ELEC5305 Project Proposal

Project Title

Explore LUFS and Standards for Broadcast and Streaming Audio

Student Information

Full Name:Jianxiang Chen

Student ID (SID): 540062964

GitHub Username: cjx259

GitHub Project Link: https://cjx259-au.github.io/elec5305-project-540062964/

Project Overview

This project investigates how programme loudness (LUFS/LKFS) and true-peak limits are measured and enforced across broadcast and major streaming contexts. Using the ITU-R BS.1770-5 algorithm as ground truth, we will implement or call MATLAB tools to compute integrated, short-term and momentary loudness, loudness range (LRA), and true peak (dBTP). We then perform standards compliance checks for EBU R128 (Europe; target −23 LUFS, true-peak ≤ −1 dBTP) and ATSC A/85 (US; target −24 LKFS) on a curated set of voice, music and advertising-style clips. Finally, we emulate platform loudness normalization (e.g., a teaching reference around −14 LUFS commonly cited for streaming) to study how starting loudness and dynamics survive downstream normalization and peak limiting. The goal is to deliver a reproducible toolbox and a practical set of recommendations for content creators and engineers.

Background and Motivation

Perceived loudness mismatches between programmes and commercials led to international standards that define how to measure and control programme loudness. The ITU-R BS.1770 family specifies K-weighting, multi-channel channel weighting, and absolute/relative gating to compute loudness in LUFS/LKFS. Regional documents such as EBU R128 and ATSC A/85 set operating targets and permitted maximum true-peak levels. Meanwhile, consumer platforms apply their own loudness normalization policies so that tracks of different masters play at roughly similar loudness; however, details and targets vary over time and are not always publicly fixed. For students in audio, understanding the measurement pipeline (what is LUFS?), the regulatory context (what must broadcasters meet?), and the platform behavior (what happens to a −12 LUFS or −20 LUFS master online?) is both academically relevant and immediately practical.

Proposed Methodology

(1) Audio material

We assemble ~1–2 hours of short clips (10–60 s each) spanning:

Speech (audiobook/news-style narration),

Music (creative-commons stems/finished songs across genres),

Promos/ads (synthetic or licensed short spots with stronger transients).
All material is mono or stereo at 48 kHz for broadcast analysis; a parallel 44.1/48 kHz copy is kept for streaming emulation. Metadata includes content type, duration, initial true-peak headroom, and crest factor.

(2) Loudness & true-peak measurement (MATLAB)

We will (a) use Audio Toolbox meters where available and (b) implement a transparent reference pipeline for auditing:

K-weighting filter (high-shelf + high-pass) designed via `designfilt`; response verified with `fvtool`.

Channel weighting & summation per BS.1770-5; stereo downmix if needed.

Gating: absolute and relative gates applied on 400 ms blocks to compute Integrated LUFS; Short-term (3 s) and Momentary (400 ms) tracked for time-varying plots.

LRA computed from the distribution of gated short-term loudness.

True peak (dBTP) estimated via 4× (and sensitivity check at 8×) oversampling to capture inter-sample peaks.
The meter is validated on reference signals and public test excerpts; acceptance tolerance ≤ 0.2 LU vs. known values.

(3) Broadcast compliance checking

For each clip we produce a compliance report against EBU R128 and ATSC A/85:

Targets: −23 LUFS (R128) and −24 LKFS (A/85).

Permitted maximum true peak: recommend ≤ −1 dBTP (R128) and platform-specific guidance for A/85 deliverables.

Outputs per clip: Integrated LUFS, LRA, dBTP, ΔLU to target, over-TP risk, and a suggested corrective gain (LU) to meet the target without clipping.
We also batch-render "before/after" versions applying only gain to meet the loudness target, verifying that true-peak remains within limits; if not, we demonstrate the impact of adding a transparent limiter vs. allowing headroom.

(4) Streaming normalization emulation

To illustrate platform behavior, we implement a teaching-oriented normalization model around −14 LUFS (commonly cited for several services, subject to change). For each source master with different starting loudness (e.g., −12/−14/−16/−20 LUFS):

Apply gain to hit the reference playback loudness.

If dBTP would exceed −1 dBTP at playback, emulate a limiter (soft vs. hard knee) and quantify artifacts.

Compare pre/post metrics: playback LUFS trajectory, dBTP, gain applied, LRA, crest factor, log-spectral distance, and over-zero/clip incidence.
This produces intuitive plots of "what the listener hears" and trade-offs between loudness, headroom, and dynamics.

(5) Statistics, visualization, and reproducibility

Each analysis exports CSV tables (`results/`) and figures (`figures/`).

We compute mean ± 95% CI for key metrics per content type, and paired tests (or bootstrap) for before/after comparisons.

Visuals: time-aligned plots of short-term/momentary LUFS, histograms of LRA, scatter of ΔLU vs. dBTP risk, and waveform snapshots showing limiter action.

All parameters are centralized in `config.m` (targets, gates, oversampling factor, limiter settings) for one-click reproducibility.

Expected Outcomes

A validated LUFS/true-peak meter (scripts and/or MATLAB workflow) that matches BS.1770 expectations within tight tolerance.

A broadcast compliance dashboard: per-clip pass/fail, recommended gain to target, and true-peak safety analysis.

A streaming normalization study that shows how different starting masters fare after playback normalization, with evidence-based guidance (e.g., “masters around −16 LUFS with adequate headroom tend to preserve dynamics with fewer limiter artifacts under −14 LUFS playback”).

Public GitHub repository and GitHub Pages site including methodology, tables, figures, and sample audio where licensing permits.

Timeline (Weeks 6–13)

Weeks 6–7: Literature review (BS.1770-5, EBU R128, ATSC A/85). Build/validate MATLAB LUFS+TP meter on references; set up repo and CI scripts.

Weeks 8–9: Run broadcast compliance on the full clip set; generate tables and first round of visuals; iterate on tolerance and edge cases (quiet speech, dense mixes).

Weeks 10–11: Implement streaming normalization emulation (−14 LUFS reference); conduct master-loudness sweep and limiter sensitivity tests; complete statistics.

Weeks 12–13: Write the final report, polish figures, and finalize GitHub documentation and demo pages.

References

ITU-R BS.1770-5 (2023), *Algorithms to measure audio programme loudness and true-peak audio level*.

EBU R128 (v5.0, 2023), *Loudness normalisation and permitted maximum level of audio signals*.

ATSC A/85 (2021), *Techniques for Establishing and Maintaining Audio Loudness for Digital Television*.

Supplementary industry documentation on streaming loudness normalization (used for emulation only; policies may change over time).