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, Valencia, 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: anthropology; 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]. Biodex™ and Humacnorm™ 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 Assignation

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

![Reference point diagram](image1)

**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.

![Reference point diagram](image2)

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

| 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.
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.

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, A₀, B₀, and C₀ are the vertices from the platform fixed to the foot, and P_M is the triangle’s center. S₁, S₂, and S₃ are fixed to the shank relative to the origin point P₀. M₁ and M₂ define the TC axis; N₁ and N₂ correspond to the ST axis. We define r₁ and r₂ 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].](image)

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:

$$d_m = \|P_0 - P_M\| = \left[ \frac{0.285 - 0.039}{2} + 0.039 \right] \cdot H = 0.162 \cdot H,$$

(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:

$$d_{p12} = \|r_1 - r_2\| = Q,$$

(2)

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

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

(3)

and for the most lateral point is:

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

(4)

The point $M_{1p}$ is the projection of $M_1$ on the sagittal plane; we calculate it from the $P$ and $O$ values.
\[ M_{1p} = (x_{MMP} - P, 0, z_{MMP} - O), \] (5)

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

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

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

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

solving for \( r_1 \) gives the following:

\[ r_1 = M_2 - \frac{M_2 - M_1}{R}. \] (8)

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

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

The distance from the origin \( P_0 \) 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

\[ A_0 = (r_p, 0, -d_m), \] (10)

for \( B_0 \) are:

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

and for \( C_0 \):

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

Where \( r_p \) is proportional to \( H \), then:

\[ r_p = \frac{2}{3} \cdot H. \] (13)

In summary, we estimate \( P_0, r_1, r_2 \); and the platform’s vertices \( A_0, B_0, C_0 \). 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:

\[ g = \begin{bmatrix} R & \tilde{p}_T \\ 0_{1 \times 3} & 1 \end{bmatrix}, \] (14)

where \( R_{3 \times 3} \) is the rotation matrix and \( \tilde{p}_T \) is the translation vector.

For the initial point \( A_0 \):

\[ g_A(0) = \begin{bmatrix} I_{3 \times 3} & \tilde{A}_0 \\ 0_{1 \times 3} & 1 \end{bmatrix}, \] (15)

for \( B_0 \):

\[ g_B(0) = \begin{bmatrix} I_{3 \times 3} & \tilde{B}_0 \\ 0_{1 \times 3} & 1 \end{bmatrix}, \] (16)

and for \( C_0 \)
\[ g_C(0) = \begin{bmatrix} I_{3 \times 3} & \hat{C}_0 \\ 0_{1 \times 3} & 1 \end{bmatrix} \] (17)

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

\[ \hat{\omega}_1 = \frac{M_2 - M_1}{\|M_2 - M_1\|}, \] (18)

and a directed vector \( \hat{r}_1 \) from \( P_O \) to \( r_1 \) is:

\[ \hat{r}_1 = r_1 - P_O, \] (19)

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

\[ \hat{\nu}_{\theta_1r_2} = -\hat{\omega}_1 \times \hat{r}_1, \] (20)

together, \( \hat{\omega}_1 \) and \( \hat{\nu}_{\theta_1r_2} \) compound the six-dimensional vector \( \hat{\xi}_1 \):

\[ \hat{\xi}_1 = \begin{pmatrix} \hat{\nu}_{\theta_1} \\ \hat{\omega}_1 \end{pmatrix}. \] (21)

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

\[ \hat{\omega}_2 = \frac{N_2 - N_1}{\|N_2 - N_1\|}, \] (22)

\[ \hat{r}_2 = r_2 - P_O, \] (23)

\[ \hat{\nu}_2 = -\hat{\omega}_2 \times \hat{r}_2, \] (24)

and:

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

We compute \( R \) for each joint \( i = 1, 2 \) from the Rodrigues’ formula:

\[ e^{(\Omega_i \theta_i)} = I_{3 \times 3} + \Omega_i \sin \theta_i + \Omega_i^2 (1 - \cos \theta_i), \] (26)

where \( \Omega \) is the skew symmetric matrix:

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

The exponential formula is:

\[ e^{\hat{\xi}_i \theta_i} = \begin{bmatrix} 0 & \omega_{yi} & -\omega_{xi} \\ \omega_{zi} & 0 & -\omega_{yi} \\ -\omega_{zi} & \omega_{xi} & 0 \end{bmatrix} \hat{\xi}_i \] (28)

and, \( \tau_i \) is translation vector:

\[ \tau_i = (I_{3 \times 3} - e^{\hat{\xi}_i \theta_i}) \hat{\omega}_i \times \hat{\nu}_i + \hat{\omega}_i \hat{\omega}_i^T \hat{\nu}_i \theta_i \] (29)

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

\[ g_A = e^{\hat{\xi}_1 \theta_1 \hat{\xi}_2 \theta_2} g_A(0) = \begin{bmatrix} R & \hat{R}_A \\ 0 & 1 \end{bmatrix}, \] (30)
where $\hat{p}_A$ is the instantaneous position vector of A, the PoE for B:

$$g_B = e^\xi_1 \theta_1 \xi_2 \theta_2 \hat{g}_B(0) = \begin{bmatrix} R & \hat{p}_B \\ 0 & 1 \end{bmatrix},$$  \hspace{1cm} (31)$$

and the PoE for C is:

$$g_C = e^\xi_1 \theta_1 \xi_2 \theta_2 \hat{g}_C(0) = \begin{bmatrix} R & \hat{p}_C \\ 0 & 1 \end{bmatrix},$$  \hspace{1cm} (32)$$

$\theta_1$ is the TC rotation angle from the zero position, and $\theta_2$ 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 PO in Figure 6.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{Figure6.png}
\caption{Vectors and points on the sagittal plane.}
\end{figure}

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 PO, A0, B0, C0, r1, r2, and the surfaces representing each group of movements. The forward kinematics with $\theta_{\text{range}} = [\theta_2\text{range}] = [-15, 15]$ and $\theta_1 = \theta_2 = 10^\circ$ is in Figure 7(b). For $\theta_{\text{range}} = [\theta_2\text{range}] = [-10, 10]$ 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.
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.

By considering the distances between the origin and the vertices, we estimate the DWS maximal length in every module.
\[ l_{\text{max}} = \max[\|p_A(\theta_1, \theta_2) - A\| + r_m] \] (33)

Here, \( l_{\text{max}} \) is the maximal possible length from the triangular inequality, \( p_A \) is the positions group in \( g_A \), \( r_m \) 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.

PO and PM 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 $\alpha$ from the XY plane to a normal vector $\hat{n}_{APB}$:

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

and, the angle $\alpha$ is:

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

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 $d_{APB_p} = \|B_p - A_p\|$. The rotation axis is in the direction $B_1 - B_3$. The $Bpr$ is $Bps$ rotated $\alpha$ in angle about this axis. In Fig 11(b), we show how to find the $Bpr$ apex, similarly to
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.

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:

$$d_{CpBp} = \|B_{ps} - C_{ps}\|.$$  \hspace{1cm} (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 220\( \Omega \) per turn. The diameter is \( D = 3.8cm \), the spring could be compressed in four turns. The maximal length is as follows:

\[
l_{\text{max}} = 4 \cdot D \cdot \pi
\]

(38)

Which is 47.75 cm approximately, this value is greater than \( l_{\text{max}} \) 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:

\[ A_v = \frac{V_o}{V_i} = \left[ 1 + \frac{R_2}{R_1} + \frac{2R_2}{R_1} \right], \]  

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

\[ V_o = A_v \cdot V_i = 4.794\,\text{V} \]
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
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:

$$l_{iMj} = d_{iMj} + \frac{m_{iMj}}{s_{iMj}}$$

(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, actives 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).

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 $M_1M_2$ segment is $r_1$. We estimate the initial TC axis approximation with:

$$M_{10} = MLP - [L, 0, K], \quad (42)$$

and

$$M_{20} = MMP - [-P, 0, O]. \quad (43)$$

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

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

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 $\omega_1$ by registering several trajectories. For each trajectory, we have a list of data points $P = [x, y, z]$, which pertain to a plane:
\[ ax + by + cz + d = 0, \]  
\[ \text{or in all n data is in a } n \times 3 \text{ matrix, solving for } z \text{ we have} \]
\[
\begin{bmatrix}
  x_0 & y_0 & 1 \\
  x_1 & y_1 & 1 \\
  \vdots & \vdots & \vdots \\
  x_{n-1} & y_{n-1} & 1
\end{bmatrix}
\begin{bmatrix}
  a \\
  b \\
  c
\end{bmatrix}
= 
\begin{bmatrix}
  z_0 \\
  z_1 \\
  \vdots \\
  z_{n-1}
\end{bmatrix}
\]  
\[ (47) \]

which has the form:
\[ Ax = B \]  
\[ \text{there are more equations, the pseudo inverse is } A^+ = (A^T A)^{-1} A^T, \text{ and the normal vector is:} \]
\[
\begin{bmatrix}
  a \\
  b \\
  c
\end{bmatrix}
= (A^T A)^{-1} A^T B
\]  
\[ (49) \]

By using the Rodrigues’ formula we found the rotated points to the XY plane by the product \( \mathbf{v} = \mathbf{n} \times \hat{k} \), with \( \hat{k} = [0, 0, 1]^T \)
\[ \mathbf{P}_r = \mathbf{P} \cos(\theta) + (\mathbf{v} \times \mathbf{P}) \sin(\theta) + \mathbf{v}(\mathbf{v} \cdot \mathbf{P}) (1 - \cos \theta). \]  
\[ (50) \]

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

The circle in XY plane can be rearranged by
\[ (x - x_c)^2 + (y - y_c)^2 = r^2 \]  
\[ (2x_c)x + (2y_c)y + (r^2 - x_c^2 - y_c^2) = x^2 + y^2 \]
\[ c_0x + c_1y + c_2 = x^2 + y^2 \]  
\[ (51) \]

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, \( \mathbf{P}_r \) we have a linear system
\[
\begin{bmatrix}
  x_0 & y_0 & 1 \\
  x_1 & y_1 & 1 \\
  \vdots & \vdots & \vdots \\
  x_{n-1} & y_{n-1} & 1
\end{bmatrix}
\begin{bmatrix}
  c_0 \\
  c_1 \\
  c_2
\end{bmatrix}
= 
\begin{bmatrix}
  x_0^2 + y_0^2 \\
  x_1^2 + y_1^2 \\
  \vdots \\
  x_{n-1}^2 + y_{n-1}^2
\end{bmatrix}
\]  
\[ (52) \]

has the form
\[ Ac = b \]  
\[ (53) \]

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

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

We found the center point \( \mathbf{C}_p = [x_c, y_c] \) and radius \( r \) by solving:
\[ 2x_c = c_0 \]  
\[ 2y_c = c_1 \]  
\[ r^2 - x_c^2 - y_c^2 = c_2 \]  
\[ (55) \]

Finally, we apply a rotation to the center in respect to the original plane. This point pertain 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 $\omega_1$, $\omega_2$ 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.](image)

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:

$$\hat{p}_l = \hat{l}_0 + \hat{l}d,$$  \hspace{1cm} (56)

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

$$(\rho - \rho_0) \cdot (n_\rho) = 0,$$  \hspace{1cm} (57)
where $p_0$ is a known point in the plane, and $n_p$ is the plane’s normal vector. solving for $d$, gives:

$$
d = (p_0 - l_0) \cdot \hat{n},
$$

and replacing in the TC axis line equation:

$$r_1 = c_1 + \omega_1 d$$

where $r_1$ is the TC axis intersection with the sagittal plane. The point $c_1$ is the center, and the axis direction $\omega_1$, both were found by circle fitting. Also, packing in six dimensional Plücker line coordinates, we have:

$$m_1 = r_1 \times \omega_1,$$

and the $l_1$ six dimensional vector is:

$$l_1 = [\omega_{1x} : \omega_{1y} : \omega_{1z} : m_{1x} : m_{1y} : m_{1z}]$$

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.
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.
The Table 2 shows the calibration results.

| 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.

| Sensor ID | l1M1 | l2M1 | l3M1 |
|-----------|------|------|------|
| SW measure | 7.622 | 5.33 | 6.384 |
| Error in cm | 0.38 | -0.030 | 0.52 |
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 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 | l1M1 | l2M1 | l3M1 | l1M2 | l2M2 | l1M3 | l2M3 | 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)  |

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](image)

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, $r_1, r_2, \omega_1, \omega_2$, and the origin established in the center of the base modules. We apply the equation:

\[ \hat{r}_1 = \tau_0 + \hat{n}_p \cdot d \] (62)

where $\tau_0$ 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 | $r$           | $\omega$          |
|------|---------------|---------------|---------------|-------------------|
| 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](image)

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 tho 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.
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:

| Abbreviation | Description |
|--------------|-------------|
| HAJ          | 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.
In Fig. A1, 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 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.
However, the mathematical model of the ankle is a representation of two hinge joints in series, as presented in Fig. A4.

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/](https://sagecell.sagemath.org/).
# degrees to radians
alpha = a * pi / 180
na = (1 - cos (alpha))
cia = cos (alpha)
sa = sin (alpha)

# rotation matrix
R = matrix ([[u[0] ^ 2 * na + ca, u[0] * u[1] * na - u[2] * sa, u[0] * u[2] * na + u[1] * sa],
[u[0] * u[1] * na + u[2] * sa, u[1] ^ 2 * na + ca, u[1] * u[2] * na - u[0] * sa],
[u[2] * u[0] * na - u[1] * sa, u[2] * u[1] * na + u[0] * sa, u[2] ^ 2 * na + ca]])
return R

# Talocrural direction
vTC = rot_ax_ang ([0, 0, 1], -6) * rot_ax_ang ([1, 0, 0], 80) * vector ([0, 0, 1])

# Subtalar direction
vST = rot_ax_ang ([0, 0, 1], 23) * rot_ax_ang ([0, 1, 0], 41) * vector ([1, 0, 0])

# Anthropometric values
vK = 12; vL = 11; vO = 16; vP = 1; vQ = 5; vR = 0.54 # anthropometric measurements
H = 1750 # heigth in mm
d_m = (0.285 - 0.039) * H / 2 # knee-ankle half distance
d_p = 0.039 * H # ankle-foot
z_p = (d_m + d_p) # z initial position
r_p = 0.055 * H ^ 2 / 3 # platform radius
ae = 2 * pi / 3 # auxiliary angle
PO = vector ([0, 0, 0]) # Origin at middle shank

# initial platform position
A0 = vector ([r_p, 0, z_p])
B0 = vector ([r_p * cos (ae), r_p * sin (ae), z_p])
C0 = vector ([r_p * cos (-ae), r_p * sin (-ae), z_p])
PM0 = (A0 + B0 + C0) / 3 # center of the platform

# marker representation
dot = point3d ((0, 0, 0), size = 5, color = 'red', opacity = .5)
dPO = dot.translate (PO)
dA0 = dot.translate (A0)
dB0 = dot.translate (B0)
dC0 = dot.translate (C0)
dPM0 = dot.translate (PM0)

r_1 = vector ([0, 0, -d_m]) # intersection point between the talocrural axis and the sagittal plane
ap = 0.039 * H
lp = 0.152 * H

# points on the malleolar medial and lateral
M_1 = r_1 - ap * vR * vTC
M_2 = r_1 + ap * (1 - vR) * vTC
tht = 41 * pi / 180

# intersection between the subtalar axis and the sagittal plane
r_2 = vQ * vector ([ - cos (tht), 0, - sin (tht) ]) + r_1

# points from the hindfoot to the midfoot
N_1 = r_2 + 0.6 * lp * vR * vST
N_2 = r_2 - 0.3 * lp * vR * vST

# representation of the kinematic chain
lrt = line ([PO, r_1])
lm12 = line ([M_1, M_2])
ln12 = line ([N_1, N_2])
lrA = line ([r_2, A0])
lrB = line ([r_2, B0])
lrC = line ([r_2, C0])

# markers representation
dr1 = dot.translate (r_1)
dr2 = dot.translate (r_2)
dM1 = dot.translate (M_1)
dM2 = dot.translate (M_2)
\textbf{dN1} = \text{dot} . \text{translate}(N_1)
\textbf{dN2} = \text{dot} . \text{translate}(N_2)

reference labels
\textbf{plbl} = \text{vector} ([0,0,8])
\textbf{lbp} = \text{text3d} ('PO', \text{PO}+ \text{plbl})
\textbf{lbp} += \text{text3d} ('A0', \text{color} = 'black'). \text{translate}(A0 + \text{plbl})
\textbf{lbp} += \text{text3d} ('B0', \text{color} = 'black'). \text{translate}(B0 + \text{plbl})
\textbf{lbp} += \text{text3d} ('C0', \text{color} = 'black'). \text{translate}(C0 + \text{plbl})
\textbf{lbp} += \text{text3d} ('PM0', \text{color} = 'black'). \text{translate}(PM0 + \text{plbl})

axes labels
\textbf{lbf} = \text{text3d} ('M1', M_1 + \text{plbl})
\textbf{lbf} += \text{text3d} ('M2', M_2 + \text{plbl})
\textbf{lbf} += \text{text3d} ('N1', N_1 + \text{plbl})
\textbf{lbf} += \text{text3d} ('N2', N_2 + \text{plbl})
\textbf{lbf} += \text{text3d} ('r1', r_1 + \text{plbl})
\textbf{lbf} += \text{text3d} ('r2', r_2 + \text{plbl})

# finding the twist unitary vectors
w1 = \text{n}((M_1 - M_2) / \text{abs}(M_1 - M_2))
w2 = \text{n}((N_1 - N_2) / \text{abs}(N_1 - N_2))

the perpendicular component
v1 = \text{n}(-w1. \text{cross}_\text{product}(r_1))
v2 = \text{n}(-w2. \text{cross}_\text{product}(r_2))

angles for the talocrural and subtalar rotations
theta1 = \text{var} ('theta_1')
theta2 = \text{var} ('theta_2')

six dimensional vector xi mapping
\textbf{xi1} = \text{matrix} ([v1[0], v1[1], v1[2], w1[0], w1[1], w1[2]]) \text{. transpose()}
\textbf{xi2} = \text{matrix} ([v2[0], v2[1], v2[2], w2[0], w2[1], w2[2]]) \text{. transpose()}

transformation of exponential matrix of rotation
Rexp1 = \text{rot ax ang}(w1, theta1)

rotation matrix component of the homogeneous transformation
vexp1 = (\text{matrix} . \text{identity}(3) - Rexp1) * (w1. \text{cross}_\text{product}(v1))

conformation of the homogeneous transformation matrix
MTH1 = (Rexp1 \text{. augment}(vexp1)). \text{stack}([0,0,0,1])

components for the subtalar axis
Rexp2 = \text{rot ax ang}(w2, theta2)
vexp2 = (\text{matrix} . \text{identity}(3) - Rexp2) * (w2. \text{cross}_\text{product}(v2))
MTH2 = (Rexp2 \text{. augment}(vexp2)). \text{stack}([0,0,0,1])

transformation matrix representing the initial position
gst0A = \text{matrix}([[1,0,0,A0[0]], [0,1,0,A0[1]], [0,0,1,A0[2]], [0,0,0,1]])
gst0B = \text{matrix}([[1,0,0,B0[0]], [0,1,0,B0[1]], [0,0,1,B0[2]], [0,0,0,1]])
gst0C = \text{matrix}([[1,0,0,CO[0]], [0,1,0,CO[1]], [0,0,1,CO[2]], [0,0,0,1]])
gst0PM = \text{matrix}([[1,0,0,PM0[0]], [0,1,0,PM0[1]], [0,0,1,PM0[2]], [0,0,0,1]])

product of exponential matrices for all the points
MTHA = MTH1 * MTH2 * gst0A
MTHB = MTH1 * MTH2 * gst0B
MTHC = MTH1 * MTH2 * gst0C
MTHPM = MTH1 * MTH2 * gst0PM

components of the group of rigid movements for the central point
f_xpm = MTHPM [0][3]
f_ypm = MTHPM [1][3]
f_zpm = MTHPM [2][3]

orthogonal direction vectors
f_xpm = \text{vector}(\text{MTHPM} [0][0], \text{MTHPM} [1][0], \text{MTHPM} [2][0]))
f_ypm = \text{vector}(\text{MTHPM} [0][1], \text{MTHPM} [1][1], \text{MTHPM} [2][1]))
f_zpm = \text{vector}(\text{MTHPM} [0][2], \text{MTHPM} [1][2], \text{MTHPM} [2][2]))
components of the three vertices of the platform

\[ f_xA = MTHA[0][3] \]

\[ f_yA = MTHA[1][3] \]

\[ f_zA = MTHA[2][3] \]

\[ f_xB = MTHB[0][3] \]

\[ f_yB = MTHB[1][3] \]

\[ f_zB = MTHB[2][3] \]

\[ fxC = MTHC[0][3] \]

\[ f_yC = MTHC[1][3] \]

\[ f_zC = MTHC[2][3] \]

saving symbolic expressions to file output

\[ tx, ty = \text{var}('t_x', 't_y') \]

\[ pmx = f_xpm \text{. subs (theta1 ==tx, theta2 == ty)} \]

\[ pmy = f_ypm \text{. subs (theta1 ==tx, theta2 == ty)} \]

\[ pmz = f_zpm \text{. subs (theta1 ==tx, theta2 == ty)} \]

\[ ax = f_xA \text{. subs (theta1 ==tx, theta2 == ty)} \]

\[ ay = f_yA \text{. subs (theta1 ==tx, theta2 == ty)} \]

\[ az = f_zA \text{. subs (theta1 ==tx, theta2 == ty)} \]

\[ bx = f_xB \text{. subs (theta1 ==tx, theta2 == ty)} \]

\[ by = f_yB \text{. subs (theta1 ==tx, theta2 == ty)} \]

\[ bz = f_zB \text{. subs (theta1 ==tx, theta2 == ty)} \]

\[ cx = f_xC \text{. subs (theta1 ==tx, theta2 == ty)} \]

\[ cy = f_yC \text{. subs (theta1 ==tx, theta2 == ty)} \]

\[ cz = f_zC \text{. subs (theta1 ==tx, theta2 == ty)} \]

\[ \text{exportg} = [\text{[pmx, pmy, pmz], [ax, ay, az], [bx, by, bz], [cx, cy, cz], [A0.n(), B0.n(), C0.n(), PM0.n()]], [r_1.n(), r_2.n(), N_1.n(), N_2.n(), M_1.n(), M_2.n()]] \]

\[ \text{with open("output.txt", "w") as f: f.write(str(exportg))} \]

scaled position vectors

\[ ar1 = \text{arrow3d}(P0, r_1, 40, \text{color='cyan'}) \]

\[ ar2 = \text{arrow3d}(P0, r_2, 40, \text{color='magenta'}) \]

\[ av1 = \text{arrow3d}(r_1, r_1 + 40*\text{w1}/\text{abs(r_1)}, 40, \text{color='red'}) \]

\[ av2 = \text{arrow3d}(r_2, r_2 + 40*\text{w2}/\text{abs(r_2)}, 40, \text{color='blue'}) \]

\[ scrs = ar1 + ar2 + av1 + av2 \]

platform initial position

\[ pltf = \text{polygon([A0.n(), B0.n(), C0.n(), PM0.n()]), color='gray', opacity=0.7} \]

showing the initial position

\[ gr = dP0 + dA0 + dB0 + dC0 + dPM0 \]

\[ gr += lrt + lm12 + ln12 + lrA + lrB + lrC \]

\[ gr += dM1 + dM2 + dr1 + dr2 + dN1 + dN2 \]

\[ gr += lbp + lbf + scrws + pltf \]

\[ gr.\text{show(aspect_ratio=1, frame=true, figsize=(1024,1024), projection='orthographic', axes=true)} \]

changing the angular parameters interactively

@interact
\[ \text{def _(t1min=slider(-90,-1,1,label='min \theta_1', default=-15), t1max=slider(1,90,1,label='max \theta_1', default=15), t2min=slider(-90,-1,1,label='min \theta_2', default=-15), t2max=slider(1,90,1,label='max \theta_2', default=15))} : \]

@interact
\[ \text{def _(txi=slider(t1min,t1max, step_size=1, label='Value \theta_1', default=0), tyi=slider(t2min,t2max, step_size=1, label='Value \theta_2', default=0))} : \]

(substitute values

\[ pmx = pmx\text{.subs(ty==txi, ty==tyi)} \]

\[ pmy = pmy\text{.subs(ty==txi, ty==tyi)} \]

\[ pmz = pmz\text{.subs(ty==txi, ty==tyi)} \]

\[ ax = ax\text{.subs(ty==txi, ty==tyi)} \]

\[ ay = ay\text{.subs(ty==txi, ty==tyi)} \]

\[ az = az\text{.subs(ty==txi, ty==tyi)} \]

\[ bx = bx\text{.subs(ty==txi, ty==tyi)} \]

\[ by = by\text{.subs(ty==txi, ty==tyi)} \]

\[ bz = bz\text{.subs(ty==txi, ty==tyi)} \]
bzs = bz.subs(tx==txi, ty==tyi)
cxs = cx.subs(tx==txi, ty==tyi)
cys = cy.subs(tx==txi, ty==tyi)
czs = cz.subs(tx==txi, ty==tyi)
# plotting
dp = 10  # data points
cpm = parametric_plot3d([f_xpm, f_ypm, f_zpm], (theta_1, t1min, t1max),
    (theta_2, t2min, t2max), plot_points=[dp, dp])
cA = parametric_plot3d([f_xA, f_yA, f_zA], (theta_1, t1min, t1max),
    (theta_2, t2min, t2max), plot_points=[dp, dp])
cB = parametric_plot3d([f_xB, f_yB, f_zB], (theta_1, t1min, t1max),
    (theta_2, t2min, t2max), plot_points=[dp, dp])
cC = parametric_plot3d([fxCB, f_yC, f_zC], (theta_1, t1min, t1max),
    (theta_2, t2min, t2max), plot_points=[dp, dp], texture="red")

# create vectors
ptpm = vector([pmxs, pmys, pmzs])
spms = vector([f_spm[0].subs(theta1==txi, theta2==tyi),
    f_spm[1].subs(theta1==txi, theta2==tyi),
    f_spm[2].subs(theta1==txi, theta2==tyi)]).n()
apms = arrow3d(ptpm, ptpm + 40*spms, 40, color='red')
spmn = vector([f_npm[0].subs(theta1==txi, theta2==tyi),
    f_npm[1].subs(theta1==txi, theta2==tyi),
    f_npm[2].subs(theta1==txi, theta2==tyi)]).n()
appm = arrow3d(ptpm, ptpm + 40*spmn, 40, color='green')
spma = vector([f_apm[0].subs(theta1==txi, theta2==tyi),
    f_apm[1].subs(theta1==txi, theta2==tyi),
    f_apm[2].subs(theta1==txi, theta2==tyi)]).n()
apma = arrow3d(ptpm, ptpm + 40*spma, 40, color='blue')
pta = vector([axs, ays, azs]).n()
ptb = vector([bxs, bys, bzs]).n()
ptc = vector([cxs, cys, czs]).n()
ptf = polygon([pta, ptb, ptc], color='green', opacity=0.7)

# plotting
fkin = dPO + apma + apmn + apms + ptf + lbp + cA + cB + cC + cpm
fkin.show(aspect_ratio=1, frame=true, figsize=(1024, 1024),
    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/

settings.outformat="pdf";
settings.prc = false;
settings.render = 0;
import three;
import graph3;
size(200, 0);
currentprojection = orthographic(30, 30, 30);
xaxis3("$x$", 0, 100, red, OutTicks(2,2));
yaxis3("$y$", 0, 100, red, Out_ticks(2,2));
zaxis3("$z$", -300, 0, red, OutTicks(2,2));

// range of motion
real tcmmin = -180, tcmmax = 180, stmmin = -180, stmax = 180;
// actual position
real theta1 = 0, theta2 = 0;

// initial points
po = (0, 0, 0);
 triple a0 = (64.1666666666667, 0.000000000000000, -283.5000000000000);
 triple b0 = (-32.0833333333333, 55.5699634095015, -283.5000000000000);
 triple c0 = (-32.0833333333333, -55.5699634095015, -283.5000000000000);
 triple pm0 = (0.000000000000000, 0.000000000000000, -283.5000000000000);
 triple ri = (0.000000000000000, 0.000000000000000, -215.2500000000000);
 triple r2 = (3.77354790111386, 0.000000000000000, -218.5302951449534);
 triple n1 = (56.0996689565118, 25.4146727215185, -275.0720864994700);
 triple n2 = (-33.7101563299267, -12.7073636075920, -190.2593994676941);
triple m1 = (3.79386995439788, 36.0962614380663, -221.649803587915);
triple m2 = (-3.23181514633894, -30.7486671509453, -209.798315462147);

// parametric functions
triple pm(pair t) {
    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.x + 1/180* pi*t.y) + 5.958036183564665* sin (1/180* pi*t.x + 1/180* pi*t.y) - 2.477249693938664* sin ( -1/180* pi*t.x + 1/180* pi*t.y) - 1.409626687020189* sin (1/180* pi*t.x) + 21.19359495348665* sin (1/180* pi*t.x) + 30.070693601917178* sin (1/180* pi*t.x) - 6.914891650849512* cos (1/180* pi*t.x + 1/180* pi*t.y) - 15.774905204951304* sin (1/180* pi*t.x) + 1.69957489375312* cos (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.x) - 219.00758680507846) ;
}

// initial position
path3 iplatform = (a0 --b0 --c0 --cycle);
draw (po --r1 ^^po --r2 ^^n1 --n2 ^^m1 --m2);
draw ( surface ( iplatform ),green + opacity (0.3) );
dot (po --a0 --b0 --c0 --pm0 --r1 --r2 --n1 --n2 --m1 --m2 , 3+ red );
label ("\text{ P_O }",po ,NE);
label ("\text{ A_0 }",a0 ,2 SW);
label ("\text{ B_0 }",b0 ,3 SE);
label ("\text{ C_0 }",c0 ,NW);
label ("\text{P_{M0 }}",pm0 ,NE);
label ("\text{ N_1 }",n1 ,NE);
label ("\text{ N_2 }",n2 ,NE);
label ("\text{ M_1 }",m1 ,NE);
label ("\text{ M_2 }",m2 ,NE);
label ("\text{ r_1 }",r1 ,NW);
label ("\text{ r_2 }",r2 ,SE);

// compute surface
pen p= rgb (0.2 ,0.5 ,0.7) + opacity (0.2) ;
surface spm = surface (pm ,( tcmin , stmin ) ,( tcmax , stmax ) ,36 ,36 , Spline );
draw (spm , lightgray + opacity (0.1) , meshpen =p, render ( merge = true ));

// surface & mesh
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.

from sage.plot.plot3d.transform import rotate_arbitrary
from sage.plot.plot3d.plot3d import axes
# centroid of the base
cb=vector([116.666,97.905,0])
angrot = -(79.91/2)*pi/180
mrotz = rotate_arbitrary((0.0,0.0,1.0),angrot)
# base sensor modules centerpoints
A= vector([0,0,0])
A= mrotz*A
Ap0 = vector([30.0,0.67,50.5]) + A # initial platform A point
bl0 = text3d("A",A+vector([10,10,10]))
B=vector([52.268,293.716,0])
B= mrotz*B
Bp0 = vector([-26.4,-92.6,-50.5]) + B
bl1 = text3d("B",B+vector([10,10,10]))
C= vector([297.73,0,0])
C= mrotz*C
Cp0 = vector([-25.67,90.99, 50.5]) + C

from sage.plot.plot3d.transform import rotate_arbitrary
from sage.plot.plot3d.plot3d import axes
# centroid of the base
cb=vector([116.666,97.905,0])
angrot = -(79.91/2)*pi/180
mrotz = rotate_arbitrary((0.0,0.0,1.0),angrot)
# base sensor modules centerpoints
A= vector([0,0,0])
A= mrotz*A
Ap0 = vector([30.0,0.67,50.5]) + A # initial platform A point
bl0 = text3d("A",A+vector([10,10,10]))
B=vector([52.268,293.716,0])
B= mrotz*B
Bp0 = vector([-26.4,-92.6,-50.5]) + B
bl1 = text3d("B",B+vector([10,10,10]))
C= vector([297.73,0,0])
C= mrotz*C
Cp0 = vector([-25.67,90.99, 50.5]) + C
# base points

0.5

tet = 32.76  # radius of the tetrahedron base

bl1 = text3d('C', C + vector([10, 10, 10]))

A1 = bt1A + A

bl1 += text3d('A1 ', A1 + vector([10, 10, 10]))

A2 = bt2A + A

A3 = bt3A + A

bt1B = vector([32.52, 3.92, 0])

bt2B = vector([-19.66, 26.21, 0])

bt3B = vector([-12.87, -30.13, 0])

B1 = bt1B + B

bl1 += text3d('B1 ', B1 + vector([10, 10, 10]))

B2 = bt2B + B

B3 = bt3B + B

bt1C = vector([27.44, -17.9, 0])

bt2C = vector([1.79, 32.71, 0])

bt3C = vector([-29.42, -14.81, 0])

C1 = bt1C + C

bl1 += text3d('C1 ', C1 + vector([10, 10, 10]))

C2 = bt2C + C

C3 = bt3C + C

t1 = polygon([A1, A2, A3, A1], color='red')

t2 = polygon([B1, B2, B3, B1], color='red')

t3 = polygon([C1, C2, C3, C1], color='red')

tp = polygon([Ap0, Bp0, Cp0, Ap0])

l1 = line([Ap0, A1])

l2 = line([Ap0, A2])

l3 = line([Ap0, A3])

l4 = line([Bp0, B1])

l5 = line([Bp0, B3])

l6 = line([Cp0, C1])

l7 = line([Cp0, C2])

tb = line([A, B, C, A]) + bl1 + axes(100) + t1 + t2 + t3 + tp + l1 + l2 + l3 + l4 + l5 + l6 + l7 + tb

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.; Hägglund, 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. Bulathsinghala, 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:10.1007/978-0-387-47415-8.
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.
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, J.; 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. 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* 2002, 35, 83–91. doi:10.1016/S0021-9290(99)00022-6.

50. Bruening, D.; Richards, J. *Foot & Ankle International* 1992, 13, 439–446. doi:10.1177/107110079201300802.

51. Leynard, A.; O’Connor, J.; Catani, F.; Giannini, S. A geometric model of the human ankle joint. *Journal of Biomechanics* 2009, 42, 585–591. doi:10.1016/S0021-9290(99)00022-6.

52. Brockett, C.L.; Chapman, G.J. Biomechanics of the ankle. *Orthopaedics and Trauma* 2006, 30, 232–238. doi:10.1016/j.ajot.2006.04.015.

53. Bruneing, D.; Richards, J. *Optimal ankle axis position for articulated boots*.; Vol. 4, 2005.

54. Leitch, J.; Stebbins, J.; Zavatsky, A.B. Subject-specific axes of the foot joint complex. *Journal of Biomechanics* 2010, 43, 2923–2928. doi:10.1016/j.jbiomech.2010.07.007.

55. Papp, W.C.H.; Chatterjee, H.J.; Soligo, C. Calculating the axes of rotation for the subtalar and talocural joints using 3D bone reconstructions. *Journal of Biomechanics* 2012, 45, 1103–1107. doi:10.1016/j.jbiomech.2012.01.011.

56. Nichols, J.A.; Roach, K.E.; Fiorentino, N.M.; Anderson, A.E. Predicting tibial 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.

57. 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.

58. Spooner, S.; Kirby, K. The subtalar joint axis location: A preliminary report; Vol. 96, 2006. doi:10.7547/0960212.

59. Lewis, G.S.; Kirby, K.A.; Piazza, S.J. Determination of subtalar joint axis location by restriction of talocural joint motion. *Gait & Posture* 2007, 25, 63–69. doi:10.1016/j.gaitpost.2006.01.001.

60. 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.

61. De Schepper, J.; Van Alsenoy, K.; Rijckaert, J.; De Muts, 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.

62. Van Alsenoy, K.; De Schepper, J.; Santos, D.; Vereecke, E.; D’Aouèt, 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.

63. Van Alsenoy, K.K.; D’Aouèt, 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.

64. Krähenbühl, N.; Horn-Lang, T.; Hintermann, B.; Knupp, M. The subtalar joint: A complex mechanism. *EFORT open reviews* 2017, 2, 309–316.

65. Jastifer, J.R.; Gustafson, P.A. The subtalar joint: biomechanics and functional representations in the literature. *The Foot* 2014, 24, 203–209.

66. 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:10.1109/73.4406.

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, Z. 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.pmjr.2018.06.013, doi:10.1016/j.pmjr.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.