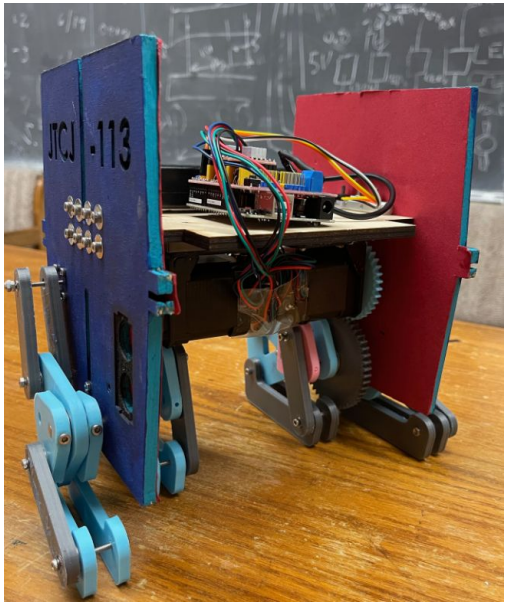
MAE 189 Capstone Design
Team 3: Gonk Walker

Team Members:
Tom Nguyen
Jonathan Chavez
Connor Linklater
Qiyuan Lu



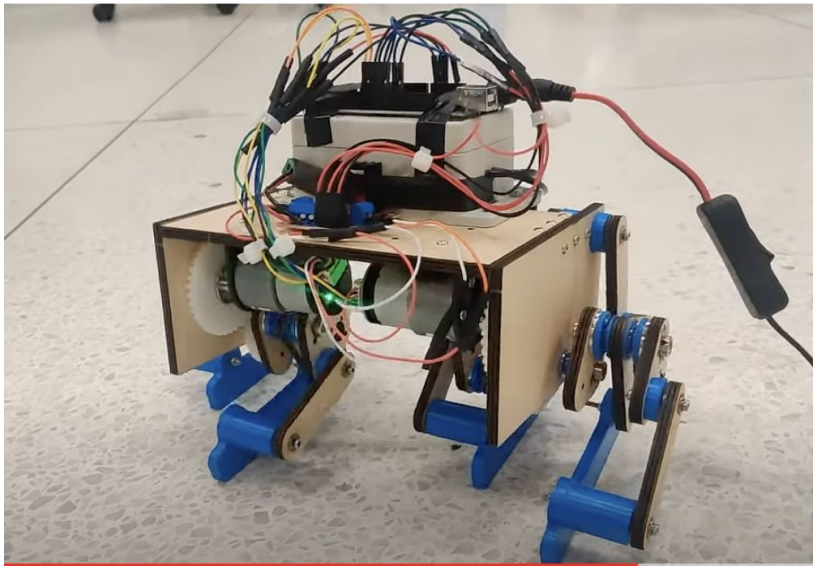
Overview and Requirements

UCI Samueli School of Engineering

- Problem Definition: A steerable mechanical walker consisting of one drive motor each side.

Design Attributes/Requirements/Objectives

- Speed > 1.5 fps
- Turning radius < 3 ft
- Weight < 7 lbs
- Remote control of speed, forward, reverse, left and right
- DIY manufacture
- Arduino Uno
- Tank Steering
- Minimal Vertical oscillation < 1”
- Option for autonomous navigation

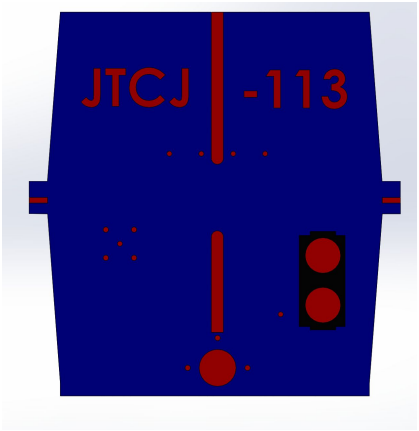


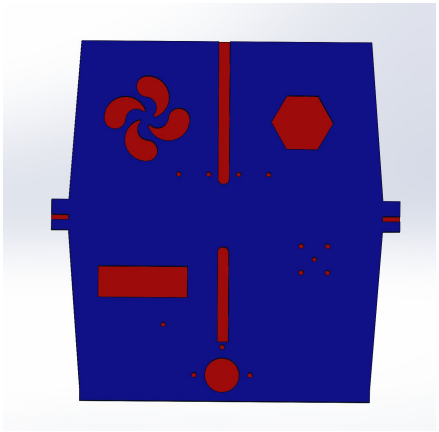
Reference design for a steerable walker developed by Chenhao of Sustech University.

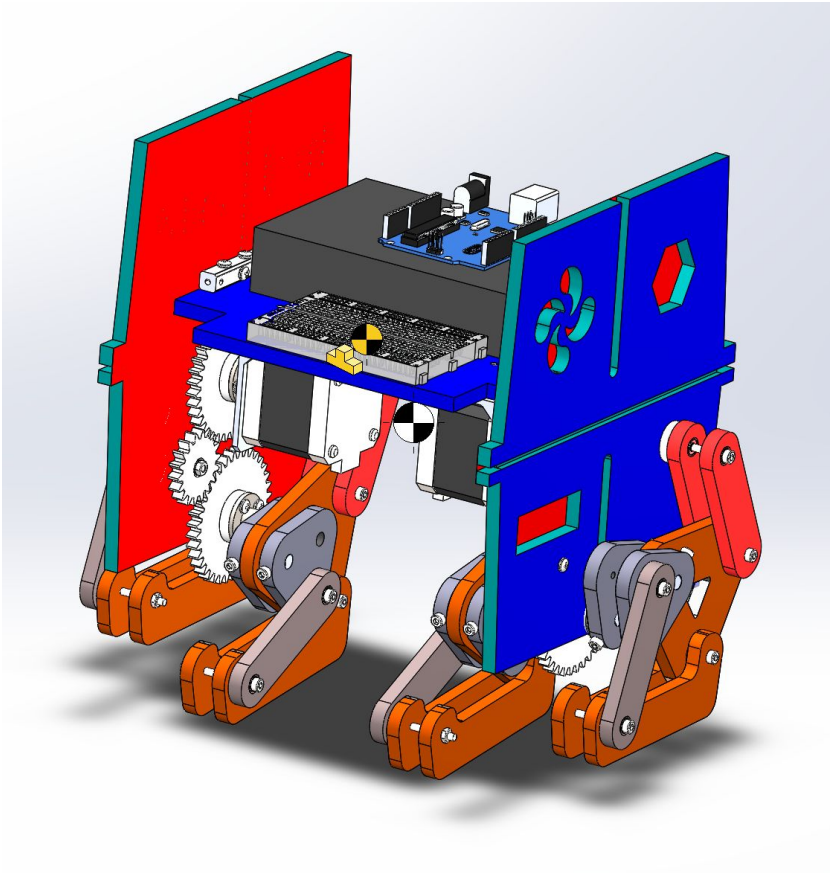
Tom Nguyen

Design Decision 1: Theme

UCI Samueli School of Engineering







Jonathan Chavez

Design Decision 2: Drive Motors

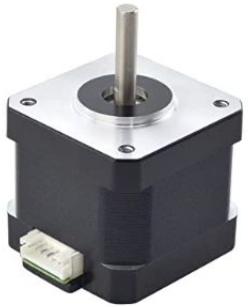
UCI Samueli School of Engineering

Gear Motor

- High Torque at low RPM
- Easy to control
- Able to control:
  - Speed
  - Direction

Stepper Motor

- High Torque at low RPM
- Easy to control
- Able to control:
  - Speed
  - Direction
  - Acceleration
  - Distance



[redacted]

[redacted]

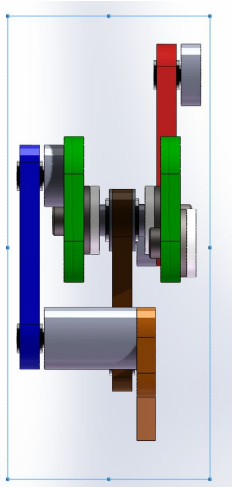
Jonathan Chavez

Design Decision 3: Leg Packaging

UCI Samueli School of Engineering

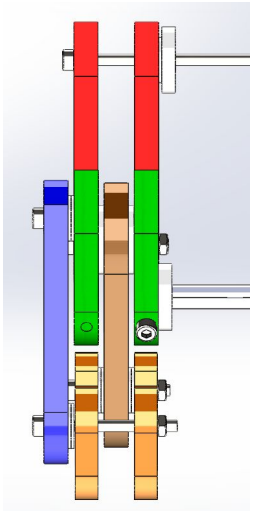
Old Leg design

- Layering of leg links with spacers required for clearance
- Layers and spaces extends the width of the leg
- Support forces create cantilever bending on chassis connection



New Leg design

- Double shear connection place loads at center of the links
- Reduces the number of spacers and the width of the leg
- Keeps support forces near chassis connection.



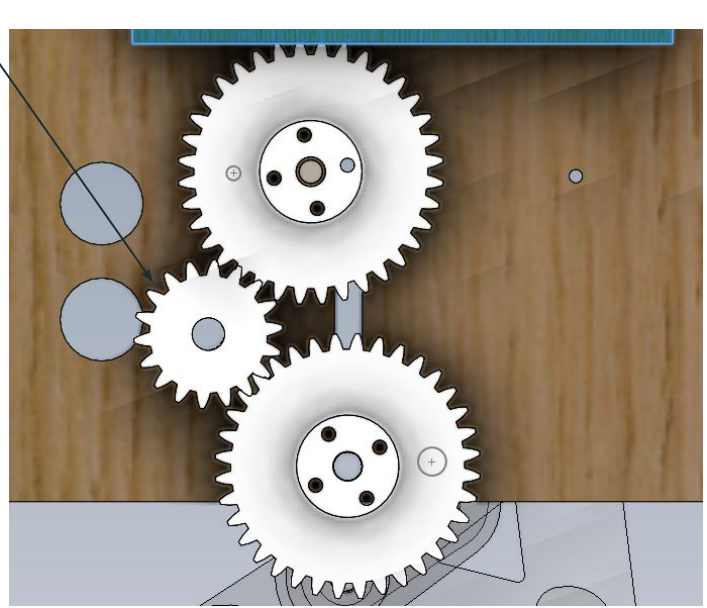
Qiyuan Lu

Design Decision 4: Driveline Packaging

UCI Samueli School of Engineering

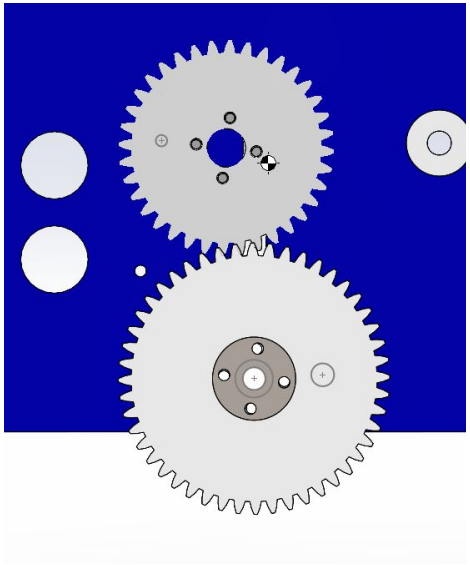
Stepper Motor Driveline Prototype A

14 Tooth Gear: Added to create space for stepper Motors which are larger than the gear motors



Stepper Motor Driveline Prototype B

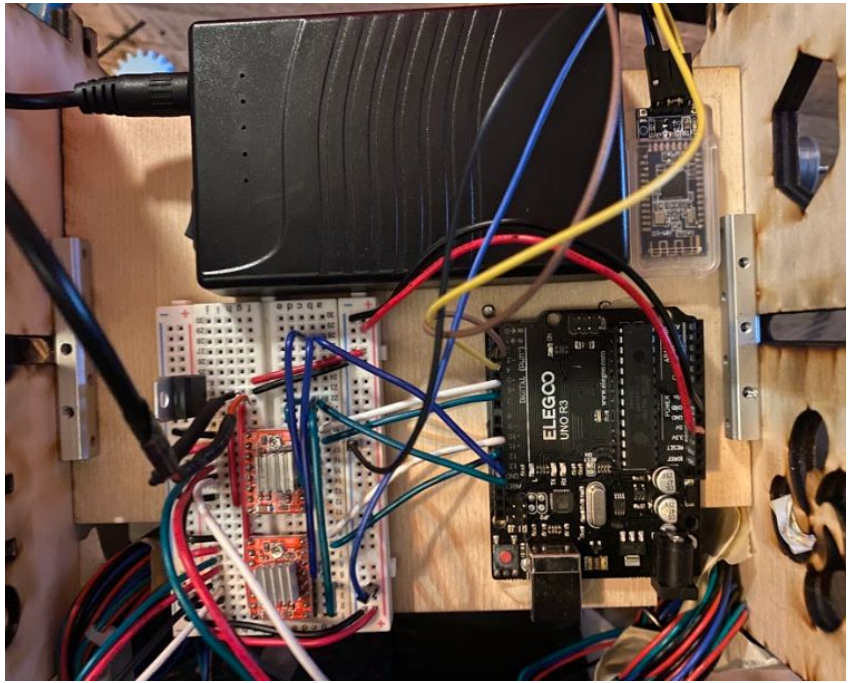
48 Tooth Gear connected to the rotating leg mechanism creating a 1:1.33 gear ratio



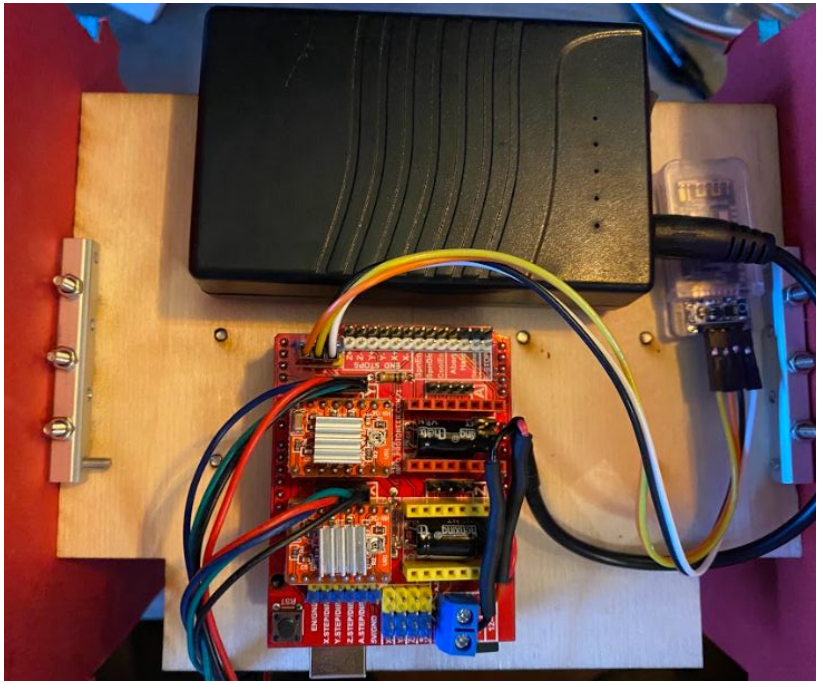
Connor Linklater

Design Decision 5: Wiring

UCI Samueli School of Engineering



Prototype A



Prototype B

Connor Linklater

Arduino Code

UCI Samueli School of Engineering

Setup Code

```
#include <math.h>

#define PUL_PIN 2   // Pulse Pin (sometimes called a step pin)
#define DIR_PIN 5   // Direction Pin
#define PUL_PIN2 3  // Pulse Pin (sometimes called a step pin)
#define DIR_PIN2 6  // Direction
#define enable 8    //enable pin

// Steps per revolution of my motor/driver combo
//SoftwareSerial BTserial(0,1);
int Motor1Interval;  //Defined step interval for each stepper motor
int Motor2Interval;
int FullRightRot;  // Rotation count rounded down by defining it as an integer
int FullLeftRot;
int k = 1;
int s = 2000;  //Time interval between steps in microseconds

float LeftStepCount = 0;  //Step count variable keeping track of how many steps have been taken for each motor
float RightStepCount = 0;
float leftdist;  // Amount of distance traveled by each motor calculated
float rightdist;

unsigned long previousMotor1Time = micros();  //Time variable of when the motor stepped last
unsigned long previousMotor2Time = micros();


char BTvar = '0';  // character variable

bool B_executed = false;
bool X_executed = false;



//SETUP

void setup() {
  //Start serial to read both leftdist and right dist, and for the HM-10 BT module
  pinMode(PUL_PIN, OUTPUT);  //LeftMotor Output pins
  pinMode(DIR_PIN, OUTPUT);
  pinMode(PUL_PIN2, OUTPUT);  //Right Motor Pins
  pinMode(DIR_PIN2, OUTPUT);
  pinMode(enable, OUTPUT);
  digitalWrite(DIR_PIN, LOW);  //Direction outputs required for forward movement
  digitalWrite(DIR_PIN2, HIGH);
  digitalWrite(enable, HIGH);
  Serial.begin(115200);



}
```

Loop Code

```
if (BTvar == 'B' && !B_executed) {  //speed up
  s = constrain(s - 100, 300, 2000);
  B_executed = true;
  BTvar = 'F';
}
if (BTvar != 'B') {
  B_executed = false;
}
if (BTvar == 'X' && !X_executed) {  //slow down
  s = constrain(s + 100, 300, 2000);
  X_executed = true;
  BTvar = 'F';
}
if (BTvar != 'X') {
  X_executed = false;
}
if (BTvar == 'S') {
  digitalWrite(enable, LOW);
  Serial.println("System Unlocked");
  /*
  Reset();
  BTvar = '0';*/
}
if (BTvar == 'C') {
  digitalWrite(enable, HIGH);
  Serial.println("System Locked");
}
if (BTvar == 'F') {
  Forward();
}
if (BTvar == 'L') {
  Left();
}
if (BTvar == 'R') {
  Right();
}
if (BTvar == 'T') {
  Reverse();
}
if (BTvar == 'Y') {
  TurnAroundR();
}
  if (BTvar == 'A') {
  TurnAroundL();
  }
}
```

Control Functions

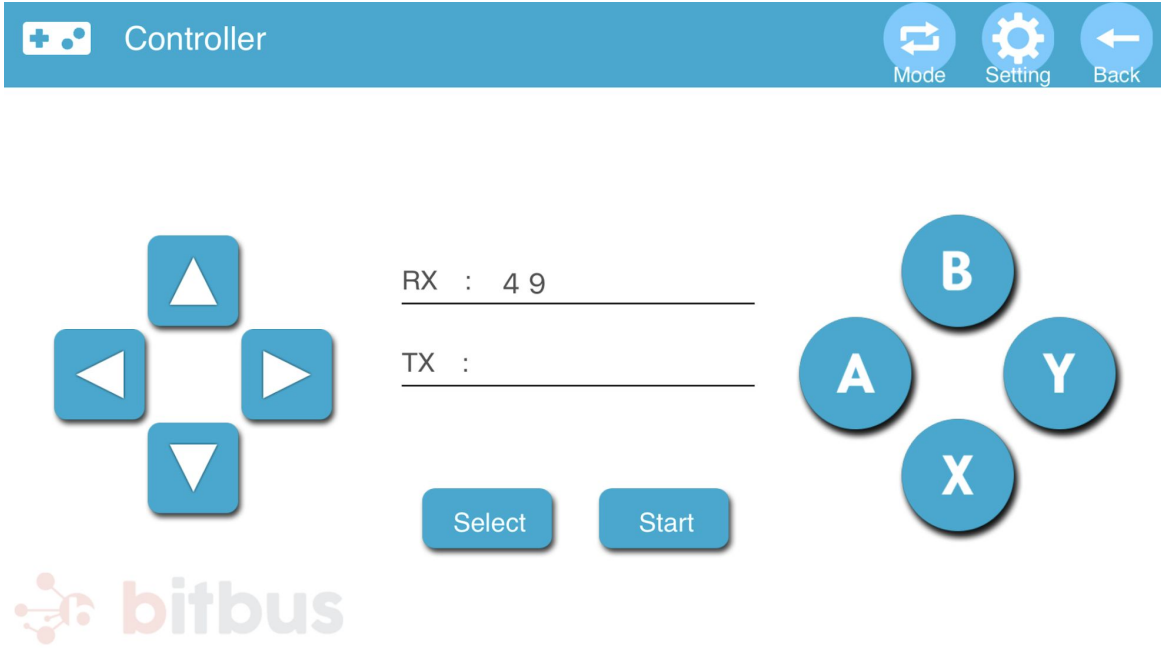
```
void Forward() {
  digitalWrite(DIR_PIN, LOW);
  digitalWrite(DIR_PIN2, HIGH);
  k = 1;
  Motor1Interval = s;
  Motor2Interval = s;
}

void Left() {
  digitalWrite(DIR_PIN, LOW);
  digitalWrite(DIR_PIN2, HIGH);
  k = 1;
  Motor1Interval = 2 * s;
  Motor2Interval = s;
}
void Right() {
  digitalWrite(DIR_PIN, LOW);
  digitalWrite(DIR_PIN2, HIGH);
  k = 1;
  Motor1Interval = s;
  Motor2Interval = 2 * s;
}

void Reverse() {
  digitalWrite(DIR_PIN, HIGH);
  digitalWrite(DIR_PIN2, LOW);
  k = 1;
  Motor1Interval = s;
  Motor2Interval = s;
}

void TurnAroundR() {
  digitalWrite(DIR_PIN, LOW);
  digitalWrite(DIR_PIN2, LOW);
  k = -1;
  Motor1Interval = s;
  Motor2Interval = 2 * s;
}

void TurnAroundL() {
  digitalWrite(DIR_PIN, HIGH);
  digitalWrite(DIR_PIN2, HIGH);
  k = -1;
  Motor1Interval = s;
  Motor2Interval = 2 * s;
}
```

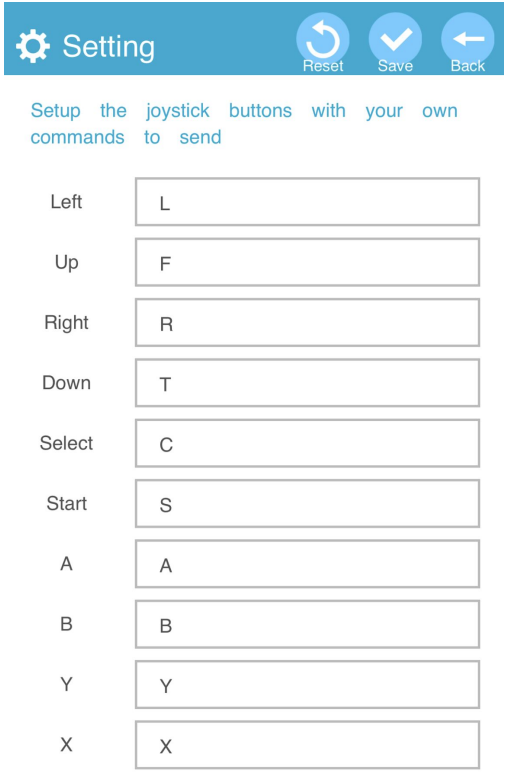
Qiyuan Lu

Bluetooth Controller

UCI Samueli School of Engineering

Controller Interface and Logic

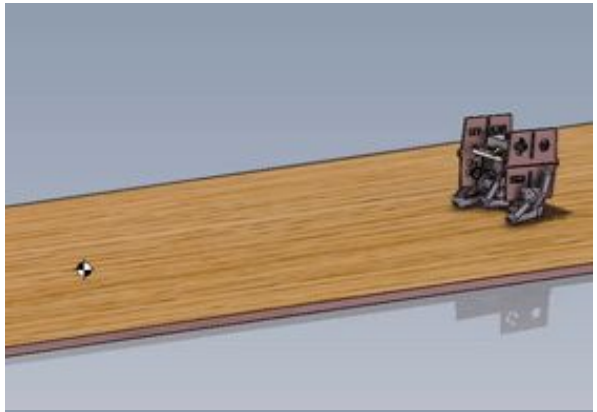




Qiyuan Lu

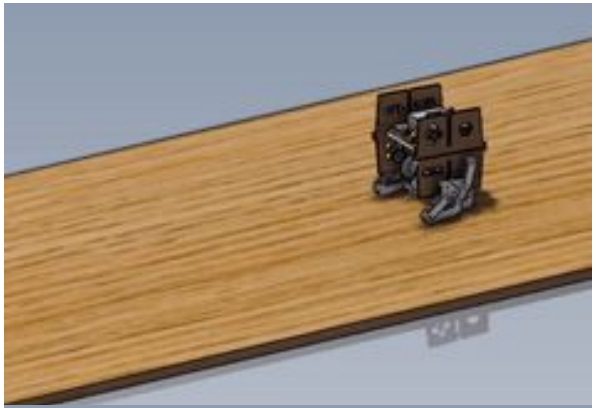
Digital Prototype: Motion Analysis

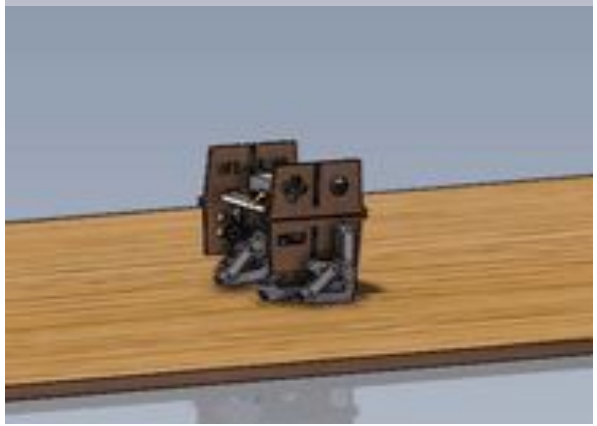
UCI Samueli School of Engineering



Walking Forward

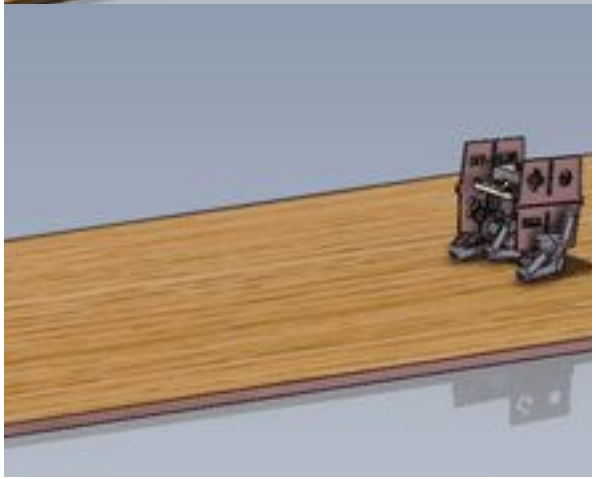
Turning Left





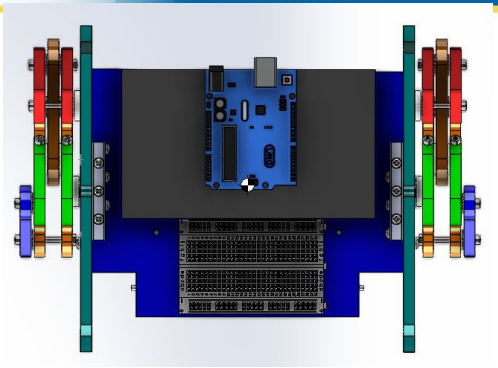
Turning Right

Reversing



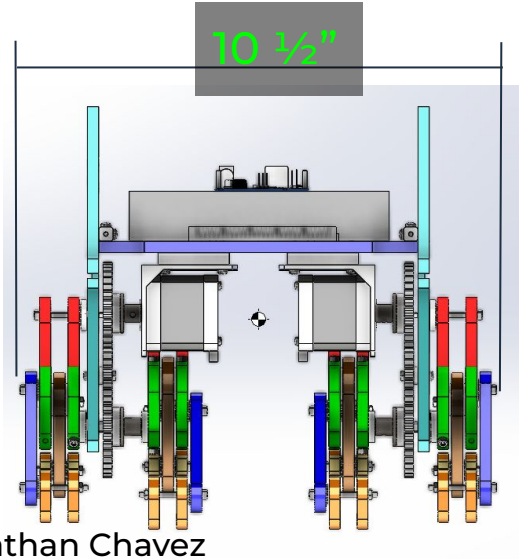
Tom Nguyen

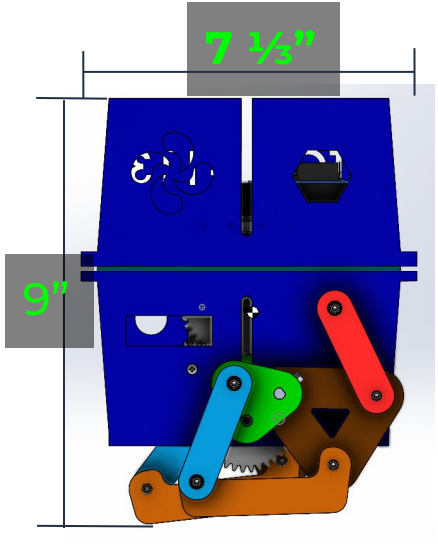
Digital Prototype A

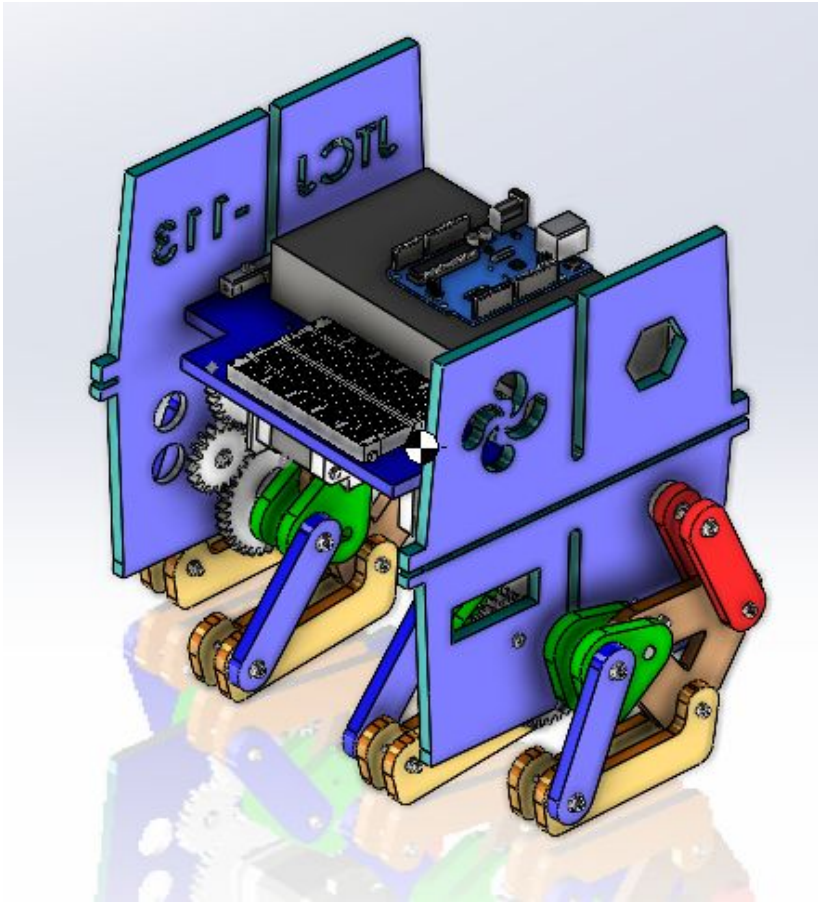


Mass of ~5.5 lbs

Overall Dimensions:
10 ½” x 7 ⅓” x 9”



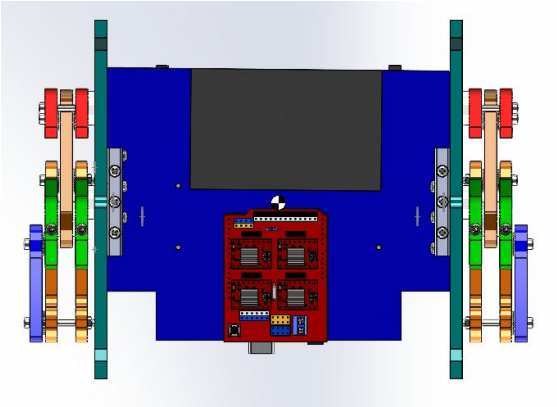




Jonathan Chavez

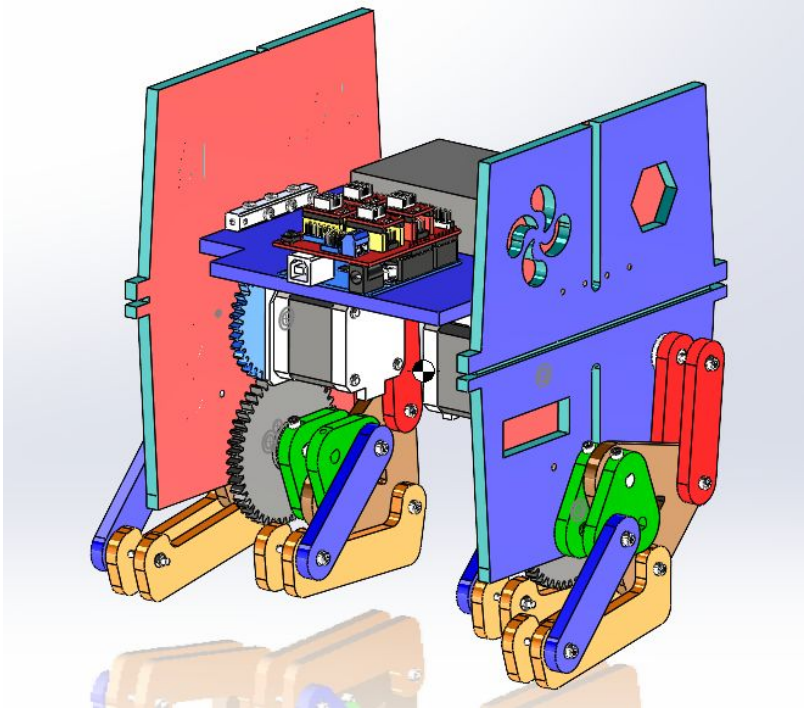
Digital Prototype B

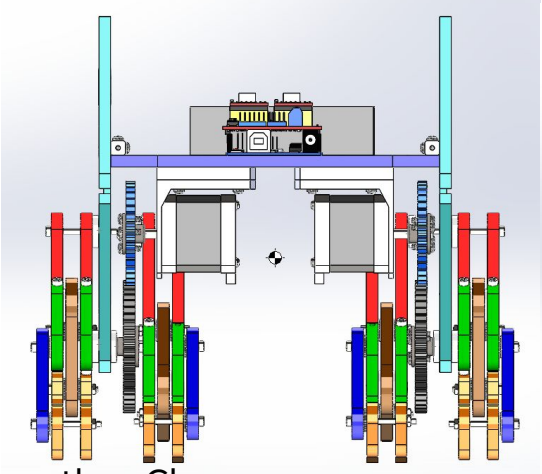
UCI Samueli School of Engineering

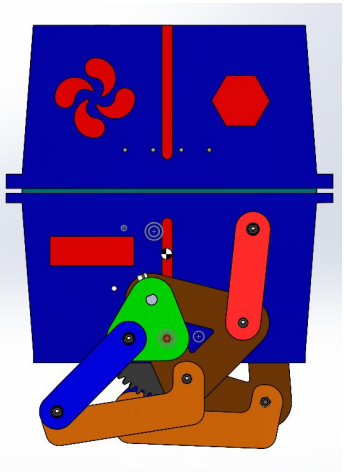


Mass of ~4.83 lbs

Overall Dimensions:
10 ½” x 7 ⅓” x 9”

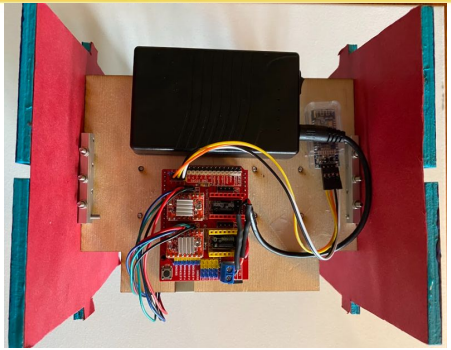






Jonathan Chavez

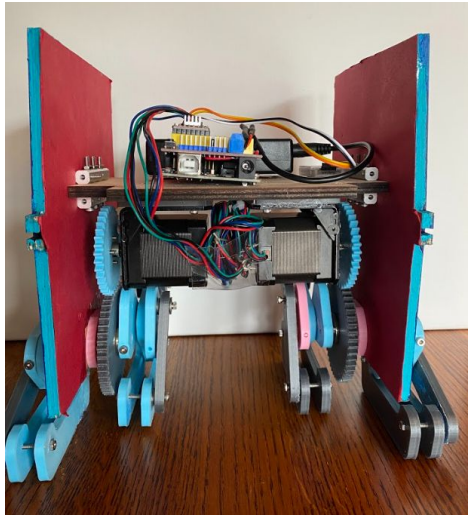
Final Design

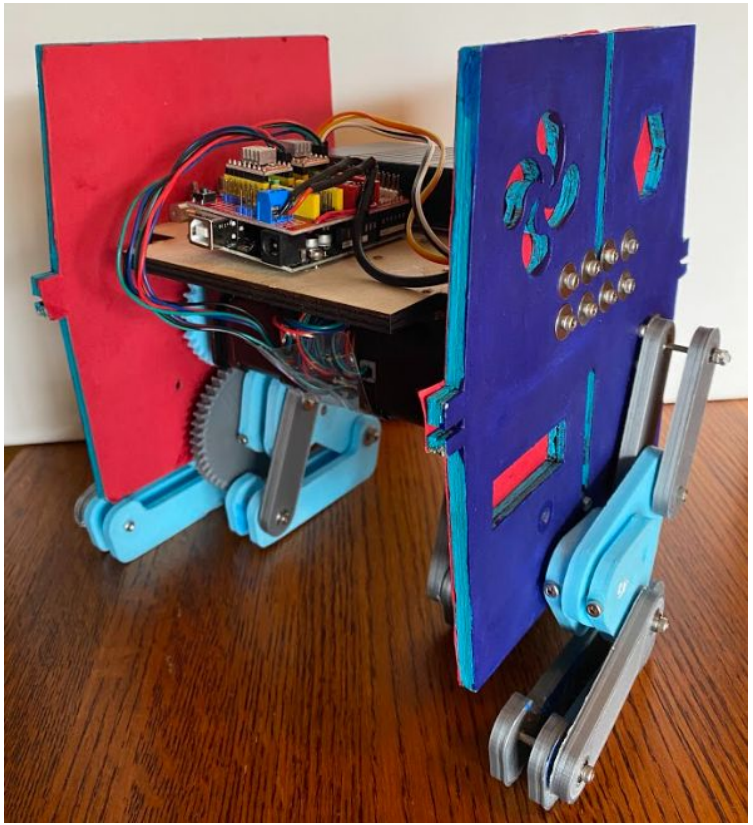


Mass of 4.463 lbs

Overall Dimensions:

$$10\frac{3}{8}" \times 7\frac{5}{16}" \times 9\frac{1}{4}"$$





Jonathan Chavez

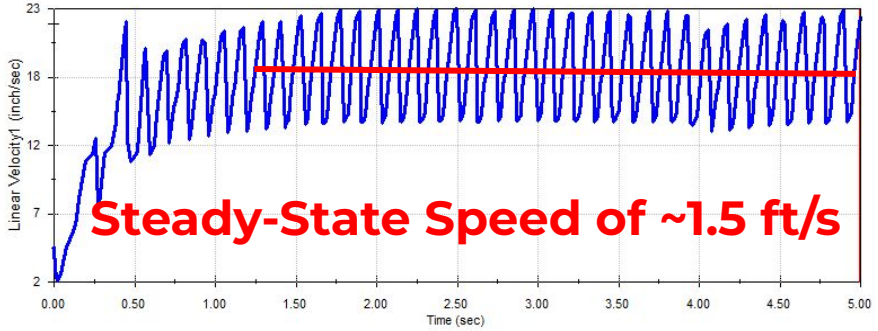
Final Design

UCI Samueli School of Engineering

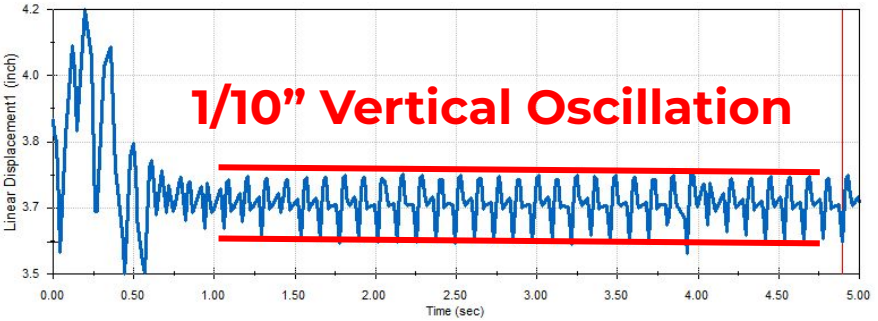
Verification of Requirements

Speed > 1.5 fps	✅
Turning radius < 3ft	✅
Weight < 7 lbs	✅
Tank Steering	✅
Minimal Vertical Oscillation < 1”	✅
Remote control controlling speed and direction of walker	✅
Uses Arduino Uno	✅
DIY Manufacturing	✅

Motion Analysis Speed of Final Design



Oscillation of CG while walking of Final Design

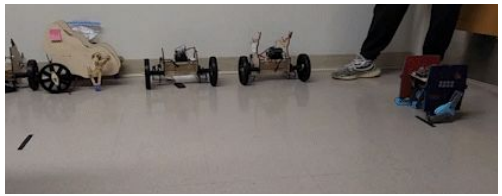


Final Design Performance Review

UCI Samueli School of Engineering

Validations of Requirements

- Weight of assembly recorded as 4 lbs 7.2 oz
- Top speed recorded at 1.1 fps
  - Time to walk 5 feet was 4.64 sec
- Smallest diameter it can turn at top speed approximately 1.5 ft



Risk Assessment

- Assessment to find highest RPM without tipping
  - Found to be 250 rpm due to insufficient torque from motor at that speed

Qiyuan Lu

Closing Statements

UCI Samueli School of Engineering

Questions and concerns:

- What could be done to make the walker faster?
- Would more durable materials make it too heavy?
- What is the actual difference between using gear motors vs stepper motors?
- What is the physical size limit of this walker with available technology?

Recommendations for the Future:

- Different stepper motor drivers that outputs more current
- Metallic/acrylic leg mechanism for more durability and reliability
- Check and verify dimensions carefully before manufacturing, prevent redundancy
- Implementation of sensors that would allow for autonomous control

Tom Nguyen

Demonstration of the Hardware

Thank You
Any questions?