ARFLO RESEARCH / TECHNICAL PAPER 003

Thermal estimation on the measurement clock.

arflo GmbH
4 November 2026 / Edition 1.0

Abstract

Power meters and temperature sensors rarely report on a common clock. This paper develops an asynchronous estimator for a first-order thermal node, using exact propagation between measurement times and a measurement update only when new information arrives. A worked example separates the deterministic error caused by stale power data from the stochastic covariance of the state estimate. Delayed measurements require chronological replay or an explicit out-of-sequence method. The result is an analytical design framework, with stated assumptions and reproducible calculations, rather than a claim about a particular installation.

1 Three times belong to every observation.

A sensor record has a measurement time, an arrival time and a processing time. A value sampled at 12:00:00 and received at 12:00:08 describes the earlier state. Giving it the arrival timestamp shifts information eight seconds into the future. Faster networking reduces this gap, but cannot repair a missing device timestamp or an unknown clock offset.

Store the physical unit, measurement interval, device timestamp, arrival timestamp and quality flags with the value. Distinguish instantaneous power from energy averaged over an interval. The observation model for an interval average must represent that average; it cannot automatically be treated as an instantaneous endpoint sample. Synchronization uncertainty is itself part of the error budget.

2 Propagate the thermal state over the actual interval.

Let θ = T - T_in denote temperature rise above a fixed inlet temperature. A lumped node with thermal capacity C and thermal resistance R obeys C·dθ/dt = P - θ/R. Here C has units J/K, R has units K/W, and the time constant τ = RC has units seconds. P is the heat input represented by the node. A varying inlet requires an additional measured input or an augmented state.

For a power input held constant over Δt, integration gives an exact transition. This avoids the artificial dynamics introduced by advancing every record with a nominal sampling period. The equations fit the state-space filtering framework [1, 2]; the thermal specialization below follows directly from the energy balance. If power changes inside the interval, split propagation at its known change times.

$$F(\Delta t) = e^{-\Delta t/\tau}, \qquad \hat{\theta}^{-} = F\hat{\theta}^{+} + R(1-F)P \tag{1}$$

The superscripts denote predicted and corrected estimates. The input coefficient R(1 - F) has units K/W.

Research origin: Memo 054 (2026-04-09). Online edition: `https://arflo.io/research/timestamp-aware-thermal-estimation/`.

3 Uncertainty propagation and measurement updates.

Represent unresolved temperature disturbances as additive white noise with intensity q in K^2/s. Let V be the scalar state-error variance in K^2. Exact integration yields the process-noise contribution below. The exponent is squared in the propagated variance, and the noise term depends on elapsed time. A fixed covariance increment per packet would make uncertainty depend on network traffic rather than on the physical model.

For a direct temperature-rise measurement y with independent zero-mean noise variance r, the innovation is y - θ^- and the scalar gain is $K = V^-/(V^- + r)$. Apply this correction once per independent observation. Repeating an unchanged cached value as if it were a fresh measurement creates false confidence. Correlated sensors require a joint observation covariance or another treatment of their shared errors.

$$V^- = F^2 V^+ + \frac{q\tau}{2}(1 - F^2) \tag{2}$$

This covariance describes the assumed stochastic disturbance model; it does not by itself cover an omitted load step.

$$\hat{\theta}^+ = \hat{\theta}^- + K(y - \hat{\theta}^-), \qquad V^+ = (1 - K)V^- \tag{3}$$

All terms refer to the measurement timestamp. These scalar formulas assume independent observation noise.

4 Worked example: an unreported power step.

Choose R = 0.05 K/W and C = 3,600 J/K, giving $\tau = 180$ s. Initially the node is at equilibrium with 500 W, so $\theta = 25$ K. At time zero the heat input rises to 800 W. An estimator that continues to hold 500 W predicts $\theta = 25$ K, while the actual model approaches the new equilibrium of 40 K. The resulting error is positive and deterministic.

After 30 s the missed step contributes 2.303 K of temperature underestimation. This number is neither a universal latency allowance nor a measured sensor error. It depends on the selected resistance, heat capacity and load step. It demonstrates why data age must be interpreted through dynamics. Relative percentage error on Celsius temperature is unsuitable as a general safety criterion, because changing the temperature origin changes that percentage.

A separate stochastic calculation uses q = 0.01 K^2/s, an initial variance of 0.25 K^2 and an elapsed interval of 30 s. The propagated variance is 0.4343 K^2. A new independent temperature observation with noise variance 0.04 K^2 gives gain 0.9157 and posterior variance 0.03663 K^2, corresponding to a standard deviation of 0.1914 K. These covariance values are conditional on the specified input and noise assumptions. They do not automatically account for the missed 300 W step; that deterministic mismatch needs its own model or bound.

$$e_\theta(a) = R\Delta P \left(1 - e^{-a/\tau}\right) \tag{4}$$

The age a starts at the unreported step; this expression assumes the old and new inputs are otherwise constant.

Table 1: Temperature underestimation caused by holding the old power input.

Age a / s	**Error / K**
1	0.083
10	0.811
30	2.303
60	4.252

5 Delayed packets need chronological treatment.

A delayed temperature sample should not be subtracted from the current predicted state as if both represented the present. One practical approach is a fixed-lag buffer containing past estimates, covariances, input changes and accepted observations. Insert the new observation at its device timestamp, correct the buffered state there, and replay every subsequent prediction and update in chronological order. An explicitly derived out-of-sequence estimator is an alternative.

Packets outside the retained history require a documented policy: retain them for offline analysis, reject them from the real-time estimator, or use a separately justified delayed-data method. Preserve record identifiers so replay does not count a sample twice. A threshold-crossing event is also not automatically a Gaussian temperature measurement: its likelihood describes an inequality and the device trigger logic.

6 Validation and operating limits.

Validation should independently vary clock offset, transport delay, missing packets, interval length, sensor bias and load transients. Report timestamp-correct temperature error in kelvin, uncertainty calibration, innovation statistics and the frequency of stale-data rejection. Compare against a trusted reference with known measurement timing. A low average error can coexist with unacceptable underestimation during a small number of fast transitions.

The first-order model cannot resolve several thermal modes or changing heat paths. Unknown power steps need a disturbance model or a bounded-error envelope; incorrect R and C require calibration or parameter uncertainty. State estimation supplies a quantified interpretation of observations. Actuator constraints and thermal protection remain separate design questions. The useful outcome is a model whose time base and uncertainty can be audited, not a universal sampling rate.

Conclusion

Treat measurement time as part of the measurement. Propagate over actual intervals, update each observation once, and distinguish model uncertainty from the bias of missing information.

References

[1] R. E. Kalman (1960), A New Approach to Linear Filtering and Prediction Problems, Journal of Basic Engineering 82(1), 35-45.. `https://doi.org/10.1115/1.3662552`.

[2] S. Särkkä and L. Svensson (2023), Bayesian Filtering and Smoothing, 2nd edition, Cambridge University Press.. `https://www.cambridge.org/core/books/bayesian-filtering-and-smoothing/F88740E8D25010CF3119A5CA379FA37A`.