

# Journal

## Contents

Journal .....	1
Monday 29-01-2024 (project week 0) .....	1
Tuesday 30-01-2024 (project week 1) .....	1
Wednesday 31-01-2024 (project week 1) .....	2
Thursday 01-02-2024 (project week 1) .....	2
Friday 02-02-2024 (project week 1) .....	3
Monday 05-02-2024 (project week 1) .....	3
Tuesday 06-02-2024 (project week 2) .....	4
Wednesday 07-02-2024 (project week 2) .....	4
Thursday 08-02-2024 (project week 2) .....	4
Friday 09-02-2024 (project week 2) .....	5
Saturday 10-02-2024 (project week 2) .....	5
Monday 12-02-2024 (project week 2) .....	5
Tuesday 13-02-2024 (project week 3) .....	6
Wednesday 14-02-2024 (project week 3) .....	7
Thursday 15-02-2024 (project week 3) .....	7
Bibliography .....	7

## Monday 29-01-2024 (project week 0)

---

### Kristoffer

Beumer

- Started project
- Had meeting with Jonas discussing what to start with the first 2 weeks
- Read these papers:
  - A visual introduction to Gaussian Belief Propagation [1]
  - Distributing Collaborative Multi-Robot Planning With Gaussian Belief Propagation [2]
- Tried compiling examples from <https://github.com/aalpatya/gbpplanner> but faced issues with missing X11 headers, even though they were installed on my system.

### Jens

Beumer

- Starting with a meeting all three (Kristoffer, Jens, Jonas)
- Read papers:
  - A visual introduction to Gaussian Belief Propagation [1]
  - Distributing Collaborative Multi-Robot Planning With Gaussian Belief Propagation [2]
- Successful compilation and run of examples from <https://github.com/aalpatya/gbpplanner>.
  - Successfully created custom environment to attempt to highlight weaknesses of the current implementation.

## Tuesday 30-01-2024 (project week 1)

---

### Kristoffer

From Home

- Created GitHub repository <https://github.com/AU-Master-Thesis/gbp-rs> as we want to rewrite the <https://github.com/aalpatya/gbpplanner> in Rust.
- Looked at different Rust simulation/visualization tools to use.
  - <https://macroquad.rs/>
  - <https://nannou.cc/>
  - <https://bevyengine.org/>
- Decided to go with **bevy** as it has a lot of community support/solutions and we thought its ECS system is really cool!.
- We read through the introduction book for bevy, to learn the core concepts behind the ECS paradigm and how applications are structured in bevy.

### Jens

From Home

- Set up Rust project structure
- Looked at the visualisation tools with Kristoffer, discussing which to go with.
- Learned Bevy and ran some examples
  - Wrote some of the examples out and mix-matched some of it to learn how it all fit together.

## Wednesday 31-01-2024 (project week 1)

---

### Kristoffer

OrbitLab

- Continued to have issues compiling the code for <https://github.com/aalpatya/gbpplanner>.
- We both decided to re-flash our OS with NixOS.
- Spent some getting acquainted with the terminology and methodology of how to do things in NixOS
- Create a `flake.nix` for both our Rust port and `gbpplanner` to create a reproducible environment, where we can compile and run the code without issue.

### Jens

OrbitLab

- Re-flash OS to NixOS
  - Learn NixOS and contemplated using hyprland
- 

## Thursday 01-02-2024 (project week 1)

---

**Kristoffer**

OrbitLab

- Continued learning about NixOS and setting up our development environment, with the tools we like to use.
- Spent some time trying to port the code from [https://colab.research.google.com/drive/1-nrE95X4UC9FBLR0-cTnsIP\\_XhA\\_PZKW?usp=sharing#scrollTo=NzotHENoaY6g](https://colab.research.google.com/drive/1-nrE95X4UC9FBLR0-cTnsIP_XhA_PZKW?usp=sharing#scrollTo=NzotHENoaY6g) to our Rust implementation.

**Jens**

OrbitLab

- Setting up NixOS and hyprland
- Migrating gbpplanner to Rust

**Friday 02-02-2024 (project week 1)**

---

**Kristoffer**

From Home

- Continued our attempt to port the code from [https://colab.research.google.com/drive/1-nrE95X4UC9FBLR0-cTnsIP\\_XhA\\_PZKW?usp=sharing#scrollTo=NzotHENoaY6g](https://colab.research.google.com/drive/1-nrE95X4UC9FBLR0-cTnsIP_XhA_PZKW?usp=sharing#scrollTo=NzotHENoaY6g) to our Rust codebase.
  - Jens wrote the code, while we both discussed how to port the Python code to Rust.

**Jens**

From Home

- Rust migration

**Monday 05-02-2024 (project week 1)**

---

**Kristoffer**

Beumer

- Read recent survey paper from 2023 [3].
  - No mention of any paper/approach using Gaussian Belief Propagation.
  - Many newer paper use AI methodologies.
    - Neural Network based
    - Genetic Algorithms
      - Ant Colony
      - artificial bee colony algorithm
  - Lin-Kernighan-Helsgaun heuristic algorithm (dunno, names sounds interesting )
  - Dynamic Particle Swarm Optimization (PSO) [ref: 126,128]

**Jens**

At Beumer

- Struggling to set up hyprland with displaylink
- Ended the struggle, and joined Kristoffer in continuing the Rust migration.

**Tuesday 06-02-2024 (project week 2)**

---

**Jens**

From Home

- Collaborative coding to migrate to Rust
  - Fixed a lot of compiler errors

**Wednesday 07-02-2024 (project week 2)**

---

**Jens**

AU 5124 139

- Attempted to continue for a while with the generic factor-graph gbp library we have been attempting to make, however, it had become too much of a headache so:
  - Started over, in a much simpler fashion
  - Supported with chatGPT
  - Supported as sparring partner, and made sure to understand things more precisely
  - Also added journal entries for all my previous weeks

**Thursday 08-02-2024 (project week 2)**

---

**Kristoffer**

AU 5124 139

- Continued working on the rewrite of gbpplanner in rust.
- Spent some time playing around with the C++ Eigen library, to confirm how various matrix operations and matrix slicing work, to correctly port them to rust.
- Reread parts of the methodology section, to better understand some of the math.

**Jens**

AU 5124 139

- Working with Kristoffer to continue translation to Rust.
- Decided to split the work load.
  - I looked at Bevy, and learned further how to work it.
  - Implemented an input-manager, such that the user can press keys on the keyboard or gamepad to interact with the simulation.
  - Applied some keybinds like movement, boost to change movement speed, and toggling of a dynamic unknown object in the simulation.

- The toggling is currently only done by setting alpha to 0/1, which should later also disable/enable the actor's hitbox.

## Friday 09-02-2024 (project week 2)

---

### Jens

From Home

- Decomposed input and objects in the Bevy ECS architecture.
- Watches episodes 1, 2, and 3 of the Bevy tutorial series.
- Decomposed the system even further to have movement handled by itself.
  - This introduced a bug where movement of objects don't stop.
- Reworked the project with a 3D scene and 3D camera.
- Created an asset loader, to handle the loading of the 3D models.
- Changed the moveable object to be a 3D model.

### Kristoffer

From Home

- Found a similar paper to the [2] called Robot Web [4], that had some interesting demo videos on their [website](#).
- My intention is to read it over the weekend or next week, to see how it differs from [2].
- Continued working on the port to Rust.
  - Abstracted the robot radius into a trait `BoundingBox`, where i am right now creating a impl for a `BoundingBox2d`
  - Created an abstraction for the `CommunicationMedia`
  - Tried using Julia to verify some of the math, but had issues getting the `Distributions` package to compile on `NixOS`
- Still need to figure out how Messages are exchanged between robots, in a way that it is decoupled from running the simulation in a single thread with the same address space to a multiprocess system.

## Saturday 10-02-2024 (project week 2)

---

### Jens

From Home

- Fixed the movement bug from yesterday, such that velocity is reset when the movement stops.
- Extended the `MovementPlugin` to handle rotation as well in a similar fashion to the movement.
  - Consider decomposing the rotation into its own plugin.
- Changed the moveable object to a box instead of the roomba.

## Monday 12-02-2024 (project week 2)

---

**Kristoffer**

From Home

- Continued working on porting gbpplanner factorgraph to Rust.
- Spend some time figuring out how to map Eigen operations to Rust nalgebra crate.
- Unsure about current design with every Variable have `Vec<Rc<Factor>>` and Factors having `Vec<Rc<Variable>>` makes the abstraction kinda awkward.
  - Thinking about restructuring and using a proper graph structure.
  - For this I think we can use the petgraph library.
- Also considered ways to abstract the connection with other robots/factorgraphs such that the library can be used in both simulation and real world setup.

**Jens**

From Home

- Finished initial camera controls.
- Set up an infinite grid to visualize the environment.

## Tuesday 13-02-2024 (project week 3)

---

**Kristoffer**

AU 5124 139

- Meeting with Andriy and Jonas.
  - Showcased and discussed what we had been doing for the first two weeks.
  - Talked about some of the shortcomings we had identified with gbpplanner, and which were relevant to pursue.
    - Try and incorporate a global planner with the GBP algorithm.
    - Use a scheme to only communicate with the robots most uncertain about, to reduce number of messages being communicated in areas with a lot of robots. Avoid  $O(n^2)$  complexity
  - Focus on first making a rewrite that reproduces the results of [2], before applying the software design decisions that we think would be better.
  - Focus on not only record what we do in text (in this journal), but also take pictures and video recordings of when things work and don't. Because they might be difficult to recreate later.
- Read “A Robot Web for Distributed Many-Device Localisation” [4].
  - Very similar to [2], I think some of the authors are the same.
  - They exchange messages using HTTP. Where each robot host a web server that the other robots can send POST requests to.
  - Similar results and findings as [2].
    - Some of their conclusions
      - “While increasing the communication radius improves the performance, 30m onwards, the difference is negligible.”
      - “Retain low ATE up to 50% message loss”

- “Demonstrates its resilience to large initialisation errors up to (0.2m, 0.2m, 0.2rad) after which it explodes”

**Jens**

AU 5124 139

- Meeting with Andriy and Jonas.
- Added zoom functionality to the camera with button bindings.
- Fixed object not visible from the beginning. Due to normalization of zero-vector.
- Same problem with moving the camera, everything would vanish after a while because of normalization of zero-vector.
- Read half of “A Robot Web for Distributed Many-Device Localisation” [4].

## Wednesday 14-02-2024 (project week 3)

---

**Jens**

AU 5124 139

- Set up mouse/touchpad keybindings for camera movement in simulation.

**Kristoffer**

AU 5124 139

- Finished reading paper [4].
- Worked on porting [2] C++ code to our Rust version.

## Thursday 15-02-2024 (project week 3)

---

**Jens**

AU 5124 139

- Made a system to add follow cameras to each robot tagged with `FollowCameraMe`.
- Made the follow camera work quite reliably and almost not laggy.

**Kristoffer**

AU 5124 139

- Ported most of `Robot`, `FactorGraph` and `Variable` class to Rust.

**15 weeks 1 day left to hand-in deadline**

## Bibliography

- [1] J. Ortiz, T. Evans, and A. J. Davison, “A visual introduction to Gaussian Belief Propagation”, *arXiv preprint arXiv:2107.02308*, 2021.

- [2] A. Patwardhan, R. Murai, and A. J. Davison, “Distributing Collaborative Multi-Robot Planning With Gaussian Belief Propagation”, *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 552–559, 2023, doi: [10.1109/LRA.2022.3227858](https://doi.org/10.1109/LRA.2022.3227858).
- [3] N. Abujabal, R. Fareh, S. Sinan, M. Baziyad, and M. Bettayeb, “A comprehensive review of the latest path planning developments for multi-robot formation systems”, *Robotica*, vol. 41, pp. 1–26, 2023, doi: [10.1017/S0263574723000322](https://doi.org/10.1017/S0263574723000322).
- [4] R. Murai, J. Ortiz, S. Saeedi, P. H. Kelly, and A. J. Davison, “A robot web for distributed many-device localisation”, *IEEE Transactions on Robotics*, 2023.