UAV Path Planning for Maximum-Information Sensing in Spatiotemporal Data Acquisition

By:

Andreas Nordby Vibeto

andreanv@stud.ntnu.no

Supervisor: **Tor Arne Johansen**

Co-supervisor: João F. Fortuna

NTNU

January, 2017

Abstract

Fixed-wing Unmanned Aerial Vehicles are extensively used for ground observation as they allow for effective and precise ways for collecting information, but it has a downside with the quality of the information being dependent on the attitude of the aircraft. There is a coupling between the sensors and the aircraft's attitude states, which is usually decoupled by mounting the sensors to a gimbal which increases the weight of the payload. This paper investigates alternative control methods that minimizes the errors caused by the aircraft attitude states when sensors are fixed to the aircraft body. The control methods will be developed with the usage of a pushbroom hyperspectral camera in mind, and they will be simulated in Matlab on a model of the Aerosonde UAV. A course PD-controller using the rudder will be derived and simulated as a way of reducing roll angle during flight. A path planner that alters a predetermined ground track that is to be surveyed based on knowledge about the roll used when changing course will also be developed. The results show that the controller using rudder to correct the course significantly reduces roll and the corresponding lateral movement of the camera footprint. Some roll is still present, and the slower response of the rudder controller constraints the degree of curves in the path. The path generated by the path planner results in smoother roll angles, and the path stays within the camera's field of view through turns. When following the generated path the aircraft still causes lateral movement of the camera footprint which causes the points of interest to move away from the center of the camera's field of view. The methods developed here show that it is possible to reduce image errors caused by the UAVs attitude states, but neither of the methods are able to guarantee that the points of interest are covered.

Preface

This paper has been developed as the result of project work performed in my fifth year of study at the Norwegian University of Science and Technology (NTNU).

I would like to thank my supervisor Professor Tor Arne Johansen for guidance and help with understanding the problem at hand and possible solutions. Thanks are also due to PhD candidate Kristoffer Gryte for letting me use and giving me technical support for his implementation of the UAV model, and to PhD candidate João F. Fortuna for giving me an introduction to hyperspectral cameras.

Andreas Nordby Vibeto
Trondheim, December 21., 2016

Contents

1 Introduction

Unmanned Aerial Vehicles (UAV) are widely used in ground observation by equipping the UAV with different kind of sensors. While the use of UAV eases many cases of ground observation, there are some difficulties related to the attitude of the aircraft. A camera that is fixed to the UAV will be coupled with the UAVs states, so that any change in attitude will cause the camera view to shift away from the points of interest. As the height increases the error increases, which means that even small attitude changes will give a difference in what is intended to be observed, and what is actually observed by the camera.

Today, it is common to attach a gimbal with a camera to the UAV to decouple the attitude of the UAV from the camera. This way the attitude of the UAV will not cause any errors in the image so that the operator can focus solely on the operation of the aircraft. However, the fuel costs of an aircraft with a gimbal attached may increase because of added weight and less effective aerodynamics.

This paper will investigate methods to reduce image errors caused by the UAVs attitude, while also avoiding the extra costs associated with a gimbal. The control methods will be developed with the usage of a pushbroom hyperspectral camera that is fixed to the UAV in mind. The alternative flight control methods that aim to decrease the errors caused by the coupling between the camera and the attitude of the UAV will be simulated, and their effects on the image error will be tested.

2 Literature Review

The literature review will be performed with focus on two control methods: controllers that reduces roll during flight and path planners that alter the path with regards to roll and the ground path that is to be surveyed. The workings of hyperstpectral cameras will also be reviewed, and their usage with UAVs.

2.1 Course controller

The course of the aircraft is normally controlled by using the ailerons to roll the aircraft, with the resulting difference between the lift vectors of each wing causing the aircraft to turn. This strategy is the most common one used in larger, manned aircrafts as it causes little drag and it is comfortable for the passengers [1]. Since this strategy leads to problems when performing ground observations, alternative control strategies that reduces roll during course change exists.

2.1.1 Rudder as a course control surface

While the rudder is most commonly used to reduce the sideslip during flight, it can also be used to create sideslip which causes the aircraft to turn. The control method is a fairly common method to avoid roll during course change, and common for these controllers is that they use the ailerons too keep the wings level during flight.

A controller using this control strategy was developed by Thomas Fisher in his paper ”Rudder Augmented Trajectory Correction for Unmanned Aerial Vehicles to Decrease Lateral Image Errors of Fixed Camera Payloads” [2]. In the paper the term ’Rudder Augmented Trajectory Correction’ (RATC) is used for a controller using the rudder to change the course, and ’Aileron Only Trajectory Correction’ (AOTC) for controllers using the ailerons as the course control surface. The implemented controller was a PD-controller simulated on a model of the Aerosonde UAV, and results focuses on image error when using a fixed camera.

The simulations showed that the course error of the two controllers were matching, both with and without wind. And unsurprisingly, the results show that the AOTC controller had much more variations in roll while the RATC controller had much more variations in sideslip. Simulations also showed that the RATC controller used much more input to its control surfaces, up to 400% more than the AOTC controller.

The difference in image error between the two controllers was bigger than the difference in course error. The image error was measured as the distance from the camera center point on the ground to the desired ground path, and while image errors for the RATC controller stayed at about 20 m the AOTC controller had a RMS error over 300 m. Field tests gave the same results and prove that RATC is a good choice for reducing image errors.

A similar approach was taken by Ahsan, Rafique and Abbas [3], but a PID controller was used instead of a PD. The controller was created from a nonlinear model which had been linearized about a stable trim point, and the resulting rudder controller was compared with a controller using the aileron for heading control.

The simulations in this paper also show that when using the rudder as a control surface compared to aileron there is less overshoot and a lower steady state error. Bode plots of the two controllers show that the rudder based course controller has a gain margin of $-24.5dB$ and a phase margin of 87.1°, while the aileron based controller has a gain margin of $-25.7dB$ and a phase margin of 94°. This means that the two controllers have similar stability features.

Mills, Ford and Mejias refers to a rudder-based course controller as a 'skid-to-turn' controller [1]. They use an UAV to survey a linear infrastructure, and the desired course is calculated based on the images of the infrastructure. The control method is called Image-Based Visual Servoing (IBVS), and it creates a model of the identified structure as a straight line that the UAV will follow while ensuring that the infrastructure stays within the cameras field of view (FOV). The controller was implemented as a PID-controller.

The controller was simulated and compared to a controller that banks the UAV to turn, and the results matches the previous results: the bank-to-turn controller reduces the track error the fastest, but the skid-to-turn controller causes less error in the images. The controllers were also tested in wind with similar results. One thing worth noting is that when the skid-to-turn controller were to intercept the structure with tailwind it resulted in a significant overshoot. In the image plane however, the error was still smaller than for the bank-to-turn controller.

2.1.2 Course controller with constraints

Banking the aircraft may cause the camera's FOV to move away from the point of interest, but when controlled the aircraft can be banked without losing the point of interest. In the paper ”Low Altitude Road Following Constraints Using Strap-Down EO Cameras on Miniature Air Vehicle” [4] constraints put on the UAV's roll and above ground level (AGL) ensure that the point of interest still stays within the camera's FOV when banking. In the paper the UAV tracks a roadway in a similar manner as was done in [1], and the constraints are calculated with regards to the camera's horizontal field of view, the assumed road width and the expected turn angle of the road.

The system was tested by simulating how the UAV would follow two 90° turns without wind. The results show that the road was lost from the camera's FOV two times during the two turns. This happened because the system did not estimate the road's path well enough, and the paper argues that by pointing the camera forward the estimation can be improved.

2.2 Path Planner

In order to successfully and efficiently execute missions using UAVs, planning is essential. Path planners are used for this is reason, and there exists many different path planners depending on the situation. A path planner may, for example, be used to avoid controlled flight into surface in mountainous areas based on maps, or it can be used in search and rescue missions to calculate in which area it is most likely to find people. In this case the path planner will be used to generate a path that will ensure that the points of interest are not lost from the camera's FOV during flight.

2.2.1 Path Planner for Ground Observation

Many path planners today are based on the result Dubin presented in 1957 [6]: the shortest path between two points in a two dimensional space consists of two circular arcs connected by a straight line. It has also been shown that the same principle can be used in three dimensions [7].

Dubin's path for UAVs can be used in several situations, and Lugo-Cardenas, Flores, Salazar, and Lozano [8] has written a paper on how a Dubin's path generator can be used to search for a missing person within a given area. The path is generated by a path generator, which then transmits the path to the path-following strategy that controls the low-level autopilot. The autopilot is responsible for maintaining a constant altitude and constant airspeed, while the path generator includes a constraint to ensure it does not generate a path which requires the UAV to exceed its maximum turning rate. When the UAV finds a point of interest, a path that circulates the point is generated. The path generator is simulated, and shows that Dubin's path is a valid choice for UAV operation.

As mentioned the aircraft course can be controlled by using the rudder. This strategy for airplane control can also be used in path planning, as was done by Yokoyama and Ochi [9]. A path planner based on Dubin's path was created with skid-to-turn dynamics in mind and it was compared with a path created by an optimization algorithm. The results show that the Dubin's path algorithm always returned a feasible path that is quasi-optimal. The Dubin's path algorithm was fast, with a mean computational time of $61.9\mu s$, which the report concludes is fast enough for the algorithm to be used for online calculations.

2.3 Hyperspectral Imaging

The control methods developed in this paper will be developed with the use of a fixed hyperspectral, pushbroom sensor in mind. A hyperspectral sensor/camera makes it possible to accurately detect types of material from the UAV by sensing the wavelength of the received light.

2.3.1 Description

Hyperspectral imaging uses basics from spectroscopy to create images, which means that the basis for the images is the emitted or reflected light from materials [10]. The amount of light that is reflected by a material at different wavelengths is determined by several factors, and this makes it possible to distinguish different materials from each other. The reflected light is passed through a grate or a prism that splits the light into different wavelength bands, so that it can be measured by a spectrometer.

When using a hyperspectral camera for ground observation from a UAV, it is very likely that one pixel of the camera covers more than one type of material on the ground. This means that the observed wavelengths will be influenced by more than one type of material. This is called a composite or mixed spectrum [10], and the spectra of the different materials are combined additively. The combined spectra can be split into the different spectra that it is build up of by noise removal and other statistical methods which will not be covered here.

2.3.2 UAV ground observation

Hyperspectral imaging is already being used for ground observation from UAVs. Its ability to distinguish materials based on spectral properties means that it can be used to retrieve information that normal cameras are not able to. For example in agriculture it can be used to map damage to trees caused by bark beetles [11], or it can be used to measure environmental properties, for example chlorophyl fluorescense, on leaf-level in a citrus orchard [12].

Systems for ground observation with hyperspectral cameras can be very complex, which often leads to heavy systems. In [13], a lightweight hyperspectral mapping system was created for the use with octocopters. The purpose of the system is to map agricultural areas using a spectrometer and a photogrammetric camera, and the final ”ready-to-fly” weight of the system is 2.0 kg. The resolution of the final images made it possible to gather information on a single-plant basis, and the georeferencing accuracy was off by only a few pixels.

The tests were performed at a low altitude, maximum 120 m. While this was mainly because of local regulations, it also gave a benefit as there was less atmosphere disturbance in the measurements. The UAVs orientation data combined with surface models was used when recovering the positional data in the images. However, they found that externally produced surface models was not accurate enough as they do not take vegetation into consideration. For this reason they supplemented the existing surface models with information gathered during flight.

2.4 Summary of Literature Review

The literature review shows that the problem reviewed in this paper has already been asdressed, and that there already exists control methods that seeks to minimize image error. The review also shows that hyperspectral pushbroom cameras are being used for ground observation today and that these systems are moving from being large and heavy, to being reduced to a size more fitting for UAVs.

One additional point made in [2] that is worth noting is that the RATC controller will ease the flight plans for ground observing. When using AOTC controllers for ground observing, extra measures often have to be taken to ensure that the entire area of interest is covered by the camera. For a typical 90° turn this could be to fly past the turn, make complete circle in the opposite direction of the turn, and then continue on the path after the 90° bend. When using the RATC controller, the flight path length and time was reduced by about 80%, and it was estimated that the reduction in length and time will give a 75% reduction in energy usage. This means that even though the paper concluded the RATC used 400% more input than the AOTC, the RATC will save time and maybe energy for complicated paths with many turns.

3 Kinematics

In order to control the UAV with regards to where the camera is pointing, a kinematic model that maps the camera focus to the UAV position and attitude is needed. An overview of the kinematic model for the UAV will be given here, as well as the method used for calculating the camera footprint based on the UAVs attitude states.

3.1 UAV States

The position of the UAV will be given in reference frame {n} using the North East Down (NED) coordinate frame:

$$\boldsymbol{p}^{n}_{b/n} = \begin{bmatrix} N \\ E \\ D \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \tag{3.1}$$

The attitude of the UAV will be given as Euler-angles:

$$\boldsymbol{\Theta}_{nb} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \tag{3.2}$$

where ϕ represents the roll angle, θ the pitch angle and ψ the heading angle. The attitude angles have the corresponding angular velocities denoted p, q and r respectively. In all the simulations done in this paper there will be no wind, and it may therefore be assumed that the course angle χ is equal to the heading ψ.

3.2 Camera Position

The position of the camera center point is coupled with the attitude of the aircraft. Figure 1 shows how the position of the camera is affected by the attitude $\boldsymbol{\Theta}_{nb}$ in the

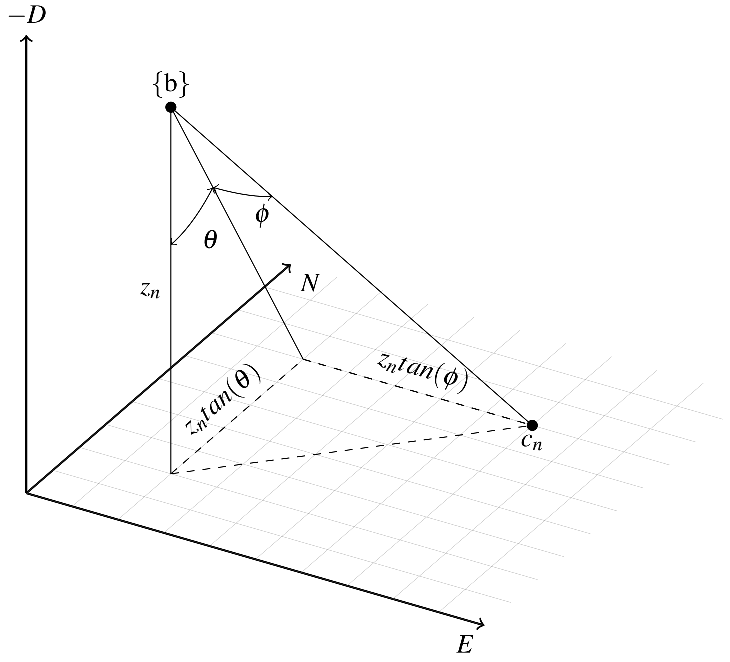

Figure 1: Illustration of how the aircraft attitude influence the camera position.

body frame {b}, and the height z_n in the NED frame {n}. This model assumes flat earth. The camera position in the body frame {b} is expressed as

$$\boldsymbol{c}_b^b = \begin{bmatrix} c_{x/b}^b \\ c_{y/b}^b \end{bmatrix} = \begin{bmatrix} z_n tan(\theta) \\ z_n tan(\phi) \end{bmatrix}. \tag{3.3}$$

In order to express the camera position $\boldsymbol{c}_b^b$ in {n}, the heading ψ of the aircraft must be taken into consideration. This is done by rotating the point $\boldsymbol{c}_b^b$ with the rotational matrix $\boldsymbol{R}_{z,\psi}$:

$$\boldsymbol{c}_b^n = \begin{bmatrix} c_{x/b}^n \\ c_{y/b}^n \end{bmatrix} = \boldsymbol{R}_{z,\psi}\boldsymbol{c}_b^b, \tag{3.4}$$

where:

$$\boldsymbol{R}_{z,\psi} = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.5}$$

The point $\boldsymbol{c}_b^n$ is not the actual position of the camera in {n} since it does not take the UAV's position into consideration. This is fixed by simply adding the UAVs position to $\boldsymbol{c}_b^n$:

$$\boldsymbol{c}^n = \begin{bmatrix} c_x^n \\ c_y^n \end{bmatrix} = \begin{bmatrix} x_n + c_{x/b}^n \\ y_n + c_{y/b}^n \end{bmatrix}. \tag{3.6}$$

3.3 Camera Angle of View

Since the camera isn't only focusing on one specific point, it can be useful describing the camera footprint of a pushbroom sensor as two extremities instead of one center point. Equation (3.3) can easily be changed to do this. Assuming the camera has an angle of view σ, the equation for the two extremities can be written as:

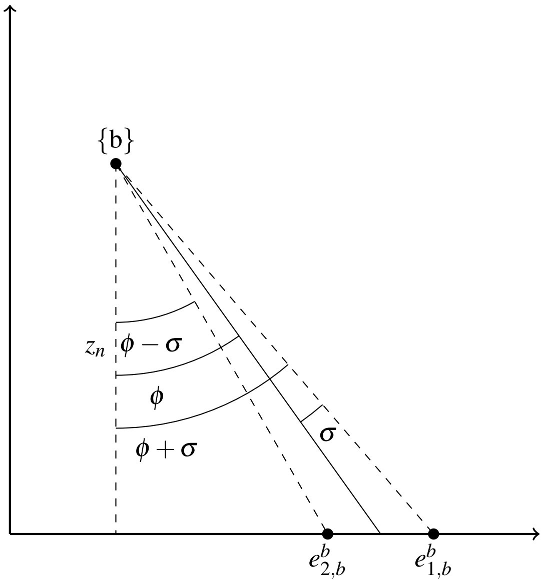

Figure 2: Illustration of how the field of view for a pushbroom sensor is calculated.

$$\boldsymbol{e}^b_{1,b} = \begin{bmatrix} e^b_{x/b} \\ e^b_{y_1/b} \end{bmatrix} = \begin{bmatrix} z_n tan(\theta) \\ z_n tan(\phi + \sigma) \end{bmatrix}, \ \boldsymbol{e}^b_{2,b} = \begin{bmatrix} e^b_{x/b} \\ e^b_{y_2/b} \end{bmatrix} = \begin{bmatrix} z_n tan(\theta) \\ z_n tan(\phi - \sigma) \end{bmatrix}. \tag{3.7}$$

The steps for translating the points to the NED frame are the same as in (3.4) and (3.6):

$$\boldsymbol{e}^n_b = \begin{bmatrix} e^n_{x/b} \\ e^n_{y/b} \end{bmatrix} = \boldsymbol{R}_{z,\psi} \boldsymbol{e}^b_b \tag{3.8}$$

$$\boldsymbol{e}^n = \begin{bmatrix} e^n_x \\ e^n_y \end{bmatrix} = \begin{bmatrix} x_n + e^n_{x/b} \\ y_n + e^n_{y/b} \end{bmatrix}. \tag{3.9}$$

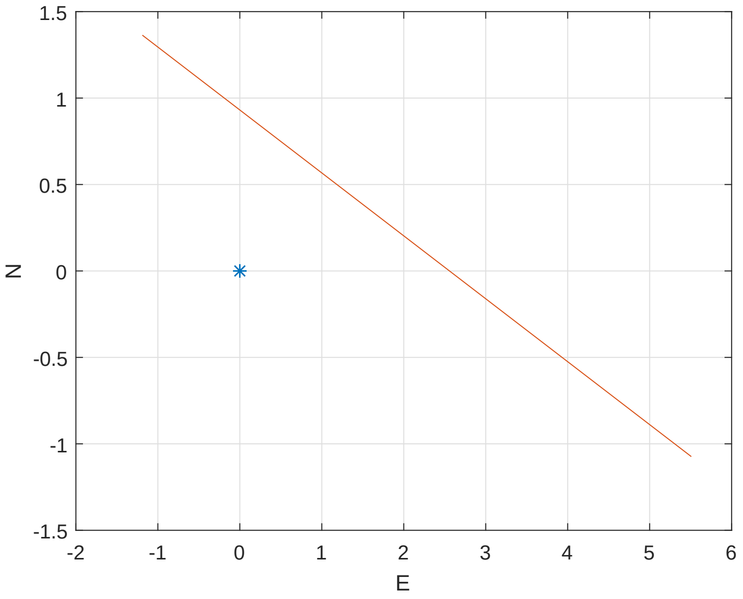

Figure 3: Graph showing the line the camera captures when the plane is positioned in the origin with an altitude of $10m$, and $\phi = -10$, $\theta = -5$ and $\psi = 20$. The field of view is 20°.

4 Controller

In this chapter the dynamic equations needed to describe how the course of the aircraft changes with the rudder will be presented, and the transfer function for the controller is derived.

4.1 Dynamics

In his paper, Fisher [2] uses the dynamic model given in Beard and McLain [5] to develop a controller that uses the rudder to change the heading. A similar controller will be used in this paper, and it will be derived by using the same steps that Fisher used.

To simplify the controller, it will be assumed that there is no wind and no sideslip β. These assumption will simplify the control problem since it can be assumed that $\chi = \psi$. It will also be assumed that the UAV is in trimmed, straight level flight, as this will simplify the system since the roll angle ϕ and pitch angle θ both can be assumed to be small.

The yaw dynamics for a UAV are (eq. 3.17, Beard and McLain [5])

$$\dot{r} = \Gamma_7 pq - \Gamma_1 qr + \Gamma_4 l + \Gamma_8 n \tag{4.1}$$

where l and n are the moments about the $\boldsymbol{i}^b$ and $\boldsymbol{j}^b$ axes respectively. The Γ equations describe the inertia of the aircraft and are expressed using elements of the inertia matrix $\boldsymbol{J}$.

The heading dynamic is expressed by the pitch rate q, the yaw rate r, and the attitude states $\boldsymbol{\Theta}_{nb}$ (eq. 3.3, Beard and McLain [5]):

$$\dot{\psi} = sin(\phi)sec(\theta)q + cos(\phi)sec(\theta)r. \tag{4.2}$$

As mentioned it is assumed that the aircraft is in trimmed straight level flight. By assuming small ϕ and θ, and also no pitch rate q, the heading dynamics can be simplified to:

$$\dot{\psi} = r, \tag{4.3}$$

which leads to:

$$\ddot{\psi} = \dot{r}. \tag{4.4}$$

The equation for the yaw dynamics (4.1) can now be written as

$$\ddot{\psi} = \dot{r} = \Gamma_4 l + \Gamma_8 n. \tag{4.5}$$

The moments l and n are the moments on the aircraft caused by the attitude states and rates, the sideslip β, and also the aileron deflection δ_a and the rudder deflection δ_r. These are given by equation 4.15 and 4.16 in Beard and McLain [5]:

$$l = \frac{1}{2}\rho V_a^2 Sb[C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{b}{2V_a}p + C_{l_r}\frac{b}{2V_a}r + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r] \tag{4.6a}$$

$$n = \frac{1}{2}\rho V_a^2 Sb[C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{b}{2V_a}p + C_{n_r}\frac{b}{2V_a}r + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r]. \tag{4.6b}$$

By continuing to follow Fishers [2] notation, equations (4.5) and (4.6a) can be combined to

$$\ddot{\psi} = \frac{1}{2}V_a^2 Sb[C_{r_0} + C_{r_\beta}\beta + C_{r_p}\frac{b}{2V_a}p + C_{r_r}\frac{b}{2V_a}r + C_{r_{\delta_a}}\delta_a + C_{r_{\delta_r}}\delta_r] \tag{4.7}$$

where

$$C_{r_0} = \Gamma_4 C_{l_0} + \Gamma_8 C_{n_0} \tag{4.8a}$$

$$C_{r_\beta} = \Gamma_4 C_{l_\beta} + \Gamma_8 C_{n_\beta} \tag{4.8b}$$

$$C_{r_p} = \Gamma_4 C_{l_p} + \Gamma_8 C_{n_p} \tag{4.8c}$$

$$C_{r_r} = \Gamma_4 C_{l_r} + \Gamma_8 C_{n_r} \tag{4.8d}$$

$$C_{r_{\delta_a}} = \Gamma_4 C_{l_{\delta_a}} + \Gamma_8 C_{n_{\delta_a}} \tag{4.8e}$$

$$C_{r_{\delta_r}} = \Gamma_4 C_{l_{\delta_r}} + \Gamma_8 C_{n_{\delta_r}} \tag{4.8f}$$

where the constants are craft-specific parameters, and

$$\Gamma_4 = \frac{J_{xz}}{J_x J_z - J_{xz}^2} \tag{4.9a}$$

$$\Gamma_8 = \frac{J_x}{J_x J_z - J_{xz}^2}. \tag{4.9b}$$

4.2 Controller Transfer Function

Since the controller is to use rudder input δ_r to alter the heading ψ, equation (4.7) can be rearranged to express these variables:

$$\ddot{\psi} = -a_{\psi_1}\dot{\psi} + a_{\psi_2}\delta_r + d_\psi \tag{4.10}$$

where

$$a_{\psi_1} = -\frac{1}{4}\rho V_a S b^2 C_{r_r} \tag{4.11a}$$

$$a_{\psi_2} = \frac{1}{2}\rho V_a^2 S b^2 C_{r_{\delta_r}} \tag{4.11b}$$

$$d_\psi = \frac{1}{2}\rho V_a^2 S b[C_{r_0} + C_{r_\beta}\beta + C_{r_p}\frac{b}{2V_a}p + C_{r_{\delta_a}}\delta_a]. \tag{4.11c}$$

a_{ψ_1} is chosen to be negative as this will ease later calculations (see (4.17a)). The Laplace transformation brings (4.10) to the form

$$\psi(s) = \frac{a_{\psi_2}}{s(s+a_{\psi_1})}\delta_r(s) + \frac{1}{s(s+a_{\psi_1})}d_\psi(s). \tag{4.12}$$

This equation show that the second term containing d_ψ acts as a disturbance for the controller. As shown in (4.11c), the inputs to this term are the sideslip β, roll rate p, and aileron deflection δ_a. Since the UAV is assumed to be in trimmed straight level flight and the controller will use the rudder to turn instead of roll it is already assumed that the roll rate p will be zero, as will the aileron deflection δ_a. During normal operation it cannot be assumed that no sideslip β will occur. However, any β is assumed to be small so that it can be removed from the controller equation. The final transfer function for the controller dynamics will then be

$$\frac{\psi(s)}{\delta_r(s)} = \frac{a_{\psi_2}}{s(s+a_{\psi_1})}. \tag{4.13}$$

In order to control the heading of the UAV with the help of the rudder, a controller must be added. The PD controller used here takes the form

$$\delta_r = ek_p + \dot{e}k_d \tag{4.14}$$

where e is defined as the error between the desired heading ψ_d and the measured heading ψ

$$e = \psi_d - \psi. \tag{4.15}$$

The transfer function between the desired heading and the measured heading is found by adding the controller to the transfer function between rudder and heading (4.13)

$$\frac{\psi}{\psi_d} = \frac{a_{\psi_2}k_p}{s^2 + (a_{\psi_1} + a_{\psi_2}k_d)s + a_{\psi_2}k_p}. \tag{4.16}$$

Since the transfer function is written in the form of a canonical second-order transfer function, the proportional gain k_p and the derivative gain k_d can be found by calculating the natural frequency ω_n and damping factor ζ. The final expressions for the gains will be

$$k_p = \frac{\omega_n^2}{a_{\psi_2}} \tag{4.17a}$$

$$k_d = \frac{2\zeta\omega_n - a_{\psi_1}}{a_{\psi_2}}. \tag{4.17b}$$

5 Path Planner

In an attempt to better track the ground path with regards to the camera, a simple path planner will be developed. The goal of the path planner is to alter the position of the aircraft so that the camera will be focused on the point of interest on the ground, regardless of the attitude of the aircraft. The path will first be generated as a Dubins path that later will be altered with regards to the kinematic model developed in chapter 3.

5.1 Dubins Path

As already presented in chapter 2.2.1, a Dubin's path consists of two circular arcs connected by a straight line [6]. The path generated here will only be in two dimensions, as it is assumed that the aircraft's autopilot will maintain a constant altitude. In order for a vehicle to follow a Dubins path, it must be possible to describe the kinematics of the vehicle as a Dubins vehicle [14]:

$$\dot{x}(t) = cos(\theta(t))u_1(t) \tag{5.1a}$$

$$\dot{y}(t) = sin(\theta(t))u_1(t) \tag{5.1b}$$

$$\dot{\theta}(t) = u_2(t) \tag{5.1c}$$

where u_1 is the linear velocity, u_2 is the angular velocity and θ is the course angle. For the kinematic model of the aircraft in this paper, u_1 equals to V, θ equals to ψ, and u_2 equals to r.

When generating a Dubins path there are four cases that need to be taken into consideration [5]. Depending on the start and end configuration a Dubins path can either start and end with a circle that the vehicle traces either clockwise or counterclockwise, and the four cases are the different combinations of start and end circles. An illustration of a simple Dubins path is shown in figure 4.

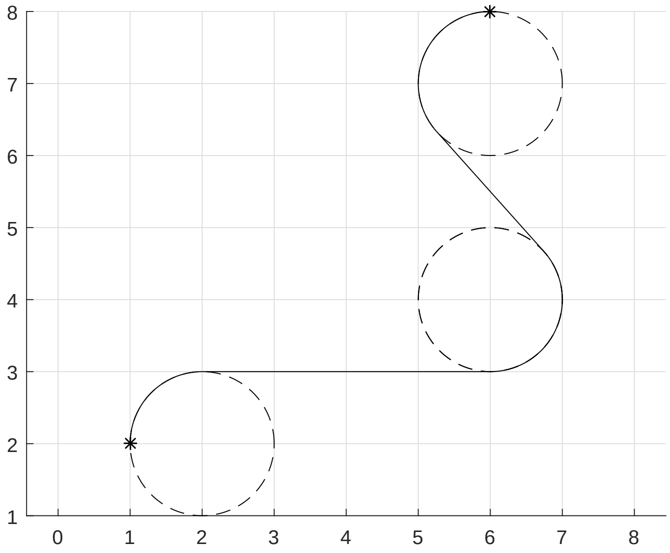

Figure 4: Illustration of a simple Dubins path.

In order to generate a Dubins path for this paper the algorithm presented in Beard & McLain was used (algorithm 7, [5]). The algorithm takes the start and end position, start and end course angle and radius of the circles as input. Based on these parameters the algorithm calculates the length of the path created by any of the four cases, and the case that gives the shortest path length is chosen. The outputs of the algorithm is the length of the path together with other parameters describing the path. The parameters that are calculated and returned by the algorithm are shown in 5.

The algorithm that generates the Dubins path only generates the path between two points, hence another algorithm is needed in order to generate the Dubins path involving several waypoints. For this another algorithm by Beard & McLain (algorithm 8, [5]) was used. This algorithm takes a list of waypoints together with the position of the aircraft and the desired turning radius as input. Based on the inputs the algorithm generates a Dubins path and then calculates where on the path the aircraft is located. It returns information about whether or not the aircraft should follow a straight line

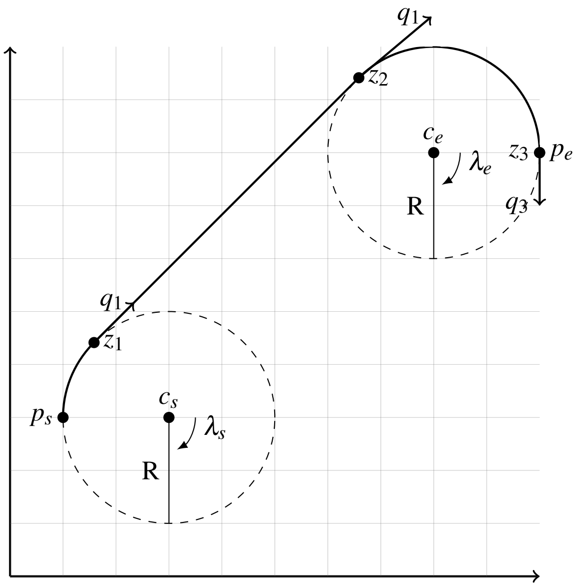

Figure 5: Illustration of the parameters returned by the Dubins algorithm.

or a circle, and the information needed to do this. This information is given to two algorithms that calculate the desired course that can be fed to the autopilot. These algorithms will be described in chapter 6.3.

5.2 Altering the original path

The Dubins path described in the previous section will represent the ground path that is to be surveyed. The problem with this path is that it will tell the aircraft to turn when it is just above the ground path, and the roll used to turn will cause the fixed camera to lose the points of interest from its field of view.

In order to compensate for the roll of the aircraft, the Dubins path will be altered so that the roll angle of the UAV will be compensated by the position of the UAV. The principle is shown in figure 6.

In order to compensate the roll the kinematic model developed in 3 can be used. For altering the path only the distance from the aircraft frame {b} to the camera point

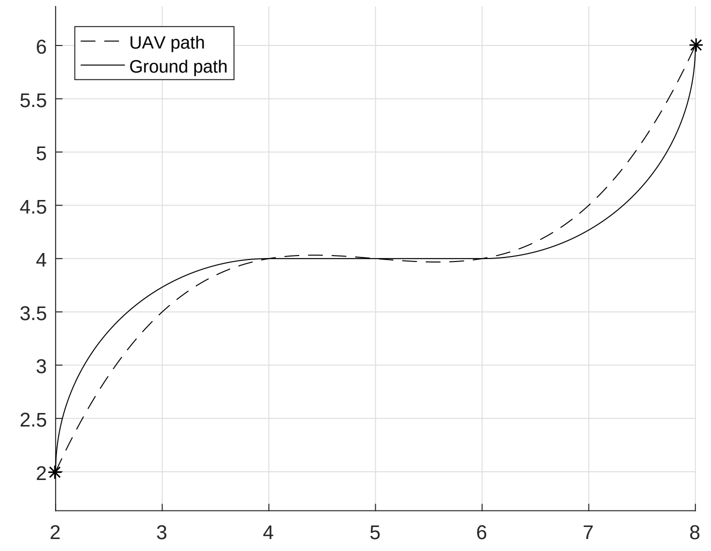

Figure 6: Illustration of the principle for altering the path.

caused by the roll is needed. This corresponds to $c^b_{y/b}$ from equation (3.3):

$$c^b_{y/b} = z_n tan(\phi). \tag{5.2}$$

$c^b_{y/b}$ only represents the distance the path is to be moved, and not the direction. The direction the path is to be moved is given by the course χ, and the direction should be perpendicular to χ as shown in figure 7. The coordinates for the new path $\boldsymbol{p}_d$ in the body frame {b} may then be written as

$$\boldsymbol{p}_{b,d} = \begin{bmatrix} x_{b,d} \\ y_{b,d} \end{bmatrix} = \begin{bmatrix} c^b_{y/b} sin(\chi) \\ -c^b_{y/b} cos(\chi) \end{bmatrix}, \tag{5.3}$$

and in the NED frame {n}:

$$\boldsymbol{p}_{n,d} = \boldsymbol{p}^n_{b/n} + \boldsymbol{p}_{b,d}. \tag{5.4}$$

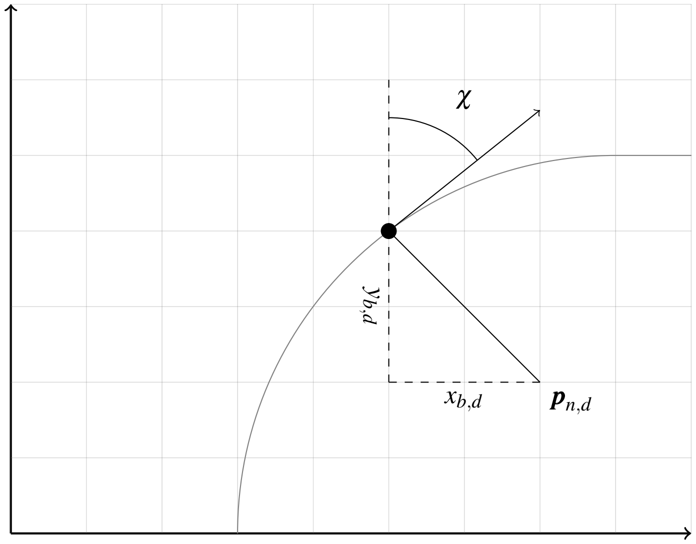

Figure 7: Illustration of the direction for altering the path.

Figure 8: The Aerosonde UAV [21].

6 Simulation

The simulations of the controller will be performed using Matlab and Simulink, simulating a model of the Aerosonde UAV.

6.1 Model

The model of the Aerosonde UAV is based on parameters and equations given by Beard & McLain [5]. The model was implemented by Gryte as a part of his master thesis [15].

The model is split into two parts, forces and aircraft dynamics. The forces module implements the equations for describing all the forces working on the aircraft, based on wind, aircraft states and the control inputs. The calculated forces are then sent to the aircraft dynamics module which calculates the new states of the aircraft based on the forces.

6.2 Autopilot

The autopilot used in the simulations has also been implemented by Gryte [15], and it is based on equations in Beard & McLain [5] as well. The autopilot have previously been used with a different UAV, and therefore the controller gains needed to be tuned in order to control the Aerosonde UAV. The controller loops are defined by relative damping factor ζ and natural frequency ω_n, and by implementing the control loops outside the model the parameters can be found separately. This makes a good starting point for further tuning.

6.3 Path Follower

Three different path followers will be used in this simulation. One of the three path followers will be used to navigate between waypoints by the rudder controller, while the other two will be used to follow the paths generated by the path planner.

The strategy used to follow Dubins path will be based on two algorithms presented by Beard & McLain [5]. The two algorithms are used to follow straight and curved line paths.

In order to follow straight line paths, the algorithm uses the position and course of the aircraft, the previous waypoint and the direction from the previous to the next waypoint as input. The previous waypoint and the direction to the next waypoint are given as output from the algorithm generating Dubins path described in chapter 5.1. The new course is calculated so that the aircrafts position will converge towards the original path.

The algorithm for following circular paths is based on following perfect circles. Therefore it takes center and radius of the circle, the direction to orbit the circle, and the current position and course of the aircraft. The desired course calculated here will also ensure that the aircrafts position converges to the circular path.

The altered path is based on the actual position of the UAV when following the Dubins path in the first simulation, and will therefore not consist of circular arcs and straight lines. Instead the path will be a continuous path which requires a different path follower. The path follower will be based on the principles of Line Of Sight (LOS) steering laws presented by Fossen [19].

Enclosure-based steering is a LOS principle that considers a circle with radius R enclosing the vehicle, which represents the LOS distance, shown in figure 9. Assuming that the radius is sufficiently large compared to the vehicle's distance from the path, the circle will intersect the path at two different points. One of the points will be in the direction of the vehicle, denoted x_{los} and y_{los}. This is the point the vehicle will be directed to, and the course to that point can be expressed as [19]:

$$\chi_d(t) = \text{atan2}(y_{los} - y(t), x_{los} - x(t)) \tag{6.1}$$

where $x(t)$ and $y(t)$ is the vehicle's current position. Using Pythagoras theorem, the points x_{los} and y_{los} can be found as:

$$[x_{los} - x(t)]^2 + [y_{los} - y(t)]^2 = R^2 \tag{6.2}$$

where R is the chosen LOS distance.

The last path follower used to navigate between waypoints is also based on an algorithm by Beard & McLain [5], and is a simpler algorithm than the one for Dubins path. It calculates the unit vectors between the waypoints and uses these to detect when the aircraft passes a waypoint. The algorithm returns the last waypoint visited and the direction for the next waypoint, which can be passed along to the algorithm to follow straight line paths already described. This will be used when simulating the controller in chapter 7.

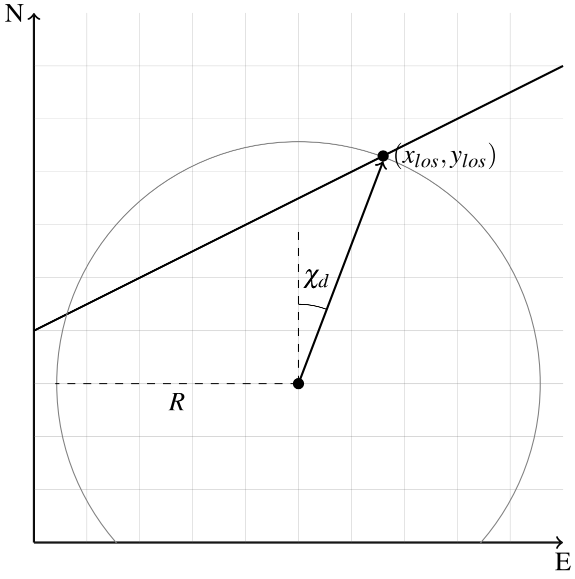

Figure 9: Illustration of enclosed line-of-sight path following.

7 Simulation of Controller

For the simulation the controller was implemented in Simulink, and it operates alongside the autopilot described in chapter 6. Since the controller will be used to control course using the rudder, the autopilot will be controlling all the other states.

7.1 Test Cases

The altitude flown at used when using a pushbroom sensor from an UAV for ground observation varies with what is being observed and the equipment used. When observing the vegetation, low-altitudes around 100 m is often used ([13], [16], [17]). However, altitudes as high as 1900 m has also been used to observe agricultural crops ([18]). In this paper simulations will be performed mainly at 100 m. The FOV for the camera will be set to 19° (approximately the same as in [13]). The cruise speed of the Aerosonde UAV is 35 m/s [5].

The controller has been tested in two different cases. The first case is a simple 45° turn in order to test the step response of the controller, and the response will be compared with a controller using ailerons to turn. In the second case the UAV will follow a ground path that is to be observed, and a comparison of the camera footprint made by the two controllers will be performed.

7.2 Results: Step Response

The step response of the controller is shown in figure 10, together with the step response of a controller using the aileron to turn. The course angle χ reaches the desired 45° (0.76 *rad*) after about 80 seconds. This is 20 seconds after the aileron-controller starts to stabilize at 45°, and while the aileron-controller is underdamped the rudder-controller is overdamped. Just after the step response has occurred, the course angle for the rudder-controller starts off in the slightly wrong direction before it starts increasing.

Figure 11 show that the pitch θ has only small variations when the step response occurs.

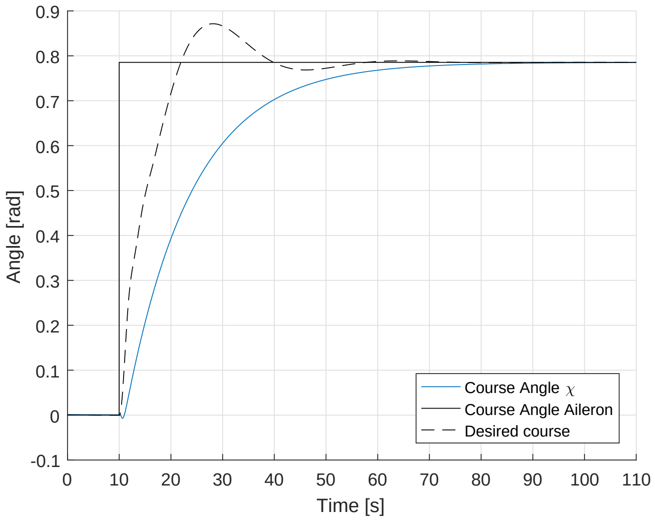

Figure 10: The step response of the course controller, compared to the step response of a course controller using the ailerons.

When the step occurs the roll ϕ changes rapidly back and forth. This is because the deflection of the rudder actuator also induces a roll on the aircraft, which can also be seen on the inputs shown in figure 12. At the time of the step both the aileron and the rudder increases rapidly, with an almost ”mirrored” response.

The bode-plot in figure 13 shows the frequency response of the rudder controller. The phase margin is 61.2° and since the phase never drops below $-180°$, this is a stable system (ch. 8.4.7, [20]). This tuning was achieved with ζ set to 0.6 and ω_n set to 3.5, which corresponds to a k_p of -0.3574 and a k_d of 0.2090.

7.3 Results: Path

The path flown and the corresponding camera footprint when the course is controlled using the rudder is shown in figures 14 and 15. The figures show that the ground path remains within the camera footprint when the ground path is straight, but the camera footprint shifts away from the ground path during turns, except from the first, gentle

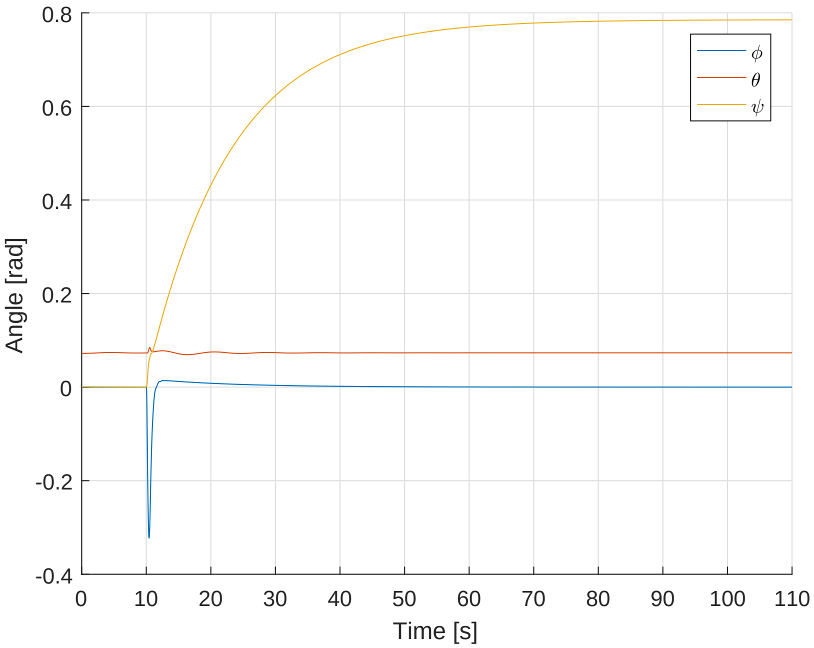

Figure 11: The attitude states of the UAV during the step response.

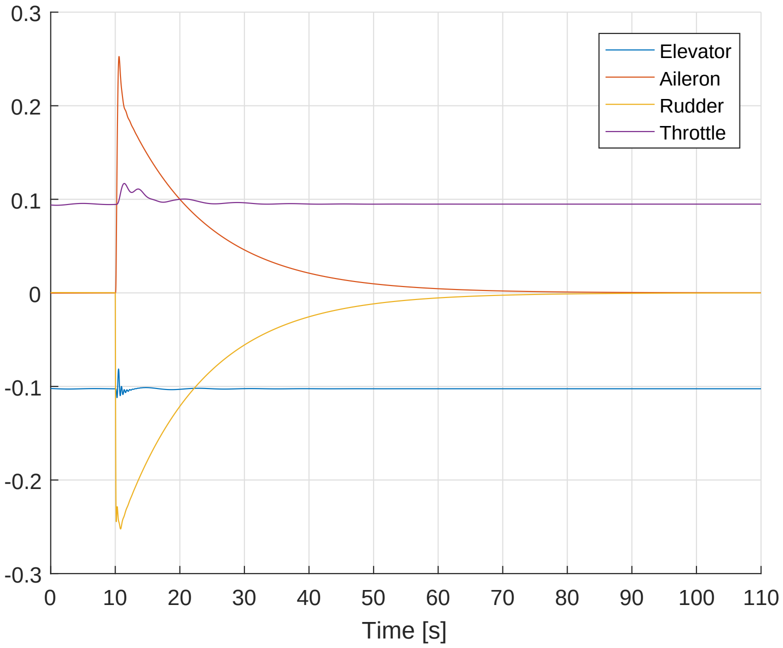

Figure 12: Input used during the step response. The input is given as a signal between ±1.

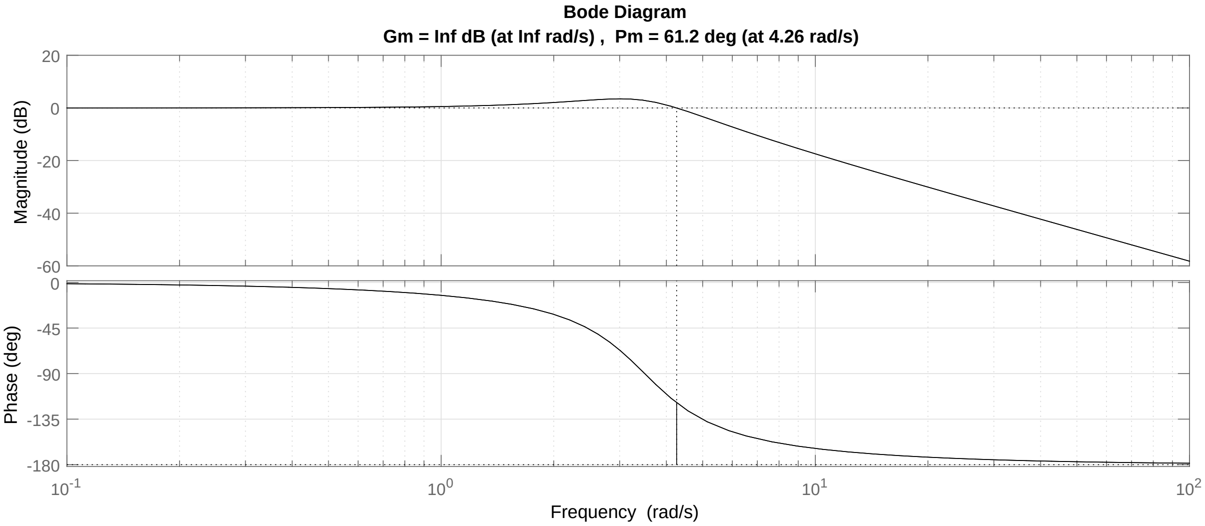

Figure 13: Bode plot the frequency response of the rudder controller.

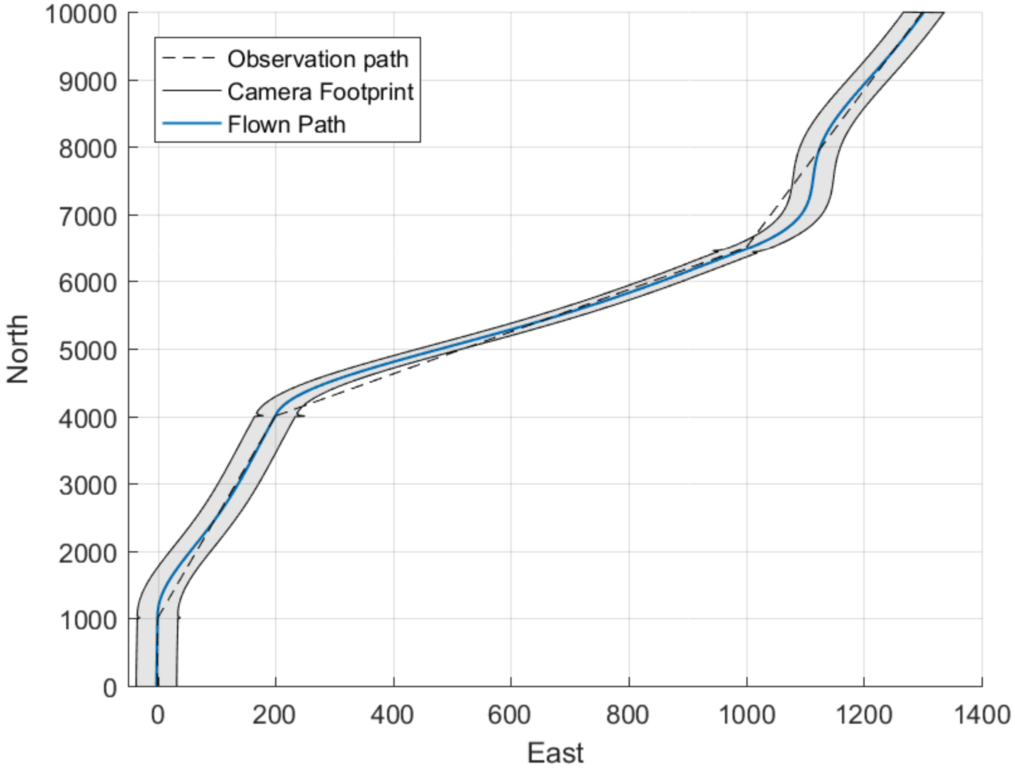

Figure 14: The path flown and the camera footprint when following the path using the rudder controller.

turn. The roll that is induced by the rudder that was seen in the step response, is visible on the camera footprint as well. At the beginning of each turn there is a small bump in the footprint, but none of the nudges moves the camera outside the observation path.

Figures 16 and 17 shows the flight of the UAV when using the aileron to alter the course. The observation path is within the camera footprint almost throughout the flight, but the edge of the camera footprint is close to the observation path at multiple times. At the beginning of every turn there is rapid shift in the camera footprint because of an abrupt roll change, and after every turn the roll of the aircraft oscillates back and forth before it settles.

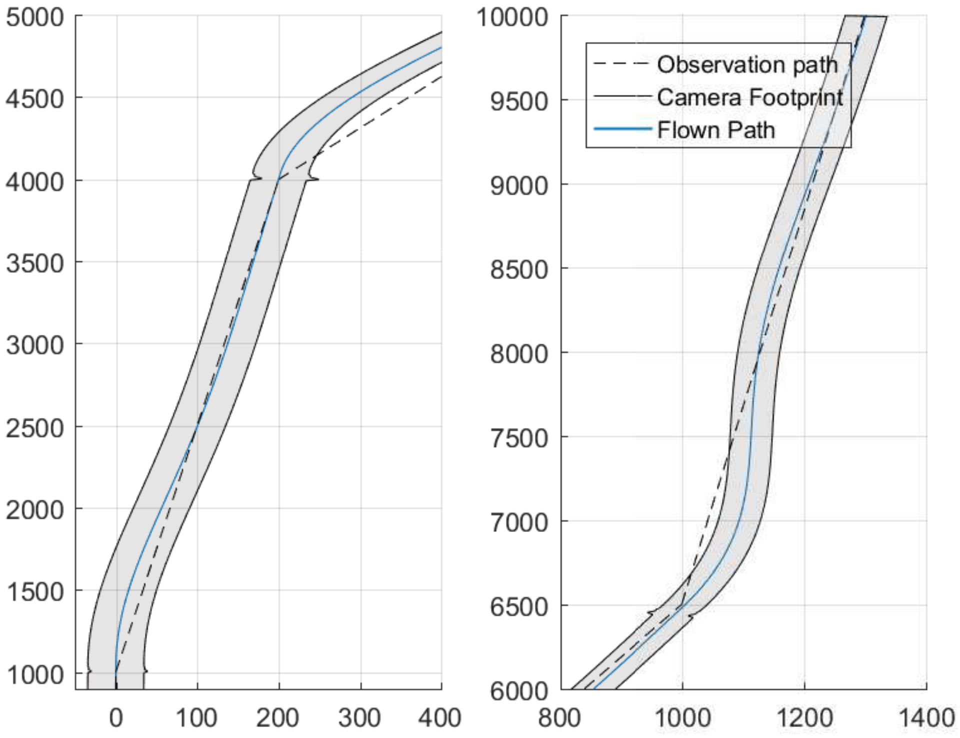

Figure 15: Detailed view of the path flown and the camera footprint when following the path using the rudder controller.

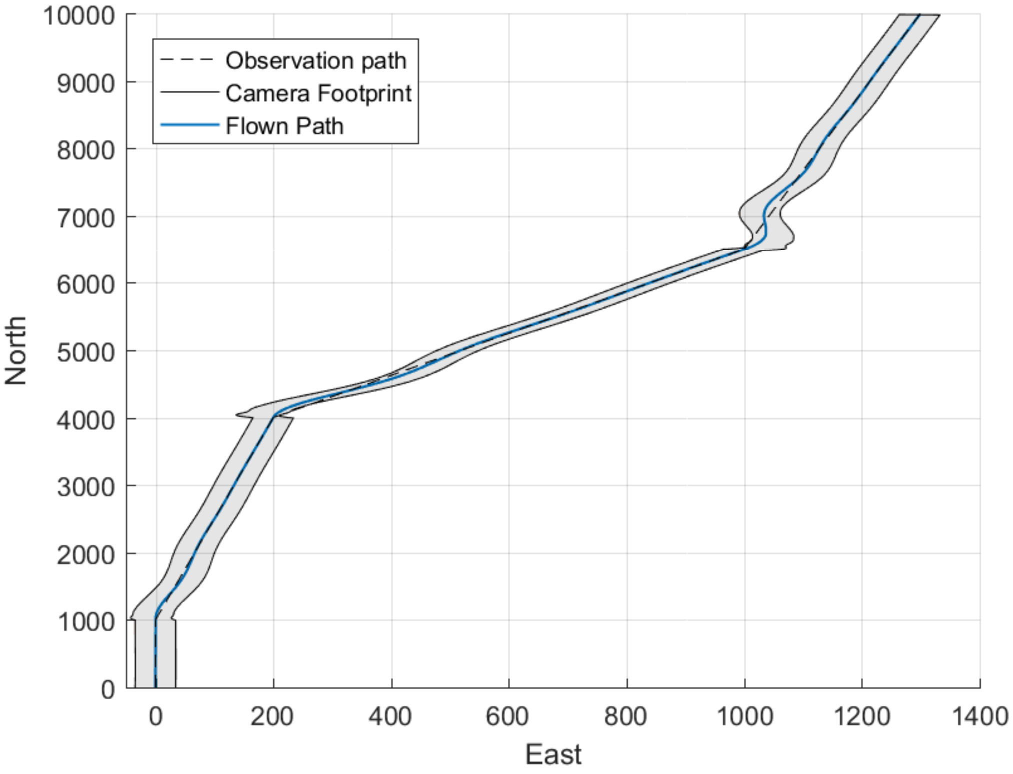

Figure 16: The path flown and the camera footprint when following the path using the aileron controller.

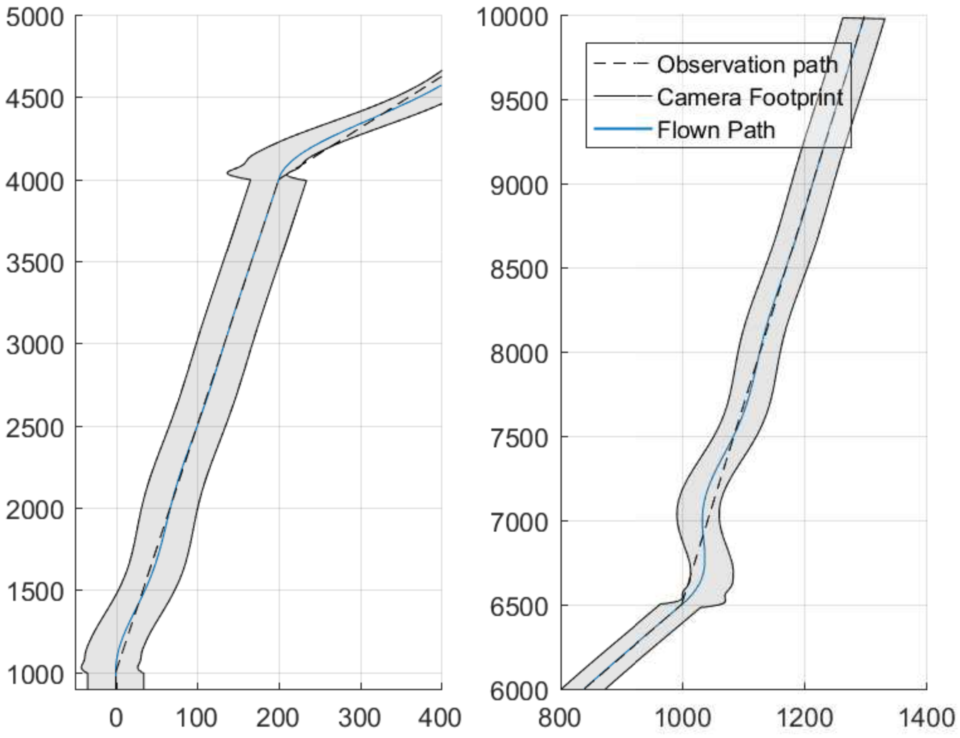

Figure 17: Detailed view of the path flown and the camera footprint when following the path using the aileron controller.

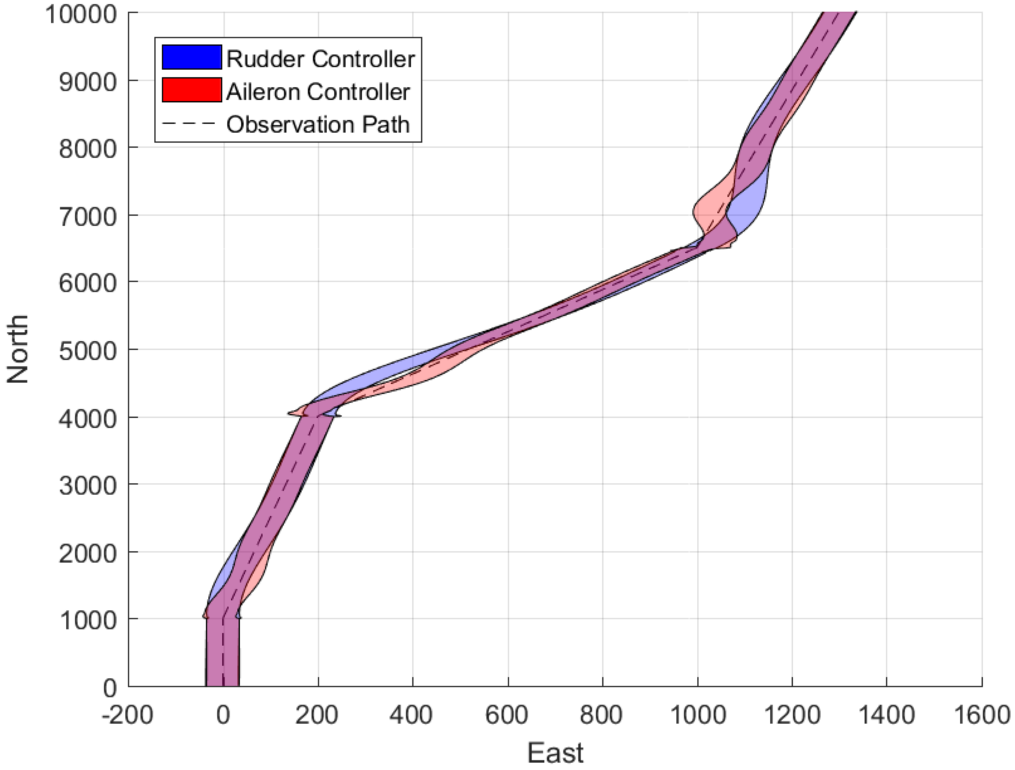

Figure 18: A comparison of the camera footprints of the two controllers.

The camera footprint made by both of the controllers is shown together in figure 18. The figure shows that the camera footprint of the aileron controller oscillates more than the footprint of the rudder controller, but it doesn't necessarily move further away from the observation path. At the beginning of the turns both of the controllers cause a jerk in the camera footprint in opposite directions. The bump of the aileron controller is bigger and remains longer than the bump for the rudder controller.

7.4 Discussion

The step response of the aileron controller was the best and the simulations of the path show that quick response is a favorable trait when tracking a path. However, the simulations show that a quick response is not the most important trait when performing ground observation. The quick response comes at the expense of lateral deviation in the camera footprint caused by the roll. The slower response of the rudder controller reduces the lateral changes in the camera footprint, but the slower response also causes the camera footprint to drift off the observation path because the controller doesn't manage to keep up with the path.

Both of the controllers causes abrupt changes in the roll of the aircraft, which again causes a lateral shift in the camera footprint. This is where the rudder controller performs better than the aileron controller. While the aileron controller is underdamped and the roll oscillates, the rudder controller makes a smoother course changes with the little nudge at the beginning of the turn as the only significant change in roll or pitch.

Even though the camera footprint made by the aileron controller covers the observation path for a larger portion of the path than the footprint made by the rudder controller, it does not necessarily result in better images. The camera footprint of the aileron controller has more lateral movement than the rudder controller has, and the images shot in sections with a lot of lateral movement is most likely not as sharp as they should be. The rudder controller do have some small, lateral jerks in the camera footprint as well, but it is possible that these can be made even smaller with better tuning of both the rudder controller and the aileron controller that keeps the wings level. The simulations show that the rudder controller performs much better in this area.

It is worth noting that the path simulated in this section was designed to test the rudder controller, and the two last turns was chosen so that the rudder controller would not be able to follow them with high enough precision so that the observation path stays within the camera footprint. The first turn can therefore be seen as a ”maximum turn angle” that the rudder controller produced in this paper can track. In addition the path planner used for both of the controllers was not designed for ground observation. This

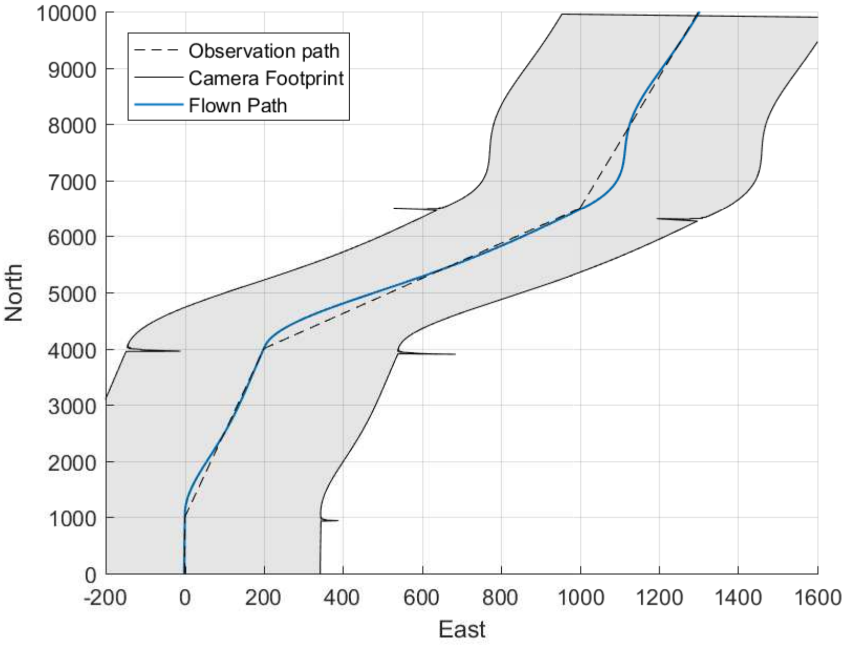

Figure 19: The camera footprint of the rudder controller flown at an altitude of 1000 m.

is because the path follower does not tell the controllers to change course for the next waypoint until the UAV has passed the current waypoint. In a situation where ground observation is the goal, the path follower should start changing course for the next waypoint before the current waypoint has been passed giving the controller more time to change the course.

The controllers were also simulated when flying at an altitude of 1000 m, and the resulting path and camera footprint for the rudder controller is shown in figure 19. The camera footprint is much larger, and therefore covers the observation path at all times. However, the lateral changes are still visible from this altitude and they are also amplified, meaning that increased altitude does not lead to more usable images. The results for the aileron controller were matching.

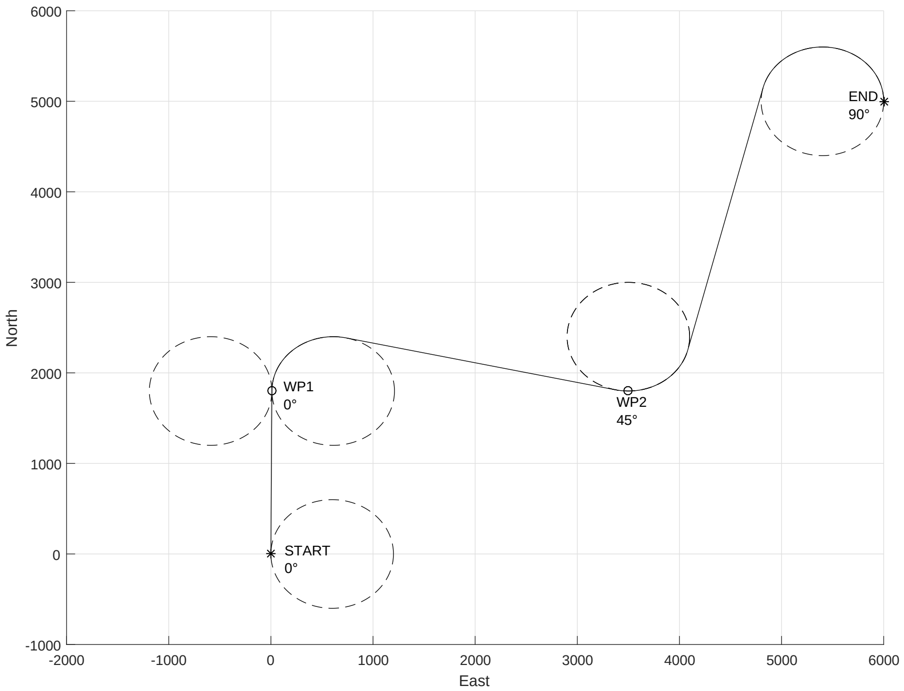

Figure 20: The path that will be simulated, with the direction associated with every waypoint.

8 Simulation of Path Planner

The same autopilot and model setup that was used for simulating the controller will be used when simulating the path planner. In addition a path follower will be used to give course commands to the autopilot during simulation, while the rest of the states will be controlled by the autopilot.

8.1 Simulation Setup

The Dubin's path that represents the ground track that is to be surveyed is shown in figure 20.

The radius of the circles in the Dubins path was chosen to 600 m, and is the same for every waypoint. If it is assumed that there is no wind and no sideslip, the equation for a coordinated turn becomes [5]:

$$R = \frac{V_g^2}{g\tan(\phi)}. \tag{8.1}$$

With R set to 600 m, and the airspeed equal to the groundspeed at 35 m/s, the corresponding roll ϕ is about 15°. This seems reasonable, as it is not expected that a UAV performing ground observation will be performing high dynamic maneuvers. The LOS distance was set to 200 m by trial and failure.

8.2 Result: Path Following

Figure 21 and figure 22 shows the result of the simulation when the UAV follows the generated Dubins path. The aircraft follows the observation path closely, but in turns it drifts slightly off and takes the outer turn.

Figure 23 shows the attitude states of the aircraft during the flight, and shows that the roll ϕ is just below 0.25 rad for each turn, which is about 15° as predicted previously. However, it can be seen that ϕ varies during the course changes, meaning that the aircraft is not doing a perfectly smooth turn. When using equation (5.2) to calculate how the path should be altered, these uneven turns will cause the path to be uneven as well. The altered path is shown together with the original flown path in figure 24. The figure shows some nudges due to the uneven turn, mainly at the beginning and end of the turns. These nudges correspond to the abrupt rise and fall of the roll angle ϕ shown in figure 23.

The altered path is supposed to counteract the slow and more constant changes in roll during a turn. The changes in roll caused by the uneven turn are much faster than the slow changes, which means that the unwanted changes in roll can be removed by passing the signal through a low-pass filter. The result of altering the path with the

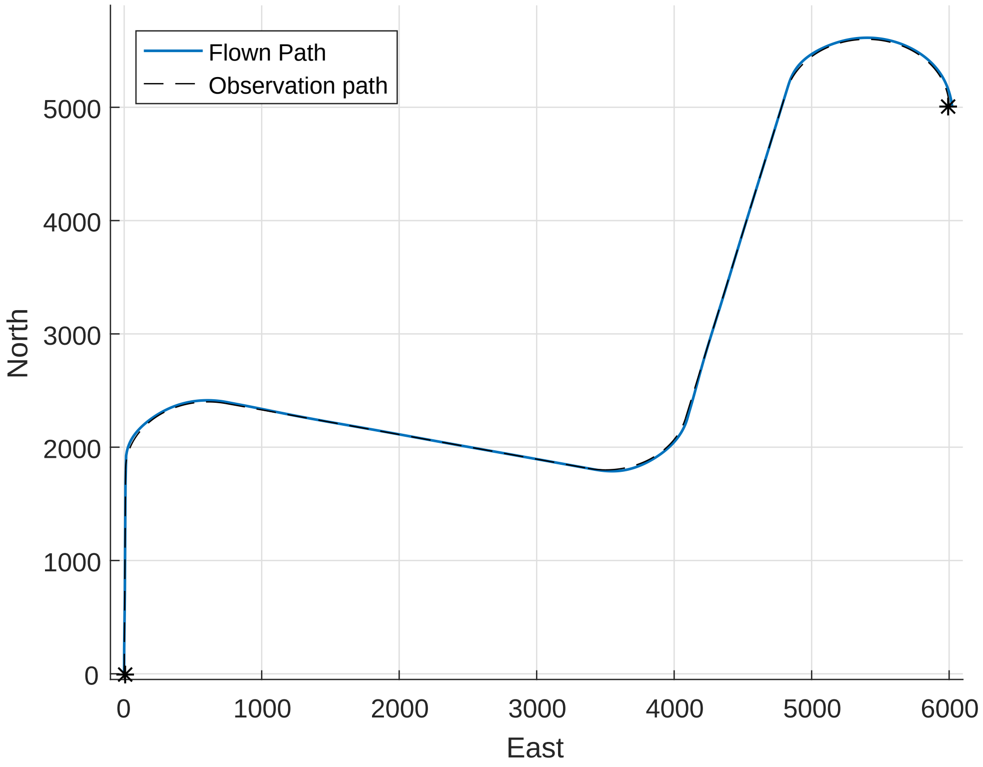

Figure 21: The path flown by the aircraft on the first run.

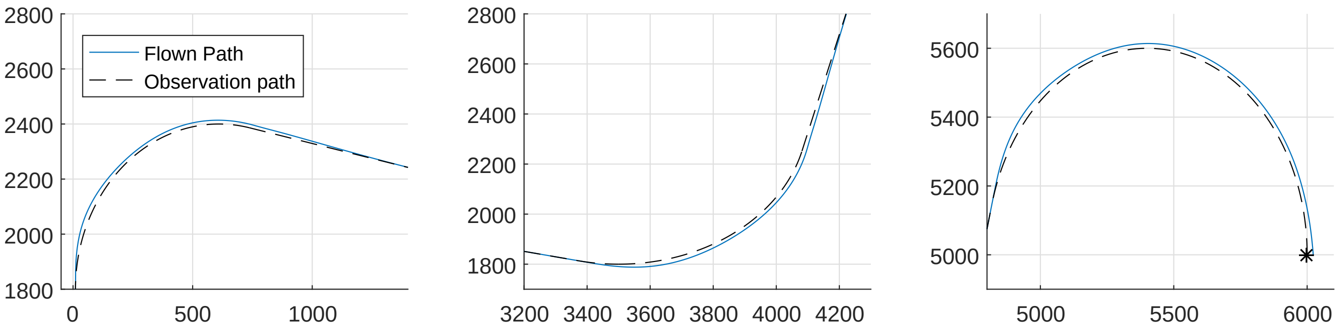

Figure 22: The path the aircraft took through the turns on the first run.

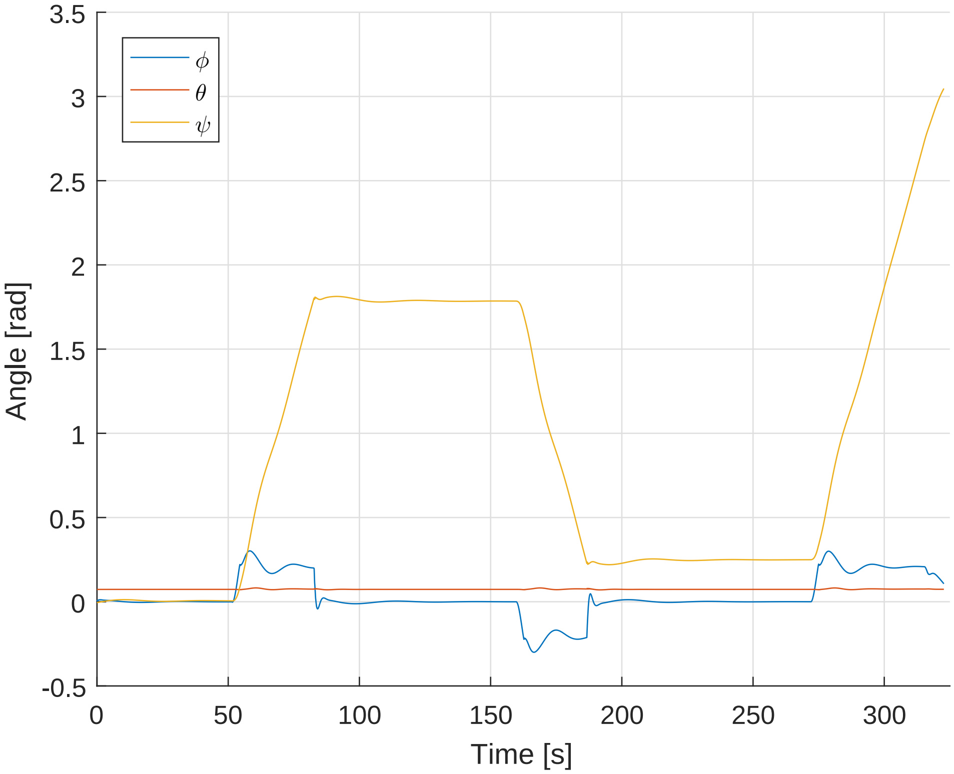

Figure 23: The attitude states of the aircraft during the first run.

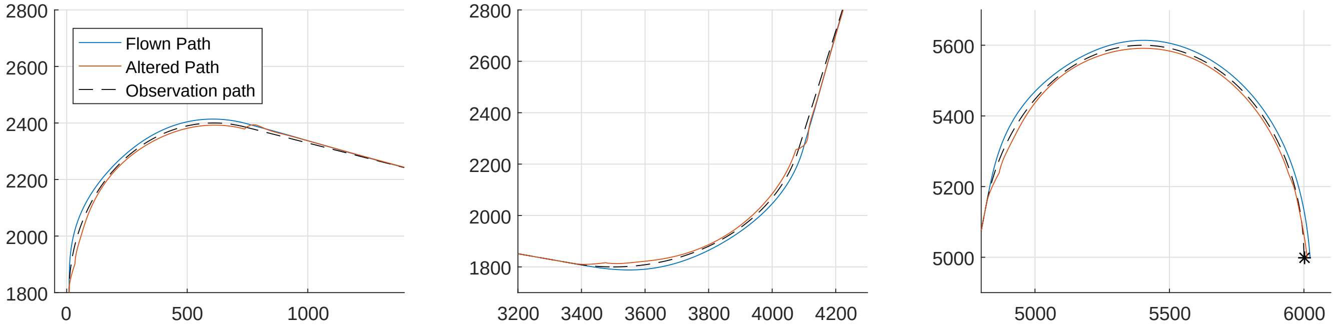

Figure 24: The figure shows how the altered path is compared to the flown path. Only the turns are shown as they are matching during the straight levelled sections.

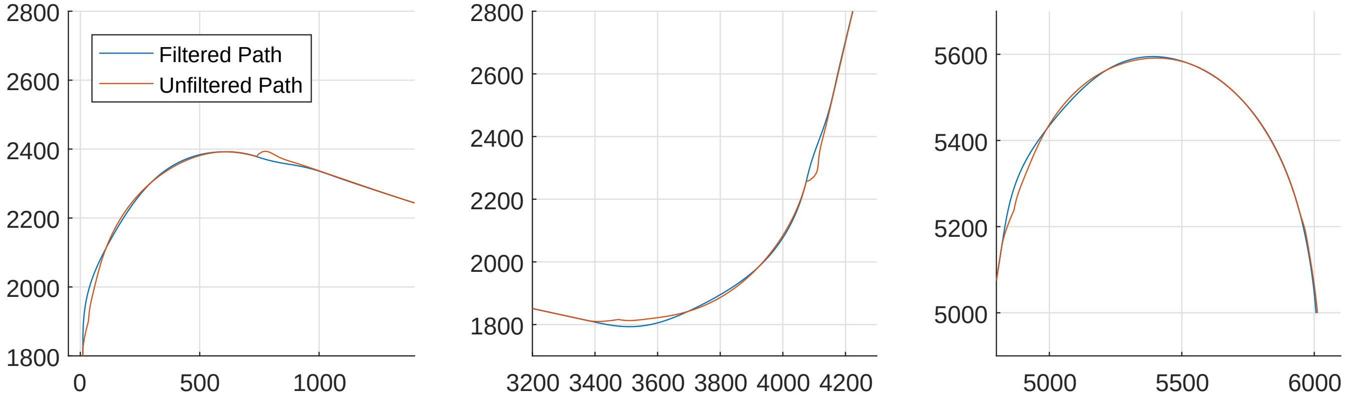

Figure 25: The path created by using the filtered signal for ϕ compared to using the unfiltered.

low-pass filtered ϕ is shown in figure 25, and it shows that the new path is smoother than the first one.

The result of the simulation when following the altered path is shown in figures 26, 27 and 28. It can be seen that instead of taking the outer path during turns, the aircraft now takes the inner turn which is what we want. The roll angle ϕ is still uneven during turns, but the changes in the roll angle happens slower than for the first run. The value of ϕ is about the same as for the first run.

8.3 Result: Camera Footprint

The camera footprint for the original path is shown in figures 29 and 30. While the camera footprint is positioned fairly straight above the observation path during the straight sections, the observation path is outside of the camera footprint during turns. The footprint drifts completely off the observation path in the beginning of the turn, but it catches up after the initial nudge. This matches the results from the previous section where the roll ϕ was the highest in the beginning of the turn. There is also a big change in ϕ at the end of the turn, which leads to a large movement of the camera footprint.

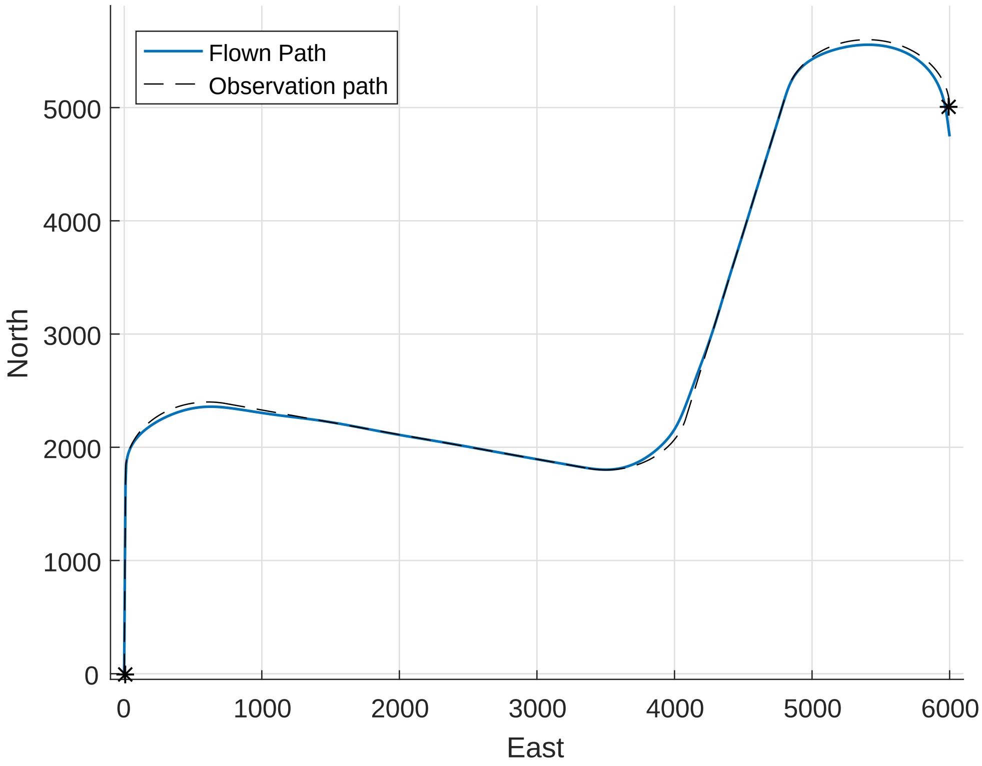

Figure 26: The path flown by the aircraft when following the altered path.

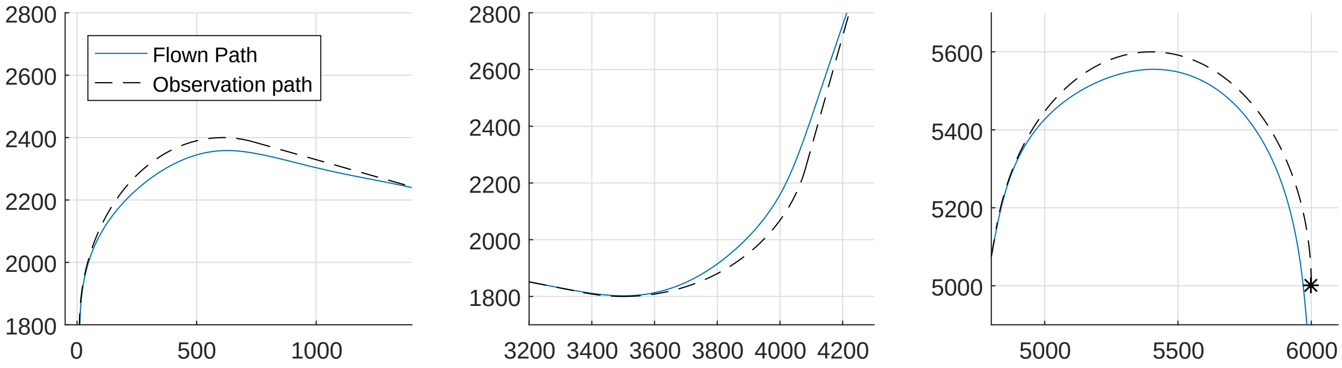

Figure 27: The path the aircraft took through the turns when following the altered path.

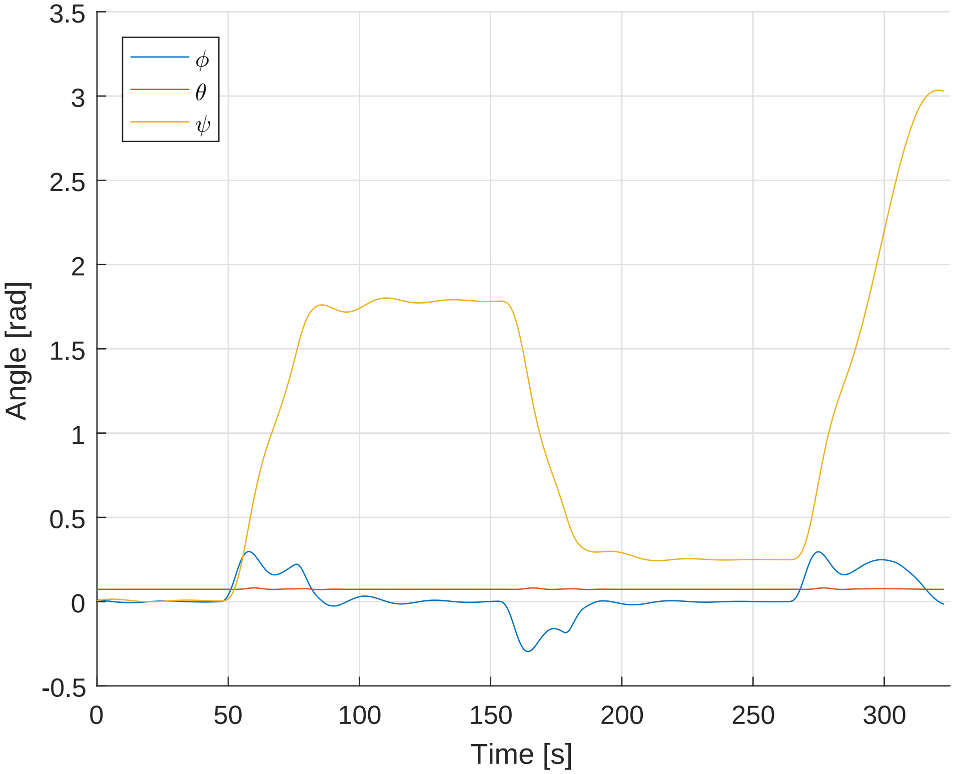

Figure 28: The attitude states of the aircraft when following the altered path.

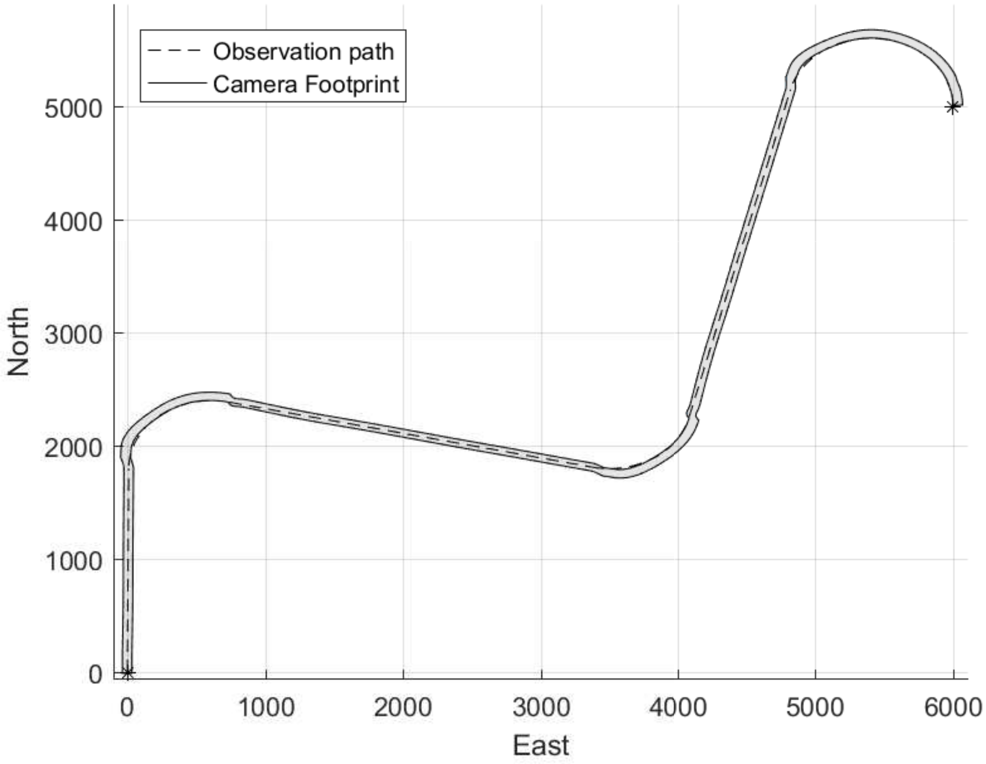

Figure 29: The camera footprint during simulation of the first path.

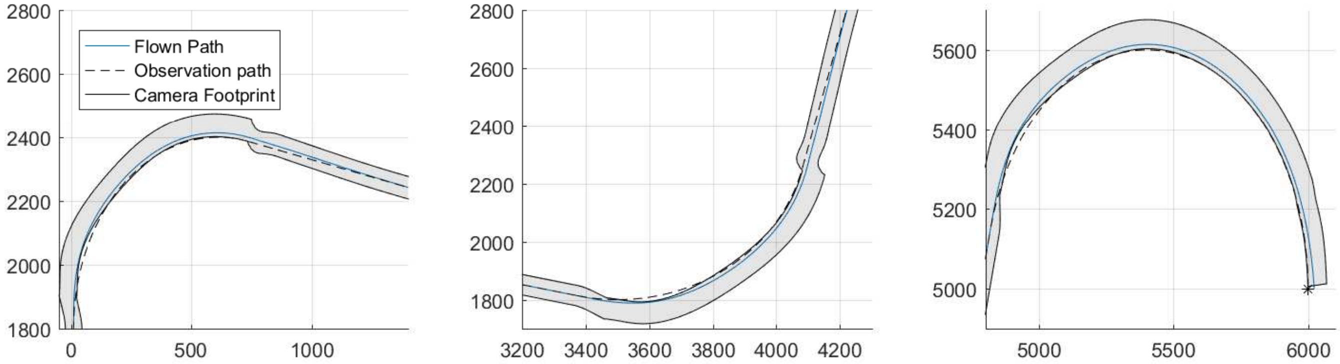

Figure 30: The camera footprint in turns during the simulation of the first path.

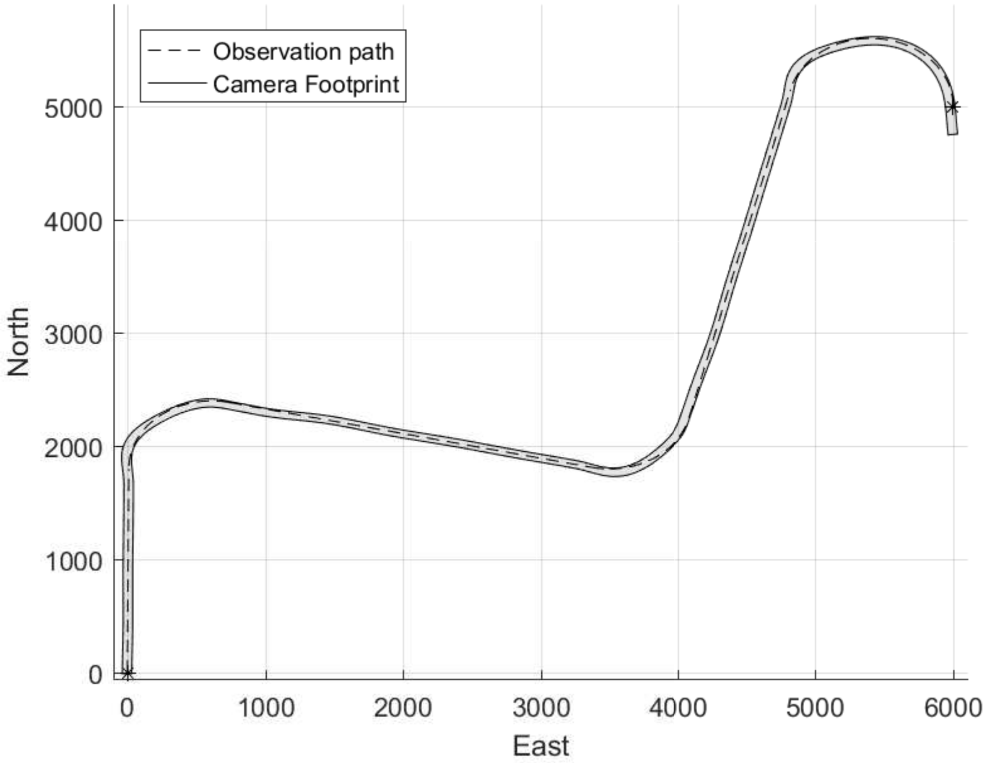

Figure 31: The camera footprint during simulation of the altered path.

For the altered path the camera footprint covers the observation path for the entire path as seen in figures 31 and 32. However, the camera footprint drifts a bit during turns so that the observation path is only visible on the edge of the footprint.

When the two paths are compared in figure 33 the biggest difference is that the when following the original path the camera takes the outer turn, and when following the altered path it takes the inner turn. The figures also show that the footprint of the altered path is smoother than the original. At the end of the two first turns for the original path the aircraft levels the wings quickly, as can be seen in figure 23, which leads to a rapid sideways shift in the camera footprint.

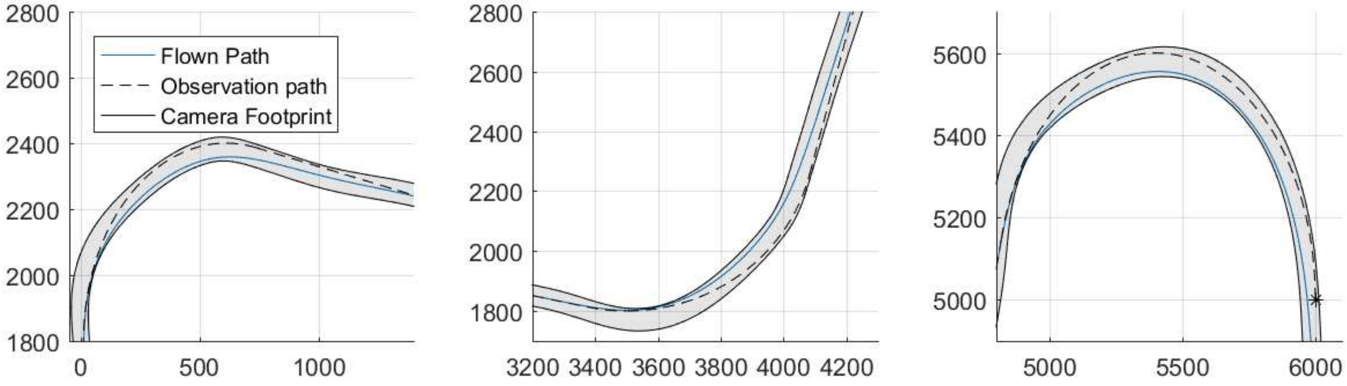

Figure 32: The camera footprint in turns during the simulation of the altered path.

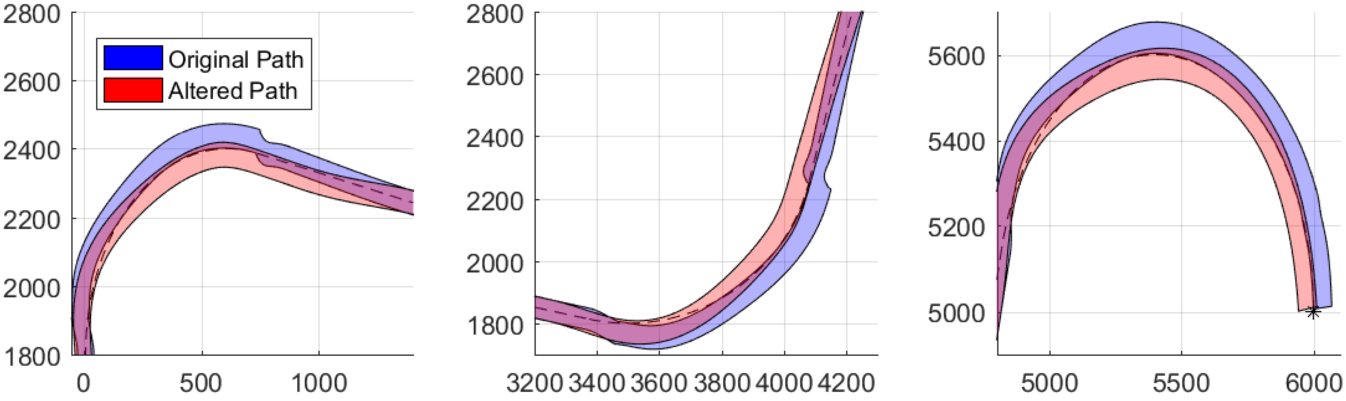

Figure 33: The camera footprint from both simulations shown together.

8.4 Discussion

The results of the simulation shows that the altered path does a better job observing the ground path than when trying to track the observation path. Tracking the ground path which in this case is to be observed leads to loss of information, which clearly is not acceptable in this types of mission.

Another problem encountered in these simulations when tracking the observation path is rapid sideways shifts caused by a rapid change in the roll. The change between images taken during these sections is big, which most likely makes it very difficult to make good use of these images, even though the observation path has been covered. The reason for these rapid changes in roll is because of how the Dubins path is tracked. The UAV switches from following a straight path with a fairly constant course to following a circular orbit with a constantly changing course in a short moment. This causes a jump in the desired course which again causes a rapid roll. This can most likely be minimized by using a different tuning of the course controller in the autopilot, but it is one of the shortcomings by tracking a Dubins path this way (Fossen [19], ch. 10.3.1).

There are also some shortcomings with the altered path used in this simulation. Even though it covered the observation path at all times, the observation path was not centered in the camera footprint. At some points during the flight, and especially in the beginning of the turns, the observation path is so close to the edge of the camera footprint that a small change in roll will most likely cause the camera to lose the ground track. One of the reasons this happens is because of the strategy used to alter the path. Since the path is only altered when there was roll present in the original flight path, the altered path will not start the turn any earlier than the original path did. Because the altered path cuts the corners of the original path but starts the turn at the same time, the turns of the altered path will be sharper than for the original, thus requiring a higher angle of roll. This is somewhat minimized by using a line-of-sight path follower that tracks the path ahead of the UAV, but the results show that this is not enough.

9 Conclusion

This paper has investigated different control methods that aim to minimize the effect the attitude of the aircraft has on a fixed camera used for ground observation. The kinematics related to camera position was investigated, and a method for describing the camera footprint of a hyperspectral pushbroom sensor was developed.

9.1 Findings

The controller simulated in this paper was a PD-controller that alters the course using the rudder to create sideslip, which again causes the course to change. The controller was tuned to perform well on a 45° step response, and a frequency analysis showed that the controller was stable. The controller was then simulated to follow a path of waypoints connected by straight lines, and its performance was compared to a course controller using aileron to change the course. The simulations show that the rudder controller causes less lateral movement of the camera footprint, but the slower response caused the rudder controller to drift further off the track than the aileron controller did. The rudder controller also induces some unwanted roll effects at the beginning of a turn.

The path planner developed and tested in this paper is a simple planner that alters a Dubins path which represents the ground track that is to be surveyed. The UAV was first simulated tracking the original Dubins path. Then the path was altered with regards to the roll used in the first run, so that the position of the new path would counteract the roll used. The simulations show that while the camera footprint drifted off the observation path when following the original Dubins path, it did not drift off when following the altered path. The roll used when tracking the altered path was also smoother than for the original path, causing less rapid lateral movement in the camera footprint. However, the observation path was not in the center of the camera footprint throughout the flight, and there is no guarantee that this method works in all cases.

9.2 Future Work

The controller needs more testing before it can be implemented and used with an actual UAV. The small jerks that occur at the beginning of turns should be possible to remove with some more tuning of both the rudder and the aileron controller. The performance of the rudder controller when exposed to wind should also be investigated.

For the path planner, future work should include investigation of methods that can guarantee to a higher degree that the observation path will stay centered in the camera footprint throughout the flight. The study should include optimization methods that can generate a path based on the ground track that is to be surveyed and knowledge about the UAV's dynamics.

The effect of pitch on the camera footprint was modelled in this paper, but not investigated. While all the simulations here were performed on a fixed altitude, methods that allow for altitude change during flight while still keeping the points of interest within the camera's FOV should be investigated.

It should also be investigated how the different UAV attitudes and movements influence the quality of the images captured by the pushbroom spectrometer. The roll of the UAV will create a wider camera footprint, which again means that the pixels closest to the UAV captures a smaller area than the pixels far away. Whether this is a problem or not, and how an eventual problem can be handled, should be further investigated.

References

[1] Mills, S., Ford, J. J., Mejias, L. (2011) *"Vision Based Control for Fixed Wing UAVs Inspecting Locally Linear Infrastructure using Skid-to-Turn Maneuvers"*, Australian Research Centre for Aerospace Automation (ARCAA), Queensland University of Technology, Australia

[2] Fisher, Thomas M. (2016) *"Rudder Augmented Trajectory Correction for Unmanned Aerial Vehicles to Decrease Lateral Image Errors of Fixed Camera Payloads"*, All Graduate Theses and Dissertations, Paper 4751, USA

[3] Ahsan, M., Rafique, H., Abbas, Z. (2012) *"Heading Control of a Fixed Wing UAV Using Alternate Control Surfaces"*, National University of Sciences and Technology, Islamabad, Pakistan

[4] Egbert, J., Beard, R. W. (2007) *"Low Altitude Road Following Contraints Using Strap-down EO Cameras on Miniature Air Vehicles"*, Proceedings of the 2007 American Control Conference, New York City, USA (IEEE)

[5] Beard, R. W., McLain, T. W. (2012) *"Small Unmanned Aircraft: Theory and Practice"*, Princeton University Press, United Kingdom

[6] Dubins, L. E (1957) *"On curves of minimal length with a constraint on average curvature, and with prescribed intial and terminal positions and tangents"*, American Journal of Mathematics, vol. 79, no. 3, pp. 497-516

[7] Owen, M., Beard, R. W., McLain, T. W. (2014) *"Implementing Dubins Airplane Paths on Fixed-wing UAVs"*, Handbook of Unmanned Aerial Vehicles, ed. Kimon P. Valavanis, George J. Vachtsevanos, Springer Verlag, Sectopn XII, Chapter 68, p. 1677-1702

[8] Lugo-Cárdenas, I., Flores, G., Salazar, S., Lozano, R. (2014) *"Dubins Path Generation for a Fixed Wing UAV"*, International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, USA

[9] Yokoyama, N., Ochi, Y. (2009) *"Path Planning Algorithms for Skid-to-Turn Unmanned Aerial Vehicles"*, Journal of Guidance, Control, and Dynamics, Vol. 32, No. 5

[10] Smith, Randall B. (2012) *"Introduction to Hyperspectral Imaging"*, MicroImages, Inc.

[11] Näsi, R., Honkavaara, E., Lyytikäinen-Saarenmaa, P., Blomqvist, M., Litkey, P., Hakala, T., Viljanen, N., Kantola, T., Tanhuanpää, T., Holopainen, M. (2015) *"Using UAV-Based Photogrammetry and Hyperspectral Imaging for Mapping Bark Beetle Damage at Tree-Level"*, Remote Sensing (2015), 7, 15467-15493

[12] Zarco-Tejada, P. J., González-Dugo, V., Berni, J. A. J. (2012) *"Fluorescence, temperature and narrow-band indices acquired from a UAV platform for water stress detection using a micro-hyperspectral imager and a thermal camera"*, Remote Sensing of Environment 117 (2012), 322-337

[13] Suomalainen, J., Anders, N., Iqbal, S., Roerink, G., Franke, J., Wenting, P., Hünniger, D., Bartholomeus, H., Becker, R., Kooistra, L. (2014) *"A Lightweight Hyperspectral Mapping System and Photogrammetric Processing Chain for Unmanned Aerial Vehicles"*, Remote Sensing (2014), 6, 11013-11030

[14] Yong, C., Barth, E. J. (2006) *"Real-time Dynamic Path Planning for Dubins' Nonholonomic Robot"*, Proceedings of the 45th IEEE Conference on Decision & Control, San Diego, CA, USA, December 13-15

[15] Gryte, K., Fossen, T. I. (2015) *"High Angle of Attack Landing of an Unmanned Aerial Vehicle"*, Norges Teknisk-Naturvitenskapelige Universitet, 2015

[16] Lelong, C. C. D., Burger, P., Jubelin, G., Roux, B., Labbé, S., Baret, F. (2008) *"Assessment of Unmanned Aerial Vehicles Imagery for Quantitative Monitoring of Wheat Crop in Small Plots"*, Sensors 2008, 8, 3557-3585

[17] Ramirez-Paredes, J., Lary, D. J., Gans, N. R. (2015) *"Low-altitude Terrestrial Spectroscopy from a Pushbroom Sensor"*, Journal of Field Robotics 33(6), 837-852 (2016), Wiley Periodicals, Inc.

[18] Asmat, A., Milton, E. J., Atkinson, P. M. (2011) *"Empirical correction of multiple flightline hyperspectralaerial image mosaics"*, Remote Sensing of Environment 115 (2011), 2664-

[19] Fossen, Thor I. (2011) *"Handbook of Marine Craft Hydrodynamics and Motion Control"*, John Wiley & Sons, Ltd.

[20] Balchen, J. G., Andreasen, T., Foss, B. A. (2003) *Reguleringsteknikk*, Institutt for Teknisk Kybernetikk, NTNU, Trondheim

[21] Stevenson, B. (21. October, 2014) *"Textron Awaits FAA Authorisation for Aerosonde Flight"* (Online, image), https://www.flightglobal.com/news/articles/textron-awaits-faa-authorisation-for-aerosonde-fligh-405027/