TSA Antagonistic Triad - Overview

Damian Crosby

February 8, 2021

1 Introduction

Actuated Universal Joint (AUJ) mechanisms are found in a wide range of robotic applications, such as confined space inspection using snake-arm robots [?], highly manoeuvrable mobile snake robots [?], and biomimetic robot tails for stability [?]. Mobile snake robots must usually incorporate electric actuators inline with their joints. This results in an AUJ having to shift the mass of the follower segments and all the actuators inside the follower segments, which results in high torque requirements and therefore Snake-arm robots and robotic tails can reduce the mass and size of the AUJ by moving their actuators away from the AUJs and use cables to transfer the force to the joints, or use hydraulic or pneumatic actuators which tend to be lighter than equivalent electric motors, at the expense of increased mass and bulk at the base of the arm or tail.

First developed by [?] in 2010, the Twisted String Actuator (TSA) uses two or more strings between two fixtures as a linear actuator. When one fixture is rotated (typically by an electric motor), the looped string twists into a helix, decreasing the distance between them. Equation 2 calculates the distance l_s for a two string system with infinite stiffness for a given rotation angle θ_s, where l_u is the unwound length between the fixtures and r_s is half the string thickness.

The primary advantage of TSA over similar linear actuators such as a leadscrew is the reduction the TSA provides is not proportional (or even slightly inversely proportional) to the mass of the actuator. By decreasing r_s (or $r_s + r_c$ when there are more than two strings, where r_c is the "core radius", or half the gap between any two strings that intersects the axis of rotation) the reduction increases for a constant l_u and θ_s, resulting in a greater reduction with no increase, or even a slight decrease, in actuator mass. While the reduction in a leadscrew can be increased by decreasing the lead on the thread (typically denoted as λ) which also has no increase in mass, this has a limited range and can quickly run up against manufacturing tolerances or material strength requirements. In order to achieve greater or more robust reductions, the screw radius has to be increased, or the driving motor has to have a larger reduction before driving the screw, both of which usually result in more material (typically steel) and therefore more mass.

However, TSA does have some drawbacks, the most significant of which is a non-linear reduction equation, which is also dependent on the motor angle θ_s (and therefore actuator position). The reduction decreases in a non-linear fashion as the angle increases, with the derivative decreasing as the angle increases. There is also the compliance of the strings to consider, depending on the thickness and material chosen, which becomes a significant factor under high forces. Both of these issues can be addressed with accurate modelling [] and/or a robust control strategy, as demonstrated in [?]. What is perhaps more an issue regarding mass for an AUJ is the unidirectional force of the TSA, which can only impart force in tension. This means that for an AUJ, a minimum of three TSA are required unless spring return mechanisms are used, which would impart additional force on the TSA and therefore reduce performance. However, the potential high force to mass ratio of the TSA due to the non-proportional reduction may adequately compensate for the additional actuator requirement.

The focus of this research is to discover if the TSA is a suitable candidate for control of an AUJ considering both the benefits and drawbacks. To this end, the first objective is to simulate a model and then construct a physical experimental prototype to validate the proposed control system. Then, the second objective is to compare the results from both the simulation and experimental prototype with other inline actuation methods for an AUJ to analyse and discuss the potential advantages and disadvantages of TSA actuation against alternatives.

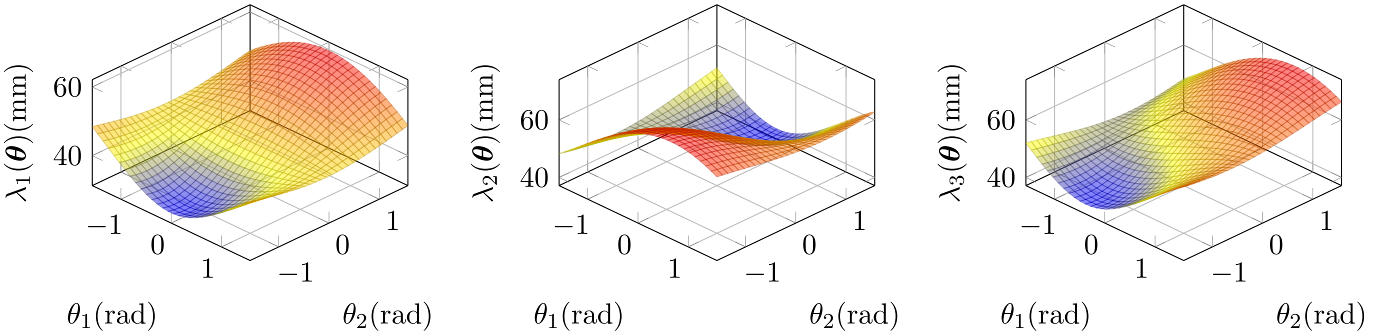

Figure 1: Surface plots of each element of the vector function $\Lambda(\boldsymbol{\theta})$, assuming coefficient values from table 1.

2 Antagonstic Triad

As mentioned in the introduction, because the TSA provides only tensile force, a minimum of 3 actuators are required. These can be arranged in a triangular configuration to create an "antagonistic triad", akin to the antagonistic pairs of muscles found in animals. Where a revolute joint would be found between the connecting ends of the actuator, a universal joint is found instead. The geometric structure of the system can be described with two equilateral triangles of inradius r on two planes separated in the z axis. The centroids are then connected via a universal joint from each plane normal to an intersecting point, described by vector $\boldsymbol{\theta} = \begin{bmatrix}\theta_1 & \theta_2\end{bmatrix}$ to denote the rotation of the second plane relative to the first, in the x and y axis around the intersecting point, and l_1 and l_2 to denote the normal distance from the intersection to the first and second plane centroids respectively. When $\boldsymbol{\theta} = \begin{bmatrix}0 & 0\end{bmatrix}$ the triangles are parallel to each other. The distance between the vertex pairs of each triangle is then denoted as $\begin{bmatrix}\lambda_1 & \lambda_2 & \lambda_3\end{bmatrix}$ for the "top", "left" and "right" vertices of the triangles. When $\boldsymbol{\theta}$ is changed, this will change λ_1, λ_2 and λ_3 respectively. A diagram of this is shown in figure 2.

To calculate the lengths of the strings for a given $\boldsymbol{\theta}$ of the AUJ, we can define a vector function $\Lambda(\boldsymbol{\theta}) = \begin{bmatrix}\lambda_1(\boldsymbol{\theta}) & \lambda_2(\boldsymbol{\theta}) & \lambda_3(\boldsymbol{\theta})\end{bmatrix}$ with l_1, l_2 and r as the coefficients, where the scalar functions are defined in equation 1.

$$
\begin{aligned}
\lambda_1(\boldsymbol{\theta}) &= \sqrt{\begin{aligned}&(l_1 + l_2\cos\theta_1\cos\theta_2 + r\cos\theta_1\sin\theta_2)^2\\ &+ (r - r\cos\theta_2 + l_2\sin\theta_2)^2\\ &+ (l_2\cos\theta_2\sin\theta_1 + r\sin\theta_1\sin\theta_2)^2\end{aligned}}\\
\lambda_2(\boldsymbol{\theta}) &= \sqrt{(a-b+c)^2 + (l_1-d)^2 + (e)^2}\\
\lambda_3(\boldsymbol{\theta}) &= \sqrt{(a+b-c)^2 + (l_1+d)^2 + (e)^2}\\
&\text{where:}\\
&\quad a = -\tfrac{\sqrt{3}r(\cos\theta_1 - 1)}{2}\\
&\quad b = l_2\cos\theta_2\sin\theta_1\\
&\quad c = \tfrac{r\sin\theta_1\sin\theta_2}{2}\\
&\quad d = \tfrac{\sqrt{3}r\sin\theta_1}{2} + l_2\cos\theta_1\cos\theta_2 - \tfrac{r\cos\theta_1\sin\theta_2}{2}\\
&\quad e = \tfrac{r\cos\theta_2}{2} - \tfrac{r}{2} + l_2\sin\theta_2
\end{aligned}
\tag{1}
$$

These were simply calculated by computing the transformation matrix for each string from one end to the other in cartesian coordinates, then acquiring the Euclidean norm of the resulting translation vector $\left\|\begin{bmatrix}x_n & y_n & z_n\end{bmatrix}\right\|_2$.

3 Twisted String Actuator (TSA)

$$l_s(\theta_s) = \sqrt{l_u^2 - \theta_s^2 r_s^2} \tag{2}$$

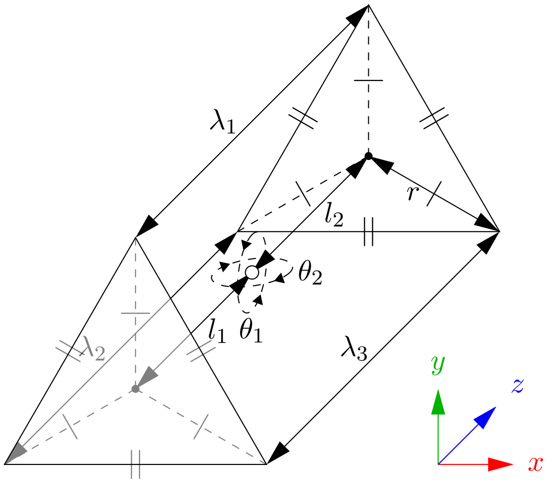

Figure 2: Kinematic diagram of the antagonistic triad.

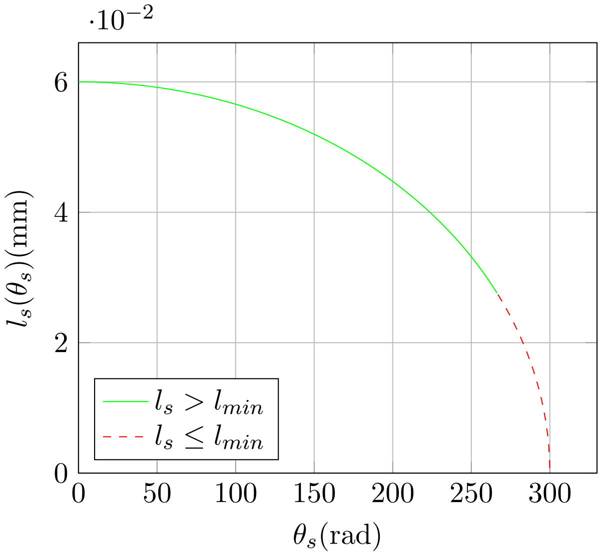

Figure 3: TSA string length against motor angle with coefficients from table 1.

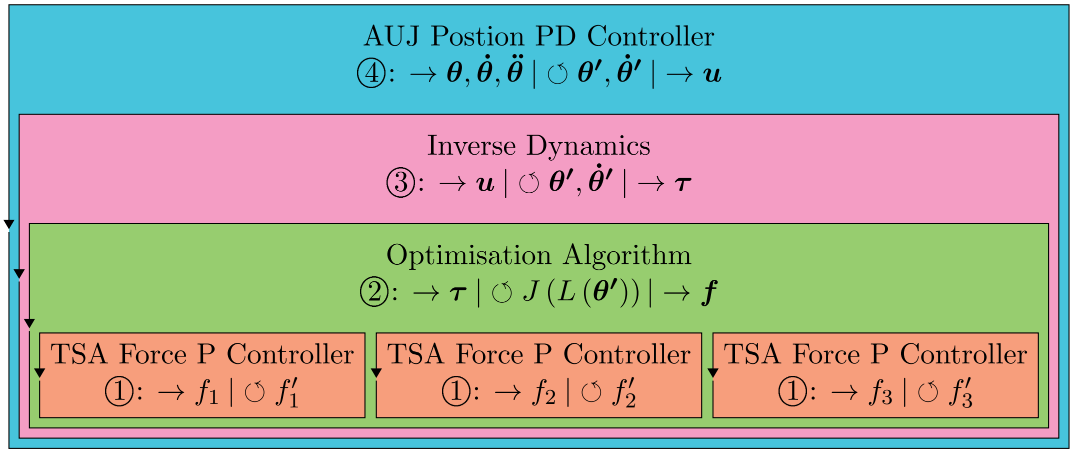
AUJ Postion PD Controller
④: $\rightarrow \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}} \mid \circlearrowleft \boldsymbol{\theta'}, \dot{\boldsymbol{\theta}}' \mid \rightarrow \boldsymbol{u}$
Inverse Dynamics
③: $\rightarrow \boldsymbol{u} \mid \circlearrowleft \boldsymbol{\theta'}, \dot{\boldsymbol{\theta}}' \mid \rightarrow \boldsymbol{\tau}$
Optimisation Algorithm
②: $\rightarrow \boldsymbol{\tau} \mid \circlearrowleft J\left(L\left(\boldsymbol{\theta'}\right)\right) \mid \rightarrow \boldsymbol{f}$
TSA Force P Controller
①: $\rightarrow f_1 \mid \circlearrowleft f_1'$
TSA Force P Controller
①: $\rightarrow f_2 \mid \circlearrowleft f_2'$
TSA Force P Controller
①: $\rightarrow f_3 \mid \circlearrowleft f_3'$

Figure 4: Outline of the cascade function.

$$l_{min} = \frac{l_u}{\sqrt{\frac{\pi^2}{2} + 1}} \approx 0.46\ l_u \tag{3}$$

Equation 2 from [?] calculates the actuator length l_s for a given θ_s, given l_u and r_s for a two string TSA. This equation assumes an infinite string stiffness, so is only reasonably accurate under low tension. Although theoretically the stroke of the TSA can be the entire domain of $[0, l_n]$, in reality the thickness of the string prevents a geometric helix from forming once the helix pitch $q < 4r_s$ (or $q < 2nr_s$ for n strings). This limits the lower bound of the stroke to the value calculated by equation 3, or approximately 46% of l_n for a two string TSA.

4 Control System

The control system is a 4 layer cascade design, using feedback of the joint position from the accelerometers and TSA force from the load cells. It uses a second order setpoint trajectory $\boldsymbol{q}$ as input, which can either be pre-defined or generated dynamically from user input. Feedback is provided by the AUJ angular position $\boldsymbol{\theta}$, angular velocity $\dot{\boldsymbol{\theta}}$, and TSA tension force $\boldsymbol{f}$.

1. C_1 AUJ Position PID Controller
2. C_2 Inverse Dynamics
3. C_3 TSA Force Optimisation Algorithm
4. C_4 TSA Force P Controller

Functions $C_{1\ldots4}$ are then combined into a cascade function: $C_4\left(C_3\left(C_2\left(C_1\left(\ldots\right),\ldots\right),\ldots\right),\ldots\right)$.

4.1 AUJ Position PID Controller

This function is a PID controller with the input $\boldsymbol{q}$ as the setpoint and the AUJ angular position $\boldsymbol{\theta}$ and velocity $\dot{\boldsymbol{\theta}}$ as feedback, plus the addition of a feedforward term for the input acceleration $\ddot{\boldsymbol{q}}$.

$$\begin{gathered} \boldsymbol{\epsilon} = \boldsymbol{q} - \boldsymbol{\theta} \\ \dot{\boldsymbol{\epsilon}} = \dot{\boldsymbol{q}} - \dot{\boldsymbol{\theta}} \\ C_1\left(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}\right) = K_P\boldsymbol{\epsilon} + K_I\int_0^t \boldsymbol{\epsilon}dt + K_D\dot{\boldsymbol{\epsilon}} + \ddot{\boldsymbol{q}} \end{gathered} \tag{4}$$

In the discrete implementation used for fixed step simulation and experimental model control, the integral term is replaced by the trapezoidal rule.

$$K_I \int_0^t \boldsymbol{\epsilon} \approx \sum_{i=0}^{N} \frac{\boldsymbol{\epsilon}\left(t_i\right) + \boldsymbol{\epsilon}\left(t_{i-1}\right)}{2} \Delta t \tag{5}$$

4.2 Inverse Dynamics

This function converts the control signal from C_1 to the desired AUJ torque using the Euler-Lagrange method in compact matrix form. Firstly the affine transformation matrix T for a coordinate frame between the AUJ pivot and the centre of mass of the follower segment can be defined. The order of R_x and R_y can be reversed, but this requires other terms to be reversed as well.

$$T\left(\boldsymbol{\theta}\right) = R_x\left(\theta_1\right) R_y\left(\theta_2\right) P_z\left(l_2\right) \tag{6}$$

Then the linear velocity jacobian J_v is simply the jacobian of the translation vector of T.

$$J_v\left(\boldsymbol{\theta}\right) = \begin{bmatrix} \frac{\partial t_{14}}{\partial \theta_1} & \frac{\partial t_{24}}{\partial \theta_1} & \frac{\partial t_{34}}{\partial \theta_1} \\ \frac{\partial t_{14}}{\partial \theta_2} & \frac{\partial t_{24}}{\partial \theta_2} & \frac{\partial t_{34}}{\partial \theta_2} \end{bmatrix} \tag{7}$$

The angular velocity jacobian J_ω is calculated using the first joint angle relative to the base frame, and the second in the frame of the first. If R_x is the first rotation in T then the first column of the jacobian is $\begin{bmatrix}100\end{bmatrix}^\intercal$ (to represent the pitch angle) and the second column is $R_x \begin{bmatrix}010\end{bmatrix}^\intercal$.

$$J_\omega\left(\boldsymbol{\theta}\right) = \begin{bmatrix} 1 & r_{x_{11}} \\ 0 & r_{x_{21}} \\ 0 & r_{x_{31}} \end{bmatrix} \tag{8}$$

Then the mass matrix D can be created from the jacobians, the mass for the follower segment m and its inertia tensor $I \in \mathbb{R}^{3\times3}$, and R_x and R_y to express the inertia in the correct frame.

$$D\left(\boldsymbol{\theta}\right) = m J_v^\intercal J_v + J_\omega^\intercal \left(R_x R_y\right) I \left(R_x R_y\right)^\intercal J_\omega^\intercal \tag{9}$$

The centrifugal/coriolis matrix C is created from the christoffel symbols of D, along with the AUJ velocity vector $\dot{\boldsymbol{\theta}}$.

$$C\left(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}\right)_{k,j} = \sum_{i=1}^{N} \frac{1}{2} \frac{\partial d_{kj}}{\partial \theta_i} + \frac{\partial d_{ki}}{\partial \theta_j} - \frac{\partial d_{ij}}{\partial \theta_k} \dot{\theta}_i \tag{10}$$

Then the gravity term G. As the gravity vector direction is the same as the z axis as in figure 2, the height is equal to $-l_2 \cos\theta_1 \cos\theta_2$, therefore the potential energy is $mg\left(-l_2 \cos\theta_1 \cos\theta_2\right)$. The jacobian of this then becomes the gravity term.

$$G\left(\boldsymbol{\theta}\right) = \begin{bmatrix} \frac{\partial mg - l_2 \cos\theta_1 \cos\theta_2}{\partial \theta_1} \\ \frac{\partial mg - l_2 \cos\theta_1 \cos\theta_2}{\partial \theta_2} \end{bmatrix} \tag{11}$$

D, C and G are then combined to form the dynamics equation C_2, along with the AUJ position and velocity vectors. C_1 is used as the acceleration term. This results in a setpoint joint torque which can be used in the optimisation algorithm.

$$C_2\left(C_1, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}\right) = D\left(\boldsymbol{\theta}\right) C_1 + C\left(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}\right) \dot{\boldsymbol{\theta}} + G\left(\boldsymbol{\theta}\right) \tag{12}$$

subsectionTSA Force Optimisation Algorithm

This function uses a modified algorithm from [?] to select an optimal force vector from the desired joint torque. A force matrix F is created from the torque input C_2, jacobian J_Λ from the vector function Λ as defined in equation 1, and minimum force constant f_{min}. f_{ii} is equal to f_{min}, while the other elements in the column are based on a calculation using $J_{\Lambda_{-i,*}}$ where $-i$ is a row removed from the matrix.

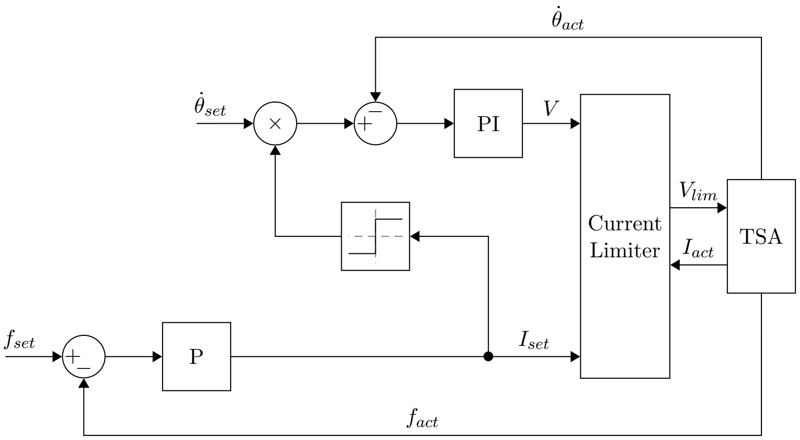

Figure 5: DCC Block Diagram.

$$\begin{aligned}
J_\Lambda &= \begin{bmatrix} \frac{\partial\lambda_1}{\partial\theta_1} & \frac{\partial\lambda_2}{\partial\theta_1} & \frac{\partial\lambda_3}{\partial\theta_1} \\ \frac{\partial\lambda_1}{\partial\theta_2} & \frac{\partial\lambda_2}{\partial\theta_2} & \frac{\partial\lambda_3}{\partial\theta_2} \end{bmatrix} \\
F\left(C_2, \boldsymbol{\theta}\right) &= \begin{cases} f_{i,i} = f_{min} \\ f_{-i,i} = -J_{\Lambda_{-i,*}}^{-\intercal}\left(J_{\Lambda_{i,*}}^{\intercal} f_{min} + C_2\right) \end{cases} \\
&= \begin{bmatrix} f_{min} & f_{12} & f_{13} \\ f_{21} & f_{min} & f_{23} \\ f_{31} & f_{32} & f_{min} \end{bmatrix}
\end{aligned} \tag{13}$$

Finally, the following algorithm selects one column of F to be the output force vector.

```
$\boldsymbol{s} \leftarrow \begin{bmatrix}\top & \top & \top\end{bmatrix}$
**if** $f_{23} > f_{min}$ **then** $s_2 \leftarrow \bot$ **else** $s_3 \leftarrow \bot$ **end if**
**if** $f_{31} > f_{min}$ **then** $s_3 \leftarrow \bot$ **else** $s_1 \leftarrow \bot$ **end if**
**if** $f_{12} \geq f_{min}$ **then** $s_1 \leftarrow \bot$ **else** $s_2 \leftarrow \bot$ **end if**
**for** $i = 1$ to $3$ **do**
    **if** $s_i \rightarrow \top$ **then** $C_3 \leftarrow \boldsymbol{f}_{*,i}$ **end if**
**end for**
```

4.3 TSA Force P Controller

This function is a P controller using the measured load cell forces $\boldsymbol{f}$. The output from this can then be used to control the motor using three different control strategies implemented within the MCDC 3002 motor controller.

$$C_4\left(C_3, \boldsymbol{f}\right) = K_{P_s}\left(C_3 - \boldsymbol{f}\right) \tag{14}$$

4.3.1 Direct Current Control (DCC)

This mode takes a preset velocity $\dot{\theta}_{set} \in [0, 4220]$ This voltage $\boldsymbol{V}$ which is multiplied by the signum of the result from the cascade function in order to ensure the motor spins in the right direction, and then uses a hardware PI controller with velocity feedback $\dot{\theta}_{act}$ to generate a control voltage. It is then passed to a current limiter with the current error (the result with the actual current I_{act} subtracted) as the limit, before being sent to the motor. This ensures the motor stops spinning when the target current is reached.

$$\begin{aligned}
\boldsymbol{\epsilon_c} &= \dot{\theta}_{set} - \dot{\boldsymbol{\theta}}_{\boldsymbol{act}} \\
\boldsymbol{V} &= \left(K_{P_c}\boldsymbol{\epsilon_c} + K_{D_c}\int_0^t \boldsymbol{\epsilon_c}\right)\operatorname{sgn}\left(C_4\left(\ldots\right)\right)\omega\left(C_4\left(\ldots\right) - \boldsymbol{I_{act}}\right)
\end{aligned} \tag{15}$$

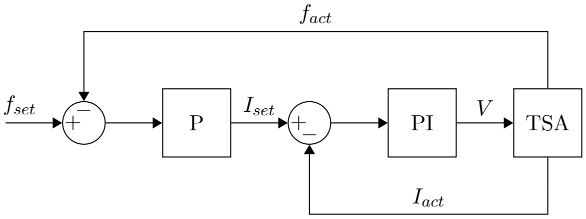

Figure 6: P Current Control Block Diagram.

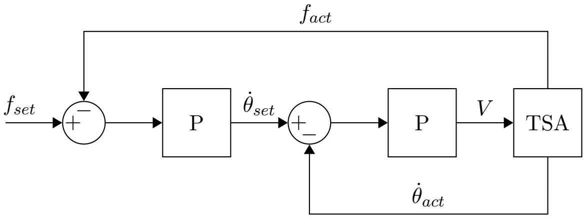

Figure 7: Velocity Control Block Diagram.

Where $\omega(\ldots) \in [0,1]$ is an unknown hardware limiting function that controls the motor speed depending on the current error.

4.3.2 P Current Controller

This strategy is a more direct method of current control, using a software P controller to directly set the voltage of the motor, using the MCDC 3002 as simply a passive amplifier. In this case the current error is passed directly to a P controller which has its output limited to prevent damage to the motors.

$$\boldsymbol{V} = K_{P_c} C_4(\ldots) - \boldsymbol{I_{act}} \tag{16}$$

4.3.3 Velocity Control

This strategy simply uses the result from the cascade function as a velocity setpoint using the hardware velocity PI controller as in the DCC.

$$\boldsymbol{V} = K_{P_c}\boldsymbol{\epsilon_c} + K_{D_c}\int_0^t \boldsymbol{\epsilon_c} \tag{17}$$

Table 1: Simulation and performance estimation coefficients.

Coefficient	**Value**	**Coefficient**	**Value**
l_1	45 mm	J	1×10^{-6} kg m^{-2}
l_2	600 µm	K_L	1000 N m^{-1}
r	13 mm	f_{min}	1 N
l_n	60 mm	K_P	800
r_s	200 µm	K_I	5000
m	72.619 13 g	K_D	50
C	0.1315 N mm	K_{P_s}	19
K_t	0.0263 N m A^{-1}	ω_s	441.9 rad s^{-1}
α_s	1×10^5 rad s^{-2}	I_s	0.19 A
τ_s	4.5 mN m		

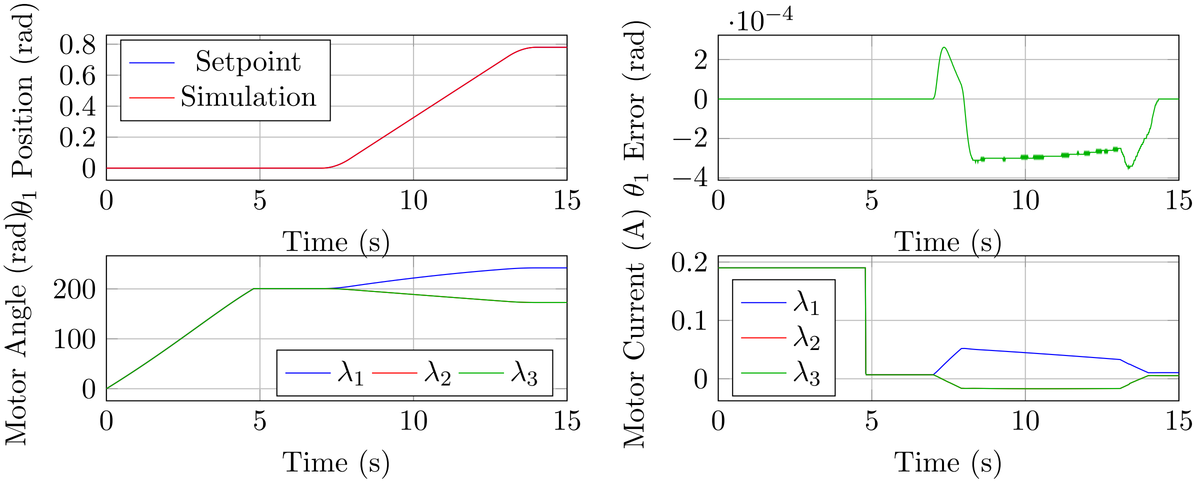

Figure 8: Simulation results for a trajectory of θ_1 from 0 to $\frac{\pi}{4}$.

5 Simulation

To design and refine the parameters of the control system, a Simscape Multibody™model of the antagonistic triad and control system was created in MATLAB®/Simulink™. This allowed for model design coeffecients $l1, 2$ and controller gains K_P, K_I, K_{P_s} to be modified in order to have the most stable control within design limits. Plots of a test joint trajectory can be seen in figure 8 with the paramaters from table 1.

5.1 TSA State Space Definition

In order to approximate a TSA plant from within a simulation, a state space model was required which takes motor current u as an input and outputs y as the TSA tension force f_i. [**?**] defines it as such, where J is the motor inertia, C is the motor coulomb friction (modified from viscous friction as the 1724TSR only has dry friction),, K_t is the motor torque constant, and K_L is the load stiffness. As the original definition is for a fixed load l_u distance from the motor a modified model is required which takes into account the varying length between the motor and load defined by $\Lambda(\boldsymbol{\theta})$. A saturation function is used to prevent incorrect compression forces when the string is slack. All of the motor coefficients were taken from the Faulhaber 1724TSR datasheet as this is the motor to be used in the experimental model. An estimated value is used for the load stiffness, this was chosen to be a high number as the model is expected to be very stiff.

$$\begin{aligned}
h(\theta_s) &= \frac{\theta r_s^2}{\sqrt{l_u^2 - \theta_s^2 r_s^2}} \\
k(\theta_s, \boldsymbol{\theta}) &= \lambda_n(\boldsymbol{\theta}) - \sqrt{l_u^2 - \theta_s^2 r_s^2} \\
\dot{\boldsymbol{x}} &= \begin{bmatrix} x_2 \\ -\frac{K_L}{J} h(x_1)\, k(x_1, \boldsymbol{\theta}) - \frac{B}{J} \operatorname{sgn}(x_2) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K_t}{J} \end{bmatrix} u \\
y &= K_L \operatorname{sat}_0^\infty k(x_1, \boldsymbol{\theta})
\end{aligned} \tag{18}$$

The state space model was then adapted to include constraints on motor velocity and acceleration set by the motor controller in order to keep the motor within design limits, by replacing $\dot{\boldsymbol{x}}$ with $\dot{\boldsymbol{x}}'$ which contains saturation functions for maximum motor velocity v_s and acceleration α_s.

$$\dot{\boldsymbol{x}}' = \begin{bmatrix} \operatorname{sat}_{\omega_s} \dot{x}_1 \\ \operatorname{sat}_{\alpha_s} \dot{x}_2 \end{bmatrix} \tag{19}$$

As the input u to the state space model was the motor current, the control signal from $C_4(\ldots)$ could be fed straight into the state space.

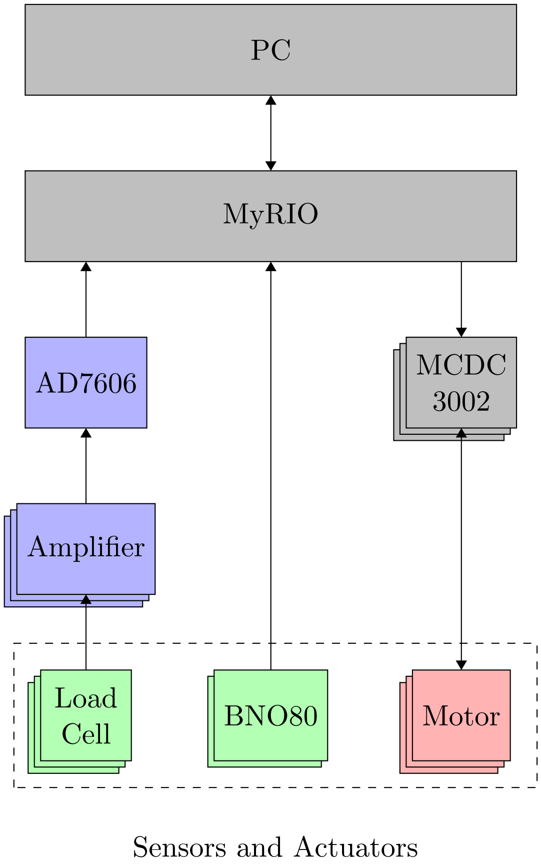

Figure 9: System architecture for the TSA Control System.

6 Experimental Model

For the experimental validation, a physical prototype of the mechanism was constructed with coefficients from table 1 as design parameters. The motors to be used would be Faulhaber 1724T024SR micromotors, selected for their compact size but high torque. These would be controlled by Fauhaber MCDC3002 motor controllers, which could interface with a National Instruments MyRIO via the USB port, using a USB to serial converter. The load cells would be Futek LCM100 minature load cells, selected for their small size. The signals from these would be amplified using a Flyde XXXX Amplifier rack and decoded using an external AD7606 ADC before being fed into the MyRIO using SPI.

6.1 TSA Control System & Motor Characterisation

Before verifying robust control of the AUJ, the control of each individual TSA needed to be tested in order to characterise the performance of the motors and to ensure the inner loop control system was robust. This would involve selecting the control strategy that gave the best performance. A test trajectory consisting of a smooth ramp followed by a sine wave was fed into the inner loop of the cascade function as $\begin{bmatrix} f(t) & f(t) & f(t) \end{bmatrix}$. Each control strategy was tested, and in the end the velocity control strategy proved most optimal, as is shown in figure 12.

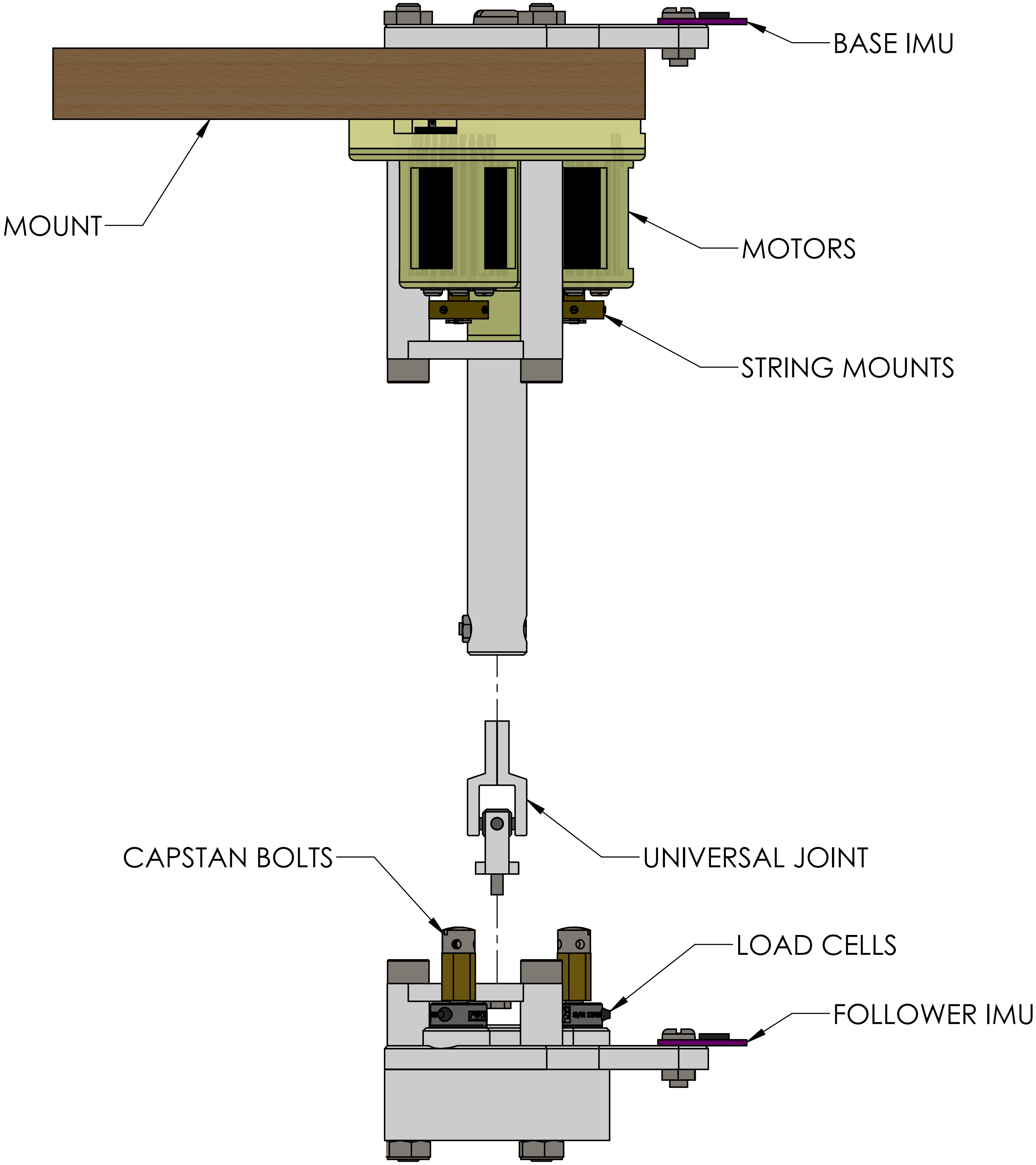

Figure 10: Schematic of the experimental model.

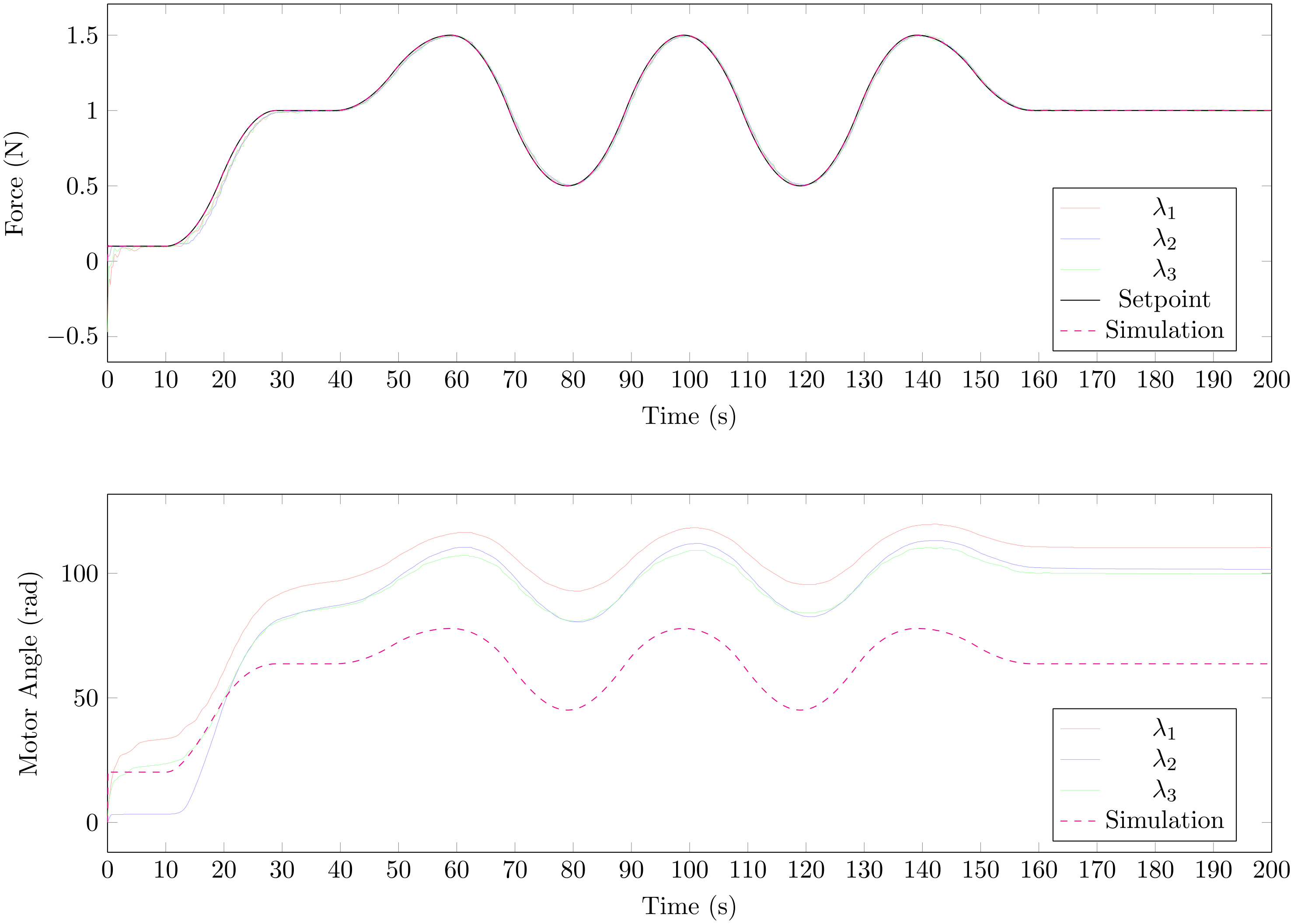

Figure 11: Load cell plot from each string using the velocity control strategy.

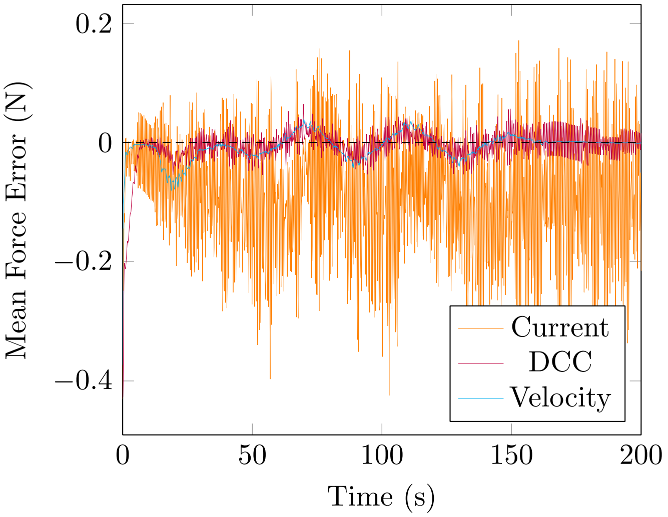

Figure 12: Mean tracking error for each control strategy.