# THERMAL-OPTIMIZATION COUPLING ANALYSIS

## Field Manual for Energy Systems Engineer

| | |
|---|---|
| Document Title: | Thermal-Optimization Coupling Analysis |
| Document Type: | Field Manual |
| Revision: | 1.0 |
| Date: | 2025-10-28 |
| Author: | RLE Scientific Instrument |
| Classification: | Technical Documentation |
| Target Audience: | Energy Systems Engineers |

**SAFETY NOTICE:**

Exercise caution when handling electrical loads and thermal sensors.
Ensure proper grounding and follow electrical safety protocols.
Monitor system temperatures to prevent thermal damage.

DOCUMENT CONTROL:

Prepared by: RLE Scientific Instrument
Reviewed by: _________________
Approved by: _________________

This document provides procedures for measuring thermal-optimization coupling in AI training workloads using RLE monitoring systems.


# THERMAL-OPTIMIZATION COUPLING ANALYSIS

## Field Manual for Energy Systems Engineer


## PURPOSE

This procedure measures the coupling between computational optimization dynamics and thermal system response in AI training workloads. We quantify the energy balance between optimization instability (gradient norm spikes) and thermal efficiency degradation (heat transfer rate changes) to enable predictive thermal management.

Why this matters: Traditional thermal monitoring detects problems after they occur. This procedure predicts thermal instability 1 second before collapse, enabling proactive energy management.

## HEAT-TRANSFER ANALOGUE MAPPING

| RLE Term | Heat-Transfer Analogue | Description |
|---|---|---|
| η_util | System efficiency | Fraction of input power converted to useful work |
| η_stability | hA stability | Effective heat-transfer coefficient stability |
| A_load | m■ or velocity term | Represents convective intensity / load acceleration |
| τ_sustain | τ_th | Thermal time constant (transient response) |

Translation Note: The RLE formulation maps directly to classical heat-transfer analysis. Use this table to connect the computational metrics to familiar thermal systems concepts.

## SYSTEM BOUNDARIES AND VARIABLES

### *Primary Measurements*

| Variable | Symbol | Units | Sensor | Range |
|---|---|---|---|---|
| Thermal Efficiency | RLE | dimensionless | Computed | 0.0 - 1.0 |
| Gradient Norm | ∇f | dimensionless | Training log | 0 - 50 |
| Heat Transfer Rate | Q■ | W | Power sensor | 0 - 300W |
| Temperature Rise | ΔT | °C | Thermocouple | 0 - 50°C |
| Time Constant | τ | s | Computed | 1 - 10s |

### *Derived Quantities*

| Variable | Symbol | Units | Calculation |
|---|---|---|---|
| Thermal Time Constant | τ_th | s | ΔT / (dT/dt) |
| Power Efficiency | η_p | dimensionless | Q■_useful / Q■_total |
| Optimization Instability | σ_opt | dimensionless | std(∇f) / mean(∇f) |
| Thermal Lag | t_lag | s | Cross-correlation peak |

## ENERGY BALANCE EQUATIONS

### *Primary Conservation Equation*

```
RLE = (η_util × η_stability) / (A_load × (1 + 1/τ_sustain))
```

Where: η_util = utilization efficiency (dimensionless), η_stability = thermal stability factor (dimensionless), A_load = load acceleration (dimensionless), τ_sustain = thermal sustainment time (s)

### *Heat Transfer Rate*

```
Q■ = hA × ΔT + m■ × cp × dT/dt
```

Where: hA = heat transfer coefficient × area (W/°C), ΔT = temperature difference (°C), m■ = mass flow rate (kg/s), cp = specific heat capacity (J/kg·°C)

### *Optimization-Thermal Coupling*

```
dRLE/dt = -k × ∇f(t - t_lag)
```

Where: k = coupling coefficient (s■¹), t_lag = thermal response lag (s)

## PROCEDURE

### *Step 1: System Initialization*

- Verify all sensors: GPU temp, power, utilization
- Initialize training workload: DistilGPT-2, 200 steps
- Set sampling rate: 1 Hz
- Record ambient conditions: T_ambient = 21°C

### *Step 2: Synchronized Data Collection*

- Start thermal monitoring: hardware_monitor_v2.py --mode both --sample-hz 1
- Start optimization logging: extended_training_with_sync.py
- Run for 90 seconds (steady-state thermal response)
- Stop both processes simultaneously

### *Step 3: Data Alignment*

- Align timestamps using shared clock reference
- Merge thermal and optimization data streams
- Verify alignment tolerance: ±2 seconds
- Record aligned sample count: N_samples

### *Step 4: Correlation Analysis*

- Calculate cross-correlation: RLE vs ∇f
- Perform lag analysis: ±3 second window
- Identify peak correlation: r_peak at t_lag
- Determine causal order: ∇f → RLE or RLE → ∇f

### *Step 5: Energy Balance Validation*

- Check thermal time constant: τ_th = ΔT / (dT/dt)
- Verify heat transfer rate: Q■ = P_measured
- Calculate efficiency drop: ΔRLE = RLE_max - RLE_min
- Confirm energy conservation: Q■_in = Q■_out + Q■_stored

# DATA SHEET TEMPLATE

## *Session Information*

| | |
|---|---|
| Session ID: ________________ | Date: ________________ |
| Duration: ________________ s | Ambient Temp: ________________ °C |
| Model: ________________ | |

# INTERPRETATION GUIDELINES

## *Thermal Response Analysis*

• If t_lag < 0: ∇f spikes precede RLE drops → Optimization instability causes thermal degradation

• If t_lag > 0: RLE drops precede ∇f spikes → Thermal throttling affects optimization

• If t_lag ≈ 0: Simultaneous coupling → Strong thermal-optimization coupling

## *Energy Efficiency Assessment*

• If r_peak > 0.5: Strong thermal-optimization coupling → Predictive capability confirmed

• If 0.3 < r_peak < 0.5: Moderate coupling → Requires controlled conditions

• If r_peak < 0.3: Weak coupling → Insufficient thermal stress or measurement issues

## *Thermal Time Constant*

• If τ_th < 2s: Fast thermal response → High thermal coupling

• If 2s < τ_th < 5s: Normal thermal response → Standard thermal behavior

• If τ_th > 5s: Slow thermal response → Thermal isolation or large thermal mass

## *Power Efficiency*

• If η_p > 0.8: High efficiency → Optimal thermal management

• If 0.6 < η_p < 0.8: Moderate efficiency → Room for improvement

• If η_p < 0.6: Low efficiency → Thermal management issues

# COMMAND REFERENCE

## *Quick Start*

# Single session analysis

```
python run_joint_session.py --model distilgpt2 --duration 90 --output results/
```

# Reproducibility analysis

```
python analysis/reproducibility_analysis.py
```

# Calibration dataset

```
python calibration_dataset.py
```

## NOTES FOR ENGINEER

This procedure measures the energy balance between computational optimization and thermal system response. The key insight is that optimization instability (high gradient norms) creates thermal stress that degrades efficiency before temperature limits are reached.

Practical application: Use this to implement predictive thermal management in AI training systems. When gradient norms spike, reduce workload intensity 1 second before thermal collapse occurs.

Validation: Run multiple sessions under identical conditions to establish reproducibility. The coupling should be consistent within ±20% across sessions.

Extension: This procedure can be adapted for any thermal system with computational workload coupling (mobile devices, edge computing, data centers).