TendonForces Forcing Module as an add-on to the PyElastica simulation software

Gabriel Tuzlaci
gabotuzlaci@gmail.com

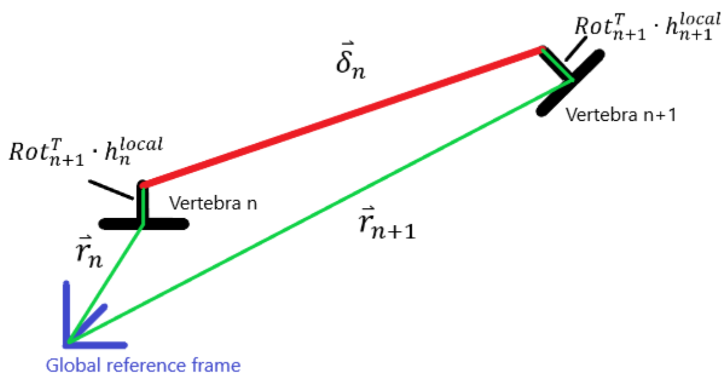
Mathematical procedure

The nature of the interaction between vertebrae, the tendon and the rod are nontrivial and are not so easily expressed mathematically. This is because these three entities interact in a nonlinear fashion: as tension increases, so does the deflection of the rod and thus the resultant forces in each of the vertebrae because of the tendon tension, which in turn affects the deflection of the rod. In a sense, it is a more complex version of the P delta problem in classical structural mechanics.

However, given the validated ability of PyElastica to numerically solve the governing equations of the rod and output the state of many variables in the system, the interaction of vertebra-tendon-rod can be implemented in a simpler manner that is better suited to match the numerical computations that the software is already doing.

The following procedure details the steps that the TendonForces module takes in order to apply tendon actuated forces and moments to the rod being simulated in the PyElastica simulator:

1. Using the position_collection data obtained from the simulator, the nodes which contain a vertebra are located in the 3D workspace, as well as their rotation matrices constructed using directors_collection. Using this data, along with the height of the vertebra, a vector is constructed which describes the relative change in position in space between the top of one vertebra to the next, as seen in the following image:



   Thus giving the following equation:

$$\vec{\delta}_n = \left(\vec{r}_{n+1} + Rot_{n+1}^T \cdot \vec{h}_{n+1}^{local}\right) - \left(\vec{r}_n + Rot_{n+1}^T \cdot \vec{h}_n^{local}\right)$$

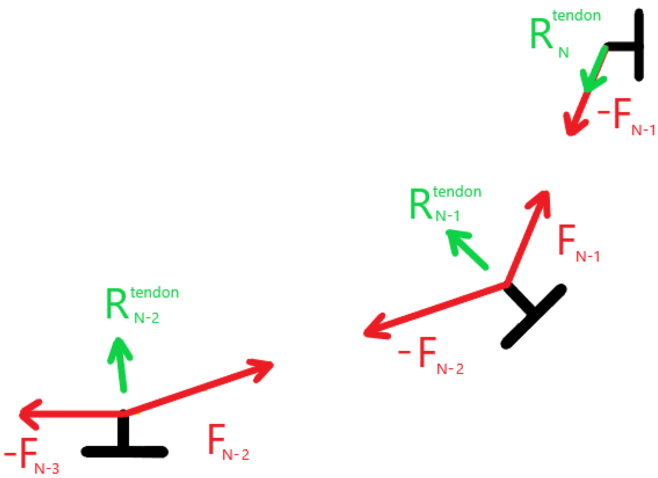
   Where $\vec{r}_n$ is the position vector of the n vertebra's node in the global reference frame (taken from position_collection), $\vec{h}_n^{local}$ is the vector which describes the local orientation and the height of the vertebra, and Rot_n is the rotation matrix associated with the n vertebra (the transposing of this matrix is done to be able to rotate the $\vec{h}_n^{local}$ vector from the local reference frame to the global reference frame, so it can then be added to $\vec{r}_n$).

2. Now, the unit-normed vector of $\vec{\delta}_n$ is calculated and then scaled by the tension T applied to the tendon, as such:

$$\vec{F}_n = T \cdot \frac{\vec{\delta}_n}{\left\|\vec{\delta}_n\right\|}$$

   Where $\vec{F}_n$ is the force vector that acts on the n vertebra, pointing in the same direction as $\vec{\delta}_n$.

3. Given that for every vertebra there are two force vectors that appear because of the tendon tension, as seen in the following image:



   Then the resultant is found by a vector sum:

$$\vec{R}_n^{tendon} = \vec{F}_n - \vec{F}_{n-1}$$

   Where $\vec{R}_n^{tendon}$ is the resultant force caused by the tendon tension in each vertebra n. It is noted that this vector sum includes a negative force, with subindex $n-1$. This is because the vector that describes this force is no other than $-\vec{\delta}_{n-1}$.

4. Because the weight of the vertebrae must also be taken into account, the final resultant force for each vertebra node is the following:

$$\vec{R}_n = \vec{F}_n - \vec{F}_{n-1} + W_n$$

   Where W_n denotes the weight vector of each vertebra n, and $\vec{R}_n$ is the final resultant force being applied to the vertebra node.

5. The effect of the torques generated along the rod because of the tendon-vertebra-rod interaction must be considered because there are forces, namely the tendon generated forces which act at points that are away from the center line of the rod. This generates torques along the rod because there are now forces and lever arms on which they act. To apply these torques, first it must be understood that PyElastica handles forces in the global reference frame and torques in the local reference frame. What follows is the rotation of the tendon forces obtained previously from the global reference frame to the local reference frame, as such:

$$\vec{R}_n^{local} = Rot_n \cdot \vec{R}_n^{tendon}$$

   Note that it is specifically $\vec{R}_n^{tendon}$ and not $\vec{R}_n$ being used, and this is because the weight included in $\vec{R}_n$ does not generate torque as it is acting upon the center line of the rod.

6. Next, the locally rotated resultant tendon force vector is cross-multiplied with the local vertebra height vector to obtain the resulting torque that will be applied to the vertebra n:

$$\vec{M}_n = h_n^{local} \times \vec{R}_n$$

7. Once calculated, $\vec{R}_n$ and $\vec{M}_n$ are applied to the system at their corresponding vertebra nodes and at the current time step.

Physical experimentation / validation

The physical experimentation and validation of TendonForces was done with a set of three experiments. The details of these experiments and their results can be found in the work (undergraduate thesis by Gabriel Tuzlaci): “*Desarrollo de Entorno en ROS2 para el Simulado y Control de un Robot Continuo Accionado por Tendones*”. The purpose of each experiment was the following:

- The building of the physical experiment environment, which consisted of an elastic silicone rod fixed in one end and fitted with 3D printed PLA discs vertebrae and a monofilament nylon wire as tendons. Measuring of the rod’s tip’s position was done using a leveled laser and grid paper placed behind the rod.

1. Experiment No. 1: calibration of the simulator’s elasticity modulus by allowing the rod to deflect under its own weight and measuring the vertical position of the rod’s tip. These measurements were taken quickly after the rod reached equilibrium at its deflection, and shortly after the rod was supported so that it would not have any deflection (this is to reduce the effect of creep exhibited by these polymers). The measurements taken (5) were identical and these were used to run simulations in the PyElastica simulator under the exact same conditions, changing only the young’s modulus until the simulated tip deflection reached the measured tip deflection.
2. Experiment No. 2: calibration of the simulator’s shear modulus by allowing the rod to deflect under its own weight as well as an actuated tendon on the rod’s horizontal plane. This was done because the rod exhibited a very slight twist when the horizontal tendon was activated, which then allowed for the calibration of the shear modulus to match the tip’s deflection. The same procedure as Experiment No. 1 was followed, except for an additional condition which was the activation of a tendon on the horizontal plane. Many simulations were carried out until the simulation’s vertical tip deflection matched the experiment’s measured vertical tip deflection.
3. Experiment No. 3: the validation of the TendonForces external forcing module was examined by carrying out a series of ten experiments and comparing the results between the simulations and the measurements. The experiments consisted on allowing for the deflection of the rod under its own weight, as well as with the activation of a tendon on the vertical plane and on the horizontal plane. The tensions applied to the tendons were not identical and they were ever increasing until the tenth experiment. The 3D position of the rod’s tip was measured and compared with the 3D position of the simulated rod’s tip. The results showed an average error in the range of 2%, low enough to be considered valid.