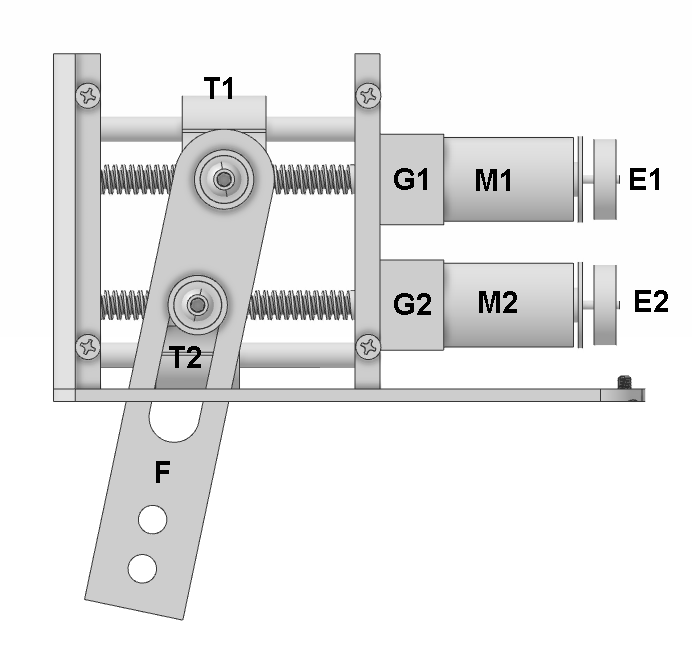
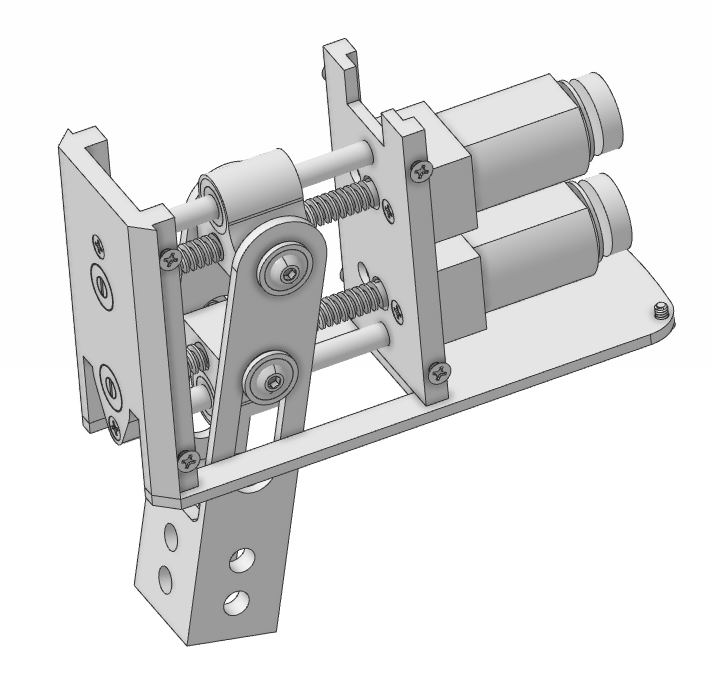
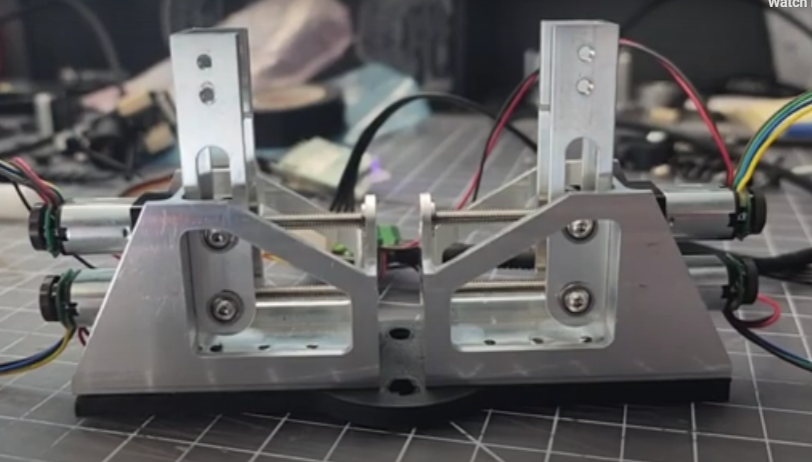
Modular Variable Gripper Brainstorming

2024-02-06



*Frame removed to view internal components*



<https://youtu.be/o8JT-9tQeA8>

Side view of latest proof of concept prototype using 50:1 geared DC motor

# CONCEPT

The design has been in the works for a long time, and is finally to the point where the electronics and firmware can be brainstormed. The objective is to design and release a gripper with the following key features:

* Works with the Lynxmotion PRO arms without issue
  + Mechanically easy to integrate
  + Same input (electrical and communication connections) as PRO
  + Similar protocol to PRO Servos
  + Integrated into PRO arm software
* Acceptable overall specifications
  + Good opening for <1Kg objects and general applications / experimentation
  + Acceptable max force
  + Not as fast as a dedicated single motor parallel gripper (high speed application)
  + Lightweight
* Novel features
  + Two motors per finger allows parallel and angular motion
  + Configured as two or three fingers
  + Three mounting positions for larger opening
  + Very low MSRP compared to many grippers on the market

# RESPONSIBILITIES

Mechanics

* Coleman (lead)
* Eric N.

Electronics

* Huzaifa (lead)
* Hirokazu (verification)

Firmware

* Hirokazu
* Yves

Testing & Integration

* Yves (PRO Arm UI)
* Geraldine (ROS 2)
* Eduardo

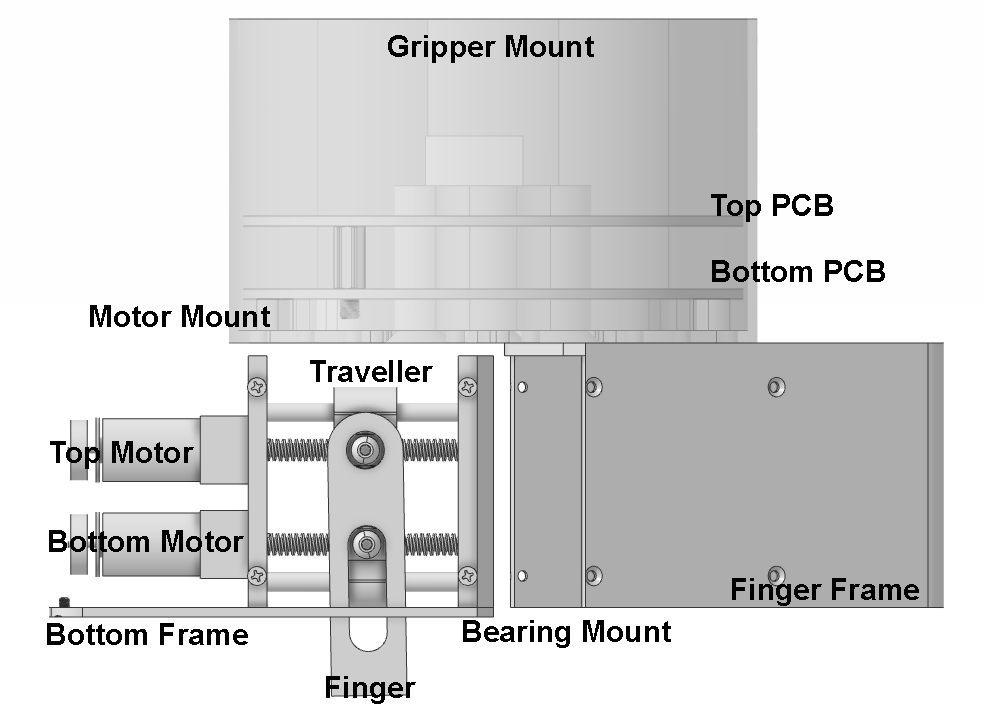
# MECHANICS

2x Identical DC Gear Motors (M1, M2) [Datasheet](https://drive.google.com/file/d/172SzxEvToy6MRe1LB5E0_673zfDenXMR/view?usp=drive_link)

* 12V nominal
* 5V encoder on motor’s rear shaft, 5 CPR (E1, E2)
* 50:1 Gear down (G1, G2)
* Tr3.5 Threaded shaft, ~34mm long

2x Identical Travelers (T1, T2)

* Tr3.5 threaded portion for motor’s shaft
* 3mm Linear bearing or bushing for stability
* Ball bearing for rotation for T1
* Ball bearing or Delrin for linear and rotational motion for T2

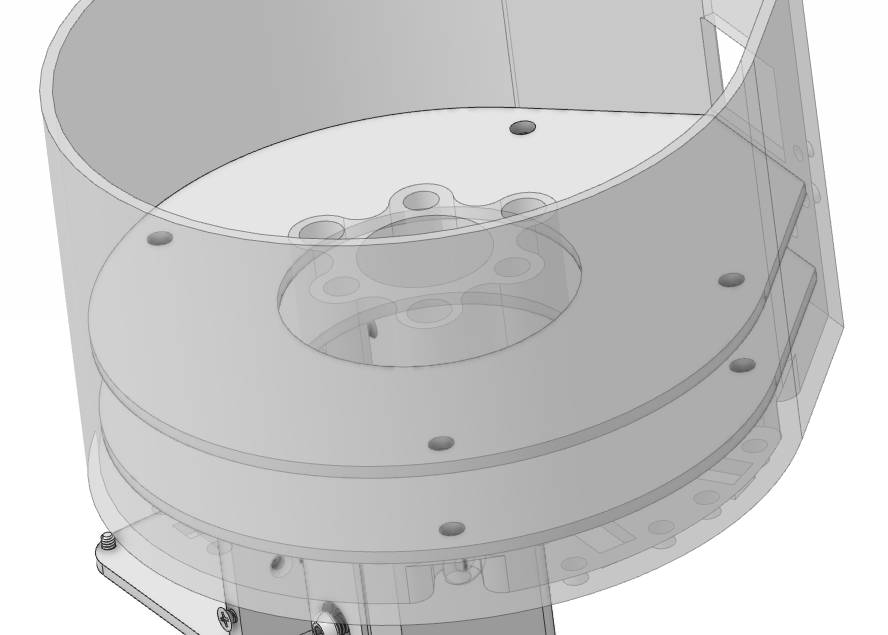


Nomenclature

# ELECTRONICS

The electronics can be mounted at the end of the last joint, between the end of the gearbox and inside the gripper mount so the fingers are not extended far beyond the last joint.

* Microcontroller: STM32F407 (same as LYN PRO Servos)
* CAN chip (same as LYN PRO servos)
* Voltage regulation (ex. 36 -> 12V for motors)
* DC Motor Controllers (Appendix)
  + Current detection (ideally stall detection and within motor controller but accessible to microcontroller)
* XT60 Connector (~30 to 36V input, perhaps a bit more/less for max)
* RJ45 Connector (same pinout as PRO servos)
* USB for reprogramming and firmware updates (perhaps hidden)
* RGB LED at 90 degrees (functionality TBD, much like the PRO servo)
* Button at 90 degrees (functionality TBD, much like the PRO servo)



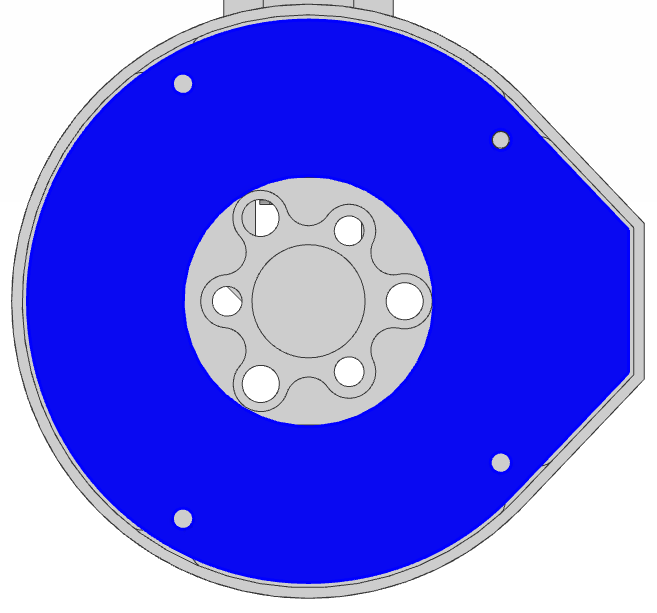
* Motor connections
  + Given there are five possible mounting points for a finger, with two motors to be connected via each opening, there can either be 10 total connectors or if the cable connecting the motor to the PCB is long enough , two openings can share the same two connectors, for 6 total.

PCB Stack

* Four M3 mounting holes
* 10mm (can be extended up to 12mm) standoffs separating PCBs
* 5mm distance between bottom PCB and
* We will provide a layout for each PCB denoting the max height for components on the top and bottom.
* Inexpensive male / female interconnect

# 

Center pattern using three screws for either LITE or STANDARD PRO servo.



**PCB shape**

* Inner radius: 35.5mm (min) to 36mm
* Outer radius: 38mm to 39.75mm (max)
* Protrusion for RJ45 & XT60: TBD

**FIRMWARE**

* Gripper can be assigned an ID, or each finger assigned an ID
* Most of the basic LSS PRO Communication protocol reused, but new commands needed for linear position and angle
* Zeroing each finger based on stall detection, each time upon startup
  + Have M1 move at slow speed until it hits the mount then
  + Have M2 move at slow speed until it hits the mount
* Parallel motion accomplished by reading the encoder counts and ensuring they are the same by adjusting speed using PID (same concept as 2WD accurately maintaining a straight line)
* Angular motion accomplished by reading the encoder counts

**SAFETY**

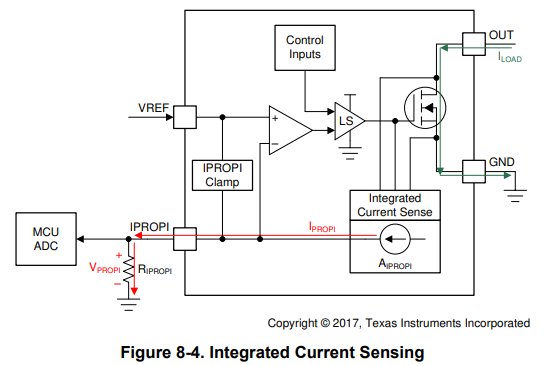
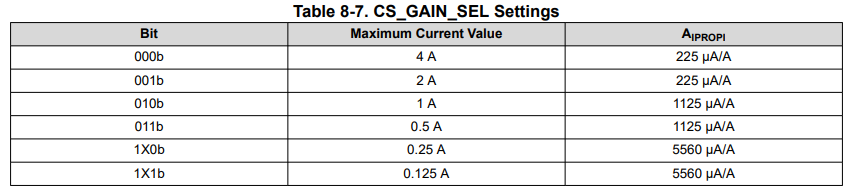
* Once each motor has been zeroed, the firmware would consider the total linear travel possible vs. total encoder counts to ensure each motor does not go beyond the travel in either direction
* The force is limited, so it’s doubtful that anyone will be harmed if the fingers close on a human finger or something else.

# UNCERTAINTIES

* (CB) Working on a way to get smooth and accurate linear and rotational motion for T2 rather than direct bearing to metal contact.
* (EN) 50:1 DC gear motor seems to provide significantly more torque and higher travel speed compared to 50:1 geared stepper motor. However, the stepper’s motion is based on steps and unless one is skipped (detected in the controller), the motion of each motor should be identical. Is DC the best option?
* (CB) Finding the best DC motor controller with all the features we want.
* (CB) The design as such does not conform to any industry standards; from the mounting (specific to the LITE and STD servos), to the electronics and communication and the fingers / mounting pattern.

# QUESTIONS

1. (EN) How accurately do you think you can make the linear position for each motor to ensure the finger remains at a fixed angle?
   1. (ED) This would depend on the number of pulses of the encoder or the number of poles of the motor if we use the sensorless approach, it would also depend on the backlash between the mechanical parts.
      1. (CB) We have received a few prototypes from this company and there’s play in the output shaft of the gear motor (the threaded shaft), and it can move into and be pulled away from the motor by perhaps 0.25mm.
   2. (ED) Since we are using a 3.5mm pitch threaded shaft, it should move 3.5mm for each revolution, so in the case of the encoder we have 5 pulses for each revolution and a gearbox of 50:1, so for the threaded shaft to make a turn it would take 50 motor revolutions equivalent to 250 encoder pulses, so we would have a resolution of 3.5mm/250 pulses which would be 0.014mm/pulse.
      1. (CB) Exactly.
   3. (ED) Using the sensorless approach, which measures the motor ripple, each pulse would correspond to the number of motor poles, in general there is one ripple for each pair of poles (so we need to know the number of poles of the motor). Both the DVR8214 and the DVR8234 have a digital output RC\_OUT with a stream of pulses corresponding to the current ripple of the motor, so as with the encoder, to control the position we can calculate the amount of pulses needed to reach the desired position.
      1. (CB) Having never used that type of technology, can it be trusted to not lose any “counts” I wonder, or is a rear shaft encoder the best option, or even a combination of both somehow?
2. (EN) How to ensure each finger is at the same linear position and angle if one motor encounters a force but not the others?
   1. (ED) To keep the same angle throughout the entire movement it is necessary that both motors move at the same speed, when one of the two detects an excess of current both motors of the same finger should be stopped. The force at T1 and T2 will be different in most cases so the driver must take this into account, the proposed drivers can control the speed with variable voltages and loads, however the DRV8214 is better than the DRV8234 because it is able to measure current with a higher resolution, the problem is that the output voltage is maximum 11V and the motors are 12V, so it would be best to use the DRV8234.
3. (CB) Given the current workload for the PRO (firmware and arm), how long would such firmware take to create?
4. (CB) Is there a way to use current detection either on the motor controller chip or separately to do force feedback?
   1. (GB) Yes, both options are available with the drivers linked on the appendix. Both drivers have a current sense output that would allow a microcontroller to detect motor stall or changes in load conditions.
   2. (GB) Both drivers you suggested have this feature, however the, DRV8214 has higher accuracy because it allows you to adjust the gain factor, for the DRV8234 the gain is set to 1500 uA/A.



# The resistor on the IPROPI pin can provide a voltage signal to the microcontroller ADC. The RIPROPI resistor can be sized based on the expected load current in the application so that the full range of the controller ADC is utilized

Both drivers support hardware stall detection by comparing the IPROPI pin voltage to the VREF pin voltage, so using the VREF pin, the driver can regulate the motor current during startup and highload events without interaction from a microcontroller.

# APPENDIX

DC Gear Motor Datasheet

<https://drive.google.com/file/d/172SzxEvToy6MRe1LB5E0_673zfDenXMR/view?usp=drive_link>

New TI Chip (available July 2024):

<https://www.ti.com/lit/ds/symlink/drv8214.pdf> (only up to 11V)

<https://www.ti.com/lit/ds/symlink/drv8234.pdf>

Sensorless position and speed control:

<https://www.ti.com/lit/ab/slvafr5/slvafr5.pdf>