Active Robot Fall-Absorption Mechanism

Mason Carter
School of Computing, Electronics and Mathematics
University of Plymouth
Plymouth, UK
mason.carter@students.plymouth.ac.uk

Luka Danilović
School of Computing, Electronics and Mathematics
University of Plymouth
Plymouth, UK
luka.danilovic@students.plymouth.ac.uk

Elliott White
School of Computing, Electronics and Mathematics
University of Plymouth
Plymouth, UK
elliott.white@students.plymouth.ac.uk

Abstract *– The purpose of this research project was to design and build a prototype robot to serve as a proof of concept for an Active Robot Fall-Absorption Mechanism. Large robots with a small foot-base will inevitably fall over, so we have designed a mechanism to dampen the fall of a robot and reduce any damage caused to it from the fall. Upon successful completion of the building of the mechanism and conducting tests, our findings indicate that with slight improvements, originally not possible due to budget constraints, this project would yield a beneficial mechanism capable of performing its task of dampening a fall of a much larger robot.*

Keywords *– Fall absorption, agonist-antagonist, variable stiffness, soft robot.*

I. Introduction

As the trend of Robotics leans towards creating taller robots, more researchers are struggling with the task of trying to keep a robot with a small foot-base upright. The robot will, inevitably, fall over. This can cause massive damage to the robot, mostly consisting of mechanical damage to the links between joints, or servo damage. Until fall prevention becomes more effective, a simpler fall-damage-reduction system would be incredibly useful.

We propose to design and create a robotic body and leg, for the provided budget of £70.00. The main body will consist of agonist-antagonist actuators and can detect when a robot has fallen past its self-balancing point. A leg will then reach out and use artificial tendons to reduce the falling speed of the body. This can be comparable to how a human might brace themselves when falling. The artificial tendons are used to store potential spring energy.

II. Design

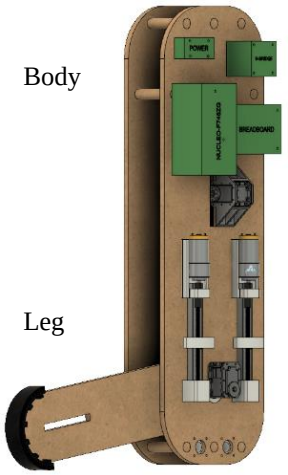

Figure 1 - Overall Design

A. Mathematical Theory

Before the release of the leg, the falling motion of a robot is the same as that of an inverted pendulum. The model of an inverted pendulum and its corresponding equations [1] can be used to calculate the angular acceleration and thus the fall time of the robot. The calculations assume no friction or other resistance to movement, and any acceleration is due to gravity.

$$\tau = mgl\sin\theta \tag{1}$$

$$\tau = I\ddot{\theta} = ml^2 \times \ddot{\theta} \tag{2}$$

$$ml^2\ddot{\theta} = mgl\sin\theta \tag{3}$$

$$\ddot{\theta} = \frac{g}{l}\sin\theta \tag{4}$$

$$t = \sqrt{(90-\theta)/\ddot{\theta}} \tag{5}$$

Where:

- τ is the net torque around the pivot point.
- m is the mass of the body.
- g is the acceleration due to gravity.
- l is the length of the mass from the pivot point.
- θ is the angular displacement from the equilibrium.
- I is the moment of inertia of point mass.
- $\ddot{\theta}$ is the angular acceleration of the pendulum.
- t is the time to fall from the current angular displacement to perpendicular with the angle of equilibrium.

The time to fall can then be used to find the 'activation window', or the amount of time the leg will have to activate and contact the ground before the body is past its maximum angle. The maximum angle is an angle from the equilibrium that we feel the body could not be saved from with the use of the leg.

$$T_A = \sqrt{(90-\theta_1)/\ddot{\theta}_1} - \sqrt{(90-\theta_2)/\ddot{\theta}_2} \tag{6}$$

Where T_A is the activation window, θ_1 is the current angular displacement, θ_2 is the maximum angle, $\ddot{\theta}_1$ is the angular acceleration for the current angular displacement, and $\ddot{\theta}_2$ is the angular acceleration for the maximum angle.

With the calculation of the activation window, we can find the torque required to accelerate the leg quickly enough that it will contact the ground first.

$$\ddot{\theta}_{LEG} = \frac{\theta_{LEG}}{{T_A}^2} \quad (7)$$

θ_{ARM} is the angle from its stored position (parallel to the body), and can be calculated using the following equation.

$$\theta_{LEG} = \theta - 90 - \sin^{-1}\frac{l_1 \sin(90-\theta)}{l_2} \quad (8)$$

Where θ is the angular displacement of the body from equilibrium, l_1 is the distance of the leg pivot to the base of the body, and l_2 is the length of the leg.

Given the angular displacement, we can then calculate the angular acceleration required to activate and contact the ground in the activation window (Results for all cases in Appendix 1).

$$\tau_{LEG} = m_{LEG} \times {l_{LEG}}^2 \times \ddot{\theta}_{LEG} \quad (9)$$

Where τ_{LEG} is the torque around the leg's pivot point, m_{LEG} is the mass of the leg, l_{LEG} is the distance of the legs centre of mass to the leg pivot point, and $\ddot{\theta}_{LEG}$ is the angular acceleration of the leg required for the activation window.

As we can only make a rough estimation as to the dimensions of the body and leg, upper, middle and lower estimations were evaluated. 20° and 70° were chosen for the minimum and maximum activation angles. The torque required to activate by the maximum angle was calculated for each degree between the minimum and maximum angle, though not including the maximum.

This produced 50 results for each estimation, and with there being 5 unknown variables that can be estimated(l, l_1, l_2, l_{LEG}, m_{LEG}), the number of results quickly increases. To more quickly process and evaluate these results, a MATLAB script was written to carry out the calculations and output a '.csv' file which can then be opened in Microsoft Excel. The variables were evaluated for values between 0.1 and 0.5, at intervals of 0.1. With the maximum angle set at 70°, we decided that the latest the leg should be able to activate would be at 45°, since the torque required for the leg to activate and contact the ground after 45° but before 70° would be incredibly large. Calculations showed that the torque required after this point could be up to 50000 Nm. With these variables, the minimum torque required would be 0.04 Nm and the maximum would be 93.7 Nm.

B. CAD and Mechanical Design

1) Design Overview

The body consist of two 9mm thick MDF, laser cut plates, (see Figure 1 for full CAD overview). The front plate, called the spine, serves as the attachment point for the PCBs, wiring, left and right tendon tensioning mechanisms, tendon guides, leg encoder Dynamixel servo, leg release Dynamixel servo and the metal leg axle with bearing assembly. The Dynamixel servos were not included in the project budget as these were available for use when the project was undertaken. The second plate at the back serves to add depth to the body and provide additional structural support to the leg axle. This plate is attached to the spine via 7 dowels. The body has rounded corners at the bottom face so that area in contact with the ground will remain constant during the tipping of the body. Lighter components such as the PCBs have been mounted towards the top of the body with heavier components being pushed towards the base as much as the design permits. An accelerometer has been mounted at the top in line with the middle of the leg and oriented along the pivoting axis. (STL files for all components are available in Appendix 2).

2) Leg Design

The Leg consist of a single link as a multiple link configuration was found to be too complicated for a negligible improvement in performance. The leg can deploy along the tipping axis in forwards or backwards directions and is slightly raised from the base of the body to allow for deployment angle greater than 90°. The base of the leg contains a series of holes for attaching the tendons as well as two pullies of constant diameter around which the tendons wrap to ensure a constant force is exerted on the leg regardless of its orientation. The tendons are kept from tangling by using guide tubes to ensure alignment between the pulleys and the tensioning mechanism. The top of the leg is outfitted with a circular profile rubber "foot" to constantly provide same area of contact with the ground. Furthermore, the foot is 3D printed using NinjaFlex filament which creates a soft model. The foot also includes a self-resettable crumple zone to provide an additional element of compliance during a fall and to minimise the impact force of the leg. The leg also has positional feedback provided by the bottom Dynamixel (with gears removed to enable 360° free rotation), while the leg release is controlled by a lever attached to a second Dynamixel positioned near the top of the leg. The whole leg is supported by a metal axle with bearings on both ends in the spine and back plate. This provides a stable axis of rotation and eliminates wobbling even in the cases of the hard impacts of the leg against the floor.

Figure 2 – Leg

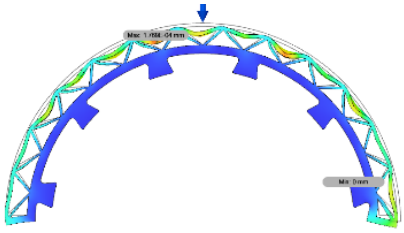

Figure 3 - Foot deformation under 500N Load

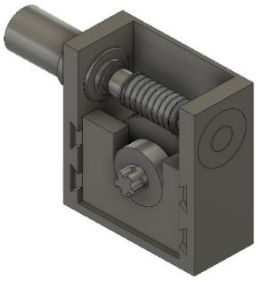

Figure 4 - Worm & Gear gearbox

3) Tensioning Mechanism Design

During the preliminary calculations phase, a goal was imposed on the tensioning mechanism so that it will be capable of 150mm of linear pull in 1 second. To achieve this a 12V DC motor was selected for its speed and torque. A 1974RPM (≈2000RPM) output was chosen for convenience with the highest torque option for the gearbox being selected at 1284gcm. To translate the motor rotation into linear movement of the tendons, a winding pulley was used with worm & gear gearbox (Figure 4) to make the tensioning mechanism non back-drivable. Therefore, the size of the pulley was calculated according to equation 10.

$$d = \frac{60LN}{\pi R}\ ; \quad \begin{cases} d = Pulley\ Diameter \\ L = Linear\ Pull\ per\ Second \\ N = Gear\ box\ ratio\ 1{:}N \\ R = RPM\ of\ the\ Motor \end{cases} \qquad (10)$$

After prototyping, it was determined that the best gearboxes worked with a ratio between 1:40 & 1:60. Therefore, number of teeth picked for the worm wheel is 48 to give a pulley diameter of 70mm. This in turn translates to a circumference of 220mm (>150mm). However, once the whole mechanism was assembled and run, there were friction problems which would cause failure of parts over prolonged use. Due to our budget constraints, we were unable to find a suitable prebuilt metal alternative and have therefore changed the design of the tensioning mechanism. Following the preliminary design review, a decision was taken to use a screw mechanism based on 3D printer Z-axis movement (Figure 5). A prototype was built, and a test was conducted to assess the viability of the mechanism. After a successful test, a full-scale device was built with support for 1.75mm and 2.5mm Fila-Flex tendon thickness. Minor

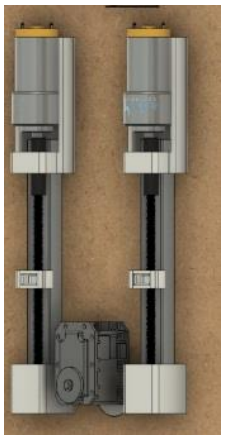

Figure 5 - Tensioning Mechanism Final Design

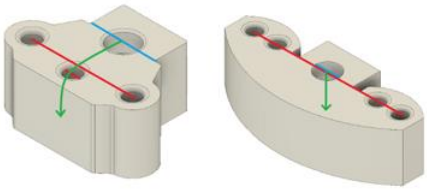

Figure 6 - Old and New Runners with respective Angular Moments

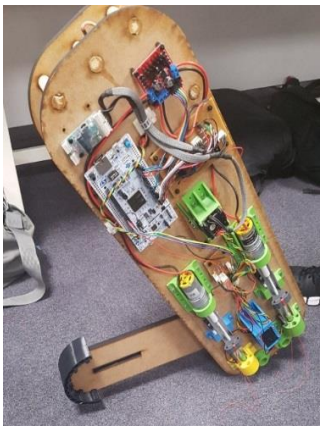

Figure 7 - Assembled Mechanism

changes were made to the shape of the tendon attachment points on the linear runner to reduce the angular moment on the runner and screw rod (see Figure 6) and therefore maximise the motor torque utilisation. This was done by aligning the mounting holes with the plane of the degree of freedom to achieve perpendicular force. The final design, when assembled, can be seen in Figure 7.

4) Tendon Design

The original tendon design was heavily inspired by the GummiArm [2]. Using a single strand of 2.85mm Fila-Flex and wrapping a string of nylon with ≈ 5mm spacing, we were able to test the tensioning mechanism but discovered that the strength of the nylon was not enough. Therefore, we switched to 1.75mm Fila-Flex and created a three-strand braid with 4th and 5th strands of nylon woven through. However, after conducting tensile testing on all configurations we realised that braid did not provide significant improvement over a straight three strand tendon, although it did add to the manufacturing complexity. To mitigate this, we switched back to a single strand of 2.85mm thick Fila-Flex. The final design that was chosen consists of double nylon strands connected to the runners on one end and via an adapter a strand of 2.85mm thick Fila-Flex on the other. Thicker filament was chosen for its strength and deformation properties. Fila-Flex then wraps around the

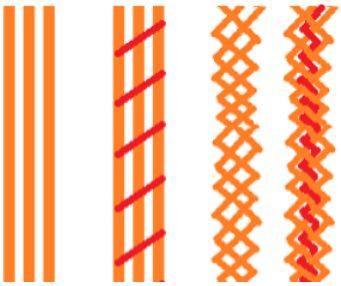

Figure 8 - Multi-Strand Tendon Designs

pulleys and connects to the leg. As seen in Figure 9, there is a secondary nylon strand present in parallel with the Fila-Flex to take the force once the Fila-Flex is stretched passed the designated limit.

Figure 9 – Compound Tendon Diagram

C. Electronics & Software Design

The whole system is built around a STM32F746ZG MCU which sits upon a NUCLEO-F746ZG development board, produced by STMicroelectronics. The motor driver selected is a Dual H-Bridge Module produced by Hobby Components, which contains a L298N produced by STMicroelectronics. The L298N can provide up to 2A continuous per channel, or 2.5A repetitive (80% on, -20% off). This motor driver module is connected to two RS PRO brushed DC geared motor. These DC motors have a stall current of 2.26A. The L298N can handle the stall current of these motors, but it should be noted that if other, more powerful motors were to be used, then a different motor driver would need to be chosen. The six-axis inertial measurement unit selected is an inexpensive MPU-6050, produced by TDK InvenSense, and comprises of a gyroscope and an accelerometer. This IMU sits on a breakout board produced by SODIAL, the GY-521. Communication to the sensor is achieved through I2C. Dynamixel servos have been used in the design. The two Dynamixel servos are RX-28s, and these require the RS-485 communication protocol. Because of this, a MAX3088 RS-485 transceiver IC has been used in line with a MAX3378 voltage-level shifter and are connected to a UART port on the NUCLEO board with a baud rate of 57600. The servos can be connected in a daisy chain-like manner. Microswitches have been used on the linear actuators to provide positional feedback of the linear runners when they reach their limit. LEDs have also been used for visual debugging and system information. An outline of the system can be seen in Figure 10.

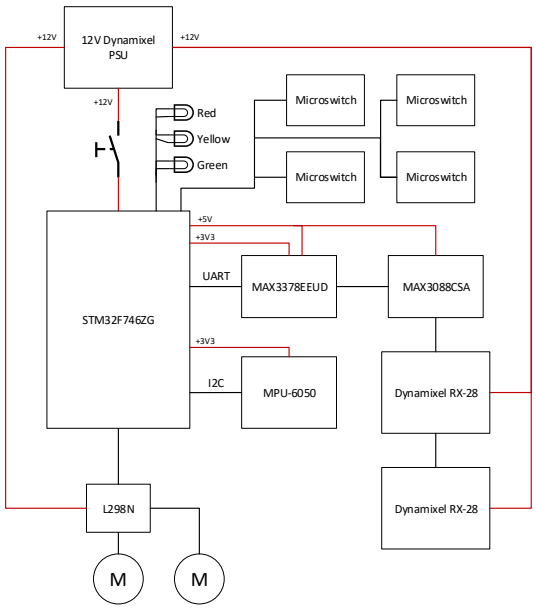

Figure 10 - Electronics System Outline

One of the Dynamixel RX-28s has been used as an encoder to provide positional feedback of the leg. To achieve this, the gears have been removed so the servo is free to spin. Communications between the RX-28s and the NUCLEO board occur by sending and receiving digital packets. A library on Mbed had been provided [3] to enable communications with the servos. However, this library had to be modified to get it working with the RX-28s. Timings in the existing library had been incorrect, so the microcontroller would completely miss the returning status packet from the servos. If the microcontroller tried to communicate with a servo that wasn't in the servo chain, then the code would enter an infinite wait. This has been changed to include a timeout if a status packet isn't received within 5000 checks of the serial input buffer. A new function had been implemented to find the servo ID numbers of the servos in the servo chain. This means that the servos could be switched out with other RX-28s, and the code would still work. The baud rates of the Dynamixel servos can be changed by modifying data in the registers on the on-board EEPROM. Because of this, some of the servos had different baud rates to each other at the start of the project. An algorithm was implemented to quickly adjust the baud rates of the servos in the chain by utilising the "broadcast" command, which sends an instruction to all the servos on the chain. This algorithm ensured that all servos in the chain had the same baud rate before the rest of the code ran. Tests were done to identify the servos in the chain and to move the servos to different positions. Reading back the positional data was also done to ensure that the communications with the servos were established correctly.

To test and use the MPU-6050, an existing header and main file had been used that was provided on Mbed [4]. These files provided all the functionality to read acceleration and angular data. The algorithm uses Quaternions to avoid gimble lock in the gyroscope. This is especially useful in this scenario as the robot will approach +/- 90 degrees in one axis. The Quaternions are then converted to Euler angles using the conversion equations 11-13.

$$\phi\ (yaw) = arctan\left(\frac{2(ab+cd)}{a^2-b^2-c^2-d^2}\right) \quad (11)$$

$$\theta\ (pitch) = -arcsin\big(2(bd-ac)\big) \quad (12)$$

$$\psi\ (roll) = arctan\left(\frac{2(ad+bc)}{a^2+b^2-c^2-d^2}\right) \quad (13)$$

To test the IMU, continual data was streamed to a serial monitor to observe the angular data of the device. The device was then rotated around its separate axis and the measured data was compared to the real angles of the device.

All the software has been written in C++ using ARM's Mbed OS, which is a platform and operating system for devices based on the 32-bit ARM Cortex-M microcontrollers. This has enabled rapid code development and fast deployment. The algorithms used focus heavily on gyroscopic and accelerometer data as it is this data that

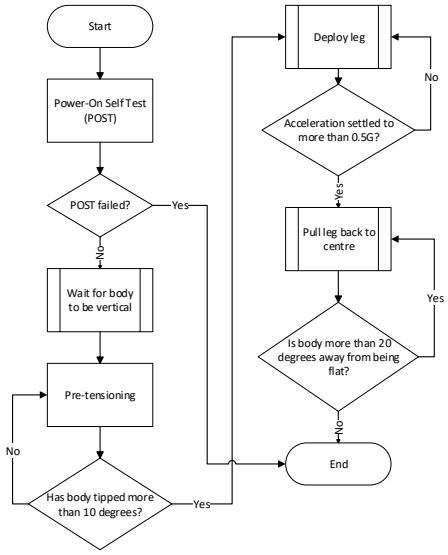

Figure 11 - Code Algorithm Outline

provides all the information about the given state of the body. Data from the IMU is gathered asynchronously as it is not critical to observe the physical state of the body at a given rate. An outline of the algorithm used for this project can be seen in Figure 11.

As seen in Figure 11, the algorithm begins by running a POST test. This ensures that all components are working and that communications with the Dynamixels are established. The pre-tensioning function tensions the tendons on the leg before the body has begun to fall. This is done by detecting which direction the body is starting to tip, and subsequently pull on the relevant tendons. If the body is tipped more than 10 degrees, this is classed as a 'fall' and the leg is deployed in the correct direction. The acceleration data is then monitored until it settles. This shows that the body has stopped falling and is resting on the leg. The leg is then pulled back towards the centre of the body using positional data from the Dynamixel on the shaft of the leg to monitor its progress.

III. Experiments

A. *Tensile Testing*

In order to pretension the agonist-antagonist joints, an elastic material would be used as a tendon. The kinetic energy from the pre-tensioning would be stored in the tendon to be released as needed. Flexible 3D printing filament (FilaFlex [5]) would be used, both because of cost and its elasticity at break of 665%.

Three strands of the filament would be used, because the pre-tensioning forces had been estimated to be too much for a single strand, and to also provide a built in redundancy in the case of a strand snapping. As with rope, it was thought that the tensile strength of the material could be increased by braiding the strands, so test samples were produced for both un-braided and braided. However, while an elastic material is needed, the elasticity of the filament seemed too high, so a compound tendon using Nylon was considered. Test samples for unbraided with nylon and braided with nylon were also produced for tensile testing. Two test samples for each configuration of the strands were tested for their tensile strength, but because of inadequate clamping

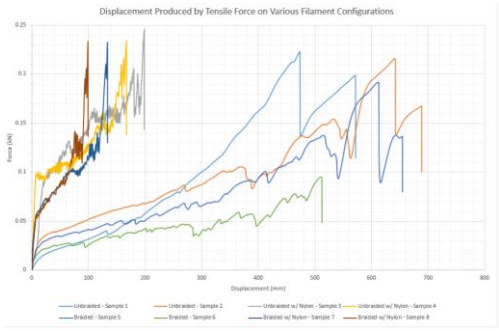

Figure 12 - Copy of the Tensile Testing Results Graph

methods the samples tended to slip through the clamping mechanism. Appendix 3 contains the results of the testing. The graph in Figure 12 and in Appendix 3 shows that while the samples with and without Nylon are clearly separated, braiding has no effect on the tensile strength of the material.

B. *Fall Testing*

In order to be able to correctly use the accelerometer and gyroscopic data, the body needed to be repeatedly pushed over and the data recorded. This allowed the system to know the state of the body based on the data. The tendons were configured in a way that would allow the body to rest at an angle with all the weight resting on the tendon. The right motor was spun until the linear actuator had pulled a substantial amount on the tendon. This can be seen in Figure 13. The body was then up-righted and pushed over with a small amount of force. The acceleration in the x-axis was monitored and plotted through a serial connection to a MatLab script. This plot can be seen in Figure 14. At sample 150, the body starts to fall. At 380, the Fila-Flex starts to dampen the fall and slow the body down, and at sample 420 the nylon takes over which causes the large jump in acceleration. As the body settles to its resting position, so does the acceleration which settles at 0.9G. It does not settle at 0G as the accelerometer is measuring the acceleration due to Earth's gravity. Initially, a value of 0 will be measured, but as the body tips over more, the value will increase until it reaches 1G when the body is perpendicular to the floor. This data has helped shape the control algorithm (Appendix 4). If the acceleration has settled to a high value, then it can be known that the body has stopped moving and is resting on the leg. The leg can then start to be pulled back towards the body and lower the body down towards the ground. It was decided that if the acceleration settles to a value above 0.5G, then the body has successfully fallen and is resting on the leg (Appendix 5 - Video).

Figure 13 - Body Resting on Leg

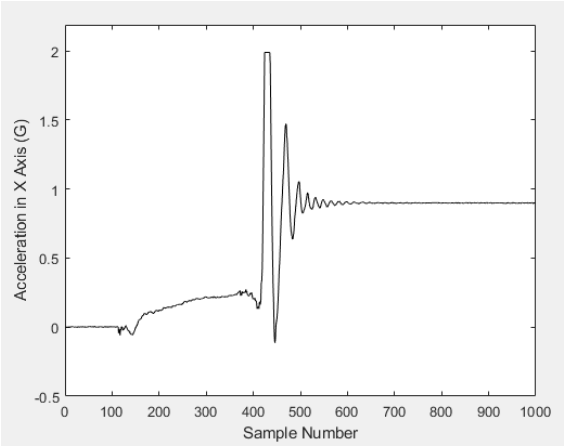

Figure 14 - Graph Showing Change in Acceleration in X-Axis

C. *Lowering Body Testing*

The next experiment was testing to see if the body could be lowered slowly to the ground once it had rested on the leg. The body was placed at an angle with all of the strain on the tendons, and then the right motor was spun in such a way that would pull the leg into the body and allow the body to reach the floor. It was found that the motor would stall and not have enough torque to move the linear actuator. The body would not be able to be lowered down to the floor. The force of the tendons pulling on the linear runner was causing too much friction between the nut and the threaded rod, stopping the motor from spinning.

D. *General Testing*

Once the body was built there was a lot of general testing of tipping the body and pushing it over, and then observing how the control algorithm was performing. During this testing, it was found that the tendons would never tension enough when the body was falling. The RPM of the motors were not high enough to move the linear actuator far enough in the time given for the body to fall over.

We also found that when the leg sprung out and the body tipped, the tendon would become displaced from the pulley (see Figure 15). This resulted in a much looser tendon as it was not stretched around the pulley. This, coupled with the fact that linear actuator would not move far enough, meant that the tendon would never be taught enough to have any effect on dampening the fall of the body.

Figure 15 - Tendon Slipped off Pulley

IV. Conclusion

We have built a platform that, with more work, will be capable of demonstrating the use of agonist-antagonist joints and artificial soft tendons to absorb a fall of a robot. The software algorithms that are in place are capable of successfully executing a fall-reduction system, but it is the hardware that causes limitations. The motors do not have enough torque to slowly lower the body down to the floor, nor have a high enough RPM to tension the leg to a high enough degree before the body reaches the floor. The biggest improvement to be made would be changing the DC motors to a more powerful version. This would enable a faster actuation of the linear actuators and allow the actuators to move when under heavy load and not stall. This change would also require a change to the motor driver IC to a part that would be able to handle the higher current. A new pulley would have to be designed to stop the nylon-to-Fila-Flex adapter from slipping off during leg deployment. This would be a simple redesign and the modification that would need to be made would be to widen the pulley to the extent that there is no space between the pulley and the body for the adapter to fall into. Another modification would be the leg pre-tensioning and release mechanism. The new control algorithm and mechanism would pre-tension both tendons at the same time regardless of which direction the body is falling. Then, upon detecting a fall, the mechanism would release one tendon and tighten the other one further. This would allow a quicker response time as there is no wait for one tendon to be tensioned and the other one un-tensioned during the time before a fall is detected. The budget of £70.00 was a major limiting factor, and with a larger budget and with the previously-mentioned changes made, we believe the system would be successful in its ability to absorb the impact from a fall and not allow any damage to be caused to the robot it is attached to.

Appendix

Appendix 1 – Force Calculations
https://github.com/ElliWhite/ROCO504/tree/master/Force%20Calculations

Appendix 2- STL Files
https://github.com/ElliWhite/ROCO504-Advanced-Robot-Design-Project/tree/master/STL%20Files

Appendix 3 – Tensile Testing Data and Results
(Graph available in “Displacement vs. Force” worksheet).
https://liveplymouthac-my.sharepoint.com/:x:/r/personal/elliott_white_students_plymouth_ac_uk/_layouts/15/Doc.aspx?sourcedoc=%7Bae856e5c-52d3-4ef5-bb82-3eb3c2f34c49%7D&action=default

Appendix 4 – GitHub Repository
(ReadMe available) https://github.com/ROCO504/2018-Group1
Populated repository can be found at:
https://github.com/elliwhite/roco504

Appendix 5 – Video demonstrating the Project
https://youtu.be/zlB8UxlkwFY

ACKNOWLEDGMENT

This research was supported by Martin Stoelen from the University of Plymouth CRNS. We thank our colleagues from University of Plymouth Robotics Department who provided insight and expertise that greatly assisted the research, although they may not agree with all of the interpretations/conclusions of this paper.

V. REFERENCES

[1] Wikipedia, "Inverted Pendulum," 15 June 2018. [Online]. Available: https://en.wikipedia.org/wiki/Inverted_pendulum. [Accessed October 2018].

[2] M. Stoelen, "GummiArm," 2017. [Online]. Available: https://mstoelen.github.io/GummiArm/. [Accessed November 2018].

[3] D. Gongora, "Dynamixel," 25 October 2015. [Online]. Available: https://os.mbed.com/users/dmgongora/code/Dynamixel/. [Accessed October 2018].

[4] K. Winer, "MPU6050IMU," 29 June 2014. [Online]. Available: https://os.mbed.com/users/onehorse/code/MPU6050IMU/.

[5] Recreus, "FilaFlex Original 82A," 2017. [Online]. Available: https://recreus.com/img/pdf/FILAFLEX%2082A%20TECHNICAL%20SHEET%20&%20MSDS%202017.pdf. [Accessed October 2018].