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: May 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

Acronyms

DGPS Differential GPS.

GLONASS Global Navigation Satellite System.

GNSS Global Navigation Satellite System.

GPS Global Positioning System.

IMC Inter-Module Communication.

INS Inertial Navigation System.

LAMBDA Least-squares AMBiguity Decorrelation Adjustment.

RTK-GPS Real Time Kinematic GPS.

RTKLIB Real-Time Kinematic Library.

SIL Software In the Loop.

TOW Time Of Week.

UAV Unmanned Aerial Vehicle.

WGS-84 World Geodetic System, 1984.

Part I

Introduction and background theory

Chapter 1

Introduction

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.

1.1 Background and motivation

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 Skywalker X8

The Skywalker X8 is fixed wing UAV in a flying wing configuration, which indicate that the UAV has no tail and clear distinction between the wings and fuselage. The X8 is a popular choice for experimental missions at the UAV-lab at the Deparment of Engineering Cybernetic since it's durable, cheap and enough space to carry experimental payload.

1.3 Previous 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.

A continuation of the work done in [Skulstad and Syversen, 2014] was done in [Frølich, 2015]. The work simulated a autonomous net landing, however no physical experiment was perform. The result in the work indicated that further work on the controllers was required, in addition to the landing path which was not suited for a Visual Line Of Sight (VLOS) UAV operation due to no spacial restrictions.

1.4 Contributions

This thesis focus on the navigation system and generation of landing path in the autonomous landing system. The navigation system apply RTK-GPS to provide high accuracy position estimation, which is needed to perform a autonomous landing. The landing path provide a flyable path from any initial position, where the length and direction of the virtual runway is determind by the operator. Through this work, the following contributions has been made:

- A landing path has been created, which provide a flyable path from any initial position.[INKULDER REF]
- The landing path has been implemented in the DUNE runtime environment, which is capable to be used in both a stationary and moving net landing. As by product, the plug-in developed by Marcus Frølich [Frølich, 2015] was altered and used to specify the parameters used to create the landing path[INKULDER REF]
- A net nest has been constructed to provide the GPS coordinates for the stationary net, in addition to the heading of the net.[INKULDER REF]
- A navigation state machine, which is used to control which position solution source should be used in the payload computer. The state machine will try to keep the RTK-GPS available as long as possible by adding the average difference to the position solution from the Pixhawk for a short duration of time until a new viable RTK-GPS message is received.[INKULDER REF]
- A navigation source interface has been created to provide a visual indicator of which navigation system that is used in the DUNE environment.
- Finally statistical data on the performance of the X8 during landing during landing has been gather to be used in further work in further developing the landing 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, which 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:

$$\psi(s) = \operatorname{atan} 2(a_x, a_y) \tag{2.4}$$

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.

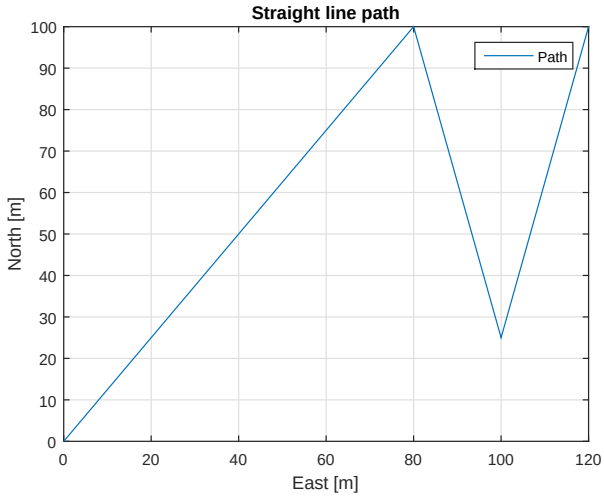


Figure 2.1: Straight line path

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. Such a path is Dubins path[Dubins, 1957], which showed that the shortest possible path for a particle that moved with unit speed with maximum curvature would consist of two circles and a straight line. The path consist of two arcs and a straight line. The straight line is tangential to both arcs. A disadvantage with Dubins path is that the curvature is discontinues, which gives a path from P_s to P_f with smoothness level of G^1.

A Dubins path that is constructed where the final orientation is fixed has four different ways to be constructed, which is determined by the rotation directions. The four types of Dubins path that is used in this thesis is given in table 2.2.

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

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

The equations that is used to construct the path is found in [Tsourdos et al., 2010] section 2.2.1. The path parameters is found by first finding the angle The start and end point of the straight line can be found with

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

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

which is used to find the tangent exit and entry point, where α is the angle between the length of the center line between the two circles, and the length of the line from the start circle to the exit tangent point. β is the angle of the center line. The exit and entry tangent point is found with the use of table **??**.

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

With the angle of the exit and entry tangent point the point is given as:

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

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

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

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

The length of the path is calculated in three parts. The first the is the arc length from the start pose to the exit tangent point, then the length of the straight line before the arc length from the entry point to the final pose. The length of the path is given as:

$$d = R_s\phi_s + d_t + R_f\phi_f \tag{2.7}$$

where d_t is the length of the straight line between the two circles, ϕ_s and ϕ_f is the arc angle for the start and final circle respectfully.

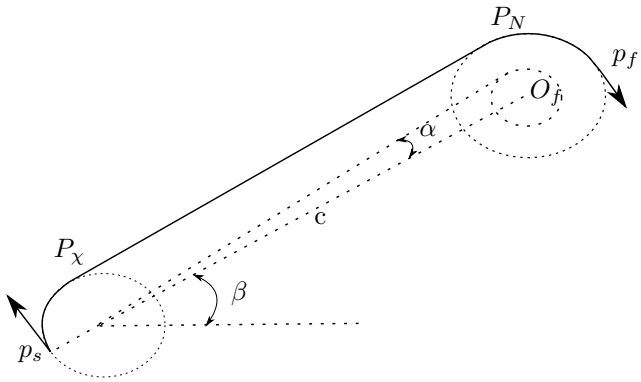


Figure 2.2: Dubins path

Chapter 3

RTKGPS theory

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.

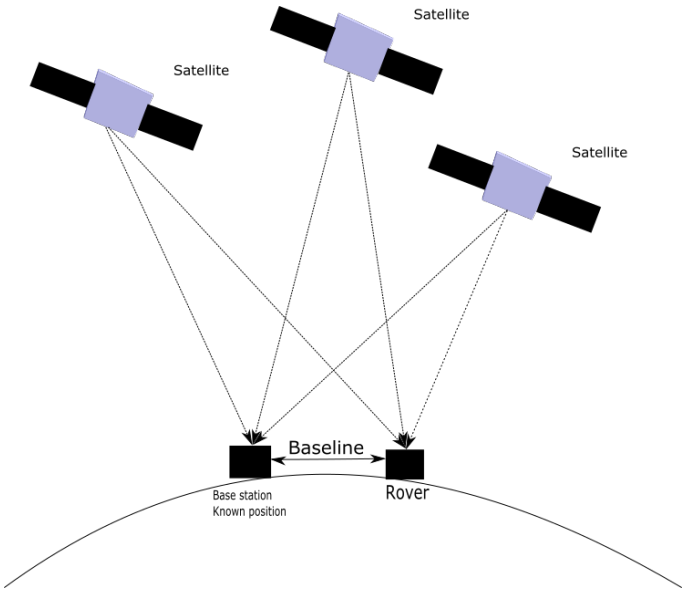


Figure 3.1: Concept figure of Differential GPS (DGPS)

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

Software

This chapter contain the software that is used for developing and testing the autonomous landing system for uav. The software that is mainly used in the uav system is based on a open-source software toolchain developed by the Underwater System and Technology Laboratory (LSTS). The toolchain supports air and ocean vehicle systems. The different components in the toolchain is IMC, DUNE, NEPTUS and Glued, which will be presented in later in this chapter.

The rtkgps solution in the system is calculated in the open-source software RtkLib [Takasu and Yasuda, 2009]. The desceription of the program is given in section **??**.

The low level control system in the uav is controlled by Ardupilot, which is a

4.1 LSTS toolchain

The software that the system is based on was developed by the Underwater Systems and Technology Laboratory (LSTS), which is called the LSTS toolchain [Pinto et al., 2013]. 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.

4.1.1 IMC

IMC [Martins et al., 2009] is design to enable interconnections between systems of vehicles, sensors and human operators, which enable the pursuit of common goal by cooperatively exchange real-time information about the environment and updated objectives. The message protocol is oriented around the message, which abstracts

hardware and communication heterogeneity with a provided shared set of messages that can be serialized and transferred over different means. The IMC protocol is defined in a single eXtensible Markup Language (XML) document, which simplify the definition of exiting messages and the creation of new messages. A single XML document ease communication between two node when both node use the same document for message definition.

4.1.2 Dune

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, which is the only way different DUNE tasks is communicating.

A DUNE task is enabled through a configuration file, where the user can choose in which profile the task should be enabled in. The different profile configuration in DUNE allows for testing the same system used in a hardware setting with a simulator.

4.1.3 Neptus

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. The main communication channel for Neptus is IMC, which makes it interoperable with DUNE or other IMC- based peer.

Neptus is able to do MRA (Mission Review and Analysis) after a mission is finished. In the MRA phase Neptus analyse the IMC logs that is collected by e.g. DUNE, such that the result from a completed mission can be presented. In addition Neptus mission review is able to create output files of the log that can be analysed in third party software like Matlab.

4.1.4 Glued

Glued is a minimal Linux operating system distribution, and design with embedded system in mind. It is platform independent, easy to configure and contain only the

necessary packages to run on a embedded system. This makes GLUED a light and fast distribution, which is ideal for a on-board operating system for a unmanned system where payload size is normally limited. GLUED is configured through a single configuration file that which can be created for a specific system.

4.2 RTK-GPS system

The RTK-GPS solution is calculated in the open-source program liberary RTKLib [Takasu and Yasuda, 2009]. The X8 runs the rover program rtkrcv which is connected over TCP/IP with the base station program str2str. Both programs are connected to a Ublox Lea M8T GNSS receiver. The GNSS receiver is programmed to send raw GNSS data to both programs, where the base station program str2str transmits without alteration the raw data to the rover program rtkrcv. The structure of the RTKLib software configured with a rover and base station is shown in figure 4.1.

The navigation system require to now the reference position of the base station in order to use the RTK-GPS solution. However the base station position is currently not part of the output message from rtkrcv. This is resolved by allowing the base station to calculate it's own position as a standalone GPS. The GPS position is transmitted to a local Dune task on the base station, where the operator can decide when the base station can be considered as fixed. When the base station is considered fixed the position is sent to the X8, where it's included in the RTK-GPS solution message.

4.2.1 RTKLIB

Real-Time Kinematic Library (RTKLIB)[Takasu and Yasuda, 2009] is a open source program package for standard and precise positioning with GNSS developed by T. Takasu. RTKLIB can be configured to apply RTK-GPS, such that raw GNSS data is used estimate the relative position of the rover with respect to the base station in real time. Figure 4.1 shows how RTKLIB can be used in a RTK-GPS mode. The two main modules here is str2str and rtkrcv. Both will be explained more closely in the following sections. The version of RTKLIB used in this thesis is RTKLIB2.4.2 [RTKLIB].

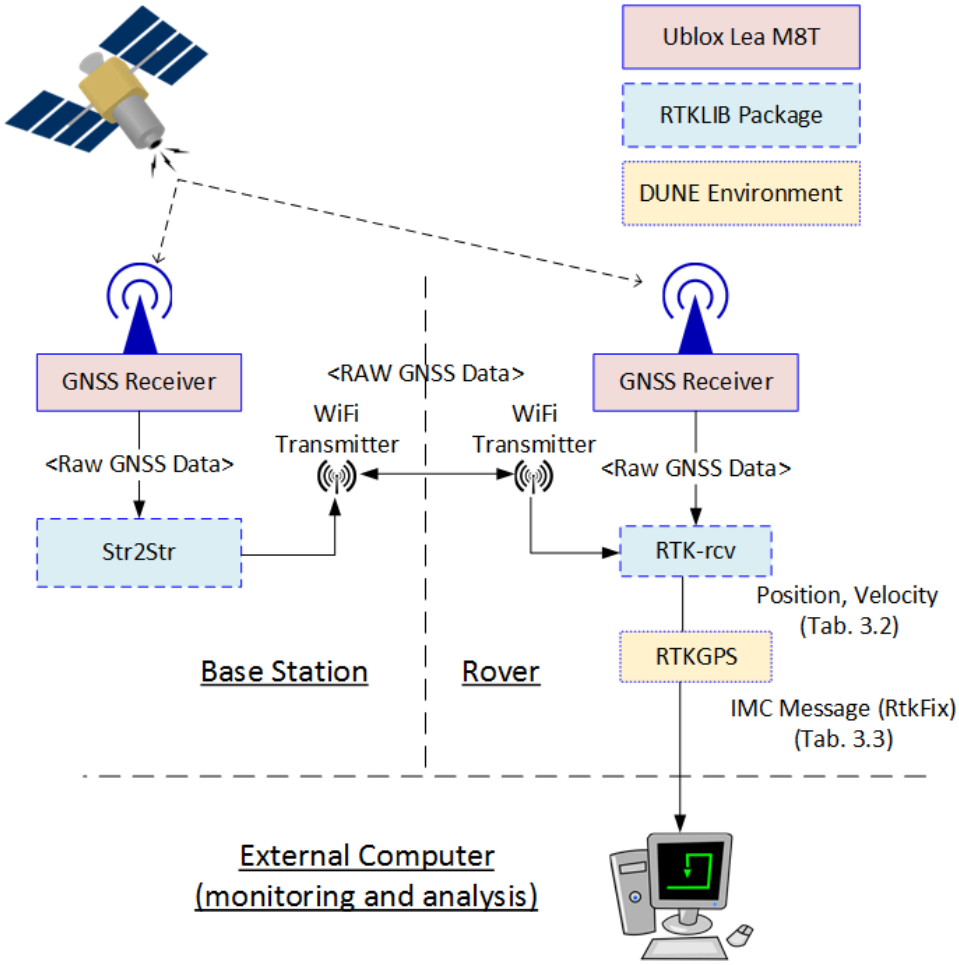


Figure 4.1: The communication structure of RTKLIB

Rtkrcv

As part of the RTKLIB Rtkrcv is used to calculate the position of the rover in real time. Rtkrcv can be configured to have two output streams. It's desired in a automatic landing system to have a velocity estimate. However this is not provided in the newest version of RTKLIB, and therefore a altered version of RTKLIB is used in the navigation system where the velocity is part of the output data. The position output is in ENU format and the full output structure is presented in table 4.1

Header	Content
1 Time	The epoch time of the solution indicate the true receiver signal reception time. Can have the following format: yyyy/mm/dd HH:MM:SS.SSS: Calender time in GPST, UTC or JST. WWWW SSSSSSS.SSS: GPS week and TOW in seconds
2 Receiver Position	The rover receive antenna position
3 Quality flag (Q)	The flag which indicates the solution quality. 1:Fixed 2:Float 5:Single
4 Number of valid satellites (ns)	The number of valid satellites for solution estimation.
5 Standard deviation	The estimated standard deviation of the solution assuming a priori error model and error parameters by the positioning options
6 Age of differential	The time difference between the observation data epochs of the rover receiver and base station in second.
7 Ratio factor	The ratio factor of "ratio-test" for standard integer ambiguity validation strategy
8 Receiver velocity	The velocity of the rover. Given only when output is in enu format

Table 4.1: Rtklib output solution format

Str2str

Str2str is used as a base station program that can receive raw GNSS data and further export data over tcp, set-up by str2str. Str2str sends out Radio Technical Commission for Maritime Service 3 (RMTC3) formatted messages, however it can be configured to send whatever comes in as input. The communication between str2str and rtkrcv is shown i figure 4.1

4.3 Ardupilot

Part II

Method

Chapter 5

Landing path

The path generator in the autonomous landing system is designed to enable UAV landing in both a stationary or moving net. The path is created in two main stages. The first is the creation of the virtual runway, which is defined as a straight line along the heading of the net. The second stage apply a lateral Dubins path 2.2 and longitudinal straight line path to create a path that ensures that the UAV is able to enter the straight line along the virtual runway at the correct height and attitude. 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.

5.1 The virtual runway

The virtual runway is inspired by the work done in [Skulstad and Syversen, 2014] where waypoint was used to create a straight line path towards the net. This method proved successful, and since the UAV descent towards the net should be as controlled as possible only small angles is used when transitioning between way-points. The straight line path is constructed relative to the net as shown in figure **??**, with

way-points given as:

$$\mathbf{WP4} = \begin{bmatrix} -a0 \\ 0 \\ h_{nc} + a1\tan(\gamma_a) \end{bmatrix} \tag{5.1a}$$

$$\mathbf{WP3} = \begin{bmatrix} a1 \\ 0 \\ h_{nc} - a1\tan(\gamma_a) \end{bmatrix} \tag{5.1b}$$

$$\mathbf{WP2} = \mathbf{WP3} + \begin{bmatrix} a2 \\ 0 \\ a2\tan(\gamma_d) \end{bmatrix} \tag{5.1c}$$

$$\mathbf{WP1} = \mathbf{WP2} + \begin{bmatrix} a3 \\ 0 \\ 0 \end{bmatrix} \tag{5.1d}$$

were the description of the parameters used is given in table 5.1. The net is placed between the fourth and third 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 ease the demand of the controllers used in the landing system.

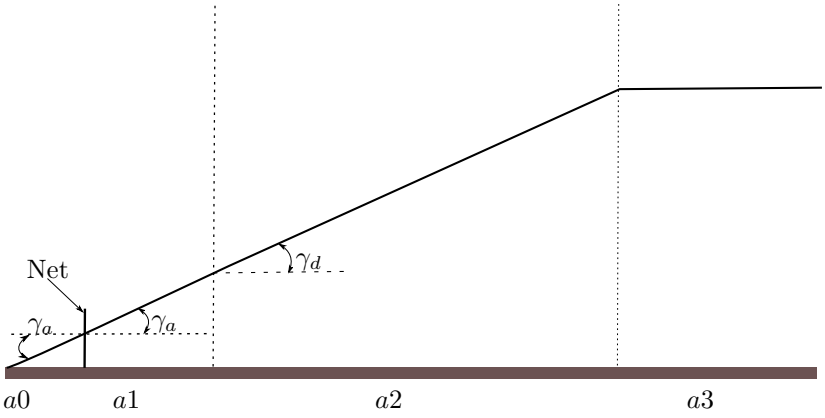
Parameter	**Description**
$a0$	The distance behind the net
$a1$	The distance in front of the net
$a2$	The length of the glide slope
$a3$	The length of the approach towards the glide slope
γ_a	The net attack angle
γ_d	The glide slope angle

Table 5.1: Net approach parameters

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

$$WP^n = R(\psi_{net})WP^b \tag{5.2}$$

were ψ_{net} is the heading of the net, and $R(\psi_{net})$ is the rotation matrix around the z-axis.



5.2 The landing approach

In order to ensure that the UAV follows the path along the virtual runway it must be at the correct height with the correct attitude from any initial position. In addition it's desirable the the decent angle is kept small to ease the strain on the control system.

The landing approach consist of a Dubins path in the lateral plane and a straight line path in longitudinal plane.

5.2.1 Lateral path

Dubins path was chosen due to it's simplicity in description and it fulfils the requirement that it's smooth enough for path following.

5.2.2 Longitudinal path

The lateral path was chosen for the autonomous landing system is Dubins path, due to it's simplicity in description and it fulfils the requirement that it's smooth enough for path following. Since it's desired for the UAV to decent with small decent angle the longitudinal path is constructed with straight lines. Combined with Dubins path, the circles in the start and end of the path becomes spirals. This increase the demands on the controllers used, since they has to both control the heading in addition to the decent rate with the same control surface. This is the main reason for small angles in the longitudinal path.

The path generation system is designed such that the operator can choose how the path should be generated. The system allows for manual creating of the path,

or simply create the shortest path from the start pose to the end pose, which is the first way-point in the path towards the net.

Chapter 6

Navigation system

The navigation system apply RTK-GPS for position and velocity measurement, which provide higher position accuracy then a standalone GPS. However the state of the RTK-GPS can change before or during a mission. Therefore the operator must be able to monitor the state of the RTK-GPS in Neptus, which is a command and control software used to create and monitor missions. In addition to monitoring the navigation system must be able to handle a drop out of the RTK-GPS. This is handled in two steps. The first is a short loss compensator stage, where the backup standalone GPS is compensated to get a position solution closer to the RTK-GPS solution. The second stage fully disconnect the RTK-GPS.

6.1 Navigation state

The navigation system consists of five states, which is controlled by a state machine. The state transitions is determined by what's available and what the operator has specified should be used. The output from the state machine is the IMC message EstimatedState, which is the only state source used in the Dune control and guidance system. The state machine also dispatch a IMC message that informs which the source used in the navigation system, and which sources is available. Currently only the RTK-GPS system is considered as a internal system, however this can be expanded to include other sensors used in the Dune system.

The input to the navigation system is IMC message ExternalNav and GpsRtkFix. The ExternalNav message is the primary state source when the RTK-GPS is not in use, and it's receive the state information from the Pixhawk mounted in the X8.

During a short loss of the RTK-GPS the position solution in the ExternalNav message is compensated to avoid sudden change in position. The short loss compensator is explain further in 6.1.1. The short loss system is implemented to avoid drop

out of the RTK-GPS when a message is delayed, or its struggling due to the dynamic behaviour of the UAV.

The state machine is depicted in figure **??**, with the edge describtino given in table 6.1.

Edge	**Event**	**Guard**
A	Event: Received External Nav message	None
B	Time out: Fix RTK-GPS solution for x seconds	None
C	Time out: x seconds since last valid GpsFixRtk	None
D	Flag: Using Rtk is set true	None
E	Flag: Using Rtk is set false	None
F	Time out: x seconds since last valid GpsFixRtk Event: Received GpsRtkFix with $type == None$	Short loss compensator: Enabled
G	Event: Received valid GpsFixRtk message	None
H	Time out: x seconds since last valid GpsFixRtk Event: Received GpsRtkFix with $type == None$	Short loss compensator: Disabled
I	Time out: x seconds since last valid GpsFixRtk	None

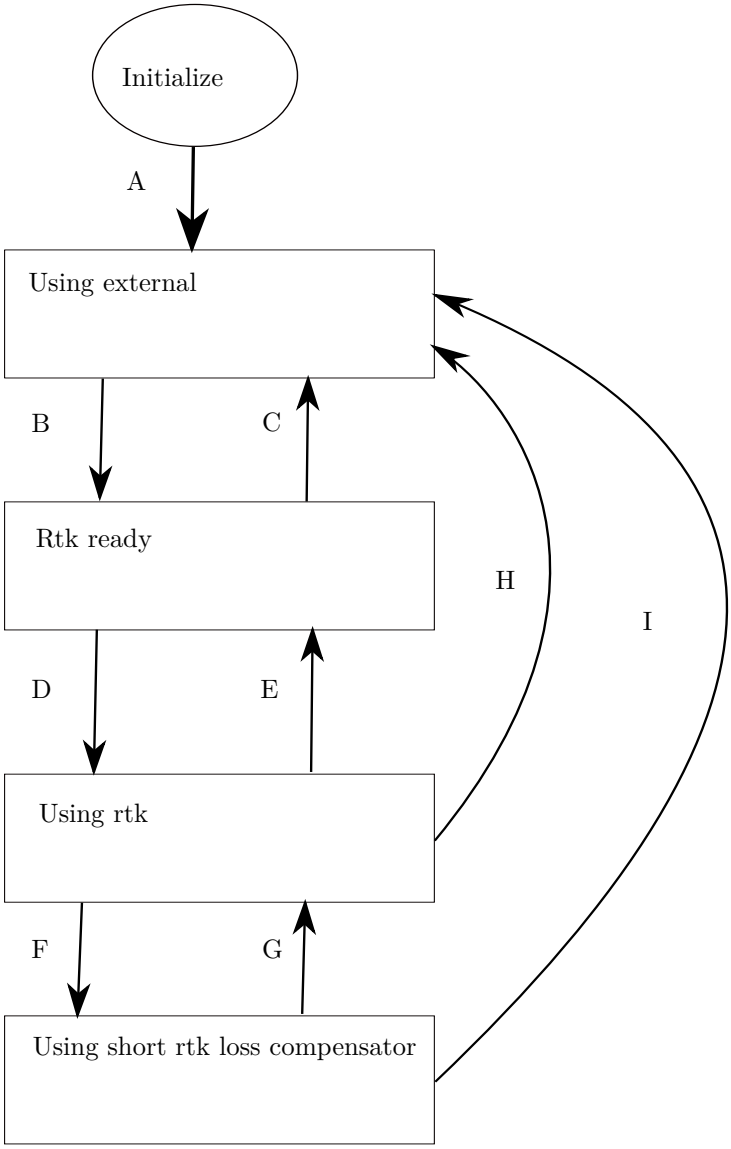
Table 6.1: Net approach parameters

6.1.1 Short loss of RTKGPS

In the event of RTK-GPS drop out a offset can be added to the position solution in order to prevent a sudden change in position. The offset is defined as the average difference between the N latest position solution from the RTK-GPS and the external navigation system:

$$offset = \frac{1}{N}\sum_{n=0}^{N}(RTKGPS(n) - External(n)) \tag{6.1}$$

where the RTK-GPS solution is displaced into the External nav system. However during the implantation it was discovered that the standard displace function in the DUNE literary was inaccurate in correctly calculating the hight of a offset point. The problem was that the geodetic latitude calculation was calculate assuming that the Earth has a spheric shape. This was solved by creating a new displace function where the geodetic latitude is calculated in according to


Initialize
A
Using external
B
C
Rtk ready
H
I
D
E
Using rtk
F
G
Using short rtk loss compensator

Chapter 7

Implementation

This chapter contain the technical specification of the navigation system and landing path system, in addition to the payload installed in the X8 and both nests.

7.1 Landing path

The landing path is specified through the Neptus plug-in LandmapLayer, which allow for real time setting of the parameters used to create the landing path. In addition the plug-in display the position of the net in the graphical map interface which is a part of Neptus. The plug-in dispatch the IMC message, LandingPlanGeneration, which has been created specifically for landing path purpose.

The IMC LandingPlanGeneration message is dispatch to the Dune system, where it's picked up by the Dune task LandingPlan. This trigger the generation of the landing path, and depending on the configuration of the Dune task and desired setting in the LandingPlanGeneration message different landing path is generated. The different variants is categorised into two groups; configuration of the Dubins path, and configuration of the virtual runway. The configuration option that result in different variants of the landing path is given in table **??**. In addition the Dune task can be configured for a dynamical landing. When configured as dynamical it's assumed that an other task is responsible for the final landing part. Therefore the vertical runway is not part of the plan, however the first part of the path is still included. When generated the plan must be started from Neptus, where the desired control configuration is added to the path.

A loiter manoeuvre can be included in the landing plan before the start of the path along the virtual runway. The loiter manoeuvre can be used to control when the UAV should start it's final approach, which is practical when performing a dynamical landing.

Parameter name	Action
Automatic (boolean)	If true a standard path where the shortest Dubins path is chosen. Otherwise a user specific path is chosen
Start circle turning counter clockwise (boolean)	If true the start arc is created such that the turning direction is counter clockwise. Otherwise clockwise. Require Automatic==false
Finish circle turning counter clockwise (boolean)	If true the finish arc is created such that the turning direction is counter clockwise. Otherwise clockwise. Require Automatic==false
Wait at loiter (boolean)	If true a unlimited loiter is included in the path before the path continue with the path along the virtual runway.

7.2 Navigation system

The navigation system is control by a state machine 6.1, which is used to control the content of the output IMC messages EstimatedState and NavSources. Depending on which state the navigation system is in the EstimatedState message will either have position solution form the RTK-GPS system or the external navigation system. During a short loss of the RTK the external navigation position is compensated with the average difference between the RTK solution and the external navigation solution.

7.2.1 Operator interface

The state of the navigation system is monitored though a interface in Neptus. The interface indicate which source the Dune system is using for state information. 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 7.1.

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

Table 7.1: Net approach parameters

7.3 Nest system

A nest system is a stationary unit with the sole purpose of providing it's position to the rest of the Dune System. As part of the navigation system the base station is defined as a nest, where the GPS position is sent to the RTK-GPS system when fixed. An other nest has been created to obtain the GPS position of the stationary net. The net nest is configured as a rover in RTK-GPS configuration, such that the position relative to the base station is in the same frame as the X8.

7.4 X8 and nest payload

The X8 and both nests are installed with a BeagleBone embedded computer which is used to run the Dune system, as well as rtklib. Rtklib is connected to Ublox Lea M8T GNSS receiver with a uart cable, which is configured with a output rate of 10Hz in all systems.

The Ublox GNSS receiver in the X8 is connected to a [NAVN ANTENNE] which is mounted on top of the X8. The Ublox in the base station nest is connected to [NAVN ANTENNE], and the ublox in the net nest is connected to [NAVN ANTENNE].

The autopilot is a Pixhawk, which runs ArduPilot. The pixhawk is connected to the beaglebone. The connection between the X8 and the nest relay on wifi connection with M5 Rockets which is mounted all systems. The communication is done with both TCP and UDP depending which message is sent.

Part III

Experiment

Chapter 8

SIL test

The system was tested through a Software In the Loop (SIL) simulation. In a SIL simulation the same code that runs in the hardware is tested against a simulator which behave similarly to the real system.

8.1 Path planing SIL

The landing path is tested in both guided and FBWA.

8.2 Navigation SIL

The navigation system was tested by creating a RTK-GPS simulator. This allowed for testing of the navigation interface, including the fixing of the base station position.

8.2.1 Short rtk loss compensator

The short rtk loss compensator was tested by adding a bias in the position solution from the simulated GpsFixRtk message.

Chapter 9

Experiment

9.1 Path

9.2 Navigation

9.2.1 Short loss compensator

Include result when short loss was not in use. Compare to result when it's in use.

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.

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

Ricardo Martins, Paulo Sousa Dias, Eduardo RB Marques, José Pinto, Joao B Sousa, and Femando L Pereira. Imc: A communication protocol for networked vehicles and sensors. In *Oceans 2009-Europe*, pages 1–6. IEEE, 2009.

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.

RTKLIB. Rtklib ver. 2.4.2 manual. http://www.rtklib.com/prog/manual_2.4.2.pdf. Accessed: 18.12.2015.

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.