Reflection and Traceability Report on Optimal EM Placement

Hussein Saad

1 Changes in Response to Feedback

1.1 SRS and Hazard Analysis

The feedback I received on my SRS is summarized in the following list:
From the **Domain Expert**:

- **Feedback:** Increase abstraction in goal statements (link).
Outcome: Did not implement this piece of feedback as OEMP solves a particular, well-defined problem and I want the goal statements to reflect that.

- **Feedback:** Move the Manipulator Jacobian to Data Definitions (link).
Outcome: Did not implement this piece of feedback since my Actuation Matrix (IM1 of the SRS), which is an instance of the Jacobian, must be a refinement of either a TM or a GD, and the Manipulator Jacobian is better defined as a TM.

- **Feedback:** Define `int_max` (link).
Outcome: It turns out the Integer type is unbounded in Python 3 (language of implementation), so I did not have to make any changes here.

- **Feedback:** R1 seems to refer to user input and NFR2 to user properties (link).
Outcome: I changed the wording so that the requirements are related more to the program than the user.

- **Feedback:** Missing units and some notation in IM1 and TM4 (link).
Outcome: Added the units and notation.

From the **Instructor** (below items are from the document attached to Issue 2):

- **Feedback:** Some spelling, grammatical and citation mistakes.
Outcome: Made the recommended changes.

- **Feedback:** Where would an intended reader get their optimization background from.
Outcome: Mentioned crash courses.

- **Feedback:** Recommended looking into under-the-table length instead of just volume.
  Outcome: This was a great recommendation - I ended up needing it for pose generation and overlapping constraints.

- **Feedback:** Asked for more rational with regards to scope decision.
  Outcome: Discussed the reason assumptions are made and how the intended use of the program influences scope decisions.

- **Feedback:** Pointed out ambiguity in NFR! (Accuracy).
  Outcome: Specified the two naïve algorithms (greedy and random) that will be used to measure OEMP's accuracy.

1.2 Design and Design Documentation

The feedback I received on my design documentation (MG + MIS) is summarized in the following list:
From the **Domain Expert**:

- **Feedback:** Clarify program's interaction with hardware (link).
  Outcome: Explained that the program interacts with no external hardware.

- **Feedback:** Suggested changing the Main Module's category to a Control Module (link).
  Outcome: Changed it to a library as it was the closest available.

- **Feedback:** The Uses Hierarchy was not acyclic (link).
  Outcome: Fixed it so it's cyclic.

- **Feedback:** generatePos uses unseeded randomness, causing non-reproducible results (link).
  Outcome: Implemented a seeded RNG for generating poses.

- **Feedback:** An instance of the current I was in lowercase (link).
  Outcome: Fixed ($i \rightarrow I$).

From the **Instructor** (below items are from the document attached to Issue 9):

- **Feedback:** Draw the Uses Hierarchy such that all arrows point down.
  Outcome: Did just that.

- **Feedback:** Transitions are for modules with state variables.
  Outcome: Moved my "procedure" code to the output section.

- **Feedback:** A file might be a more flexible choice for parameter input.
  Outcome: InputParameters now loads and verifies a JSON config file.

- **Feedback:** Some scalar variables were bolded.
  Outcome: Unbolded them.

1.3 VnV Plan and Report

The feedback I received on my VnV Plan is summarized in the following list:
From the **Domain Expert**:

- **Feedback:** Asked to clearly declare what PV and TL mean in the test case table (link).
  Outcome: Added a note with a link to the Symbols and Abbreviations table which contains an explanation of the error message abbreviations.

- **Feedback:** Pointed out missing references (link).
  Outcome: Made the necessary citations.

- **Feedback:** Recommended testing `cvxpy` as part of NFRS (link).
  Outcome: I did not follow through as plenty of system tests (e.g. 4.1.2, 4.1.3) test the correctness of `cvxpy`.

- **Feedback:** Recommended testing for input type mismatches (link).
  Outcome: Added 5 test cases for input types.

From the **Instructor** (below items are from the document attached to Issue 9):

- **Feedback:** Do not rely on ad-hoc feedback for document (SRS, Design, VnV) strictly.
  Outcome: Implemented a checklist which reviewers will complete to increase the reliability of the documentation.

- **Feedback:** walkthroughs are done as group, not individually.
  Outcome: Changed implementation verification description to include group walkthroughs.

- **Feedback:** Name your testcases.
  Outcome: Did just that.

- **Feedback:** Explain how the Output row of Table 4 is supposed to be interpreted.
  Outcome: Added some text explaining how subscripts of the vector x represent the indices that correspond to the poses selected.

2 Challenge Level and Extras

2.1 Challenge Level

This is an advanced project - the program is an implementation of an algorithm designed to solve the optimal magnet arrangement problem in magnetic actuation systems, which has recently become a topic of interest for many roboticists, both applied and academic.

3 Design Iteration (LO11 (PrototypeIterate))

The design and implementation of OEMP saw some major changes over the course of planning and development. Since this is a research project, the emphasis at the beginning was on the correct implementation of the equations and algorithms (described in the TMs and IMs in the SRS), and perhaps, the user experience was neglected. After the first round of implementation, it was clear that although our algorithms might be correctly implemented, a slow user experience takes away from the usability of our program. Since the end goal is to package the program to be used by engineers, I decided to refactor and make some adjustments to OEMP's design. The most significant change to the user experience was the migration from console input (one parameter at a time), to a centralized configuration file. This drastically increased the usability of the program and reduced development and testing time as well.

4 Design Decisions (LO12)

4.1 Limitations

To avoid overlap between EMs, we currently approximate each coil as a sphere and perform spherical overlap checks. This approach leaves a portion of the under-the-table space unusable. Ideally, we would implement cylindrical overlap detection to more accurately capture each EM's shape, but with the added complexity, adequate testing and development might be unachievable in a single semester.

4.2 Assumptions

The program assumes all ElectroMagnets are identical and receive the same current. This allowed for simpler problem setup to test the convex optimization solver on optimal arrangement problem. Had we allowed for different EMs within the same system, most modules would have increased in complexity, which, given the scope of our project and the existing designs in literature, is unnecessary.

4.3 Constraints

Time was definitely the biggest constraint. Although the problem was clearly defined and we used proven tools, limited time hurt our testing progress and prevented much comparison with existing solutions.

5 Economic Considerations (LO23)

Magnetic actuation systems have been deployed (see Stereotaxis Niobe) in medical settings, however, there are still many challenges that need to be overcome before their widespread adoption. This program contributes to the development of such systems however it is to be used by academics and engineers who are involved in the design and testing of magnetic actuation systems. The program will remain open source, and we hope to attract users through conferences and workshops mainly.

6 Reflection on Project Management (LO24)

6.1 How Does Your Project Management Compare to Your Development Plan

The development of the program and its documents required more time than initially planned. The use of GitHub as a medium for feedback allowed for constant communication with the domain expert and instructor, however, other potential reviewers of the project were not adequately kept in the loop. The final implementation of the project was as planned, but the developer wishes they had spent more time and resources towards planning and executing a more thorough VnV, especially its Non-functional Requirements.

6.2 What Went Well?

The MIS document included almost all the design and implementation details I needed to begin building, and it proved invaluable when writing code. I already had a clear, high level idea of each module's purpose and how it interacts with the others.

6.3 What Went Wrong?

I struggled with configuring GitHub Actions and CI/CD pipelines — things that would have greatly improved my workflow. Writing tests also took far longer than I anticipated, and I didn't have enough time to rigorously evaluate the algorithm's performance to the standard I wanted.

6.4 What Would you Do Differently Next Time?

I would begin implementation earlier. Although a fleshed-out design provides guidance for implementation, I underestimated how many challenges only surface once you start building. Early prototyping would have given crucial feedback, which I would've used to refine my designs toward implementable solutions, and further from "in a vacuum" idealistic ones.