




Article

The Turmell-Meter: A Device for Ankle Joint Axis Estimation Applying Product of Exponentials

Óscar Agudelo ^{1,†} , Ángel Valera ^{2,†}  and Julio Vargas-Riaño ^{3,*} 

¹ Facultad de Ciencias Básicas e Ingeniería, Universidad de los Llanos, Villavicencio, Meta, Colombia; oscar.agudelo@unillanos.edu.co

² Departamento de Ingeniería de Sistemas y Automática, Universitat Politècnica de València, Valencia, Spain; giuprog@isa.upv.es

³ PhD student in Robotics, Automation and Industrial Computer Science, Universitat Politècnica de Valencia, València, Spain; julio_h_vargas_r@ieee.org

* Correspondence: julio_h_vargas_r@ieee.org

† These authors contributed equally to this work.

Abstract: The human ankle is a complex joint, most commonly represented as talocrural and subtalar axes. It is difficult to locate and take in vivo measurements of the ankle joint. There are no instruments for patients lying on a bed or the floor; that can be used in outdoor or remote sites. We have developed a "Turmell-meter" to address these issues. We started with the study of ankle anatomy and anthropometry, then we used the product of exponentials' formula to visualize the movements. Furthermore, we built a prototype using human proportions and statistics. For pose estimation, we used a trilateration method by applying tetrahedral geometry. Additionally, we computed the axis direction by fitting 3D circles, plotting the manifold and chart as an ankle joint model. We presented the results of simulations, a prototype comprising 45 parts, specifically designed draw-wire sensors, and electronics. Finally, we tested the device by capturing positions and fitting them into the bi-axial ankle model as a Riemannian manifold. The Turmell-meter is intended to be a hardware platform for human ankle joint axis estimation, it is adjustable and has an easy setup. The proposed model has the properties of a chart in a geometric manifold, we provided the details

Keywords: anthropometry; biomechanics; coordinate measuring machines; in vivo; kinematics; mechanical sensors; sensor arrays; human ankle model, operational amplifiers; pose estimation; position measurement; rehabilitation robotics; biomedical informatics; product of exponentials formula; Riemannian manifolds

1. Introduction

In this work, we present a device intended for the study of the human ankle joint (HAJ). Modeling and measuring this lower limb joint is essential in physiology, biomechanics, and rehabilitation (also in humanoid robotic limb development).

The HAJ is fundamental for human locomotion. And the ankle joint sprain is the most common lower limb injury in sports[1], football[2], basketball[3], high school sports[4], military academy[5], service[6] and physical training[7]. It causes chronic ankle instability[8], and is costly for the healthcare system[9]. Treatment and healing require measuring the range of motion.

Similar to other characteristics, the HAJ model is unique for each person. Individual variations and anthropometric measurements depend on gender, age, and phenotype. There are few types of equipment for *in vivo* ankle joint measurements on reduced spaces such as beds or for patients laying on the floor in remote places.

We present the model of the Human Ankle Joint through the Product of Exponential (PoE) formula. This method only requires two reference frames: the shank and the foot. We used this method instead of the Denavit-Hartenberg convention, which requires internal measurements in bones. The mathematical foundations are coherent with screw theory, introduced by Ball[10]; included in books[11–19], applied to multi-body systems, and geometric representation in [20–29]. Modern robotics used the product of exponential mapping to rigid body kinematics [30]. Studies on the human jaw [31,32], and the human knee [33–35], applied screw theory. Also, works dedicated to tracking limb position used inertial units [36–42].

There are different HAJ models in the literature; we focus on the two-axes approach. The International Society of Biomechanics (ISB) recommendations included it [43], also in anatomy and biomechanics books [44–48], and simulation software [49]. We found models of the ankle joints in several articles [50–56]. Contributions to the study of the ankle joint axes are in [57–60]. Finally, we found interesting research about the subtalar axis [61–68] and other functional representations [69].

Draw-wire sensors (DWS) are in robotics applications [70–72], also in linear position tracking [73], and easy robot programming [74]. Inertial measurement units (IMU) were post-processed and complemented with other sensors [75–79]. We shall employ our device for the HAJ bi-axial measurements and for other models too [80,81]. BiodexTM and HumacnormTM manufactures general kinetics machines

The Product of Exponentials (PoE) formula requires the body's initial pose; we use DWS and trilateration to find the position. Also, for tracking the foot position, we use IMU. They integrate the acceleration in real-time, which causes position estimation drift. Therefore, we combine IMU with DWS to accuracy enhancing. And the trilateration method solves the initial position problem.

A description of the anatomy and biomechanics of the two axes HAJ is in appendix A. In Fig. A1, we realize that the HAJ has parallel chains joined by nontrivial surfaces in contact. Also, in living people, it is difficult to localize the reference frame of human bones. There are other methods for HAJ axes characterization; our device is an alternative method when we cannot use cameras or other optical devices because of obstacles, space, or illumination issues.

For the device's design and implementation, we start with anatomy, statistics, proportions, and anthropometry. Then we simulate ankle joint movements by using the PoE formula. We describe the trilateration method to find the platform pose. We also show the device's entire design and implementation process. Also, we calibrate and test the device in a healthy patient and model the axis and represent the HAJ movements as a manifold chart.

2. Materials and Methods

In this section, we detail the design and implementation of the device. First, we show the simulation using anthropometric values and the PoE formula. Using the simulation plots, we estimate the DWS maximal length. Next, we present the device's geometrical design and the trilateration method. Additionally, the computer-aided design (CAD) of electronics and mechanical parts. Additionally, we implement the firmware and visualization software, and finally we compute the axis position by circle fitting and modeling the ankle joint as a manifold chart.

2.1. Ankle Joint Simulation

For the simulation with the PoE formula, we adapt the data from [82], proportions from [83,84], and statistics from [85].

2.1.1. References Assignment

Figure 1 presents the reference points and the mean distances taken from [82].

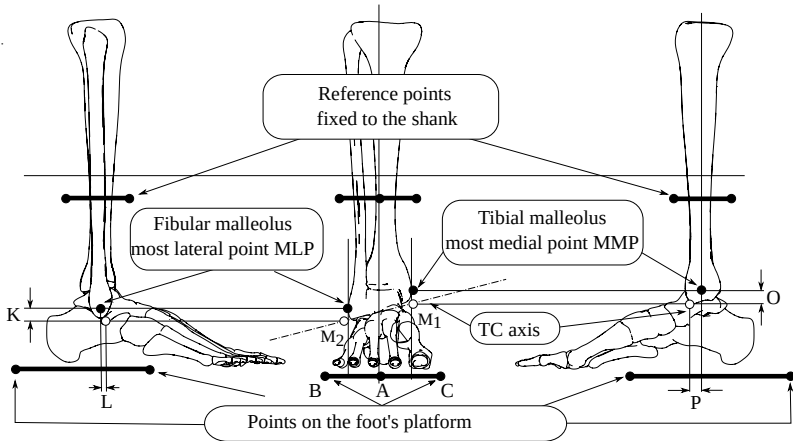


Figure 1. Reference points from anthropometric values K, L, O, and P.

A, B, and C are the triangle's vertices in a platform fixed to the foot, the K, L, and O distances from the most medial and lateral points from the black-filled to the white-filled marker. M₁ and M₂ define the TC axis. We show top-transverse and right-lateral views in Figure 2 with distances Q, W, and w. N₁ and N₂ determine the ST axis.

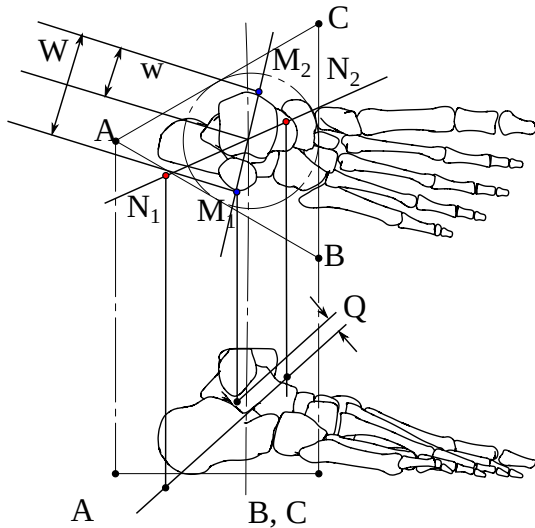


Figure 2. Q, W, and w distances from lateral and transverse views.

Table 1 enumerates the mean values of Figures 1 and 2.

Table 1. Mean values of anthropometric measurements .

Variable	K (cm)	L (cm)	O (cm)	P (cm)	Q (cm)	R=W/w
Mean	1.2	1.1	1.6	1.0	0.5	0.54

In Figure 3, we show the ST and TC axes from several viewpoints. The TC axis refers to the sagittal plane and the ST to the transverse plane.

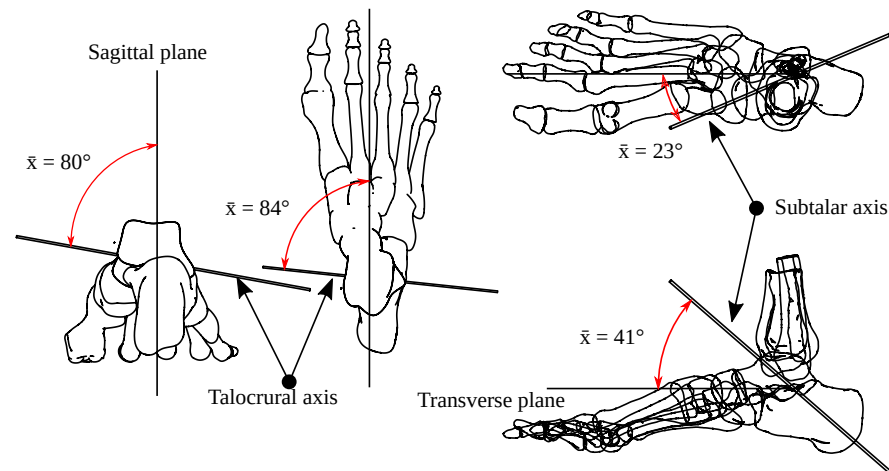


Figure 3. Q, W, and w distances from lateral and transverse views.

2.1.2. Anatomical and Geometrical Correspondence

We define the sagittal (lateral) plane as the X-Z plane (perpendicular to the y-axis). The coronal (frontal) plane is the Y-Z plane (x-axis is normal to it); the transverse (axial) plane is the X-Y plane (perpendicular to the z-axis). Figure 4, left, shows this corresponding references.

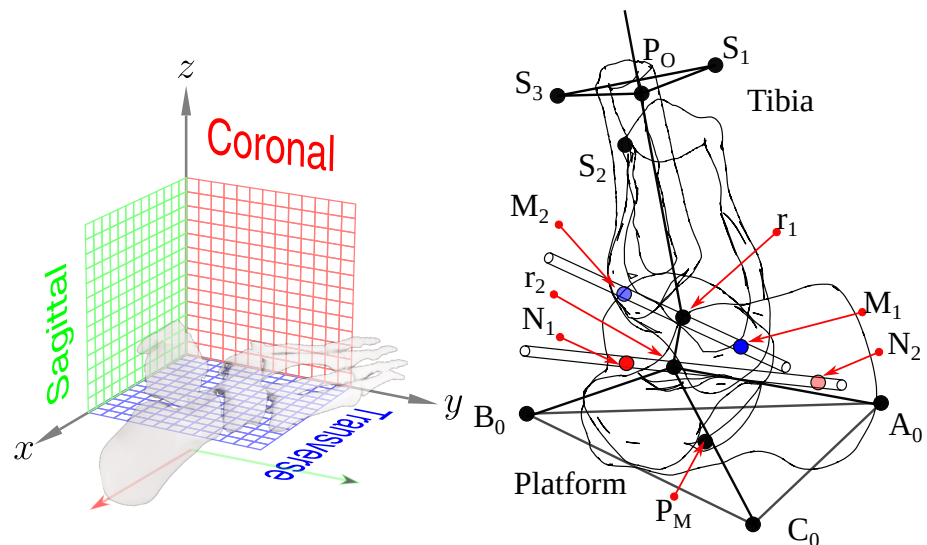


Figure 4. Planes, axes, and points corresponding references.

With this reference frame, we can define the TC axis orientation from a unitary vector in the z-direction. We first rotate it -80° around the x-axis; then we turn it -6° around the z-axis. A unitary vector in the x-axis direction defines the ST axis, rotating 41° about the y-axis, followed by a 23° rotation around the z-axis.

We show the fibula, tibia, talus, calcaneus 3D position, reference points, TC, and ST axes in Figure 4, right.

In this image, A_0 , B_0 , and C_0 are the vertices from the platform fixed to the foot, and P_M is the triangle's center. S_1 , S_2 , and S_3 are fixed to the shank relative to the origin point P_0 . M_1 and M_2 define the TC axis; N_1 and N_2 correspond to the ST axis. We define r_1 and r_2 as the sagittal plane intersection with the TC and ST axes.

2.1.3. Size and Dimensions

We estimate the device dimensions from anthropometric proportions in [83] and use the segments proportions shown in Fig. 5.

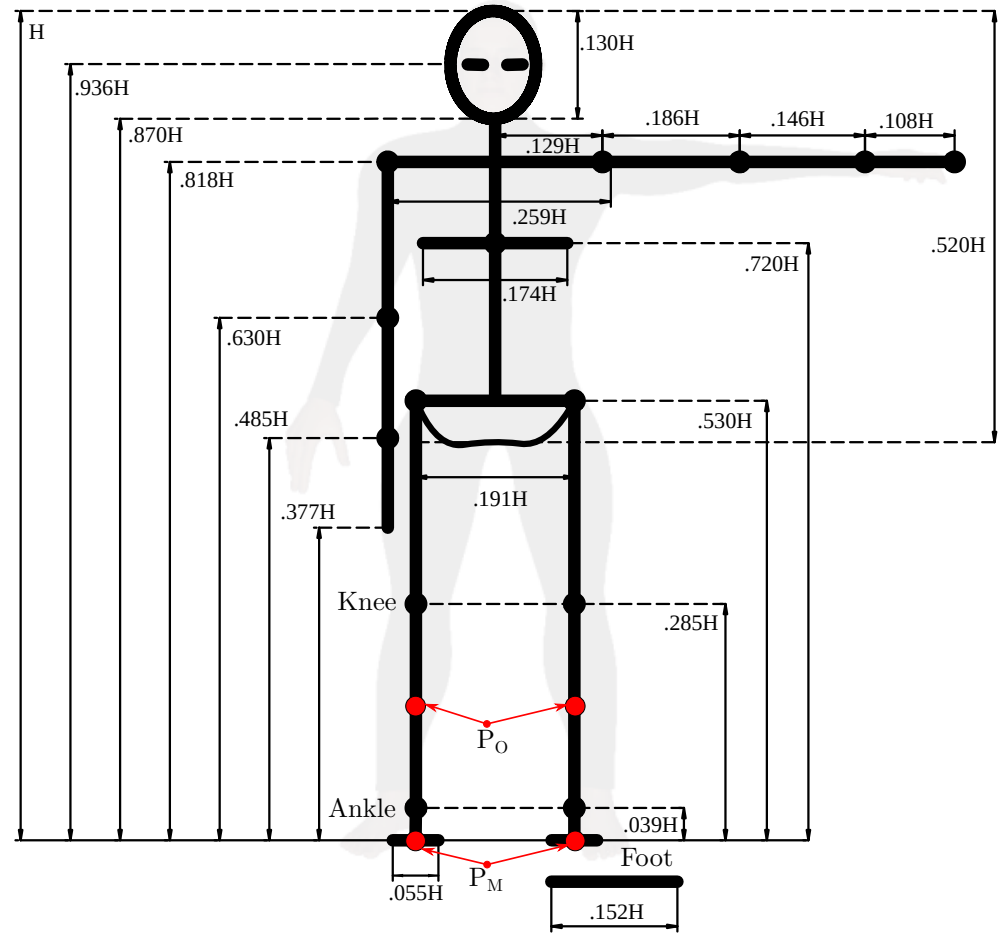


Figure 5. Body segment proportions [83].

We select the origin of coordinates between the knee and the ankle, d_m is the distance from P_M to P_O . This distance is proportional to the body's height H . To do so, we define d_m as follows:

$$d_m = \|P_O - P_M\| = \left[\frac{0.285 - 0.039}{2} + 0.039 \right] \cdot H = 0.162 \cdot H, \quad (1)$$

and, according to [85], the mean height H of an adult male is 175 cm; by substituting this value into the equation, the knee-ankle distance is 28.35 cm. The distance d_{p12} between points r_1 and r_2 about the TC and ST axes on the sagittal plane is:

$$d_{p12} = \|r_1 - r_2\| = Q, \quad (2)$$

the projection of the most medial point (MMP) on the sagittal plane is:

$$P_{MMP} = (x_{MMP}, 0, z_{MMP}), \quad (3)$$

and for the most lateral point is:

$$P_{MLP} = (x_{MLP}, 0, z_{MLP}). \quad (4)$$

The point M_{1p} is the projection of M_1 on the sagittal plane; we calculate it from the P and O values.

$$M_{1p} = (x_{MMP} - P, 0, z_{MMP} - O), \quad (5)$$

also, M_{2p} is M_2 ; we estimate the projection from L and K through:

$$M_{2p} = (x_{MLP} - L, 0, z_{MLP} - K). \quad (6)$$

Therefore, the segment $\overline{M_2M_1}$ has the sagittal projection $\overline{M_{2p}M_{1p}}$; it has the same proportional relation $R = W/w$ in respect to $\overline{M_{2p}r_1}$, then:

$$\frac{M_2 - M_1}{M_2 - r_1} = \frac{W}{w} = R, \quad (7)$$

solving for r_1 gives the following:

$$r_1 = M_2 - \frac{M_2 - M_1}{R}. \quad (8)$$

By knowing the distance Q projected in the sagittal plane and r_1 , the angle 41° we calculate r_2 from

$$r_2 = Q[\cos(41^\circ), 0, \sin(41^\circ)] + r_1, \quad (9)$$

The distance from the origin P_O to the plantar surface of the foot is d_m , we choose a circumscribed equilateral triangle with vertices A_0, B_0, C_0 as the platform base. The coordinates of A_0 are

$$A_0 = (r_p, 0, -d_m), \quad (10)$$

for B_0 are:

$$B_0 = (r_p \cos \frac{2}{3}\pi, r_p \sin \frac{2}{3}\pi, -d_m), \quad (11)$$

and for C_0 :

$$C_0 = (r_p \cos -\frac{2}{3}\pi, r_p \sin -\frac{2}{3}\pi, -d_m), \quad (12)$$

Where r_p is proportional to H , then:

$$r_p = \frac{2}{3} \cdot H. \quad (13)$$

In summary, we estimate P_0, r_1, r_2 ; and the platform's vertices A_0, B_0 , and C_0 . They aren't arbitrary selected, on the contrary, we employed anthropometry, statistics, and proportions.

2.1.4. Product of Exponentials Formula

In this section, we employ the PoE formula. We follow the intuitive concept that inter-bone contact surfaces determine HAJ movements. Therefore, we represent these movements as a Special Euclidean group $SE(3)$ in matrix form:

$$\mathfrak{g} = \begin{bmatrix} R & \hat{p}_T \\ 0_{1 \times 3} & 1 \end{bmatrix}, \quad (14)$$

where $R_{3 \times 3}$ is the rotation matrix and \hat{p}_T is the translation vector.

For the initial point A_0 :

$$\mathfrak{g}_A(0) = \begin{bmatrix} I_{3 \times 3} & \hat{A}_0 \\ 0_{1 \times 3} & 1 \end{bmatrix}, \quad (15)$$

for B_0 :

$$\mathfrak{g}_B(0) = \begin{bmatrix} I_{3 \times 3} & \hat{B}_0 \\ 0_{1 \times 3} & 1 \end{bmatrix}, \quad (16)$$

and for C_0

$$\mathbf{g}_C(0) = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \hat{\mathbf{C}}_0 \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (17)$$

We define $\hat{\omega}_1 = (\omega_{x1}, \omega_{y1}, \omega_{z1})$ as a unitary vector for the TC axis direction given by:

$$\hat{\omega}_1 = \frac{\mathbf{M}_2 - \mathbf{M}_1}{\|\mathbf{M}_2 - \mathbf{M}_1\|}, \quad (18)$$

and a directed vector $\hat{\mathbf{r}}_1$ from \mathbf{P}_O to \mathbf{r}_1 is:

$$\hat{\mathbf{r}}_1 = \mathbf{r}_1 - \mathbf{P}_O, \quad (19)$$

then, an orthogonal vector to $\hat{\mathbf{r}}_1$ and $\hat{\omega}_1$ is:

$$\hat{\nu}_{\theta_1 r_{2z}} = -\hat{\omega}_1 \times \hat{\mathbf{r}}_1, \quad (20)$$

together, $\hat{\omega}_1$ and $\hat{\nu}_{\theta_1 r_{2z}}$ compound the six-dimensional vector $\hat{\xi}_1$:

$$\hat{\xi}_1 = \begin{pmatrix} \hat{\nu}_1 \\ \hat{\omega}_1 \end{pmatrix}. \quad (21)$$

In the same way, there are correspondent vectors for the TC axis:

$$\hat{\omega}_2 = \frac{\mathbf{N}_2 - \mathbf{N}_1}{\|\mathbf{N}_2 - \mathbf{N}_1\|}, \quad (22)$$

$$\hat{\mathbf{r}}_2 = \mathbf{r}_2 - \mathbf{P}_O, \quad (23)$$

$$\hat{\nu}_2 = -\hat{\omega}_2 \times \hat{\mathbf{r}}_2, \quad (24)$$

and:

$$\hat{\xi}_2 = \begin{pmatrix} \hat{\nu}_2 \\ \hat{\omega}_2 \end{pmatrix}. \quad (25)$$

We compute \mathbf{R} for each joint $i = 1, 2$ from the Rodrigues' formula:

$$e^{(\Omega_i \theta_i)} = \mathbf{I}_{3 \times 3} + \Omega \sin \theta_i + \Omega^2 (1 - \cos \theta_i), \quad (26)$$

where Ω is the skew symmetric matrix:

$$\Omega = \begin{bmatrix} 0 & -\omega_{zi} & \omega_{yi} \\ \omega_{zi} & 0 & -\omega_{xi} \\ -\omega_{yi} & \omega_{xi} & 0 \end{bmatrix}. \quad (27)$$

The exponential formula is:

$$e^{\hat{\xi}_i \theta_i} = \begin{bmatrix} e^{\Omega_i \theta_i} & \tau_i \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}, \quad (28)$$

and, τ_i is translation vector:

$$\tau_i = (\mathbf{I}_{3 \times 3} - e^{\hat{\omega}_i \theta_i}) \hat{\omega}_i \times \hat{\nu} + \hat{\omega}_i \hat{\omega}_i^T \hat{\nu}_i \theta_i \quad (29)$$

Points A, B, and C have invariant relative positions, and there are two rotating joints; the PoE formula for A is:

$$\mathbf{g}_A = e^{\hat{\xi}_1 \theta_1} e^{\hat{\xi}_2 \theta_2} \mathbf{g}_A(0) = \begin{bmatrix} \mathbf{R} & \hat{\mathbf{p}}_A \\ \mathbf{0} & 1 \end{bmatrix}, \quad (30)$$

where \hat{p}_A is the instantaneous position vector of A, the PoE for B:

$$\mathbf{g}_B = e^{\hat{\xi}_1 \theta_1} e^{\hat{\xi}_2 \theta_2} \mathbf{g}_B(0) = \begin{bmatrix} \mathbf{R} & \hat{p}_B \\ 0 & 1 \end{bmatrix}, \quad (31)$$

and the PoE for C is:

$$\mathbf{g}_C = e^{\hat{\xi}_1 \theta_1} e^{\hat{\xi}_2 \theta_2} \mathbf{g}_C(0) = \begin{bmatrix} \mathbf{R} & \hat{p}_C \\ 0 & 1 \end{bmatrix}, \quad (32)$$

θ_1 is the TC rotation angle from the zero position, and θ_2 is the ST rotation from the zero position. For the sake of clarity, we show the section of the ankle with the vectors \hat{r}_1 , $\hat{\omega}_1$, \hat{v}_1 and \hat{r}_2 ; also the points A, B, C and P_O in Figure 6.

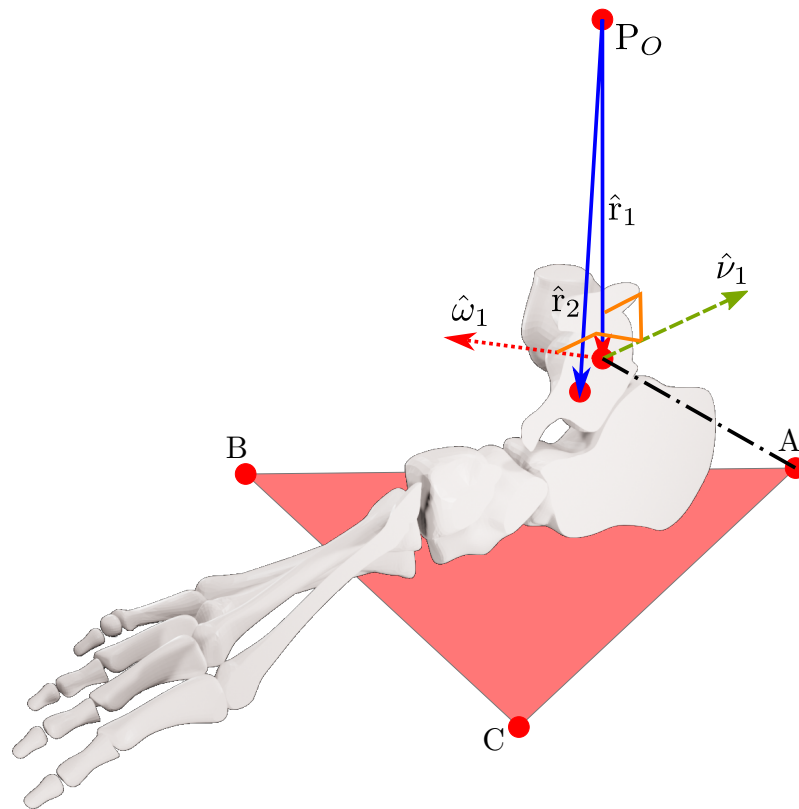


Figure 6. Vectors and points on the sagittal plane.

2.1.5. Forward Kinematics

In this subsection, we show the simulation of the movements of the ankle by using the measurements and the PoE. The code is in SageMath Computer Algebraic System (CAS), it let us manage symbolic notation, and interactive plotting in a Jupyter notebook, the listings are in appendix B. We show the SageMath simulation complete code in 1. In B2 is the code in Asymptote for visualization and printing.

The simulation plot for the platform's central point is in Figure 7(a). We show the points P_O , A_0 , B_0 , C_0 , r_1 , r_2 , and the surfaces representing each group of movements. The forward kinematics with $\theta_{1range} = \theta_{2range} = [-15, 15]$ and $\theta_1 = \theta_2 = 10^\circ$ is in Figure 7(b). For $\theta_{1range} = \theta_{2range} = [-10, 10]$ and $\theta_1 = \theta_2 = 5^\circ$ is in Figure 7(c).

Such representation lets us compute the ankle joint ROM in all directions. Groups of A, B, C, and PM movements are smooth surfaces or geometric manifolds. They have two DOF, with a limited domain due to the axes ROM.

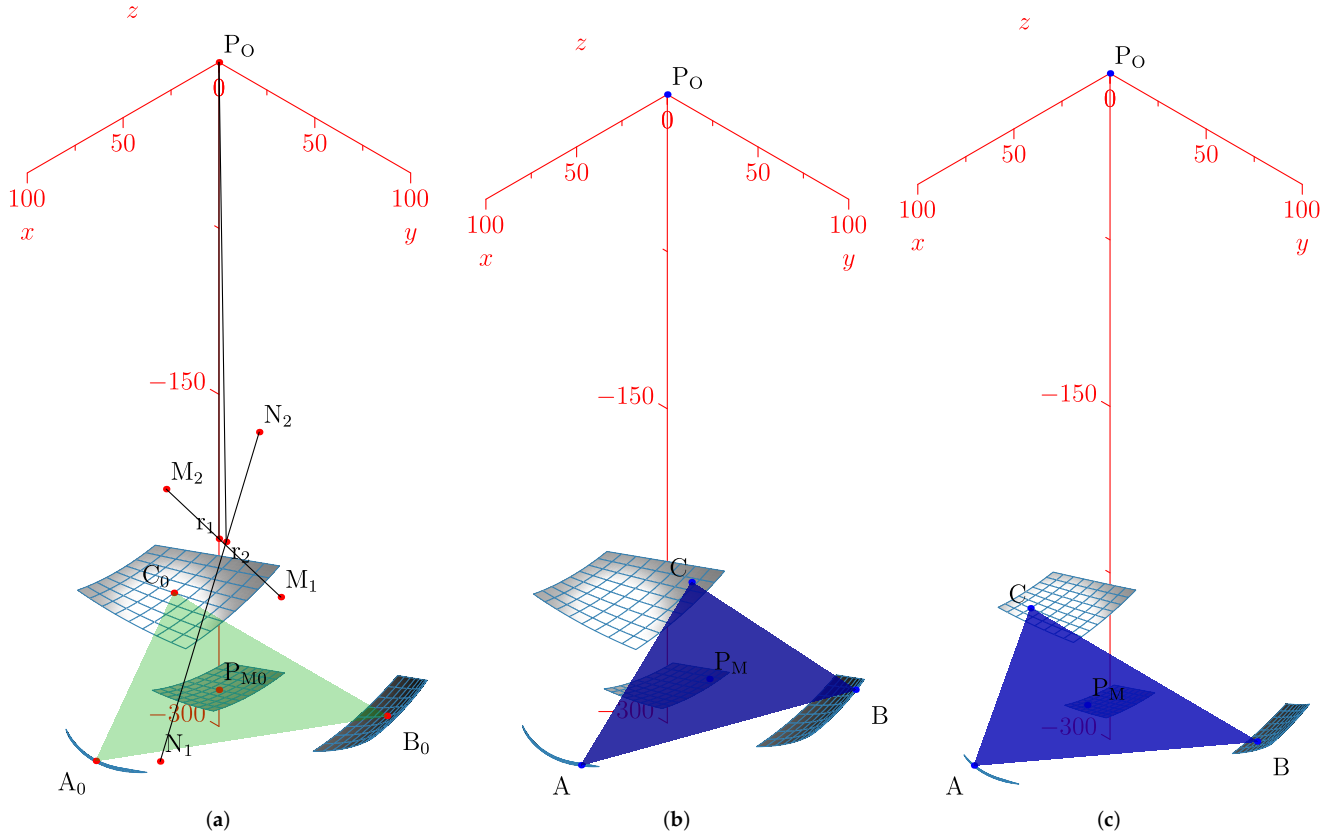


Figure 7. Forward kinematics for (a) initial position and (b) $\theta_{1range} = \theta_{2range} = [-15, 15], \theta_1 = \theta_2 = 10^\circ$ and (c) $\theta_{1range} = \theta_{2range} = [-10, 10], \theta_1 = \theta_2 = 5^\circ$

2.1.6. Geometric Design and Trilateration Method

Based on the forward kinematics we show a geometric design in the Figure 8(a) the platform center, and in Figure 8(b) are the vertices.

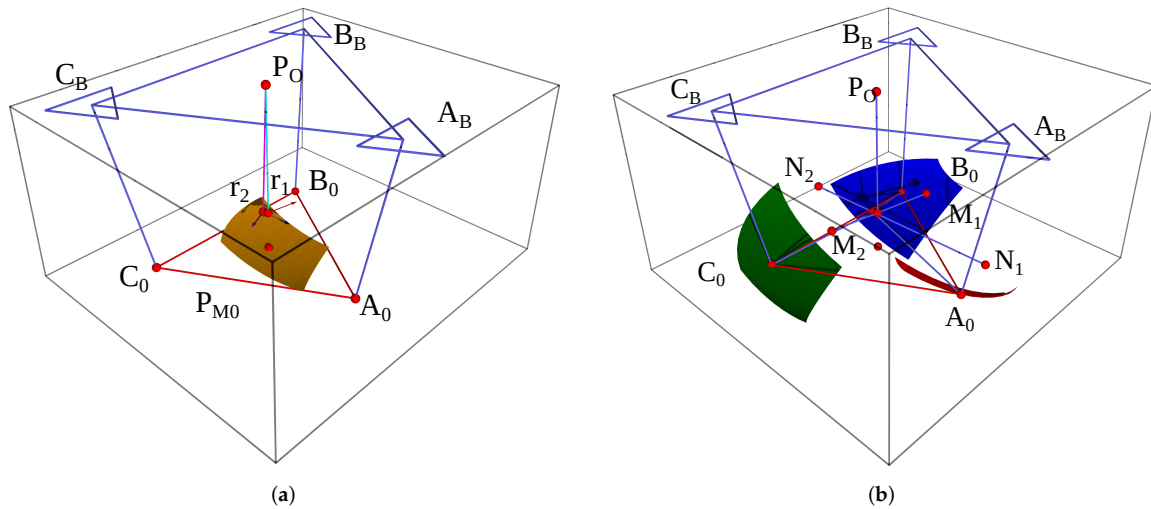


Figure 8. Geometric design: (a) is the platform center, base and r_1, r_2 and (b) platform vertices with talocrural and subtalar axis.

By considering the distances between the origin and the vertices, we estimate the DWS maximal length in every module.

$$l_{max} = \max[\|p_A(\theta_1, \theta_2) - A\| + r_m] \quad (33)$$

Here, l_{max} is the maximal possible length from the triangular inequality, p_A is the positions group in \mathfrak{g}_A , r_m is the module's radius, and A_B is the base point.

The main design requirement is the localization of three points attached to the foot. We estimate the actual position employing an array of DWS in a tetrahedral structure to find the apex, which is a platform vertex. In Figure 9 we show the design structure.

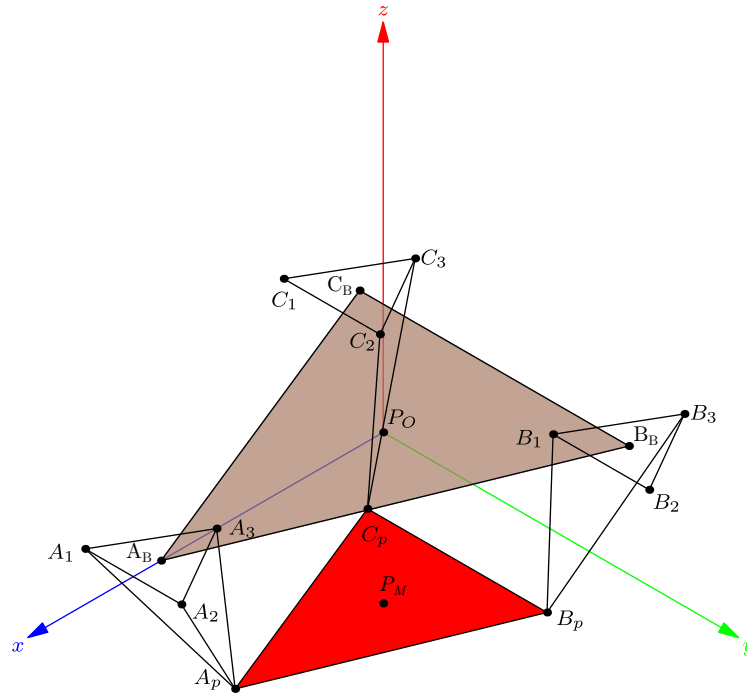


Figure 9. Geometric design of the DWS arrays.

P_O and P_M are the base and platform reference frames. The platform has known dimensions and the number of sensors is seven. First, we compute A_p from three distances: $l_{A1} = \|A_p - A_1\|$, $l_{A2} = \|A_p - A_2\|$ and $l_{A3} = \|A_p - A_3\|$. Then, we compute B_p and B_p apexes after A_p employing two DWS.

2.2. Finding the Apex in Tetrahedron A

In this section, we compute the tetrahedron T_A with base $\triangle_A = [A_1, A_2, A_3]$ and apex A_p . Figure 10 shows the method we use.

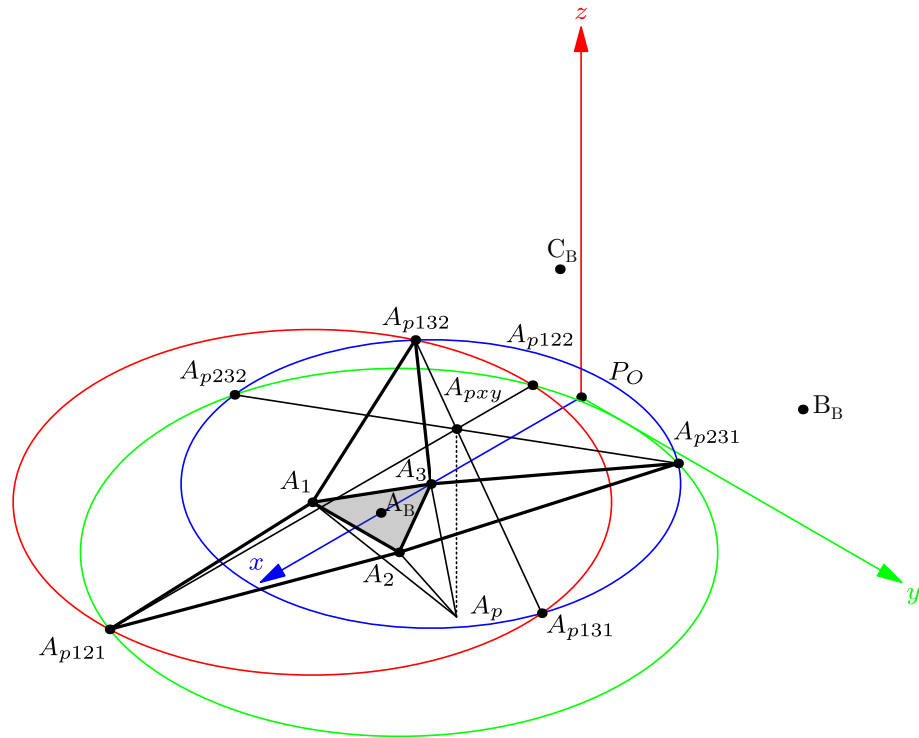


Figure 10. Finding the apex A_p .

In Figure 10 we see that triangles $\triangle_{132} = [A_1, A_3, A_{p132}]$ and $\triangle_{231} = [A_2, A_3, A_{p231}]$ are two sides of the tetrahedron T_A developed on the base plane.

We compute the A_{p132} and A_{p231} orthogonal projection on each adjacent side of the module base triangle $\triangle[A_1, A_2, A_3]$ by tracing a circle centered on A_1 with radius $\|A_p - A_2\|$ and the circle centered on A_3 with radius $\|A_p - A_3\|$; resulting in A_{p132} and A_{p131} intersection points. In addition, the circle centered on A_2 with radius $\|A_p - A_2\|$ intersects the circle centered in A_3 at points A_{p231} and A_{p232} . The segment from A_{p132} to A_{p131} intersects the so defined by points A_{p231} and A_{p232} at A_{pxy} . In the case of tetrahedron T_A , we determine $A_{pxy} = (A_{px}, A_{py}, 0)$ as A_p projection on the base plane. It is easy to realize that the height of T_A is the absolute value of the A_{pz} coordinate. Then, we can find the distance from A_{pxy} to A_3 as a triangle $\triangle[A_{pxy}, A_3, A_p]$ side; the other is A_{pz} , and the hypotenuse is the distance $l_{A3} = \|A_p - A_3\|$, then, A_{pz} is:

$$A_{pz} = \sqrt{l_{A3}^2 - (A_{pxy} - A_3)^2} \quad (34)$$

2.3. Tetrahedrons B and C Apexes

In this subsection, we show that, by knowing A_p , the point B_p needs two sensors to be found. To determine the result of the tetrahedron T_B , we consider the Figure 11(a) the base of a triangle $\triangle[B_1, B_3, A_p]$.

We compute the angle α from the XY plane to a normal vector \hat{n}_{ApB} :

$$\hat{n}_{ApB} = \frac{(B_3 - A_p) \times (B_1 - A_p)}{\|(B_3 - A_p) \times (B_1 - A_p)\|}, \quad (35)$$

and, the angle α is:

$$\alpha = \arccos(\hat{n}_{ApB} \cdot \hat{n}_z), \quad (36)$$

where \hat{n}_z is the unitary vector normal to the XY plane.

The tetrahedron sides are the lengths $l_{B1} = \|B_p - B_1\|$, $l_{B3} = \|B_p - B_3\|$ and $d_{ApBp} = \|B_p - A_p\|$. The rotation axis is in the direction $B_1 - B_3$. The B_{pr} is B_{ps} rotated α in angle about this axis. In Fig 11(b), we show how to find the B_{pr} apex, similarly to

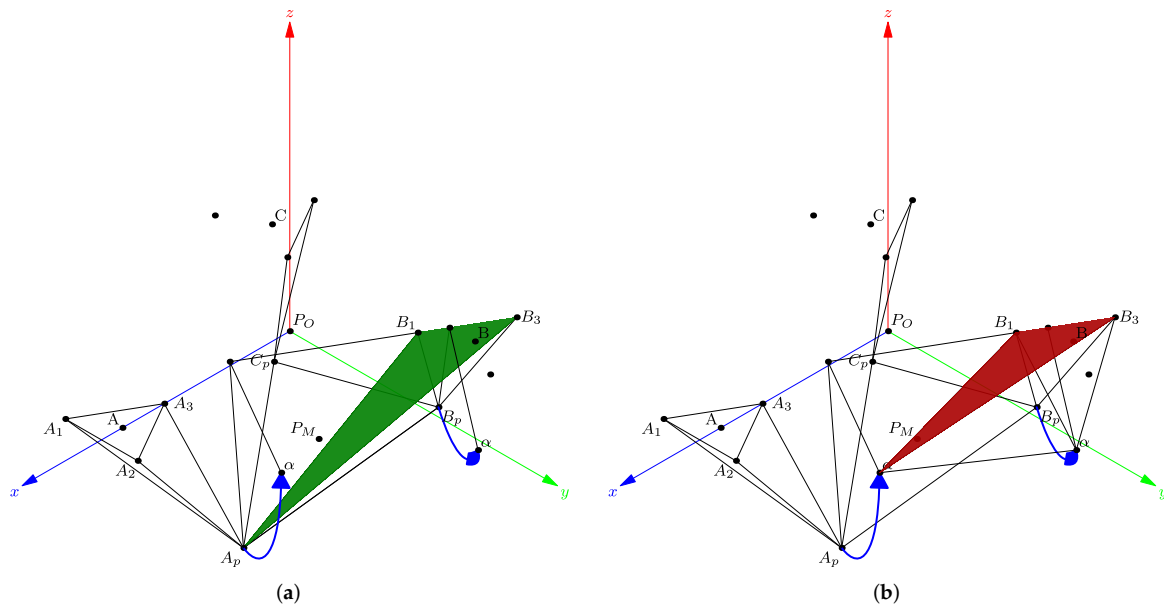


Figure 11. Rotation of α angle about the axis $B_1 - B_3$: (a) original tetrahedron, T_B (b) rotated tetrahedron.

that of a tetrahedron T_A . Finally, when B_{pr} is found, the contrary rotation about the axis $B_1 - B_3$ gives the B_{ps} .

There are two possible apex values: B_{ps1} over, and B_{ps2} below of the XY plane. We show the B_{pr} apex below the XY plane in Figure 12.

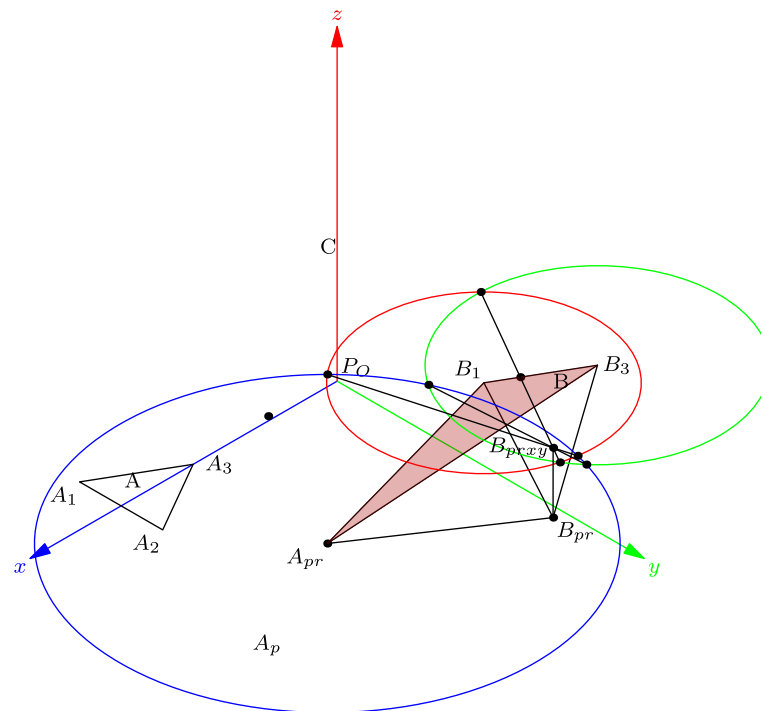


Figure 12. Finding the apex B_{pr} .

We use the same method to solve the T_C apex. For the correct apex selection, the condition when the side of the platform distance d_{CpBp} is:

$$d_{CpBp} = \|B_{ps} - C_{ps}\|. \quad (37)$$

2.4. Computer Aided Designs

In this section, we design DWS to measure the lengths of the tetrahedrons sides; they are arranged as structural parts. Their maximal length estimation is from the forward kinematics simulation. We design the shank attachment from the dimensions, proportions and statistical data.

2.4.1. Draw-wire Sensor

We use flat springs. They aren't exposed to a high load against gravity, and are in two or three concurrent groups. In Figure 13 we depict the design, composed of three 3D printed parts, potentiometer, flat spring, bolts, and nuts.

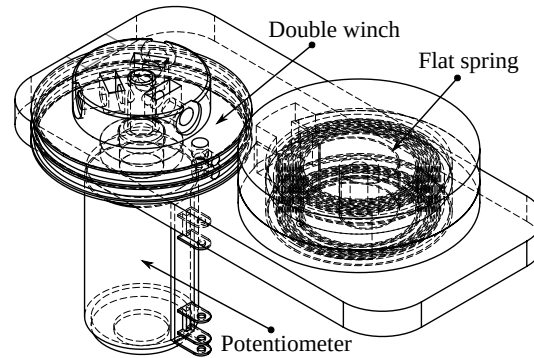


Figure 13. Draw-wire sensor design.

A two-coil winch drives the potentiometer; a flat spring retracts a wire attached to the winch. When we pull the wire, the spring retracts it. The value of each turn is from the nominal value of the potentiometer, $R_n = 2.2k\Omega$, divided into ten turns, that is 220Ω per turn. The diameter is $D = 3.8\text{cm}$, the spring could be compressed in four turns. The maximal length is as follows:

$$l_{max} = 4 \cdot D \cdot \pi \quad (38)$$

Which is 47.75 cm approximately, this value is greater than l_{max} for all groups of movements.

2.4.2. Mechanical Parts

The attachment on the calf has a size according to the simulation. We use the mesh model of a leg to guide the shape of the calf support, as in Figure 14(a). Also, we scale and divide his structure into seven parts for 3D printing. An aluminum tube is the support structure as in 14(b), and a neoprene band attaches the shank to the support with Velcro fabric.

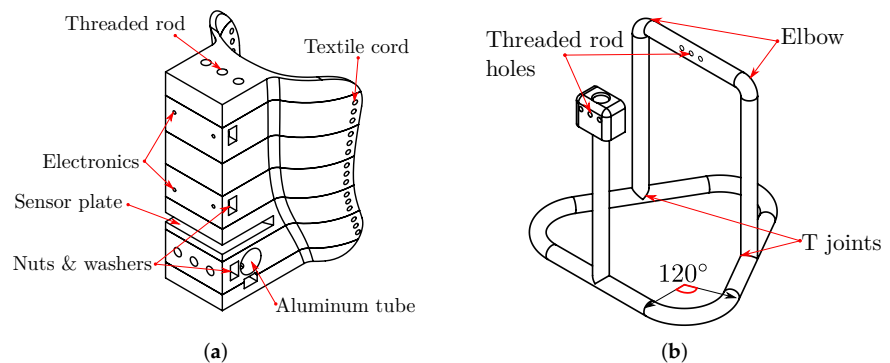


Figure 14. Mechanical attachment: (a) calf support, and (b) aluminum tube structure.

All the DWS modules are in a plate, the A module has three DWS, B and C modules has two DWS as in Figure 15(a). The design of the foot attachment is from standard measurements to adjust the foot's length and width, as in Figure 15(b).

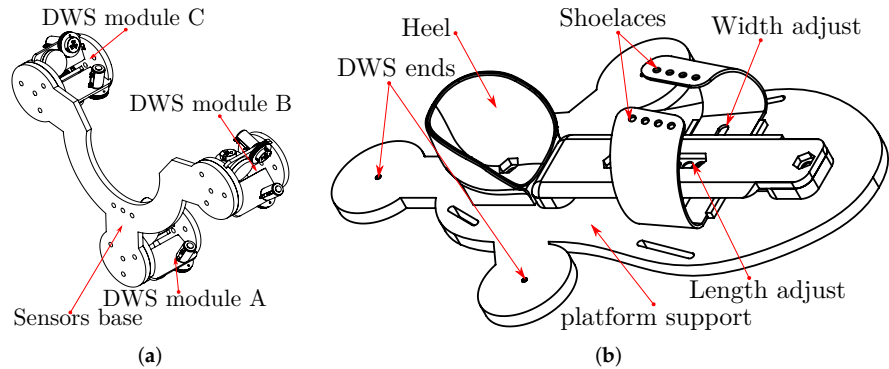


Figure 15. Base and platform: (a) DWS modules support (b) platform with foot's size adjustment.

2.4.3. Electronics

Two operational amplifiers in instrumentation configuration are the base block of acquisition system, as the Figure 16 shows. We employ the KiCad software for the circuit design.

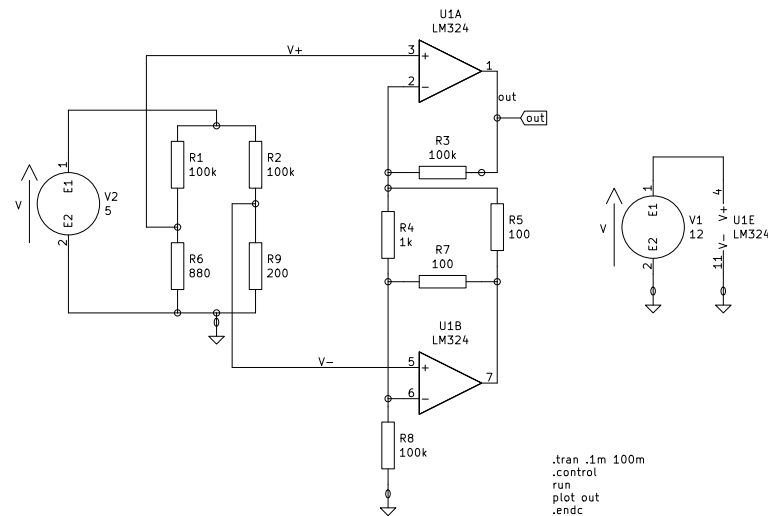


Figure 16. Two Op. Amp. instrumentation amplifier.

The voltage gain in the instrumentation amplifier is:

$$A_v = \frac{v_o}{v_i} = \left[1 + \frac{R_2}{R_1} + \frac{2R_2}{R_1} \right], \quad (39)$$

By selecting $R_2 = 100k\Omega$, $R_1 = 1k\Omega$, and $R_G = 5k\Omega$, the voltage gain is 141. With 34mV as voltage input, we get:

$$v_o = A_v \cdot v_i = 4.794V \quad (40)$$

The final acquisition circuit has seven instrumentation amplifiers, with bias and gain trimmers for calibration. We design the printed board circuit as an TMArduino Mega 2560 Shield. And assemble the components to the board by throw-hole soldering. We feed the circuits with a power system with two 18650 Li-Ion batteries in series, a backup pack, a Battery Management System (BMS); a 5V buck and a 12V boost converters. The

Figure 17 shows the schematics. Finally, we add connectors for the MPU's, OLED and Bluetooth module.

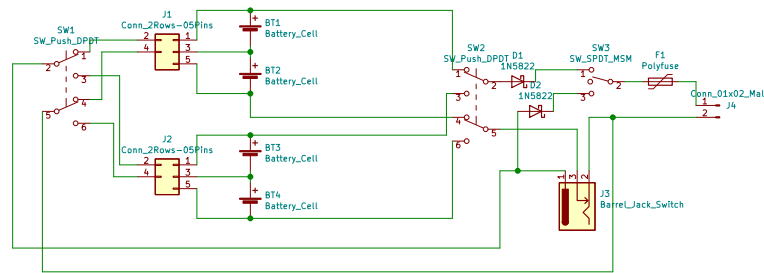


Figure 17. Power system with backup, BMS, boost and buck converters.

2.4.4. Electronics casing.

We export the KiCad printed circuit design to FreeCAD StepUp to design the case containing all the components, focusing on a compact configuration design. The two main electronic components are the Arduino Mega 2560 and the Orange Pi One single board computer. We place the components such as the Dual Pole Dual Throw (DPDT) toggle switches symmetrically on the box sides. The Figure 18 shows the main sides and the final assembly of the electronics case.

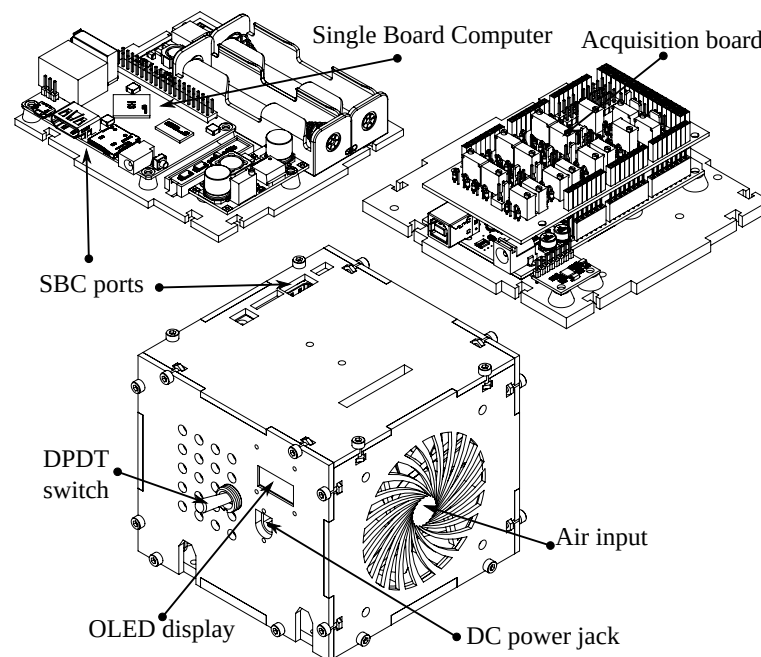


Figure 18. Modular electronics casing.

Each box has attached components to optimize the space. We test every component, and then install the support structure.

2.4.5. Final mechanical assembly

The prototype consist of 45 3D printed parts, the union of main components is by an 8 mm steel threaded rod. The sub-assemblies uses M3 bolts and nuts. Figure 19 shows the assembly CAD.

2.5. Calibration and Validation Software

Calibration is with the Arduino board connected to the PC, running a calibration program in Processing. The basic program requests the IMU readings from the accelerometer and gyroscope data, and capture readings from the ADC inputs. The raw

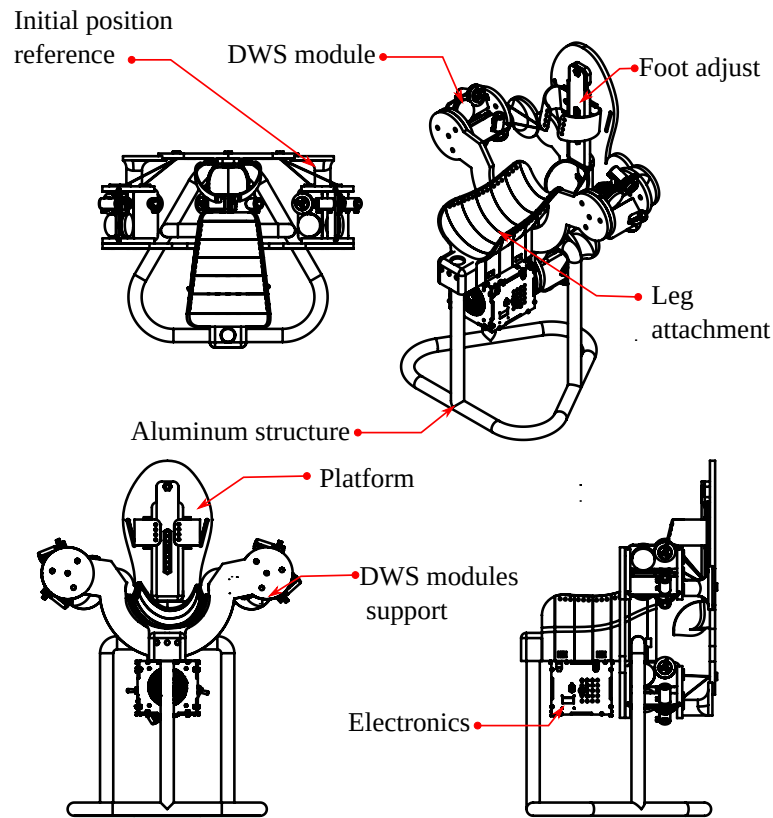


Figure 19. Complete prototype.

data are signed integer values 2 bytes wide, the two 1-byte registers converted to 2-byte integers. An exponentially weighted moving average (EWMA) program, filter the raw signals and send via a serial port to the PC. The lengths compute are from the initial values plus the scaled sensor inputs with:

$$l_{iMj} = d_{iMj} + \frac{m_{iMj}}{s_{iMj}}, \quad (41)$$

here, l_{iMj} is the length in cm from the i wire to the j module, d_{iMj} is the initial distance, m_{iMj} is the measured digital value, and s_{iM} is the scale factor in digital units per cm.

We present a rendered image with a scaled 175 cm model in the Figure 20.

2.5.1. Procedure for Finding the Platform Positions

The shank and the foot must be fixed to the base and platform. Then we mark the MMP and the MLP. To do so, we design a detachable reference point from the module A.

Initially, we capture and register the MMP and MLP. In the Figure 21 is the detailed view.

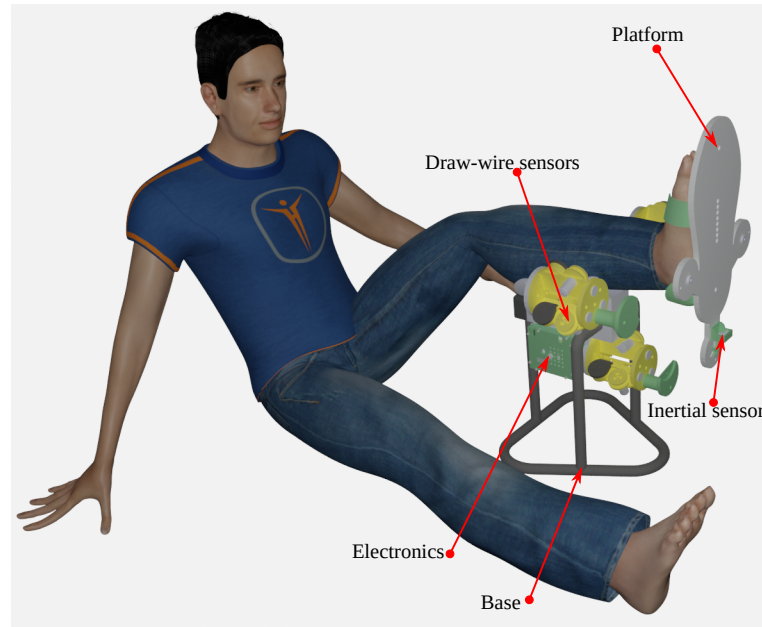


Figure 20. Rendered image with a 175 cm height patient.

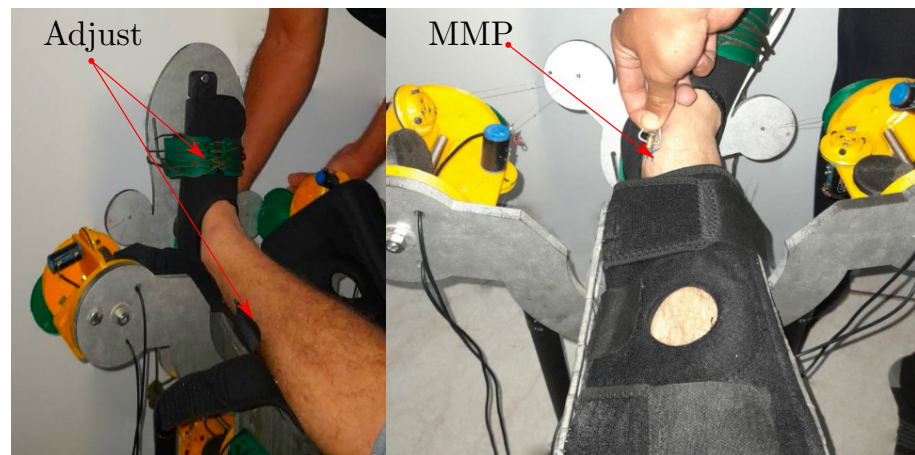


Figure 21. Assembled Structure.

We compute the platform position from the seven sensor lengths. The main steps for capture data series is:

1. Capture the initial position at horizontal relative position from $dr = IMU2 - IMU1$ readings;
2. Compute jerk $jrk = |dr_i - dr_{i-1}|$;
3. Move the foot continuously until jerk cross zero again.

First, we start capturing by activating a button in the computer software. Every time, we compute the absolute difference from IMU2 to the IMU1 readings. If the differences are constants, then, there is not platform-base relative movement. We compute the jerk by relative acceleration finite differentiation. The data captured finish when the jerk cross zero coordinates. It means that detection of jerk change, activates the capture of the IMU's data.

The symbolic equations to find A_p , B_p and C_p from the captured data, were found by the SageMath CAS. By using the prototype dimensions, and the sensor lengths, we compute the platform's position and orientation. In this case, the origin is from the initial DWS lengths l_{Mi0} , where M is the module A, B or C; and i is the sensor number $i=1,2,3$.

After MLP and MMP registering, we attach the apex of the module A to the platform. And define the sagittal plane perpendicular to the ABC base plane and intersecting the A point. By implementing the trilateration method before mentioned, we compute the points A_0 , B_0 , and C_0 .

The Figure 22(a) illustrate the points positions with the device in the initial portable configuration. The apexes' computation are in Figures 22(b), 22(c) and 22(d).

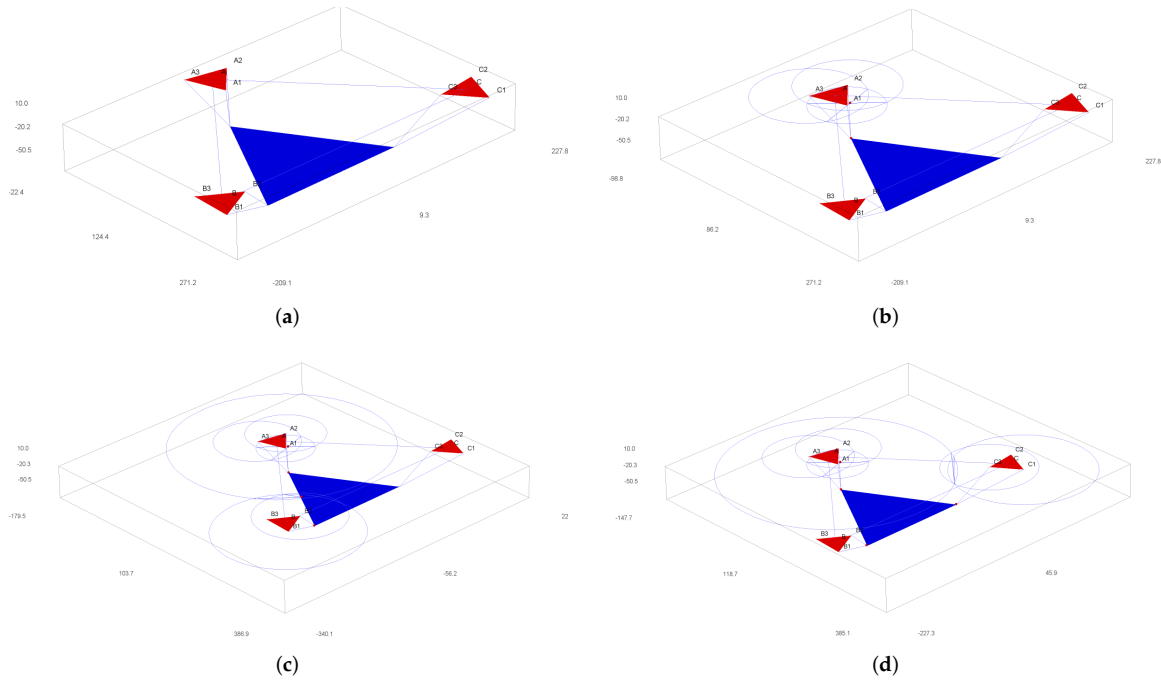


Figure 22. Computed positions from sensor lengths at portable configuration: (a) the rest position, (b) apex A, (c) apex B and (d) apex C.

2.5.2. Computing the axis position and direction

To compute the initial TC axis position, we employ dorsal and plantar movements, because the TC axis is most dominant in such movements. An initial approximation, is from the MMP and MLP points. The sagittal plane intersection with the $\overline{M_1M_2}$ segment is r_1 . We estimate the initial TC axis approximation with:

$$M_{1_0} = MLP - [L, 0, K], \quad (42)$$

and

$$M_{2_0} = MMP - [-P, 0, O]. \quad (43)$$

from these values, we solve for r_1 from the intersection of the line:

$$\mathbf{P} - M1 = \rho \frac{(M_2 - M_1)}{\|M_2 - M_1\|} \quad (44)$$

and the plane $y = 0$.

The ST axis sagittal intersection r_2 initial point is:

$$r_2 = r_1 + \frac{\sqrt{2}}{2} [Q, 0, Q] \quad (45)$$

This is a statistical initial estimation, we use for comparison with the orientation captured by doing circle fitting to A, B and C trajectories. First, we found the TC axis orientation ω_1 by registering several trajectories. For each trajectory, we have a list of data points $P = [x, y, z]$, which pertain to a plane:

$$ax + by + cz + d = 0, \quad (46)$$

or in all n data is in a $n \times 3$ matrix, solving for z we have

$$\begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_{n-1} & y_{n-1} & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = - \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_{n-1} \end{bmatrix} \quad (47)$$

which has the form:

$$Ax = B \quad (48)$$

there are more equations, the pseudo inverse is $A^+ = (A^T A)^{-1} A^T$, and the normal vector is:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = (A^T A)^{-1} A^T B \quad (49)$$

By using the Rodrigues' formula we found the rotated points to the XY plane by the product $v = n \times \hat{k}$, with $\hat{k} = [0, 0, 1]^T$

$$P_r = P \cos(\theta) + (v \times P) \sin(\theta) + v(v \cdot P)(1 - \cos\theta). \quad (50)$$

where $\theta = \arccos\left(\frac{n \cdot \hat{k}}{\|n\|}\right)$.

The circle in XY plane can be rearranged by

$$\begin{aligned} (x - x_c)^2 + (y - y_c)^2 &= r^2 \\ (2x_c)x + (2y_c)y + (r^2 - x_c^2 - y_c^2) &= x^2 + y^2 \\ c_0x + c_1y + c_2 &= x^2 + y^2 \end{aligned} \quad (51)$$

where $c = [c_0, c_1, c_2]^T$ with $c_0 = 2x_c$, $c_1 = 2y_c$, and $c_2 = r^2 - x_c^2 - y_c^2$.

By taking the rotated points, P_r we have a linear system

$$\begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_{n-1} & y_{n-1} & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} x_0^2 + y_0^2 \\ x_1^2 + y_1^2 \\ \vdots \\ x_{n-1}^2 + y_{n-1}^2 \end{bmatrix}. \quad (52)$$

has the form

$$Ac = b \quad (53)$$

Again, we have more equations than unknowns, then, we search for the c values that minimize the squared difference $\|b - Ac\|^2$.

$$\arg \min_{c \in \mathbb{R}^3} \|b - Ac\|^2. \quad (54)$$

We found the center point $C_p = [x_c, y_c]$ and radius r by solving:

$$\begin{aligned} 2x_c &= c_0 \\ 2y_c &= c_1 \\ r^2 - x_c^2 - y_c^2 &= c_2 \end{aligned} \quad (55)$$

Finally, we apply a rotation to the center in respect to the original plane. This point pertains to the TC axis, for each trajectory A, B, C we get three planes, and three centers, the TC line direction is parallel to the planes normal vectors. The information is complete by knowing the plane orientation. The estimation of the ST axis is similar, but employing

trajectories from inversion movements. This is a basic estimation, by doing optimization to the product of exponential formula, we enhance the accuracy of the axis position estimation.

2.5.3. Ankle Joint movements as a Manifold

In this subsection we explain how the centers r_1, r_2 and directions ω_1, ω_2 defines a manifold representing the HAJ movements. The circle center points calculated pertain to the TC and ST axis, they are the initial data to fit the product of exponential formula. In the Figure 23(a) we show the complete platform's center point manifold. It is topologically similar to a torus.

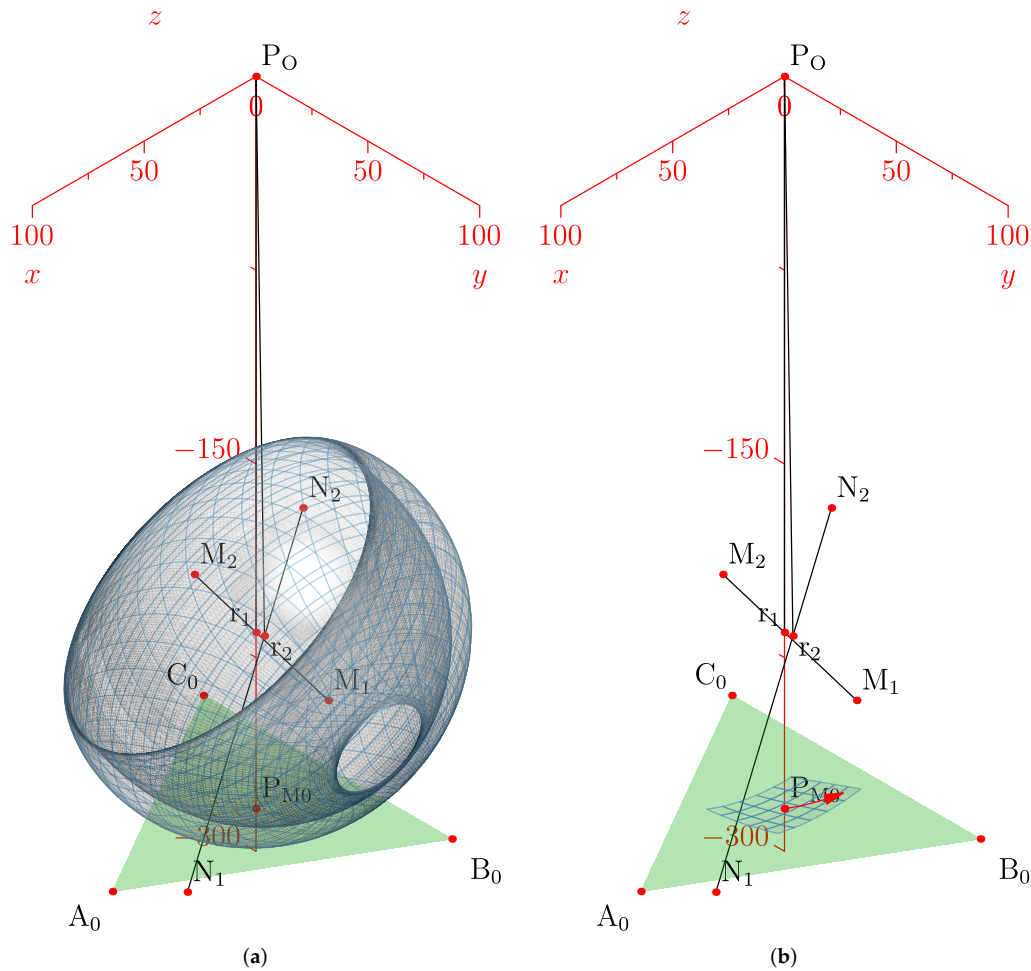


Figure 23. Simulation of the platform central point with variations in the mean statistical values: (a) Platform's center point manifold, (b) Manifold chart and a geodesic.

A manifold chart represents the range of motion limits, we show an example of geodesic as a trajectory on the manifold in the Figure 23(b); it explains how to map ankle coordinates, and a straight trajectory with initial velocity and no external forces action. We have the data necessary for the line intersection with the sagittal plane, the center points and the direction gives a line:

$$\hat{p}_l = \hat{l}_0 + \hat{l}d, \quad (56)$$

where \hat{p} is the parametric line point, \hat{l}_0 is a known point in the line, and $d \in \mathfrak{R}$, replacing the parametric equation in the plane equation:

$$(\hat{p} - \hat{p}_0) \cdot (\hat{n}_p) = 0, \quad (57)$$

where \hat{p}_0 is a known point in the plane, and \hat{n}_p is the plane's normal vector. solving for d , gives:

$$d = \frac{(\hat{p}_0 - \hat{l}_0) \cdot \hat{n}}{\hat{l} \cdot \hat{n}}, \quad (58)$$

and replacing in the TC axis line equation:

$$r_1 = c_1 + \omega_1 d \quad (59)$$

where r_1 is the TC axis intersection with the sagittal plane. The point c_1 is the center, and the axis direction ω_1 , both were found by circle fitting. Also, packing in six dimensional Plücker line coordinates, we have:

$$\hat{m}_1 = r_1 \times \omega_1, \quad (60)$$

and the l_1 six dimensional vector is:

$$l_1 = [\omega_{1_x} : \omega_{1_y} : \omega_{1_z} : m_{1_x} : m_{1_y} : m_{1_z}] \quad (61)$$

We include those data for the PoE formula simulation and the manifold representation.

3. Results

We organize this section as follows: first we show the simulation, second the final prototype, third, the trilateration, the axis orientation, and finally an ankle manifold representation.

3.1. Simulation Results

In this subsection, we put different values from the table 1 to estimate the workspace and range of motion. First, we show the variation of mean values results, and second the platform position simulation by changing range of movement and angles.

3.1.1. Changing statistical mean values

The Figure 24(a) shows the complete manifold, taking to account intervals $\theta_1, \theta_2 \in [-180^\circ, 180^\circ]$. It also shows the platform's initial position, the TC axis reference, the initial ST reference, the initial orientation and a parametric trajectory with equal angle rate variation. In the figure 24(b), is the attaching point A simulation, the Figures 24(c) and 24(d) depicts the simulations of B and C, respectively.

In the Figure 25(a) we show the platform's central point simulation with variations in 10% below the statistical mean values; Figure 25(b) we show the simulation changing 10% over the statistical mean values; Figure 26(a) is the attaching point A simulation adding the 10% of the mean values, and 26(b) is subtracting 10% of the mean values. Figures 27(a) and 27(b) are the results for the platform attaching point B.

Finally, by changing the range of angles and values, two examples capture of the simulation are in Figures 29(a) and 29(b).

We show the results for the attaching point C in Figures 28(a) and 28(b)

3.2. Final Prototype

In this section, we describe the results of the TM design, which are the assembled device and calibration. We try several designs and finally the CAD model is in [86]. First, we show images of the connected electronics parts. Second, we assemble the structure and perform calibration. Third, the device calibration results and probe the device in a healthy patient to validate the prototype adaptability. We print the structural parts using ABS, the draw-wire sensor using PLA; PETG is in the supports and the case.

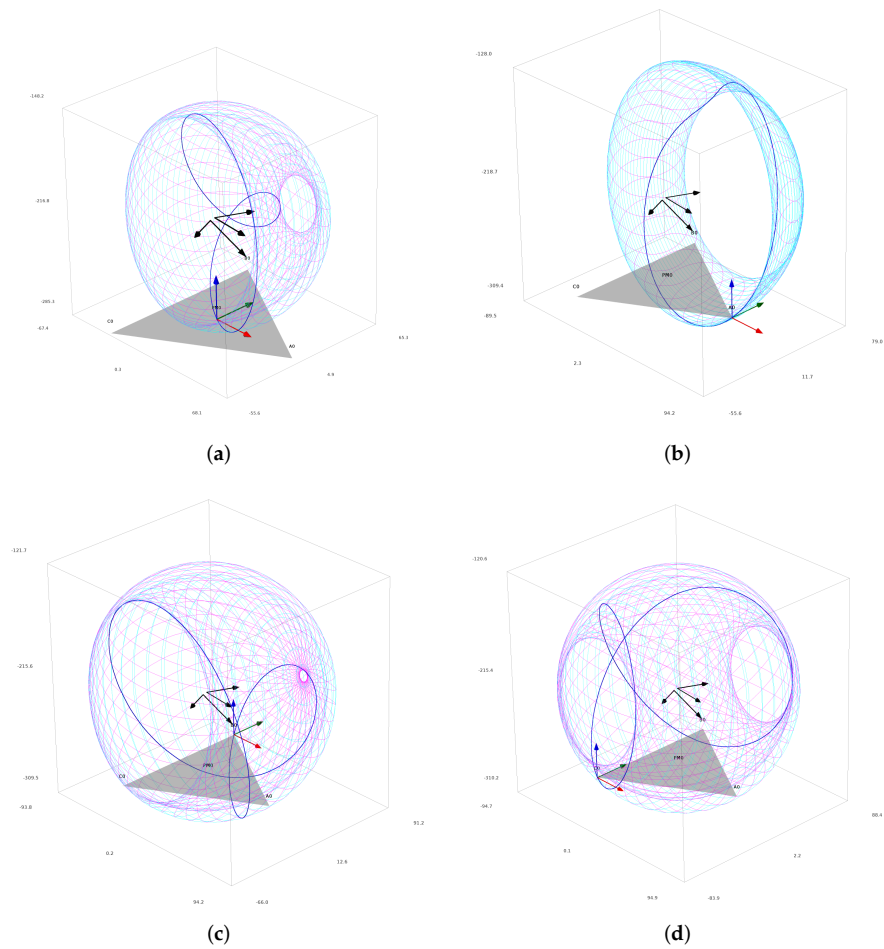


Figure 24. Simulation of all points: (a) platform's central point (b) attachment a, (c) attachment b, and (d) attachment c.

3.2.1. Printed and Connected electronics

We place the electronics in each side. In Figure 30, then connect the box sides; and charge the batteries.

3.2.2. Printed and Assembled Structure

We assemble all structural components carefully, and putting together with stainless-steel threaded rods; then we put the draw-wire sensors, the acquisition board, connections, and final structure for calibration. The Figure 31 shows the assembly.

3.2.3. Calibration Results

We calibrate the system by using a personal computer. The resulting calibration, and measures of the lengths are in Figure 32. The lecture, is at the initial position, then we compare with the SolidWorks model measurements.

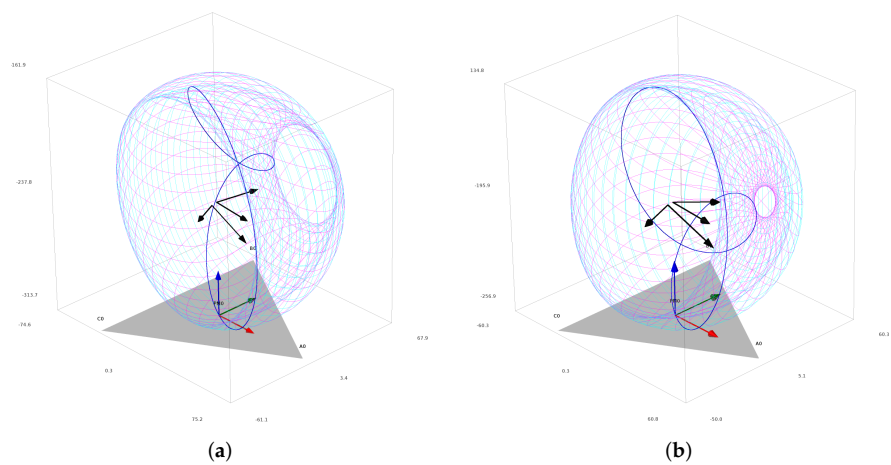


Figure 25. Simulation of the platform central point with variations in the mean statistical values: (a) 10% below, and (b) 10% over.

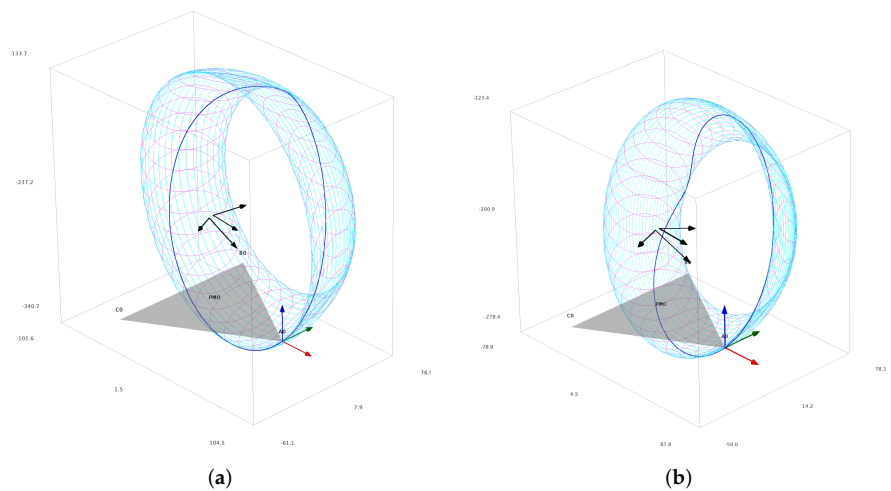


Figure 26. Simulation of the platform's attaching point A: (a) mean values plus 10% (b) mean values minus 10%.

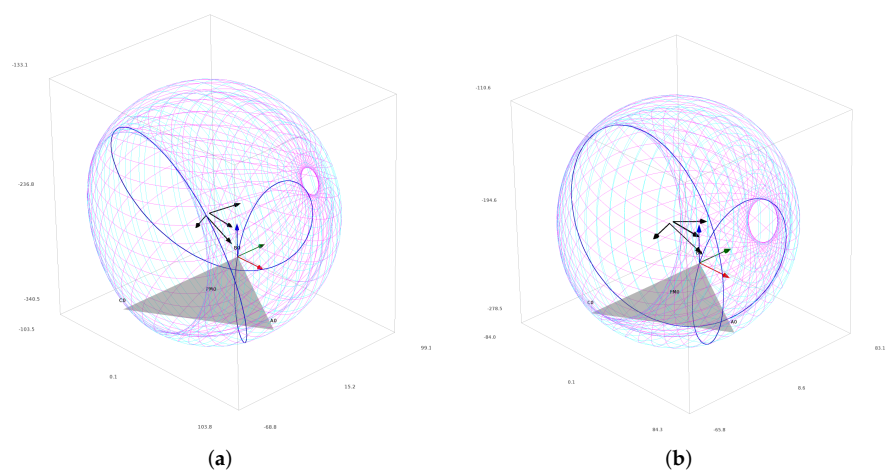


Figure 27. Attaching point B simulation: (a) adding 10% to the statistic mean values, (b) subtracting 10%.

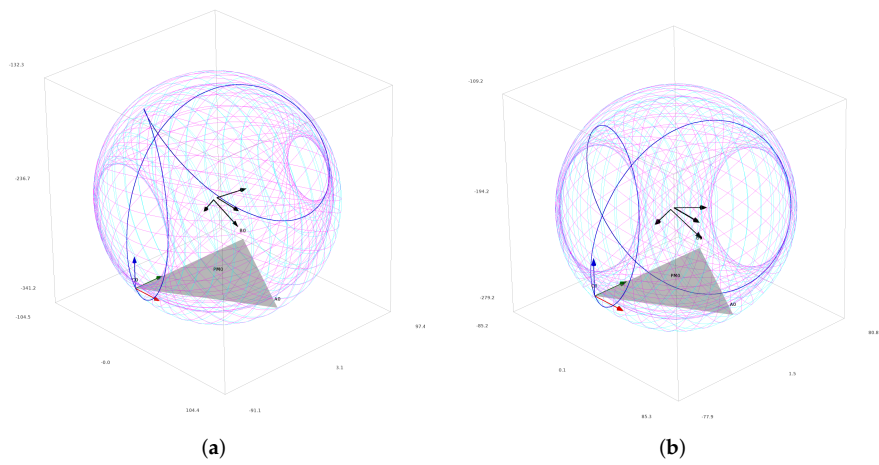


Figure 28. Simulation results for C: (a) mean values plus 10% (b) mean values minus 10%.

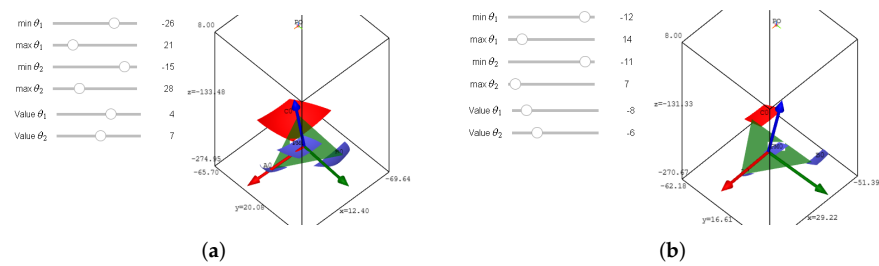


Figure 29. Position and orientation examples: (a) for $\theta_1 \in [-26, 21]$, $\theta_2 \in [-15, 28]$ in position $[\theta_1, \theta_2] = [4, 7]$ (b) for $\theta_1 \in [-12, 14]$, $\theta_2 \in [-11, 7]$ in position $[\theta_1, \theta_2] = [-8, -6]$

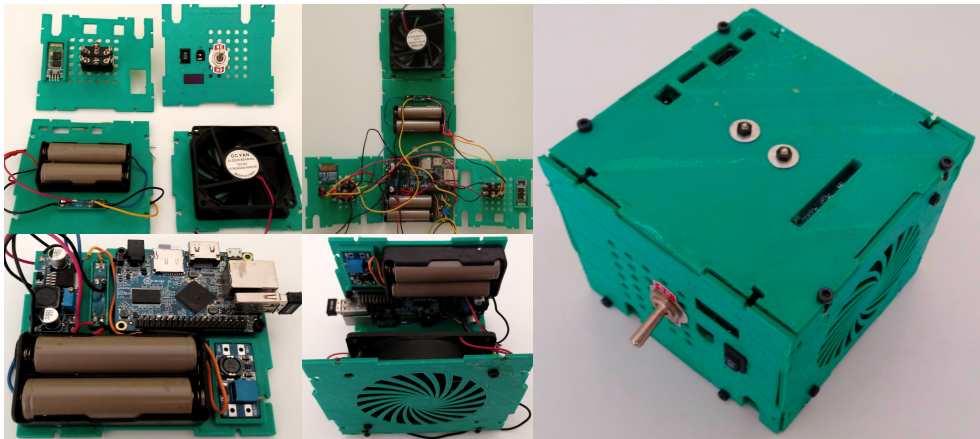


Figure 30. Connections and electronics.

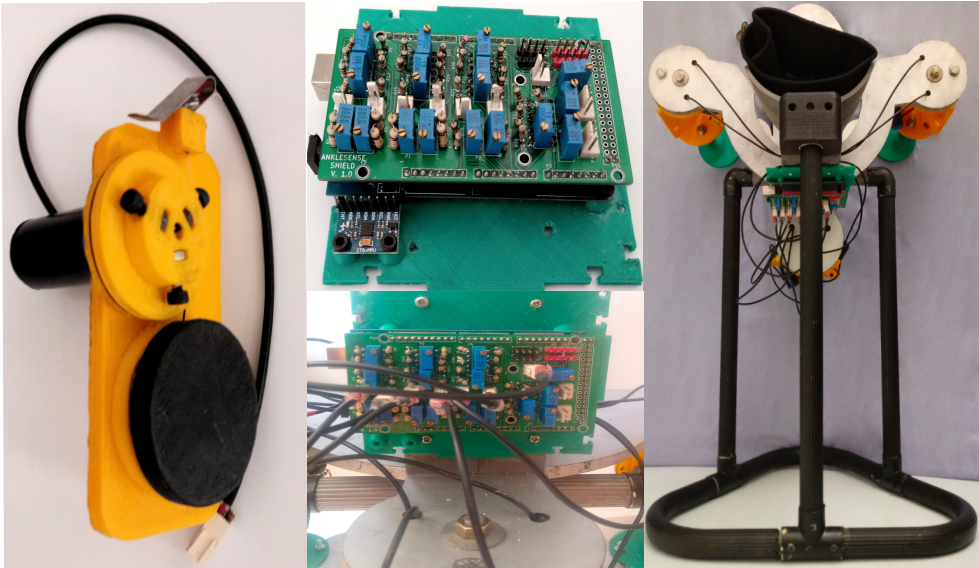


Figure 31. Assembled Structure.

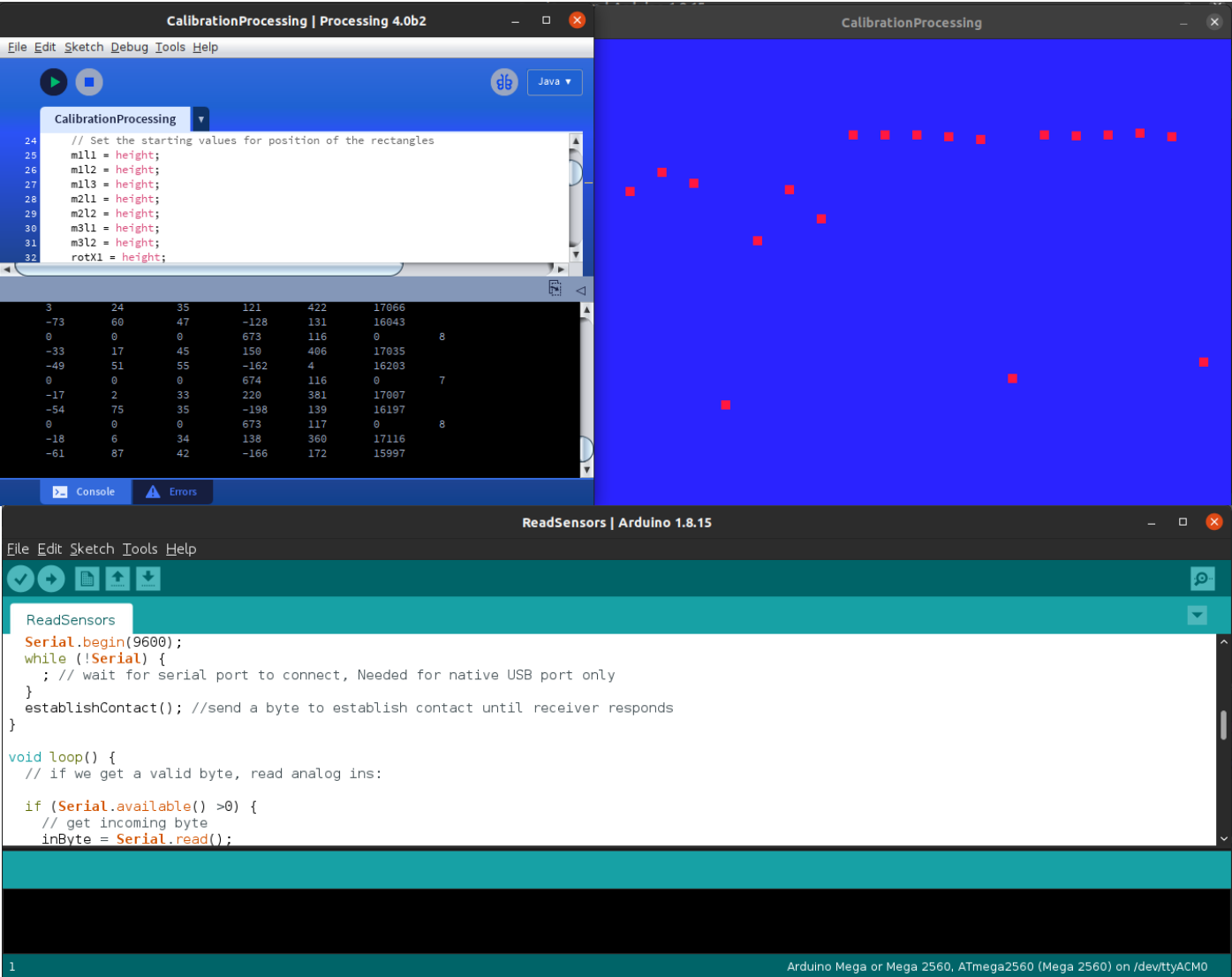


Figure 32. Processing calibration interface.

404 The Table 2 shows the calibration results.

Table 2. Calibration results with digital lectures and real measurements

Sensor ID	l1M1	l2M1	l3M1	l1M2	l2M2	l1M3	l2M3
BCD value	239	330	246	265	177	252	242
Measure,cm	8.0	5.3	6.9	13.0	8.4	7.8	11.5

The Figure 33(a) shows the length with SolidWorks Measurement tool for module A, sensor 1; the lecture for sensor 2 is in Figure 33(b). In Figure 33(c) is the sensor 3 length.

The Table 3 shows the error measured in real prototype and in SolidWorks.

Table 3. Error compared with SolidWorks CAD measurements.

Sensor ID	l1M1	l2M1	l3M1
SW measure	7.622	5.33	6.384
Error in cm	0.38	-0.030	0.52

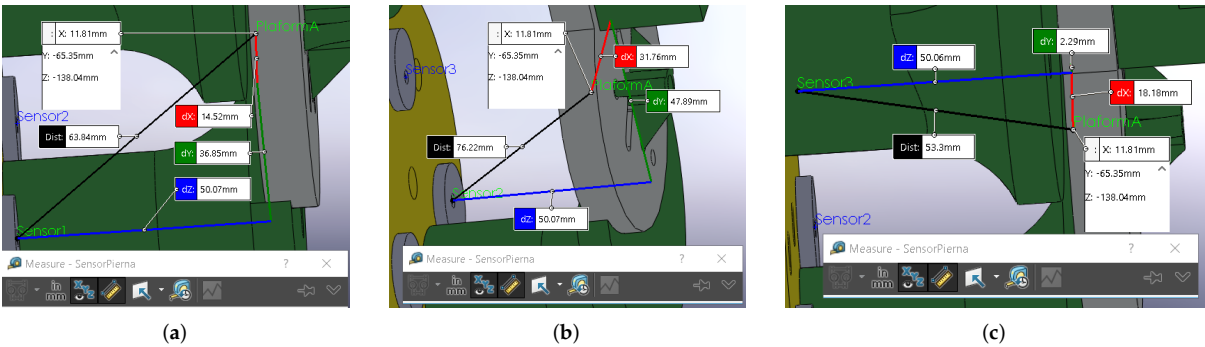


Figure 33. Measuring in SolidWorks: (a) sensor 1 (b) sensor 2 (c) sensor 3

3.3. Trilateration Results

In this section we use the measurements from the sensors to compute trilateration, then we compare with the simulation results. The foot and shank fit in the adjustable platform and support structure, respectively, as is shown in the initial procedure shown in the 21. By introducing the lengths' data to the virtual model, we compute the position for different values. In the Table 4, we show positions computed from the sensor lengths. The resulting figures for each trilateration computation are in 34(a), 34(b), 34(c), and 34(d).

Table 4. Sensor lectures and position points

Sensor ID	I1M1	I2M1	I3M1	I1M2	I2M2	I1M3	I2M3	A	B	C
Pos1. ,cm	11.0	12.6	12.5	14.8	10.8	15.2	11.9	(-11.7, -1.03, -11.0)	(5.97, -9.94, -8.93)	(4.88, 9.90, -10.12)
Pos2. ,cm	10.2	11.7	11.6	15.2	11.3	15.5	12.2	(-12.1, -0.85, -10.2)	(5.57, -9.99, -9.40)	(4.65, 9.89, -10.4)
Pos3. ,cm	9.40	10.8	10.8	15.6	11.7	15.8	12.5	(-12.4, -0.66, -9.39)	(5.14, -10.0, -9.84)	(4.39, 9.87, -10.6)
Pos4. ,cm	8.56	9.89	9.95	16.0	12.2	16.0	12.7	(-12.7, -0.48, -8.53)	(4.67, -10.0, -10.2)	(4.12, 9.86, -10.8)

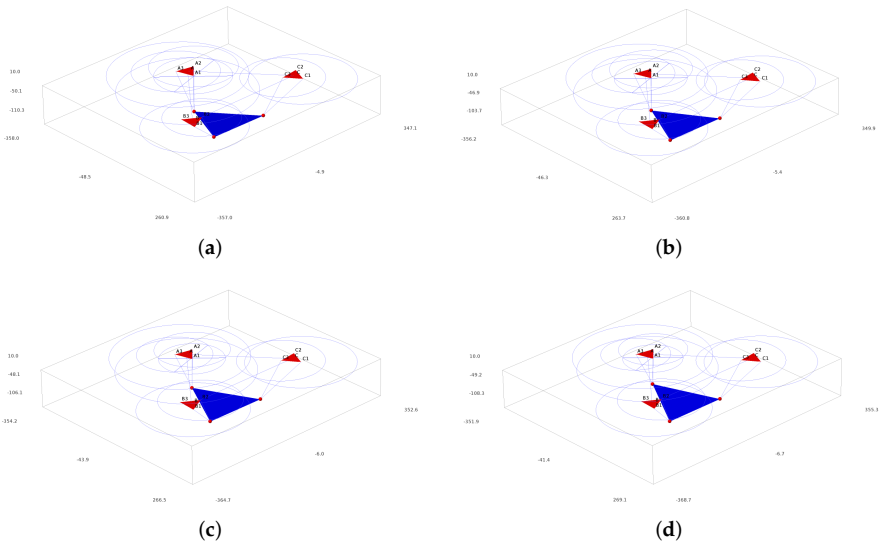


Figure 34. Trilateration results: (a) position 1 (b) position 2 (c) position 3 (d) position 4

3.4. Axis Orientation

In this section, we show the results of model fitting to solve the axis orientation. First, we show the TC axis, and then the ST axis.

3.4.1. TC Axis Circle Fitting

The results of circle fitting for trajectories A, B, and C are in the Table 5, corresponding to ankle joint plantar/dorsiflexion movements.

Table 5. TC Axis Circle Fitting

Trajectory	center	direction	radius
A	(0.8649, 21.38, -67.12)	(-0.089, -0.95, 0.31)	76.66
B	(5.713, 55.31, -78.24)	(-0.089, -0.95, 0.31)	52.46
C	(-2.442, -26.69, -53.15)	(-0.089, -0.95, 0.31)	72.06
PM	(1.552, 16.42, -66.83)	(-0.089, -0.95, 0.31)	53.75

The figures 35(a), 35(a), 35(b), and 35(c) depicts the circle fitting for trajectories A, B, C and PM, respectively.

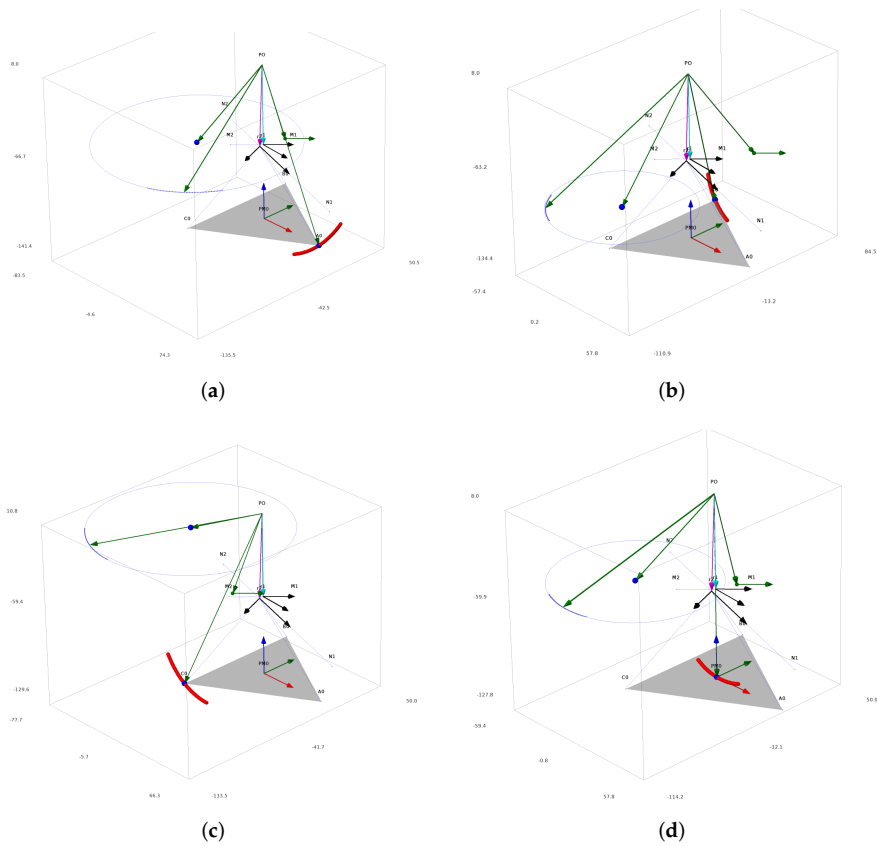


Figure 35. TC axis circle fitting: (a) trajectory A (b) trajectory B (c) trajectory C (d) trajectory PM

3.4.2. ST Axis Circle Fitting

The results of ST circle fitting for trajectories A, B, C, and PM are in the Table 6, corresponding to ankle joint inversion movements.

Table 6. ST Axis Circle Fitting

Trajectory	center	direction	radius
A	(44.44, 18.25, -90.08)	(-0.75, -0.28, 0.60)	24.28
B	(17.57, 6.768, -69.25)	(-0.75, -0.28, 0.60)	65.67
C	(1.578, 1.819, -58.07)	((-0.75, -0.28, 0.60)	69.35
PM	(20.87, 8.882, -72.81)	(-0.75, -0.28, 0.60)	38.75

The figures 36(a), 36(b), 36(c), and 36(d) depicts the circle fitting for trajectories A, B, C and PM, respectively.

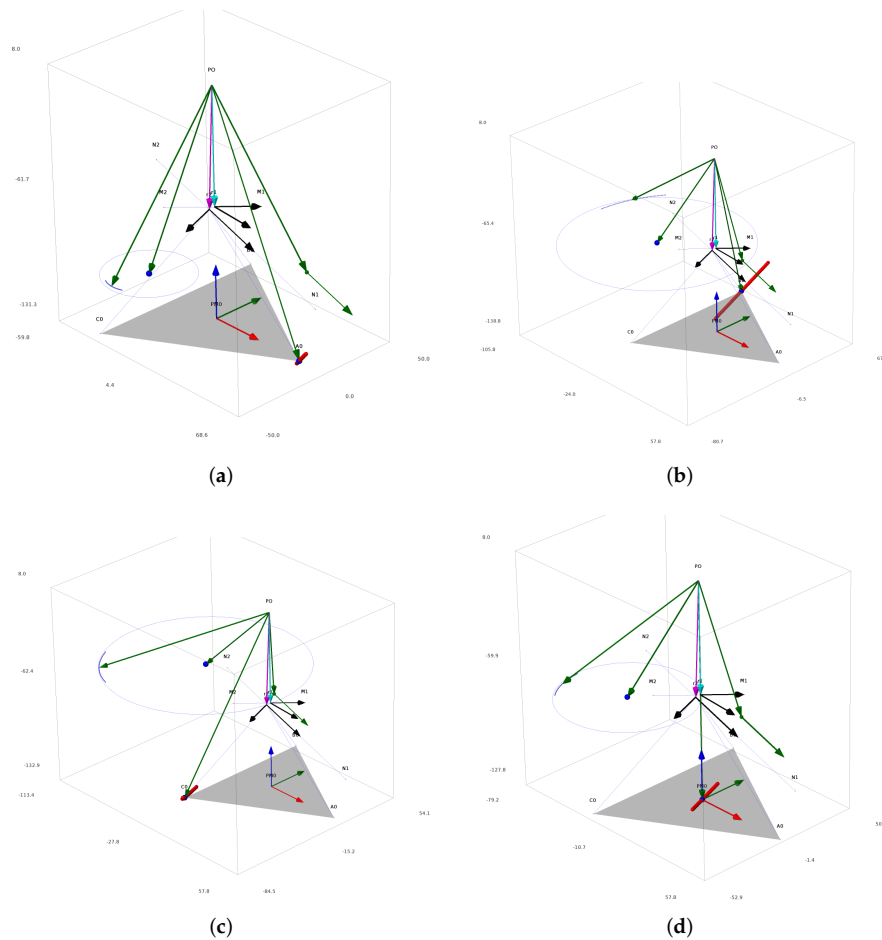


Figure 36. ST axis circle fitting: (a) trajectory A (b) trajectory B (c) trajectory C (d) trajectory PM

3.5. Ankle Manifold representation

In this section, we show the results in the software SageMath Manifolds. We load the model and visualize it as a manifold, we show the axis and the sagittal plane intersection. With the model parameters loaded, r_1 , r_2 , ω_1 , ω_2 , and the origin established in the center of the base modules. We apply the equation:

$$\hat{r}_1 = \bar{c}_0 + \hat{n}_p \cdot d \quad (62)$$

where \bar{c}_0 is the median center computed from trajectories A, B and C center fitting, and \hat{n}_p is the median planes normal vectors containing the circles. The Table 7 shows values for the TC axis in PM chart.

Table 7. Axis Estimation Data

Axis	median center	median normal	r	ω
TC	(19.2, 7.83, -71.0)	(-0.750, -0.280, 0.600)	(-1.74, 0.000, -54.3)	(-0.750, -0.280, 0.600)
ST	(1.21, 18.9, -67.0)	(-0.0890, -0.950, 0.310)	(-0.562, 0.000, -60.8)	(-0.0890, -0.950, 0.310)

--- In the Table 8, we show the Plucker coordinates for the TC and ST axis.

Table 8. Plucker Line Coordinates

Axis	Plucker Line Coordinates
TC	[-0.750 : -0.280 : 0.600: -15.2: 41.7: 0.487]
ST	[-0.0890: -0.950: 0.310: -57.8: 5.59: 0.534]

Finally, the Figure 37(a) shows the ankle manifold, and the Figure 37(b), the chart representing the range of movement, and angle coordinates.

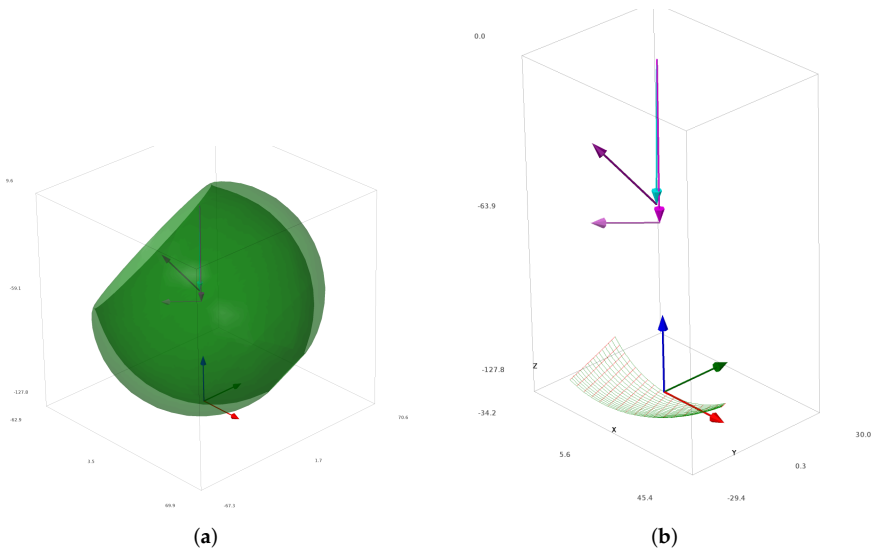


Figure 37. Ankle Joint Manifold (a) Manifold for PM (b) chart with ankle axis coordinates

4. Discussion

In this work, we addressed the ankle model with an alternative approach. We used statistical measurements for the design and implementation of a new device, specially designed to capture the human ankle joint movements. In animal joints, it is difficult to put encoders and linear sensors to measure the range of movement of complex joints in each internal living reference frame. The product of exponential functions formula uses only two frames, and it is useful in this case application. Also, in our work we used an original trilateration method for finding the device’s platform position, as an analytic method, avoiding numerical approximations that can diverge. We proposed the Ankle joint model as a Riemannian manifold. We can define a chart as a subset of such manifold with angle coordinates for measuring the range of movement. Our presented device is lightweight, is non-invasive, and can be used in remote places, on beds or on the floor. By characterizing the ankle parameters, symmetry studies can be done by correlating the left and right ankle joints. The device configuration can be enhanced in future versions, the draw wire sensors used can be changed from analog to digital encoders connected by a CAN bus, reducing wiring, space, weight, and energy consumption. In the future, the model will be used for the synthesis and reconfiguration of an ankle parallel rehabilitation robot. By employing the axis location and the screw theory, forces and torques can be studied as reciprocal screws to the axis location in a re-configurable platform. The robot will be lightweight because of the use of cable driven actuators, inspired in antagonistic muscles; working with reciprocal inhibition for energy optimization. The structure will be reconfigured taking into account the ankle joint as a central mast, and referenced with MMP and MLP markers. The Figure 38 shows a schematic of the re-configurable approach.

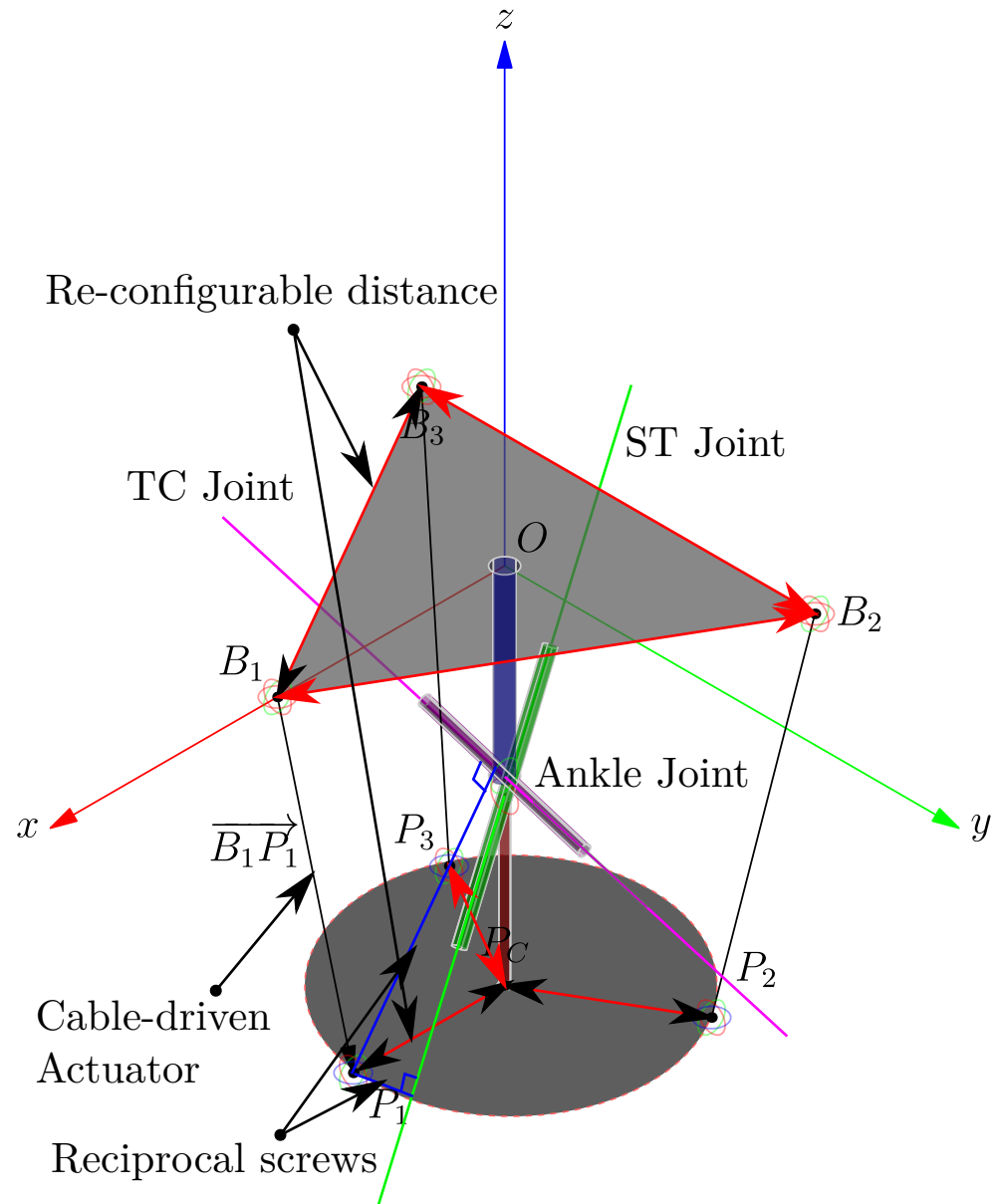


Figure 38. Re-configurable cable-driven robot concept.

Other applications are, for example, by visualizing the platform trajectories one can explain how the calcaneal Achilles insertion is near to the platform's A point. The platform's normal vector changes abruptly near to this region, as was depicted in Figures 24(b), 26(a), and 26(b). Also, Riemannian models, has different properties. We will explore diagnosis and treatments based on this model and metrics by employing machine learning algorithms. This approach can be applied to other joints in humans and other animals, by designing specialized re-configurable hardware and software. Tracking the parameters in different ages and weight conditions, and comparing the ankle models in healthy and injured people.

5. Conclusions

The ankle is the most commonly injured joint of the lower limb, fundamental in fine body balance; it is important to measure the range of motion by in vivo methods for patients laying down in reduced or remote places. We proposed a device based on the ankle anatomy and anthropometry. Additionally, we used a model of a Riemannian manifold, which can be characterized by the device captured data. The simulations en-

abled us to design the size of the device and maximal length of the wires. We presented a trilateration method by using tetrahedrons projected on the base as an efficient alternative to 3D sphere intersections. The draw-wire sensors are modular and a structural part of the device, which is lightweight and portable. The assembly of the electronics is also modular, and other single-board computers and microcontroller boards can be used. The TM will also be used for ankle characterization and diagnosis for rehabilitation robotics, prosthesis, and orthosis design.

6. Patents

We are working in newer device version, with compact design and digital draw wire sensors. A more powerful single board computer, with machine learning capabilities, will be used for diagnosis and rehabilitation.

Author Contributions: Conceptualization, Julio-H Vargas-Riaño, Ángel Valera and Óscar Agudelo; methodology, Ángel Valera and Óscar Agudelo; software, Julio-H Vargas-Riaño; validation, Ángel Valera and Óscar Agudelo; formal analysis, Julio-H Vargas-Riaño; investigation, Julio-H Vargas-Riaño; resources, Ángel Valera and Óscar Agudelo; data curation, Ángel Valera and Óscar Agudelo; writing—original draft preparation, Julio-H Vargas-Riaño; writing—review and editing, Ángel Valera and Óscar Agudelo; visualization, Julio-H Vargas-Riaño; supervision, Ángel Valera; project administration, Ángel Valera ; funding acquisition, Julio-H Vargas-Riaño, Ángel Valera and Óscar Agudelo All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Colciencias-Colfuturo PhD Scholarships Program Educational Credit Forgivable grant number 568.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The code and CAD electronics and mechanical designs are available.

Acknowledgments: The authors thank the Colfuturo Colciencias Collaboration for supporting this work, as well as the Universitat Politècnica de València and the Universidad de los Llanos.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

HAI	Human Ankle Joint
ISB	International Society of Biomechanics
DWS	Draw-wire Sensors
IMU	Inertial Measurement Units
PoE	Product of Exponentials
DoF	Degrees of Freedom
RoM	Range of Motion
BMS	Battery Management System

Appendix A. The Ankle Joint

In this section, we start with the ankle description, which presents a complex movement. First, we study the shank, ankle, and foot bones. Then, we analyzed the ankle movements based on the anatomic spatial and functional representation.

Appendix A.1. Bones

We start with an understanding of inter-bone contact surfaces when studying ankle movements. In Fig. A1, we identify the names of the bones of the left and right feet.

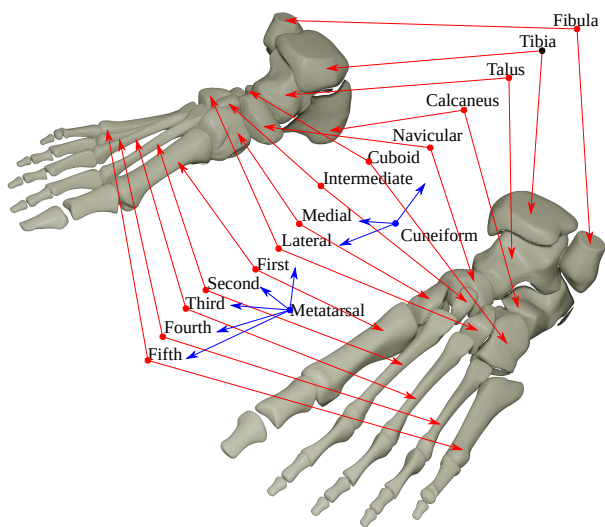


Figure A1. Foot and Ankle Bones.

In Fig. A2, we use the right-hand rotation convention and present the movements systematically. Also, we organize those movements into two rows, corresponding to pronation and supination. In addition, we show the hindfoot and midfoot are the most involved segments in ankle movements.

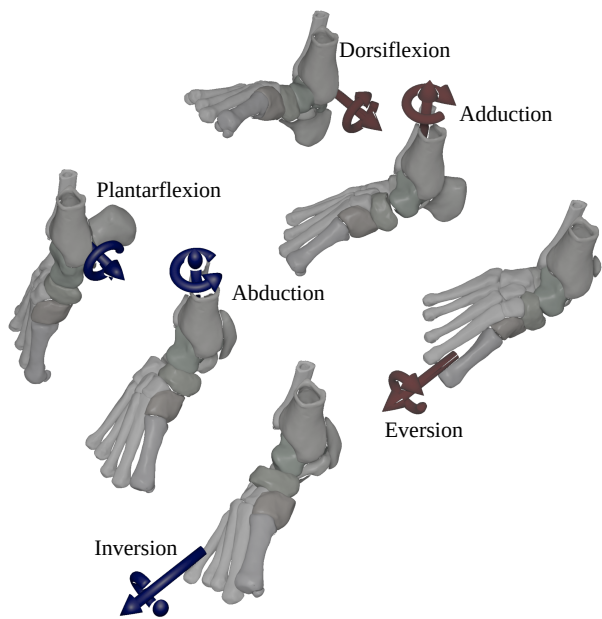


Figure A2. Ankle movements relative to the x, y, and z axes.

Appendix A.2. Kinematic model

The most accepted approach for modeling the ankle joint is the biaxial movement. It results from the interaction of several bones, such as the fibula, tibia, talus, calcaneus, navicular, cuboid, and three cuneiform bones. As shown in Fig. A3, the first axis corresponds to the rotation from the talus regarding the tibia-fibula fixed joint.

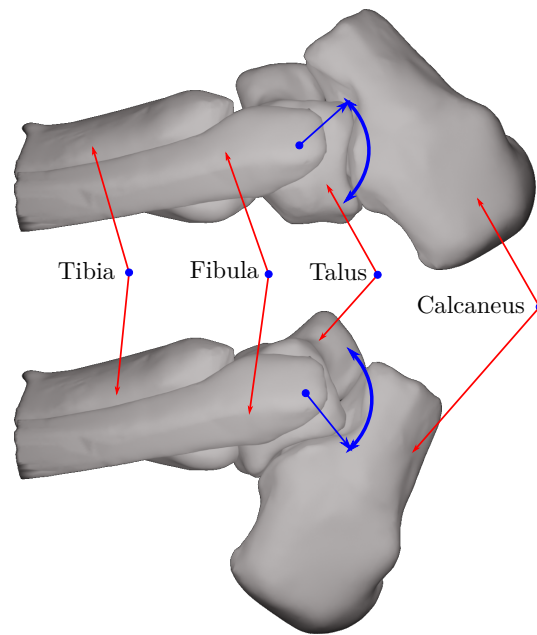


Figure A3. Talocrural rotation.

However, the mathematical model of the ankle is a representation of two hinge joints in series, as presented in Fig. A4.

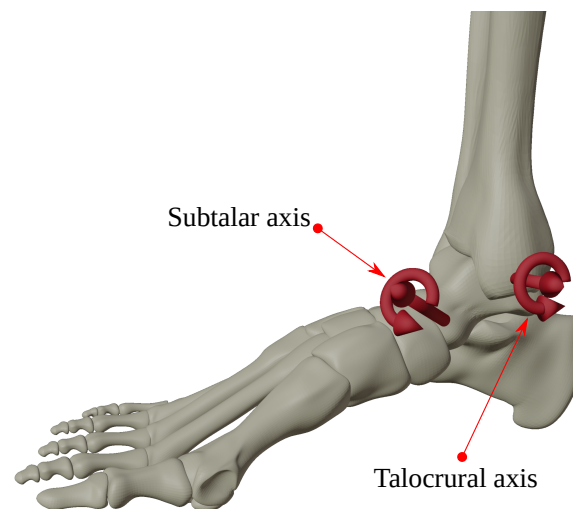


Figure A4. Human ankle joint biaxial model.

We denote the first axis as the talocrural (TC) axis. Some sources name this joint ?mortise? and ?tenon? because it is similar to this architectonic structure. The second axis is the subtalar (ST) joint. The bones involved in this rotation are the talus, the calcaneus, the navicular, and the foot's cuneiform bones. To identify the TC and ST axes, we should define each ankle joint bone's reference frame.

Appendix B. Listings

The SageMath code in listing B1 can be copied and executed directly in a SageMath Cell <https://sagecell.sagemath.org/>

```
1 #Ankle joint forward kinematics by using Product of Exponentials Formula
2 from sage.plot.plot3d.shapes2 import *
3
4 # Rotation axis in a unitary vector and angle in degrees
5 def rot_ax_ang(u,a):
```

```

6     #degrees to radians
7     alpha=a*pi/180
8     na=(1-cos(alpha))
9     ca=cos(alpha)
10    sa=sin(alpha)
11    #rotation matrix
12    R=matrix([[u[0]^2*na+ca, u[0]*u[1]*na-u[2]*sa, u[0]*u[2]*na+u[1]*sa],
13              [u[0]*u[1]*na+u[2]*sa, u[1]^2*na+ca, u[1]*u[2]*na-u[0]*sa],
14              [u[2]*u[0]*na-u[1]*sa, u[2]*u[1]*na+u[0]*sa, u[2]^2*na+ca]])
15    return R
16
17    #Talocrural direction
18    vTC=rot_ax_ang([0,0,1], -6)*rot_ax_ang([1,0,0], 80)*vector([0,0,1])
19    #Subtalar direction
20    vST=rot_ax_ang([0,0,1], 23)*rot_ax_ang([0,1,0], 41)*vector([1,0,0])
21
22    #Anthropometric values
23    vK=12; vL=11; vO=16; vP=1; vQ=5; vR=0.54 #anthropometric measurements
24    H=1750 #height in mm
25    d_m=(0.285-0.039)*H/2 #knee-ankle half distance
26    d_p=0.039*H #ankle-foot
27    z_p=-(d_m+d_p) # z initial position
28    r_p=0.055*H*2/3 #platform radius
29    ae=2*pi/3 #auxiliar angle
30    P0=vector([0,0,0]) #Origin at middle shank
31
32    #initial platform position
33    A0=vector([r_p,0,z_p])
34    B0=vector([r_p*cos(ae),r_p*sin(ae),z_p])
35    C0=vector([r_p*cos(-ae),r_p*sin(-ae),z_p])
36    PM0=(A0+B0+C0)/3 #center of the platform
37
38    #marker representation
39    dot=point3d((0,0,0),size=5,color='red',opacity=.5)
40    dP0=dot.translate(P0)
41    dA0=dot.translate(A0)
42    dB0=dot.translate(B0)
43    dC0=dot.translate(C0)
44    dPM0=dot.translate(PM0)
45
46    r_1=vector([0,0,-d_m]) #intersection point between the talocrural axis and the sagittal plane
47    ap=0.039*H
48    lp=0.152*H
49
50    #points on the malleolar medial and lateral
51    M_1=r_1-ap*vR*vTC
52    M_2=r_1+ap*(1-vR)*vTC
53    tht=41*pi/180
54
55    #intersection between the subtalar axis and the sagittal plane
56    r_2=vQ*vector([-cos(tht),0,-sin(tht)])+r_1
57
58    #points from the hindfoot to the midfoot
59    N_1=r_2+0.6*lp*vR*vST
60    N_2=r_2-0.3*lp*vR*vST
61
62    #representation of the kinematic chain
63    lrt=line([P0,r_1])
64    lm12=line([M_1,M_2])
65    ln12=line([N_1,N_2])
66    lrA=line([r_2,A0])
67    lrB=line([r_2,B0])
68    lrC=line([r_2,C0])
69
70    #markers representation
71    dr1=dot.translate(r_1)
72    dr2=dot.translate(r_2)
73    dM1=dot.translate(M_1)
74    dM2=dot.translate(M_2)

```

```

75 dN1=dot.translate(N_1)
76 dN2=dot.translate(N_2)
77
78 #reference labels
79 plbl=vector([0,0,8])
80 lbp = text3d('P0',P0+plbl)
81 lbp += Text('A0',color='black').translate(A0++plbl)
82 lbp += text3d('B0',B0++plbl)
83 lbp += text3d('C0',C0++plbl)
84 lbp += text3d('PM0',PM0++plbl)
85 #axes labels
86 lbf = text3d('M1',M_1++plbl)
87 lbf += text3d('M2',M_2++plbl)
88 lbf += text3d('N1',N_1++plbl)
89 lbf += text3d('N2',N_2++plbl)
90 lbf += text3d('r1',r_1++plbl)
91 lbf += text3d('r2',r_2++plbl)
92
93
94 #finding the twist unitary vectors
95 w1=n((M_1-M_2)/abs(M_1-M_2))
96 w2=n((N_1-N_2)/abs(N_1-N_2))
97
98 #the perpendicular component
99 v1=n(-w1.cross_product(r_1))
100 v2=n(-w2.cross_product(r_2))
101
102 #angles for the talocrural and subtalar rotations
103 theta1=var('theta_1')
104 theta2=var('theta_2')
105
106 #six dimensional vector xi mapping
107 xi1=matrix([v1[0],v1[1],v1[2],w1[0],w1[1],w1[2]]).transpose()
108 xi2=matrix([v2[0],v2[1],v2[2],w2[0],w2[1],w2[2]]).transpose()
109
110 #transformation of exponential matrix of rotation
111 Rexp1=rot_ax_ang(w1,theta1)
112
113 #rotation matrix component of the homogeneous transformation
114 vexp1=(matrix.identity(3)-Rexp1)*(w1.cross_product(v1))
115
116 #conformation of the homogeneous transformation matrix
117 MTH1=(Rexp1.augment(vexp1)).stack(vector([0,0,0,1]))
118
119 #components for the subtalar axis
120 Rexp2=rot_ax_ang(w2,theta2)
121 vexp2=(matrix.identity(3)-Rexp2)*(w2.cross_product(v2))
122 MTH2=(Rexp2.augment(vexp2)).stack(vector([0,0,0,1]))
123
124 #transformation matrix representing the initial position
125 gstOA=matrix([[1,0,0,A0[0]],[0,1,0,A0[1]],[0,0,1,A0[2]],[0,0,0,1]])
126 gstOB=matrix([[1,0,0,B0[0]],[0,1,0,B0[1]],[0,0,1,B0[2]],[0,0,0,1]])
127 gstOC=matrix([[1,0,0,C0[0]],[0,1,0,C0[1]],[0,0,1,C0[2]],[0,0,0,1]])
128 gstOPM=matrix([[1,0,0,PM0[0]],[0,1,0,PM0[1]],[0,0,1,PM0[2]],[0,0,0,1]])
129
130 #product of exponential matrices for all the points
131 MTHA=MTH1*MTH2*gstOA
132 MTHB=MTH1*MTH2*gstOB
133 MTHC=MTH1*MTH2*gstOC
134 MTHPM=MTH1*MTH2*gstOPM
135
136 #components of the group of rigid movements for the central point
137 f_xpm=MTHPM[0][3]
138 f_ypm=MTHPM[1][3]
139 f_zpm=MTHPM[2][3]
140 #orthogonal direction vectors
141 f_spm=vector([MTHPM[0][0],MTHPM[1][0],MTHPM[2][0]])
142 f_npm=vector([MTHPM[0][1],MTHPM[1][1],MTHPM[2][1]])
143 f_apm=vector([MTHPM[0][2],MTHPM[1][2],MTHPM[2][2]])

```

```

144
145 #components of the three vertices of the plattform
146 f_xA=MTHA[0][3]
147 f_yA=MTHA[1][3]
148 f_zA=MTHA[2][3]
149 f_xB=MTHB[0][3]
150 f_yB=MTHB[1][3]
151 f_zB=MTHB[2][3]
152 f_xC=MTHC[0][3]
153 f_yC=MTHC[1][3]
154 f_zC=MTHC[2][3]
155
156 #saving symbolic expressions to file output
157 tx, ty =var('t_x', 't_y')
158 pmx=f_xpm.subs(theta1==tx, theta2==ty)
159 pmy=f_ypm.subs(theta1==tx, theta2==ty)
160 pmz=f_zpm.subs(theta1==tx, theta2==ty)
161 ax=f_xA.subs(theta1==tx, theta2==ty)
162 ay=f_yA.subs(theta1==tx, theta2==ty)
163 az=f_zA.subs(theta1==tx, theta2==ty)
164 bx=f_xB.subs(theta1==tx, theta2==ty)
165 by=f_yB.subs(theta1==tx, theta2==ty)
166 bz=f_zB.subs(theta1==tx, theta2==ty)
167 cx=f_xC.subs(theta1==tx, theta2==ty)
168 cy=f_yC.subs(theta1==tx, theta2==ty)
169 cz=f_zC.subs(theta1==tx, theta2==ty)
170 exportg = [[pmx,pmy,pmz],[ax,ay,az],[bx,by,bz],[cx,cy,cz],
171             [A0.n(),B0.n(),C0.n(),PM0.n()],
172             [r_1.n(),r_2.n(),N_1.n(),N_2.n(),M_1.n(),M_2.n()]]
173 with open("output.txt", "w") as f: f.write(str(exportg))
174
175 #scaled position vectors
176 ar1=arrow3d(P0,r_1,40,color='cyan')
177 ar2=arrow3d(P0,r_2,40,color='magenta')
178 aw1=arrow3d(r_1, r_1 + 40*w1,40, color = 'red')
179 aw2=arrow3d(r_2, r_2 + 40*w2,40, color = 'blue')
180 av1=arrow3d(r_1, r_1 + 40*v1/abs(r_1),40, color = 'orange')
181 av2=arrow3d(r_2, r_2 + 40*v2/abs(r_2),40, color = 'purple')
182 scrws = ar1 +ar2+ aw1+aw2 + av1 + av2
183
184 #platform initial position
185 pltf= polygon([A0,B0,C0],color='gray',opacity=0.7)
186 #showing the initial position
187 gr = dP0+dA0+dB0+dC0+dPM0
188 gr+= lrt+lm12+ln12+lrA+lrB+lrC
189 gr+= dM1+dM2+dr1+dr2+dN1+dN2
190 gr+= lbp+lbf+scrws+pltf
191 gr.show(aspect_ratio=1,frame=true,figsize=(1024,1024),
192         projection='orthographic', axes = true)
193
194 #changing the angular parameters interactively
195 @interact
196 def _(t1min=slider(-90,-1,label=r'min $\theta_1$', default=-15),
197       t1max=slider(1,90,label=r'max $\theta_1$', default=15),
198       t2min=slider(-90,-1,label=r'min $\theta_2$', default=-15),
199       t2max=slider(1,90,label=r'max $\theta_2$', default=15)):
200
201     @interact
202     def _(txi=slider(t1min,t1max, step_size=1, label=r'Value $\theta_1$', default=0),
203           tyi=slider(t2min,t2max, step_size=1, label=r'Value $\theta_2$', default=0)):
204         #substitute values
205         pmxs=pmx.subs(tx==txi, ty==tyi)
206         pmys=pmy.subs(tx==txi, ty==tyi)
207         pmzs=pmz.subs(tx==txi, ty==tyi)
208         axs=ax.subs(tx==txi, ty==tyi)
209         ays=ay.subs(tx==txi, ty==tyi)
210         azs=az.subs(tx==txi, ty==tyi)
211         bxs=bx.subs(tx==txi, ty==tyi)
212         bys=by.subs(tx==txi, ty==tyi)

```

```

213 bzs=bz.subs(tx==txi, ty==tyi)
214 cxs=cx.subs(tx==txi, ty==tyi)
215 cys=cy.subs(tx==txi, ty==tyi)
216 czs=cz.subs(tx==txi, ty==tyi)
217 #plotting
218 dp=10 #data points
219 cpm=parametric_plot3d([f_xpm, f_ypm, f_zpm],(theta_1, t1min, t1max),
220                        (theta_2, t2min, t2max),plot_points=[dp,dp])
221 cA=parametric_plot3d([f_xA, f_yA, f_zA], (theta_1, t1min, t1max),
222                      (theta_2, t2min, t2max),plot_points=[dp,dp])
223 cB=parametric_plot3d([f_xB, f_yB, f_zB], (theta_1, t1min, t1max),
224                      (theta_2, t2min, t2max),plot_points=[dp,dp])
225 cC=parametric_plot3d([f_xC, f_yC, f_zC], (theta_1, t1min, t1max),
226                      (theta_2, t2min, t2max),plot_points=[dp,dp],texture="red")
227 #create vectors
228 ptpm=vector([pmxs,pmys,pmzs])
229 spms=vector([f_spm[0].subs(theta1==txi, theta2==tyi),
230             f_spm[1].subs(theta1==txi, theta2==tyi),
231             f_spm[2].subs(theta1==txi, theta2==tyi)]).n()
232 apms=arrow3d(ptpm, ptpm + 40*spms,40, color='red')
233 spmn=vector([f_npm[0].subs(theta1==txi, theta2==tyi),
234             f_npm[1].subs(theta1==txi, theta2==tyi),
235             f_npm[2].subs(theta1==txi, theta2==tyi)]).n()
236 apmn=arrow3d(ptpm, ptpm + 40*spmn,40, color='green')
237 spma=vector([f_apm[0].subs(theta1==txi, theta2==tyi),
238             f_apm[1].subs(theta1==txi, theta2==tyi),
239             f_apm[2].subs(theta1==txi, theta2==tyi)]).n()
240 apma=arrow3d(ptpm, ptpm + 40*spma,40, color='blue')
241 pta=vector([axs,ays,azs]).n()
242 ptb=vector([bxs,bys,bzs]).n()
243 ptc=vector([cxs,cys,czs]).n()
244 ptf=polygon([pta,ptb,ptc],color='green',opacity=0.7)
245 #plotting
246 fkin=dP0+apma+apmn+apms+ptf+lbp+cA+cB+cC+cpm
247 fkin.show(aspect_ratio=1,frame=true,figsize=(1024,1024),
248           projection='orthographic', axes = true)

```

Listing 1: Sagemath Forward Kinematics

For visualization and interactive view in ®Acrobat Reader, we exported the symbolic code to Asymptote, the listing B2 can be executed in <http://asymptote.ualberta.ca/>

```

1 settings.outformat="pdf";
2 settings.prc= false;
3 settings.render = 0;
4 import three;
5 import graph3;
6 size(200, 0);
7 currentprojection=orthographic(30,30,30);
8 xaxis3("$x$",0,100,red,OutTicks(2,2));
9 yaxis3("$y$",0,100,red,OutTicks(2,2));
10 zaxis3("$z$",-300,0,red,OutTicks(2,2));
11
12 //range of motion
13 real tcmin=-180, tcmax = 180, stmin = -180, stmax = 180;
14 //actual position
15 real theta1 = 0, theta2 = 0;
16
17
18 //initial points
19 triple po = (0, 0, 0);
20 triple a0 = (64.1666666666667, 0.000000000000000, -283.500000000000);
21 triple b0 = (-32.0833333333333, 55.5699634095015, -283.500000000000);
22 triple c0 = (-32.0833333333333, -55.5699634095015, -283.500000000000);
23 triple pm0 = (0.000000000000000, 0.000000000000000,
24             -283.500000000000);
25 triple r1 = (0.000000000000000, 0.000000000000000, -215.250000000000);
26 triple r2 = (-3.77354790111386, 0.000000000000000, -218.530295144953);
27 triple n1 = (56.0996689565118, 25.4146727215185, -275.072086499470);
28 triple n2 = (-33.7101563299267, -12.7073363607592, -190.259399467694);

```

```

29 triple m1 = (3.79386995439788, 36.0962614380663, -221.649803587915);
30 triple m2 = (-3.23181514633894, -30.7486671509453, -209.798315462147);
31
32 //parametric functions
33 triple pm(pair t) {
34     return (-16.48474746562729*cos(1/180*pi*t.x + 1/180*pi*t.y) - 10.166838471101048*cos(-1/180*pi*t.x +
        1/180*pi*t.y) + 25.431588522123675*cos(1/180*pi*t.x) - 1.0075390222555773*cos(1/180*pi*t.y) -
        30.07656063465307*sin(1/180*pi*t.x + 1/180*pi*t.y) + 6.799869652374312*sin(-1/180*pi*t.x + 1/180*pi
        *t.y) - 29.968498301984226*sin(1/180*pi*t.x) + 4.11787048538047*sin(1/180*pi*t.y) +
        2.227536436860234, -3.5988642455907733*cos(1/180*pi*t.x + 1/180*pi*t.y) - 0.1574203300851842*cos
        (-1/180*pi*t.x + 1/180*pi*t.y) - 7.851215460082017*cos(1/180*pi*t.x) - 9.586093459590682*cos(1/180*
        pi*t.y) + 5.9580368183586465*sin(1/180*pi*t.x + 1/180*pi*t.y) - 2.4772749693983664*sin(-1/180*pi*t.
        x + 1/180*pi*t.y) - 1.4096266870201895*sin(1/180*pi*t.x) + 39.17892057319622*sin(1/180*pi*t.y) +
        21.193593495348665, -30.070693601917178*cos(1/180*pi*t.x + 1/180*pi*t.y) - 6.914891650849512*cos
        (-1/180*pi*t.x + 1/180*pi*t.y) - 29.206425431552418*cos(1/180*pi*t.x) + 1.6995974893975312*cos
        (1/180*pi*t.y) + 15.774905204951304*sin(1/180*pi*t.x + 1/180*pi*t.y) - 9.941327559108188*sin
        (-1/180*pi*t.x + 1/180*pi*t.y) - 25.716232764059498*sin(1/180*pi*t.x) - 6.946353624049991*sin
        (1/180*pi*t.y) - 219.00758680507846);
35 }
36
37
38
39
40 //initial position
41
42 path3 iplatform = (a0--b0--c0--cycle);
43 draw(po--r1^^po--r2^^n1--n2^^m1--m2);
44 draw(surface(iplatform),green+opacity(0.3));
45 dot(po--a0--b0--c0--pm0--r1--r2--n1--n2--m1--m2, 3+red);
46 label("$\rm{P_0}$",po,NE);
47 label("$\rm{A_0}$",a0,2SW);
48 label("$\rm{B_0}$",b0,3SE);
49 label("$\rm{C_0}$",c0,NW);
50 label("$\rm{P_{MO}}$",pm0,NE);
51 label("$\rm{N_1}$",n1,NE);
52 label("$\rm{N_2}$",n2,NE);
53 label("$\rm{M_1}$",m1,NE);
54 label("$\rm{M_2}$",m2,NE);
55 label("$\rm{r_1}$",r1,NW);
56 label("$\rm{r_2}$",r2,SE);
57
58 //compute surface
59 pen p=rgb(0.2,0.5,0.7)+opacity(0.2);
60 surface spm=surface(pm,(tcmin,stmin),(tcmax,stmax),36,36,Spline);
61
62 // surface & mesh
63 draw(spm,lightgray+opacity(0.1),meshpen=p,render(merge=true));

```

Listing 2: Asymptote code

The simulation of the real position is in the listing B3.

```

1 from sage.plot.plot3d.transform import rotate_arbitrary
2 from sage.plot.plot3d.plot3d import axes
3 #centroid of the base
4 cb=vector([116.666,97.905,0])
5 angrot =(79.91/2)*(pi/180)
6 mrotz = rotate_arbitrary((0.0,0.0,1.0),angrot)
7 #base sensor modules centerpoints
8 A= vector([0,0,0])
9 A= mrotz*A
10 Ap0= vector([30,-0.67,-50.5]) + A #initial platform A point
11 lbl = text3d('A',A+vector([10,10,10]))
12 B=vector([52.268,293.716,0])
13 B= mrotz*B
14 Bp0 = vector([-26.4,-92.6,-50.5]) + B
15 lbl += text3d('B',B+vector([10,10,10]))
16 C= vector([297.73,0,0])
17 C= mrotz*C
18 Cp0 = vector([-25.67,90.99, -50.5]) + C

```



```

19 lbl += text3d('C',C+vector([10,10,10]))
20 #base points
21 rtb=32.76 #radius of the tetrahedron base
22 bt1A=vector([31.9,-7.44,0])
23 bt2A=vector([-9.51,31.5,0])
24 bt3A=vector([-22.39,-23.91,0])
25 A1=bt1A+A
26 lbl += text3d('A1',A1+vector([10,10,10]))
27 A2=bt2A+A
28 A3=bt3A+A
29 bt1B=vector([32.52,3.92,0])
30 bt2B=vector([-19.66,26.21,0])
31 bt3B=vector([-12.87,-30.13,0])
32 B1=bt1B+B
33 lbl += text3d('B1',B1+vector([10,10,10]))
34 B2=bt2B+B
35 B3=bt3B+B
36 bt1C=vector([27.44,-17.9,0])
37 bt2C=vector([1.79,32.71,0])
38 bt3C=vector([-29.42,-14.81,0])
39 C1=bt1C+C
40 lbl += text3d('C1',C1+vector([10,10,10]))
41 C2=bt2C+C
42 C3=bt3C+C
43 t1=polygon([A1,A2,A3,A1],color='red')
44 t2=polygon([B1,B2,B3,B1],color='red')
45 t3=polygon([C1,C2,C3,C1],color='red')
46 tp=polygon([Ap0,Bp0,Cp0,Ap0])
47 lA1=line([Ap0,A1])
48 lA2=line([Ap0,A2])
49 lA3=line([Ap0,A3])
50 lB1=line([Bp0,B1])
51 lB3=line([Bp0,B3])
52 lC1=line([Cp0,C1])
53 lC2=line([Cp0,C2])
54 tb=line([A,B,C,A])+lbl+axes(100)+t1+t2+t3+tp+lA1+lA2+lA3+lB1+lB3+lC1+lC2
55 tb.show(viewer='threejs', figsize=1000, frame_aspect_ratio=[1,1,1], aspect_ratio=[1,1,1], axes='true')

```

Listing 3: Real dimensions

References

1. Fong, D.T.P.; Hong, Y.; Chan, L.K.; Yung, P.S.H.; Chan, K.M. A Systematic Review on Ankle Injury and Ankle Sprain in Sports. *Sports Medicine* **2007**, *37*, 73–94. doi:10.2165/00007256-200737010-00006.
2. Waldén, M.; Häggglund, M.; Orchard, J.; Kristenson, K.; Ekstrand, J. Regional differences in injury incidence in European professional football. *Scandinavian journal of medicine & science in sports* **2013**, *23*, 424–430.
3. McKay, G.D.; Goldie, P.A.; Payne, W.R.; Oakes, B.W. Ankle injuries in basketball: injury rate and risk factors. *British Journal of Sports Medicine* **2001**, *35*, 103–108.
4. Swenson, D.M.; Collins, C.L.; Fields, S.K.; Comstock, R.D. Epidemiology of US High School Sports-Related Ligamentous Ankle Injuries, 2005/06–2010/11. *Clinical Journal of Sport Medicine* **2013**, *23*, 190–196. doi:10.1097/JSM.0b013e31827d21fe.
5. Waterman, B.R.; Belmont, P.J.; Cameron, K.L.; DeBerardino, T.M.; Owens, B.D. Epidemiology of Ankle Sprain at the United States Military Academy. *Am J Sports Med* **2010**, *38*, 797–803. Publisher: SAGE Publications Inc STM, doi:10.1177/0363546509350757.
6. Bulathsinhala, L.; Hill, O.T.; Scofield, D.E.; Haley, T.F.; Kardouni, J.R. Epidemiology of Ankle Sprains and the Risk of Separation From Service in U.S. Army Soldiers. *J Orthop Sports Phys Ther* **2015**, *45*, 477–484. doi:10.2519/jospt.2015.5733.
7. Sa, A.; Km, W.; Ra, S.; Sk, B. Epidemiological patterns of musculoskeletal injuries and physical training. *Medicine and science in sports and exercise* **1999**, *31*, 1176–1182. doi:10.1097/00005768-199908000-00015.
8. Herzog, M.M.; Kerr, Z.Y.; Marshall, S.W.; Wikstrom, E.A. Epidemiology of Ankle Sprains and Chronic Ankle Instability. *J Athl Train* **2019**, *54*, 603–610. doi:10.4085/1062-6050-447-17.
9. De Boer, A.S.; Schepers, T.; Panneman, M.J.; Van Beeck, E.F.; Van Lieshout, E.M. Health care consumption and costs due to foot and ankle injuries in the Netherlands, 1986–2010. *BMC Musculoskeletal Disorders* **2014**, *15*, 128. doi:10.1186/1471-2474-15-128.
10. Ball, R.S. *A Treatise on the Theory of Screws*; Cambridge University Press, 1998.
11. Featherstone, R. *Robot Dynamics Algorithms*; The Springer International Series in Engineering and Computer Science, Springer US, 1987. doi:10.1007/978-0-387-74315-8.
12. Tsai, L.W. *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*; John Wiley & Sons, 1999.
13. Kurfess, T.R. *Robotics and Automation Handbook*; CRC Press, 2004.

14. Siciliano, B.; Khatib, O. *Springer Handbook of Robotics*; Springer Science & Business Media, 2008.
15. Kong, X.; Gosselin, C.M. *Type Synthesis of Parallel Mechanisms*; Springer Berlin Heidelberg, 2010. Google-Books-ID: B7iocQAACAAJ.
16. Angeles, J. *Fundamentals of robotic mechanical systems: theory, methods, and algorithms*; Springer: New York; Hong Kong, 2003. OCLC: 56118684.
17. Taghirad, H.D. *Parallel Robots: Mechanics and Control*; CRC Press, 2013.
18. Zhao, J. *Advanced theory of constraint and motion analysis for robot mechanisms*; Academic Press: Oxford [UK]; Waltham, MA, 2014. OCLC: 870340086.
19. Gallardo-Alvarado, J. *Kinematic Analysis of Parallel Manipulators by Algebraic Screw Theory*; Springer International Publishing: Cham, 2016. doi:10.1007/978-3-319-31126-5.
20. Müller, A. Screw and Lie group theory in multibody kinematics. *Multibody System Dynamics* **2018**, 43, 37–70. doi:10.1007/s11044-017-9582-7.
21. Minguzzi, E. A geometrical introduction to screw theory. *European Journal of Physics* **2013**, 34, 613–632. Publisher: IOP Publishing, doi:10.1088/0143-0807/34/3/613.
22. Terze, Z.; Mueller, A.; Zlatar, D. Lie-Group Integration Method for Constrained Multibody Systems in State Space. *Multibody System Dynamics* **2015**, 33, 1–33. doi:10.1007/s11044-014-9439-2.
23. Pardos-Gotor, J. *Screw Theory for Robotics - A practical approach for modern Robot Mechanics - A compelling computational approach for Screw Theory KINEMATICS.*; 2018.
24. Solà, J.; Deray, J.; Atchuthan, D. A micro Lie theory for state estimation in robotics. *ArXiv* **2018**.
25. Selig, J. *Geometric Fundamentals of Robotics*; 2005.
26. Eade, E. Lie Groups for 2D and 3D Transformations. p. 25.
27. Stefano Stramigioli and Herman Bruyninckx. *Geometry and Screw Theory for Robotics*.
28. Hestenes, D. New Tools for Computational Geometry and Rejuvenation of Screw Theory. In *Geometric Algebra Computing: in Engineering and Computer Science*; Bayro-Corrochano, E.; Scheuermann, G., Eds.; Springer: London, 2010; pp. 3–33. doi:10.1007/978-1-84996-108-0_1.
29. Corrochano, E.B.; Sobczyk, G. Applications of Lie Algebras and the Algebra of Incidence. In *Geometric Algebra with Applications in Science and Engineering*; Corrochano, E.B.; Sobczyk, G., Eds.; Birkhäuser: Boston, MA, 2001; pp. 252–277. doi:10.1007/978-1-4612-0159-5_13.
30. Lynch, K.M.; Park, F.C. *Modern Robotics*; Cambridge University Press, 2017. Google-Books-ID: 5NzFDgAAQBAJ.
31. Gal, J.A.; Gallo, L.M.; Palla, S.; Murray, G.M.; Klineberg, I.J.; Johnson, C.W. Characterisation of human jaw biomechanics based on screw theory. *Australian Journal of Mechanical Engineering* **2003**, 1, 11–16. doi:10.1080/14484846.2003.11464460.
32. Gal, J.A.; Gallo, L.M.; Palla, S.; Murray, G.; Klineberg, I. Analysis of human mandibular mechanics based on screw theory and in vivo data. *Journal of biomechanics* **2004**, 37, 1405–1412.
33. Martelli, S.; Visani, A. COMPUTER INVESTIGATION INTO THE ANATOMICAL LOCATION OF THE AXES OF ROTATION IN THE KNEE. *Journal of Mechanics in Medicine and Biology* **2002**, 02, 433–447. doi:10.1142/S0219519402000411.
34. Wolf, A.; Degani, A. Classifying Knee Pathologies using Instantaneous Screws of the Six Degrees-of-Freedom Knee Motion. The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006., 2006, pp. 1047–1052. doi:10.1109/BIOROB.2006.1639230.
35. Wolf, A. Instantaneous Screws of Weight-Bearing Knee: What Can the Screws Tell Us About the Knee Motion. *Journal of Biomechanical Engineering* **2014**, 136, 074502. doi:10.1115/1.4027055.
36. Liu, S.; Zhang, J.; Zhang, Y.; Zhu, R. A wearable motion capture device able to detect dynamic motion of human limbs. *Nature Communications* **2020**, 11, 5615. Bandiera_abtest: a Cc_license_type: cc_by Cg_type: Nature Research Journals Number: 1 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Biomedical engineering;Health care Subject_term_id: biomedical-engineering;health-care, doi:10.1038/s41467-020-19424-2.
37. McGrath, T.; Fineman, R.; Stirling, L. An Auto-Calibrating Knee Flexion-Extension Axis Estimator Using Principal Component Analysis with Inertial Sensors. *Sensors (Basel, Switzerland)* **2018**, 18, 1882. doi:10.3390/s18061882.
38. Krishnan, R.; Cruciani, S.; Gutierrez-Farewik, E.; Björzell, N.; Smith, C. Reliably Segmenting Motion Reversals of a Rigid-IMU Cluster Using Screw-Based Invariants. *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)* **2018**. doi:10.1109/HUMANOIDS.2018.8624969.
39. Ancillao, A.; Vochten, M.; Aertbeliën, E.; Decré, W.; Schutter, J. Estimating the Instantaneous Screw Axis and the Screw Axis Invariant Descriptor of Motion by Means of Inertial Sensors: An Experimental Study with a Mechanical Hinge Joint and Comparison to the Optoelectronic System. *Sensors* **2020**. doi:10.3390/s20010049.
40. Digo, E.; Pierro, G.; Pastorelli, S.; Gastaldi, L. Tilt-Twist Method Using Inertial Sensors to Assess Spinal Posture During Gait. *Advances in Service and Industrial Robotics*. Springer, Cham, 2019, pp. 384–392. doi:10.1007/978-3-030-19648-6_44.
41. Parsa, K.; Angeles, J.; Misra, A. Pose-and-twist estimation of a rigid body using accelerometers. *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, 2001, Vol. 3, pp. 2873–2878 vol.3. ISSN: 1050-4729, doi:10.1109/ROBOT.2001.933057.
42. Barrau, A.; Bonnabel, S. A Mathematical Framework for IMU Error Propagation with Applications to Preintegration. *2020 IEEE International Conference on Robotics and Automation (ICRA)* **2020**. doi:10.1109/ICRA40945.2020.9197492.

43. Wu, G.; Siegler, S.; Allard, P.; Kirtley, C.; Leardini, A.; Rosenbaum, D.; Whittle, M.; D'Lima, D.D.; Cristofolini, L.; Witte, H.; Schmid, O.; Stokes, I.; Standardization and Terminology Committee of the International Society of Biomechanics. ISB recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion—part I: ankle, hip, and spine. International Society of Biomechanics. *Journal of Biomechanics* **2002**, *35*, 543–548.
44. Mann, R.A. Biomechanics of the Ankle. In *Joint Surgery Up to Date*; Hirohata, K.; Kurosaka, M.; Cooke, T.D.V., Eds.; Springer Japan: Tokyo, 1989; pp. 73–81. doi:10.1007/978-4-431-68096-3_8.
45. Winter, D.A. *Biomechanics and motor control of human movement*; John Wiley & Sons, 2009.
46. Dawe, E.J.C.; Davis, J. (vi) Anatomy and biomechanics of the foot and ankle. *Orthopaedics and Trauma* **2011**, *25*, 279–286. doi:10.1016/j.mporth.2011.02.004.
47. Coughlin, M.J.; Saltzman, C.L.; Mann, R.A. *Mann's Surgery of the Foot and Ankle E-Book: Expert Consult - Online*; Elsevier Health Sciences, 2013. Google-Books-ID: DYErAQAAQBAJ.
48. Xie, S.S. Kinematic and Computational Model of Human Ankle. In *Advanced Robotics for Medical Rehabilitation: Current State of the Art and Recent Advances*; Xie, S.S., Ed.; Springer Tracts in Advanced Robotics, Springer International Publishing: Cham, 2016; pp. 185–221. doi:10.1007/978-3-319-19896-5_7.
49. Delp, S.; Loan, J.; Hoy, M.; Zajac, F.; Topp, E.; Rosen, J. An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. *IEEE Transactions on Biomedical Engineering* **1990**, *37*, 757–767. doi:10.1109/10.102791.
50. Donatelli, R. Normal Biomechanics of the Foot and Ankle. *Journal of Orthopaedic & Sports Physical Therapy* **1985**, *7*, 91–95. doi:10.2519/jospt.1985.7.3.91.
51. Dul, J.; Johnson, G.E. A kinematic model of the human ankle. *Journal of Biomedical Engineering* **1985**, *7*, 137–143. doi:10.1016/0141-5425(85)90043-3.
52. Böhler, A. The biomechanics of the foot. *Clinical Prosthetics and Orthotics* **1986**, *10*, 8–14.
53. Lundberg, A.; K Svensson, O.; Németh, G.; Selvik, G. *The axis of rotation of the ankle joint*; Vol. 71, 1989. doi:10.1302/0301-620X.71B1.2915016.
54. Singh, A.K.; Starkweather, K.D.; Hollister, A.M.; Jatana, S.; Lupichuk, A.G. Kinematics of the Ankle: A Hinge Axis Model. *Foot & Ankle International* **1992**, *13*, 439–446. doi:10.1177/107110079201300802.
55. Leardini, A.; O'Connor, J.; Catani, F.; Giannini, S. A geometric model of the human ankle joint. *Journal of Biomechanics* **1999**, *32*, 585–591. doi:10.1016/S0021-9290(99)00022-6.
56. Brockett, C.L.; Chapman, G.J. Biomechanics of the ankle. *Orthopaedics and Trauma* **2016**, *30*, 232–238. doi:10.1016/j.mporth.2016.04.015.
57. Bruening, D.; Richards, J. *Optimal ankle axis position for articulated boots*; Vol. 4, 2005.
58. Leitch, J.; Stebbins, J.; Zavatsky, A.B. Subject-specific axes of the ankle joint complex. *Journal of Biomechanics* **2010**, *43*, 2923–2928. doi:10.1016/j.jbiomech.2010.07.007.
59. Parr, W.C.H.; Chatterjee, H.J.; Soligo, C. Calculating the axes of rotation for the subtalar and talocrural joints using 3D bone reconstructions. *Journal of biomechanics* **2012**, *45*, 1103–1107. doi:10.1016/j.jbiomech.2012.01.011.
60. Nichols, J.A.; Roach, K.E.; Fiorentino, N.M.; Anderson, A.E. Predicting tibiotalar and subtalar joint angles from skin-marker data with dual-fluoroscopy as a reference standard. *Gait & Posture* **2016**, *49*, 136–143. doi:10.1016/j.gaitpost.2016.06.031.
61. Kirby, K.A. Subtalar Joint Axis Location and Rotational Equilibrium Theory of Foot Function. *Journal of the American Podiatric Medical Association* **2001**, *91*, 465–487. doi:10.7547/87507315-91-9-465.
62. Spooner, S.; Kirby, K. *The subtalar joint axis locator: A preliminary report*; Vol. 96, 2006. doi:10.7547/0960212.
63. Lewis, G.S.; Kirby, K.A.; Piazza, S.J. Determination of subtalar joint axis location by restriction of talocrural joint motion. *Gait & Posture* **2007**, *25*, 63–69. doi:10.1016/j.gaitpost.2006.01.001.
64. Lewis, G.S.; Cohen, T.L.; Seisler, A.R.; Kirby, K.A.; Sheehan, F.T.; Piazza, S.J. In vivo tests of an improved method for functional location of the subtalar joint axis. *Journal of Biomechanics* **2009**, *42*, 146–151. doi:10.1016/j.jbiomech.2008.10.010.
65. De Schepper, J.; Van Alsenoy, K.; Rijckaert, J.; De Mits, S.; Lootens, T.; Roosen, P. Intratest reliability in determining the subtalar joint axis using the palpation technique described by K. Kirby. *Journal of the American Podiatric Medical Association* **2012**, *102*, 122–129. Publisher: The American Podiatric Medical Association.
66. Van Alsenoy, K.; De Schepper, J.; Santos, D.; Vereecke, E.; D'Août, K. The Subtalar Joint Axis Palpation Technique: Part 1 – Validating a Clinical Mechanical Model. *Journal of the American Podiatric Medical Association* **2014**, *104*. doi:10.7547/0003-0538-104.3.238.
67. Van Alsenoy, K.K.; D'Août, K.; Vereecke, E.E.; De Schepper, J.; Santos, D. The Subtalar Joint Axis Palpation Technique: Part 2: Reliability and Validity Results Using Cadaver Feet. *Journal of the American Podiatric Medical Association* **2014**, *104*, 365–374. Publisher: American Podiatric Medical Association.
68. Krähenbühl, N.; Horn-Lang, T.; Hintermann, B.; Knupp, M. The subtalar joint: A complex mechanism. *EFORT open reviews* **2017**, *2*, 309–316.
69. Jastifer, J.R.; Gustafson, P.A. The subtalar joint: biomechanics and functional representations in the literature. *The Foot* **2014**, *24*, 203–209.
70. Andrade-Cetto, J.; Thomas, F. A Wire-Based Active Tracker. *IEEE Transactions on Robotics* **2008**, *24*, 642–651. Conference Name: IEEE Transactions on Robotics, doi:10.1109/TRO.2008.924260.

71. Thomas, F.; Ottaviano, E.; Ros, L.; Ceccarelli, M. Coordinate-free formulation of a 3-2-1 wire-based tracking device using Cayley-Menger determinants. 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422), 2003, Vol. 1, pp. 355–361 vol.1. ISSN: 1050-4729, doi:10.1109/ROBOT.2003.1241621.
72. Thomas, F.; Ros, L. Revisiting trilateration for robot localization **2005**. Publisher: 'Institute of Electrical and Electronics Engineers (IEEE)', doi:10.1109/tro.2004.833793.
73. Salleh, S.; Rahmat, M.F.; Othman, S.M.; Abidin, H.Z. Application of draw wire sensor in the tracking control of an electro hydraulic actuator system **2015**. doi:10.11113/jt.v73.4406.
74. Jiafan, Z.; Jinsong, L.; Liwei, Q.; Dandan, Z. Kinematic analysis of a 6-DOF wire-based tracking device and control strategy for its application in robot easy programming. 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2009, pp. 1591–1596. doi:10.1109/ROBIO.2009.5420394.
75. Bulling, A.; Blanke, U.; Schiele, B. A Tutorial on Human Activity Recognition Using Body-worn Inertial Sensors. *ACM Computing Surveys* **2014**, 46, 1–33. Publisher: ACM, doi:10.1145/2499621.
76. Chermak, L.; Aouf, N.; Richardson, M.A.; Visentin, G. Real-time smart and standalone vision/IMU navigation sensor **2016**. Publisher: 'Springer Science and Business Media LLC', doi:10.1007/s11554-016-0613-z.
77. Ong, Z.C.; Noroozi, S. Development of an economic wireless human motion analysis device for quantitative assessment of human body joint **2018**. Publisher: 'Elsevier BV', doi:10.1016/j.measurement.2017.10.056.
78. Porciuncula, F.; Roto, A.V.; Kumar, D.; Davis, I.; Roy, S.; Walsh, C.J.; Awad, L.N. Wearable Movement Sensors for Rehabilitation: A Focused Review of Technological and Clinical Advances. *PM&R* **2018**, 10, S220–S232. _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1016/j.pmrj.2018.06.013>, doi:10.1016/j.pmrj.2018.06.013.
79. Wahyudi, W.; Listiyana, M.S.; Sudjadi, S.; Ngatelan, N. Tracking Object based on GPS and IMU Sensor **2018**.
80. Gregorio, R.D.; Parenti-Castelli, V.; O Connor, J.J.; Leardini, A. Mathematical models of passive motion at the human ankle joint by equivalent spatial parallel mechanisms. *Medical Biological Engineering Computing* **2007**, 45, 305–313. doi:10.1007/s11517-007-0160-7.
81. Alexandru, P. *Structural-Kinematic Modeling of Human Body Ankle Joint Mechanical Systems – Part I*.
82. Isman, R.E.; Inman, V.T.; Poor, P.M. Anthropometric studies of the human foot and ankle. *Bull Prosthet Res* **1969**, 11, 97–129.
83. Drillis, R.; Contini, R.; New York University.; School of Engineering and Science. *Body segment parameters*; New York University, School of Engineering and Science: New York, N.Y., 1966. OCLC: 22352502.
84. Hebbelinck, M.; Ross, W.D. Kinanthropometry and biomechanics. In *Biomechanics IV*; Springer, 1974; pp. 535–552.
85. Fryar, C.D.; Carroll, M.D.; Gu, Q.; Afful, J.; Ogden, C.L. Anthropometric Reference Data for Children and Adults: United States, 2015-2018. *Vital & Health Statistics. Series 3, Analytical and Epidemiological Studies* **2021**, pp. 1–44.
86. Vargas Riaño, J.; Valera, Á.; Agudelo Varela, O. Turmell-metre | 3D CAD Model Library | GrabCAD.