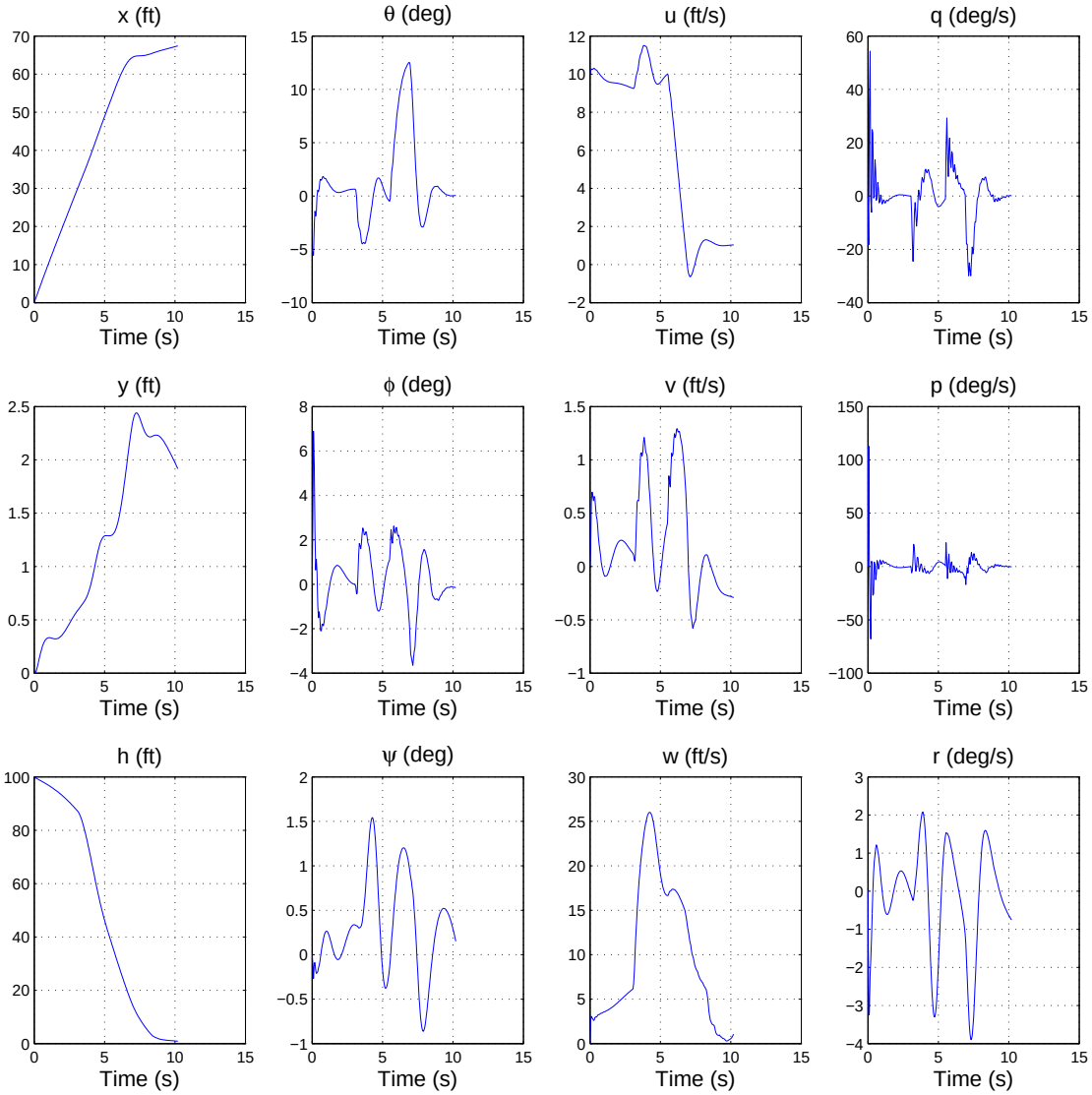


Figure 4.7: Helicopter Position, Orientation, Velocity, and Angular Rate State Histories for a Sample TREX 600 Autorotation Simulation (see Figure 4.8 for main rotor states)

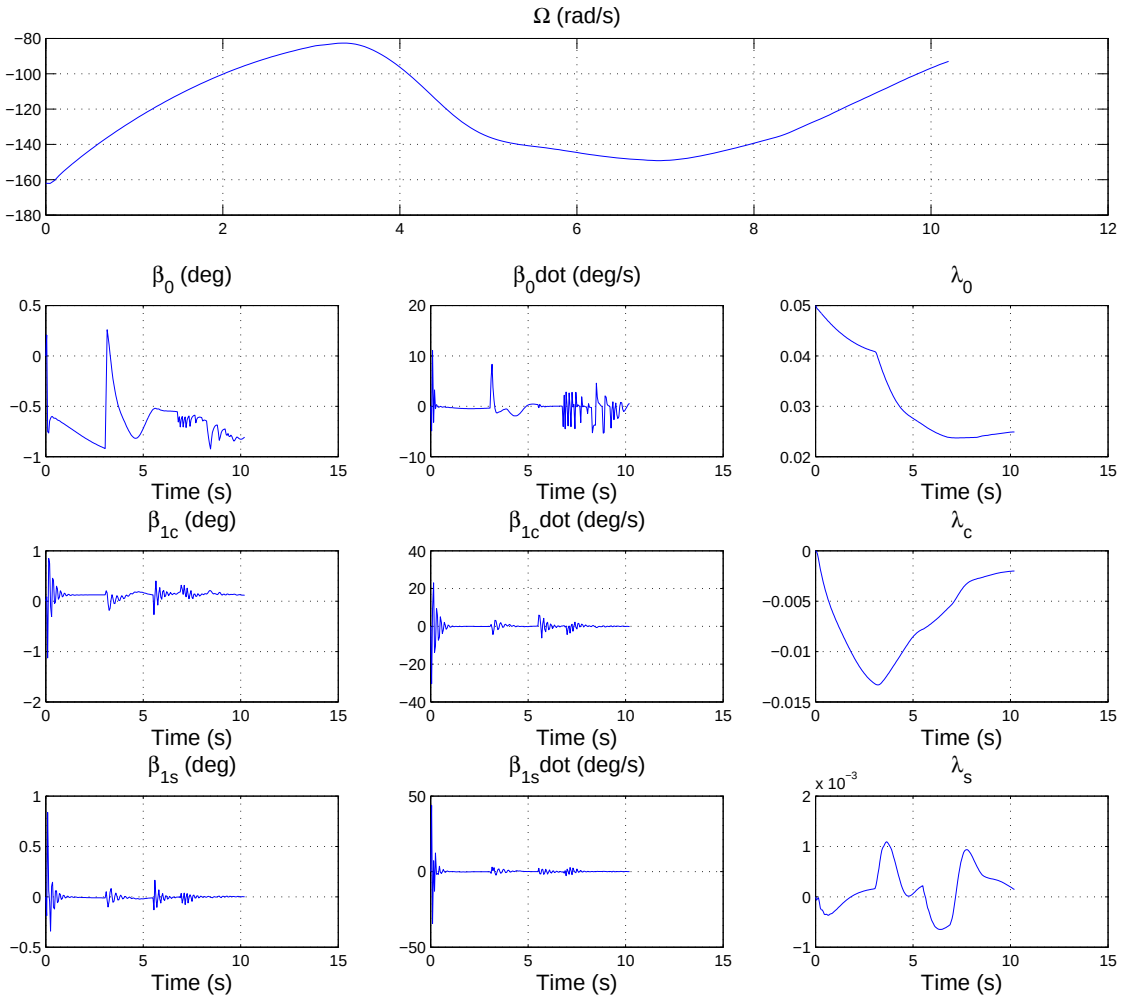


Figure 4.8: Helicopter Main Rotor State Histories for a Sample TREX 600 Autorotation Simulation

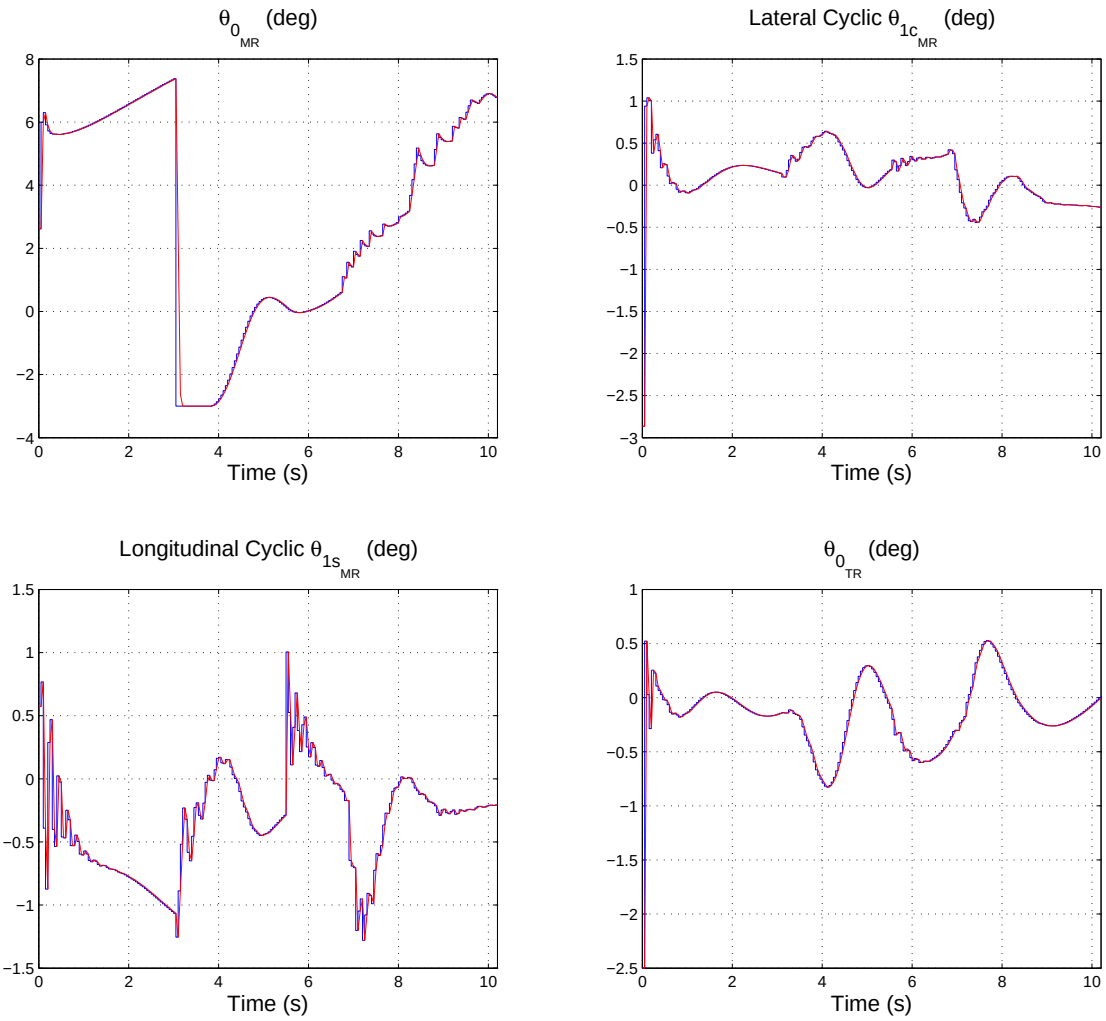


Figure 4.9: Control Histories for a Sample TREX 600 Autorotation Simulation

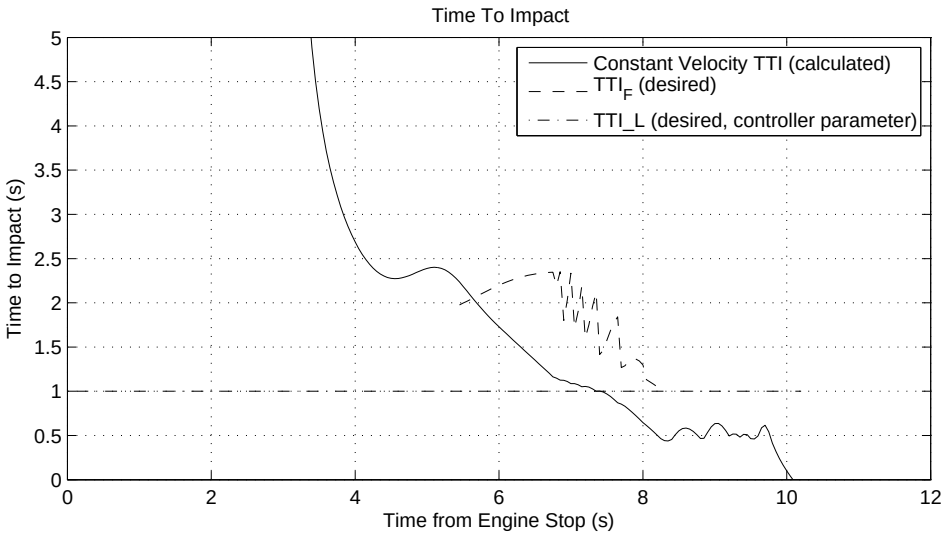
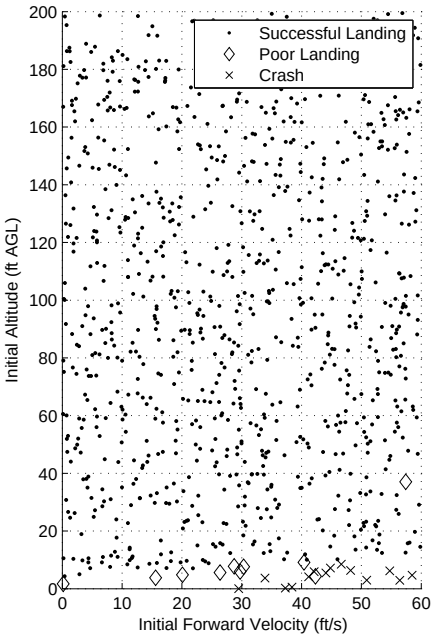


Figure 4.10: Controller Internal Time to Impact Variables for a Sample TREX 600 Autorotation. This plot shows the values of several internal controller states. Note that $TTLE$ can be read off the plot as the difference between TTI_F and `TTI_L`

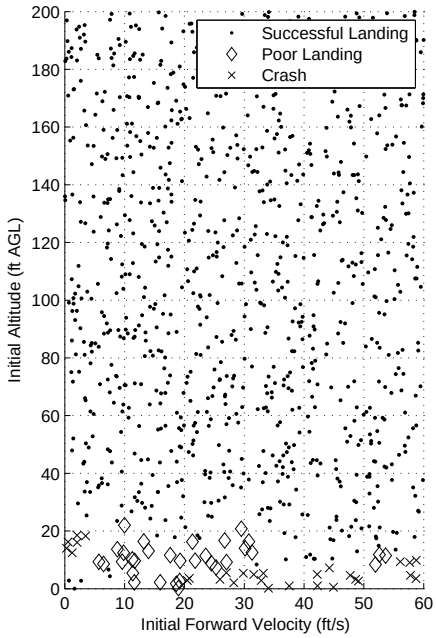
wide variety of initial conditions. As the handoff delay is increased, performance becomes significantly worse because of the decrease in rotor RPM caused by the controller attempting to maintain altitude before handoff. The controller is able to handle a doubling of the aircraft's weight without change when the initial altitude is high, but it is unable to handle this change without parameter adjustment when the initial altitude is low. These results clearly show that the controller is scalable for use in small unmanned helicopters as well as large ones.

Table 4.4: Conditions for Good and Poor Landings for the TREX 600. Simulations that do not meet these criteria are considered crashes. Conditions are applied to the absolute value of the parameter unless otherwise noted.

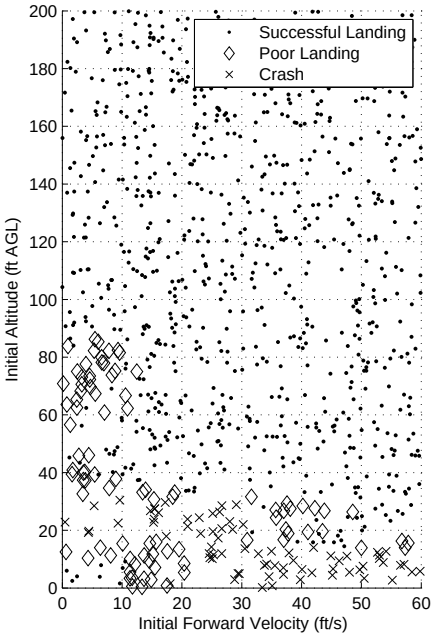
Parameter	*Condition for Good Landing*	*Condition for Poor Landing*
Roll angle, ϕ	$< 10^\circ$	$< 20^\circ$
Pitch angle, θ	$< 12^\circ$	$< 20^\circ$
Forward Speed, $\dot{x}$	< 10 ft/s	< 20 ft/s
Lateral Speed, $\dot{y}$	< 5 ft/s	< 10 ft/s
Vertical Speed, $\dot{z}$	< 5 ft/s	< 12 ft/s
Roll Rate, p	< 20 °/s	< 40 °/s
Pitch Rate, q	-30 °/s $< q < 20$ °/s	-50 °/s $< q < 40$ °/s
Yaw Rate, r	< 20 °/s	< 40 °/s



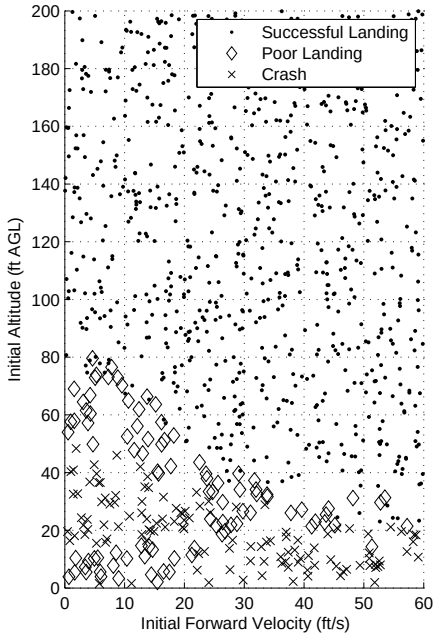
(a) Immediate handoff



(b) 1 sec handoff delay



(c) 3 sec handoff delay



(d) 1.5 sec delay, double weight

Figure 4.11: TREX 600 Monte Carlo Simulation Results

5. CONCLUSION

A new highly scalable expert system control law for autonomous autorotation has been proposed and validated through simulation. This controller uses a blended set of control laws to guide the helicopter through the phases of an autorotation. One key innovation is the use of approximate reasoning based on the desired time to impact during the flare phase. Monte Carlo simulations using a high fidelity model of a full size helicopter clearly demonstrate that the controller has the potential to expand the safe flight envelope for single engine rotorcraft. Simulations of a small remote control helicopter indicate that the control law is scalable to a wide range of sizes.

REFERENCES

[1] J. G. Leishman, *Principles of Helicopter Aerodynamics*, 2nd ed. New York: Cambridge University Press, 2006.

[2] W. Johnson, "Helicopter optimal descent and landing after power loss," NASA, Tech. Rep. TM-73,244, 1977.

[3] A. Y. Lee, A. E. Bryson, and W. S. Hindson, "Optimal Landing of a Helicopter in Autorotation," *Journal of Guidance, Control, and Dynamics*, vol. 11, no. 1, pp. 7–12, 1988.

[4] E. N. Bachelder, D.-C. Lee, and B. L. Aponso, "Autorotation Flight Control System," Patent 7 976 310 B2, 2011.

[5] K. Hazawa, J. Shin, D. Fujiwara, D. Igarashi, Kazuhiro Fernando, and K. Nonami, "Autonomous Autorotation Landing of Small Unmanned Helicopter," *Transactions of the Japan Society of Mechanical Engineers*, vol. 70, no. 698, pp. 2862–2869, 2004.

[6] P. Abbeel, A. Coates, T. Hunter, and A. Y. Ng, "Autonomous autorotation of an RC helicopter," in *Experimental Robotics*. Springer, 2009, pp. 385–394.

[7] K. Dalamagkidis, K. P. Valavanis, and L. A. Piegl, "Autonomous Autorotation of Unmanned Rotorcraft using Nonlinear Model Predictive Control," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, pp. 351–369, Sep. 2009.

[8] L. Dooley and R. Yeary, "Flight test evaluation of the high inertia rotor system," US Army Research and Technology Laboratories, Tech. Rep. USARTL-TR-79-9, Jun. 1979.

[9] T. Yomchinda, J. Horn, and J. Langelaan, "Flight Path Planning for Descent-phase Helicopter Autorotation," in *AIAA Guidance, Navigation, and Control Conference*, Portland, Oregon, Aug. 2011, pp. 1–24.

[10] S. Tierney and J. Langelaan, "Autorotation path planning using backwards reachable set and optimal control," in *American Helicopter Society Forum*, 2010.

[11] J. Keller, R. M. Jr, J. Horn, and T. Yomchinda, "Active Flight Control and Appliqué Inceptor Concepts for Autorotation Performance Enhancement," in *American Helicopter Society Forum*, Virginia Beach, VA, 2011.

[12] M. Livchitz, A. Abershitz, U. Soudak, and A. Kandel, "Development of an automated fuzzy-logic-based expert system for unmanned landing," *Fuzzy Sets and Systems*, vol. 93, no. 2, pp. 145–159, 1998.

[13] B. Taylor and J. Rogers, "Experimental Investigation of Helicopter Weight and Mass Center Estimation," in *AIAA Aerospace Sciences Meeting*, Jan. 2013.

[14] P. D. Talbot, B. E. Tinling, W. A. Decker, and R. T. N. Chen, "A Mathematical Model of a Single Main Rotor Helicopter for Piloted Simulation," NASA, Tech. Rep. TM-84281, 1982.

[15] R. T. N. Chen, "A Simplified Rotor System Mathematical Model for Piloted Flight Dynamics Simulation," NASA, Tech. Rep. TM-78575, May 1979.

[16] R. Chen, "Effects of primary rotor parameters on flapping dynamics," NASA Ames Research Center, Tech. Rep. TP-1431, Jan. 1980.

[17] R. E. Sheldahl and P. C. Klimas, "Aerodynamic characteristics of seven symmetrical airfoil sections through 180-degree angle of attack for use in aerody-

namic analysis of vertical axis wind turbines," Sandia National Labs, Tech. Rep. SAND-80-2114, Mar. 1981.

[18] D. A. Peters and N. HaQuang, "Dynamic Inflow for Practical Applications," *Journal of the American Helicopter Society*, pp. 64–68, Oct. 1988.

[19] D. M. Pitt and D. A. Peters, "Theoretical prediction of dynamic-inflow derivatives," *Vertica*, vol. 6, pp. 21–34, 1981.

[20] D. A. Peters and G. H. Gaonkar, "Effectiveness of Current Dynamic-Inflow Models in Hover and Forward Flight," *Journal of the American Helicopter Society*, pp. 47–57, Apr. 1986.

[21] H. Heyson, "Ground effect for lifting rotors in forward flight," NASA, Tech. Rep. D-234, May 1960.

[22] L. M. Free and L. J. Hepler, "Height-Velocity Test: AG-1G Helicopter at Heavy Gross Weight," US Army Aviation Systems Command, Edwards AFB, CA, Tech. Rep. USAASTA Project No. 74-19, Jun. 1974.