Large scale 3d printing with CDPR

Cadence Boronkay

Research, planning, scope

Design goals + constraints

- Scalable
- Modular for in situ applications
- Precision satisfactory for 3d printing
- Visually/aesthetically appealing

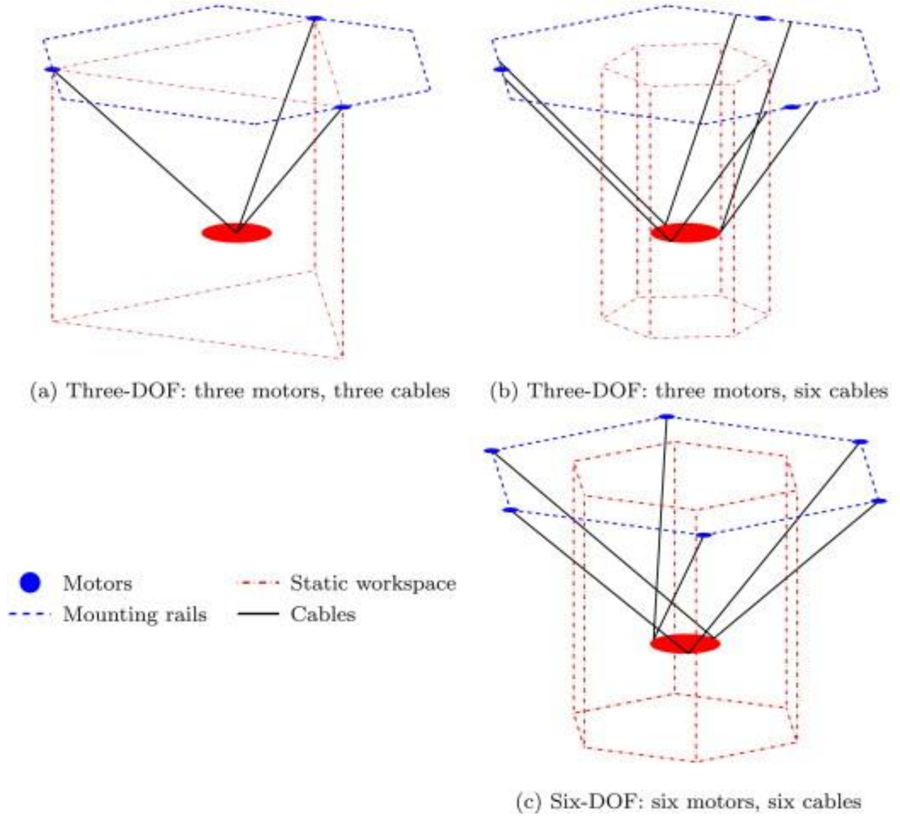
Previous work

- CDPR robots have been used at large scales (eg. construction), but typically for pick and place purposes, rather than additive manufacturing.
- Some 3d CDPR printers have been built at smaller scales, and with less modularity (fully constrained, many cables, fixed winch locations).
- There is a literature available on fully constrained CDPR printers but much less on underactuated (cable suspended) CDPRs.

CDPR constraining

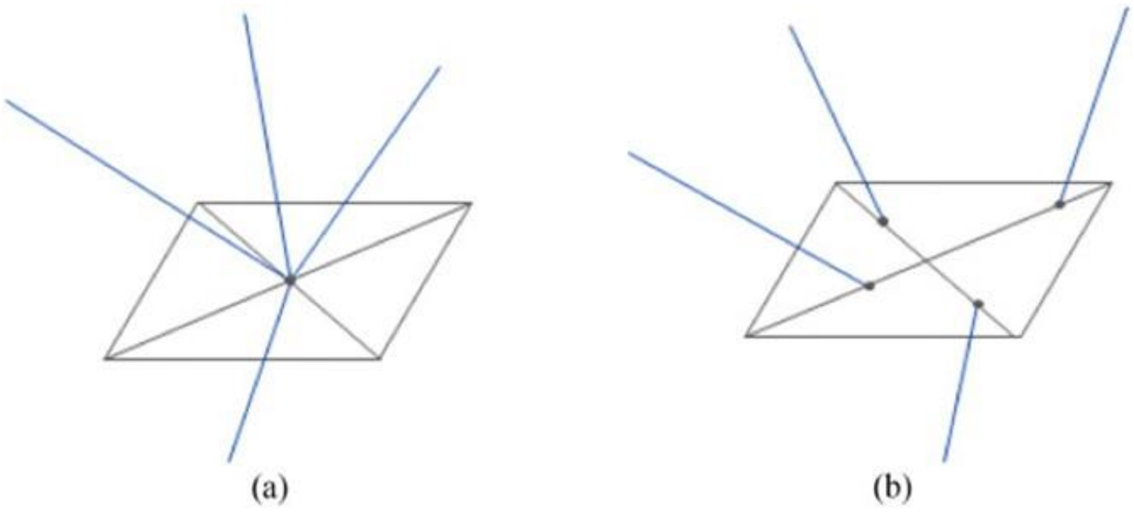
- m = number of cables, n = degrees of freedom
  - $m < n + 1$: under-constrained/underactuated (cable suspended)
  - $m = n + 1$: fully constrained
  - $m > n + 1$: redundantly constrained
- For 6 DOFs (full control of position and orientation), 8 cables are typically used.
  - This is more complex, as one must avoid cable-cable and cable-end effector interference.

CDPR constraining (cable arrangements)



Barnett, Eric, and Clément Gosselin. "Large-Scale 3D Printing with a Cable-Suspended Robot." *Additive Manufacturing*, vol. 7, 2015, pp. 27–44, https://doi.org/10.1016/j.addma.2015.05.001.

Choice of # cables will effect workspace size.



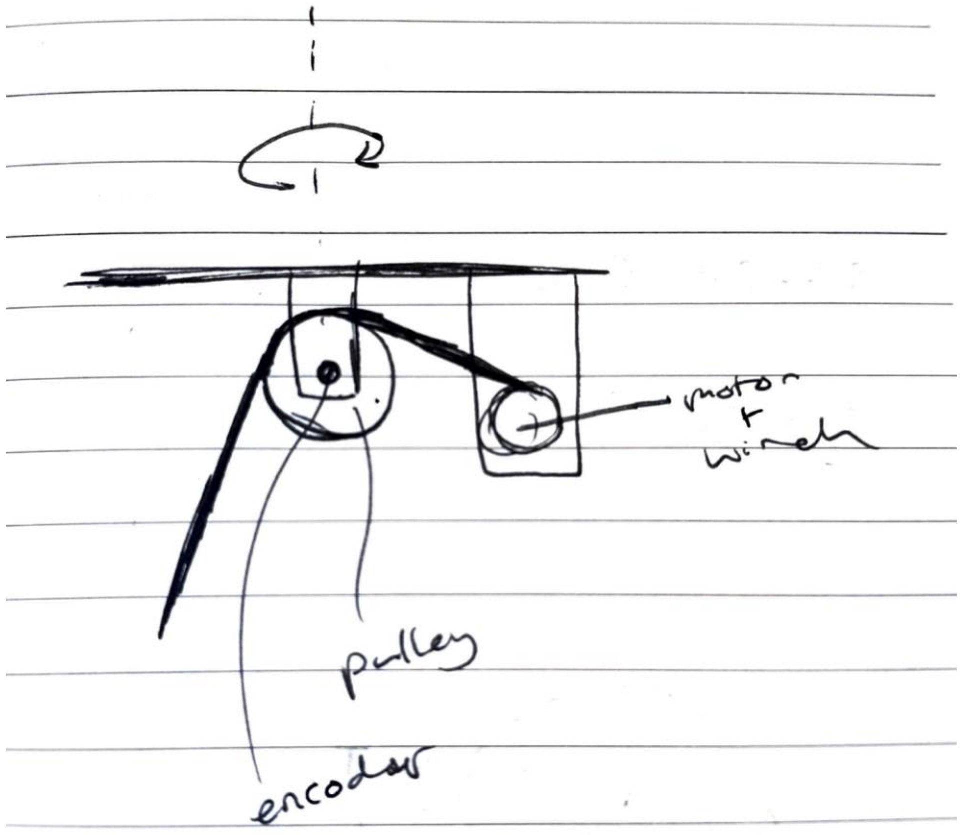
Ben Hamida, Ines, et al. "Multi-Objective Optimal Design of a Cable Driven Parallel Robot for Rehabilitation Tasks." *Mechanism and Machine Theory*, vol. 156, 2021, p. 104141, https://doi.org/10.1016/j.mechmachtheory.2020.104141.

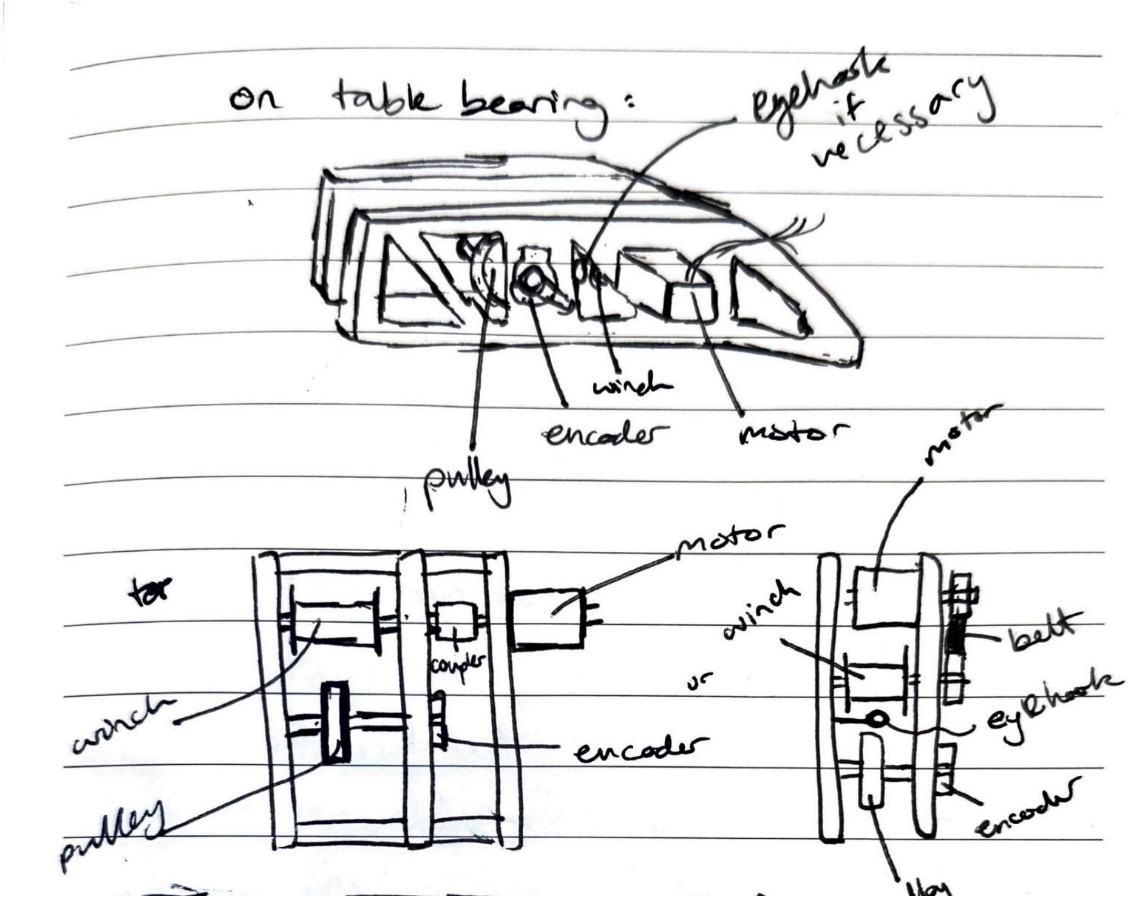
4 cables with unaligned attachment points allow for 3 DOF plus additional rotational mobility.

Controls

- Encoder options
  - Encoder on motor/winch (will have error due to irregular winch diameter from cable spooling)
  - *Rotary encoder on pulley (requires no cable slip on pulley, nonaxial forces due to the pulley on the encoder shaft could be problematic)
  - Draw wire/cable pull encoder (limits scale-up, maximum draw length of encoder determines possible cable travel).
- Still researching software options for trajectory control/path generation from G-code.
  - WireX, Hangprinter…

Initial design sketching





Design Considerations

- Pulley
  - U-groove
  - Eyehook to guide cable (prevent derailing)
  - For encoder, cable must not slip on pulley (pulley needs to rotate with cable the entire time)
    - Rubber coating? High traction material?
  - If slippage is unavoidable, encoder could be moved to separate measuring wheel. (still need to somehow prevent slippage -> maybe dual-side wheels?)
  - Check mechanical + dynamic calculations: need weight of end-effector, pulley material for friction coefficient, motor torque, etc.

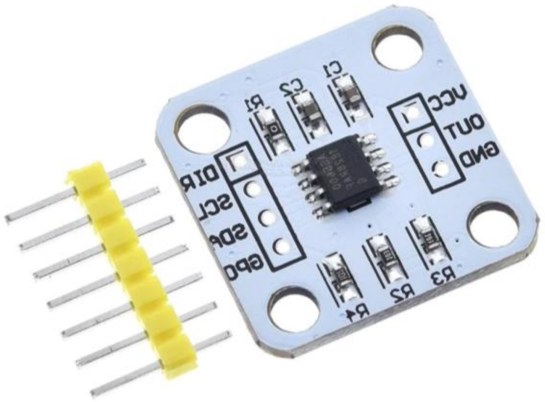
Design Considerations

- Turntable
  - Probably best to put pulley on a table bearing - more accurate cable length measurements and less non-axial loads on cable and pulley. (Expected to be more significant on large-scale).
  - Put entire module on turntable or just pulley?
    - I think more entire is more modular and elegant, but possibly unnecessary).
  - Table bearing axis of rotation needs to be centered above last point of contact (pulley) for controls. (Parametric measurements done from here too).

Planning

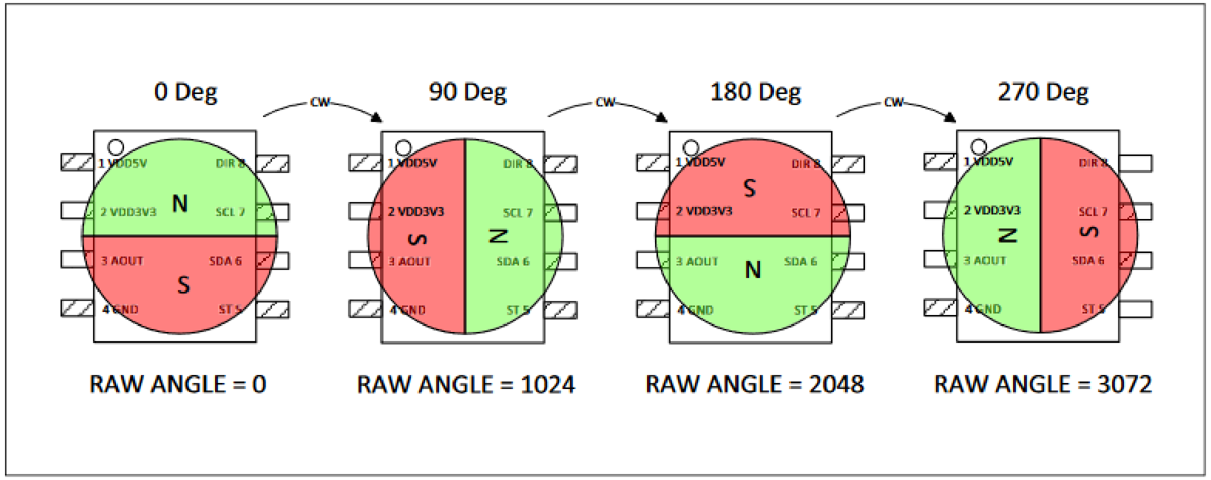
- Small-scale prototype design, continue controls research
- Fabricate prototype, assembly 1 module (module = 1 set of motor/winch/pulley)
- Test module
- Revision
- Fabricate + assemble *m* modules
- Assemble *m* cable system
- Test system (small scale CDPR)
- Revise
- Work on scale-up of module

Magnetic position encoder



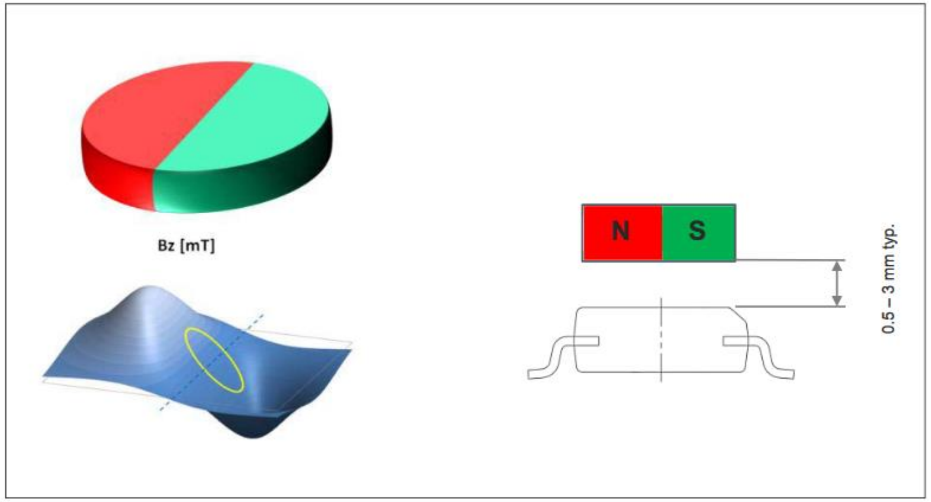
- Hall sensor AS5600 magnetic position encoder
  - Non-contact, high precision (4096 positions = 0.087 degrees res)

Figure 35:
Raw Angle in Clockwise Direction



ams Datasheet

Figure 40:
Magnetic Field Bz and Typical Airgap



Max allowable displacement of rotational axis is 0.25 mm

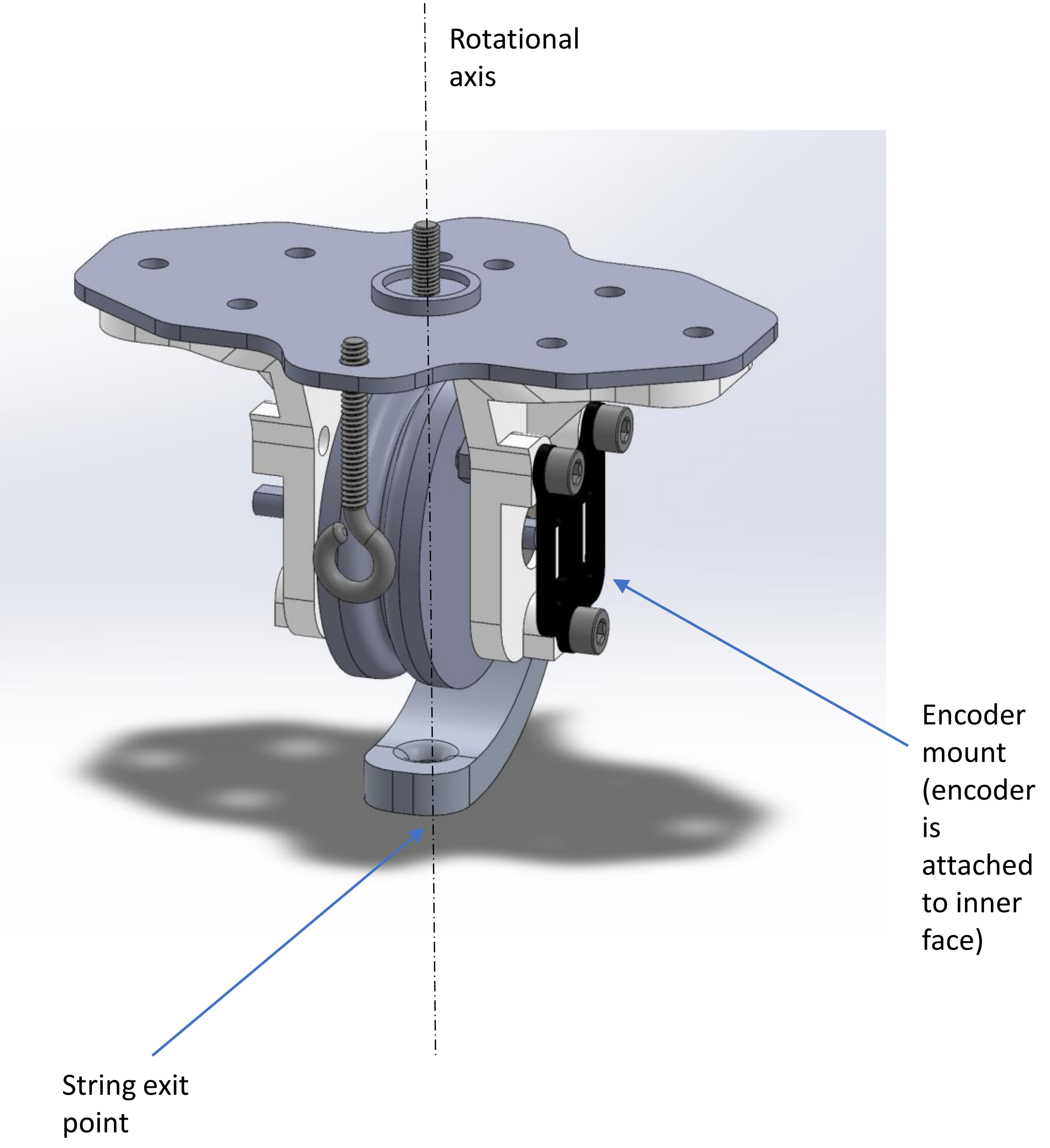
Useful: https://curiousscientist.tech/blog/as5600-magnetic-position-encoder

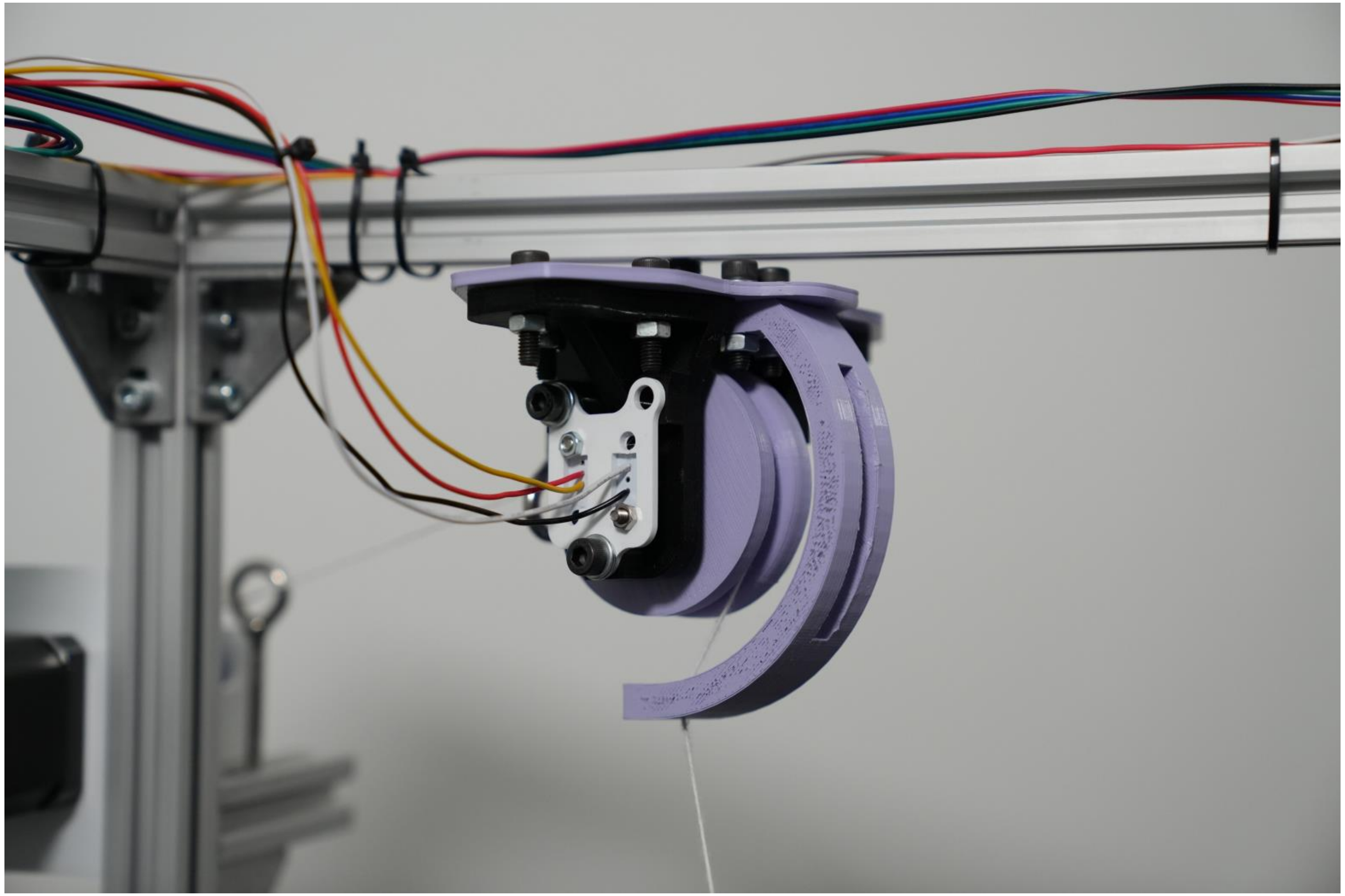
Mechanical Design

Small-scale version of CDPR mechanism

Pulley module

- Pulley for rope (currently thin string) needs to have enough friction to prevent slippage between string and pulley, as slippage would cause tracking error with the encoder.
- Need string exit point to be fixed. Since these pulleys rotate, the exit point is directly below the rotational axis such that rotation does not effect where the string ends contact with the module. (However, with both the rope guide and the eyebolt, it seems unnecessary to have the ability to rotate).
- Magnet for encoder is glued to the end of the shaft. Must be concentric and within 1-3mm of the encoder itself.



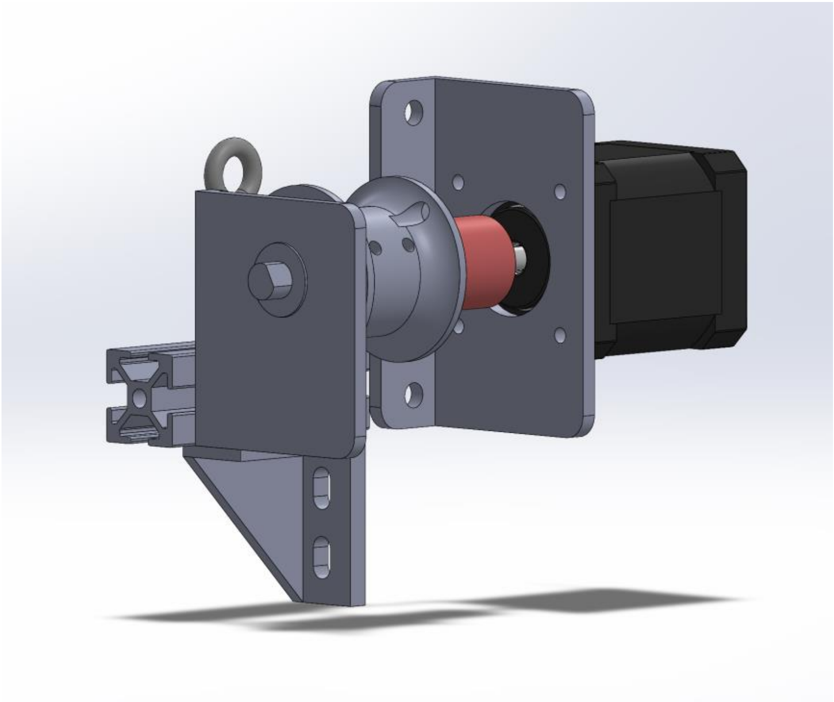


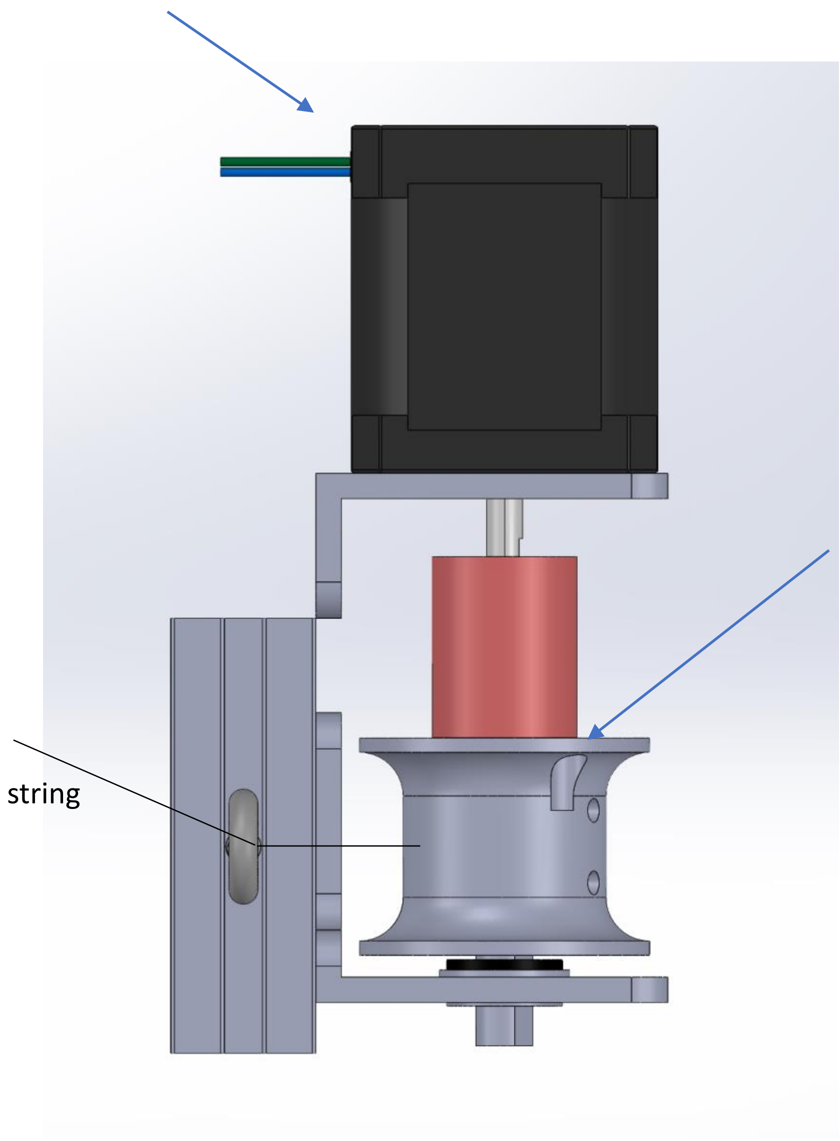
Pulley module BOM

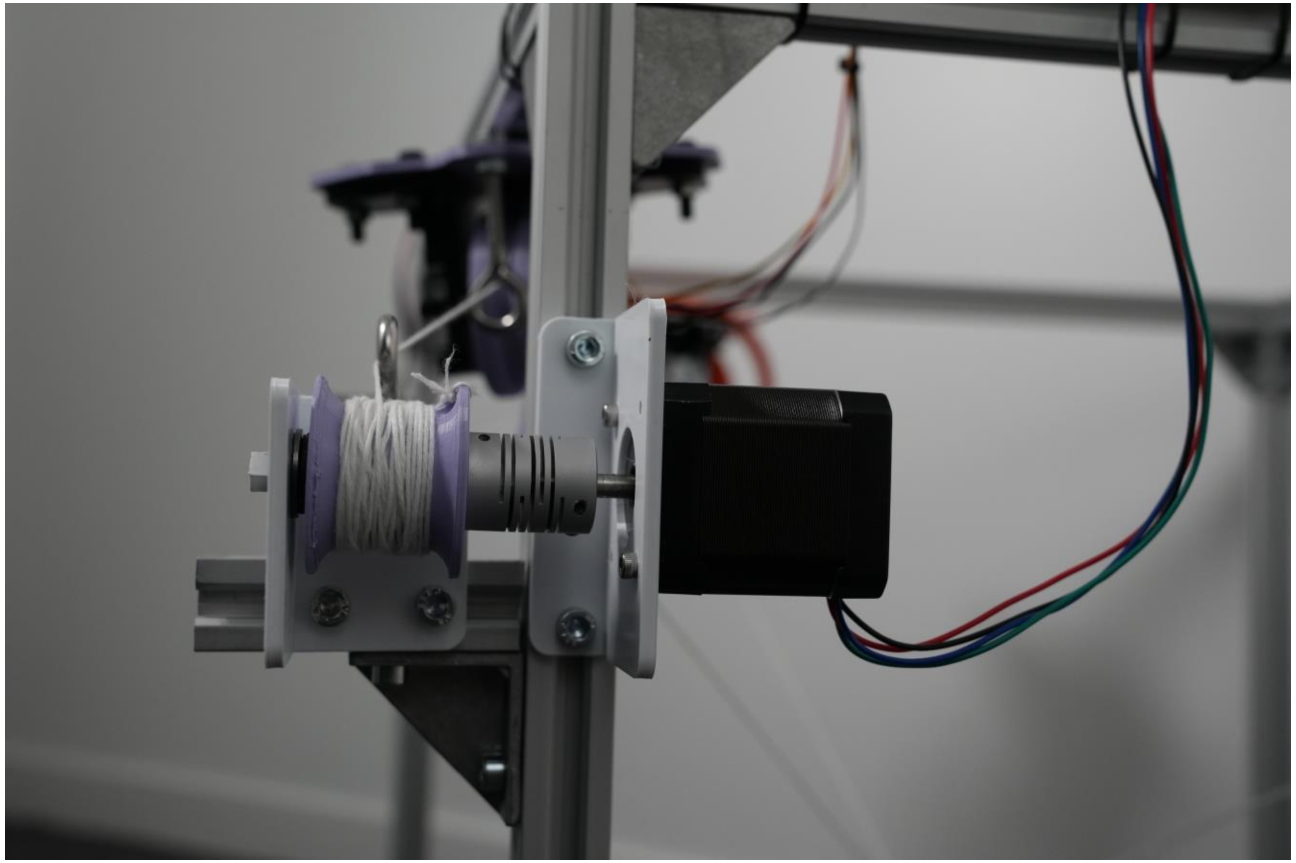
Single Pulley Module			
Part Name	**Qty**	**Process/Material**	**Remark**
Table base	1	3D printed (PLA)	
5mm Flanged Bearing (16mm OD)	1	-	
Small scale pulley	1	3D printed (PLA)	
Pulley mount	2	3D printed (PLA)	Designed for PLA although mine were printed in PETG
6mm flanged bearing (13mm OD)	2	-	
Encoder mount	1	3D printed (PLA)	
6mm d-shaft (L 55mm)	1	3D printed (PLA)	Steel shaft can be purchased but often run oversized for fitting in bearing
Rope guide	1	3D printed (PLA)	
AS5600 magnetic encoder	1	-	and accompanying magnet (will be fixed to end of shaft)
M5 eyebolt (L 3.5 cm)	1	-	
6mm washers (1.2mm thickness)	6	Nylon	
M5 socket head bolt (L 16mm)	9	-	
M5 socket head bolt (L 18mm)	3	-	
M3 screw (L 5mm)	2	-	
M5 washers (1mm thickness)	8	Nylon	Or can use equivalent 8mm spacer
M5 washers	7	Steel	
M5 hex nuts	13	-	
M3 hex nuts	2	-	
M5 extrusion insert nut	1	-	If mounting module to aluminum extrusion

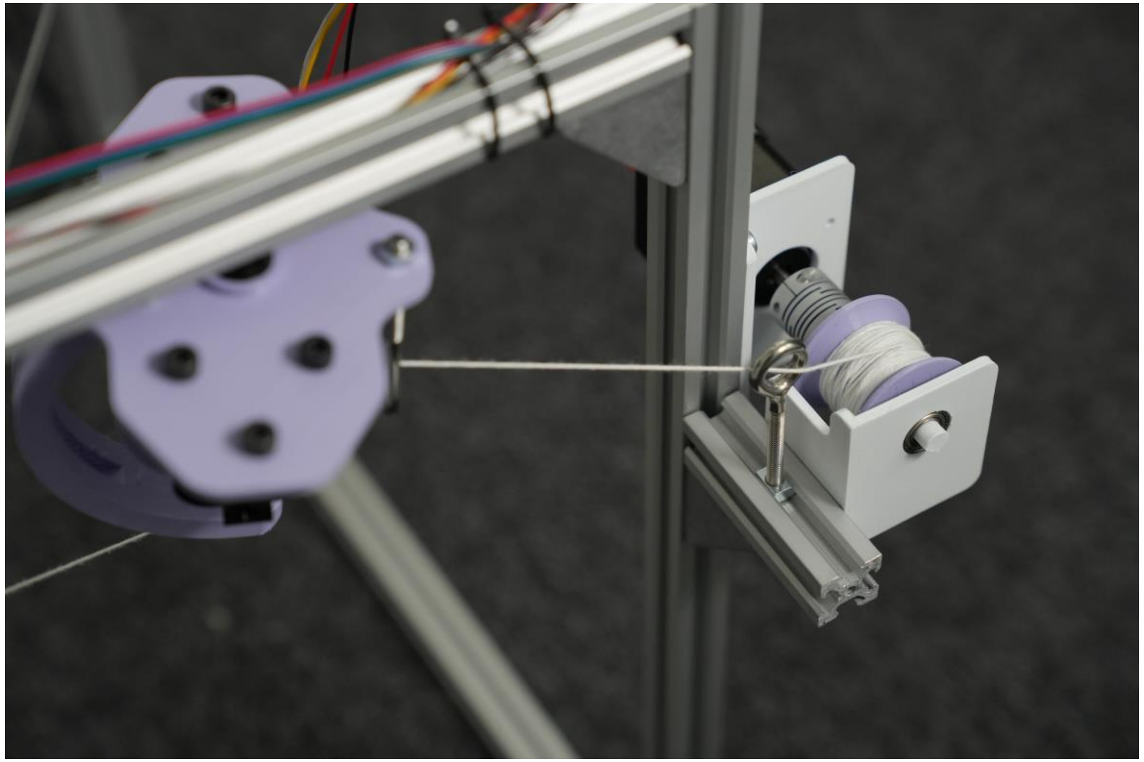
Winch module

- Compact, such that it can be placed anywhere
- Eyebolt to prevent string from derailing off of winch spool
- Most parts 3D printed





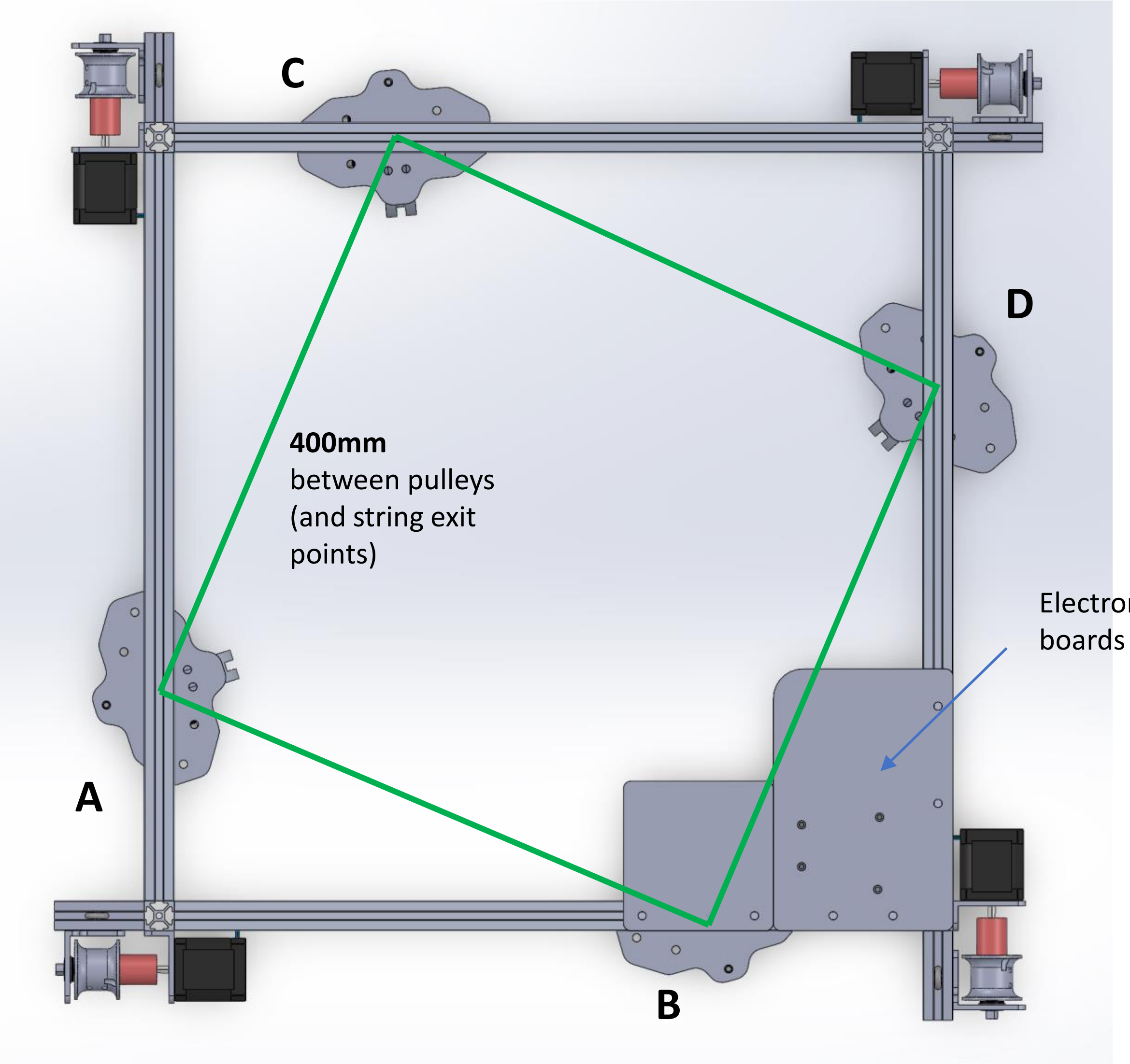


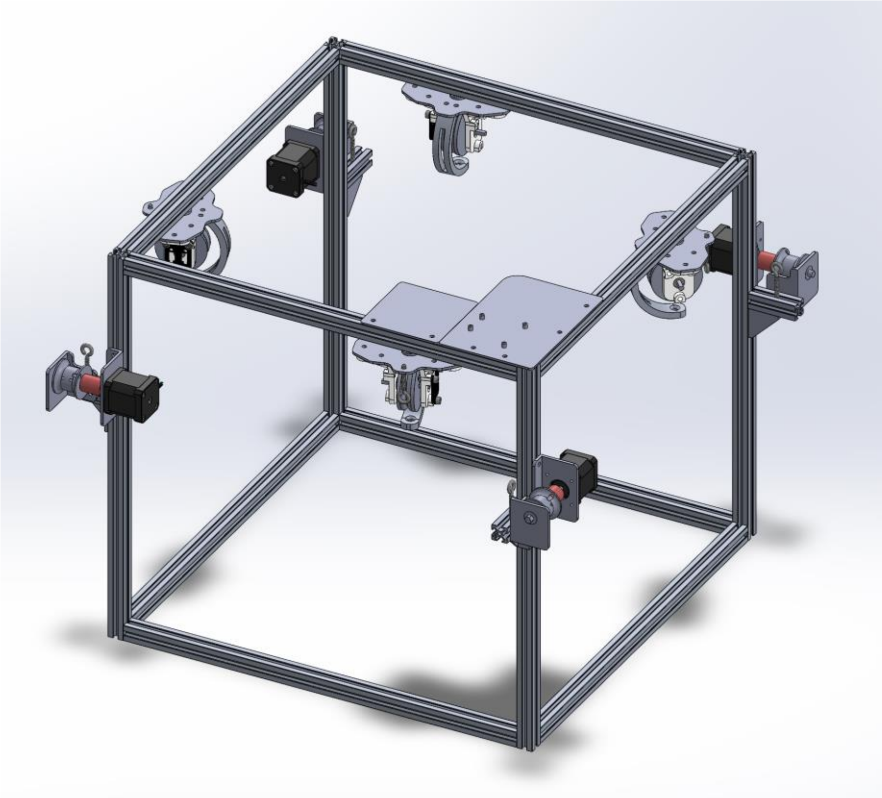


Winch module BOM

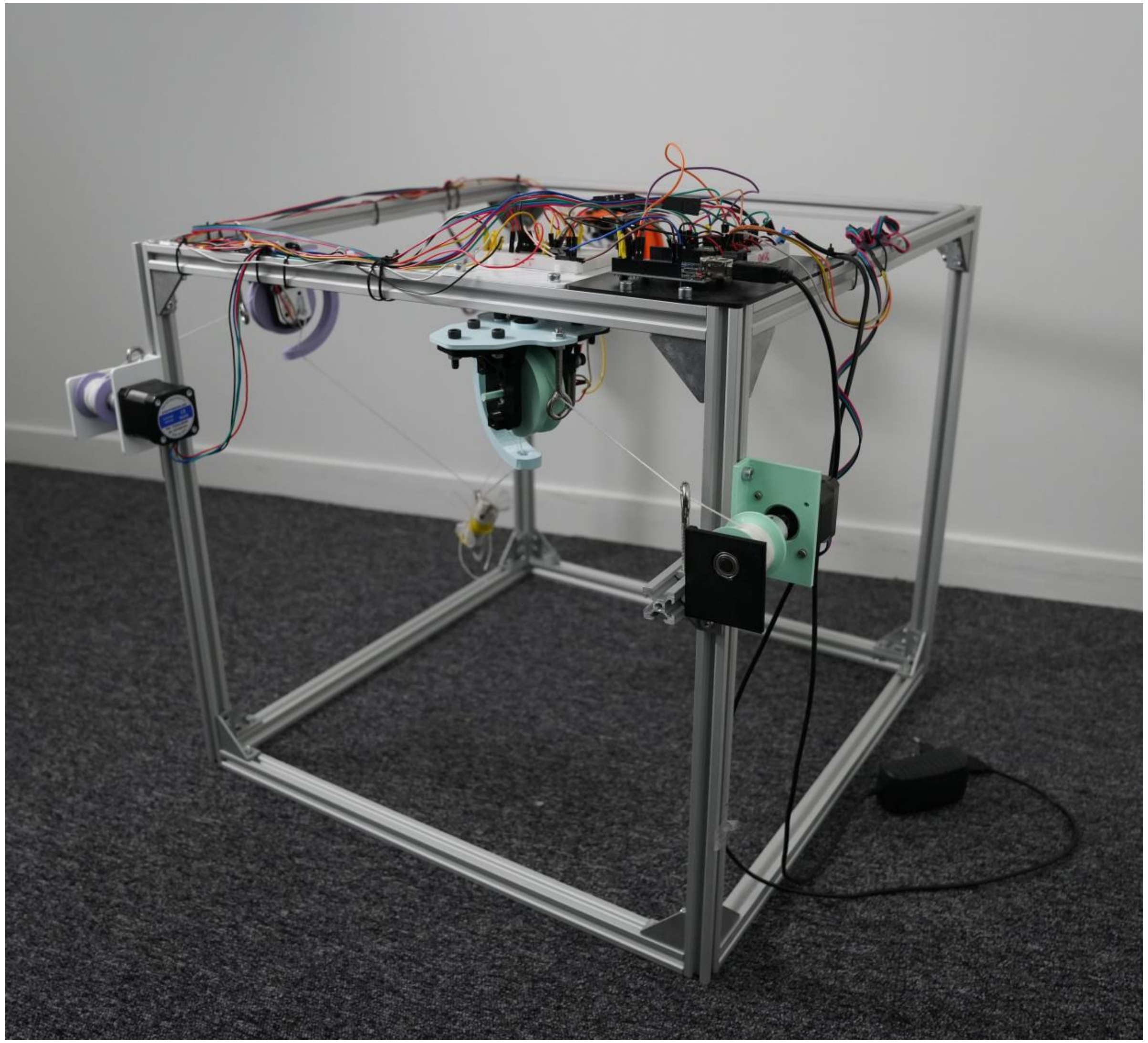
Single Winch Module			
Part Name	**Qty**	**Process/Material**	**Remark**
NEMA 17 stepper motor	1	-	
Motor mount	1	3D printed (PLA)	
Winch bearing mount	1	3D printed (PLA)	
2020 T-slot profile bar (L 60mm)	1	Aluminum	
5mm to 8mm shaft coupler	1	-	
8mm flanged bearing (16mm OD)	1	-	
Winch spool	1	3D printed (PLA)	
8mm d-shaft (L 55mm)	1	3D printed (PLA)	
M5 eyebolt (L 3.5 cm)	1	-	
Extrusion corner bracket	1	-	
M8 washer (1.4mm thickness)	1	Nylon	
M5 bolt (L 8mm)	8	-	
M5 extrusion insert nut	9	-	If mounting module to aluminum extrusion
M3 screw (L 10mm)	2	-	

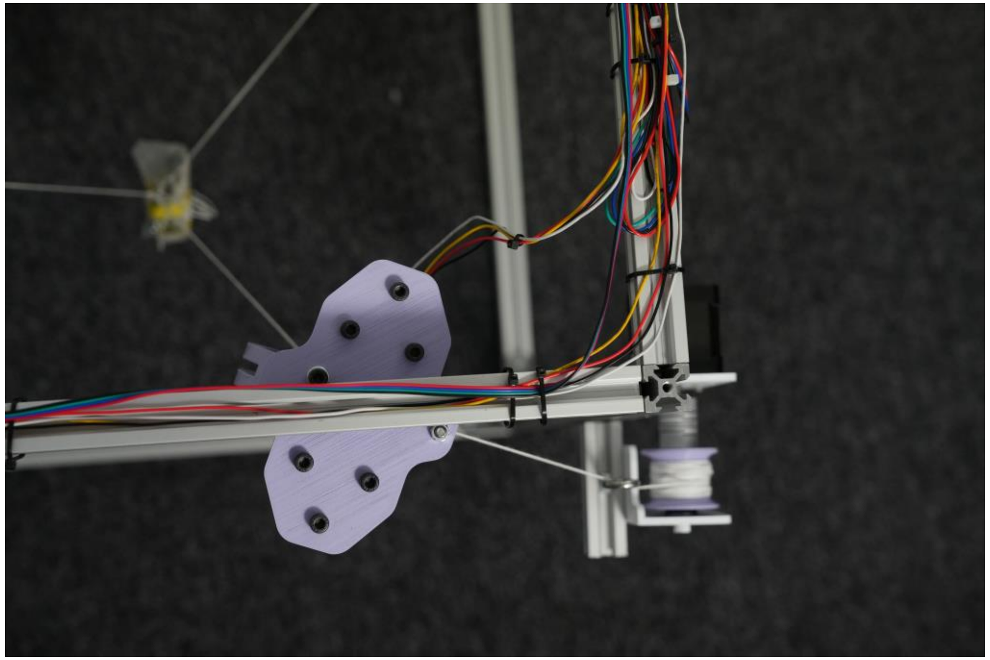
Structural schematic

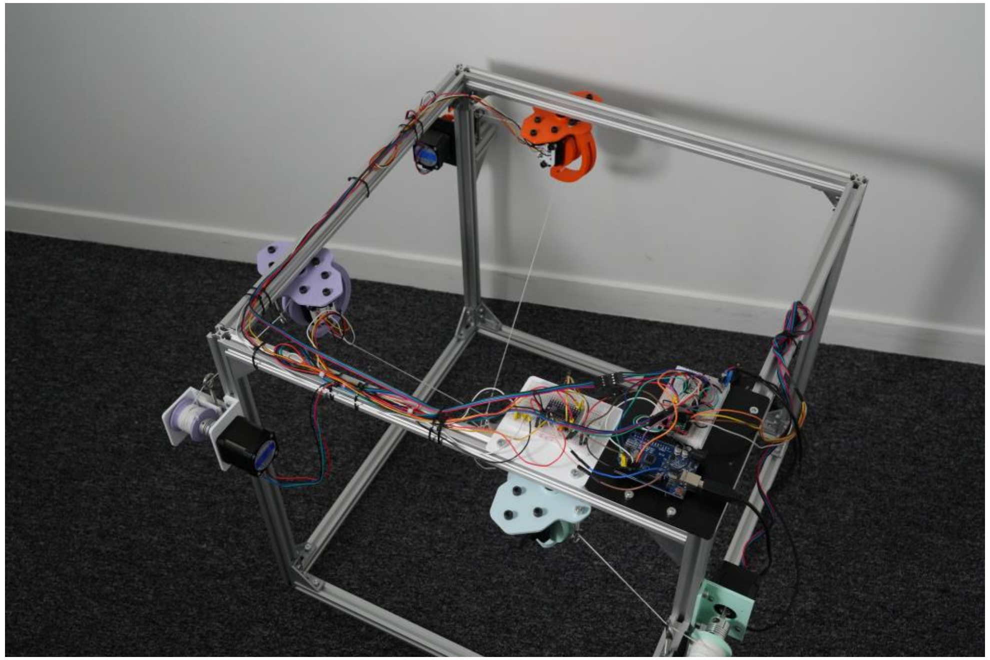




- Uses 2020 aluminum extrusion for easy assembly (corner brackets for frame connections not shown).
- ~70 cm x 70 cm footprint
- Distance between pulleys can change but calculations for cable lengths in Python program will need to be adjusted
- Pulleys are labeled for reference in code





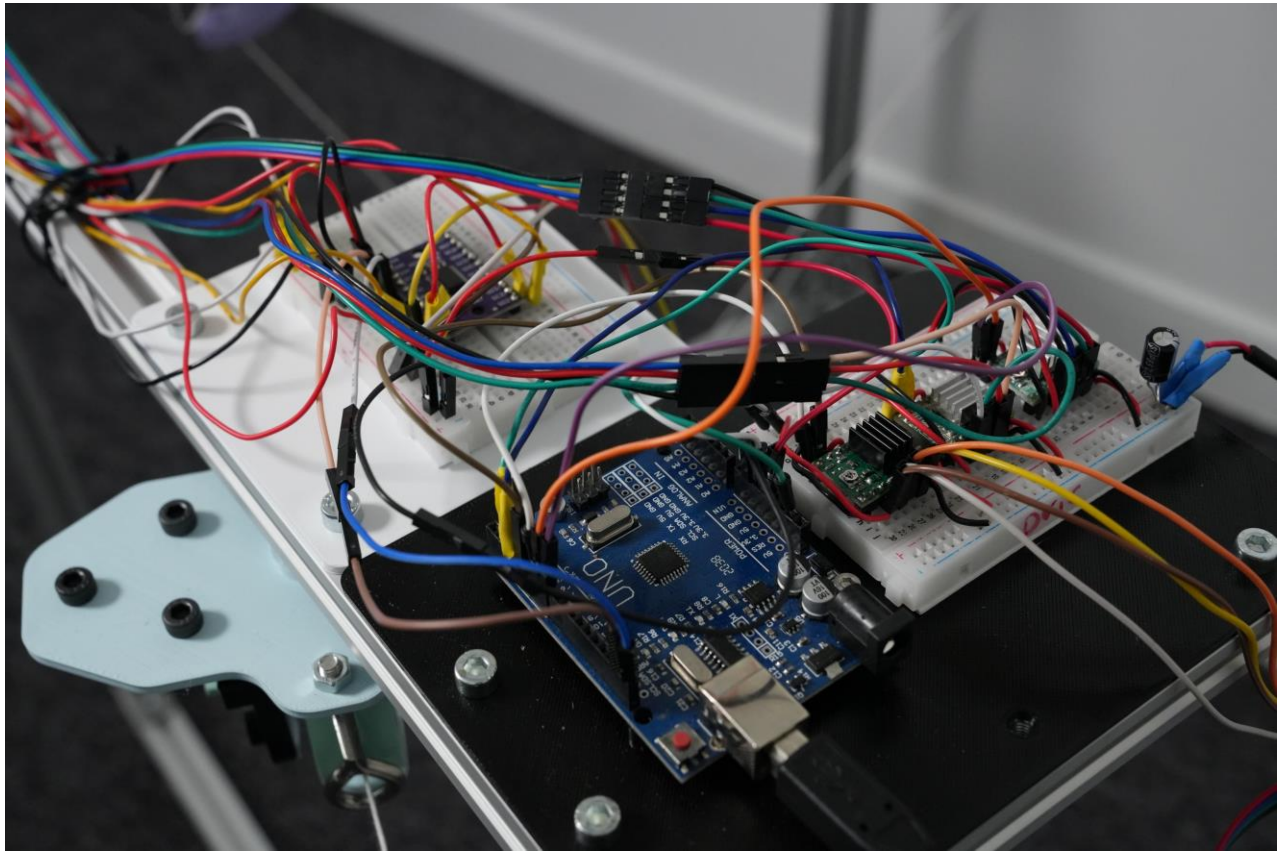


*4th module not yet added

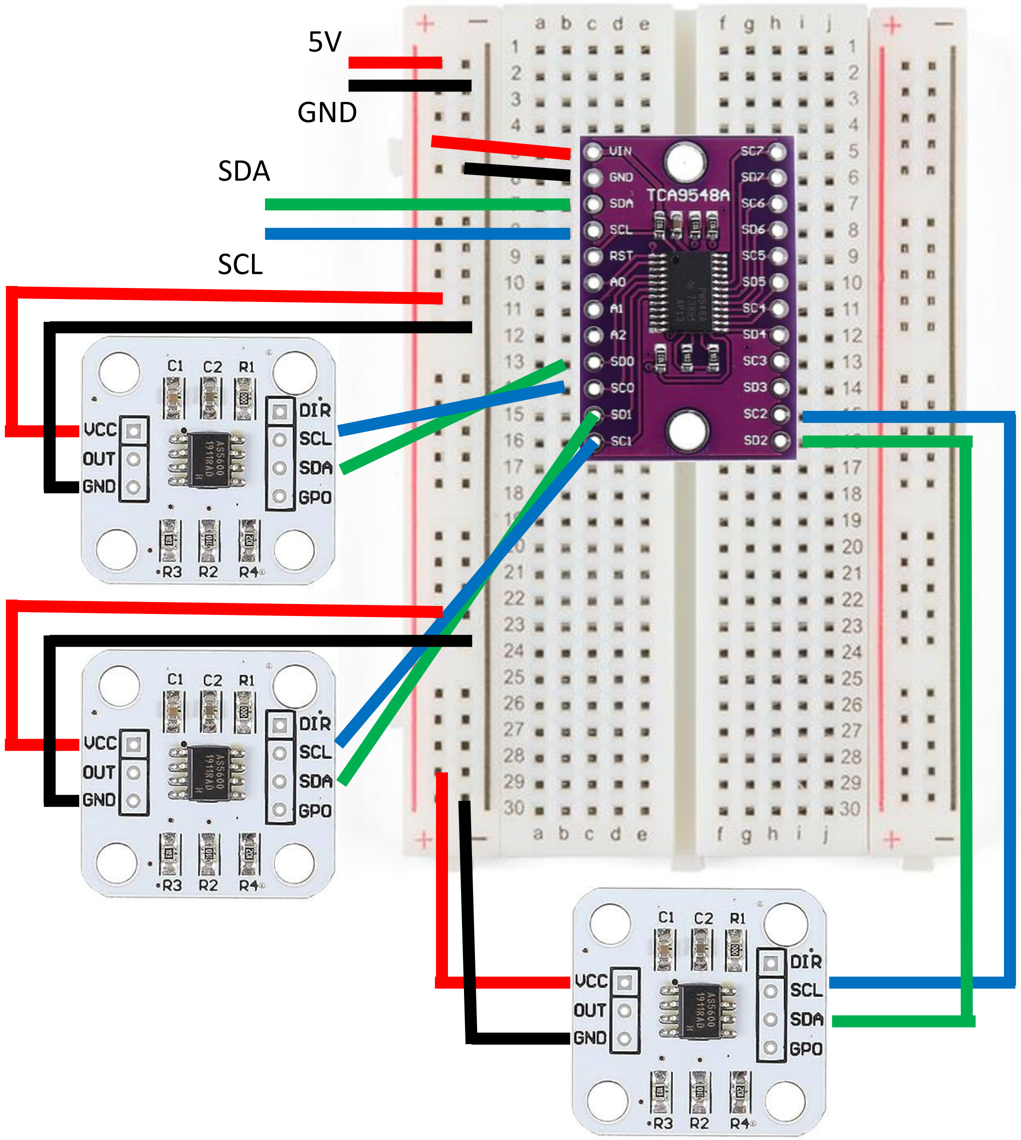
Electronics/hardware

Components

- Arduino Uno
- A4988 stepper motor drivers (1 per motor)
- AS5600 magnetic encoder (1 per pulley)
- I2C multiplexer
- 100 µF capacitor
- 12V power supply



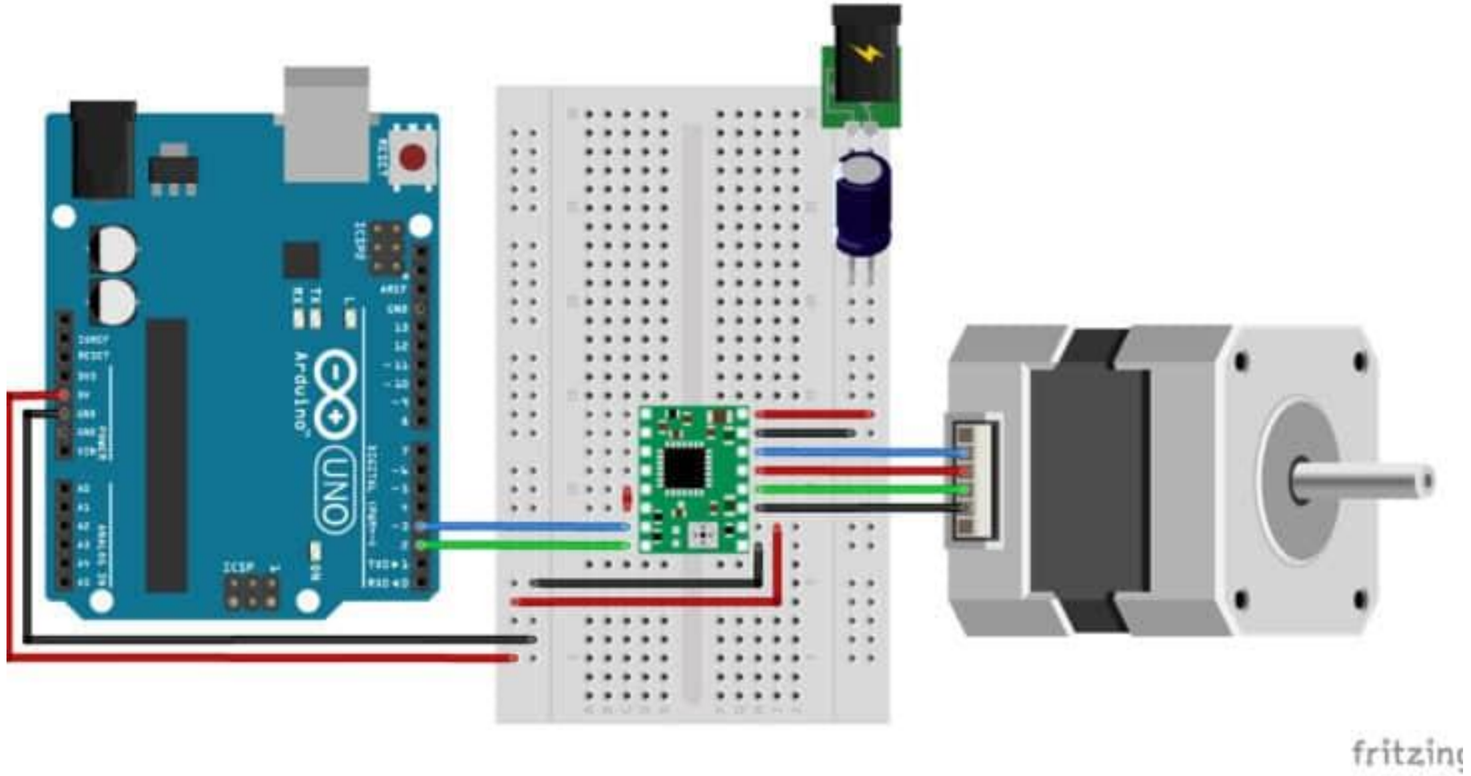
I2C and encoders schematic



Can add up to 8 encoders easily with 1 I2C multiplexer in this method

Motor driver schematic

- Add motor drivers and motors with same connections
- Do not plug/unplug motors when 12V power supply is connected and on!



https://www.makerguides.com/a4988-stepper-motor-driver-arduino-tutorial/

Software

Converting G-code to cable length coordinates

- Python program: gcode_to_ourcoords.py
  reads a G-code txt file and outputs a string of coordinates that can be copied into the Arduino sketch to control the pulley system.
- It also outputs the number of coordinates, which will need to be set in the Arduino sketch.
- Currently works best with not too long, cleaned up G-code (removing non-coordinate G-code lines/commands)
- Calculation section will need to be adjusted for different pulley configurations (eg. different distance between pulleys)
- Python program and example G-code can be found in GitHub repository: https://github.com/cxboronkay/Cable-Driven-System-for-3D-Printing

```
PS C:\Users\Cady\Documents\MIT\MISTI\DVIC>  c:; cd 'c:\Users\Cady\Documents\MIT\MISTI\DVIC'; & 'C:\Users\Cady\AppData\Local\Programs\Python\Python311\python.exe' 'c:
\Users\Cady\.vscode\extensions\ms-python.python-2023.12.0\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '49844' '--' 'C:\Users\Cady\Documents\MIT\MI
STI\DVIC\gcode_to_ourcoords.py'
G-code Coordinates
[X(60.0) Y(60.0) Z(40.0), X(80.0) Y(60.0) Z(40.0), X(80.0) Y(80.0) Z(40.0), X(40.0) Y(80.0) Z(40.0), X(40.0) Y(40.0) Z(40.0), X(80.0) Y(40.0) Z(40.0), X(80.0) Y(60.0
) Z(40.0)]
----------------------------------------------------------------------
CDPR Coordinates
[(248, 339, 339), (274, 305, 359), (299, 327, 327), (256, 393, 289), (204, 361, 361), (256, 289, 393), (274, 305, 359)]
----------------------------------------------------------------------
Number of coordinates:
7
Copy this to Arduino sketch:
{
{248,339,339},
{274,305,359},
{299,327,327},
{256,393,289},
{204,361,361},
{256,289,393},
{274,305,359},
}
```

Controlling the machine

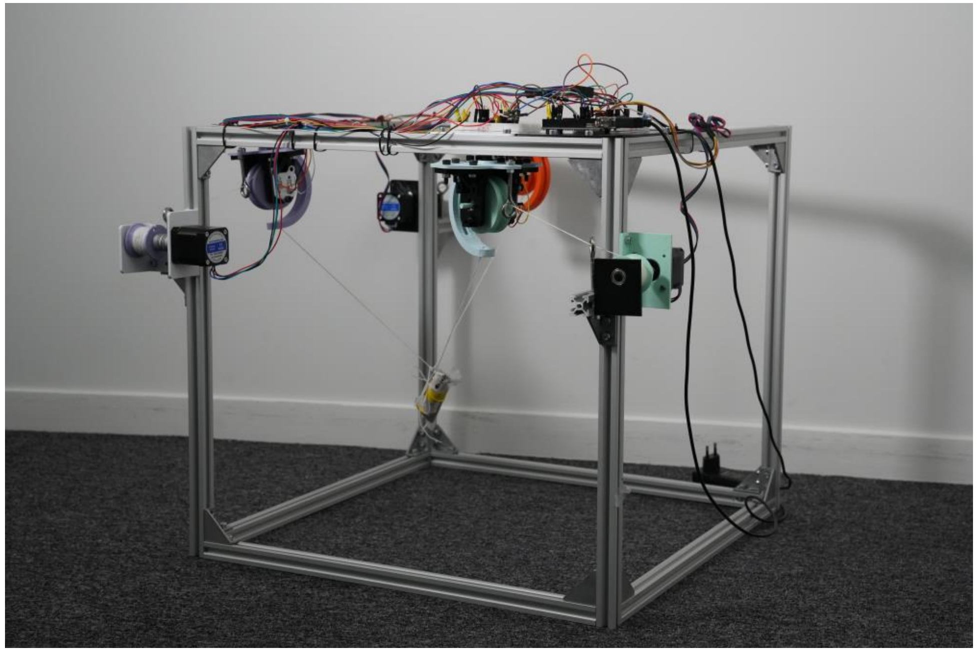
- Arduino sketch for performing trajectory described by translated G-code coordinates: run_gcode_3pulley.ino
- Copy coordinates from python program where indicated. Also adjust numcoords value and the initiation length for the coordinates list.
- Manually rotate winches to position end effector at the starting position noted in the sketch (originally 200, 350, 350 mm).
- Once uploaded to Arduino, begin Serial and turn on 12V power. There is an intentional 10 second delay before it will begin to move (can be shortened or removed if desired). It will then move to each coordinate in the list before stopping.
- Next location, current coordinates, and speeds will all appear in the Serial print out.
- Pulleys A, B, and C refer to the same pulleys noted in the structural schematic shown in the Mechanical Design section.
- https://github.com/cxboronkay/Cable-Driven-System-for-3D-Printing

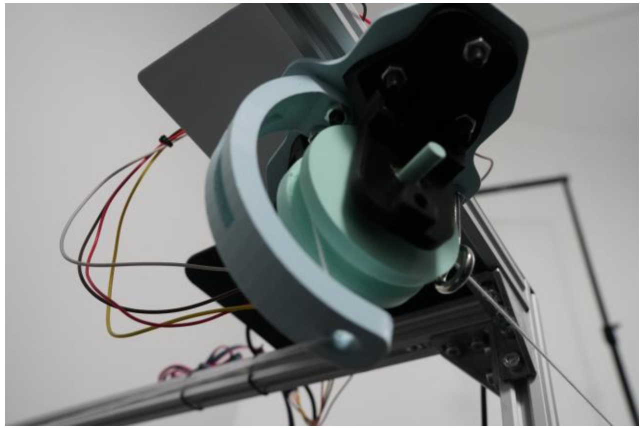
Future steps

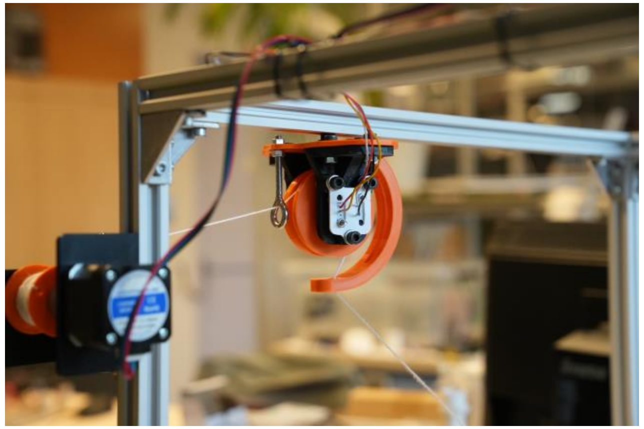
Next

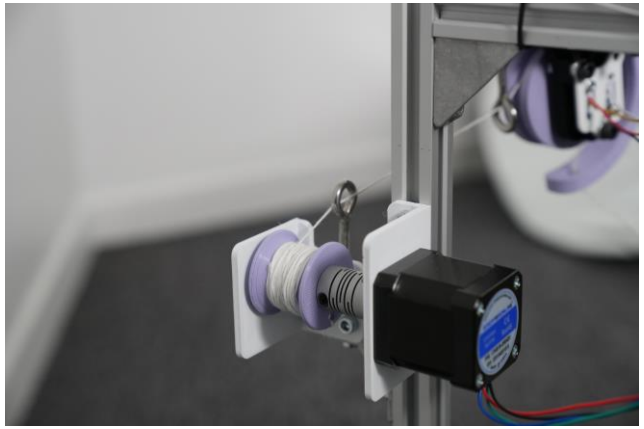
- To get closer to the end goal of a large-scale CDPR 3D printer, next steps include:
  - Adding 4^{th} pulley to gain another degree of freedom (be able to partially control angle of end effector).
  - Improve software and control over trajectories.
  - Scale up to be able to support the 3D printing nozzle.
  - Make design more modular (mountable to anything, sturdier, etc.)

Thank you!!









Contact:

Cadence Boronkay

boronkay@mit.edu

cwxboronkay@gmail.com