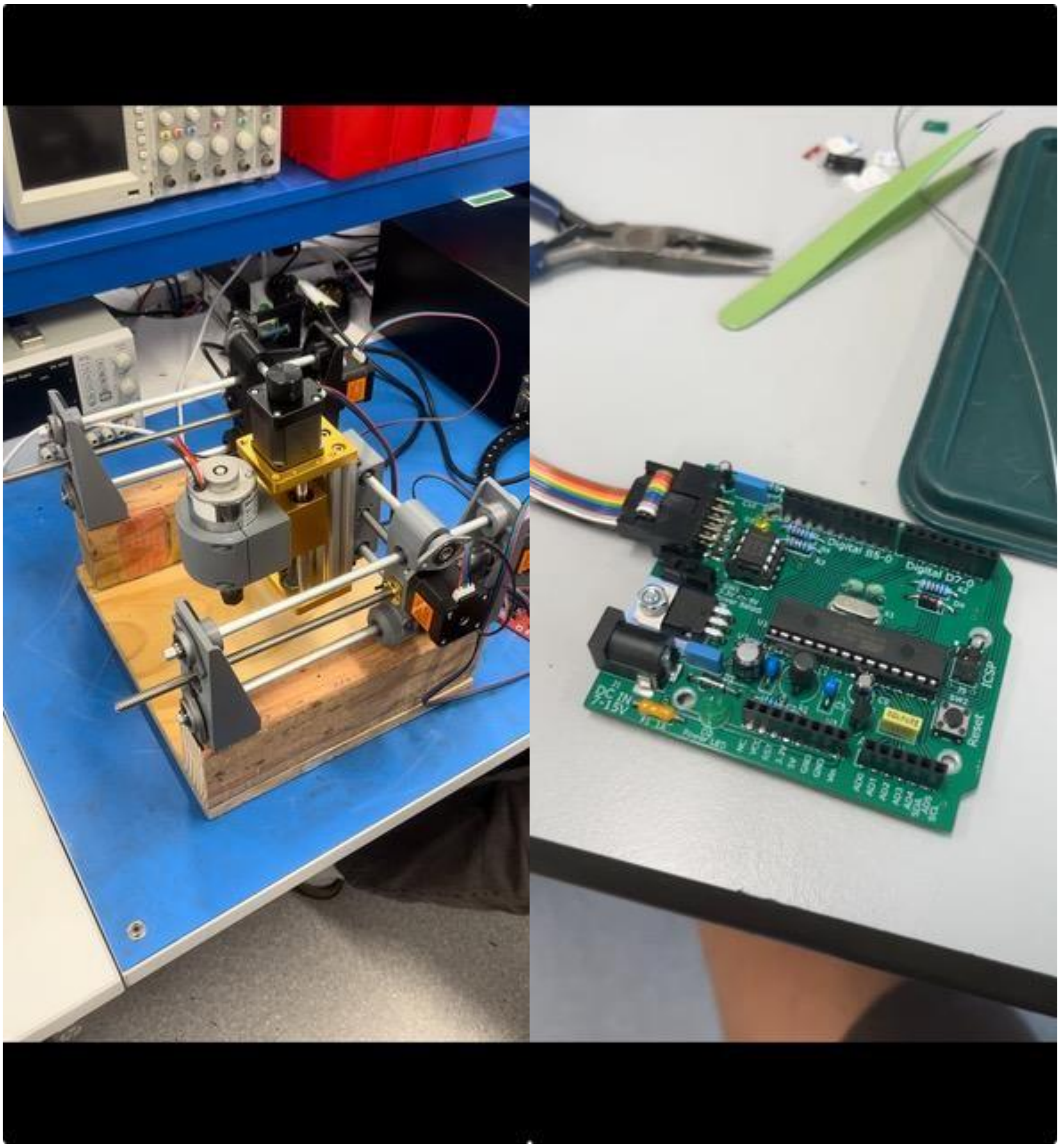
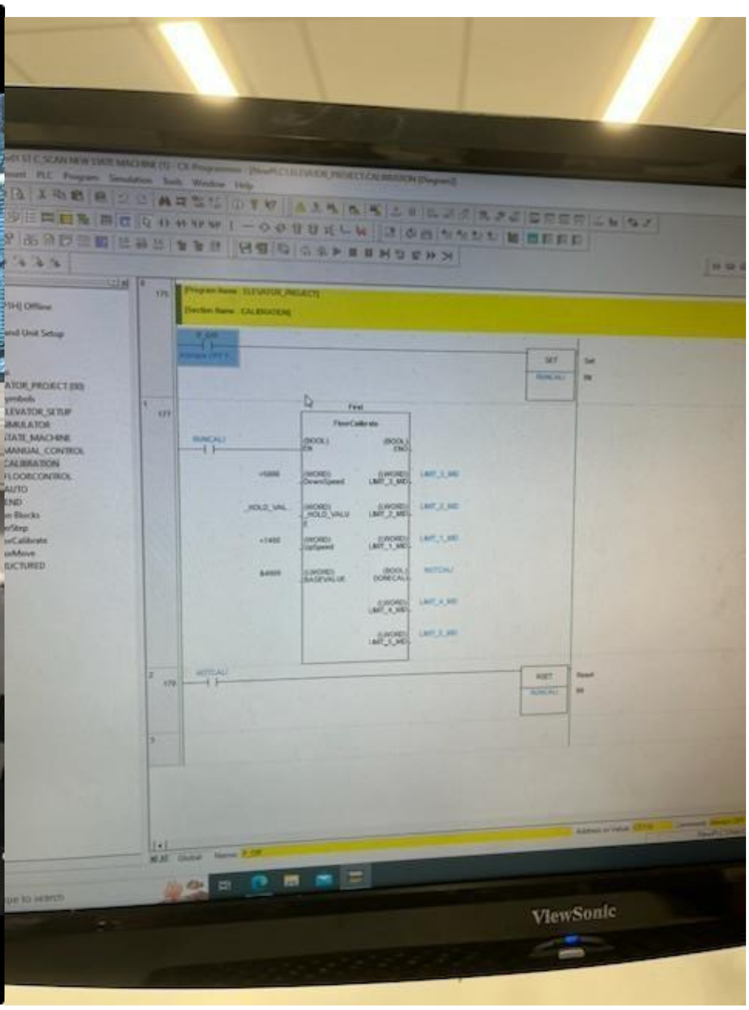
Mo Balfour-Lochner – Project Portfolio

A few projects completed very early on in my journey to becoming a Mechatronics engineer were soldering up a USB to serial converter and designing, building and programming a desktop CNC mill. Photos of these can be seen below.



Last year I completed two main mechatronics related projects. In the first we were required to write a program for a functioning elevator using a combination of the Ladder Logic and Structured Text programming languages. An Omron CP1H PLC was used as the controller to interface with a physical model elevator and its necessary actuators and sensors. All the code was written on the CX Programmer software. Overall, this project went very well and although it was very difficult it was quite enjoyable. A photo of the model elevator and one of our ladder logic function blocks can be seen below. My group partner and I invested a lot of time into this project and in turn produced a near perfectly functional elevator with no way to break the program. We received an A+ for this which was satisfying.

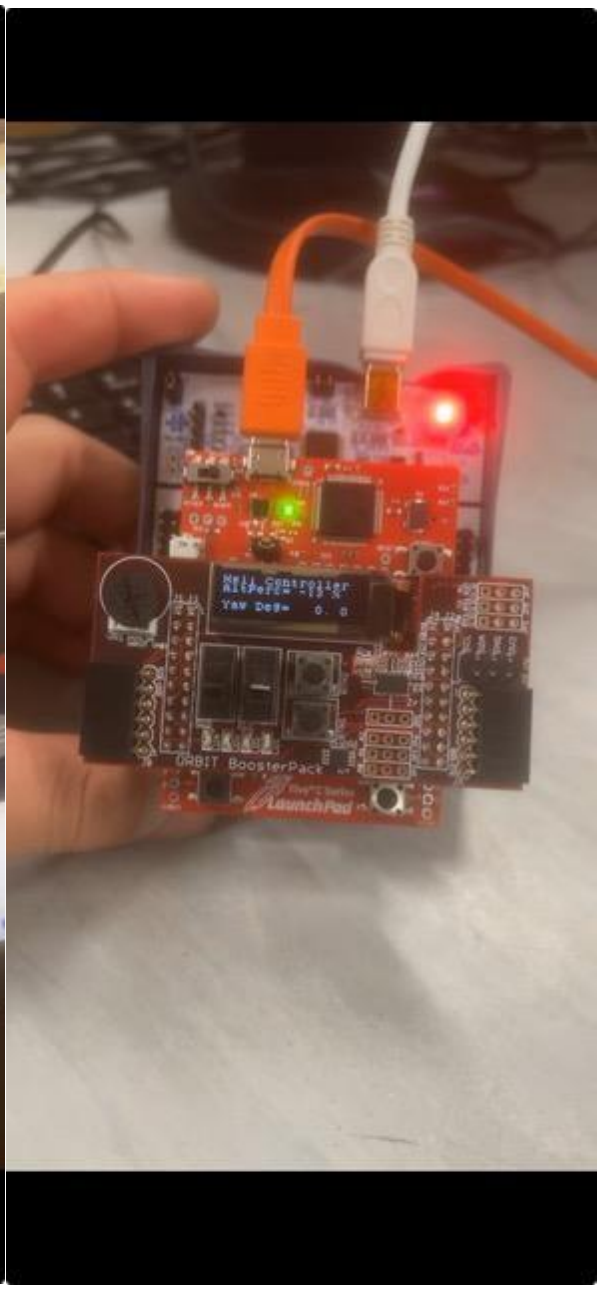


The second important project was a Line Following Robot. This project involved conducting electronic simulation, designing a custom circuit and PCB layout using Kicad, soldering up said PCB, programming an ATmega4808 microcontroller to control the robot, designing mechanical parts in CAD, and finally assembling and testing the robot so that it could follow a line course in the shortest possible time. A photo of our line following robot is shown below.

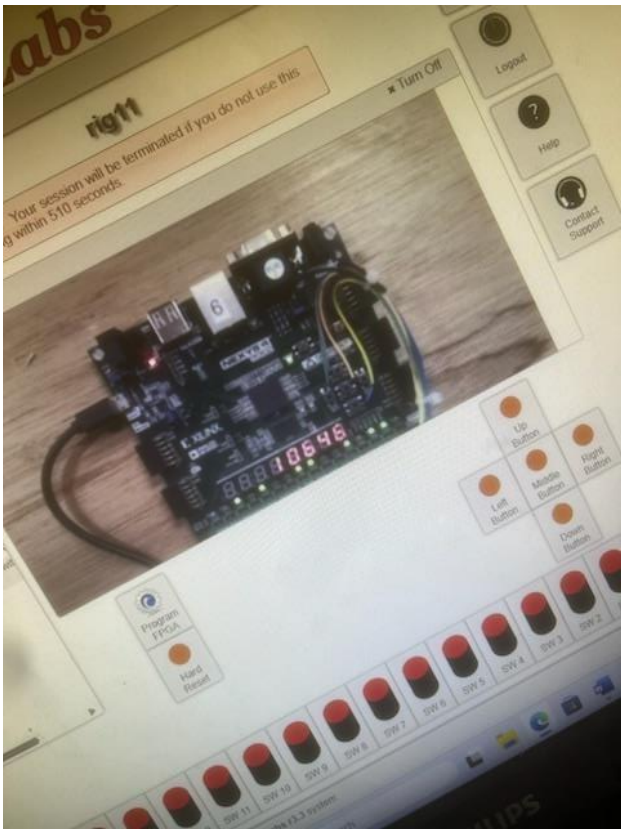
We decided to use our PCB as the chassis for our robot making it incredibly light weight and small. This was a good idea and stood out from the rest of the robots, unfortunately on the race day we had issues with the DC motors so it did not go as well as it could have. Despite this it was still heaps of fun and I learnt lots.

This year a few key projects have been programming a TIVA board to control an RC helicopter, designing and manufacturing an aluminium structure to break at a specified load, designing CMOS logic gates for unique Boolean expressions, using FPGA's to implement a reaction timer, designing a bespoke bearing and clutch system and lastly

Robocup, which involves designing and programming a fully autonomous weight collecting robot. Some photos of these things are shown below.



This project involved implementing a kernel (task schedular), using interrupts and creating a state machine to control the various flight operations required.



The reaction timer project seen above involved using Vivado and VHDL to program a reaction timer onto an FPGA. The reaction timer needed to have a pseudo-random number generator to stop people from anticipating the start of the timer. In addition, an ALU and memory had to be implemented to calculate the average, max and min reaction times of the last three recorded times which then needed to be displayed. An error message also had to be displayed if the stop button was pressed during the countdown, this all required a complex finite state machine to be implemented.

Another project for this course was to design and analyse a CMOS inverter for a given logic expression. First, we simplified the expression using Boolean algebra operations and then produced an inverter gate with pull-up and pull-down networks to match the truth table of the simplified expression. The network was designed to factor in the lower charge carrier mobility of PMOS devices so that the overall switching times were still symmetric. The final design was implemented and analysed in a program called TINA. A photo of the expression simplification working and the final design are shown below.

48958410	((\D)+(D+B)).((C.C).(\A))

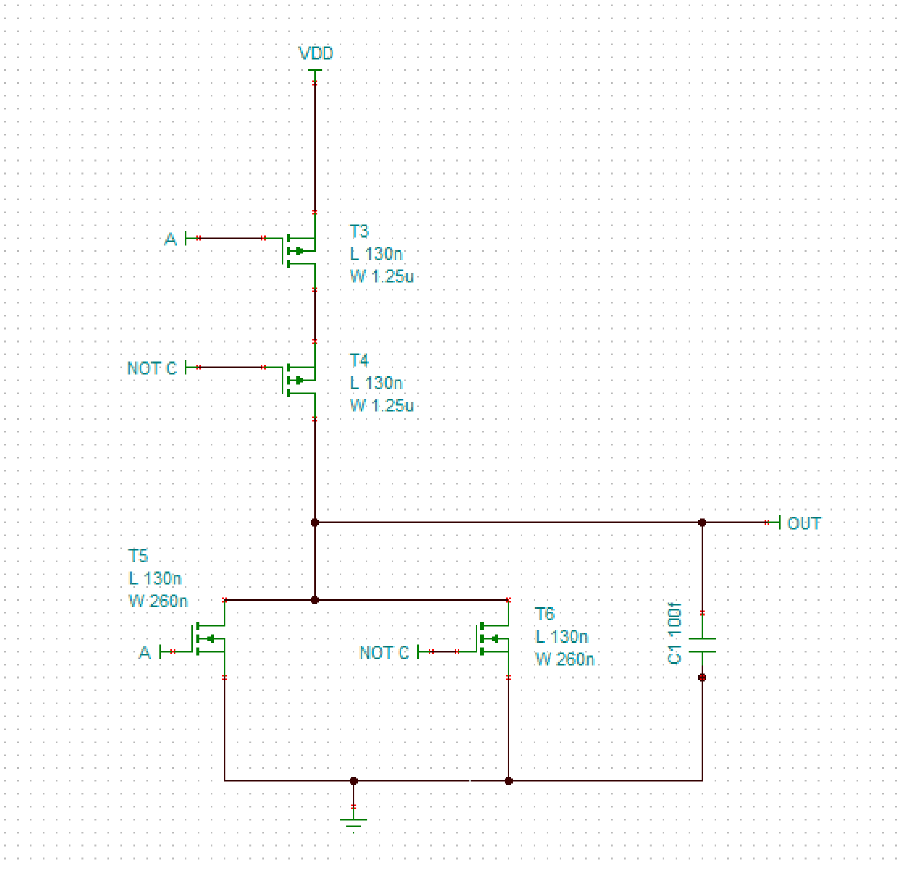
$\Rightarrow (\bar{D} + (D+B)) \cdot ((C \cdot C) \cdot \bar{A})$ $\Rightarrow$ Idempotent Law $C \cdot C = C$

$\Rightarrow ((\bar{D} + D) + B) \cdot (C \cdot \bar{A})$ $\Rightarrow$ Associative Law $\bar{D} + (D+B) = (\bar{D} + D) + B$

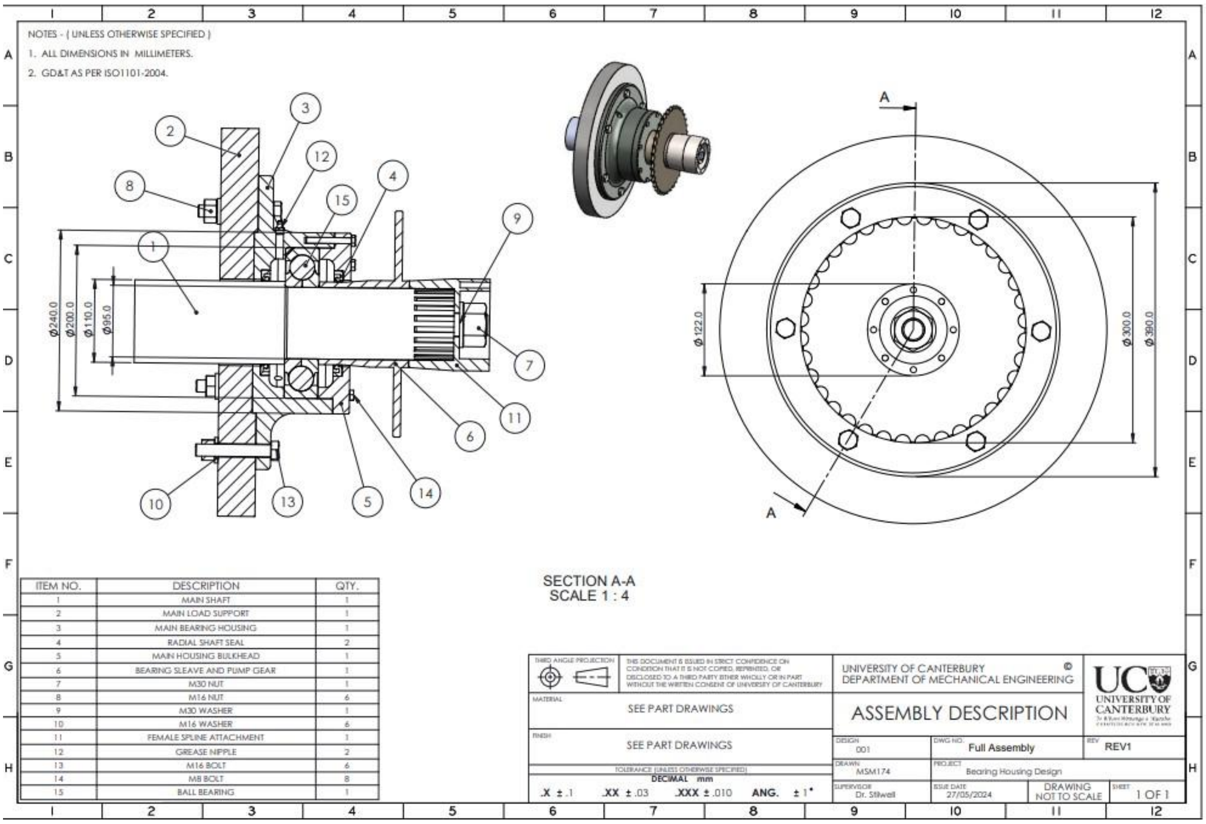
$(1 + B) \cdot (C \cdot \bar{A})$ $\Rightarrow$ Complement Law $\bar{D} + D = 1$

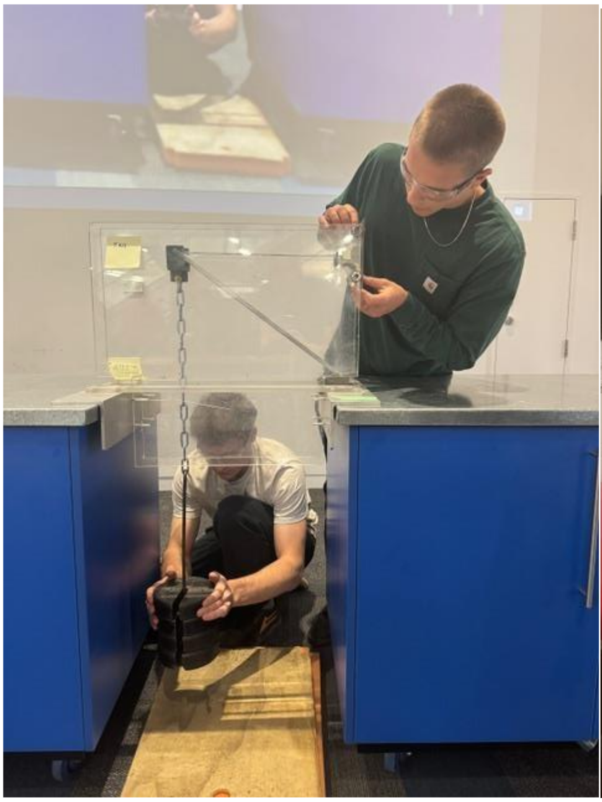
$1 \cdot (C \cdot \bar{A})$ $\Rightarrow$ Null law $1 + B = 1$

$\Rightarrow C\bar{A}$ $\Rightarrow$ Identity Law $1 \cdot C = C$



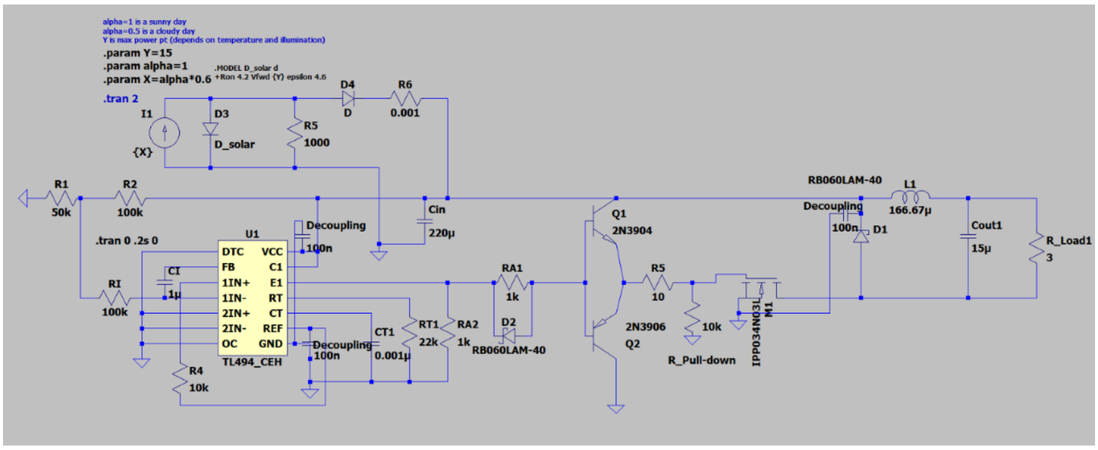
This year we also completed a project in which we had to design a bespoke bearing housing and clutch assembly for an example business venture. This involved carrying out calculations to determine suitable shaft diameters, bearing types, lubrication types and systems, clutch plate materials, number of plates, type of clutch etc. This was an interesting project as it essentially involved going through an entire engineering process from determining the client's needs to providing them with a well-engineered solution. I found this quite cool as it showed us that our growth as engineers and the real-life applications of what we learnt. A drawing of our bearing housing design can be seen below.

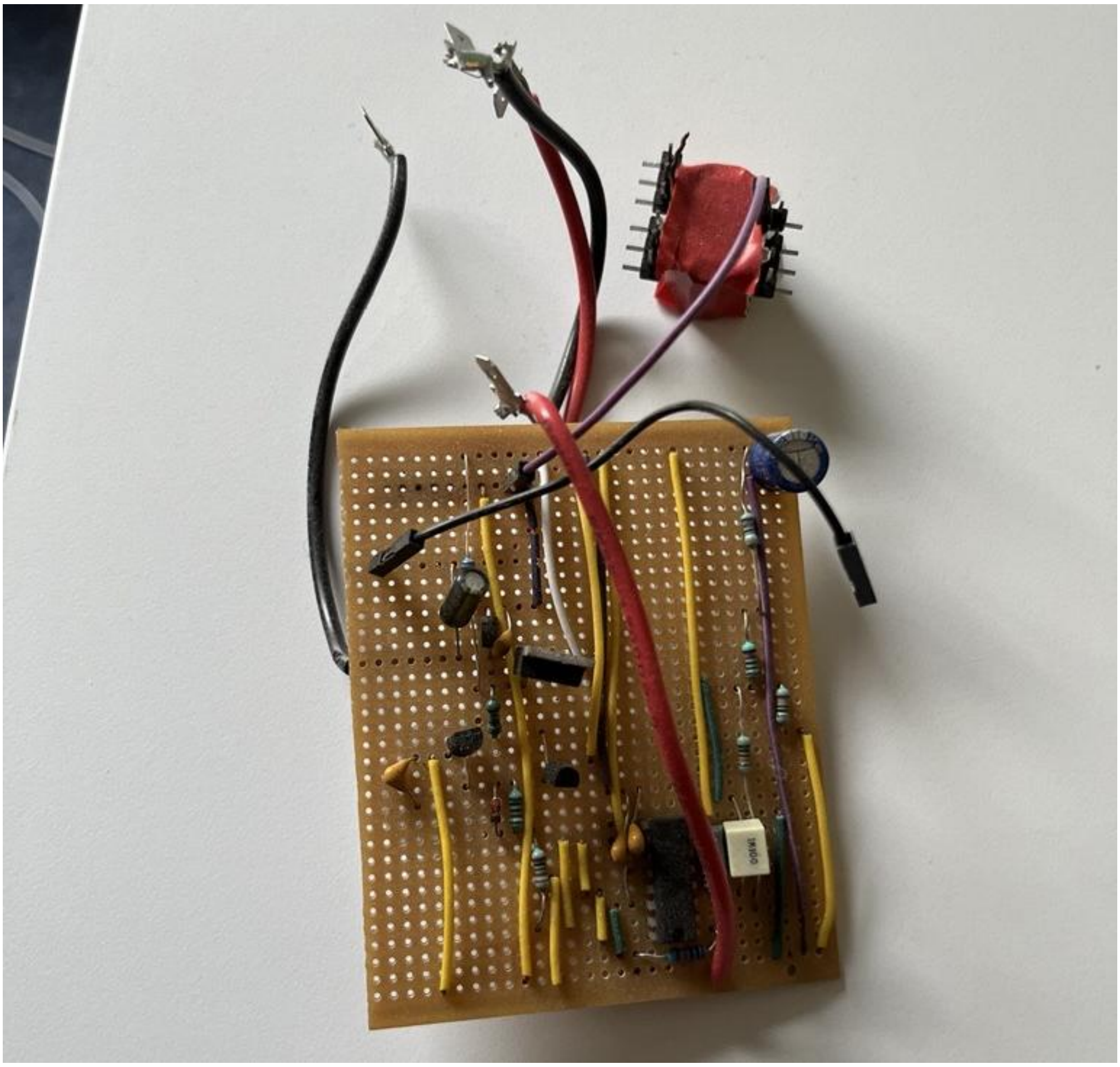




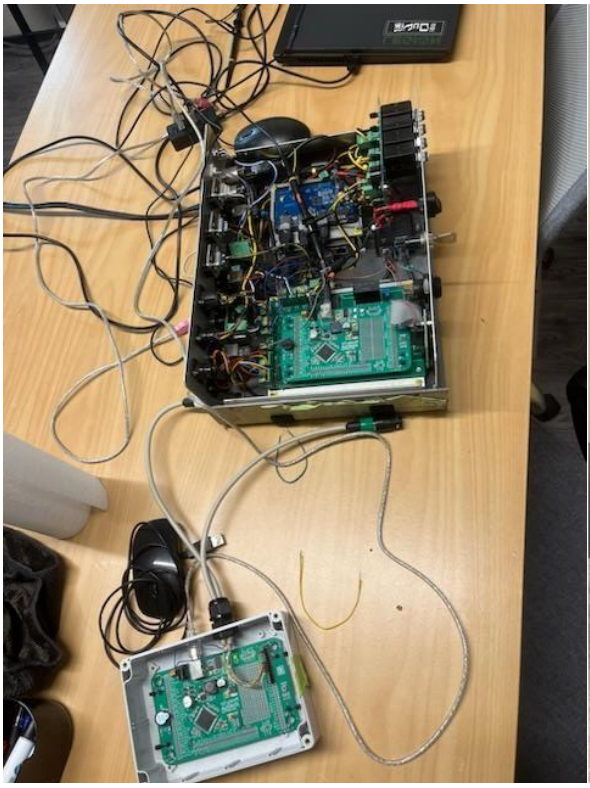
In the aluminium structure project, around 160 teams of two competed to design a structure that had to break at a specified calculated weight between 20 and 39 kg. My partner and I decided to use a 2-member structure to reduce the weight as strength to weight ratio was one of the things points were awarded for. As seen in the photo the structure has 3 pin joints however, the top support is an axially constrained roller bearing which had to remain within ± 3 mm of the specified dimensions, thus the structure also had to support its self-weight. Going with a 2-member structure made the calculations and analysis significantly harder as the system became very indeterminate. It also meant we had to be much more innovative with our design. I proposed using a rectangular beam for our compressive member made up of two pieces bent at right angles with teeth cut out of them so that they would slot (and glue) together nicely. In the end our design broke at the target weight, which was 36 kg and weighed only 52 g. We were the only successful 2-member structure, the only people to reach the highest strength to weight ration which was 550+ (ours was 692.3) and therefore the only people to get a perfect score on the project. There were 6 awards given out and we won four of them (the two we didn't win were for the most catastrophic failure and for the heaviest design). Overall, this project was super satisfying as we took an extra risk compared to our peers, put in the extra work and it paid off!

I also completed a cool project in which myself and two others developed the required circuitry for an analogue motor driver. This was used on a small remote-control cart that was powered by a solar panel. The circuit needed to contain a buck converter, a push/pull emitter follower and a closed-loop feedback control system. The inductor required for the buck converter was calculated, designed and made to match the calculations. This circuitry meant that the output current and voltage were adjusted so that the maximum possible power was always seen at the motor – this is essential for operating under varying loads and the motors characteristics. We also learnt how to use LTSpice for design and analysis. A photo of the LTSpice model and the actual circuit are shown below.





Another project I was involved with was during my internship last summer. I was tasked with redesigning the circuitry for one of the daughter boards used in a modular road surveying system. This involved using Kicad and knowledge of circuit design to restructure the current design and fix a few issues the company was having with the board. I was also tasked with identifying why the high-accuracy odometer, attached to the survey vehicles wheel to trigger data collection, was jittering and then provide a solution to stop this from happening. Finally, I was tasked with setting up an NFS server so that one of the boards used for development could be programmed from any PC in the office without any physical connection between them. This was awesome as the work that I was responsible for had a direct effect on the company and system thus, I could feel like the work I was doing was valuable. A few photos of this are shown below.

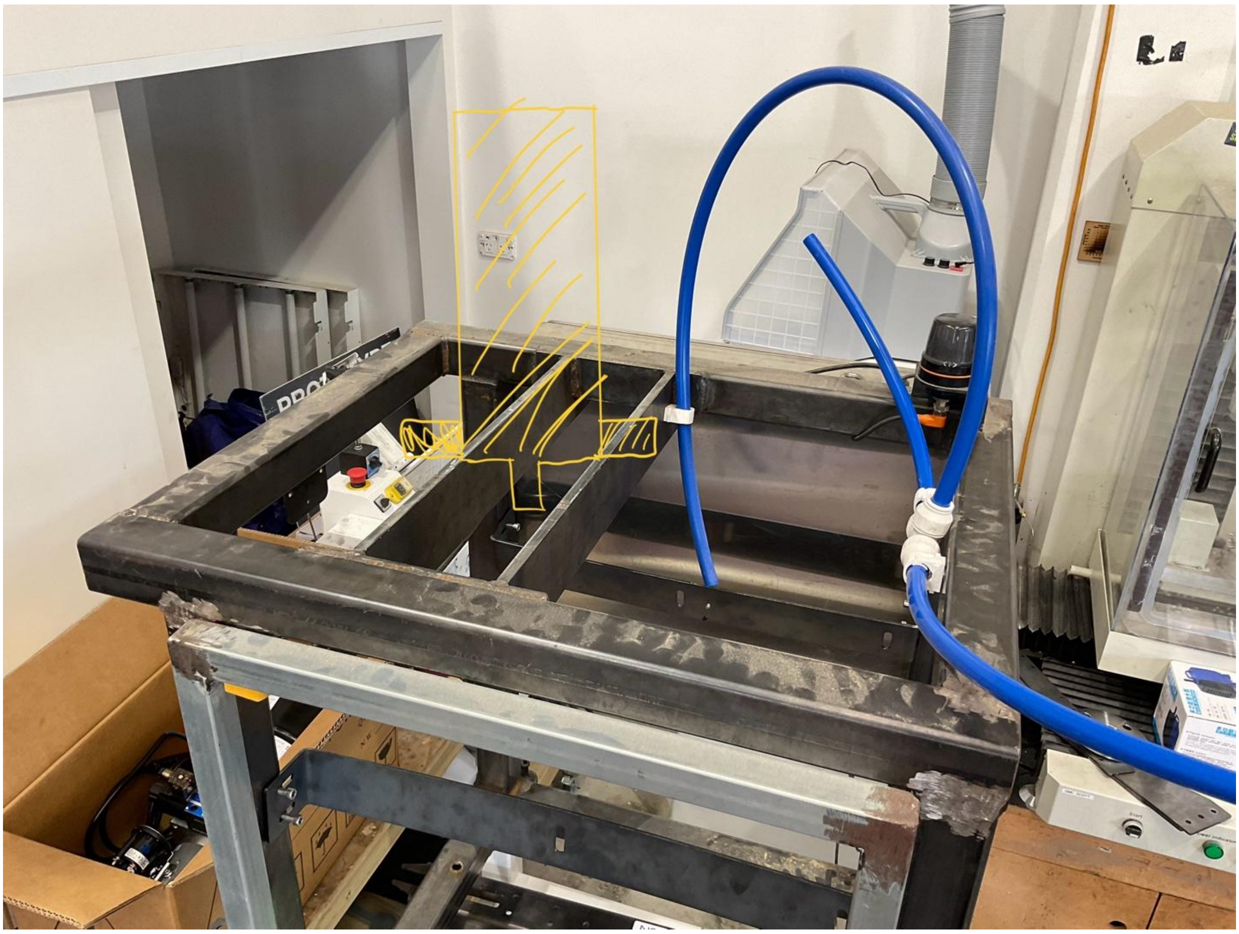


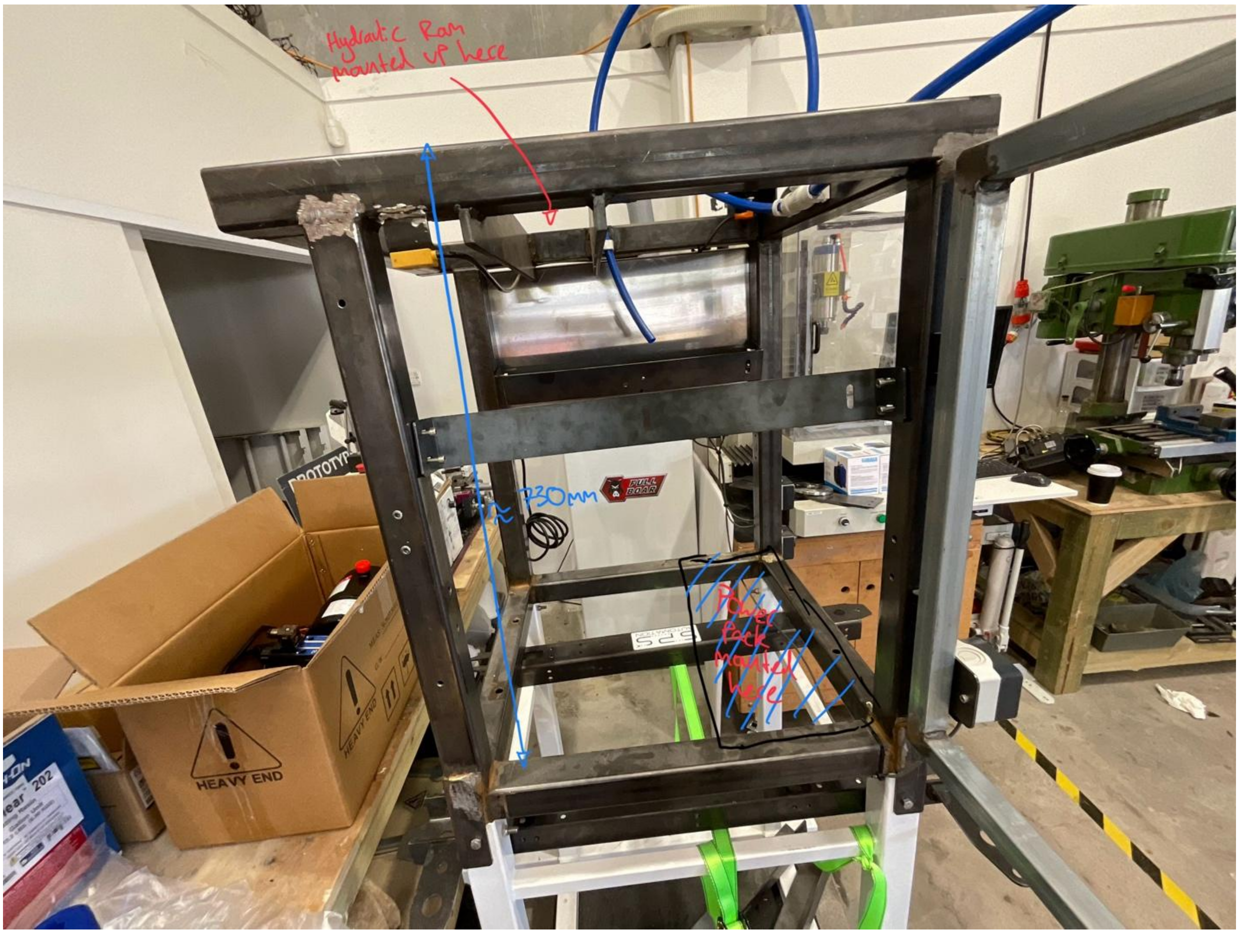
My latest project has been designing and building a small-scale injection molder for my summer internship with SPS Automation.

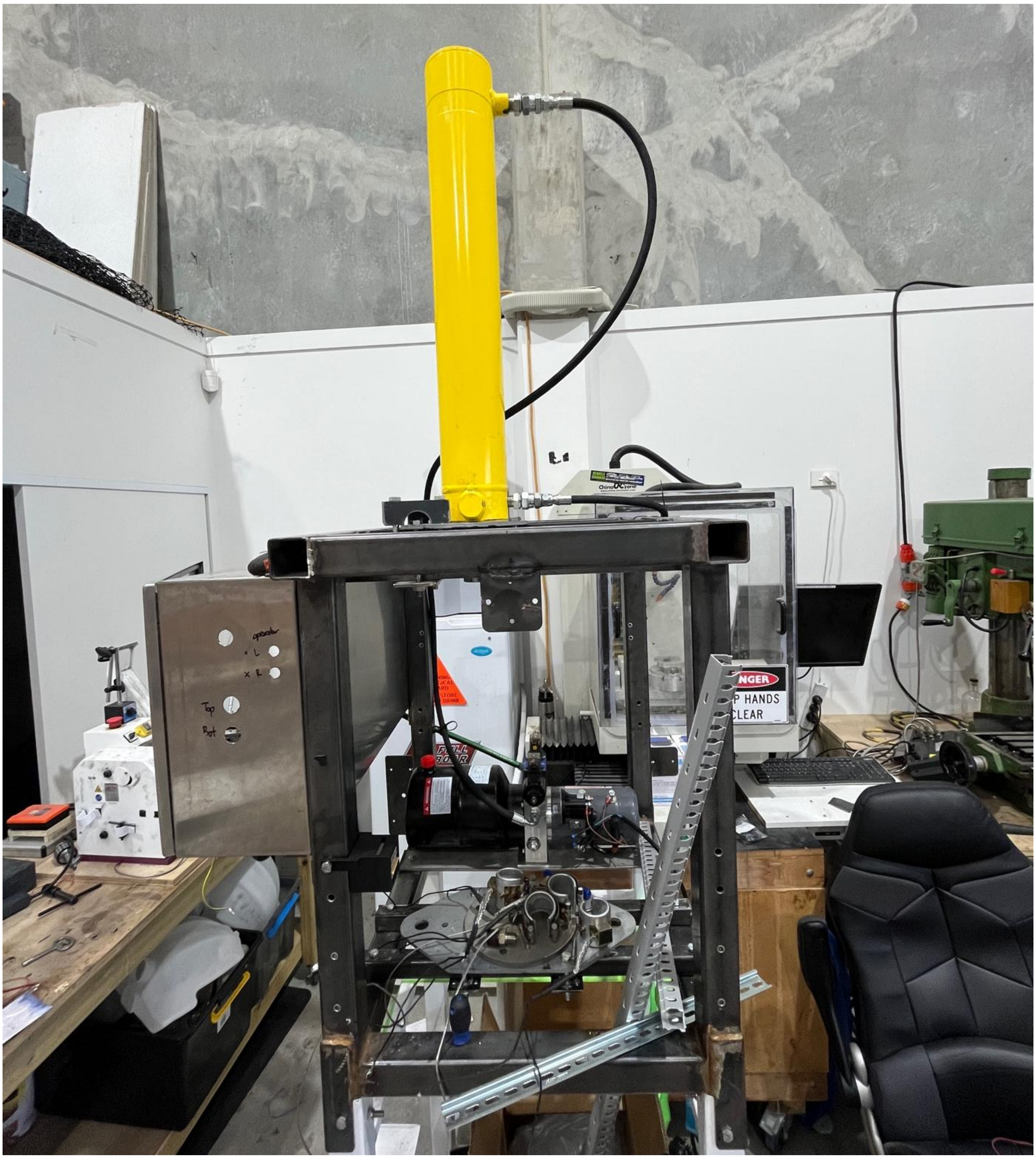
This was completed the summer before my final year of engineering and involved myself and one other intern completing the entire engineering process on our own project. It felt great to have control and freedom in the project over things such as how we wanted the design to work, how much it should cost, when each part of the project should be completed, what sort of (safety) features it should have and how it should be powered. I think getting used to thinking about these things is an invaluable lesson that I have taken from the project. A few photos from the setup are shown below.

As can be seen, the frame was mostly made up of welded steel frame section meaning that I also got to learn how to TIG weld a bit better which was fun.


SMOOTH-ON
Crystal Clear 202
Make It Now!
POWER UNIT
HEAVY END
HEAVY END
FULL ROAR



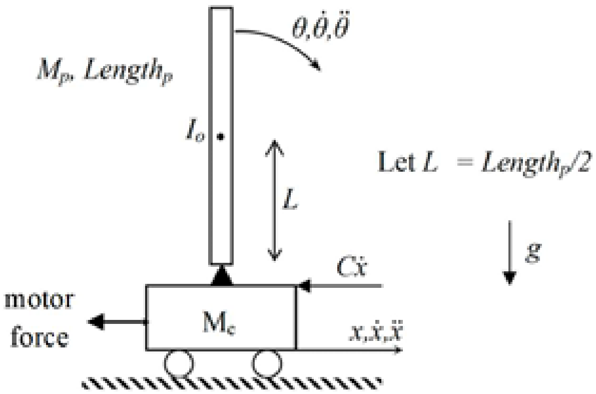

Hydraulic Ram
mounted up here
~730mm
Power
Pack
mounted
here
HEAVY END

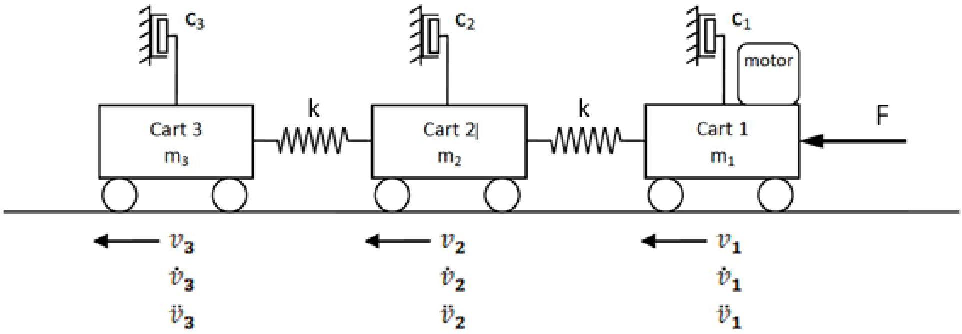

Top
Rot
CLEAR

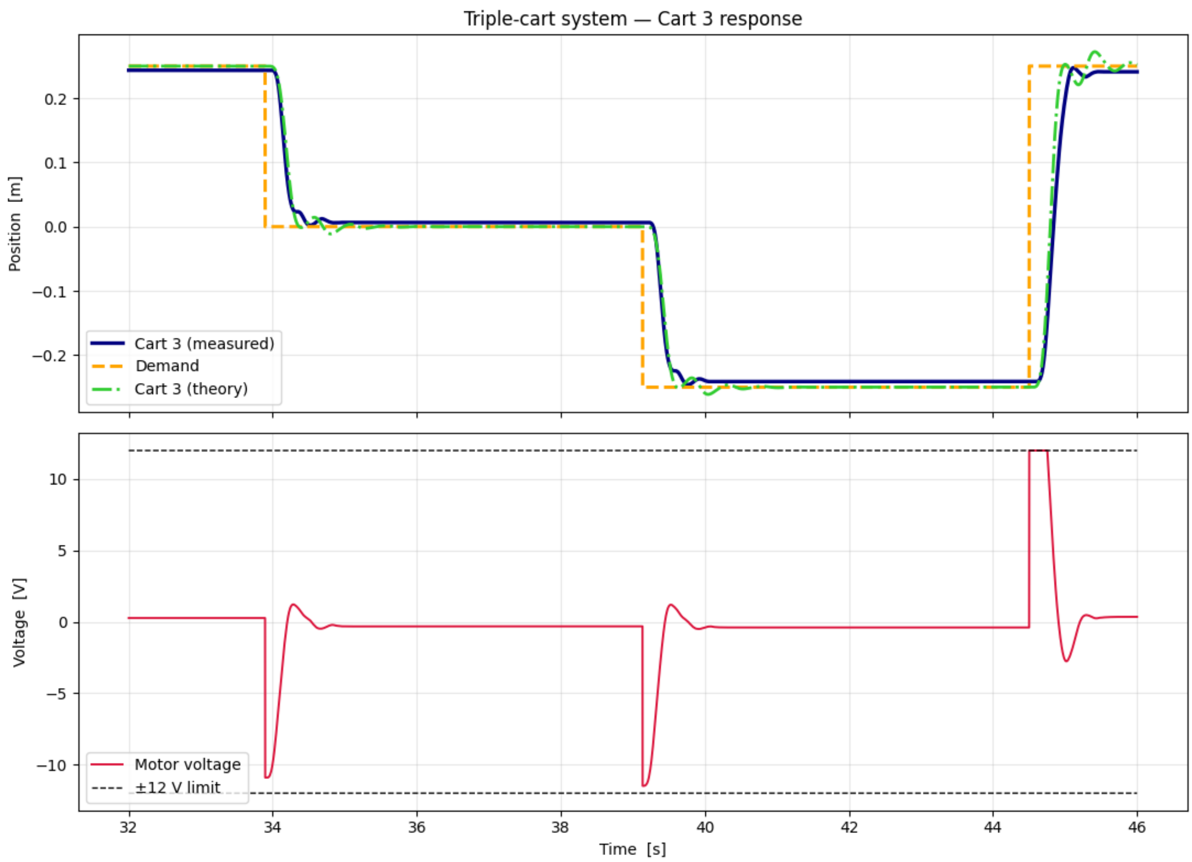
Band heaters were used in combination with a custom machined 316 stainless steel heating chamber. A 316 stainless steel ramrod was also used with a very fine tolerance used on the last 50 mm to ensure it made a good seal with the chamber. A Bucher hydraulic power pack and 20 Ton hydraulic ram were used to get the driving force required to squirt out the molten plastic.

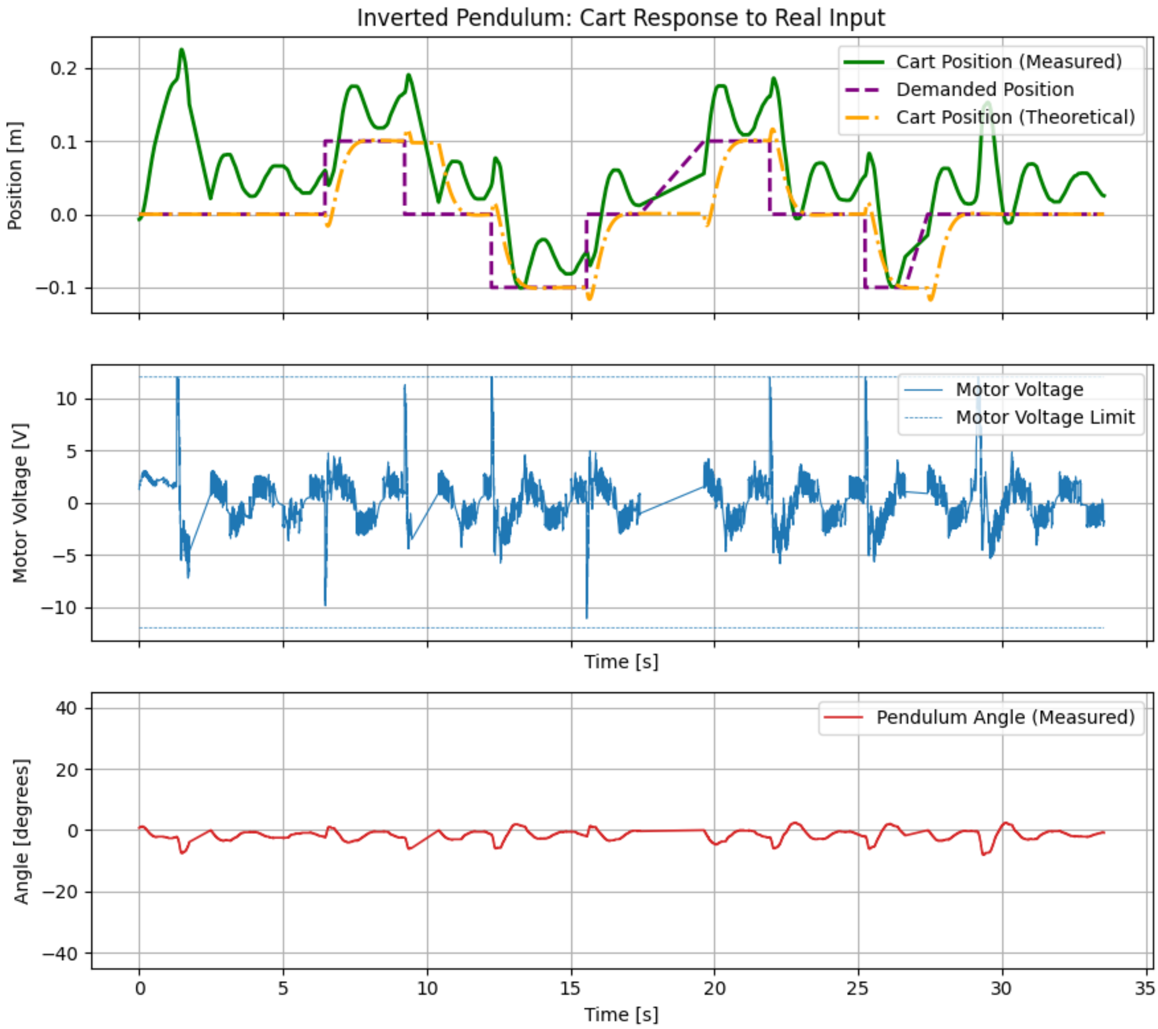
Since my summer internship at SPS Automation I have completed various interesting university projects. The first involved writing python scripts to implement control systems needed to stabilize an inverted pendulum and track and control the motion of a triple cart

SMD system. Diagrams of the physical test systems as well as some plots showing the responses achieved using the linear quadratic regulator (LQR) method are seen below.









In a second project, I created a Neural Attenuation Field (NAF) neural network used for 3D image reconstruction of X-rays. This involved using known geometries from the cone-beam CT (CBCT) scan data to cast rays from the X-ray source through each pixel on the detector plane. The rays were discretized into a sequence of sample points which could then be passed through a positional encoding function. The encoded points are then passed through our network which predicts a scalar attenuation coefficient for each point, the distance between the sampled points as well as the attenuated values were then used to compute the overall attenuation along a given ray using a discretized form of the Beer-Lambert law.

The resulting attenuation values were used to generate a synthetic projection image which could be compared against the true projection image at that point. From this a loss is calculated and used to update the network with standard backpropagation and optimization. This process is repeated until the network has a reasonable output. At this point the network represents a model of attenuation throughout the given 3D volume and can be queried to generate arbitrary slices of the volume. We were tasked with adding an additional more complex feature to the architecture of the network. For this I attempted to implement multi-headed self-attention (MHSA) with residual connection as well as convolution in two separate models. The images below show an overview of the setup provided in the assignment brief as well as some code snippets and results.

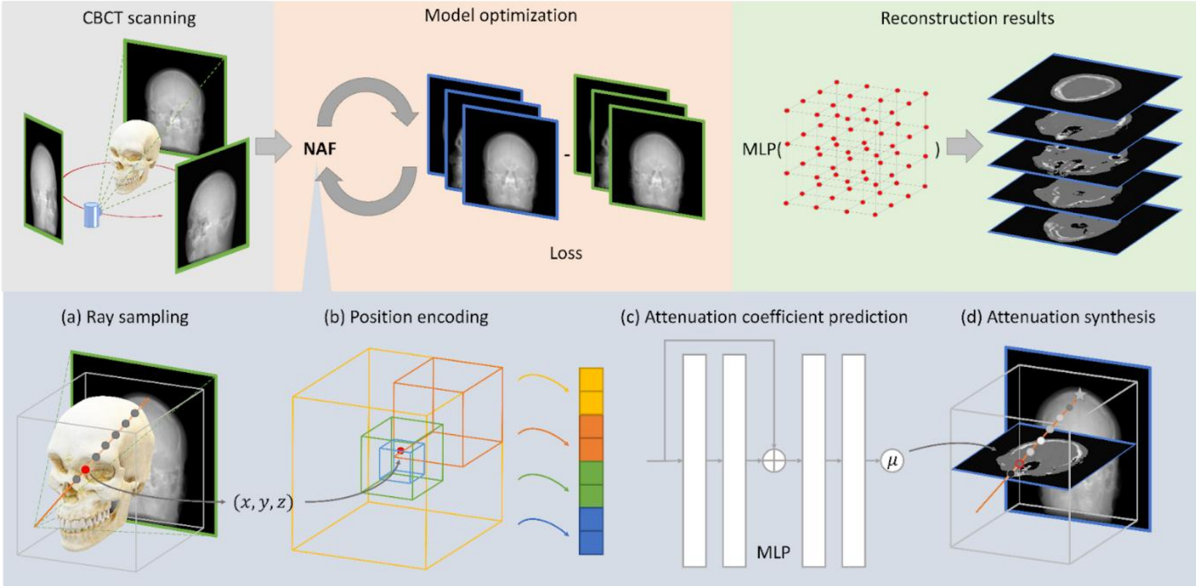


Figure 1: A overview image taken from the project brief credit to COSC440 and James Atlas.

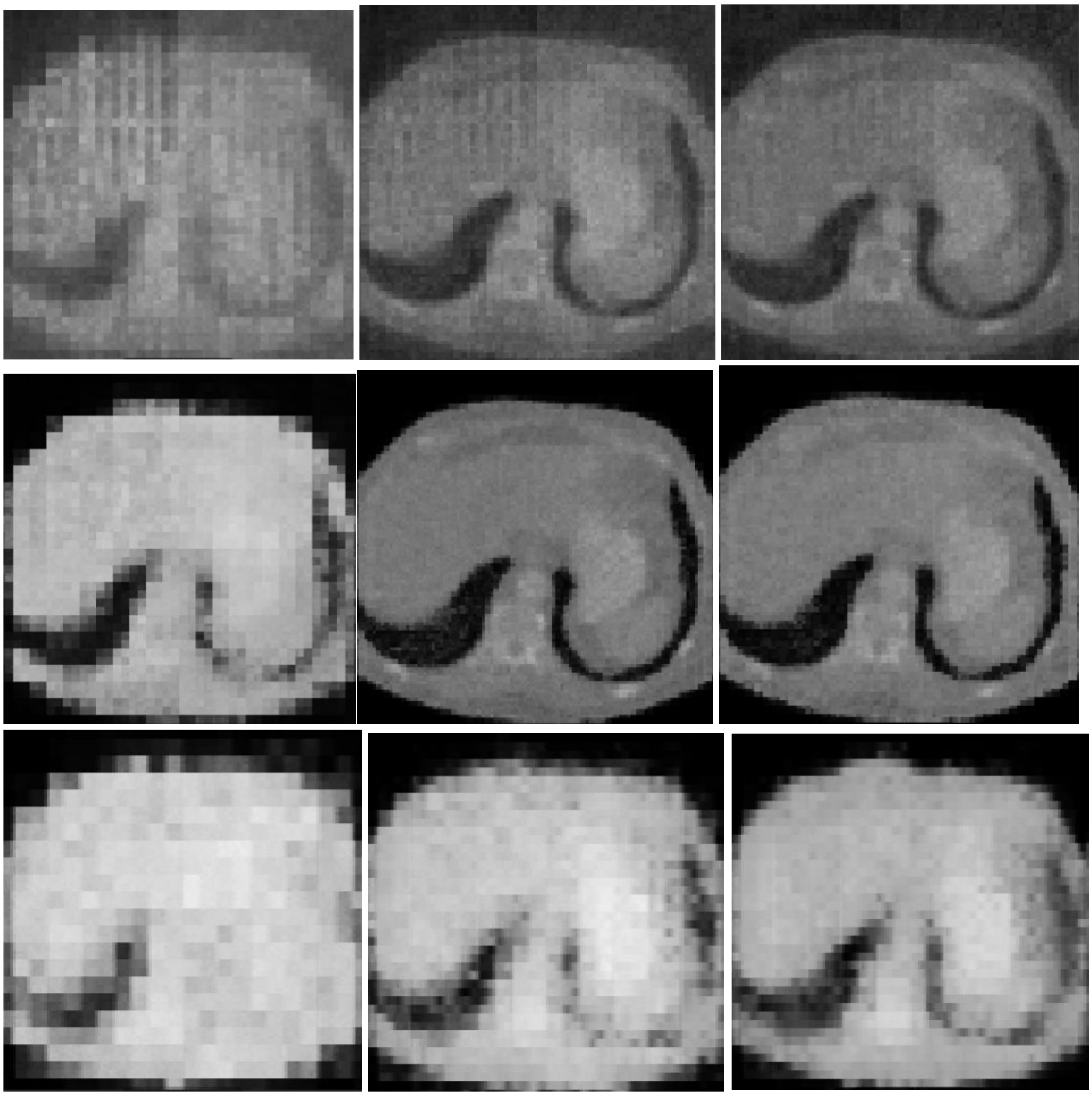


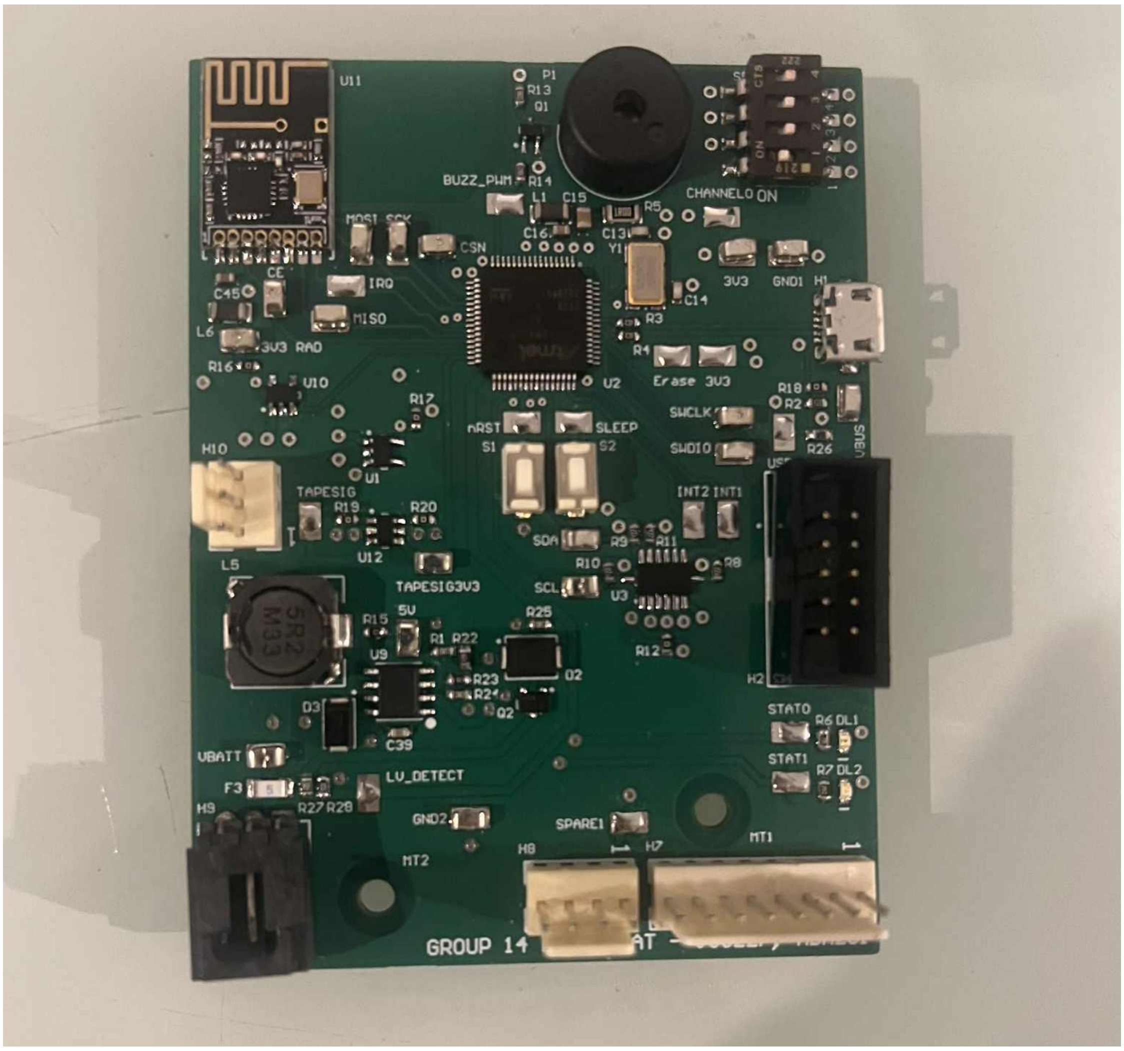
Figure 2: Reconstructions of the chest data set (birds eye view) output by the residual + MHSA modified model (top), the original model (middle), and the convolution modified model at different epochs.

model_attn_soft.py > ModelAttnSoft > __init__

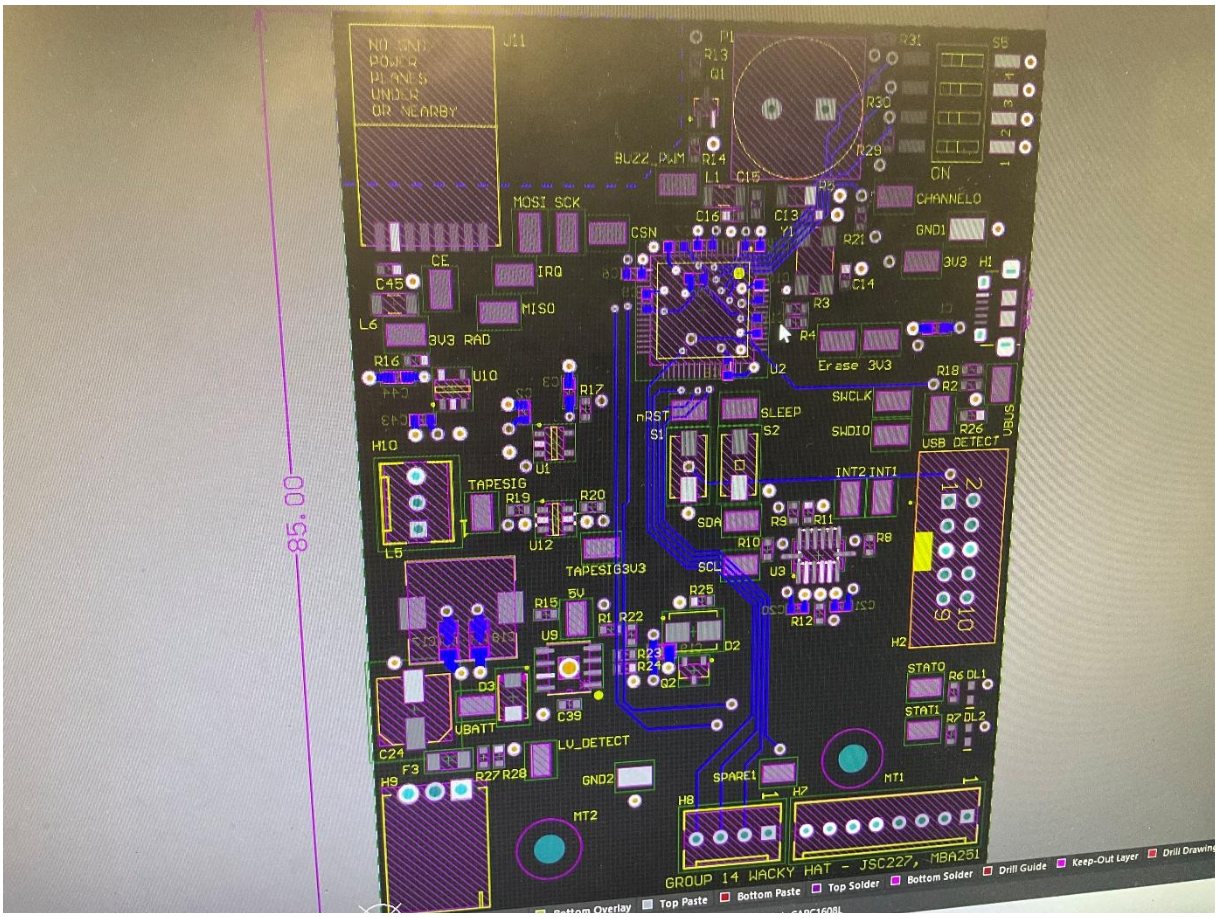
```python
import tensorflow as tf


class ResBlock(tf.keras.layers.Layer):
    def __init__(self, hidden):
        super().__init__()
        self.d1 = tf.keras.layers.Dense(hidden, activation="linear")
        self.act = tf.keras.layers.LeakyReLU(0.2)
        self.d2 = tf.keras.layers.Dense(hidden, activation="linear")

    def call(self, x):
        h = self.act(self.d1(x))
        h = self.d2(h)
        return x + self.act(h)                    # residual


class ModelAttnSoft(tf.keras.layers.Layer):
    def __init__(self, encoder,
                 hidden=64,
                 num_blocks=3,
                 n_points=192):
        super().__init__()
        self.encoder   = encoder
        self.n_points  = n_points
        self.hidden    = hidden

        self.fc_in     = tf.keras.layers.Dense(hidden)

        # attention block
        self.ln_attn   = tf.keras.layers.LayerNormalization(epsilon=1e-5)
        self.attn      = tf.keras.layers.MultiHeadAttention(
                             num_heads=2, key_dim=hidden // 2)

        self.blocks    = [ResBlock(hidden) for _ in range(num_blocks)]


        self.fc_out    = tf.keras.layers.Dense(1, activation=tf.nn.softplus)

    def call(self, pts):
        """
        pts: [N, 3]
        """
        x = self.encoder(pts)
        x = self.fc_in(x)


        N = tf.shape(x)[0]
        if tf.math.equal(tf.math.floormod(N, self.n_points), 0):
            B = N // self.n_points
            x3 = tf.reshape(x, (B, self.n_points, self.hidden))

            attn_out = self.attn(self.ln_attn(x3), self.ln_attn(x3))
            x3 = x3 + attn_out
            x  = tf.reshape(x3, (-1, self.hidden))

        for blk in self.blocks:
            x = blk(x)

        return self.fc_out(x)
```

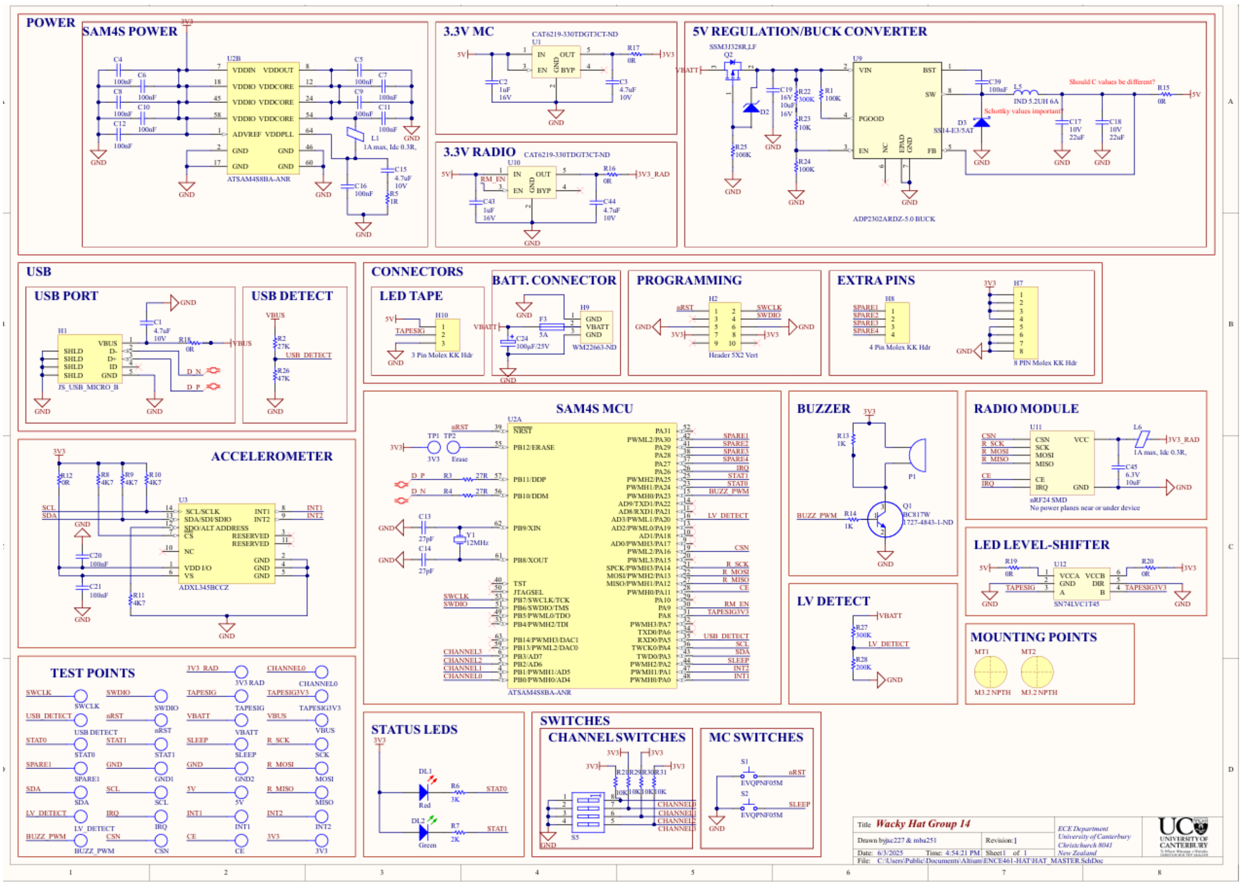
A third project completed is known as the Wacky Races. In this project groups of four students produce a small scale radio controlled car and a corresponding controller which uses an accelerometer to determine the orientation of the controller and convert this into left and right duty cycles to drive the car in this direction. The controller is mounted to a hat so that the driver must steer with their head. Students are also encouraged to dress up and add “dastardly stuff” to their designs. The groups of four split into pairs with one pair responsible for the car and one responsible for the hat, I worked on the hat.

The project started with a full design of the PCB schematic and physical layout using Altium. Once received the boards were then assembled using the university pick-and-place machines and oven. The boards then had to be boot-flashed and code was developed for reading the accelerometer data, transmitting duty cycles via radio and many other little things such as playing interesting noises from a buzzer on the hat when a bumper on the car is hit. Below are some photos of our hat PCB and schematic as well as my team on race day and me driving the car.



U11
P1
R13
Q1
BUZZ_PWM
R14
L1
C15
R5
CHANNEL0
ON
MOSI
SCK
CSN
C16
C13
Y1
3V3
GND1
H1
CE
IRQ
C45
MISO
C14
R3
L6
3V3 RAD
R4
Erase
3V3
R16
U10
U2
R18
R2
R17
nRST
SLEEP
SWCLK
S1
S2
SWDIO
R26
VBUS
H10
U1
INT2
INT1
TAPESIG
R19
R20
SDA
R9
R11
U12
R10
R8
TAPESIG3V3
SCL
U3
L5
R15
5V
R25
R1
R22
R12
U9
R23
O2
H2
R24
Q2
D3
STAT0
R6
DL1
C39
VBATT
STAT1
R7
DL2
F3
LV_DETECT
R27
R28
H9
GND2
SPARE1
H8
H7
MT1
MT2
GROUP 14


NO GND POWER PLANES UNDER OR NEARBY
85.00
GROUP 14 WACKY HAT - JSC227, MBA251
Bottom Overlay
Top Paste
Bottom Paste
Top Solder
Bottom Solder
Drill Guide
Keep-Out Layer
Drill Drawing


POWER
SAM4S POWER
3.3V MC
5V REGULATION/BUCK CONVERTER
3.3V RADIO
USB
USB PORT
USB DETECT
CONNECTORS
LED TAPE
BATT. CONNECTOR
PROGRAMMING
EXTRA PINS
ACCELEROMETER
SAM4S MCU
BUZZER
RADIO MODULE
LED LEVEL-SHIFTER
LV DETECT
MOUNTING POINTS
TEST POINTS
STATUS LEDS
SWITCHES
CHANNEL SWITCHES
MC SWITCHES
Should C values be different?
Schottky values important!
No power planes near or under device
Title Wacky Hat Group 14
Drawn byjsc227 & mba251
ECE Department
University of Canterbury
Christchurch 8041
New Zealand

My latest and greatest project has been my final year project which I am completing on the University of Canterbury Motorsport (UCM) team. My role in the team this year is on the powertrain sub-team. I am responsible for the complete design and manufacture of our power transmission from motors to wheels. This includes gears, gearbox assembly and uprights (housing), motor mounting, spindle (interface to wheel). Some of the things I have been doing for this are:

- Research into gear theory
- Track days where we test on 2024's car, maintenance of the 2024 gearbox e.g. disassembly to ensure parts are still ok, oil changes etc.
- Writing python scripts for; determining valid gear configurations, optimal gear ratio (based on simulation of acceleration event at competition), gear analysis to check all parameters of gearbox design are suitable e.g. adequate backlash, centre distance spacing of planets for correct meshing, required profile shifts to ensure no undercutting of teeth.
- Manufacture of gears – this is done using an EDM wire-cut machine. 2024 was the first year that we produced our gears this way and it took around 3 months as the workshop staff were not trained on how to use the machine. This year I taught myself to use the machine and got permission to operate it myself allowing me to complete the gears in under two weeks. I also figured out how to directly import DXF's into the machine increasing the freedom of design as I can use the exact tooth profile that I designed whereas in 2024 they used an embedded gear generating tool on the machine which is less capable and does not allow for fine tuning of as many parameters.

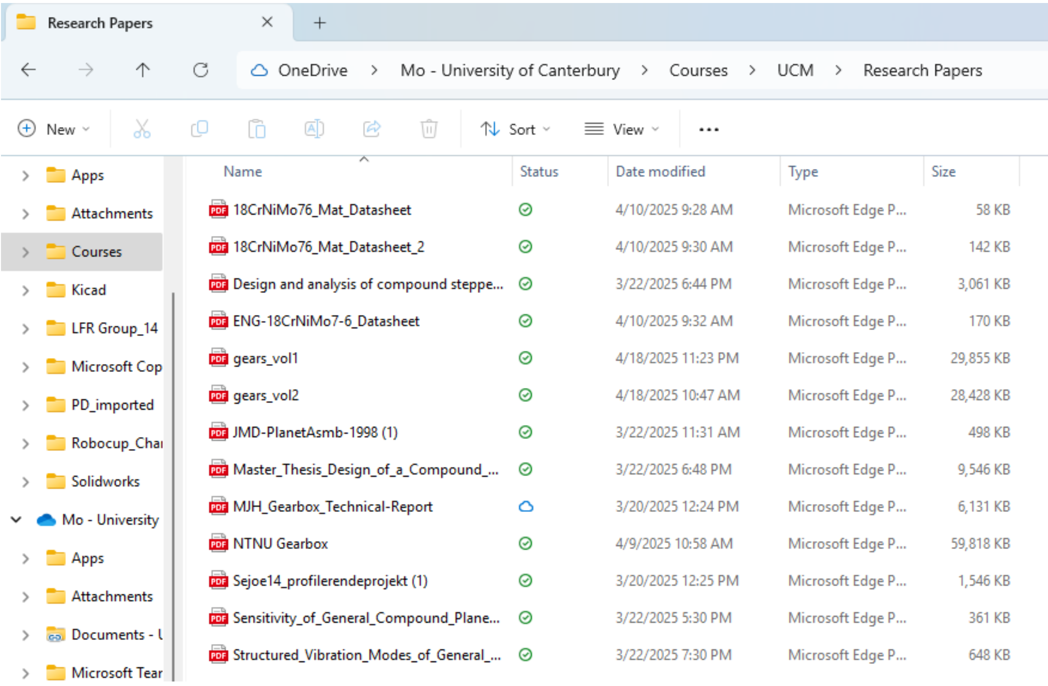
- Analysis of gear pairs and overall gearbox using KISSsoft and KISSsys to determine FOS's, lifetime, forces on selected bearings, failure modes etc (includes dynamic simulation of geartrain).
- CAD models of gears, uprights, spindles, full assembly as well as engineering drawings for manufacture and sponsorship enquiries.
- Sponsorship requests and organisation of manufacture and processing e.g. heat treatment, material orders. I secured a fully sponsorship for isotropic finishing of the gears which should improve their lifetime significantly.
- Discussions with members of industry about technical design factors e.g. titanium 3D printing, gear material selection.
- Investigation of forces and load cases around rear uprights.
- Topology optimisation of titanium 3D printed uprights using nTop (software). As well as all the necessary post-processing to produce a valid, printable CAD part.

I am incredibly stoked on this project and love the feeling of working in a team with an overall goal in mind and deadlines to meet (although they are stressful!). Due to this I have been set on making the most out of my time working on the project from the beginning. This means I have been taking any opportunity I can to learn or do more and be involved. Some of the things I've done include:

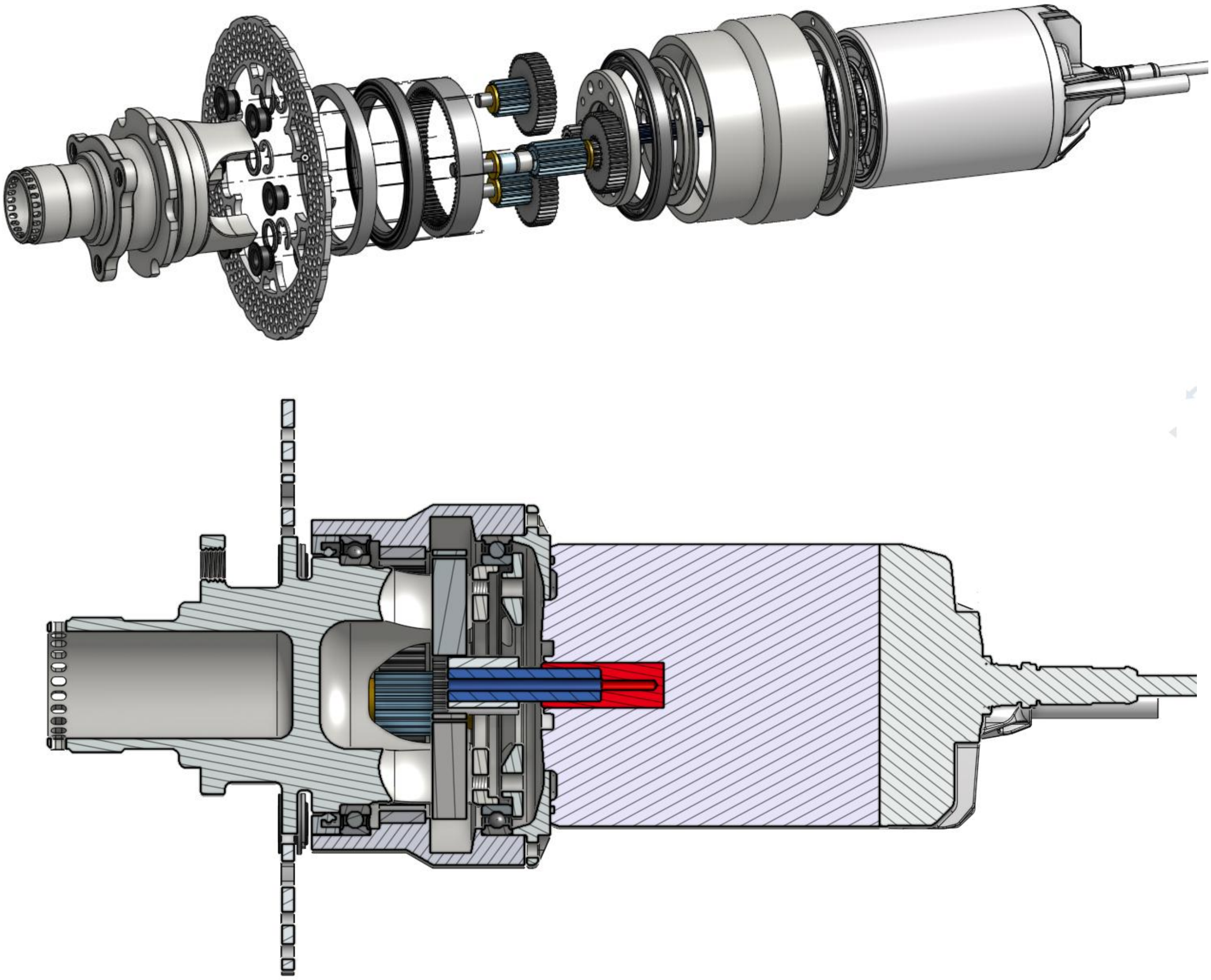
- Helping with the chassis manufacture by priming and sanding plugs, helping with composites e.g. doing infusions as well as wet lays for the carbon fibre chassis.
- Getting involved in any extra-curricular club activities I can e.g. attending SouthMach, Two Rivers Motoring Extravaganza, E-Velocity and club information events.
- Setting up a minute taking system and taking minutes for all UCM full team and powertrain meetings as previously there was none.
- I will be designing and manufacturing the gearbox for UCM2026 later this year once 2025's is completed. This is atypical and is only happening because I am very keen to do it. It is also essential as the car is going to be four-wheel drive next year meaning six full gear sets will need to be produced (four plus two spare sets), the front upright (which will now have to house gearboxes) will need to be redesigned and optimised to allow room for the steering, and we will be advancing to an entirely different gearbox design which removes the need for a spindle and interfaces directly with custom carbon fibre wheels. There would not be enough time to go ahead with these changes if it is not started this year.

Below are some photos of the things mentioned above.

A photo of UCM24's spare ring gear that has been superfinished (isotropic finishing) and has since had a StealthCoat (TM) (Tungsten Disulphide) coating added to it. A process that I was able to get as a sponsorship deal for all UCM25's gears.

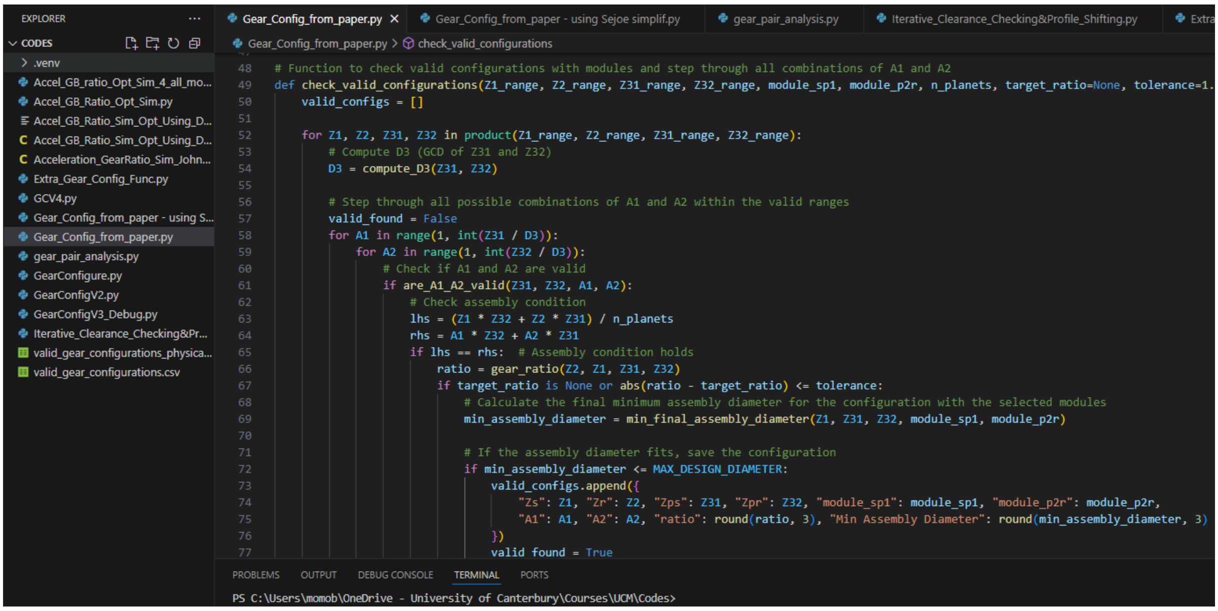


A screenshot showing a small collection of the research papers that I have used during my research and design.

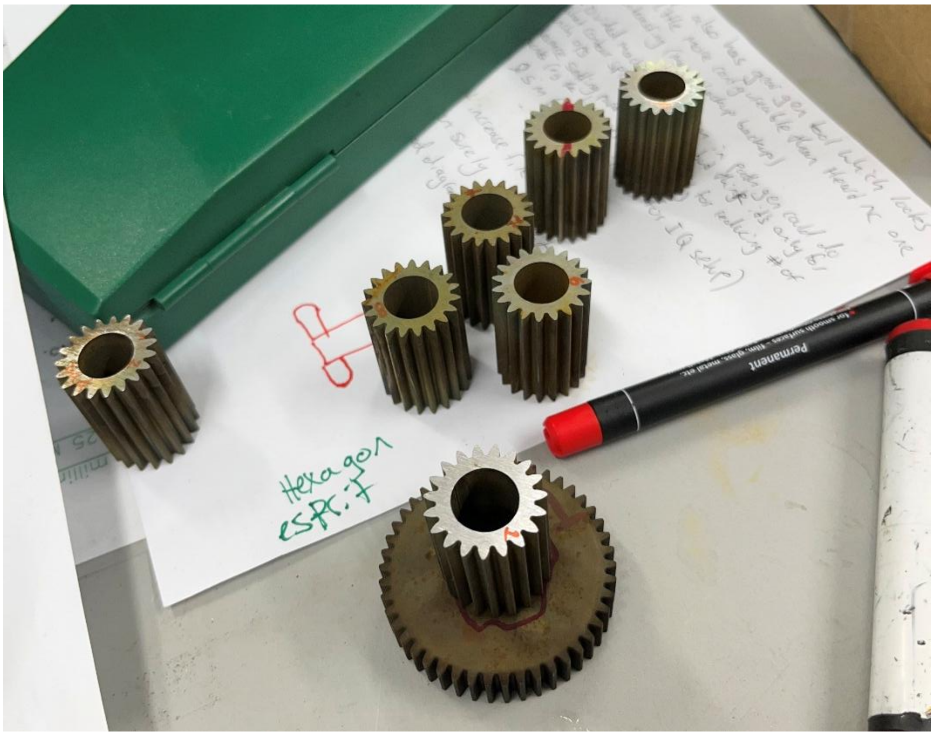


Exploded and section views of gearbox assembly.

```
Gear_Config_from_paper - using Sejoe simplif.py"
    Zs  Zr  Zps  Zpr  module_sp1  module_p2r  A1  A2   ratio  Min Assembly Diameter
0   15  75   41   19           1           1  46   6  11.789                   97.0
1   15  81   43   23           1           1  48   4  11.096                  101.0
2   15  87   47   25           1           1  52   4  11.904                  109.0
3   16  79   43   20           1           1  34  13  11.616                  102.0
4   16  85   47   22           1           1  21  21  12.349                  110.0
5   16  86   47   23           1           1  21  21  11.984                  110.0
6   16  88   47   25           1           1  21  21  11.340                  110.0
7   16  89   47   26           1           1  21  21  11.055                  110.0
8   17  79   43   19           1           1  63   1  11.517                  103.0
9   17  80   44   19           1           1  35  14  11.898                  105.0
10  17  82   46   19           1           1  67   2  12.678                  109.0
11  17  85   46   22           1           1  44  10  11.455                  109.0
Required profile shift for z teeth: 0.016633181894903332
Computed center distances:
Zr: 82.000 mm
a_sun_planet: 31.500 mm
a_planet_ring: 31.500 mm
a_total: 31.500 mm
```

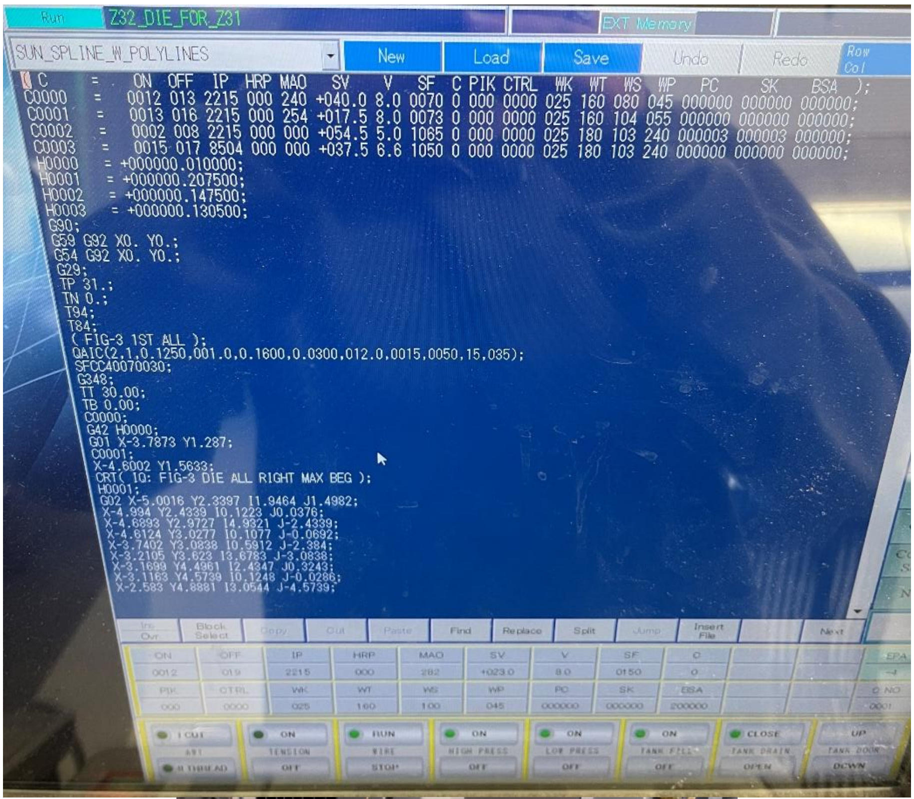


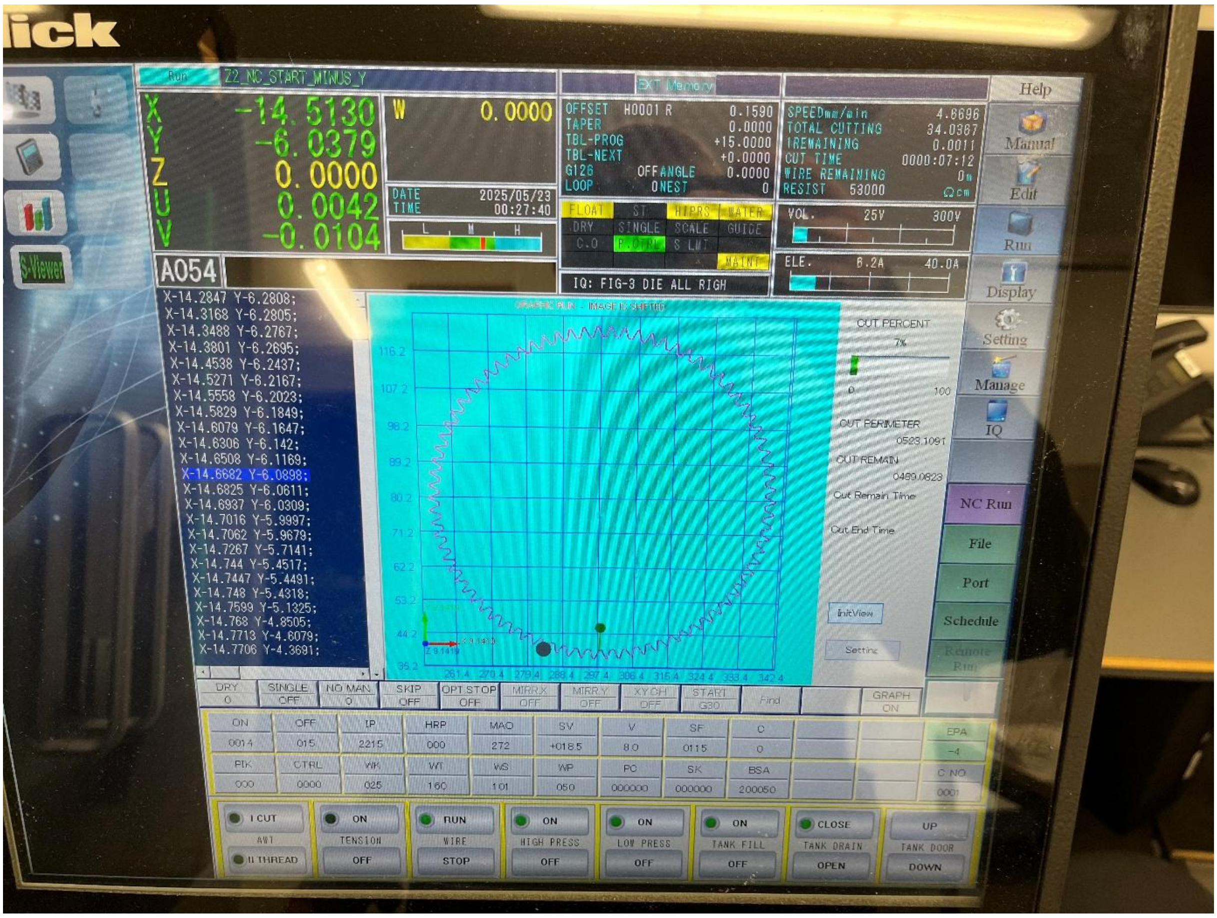
Screenshots showing the output of my configuration finding script and my UCM codes folder with the files inside it, these files include an acceleration sim of the car to determine optimal gear ratio, a code that applies mathematical constraints found during my research that ensure valid configurations of numbers of teeth and modules etc. as well as a full gear analysis program for analysing gear pairs, their parameters and their interfacing.


Hexagon
Permanent


ALC600G
Motor Drive
e-cut EDM
Sodick

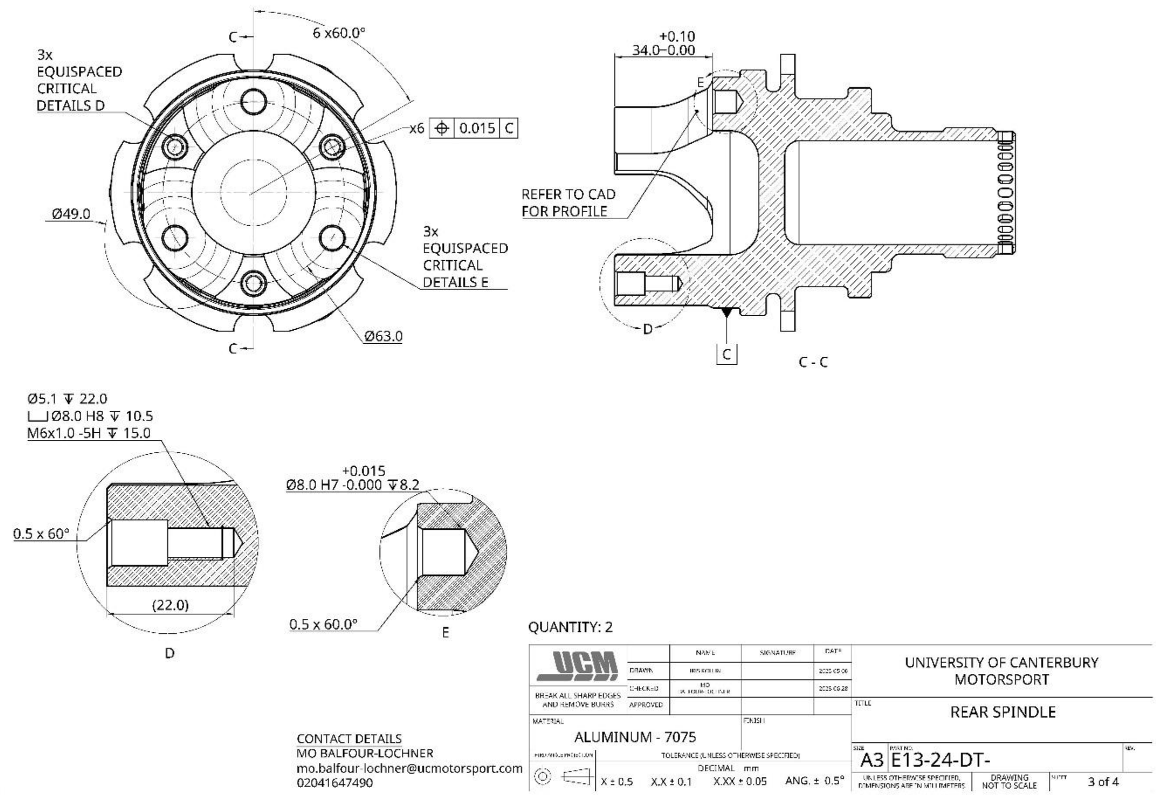
Photos showing some of the small planet gears that I cut (top) using the wire cut machine pictured (middle) and a photo (bottom) showing this year’s test stepped-planet compound planetary gear set (foreground) and 2024's test set.


Run
Z32_DIE_FOR_Z31
EXT Memory
SUN_SPLINE_W_POLYLINES
New
Load
Save
Undo
Redo
Row Col
C = ON OFF IP HRP MAO SV V SF C PIK CTRL WK WT WS WP PC SK BSA);
C0000 = 0012 013 2215 000 240 +040.0 8.0 0070 0 000 0000 025 160 080 045 000000 000000 000000;
C0001 = 0013 016 2215 000 254 +017.5 8.0 0073 0 000 0000 025 160 104 055 000000 000000 000000;
C0002 = 0002 008 2215 000 000 +054.5 5.0 1065 0 000 0000 025 180 103 240 000003 000003 000000;
C0003 = 0015 017 8504 000 000 +037.5 6.6 1050 0 000 0000 025 180 103 240 000000 000000 000000;
H0000 = +000000.010000;
H0001 = +000000.207500;
H0002 = +000000.147500;
H0003 = +000000.130500;
G90;
G59 G92 X0. Y0.;
G54 G92 X0. Y0.;
G29;
TP 31.;
TN 0.;
T94;
T84;
(FIG-3 1ST ALL);
QAIC(2,1,0.1250,001.0,0.1600,0.0300,012.0,0015,0050,15,035);
SFCC40070030;
G348;
TT 30.00;
TB 0.00;
C0000;
G42 H0000;
G01 X-3.7873 Y1.287;
C0001;
X-4.6002 Y1.5633;
CRT(IQ: FIG-3 DIE ALL RIGHT MAX BEG);
H0001;
G02 X-5.0016 Y2.3397 I1.9464 J1.4982;
X-4.994 Y2.4339 I0.1223 J0.0376;
X-4.6893 Y2.9727 I4.9321 J-2.4339;
X-4.6124 Y3.0277 I0.1077 J-0.0692;
X-3.7402 Y3.0838 I0.5912 J-2.384;
X-3.2105 Y3.623 I3.6783 J-3.0838;
X-3.1699 Y4.4961 I2.4347 J0.3243;
X-3.1163 Y4.5739 I0.1248 J-0.0286;
X-2.583 Y4.8881 I3.0544 J-4.5739;
Find
Replace
Split
Insert File
Next

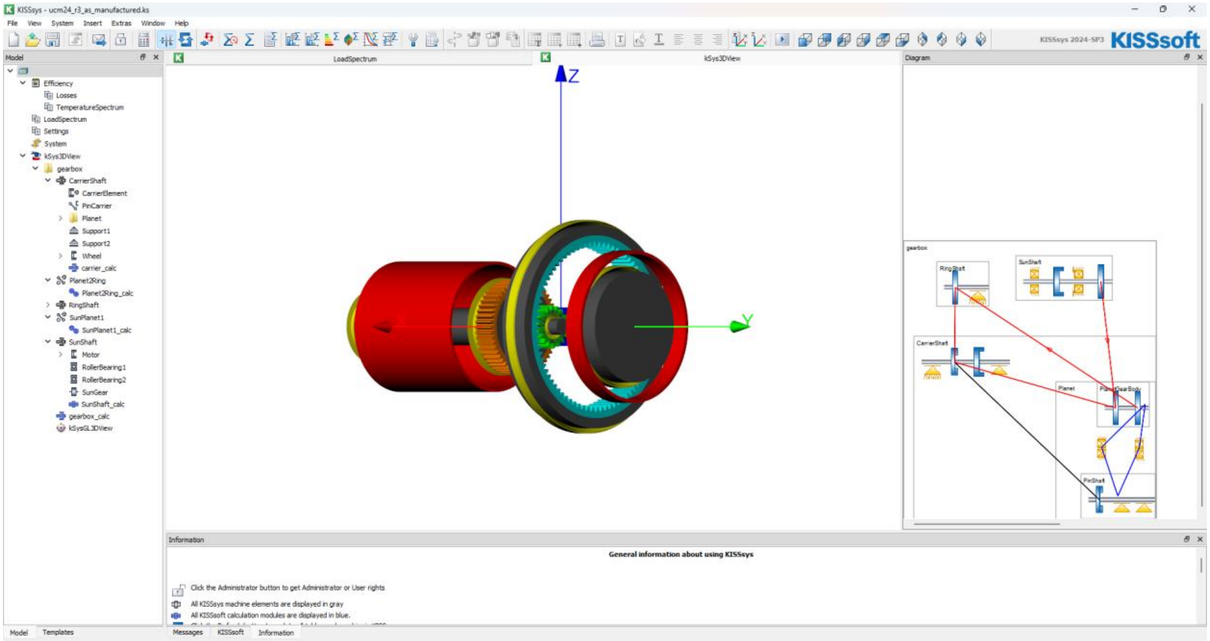


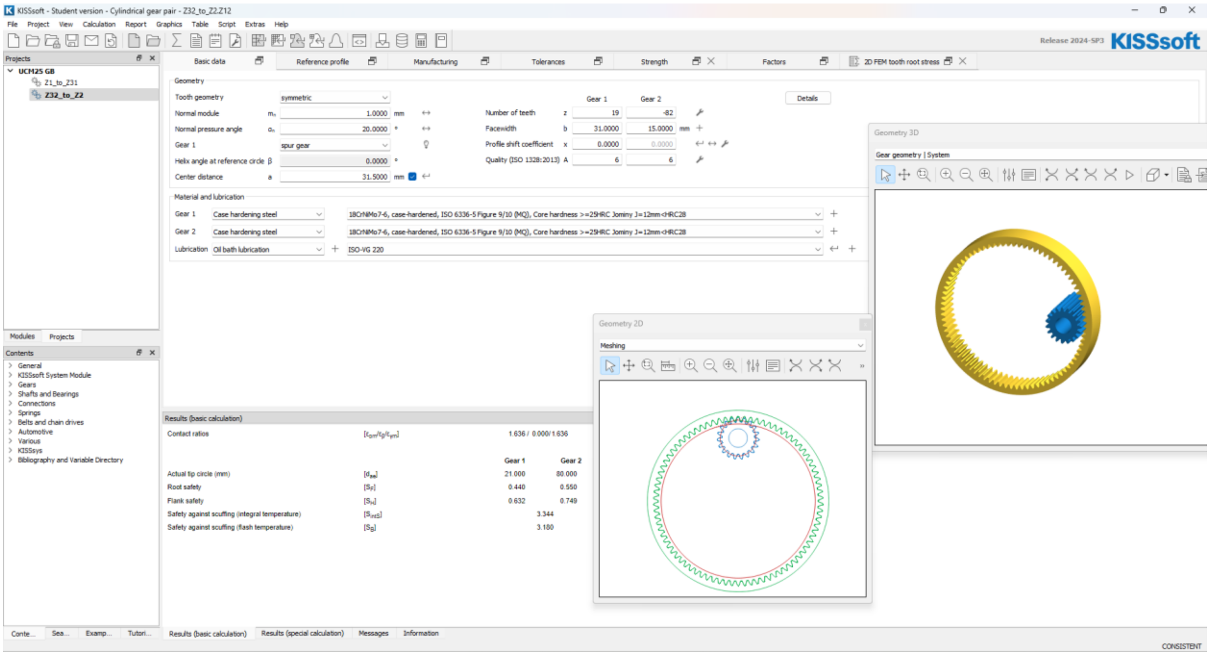
Some photos of machining paths and the corresponding G-Code I generated from my imported gear models to drive the wire cut machine as well as a photo of one of my stock pieces in the machine.

Photos of me suited up when prepping the plugs that are used for making the chassis molds (left) and a finished layup of the bulkhead ready for infusion (right).



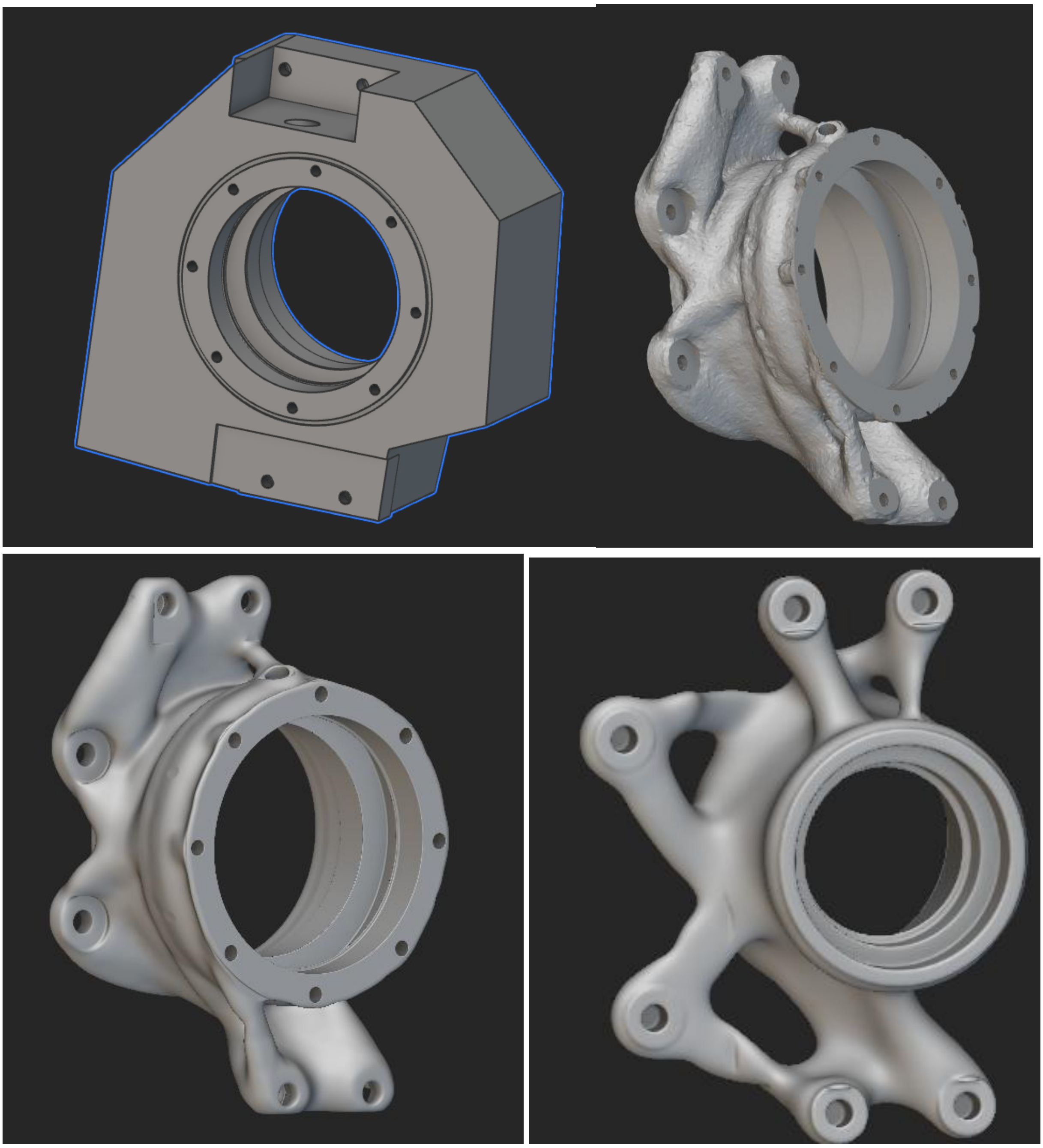
Photos showing a drawing I produced for the rear spindle followed by the finished physical product.


KISSsys - ucm24_r3_as_manufactured.ks
File View System Insert Extras Window Help
KISSsys 2024-SP3 KISSsoft
Model
Efficiency
Losses
TemperatureSpectrum
LoadSpectrum
Settings
System
kSys3DView
gearbox
CarrierShaft
CarrierElement
PinCarrier
Planet
Support1
Support2
Wheel
carrier_calc
Planet2Ring
Planet2Ring_calc
RingShaft
SunPlanet1
SunPlanet1_calc
SunShaft
Motor
RollerBearing1
RollerBearing2
SunGear
SunShaft_calc
gearbox_calc
kSysGL3DView
LoadSpectrum
kSys3DView
Diagram
Z
Y
Information
General information about using KISSsys
Click the Administrator button to get Administrator or User rights
All KISSsys machine elements are displayed in gray
All KISSsoft calculation modules are displayed in blue.
Model Templates
Messages KISSsoft Information


KISSsoft - Student version - Cylindrical gear pair - Z32_to_Z2.Z12
File Project View Calculation Report Graphics Table Script Extras Help
Release 2024-SP3 KISSsoft
Projects
UCM25 GB
Z1_to_Z31
Z32_to_Z2
Basic data
Reference profile
Manufacturing
Tolerances
Strength
Factors
2D FEM tooth root stress
Geometry
Tooth geometry symmetric
Gear 1 Gear 2
Details
Normal module 1.0000 mm
Number of teeth 19 -82
Normal pressure angle 20.0000 °
Facewidth 31.0000 15.0000 mm
Gear 1 spur gear
Profile shift coefficient 0.0000 0.0000
Helix angle at reference circle 0.0000 °
Quality (ISO 1328:2013) 6 6
Center distance 31.5000 mm
Material and lubrication
Gear 1 Case hardening steel 18CrNiMo7-6, case-hardened, ISO 6336-5 Figure 9/10 (MQ), Core hardness >=25HRC Jominy J=12mm<HRC28
Gear 2 Case hardening steel 18CrNiMo7-6, case-hardened, ISO 6336-5 Figure 9/10 (MQ), Core hardness >=25HRC Jominy J=12mm<HRC28
Lubrication Oil bath lubrication ISO-VG 220
Geometry 3D
Gear geometry | System
Geometry 2D
Meshing
Modules Projects
Contents
General
KISSsoft System Module
Gears
Shafts and Bearings
Connections
Springs
Belts and chain drives
Automotive
Various
KISSsys
Bibliography and Variable Directory
Results (basic calculation)
Contact ratios 1.636 / 0.000 / 1.636
Gear 1 Gear 2
Actual tip circle (mm) 21.000 80.000
Root safety 0.440 0.550
Flank safety 0.632 0.749
Safety against scuffing (integral temperature) 3.344
Safety against scuffing (flash temperature) 3.100
Results (basic calculation) Results (special calculation) Messages Information
CONSISTENT

Contents

Photos showing the KISSsys model of the entire stepped-planet compound planetary gearbox used for analysis of the gears and geartrain (top), a photo showing the ring to small planet gear pair model setup in KISSsoft – this is used for configuring the many gear parameters and running analyses (middle), a photo of the table of contents from the 64 page report that is generated for each gear pair using KISSsoft and details everything there is to know about them.



Some screenshots showing the models at various steps within nTop (the software I have been using for topology optimization and static FEA). Top left is the initial input model, top right is the topology optimization result, bottom left is the result once some initial clean-up has been applied, bottom right is a rough topology optimization on the front upright I ran as I have been assisting with their design.

Some screenshots showing the topology optimised part once it has been exported from nTop as a .step file, imported to our CAD software (Onshape) and further refined.