Budapesti Műszaki és Gazdaságtudomái Egyetem
Villamosmérnöki és Informatikai Kar
Department of Automation and Applied Informatics

System and Control Design of an Unmanned Surface Vehicle

Diplomaterv

Készítette
Fodor Attila

Konzulens
Kundra László

May 16, 2014

Contents

Kivonat

A Diplomaterv egy részben és teljesen autonóm működésre képes hajó vezérlőrendszerének fejlesztését mutatja be, különös tekintettel az irányítástechnikai, pályatervezési és rendszertechnikai tervezés részleteire. A dokumentum részletesen ismerteti a

Abstract

The Thesis describes the development of a marine Unmanned Surface Vehicle, a boat that is capable of operating in semi- or full autonomous modes. There is

Introduction

Seaborne monitoring and measurements are often expensive and time consuming, though they can have a large impact on the area, where the data have been obtained. In 2011 the naval safe zone around the Fukusima accident was laid down based only on estimates of the radiation contamination[?], because low amount of measurement data was available due to high measurement costs and risks. Another notable area in need of thorough oceanographic surveying is the area of Greenland, because its coastal waters are very poorly mapped up to present day[1]. Ships have to sail in a roundabout way around the island, because the risk of wrecking in unknown littoral area is very high. This is a huge waste of time and natural resources annually, but complete bathymetric survey of the danger zone is impossible due to the shortages of funds and manpower.

Oceanography and Bathymetry

Not many people come across these expressions often. Oceanography can be summarized as a branch of Earth science that studies the ocean. It includes a wide range of topics, like the study of marine ecosystem, ocean currents, plate tectonics and the geology of the sea floor. Bathymetry is the study of underwater depth of lake or ocean floors. The advancing technology allows even more possibilities, like the aforementioned radiological measurements, or general monitoring of the seas. These areas are not the only possible applications, but they are relatively straightforward and general for easy demonstration.

Advantages of autonomous vessels

Oceanographic measurements today are carried out by large crewed ships, but in many cases they could be replaced by a number of smaller crafts, in order to reduce time and cost and increase the available manpower in other tasks. These vessels have the ability to sail previously unsurveyable areas as well, thanks to the shallower draught and smaller size. Although the number of Autonomous Underwater Vehicles (AUV) are rapidly increasing[?], the the Unmanned Surface Vehicles (USV) are still used mostly for military and academic purposes[?].

[Fleet class USV wrapfigure]

Limitations

The most notable limitation of the current mobile robotic technologies are their small size itself. This results in relatively small range based on their smaller power capabilities, and narrow field of application. Conventional electric powered boats can operate reliably only in the littoral or riverine zones, where the mission area is relatively small.

Scope of duties

The goal of this project is to develop an autonomous surface vehicle, which is capable of conducting multiple seaborne measurements. In order to execute an oceanography mission, the ship must be capable of autonomous navigation based on Global Positioning System (GPS) and an Internal Measurement Unit (IMU), to further enhance the precision in hazardous environment.

Figure 1: *The AAUSHIP prototype on her maiden voyage*

Development guidelines

To keep a sensible and extendable system, the control software of the ship follows the guidelines of the Model-Based Design approach. The system software can be divided to two different modules, one that is dependent, and one that is independent from the physical characteristics of the vessel. In order to maximize the extendability of the system, the fix (independent) modules must be completely general, but as thorough as possible, in order to keep the complexity of the changeable (dependent) software of the robot minimal.

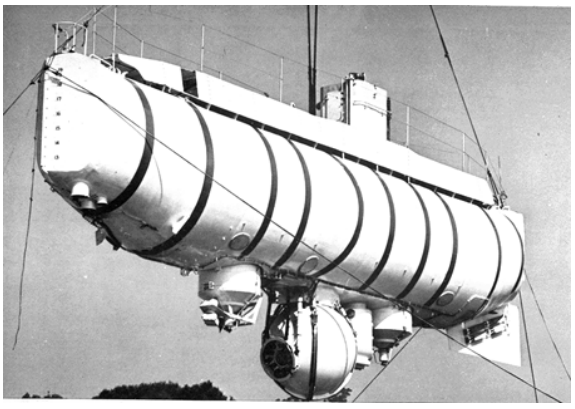


Figure 2: *Bathyscaphe Trieste, the first manned vessel that reached the bottom of Challenger Deep (Mariana Trench)[2]*

System Requirements

Requirement planning and Hazard Analysis has always been a vital process in the development of mission-critical systems. In the automotive, marine, aerospace and military technology industries several Funstional Safety standards exist to guide the development and verification procedures, and ensure that the final product meets a certain quality and level of reliability. Losing or sinking the prototype due to faulty design or construction would have a devastating effect on my thesis, therefore I treated the system as mission-critical, trying to ensure very high reliability. Myí goal isn't the full compliance to all possible safety regulations, but there are some well designed procedures in these standards that can enhance the quality and reveal certain problems that can be dealth width before they become hazards.

General guidelines

Defining the general requirements of the system, especially without a hard functional requirement system, or we could say "out of thin air" is not straightforward. A good entrance to the world of Unmanned surface Vehicles is a document[3] published by the DoN about general recommendations to companies developing Unmanned Surface Vehicles. According to the document, the requirements for developing autonomous boats for the US-Navy can be summarized in the following points:

1. Align to the 4 classes of vehicles, with common core systems and interfaces to the greatest degree possible. Comply with the PEO-LMW-chartered and industry-led unmanned systems standards being developed. Standardize the vehicle interface to the host as well as within the vehicle, with standards for each class and common vehicle functions leveraged among different classes.
2. Wean from the bandwidth. Greater autonomy must be developed to reduce data requirements sent "to" the USV, and more advanced automated target recognition must be developed to reduce the data requirements "from" the USV.
3. Invest in a balanced USV technology program, that includes autonomy, collision avoidance, coupled payloads / weapons, launch / recovery and advanced hull / mechanical / electrical systems
4. Ensure manual, semi-autonomous and autonomous operation modes, but concentrate on manual operation first
5. In case of weaponized systems, investigate and apply the rules of maritime law and laws of war
6. Continously deploy modules to receive operator feedback. Develop with consistent navy guidance. Continue the outreach to Navy operational, doctrine, and training commands to expand and refine employment concepts for USVs

(1.) is only slightly relevant. Alignment to the navy classes in not important, but a standardization of the interfaces is advantageous. Therefore the control system should be designed for a general mobile robotic system, not ship-specific. The map and environment should be extendable to the third dimension, and must be able to track changes over time. The tasks of the low level control must be as specific and as basic as possible. All calculations and control should be implemented in the high level controller. Certain modules of the high level controller must be parametrized, dependent on the low level and hardware characteristics.

The point of these guidelines is to create a system of components with high reusability. Ideally a general Cross-platform High Level Controller controls different kind of vessels, through a standard interface. When changing to a different vehicle, only the Low Level Controller must be replaced, which stores the characteristics of the vessel and implements the actuator control.

(2.) is a major guideline in the development of all unmanned systems. Compliance to the highest level should be sought.

(3.) is very important in production systems, but the current area of application is academic. Ensuring a balanced technology would result in unsutainable amount of work, therefore already available consumer products and rapid prototyping technologies are used to great extent, wherever possible.

(4.) Straightforward, complex features should be introduced gradually. More information about the operation modes in its respective subsection.

(5.) Weaponization is not in the scope of the thesis.

(6.) The academic environment ensures a certain level of feedback. In addition, contact with military and civilian naval personell is planned.

Operation models

A typical oceanography application starts on the shore. A science team analyzes the currently available data, then marks the area of measurements on the map. The map data is transformed to a measurement path by the scientists or by the automatic waypoint planner of the ship. The autonomous surface vehicle is then outfitted with the right sensors for the task, and a manned ship transports it close to its destination. The crew can set the research vessel to manual, automatic or fully autonomous mode.

Manual control In this mode it's possible for the operator to control the movement of the ship in world or body coordinates, like an industrial robot, or the actuators themselves. The primary intent of this mode is for testing and recovery.

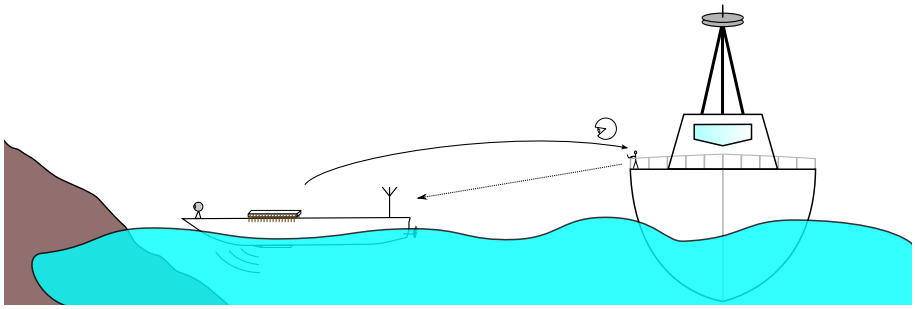


Figure 3: *Manual remote control*

Semi-autonomous (automatic) control In automatic mode the "brains", remain on the manned vessel, and the research craft remains in wireless connection with the Mothership. When the measurements are complete, the oceanographer returns to the mothership. In automatic mode it is always possible to switch to manual control and back, or update the measurement path, etc.

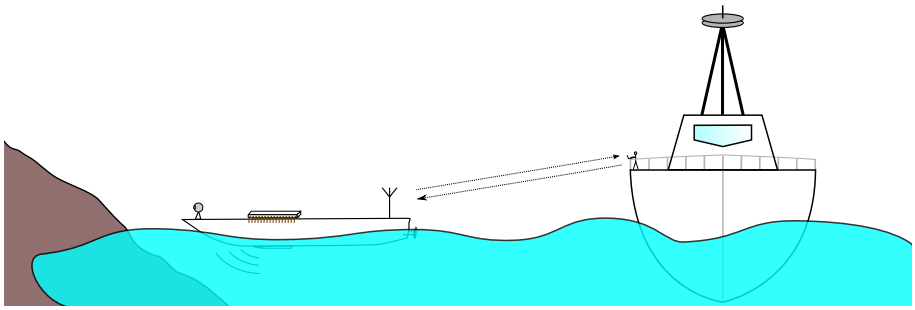


Figure 4: *Automatic supervised control*

Autonomus control In Autonomus mode the ship carries everything that is needed to complete the task. Connection to the mothership can be cut, and the vessel carries the orders out autonomusly. Intervention in this mode is only possible until the oceanographer is in range of the mothership, or a special long range radio or satellite link can be installed to provide a low-bandwidth, but constant connection.

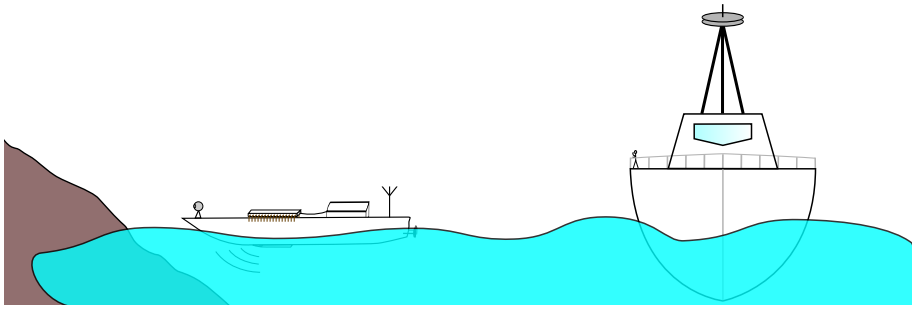


Figure 5: *Autonomus control*

Squadron mode In case there are multiple research vessels, they can form a squadron. In this setup only one of the ships needs to be in range of the crewed mothership to control all of them, as each ship can function as a range-extender unit. Alternatively, one autonomous craft outfitted width long-range communication can control other crafts in automatic mode, taking over the role of the mothership, overseeing and planning the actions of the swarm.

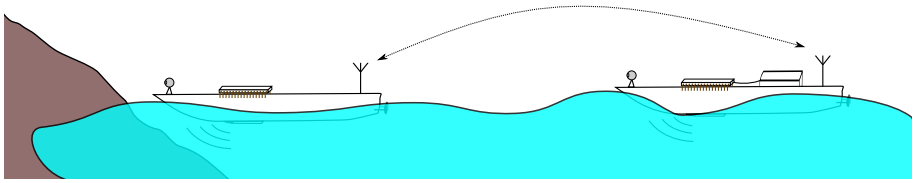


Figure 6: *Squadron mode*

Code: Main.py
https://www.dropbox.com/s/h1067ywmdajkegk/Main.py

The mission planning steps are implemented in Main.py, which is the main part of the program, responsible for initializing the system, setting the target area, mode of routing, navigation, etc. Later Main.py will provide a Graphical User Interface as well.

Navigational requirements of mobile robots

Due to the fact that mobile robots are operating in a constantly changing environment, they must meet certain requirements. The most basic operational requirements can be summarized[4], and thus provide a basis for the navigation amd pathplanning algorithm:

- Convergence
  - The robot must reach the target in finite time, or
  - If the robot is unable to reach the target, it must recognize this in finite time
- Learning
  - The robot must learn its environment during the navigation
  - Creation of maps, obstacles and Feasible Free Space

- Monotonity
  - The shortest route to the target must not increase during movement and learning
- Algorithm
  - The navigation algorithm must not limit the complexity of the environment
  - The environmental dependency of the algorithm must be minimal

The verification of the navigation module is implemented in Python, as a general mobile robot pathplanning problem. A more complex approach is the

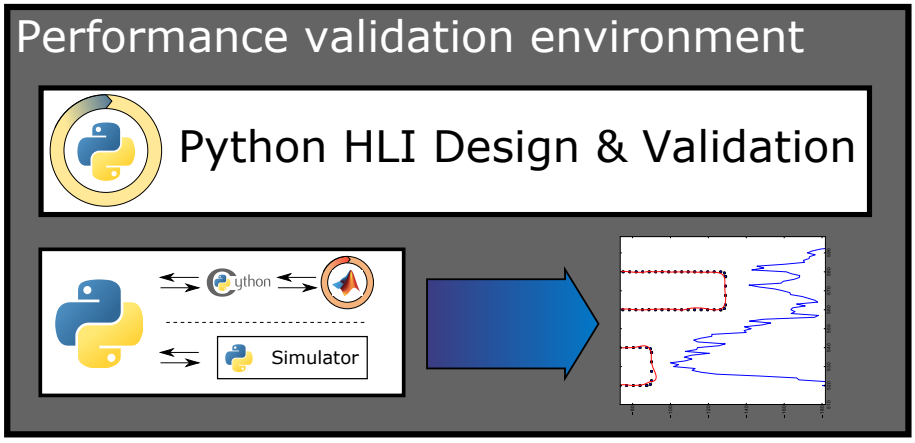


Figure 7

Software requirements of the system

TODO, based on criticality of systems

Requirements of survey ships

In order to execute a rational oceanography task, a number of valid objectives must be set. These objectives are usually one or more of the following[5]:

- A set of geographical measurement locations, with or without time and measurement type conditions
- A certain area of interest
- Maximal action duration requirement
- Other

The task planning is usually carried out by the scientist group, but some auto-planning modes need to be supported. A typical task is the creation of a measurement-grid, with preset definition, in a certain geographical area. The High Level Controller (HLC) uses a Mission planning time algorithm, to support the auto-generation of the waypoints. This module is the "Waypoint planner", which outputs a list of coordinates that contains the measurement points. The Waypoint planner must be operable during the execution of the mission, so a system reset will not cause termination.

Requirements of the control system

Every ship must meet certain maneuvering requirements[6, p. 53]. They must possess course stability, as they must be able to maintain a direction or course. The drift angle, which is the angle between the heading and the actual course of the ship must also not show large fluctuations. The ship must be able to change course relatively fast, without significant overshoot, and the path width must be limited. The ship must also be able relatively well to accelerae and come to a full stop, and must remain well maneuverable during this time. Also, the ship must be able to maneuver with low speeds as well.

Translating these requirements to the language of a control engineer results in the following points:

- The controller can effectively estimate the current state of the ship
- The controller ensures the asymptotic stability of the closed-loop system for all controllable states and all possible inputs

The control system will be developed in MATLAB-Simulink using the guidelines of the Model Based Design approach.

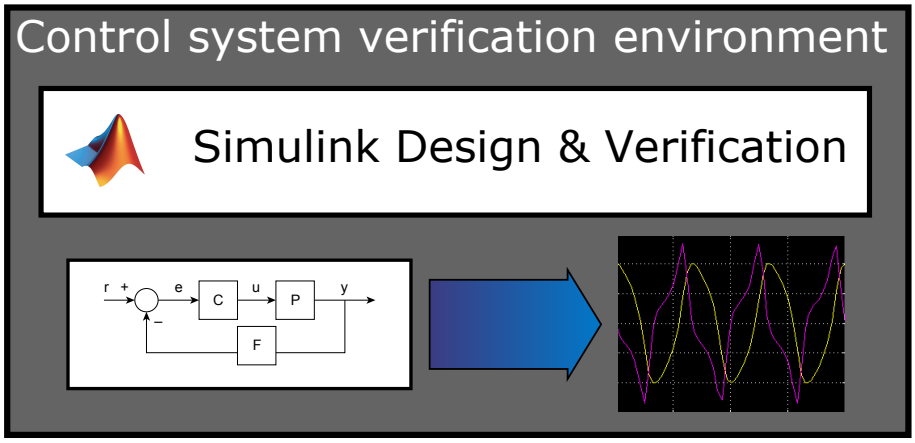


Figure 8: *Control system design and verification*

Requirements of the communication:

TODO

- The module can transmit and receive serial Bluetooth data
- The module can automatically re-establish connection with the slave Bluetooth devices (GPS and client)

Requirements of the client software module

TODO

- The client can transmit and receive serial Bluetooth data
- The client can provide valid real-time information about the system
- The client can automatically re-establish connection with the master Bluetooth device (HLI)

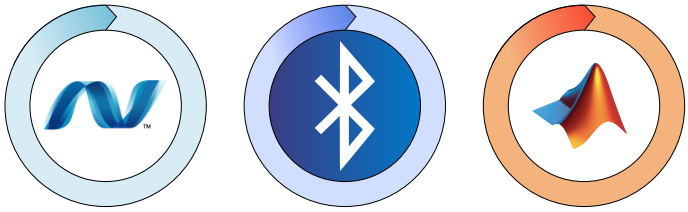


Figure 9

Common ship body and propulsion types

Throughout history, countless ship types have been brought to life, due to the various and ever-changing environmental challenges they had to overcome. Therefore they can be grouped or ordered several different ways, based on their shape, propulsion, area of application, origin and so forth. Modern ships have introduced remarkably advanced or creative propulsion technologies, Sky-sails resembling huge parachutes[7, p 48.] that can be retrofitted to every ship, and even more are still in development, like the Magnetohydrodynamic propulsion[7, p. 61.]. This section will give a quick overview about the most widespread possibilities, but the thesis will analyze only some of them.

Hull types

The lore of shipbuilding[1] is centered around the design of the hull, and it's probably a sensible way to start categorizing ships. The fundamental concept of a ship hull design is the low resistance along the direction of forward motion. A lower drag results in faster top speed and lower energy requirements. Lateral shape is only mildly important in the design of self propelled ships, however sailing with a steep angle to the wind is only possible by balancing the front and lateral resistances carefully[8].

Lateral surface affects the ability of the ship to sail and turn. Large lateral surface results in good wind-response but generally decreases the turning agility and often increases the frontal surface and friction.

Stability of a ship

Stability is the ability of the vessel to return to it's previous position[9]. This righting movement is the result of the relationship between the **Center of Mass** (CoG), and the **Center of Buoyancy** (CoB) of the ship. The stability can be divided to three different movements, independent from each other.

Static stability is the stability at rest, without any external forces. In contrary to the CG, which is a point fixed to the body of the ship[2], the CoG constantly changes it's position as the ship heels or trims. If the CoB and CoG don't align vertically, the ship keeps changing it's attitude, until they do.

[1] Just like many other scholars of ancient history, early masters of shipbuilding were considered artists [?]

[2] Unless the ship features moving ballast, like the sailor of a dinghy

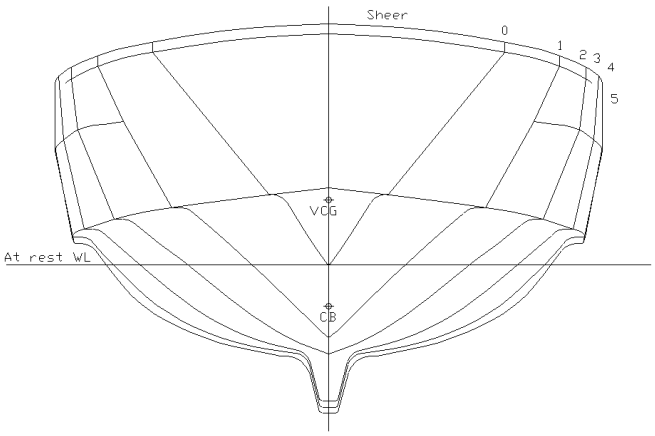


Figure 10: *Static stability of a still boat*
http://www.brayyachtdesign.bc.ca/

Form stability As the boat moves, some area submerges, some reveals from the water. As the hull is heeling, the CoB moves, and depending on the form stability the boat will either overturn, or enough counter-tourqe will build up to reverse the tilting motion.

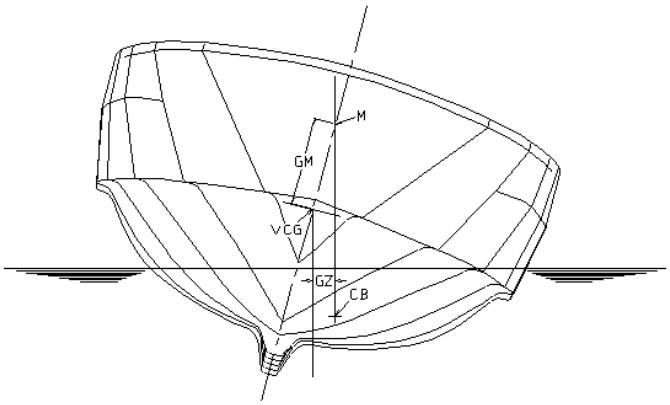


Figure 11: *Form stability of a heeling boat*
http://www.brayyachtdesign.bc.ca/

Dynamic stability is generated when the ship is moving, and the effect is increased with speed. The dynamic stability can either increase or decrease the overall stability, but most modern boats are considerably more stable at higher speeds, due to special hull shaping. As the pressure increases under the hull, the dynamic forces will become stronger compared to the effects of buoyancy, and a round-bottomed hull can become unstable, as the width and shape of the hull beneath the water is significantly narrower than during static buoyancy.

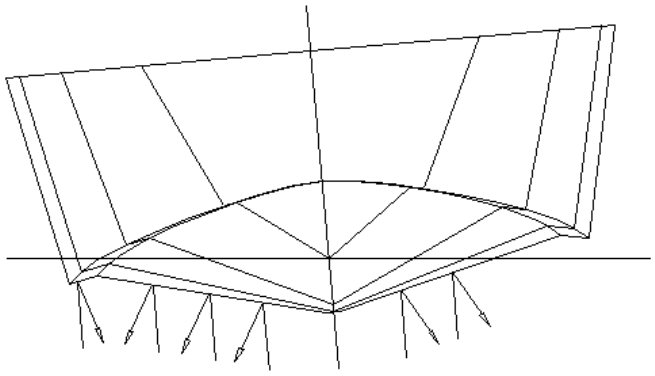


Figure 12: *Dynamic stability of a speeding boat*
http://www.brayyachtdesign.bc.ca/

Stability in general It's notable, that even though a boat with high form and dynamic stability can offer a pleasant still water comfort, also provides a very rough ride at higher speeds, due to the quick hull reactions. On the contrary, a smooth wave-cutter promises constant swaying and vertical movement in a harbor.

One of the earliest efforts to increase stability was the extra **Ballast** added to the bottom of the ship, ranging from a couple of rocks to a keel mounted metal fin. The principle of the ballast is to lower the center of gravity. Many modern sailboats feature a lower CoG than the CoB therefore becoming immune to capsizing. Even after a complete turnover, the boat will eventually return to it's original state[3]. Another, but considerably different effort to ensure stability is a wide **Beam**ed hull. While the ballast provides good ultimate stability, it lacks the initial stability, and the body can gather up quite some tilt until the gravity starts working. Contrary to the ballast, a wide beam provides good initial stability but does not provide the righting movement after a certain point. However, wide beam monohulls can be dangerous, because it reaches the peak of initial stability quickly and the ship is more likely to capsize, but multihulls[4] can boast exceptional stability qualities.

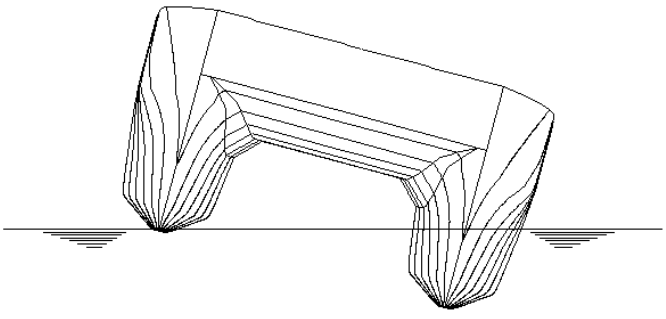


Figure 13: *Wide beam multihull boat*
http://www.brayyachtdesign.bc.ca/

The response speed of the ship can be adjusted by the placement of equipment and cargo

[3]Given that no structural damage happened during this undesirable event
[4]Catamarans, trimarans

on the ship. By moving weight away from the centerline horizontally, the increased **inertia** dampens the quick movements caused by high initial stability. If a ship is close to turning over completely, a well-designed **Superstructure** can still have a positive impact on the stability, and save the day. Unfortunately, a large superstructure is effected by heavy winds, which can be a problem, if the ship lacks a good stability.

In conclusion, adequate stability can be ensured with common sense during the design process, but an all round stability is the result of careful planning.

Sailing rigg Many centuries had to pass until humanity discovered the way to sail sideways to the wind, and even towards the wind in some degree. Modern sailboats still provide a good alternative to motorised ships especially in still water. Contrary to the belief, sailing ships are not the only wind powered ships currently existing, although they proved to be the most efficient. However, Rotor ships are still manufactured[7, p. 47.], mostly as part of a hybrid propulsion system.

Wrapfig Here.

Regulations in the European Union even prevent the usage of petrol engine in Lakes unless sailing in the harbour or under hazardous conditions, and sailboats are still popular for recreational use. The sails, the mast, the ropes and other similar equipment are all part of the rig. The rigging has many various forms depending on the number of masts, size of the ship and the type of sails. The most widespread type of modern and sailing yachts is the single masted rig with two triangular sails. The fore-sail is fixed while the mainsail can be rotated with the boom. There are a lot of additional sail types, especially on the racing boats. Most common of them is the spinnaker, which is a large parabolic sail used for reaching, often featuring colorful graphics.

Wrapfig: To see a long boomed gaffer overtaking from dead upwind with spinnaker on one side, and main, topsail and watersail on the other, is devastating. She seems to fill the sky, while the air for 50yds ahead of her is so still that the victim's pipe-smoke goes straight up as his yacht is inexorably overhauled. —Cunliffe, Tom, (1992). Hand, Reef and Steer, Adlard Coles Nautical, London

Mechanized propeller propulsion Every ship, even before the steam engine was invented featured some sort of mechanical propulsion, most notably rowing galleys. Modern yachts offer petrol or electric engines instead. In case of smaller boats the drive train consists of a shaft connected to the engine, a propeller and a rudder. Jet propulsion, which only the fastest sports and jetskis feature is less common, because they sacrifice efficiency for speed. Large ships and submarines are often outfitted width a nuclear power plant, although for different reasons. In case of large vessels the primary motivation is the almost unlimited range, because a nuclear reactor can provide enough power to keep the ship on the sea for a very long time, and the replenishment of fuel has become only a small nuisance[**?**,

nuclearrange]On the contrary the primary advantage of the nuclear engines in a submarine is the fact that it doesn't require air for the power production process. The underwater range of early submarines were limited because they had to emerge from time to time to the surface where they are exposed to detection and enemy fire[**?**, nuclearsubmarine]

Design conclusions

In case of a civilian unmanned surface vehicle many problems described above can be ignored because of the small size of the boat, the lack of human crew and non-military application. Stability is very important in case of sailing vessels, because stronger wind can create a large torque on the ship. Single and multihull vessels respond to this torque entirely differently. The single hull boat is stabilised by a keel mounted ballast that provides low initial stability, the boat will heel and the effective sail size will reduce, but ultimately will remain stable thanks to the reserve stability. On the contrary, a multi-hull ship featuring high initial stability will respond with rapidly increasing speed. This characteristic has made catamarans and especially trimarans very popular among racers, but they carry the dangers of tripping over. And the righting of a capsized trimaran is hardly possible without external help. In conclusion, a trimaran can easily outperform a classic sailboat in terms of speed and agility, but they can be used safely in a controlled environment only. On the other hand a single-hull boat with a weighted keel and sufficiently watertight inner compartments is much more reliable in the harsh environment of an ocean.

During the thesis two different ships types will be designed. The first vessel is a short range electric motorised boat, featuring a multi-hull trimaran design. Primary application areas are short missions requiring high speed, agility and quick mobilisation. The ship class has been named after a small but agile carnivorous bird: the Kestrel.

The second is a single-hull sailboat capable of conducting long-range measurements where regular maintenance is impossible and high reliability must be ensured. This ship class was named after the iconic bird of long annual travel: the Stork.

[Table comparing the pros and cons of the designs]

Environmental model

In order to estimate the behavior of an object in natural environment, the ambient effects must be estimated first. The two major environmental factors are the effects of the sea and wind. They generate periodic loads that stirs and slowly wears down objects they come in contact width. An ocean surface is almost never still, as waves carry the memory of distant and past storms for a long way. However, the ever-changing and chaotic surface of the sea is composed of regular waves, each having its own frequency, direction and amplitude.

Wave model

A simple, but adequately complex irregular wave model[10, p. 14] can be derived from the basic hydrodynamic principles using superposition of regular waves[6, p. 19].

$$\zeta(x, y, t) = T + H_s * \sum_{i=1}^{10}(cos(k. * (x. * sin(\chi) + y * cos(\chi)) + \Omega * \omega_i * t + \Phi)) \qquad (1)$$

By computing the current water height at every surface points, it's possible to visualize the qeuation above:

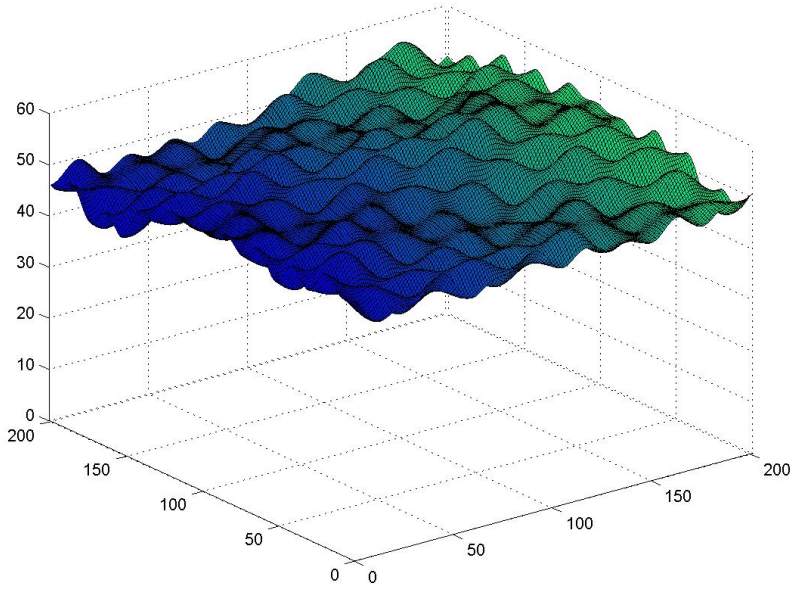


Figure 14: *Simulation result of the irregular wave model*

Wind model

TODO: Beaufort-scale, wind-waves

Behaviour of objects in waves

The movement of ships in water is the result of a number of external forces, like gravity, buoyancy, waves, wind, etc. and the damping effect of the inertia of the ship.

Behaviour in water can be separated to six different motions.

3 translational:

- *Surge* in the longitudinal **x**-direction, positive forwards
- *Sway* in lateral **y**-direction, positive to port (left) side
- *Heave* in the vertival **z**-direction, positive upwards

and 3 rotational:

- Φ - *Roll* around the longitudinal **x**-axis, positive right turning
- Θ - *Pitch* around lateral **y**-axis, positive right turning
- Ψ - *Yaw* around the vertival **z**-axis, positive right turning

These components are built up of multiple sub-components possessing small amplitudes and different frequencies.

[img from ShipHydromechanics 38.pg perhaps?]

In hydrodynamics, the geometrical extent of the floating body must be considered. A wave traveling beneath the ship acts as a force distributed in space and time, first pushing the bow, then the stern of the ship upwards. By sarcificing precision, the approximate forces acting on the hull can be modeled using several approximation points.

Let's introduce the dynamic model of a floating rigid body. As stated earlier, it has 6 movements, which results in 6 state variables:

[table: x1:Surge, x2: Sway, x3: Heave, x4:Roll, x5:Pitch, x6:Yaw]

For each state variable, a state-transition equation can be defined based on external forces:

$$\dot{Surge}_{x_1} = \frac{F_F}{m} \tag{2}$$

$$\dot{Sway}_{x_2} = \frac{F_S}{m} \tag{3}$$

$$\dot{Heave}_{x_3} = \frac{F_V}{m} \tag{4}$$

$$\dot{Roll}_{x_4} = \frac{T_x}{I_x} \tag{5}$$

$$\dot{Pitch}_{x_5} = \frac{T_y}{I_y} \tag{6}$$

$$\dot{Yaw}_{x_6} = \frac{T_z}{I_z} \tag{7}$$

Now the problem is simplified to determining the external forces affecting the body. In order to do this, we apply Archimedes' law[**?**] on a number of control points. Let's start with 4 points, distributed equally from each other and the edges of a rectangular body.

[illustration (Autodesk Inventor?)]

Each control point has a position and height that can be determined based on the state variables and it's position on the surface of the object, using trigonometrical functions. Moreover, each control point has a constant volume, that depends on the volume of the initial object and the number & distribution of control points. In order to compute the actual positions of the control points, we use the 3D rotation formula.

$$R_x(\Theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Theta & -\sin\Theta \\ 0 & \sin\Theta & \cos\Theta \end{bmatrix} \tag{8}$$

$$R_y(\Theta) = \begin{bmatrix} \cos\Theta & 0 & \sin\Theta \\ 0 & 1 & 0 \\ \sin\Theta & 0 & \cos\Theta \end{bmatrix} \tag{9}$$

$$R_z(\Theta) = \begin{bmatrix} \cos\Theta & \sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

It's possible to combine the three rotation matrices into a single formula, but I believe it's more structured and easier to work width when kept separate. Using these formulae the position of a single control point, relative to the ship's centre of mass:

$$\begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix} = R_x(\Phi) * R_y(\Theta) * R_z(\Psi) * \begin{bmatrix} C_a \\ C_b \\ C_c \end{bmatrix} \tag{11}$$

Where a, b and c represents the coordinates in the **body** frame, while x, y and z represents the coordinates in the **local** coordinate system. Now that the **relative** positions of the control points are known, to obtain the **world** coordinates of the points, they must be offset with the position of the body.

$$\begin{bmatrix} W_x \\ W_y \\ W_z \end{bmatrix} = \begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix} = \begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} \tag{12}$$

$$\mathbf{W} = \mathbf{C} + \mathbf{S} \tag{13}$$

Now we only have to solve the wave model equation in these points only, to determine the current approximated waterline. Based on the sinkage of each control point, the current force of buoyancy can be approximated. This force is not always vertical, but normal to the surface of the wave, thus generating not only torque, but slight translational movement as well. It's important to note, that the buoyant forces are calculated based on volume, but the torques are affected by these forces and the center of gravity as well.

Damping As it should have become trivial by now, the motion of the ship, just like the the sea, is the superposition of a number of harmonic movements. The friction, torrents and other effects dampen this oscillation. In order to approximate these restricting forces, the model must be extended with other elements.

Maneuvering model

Simple ship dynamic models can be formulated quickly by defining and analyzing the most common propulsion methods. However, these are only vague approximations of the system, and contain few elements of the complex and powerful hydrodynamics.

First the simple ship models are introduced, then an attempt is made to more accurately describe the dynamical system of a ship.

Differential drive

The differential drive system is very common among simple mobile robotic systems. The movement is based on two separately driven wheels placed on either side of the robot body. It can thus change its direction by varying the relative rate of rotation of its wheels, therefore does not require an additional steering mechanism. A twin-screw ship has a somewhat similar layout. Using the engines positioned in a lateral offset to the centerline of the body, the ship can create a torque and propelling force affecting the body.

[figure about diff drive]

This type of ship can be modeled as a 2-DOF mobile robot (frame fixed to the body of the ship).

The dynamical system of the parallel propulsion engines can be formulated easily:

Figure here! d is the distance of the wheel from the center

$$\dot{x} = \frac{v_1}{2} + \frac{v_2}{2} \tag{14}$$

$$\dot{\Theta} = \frac{v_1 - v_2}{2 * d} \tag{15}$$

From a control engineer's point of view the most important aspect of this control system is its linearity. Designing a basic controller for this type of robot is a walk in the park.

The mechanical advantage of this layout is the very high reliability, because the propellers are fixed in a certain angle, therefore they can be build more roboustly, thus decreasing the chance of physical failture. Another advantage is the ability of the ship to turn around it's central axis, enabling precise maneuvering, however, this is seldom used, because it's very ineffective with ships. Usually older large container-ships employ this type of propulsion. Newer large-scale ship design have kept the multi-screw layout, but also included a rudder directly after the propellers to enhance turning.

Rudder

The rudder is a controllable part of the ship, creating a tourque on the body of the ship, using hydrodynamical effects, much like the rudders of airplanes. The generated torque depends on the traveling speed. A ship like this can usually not rotate around it's center axis, nor travel backwards efficiently.

A representation of a rudder ship can be modeled as a surface car with Ackermann steering. This model is very vague, but is useful for the testing of the higl level course controller and pathplanning.

$$\dot{x} = v \tag{16}$$

$$\dot{\Theta} = \frac{v * tan(\Phi)}{L} \tag{17}$$

The type of the rudder and its placement have a major effect on the controllability of the ship.

[rudder types figures]

All sorts of rudders are required to produce large turning forces, but the rudder size must be balanced between to a good generated torque to draft ratio. To determine the approximate size of an optimal rudder for a given ship, a certain rule of thump value of Det Norske Veritas for a minimum projected rudder area can be applied:

$$A_r approx = \frac{d * Lpp}{100}(1.0 + 25.0 * (fracBL_{pp})^2) \tag{18}$$

where:

$$A_r = projectedrudderarea \tag{19}$$
$$L_{pp} = lengthbetweenperpendiculars \tag{20}$$
$$B = beam \tag{21}$$
$$d = draft \tag{22}$$

"This formula 4.1 applies only to rudder arrangements in which the rudder is located directly behind the propeller. For any other rudder arrangement Det Norske Veritas requires an increase in the rudder area by - at least - 30 percent. A twin screw (or more) arrangement should be combined with rudders located directly behind the propellers for maximum low speed maneuverability. A single rudder placed between two propellers may be inadequate, because the rudder blade does not swing succiently into the tow of a propeller to generate the needed turning moment." (copied, to restructure)

The rudder of the ship acts as an airfoil that produces lift and drag. The [figure] shows a rudder placed in constant flow. The total force can be decomposed. (nem értem, 55. oldal ship hydromechanics)

$$C_L = projectedrudderarea \tag{23}$$
$$L_{pp} = lengthbetweenperpendiculars \tag{24}$$
$$B = beam \tag{25}$$
$$d = draft \tag{26}$$

[figure: rudder forces]

Sailing ship

The sailing ship is usually controlled by the rudder. It's possible to alter the course by adjusting the sails or tilting the mast (e.g.: windsurfing). the center of the sails is shifting, and will produce a torque around the center of lateral surface. There are many forces affecting the sails and the body of the ship, but it's way out of the scope of this text to model all of them. However, they can be generalized in two forces, the lift and the drag.

Lift is the force generated on the sails by the wind, affecting paralell with the body of the ship, and drag is the force generated perpendicular to the body of the ship. These forces are greatly and nonlinearly dependent on the strength and direction of the wind, current speed, air pressure, wind shears, body drift and keel form, local temperature gradients and other circumstances.

Simulating all these effects is a scientific field of its own. Instead, we are using polar sailing diagrams to approximate the maneuvering performance of the boat. These diagrams, often used as a marketing tool are a widespread visual representation of the capabilities of a specific ship. By allowing the software to process these diagrams, the controller and pathplanner can be effectively customized.

The resulting software is heavily based on the supplied sailing diagram, but automatic calibration is also possible using wind sensors and GPS.

$$\dot{x} = v_x \tag{27}$$

$$\dot{y} = v_y \tag{28}$$

$$\dot{v_x} = lift - \frac{v_x}{bodydrag_x} \tag{29}$$

$$\dot{v_y} = drag - \frac{v_y}{bodydrag_y} \tag{30}$$

$$\dot{\Theta} = \frac{v_x * tan(\Phi)}{L} \tag{31}$$

Building simulation environment

Based on a survey completed in the Brazilian and United States Navy the importance of various aspects of the simulation can be determined[10, p. 6]. The result of the analysis show that the most essential areas are the acceleration, deceleration and turning rates of the ship. According to the respondents the simulation of the possible visibility issues are also important, and the environmental effects also scored high on the priority list. The results indicate that that the distribution and the wake of ship is less significant in a virtual environment.

Code: Simulator.py
https://www.dropbox.com/s/agron9anslev6xw/Simulator.py

According to the guidelines of the seamless simulator, this part of the system has its own distinctive object. Everything related to the simulator and everything that is unknown during the mission is stored and handled here.

Uncontrolled system simulation results

Control system design

The control of ships is based on an ancient knowledge, as it is the oldest vehicle known to men.

The path following control problem can be simplified to a course control problem, if the prescribed path is populated by waypoints sufficiently close to each other. Each waypoint is marked as visited it to ship approaches them to a certain none zero distance. However, some external conditions like strong wind or current might cause serious drift that causes the course-based navigation to become dangerous, especially in narrow passages and rivers. Therefore the controller has been extended with position-control.

The combined control problem

The path following problem can be traced back to an inverted pendulum system, which is a typical problem of controlling and unstable equilibrium.

[figure, way to trace back]

The following control system has been developed for the Rudder ship type, in order to parallelize efforts with the RobonAUT controller development, which is generally the same control problem, if the simplified system dynamics are used.

The path following problem can be traced back to a balancing problem, specifically where the ball must be kept in the center of a seesaw: [illusztrációk] We know the dynamic behavior of the system:

$\dot{d} = g - Delta_dot$

So the control problem of the boat (or car) following the path (or line) is in a narrower sense the same as leveling the seesaw with the ball in the center. [labjegyzet: in a narrower sense: this statement is only valid until the ball hits the end of the seesaw (or falls out) and the divergence of the car (δ) is in $[-pi/2, pi/2]$]

A wide range of controllers has been evaluated in order to determine the best approach to solving this control system.

Requirements of control

Controllers for deterministic systems

Using the linearized state transition matrices the transfer function of the system can be formulated using:

$$H(s) = C * (sI - A)^{-1}B + D \tag{32}$$

The resulting system is a 2nd order integrator, which isn't a surprise, if we consider the linearized output d based on the input Φ. Note though, That this transfer function is valid only in a very limited range of the state variables. Due to the nonlinear system dynamics, d is periodic, unless Phi is none zero and not infinitely small.

Using this transfer function a PID steering controller can be calculated to control the system.

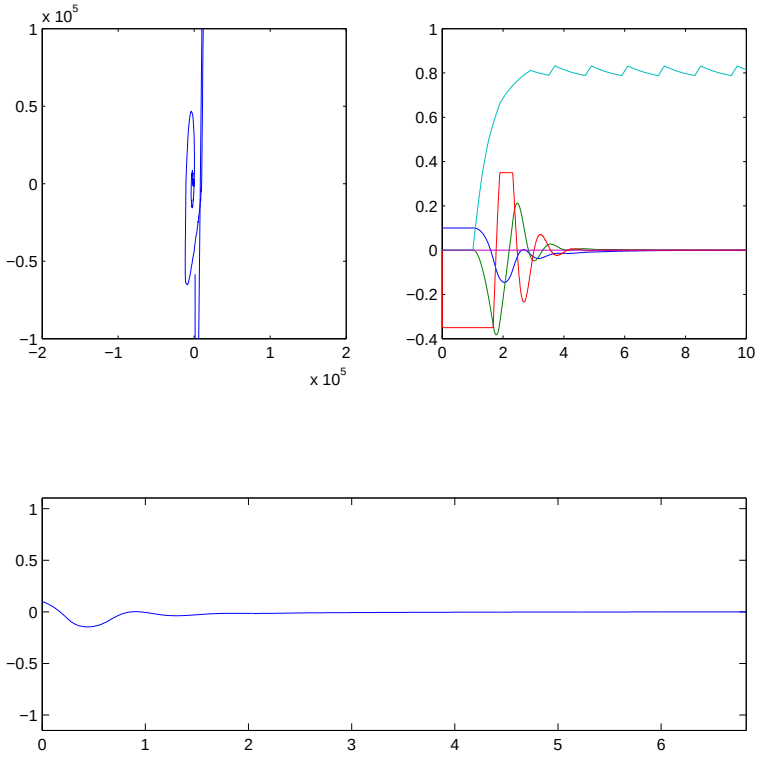


Figure 15

The system response quality is generally low but adequate, but the serious problems arise when the system starts with larger initial conditions that are significantly different from the approximation point.

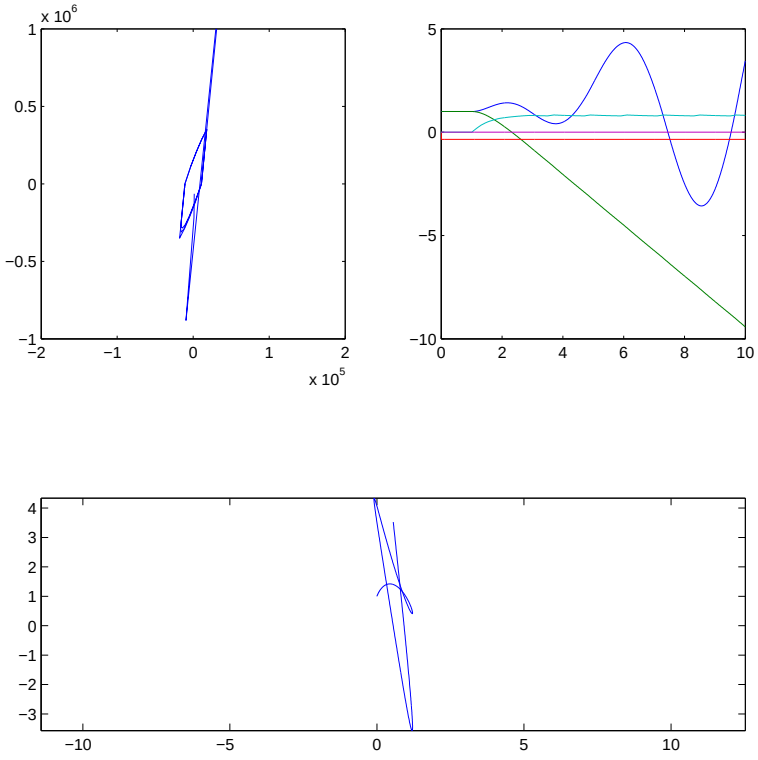


Figure 16

The conclusion is that it's unwise to use the pole cancelation based PID control to regulate an unstable, significantly nonlinear system. The robustness of the PID is generally low, and the system response is slow. The resulting control system is slow and unreliable.

State-feedback controller

In order to enhance the system response, a full state feedback controller can be computed using the same linearized system. The pole placement method allows the efficient control of unstable system, but how does it's robustness fair against the nonlinearity of the system?

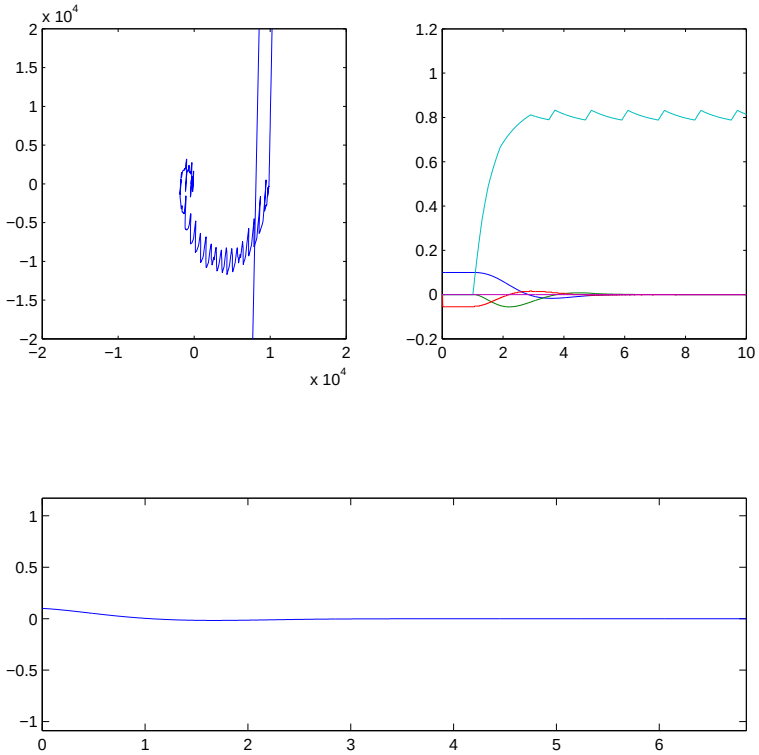


Figure 17: *The physical layout of the system*

The response speed is much better than the respone of the PID controller, but it contains an overshoot, which is the result of the nonlinearities. Here however the pole placement method placed the poles to imperfect locations, instead of entirely missing a pole with a zero.

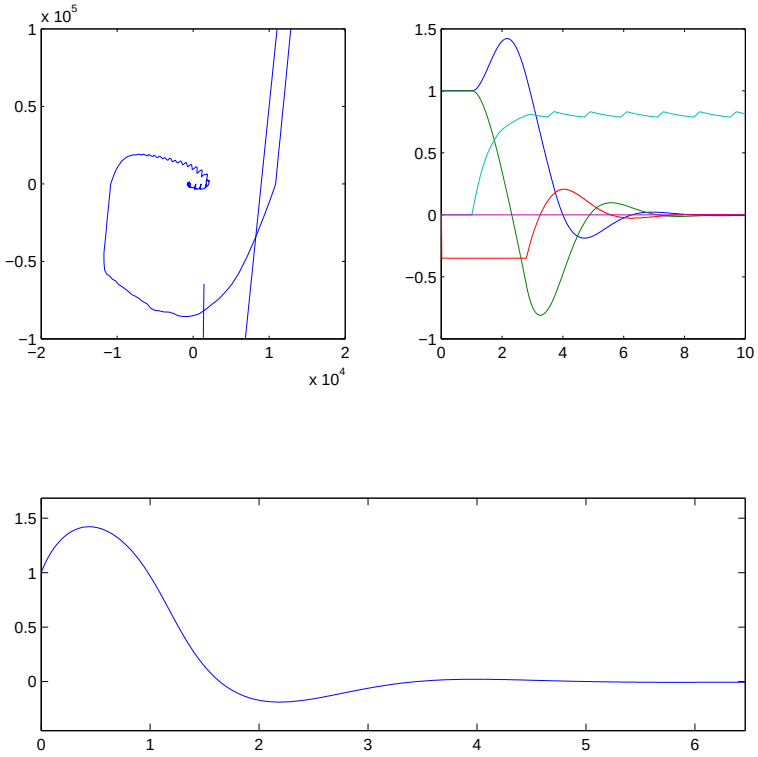


Figure 18: *The physical layout of the system*

As expected, the simulation result of the larger initial conditions is overshoot and oscillation, but the system will quickly settle. Of course this is not good enough yet, because the system is balancing on the edge of instability, and the effects of nonlinear sensors and measurement limitations will have additional negative effects that can cause instability.

Conclusion: the full state feedback linear controller can control the system in most cases, but is unfit to control a mission-critical system, because it has inadequate robustness and reliability.

Hybrid switching state-feedback controller

The instability of the linearized state-feedback controller at high Delta can be corrected by using a Hybrid controller. the Hybrid is the bridge between linear and nonlinear control systems. The controller contains multiple state feedback controller implementations of the same system, linearized around different approximation points. The control task consists of finding the best controller for the current states, and controlling the system as a locally linear system.

For small initial conditions the Hybrid behaves exactly like a regular linear full state feedback controller.

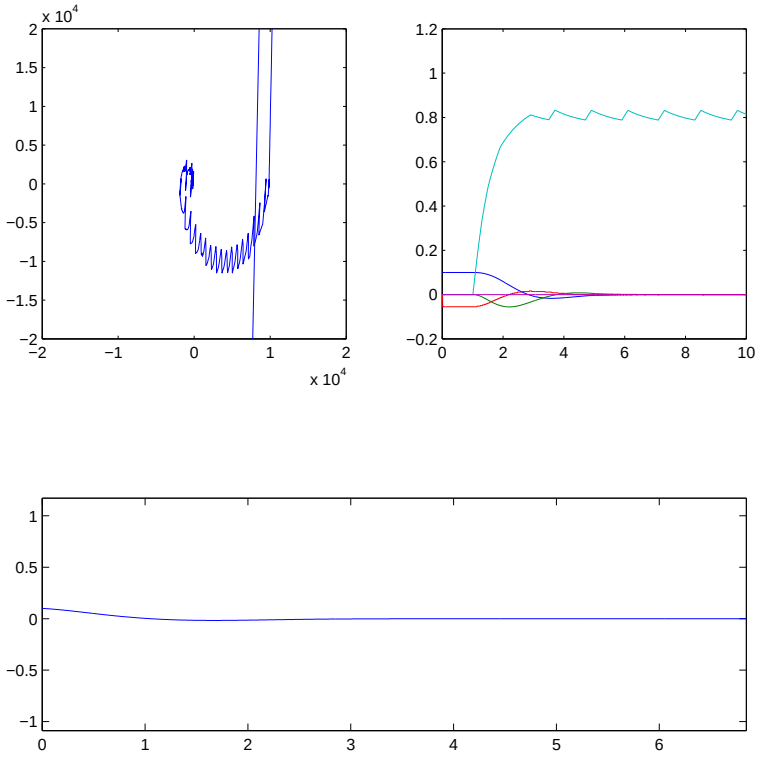


Figure 19

For large initial conditions however, a different controller implementation is selected to increase the response quality. In this case however the system reaction involves only small nonlinearities, therefore a slight decrease in settling time can be observed, but nothing major. As the Φ_{max} is increased, so does the nonlinearity of the control signal, and the more the response quality is increased by the hybrid controller compared to the linear.

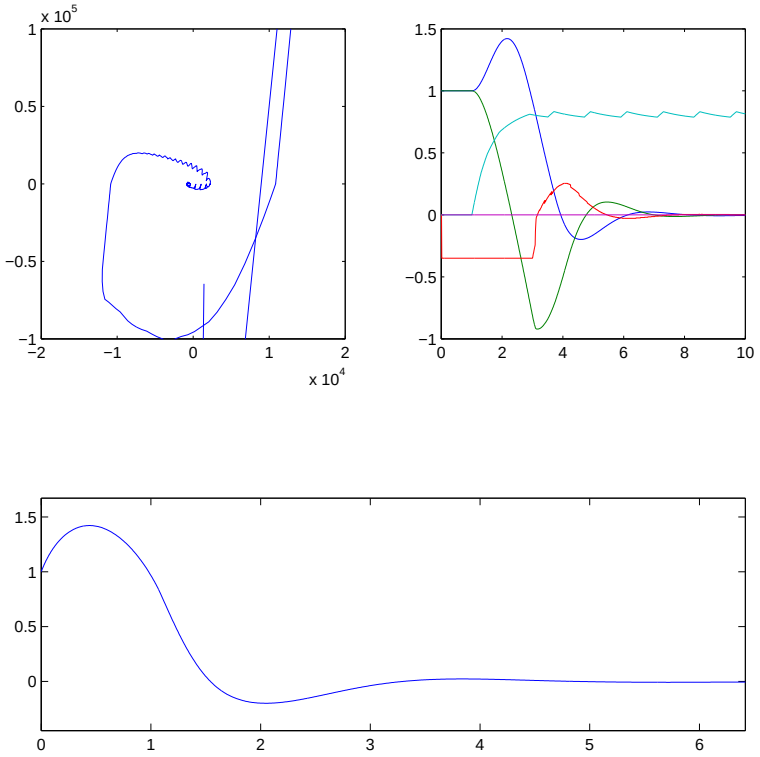


Figure 20

The Hybrid controller has serious dangers and disadvantages however. If the control switching is not implemented correctly, it is possible to destabilize a stable system, and trigger the divergence of the states. Additional disadvantages include the unwanted and unpredictable system responses if a controller switch occurs. This problem can be reduced by controlling the change of the control signal, instead changing the control signal directly, therefore the jumps in the control signal can be eliminated. Another disadvantage is the possible necessity of having a large number of controller implementations. For example, linearizing around 3 state variables with a definition of 10 linearizations each results in 1000 separate controller implementations.

If having a large number of control implementations is not a problem, the process can be automated using the symbolic toolbox of MATLAB (or any other similar software) if the correct nonlinear state-transition functions are know. The symbolic toolbox allows the automated formulation of the Jacobi matrices and the software can compute the linearized implementations cyclikally, then store them in a lookup table.

Fuzzy controller

Though linearizing around many approximation points can result in a relatively smooth switching experience, the number of required controllers can quickly become unreasonably high. The fuzzy controllers address these problems and provide a solution for smooth switching and low number of controllers. Additional advantage of the fuzzy controller is the Fuzzy Control Language, which is the de-facto standard Domain Specific Language for developing fuzzy logic and controllers. The use of FCL helps to understand and design the switching methods which is an important part of the Hybrid controller design.

The Fuzzy controller has an important property that differenciates it from the rest of the controllers evaluated here: the formulation of the control system is based on "liquid" facts, instead of solid data. This results in a less-optimal control, but can ensure stability for systems with unknown dynamics. The fuzzy controller is expected to produce a system response of lesser quality, however it'll have an important role in controlling the more complex, only partially known ship dynamics presented earlier.

Unfortunately there was no time to finish the testing of the Fuzzy controller, therefore the results will be included in a later version of the document.

State Ordinance Controller

The system response can be enhanced further using unconventional controllers. This "State Ordinance Controller" (OC) couldn't be any more unconventional, because it's the result of my early ignorant implementation of a sliding mode controller, but it can provide surprisingly good system responses and boasts a very simple mathematical background and implementation.

The essence of the OC is to determine a priority order of state variables and control the individual states separately by specifying spaces and surfaces. By forcing the state variables into the spaces and onto the surfaces, arbitrary system response can be achieved (of course the physical limitations apply here as well).

The example system is the steering control of the car. The divergence (δ) is forced onto a surface function of distance (d) by controlling Φ. If Φ is controlled correctly, δ will stay on the surface. If the surface has been defined in a way that if δ is on the surface then

$$\dot{d} = f(x)$$

function is negative-definite, then d and δ will approach zero together.

[figure]

Considering the nonlinearities of the system

$$max(|\dot{d}|) = v$$

if $\delta = +-pi/2$. This generates the highest-priority natural space for δ.

So δ is forced to onto the following final function:

[figure]

So a basic implementation of the control system could be:

$$\Phi = sgn(sat(d, [-\frac{\pi}{2}; \frac{\pi}{2}]) - \delta) * \Phi_{max} \tag{33}$$

Of course the controller response would be a high-frequency switching of Φ, resulting in Zeno's paradoxes, which is undesireable. So instead of the signum function another saturation function can be used with values between -1 and 1, with an additional permissible error (ε) parameter. The higher the permissible error, the lower the frequency of the chattering will become.

$$\Phi = \frac{sat(sat(d, [-\frac{\pi}{2}; \frac{\pi}{2}]) - \delta, [-1; 1])}{\varepsilon} * \Phi_{max} \tag{34}$$

Re resulting system response:

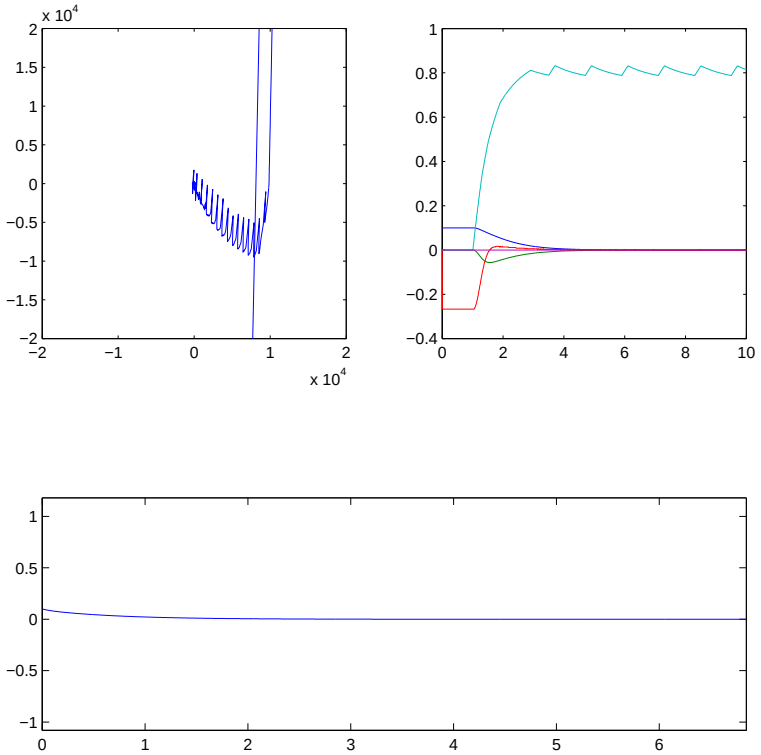


Figure 21

And the resulting system response fir large initial conditions:

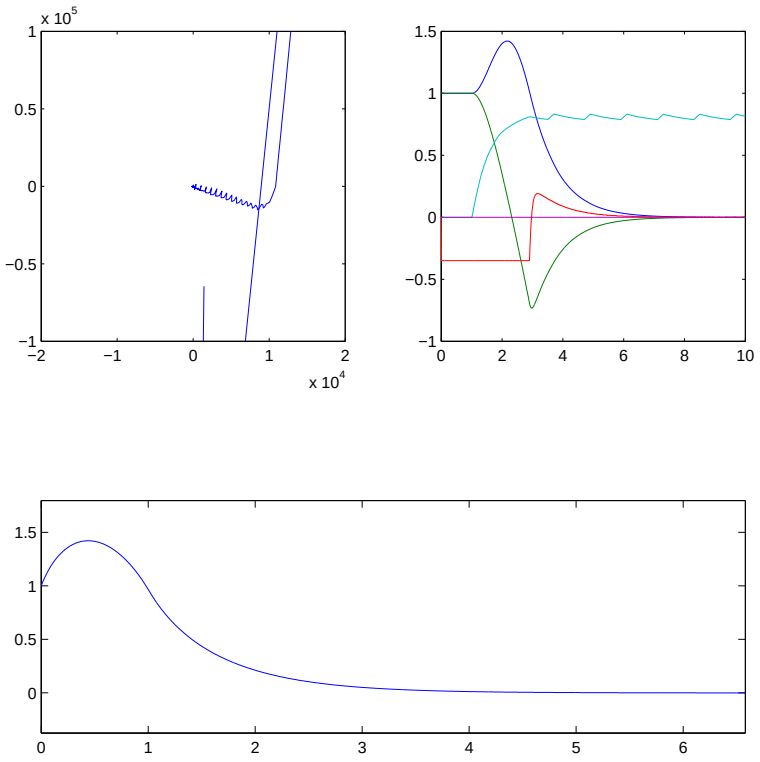


Figure 22

By changing the natural space of δ, different trajectories can be achieved, like a shallower approach to the path:

[figure][figure]

Additional spaces or surfaces can be introduced, like the limitation of Φ based on the speed and the limitation of the acceleration based on Φ for a basic traction control.

The resulting control system excels in robustness and system response speed, but the order of prioritizing the state variables requires some intuition (general rules of thumb can be applied however). The downsides are the virtually nonexistent documentation (excluding this document) and lack of mathematical background (yet).

The result is a robust and fast controller for any initial conditions and some system inaccuracies, and can be considered reliable because the controller can be formulated directly to the nonlinear system.

I consider this type of controller the bridge between fuzzy control and the Sliding Mode Controller.

Sliding Mode Controller

The Sliding Mode Controller is an ambivalent control method. It's a very powerful tool to directly control nonlinear systems, but the formulation can be tricky.

"The common Lyapunov function is an elusive beast that you can almost never find." [Magnus Egerstedt, Associate Chair for Research and External Affairs, Georgia Institute of Technology]

The central concept of the Sliding Mode Controller is to combine the deviations of the state variables from the reference input and unite them in the combined error function. Then use the Lyapunov function of the dynamical system to prove that there exists a controller that the trajectory of the combined error will hit a surface, starting from every possible initial condition, and then it will move along the surface to reach zero.

In practice a Lyapunov candidate function can be determined, then design the controller based on the combined error function. Formal proving that such a controller exists is very hard, but in the current control case general considerations can lead to the same conclusion: it does. If $\Phi = 0$ then the system is stable (not asymptotically stable, but stable). Therefore a sliding surface can be determined for the combined error function.

The system response is quick and lacks the overshoot

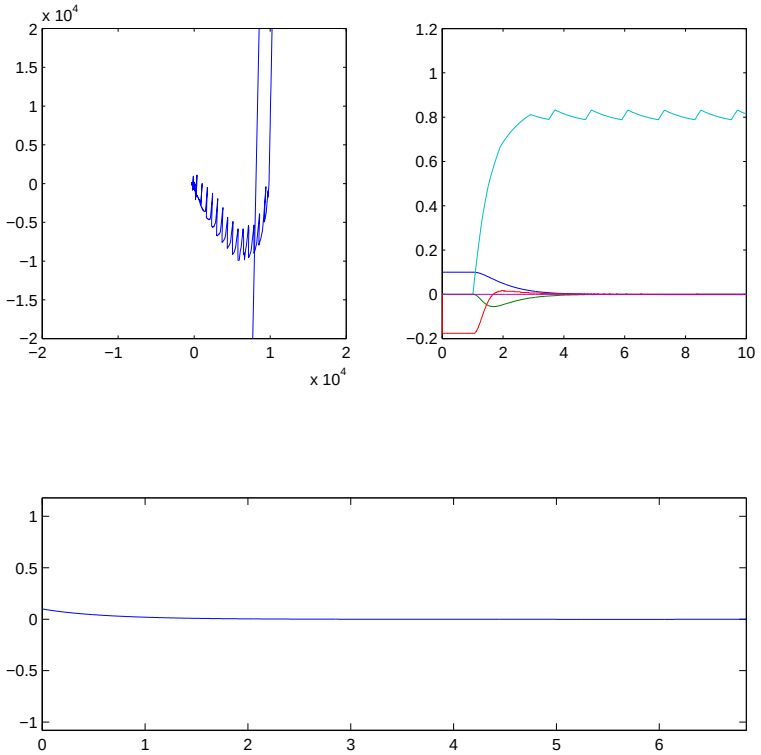


Figure 23: *The physical layout of the system*

And very reliable for larger initial conditions as well.

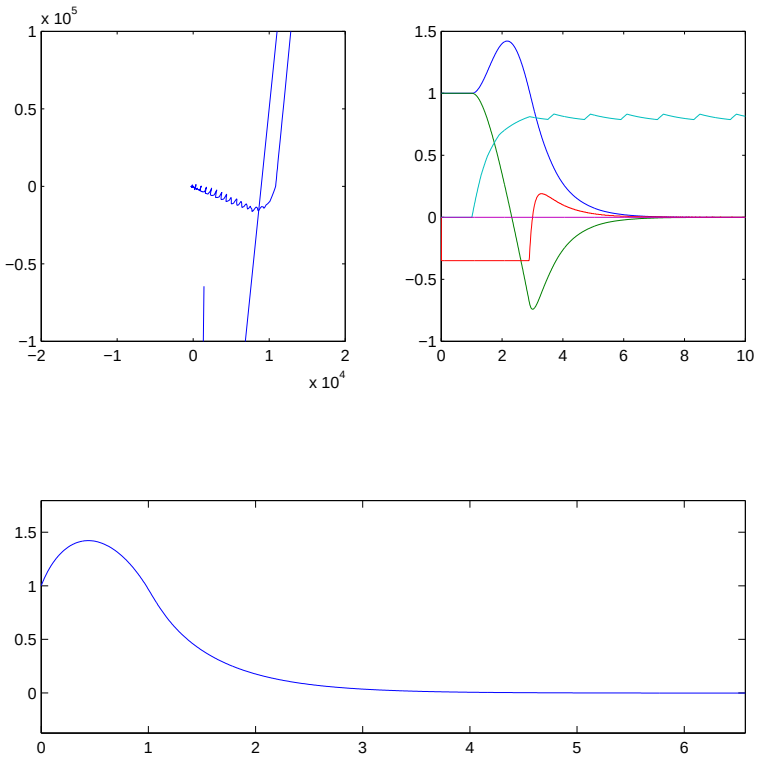


Figure 24: *The physical layout of the system*

Model predictive control

The model predictive control was not evaluated, because the control problem at hand allows quick data acquisition and has virtually zero time-delays. The real strength of model-based control is to predict the states and outputs of a complex system based on numerous inputs, if relatively large time delays are present. The model-based control is ineffective at small scale operations, because it's very CPU-heavy, and it's prediction capabilities are useless with a fast system responses.

Alas, a precise model of a large scale ship includes several underlying states with delayed effects, thus calling for a model predictive control, because a single tiny overshoot in the orientation control of a container-transport can be measured in kilograms of fuel.

Comparsion of results

The evaluation results of the controller types are depicted in the following diagram:

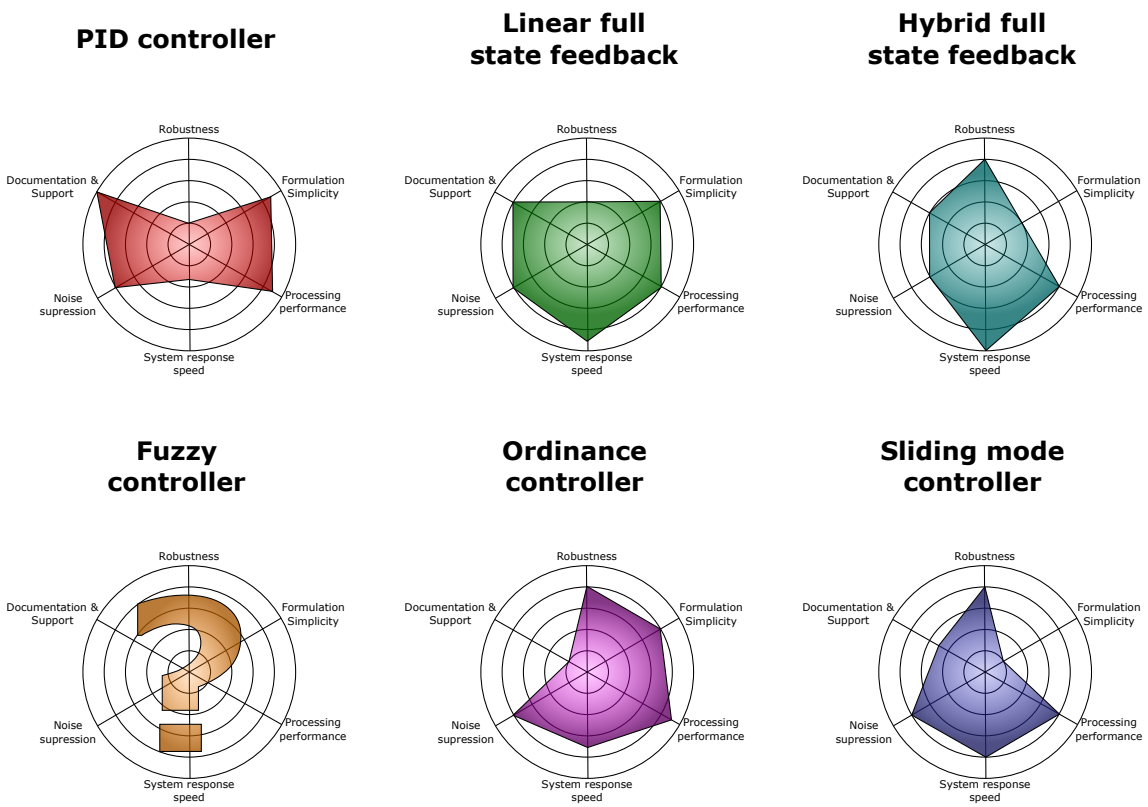


Figure 25: *The physical layout of the system*

Final results can't be concluded yet, because the evaluation of the Fuzzy controller is missing, and the final system is expected to be much more complex than the current test system. Also extensive research must be made to determine the robustness against noise and limited measurements and outputs.

However, the linear full state feedback controller has proven itself to be superior to the other evaluated control systems, because of it's simple formulation and quality system respone. However, the reliability of the Linear state-feedback controller quickly decreases as the system nonlinearities and uncertanties increase. In that case the use of a sliding mode controller or ordinance controller is justified, depending on the complexity of the system. The ordinance controller might provide quick and effective results, but if an unambiguous priority can't be established, then the formulation of a comprehensive sliding mode controller is necessary.

Logical system

This section will give a preliminary overview about the logical and physical layout of the system. The design steps were implemented using the guidelines of the Model Based Design approach, therefore the the logical layout resembles the physical. However, for a lighter approach, the layouts will be introduced separately and connected later.

Concept of separation

This basic layout describes the logical connections of the system:

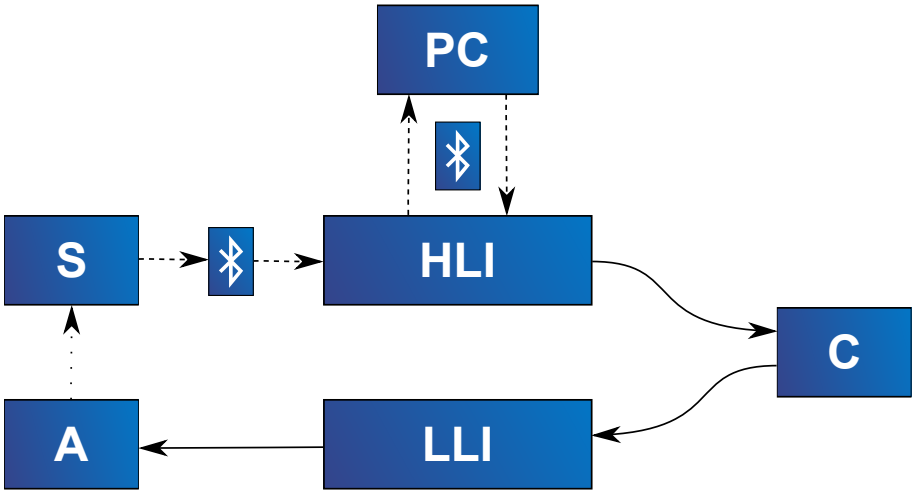


Figure 26: *The logical layout of the system*

The High Level Interface (HLI) processes the position and orientation sensor data input from Bluetooth serial port. During regular navigation the HLI applies the filtering to the inputs, then outputs the estimated orientation divergence from the path, and the estimated distance from the path. The Controller (C) processes this data and returns the required inputs to the system. The LLI processes this physical signal and transforms them to PWM signals, then applies it to the power electronics and the Actuators (A), which will have the final effect on the system.

It's important to note in this diagram, that the Controller and is separated from the rest of the software. This enables the user to radically modify the configuration of the robot, or try multiple controller implementations and switch between them real-time. The Such a switching system can handle possibly expected failures and breakdowns up to a certain level (like the malfunction of one of the engines of a twin-propeller aircraft).

System design

Physical layout

The more detailed physical layout diagram shows the actual relations of the modules of the system to each other. There are several rapid prototyping technologies working together, all of them based on very high level languages.

The embedded system is running Angström Linux, which is a lightweight linux distribution. It enabled the use of Python and USB Hardware-extensions, a generic cheap USB Bluetooth Dongle and a webcam (The webcam is working, but it's not supported yet on the software level). Using Linux as the operating system of the embedded device has the additional advantage of manually controlling and updating the system through SSH.

The HLI and LLI are both implemented in Python. They communicate through function calls implemented via Cython. The actual Controller and Kalman Filter run natively, the .c/.h source code is compiled to .so files by the Cython compiler, which can be imported into the main python software as a static function library.

The source code files are generated by MATLAB Embedded Coder, based on the Simulink model of the control system. The resulting functions are more like objects, containing inner states and member functions, but only one instance can exist of the same controller object.

The GPIOs are controlled from Python as well, using system calls.

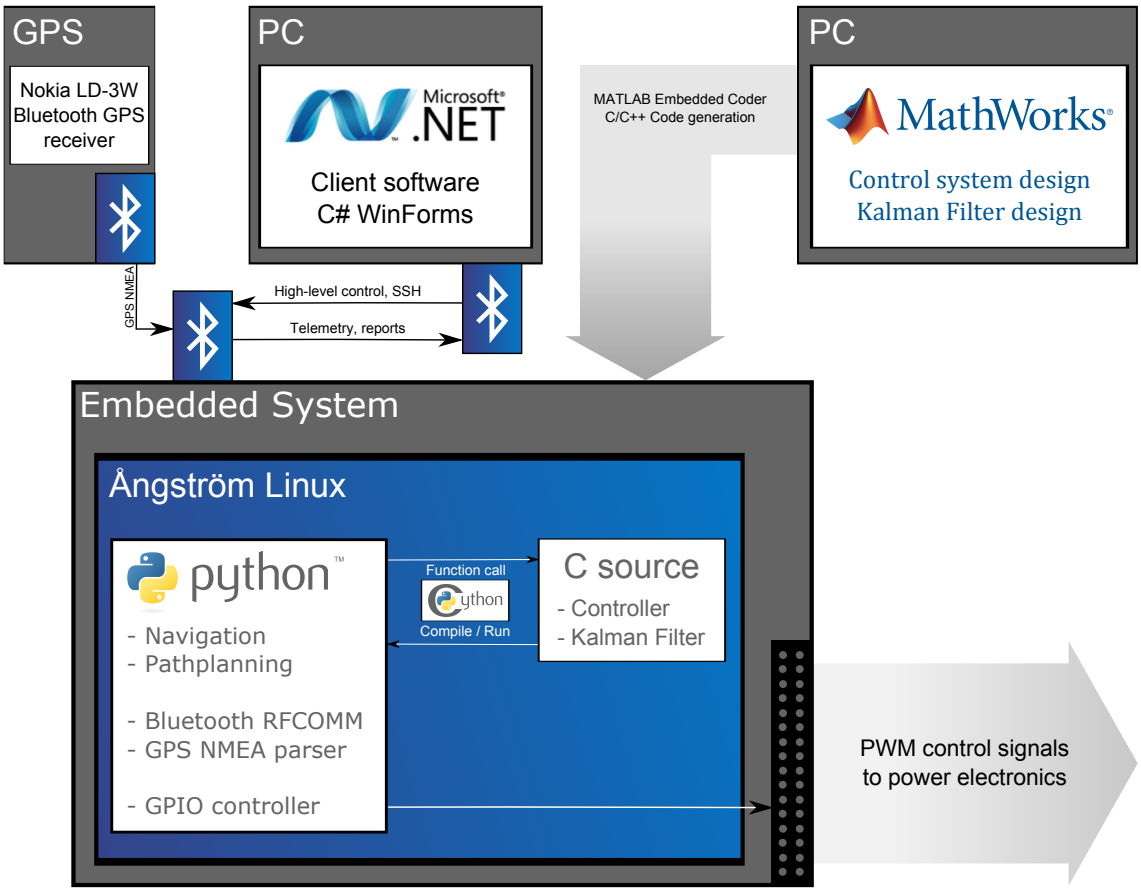


Figure 27: *The physical layout of the system*

The Embedded system has an USB Bluetooth hardware extension that communicates with the GPS receiver and the Client software. It's important to note, that the GPS receiver and the Embedded system are physically close to each other (relative to the scale of the ship), always within the range of the Bluetooth connection. Using wireless communication however enables the distant placement (relative to the scale of the circuit board) in order to ensure high GPS reception and a protected environment for the embedded system as well (e.g.: top of the mast and the belly of the boat).

GPS Acquisition

The localization procedure is implemented by a GPS (Global Positioning System) receiver, connected to the BBB via Bluetooth. The actual device is a Nokia LD-3W GPS device produced for Bluetooth-enabled smartphones without GPS connectivity. The receiver includes a battery with a relatively long life. Solar charging of the device is possible. [wrapfigure] The Nokia LD-3W transforms the GPS signals to NMEA sentences that can be parsed to extract the current position, speed and much else. The NMEA 0183 is a standard marine serial electronics communication interface of the National Marine Electronics Association [`www.nmea.org/content/nmea_standards/nmea_0183_v_410.asp`].

The communication over the NMEA 0183 interface consists of NMEA sentences. There are many predefined sentence templates, but the Nokia LD-3W supports only some of them.

The beginning of each sentence is marked by a $ character, immediately followed by the 5 character of the Sentence type that defines the interpretation. The data fields are submitted in a specific order, distinguished from each other by a comma character.

For the purpose of basic navigation, the processing of the GPRMA sentence is more than sufficient. A typical Sentence of this kind looks like the following:

(Graphic) $GPRMA, Data status, Latitude, N/S, Longitude, W/E, N/A, N/A, Speed, Course, Magnetic variation, Direction of variation(E/W), Checksum

$ - Sentence start character Data status - A = OK, B = Warning Latitude - North / South position on the Globe. Lines of constant latitude run parallel to the equator. Longitude - East / West position on the Globe. Lines of constant latitude run perpendicular to the equator. Speed - Speed over ground, in knots Course - Course over ground in degrees (90 -> East) Magnetic variation - E/W angle correction required to apply to the compass at the current position to determine True North. Magnetic variation effect increases towards the poles. Checksum - Can be checked for sentence errors.

Along the GPRMA, the GPGGA sentences provide useful informations about the position.

$GPGGA, Time, Latitude, N/S, Longitude, E/W, Fix quality, Satellites, Precision, Altitude, Unit, Height above ellipsoid, Unit, DGPS last update, DGPS station ID, Checksum

Time - current time in HHMMSS Fix quality - 0: no fix, 1: GPS fix, 2: DGPS fix Satellites - Number of fixed satellites Precision - Horizontal dilution of precision: sufficient under 2.0[ref] Altitude - Current altitude on a sphere Height above ellipsoid - Height of Geoid above WGS84 ellipsoid: used for correction of Altitude

The GPS acquisition is implemented in Python by the Communication module of the HLI. The script uses linux-specific system calls to connect to the Bluetooth device and receive data, using the PyBluez module.

Bluetooth connectivity

Sensors and effectors

Hardware design

The central unit of the system is the BeagleBone Black single-board computer. This computer produced by Texas Instruments was chosen because it's high performance to cost ratio (important for Python) and it's extensive peripheral support. The BeagleBone Black (BBB) is shipped with preinstalled Angström Linux, which is a lightweight Linux distribution developed for embedded devices. The advantages of using a linux distribution as an embedded operating system include the easy remote access to the system core via SSH, integrated package manager (opkg) with precompiled software distributions for the system (Python, bluez), and generally high community support.

The BBB doesn't contain any integrated sensors, but can be extended with Capes, specialized for different tasks. As the BMESHIP is in a prototype stage, only the most essential sensors will be installed, without the assistance of Capes.

The system contains only a small amount of electronics, assembled on a breadboard. The power to the BeagleBone is supplied from the 12V battery by a 5V voltage regulator IC through USB Cable. Unfortunately the BeagleBone shuts down the USB devices if it's powered through the regular 2.1 / 5.5 mm power cable, therefore we "trick" the device into thinking that it's supplied from a regular usb device.

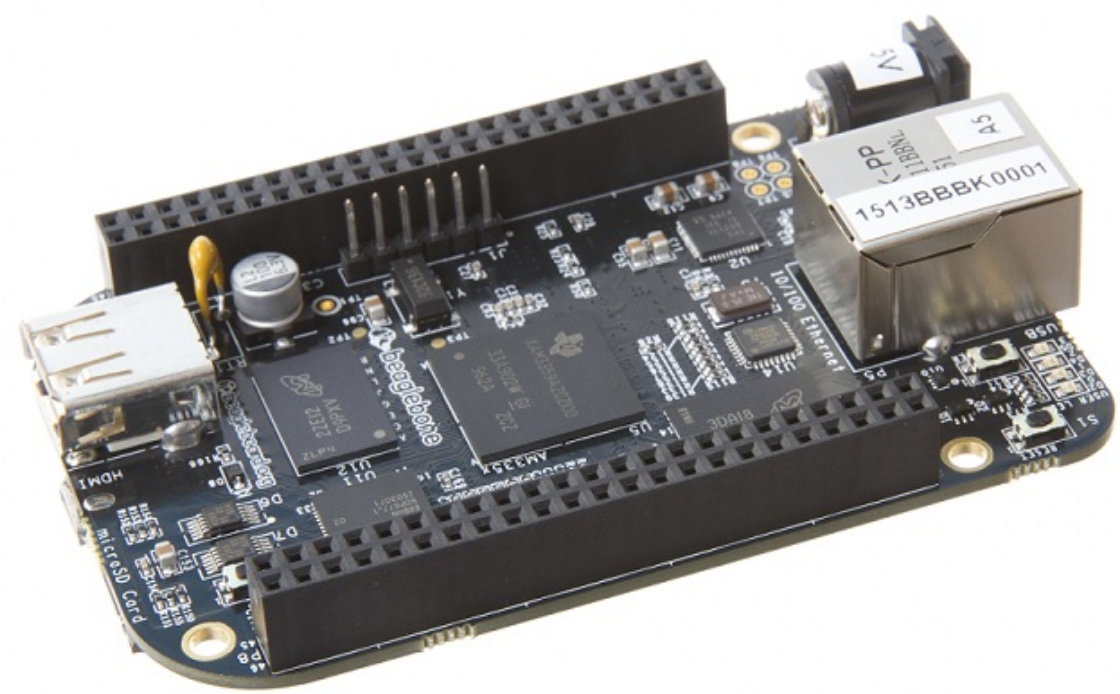


Figure 28: *The physical layout of the system*

Additional electronics include the motor controller board and the servo control, but unfortunately not all of the required parts are available yet.

Remote control

The Man-in-the-loop operation mode has been realized using a commercial two-channel 2.4GHz RC transmitter and receiver. CH1 is responsible for the throttle, or rig control, while CH2 controls the course through the rudder. The output of both channels are standard servo signals.

[wrapfigure about servo signals]

The processing of the servo signals require very high sampling, which is hardly possible with the already presented prototype-boards. A quick but inaccurate solution can be used to process the servo signals, using the ADC

Wind sensor

In order to effectively control a sailing ship, the wind direction must be determined with sufficient precision relative to the body of the ship. Note that sufficient in this context means a generally low precision requirement, but the availability and reliability of that measurement is critical. Several different methods of detection have been investigated:

Incremental encoder Detection is possible using incremental encoders, and a very high angle precision can be assured. However, in case of a system error or restart, the encoder must be re-calibrated to rely on it again. In addition, some amount of missed signals is inevitable, which would introduce an absolute error to the measurement that can't be neglected.

+ Low cost + Very high precision + Quick response

- On-site hardware processing required - Requires frequent recalibration - Requires external recalibration hardware

Of course an absolute angle encoder is immune to these problems. An enclosed system is capable of providing precise, reliable and always available data about the wind direction. Alas, the price range of the cheapest of these sensors is way out of the scope of the current project.

+ High precision + Quick response + Always available, reliable

- Very-very expensive

A slightly different approach is to use a multiturn potentiometer and estimate the angle position using an Analog to Digital Converter. Though unfortunately such multiturn potentiometers are unable to perform arbitrary number of rotations in the same direction, therefore their ability is limited.

+ Low cost + Easy to process + sufficient precision

- Can't rotate infinitely

There's a somewhat similar approach that consists of a permanent magnet and a number of analog magnetic field sensors. Either the magnet or the Hall effect sensors are attached to the rotating wind-indicator, while the other part remains fixed to the ship. Processing the signals of 2 (or 4 to compensate for noise) sensors with an ADC and applying adequate trigonometrical processing on the raw data results in an absolute wind direction measurement. Though the required sensor post-processing is heavy on a smaller microcontroller, because of the required inverse trigonometrical functions, this is hardly a factor on a single board computer like the BeagleBone.

+ Low cost + Sufficient precision + Absolute, reliable

- Complex software post-processing required - Susceptible to noise

After carefully considering the possibilities, the Magnet-based wind detection was chosen. Optionally a wind speed detector can be assembled using incremental encoders.

[img and schematic of the sensor]

Hardware

Assembly

Path planning

Basic marine navigation is plain, because there are few distracting elements in the environment. Althpugh the maneuvering of ships is rarely restricted to open seas, without any obstacles to avoid, and even then sailing vessels require active planning, as they rarely move parallel to the wind direction, or other disturbances may apply, like ocean currents.

The problems of navigation and pathplanning will be introduced gradually. There are two kinds of boats (bowerboats and sailboats), two types of environments (still water and riverine) and two areas of applications (shortest route between points or shortest traversal of an area). In summary there are eight problems to be solved, each getting more complex than the one before. Fortunately, it's enough to solve only some key problems, then combine the findings to a roboust pathplanner.

Model of maps and obstacles

The forms of obstacles in the sea can vary between wide levels of complexity. The operation space can be either be completely open or closed, and they might contain very simple or very complex inner structures. Or the free space can be just a strait or as narrow as a river. In the latter case especially, the local traffic of other vehicles must never be ignored, or accidents might occur.

Figure 29: *Isle of Skye and Lake Balaton photographed by Chris Hadfield*

Global Path planning

Several algorithms have been developed for global path planning, and this field of study is still researched extensively. The different methods result in very different paths, some of them are optimal, others are not, and some algorithms might even fail to create a path at all. Their performance varies greatly by the different environments they are executed in. A complete overview of all possible motion planning strategies is not in the scope of this document. Some that might be fast, reliable and optimal in certain areas might fail completely in others. In order to find the best navigation solution for each environment, the following algorithms will be analysed and compared to each other in different settings.

Cell decomposition Cell decomposition methods are the most used and researched path planning field in outdoor robotics [**?**]pathplanningoverview). The region is broken down into discrete, non-overlapping fields. the union of these fields is the original region. The method has three sub-categories, the Approximate, Adaptive and Exact Cell Decomposition. We will only discuss the Approximate method.

The most widespread of all Cell Decomposition algorithms is the A* search. It lays a grid down over the area to obtain the cell points, representing a graph. Then a best-first search is used to determine the least-cost path between the starting point and the goal. While traversing the graph, A* follows the path of lowest expected total cost. It keeps a sorted priority queue of the cells that need to be examined.

The order of the nodes to be visited are determined based on a cost function based on the past path-cost and the future-cost estimated by heuristics.

[image collage showing how A* works in different environments]

Roadmap Roadmap methods use sparse graphs to represent the connectivity between free spaces. The most commonly known algorithms in this group are the visibility graph, or Voronoi-diagram based. Although they have common basics and restrictions, they are significantly different. The former produces an optimal, but unsafe path, as it passes very close to the vertices of the obstacles, the latter generates a rather safe, but suboptimal path, because it aims to keep maximum distance from surrounding objects.

A recent approach of the problem is the Probabilistic Road Map pathplanning (PRM). It generates several random nodes around the map and checks their connectivity. The PRM results in a quick but imperfect roadmap, as it sometimes fails to detect existing connectivity or becomes inefficient, especially in case of narrow passages between vast open spaces. This problem can be solved by a different node placement strategy based on gaussian distribution distance from impassable areas.

Wavefront

Comparison of algorithms [comparison of path length, execution time, execution efficiency (path length / execution time) in different settings]

Local planning and collision avoidance

Navigation of motorboats

The navigation of a motorboat in an open space is the most basic path planning problem. If there are no moving obsticles in sight, our only task is to keep the boat from running

ashore. In order to do this, we can use a geometric pathplanning algorithm, the visibility graph search method to find the shortest euclidean path.

[image of solution]

It's important however, that turning with ships can be a costly maneuver. Therefore the cost distance between each node should be increased with an angle cost as well, as in some cases the ship must slow down to complete the maneuver.

In case of a very complex environment, or if for some reason the visibility graph algorithm can not provide a shortest route, a grid-based ripple algorithm is used instead. In this case it's hard to incorporate the turning penality into the algorithm, so this is an emergency-pathplanning method only.

[image of ripple solution]

Navigation of sailboats

The navigation of sailboats is much more complex, even in a static environment. The navigation problem has many solutions, and it's hard to determine which is optimal.

[]

Navigation in moving water

In addition to the problems that a small and crowded

Map reading

The main method of navigation is via GPS and known shoreline database. The ship must be outfitted with proximity sensors later, in order to ensure safe navigation through traffic. During the simulation a reference shore is being generated that simulates a typical fjordic environment: Figure 30. The map generator is using Random Walk Functions to generate the coastline(s).

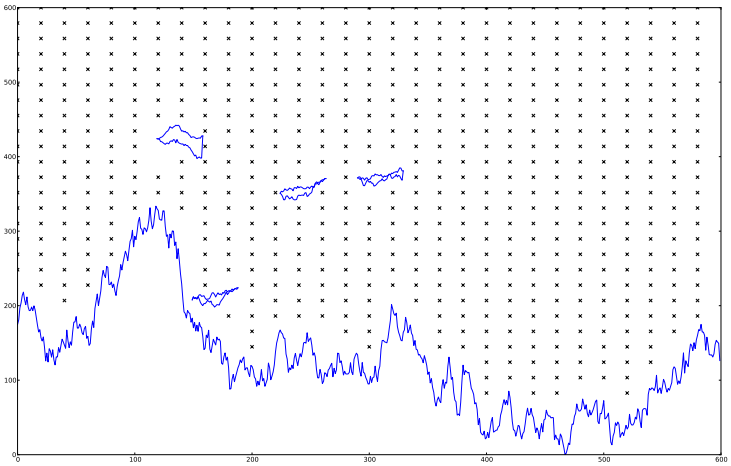


Figure 30: *The coastlines*

The generated coastlines are converted to the actual Map, which is a number of surfaces: Figure 31.

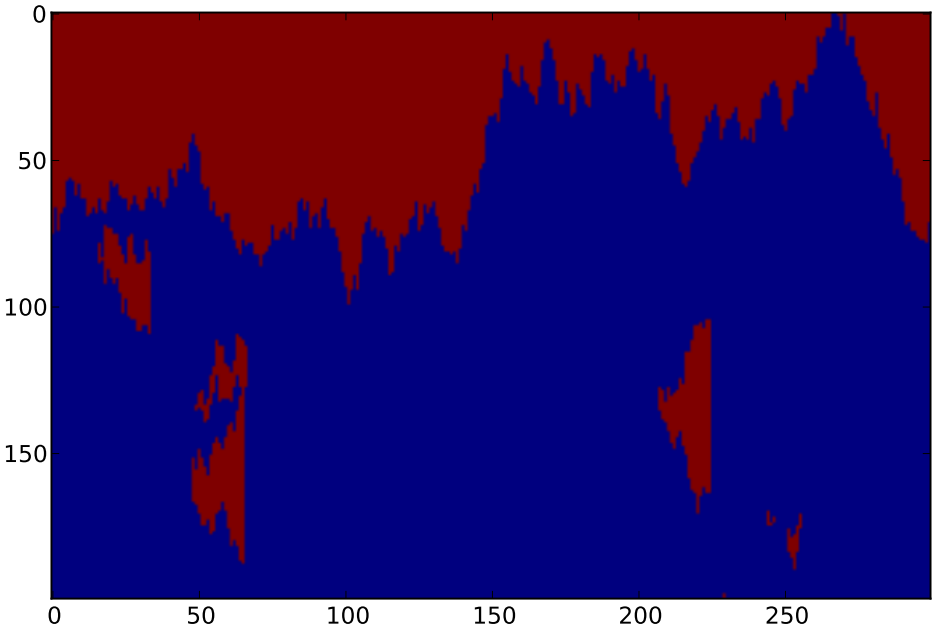


Figure 31: *The solidified map (the picture shows a north ("upside down") coast)*

Reading a solid map is possible from image files as well, using the Python SciPy library

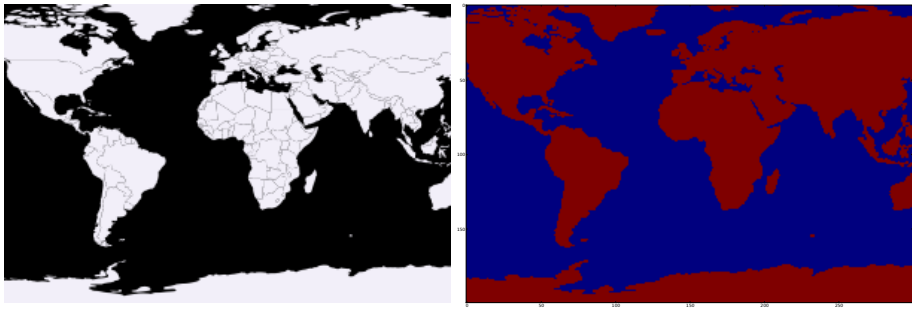


Figure 32: *Image file processed to Map object*

Routing

Upon entering the water, the first task of the vessel in automatic or autonomus mode is to determine the course. Currently the software can not handle rivers, strong wind and magnetic declination, therefore the course is assumed to be identical to the heading.

The Waypoint planner is responsible for the generating and ordering of the key measurement points. Visiting a set of waypoints on a given map leads to the Traveling Salesman NP-complete problem. Even using dynamic programming, determining the best route with the Held-Karp algorithm the program requires $O(N^2 2^2)$ steps [wiki]. The problem has been in the crosshair of mathematics for a long time, but a fast exact solution has never been found. The waypoint-planning is not time-critical, but a sufficient length of path should be calculated in a reasonable amount of time.

Fortunately the arrangement of the measurement points is relatively dense and predictable, allows only neighboring travels and repeated visits. In order to reach a suitable algorithm, some heuristics of the path planning needs to be examined.

Code: Ship.py
https://www.dropbox.com/s/fmtsaatql7jqjhw/Ship_nofilter.py

Ship.py implements the Ship Object. Everything related to the handling of a ship is encapsulated in a Ship instance. Multiple instances can be created based on the same object, with different parameters, and they are using the same resources.

In a simple coastline the points can be placed in a certain order relative to the coast, to achieve a sufficient result. If the shore is relatively smooth, the ship can also travel parallel to the coast, to decrease the required number of turns.

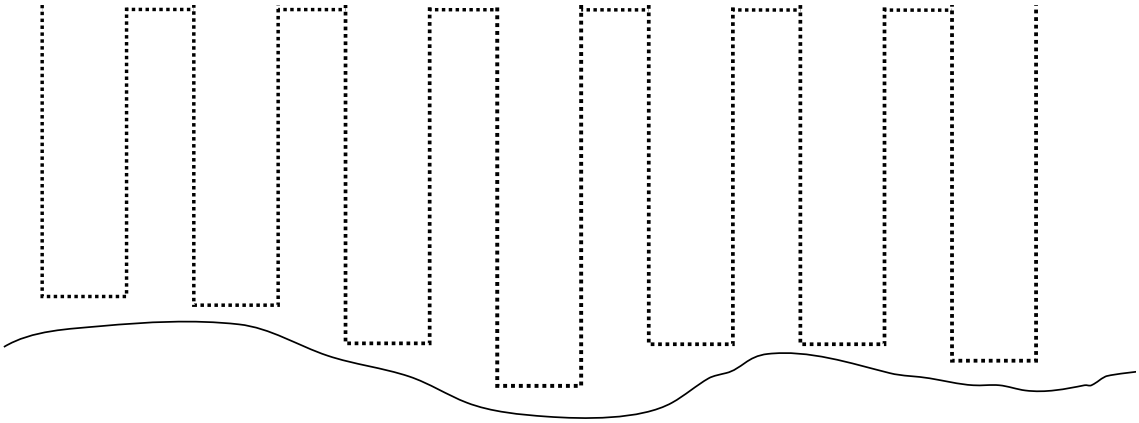


Figure 33: *A simple coast and a sufficient path*

Introducing isles and deep bays to the area complicate the situation. Some waypoints need to be removed, and an avoiding path needs to be taken. This will cause many redundant measurements and very sub-optimal path, if the island density and the complexity of the coast is high.

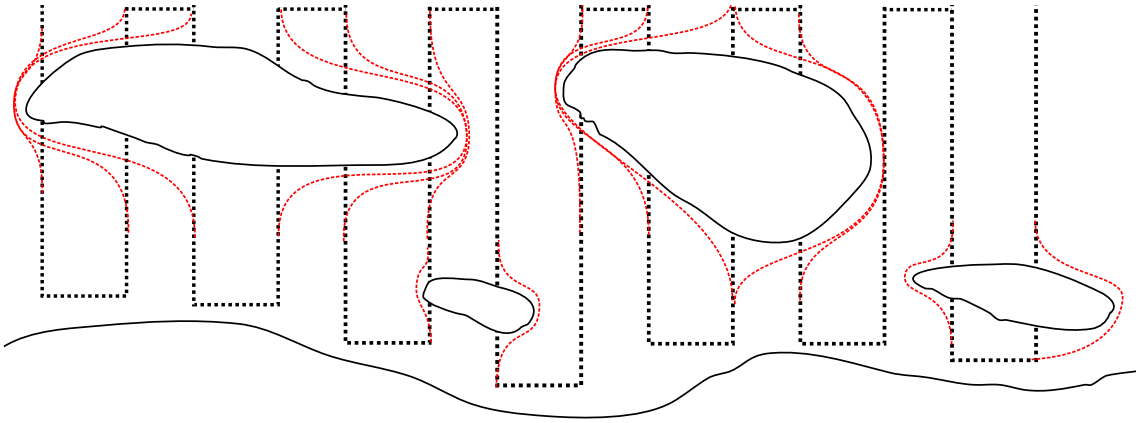


Figure 34: *Coast with islands*

Greedy traversal stretches a measurement grid over the area, and the points are traversed in an unpredictable way, using a neirest neighbourgh algorithm.

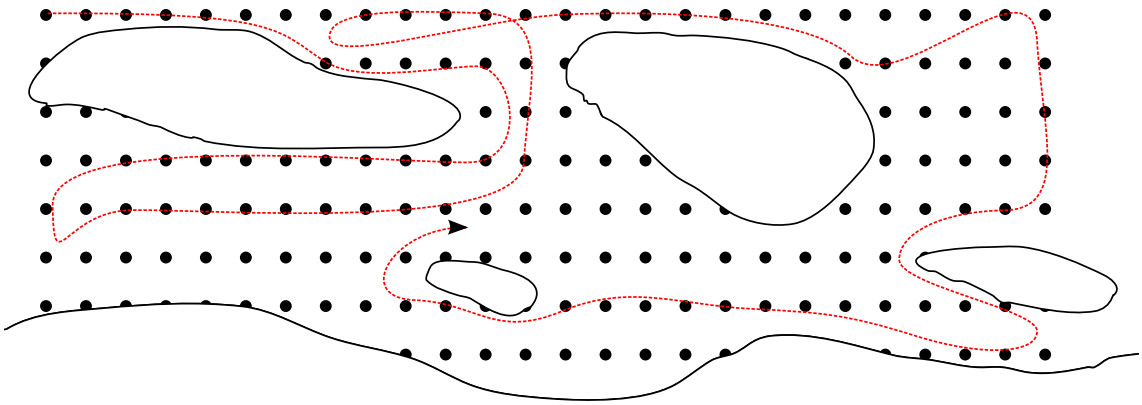


Figure 35: *A random path based on graph traversal*

According to [citation] heuristic closest neighbour search is usually 5-10% worse than the ideal solution. Using the closest neighbour heuristic, the simulation leads to the following path: Figure 36.

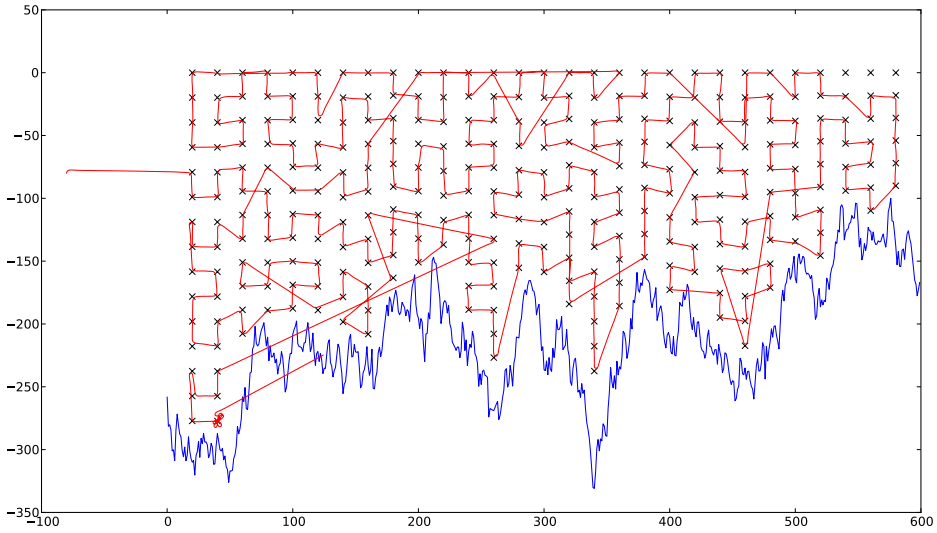


Figure 36: *Neirest neighbourgh algorithm*

Lowest cost heuristics

Even without major nautical engineering skills the following presumption can be made: If there are two paths with identical start and finish and identical lengths (start != finish) the path with less turning is more effective. This presumption can be extended to the following theory:

The most effective nautical path between two given points is the one with the least sum cost. The sum cost is the traveled distance times distance cost, plus absolute turnining times turning cost.

$$\Sigma C = dist * C_{dist} + |turn| * C_{turn} \tag{35}$$

The cost of the distance and the turning depends on the type of the ship. A large, deep draught cargo ship capable of low speed will have a much higher $\frac{Distance}{Turn}$ cost ratio than a narrow military cruiser.

Using the considerations above a cost based nearest neighbour algorithm is introduced, where the lowest distance is replaced with lowest cost. The algorithm checks every point to determine which is the cheapest destination. To avoid path-loops in the map, the waypoints already visited are stored in a list. If the examined point is already in the list, the cost is increased with a redundancy value, so the algorithm will chose a different, slightly more expensive, but still unvisited measurement point. The algorithm also checks, if the path leading to the examined waypoint is clear of obsticles.

Running the simulation with the above pathplanner results in the following course: Figure 37.

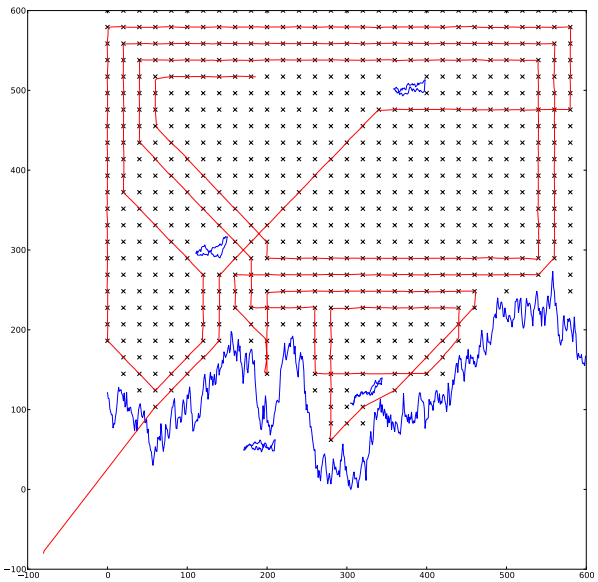


Figure 37: *Lowest cost algorithm*

This navigation method works adequatly in open environments. However, if the map is consisted of tight narrows and hard to navigate areas, the algorithm above fails to exit certain areas of the map. To be able to operate in such shores, a different kind of routing is required.

I call this method "Ripple", because the points examined are moving away from the surface vessel in a circular shape (well, in a rectangular actually) like a ripple. The ripple checks every neighbouroing waypoint, that wasnt checked so far, and stores the previous life of the ripple. If a point is found that has not been visited before, a list of waypoints are returned, and the ship will sail through them, and reach the destination.

Figure 38 illustrates this navigation method by simulating a task on the map of the Earth.

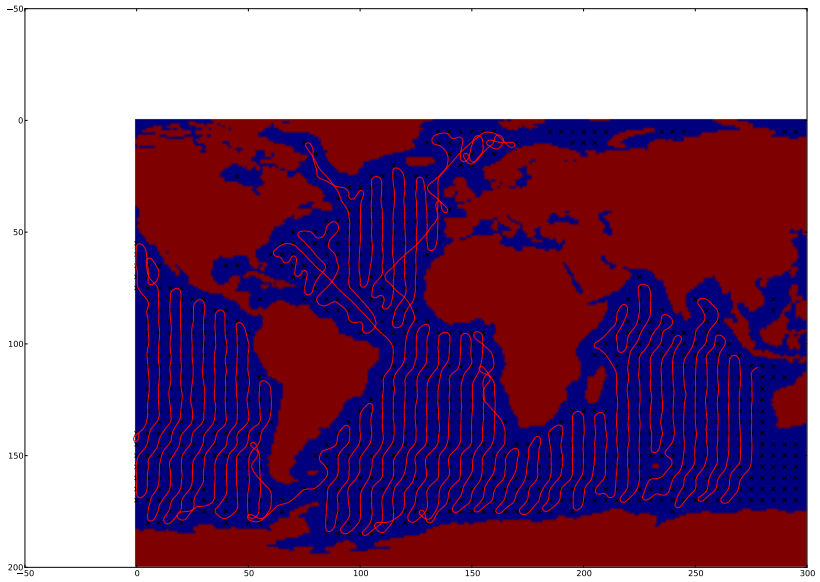


Figure 38: *Ripple routing algorithm*

Navigation

Once the required course has been set, the control system will take over the handling of the ship. The Navigation is the first and highest layer of control in the High Level Controller. In order to explain the navigation, the coordinate systems must be defined first.

Frames of reference

The operator team and the GPS sensor are using the Earth Centered Earth Fixed (ECEF) coordinate systems. Navigation on a spherical surface would introduce a lot of unnecessary calculations in every control cycle, since a plane can be fit onto the surface with minimal error, because the small vessel has limited maximum range.

During the Navigation the North East Down coordinate system is used to determine the attitude and position of the ship. The NED coordinate system is centered to the first GPS measurement coordinate.

A third Body Frame is also defined, the dynamic system of the ship was calculated in this frame.

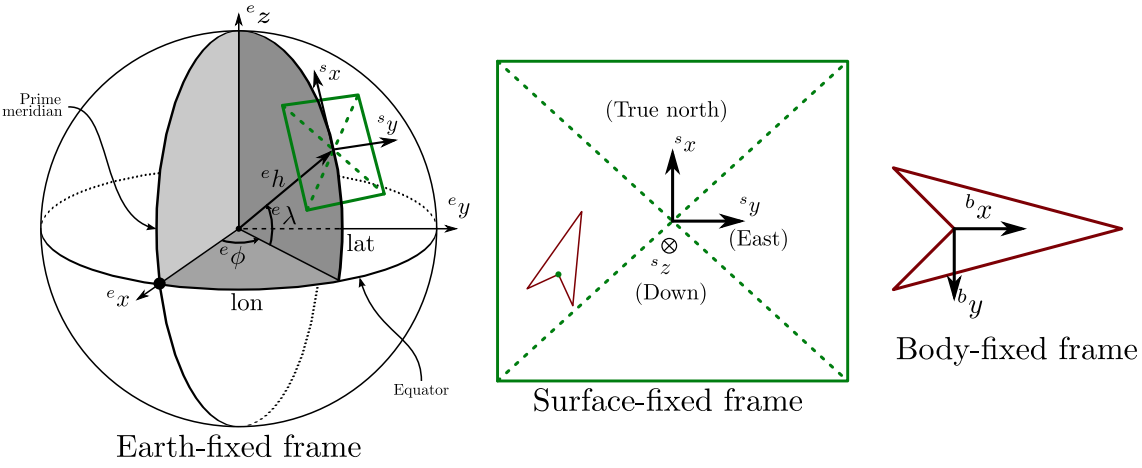


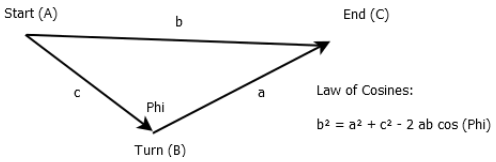
Figure 39: *Used coordinate systems*

$$\lambda = longitude \quad \phi = latitude \tag{36}$$

$$GPS_{[\lambda,\phi,h]} \Longleftrightarrow [X_{NED}; Y_{NED}; Z_{NED}] \tag{37}$$

Required heading

The heading of the ship is defined in NED coordinate system. The required heading is determined by the Law of Cosines, based on the Position of the Ship and the Position of the next Sub-Waypoint.



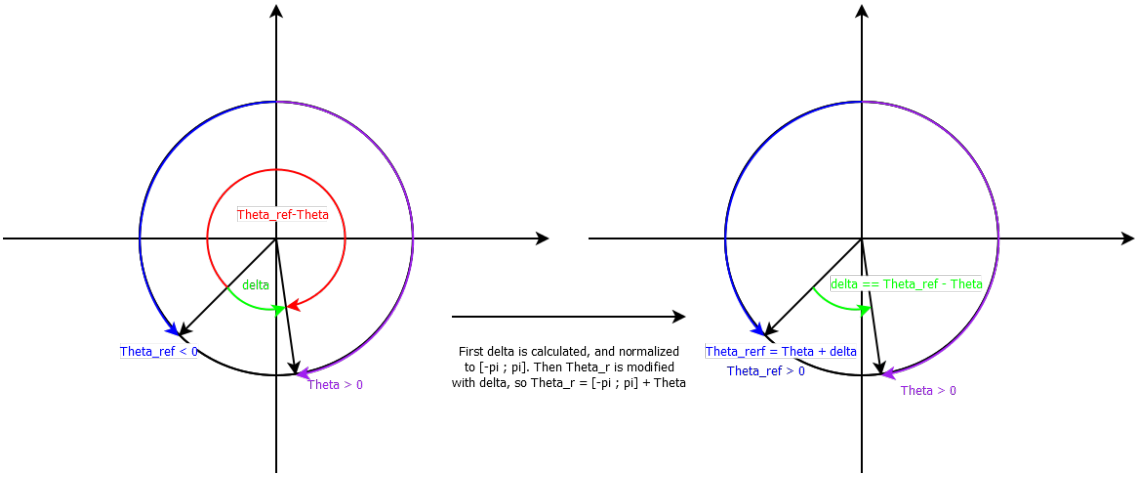
Problems rise and corrections are necessary, if the heading of the ship θ is $\theta < -\pi$ or $\theta < \pi$. The heading of the ship is calculated based on the Gyro sensor and the heading can have any value in the form of:

$$\theta = [-\pi; \pi] \pm 2 \cdot k \cdot \pi \tag{38}$$

Code: FunctionLibrary.py
https://www.dropbox.com/s/7nx7helisvss21a/FunctionLibrary.py

Some functions required by the software are general parametric mathematical functions, which can not be connected to any object. These functions, like the Law of Cosines, and the distance calculator function can be found in the Function Library.

Before invoking the control procedure, all of the heading angles must be transformed into the $[-\pi;\pi]$ interval. This procedure causes a possible error though.



The required heading or the heading of the ship must be transformed into a different representation, where

$$|\theta_r - \theta| < \pi \tag{39}$$

To keep a consistent heading representation, first the deviation angle

$$\theta = \pi_r - \pi \tag{40}$$

is calculated, than transformed to the $[-\pi;\pi]$ interval and finally, with δ we can transform θ_r to

$$\theta_r(\theta) = \theta + \delta \tag{41}$$

If the conditions above are met, θ and $\theta_r(\theta)$ will always yield values that result in correct controller output.

Waypoint planning

Waypoint ordering

Comparsion of methods (length/WP)

Map reading

Swarm intelligence

Dynamic pathplanning, time-scaling

Cooperative strategies

Formations

Measurements and results

Testing the navigation algorithm the boat is difficult in the marine environment. Exploiting the model-based design of the system enables us to validate the navigation software using a different measurement platform. The vehicle control layer of the boat can be substituted for a car's. Using the predefined input and output interfaces and replacing some auxiliary elements of the electronics, an Unmanned Ground Vehicle can be prepared for the task.

[pictures]

Navigation of an Unmanned Ground Vehicle

Navigation of a motorized USV in still water

Navigation of a sailing USV in still water

Navigation of a motorized USV in a river

Navigation of a sailing USV in a river

Conclusion

This document is meant to succeeds my previous work[11]. Several major functionalities have been added, and the engine behind under the hood has been changed, and will keep changing.

My most important observation during the Project Laboratory I. was that the perceived significance of the system components might differ from the reality. The successful implementation of the different pathplanning systems and navigation required a very solid and relatively simple Map object, that can be relied on.

With the major components of the HLC now complete, up and running, it's possible to move to the next phase. In Project Laboratory II. the LLC will be designed, the communication protocol between the objects and the ship itself, in order to accomplish a more-or-less working prototype by the end of 2013.

Acknowledgment

Don't forget to say thanks list
Patrick Bray - friendly and helpful guy who allowed the use of his stability drawings.

References

[1] W. Weinrebe, A. Kuijpers, I. Klaucke, and M. Fink. Multibeam bathymetry surveys in fjords and coastal areas of west-greenland. *AGU Fall Meeting Abstracts*, page A1152, Dec 2009. Provided by the SAO/NASA Astrophysics Data System.

[2] V Brand. Submersibles - manned and unmanned. *South Pacific Underwater Medicine Society Journal 7 (3)*, 1977. issn: 0813-1988.

[3] The navy unmanned surface vehicle (usv) master plan. Technical report, Department of the Navy, United States of America, Jul 2007.

[4] Dr. Vajda Ferenc. Mobilis robotok navigációs kérdései. Technical report, Digitális célrendszerek kutatócsoport, Budapesti Műszaki és Gazdaságtudományi Egyetem, Sep 2013.

[5] Robert H. Stewart. *Introduction To Physical Oceanography*. Department of Oceanography, Texas A & M University, 2008.

[6] J.M.J. Journée and Jakob Pinkster. Introduction in ship hydromechanics. Technical report, Delft University of Technology, Apr 2002.

[7] *Future Ship Powering Options: Exploring alternative methods of ship propulsion*. Royal Academy of Engineering, United Kingdom, 2013.

[8] Tóth Kálmán. *Vitorlázás*. Műszaki Könyvkiadó, 1970.

[9] Patrick J. Bray. *Stability – What is it and how does it work?* Bray Yacht Design and Research LTD.

[10] David Leonardo Lencastre Sicuro. Centralized state estimation of distributed maritime autonomous surface oceanographers. Semester project, Section for Control and Automation, Department of Electronic Systems, Aalborg University, Denmark, 2013.

[11] Rasmus L. Christensen, Frederik Juul, Nick Østergaard, Tudor Muresan, and Attila Fodor. Centralized state estimation of distributed maritime autonomous surface oceanographers. Semester project, Section for Control and Automation, Department of Electronic Systems, Aalborg University, Denmark, 2013.

Appendices

A TeXnicCenter felülete

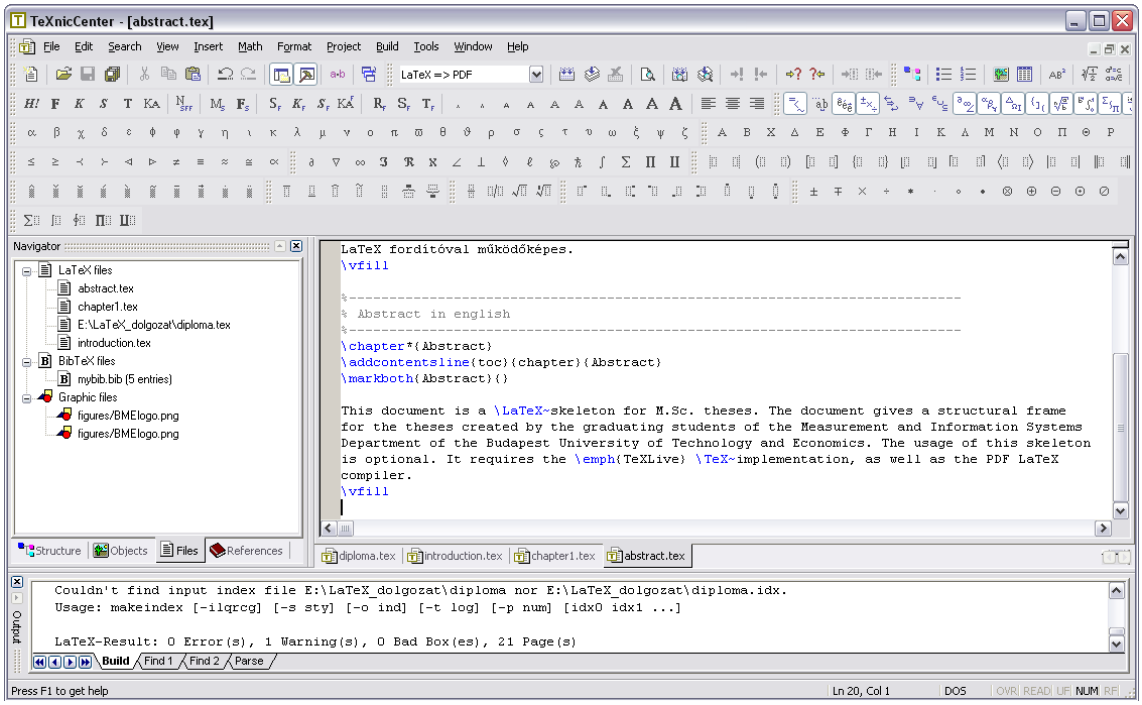


Figure F.0.40: *A TeXnicCenter Windows alapú LaTeX-szerkesztő.*

Válasz az „Élet, a világmindenség, meg minden” kérdésére

A Pitagorasz-tételből levezetve

$$c^2 = a^2 + b^2 = 42. \tag{F.0.1}$$

A Faraday-indukciós törvényből levezetve

$$\mathrm{rot}\, E = -\frac{dB}{dt} \qquad \longrightarrow \qquad U_i = \oint_{\mathbf{L}} \mathbf{Edl} = -\frac{d}{dt} \int_{A} \mathbf{Bda} = 42. \tag{F.0.2}$$