Salamander Robot Team (LM2) Final Presentation:

Design and Control of a Highly-Articulated Salamander-Inspired Robot for Future Search and Rescue Applications

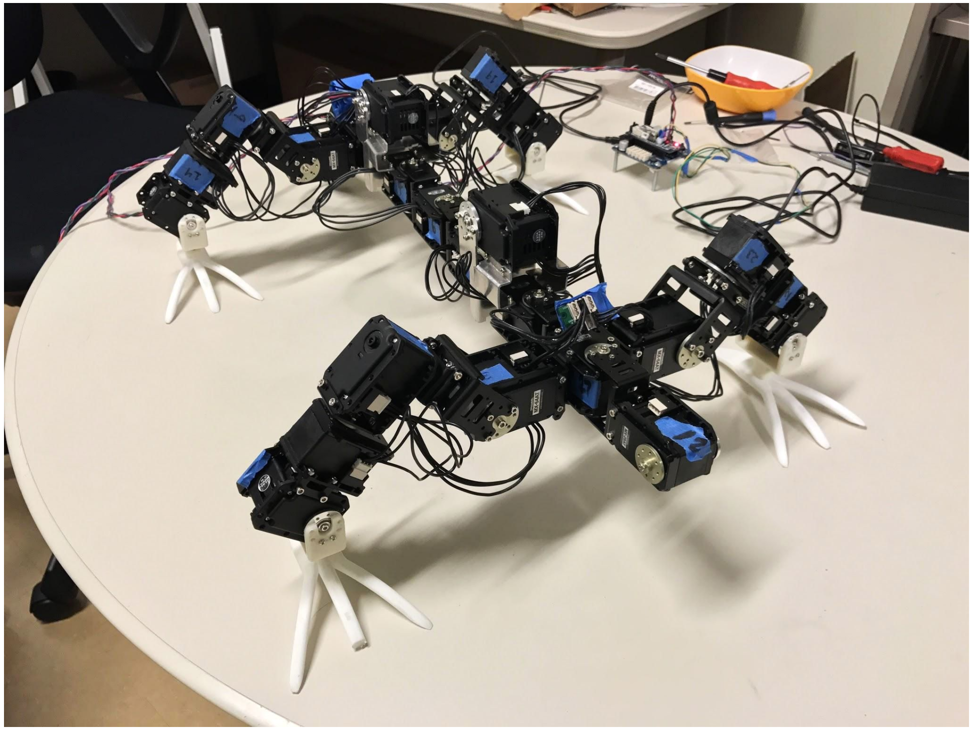
Austin Bush, Sunit Kulkarni, Hariank Mistry, Alex Popescu, Jonathan Rundquist, Shashwat Sitesh, Brian Weaver, Calvin Yao

Advisors: L.S. Milor (ECE), P.A. Vela (ECE), A.S. Jariwala (ME), M.L. Realff (MSE)

April 20, 2017

Outline

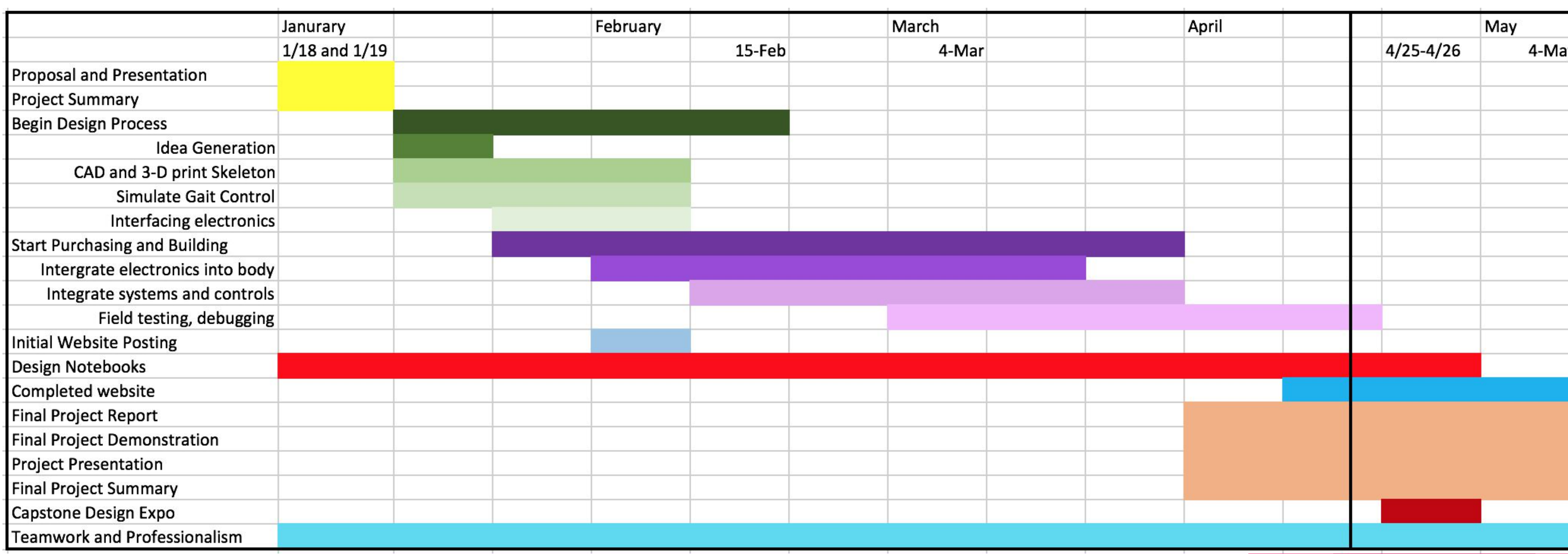
I. Introduction
II. Updated Goals
III. Updated Specifications
IV. Mechanical Design
    A. Design Approach
    B. Results & Verification
V. Interfacing
    A. Design Approach
    B. Results & Verification
VI. Controls
    A. Design Approach
    B. Results & Verification
VII. Lessons Learned
VIII. Future Work



Introduction

- <u>Goal of presentation:</u> present results and current status of project, as well as design choices and methodologies.

Current Status: A walking robot, ready for quantitative testing



Mechanical Design

Design and Fabrication
of the
Body, Feet, Tail, Head

Jonathan Rundquist

Austin Bush

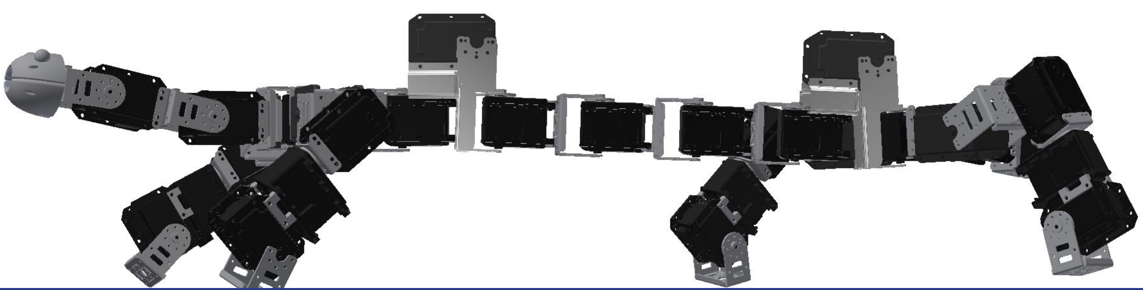
Brian Weaver

Sunit Kulkarni

Key Design Constraints & Limitations

- ✓ Small and agile enough to navigate under unknown rubble
- ✓ Light enough to be easily deployed to disaster environments
  - Must be able to perform reliably in numerous disaster environments

Needs more testing



Updated Specifications

Item	Design Specs	Actual Specs
Weight	< 25 kg	~5 kg
Bounding volume	< 2 x 0.5 x 0.5 m	0.63 x 0.41 x 0.2 m
Traversable terrain height deviation	> 5 cm	-
Traversable grade	> 3 %	-
Turn radius	< 2 m	1m (simulation)
Traversal speed	> 10m / minute	6m / min
Man-portable	Yes	Yes
Tether length	> 3 m	2.44 m

Possible Future Functionality
Obstacle avoidance
Autonomous searching
Camera-enabled localization
Environment mapping
Localization

Improvement on Past Designs

Increased Mobility

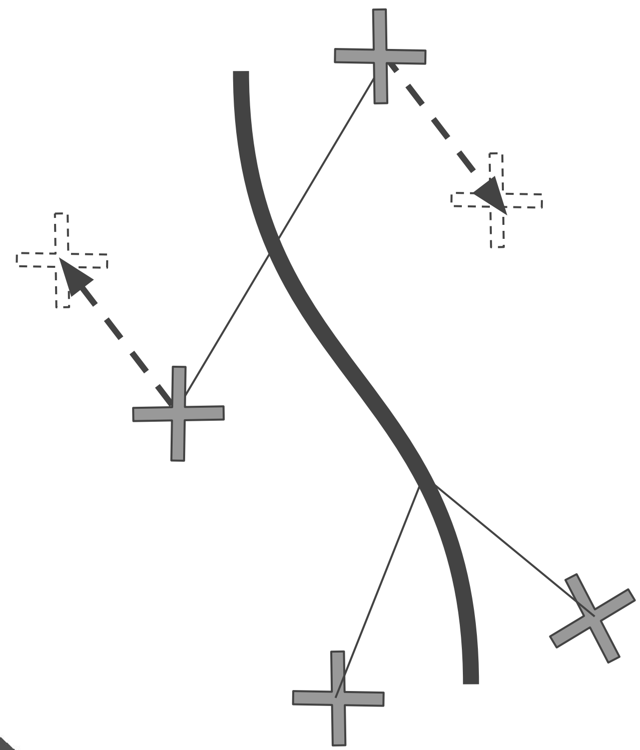
- 26 DOF adding more control to the robot
- Compliant feet increasing traversability
- Vertical mobility in the spine for increased maneuverability

Leg Design

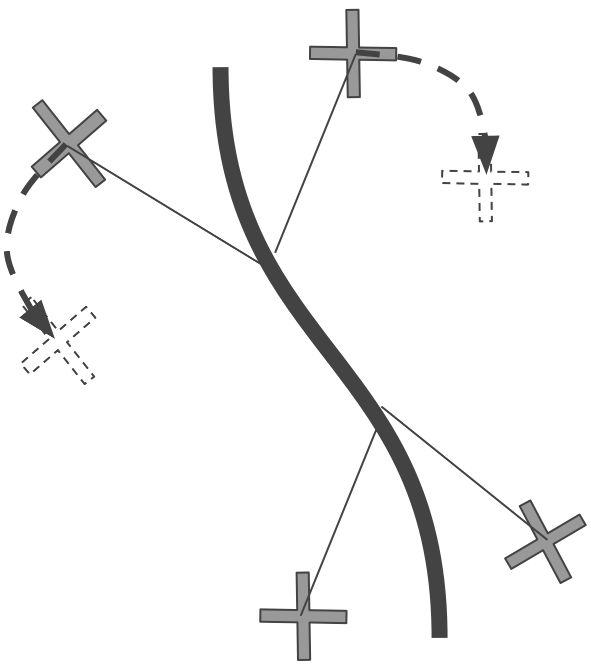
4 Degrees of Freedom
Per Leg

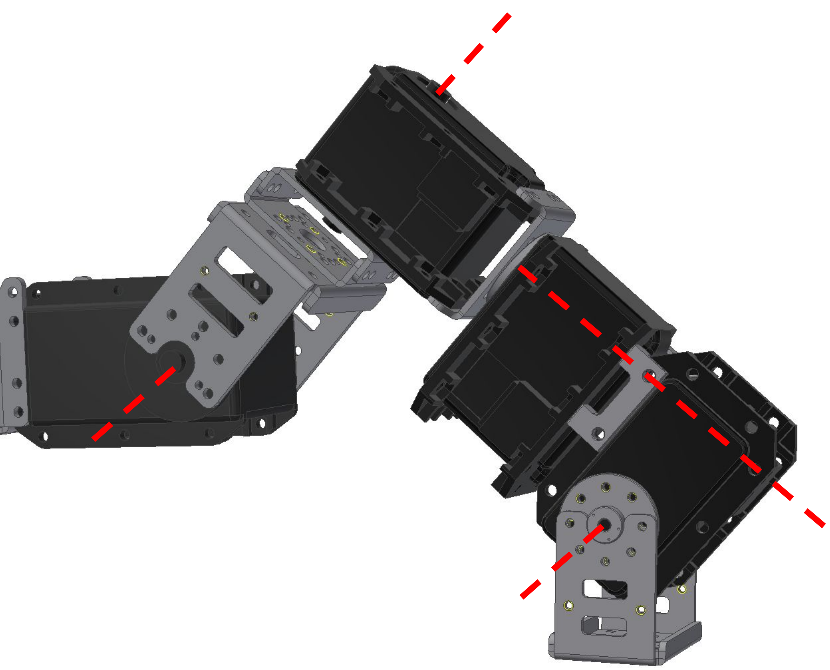
5 with shared
Shoulder Joint

Opposite but Linear Path



Independent but Radial Path

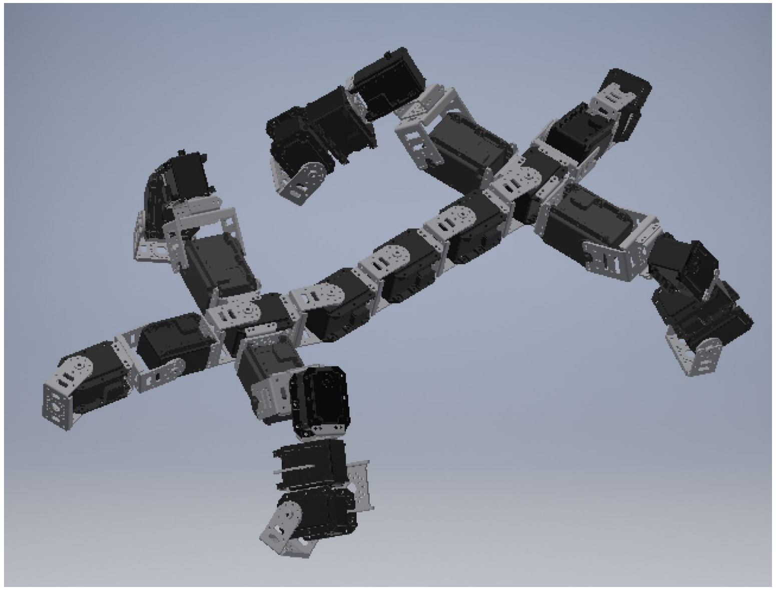


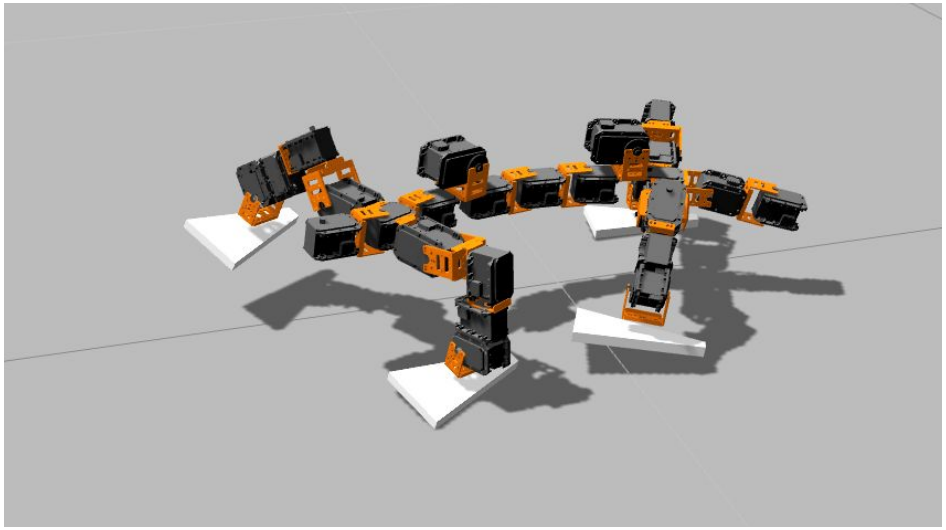


Spine Design

Talk about:

- Spine needed to be long enough to support the desired walking gait (lgs don't run into each other)
- 6 joints in total (without tail)
- Spine needed more vertical mobility for stairs and such
- Two vertical motors were mounted to the spine, giving larger walking range and vertical movement



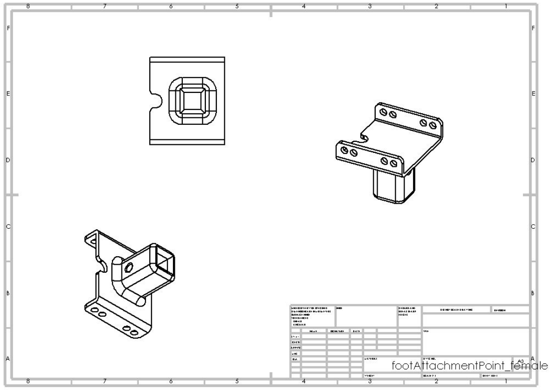


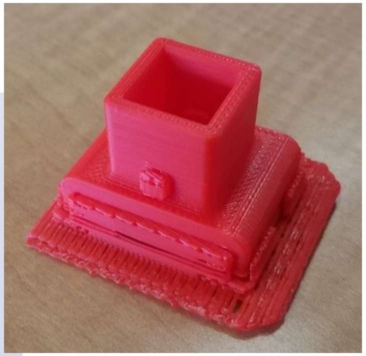
Foot Design
Concept Generation

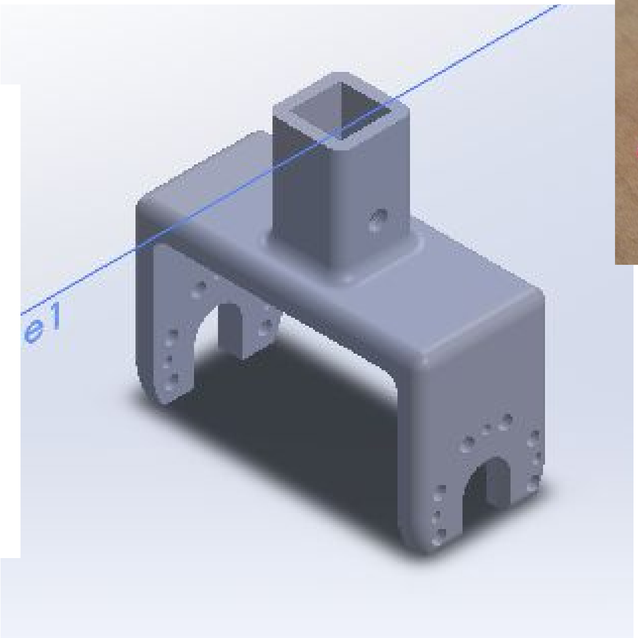
- Group effort to create a multitude of possibilities
- Not all concepts were practical
- Needed a way of attaching numerous foot designs for testing

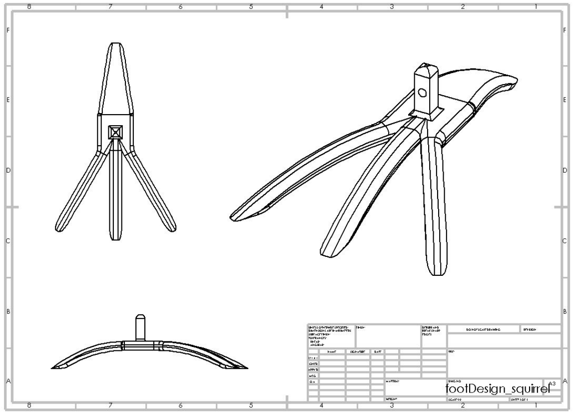
Foot Design
Interfacing

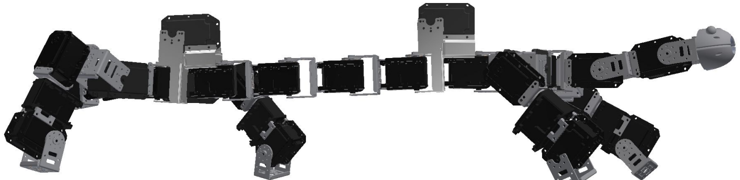
- Developed the Foot Attachment Point models
- Several different iterations as robot design evolved
- Allowed for easy iteration of foot designs





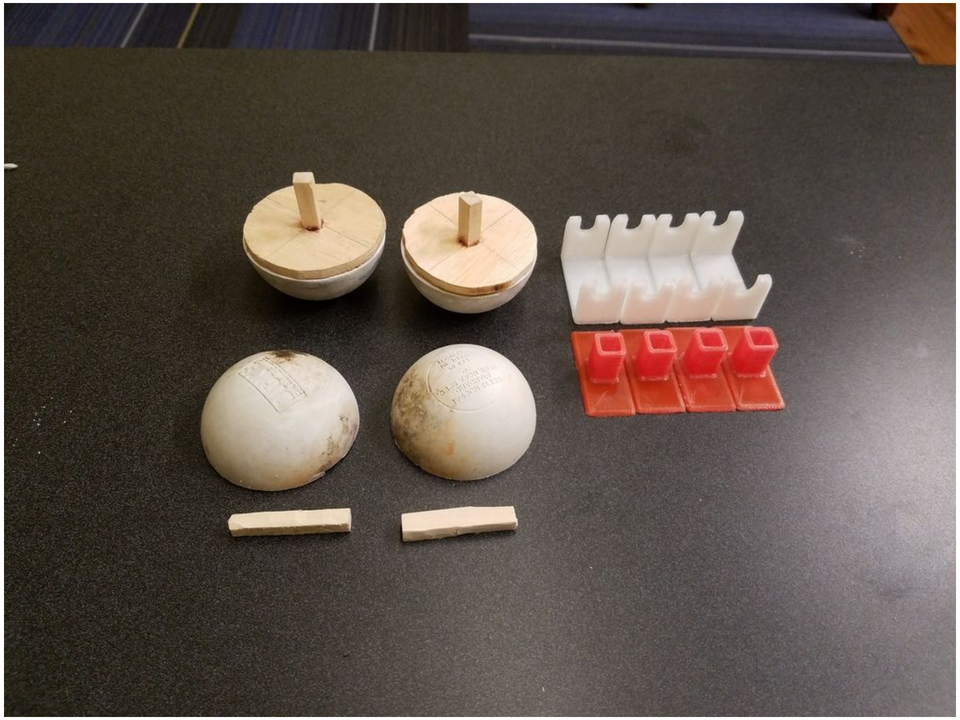






Foot Design
Recycled Ball Feet

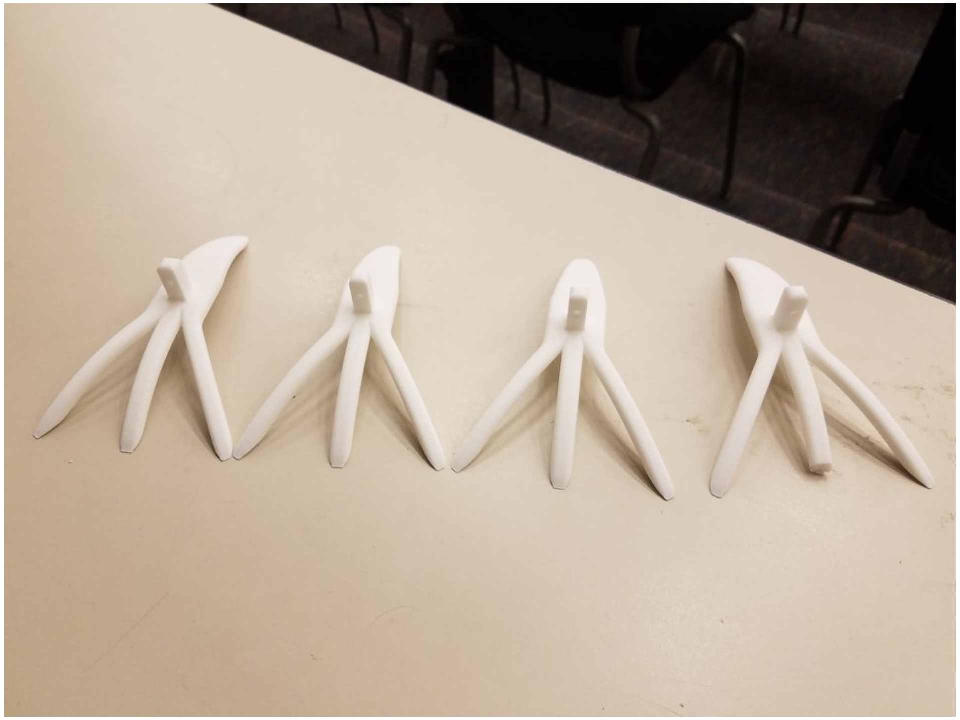
- Made of half a Lacrosse ball
- Good traction on flat surfaces
- Easy to develop gaits

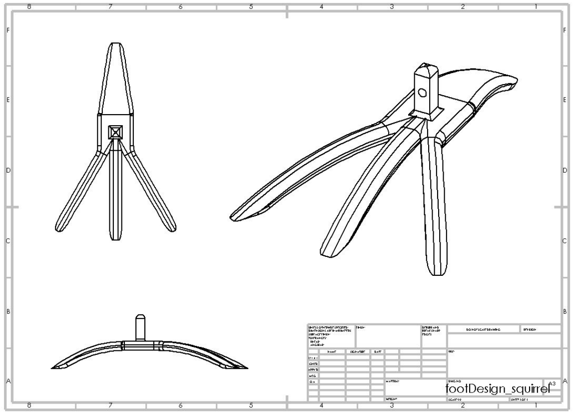


Foot Design

Bio inspired rigid

- Good balance
- Potential climbing modification
- Requires more complex gaits





Foot Design

Compliant advanced design

- Good off road performance
- Still in prototype phase

Head and Tail

Head

- Houses camera, speaker and mic
- Improves stability
- Can grow to house hardware in the future

Tail

- Improves walking and climbing stability

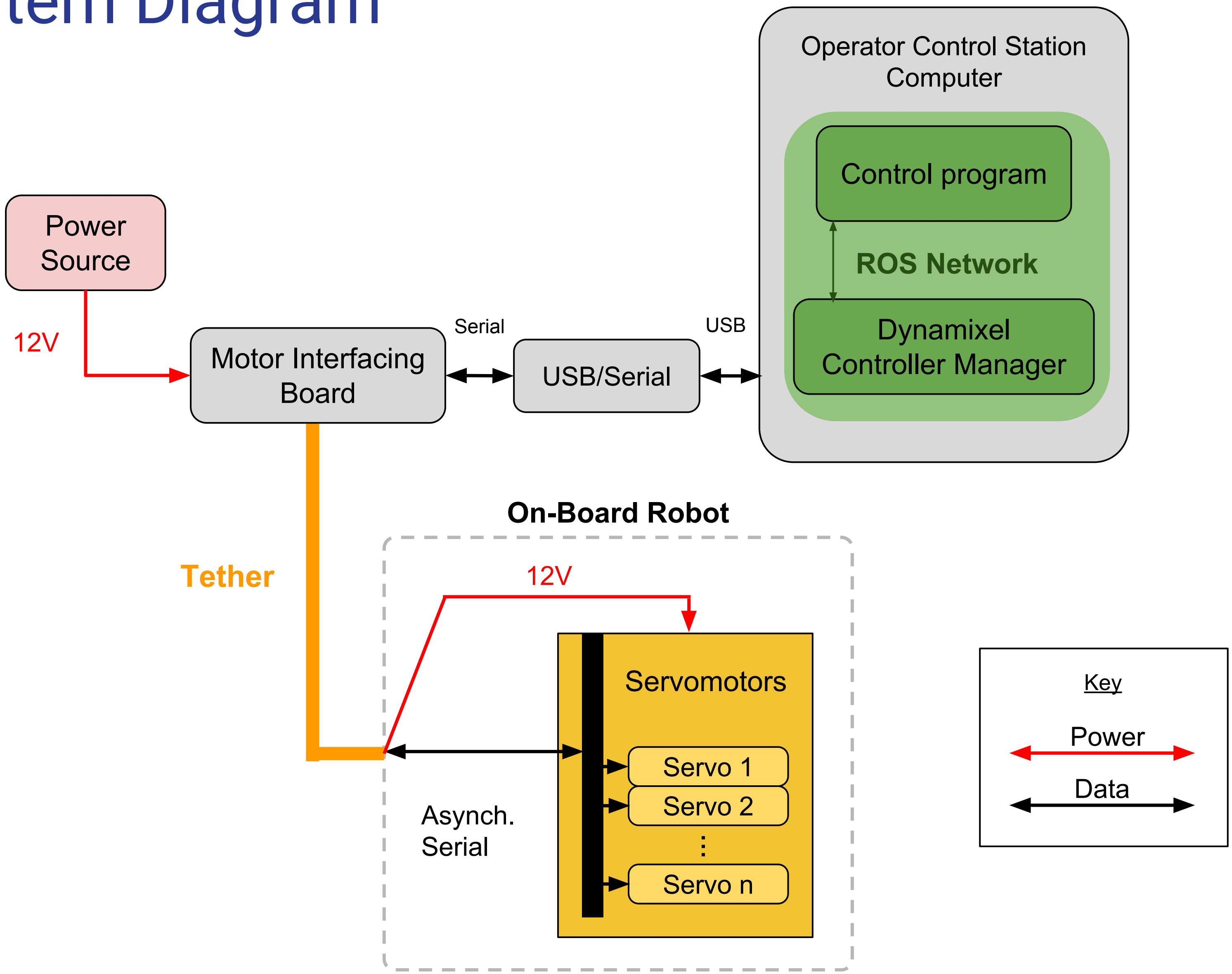
Interfacing

Hardware Development for Control and Operation of the Robot

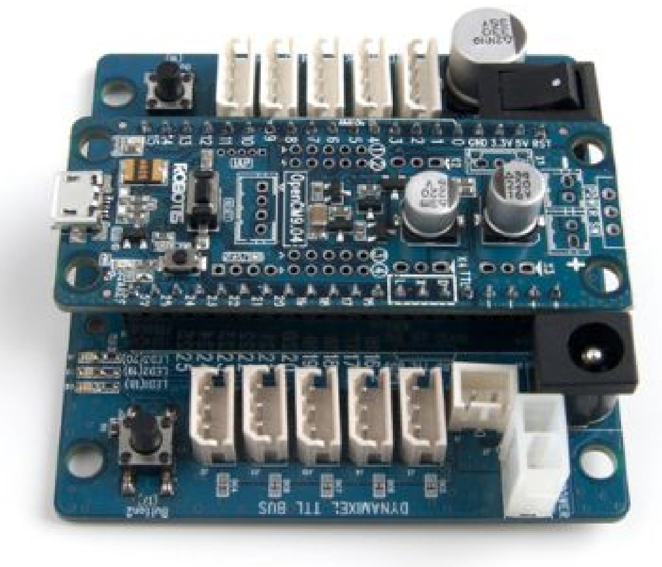
Shashwat Sitesh

Hariank Mistry

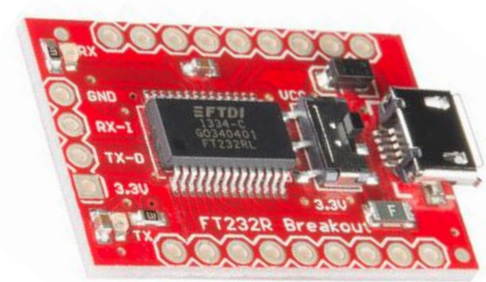
System Diagram



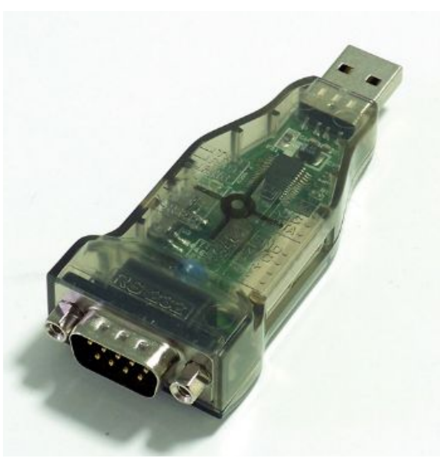
Hardware Interfacing



- OpenCM 9.04 Microcontroller mounted on OpenCM 485 Expansion Pack to connect to the dynamixel servo

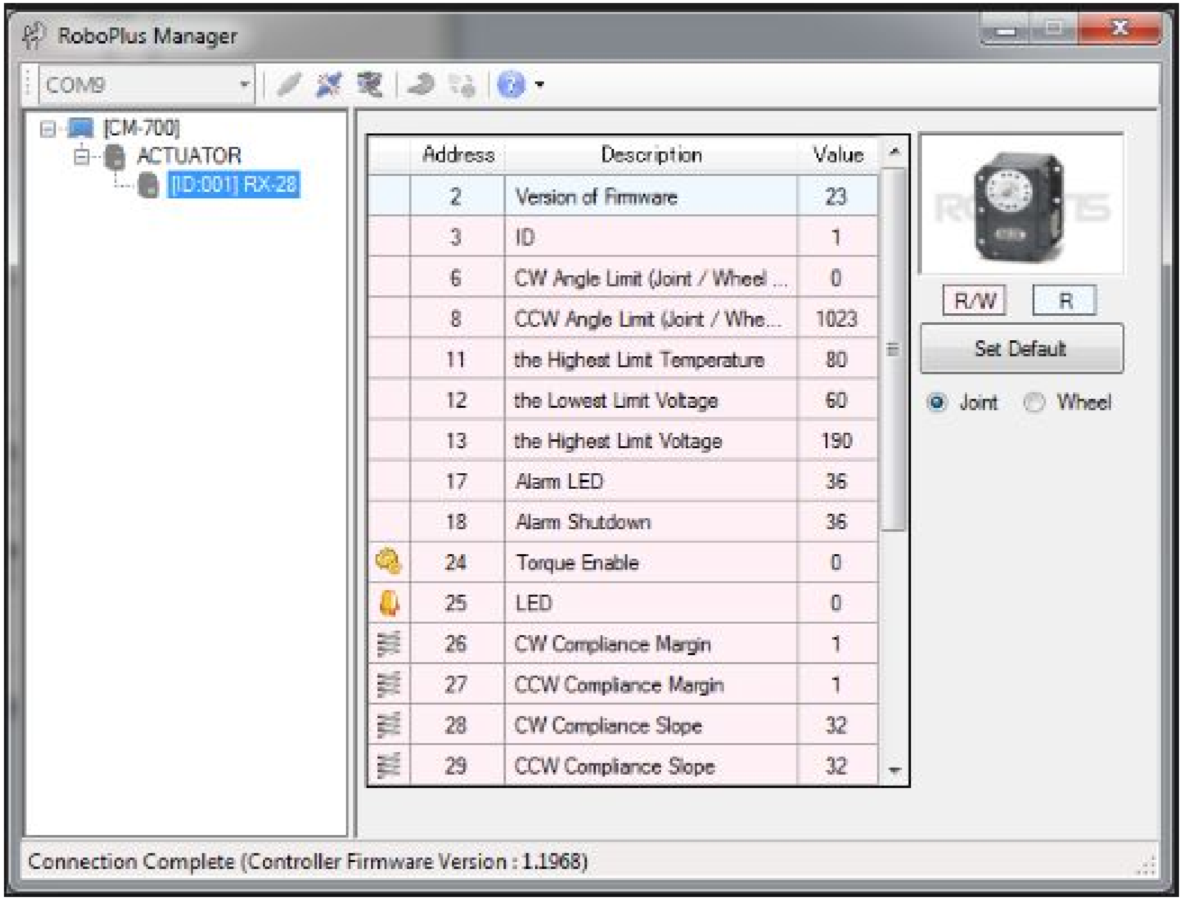


- FT23R USB to Serial Interface Board



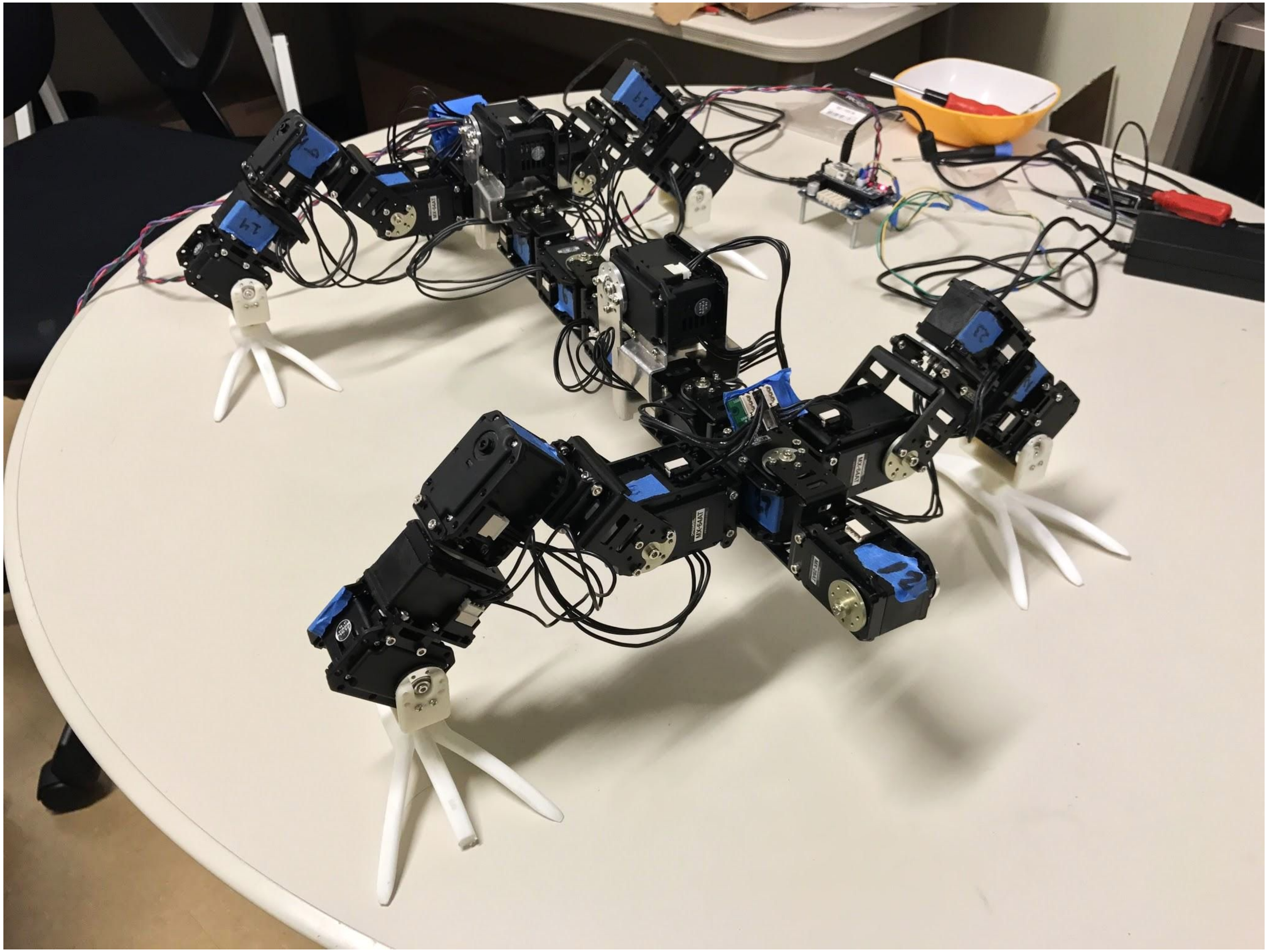
- USB2Dynamixel used to update firmware and baudrate

Hardware Interfacing



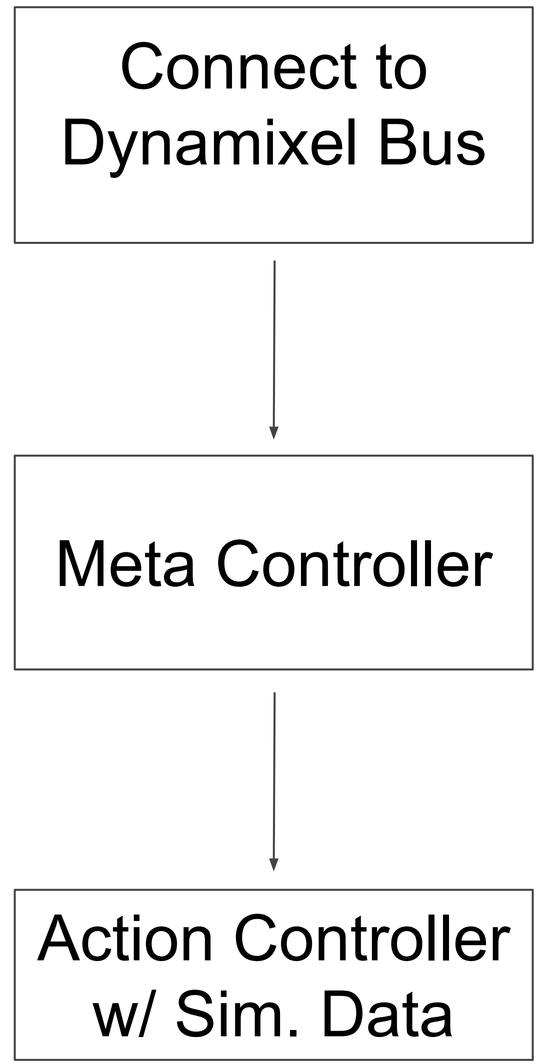
- RoboPlus is a PC based tool that can be used to set parameters

Hardware Interfacing



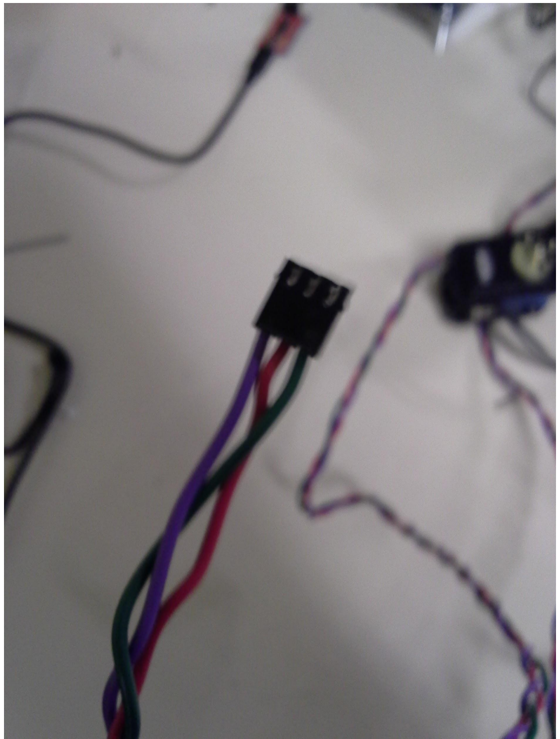
Software Interfacing: ROS

- Use Dynamixel ROS Stack: Python and C++ Interface for communicating to motors via serial commands
- Motors are controlled with an Action Server, which coordinates sending position and velocity commands
- ROS interface is modular and extendable so future robot users can easily program custom trajectories



Power Tether

- Built 3m long tether using 20 gauge wire
- Capable of providing approximately 6A of current and 72W of power
- Depending on the gait algorithm, more power may be required to control more motors at once



Controls

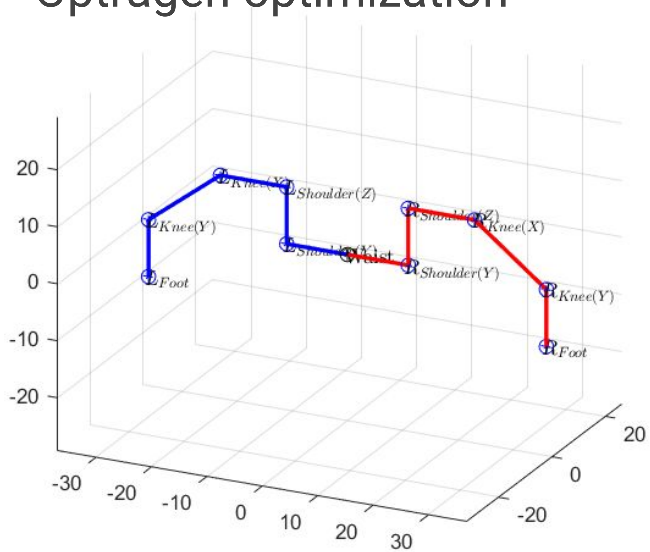
Salamander Gait Control

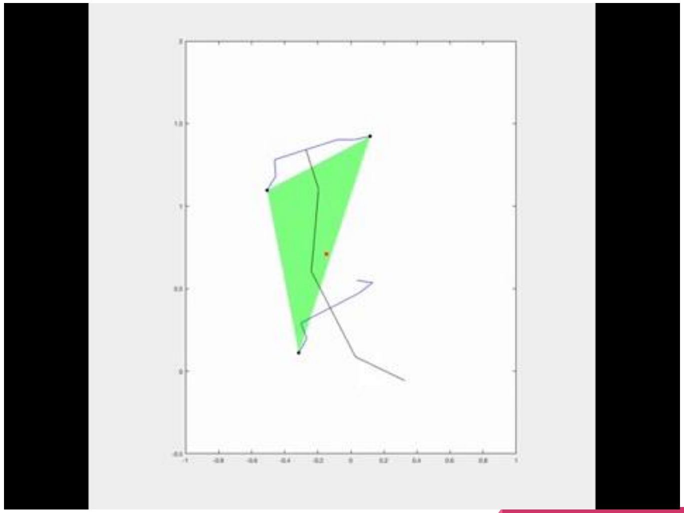
Alex Popescu

Calvin Yao

Design Approach: Kinematic Modeling

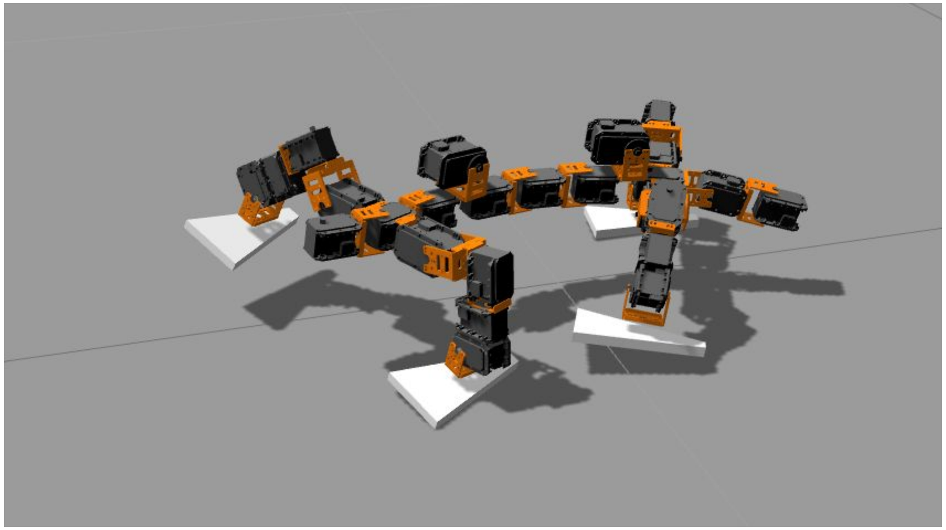
- MATLAB kinematic modeling of legs
- Spine movement with “locking down” feet
- Optragen optimization





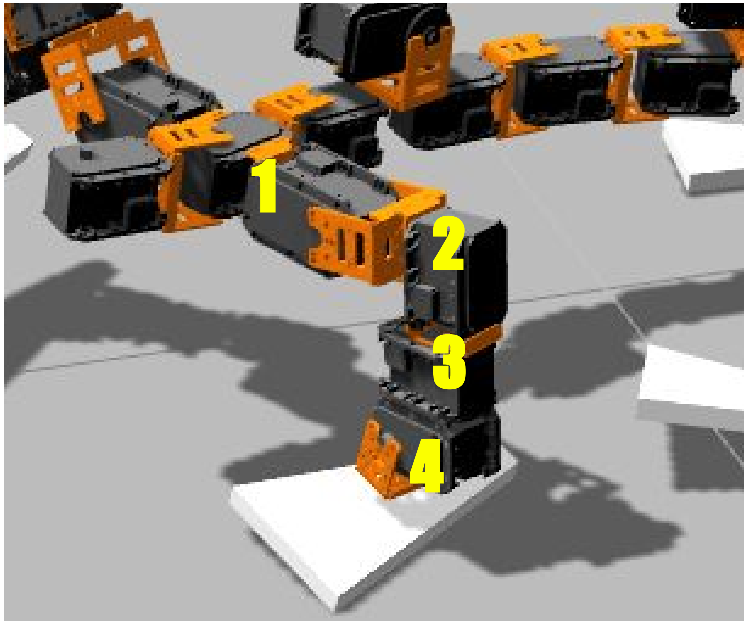
Dynamics Modeling and Simulation

- Benefits of simulation
  - No possibility of robot damage
  - 2-5X faster than real-time, parallelized processing
  - No cost to purchase parts
- Gazebo dynamics simulation can simulate
  - PID control of motors
  - Different foot geometries
  - Foot slippage and ground reaction forces
  - Complicated terrain
- Control of each joint is achieved using a custom C++ shared library "plugin"



Gait Parameterization

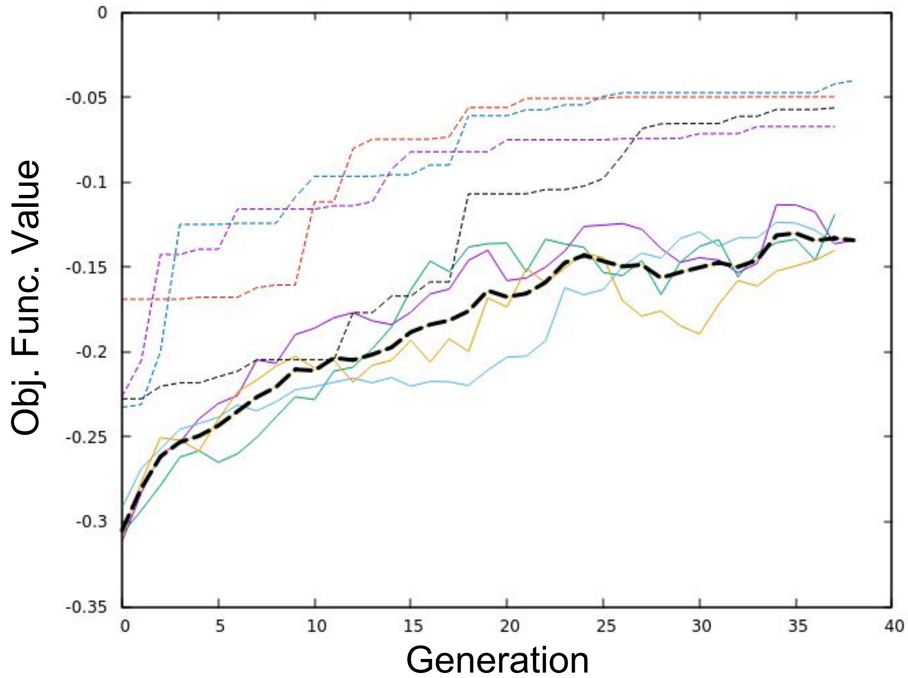
- Used 19 numbers to represent a gait
  - Spine amplitude, phase
  - For each leg:
    - J1 amplitude, phase, bias
    - J2 amplitude, phase
    - J4 amplitude, phase, bias
- Each joint is a sine wave with 3 parameters: bias, amplitude, and phase



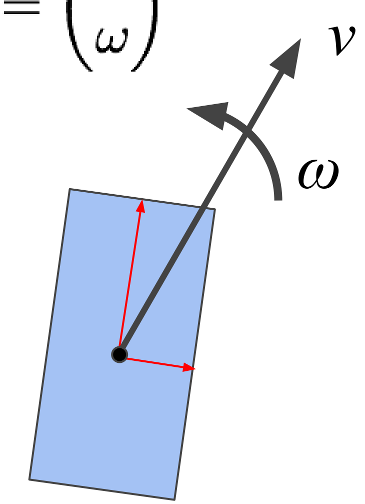
Gait Optimization with Genetic Algorithm

- Benefits of Genetic Algorithms (GA):
  - Robust to many local minima in objective function
  - Objective fn. derivative unknown
  - Works for non-smooth functions

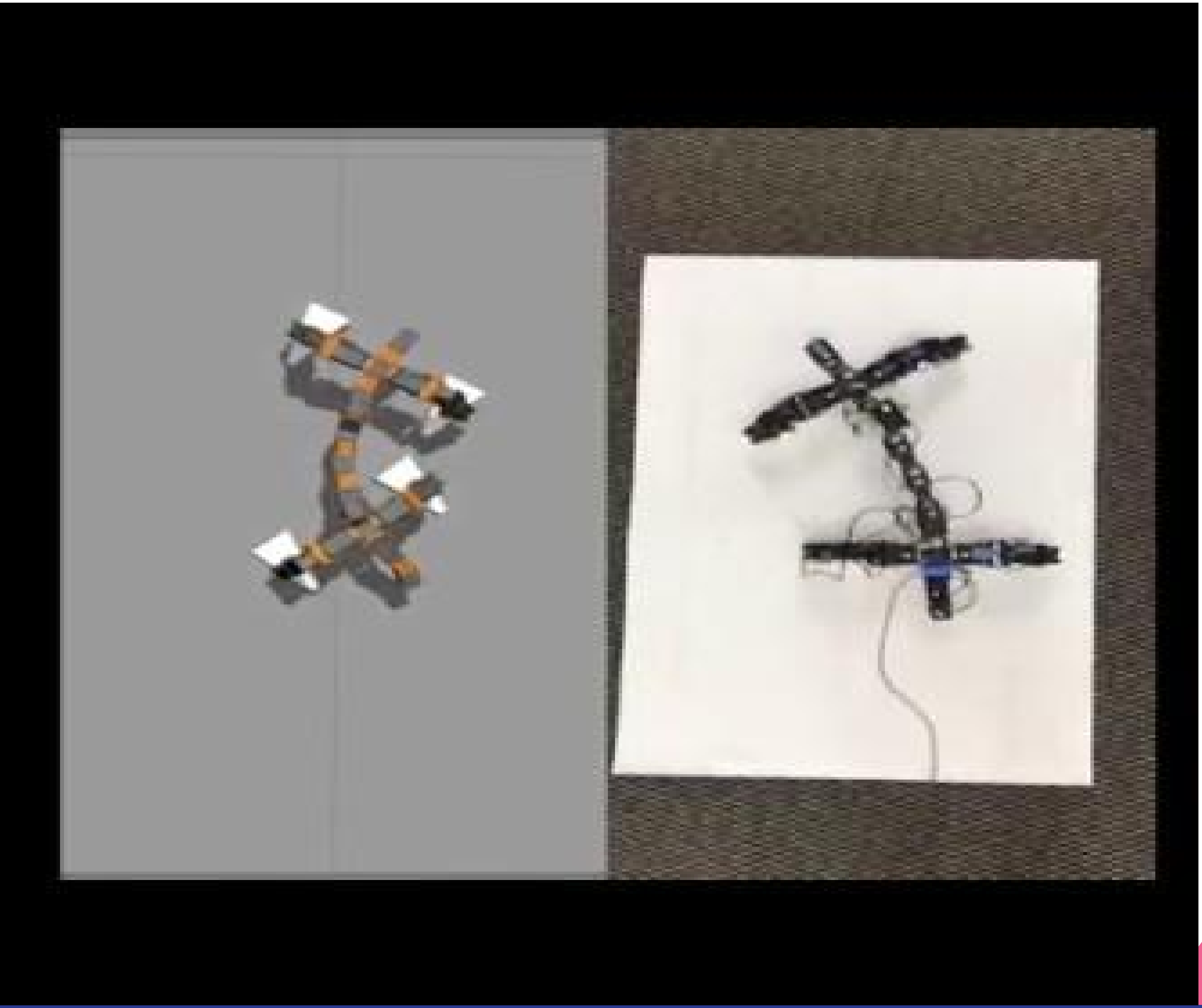
$$f = (\xi - \xi_{goal})^2 - C_{energy}$$



$$\xi = \begin{pmatrix} \vec{v} \\ \omega \end{pmatrix}$$



Results: Simulation vs. Experiment



Simulation: 0.5x speed
Real video: 1.5x speed

Current Task: Operator Control

- Map goal twist xi to a gait parameter vector
- Linearly interpolate between known gaits
- Then, an operator can command the robot twist
- Joystick control: demo goal

Lessons Learned

- Check orientation of servos before installation

Future Work

- Compliant tail
- More complex feet
- Better physics models
- Energy-efficiency optimization
- Feedback control using sensors
- Approach rescue robotics application

Thank you! Questions?