FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Interferential-Therapy-Based Functional Electrical Stimulation for Optimized Ligaments Regeneration

João Pedro Teixeira Calvo



Dissertation submitted in partial fulfillment of the requirements for the degree of Master in Bioengineering - Specialization in Biomedical Engineering

Supervisor: Prof. José Machado da Silva

October 30, 2026

Interferential-Therapy-Based Functional Electrical Stimulation for Optimized Ligaments Regeneration

João Pedro Teixeira Calvo

Master in Bioengineering - Specialization in Biomedical Engineering

October 30, 2026

Abstract

Anterior cruciate ligament (ACL) injuries are among the most common knee injuries, affecting more than 120,000 individuals annually in the United States alone. Post-reconstruction quadriceps rehabilitation is often limited by pain-related inhibition and incomplete voluntary activation, motivating stimulation-assisted approaches such as functional electrical stimulation (FES) and interferential current therapy (IFT). This dissertation addresses the design, implementation, and bench characterization of an EMG-triggered, IFT-based FES system for post-ACL-reconstruction rehabilitation, in which knee stimulation is contingent on the patient's volitional quadriceps activation.

The system integrates three subsystems: a single-supply surface EMG (sEMG) acquisition chain built around an INA326 instrumentation amplifier with DC-offset cancellation, active filtering, right-leg drive, and a mid-supply reference; a stimulation stage built around a TDA2002 transconductance amplifier driving a TST 10/020 step-up transformer; and a modular Python application that synthesizes the interferential waveform, digitizes sEMG through the laptop's sound card, and governs a four-stage state machine based on the block-wise RMS of the acquired signal.

The sEMG chain achieved a bandpass response between approximately 10 and 500 Hz and an input-referred noise of 0.72 µV RMS over 20–500 Hz. The stimulation circuit, bench-tested at a 4 kHz carrier against a 330 Ω surrogate load, yielded a cubic fit of maximum secondary current versus volume setting with $R^2 = 0.9961$, with secondary current below approximately 60 mA peak. The three subsystems were integrated and exercised end-to-end, with voluntary quadriceps contraction reliably triggering interferential stimulation within a timed session, demonstrating the functional feasibility of the closed loop. However, validation was limited to short, self-administered, non-patient sessions; the system remains open-loop regarding delivered current, and only premodulated IFC is currently deliverable. The prototype is therefore best understood as a characterized research platform rather than a validated therapeutic device.

Keywords: functional electrical stimulation, interferential current therapy, surface electromyography, anterior cruciate ligament, rehabilitation

Resumo

As lesões do ligamento cruzado anterior (LCA) estão entre as lesões do joelho mais comuns, afetando mais de 120 000 pessoas anualmente, apenas nos Estados Unidos. A reabilitação do quadríceps, após a reconstrução, é frequentemente limitada pela inibição relacionada com a dor e pela ativação voluntária incompleta, o que motiva o recurso a abordagens assistidas por estimulação, tais como a estimulação elétrica funcional (FES) e a terapia por corrente interferencial (IFT). Esta dissertação aborda a conceção, implementação e caracterização em laboratório de um sistema de FES baseado em IFT e acionado por EMG para a reabilitação do LCA apõs reconstrução cirúrgica, no qual a estimulação do joelho depende da ativação voluntária do quadríceps pelo próprio paciente.

O sistema integra três subsistemas: uma cadeia de aquisição de EMG de superfície de alimentação única (sEMG) construída em torno de um amplificador de instrumentação INA326 com cancelamento de desvio de corrente contínua, filtragem ativa, filtragem ativa do modo comum com um módulo *right leg drive* e uma referência de alimentação intermédia; um estágio de estimulação construído em torno de um amplificador de transcondutância TDA2002 que aciona um transformador elevador TST 10/020; e uma aplicação modular em Python que sintetiza a forma de onda interferencial, digitaliza o sEMG através da placa de som de um computador portátil e controla um sistema de quatro de quatro estágios com base no valor RMS por bloco do sinal adquirido.

A cadeia sEMG atingiu uma resposta de passa-banda entre aproximadamente 10 e 500 Hz e um ruído, em relação à entrada, de 0,72 µV RMS na gama de 20–500 Hz. O circuito de estimulação, testado em bancada com onda sinusoidal de 4 kHz contra uma carga simulada de 330 Ω, apresentou um ajuste cúbico da corrente secundária máxima em função da percentagem de volume com R^2 = 0,9961, com a corrente secundária abaixo de aproximadamente 60 mA de pico. Os três subsistemas foram integrados e testados de ponta a ponta, tendo a contração voluntária do quadríceps desencadeado de forma fiável a estimulação interferencial no âmbito de uma sessão cronometrada, o que demonstrou a viabilidade funcional do circuito fechado. No entanto, a validação limitou-se a sessões curtas, autoadministradas e sem a participação de pacientes; o sistema permanece em circuito aberto no que diz respeito à corrente fornecida, sendo que, atualmente, apenas é possível aplicar IFC pré-modulada. O protótipo deve, portanto, ser entendido como uma plataforma de investigação caracterizada, em vez de um dispositivo terapêutico validado.

Palavras-chave: estimulação elétrica funcional, terapia com correntes interferenciais, eletromiografia de superfície, ligamento cruzado anterior, reabilitação

Framing within the Sustainable Development Goals

Engineering projects are increasingly expected to be assessed not only on technical merit but also on their contribution to broader societal goals. The United Nations' 2030 Agenda, structured around 17 Sustainable Development Goals (SDGs), offers a widely adopted reference for this kind of assessment.

The EMG-triggered, IFT-based FES system developed in this dissertation intersects with this agenda at three points, summarized in Table 1: the clinical motivation behind the device (health and rehabilitation), the technical approach taken to build it (low-cost innovation), and a longer-term implication of that approach (equitable access). Each row lists the specific target invoked and indicators that could, in principle, be used to assess the project's contribution, though most remain unmeasured in the present bench-validation stage of the work.

Table 1: Framing of the sustainable development goals in this project.

SDG	**Target**	**Contribution**	**Indicators**
3 – Good Health and Well-being	3.8 – Achieve universal health coverage, including access to quality essential healthcare services	Couples voluntary quadriceps effort to synchronized IFT, supporting neuromuscular re-education after ACL reconstruction	Comfort vs. conventional FES; recovery outcomes
9 – Industry, Innovation and Infrastructure	9.5 – Research and innovation capacity	Repurposes a computer sound card as the acquisition/stimulation interface, a lower-cost alternative to dedicated lab instrumentation	Bill-of-materials cost vs. commercial systems; bench-validated performance (Ch. 4)
10 – Reduced Inequalities	10.2 – Empower and promote the social, economic and political inclusion of all	Lower hardware cost and complexity could extend IFT-based rehabilitation to resource-limited clinics	Unit cost vs. commercial equivalents; feasibility in low-resource settings

This framing is motivated by the scale and cost of ACL rehabilitation itself: recovery is typically long, physically demanding, and dependent on consistent patient engagement, and access

to the equipment that supports it is not uniform across care settings (Subsection 2.1.1). A system capable of delivering targeted, effort-contingent stimulation on inexpensive hardware speaks to that disparity in principle. That said, the present dissertation reports a bench-characterized proof of concept rather than a clinically validated device (Chapter 4), so the health and equity contributions listed in Table 1 should be read as the rationale motivating the design choices made in Chapter 3, not as outcomes demonstrated by this work; closing that gap is identified as a priority for future work in Chapter 5.

Acknowledgements

I would like to begin by thanking my parents and my sister, who have been my anchor throughout this journey. Their unwavering support, especially during the most difficult moments, gave me the strength not only to complete this thesis, but to finish the course itself. Whatever I have accomplished during my degrees, I owe in no small part to them.

To Renata, my girlfriend, I owe a special debt of gratitude. No one contributed more to this thesis than she did, through her careful revision of the text, her help gathering materials, and, above all, her constant emotional support on the days when I needed it most. She has been by my side through every setback and every small victory of this work, and I consider myself truly fortunate to have both her and my family with me on this path.

I am sincerely grateful to Professor José Machado da Silva for his guidance throughout the development of this work. Thank you for your constant availability and for the direction you provided at every stage.

I would also like to thank Vítor, the laboratory technician, for his invaluable help with soldering, component sourcing, and all the behind-the-scenes work that made this project possible.

Finally, I want to thank my friends from the "Fora d'Água" group. The comradery, the good moments, and the memories we have built together are ones I will treasure forever, and I am grateful to have shared this journey alongside them.

João Calvo

"Tenho uma espécie de dever de sonhar sempre,
pois, não sendo mais, nem querendo ser mais,
que um espectador de mim mesmo, tenho que
ter o melhor espetáculo que posso."

Fernando Pessoa

Contents

1 Introduction **1**
- 1.1 Framework and motivation . . . 1
- 1.2 Objective . . . 2
- 1.3 Structure . . . 3

2 State of the Art **4**
- 2.1 The Anterior Cruciate Ligament (ACL) and Proprioception . . . 4
  - 2.1.1 Tendons, Ligaments, and the Knee Joint . . . 4
  - 2.1.2 Anatomy and Biomechanics of the ACL . . . 6
  - 2.1.3 Sensory Function and Mechanoreceptors of the ACL . . . 9
  - 2.1.4 Pathophysiology of the ACL Tear . . . 10
- 2.2 ACL Reconstruction and Rehabilitation . . . 11
  - 2.2.1 ACL Reconstruction (ACLR) Surgery . . . 11
  - 2.2.2 Graft Healing and Ligamentization . . . 13
  - 2.2.3 Clinical Outcomes and Limitations . . . 14
  - 2.2.4 Standard Post-Operative Rehabilitation . . . 16
- 2.3 Electrical Stimulation in Rehabilitation . . . 20
  - 2.3.1 Functional Electrical Stimulation (FES) . . . 21
  - 2.3.2 The Electrode-Skin Interface: Impedance and Low-Frequency Limitations 22
  - 2.3.3 Interferential Therapy (IFT) . . . 24
- 2.4 Surface Electromyography (sEMG) for Neuromuscular Feedback . . . 26
  - 2.4.1 Physiological Origins of the EMG Signal and Motor Learning . . . 27
  - 2.4.2 sEMG Acquisition and Conditioning . . . 28
- 2.5 Rationale for EMG-Controlled, IFT-Based FES . . . 32

3 Materials and Methods **33**
- 3.1 sEMG Acquisition Circuit . . . 33
- 3.2 IFT stimulation circuit . . . 35
  - 3.2.1 Bench characterization setup . . . 38
- 3.3 Software . . . 39
  - 3.3.1 Waveform Generation . . . 40
  - 3.3.2 Amplitude Calibration and Locking . . . 40
  - 3.3.3 Stimulation Streaming . . . 41
  - 3.3.4 sEMG Acquisition . . . 42
  - 3.3.5 Session Control Logic . . . 42
  - 3.3.6 Graphical User Interface . . . 43

4 Results and Discussion **44**
4.1 IFT Stimulation Circuit 44
4.2 Software 46

5 Conclusions and Future Work **53**
5.1 Conclusions 53
5.2 Future Work 54

A Appendix **56**

References **93**

List of Figures

2.1 Schematic depiction of the structure of a tendon (A) and a ligament (B) (adapted from [15]). . . . 5
2.2 Kinematic behavior and tension variation of the anteromedial (AM) and posterolateral (PL) bundles during knee flexion and extension movements (adapted from [5]). . . . 7
2.3 Microscopic structure and hierarchical organization of collagen fibers in the extracellular matrix of the ACL (adapted from [23]). . . . 8
2.4 Sagittal magnetic resonance imaging (MRI) of the knee demonstrating ACL pathology and post-reconstruction state. (A) Complete anterior cruciate ligament (ACL) tear with loss of normal ligamentous fiber continuity (red arrow). (B) Reconstructed ACL displaying a continuous, well-tensioned graft positioned within the anatomical joint axis (red arrow). Adapted from [2]. . . . 12
2.5 Clinical classification of arthrogenic muscle inhibition (AMI), ranging from normal vastus medialis function (Grade 0) to chronic, irreducible extension deficit (Grade 3). Grade 2 is associated with hamstring contracture and extension deficit. Adapted from [9]. . . . 15
2.6 Equivalent circuit obtained when using one stimulation electrode against the skin (adapted from [61]). . . . 23
2.7 Example of wave interference between two sinusoids with slightly different frequencies. . . . 25
2.8 Schematic illustration of a motor unit and how a surface EMG signal could be recorded from a muscle using a bipolar electrode (two electrode contacts) (adapted from [67]). . . . 27
2.9 Example of Motor Units [67]. . . . 28
2.10 Model circuit of the front-end amplifier with differential A) and common-mode B) input signals. Adapted from [73]. . . . 29
2.11 Model Diagram of the Equivalent Input Noise of an FE Amplifier. Adapted from [75] 31

3.1 Schematic of the sEMG acquisition PCB, with colored boxes highlighting the individual functional stages. . . . 33
3.2 Measured magnitude of the frequency response of the sEMG acquisition module, with the yellow line marking the 500 Hz upper cutoff frequency. . . . 35
3.3 Schematic of the developed stimulation circuit, with colored boxes to differentiate the main functional blocks. . . . 36

4.1 Bird's-eye view of the developed hardware modules. . . . 44

4.2 Experimental setup of the integrated system, with a numbered key that legends all its important components. 45
4.3 Primary voltage (left) and secondary current (right) versus volume setting. 46
4.4 Maximum secondary current versus volume setting, with polynomial cubic fit. . . 47
4.5 Operator-facing window. 48
4.6 Patient window in the preparation phase. 49
4.7 Session pipeline from the patient's point of view. 50
4.8 Oscilloscope trace validating the closed-loop control logic. The IFT stimulation is triggered precisely when the acquired sEMG RMS signal exceeds the predefined activation threshold (200 mV in this test case). 51
4.9 Operator window during the session, showing the acquired EMG signal, latest calculated RMS, and elapsed time. 52

List of Tables

1 Framing of the sustainable development goals in this project. iii

2.1 General composition of tendons and ligaments. 5

Abbreviations

ACL	Anterior Cruciate Ligament
ACLR	Anterior Cruciate Ligament Reconstruction
AE-ACLR	All-Inside, All-Epiphyseal Reconstruction
AM	Anteromedial
AMI	Arthrogenic Muscle Inhibition
BEAR	Bridge-Enhanced ACL Repair
CKC	Closed-Kinetic-Chain
CMRR	Common-Mode Rejection Ratio
CNS	Central Nervous System
CSA	Cross-Sectional Area
DB	Double-Bundle
DRL	Driven Right Leg
ECG	Electrocardiogram
ECM	Extracellular Matrix
EMF	Electromotive Force
EMG	Electromyography
sEMG	Surface Electromyography
ES	Electrical Stimulation
FES	Functional Electrical Stimulation
GAGs	Glycosaminoglycans
GUI	Graphical User Interface
HPF	High-Pass Filter
IFC	Interferential Current
ICS	Interferential Current Stimulation
IFT	Interferential Therapy
IKDC	International Knee Documentation Committee
JPS	Joint Position Sense
KOOS	Knee Injury and Osteoarthritis Outcome Score
LCL	Lateral Collateral Ligament
LPF	Low-Pass Filter

LSI	Limb Symmetry Indices
MCL	Medial Collateral Ligament
MMP	Matrix Metalloproteinase
MU	Motor Unit
MUAP	Motor Unit Action Potential
NMES	Neuromuscular Electrical Stimulation
OKC	Open-Kinetic-Chain
PCB	Printed Circuit Board
PCL	Posterior Cruciate Ligament
PL	Posterolateral
PLI	Power-Line Interference
PNS	Peripheral Nervous System
PSRR	Power Supply Rejection Ratio
RLD	Right Leg Drive
RMS	Root Mean Square
ROM	Range of Motion
RTS	Return to Sport
SB	Single-Bundle
SKV	Subjective Knee Evaluation Form
SDGs	Sustainable Development Goals
SNR	Signal-to-Noise Ratio
TENS	Transcutaneous Electrical Nerve Stimulation
VEGF	Vascular Endothelial Growth Factor
VMO	Vastus Medialis Obliquus

Chapter 1

Introduction

1.1 Framework and motivation

Anterior cruciate ligament (ACL) tears are among the most common serious knee injuries in sports medicine, affecting more than 120,000 individuals annually in the United States alone, predominantly in high school and college athletes. Occurring mainly through non-contact mechanisms during cutting, pivoting, and landing tasks, the incidence of ACL tears continues to rise, with female athletes facing a 2.1 to 3.4 times higher risk than men in sex-comparable sports, and high school career risks reaching 5% to 10% [1].

Anterior cruciate ligament reconstruction (ACLR) has become the standard of care for active individuals, aiming to restore the mechanical stability and to protect secondary joint structures from further degeneration [2, 3]. However, restoring the mechanical integrity of the knee does not equate to restoring its full function: complete sensorimotor recovery after ACLR is rarely achieved, even when graft fixation and postoperative healing proceed without complication [4, 3]. This gap stems from the ACL's dual role as both a mechanical restraint and a sensory organ. Its network of mechanoreceptors is irreversibly lost at the moment in cases of total rupture and is not restored by reconstruction [5, 6], resulting in persistent deafferentation that disrupts the proprioceptive feedback loop between the joint and the central nervous system [7, 8].

Clinically, this manifests as arthrogenic muscle inhibition (AMI), a reflex-mediated suppression of the excitability of the quadriceps motor neurons, which particularly affects the vastus medialis obliquus and results in a deficit in voluntary activation that is resistant to conventional strengthening exercises [9, 4]. Since this inhibition persists long after the pain and swelling have resolved, it compromises the early reactivation of the quadriceps, on which standard rehabilitation protocols depend, delaying and, in some cases, permanently limiting the restoration of muscle symmetry and dynamic joint stability [10, 9].

Electrical stimulation has long been proposed to address this issue. Neuromuscular electrical stimulation (NMES) and functional electrical stimulation (FES) can directly activate the quadriceps, bypassing arthrogenic inhibition, and have been shown to improve strength recovery beyond what exercise alone can achieve [11, 12]. However, conventional FES protocols remain open-loop:

stimulation is applied as a preprogrammed pattern, without any contribution from the patient's own neural activity, which limits its ability to activate the voluntary motor pathways that are essential for lasting neurorehabilitation [11].

A second limitation, largely independent of the first, concerns the stimulus itself. Low-frequency currents are significantly attenuated by the skin's capacitive impedance, necessitating the use of higher stimulation amplitudes that cause discomfort and superficial sensory irritation [13] to reach deeper anatomical structures. Interferential therapy (IFT) circumvents this trade-off by delivering medium-frequency carrier currents that cross the skin with minimal resistance, reconstituting a low-frequency therapeutic pulse only once inside the target tissue [13]. Despite this favorable profile, combining IFT with functionally triggered FES remains largely unexplored. Osório et al. [14] studied proportional force control based on IFT, leaving the question of whether IFT can be utilized as a closed-loop rehabilitation modality rather than a purely analgesic one.

These three observations, that AMI persistently suppresses voluntary quadriceps activation, that closing the loop with the patient's own neural drive is central to effective motor rehabilitation, and that IFT offers a more comfortable and deeper-penetrating route for delivering that stimulus, together frame the motivation for this dissertation. Surface electromyography (sEMG) offers a direct, non-invasive readout of the patient's residual voluntary effort, which can be used not only to monitor rehabilitation but also to actively trigger it: by synchronizing an IFT-based stimulus so that electrical assistance is delivered in sync with, and dependent on, the patient's own attempt to contract the quadriceps. This dissertation, therefore, addresses the design and validation of such an EMG-triggered, IFT-based FES system, aiming to couple voluntary muscular effort with a synchronized electrical stimulus in a closed feedback loop, as detailed in the following section.

1.2 Objective

The overarching purpose of this dissertation was to design and implement a functional electrical stimulation (FES) system based on interferential therapy (IFT) intended to promote the integration and regeneration of the reconstructed anterior cruciate ligament (ACL) during rehabilitation.

In this system, stimulation would be triggered by the patient's voluntary muscular effort, detected through surface electromyographic signals (sEMG) from the quadriceps muscle complex. These biopotentials would be acquired and pre-processed, then streamed to a laptop through the microphone input for digitization. A software layer would then process the incoming signal in real time, estimating its root-mean-square (RMS) amplitude to determine whether the muscular contraction exceeded a predefined activation threshold. Once this criterion was met, the software would generate the necessary waveforms for interferential current stimulation and output them through the laptop's headphone port to a purpose-built stimulation circuit. The latter would amplify and send the signal(s) to electrodes positioned around the knee, intending to deliver an interference wave capable of penetrating the joint and stimulating the neo-ligament with the low-frequency beat.

In this way, the system aimed to couple the patient's volitional activity with an electrical stimulus delivered synchronously to the healing ligament, thereby establishing a closed feedback loop between muscular effort, sensory afferents, and the neo-ligament. Ultimately, this coupling could hypothetically support neurological reintegration of the graft, contributing to its structural regeneration.

The specific aims of this dissertation were:

- Design and validate a single-supply sEMG acquisition circuit to record, filter, and amplify quadriceps muscle activity, and evaluate its performance following standard skin-preparation protocols.
- Design and validate a stimulation circuit driven via laptop audio output, incorporating an appropriate transformer and an integrated audio amplifier, and characterize its performance in terms of heat dissipation and output signal fidelity.
- Develop a cross-platform Python-based software architecture that processes digitized sEMG in real time to compute RMS and trigger stimulation upon threshold crossing, generates necessary waveforms for both true and premodulated interferential currents, and provides wave interference visualization to support parameter selection and interpretability.
- Integrate stimulation, acquisition, and software subsystems into a closed-loop IFT-based FES platform and evaluate its functionality.

1.3 Structure

This dissertation is structured into five main chapters, each providing a comprehensive literature review, presenting the developed work, and discussing the findings and their implications for the field. Chapter 2 provides an overview of the anatomy and physiology of the ACL, along with the theoretical principles of surface Electromyography (sEMG) and electrotherapeutic modalities applied to post-operative ACL rehabilitation. Chapter 3 details the methodology, focusing on the hardware development and the software implementation for real-time sEMG acquisition and stimulation control. Chapter 4 presents the experimental results and engages in a critical discussion of the findings and their implications for the field. Finally, Chapter 5 summarizes the main conclusions and outlines potential directions for future work.

Chapter 2

State of the Art

2.1 The Anterior Cruciate Ligament (ACL) and Proprioception

2.1.1 Tendons, Ligaments, and the Knee Joint

Tendons and ligaments are dense regular connective tissues [15], typically differentiated by their anatomical attachments: tendons connect muscle to bone, while ligaments connect bone to bone. However, this distinction can sometimes be ambiguous. For instance, the patellar tendon links the quadriceps muscle complex to the tibia but is referred to as a ligament because it has the patellar bone embedded within it [16]. Nonetheless, both structures carry out important roles in joints, that is, regions of the skeleton where two or more bones meet. More specifically, tendons transmit muscle-generated forces to produce joint movement, while ligaments stabilize joints by constraining motion to its normal physiological range [17].

Overall, tendons and ligaments share the same fundamental composition. Both tissues are greatly populated by fibroblasts (named tenocytes in tendons), which are flattened, spindle-shaped cells that synthesize and maintain the extracellular matrix (ECM) [15, 17]. The ECM consists of fibers (collagen and elastin) embedded in the ground substance, an amorphous and viscous network of proteins mainly associated with carbohydrates and water (proteoglycans and glycoproteins) [15]. However, the concentration and physical organization of these components differ slightly between the two tissues, as can be seen in Table 2.1, leading to distinct mechanical properties that allow them to perform their specific function [16] better.

Collagen is the predominant fibrous component, accounting for 85% [18] and 80% of the dry weight of tendons and ligaments, respectively[19]. Among all forms of collagen, type I is the most common in both tissues and is slightly more abundant in tendons. This type is characteristically flexible, strong, and resistant to force, tension, and stretch. The second most prevalent collagen in these tissues is type III, narrower than type I. Ligaments contain twice as much type III collagen as tendons. Other forms of collagen, namely II, V, VI, and XII, are also present in residual amounts [17]. Ultimately, collagen acts as the primary structural organizer for both tissues.

As mentioned previously, not only the composition but also the spatial organization of the molecular components of tendons and ligaments is directly linked to their specific mechanical

Table 2.1: General composition of tendons and ligaments.

Content/Feature	**Tendons**	**Ligaments**
Fibroblasts	20% of total volume [6]	
ECM	80% of total volume [6]	
Fibers	60-70% of ECM [6]	
Collagen (dry weight)	85% [4]	80% [5]
Type I	95-99% of total collagen [6] [5]	90% of total collagen [5]
Type III	1-5% of total collagen [6]	10% of total collagen [6]
Elastin (dry weight)	1-2% [6]	1% [6]
Ground Substance	20-30% of ECM [6]	
Proteoglycans and Glycoproteins (dry weight)	<2% [6]	1-5% [6]
Water (% of ground substance)	<95% [6]	>95% [6]

properties, allowing each structure to function effectively [16]. As illustrated in Figure 2.1, tendons have a highly parallel fiber arrangement aligned with their long axis to handle uniaxial loads (A). Ligaments often exhibit a slightly more interwoven or layered multidirectional structure to withstand forces from multiple angles (B).

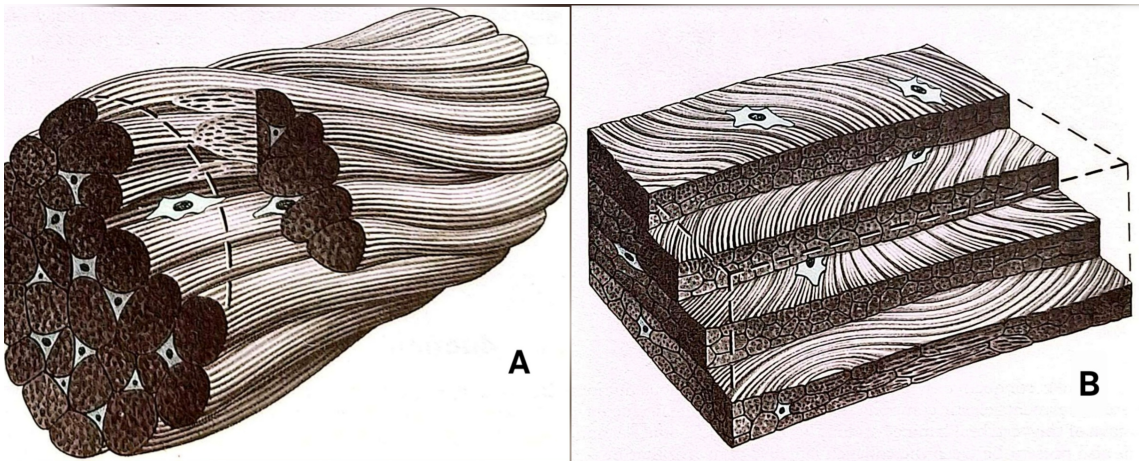


Figure 2.1: Schematic depiction of the structure of a tendon (A) and a ligament (B) (adapted from [15]).

Moreover, ligaments usually contain more ground substance when compared to tendons as a functional adaptation to the shearing movements between different fiber bundles, allowing relative movements between them and greater extensibility [16].

The human knee joint is one of the largest and most complex load-bearing articulations in the body [20]. Functionally classified as a bi-condylar type of synovial joint, it primarily facilitates flexion and extension, while also allowing for a small degree of medial and lateral rotation. This joint is formed by four bones: the femur, the tibia, the patella, and the fibula, which provide a surface for important ligamentous and muscular attachments. These bones articulate through two distinct interfaces: the tibiofemoral joint, which is the primary weight-bearing articulation, and the

patellofemoral joint, where the patella slides within the femoral intercondylar groove. To ensure smooth movement, these articulating surfaces are covered with articular cartilage, an extremely hard and smooth tissue designed to minimize friction forces [21].

A sophisticated system of static and dynamic stabilizers maintains the knee's stability. Static stability is provided by the joint capsule and a network of six primary ligaments. Among these, the medial and lateral collateral ligaments (MCL and LCL) are vital for resisting valgus and varus forces, respectively. In the center of the joint, the anterior cruciate ligament (ACL) and posterior cruciate ligament (PCL) cross each other to resist anterior and posterior shearing forces. Dynamic stability is primarily offered by the antagonistic muscle pairs, the quadriceps and hamstrings, and their associated tendons. The quadriceps act as strong extensors via the patellar tendon, while the hamstrings provide flexion and stability to the joint line [21].

2.1.2 Anatomy and Biomechanics of the ACL

The Anterior Cruciate Ligament (ACL) has long been a focus of research, with dedicated studies dating back to the Egyptian Civilization [22]. Anatomically, the ACL is a dense, band-like regular connective tissue structure situated centrally within the intercondylar region of the knee joint [5, 20, 23]. Although enclosed within its own synovial sheath, the ACL remains an intra-articular but extra-synovial structure [24], originating from a facet on the posteromedial aspect of the lateral femoral condyle, where its attachment site displays a shape comparable to a vertically disposed semicircle [24, 5, 25]. From this femoral origin, the ligament courses inferiorly, medially, and anteriorly in an arcuate trajectory toward the tibia [23, 5, 26], inserting into a wide, depressed fossa located on the anterior part of the intercondylar tibial area, anterior and lateral to the medial tibial spine. The tibial attachment footprint is wider and stronger than the femoral insertion, extending fibers beneath the transverse intermeniscal ligament and blending with the horns of the lateral meniscus [5].

One distinctive morphological feature of the ACL is its cross-sectional area (CSA), which varies considerably along its main axis, making the ligament resemble an hourglass or bow tie. [24]. The average CSA at the isthmus is around 40 mm^2 in full extension, increasing to about 44 mm^2 at 90° of flexion [24], with the overall CSA expanding progressively from the femoral origin to the distal tibial insertion (ranging from approximately 34 mm^2 proximally to 42 mm^2 distally). Overall, the ACL exhibits an irregular, fan-shaped architecture, thinner at its midsection (the isthmus) but gradually wider as it moves to the bony interfaces [20, 24, 23, 5].

This signature bow-tie has a total length between 22 mm and 41 mm (with a mean length of 27 to 38 mm), while its total width ranges from 7 mm to 12 mm [5]. This architectural layout expands the ligament's coverage area for comprehensive joint support while cleverly preventing impingement against the intercondylar fossa during full extension [23, 5].

The ACL is not a single homogeneous structure, but a double-bundle configuration first described by the Weber brothers in 1836. These functional units are universally recognized as the anteromedial (AM) and posterolateral (PL) bundles, named according to their relative tibial attachment sites [23, 24, 5].

The AM bundle originates at the most anterior and proximal aspect of the femoral attachment, inserting at the anteromedial aspect of the tibia with an average length of 34 mm, making it slightly thicker than the PL bundle [5, 24]. During knee flexion, the AM bundle becomes taut and elongates to curb excessive anterolateral tibial translation, increasing in length by roughly 12% at 90º [23, 5]. Conversely, the PL bundle originates at the postero-distal aspect of the femur and inserts at the posterolateral tibial site, with a shorter average length of 22.5 mm [5]. In turn, it is tightest during extension and becomes slack during passive flexion (shortening by approximately 32% at 90°), primarily controlling rotational stability near full extension [23, 5].

The two bundles combine into a dynamic structure that works together to allow smooth and safe movement of the knee joint. For instance, in the flexional movement, the orientation of their respective bony insertions causes the AM to spiral around the PL, dynamically shifting between load-bearing responsibilities throughout the range of motion [5], as can be seen in Figure 2.2. By combining its two distinct yet complementary functional bundles, the ACL acts as the primary passive restraint against anterior translation of the tibia relative to the femur, restraining up to 85% of total anterior displacement forces, while also preventing hyperextension and constraining internal and external tibial rotation [20, 23, 24].

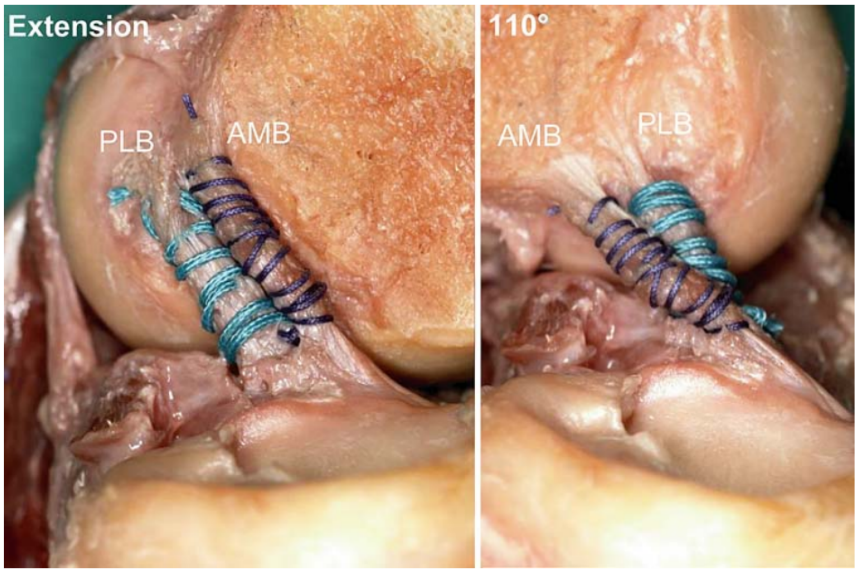


Figure 2.2: Kinematic behavior and tension variation of the anteromedial (AM) and posterolateral (PL) bundles during knee flexion and extension movements (adapted from [5]).

From a molecular point of view, under normal physiological conditions, water constitutes 60% to 80% of the total wet weight of the ACL [5], whilst collagen accounts for nearly 75% of its dry weight [24, 23]. This fibrous matrix is hierarchically organized from triple-helical polypeptide chains into microfibrils, subfibrils, fibrils, subfascicular units, and larger fascicles enclosed by the paratenon, epitenon, and endotenon, as can be seen in Figure 2.3 [23, 5].

Regarding collagen content, Type I is, expectedly, the most prevalent, representing 80% to 90% of the total collagen content of the ACL. This collagen form is arranged in dense bundles parallel to the longitudinal axis to provide primary tensile strength and resistance against high

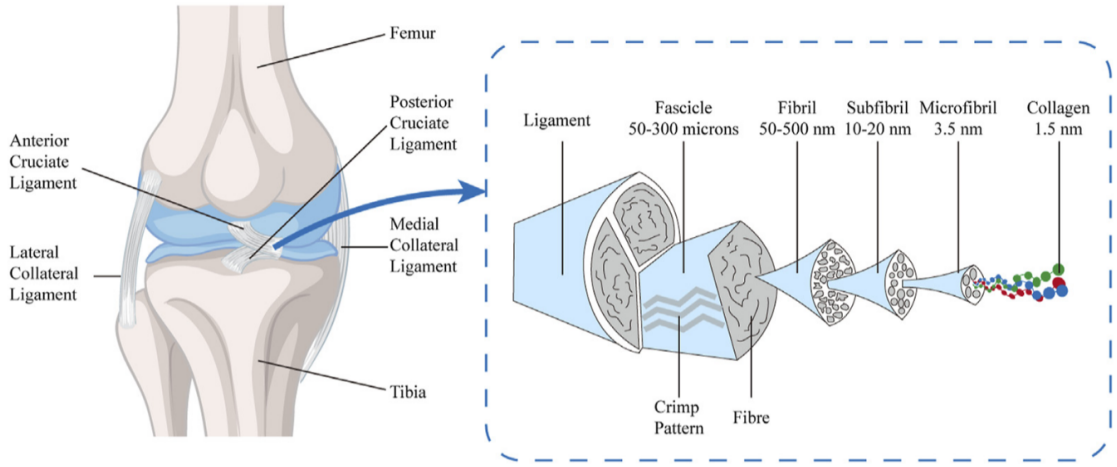


Figure 2.3: Microscopic structure and hierarchical organization of collagen fibers in the extracellular matrix of the ACL (adapted from [23]).

loads [23, 5, 24]. Type III collagen comprises roughly 10% of the ACL's total collagen content, localized mainly in loose connective tissue between Type I bundles to confer elasticity, pliability, and vascular anchoring [5, 24]. Type II collagen, although absent in standard ligaments, is specifically localized in the fibrocartilaginous attachment zones, signifying a regional adaptation to compressive and shearing forces [5]. Types IV and VI collagen serve functions in vascular basement membranes and as functional gliding mechanisms between fibrillar units, respectively [23, 5].

The remaining portion of dry weight consists of proteoglycans, glycosaminoglycans (GAGs), elastin (approximately 1%), and specialized cells [23, 24]. Elastin allows quick elastic recoil post-loading, while the high concentration of negatively charged GAGs attracts water molecules to provide crucial shock-absorbing and viscoelastic capabilities [23, 5].

After considering the histological specificities of the ACL, one can divide it into three distinct zonal domains [23, 5, 24]:

- a proximal zone, rich in type II collagen and highly populated with cells;
- a middle fusiform zone, characterized by high collagen density, low cellularity, and elongated fibroblasts, which are aligned with a characteristic sinusoidal crimp pattern (amplitude and periodicity of 45 to 60 µm), as displayed in Figure 2.3. This section acts as a mechanical buffer;
- a distal zone, rich in chondroblasts and fibrocartilage, forming a mineralized transitional junction near subchondral bone to withstand shearing stress.

Due to its anisotropic nature and non-linear stress-strain behavior, the ACL exhibits low initial stiffness at low loads as collagen crimp straightens out, displaying progressively higher stiffness as parallel collagen fibers are recruited under higher forces [23, 5]. Because spontaneous self-healing of a completely ruptured ACL is virtually non-existent [20], replicating this complex,

multi-bundle, and stratified anisotropic structure remains a major technical challenge in surgical reconstruction [20, 23, 5].

2.1.3 Sensory Function and Mechanoreceptors of the ACL

The anterior cruciate ligament functions not merely as a passive mechanical constraint, but as an active sensory organ, fully integrated with both the peripheral (PNS) and central (CNS) nervous systems. The innervation of the ACL originates from the posterior articular branches of the tibial nerve [5, 26, 6], whose nerve fibers penetrate the posterior capsule of the knee joint, running through the ligament alongside the synovial and periligamentous vessels until they reach the infrapatellar fat pad [5, 26]. Most of this neural supply consists of vasomotor autonomic fibers associated with the intraligamentous vasculature, regulating its blood flow. However, independent somatosensory afferent fibers also run among the fascicles; these can be larger and myelinated (2–10 µm in diameter) or smaller and unmyelinated (1 µm in diameter).

In particular, myelin-coated fibers are sensitive to stretch or tension and are therefore referred to as mechanoreceptors. These specialized structures are predominantly concentrated in the peripheral subsynovial layer and in the vicinity of the femoral and tibial bone insertions [5]. They are classified into the following types:

- Ruffini Receptors: Slow-adapting receptors sensitive to tensile forces and stretching. Located on the ligament surface, they are more densely packed in the femoral portion, where mechanical deformation is greater, and play a crucial role in static joint position sense (JPS). [5, 26].

- Vater-Pacini Receptors: Rapidly adapting receptors responsive to acceleration and sudden movements. Situated at the proximal (femoral) and distal (tibial) ends of the ACL, they act primarily in kinesthetic control (proprioception) [5, 26].

- Golgi-like Tension Receptors: Slow-adapting receptors sensitive to tensile forces, distributed near the femoral and tibial bone insertions and beneath the synovial membrane [5, 26].

As for the unmyelinated fibers, these are nociceptors, that is, free nerve endings that signal pain. They also perform a local efferent function by releasing neuropeptides with vasoactive effects, which modulate tissue homeostasis and the late remodeling of grafts. This topic will be discussed in a later chapter [5, 26].

The sensory information detected by these mechanoreceptors forms the afferent arm of a complex neural loop integrated with the spinal cord, lower brain centers, and the cerebral cortex [5, 6, 27]. Mechanical deformations within the ligament activate group II and III nerve fibers, which in turn influence the output of muscle spindles through the fusimotor system [5]. This interaction triggers reflex motor responses in the surrounding periarticular musculature, a neurophysiological mechanism widely described as the "ACL reflex" [5, 28].

Although historically hypothesized to act as an automatic protective shield against joint overload, electrophysiological timing demonstrates that this excitatory reflex cannot serve as an immediate automatic protective mechanism against acute tear events. With a reflex latency of approximately 70 ms, at least 110 ms (accounting for reflex time and electromechanical delay) must pass before the surrounding muscles can produce substantial forces following a load applied to the ACL. In contrast, mechanical injury occurs much faster [5]. Consequently, the primary functional role of the ACL reflex is not immediate structural protection, but rather the continuous updating of muscle programs necessary for normal joint movement, directly influencing the maximal voluntary contraction exertion of the *quadriceps femoris* and ensuring coordinated dynamic muscle stabilization [5, 6, 28].

2.1.4 Pathophysiology of the ACL Tear

An ACL tear happens when this ligament is partially or totally ruptured under multiplanar loading. In sports, more than half of the ACL injuries are non-contact [29, 30, 31], occurring during the execution of dynamic movements like cutting, pivoting, sudden deceleration, or landing from a jump. However, ACL failure is not limited to physical activity; it can occur whenever excessive anterior tibial translation, internal tibial rotation, and valgus collapse occur simultaneously, straining the ligament beyond its structural threshold. These events frequently take place within 17 to 60 milliseconds following initial foot-ground contact, a too narrow a window for the aforementioned ligament-muscle protective reflexes to engage rapidly enough to counter these excessive forces [7].

An acute ACL rupture initiates a multifaceted cascade involving biomechanical disturbances, an acute inflammatory response, and complex sensorimotor impairment [3, 7]. From a biomechanical standpoint, ACL loss alters normal knee kinematics. It displaces the joint's center of rotation, impairing coupled rotation during flexion and increasing functional stress on secondary soft tissue support structures, such as the menisci and joint capsule [24, 3].

Beyond mechanical instability, a complete ACL tear causes profound deafferentation due to the loss of mechanoreceptors [5, 6, 28, 27]. This abrupt interruption of sensory feedback deprives the central nervous system (CNS) of the proprioceptive information required to monitor joint position and loading accurately, creating a severe sensorimotor system error characterized by uncoordinated and delayed muscular activation [7].

Simultaneously, the acute trauma triggers hemarthrosis and a rapid influx of inflammatory cytokines, including IL-1β and IL-6. These chemical mediators sensitize Group III (Aδ) and IV (C) afferent nerve fibers. This peripheral nociceptive sensitization promotes the release of substance P and glutamate into the dorsal horn of the spinal cord, inducing central sensitization and altering the balance between excitatory and inhibitory interneurons [7].

This altered spinal circuit directs a persistent inhibitory impulse to alpha motor neurons, culminating in arthrogenic muscle inhibition (AMI). AMI selectively prevents the voluntary recruitment of the quadriceps muscles, particularly the *vastus medialis obliquus*. Compounding this neurological blockade, post-injury catabolic conditions trigger the upregulation of myostatin, a negative

regulator of muscle growth, which inhibits muscle protein synthesis and accelerates rapid quadriceps atrophy [7].

Quadriceps weakening acts as an involuntary systemic compensatory mechanism to reduce anterior tibial traction on the impaired knee, further destabilizing joint kinetics and predisposing the joint environment to long-term degeneration [3, 7]. Consequently, functional instability is largely driven by this lack of coordinated muscle stabilization secondary to diminished sensory input [6, 28]. To compensate for the loss of ACL proprioception, the CNS undergoes cortical reorganization. This places greater demands on brain motor planning centers and leads to an over-reliance on the visual system for postural control and movement execution, a neuroplastic shift that may ultimately increase the risk of reinjury [6, 27].

2.2 ACL Reconstruction and Rehabilitation

2.2.1 ACL Reconstruction (ACLR) Surgery

Surgical reconstruction remains the current standard of care for the treatment of complete anterior cruciate ligament (ACL) tears, particularly in active individuals and competitive athletes who participate in sports involving cutting and rotational movements, as they face a high risk of secondary meniscal and chondral injuries without an intact ligament [2]. While historical primary repair and contemporary biological reinforcements, such as the bridge-enhanced ACL repair (BEAR) procedure, have attracted renewed interest, traditional ACL repair techniques without reinforcement have historically shown graft failure rates up to five to ten times higher than those of reconstruction, leading to a consensus in favor of formal structural reconstruction, as shown in Figure 2.4 [2, 32].

The timing of the procedure is critical; reconstruction is typically scheduled three to six weeks post-injury to allow acute joint inflammation and hemarthrosis to subside, thereby minimizing the risk of postoperative arthrofibrosis, while delaying surgery beyond three to six months significantly increases the incidence of secondary articular degeneration and graft failure. In pediatric and adolescent populations with open growth plates, specialized techniques such as All-Inside, All-Epiphyseal Reconstruction (AE-ACLR), or physeal-sparing iliotibial band procedures are used to restore stability without disrupting skeletal growth [2].

The primary objective of anatomic ACL reconstruction is to restore native ligament dimensions, collagen orientation, and footprint kinematics to re-establish physiological knee biomechanics [24]. Achieving this goal requires strict adherence to anatomical principles, including accurate alignment of bone tunnels within native insertion sites, individualized graft sizing, and optimal bundle tensioning [24]. Recent anatomical insights highlight that femoral footprint geometry varies; for instance, a "C-shaped" tibial insertion site may be best restored via a double-bundle (DB) approach, whereas an "elliptical" footprint is effectively addressed using a single-bundle (SB) technique [24].

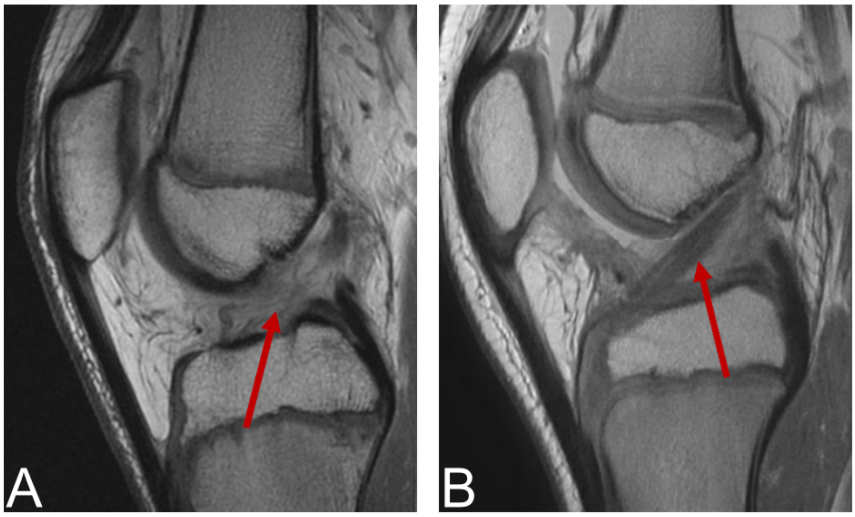


Figure 2.4: Sagittal magnetic resonance imaging (MRI) of the knee demonstrating ACL pathology and post-reconstruction state. (A) Complete anterior cruciate ligament (ACL) tear with loss of normal ligamentous fiber continuity (red arrow). (B) Reconstructed ACL displaying a continuous, well-tensioned graft positioned within the anatomical joint axis (red arrow). Adapted from [2].

Historically, the transtibial drilling technique frequently restricted the orientation of the femoral tunnel, placing grafts too far anteriorly and vertically, which failed to restore rotational stability and increased the risk of notch impingement and graft failure [33]. Independent tunnel drilling techniques (e.g., via an anteromedial portal) allow surgeons to place the femoral socket within 2 mm of the true geometric center of the ACL footprint, typically located at 43% of the proximal-to-distal length of the lateral intercondylar notch wall and approximately 2.5 mm anterior to the posterior articular margin [33]. While anatomical DB reconstruction theoretically provides superior dynamic rotational control by replicating both the anteromedial and posterolateral bundles, long-term clinical data spanning over five years demonstrate equivalent mid-to-long-term outcomes, graft survival, and protection against osteoarthritis progression when compared to an anatomically placed SB reconstruction [24, 33].

The choice of procedure and graft is tailored to the individual patient, taking into account their characteristics, activity requirements, and surgeon's experience [2, 34]. Arthroscopic reconstruction on an outpatient basis is usually performed using autografts harvested from the quadriceps tendon, bone-patellar tendon-bone, or multistrand hamstring tendons, the latter of which has become a widespread standard due to lower donor-site morbidity, reduced postoperative pain, and faster early recovery [2, 34]. Conversely, allografts are generally avoided in young patients due to significantly higher failure rates [2]. Following graft fixation with interference screws or suspensory devices, supervised multi-phase rehabilitation begins within days [35, 36]. Recovery prioritizes early quadriceps reactivation and range of motion restoration, progressing to running in a straight line at 3 to 5 months, rotational exercises at 5 to 8 months, and a full return to high-demand competitive sports between 9 to 12 months, following objective tests of function, strength,

and sensorimotor testing [36].

2.2.2 Graft Healing and Ligamentization

The biological healing process following anterior cruciate ligament (ACL) reconstruction using tendon grafts is a complex, multi-phase cascade essential for restoration of joint function [37]. Graft healing primarily involves two distinct phenomena: the incorporation of the graft into the surrounding bone within the tunnel and the process of intra-articular remodeling, commonly referred to as "ligamentization" [38, 39]. Broadly classified into three continuous stages: the early healing phase, the proliferation phase, and the maturation/ligamentization phase. These biological events determine the final structural and mechanical properties of the reconstructed knee [38, 39, 40].

The early graft healing phase extends from implantation until approximately the fourth postoperative week [38, 40]. During this initial stage, the graft undergoes avascular necrosis, predominantly in its central portion, accompanied by marked hypocellularity and a complete absence of revascularization [38, 39, 40]. Cell necrosis triggers the local release of pro-inflammatory cytokines such as TNF-α, IL-1β, and IL-6, along with various chemokines [38, 40, 37]. This molecular signaling cascade recruits host inflammatory cells, including neutrophils and macrophages, to clear necrotic debris and initiate cell migration and extracellular matrix (ECM) synthesis [38, 40, 37]. Original graft cells do not remain viable and are completely replaced by extrinsic cells derived from the synovial fluid, remnants of the native ACL stump, or bone marrow elements within 2 to 4 weeks [38]. Although collagen degradation begins around 3 weeks, the overall collagen structure and wave-like pattern are temporarily preserved, preventing an immediate collapse in the graft's mechanical strength during the first month [38].

The proliferation phase, which occurs between the fourth and twelfth postoperative weeks, is characterized by a peak in cellular and matrix remodeling activity, which paradoxically coincides with the graft's lowest mechanical strength [38, 40, 37]. Hypoxia during the early necrotic phase positively regulates vascular endothelial growth factor (VEGF) and hypoxia-inducible factor 1-alpha (HIF-1α), driving intense angiogenesis from the periphery toward the graft core [38, 40, 37]. Infiltrating mesenchymal stem cells and activated fibroblasts release essential growth factors (such as bFGF, TGF-β1, and PDGF isoforms) that regulate collagen production and ECM organization [38, 40, 37]. Additionally, contractile myofibroblasts are expressed to exert isometric tension, helping to restore the in situ tension required for complete maturation [38]. As intense revascularization and cell turnover destabilize the matrix, mechanical failure during this phase shifts from graft expulsion at the bone tunnel interface (typical of the 2-to-4-week mark) to intra-articular ruptures in the substantia media, around 6 to 8 weeks after surgery [38].

The maturation or ligamentization phase follows the proliferation phase and extends from several months to years post-surgery [38, 40, 37]. During this long-term remodeling, cell density and vascularization gradually decrease to levels comparable to those of a native ACL, while fibroblasts assume a mature, fusiform morphology [38]. Mechanotransduction via integrin-mediated pathways (such as FAK, Wnt/β-catenin, and ERK) promotes the realignment of collagen along

lines of mechanical tension, balancing degradation by matrix metalloproteinases (MMPs) with the synthesis of type I collagen [37]. Despite these structural adaptations, animal models demonstrate that the original bimodal collagen fibril distribution never fully recovers, permanently shifting to a unimodal pattern of small-diameter fibrils, with mechanical properties rarely surpassing 50% to 60% of the intact ACL [38, 40].

Studies of human biopsies show notable differences in the timing and severity of these biological processes compared with animal models [38, 40]. Human ACL grafts exhibit significantly less necrosis (rarely exceeding 30%), retain viable intrinsic cells, and undergo less extensive overall remodeling [38, 40]. Consequently, the proliferation phase in humans is often delayed, extending from 3 to 10 months after surgery, with complete histological maturation and a "ligamentous" appearance only being achieved between 1 and 3 years after reconstruction [38, 40]. At the same time, the incorporation of the graft into the tunnel varies according to the type of graft [40]. Bone-to-bone union in bone block grafts resembles the union of complex fractures and is completed around 16 weeks. In contrast, soft tissue (tendon-to-bone) integration occurs via a fibrovascular interface, the formation of Sharpey's fibers, and progressive mineralization over a period of 6 to 12 months [40]. Proper postoperative management and mechanical stimuli remain crucial throughout these stages [38, 37]. While total tension deprivation induces rapid collagen bundle defragmentation and a loss of tensile strength, excessive overloading risks premature graft stretch-out and mechanical failure [38, 37]. Controlled mechanical loading activates beneficial mechanotransduction pathways, promoting the structural adaptation and tissue integration required for successful clinical outcomes [38, 37].

2.2.3 Clinical Outcomes and Limitations

Anterior cruciate ligament reconstruction (ACLR) is performed to restore the mechanical stability and kinematics of the joint, but complete functional and sensory recovery is rarely fully achieved [4, 3]. One of the main post-surgical challenges stems from the loss of native mechanoreceptors in the torn ligament [4, 41], alongside acute changes in brain plasticity following an ACL tear, which impair both cognitive and motor functions [9]. Rupture of the ACL's neural arc leads to persistent proprioceptive deficits that impair dynamic balance, motor control, and normal gait patterns [4, 8]. Because proprioceptive recovery is associated with improved patient satisfaction and self-reported functional outcomes, sometimes more strongly than passive mechanical stability alone, an incomplete sensory restoration may prevent patients from returning to their pre-injury level of performance [8].

At the muscular level, the most prevalent limitation, and one that is resistant to conventional rehabilitation, is persistent quadriceps atrophy, specifically affecting the *vastus medialis obliquus* (VMO) [25], and a deficit in voluntary activation [40, 4]. This phenomenon is primarily driven by arthrogenic muscle inhibition (AMI), a pain-induced presynaptic and neurological reflex response caused by pain, joint effusion, and alterations in joint receptors. AMI reduces motor neuron excitability, preventing effective VMO contraction and heel lift, while often causing compensatory activity in the rectus femoris and tibialis anterior. Crucially, this arthrogenic motor inhibition can

be accompanied by hamstring reflex hypertonia, inducing analgesic flexion contracture, reduced range of motion, and severe biomechanical impairment [9].

To detect these impairments preoperatively and identify patients at risk of postoperative knee stiffness, clinical classification systems divide AMI into distinct grades based on severity and reducibility. Grade 0 reflects normal VMO function. Grade 1 denotes isolated VMO motor inhibition, further stratified into Grade 1A (easily reversible via immediate bedside exercises) and Grade 1B (requiring extended, targeted rehabilitation). Grade 2 encompasses VMO inhibition combined with active extension deficits caused by reflex hamstring contracture, similarly divided into Grade 2A (reversible in consultation) and Grade 2B (recalcitrant to quick intervention). Finally, Grade 3 represents chronic and irreducible passive extension deficits resulting from structural posterior capsule retraction, where physiotherapy is insufficient and extensive posterior arthrolysis becomes necessary [9].

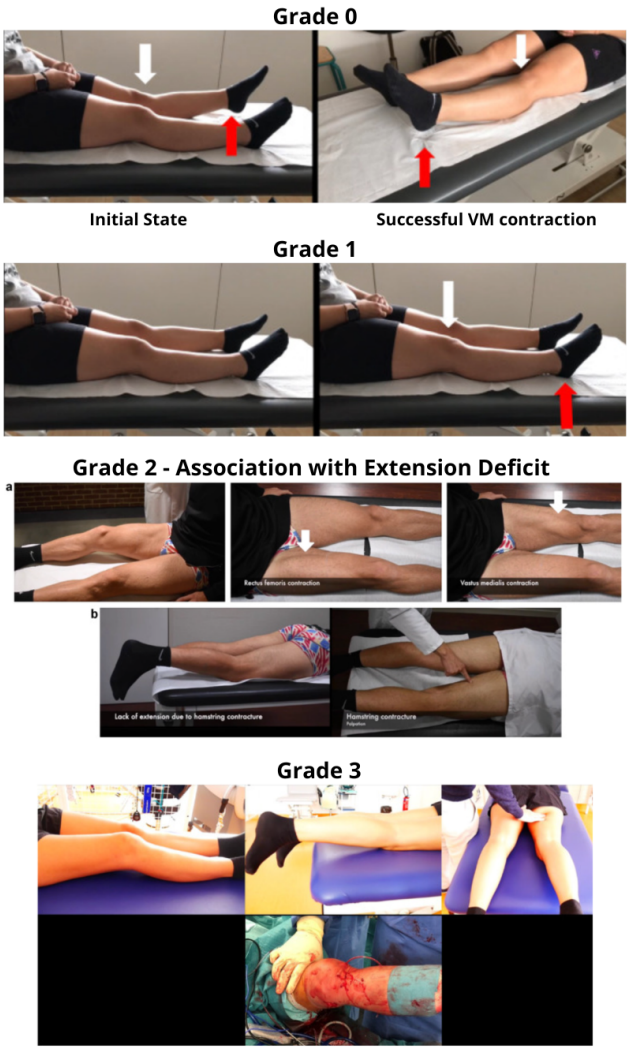


Figure 2.5: Clinical classification of arthrogenic muscle inhibition (AMI), ranging from normal vastus medialis function (Grade 0) to chronic, irreducible extension deficit (Grade 3). Grade 2 is associated with hamstring contracture and extension deficit. Adapted from [9].

Recent prospective data have shown that AMI is highly prevalent in acute ACL tears (<6 weeks), affecting over 55% of patients and significantly impairing preoperative clinical outcome scores (Lysholm, IKDC, SKV, KOOS). Although approximately 80% of these cases are easily reversible (Grades 1A/2A), nearly 12% present with resistant forms (Grades 1B/2B) that require specific preoperative intervention. Failing to identify and treat this reflex motor inhibition preoperatively dramatically increases the risk of severe short- and long-term complications. Untreated AMI leads to postoperative knee stiffness, reported in up to 15% to 23% of acute knee procedures, persistent flexion deficits, high re-injury rates, and accelerated early-onset osteoarthritis, driven by altered joint loading and loss of neuromuscular shock absorption [9].

Furthermore, site-specific morbidities associated with graft selection, such as persistent anterior knee pain associated with bone-patellar tendon-bone autografts, or flexion deficits and elevated infection risks linked to hamstring autografts, exacerbate these clinical limitations. Ultimately, surgical stabilization alone is insufficient without a comprehensive preoperative evaluation of AMI and an integrated sensorimotor rehabilitation strategy.

2.2.4 Standard Post-Operative Rehabilitation

Standard post-operative rehabilitation following anterior cruciate ligament reconstruction (ACLR) has traditionally been structured according to sequential phases, with progression largely guided by the expected biological healing process and the resolution of clinical impairments. Although contemporary approaches increasingly emphasize criterion-based progression rather than exclusively time-based milestones, traditional protocols generally follow a progressive sequence of pain and effusion control, restoration of range of motion (ROM), quadriceps activation and strengthening, proprioceptive and neuromuscular training, running and agility, and, ultimately, return to sport (RTS) [32, 10, 42, 43, 44]. The precise timing and content of each phase vary by graft type, concomitant procedures, surgical technique, and patient presentation; nevertheless, the overall objective remains restoring knee function while protecting the healing graft [32, 42].

Phase 1: Pain, inflammation, and early protection

The initial post-operative phase focuses on controlling pain and joint effusion while protecting the reconstructed ligament and facilitating early functional recovery. Joint effusion can contribute to pain, restricted ROM, and arthrogenic quadriceps inhibition, thereby interfering with subsequent rehabilitation [10, 42]. Consequently, early management commonly includes cryotherapy, compression, elevation, and active muscle-pump exercises involving the quadriceps and calf musculature [42, 43]. Although postoperative bracing has traditionally been used to protect the knee and restrict potentially harmful movements, its routine use remains controversial. The evidence summarized in the literature indicates that bracing for postoperative pain control, ROM, knee stability, or prevention of re-injury is not more beneficial than standard rehabilitation [44, 45]. Contemporary protocols tend to individualize brace use based on surgical considerations, symptoms, and patient confidence rather than treating it as an obligatory component of rehabilitation.

Weight-bearing is also progressively introduced during this early phase. Historically, rehabilitation protocols often restricted weight-bearing for several weeks; however, accelerated protocols

increasingly permit immediate or early weight-bearing according to pain, effusion, ROM, and quadriceps control [44, 45]. Early weight-bearing has not been consistently associated with increased knee laxity and may help restore a normal gait pattern [45]. However, protocols following ACL repair may require greater protection because the repaired tissue has different biological and mechanical characteristics from a reconstructed graft [32, 42].

Phase 2: Restoration of range of motion

Recovering full knee ROM, particularly extension, is a fundamental objective of early rehabilitation. Loss of knee extension can alter tibiofemoral and patellofemoral joint mechanics, increase abnormal articular contact pressures, and contribute to quadriceps inhibition [10, 43]. Small extension deficits of approximately 3-5° have been associated with poorer subjective and objective outcomes, while early restoration of full extension may reduce the risk of arthrofibrosis [10]. Therefore, traditional protocols prioritize achieving complete passive extension as early as possible, frequently within the first 1 to 2 weeks after surgery. Low-load, prolonged stretching, calf stretching, patellofemoral mobilization, and positioning techniques such as prone or bag hangs may be employed when extension remains limited [10, 42].

Flexion is also progressively restored, although the emphasis is generally placed first on achieving full extension. Early protocols commonly target approximately 90° of ROM during the first postoperative week and more than 110-120° of flexion during the following weeks [10, 42]. Active-assisted flexion, wall slides, stationary cycling, and other controlled ROM exercises can be introduced progressively according to symptoms and clinical milestones [10, 43]. The literature suggests that early ROM does not adversely affect graft stability and may prevent the deleterious consequences associated with immobilization [44]. Thus, despite historical variations in the timing and extent of ROM restrictions, early restoration of symmetrical ROM has become a central component of standard ACL rehabilitation [43, 45].

Phase 3: Progressive quadriceps strengthening

Restoring quadriceps strength and activation is another major priority after ACLR. Quadriceps weakness is particularly pronounced during the first months after surgery and may persist for many months or even years [10]. This deficit is partly explained by arthrogenic muscle inhibition associated with joint effusion and altered afferent input from the injured knee, which reduces the patient's ability to voluntarily activate the quadriceps [10, 43]. Consequently, quadriceps activation exercises are initiated immediately or very early after surgery, commonly through isometric quadriceps sets and straight-leg raises, with progression according to the patient's ability to perform these exercises without an extension lag [10, 42]. Neuromuscular electrical stimulation (NMES) may be used as an adjunct when voluntary quadriceps activation is insufficient. Several studies have reported improved quadriceps strength and functional outcomes when electrical stimulation is combined with voluntary exercise. However, the quality and consistency of the available evidence mean that NMES should be considered an adjunct rather than an essential component of rehabilitation [10, 42]. Its primary rationale is to facilitate motor-unit recruitment during a period in which arthrogenic inhibition limits voluntary activation.

As quadriceps control improves, strengthening progresses from low-load and isometric exercises to isotonic and resistance-based exercises. Closed-kinetic-chain (CKC) exercises, including squats, step-ups, leg press, and related functional movements, have traditionally been incorporated early because they allow progressive loading while providing functional and neuromuscular benefits [10, 42]. Open-kinetic-chain (OKC) exercises have historically been introduced more cautiously because of concerns regarding anterior tibial translation and ACL strain. Nevertheless, subsequent evidence suggests that appropriately dosed OKC exercises performed within controlled ROM can be incorporated without necessarily compromising graft stability, while potentially providing additional quadriceps strength gains [10, 42]. Thus, contemporary protocols generally combine CKC and OKC exercises according to the stage of rehabilitation, graft characteristics, ROM, symptoms, and strength deficits.

Traditional phase-based rehabilitation progressively shifts from muscular activation and endurance towards the development of maximal strength and power. One criterion-based framework describes an initial protection and muscle-initiation phase during the first 6 weeks, followed by muscular endurance development from approximately weeks 7 to 14, strength development from weeks 15 to 21, and a subsequent power, speed, and agility phase from approximately week 22 onwards [43]. Although these time frames should not be interpreted as rigid requirements, they illustrate the traditional progression from low-load exercise and movement restoration towards increasingly demanding functional loading.

Phase 4: Proprioceptive and neuromuscular training

Following ACL injury and reconstruction, alterations in afferent input and sensorimotor control may affect joint position sense, balance, muscle activation, and dynamic knee stability. Consequently, proprioceptive and neuromuscular exercises are progressively incorporated into rehabilitation. Early interventions may include bilateral and unilateral balance activities, unstable surfaces, weight-shifting exercises, and controlled single-leg tasks, progressing towards perturbation exercises and more dynamic multiplanar movements [10, 43]. The objective is not only to restore proprioception but also to improve the patient's ability to stabilize the knee during functional activities dynamically. The literature indicates that balance and neuromuscular re-education can improve self-reported knee function without adversely affecting knee laxity or strength [10]. However, evidence on the precise contribution of proprioceptive training remains heterogeneous, partly because strengthening and proprioceptive exercises often overlap in their physiological demands [43]. This is particularly relevant because dynamic sporting activities require integrating strength, sensory information, postural control, and rapid neuromuscular responses rather than isolated proprioceptive function. Accordingly, later-stage rehabilitation increasingly incorporates single-leg movements, perturbations, landing exercises, plyometrics, deceleration, and changes of direction to reproduce the demands of sport [10, 43].

Phase 5: Running, agility, and progression towards sport-specific activity

Once adequate ROM, quadriceps strength, and neuromuscular control are achieved, rehabilitation progresses toward higher-level functional activities. Running is traditionally introduced several months after ACLR, although the precise timing differs substantially between protocols.

Some conventional protocols permit running at approximately 2 to 3 months, whereas others recommend a later progression, with running commonly initiated around 3 to 5 months following surgery [32, 43, 44] . Importantly, contemporary criterion-based approaches emphasize that running should not be initiated solely because a specific postoperative time point has been reached. Rather, the patient should demonstrate adequate strength, minimal or absent effusion, full ROM, pain-free loading, and sufficient dynamic control [10, 42].

Running progression may initially involve reduced-load environments, such as an anti-gravity treadmill or aquatic running, before progressing towards full weight-bearing running [42, 43]. The load and speed are then increased progressively while monitoring symptoms and knee response. Persistent or increasing pain and effusion are generally interpreted as signs that the current training load exceeds the patient's tolerance and should therefore result in modification or temporary regression of the programme [10, 42].

Agility, plyometric, and sport-specific exercises are then introduced to prepare the athlete for the multidirectional, high-rate loading demands of sport. Agility training develops acceleration, deceleration, and changes of direction, whereas plyometric exercises target rapid force production and neuromuscular control during jumping and landing [10, 43]. These exercises progress from simple bilateral movements to unilateral and multidirectional tasks, with increasing speed, intensity, and sport specificity [42]. Particular attention is given to landing mechanics and dynamic lower-limb alignment because appropriate neuromuscular control may help reduce potentially injurious movement patterns [10].

Phase 6: Return-to-Sport

Traditionally, RTS following ACLR has been strongly influenced by time since surgery, with some protocols historically permitting return to contact or pivoting sports at approximately 6 months and accelerated protocols reporting return to sports between 4 to 6 months [43, 44]. However, time alone is increasingly considered insufficient to determine readiness for unrestricted sport participation. Significant quadriceps strength, balance, functional performance, and force-attenuation deficits may still be present at 6 months following surgery [44]. Therefore, a purely time-based approach may expose athletes to substantial residual impairments despite apparently adequate postoperative healing.

Standard RTS assessment consequently incorporates a combination of objective and subjective criteria. Common measures include quadriceps strength testing, limb symmetry indices (LSI), hop tests, balance assessments, knee stability, ROM, pain and effusion status, and patient-reported outcomes [10, 44]. A limb symmetry threshold of approximately 90% has frequently been used for quadriceps strength and functional performance. However, this criterion should be interpreted alongside other measures rather than considered sufficient on its own [10, 42]. In addition, the athlete should demonstrate full or symmetrical ROM, absence of significant pain or effusion, adequate dynamic knee stability, and appropriate psychological readiness [42, 44].

Functional testing commonly includes the single hop for distance, triple hop, crossover hop, and timed hop tests, which provide information regarding unilateral lower-limb function and dynamic stability [10]. More comprehensive criterion-based approaches may additionally incor-

porate the Y-Balance Test, single-leg squat, single-leg drop or landing assessments, hamstring-to-quadriceps strength ratios, and sport-specific agility tests [42]. Using several complementary measures is particularly relevant because no single test captures all the physical and psychological demands associated with successful RTS [10].

Importantly, clearance for sport should not be interpreted as an immediate transition from rehabilitation to unrestricted competition. Once RTS criteria are met, athletes are generally expected to progress through lower-level practice activities before gradually increasing exposure to full training and competition, while monitoring pain, effusion, ROM, strength, and confidence [10]. This gradual progression reflects the principle that successful RTS requires not only restoring isolated physical capacities but also tolerating repeated, sport-specific loads under increasingly demanding conditions.

Overall, traditional post-operative ACL rehabilitation can therefore be understood as a sequential progression from symptom and effusion control, restoration of ROM, quadriceps activation and strengthening, proprioceptive and neuromuscular re-education, progressive running and agility, and finally sport-specific preparation and RTS. Although these phases remain clinically useful for organizing rehabilitation, the literature increasingly supports combining them with objective progression criteria rather than relying exclusively on postoperative time [32, 10, 42, 44, 45]. This distinction is particularly important for ACL repair, for which postoperative rehabilitation protocols have historically been extrapolated from ACLR because of the limited evidence specific to repair procedures [32, 42]. Consequently, rehabilitation following ACL repair should consider both the biological healing requirements of the repaired tissue and objective functional milestones, with progression individualized according to symptoms, strength, ROM, neuromuscular control, and sport-specific demands.

2.3 Electrical Stimulation in Rehabilitation

Electrical stimulation (ES) is a broad term for the application of electrical currents to the human body, producing changes in neural activity [46]. Historians have traced the usage of ES for medicinal purposes back to the Ancient Civilizations of Egypt, Rome, and Greece. In this period of history, fish with electrical properties were prescribed for pain relief in conditions such as gout, arthritis, and headaches [47, 48].

Through time, several different ES modalities have been tested and brought to clinical practice. Interestingly, the naming of these distinct techniques is usually unconsensual, since healthcare providers and researchers often disagree on the terminology used, inadvertently spreading incorrect and misleading jargon [49]. This “name game”, a term coined by Gad Alon in a conference lecture in 2015 [50], is believed to be one of the reasons why ES has yet to reach its full therapeutic, functional, and rehabilitative potential [49]. The most widely used types of electrical stimulation in medicine are Transcutaneous Electrical Nerve Stimulation (TENS), Neuromuscular

Electrical Stimulation (NMES), and Functional Electrical Stimulation (FES). Interferential Therapy (IFT) is another common choice among practitioners [46] and will be discussed later in a dedicated chapter.

Transcutaneous Electrical Nerve Stimulation (TENS) uses electrodes placed on the skin surface to stimulate peripheral nerves with low-voltage electrical currents, with the goal of modulating pain perception. It does so by inhibiting the transmission of afferent pain nerve impulses and/or stimulating the release of endorphins through activation of the endogenous opioid system [51, 52]. Therefore, TENS is not curative [49], but rather an efficacious pain-relieving technique when applied during or after treatment [53].

Another technique that uses transcutaneous electrical currents is Neuromuscular Electrical Stimulation (NMES). However, NMES applies these stimuli to a group of muscles, causing them to contract in a way that mimics the body's nervous system during voluntary muscle activation [12, 54]. NMES is frequently used in rehabilitation to address muscle weakness, including quadriceps weakness following anterior cruciate ligament reconstruction (ACLR). Evidence reported in the provided literature suggests that combining NMES with exercise can improve quadriceps strength more effectively than exercise alone, particularly during the early postoperative period [16]. Electrical stimulation has also been associated with reduced quadriceps atrophy and improved muscle strength. However, improvements in other outcomes, such as range of motion, laxity, subjective function, and return-to-sport duration, have not been consistently demonstrated [18].

2.3.1 Functional Electrical Stimulation (FES)

Functional Electrical Stimulation (FES) is a related but functionally distinct application of electrical stimulation. While NMES primarily aims to induce muscle contractions, FES applies electrical stimulation in coordination with a specific functional task or movement. Thus, when stimulation is employed during the performance of specific tasks or daily functions, the term FES is considered more appropriate than NMES [11]. FES specifically targets weakened or paralyzed muscles to generate functional movement [55]. In this context, FES can reproduce or assist movements such as walking, grasping, pedaling, transferring, and standing in individuals with impaired voluntary motor function [56].

FES involves artificially inducing electrical currents in specific motor neurons to generate muscle contractions. The stimulation is delivered through electrodes, which can be transcutaneous, percutaneous, epimysial, or positioned around the nerve innervating the target muscle. The resulting muscle tension depends primarily on stimulation intensity and frequency. Stimulation intensity is related to the total electrical charge transferred to the muscle, which is influenced by parameters such as pulse amplitude, pulse duration, frequency, and the shape of the pulse train [57]. Consequently, the force generated by FES depends on the number of motor units recruited and their activation rate [58]. By modulating the stimulation parameters, the tension produced by the target muscles can be controlled and, consequently, the torque and movement generated around a joint can be regulated [57].

The functional nature of FES is particularly relevant to motor rehabilitation. Unlike conventional muscle stimulation, FES provides stimulation in a coordinated and rhythmic pattern associated with the targeted movement, which may contribute to the recovery of muscle control. Furthermore, FES combines neuromuscular training with principles of motor learning, potentially enhancing the recovery of muscle function. In the context of ACLR rehabilitation, for example, FES applied to the quadriceps during gait may enhance muscle function and contribute to more symmetrical loading of the lower limbs during walking [11]. More broadly, studies have demonstrated that FES can activate the quadriceps, improve lower-limb symmetry, and optimize gait patterns following ACLR [58]. Electrical stimulation of lower-limb muscles can also induce action potentials in nerve axons and, by varying the stimulation current, produce different levels of muscle contraction, potentially increasing voluntary muscle contraction strength and improving functional anticipatory activity of the lower limb [58].

Moran et al. (2019) [11] conducted a randomized controlled pilot study to investigate the effects of functional electrical stimulation (FES) of the quadriceps during gait in the early stages of rehabilitation after ACL reconstruction. Forty patients were initially randomized to an FES group, which received quadriceps stimulation synchronized with gait, or to an NMES group, which received conventional stimulation; 23 participants completed the study. Both interventions were performed for 10 minutes, three times a week, in addition to a standardized rehabilitation program. In the FES group, stimulation was synchronized with gait events, beginning at heel strike and generally continuing through the loading response phase. In contrast, the NMES group received stimulation with a 10-second on/10-second off cycle. Outcomes included gait speed, gait symmetry, and peak isometric quadriceps strength. After four weeks, both groups recovered preoperative gait speed and symmetry, with no significant differences between the groups. However, the FES group recovered significantly more quadriceps strength, reaching 82% of preoperative strength, compared to 47% in the NMES group ($p = 0.02$), and demonstrated greater strength symmetry between limbs (0.63 ± 0.15 vs. 0.39 ± 0.18; $p = 0.01$). These findings suggest that integrating FES with gait training may provide an advantage over conventional NMES for early recovery of quadriceps strength after ACL reconstruction.

2.3.2 The Electrode-Skin Interface: Impedance and Low-Frequency Limitations

Transcutaneous Electrical Stimulation modalities, as the name indicates, apply electrical currents through the skin, also called the integumentary system. This organ, the largest in the human body, is divided into the epidermis, an outermost avascular layer populated by keratinized epithelial cells; the dermis, a core layer primarily composed of connective tissue; and the hypodermis, or subcutaneous layer, the deeper fat-storing layer that anchors the whole system [59, 60]. This stratified, heterogeneous composition creates nonuniform opposition to current flow, so to understand how stimulation waveforms with different parameters (such as frequency or amplitude) are transmitted through the skin to reach deeper body structures, it may be useful to consider the electrical circuit equivalent presented in Figure 2.6.

This electrical circuit can be divided into 4 stages:

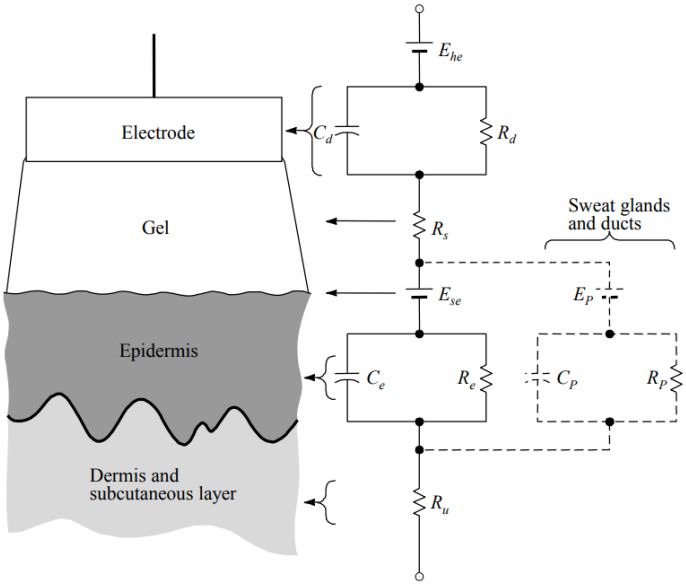


Figure 2.6: Equivalent circuit obtained when using one stimulation electrode against the skin (adapted from [61]).

1. The electrode-gel interface, modeled as a parallel RC circuit with a purely resistive impedance at low frequencies (R_d) and a capacitive reactance (C_d) that represents the charge separation at the interface.

2. The gel resistance (R_s), introduced by this electrolytic substance that is frequently placed under electrodes, enhances the contact between them and the skin by filling gaps at the interface [62].

3. The gel-epidermis interface, also represented as a parallel RC circuit. In this case, the capacitive branch (C_e) reflects the behavior of the *stratum corneum* (SC) [62], the uppermost epidermal layer formed by keratin and dead keratinocytes [59].

4. The dermis and underlying subcutaneous tissues are more conductive due to a higher water content than the epidermis [62], hence their representation as a single resistance (R_u).

An additional parallel RC path may be considered to account for the contribution of sweat glands and ducts. However, this additional path is usually neglected [61], and this shall be the case for this analysis. Therefore, the total impedance of the skin is obtained by adding the individual contributions from each of the highlighted stages:

$$Z_{total} = Z_{electrode\text{-}gel} + Z_{gel} + Z_{gel\text{-}epidermis} + Z_{dermis} = \frac{1}{j2\pi f C_d + \frac{1}{R_d}} + R_s + \frac{1}{j2\pi f C_e + \frac{1}{R_e}} + R_u = \frac{R_d}{1 + j2\pi f C_d} + R_s + \frac{R_e}{1 + j2\pi f C_e} + R_u \tag{2.1}$$

Looking at Equation 2.1, it is noticeable that the total skin impedance is inversely proportional to the frequency of the stimulation waveform, a relation introduced by the capacitive reactances observed at the electrode-gel and gel-epidermis interfaces. As a result, low-frequency currents face increased electrical opposition, making them less effective in stimulation applications that target deeper muscles or structures, since higher intensities are required to reach those targets [13]. In contrast, alternating waveforms of medium or high frequencies encounter a lower impedance and flow through tissues more easily, but fail to stimulate them directly, losing the therapeutic effect.

2.3.3 Interferential Therapy (IFT)

To overcome this trade-off, Austrian physicist Hans Nemec introduced the concept of interferential therapy (IFT) in the early 1950s [63]. This stimulation modality exploits the principle of wave interference, a physical phenomenon in which two or more alternating waveforms interact to form a resultant signal of greater (constructive interference) or lower (destructive interference) amplitude [13]. The magnitude of this combination depends on the phase difference between the interacting waveforms, provided that they share the same frequency. The closer their difference in position (in degrees) is to a multiple of 180°, the greater the magnitude of the interference.

Furthermore, interference also occurs between waves of slightly differing frequencies. In this case, the two waveforms merge into a signal whose amplitude oscillates up and down, as the waves alternately move into each other and drift out of phase, respectively [13, 64]. This pattern is known as a beat in acoustics, and the rate at which the new signal's amplitude swings is the beat frequency. To better understand this "beat effect", one can translate it into mathematical terms using sinusoids:

$$\begin{aligned}
&\sin(2\pi f_1 t)+\sin(2\pi f_2 t)=\sin(2\pi\frac{f_1+f_1}{2}t)+\sin(2\pi\frac{f_2+f_2}{2}t)=\\
&\sin(2\pi\frac{f_1+f_2}{2}t+2\pi\frac{f_1-f_2}{2}t)+\sin(2\pi\frac{f_1+f_2}{2}t-2\pi\frac{f_1-f_2}{2}t)=\\
&\sin(2\pi\frac{f_1+f_2}{2}t)\cos(2\pi\frac{f_1-f_2}{2}t)+\cos(2\pi\frac{f_1+f_2}{2}t)\sin(2\pi\frac{f_1-f_2}{2}t)+\\
&\sin(2\pi\frac{f_1+f_2}{2}t)\cos(2\pi\frac{f_1-f_2}{2}t)-\cos(2\pi\frac{f_1+f_2}{2}t)\sin(2\pi\frac{f_1-f_2}{2}t)=\\
&2\sin(2\pi\frac{f_1+f_2}{2}t)\cos(2\pi\frac{f_1-f_2}{2}t)
\end{aligned} \tag{2.2}$$

Equation 2.2 establishes that the interaction between two sine waves results in a signal with a frequency that is the average of the carrier frequencies (f_1 and f_2) multiplied by a cosinusoidal term with a frequency of half the difference between f_1 and f_2 [64]. Overall, the process is an example of amplitude modulation [13] that depends on an envelope that crosses zero twice per wavelength. Thus, the beat frequency is actually twice the frequency of the cosine envelope, corresponding to the difference between the two frequencies.

$$f_{beat}=|f_1-f_2| \tag{2.3}$$

For instance, in a scenario with two sine waves alternating at $f_1 = 1000$ Hz and $f_2 = 1050$ Hz, the resulting interference pattern would exhibit a frequency of 1025 Hz and an amplitude modulated at a rate of 50 Hz, as shown in Figure 2.7. Interference is not limited to sine waves and can also occur with rectangular oscillator patterns [65].

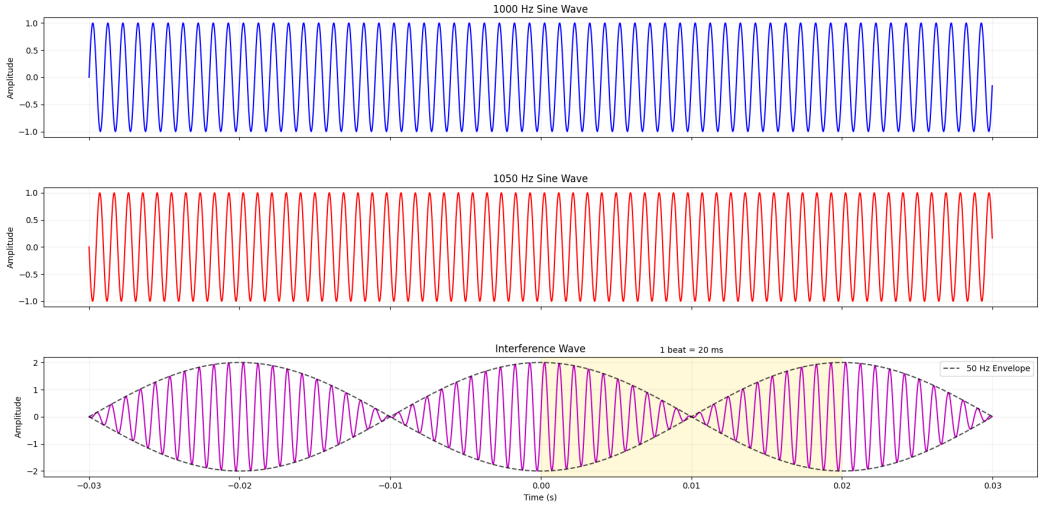


Figure 2.7: Example of wave interference between two sinusoids with slightly different frequencies.

In IFT, this low-frequency beat is either applied to or generated directly inside the treatment area. In the first case, the stimulation device generates the amplitude-modulated pattern beforehand, requiring only two electrodes to deliver a "premodulated interferential current (IFC)". In the second case, a quadripolar electrode arrangement is necessary to output the currents separately, allowing them to merge into a "true IFC" within the target tissue, combining the therapeutic benefits of low-frequency signals with the lower skin impedance of medium-frequency stimuli, allowing the current to reach deeper tissues within the human body painlessly.

Beyond bridging the impedance trade-off between low- and medium-frequency stimulation, IFT is used clinically for two complementary purposes: analgesia and muscle activation. Its pain-relieving effect is generally attributed to two physiological mechanisms. First, activation of large, myelinated sensory afferents by the interferential beat is thought to interfere with the transmission of nociceptive signals carried by smaller fibers at the spinal cord, in accordance with the gate-control theory of pain [13]. Second, sustained stimulation may recruit descending inhibitory pathways within the central nervous system, promoting the release of endogenous opioids and other inhibitory neurotransmitters that further suppress nociceptive transmission; concurrent muscle contractions may also improve local circulation and help clear inflammatory substances from the treated region [13].

When the amplitude-modulated envelope reaches motor nerves rather than purely sensory afferents, IFT can also elicit muscle contractions. Because both interfering carriers alternate

throughout the stimulation cycle rather than being switched on and off as a single pulsed waveform, the resulting motor-unit recruitment pattern has been described as more asynchronous than that produced by conventional low-frequency NMES, potentially resembling voluntary contraction more closely [13]. This property, combined with lower perceived discomfort at the skin-electrode interface, has motivated the use of IFT not only for analgesia but also to assist strengthening exercises, gait training, and other functional tasks. This is a similar rationale to the one underlying the EMG-triggered, IFT-based FES approach developed in this dissertation.

Despite this physiological plausibility, the strength of clinical evidence supporting IFT varies considerably by indication. Beyond pain and muscle activation, IFT has also been proposed for edema reduction, improvement of peripheral blood flow, temporary reduction of spasticity, pelvic-floor stimulation for urinary incontinence, and support of bone and soft-tissue healing, although the underlying trials remain limited, particularly for the latter applications [13]. Reflecting this, a 2025 UnitedHealthcare medical policy on electrical stimulation for pain and muscle rehabilitation explicitly notes that the evidence for IFT and related modalities remains inconsistent, frequently drawn from small, short-term, or inadequately controlled studies, and does not currently list IFT among the modalities it considers medically necessary [51]. This gap between decades of physiological rationale and comparatively sparse controlled clinical validation motivates the discussion revisited later in this chapter, where the more recent characterization of IFT-based proportional force control by Osorio et al. [14] is discussed.

2.4 Surface Electromyography (sEMG) for Neuromuscular Feedback

Surface electromyography (sEMG) is a non-invasive electrophysiological diagnostic technique that records the electrical activity of skeletal muscles through electrodes placed directly on the skin surface [66]. At its primary physiological level, skeletal muscle functions to convert chemical energy into mechanical work through contraction [67]. Controlled by somatic motor nerves originating in the brain or spinal cord, each muscle consists of cylindrical fibers bounded by connective tissue. In contrast, a single motor neuron can branch to innervate multiple muscle fibers, each innervated by exactly one motor neuron (Figure 2.8) [67].

By registering the electrical currents generated during voluntary movements or artificially elicited contractions, sEMG provides a quantitative window into neuromuscular function, allowing clinicians to evaluate muscle activation timing, coordination, myoelectric manifestations of fatigue, and changes in neural drive following therapeutic interventions [66]. Unlike intramuscular needle electromyography, which requires electrode insertion to isolate deep or single-fiber potentials, sEMG captures the collective electrical manifestation of underlying muscle activity, making it a non-invasive procedure well suited for dynamic kinesiological evaluations and rehabilitation protocols [68].

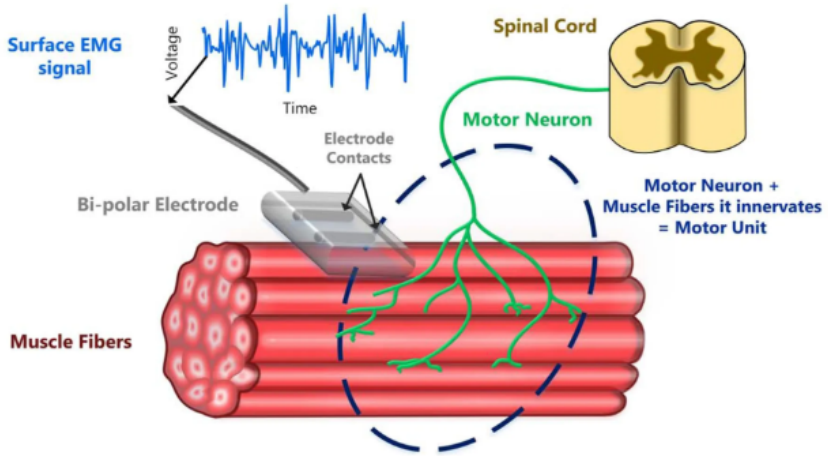


Figure 2.8: Schematic illustration of a motor unit and how a surface EMG signal could be recorded from a muscle using a bipolar electrode (two electrode contacts) (adapted from [67]).

2.4.1 Physiological Origins of the EMG Signal and Motor Learning

At its fundamental physiological level, the sEMG signal represents the peripheral electrical manifestation of a complex hierarchy of motor commands originating within the central nervous system (CNS) [69]. The primary functional entity of this neuromuscular pathway is the motor unit (MU) (Figure 2.9), classically defined by the British physiologist Charles Sherrington as the "final common pathway" of motor control [70]. The complex sEMG signal arises from two principal bioelectric activities: the propagation of motor nerve impulses across neuromuscular junctions and the propagation of muscle action potentials along the sarcolemma and T-tubular system during excitation-contraction coupling [67].

Although individual muscle fibers generate weak electrical impulses (<100 µV), the spatial and temporal summation of these potentials across an activated motor unit forms a Motor Unit Action Potential (MUAP) [66, 67]. Because active MUs fire asynchronously with varying amplitudes, and because membrane propagation involves continuous depolarization and repolarization, the resulting surface voltage exhibits complex positive and negative electrical spikes that strongly depend on electrode placement relative to the active fibers [67].

To control muscle force and execute coordinated movements, the central nervous system regulates two key physiological mechanisms: the number of active motor units (recruitment) and the frequency of motor neuron impulses (rate coding) [67]. Together, these mechanisms dictate the overall intensity of the neural drive. The recruitment and rate coding dynamics of the motor neuron (MN) pool are determined by the amplitude of the net excitatory synaptic drive the MNs receive and the intrinsic excitability of the MNs, which is continuously influenced by neuromodulatory inputs [71].

Strategy-wise, motor units are recruited according to an orderly physiological rule known as the "size principle," introduced by Henneman and Mendell in the 1960s [72]. Under this principle, smaller motor neurons innervating fatigue-resistant, slow-twitch muscle fibers are activated

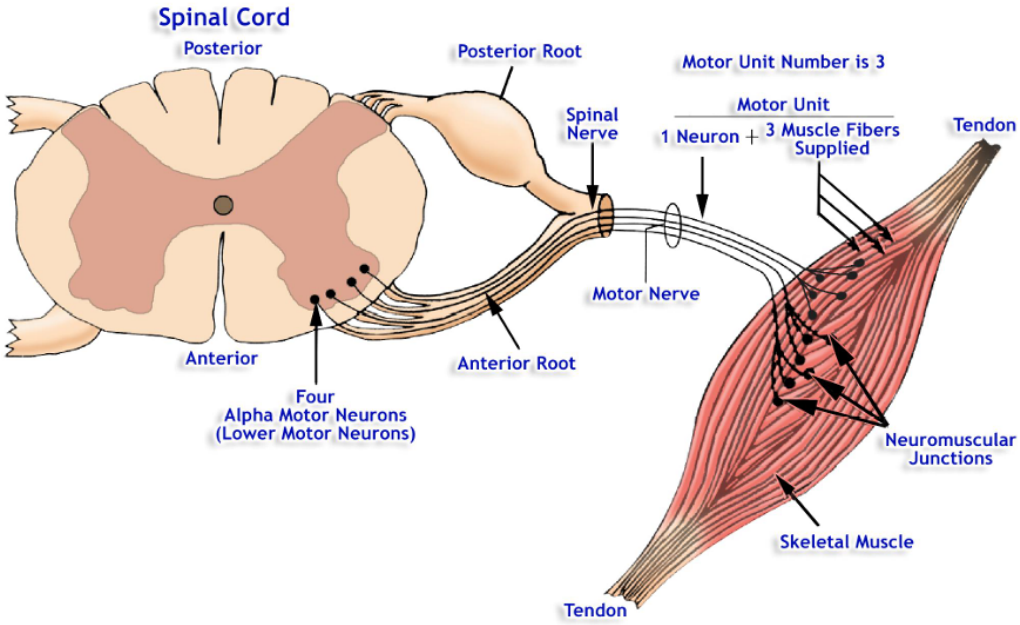


Figure 2.9: Example of Motor Units [67].

first during low-intensity tasks. As mechanical force demands increase, the nervous system progressively recruits larger motor neurons supplying fast-twitch fibers, which generate greater force but fatigue more rapidly [72]. Managed through central synaptic input and the intrinsic membrane properties of motor neurons, this strategy enables graded contractions, allowing the muscle to adapt its force output proportionally to the charge of an amplifier, which is important because it measures forces [72, 67]. Furthermore, even at rest *in vivo*, skeletal muscles exhibit tonus, a constant state of slight tension caused by the readiness without inducing fatigue [67].

2.4.2 sEMG Acquisition and Conditioning

To acquire any electrophysiological signal, it is necessary to read and amplify it while filtering unwanted contributions that hinder visualization and interpretation. According to a tutorial by Merletti and Cerone [73], four important features of an amplifier are important considerations, since it measures

1. Input Impedance
2. Differential and Common Mode Amplifications
3. Power Supply Rejection Ratio (PSRR)
4. Input Voltage and Current Noise

The Input Impedance (Z_i) of an amplifier is an important consideration because it measures how much current the component draws from the source signal. Ideally, the FE amplifier should require a minimal current to operate, so that it amplifies the signals of interest without distortion.

This opposition to current flow exists between each differential input and the reference, and is modeled as a large resistance R_i in parallel with a parasitic capacitance C_i. The latter introduces a frequency (f) dependence for this quantity, as evidenced in Equation 2.4.

$$|Z_i(f)| = \frac{R_i}{\sqrt{1+(2\pi R_i C_i)^2}} \tag{2.4}$$

The previous relation can be used to compute the values of Z_i for a given frequency, as long as the values of both R_i and C_i are correctly reported by the manufacturers in the amplifier's datasheet [73]. In particular, it is important to know the value of Z_i at the power line frequency (50 Hz). To understand its importance, it is useful to consider circuit B) from Figure 2.10, which represents the FE amplifier with shorted inputs and the common-mode voltage sensed between node A and ground.

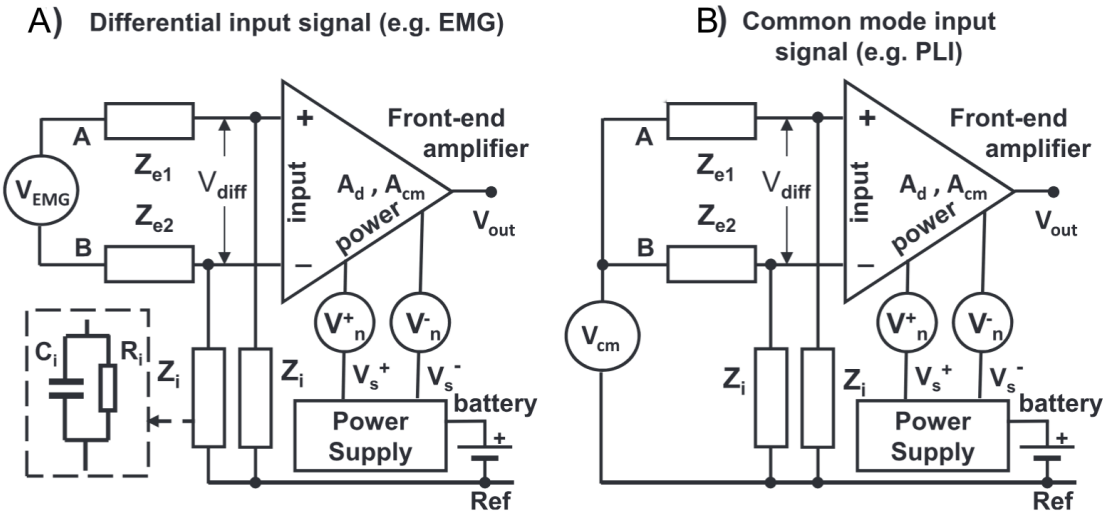


Figure 2.10: Model circuit of the front-end amplifier with differential A) and common-mode B) input signals. Adapted from [73].

This arrangement introduces two voltage dividers, created by the impedances of the electrode-skin interfaces, Z_{e1} and Z_{e2}, and the input impedance of the FE Amplifier, Z_i. Hence, the voltages in A (V_A) and B (V_B) become:

$$V_A = V_{CM} \times \frac{Z_i}{Z_i + Z_{e1}}; V_B = V_{CM} \times \frac{Z_i}{Z_i + Z_{e2}}. \tag{2.5}$$

Accordingly, and considering that Z_{e1} and Z_{e2} are much less than Z_i, the differential voltage V_{diff} seen at the input of the amplifier can be written as follows:

$$V_{diff} = V_A - V_B = V_{CM} \times \frac{Z_i(Z_{e2} - Z_{e1})}{(Z_i + Z_{e2})(Z_i + Z_{e1})} \approx V_{CM} \times \frac{Z_{e2} - Z_{e1}}{Z_i}. \tag{2.6}$$

Equation 2.6 reveals that any existing difference between the electrode-skin impedances leads to an unwanted differential voltage that will be amplified A_d times, ultimately contaminating the

output signal. Thus, the amount of common-mode voltage is inversely proportional to Z_i. Furthermore, because one of the main common-mode signals picked up by an acquisition system is 50/60 Hz power-line interference (PLI), the higher the input impedance at this frequency, the less PLI is added to the output signal. For this reason, the authors of the tutorial suggest selecting a device with a minimum of 300 MΩ of input impedance at the power line frequency [73].

Two other features worth considering when choosing an FE amplifier are its differential (A_d) and common-mode (A_{CM}) amplification gains. These factors play a significant role in amplifying the relevant components of the output signal while suppressing unwanted ones. Considering diagram A in Figure 2.10 and the differential term related to electrode-skin impedances established in Equation 2.6, the total amplifier output voltage becomes:

$$V_{out} = V_{CM} \times A_{CM} + V_{diff} A_d = V_{CM}(A_{CM} + A_d \frac{Z_{e2} - Z_{e1}}{Z_i}) + A_d V_{EMG}, \tag{2.7}$$

where the last parcel corresponds to the amplified signal of interest. The first term is related to the common-mode voltage and imposes a minimal A_{CM} factor $(A_{CM} << 1)$ for a device to be considered a "good" amplifier [73]. To further analyze these effects that appear at the output of the amplifier, one can divide both sides of Equation 2.7 by A_d and obtain an equivalent input voltage (V_{in-eq}) from V_{out}:

$$V_{in-eq} = V_{CM}(\frac{A_{CM}}{A_d} + \frac{Z_{e2} - Z_{e1}}{Z_i}) + V_{EMG} = V_{CM}(\frac{1}{CMRR} + \frac{Z_{e2} - Z_{e1}}{Z_i}) + V_{EMG}. \tag{2.8}$$

Equation 2.8 introduces the notion of Common-Mode Rejection Ratio (CMRR), defined by the quotient between differential and common-mode gains. CMRR indicates how well a device rejects or attenuates common-mode signals, such as PLI. From the previous relation, it is clear that CMRR should be large, with acceptable values ranging from 90 to 110 dB at 50-60 Hz to effectively reject common-mode voltages on the order of V while keeping the mV differential signals intact. In addition, the previous equation states that the CMRR has the same relevance as the $\frac{Z_{e2}-Z_{e1}}{Z_i}$ term, meaning that, in practice, it is not worth buying an expensive integrated circuit with extremely high CMRR unless the skin-electrode impedances can be balanced by proper skin treatment [73].

In addition to rejecting common-mode signals, FE amplifiers should also block fluctuations produced by the supply DC/DC converters. This capability is measured by the Power Supply Rejection Ratio (PSRR), making it another relevant parameter when selecting a device for sEMG acquisition. A typical value is 60 dB [73].

Finally, the intrinsic noise of the FE amplifier also needs to be considered, since it limits the smallest detectable sEMG signal [73]. This internal noise is modeled by an equivalent input voltage noise spectral density (e_n, expressed in $\mathrm{nV}/\sqrt{\mathrm{Hz}}$) in series with the inverting input of the amplifier, and an equivalent input current noise spectral density (i_n, expressed in $\mathrm{fA}/\sqrt{\mathrm{Hz}}$ or $\mathrm{pA}/\sqrt{\mathrm{Hz}}$) flowing through the source impedance R_s [74]. A model diagram of this circuit is presented in Figure 2.11.

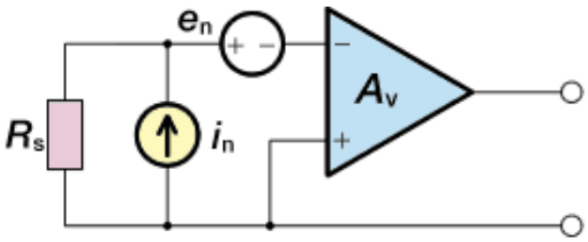


Figure 2.11: Model Diagram of the Equivalent Input Noise of an FE Amplifier. Adapted from [75]

.

Considering the total electrode-skin source resistance ($R_s = R_{e1} + R_{e2}$) and the thermal (Johnson-Nyquist) noise generated by the electrodes ($e_t = \sqrt{4kTR_s}$, where k is the Boltzmann constant and T is the absolute temperature) [76], the total equivalent input noise spectral density $e_{n,\text{total}}$ can be expressed as:

$$e_{n,\text{total}} = \sqrt{e_n^2 + (i_n \cdot R_s)^2 + 4kTR_s}. \tag{2.9}$$

Integrating Equation 2.9 across the typical sEMG bandwidth B (commonly 20 Hz to 500 Hz) [73] yields the total root-mean-square (RMS) noise voltage:

$$V_{n,\text{RMS}} = \sqrt{\int_{f_L}^{f_H} e_{n,\text{total}}^2(f)\, df}. \tag{2.10}$$

For surface EMG applications, where signal amplitudes during weak contractions can fall below 10-20 μV, the total baseline RMS noise should ideally not exceed 1-2 μV_{RMS} across the full bandwidth [73]. Because the electrode-skin interface typically presents impedances in the range of tens to hundreds of kΩ, a FE amplifier with FET (or CMOS) inputs is often preferred over bipolar (BJT) inputs [77]. Although FET amplifiers generally exhibit slightly higher voltage noise (e_n), their extremely low current noise ($i_n \approx \text{fA}/\sqrt{\text{Hz}}$) prevents the $i_n \cdot R_s$ term in Equation 2.9 from dominating and degrading the signal-to-noise ratio (SNR).

The noise that can affect the sEMG signal is not limited to the one generated inside the FE amplifier. As previously mentioned, PLI is a common contaminant of biopotential signals, mainly due to the stray capacitive coupling between the subject and the power line source. This parasitic capacitance can be mitigated to a limited extent by keeping power cables and equipment at some distance from the participant and the amplifier, but this is sometimes not possible/sufficient. In that case, PLI can be reduced by employing the Driven Right Leg (DRL/RLD) technique, that is, by extracting V_{CM} from the inputs (subject or internal circuitry of the FE amplifier), inverting it, and actively driving it back to the subject's body via a reference electrode. Although complete cancellation is unachievable, attenuations of 5-10 times are possible with this topology [73].

In addition to using the DRL sub-circuit, one can incorporate an analog notch filter in the signal conditioning block. This approach alters the features of the sEMG and, according to Merletti and

Cerone, is only acceptable for biofeedback applications. Also, such a filter can only be designed to act on the first harmonic (50 Hz), meaning that the 100 and 150 Hz power-line harmonics may pass unaffected and contaminate the signal of interest. Naturally, more complex solutions exist to tackle the harmonic issue, namely advanced digital notch filters and spectral interpolation [73, 78].

Other notable sEMG noise contributors include low-frequency artifacts caused by loss of contact between the electrode and the subject due to cable movement, for instance, and background random noise. These unwanted signals can be reduced by adding a high-pass filter (HPF) and a low-pass filter (LPF), respectively, to the analog filtering segment of the acquisition circuit. For the HPF, a cut-off frequency (f_c) between 10 and 30 Hz is recommended, whilst a f_c of 350-500 Hz is suggested for the LPF. Both filters should be at least order 2; that is, they should decrease the signal amplitude by 40 dB per decade above/below the chosen cut-off frequencies [73].

Overall, the sEMG signal should be amplified by an FE amplifier chosen according to the four characteristics above and driven to a conditioning sub-circuit that acts as a bandpass filter for frequencies within the electromyographic bandwidth.

2.5 Rationale for EMG-Controlled, IFT-Based FES

Functional Electrical Stimulation (FES) relies on low-frequency electrical pulses to trigger targeted motor recruitment during voluntary movement or functional tasks. However, as detailed in subsection 2.3.2, conventional low-frequency transcutaneous stimuli encounter high capacitive skin impedance (Z_{total}), which requires high voltage amplitudes to reach deeper neuromuscular structures. This high skin resistance often causes patient discomfort, superficial sensory irritation, and accelerated muscular fatigue due to non-physiological motor-unit recruitment. To overcome these skin impedance limitations, Interferential Therapy (IFT) uses medium-frequency carrier waves that pass through the skin layers with minimal electrical resistance, generating a therapeutically active low-frequency beat (f_{beat}) at the depth of the target tissue; integrating IFT with FES results in IFT-based FES.

Although IFT is well established in pain management, its application within FES remains under-explored. A recent study by Osorio et al. [14] provided key empirical evidence that IFT can achieve proportional control of muscle force, along with greater sensory comfort. Although higher energy requirements and complex neural activation mechanisms remain technical challenges, these findings confirm the therapeutic potential of IFT-based FES for motor rehabilitation.

Furthermore, conventional FES systems traditionally operate in a passive, open-loop mode, delivering preprogrammed electrical patterns where the user can only switch the device on or off [79]. Clinical evidence indicates that passive stimulation yields inconclusive results in motor rehabilitation because it lacks active patient involvement and fails to induce systematic neurorehabilitation across brain-nerve-muscle networks [79]. Optimal motor recovery requires the explicit integration of central motor intention. Surface electromyography (sEMG) provides a direct readout of neural activity and subjective movement intention, serving as a physiological control input to bridge voluntary effort with electrical assistance [80].

Chapter 3

Materials and Methods

3.1 sEMG Acquisition Circuit

As mentioned previously, the system required a specific circuit to acquire surface electromyographic signals. For this purpose, we used an analog printed circuit board (PCB), whose design was inspired by a publicly available ECG Signal Conditioner [81]. A schematic of this circuit is presented in Figure 3.1

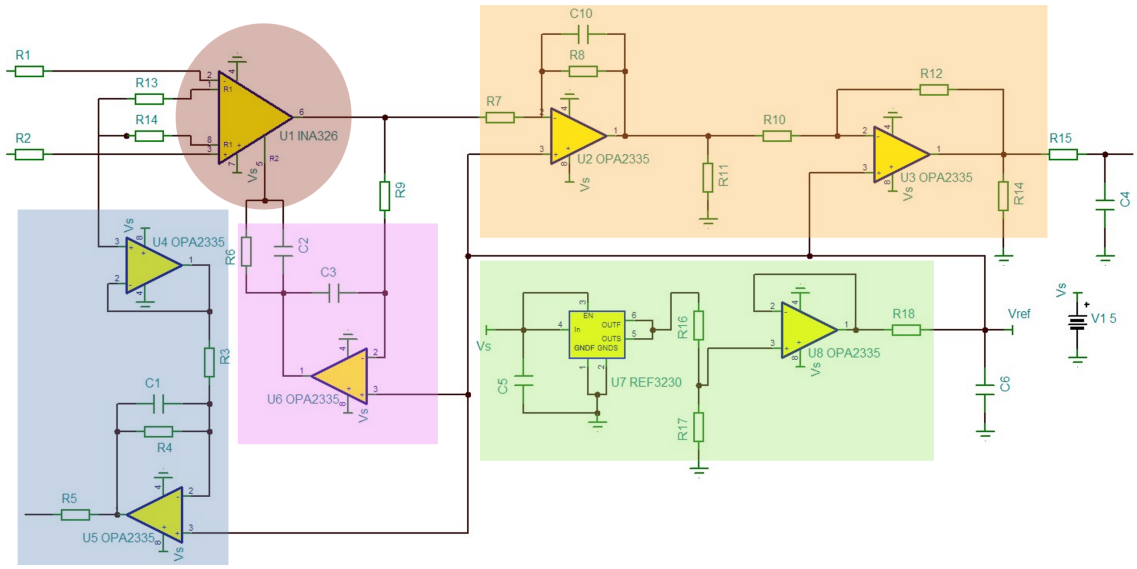


Figure 3.1: Schematic of the sEMG acquisition PCB, with colored boxes highlighting the individual functional stages.

The differential biopotential is first amplified by an INA326 (U1) instrumentation amplifier (highlighted in red), a precision, true 5 V single-supply device with rail-to-rail input and output, whose gain is set externally by R13 and R14 on its dedicated R_1 and R_2 pins as $G = 2R_2/R_1$ [82]. The electrode leads reach its inputs through the series resistors R1 and R2, which limit the input current and attenuate radio-frequency interference. This device complies with the minimum required features specified in Subsection 2.4.2, having a typical CMRR of 114 dB, an input

impedance of 623 MΩ at 50 Hz, a PSRR of 94 dB [83] and an input-referred noise voltage of 0.72 μV_{RMS} in the 20–500 Hz range, a value obtained using Equation 2.10.

The remaining blocks use six OPA2335 amplifiers (three dual packages), miniature high-precision operational amplifiers also capable of operating on a 5V single supply. U6 forms a DC servo, as highlighted in pink in the figure above, integrating the output of U1 sampled through R9, with R6 in parallel with C2 and with C3 setting the time constant, and returns the result to the reference input of the instrumentation amplifier, thereby canceling the slowly varying electrode-skin offsets. The signal is then amplified and band-limited by two cascaded inverting stages (in yellow) U2 (input R7, feedback R8 in parallel with C10) and U3 (input R10, feedback R12), both referenced to V_{ref}, and finally by the passive pole formed by R15 and C4. These components were tuned to place the high-frequency cutoff near 500 Hz.

The remaining blocks use six OPA2335 amplifiers (three dual packages), miniature high-precision operational amplifiers also capable of operating on a 5 V single supply. U6 forms a DC servo (marked in pink), integrating the output of U1 sampled through R9, with R6 in parallel with C2 and with C3 setting the time constant, and returns the result to the reference input of the instrumentation amplifier, thereby canceling the slowly varying electrode-skin offsets. The signal is then amplified and band-limited by two cascaded inverting stages (shown in yellow) U2 (input R7, feedback R8 in parallel with C10) and U3 (input R10, feedback R12), both referenced to V_{ref}, and finally by the passive pole formed by R15 and C4. These components were tuned to place the high-frequency cutoff near 500 Hz.

To mitigate the 50/60 Hz powerline interference and improve the overall CMRR, a Right Leg Drive (RLD, highlighted in blue) topology senses the common-mode potential at the amplifier inputs with the unity-gain buffer U4 and reinjects it, inverted and low-pass shaped by the integrator U5 (input R3, feedback R4 in parallel with C1), into the reference electrode through the current-limiting resistor R5. The DC bias about which this single-supply chain operates is generated by U7, a REF3230 precision voltage reference decoupled by C5, divided by R16 and R17 and buffered by U8. This V_{ref} net, distributed through R18 and decoupled by C6, biases the servo loop, the non-inverting inputs of U2 and U3, and the RLD stage, while all amplifiers are powered from the 5 V rail V1. It therefore provides the stable baseline about which the AC signal swings before digitization by the analog-to-digital converter (ADC) of the laptop's sound card.

The transfer characteristic of the assembled board was verified experimentally by injecting a sinusoidal differential signal of fixed amplitude at the electrode terminals and sweeping its frequency, while the output amplitude was recorded and converted to gain. The resulting magnitude response is shown in Figure 3.2, plotted both in decibels and as a linear voltage gain.

The measured response confirms the intended bandpass behaviour. Between 10 Hz and roughly 250 Hz the module presents an essentially flat passband with a gain of approximately 52 dB, that is, a linear voltage gain close to 400 V/V, which is the value used to convert the digitized samples back into electrode-referred voltages in Subsection 3.3.4. Above this plateau the gain decreases monotonically, crossing the -3 dB level relative to the passband at approximately 500 Hz, in agreement with the cutoff targeted by the design of the U2 and U3 stages and by the output net-

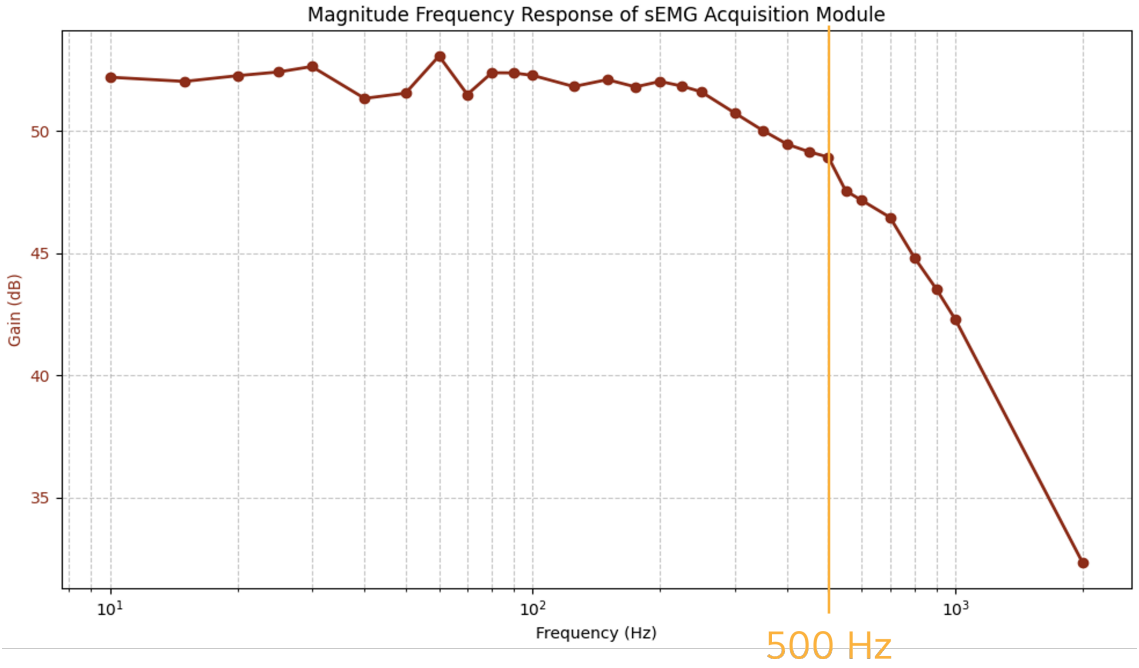


Figure 3.2: Measured magnitude of the frequency response of the sEMG acquisition module, with the yellow line marking the 500 Hz upper cutoff frequency.

work R15–C4. The attenuation then steepens progressively, reaching about 42 dB at 1 kHz and 32 dB at 2 kHz; the slope over the last octave, close to 10 dB/octave, is consistent with the asymptotic behaviour of two cascaded poles rather than a single-pole roll-off, and provides the anti-aliasing margin required by the sound-card digitization described in Subsection 3.3.4.

A localized reduction of the order of 1 dB is visible in the 40–70 Hz interval, superimposed on the otherwise flat passband. This attenuation is attributed to the common-mode feedback introduced by the RLD loop: by actively driving the subject's reference potential, the loop reduces the common-mode component at the amplifier inputs in the frequency range where powerline interference is concentrated, and this rejection is partially reflected in the measured differential transfer function. The feature is therefore a deliberate consequence of the driven-reference topology rather than a deficiency of the signal path, and it is beneficial in the present application, since the sEMG spectral content lost around 50 Hz is small compared with the powerline interference suppressed there.

3.2 IFT stimulation circuit

Complementing the sEMG acquisition module, a dedicated hardware stage was needed to amplify the software-generated stimulation signals and safely deliver them to the patient's knee. For this purpose, the stimulation circuit was designed around two main components: an audio amplifier and a transformer, as shown in Figure 3.3.

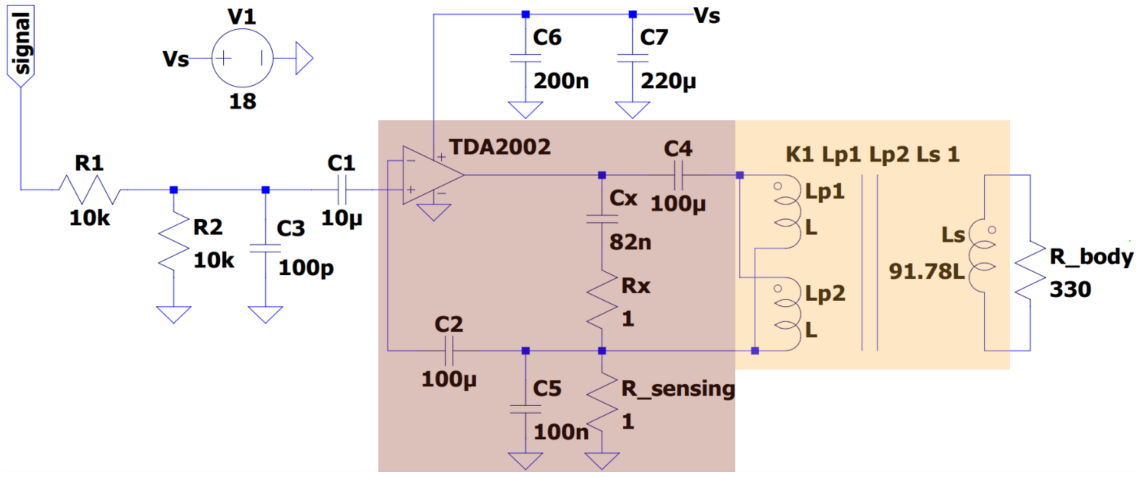


Figure 3.3: Schematic of the developed stimulation circuit, with colored boxes to differentiate the main functional blocks.

At the core of the amplification network is the TDA2002 audio power amplifier, selected for its high output current capability, low harmonic and crossover distortion, and integrated thermal-limiting and short-circuit protection features [84]. To meet the voltage and power demands of the stimulation waveforms, the amplifier is powered by a dedicated 18 V supply. As shown in Figure 3.3, the device is used as the power-output stage, taking the computer-generated signal through the input network formed by R_1, R_2, C_1 and C_3. Resistors R_1 and R_2 form a voltage divider that attenuates the signal to about half, allowing a wider voltage range without saturating the amplifier's output. In turn, C_3 acts as a high-frequency noise-suppression capacitor, while C_1 AC-couples the source and blocks the DC offset from the sound card.

Contrary to the manufacturer's suggested application circuit, the TDA2002 is here configured as a transconductance amplifier (displayed in red), where the transformer's winding is driven from the amplifier output through the coupling capacitor C_4, while the winding's return current flows through the 1Ω sensing resistor, $R_{sensing}$. Consequently, the voltage developed across this resistor equals the primary current, i.e., $V_{sensing} = I_p \times R_{sensing} = I_p$. The sensed voltage is filtered by C_5 and AC-coupled to the TDA2002 inverting input through C_2, thereby establishing the negative feedback that causes the amplifier to adjust its output voltage in response to changes in primary current.

Moreover, the $C_x - R_x$ series branch is connected across the transformer primary path and functions as a Zobel/Boucherot output-stability network [85]. At lower frequencies, C_x presents a high impedance, and the branch has negligible influence on the intended stimulation waveform. At higher frequencies, its impedance decreases and the branch provides a damped, predominantly resistive load to the amplifier. This reduces high-frequency ringing and improves stability when driving the inductive transformer primary.

Finally, the transformer itself (shown in yellow) is the interface between the current-regulated amplifier stage and the subject's knee. For this purpose, the two independent 24 V, 0.21 A coils were connected in parallel to make up the primary winding, leaving the 230 V coil as the secondary

winding. This arrangement is the opposite of the transformer's nominal configuration [86] and was chosen for step-up operation, allowing the relatively low compliance voltage available from the TDA2002 supply to be transformed into a substantially higher output voltage at the electrode terminals.

This voltage amplification comes at the expense of a proportionally lower available output current, which follows directly from how the transformer's turns ratio is established. Both windings are wrapped around the same ferromagnetic core and are therefore linked by a common time-varying magnetic flux, $\Phi(t)$, produced by the current flowing in the driven winding. By Faraday's law of electromagnetic induction, the electromotive force (EMF) induced in a winding is proportional to its number of turns and to the rate of change of this shared flux:

$$e_p = -N_p \frac{d\Phi}{dt}, \qquad e_s = -N_s \frac{d\Phi}{dt}, \tag{3.1}$$

where N_p and N_s are the number of turns of the primary and secondary windings, respectively, and the negative sign, consistent with Lenz's law, indicates that the induced EMF in each coil opposes the flux change that produces it. Since the same flux $\Phi(t)$ links both windings, dividing these two expressions eliminates the flux-derivative term and yields the ideal transformer voltage relation:

$$\frac{V_s}{V_p} = \frac{N_s}{N_p} = n, \tag{3.2}$$

with n being the transformer's turns ratio. As evidenced by the above relation, n can be obtained either from the number of turns of each winding or the voltage across them. Hence, by applying the known primary and secondary voltages from the manufacturer [86] to Equation 3.2, the turns ratio can be estimated:

$$n = \frac{230}{24} \approx 9.58. \tag{3.3}$$

Combining the conservation of power principle followed by all electrical transformers,

$$P_p = P_s \Leftrightarrow V_p \times I_p = V_s \times I_s. \tag{3.4}$$

with Equations 3.2 and 3.3 gives the current transformation ratio:

$$\frac{I_s}{I_p} = \frac{1}{n} \approx 0.104. \tag{3.5}$$

It should be noted, however, that the transformer's ferromagnetic core and winding losses, along with leakage inductance, introduce frequency-dependent deviations from the ideal turns-ratio behavior. The current-feedback loop described previously partly compensates for these, but they are not eliminated at all frequencies.

In addition to voltage transformation, the transformer provides galvanic isolation between the amplifier/computer circuitry and the patient-side output, as energy transfer between the driven and

output windings occurs through magnetic coupling rather than a direct conductive path. Consequently, under normal operating conditions, there is no direct electrical connection between the subject and the mains/computer-referenced circuitry upstream of the transformer, reducing the risk of hazardous fault currents reaching the patient. This isolation is bounded by the TST 10/020 insulation rating, specified to withstand a 4 kV, 60 s dielectric test between windings [86].

This rating characterizes the transformer under standard industrial test conditions and does not, by itself, constitute certified medical-grade patient isolation. For the bench-prototype stage described here, this level of isolation was considered acceptable under supervised laboratory conditions. However, compliance with IEC 60601-1 leakage-current limits and IEC 61558-2-15 requirements for isolating transformers used in medical locations would be necessary before any unsupervised or clinical use [87, 88].

3.2.1 Bench characterization setup

The bench characterization of the circuit in Figure 3.3 consisted of applying a 4 kHz sinusoidal input to measure or derive the maximum and minimum current and voltage values at the transformer's windings. A sine wave was chosen instead of an interferential one, due to its better adequacy for the task. The carrier frequency was fixed at 4 kHz, a common carrier value used in IFT, and the operating system's volume percentage was swept over its full range for each measurement. The measurements were conducted with a WaveAce 102 oscilloscope from Teledyne LeCroy.

For these measurements, the secondary winding terminals were closed on a single 330 Ω resistor acting as a surrogate for the knee, rather than on a subject. This substitution was a prerequisite for the amplitude calibration of Subsection 3.3.2: since the current-feedback loop regulates primary current, the voltage and current actually delivered at the output depend on the load presented to the secondary, so a repeatable, exactly known load was required before any statement could be made about the current that a given volume setting would deliver. This value was not arbitrary, as a surrogate much lower than the physiological impedance would overstate the deliverable current, while one much higher would understate it; 330 Ω was instead derived from the electrical properties of the tissues interposed between the electrodes at the operating carrier frequency.

Biological tissues are dispersive dielectrics, so conductivity σ varies with frequency. Reference values for σ at a few kilohertz were taken from the IT'IS Foundation tissue-properties database, based on the Cole–Cole dispersion relations compiled by Gabriel [89, 90]: approximately 0.3–0.6 S/m for skeletal muscle, 0.1–0.2 S/m for cartilage, and only 0.02–0.04 S/m for adipose tissue. Modeling the inter-electrode current path as a series stack of skin, adipose, muscle, cartilage, muscle, adipose and skin layers, the total resistance can be estimated as

$$R_{\text{tissue}} \approx \sum_i \frac{t_i}{\sigma_i A_i}, \tag{3.6}$$

where t_i, σ_i and A_i are the thickness, conductivity and effective cross-sectional area of the i-th layer. Equation 3.6 shows why the poorly conducting layers dominate: adipose tissue and cartilage,

though comparatively thin, contribute disproportionately to the total resistance since σ_i appears in the denominator. For plausible layer thicknesses, electrode areas, and an inter-electrode path of roughly 8 cm, this places the expected load in the 200–600 Ω range. The latter interval is relatively broad, since it also depends on electrode placement, individual tissue composition, and hydration state.

Because this interval spans tissue conductivities that are each uncertain by a multiplicative factor rather than an additive offset, and R_{tissue} in Equation 3.6 scales with $1/\sigma_i$, the resulting composite resistance is better characterized by the geometric mean of the range than by its arithmetic mean. The geometric mean of 200 and 600 Ω,

$$R_{\text{gm}} = \sqrt{200 \times 600} \approx 346\ \Omega, \tag{3.7}$$

is markedly closer to the standard E12 value of 330 Ω than the arithmetic mean of 400 Ω is. The value of 330 Ω was therefore selected as the closest available standard resistance to R_{gm}, while also being available as a single, high-tolerance component rather than a series combination.

Representing the knee as purely resistive is justified by the joint's impedance being strongly resistance-dominated at these frequencies: in a case–control study of knee osteoarthritis, Das et al. reported a mean reactance of 79.96 ± 27.89 Ω against a mean resistance of 445.76 ± 150.43 Ω (phase angle $11.57 \pm 4.74^{\circ}$), with the control group showing an even smaller reactive fraction [91]. Neglecting a reactive component of this magnitude alters the impedance modulus by only a few percent. Hence, a resistive surrogate reproduces the amplifier's loading to within the uncertainty already introduced by electrode placement. Their resistance figures also corroborate the order of magnitude from Equation 3.6. However, read the comparison with care: those measurements used a whole-body, hand-to-foot analyzer rather than a local knee measurement and thus support the general magnitude and resistive character of the load rather than the specific value adopted here.

For every volume setting, the primary voltage, V_p, was obtained indirectly rather than measured directly across the winding terminals. The voltage at the node between the capacitor C_4 and the primary winding was measured relative to ground, together with the voltage developed through the sensing resistor, $R_{sensing}$. Knowing that the winding's return current flows entirely through the latter, subtracting the two readings isolates the voltage drop across the primary winding itself, yielding V_p. As established previously, the voltage measured across $R_{sensing}$ equals the primary current, allowing I_p to be obtained directly from an oscilloscope voltage reading.

The secondary-side quantities were not measured directly on the transformer's high-voltage winding. Still, they were instead derived from the primary voltage and current measurements using Equations 3.3 and 3.5, respectively.

3.3 Software

The FES system proposed in this thesis requires stimulation delivery to be made contingent on the patient's own volitional muscle activation within a structured rehabilitation session. To coordinate

this, a custom control application was implemented in Python using well-documented external libraries such as PySide6, NumPy, Matplotlib, pycaw, and sounddevice.

Real-time signal generation and acquisition were built on top of the sounddevice library, which provides low-latency access to the operating system's audio input/output streams via PortAudio [92], effectively repurposing the computer sound card as both a stimulation waveform generator and sEMG signal digitizer. The application is split into two windows, an operator-facing window and a patient-facing window, so that the clinician retains full access to configuration and monitoring. At the same time, the patient sees only the minimal feedback needed to guide the exercise. This split, developed with the PySide6 (Qt) [93] framework, is detailed in Subsection 3.3.6 and is referred to throughout the remaining subsections as the GUI thread.

Beyond this two-window split, the application follows a modular architecture that separates waveform synthesis, real-time streaming, sEMG acquisition, and session-level control logic into independent components that communicate through explicit, thread-safe interfaces. The following subsections describe each component, motivating each with the requirement it was built to satisfy before detailing its implementation.

3.3.1 Waveform Generation

The stimulation waveform must be computed identically whether it is previewed statically in the operator-facing window or generated continuously for output, so the operator gets a faithful representation of what stimulation will be administered to the patient. Additionally, for the interference waveform output stream, the carrier phase must be preserved across block boundaries because the audio callback synthesizes the target signal in discrete, fixed-sized blocks. Without this continuity, each new block would restart the phase at $t = 0$, and the resulting step in instantaneous amplitude would be delivered to the patient as an unwanted transient current spike through the electrodes.

The waveform math was therefore isolated from any GUI or audio-stream state, in the `signal_generator.py` script, which contains two functions: `carrier_wave`, intended for one-shot static plotting of the interference waveform, and `carrier_wave_stateful`. This variant persists the instantaneous phase between successive calls and is suitable for use in the synchronous callback. The streaming class, not this module, routes these carriers into true or premodulated IFC output, as described below.

3.3.2 Amplitude Calibration and Locking

The stimulation amplitude is set by the operating system's master output volume, without any dedicated amplitude controller in the software. This requires characterizing the relationship between the sound bar volume percentage and the resulting current delivered to the patient so that the clinician can select and report intensity in physically meaningful units rather than an arbitrary percentage. This characterization was carried out as part of the bench tests of the IFT stimulation circuit, with results reported in Section 4.1.

This requirement is met by the dedicated script `volume_lock_calibration.py`, which centralizes the calibration curve and the helper functions used to read and set the Windows volume level. Its core routine, `volume_to_predicted_current_ma()`, converts the master volume percentage into the corresponding maximum secondary-winding current, $I_{s,\max}$, via a cubic curve fitted to oscilloscope measurements, as will be detailed in Subsection 4.1. By giving a quantitative estimate of the current reaching the patient, this function allows the operator-facing interface to report predicted amplitude alongside the selected IFC parameters.

Calibration is performed once before a session using the same module's `get_output_volume_scalar()` and `set_output_volume_scalar()` functions, which query and set the Windows master volume through the pycaw library. The output level is first set to zero, a temporary stimulation stream is started, and the operator gradually raises the mixer level until the desired therapeutic sensation is reached. The resulting scalar value is stored and reapplied once, at the start of the following session. Additionally, a periodic poll in the operator-facing window (Subsection 3.3.6) tracks the current system volume throughout the session. It updates the displayed predicted current accordingly, so that any drift is visible to the operator rather than silently corrected. This should be regarded as a limitation of the current implementation relative to a true hardware- or OS-level lock.

3.3.3 Stimulation Streaming

Delivering the stimulation waveform continuously to the patient imposes a strict timing requirement: the audio output buffer must be refilled without interruption, since any discontinuity in the output stream translates directly into a gap in the delivered current through the electrodes. This is complicated because the operator stimulation parameters (f_1, f_2, beat frequency, and IFC mode) are set from the GUI thread. At the same time, the waveform itself must be generated on a separate, real-time audio thread that cannot block, allocate memory unpredictably, or perform disk I/O without risking underflows in the output.

Accordingly, a dedicated class named SignalStream manages output streaming and owns the sounddevice output stream and the phase accumulators of both carriers. Cross-thread access is resolved through a single-writer/single-reader pattern: the GUI thread writes parameters under a lock, and the callback reads them once per invocation. At the same time, the callback thread exclusively owns the phase state and needs no synchronization. Restricting the lock to this narrow scope minimizes the risk of priority inversion or missed audio deadlines.

Within the callback, the selected IFC mode determines whether the two sinusoidal carriers are written to separate stereo channels (true IFC) or numerically summed and normalized before being written to a single channel (premodulated IFC), with any unused channel set to silence. Buffer underflow events reported by PortAudio are counted but do not currently halt stimulation or notify the operator, thus favoring uninterrupted delivery over immediate error signaling. Preserving a record of the delivered waveform without compromising this real-time guarantee is achieved by buffering the generated samples and periodically flushing up to 64 blocks per call to a log file with

the same producer/consumer pattern, deferring the comparatively slow disk I/O to the 1 Hz control loop of Subsection 3.3.5 instead of executing it inside the audio callback.

3.3.4 sEMG Acquisition

Acquiring the conditioned sEMG signal from the circuit described in Section 3.1 imposes the same real-time constraints as stimulation output, but in the input direction, with the added requirement that the fixed gain introduced by the acquisition electronics be reversed in software. This lets the activation threshold be defined and reported in physiologically meaningful units (mV), rather than raw digitization values. In this case, a dedicated class owns a sounddevice input stream and, within its callback, converts each incoming audio block back into the original sEMG voltage by dividing out the acquisition gain. Since the sounddevice input is normalized to ± 1 at the digitizer's full-scale receiver input voltage, each block is rescaled by FS/G mV, where FS is the full-scale receiver voltage and G is the overall acquisition gain, yielding the sEMG amplitude in mV at the electrode.

Each input audio block is then reduced to a single scalar by computing the root-mean-square (RMS) directly inside the callback, as

$$\mathrm{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}, \tag{3.8}$$

where x_i, with $i = 1, \ldots, N$, are the N samples of the current block, already expressed in the original sEMG voltage. This block RMS is stored and exposed to the session controller through a thread-safe getter method, and is the quantity compared against an operator-defined activation threshold, adjustable in 0.1 mV steps between 0.1 and 1.0 mV, to trigger the green-to-yellow stage transition (Subsection 3.3.5). The RMS is thus recomputed independently for each incoming block, rather than over a sliding window with overlap or subject to smoothing, debouncing, or a minimum-hold-time criterion.

Similar to the stimulation stream, buffer overflow events are counted but do not interrupt acquisition, and the acquired signal is preserved without blocking the real-time thread by buffering captured samples and flushing them outside the audio callback. A separate, unbounded buffer accumulates samples between GUI redraws. It is drained on each read, letting the acquired signal be displayed in the operator window (Subsection 3.3.6) without under- or over-sampling the plot relative to the true acquisition rate.

3.3.5 Session Control Logic

The desired coordination between stimulation output and EMG acquisition required an explicit representation of the session's progress through its constituent phases, along with well-defined rules for when stimulation and acquisition are each engaged. This was achieved with a finite-state machine with four stages, the first consisting of a 10 second initial preparation stage. Second, a voluntary contraction stage under EMG monitoring is held until the activation threshold defined

by the clinician is crossed, at which point the session transitions to a maximal-contraction stage of 5 seconds. During this stage, interferential currents are delivered to the knee as positive reinforcement, and EMG acquisition is stopped so the electrodes don't pick up stimulation waves. Finally, the cycle reaches a rest stage of 4 seconds, then repeats from the contraction stage for the remainder of the session. Before each new contraction, the RMS value is reset, requiring the patient to reach the activation threshold again.

The controller itself contains no timing mechanism: its `tick_second()` and `check_emg_threshold()` methods are purely reactive and advance the session only when invoked. This design is deliberately clock-driven rather than self-timed, with the invocation cadence supplied externally to the controller by two Qt timers owned by the patient-facing window (Subsection 3.3.6): a 1 Hz QTimer calls `tick_second()` to advance stage countdowns and the total elapsed treatment time, while a separate 10 Hz QTimer calls `check_emg_threshold()` to poll the latest EMG RMS during the contraction stage. This inversion of control decouples the state machine from the Qt event loop, but requires the patient-facing window to invoke both methods correctly and at the expected rates.

3.3.6 Graphical User Interface

As introduced above, the system is split into an operator-facing window and a patient-facing window, matching the different needs and access levels of the two users. The operator must be able to configure and monitor every aspect of the stimulation and acquisition chain, whilst the patient must receive only the minimal feedback necessary to guide the exercise, without exposure to or control over the underlying stimulation parameters.

The clinician window exposes all stimulation parameters, namely carrier frequency, beat frequency, IFC mode, EMG activation threshold, and treatment duration, and renders a live preview of the resulting interference waveform before a session begins. It also hosts the volume calibration control for the dedicated routine described in Subsection 3.3.2. A periodic 300 ms poll of the system volume keeps the operator window's predicted current display updated throughout the session, including during any drift from the calibrated level.

In turn, the patient-facing window displays a single, large, color-coded status indicator corresponding to the current session stage. The timers introduced in Subsection 3.3.5 are instantiated in this window and started when the session begins. This window also relays the session state, that is, elapsed time, latest RMS, and raw sEMG samples for plotting, back to the operator-facing window through the Qt signal/slot mechanism. The communication between windows is thus mediated by explicit signals instead of a shared mutable state, and all cross-thread exchanges between the GUI and the audio callback are handled by the locking schemes described in Subsections 3.3.3 and 3.3.4.

Chapter 4

Results and Discussion

Before analyzing the performance of the individual subsystems and the software interface, it is helpful to visualize the complete experimental setup. Figure 4.1 shows the physical realization of the primary hardware modules, consisting of the IFT stimulation circuit (in the upper breadboard) and the sEMG acquisition module (in the lower breadboard), as well as the step-up transformer (inside the red tin box). The full integrated system, demonstrating how these custom modules connect to the standard peripherals and the control software, is depicted in Figure 4.2.

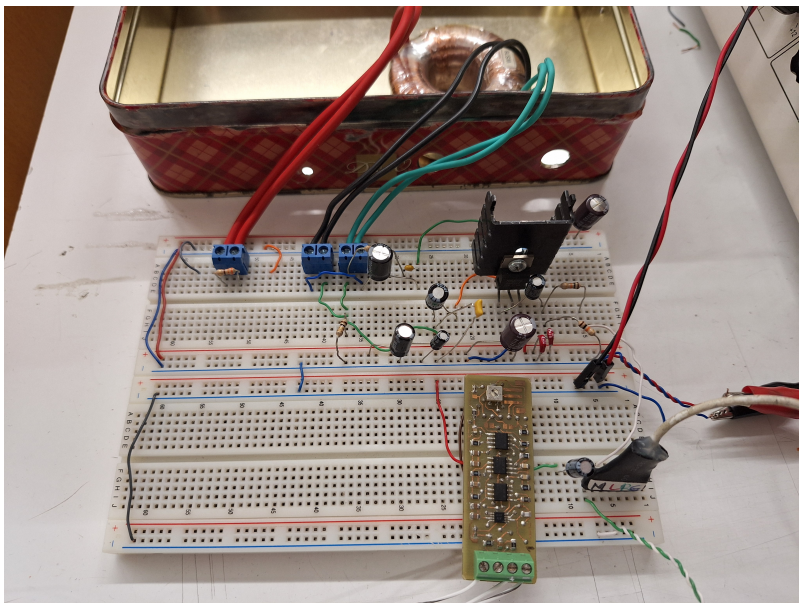


Figure 4.1: Bird's-eye view of the developed hardware modules.

4.1 IFT Stimulation Circuit

The primary voltage and secondary current obtained for each volume setting are presented in Figure 4.3. In both cases, the minimum values were plotted against negative volume percentages to assess the symmetry between the crests and valleys of the measured quantities.

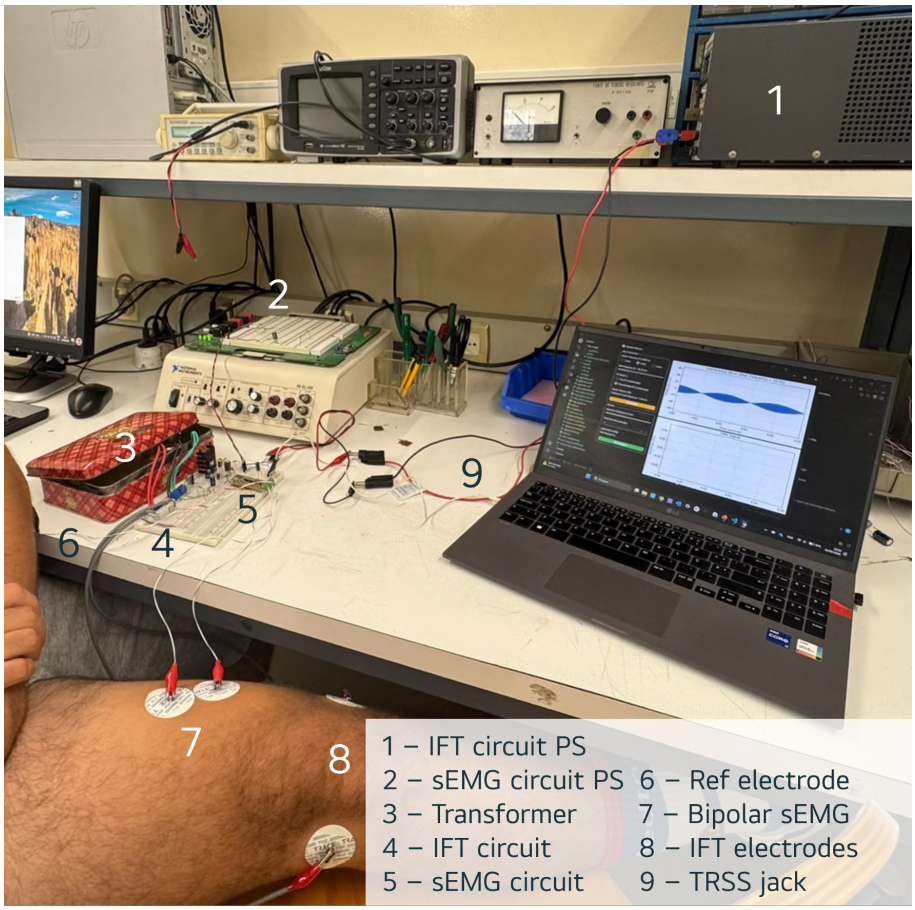


Figure 4.2: Experimental setup of the integrated system, with a numbered key that legends all its important components.

The pair of plots in Figure 4.3 exhibits an approximately sigmoid shape, compressed near the origin with an increased slope through the mid-range volume settings, followed by a gradual flattening as the output approaches saturation near the 100% extremes. In particular, the primary voltage trace shows a plateau near the 60% volume setting, appearing at both the maximum and minimum values. The same is not verified in the secondary current graph, which shows a small, sudden decrease in the current's absolute value around the 70% volume setting. This was likely an outlier in the measurements. The similar behavior across the volume sweep seen in both traces is naturally consistent with the two quantities being related through the transformer's turns ratio (Equation 3.2).

The non-linearity of the transfer curve is mainly due to the logarithmic scale typically used in the audio taper effect used in electronic audio devices.

Figure 4.4 isolates the maximum secondary current, $I_{s,\max}$, at each volume setting and fits it with a third-order polynomial:

$$I_{s,\max}(x) = -1.01 \times 10^{-4} x^3 + 1.64 \times 10^{-2} x^2 - 2.86 \times 10^{-2} x + 2.29 \times 10^{-1}, \tag{4.1}$$

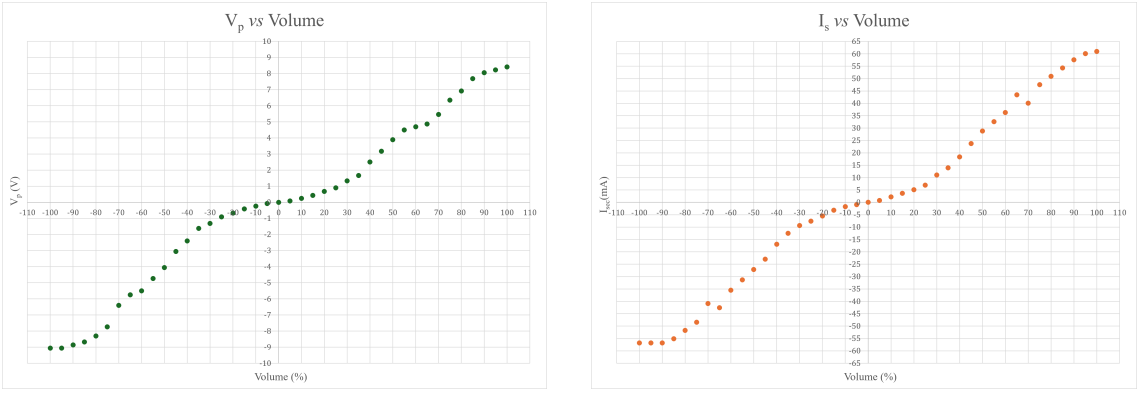


Figure 4.3: Primary voltage (left) and secondary current (right) versus volume setting.

where x is the volume setting in percent and $I_{s,\max}$ is expressed in mA. This equation closely tracks the measured data across the full 0-100% range, with a coefficient of determination of $R^2 = 0.9961$, supporting the choice of a cubic model over the sweep. Equation 4.1 is the same expression implemented in `volume_lock_calibration.py` (Subsection 3.3.2) to convert the operating system's volume percentage into the Predicted Current Maximum indicator shown on the operator-facing window (Subsection 3.3.6). Consequently, the accuracy of this software-reported estimate is bounded by the goodness of fit demonstrated here, and any future change to the transformer, amplifier gain, or sensing resistor would require this bench characterization and cubic fit to be repeated before the calibration routine could be trusted again.

Across the full volume range, the secondary current does not exceed approximately 60 mA peak, as shown in Figure 4.3. However, this value was measured using the 330 Ω surrogate load described in Subsection 3.2.1, a tissue-conductivity-based estimate that accounts only for bulk conduction and therefore omits the electrode-skin interface. Since the current-feedback loop regulates primary rather than secondary current (Section 3.2), the secondary current actually delivered depends on the impedance presented to the secondary winding and, following the same reasoning used to bound the 330 Ω estimate, an omitted series impedance can only raise the effective load above 330 Ω, not lower it. Consequently, the 60 mA peak figure should be read an estimated upper bound, obtained under a load likely somewhat lower than that presented by an actual patient.

4.2 Software

Figure 4.5 shows the operator-facing window of the developed application, which also serves as the main hub of the software. The clinician can adjust the stimulation parameters in a dedicated "Wave Parameters" box in the upper-left corner of the window. By default, the system selects a carrier frequency of 4 kHz, although two additional options are available: 2 kHz and 6 kHz. The beat frequency can be set to any value between 0 and 250 Hz through a dedicated slider control, with the system preset to the conventional 50 Hz value by design. Any change to either parameter is immediately reflected in the interference waveform drawn in the top plot, which serves as an

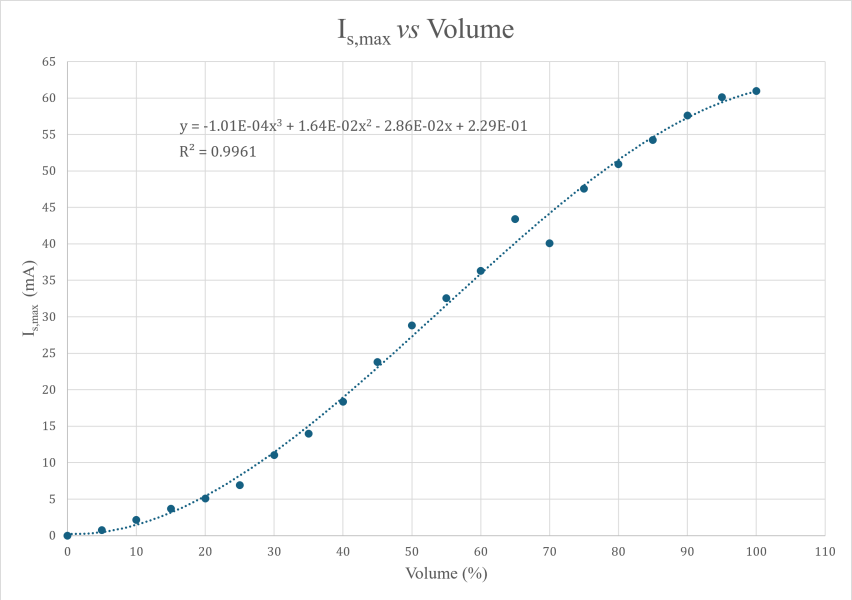


Figure 4.4: Maximum secondary current versus volume setting, with polynomial cubic fit.

interactive preview of the expected current delivered to the subject. Beyond this preview role, it also serves a didactic purpose, allowing the clinician to observe directly how each parameter change affects the resulting waveform.

Below the frequency selection controls is the IFC Mode sub-group, where the operator can choose between True IFC and Premodulated IFC mode (the latter being the default). This selection determines the nature of the signal routed through the sound card: in True IFC mode, the left channel outputs a sine wave at the carrier frequency (f_1). In contrast, the right channel outputs a second sinusoid whose frequency corresponds to the sum of the carrier and beat frequencies. The preview plot is unaffected by this choice, as it represents the expected interference pattern regardless of whether it is computed numerically or generated physically within the target tissue. However, the stimulation hardware currently accepts only a single input signal, so True IFC mode is not yet operational with the current circuit implementation.

Beneath the IFC selection controls, the System Volume indicator periodically polls the Windows volume mixer to display the actual output level, as detailed in Subsection 3.3.2. This widget is complemented by the Predicted Current Maximum indicator, which reports a current estimate for the selected volume percentage. This value is derived from the cubic fit that relates volume setting to secondary current, established during the bench characterization presented in [REF to calibration curve results]. Accordingly, the preview plot also updates dynamically with changes in volume percentage, since this parameter governs the amplitude of the stimulation waveform.

Finally, the operator sees an orange Set Volume button, which triggers the amplitude calibration routine described in Subsection 3.3.2 when pressed. In that case, the operating system volume is reset to zero, allowing the clinician to raise it gradually until the patient reports a stimulation sensation that is strong yet comfortable. Once this level is reached, the clinician terminates the calibration routine by pressing the same button, which changes to red during the process and displays

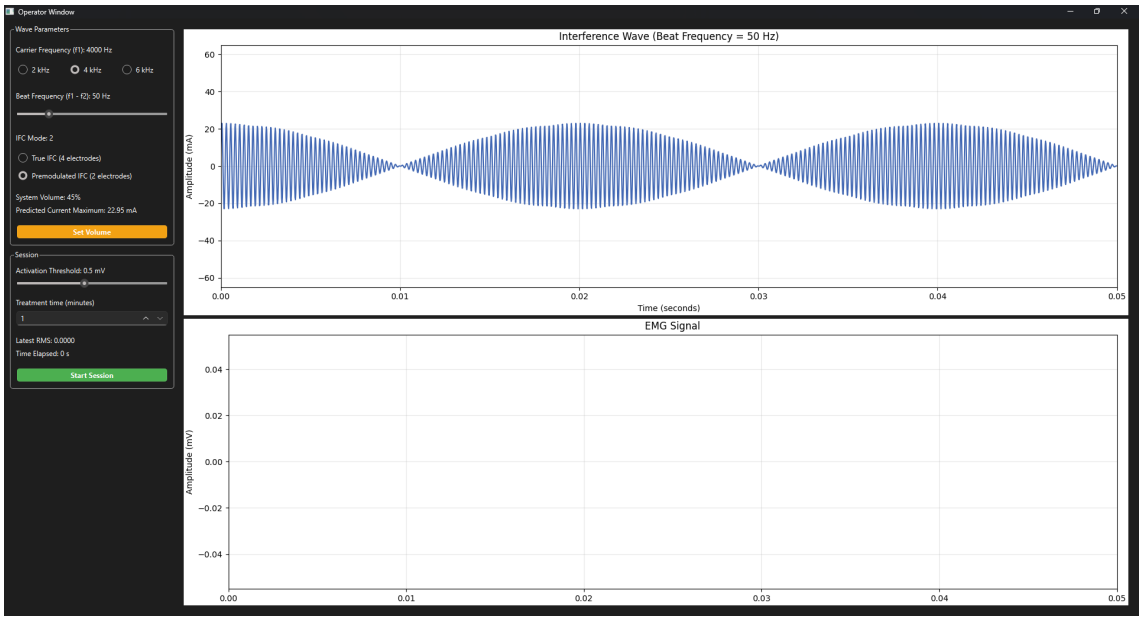


Figure 4.5: Operator-facing window.

an "End Volume Setting" label.

This calibration, and the Predicted Current Maximum indicator derived from it, is established once before the session and is not checked again while stimulation is being delivered: no closed-loop measurement of the delivered current feeds back into the software, so the system relies entirely on this pre-session calibration rather than on real-time verification that the indicated current is actually reaching the patient. Consequently, any change in delivered current during a session, whether from electrode-contact drift, cable or connector issues, or the volume and Wave Parameters drift discussed further below, would not be detected by the software and would not be reflected in the Predicted Current Maximum indicator, which continues to report the value corresponding to the calibrated state rather than the present one. Closing this loop would require an independent current-sensing measurement on the secondary side of the stimulation circuit, fed back into the software in real time, which is not part of the present implementation.

Apart from the "Wave Parameters" box, the interface includes a separate Session container, featuring an "Activation Threshold" slider, preset to 0.5 mV, which defines the RMS level the patient must exceed during the session to trigger a stage transition. Under this control is a Treatment time Spin Box that lets the operator define the session duration through the increment arrows or by entering an integer value directly. Two additional indicators complete this container, respectively displaying the most recently computed EMG RMS value and the total elapsed session time.

Once the clinician has configured both the wave and session parameters, they initiate the treatment session by clicking the "Start Session" button (in green). This action immediately opens the patient-facing window in the Preparation phase, shown in Figure 4.6. This window is intended to be dragged to a second monitor positioned in front of the patient, so that the patient's attention remains focused exclusively on the color-coded stage indicators. Such a configuration effectively

extends the two-window software architecture into the physical setup, leaving the operator free to monitor the acquired EMG signal, the latest RMS value, and the total elapsed time.

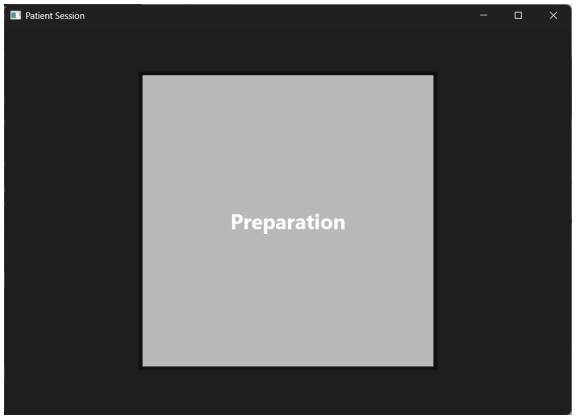


Figure 4.6: Patient window in the preparation phase.

Ten seconds after the onset of the Preparation phase, the colored square in Figure 4.6 turns green and displays the label "Contract," instructing the patient to contract the quadriceps, in accordance with the state-machine logic detailed in Subsection 3.3.5. During this stage, the clinician should encourage the patient to progressively increase the contraction intensity while monitoring the corresponding RMS value on the operator window. Once the block RMS exceeds the predefined activation threshold, the feedback message updates to "Maximum" within a yellow square, prompting the patient to reach maximal voluntary contraction and sustain it for the duration specified Subsection 3.3.5. This indicator communicates which stage is active but not progress within it: no countdown or other quantitative cue informs the patient of how much of the required hold time remains during the Maximum stage, or of the Preparation and Rest intervals, leaving the patient to rely on the operator's verbal guidance or on their own estimation of elapsed time.

The system then transitions to the "Rest" phase, during which the patient receives no stimulation for the corresponding fixed interval, before the cycle resumes at the contraction phase and repeats until the treatment duration defined by the clinician has elapsed. Figure 4.7 presents an overview of the session pipeline from the patient's perspective, mirroring the four-stage cycle (Preparation, Contraction, Maximum, Rest) formalized as a state machine in Subsection 3.3.5.

The physical execution of this software-driven transition was validated at the hardware level. As demonstrated by the oscilloscope capture in Figure 4.8, the system successfully synchronizes the acquisition and stimulation modules, ensuring that the high-frequency IFT stimulation is exclusively delivered when the real-time sEMG RMS value surpasses the selected activation threshold.

This Contraction-to-Maximum transition rests on a block-wise RMS estimate that, as noted in Subsection 3.3.4, is recomputed independently for each incoming block rather than over a smoothed or overlapping window, and is not subject to de-bouncing or a minimum-hold-time criterion before a stage transition is accepted. No faulty or missed transitions attributable to this were observed during the short, approximately one-minute self-administered sessions conducted by the author during development, in which the activation threshold was consistently crossed once

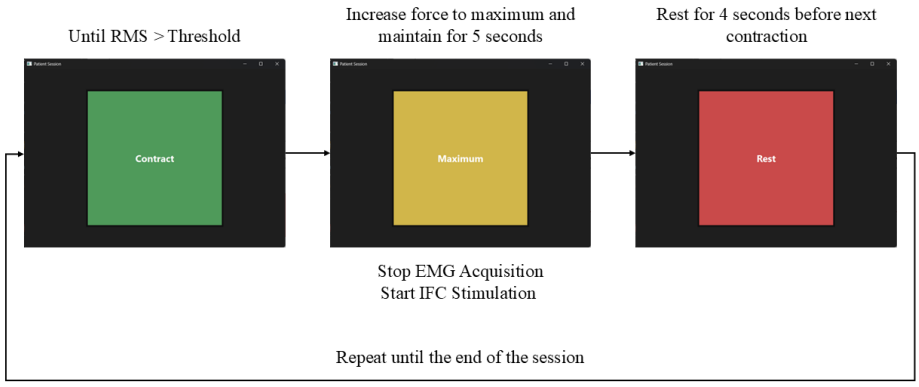


Figure 4.7: Session pipeline from the patient's point of view.

the intended contraction intensity was reached. These sessions, however, were neither performed on a patient population nor extended in duration or repetition count, so they do not constitute a systematic test of this failure mode. The block-wise estimate remains, by construction, more sensitive to an isolated noise transient or motion artifact than a windowed alternative, and this sensitivity was not stress-tested under conditions specifically designed to provoke it. Windowed smoothing of the RMS estimate and a minimum-hold-time criterion before accepting a threshold crossing are therefore identified as future-work directions worth testing, rather than as corrections to an observed failure.

Concurrently, the operator-facing window displays the acquired EMG signal, the real-time block RMS, and a chronometer indicating the total elapsed session time, as shown in Figure 4.9. Once the defined treatment time elapses, the patient window closes automatically, stopping stimulation, acquisition, and time tracking. Should any issue arise during the session, the operator may also terminate it manually via the "End Session" button, without needing to wait for the session's natural conclusion. The acquired EMG and stimulation signals are subsequently stored in CSV files for later visualization and analysis.

This storage, however, is limited to the raw sEMG signal: the derived per-block RMS trace displayed in real time and used to drive the stage-transition logic of Subsection 3.3.5 is not written to file. Whether this trace could be reconstructed exactly from the stored raw signal alone, which would require the block boundaries used during acquisition to be recoverable from the saved data, has not been established, so its absence should be treated as a loss of directly usable information rather than a redundancy. As the RMS is already computed once per block during the session, appending it to the existing CSV output would add negligible overhead while preserving the exact signal that governed the patient's session, supporting post hoc verification of stage-transition timing and cross-session comparison.

Separately, the underflow and overflow counters maintained by the stimulation and acquisition

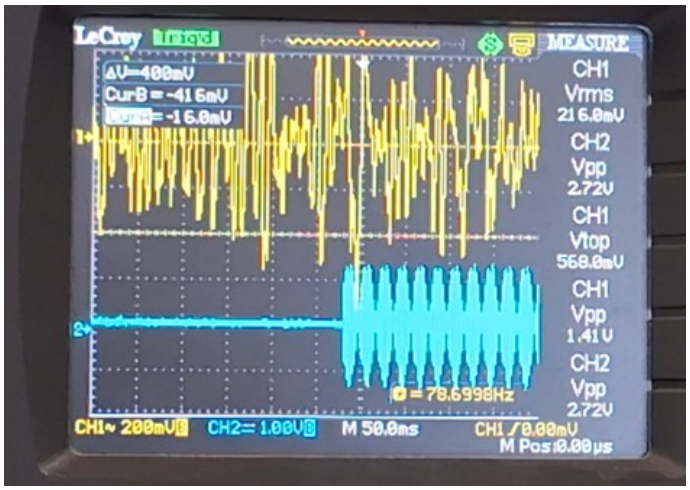


Figure 4.8: Oscilloscope trace validating the closed-loop control logic. The IFT stimulation is triggered precisely when the acquired sEMG RMS signal exceeds the predefined activation threshold (200 mV in this test case).

streams (Subsections 3.3.3 and 3.3.4) are not persisted or surfaced anywhere in the current implementation: although both streams already track these events internally, favoring uninterrupted delivery and acquisition over interrupting a session to report them, the resulting counts are discarded rather than logged. Consequently, neither the operator during a session nor the analyst reviewing the stored CSV files afterward can determine whether a timing dropout occurred, or when. A future revision could log these counters, for instance by appending them to the session's CSV output or a dedicated summary file, or expose them as an operator-facing indicator, providing a basis for diagnosing timing artifacts post hoc rather than relying on the operator noticing symptoms in real time.

Amplitude calibration and stimulation delivery are mutually exclusive processes and cannot occur simultaneously. Apart from this design choice, no equivalent interlock exists for the Wave Parameters controls for carrier frequency, beat frequency, and IFC mode, which remain fully live and enabled during an active treatment session and the calibration routine in Subsection 3.3.2. Any change made through them is applied immediately to the ongoing stimulation output rather than being deferred or rejected. This is the same live-update path that makes the parameter preview useful before a session begins. Still, during calibration or delivery, it allows the clinician to inadvertently alter the interference pattern actually being delivered to the patient. Disabling the Wave Parameters controls, or requiring an explicit confirmation before applying changes while a session or calibration is in progress, is identified as a safety-relevant improvement for future revisions of the interface.

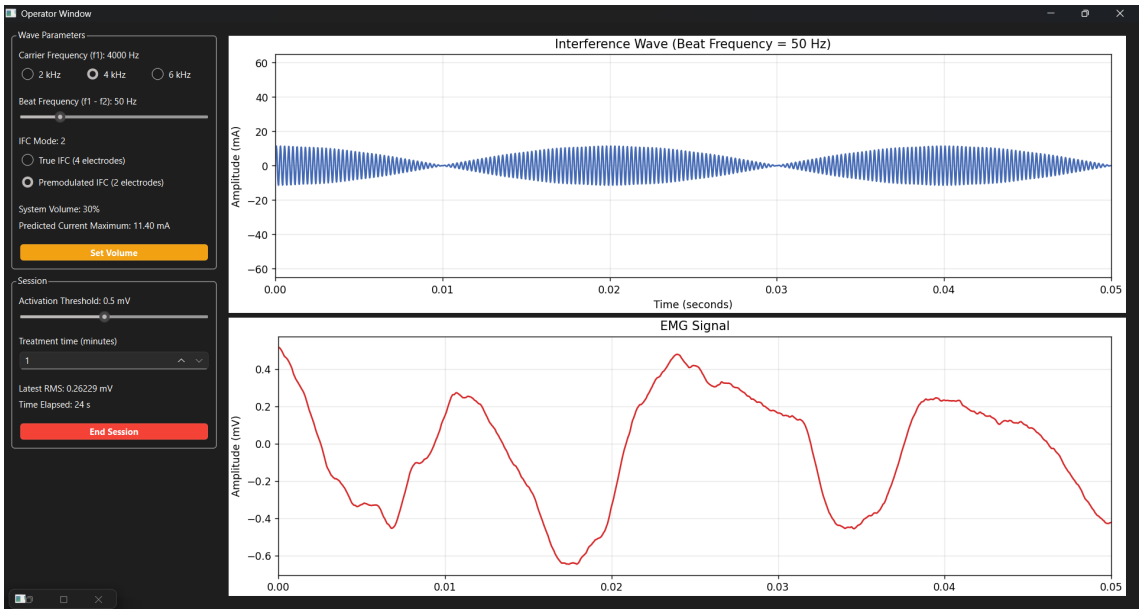


Figure 4.9: Operator window during the session, showing the acquired EMG signal, latest calculated RMS, and elapsed time.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This dissertation set out to design, implement and characterize an EMG-triggered, interferential-therapy-based functional electrical stimulation system for post-ACL-reconstruction rehabilitation, in which stimulation of the knee is made contingent on the patient's own volitional quadriceps activation. A single-supply sEMG acquisition chain built around an INA326 instrumentation amplifier, with DC-offset cancellation, active filtering, right-leg drive and a mid-supply reference (Section 3.1), achieved a bandpass response between approximately 10 and 500 Hz and an input-referred noise of 0.72 μV_{RMS} over 20–500 Hz (Equation 2.10), and reliably drove the session logic from voluntary contractions in practice.

Secondly, a stimulation stage built around a TDA2002 operated as a transconductance amplifier, driving a TST 10/020 transformer in step-up configuration (Section 3.2), was characterized at a 4 kHz carrier against a 330 Ω surrogate load. From the bench tests, a maximum secondary current cubic dependence on volume setting was derived, with $R^2 = 0.9961$ (Equation 4.1), the same expression used in software to report the Predicted Current Maximum. Its value not exceed approximately 60 mA peak across the sweep, though these secondary-side figures were derived through the turns ratio rather than measured directly, and the sweep covered a single carrier and a single load.

Finally, a modular Python application synthesizes stimulation waveforms and digitizes sEMG through the sound card, preserving phase continuity across the streaming path and using a four-stage state machine (Preparation, Contraction, Maximum, Rest) to make stimulation conditional on the block RMS (Equation 3.8) exceeding an operator-defined threshold, with separate clinician and patient-facing windows and support for both true and premodulated IFC synthesis. The three subsystems were integrated and exercised end to end, with voluntary contraction reliably triggering interferential stimulation within a timed session, demonstrating the functional feasibility of this closed loop, which was the central technical claim of this work.

However, testing was limited to short, self-administered sessions, with no patients involved, so the underlying hypothesis that synchronizing stimulation with volitional effort supports neu-

rological re-integration and graft regeneration remains untested and should be read as motivating rationale rather than a finding. Two further limitations should be stated plainly: the system is open-loop with respect to delivered current, with no feedback on the amplitude actually reaching the patient, and the true IFC mode, though implemented in software, cannot currently be delivered because the stimulation circuit accepts only a single input channel, so all results reported here concern premodulated IFC.

Overall, this dissertation delivers a functioning, characterized bench prototype together with an explicit account of where the evidence stops: a research platform rather than a therapeutic device.

5.2 Future Work

The most immediate step is enabling genuine interferential currents, which requires replacing the single-channel TDA2002 with a dual amplifier such as the TDA2005 [94], feeding a quadripolar electrode arrangement through two independent secondary circuits, each with its own transformer, sensing resistor and feedback loop, and re-characterizing the calibration of Equation 4.1 per channel; whether the resulting interference pattern actually forms at the intended depth would still need confirmation through a finite-element knee model or phantom measurements.

A related extension is a multi-channel system stimulating the knee joint and the quadriceps muscle complex independently, for which a single dual amplifier in stereo configuration is the simplest solution if only one joint and one muscle channel are needed, while true IFC combined with an independent muscle channel would require three outputs, most economically obtained from two dual-amplifier devices. Beyond a certain channel count, however, the sound card-based architecture that keeps the present system inexpensive and portable would need to be reconsidered.

The modular separation of synthesis from streaming also makes it straightforward to add a burst-modulated modality such as the Kotz (Russian) current [95] for muscle stimulation, though it would need its own bench characterization and conversion routine. A more foundational need is an isolated secondary-side current sensor feeding back into the software, which would validate the derived quantities reported here against direct measurement, replace the set-once calibration with continuous verification, and provide the hard safety limit that any automated control would require. The bench characterization itself could also be extended across the physiological load range of roughly 200–600 Ω implied by Equation 3.6 and across the 2 and 6 kHz carriers already offered in software but not yet characterized.

Several contained software improvements would also be worthwhile, including testing and evaluating different RMS estimation strategies to reduce sensitivity to noise transients, logging of the RMS trace and stream underflow/overflow counters for post hoc diagnosis, safeguards against changing wave parameters mid-session, a progress indicator for the patient-facing window, and replacing the operating-system volume mixer with a dedicated amplitude control for genuine cross-platform operation. A more ambitious direction would be making the stimulation responsive to muscular state in real time. Spectral fatigue monitoring from the sEMG median frequency [96]

could serve as a initial approach, as it doesn't require hardware changes. This could in principle inform an adaptive scheme, potentially based on reinforcement learning, that adjusts stimulation parameters online rather than holding them fixed for a session. Neither a concrete roadmap nor the shape of such a system has been worked out at this stage, and both would need to be developed before the idea moves beyond a research direction worth pursuing.

Ultimately, validation with patients is the precondition for any clinical claim about this system, and would also be the setting in which the software improvements proposed above could be evaluated in practice. The distance between the present platform and a clinically useful device is therefore traversed, in order, by closing the current-measurement loop, enabling genuine interferential delivery, extending the bench characterization, satisfying the applicable safety standards, and only then evaluating the therapeutic hypothesis in patients.

Appendix A

Appendix

signal_generator.py

```
"""
signal_generator.py

This script contains pure waveform helpers for one-shot UI plotting (carrier_wave)
and audio stream (carrier_wave_stateful).
None of these functions touch any widget state, they only depend on their
explicit arguments.
"""
import numpy as np

def carrier_wave(
    freq: float, time: np.ndarray, phase: float = 0.0, ampl: float = 0.50
) -> np.ndarray:
    """
    Returns a sinusoidal carrier wave for a given frequency, time array, and
    optional starting phase. The function is pure and is used for plotting.

    Parameters
    ----------
    freq : float
        The carrier frequency.
    time : ndarray
        The time array.
    ampl : float
        The target amplitude for the carrier wave.
    phase : float (default = 0.0)
        The phase at which the wave should start

    Returns
    -------
    wave : ndarray
        The carrier wave, same shape as time.
```

```
    """
    return ampl * np.sin(2 * np.pi * freq * time + phase)

def carrier_wave_stateful(
    freq: float, time: np.ndarray, start_phase: float
) -> tuple[np.ndarray, float]:
    """
    Returns one waveform block and the wrapped phase to reuse in the next
    callback. This keeps audio continuous across callback boundaries and
    avoids clicks from phase resets. Otherwise, these sudden changes would
    translate into unwanted transient current spikes through the electrodes.

    Parameters
    ----------
    freq : float
        The carrier frequency.
    time : ndarray
        The time array.
    start_phase : float
        The phase at which the waveform block should start.

    Returns
    -------
    wave : ndarray
        The waveform block, same shape as time.
    next_phase : float
        The phase to pass as start_phase in the following call, wrapped to
        [0, 2*pi).
    """
    # Create instantaneous phase at every block sample
    phase_array = 2 * np.pi * freq * time + start_phase

    # Generate new sinewave with updated phase
    wave = np.sin(phase_array)

    # Get next phase value from last sample in block
    # This preserves continuity across sounddevice callback blocks
    next_phase = phase_array[-1] + 2 * np.pi * freq * (time[1] - time[0])

    return wave, next_phase % (2 * np.pi)
```

volume_lock_calibration.py

```
"""
volume_lock_calibration.py

This module contains the Windows output-volume helpers and the calibration
```

```
curve that converts the current OS volume percentage into the expected
patient current.
"""
import sys
from pycaw.pycaw import AudioUtilities

def volume_to_predicted_current_ma(volume_percent: float) -> float:
    """
    Returns the expected maximum current in the patient in mA, estimated from
    the Windows output volume percentage via an oscilloscope-fitted cubic
    curve (Isec,max vs Volume %).

    Parameters
    ----------
    volume_percent : float
        The Windows master output volume, in percent (0-100).

    Returns
    -------
    predicted_current_ma : float
        The predicted maximum current in mA. If negative, 0 is returned.
    """
    x = volume_percent
    predicted_current_ma = (
        -1.01e-04 * x**3 + 1.64e-02 * x**2 - 2.86e-02 * x + 2.29e-01
    )

    return max(0.0, predicted_current_ma)


def get_output_volume_scalar() -> float:
    """Reads the current Windows master volume (0.0 to 1.0).

    Returns:
        float: The master volume scalar, from 0.0 (muted) to 1.0 (full volume).
    """
    if sys.platform != "win32":
        return 1.0  # non-Windows: assume full volume, no mixer to query
    try:
        device = AudioUtilities.GetSpeakers()
        if device is not None:
            return device.EndpointVolume.GetMasterVolumeLevelScalar()
    except Exception as exc:
        # pycaw/COM calls can fail for many reasons (no device, driver issue);
        # don't crash calibration
        print(f"Could not read output volume: {exc}")
    return 1.0

def set_output_volume_scalar(target_scalar: float) -> None:
```

```
    """Sets the Windows master volume to a specific scalar (0.0 to 1.0).

    Parameters:
        target_scalar (float): The desired master volume, from 0.0 (muted) to
            1.0 (full volume).

    Returns:
        None
    """
    if sys.platform != "win32":
        return  # no-op on non-Windows platforms
    try:
        device = AudioUtilities.GetSpeakers()
        if device is not None:
            device.EndpointVolume.SetMasterVolumeLevelScalar(target_scalar, None)
    except Exception as exc:
        # same rationale as above: fail silently rather than blocking calibration
        print(f"Could not set output volume to {target_scalar}: {exc}")
```

signal_stream.py

```
"""
signal_stream.py

This script contains the dedicated SignalStream class, responsible for
setting stream parameters, starting and stopping output streams, and
filling the output buffer continuously without delay and discontinuities.
"""

import threading
import numpy as np
import sounddevice as sd
from signal_generator import carrier_wave_stateful

class SignalStream:
    """
    Manages a real-time audio output stream for interferential current stimulation.

    Owns the sounddevice.OutputStream, the phase state of both carriers,
    the thread-safe parameter set shared between the GUI thread (writer)
    and the PortAudio callback thread (reader). Only handles threading,
    lifecycle, and buffer writes.

    Parameters
    ----------
    samplerate : int, optional
        Audio sample rate in Hz (default 48000).
```

```
    channels : int, optional
        Number of hardware output channels (default 2).

    Attributes
    ----------
    fs : int
        Sample rate in Hz.
    channels : int
        Number of output channels.
    f1, f2 : float
        Current carrier and modulator frequencies in Hz, set via set_parameters.
    beat : float
        Current beat frequency in Hz.
    ifc_mode : int
        2 for premodulated (2-electrode) IFC, 4 for true (4-electrode) IFC.
    phase1, phase2 : float
        Persisted phase state for each carrier, updated only by the audio
        callback thread to maintain phase continuity across blocks.
    stream : sounddevice.OutputStream or None
        The active output stream, or None when stopped.
    underflow_count : int
        Number of buffer underflows (glitches), updated only by the audio callback.

    Methods
    -------
    set_parameters(f1, f2, beat, ifc_mode)
        Update stimulation parameters from the GUI thread.
    start()
        Open and start the output stream.
    stop()
        Stop and close the output stream.
    configure_ifc_logging(path, enabled, reset_file)
        Configure IFC signal output logging.
    flush_ifc_log_buffer(max_blocks)
        Flush buffered IFC samples to disk.
    """
    def __init__(self, samplerate: int = 48000, channels : int = 2) -> None:
        """
        Initializes the SignalStream with streaming sample rate and
        number of output channels.

        Parameters
        ----------
        samplerate : int, optional
            Audio sample rate in Hz (default 48000).
        channels : int, optional
            Number of hardware output channels (default 2).
        """
        self.fs = samplerate
```

```
        # channels = 2 -> standard laptop headphone jack is a stereo output (L+R)
        self.channels = channels
        self._param_lock = threading.Lock()

        # Default parameters shared between GUI thread (writer) and
        # audio thread (reader)
        self.f1 = 4000.0
        self.f2 = 4050.0
        self.beat = 50.0
        self.ifc_mode = 2  # 2 = premodulated, 4 = true IFC
        self.phase1 = 0.0
        self.phase2 = 0.0

        self.stream = None

        # incremented by the audio thread, read by the GUI thread
        self.underflow_count = 0
        self._log_lock = threading.Lock()
        self._log_enabled = False
        self._log_path = "ifc_signal.csv"
        self._log_buffer = []

    def set_parameters(self, f1: float, f2: float, beat: float,
                       ifc_mode: int) -> None:
        """
        Update stimulation parameters. Called from the GUI thread when
        sliders/buttons change; safely synchronized with the audio
        callback thread via a lock.

        phase1/phase2 are only ever read/written by the audio callback
        thread, never by the GUI thread, so no lock is needed for them
        specifically.

        Parameters
        ----------
        f1 : float
            Carrier frequency in Hz.
        f2 : float
            Modulator frequency in Hz.
        beat : float
            Beat frequency in Hz (f2 - f1).
        ifc_mode : int
            2 for premodulated (2-electrode) IFC, 4 for true (4-electrode) IFC.
        """
        # with statement to acquire threading object and automatically
        # release it once all GUI elements are assigned
        with self._param_lock:
            self.f1 = f1
            self.f2 = f2
```

```
            self.beat = beat
            self.ifc_mode = ifc_mode

    def _callback(
        self, outdata: np.ndarray, frames: int, time_info, status
    ) -> None:
        """
        sounddevice output callback. Called repeatedly by PortAudio on
        its own real-time thread to fill one audio block.

        Must write into outdata in-place and must not block, allocate
        unpredictably, or perform I/O, since delays here cause audio
        dropouts and unpredictable gaps in stimulation output.

        Parameters
        ----------
        outdata : ndarray
            Output buffer to fill in-place, shape (frames, channels).
        frames : int
            Number of samples requested for this block.
        time_info : CData
            PortAudio timing info (unused).
        status : CallbackFlags
            Set if an underflow/overflow occurred. No abrupt stoppage
            for now, neither signalling to GUI.
        """
        if status:
            self.underflow_count += 1  # non-blocking call

        with self._param_lock:
            f1, f2, beat, mode = self.f1, self.f2, self.beat, self.ifc_mode

        block_time = np.arange(frames) / self.fs

        # Phase state now lives inside the class
        # next_phase persists between calls, giving click-free
        # continuous output across buffer boundaries
        wave1, self.phase1 = carrier_wave_stateful(
            freq=f1, time=block_time, start_phase=self.phase1
            )
        wave2, self.phase2 = carrier_wave_stateful(
            freq=f2, time=block_time, start_phase=self.phase2
            )

        n_ch = outdata.shape[1]
        if mode == 4:
            # if true IFC mode is activated, stream each carrier to a
            # headphone channel
            # sounddevice requires in-place buffer writes
```

```
174             outdata[:, 0] = wave1
175             if n_ch > 1:
176                 outdata[:, 1] = wave2
177         else:
178             # premodulated IFC is activated, sum carriers to get the
179             # premodulated wave
180             interference_wave = wave1 + wave2
181             outdata[:, 0] = interference_wave / 2.0 # normalize wave to +-1
182             if n_ch > 1:
183                 outdata[:, 1] = 0.0 # set right channel to zeros (silence)
184 
185         if self._log_enabled:
186             with self._log_lock:
187                 # .copy() required: outdata is a reusable buffer
188                 # PortAudio overwrites next call
189                 self._log_buffer.append(outdata[:, 0].copy())
190 
191     def start(self) -> None:
192         """
193         Opens and starts the sounddevice.OutputStream if not already running.
194 
195         Guarded against re-entrancy by the 'None' argument: if a stream
196         is already active, calling this again is a no-op, preventing
197         accidental creation of multiple overlapping streams from rapid
198         toggling of the start/stop button.
199         """
200         if self.stream is not None:
201             return
202 
203         self.stream = sd.OutputStream(
204             samplerate=self.fs,
205             channels=self.channels,
206             callback=self._callback,
207         )
208 
209         self.stream.start()
210 
211     def stop(self, flush_log: bool = True) -> None:
212         """
213         Stops and closes the active output stream, if any, and resets
214         the stream reference to None so a subsequent start() creates a
215         fresh stream.
216 
217         Parameters
218         ----------
219         flush_log : bool, optional
220             Whether to synchronously flush pending IFC samples after stopping.
221             Stage transitions can defer this work to avoid blocking the GUI thread.
222         """
```

```
        if self.stream is not None:
            self.stream.stop()
            self.stream.close()
            self.stream = None

        if flush_log:
            self.flush_ifc_log_buffer()

    def configure_ifc_logging(
        self, path: str = "ifc_signal.csv", enabled: bool = True,
        reset_file: bool = True,
    ) -> None:
        """
        Configure IFC signal output logging.

        Parameters
        ----------
        path : str, optional
            Destination file used to store logged IFC samples.
        enabled : bool, optional
            If True, callback-generated samples are buffered for later flush.
        reset_file : bool, optional
            If True and logging is enabled, truncate/create the file and write
            a header line.

        Notes
        -----
        This method only updates logging configuration and file header state.
        It does not flush or clear the in-memory log buffer.
        """
        self._log_path = path
        self._log_enabled = enabled

        if self._log_enabled and reset_file:
            with open(self._log_path, "w", encoding="utf-8") as out_file:
                out_file.write("IFC Signal\n")

    def flush_ifc_log_buffer(self, max_blocks: int = 128) -> int:
        """
        Flush buffered IFC samples to disk.

        Parameters
        ----------
        max_blocks : int, optional
            Maximum number of buffered blocks to write in this call. A block
            corresponds to one callback chunk appended to ``_log_buffer``.

        Returns
        -------
```

```
        int
            Number of individual samples written to the output file.

        Notes
        -----
        The buffer is shared with the audio callback thread. Access is protected
        with ``_log_lock`` to avoid race conditions while slicing/removing blocks.
        """
        if not self._log_enabled:
            return 0

        with self._log_lock:
            if not self._log_buffer:
                return 0

            # Drain only a bounded number of blocks to keep flush
            # latency predictable
            blocks_to_write = self._log_buffer[:max_blocks]
            del self._log_buffer[:max_blocks]

        samples = np.concatenate(blocks_to_write)
        with open(self._log_path, "a", encoding="utf-8") as out_file:
            for value in samples:
                out_file.write(f"{float(value)}\n")

        return int(samples.size)
```

emg_acquisition.py

```
"""
emg_acquisition.py

This script contains the dedicated EMGAcquisition class, responsible for capturing
the EMG input stream, converting samples to sEMG voltage, computing per-block RMS
for activation-threshold checks, and optionally logging samples to CSV.
"""

import sounddevice as sd
import numpy as np
import threading


RECEIVER_FULL_SCALE_VOLTS = 1.0
EMG_VOLTAGE_GAIN = 400.0
EMG_SAMPLE_SCALE_MV = (
    RECEIVER_FULL_SCALE_VOLTS * 1000.0 / EMG_VOLTAGE_GAIN
)
```

```
19
20  class EMGAcquisition:
21      """
22      Manages the real-time acquisition of surface electromyographic (sEMG)
23      signals from the patient's quadriceps.
24
25      Owns the sounddevice.InputStream and the thread-safe state read by the
26      GUI thread.
27
28      Parameters
29      ----------
30      samplerate : int, optional
31          Audio sample rate in Hz (default 44100) -> enough for the EMG signal
32      channels : int, optional
33          Number of hardware input channels (default 1).
34
35      Attributes
36      ----------
37      fs : int
38          Sample rate in Hz.
39      channels : int
40          Number of input channels.
41      latest_emg_signal : np.ndarray
42          Latest acquired signal block, expressed as original sEMG in mV.
43      """
44      def __init__(self, samplerate: int = 44100, channels : int = 1) -> None:
45          """
46          Initializes the EMGAcquisition with streaming sample rate and
47          number of input channels.
48
49          Parameters
50          ----------
51          samplerate : int, optional
52              Audio sample rate in Hz (default 44100) -> enough for the EMG signal
53          channels : int, optional
54              Number of hardware input channels (default 1).
55
56          Attributes
57          ----------
58          fs : int
59              Sample rate in Hz.
60          channels : int
61              Number of input channels.
62          latest_emg_signal : np.ndarray
63              Copy of the signal received in the latest input stream block.
64          last_rms : float
65              Computed RMS value of the latest signal block, or 0.0 after
66              capture starts and before the first block is received.
67          overflow_count : int
```

```
            Number of buffer overflows (#samples received > #samples
            requested), updated only by the audio callback.
        stream : sounddevice.InputStream or None
            The active input stream, or None when stopped.
        """
        self.fs = samplerate
        self.channels = channels
        self.stream = None
        self._lock = threading.Lock()
        self.latest_emg_signal = np.array([], dtype=np.float32)
        self._plot_buffer = []
        self.last_rms = 0.0
        self.overflow_count = 0
        self._log_enabled = False
        self._log_path = "emg_signal.csv"
        self._log_buffer = []

    def _callback(
            self, indata: np.ndarray, frames: int, time_info, status
        ) -> None:
            """
            sounddevice input callback. Called repeatedly on its own
            real-time thread to read one signal block, compute its RMS, and
            buffer it for optional logging.

            Parameters
            ----------
            indata : np.ndarray
                Input signal block, shape (frames, channels).
            frames : int
                Number of samples in this block (unused).
            time_info : CData
                PortAudio timing info (unused).
            status : CallbackFlags
                Set if an overflow occured. No abrupt stoppage for now,
                neither signalling to GUI.

            """
            if status:
                self.overflow_count += 1  # non-blocking call

            # sounddevice samples are normalized to +/-1 at the 2 Vpp
            # receiver input.
            # Convert receiver voltage back to the original sEMG voltage in mV.
            block = indata[:, 0].copy() * EMG_SAMPLE_SCALE_MV
            rms = float(np.sqrt(np.mean(block * block)))

            # acquire input thread and store latest signal block and computed RMS
            with self._lock:
```

```
                self.latest_emg_signal = block
                self.last_rms = rms
                self._plot_buffer.append(block)
                # save threading object in input log buffer
                if self._log_enabled:
                    self._log_buffer.append(block)


    def start(self) -> None:
        """
        Opens and starts sounddevice.InputStream if not already running.

        Guarded against re-entrancy by the 'None' argument: if a stream is already
        active, calling this again is a no-op, preventing accidental creation of
        multiple overlapping streams.

        The cached RMS value is reset to 0.0 before each new capture stream starts.
        """
        if self.stream is not None:
            return

        with self._lock:
            self.last_rms = 0.0

        self.stream = sd.InputStream(
            samplerate=self.fs,
            channels=self.channels,
            callback=self._callback,
        )

        self.stream.start()

    def stop(self) -> None:
        """
        Stops and closes the active input stream, if existing, and resets the
        ↪  stream
        reference to None so a subsequent start() creates a fresh stream.
        """
        if self.stream is not None:
            self.stream.stop()
            self.stream.close()
            self.stream = None

        self.flush_emg_log_buffer()

    def get_latest_signal(self) -> np.ndarray:
        """
        Thread-safe getter of the latest acquired EMG signal block.

```

```
        Warning
        -------
        Method still unused. Intended used for plotting acquired signal in
        the operator GUI.
        """
        with self._lock:
            return self.latest_emg_signal.copy()

    def get_last_rms(self) -> float:
        """
        Thread-safe getter of the latest computed rms; used in
        `session_controller.py` for frequently comparing the rms with the
        predefined activation threshold.
        """
        with self._lock:
            return self.last_rms

    def get_pending_plot_samples(self) -> np.ndarray:
        """Return all EMG samples captured since the previous plot update."""
        with self._lock:
            if not self._plot_buffer:
                return np.array([], dtype=np.float32)

            samples = np.concatenate(self._plot_buffer)
            self._plot_buffer.clear()
            return samples

    def configure_emg_logging(
        self, path: str = "emg_signal.csv", enabled: bool = True,
        reset_file: bool = True,
    ) -> None:
        """
        Configure EMG signal output logging.

        Parameters
        ----------
        path : str, optional
            Destination file used to store logged EMG samples.
        enabled : bool, optional
            If True, callback-generated samples are buffered for later flush.
        reset_file : bool, optional
            If True and logging is enabled, truncate/create the file and write
            a header line.

        Notes
        -----
        This method only updates logging configuration and file header state.
        It does not flush or clear the in-memory log buffer.
        """
```

```
        self._log_path = path
        self._log_enabled = enabled

        if self._log_enabled and reset_file:
            with open(self._log_path, "w", encoding="utf-8") as out_file:
                out_file.write("EMG Signal (mV)\n")

    def flush_emg_log_buffer(self, max_blocks: int = 128) -> int:
        """
        Flush buffered EMG samples to disk.

        Parameters
        ----------
        max_blocks : int, optional
            Maximum number of buffered blocks to write in this call. A block
            corresponds to one callback chunk appended to ``_log_buffer``.

        Returns
        -------
        int
            Number of individual samples written to the output file.

        Notes
        -----
        The buffer is shared with the audio callback thread. Access is protected
        with ``_lock`` to avoid race conditions while slicing/removing blocks.
        """
        if not self._log_enabled:
            return 0

        with self._lock:
            if not self._log_buffer:
                return 0

            # Drain only a bounded number of blocks to keep flush latency
            # predictable
            blocks_to_write = self._log_buffer[:max_blocks]
            del self._log_buffer[:max_blocks]

        samples = np.concatenate(blocks_to_write)
        with open(self._log_path, "a", encoding="utf-8") as out_file:
            for value in samples:
                out_file.write(f"{float(value)}\n")

        return int(samples.size)
```

session_controller.py

```
"""
session_controller.py

Session-level state machine that coordinates IFC stimulation output and EMG
acquisition during a treatment cycle.

Notes
-----
- Time progression is externally driven: callers must invoke tick_second()
once per second.
- EMG trigger checks are externally driven: callers should invoke
check_emg_threshold() periodically during the green stage.
"""
from dataclasses import dataclass
from emg_acquisition import EMGAcquisition
from signal_stream import SignalStream
from volume_lock_calibration import volume_to_predicted_current_ma

# Session stages:
# red    -> rest
# green  -> voluntary contraction with EMG monitoring
# yellow -> post-threshold maximal contraction window
# grey   -> preparation stage before entering cyclic red/green logic
STAGE_RED = 0
STAGE_GREEN = 1
STAGE_YELLOW = 2
STAGE_GREY = 3

@dataclass
class SessionSnapshot:
    """
    Immutable-style view of current session state for UI and control polling.

    Fields
    ------
    stage : int
        Current stage identifier (0 red, 1 green, 2 yellow, 3 grey).
    stage_timer : int
        Remaining seconds in the current stage countdown.
        For green stage this is always 0 (open-ended until threshold trigger).
    treatment_elapsed : int
        Total elapsed treatment time in seconds.
    treatment_total : int
        Total treatment duration in seconds.
    message : str
        Human-readable status string suitable for UI display.
    stimulation_on : bool
```

```
        Whether IFC output is currently active.
    completed : bool
        True after total treatment time has been reached.
    volume_percent : float
        Latest Windows output volume, in percent, used to compute predicted
        current.
    predicted_current_ma : float
        Estimated output current in mA derived from the volume-percent
        calibration curve.
    activation_threshold : float
        EMG RMS threshold in original sEMG mV required to trigger transition
        green -> yellow.
    """

    stage: int
    stage_timer: int
    treatment_elapsed: int
    treatment_total: int
    message: str
    stimulation_on: bool
    completed: bool
    volume_percent: float
    predicted_current_ma: float
    activation_threshold: float


class SessionController:
    """
    Coordinate stage transitions between IFC stimulation and EMG acquisition.

    The controller is clock-driven by external calls:
    - Call tick_second() once per second.
    - Call check_emg_threshold() periodically during green stage.

    This class owns streaming resources and exposes snapshots for UI polling.
    """
    def __init__(
        self,
        f1: float,
        f2: float,
        beat: float,
        ifc_mode: int,
        volume_percent: float,
        activation_threshold: float,
        treatment_minutes: float,
        fs_out: int = 48000,
        fs_in: int = 44100,
    ) -> None:
        """
```

```
        Initialize a treatment session controller.

        Parameters
        ----------
        f1 : float
            First carrier frequency in Hz.
        f2 : float
            Second carrier frequency in Hz.
        beat : float
            Beat frequency parameter for IFC synthesis.
        ifc_mode : int
            IFC mode selector used by SignalStream.
        volume_percent : float
            Windows output volume, in percent, used to derive predicted current.
        activation_threshold :
            EMG RMS threshold in original sEMG mV to trigger yellow stage.
        treatment_minutes : float
            Total treatment duration in minutes.
        fs_out : int, optional
            Output sample rate for IFC streaming (default 48000).
        fs_in : int, optional
            Input sample rate for EMG acquisition (default 44100).
        """
        self.f1 = f1
        self.f2 = f2
        self.beat = beat
        self.ifc_mode = ifc_mode

        self.activation_threshold = activation_threshold
        # floor of 1s guards against a zero/negative duration completing
        # before any stage runs
        self.treatment_total_seconds = max(
            1, int(round(treatment_minutes * 60.0))
        )
        self.treatment_elapsed_seconds = 0

        self.volume_percent = volume_percent
        self.predicted_current_ma = volume_to_predicted_current_ma(
            self.volume_percent
        )

        self.signal_stream = SignalStream(samplerate=fs_out)
        self.emg = EMGAcquisition(samplerate=fs_in)

        # Start in preparation (grey) state; session stays idle until start().
        self.stage = STAGE_GREY
        self.stage_timer = 10
        self.running = False
        self.completed = False
```

```
146         self.stimulation_on = False
147
148     def start(self) -> SessionSnapshot:
149         """
150         Start a new session cycle and reset runtime counters.
151
152         Enables IFC and EMG logging, enters preparation (grey) stage,
153         and returns the initial snapshot for UI synchronization.
154
155         Returns
156         -------
157         SessionSnapshot
158             Current session snapshot after initialization.
159         """
160         self.running = True
161         self.completed = False
162         self.treatment_elapsed_seconds = 0
163
164         self.signal_stream.configure_ifc_logging(
165             path="ifc_signal.csv", enabled=True, reset_file=True
166         )
167         self.emg.configure_emg_logging(
168             path="emg_signal.csv", enabled=True, reset_file=True
169         )
170
171         self._set_stage(STAGE_GREY)
172         return self.get_snapshot()
173
174     def stop(self) -> None:
175         """
176         Stop the session and release active streaming resources.
177
178         Stops EMG capture and stimulation output, then flushes pending log buffers.
179         """
180         self.running = False
181         self._stop_emg_capture()
182         self._stop_stimulation()
183         self.signal_stream.flush_ifc_log_buffer()
184         self.emg.flush_emg_log_buffer()
185
186     def update_volume_percent(self, volume_percent: float) -> SessionSnapshot:
187         """
188         Update predicted current from the current Windows output volume.
189
190         If the session is in grey stage and stimulation is already active,
191         stimulation is restarted so new parameters take effect immediately.
192
193         Parameters
194         ----------
```

```
        volume_percent : float
            New Windows output volume, in percent.

        Returns
        -------
        SessionSnapshot
            Updated session snapshot.
        """
        self.volume_percent = volume_percent
        self.predicted_current_ma = volume_to_predicted_current_ma(
            self.volume_percent
        )

        self._apply_stream_params()

        # Apply new amplitude immediately during preparation stimulation.
        # Note: unreachable while _start_stimulation() in _set_stage(GREY)
        # stays commented out.
        if self.running and self.stage == STAGE_GREY and self.stimulation_on:
            self._restart_stimulation()

        return self.get_snapshot()

    def tick_second(self) -> SessionSnapshot:
        """
        Advance session state by one second.

        This method must be called by an external 1 Hz timer. It updates
        treatment elapsed time, performs stage countdown transitions, and
        incrementally flushes signal log buffers.

        Returns
        -------
        SessionSnapshot
            Updated session snapshot.
        """
        if not self.running or self.completed:
            return self.get_snapshot()

        self.treatment_elapsed_seconds += 1
        if self.treatment_elapsed_seconds >= self.treatment_total_seconds:
            self.completed = True
            self.stop()
            return self.get_snapshot()

        if self.stage in (STAGE_GREY, STAGE_RED, STAGE_YELLOW):
            if self.stage_timer > 0:
                self.stage_timer -= 1

```

```
            if self.stage_timer == 0:
                if self.stage == STAGE_GREY:
                    self._set_stage(STAGE_GREEN)
                elif self.stage == STAGE_RED:
                    self._set_stage(STAGE_GREEN)
                elif self.stage == STAGE_YELLOW:
                    self._set_stage(STAGE_RED)

        self.signal_stream.flush_ifc_log_buffer(max_blocks=64)
        self.emg.flush_emg_log_buffer(max_blocks=64)

        return self.get_snapshot()

    def check_emg_threshold(self) -> tuple[bool, float, SessionSnapshot]:
        """
        Check whether EMG RMS exceeds activation threshold during green stage.

        Returns
        -------
        tuple[bool, float, SessionSnapshot]
            A tuple of:
            - triggered: True when current RMS exceeds activation threshold.
            - current_rms: Latest RMS value measured.
            - snapshot: Updated session snapshot.
        """

        if not self.running or self.completed or self.stage != STAGE_GREEN:
            return False, 0.0, self.get_snapshot()

        current_rms = self.emg.get_last_rms()
        if current_rms > self.activation_threshold:
            self._set_stage(STAGE_YELLOW)
            return True, current_rms, self.get_snapshot()

        return False, current_rms, self.get_snapshot()

    def get_snapshot(self) -> SessionSnapshot:
        """
        Return a snapshot of current session state for UI/control polling.
        """
        return SessionSnapshot(
            stage=self.stage,
            stage_timer=self.stage_timer,
            treatment_elapsed=self.treatment_elapsed_seconds,
            treatment_total=self.treatment_total_seconds,
            message=self._build_message(),
            stimulation_on=self.stimulation_on,
            completed=self.completed,
            volume_percent=self.volume_percent,
```

```
            predicted_current_ma=self.predicted_current_ma,
            activation_threshold=self.activation_threshold,
        )

    def _set_stage(self, new_stage: int) -> None:
        """
        Transition to a new stage and apply stage-specific side effects.

        stage actions:
        - grey: prepare countdown.
        - red: stop EMG and stimulation for rest.
        - green: stop stimulation and start EMG monitoring.
        - yellow: stop EMG and restart stimulation for high-effort window.
        """
        self.stage = new_stage

        if new_stage == STAGE_GREY:
            self.stage_timer = 10
        elif new_stage == STAGE_RED:
            self.stage_timer = 4
            self._stop_emg_capture()
            self._stop_stimulation()
        elif new_stage == STAGE_GREEN:
            self.stage_timer = 0
            self._stop_stimulation()
            self._start_emg_capture()
        elif new_stage == STAGE_YELLOW:
            self.stage_timer = 5
            self._stop_emg_capture()
            self._start_stimulation()

    def _apply_stream_params(self) -> None:
        """Apply current IFC synthesis parameters to the output stream."""
        self.signal_stream.set_parameters(
            f1=self.f1,
            f2=self.f2,
            beat=self.beat,
            ifc_mode=self.ifc_mode,
        )

    def _start_stimulation(self) -> None:
        """Start IFC stimulation with the currently configured parameters."""
        self._apply_stream_params()
        self.signal_stream.start()
        self.stimulation_on = True

    def _stop_stimulation(self) -> None:
        """Stop IFC output without blocking on a full log-buffer flush."""
        self.signal_stream.stop(flush_log=False)
```

```
342         self.stimulation_on = False
343
344     def _restart_stimulation(self) -> None:
345         """Restart IFC stimulation to apply updated runtime parameters."""
346         self.signal_stream.stop(flush_log=False)
347         self._apply_stream_params()
348         self.signal_stream.start()
349         self.stimulation_on = True
350
351     def _start_emg_capture(self) -> None:
352         """Start EMG acquisition."""
353         self.emg.start()
354
355     def _stop_emg_capture(self) -> None:
356         """Stop EMG acquisition."""
357         self.emg.stop()
358
359     def _build_message(self) -> str:
360         """Build a UI status message from the current stage and timer."""
361         if self.completed:
362             return "Session complete"
363
364         if self.stage == STAGE_RED:
365             return f"Take a rest for {self.stage_timer}s"
366         if self.stage == STAGE_GREEN:
367             return "Contract"
368         if self.stage == STAGE_YELLOW:
369             return f"Contract as much as possible for {self.stage_timer}s"
370         return f"Prepare for the exercise in {self.stage_timer}s"
```

patient_window.py

```
1  """
2  patient_window.py
3
4  Patient-facing session window with color stage feedback and timer-driven updates.
5  """
6
7  import numpy as np
8  from PySide6.QtCore import QTimer, Signal, Qt
9  from PySide6.QtWidgets import (
10     QLabel,
11     QMainWindow,
12     QVBoxLayout,
13     QWidget,
14 )
15
```

```
from session_controller import (
    STAGE_GREEN,
    STAGE_GREY,
    STAGE_RED,
    STAGE_YELLOW,
    SessionController,
    SessionSnapshot,
)

# Maps each stage id to the (label text, background color) shown on the stage
↪  square
STAGE_DISPLAY = {
    STAGE_GREY: ("Preparation", "#b8b8b8"),
    STAGE_RED: ("Rest", "#c94a4a"),
    STAGE_GREEN: ("Contract", "#4f9a5a"),
    STAGE_YELLOW: ("Maximum", "#d1b64a"),
}

class PatientWindow(QMainWindow):
    """Display the patient session state and relay user input to the controller."""
    # Emitted when the session ends, naturally or via window close
    session_finished = Signal()
    rms_updated = Signal(float)  # emitted with the latest EMG RMS value (mV)
    # Emitted with total elapsed treatment time (seconds)
    time_updated = Signal(int)
    # Emitted with newly captured EMG samples for plotting
    emg_signal_updated = Signal(np.ndarray)

    def __init__(self, session_config: dict, parent=None) -> None:
        """Create the window, start the session controller, and begin polling.

        session_config keys are forwarded directly to SessionController's
        constructor.
        """
        super().__init__(parent)

        self.setWindowTitle("Patient Session")
        self.resize(760, 520)

        self.controller = SessionController(**session_config)

        self._init_ui()

        self.tick_timer = QTimer(self)
        self.tick_timer.timeout.connect(self._on_tick_second)

        self.emg_check_timer = QTimer(self)
        self.emg_check_timer.timeout.connect(self._on_emg_check)

```

```
        snapshot = self.controller.start()
        self._refresh_ui(snapshot)

        self.tick_timer.start(1000)
        self.emg_check_timer.start(100)

    def _init_ui(self) -> None:
        """Build the patient-facing layout containing only the centered
        stage square."""
        central_widget = QWidget()
        self.setCentralWidget(central_widget)

        root_layout = QVBoxLayout(central_widget)
        root_layout.setAlignment(Qt.AlignmentFlag.AlignCenter)

        self.stage_square = QLabel("Preparation")
        self._init_stage_square(self.stage_square)
        root_layout.addWidget(self.stage_square)

    def _init_stage_square(self, label: QLabel) -> None:
        """Apply base styling to the single dynamic stage indicator square."""
        label.setMinimumHeight(400)
        label.setMinimumWidth(400)
        label.setAlignment(Qt.AlignmentFlag.AlignCenter)
        label.setStyleSheet(
            "border: 5px solid #111;"
            "font-size: 28px; font-weight: 700;"
        )

    def _on_tick_second(self) -> None:
        """Advance the session clock and stop polling when treatment is done."""
        snapshot = self.controller.tick_second()
        self._refresh_ui(snapshot)

        # Broadcast the elapsed time to main_window.py
        self.time_updated.emit(snapshot.treatment_elapsed)

        if snapshot.completed:
            self._end_session()


    def _on_emg_check(self) -> None:
        """Poll the EMG controller and refresh the UI after a threshold event."""
        triggered, latest_rms, snapshot = self.controller.check_emg_threshold()

        # Broadcast the value to any listeners in main_window.py
        self.rms_updated.emit(latest_rms)

        plot_samples = self.controller.emg.get_pending_plot_samples()
```

```
        if plot_samples.size > 0:
            self.emg_signal_updated.emit(plot_samples)

        if triggered:
            self._refresh_ui(snapshot)

    def _refresh_ui(self, snapshot: SessionSnapshot) -> None:
        """Synchronize the stage indicator from a snapshot."""
        self._refresh_stage_square(snapshot.stage)

    def _refresh_stage_square(self, stage: int) -> None:
        """Update the single stage square with text and color for current stage."""
        stage_text, stage_color = STAGE_DISPLAY.get(stage, ("Unknown", "#9c9c9c"))
        self.stage_square.setText(stage_text)
        self.stage_square.setStyleSheet(
            f"background-color: {stage_color};"
            "border: 5px solid #111;"
            "font-size: 28px; font-weight: 700;"
        )

    def _end_session(self) -> None:
        """End the session by closing the window; closeEvent() performs the
        actual teardown."""
        self.close()

    def _stop_timers(self) -> None:
        """Stop background timers if they are currently running."""
        if self.tick_timer.isActive():
            self.tick_timer.stop()
        if self.emg_check_timer.isActive():
            self.emg_check_timer.stop()

    def closeEvent(self, event) -> None:
        """Ensure controller and timers are stopped when the window closes."""
        self.controller.stop()
        self._stop_timers()
        self.session_finished.emit()
        super().closeEvent(event)
```

main_window.py

```
"""
main_window.py

This script contains the operator-facing main window (EMGTriggeredIFTModule),
responsible for building the Qt UI, previewing the interference waveform,
polling the Windows output volume, and launching/ending the patient-facing
```

```
session (PatientWindow + SessionController).
"""

import sys
import numpy as np

# Matplotlib integration for Qt
import matplotlib
matplotlib.use('QtAgg')
from matplotlib.backends.backend_qtagg import FigureCanvasQTAgg as FigureCanvas
from matplotlib.figure import Figure

# PySide6 UI Modules
from PySide6.QtWidgets import (
    QApplication, QMainWindow, QWidget, QVBoxLayout,
    QHBoxLayout, QLabel, QSlider, QGroupBox,
    QRadioButton, QButtonGroup, QPushButton, QSpinBox
)
from PySide6.QtCore import Qt, QTimer

from signal_generator import carrier_wave
from signal_stream import SignalStream
from volume_lock_calibration import (
    get_output_volume_scalar,
    set_output_volume_scalar,
    volume_to_predicted_current_ma,
)
from patient_window import PatientWindow

# Horizontal axis limit, a bit over the maximum 61 mA expected patient current
CURRENT_Y_LIMIT  = 65 # mA

class MplCanvas(FigureCanvas):
    """
    Custom Matplotlib canvas widget to host the a plot.
    """
    def __init__(self, parent=None, width=8, height=6, dpi=100) -> None:
        # Create Matplotlib Figure with a single subplot
        self.fig = Figure(figsize=(width, height), dpi=dpi, layout='constrained')
        self.ax = self.fig.subplots()
        super().__init__(self.fig)


class EMGTriggeredIFTModule(QMainWindow):
    """
    Main Application Window for Amplitude Modulation / Wave Interference
    visualization.

    Responsibilities of this class:
```

```
        - Build and lay out Qt widgets.
        - Read widget state (sliders, radio buttons) into plain numbers.
        - Call pure functions from signal_generator.py to compute waveforms.
        - Push results back into the plot via set_data().

    This class does not compute waveform math itself and delegates session
    execution to a dedicated patient window and controller.
    """
    def __init__(self) -> None:
        super().__init__()

        self.setWindowTitle("Operator Window")
        self.resize(1000, 700)

        # Signal Parameters
        self.T = 0.05         # Total time duration for plot (seconds)
        self.fs = 48000    # Sampling rate (Hz) - default of computer soundcard
        self.time = np.linspace(0, self.T, int(self.T * self.fs), endpoint=False)

        # State mirrors for the currently selected parameters
        # Read by _update_interference_plot()
        self.latest_f1 = 4000
        self.latest_f2 = 4050
        self.latest_IFCmode = 2
        self.latest_volume_percent = 0.0
        self.predicted_I_max = 0.0
        self.patient_window = None
        self._last_volume_percent = None
        self._emg_plot_samples = np.array([], dtype=np.float32)

        # New state variables for volume calibration
        self.is_setting_volume = False
        self.calibration_stream = None
        self.locked_volume_scalar = None

        # Setup GUI Components
        self._init_ui()
        self._init_plots()
        self._update_interference_plot()

        self.volume_poll_timer = QTimer(self)
        self.volume_poll_timer.timeout.connect(self._poll_volume)
        self.volume_poll_timer.start(300)

############################## UI CONSTRUCTION ##################################

    def _init_ui(self) -> None:
        """Initializes all widgets, layouts, and slider/button signals."""
        # Main central widget and horizontal layout
```

```
105         central_widget = QWidget()
106         self.setCentralWidget(central_widget)
107         main_layout = QHBoxLayout(central_widget)
108 
109         # --- Sidebar Container ---
110         sidebar_widget = QWidget()
111         sidebar_layout = QVBoxLayout(sidebar_widget)
112         sidebar_layout.setContentsMargins(0, 0, 0, 0)
113         sidebar_widget.setFixedWidth(280)
114 
115         # Control Panel Sidebar
116         wave_panel = QGroupBox("Wave Parameters")
117         wave_layout = QVBoxLayout(wave_panel)
118         control_layout = QVBoxLayout()
119 
120         # Adjust internal layout spacing (controls tight vertical spacing)
121         control_layout.setSpacing(6)
122 
123         # --- Carrier Frequency (f1) ---
124         self.f1_label = QLabel()
125 
126         # Create radio buttons to choose carrier frequency
127         self.f1_buttons = QButtonGroup(self)
128         f1_row = QHBoxLayout()
129         self.f1_button_2k = QRadioButton("2 kHz")
130         self.f1_button_4k = QRadioButton("4 kHz")
131         self.f1_button_6k = QRadioButton("6 kHz")
132 
133         # Add buttons to button group
134         # Button ID is the numeric value of carrier frequency
135         self.f1_buttons.addButton(self.f1_button_2k, 2000)
136         self.f1_buttons.addButton(self.f1_button_4k, 4000)
137         self.f1_buttons.addButton(self.f1_button_6k, 6000)
138 
139         # Set 4 kHz as default
140         self.f1_button_4k.setChecked(True)
141 
142         # Connect button to click signal
143         self.f1_buttons.idClicked.connect(self._update_interference_plot)
144 
145         # --- Beat Frequency (beat) Slider ---
146         self.beat_label = QLabel()
147         self.beat_slider = QSlider(Qt.Orientation.Horizontal)
148         self.beat_slider.setRange(0,250)
149         self.beat_slider.setValue(50)
150         self.beat_slider.valueChanged.connect(self._update_interference_plot)
151 
152         # # Create f2 (modulating frequency) label to be updated according to the
153         # # beat frequency and to be shown in the plot title
```

```
154         # self.f2_label = QLabel()
155
156         # Create radio buttons to select stimulation mode:
157         # - True IFC = 4 electrodes, interference happens in the targeted structure
158         # - Premodulated IFC = 2 electrodes, interference is computed and emitted
159         # in single channel
160         self.IFCmode_label = QLabel()
161         self.IFCmode_buttons = QButtonGroup(self)
162         self.trueIFC_button = QRadioButton("True IFC (4 electrodes)")
163         self.premodIFC_button = QRadioButton("Premodulated IFC (2 electrodes)")
164
165         # --- Windows Output Volume / Predicted Current (read-only) ---
166         self.volume_label = QLabel()
167         self.ampl_label = QLabel()
168
169         # Add mode buttons to respective button group
170         self.IFCmode_buttons.addButton(self.trueIFC_button, 4)
171         self.IFCmode_buttons.addButton(self.premodIFC_button, 2)
172         self.premodIFC_button.setChecked(True) # set premodulated IFCs as default
173         # connect button to click signal
174         self.IFCmode_buttons.idClicked.connect(self._update_interference_plot)
175
176         # --- Activation Threshold Slider ---
177         # Range in tenths of mV (1-10 -> 0.1-1.0 mV), capped near the realistic RMS
178         # a contraction can reach (well below the 2.5 mV electrical ceiling of
179         # EMG_SAMPLE_SCALE_MV in emg_acquisition.py).
180         self.activation_label = QLabel()
181         self.activation_slider = QSlider(Qt.Orientation.Horizontal)
182         self.activation_slider.setRange(1, 10)
183         self.activation_slider.setValue(5)
184         self.activation_slider.valueChanged.connect(self._update_interference_plot)
185
186         # --- Treatment Time ---
187         self.treatment_label = QLabel("Treatment time (minutes)")
188         self.treatment_spinbox = QSpinBox()
189         self.treatment_spinbox.setRange(1, 120)
190         self.treatment_spinbox.setValue(1)
191
192         # Volume Setting Button
193         self.set_volume_button = QPushButton("Set Volume")
194         self.set_volume_button.setStyleSheet(
195             "background-color: #F2A113; color: white; font-weight: bold;"
196             )
197         self.set_volume_button.clicked.connect(self._set_volume)
198
199         # Session Control Button
200         self.start_session_button = QPushButton("Start Session")
201         self.start_session_button.setStyleSheet(
202             "background-color: #4CAF50; color: white; font-weight: bold;"
```

```
            )
        self.start_session_button.clicked.connect(self._start_session)

        # Add f1 elements to sidebar - buttons need to be added individually
        wave_layout.addWidget(self.f1_label)
        f1_row.addWidget(self.f1_button_2k)
        f1_row.addWidget(self.f1_button_4k)
        f1_row.addWidget(self.f1_button_6k)
        wave_layout.addLayout(f1_row)

        # Add beat elements to sidebar
        wave_layout.addSpacing(10)
        wave_layout.addWidget(self.beat_label)
        wave_layout.addWidget(self.beat_slider)

        # Add mode buttons to sidebar
        wave_layout.addSpacing(10)
        wave_layout.addWidget(self.IFCmode_label)
        wave_layout.addWidget(self.trueIFC_button, 4)
        wave_layout.addWidget(self.premodIFC_button, 2)

        # Add volume/predicted current labels to sidebar
        wave_layout.addSpacing(10)
        wave_layout.addWidget(self.volume_label)
        wave_layout.addWidget(self.ampl_label)

        wave_layout.addSpacing(10)
        wave_layout.addWidget(self.activation_label)
        wave_layout.addWidget(self.activation_slider)

        wave_layout.addSpacing(10)
        wave_layout.addWidget(self.treatment_label)
        wave_layout.addWidget(self.treatment_spinbox)

        wave_layout.addSpacing(10)
        wave_layout.addWidget(self.set_volume_button)

        # --- Contour 2: Session Control Group ---
        session_panel = QGroupBox("Session")
        session_layout = QVBoxLayout(session_panel)
        session_layout.setSpacing(6)

        self.main_rms_label = QLabel("Latest RMS: 0.0000")
        self.main_time_label = QLabel("Time Elapsed: 0 s")

        session_layout.addWidget(self.main_rms_label)
        session_layout.addWidget(self.main_time_label)
        session_layout.addSpacing(10)
        session_layout.addWidget(self.start_session_button)
```

```
252
253         # Assemble Sidebar
254         sidebar_layout.addWidget(wave_panel)
255         sidebar_layout.addWidget(session_panel)
256         sidebar_layout.addStretch()
257
258         # --- Plot Canvas Area ---
259         self.interference_canvas = MplCanvas(self)
260         self.emg_canvas = MplCanvas(self)
261
262         plot_layout = QVBoxLayout()
263         plot_layout.addWidget(self.interference_canvas)
264         plot_layout.addWidget(self.emg_canvas)
265
266         plot_container = QWidget()
267         plot_container.setLayout(plot_layout)
268
269         # Add sidebar and plot container to main layout
270         main_layout.addWidget(sidebar_widget)
271         main_layout.addWidget(plot_container)
272
273     def _init_plots(self) -> None:
274         """
275         Initializes plot lines and sets static axis limits/formatting once.
276         Prevents visual jumpiness when updating slider values.
277         """
278         # Plot 1: Interference Wave
279         self.interference_line, = self.interference_canvas.ax.plot(
280             [], [], color='#4268b4'
281         )
282         self.interference_canvas.ax.set_xlabel('Time (seconds)')
283         self.interference_canvas.ax.set_ylabel('Amplitude (mA)')
284         self.interference_canvas.ax.set_ylim(-CURRENT_Y_LIMIT , CURRENT_Y_LIMIT )
285         self.interference_canvas.ax.set_xlim(0, self.T)
286         self.interference_canvas.ax.grid(True, alpha=0.3)
287
288         # Plot 2: Real-time acquired EMG signal
289         self.emg_line, = self.emg_canvas.ax.plot([], [], color='#d62728')
290         self.emg_canvas.ax.set_ylabel('Amplitude (mV)')
291         self.emg_canvas.ax.set_xlim(0, self.T)
292         self.emg_canvas.ax.grid(True, alpha=0.3)
293
294 ################################### SLOTS ###################################
295
296     def _update_interference_plot(self) -> None:
297         """
298         Reads current widget state, delegates waveform math to
299         signal_generator.py, and pushes the interference wave to the
300         dedicated plot.
```

```
        """
        self.latest_f1 = self.f1_buttons.checkedId()
        beat = self.beat_slider.value()
        self.latest_f2 = self.latest_f1 + beat
        self.latest_IFCmode = self.IFCmode_buttons.checkedId()

        # Update text labels
        self.f1_label.setText(f"Carrier Frequency (f1): {self.latest_f1} Hz")
        # self.f2_label.setText(f"Modulator Frequency (f2): {self.latest_f2} Hz")
        self.beat_label.setText(f"Beat Frequency (f1 - f2): {beat} Hz")
        self.IFCmode_label.setText(f"IFC Mode: {self.latest_IFCmode}")
        self.volume_label.setText(
            f"System Volume: {self.latest_volume_percent:.0f}%"
        )
        self.ampl_label.setText(
            f"Predicted Current Maximum: {self.predicted_I_max:.2f} mA"
        )
        self.activation_label.setText(
            f"Activation Threshold: {self._activation_threshold_value():.1f} mV"
        )

        # Generate signals from helper functions in signal_generator.py
        wave1 = carrier_wave(
            freq=self.latest_f1, time=self.time, ampl=self.predicted_I_max
        )
        wave2 = carrier_wave(
            freq=self.latest_f2, time=self.time, ampl=self.predicted_I_max
        )
        interference_wave = (wave1 + wave2) / 2.0

        # Update line data
        self.interference_line.set_data(self.time, interference_wave)

        # Update plot titles dynamically
        self.interference_canvas.ax.set_title(
            f'Interference Wave (Beat Frequency = {beat} Hz)'
        )
        self.emg_canvas.ax.set_title('EMG Signal')

        # Redraw canvases efficiently
        self.interference_canvas.draw_idle()
        self.emg_canvas.draw_idle()

    def _activation_threshold_value(self) -> float:
        """Convert the activation slider position (1-10) into an EMG RMS
        threshold in mV."""
        return self.activation_slider.value() / 10.0

    def _poll_volume(self) -> None:
```

```
        """Read the Windows output volume and refresh dependent
        labels/plot on change."""
        volume_percent = get_output_volume_scalar() * 100.0
        if volume_percent == self._last_volume_percent:
            return

        self._last_volume_percent = volume_percent
        self.latest_volume_percent = volume_percent
        self.predicted_I_max = volume_to_predicted_current_ma(volume_percent)
        self._update_interference_plot()

        if self.patient_window is not None:
            self.patient_window.controller.update_volume_percent(volume_percent)

    def _start_session(self) -> None:
            """Toggle a treatment session: launch a configured
            PatientWindow, or end the active one."""
            if self.is_setting_volume:
                return

            # If a session is already active, shut it down
            if self.patient_window is not None:
                self.patient_window._end_session()
                return

            # Lock the system volume to the calibrated level before starting
            if self.locked_volume_scalar is not None:
                set_output_volume_scalar(self.locked_volume_scalar)

            # 2. Prevent opening multiple patient windows
            if self.patient_window is not None and self.patient_window.isVisible():
                self.patient_window.raise_()
                self.patient_window.activateWindow()
                return

            # 3. Initialize the actual session
            beat = self.beat_slider.value()
            session_config = {
                "f1": float(self.latest_f1),
                "f2": float(self.latest_f1 + beat),
                "beat": float(beat),
                "ifc_mode": int(self.latest_IFCmode),
                "volume_percent": float(self.latest_volume_percent),
                "activation_threshold": float(self._activation_threshold_value()),
                "treatment_minutes": float(self.treatment_spinbox.value()),
                "fs_out": int(self.fs),
                "fs_in": 44100,
            }

```

```
            # Update button to act as the End Session switch
            self.start_session_button.setText("End Session")
            self.start_session_button.setStyleSheet(
                "background-color: #F44336; color: white; font-weight: bold;"
            )

            self.patient_window = PatientWindow(
                session_config=session_config, parent=self
            )
            self._emg_plot_samples = np.array([], dtype=np.float32)
            self.patient_window.session_finished.connect(self._on_session_finished)

            # New signal connections
            self.patient_window.rms_updated.connect(self._update_main_rms)
            self.patient_window.time_updated.connect(self._update_main_time)
            self.patient_window.emg_signal_updated.connect(self._update_emg_plot)
            self.patient_window.show()

    def _update_main_rms(self, latest_rms: float) -> None:
        """Update the sidebar RMS label from the patient window's
        rms_updated signal."""
        self.main_rms_label.setText(f"Latest RMS: {latest_rms:.5f} mV")

    def _update_main_time(self, elapsed_seconds: int) -> None:
        """Update the sidebar elapsed-time label from the patient
        window's time_updated signal."""
        self.main_time_label.setText(f"Time Elapsed: {elapsed_seconds} s")

    def _update_emg_plot(self, emg_block: np.ndarray) -> None:
        """Append captured EMG samples and display a rolling time window."""
        if self.patient_window is None:
            return
        fs_in = self.patient_window.controller.emg.fs
        max_samples = int(self.T * fs_in)
        self._emg_plot_samples = np.concatenate(
            (self._emg_plot_samples, emg_block)
        )[-max_samples:]
        time_axis = np.arange(self._emg_plot_samples.size) / fs_in

        self.emg_line.set_data(time_axis, self._emg_plot_samples)
        self.emg_canvas.ax.relim()
        self.emg_canvas.ax.autoscale_view(scalex=False, scaley=True)
        self.emg_canvas.draw_idle()

    def _on_session_finished(self) -> None:
        """Reset the internal state and UI when the patient window closes."""
        self.patient_window = None

        # Revert button appearance to default state
```

```
448         self.start_session_button.setText("Start Session")
449         self.start_session_button.setStyleSheet(
450             "background-color: #4CAF50; color: white; font-weight: bold;"
451         )
452 
453     def _set_volume(self) -> None:
454         """Toggle volume calibration: preview the wave at an adjustable
455         volume, then lock it in."""
456         if not self.is_setting_volume:
457             # --- START CALIBRATION ---
458             self.is_setting_volume = True
459             self.start_session_button.setEnabled(False)
460 
461             # Update UI button appearance
462             self.set_volume_button.setText("End Volume Setting")
463             self.set_volume_button.setStyleSheet(
464                 "background-color: #F44336; color: white; font-weight: bold;"
465             )
466 
467             # Drop Windows sound bar to 0% so that operator can gradually
468             # increase volume until patient can feel the current
469             set_output_volume_scalar(0.0)
470 
471             # Initialize a temporary stream for testing
472             if self.calibration_stream is None:
473                 self.calibration_stream = SignalStream(
474                     samplerate=self.fs, channels=2
475                 )
476 
477             beat = self.beat_slider.value()
478             self.calibration_stream.set_parameters(
479                 f1=float(self.latest_f1),
480                 f2=float(self.latest_f1 + beat),
481                 beat=float(beat),
482                 ifc_mode=int(self.latest_IFCmode),
483             )
484             self.calibration_stream.start()
485 
486         else:
487             # --- END CALIBRATION ---
488             self.is_setting_volume = False
489             self.start_session_button.setEnabled(True)
490 
491             # Revert UI button appearance
492             self.set_volume_button.setText("Set Volume")
493             self.set_volume_button.setStyleSheet(
494                 "background-color: #F2A113; color: white; font-weight: bold;"
495             )
496 
```

```
            # Stop the temporary testing stream
            if self.calibration_stream is not None:
                self.calibration_stream.stop()

            # Read the adjusted volume from the OS and lock it in memory
            self.locked_volume_scalar = get_output_volume_scalar()

def main():
    app = QApplication(sys.argv)
    window = EMGTriggeredIFTModule()
    window.show()
    sys.exit(app.exec())

if __name__ == "__main__":
    main()
```

References

[1] Christopher C. Kaeding, Benjamin Léger-St-Jean, and Robert A. Magnussen. Epidemiology and Diagnosis of Anterior Cruciate Ligament Injuries. *Clinics in Sports Medicine*, 36(1): 1–8, January 2017. ISSN 1556-228X. doi: 10.1016/j.csm.2016.08.001.

[2] Shevaun M. Doyle, Peter D. Fabricant, Stephen Fealy, Daniel W. Green, Michelle E. Kew, and Madeline Kratz. ACL Reconstruction Surgery: Procedure and Recovery, July 2024. URL https://www.hss.edu/health-library/conditions-and-treatments/list/acl-surgery.

[3] Jeffrey D. Hassebrock, Matthew T. Gulbrandsen, Walker L. Asprey, Justin L. Makovicka, and Anikar Chhabra. Knee Ligament Anatomy and Biomechanics. *Sports Medicine and Arthroscopy Review*, 28(3):80–86, September 2020. ISSN 1062-8592. doi: 10.1097/JSA.0000000000000279. URL https://journals.lww.com/10.1097/JSA.0000000000000279.

[4] Sinem Suner Keklik, Nevin Güzel, Gamze Çobanoğlu, Nihan Kafa, Muhammet Baybars Ataoğlu, and Zekeriya Öztemür. Evaluation of proprioception in patients who underwent ACL reconstruction: measurement in functional position. *Turkish Journal of Medical Sciences*, 51(4):2036–2042, August 2021. ISSN 1303-6165. doi: 10.3906/sag-2004-110.

[5] V. B. Duthon, C. Barea, S. Abrassart, J. H. Fasel, D. Fritschy, and J. Ménétrey. Anatomy of the anterior cruciate ligament. *Knee Surgery, Sports Traumatology, Arthroscopy*, 14(3): 204–213, March 2006. ISSN 0942-2056, 1433-7347. doi: 10.1007/s00167-005-0679-9. URL https://esskajournals.onlinelibrary.wiley.com/doi/10.1007/s00167-005-0679-9.

[6] Jonathan Kosy and Vipul Mandalia. Anterior Cruciate Ligament Mechanoreceptors and their Potential Importance in Remnant-Preserving Reconstruction: A Review of Basic Science and Clinical Findings. *The Journal of Knee Surgery*, 31(08):736–746, September 2018. ISSN 1538-8506, 1938-2480. doi: 10.1055/s-0037-1608941. URL http://www.thieme-connect.de/DOI/DOI?10.1055/s-0037-1608941.

[7] Mikołaj Stańczak, Bram Swinnen, Bartłomiej Kacprzak, Artur Pacek, and Jakub Surmacz. Neurophysiology of ACL Injury. *Orthopedic Reviews*, 17, February 2025. doi: 10.52965/001c.129173. URL https://orthopedicreviews.openmedicalpublishing.org/article/129173-neurophysiology-of-acl-injury.

[8] Mandeep S. Dhillon, Kamal Bali, and Sharad Prabhakar. Proprioception in anterior cruciate ligament deficient knees and its relevance in anterior cruciate ligament reconstruction. *Indian Journal of Orthopaedics*, 45(4):294–300, July 2011. ISSN 1998-3727. doi: 10.4103/0019-5413.80320.

[9] Bertrand Sonnery-Cottet, Thomas Ripoll, and Etienne Cavaignac. Prevention of knee stiffness following ligament reconstruction: Understanding the role of Arthrogenic Muscle Inhibition (AMI). *Orthopaedics & traumatology, surgery & research: OTSR*, 110(1S):103784, February 2024. ISSN 1877-0568. doi: 10.1016/j.otsr.2023.103784.

[10] Douglas Adams, David Logerstedt, Airelle Hunter-Giordano, Michael J. Axe, and Lynn Snyder-Mackler. Current Concepts for Anterior Cruciate Ligament Reconstruction: A Criterion-Based Rehabilitation Progression. *Journal of Orthopaedic & Sports Physical Therapy*, 42(7):601–614, July 2012. ISSN 0190-6011, 1938-1344. doi: 10.2519/jospt.2012.3871. URL https://www.jospt.org/doi/10.2519/jospt.2012.3871.

[11] Uria Moran, Uri Gottlieb, Arnon Gam, and Shmuel Springer. Functional electrical stimulation following anterior cruciate ligament reconstruction: a randomized controlled pilot study. *Journal of NeuroEngineering and Rehabilitation*, 16(1):89, December 2019. ISSN 1743-0003. doi: 10.1186/s12984-019-0566-0. URL https://jneuroengrehab.biomedcentral.com/articles/10.1186/s12984-019-0566-0.

[12] Fabian Sanchis-Gomar, Sergio Lopez-Lopez, Carlos Romero-Morales, Nicola Maffulli, Giuseppe Lippi, and Helios Pareja-Galeano. Neuromuscular Electrical Stimulation: A New Therapeutic Option for Chronic Diseases Based on Contraction-Induced Myokine Secretion. *Frontiers in Physiology*, 10:1463, 2019. ISSN 1664-042X. doi: 10.3389/fphys.2019.01463.

[13] G. C. Goats. Interferential current therapy. *British Journal of Sports Medicine*, 24(2):87–92, June 1990. ISSN 0306-3674. doi: 10.1136/bjsm.24.2.87.

[14] Rodrigo Osorio, Jack Edmondson, Siobhan Mackenzie Hall, Francisco Saavedra, Javier Sáez, Adrian Poulton, James FitzGerald, Pablo Aqueveque, Brian Andrews, and Jonathan Jarvis. Characterization and Evaluation of Interferential Current Stimulation for Functional Electrical Stimulation. *Artificial Organs*, 49(12):1809–1820, December 2025. ISSN 0160-564X, 1525-1594. doi: 10.1111/aor.15027. URL https://onlinelibrary.wiley.com/doi/10.1111/aor.15027.

[15] Henry Gray and Peter L. Williams, editors. *Gray's anatomy*. Churchill Livingstone, Edinburgh, 37. ed edition, 1989. ISBN 978-0-443-02588-4.

[16] Adam P. Rumian, Andrew L. Wallace, and Helen L. Birch. Tendons and ligaments are anatomically distinct but overlap in molecular and morphological features–a comparative study in an ovine model. *Journal of Orthopaedic Research: Official Publication of the Orthopaedic Research Society*, 25(4):458–464, April 2007. ISSN 0736-0266. doi: 10.1002/jor.20218.

[17] Kelc Robi, Naranda Jakob, Kuhta Matevz, and Vogrin Matjaz. The Physiology of Sports Injuries and Repair Processes. In Michael Hamlin, editor, *Current Issues in Sports and Exercise Medicine*. InTech, May 2013. ISBN 978-953-51-1031-6. doi: 10.5772/54234. URL http://www.intechopen.com/books/current-issues-in-sports-and-exercise-medicine/the-physiology-of-sports-injuries-and-repair-processes.

[18] Kalpesh Parmar. Tendon and ligament: basic science, injury and repair. *Orthopaedics and Trauma*, 32(4):241–244, August 2018. ISSN 18771327. doi: 10.1016/j.mporth.2018.05.008. URL https://linkinghub.elsevier.com/retrieve/pii/S1877132718300733.

[19] Mark D. Miller and Stephen R. Thompson, editors. *Miller's review of orthopaedics*. Elsevier, Philadelphia, PA, eighth edition edition, 2020. ISBN 978-0-323-60978-4. OCLC: 1112145330.

[20] M. Marieswaran, Ishita Jain, Bhavuk Garg, Vijay Sharma, and Dinesh Kalyanasundaram. A Review on Biomechanics of Anterior Cruciate Ligament and Materials for Reconstruction. *Applied Bionics and Biomechanics*, 2018:1–14, 2018. ISSN 1176-2322, 1754-2103. doi: 10.1155/2018/4657824. URL https://www.hindawi.com/journals/abb/2018/4657824/.

[21] Physiopedia. Knee. https://www.physio-pedia.com/index.php?title=Knee&oldid=368494, May 2025. [Online; accessed 29-December-2025].

[22] Thore Zantop, Wolf Petersen, Jon K. Sekiya, Volker Musahl, and Freddie H. Fu. Anterior cruciate ligament anatomy and function relating to anatomical reconstruction. *Knee Surgery, Sports Traumatology, Arthroscopy*, 14(10):982–992, October 2006. ISSN 0942-2056, 1433-7347. doi: 10.1007/s00167-006-0076-z. URL https://esskajournals.onlinelibrary.wiley.com/doi/10.1007/s00167-006-0076-z.

[23] Xuezhi Yu, Jiahui Hu, Yifan Li, Yu Wen, and Bin Li. ACL injury management: a comprehensive review of novel biotherapeutics. *Frontiers in Bioengineering and Biotechnology*, 12:1455225, November 2024. ISSN 2296-4185. doi: 10.3389/fbioe.2024.1455225. URL https://www.frontiersin.org/articles/10.3389/fbioe.2024.1455225/full.

[24] R. Morales-Avalos, E.M. Torres-González, J.R. Padilla-Medina, and J.C. Monllau. ACL anatomy: Is there still something to learn? *Revista Española de Cirugía Ortopédica y Traumatología*, 68(4):422–427, July 2024. ISSN 18884415. doi: 10.1016/j.recot.2023.02.005. URL https://linkinghub.elsevier.com/retrieve/pii/S188844152300070X.

[25] Edwin Ocran. Anterior Cruciate Ligament: Anatomy and Function. https://www.kenhub.com/en/library/anatomy/anterior-cruciate-ligament, October 2023.

[26] Physiopedia contributors. Anterior cruciate ligament (ACL). Physiopedia. URL https://www.physio-pedia.com/index.php?title=Anterior_Cruciate_Ligament_(ACL)&oldid=372617. [Online; accessed 29-December-2025].

[27] Thilina N Vitharana, Enda King, and Kieran Moran. Sensorimotor Dysfunction Following Anterior Cruciate Ligament Reconstruction- an Afferent Perspective: A Scoping Review. *International Journal of Sports Physical Therapy*, 19(1):1410–1437. ISSN 2159-2896. doi: 10.26603/001c.90862. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC10761632/.

[28] Mandeep Sing Dhillon, Kamal Bali, and Sharad Prabhakar. Differences among mechanoreceptors in healthy and injured anterior cruciate ligaments and their clinical importance. *Muscles, Ligaments and Tendons Journal*, 2(1):38–43, June 2012. ISSN 2240-4554. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC3666492/.

[29] Lionel Chia, Danilo De Oliveira Silva, Matthew Whalan, Marnee J. McKay, Justin Sullivan, Colin W. Fuller, and Evangelos Pappas. Non-contact Anterior Cruciate Ligament Injury Epidemiology in Team-Ball Sports: A Systematic Review with Meta-analysis by Sex, Age, Sport, Participation Level, and Exposure Type. *Sports Medicine*, 52(10):2447–2467, October

2022. ISSN 1179-2035. doi: 10.1007/s40279-022-01697-w. URL https://doi.org/10.1007/s40279-022-01697-w.

[30] Huijuan Shi, Yanfang Jiang, Shuang Ren, Xiaoqing Hu, Hongshi Huang, and Yingfang Ao. Sex differences in the knee orthopaedic injury patterns among recreational alpine skiers. *BMC Sports Science, Medicine and Rehabilitation*, 12(1):74, December 2020. ISSN 2052-1847. doi: 10.1186/s13102-020-00224-6. URL https://doi.org/10.1186/s13102-020-00224-6.

[31] Tayt M. Ellison, Ilexa Flagstaff, and Anthony E. Johnson. Sexual Dimorphisms in Anterior Cruciate Ligament Injury: A Current Concepts Review. *Orthopaedic Journal of Sports Medicine*, 9(12):23259671211025304, December 2021. ISSN 2325-9671. doi: 10.1177/23259671211025304. URL https://doi.org/10.1177/23259671211025304.

[32] Jonathan D. Hughes, Cort D. Lawton, Danyal H. Nawabi, Andrew D. Pearle, and Al Et. Anterior Cruciate Ligament Repair: The Current Status. *Journal of Bone and Joint Surgery*, 2020. doi: 10.2106/JBJS.20.00509. URL https://sci-hub.box/10.2106/JBJS.20.00509.

[33] K. Markatos, M. K. Kaseta, S. N. Lallos, D. S. Korres, and N. Efstathopoulos. The anatomy of the ACL and its importance in ACL reconstruction. *European Journal of Orthopaedic Surgery & Traumatology*, 23(7):747–752, October 2013. ISSN 1633-8065, 1432-1068. doi: 10.1007/s00590-012-1079-8. URL http://link.springer.com/10.1007/s00590-012-1079-8.

[34] Kenji Hara, Tomoyuki Mochizuki, Ichiro Sekiya, Kumiko Yamaguchi, Keiichi Akita, and Takeshi Muneta. Anatomy of Normal Human Anterior Cruciate Ligament Attachments Evaluated by Divided Small Bundles. *The American Journal of Sports Medicine*, 37(12):2386–2391, December 2009. ISSN 0363-5465, 1552-3365. doi: 10.1177/0363546509340404. URL https://journals.sagepub.com/doi/10.1177/0363546509340404.

[35] National Health Service. How ACL (anterior cruciate ligament) surgery is done. https://www.nhs.uk/tests-and-treatments/acl-anterior-cruciate-ligament-surgery/how-its-done/. [Online; accessed 5-September-2026].

[36] Hospital for Special Surgery. ACL Reconstruction Surgery: Procedure and Recovery. https://www.hss.edu/health-library/conditions-and-treatments/list/acl-surgery, September 2026. Published: Hospital for Special Surgery.

[37] Bartłomiej Kacprzak. Molecular Biology of ACL Graft Healing: Early Mechanical Loading Perspective. *Orthopedic Reviews*, 17, July 2025. ISSN 2035-8164. doi: 10.52965/001c.140716. URL https://orthopedicreviews.openmedicalpublishing.org/article/140716-molecular-biology-of-acl-graft-healing-early-mechanical-loading-perspective.

[38] S. U. Scheffler, F. N. Unterhauser, and A. Weiler. Graft remodeling and ligamentization after cruciate ligament reconstruction. *Knee Surgery, Sports Traumatology, Arthroscopy*, 16(9):834–842, September 2008. ISSN 0942-2056, 1433-7347. doi: 10.1007/s00167-008-0560-8. URL https://esskajournals.onlinelibrary.wiley.com/doi/10.1007/s00167-008-0560-8.

[39] Shiyi Yao, Bruma Sai-Chuen Fu, and Patrick Shu-Hang Yung. Graft healing after anterior cruciate ligament reconstruction (ACLR). *Asia-Pacific Journal of Sports Medicine, Arthroscopy, Rehabilitation and Technology*, 25:8–15, July 2021. ISSN 2214-6873. doi: 10.1016/j.asmart.2021.03.003.

[40] Evan Boyd, Nathan K. Endres, and Andrew G. Geeslin. Postoperative healing and complications based on anterior cruciate ligament reconstruction graft type. *Annals of Joint*, 9:30, 2024. ISSN 2415-6809. doi: 10.21037/aoj-24-3.

[41] Dae-Hee Lee, Jin-Hyuck Lee, Sung-Eun Ahn, and Min-Ji Park. Effect of Time after Anterior Cruciate Ligament Tears on Proprioception and Postural Stability. *PloS One*, 10(9): e0139038, 2015. ISSN 1932-6203. doi: 10.1371/journal.pone.0139038.

[42] Brett A. Bousquet, Luke O'Brien, Steve Singleton, and Michael Beggs. POST-OPERATIVE CRITERION BASED REHABILITATION OF ACL REPAIRS: A CLINICAL COMMENTARY. *International Journal of Sports Physical Therapy*, 13(2):293–305, April 2018. ISSN 2159-2896.

[43] John T. Cavanaugh and Matthew Powers. ACL Rehabilitation Progression: Where Are We Now? *Current Reviews in Musculoskeletal Medicine*, 10(3):289–296, September 2017. ISSN 1935-973X, 1935-9748. doi: 10.1007/s12178-017-9426-3. URL http://link.springer.com/10.1007/s12178-017-9426-3.

[44] Joanna Kvist. Rehabilitation following anterior cruciate ligament injury: current recommendations for sports participation. *Sports Medicine*, 34(4):269–280, 2004. ISSN 0112-1642. doi: 10.2165/00007256-200434040-00006.

[45] Tolga Saka. Principles of postoperative anterior cruciate ligament rehabilitation. *World Journal of Orthopedics*, 5(4):450, 2014. ISSN 2218-5836. doi: 10.5312/wjo.v5.i4.450. URL http://www.wjgnet.com/2218-5836/full/v5/i4/450.htm.

[46] Alan Hamlet. Understanding the Many Types of Electrical Stimulation: A Guide to TENS, NMES, FES, and More - Myolyn, February 2025. URL https://myolyn.com/different-types-of-electrical-stimulation-the-name-game/.

[47] Philip L. Gildenberg. History of Electrical Neuromodulation for Chronic Pain: Table 1. *Pain Medicine*, 7(suppl 1):S7–S13, May 2006. ISSN 1526-2375, 1526-4637. doi: 10.1111/j.1526-4637.2006.00118.x. URL https://academic.oup.com/painmedicine/article-lookup/doi/10.1111/j.1526-4637.2006.00118.x.

[48] Andrzej Krawczyk. Electrostimulation in medicine - history and contemporary usage. *Przegląd Elektrotechniczny*, 1(12):232–235, December 2018. ISSN 0033-2097. doi: 10.15199/48.2018.12.52. URL http://sigma-not.pl/publikacja-117349-2018-12.html.

[49] MYOLYN. Understanding the Various Types of Electrical Stimulation Technology: The Name Game. White paper, MYOLYN. URL https://myolyn.com/wp-content/uploads/2026/06/MYOLYN-A-Systematic-Review-of-Modern-Clinical-Research-on-the-Benefits-of-FES-Cycling-Longform.pdf#gf_8.

[50] Gad Alon. Functional electrical stimulation (FES): The science is strong, the clinical practice not yet – A review of evidence. *Annals of Physical and Rehabilitation Medicine*, 59:e26–e27, September 2016. ISSN 18770657. doi: 10.1016/j.rehab.2016.07.061. URL https://linkinghub.elsevier.com/retrieve/pii/S1877065716301415.

[51] United Healthcare Commercial and Individual Exchange. Electrical Stimulation for the Treatment of Pain and Muscle Rehabilitation, May 2025.

[52] Physiopedia. Transcutaneous Electrical Nerve Stimulation (TENS), 2026. URL https://www.physio-pedia.com/Transcutaneous_Electrical_Nerve_Stimulation_(TENS). Online, [accessed: Jan. 6, 2026].

[53] Mark I. Johnson, Carole A. Paley, Gareth Jones, Matthew R. Mulvey, and Priscilla G. Wittkopf. Efficacy and safety of transcutaneous electrical nerve stimulation (TENS) for acute and chronic pain in adults: a systematic review and meta-analysis of 381 studies (the meta-TENS study). *BMJ Open*, 12(2):e051073, February 2022. doi: 10.1136/bmjopen-2021-051073. URL http://bmjopen.bmj.com/content/12/2/e051073.abstract.

[54] Paul W. Ackermann, Robin Juthberg, and Johanna Flodin. Unlocking the potential of neuromuscular electrical stimulation: achieving physical activity benefits for all abilities. *Frontiers in Sports and Active Living*, Volume 6 - 2024, 2024. ISSN 2624-9367. URL https://www.frontiersin.org/journals/sports-and-active-living/articles/10.3389/fspor.2024.1507402.

[55] Wenjing Shan, Tingting Zheng, Jing Zhang, and Rizhao Pang. Effect of electrical stimulation on functional recovery of lower limbs in patients after anterior cruciate ligament surgery: a systematic review and meta-analysis. *BMJ open*, 15(7):e089702, July 2025. ISSN 2044-6055. doi: 10.1136/bmjopen-2024-089702.

[56] Tiago Coelho-Magalhães, Christine Azevedo Coste, and Henrique Resende-Martins. A Novel Functional Electrical Stimulation-Induced Cycling Controller Using Reinforcement Learning to Optimize Online Muscle Activation Pattern. *Sensors*, 22(23):9126, November 2022. ISSN 1424-8220. doi: 10.3390/s22239126. URL https://www.mdpi.com/1424-8220/22/23/9126.

[57] Cheryl L. Lynch and Milos R. Popovic. Functional Electrical Stimulation. *IEEE Control Systems Magazine*, 28(2):40–50, April 2008. ISSN 1066-033X, 1941-000X. doi: 10.1109/MCS.2007.914689. URL https://ieeexplore.ieee.org/document/4472378/.

[58] I.-Lin Wang, Fei-Fei Lu, Si-Min Lian, Lijuan Chang, and Yi-Ming Chen. Effect of functional electrical stimulation on maximum joint angles and gait asymmetry in female athletes post-anterior cruciate ligament reconstruction when crossing obstacles. *BMC sports science, medicine & rehabilitation*, 18(1):134, January 2026. ISSN 2052-1847. doi: 10.1186/s13102-025-01486-8.

[59] Hani Yousef, Mandy Alhajj, Adegbenro O. Fakoya, and Sandeep Sharma. Anatomy, Skin (Integument), Epidermis. In *StatPearls*. StatPearls Publishing, Treasure Island (FL), 2025. URL http://www.ncbi.nlm.nih.gov/books/NBK470464/.

[60] Zahra Lotfollahi. The anatomy, physiology and function of all skin layers and the impact of ageing on the skin. *Wound Practice and Research*, 32(1), March 2024. ISSN 18376304. doi: 10.33235/wpr.32.1.6-10. URL https://journals.cambridgemedia.com.au/wpr/volume-32-number-1/anatomy-physiology-and-function-all-skin-layers-and-impact-ageing-skin.

[61] John Goodwin Webster. *Medical instrumentation: application and design*. J. Wiley, Hoboken, N.J, 4th ed edition, 2010. ISBN 978-0-471-67600-3.

[62] Koosha Ehtiati, Johannes Eiler, Agnieszka Bochynska, Louis Lech Nissen, Esben Strøbech, Lene Feldskov Nielsen, and Esben Thormann. Skin and Artificial Skin Models in Electrical Sensing Applications. *ACS Applied Bio Materials*, 6(8):3033–3051, August 2023. ISSN 2576-6422, 2576-6422. doi: 10.1021/acsabm.3c00356. URL https://pubs.acs.org/doi/10.1021/acsabm.3c00356.

[63] Fairview Info. Interferential Therapy (IFT) Newsletter, August 2024. URL https://www.fairviewrehabcentre.com/2024/08/31/interferential-therapy-ift-newsletter-august-2024/. [Online; accessed 21-April-2026].

[64] Academo. Wave interference and beat frequency. Academo.org. URL https://academo.org/demos/wave-interference-beat-frequency/. [Online; accessed 9-July-2026].

[65] John Ozcan, Alex R Ward, and Valma J Robertson. A comparison of true and premodulated interferential currents. *Archives of Physical Medicine and Rehabilitation*, 85(3):409–415, March 2004. ISSN 00039993. doi: 10.1016/S0003-9993(03)00478-7. URL https://linkinghub.elsevier.com/retrieve/pii/S0003999303004787.

[66] Veysel Alcan and Murat Zinnuroğlu. Current developments in surface electromyography. *Turkish Journal of Medical Sciences*, 53(5):1019–1031. ISSN 1300-0144. doi: 10.55730/1300-0144.5667. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC10763750/.

[67] BIOPAC Systems, Inc. Introductory EMG Guide. Application Guide, BIOPAC Systems, Inc., April 2019.

[68] Gianluca Bertoni, Gaia Leuzzi, Mirko Job, Marica De Simone, and Marco Testa. Exploring knowledge, perception, and use of surface electromyography in physiotherapy post graduate trainees in Italy: a single center preliminary survey. *Frontiers in Rehabilitation Sciences*, Volume 5 - 2024, 2024. ISSN 2673-6861.

[69] Philip A. Parker and Roberto Merletti, editors. *Electromyography: physiology, engineering, and noninvasive applications*. IEEE Press series in biomedical engineering. Wiley-Interscience, Hoboken, N.J, 2004. ISBN 978-0-471-67580-8 978-0-471-67838-0 978-1-60119-509-8. doi: 10.1002/0471678384.

[70] David N. Levine. Sherrington's "The Integrative action of the nervous system": a centennial appraisal. *Journal of the Neurological Sciences*, 253(1-2):1–6, February 2007. ISSN 0022-510X. doi: 10.1016/j.jns.2006.12.002.

[71] Arnault H. Caillet, Andrew T.M. Phillips, Luca Modenese, and Dario Farina. NeuroMechanics: Electrophysiological and computational methods to accurately estimate the neural drive to muscles in humans in vivo. *Journal of Electromyography and Kinesiology*, 76:102873, June 2024. ISSN 1050-6411. doi: 10.1016/j.jelekin.2024.102873.

[72] Sid Gilman and Ann Arbor. Handbook of physiology. Section 1: The nervous system, vol II. Motor control, parts 1 and 2. Section editors: John M. Brookhart and Vernon B. Mountcastle volume editor: Vernon B. Brooks Bethesda, MD, American Physiological Society, 1981 1480 pp, illustrated. *Annals of Neurology*, 13(1):111–111, January 1983. ISSN 0364-5134, 1531-8249. doi: 10.1002/ana.410130130.

[73] R. Merletti and G.L. Cerone. Tutorial. Surface EMG detection, conditioning and preprocessing: Best practices. *Journal of Electromyography and Kinesiology*, 54:102440, October 2020. ISSN 10506411. doi: 10.1016/j.jelekin.2020.102440. URL https://linkinghub.elsevier.com/retrieve/pii/S1050641120300821.

[74] Glen Brisebois. Op amp selection guide for optimum noise performance. In *Analog Circuit Design*, pages 907–908. Elsevier, 2015. ISBN 978-0-12-800001-4. doi: 10.1016/B978-0-12-800001-4.00422-1.

[75] NF corporation. Equivalent Input Noise, March 2020. URL https://www.nfcorp.co.jp/english/techinfo/technicalvocabulary/equivalent_input_noise.html. [Online; accessed 27-August-2026].

[76] Jifeng Qu, Samuel P. Benz, Kevin Coakley, Horst Rogalla, Weston L. Tew, D. Rod White, Kunli Zhou, and Zhenyu Zhou. An improved electronic determination of the Boltzmann constant by Johnson noise thermometry. *Metrologia*, 54(4):549–558, Aug 2017. doi: 10.1088/1681-7575/aa781e.

[77] Marek Lis. Trade-offs between CMOS, JFET, and bipolar input stage technology. Application Report SBOA355, Texas Instruments Inc., May 2019. URL https://www.ti.com/lit/an/sboa355/sboa355.pdf.

[78] D. T. Mewett, K. J. Reynolds, and H. Nazeran. Reducing power line interference in digitised electromyogram recordings by spectrum interpolation. *Medical and Biological Engineering and Computing*, 42(4):524–531, July 2004. ISSN 0140-0118, 1741-0444. doi: 10.1007/BF02350994. URL http://link.springer.com/10.1007/BF02350994.

[79] Bethel A. C. Osuagwu, Emily Whicher, and Rebecca Shirley. Active proportional electromyogram controlled functional electrical stimulation system. *Scientific Reports*, 10(1):21242, December 2020. ISSN 2045-2322. doi: 10.1038/s41598-020-77664-0. URL https://doi.org/10.1038/s41598-020-77664-0.

[80] Y. Chen, C. Dai, and W. Chen. A Real-time EMG-controlled Functional Electrical Stimulation System for Mirror Therapy. In *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4, October 2019. ISBN 2163-4025. doi: 10.1109/BIOCAS.2019.8919069. Journal Abbreviation: 2019 IEEE Biomedical Circuits and Systems Conference (BioCAS).

[81] Rust. ECG Signal Conditioner – Electronic Circuit Diagram. https://freecircuitdiagram.com/1694-ecg-signal-conditioner/, March 2010. [Online; accessed 19-June-2026].

[82] Burr-Brown Products from Texas Instruments. INA326, 327: Precision, Rail-To-Rail I/O Instrumentation Amplifier datasheet (Rev. D), November 2004. URL https://www.ti.com/lit/ds/symlink/ina326.pdf?ts=1782040843941.

[83] INA326EA/250 Texas Instruments | Mouser. URL https://pt.mouser.com/pt/ProductDetail/Texas-Instruments/INA326EA-250?qs=VBduBm9rCJQqO1eipsjOZw%3D%3D. [Online; accessed 25-August-2026].

[84] Fairchild Linear Integrated Circuits. TDA2002·TDA2002A 8 watt audio power amplifiers.

[85] Douglas Self. *Audio power amplifier design handbook*. Newnes, Oxford Boston, 4th ed edition, 2006. ISBN 978-0-7506-8072-1.

[86] TME. TST 10/020 INDEL - Transformer: toroidal | 10VA; 230VAC; 24V; 24V; 0.21A; 0.21A; Ø: 57mm; TST10W/2X24V | Transfer Multisort Elektronik. URL https://www.tme.eu/en/details/tst10w_2x24v/toroidal-transformers/indel/tst-10-020/. [Online; accessed 12-April-2026].

[87] IEC 60601-1: Medical electrical equipment—Part 1: General requirements for basic safety and essential performance, December 2005. URL https://webstore.iec.ch/en/publication/2606.

[88] IEC 61558-2-15: Safety of transformers, reactors, power supply units and combinations thereof—Part 2-15: Particular requirements and tests for isolating transformers for medical IT systems for the supply of medical locations, May 2022. URL https://webstore.iec.ch/en/publication/75789.

[89] C. Baumgartner, P. A. Hasgall, F. Di Gennaro, E. Neufeld, B. Lloyd, M. C. Gosselin, D. Payne, A. Klingenböck, and N. Kuster. IT'IS Database for Thermal and Electromagnetic Parameters of Biological Tissues, August 2025. URL https://itis.swiss/virtual-population/tissue-properties/overview.

[90] C Gabriel, S Gabriel, and E Corthout. The dielectric properties of biological tissues: I. Literature survey. *Physics in Medicine and Biology*, 41(11):2231–2249, November 1996. ISSN 0031-9155, 1361-6560. doi: 10.1088/0031-9155/41/11/001. URL https://iopscience.iop.org/article/10.1088/0031-9155/41/11/001.

[91] Udayan Das, Abhay Tyagi, Akshay Maharia, Barsha B Parida, Ganesh Rajesh Pande, Vedaant Parekh, Debashish Mishra, and Ipsita Debata. Comparison of Bioelectrical Impedance Parameters in Patients With Knee Osteoarthritis: A Case-Control Study. *Cureus*, November 2025. ISSN 2168-8184. doi: 10.7759/cureus.96766. URL https://www.cureus.com/articles/429915-comparison-of-bioelectrical-impedance-parameters-in-patients-with-knee-osteoarthritis-a-case-control-study.

[92] Matthias Geier. {python-sounddevice: Play and Record Sound with Python, January 2026. URL https://github.com/spatialaudio/python-sounddevice/.

[93] Qt for Python Team. PySide6: Python bindings for the Qt cross-platform application and UI framework, May 2026. URL https://doc.qt.io/qtforpython-6/index.html. Software package.

[94] STMicroelectronics. TDA2005 - 20 W bridge/stereo amplifier for car radio, September 2013. URL https://www.st.com/resource/en/datasheet/tda2005.pdf.

[95] Alex R. Ward and Nataliya Shkuratova. Russian electrical stimulation: the early experiments. *Physical Therapy*, 82(10):1019–1030, 2002. doi: 10.1093/ptj/82.10.1019. URL https://pubmed.ncbi.nlm.nih.gov/12350217/.

[96] Mario Cifrek, Vladimir Medved, Stanko Tonković, and Saša Ostojić. Surface EMG based muscle fatigue evaluation in biomechanics. *Clinical Biomechanics*, 24(4):327–340, 2009. doi: 10.1016/j.clinbiomech.2009.01.010. URL https://pubmed.ncbi.nlm.nih.gov/19285766/.