Structuring a Database for Evaluating Water Pipeline Navigation Algorithms

OLIVER WILLIAM DAVIES, University of Twente, The Netherlands

Rosen-NXT examines water pipelines with Pipeline Inspection Gauges (PIGs). These are equipped with sensors to map the trajectory of the pipeline. A trajectory estimation navigation algorithm generates this trajectory. This paper proposes a data structure and methodology to evaluate the quality of such an algorithm structurally while following the design science methodology of Roel Wieringa. The proposed solution generates trajectories leaving Above Ground Markers (AGMs) out of the generation process and measures the longitudinal, lateral and elevation error at the timestamp of the left-out marker.

Additional Key Words and Phrases: Trajectory estimation algorithm, water pipeline, database, Above Ground Marker

1 INTRODUCTION

Rosen-NXT[1] is a technology group which split from its parent company, Rosen group[2], at the beginning of 2024. Rosen-NXT focuses on future-based innovations to offer solutions to customers in challenging environments. The Waterline Integrity Solutions branch of Rosen-NXT focuses on water supply and ensuring that the water line infrastructure meets the needs of communities around the world. This paper describes a research project done in collaboration with Rosen-NXT. Over the past weeks, we have investigated how they can best structure their available data and define metrics such that their algorithms can be structurally evaluated. Section 1.1 will contain background information about Rosen-NXT, its water supply branch and its navigation algorithms, and it will expand on Rosen-NXT's current situation and the problems with the system. In section 2, the research questions and the methodology of answering them will be specified, followed by a more in-depth survey of the problem in section 3. This report will include requirements engineering of the desired solution in section 4 and finally a proposed solution for Rosen-NXT's navigation algorithm evaluation methodology in section 5. The solution will be evaluated in section 6 based on semi-structured interviews conducted with Rosen-NXT team members. The document will be concluded in section 7 with a summary and the answers to the research questions, a reflection on the project procedure and suggestions on future work to be done.

1.1 Background

The water line infrastructure is ageing; because of inadequate maintenance, existing water distribution systems have deteriorated, which leads to high leakage losses [4]. Globally, water loss in water distribution networks totals over 30% [7], and can reach up to 70% in some countries [3]. To combat this water loss, Rosen-NXT designs and builds tools to inspect water pipelines and detect potential fragilities. These tools include in-line inspection tools equipped with multiple sensors and technologies to detect fragilities in pipelines [3]. One of the reasons fragility or damage of pipelines is caused is by bending of the pipe or shear forces [7]. This can be caused by multiple external factors, ranging from the earth freezing and thawing to earthquakes and landslides [1]. As a consequence, if the trajectory of the pipeline has changed significantly compared to its original construction, it is logical to assume that the probability of bending or shear forces creating fragilities in the pipeline will be higher. Besides, knowing the geographic trajectory of their pipelines is simply necessary for Rosen-NXT customers to know where they need to perform maintenance. While trajectories of above-ground pipelines can be determined with *global positioning system* (GPS) technology, this cannot be done for underground pipelines; so, a *pipeline inspection gauge* (PIG) travels inside the pipeline to determine the trajectory [9]. There are multiple tool and sensor setups possible for pipeline trajectory determination [2, 5, 6, 9], and most are composed of an *inertial measurement unit* (IMU) and odometers to determine the position of the tool [9]. This is also the case for the tool that Rosen-NXT uses. Based on the measured IMU and odometer data, along with configuration and calibration data, an estimated trajectory of the pipeline needs to be given. A pipeline can also be fitted with several *above ground markers* (AGM), which give the timestamp at which the PIG is at the marker's DGPS coordinates. Using the data of these AGMs, along with the sensor data from the PIG, Rosen-NXT defines navigation algorithms to output the estimated path of the pipeline as accurately as possible. It is important to note that the number of above-ground markers, and thus the amount of available "ground truths", is very limited - between twenty and forty - per pipeline.

1.2 Problem statement

Currently, Rosen-NXT stores its data in a file-based system. The data is sorted first by the (around ten) pipelines which they have inspected. Every pipeline contains several 'runs' of Rosen-NXT's tool within the pipeline. In each run, there is raw sensor data, sorted by groups of sensors. All the file types of the data differ entirely; some of the data is contained in .csv files, some are contained in .srh5 files[4], some are .mp4 videos and some is in .gpkg Gis files. Where each relevant piece of information is stored, and in which file type, differs per piece of information, and is among many other files which are uninteresting for making or evaluating a navigation algorithm. What's more, not every run uses the same pipeline tool with the same sensors; in other words, for some runs useful data is simply not available. For these reasons, Rosen-NXT plans to move

[1]https://www.rosen-nxt.com/

[2]https://www.rosen-group.com/en

[3]https://www.rosen-group.com/en/business-fields/oil-and-gas/pipelines/pipeline-inspection

[4].srh5 is Rosen-NXT's internal filetype, similar to .hdf5

TScIT 41, July 5, 2024, Enschede, The Netherlands

from a file-based system to a data warehouse during the duration of this project, so that relevant data can be retrieved for different tasks Rosen-NXT needs to fulfil.

Besides this setup, Rosen-NXT does not currently have a way to structurally evaluate its navigation algorithms on sensor data from real customer runs to determine the most accurate one. As mentioned above, there are very few ground truth values for a trajectory of a real-life pipeline inspection run. So, at this time, navigation algorithms are compared on single runs and the best one is decided mainly by a visual estimation compared to a previously delivered productive trajectory. This project presents a solution to this specific aspect of the data structure. Based on this problem, we have defined the goal of the solution.

1.3 Goal

Rosen-NXT wishes to have a data warehouse containing important data for structurally evaluating its navigation algorithms. If a new algorithm is proposed, there should be a database containing a reference to the input data in the data warehouse and the corresponding trajectories of the current 'best' algorithm, alongside metrics which define how good the current algorithm is, such that the new algorithm can run on the available input data and generate its trajectories with its metrics. With the available metrics, a comparison can then be made more structurally which algorithm is better overall.

2 METHODOLOGY

To achieve the goal outlined in the previous section, the following research question has been defined.

2.1 Research questions

- **RQ**: How should Rosen-NXT approach systematically evaluating the quality of new potential trajectory generation navigation algorithms compared to their current working algorithm?

To answer this research question, we define two subquestions:

- **SQ1**: What data structure to structurally evaluate different navigation algorithms on a set benchmark is effective for Rosen-NXT within their software constraints?
- **SQ2**: Which metrics should we use to evaluate the quality of a navigation algorithm?

To offer Rosen-NXT a solution to their data warehouse problem, the data structure has been based on a design and engineering cycle, as described by Roel Wieringa in "Design Science Methodology for Information Systems and Software Engineering" [8]. Due to time constraints in implementing the solution, one design cycle has been executed and the user evaluation will be passed on to Rosen-NXT directly for future improvements. First, we will document in more detail the status quo of Rosen-NXT's data structure and algorithm evaluation methodology, and the aspects that are viewed as suboptimal and for which a solution is desired will be specified. Following this, team meetings and brainstorming sessions yielded the database and the evaluation methodology requirements. In gathering the requirements the team helped to answer SQ2. Based on the requirements, we will propose a data setup for the quality comparison of navigation algorithms to answer SQ1. A user evaluation of the proposition has been conducted with Rosen-NXT employees to define whether it is an improvement, which future developments are necessary, and whether our overarching research question has been answered. These will be documented for future design iterations.

3 PROBLEM INVESTIGATION

In this section, a more detailed investigation into Rosen-NXT's problem will be made based on the problem statement as described in section 1.2.

3.1 Inconsistent availability/structure of input data

Rosen-NXT's current data storage system is file-based. For all customer runs, there is a subdirectory for each pipeline. For each pipeline, some folders contain general information, and the rest of the subdirectories are sorted by run. For every run, the data is divided into subdirectories of different sensor groups. In each of these sensor groups, raw data and pre-processed data from all the sensors in that group which have been used during the run can be found.

While the sensor group setups are generally pretty consistent per pipeline, small changes have been made over time, and further in the past, entirely different setups have been used, causing differing sensor data locations per run. Furthermore, each run uses different sensors. Which sensors have been used during a run can be found in a summary of the tool setup, which is occasionally included in a subdirectory of the run, but the only way to reliably check whether a sensor has been used during a run is to search for its data file manually. Besides this, as mentioned in section 1.2, the file format of each used sensor differs per sensor. For all runs since 2022, which were deemed the relevant runs for the benchmark, we made a categorization of the locations of all relevant data in Rosen's file system or indicated it as unavailable. Documentation of which sets of sensor data are available for which run can be found in Appendix A. For each run, based on the availability of the relevant sensor data, Rosen-NXT denotes a colour-coded estimation of the usefulness of the runs; a green run will most probably have sufficient data to generate an interesting trajectory, a red run has too little available data and a yellow run is uncertain and needs to be evaluated at a later stage. A first glance at this table suggests that there could be some runs with enough sensor data to generate a reasonable trajectory, but that the majority of runs miss important sets of sensor data for trajectory generation. In conclusion, there is a limited and inconsistent availability of interesting sensor data for trajectory generation, and the location of the available data differs per run.

3.2 Unstructured algorithm evaluation

Currently, Rosen-NXT evaluates its algorithms mainly on an ad-hoc basis. There are a couple of pipeline inspection runs for which the team internally knows that the data quality is high, so an algorithm is run on the data from those runs and the resulting trajectory is intuitively evaluated on whether it makes sense. Alternatively, trajectory generation algorithms are tested on indoor or outdoor test loops on the Rosen Group campus, for which the exact trajectory is known, but it is not possible to entirely accurately reproduce real-life situations in a factory setting.

3.3 Lack of ground truth data

For an underground pipeline, the trajectory of the pipeline is not visible to the naked eye. Sometimes, Rosen-NXT's client has documentation of the pipeline trajectory when they placed the pipeline, but this documentation is not always accurate to begin with and logically does not take any potential shifting of the pipeline over time into account. As for sensors for trajectory generation, these often have inaccuracies: odometers can suffer slippage and IMUs can show random walk, for example. So, there are very limited aspects of a pipeline inspection which can be trusted absolutely as a ground truth for the position of the pipeline. The only features that are reliable documentation about the trajectory of the pipeline, and thus could be used for evaluation of navigation algorithms, are Above-Ground Markers (AGMs); boxes of which the DGPS coordinates are known, which save a timestamp as soon as the PIG passes underneath it. The amount of AGMs placed along a pipeline differs per run and pipeline, but in general, the aim is to place an AGM along the pipeline approximately every 500m - 1km. Note that if the pipeline has shifted, so the PIG does not pass under the AGM, it will not trigger; in this case, we do not know the position of the pipeline, but we only know where it is not. Furthermore, note that we can not only use AGMs for algorithm evaluation; AGMs are also an important part of the algorithm to generate the trajectory in the first place. Navigation algorithms are configured to ensure the trajectories pass through the coordinates that are known locations of the pipeline. If all of the AGMs are used for evaluation purposes, and not for generation purposes, the quality of the generated trajectory would be considered to be so much lower that the evaluation of the trajectory would no longer be of interest.

4 REQUIREMENTS FOR A SOLUTION

In the first weeks of the project, multiple meetings were held with stakeholders for this system within Rosen-NXT to determine requirements for the treatment of the aforementioned problems. Below, a short documentation of the desired processes is first made based on the meetings. After this, the concrete requirements are specified below and are justified by contribution arguments in line with Roel Wieringa's guidelines for requirement specifications [8].

4.1 Desired system

4.1.1 Data structure. One of the aspects which was discussed within the team, and with data experts elsewhere in Rosen-NXT, relates to the team's internal data setup. Currently, all the relevant data is file-based, with all its structure and availability problems as discussed in section 3.1. The water solutions team of Rosen-NXT is working on moving to an internal data warehouse so that all its data can be moved into a large structured storage space and can be retrieved by different modules that the team develops. For trajectory generation and its evaluation, the entire desired data setup can be broken down into a few steps:

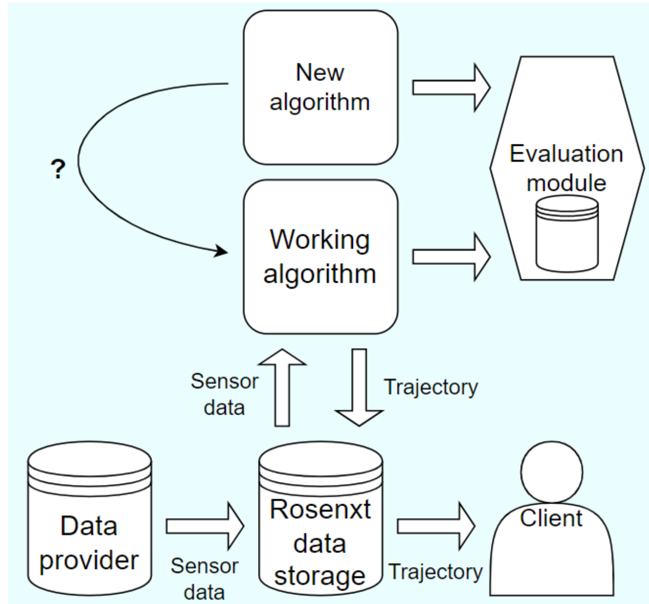

Fig. 1. An overview of Rosen-NXT's desired data setup

- The raw data from a pipeline run is made available to Rosen-NXT by Datathena, their data centre. This data is loaded into the data warehouse.
- The raw data is pre-processed into more meaningful data.
- The meaningful data is passed to a navigation algorithm which outputs an estimated trajectory.
- The estimated trajectory of the pipeline is placed into a database for other departments or clients to retrieve if necessary.
- A "benchmark database" accesses an estimated trajectory and its associated input data and stores metrics describing the accuracy of this trajectory.

The desired data setup is described internally with Figure 1.

4.1.2 Algorithm comparison methodology. To check whether a new proposed algorithm performs better than the current working algorithm, both algorithms must be tested with the same input data and evaluated on the same metrics so that a fair comparison can be made. This is a process similar to A-B testing, a methodology commonly used in marketing analytics.

Regarding the methodology to compare algorithms on a specific run, the following considerations have been made during the process of conceptualizing the system:

- There needs to be a fair and unbiased division of the available AGMs between trajectory generation and trajectory evaluation.
- Evaluation metrics need to be stored which describe the accuracy of the generated trajectory.
- Performance metrics of the algorithm need to be stored to describe logistical aspects of running the algorithm, such as running time and memory cost.

4.1.3 Evaluation metrics. Ultimately, a working algorithm and a proposed new algorithm will be compared to one another with pre-determined metrics to decide which performs better. To determine the accuracy of a trajectory estimation, some interesting possible metrics have been discussed:

- The lateral, longitudinal and height error of the trajectory against each "evaluation AGM" gives information about potential biases of the algorithm towards certain directions.
- The angle of the estimated trajectory right before and right after an AGM can be estimated. This information can give suggestions about a potential angle bias of the algorithm in a certain direction.
- The distance of the estimated trajectory from a client-supplied trajectory can be determined. This client trajectory is not a ground truth, but with this trajectory, a general suggestion can be made about whether certain sections of trajectory estimations are reasonable.

4.2 Requirements specification

4.2.1 Data structure.

- If Rosen-NXT's data would satisfy the requirement that it is easily retrievable via a data warehouse, given that their relevant real-life sensor data, algorithm results and client deliverables are migrated to some data warehouse system, it would satisfy their goal of being able to easily retrieve desired data for each of their different analysis or calculation modules.
- If a computed trajectory meant for distribution to the client is saved in the data warehouse beforehand, given that a working trajectory can take hours or days to generate, it would satisfy Rosen-NXT's goal of being able to provide a working trajectory when requested within seconds.
- If the analytics layer of the data warehouse would have a module which compares the performance of two algorithms on real-life sensor data sets, given that the same methodology would be used to generate trajectories for both algorithms on the same input data, it would satisfy the goal of being able to compare two algorithms with one another with pre-defined metrics to determine the better one to generate a productive trajectory.

4.2.2 Algorithm comparison methodology.

- If the AGMs used to generate a trajectory are split into a "generation" and "evaluation" set, given that AGMs are both crucial for the generation of a trajectory as they are the only sensor data considered as "ground truth", it would contribute to the goal of being able to define some evaluation metrics based on ground truth without sacrificing too much trajectory generation accuracy.
- If two evaluation trajectories are generated, with every AGM contained in the evaluation set of one trajectory and the generation set of the other, it would contribute to the goal of weighting all AGMs fairly between evaluation and generation.
- If both the current productive algorithm and the proposed new algorithm are run, given the same sensor data and the same evaluation methodology, it would be able to contribute to the goal of comparing both algorithms fairly.

4.2.3 Evaluation metrics.

- If the coordinates of the trajectory are checked at the same timestamp that an evaluation AGM is triggered, given that a perfect navigation algorithm should give the same coordinates as the AGM, being able to see the difference in coordinates would contribute to the goal of saying something about the accuracy of the algorithm.
- If the lateral distance between an evaluation AGM and the trajectory is measured, the lateral distance being the distance perpendicular to the direction of the pipeline, this error can contribute to the goal of giving information about the quality of handling IMU data by the algorithm.
- If the longitudinal distance between an evaluation AGM and the trajectory is measured, the longitudinal distance being the distance along the direction of the pipeline, this error can contribute to the goal of giving information about the quality of handling odometer data by the algorithm.
- If the height distance between an evaluation AGM and the trajectory is measured, this error can contribute to the goal of giving information about the quality of handling IMU data by the algorithm.
- If two algorithms are compared against one another, given that one has a lower average lateral error, longitudinal error and height error, this would contribute to the goal of being able to decide the better algorithm based on the lower error.

5 PROPOSED SOLUTION

In this section, a solution is proposed which aims to satisfy the requirements as specified in chapter 4.2.

5.1 Navigation algorithm evaluation module

Currently, Rosen-NXT maintains its data in a large hierarchical file system, but in the coming months, it plans to migrate to a data warehouse system. The system proposed is a "navigation algorithm evaluation module", which has read-only access to any data it needs and stores the metrics it generates in a database within the module. The data that the module can access is sensor data from real-life pipeline inspection runs which can be used to generate trajectories. The module is a boolean for each run on whether the data gathered is useful for trajectory generation or not. Since the module has read-only access to the data, it can be used both for the current file system and for the eventual data warehouse. Furthermore, the module can access all the developed navigation algorithms and knows which one is the current working algorithm which is used for trajectory estimations for clients.

5.2 Algorithm evaluation methodology

When an algorithm is to be tested, it is passed to the navigation algorithm evaluation module. The methodology of the module is explained step-by-step in the following sections.

5.2.1 Adequate sensor data. As indicated before, for every pipeline inspection run, a boolean indicates whether the data gathered is useful data for trajectory generation. So, this boolean gives a list of runs which are suitable for use in the evaluation module. Sensor data from a pipeline inspection run of which it has been indicated that it is suitable for algorithm evaluation will first be queried from the file-system/data warehouse.

5.2.2 K-fold cross validation. Once the sensor data for trajectory generation has been retrieved, the module will have files with sensor data to generate a trajectory. The module will also have a list of all Above Ground Markers which have been triggered along the trajectory. These markers need to be used for both trajectory generation and trajectory evaluation fairly. This is achieved by splitting the markers into two sets of alternating AGMs. When a trajectory is generated with an algorithm, one set containing half of the AGMs is used to generate the trajectory, and the other half of the AGMs is used to evaluate the trajectory. A second trajectory is then generated with the second half of the AGMs, and the first half is used to evaluate the resulting trajectory. This is done to ensure that each AGM has the same influence on the eventual metrics.

First, note that it is also possible to split the AGM list into three sets, or any other integer up to the length of the list. The decision has been made to use two sets, because generating a trajectory can take quite some hours, and the amount of time it will take to generate a trajectory using the evaluation module is a linear function of the amount of 'folds' we wish to create. If the navigation algorithms to be tested become quicker, the decision can be made to choose a higher 'k' for this k-fold cross-validation.

Furthermore, a small nuance is that the first and last AGM of a pipeline represents the starting and ending points of the trajectory. These are crucial for trajectory generation, so they are only included in the trajectory generation and are never left out for evaluation.

5.2.3 Distance measurement. For every AGM left out of the trajectory generation, the distance to the generated trajectory must be determined. Every AGM has a timestamp, a longitude and a latitude, and some have an elevation measurement. The time stamp of the entry of the generated trajectory which is equal to the time stamp of the AGM is retrieved. This entry shows the estimated latitude, longitude and height of the PIG at the given timestamp. Subtracting the latitude value of the previous timestamp from the current latitude value, and doing the same for the longitudinal and height values, gives a three-dimensional direction vector for the pipeline at that timestamp. Based on this vector, the distance between the AGM coordinates and the trajectory-generated coordinates can be broken down into three differences: a longitudinal difference, a lateral difference and an elevation difference. These three errors are saved for every AGM.

5.2.4 Evaluation metrics. Once three distance metrics have been saved for every AGM, an average lateral error, longitudinal error and height error can be defined for the generated trajectory by averaging these errors for every AGM in the evaluation set. These average errors should also be saved. For both generated trajectories, these average errors can in turn be averaged, to determine an average lateral error, longitudinal error and elevation error for the algorithm with the given sensor input data.

5.2.5 Algorithm comparison. The methodology described above for one set of sensor data should be executed with all remaining data sets which have been deemed appropriate for algorithm evaluation with the same algorithm so that a list is returned containing these three defined metrics for every interesting dataset. After this, the same approach should be taken for the working algorithm on all the data sets. The result will be two lists of lateral, longitudinal and elevation errors for both algorithms on the same input data sets. With these metrics, a comparison can be made to which algorithm performs better for trajectory generation.

5.3 Data structure

As mentioned above, the benchmark module will act independently of the larger data warehouse, in a structure resembling Figure 1.

Furthermore, the navigation algorithm evaluation module will have its internal database, to store metrics of pipelines and historical records of evaluation operations. Besides these values, it is also considered important to store the intermediate error values; since the eventual comparison metrics are averages, a high average error can be caused by consistently high errors per AGM, or consistently lower errors with a couple of massive outliers, for example. This information is not retraceable in the final metric, so Rosen-NXT must have access to the intermediate error values to be able to gather more detailed information on an algorithm's performance in specific situations. For every run, the same methodology will be used and the same values will be determined, so the structure of the data is rigid rather than flexible. For this reason, a relational SQL database is the best fit. The database structure of the evaluation module can be found in Appendix B. Besides the standard database relationships, there are some aggregation relationships in the database which are derived upon loading the data into the database. These are the following:

- When an `evaluation_run` is loaded with integer value `k_folds`, k entries of `evaluation_trajectory` are created with the 'fold' column ranging from 1 to k for those entries.
- When an `evaluation_trajectory` is loaded with integer value `rtp_numbers`, that number of entries of `evaluation_rtps` are created with the `rtp_number` value ranging from 1 to the value of `rtp_numbers` for those entries.
- The values `average_lateral_error`, `average_longitudinal_error` and `average_elevation_error` in `evaluation_trajectory_metrics` are averages of `lateral_error`, `longitudinal_error` and `elevation_error` in `evaluation_rtps`, grouped by `evaluation_run` and `fold`.
- The values `average_lateral_error`, `average_longitudinal_error` and `average_elevation_error` in `evaluation_run_metrics` are averages of `average_lateral_error`, `average_longitudinal_error` and `average_elevation_error` in `evaluation_trajectory_metrics`, grouped by `evaluation_run`.

6 USER EVALUATION

In this section, documentation is included of a user evaluation of the proposed solution. This evaluation was made by conducting semi-structured interviews with five members of the Waterline Integrity Solutions team of Rosen-NXT who are most involved with the execution and development of navigation algorithms. Three open-ended questions were posed to the interviewees, regarding their thoughts on the general methodology, data structure and metrics of the proposed solution, respectively. The main takeaways from each interview regarding advantages, disadvantages and possible future developments to the system are summarized in the following sections. The main takeaways have been included in the discussion on the future work, in section 7.4.

6.1 Interview 1

This team member had suggestions for changes to the proposed metrics. Firstly, the average errors are absolute and are not normalized compared to the distance to the closest AGM. For example, an error of one metre for an AGM twenty metres away from the previous AGM is more concerning than an error of one metre for an AGM five hundred metres away from the previous one. Furthermore, an additional metric that can be calculated is the angle change through an AGM, by determining the angle between the direction vector of the pipeline right before and right after the timestamp of the AGM. This can help to show consistent angle biases of an algorithm.

6.2 Interview 2

The interviewee was positive about the overall methodology allowing metrics to be defined for real-life runs. One disadvantage of the methodology concerns the frequency of the AGMs; generating a trajectory with AGMs that are spaced up to one kilometre apart can already be challenging, so if half of the markers would be left out this could have a significant impact on the accuracy of the trajectory. Given that some algorithms perform very well with enough AGMs, but very poorly with fewer, the quality of the evaluation module could be impacted significantly by leaving out AGMs. A solution could be to raise the amount of folds an algorithm is run for, but the feasibility of this depends on the time and resources required to run an algorithm once.

6.3 Interview 3

This interviewee pointed out that this methodology returns an average lateral, longitudinal and elevation error for every evaluation run, but also suggested that these errors be averaged over all runs with a specific algorithm so that algorithms could be compared against each other with one glance, to get a first idea which algorithm performs better. Furthermore, they mentioned that a simple boolean for sensor datasets to decide whether they are suitable for algorithm evaluation or not is potentially too naive; some datasets may be suitable for some specific test cases, but should not be included otherwise, for example.

6.4 Interview 4

This team member also indicated that the metrics should be normalised over the distance to the closest AGM. They also suggested that comparing the trajectory to an AGM at the same time stamp is useful for the longitudinal error of the AGM since it makes stretching errors more visible, but for determining the lateral and height errors, it would be more interesting to consider the point of the trajectory closest to the AGM, since that would give more information about how the algorithm handles the IMU data.

6.5 Interview 5

The interviewee pointed out the issue that an average elevation error is given, but that not every AGM has a height coordinate. So, the sample size of this average error should be given, or it should be omitted for some sensor data that lacks a height coordinate.

7 CONCLUSIONS

7.1 Summary

Ten weeks ago, I started at Rosen-NXT intending to help them create a data structure which will assist them in structurally evaluating potential new navigation algorithms they may develop. Based on preliminary meetings within the company, research questions were defined, of which the answers will be discussed in section 7.2. Once these were defined, the requirements for a solution were decided in consultation with the team. This resulted in some changes in the scope of the project which will be defined in section 7.3.1. With this, we proposed a solution to Rosen-NXT as specified in 5. This next section will discuss the answer it has given to our research question. Finally, a reflection on the process of the project will be given and future developments which can be made to the project will be discussed.

7.2 Answer to research questions

7.2.1 On answering SQ1. As discussed in section 5, Rosen-NXT will eventually move from a file-based data system to its internal data warehouse. A navigation algorithm evaluation module is a useful way to define the data structure for algorithm evaluation because it is entirely separated from the productive layer, and only accesses the data layer to read data for testing purposes. For this module, a database needs to be made to keep track of the measured metrics. Since the metrics that will be stored in the database are always the same values, the data structure can be rigid. So, a relational SQL database makes the most sense in this case. Ideally, a PostgreSQL database would be a logical solution for Rosen-NXT due mainly to its scalability and performance, which is useful for data structures which will continuously enlarge. However, PostgreSQL is not permitted on the Rosen-NXT IT systems. So, for now, a Microsoft Access database has been chosen as this is the only programme which allows for the creation of SQL databases, and the connection to Python will be made with SQLAlchemy.

7.2.2 On answering SQ2. The most important metric for seeing whether a navigation is good or not is the distance from an Above-Ground Marker not included in the trajectory generation, as this AGM can be considered a ground truth for where the PIG is located at a given timestamp. So, if the trajectory does not pass near this location, the algorithm has a significant issue.

It is specifically interesting to consider the distance between an AGM and the trajectory in three directions: longitudinal, along the direction of the trajectory, lateral, perpendicular to the direction of the trajectory along the same height, and elevation, perpendicular to the direction of the trajectory along the same latitude and longitude. These distances give information about the algorithm's quality of handling odometer data and IMU data respectively.

7.2.3 On answering RQ. A good way to approach the systematic evaluation of navigation algorithms is outlined in section 5. This methodology ensures that algorithms are evaluated against AGMs, which are important elements of the ground truth, and ensures that testing trajectories are generated with metrics and stored in a scalable relational database.

7.3 Reflection

7.3.1 Project scope evolution. Originally, the project was, to my understanding, aimed at creating the data structure for Rosen-NXT optimally and focusing on elements such as ensuring the data migration happens and the eventual structure is efficient for Rosen-NXT's use cases. However, this quickly proved to be out of the scope of the project in terms of time constraints; especially as the timeline of the data migration was delayed for some months. Instead, I focussed solely on the evaluation module and its data structure during this project.

7.3.2 Timeline. During this project, there have been some significant issues around the timeline. By the planning indicated in the proposal, the first four weeks would be spent finalizing the requirements for the system, so that I could spend five weeks on proper implementation and document the implementation in this report. However, unfortunately, some significant issues cropped up regarding my access to the software I needed for the project. For this reason, I could only start implementation properly by the end of the sixth week. This means that the solution is currently still being implemented, but that there is a lack of testing and user validation of the system itself; only an evaluation based on the explained system. For this reason, future developments to this system include proper finalization of the current design cycle.

7.4 Future work

In this section, future developments to the navigation algorithm evaluation module are specified. These have been determined based on internal meetings and the user evaluation as documented in section 6.

7.4.1 Data structure. In the next few months, Rosen-NXT plans to move from a file-based system to a data warehouse. This change makes sense to create a more intuitive layering system in the data. This should ensure that interesting data is more easily and consistently retrievable, rather than having to manually search through terabytes of messy files.

Regarding the data structure of the evaluation module, we recommend switching from a Microsoft Access database to PostgreSQL as soon as the software is made accessible because PostgreSQL is well-known for its scalability and performance. Most importantly, the software is not outdated.

7.4.2 Metrics. In the proposed solution, the lateral, longitudinal and elevation errors are described as the three most important metrics to evaluate navigation algorithms. However, these errors should be normalised by the distance of the AGM to the nearest other AGM. It is also notable that the elevation of an AGM is not always given; so, a system should be made to handle AGMs without a height measurement. Besides these metrics, however, there are some interesting other metrics which could give more information about the algorithm.

One of these metrics is the angle between the direction of the trajectory entering an AGM and the direction of the trajectory exiting the AGM. When generating a trajectory, the sensor data is translated into a trajectory, and this trajectory is fitted to the AGMs so that the pipeline goes through the AGM. Typically, a good navigation algorithm should return around 180°, since the pipeline path is then correctly estimated to go under the AGM. If, however, the algorithm has a set bias to the left, for example, this will mean that the angle of the trajectory entering and exiting the AGM will be more acute, and will often be similar after every AGM. Denoting this metric for the trajectory generation AGMs can give information about the potential biases of an algorithm.

Furthermore, some clients provide their trajectory of the pipeline to Rosen-NXT as a piece of ground truth data. However, this trajectory often has errors, and over time pipelines shift, meaning that this data is not useful to make meaningful claims about the quality of a navigation algorithm. Nevertheless, denoting the distance between the client trajectory and the generated trajectory can give some information about whether the generated trajectory is somewhat reasonable.

Finally, besides qualitative evaluation metrics on an algorithm, it could also be interesting to denote performance metrics: for example, how long an algorithm takes to run or how much memory it takes up. If the productive algorithm takes hours or days to generate the best trajectory, for example, an algorithm which generates a marginally worse trajectory in minutes could be useful for some use cases.

7.4.3 Fold optimization. Currently, an algorithm is run twice with two different generation-evaluation sets of AGMs. This has been decided due to time efficiency constraints, but whether this is the best number of runs to make conclusions on the quality of navigation algorithms is uncertain. In the future, algorithm metrics could be tested for a larger number of folds. Tests could be done to examine the correlation between the number of folds executed and the accuracy of the resulting determination of the quality of the algorithm, in some way.

7.4.4 Testing. To be able to use this algorithm productively, the system needs to be subject to unit-testing, end-to-end testing and integration testing to ensure that all methods and database queries are correct.

Finally, tests need to be done on the current sensor data sets to determine which ones are useful for navigation algorithm evaluation, which ones are only useful in specific test scenarios and which ones are not at all useful. This testing could help to determine which types of sensor data are criteria for the effective running of navigation algorithms.

ACKNOWLEDGMENTS

Over the past ten weeks, I have worked hard to offer a significant contribution to Rosen-NXT and have learned a lot in the process, most notably how software development teams approach large-scale projects. I would like to extend my thanks to the Waterline Integrity Solutions team of Rosen-NXT for allowing me to work on this challenging and meaningful project. Special thanks go out to Klaas Kole, Reza Serajeh and Nicolas-Alin Stoian for all their help and supervision. From the University of Twente, I extend my utmost appreciation to Faizan Ahmed, who always offered constructive feedback and suggestions and helped me approach the challenges that came up during the project.

REFERENCES

[1] Juan Pablo Alvarado-Franco, David Castro, Nicolas Estrada, Bernardo Caicedo, Mauricio Sánchez-Silva, Luis A. Camacho, and Felipe Muñoz. 2017. Quantitative-mechanistic model for assessing landslide probability and pipeline failure probability due to landslides. *Engineering Geology* 222 (May 2017), 212–224. https://doi.org/10.1016/j.enggeo.2017.04.005

[2] Dongjun Hyun, Minsu Jegal, and Hyun Seok Yang. 2010. Compact self-contained navigation system with MEMS inertial sensor and optical navigation sensor for 3-D pipeline mapping. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1488–1493. https://doi.org/10.1109/IROS.2010.5649766 ISSN: 2153-0866.

[3] M. A. Prieto, M. A. Murado, J. Bartlett, W. L. Magette, and Thomas P. Curran. 2015. Mathematical model as a standard procedure to analyze small and large water distribution networks. *Journal of Cleaner Production* 106 (Nov. 2015), 541–554. https://doi.org/10.1016/j.jclepro.2014.12.011

[4] Michael Rouse. 2014. The worldwide urban water and wastewater infrastructure challenge. *International Journal of Water Resources Development* 30, 1 (Jan. 2014), 20–27. https://doi.org/10.1080/07900627.2014.882203 Publisher: Routledge _eprint: https://doi.org/10.1080/07900627.2014.882203.

[5] Hussein Sahli and Naser El-Sheimy. 2016. A Novel Method to Enhance Pipeline Trajectory Determination Using Pipeline Junctions. *Sensors* 16, 4 (April 2016), 567. https://doi.org/10.3390/s16040567 Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

[6] H. Sahli, A. Moussa, A. Noureldin, and N. El-Sheimy. 2014. Small Pipeline Trajectory Estimation Using MEMS Based IMU. 154–161. http://www.ion.org/publications/abstract.cfm?jp=p&articleID=12543 ISSN: 2331-5954.

[7] Ridwan Taiwo, Ibrahim Abdelfadeel Shaban, and Tarek Zayed. 2023. Development of sustainable water infrastructure: A proper understanding of water pipe failure. *Journal of Cleaner Production* 398 (April 2023), 136653. https://doi.org/10.1016/j.jclepro.2023.136653

[8] Roel Wieringa. 2014. *Design Science Methodology for Information Systems and Software Engineering.* https://link.springer.com/book/10.1007/978-3-662-43839-8

[9] Shuo Zhang and Stevan Dubljevic. 2021. Trajectory determination for pipelines using an inspection robot and pipeline features. *Metrology and Measurement Systems* (April 2021), 439–453. https://doi.org/10.24425/mms.2021.137134

A APPENDIX A: SENSOR DATA FILE LOCATIONS

Pipeline	Run	Run date	Tool info	IMU STIM	MTI front	MTI rear	Dipper front	Dipper rear	Odometer	Accelerometer	RTPS	MP4 files	EB
300HAVUFF	1	23-06-2022	✔	✖	✔	✖	✖	✖	✖	✔	✔	✖	✖
300HAVUFF	2	23-06-2022	✔	✖	✔	✖	✖	✖	✖	✔	✔	✖	✖
300HAVUFF	3	26-01-2023	✔	✖	✔	✖	✖	✖	✖	✔	✔	✖	✖
300HAVUFF	4	26-01-2023	✔	✖	✔	✖	✖	✖	✖	✔	✔	✖	✖
400STESTE	1	14-05-2024	✔	✖	✖	✖	✖	✖	✖	✖	✔	✖	✖
400STESTE	2	15-05-2024	✔	✖	✖	✖	✖	✖	✖	✖	✔	✖	✖
400STESTE	3	15-05-2024	✔	✖	✖	✖	✖	✖	✖	✖	✔	✖	✖
1100ZEVROO	1	11-10-2023	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
1100ZEVROO	2	05-12-2023	✔	✔	✔	✔	✔	✔	✔	✔	✔	✖	✖
1100ZEVROO	3	06-12-2023	✔	✔	✔	✔	✔	✔	✔	✔	✔	✖	✖
1400ZEVZOE	1	19-04-2023	✔	✔	✔	✖	✖	✖	✔	✔	✔	✖	✔
1600WINWES	1	17-05-2022	✔	✔	✔	✖	✔	✔	✔	✔	✔	✔	✔
1600WINWES	2	18-05-2022	✔	✔	✔	✖	✔	✔	✔	✔	✔	✔	✔
1600WINWES	3	02-11-2022	✔	✔	✔	✖	✔	✔	✔	✔	✔	✔	✔
1600WINWES	4	03-11-2022	✔	✔	✔	✖	✖	✔	✔	✔	✔	✔	✔
1600WINWES	5	22-11-2022	✔	✔	✔	✖	✔	✔	✔	✔	✔	✔	✔
1600WINWES	6	23-11-2022	✔	✔	✔	✖	✔	✔	✔	✔	✔	✔	✔
1600WINWES	7	24-11-2022	✔	✔	✔	✖	✔	✔	✔	✔	✔	✔	✔

Table 1. A documentation of which sensor data is available for every run since 2022

B APPENDIX B: NAVIGATION ALGORITHM EVALUATION MODULE DATABASE STRUCTURE DIAGRAM

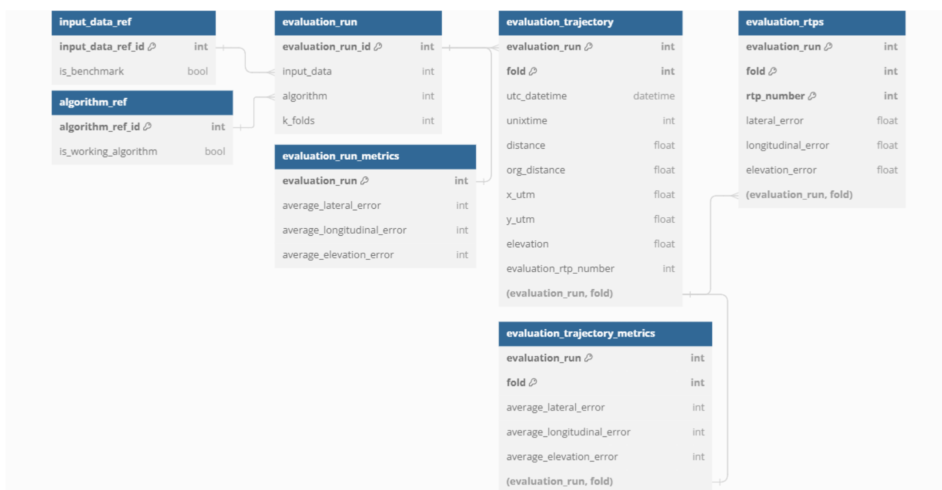

Fig. 2. Diagram of the navigation algorithm evaluation module database schema