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

By:
Andreas Nordby Vibeto
andreanv@stud.ntnu.no

December, 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, so 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 gimbal. The control methods will be developed with the usage of a hyperspectral camera that is fixed to the UAV in mind. 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 effect on the image error will be tested.

2 Literature Review

The literature review will be done with focus on two control methods: a controller that reduces roll during flight, and a path planner that alters the path with regards to roll and the ground points that are to be observed. 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 in larger, manned aircrafts as it is causes little drag and it is comfortable for the passengers [1].

Banking to change the course of the aircraft leads to problems when performing ground observation, which is solved by decoupling the roll from the sensors by using a gimbal. In order to avoid the extra payload that the gimbal is, there exists different control strategies to reduce roll during course change.

CLEAN UP TERMINOLOGY REGARDING HEADING/COURSE

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 wich causes the aircraft to turn. The control method is a fairly common method to avoid roll during course change. KILDE HADDE VÆRT FINT. Common for these controllers is that they use the ailerons too keep the wings level during flight.

A controller using this control strategy has been created by Thomas Fisher in his paper "Rudder Augmented Trajectory Correction for Unmanned Aerial Vehicles to Decrease Lateral Image Errors of Fixed Camera Payloads" [2]. Here 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 for the two controllers were matching, both with and without wind. An unsurprisingly, the results show that the AOTC controller had much more changes in roll and the RATC controller had much more changes in sideslip. The biggest difference was that the RATC controller used much more input to its control surfaces, up to 400% more than the AOTC controller. BETTER WORD FOR "MORE CHANGE"?

When comparing the image error for the two controllers there was a big difference in performance. The image error was measured as the distance from the camera centre point to the desired ground path, and while image errors for the RATC controller stayed at about 20 m the AOTC had a RMS error over 300 m. Field tests show the same results and prove that RATC is a good choice for reducing image erros.

A similar approach was taken by Ahsan, Rafique and Abbas in [3], but a PID controller was used instead of a PD. The controller was created from a nonlinear model which have 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 shows that when using rudder as a control surface the aircraft has better response compared to aileron, with 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. THIS PARAGRAPH COULD BE BETTER

Mills, Ford and Mejias refers to a rudder-based course controller as a 'skid-to-turn' controller in [1], and in the paper it is used to control a UAV that uses a camera to survey a linear infrastructure. To control the aircraft based on images a variation of controller called Image-Based Visual Servoing (IBVS) is used. This method identifies the structure that is to be surveyed, and creates a model of it as a straight line. This line becomes the track the UAV is supposed to follow, and the UAV will seek to minimize the track error. If required by the steady-state errors the wings may not be levelled and the aircrafts heading may not follow the track, as long as the line is within the camera's field of view (FOV). A PID controller is used to control the heading using the rudder.

The controller was simulated compared to a controller that banks the UAV to turn, and the results matches the previous results. Even though the bank-to-turn controller reduces the track error faster than skid-to-turn, skid-to-turn causes much less error in the image plane. Even though the camera's FOV at all times covered the structure being surveyed, the roll the UAV makes back and forth changes so quick that the images retrieved might very well be to blurry to be usable. The controllers were also tested in wind with similar results. One thing worth noticing 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 much smaller than for the bank-to-turn controller.

2.1.2 Heading controller with constraints

Since banking the aircraft quickly causes the camera's FOV to be away from our point of interest, it is possible to put constraints on the banking angle to ensure the camera stays focused on what we want. In [4] this is done by putting constraints on the UAV roll and the above ground level (AGL) to track a roadway. The constraints are calculated from the camera's horizontal field of view, the assumed road width, and the expected turn angle of the road. In addition the AGL will influence the constraint for roll since these are highly connected.

The architecture of this system includes a 'Constraints Governor' that receives input from an image processing unit about the road it is following and telemetry input from the UAV about its position and heading. Based on these inputs the constraints ared calculated so that the road will stay within the camera's FOV. These constraints was forwarded to a previously made controller.

The system was simulated and only tested without wind. The simulated UAV successfully followed two 90° turns with only losing the road from the camera's FOV two times. This was because the system did not estimate the road path well enough, and the paper argues that by pointing the camera forward the estimation can be improved.

2.1.3 Summary of review

Turning using rudder is the most common way to reduce the roll during UAV operations, and it gives a significant reduction in the image error because of this. These papers also show that even though it does not perform as good as the traditional controllers using ailerons when it comes to position and path following, the rudder controllers perform better in the image plane despite the slower convergence rate.

One interesting point made in [2] is that the RATC controller will ease the flight plans for ground observing. When using AOTC controllers for ground observing extra measures often has 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 then 90° bend. When using the RATC controller developed, the flight path length and time was reduced by about 80%, and the energy spent flying the corner was estimated to give an 75% redutcion in energy spent. 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.

2.2 Path Planner

When an UAV is to operate autonomously there are other things than controlling the UAV states that needs to be taken into consideration. The path planner plays an important role in telling the controllers what the states should be. There exists many different path planners depending on the situation. A path planner may, for example, be used to aviod 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 follow a predetermined path in order to observe the ground with a camera.

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 principles can be used in three dimensions [7].

Dubin's path for UAVs can be used in several situations, and in [8] it is demonstrated 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 generater, which then transmits the path to the path-following strategy. This strategy 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 previously in this paper, an airplane may be turned without banking 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 compared with an rigorous optimization algorithm, in order to check the quasi-optimality of the Dubins's-based path. 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 camera in mind. A hyperspectral camera makes it possible to accurately detect types of material from the UAV, but is also sensitive to errors.

SPECIFY THAT IT IS A PUSHBROOM SENSOR

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 reflicted light from materials [10]. The amount of light that is reflected by a material at different wavelengths is decided 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 drone, 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 spectras of the different materials are combined additively. The combined spectra can be split into the different spectras that it is build up of by removing noise 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 done 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.

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. 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:

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

3.1 Wind

Wind will introduce what is called crab angle χ_c in the horizontal plane and the angle of attack α in the vertical plane. This will change the UAVs actual course χ and the air-mass-referenced flight-path angle γ_a to [5]:

$$\chi = \psi + \chi_c \tag{3.3a}$$

$$\gamma = \theta + \theta_a. \tag{3.3b}$$

The air-mass-referenced flight-path angle γ_a is defined as the vertical angle of the air-speed V_a relative to the north-east plane. These angles will only affect the navigational part, and not where the camera is pointed.

3.2 Camera Position

The point where the camera is pointing 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 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 is expressed as

$$\boldsymbol{c}_b^b = \begin{bmatrix} c_{x/b}^b \\ c_{y/b}^b \end{bmatrix} = \begin{bmatrix} z_n tan(\theta) \\ z_n tan(\phi) \end{bmatrix}. \tag{3.4}$$

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 $\boldsymbol{R}_{z,\psi}$:

$$\boldsymbol{c}_b^n = \begin{bmatrix} c_{x/b}^n \\ c_{y/b}^n \end{bmatrix} = \boldsymbol{R}_{z,\psi} \boldsymbol{c}_b^b, \tag{3.5}$$

where:

$$\boldsymbol{R}_{z,\psi} = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.6}$$

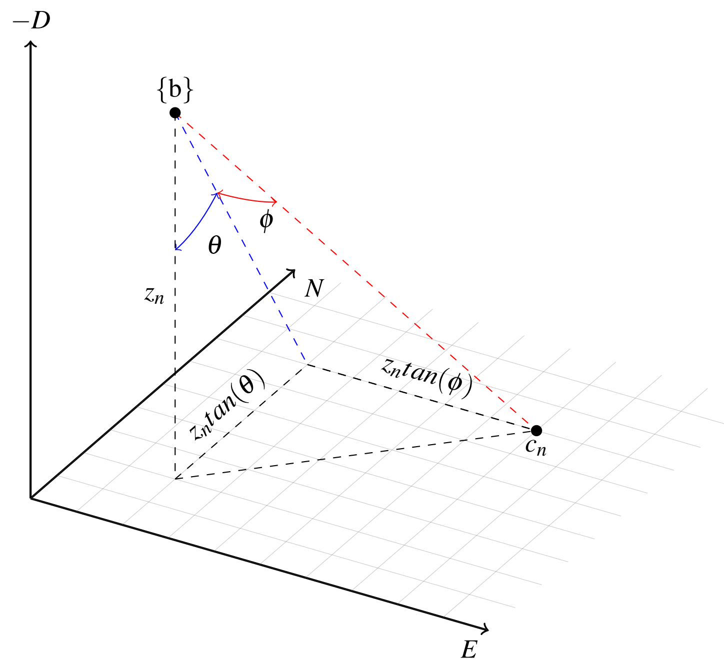

Figure 1: Illustration of how the angles influence the camera position.

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

$$\boldsymbol{c}^n = \begin{bmatrix} c_x^n \\ c_y^n \end{bmatrix} = \begin{bmatrix} x_n + c_{x/b}^n \\ y_n + c_{y/b}^n \end{bmatrix}. \tag{3.7}$$

3.3 Camera Angle of View

Since the camera isn't only focusing on one specific point, it can be useful describing the camera point of focus as two extremities instead of one center point. Equation (3.4) can easily be changed to do this. Assuming the camera has an angle of view σ, the equation now becomes:

$$\boldsymbol{e}_{1,b}^b = \begin{bmatrix} e_{x/b}^b \\ e_{y_1/b}^b \end{bmatrix} = \begin{bmatrix} z_n tan(\theta) \\ z_n tan(\phi+\sigma) \end{bmatrix}, \ \boldsymbol{e}_{2,b}^b = \begin{bmatrix} e_{x/b}^b \\ e_{y_2/b}^b \end{bmatrix} = \begin{bmatrix} z_n tan(\theta) \\ z_n tan(\phi-\sigma) \end{bmatrix}. \tag{3.8}$$

The steps for translating the points to the NED frame are the same as in (3.5) and (3.7):

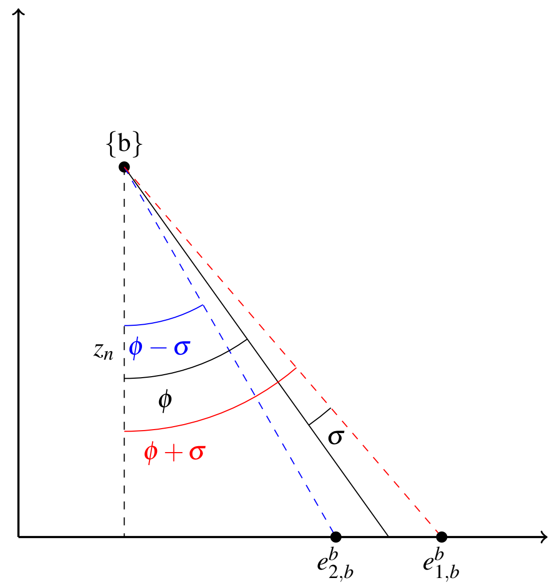

Figure 2: Illustration of how the field of view is calculated.

$$\boldsymbol{e}^n_b = \begin{bmatrix} e^n_{x/b} \\ e^n_{y/b} \end{bmatrix} = \boldsymbol{R}_{z,\psi}\boldsymbol{e}^b_b \tag{3.9}$$

$$\boldsymbol{e}^n = \begin{bmatrix} e^n_x \\ e^n_y \end{bmatrix} = \begin{bmatrix} x_n + e^n_{x/b} \\ y_n + e^n_{y/b} \end{bmatrix}. \tag{3.10}$$

4 Controller

4.1 Dynamics

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

To simplify the controller, it will be assumed that there is no wind and no sidslip 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. This will simplify the system since the roll angle ϕ and pithc angle θ both can be assumed to be small.

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

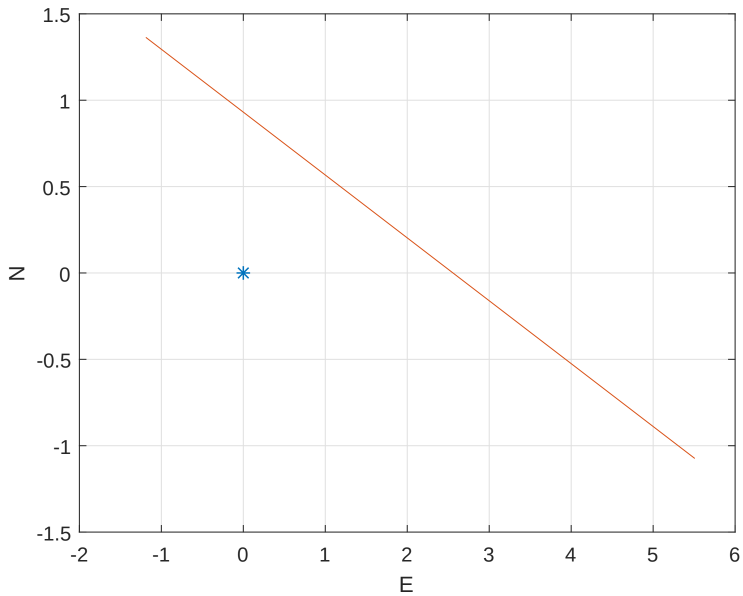

Figure 3: Graph showing the line the camera captures when the plane is positioned in the origin with an altitude of $10m$, and $\phi = -10$, $\theta = -5$ and $\psi = 20$. The field of view is $20°$.

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

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 expresed 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 (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 get

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

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 change 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 p will be zero, as will the aileron deflection δ_a. During normal operation it cannot be assumed that no sideslip 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 Dubin's 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 aircrafts autopilot will maintain a constant height. 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 heading angle. For the kinematic model of the aircraft in this paper, u_1 equals to V, θ equals to ψ.

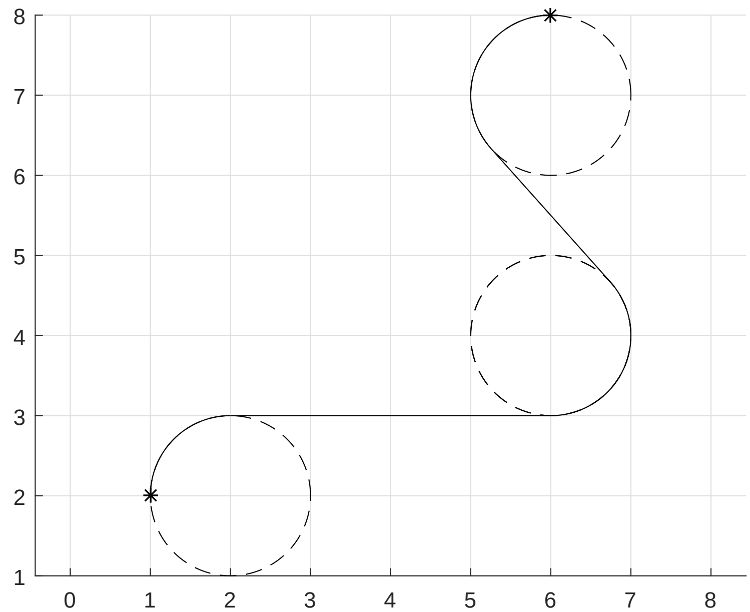

Figure 4: Illustration of a simple Dubins path.

When generating a Dubins path there are foure 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 circle.

In order to generate a Dubins path for this paper the algorithm proposed in Beard & McLain was used (algorithm 7, [5]). The algorithm takes the start and end postion, start and end heading 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 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. It returns information about whether or not the aircraft should follow a straight line or track a circle, and the information needed to do this. This information is given to two algorithms

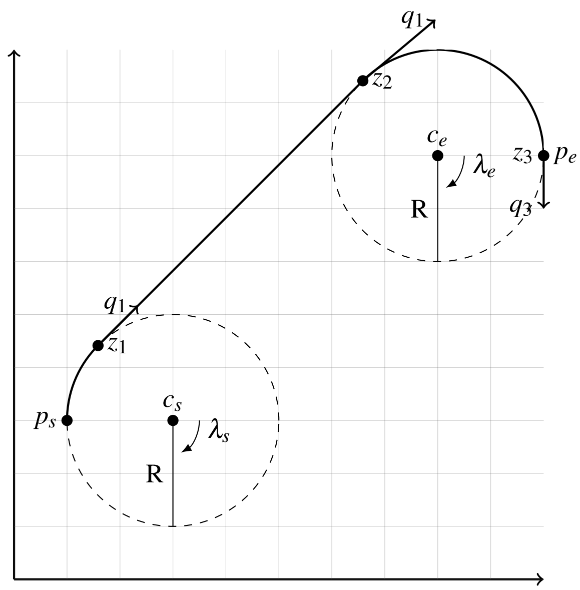

Figure 5: Illustration of the parameters returned by the Dubins algorithm.

that calculate the desired heading that can be fed to the autopilot. These algorithms will be described in chapter 8.1.

5.2 Altering the original path

The Dubins path described in the previous section will be used to generate a path based on the initial waypoints that the UAV is to observe. 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 camera is always pointing at the points of interest. 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 caused by the roll is needed. This corresponds to $c^b_{y/b}$ from equation (3.4):

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

$c^b_{y/b}$ 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 heading ψ, and the direction should

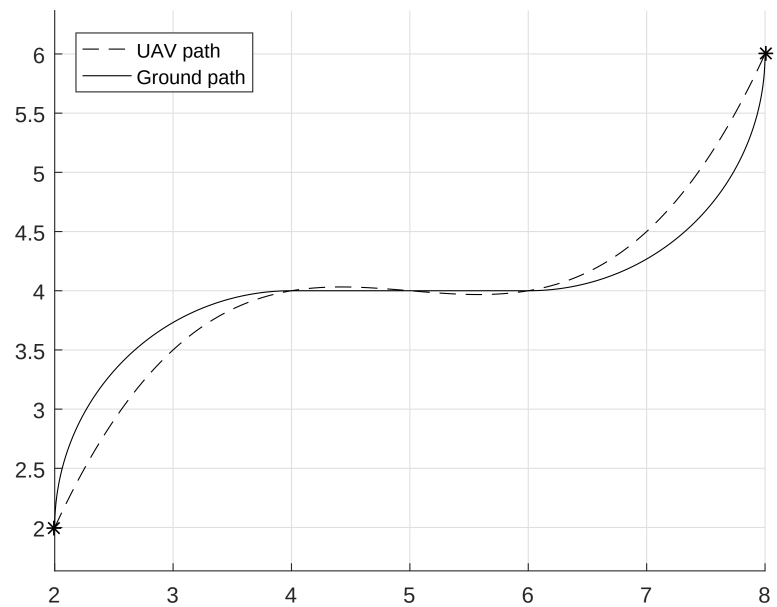

Figure 6: Illustration of the principle for altering the path.

be perpendicular to ψ as shown in figure 7. The coordinates for the new path $\boldsymbol{p}_d$ in the body frame $\{b\}$ then becomes

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

and in the NED frame $\{n\}$:

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

6 Simulation

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

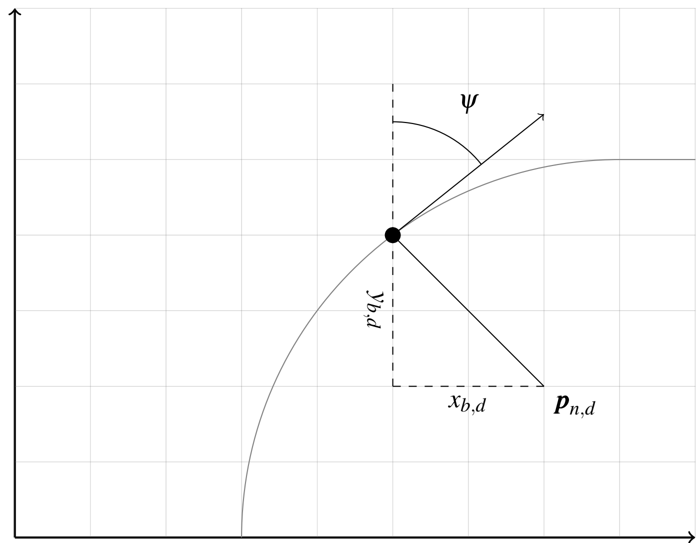

Figure 7: Illustration of the direction for altering the path.

6.1 Model

The model of the Aerosonde UAV is based on parameters and equations given by Beard & McLain [5]. (Referere til Gryte?)

The model is split into to 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 do 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 have also been developed by Gryte (Skal det refereres?), and it is also based on equations in Beard & McLain [5]. The autopilot have previously been used with a different UAV, and therefore needed to be tuned to work with the Aerosonde. The controller loops are defined by relative damping factor ζ and natural frequency ω, and by implementing the control loops outside the model the parameters can be found seperately. This makes a good starting point for further tuning.

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 and actuators.

7.1 Controller Implementation

The controller was implemented using a simple block diagram in Simulink, with desired course as input and rudder control as output. As a starting point for the controller tuning the control loop was simulated in an open loop.

7.2 Test Cases

The altitude 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], [15], [16]). However, altitudes as high as 1900 m has been used to observe agricultural crops [17]. In this paper simulations will be performed mostly at 100 m, with some simulations at higher altitudes for comparison. The FOV for the camera will be set to 19° (approximately the same as in [13]).

The controller has been tested in three different cases. The first case is a simple 45° turn in order to test the step response of the controller. The second case will follow a path in order to compare the controller with a ”regular” course controller. The third and last case is the same path as in the second case, but with wind.

7.3 Results Case 1

7.4 Results Case 2

7.5 Results Case 3

7.6 Results

8 Simulation of Path Planner

The same autopilot that was used for simulating the controller will be used when simulating the path planner. A path follower will be used to give course commands to the autopilot during simulation, while the rest of the states will be controlled only by the autopilot.

8.1 Path Follower

Two different path followers will be used in this simulation. The first path follower will be used to follow the Dubins path, while the second will be used to follow the continuous path that is generated as an improvement to the Dubins path.

The strategy used to follow Dubins path will be based on two algorithms presented in [5] by Beard & McLain. 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 heading of the aircraft, the previous waypoint and the direction from the previous to the next waypoint as input. The previous waypoint and 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 heading of the aircraft. The heading calculated here will also ensure that the aircrafts position converges to the cirular path.

The altered path is based on the route that is flown 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 [18].

Enclosure-based steering is LOS principle that considers a circle with radius R enclosing the vehicle, which represents the LOS distance. Assuming that the radius is suffieciently 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 [18]:

$$\chi_d(t) = \text{atan2}(y_{los} - y(t), x_{los} - x(t)) \tag{8.1}$$

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:

$$[x_{los} - x(t)]^2 + [y_{los} - y(t)]^2 = R^2 \tag{8.2}$$

where R is the chosen LOS distance.

8.2 Simulation Setup

The simulation was performed using the same Simulink and Matlab setup as previously, using the path followers to generate the desired course angle. The Dubin's path that will be used as a reference is shown in figure 8.

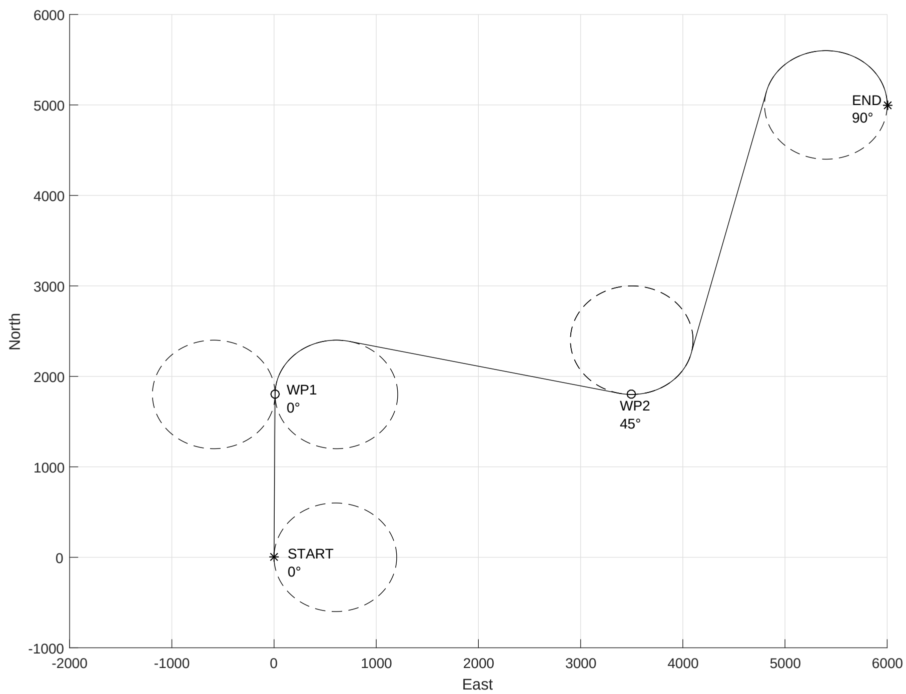

Figure 8: The path that will be simulated, with the direction associated with every waypoint.

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 wing and no sideslip, the equation for a coordinated turn becomes [5]:

$$R = \frac{V_g^2}{g\tan(\phi)}. \tag{8.3}$$

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 maneouvres. The LOS distance was set to 200 m by trial and failure.

8.3 Result: Path Following

Figure 9 and figure 10 shows the result of the simulation when the UAV follows the generated Dubins path that is to be observed. The aircraft follows the observation path closely, and in turns it drifts slightly off and takes the outer turn.

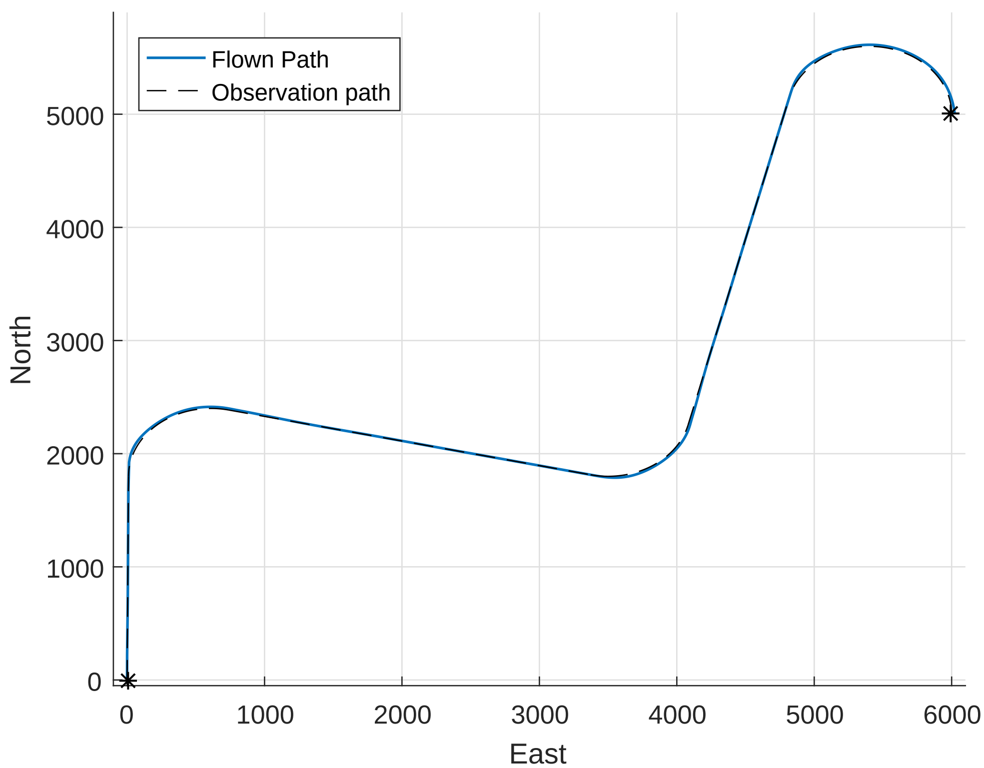

Figure 9: The path flown by the aircraft on the first run.

Figure 11 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 12. The figure shows some ”NUDGES” due to the uneven turn, mainly at the beginning and end of the turns.

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 than

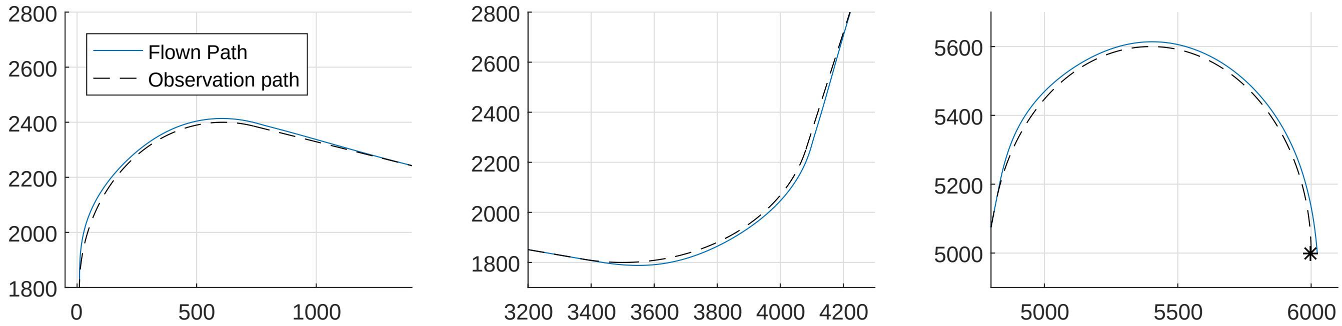

Figure 10: The path the aircraft took through the turns on the first run.

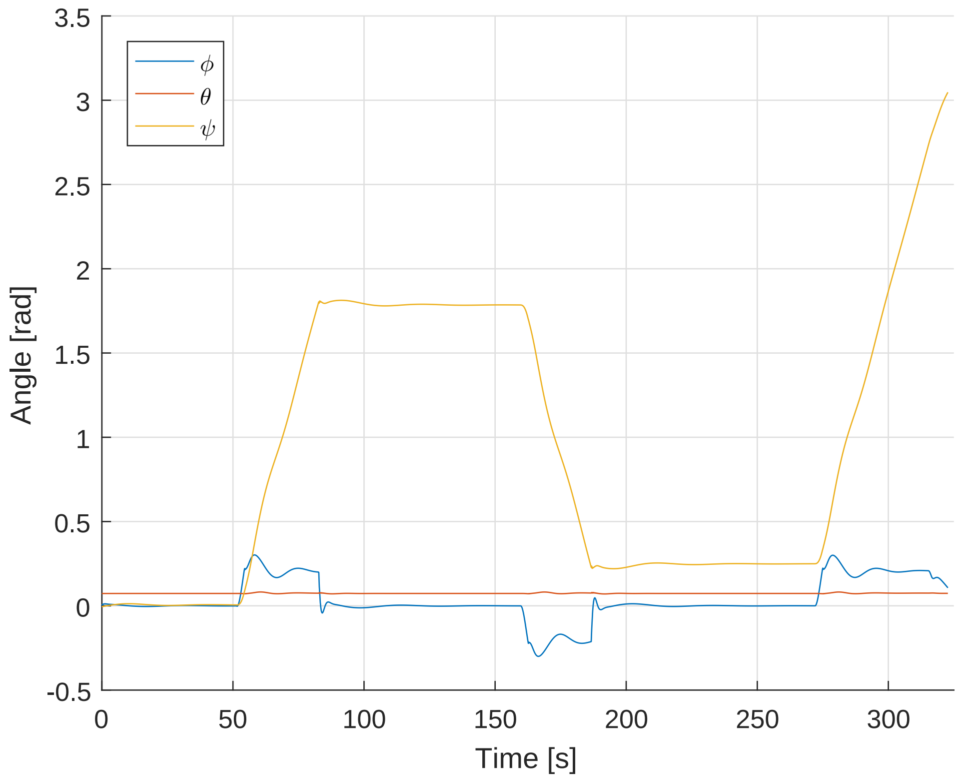

Figure 11: The attitude states of the aircraft during the first run.

the slow changes. This 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

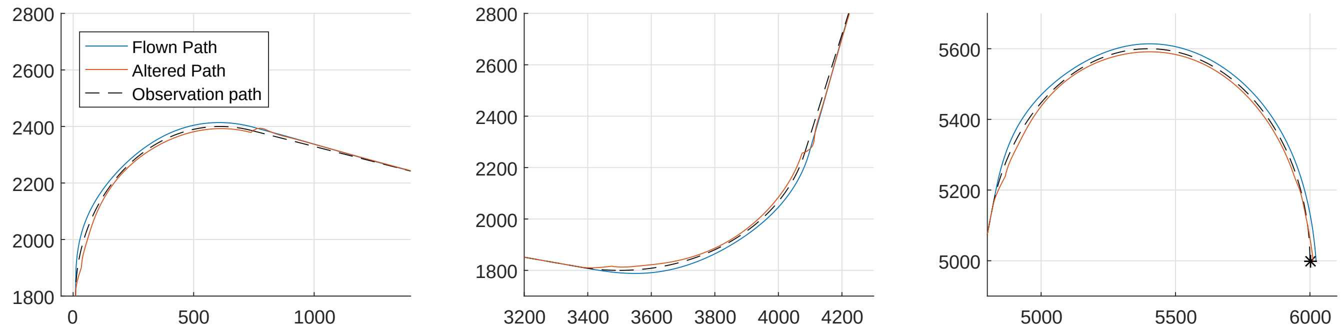

Figure 12: The figure shows how the altered path is compared to the flown path. Only the turns are shown as they are matching during the straight levelled sections.

low-pass filtered ϕ is shown in figure 13, and the result that the new path is smoother than the first one.

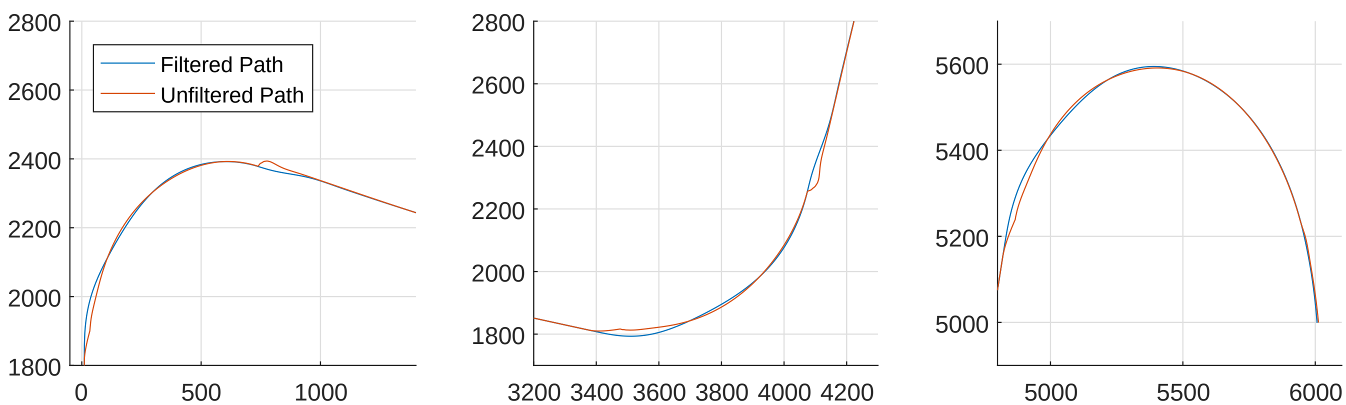

Figure 13: The path created by using the filtered signal for ϕ compared to using the unfiltered.

The result of the simulation when following the altered path is shown in figures (14), (15) and (16). 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. It can be seen that ϕ is still uneven during turns, and about the same value as during the first run.

8.4 Result: Camera Footprint

The camera footprint for the original path is shown in figures 17 and 18. While the camera footprint is positioned fairly straight above the observation path, the observa-

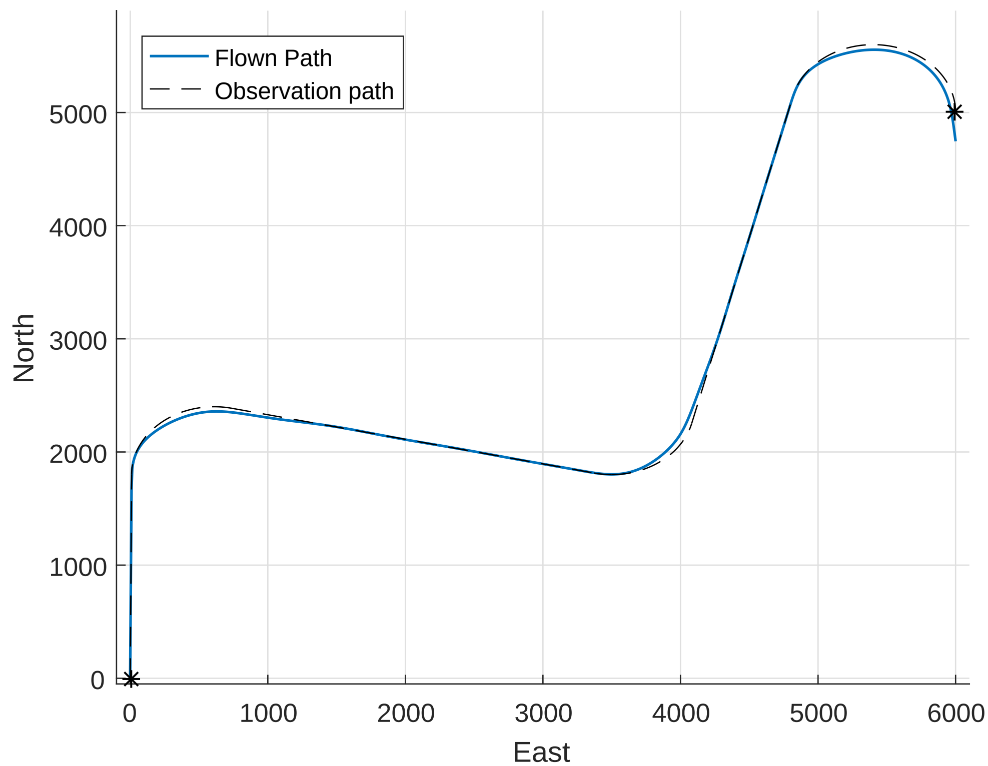

Figure 14: The path flown by the aircraft when following the altered path.

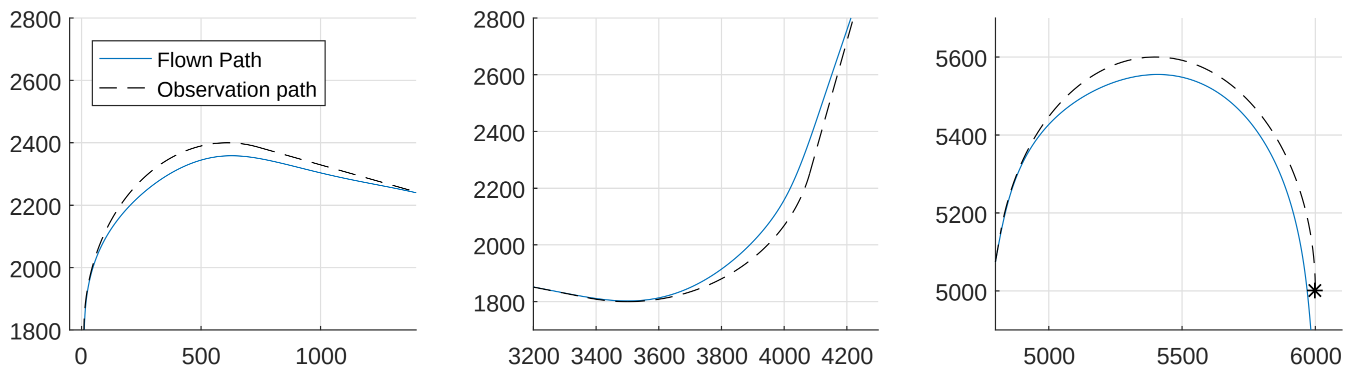

Figure 15: The path the aircraft took through the turns when following the altered path.

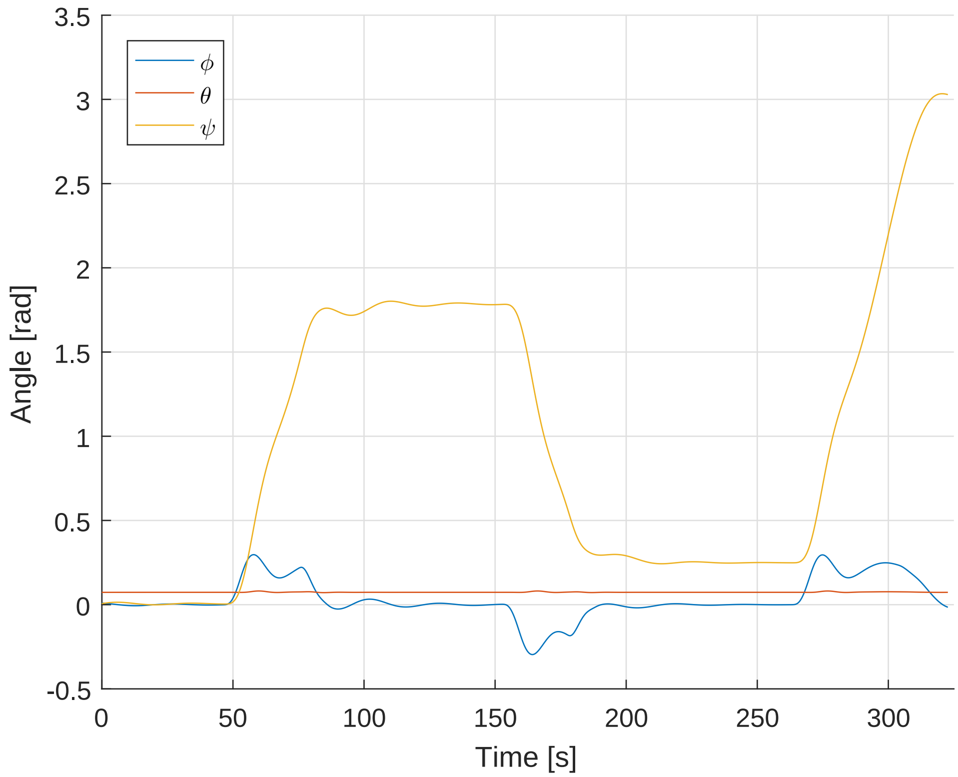

Figure 16: The attitude states of the aircraft when following the altered path.

tion 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.

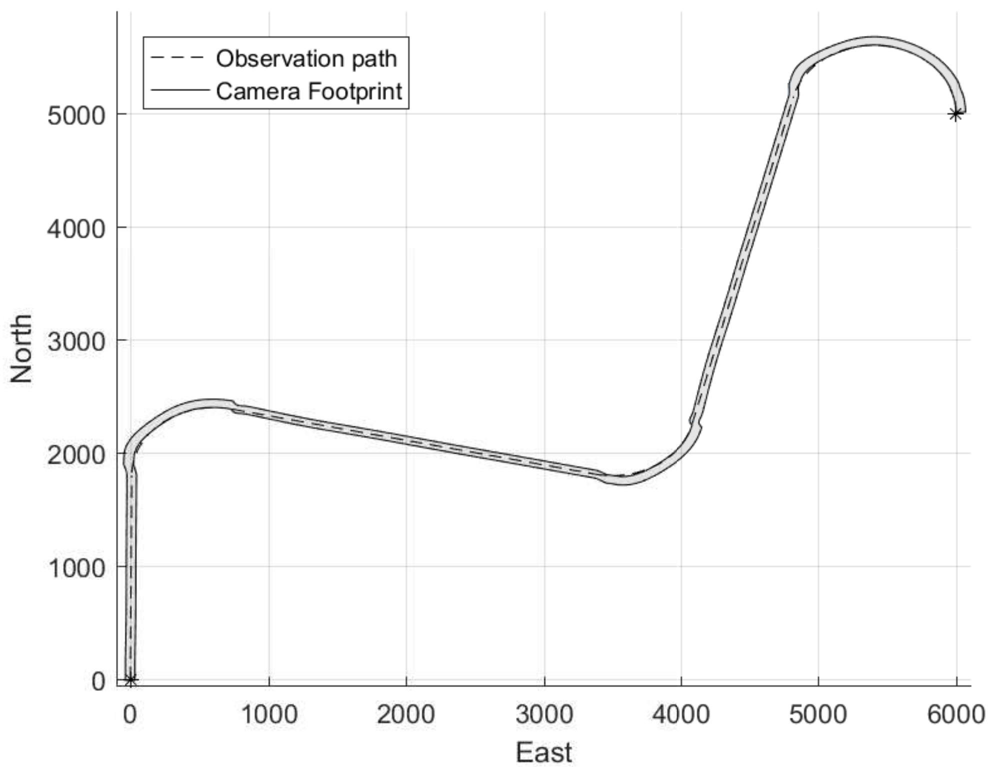

Figure 17: The camera footprint during simulation of the first path.

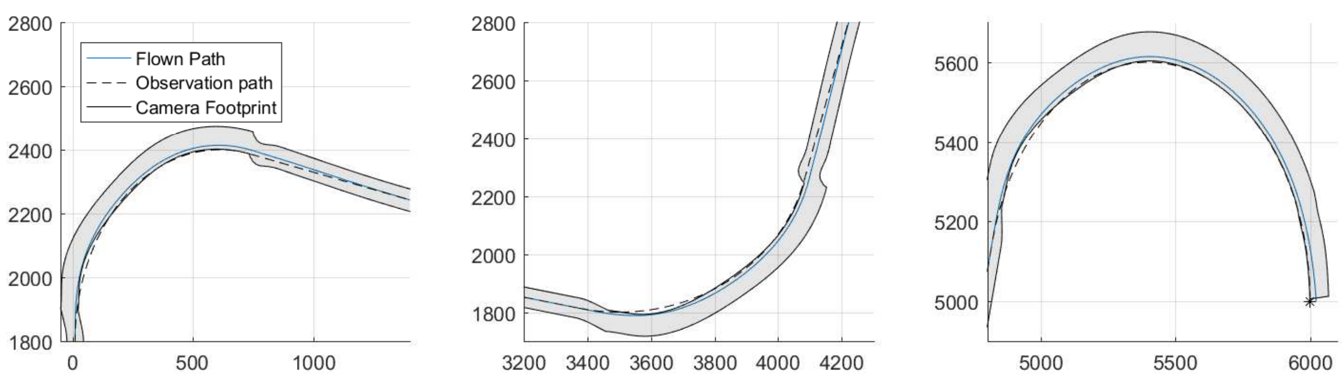

Figure 18: The camera footprint in turns during the simulation of the first path.

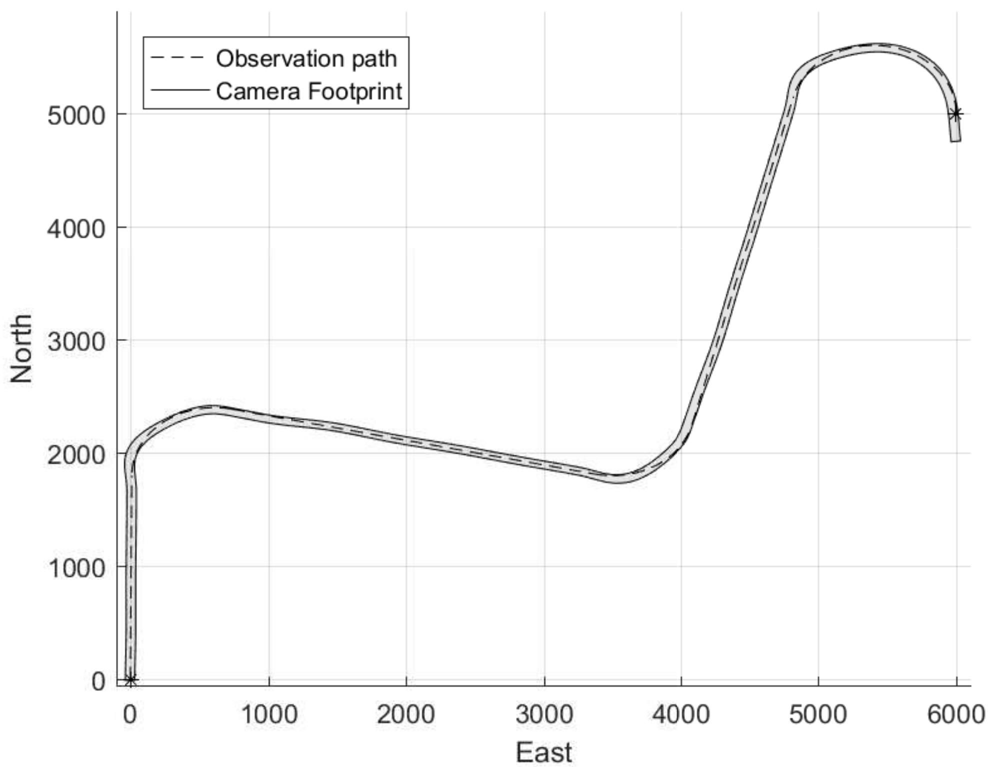

Figure 19: The camera footprint during simulation of the altered path.

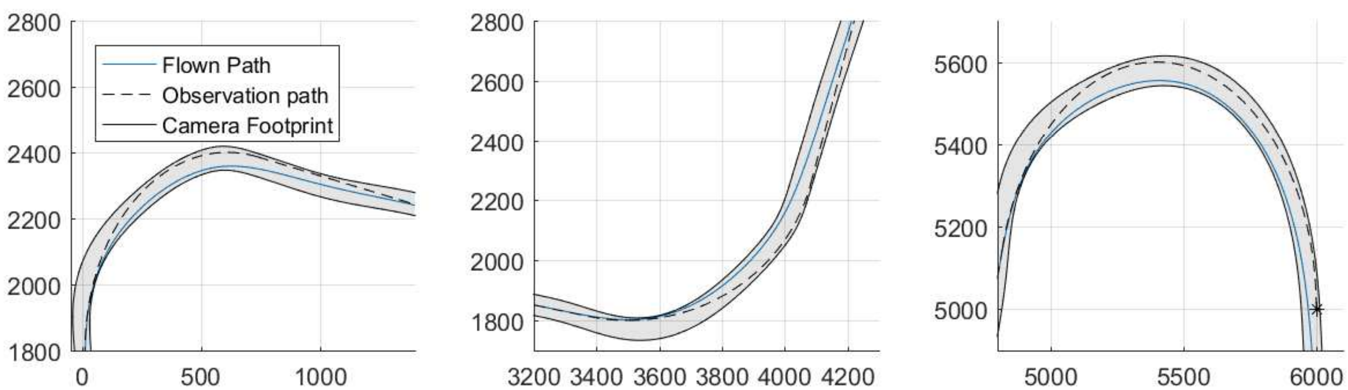

Figure 20: The camera footprint in turns during the simulation of the altered path.

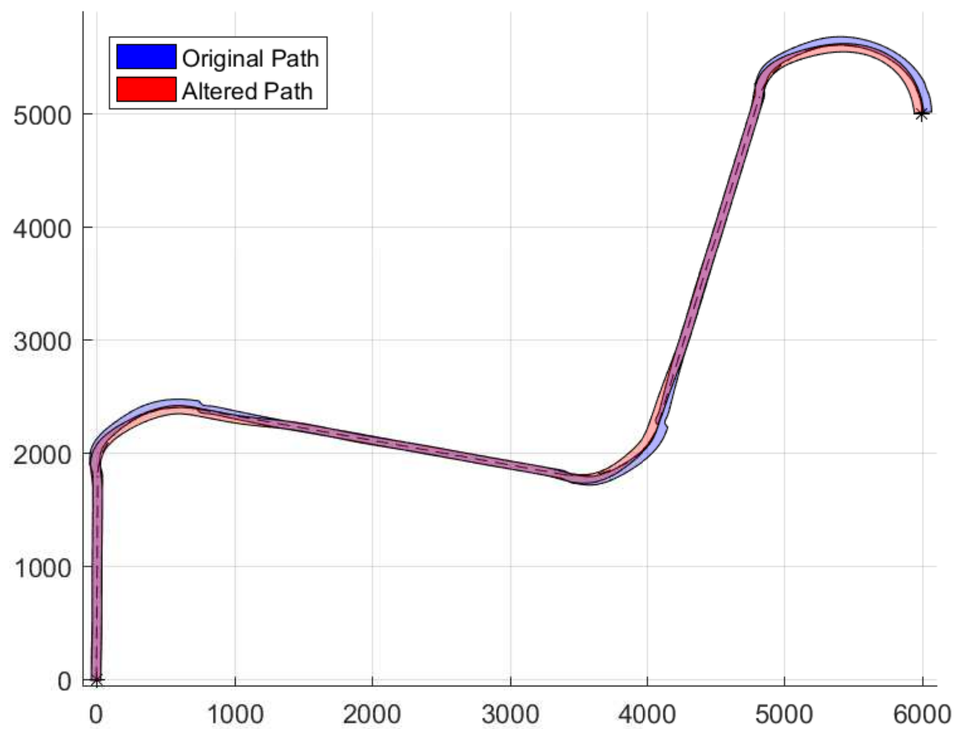

Figure 21: The camera footprint during simulation of the altered path.

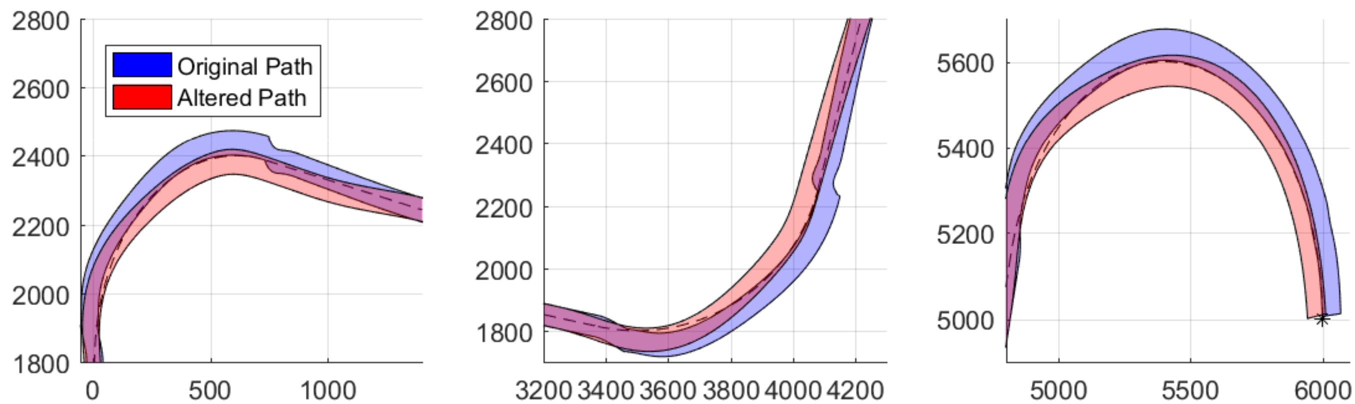

Figure 22: The camera footprint in turns during the simulation of the altered path.

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] 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

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

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

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