NTNU – Trondheim
Norwegian University of
Science and Technology

POSITION CONTROL FOR AUTOMATIC LANDING OF UAV IN A NET ON SHIP

Kjetil Hope Sørbø

Submission date: April 2016
Responsible professor: Thor Inge Fossen, Tor Arne Johansen

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Abstract

Automatic landing of a fixed wing Unmanned Aerial Vehicle (UAV) in a net on a ship require an accurate positioning system. There exist today high-end systems with such capability for special applications, e.g military systems and costly commercial systems, which restrict the availability of such systems. To increase the general availability these systems must consist of low-cost components. Here, an alternative is the use of low-cost Global Navigation Satellite System (GNSS) receivers and apply Real Time Kinematic GPS (RTK-GPS), which can provide centimeter level position accuracy. However the processing time for the RTK-GPS system results in degraded accuracy when exposed to highly dynamical behaviour.

This work present two alternative software and hardware position systems suitable for use in navigation system which apply RTK-GPS, namely Real-Time Kinematic Library (RTKLIB) with a Ublox Lea M8T receiver and a Piksi system. Both the Piksi and the Ublox receiver are single-frequency GNSS receivers. These systems will in this work be compared and their individual capability to provide accurate position estimate will be evaluated.

The RTK-GPS system is implemented in DUNE (DUNE:Unified Navigation Environment) framework running on an embedded payload computer on-board an Unmanned Aerial Vehicle (UAV).

The performance of these position systems are in this work investigated by experimental testing. The testing showed that the RTKLIB performed better than the Piski alternative, and further showed the tested navigation system provide sufficient quality for integration into a control and guidance system, allowing for automatic landing of an UAV in a net.

Contents

Chapter 1

Introduction

1.1 Background and motivation

Recent development of flying UAVs has been recognized to provide an attractive alternative to work previously performed by manned operations. Typical work which has attracted attention includes inspection, aerial photography, environmental surveillance and search and rescue. Today UAVs are mostly operated over land, however in the future this will include over sea as well. This will give some challenges which must be overcome. One of these challenges is that the UAV need to be able to perform a autonomous landing.

An UAV can provide an attractive alternative for many maritime operation where today manned aircraft or satellites is the only solution. In the maritime sector UAV can be used in iceberg management, monitoring of oil spills, search and rescue and maritime traffic monitoring.

An important premise for successful and safe UAV operation, in particular at sea, is the provision of a robust system for safe landing of the UAV on a vessel following completed operations. A autonomous landing system require a robust guidance and navigation system, as well as the ability to generate a flyable landing path during flight operation that is within the operation criteria set by the operator. A requirement for the system is that the operator is able to monitor the state of the uav, including the state of the navigation system e.g gps system. If the gps system looses its fixed solution during a critical phase a abort with an evasive manoeuvre might be required. The decision of whether the uav should perform a evasive manoeuvre will be further explored in section (REF:EVASIVE).

Due to regulatory mandate there are restriction on the size of operational area for a uav. Different types of operation is LOS,EVLOS,BVLOS. During LOS the uav must be in line of sight of the operator, which restrict the area were a autonomous landing can take place. In addition there is the risk of loosing satellites during high dynamic behaviour which limits the type of landing path that are available. Therefore a flyable path must be generated for a arbitrary pose, which will provide a gentle landing for the uav.

The scope of this thesis is the design, implementation and testing of an autonomous landing system for a uav. The main focus in this thesis will be on the navigation and path planning of the landing system. A fellow master student had the main focus of developing the control and guidance system, which will be explain in section ref.

This thesis contain a concept of a robust navigation system for autonomous landing of a UAV, and the implementation of the autonomous landing system. The landing system has been implemented together with a other student, and is a continuation of the master thesis from [Frølich, 2015] and [Skulstad and Syversen, 2014]. The navigation system that has been implemented apply rtk-gps for position estimation.

1.2 Related work

A disadvantage with a net recovery system that is stationary on the deck of a ship is the space requirement for the net, including the safty zone for the personnel required for the uav operation. The paper (Ref multicopter paper when published) addresses this problem by moving the net away from the ship by the means of multirotor uavs. The proposed net recovery system has the advantage that motion induced by the sea is removed, however there is the risk of losing the uav when colliding with the net. A solution that is currently explored is the use of hooks on the uav, which will allow it to grip the net.

A low-cost net recovery system for UAV with single-frequency RTK-GPS was described in the paper [Skulstad et al., 2015], which was a result of the work done in the master thesis [Skulstad and Syversen, 2014]. The system presented applied RTKLIB together with low-cost single frequency Global Positioning System (GPS) receivers as navigation system with a customized Ardupilot software. The complete system was able to perform a net landing, however the result showed that further work would require better controllers, and a more robust navigation system.

Chapter 2

Path planning theory

An autonomous system must be able to create a plan on how the system should move around in the surrounding environment in a feasible way. A minimum requirement for a path is that it is connected. The connection level can be described by the paths smoothness. Parametric continuity is denoted C^n were n is the degree of smoothness. The order of n implies that the n first parametric derivatives match at a common point for two subsequent paths [Barsky and DeRose, 1989]. Geometric continuity is a relaxed from of parametric continuity in witch discontinuousness in speed is allowed. A table of geometric and parametric continuity lists the requirement for each smoothness level 2.1, which is based definitions presented in [Barsky and DeRose, 1989]. Geometric continuity is sufficient for a path following system, witch is the main focus of this thesis. Geometric continuity is denoted as G^n were n is the order of continuity.

Geometrical smoothness level	**Description**
G^0	All subpaths are connected
G^1	The path-tangential angle is continuous
G^2	The center of curvature is continuous
Parametric smoothness level	**Description**
C^0	All subpaths are connected
C^1	The velocity is continuous
C^2	The acceleration is continuous

Table 2.1: Smoothness definitions

The definition used for path in this thesis is equation 1.2 in [Tsourdos et al., 2010] which state:

$$P_s(x_s, y_s, z_s, \theta_s, \psi_s) \xrightarrow{r(q)} P_f(x_f, y_f, z_f, \theta_f, \psi_f) \tag{2.1}$$

where the subscripts s and f denotes the start pose and finish pose respectfully with $r(q)$ as the path.

2.1 Straight lines

The simplest form on path is a straight line between P_s and p_f. The straight line is given as

$$x(s) = a_x s + b_x \tag{2.2a}$$
$$y(s) = a_y s + b_y \tag{2.2b}$$

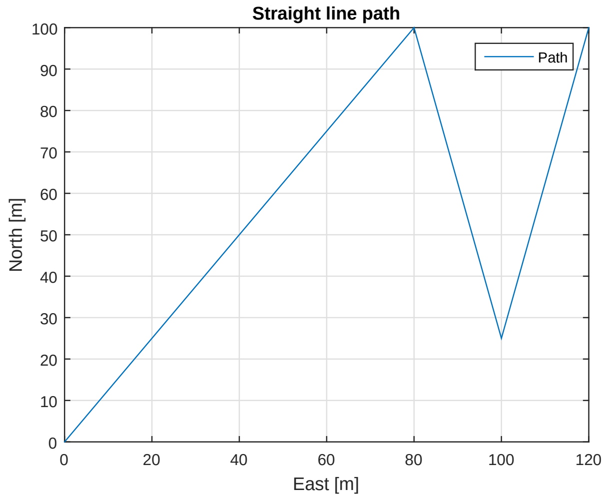
with $s \in [0, 1]$, where s has not necessary a physical meaning. Then the parametrisation of the straight line is:

$$x(0) = b_x \rightarrow b_x = x_s \tag{2.3a}$$
$$x(1) = x_f = a_x + b_x \rightarrow a_x = x_f - b_x \tag{2.3b}$$
$$y(0) = b_y \rightarrow b_y = y_s \tag{2.3c}$$
$$y(1) = y_f = a_y + b_y \rightarrow a_y = y_f - b_y \tag{2.3d}$$
$$\tag{2.3e}$$

The tangential vector for a straight line is given as:

$$\psi(s) = \operatorname{atan2}(a_x, a_y) \tag{2.4}$$

A path constructed by straight lines is G^0, however since the tangential vector is discontinuous between two line segments with different heading it's not G^1. The disadvantage with a path which is G^0 is that large discontinuity between two tangential vectors will cause problem for a control system.



The simplest for of creating a path is a straight line between two way-points. The advantage with a straight line is that it's easy for a guidance system to follow the line, however it will experience a jump in reference when transitioning to another straight line due to discontinuous tangential vector.

2.2 Dubins path

An alternative to a straight line path is a path constructed by straight lines and circle. Rudolf Dubin showed [Dubins, 1957] that the shortest possible path for a particle that moved with unit speed with maximum curvature would consist of three pieces. The path is considered as the shortest path from P_s to P_f, however the curvature is discontinues, which gives a smoothness level of G^1 and C^0. Dubins path is the shortest path from from one way-point to the other which is continues. Dubins path can be created for a three dimensional case, however a simplification is made in which only a planar version of the Dubins path is examined. A Dubin path with fixed end orientation can be constructed in four different way.

Right to Right
Right to left
Left to Right
Left to left

Table 2.2: Turning direction for Dubins path with fixed final orientation

Allowing the finish orientation to be free will add four more variants of the Dubins path. The path consist of two arcs and a straight line. The straight line is tangential to both arcs. The start and end point of the straight line can be found with

$$\alpha = \arcsin\left(\frac{\rho_f - \rho_s}{|c|}\right) \tag{2.5a}$$

$$\beta = \arctan\left(\frac{y_{cf} - y_{cs}}{x_{cf} - x_{cs}}\right) \tag{2.5b}$$

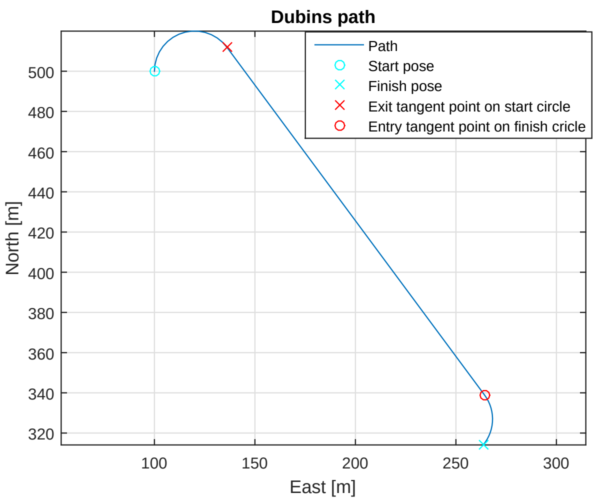
	Turn angle
ϕ_{right}	$\alpha + \beta + \frac{\pi}{2}$
ϕ_{left}	$\beta - \alpha + \frac{3\pi}{2}$

$$x_{P_\chi} = x_{cs} + R_s \cos(\phi) \tag{2.6a}$$

$$y_{P_\chi} = x_{cs} + R_s \sin(\phi) \tag{2.6b}$$

$$x_{P_N} = x_{cf} + R_f \cos(\phi) \tag{2.6c}$$

$$y_{P_N} = x_{cf} + R_f \sin(\phi) \tag{2.6d}$$



Chapter 3

RTKGPS

This chapter present some of the basic theory behind rtkgps.

When phase measurement is applied an important part. Integer ambiguity is the uncertainty of the number of whole cycles between the receiver and a satellite.

3.1 Real time kinematic GPS

In [Misra and Enge, 2011] section 7.2.2 Real Time Kinematic GPS (RTK-GPS) is defined as a rover that receive raw measurements from a reference receiver which is transmitted over a radio link, with a key feature that the rover is able to estimate the integer ambiguities while moving. The reference receiver is usually defined as a base station, and the integer ambiguity is the uncertainty of the number of whole phase cycles between the receiver and a satellite. With the measurements from the base station the rover is able to calculated the distance between itself and the base station, where the distance is referred to as a baseline. The length of the baseline affect the accuracy of the RTK-GPS solution, due to increased effect of atmospheric disturbance, which is further explain in 3.2.2. However with a short baseline, e.g. $1 - 2km$, the atmospheric condition can be considered equal for the base station and the rover, which keeps the solution at centimetre level accuracy.

The ability for the rover to resolve the integer ambiguity is a key feature in RTK-GPS. A well used method was purposed in the article [Teunissen, 1994] which decorrelate the integer ambiguities such that a efficient computation of the least square estimate can be performed. The search method is further explained in [Teunissen, 1995]. A estimate of the integer ambiguity with sufficient high degree of certainty is referred to as a FIX solution, otherwise the solution is degraded to FLOAT where the integer ambiguity is allowed to be a decimal or a floating point number. When the solution is categorised as FIX the accuracy of the solution is considered on centimetre level, while with a FLOAT solution the accuracy is at a decimetre level.

RTK-GPS can either provide a kinematic setting or a moving baseline setting. The difference between the two is that in kinematic the base station has a known stationary position, while in moving baseline the base station position is unknown and allowed to move. The unknown base station position is calculated with a single receiver, with the accuracy that entails. Therefore the RTK-GPS system with a moving baseline configuration can never have better global accuracy then what it will get with a single receiver. The advantage with the moving baseline configuration is that RTK-GPS can be used to find the relative position between two dynamical system using GNSS in real time. This will be the case in automatic ship landing system, where the base station is on a ship, thus must be allowed to move. The advantage with kinematic mode is that it can give a more accurate position estimate, where the relative position of the rover can be given in either the North East Down (NED) or East North Up (ENU) frame.

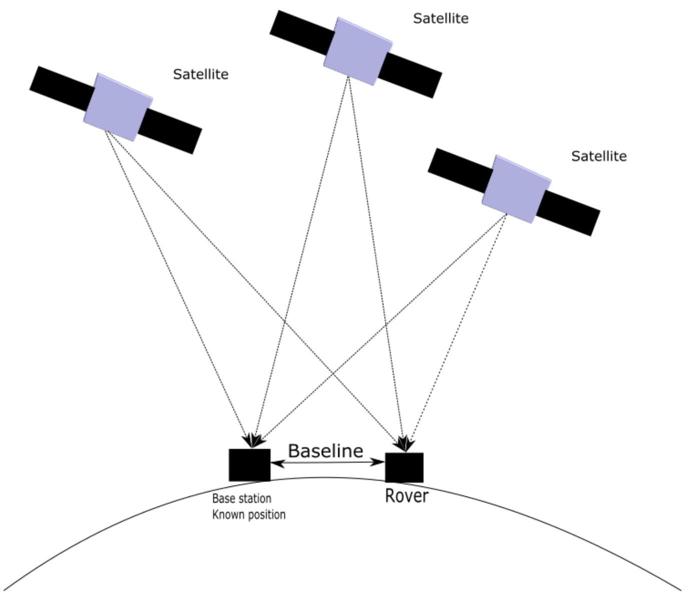


Figure 3.1: Concept figure of Differential GPS (DGPS)

3.2 Error sources

In order to get high accuracy in the position estimation the different error sources must be identified and removed if possible. This section will identify some of the most significant error sources that can affect the GNSS signal, and how to remove or mitigate them in the estimation.

3.2.1 Clock error

There is drift in both the satellite clock and the receiver clock. The atomic clock in the satellites makes the clock drift negligible from the user perspective. The receiver clock tend to drift, and if not taken into account will cause large deviations in the position estimate from the true position. This error is remove by including a fourth satellite in the position computation. The satellite clock error is given in the satellite message.

3.2.2 Ionospheric and tropospheric delays

When the GPS signals travel though the atmosphere there will be a delay caused by the different atmospheric layers. The atmosphere change the velocity of wave propagation for the radio signal, which results in altered transit time of the signal.

Ionospheric delay

Gas molecules in the ionosphere becomes ionized by the ultraviolet rays that is emitted by the sun, which release free electrons. These electron can influence electromagnetic wave propagation, such as GNSS signals. In [Vik, 2014] section 3.5.1 it's stated that the delay caused by the ionosphere usually is in the order of $1 - 10$meters. The error can be mitigated by using a double frequency receiver, or by applying a mathematical model to estimate the delay. Both those methods are with a single receiver, however by including a second receiver in a network, e.g. RTK-GPS, the GNSS solution system can assume that both receiver receive signal in the same epoch, which means that the signals have experienced the same delay. The rover is then able to remove the error induced from ionospheric disturbance.

Tropospheric delay

The tropospheric delay is a function of the local temperature, pressure and relative humidity. The effect of tropospheric delay can vary from 2.4 meters to 25 meters depending on the elevation angle of the satellites,[Vik, 2014] section 3.5.1. The error can be mitigated by applying a mathematical model to estimate the tropospheric delay, or by using a elevation mask can remove all satellites with a elevation angle bellow a certain threshold. Similar to ionospheric delay, tropospheric delay can be removed when using two receivers in a network by assuming that the single received by both receivers has experienced the same delay.

3.2.3 Multipath

One of the primary source of error in in a GNSS receiver is multipath. Multipath happens when the satellite signal is reflected by a nearby surface before if reach the GNSS antenna. The delay introduced in the signal can make the receiver believe that its position is several meters away form its true position. The easiest way to mitigated multipath is to place the antenna at a location with open skies, with no tall structures nearby. The effect can also be mitigated by choosing a antenna with good multipath rejection capability.

Multipath error uncorrelated between receivers, thus the local receiver must be able to correct for multipath error locally.

Chapter 4

System

The software that is used in the autonomous landing system is based on the LSTS toolchain [Pinto et al., 2013], with RTK-GPS solution calculated in RTKLib. The toolchain was developed for support of networked heterogeneous air and ocean vehicle systems. The toolchain supports interactions over wireless network, and is supports interaction with different system responseable for the low end control. The toolchain contain four different modules, namely Inter-Module Communication (IMC), DUNE, NEPTUS and Glued.

DUNE (DUNE Uniform Navigation Environment) is a runtime environment for unmanned systems on-board software written in C++. DUNE is capable to interact with sensors, payload and actuators, in addition to communication, navigation, control, manoeuvring, plan execution and vehicle supervision. The software separate operations into different task that each has there own thread of execution. DUNE apply a message bus that is responsible for forwarding IMC message from the producer to all registered receivers.

Neptus is a Command and Control software which is used to command and monitor unmanned systems that is written in Java. Neptus is able to provide coherent visual interface to command despite the heterogeneity in the controlled system that it is interacting with. This allow the operator to command and control unmanned system without the need to dwell into specific command and control software in the unmanned system.

4.1 Navigation system

The navigation system receive state information mainly from the pixhawk, however the the system is able to receive position and velocity from a RTK-GPS subsystem were the solution is calculated in RTKLib. The system decide if it should use RTK-

GPS through a DUNE task which manage the the source of the navigation data, which will be referred to as the navigation task.

The operator can monitor which source is used in the navigation task through a interface that indicate which source is available.

4.1.1 RTK-GPS system

The navigation system receive its RTK-GPS solution from a DUNE task, which is connected to the open-source program RTKLib [Takasu and Yasuda, 2009]. RTKLib contains several separated program that can be used in the field of RTK-GPS, where the program rtkrcv is used in the UAV to calculate the RTK-GPS solution. The program that is used in the base station is str2str, which retrieve the raw data from a GNSS receiver and transmits the data over tcp to the UAV. The GPS receivers that is used in both the UAV and base station is a Ublox Lea M8T. The structure of the RTKLib software configured with a rover and base station is shown in figure 4.1.

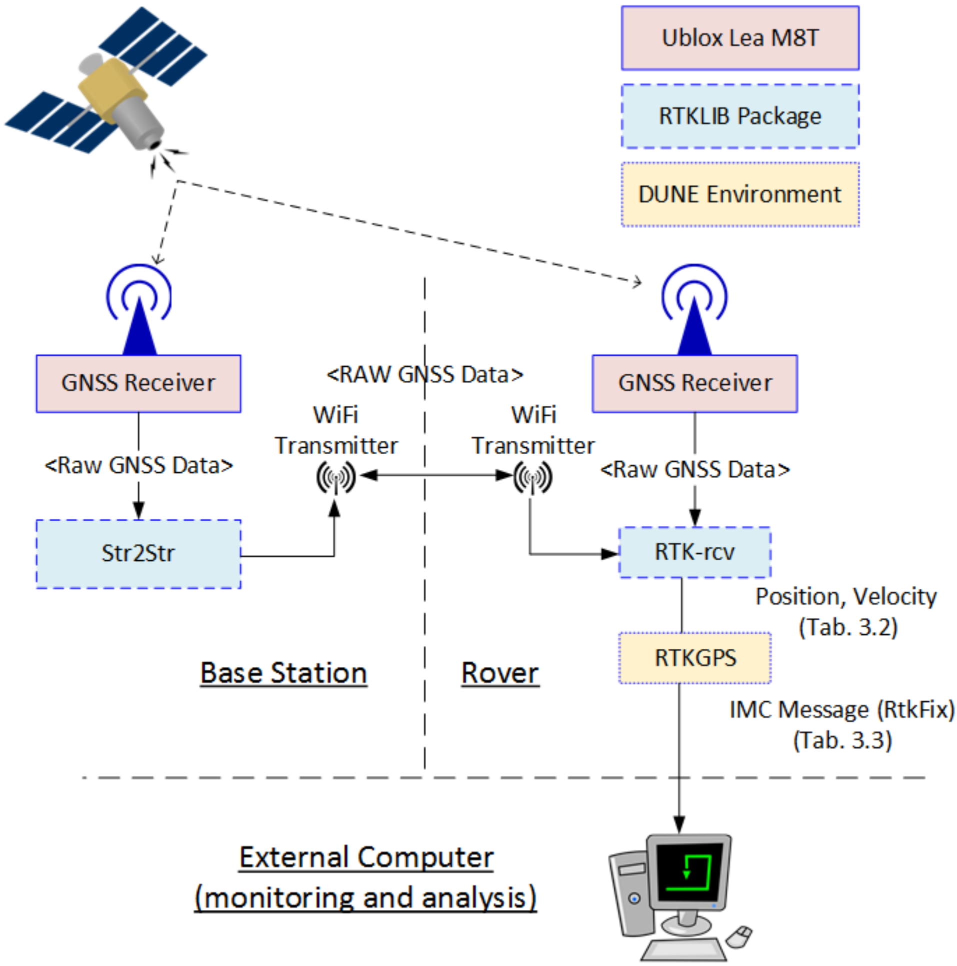


Figure 4.1: The communication structure of RTKLIB

As part of the RTK-GPS system the base station has it's own DUNE task running that calculate the position of the base station. The operator has to decide when the the GPS position of the base station should be considered as fixed, which will result in that the RTK-GPS DUNE task can include the base station in it's IMC message. This is require since RTKLib does not include the base position in it's output message.

4.1.2 Navigation source

The source of the position and velocity solution which is the output from the navigation system is decided in a DUNE task that consume both sensor data from the RTK-GPS system and the pixhawk. A requirement for use of RTK-GPS is that the position of the base station is fixed and included in the IMC message from the RTK-GPS task, in addition that the operator has decided that RTK-GPS should be used if available and the RTK-GPS solution is fixed.

The navigation allows for use of float solution of RTK-GPS given that it has previous been fixed, for a limited time.

4.1.3 Operator interface

A interface for navigation source monitoring has been created in Neptus in order to ease monitoring of which source the UAV is using as position and velocity source. The interfaced apply a color code to indicate which source is currently in use in addition to all sensor system that are available, as seen in table 4.1.

Color	**Description**
White	Not available
Yellow	Available, but not in use
Green	Available, and in use

Table 4.1: Net approach parameters

4.2 Path generation

The landing plan is design with the assumption that the aerial space in which the uav operates is limited. The limitation can be the range of the uav, regulatory limits or weather. In addition the autonomous landing system must be able to perform a safe landing from any initial position. Furthermore the size of the virtual run way should be constructed by the operator. The type of uav operation dictates the maximum size of the landing path. Different types of uav operation is LOS, ELOS, BLOS and BRLOS. Only the first is considered in this thesis, which means that the pilot must have the uav in view during the entire flight. The operator must also be able to control the angle which the uav is descending, which means that the height of the landing path is fixed. Therefore the uav must be at the correct altitude before it can start descending toward the landing net.

A landing plan that is proposed consist of Dubins path 2.2 in the lateral plane, and straight lines in the longitudinal plane. The path is generated from an arbitrary start positing, and will create a continuous path toward the landing approach. The design is inspired by the work done in [Skulstad and Syversen, 2014] were way-points was used to guide the uav toward the landing approach.

The plan is generated in a Dune task which receive a plan generation request from Neptus. Then a plan is created, which is sent to the Dune task Plan manager, which in turn sent desired state to the guidance system. The plan is decoupled from the guidance system, which allow for use of different control design when executing the landing plan.

The Dubin path is constructed with a followPath manoeuvre.

The landing plan generated by the Dune task can be requested for review by the operator in Neptus. The plan will not start before the operator has reviewed the plan, and approved by uploading it to the uav. In the uploaded version a specification list on which controller that should be used is included.

4.2.1 The net approach

The net approach path consist a straight line in the lateral plain, and straight lines in the longitudinal plane. The net approach way-points are defined relative to the position of the net, which has been defined as origo. The four way-points in the net approach is defined as follows:

$$\mathbf{WP1} = \begin{bmatrix} -a0 \\ 0 \\ h_{nc} + a1\tan(\gamma_a) \end{bmatrix} \tag{4.1a}$$

$$\mathbf{WP2} = \begin{bmatrix} a1 \\ 0 \\ h_{nc} - a1\tan(\gamma_a) \end{bmatrix} \tag{4.1b}$$

$$\mathbf{WP3} = \mathbf{WP2} + \begin{bmatrix} a2 \\ 0 \\ a2\tan(\gamma_d) \end{bmatrix} \tag{4.1c}$$

$$\mathbf{WP4} = \mathbf{WP3} + \begin{bmatrix} a3 \\ 0 \\ 0 \end{bmatrix} \tag{4.1d}$$

were the description of the parameters used is given in table 4.2. The net is placed between the first and second way points such that transitional behaviour do not occur during the finale stage of the net landing. In addition the path has been made with the assumption that the γ_a and γ_d is considered small. This assumption is made to easy the demand of the controllers used in the landing system.

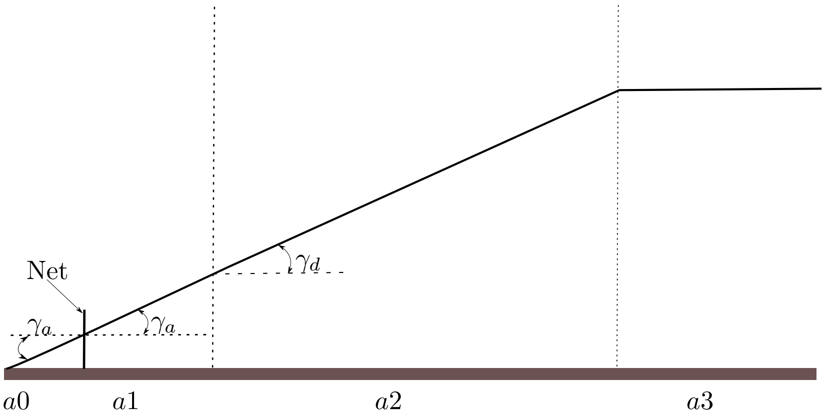
Parameter	**Description**
$a0$	The distance behind the net
$a1$	The distance in front of the net
$a2$	The length of the glide slope
$a3$	The length of the approach towards the glide slope
γ_a	The net attack angle
γ_d	The glide slope angle

Table 4.2: Net approach parameters

The way point vectors are rotated into the NED frame by a rotation around the z-axes.

$$WP^n = R(\psi_{net})WP^b \tag{4.2}$$

were ψ_{net} is the heading of the net.



4.2.2 The landing path approach

In order for the UAV to be able to start the landing approach the UAV must be at the correct altitude with the correct orientation from any initial position. That require a path which the UAV is able to follow. The problem can be separated into two parts. The first is the longitudinal parts, while the other is the lateral. The longitudinal path will be more exposed to large changes in heading, which puts demands of the type of path that can be chosen. However since a time demand is not required the problem becomes a path following problem. The problem has been solved by creating a Dubins path in the longitudinal plane.

A desired behaviour in the lateral plane is that the UAV should have a controlled decent. Given this desired behaviour it was decided that the lateral plane should be a glide slope, which means that the decent angle can be considered small. The strategy chosen to solve this problem is the use of straight lines in the lateral plane. The straight lines are is included with Dubins path such that the turn in Dubins path becomes a spiral.

At the end of Dubins path the path is checked if the correct height has been reach. If the correct height has not been reached the path will create a spiral that converge to the correct height. When the correct height has been reach a last arc is created in order for the UAV to get the correct orientation.

The landing path can be created in two modes, which is manual or automatic. The manual mode allow the operator to design the landing path by defined the turning direction for the start and final circle. This enable the operator to specify a path that fills a operational demand, which would not necessary be the path when in automatic.

In automatic mode all four variants, which is given in table 2.2, calculated. After the length of the paths are calculated, where the shortest path is chosen to be the landing path. The landing path approach can be generated in to different way. One mode allow for manual deciding which side the start and finish circle should be in respect on the start pose, and the net landing approach. This allow the operator full control over the landing path, and can choose a landing path that is operation feasible and not necessary the shortest path.

Extra way point

To ensure that the path generation system will generate å flyable path an extra way point is added in the case of the start pose in cause an infeasable dubins path, or an spesial case of Dubins path which has not been implemented. The two case are

4.3 Evasion

To ensure the safety of the operator a evasion controller is used to abort the landing when a successful landing is deemed infeasible or improbable.

4.4 Guidance system

The guidance system consist of two part. Sliding mode controller, and a los controller

4.4.1 Sliding mode controller

For course control the system use a sliding mode controller that was proposed in the paper [Fortuna and Fossen, 2015], which USGES stability property.

$$\dot{x} = V_a \cos(\psi) + W_x = V_g \cos(\psi) \tag{4.4a}$$

$$\dot{y} = V_a \sin(\psi) + W_y = V_g \sin(\psi) \tag{4.4b}$$

$$\dot{phi} = \frac{g}{V_a} \tan(\varphi) \tag{4.4c}$$

$$\dot{\varphi} = -\frac{\varphi - \varphi_{cmd}}{T_\varphi} \tag{4.4d}$$

$$W = \sqrt{W_x^2 + W_y^2} \tag{4.4e}$$

$$\epsilon(\varphi) = \begin{cases} \cos(\varphi), & \text{if } |\cos(\varphi)| \geq \epsilon' \\ \epsilon', & \text{if } 0 \leq \cos(\varphi) < \epsilon' \\ -\epsilon', & \text{if } -\epsilon' < \cos(\varphi) < 0 \end{cases} \tag{4.5}$$

$$\chi = \tan^{-1}(\frac{\dot{y}}{\dot{x}}) \tag{4.6}$$

$$\dot{\chi} = \frac{g\sin(\varphi)(V_a + \cos(\psi)W_x + \sin(\psi)W_y}{\epsilon(\varphi)V_g^2} \tag{4.7}$$

$$\chi_d = \tan^{-1}(-\frac{t}{\Delta} \tag{4.8a}$$

$$\dot{\chi_d} = -\frac{\Delta}{\Delta^2 + y^2}\dot{y} \tag{4.8b}$$

$$\ddot{\chi} = \frac{\Delta}{(\Delta^2 + y^2)^2}(2y\dot{y}^2 - (\Delta^2 + y^2)\ddot{y}) \tag{4.8c}$$

$$\tilde{\chi} = \chi - \chi_d \tag{4.9}$$

$$s = \dot{\tilde{\chi}} + \lambda\tilde{\chi} \tag{4.10}$$

$$u = -\lambda\dot{\tilde{\chi}} - \rho\mathrm{sgn}(s) - K_d s \tag{4.11}$$

References

Brian A Barsky and Tony D DeRose. Geometric continuity of parametric curves: three equivalent characterizations. *IEEE Computer Graphics and Applications*, (6):60–68, 1989.

Lester E Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of mathematics*, 79(3):497–516, 1957.

Joao Fortuna and Thor I Fossen. Cascaded line-of-sight path-following and sliding mode controllers for fixed-wing uavs. In *Control Applications (CCA), 2015 IEEE Conference on*, pages 798–803. IEEE, 2015.

Marcus Frølich. Automatic ship landing system for fixed-wing uav. Master's thesis, Norwegian University of Science and Technology, NTNU, 2015.

Pratap Misra and Per Enge. *Global Positioning System: Signals, Measurements and Performance Second Edition.* Lincoln, MA: Ganga-Jamuna Press, 2011.

Joel Pinto, Paulo S Dias, Rui P Martins, Joao Fortuna, Eduardo Marques, and Joao Sousa. The lsts toolchain for networked vehicle systems. In *OCEANS-Bergen, 2013 MTS/IEEE*, pages 1–9. IEEE, 2013.

Robert Skulstad and Christoffer Lie Syversen. Low-cost instrumentation system for recovery of fixed-wing uav in a net. Master's thesis, Norwegian University of Science and Technology, NTNU, 2014.

Robert Skulstad, Christoffer Lie Syversen, Mariann Merz, Nadezda Sokolova, Thor I Fossen, and Tor A Johansen. Net recovery of uav with single-frequency rtk gps. In *Aerospace Conference, 2015 IEEE*, pages 1–10. IEEE, 2015.

Tomoji Takasu and Akio Yasuda. Development of the low-cost rtk-gps receiver with an open source program package rtklib. In *international symposium on GPS/GNSS*, pages 4–6. International Convention Centre Jeju, Korea, 2009.

Peter JG Teunissen. A new method for fast carrier phase ambiguity estimation. In *Position Location and Navigation Symposium, 1994., IEEE*, pages 562–573. IEEE, 1994.

Peter JG Teunissen. The least-squares ambiguity decorrelation adjustment: a method for fast gps integer ambiguity estimation. *Journal of Geodesy*, 70(1-2): 65–82, 1995.

Antonios Tsourdos, Brian White, and Madhavan Shanmugavel. *Cooperative path planning of unmanned aerial vehicles*, volume 32. John Wiley & Sons, 2010.

Bjørnar Vik. Integrated satellite and inertial navigation systems. *Department of Engineering Cybernetics, NTNU*, 2014.