TTK4530

AUV Pipeline Following and Inspection

Anders Garmo

December 19, 2008

NTNU
Norwegian University of
Science and Technology

Abstract

This report considers the problem of pipeline inspection using an AUV. A basic controller and guidance system for a two-dimensional path following problem are derived using traditional control theory. The guidance scheme proposed is a lookahead type algorithm, and its performance are discussed.

The decision system and flow of the inspection and search process is analysed. Some search patterns are proposed and discussed with regard to energy optimality.

A Kalman filter was derived to smooth the sensor output and help predict where the pipeline are, if the pipeline is buried or sensor data otherwise are unavailable.

Preface and Acknowledgements

This report is the final report of the fall project at the Norwegian University of Science and Technology. This project is to prepear the student for the master thesis the next semester. It is a project on how to write a project, so to speak. My opinion is that it is better to do the faults and wrongdoings here, than during the master thesis work. This project can be used as a stepping stone and a pre-study of the problem to be undertaken in the master thesis.

For me this project has been a project to get to know all the problems associated with having an Autonomous Vehicle roaming the sea depths. We humans, as sentient beings are perfectly capable of taking the numerous decisions where to go and what to do next. An Autonomous Vehicle needs to be programmed, and for every single decision it needs to make, there must be some kind of rule. This makes the project huge and it quickly gets out of hand for the designer working with it, and so it did for me, too.

To create this kind of system, much testing are needed, and a team of engineers to possibly be able cover all the aspects associated with a pipeline inspection mission. Kongsberg Maritime has been working on *HUGIN 1000* for more than 15 years, and designing an AUV for this kind of application are not done in half a semester at NTNU.

I would like to thank Bjørn Gjelstad and Øystein Engelhardsen at AUV R&D department, Kongsberg Maritime, for inviting me to the to Kongsberg Maritimes premises at Horten and showing me what the *HUGIN 1000* AUV really looks like.

Anders Garmo
December 18 2008

Contents

Preface and Acknowledgements **iii**

Contents **v**

List of Figures **vii**

Abbreviations & Notation **ix**

Introduction **xi**

1 Theory **1**
1.1 Pipeline Inspection Fundamentals 1
1.2 Different Pipeline Inspection Schemes Discussed in Literature . 2
1.3 Reference Systems 3
1.4 Hydrodynamic Model 4
1.5 Camera Theory 4
1.6 Kalman Filter 6
1.7 Guidance 6

2 Modelling **11**
2.1 Problem Outline 11
2.2 Pipeline Representation 12
2.3 Kalman Filter 12
2.4 Controller Design 14
2.5 Summary of Assumptions 17
2.6 Guidance System 17
2.7 Overall System 20

3 Simulations and Results **23**
3.1 Matlab 23
3.2 Simulation Scenarios 24
3.3 Results 25

4 Discussion **35**
4.1 Guidance System 35
4.2 Roll Stabilisation 36
4.3 Energy Consumption 37
4.4 Optimal Search Pattern 39
4.5 Validation of the Results 39

5 Conclusion **41**
5.1 Further Work 42

References **43**

Appendix: Matlab Scripts **45**
The Descision block 45
Camera Simulator 47
Kalman Filter 49

List of Figures

1 *HUGIN* Vehicle xii

1.1 Pinhole Camera Model showing two dimensions 5
1.2 Variables associated with path following 7

2.1 Block diagram the path following controller 12
2.2 Guidance System Block 17
2.3 Flow diagram of the search procedure in the *searching*-mode . . . 18
2.4 A turning trajectory for one possible contract 19
2.5 Differnet search pattern 20
(a) Divergent Zig-Zag pattern for lost pipeline during tracking 20
(b) Outwards spiral pattern for initial pipeline search 20
2.6 Flow diagram of pipeline inspection mission 20

3.1 The Simulink Diagram of the implemented Guidance System . . . 24
3.2 The Forces associated with the AUV maneuvering 25
(a) Coriolis Forces 25
(b) Damping Forces 25
3.3 North East path of AUV without Current 26
3.4 Surge-, Depth- and Heading- Reference vs. Actual Values 27
3.5 Trajectory plots of the 2nd scenario 27
(a) NE Path with Waypoints 27
(b) NE path with Heading 27
3.6 Reference plots for the 2nd scenario 28
3.7 Trajectory plots from the 3rd scenario 29
(a) NE Path with Waypoints 29
(b) NE path with Heading 29
3.8 Trajectory plots for the 4th scenario 30
(a) NE Path with Waypoints 30
(b) NE path with Heading 30
3.9 Reference plots for the 4th scenario 31
3.10 Filter Test 32

4.1 Plots describing the close relation of yaw moment and roll angle . 36
(a) Roll Angle 36
(b) Yaw Moment 36

Abbreviations & Notation

AUV	Autonomous Underwater Vehicle
CB	Centre of Buoyancy
CG	Centre of Gravity
DOF	Degree of Freedom
DVZ	Deformable Virtual Zones
ECEF	Earth-Centred Earth-Fixed
LOS	Line-of-Sight
MPC	Model Predictive Control
MPG	Model Predictive Guidance
NED	North-East-Down
PNG	Proportional Navigation Guidance
P_c	Point decomposed in the camera frame
P_i	Point represented in image coordinates
P_w	Point decomposed in world frame (NED frame)
ROV	Remotely Operated Vehicle
$\mathbf{x}_c$	2D point decomposed in the camera frame
$\mathbf{x}_i$	2D point in image coordinates

Introduction

Today there are over 3.5 million km of transmission pipelines in the world. 231 900 km pipelines are planned or under construction [DNV]. Pipelines currently in place need to continue operating, many of them far longer than they were initially intended. Great effort are taken to inspect and maintain the pipelines to keep them in satisfying order. The pipeline needs to be in working order to be operational. Potential leaks might cause irreversible environmental damages, and companies looses money when the flow through the pipeline are stopped.

There are two ways of inspecting pipelines; *internal* and *external* inspection. The internal inspection methods, includes stopping the flow in the pipeline, open it and insert a Pipeline Inspection Gauge (PIG) which travels inside the pipeline and uses various sensors to determine the state of the pipeline. The other method, *exterior* pipeline inspections are today mostly done using Remotely Operated Vehicles, ROVs. The ROVs are tethered unmanned underwater vehicles, and are the work horse of the offshore industry. They are versatile tools capable of accomplishing most missions associated with pipeline inspections and repair. However, they need well-equipped, expensive support vessels and a large crew to accompany the inspection mission which of course are very costly for the pipeline owner. British Petroleum stats that the ROV are "overactuated" for the pipeline inspection case.[PK04]

An Autonomous Underwater Vehicle (AUV) is a suiting tool for pipeline inspection. An AUV is a untethered unmanned underwater vehicle. Opposed to the ROVs, which operation radius are limited by the tether, the operation radius of the AUV are limited by power consumption and battery life. An AUV comes in many forms, small or large. It may need minimal support crew which will minimise the cost for the pipeline owners. It can be made small enough to be launched from small ships, and even from shore to inspect pipelines going to the oil refinery on land. This would be a great advantage for the pipeline owners to cut costs, instead of hiring large support vessels and crew to support a pipeline inspection mission.

The speed of the inspections is also an important issue. A typical ROV has cruise speeds from $1-2$ knots ($0.5-1$ m/s), while an AUV has cruise speeds in the regime $2-6$ knots ($1-3$ m/s). This inclines that an AUV might cover larger area of pipelines than an ROV.

British Petroleum estimates that they can save up to 30 % by using AUVs for pipeline inspections instead of ROVs, which motivates the research in the area. The market for AUV doing pipeline inspections are present. The technology needed are present, and there are a number of companies developing AUVs for pipeline following. SeaByte and Subsea7 have conducted a successfull pipeline inspection mission using the *Geosub* AUV and claimed the world record in the longest uninterrupted pipeline inspection mission. The AUV inspected 22.2 km of pipeline at 4 knots without being interrupted. [Sea], [Bø08]

This report will look at the possibility to give the Kongsberg Maritime *HUGIN 1000* AUV the given abilities to track and follow a subsea pipeline. The AUV have hovering capabilities and are controllable in 5 degrees of freedom (DOF).The AUV are assumed stable in the *roll*. Although, the hovering abilities are present, this will not be used when designing the guidance system, because of the focus on energy efficiency.

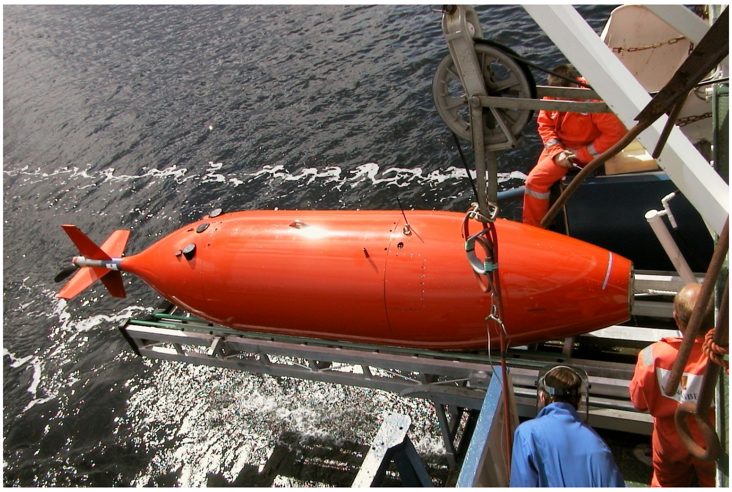


Figure 1: *HUGIN 3000* under launch. *HUGIN 1000* are somewhat smaller than *HUGIN 3000* but identical in shape. *xpda.com*

The pipeline detection equipment is a downward looking camera with sufficient lighting to operate at about 3-5 meters above the seabottom. Of course the visibility conditions will change according to depth and amount of particles in the water. This is probably an optimistic assumption, and at great depth the visibility will probably be less than 3 meters.

This report considers how to automate the pipeline inspections process. It will consider the possibility that the pipeline is buried under mud and not visible to the camera or buried on purpose to make it more robust towards environment forces and erosion, in both cases the important thing is to reacquire the pipeline at the end of the buried stretch. In that case some kind of an estimator have to be used to predict where the pipeline is headed.

A pipeline inspection mission can be divided into 4 parts:

1. Initialisation and initial descent
2. Search for and Acquire pipeline
3. Track and Inspect pipeline
4. Ascent to the surface and deliver the acquired data

The *Initial descent* and *final ascent* are self explanatory. But what happens if the pipeline are not at the initial position when, when the AUV arrives at that location? It is obvious that some kind of search need to be initiated.

The *Search and Acquire* part will need the vessel to move in some kind of search pattern if the pipeline is not located exactly where the initial position data states it to be. This search pattern should also be used if the pipeline is lost during tracking.

The *Track and Inspect* part needs a guidance system which is capable of keeping the vessel on top of the pipeline or in the vicinity, independent of the current in the area and other possible disturbances. This is necessary because the primary mission for the AUV is to provide video of the pipeline which can be used to determine the state and well-being of the pipeline.

This report is divided into four chapters;

1. Theory. Describes the necessary theory needed to understand the proposed guidance system, and contains a summary of literature on the pipeline following subject.

2. Modeling, assumptions about the model and how it is modeled. A Kalman Filter and a flight-mode controller will be derived. A guidance system will be proposed and desired behaviour will be addressed.

3. Implementation and simulation of the proposed system. How the system is implemented and how it performs in certain situations will be treated in this chapter.

4. Discussion of the results given by the simulations, and various aspects about the proposed guidance system will be discussed. A review of how valid the results will also be given here.

5. Summary of the results, together with conclusion and future work will be treated in the last section.

Chapter 1

Theory

Pipeline inspection are a complex problem and include many engineering disciplines. This chapter introduces some of the basics about pipeline inspections when designing a guidance system capable of pipeline inspection. Basics on the mathematical model for *HUGIN 1000* will treated, together with some theory on how to model a camera view and how movement of points are described in the camera view.

1.1 Pipeline Inspection Fundamentals

It is obvious that sensors are very important when designing a guidance system for an AUV capable of pipeline inspection. The AUV must be able to "see" the pipeline and move too it. It is also very important that the guidance system treats the information from the sensors in the right way, so weighting of the sensors will be an important issue. The standard sensor suite of the *HUGIN 1000* AUV are amongst others[Gje08]:

- GPS aided High Precision Acoustic Positioning (HiPAP)
- Velocity Aided Inertial Navigation System
- Doppler Velocity Log
- Depth Sensor
- Magnetic and Gyro Compass
- Sidescan Sonar
- Multibeam Echosounder
- Synthetic Aperture Sonar

Sensors like the Multibeam Echosounder, Sidescan Sonar and Synthetic Aperture Sonar are suited for pipeline inspection.

Sound is the prime medium in underwater applications. Sound waves in water travels roughly 5 times as fast than in air. This have given raise to many sensors utilising sound under water. One of this sensors is the Sidescan Sonar which operates in the 100-500 kHz frequency area. The sonar emits a fan-shaped pulse of sound towards the sea floor. The echo of each beam are recorded in cross-track lines, representing slices. As the sonar moves forward these slices can be assembled into a greyscale picture, making a profile of the bottom. Dark spaces corresponds to poorly reflecting materials, while bright spots are highly reflective material. Pipelines and other man made structures show up on the sonar as distinct shadows, and with processing of the image they are relatively easy to detect, even at large distances. A Sidescan sonar can be effective on distances as large as 300 meters, which makes this the primary sensor for locating pipelines.[FC90] [PRB02]

A Multibeam Echosounder is another sensor suited for pipeline location and tracking. Usually an echosounder emits one beam of sound and measures the time of travel of the sound beam. A multibeam echosounder emits a fan of sound beams downwards, which can be used to map a larger seabottom area around the vessel. This is useful when tracking the pipeline and can help with pipeline identification. The use of á priori information about the pipeline, such as diameter of the pipeline can help the recognition steps. This sensor can be used for detecting freespans if the length to the sea bottom should be much longer than the length to the pipeline.

1.2 Different Pipeline Inspection Schemes Discussed in Literature

A literature study was performed to see what other people have accomplished on the subject on AUV and pipeline following.

[Hal91] describes briefly a prototype AUV equipped with a camera and sonar to carry out pipeline inspection missions. The goal of this prototype was to prove that AUVs could be useful in pipeline inspection missions. It is a light-weight AUV for about 40 kg and about a meters long. The AUV uses controllers for heading, speed and depth and visual guidance to follow the pipeline. This is a part of the research SINTEF Automatic Control did in the early 1990s and proved that there are applications that AUVs will be superior to ROVs.

[NSAB03] proposes to use a two-stage guidance problem. When submerging and ascending to the surface, the AUV uses a LOS guidance law at full speed. When the pipeline is acquired, the guidance law are changed to a visual guidance scheme, which allows for a precise guidance over the subsea cable or pipeline. This is a good principle, which probably will prove robust towards ocean currents, but the downside is that cameras have poor visibility at great sea depths.

In [NRS04] a guidance system using Model Predictive Control (MPC) and PNG law was used to make a guidance system able to track cables and pipelines on the sea bed, using Doppler Velocity Log and Electro-Magnetic sensors to find the cable/pipeline. The authors utilises a pure pursuit guidance scheme, similar to what predators do when they are hunting pray. The cable/pipeline and AUV is formulated as moving points with mass. The AUV engages in a tail chase with the pipeline "point", but the AUV never catches up with it. This method is robust to model uncertainties and handles constraints of the vessel in a systematic way because of how MPC are formulated. The controller however, shows problems when a current are introduced in the system. The AUV drifts of the pipeline, but catches up with the waypoints in the end. This is due to the fact that the guidance system are in a pursuit with the pipeline and the trajectory the AUV follows are not important. The guidance system only tries to catch up with the pipeline “point”.

In [For01] a visual guidance system for inspection of underwater structures are presented. The visual system uses an Extended Kalman Filter to smooth and predict where the structure, i.e. pipeline, will move in the next sample interval. This will greatly speed up the image processing. A three dimensional model of the scene is constructed, which then allows the guidance system to calculate the input to the controllers. The article presents a very robust vision guidance system. The Extended Kalman Filter implemented gives good results, and the maximum error in xz-plane with current are around 3 meters. This can be improved by adding depth measurement. The system can process image frames at 4 frames a second which gives good resolution for the scene reconstruction algorithm.

[PJL05] proposes a reactive control approach to pipeline tracking, using a profiling sonar. Reactive control originated from the field of obstacle avoidance in autonomous land and air vehicles. It uses *Deformed Virtual Zones* (DVZ) which describes the interaction between the AUV and the pipeline. The controller tries to minimise the deformation of the DVZ and calculates a feasible control input. The DVZ in this case is a prism with a cylindrical cavity directly underneath the AUV. If the AUV moves away from the pipeline the DVZ will be deformed and the controller will try to counteract the motion. This is a computational inexpensive way of achieving a good pipeline following. This method shows promising results but has yet to be implemented and tested in real-life scenarios.

In [KU03] a fuel-optimal tracking controller is derived to minimise the fuel consumption of the AUV. It uses the fact that the least fuel consuming path is the shortest one. This paper derives a fuel optimal controller using the estimated fuel consumption as a minimisation index.

[PRB02] describes two techniques for detecting and tracking pipelines using Side Scan Sonar and Multi-Beam Echosounders. Prior knowledge about the pipeline are utilised for the recognition of the pipeline. A pipeline creates very distinct shadows in a sidescan sonar image and can be with some processing, easy to detect even if the pipeline are far away. An algorithm are proposed for this recognition and separation of the pipeline structure from the background. It spearates the sidescan sonar image into shadow regions and non-shadow regions. These are matched to pipeline-like characteristics to determine if it really is a pipeline. The simulations shows that the mean error in position when detecting a half buried pipeline are about 0.026 meters, 0.13 meters when the pipeline are trenched and 0.22 when the pipeline are spanned. When the system are tracking the pipeline the errors are even less, around 0.0104 in the half-buried pipeline case.

1.3 Reference Systems

Movement must be described relative to something. This is the task of the reference systems. There are 4 important reference system. The ECI (Earth Centre Inertial) which is a truly inertial reference frame, i.e. it is not accelerating. Its axis are pointing through the north pole and through the equatorial line of the Earth, fixed toward stationary points in space. The centre is as the name suggests in the centre of the Earth.

ECEF (Earth-Centre Earth-Fixed) is another reference frame. It is defined same as the ECI coordinate system, but the axis are rotating with the same rate as the Earths rotation rate. This means that this frame is not strictly inertial, but the angular velocity of the Earths rotation are considered very small, and can be neglected compared to other velocities in the same frame. Position on the earth are described by *longitude* and *latitude*.

An important reference frame when considering local motion are the NED (North East Down) frame. This frame is defined as the tangent plane on the current position, and moving with the object. The axis are pointing towards north, east and down. This frame can be used in local, and small areas, but are not valid for intercontinental travel. This frame will primarily be used in this report.

The last but important frame are the Body frame, which is the local frame of the object of interest. The body frame are defined as the x-axis along its forward movement, y-axis to the right of the movement direction, and the last pointing downward, to complete the right-hand system. The origin are defined in the Centre of Gravity of the object. This frame are convenient when defining velocities, forces and moments.

1.4 Hydrodynamic Model

An Autonomous Underwater Vehicle is a complex, nonlinear and coupled process. The model which is used in this report, uses the 6 DOF model described in [Fos02].

$$\dot{\eta} = \mathbf{J}(\eta)\nu \tag{1.1}$$
$$\mathbf{M}\dot{\nu} + \mathbf{C}(\nu)\nu + \mathbf{D}(\nu)\nu + \mathbf{G}(\eta) = \tau \tag{1.2}$$

where

$$\eta = [N\ E\ D\ \phi\ \theta\ \psi]^T$$
$$\nu = [u\ v\ w\ p\ q\ r]^T$$
$$\tau \in \mathbb{R}^6$$

The Equations (1.1) and (1.2) describes the kinematics and dynamics of the model. It is in the mathematical sense just the same as a Mass-damper-spring system. The Coriolis term, $\mathbf{C}(\nu)$, is a skew-symmetric matrix appearing because the dynamics are formulated in a non-inertial frame.

The $\mathbf{D}(\nu)\nu$ matrix are the damping forces, including drag from the surrounding water. $\mathbf{G}(\eta)$ are the restoring forces acting on the AUV, which are the gravity and buoyancy forces, and are represented by six-dimensional vector.

The $\mathbf{J}(\eta)$ matrix is the rotation matrix of Euler angles which relates the velocity of the vessel to actual movement in the NED reference system.

1.5 Camera Theory

The camera properties or parameters can be divided into two categories; *intrinsic* parameters and *extrinsic* parameters. The intrinsic parameters are constant parameters and vary from camera to camera, and represent the focus distance and image distortion of the pixels located away from the centre of the camera image. The extrinsic parameters relate the position of the point relative to camera coordinates. These parameters are of course dependant on the position of the camera and change with time. This imply that coordinates of a 2D point in the camera have to be transformed into a 3D point which can be used by the control system on board the AUV. Since we are going from less knowledge about a point to more knowledge about a point, some things are needed to be estimated or measured to gain the ability to solve the 2D to 3D problem exactly.[SHV06]

Suppose a point in the world reference system, denoted by, $P_w \in \mathbb{R}^3$. The same point represented in the camera frame, P_c are related to P_w by a rotation matrix, $\mathbf{R} \in SO(3)$. This gives the following equation:

$$P_w = \mathbf{R}P_c + O(t) \tag{1.3}$$

where $O(t)$ is the origin of the camera frame. This means that the point in the camera view is described by the equation

$$\begin{aligned} P_c &= \mathbf{R}^T(P_w - O(t)) \\ \mathbf{R}^T &= \mathbf{R}^{-1} \end{aligned} \tag{1.4}$$

A pinhole camera model are used to capture P_c to image coordinates, $P_i \in \mathbb{R}^2$. The principle behind a pinehole camera is that all lightbeams passes through a infinitesmall hole, or point, located in the origin of the camera frame.

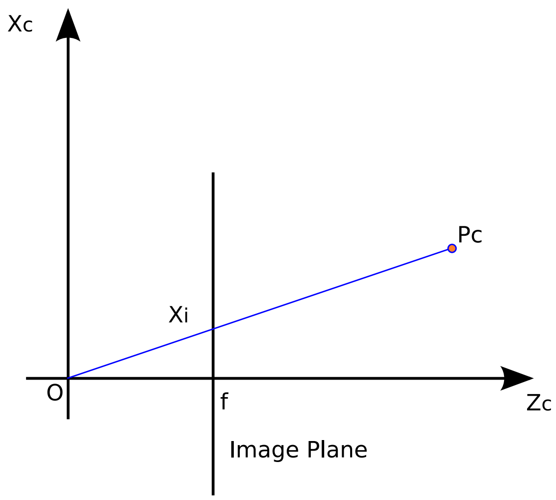


Figure 1.1: Pinhole Camera Model showing two dimensions

As seen from Figure 1.1 the image plane is located a distance f from the image plane. The point is projected through the hole, and onto the image plane. The principal axis, i.e. the z-axis of the camere is pointing in the direction of the observed point. For simplicity the image plane is located in the front of the pinhole (this is only possible in theoretical case, and would not be possible in a real camera application). From this the perspective equations are derived. [SHV06]

$$P_i = \begin{bmatrix} \frac{f}{z_c} & 0 \\ 0 & \frac{f}{z_c} \end{bmatrix} \begin{bmatrix} x_c \\ y_c \end{bmatrix} \tag{1.5}$$

The Equation (1.5) are the perspective equations, which transforms the observed point into image coordinates.

1.6 Kalman Filter

Kalman filtering are a powerful and versatile tool in estimation and sensor fusion. A Kalman filter is usually employed in navigation applications to fuse GPS and INS together. By this way one will have the speed and resolution of an INS system and the precision of the GPS system.

The Kalman Filter is an optimal filter, and can be employed in almost any application. It is optimal in the sense of providing the minimum-variance estimate of the predicted process. The Kalman filter is a linear filter, and can only guarantee optimality for linear systems. There is a nonlinear version of the Kalman filter, the Extended Kalman Filter. The EKF uses the same assumptions as the linear version, but uses the nonlinear model to predict the state forward, while it uses a linearised version of the measurement model when comparing the predicted values and the measurement. When updating the Covariance matrix the system equations are linearised around the current estimate and updated according to the same update laws as for the linear version. [BH97]

1.7 Guidance

Most of the guidance algorithms originates from airborne missile systems. These guidance laws have been well documented and proved to work in numerous cases. Common guidance schemes such as Line-of-sight (LOS) and Proportional Navigation Guidance (PNG) and various implementations of these are employed in most guidance systems today.

Guidance are defined according to [Shn98]

Definition 1. *The process of guiding the path of an object towards a given point, which in general may be moving.*

It is also convenient to define two levels of guidance.

Definition 2. *The process of making an object converge geometrically to a given point or path, is known as path following.*

Definition 3. *The process of making an object follow a geometric path with given dynamical constraints possibly dependant on position and time at the given path, is known as trajectory tracking.*

The guidance systems discussed in this report will concern the second definition. The second problem will be disregarded and assumed that the dynamical constraints are constant and met. [BF05]

The guidance system need to address the importance of a optimal paths. Optimal in the sense of energy consumption. A guidance system should be able to give feasible commands to the lower level control system which controls the actuators, and should be able to handle most situations it may encounter. The guidance system should decide the best trajectory to be followed based on the target location and physical capabilities of the system.[NSAB03]

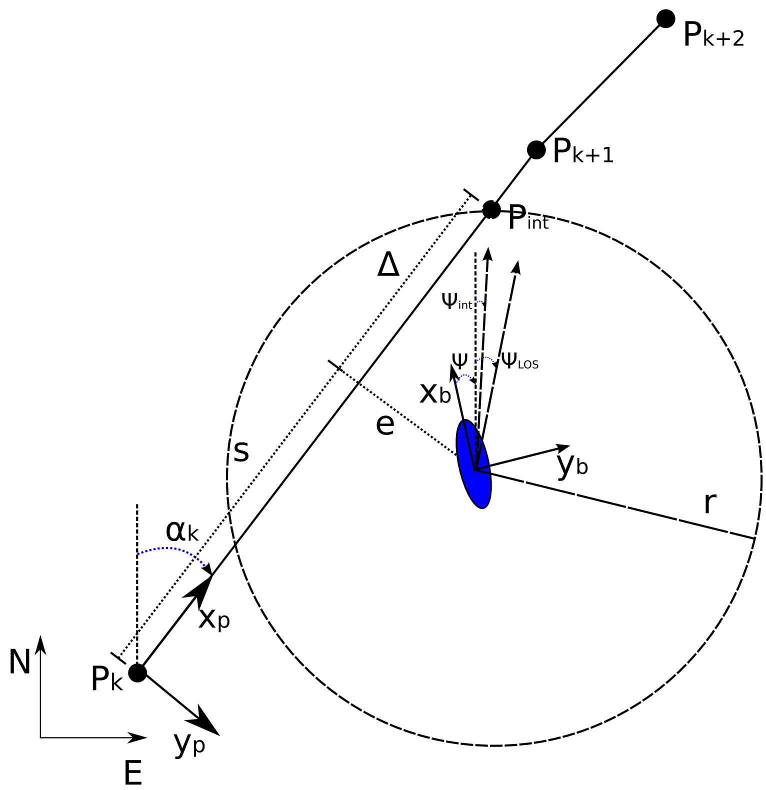


Figure 1.2: Variables associated with path following

The Figure 1.2 shows the variables which are important for linear path following. The cross-track error, e are an important aspect here. It is the lateral position error decomposed in the desired path reference frame. Another variable worth noting is Δ which is the look-ahead distance. This is analogous to when you drive a car you look farther down the road to better maneuver the car. A great look-ahead distance yields less aggressive heading reference, but slower convergence of the cross-track error. A lower look-ahead distance gives an aggressive heading reference and fast convergence of the cross-track error. The s-variable are called the along-track distance from point P_k.

The choice of linear path following is because the simplicity associated with the implementation, but this concept might easily be generalised to non-linear paths with the use of Serret-Frenet frame [EG02].

Line-of-Sight Guidance Law

Line-of-sight-guidance are the most common principle used. The LOS-algorithm computes the line-of-sight angle from the present location to the target location, and uses this angle as a reference heading.

The LOS-angle is fed directly into the heading controller as a reference. To make this law more tolerant to ocean currents and disturbances, a modified guidance law is presented. This uses the Side Slip angle, defined as:

$$\beta = \sin^{-1}\left(\frac{v}{\sqrt{u^2 + v^2 + w^2}}\right) \tag{1.6}$$

The Side Slip angle are the deviation of the velocity vector from the current heading of the vessel. The new heading reference is then taken as

$$\psi_d = \psi_{LOS} - \beta \tag{1.7}$$

Where ψ_{LOS} is the LOS-angle from the current position to the next waypoint.

Radius-based Guidance

Radius-based guidance uses the point where the radius around the current location intersects with the path, denoted on Figure 1.2 as P_{int}. This means assigning the desired heading as

$$\psi_d = \tan^{-1}\left(\frac{y_{int} - y}{x_{int} - y}\right) \tag{1.8}$$

where

$$(x_{int} - x)^2 + (y_{int} - y)^2 = r^2 \tag{1.9}$$

If r is chosen sufficiently large the equations above will have a solution, i.e $r > |e|$ otherwise there will exist no intersection point on the track line. [Bre08]

Lookahead-based Guidance

By using the direction of the line segment, path or pipeline, which finally are the desired heading that we want to achieve, ψ_d can be chosen as:

$$\psi_d = \alpha_k + \psi_r \tag{1.10}$$

where α_k is the direction of the path and

$$\psi_r = \tan^{-1}\left(\frac{-e}{\Delta}\right) \tag{1.11}$$

which can be seen as a correction term to make the desired heading converge towards the path.

By choosing ψ_r in this way the heading is always directed towards the lookahead point at the path. It is then easily seen that the cross-track error will converge to zero. This method is less computationally intensive than the previously stated Radius-based guidance. [Bre08]

Proportional Navigation

Proportional navigation guidance (PNG) are shown to give less interception time than LOS guidance, and thereby reducing the distance travelled. [NSAB03]

The PNG-law can be stated as the following:

$$\eta_c = N' V_c \dot{\lambda} \tag{1.12}$$

where η_c is the acceleration command, N' is the navigation ratio, V_c is the closing velocity, and $\dot{\lambda}$ is the line of sight angular velocity. The navigation ratio is a tuning parameter which will give higher demanded acceleration and thereby reaching the target in less time.

Proportional Navigation Guidance can be shown to be optimal in case of a non-maneuvering target. When presented with maneuvering targets the scheme does not perform that well, but lots of solutions have been proposed to make these laws more effective when dealing with moving targets. In this case the targets will be non-moving waypoints and the maneuvering laws will not be discussed here.

Various Guidance Concepts

A unified path following controller are derived in [BF06]. The authors proposes a singel control structure which will work for the entire non-zero velocity regime. The controller is derived using backstepping. The concept are shown to be Uniform Global Exponentially Stable under some assumptions about the guidance signals. The controller will not work without properly generated guidance references. The AUV considered in the article are fully actuated for low speeds but becomes under-actuated for higher speeds. The controller guarantees that the AUV converges towards the desired path regardless of if it is underactuated or fully actuated.

In [BPP06] an optimal cross-track guidance scheme is proposed. It seeks to minimise the crosstrack-error, depth, pitch and yaw by using the pitch rate and yaw rate, this is called Model Predictive Guidance, because it resembles a Model Predictive Controller. Damping and can be added to reduce the commanded pitch and yaw rates to counteract overshoot. The guidance scheme are compared against a LOS guidance system and gives good results.

Chapter 2

Modelling

This chapter summarises the problem and assumptions taken considering the pipeline following problem. A controller will be derived for tracking and maneuvering of the AUV vehicle. A Kalman filter will be derived for the smoothing of measurements and predictions of the pipeline. A guidance algorithm will be presented and discussed. The behaviour of the system will be treated in the last sections of this chapter, where a proposed flow of the decision process will be presented.

2.1 Problem Outline

To solve the pipeline following problem it is important to have a clear formulation of the problem. This is a path following problem, since the pipeline can be represented as a continuous path in space. The geometric convergence to the path is the primary objective, and any dynamical constraints along the path, i.e velocity constraints, are considered secondary to the goal of geometric convergence. When there are dynamic constraints the problem is a Trajectory Tracking Problem. If only geometric convergence are considered the problem is called a Path Following Problem, as stated in the definitions in Chapter 1.7

In this report the pipeline will be formulated as a two dimensional path. This because the pipeline are laid on the sea bottom, and are assumed to follow the bottom signature. Then the path following is reduced to a two dimensional problem. The *heave* state can be decoupled from the guidance system design. The depth controller then needs to keep a constant height above the sea bottom. This ignores the possibility of free spanning pipelines.

In order to limit the problem and ease the implementation, the pipeline are assumed to be a straight or nearly straight, piece-wise continuous line segment. The pipeline can only change direction at defined places called junctions. This reduces the application of this guidance system to only concider long, continuous stretches of piplines.

An AUV carrying out a pipeline inspection mission in most cases, renders the low speed assumption valid. The *HUGIN 1000* AUV are designed for speeds from 0-3 m/s. The inspection speed assumed in the rest of the report are around 1 m/s. This are relatively low speed, and the quadratic terms in the Coriolis/centripetal or Damping matices can be neglected.

As outlined in Figure 2.1 the building blocks of the pipeline following con-

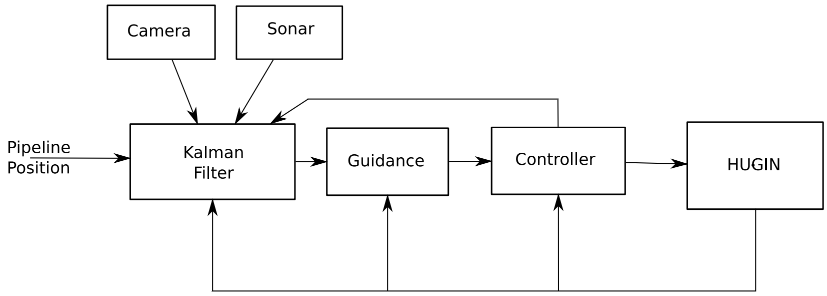


Figure 2.1: Block diagram the path following controller

trol system are the cruising controller, guidance system and the filter which fuses the known data about the pipeline and the measured pipeline data. To recreate the pipeline coordinates in three dimensions the distance to the sea bottom is measured by a sonar. The filter uses prior knowledge about the pipeline to-be-followed, which are assumed to be inaccurate. The filter uses this information together with the altitude information and camera output to estimate the position of the pipeline. If the pipeline should be lost at any time the filter will try to predict where the pipeline are, and still give input to the guidance block.

The system should be able to handle buried sections of the pipeline. The camera will not be able to sense the pipeline during the buried stretch and other sensors should be included. A bottom penetrating sonar might be used for this purpose, or a magnetic pipe tracker. These sensors are able to sense the pipeline even if buried up to 3 meters. [Bø08] These sensors will better the AUVs chances to regain track after the buried stretch, than if it were to relay just on dead reckoning from the predicted pipeline information alone.

2.2 Pipeline Representation

The representation of the pipeline is important because it is used in the Kalman filter to predict where the pipeline is going. The pipeline is parametrised by $\varpi \in \mathbb{R}$ to get a continuous and smooth pipeline.

In this report the pipeline are parametrised as:

$$P_w(\varpi) = \begin{bmatrix} \varpi \\ k_y \varpi \\ z(\eta) \end{bmatrix} \quad \varpi \in (0, \infty) \tag{2.1}$$

where k_y are constant and $z(\eta)$ is some function described by the sea bottom and are assumed known. This represents the straight line as the pipeline are assumed to be.

2.3 Kalman Filter

The Kalman filters purpose is to smooth the camera output and predict forward where the pipeline will be in the future, to supply a better heading reference for the guidance controller.

The position of the pipeline are given some uncertainty because the pipeline may have moved after it was laid, or the navigation system of the AUV might be erroneous and give wrong position. This gives the following model of the pipeline:

$$P_w(\varpi) = \begin{bmatrix} \varpi \\ k_y \varpi \\ z(\eta) \end{bmatrix} + \mathbf{B}\delta \tag{2.2}$$

The function δ are a slowly varying disturbance which can be modeled as a *1st order Markov Process*, and $\mathbf{B}$ is some 2x3 matrix.

$$\dot{\delta} = -\mathbf{T}\delta + w \tag{2.3}$$

where $w \in \mathbb{R}^2$, are a zero mean, unity variance, white noise process, which describes the error in the position. The matrix $\mathbf{T}$ specifies how the error evolves. Because the error are slowly varying the eigenvalues of $\mathbf{T}$ should be chosen large. Time differentiating the position estimate gives

$$\dot{P_w}(\varpi) = \begin{bmatrix} \dot{\varpi} \\ k_y \dot{\varpi} \\ \dot{z}(\eta) \end{bmatrix} + \dot{\delta} \tag{2.4}$$

By setting $\varpi = N(t)$, i.e the North position and completely disregarding the depth coordinate. gives the following

$$\dot{P'_w} = \begin{bmatrix} n \\ k_y n \end{bmatrix} + \dot{\delta} \tag{2.5}$$

where n is the rate that the AUV are moving towards north. Choosing the state vector and input as:

$$x = [P_w'^T \quad \delta^T]^T \quad \Rightarrow \quad \dot{x} = [\dot{P_w'^T} \quad \dot{\delta^T}]^T \quad u = n \tag{2.6}$$

This calls for the following state space representation

$$\dot{x} = \begin{bmatrix} 0 & -T \\ 0 & -T \end{bmatrix} x + \begin{bmatrix} 1 \\ k_y \\ 0 \\ 0 \end{bmatrix} u + \begin{bmatrix} \mathbf{0}_{2x2} \\ \mathbf{I}_{2x2} \end{bmatrix} w \tag{2.7}$$

This state space are non-minimal because of the linear dependent rows and could be eliminated, but it makes no difference with regard to the estimation.

The measurement model are more complex. In order to compare the measurement from the camera with the state space model, the perspective equations from Section 1.5 are needed. By using Equations (1.4) and (1.5), we can transform the point from world coordinates, i.e. coordinates decomposed in the NED-frame, to image coordinates. We want $y = P_i$

$$y = P_i = \mathbf{H}P_c = \mathbf{H}\mathbf{R}^T(P_w - O(t))$$

$O(t)$ is the origin of the camera frame and is equal to the position vector $\eta' = [N(t), E(t)]^T$ of the AUV in the NED frame. By including the position vector in the input to the filter and using $P_w = \mathbf{C}x$, the measurement model are concluded.

$$y = \mathbf{H}\mathbf{R}^T\mathbf{C}x - \mathbf{H}\mathbf{R}^T O(t) = \mathbf{C}'x + \mathbf{D}u \tag{2.8}$$

where $\mathbf{C}'$ and $\mathbf{D}$ are appropriate matrices for selecting the right states. $\mathbf{R}$ are the Rotation Matrix from World coordinates to Camera coordinates. This gives the following model for the Kalman filter

$$x = \mathbf{A}x + \mathbf{B}u + \mathbf{E}w \tag{2.9}$$

$$y = \mathbf{C}'x + \mathbf{D}u + v \tag{2.10}$$

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & -T_1 & 0 \\ 0 & 0 & 0 & -T_2 \\ 0 & 0 & -T_1 & 0 \\ 0 & 0 & 0 & -T_2 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 1 & 0 & 0 \\ k_y & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.11}$$

$$\mathbf{E} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{C}' = \begin{bmatrix} \frac{f}{z_c}\cos\psi & \frac{f}{z_c}\sin\psi & 0 & 0 \\ -\frac{f}{z_c}\sin\psi & \frac{f}{z_c}\cos\psi & 0 & 0 \end{bmatrix} \tag{2.12}$$

$$\mathbf{D} = \begin{bmatrix} 0 & -\frac{f}{z_c}\cos\psi & -\frac{f}{z_c}\sin\psi \\ 0 & \frac{f}{z_c}\sin\psi & -\frac{f}{z_c}\cos\psi \end{bmatrix} \tag{2.13}$$

w, v are a vectors of unity white noise and have covariance $E(ww^T) = \mathbf{Q}$ and $E(vv^T) = \mathbf{W}$. The filter should be run at the same frequency as the guidance system to provide output for the guidance system. The prediction will be updated whenever possible, which means when the pipeline are visible in the camera, after the image processing are finished.

Because of the relatively high frequency of the filter the parameters in the measurement model regarding heading and depth, i.e. ψ and z_c can be assumed constant during the sample period. This due to the slow dynamics of the AUV, compared to filter rate. This gives a linear model which can be guaranteed to be optimal for the current case.

2.4 Controller Design

The model discussed in Section 1.4 is used to derive the controller equations for the AUV control system.

The control system which supply the lower-level control system with forces and moments references, is divided into 3 sub systems:

- Speed control
- Depth control
- Heading control

This is called the flightmode controller, which is used for normal pipeline tracking, descent and ascent. This type of controller where chosen because it will be more energy efficient than other more actuated controllers. The second reason is even if the AUV is almost fully actuated, i.e controllable in 5 DOF, the tunnel thrusters which have to be present for this degree of actuation are almost useless for higher velocities. This renders just control in 3 DOF, *surge, pitch* and *yaw*. This will probably give the most energy efficient pipeline following, because the control are based on the cheapest control modes available at the AUV.

The *HUGIN*-type AUV is a slender-body AUV. This makes it possible to neglect some coupling effects between the states in the dynamic model. The *longitudinal* states (*surge, heave, pitch*) can be decoupled from the *lateral* states (*sway, roll, yaw*). This will be utilised in the next sections when deriving the control model and controller equations.

All coefficient next are defined in accordance with [SNA50].

Speed Controller

The speed controller is derived form the *surge*-subsystem, called *surge*-model [Fos02]. Under the slow speed assumption, the Coriolis/centripetal-matrix is assumed zero, $\mathbf{C}(\nu)\nu \approx 0$

$$(m - X_{\dot{u}})\dot{u} - X_u u - X_{|u|u}|u|u = \tau_1 \tag{2.14}$$

Setting $\tau_1 = -K_p\tilde{u} - K_i \int \tilde{u}dt$ the error in the velocity reference will go to zeros. The velocity reference is assumed constant and the PI controller guarantees that the error will go to zero.

Depth Controller

To derive the depth controller in the cruising control system the *longitudinal*-subsystem is used as the control model [Fos02]. By assuming that the lateral states i.e v, p, r, ϕ, are small, the kinematics can be derived as follows:

$$\begin{bmatrix} \dot{d} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} + \begin{bmatrix} -\sin\theta \\ 0 \end{bmatrix} u \tag{2.15}$$

The dynamics of the system then becomes

$$\begin{aligned} & \begin{bmatrix} m - X_{\dot{u}} & X_{\dot{w}} & mz_g - X_{\dot{q}} \\ X_{\dot{w}} & m - Z_{\dot{w}} & mx_g - Z_{\dot{q}} \\ mz_g - X_{\dot{q}} & mx_g - Z_{\dot{q}} & I_y - M_{\dot{q}} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \end{bmatrix} \\ + & \begin{bmatrix} -X_u & -X_w & -X_q \\ -Z_u & -Z_w & -Z_q \\ -M_u & -M_w & -M_q \end{bmatrix} \begin{bmatrix} u \\ w \\ q \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -(m - X_{\dot{u}})u \\ 0 & (Z_{\dot{w}} - X_{\dot{u}})u & mx_g u \end{bmatrix} \begin{bmatrix} u \\ w \\ q \end{bmatrix} \\ & + \begin{bmatrix} 0 \\ 0 \\ Wz_b \sin\theta \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_3 \\ \tau_5 \end{bmatrix} \end{aligned} \tag{2.16}$$

Since the *surge*-speed are stabilised with the controller derived in the previous section, the surge equation can be removed from the system under the assumption $u = u_0$. Also by assuming that the *heave*-velocity and θ are small the control model becomes:

$$\begin{bmatrix} \dot{d} \\ \dot{\theta} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & -u_0 & 0 \\ 0 & 0 & 1 \\ 0 & -\frac{1}{\gamma}W & -\frac{1}{\gamma}M_w(m - Z_{\dot{w}}) - M_w Z_q \end{bmatrix} \begin{bmatrix} d \\ \theta \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\gamma} \end{bmatrix} \tau_5 \tag{2.17}$$

where $\gamma = mI_y - mM_{\dot{q}} - I_y Z_{\dot{w}} + Z_{\dot{w}} M_{\dot{q}}$.

When closing the loop the following is proposed PID-like controller

$$\begin{aligned} \tau_5 &= -K_{dp}\tilde{d} + K_{dd}\theta + K_{dd2}q \\ \tilde{d} &= d - d_d \quad \dot{\tilde{d}} = -u_0\theta \quad \ddot{\tilde{d}} = -u_0 q \end{aligned} \tag{2.18}$$

A similar controller are implemented in [Jav94] where it it also shown to be asymptotically stable.

Heading Controller

Using the *lateral*-subsystem representation from [Fos02]. Under the assumptions that w, p, q, r, ϕ, and θ from the longitudinal subsystem are small, the kinematics are reduced to:

$$\dot{\phi} = p \tag{2.19}$$

$$\dot{\psi} = r \tag{2.20}$$

The low-speed assumption is utilised, higher order velocity terms are neglected, and constant *surge*-velocity $u = u_0$ are assumed. This gives the following system:

$$\begin{aligned} &\begin{bmatrix} m - Y_{\dot{v}} & -mz_g - Y_{\dot{p}} & mx_g - Y_{\dot{r}} \\ -mz_g - Y_{\dot{p}} & I_x - K_{\dot{p}} & I_{zx} - K_{\dot{r}} \\ mx_g - Y_{\dot{r}} & I_{zg} - K_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \end{bmatrix} \\ &+ \begin{bmatrix} -Y_v & -Y_p & -Y_r \\ -M_v & -M_p & -M_r \\ -N_v & -N_P & -N_r \end{bmatrix} \begin{bmatrix} v \\ p \\ r \end{bmatrix} + \begin{bmatrix} 0 & 0 & (m - X_{\dot{u}})u \\ 0 & 0 & 0 \\ (X_{\dot{u}} - Y_{\dot{v}})u & 0 & mx_g u \end{bmatrix} \begin{bmatrix} v \\ p \\ r \end{bmatrix} \\ &+ \begin{bmatrix} 0 \\ W z_b \sin\phi \\ 0 \end{bmatrix} = \begin{bmatrix} \tau_2 \\ \tau_4 \\ \tau_6 \end{bmatrix} \end{aligned} \tag{2.21}$$

The *roll*-state can be removed from the equations because of the assumptions of small $\dot{p}, p$ and because of the vessel are considered asymptotically stable in roll because of the offset in the buoyancy point. The *sway*-velocity can be neglected, the *sway* subsystem can be removed as well.

The following control model are used to derive the heading controller. The model is really the 1st order Nomoto model, and have become famous for it simplicity yet prove to give good results.

$$\begin{bmatrix} \dot{\psi} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{N_r - mx_g u_0}{I_z - N_{\dot{r}}} \end{bmatrix} \begin{bmatrix} \psi \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{I_z - N_{\dot{r}}} \end{bmatrix} \tau_6 \tag{2.22}$$

The following heading controller are proposed:

$$\tau_6 = T\dot{r}_d + r_d - K_{hp}\tilde{\psi} - K_{hi}\int \tilde{\psi} dt - K_{hd}\dot{\tilde{\psi}} \tag{2.23}$$

The terms $T\dot{r}_d + r_d$ are reference feed forward terms which will guarantee perfect tracking during course-changing manoeuvres according to [Fos02].

2.5 Summary of Assumptions

A summary of the assumptions are given here:

1. The pipeline is laid on the sea bottom, which gives the pipeline the same height signature as the sea bottom. The guidance problem is then reduced to a two-dimensional path following problem. No free-spanning pipelines are treated.

2. Pitch- and roll angles, are assumed small together with the corresponding pitch- and roll rates. The values are assumed to be in the vicinity of $\pm 10°$ and ± 0.05rad/s

3. *sway* and *heave* velocities are small compared to *surge* velocity and any cross-coupling terms may be neglected.

4. The full state are assumed perfectly known.

2.6 Guidance System

An autonomous system is by definition a system that will have minimal interaction from humans. It is supposed to do things on its own. This is the guidance systems task. Figure 2.2 shows a proposal of a guidance and decision system for the *HUGIN 1000* vehicle.

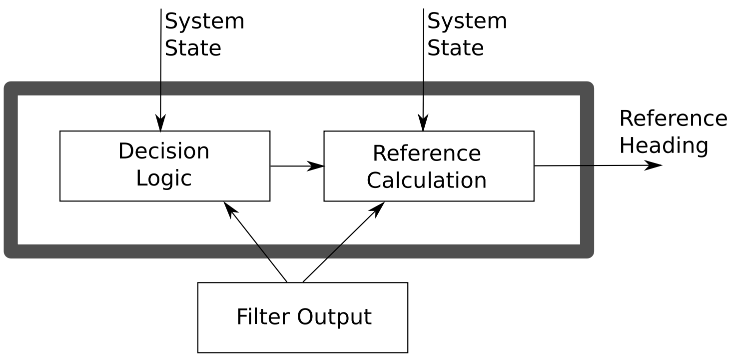


Figure 2.2: Guidance System Block

The AUV guidance system is assumed to have three modes, *searching*, *tracking* and *initialise/finalise*. The *initialise/finalise*-mode is at the beginning and end of a mission and will not be considered in this report.

The *searching*-mode is when the AUV are looking for the pipeline. When the pipeline is laid on the sea floor, the position may be more or less inexact, both because of sometimes the great distance from the sea level to the sea bottom, and sometimes because the pipeline have "sagged", i.e moved away from the initial position because of movement in the sea bottom caused by ocean current and other environmental forces.

Figure 2.3 shows the flow diagram of the searching mode. The system distinguishes between two cases; the cases when the pipeline have been located before, and when it is to be found for the first time. If the pipeline have not been

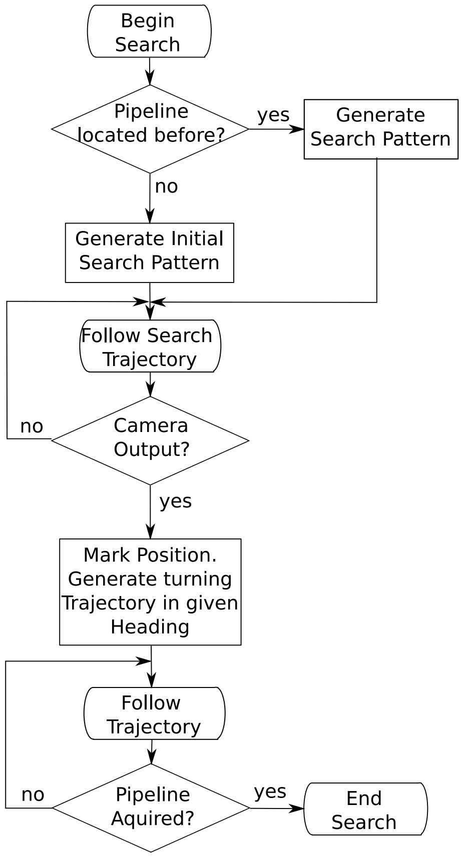


Figure 2.3: Flow diagram of the search procedure in the *searching*-mode

located before, the system generates a initial search pattern, which should be designed to cover larger area in all directions. If on the other hand the pipeline have been located before, another pattern should be generated, which searches the area in the direction the pipeline were last known to go. The system then follows this search pattern until it gets output from the camera, or other possible sensors. Then the position are marked and a turning trajectory are generated. When the pipeline is reacquired the guidance system goes to the tracking mode.

When in the *tracking*-mode the AUV tries to follow the pipeline as closely as possible, and keep the pipeline inside the field of view as best as possible.

Reference Calculation

A look-ahead based guidance algorithm is chosen for it's simplicity and robustness. All together a pipeline are made up of mostly linear segments or at least almost linear segment. One can then assume that the direction of the pipeline can be known exact. No sudden turns are assumed and pipeline junctions are assumed to be non-existing. This gives the following equations for the reference

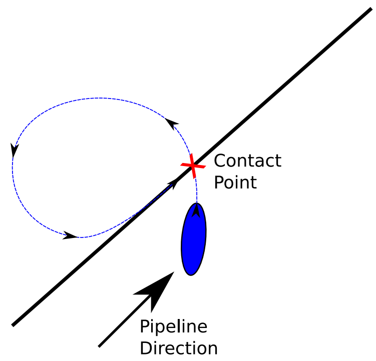


Figure 2.4: A turning trajectory for one possible contract

heading:

$$\psi_d = \alpha_k + \psi_r \tag{2.24}$$

$$\psi_r = \tan^{-1}\left(\frac{-e}{\Delta}\right) \tag{2.25}$$

$$e = -(n(t) - p_x)\sin\alpha_k + (e(t) - p_y)\cos\alpha_k \tag{2.26}$$

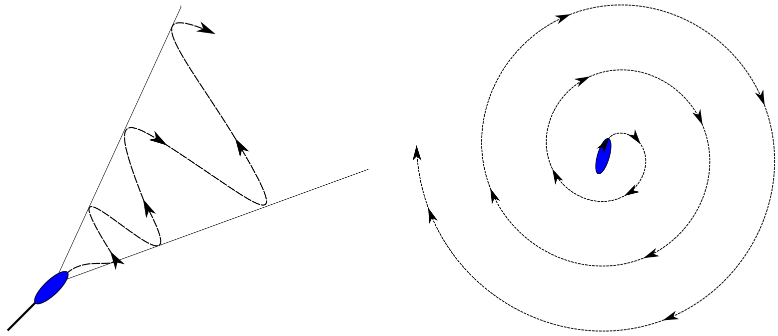
In the presence of ocean current it will cause the heading to "lean" towards the current, if the look-ahead distance is not chosen too big. The "leaning" are because the current will try to push the AUV off the pipeline and the guidance will compensate for this. At last some kind of equilibrium between the current and the AUV heading will be reached. This will will work well under the assumption that the current are constant.

Search Patterns

It is obvious that some kind of search pattern must be implemented for this kind of application. This might be customised for every mission or it might be selected from a library of suiting search patterns. This section will look at some search patterns which might be suiting for the pipeline search application.

The first case is when the pipeline are not in its predetermined position or the position sensors gives out erroneous readings, some kind of search need to be initiated to find the pipeline. This could be something like the depicted case in Figure 2.5(b). Spiral pattern going outwards to try to find the pipeline in the vicinity of the AUV position. This will make the AUV cover large a large area around where the pipeline are supposed to be. This should be aided with other sensors, for example with a Sidescan Sonar, which provides sensor data in the horizontal plane around the AUV.

The other case, is when the pipeline is lost during tracking. Since the general direction of the pipeline are assumed known, a divergent zig-zag pattern around the pipeline direction, from the last known direction, might be a suiting pattern. This will make the AUV cover the area where the pipeline are assumed to go.



(a) Divergent Zig-Zag pattern for lost pipeline during tracking

(b) Outwards spiral pattern for initial pipeline search

Figure 2.5: Differnet search pattern

2.7 Overall System

The system flow are depicted in Figure 2.6. This is how the overall system behaves during a pipeline inspection mission. The system will start by moving to

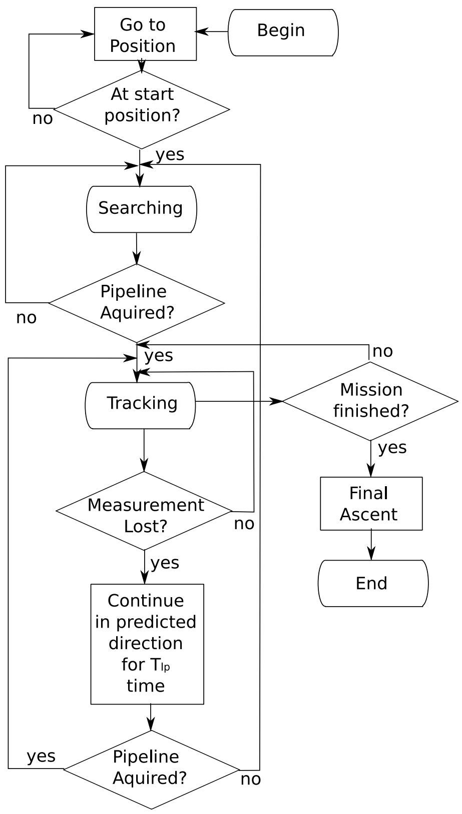


Figure 2.6: Flow diagram of pipeline inspection mission

the point where the inspection should start. Then the searching procedure will be initiated if the AUV does not make contact with the pipeline immediately. It will search for the pipeline until it has located it, and then continue tracking until the the goal are met or the on-board energy storage becomes depleted.

There are a number of design parameters needed to be set. These parameters might be constant or changing between missions. They are summarised in Table 2.1 These parameters should be chosen in accordance with what kind

Design Parameter	Description
T_{lp}	The Dead Reckoning time. Time before pipeline are considered lost.
Δ	The lookahead distance, deciding the convergence of the heading to the path
P_w	The predicted path of the pipeline in world coordinates or a parametrisation of it
$\mathbf{T}$	The values of this matrix must be chosen in accordance with the uncertainty of the predicted pipeline path. Usually chosen large
$\mathbf{Q}$ & $\mathbf{W}$	The covariance matrices of the filter, which specifies how much one wants to trust the predicted values of the pipeline and the uncertainty of the measured ones. Typically the measurements should be values more than the predicted values

Table 2.1: Summary of design parameters in the system

of environmental forces there are in the area. Also the search patterns should be customised for every mission. To set how much area the search patterns need to cover, and vary the spreading angle of the divergent zig-zag pattern in accordance with how certain the estimate of the direction of the pipeline is.

Chapter 3

Simulations and Results

To test the performance of the proposed guidance system, it is implemented in matlab/simulink. A number of scenarios are used to test how it performs in various conditions.

3.1 Matlab

The mathematical model of the *HUGIN 1000* AUV are implemented in simulink using the *GNC* toolbox available from `www.marinecontrol.org` with some modifications to the 6DOF model.

The Camera output simulator were programmed in matlab. It inputs the position of the AUV and transforms it to body coordinates to calculate the field of view of the camera. The camera simulator is based on the pinhole camera model with unity focus distance, and a view angle of about 45 degrees. The program then calculates the field of view of the camera, and checks if there are any part of the pipeline inside the field of view. The output of the camera are three points taken out at the top, the middle and the bottom of the field of view. See Algorithm 1

Algorithm 1: Camera Simulator($\eta(t)$, $z_b(t)$, f)

FieldOfView $= f(\eta(t), z_b(t), f)$
if Pipeline inside FieldOfView **then**
  Convert PipelineSegment inside FieldOfView to image coordinates
  Pick 3 Points at specified location from PipelineSegment
  return 3 Points
else
  return 0
end if

A sonar which determines the altitude are implemented using a look-up table with a predefined bottom profile.

The decision logic is implemented as a state machine with three states, and calculates the desired heading dependent on what state the system is in. This implementation is in correspondence with Figure 2.6.

The filter was created using a m-file and global variables for the filter parameters. The filter parameters are as follows:

$$P_0 = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \quad W = 0.1\mathbf{I}_{2x2} \quad Q = 10\mathbf{I}_{2x2}$$

The $\mathbf{T}$ matrix are chosen to be diagonal with 1000 at both elements.

The final simulink diagram are shown in figure 3.1. The structure of the

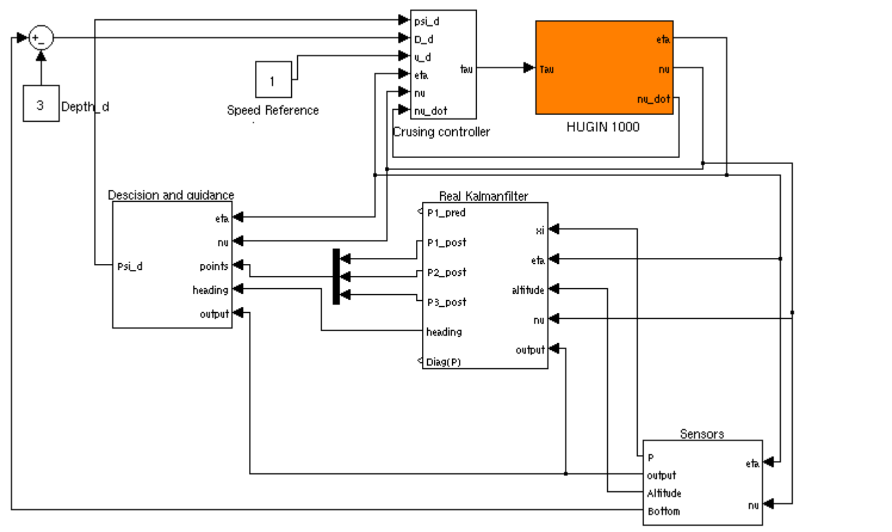


Figure 3.1: The Simulink Diagram of the implemented Guidance System

model are strictly modular and the blocks shown in the simulink diagram have the same function as in Figure 2.1. This allows for updates and improved blocks to be implemented later, without designing a new system.

The matlab-scripts are showen in the Appendix.

3.2 Simulation Scenarios

To test the performance of the guidance system some scenarios are proposed. In all the scenarios, the pipeline are located at the sea bottom, around 10 meters bellow the surface. The depth is arbitrarily chosen and the value does change the scenarios except that the submerging time is longer. The AUV will start at the surface and submerge towards the starting point of the mission.

1st Scenario. The pipeline are at the exact location according to predefined data. Environmental disturbances such as currents are turned off. The pipelien are continuously visible for the camera the whole inspection distance. Reference simulation.

2nd Scenario. Exact as over but with environmental forces turned on.

3rd Scenario. The pipeline are at the exact location where it initially was laid. A section of the pipeline is buried and not visible for the camera. Environmental forces are turned on.

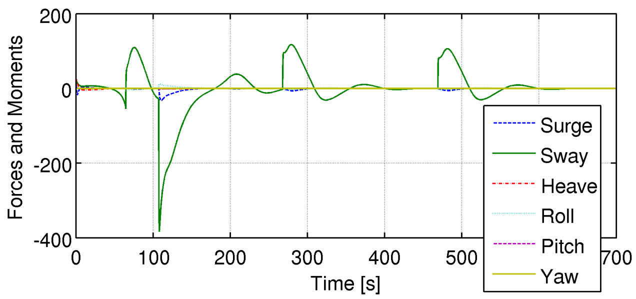
4th Scenario. The á priori information about the pipeline are offset about 50 meteres to test the ability of the guidance system to search for the pipeline. Environmental forces are turned on.

3.3 Results

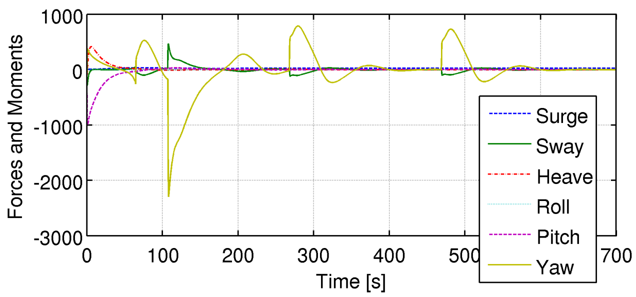
The matlab/simulink implementation were simulated with the given setup. A simulation were done to check if the low speed assumptions were valid. After this the simulation of the scenarios were done and produced the following results.

Test of the Low-speed Assumption

In figures 3.2(a) and 3.2(b) the forces and moments created by the Coriolis/centripetal and damping matrices are recorded. In Figure 3.2(a) the sway degree of freedom are dominant, and peaking about -400N during the turning manoeuvres of the AUV. The forces and moments created by the Coriolis terms are partially counteracted by the damping terms, that also have greater magnitude than the Coriolis terms. This suggests that the Coriolis/centripetal forces can



(a) Coriolis Forces



(b) Damping Forces

Figure 3.2: The Forces associated with the AUV maneuvering

be neglected and compensated for in the controller using integral terms.

1st Scenario

This is a reference test to see how the guidance system performs on the ideal case. This is to show the sensitivity to the environmental disturbances, introduced in later scenarios.

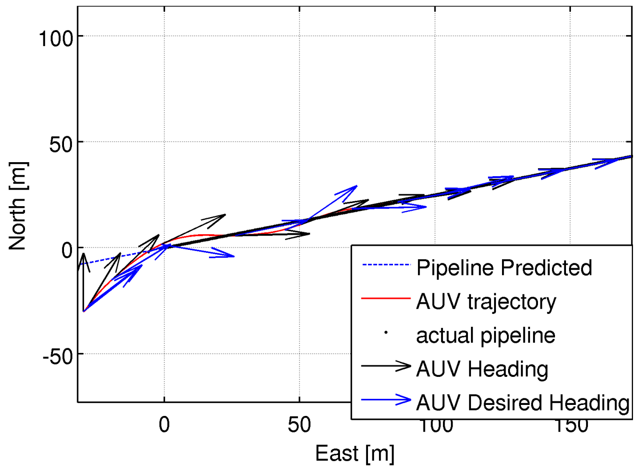


Figure 3.3: North East path of AUV without Current

It can be seen from both Figure 3.3 and the third plot on Figure 3.4 that the heading reference, ψ_d have some oscillatory nature. This is because of the relatively low look-ahead distance defined in the guidance algorithm. This can be analogous to when driving a car and you fix your gaze on the road not very far ahead of the car, you will get more uneasy driving and jerking motion.

The depth reference is given by the bottom, created using a look-up table of a sinusoidal plane. The reference are followed pretty well. The delay on the action by the controller are created because the *heave* direction are not directly controlled, but are relayed through the *pitch* degree of freedom. This could probably have been reduced by feed-forwarding the reference into the controller as well. This will help the controller predict the motion and compensate for it when it happens.

It is worth noting that the *surge* speed overshoots, but is not seen as a problem for further simulation and analysis. The overshoot can be removed by including derivative action in the Speed controller. This would reduce the commanded force when reaching the set point.

2nd Scenario

The environmental forces are now turned. The current is assumed only effective in the North East plane and has no effect in the *heave*-direction. The current is moving from north-west to south-east, heading $-45°$ and have a strength of 0.3 m/s.

In Figure 3.5(a) the search waypoints are shown. The reason for the extra "de-tour" away from the pipeline are because of the short time delay before considering the pipeline lost. As seen in the first Scenario there are oscillations in the heading reference. This causes the AUV to drift off the pipeline trajec-

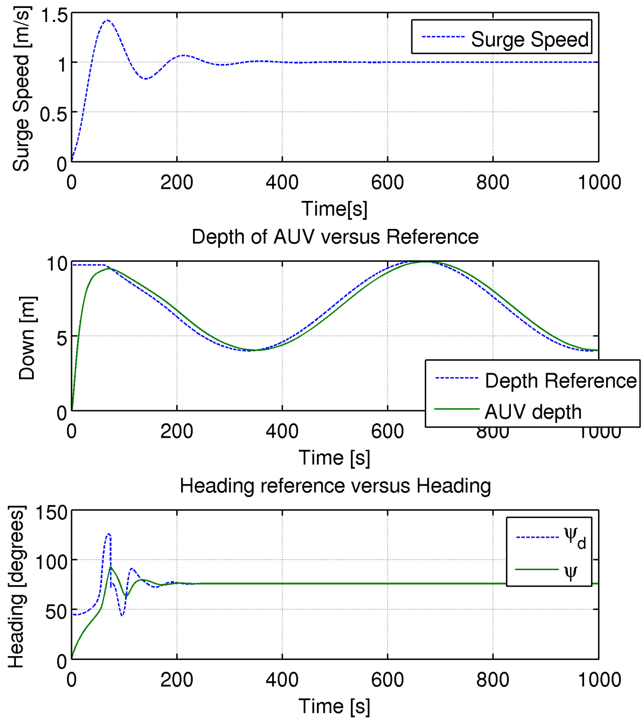
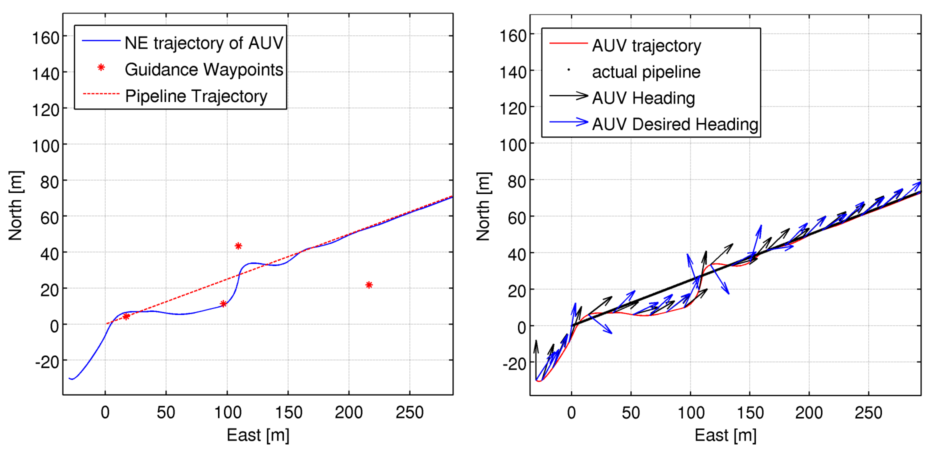


Figure 3.4: Surge-, Depth- and Heading- Reference vs. Actual Values



(a) NE Path with Waypoints (b) NE path with Heading

Figure 3.5: Plots of the AUV Showing Trajectory, Guidance Waypoints and Heading of AUV second scenario

tory and therefor lose visual contact with it. The system considers the pipeline as lost and generates a search pattern, the "divergent zig-zag" spoken of in Chapter 2.6. The time before going into search mode are set to 25 seconds.

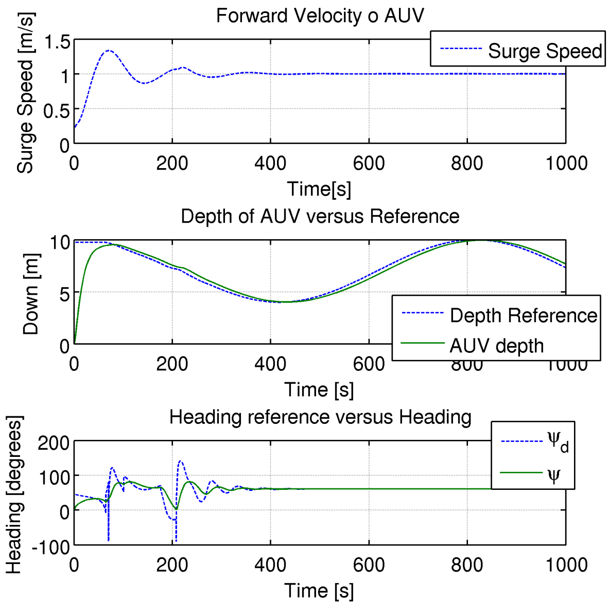


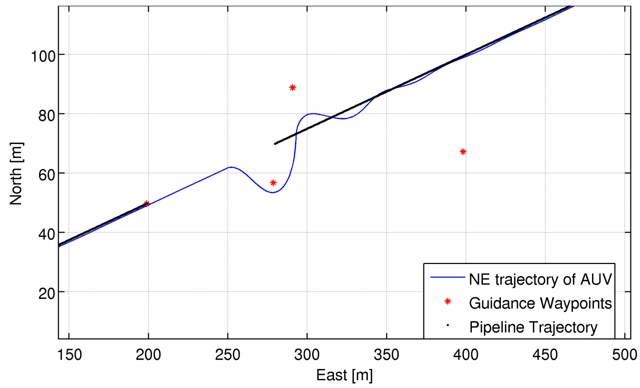
Figure 3.6: Surge-, Depth- and Heading- Reference with Current influence second scenario

When current are introduced the AUV heading are "leaned" toward the current. In Figure 3.5(b) the current are coming from north-west, corresponding to the upper right corner in the figure. This current will try to push the AUV off the pipeline. The guidance system answers this disturbance by adjusting the heading reference towards the current. This is one of the reasons why the lookahead distance are chosen small, because of this the AUV will not drift far away from the pipeline. Since the current is constant, a equilibrium is achieved between the desired heading and the current. The heading converges towards around 60° instead of the pipeline direction of 75°.

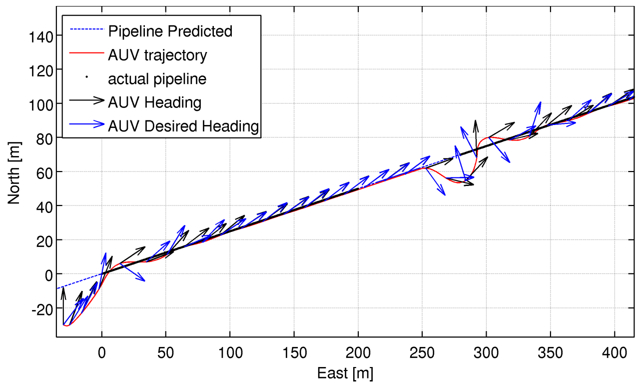
The AUV are not directly on top of the pipeline anymore, but it is not desirable to be exactly over the pipeline all the time. To strictly control the AUV to lay exactly over the pipeline would use much of the limited power supply. Also, the objective of the AUV are not to stay exactly over the pipeline, but to provide good pictures and sensor data for later inspection by humans. It might be much easier to analyse a stable picture than a tightly regulated motion which might cause a noisy picture.

3rd Scenario

The setup for this scenario, is a simulated burrial of the pipeline at approximately 50 meters North and 200 meters East. At this point the camera looses track of the pipeline. The guidance system will command the AUV to continue following the predicted pipeline until some time limit are reached. This time limit are set to 60 seconds. From Figure 3.7 it can be seen that the AUV follows the predicted pipeline for about 50 meters and then engages in the predetermined search pattern.



(a) NE Path with Waypoints



(b) NE path with Heading

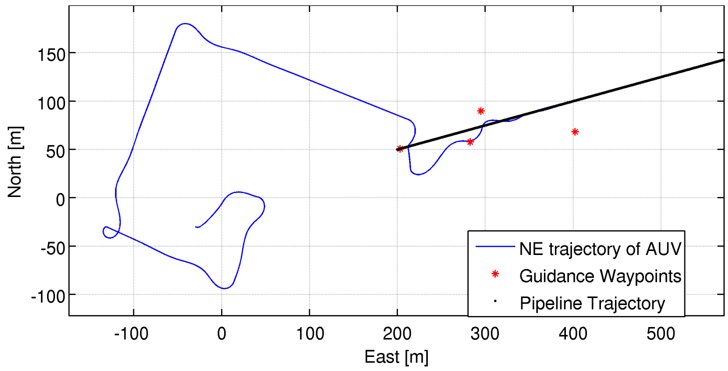
Figure 3.7: Plots of the AUV Showing Trajectory, Guidance Waypoints and Heading of AUV for the 3$^{\mathrm{rd}}$ Scenario

The meaning of following the predicted pipeline is of course to get back on track at the end of the buried stretch. Sometimes pipelines are buried on purpose to seal them from environmental erosion, and sometimes this happens unintended. In either case, the need of more sensors which can penetrate the sea bottom and locate the pipeline are needed.

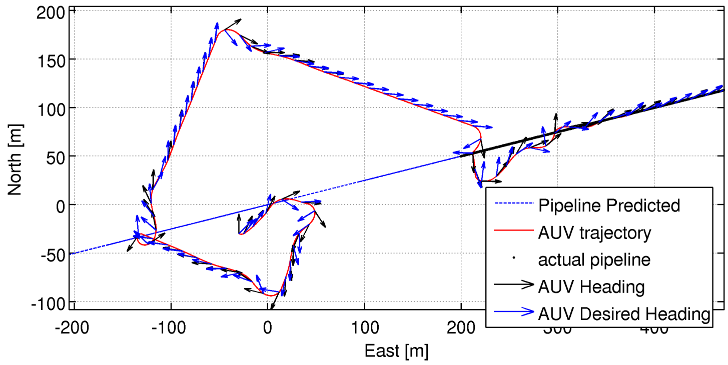
There are no turning trajectory implemented at this point, and will not be used in the next scenarios either.

4$^{\mathrm{th}}$ Scenario

The fourth Scenario is to demonstrate the capabilities for the AUV to acquire the pipeline, when the pipeline is offset from where it was thought to be.



(a) NE Path with Waypoints



(b) NE path with Heading

Figure 3.8: Plots of the AUV Showing Trajectory, Guidance Waypoints and Heading of AUV for the 4$^{\text{th}}$ Scenario

When the AUV reaches the initial position where the pipeline were meant to be. It does not make contact with it there, then engages in a spiral search pattern. This search pattern are administered by waypoints and the guidance uses the straight lines between the waypoints as the track to follow. This causes the more polygonal look than spiral.

From Figure 3.8 it can be seen that the AUV sometimes takes an extra turn when switching between waypoint four and five. This is due to the calculation of the desired heading angle. The *matlab*-function, *atan2()* are used which outputs the angle in the domain $(-\pi, \pi)$. Since the output from the AUV model are defined for all values, and this are fed back to the controller a heading reference of 0 means that it must "unwind" all turns it has done, because this accumulate the yaw value beyond 2π. This is an implementation issue and is not caused by the theoretical guidance system.

The plot also shows where the filter predicts the pipeline to be. This is because the filter predicts the pipeline dependent on the position of the AUV. But as long as the sensors do not get a reading, the AUV continues on the search trajectory.

The large overshoot when the AUV have located the pipeline are due to the current. This pushes the AUV away from the trajectory. The current increases the velocity in a direction and causes the turning in that direction to be more difficult than turning the other way. This is also a product of the relatively large course-changing maneuver. This can somewhat be reduces by using a turn trajectory to get the AUV on the right track and the same direction as the pipeline.

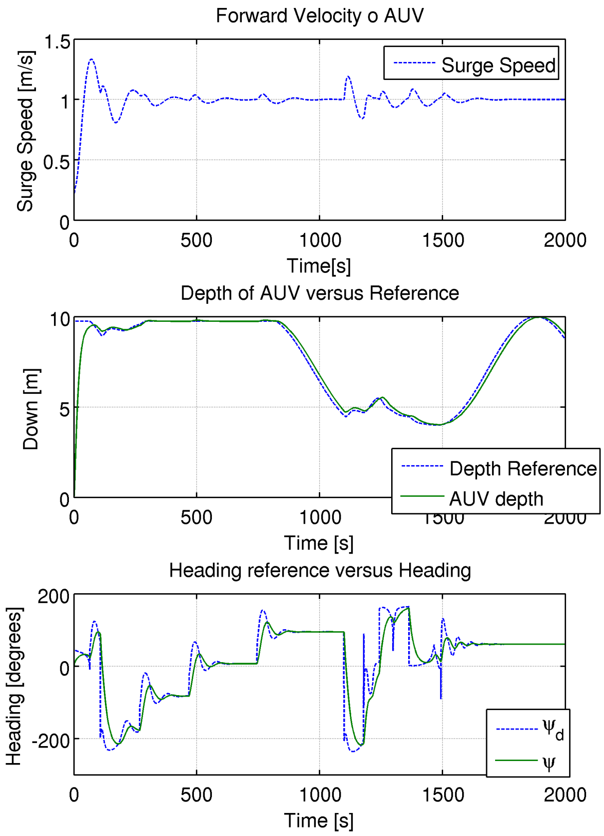


Figure 3.9: Surge-, Depth- and Heading- Reference with Current influence for the fourth scenario

The plots from the velocity, depth and heading references for the fourth scenario, shown in Figure 3.9, are worth noting. It can be seen from the first plot in the figure that during large course-changing manoeuvres the surge velocity first gets a overshoot and then decreases under the set point of 1 m/s. This is probably because of the how the current are included in the simulations. We see that the course changes from around 90° to −200° over short time. The surge speed changes correspondingly in the same area. This is because when the course angle are about 45°, the velocity vector are parallel to the current, and at −45° the velocity vector are still parallel but in the opposite direction of the current. This causes the marked changes in the surge speed seen in Figure 3.9. The speed controller cannot anticipate this, and therefore does not counteract it.

Test of the Prediction Filter

The filter data are recorded at the fourth simulation. Figure 3.10 shows the error values in the predicted pipeline and the updated in the first two plots. The last one shows the values of the diagonal of the Kalman covariance matrix.

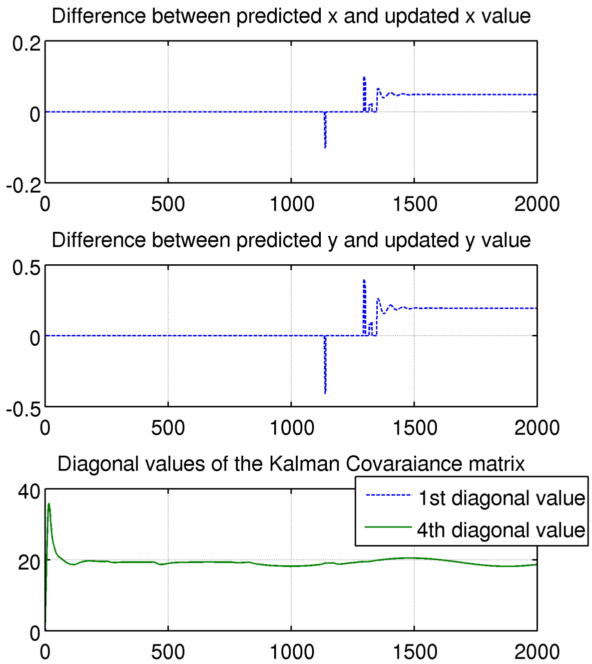


Figure 3.10: The error values of the predicted versus updated x and y components of the pipeline are shown in the first two plots. The last plot shows the diagonal values of the covariance matrix.

The peaks in the error values corresponds to when the AUV makes contact with the pipeline. The filter updates the predicted values only when there are a valid output form the sensors. Also seen in the figure are constant offset when the AUV has acquired the pipeline. This is because the predicted values are calculated from the current position of the AUV, and the updated estimate from the camera gives the correct position of the pipeline. Due the fact that the AUV are not directly over the pipeline, this information differs and causes the constant error between the predicted and updated estimate. Because the model used in the filter have two linear dependent states, two and two values of the covariance matrix are equal, i.e. the third state equals fourth and first equals second.

Although this filter works, it is not a very good way of doing it. The predicted position of the pipeline are calculated by using the position of the AUV. A better way of this is to use a look-up table with the predicted pipeline position. This values could be looked up when a measurement from the camera or other sensors were available and corrected.

Chapter 4

Discussion

This chapter will summarise and discuss the results given in the previous chapter.

4.1 Guidance System

This guidance system are designed for long, straight pipeline stretches. It will not be able to handle sudden turns in the pipeline direction, without specifying this in the guidance system. This can be included as waypoints where the pipeline are changing, and be included as a condition; *when the AUV reaches certain position the direction of the pipeline changes*. This requires very exact knowledge about the pipeline and are really the case. This is why there should be a more autonomous way of doing this. This motivates the use of more sensors than just a camera to follow the pipeline. The camera have a very limited field of view, usually restricted to less than 3 meters. A Sidescan Sonar combined with a Forward Looking Sonar, which provides sensor data of the pipeline in front of the AUV will give the guidance system some data to decide and predict what it will do if there is a sharp turn in the pipeline.

Sharp turns are a product of T-junctions and other couplings of pipelines. These junctions are usually well documented, and given good and exact locations. But if the navigation sensors of the AUV have large uncertainties, it might look as they are in the wrong place, for the AUV. This suggests that information about the pipeline should be treated with care. Because the errors in an AUV navigation system might be substantial and provide that á priori information about the pipeline will be unusable.

The navigation system of *HUGIN 1000* are a Velocity Aided Inertial Navigation System. This utilises a Doppler Velocity Log to measure the velocity relatively to the sea bottom and input this to the INS system. The INS systems installed on the *HUGIN 1000* are in the 1 nmi/h class, i.e the INS system drifts less than 1 nmi in an hour. This results in a drift in the navigation system equal to 0.11 % of the travelled distance along the track, and about 0.03 % error in the across distance of a straight line track according to [JGHV03]. This can be enough to throw the guidance system off course, because the field of view of the camera are relatively small.

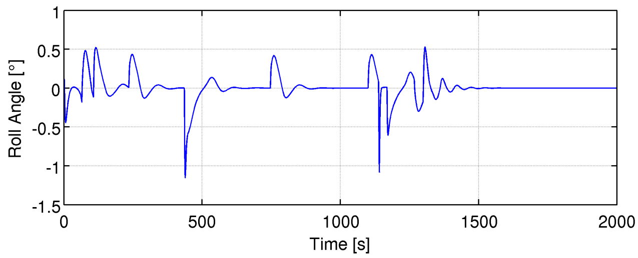
There are ways of improving the INS drift and thereby improving the position estimate, one is to use GPS update fixes, but this requires to surface the

AUV once in a while. This is of course not a good idea when the AUV are at great depths. There are possibilities to use sea bottom anchored position buoys, which exact position are known and the AUV might use these buoys by pinging them and getting a updated position estimate. This is a good idea if the pipeline infrastructure admits this. Say that this position buoys are placed at the same time as the pipeline are laid.

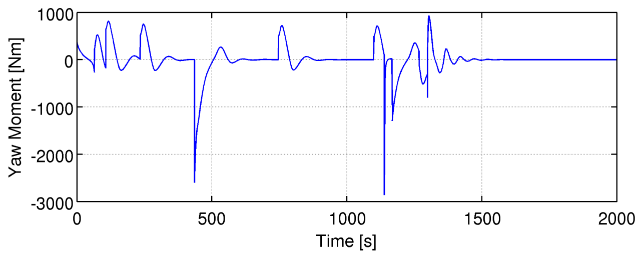
The problem regarding when $\psi \to 2\pi$, as mentioned in the previous chapter is a implementation issue. There are a number of solutions for this. The first is to limit the sensor output, which is the case in the real world, since a compass measuring *yaw* only are defined for $(0, 2\pi)$. The controller can handle this by including a check whether if its heading are larger than π, the given command will be to the right, and opposite if the measured heading are smaller than π.

4.2 Roll Stabilisation

The mission of the AUV are to provide good data for later use, i.e good pictures to be analysed later. The camera on the AUV are mounted downwards and the field of view are affected by roll and pitch. Pitch are a control angle, but the Roll motion have no direct control measures. The roll angle need to be as close to 0 as possible to provide best pictures of the current sea bottom. The plots in Figure 4.1 are taken from the 4^{th} Scenario described in Chapter 3, to look at *HUGIN 1000* 's stability in roll.



(a) Roll Angle



(b) Yaw Moment

Figure 4.1: Plots describing the close relation of yaw moment and roll angle

Figure 4.1 shows the relation between commanded yaw moment and the roll angle. Clearly there are coupling effects which causes the roll angle to change a few degrees. The roll angle magnitude are about 1° when the *surge*-velocity are 1 m/s. The values of the *roll* angle are doubled when the *surge*-velocity are doubled. But the roll angle are not of great concern when carrying out this kind of inspection missions, as these simulations show.

The main propeller gives a moment in roll. This moment are disregarded in the simulations, but might become important. The main propeller are described by the following relations

$$\begin{aligned} \tau_1 &= T_{nn}n^2 + T_{un}nu \\ \tau_4 &= Q_{nn}n^2 + Q_{un}nu \end{aligned} \tag{4.1}$$

where n are the angular speed of the main propeller in revolutions per minute, and u are again the surge velocity. The second terms in the equation above are terms describing loss-of-force due to forward speed. The moment in roll are actually countered by weighting the AUV down on one side so that when the it reaches cruise velocity the AUV have zero roll angle. [Gje08]

The *HUGIN 1000* AUV are designed to be asymptotically stable in roll, due to the centre of buoyancy (CB) are located not exactly in the centre of gravity (CG). The AUV would not need roll stabilisation because it in fact shows to be very stable in roll.

4.3 Energy Consumption

The energy consumption of an AUV are of extreme importance. When a customer are buying this type of vessel, one of the criteria are operation time. For an AUV which are untethered and have limited power supply the operation time might vary greatly due to installed sensors and operating conditions.

The standard sensor suite on the *HUGIN 1000* AUV are a Side Scan Sonar, Doppler Velocity Log, Depth meter, Multibeam Echosounder, and the INS navigation system. The operation time would be around 10-20 hours, dependent on velocity and sensor use. The objective of the guidance system after the inspection part, is to maximise the operation time. This can be done in numerous ways, and should be a part of the design procedure when designing the guidance system for the real thing.

The actuator set-up of *HUGIN 1000* are 4 rudders, one main propeller and 4 thrusters. This renders the AUV controllable in 5 DOFs. The thrusters demand more power than the main propeller and rudders. Since this report considered the most energy efficient control, the thrusters have been disregarded throughout this report, but they will be of use if more advanced features should be implemented in the guidance system, such as docking of the AUV to an underwater charging station.

The most energy efficient way of guiding an AUV is to use its main propeller and rudders. But there might be different ways of optimising the power consumption with regard to how the AUV are searching for, and tracking the pipeline. The moment produced by, for instance the vertical rudders, are proportional to the *surge*-velocity squared.

$$\tau_6 = Y_{u\psi}\delta u^2 \tag{4.2}$$

where δ are the angle of the rudder, and $Y_{u\psi}$ are some constant describing the rudder areal. This means that the effect of the rudders are greatly dependent on the forward speed. This is not taken into account in the designed controller and should be compensated for in a more advanced controller. This produces problems when the AUV are moving at greater velocities. Since the controller only outputs moments which are not dependent on velocity, the AUV will have less maneuverability at higher speeds. This causes the proposed guidance scheme to fail at surge speeds higher than 1 m/s if the controller are not retuned.

The energy consumption are very dependent on how the AUV moves. If it is taking unnecessary turns the movement pattern are not optimal. If the forward speed are taken constant, the power consumption will decrease along with the AUV time-of-flight. In this sense the shortest distance travelled is the most optimal movement pattern. This motivates the designer to make the guidance system to always calculate the shortest path to the goal, if the environment allows it. This opens for a more advanced mode of the tracking system, where the constraint that the AUV should strictly follow the pipeline, are relaxed. If sensors like Sidescan Sonar or other sensors with longer range than a video camera are included, these can be used to give good data about the pipeline even if it is not in the camera field of view. The relaxation in the tracking constraints might allow the AUV to move more efficient in terms of energy consumption.

[KU03] proposes an optimal guidance scheme. Using the fuel consumption as a performance index and optimise with regard to the ocean currents in the area of interest. It requires knowledge of the currents in the area á priori, and utilises this to calculate the optimal trajectory from one point to the next. This gives good theoretical results, but the downside is that knowledge about the current forces in an area are rarely known, and if they are known they are probably not very exact. But this scheme might help to save energy, but are only applicable to a pipeline inspection mission when the AUV are moving to, or from the pipeline.

Either way, it is important to chose the right way to go when you have limited power capability. This motivates the use of more sensors, to increase the chances of not giving bad and erroneous sensor data to the guidance system. The guidance system needs to be robust and take the right decision. To maximise the operation time, it can not afford to take wrong decisions. This is almost impossible to achieve and can only be done through extensive testing of the complete system.

The velocity of the AUV are of great importance. Since the AUV will experience drag effects by the water, and those effects are proportional to the velocity, the operation time will be greatly dependant on the speed. The equation below describes the relation between speed and range [Sha07]

$$R = \frac{E}{K_D} u^{-2} \tag{4.3}$$

where R is the range the AUV can travel in meters, E is the available energy in Joules, K_D is the efficient drag coefficient in W m^3/s^3 and u is the velocity in m/s. This shows that the guidance system should control the speed in preference with what is most important. This relation are good for torpedo shaped AUVs such as *HUGIN 1000* , but it only apply when the AUV are moving at cruise speed, and not when it are using its thrusters.

4.4 Optimal Search Pattern

As discussed above, the need for efficient search pattern are crucial to maximising the operation time. Search patterns might vary form case to case. The search patten should be dependent on how well the mission area are known and how sure one can be about the á priori data about the pipeline, sea bottom obstacles and other features. Are there obstacles in the area which might be used for position identification and navigation? Or are there obstacles in the area that the AUV should avoid completely, such as mines? This is information which are crucial when customising the search patterns.

In general the more you know about the mission area, the better can you prepare the AUV for the mission. If there are known features on the sea bottom, those can be used as reference marks and navigation of the AUV. This allows for more customised search pattern which suit the mission area.

4.5 Validation of the Results

The analysis done in this report are meant to be a pre-study of the problems associated with this complex pipeline inspection problem. Most of the topics covered in this report need more study to get valid results. But it shows some key features for a future design.

The distance covered by piplines are large, but most of the pipelines laid today are flexible pipelines and can curve quite a lot, and the assumption about linear pipeline segments are a bit to simplifying. On the other hand it is not difficult to generalise the guidance system to consider nonlinear paths.

The visibility is another topic about this simulations. The visibility are quickly degrading as depth increases, and to get good pictures there must have a good source of light. The visibility at 300 meters are probably less than 3 meters. This will greatly decrease the field of view of the camera.

5 Conclusion

In this report, the topic of designing a guidance system capable of pipeline inspection for the *HUGIN 1000* AUV has been treated. This report have looked at some basics regarding pipeline inspection. A literature study on how pipeline inspection can be preformed was done, and summarised in the report. Some basics about guidance was reviewed.

A proposed guidance system was also designed. The system includes a flight-mode controller, a guidance algorithm and a Kalman filter to fuse measurements and prior knowledge about the pipeline together. The filter will also smooth the output from the sensors to provide a continuous reference for the guidance algorithm.

The controller designed consists of three independent PI-, and PID-controllers, which were tuned according to stability and simulations. They are tuned with regard to velocities around 1 m/s. Especially the heading controller needs to be re-tuned to make it compatible with velocities higher than this.

The behaviour of the guidance system was defined and examined. Some search patterns were discussed, and found that they should be customised with regard to how well the mission area are known and how certain the data about the pipeline are.

The system was implemented in Matlab/simulink and 4 independent simulations were run, demonstrating different abilities with the given guidance system. The system gave results and the given setup worked, also in the presence of ocean current, but are not as robust as it should be. The simulations were discussed with regard to energy efficiency and pipeline following capabilities. The most optimal path concerning energy optimality when the velocity are constant, is the shortest path, in most cases the linear segments between two points.

Whenever the AUV is tracking, the motion above the pipeline should be constrained to be in the vicinity of the pipeline, but not strictly above it all the time. The primary mission of the inspection should be to provide good, easy-to-analyse sensor data, which in most cases include trying to maneuver the AUV as smoothly as possible.

The camera should be aided with other sensors, such as Side Scan Sonar and Multibeam Echosounders. Theses sensors help with detection of the pipeline and will also provide data about the condition of the pipeline.

The strict modularity of the guidance system presented in this report allows for easy upgrades and improvement to the different blocks in the system. In further work with this problem the guidance system will be easy to upgrade, with more advanced controllers or guidance algorithms. The simulation environment present, gives good results.

5.1 Further Work

There are much work do be done before this can be transformed into real life application.

First the guidance system should handle three dimensional guidance with nonlinear paths. A possible way to do this is to use unified guidance controller proposed in [BF06]. This is a controller for the whole non-zero speed regime.

The Kalman filter should be designed to include readings from other sensors like Sidescan Sonar, Echosounders, and other possible sensors available for the inspection mission, such as a Synthetic Aperture Sonar. The filter should be expanded to be capable of three dimensional prediction and a more sophisticated prediction might be used to get good results for the predictions.

It should be room for making the guidance system control the inspection velocity. If there are large stretches of pipeline which requires little maneuvering, the speed can be increased or if there the sensors show parts of the pipeline which shows signs of corrosion, the inspection velocity can be slowed to give more data about the area.

Develop some kind of a *power* mode and *economy* mode for the guidance system, where the *power* mode can be used when there are strong currents in the area, to help the AUV continue the tracking procedure, or to quickly get the AUV to the desired location. The *economy* mode can be utilised when tracking the pipeline and environmental forces are not very dominating.

References

[BF05] Morten Breivik and Thor I. Fossen. Principles of guidance-based path following in 2d and 3d. *44th IEEE Conference on Decision and Control*, 2005.

[BF06] Morten Breivik and Thor I. Fossen. A unified control concept for autonomous underwater vehicles. *Proceedings of the 2006 American Control Conference*, 2006.

[BH97] Robert Grover Brown and Patrick Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering.* John Wiley and Sons, 3rd edition, 1997.

[BPP06] Even Børhaug, Kristin Pettersen, and Alexey Pavlov. An optimal guidance sceme for cross-track control of underactuated underwater vehicle. *14th Mediterranean Conference on Control and Automation, 2006. MED '06*, 2006.

[Bre08] Morten Breivik. Guidance laws for planar path following. 2008.

[Bø08] Even Børhaug. Phd trail lecture on pipeline inspection, november 28th 2008. PhD Trail Lecture, November 2008.

[DNV] Dnv subsea pipeline requirements (accessed: December 3rd 2008). `http://www.dnv.com/industry/energy/segments/onshore_pipeline/index.asp`.

[EG02] Olav Egeland and Jan Tommy Gravdahl. *Modeling and Simulation for Automatic Control.* Marine Cybernetics, 1 edition, 2002.

[FC90] John Perry Fish and H. Arnold Carr. *Sound underwater images : a guide to the generation and interpretation of side scan sonar data.* Lower Cape Publishing, 2nd edition, 1990.

[For01] Gian Luca Foresti. Visual inspection of sea bottom structures by an autonomous underwater vehicle. *IEEE Transactions on systems, man, and cybernetics - Part B Cybernetics*, 31(5):691–705, 2001.

[Fos02] T. I. Fossen. *Marine Control Systems.* Marine Cybernetics, 2002.

[Gje08] Bjørn Gjelstad. Personal Correspondance, November 2008.

[Hal91] Jan Olav Hallset. A prototype autonomous undewater vehicle for pipeline inspection. *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, 1991.

[Jav94] Bjørn Javling. The ndre-auv flight control system. *IEEE Journal of Oceanic Engineering*, 19(4), 1994.

[JGHV03] B. Jalving, K. Gade, O.K. Hagen, and K. Vestgard. A toolbox of aiding techniques for the hugin auv integrated inertial navigation system. *OCEANS 2003. Proceedings*, 2:1146–1153 Vol.2, Sept. 2003.

[KU03] Kangsoo Kim and Tamaki Ura. Fuel-optimal guidance and tracking control of auv under current interaction. *Proceedings of the 13th International Offshore and Polar Engineering Conference*, 2003.

[NRS04] W. Naeem and S. M. Ahmad R. Sutton. Pure pursuit guidance and model predictive control of an autonomous underwater vehicle for cable/pipeline tracking. *Journal of Marine Science and Environment*, pages 15–25, 2004.

[NSAB03] W. Naeem, R. Sutton, S.M. Ahmad, and R.S. Burns. A review of guidance laws applicable to unmanned underwater vehicles. *The Journal of Navigation*, pages 15–29, 2003.

[PJL05] Pedro K. Paim, Bruno Jouvencel, and Lionel Lapierre. A reactive control approach for pipeline inspection with an auv. *OCEANS 2005 Proceedings of MTS/IEEE*, 2005.

[PK04] Andrew C. Palmer and Roger A. King. *Subsea Pipeline Engineering.* Penwell Books, 2004.

[PRB02] Y.R. Petillot, S.R. Reed, and J.M. Bell. Real time auv pipeline detection and tracking using side scan sonar and multi-beam echo-sounder. *Oceans '02 MTS/IEEE*, 1:217–222 vol.1, Oct. 2002.

[Sea] Seabyte webpage. `http://www.seebyte.com/News/worldRecord141106.html`.

[Sha07] Vikrant P. Shah. Design considerations for engineering autonomous underwater vehicles. Master's thesis, Massachusetts Insitute of Technology, 2007.

[Shn98] N. A Shneydor. *Missile Guidance and Pursuit: Kinematics, Dynamics and Control.* Horwood Publishing, Ltd, 1998.

[SHV06] M. W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Modeling and control.* John Wiley and Sons, Inc, 2006.

[SNA50] The society of naval architects and marine engineers. nomenclature for treating the motion fo a submerged body through a fluid. *Technical and Research Bulletin*, (1-5), 1950.

Appendix: Matlab Scripts

The Descision block

```
function y = descicion(u)

    global WP pipeline_dir trajectory_generated

    %input: Output, eta, Pipeline trajectory,
    eta = u(1:6);
    nu = u(7:12);
    p = u(13:21);
    heading = u(22);
    output = u(23);
    t = u(24); %current time

    lookahead = 5; %5
    lookahead_s = 12; %6

    persistent mode current current_s detected_pos time_since_contact last_known_pos
    % mode = 2 goto; mode=1 serach; mode = 0 track

    if isempty(mode) || isempty(current) || isempty(detected_pos) || isempty(current_s)
        mode = 2; %start in goto mode
        current = 2;
        current_s = 2;
       WP = [eta(1:2), WP];
        detected_pos = [0; 0];
        time_since_contact = 0;
        trajectory_generated=0;
        last_known_pos = [0 ; 0];
    end



    switch mode

        case 2 %goto mode
            r2 = sqrt((WP(1,current)-eta(1))^2 + (WP(2,current)-eta(2))^2);
            y = atan2(WP(2, current)- eta(2), WP(1, current) - eta(1));
               if r2 <= 5
                  current = current +1

                  if output == 1
                       mode = 0
                       detected_pos = p(4:5)
                       return;
                  else
                       mode = 1
                       return;
                  end
               end


        case 1 %search mode Generate trajectory
            if output == 1 % && trajectory_generated ~= 1
                detected_pos = p(1:2)
                y = atan2(detected_pos(2) - eta(2), detected_pos(1) - eta(1))
                mode = 0
                trajectory_generated = 1
                return
            else
               if trajectory_generated == 2
                     [WP, current_s] = generate_WP(last_known_pos, pipeline_dir);
               elseif trajectory_generated == 0
                    [WP, current_s] = generate_initial_search(WP(1:2, 2));
```

```
                end


                r2_t = sqrt((WP(1,current_s)-eta(1))^2 + (WP(2,current_s)-eta(2))^2);

                if r2_t <= 6
                   current_s = current_s + 1
                end
                xi_wp = atan2(WP(2, current_s)-WP(2, current_s -1), WP(1, current_s)
                                                    - WP(1, current_s -1));

                if xi_wp < 0 %fix for not turning all the way around
                    xi_wp = xi_wp + 2*pi;
                end

                e = -(eta(1) - WP(1, current_s -1))*sin(xi_wp) + (eta(2)
                                                    - WP(2, current_s -1))*cos(xi_wp);
                y = xi_wp + atan2(-e, lookahead_s);

                if output == 1 && (trajectory_generated == 1 || trajectory_generated == 3)
                    mode = 0;
                    return;
                elseif output == 1
                    mode = 0
                    detected_pos = p(4:5)
                    trajectory_generated = 1
                    return;
                end

            end

        case 0 %track mode
            xp = heading;
            if output == 0
                %lost measurement
                if time_since_contact == 0
                    time_since_contact = t
                    last_known_pos = p(4:5);
                end
            end

            if abs(xp-eta(6)) > pi/2
                xp = -xp;
            end

            e = -(eta(1) - p(1))*sin(eta(6)) + (eta(2) - p(2))*cos(eta(6));

            xr = atan2(-e, lookahead);

            psi_d = xp + xr;
            y = psi_d;

            if t - time_since_contact == 25 %60 scn3 %25 scn 1 og 2 og 4
                mode = 1 %search mode
                time_since_contact = 0;
                trajectory_generated = 2;
                return;
            end

    end
    last_psi_d = y;
end


function [waypoint, current] = generate_WP(eta_t, pipeline_dir)

    global trajectory_generated

    %position from origo of NED
    r1 = 80;
    r2 = 100;
    r3 = 200;

    %Search boundaries
    theta_min = pipeline_dir - pi/20;
    theta_max = pipeline_dir + pi/20;

    waypoint = [eta_t(1), eta_t(1)+r1*cos(theta_max), eta_t(1)+r2*cos(theta_min),
                              eta_t(1)+r3*cos(theta_max);
                eta_t(2), eta_t(2)+r1*sin(theta_max), eta_t(2)+r2*sin(theta_min),
                              eta_t(2)+r3*sin(theta_max)]

    trajectory_generated = 1;
    current = 2;
```

```
end


function [waypoint, current] = generate_initial_search(eta_0)
    %generate spiral pattern around initial condition.
    global trajectory_generated
    theta = 0:0.1:8*pi; %4 omdreininger
    b = 30;
    r = theta + b*theta; %spiral i polarkoordinater
    [x, y] = pol2cart(theta, r);
    waypoint = [eta_0(1); eta_0(2)];
    for i = 1:15:size(x, 2)
        waypoint = [waypoint, [eta_0(1)+x(i); eta_0(2)+y(i)]]
    end
    trajectory_generated = 1;
    current = 2;

end

function [waypoint, current] = generate_turn_trajectory(detected_pos, eta_t, pipeline_dir)
    %takes in detected point, pipeline direction and wp vector
    %outputs new wp vector with new wps

    global trajectory_generated
    psi = eta_t(6);

    r1 = 20;
    r2 = 50;
    r3 = 60;
    r4 = 60;
    if psi > pi/2
        theta1 = psi + pi/6;
        theta2 = psi + pi/3;
        theta3 = psi + 2*pi/5;
        theta4 = psi + 2*pi/3;
    else
        theta1 = psi - pi/6;
        theta2 = psi - pi/3;
        theta3 = psi - 2*pi/5;
        theta4 = psi - 2*pi/3;
    end

     if (psi > pipeline_dir + pi/8) || psi < (pipeline_dir - pi/8)
        waypoint = [eta_t(1), eta_t(1)+r1*cos(theta1), eta_t(1)+r2*cos(theta2),
                eta_t(1)+r3*cos(theta3), detected_pos(1), detected_pos(1)+r2*cos(pipeline_dir);
                    eta_t(2), eta_t(2)+r1*sin(theta1), eta_t(2)+r2*sin(theta2),
                eta_t(2)+r3*sin(theta3), detected_pos(2), detected_pos(2)+r2*sin(pipeline_dir)]
     else
         waypoint = [eta_t(1), detected_pos(1);
                     eta_t(2), detected_pos(2)];
     end

    trajectory_generated = 3
    current = 2;


end
```

Camera Simulator

```
function [x] = camsim3(u)

%% Initialization
global pipeline focus

t = u(32);
noise = u(33);
variance = u(34);

DimX = 400;
DimY = 400;



persistent output isOutput



if mod(t, 1) == 0 %Every 1 second there is a new sample
    eta = u(1:6);
    bottom = u(13);
%% Rotation matrix
    Rot = [u(14:16) u(17:19) u(20:22)];
```

```
    Trans = [u(23:25) u(26:28) u(29:31)];


    %% Check if pipeline is inside FOV

    z = bottom - eta(3); %altitude

    fov = z*tand(45/2);

    eta_b = Rot(1:2, 1:2)'*eta(1:2);


        x_max = eta_b(1) + fov;
        x_min = eta_b(1) - fov;
        y_max = eta_b(2) + fov;
        y_min = eta_b(2) - fov;


    pipeline_b = zeros(size(pipeline,1), 3);


    pipeline_inside = [];
    for i = 1:size(pipeline,1)


        if z <= 6
            pipeline_b(i,:)= (Rot'*pipeline(i,:)')';
            if (pipeline_b(i,1) <= x_max) && (pipeline_b(i,1) >= x_min) %inside x direction
                if (pipeline_b(i,2) <= y_max) && (pipeline_b(i, 2) >= y_min) %inside y direction
                    pipeline_inside = [pipeline_inside; pipeline_b(i,:)]; %#ok<AGROW>

                    %         disp('ingen punkter')
                end
            end
        else
%               disp('too dark');
        end
    end

    if isempty(pipeline_inside)
    %     disp('No points inside');
        P = zeros(6,1);
        isOutput = 0;
    else
        % convert pipeline_inside into screen cooridnates

        %perspektivligningene
        perspektiv = diag([(1/z)*focus (1/z)*focus]);

        temp = size(pipeline_inside,1);

        temparr = [];
        for i = 1:size(pipeline_inside)
            temparr = [temparr; (perspektiv*(pipeline_inside(i, 1:2)' - eta_b(1:2)))'];
        end
        [r, c, v] = find(0.05 > temparr & temparr > -0.05);
        if isempty(r)
            r = ceil(temp/2);
        end

        B = sortrows(temparr, 1);

        P = [B(1,1:2)';
             temparr(r(1),1:2)';
             B(size(B,1),1:2)'];
         isOutput = 1;
    end

    if noise == 1

        noise = variance.*randn(6,1);

        output = P + noise;
    else
        output = P;
    end

    x = [output; isOutput];
else
    if isempty(output)
        output = zeros(6,1);
        x = [output; isOutput];
    else
        if isempty(output)
```

```
            output = zeros(6,1);
        end
        x = [output; isOutput];
    end
end
```

Kalman Filter

```
function y = kalmanfilter3(u)

global n e W Q P0 x0 focus

persistent P_apr x_apr temp

if isempty(P_apr) || isempty(x_apr) || isempty(temp)
    P_apr = P0;
    x_apr = x0;
    temp = [0; 0];
end

p = u(1:6);

eta = u(7:12);
nu = u(14:19);
z_c = u(13);

output = u(20);
t = u(21);
z_b = z_c + eta(3);

psi = eta(6);

R = [cos(psi) -sin(psi) 0;
     sin(psi) cos(psi) 0;
     0          0       1];


u = [(R(1,1:2)*nu(1:2)); R(1:2, 1:2)'*eta(1:2)];

% model equations
t1 = 1/2000;
t2 = 1/2000;
A = [0 0 -t1 0;
     0 0 0 -t2;
     0 0 -t1 0;
     0 0 0 -t2];
B = [n, 0, 0;
     e, 0, 0;
     0, 0, 0;
     0, 0, 0];
f = focus;

C = [(f/z_c)*cos(psi), (f/z_c)*sin(psi), 0, 0;
     -f/z_c*sin(psi), f/z_c*cos(psi), 0, 0];

D = [0, -f/z_c*cos(psi), -f/z_c*sin(psi);
     0, f/z_c*sin(psi), -f/z_c*cos(psi)];

E = [0, 0;
     0, 0;
    eye(2)];

[Ad, Bd] = c2d(A, B, 0.1);
[Ad, Ed] = c2d(A, E, 0.1);
x = [];

    x_apr = [n*eta(1);
             e*eta(1);
             x_apr(3);
             x_apr(4)];

    x = [x_apr; x_apr; x_apr];

    K = P_apr*C'*inv((C*P_apr*C' + W));

    x_aprm = [];

    for i = 1:4:12
        %predict
        x_apr = Ad*x(i:i+3) + Bd*u;
        x_aprm = [x_aprm; x_apr];
    end
```

```
switch output
    case 1
        for i = 1:2:6
            x_post = x_apr + K*([p(i); p((i+1))] - C*x_apr - D*u);
            x = [x; x_post(1:4)];
        end
    case 0
        %set x = x_aprm
        x = x_aprm;
end
    P_post = (eye(4) - K*C)*P_apr*(eye(4) - K*C)' + K*W*K';


P_apr = Ad*P_post*Ad' + Ed*Q*Ed';

if temp ~= zeros(2,1)
    pipeline_heading = atan2(abs(x(2)-temp(2)), abs(x(1)-temp(1))); %use prior estimate

else
    pipeline_heading = 0;
end

if mod(t,20) == 0
    temp = x(9:10);
end


%% output
y = [x_apr(1:2); z_b; x(1:2); z_b;x(5:6); z_b; x(9:10); z_b; pipeline_heading; diag(P_apr)];
```