The Dissertation Committee for Ofer Eldad Certifies that this is the approved version of the following dissertation:

Uncertainty Management in Solar Sail Attitude Control

Committee:

Wallace Fowler, Supervisor

Christian Claudel

E. Glenn Lightsey

Efstathios Bakolas

Brandon Jones

Uncertainty Management in Solar Sail Attitude Control

by

Ofer Eldad, B.S.; M.E.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

The University of Texas at Austin

December 2016

To Anat, Amos, and Oscar for their love.

Acknowledgements

I would like to start by thanking three people that I feel have acted as my advisors, either officially or unofficially throughout the years. First, Dr. Glenn Lightsey for warmly welcoming me into the department and introducing me to solar sails. Thank you for supporting me along the way and providing helpful advice and feedback even after leaving the university. I would also like to thank Dr. Christian Claudel for taking me on as a research assistant and infecting me with the necessary optimism for tackling new and interesting problems that I probably would not have attempted otherwise. I have learned much from our various explorations and discussions and above all appreciate your kind attitude and support. Finally, I would like to thank Dr. Wallace Fowler for taking on the capacity of my advisor and always having an open door and attentive ear to help guide me along the way.

Thank you to the rest of my committee members Dr. Efstathios Bakolas and Dr. Brandon Jones for taking the time to review my dissertation and participating in my defense. To Brandon Schmitt and his team for welcoming me to the Sunjammer mission. The project will always be one of the lasting memories and significant experiences from my PhD. I would like to thank my friends for making my time in Austin one of the more enjoyable periods in my life. Knowing that I could walk down the hall and always find a helping hand, or a squash partner, has made the difference between good days and bad.

Thank you to my parents, my brother Dan, and sister Osnat. Your love and support, even from thousands of miles away, have made a world of a difference. Finally, to my wife Anat, the greatest treasure I discovered in Austin, for everything that you are and for being a constant reminder of the important things in life.

Uncertainty Management in Solar Sail Attitude Control

Ofer Eldad, Ph.D.

The University of Texas at Austin, 2016

Supervisor: Wallace Fowler

Solar sails are emerging as a viable alternative to conventional forms of propulsion. Still at their infancy and relatively untested, many sources of uncertainty remain that are unique to solar sails and which will continue to affect their design as solar sails increase in performance and size. Controlling their attitude in the context of these uncertainties therefore becomes critical to spaceflight missions that will explore our solar system and beyond from new, previously-unattainable perspectives. Two distinct frameworks are developed to manage these uncertainties to control the attitude of solar sails. The first utilizes a provided system model and utilizes an observation of the control history to operate the sail about an equilibrium position that is passively stable. The approach utilizes past information about controller input for the purposes of rejecting disturbances that arise from several sources of uncertainty.

The second approach is forward-looking and is inspired by trajectory-based reachability analysis. This approach was developed in the context of six degree-of-freedom supervised control of an unmanned aerial vehicle whose faster dynamics in a disturbance-rich environment provide a computational challenge and thus require machine-learned approximations to a reachable set of safe inputs.

These methodologies are then applied to the original solar sail attitude control problem. Predictions are made about the future state of the sail after performing a minimum-time large angle maneuver. Uncertainty distributions are assumed a priori and are then used to create a buffer angle for the maneuver such that no overshoot occurs within a tunable statistical measure of safety. Uncertainties handled in this way include the sail effective reflectivity, flexural rigidity, and moment of inertia. However, the framework is designed to be very adaptable and so is able to accommodate arbitrary sources of uncertainties and flexible modeling techniques. Utilization of machine learning allows for arbitrary complexity in the simulation and modeling framework without impacting the on-board computational requirements of the solar sail hardware.

Table of Contents

Acknowledgements v

Abstract vi

List of Tables x

List of Figures xi

Chapter 1 Introduction 1

1.1 Contributions 4

1.2 Dissertation Organization 5

Chapter 2 Propellantless Attitude Control of a Nonplanar Solar Sail 7

2.1 Introduction 8

2.2 The Sail Model 10

2.3 Solar Radiation Forces and Moments Model 11

2.4 Passive Stability 14

2.5 Control Law and Moment Allocation 15

2.6 Trim Angles and their Autonomous Adjustment 16

2.7 Effect of Passive Stability 20

2.8 Results 24

2.9 Conclusion 31

Chapter 3 Real time Computation of Safe Input Sequences for UAVs 32

3.1 Introduction 32

3.2 Dynamical Model of a Fixed-Wing UAV 36

3.3 Algorithm Overview 42

3.3.1 Functional Relation Generation (Performed Offline) 42

3.3.2 Online Usage 43

3.4 Cost Function Calculation 45

3.5 Ellipsoidal Approximation 48

3.6 Machine Learning Representation 51

3.7 Experimental Platform 55
3.8 Results and Limitations 57
3.8 Conclusions 62

Chapter 4 Minimum-Time Attitude Control of Deformable Solar Sails with Model Uncertainty 64
4.1 Introduction 64
4.2 Deformable Model 68
4.3 Time-Optimal Control 74
4.4 Trajectory-Based Reachability 79
4.5 Machine Learning for Uncertainty Management 84
4.6 Conclusions 89

Chapter 5 Conclusions 91
5.1 Summary of work 91
5.2 Future Work 93

Appendix A: Chapter 2 Coordinate Systems 95

Appendix B: Chapter 3 Nomenclature and Coordinate Systems 97

Bibliography 100

List of Tables

Table 2.1: Maneuvers used for controller performance testing 24

Table 2.2: Summary of disturbance sources and their magnitude 25

Table 2.3: Performance metrics for controller evaluation 26

Table 2.4: Baseline performance metrics (no disturbances) 29

Table 2.5: Simulation results 30

Table 3.1: Range of Applicability of Machine Learned Function 54

Table 4.1: Sail properties 67

Table 4.2: Sail boom tip deflection by source 70

Table 4.3: Minimum-time results for three example solar sails 79

Table 4.4: System identification results 86

Table B.1: Undefined constants used in Chapter 3 98

Table B.2: Undefined functions of angle of attack used in Chapter 3 (dimensionless) 99

List of Figures

Figure 1.1 Three generations of solar sails next to the empire state building 2

Figure 2.1: Ground deployment of one quadrant of Sunjammer 7

Figure 2.2: Sail schematic and axis definition 8

Figure 2.3: Resultant force from specular and diffuse solar radiation.....12

Figure 2.4: Experimental vs. flat-plate model moments.....13

Figure 2.5: Passive stability in a Sun-pointing orientation 14

Figure 2.6: Moment allocation scheme for generating moments about the y-axis 16

Figure 2.7: Effect of trim angle adjustment on control vane motion.....19

Figure 2.8: Effect of trim angle adjustment on pointing accuracy 20

Figure 2.9: Control effort without passive stability 21

Figure 2.10: Spacecraft attitude without passive stability 22

Figure 2.11: Control effort with passive stability 23

Figure 2.12: Attitude with passive stability 23

Figure 2.13: Sample attitude time history for the baseline case - maneuver 1 27

Figure 2.14: Cant angle time history for baseline case – maneuver 1 28

Figure 2.15: Twirl angle time history for baseline case – maneuver 1.....28

Figure 3.1: Simulated Input Signals over the Receding Time Horizon.....40

Figure 3.2: Example Disturbance 41

Figure 3.3: Algorithm overview 45

Figure 3.4: Polygon Pαβ - Allowable envelope for angle of attack and sideslip ..46

Figure 3.5: Comparison of convex hull, ORH, and ellipsoidal approximation 48

Figure 3.6: Bormatec MAJA UAV during initial test flights 55

Figure 3.7: Hardkernel Odroid XU-3 (bottom) interfaced with laptop and Pixhawk autopilot (above Odroid) 56
Figure 3.8: Safety ellipsoid for minimum height constraint – with disturbances 57
Figure 3.9: Simulation of UAV height 58
Figure 3.10: Simulation of UAV z-acceleration 59
Figure 3.11: Simulation of UAV angle of attack 59
Figure 3.12: Simulation of UAV airspeed 60
Figure 3.13: Safety ellipsoid for minimum height constraint – no disturbances 60
Figure 3.14: Safe and unsafe trajectories - no disturbances 61
Figure 3.15: Example of non-connected safe input set 62
Figure 4.1: Maximal boom deflection – Sail 1 72
Figure 4.2: Maximal boom deflection – Sail 2 72
Figure 4.3: Maximal boom deflection – Sail 3 73
Figure 4.4: Maximal boom slope – Sail 3 74
Figure 4.5: Maximal available moment 76
Figure 4.6: Desired tip-vane angle 76
Figure 4.7: Convergence to desired switch and final times 78
Figure 4.8: Pitch angle time history 78
Figure 4.9: Pitch rate time history 79
Figure 4.10: Convergence on buffer angle –Iteration 0, Buffer = 0° 81
Figure 4.11: Convergence on buffer angle –Iteration 1, Buffer = 3.1° 82
Figure 4.12: Effect of uncertainty on required buffer angle – Sail 1 83
Figure 4.13: Effect of uncertainty on required buffer angle – Sail 2 83
Figure 4.14: Effect of uncertainty on required buffer angle – Sail 3 84
Figure 4.15: Learned flexural rigidity for Sail 3 87

Figure A.1 Vane frames (in red) and sail frame (in black) when 0° cant and twirl angles 95
Figure A.2: Sun pointing frame [12].....96

Chapter 1
Introduction

Solar sails are spacecraft that utilize large sails to reflect sunlight as a means for thrust generation. They enable a multitude of new mission concepts and unique non-Keplerian orbits that are either currently unattainable or prohibitively expensive. By exploiting the momentum carried by photons from the Sun, spacecraft that utilize solar sails have the potential of traveling to regions currently unreachable by conventional forms of propulsion. Owing to the continuous force supplied by the solar radiation pressure (SRP), solar sails can be particularly beneficial in missions requiring high Δv. The possibilities are mostly limited by the lifetime of the sail. Missions such as a Mercury sample return [1], Solar Polar Orbiter, Kuiper-Belt fly-through [2], and many others have been suggested to take advantage of solar sails' properties.

The existence of a pressure produced by light was shown by Maxwell in 1873, and its application to solar sailing design began showing up the 1920s in works by soviet scientists Friedrich Zander and Konstantin Tsiolkovsky. Solar sailing became popularized in the US with Arthur C. Clarke's "Sunjammer" short science fiction story first published in 1963. In the 1970s NASA began low-level studies of solar sailing which shifted towards design of a solar sail that could rendezvous with the Halley comet in the mid-1980s. An initial design of an $800m \times 800m$ sail was later dropped in favor of a spin-stabilized heliogyro design which had twelve 7.5km long blades of film [14]. The mission design shifted once again and an alternative form of propulsion was selected and the mission was later cancelled due to cost concerns. Recent missions such as JAXA's IKAROS, NASA's Nanosail-D, The Planetary Society's LightSail 1 along with the upcoming LightSail 2 highlight growing international interest in solar sailing as an alternative to traditional forms of propulsion.

In order to produce appreciable accelerations that enable a greater variety of orbits, the ratio of solar sail area to overall spacecraft mass must remain high and in fact increase. This has led a drive towards lighter sail structures and thinner sails along with an effort to increase solar sail size and performance as highlighted by NASA's technology roadmap [3]. The potential of using solar sails is also increasing as satellite masses have decreased due to miniaturization technologies.

Current solar sail designs have been modestly sized. LightSail has an area of 32 m^2, Nanosail-D has an area of 10 m^2, and IKAROS an area of 200 m^2. The proposed and defunded Sunjammer mission had the biggest area at 1200 m^2, however, it too is small when compared to the solar sail sizes envisioned for truly revolutionary missions such as the Halley rendezvous mission (640,000 m^2). In July 2015, NASA's technological roadmap projected solar sail technology to progress from a $40m \times 40m$ 1st generation sail with areal density of 10 to 25 g/m^2 to a $150m \times 150m$ 2nd generation sail with an areal density of <10 g/m^2 to a 3rd generation sail that is $300m \times 300m$.

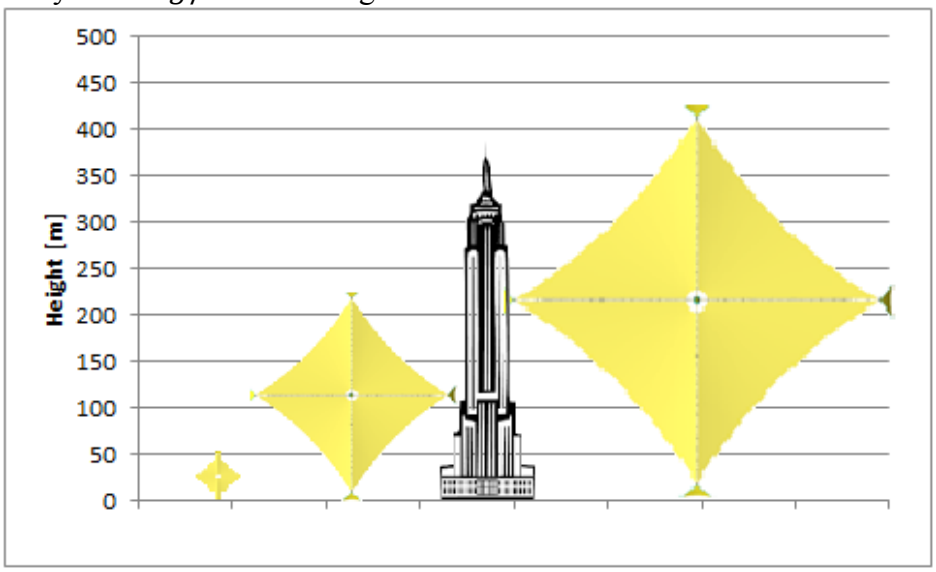

Figure 1.1 Three generations of solar sails next to the empire state building

The report cites ultra-lightweight flexible materials which are more mass efficient than metalized films with sufficient thermal and mechanical durability and flexibility for long duration missions as the key technological development needed to achieve progress between generations of solar sails.

The force produced by a solar sail can be directed in order to change the velocity of the spacecraft. This change of velocity can thus directly affect the orbit's semi-major axis, inclination, and eccentricity, and can indirectly change all other parameters of the orbit and trajectory. Thus control of the spacecraft's trajectory is accomplished by controlling the sail's attitude relative to the Sun, leading to a variety of solutions that have been proposed for solar sail attitude control [4, 5]. Since solar sails provide a low thrust, missions that utilize solar sails are typically longer. The Halley rendezvous trajectory would have taken approximately 4 years just to achieve rendezvous, and the Mercury sample return mission proposed in [1] would take 4.4 years. To take full advantage of the long duration of solar sail missions, an approach which requires little to no fuel is preferred. Proposed solutions include actively changing the mass distribution of the sail [6], utilizing control vanes [7], sail optical properties manipulation, and the use of highly efficient pulse plasma thrusters [8].

Regardless of the means of providing attitude control, most proposed solutions approximate the sail geometry as a flat plate and thus ignore significant shape-related effects [9]. In addition, the rigid body assumption for the sail is usually used, which will become less valid as the achievable solar sail size increases and its areal density decreases. A control scheme that appears to work well utilizing a flat, rigid plate model may fail when considering moment biases and sail imperfections that are expected in the sail's real environment. Flight data from the IKAROS mission suggests that shape effects are not negligible [10]; therefore, they must be taken into consideration in a realistic

control system design. While the effects of a shaped sail have been presented in literature [9], they have only recently started to be incorporated in a design of closed-loop attitude control systems.

1.1 CONTRIBUTIONS

This dissertation creates new algorithms which facilitate the use of higher performance sails of increasing flexibility and robustly manage uncertainties in the attitude control of solar sails, including unknown model parameters. Several approaches are taken to this end. Initially, a simulation framework for attitude determination and control is implemented. The developed algorithm uses a concept derived from trimming aircraft about an equilibrium state which is applied to propellantless solar sail attitude control. In addition, this controller incorporates a tunable magnitude of passive stability about this equilibrium state to further reduce required control action. Since it is difficult to test and model the forces and moment experienced by solar sails, a robust and adaptable controller is needed. A formulation for observing past system behavior in order to estimate and correct for disturbances is developed in this framework in order to achieve the desired robustness. Then, the perspective of the research changes from looking at past behavior to looking at possible future states of the system inspired by reachability concepts. This approach is tested using a UAV platform which leads to the development of real time computation of safe future input sets for UAVs. Finally, the forward-looking perspective is adapted in development of large-angle minimum-time attitude control of a flexible solar sail with model uncertainties. Another contribution that is made in the forward-looking approach is creating a separation between real time control and offline predictions of future states such that high levels of performance can be obtained in a realizable framework, i.e. one which does not require unrealistic

computational power. This is accomplished using machine learning. Machine learning is also shown to assist in reducing uncertainties in model parameters through a proposed offline learning and quick on-orbit calibration procedure.

1.2 Dissertation Organization

This dissertation is divided into three main chapters. Chapter 2 is inspired by the author's work in developing an attitude control system for the Sunjammer solar sailing mission. The chapter develops a robust disturbance rejection controller wrapped around a standard PD controller for a propellantless solar sail. The control system incorporates passive stability and utilizes the concept of learned equilibrium conditions based on control surface time histories to minimize the required motion in the actuating tip-vanes and to counteract various sources of uncertainties. These uncertain parameters include bias and scaling in the modeled solar radiation pressure, center of mass – center of pressure offset, spacecraft moments of inertia, actuator orientation, and state estimation errors. The model used for this chapter is one which includes effects of sail billowing and wrinkling, however, since it was provided as a "black-box", little information is available on how it was constructed and thus it is treated as a reference model with the above sources of uncertainties.

Chapter 3 shifts focus away from solar sails in order to investigate reachability concepts and the possibility of using them in real time embedded systems. The testbed used for this investigation is a UAV where control of its six degrees of freedom is explored. A trajectory-based reachability algorithm is developed in which all admissible inputs and disturbances are used in generating trajectories from an uncertain initial state over a fixed time horizon. The set of inputs which do not violate state constraints are deemed safe and are approximated using an ellipsoid. The algorithm utilizes a machine-

learned ellipsoidal approximation for a range of UAV states in order to be able to rapidly formulate a sense of allowable inputs and supervise and correct any inputs from an autopilot or human operator so that the UAV remains in a safe state. Development of these control algorithms on a readily available platform with accessible and modifiable hardware and software helps validate them and provides intuition for the solving the problem at hand.

Chapter 4 then applies the techniques introduced in Chapter 3 back to solar sails in order to construct a time-optimal control for large angle maneuvers about the pitch axis using a simplified flexible solar sail model. The model uncertainties of three sails of increasing size and flexibility are investigated in this process. These uncertainties lead to a required buffer on the target orientation in order to avoid overshoot. A machine-learned relationship between model uncertainties and the required buffer angle is used for computational reasons. The model uncertainties included in this chapter are the following sail parameters: flexural rigidity, surface reflectance, and moment of inertia. A process for reducing the uncertainties of these parameters through learning the sail's dynamic behavior in a given maneuver is established. This approach allows for more efficient maneuvers with a smaller required buffer angle to avoid overshoot.

The dissertation closes with concluding remarks and a discussion of future work as an extension to topics covered herein.

Chapter 2
Propellantless Attitude Control of a Nonplanar Solar Sail

As an initial example, a robust propellantless attitude control scheme for solar sails is developed for a square sail with articulating control vanes[1]. The algorithm was developed in support of NASA's Sunjammer mission, which was to use a $35m \times 35m$ solar sail for technology demonstration. After a preliminary design activity, the mission was cancelled by NASA in 2015.

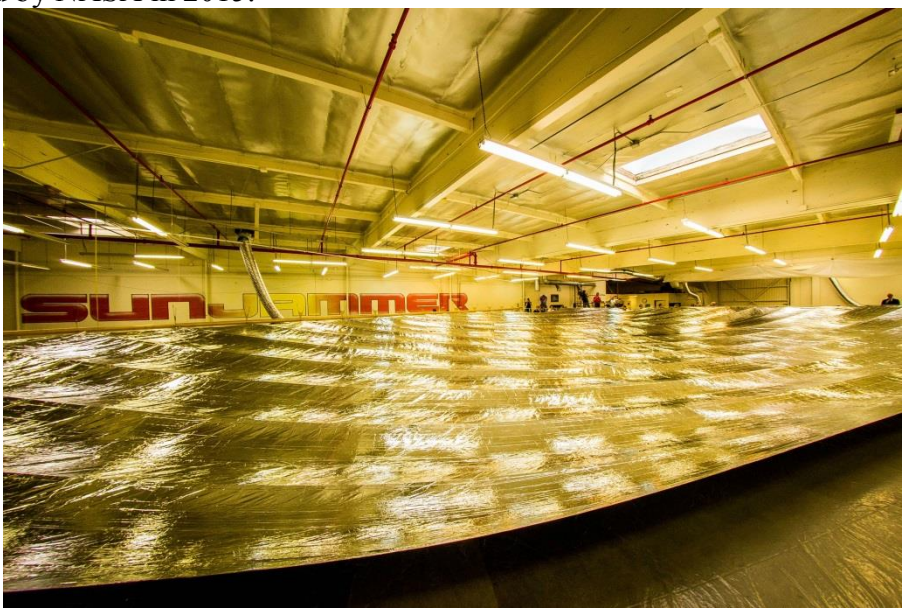

Figure 2.1: Ground deployment of one quadrant of Sunjammer

The algorithm incorporates an experimentally-derived sail moment and force model which includes non-planar sail effects. For example, the model produces a far

[1] The work discussed in this chapter has been published in a peer-reviewed journal and presented at a conference. Analysis and write-up was performed by Ofer Eldad with supervision by Glenn Lightsey.

- Eldad, O., and Lightsey, E.G., "Propellantless Attitude Control of a Nonplanar Solar Sail," *Journal of Guidance, Control, and Dynamics,* Vol. 38. No. 8, 2015, pp. 1531-1534
- Eldad, O. and Lightsey, E.G., "Attitude Control of the Sunjammer Solar Sail Mission" *Proceedings of the AIAA/USU Conference on Small Satellites*, AIAA/Utah State Univ. Paper SSC14-X-4, Logan, UT, 2014.

greater restoring moment than predicted by the idealized flat-plate model. The controller design builds upon a novel method of eliminating moment bias through the use of trim angles and takes advantage of passive stability about 2 axes. The trim angles are calculated a-priori but are adjusted on-board based on observing the control history of a standard PD-controller. The design is shown to achieve a pointing accuracy better than 2 degrees for a variety of maneuvers and under a wide range of disturbances.

2.1 INTRODUCTION

This chapter investigates 3-axis attitude control of a standard square-sail design with four control vanes, one at each corner of the sail shown in Figure 2.2.

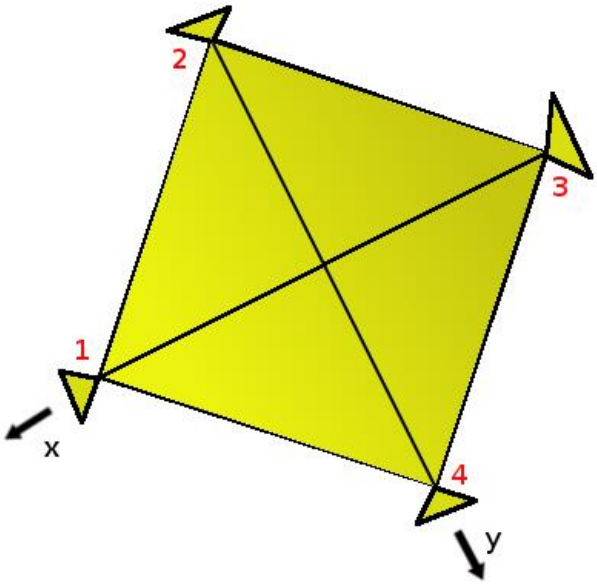

Figure 2.2: Sail schematic and axis definition

A proposed control strategy that incorporates passive stability and trimming the sail moments about the x- and y-axes has been previously presented [11]. In a sun-pointing configuration, stability about the x-axis is achieved by moving both control vanes 2 and 4 away from the Sun by an identical angle. This configuration is passively

stable since a positive rotation about the x-axis results in vane 4 having a larger projected area towards the Sun, while vane 2's projected area to the Sun is decreased. This combination produces a negative moment about the x-axis which forces the sail back toward the equilibrium sun-pointing orientation. This effect is similar to the dihedral effect on an aircraft. A similar methodology is then applied to stabilize rotation about the y-axis using control vanes 1 and 3. The vane angles can be adjusted to achieve passive stability in all other orientations. To accurately control the sail's attitude, this strategy requires accurate pre-determination of the on-orbit moments.

This chapter extends this approach in several ways and demonstrates its effectiveness in controlling solar sail attitude in the presence of disturbance torques. These torques arise primarily from a lack of an accurate prediction of the solar radiation pressure moments and forces, deviation of the flexible solar sail from the rigid body assumption, unmodeled lifetime changes of the sail material's optical properties, and on-orbit faults such as rips in the sail material. The controller uses an initial guess of the trim angles necessary for each desired attitude relative to the Sun and adjusts this guess based on the observed sail behavior. In addition, the controller implements a standard PD controller in order to approach the trimmed orientation and facilitate convergence and disturbance rejection.

The control scheme developed here uses the command time history to estimate the moment bias experienced by the spacecraft and is thus much more capable in achieving robust attitude control in the presence of model uncertainty and error. The performance of the controller is determined using variations in solar radiation pressure moment and forces (scaling and bias errors), sailcraft moments of inertia, unmodeled bend and twist of the control vanes, and a z-axis center of mass offset. The demonstration of a robust propellantless closed-loop attitude control system utilizing non-flat-plate sail forces and

moments creates a flyable and realizable design that enables wider adoption of solar sails as an alternative means of spacecraft propulsion.

The remainder of the chapter is organized as follows: Sections 2.2 and 2.3 discusses the sail dynamic model and the forces and moments model. Section 2.4 discusses the concept of passive stability. Section 2.5 then describes how the desired moment is calculated and how it is allocated to the four vanes for attitude control. Trim angle and the algorithm for their autonomous adjustment are presented in Section 2.6 and the effects of passive stability is shown in Section 2.7. Results of applying this algorithm to the solar sail are shown in Section 2.8 and this chapter end with concluding remarks in Section 2.9.

2.2 THE SAIL MODEL

A rigid body model of the sail is assumed, and thus the equations of motion are,

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2}(q_4\boldsymbol{\omega} - \boldsymbol{\omega} \times \boldsymbol{q}) \tag{2.1}$$

$$\dot{q}_4 = -\frac{1}{2}\boldsymbol{\omega}^{\boldsymbol{T}}\boldsymbol{q} \tag{2.2}$$

$$\dot{\boldsymbol{\omega}} = I^{-1}(\boldsymbol{M} + \boldsymbol{d} - \boldsymbol{\omega} \times I\boldsymbol{\omega}) \tag{2.3}$$

where $\boldsymbol{q}$ is a quaternion representing the orientation of the body frame in relation to an inertial frame, $\boldsymbol{\omega}$ is the angular velocity vector, $\boldsymbol{M}$ is the control input torque applied by the sail and vanes about the system's center of mass, $\boldsymbol{d}$ is the disturbance torque, and I is the system's inertia matrix. Moments of inertia representative of the Sunjammer solar sail are used in subsequent simulations. The inertia matrix utilizes the body axes defined in Figure 2.2 and is diagonal due to symmetry when all vane angles are zero.

Disturbance torques arise primarily from a lack of an accurate prediction of the solar radiation pressure moments and forces, deviation of the flexible solar sail from the rigid body assumption.

While quaternions are used in the control algorithm, the attitude of the sail is presented in terms of 3 sun angles for the purpose of readability. These angles represent a 3-2-3 rotation sequence of Euler angles. Beginning with a coordinate system centered at the sail with its z-axis pointing towards the Sun and the x-axis parallel to the ecliptic plane, the Euler sequence rotates by a top angle about the 3-axis, by a sun-incidence angle about the 2-axis and ends with a rotation by a flat-spin angle about the 3-axis to arrive at the body frame. The sun-incidence angle most directly affects the magnitude of the force created by the solar sail, while the top angle is used to direct the force vector in the desired direction. An explanation of the coordinate systems used in this chapter is available in Appendix A.

2.3 Solar Radiation Forces and Moments Model

A commonly used model in literature for the force produced by solar radiation is presented in [14] and reproduced in Equation 2.4.

$$F = PA\{[(1+\sigma_{rs})\cos^2\alpha + B_f\sigma_{rd}\cos\alpha]\hat{\boldsymbol{n}} + (1-\sigma_{rs})\cos\alpha\sin\alpha\,\hat{\boldsymbol{t}}\} \quad (2.4)$$

where P is the solar radiation pressure equal to $4.56\times10^{-6}\ N/m^2$ at the Earth, A is the area of the reflective surface, σ_{rs} is the coefficient of specular reflection, σ_{rd} is the coefficient of diffuse reflection, $\hat{\boldsymbol{n}}$ and $\hat{\boldsymbol{t}}$ are the surface normal and transverse unit vectors, B_f is the non-Lambertian coefficient, and α is the sun-incidence angle as illustrated in Figure 2.3 for the 2-dimensional case:

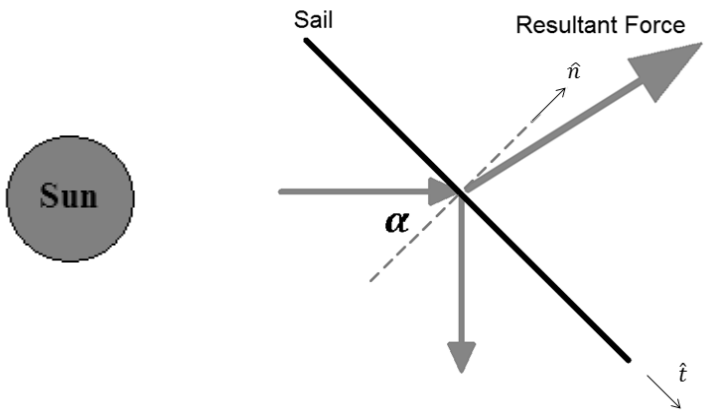

Figure 2.3: Resultant force from specular and diffuse solar radiation

This model accounts for both diffuse and specular reflection, but will only produce a moment about the center of mass if it is offset from the center of pressure of the sail. It does not account for moments created due to sail billowing, wrinkling of the sail material, and deformable-body effects.

An alternative force and moments model that was derived experimentally for the Sunjammer mission was provided by L'Garde which shows a significantly greater restoring moment at increasing sun-incidence angle than the flat-plate model. The moments generated by the two models are compared using Sunjammer's sail area [13] with a CM-CP offset of 10cm (the expected offset for Sunjammer) and the following value for the coefficient of specular reflection, $\sigma_{rs} = 0.827$ as suggested in [14].

While both models produce a restoring moment due to the center of mass being located between the center of pressure and the Sun, the magnitude of the restoring moment in the experimental model is significantly larger. At a sun-incidence angle of 40° the experimental model predicts a restoring moment of -1.66 milli-Newtons compared with the moment generated by the flat plate model of -0.09 milli-Newtons. The order of magnitude difference can have a significant effect on the attitude control scheme. A

control system developed in light of the flat-plate model combined with a CM-CP offset may not be well suited to control a solar sail that experiences much larger disturbance torques resulting from non-flat-plate behavior.

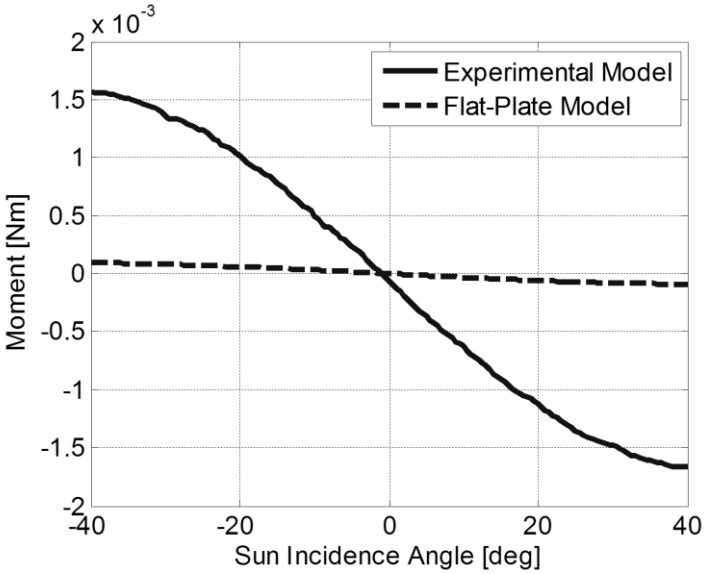

Figure 2.4: Experimental vs. flat-plate model moments

The experimental solar radiation pressure forces and moments are tabulated for both the main sail and the control vanes as a function of the projection of the surface-sun vector onto the plane of the sail and vanes. The tables are then accessed using the system's knowledge of the main sail's orientation relative to the Sun and orientation of each vane relative to the main sail.

The moments and forces obtained from these tables are then combined to calculate the overall moment produced by the sail and vanes about the system's center of mass as shown in Equation 2.5.

$$\boldsymbol{M}_{app} = \boldsymbol{M}_{sail} + \sum_{i=1}^{4} \boldsymbol{M}_{vane_i} - \boldsymbol{r}_{CM} \times \left(\boldsymbol{F}_{sail} + \sum_{i=1}^{4} \boldsymbol{F}_{vane_i}\right) \quad (2.5)$$

2.4 Passive Stability

The controller takes advantage of passive stability about the x- and y- axes. The concept of solar sail passive stability has been discussed in [11] and is shown in Figure 2.5 using a 2-dimensional representation of the sail.

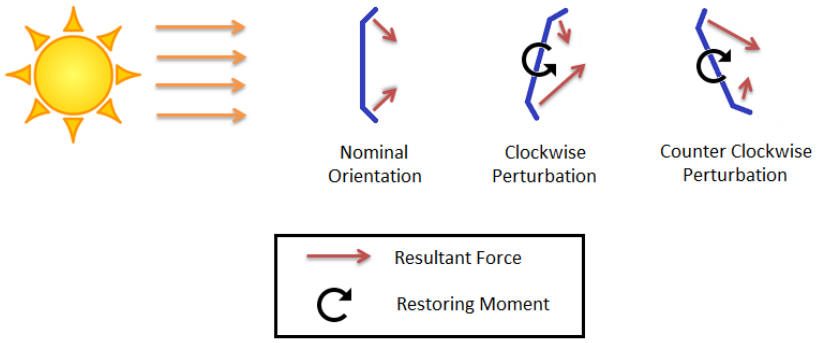

Figure 2.5: Passive stability in a Sun-pointing orientation

The sail in Figure 2.5 has the top and bottom vanes canted away from the Sun to generate passive stability. In the center diagram, the sail has been rotated counter-clockwise due to some perturbation. This motion away from the Sun-pointing orientation results in a greater force applied on the top vane and a smaller force applied to the bottom vane, which creates a restoring moment. Any motion away from a Sun-pointing orientation causes a restoring moment and maintains the sail in the Sun-pointing orientation when both vanes are canted away from the Sun by an equal amount. As discussed above, this restoring moment also exists due to an imperfect sail and the CM-CP offset; however, cant angles can be designed to determine the magnitude of the restoring torque of the sail in this orientation. Furthermore, the same concept can be applied to other orientations relative to the Sun, which require opposing vanes to have different cant angles.

2.5 Control Law and Moment Allocation

The control law is comprised from a standard proportional-derivative algorithm and a term that accounts for the moment bias estimate as shown in Equation 2.6.

$$\boldsymbol{M}_{des} = -K\hat{\boldsymbol{q}}_{\boldsymbol{Err}} - C\hat{\boldsymbol{\omega}}_{Err} + \hat{\boldsymbol{M}}_{bias} \tag{2.6}$$

where K and C are pre-determined gain matrices, $\hat{\boldsymbol{q}}_{\boldsymbol{Err}}$ is the vector component of the error quaternion, $\hat{\boldsymbol{\omega}}_{Err}$ is the error in angular velocity, and $\hat{\boldsymbol{M}}_{bias}$ is the estimate of the moment bias in the current orientation.

If the desired moment calculated by the controller is above the threshold for the use of the trim angles, the commanded moment is allocated to the control vanes such that each axis is controlled independently. Moment around the sail's x-axis is generated by canting vanes 2 and 4 orthogonal to each vane's rotational axis. Similarly, moment around the sail's y-axis is generating by canting vanes 1 and 3 orthogonal to each vane's rotational axis. To generate moment about the roll, or z-axis, all 4 vanes are twirled about the vane's x-axis by the same angle. The motion of the vanes is assumes instantaneous since they can be moved into any orientation in less than a minute whereas large attitude maneuvers take place in a matter of hours.

Generating the required moment about the x- and y- axis can be seen to have an infinite set of solutions due to the two degrees of freedom capable of generating this moment. The process of allocating moment is thus performed by first rotating the appropriate vane towards a more Sun-facing orientation where the appropriate vane is determined through Equation 2.7 (using the y-axis as an example):

$$Controlled\ vane = \begin{cases} 1, & M_{y_{des}} - M_{y_{cur}} \geq \epsilon_{d_y} \\ 3, & M_{y_{des}} - M_{y_{cur}} \leq -\epsilon_{d_y} \\ none, & -\epsilon_{d_y} < M_{y_{des}} - M_{y_{cur}} < \epsilon_{d_y} \end{cases} \tag{2.7}$$

where $M_{y_{des}}$ is the desired moment about the y-axis as calculated by the control law, $M_{y_{cur}}$ is the current moment generated by the sail and the vanes, and ϵ_{d_y} is the threshold used to avoid excessive motion of the vanes. If the desired moment cannot be obtained by rotating the appropriate vane toward the Sun, the opposite vane is rotated away from the Sun. The moment distribution control logic for the y-axis is presented in Figure 2.6. Each green diamond represents a decision point in the logic.

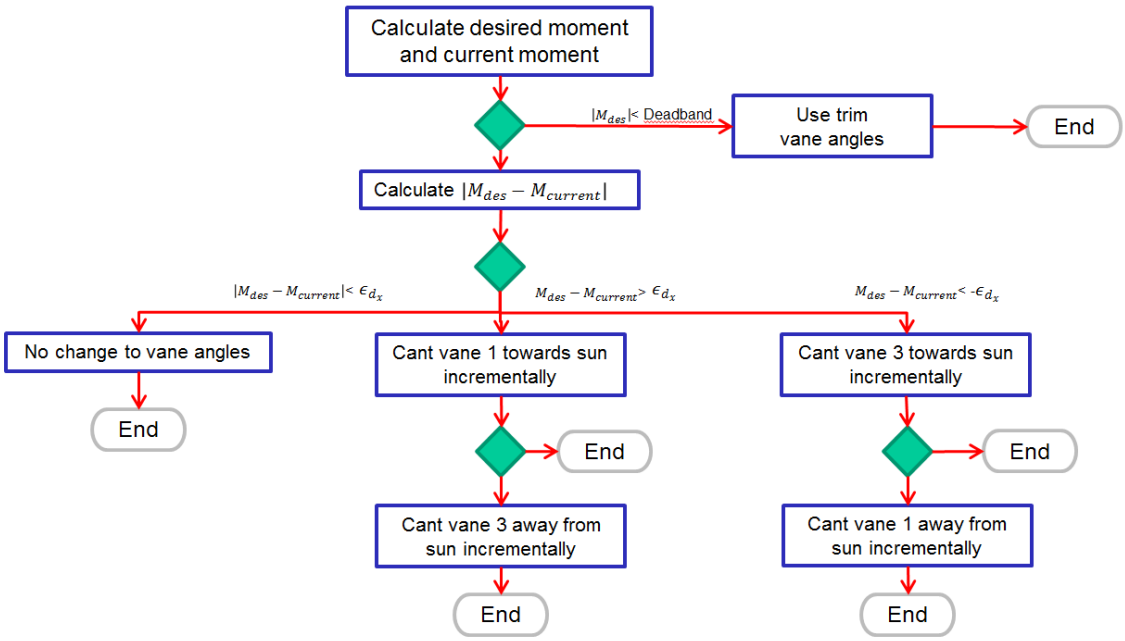

Figure 2.6: Moment allocation scheme for generating moments about the y-axis

2.6 Trim Angles and their Autonomous Adjustment

In order to counteract the restoring moment generated by the sail while minimizing the required motion in the vanes, the control scheme utilizes trim angles for the vanes. In a trimmed orientation, the net moment acting on the center of mass is zero. A trim table is generated a priori based on the model of the forces and moments generated on the sail and vanes at each orientation of the main sail relative to the Sun. Since the trim orientation should induce no moment on the sail, this orientation of the

vanes is used whenever the desired moment calculated by the control law is below a pre-determined threshold.

Utilizing trim angles allows the PD controller to operate about an equilibrium position. This removes the effects of the restoring moments caused by the CM-CP offset, the imperfect optical properties of the sail, and the deviation of the true sail shape from that of a flat plate under the effect of the solar radiation pressure. However, populating an a priori trim table requires a force and moment model sufficiently accurate to determine the equilibrium position of the vanes. Inaccuracies of the moment and force models will result in excessive actuation to counteract the unmodeled moment bias. The source of these inaccuracies include static and dynamic bending and twisting of the sail and vane booms, deviation of the observed and modelled shape effects from those effects on orbit, and other simplifications made in the modeling process. Furthermore, on-orbit faults such as part of the sail tearing or not deploying correctly and decay of the sail's optical properties can result in a significant difference between the calculated trim angles and those needed to trim the sail on orbit.

Therefore, an on-board adjustment of the trim table is introduced to handle these unmodeled or poorly modeled effects. The adjustment is made by maintaining a running average of the motion of the control vanes. In the presence of an unaccounted moment, the PD controller will continuously adjust the vanes. This is a result of the controller commanding the vanes to move between the moment necessary to overcome the bias and the trim orientation of the vanes. When the trim orientation of the vanes is significantly different than the actual trim condition, the resulting motion of the vanes will experience jitter. The control scheme monitors the motion of the vanes and uses that motion to estimate the true moment bias on the sail.

In order to avoid the task of updating the moment and force tables as a result of the observed attitude dynamics, the tables are assumed to be true and the observed deviations are lumped into the moment bias estimate accounted in the control law in Equation 6. Therefore, each time the trim angles are adjusted for a particular orientation due to control jitter of the vanes, the moment bias is updated. This creates a convergence upon the true moment bias which can be seen in the control history as a significant reduction in the high frequency jitter of the control vanes.

The pointing accuracy is improved using this technique since an unmodeled moment bias creates a bias in the attitude error. By estimating the moment bias and accounting for it by adjusting the trim angles, the pointing error is significantly reduced.

The effects of the moment bias and its elimination through trim angle adjustment are seen in Figure 2.7 and Figure 2.7 where the bias adjustment begins at t = 15 hours. The unmodeled moment bias in this case is introduced through an error in the pre-calculated trim angle for vane #1 shown in the top section of Figure 2.7. The orientation of the vane that is appropriate for the desired attitude control is approximately -8°, however, the original commanded value was set to +4° for demonstration purposes. The vane is seen to move from the trim angle (when no moment is desired) to approximately -8° in order to exert the desired moment and then cycles back to the trim angle. The amplitude of the oscillations increases as the PD controller attempts to eliminate the pointing error. The attitude shown in Figure 2.8 is approaching the desired Sun-incidence angle of 35°, however, at the 15-hour mark it can be seen that a pointing bias exists that the controller will not be able to overcome. The oscillating motion of the vanes will continue and the pointing bias will persist without the adjustment of the trim angles.

After the observation period of 15 hours has concluded, the algorithm determines the new best estimate of the actual trim angle. Once the adjustment mode is enabled at

t = 15 hours, the vane angle quickly converges to the correct trim orientation based on the observed motion. This adjustment eliminates the moment bias in the controller and allows the sail to converge to the target orientation marked by the dashed line in Figure 2.8.

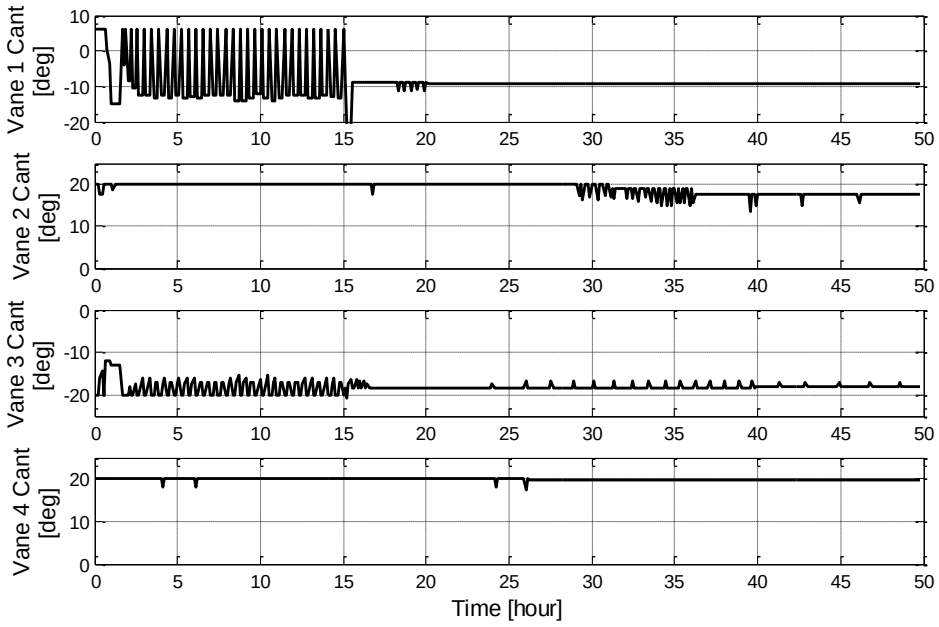

Figure 2.7: Effect of trim angle adjustment on control vane motion

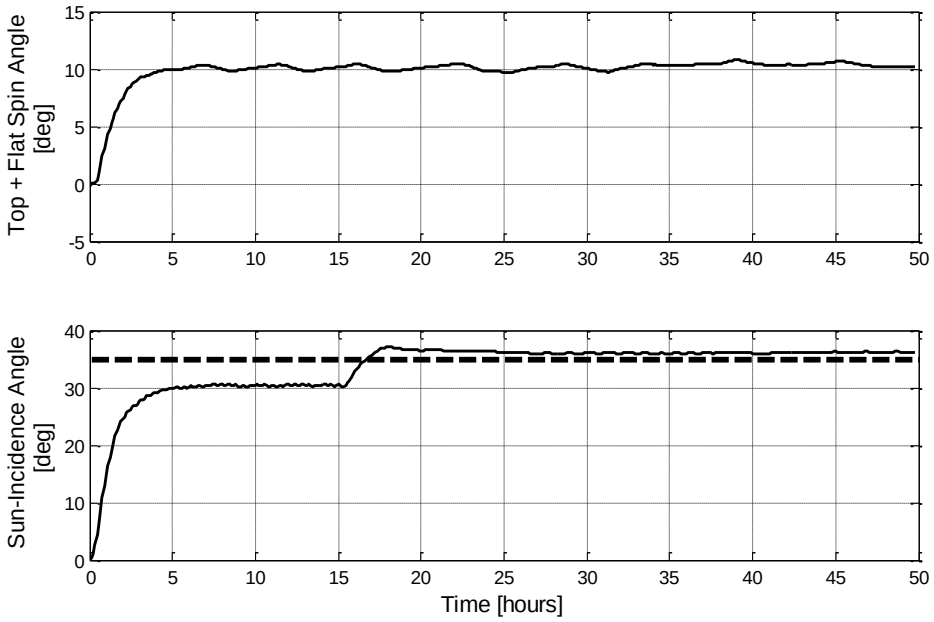

Figure 2.8: Effect of trim angle adjustment on pointing accuracy

2.7 EFFECT OF PASSIVE STABILITY

Utilizing passive stability in the control scheme as described in Section 2.4 allows for further reduction in the required control effort to maintain a given attitude. This effect is shown in Figure 2.9-2.12. Figure 2.9 shows the control effort required to reorient the sail by a Sun-incidence angle of 35 degrees. The reorientation is achieved in approximately five hours as seen by the convergence of the cant angles. However, after approximately 20 hours, the cant angles begin to oscillate and limit cycle about the desired orientation as the controller attempts to maintain the spacecraft's attitude.

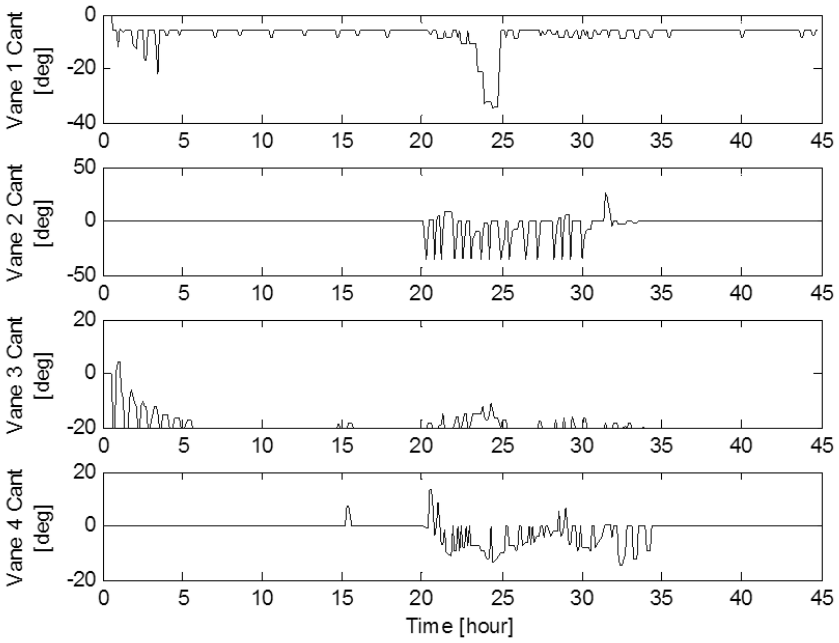

Figure 2.9: Control effort without passive stability

The instability in the controller vanes can be seen to affect the attitude of the spacecraft in Figure 2.10 where the Euler angles are seen to oscillate for approximately 10 hours before finally settling back down to the desired values.

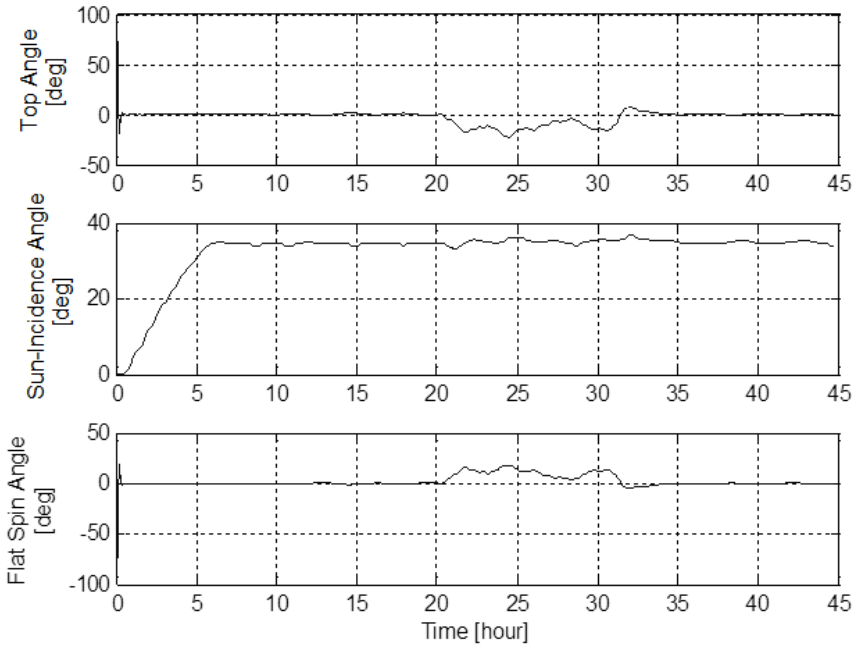

Figure 2.10: Spacecraft attitude without passive stability

With the addition of passive stability, the control effort is seen to lack the oscillation and maintains the desired vane orientation as seen in Figure 2.11. Only minor and infrequent adjustments are made to maintain the attitude of the spacecraft.

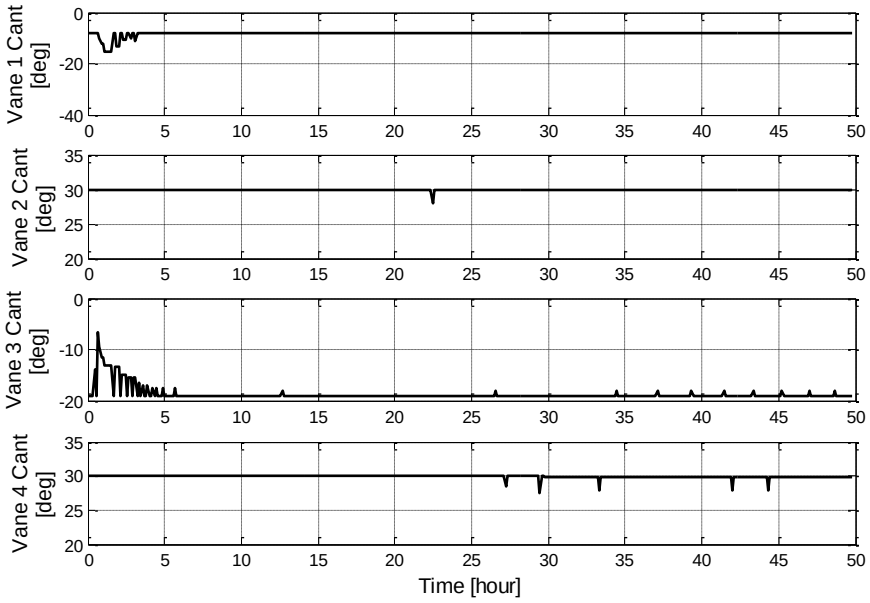

Figure 2.11: Control effort with passive stability

Consequently, the attitude, as shown in Figure 2.12 is also stable and does not vary greatly from the desired orientation once it converges.

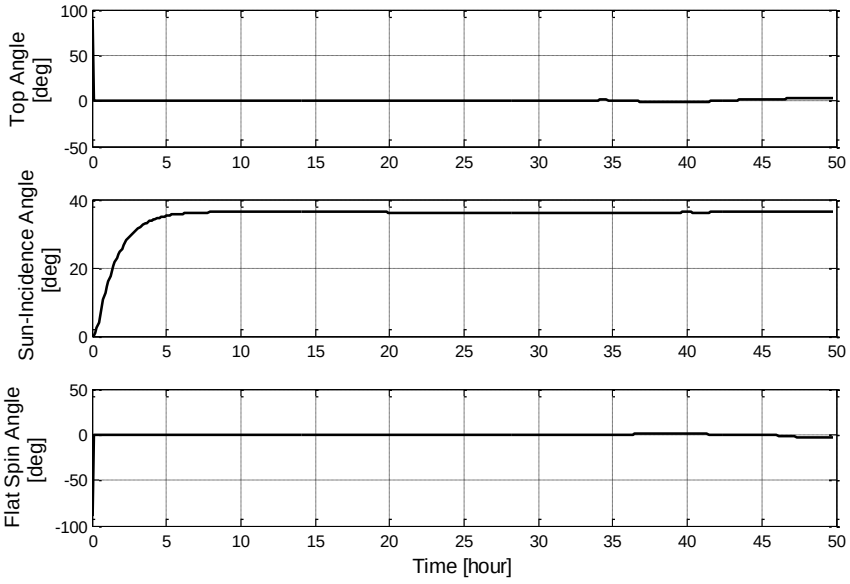

Figure 2.12: Attitude with passive stability

2.8 RESULTS

Investigation of the controller performance was conducted using 50-hour simulations of six baseline rest-to-rest maneuvers as detailed in Table 2.1. While the controller uses quaternions, these maneuvers are defined in terms of the three Euler angles introduced in Section 2.2: top angle θ_t, Sun-incidence angle θ_{SI}, and flat-spin angle θ_{FS}.

Table 2.1: Maneuvers used for controller performance testing

	Starting Orientation			Target Orientation		
Maneuver	θ_t	θ_{SI}	θ_{FS}	θ_t	θ_{SI}	θ_{FS}
1	0°	0°	0°	10°	35°	0°
2	0°	0°	0°	0°	5°	0°
3	0°	0°	0°	5°	0°	0°
4	0°	0°	0°	180°	45°	0°
5	0°	0°	0°	10°	-25°	0°
6	0°	0°	0°	0°	0°	0°

The orientations achieved by maneuvers 1 and 5 are those that maximize or nearly maximize the transverse force in the orbital velocity direction or against it. These orientations would typically be used to increase or decrease the spacecraft's heliocentric semi-major axis. Maneuvers 2, 3, and 4 were chosen to ensure good performance of the control scheme for both small and large slews, and the maneuver 6 serves to test how well the controller can maintain its attitude. It was deemed unnecessary to test changes in flat-spin angle, θ_{FS}, since it is typically held at zero [11].

These maneuvers were used to test both a baseline case where no disturbances are present and various other cases in which disturbances were introduced to the system. The disturbances provide a test of the robustness of the design in the presence of significant uncertainty. The sources of the disturbances are summarized in Table 2.2. The disturbance levels were used as an amplitude for a uniform error distribution for that

case. To illustrate using the first error source, 50 simulations were conducted for each one of the six maneuvers with the true moments of inertia uniformly distributed between -5% and +5% of the moments of inertia used by the attitude controller.

Table 2.2: Summary of disturbance sources and their magnitude

Disturbance	Disturbance levels used
Sailcraft moments of inertia	$\pm$5%, $\pm$10%, $\pm$20%, and $\pm$50% error in all moment of inertia matrix components
Forces and Moment coefficient scaling	$\pm$5%, $\pm$10%, $\pm$20%, and $\pm$30% Of all forces and moments
Moment bias	$5 \times 10^{-5}\ Nm$
	$1 \times 10^{-4}\ Nm$
	$2 \times 10^{-4}\ Nm$
	$3 \times 10^{-4}\ Nm$
Unmodeled sail static bend	Bending model error: 1°, 2°, 5°, 10°
Unmodeled sail static twist	Twisting model error: 1°, 2°, 5°, 10°
Center of mass offset	-0.5m to +1.1m in the z-axis

Each error level was simulated 50 times for each one of the six test maneuvers. Running the simulation 50 times for the same case allows for statistical characterization of the controller performance in light of the error level as well as its interaction with the Gaussian error introduced by the estimation algorithms used.

The moment bias levels were selected based on a percentage of the maximum moment produced by the vanes ($1x10^{-3}Nm$). Moment bias levels tested represent 5%, 10%, 20%, and 30% of this maximum moment. The center of mass (CM) z-axis offset, unlike the other disturbances, is known to the control system since this parameter can be analyzed on the ground. Thus, this performance test serves to indicate the control authority of the system as opposed to its handling of uncertainty. Uncertainty in z-axis CM-location is captured by the moment bias cases. Variations in the x- and y-coordinate

of the center of mass were not tested since these can usually be easily controlled in the structural design of the spacecraft.

The effect of each disturbance was checked independently of the other disturbances. One disturbance was introduced while the others were kept at zero in order to characterize its effect on the controller. Interactions between the various disturbances which may lead to poorer performance were not investigation in the scope of this work. One such interaction can be moments of inertia being larger than expected while the available control moment being smaller than expected. These two can act in unison to reduce the control authority of the controller.

Evaluation of the control performance was based on performance metrics relating to pointing error and control effort. A definition of these metrics as well as target values listed for each is shown in Table 2.3.

Table 2.3: Performance metrics for controller evaluation

Metric	Description	Target Value
Control Effort	Average cant motion of the four vanes over the last five hours of the simulation	$< 10°/hr$
	Average twirl motion of the four vanes over the last five hours of the simulation	$< 5°/hr$
Pointing Accuracy	Angle represented by the error quaternion between the desired orientation and the true (not estimated) attitude quaternion: $\tilde{\boldsymbol{q}}_{err} = \tilde{\boldsymbol{q}}_{des}^{-1} \otimes \tilde{\boldsymbol{q}}_{true}$ $\theta_{err} = 2\cos^{-1}\left(\tilde{\boldsymbol{q}}_{err}(4)\right)$	$< 2°$

The measure of centrality used for this analysis is the upper limit of a 95% confidence interval based on the median of the individual simulated runs. This upper limit is used for both the pointing accuracy and the control effort. The median is used as opposed to the mean so as to reduce the effects of outliers. Control effort is being used as

a performance metric since it is desired to reduce motions of the vanes that may induce dynamic modes in the sails that are not modeled in this analysis. The target values were chosen based on the results of running the controller without any induced disturbances, the baseline case.

A time history of 50 simulations that were run with maneuver 1 for the baseline case is shown in Figure 2.13.

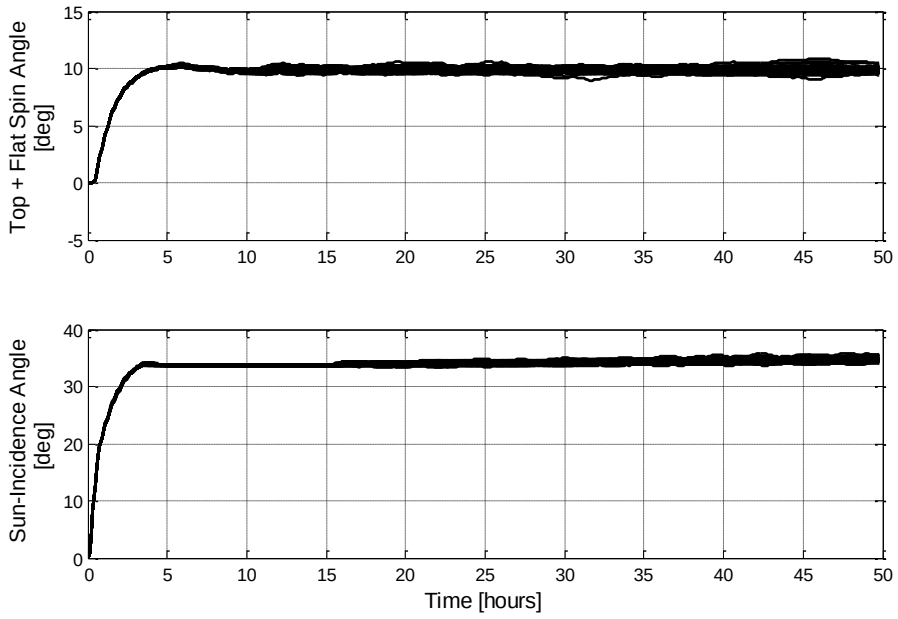

Figure 2.13: Sample attitude time history for the baseline case - maneuver 1

In this simulation sample, the spacecraft is shown to repeatedly achieve its target orientation in approximately three hours. The control effort time history for the same baseline maneuver can be seen in Figure 2.14 and Figure 2.15. The controller parameters were chosen such that there is minimal overshoot in further effort to reduce actuation.

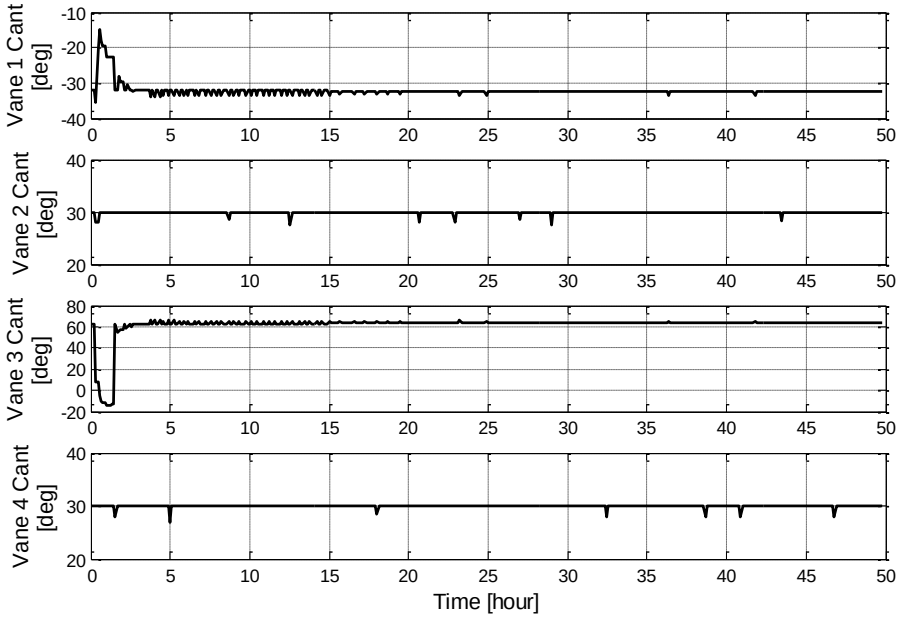

Figure 2.14: Cant angle time history for baseline case – maneuver 1

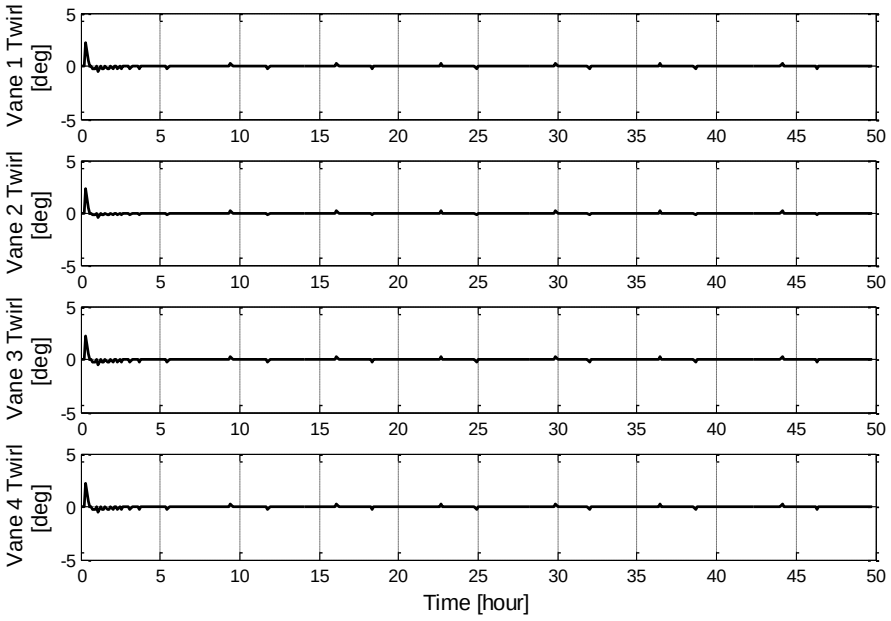

Figure 2.15: Twirl angle time history for baseline case – maneuver 1

Due to the lack of disturbances after the moment bias has been adjusted, minimal actuation in cant and twirl are required to maintain the attitude once the desired orientation has been reached.

The baseline performance of the controller without any disturbances added for all six maneuvers is presented in Table 2.4.

Table 2.4: Baseline performance metrics (no disturbances)

Maneuver	Upper Limit of 95% Confidence Interval		
	Pointing Error	Average Cant Motion	Average Twirl Motion
1	0.80°	$0.52°/hr$	$0.08°/hr$
2	0.55°	$0.46°/hr$	$0.10°/hr$
3	1.11°	$0.60°/hr$	$0.10°/hr$
4	0.60°	$0.56°/hr$	$0.05°/hr$
5	1.30°	$1.07°/hr$	$0.10°/hr$
6	1.08°	$0.64°/hr$	$0.10°/hr$

The errors shown in the baseline case above are a result of errors in the estimation algorithms, a restriction on the frequency of actuation, and the finite rotation step size allowed for each of the vanes. The deadband used in an effort to reduce the frequency of actuation was the major source of pointing error.

Table 2.5 summarizes the simulation results for the different cases outlined in Table 2.2. The pointing error column indicates the maximum of the upper limit of the 95% confidence level out of the six maneuvers run for each disturbance case. The disturbance level tolerated column indicates the amplitude of the disturbance for which the controller was still able to meet the target performance metrics.

Table 2.5: Simulation results

Disturbance Source	Disturbance Level Tolerated	Pointing Error	Max Cant Motion	Max Twirl Motion
Moment of inertia error	± 50%	1.70°	$1.53°/hr$	$0.10°/hr$
Scaling error in moment and force coefficients	± 30%	1.37°	$1.27°/hr$	$0.10°/hr$
Disturbance Moment Bias – x-axis	$3 \times 10^{-4}\ Nm$	1.58°	$2.99°/hr$	$0.46°/hr$
Disturbance Moment Bias – y-axis	$3 \times 10^{-4}\ Nm$	1.84°	$1.39°/hr$	$0.10°/hr$
Disturbance Moment Bias – z-axis	$2 \times 10^{-4}\ Nm$	2.13°	$7.96°/hr$	$0.60°/hr$
Unmodeled Vane Bend	± 5°	2.25°	$1.24°/hr$	$0.10°/hr$
Unmodeled Vane Twist	± 5°	2.16°	$4.78°/hr$	$0.76°/hr$
Varying z-axis CG Location	-0.5m to +1.1m	1.44°	$1.14°/hr$	$1.42°/hr$

The controller was able to meet its requirements in the presence of the maximal levels of disturbances in all the cases tested except for the highest level tested for unmodeled vane twisting, bending, and z-axis bias. Unmodeled twisting and bending of the vanes of 10° caused a maximum pointing error that exceeded 2°. It should be noted though, that typical vane twist commands are less than 1° in magnitude, in which case unmodeled twist of 10° may be overly conservative. Furthermore, the upper bound of the 95% confidence interval of the median pointing error at the 10° level while above the 2° threshold was still under 2.5° for all cases. The deviation was seen only in maneuver 2 for both unmodeled twirl and unmodeled bend while the deviation in the z-axis moment bias case was observed only in maneuver 5. The pointing error fell below 2° for unmodeled vane twist and cant of 5° and z-axis moment bias of $2 \times 10^{-4} Nm$.

The controller was able to meet the target value for average cant and twirl motion for all tested cases. Twirl motion was kept to less than 1° per hour for almost all cases.

Cant motion was kept to less than 3° per hour in most cases with a notable exception being moment bias in the z-axis.

2.9 Conclusion

A propellantless solar sail attitude control scheme using actuating control vanes has been developed. The technique was demonstrated by analysis using the Sunjammer mission design as an example application. By performing on-orbit adjustment of the a priori trim angles, the controller was shown to estimate and overcome moment biases arising from a nonplanar solar sail along with biases arising from various other sources. Estimating the moment bias and using this estimate in the control law greatly improved the performance in the presence of these disturbances. Incorporation of a more realistic sail model in conjunction with a sensitivity analysis increases the robustness of the algorithm and allows for its use in real solar sailing missions. The results show that the controller is able to meet the target pointing error of less than 2° over a wide variety of disturbance conditions and amplitudes with an acceptable level of actuation.

Chapter 3
Real time Computation of Safe Input Sequences for UAVs

Fixed-wing unmanned aerial vehicles (UAVs) typically operate in disturbance-rich environments, and their dynamics are governed by high-dimensional, coupled non-linear equations of motion. Devising a real time flight envelope protection system for UAVs somehow requires simplifications of the dynamical model. Indeed, the reachability analysis of a system with a large number of degrees of freedom cannot be accomplished in a reasonable amount of time with currently available embedded computers, even for very short time horizons. In this chapter it is shown that through offline simulation of the reachable states and a compact representation of the future safe-input set, a fixed-wing UAV is statistically ensured to remain within its pre-defined safe region. This is achieved over a specified time horizon, while maximizing the authority of the controller (human or autopilot)[2]. This future safe-input set is here characterized by a hyper-ellipsoid. Machine learning is used to allow the system to learn the safe input sequence set as a function of its current state, and to then use this knowledge to project potentially unsafe input signals inside the safe input sequence set. This procedure is computationally very efficient, and can be applied to human-in-the-loop and autonomous control of unmanned aerial vehicles.

3.1 INTRODUCTION

The operation of unmanned aerial vehicles (UAVs) in demanding environments can significantly increase their utility and viability as replacements of manned platforms which can inherently tolerate less risk. Among UAVs, fixed-wing UAVs are unmanned

[2] The work presented in this chapter has been published in a peer-reviewed journal. Analysis and write-up was performed by Ofer Eldad with supervision by Christian Claudel.

- Eldad, O., and Claudel, C., "Real-time Computation of Safe Input Sequences for UAVs," *Journal of Guidance, Control, and Navigation,* July 2016.

airplanes for which lift is primarily generated by a wing. Though rotorcraft UAVs are easier to fly and operate than fixed-wing UAVs, the latter typically have a much longer endurance, autonomy, and top speed. Unfortunately, fixed-wing UAVs are also prone to catastrophic aerodynamic events: flutter (overspeed), aerodynamic stalls, spins or high wing forces can cause the loss of control or structural damage to an UAV. These phenomena are typically avoided by operating the UAV in a narrow flight envelope in relatively good weather, though this also greatly reduces the operational capabilities of UAVs. Increasing the robustness of the UAVs' control system (whether piloted by a human or an autopilot) in light of unknown disturbances allows for a higher mission success rate and can push their flight envelope into previously unattainable regions, increasing the reliability of the system in severe weather conditions or in low-speed loiter missions. Verifying the robustness of the control system to disturbances is also an important part of this process.

Reachability analysis can enable a UAV controller to be more robust to disturbances, and to operate the UAV in a much larger flight domain. The question that is addressed in this chapter is the following: does the state of the system remain in a safe region over some time horizon given uncertain knowledge of the current state, and given bounded disturbances, and can the answer to this question be determined in real time on an embedded platform? The exact computation of the reachable set in high dimensional systems is very difficult and usually intractable in real time, even for some linear systems [15]. Commonly available methodologies for reachability analysis are either intended for linear / affine systems [16, 17, 18, 19], switched/hybrid linear systems [20, 21, 22], or non-linear systems with a low dimensionality, typically 3 or 4. In the latter case, the computational time is exponential in the dimensionality of the system, which prevents the reachable set of the high dimensional dynamical system that describes an UAV from

being computed in real time. The reachability methods outlined earlier typically rely on dynamic programming, or on level set methods [23, 24], and have additionally incorporated concepts from viability theory [25, 26, 27]. An approximate approach for solving the reachability / viability problem using a neural network approach has been proposed in [28].

Forward simulations can also be used to compute the reachable set of a system, though a considerable number of simulation runs is required to accurately characterize the reachable set [29, 30]. These simulations require too many computational resources to be used in real time on an embedded hardware platform.

The equations of motion of a fixed wing UAV are non-linear, coupled, and of degree 12 in their usual state-space formulation [31]. Therefore, most reachability analysis tools are ill-suited to these particular systems [32]. This is especially true when considering performing this analysis in a way that is sufficiently accurate and fast as to be used as part of an active real-time control system for the UAV. Linearization schemes have been proposed in the literature and are typically approached by either separating the lateral-directional and longitudinal degrees of freedom and considering them separately, or by linearizing about various equilibrium points of interest [31, 33]. However, both methodologies do not allow for analysis of the complete coupled dynamics over the full range of state values and of the possible disturbances or maneuvers across the flight envelope of the UAV.

Once a reachable set has been computed, it must be represented in a compact manner, given the limited computational resources available. Diverse representations can be found in the literature, including polyhedra obtained from convex hull computations [34, 35, 36], oriented rectangular hulls [37], hyper-rectangles [38], and ellipsoids [15, 39].

Machine learning has been employed in various aerospace control applications in literature. Applications include trajectory prediction [40], low level control [41], navigation [42], and incorporation with classical control methodologies [43, 44]. Machine learning has recently been proposed in the context of reachability analysis of dynamical systems. Machine learning was shown in [32] to provide four orders of magnitude improvement in computation time when classifying a cost-limited reachable set. The downside of using machine learning is that theoretical correctness guarantees can no longer be made and that many learning input sets are needed to cover the entire domain of interest. In safety-critical applications, this concern can be alleviated by adding a buffer region and employing statistical measures of safety that allow the user to judiciously select his or her desired level of risk.

This chapter proposes a new method for determining the set of inputs sequences over a finite time horizon that will keep the trajectory of the aircraft in a safe region as defined by multiple state constraints. Trajectories are simulated offline over a finite-time horizon using a range of possible input and disturbance signals to determine the set of safe inputs. This set is then approximated by an ellipsoid. Note that while ellipsoidal approximation of the safe set is utilized, ellipsoidal calculus is not used in the reachability analysis, but rather is accomplished through simulated trajectories. The functional relationship between the initial state of the system, the disturbance set and the ellipsoidal approximation of the safe input sequence set is then learned offline using a technique known as supervised machine learning. The ellipsoidal approximation of the safe input sequence set can thus be established in real-time by computing the image of a vector by a given function. Offline computation allows for more accurate simulation of the complete non-linear coupled equations of motion, and does not pose a problem in practice, since the parameters of the dynamical model are time invariant.

The remainder of this chapter is organized as follows: Section 3.2 presents the dynamical model used to describe the fixed-wing UAV equations of motion along with the form of the input and disturbance signals used in the simulation. Section 3.3 provides an overview of the main algorithm discussed in the chapter. Section 3.4 discusses the method used to calculate the cost function, and how the cost function can be used to generate a set of points that define the safe input signals over the given time horizon. Section 3.5 defines an ellipsoidal approximation of this set. Section 3.6 introduces a machine learning framework to learn the functional relationship between the current state of the UAV, the set of disturbances, and the ellipsoidal approximation of the safe input sequence set. Section 3.7 discussed the experimental platform and Section 3.8 provides results for an example case and discusses some limitations of the proposed algorithm followed by concluding remarks.

3.2 Dynamical Model of a Fixed-Wing UAV

The aircraft is modeled as a six degree of freedom rigid body, as described in [31]. The equations of motion are written as twelve coupled non-linear differential equations in state-space form. The system states are,

$$\boldsymbol{X} = \begin{bmatrix} x \\ y \\ h \\ u \\ v \\ w \\ \phi \\ \theta \\ \psi \\ p \\ q \\ r \end{bmatrix} \tag{3.1}$$

where *x,y,h* represent an inertial position, *u, v,* and *w* are given in the body frame, attitude is defined by the three Euler angles ϕ, θ, ψ, where an inertial frame is rotated into the body frame using a 3-2-1 rotation sequence by angles $\psi, \theta,$ and ϕ respectively, and *p, q*, and *r* are instantaneous rotation rates, and thus not simple derivatives of the Euler angles. The Euler angle sequence as well as the rotation rate explanation is described in Appendix B. The inputs to the system come from the aircraft elevator, ailerons, motor, and rudder, and can be written as, $u = [\delta_a \quad \delta_e \quad \delta_t \quad \delta_r]^T$.

Using the above states and inputs, the equations of motion become:

$$\dot{\boldsymbol{X}} = f(\boldsymbol{X}, \boldsymbol{u}, \boldsymbol{d}) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{h} \\ \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}$$

$$= \begin{bmatrix} (cos\theta cos\psi)u + (sin\phi sin\theta cos\psi - cos\phi sin\psi)v + (cos\phi sin\theta cos\psi + sin\phi sin\psi)w \\ (cos\theta sin\psi)u + (sin\phi sin\theta sin\psi + cos\phi cos\psi)v + (cos\phi sin\theta sin\psi - sin\phi cos\psi)w \\ usin\theta - vsin\phi cos\theta - wcos\phi cos\theta \\ rv - qw - gsin\theta + \frac{\rho V_a^2 S}{2m}\left(C_X(\alpha) + \frac{C_{X_q}(\alpha)cq}{2V_a} + C_{X_{\delta_e}}(\alpha)\delta_e\right) + \frac{\rho S_{prop} c_{prop}}{2m}((k_{motor}\delta_t)^2 - V_a^2) \\ pw - ru + gcos\theta sin\phi + \frac{\rho V_a^2 S}{2m}\left(C_{Y_0} + C_{Y_\beta}\beta + \frac{C_{Y_p} bp}{2V_a} + \frac{C_{Y_r} br}{2V_a} + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r\right) \\ qu - pv + gcos\theta cos\phi + \frac{\rho V_a^2 S}{2m}\left(C_Z(\alpha) + \frac{C_{Z_q}(\alpha)cq}{2V_a} + C_{Z_{\delta_e}}(\alpha)\delta_e\right) \\ p + qsin\phi tan\theta + rcos\phi tan\theta \\ qcos\phi - rsin\phi \\ qsin\phi sec\theta + rcos\phi sec\theta \\ \Gamma_1 pq - \Gamma_2 qr + \frac{1}{2}\rho V_a^2 Sb\left(C_{p_0} + C_{p_\beta}\beta + \frac{C_{p_p} bp}{2V_a} + \frac{C_{p_r} br}{2V_a} + C_{p_{\delta_a}}\delta_a + C_{p_{\delta_r}}\delta_r\right) \\ \Gamma_5 pr - \Gamma_6(p^2 - r^2) + \frac{\rho V_a^2 Sc}{2J_y}\left(C_{m_0} + C_{m_\alpha}\alpha + \frac{C_{m_q} cq}{2V_a} + C_{m_{\delta_e}}\delta_e\right) \\ \Gamma_7 pq - \Gamma_1 qr + \frac{1}{2}\rho V_a^2 Sb\left(C_{r_0} + C_{r_\beta}\beta + \frac{C_{r_p} bp}{2V_a} + \frac{C_{r_r} br}{2V_a} + C_{r_{\delta_r}}\delta_r\right) \end{bmatrix} \tag{3.2}$$

where, $C_X(\alpha), C_{X_q}(\alpha), C_{X_{\delta_e}}(\alpha), C_Z(\alpha), C_{Z_q}(\alpha), C_{Z_{\delta_e}}(\alpha)$ are nonlinear functions of α, g is the acceleration due to gravity, and ρ is a slowly varying function of altitude and the rest are constant system parameters, see [31] for more detail and Appendix B for complete nomenclature of all terms in the equations of motion.

$$\alpha \equiv \tan^{-1}\left(\frac{w_r}{u_r}\right)$$

$$\beta \equiv \sin^{-1}\left(\frac{v_r}{V_a}\right)$$

$$V_a = |\boldsymbol{V_a}| = \sqrt{u_r^2 + v_r^2 + w_r^2}$$

where,

$$\begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} = \begin{bmatrix} u - u_w \\ v - v_w \\ w - w_w \end{bmatrix}$$

u, v, w, are the 3 components of the vehicle velocity in the body frame, and $u_w, v_w, and\ w_w$ are the components of the wind velocity. u_r, v_r, w_r are thus the velocities of the vehicle relative to the wind.

Equation 3.2 shows the coupling between the lateral-directional and longitudinal equations as well as various nonlinearities inherent to the system. Wind disturbances appear in the system dynamics through the angle of attack, angle of sideslip, and airspeed.

To obtain the reachable states through simulation, four separate input functions are created that are consistent with the physical performance of the actuators. The possible input signals are characterized by a finite number of values, the function being constructed by linearly increasing the value of the input between these points at a given rate to account for the slew rate or time constants of the actuators. In the present application, each input function is assumed to be characterized by two values describing the position of each output at two different points in time, the midpoint time and the final time. These values are randomly generated using a uniform distribution over the range of allowable values for each input. The input is then varied linearly from the initial input state to the midpoint value and again linearly from the midpoint value to the final value,

as shown in Figure 3.1. If the desired midpoint value is reached before the midpoint time, the input value remains constant until the midpoint time.

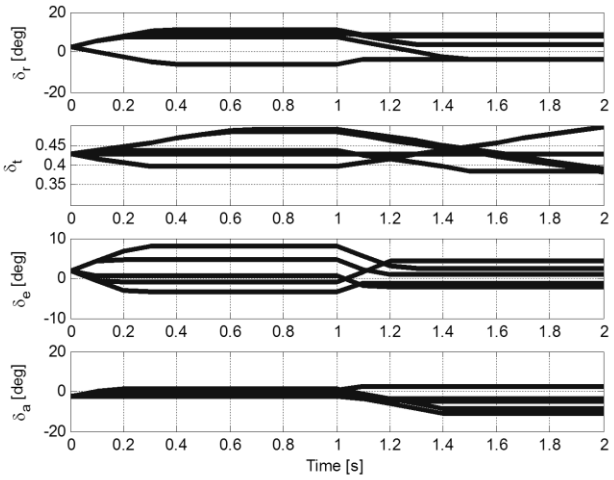

Figure 3.1: Simulated Input Signals over the Receding Time Horizon

This class of input signals is chosen in order to simulate a wide range inputs that covers the reachable states while keeping the problem numerically tractable, and being consistent with the physics of the actuators. By analyzing the trajectories obtained using the above input definition, it appears that sufficient variation in the final state is achieved to cover potential controller input.

Each combination of initial condition and input signal is simulated with a set of disturbances. These disturbances are chosen as ramp functions starting at zero and reaching a predetermined maximum level at the end of the time horizon, though more complex disturbance models could be thought of. At the end of the time horizon each axis' disturbance will either be zero, a predetermined maximum value, or a predetermined minimum value. With three components of wind, there are 27 possible disturbance signals simulated for each initial condition / input signal combination. For

clarification, an example of one of the 27 disturbances is shown in Figure 3.2. This is the case where the x-component of the disturbance ramps up to a maximum value, the y-component ramps down to a minimum value, while the z-component stays a constant 0.

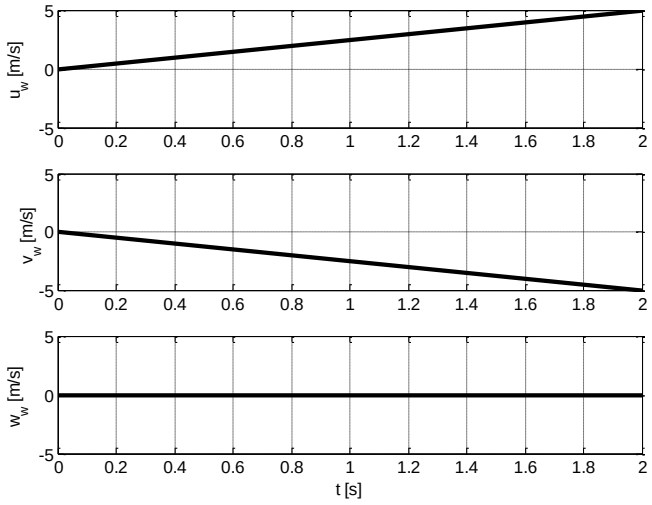

Figure 3.2: Example Disturbance

The time horizon chosen for these simulations is 2 seconds. This time horizon allows the control system of the aircraft sufficient time to adjust based on the computed ellipsoid. Using a relatively short time horizon also increases the validity of the approximations used for the allowable input signals and disturbances. In addition, the machine-learned function to approximate the safety ellipsoid can be applied over a wider range of initial conditions as the time horizon becomes shorter. Finally, UAVs are usually very small, and thus have very fast dynamics, which can be accurately captured over a short time horizon. Airframe dynamics that are longer than two seconds will be captured over time in the receding horizon approach.

Once the inputs and disturbances for each trajectory are defined, the equations of motion are integrated forward in time. Since these simulations are performed offline,

arbitrarily accurate integrators can be used without increasing the online computation time. In addition to the variation of inputs and disturbances, it is critical to include uncertainty in the initial condition and initial input value, to account for UAV state estimation uncertainty. This leads to an increasing number of simulations needed to accurately capture the reachable space.

3.3 ALGORITHM OVERVIEW

The algorithm at the core of this chapter can be separated into two main parts: Offline generation of a functional relation between the state and safety ellipsoid and online usage of the functional relation for evaluating the safety of candidate input signals in real-time, as shown in Figure 3.3. The reachable states are generated through multiple simulations with random inputs, and the inputs which correspond to a safe state are characterized by an ellipsoid. Approximately 5,000 simulations were required to sufficiently capture the reachable states.

3.3.1 Functional Relation Generation (Performed Offline)

The generation of a functional relation between the current states and inputs of a system and its approximated safe input sequence set is done through the following steps:

- Step 1: Define the current system state center $\boldsymbol{X_c}$ and current actuator state center, $\boldsymbol{u_c}$ along with their associated uncertainty.
- Step 2: Choose a particular initial system and actuator states, $\boldsymbol{X_0}, \boldsymbol{u_0}$ within the predefined range of uncertainty and set the number of simulations to run, p, using these initial conditions.
- Step 3: Define the set of allowable inputs $\boldsymbol{u}(\boldsymbol{t})$ for $t \in [t_0, T]$
- Step 4: Define the set of allowable disturbances $\boldsymbol{d}(\boldsymbol{t})$ for $t \in [t_0, T]$
- Step 5: Simulate the trajectory of the system with disturbance set to obtain

$$\boldsymbol{X}(t) = f(\boldsymbol{X_0}, \boldsymbol{u_0}, \boldsymbol{d}(\boldsymbol{t}), \boldsymbol{u}(\boldsymbol{t})) \text{ for } t \in [t_0, T]$$

- Step 6: Assign each trajectory a cost based on pre-defined constraints $J(\boldsymbol{X}(t))$
- Step 7: Define a set of points based on input sequences whose associated costs are below a threshold value.

$$\boldsymbol{G_i} = \left\{ \left\{ \boldsymbol{u}_i\left(\frac{T}{2}\right), \boldsymbol{u_i}(T) \right\} : J_i\big(\boldsymbol{X}(t)\big) < J_{max} \right\}, \boldsymbol{G_i} \in \mathbb{R}^8, i = 1,2, \dots p$$

- Step 8: Construct an ellipsoidal approximation, E, to this set of points and retain the center of the ellipsoid $\boldsymbol{c}(E)$, the eigenvalues of the shape matrix defining the ellipsoid $\boldsymbol{\lambda}(E)$, and their associated eigenvectors $\boldsymbol{v}(E)$
- Step 9: Repeat steps 2-8 to obtain multiple ellipsoidal approximations based on multiple initial conditions centered around $\boldsymbol{X_c}, \boldsymbol{u_c}$.
- Step 10: Use each of the initial states $\boldsymbol{X_0}, \boldsymbol{u_0}$ and their associated ellipsoidal approximation defined by $\boldsymbol{c}, \boldsymbol{\lambda}, \boldsymbol{v}$ as inputs and outputs respectively to obtain a functional relationship between them using a machine learning algorithm.

3.3.2 Online Usage

- Step 1: Obtain current system state estimate and actuator location, $\widehat{\boldsymbol{X}}_\boldsymbol{0}, \widehat{\boldsymbol{u}}_\boldsymbol{0}$
- Step 2: Use the previously generated functional relationship to obtain $\hat{\boldsymbol{c}}, \hat{\boldsymbol{\lambda}}$, and $\hat{\boldsymbol{v}}_i$ and thus an estimate of the safety ellipsoid, $\hat{E}$.
- Step 3: Shape the control signal by projecting the desired control as calculated by the onboard controller inside the safety ellipsoid.

The functional relationship used in the offline portion of the algorithm must be stored on board the aircraft so that it can be accessed during the online portion. The storage requirement is insignificant due to the compactness of the representation.

As discussed in detail in Section 3.6, this algorithm must be run for a given range of $\boldsymbol{X_c}$ and $\boldsymbol{u_c}$ for applicability over the entire state space. The last step of the online portion depends on the type of controller used to operate the UAV (human or autopilot), and on the objectives of the controller. For example, a human controller may only be interested in limiting the current values of the control, and thus projecting the set of desired inputs on the intersection of the ellipsoid and the set associated with current controls. The projection can be chosen to prioritize some inputs over some other inputs. For example prioritizing thrust control over elevator control would make the UAV pitch down when it is about to stall, while prioritizing the elevator over the thrust would increase the engine thrust in the same situation while keeping the same elevator position. For autopilots following a given objective, this last step can be identical to the human controller case. It could also be used as part of a Model Predictive Control (MPC) approach to define the constraints of the optimization problem over some time horizon. The process used in determining the safe input sets is outlined in Figure 3.3.

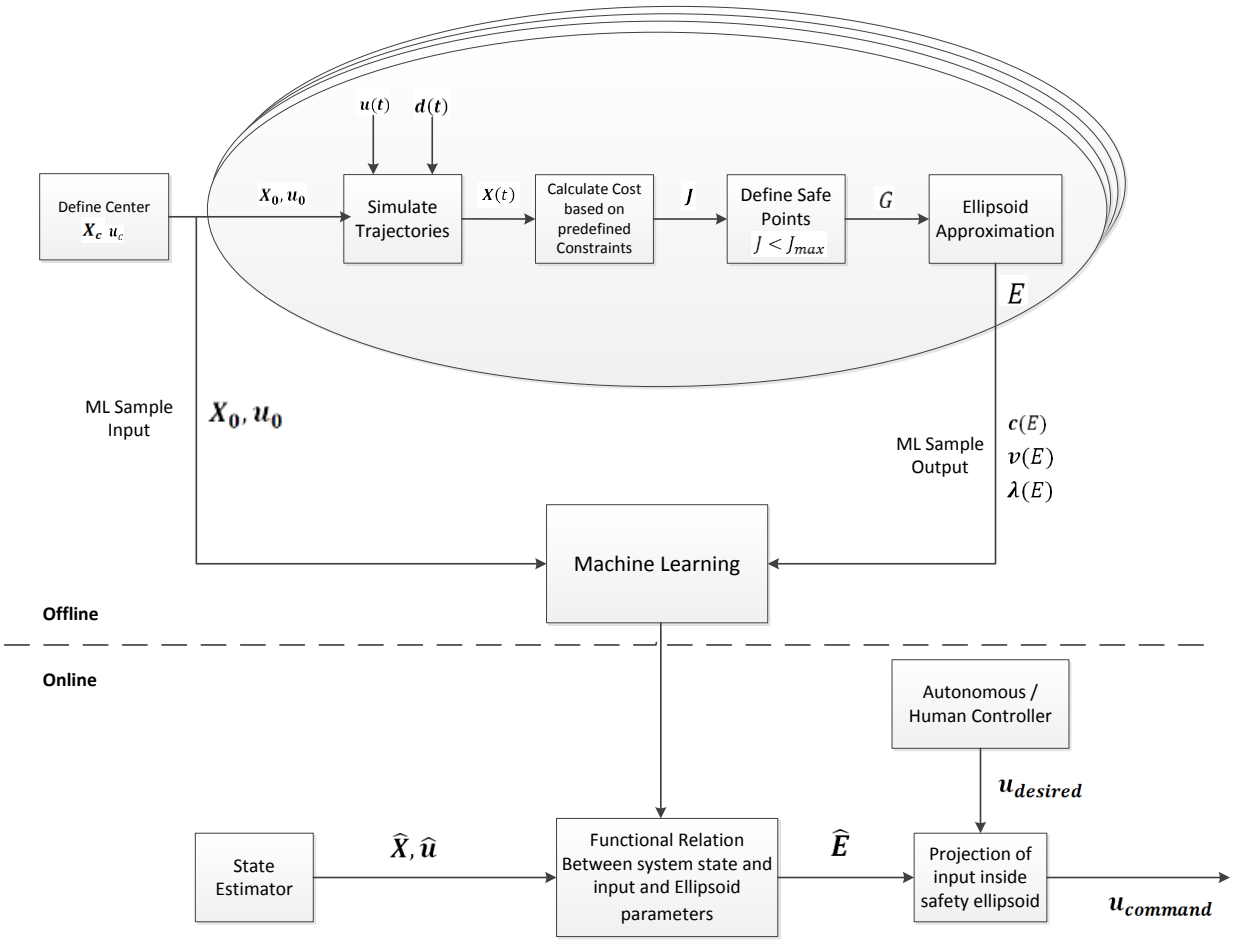

Figure 3.3: Algorithm overview

3.4 Cost Function Calculation

The simulated trajectories must be characterized as “safe” or “unsafe”, which is done through a cost function. The cost function for each trajectory is the sum of four different functions that involve the angle of attack and angle of sideslip (defining the risk of aerodynamic stall or spin), maximum allowable vertical acceleration (defining the risk of structural damage to the wings), minimum allowable height (defining the risk of crash to the ground), and maximum allowable velocity (defining the risk of aerodynamic flutter or structural damage to the wings), as shown in Equation 3.3.

$$J(\boldsymbol{X}(t)) =$$

$$w_1 J_{\alpha\beta_{max}}(\boldsymbol{\alpha}(t), \boldsymbol{\beta}(t), P_{\alpha\beta}) + w_2 J_{a_{z_{max}}}\left(a_{z_{max}}, \boldsymbol{a_z}(t)\right) + w_3 J_{V_{max}}(V_{max}, \boldsymbol{V}(t)) + w_4 J_{h_{min}}(h_{min}, \boldsymbol{h}(t)) \tag{3.3}$$

where w_1, w_2, w_3, w_4 are the weights assigned for each term of the cost function. All cost functions are evaluated based on the aircraft's states at pre-determined regular intervals during its trajectories. The cost function for the angle of attack and sideslip is based on the distance between the point defined by angle of attack and sideslip to the polygon $P_{\alpha\beta}$ shown in Figure 3.4 that defines the allowable envelope. The allowable envelope protects against stall at high and low angles of attack structural failure at high angles of sideslip and dynamic instabilities. This envelope was chosen based on an examination of envelopes of similar shape in [45]. When applying this algorithm, the user must judiciously modify this envelope to apply for their specific aircraft.

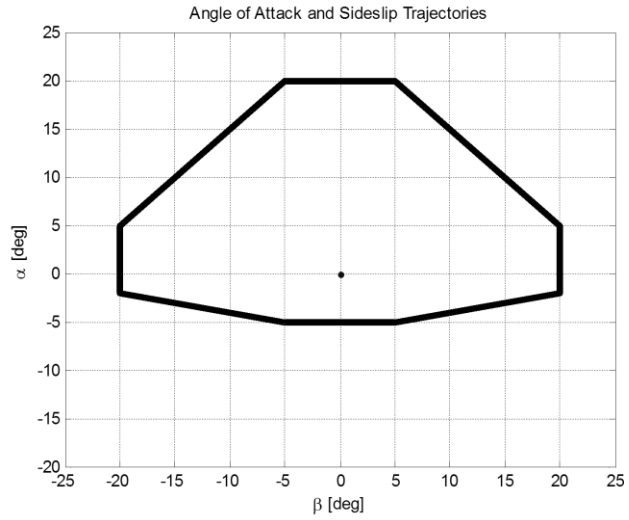

Figure 3.4: Polygon $P_{\alpha\beta}$ - Allowable envelope for angle of attack and sideslip

$$J_{\alpha\beta_{max}} = \max_{\{\boldsymbol{\alpha}(t),\boldsymbol{\beta}(t)\}} J_{\alpha\beta}(\alpha, \beta, P_{\alpha\beta});$$
$$J_{\alpha\beta}(\alpha, \beta, P_{\alpha\beta}) = \begin{cases} 0, & \text{if } \{\alpha, \beta\} \text{ is inside } P_{\alpha\beta} \\ \text{distance}(\{\alpha, \beta\}, P_{\alpha\beta}), & \text{if } \{\alpha, \beta\} \text{ is not inside } P_{\alpha\beta} \end{cases} \quad (3.4)$$

The distance function in Equation 3.4 is defined by the L2-norm. The three other cost functions are defined by logarithmic barrier functions as shown in Equation 3.5.

$$J_{a_{z_{max}}} = \max_{\boldsymbol{a}_z(t)} J_{a_z}(a_{z_{max}}, a_z);$$
$$J_{a_z}(a_{z_{max}}, a_z) = \begin{cases} -log(a_{z_{max}} - a_z), & a_z < a_{z_{max}} \\ B & a_z \geq a_{z_{max}} \end{cases}$$
$$J_{V_{max}} = \max_{\boldsymbol{V}(t)} J_V(V_{max}, V); \qquad J_V(V_{max}, V) = \begin{cases} -log(V_{max} - V), & V < V_{max} \\ B & V \geq V_{max} \end{cases}$$
$$J_{h_{min}} = \max_{\boldsymbol{h}(t)} J_{h_{min}}(h_{min}, h); \qquad J_{h_{min}}(h_{min}, h) = \begin{cases} -log(h - h_{min}), & h > h_{min} \\ B & h \leq h_{min} \end{cases} \quad (3.5)$$

where B is a very large constant.

The trajectories that correspond to a cost function smaller than a threshold cost J_{max} are determined to be safe over the chosen time horizon. The input signal associated with each safe trajectory, as defined by the value of each of the four inputs (aileron, elevator, motor thrust, and rudder) at the midpoint time and final time of the simulation, is captured as a point in 8-dimensional space.

This process is repeated such that a set of points, $\boldsymbol{G_i}$, is defined,

$$\boldsymbol{G_i} = \left\{ \left\{ u_i\left(\frac{T}{2}\right), u_i(T) \right\} : J_i(\boldsymbol{X}(t)) < J_{max} \right\}, \boldsymbol{G_i} \in \mathbb{R}^8, i = 1,2, \dots p \quad (3.6)$$

where p is the number of simulated trajectories.

All weights used in the cost function along with the maximum thresholds for each individual cost function are application-specific and depend on the user's approach to safety. They can also be modified for different phases of a mission where risk priorities and tolerances may change. For example, by increasing w_2 in relation to w_1 may disallow

trajectories that produce a larger acceleration in the z-direction and may allow a larger deviation from the envelope presented in Figure 3.4.

3.5 Ellipsoidal Approximation

The safe points found in the previous section define a region in the input signal set that is considered safe. This region has no specific topological property, and may not be convex or even not connected. To describe the region a convex hull approximation of this set can be used for example, though in this situation the number of coefficients required to define this region greatly increases with the number of safe points, which makes it unsuitable for use in a real-time system. Therefore, a more compact approximation is needed to keep the problem tractable in real-time. Two promising alternatives were considered, the oriented rectangular hull (ORH) presented in [37] and an ellipsoidal approximation [46]. Both alternatives require a fixed number of coefficients for their definition. The three representations, projected into the elevator input signal subspace for clarity are shown in Figure 3.5.

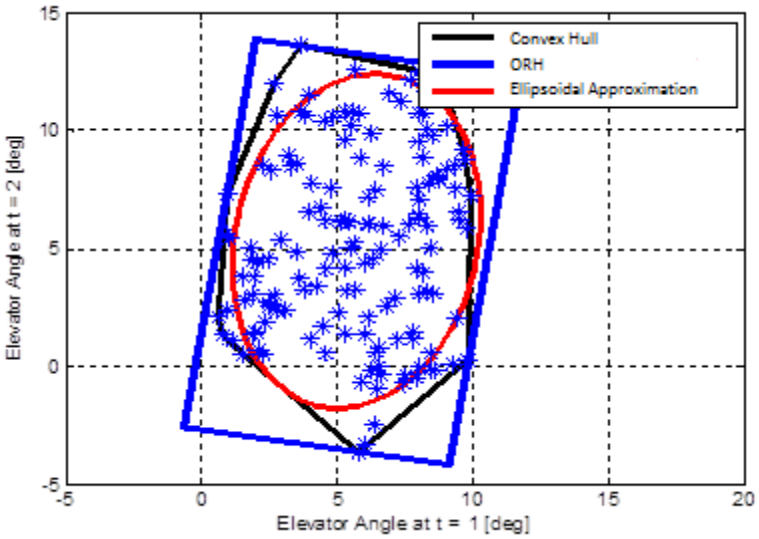

Figure 3.5: Comparison of convex hull, ORH, and ellipsoidal approximation

Ideally, the approximation would be such that every safe point is inside the convex shape and the volume of the convex shape is as close as possible to the volume of the convex hull. After comparing these two approaches, the ellipsoidal approximation is found to be superior. Metrics used for comparison include the percentage of safe points correctly captured inside the geometry and the overall volume of the approximation compared with the convex hull. A major drawback of the ORH is its over-approximation of the safe set which in this case leads to unsafe inputs to be deemed as safe.

Using the set of points in G, an ellipsoidal representation of the set is constructed. The ellipsoidal representation requires $n^2 + 2n$ coefficients, where n is the dimensionality of the input sequences (in this case, n=8). Note that the number of coefficients is independent of the number of simulations that use the same initial condition and thus the number of points that define the input signals. This allows one to use as many input signals as desired to fully capture the shape of the safe input sequence set.

The construction of the ellipsoid is accomplished by first determining the center of the ellipsoid,

$$\boldsymbol{c} = \frac{1}{p}\sum_{i=1}^{p} \boldsymbol{G_i} \tag{3.7}$$

All points are then expressed relative to the center:

$$\overline{\boldsymbol{G}}_{\boldsymbol{i}} = \boldsymbol{G_i} - \boldsymbol{c} \tag{3.8}$$

The points are then arranged in a matrix whose covariance is calculated. The singular value decomposition of the covariance matrix is then calculated in order to compute the eigenvalues and eigenvectors that define the shape and size of the ellipsoidal approximation using principal component analysis [47, 48].

$$\overline{\boldsymbol{G}} = \begin{bmatrix} \overline{\boldsymbol{G}}_{\boldsymbol{1}} & \cdots & \overline{\boldsymbol{G}}_{\boldsymbol{p}} \end{bmatrix} \tag{3.9}$$

$$Cov(\overline{\boldsymbol{G}}) = \frac{1}{p-1}\overline{\boldsymbol{G}}\overline{\boldsymbol{G}}^{\boldsymbol{T}} = U\Sigma V^T$$

where $U \in \mathbb{R}^{n\times n}$, $\Sigma \in \mathbb{R}^{n\times n}$ are used in the following definition of the ellipsoid,

$$E = \{\boldsymbol{x}|(\boldsymbol{x}-\boldsymbol{c})^T U\Sigma^2 U^T(\boldsymbol{x}-\boldsymbol{c}) \leq 1\} \tag{3.10}$$

noting that $U = V$ since the covariance matrix is symmetric by construction and positive definite when an ellipsoid exists. The direction of the principal axes of the ellipsoid is defined by the columns of U and their lengths by the diagonal entries in Σ. The lengths of the principal axes can be scaled in order to change the desired buffer distance from the undesirable input signals, those whose associated cost function is higher than J_{max}. This scaling can be tuned judiciously during implementation in which a conservative approach would reduce the size of the ellipsoid using a smaller scaling factor at the expense of limiting the allowable input signal and potentially causing a decrease in the performance of the control system through a reduction of the allowed flight domain.

Finding whether a point $\boldsymbol{x}$ is inside the ellipsoid is fairly straightforward by determining if the inequality in Equation 3.10 holds, and there exist algorithms to efficiently find distances to the ellipsoid and find the closest point on the ellipsoid to a given point using a quadratic minimization formulation [49]. In this formulation, the definition is of the ellipsoid matrix is transformed to,

$$E = \{\boldsymbol{x}|(\boldsymbol{x}-\boldsymbol{c})^T LL^T(\boldsymbol{x}-\boldsymbol{c}) \leq 1\} \tag{3.11}$$

$$E = \{\boldsymbol{y}|\boldsymbol{y}^T\boldsymbol{y} \leq 1\} \tag{3.12}$$

with,

$$\boldsymbol{y} \equiv L^T(\boldsymbol{x}-\boldsymbol{c}) \tag{3.13}$$

So that the minimizing the distance, $s \equiv \sqrt{(\boldsymbol{x}-\boldsymbol{k})^T(\boldsymbol{x}-\boldsymbol{k})}$, between a point **k** and a point $\boldsymbol{x}$ on the ellipse, becomes, after substitution for $\boldsymbol{x}$, a matter of minimizing the following quantity,

$$\mathrm{X} \equiv \frac{1}{2}\boldsymbol{y}^T L^{-1}L^{-T}\boldsymbol{y} + (\boldsymbol{c}-\boldsymbol{k})^T\boldsymbol{y} \tag{3.14}$$

subject to $\|\boldsymbol{y}\| \leq 1$, which is a convex quadratically constrained quadratic program due to the positive definite nature of the ellipsoid shape equation, when one exists. This optimization problem can be solved using the Lagrange Multiplier technique to augment the cost function as show in Equation 3.15.

$$\mathrm{X} \equiv \frac{1}{2}\boldsymbol{y}^{\boldsymbol{T}} L^{-1} L^{-T} \boldsymbol{y} + (\boldsymbol{c} - \boldsymbol{k})^T \boldsymbol{y} - \lambda(1 - \boldsymbol{y}^T \boldsymbol{y}) \tag{3.15}$$

setting $\frac{\partial X}{\partial \boldsymbol{y}} = 0$ gives rise to the optimality condition,

$$(L^{-1}L^{-T} + \lambda I)\boldsymbol{y} + (\boldsymbol{c} - \boldsymbol{k}) = 0 \tag{3.16}$$

$$\boldsymbol{y}^T \boldsymbol{y} = 1 \tag{3.17}$$

such that,

$$\boldsymbol{y} = -(L^{-1}L^{-T} + \lambda I)^{-1}(\boldsymbol{c} - \boldsymbol{k}) \tag{3.18}$$

$$\boldsymbol{y}^T \boldsymbol{y} = \|(L^{-1}L^{-T} + \lambda I)^{-1}(\boldsymbol{c} - \boldsymbol{k})\|^2 \tag{3.19}$$

The problem is thus reduced to optimizing the (scalar) Lagrange multiplier λ such that the difference between Equation 3.19 and 3.17 is minimized. This can be accomplished with a variety of optimization techniques. A simple one-dimensional search accomplishes this task with sufficient accuracy in under 2 milliseconds on MATLAB using an Intel i7-3610QM CPU running Windows 7 (commercial laptop) with $\boldsymbol{y} \in \mathbb{R}^8$. Therefore, such a system could be used in real time where the control cycle is approximately 10 Hz (100 ms).

3.6 Machine Learning Representation

The ellipsoid approximation for the safe set inputs discussed in the previous section is only valid for a given initial state and its associated uncertainty. In order to use this in real time, one needs to have this information for all possible initial states, which lie in a twelve dimensional space, which represents extreme memory requirements, far beyond the available memory in typical embedded systems. Therefore, a machine

learning framework is used to learn the relationship between the initial conditions and the ellipsoidal approximation of the safe input sequence set. The inputs of the machine learning include the center of the initial state $\boldsymbol{X_c} \in \mathbb{R}^{12}$ and input $\boldsymbol{u_c} \in \mathbb{R}^4$ while the output consists of the 80 coefficients needed to define the ellipsoid as discussed above. In order to have an accurate functional relationship between the inputs and outputs, a large number of simulations is required while varying the inputs to cover as much of the state space as possible. In the case discussed in this chapter, it was found that 5,000 simulations were sufficient to cover the state space. This determination is based on comparing the reachable set obtained using 5,000 simulations to test cases that involved a much larger number of simulations where no discernable difference was observed.

Machine learning is accomplished using MATLAB's neural network toolbox. The toolbox uses an algorithm based on minimization of a nonlinear function developed independently by Levenberg [50] and Marquardt [51] to update the weights applied in the neural network [52]. The number of nodes in the hidden layer of MATLAB's neural network affects the performance of the network. A small number of nodes may not capture the relationship between inputs and output fully, while a large number may cause the network to be overly sensitive. The accuracy of the machine-learned function is found to be highest when using 40 nodes in one hidden layer of the artificial neural network.

Ordering the machine learning inputs created in the data generation phase is trivial, however, the desired outputs must be carefully ordered to achieve the desired accuracy of the desired functional relationship. The eigenvectors and eigenvalues representing the shape and size of the ellipsoid's principal axes that are associated with a certain input signal must all be stored in the same location. Since the numerical procedure orders the eigenvectors according to their associated eigenvalues which are in turn stored according to an increasing or decreasing order, this order must be revised. This is

accomplished by selecting a baseline eigenvector for a given column in U, and then switching the order of eigenvectors (and associated eigenvalues) of the other ellipsoidal representations such that the eigenvector with the smallest angular deviation from the baseline eigenvector is stored in the same location. Following this procedure greatly improves the convergence of the learning step.

Given the highly nonlinear nature of the relation between inputs and outputs, the accuracy of the functional relation reduces outside of a narrow region. Outside of this range, new functional relations must be calculated such that the desired subset of the state space is covered by machine-learned functional relations. Online, the correct functional relation must be retrieved via a lookup-table in order to obtain the correct ellipsoidal approximation. It is found that the machine learning function is able to match the volume of the ellipsoidal approximation with 3% error over a range of ±15% from a baseline initial condition, where 15% refers to the total possible absolute deviation of each state or actuator input with respect to a nominal value. This translates to a difference of under 6% in the number of points that are within the machine learned ellipsoid vs. an ellipsoid that is constructed from direct simulation of the trajectories. Achieving this level of accuracy requires the use of 5000 simulations around each center. The large number of simulations accounts for the need to learn the effects of the variations and disturbances on the final ellipsoid size and shape. Further inspection reveals that certain states in the initial condition have a more dominant effect in terms of accuracy of the machine learned function. The practical range of inputs that can be used for each learned representation is shown in Table 3.1, for each state.

Table 3.1: Range of Applicability of Machine Learned Function

State	Range
x-axis velocity	±5 m/s
y-axis velocity	±4 m/s
z-axis velocity	±4 m/s
Roll angle	±8°
Pitch angle	±8°
Yaw angle	±8°
Roll Rate	$\pm 20°/s$
Pitch Rate	$\pm 10°/s$
Yaw Rate	$\pm 20°/s$

The range allows for accurate ellipsoids to be generated over a range of 10 degrees in angle of attack and angle of sideslip for typical flight conditions. Variation of the initial actuator positions or the initial position of the UAV has little effect on the accuracy of the machine-learned function and is thus not shown in the above table.

The time required to generate the safety ellipsoid coefficients using the machine-learned function, assign the output of the function to the appropriate vectors and matrices, and determine whether or not the desired control command is safe (or project this command onto the safe ellipsoid) is under a millisecond on MATLAB using an Intel i7-3610QM CPU running Windows 7 (commercial laptop), which is considerably faster than the dynamics of the system. Therefore, this approach is well suited for implementation as a real time flight envelope protection system. The time required for learning the functional relationship between inputs and safety ellipsoids is on the order of hours on the machine described above, and depends on the number of simulated trajectories and the size of the neural network used.

3.7 Experimental Platform

The online portion of this algorithm has yet to be flown and thus the results section that follows displays results from simulation. This section provides an example of a flight system and computer that can be used to implement the algorithm described in this chapter. The experimental platform consists in a Bormatec Maja UAV (shown in Figure 3.6) equipped with an Odroid XU-3, receiving air data from and sending data to a Pixhawk autopilot (both shown in Figure 3.7). Preliminary testing of the proposed algorithm on the Odroid XU-3 indicates that sufficient computational power is available to perform the desired operations in real time, at a rate of approximately 20 Hz. This rate compares favorably to the actuator time constants of about 0.2-0.3 seconds. The small form factor and low weight of the Odroid XU-3 result in little effect on the handling characteristics of even this relatively small platform.

Figure 3.6: Bormatec MAJA UAV during initial test flights

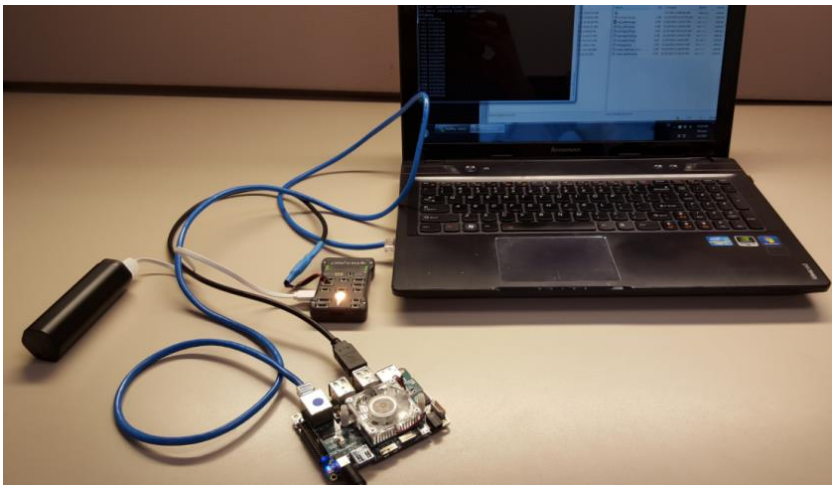

Figure 3.7: Hardkernel Odroid XU-3 (bottom) interfaced with laptop and Pixhawk autopilot (above Odroid)

In this implementation, the Odroid XU-3 receives data from the autopilot over a serial connection at a rate of 10 Hz (which is used to estimate the position, attitude, airspeed, angle of attack and sideslip of the UAV [53]) and sends commands to the Pixhawk to dynamically limit the minimum or maximum values sent to the UAV actuators in real time. The Pixhawk is itself connected to a Remote Control receiver, and relays human controller inputs to the Odroid computer, which modifies these inputs if necessary to allow the UAV to remain in the safe set. Possible ways of modifying the user inputs include projecting them onto the safe ellipsoid, which amounts to solving a low-dimensional Quadratic Program (QP) in real time. The function used to describe distances in the input space can be adjusted to give more weight (more priority) to specific inputs, for example to give priority to the current trajectory or to the current airspeed, depending on the objectives of the flight.

3.8 RESULTS AND LIMITATIONS

Usage of this method is demonstrated with an example scenario in which the UAV is descending towards the pre-set minimum height of 500 meters. At the start of the scenario the UAV is at a height of 510 meters and at a pitch angle of 15 degrees below the horizon with disturbances up to a maximum value of ±5 m/s in each axis taken into consideration. In this example the maximum z-acceleration was set to 50 m/s^2 and the maximum airspeed to 40 m/s and all weights of the cost function were set to 1.

The resulting safety ellipsoid for the future elevator commands is shown in Figure 3.8.

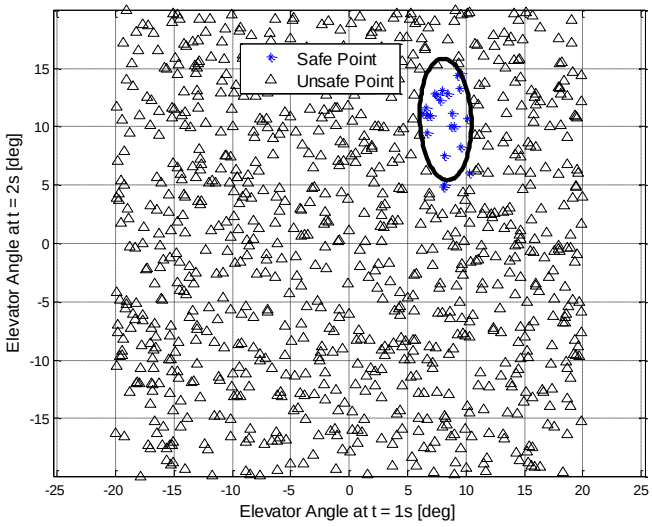

Figure 3.8: Safety ellipsoid for minimum height constraint – with disturbances

Since the ellipsoidal approximation is not a perfect representation of the safe input set, one point can be seen inside the ellipsoid that is not safe. The case of unsafe points inside the safety ellipsoid represents inputs that are deemed by the ellipsoidal approximation as safe, but will in fact result in a violation of one of the constraints.

The opposite case, where safe inputs sets are outside the safety ellipsoid is also seen in Figure 3.5, again due to the imperfection of the ellipsoidal approximation. This will result in the control supervisor to unnecessarily modify a proposed control command to make it safer even though the original command would not have resulted in a violation of the constraints. This may result in unnecessary degradation of the control system. Balancing between these two undesired effects of the approximation can thus be managed during the implementation of the system. This process will require simulating the scenarios of interest and adjusting the scaling of the ellipsoid to arrive at behavior that best satisfies the system's requirements.

The trajectory of the UAV is now simulated with a new randomly generated set of five inputs (along with 27 disturbances for each input signal), both belonging to the safety set and not. These inputs can be thought of as those generated by a human or computer controller in real time. The resulting height of the UAV is plotted in Figure 3.9 where trajectories are differentiated based on whether or not they are inside or outside the safety ellipsoid.

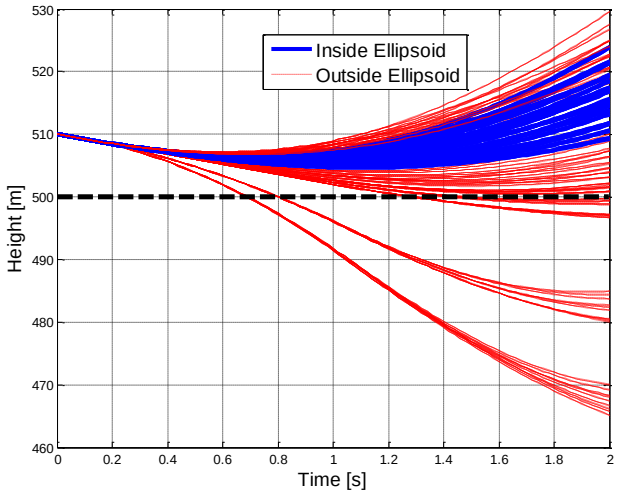

Figure 3.9: Simulation of UAV height

Figure 3.9 shows that no trajectories that were considered safe (being inside the safety ellipsoid) violated the minimum height constraint. The system states that comprise the three other cost sub-functions are shown in Figures 3.10-3.12. The envelope in Figure 3.4 is not presented since the value of the angle of sideslip is very near zero for this maneuver. Instead, just the angle of attack is plotted with its appropriate bounds.

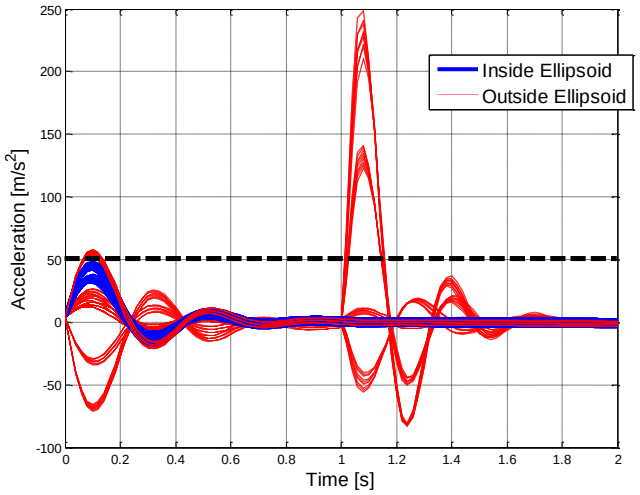

Figure 3.10: Simulation of UAV z-acceleration

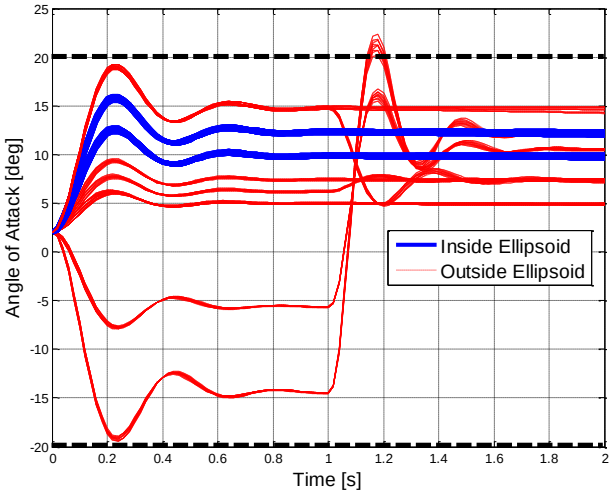

Figure 3.11: Simulation of UAV angle of attack

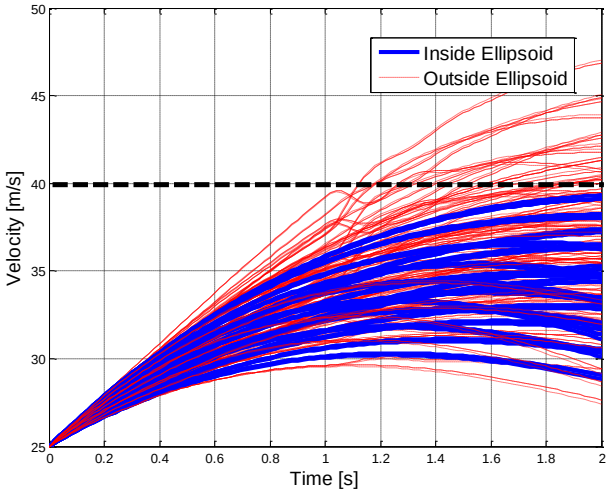

Figure 3.12: Simulation of UAV airspeed

The same scenario above is simulated without disturbances. Figure 3.13 shows that the safety ellipse now expands to include more elevator values as expected.

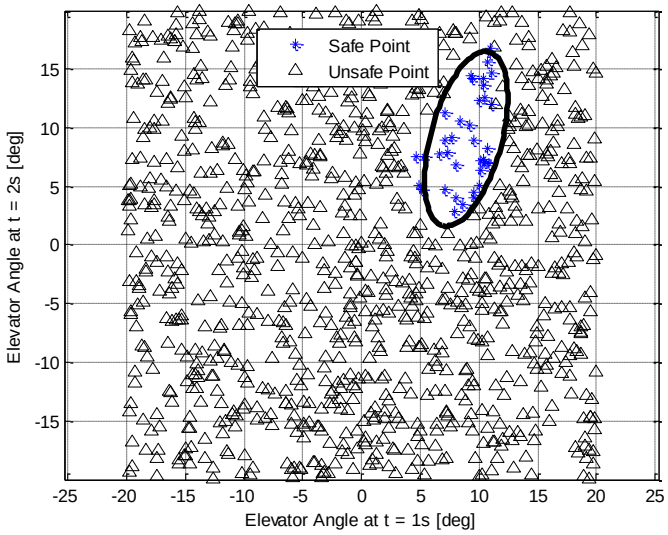

Figure 3.13: Safety ellipsoid for minimum height constraint – no disturbances

The system states that comprise the four cost sub-functions are shown in Figure 3.14.

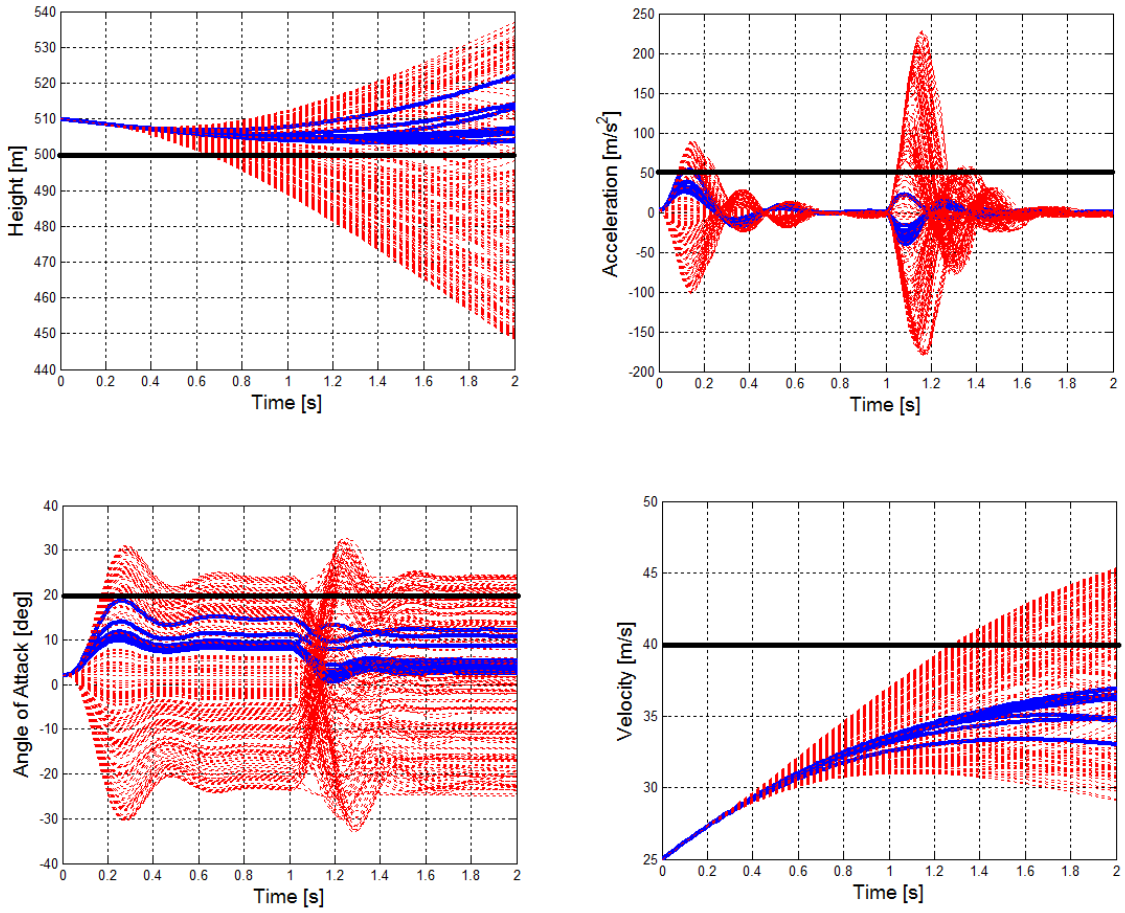

Figure 3.14: Safe and unsafe trajectories - no disturbances

The limiting cost sub-functions in this case are seen to be the maximal acceleration and angle of attack.

A limitation of approximating the future input-set as a single ellipsoid arises from situations in which the desired set is theoretically non-convex, or worse, non-connected. Scenarios in which maintaining the current control command is unsafe and the required command to reach a safe state is contained in unconnected regions of the future input set will result in safe input sets that are better represented by several ellipsoids, as is depicted in Figure 3.15. This figure illustrates a scenario involving a vertical dive by a UAV, in

which the maximum airspeed constraint becomes violated without an elevator input. A safe elevator input can be either positive or negative (resulting in positive or inverted flight), which is shown by two safe regions separated by an unsafe region.

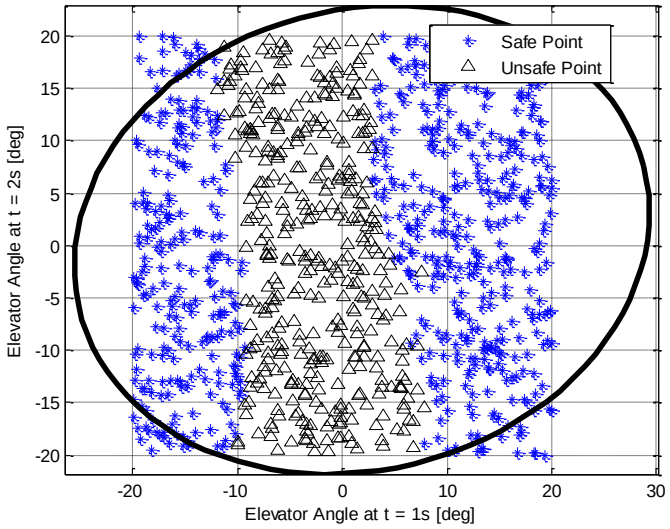

Figure 3.15: Example of non-connected safe input set

Other non-connected input sequences may arise depending on the choice of cost functions used. Splitting the safe sequences into multiple ellipsoids or alternative representations has not been explored in this work, however, the basic methodology would still hold true. The likelihood of encountering non-connects inputs sequences must be investigated for the system of interest.

3.8 CONCLUSIONS

This chapter describes a new methodology for incorporating simulation-based reachability analysis in determining the safe set of input signals over a finite time horizon. By representing the future safe input signals compactly in the form of an 8-dimensional ellipsoid and by utilizing machine learning to characterize this ellipsoid low-powered embedded system can readily access the output of these extensive offline

computations. The methodology enables real time systems to obtain the coefficients of the safety ellipsoid with little computational effort, well within the time required to generate a new control solution.

This methodology described in this chapter is particularly well suited for model predictive control. It enables the optimization of the control strategy while providing a statistical measure that the resulting optimal control sequence remains within the safety ellipsoid.

Chapter 4
Minimum-Time Attitude Control of Deformable Solar Sails with Model Uncertainty

This chapter develops a new algorithm for large-angle pitch maneuvers of deformable solar sails in minimum-time that avoids overshooting the target angle given a set of model uncertainties[3]. The chapter uses a simplified Euler-Bernoulli beam to model the sail's flexible booms. Model uncertainties include the flexural rigidity of the sail's booms, the effectiveness of the sail's reflectivity, and its moment of inertia about the pitch axis. The effect of each of these sources of uncertainty is investigated for a set of three sails of increasing size. The algorithm relies on trajectory-based reachability techniques introduced in the previous chapter to obtain a distribution of final states at the end of a large-angle maneuver that depend on the estimated model uncertainties. Using a tunable measure of statistical safety, the algorithm determines the required buffer angle to avoid overshooting the target attitude. Machine learning is used for reducing the parameter uncertainties based on a calibration maneuver. In addition, it is used to obtain a fast-access relationship between the current uncertainty in the model parameters and the required buffer angle.

4.1 Introduction

Solar sail attitude control has traditionally been approached using a rigid-body assumption [4, 5, 7, 8]. As discussed in the introduction, NASA's technological roadmap [3] for solar sails highlights the need for larger sails whose performance, measured by the

[3] The work presented in this chapter has been submitted for publication in a peer-reviewed journal. Analysis and write-up was performed by Ofer Eldad with supervision by Glenn Lightsey and Christian Claudel.

- Eldad, O., Lightsey, E.G., and Claudel, C., "Minimum-time attitude control of deformable solar sails with model uncertainty," *Journal of Spacecraft and Rockets,* Submitted July 2016.

ratio of area to mass, increases as well. Higher performance sails can achieve greater accelerations that allow them to obtain a wider range of orbits than could otherwise be reached. As sails get larger and lighter, the rigid-body assumption becomes less valid and the need for incorporation of flexibility effects increases.

One significant challenge in determining the flexibility of the sail is the inability to deploy and test an entire sail on Earth. The sail cannot support its own weight and even the first generation of sails is extremely large compared to current spacecraft. The solar sail designed for NASA's Sunjammer mission was a $35m \times 35m$ spacecraft whereas typical large satellites have a diameter of less than 5 meters and length of less than 10 meters for the main satellite bus (excluding solar panels). Ground vibrational testing of the entire sail system to determine its structural properties thus becomes unrealistic. This challenge will only increase as sails become larger and more flexible. It is also difficult to determine how the sail material will billow and wrinkle once it is deployed in space and thus the forces and moments generated by the sail are hard to predict.

Modeling the flexible characteristics of solar sails has been a subject of recent publications, much of which stems from Taleghani, Sleight, and Muheim's work [54, 55] from the Army Research Lab and NASA-Langley. In [55], parametric studies of square sails are performed using finite element analysis to examine the effects of sail size, stiffness, and sail membrane parameters on static sail deflection and natural frequencies. The baseline sail used in the study included a side length of 150m. This baseline sail's parameters were adopted by subsequent papers. The computational expense of running a finite element model has led to the development of various reduced models for solar sails. These reduced models use a 2-dimensional distributed parameter model [56], models that include foreshortening effects due to foreshortening deformation of sails and booms [57] and a model utilizing the assumed modes method [58, 59, 60]. Results from these

formulations compare favorably with finite element analyses. The models that were developed in these manuscripts assumed perfect knowledge of the sail's structural behavior. Importantly, every model of a standard square solar sail (without additional masses along the boom) that included the effects of structural damping showed that all vibrations are damped down to insignificant amplitudes with a time scale of a few seconds compared to maneuver times greater than 30 minutes.

In this chapter, an algorithm is proposed for performing large angle pitch maneuvers of a flexible square solar sail with a statistical measure of safety to ensure that the maneuver will not cause the sail to overshoot its target attitude. Overshoot is undesirable in conservative design of control systems. Solar sails are particularly averse to undesirably large angles away from the sun-pointing vector. The larger the angle between the sail-normal vector and the vector pointing from the sail to sun, the smaller the control authority of the standard square propellantless sail; thus the smaller its ability to reject disturbances. In addition, a state in which the sail-normal vector is flipped may be unrecoverable and can create a mission-ending scenario.

The model used in this chapter is a quasi-static Euler-Bernoulli beam in which only the static deflection of the beam is considered. Based on the results of previous research, the flexible dynamics of the system are ignored assuming they will be damped out quickly relative to the attitude dynamics of the vehicle. The computational complexity of using a finite-element model for on-board attitude control and the uncertainty in the flexible characteristics that is inherent to any model lends itself to the use of a simple model in which the use of an algorithm for handling this uncertainty is preferred over detailed modeling techniques.

This chapter proposes a way to handle inherent uncertainties in the flexible effects as well as other model uncertainties such as the sail effectiveness in reflecting solar

radiation and the sail's moment of inertia. Results are shown using three sail geometries of increasing size to highlight the increasing relative significance of flexural rigidity on solar sail attitude control. Sail 1 is similar to the design used in Chapter 2 of this dissertation and is thus similar to the Sunjammer spacecraft design. Sail 2 is identical to the baseline model presented in [2] for the purpose of comparison. Sail 3 is a very large and flexible sail used to demonstrate the effects of flexibility. The properties of these sails are given in Table 4.1

Table 4.1: Sail properties

Property	Sail 1	Sail 2	Sail 3
L_s Sail side length $[m]$	35	150	750
EI Flexural Rigidity $[N-m^2]$	6,500	35,000	158,000
J_{yy} Pitch MOI $[kg-m^2]$	6,000	55,000	687,500
η Sail effectiveness	1.80	1.80	1.80
A_v Control vane area $[m^2]$	15	32	80

Sail effectiveness for all sails was chosen based on specular and diffuse reflection coefficients, front and back sail emissivities, and front and back non-Lambertian coefficient given in [14]. An ideal sail would have sail effectiveness of 2. Flexural rigidity for Sail 2 was set to be identical to the baseline presented in [54]. Flexural rigidity for Sail 1 was chosen based on scaling with respect to sail side length, and flexural rigidity for Sail 3 was chosen in order to simulate a very flexible sail with significant deflection. The pitch moments of inertia are scaled with the length squared, but then decreased by a factor of two to account for higher performance sails in future

generations of sail technology as per NASA's technological roadmap [3]. The vane area for Sail 1 is based on the design presented in Chapter 2. Sail 2 and Sail 3 vane area is chosen such that the characteristic angular acceleration is similar for all three sails.

Section 4.2 describes a simplified flexible model that is used to model each sail. Section 4.3 utilizes the flexible model to develop a bang-bang controller that takes into account the varying maximal moment that can be applied as a function of the sail's angle relative to the sun. Minimum-time maneuvers for solar sails were addressed in [8], but used unrealizable control moment generation with tip-vanes and assumed perfect knowledge of system state and model parameters. In Section 4.4, the minimum-time controller developed in this chapter is then used with the assumed uncertainty in the system model to calculate the required buffer angle to ensure no overshoot by simulating multiple trajectories forward in time in Section 4.4. Machine learning is used in Section 4.5 to reduce the uncertainty in the key system model parameters and to derive a relation between these uncertainties and the desired maneuver to the required buffer angle. Finally, conclusions are presented in Section 4.6.

4.2 DEFORMABLE MODEL

The deformable model used for static bending is one derived from Euler-Bernoulli beam theory in which the deflection of the beam is related to the applied load through Equation 4.1:

$$EI\frac{d^4w}{dx^4} = q(x) \tag{4.1}$$

where E is the elastic modulus of the beam, I is the second moment of area of the beam (their product is the beam's flexural rigidity), $w(x)$ is the deflection of the beam, $q(x)$ is a distributed load, and x is the coordinate originating from the sail center and extending to the boom tip.

The slope, bending moment, and shear force in the beam can be found through successive derivatives of the deflection of the beam.

$$EI\frac{dw}{dx} = \theta(x) \tag{4.2}$$

$$EI\frac{d^2w}{dx^2} = M(x) \tag{4.3}$$

$$EI\frac{d^3w}{dx^3} = -V(x) \tag{4.4}$$

Each boom is modeled as a cantilevered beam off of the center of the sail. In order to solve for $w(x), x \in [0, \mathrm{L}]$, Equation 4.1 must be integrated four times while applying the appropriate boundary conditions at the beam origin (no boom deflection or slope) and tip (no applied moment and shear force equivalent to point load of the control vane).

The load applied to the beam arises from three sources, the solar radiation pressure on the sail membrane, the applied force by the tip vane – through solar radiation pressure, and the force caused by the rotation rate of the sail. The deflection is calculated numerically by dividing the beam into equal segments. The solar pressure applied to the sail is assumed to be equally distributed among the four sail booms. The force applied by the tip vane is modeled as a point load at the boom tip, assuming that the vane length is insignificant compared to the length of the sail, and the force due to the rotation rate is distributed according to Equation 4.5

$$F_{rot} = m_{dx}\omega^2 x \tag{4.5}$$

where m_{dx} is the mass of the boom element, ω is the rotation rate of the sail.

The relative magnitude of each of the three forces is examined using the maximal attainable rotation rate during a largest-angle maneuver considered in this chapter (65°) using the chosen parameters of each of the sails. The effects of each source of deflection for the three sails are summarized in Table 4.2. The resulting deflection from each of the three forces can be simply added to obtain the overall deflection under Euler-Bernoulli

beam theory and thus the contribution was found by simply dividing the deflection caused by one source by the overall deflection from all sources combined.

Table 4.2: Sail boom tip deflection by source

Sail	Sail Contribution [%]	Vane Force Contribution [%]	Rotation Contribution [%]
1	97.28	1.94	0.77
2	99.62	0.10	0.27
3	99.77	0.01	0.22

The insignificance of the deflection due to angular velocity and tip vane force even in the most extreme scenario results in neglecting these terms in this chapter. The vane force and rotation contribution are non-negligible for Sail 1, however, its overall deflection, shown below, is negligible due its relatively small size.

The distributed load is thus determined by calculating the effective area of solar sail material supported by each boom element and a simplified model of the force due to solar radiation pressure relative to the model used in Equation 2.4. In the simplified model, shown in Equation 4.6, the force normal to the sail surface depends on a reflectivity effectiveness parameter, η, and the angle relative to the sun, α. The effectiveness parameter allows deviation from the perfectly reflecting sail in which $\eta = 2$.

$$\boldsymbol{F} \cdot \hat{\boldsymbol{n}} = \eta P A \cos^2 \alpha \tag{4.6}$$

With the standard parameters for specular and diffuse reflection provided in [14] the relative error introduced by the simplified model (comparing equation 2.4 and 4.6) is less than 3%. The load for each boom element was then applied at the center of the element. The contribution of the applied load at the i^{th} element on the overall deflection

of the beam is given by Equation 4.7 for deflection of a cantilevered beam due to a point load [63].

$$w_i(x) = \begin{cases} \frac{P_{eff_i} x^2 (3x_i - x)}{6EI}, & 0 < x < x_i \\ \frac{P_{eff_i} {x_i}^2 (3x - x_i)}{6EI}, & x_i < x < L \end{cases} \tag{4.7}$$

where P_{eff_i} is the effective load applied to the beam element and x_i is the location of the i^{th} element. Obtaining the overall deflection of the boom is a matter of summing the contribution of each individual beam element.

$$w(x) = \sum_{i=1}^{numElem} w_i(x) \tag{4.8}$$

Since the applied load on each element is dependent on its projected area toward the Sun which in turn depends on the slope of the flexible sail's deflection, $\theta(x)$, a number of iterations may be needed in order to converge on the final shape of the sail.

The maximal deflection (when pointing directly at the sun) of each sail is shown in Figure 4.1, Figure 4.2, and Figure 4.3. Additionally, the boom slope for Sail 3 is given in Figure 4.4. Results from successive iterations are shown, starting from an un-deformed boom. Convergence was determined by a change in tip deflection of less than 1cm from the previous iteration.

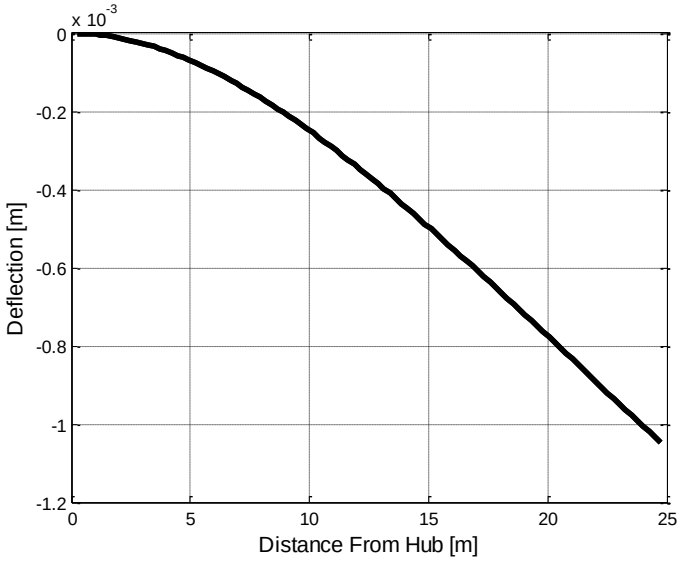

Figure 4.1: Maximal boom deflection – Sail 1

Tip deflection of Sail 1 is around 1mm which does little to effect the moment generated about the center of mass.

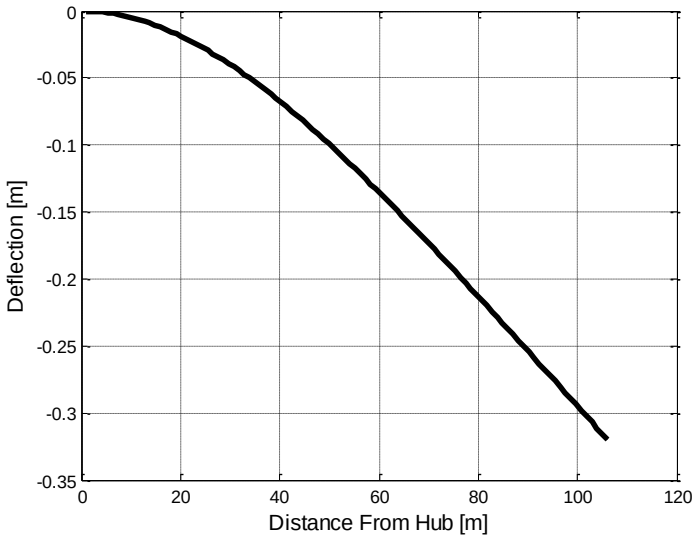

Figure 4.2: Maximal boom deflection – Sail 2

The tip deflection for Sail 2 of 0.320 meters agrees with the predicted tip deflection of between 0.25 meters and 0.35 meters (depending on sail membrane thickness) for the multi-point connected sail for which a detailed finite element analysis is presented in [1] and which subsequent models in literature have confirmed using various methodologies. The close agreement helps to validate the model used in this chapter.

While Sail 1 and Sail 2 converged immediately, it took 7 iterations for the highly flexible Sail 3 to converge. The solution oscillates around the correct deflection due to the changes in effective solar pressure at different boom slopes. The solution after one iteration overshoots the true solution and the decreased effective solar pressure causes the next iteration to undershoot the solution and so on until convergence. The final, converged, iteration is shown in red for both sail deflection and slope.

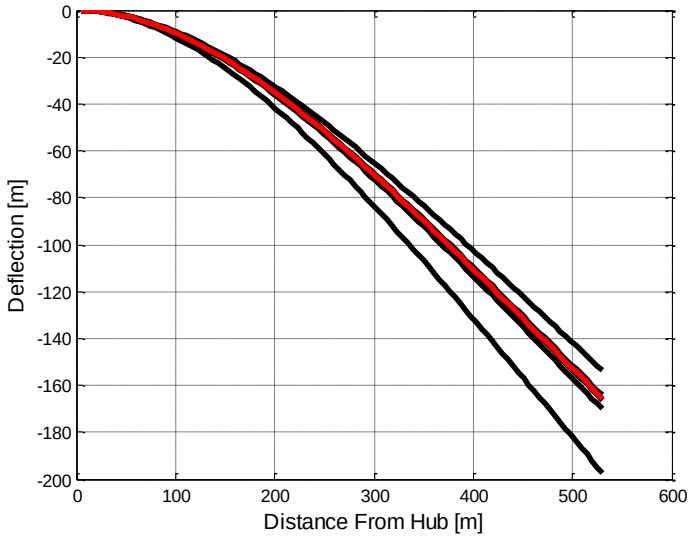

Figure 4.3: Maximal boom deflection – Sail 3

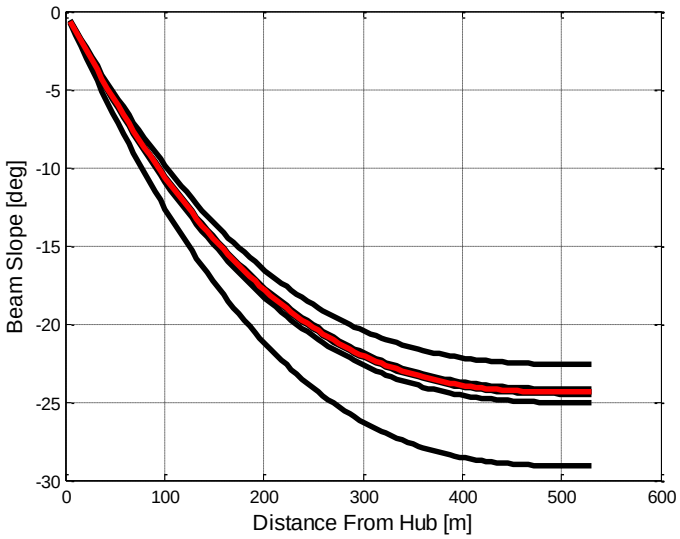

Figure 4.4: Maximal boom slope – Sail 3

Both the deflection and the slope are significant in the case of Sail 3 and will have an effect on its attitude dynamics.

4.3 TIME-OPTIMAL CONTROL

The attitude dynamics of the quasi-static flexible sail about the pitch axis are modeled in Equation 4.9.

$$\ddot{\alpha} = \frac{M}{J_{yy}} \tag{4.9}$$

where α is the pitch angle relative to the Sun, M is the applied moment, and J_{yy} is the pitch moment of inertia.

Using the parameters of the deformable sail, a bang-bang controller was implemented for a rest-to-rest maneuver. Maneuvering the solar sail from one attitude to another in minimum-time requires applying the maximal moment to produce rotation towards the desired attitude and a maximal moment in the opposite direction in order to stop the rotation rate just when the desired attitude is achieved [61].

Using the double integrator model of Equation 4.9, a sail initially at rest with, $\alpha(t_0) = \alpha_0$, and a final orientation of $\alpha(t_f) = \alpha_f$, the classical minimum-time solution with a constant allowable applied moment is defined by Equation 4.10

$$M = \begin{cases} M_{Max}, & 0 < t < \frac{tf}{2} \\ -M_{Max}, & \frac{tf}{2} < t < t_f \end{cases} \tag{4.10}$$

where the final time can be found to be,

$$t_f = \sqrt{\frac{4J_{yy}(\alpha_f - \alpha_0)}{M_{Max}}} \tag{4.11}$$

However, in the case studied in this chapter, the control moment is supplied by a tip vane and thus is dependent on α and the deflection of the boom tip. The desired rotation of the boom tip, γ, is also dependent on the slope of the boom tip. The maximal available moment can be formulated as an optimization problem.

$$M_{Max}(\alpha) = \max_{\gamma} \boldsymbol{r}\big(w(L)\big) \times \boldsymbol{F}_{Vane}(\gamma, \alpha, \theta) \tag{4.12}$$

Where $\boldsymbol{r}$ is a vector pointing from the center of the sail to the boom tip, and $\boldsymbol{F}_{Vane}$ is the force generated by the solar radiation pressure on the tip vane. Equation 4.12 is solved numerically to arrive at $M_{Max}(\alpha)$. The maximal angular acceleration as a function of pitch angle, which is simply the maximal moment divided by the pitch moment of inertia is shown in Figure 4.5. The optimal vane angle to achieve the maximal moment is shown in Figure 4.6 for all three sails.

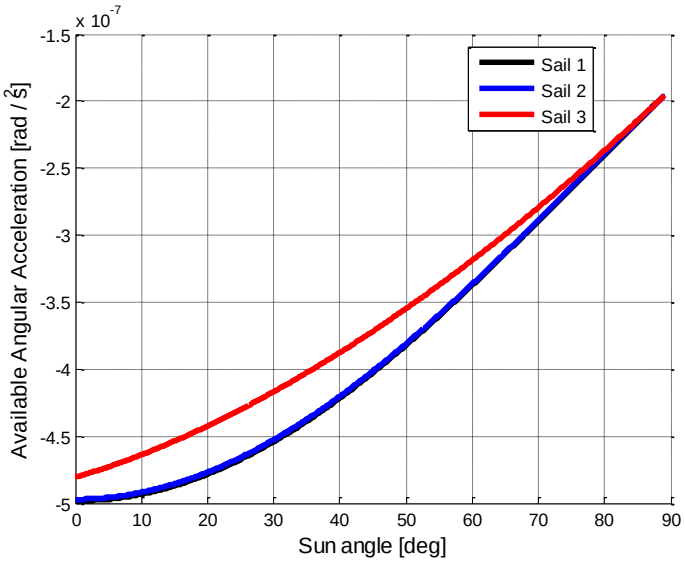

Figure 4.5: Maximal available moment

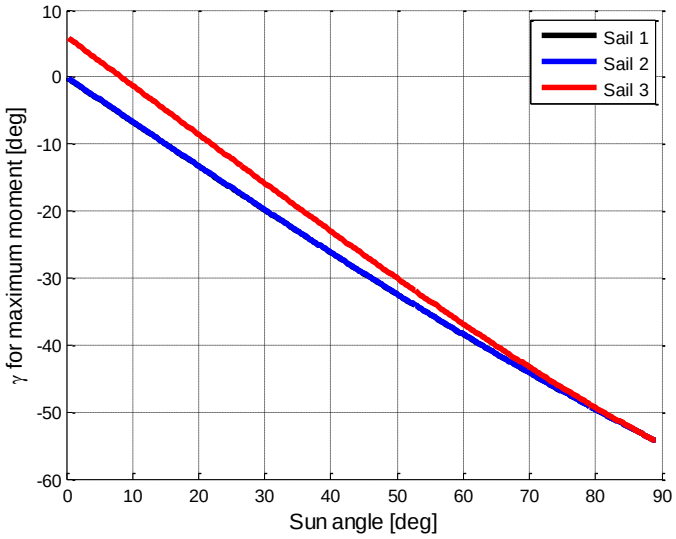

Figure 4.6: Desired tip-vane angle

The maximal acceleration and optimal vane angle for Sail 1 and Sail 2 nearly overlap while Sail 3 shows significant deviation due to its substantial flexibility and thus significant deflection and slope of its boom.

Due to this varying maximum moment, the maneuver time changes and the switch time no longer occurs in the middle of the maneuver. Solving for these is accomplished in an iterative gradient method suggested by [62] where an initial guess of the switch time, t_{s_0}, and final time, t_{f_0}, is updated as follows:

$$t_s(k) = t_s(k-1) - \beta_s\left(x_f - x_{f_{desired}}\right)^T \kappa_s\left(t_f\right) \tag{4.13}$$

$$t_f(k) = t_f(k-1) - \beta_f\left(x_f - x_{f_{desired}}\right)^T \kappa_f\left(t_f\right) \tag{4.14}$$

Where β_s and β_f are constants that affect the speed of convergence, $x_f = \left[\alpha\left(t_f\right) \quad \dot{\alpha}\left(t_f\right)\right]^T$, and κ_s and κ_f are the gradients of x with respect to the switch and final time respectively, evaluated at the final time, as defined in Equations 4.15 and 4.16:

$$\kappa_s\left(t_f\right) = \left[\frac{\partial \alpha}{\partial t_s} \quad \frac{\partial \dot{\alpha}}{\partial t_s}\right]^T \Bigg|_{t=t_f} \tag{4.15}$$

$$\kappa_f\left(t_f\right) = \left[\frac{\partial \alpha}{\partial t_f} \quad \frac{\partial \dot{\alpha}}{\partial t_f}\right]^T \Bigg|_{t=t_f} \tag{4.16}$$

Both of which are obtained using a numerical gradient formulation. A typical convergence of a naïve initial guess for switch time and final time for a 45° maneuver of Sail 2 is shown in Figure 4.7. This guess uses Equation 4.11 and thus underestimates the final time and overestimates the switch time since it assumes the maximal moment is available throughout.

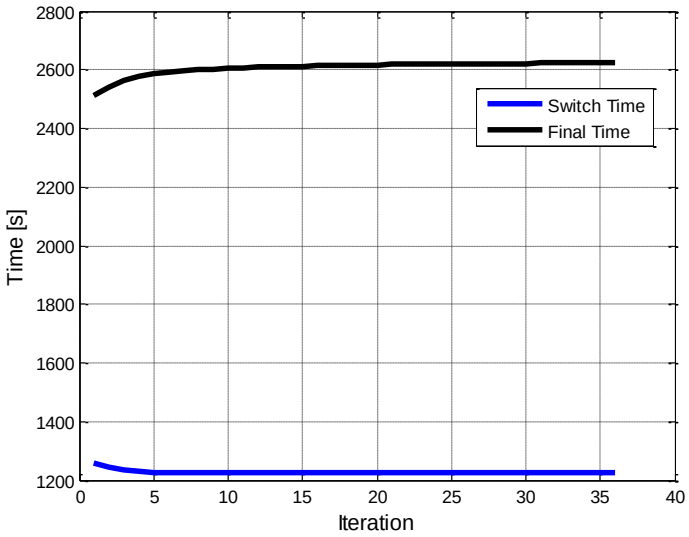

Figure 4.7: Convergence to desired switch and final times

The corresponding pitch angle and rotation rate time histories are given in Figure 4.8 and Figure 4.9 confirming that the desired rest-to-rest maneuver was accomplished, with a final angle error of $0.003°$ and rate error of $6.5 \times 10^{-5}\ °/s$.

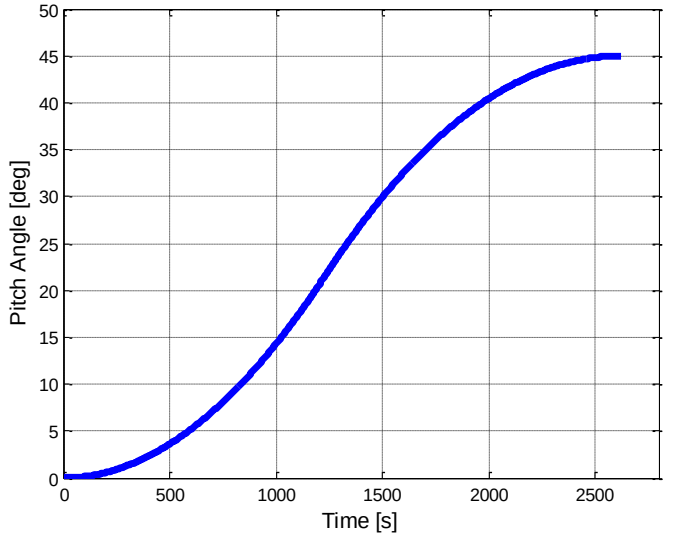

Figure 4.8: Pitch angle time history

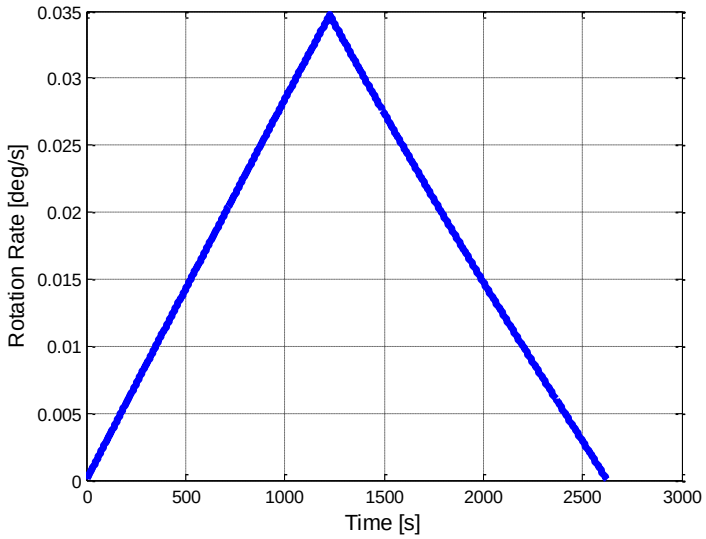

Figure 4.9: Pitch rate time history

Switch and final times for the three solar sail examples used in this chapter for the same 45-degree maneuver are given in Table 4.3 along with the final attitude and rate error along with the number of iterations required to arrive at the solution and the computational time in MATLAB using an Intel i7-3610QM CPU running Windows 7 (commercial laptop).

Table 4.3: Minimum-time results for three example solar sails

Sail	Switch Time [s]	Final Time [s]	Attitude Error [º]	Rate Error [º/s]	Iterations [#]	Computation Time [s]
1	1226	2621	0.016	7.7×10^{-6}	60	13.8
2	1227	2623	0.003	6.4×10^{-6}	36	8.2
3	1244	2706	0.013	9.2×10^{-6}	60	14.3

4.4 TRAJECTORY-BASED REACHABILITY

The switch and final times calculated in the previous section are now used for a trajectory-based reachability approach for ensuring within a statistical measure of safety, that there is no overshoot past the desired pitch angle. This approach takes into account

an estimate of the uncertainty around three key parameters that affect the final attitude of the sail: the flexural rigidity of the beam, EI, the sail effectiveness parameter, η, and the sail's moment of inertia about the pitch axis, J_{yy}. The approach is outlined below:

1. Obtain estimate of key parameters, $\widehat{EI}$, $\hat{\eta}$, $\hat{J}_{yy}$, along with an assumed Gaussian error distribution where each parameter's distribution is independent of the other's.
2. Define the initial pitch angle α_0 and a target pitch angle for this iteration $\alpha_f(k)$
3. Obtain switch time and final time, t_s, t_f, for the desired maneuver and system model parameters.
4. Simulate the maneuver p times using different admissible values from each parameter's error distribution and using a nominal switch time and final time.

$$\boldsymbol{\alpha}_i(t) = f\left(\alpha_0, \alpha_f(k), EI^*, \eta^*, J_{yy}^*, t_s, t_f\right), \qquad i = 1 \dots p$$
$$EI \sim \mathcal{N}\left(\widehat{EI}, \sigma_{EI}\right)$$
$$\eta \sim \mathcal{N}\left(\hat{\eta}, \sigma_\eta\right)$$
$$J_{yy} \sim \mathcal{N}\left(\hat{J}, \sigma_J\right)$$

   Where the "*" notation indicates a specific parameter chosen from the random distribution.
5. Calculate the upper limit of confidence interval of the final pitch angle

$$\alpha_u = \overline{\alpha_f} + z^* \sigma_{\alpha_f}$$

   where $\overline{\alpha_f}$ and σ_{α_f} are the mean and standard deviation of the final angle obtained in the p simulations and z^* is determined by the user's desired measure of safety
6. Adjust target pitch angle α_f for the next iteration according to this limit

$$\alpha_f(k+1) = \alpha_f(k) + \beta\left(\alpha_f(k) - \alpha_u\right)$$

where β affects the convergence speed.

7. If not converged, increment k and return to step 1.

Convergence to the desired buffer angle can be seen with the trajectories shown in Figures 4.10 and 4.11 for the case of maneuvering Sail 1 from a pitch angle of 0° to 45° with $\sigma_J = 0.03J_{yy}, \sigma_{EI} = 0.1EI$, and $\sigma_\eta = 0.05\eta$ using 400 simulations.

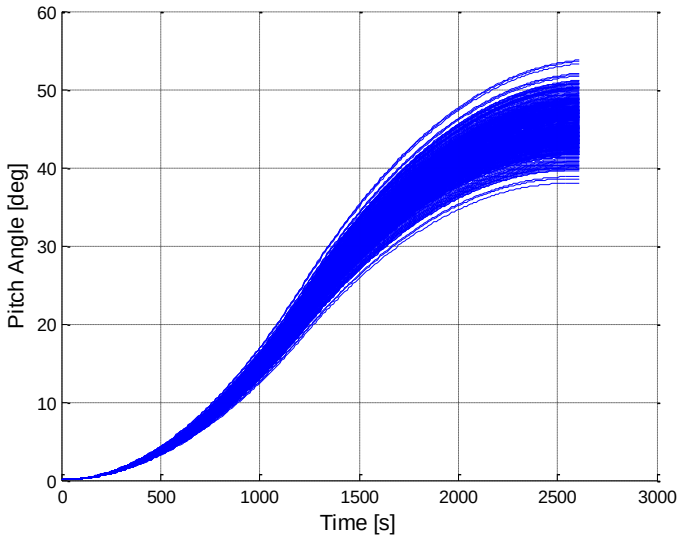

Figure 4.10: Convergence on buffer angle –Iteration 0, Buffer = 0°

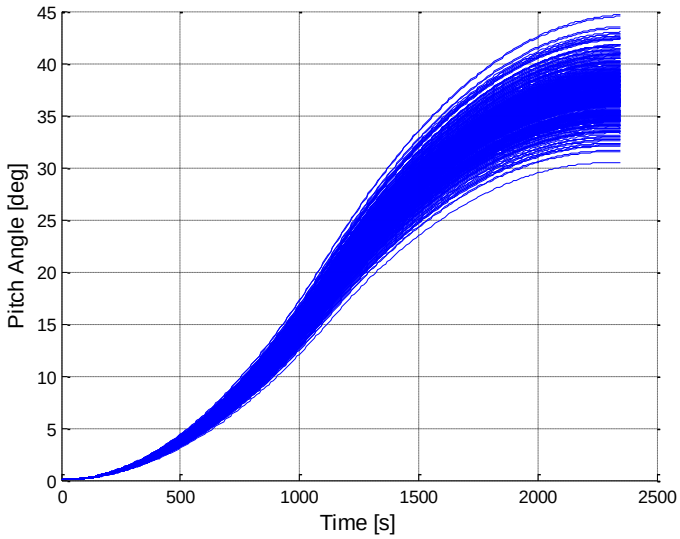

Figure 4.11: Convergence on buffer angle –Iteration 1, Buffer = 3.1°

In Figure 4.10, many trajectories overshoot the desired pitch angle while in Figure 4.11 no trajectories go beyond the desired pitch angle of 45°.

Naturally, as the uncertainty in any of the key parameters increases so does the required buffer to avoid overshooting. This effect is shown in Figure 4.12 through Figure 4.14 where the buffer angle is plotted against each of the three parameters' uncertainty for each one of the three sails. Each data set was produced while assuming the other two parameters are perfectly known. A baseline pitch maneuver starting at a sun-pointing orientation and ending at 45° to the sun is tested using each of the three sails.

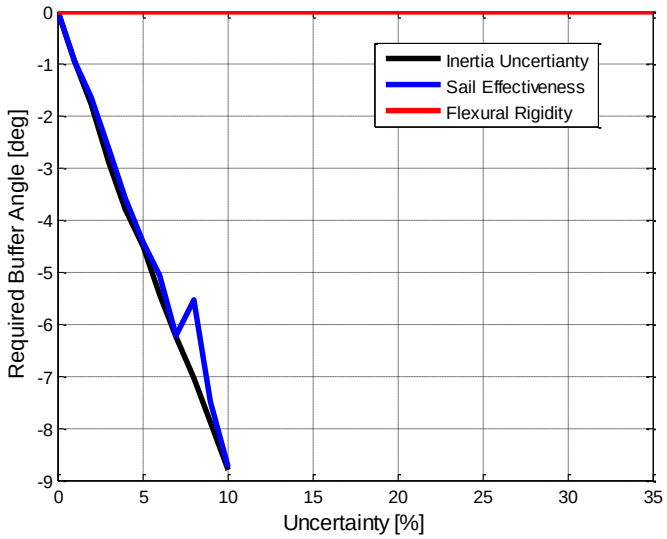

Figure 4.12: Effect of uncertainty on required buffer angle – Sail 1

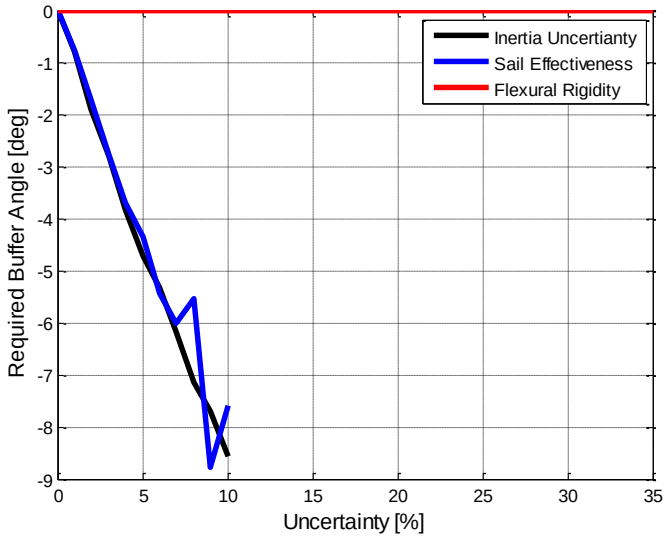

Figure 4.13: Effect of uncertainty on required buffer angle – Sail 2

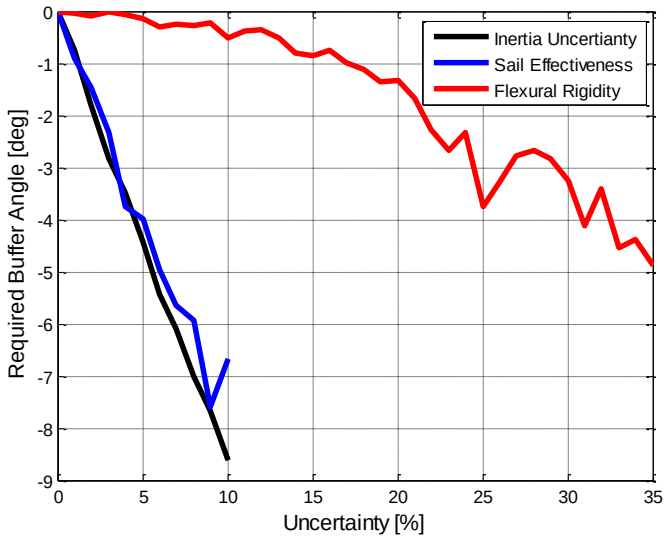

Figure 4.14: Effect of uncertainty on required buffer angle – Sail 3

Even with a 35% uncertainty in the flexural rigidity of Sail 1 and Sail 2, the flexible effect is so small that it is of little consequence with regard to the distribution of final pitch angle as indicated by the red line in Figure 4.12 and 4.13 remaining at 0. The uncertainty in moment of inertia and sail reflectivity effectiveness have a similar effect on the required buffer angle since they both directly affect that angular acceleration of the sail. Due to the high flexibility of Sail 3, uncertainty in its flexibility does indeed play a significant role in the final attitude of the sail.

4.5 Machine Learning for Uncertainty Management

Machine learning was used in two different ways in this analysis. The first was for the purpose of system identification in order to reduce the uncertainty in the key system parameters. In addition, machine learning was used to find an easily accessed relation between the levels of uncertainty in these parameters along with the desired change in pitch angle to the required buffer angle to be taken.

As in Chapter 3, machine learning is accomplished using MATLAB's neural network toolbox. The toolbox uses an algorithm based on minimization of a nonlinear function developed by Levenberg and Marquardt to update the weights applied in the neural network with the performance of the network calculated using mean squared error. The input data was split such that 70% was used for updating the network nodes' weight and 30% was used for testing and validation.

The effects of model uncertainty on the required buffer angle motivate an effort to reduce this uncertainty. An approach taken here is system identification through machine learning where the number of nodes in the hidden neural network was 20. This number was found to give the best results using a trial and error approach. A baseline maneuver where the vane angle is held constant for a fixed period of time is used to identify the key model parameters discussed in this chapter.

$$\widehat{EI} = f\big(\boldsymbol{\alpha}(t), \dot{\boldsymbol{\alpha}}(t)\big)$$

$$\hat{\eta} = f\big(\boldsymbol{\alpha}(t), \dot{\boldsymbol{\alpha}}(t)\big)$$

$$\hat{J}_{yy} = f\big(\boldsymbol{\alpha}(t), \dot{\boldsymbol{\alpha}}(t)\big)$$

The relation is learned by executing multiple simulations of the baseline maneuver while varying values of each of the key parameters by an assumed initial uncertainty and reporting the true value of each parameter along with the pitch time history to the learning algorithm. The standard deviation of the variation of the flexural rigidity, sail effectiveness, and moment inertia was 25%, 10%, and 2% of their baseline values. This choice of variation reflects little knowledge of the effective flexural rigidity of the sail, a moderate knowledge of the sail reflectivity effectiveness, and accurate knowledge of the sail's moment of inertia which are easier to model prior to launch.

Using these ranges of parameter variation, the resulting standard deviations of the errors between the learned parameters and the true parameters for an independent set of 250 simulation runs are given in Table 4.4.

Table 4.4: System identification results

Sail	$\sigma\left(\widehat{EI} - EI_{true}\right)$	$\sigma\left(\widehat{J_{yy}} - J_{true}\right)$	$\sigma(\hat{\eta} - \eta_{true})$
Sail 1	1.98%	24.31%	2.00%
Sail 2	2.04%	23.95%	2.01%
Sail 3	1.64%	1.94%	1.94%

The flexural rigidity of the third sail was much more easily learned by this process compared to the other two sails. This is a direct result of the much greater impact of a highly flexible sail on the dynamics of the system as discussed in the previous section. Therefore, while the system's flexural rigidity of Sail 1 and Sail 2 cannot be easily learned, it will have little impact on the required buffer angle and can thus be eliminated as a variable in these cases. These reduced levels of uncertainty can now be used to reduce the buffer angle required to avoid overshoot as discussed in the previous section. Figure 4.15 shows the true and learned scaled flexural rigidity using the above process for Sail 3.

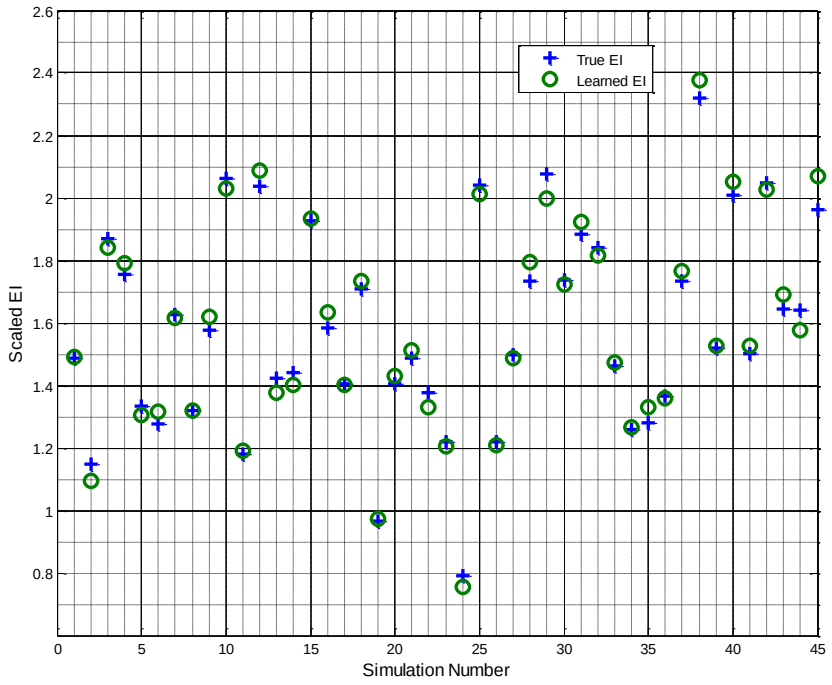

Figure 4.15: Learned flexural rigidity for Sail 3

The good agreement between the truth and learned flexural rigidity over a wide range of values shows the value to this method of system identification using a fairly short process that can be performed in under an hour on-orbit as other checkout procedures are being done.

The lowest initial uncertainty between the moment of inertia and sail reflectivity effectiveness serves as a lower bound to identifying both of these parameters. They both act to affect the attitude dynamics in the same way as they both are directly involved in determining the angular acceleration. Therefore, only using the observations of attitude time history makes it difficult to discern between the two in the machine learning process. One way to isolate the effect of sail reflectivity effectiveness is to add observations of sail position at different points in time since position is directly affected

by the sail effectiveness and not by the sail's moment of inertia. These observations may require a longer time history for calibration depending on the accuracy of position estimation of the sail (be it on-board, or using ground-based ranging techniques).

Machine learning is also used to assist in finding the desired buffer angle given model uncertainties and initial and final pitch angles.

$$\theta_{buffer} = f\left(\sigma_{EI}, \sigma_{\eta}, \sigma_{J}, \alpha_0, \alpha_f\right)$$

The process of learning the buffer angle is identical to that described above for the purpose of system identification, only that in this case, the number of nodes in the hidden neural network layer is four. The standard deviations of the uncertainties were taken to be those learned using the system identification process, the initial pitch angle was taken to be 0 and the final pitch angle was taken to vary between 35° and 65°. The root-mean-square of the difference between the true calculated buffer angle described in the previous section to that learned by the relation above are 0.61°, 0.53°, and 1.25° for Sail 1, 2, and 3 respectively. The small magnitude of these errors compared to the large angle maneuvers shows the viability of this approach.

The advantage of using a machine-learned relation to obtain the required buffer angle for every maneuver is on-board computational time with no significant requirement on additional data storage for the learned relations. The time to learn the required buffer angle using the machine-learned function is more than five orders of magnitude quicker than that of directly calculating the buffer angle. The machine-learned function can be evaluated in approximately 1 millisecond, while the direct calculation of each required buffer angle takes an average of 500 seconds on the same commercial laptop described above. In addition, this process allows for separation between learning this relation and on-board operations. Therefore, the relation that produces the required buffer angle may be learned a-priori on ground-based computers and then only the function needs to be

evaluated in the solar sail's on-board computer. Utilizing systems that are disconnected from the on-board hardware allows for increasing the accuracy of each algorithm used in the process including sail deflection, minimum-time, and the overall trajectory-based reachability algorithm.

4.6 CONCLUSIONS

This chapter describes an algorithm for uncertainty management for three key parameters that are not easily modeled or tested that have significant impact on solar sail attitude control. A minimum time control law using boom-tip vanes was shown to significantly improve performance over the proposed solution in Chapter 2. In this chapter, Sail 1, which is similar to the sail model used in Chapter 2, was able to perform a 45° maneuver in approximately 45 minutes. In comparison, in Chapter 2 a 35° maneuver was shown to take approximately 5 hours. The desire to not overshoot the target attitude given an unavoidable model uncertainty motivated development of a trajectory-based reachability algorithm that calculates a safe buffer for the minimum-time controller. An arbitrary large-angle maneuver can thus be planned and performed and the sail can reach arrive within a small buffer of its intended attitude. This maneuver can be iterated upon more than once to get arbitrarily close to the desired attitude.

An added benefit of the ability to learn the model parameters with a simple on-orbit maneuver is that the time-varying optical properties of the sail due to material degradation can be taken into account without any algorithmic changes or controller tuning. When the optical properties of the sail degrade, the learned sail effectiveness will be estimated and updated accordingly. In addition, re-learning the sail effectiveness can take into account changes in the distance between the sail and the sun that change the solar radiation pressure. This re-learning can be done using the same on-orbit maneuver

used initially and should not represent a burden on the mission due to its simplicity. Long-duration interplanetary missions for which the solar sail is a viable and sometimes only option will require handling these changes.

Chapter 5
Conclusions

5.1 SUMMARY OF WORK

This dissertation discussed different approaches to handle the unique challenges posed by solar sail attitude control today with a look towards future challenges of solar sail technology and the barriers to its progress that the technology faces. The algorithms that were developed can be classified into two categories, backward and forward looking.

In the first category developed in Chapter 2, the main focus was to observe the time history of the control action as it worked to actively counteract and reject disturbances that arose due to various sources of uncertainties. Instead of direct handling of these uncertainties, this approach focused on the action that could be taken to ensure that the sail operated about an equilibrium position regardless of the disturbance moment imposed on it. The control system worked to reduce motion in the control vanes and showed the benefit of incorporating passive stability as part of the controller. The controller was shown to obtain a pointing error of less than 2° and low system actuation once an equilibrium position was estimated and used.

The second approach, inspired by reachability analysis, focused on predicted possible future states and modifying the control scheme accordingly. In the case of the UAV testbed, this was done by supervising the low-level controller, be it PID auto-pilot or human operator, and projecting the commanded control signal inside a safety ellipsoid that approximates the safe set of inputs at that point in time. The safe set of inputs is a sequence of control inputs for which the UAV will not violate a set of state constraints that are deemed unsafe.

In the case of the solar sail, predictions were made about the required size of the buffer angle from the target attitude to ensure that the unsafe state of overshoot is not

encountered. This was accomplished within the framework of a flexible sail with significant uncertainties as well as substantial uncertainties in the sail's reflectivity effectiveness which can be used to account for non-flat-plate effects such as sail wrinkling and billowing and optical property degradation over time.

Machine learning was shown to provide substantial benefits both in system identification to reduce uncertainties and to allow for separation between computationally expensive reachability analysis and the need for feasible real time controllers in embedded systems. This allows utilization of arbitrarily accurate models and simulations to be performed offline and creates a very flexible architecture that does not make significant demands of on-board computing capabilities. The learned set of safe inputs could be accessed and modified on the order of milliseconds and is thus easily incorporated as part of the control system. The easily accessible learned relationship between model uncertainties and required buffer angle is similarly beneficial and additionally can be used by mission planners to assess their design effort and allows them to "buy-down" uncertainties in the areas which would be most cost-effective in terms of improving system performance.

The main innovations in this work that can thus be summarized as an algorithm for the autonomous adjustment of trim angles that allows the controller to operate around an equilibrium point that is determined by observing the control history, a way to constrain future inputs in real time in order to provide a statistical measure of safety against a set of arbitrary constraints, and a method that allows for optimal time reorientation while not sacrificing safe operations in the light of the unique uncertainties that exist in solar sail design.

5.2 FUTURE WORK

The approach taken in Chapter 4 for large-angle maneuvers can be repeated numerous times for smaller and smaller angles decreasing the buffer between the planned target and the ultimate desired orientation. However, the benefit of using a minimum-time maneuver diminishes for smaller-angle maneuvers. It is likely that a combination of the algorithm proposed in Chapter 2 will be more suitable for the final part of the maneuver. Taking advantage of the passive stability inherent in the square solar sail design and safe maneuvering employed by the first algorithm will assist in minimizing control effort and facilitate accurate pointing. The algorithms used in Chapter 2 will benefit from the reduced system uncertainty due to the learned parameter estimation of Chapter 4.

The adaptable nature of the overall algorithm presented in Chapter 4 allows for various extensions within the same framework. Since the machine learning of the buffer angle is accomplished offline, other, more sophisticated sail models can be used. This includes the possibility of using finite element analysis offline and incorporating its results in real time control through machine learning. Other extensions can include a more sophisticated model of the solar radiation force on the sail. Possible future adaptation of the algorithm may include fault handling such as a tear in one of the sail's quadrants. Instead of incorporating just one sail reflectivity effectiveness parameter, one can extend the concept to allow for an effectiveness parameter for subsections of the sail. This may allow for fault detection if one of the effectiveness parameters is deemed to be significantly different than the others.

A natural extension of this work is to utilize the algorithms presented in Chapter 4 for 3-axis attitude control. Extension is trivial for independent control about each axis, and so successive reorientations are possible without any change to the algorithm.

However, simultaneous rotation about a transverse axis and the axis normal to the sail may require some adaptation to account for the effect of rotating the vanes for the purpose of transverse control on the availability of moment to be applied about the axis normal to the sail. It is worth noting that using minimum-time control for rotation about the axis normal to the sail may be of limited interest in the context of ensuring no overshoot since this may be of little consequence in terms of mission safety.

Though the focus of this dissertation has been on solar sail attitude control and to some degree UAV control, it can be further generalized. The framework developed, particularly in chapter 3 and chapter 4, can easily be adapted to control of any dynamic system. The dynamic model would need reformulation and the parameters of interest would change, however, the basic premise would hold. The trajectory-based reachability analysis can be used with any system where some states are undesirable or considered unsafe. Any such analysis would benefit from the decoupling methodology suggested here in order to apply it to a system in real time.

Appendix A: Chapter 2 Coordinate Systems

This section gives details of the coordinate systems used in Chapter 2. The main sail coordinate system and the vanes' coordinate system are shown in Figure A.1.

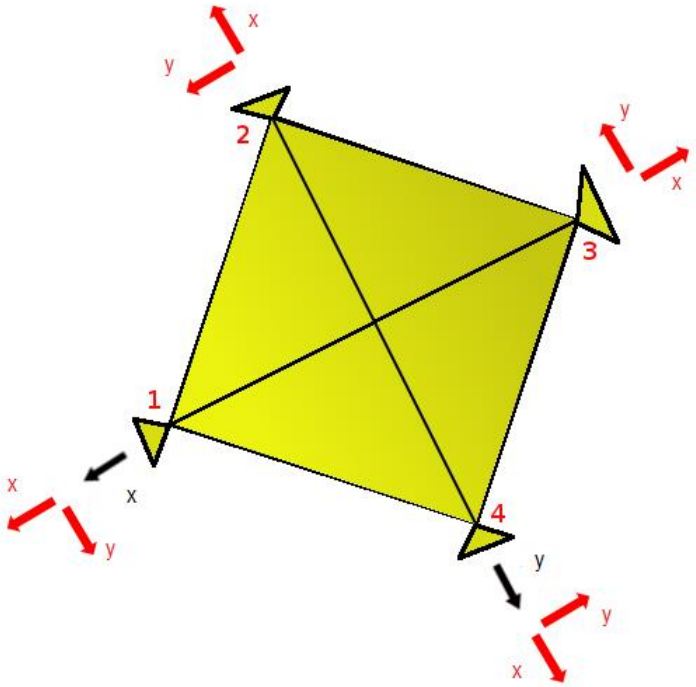

Figure A.1 Vane frames (in red) and sail frame (in black) when 0° cant and twirl angles

The vane coordinate system is fixed on each of the vanes and changes with respect to the sail frame as each vane is canted and twirled as commanded by control system. The origin of the frame is at the point connecting the vane to the boom. When all cant and twirl angles are zero, vane 1's frame is aligned with the main sail frame. The z-axis is normal to the vane and sail, the x-axis extends in the direction of the boom through the center of the vane, and the y-axis completes the right-handed system. Vane 2's frame is rotated by 90 degrees about the vane/sail z-axis as shown in Figure A.2.

The sun-pointing frame is a frame that is centered at the sail-frame origin, with the z-axis pointed in the direction of the sun, the x-axis parallel to the ecliptic plane, and the y-axis completing the right-handed coordinate system. This frame is called the Sun frame (in yellow) in the following figure by Derbes [1].

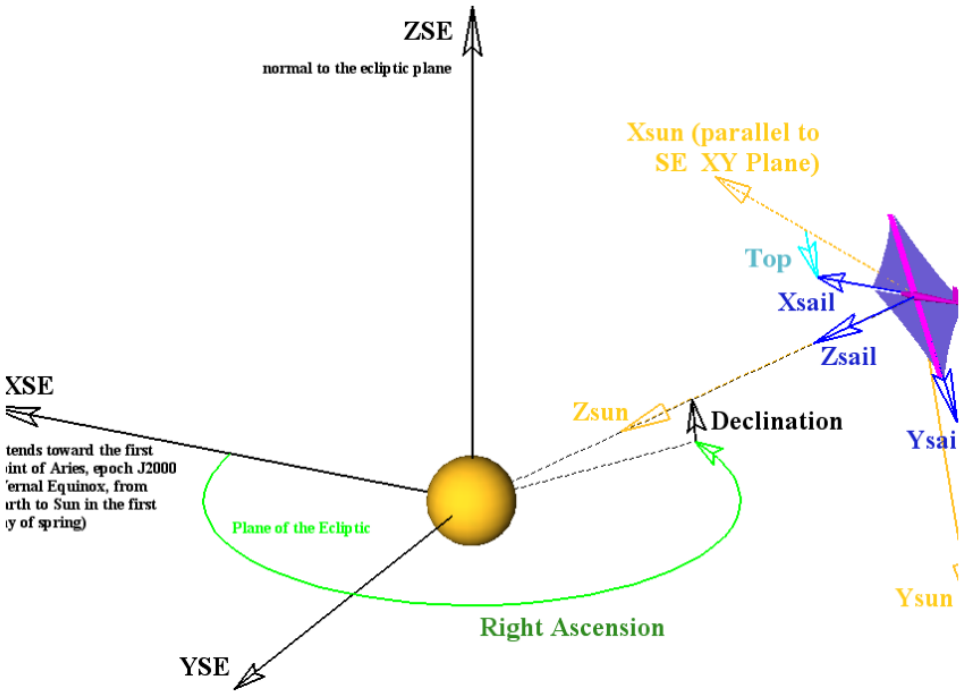

Figure A.2: Sun pointing frame [12]

This frame is used in order to describe the orientation of the sail-frame in terms of sun-angles (top angle, sun-incidence angle, and flat-spin angle) which are a set of Euler angles that describe a 3-2-3 rotation.

Appendix B: Chapter 3 Nomenclature and Coordinate Systems

This section gives details of the coordinate systems used in Chapter 3 along with nomenclature not defined as part of the main chapter for brevity.

In order to arrive at the UAV body frame from an inertial frame three successive rotations are performed. First, the inertial reference frame is rotated about the inertial frame's 3-axis by an angle ψ into the first intermediate frame. The first intermediate frame is then rotated about its 2-axis by an angle θ to arrive at the second intermediate frame. Finally, the second intermediate frame is rotated about its 1-axis by an angle ϕ to arrive at the body frame. This sequence amounts to a 3-2-1 rotation sequence. The transformation between the inertial frame to the body frame thus becomes:

$$R_i^b = \begin{bmatrix} cos\theta cos\psi & cos\theta sin\psi & -sin\theta \\ sin\phi sin\theta cos\psi - cos\phi sin\psi & sin\phi sin\theta sin\psi + cos\phi cos\psi & sin\phi cos\theta \\ cos\phi sin\theta cos\psi + sin\phi sin\psi & cos\phi sin\theta sin\psi - sin\phi cos\psi & cos\phi cos\theta \end{bmatrix} \tag{B.1}$$

The instantaneous rotation rates, *p,q,* and *r,* are in the UAV's body frame and since the Euler angles are given in intermediate frames, they must be rotated into the same frame in order to be related to each other. This process of appropriate rotations is shown in Equation B.2

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\phi & sin\phi cos\theta \\ 0 & -sin\phi & cos\phi cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{B.2}$$

The rotation matrix can then be inverted to arrive at the relation between the Euler angle derivatives and the instantaneous rotations as shown in the equations of motion in Chapter 3.

Nomenclature for undefined constants and variables in Chapter 3 is given below.

Table B.1: Undefined constants used in Chapter 3

Constant	Definition	Units
S_{prop}	Propeller surface area	m^2
c_{prop}	Propeller mean chord	m
k_{motor}	Motor effectiveness	$-$
m	Aircraft mass	kg
b	Wing span	m
c	Wing mean chord	m
S	Wing area	m^2
C_{Y_0}	Zero side force coefficient	$-$
C_{Y_β}	Side force coefficient due to angle of sideslip	$-$
C_{Y_p}	Side force coefficient due to roll rate	$-$
C_{Y_r}	Side force coefficient due to yaw rate	$-$
$C_{Y_{\delta_a}}$	Side force coefficient due to aileron deflection	$-$
$C_{Y_{\delta_r}}$	Side force coefficient due to rudder deflection	$-$
C_{p_0}	Zero roll moment coefficient	$-$
C_{p_β}	Roll moment coefficient due to angle of sideslip	$-$
C_{p_p}	Roll moment coefficient due to roll rate	$-$
C_{p_r}	Roll moment coefficient due to yaw rate	$-$
$C_{p_{\delta_a}}$	Roll moment coefficient due to aileron deflection	$-$
$C_{p_{\delta_r}}$	Roll moment coefficient due to rudder deflection	$-$
C_{m_0}	Zero Pitch moment coefficient	$-$
C_{m_α}	Pitch moment coefficient due to angle of attack	$-$
C_{m_q}	Pitch moment coefficient due to pitch rate	$-$
$C_{m_{\delta_e}}$	Pitch moment coefficient due to elevator deflection	$-$

C_{r_0}	Zero yaw moment coefficient	–
C_{r_β}	Yaw moment coefficient due to angle of sideslip	–
C_{r_p}	Yaw moment coefficient due to roll rate	–
C_{r_r}	Yaw moment coefficient due to yaw rate	–
$C_{r_{\delta r}}$	Yaw moment coefficient due to rudder deflection	–
J_i	Moment of inertia about the i axis	$kg-m^2$
J_{ij}	The i-j product of inertia	$kg-m^2$
Γ	$J_x J_z - J_{xz}^2$	kg^2-m^4
Γ_1	$\frac{J_{xz}(J_x - J_y + J_z)}{\Gamma}$	–
Γ_2	$\frac{J_z(J_z - J_y) + J_{xz}^2}{\Gamma}$	–
Γ_5	$\frac{J_z - J_x}{J_y}$	–
Γ_6	$\frac{J_{xz}}{J_y}$	–
Γ_7	$\frac{J_x(J_x - J_y) + J_{xz}^2}{\Gamma}$	–

Table B.2: Undefined functions of angle of attack used in Chapter 3 (dimensionless)

Variables	Definition
$C_X(\alpha)$	Axial force coefficient
$C_{X_q}(\alpha)$	Axial force coefficient due to pitch rate
$C_{X_{\delta_e}}(\alpha)$	Axial force coefficient due to elevator deflection
$C_Z(\alpha)$	Normal force coefficient
$C_{Z_q}(\alpha)$	Normal force coefficient due to pitch rate
$C_{Z_{\delta_e}}(\alpha)$	Normal force coefficient due to elevator deflection

Bibliography

[1] Hughes, G. W., Macdonald, M., and McInnes, C.R., "Analysis of a Solar Sail Mercury Sample Return Mission," *55th International Astronautical Congress,* IAC-04-Q.2.B.08, Vancouver, Canada, 2004.

[2] Macdonald, M. and McInnes, C. R. "Solar Sail Science Mission Applications and Advancement," Advances in Space Research. Vol. 48, No. 11, 2011, pp. 1702-1716.

[3] NASA Technology Roadmaps Technology Area 2 – In-Space Propulsion Technologies, NASA, Washington, DC, 2015.

[4] Wie, B., "Solar sail attitude control and dynamics, part 1," Journal of Guidance, Control and Dynamics, Vol. 27, No. 4, 2004.
doi: 10.2514/1.11134

[5] Wie, B., "Solar sail attitude control and dynamics, part 2," Journal of Guidance, Control and Dynamics, Vol. 27, No. 4, 2004.
doi: 10.2514/1.11133

[6] Wie, B., "Solar-Sail Dynamics and Control," *Space Vehicle Dynamics and Control,* 2nd ed., AIAA Education Series, Reston, 2008, pp. 788-794.
doi: 10.2514/4.860119

[7] Mettler, E., Acikmese, A. B., and Ploen, S. R. "Attitude Dynamics and Control of Solar Sails with Articulated Vanes." AIAA Guidance, Navigation, and Control Conference and Exhibit, AIAA 2005-6081, San Francisco, California, 2005.

[8] Wie, B., Murphy, D., Paluszek, M., and Thomas, S. "Robust Attitude Control Systems Design for Solar Sails, Part 2: MicroPPT-based Secondary ACS." AIAA Guidance, Navigation, and Control Conference, AIAA 2004-5011, Providence, RI, 2004.

[9] Derbes, B. and Lichodziejewski, D. “Residual Surface, Shape, and Mass Property Errors: Effect on Solar Sail Navigation and Attitude Control.” 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA 2007-2414, Honolulu, Hawaii, 2007.

[10] Tsuda, Y., Saiki, T., Funase, R., Shirasawa, Y. and Mimasu, Y. “Shape Parameters Estimation of IKAROS Solar Sail Using In-Flight Attitude Determination Data.” 52nd AIAA/ASME/ASCE/AHS/ASC Structure, Structural dynamics and Materials Conference, Denver, Colorado, 2011.

[11] Derbes, B., Lichodziejewski, D, Veal, D. “A Yank and Yaw Control System for Solar Sails,” 14th AAS/AIAA Space Flight Mechanics Conference, AAS 04-284, Maui, Hawaii. 2004.

[12] Derbes, B., Lichodziejewski, D., Ellis, J., Scheeres, D., “Sailcraft coordinate systems and format for reporting propulsive performance,” *14th AAS/AIAA Space Flight Mechanics Conference*, AAS 04-100, Maui, Hawaii, 2004.

[13] Eldad, O., Lightsey, G., “Attitude Control of the Sunjammer Solar Sail Mission,” *Proceedings of the AIAA/USU conference on Small Satellites*, Propulsion, SSC14-X-4. http://digitalcommons.usu.edu/smallsat/2014/Propulsion/3/ [retrieved 10 October 2014].

[14] McInnes, C. R., “Solar Sail Force Models,” *Solar Sailing: Technology, Dynamics and Mission Applications*, Springer, London, 2004, pp. 32-55.
doi: 10.2514/2.4604

[15] Kurzhanskiy, Alex A., and Varaiya P., “Ellipsoidal Techniques for Reachability Analysis of Discrete-Time Linear Systems,” *IEEE Transactions on Automatic Control,* Vol. 52 No.1, 2007, pp. 26-38.
doi: 10.1109/TAC.2006.887900

[16] Asarin, E., Bournez, O., Dang, T., and Maler, O., "Approximate Reachability Analysis of Piecewise-Linear Dynamical Systems," *Proceedings of the 3rd International Conference of Hybrid Systems: Computation and Control*, Vol. 1790 of LNCS, Springer, Pittsburgh, PA, USA, 2000, pp. 20-31.
doi: 10.1007/3-540-46430-1_6

[17] Girard, A., "Reachability of Uncertain Linear Systems Using Zonotopes," *Proceedings of the 8th International Conference of Hybrid Systems: Computation and Control*, Volume 3414 of LNCS., Springer, Zurich, Switzerland, 2005, pp. 291–305.
doi: 10.1007/978-3-540-31954-2_19

[18] Broucke, M.E., and Ganness, M., "Reach control on simplices by piecewise affine feedback," *SIAM Journal on Control and Optimization*. Vol. 52 No. 5, 2014, pp. 3261-3286
doi: 10.1109/ACC.2009.5160258

[19] Kazerooni, E.S., and Broucke, M.E., "Reach controllability of single input affine systems on a simplex," *IEEE Transactions on Automatic Control*. Vol. 59, No. 3, 2014, pp. 738-744.
doi: 10.1109/TAC.2013.2274707

[20] Kurzhanski, A. B., and Varaiya, P., "Ellipsoidal Techniques for Reachability Analysis: Internal Approximation," *Systems & Control letters,* Vol. 41 No. 3, 2003, pp. 201-211.
doi: 10.1016/S0167-6911

[21] Kapinski, J., and Krogh, B., "Verifying Switched-Mode Computer Controlled Systems," *Proceedings of the 2002 IEEE International Symposium on Computer*

Aided Control System Design, IEEE, Glasgow, Scotland, 2002, pp. 98-103.
doi: 10.1109/CACSD.2002.1036936

[22] Lafferriere, G., Pappas, G.J., and Yovine, S., "Symbolic reachability computation for families of linear vector fields," *Journal of Symbolic Computation*, Vol. 32 No. 3, 2001, pp. 231-253.
doi: 10.1006/jsco.2001.0472

[23] Mitchell, I., Bayen A.M., and Tomlin, C.J., "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on Automatic Control,* Vol. 50 No. 7, 2005, pp. 947-957.
doi: 10.1109/TAC.2005.851439

[24] Chutinan, A., and Krogh, B.H., "Computational Techniques for Hybrid System Verification," *IEEE Transactions on Automatic Control,* Vol. 48 No. 1, 2003, pp. 64-75.
doi: 10.1109/TAC.2002.806655

[25] Desilles, A., Zidani, H., Cruck, E., "Collision Analysis for an UAV," *Proceedings of Guidance, Navigation, and Control Conference*, AIAA 2012-4526, Minneapolis, MN, USA, 2012.
doi: 10.2514/6.2012-4526

[26] Aubin, J.P, Bayen, A.M., and Saint-Pierre, P., *Viability Theory: new directions*. Springer Science and Business Media, 2011.

[27] Bayen, A.M., Crück, E., and Tomlin, C.J., "Guaranteed overapproximations of unsafe sets for continuous and hybrid systems: solving the hamilton-jacobi equation using viability techniques," *Hybrid Systems: Computation and Control*. Springer Berlin Heidelberg, 2002, pp. 90-104.
doi: 10.1007/3-540-45873-5_10

[28] Niarchos, K. N., and Lygeros, J., "A Neural Approximation to Continuous Time Reachability Computations." *Proceedings of the 45th IEEE Conference on Decision and Control*, IEEE, San Diego, CA, USA, 2006, pp. 6313-6318.
doi: 10.1109/CDC.2006.377358

[29] Maidens, J., and Arcak, M., "Reachability Analysis of Nonlinear Systems Using Matrix Measures," *IEEE Transactions on Automatic Control*, Vol. 60 No.1, 2015, pp. 265-270.
doi: 10.1109/TAC.2014.2325635

[30] Julius, A. A., and Pappas, G.J., "Trajectory Based Verification Using Local Finite-Time Invariance," *Proceedings of the 12th International Conference of Hybrid Systems: Computation and Control*, Vol. 5469 of LNCS, Springer, San Francisco, CA, USA, 2009, pp. 223-236.
doi: 10.1007/978-3-642-00602-9_16

[31] Beard, R.W., and McLain, T.W., *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, Princeton, NJ, 2012. Chap. 4.

[32] Ross, A.E., et al., "A Machine Learning Approach for Real-Time Reachability Analysis," *Intelligent Robots and Systems, IEEE/RSJ International Conference on,* IEEE, Chicago, IL, USA, 2014, pp. 2202-2208.
doi: 10.1109/IROS.2014.6942859

[33] Lavretsky, E., and Wise, K., *Robust and Adaptive Control: With Aerospace Applications*. Springer Science & Business Media, 2012.
doi: 10.1007/978-1-4471-4396-3

[34] Greenstreet, M.R., and Mitchell, I., "Reachability Analysis Using Polygonal Projections," *Proceedings of the 2nd International Conference of Hybrid Systems: Computation and Control*, Vol. 1569 of LNCS, Springer, Berg en Dal, The

Netherlands, 1999, pp. 103-116.
doi: 10.1007/3-540-48983-5_12

[35] Chutinan, A., and Krogh, B., "Verification of Infinite-State Dynamic Systems Using Approximate Quotient Transition Systems," *IEEE Transactions on Automatic Control,* IEEE, Vol. 46, No. 9, 2001, pp. 1401-1410.
doi: 10.1109/9.948467

[36] Greenstreet, M. R., and Mitchell, I., "Integrating Projections," *Proceedings of the 1st International Conference of Hybrid Systems: Computation and Control*, Vol. 1386 of LNCS, Springer, Berkeley, CA, USA, 1998, pp. 159-174.
doi: 10.1007/3-540-64358-3_38

[37] Stursberg, O., and Krogh, B., "Efficient Representation and Computation of Reachable Sets for Hybrid Systems," *Proceedings of the 6th International Conference of Hybrid Systems: Computation and Control*, Vol. 2623 of LNCS, Springer, Prague, The Czech Republic, 2003, pp. 482-497.
doi: 10.1007/3-540-36580-X_35

[38] Bournez, K., Maler, O., and Pnueli, A., "Orthogonal Polyhedra: Representation and Computation," *Proceedings of the 2nd International Conference of Hybrid Systems: Computation and Control*, Vol. 1569 of LNCS, Springer, Berg en Dal, The Netherlands, 1999, pp. 46-60.
doi: 10.1007/3-540-48983-5_8

[39] Botchkarev, O., and Tripakis, S., "Verification of Hybrid Systems with Linear Differential Inclusions Using Ellipsoidal Approximations," *Proceedings of the 3rd International Conference of Hybrid Systems: Computation and Control*, Vol. 1790 of LNCS, Springer, Pittsburgh, PA, USA, 2000, pp. 73-88.
doi: 10.1007/3-540-46430-1_10

[40] de Leege, A.M.P., van Paassen, M.M., and Mulder, M., "A Machine Learning Approach to Trajectory Prediction," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA 2013-4782, Boston, MA, USA, 2013.
doi: 10.2514/6.2013-4782

[41] Enns, R., and Si, J., "Apache Helicopter Stabilization Using Neural Dynamic Programming," *Journal of Guidance, Control, and Dynamics,* Vol. 25, No. 1, 2002, pp. 19-25.
doi: 10.2514/2.4870

[42] Woodbury, T., Dunn, C., and Valasek, J., "Autonomous Soaring Using Reinforced Learning for Trajectory Generation," *Proceedings of the 52nd AIAA Aerospace Sciences Meeting*, AIAA 2014-0990, National Harbor, MD, USA, 2014.
doi: 10.2514/6.2014-0990

[43] Ferrari, S., and Stengel, R. F., "Classical/neural Synthesis of Nonlinear Control Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 3, 2002, pp. 442-448.
doi: 10.2514/2.4929

[44] Neidhoefer, J., and Krishnakumar, K., "Nonlinear Control Using Neural Approximators With Linear Control Theory," *Proceedings of the Guidance, Navigation, and Control Conference,* AIAA, A97-37041, New Orleans, LA, USA, 1997.
doi: 10.2514/6.1997-3536

[45] "Airplane Upset Recovery Training Aid – Revision 2" Flight Safety Foundation Report, November 2008, http://flightsafety.org/archives-and-resources/airplane-upset-recovery-training-aid [retrieved 6 May 2016].

[46] Kurzhanski, A.B., and Varaiya, P., *Dynamics and control of trajectory tubes*. Springer, 2014. Chap. 3
doi: 10.1007/978-3-319-10277-1

[47] Pearson, K., "On Lines and Planes of Closest Fit to Systems of Points in Space," *Philosophical Magazine,* Vol. 2 No. 11, 1901, pp. 559–572.
doi: 10.1080/14786440109462720.

[48] Jolliffe, I., ed., "Principal Component Analysis," *Series in Statistics.* Springer, 1986.
doi: 10.1007/b98835

[49] Pope, S.B., "Algorithms for Ellipsoids," Technical Report FDA-08-01, Cornell University, Ithaca, NY, 2008.

[50] Levenberg, K., "A Method for the Solution of Certain Non-Linear Problems in Least Squares," *Quarterly of Applied Mathematics,* Vol. 2, 1944, pp. 164-168.

[51] Marquardt, D.W., "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *Journal of the Society for Industrial & Applied Mathematics,* Vol. 11 No. 2, 1963, pp. 431-441.
doi: 10.1137/0111030

[52] Hagan, M.T., and Menhaj, M.B., "Training feedforward networks with the Marquardt algorithm," *IEEE transactions on Neural Networks*, Vol. 5 No. 6 1994, pp. 989-993.
doi: 10.1109/72.329697

[53] Shaqura, M., and Claudel, C., "A Hybrid Systems Approach to Airspeed, Angle of Attack and Sideslip Estimation in Unmanned Aerial Vehicles," *Proceedings of the International Conference on Unmanned Aircraft Systems*, IEEE, Denver, CO, USA, 2015, pp. 723-732.
doi: 10.1109/ICUAS.2015.7152355

[54] Sleight, D.W., and Muheim, D. M., "Parametric Studies of Square Solar Sails Using Finite Element analysis," *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, Palm Springs, CA, AIAA Paper 2004-1509, 2004.

[55] Taleghani, B. K., Sleight, D.W., Muheim, D. M., Belvin, K., and Wang, J. T., "Assessment of Analysis Approaches for Solar Sail Structural Response," 39th AIAA/ASME/SAE/ASEE Joint Propulsion conference and Exhibit, Huntsville, AL, AIAA Paper 2003-4796, 2003.

[56] Smith, S. W., Song, H., Baker, J. R., Black, J., and Muheim, D. M., "Flexible Models for Solar Sail Control," *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, Austin, TX, AIAA Paper 2005-1801, 2005.

[57] Li, Q, Ma, X., and Wang, T., "Reduced model for flexible solar sail dynamics," *Journal of Spacecraft and Rockets* Vol. 48, No. 3, 2014, pp. 446-453.

[58] Liu, J., Ciu, N., Shen, F., and Rong, S., "Dynamics of highly-flexible solar sail subjected to various forces." *Acta Astronautica* Vol. 103, 2014, pp. 55-72.

[59] Liu, J., Rong, S., Shen, F., Cui, N., "Dynamics and Control of a Flexible Solar Sail." Mathematical Problems in Engineering 2014, pp. 1-25.

[60] Liu, J, Ciu, N., Shen, F., Rong, S., and Wen, X., "Dynamic modeling and analysis of a flexible sailcraft." *Advances in Space Research,* Vol. 56 No. 4, 2015 pp. 693-713.

[61] Zhang, Jin, Zhai, Kun and Wang TianShu "Control of large angle maneuvers for the flexible solar sail" *Science China, Physics mechanics and astronomy*. April 2011 Vol.54 No.4, pp. 770–776
doi: 10.1007/s11433-011-4277-1

[62] Wen JJ, Desrochers AA. “An Algorithm for Obtaining Bang-Bang Control Laws,” *Journal of dynamic systems measurement, and control,* Vol. 109, No. 2, 1987, pp. 171-175.
doi:10.1115/1.3143835.

[63] Ruina, A.L., and Pratap, R.. “Tension Shear and Bending Moment,” Introduction to statics and dynamics. Oxford University Press, 2002.