ACC RPM Monitor - Primary Algorithms

Technical Documentation

October 27, 2025

1 Introduction

This document describes the primary algorithms used in ACC RPM Monitor for shift point optimization, lap performance analysis, and audio feedback generation. The application operates in three distinct modes: Standard (fixed shift points), Adaptive (continuous 15-second learning), and Performance Learning (machine learning-based optimization).

Contents

2 Shift Pattern Detection Algorithm

2.1 Overview

The Shift Pattern Detection algorithm monitors real-time telemetry data to detect gear changes and correlate them with lap performance metrics. This forms the foundation for learning optimal shift points.

2.2 Algorithm: Shift Event Detection

Algorithm 1 Shift Event Detection

```
Input: current_gear, previous_gear, current_rpm, previous_rpm, throttle, speed
Output: shift_detected (boolean)
if current_gear ≠ previous_gear then
    // Check for valid shift
    if current_gear ≥ 1 and previous_gear ≥ 1 then
        is_upshift ← current_gear > previous_gear
        if is_upshift then
            if throttle ≥ 0.3 and previous_rpm ≥ 3000 then
                return true
            end if
        else
            if current_rpm ≥ 3000 then
                return true
            end if
        end if
    end if
end if
return false
```

Parameters:

- `MinThrottleForUpshift` = 0.3 (30% throttle required for upshift to count)
- `MinRPMForShift` = 3000 RPM (filters engine braking/neutral shifts)

3 Lap Performance Analysis

3.1 Overview

The Lap Performance Analysis algorithm tracks lap completion, validates laps based on off-track metrics, and associates shift events with lap outcomes to enable performance correlation.

3.2 Algorithm: Lap Validity Determination

Key Parameters:

- Off-track time threshold: 3.0 seconds cumulative (changed from 2.0 in v3.4.0)
- Off-track detection threshold: 0.5f = 50% vertical position delta from racing line

Algorithm 2 Lap Validity Determination

```
Input: lap_time_ms, off_track_time_sec, is_valid_by_acc
Output: lap_valid (boolean)
THRESHOLD_TIME ← 3.0 // seconds cumulative off-track
THRESHOLD_RATIO ← 0.5 // 50% of track considered off-track
// Primary check: ACC validity flag
is_valid_acc ← is_valid_by_acc
// Secondary check: Metrics-based validation
is_valid_metrics ← (lap_time_ms > 0) and
(lap_time_ms < INT_MAX) and
(off_track_time_sec < THRESHOLD_TIME)
// Both checks must pass
return is_valid_acc and is_valid_metrics
```

- Validation requires both ACC flag **and** metrics checks to pass

4 Optimal Shift Point Analysis

4.1 Overview

The Optimal Shift Point Analysis algorithm groups shift events by RPM buckets, correlates them with lap performance, and identifies the RPM bracket that produces the best average lap times.

4.2 Algorithm: RPM Bucket Performance Analysis

Scoring Function:

$$\text{Score} = \overline{\text{LapTime}} + (1000 \times \overline{\text{OffTrackTime}})$$

This composite score weighs lap time (primary) and off-track time (secondary) to find the shift point that maximizes performance while maintaining track control.

5 Adaptive Learning Algorithm

5.1 Overview

The Adaptive Learning algorithm continuously updates shift points every 15 seconds based on real-time throttle and RPM data. It provides fast feedback for drivers who want quick optimization without waiting for full lap analysis.

5.2 Algorithm: Continuous Shift Point Update

Data Collection Criteria:

- Throttle $\geq 85\%$: Ensures acceleration phase data
- Speed > 5 km/h: Filters standing starts and slow sections
- RPM rising at > 100 RPM/sec: Captures actual acceleration
- Update every 15 seconds: Balances responsiveness with statistical significance

Algorithm 3 Optimal Shift Point Analysis

```
Input: shift_events (list of ShiftEvent), lap_performances (list of LapPerformance)
Output: optimal_shift_points (Dictionary of gear to RPM)
MIN_VALID_LAPS ← 2 // for report generation
MIN_SHIFTS_PER_GEAR ← 5 // samples needed per gear
RPM_BUCKET_SIZE ← 200 // RPM grouping granularity
for each gear g from 1 to 6 do
    valid_laps ← filter(lap_performances, is_valid = true)
    if count(valid_laps) < MIN_VALID_LAPS then
        continue
    end if
    gear_shifts ← filter(shift_events, gear = g, is_upshift = true)
    if count(gear_shifts) < MIN_SHIFTS_PER_GEAR then
        continue
    end if
    // Group shifts by RPM buckets
    buckets ← group_by(gear_shifts, rpm)
    // Calculate composite score for each bucket
    for each bucket b in buckets do
        if count(b) ≥ 2 then
            avg_lap_time ← mean(lap_times in b)
            avg_off_track ← mean(off_track_times in b)
            score ← avg_lap_time + (avg_off_track × 1000)
            if score is minimum then
                optimal_shift_points[g] ← b.rpm
            end if
        end if
    end for
end for
return optimal_shift_points
```

Algorithm 4 Adaptive Shift Point Learning

```
Input: telemetry stream (real-time data), update interval = 15 seconds
Output: updated shift points (periodic updates)
THROTTLE_THRESHOLD ← 0.85 // 85 percent
SPEED_THRESHOLD ← 5.0 // km/h minimum
RPM_RATE_THRESHOLD ← 100 // RPM/sec rising
last_update ← current_time
collected_data ← empty list
while true do
    for each telemetry frame do
        if throttle ≥ THROTTLE_THRESHOLD and speed > SPEED_THRESHOLD and rpm_rate > RPM_RATE_THRESHOLD then
            collect(current_rpm, current_gear, lap_position)
        end if
        if current_time − last_update ≥ update_interval then
            shift_points ← analyze(collected_data)
            apply(shift_points)
            emit(AudioFeedback)
            collected_data ← clear
            last_update ← current_time
        end if
    end for
end while
```

6 Audio Feedback System

6.1 Overview

The Audio Feedback System generates real-time pitch-based guidance indicating whether the driver should shift earlier (high pitch), at the optimal point (neutral pitch), or later (low pitch). Two audio profiles are available: Normal (responsive) and Endurance (low-fatigue).

6.2 Algorithm: Tone Profile Selection

Audio Feedback Interpretation:

- **High Pitch**: Current RPM is too low (shifted too early) - shift later next time
- **Neutral Pitch**: Current RPM is in optimal range - shifting at correct point
- **Low Pitch**: Current RPM is too high (shifted too late) - shift earlier next time

Algorithm 5 Audio Feedback Tone Selection

```
Input: current_rpm, recommended_shift_rpm, audio_profile
Output: tone_profile (ToneProfile object)
OPTIMAL_BAND ← 175 // RPM band around recommended point
// Calculate RPM distance from recommended
rpm_delta ← current_rpm − recommended_shift_rpm
if rpm_delta > OPTIMAL_BAND then
    // Shifting too late: use low pitch to indicate shift earlier
    if audio_profile == Normal then
        return ToneProfile(400 Hz, 0 Hz, 150 ms)
    else
        return ToneProfile(400 Hz, -15 Hz glide, 120 ms)
    end if
else if rpm_delta < −OPTIMAL_BAND then
    // Shifting too early: use high pitch to indicate shift later
    if audio_profile == Normal then
        return ToneProfile(950 Hz, +10 Hz glide, 100 ms)
    else
        return ToneProfile(650 Hz, +10 Hz glide, 60 ms)
    end if
else
    // Optimal range: use neutral mid pitch
    if audio_profile == Normal then
        return ToneProfile(600 Hz, no glide, 140 ms)
    else
        return ToneProfile(500 Hz, no glide, 130 ms)
    end if
end if
```

Algorithm 6 Audio Stop Decision Logic

```
Input: rpm_rate (RPM/sec), tone_duration_elapsed, tone_duration_limit
Output: should_stop_audio (boolean)
RPM_RATE_THRESHOLD ← 50 // RPM/sec
if rpm_rate < RPM_RATE_THRESHOLD then
    // Driver stopped accelerating
    return true
end if
if tone_duration_elapsed ≥ tone_duration_limit then
    // Tone natural duration expired
    return true
end if
return false
```

Algorithm 7 Calculate RPM Rate of Change

```
Input: rpm_history (queue of rpm, timestamp pairs)
Output: rpm_rate (RPM/sec)
WINDOW_MS ← 200 // Lookback window
current_time ← now
window_start ← current_time − WINDOW_MS
// Remove old entries outside window
while rpm_history.front().timestamp < window_start do
    remove(rpm_history.front())
end while
if count(rpm_history) < 2 then
    return 0
end if
// Calculate rate over window
rpm_delta ← rpm_history.back().rpm − rpm_history.front().rpm
time_delta ← (rpm_history.back().timestamp − rpm_history.front().timestamp) / 1000
return rpm_delta / time_delta
```

6.3 Algorithm: Intelligent Audio Stop Condition

7 RPM Rate Calculation

7.1 Algorithm: Rolling RPM Rate

8 Data Quality Metrics

8.1 Algorithm: Learning Rate Calculation

9 Performance Metrics

9.1 Off-Track Detection

The application detects off-track events using normalized position delta:

$$\text{is\_off\_track} = |\Delta\text{normalized\_position}| > 0.5$$

This corresponds to approximately 50% distance from the racing line, with cumulative tracking across the lap.

9.2 Lap Time Calculation

Lap times are measured in milliseconds and include all sectors from S/F line crossing to the next crossing. Incomplete laps or laps exceeding a logical maximum are marked invalid.

9.3 Shift Quality Metrics

For each shift, the following metrics are tracked:

- **Shift RPM**: Engine speed at moment of gear change

Algorithm 8 Calculate Learning Quality Score

```
Input: valid_laps, total_shifts, data_consistency
Output: quality_score (0.0 to 1.0)
EXCELLENT_THRESHOLD ← 5 // laps
GOOD_THRESHOLD ← 3 // laps
// Base score from lap count
if valid_laps ≥ EXCELLENT_THRESHOLD then
    lap_score ← 1.0
else if valid_laps ≥ GOOD_THRESHOLD then
    lap_score ← 0.7
else
    lap_score ← valid_laps / GOOD_THRESHOLD
end if
// Consistency bonus from shift variance
if shift_variance is low then
    consistency_bonus ← 0.2
else
    consistency_bonus ← 0.0
end if
return min(1.0, lap_score + consistency_bonus)
```

- **Post-Shift RPM**: Engine speed immediately after shift completes
- **Speed Loss**: Velocity change during shift (indicator of smooth execution)
- **Throttle Position**: Percent throttle during shift (70%+ indicates aggressive driving)

10 Audio Profile Specifications Table

11 Summary

The ACC RPM Monitor employs a multi-layered approach to shift point optimization:

1. **Real-time detection** via the Shift Event Detection algorithm
2. **Statistical analysis** through RPM Bucket Performance Analysis
3. **Continuous learning** with the Adaptive algorithm (15-second updates)
4. **Intuitive feedback** via the Audio Feedback System with dual profiles
5. **Quality validation** ensuring only meaningful data influences recommendations

These algorithms work in concert to provide drivers with scientifically-grounded shift point recommendations that maximize lap performance while maintaining vehicle control.

Audio feedback uses an intuitive pitch-based system: high pitch indicates shifting too early (shift later), neutral pitch indicates optimal shifting, and low pitch indicates shifting too late (shift earlier).

Parameter	**Too Early (High)**	**Optimal (Neutral)**	**Too Late (Low)**
		Normal Profile	
Frequency (Hz)	950	600	400
Duration (ms)	130	140	150
Attack (ms)	5	5	5
Decay (ms)	120	135	145
Sustain Level	60%	55%	50%
dB Level	-3	0	-2
Glide	+10 Hz/100ms	None	None
Waveform	Rounded	Sine	Triangle
		Endurance Profile	
Frequency (Hz)	650	500	400
Duration (ms)	110	130	140
Attack (ms)	8	10	8
Decay (ms)	130	100	160
Sustain Level	57%	52%	48%
dB Level	-3	0	-3
Glide	+10 Hz/60ms	None	-15 Hz/120ms
Waveform	Sine	Sine	Sine

Table 1: Audio Profile Specifications (v3.4.0) - Column headers show RPM situation relative to recommended shift point