# Design, Implementation, and Testing of a Force-Sensing Quadrupedal Laminate Robot

Benjamin Shuch, Eric Rogers, Daniel M. Aukes
Fulton Schools of Engineering
Arizona State University, Mesa AZ

*Abstract:* In this article we present a low-cost force-sensing quadrupedal laminate robot platform. The robot has two degrees of freedom on each of four independent legs, allowing for a variety of motion trajectories to be created at each leg, thus creating a rich control space to explore on a relatively low-cost robot. This platform allows a user to research complex motion and gait analysis control questions, and use different concepts in computer science and control theory methods to permit it to walk.

The motion trajectory of each leg has been modeled in Python. Critical design considerations are: the complexity of the laminate design, the rigidity of the materials of which the laminate is constructed, the accuracy of the transmission to control each leg, and the design of the force sensing legs.

## Introduction

### A. *Previous Research*

#### a. Robot Manufacturing

Laminate manufacturing has enabled the rapid development and creation of inexpensive complex robots. These robots use rigid links attached to each other via soft, flexure-based hinges. These hinges are the primary method of transmitting motion through a mechanism. Using design tools written in the Python programming language and popupCAD, a design tool for computing laminate manufacturing geometries [3], we can quickly design, create, and manufacture robots with these hinges. They are made from low-cost materials [] such as paper, film, and sheet adhesives, and are actuated with RC hobby servos.

#### b. Existing Robots

Many other examples of robots have been made possible because of laminate manufacturing [2] [9] [10]. Some of the most notable are DASH [5] and HAMR [4]. DASH has six legs and uses two motors. HAMR uses piezoelectric actuators on each of the legs to control its motion. HAMR is the closest robot to this platform in that it has 2 actuators per leg, but it does not have force sensing embedded in each of its feet, which means that it is much more reliant on external sensing strategies such as high-speed motion capture.

Figure (1): Image of discussed robot

Other larger more complex robots already have similar quadrupedal force sensing platforms, such as BigDog [11] and StarlETH [7]. Big dog has a variety of on board systems tha power, control and sense throughout the robot. It is able to move over many different types of terrain, and is an incredibly high functioning robot, however, the cost to build the robot was around 32 million dollars[1]. StarlETH has controllable system torque and is useful for highly dynamic maneuvers. These robots are effective for gait and motion analysis, but they are expensive to manufacture and operate, and less accessible to research labs.

Although robots such as HAMR and DASH are quicker and smaller than this robot, their mechanical complexity creates a small control space. These robots move because the mechanisms governing their gait and locomotion were tuned to work with a minimum number of motors by expert designers. This has resulted in a simplification of the controls space which does not permit a thorough exploration of the many possible gaits and locomotory strategies which are available to animals or more complex robots. This robot provides a rich control space by providing a

[1]From an article on IEEE.org. https://spectrum.ieee.org/automaton/robotics/military-robots/boston-dynamics-ls3-robot-mule

variety of different leg trajectories that a user can take advantage of, and force sensing on each of the robots feet. This robot can be programmed to move using traditional control and optimization methods, and others like machine learning. [12] presented an example of a laminate robot called C-Turtle [8] that learned how to move better using different machine learning algorithms.

### *B. Motivation*

The objective of this paper is to illustrate a laminate quadrupedal robot which demonstrates the usefulness of being able to dynamically adapt locomotory patterns to changing environmental stimuli. This can be sensed through the distribution of weight on the feet. Using two actuators per leg will create a rich control space, permitting the study of gaits, balance, high-level control, and tasks-based decision making. We seek to enable this on a low-cost, open platform so that robot designs and experimental results can be distributed, shared, and compared easily across labs. This robot provides more functionality than a “toy” -- its control problems are only partially solved through careful mechanical design and analysis. There is little new information that can be learned from a controls perspective when all the actuators are programmed via the mechanical design.. Finally, by providing the controller with rich state information, such as the loads distributed on each foot, the task of balancing in an unstructured terrain can be made closed-loop.

### *C. Design Methodology*

By utilizing the laminate manufacturing process, PopupCAD, 3D CAD software, Python, a laser cutter, and thin sheets of material, we can rapidly prototype robots. In Solidworks, we can create the topology for our designs. These designs can be exported to PopupCAD, and turned into the DXFs we need to laser cut. The DXFs are cut out, and the robot is put together. The servos and 3D printed parts are placed in their designated locations. The process for completely building an iteration of each robot takes less then a day to complete.

In Python we can model the leg trajectory of the four-bar leg. The goal is the maximize the motion output of the servos to allow for the maximum output torque. By being able to change the size of the linkages on the leg and the angles between them by changing one line of code, we can quickly find the best solution for our current design. It also allows us to explore basic control methods using sine wave functions to control the trajectory.

Both methods are valuable for the robots design process. By physically iterating the robot, we can quickly validate physical design issues that do not appear in the Python simulation. However, the Python simulation allows us to assess the range of motion that the robot is capable of, something that would be difficult to do if the physical prototype does not work in the first place. The combined information from these two methods allows us to make justified changes to the robot. These could include, improving the motion of our robot, change its size and weight, and choosing the most applicable materials.

We also are adding force sensing capabilities to the robot. This will create a more controllable platform, and allow users to control the level of each foot based on more complex programming algorithms. The sensor is a flex sensor, and the foot itself is made out of fiberglass to allow for deflection. The sensor is placed onto the foot, and four feet are placed onto the robot.

| | |
|---|---|
| Total Cost ($) | 90 |
| Total Weight (g) | 396 |
| Size X, Y, Z (cm) | 16x26x16 |
| Total Actuators (#) | 8 |
| Total Sensors (#) | 4 |

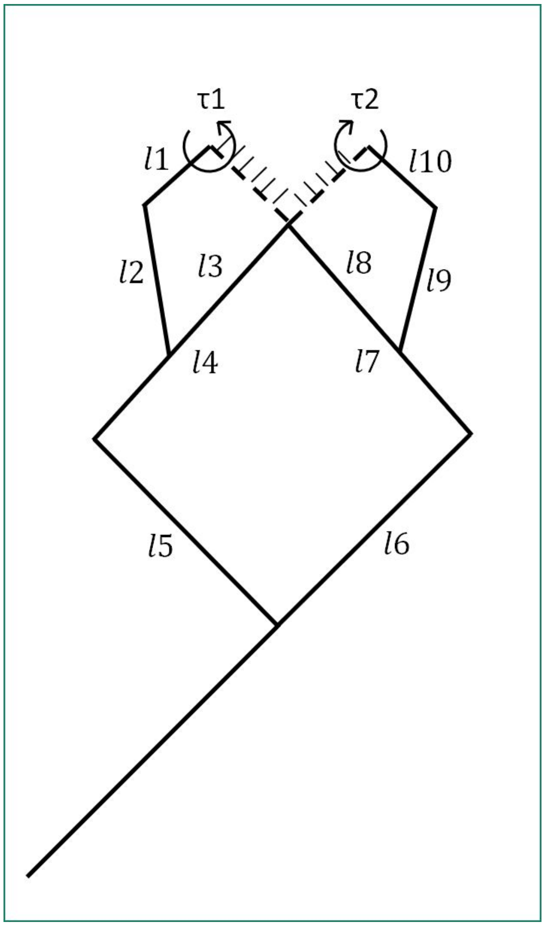
Table (1): Robot Characterization

## Design and Manufacturing

### *A. Laminate Manufacturing*

The robot is built out of laminate materials, meaning that it uses paper or fiberglass to create rigid layers, and a thin film or nylon to create a flexible layer in between them. Sheets of adhesive bind the layers together after being heated.

In order to design the laminate device, DXF generating tools in Python were used for early prototypes, and popupCAD was used for later ones. Each layer is cut out individually on a laser cutter, and then stacked and bonded together for a final cut. The resulting laminate is then cut again, releasing it from the surrounding web of scrap material. The device created by this process is then assembled into a robot by connecting actuators and is controlled by a controller.




Figure(2): Labeled Kinematics Side View

## B. Four-Bar Two DOF Actuated Leg

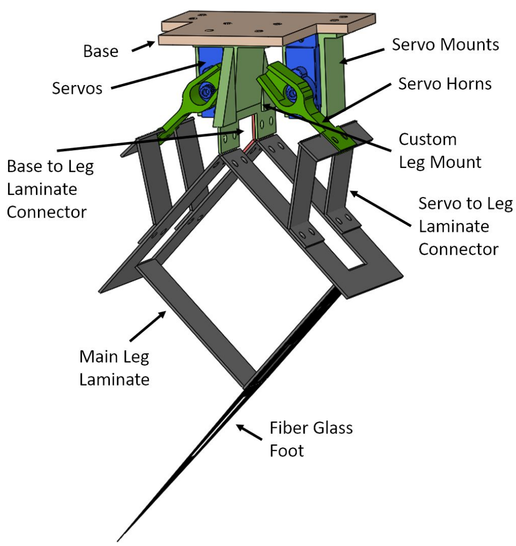
| Variable | Value | Variable | Value |
|---|---|---|---|
| $l1$ | 1.0 | $l6$ | 2.5 |
| $l2$ | 1.35 | $l7$ | 2.5 |
| $l3$ | 1.5 | $l8$ | 1.5 |
| $l4$ | 2.5 | $l9$ | 1.35 |
| $l5$ | 2.5 | $l10$ | 1.0 |

Table (2): Lengths of Kinematic Linkages

A single leg of this robot consist of three four-bar linkages connected in a parallel-series configuration as seen in Figure (2). This system is actuated by two servo motors fixed to the body and connected to two joints $l1$ and $l10$. The output four-bar linkages, consisting of links $l1$, and $l2$, connect to the body linkage $l4$, a distance $l3$ away on the left side, and the other output four-bar linkage consisting of $l9$, and $l10$, are connected to the body linkage $l7$ a distance $l8$ away on the right side. These output four-bars permits these two actuators to control the position of a foot connected to the four-bar mechanism output, consisting of linkages $l4$, $l5$, $l6$, and $l7$, with two degrees of freedom. The lengths of the links and position of the servos have been carefully selected to maximize the range of motion of the output, or foot of the leg, which is discussed in the analysis section. The values for those lengths can be found in Table (2). Connections made between off-the-shelf purchased components, such as the servos, are provided by modular 3D-printed parts. These parts are specifically designed to interface with the laminate system with a focus on aligning the joint axes from servos along the laminate midplane, permitting direct connections to moving parts. Focus has also been paid to ensure that thin laminates are constructed or reinforced with stiff materials, with the goal of improving the lifespan of the transmission when under payload. This leg design is reused for all four legs of the robot, and connected to the main body in a modular way, so that the location and orientation can be adjusted quickly. This permits rapid redesign of the legs or body as the design evolves.

## C. Laminate Topology and Assembly

Different layouts of the same mechanism have been investigated throughout the course of this project. The mechanism was originally designed using a single laminate layer. When it was compared to a modular mechanism made out of multiple layers of laminates, it was determined that using a multiple layer device provided better structural stability. A modular laminate mechanism is made out of multiple separate laminate parts that are connected using screws or plastic rivets. Each leg is symmetric along its sagittal plane, with the motors mounted directly overhead, which allows the forces and torques transmitted through the linkage to be balanced evenly. The design seen in Figure (2) was selected for its superior stiffness and cancellation of unwanted torsional effects at the joints.




Figure(3): Labeled Isometric Full Leg

The current design fully labeled can be seen in Figure (3). Fiberglass is ideally used for all of the laminates because of its durability and ability to be cut with a laser cutter. The base is connected to the servo mounts and the custom leg connector. Servos are placed in the mounts and connected to servo horns. The laminate pieces are individually manufactured through the laminate manufacturing process and then brought together to build the rest of the leg. There are three laminate parts in total, the main four-bar leg, the part that connects the servos to it which creates two more four-bars, and the connector that holds the four-bar to the base. There is a single point of the four bar that stays stationary and is mounted using the custom leg connector, to the base to leg laminate connector. The foot is connected by plastic rivets to one side of the four-bar mechanism. The flex sensor is attached to the foot by using tape and laminating methods. The wires from the flex sensor are strung through the open parts of the leg and tied to the servos to keep them from obstructing the motion of the four bar. The leg is then connected to a base that holds three more legs, and that is how this device is made quadrupedal. The servos and flex sensors are connected to a primary controller that controls the robots motion and intakes data from the sensors.

### *D. Force Sensing Feet*

There is a flex sensor attached to each leg of the robot. The purpose of these sensors is to provide the controller with load distribution information about each foot.

The variety of thicknesses available and the ability to customize the geometry of the foot with laser cutting permits us to tune the stiffness of the foot in such a way that it is responsive enough to be sensed by the flex sensor, yet rigid enough to support the weight of the robot. Due to the given mass of the robot, a thickness of 0.2 mm was chosen for the thickness of the feet.

The profile of the foot itself is an important aspect for understanding how the foot deflects under load due to contact with the ground. This curvature is important because it determines the overall sensitivity of the flex sensor which it is attached to. Our approach, which uses Euler-Bernoulli beam theory, produces a profile which creates a constant radius of curvature in order to maximize the flex-sensor signal. This process is described in the cantilever beam bending analysis part of the Analysis section.

## Analysis

### *A. Python Leg Trajectory*

A 2D motion trajectory of the leg was modeled in Python using libraries such as numpy and scipy[2]. By modeling leg design, and being to change the lengths of the linkages and the angles between them, we can make useful design choices when improving the kinematics of the leg. Constraint equations were generated that define the lengths of links, and were programmed to produce a non-zero error when they were invalid. This was fed into a nonlinear minimization function provided by scipy, with an initial guess. The process is iterated with different angles to make a leg trajectory that can be plotted. This method is effective for modeling the 2D motion of the leg as a function of actuator inputs.

While the design can be implemented in a variety of ways, the 2D planar kinematic model established in Python is sufficient for understanding the motion of the leg. This allows us to visually identify and debug undesirable leg trajectories, singularities, and other issues that can occur in a motion trajectory. One of the primary goals of the model was to increase the range of the servo actuators in each leg to maximize the range of motion of each servo.

Time-offset sine waves are used to create cyclical motion at the foot the actuators according to the rule:

$$\theta_1 = A_1 sin(t - \phi_1) + offset_1$$
$$\theta_2 = A_2 sin(t - \phi_2) + offset_2$$

Where $A_1$ and $A_2$ are the angle range of each servo, $\phi_1$ and $\phi_2$ are the offsets in the sin cycle, and $offset_1$ and $offset_2$ are offsets that place the rotating linkages $l1$ and $l10$, seen in Figure (2), in the

[2] The code for this can be found at: https://github.com/bshuch/leg-trajectory

correct starting location. This method of creating a gait gives our control space six variables of which to create a cyclical trajectory instead of a infinite control space. This is what we did for our analysis, but more general time based signals could be applied in the future.

There are two actuators on each leg, which makes it possible to create a number of different trajectories with the same sine wave function by modifying the lengths of the linkages, the range of angles they can cycle through, their offset in the sine wave cycle, and allows for more results than a traditional more complex linkage would allow for. Figure (4) displays a variety of such patterns generated on the same leg design. The values for each of the variables for the 50 degree range can be seen in Table (3), and the values for the linkages can be seen in Table (2).

This simulation was used during the design phase to find the maximum ranges of motion permitted by design candidates while design parameters were still being investigated. For example, in early stages, servo 1 could move 120 degrees of motion without problems, but servo 2 could only move around 15 degrees. By modifying the design in the simulation, the servos were able to create better motion trajectories in each iteration.

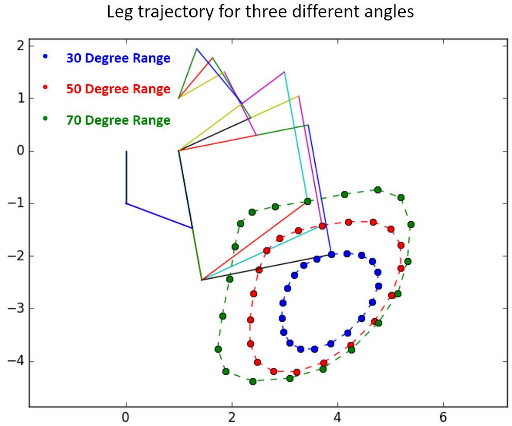



Figure (4): Leg Trajectory with three servo

| Variable | Value (°) |
|---|---|
| $A_1$ | 50 |
| $A_2$ | 50 |
| $\Phi_1$ | 0 |
| $\Phi_2$ | 90 |
| $offset_1$ | -90 |
| $offset_2$ | 90 |

Table (3): Leg Trajectory Function Values

### B. Cantilever Beam Bending Analysis

The goal of the cantilever beam analysis is to show what profile of beam gives a constant radius of curvature, and to design the feet in this way to maximize the output signal of the sensors. The feet are treated as cantilever beams. If they were rectangular beams, their moment of inertia would be:

$$I = \frac{b*h^3}{12}$$

Where $b$ is the base of the foot in meters, and $h$ is the height or thickness in meters. If the profile of the foot changes from a rectangle to a triangle, the moment of inertia changes, and there is now two $b$ variables, $b1$ and $b2$. The variable $b1$ is the start width, and $b2$ is a number closer to zero representing the end of the foot. The variable $b$ goes from being a single value to an equation:

$$b = b1 * \frac{l-x}{l} + b2 * \frac{x}{l}$$

Where $b1$ is the initial starting base, $b2$ is the end base, $l$ is the length of the beam, and $x$ is a change number between zero and the length of the beam. When $x = 0$, $b = b1$, and when $x = l$, $b = b2$. This means that the foot starts at a thickness the same as $b1$ and ends at a thickness the same as $b2$. This can visually be seen in Figure (5), where the red line is the starting point of the beam and the point near $b2$ is the end. The equation for the moment of inertia now looks like this:

$$I = h^3 * (\frac{1}{12} * (b1 * \frac{l-x}{l} + b2 * \frac{x}{l}))$$

The following are the formulas used to determine the output of the foot deflection, where M is the torque or moment on the beam, P is the point force, and E is the Modulus of Elasticity of the material. In the case of a point load,

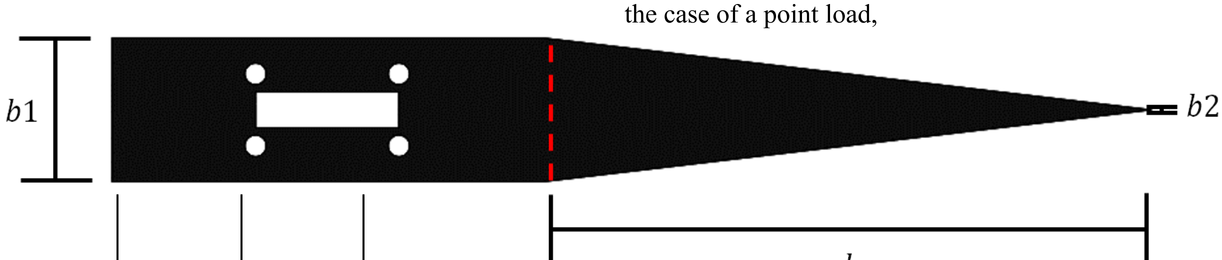



Figure (5): Labeled Foot

$$M = P * (l - x)$$

The second derivative of deflection determines concavity,

$$d\theta = \frac{12*P*(l-x)}{E*h^3*(\frac{b1*(l-x)}{l}+\frac{b2*x}{l})}$$

The second derivative is integrated to get $\theta$ the first derivative, which will give the slope, and then is then integrated again to give the equation for the deflection. These two equations are too long to show.

## Experiment

### *A. Forces Applied to Feet*

Data from the sensors was recorded using an Arduino serial input linked to a Matlab code. The Arduino code displays the reading from the flex sensor, and the Matlab code recorded a data point for each cycle of the arduino code.

Each force sensor was calibrated using a weight of 100 grams. Data was recorded with 0 grams placed onto each feet, and then data was recorded for 100 grams placed onto each foot individually. That data is then used to create a linear relationship between force and the ADC serial input reading from the Matlab code, where the force applied to the foot is either 0N or 100 grams * 9.81 m/s^2 to create .981N.

Once the force sensors were calibrated, a multiple position experiment was run on the robot. The robot moved from a stationary standing position where the legs are mostly level with each other to one where legs 3 and 4 were raised higher than legs 1 and 2. Then, leg 4 was lifted and the other three legs are moved into a position that compensates for that. From this data the location of the resultant force can be calculated and plotted. The formula for calculating its location is as follows:

$$Xpos = \frac{x1*F1 + x2*F2+x3*F3+x4*F4}{F1+F2+F3+F4}$$

$$Ypos = \frac{y1*F1 + y2*F2+y3*F3+y4*F4}{F1+F2+F3+F4}$$

These formulas tell the X and Y locations of the resultant force based on the force readings from the force sensors.

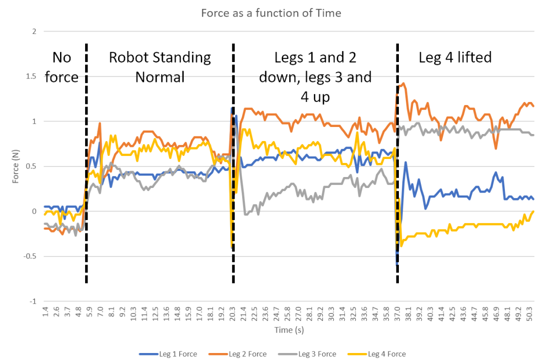



Figure (6): Graph of force sensing feet transitions

## Results and Discussion

The evolution of the forces in the feet of the robot can be seen in Figure (6) as it transitions between several postures. The graph that describes the position of the resultant forces for the different sections of that graph is Figure (7). It takes one data point from each of the four sections of this graph and, are put through calculations found in the previous section to find the location of that force. The sensors are not perfectly calibrated, so that is why the position of the point when there is no force on the legs is not perfectly centered. However, this does not ruin the integrity of the data. It makes sense that when leg 4 is lifted, the force shifts towards the opposite leg on the opposite side, so the overall trends from the robots force changes seem to make sense. However, more data needs to be collected to further analyze all of legs being lifted off the ground to have a nicer plot showing more data points.

The force sensors are able to provide readings that are sufficient to justify the further use of them. There is definitely potential for a programming solution that intakes this data real time and allows the user to control where the location of the resultant force on the robot is. This means that the robot can use information from the sensors to change its motion, meaning that the platform does what it is supposed to.

Overall, this platform is accessible to many labs because of its price, and its manufacturing requirements of having a laser cutter, heat press, and 3D printer. Laminate manufacturing makes this possible in the first place.

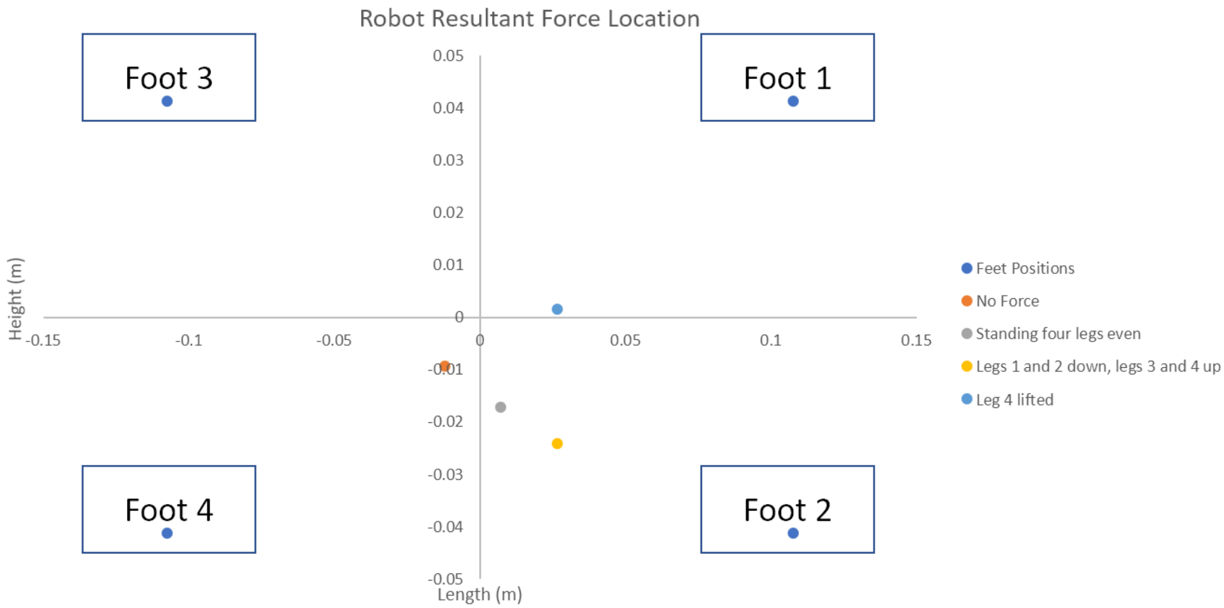



Figure (7): Resultant Force Location

## Conclusions & Future Work

In this paper, we have presented a quadrupedal laminate robot that offers a rich control space. This platform allows a user to learn more about motion gaits from quadrupedal robots, and to apply various machine learning and optimization methods to allow it to move. Each leg provides two actuators to create interesting motion paths over more mechanically defined ones, and there is a flex force sensor on each of the legs to provide the user feedback from the motion of the robot.

Potential redesign includes using as rigid, light, and as flat materials as possible to create the main body. Servos can be replaced with more powerful ones to increase the total weight the robot can carry, and the controller can be replaced with a custom PCB to reduce the weight added to the robot as much as possible, while also powering it with a battery to make it completely standalone. A wireless controller could be used to control the robot and send data back to a computer. This would get rid of any issues caused by wires creating forces where they are unwanted.

This robot is not perfect and is prone to tipping. A redesign would help mitigate that, however this issue underscores the reason why balance-sensing is so important in a non-optimized mechanical design. This robot needs good control in order to walk and balance well. This paper's goal was to introduce the design and implementation of the platform, and motivate the need for such control in future work.

## References


[1] D. M. Aukes, B. Goldberg, M. R. Cutkosky, and R. J. Wood, "An analytic framework for developing inherently-manufacturable pop-up laminate devices," Smart Materials and Structures, vol. 23, no. 9, p. 094013, Sep. 2014.

[2] D. M. Aukes, Ö. Ozcan, and R. J. Wood, "Monolithic Design and Fabrication of a 2-DOF Bio-Inspired Leg Transmission," in Biomimetic and Biohybrid Systems, vol. 8608, A. Duff, N. F. Lepora, A. Mura, T. J. Prescott, and P. F. M. J. Verschure, Eds. Cham: Springer International Publishing, 2014, pp. 1–10.

[3] D. M. Aukes and R. J. Wood, "PopupCAD: a tool for automated design, fabrication, and analysis of laminate devices," 2015, p. 94671B.

[4] A. T. Baisch, O. Ozcan, B. Goldberg, D. Ithier, and R. J. Wood, "High speed locomotion for a quadrupedal microrobot," The International Journal of Robotics Research, vol. 33, no. 8, pp. 1063–1082, Jul. 2014.

[5] P. Birkmeyer, K. Peterson, and R. S. Fearing, "DASH: A dynamic 16g hexapedal robot," 2009, pp. 2683–2689.

[6] P. S. Gollnick, S. P. Magleby, and L. L. Howell, "An Introduction to Multilayer Lamina Emergent Mechanisms," Journal of Mechanical Design, vol. 133, no. 8, p. 081006, 2011.

[7] M. Hutter, C. Gehring, M. Bloesch, M. A. Hoepflinger, C. D. Remy, and R. Siegwart, "STARLETH: A COMPLIANT QUADRUPEDAL ROBOT FOR FAST, EFFICIENT, AND VERSATILE LOCOMOTION," in Adaptive Mobile Robotics, WORLD SCIENTIFIC, 2012, pp. 483–490.

[8] A. Jansen, K. S. Luck, J. Campbell, H. B. Amor, and D. M. Aukes, "Bio-inspired Robot Design Considering Load-Bearing and Kinematic Ontogeny of Chelonioidea Sea Turtles," in Biomimetic and Biohybrid Systems, vol. 10384, M. Mangan, M. Cutkosky, A. Mura, P. F. M. J. Verschure, T. Prescott, and N. Lepora, Eds. Cham: Springer International Publishing, 2017, pp. 216–229.

[9] Y. Mulgaonkar et al., "The flying monkey: A mesoscale robot that can run, fly, and grasp," 2016, pp. 4672–4679.

[10] C. D. Onal, R. J. Wood, and D. Rus, "An Origami-Inspired Approach to Worm Robots," IEEE/ASME Transactions on Mechatronics, vol. 18, no. 2, pp. 430–438, Apr. 2013.

[11] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, "BigDog, the Rough-Terrain Quadruped Robot," IFAC Proceedings Volumes, vol. 41, no. 2, pp. 10822–10825, 2008.

[12] K. Sebastian Luck, J. Campbell, M. Jansen, D. Aukes, and H. Ben Amor, "From the Lab to the Desert: Fast Prototyping and Learning of Robot Locomotion," 2017.

[13] B. G. Winder, S. P. Magleby, and L. L. Howell, "Kinematic Representations of Pop-Up Paper Mechanisms," Journal of Mechanisms and Robotics, vol. 1, no. 2, p. 021009, 2009.

https://spectrum.ieee.org/automaton/robotics/military-robots/boston-dynamics-ls3-robot-mule