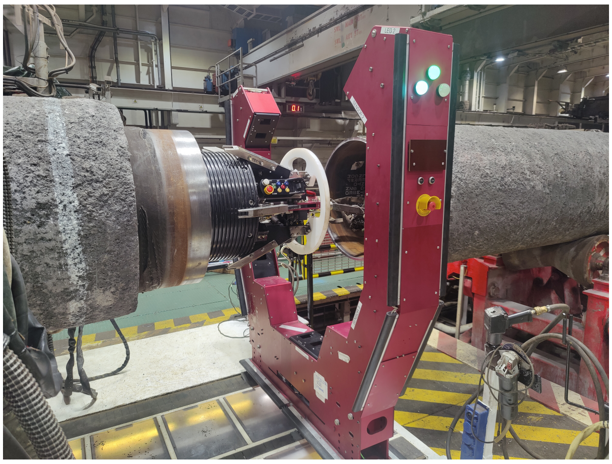
WI5118: AM Internship (cohort 2023)

Line Up Automation

Philip Soliman (4945255)

November 25, 2024



University supervisor:
Dr. Ir. L. Rens
Delft University of Technology
Faculty of Electrical Engineering, Mathematics and
Computer Science

Company supervisor:
J. Ramlakhan
Senior R&D Engineer
Research & Development

Contents

1 Company description

Allseas Engineering B.V. is a Dutch company that specializes in offshore pipeline installation and subsea construction. The company was founded in 1985 by Edward Heerema and is headquartered in Delft, the Netherlands. Allseas is known for its innovative engineering solutions and state-of-the-art vessels, such as the Pioneering Spirit, which is the largest construction vessel in the world. The company has a global presence with offices in the Netherlands, Switzerland, and the United Kingdom, and employs over 2,500 people worldwide.

Allseas' core business is the installation of subsea pipelines and the construction of offshore structures for the oil and gas industry. The company's vessels are equipped with advanced technology and equipment to perform a wide range of offshore operations. [1]

Allseas' organizational structure is divided into several departments, which are shown in Figure 1.

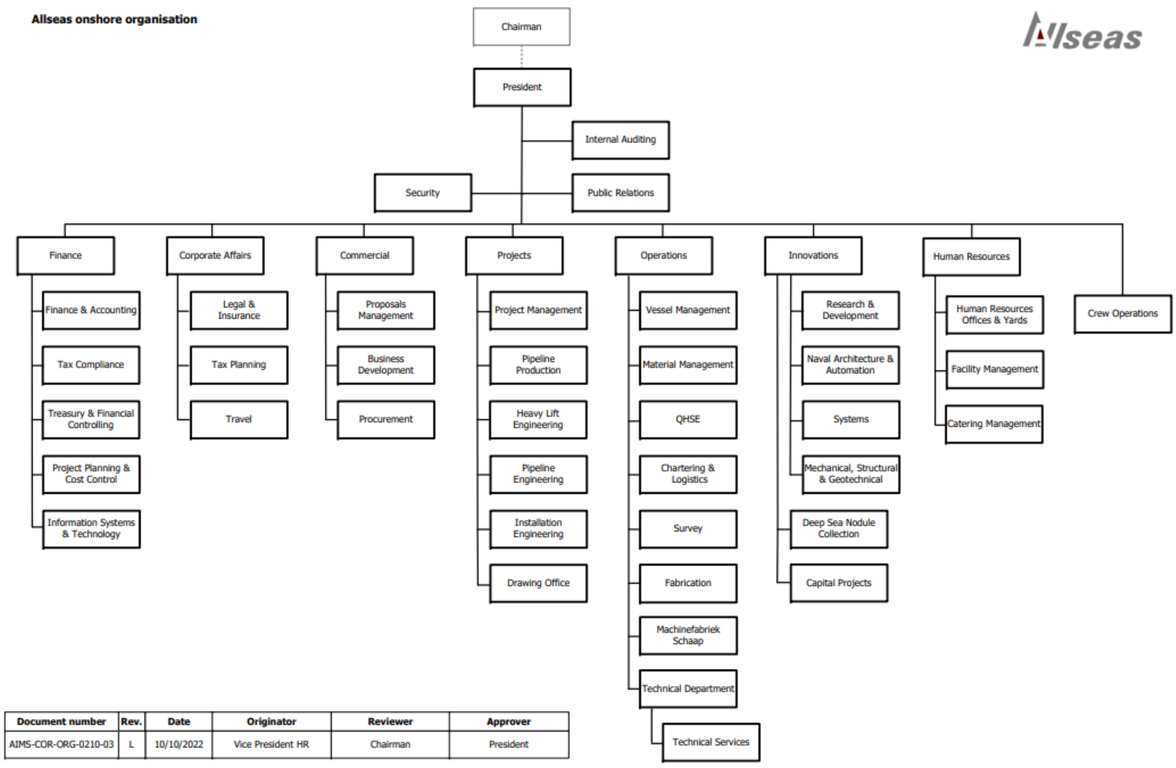


Figure 1: Allseas organizational structure.

The company's innovations department is responsible for the design and development of new technologies and solutions to improve the efficiency and safety of offshore operations. It is in this department that the internship project is situated. Figure 2 shows the organizational structure of the innovations department. In particular, the project is performed in the sub-department of Research & Development Delft, which is led by the department manager (Dominik Bujakiewicz). The project is supervised by the senior R&D engineer (J. Ramlakhan) and performed by the intern (Philip Soliman).

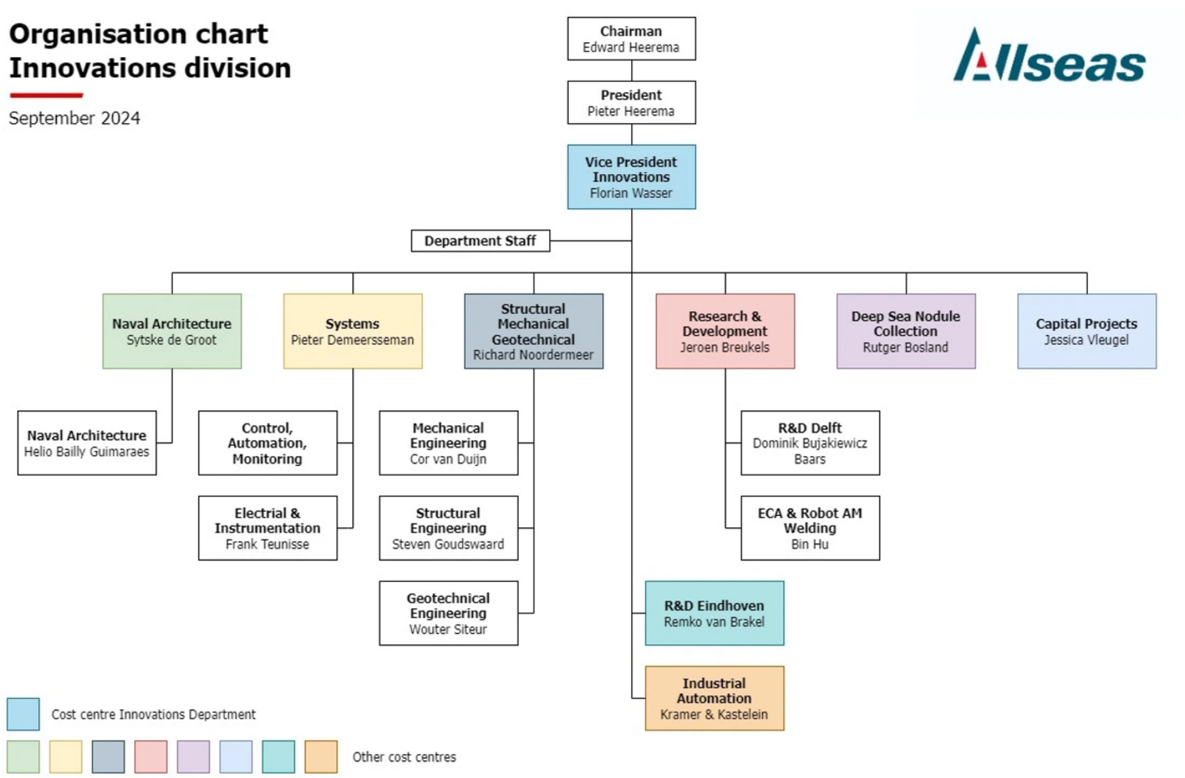


Figure 2: Allseas innovations department organizational structure.

2 Problem description

During pipelay every few minutes a new pipe (new joint) section must be fitted onto the end of the existing pipeline (main line). For decades in the offshore industry this has been a manual process. Allseas is working on a method to automate this.

2.1 Main requirements for automated line-up

Several projects have been initiated to automate the line-up process starting as early as 2008. However, These have all been unsuccessful. The latest project was started in 2017 and is still ongoing. The main requirements are:

1. Reducing line-up cycle times in the beadstall; This will result in significant cost reduction by decreasing vessel production time.
2. Increasing accuracy of line-ups (reduce amount of root pass weld errors); This will also reduce costs due to decreasing vessel production time.
3. Providing a safer work environment for the people working in the beadstall.

2.2 Current situation

The latest iteration of automated line-up is a system that uses 5 laser line scanners to measure the pipe ends' position and orientation (see figure 3).

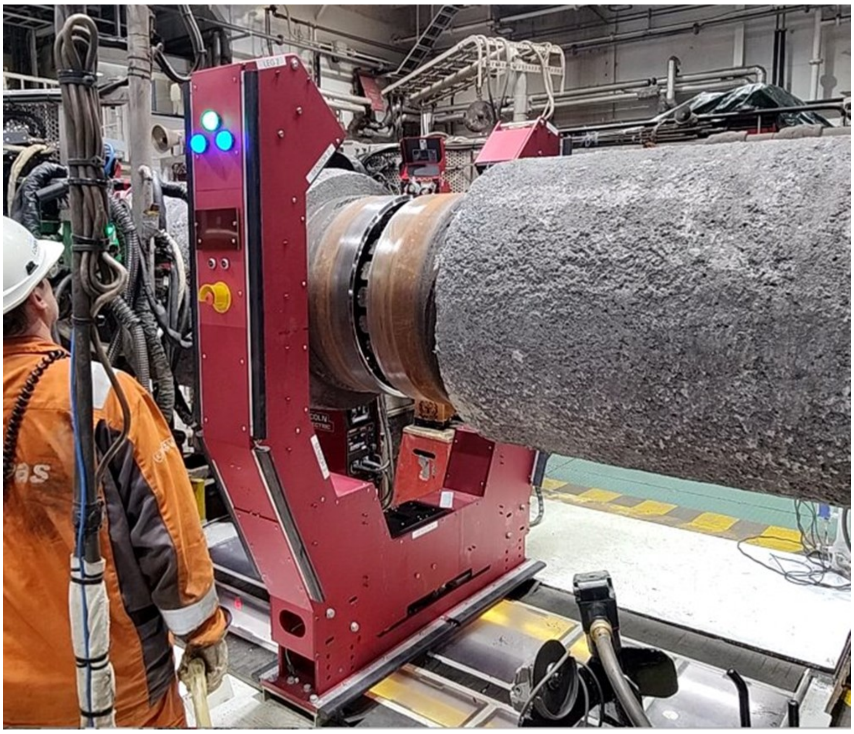


Figure 3: The portal or 'U-Frame'. The mainline is held in place in the background, while the new joint is lined up using instructions from automated line-up. The white stickers on the side of the frame indicate the position of the laser line scanners. Three on the side from which the picture is taken and two on the other side of the pipe. The U-Frame is mounted on the interface frame, which in turn is connected to the subframe. The subframe has motorized rollers and linear guides that allow the U-Frame to move with the new joint. This way the U-Frame ensures optimal visibility of the pipe ends for the laser line scanners. Additionally, the U-Frame can be moved out of way of other equipment, personel or operations when not in use.

The laser line scanners generate 4192 data points in 5 different planes with a frequency of $60 \sim 70$ Hz. Initially, a line finding algorithm derived from a Hough transform [3] is used to recover approximately straight sections from the laser's projection on the pipe. This allows for the recognition of the pipe ends or 'bevels' from the data.

Subsequently, the found pipe ends of the 5 sensors are combined to form a 3D representation of the pipe face. This is done using using a combination of Newton-Raphson optimization and a least squares fit.

2.3 Internship goals

The internship aims to improve the current automated line-up system by:

- **a** **Decreasing code execution time** by searching for and replacing slow or redundant operations;
- **b** **Writing maintainable code** by refactoring the current codebase;
- **c** **Increasing machine accuracy** by comparing different pipe fitting algorithms.

3 Approach

In this section the approach to the problem is described. The approach is divided into several steps. The steps are ordered in a way that makes sense for the development of the project. The steps are as follows:

3.1 Understand automated line-up

Before any improvements can be made to the system, it is important to understand how the system works. This includes understanding the hardware, the software and the mathematical model that is used to fit the pipe ends. Next to that, getting to know the engineers that work on the project and the test set-up at the yard is also important. All of this provides a wholisitc view of the system, which in turn highlights the signifigance of the improvements described in Section 2.3.

📖 section sections 4.1 to 4.3

3.2 Familiarize with the code base

Read the code starting form the lowest level functions reading in the data received rom the laser line scanners to the highest level functions that combine the processed data to fit the pipe ends.

📖 section 4.4

3.3 Create a profiler for the code

The best way to improve code speed is to know where the bottlenecks are. A profiler will help to identify these bottlenecks. Focusing on the most time consuming parts of the code prevents wasting resources on speeding up parts of the code that are not used often.

✅ internship goal 2.3 **a**
📖 section 5

3.4 Optimize the code

Once the major bottlenecks are identified, the next step is to find more efficient ways to perform the operations in question. Adjustments may involve reducing memory allocation, re-ordering operations, correcting or updating algorithms, implenting precomputation of constant values or optimising data locality.

✅ internship goal 2.3 **a**
📖 section 6

3.5 Identify unreadable code

Halfway during the development between 2017 and the time of this internship a new standard for clean, readable code has been introduced within Allseas. Parts of the old code have also been improved, but quite some code still needs a bit of restructuring. This step serves to identify the pieces of code that are least readable and need to be updated to the new Allseas coding standard.

✅ internship goal 2.3 **b**
📖 sections 6

3.6 Determine and improve upon current accuracy

Run new fitting algorithms on generated data to test the accuracy of the new algorithms. This will help to determine if the new algorithms are more accurate than the old ones. Do the same with recorded data from the test set-up to see if the new algorithms are also more reliable. Finally, test the algorithms on the real-time data from the test set-up to see if the new algorithms are also faster.

✅ internship goal 2.3 **c**
📖 section 7

3.7 Expand the fitting model to recognise ILUC

During line up of the new join with the main line, there is more equipment present in and around the latter. One major component present in the main line is the 'Internal Line Up Clamp' (ILUC) and is shown in Figure .

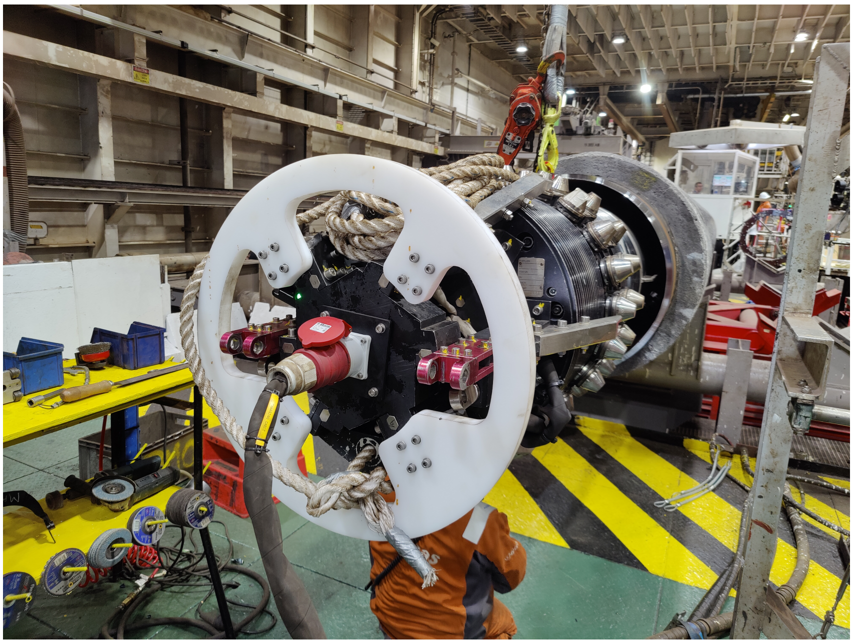


Figure 4: The Internal Line Up Clamp (ILUC). The ILUC sits in the main line, while the new join is shifted over it during line up. Its purpose is to steady the main line and new joint pipe ends after line up and during welding. Using a set of pneumatic knobs portruding from its exterior it pushes both pipe ends outward, ensuring a circular fit. Part of the ILUC is a black ribbed cylinder.

Recognising the ILUC gives a good indication of where the main line is before it can be measured, seeing as the U-Frame passes over the ILUC before it reaches the main line. Early estimation of the positon and orientation of the main line benefits the overall speed of the process, because less corrections have to be made in the final line up stage.

Therefore, the final step of the internship is the development of a new part of the recognition software finds the a specific part of the ILUC; its ribbed cylinder. As before, accuracy and reliability can be tested using the test set-up in the yard. Furthermore, a combination of different kinds of line, plane and circle fitting algorithms based on minimization of geometric distances can be used for this. Lastly, Where pipe ends tend to have deformations due to manufacturing, pre-heating or bevelling, the ILUC is a near-perfect cylinder with a defined radius. This can be used to improve the accuracy of the fitting algorithms.

✅ internship goal 2.3 **c**
📖 section 8

4 Understanding line up automation

This section describes basic concepts of the line up automation (LUA) system. The automated line-up system is a system that aligns two pipes together. The system consists of a U-Frame, laser line scanners, Line up Car Assemblies (LCAs) and a sub-frame. The U-Frame houses the laser line scanners and is mounted on the sub-frame. The sub-frame is responsible for the U-Frame's movements. The LCAs are devices that control the new joint's position and orientation. The laser line scanners are sensors that detect the new joint's position.

4.1 The U-frame

Figure 5 shows the U-Frame. The U-frame is a part of the automated line-up system. It is precisely machined to ensure that the positions of the laser line scanners are known as accurately as possible.

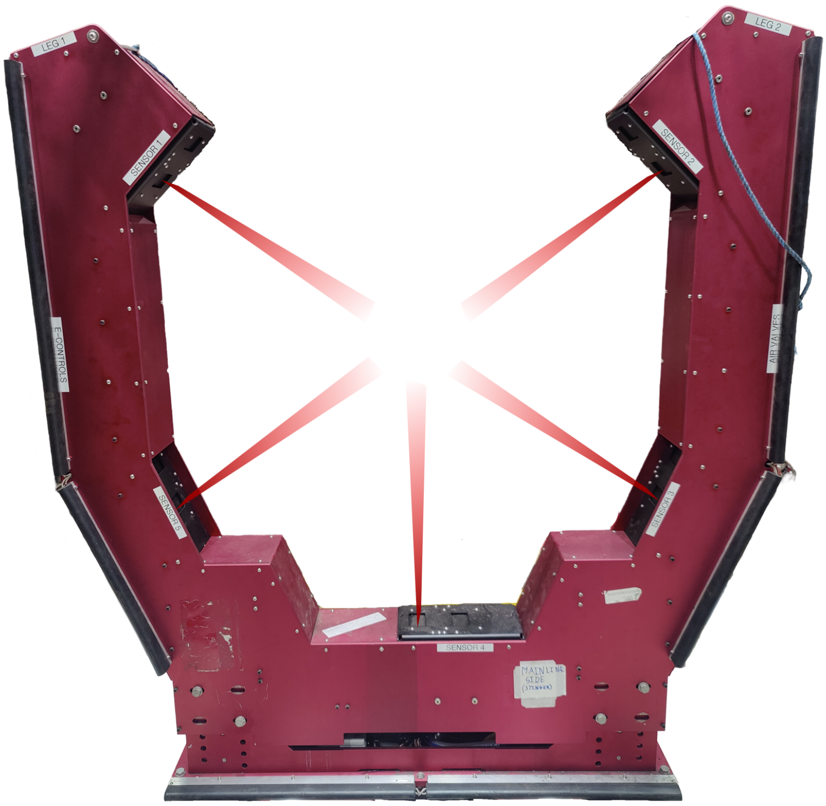


Figure 5: The U-Frame. Most importantly, the U-frame houses five laser line scanners that are used to detect pipes and other objects like the ILUC. The U-Frame is mounted on an interface frame, which in turn is mounted on the sub-frame (not visibile here). The sub-frame is responsible for the U-Frame's movements. To do ►*change laser light to blue*◄

4.2 Laser line scanners

The laserline scanners measure in 2 dimensions, width (x-direction) and depth (z-direction). The scanners each generate 4192 data points in any one measurement. The frequency of the measurements in only limited by how fast the laser line scanner runs (see section 4.4). The maximum scanning frequency of the lasers lies anywhere between 600 and 9000 Hz [4].

An important aspect of the laser line scanners is their Field of View (FOV). The FOV is the area in which the laser line scanners can detect objects. Figure 6 gives the FOV of the laser line scanners.

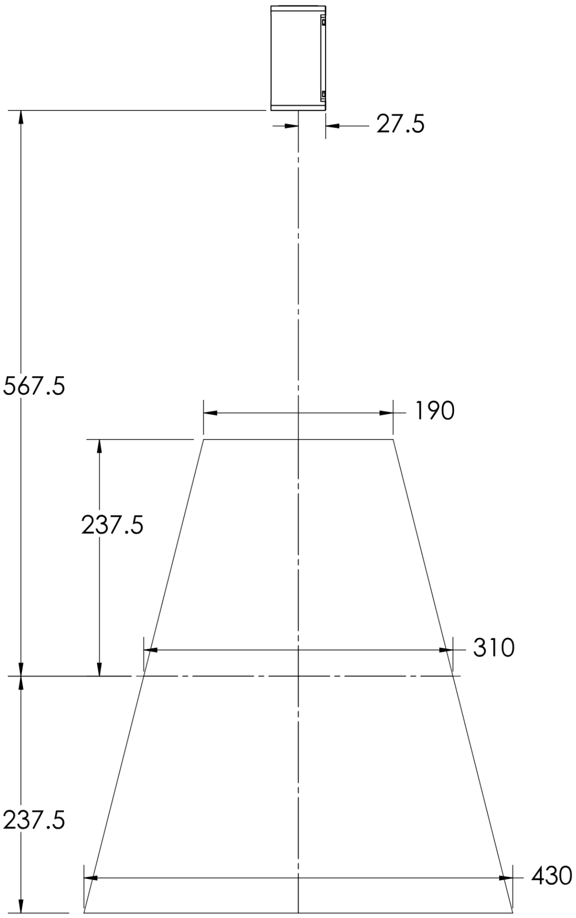


Figure 6: Field of View (FOV) of the laser line scanners. Measurements are in millimeters [5].

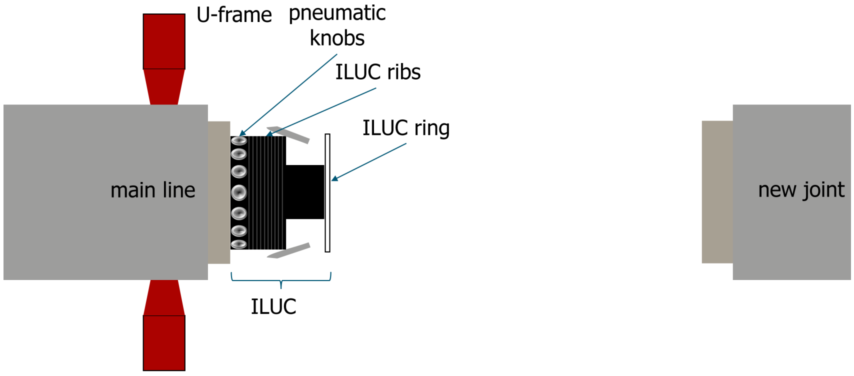
From figure 6 it can be seen that the FOV of the laser line scanners is limited. In particular, the width of the frame itself is $\sim$ 20 cm, which means the laser line scanners can see $\sim$ 10 cm beyond either side of the frame. This is why it is necessary for the U-Frame to move.

Depending on the distance of an object to a laser line scanner, the accuracy its measurement can vary. For an object close to the laser line scanners the resolution is $47\mu m$ in the x-direction. For an object further away from the laser line scanners the resolution is $104\mu m$ in the x-direction [4].

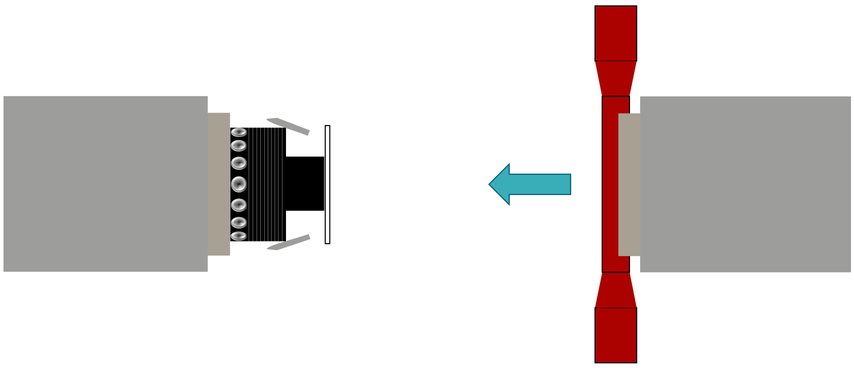
In terms of the z-direction, an indication of accuracy is given by the repeatability. This value is determined by using a flat target and calculating the 95% confidence variation of the average height over 4096 measurement frames. z values are averaged over the full FOV. The repeatability comes out to be $2\mu m$ [4].

4.3 Order of operations automated line-up

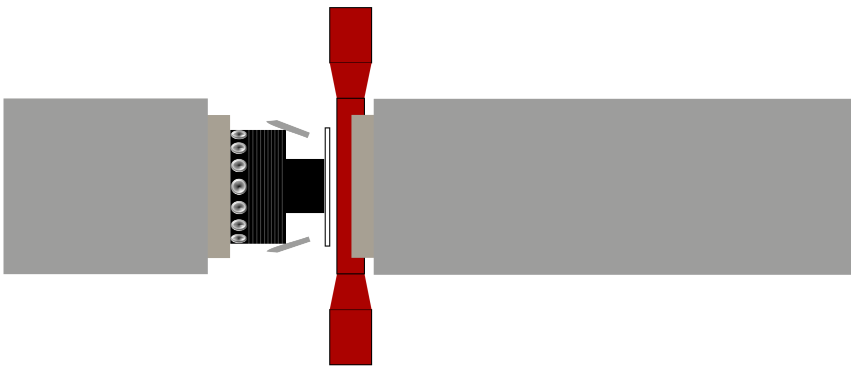
Figure 7 shows the order of operations during automated line-up.



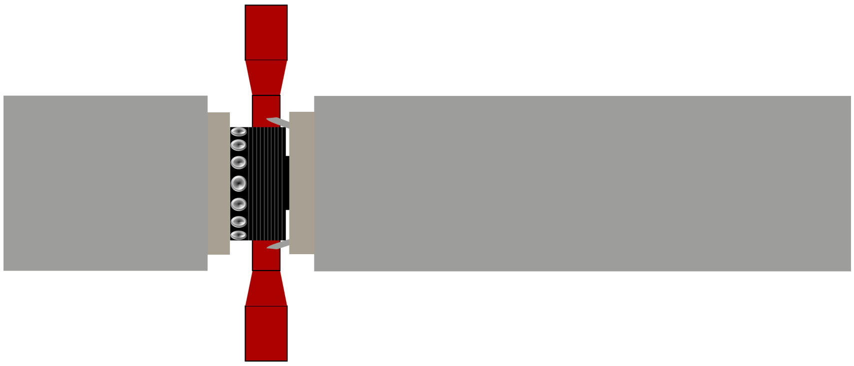
(a) Initial state of automated line up. The main line is depicted on the left, while the new joint is depicted on the right. The U-Frame is in its starting position. The ILUC sits in the main line. Visible parts of the ILUC are its pneumatic knobs, black ribbed cylinder (ILUC ribs) and white ring (ILUC ring).



(b) The U-Frame moves to the right until it encounters the new joint, which is moved towards the main line. The U-Frame reverses direction and tracks the new joint upon its recognition by the laser line scanners.



(c) The laser line scanners detect the ILUC ring. The first adjustments are made to the new joint's position. This is done to ensure that the new joint is shifted over the ILUC ring and other parts of the ILUC.



(d) The laser line scanners detect the ILUC ribs. This can be used to further adjust the new joint's position, matching more closely the orientation of the main line.

(e) Both pipe ends are fitted together during final line up. Adjustments are made to the new joint's position until a certain minimum gap tolerance is reached.

(f) U-Frame moves back to its starting position. The new joint is now fitted to the main line. The pipelines are now ready for welding.

Figure 7: Simplified order of operations during automated line-up. This procedure is repeated for each new joint that is added to the main line. After final line up in 7e is completed, the new joint is welded to the main line. Then, the whole pipeline is moved forward and the process is repeated for the next new joint.

Figure 7 offers only simplified overview of the automated line-up process. Not shown in the figure are the;

- **sub-frame**: system that controls the U-Frame's movements;
- **laser line scanners**: sensors that detect the new joint's position;
- **Line up Car Assemblies (LCAs)**: devices that precisely control the new joint's position and orientation;

4.4 Flow of line up automation software

The code flow of the laser line scanner software is shown in figure 8. This is an high-level overview of the software's operation.

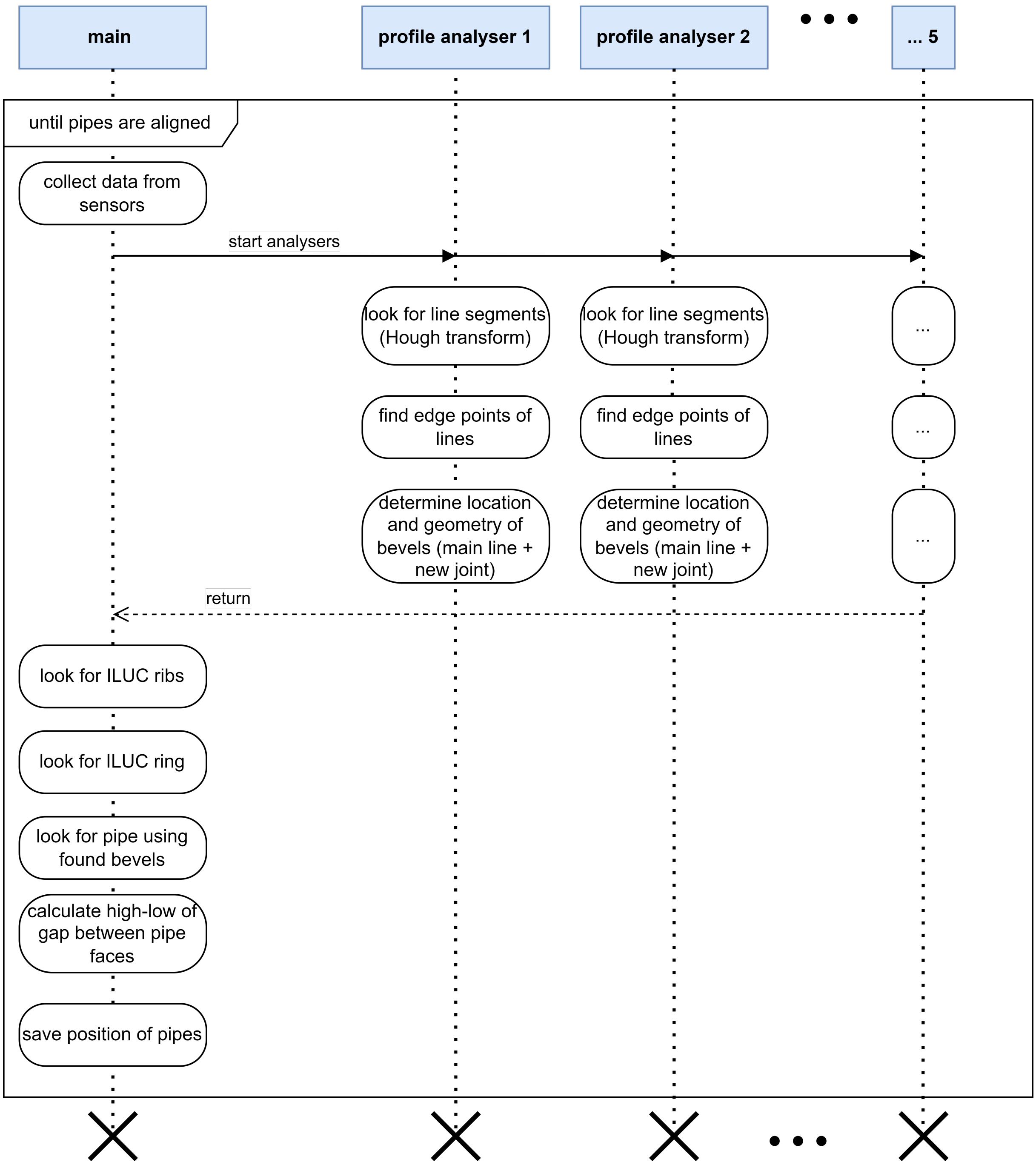


Figure 8: Flow of the laser line scanner software. The software is a multithreaded application. The blue boxes with bold text indicate the birth of these threads. The dotted lines are the 'life-lines' of the threads, which always terminate in an '×'. The frame symbolises the main loop of the software. The top-left compartment in the frame specifies the loop-condition; 'until pipes are aligned'. The rounded boxes on the life-lines indicate functions that are called. Note that only the main thread can call functions. The main thread is responsible for the initialisation of the software and the creation of the other threads.

As can be seen in figure 8, the laser line scanner software assigns a seperate thread for the analysis of the data coming from each of the five laser line scanners. These analyser threads are responsible for finding line-segments, edge points and pipe bevels in their respective 2-dimensional data sets. Once this preliminary, disjoint analysis is done, the main thread combines the processed data into a 3-dimensional model of the ILUC ribs, -ring and/or pipes. Specifically, the models for the pipes are used to calculate the high-low values of the gap between the new joint and the main line. The high-low values determine if the pipes are lined-up correctly. If the gap is within a certain tolerance, the software will signal that the pipes are aligned and it terminates the main loop.

Comparing Figures 7 and 8, it becomes clear that the relative activity of the various parts of the software change depending on the stage of the automated line-up process. For example, determining the location and geometry of a pipe's bevel is only necessary as soon as the new joint comes into Field of View (FOV) of the laser line scanners (step 7b). Then, the activity roughly doubles as soon as the main line comes into FOV of the laser line scanners during final line up (step 7e). Furthermore, the finding of lines and their edges in each of the profile analyser threads becomes more computationally expensive for the ILUC ribs. This is further explored in section 5.

5 Profiler

Main change: Add debug and symbol flags at compilation of objects and linking of main executable

Reason: Adding these flags causes the (MSVC) compiler to generate symbol (for Windows .pdb) files, which are required by a (Microsoft) profiler.

Microsoft Visual Studio (Community >= 2019, probably earlier versions as well) come with an installation of VSDiagnostics; a command-line profiling tool. VSDiagnostics generates *.diagsession files, which VS interprets and visualizes. For this reason, it is in my opinion more convenient to call VSDiagnostics indirectly (from within VS). Below I describe the steps to do this.

We need to let VS know where it can find the symbol file corresponding to the csapp executable. To do this in VS go to Toolbar>Debug>Options>Debugging>Symbols and add the full path to "csapp.pdb" (without quotation marks). Note: it should be in the same folder as "csapp.exe"

Now we can start profiling: 1. Go to: Toolbar>Debug>Perfomance Analyzer 2. Set target to Executable 3. Locate "csapp.exe" (optionally provide some CL-args) 4. Check "CPU usage" (optionally change the sampling rate by clicking) 5. Check "start without collecting" to start csapp without immediately collecting profiling data 6. Resume/pause profiling by clicking the "Record" button 7. Stop profiling session by clicking the "Stop collecting" button

Some general info on the concept of profiling. Though this page is more focused on solutions developed in VS itself it may still be useful to checkout for a more detailed decsription of VS' profiling capabilities. More info on some the CPU usage stats that VS presents in its profiling reports. Lastly, Microsoft recommends to **not exceed a trace duration of 5 min**, as to limit analysis time.

Phase	Description	Duration	Experimental Setup (Office)
Joint recognition	LUA software detects joint as it moves towards mainline. U-Frame moves with joint after (successful) detection.	5 to 10 s	Flat surface resembling the joint (back of the ILUC-ribs mockup) is quickly brought into frame and subsequently kept still.
ILUC ring recognition	Joint and U-Frame move until ILUC ring is detected, after which partial line up takes place.	idem	Next to the joint mockup, a second flat surface is brought into frame representing the ILUC-ring.
ILUC ribs recognition	The ILUC-ring moves into the joint during the aforementioned partial line-up, after which the ILUC-ribs are detected. The joint is moved over the ribs towards the mainline.	idem	A joint mockup (a side of the ILUC-ring mockup) is still in frame. A ILUC-ribs mockup is slowly moved into frame, revealing all 9 of its ribs.
Final line up	The mainline gets detected and the joint is lined up with it.	idem	The joint mockup and a second flat surface (mainline mockup) are slowly moved together.

Table 1: Experimental Phases and Setups

Additionally, every profiling test (trace) is approximately 20 seconds in length, in keeping with the profiling guidelines. Every trace is preceded by a 20 second "settling period" during which the LUA software runs, but no profiling or setup movement takes place. The profiling sampling rate is >=1000 Hz.

Below we present the profiling results obtained during testing with the setup described. Shown are the "top functions" sorted by self-time, i.e. the functions that the CPU spends most time on, excluding time spent on subroutines/-functions.

We identify the following functions as bottlenecks: 1. “‘LineFinder::calculateNrPointsInEachBin“‘
2. “‘PointsIdentifier::gapToPreviousPointTooBig“‘ 4. pointer get operator: “‘->“‘ 3. “‘LineFinder::calculateNrVa

We add these functions to the above list of improvements.

Additionally, we also find that that the perfomance is not hindered by GOCator’s data acquisition time. That is to say, CPU resources spent on acquiring data from the GOCator system are negligable compared to our own source code. This improvement direction is subsquently closed.

6 Optimize & refactor code

To do ►*talk about precalculating sin and cosine values,* ◄

7 Accuracy of circle fits

7.1 Circle fit error estimation

[2] Hyper fit (hier "Hyper") is zelfs beter dan least squares (heet hier "geom" voor "geometric fit", omdat die de geometrische afstanden minimaliseert).

Hyper heeft dezelfde variantie (in de 1e orde) als least squares

$$E[(d_1Q)(d_1Q)^T] = s^2(W^TW)^(-1).$$

Hyper heeft daarentegen geen bias, waar least squares een bias heeft van $2 * s^2/R$. Beiden kunnen we afschatten we met waardes die we halen uit line up: hoeken van de lasers straal van de pijp standaard deviatie in van de noise in de laser data Variabelen & hun betekenis:

1. $Q = (a, b, r)^T$: de vector met parameters die we proberen af te schatten. middelpunt van de cirkel (a,b) en straal r
2. d_1Q: alleen de eerste orde termen in de Taylor expansie van Q
3. A, B, R: de "echte" waarde van de x/y-coordinaat en straal van de cirkel <- straal weten we in het geval van de ILUC en pijp (ongeveer dan)!!!
4. $U_i = \cos(\Phi_i) = (X_i - A)/R$: genormaliseerde waarde van de "echte" waarde X_i
5. $V_i = \sin(\Phi_i) = (Y_i - B)/R$: genormaliseerde waarde van de "echte" waarde Y_i
6. W: de genormaliseerde "echte" waarde matrix met $W_i = (U_i, V_i, 1)^T$
7. Φ_i: hoek die correspondeert met de "echte" coordinaat (X_i, Y_i) <– deze weten we; de hoeken van de lasers in het frame!!!
8. $x_i = X_i + dx_i$: de gemeten waarde van de x-coordinaat. Bestaat uit de "echte" waarde x_i en de fout/noise (zie hieronder)
9. $y_i = Y_i + dy_i$: idem, maar dan y
10. $dx_i, dy_i \sim N(0, s^2)$: fout/noise in de data. Normaal verdeeld met bias 0 en variantie s^2
11. s: standaard deviatie van de afwijking/noise in de data.

8 ILUC Recognition

9 Conclusion

Perform tests in the yard with the complete system, to test new code for points described in proposed approach: - Create a profiler for the code to search for slow operations. Milestone: profiler gives insight into performance of different parts of the code and allows comparison between different versions of said code. This way we can determine which is optimal - Find more efficient ways to perform the needed operations. Milestone: Increase overall code speed. Use profiler as quantification - Isolate parts of the mathematical model and test accuracy. Milestone: Mathematical as well as experimental estimation for line-up accuracy meets 0.1 mm requirement - Expand the fitting model to increase the workability of the complete system. Milestone: Successful and efficient recognition of ribbed cylinder allows for good pre-line-up of pipes bsectionProfiler

10 Skills learned

Bibliography

Allseas. (2024). *About allseas*. URL.

Al-sharadqah, A., & Chernov, N. (2009). Error analysis for circle fitting algorithms. *Electronic Journal of Statistics*, *3*. https://doi.org/10.1214/09-EJS419

Duda, R. O., & Hart, P. E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, *15*(1), 11.15. https://doi.org/10.1145/361237.361242

Technologies, L. (2024a). *Gocator 2650 smart 3d laser line profile sensor*. URL.

Technologies, L. (2024b). *Gocator 2650 smart 3d laser line profile sensor*. URL.