Final Presentation Project 3 - Jansen's Pet

Mechanical Design Team Lead: Thanh-Truc Ngo
Electrical Design Team Lead: Marc Ono
Project Sponsor: Professor McCarthy

Marc

Steerable Walker Project

Project Definition & Overview

We have designed and built a two-legged mechanical walker that can be remote controlled and can be steered.

Important Characteristics of the Design

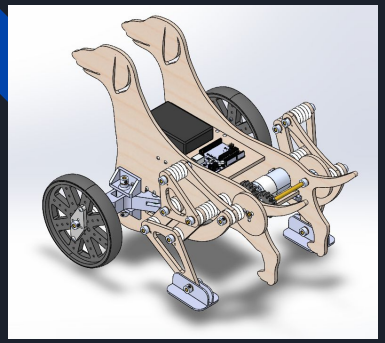
Two Jansen-style legs in the rear to drive the mechanism and two wheels in the front with a steering linkage to steer it.

It fits in a 20”x12”x10” box and weighs about 5 lbs.

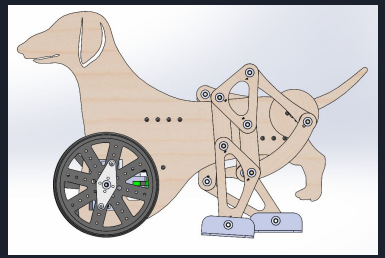
Battery powered and controlled with an infrared remote.

Marc

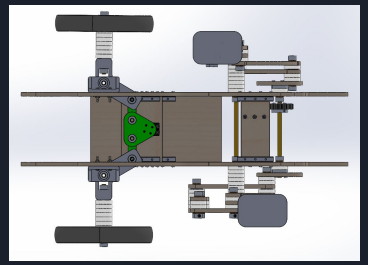
Prototype B



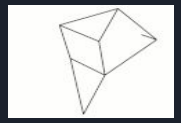
Isometric View



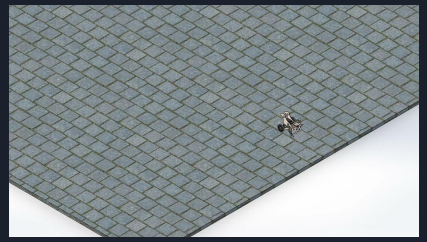
Front View



Bottom View (Steering Mechanism)

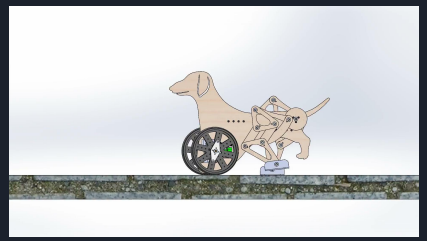


"Jansen's Linkage" From Wikipedia

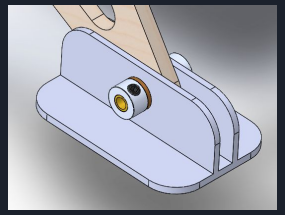


Walker Going Straight

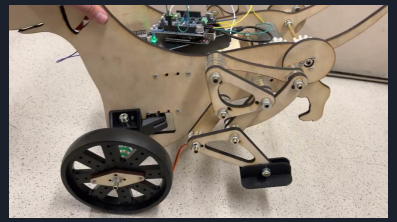
15 Degrees Right Turn



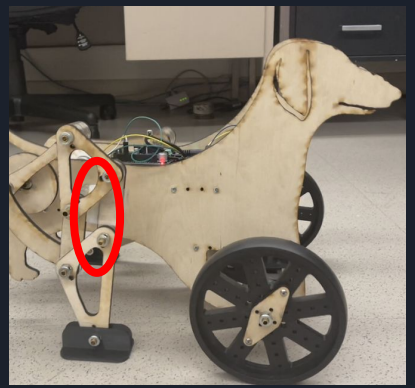
15 Degrees Left Turn with Added Weight



Feet

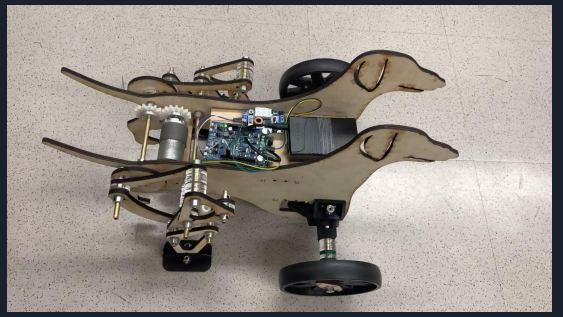


Singularity

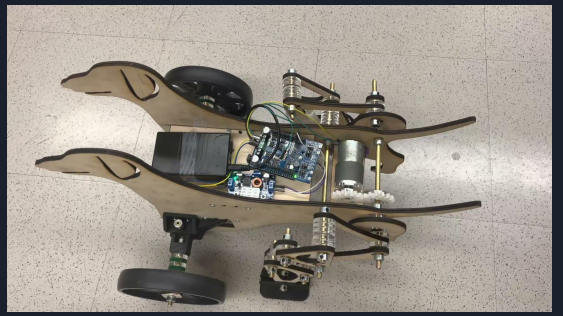


Solution

Walker Going Straight



Walker Turning Left and then Right

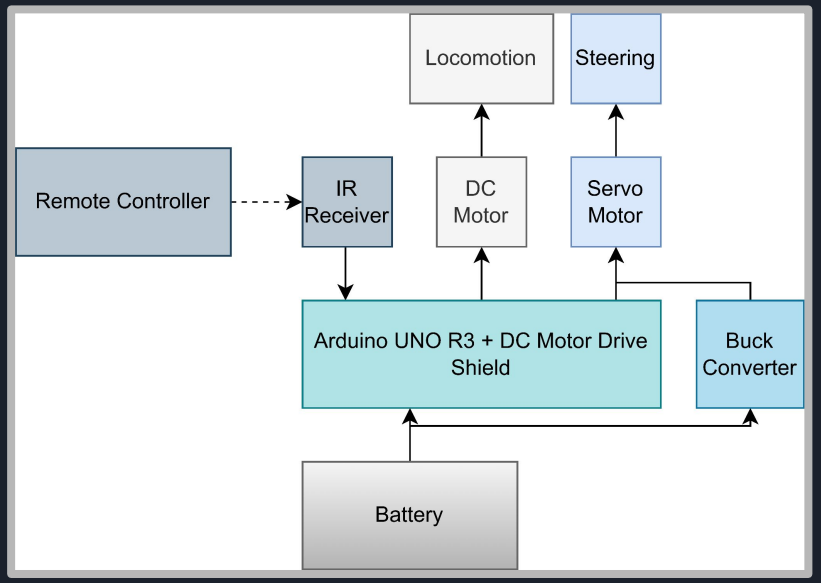


Steering Mechanism

Thanh

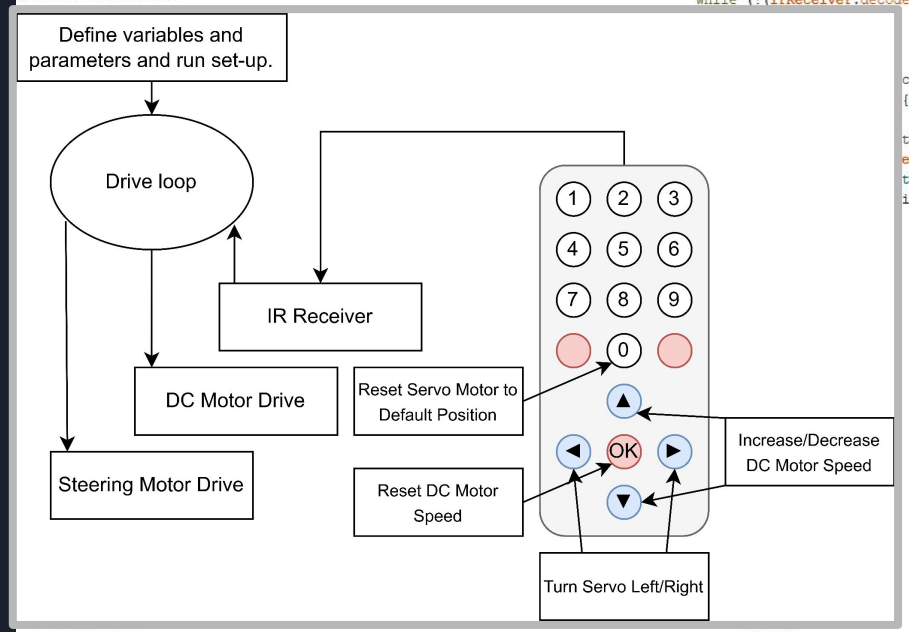
Electronics

- Arduino UNO R3
- IR controls
- DC motor for legs, servo motor for steering
- Battery Powered



Marc

Software




```
//Cytron Motor Shield Library
#include "CytronMotorDriver.h"

//IR Remote Library
#include <IRremote.h>

//Servo Library -- diables PWM on pins 9 and 10
#include <Servo.h>
```

```
//Start IR receiver on the arduino
//pin number and system command DISABLE_LED_FEEDBACK
//disables led feedback when IR signal is received
IrReceiver.begin(IR_RECEIVE_PIN, DISABLE_LED_FEEDBACK);

//Attach servo to a pin
servo1.attach(servo_pin);
```

```
  //Turn off motors
  motor.setSpeed(0);
}

void loop() {

  //while the IR sensor is ac
  while (!(IrReceiver.decode
```

```
    default:
      Serial.println("Input no
      break;

  }

  Check_counter();

  pwm_value = motor_counter *
  motor.setSpeed(pwm_value);
  Serial.println(motor_counter

  servo_angle = servo_counter
  servo1.write(90 + servo_angl
  Serial.println(servo_counter
```

```
    break;

  //OK button
  case 28:
    Stop();
    break;

  // 0
```

```
}
```

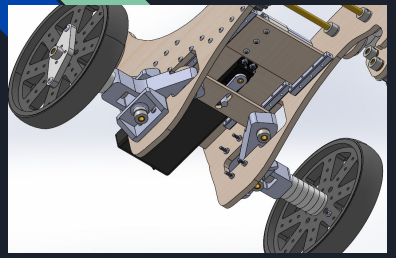
```
//Functions
//increase forward counter
void Forward() {
  motor_counter = ++motor_counter;
  return;
}

//decrease forward counter
void Backwards() {
  motor_counter = --motor_counter;
  return;
}

//increase servo motor counter
void Right() {
  servo_counter = ++servo_counter;
  return;
}

//decrease servo motor counter
void Left() {
  servo_counter = --servo_counter;
  return;
}

//stop motors
void Stop() {
  motor_counter = 0;
  return;
```

```
//Define variables for the counter and angle for servo
float servo_counter = 0;
float servo_angle = 0;
const float servo_counter_max = 10;
const float servo_increment = 15 / servo_counter_max; //max servo angle is 15
//servo_angle = servo_counter * servo_increment (defined later)
```

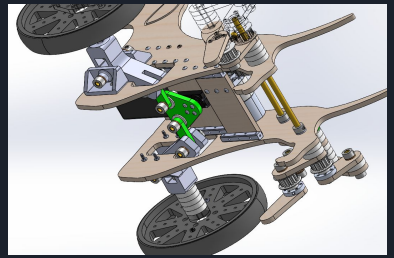
Marc

Prototype A

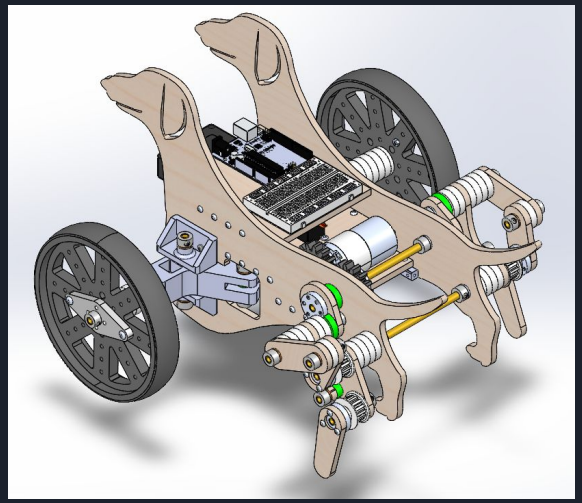
Before



After

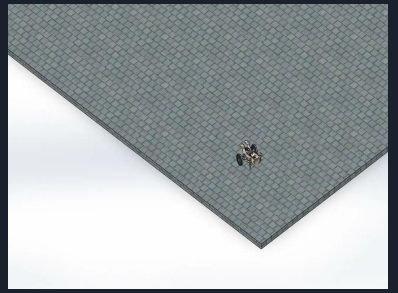


Bell crank changed from slot to two separate tie rods connected to a pivot ⇒ reduces friction

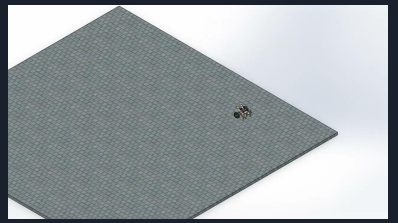


Isometric View of Walker in SolidWorks

Motion Analysis of Walker Going Straight

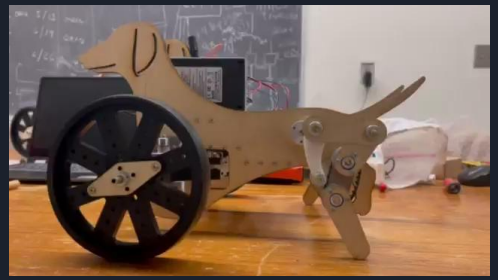


Walker Turning at 15 Degrees to the Right



Walker Turning at 15 Degrees to the Left

Performance



- Belt not tight enough
- Linkage interfering with belt

Steering Mechanism

Thanh

Conclusions

- Designing a steerable mechanical walker integrates mechanical design, electronics, and software
- A walker is sensitive to the location of the center of gravity, because the forces in the walker change as weight is shifted from one foot to the other.
- Future recommendations
  - Try to make lighter and faster
  - Make it 4 legs instead of 2 wheels and 2 legs
  - Adding lights, sound
  - Adding sensors
    - Pathfinding
    - Food delivery

Marc