

Capstone Project I Report

BioInspired Exoskeleton for Rehabilitation of Lower Parts of the Limb

Alen Smailov and Nurlybek Smitov

A thesis submitted in part fulfilment of the degree of

BS in Robotics Engineering

Supervisor: Aibek Nietkaliev

Instructor: Togzhan Syrymova



Department of Robotics and Mechatronics
School of Engineering and Digital Sciences
Nazarbayev University
May 4, 2026

Table of Contents

Abstract	3
1 Introduction	4
2 Literature Review & Existing Solutions Analysis	6
3 Biomechanical Modeling of the knee (Locating the ICR)	12
3.1 Modeling Assumptions	12
3.2 Knee Kinematic Equations from Literature	12
3.3 MATLAB Implementation	13
4 Crossed four-bar knee mechanism kinematics	16
4.1 Geometry and notation	16
4.2 Position analysis of the crossed four-bar mechanism	17
4.3 Optimisation-based parameter tuning	18
4.4 Outputs and visualisation	19
5 Exoskeleton Knee CAD Design	22
5.1 Exoskeleton knee CAD design	22
5.2 Physical prototype implementation	28
6 Running the Exoskeleton	37
6.1 Hardware setup	37
6.2 Control software	39
6.3 Test protocol and observations	40
7 Evaluation Methodology	41
7.1 Anatomical reference trajectory	41
7.2 Mechanism trajectory	41
7.3 Misalignment metric for the mechanism	42
7.4 Hinge baseline and misalignment	42
8 Results	44
8.1 Trajectory comparison	44

8.2	Misalignment versus flexion	47
8.3	Configuration continuity check	48
8.4	Experimental motor-control results	48
8.5	Summary of findings	50
9	Discussion	51
9.1	Quality of ICR matching	51
9.2	From kinematics to a working prototype	52
9.3	Limitations and simplifications	52
10	Conclusion and Future Work	55
A	Motor-Control Firmware	57

Abstract

Exoskeleton for lower parts of the limb, specifically knee, are increasingly used for both gait assistance and rehabilitation. After surviving car crashes, stroke, other reasons causing dysfunction of the knee joint, or just getting old significantly harms the ability to make flexion and extension movements of the knee. The purpose of the exoskeleton is to help patients by providing continuous treatment which helps making these movements, developing muscles, and enhancing neural connections. However, many existing devices have one common problem: they do not consider the polycentric behavior of the knee joint. Existing design solutions still model the tibiofemoral joint, the joint that connects two bones: Tibia and Femur, as a simple hinge with a fixed center of rotation. However, in reality, the human knee has a moving Instantaneous Center of Rotation (ICR), whose position is dependent on the Flexion angle of the knee. The movement of the ICR in our model is analyzed only in the sagittal plane, forward Z Axis plane. The resulting mismatch between the shifted ICR and center of rotation of the hinge results in joint misalignment, uncomfortable interface forces and reduced effectiveness of assistance.

This thesis investigates a bioinspired knee joint mechanism based on a four bar linkage system that also has polycentric ICR. First, the sagittal-plane kinematic model of the human knee is analyzed to establish the locations of the anatomical ICR over the range of F. This model is then used in order to compute the reference ICR trajectory which will be used for optimization of the mechanism. Then, a planar cross four bar linkage mechanism is proposed as the exoskeleton knee joint. The geometry, notations, and mathematical model of the system is defined in the paper. The kinematic equations are derived from the loop closure condition and are expressed in the form that is suitable for numerical solution. Using these equations, the mechanical ICR trajectory of the linkage is computed as well for the same flexion range.

The mechanical ICR path of the cross four bar linkage system is quantitatively compared with the anatomical reference trajectory and with the trajectory implied by a simple hinge joint. The results will demonstrate effectiveness of the proposed linkage, while remaining mechanical simple and compatible with rotatory actuators. Finally, a preliminary CAD model of the exoskeleton is demonstrated in the paper.

Chapter 1: Introduction

Research Problem

The disfunction of the lower parts of the limb specifically the knee is widespread problem, a lot of people suffer nowadays. The reasons include various syndromes, illnesses, and injuries. However, the most popular, as well as with the most severe symptoms cause is the stroke. In one year, there are nearly 9 million stroke survivors [1]. After surviving stroke it is essential to provide help, to facilitate rehabilitation. Recovery of lower limb motor function depends on the reorganization of brain function and activation of cortical excitability in damaged brain areas, [2] . In order to recover people's muscle movement and their gaits, the exoskeleton which assists with knee flexion movement.

Research Motivation

Most of the existing conventional approaches, such as wheelchairs and crutches, have failed to provide complete independence to the users [3]. Therefore it is essential to create an exoskeleton, which will be effective, transparent, and not difficult to build at the same time.

This project will focus on development of exoskeleton model. There are many exoskeleton solutions , [3], [4], [5], [6], some of which considered biomechanical structure of the knee joint, [7], [8], [9], [10], [11]. Later in the article, the solutions which considered biomechanics will be analyzed and compared.

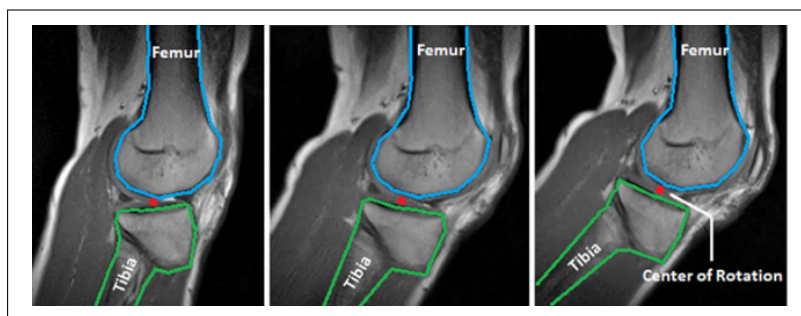


Figure 1.1: MRI scan of the knee joint including Femur and Tibia bones, and their movement [10].

In fact, the overall kinematic movement of the knee joint is too complex. It includes 5 degrees of freedom [DOFs] [12]. The ultimate goal of the project, however is to create a unique and novel design that will create freedom for knee movement for x and y axis, without causing misalignment.

Research Applications

This research project involves different applications mainly focused in medical and recovery fields.

Post Stroke Rehabilitation:

Provides training for stroke survivors, practicing flexing and extension movement of the leg. Promotes motor recovery and neuroplasticity. Furthermore, with the more flexible design, the patients will also have chance to improve their gait pattern, by guiding natural joint motion.

Elders' health support:

At the elderly age it is common trait to have a weakened lower limb muscle system which sometimes will fail to effectively and easily create movement of lower parts of the body. The exoskeleton can provide partial assistance during sit-to-stand motion [3], walking, flexing and extension process of the leg.

Therapists' workload reduction:

More therapists will be able just prescribe the exoskeleton rehabilitation process, without taking a lot of time manually helping patients, thus having more time to serve more patients.

Room for Future Research:

Potentially, our project can serve as a room for future research in the field, being cited by other researchers and providing them our solution design.

Chapter 2: Literature Review & Existing Solutions Analysis

It is reasonable to state that there is a plenty of existing research on topic of lower limb exoskeletons. The areas vary from analysis of their impact on patients' rehabilitation and gait recovery to proposal of new design solutions, using different approaches, actuators, budget etc.

Effects of exoskeleton assisted therapy

Fan et al.[2] explored effects of exoskeleton on patients for example. The research strategy was comparison of 2 groups: one with BEAR-H1 exoskeleton assisted training, and other with conventional training. Both groups improved gait but the results were better for the first group. However, Chang et al. [13], who made an analysis of different exoskeleton and their results on patients, stated that the role of robot-assisted in stroke rehabilitation is just an adjunct instead of complete replacement for conventional rehabilitation processes.

Knee Joint Biomechanical Analysis

As stated before, due to Tibia's movement along the Femur bone, using 1 degree of freedom joint will cause misalignment, because the center does not change its place, as it does naturally in the knee. Yu et al. [10], measures misalignment across different angles of knee flexion.

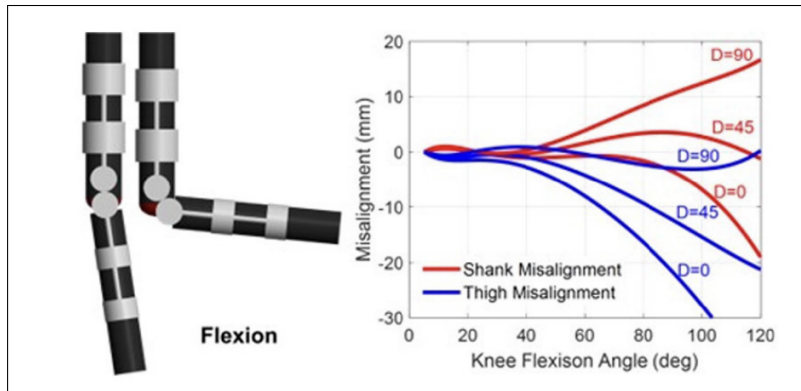


Figure 2.1: Misalignment versus knee flexion angle graphs [10].

Misalignment and its Consequences

In most of the existing knee exoskeletons the joint is implemented as a simple hinge aligned with a single, fixed, assumed anatomical knee axis. However, the human knee does not behave as an ideal revolute joint, as its instantaneous centre of rotation changes location with a curved path relative to the tibia during the flexion and extension. In result, any single hinge exoskeleton will inevitably experience kinematic misalignments between the device and the underlying anatomical joint.

Yu et al. (2022) report that even moderate offsets between the anatomical and mechanical joint centers can lead to significant relative motion gap between the shell and the limb, increased shear forces at the cuffs and straps, and non uniform pressure on the soft tissues. [10] During dynamic tasks such as walking or sit-to-stand transitions this misalignment can manifest as:

- sliding of the cuffs on the skin and soft tissue;
- local pressure peaks at the edges of the shells and straps;
- unwanted parasitic forces and torques transmitted to the limb.

The effects described above do not only reduce the comfort during the treatment, but also can result in the limited range of motion the user is willing to perform while wearing the device. During the rehabilitation, the increased misalignment can result in ineffective therapy or increased risk of skin irritation, bruising, or joint overloading. Therefore, reducing the mismatch between the trajectories of the anatomical and mechanical ICRs' is an essential goal for the knee exoskeleton intended for long term, intensive use.

Existing Design Solutions

Kim et al. [6] created a novel design of knee exoskeleton which is efficient in terms of energy maintenance while load carrying augmentation. Jain et al. [3] used a linear motor to create a knee exoskeleton for sit-to-stand and walking motions. Wu et al. [4] used magnetic rings in their actuator to make a 1 degree of freedom motor providing assistance with flexion and extension movement of the knee. Witte et al. [5] proposed a lightweight exoskeleton solution, which utilizes Bowden cables.

All of the solutions above, however, do not consider the knee joint movement. There are some works, which do. They come up with unique methods using various actuators and approaches, eventually decreasing misalignment and providing freedom in movement.

Design Concept Selection

There are a lot of existing solutions, some of which were described in previous subsection. However the goal of this project is to focus on minimizing misalignment caused by knee flexion movement. In this section solutions, designs closest to the solution will be provided and analyzed.

Even though it is impossible to avoid misalignment totally, Kim et al. [9]. [9] have done significant effort to minimize the misalignment, using bearing and Femur imitating guide rails.

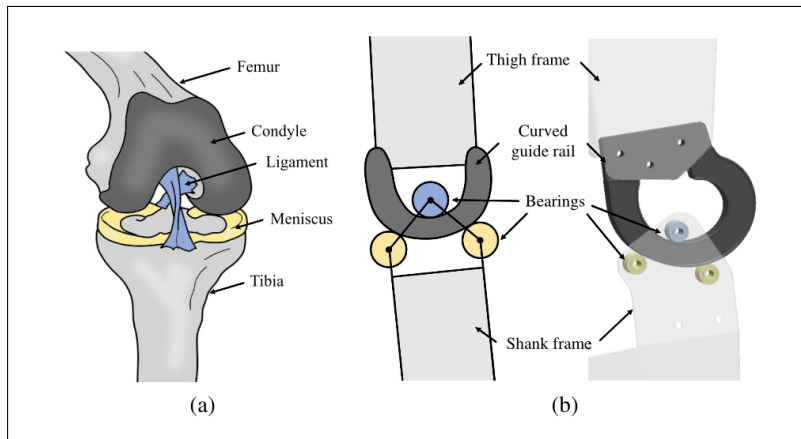


Figure 2.2: Kim et al. solution to minimize the misalignment [9].

Jin et al. [7], have proposed a push-pull actuator that has ropes which imitates muscles. Using servo motor they control the number of 'muscles' involved in the flexion-extension movement thus controlling the tightness of the actuator.

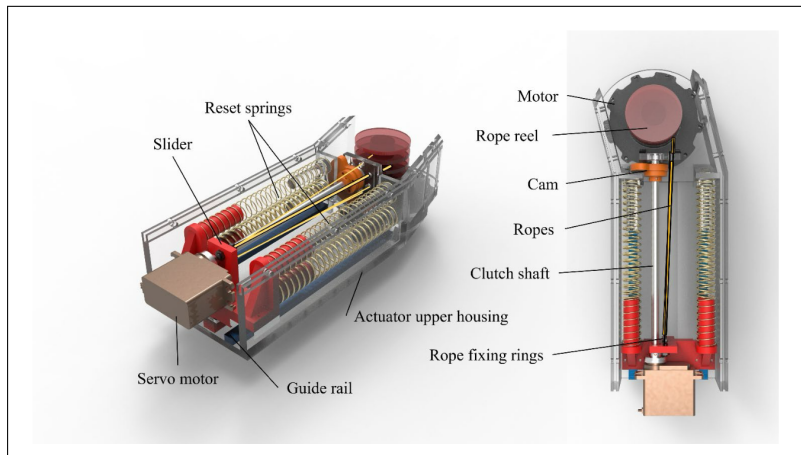


Figure 2.3: Actuator proposed by Jin et al. [7].

Yu et al. [10] also proposed an actuator that tries to replicate the movement of the Tibia during the flexion and extension of the knee.

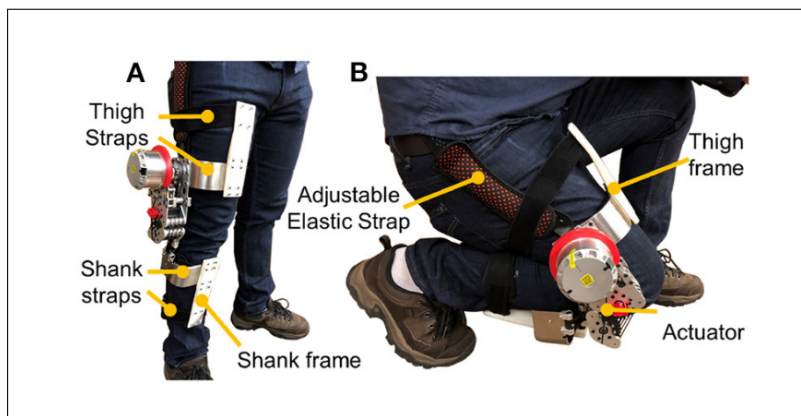


Figure 2.4: Actuator proposed by Yu et al. [10].

The most suitable, in our opinion, work was presented by Gao et al. [14]. They used the 4 bar mechanism, with 2 intersecting links to mimic the anatomic ICR movement. The intersection of the links represented the mechanical ICR. We find the work most suitable for us, as they demonstrated the good performance in matching the ICR movement after the optimization of their exoskeleton. Based on their model we will try to come up with our own mathematical model that will also try to minimize misalignment by matching ICRs' movements.

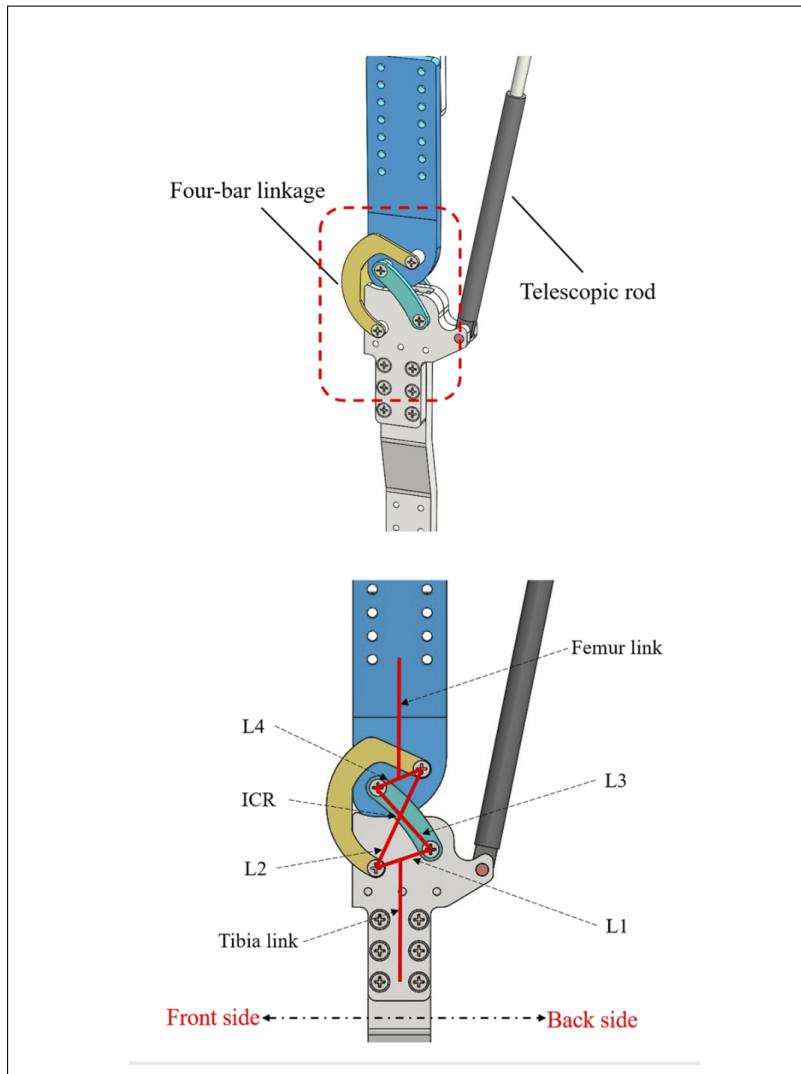


Figure 2.5: four cross bar linkage proposed by Gao et al. [14].

Initial Concept: Gear-Based Mechanism

The initial idea for the concept of the joint mechanism was to use a pair of gears to mimic the rolling and sliding interaction between the Femur and Tibia. The idea was to have two meshing gear segments, one of which had fixed position, mounted on the thigh and shank shells that generate a non-circular rotatory motion. Initially this approach appeared attractive to use because:

- the transmission ratio and motion profile can, in principle, be encoded directly into the tooth geometry;
- gears can be compact in the sagittal plane and can be combined with rotary actuators;
- the mechanism has a single degree of freedom, which simplifies control.

However, as the concept was developed further, several serious drawbacks emerged:

- Sensitivity to misalignment and manufacturing tolerances: even small errors in positioning the gear centres or in tooth geometry can lead to binding, backlash or uneven contact.
- Difficult contact with soft tissue: large, rigid gear arcs positioned close to the knee joint leave little room for padding and adjustment, making it hard to accommodate different leg shapes.
- Limited ability to tune the ICR path: designing the tooth profiles to reproduce a specific anatomical ICR trajectory is complex, and small changes require re-machining the gear segments.
- Weight and bulk: robust gear segments with sufficient tooth height add mass around the knee and increase the inertia of the moving parts.

In practice, it became clear that, although gear mechanisms can generate sophisticated relative motions, they are poorly suited for a wearable device where comfort, adjustability and soft-tissue interaction are critical. This motivated the search for an alternative kinematic structure with simpler geometry, easier adjustability and a more straightforward relation between link dimensions and ICR motion.

Final Concept: Crossed Four-Bar Linkage

After exploring gear-based mechanisms, we turned to linkage structures as an alternative way to generate a moving ICR. Four-bar linkages are widely used in mechanical design because they are simple, planar and can approximate complex motion with a small number of rigid links. In particular, a *crossed* four-bar linkage, where the coupler links intersect, naturally produces an instantaneous centre of rotation at the intersection point of the links.

In the proposed design, two of the links are attached to the thigh and shank shells, while the other two form a crossed pair between them. The intersection of the crossed links defines the mechanical ICR of the mechanism. By adjusting the link lengths and the positions of the attachment points on the shells, the path of this intersection point can be shaped to resemble the anatomical ICR trajectory.

Compared to the initial gear concept, the crossed four-bar linkage offers several advantages:

- **Geometric transparency:** the effect of changing a link length or joint location on the ICR path is easier to understand and tune.
- **Planar, low-profile layout:** the linkage can be arranged close to the limb in the sagittal plane, leaving space for padding and straps.
- **Single degree of freedom:** the mechanism is compatible with simple actuation schemes or can be used passively with external supports.
- **Manufacturability:** links and joints can be produced using standard machining or additive manufacturing without complex tooth profiles.

Based on the literature on knee joint misalignment and our own design constraints (limited space around the knee, need for adjustability, and a preference for simple planar mechanisms), we therefore adopted a crossed four-bar linkage as the final concept. The remainder of

this work focuses on modelling the anatomical ICR path, deriving the kinematics of the chosen linkage geometry, and evaluating how well the resulting mechanical ICR trajectory approximates the anatomical one in the chosen coordinate frame.

Chapter 3: Biomechanical Modeling of the knee (Locating the ICR)

3.1 Modeling Assumptions

In this work, we do not build a new knee model from scratch. Instead, we adopt an existing three-dimensional tibiofemoral kinematic model from the literature and adapt it for our exoskeleton design. Since the proposed knee joint mechanism is planar and mounted on the lateral side of the leg, we restrict our analysis to motion in the sagittal plane and make several simplifying assumptions.

First, we assume that the femur and tibia behave as rigid bodies connected by an idealized joint whose motion is fully described by a single flexion angle F . Secondary motions such as varus/valgus rotation and internal/external rotation are taken from the reference 3D model, but they are not explicitly reproduced by the exoskeleton mechanism. Instead, they are only used within the kinematic model to determine the position of the instantaneous center of rotation (ICR).

Second, we approximate the 3D motion by its sagittal-plane projection. In the original model, the tibiofemoral motion is described in three dimensions using rotations and translations in all anatomical directions. For the purposes of this study, we are interested only in how the knee joint center moves in the sagittal plane, because our cross four-bar mechanism lies in this plane. Lateral translations, out-of-plane rotations, patellofemoral motion, and soft-tissue deformations are therefore neglected.

Third, we consider a purely kinematic analysis. Ground reaction forces, muscle forces, ligament forces and inertial effects are not included. The model provides the relative motion between femur and tibia for a given flexion angle F ; we do not attempt to compute joint reaction forces or required actuator torques at this stage.

Finally, the adopted kinematic model is based on measurements from a generic subject. We assume that this motion pattern is representative for the target patient population and do not perform subject-specific scaling or personalization. As a result, the anatomical ICR trajectory obtained from the model should be interpreted as an approximate “average” reference path that our exoskeleton mechanism aims to follow, rather than an exact description of any individual patient’s knee.

3.2 Knee Kinematic Equations from Literature

In the adopted tibiofemoral kinematic description, the sagittal-plane motion relevant to a planar exoskeleton can be represented by the anterior–posterior and proximal–distal displacements of the knee joint centre as functions of the flexion angle F . Consistent with the modelling approach in Gao et al. [14], we use best-fit polynomial relationships to parameterize

these displacements (with F in degrees and distances in millimetres)

$$Y_{\text{dist}}(F) = -0.0683F + 8.804 \times 10^{-4}F^2 - 3.375 \times 10^{-6}F^3, \quad (3.1)$$

$$Z_{\text{dist}}(F) = -0.1283F + 4.796 \times 10^{-4}F^2, \quad (3.2)$$

where distances are expressed in millimetres. Because the proposed crossed four-bar mechanism is planar and lies in the sagittal plane, the anatomical reference trajectory used for mechanism tuning is obtained via the planar approximation

$$Z_{\text{anat}}(F) = Z_{\text{dist}}(F), \quad Y_{\text{anat}}(F) = Y_{\text{dist}}(F), \quad (3.3)$$

so that the sagittal-plane anatomical reference curve is defined by the locus $(Z_{\text{anat}}(F), Y_{\text{anat}}(F))$ over the flexion range of interest.

For consistent comparison between anatomical and mechanical trajectories, we shift the reference so that full extension ($F = 0^\circ$) coincides with the origin of the tibia-fixed sagittal frame:

$$\tilde{Z}_{\text{anat}}(F) = Z_{\text{anat}}(F) - Z_{\text{anat}}(0), \quad \tilde{Y}_{\text{anat}}(F) = Y_{\text{anat}}(F) - Y_{\text{anat}}(0). \quad (3.4)$$

The shifted trajectory $(\tilde{Z}_{\text{anat}}(F), \tilde{Y}_{\text{anat}}(F))$ is used as the fixed anatomical target in the evaluation and optimisation procedures presented later in the report.

3.3 MATLAB Implementation

The anatomical reference model is implemented in MATLAB by sampling the flexion angle on a uniform grid $F \in [0^\circ, 120^\circ]$ with 1° resolution. For each sample, Equations (3.1)–(3.2) are evaluated to obtain $Y_{\text{dist}}(F)$ and $Z_{\text{dist}}(F)$, followed by the planar mapping in Equation (3.3). The shift in Equation (3.4) is then applied to produce the shifted reference components $\tilde{Y}_{\text{anat}}(F)$ and $\tilde{Z}_{\text{anat}}(F)$.

The resulting reference is visualised in three complementary ways: (i) proximal–distal translation $\tilde{Y}_{\text{anat}}(F)$ versus flexion angle, (ii) anterior–posterior translation $\tilde{Z}_{\text{anat}}(F)$ versus flexion angle, and (iii) the sagittal-plane locus $(\tilde{Z}_{\text{anat}}(F), \tilde{Y}_{\text{anat}}(F))$. These plots provide both an interpretable decomposition of the motion and a direct geometric target for the mechanical ICR trajectory of the exoskeleton mechanism.

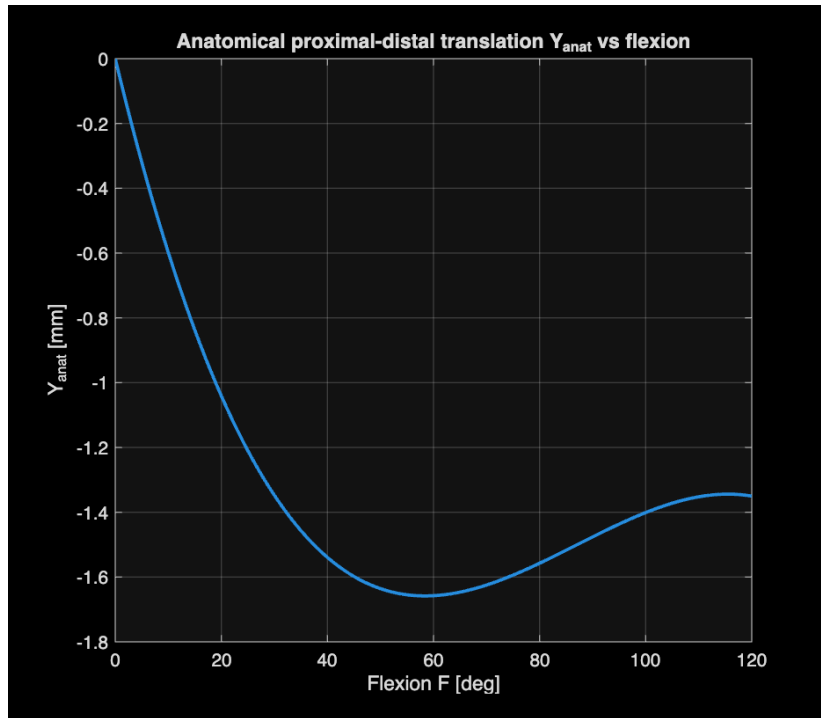


Figure 3.1: Proximal–distal translation $\tilde{Y}_{\text{anat}}(F)$ of the sagittal-plane anatomical reference as a function of knee flexion angle F .

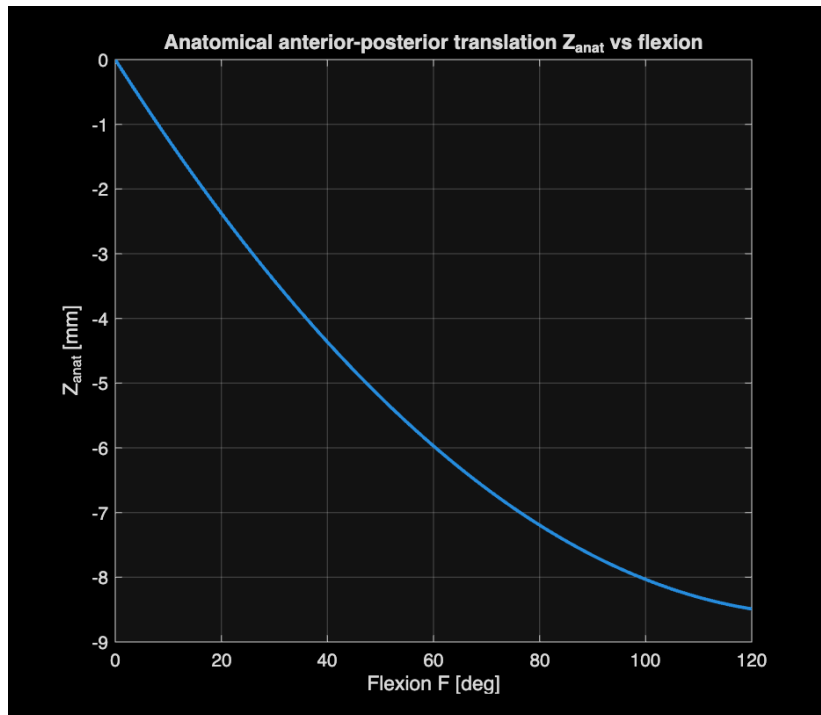


Figure 3.2: Anterior–posterior translation $\tilde{Z}_{\text{anat}}(F)$ of the sagittal-plane anatomical reference as a function of knee flexion angle F .

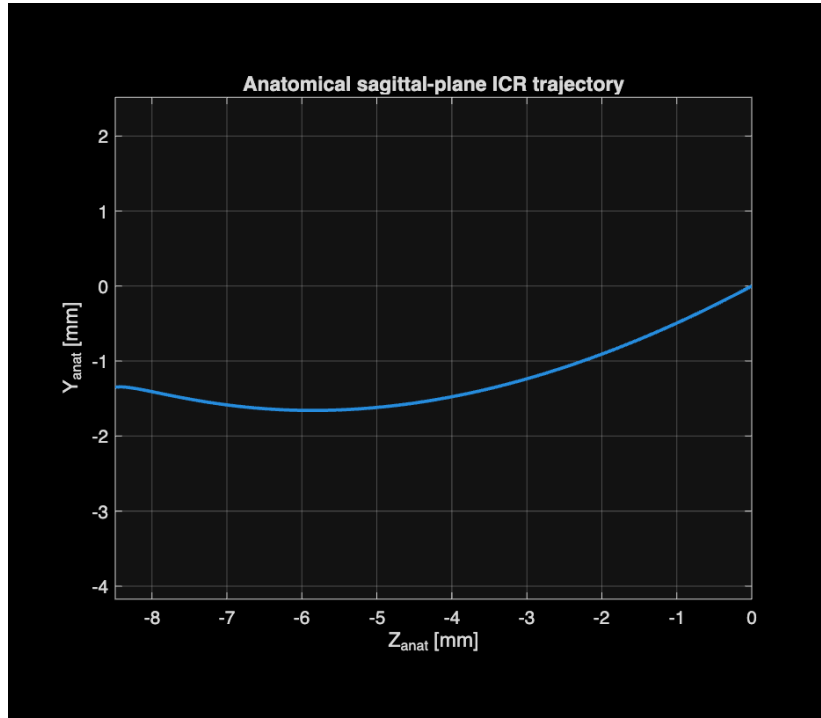


Figure 3.3: Sagittal-plane anatomical reference trajectory $(\tilde{Z}_{\text{anat}}(F), \tilde{Y}_{\text{anat}}(F))$ used as the target ICR path for mechanism tuning and evaluation.

Chapter 4: Crossed four–bar knee mechanism kinematics

4.1 Geometry and notation

We model the knee exoskeleton as a planar crossed four–bar linkage in the sagittal plane, following the general configuration used in Gao et al. [14]. A crossed four–bar mechanism is well suited for knee assistance because it can generate a *moving* instantaneous centre of rotation (ICR). Compared to an ideal hinge joint (fixed rotation centre), a moving ICR better reflects tibiofemoral rolling–sliding behaviour and may reduce relative motion between the exoskeleton and the wearer.

The linkage consists of four rigid links connected by revolute joints forming a quadrilateral $ABCD$ in the tibia–fixed sagittal plane. The fixed base link is $L_1 = \overline{AD}$, attached to the tibial segment and connecting the distal attachment point A to a proximal joint D . The two crossed links are

$$L_2 = \overline{AC}, \quad L_3 = \overline{DB},$$

and the coupler is

$$L_4 = \overline{CB}.$$

The mechanical ICR of the mechanism is defined as the intersection of the extended lines along the crossed links AC and DB . When the linkage remains in the crossed configuration, this intersection lies in a physically meaningful region near the knee joint.

We adopt a tibia–fixed (Z, Y) coordinate frame with origin at the distal attachment point A . The Z –axis is aligned with the tibial shaft in the anterior–posterior direction and the Y –axis points proximally along the shank. The absolute orientation of the base link L_1 is described by an angle β measured from the positive Z –axis. The coordinates of A and D are therefore

$$A = (Z_A, Y_A) = (0, 0), \quad D = (Z_D, Y_D) = (L_1 \cos \beta, L_1 \sin \beta). \quad (4.1)$$

The crossed link L_2 has absolute orientation $\beta + \alpha$, where α is the input angle of the mechanism. The coordinates of joint C become

$$C = (Z_C, Y_C) = (L_2 \cos(\beta + \alpha), L_2 \sin(\beta + \alpha)). \quad (4.2)$$

The remaining joint B is determined from link-length constraints (Section 4.2). This mirrors the physical mechanism: once the input is prescribed and link lengths are fixed, the coupler closes the loop and forces the position of B .

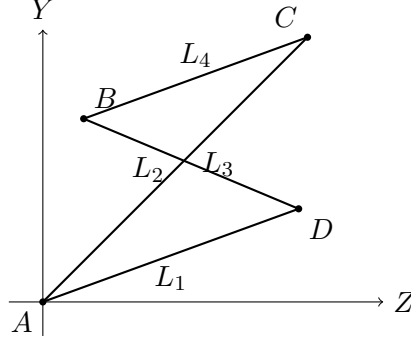


Figure 4.1: Geometry and notation of the crossed four-bar knee mechanism in the tibia-fixed (Z, Y) frame. The mechanical ICR is defined as the intersection of the crossed links AC and DB .

4.2 Position analysis of the crossed four-bar mechanism

This section describes the forward kinematics used to compute the mechanism configuration and the resulting mechanical ICR trajectory. The input to the analysis is the knee flexion angle F sampled on a grid, together with an input mapping $\alpha(F)$ that represents how the mechanism input changes with knee flexion. The output is the set of joint coordinates (A, B, C, D) and the mechanical ICR point $I_{\text{mech}}(F)$ for each flexion sample.

Two aspects are critical for a meaningful result. First, the linkage must remain feasible over the full flexion range: the loop can only close if the circle-circle intersection used to determine B exists for every F . Second, the solution must remain on the crossed branch; switching to the open branch produces a different kinematic behaviour and an inconsistent ICR. For these reasons, feasibility and continuity checks are integrated directly into the computation.

4.2.1 Prescribed input and known joint locations

For each flexion sample F , the mechanism input angle $\alpha(F)$ is prescribed. Geometrically, α controls the direction of link L_2 relative to the base orientation β , and therefore sets the location of joint C on a circle of radius L_2 about point A . Once $\alpha(F)$ is fixed, joints A , D and C are known immediately from the link geometry. Specifically, using the tibia-fixed frame introduced in Section 4.1, we write

$$A(F) = (0, 0), \quad (4.3)$$

$$D(F) = (L_1 \cos \beta, L_1 \sin \beta), \quad (4.4)$$

$$C(F) = (L_2 \cos(\beta + \alpha(F)), L_2 \sin(\beta + \alpha(F))). \quad (4.5)$$

These define the two circle centres used to compute $B(F)$: $D(F)$ is the centre for the circle of radius L_3 and $C(F)$ is the centre for the circle of radius L_4 .

4.2.2 Computing joint B via circle–circle intersection

Joint B is constrained by fixed link lengths $L_3 = \overline{DB}$ and $L_4 = \overline{CB}$, therefore it must satisfy

$$\|B(F) - D(F)\| = L_3, \quad \|B(F) - C(F)\| = L_4. \quad (4.6)$$

Geometrically, Equation (4.6) defines two circles in the (Z, Y) plane. If they intersect, the intersection points provide valid candidates for $B(F)$. If they do not intersect, the mechanism cannot close for that particular F and parameter set. This explicit feasibility check prevents numerical failures and provides a physically meaningful constraint during optimisation.

In general, two intersection points exist, denoted $B_1(F)$ and $B_2(F)$, corresponding to the open and crossed configurations. Both satisfy the length constraints, so an additional rule is required to select the physically relevant crossed branch.

4.2.3 Crossed-branch selection and continuity

For each candidate point $B_i(F)$ we compute the orientation of link L_3 as

$$\delta_i(F) = \text{atan2}(Y_{B_i}(F) - Y_D(F), Z_{B_i}(F) - Z_D(F)). \quad (4.7)$$

At $F = 0^\circ$, the crossed assembly corresponds to one candidate (initialized using a simple geometric rule). For subsequent samples, we choose the candidate whose $\delta_i(F)$ is closest to the previously selected value. This prevents branch switching and ensures a smooth configuration evolution. The resulting $\delta(F)$ is shown in Figure 4.2.

4.2.4 Mechanical ICR computation

Once $A(F)$, $C(F)$, $D(F)$ and the selected $B(F)$ are known, the mechanical ICR is computed as the intersection of links AC and DB . We write the two lines in parametric form

$$\mathbf{r}_{AC}(s) = A(F) + s(C(F) - A(F)), \quad \mathbf{r}_{DB}(u) = D(F) + u(B(F) - D(F)), \quad (4.8)$$

and solve $\mathbf{r}_{AC}(s) = \mathbf{r}_{DB}(u)$ for the parameters (s, u) . The intersection point defines

$$I_{\text{mech}}(F) = \begin{bmatrix} Z_{\text{mech}}(F) \\ Y_{\text{mech}}(F) \end{bmatrix} = A(F) + s^*(F)(C(F) - A(F)). \quad (4.9)$$

Repeating this computation across all sampled flexion angles produces the mechanical ICR trajectory. For interpretation, we visualise the motion both as component translations $Z_{\text{mech}}(F)$ and $Y_{\text{mech}}(F)$ and as the sagittal-plane path $(Z_{\text{mech}}(F), Y_{\text{mech}}(F))$.

4.3 Optimisation-based parameter tuning

The forward-kinematics procedure above defines a mapping from mechanism parameters to a predicted mechanical ICR trajectory. To improve agreement with the anatomical reference

trajectory derived in Chapter 3, the mechanism parameters are tuned using particle swarm optimisation (PSO), following Gao et al. [14]. PSO is appropriate here because the objective is nonlinear, the design space is multi-parameter, and feasibility constraints can create discontinuities that are difficult for gradient-based methods. The design vector is

$$\mathbf{x} = [L_1, L_2, L_3, L_4, \beta, k, \alpha_0]^T, \quad (4.10)$$

and the input mapping is approximated by

$$\alpha(F) = kF + \alpha_0, \quad (4.11)$$

with F in radians. The parameters (k, α_0) capture a scaling and offset relationship between knee flexion and the mechanism input angle. For each candidate $\mathbf{x}$, the forward kinematics is evaluated over the entire flexion range to produce the mechanical trajectory. Candidates that lose circle intersection for B or switch branches are penalized. The objective function and quantitative tracking metrics are defined in Chapter 6, and the resulting tuned performance is reported in Chapter 7.

4.4 Outputs and visualisation

This section summarises the outputs of the position analysis and clarifies how they are used later for evaluation. For each flexion sample F , the procedure yields the mechanical ICR coordinates

$$I_{\text{mech}}(F) = (Z_{\text{mech}}(F), Y_{\text{mech}}(F)).$$

We report the trajectory in three complementary visualisations. First, $Z_{\text{mech}}(F)$ describes the anterior–posterior translation of the ICR as a function of flexion. Second, $Y_{\text{mech}}(F)$ describes the proximal–distal translation. Finally, plotting Y_{mech} against Z_{mech} provides the sagittal-plane ICR path, which is the most direct representation for comparing with the anatomical reference trajectory.

These plots are not only descriptive; they are also the basis for quantitative misalignment measures (Chapter 6). Comparing the mechanical and anatomical coordinates sample-by-sample enables the computation of an error curve versus flexion, as well as aggregate metrics such as RMS and maximum deviation.

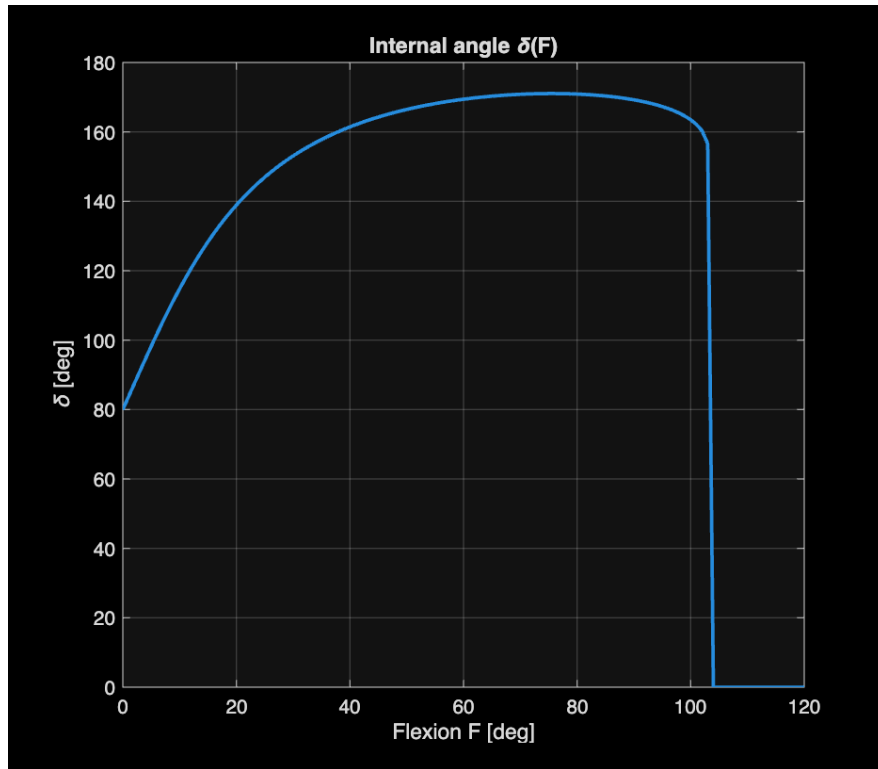


Figure 4.2: Internal angle $\delta(F)$ of link L_3 as a function of the knee flexion angle F .

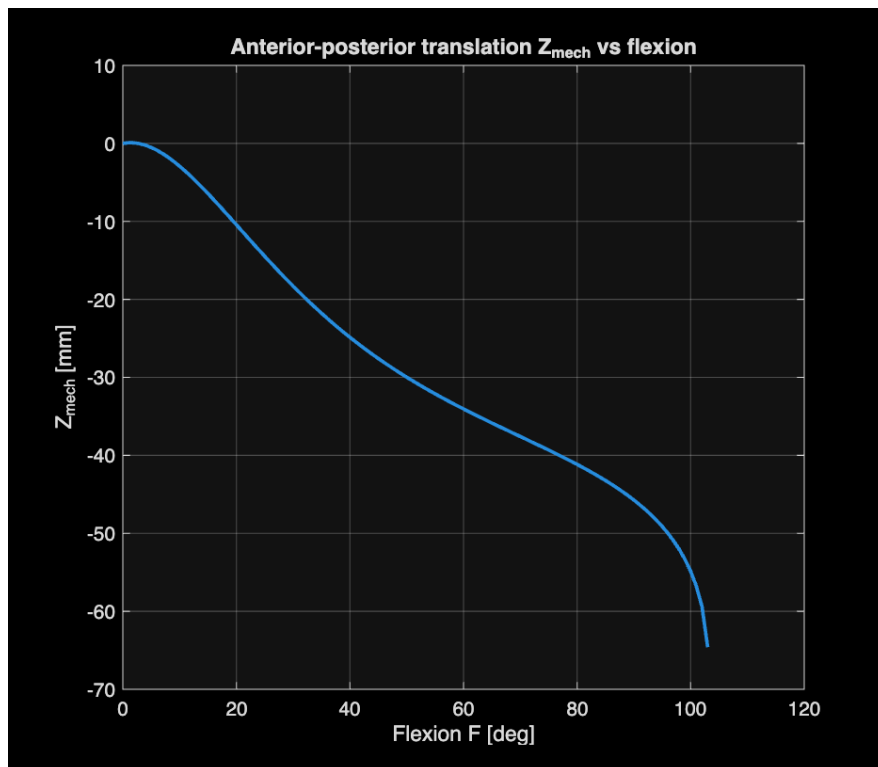


Figure 4.3: Anterior-posterior translation $Z_{\text{mech}}(F)$ of the mechanical ICR as a function of flexion angle F .

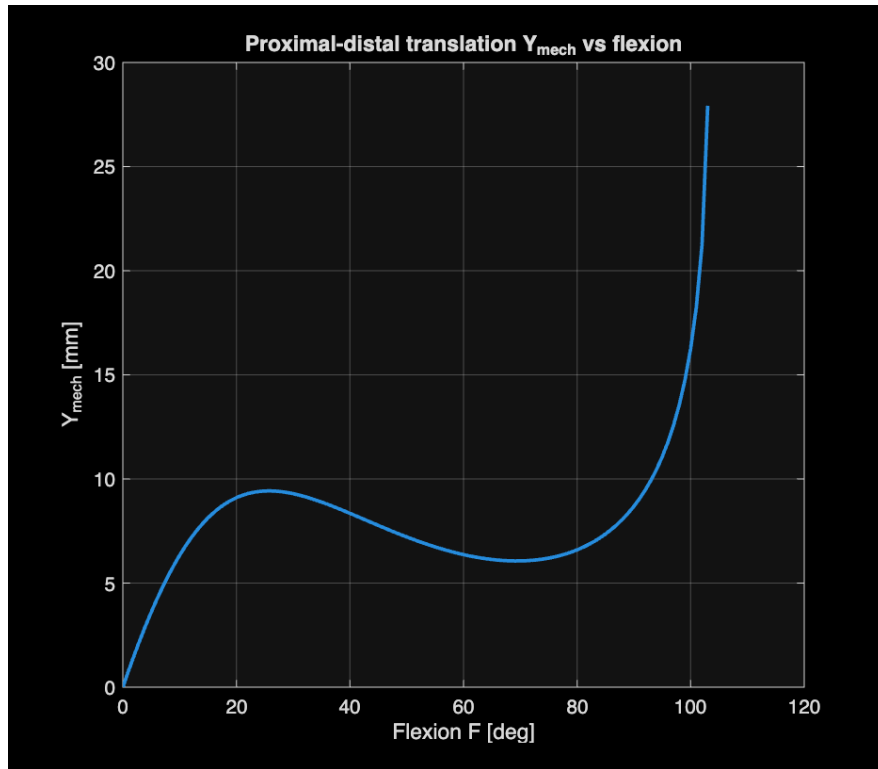


Figure 4.4: Proximal-distal translation $Y_{\text{mech}}(F)$ of the mechanical ICR as a function of flexion angle F .

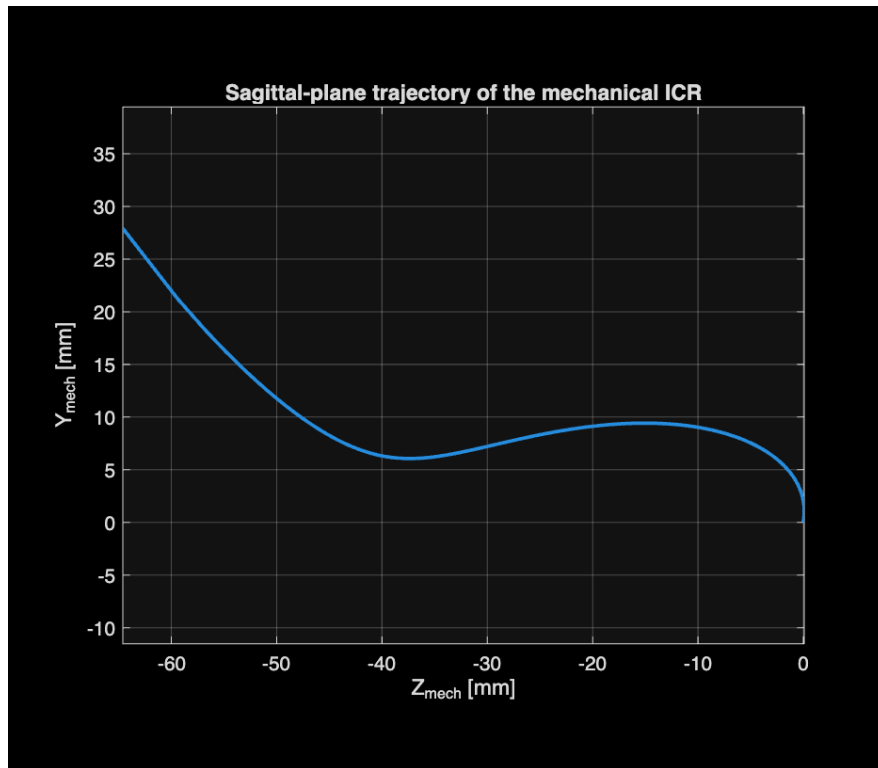


Figure 4.5: Sagittal-plane trajectory of the mechanical instantaneous centre of rotation $(Z_{\text{mech}}(F), Y_{\text{mech}}(F))$.

Chapter 5: Exoskeleton Knee CAD Design

5.1 Exoskeleton knee CAD design

This chapter describes the preliminary three-dimensional CAD design of the knee exoskeleton joint that implements the crossed four-bar mechanism. The goal is to show how the planar linkage can be embedded into a wearable structure that interfaces with the user's thigh and shank, and to verify that the required range of motion can be achieved without geometric interference.

5.1.1 Exoskeleton overview

The exoskeleton knee joint is realised as a pair of rigid shells attached to the distal thigh and proximal shank, respectively. These shells provide interface surfaces for straps or padding and act as mounting platforms for the crossed four-bar mechanism. The links L_1 – L_4 derived in Chapter 4 are implemented as rigid bars or plates connecting bolted joints on the shells.

Figure 5.1 shows the overall CAD assembly of the knee exoskeleton joint. The thigh shell carries the proximal attachment points of links L_2 and L_4 , while the shank shell carries the distal attachment points of links L_1 and L_3 . A rotary actuator can be mounted coaxially with one of the revolute joints (for example at the proximal joint of L_2), so that the motor torque is transmitted directly into the four-bar mechanism without additional transmission stages. The geometric parameters of the mechanism are identical to those used in the kinematic analysis of Chapter 4.

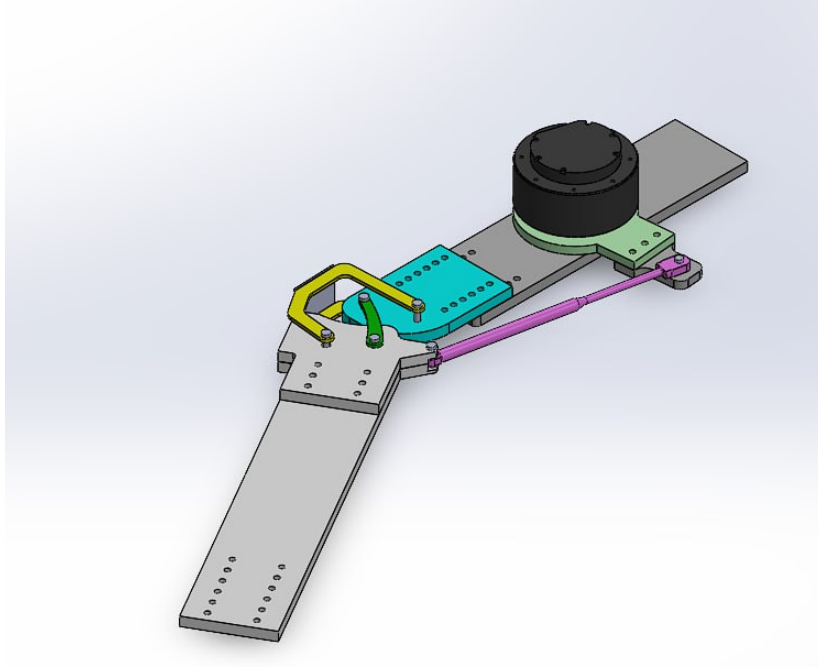


Figure 5.1: Overall CAD assembly of the knee exoskeleton joint with the crossed four-bar mechanism mounted between the thigh and shank shells.

5.1.2 Mapping of link lengths to shell bolt positions

To ensure that the mechanical ICR trajectory of the CAD model matches the kinematic model from Chapter 4, the attachment points of the links on the shells must be placed so that the distances between them are equal to the link lengths L_1 – L_4 used in the analysis.

In the planar model, the four joints are denoted by A , B , C and D (Section 4.1). In the CAD assembly, these points are implemented as pairs of coaxial bolt holes on the inner surfaces of the thigh and shank shells:

- point A is located on the shank shell and represents the distal reference point of the tibia-fixed frame (Z, Y);
- point D is also on the shank shell, at a distance L_1 from A , and forms the distal attachment of link L_3 ;
- point C is on the thigh shell, at a distance L_2 from A , forming the proximal attachment of link L_2 ;
- point B is on the thigh shell, at a distance L_4 from C and L_3 from D , forming the proximal attachment of link L_3 .

By construction, the distances $|AD|$, $|AC|$, $|DB|$ and $|CB|$ in the CAD model are therefore equal to the link lengths L_1 , L_2 , L_3 and L_4 used in the planar kinematic model.

Figure 5.2 illustrates the mapping between the planar four-bar geometry and the three-dimensional shell model. The bolt positions corresponding to joints A – D are shown on the inner surfaces of the shells together with the key dimensions. In practice, these distances are chosen according to anthropometric constraints (for example, the approximate distance

between the knee joint line and the chosen reference points on the thigh and shank) and then adjusted to match the final optimised values of L_1 – L_4 obtained in Chapter 4.

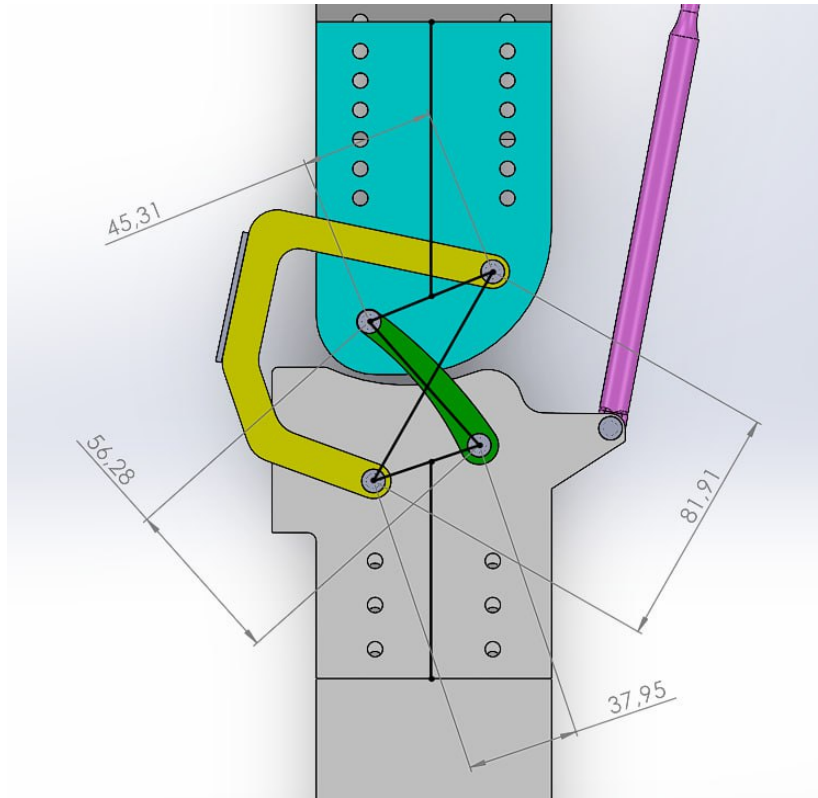


Figure 5.2: Mapping between the planar four-bar link lengths L_1 – L_4 and the bolt positions A – D on the thigh and shank shells in the CAD model.

5.1.3 Assembly and range of motion in CAD

After positioning the bolt holes for joints A – D on the shells, the links L_1 – L_4 are assembled as rigid bodies with revolute joints at their ends. In the CAD environment the range of motion of the knee joint can then be simulated by rotating the thigh shell relative to the shank shell while constraining the four-bar mechanism.

Figures 5.3–5.6 show snapshots of the CAD assembly at different knee flexion angles. In full extension ($F \approx 0^\circ$), the shells are aligned and the four-bar mechanism is close to its reference configuration. As the flexion angle increases, the crossed links L_2 and L_3 rotate relative to each other and the coupler link L_4 moves accordingly, reproducing the kinematic behaviour analysed in Chapter 4. Visual inspection of these configurations provides a qualitative check that the mechanism remains within the intended working quadrant and that the shells can accommodate the motion without geometric interference.

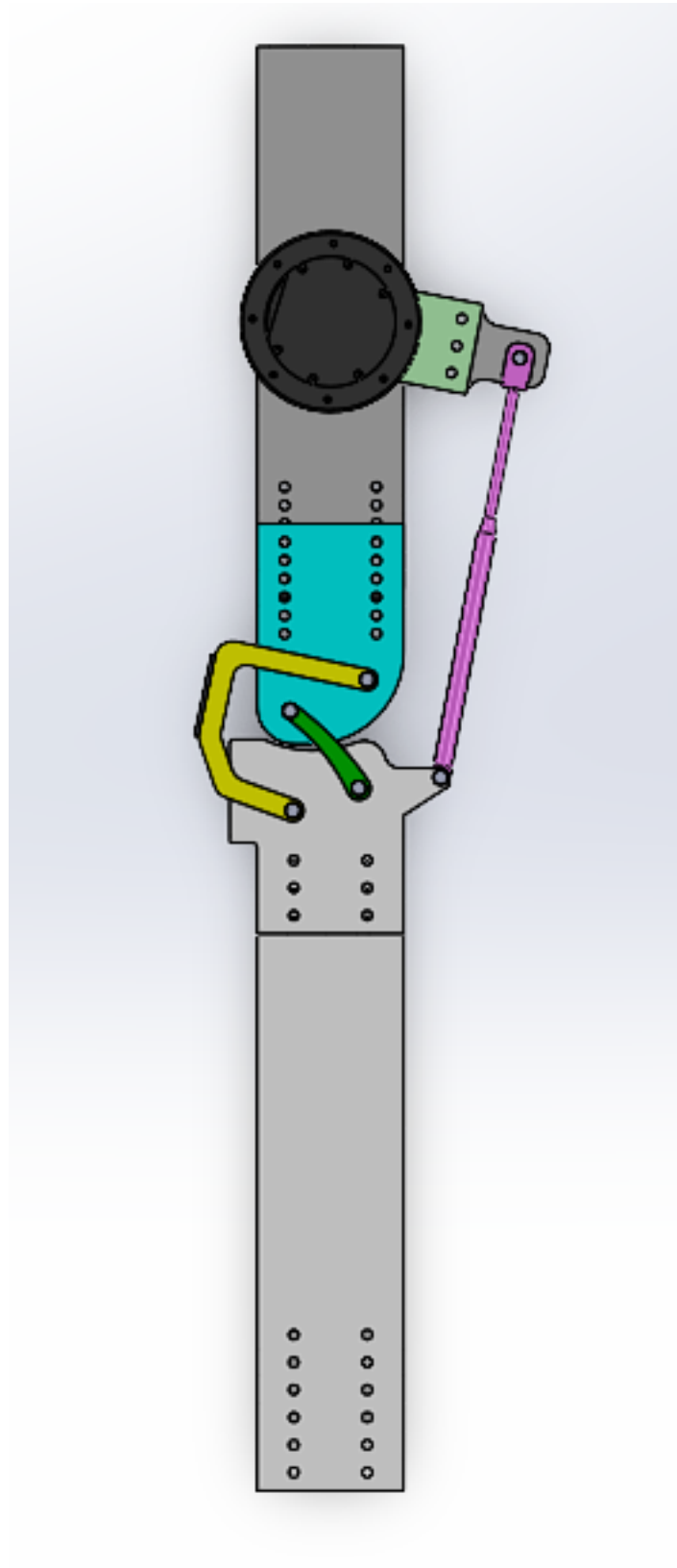


Figure 5.3: CAD assembly of the knee exoskeleton joint at approximately $F = 0^\circ$ (full extension).

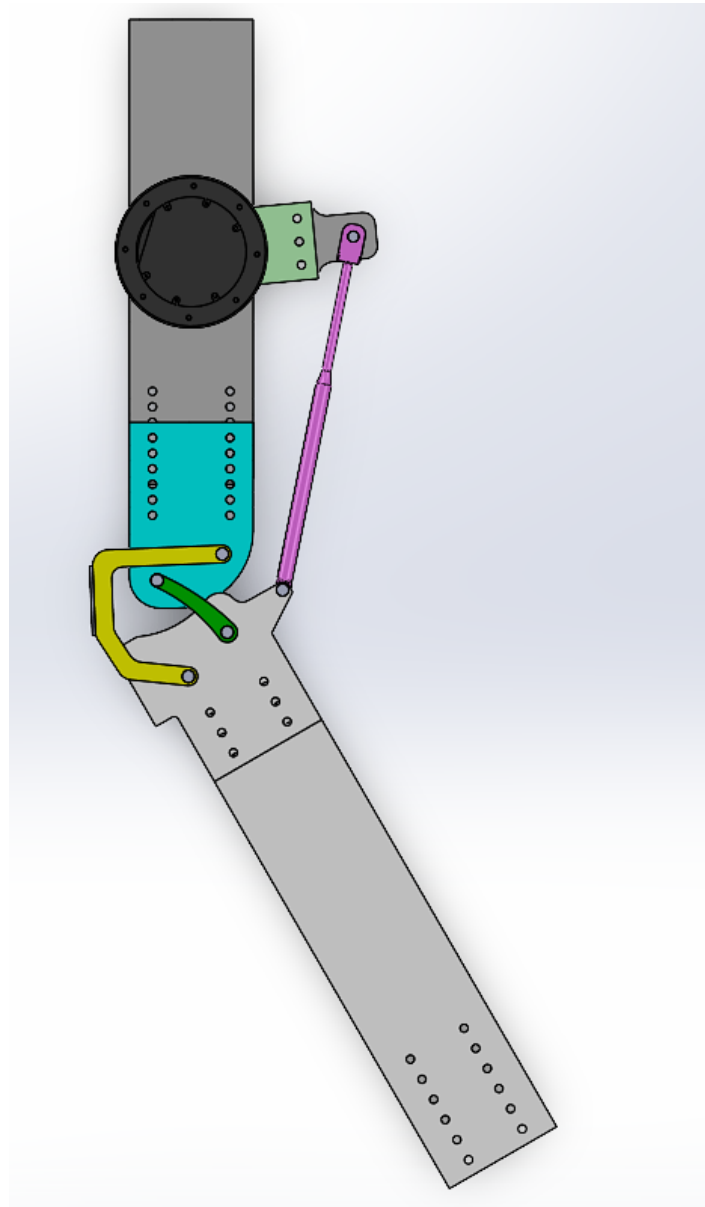


Figure 5.4: CAD assembly of the knee exoskeleton joint at approximately $F = 30^\circ$ knee flexion.

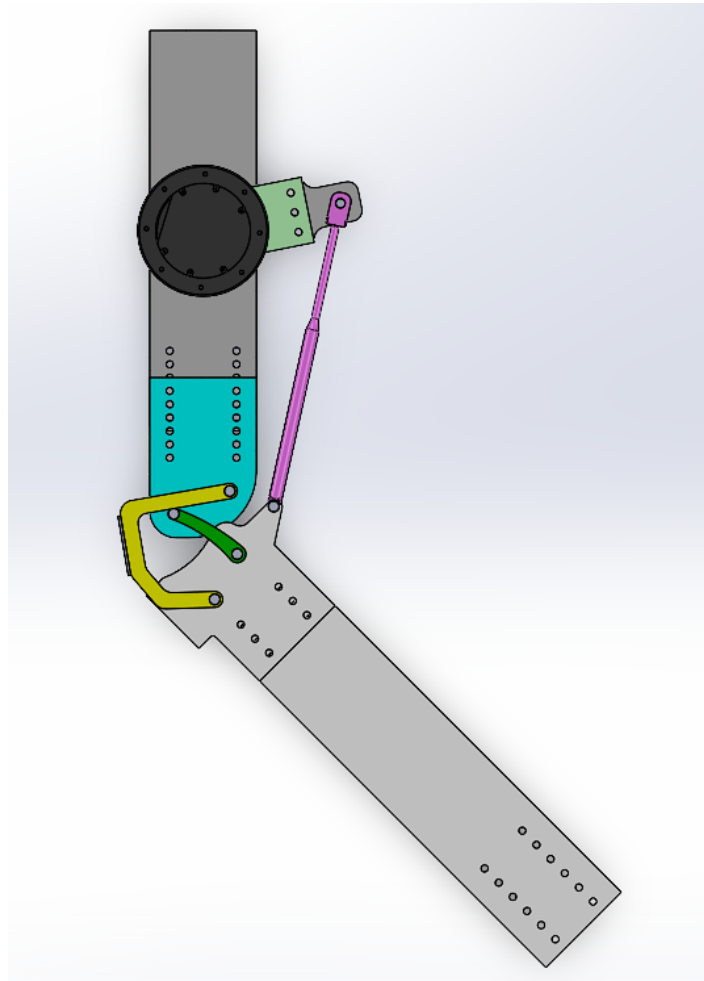


Figure 5.5: CAD assembly of the knee exoskeleton joint at approximately $F = 45^\circ$ knee flexion.

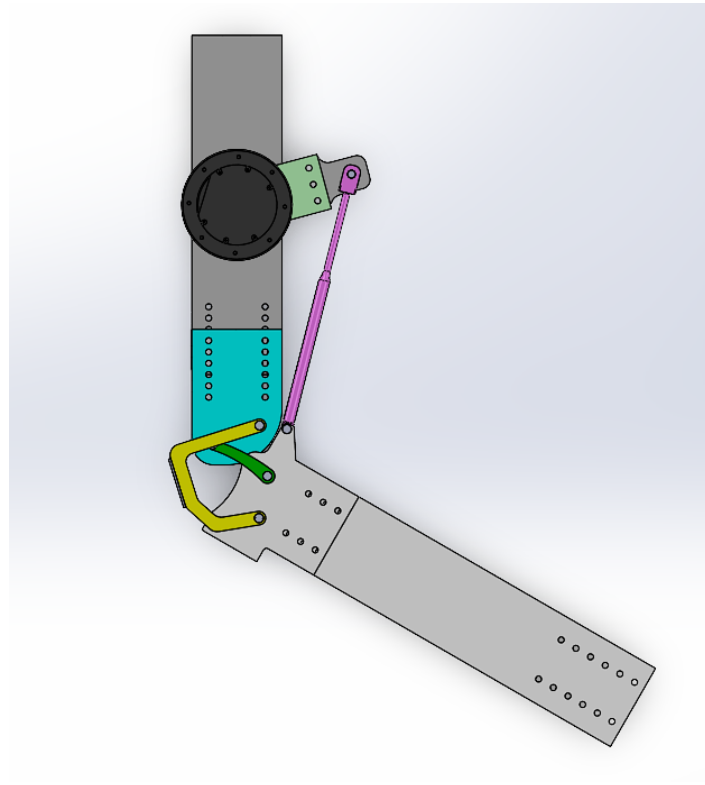


Figure 5.6: CAD assembly of the knee exoskeleton joint at approximately $F = 60^\circ$ knee flexion (if achievable in the current design).

In future work, the CAD model can be used to export the joint positions and link geometries to multibody simulation software, enabling dynamic analysis and the study of actuator requirements and contact forces between the exoskeleton and the limb.

5.2 Physical prototype implementation

After finalising the CAD model, the design was transferred to a physical prototype using fused-deposition-modelling (FDM) 3D printing. The transition from a virtual CAD assembly to a working physical device proved to be non-trivial: features that look correct in CAD do not always translate to a robust, wearable mechanism. This section describes the iterative process of building, testing, and refining the prototype, and documents the obstacles that were encountered along the way.

5.2.1 Prototype versions and iterations

Several iterations of the exoskeleton were produced before arriving at the configuration used for testing. Each version was driven either by a manufacturing failure, a kinematic interference, or a structural weakness discovered during handling and motion testing. The main iterations are summarised below.

Version 1 — first full assembly. The first complete prototype was printed using standard PLA filament with default infill settings on a hobby-grade FDM printer. This version proved the overall layout of the crossed four-bar mechanism and the shell interfaces, but revealed two immediate problems: (i) several pairs of moving parts came into contact during flexion, mechanically blocking the mechanism well before the intended range of motion was reached, and (ii) the printed plastic showed visible flex and creep around the bolted joints under modest hand loads, indicating that the PLA + low-infill combination was too weak for a load-bearing wearable device.

Version 2 — interference fixes and revised shell geometry. In the second iteration, the inner edges of the shells and the link bodies were locally trimmed to remove the contact points identified in Version 1. The bolt-hole spacing on the shells was also revised to better match the optimised link lengths L_1 – L_4 from Chapter 4. After reprinting, the mechanical stops were eliminated and the mechanism could travel through its full intended range without binding.

Version 3 — material upgrade and structural reinforcement. Even after the interference issues were solved, several parts continued to fail under load: links cracked near the bolt holes, and the shells deformed when the straps were tightened around the leg. To address this, the prototype was reprinted using Bambu Lab PLA on a Bambu Lab printer with higher infill density and improved layer adhesion. The structural integrity of the parts improved noticeably, although, as discussed in Section 5.2.4, the device still remained mechanically the weakest link of the overall system.

Version 4 — final integrated prototype. The final version of the prototype combines the geometry of Version 2 with the upgraded Bambu material from Version 3, plus the dedicated motor mount and strap holders described in Sections 5.2.2–5.2.3. This is the configuration used for the experiments reported in Chapter 6.

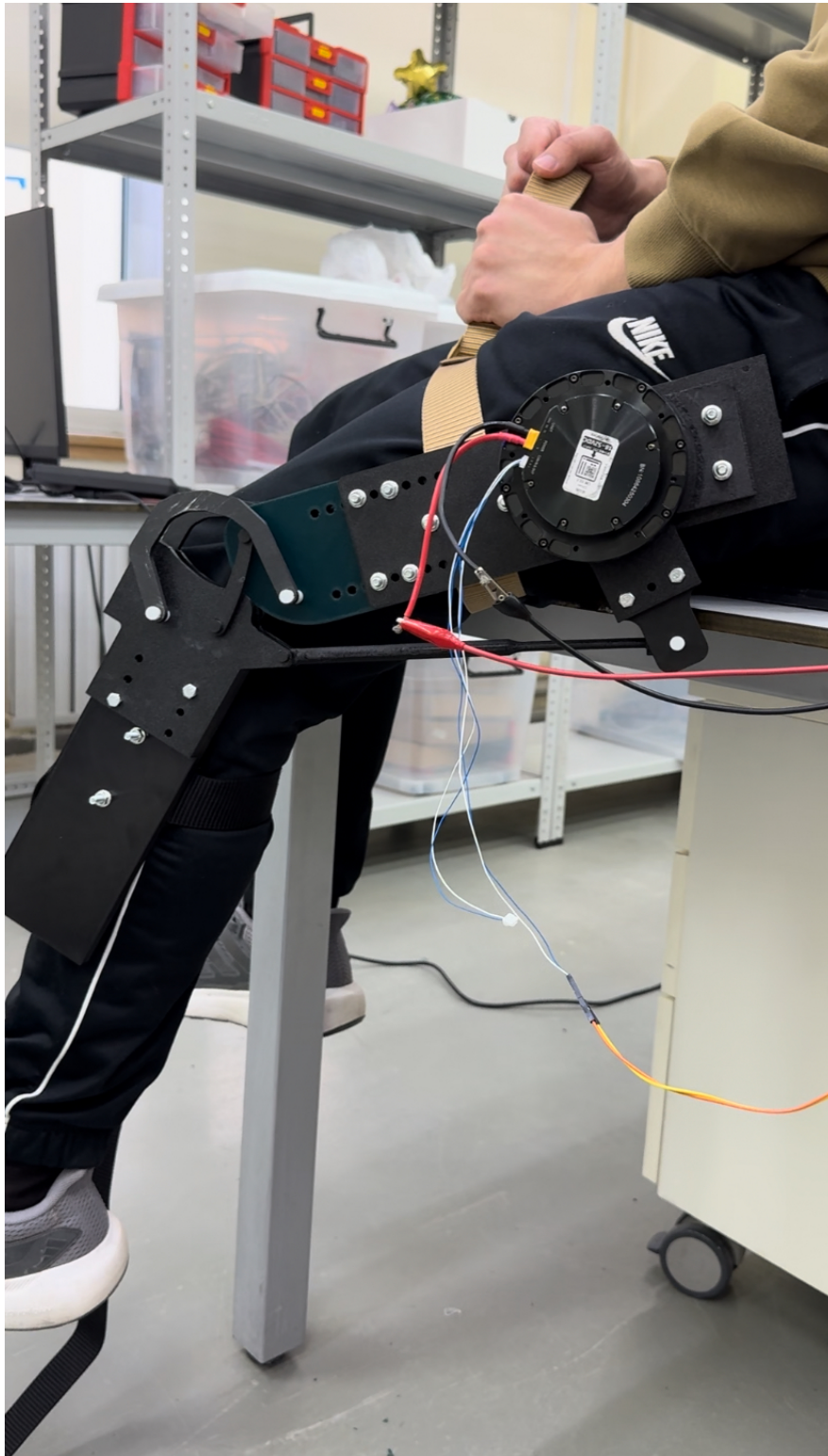


Figure 5.7: Final assembled prototype of the crossed four-bar knee exoskeleton used for control experiments.

5.2.2 Stationary motor mount and shaft drive

In the original CAD concept (Section 5.1.1) the rotary actuator was assumed to be mounted coaxially with one of the revolute joints of the four-bar mechanism, so that the motor body

would rotate together with the moving links. While conceptually simple, this configuration was found to be impractical for our prototype: the AK80-6 actuator is comparatively heavy, and attaching its body to the moving frame would have introduced a large inertial load and increased the bending stress on the printed plastic links during flexion and extension.

To resolve this, a dedicated motor-mount subassembly was designed that keeps the actuator body stationary relative to the thigh shell, while the motor output shaft drives the four-bar mechanism through an intermediate connection. In this arrangement only the lightweight shaft adapter and the linkage itself move during operation; the bulk of the motor mass remains fixed to the thigh frame. This change had three practical benefits: it reduced the moving inertia of the mechanism, it removed the cyclic bending load that the original co-rotating mount would have applied to the printed parts, and it simplified the routing of the CAN-bus and power cables, which no longer have to follow the moving link.

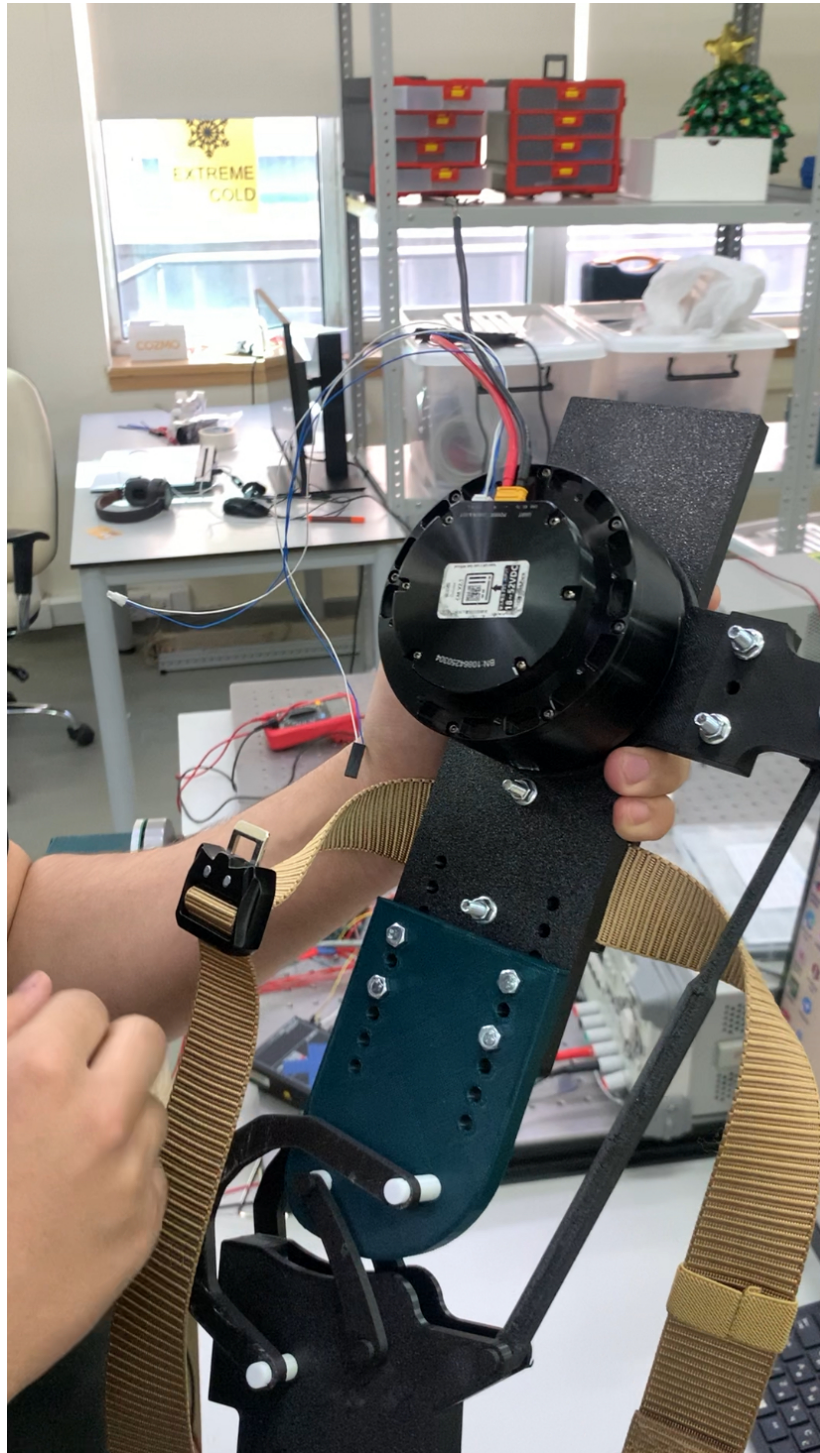


Figure 5.8: Stationary motor-mount subassembly. The AK80-6 body is rigidly attached to the thigh frame, while only the output shaft drives the crossed four-bar mechanism.

5.2.3 Strap holders and patient-contact protection

Early fitting of the prototype on a wearer revealed two ergonomic problems that were not visible in CAD: there were no defined locations for the straps, which made the device shift on the leg during motion, and the heads of the bolts that join the printed parts pressed directly against the skin and clothing of the wearer, producing localised pressure points.

To address both issues simultaneously, two pairs of dedicated strap-holder features were added on the outer surfaces of the thigh and shank shells. These features serve two purposes:

- **Strap routing.** They provide flat slots through which the Velcro straps are threaded, fixing the position of the straps on the shells and preventing them from sliding along the leg during flexion and extension.
- **Bolt-head protection.** They are positioned and shaped so that the protruding bolt heads of the underlying joints sit inside a recess and no longer come into direct contact with the wearer. The strap holder effectively acts as a soft spacer between the metal hardware and the leg.

The strap-holder features were printed as separate parts and bolted onto the shells, which made it possible to iterate on their geometry without reprinting the main shell components.



Figure 5.9: Strap-holder features added to the outer surfaces of the thigh and shank shells. They route the Velcro straps and shield the wearer from the bolt heads of the underlying joints.

5.2.4 Obstacles encountered during prototyping

Several obstacles emerged during the transition from CAD to a working device. The most important ones are listed below, since they directly shaped the choice of control strategy described in Chapter 6.

Mechanical interference between printed parts. As mentioned above, the first prototype suffered from several unintended mechanical stops where moving parts touched non-moving parts before the intended range of motion was reached. These could not be predicted

reliably from the CAD assembly alone, because they depended on small print tolerances and on the stack-up of multiple parts. Fixing them required local trimming of the shells and links and a full reprint of the affected components.

Material strength and broken parts. The standard PLA used in early iterations was not strong enough for the loads applied during normal handling and motion testing. Several failures were observed, including cracks propagating from bolt holes in the links and delamination between layers in the shells. Photographs of the failed components are shown in Figure 5.10. Switching to Bambu Lab PLA with higher infill noticeably improved structural integrity, but the printed plastic still represented the mechanical weak point of the overall system. This finding strongly influenced our decision (discussed in Chapter 6) to operate the motor in a conservative, position-limited regime rather than pursuing aggressive torque control.

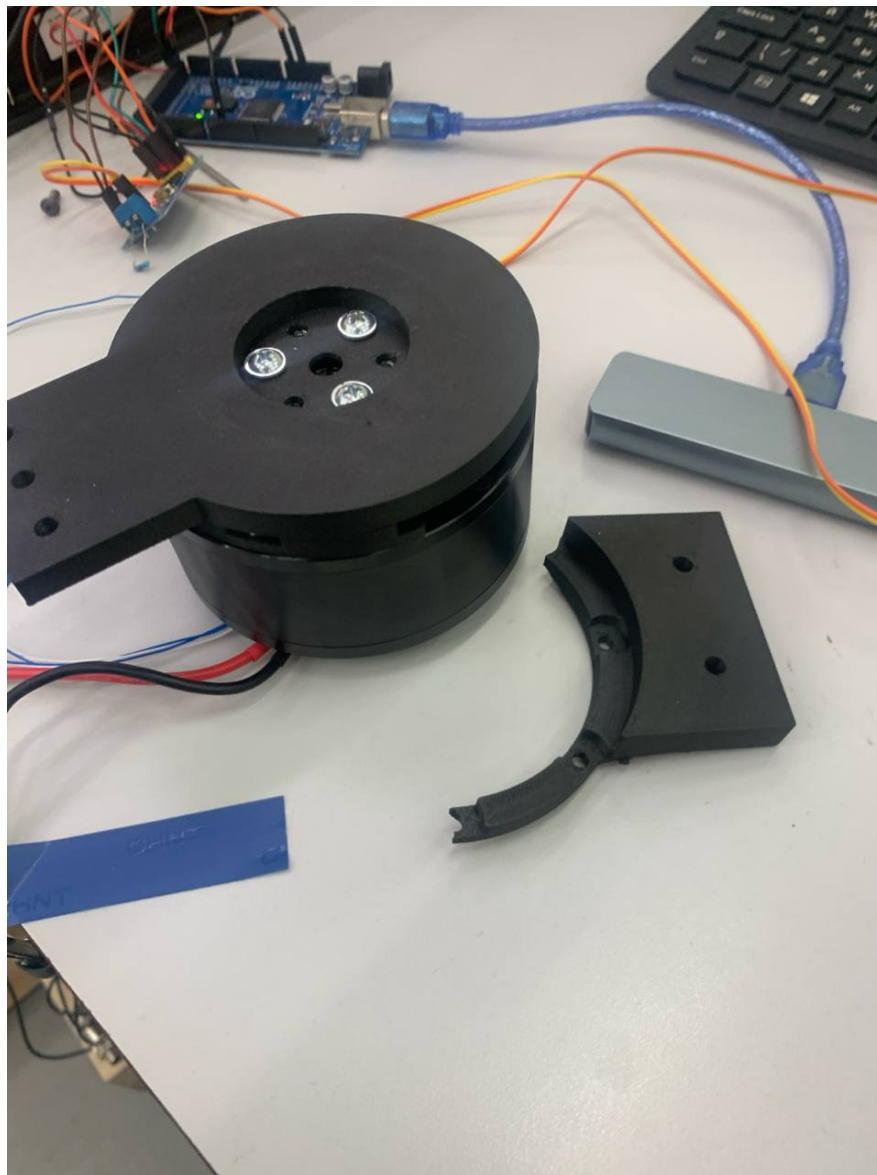


Figure 5.10: Examples of failures observed during prototyping: cracks originating at bolt holes and delamination of the printed shells.

Print orientation and reprints. Some parts had to be reprinted multiple times because the original print orientation placed the layer lines along the direction of maximum stress, which led to early delamination. Reorienting the parts so that the layer lines were perpendicular to the principal load direction, together with the move to Bambu PLA, gave the most reliable results.

Overall assessment of the prototype. At the end of the iteration cycle the prototype was functional and could be used for closed-loop motor experiments, but it remained, by an honest assessment, fragile. It was suitable for benchtop testing of the control strategy and for demonstrating the kinematic concept, but not yet robust enough to be loaded with a representative patient leg or to be used outside a controlled laboratory environment. Strengthening the structure — for example, by replacing critical printed parts with machined aluminium — is therefore identified in Chapter 10 as one of the highest-priority items for future work.

Chapter 6: Running the Exoskeleton

The previous chapters describe the design of the crossed four-bar knee mechanism and the iterative process of building a physical prototype. The next logical step is to bring the prototype to life: drive it with a motor, observe how it behaves under closed-loop control, and document the limitations that emerged once a real load was applied to a real, 3D-printed structure. This chapter describes the hardware setup, the control software, the safety considerations that shaped the test protocol, and the outcomes of the experimental runs.

6.1 Hardware setup

The actuated prototype combines an off-the-shelf high-torque robotics motor, a microcontroller running a custom firmware, and a current-limited bench power supply. The intention was to keep the electrical setup as simple and auditable as possible, so that any unexpected behaviour could be traced back to either the firmware or the mechanical structure rather than to obscure power-electronic effects.

6.1.1 Actuator

The mechanism is driven by a CubeMars AK80-6 quasi-direct-drive actuator with an integrated 1:6 planetary gearbox. The actuator was selected for three reasons: it is back-drivable, which is desirable for safe interaction with a patient's leg; it accepts MIT-style impedance commands over CAN, which expose position, velocity, stiffness, damping, and feedforward torque in a single message; and it provides position and velocity feedback at high rate, which is needed for closed-loop control.

6.1.2 Microcontroller and CAN interface

A standard Arduino Mega 2560 board is used as the main controller, together with an MCP2515 CAN-bus controller and a TJA1050 transceiver mounted on a breakout board. The MCP2515 is connected to the Arduino over SPI. This combination is widely supported, well-documented, and easy to debug from a serial terminal, which made it well suited for a first-time integration with the AK80-6.

6.1.3 Power supply

The motor is powered from a benchtop laboratory power supply, configured with a voltage limit of 20 V and a current limit of 2 A. The current limit is the most important parameter: it deliberately caps the maximum torque the motor can apply, so that even if the firmware commands a large torque the motor cannot pull more than a fixed amount of mechanical

power. Given the fragility of the printed structure, this hard-coded current limit was treated as the primary safety feature of the entire setup.

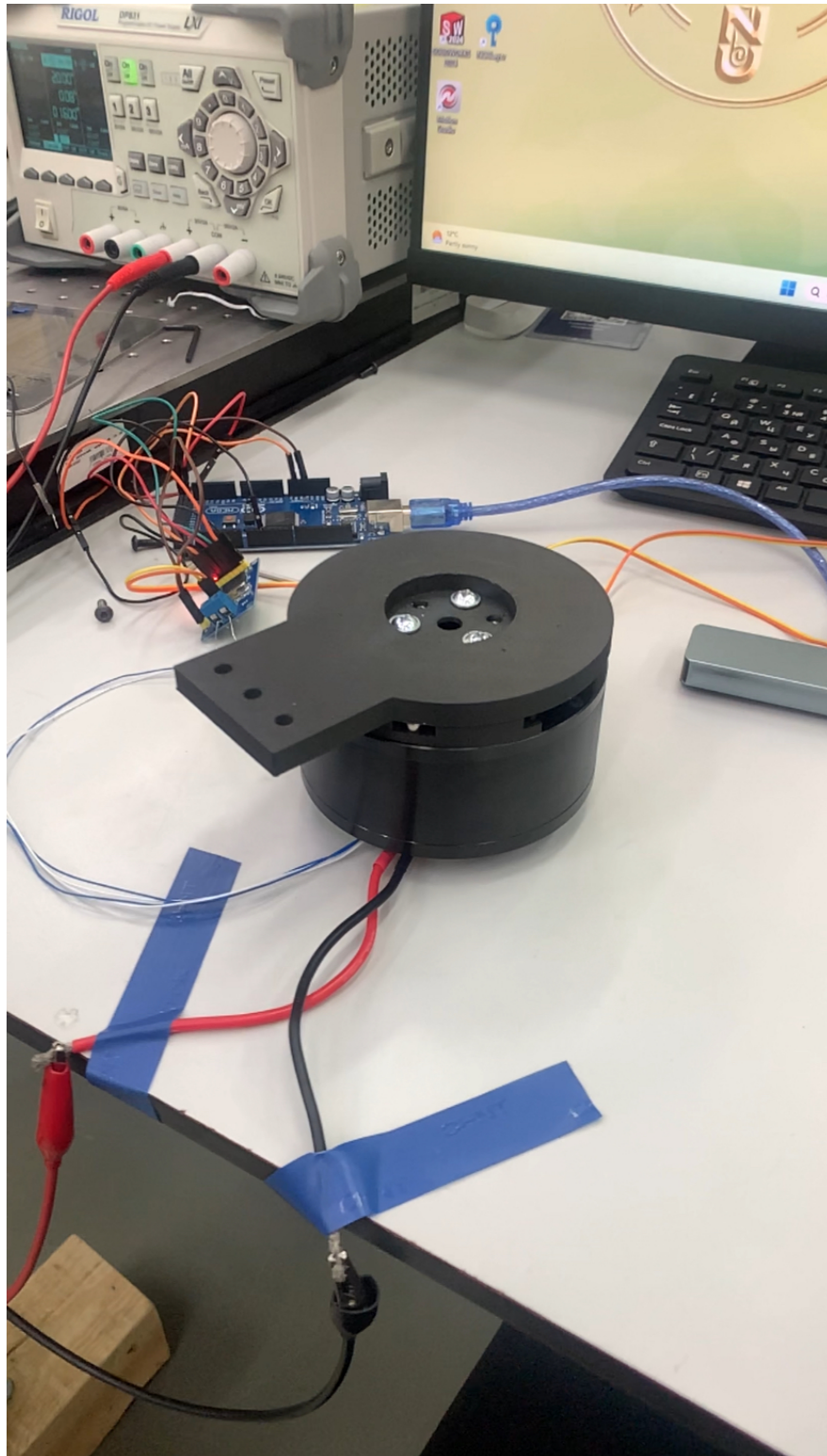


Figure 6.1: Hardware setup: AK80-6 actuator, Arduino Mega with MCP2515 CAN shield, and laboratory power supply configured for 20 V / 2 A operation.

6.2 Control software

The control firmware is implemented in C++ and runs on the Arduino at 50 Hz (control loop) and 1 Mbit/s (CAN bus). The full source code is included as an appendix; this section summarises its structure and the design choices behind it.

6.2.1 MIT impedance protocol

Each control cycle the firmware packs a target position, target velocity, stiffness K_p , damping K_d , and feedforward torque τ_{ff} into an 8-byte CAN frame using the MIT bit packing convention. The motor applies the closed-loop law

$$\tau = K_p (p_{\text{des}} - p) + K_d (v_{\text{des}} - v) + \tau_{\text{ff}}, \quad (6.1)$$

internally, and returns measured position and velocity in the same frame format. This is the standard impedance-control interface for the AK80-6 family and allowed us to trade off compliance and tracking accuracy by tuning a small number of high-level gains.

6.2.2 From torque sensing to restricted sinusoidal motion

Initially, the plan was to close an outer torque-sensing loop around the motor, so that the patient's interaction force could be detected and the machine would assist only as much as needed. A first version of the firmware therefore included a tube-MPC structure with three error zones, an assist-as-needed (AAN) gain that adapted slowly over time, and a gravity compensation feedforward term based on a simple shank-mass model. The relevant parameters and the corresponding code paths are still present in the source listing in the appendix.

In practice, however, the printed structure was too fragile to be exposed to the loads that even a moderate torque-sensing controller could request. A small overshoot or a momentary lag in feedback could have driven the motor hard enough to crack a link or damage a shell, which had already been observed during prototyping (Section 5.2.4). After several near-failures during benchtop tuning, the testing strategy was deliberately downgraded to a much more conservative regime:

- the motor follows a pre-computed sinusoidal reference trajectory between two soft position limits, with no torque sensing in the loop;
- the stiffness K_p and damping K_d used in Equation (6.1) are kept at modest, fixed values;
- the feedforward torque is set to zero, so that the only torque the motor produces is whatever the inner closed loop needs to track the reference position;
- the supply current limit (Section 6.1.3) caps the maximum mechanical effort regardless of what the firmware asks for.

This combination is far less sophisticated than the original AAN / gravity-compensation scheme, but it was the only configuration that could be operated repeatedly without risking destruction of the printed parts. It is the configuration used for all experimental runs reported in this chapter.

6.2.3 Reference trajectory

The reference trajectory is a smooth cosine-shaped flexion-extension profile

$$r(t) = p_{\text{home}} + \frac{1}{2} \Delta p (1 - \cos(2\pi t/T)), \quad (6.2)$$

where Δp is the peak-to-peak travel and T is the cycle period. This shape is continuous in both position and velocity, which is important for the printed mechanism because step changes or hard reversals would excite high-frequency loads in the plastic. The home position p_{home} was set so that the mechanism started in approximately full extension, and the reference was clamped to the soft position limits in software. The motor was tested at several different amplitudes (Δp) and several different cycle periods T to explore how the mechanism behaved across a range of speeds.

6.3 Test protocol and observations

Each test run followed the same simple protocol. The exoskeleton was placed on the bench, with the leg side of the device unloaded (no human leg attached) so that any failure would damage only the printed parts and not the user. The power supply was enabled with the 2 A current limit; the Arduino was reset; the motor was zeroed; and a single command through the serial monitor started the sinusoidal trajectory. The serial monitor was used both to send the start/stop commands and to capture a running CSV stream of the measured position, the reference position, and the tracking error at 5 Hz. Screenshots of these serial sessions are included in Chapter 8.

Across the runs, the motor was able to drive the four-bar mechanism through the prescribed sinusoidal motion at several different amplitudes and frequencies. Within the limits set by the soft position constraints and the current cap, the actuator produced visibly smooth flexion and extension and the mechanism remained in its crossed configuration throughout each cycle. At the same time, the printed structure clearly was the limiting factor of the system: parts of the device flexed visibly under the inertia of the moving links, and any attempt to push the operating regime — larger amplitudes, faster cycles, or stiffer position control — noticeably increased the risk of cracking a link or stripping a printed bolt seat.

The next chapter formalises how we evaluate the kinematic performance of the mechanism (the original tracking-of-the-anatomical-ICR question), and the chapter after that combines the kinematic results with the experimental observations described here.

Chapter 7: Evaluation Methodology

The purpose of this chapter is to define how the performance of the proposed crossed four-bar knee mechanism is evaluated against the anatomical motion of the human knee and against a conventional single-axis hinge joint. All quantities are expressed in the tibia-fixed (Z, Y) coordinate frame introduced in Chapter 3 and used consistently in the kinematic analysis of Chapter 4. The evaluation is based on comparing trajectories of the instantaneous centre of rotation (ICR) and on quantitative misalignment metrics derived from these trajectories. This chapter describes the data used, the construction of the relevant trajectories, and the scalar measures of misalignment that are reported later in the results chapter.

7.1 Anatomical reference trajectory

In Chapter 3 we constructed an anatomical sagittal-plane reference trajectory in the tibia-fixed (Z, Y) frame over a specified range of knee flexion angles. After shifting the reference so that the knee joint centre at full extension $F = 0^\circ$ lies at the origin, the anatomical trajectory is written as

$$\mathbf{p}_{\text{anat}}(F) = \begin{bmatrix} Z_{\text{anat}}(F) \\ Y_{\text{anat}}(F) \end{bmatrix}, \quad F \in [0, F_{\text{max}}], \quad (7.1)$$

where $Z_{\text{anat}}(F)$ and $Y_{\text{anat}}(F)$ denote the anterior-posterior and proximal-distal coordinates of the anatomical reference point at flexion angle F . For numerical evaluation, the continuous functions are sampled at a discrete set of flexion angles $F_i \in \{F_1, \dots, F_N\}$. These sampled points act as the ground truth against which all mechanical trajectories are compared.

7.2 Mechanism trajectory

The crossed four-bar mechanism analysed in Chapter 4 has a mechanical ICR defined as the intersection of the two crossed links $L_2 = \overline{AC}$ and $L_3 = \overline{DB}$ in the quadrilateral $ABCD$. For a given knee flexion angle F , an input mapping prescribes the input angle of link L_2 , $\alpha = \alpha(F)$, and the remaining joint locations are computed via planar position analysis in the (Z, Y) frame.

In the revised implementation, the mechanism configuration is obtained by planar position analysis using geometric constraints: the moving joint B is computed by circle-circle intersection in the (Z, Y) plane (Chapter 4.3). This makes feasibility explicit: if the circles do not intersect at some flexion sample, the mechanism cannot realise that configuration for the tested geometry.

Because circle-circle intersection yields two candidates (open and crossed configurations), a continuity rule is applied across flexion samples to maintain the physically relevant crossed branch and avoid discontinuities in the trajectory.

Once the mechanism configuration is determined for each flexion sample, the mechanical ICR at flexion angle F is computed as the intersection point of the two straight lines passing through AC and DB , respectively. This yields the mechanical trajectory

$$\mathbf{p}_{\text{mech}}(F) = \begin{bmatrix} Z_{\text{mech}}(F) \\ Y_{\text{mech}}(F) \end{bmatrix}, \quad (7.2)$$

which is sampled at the same discrete angles F_i .

7.3 Misalignment metric for the mechanism

To quantify how closely the mechanical ICR follows the anatomical reference, we define a misalignment metric based on the Euclidean distance between corresponding points of the two trajectories in the (Z, Y) plane. For each flexion sample F_i , the instantaneous misalignment of the mechanism is

$$d_{\text{mech}}(F_i) = \|\mathbf{p}_{\text{mech}}(F_i) - \mathbf{p}_{\text{anat}}(F_i)\|_2 = \sqrt{(Z_{\text{mech},i} - Z_{\text{anat},i})^2 + (Y_{\text{mech},i} - Y_{\text{anat},i})^2}. \quad (7.3)$$

This choice of metric has two important properties:

- It measures misalignment in physical units of length (millimetres), which is directly related to the relative motion between the exoskeleton joint and the user's anatomical joint centre.
- It treats deviations in the anterior–posterior and proximal–distal directions symmetrically, without privileging one axis.

In addition to the angle-dependent distance $d_{\text{mech}}(F_i)$, we report two scalar summaries over the full flexion range:

$$d_{\text{RMS,mech}} = \sqrt{\frac{1}{N} \sum_{i=1}^N d_{\text{mech}}(F_i)^2}, \quad d_{\text{max,mech}} = \max_{i=1,\dots,N} d_{\text{mech}}(F_i). \quad (7.4)$$

The root-mean-square value $d_{\text{RMS,mech}}$ characterises the overall average misalignment, while $d_{\text{max,mech}}$ indicates the worst misalignment encountered within the flexion range considered.

7.4 Hinge baseline and misalignment

To provide a baseline for comparison, we also consider a conventional single-axis hinge joint. In this configuration the exoskeleton knee joint centre is fixed at a constant position

$$\mathbf{p}_{\text{hinge}} = \begin{bmatrix} Z_h \\ Y_h \end{bmatrix} \quad (7.5)$$

in the tibia-fixed (Z, Y) frame. Since the hinge position does not move with flexion, its trajectory is constant: $\mathbf{p}_{\text{hinge}}(F_i) = \mathbf{p}_{\text{hinge}}$ for all i . The instantaneous hinge misalignment at

flexion angle F_i is defined as

$$d_{\text{hinge}}(F_i) = \|\mathbf{p}_{\text{hinge}} - \mathbf{p}_{\text{anat}}(F_i)\|_2 = \sqrt{(Z_h - Z_{\text{anat},i})^2 + (Y_h - Y_{\text{anat},i})^2}. \quad (7.6)$$

As for the mechanism, we compute RMS and maximum hinge misalignment:

$$d_{\text{RMS,hinge}} = \sqrt{\frac{1}{N} \sum_{i=1}^N d_{\text{hinge}}(F_i)^2}, \quad d_{\text{max,hinge}} = \max_{i=1,\dots,N} d_{\text{hinge}}(F_i). \quad (7.7)$$

In Chapter 7, hinge misalignment is plotted alongside baseline and optimised mechanism misalignment to provide a direct reference for the benefit of an anatomically inspired moving ICR.

Chapter 8: Results

This chapter reports the trajectory tracking performance of the crossed four-bar knee mechanism. Using the evaluation methodology defined in Chapter 7, we compare the mechanical instantaneous centre of rotation (ICR) trajectory against the anatomical reference trajectory and a single-axis hinge baseline.

In the updated model, parameter tuning was used to improve ICR tracking. For all plots below, we show three cases: (i) the anatomical reference, (ii) the baseline mechanism (before tuning), and (iii) the optimised mechanism (after tuning). The hinge baseline is included where relevant as a reference for a fixed rotation centre.

8.1 Trajectory comparison

8.1.1 Sagittal-plane ICR path

Figure 8.1 compares the sagittal-plane ICR trajectory in the tibia-fixed (Z, Y) frame. This plot is the most direct visual indicator of how well the mechanism reproduces the target ICR motion, since it shows the full path rather than individual coordinate components.

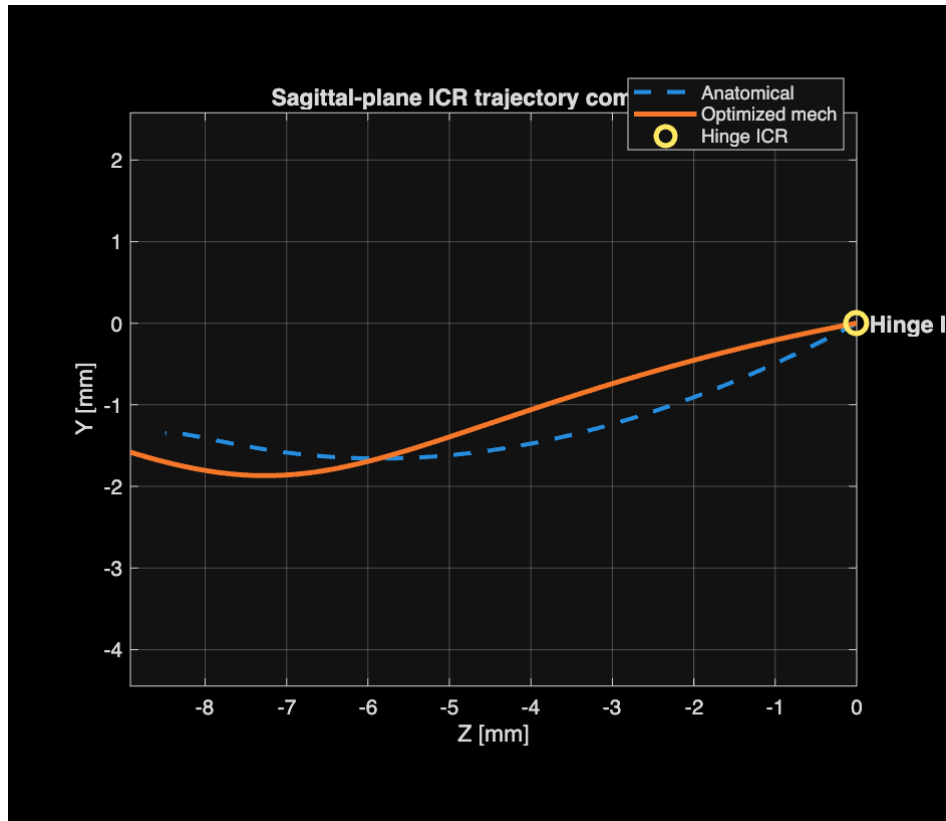


Figure 8.1: Sagittal-plane ICR trajectory comparison in the tibia-fixed (Z, Y) frame: anatomical reference, baseline mechanism, and optimised mechanism.

The baseline mechanism exhibits a clear geometric mismatch relative to the anatomical trajectory: the baseline path follows a noticeably different arc and does not remain close to the anatomical curve over the full flexion range. After optimisation, the mechanical ICR path moves substantially closer to the anatomical reference. The improved overlap indicates that the tuned geometry and input mapping reduce trajectory error over the flexion range considered.

8.1.2 Component-wise comparison: $Z(F)$ and $Y(F)$

To interpret which aspects of the motion contribute most to the trajectory mismatch, we also compare the anterior–posterior component $Z(F)$ and the proximal–distal component $Y(F)$ as functions of flexion. These component plots are useful because two different mechanisms can produce similar-looking paths while still differing substantially in one coordinate direction.

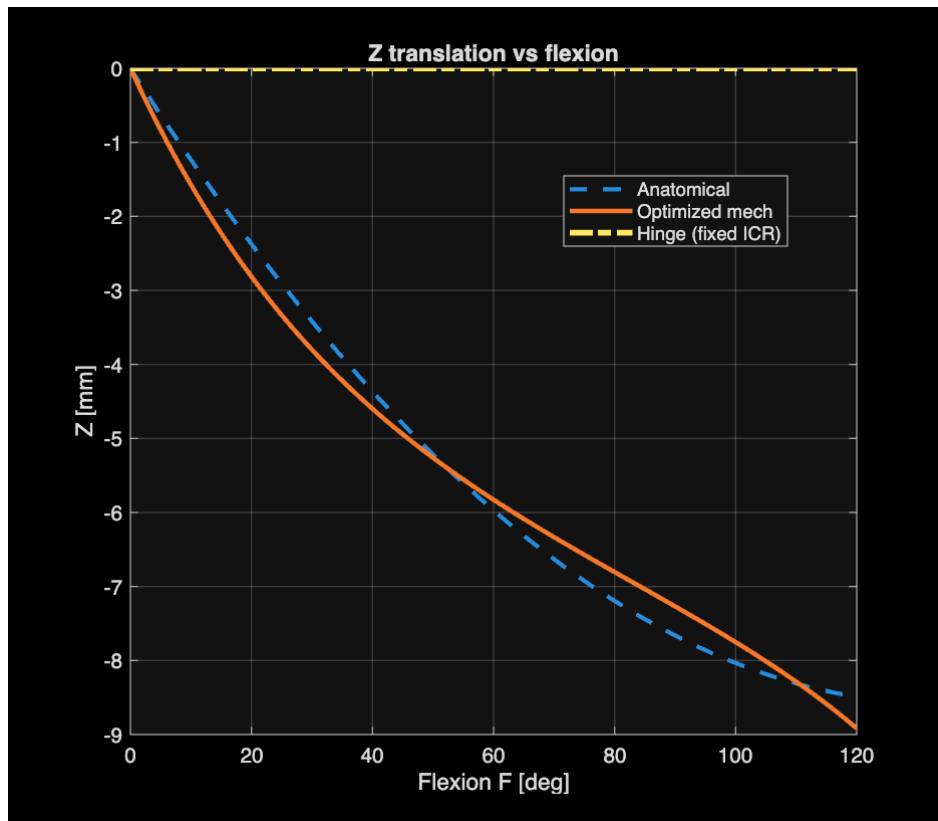


Figure 8.2: Anterior–posterior translation $Z(F)$ comparison: anatomical reference vs baseline and optimised mechanism, with hinge baseline.

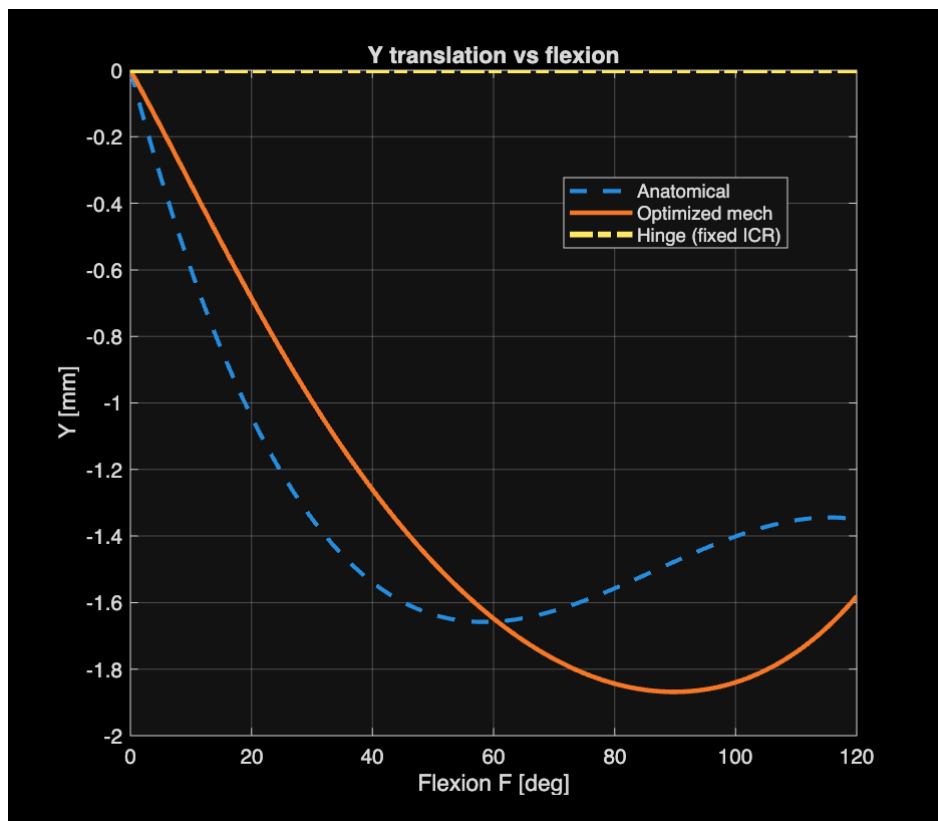


Figure 8.3: Proximal–distal translation $Y(F)$ comparison: anatomical reference vs baseline and optimised mechanism, with hinge baseline.

In the baseline mechanism, the deviation from the anatomical trajectory is visible in both components, indicating that the mismatch is not limited to a single direction. The tuned mechanism reduces these deviations: the optimised $Z(F)$ and $Y(F)$ curves remain close to the anatomical reference over most of the flexion range. In contrast, the hinge baseline remains offset because a fixed rotation centre cannot reproduce the anatomical variation of the ICR with flexion.

8.2 Misalignment versus flexion

While trajectory plots provide qualitative insight, misalignment curves quantify the error at each flexion angle. Figure 8.4 reports the pointwise Euclidean distance between the mechanical and anatomical ICR points in the (Z, Y) plane, as defined in Chapter 7.

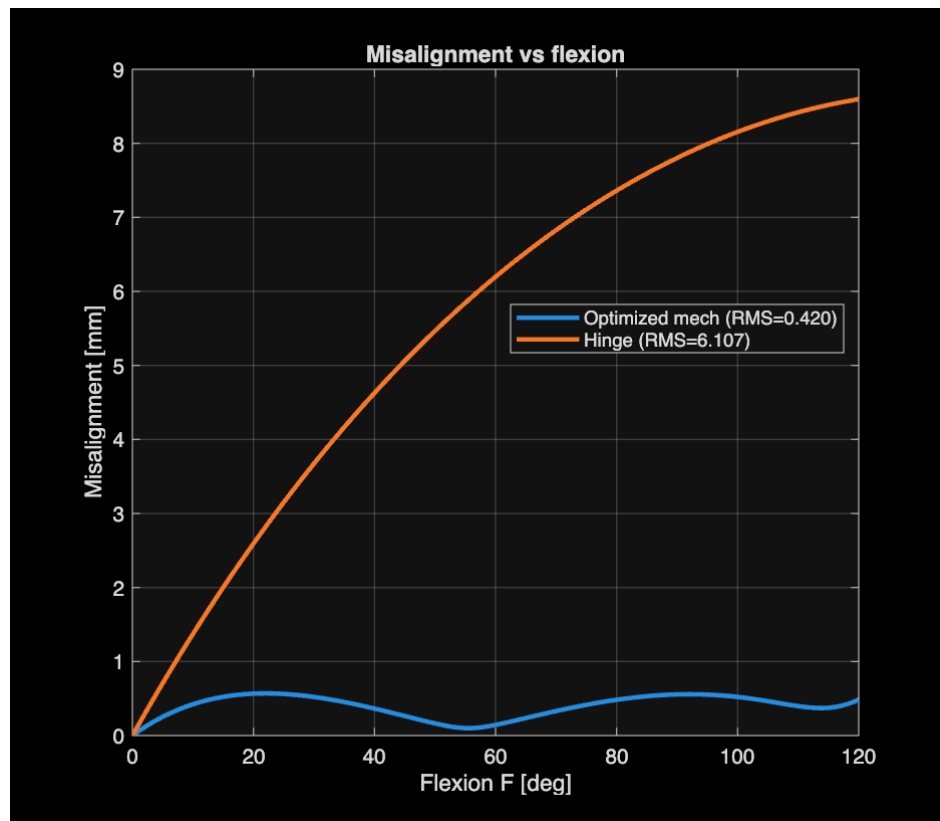


Figure 8.4: Pointwise misalignment versus flexion: comparison of baseline mechanism, optimised mechanism, and hinge baseline against the anatomical reference.

The baseline curve shows substantially larger error across the flexion range, confirming that the initial parameter set does not reproduce the anatomical ICR trajectory well. After optimisation, the misalignment curve drops dramatically and remains low across most flexion angles. This indicates that tuning improves both average tracking quality and worst-case deviations. The hinge baseline remains comparatively high because a fixed centre of rotation is fundamentally unable to follow a moving anatomical ICR trajectory.

8.3 Configuration continuity check

A key requirement of the crossed four-bar mechanism is that the numerical solution remains on the crossed branch over the full flexion range. Branch switching would introduce discontinuities in the internal configuration and produce unrealistic jumps in the computed ICR trajectory. Figure 8.5 shows the internal angle/orientation variable used in the continuity-based selection procedure.

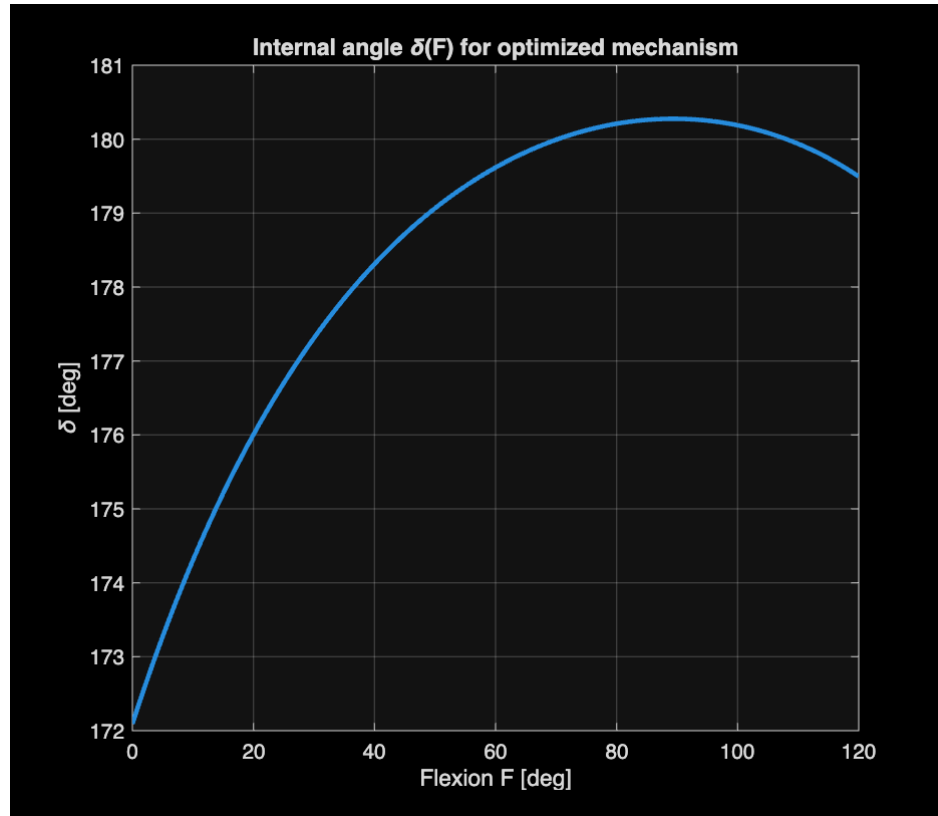


Figure 8.5: Internal angle/orientation variable used for continuity-based selection of the crossed branch as a function of flexion.

The continuity plot indicates smooth evolution across flexion without abrupt jumps that would suggest branch switching. This supports the validity of the computed optimised trajectory, since discontinuous configuration changes would generate unrealistic mechanical ICR behaviour and invalidate the comparison.

8.4 Experimental motor-control results

The kinematic and misalignment results above are computed in software, from the geometric model of the mechanism. The experiments described in Chapter 6 provide a complementary, qualitative check that the physical prototype can actually be driven through the intended range of motion under closed-loop control. This section reports those experimental observations.

8.4.1 Tracking the sinusoidal reference

In every test run, the AK80-6 was able to drive the four-bar mechanism along the cosine-shaped reference trajectory of Equation (6.2) from the soft lower limit to the soft upper limit and back without losing CAN feedback or hitting an unintended mechanical stop. The serial monitor stream confirmed that the measured position followed the reference smoothly throughout each cycle. A representative screenshot of a serial session, showing the running CSV stream of time, reference angle, measured angle, and tracking error, is given in Figure 8.6.

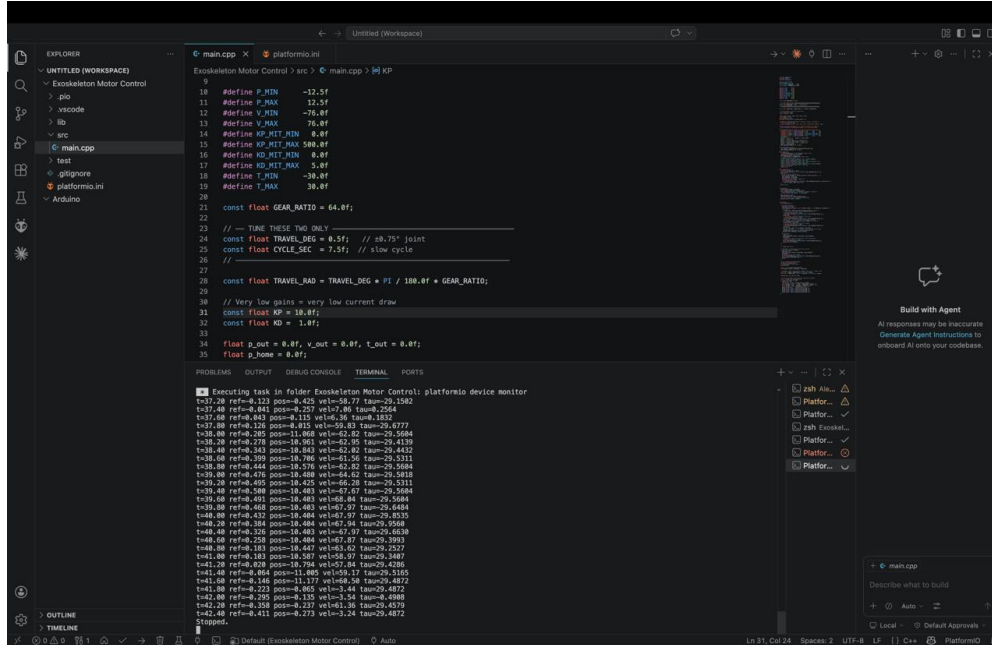


Figure 8.6: Screenshot of a serial-monitor session during a sinusoidal flexion-extension run. The CSV columns correspond to time, reference angle, measured angle, tracking error, and internal control variables of the firmware.

8.4.2 Behaviour at different amplitudes and frequencies

The motor was tested with several combinations of peak-to-peak amplitude Δp and cycle period T , spanning slow, large-amplitude motions suitable for a rehabilitation cadence as well as faster, smaller-amplitude motions used to probe the responsiveness of the controller. Within the range of values that we judged safe for the printed structure, the mechanism tracked the sinusoidal reference cleanly in all configurations. As expected, larger amplitudes and shorter periods produced larger inertial loads on the links and were therefore approached cautiously; for the fastest cases tested, visible elastic flex of the printed parts could be observed through the cycle, even though the controller itself remained stable.

8.4.3 The structure as the limiting factor

The most important experimental observation is not about the controller at all: it is that the printed structure, rather than the motor or the firmware, sets the practical operating envelope of the device. As discussed in Section 6.2.2, the original plan to use torque sensing and an assist-as-needed scheme had to be abandoned because the loads such a controller would have demanded were beyond what the printed parts could safely tolerate. The conservative

position-controlled regime described in Chapter 6 was therefore the regime under which all data presented above were collected.

This observation has two implications for how the experimental results should be interpreted. First, the apparently clean tracking of the sinusoidal reference does *not* demonstrate that the device is ready for use with a patient leg: it demonstrates only that, with the leg unloaded and within deliberately conservative position limits, the motor and CAN stack work as intended. Second, any future quantitative evaluation of the assistance behaviour — for example, measuring tracking error against a human-led reference under AAN control — depends much more on strengthening the structure than on improving the firmware. Replacing the weakest printed parts with metal counterparts is therefore identified in Chapter 10 as the highest-priority next step.

8.5 Summary of findings

Overall, the optimised crossed four-bar mechanism produces substantially improved agreement with the anatomical reference trajectory compared to the baseline mechanism. The improvement is visible in the sagittal-plane ICR path, in the component-wise $Z(F)$ and $Y(F)$ curves, and in the pointwise misalignment curve. The hinge baseline remains a useful reference demonstrating that a fixed rotation centre cannot capture the anatomical ICR variation. These results support using the tuned mechanism parameters for subsequent design stages, including CAD integration and fabrication.

In addition to these kinematic results, the experiments of Section 8.4 confirm that the assembled prototype can be driven through the intended range of motion under closed-loop position control. The motor and CAN stack are not the limiting factor of the current device; the printed structure is. Together, the kinematic and experimental findings indicate that the design is ready for a more ambitious mechanical revision (for example, a metal-link version) before moving on to richer control strategies and to evaluation with human users.

Chapter 9: Discussion

This chapter interprets the results of Chapter 8 along two axes. The first axis is kinematic: how well the optimised crossed four-bar mechanism reproduces the anatomical ICR trajectory of the knee joint. The second axis is experimental: what we learned from actually building and running the prototype, and what it implies for the next iteration of the design.

9.1 Quality of ICR matching

The primary aim of the kinematic design was to align the instantaneous centre of rotation (ICR) of the mechanical knee joint with the anatomical ICR of the human tibio-femoral joint across a physiologically relevant flexion range. The results in Chapter 8 show that, after parameter tuning, this objective is largely achieved.

The comparison plots in Figures 8.1, 8.2 and 8.3 show that the optimised mechanical ICR closely overlaps the anatomical ICR throughout the 0–120° flexion range, both in the sagittal-plane path and in the individual $Z(F)$ and $Y(F)$ components. The pointwise misalignment curve in Figure 8.4 reduces this comparison to a single number per flexion sample and confirms that the optimised mechanism stays sub-millimetre across most of the range, while the hinge baseline grows almost monotonically with flexion to nearly an order of magnitude larger error at full flexion. In the same units used by Yu et al. [10], the optimised mechanism is therefore well within the band where shear forces and cuff sliding are expected to be substantially reduced relative to a fixed-centre hinge.

Beyond the numerical agreement, the mechanism exhibits several structural properties that were verified during the analysis:

- The mechanical ICR trajectory is *stable and smooth*. It remains inside the quadrilateral $ABCD$ for all simulated flexion angles and does not exhibit sudden jumps or reversals, which is important for predictable kinematics and for avoiding singular configurations.
- The continuity of the internal angle $\delta(F)$ (Figure 8.5) confirms that the linkage stays on the intended crossed branch throughout the flexion range. Branch switching would have invalidated the comparison; its absence means that the reported misalignment values are mechanically realisable.
- The mechanism provides a non-circular, non-fixed ICR, in contrast to a simple hinge. This is the conceptual property the design set out to deliver, and the quantitative results confirm it.

Taken together, these observations support the central claim of the project: that a planar crossed four-bar linkage, with link lengths and input mapping chosen by an explicit optimisation, can track the sagittal ICR of the knee far more closely than a fixed-axis hinge. The remaining gap between the optimised curve and the anatomical reference is small in absolute terms and is, in our judgement, dominated by simplifications discussed in Section 9.3 rather than by a fundamental limitation of the mechanism.

9.2 From kinematics to a working prototype

The kinematic analysis above is performed entirely in software. The work reported in Chapters 5.1 and 6 attempts the much harder step of turning that geometry into a working device, and the results of that effort are mixed in a way that is worth discussing explicitly.

On the positive side, the assembled prototype is a complete, integrated system: a 3D-printed mechanical structure that implements the optimised linkage, a stationary motor mount that drives the mechanism through its shaft, ergonomic strap holders that double as bolt-head shields, and a microcontroller-based control stack that talks to the AK80-6 over CAN. The device can be commanded to follow a sinusoidal flexion-extension profile and does so reliably at several amplitudes and frequencies. None of these pieces existed at the time of the original CAD chapter, and bringing them together was the main engineering effort of the second half of the project.

On the negative side, the prototype is mechanically fragile, and this fragility forced a substantial retreat from the original control ambitions. The intended controller was an interaction-aware, assist-as-needed scheme with explicit gravity compensation; what actually runs on the device is a conservative, position-controlled sinusoidal trajectory with fixed stiffness and damping and zero feedforward torque. The reasons for this retreat are documented in Sections 5.2.4 and 6.2.2: PLA links cracked at bolt holes, shells delaminated, and the structure visibly flexed under loads that a proper torque-sensing controller would have considered routine. Switching to Bambu Lab PLA and reorienting prints helped, but did not eliminate the underlying issue, which is that printed plastic is the wrong material for the load-bearing parts of a wearable robot.

The honest reading of the experimental work is therefore that the device is a solid *kinematic* demonstrator and a partial *control* demonstrator. It validates that the optimised linkage geometry can be embedded into a real assembly and driven in a controlled manner, but it does not yet demonstrate the closed-loop interaction-aware behaviour that a clinical rehabilitation device would need. The most direct way to close this gap, as discussed in Section 10, is not to refine the firmware but to replace the weakest printed parts with metal so that a more demanding controller can actually be tested without destroying the hardware.

9.3 Limitations and simplifications

Several modelling and methodological choices limit the conclusions that can be drawn from the present study. The most important limitations are summarised below.

Planar approximation

Both the anatomical model in Chapter 3 and the crossed four-bar mechanism in Chapter 4 are treated as planar systems in the sagittal plane. The polynomials from Gao et al. include varus/valgus and internal/external rotation components, but these are effectively collapsed into a two-dimensional (Z, Y) trajectory. Similarly, the exoskeleton mechanism is assumed to operate in a perfectly sagittal plane with no out-of-plane motion, play or misalignment. In reality, the human knee exhibits coupled three-dimensional motion, and soft tissues introduce compliance and variability that are not represented here.

Simplified anatomical data

The anatomical ICR trajectory is based on polynomial fits derived from a specific experimental dataset. In this work we use those polynomials directly, without subject-specific scaling or statistical analysis. As a result, the “anatomical” reference represents an average behaviour rather than the anatomy of a particular user. Moreover, the tibia-fixed coordinate frame is defined according to the conventions in Gao et al., and small differences in landmark placement or axis definition would change the numerical values of $(Z_{\text{anat}}, Y_{\text{anat}})$.

Single misalignment metric

Misalignment was quantified using only the Euclidean distance between the anatomical and mechanical ICRs in the (Z, Y) plane. This scalar measure is convenient and intuitive, but it does not capture all aspects of user comfort or joint safety. For example, it does not distinguish between misalignment in the anterior-posterior and proximal-distal directions, does not account for contact forces or pressure distributions under the cuffs, and does not consider dynamic effects during walking or other tasks. A more complete evaluation would require coupling the kinematic model with contact models and, ideally, experimental measurements on human subjects.

Idealised mechanical structure

The CAD model in Chapter 5.1 represents an ideal rigid structure. The analysis neglects backlash in the joints, compliance in the shells and straps, manufacturing tolerances, and the influence of actuators and transmission elements. In practice, the as-built prototype shows visible compliance in the printed parts under load, as documented in Section 5.2.4. These effects may alter the effective ICR of the assembled device and partly explain why a richer control strategy could not yet be safely tested.

Conservative control regime

The experiments in Chapter 6 were performed with a deliberately simplified control law: position tracking of a sinusoidal reference, with fixed stiffness and damping, no torque sensing in the loop, and no assist-as-needed adaptation. The torque-sensing and AAN scaffolding remains in the source code but is not exercised. As a consequence, the experimental results presented in Section 8.4 should be read as a feasibility check on the actuator, the CAN stack, and the mechanical assembly, not as a clinical-style evaluation of assistance performance.

No human-subject testing

No data were collected with a human leg loaded into the device. All experiments were performed on the bench with the device unloaded. Effects related to fit, soft-tissue motion, strap compliance, and patient interaction are therefore outside the scope of the present results and are identified as future work.

Summary

In summary, the optimised crossed four-bar mechanism achieves a close quantitative match to the sagittal-plane anatomical ICR trajectory and a large reduction in misalignment relative to a fixed-axis hinge. The physical prototype, while functional, is mechanically fragile and forced the control strategy to be simplified well below its original ambition. The limitations listed above — in particular the planar approximation, the simplified anatomical data, the conservative control regime, and the absence of human-subject testing — define a clear set of priorities for future iterations of the device.

Chapter 10: Conclusion and Future Work

In this project we designed, analysed, built, and ran a crossed four-bar knee exoskeleton mechanism intended to better follow the anatomical motion of the human knee joint. The work spans the full chain from biomechanical modelling to a working actuated prototype.

On the modelling side, a tibia-fixed planar coordinate frame was defined and used consistently to describe both the anatomical knee joint centre trajectory and the motion generated by the mechanism. Using this frame, we obtained a polynomial description of the anatomical instantaneous centre of rotation and derived kinematic equations for the four-bar linkage, including a loop-closure relation and a Freudenstein-type scalar equation. These equations were solved numerically over a prescribed flexion range to compute the mechanical ICR trajectory, and an optimisation procedure was used to tune the link lengths and input mapping. An evaluation methodology was proposed to compare the anatomical trajectory with the optimised four-bar mechanism and with a simple hinge joint, using Euclidean distance in the (Z, Y) plane as the misalignment metric. The optimised mechanism reduces the RMS misalignment by more than an order of magnitude relative to the hinge baseline across the full $0\text{--}120^\circ$ flexion range.

On the implementation side, the mechanism was embedded into a 3D-printed exoskeleton structure with thigh and shank shells, a stationary motor mount that drives the linkage through the AK80-6 output shaft, and dedicated strap holders that route the Velcro straps and shield the wearer from bolt heads. The prototype went through several iterations before arriving at a configuration that worked: mechanical interferences between printed parts had to be removed, and the original PLA had to be replaced by Bambu Lab PLA before the structure could survive normal handling and motion testing. The actuated system uses an Arduino Mega with an MCP2515 CAN interface to talk to the AK80-6 over a 1 Mbit/s CAN bus using the MIT impedance protocol, with a benchtop power supply limited to 20 V and 2 A as a hard upper bound on applied torque. Although the original control plan included torque sensing, gravity compensation and an assist-as-needed adaptation, fragility of the printed structure forced us to operate the device in a much simpler regime: a pre-computed sinusoidal flexion-extension trajectory tracked with fixed stiffness and damping. Within this conservative regime, the motor reliably drove the mechanism through its intended range of motion at multiple amplitudes and frequencies.

Overall, the project delivers a complete modelling and evaluation framework for crossed four-bar knee exoskeletons, a working actuated prototype, and a clear-eyed account of where the current device succeeds and where it does not. The kinematic concept is validated; the mechanical implementation is functional but fragile; and the control software is ready for richer strategies as soon as the hardware can support them.

Future work

Several directions can be pursued to improve and extend this work. They are listed roughly in the order in which we would tackle them.

- **Strengthen the structure.** Replace the most heavily loaded printed parts (the four links and the shell regions around the bolt holes) with machined aluminium or with metal-reinforced printed parts. This is the single change with the largest expected impact: it unlocks the use of the controllers that already exist in the firmware and removes the fragility that limited the present experiments.
- **Re-enable the full control stack.** Once the structure is trustworthy, restore the tube-MPC error zones, the assist-as-needed gain adaptation, and the gravity-compensation feedforward in the active firmware path. Validate them on the bench with a dummy load before any human-subject testing.
- **Quantitative experimental validation.** Use motion capture or an external angle sensor to measure the actual ICR of the assembled device under load and compare it directly to the kinematic prediction of Chapter 4. This converts the present qualitative feasibility check into a quantitative validation of the model.
- **Human-subject testing.** After the above, evaluate the device with healthy volunteers wearing it through a flexion–extension protocol, instrumenting the cuff–skin interface to measure shear and pressure, and comparing the measured values against a fixed-axis hinge baseline.
- **Three-dimensional and subject-specific extensions.** Lift the planar approximation by including varus/valgus and internal/external rotation, and replace the generic anatomical polynomials with subject-specific data obtained from imaging or motion capture.
- **Alternative mechanisms.** Explore adjustable or reconfigurable linkages, and alternative topologies such as remote-centre-of-motion mechanisms, that may offer better kinematic fidelity while remaining wearable and compact.

Together, these directions chart a clear path from the current proof-of-concept device to a clinically usable knee rehabilitation exoskeleton. The hardest item on the list, in our experience, is the first one — and the rest only become meaningful once it is done.

Appendix A: Motor-Control Firmware

This appendix lists the firmware that runs on the Arduino Mega in Chapter 6. The code includes scaffolding for the originally planned tube-MPC, assist-as-needed and gravity-compensation features (constants, helper functions, and state variables) for documentation purposes. As discussed in Section 6.2.2, the experimental runs used only the inner sinusoidal-tracking path with fixed stiffness and damping; the AAN adaptation, error zones and feedforward torque are not exercised in the configuration that produced the data in Section 8.4.

```
1 #include <Arduino.h>
2 #include <SPI.h>
3 #include <mcp_can.h>
4
5 // -- Hardware -----
6 #define CAN_CS_PIN 53
7 MCP_CAN CAN(CAN_CS_PIN);
8
9 const uint8_t MOTOR_ID = 0x01;
10 const uint32_t FEEDBACK_ID = 0x01;
11
12 // -- Rehabilitation parameters -----
13 const float TRAVEL_DEG = 90.0f;
14 const float POS_MIN_DEG = 0.0f;
15 const float POS_MAX_DEG = 90.0f;
16 const float CYCLE_SEC = 10.0f;
17 const float GEAR_RATIO = 6.0f; // AK80-6
18
19 // -- MIT protocol limits -----
20 #define P_MIN -12.5f
21 #define P_MAX 12.5f
22 #define V_MIN -76.0f
23 #define V_MAX 76.0f
24 #define KP_MIT_MIN 0.0f
25 #define KP_MIT_MAX 500.0f
26 #define KD_MIT_MIN 0.0f
27 #define KD_MIT_MAX 5.0f
28 #define T_MIN -30.0f
29 #define T_MAX 30.0f
30
31 // -- Control constants -----
32 const float KP_FULL = 150.0f;
33 const float KP_MIN_C = 50.0f;
34 const float KD_CTRL = 3.0f;
35
36 // -- Tube MPC parameters -----
37 const float TUBE_INNER_DEG = 8.0f;
38 const float TUBE_OUTER_DEG = 22.0f;
39 const float TUBE_INNER = TUBE_INNER_DEG * PI / 180.0f;
40 const float TUBE_OUTER = TUBE_OUTER_DEG * PI / 180.0f;
41
```

```

42 // -- AAN parameters -----
43 const float AAN_MIN      = 0.10f;
44 const float AAN_DECREASE = 0.002f;
45 const float AAN_INCREASE = 0.008f;
46
47 // -- Gravity compensation -----
48 const float SHANK_MASS = 5.0f;
49 const float SHANK_LEN  = 0.50f;
50
51 // -- State -----
52 float p_out  = 0.0f;
53 float p_home = 0.0f;
54 bool  motor_on = false;
55 bool  running  = false;
56 unsigned long last_fb      = 0;
57 unsigned long session_start = 0;
58 float aan_gain = 1.0f;
59 float sum_err  = 0.0f;
60 long  n_samp   = 0;
61 float max_err  = 0.0f;
62 int   n_cyc    = 0;
63
64 // -- MIT protocol helpers -----
65 unsigned int float_to_uint(float x, float mn, float mx, int bits) {
66     x = constrain(x, mn, mx);
67     return (unsigned int)((x - mn) / (mx - mn) * (float)((1L << bits) - 1L));
68 }
69
70 float uint_to_float(unsigned int x, float mn, float mx, int bits) {
71     return (float)x / (float)((1L << bits) - 1L) * (mx - mn) + mn;
72 }
73
74 void send_mit(float pos, float vel, float kp, float kd, float tff) {
75     unsigned int p_int = float_to_uint(pos, P_MIN, P_MAX, 16);
76     unsigned int v_int = float_to_uint(vel, V_MIN, V_MAX, 12);
77     unsigned int kp_int = float_to_uint(kp, KP_MIT_MIN, KP_MIT_MAX, 12);
78     unsigned int kd_int = float_to_uint(kd, KD_MIT_MIN, KD_MIT_MAX, 12);
79     unsigned int t_int  = float_to_uint(tff, T_MIN, T_MAX, 12);
80     byte buf[8];
81     buf[0] = p_int >> 8;
82     buf[1] = p_int & 0xFF;
83     buf[2] = v_int >> 4;
84     buf[3] = ((v_int & 0xF) << 4) | (kp_int >> 8);
85     buf[4] = kp_int & 0xFF;
86     buf[5] = kd_int >> 4;
87     buf[6] = ((kd_int & 0xF) << 4) | (t_int >> 8);
88     buf[7] = t_int & 0xFF;
89     CAN.sendMsgBuf(MOTOR_ID, 0, 8, buf);
90 }
91
92 void send_cmd(byte cmd) {
93     byte buf[8] = {0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,cmd};
94     CAN.sendMsgBuf(MOTOR_ID, 0, 8, buf);
95 }

```

```

96
97 bool read_fb() {
98     if (CAN.checkReceive() != CAN_MSGAVAIL) return false;
99     unsigned long rxId; byte len, buf[8];
100     CAN.readMsgBuf(&rxId, &len, buf);
101     if ((rxId & 0x7FF) != FEEDBACK_ID) return false;
102     if (len < 5) return false;
103     uint16_t p_raw = ((uint16_t)buf[1] << 8) | (uint16_t)buf[2];
104     p_out = uint_to_float(p_raw, P_MIN, P_MAX, 16);
105     last_fb = millis();
106     return true;
107 }
108
109 // -- Reference trajectory (cosine ramp) -----
110 float TRAVEL_RAD;
111
112 float ref_pos(float t_sec) {
113     float phase = fmod(t_sec, CYCLE_SEC) / CYCLE_SEC;
114     return p_home + (TRAVEL_RAD * 0.5f) * (1.0f - cosf(TWO_PI * phase));
115 }
116
117 float ref_vel(float t_sec) {
118     float phase = fmod(t_sec, CYCLE_SEC) / CYCLE_SEC;
119     return (TRAVEL_RAD * PI / CYCLE_SEC) * sinf(TWO_PI * phase);
120 }
121
122 // -- Gravity compensation -----
123 float gravity_tff(float p_rel) {
124     float theta = constrain(p_rel / GEAR_RATIO, 0.0f, PI / 2.0f);
125     float t_joint = SHANK_MASS * 9.81f * SHANK_LEN * cosf(theta);
126     return constrain(t_joint / GEAR_RATIO, T_MIN, T_MAX);
127 }
128
129 // -- Motor connect / stop -----
130 bool connect_motor() {
131     unsigned long id; byte l, b[8];
132     while (CAN.checkReceive() == CAN_MSGAVAIL) CAN.readMsgBuf(&id, &l, b);
133     last_fb = 0;
134     for (int i = 0; i < 20; i++) {
135         send_cmd(0xFD); delay(50);
136         send_cmd(0xFC); delay(100);
137         unsigned long t0 = millis();
138         while (millis() - t0 < 150) read_fb();
139         if (millis() - last_fb < 300) return true;
140     }
141     return false;
142 }
143
144 void stop_motor() {
145     running = false; motor_on = false;
146     send_mit(p_out, 0, 0, KD_CTRL, 0);
147     delay(100);
148     send_cmd(0xFD);
149     float mean_err = (n_samp > 0) ? sum_err/(float)n_samp*180.0f/PI : 0.0f;

```

```

150 Serial.println("=== SESSION REPORT ===");
151 Serial.print("Cycles: "); Serial.println(n_cyc);
152 Serial.print("Max err: "); Serial.print(max_err*180.0f/PI, 1); Serial.
    println(" deg");
153 Serial.print("Mean err: "); Serial.print(mean_err, 1); Serial.
    println(" deg");
154 Serial.println("S = new session");
155 }
156
157 // -- Setup -----
158 void setup() {
159     TRAVEL_RAD = TRAVEL_DEG * PI / 180.0f;
160     Serial.begin(115200);
161     delay(500);
162     while (Serial.available()) Serial.read();
163     while (CAN.begin(MCP_ANY, CAN_1000KBPS, MCP_8MHZ) != CAN_OK) {
164         Serial.println("CAN error..."); delay(500);
165     }
166     CAN.setMode(MCP_NORMAL);
167     CAN.init_Mask(0, 0, 0x00000000);
168     CAN.init_Mask(1, 0, 0x00000000);
169     for (int i = 0; i < 6; i++) CAN.init_Filt(i, 0, 0x00000000);
170     Serial.println("AK80-6 Knee Rehab | S=start X=stop R=reset");
171 }
172
173 // -- Main loop -----
174 void loop() {
175     while (read_fb()) {}
176
177     if (Serial.available()) {
178         delay(10); char c = 0;
179         while (Serial.available()) {
180             char r = Serial.read();
181             if (r >= 32 && c == 0) c = toupper(r);
182         }
183         if (c == 'R') { aan_gain=1.0f; sum_err=0; n_samp=0; max_err=0; n_cyc=0; }
184         if (c == 'S') {
185             if (motor_on) { Serial.println("Already running. X = stop."); return; }
186             if (!connect_motor()) { Serial.println("ERROR: no motor response.");
187                 return; }
188             send_cmd(0xFE); delay(200);
189             unsigned long id; byte l, b[8];
190             while (CAN.checkReceive() == CAN_MSGAVAIL) CAN.readMsgBuf(&id, &l, b);
191             for (int i = 0; i < 30; i++) { read_fb(); delay(20); }
192             p_home=0.0f; p_out=0.0f; motor_on=true; running=true;
193             aan_gain=1.0f; session_start=millis();
194             sum_err=0; n_samp=0; max_err=0; n_cyc=0;
195             Serial.println("Running!");
196             Serial.println("t,ref_deg,pos_deg,err_deg,cyc");
197         }
198         if (c == 'X') { stop_motor(); return; }
199     }
200     if (!motor_on || !running) return;

```

```

201
202 // Watchdog
203 if (millis() - last_fb > 2000) {
204     Serial.println("No feedback -- emergency stop.");
205     stop_motor(); return;
206 }
207
208 // 50 Hz control loop
209 static unsigned long last_ctrl = 0;
210 unsigned long now = millis();
211 if (now - last_ctrl < 20) return;
212 last_ctrl = now;
213
214 float t_sec = (now - session_start) * 0.001f;
215 float r_pos = ref_pos(t_sec);
216 float r_vel = ref_vel(t_sec);
217 float p_rel = p_out - p_home;
218 float error = r_pos - p_out;
219 float abserr = fabsf(error);
220
221 // Cycle counter
222 static float prev_r = 0.0f;
223 float mid = p_home + TRAVEL_RAD * 0.5f;
224 if (r_pos >= mid && prev_r < mid) n_cyc++;
225 prev_r = r_pos;
226
227 // Position limits and command
228 float pos_min_rad = p_home + POS_MIN_DEG * PI / 180.0f;
229 float pos_max_rad = p_home + POS_MAX_DEG * PI / 180.0f;
230 float r_clamped = constrain(r_pos, pos_min_rad, pos_max_rad);
231 send_mit(r_clamped, r_vel, KP_MIN_C, KD_CTRL, 0.0f);
232
233 // Statistics
234 sum_err += abserr; n_samp++;
235 if (abserr > max_err) max_err = abserr;
236
237 // CSV output at 5 Hz
238 static unsigned long last_print = 0;
239 if (now - last_print >= 200) {
240     last_print = now;
241     Serial.print(t_sec, 2); Serial.print(",");
242     Serial.print((r_pos - p_home) * 180.0f / PI, 2); Serial.print(",");
243     Serial.print(p_rel * 180.0f / PI, 2); Serial.print(",");
244     Serial.print(error * 180.0f / PI, 2); Serial.print(",");
245     Serial.println(n_cyc);
246 }
247 }

```

Listing A.1: AK80-6 knee-rehabilitation firmware (main.cpp) running on Arduino Mega with MCP2515 CAN interface.

Bibliography

- [1] Charles D. A. Wolfe. “The impact of stroke”. In: *British Medical Bulletin* 56.2 (2000), pp. 275–286. DOI: 10.1258/0007142001903120.
- [2] Tao Fan et al. “Effects of exoskeleton rehabilitation robot training on neuroplasticity and lower limb motor function in patients with stroke”. In: *BMC Neurology* 25 (2025), p. 193. DOI: 10.1186/s12883-025-04203-7.
- [3] Prakhar Jain et al. “Linear actuator-based knee exoskeleton for stand-sit-stand motions: a bond graph approach”. In: *Simulation: Transactions of the Society for Modeling and Simulation International* 98.7 (2022), pp. 627–644. DOI: 10.1177/00375497211073583.
- [4] Lin Wu et al. “Novel Design on Knee Exoskeleton with Compliant Actuator for Post-Stroke Rehabilitation”. In: *Sensors* 25.19 (2025), p. 6652. DOI: 10.3390/s25196652.
- [5] Kirby Ann Witte, Andreas M. Fatschel, and Steven H. Collins. “Design of a Lightweight, Tethered, Torque-Controlled Knee Exoskeleton”. In: *2017 International Conference on Rehabilitation Robotics (ICORR)*. London, UK: IEEE, July 2017, pp. 1–6. DOI: 10.1109/ICORR.2017.8009370.
- [6] Hyo-gon Kim, Sangdeok Park, and Hyeongcheol Lee. “Design of a novel knee joint for an exoskeleton with good energy efficiency for load-carrying augmentation”. In: *Journal of Mechanical Science and Technology* 29.11 (2015), pp. 4693–4699. DOI: 10.1007/s12206-015-1039-3.
- [7] Shikai Jin, Bin Liu, and Zhuo Wang. “A Bionic Knee Exoskeleton Design with Variable Stiffness via Rope-Based Artificial Muscle Actuation”. In: *Biomimetics* 5.4 (2020), p. 65. DOI: 10.3390/biomimetics5040065.
- [8] Hongyan Tang et al. “Design and optimization of a novel sagittal-plane knee exoskeleton with remote-center-of-motion mechanism”. In: *Mechanism and Machine Theory* 194 (2024), p. 105570. DOI: 10.1016/j.mechmachtheory.2023.105570.
- [9] Taeyeon Kim, Mingoo Jeong, and Kyoungchul Kong. “Bioinspired Knee Joint of a Lower-Limb Exoskeleton for Misalignment Reduction”. In: *IEEE/ASME Transactions on Mechatronics* 21.6 (2016), pp. 2975–2983. DOI: 10.1109/TMECH.2016.2593942.
- [10] Shuangyue Yu et al. “Bio-inspired design of a self-aligning, lightweight, and highly-compliant cable-driven knee exoskeleton”. In: *Frontiers in Human Neuroscience* 16 (2022), p. 1018160. DOI: 10.3389/fnhum.2022.1018160.
- [11] Besir Celebi, Mustafa Yalcin, and Volkan Patoglu. “ASSISTON-KNEE: A Self-Aligning Knee Exoskeleton”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Tokyo, Japan: IEEE, Nov. 2013.
- [12] Frederick F. Buechel and Michael J. Pappas. *Principles of Human Joint Replacement: Design and Clinical Application*. 2nd ed. Cham: Springer, 2015. DOI: 10.1007/978-3-319-15311-7.
- [13] Won Hyuk Chang and Yun-Hee Kim. “Robot-assisted Therapy in Stroke Rehabilitation”. In: *Journal of Stroke* 15.3 (2013), pp. 174–181. DOI: 10.5853/jos.2013.15.3.174.
- [14] Moyao Gao et al. “Design and optimization of exoskeleton structure of lower limb knee joint based on cross four-bar linkage”. In: *AIP Advances* 11.6 (2021), p. 065124. DOI: 10.1063/5.0053899.