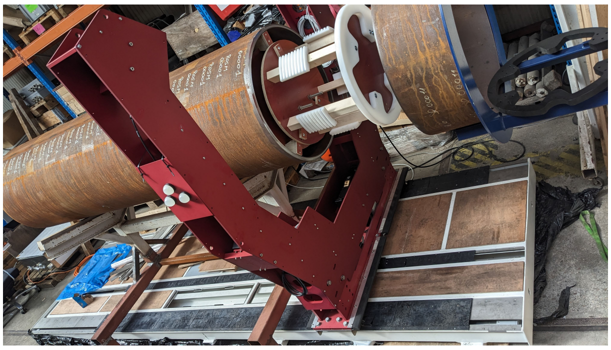
WI5118: AM Internship (cohort 2023)

Line-Up Automation

Philip Soliman (4945255)

October 2, 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 . . . 3
2 Problem description . . . 4
3 Approach . . . 5
3.1 Understand the complete problem, including the working of the machine 5
3.2 Familiarize with the code base . . . 5
3.3 Identify unreadable parts that need maintainability improvements . . . 5
3.4 Create a profiler for the code to search for slow operations . . . 5
3.5 Find more efficient ways to perform the needed operations . . . 5
3.6 Isolate parts of the mathematical model and test accuracy . . . 5
3.7 Expand the fitting model to increase the workability of the complete system 5
4 Results . . . 6
5 Conclusion . . . 7
6 Theory . . . 8
6.1 Circle fit error estimation . . . 8
7 Skills learned . . . 9

1 Company description

2 Problem description

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

To measure the pipe ends' position and orientation, 5 laser line scanners are used. These scanners scan in 5 different planes and each generate roughly 4000 (x, z) data points in their respective plane at 60 70 Hz. Within the clouds of data points the patterns representing the pipes are recognized, with the use of a line finding algorithm, derived from a Hough transform. The data of the 5 sensors is combined and the pipe ends are fitted in 3D with a combination of Newton-Raphson optimization and a least squares fit. All done in C++.

The internship problem entails 3 aspects: - Improving code speed by searching for and eliminating slow operations - Improving accuracy by testing different fitting algorithms - Improving code maintainability

3 Approach

3.1 Understand the complete problem, including the working of the machine

- Read documentation, look at videos from real production - Get to know the engineers - See the test set-up at the yard

3.2 Familiarize with the code base

- Read the code and try to see the bigger picture

3.3 Identify unreadable parts that need maintainability improvements

Halfway during the development a new standard for 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 - Identify the pieces of code that are least readable - Get to know the Allseas coding standard - Apply improvements along the way while working on following points

3.4 Create a profiler for the code to search for slow operations

- Identify the bottlenecks in the code

3.5 Find more efficient ways to perform the needed operations

- Depending on the type of bottleneck do something about * memory allocation * order of operations * bug fixes for wrongly implemented algorithm logic * precomputation of constant values * optimize data locality

3.6 Isolate parts of the mathematical model and test accuracy

- Create tests using generated data in Matlab - Run tests on recorded data from the test set-up - Test the resulting accuracy of the process in the test set-up

3.7 Expand the fitting model to increase the workability of the complete system

During pipelay there is more equipment present in and around the pipe. Being able to recognize more than just the pipes could help by predicting where the pipe will be before we can measure it. This will benefit the overall speed of the process. - Assist in the development of a new part of the recognition software that will find a ribbed cylinder (without slowing down the program) - Test the accuracy and reliability with the test set-up in the yard - Different kinds of line, plane and circle fitting algorithms based on minimization of geometric distances can be used for this

4 Results

5 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

6 Theory

6.1 Circle fit error estimation

Al-sharadqah and Chernov 2009 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

$$E[(d_1 Q)(d_1 Q)^T] = s^2 (W^T W)^(- 1).$$

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. $d_1 Q$: 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.

7 Skills learned

Bibliography

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