

xJüs: Passive-Spine Hexapod

Senior Project with Design, Mid-Year Progress Report

Hayk Martirosyan, Greg Hughes, Thomas Owlett,
Brendan O’Leary, and Christopher Payne

Department of Mechanical and Aerospace Engineering, Princeton University
Department of Electrical Engineering, Princeton University

January 11, 2013

0 Abstract

Mid-year progress report of a research and design effort to create a segmented robotic hexapod with a passively compliant spine. The goal of the project is to improve performance of the widely researched RHex design over rough terrain and obstacles by incorporating a flexible backbone to allow for more natural motion, similar to the way that mammals and reptiles conform their bodies during locomotion. A comprehensive simulation was created and a prototype hexapod termed xJüs was designed but not yet manufactured. A two-legged test rig was built and achieves successful locomotion. The design of mechanical, electronic, and software systems for the prototype are discussed, as well as preliminary findings from simulation studies and the test rig.

1 Introduction

The field of robotics is increasingly utilizing biological mimicry in an effort to capture stable and efficient modes of locomotion seen in nature. In particular, hexapods like cockroaches can traverse a huge range of terrain with relatively little energy. Many hexapodal robots with this biological inspiration have been designed in recent years, but possibly the most successful adaptation is the RHex, spawned from a combination of various research universities, DARPA, and Boston Dynamics¹. By using rotary, compliant legs with a curved shape, the RHex is highly versatile yet at the same time simple enough to require only one active degree of freedom per leg. The RHex and related machines owe their dexterity to various passively actuated compliant degrees of freedom that provide adaptation to the environment at high frequencies and force ranges.²

This project aims to further model biological locomotion by incorporating a passive backbone into a RHex-like design. Having a fully rigid chassis lacks the flexibility and natural movement of reptiles and mammals, which employ many passive degrees of freedom in their bodies for robust navigation.

¹Boston Dynamics: RHex Rough-Terrain Robot <http://www.youtube.com/watch?v=ISznqY3kESI>

²U. Saranli and M. Buehler and D. E. Koditschek, RHex: A Simple and Highly Mobile Hexapod Robot, <http://kodlab.seas.upenn.edu/uploads/Kod/Saranli01.pdf>

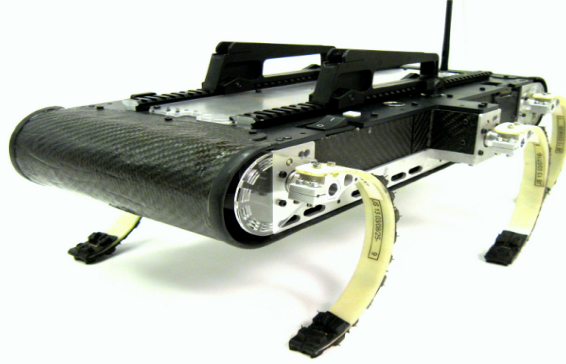


Figure 1: X-RHex, a RHex variant from the University of Pennsylvania

Observations from nature and the rigid RHex design suggest that a body with the properly designed passive articulation would lead to improved behavior. Our goal is to design and build a RHex-like hexapod that utilizes this passive backbone to achieve increased stability and energy retention over irregular terrain.

The rigid chassis of the conventional RHex will be divided into three segments, each supporting a single pair of legs and associated electronics. A unique joint has been designed for the purpose of providing two passively actuated degrees of freedom between each of the chassis elements. These segment pairs will be capable of pitching and rolling with respect to each other. The third degree of freedom, yaw, was eliminated due to the instability it introduces in forward locomotion. In total, four degrees of freedom are introduced from the passive spine, in addition to the six active degrees of freedom in the legs. Each of these passive degrees of freedom are independent and have adjustable stiffnesses, with the ability to lock them into place to act as a rigid chassis.

We are currently at the halfway point for the project and this paper discusses our progress over the past four months. A comprehensive simulation of our design has been built, which allows for open experimentation with the flexible-spine morphology both in terms of physical characteristics and control methods. The hexapod has been fully designed in software and significant analysis has been performed. The bulk of our mechanical and electrical components have been purchased and tested, and a test cart consisting of two legs and two back wheels has been successfully operated.

Moving forward, construction of the final design will commence on schedule. Experiments regarding the performance of our design in terms of stability and efficiency will be performed first on the simulation and later the physical robot. Control methods and high-level locomotion routines will be further developed and fine-tuned. The following sections describe our progress and preliminary findings in detail.

2 Simulation and Control Methods

2.1 Overview

To motivate our design and create a platform for rapid experimentation, a full and interactive 3D simulation was developed that combines rigid-body physics with custom dynamics and control algorithms. The simulation provides an environment in which the keyboard is used to specify

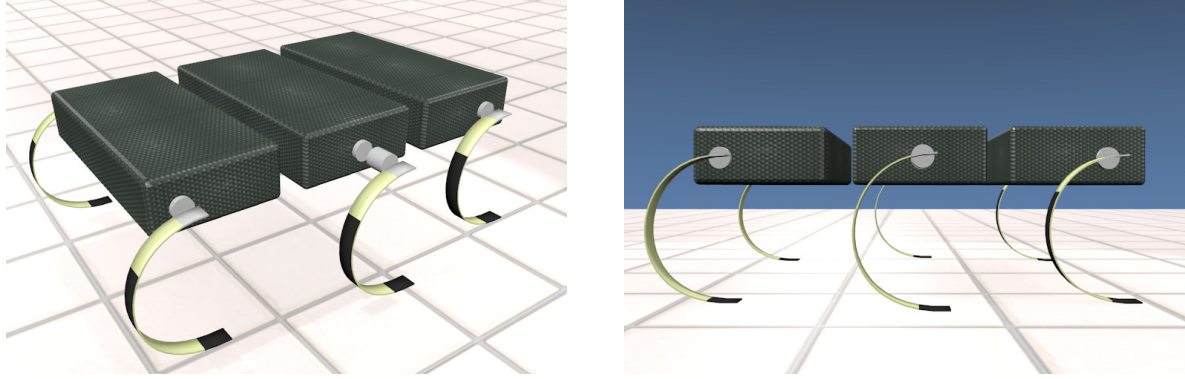


Figure 2: Rigid body model in the simulation environment

top-level commands such as standing, walking, and turning, which are then interpreted by control algorithms that apply PD feedback to actuate the simulated legs toward desired trajectories. The simulation includes a realistic motor model and consistent units such that various quantitative metrics regarding stability and efficiency can be measured. The result is a real-time interactive tool that can be used to test virtually any modification to the design, optimize parameters, and run experiments over simulated terrains - all with reasonable confidence in the realism of the simulated robot's behavior.

2.2 Simulation Environment

The simulation was created in Blender³, an open-source 3D graphics program with an embedded version of the Bullet physics engine. Bullet is a very popular engine and is the core physics library used in many well-known games, movies, and simulation environments⁴. It includes support for rigid and soft-body dynamics, 6DOF constraints, and accurate friction models. In this simulation, Blender is used to model the 3D bodies that make up the robot and the terrain it traverses, and Bullet is used to handle the rigid-body dynamics between these bodies. The rest of the dynamics, including the motor models and compliant spine joints, are programmed in Python and interact with Bullet by applying forces and torques to the rigid bodies. Blender is written entirely in Python⁵ and the simulation was made using a combination of the graphical interface and written Python code. OpenGL is used for real-time rendering at 120FPS with five physics substeps per frame, resulting in a working simulation timestep of 1.7 milliseconds and a logic timestep of 8.3 milliseconds. Arbitrarily small timesteps are achievable if recorded data instead of real-time interactivity is sufficient.

The robot model is a simplified version of the planned final design, consisting of rigid body segments constrained to each other with specified degrees of freedom. Each body has an associated mass, inertia tensor, and collision mesh. Three identical chassis segments each have box collision meshes, and are attached together to allow pitching and rolling but not yawing, as can be seen in Figure 3. Each leg system consists of one motor cylinder fixed to the chassis and one motor cylinder fixed to the leg. Each pair of motor cylinders are allowed relative rotation about their shaft axis, providing

³Blender, <http://blender.org>

⁴Bullet Physics Library, <http://bulletphysics.org>

⁵Blender Python API, http://www.blender.org/documentation/blender_python_api_2_65_5/

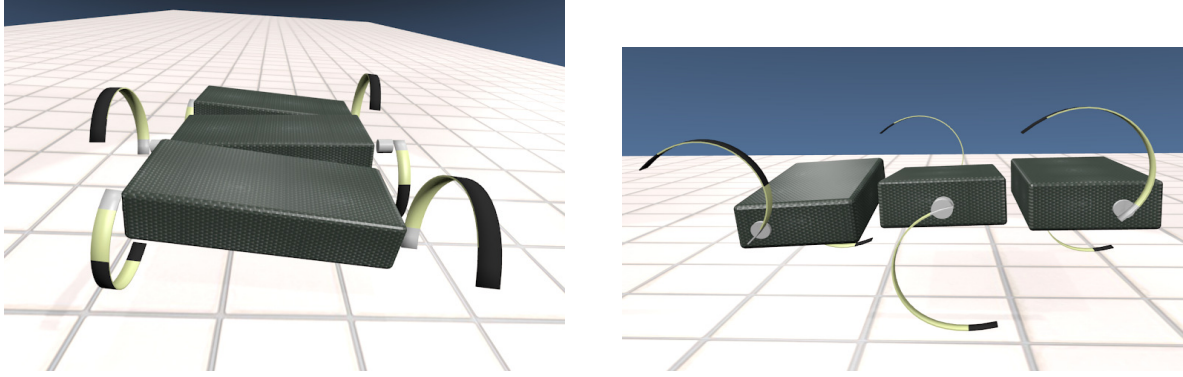


Figure 3: Simulated walking routine with compliant chassis segments

one degree of freedom per leg. The legs are modeled as thin curved strips, but their collision meshes are treated as convex due to significant performance gains.

Every polygon is assigned a material that has associated coefficients of friction and elasticity (restitution), as well as properties of its appearance within the simulation. Each texture in the simulation corresponds to a different material with realistic coefficients chosen to mimic what our final design would have. For every collision between two polygons, the physics engine takes into account the friction and elasticity coefficients of both colliding materials.

Efforts were made to ensure the quantitative rigor of this platform. A simple pendulum constructed in Blender yielded identical periods and trajectories to numerical solutions in *Mathematica*⁶ for varying masses and lengths. The results remained consistent when damping and forcing was applied using Python, in an identical manner to actuation applied in the robot simulation. All signs show that mass, length, and time units in the simulation are consistent with expectations and measurements can be considered fairly rigorous. Blender is a growing platform for robotics with developing support⁷, and multiple robotics frameworks have been created based on the combination of Blender and Bullet, the most prolific of which is MORSE⁸, a platform used by several research laboratories. Though this project did not use one of these established frameworks, the core methods remain identical.

2.3 Motor Model

Actuation of the six legs is simulated by a motor model and set of control algorithms primarily based on previous literature regarding the RHex platform. The motor model computes the output torque τ for a given leg based on the input voltage V_{pd} , the shaft angular velocity $\dot{\theta}$, and various motor-specific parameters. The model used is taken from previous work⁹, where the output torque τ is given by:

$$\tau = \eta N K_t \left(\frac{dV_s - K_s \dot{\theta}}{R_a + d^2 R_{amp}} \right)$$

⁶Wolfram Mathematica, <http://www.wolfram.com/mathematica/>

⁷Blender for Robotics, <http://wiki.blender.org/index.php/Community:Science/Robotics>

⁸MORSE, <http://www.openrobots.org/morse/doc/stable/morse.html>

⁹Justin Seipel, Pei-Chun Lin and Philip Holmes. "RHex-SLIP: A Model of the Robotic Hexapod RHex in the Sagittal Plane. 2006

where d is the dimensionless duty factor that relates the applied voltage V_{pd} to the source voltage V_s :

$$d = \max \left\{ -1, \min \left\{ +1, \frac{V_{pd}}{V_s} \right\} \right\}$$

The system parameters are given in Table 1. They are taken from the specifications of purchased components, except for K_t and K_s , which were chosen such that the model reflects the stall torque and no load speed of the motor.

Symbol	Constant	Value
N	gear ratio	$\frac{729}{25}$
η	net efficiency	0.60
V_s	source voltage	22.2V
R_a	motor resistance	$2.36 \frac{V}{A}$
R_{amp}	amplifier resistance	$0.5 \frac{V}{A}$
K_t	conversion factor	$0.016 \frac{Nm}{A}$
K_s	conversion factor	$0.75Vs$

Table 1: System Parameters

This model represents a linear torque-speed curve. Curves for various duty cycles are shown by Figure 4.

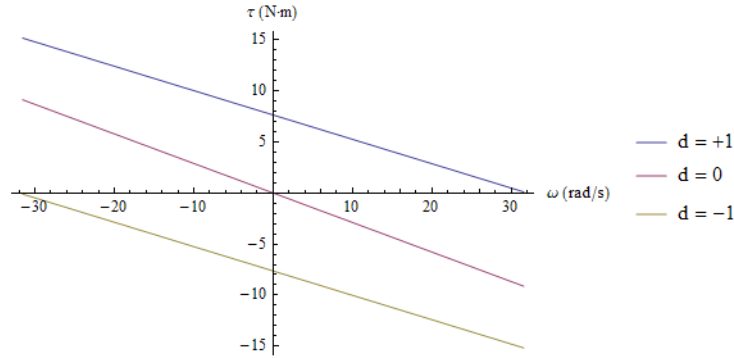


Figure 4: Duty Cycle Curves

2.4 Control System

The simulation uses a modular hierarchy of control methods that serve to translate high-level commands received from the keyboard into torques applied to the motor-cylinder rigid bodies. The very last step of actuation is to provide a feedback voltage V_{pd} to the motor model, which then outputs a torque τ and reaction torque $-\tau$ to the relevant pair of motor cylinders using the equations described in the previous section.

Calculating V_{pd} for each leg first requires generating a target trajectory for each leg. When the user invokes a keyboard command, a locomotion routine is activated which signifies a high-level motion such as walking, turning, or standing in place. For each leg, the high-level routine calls a trajectory generation function that calculates a desired angle function $\theta_d(t)$. The trajectory generation function is called with various parameters such as the desired period of revolution,

ground contact angle, turning magnitude, and relative timings of the legs, dependent on the goals of the chosen high-level routine. $\theta_d(t)$ outputs the target angle of a certain leg at any time t measured from start of the routine. The method of calculating $\theta_d(t)$ is discussed in the following section for clarity.

Once leg target angles are calculated for a given simulation frame, the high-level routine invokes a simple proportional-derivative feedback loop to calculate the motor input V_{pd} . The feedback loop takes the following error $\theta - \theta_d$ and uses specified gains k_p and k_d to calculate V_{pd} , which is fed to the motor model for low-level actuation.

Assuming a simple motor plant:

$$\tau = J\ddot{\theta}$$

A basic LTI state space for the shaft angle θ of a leg is given by:

$$\frac{d}{dt} \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{J} \end{pmatrix} \tau$$

Changing variables from the angle θ to the angle error $\theta - \theta_d$:

$$x = \begin{pmatrix} \theta - \theta_d \\ \dot{\theta} - \dot{\theta}_d \end{pmatrix}$$

$$\dot{x} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \ddot{\theta}_d - \begin{pmatrix} 0 \\ \frac{1}{J} \end{pmatrix} \tau$$

$\ddot{\theta}_d$ is a feedforward term that represents the acceleration of the generated trajectory, a known function. By substituting u as a generic input, we simplify:

$$u = \ddot{\theta}_d - \tau/J$$

$$\dot{x} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$

Applying state feedback for PD control, an expression for the desired torque output τ is derived:

$$u = -Kx = - \begin{pmatrix} k_p & k_d \end{pmatrix} x$$

$$\tau/J - \ddot{\theta}_d = \begin{pmatrix} k_p & k_d \end{pmatrix} x$$

$$\tau = J \left(\ddot{\theta}_d + \begin{pmatrix} k_p & k_d \end{pmatrix} \begin{pmatrix} \theta - \theta_d \\ \dot{\theta} - \dot{\theta}_d \end{pmatrix} \right)$$

$$\tau = J \left(\ddot{\theta}_d + k_p(\theta - \theta_d) + k_d(\dot{\theta} - \dot{\theta}_d) \right)$$

This is the control algorithm for τ . Combined with the motor model, this is a formal method of calculating V_{pd} for every frame, assuming the trajectory function is known. The following section discusses how $\theta_d(t)$ is calculated.

2.5 Trajectory Generation

The primary trajectory generation function expands on the fundamental tripod gait of the RHex morphology¹⁰. The main idea for this walking trajectory is that three legs are contacting the ground at all times, providing a stable base for locomotion. Given a revolution period T , each leg spends half the period sweeping a ground contact angle θ_g (generally 45° - 90°) slowly, and then comes full circle over the air angle θ_a with the second half of the revolution period. The velocity of this profile is a periodic piecewise constant function given by:

$$\dot{\theta}_d(t) = \begin{cases} \omega_g & 0 \leq \text{mod}(t, T) < \frac{T}{2} \\ \omega_a & \frac{T}{2} \leq \text{mod}(t, T) < T \end{cases}$$

where ω_g and ω_a are the ground and air velocities, derived from θ_g and T :

$$\begin{aligned} \theta_a &= 360^\circ - \theta_g \\ \omega_g &= 2\frac{\theta_g}{T} \\ \omega_a &= 2\frac{\theta_a}{T} \end{aligned}$$

To avoid the use of piecewise functions and to smooth the ground contact transition, a Fourier series of $\dot{\theta}_d(t)$ is used instead of the original. This has the benefit of being a continuous function that is easier to deal with both programatically and mathematically. The Fourier series of $\dot{\theta}_d(t)$ with M nonzero terms when fully simplified is given by:

$$\dot{\theta}_d(t) = \frac{2\pi}{T} + \sum_{m=1}^M \frac{8(\theta_g - \pi)}{(2m-1)\pi T} \sin\left(\frac{2\pi(2m-1)}{T}t\right)$$

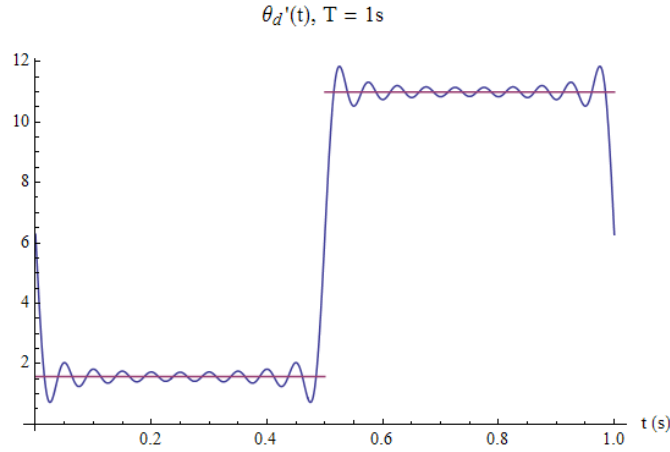


Figure 5: A plot of $\dot{\theta}_d(t)$ for $T = 1s$, $\theta_g = 45^\circ$, and $M = 10$, with both the piecewise and Fourier variants.

The Fourier approximation of $\dot{\theta}_d(t)$ is easily differentiable to yield an expression for $\theta_d(t)$:

$$\theta_d(t) = \left(\frac{2\pi}{T}\right)t + \sum_{m=1}^M \frac{4(\theta_g - \pi)}{(2m-1)^2\pi^2} \left(1 - \cos\left(\frac{2\pi(2m-1)}{T}t\right)\right)$$

Figure 6 shows $\theta_d(t)$ for the same trajectory as the plotted $\dot{\theta}_d(t)$, and we see the result is highly linear, but with a smooth transition between the ground and air regimes.

¹⁰U. Saranli, M. Buehler, and D.E. Koditschek. RHex: A simple and highly mobile hexapod robot. Int. J. Robotics Research, 20(7):616-631, 2001

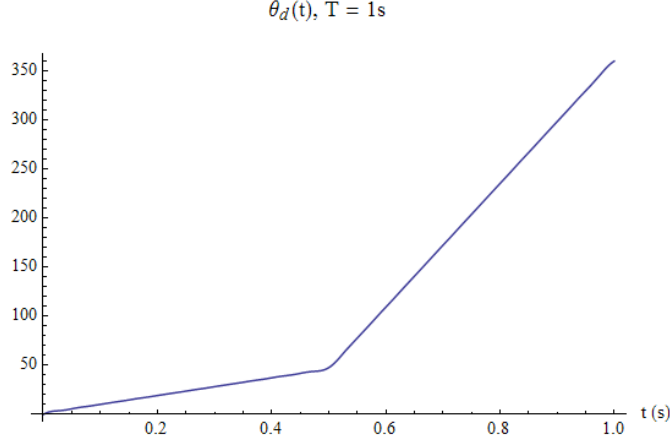


Figure 6: Fourier approximation of $\theta_d(t)$, the shaft angle, for one revolution

It is this function that is used by the trajectory generation function of the simulation, and will also be used for control the physical robot. By varying θ_g , T , and the starting time of each leg's trajectory, a wide range of locomotion routines are achieved. This method of trajectory generation has been successfully demonstrated for the simulation as well as for driving our actual motors.

2.6 Next Steps

At the current stage of development, the simulation is an accurate representation of the dynamics of our proposed flexible-spine hexapod. Moving forward, it will be used as a testbed for optimizing parameters and general experimentation to guide our design decisions as plans for construction of the physical robot prototype are finalized. Formal simulated experiments comparing the compliant chassis joints to a fully rigid chassis have not yet been conducted, but qualitative results show some improved behavior over rough terrain. Simulated experiments will be conducted using various metrics of stability and efficiency to further motivate our novel step and build reasonable expectations for our final design. For example, power usage over stochastically generated terrain can be measured for various joint stiffnesses. Once construction is complete, analogous experiments will be attempted with the final design and results will be compared for further evaluation of the simulation platform.

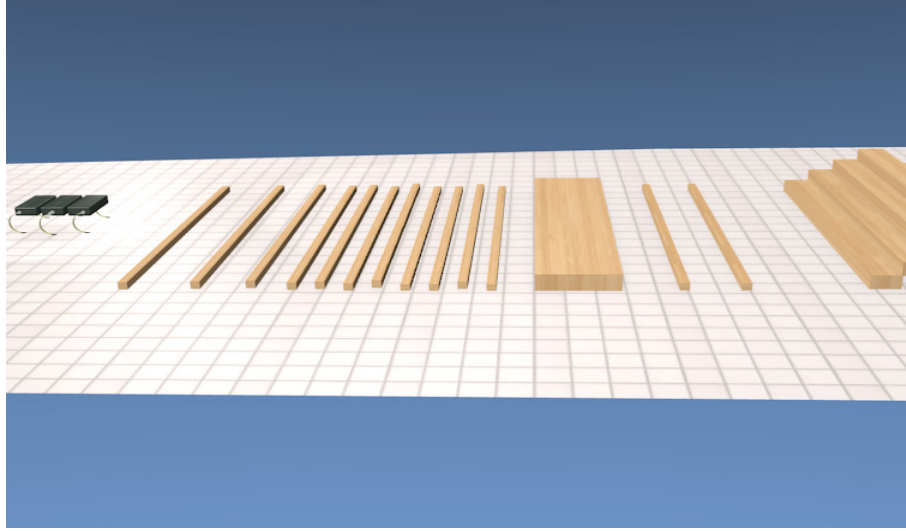


Figure 7: Example of a simulated obstacle course

3 Design

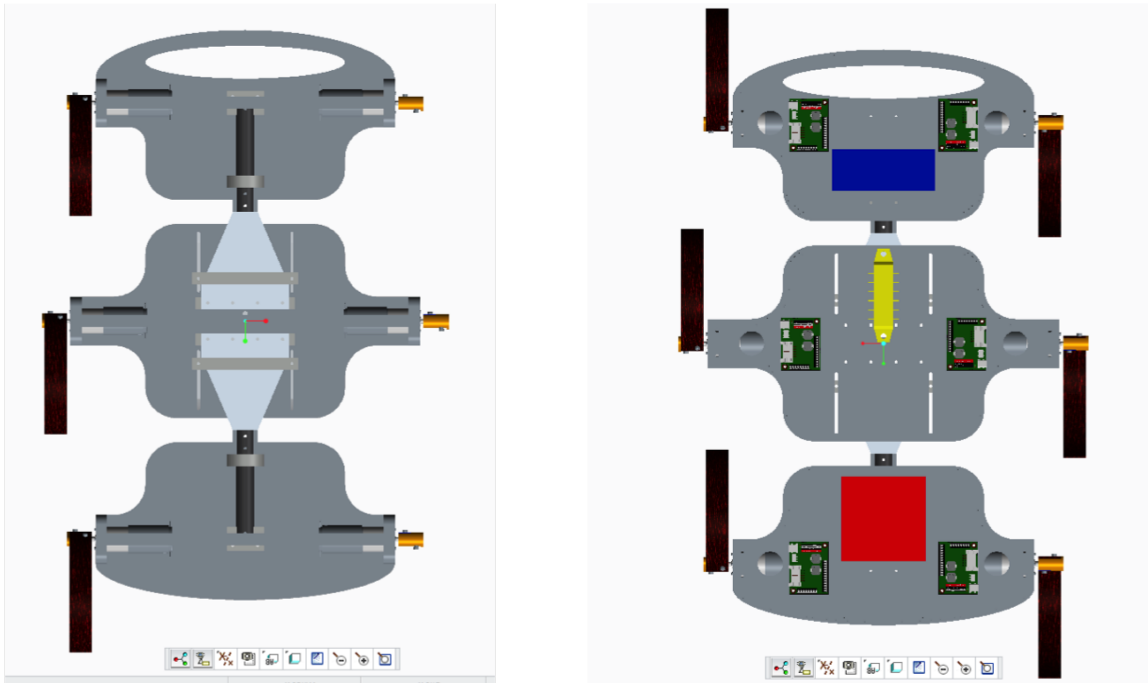


Figure 8: Planar views of our design. Left: Bottom View. Right: Top View.

3.1 Overview

Our objective is to design and build a hexapod that expands upon previous RHex variants¹¹ by incorporating a passive backbone that increases versatility and stability. Constructing this robot is a considerable undertaking given our time constraints, so a large focus is put on simplicity of design and modularity. The core of our research effort is dedicated to the design and implementation of a passive backbone allowing two degrees of freedom between the chassis segments. For this reason, we have adapted a similar morphology for the body and legs as that used in the Research RHex¹² and Edubot¹³. Many of our mechanical and electrical systems are modeled after those of the Edubot, but on a larger scale, similar to that of the Research RHex.

The primary focus of mechanical design was placed on the joints between chassis segments. After considering the use of torsion springs for the compliant degrees of freedom in each joint, we opted for an alternative mechanism using tapered beams of spring steel. An adjustable fulcrum along the median of the tapered beam controls the pitch and the placement of a clamp along the length of a rectangular beam adjusts the amount of roll. With the design of the joint complete, our plan is to build a model of the joint to confirm its motion capabilities before we manufacture the rest of the hexapod. The dimensions of body components are currently similar to the Research RHex, but further simulation studies will be done to optimize parameters. Additionally, we opted to use spring steel for the compliant legs rather than composites to simplify manufacturing.

Our electrical system was designed with simplicity in mind, taking advantage of off-the-shelf components whenever possible. Each motor stack consists of a 22-Watt DC motor with a 33:1 planetary gearbox, an encoder, and a Maxon EPOS2 digital position controller. The motor controllers are connected in a CAN network and receive commands from an onboard Linux CPU. Considerable time has been and will continue to be attributed to interfacing with the motor controllers to achieve desired motor performance. Primary power is supplied by a 2200mAh 22.2v LiPo battery, and the CPU will have an auxiliary battery.

Based off of the electrical components already purchased and the current dimensioning of the mechanical components in our design, the robot's estimated weight is 4467g or 9.85lbs. This is significantly lighter than the Research RHex, totaling 8200g. The higher power to weight ratio of our robot will enable more gait patterns and locomotion strategies.

Our initial budget estimate for this project was \$5725 for all mechanical and electrical parts. So far a total of \$4574.88 has been spent, \$269.45 of which was added to our initial estimate, such as a battery charger and supplies for the manufacture of the test cart.

¹¹RHex variants, <http://www.rhex.web.tr/>

¹²Research RHex, <http://kodlab.seas.upenn.edu/RHex/ResearchRHex>

¹³EduBot, <http://kodlab.seas.upenn.edu/RHex/EduBot>

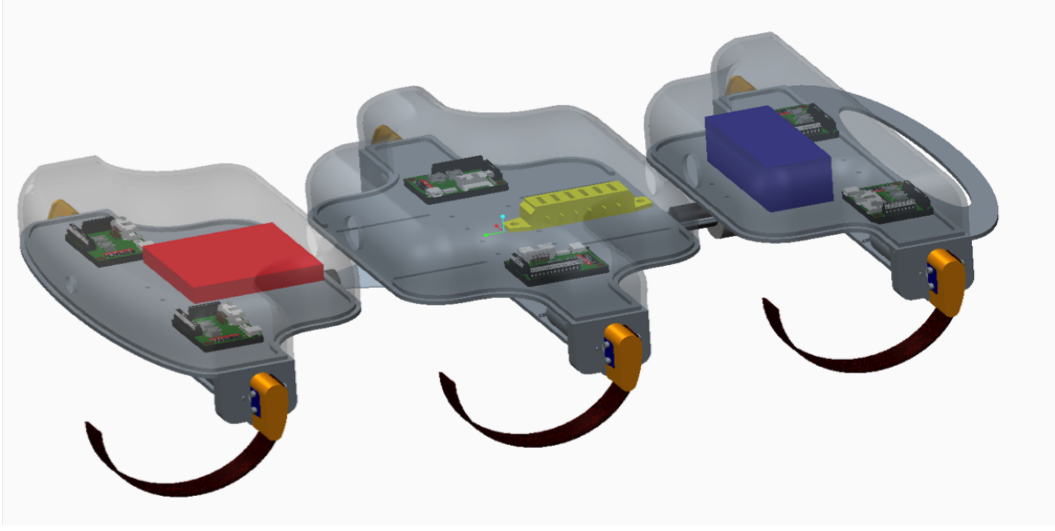


Figure 9: CAD model, 3D view

3.2 Actuation and Power System

Given the estimated parameters of our design, we can estimate the torque needed for locomotion via the tripod gait:

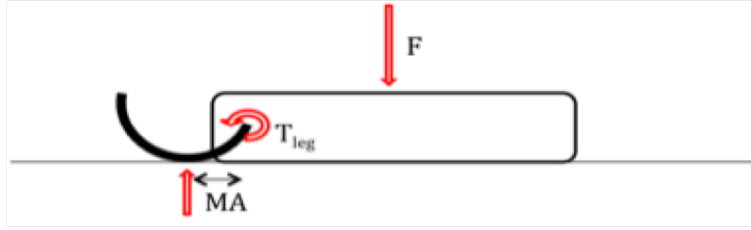


Figure 10: Free body diagram for motor torque calculation

$$\text{Moment Arm } MA = 1.5''$$

$$\text{Force } F = M = 10\text{ lbf}$$

$$\text{Torque applied by motors } T_{tot} = F * MA = 15\text{ lbf}\cdot\text{in}$$

$$\text{Torque per leg } T_{leg} = T_{tot}/3 = 5\text{ lbf}\cdot\text{in} = \mathbf{0.565\text{ Nm}}$$

This is the maximum moment arm applicable to the tripod gait, in the case of effectively lifting the chassis from the ground with only three legs. Typical locomotion will require less torque, and standing up from a rest position will utilize all six legs.

Using the upper bound of the torque needed per leg of 0.565 Nm, this value can be scaled to an associated current draw upper bound. Using the motor torque conversion factor of 25.8 mNm/A and keeping in mind the gearbox ratio of 729/25 we arrive at a maximum current load of 0.75A, which is approximately 25% below the maximum continuous current rating of 1.08A per motor.

$$565\text{ mNm} * (25.8\text{ mNm/A})^{-1} * (729/25)^{-1} = 0.75\text{ A}$$

We selected a Blue LiPo battery with a capacity of 2.2Ah to service the motors, providing a lower bound of 29 minutes of runtime.

$$(0.75A/\text{leg}) * (6 \text{ legs}) * (2.2Ah)^{-1} = 29 \text{ min}$$

In reality runtime will be much longer as this calculation assumes a maximum torque load. As we continue to develop the power system and software simulations we will have a better picture of the true torque profile and its accompanying power use, a relationship dictated by both the motor torque conversion power and inefficiencies in the motor controller, motor and accompanying wires. The chosen battery carries a voltage of 22.2V as a result of 6 internal 3.7V LiPo cells, a voltage that is related to the motor characteristics through the following relation:

$$V_s = 1(V) + \frac{U_N}{n_0} \cdot \left(\frac{\Delta n}{\Delta M} \cdot M_B + n_B \right) \cdot \frac{1}{0.9}$$

Known values:

- Operating torque M_B [mNm]
- Operating speed n_B [min^{-1}]
- Nominal motor voltage U_N [Volt]
- Motor no-load speed at U_N, n_0 [min^{-1}]
- Speed/torque gradient of the motor $\Delta n / \Delta M$ [$\text{min}^{-1} \text{mNm}^{-1}$]

These voltage and run-time characteristics matched our electrical needs and most importantly provided the power in a high density form to minimize the weight of the hexapod. Table 2 compares the vital characteristics of our design's battery to that of the EduBot. For added protection, all individual power lines leading from the battery bus to the motors will be fused at 2A. A bypass capacitor might be tied to the battery output to dampen high frequency voltage fluctuations that affect the controller operation. The BeagleBoard-XM will be operated on a separate 5VDC power source to isolate the digital electronics from the voltage fluctuations caused by the inductive nature of the motors. Currently we are powering the board using a desktop mounted 5VDC power source but will be moving to a mobile battery pack after a suitable one has been selected.

Attribute	xJüs	EduBot
Battery Type	LiPo	Lipo
Bus Voltage [V]	22.2	14.8
Battery Capacity [Wh]	48.8	20-30
Battery Quantity	1	1-2
Dimensions [mm]	35 x 104 x 43	66 x 74 x 34-50
Battery Mass [g]	401	122-160
Robot Mass Dedicated to Battery [%]	8.9	3.6-8.9
Battery Energy Density [Wh/kg]	122	158-187
Robot Energy Density [Wh/kg]	10.8	5.8-16.7

Table 2: Comparison of battery properties

We can also estimate the power draw of our motors, approximated with the target motor torque operating at 3rps:

$$P_{leg} = T * w = 0.5Nm * 3(2\pi) = 9.42W \text{ (per leg)}$$

$$P_{tot} = 6(9.42W) = 56.5W$$

This is a rough estimate of the maximum continuous power output needed for the tripod gait. This value must be divided by the efficiencies of the motor, gearbox, and controller to calculate the power draw needed from the battery.

3.3 Structural Components

The chassis segments consist of three 0.25" aluminum plates connected by two compliant joints. Each section has its own carbon fiber housing above it to protect the electronics from damage and dirt. The aluminum plates have cutout recesses to reduce the robot's weight without compromising the structural integrity. Because the joint allows for roll and pitch between the plates, there is much less torsional stress between the body segments as the robot switches from one set of legs to the other. To minimize chassis weight, sensitivity studies will be conducted to optimize the removal of material from the cutout sections.

The legs of the robot are constructed using 1074/1075 spring steel. This steel was chosen for its ability to deform elastically and recover some spent mechanical energy, as well as its relatively high yield strength (50,000 psi). The formula used to determine the dynamic force for the finite element analysis is:

$$\text{Dynamic Load} = \frac{\text{Robot Mass}}{\# \text{ of Contact Legs}} * \text{Dynamic Load Factor} * g$$

A dynamic load factor of 2 was selected for the legs for consistency with the flexible joint deformation calculations. Using the estimated total robot mass of 4.467 Kg, the dynamic load for each leg is 29.2 N. Figure 11 shows the simulated displacement and stress using this calculated dynamic load and a spring steel thickness of 0.072 inches. The simulation shows a maximum displacement of 0.30 inches and a maximum stress of 22,888 psi, which is well below the yield stress of 50,000 psi.

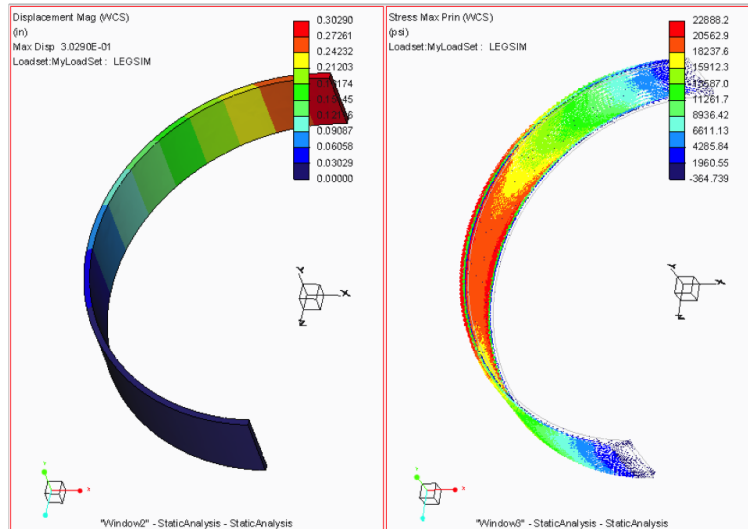


Figure 11: Leg stress and displacement under load

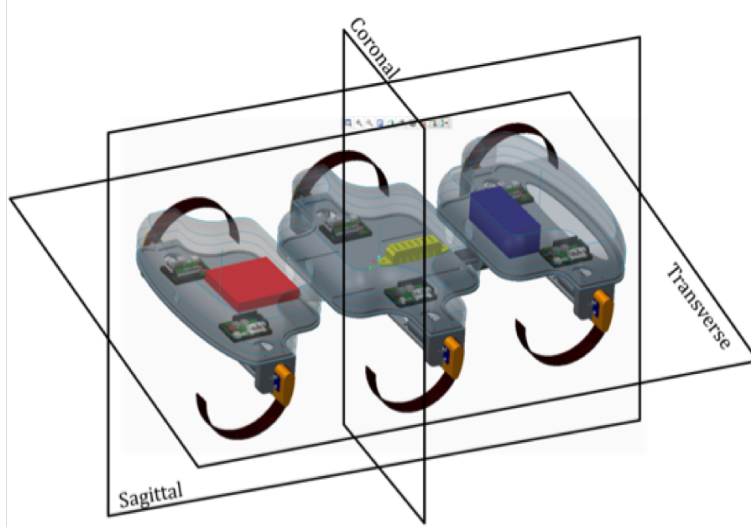


Figure 12: Isometric view of the segmented body and labeled planes of the body-axis reference frame

3.4 Spinal Joints

The addition of structural flexibility to the hexapodal design potentially increases stability and efficiency traversing obstacles. It is also intended to expand the range of terrain that the robot is capable of traversing. The segmented design of three chassis elements connected by two passively actuated joints allows this flexibility to be achieved while retaining rigid platforms for the mounting of required hardware.

The joint is responsible for the six degrees of freedom between each segment pair. Inspired by biological vertebrae, the three rotational degrees of freedom are emphasized, leaving the three translational ones rigid. Compression and restoration of the semicircular legs already conserve some of the energy used in each step. Rotation in the coronal plane is intended to increase this conservation by allowing the segments to roll several degrees with respect to each other within the period of the tripod gait. The second rotational degree of freedom imitates the arching ability of an animal's spine by pitching the segments. This flexible curvature of the robot's chassis in the sagittal plane allows the third set of legs to be in contact with the ground when the slope beneath the robot varies dramatically along its length. Such an instance occurs while mounting a stair-like obstacle: the front segment pitches upwards allowing the legs of the middle and rear segments to remain in contact with the flat ground until they too encounter the first step. The final degree of freedom in this body-axis frame of reference is rotation between segments in the transverse plane. This freedom allows the robot to turn aside from obstacles and, by affecting the direction of forward progress, naturally leads to a path of steepest descent. Due to our intent of accurately controlling the direction of the robot's locomotion, yawing motion has been deemed unfavorable and will be rigidly restricted.

These locomotive goals are the purpose the joint must serve and, together with certain other restrictions, define its design. Four degrees of freedom must be rigidly constrained leaving the roll and pitch motions to be passively actuated such that the ideal degree of flexibility is attained. As this is a research project meant to test the effectiveness of such flexibility, the stiffness of the two allowed motions must be tunable, either continuously or in discreet intervals, and capable of a fully

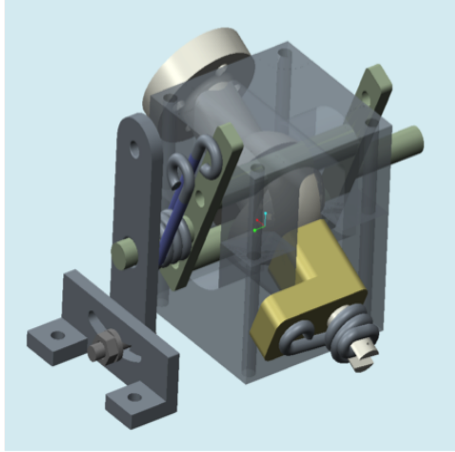


Figure 13: First joint design: uses torsion springs to passively activate the roll and pitch motions of the ball-axle

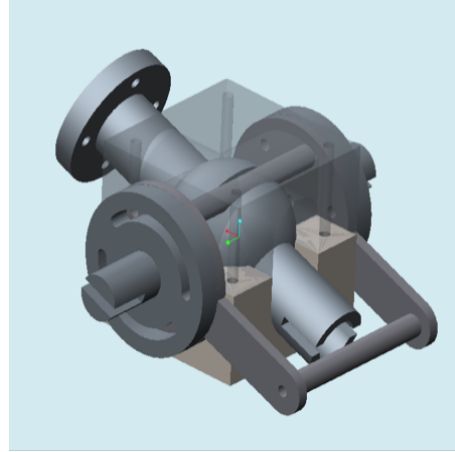


Figure 14: Second joint design: uses cantilever beams fit through the two open slots to apply restoring torques to these motions. Each had an adjustable equilibrium pitch angle

rigid condition for comparison. Weight is a significant limiting factor because increasingly powerful motors, required to compensate for a heavier robot, are increasingly expensive. Designers of the very similar EduBot have also emphasized a positive correlation between weight and difficulty in control. The size of the joint system is restricted due to the necessity of fitting it on the robot's chassis segments while retaining enough surface space for hardware. The problem may be reduced to these requirements: restriction of four degrees of motion and a compact, adjustable method of passive actuation for the remaining two degrees.

The original design was based around a ball joint with a protruding axle (see Figure 13). With the joint's housing mounted to the middle segment and the end of the axle attached to either end segment, the axle's rotations allow the segment pairs to pitch and roll with respect to each other. Furthermore, the two degrees of freedom within the joint can be mechanically and independently controlled: pitch rotation via a fork slotted through the ball, and roll rotation at the back end of the axle. Either of the motions can be made completely rigid or passively actuated by a single torsion spring without affecting the other. The plan was to simply replace the torsion springs to acquire different degrees of stiffness. However, difficulty arose in the selection of torsion springs, because the estimated dynamic loads on the segments required torques of up to 60 lbf-in at the joint. Achieving this torque, after less than 20 degrees of rotation, requires a level of stiffness such that the spring size becomes cumbersome. Larger springs require a scaled up joint and this became a rather dense design, involving relatively large volumes of plastic and metal.

The next variation sought to preserve the ball joint design while replacing the too large torsion springs with more compact and continuously adjustable thin, cantilever beam springs which would have a moment applied around the free end. The desired maximum angles of pitch and roll created stress in the beam such that its length had to be too long to fit on top of the chassis with the rest of the joint system.

At this point in the design process it was decided that the joint was becoming too complicated and a new track was taken, combining the two previously summarized requirements into a single solution.

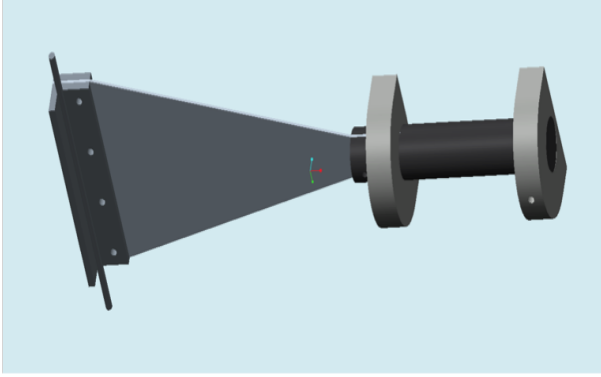


Figure 15: Third joint design: relied upon a welded pin and a polyurethane torsion rod

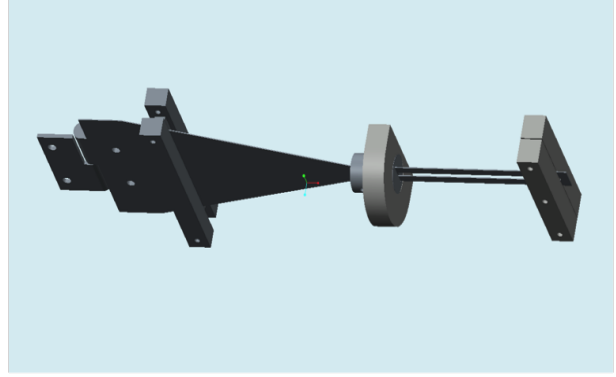


Figure 16: Current joint design: replaced the pin with a hinge, torsion rod with a narrow extension of the same plate, and includes the translatable fulcrum for adjusting stiffness in pitch

Rather than create a rigid joint which allows motion in only the two desired degrees of rotation and then add a system of passive actuation to it, the joint itself could be made from a flexible material. The cantilever thin beam spring from the prior version can be increased in length and width to handle the stress, providing pitch flexibility as well as the structural connection between segments. Roll flexibility requires a much narrower extension of the plate which is otherwise too rigid in that mode of torsion (see Figure 15). By relocating the beam spring to the underside of the chassis, space is preserved on top for the hardware.

The bending dynamics of a cantilever beam result in the maximum stress occurring at the clamped end; the stress then decreases linearly to zero at the loaded end. By tapering the width of the beam, the stress can be made constant throughout the spring and a greater deflection and angle of deflection acquired at the load end without raising the maximum stress (see Figure 17). However, this creates a problem with adjusting the stiffness. As the location of the clamp is moved towards the load end in order to stiffen the joint, the maximum stress is occurring at an increasingly small cross section and so increases. In order to prevent stress from increasing with stiffness, the cantilever beam was replaced by an overhanging beam pinned at the base and simply supported from an adjustable fulcrum. By moving the fulcrum towards the load end, the spring can be stiffened while still utilizing the length of the beam behind the fulcrum, unlike with the clamped cantilever. With this pinned, over-hanging beam, the maximum stress occurs at the fulcrum and decreases as the fulcrum moves towards the loaded end (see Figure 18). This serves to counteract the increasing stress caused by the tapered width and shrinking cross sectional area. The over-hanging beam maximizes the ratio of end deflection to stress and allows for an adjustable stiffness without significantly changing the maximum stress.

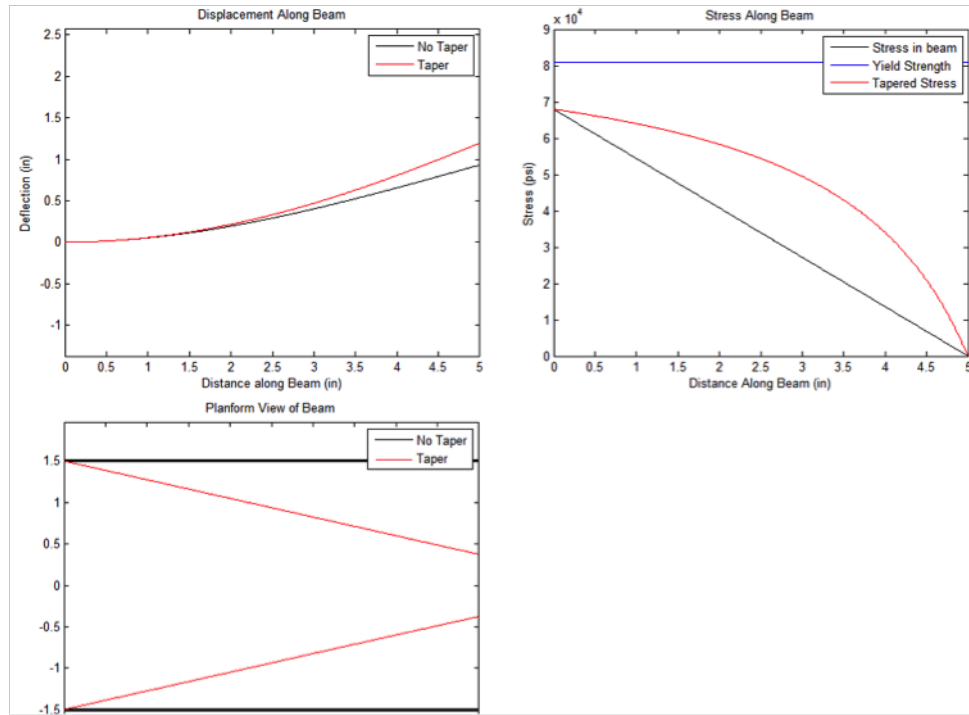


Figure 17: Plots of deflection, stress, and planform shape of the cantilever beam comparing a constant width (black) with tapered (red).

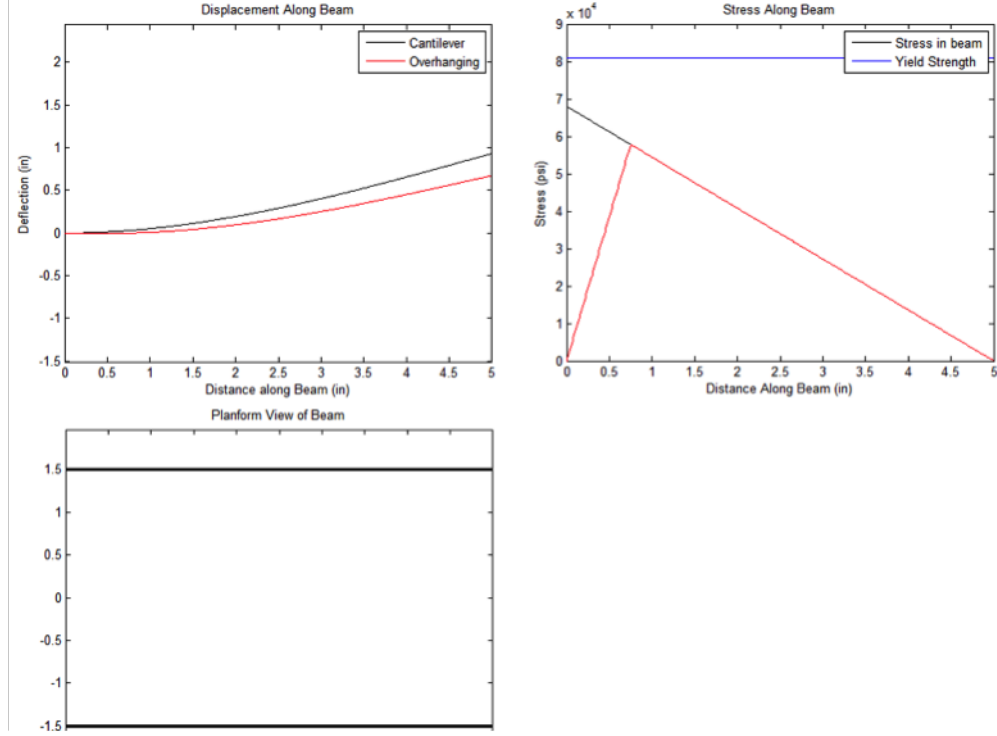


Figure 18: Plots of deflection, stress, and planform shape comparing cantilever (black) to overhanging (red) type beam. The overhanging beam sacrifices some deflection for less stress and adjustable stiffness. The fulcrum of the overhanging beam is located 0.75in from the pinned end in this instance.

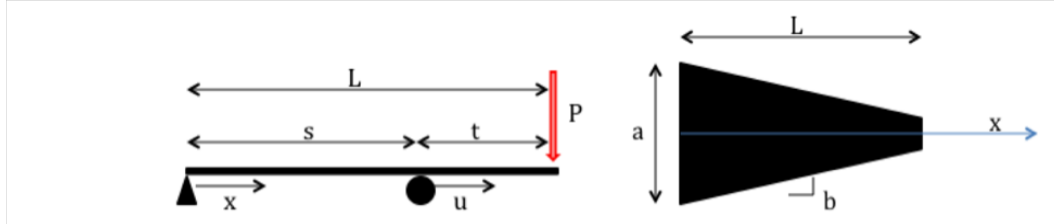


Figure 19: Left: Supports for triangular section of joint. Right: planform view

$$\sigma_{max} = \frac{M_{max}c}{I} = \frac{P \cdot t \cdot \frac{h}{2}}{w \cdot \frac{h^3}{12}} = \frac{6P}{h^2} \frac{L-s}{w}$$

$$w = a(1 - bs) = \frac{a}{L}(L - bLs)$$

$$\sigma_{max} = \frac{6P}{h^2} \frac{L-s}{\frac{a}{L}(L - bLs)} = \frac{6PL}{ah^2} \frac{L-s}{L - bLs}$$

$$\text{As } b \rightarrow \frac{1}{L} \sigma_{max} = \frac{6PL}{ah^2}, \text{ constant}$$

$$\text{deflection at } x = \frac{1}{EI} \left(\frac{Pt}{6s} x^3 - \frac{Pts}{6} x \right)$$

$$\text{deflection at } u = \frac{1}{EI} \left(\frac{Pt}{2} u^2 - \frac{P}{6} u^3 + \frac{Pts}{3} u \right)$$

3.5 Electronic System

In order to provide actuation in a stable and precise manner we chose a pre-assembled EPOS PID control system provided by Maxon Motors¹⁴ combined with a BeagleBoard-XM single-board computer¹⁵. On a modular scale, the components are arranged in the manner shown in Figure 20 to implement closed loop control.

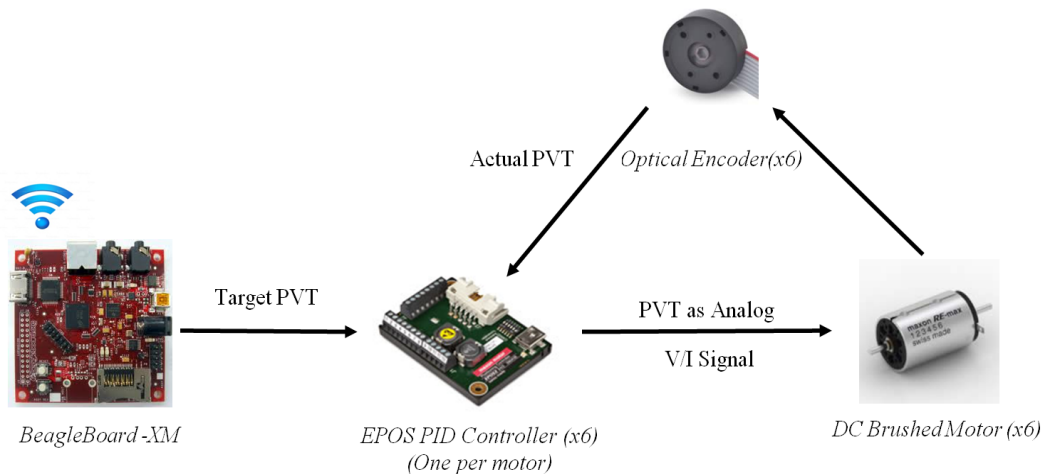


Figure 20: Control Loop

On this scale, the user accesses the BeagleBoard-XM via WiFi to initiate a particular high-level locomotion routine. This routine ranges from simple in nature such as move forward a half meter to more complex functions such as move towards most prominent lightsource. The BeagleBoard-XM, running Python/C++ scripts on Linux, then calls the appropriate EPOS microcontroller API methods in order to execute the associated Position, Velocity, Time (PVT) trajectory as generated with the control algorithms described in the simulation section.

In order to efficiently communicate between the BeagleBoard-XM and each of the EPOS controller units, a CAN network is utilized. We assign each controller in the network a particular node ID by aligning binary DIP switches on the controller. All the nodes are then linked together via a serial CAN COMM consisting of a CAN High and CAN Low line, with all nodes commonly grounded. We wound the three lines together in order to provide equal capacitance throughout the network to minimize noise interference. Finally, a master node links the CAN network via USB to the controlling BeagleBoard-XM. When a packet is sent using the CAN network standard, its contents are read by only the node that corresponds to the node ID found in the packets header.

It should be noted that at this time we are only using two nodes that are connected to our laptop via the USB protocol. We are currently using the EPOS Controller PC GUI, rather than the Linux EPOS Controller API, to control the two motors on our test rig. The GUI provides an intuitive interface to use as we learn how to control the motors at the expense of having limited functionality especially when attempting to synchronize node behavior. Once we have finished this testing phase, we will develop scripts that call the EPOS Controller API in order to control all six nodes in a precise and synchronous manner.

¹⁴Maxon Motors, <http://www.maxonmotorusa.com/>

¹⁵Beagleboard XM, <http://beagleboard.org/hardware-xm>

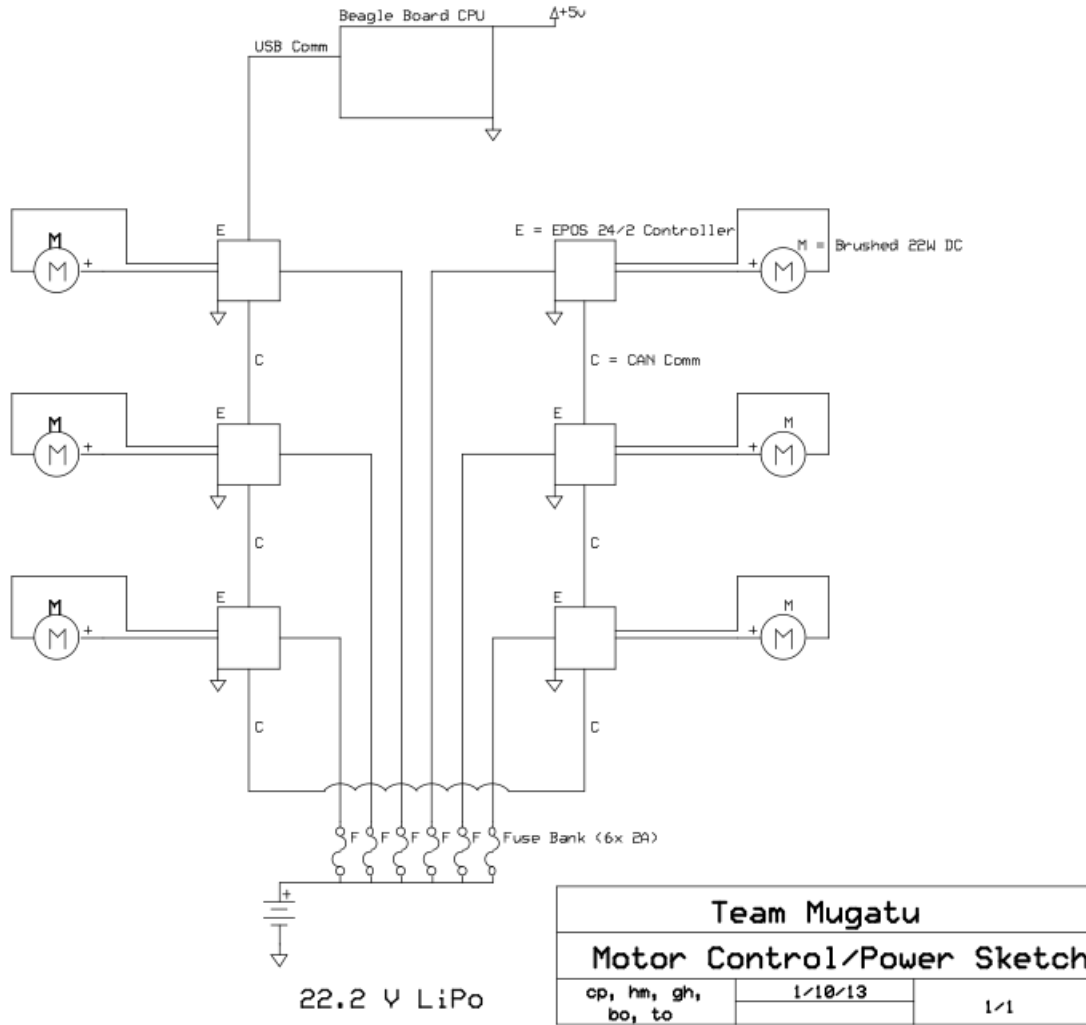


Figure 21: Circuit diagram of our design, demonstrating the communications network and power system

4 Test Rig

The test cart (Figure 22) was constructed to demonstrate the operation of our mechatronic systems. It has two legs instead of our final six to simplify construction, wiring, and programming. The back of the test cart is supported by two wheels, giving the rig additional stability and allowing it to stand on one of the two front legs.

Because the test cart is a simplified version of the final robot, the specifications and methods of construction are as close to those of our final robot as possible. The platform of the cart is made of a sheet of PVC plastic, which was chosen because of its low cost and ease of machining. The front legs are bent into 7" diameter semicircles, which is the preliminary size of the legs that will be used on the final robot. The legs themselves are made of one inch wide strips of 0.072" thickness spring steel, which was chosen for the six legged robot based on finite element analysis conducted in Creo (see Figure 11). The two-piece joint that affixes the steel leg to the motor was machined

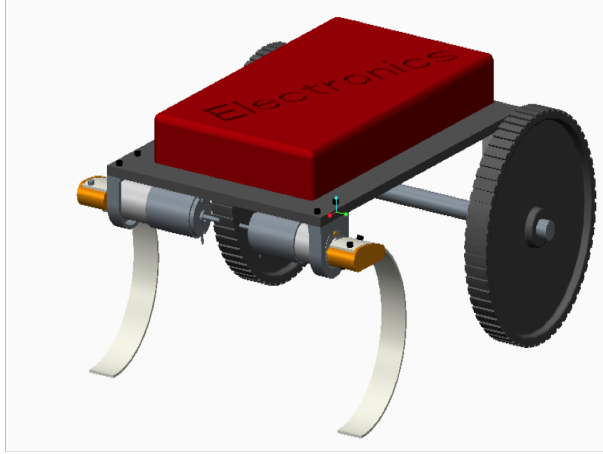


Figure 22: CAD model of the test cart

out of aluminum to the specifications of the final robot. The aluminum motor mounts, which were machined using a mill for the test cart, will be machined using CNC for the final robot.

Currently, the test cart is capable of successful basic locomotion. We are able to generate trajectories on a laptop and send commands to the network of motor controllers, which then power the legs with various gaits. The next step is to write software for our onboard CPU such that the cart does not need to be tethered to a laptop. Overall, power and torque observations made about the test cart match up well both with our initial calculations and results from simulation studies. Progress on the project has suffered no major setbacks, and we proceed on schedule toward the construction of our main prototype.

5 Acknowledgements

This work is funded by the Princeton University School of Engineering and Applied Science, the Department of Mechanical and Aerospace Engineering, and the Department of Electrical Engineering.

The authors would like to thank Professors Daniel Nosenchuck, Phillip Holmes, and Andrew Houck for acting as advisers, and Dr. Haldun Komsuoglu for providing invaluable support.