Reflection Report on Mechatronics

Team 28, Controls Freaks
Abhishek Magdum
Dharak Verma
Jason Surendran
Laura Yang
Derek Paylor

1 Changes in Response to Feedback

1.1 SRS and Hazard Analysis

The SRS saw some of the largest revisions for rev1. Feedback drove the addition of monitor & control variables to the context diagram discussion, greater detail especially in functional requirements (easier now that the designs had been done), and adding a list of figures to track our continual references to them. The rev0 VnV plan and report, along with our own developments in getting the project working, revealed that many more functional requirements were needed (and needed testing). As such, several more requirements were added for each system module. Hazard analysis was given scarce feedback, and saw much less rework for rev1. The focus was on this core stream: SRS > VnV Plan > VnV Report.

1.2 Design and Design Documentation

There was no feedback provided for design documents to improve from. The team updated the design documents to reflect new SRS requirements, and minor tweaks to module interactions.

1.3 VnV Plan and Report

Feedback from the VnV Plan was integrated by making our test plans much more specific as far as inputs and expected outputs, and the environment in which the testing occur ed. Along with the plethora of new SRS requirements, the VnV Plan functional tests grew massively. Feedback for the VnV Report was strong. We carried the same format, but of course updated with all the new test plans in the VnV Plan.

2 Design Iteration (LO11)

The final system architecture was arrived at in early 2023, which ended up following a similar architecture as the one shown first at the POC demo, wherein the 'Governor' was the top-layer module over all the other layer 2 modules. The team had agreed to rearrange the system into 3 layers (placing the lower level Driver interface and Motor interface out of communication from the Governor), then moved to encapsulate the Driver interface, Motor interface and Vehicle Dynamics into one larger module called Torque Path controller (as they all relate to translating the driver inputs into torque output, sometimes called a 'torque path'). These moves were scrapped during implementation, as it was deemed too tedious and complicated to add further abstraction between the IO (driver, motor) and the overall control (governor), and would actually restrict the flow of useful details as a result of this abstraction.

3 Design Decisions (LO12)

Design decisions were based fundamentally on the principal of modularity, or perhaps separation of concern, and based on some industry knowledge. For example, the concept of a 'torque path' (described above) is an existing concept that separates this path into subsystems. The module that simultaneously handles the state of 2 motors (Motor Interface) already has a lot to deal with. It shouldn't, for example, also have to perform the complicated task of computing an appropriate torque request (Vehicle Dynamics), especially when this includes features such as traction control or torque vectoring. Even without these additional features (as they could be added later), as a means of separating concerns we split this responsibilities to 2 modules. In a similar fashion, neither should have to process what the driver wants based on pedal input - this should already be handled elsewhere, hence the need for a Driver interface module. This also separates the large (and critical) component of monitoring the many driver inputs for errors, a task that could still use more sophisticated sensor diagnostic methods. Lastly, the design necessitated a higher layer module (Governor) for dictating states like 'ready to drive' or 'error -coast down' that involved several subsystems, whereby the subsystems communicate there state to Governor rather than each communicating to one another.

4 Economic Considerations (LO23)

Economic considerations had no effect on our project as the MFE team is the sole user, as such they provided the hardware required to test and run our software. As such, we had all the resources required to develop the software system to control their motors.

5 Reflection on Project Management (LO24)

Initially, tracking of project tasks were conducted through having team calls, where we split up the tasks and had team members volunteer for which sections of the project they wanted to work on. However, after the initial demo of the project, feedback from the TAs and Dr. Smith revolved around having a software tool used for tracking. Therefore we shifted to using Github issues as a way to track the tasks of our project. We would have weekly calls, where we would discuss the relevant tasks we have to work on, decide on who was going to work on those tasks, discuss any barriers to progress on already existent tasks, and then have any member discussions on how to improve our project.

5.1 How Does Your Project Management Compare to Your Development Plan

Communication plan, team meeting plan, team member roles, and workflow plan were all done according to the Development Plan documentation.

5.2 What Went Well?

Using Github Issues allowed for a clear way to track what each group member was working on, and how long they were working on those specific tasks. This provided a transparent way to see which tasks would require additional resources to complete, and which tasks could be completed quicker. It also allowed for an overall visualization of how our project was progressing, and allowed us to see if we were moving slower than expected. Additionally, it provided an electronic ”paper trail” about design decisions and allowed us to reference the discussions of specific tasks in case we wanted to double check our information. Having weekly calls also provided a way to keep up to date on our group members, and see if there were group members who could assist each other if they have the band-with.

5.3 What Went Wrong?

The team had a slow pipeline for several months of obtaining pertainant details from MFE. The requirements of the control system (in terms of monitor and control variables) were not known until late into the Fall term. Moreover, the state of the team's LV test bench (facilitating our testing with the motors) was also in a state of flux and led to setbacks well into February, and continued later through unprepared CAN drivers (hence using a laptop to run the control system for the demonstrations). This of course is not totally the fault of MFE. From the start, the entire team should've been brought closer to the MFE team's operations, acquiring Hatch centre permits earlier, and interfacing with team lead's directly for clarification on the state of the car and the role of the control system as it applies to their components on the car.

5.4 What Would you Do Differently Next Time?

In addition to the above comment, next time, the goal would be to start development of the project sooner rather than focusing too much on documentation. Early documentation was purely speculative - though it did force us to consider various elements of the project, documents were far easier to write once we had done the design and implementation work. Moreover, getting started on implementation instead of endlessly speculating down a decision tree (even if your chosen path isn't your end path) always helps you learn about your problem more, which hastens the feedback loop and allows you to reach a viable product quicker.