

Digital Twin of a Hot-Source / Sample / Cold-Sink Thermal-Conductivity Rig: Physics, Architecture, Action Plan, Lessons and Reading List

Build a **1D physics-based "digital shadow" first, then close the loop into a minimal digital twin**. The physics model is a 1D finite-difference model of the heat equation. It is calibrated live against temperature data by estimating thermal conductivity k , contact resistance R_c and a heat-loss coefficient h , and it then drives the rig: it declares steady state, steps the heater set-point and flags faults. Use a plain Python stack (NumPy/SciPy, pyserial/MQTT, SQLite/InfluxDB, Streamlit), not an industrial twin platform. Design the code around generic Asset / Model / Estimator / Store interfaces so a future reactor twin fed by the catalysis database can reuse everything except the physics module.

TL;DR

- **What to build:** a rig resembling a meter-bar / guarded-comparative cut-bar (ASTM E1225 / ASTM D5470 style), plus:
 - a 1D finite-difference model;
 - a least-squares + Kalman-filter layer that estimates k , R_c and h live;
 - a Streamlit dashboard.Steady-state Fourier's law ($k = QL/(A\Delta T)$) is the baseline. Transient inverse estimation is what makes it a twin rather than a calculator.
- **How to run it:** six phases over about one semester (~16 weeks), with model and hardware built in parallel. The hard problems are physical, not software:
 - contact resistance;
 - lateral heat losses;
 - sensor inter-calibration (cheap sensors are only $\pm 0.5^\circ\text{C}$ to $\pm 2^\circ\text{C}$ absolute);
 - hour-scale steady-state waits.
- **What the literature says:** the National Academies warn that "interest and enthusiasm around digital twin solutions currently outweighs evidence of their success". The successes (Aarhus incubator, Willcox's UAV twin, Rolls-Royce engines) share three traits: a fit-for-purpose model, few physically meaningful tunable parameters, and continual verification, validation and uncertainty quantification (VVUQ). Copy that discipline, not the platforms.

Assumptions

- **Team:** 3–6 CS undergraduates, fluent in Python and Git, new to heat transfer, about 6–10 hours per week each.
- **Hardware:**
 - Makerspace access: FDM printers, drill press, soldering, bench power supply.
 - Hobby-grade parts: ESP32 or Arduino, Raspberry Pi, thermistors / DS18B20 / thermocouples with MAX31855 or MAX31865 boards, a 10–30 W resistive heater or Peltier module, an aluminium heat sink with fan, insulation.
 - No dedicated DAQ initially.
- **Schedule and oversight:** one semester for the core twin. Prof. Khare (or an engineering colleague) reviews the physics and safety.

Part 1 — Modelling the Experiment as a Digital Twin

1.1 Physics, in CS-friendly terms

Fourier's law says heat flux is proportional to the temperature gradient: $q'' = -k \, dT/dx$, with k in $W/(m \cdot K)$.

The circuit analogy. Treat ΔT like voltage, heat rate Q (W) like current, and **thermal resistance $R = L/(kA)$** (K/W) like electrical resistance.

Steady state works like a series circuit:

$$Q = \Delta T_{\text{total}} / (R_{\text{c,hot}} + L/(kA) + R_{\text{c,cold}}).$$

If you know Q and measure the drop ΔT_s across the sample, then **$k = Q \cdot L / (A \cdot \Delta T_s)$** .

Comparative (meter-bar) trick. Heater power is not the heat that actually passes through the sample, because some is lost. So place reference bars of known k_{ref} in series with the sample. The same Q flows through both: [arxiv](#)

↗

$$k_s = k_{\text{ref}} (\Delta T_{\text{ref}} / L_{\text{ref}}) (L_s / \Delta T_s).$$

With one reference bar above the sample and one below, the mismatch between their two flux estimates is a live measure of heat loss.

Transient (heat equation), 1D, with an optional lateral-loss "fin" term:

$$\rho c \, \partial T / \partial t = \partial / \partial x (k \, \partial T / \partial x) - (hP/A)(T - T_{\infty}).$$

- Diffusivity $\alpha = k/(\rho c)$ sets the time scale $\tau \approx L^2/\alpha$.
- Example: a 10 mm PMMA disc ($\alpha \approx 1 \times 10^{-7} \text{ m}^2/\text{s}$) gives $\tau \approx 1000 \text{ s}$.
- The metal blocks add thermal mass, so expect **tens of minutes to ~2 h** to reach steady state. That wait is why transient estimation matters.

Boundary conditions:

- Dirichlet: fixed T, e.g. a temperature-controlled block.
- Neumann: fixed flux. A constant-power heater, or zero flux for perfect insulation.
- Robin / convective: $-k \partial T / \partial x = h(T - T_{\infty})$. Use this for a fan-cooled sink and for lateral losses.
- Contact resistance: a temperature jump $\Delta T = Q \cdot R_c$ at each interface. Model it as an interface conductance. It is often the dominant error, and it changes with grease, clamping pressure and surface flatness.

Lumped capacitance. If the Biot number $Bi = hL_c/k_{\text{block}} < 0.1$, a block is effectively isothermal: $m c dT/dt = Q_{\text{in}} - Q_{\text{out}}$. Aluminium blocks qualify. This turns the rig into a small RC network, the same structure the Aarhus incubator twin uses. [arxiv](#)
↗

Identifiability — the non-obvious part. At steady state, with one sample thickness, k and R_c are confounded: you only measure their series sum. Three ways to separate them:

1. Put sensors inside the sample or on its faces.
2. Test 2–4 thicknesses and regress total resistance against L. The slope gives 1/(kA) and the intercept gives the contact term (the ASTM D5470 logic).
3. Use transient data.

Similarly, h_loss and the true delivered power are confounded unless you use reference bars. Design the sensor layout with this in mind *before* building.

1.2 Which standard method your rig resembles

Method (standard)	Principle	Use	Relevance
Guarded hot plate (ASTM C177, ISO 8302)	Absolute steady state; a guard heater eliminates lateral loss	Insulators	Shows the value of guarding (v2 upgrade)
Heat flow meter (ASTM C518, ISO 8301)	A calibrated heat-flux transducer replaces the power measurement arxiv ↗	Insulation boards	Optional heat-flux sensor later

Method (standard)	Principle	Use	Relevance
Guarded-comparative cut-bar (ASTM E1225)	Sample in series with reference bars, plus a guard	0.2–200 W/(m·K), 90–1300 K astm ↗	Closest match to your source-sample-sink stack
Meter-bar, thin materials (ASTM D5470)	Hot/cold meter bars; multi-thickness regression separates contact resistance	Thermal interface materials	Closest match for thin samples
Transient plane source (ISO 22007-2)	A sensor acts as both heater and thermometer; fit the transient	Wide range	Template for the transient inverse fit
Laser flash (ASTM E1461)	Pulse one face; $\alpha \approx 0.1388 L^2/t^{1/2}$	Diffusivity	Transient data identify α ; you also need ρc

ASTM E1225's scope covers "materials with effective thermal conductivities in the range $0.2 < \lambda < 200$ W/(m·K) over the temperature range between 90 K and 1300 K". It defines a homogeneous specimen as one whose apparent conductivity "does not vary with changes of thickness or cross-sectional area by more than $\pm 5\%$ ". [astm](#)
↗

That $\pm 5\%$ criterion is a ready-made validation test for your twin.

Sizing consequence. For a 40 mm × 5 mm PMMA disc ($k \approx 0.2$) at $\Delta T = 20$ K, only $Q \approx 1$ W flows. An aluminium disc of the same size would show $\Delta T \approx 0.02$ K, which hobby sensors cannot resolve. Low- k samples suit thin discs; high- k samples need long bars (Searle-style) and much better insulation. **For v1, target 0.1–5 W/(m·K):** polymers, glass, ceramics, 3D-printed coupons.

1.3 Numerical approaches

Approach	When	Python
Analytical (series resistances; Fourier-series slab transient)	Week 1: test oracle	NumPy
Lumped RC network (3–10 nodes)	First twin model; fast enough to run inside a filter	scipy.integrate.solve_ivp

Approach	When	Python
1D finite difference, method of lines: $dT/dt = A(\theta)T + B(\theta)u$	Main v1 model (contact nodes, fin loss)	SciPy sparse; <code>solve_ivp(method="BDF")</code>
Explicit FTCS, $r = \alpha \Delta t / \Delta x^2$	Teaching only. Stable only if $r \leq \frac{1}{2}$	NumPy
Implicit (backward Euler / Crank–Nicolson)	Production stepping; unconditionally stable	<code>scipy.linalg.solve_banded</code>
Finite volume	Layered materials, flux-conservative	FiPy
FEM, 2D axisymmetric / 3D	v2: check the 1D assumption, size insulation	scikit-fem (light); FEniCSx (heavier; use Docker/WSL)
Reduced-order model (POD/DMD)	Only if FEM is too slow for real time	NumPy SVD, PyDMD
Surrogates (GP / NN / PINN)	Only after a physics model exists	scikit-learn, PyTorch, DeepXDE

First iteration: analytical + RC + 1D finite difference. Use 2D scikit-fem once as a model-form check.

This follows Wright & Davidson (UK National Physical Laboratory). They say a twin's model must be physics-based enough that its parameters can be updated from measurements, accurate enough for the purpose, and quick enough to run. They warn that purely data-driven models are valid only within their training range. `medium`

↗

1.4 Digital-twin layers for your rig

Layer	Implementation
Physical asset	Heater block → upper reference bar → sample → lower reference bar → sink; insulated sleeve; spring clamp
Sensing / DAQ	6–10 temperature channels + heater V/I + ambient; ESP32 with MAX31865/ADS1115/MAX31855, INA219; 1 Hz

Layer	Implementation
Data pipeline	JSON lines over serial or MQTT → Python ingest → SQLite/InfluxDB + Parquet archive
Virtual model	1D finite difference, $\theta = [k, R_{c,hot}, R_{c,cold}, h_{loss}, pc]$
Synchronisation	Windowed least squares + Kalman filter / unscented Kalman filter (UKF)
Analytics	Time-to-steady-state forecast, final-k prediction, residual-based fault flag
Visualisation	Streamlit (or Grafana)
Feedback (virtual → physical)	Auto-advance the set-point when k converges; safety cut-off; "insufficient excitation" alert

Reference architectures and where you sit:

- **Grieves (2002, University of Michigan "Conceptual Ideal for PLM", later the "Mirrored Spaces" / "Information Mirroring" model):** three parts — physical product, virtual product, and the data connections between them. [encyclopedia +5](#)
↗
- **NASA (Glaessgen & Stargel, 2012):** "an integrated multiphysics, multiscale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin." [scitepress](#)
↗
Your twin is of *this* as-built rig, with its own contact resistances.
- **Kritzinger et al. (2018):** three levels — digital model (manual data flow), digital shadow (automated physical→digital flow only) and digital twin (automated both ways). They found that literature on true twins "is scarce". [tuwien](#)
↗
[itcon](#)
↗

Realistic target: a digital shadow by week 8 and a minimal twin by the end, with set-point control and the interlock as the automated return path.

- **Tao et al., five dimensions** (physical entity, virtual entity, services, DT data, connections): [researchgate](#)
↗
[springer](#)
↗

map your repo modules onto these, and name the services explicitly — k estimation, time-to-steady-state, anomaly detection.

- **ISO 23247 (2021):**

- Four parts: overview, reference architecture, digital representation of observable manufacturing elements, information exchange. [arxiv](#)
↗
- Built on the ISO/IEC 30141 IoT reference architecture. [bsigroup](#)
↗
- Mapping: your rig is an "observable manufacturing element"; ingest is the device-communication entity; model + estimator form the digital-twin entity; the dashboard is the user entity.
- Parts 5 (digital thread) and 6 (composition) are in development. [arxiv](#)
↗

[arxiv](#)

↗

- **NASEM (2023):** a twin "is dynamically updated with data from its physical twin, has a predictive capability, and informs decisions that realize value. The bidirectional interaction between the virtual and the physical is central." The model must be "**fit for purpose**", and VVUQ must be continual. [nationalacademies](#)
↗

[energy](#)

↗

- **Digital Twin Consortium Capabilities Periodic Table** (launched March 2022; now v1.2): a technology-agnostic **requirements checklist**, not an architecture. Tick only what you need, e.g. Data Acquisition & Integrity, Data Streaming, Real-Time Monitoring, Prediction, Alerts.

[digitaltwinconsortium +2](#)

↗

- **AIAA/AIA (2020) "Digital Twin: Definition & Value"** and Gartner-style definitions: these stress the value of the decisions a twin supports. Useful for the motivation section of your write-up.

Realistic level: a single-asset, probabilistic twin, synchronised every few seconds, with a narrow but real virtual→physical loop.

1.5 Synchronisation = inverse problem

A. Steady-state algebra.

- Detect steady state: $|dT/dt| < 0.01 \text{ K/min}$ on all sensors for 10 min, using a rolling regression slope.
- Then apply the comparative formula.
- Repeat across powers and thicknesses, and regress to separate R_c .

B. Batch nonlinear least squares with `scipy.optimize.least_squares` :

- Residuals: $T_{\text{meas}} - T_{\text{model}}(\theta)$.
- Settings: `loss="soft_l1"`, `x_scale`, bounds.

- Covariance $\approx s^2(J^T J)^{-1}$, which gives error bars from day one.
- Plot Beck-style **scaled sensitivity coefficients** $\theta \partial T / \partial \theta$ for each sensor. If two parameters have proportional curves, they are not separately identifiable. Use this before building to choose sensor positions and the excitation profile (power steps or a pseudo-random binary sequence, PRBS).

C. Recursive estimation.

- A linear Kalman filter (KF) for the temperature field, with θ fixed. The method-of-lines model is linear in T , so the KF is exact. The Aarhus tutorial uses KF residuals to detect a lid opening on their incubator. [arxiv](#)
[github](#)
- Then a UKF on the augmented state $[T_1, \dots, T_N, k, R_c, h]$ with random-walk parameters, via `filterpy.kalman.UnscentedKalmanFilter`.
- A dual EKF is a published alternative for thermal twins of power electronics. [researchgate](#)
- An ensemble KF was applied to estimating thermal conductivity and heat-transfer coefficient "based on twin experiments" in casting (Oka & Ohno, 2020, *Int. J. Heat Mass Transfer* 149:119232); a companion paper (Natsume, Oka, Ogawa & Ohno, 2020, *IJHMT* 150:119222) covers the time-dependent h case. It is a good option if the model becomes FEM.
- Pragmatic recipe: a 1 Hz KF for state, plus a least-squares refit of θ every 5 minutes on a sliding window. Add the UKF in Phase 4b.

D. Bayesian calibration (offline):

- Tools: `emcee` (ensemble MCMC) or PyMC (wrap the model via `pm.Potential` or a custom Op), with ArviZ diagnostics.
- Priors: physical, e.g. PMMA $k \in [0.15, 0.25]$; log-normal for R_c .
- If residuals show structure, add a Kennedy–O'Hagan GP discrepancy term.
- This is the Kapteyn–Pretorius–Willcox (2021) view in miniature: each asset maintains a unique "digital state" on a shared model, updated by Bayesian inference. [mit](#)

E. Sensitivity analysis and UQ.

- SALib (Sobol / Morris) ranks which uncertain inputs dominate k : sensor offsets, L , A , R_c , h .
- Monte Carlo propagation of sensor accuracy will show that relative sensor offsets dominate.

1.6 Data-driven and hybrid options

Option	Now?	Reason
GP surrogate of the finite-	Maybe (Phase	Speeds up MCMC; teaches UQ

Option	Now?	Reason
difference model	4)	
GP discrepancy model	Stretch goal	Honest treatment of systematic misfit
Neural-network surrogate	No	The 1D model already runs in milliseconds
Physics-informed neural network (PINN) for the inverse k problem	Side notebook	Educational, but slower and less accurate than least squares on finite differences
ANN replacing CFD inside a Kalman smoother	Remember the pattern	Published near-real-time inverse estimation for thermal twins; relevant for the reactor arxiv ↗
Hybrid physics + ML residual	Yes, for the reactor	Physics gives structure, ML learns what physics misses

Bridge to catalysis. A lab reactor twin swaps [fd1d.py](#) for species mass balances plus an energy balance. That energy balance is your heat equation with a reaction heat source. Arrhenius parameters (A , E_a , orders) become θ . The catalysis database supplies the **priors**; live data update the posteriors exactly as k is updated now. The industrial precedent is model-based digital design of fixed-bed catalytic reactors: it first builds an "experiment digital twin" of the lab kinetics, then scales it to the industrial reactor. [acs](#)
↗

Part 2 — Action Plan

2.1 Phases

Phase	Weeks	Deliverable	Owners
0 Scoping	1–2	4-page scope: k range, geometry, sensor plan, capabilities checklist, safety plan; repo + CI skeleton	All
1 Model	2–5	analytical.py , lumped.py , fd1d.py ; finite difference matches the analytical solution within 0.5 %; convergence plot; explicit-stability demo ($r = 0.49$ vs 0.51)	Model team (2)

Phase	Weeks	Deliverable	Owners
2 Rig	2–8	CAD + bill of materials; built rig; inter-calibration report; first steady-state k of a reference within $\pm 10\%$	Hardware team (2)
3 Pipeline	5–8	JSON schema, MQTT, storage, replay tool → digital shadow	Data team (1–2)
4 Synchronisation	7–12	4a least squares + sensitivity; 4b KF/UKF live; 4c Bayesian posterior	Model team + 1
5 Dashboard + feedback	9–12	Streamlit live view; automated set-point stepping; interlock → minimal twin	Data team
6 Validation & write-up	12–16	V&V report (≥ 3 materials, ≥ 2 thicknesses); MkDocs site; poster or paper	All

Team size. With 3 people, one each on model, hardware and data, all reviewing pull requests. With 6, pairs, plus rotating "integration owner" and "documentation owner" roles. Hold a weekly 30-minute end-to-end run, even with stubs, to catch interface drift.

Phase 0 checklist.

- Read: Wright & Davidson, Kritzing, the NASEM highlights, the Aarhus incubator paper, and Incropera/Bergman chapters 1–3 and 5.
- Decide: comparative meter-bar; low- k discs 30–50 mm diameter, 3–10 mm thick; 25–80 °C.
- Write down **the decision the twin informs**, e.g. "declare k with $\leq 5\%$ (1σ) uncertainty as early as possible, then auto-advance the test."

2.2 Phase 1 — Python model modules

- `physics/analytical.py` : `steady_profile(layers, T_hot, T_cold)` ; `slab_transient(x, t, alpha, L, T0, T1, n_terms)` .
- `physics/lumped.py` : RC network from YAML → `solve_ivp` .
- `physics/fd1d.py` :
 - `build_operator(mesh, k(x), rho_c(x), R_c, h_loss, P/A)` → sparse A, B.
 - Steppers: explicit, backward Euler, Crank–Nicolson (`solve_banded`), and a BDF path.
 - Boundary-condition classes: `Dirichlet` , `Neumann` , `Robin` .
 - Contact resistance as harmonic-mean interface conductance plus $1/R_c$.
- `physics/fem2d.py` (optional): scikit-fem axisymmetric model to quantify radial-loss error.
- `viz/plots.py` : Matplotlib for figures, Plotly for interactive views.
- Tests (`pytest` , `hypothesis`):
 - finite difference vs analytical;

- energy conservation to $1e-8$;
- Crank–Nicolson second-order convergence;
- explicit instability above $r = \frac{1}{2}$;
- monotone profile for positive ΔT .

2.3 Phase 2 — rig and instrumentation

Stack, bottom to top:

1. Finned sink + fan (or a water block).
2. Aluminium/brass reference bar, 25–40 mm, with 2–3 drilled sensor holes.
3. Sample.
4. Upper reference bar.
5. Aluminium heater block (cartridge, power-resistor or Kapton heater).
6. Spring clamp of known force.

Wrap the stack in mineral wool or ceramic fibre inside a printed housing.

Printing note. PLA softens at about 55–60 °C. Use PETG, ASA, PC or nylon near the hot side, and keep printed parts out of load paths at temperature.

Sensors:

Sensor	Accuracy (datasheet)	Speed / resolution	Use
DS18B20 (1-Wire)	$\pm 0.5\text{ }^{\circ}\text{C}$, -10 to $+85\text{ }^{\circ}\text{C}$	$0.0625\text{ }^{\circ}\text{C}$ at 12-bit; up to 750 ms	Ambient, sink, housing; quick prototype analog ↗ studiopieters ↗
K thermocouple + MAX31855 (SPI)	$\pm 2\text{ }^{\circ}\text{C}$ (K-type, -200 to $+700\text{ }^{\circ}\text{C}$)	$0.25\text{ }^{\circ}\text{C}$	Heater block / hot points; ΔT only after inter-calibration analog ↗
10 k Ω NTC bead + ADS1115	$\sim \pm 0.1\text{--}0.2\text{ }^{\circ}\text{C}$ after Steinhart-Hart calibration	Fast, high sensitivity	Primary bar and sample-face sensors
PT100/PT1000 RTD + MAX31865	Class A/B; very stable	15-bit	Calibration reference
Thin-film heat-flux sensor	Calibrated	μV output	v2 check of Q

Inter-calibration is non-negotiable.

- Two independent $\pm 0.5\text{ }^{\circ}\text{C}$ sensors give a ΔT uncertainty of about 0.7 K. Keeping ΔT error $\leq 5\%$ would then need $\Delta T \geq 14\text{ K}$; with $\pm 2\text{ }^{\circ}\text{C}$ thermocouples you would need about 57 K.
- Only the relative offsets matter for ΔT . So soak all sensors together in one isothermal aluminium block (or a stirred bath) at ≥ 3 temperatures.
- Fit per-sensor offset/gain into [configs/calibration.yaml](#). Relative error typically drops to about 0.05–0.1 K.

Placement and acquisition.

- Drill holes to the centreline, use thermal paste, and record positions to $\pm 0.1\text{ mm}$.
- Use 3 sensors per bar, so non-linearity of the bar profile flags losses.
- Add an ambient sensor and an outer-insulation sensor.
- Measure heater power with an INA219/INA226; prefer a DC bench supply over PWM.
- Sample at 1 Hz with on-MCU averaging. Timestamp on the Pi using NTP, and keep MCU sequence numbers.

2.4 Phase 3 — data pipeline

- **Firmware** (PlatformIO, Arduino-ESP32 or MicroPython) emits: `{"seq":1234,"t_mcu_ms":...,"T":{"ref_top_1":41.23,...},"V":12.01,"I":0.83,"fw":"0.3.1"}`
- `io/serial_reader.py` : `serial.Serial(port,115200,timeout=1)` , line framing, sequence-gap detection, reconnect.
- `io/mqtt_bridge.py` : `paho-mqtt` with a Mosquitto broker. Subscribes to `rig1/telemetry` ; publishes commands on `rig1/cmd` — the virtual→physical channel.
- `io/schema.py` : `pydantic` validation of units and ranges; reject and log bad packets.
- `io/storage.py` :
 - SQLite first (fine at 1 Hz × 10 channels).
 - Later, InfluxDB (pairs with Grafana) or TimescaleDB.
 - Archive each run as Parquet with `run_metadata.yaml` : sample, thickness, clamp force, calibration hash, firmware version, git commit.
- `io/replay.py` : re-streams stored runs at ×1–×100. Use it to develop estimators without the rig and for regression tests.
- **pandas** for resampling and rolling-slope steady-state detection.

2.5 Phase 4 — synchronisation modules

- `estimation/steady.py` : steady-state detector, comparative k, multi-thickness OLS (`statsmodels`) to separate `R_c`.
- `estimation/least_squares.py` : windowed fit, Jacobian covariance, sensitivity plots.
- `estimation/kalman.py` : `FilterPy` `KalmanFilter` (F, B from the discretised model; Q, R from calibration noise), then `UnscentedKalmanFilter` with `MerweScaledSigmaPoints` . Pin the version; Labbe's free book is the companion.
- `estimation/bayes.py` : `emcee` or `PyMC` + `ArviZ` (R-hat, effective sample size, posterior predictive checks).
- `twin/state.py` : `TwinState(t, T_nodes, theta_mean, theta_cov, health_flags)` .
- `twin/runner.py` : the event loop — ingest → KF predict/update → periodic θ refit → forecast → decide/command → persist.

2.6 Phase 5 — dashboard

- **Streamlit** (`dashboard/app.py`), showing:
 - the live $T(x)$ profile with model overlay and a $\pm 2\sigma$ band;
 - k, `R_c` and h traces with uncertainty;
 - normalised innovations (the fault detector);
 - predicted time-to-steady-state and final k;
 - buttons that publish MQTT commands.
- **Alternatives**: Grafana on InfluxDB for no-code time-series panels; Dash only if you need complex callbacks; a Plotly 3D cylinder is plenty for "3D". Aarhus used the lightweight Godot

engine; github

↗

Omniverse is overkill.

2.7 Phase 6 — validation and documentation

- **Validation matrix:** ≥ 3 known materials (e.g., PMMA ~ 0.19 , borosilicate $\sim 1.1 \text{ W}/(\text{m}\cdot\text{K})$, printed PLA) $\times \geq 2$ thicknesses $\times 2$ powers.
 - **Report:**
 - $k \pm U$ (coverage factor 2) per the GUM;
 - the $\pm 5\%$ thickness-independence check;
 - residual statistics;
 - twin metrics: estimate-convergence time vs steady-state time, forecast error, anomaly false-alarm rate, end-to-end latency.
 - **Separate verification from validation.** Verification asks whether the solver is correct (analytical comparisons, convergence order). Validation asks whether the model is right for the rig (held-out runs). Re-validate after any hardware change, per NASEM's continual-VVUQ point. nationalacademies
- ↗
- energy
- ↗
- **MkDocs site:**
 - architecture diagram mapped to ISO 23247 and Tao's five dimensions;
 - derivations;
 - calibration standard operating procedure (SOP);
 - test SOP;
 - data dictionary;
 - assumption-change log.

2.8 Repository structure and practice



```

thermal-twin/
├── pyproject.toml      # pinned deps; package "thermotwin"
├── configs/           # rig_v1.yaml, calibration.yaml, estimator.yaml
├── firmware/esp32/    # PlatformIO; JSON schema v1
├── hardware/          # CAD (STEP+source), BOM.csv, wiring, photos
├── src/thermotwin/
│   ├── physics/       # analytical, lumped, fd1d, fem2d, materials
│   ├── estimation/    # steady, least_squares, kalman, bayes, sensitivity
│   ├── io/            # serial_reader, mqtt_bridge, schema, storage, replay
│   ├── twin/         # state, runner, decisions, safety
│   └── interfaces.py  # Asset / Model / Estimator / Store protocols
├── dashboard/app.py
├── notebooks/        # exploratory only, never imported
├── data/             # raw/ immutable, processed/; DVC or git-lfs
├── tests/            # unit, property, replay-regression, @pytest.mark.hil
├── docs/             # MkDocs
└── .github/workflows/ci.yml # ruff, mypy, pytest

```

Practices:

- Trunk-based Git: short branches, one reviewer per pull request, green CI.
- Tagged releases, e.g. `v0.3-digital-shadow`.
- `pre-commit` with ruff and mypy.
- Immutable raw data; every artefact records its commit and config hash.
- `interfaces.py` makes `FD1DConductionModel` and a future `PackedBedReactorModel` both expose `step(state,u,θ,dt)`, `observe(state)` and `params()`, so estimators are reused unchanged.

2.9 Toolkit landscape

Tool	Verdict
Plain Python stack	Adopt: transparent, free, maximum learning
Mosquitto (MQTT), InfluxDB + Grafana	Adopt (Grafana optional)
Node-RED	Optional glue for demos; keep testable logic out of it
ThingsBoard	Skip for v1; consider for many rigs
Eclipse Ditto	Skip; one Phase-6 spike to learn "twin as an API"

Tool	Verdict
FMI/FMU (FMPy, pythonfmu , OpenModelica)	Learn lightly: export the model as an FMU in Phase 6 for future co-simulation
Modelica / OpenModelica	Good for the reactor phase
Simulink/Simscape	Only if licensed and you need model predictive control (MPC)
COMSOL	Use once (if licensed) as an independent 2D/3D reference
OpenFOAM	Skip (no flow)
Ansys Twin Builder	Skip (licence, complexity)
Azure Digital Twins (DTDL)	Skip implementation; read the DTDL spec for schema design
AWS IoT TwinMaker	Skip; cost and lock-in (check current service status)
NVIDIA Omniverse	Skip; visual, GPU-heavy
Asset Administration Shell (Eclipse BaSyx)	Optional Phase-6 exercise describing the rig as an AAS

Evidence for plain Python first. Aarhus University tried to implement their simple incubator twin in each open-source DT framework they surveyed. It fitted only 5 of the 14. They also found some tools marketed as DT frameworks cover only models or shadows. [researchgate](#)

[github](#)

Your bottleneck is physics, estimation and data quality, not orchestration.

2.10 Standards and good practice

- **Measurement:** ASTM E1225, D5470, C177/ISO 8302, C518/ISO 8301, E1461, ISO 22007-2; uncertainty per the GUM (JCGM 100:2008).
- **Digital twin:** ISO 23247-1...4:2021; ISO/IEC 30173:2023 (concepts and terminology); ISO/IEC 30141 (IoT reference architecture).
- **Interoperability:** FMI (Modelica Association); Asset Administration Shell (IDTA; IEC 63278); OPC UA; MQTT (OASIS).

- **IEEE:** IEEE 2888 series (cyber-physical interfacing). Check current working-group status before citing.
- **Practice:** FAIR data, a written VVUQ plan, and an electrical/heating risk assessment.

2.11 Pitfalls

Pitfall	Symptom	Mitigation
Contact resistance dominates	k varies with thickness or re-clamping	Paste/pads; spring preload; multi-thickness regression; explicit R_c
Lateral losses	Upper and lower bar fluxes disagree; curved bar profiles	Insulation; mean stack temperature near ambient; guard heater (v2); fin term
Sensor offsets	Non-zero ΔT with heater off	Isothermal inter-calibration; zero-power check every run
Self-heating (thermistor/RTD)	Upward drift; sensitivity to airflow	$\sim 100 \mu A$ excitation; ratiometric; duty-cycling
Thermocouple cold-junction error	Offsets that track board/room temperature	Keep boards away from the heater; inter-calibrate; thermistors for ΔT
Noise / ground loops	Spikes with PWM	Twisted pairs, ungrounded junctions, averaging, DC supply
Steady-state waits	1-2 h per point	Transient estimation; pre-heating; smaller masses
Poor identifiability	Large, correlated uncertainties	Sensitivity study before building; power steps / PRBS
Mis-sized sample	ΔT or Q too small	$R = L/(kA)$ sizing in Phase 0
Printed-part creep	Clamp relaxes	High-temperature filament; metal load path
Timestamp loss	Estimator diverges	Sequence numbers; single time authority; replay tests
Safety	Overheating, burnt insulation	Hardware thermal fuse independent of software; <code>safety.py</code> interlock; current-

Pitfall	Symptom	Mitigation
		limited, fused supply; heat-rated wiring; supervised early runs
Over-engineering	Cloud/Kubernetes, no physics	Freeze the stack in Phase 0; digital shadow by week 8 as a hard gate

Part 3 — Past Implementations and Lessons

Case	What they did	What worked	Difficulties	Apply
NASA Apollo → Glaessgen & Stargel (2012)	Apollo's physical ground twins; 2012 defined the DT as a probabilistic, as-built simulation for certification/sustainment semanticscholar ↗ researchgate ↗	Clear vision	Long-term vision; "ultra-high fidelity" exceeded compute nasa ↗	Keep <i>probabilistic, as-built</i> ; drop ultra fidelity arxiv ↗
Kapteyn, Pretorius & Willcox (2021), UAV	Dynamic Bayesian network + component-based reduced-order models; experimental calibration nature ↗ utexas ↗	Principled, scalable updating with end-to-end UQ nature ↗	Needs reduced-order-model libraries and expertise	θ -posterior + KF i: a miniature versio
Niederer et al. (2021)	Cardiac + aerospace exemplars	Clear challenge list	Twins are "bespoke technical solutions that are difficult to scale" kcl ↗	Reuse-oriented interfaces
Aarhus incubator (2021–2025)	Insulated box + heatbed; RC models; RabbitMQ;	A 4-parameter model captured	Only 5/14 frameworks	Closest template clone the repo,

Case	What they did	What worked	Difficulties	Apply
	KF anomaly detection; open source arxiv researchgate	heatbed thermal inertia better than a 2- parameter one	usable; researchgate maintenance	copy RC + KF + messaging arxiv
Machine-tool thermal twin (Lang, Talleri, Mayr, Wegener & Bambach, ETH Zurich, <i>Manuf. Lett.</i> 41:208–218, 2024)	Thermomechanical model + KF state observer; bench then 5- axis machine sciencedirect	Estimation with few sensors vs prior 284-sensor interpolation; sciencedirect RMSE at unobserved sensors only 2.7 °C on the bench and 1.3 °C on the machine; thermal-error RMSE cut from 67.4 µm to 33.3 µm	Model reduction, long-term drift	Model + KF beats adding sensors
Power- electronics thermal twins	PSO identification + dual EKF online researchgate	Parameter drift / KF residuals reveal thermal- path ageing researchgate	Coexisting ageing mechanisms confound parameters researchgate	Track R_c drift after re-clamping
Inverse heat transfer twins	ANN replaces CFD in an extended Kalman smoother; EnKF for k and h in casting sciencedirect arxiv	Near-real-time inversion	Surrogate training; sensor placement	Surrogates only if physics is slow

Case	What they did	What worked	Difficulties	Apply
GE Predix / Digital Wind Farm (2015)	IIoT platform + asset twins	Condition monitoring on sub-second data	Company-sourced value claims; analysts called it "more evolutionary than revolutionary"; Predix was costly and some divisions struggled to use it	Start from a decision, not a platform windpowermonthr ↑ harvard ↑
Rolls-Royce IntelligentEngine / EHM	Engine data to a ground twin; EHM can "talk back" and refocus monitoring	Time-on-wing, predictive maintenance	Vendor-reported outcomes; heavy data-integration investment	Let the twin change <i>what</i> is measured rolls-royce ↑ rolls-royce ↑
Catalytic/process reactors	Fixed-bed digital design via an "experiment digital twin"; pilot steam-methane-reforming (SMR) + water-gas-shift (WGS) twin; Aspen + Simulink MPC twins; plasma-reactor twin calibrating a few PDE coefficients	SMR/WGS twin RMSE 12.5 °C (reformer), 13.9 °C (WGS), 1.9 % H ₂ yield; plasma twin matched chamber pressure to 0.10 % on average	Scarce custom-modelling skills and maintenance; multitubular reactors "highly complex"; acs ↑ psenterprise ↑ vendor-heavy sources	Calibrate a few meaningful parameters, not b ML models acs ↑ arxiv ↑
Pharma continuous manufacturing	Mechanistic, hybrid and data-driven twins	Hybrid strategies	"Real-time applications are rarely	Expect to stall at the shadow stage

Case	What they did	What worked	Difficulties	Apply
			seen"; integrated twins "still in [their] infancy"; data integration, accuracy, regulation sciencedirect +2 ↗	closed loop is an achievement
Bioprocess DT core-model tool	Auto-assembles kinetic + physico-chemical + reactor submodels	Development and compute time cut by 90 %	Tool-specific	Kinetics / transpo / reactor decomposition dc ↗
Self-driving labs	Burger et al. (2020): mobile robot ran 688 experiments over 8 days across 10 variables via batched Bayesian optimisation; found photocatalysts ~6× more active researchgate ↗ nature ↗	Closed-loop, model-guided experimentation	Robust automation needed; human oversight required nature ↗	The "next set- point" loop is a micro self-driving lab
Buildings, batteries, additive manufacturing (thermal)	RC-network building twins; equivalent-circuit + thermal battery twins with KF; melt-pool thermal twins	Low-order RC + KF is the workhorse	Sparse sensors, drift; AM models too slow for real time	Your RC + KF core is industry- standard

On the numbers. Emerj relays GE CTO Colin Parris's claim of "over 1.2 million digital twins" generating "approximately \$600 billion in value" between 2016 and 2017. [emerj](#)
[↗](#)

A Microsoft customer story says Rolls-Royce engine monitoring "tracks more than 10,000 engine parameters" and "detects and prevents around 400 unplanned maintenance events"

annually". [microsoft](#)

↗

Treat both as vendor claims, not evidence.

Reviews and takeaways:

- **Tao et al. (2019, IEEE TII):** industrial state of the art → name your services.
- **Kritzinger et al. (2018):** model / shadow / twin taxonomy; true twins are rare [tuwien](#)
↗
→ report your level honestly.
- **Fuller et al. (2020, IEEE Access):** challenges in IT infrastructure, data quality, security, trust and standards → treat data quality as a requirement.
- **Jones et al. (2020, CIRP JMST):** characterised twins by 13 characteristics, including fidelity, twinning rate and the physical-to-virtual and virtual-to-physical connections → document your twinning rate.
- **Rasheed, San & Kvamsdal (2020, IEEE Access):** physics-based, data-driven and hybrid models; reduced-order models; data assimilation → hybrids and ROMs as enablers.
- **VanDerHorn & Mahadevan (2021, DSS):** reviewed 46 definitions and proposed a generalised one [colab](#)
↗
→ write down purpose and decisions.
- **Wright & Davidson (2020):** "a digital twin without a physical twin is a model" [springer](#)
↗
→ your acceptance criteria for [fd1d.py](#).
- **Niederer et al. (2021) and Kapteyn et al. (2021):** scale needs lower barriers and Bayesian foundations. [sciencedaily](#)
↗
[utexas](#)
↗
- **NASEM (2023):** fit for purpose, continual VVUQ, sustainment costs. [nationalacademies](#)
↗

Prioritised lessons:

1. Start from the decision, not the platform.
2. Use few, physical, continuously updated parameters.
3. Estimation beats adding sensors.
4. VVUQ and data quality are continual.
5. Design for reuse.
6. Plan a real, small virtual→physical loop so you do not stall at the shadow stage.

Part 4 — Extending Toward the Catalysis Database

- **Abstractions:** `Asset`, `Model` (`step`, `observe`, `params`, `param_priors`), `Estimator`, `Store`, plus a read-only `KnowledgeBase` adapter.
- **Adapter output:** database records become `ParameterPrior` objects with value, uncertainty, units, provenance (database, record ID, DOI) and validity range (temperature, catalyst, facet).
- **Know the sources:**
 - **Catalysis-Hub.org** (Winther et al., 2019): "more than 100,000 chemisorption and reaction energies obtained from electronic structure calculations". [nih](#)
↗
 - **CatApp** (Hummelshøj et al., 2012): about 3,000 reactions on mostly close-packed transition-metal surfaces. It stores no atomic structures, which limits reproducibility. [nature](#)
↗
 - **NIST Chemical Kinetics Database (SRD 17):** gas-phase elementary reactions only, not surface catalysis. A 2020 paper noted no updates since 2015, [nist](#)
↗
[copernicus](#)
↗
and published size figures conflict. [researchgate](#)
↗
[academia](#)
↗
- **Modelling path:**
 1. Isothermal CSTR with power-law kinetics (an ODE as easy as `lumped.py`).
 2. Non-isothermal packed bed (reuses `fd1d.py`).
 3. Microkinetics (e.g., Cantera surface kinetics) seeded from DFT energies.
 4. Hybrid physics + ML residuals trained across the database.
- **Experiment design:** reuse the set-point loop as Bayesian experimental design, the self-driving-lab pattern (Burger et al., 2020; Abolhasani & Kumacheva, 2023). [nature](#)
↗
[nature](#)
↗

Part 5 — Reading List

5.1 Referenced in this report

Digital-twin foundations

- Grieves, M. (2014). *Digital Twin: Manufacturing Excellence through Virtual Factory Replication* (white paper).

- Grieves, M. & Vickers, J. (2017). "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems," Springer.
- Glaessgen, E. & Stargel, D. (2012). "The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles." AIAA 2012-1818. DOI 10.2514/6.2012-1818. [semanticscholar](#)
↗
<https://ntrs.nasa.gov/api/citations/20120008178/downloads/20120008178.pdf>
- Kritzinger, W., Karner, M., Traar, G., Henjes, J. & Sihn, W. (2018). *IFAC-PapersOnLine* 51(11):1016–1022. DOI 10.1016/j.ifacol.2018.08.474 [tuwien](#)
↗
- Tao, F., Zhang, H., Liu, A. & Nee, A.Y.C. (2019). "Digital Twin in Industry: State-of-the-Art." *IEEE TII* 15(4):2405–2415. DOI 10.1109/TII.2018.2873186 [crossref](#)
↗
- Tao, F., Zhang, M. & Nee, A.Y.C. (2019). "Five-Dimension Digital Twin Modeling and Its Key Technologies," in *Digital Twin Driven Smart Manufacturing*, pp. 63–81. [springer](#)
↗
- Fuller, A., Fan, Z., Day, C. & Barlow, C. (2020). *IEEE Access* 8:108952–108971. DOI 10.1109/ACCESS.2020.2998358 [crossref](#)
↗
- Jones, D., Snider, C., Nassehi, A., Yon, J. & Hicks, B. (2020). *CIRP JMST* 29A:36–52. DOI 10.1016/j.cirpj.2020.02.002 [crossref](#)
↗
- Rasheed, A., San, O. & Kvamsdal, T. (2020). *IEEE Access* 8:21980–22012. DOI 10.1109/ACCESS.2020.2970143 [crossref](#)
↗
- Wright, L. & Davidson, S. (2020). *Adv. Model. Simul. Eng. Sci.* 7:13. DOI 10.1186/s40323-020-00147-4 [springer](#)
↗
- VanDerHorn, E. & Mahadevan, S. (2021). *Decision Support Systems* 145:113524. DOI 10.1016/j.dss.2021.113524 [scirp](#)
↗
- Niederer, S.A., Sacks, M.S., Girolami, M. & Willcox, K. (2021). *Nat. Comput. Sci.* 1:313–320. DOI 10.1038/s43588-021-00072-5 [sciencedaily](#)
↗
- Kapteyn, M.G., Pretorius, J.V.R. & Willcox, K.E. (2021). *Nat. Comput. Sci.* 1:337–347. DOI 10.1038/s43588-021-00069-0 [osti](#)
↗
[tandfonline](#)
↗
- NASEM (2023). *Foundational Research Gaps and Future Directions for Digital Twins*. DOI 10.17226/26894. Highlights: <https://nap.nationalacademies.org/resource/26894/RH-digital-twins.pdf>. Full text: <https://www.ncbi.nlm.nih.gov/books/NBK605506/>
- AIAA/AIA (2020). *Digital Twin: Definition & Value*.
- Digital Twin Consortium, Capabilities Periodic Table:
<https://www.digitaltwinconsortium.org/initiatives/capabilities-periodic-table/>

Thermal and lab twins, estimation

- Feng, H. et al. (2021). "The Incubator Case Study for Digital Twin Engineering." arXiv:2102.10390. Code: https://github.com/INTO-CPS-Association/example_digital-twin_incubator
- Feng, H., Gomes, C. & Larsen, P.G. (2023). "Model-Based Monitoring and State Estimation for Digital Twins: The Kalman Filter." arXiv:2305.00252
- Gomes, C. et al. (2025). "Digital Twin Tutorial: The Incubator Case Study." LNCS, pp. 68–101. DOI 10.1007/978-981-96-4656-2_3 [github](#)
↗
- Gil, S. et al. (2024). "Survey on open-source digital twin frameworks." *Softw. Pract. Exp.* DOI 10.1002/spe.3305 [github](#)
↗
- Lang, S., Talleri, S., Mayr, J., Wegener, K. & Bambach, M. (2024). "Kalman filter-driven state observer for thermal error compensation in machine tool digital twins." *Manuf. Lett.* 41:208–218. DOI 10.1016/j.mfglet.2024.09.025.
<https://www.sciencedirect.com/science/article/pii/S2213846324000877>
- Kuprat, J., Debbadi, K., Schaumburg, J., Liserre, M. & Langwasser, M. (2024). "Thermal Digital Twin of Power Electronics Modules for Online Thermal Parameter Identification" (PSO + DEKF; "experimentally validated in the laboratory"). *IEEE JESTPE*. DOI 10.1109/JESTPE.2023.3328219
- Zhang, X. et al. (2023). ANN-based extended Kalman smoothing for inverse heat transfer. arXiv:2304.00021
- Oka, Y. & Ohno, M. (2020). "Parameter estimation for heat transfer analysis during casting processes based on ensemble Kalman filter." *Int. J. Heat Mass Transfer* 149:119232. DOI 10.1016/j.ijheatmasstransfer.2019.119232. Companion: Natsume, Oka, Ogawa & Ohno (2020), *IJHMT* 150:119222.

Industry and process

- Spatenka, S., Matzopoulos, M., Urban, Z. & Cano, A. (2019). "From Laboratory to Industrial Operation: Model-Based Digital Design and Optimization of Fixed-Bed Catalytic Reactors." *Ind. Eng. Chem. Res.* 58(28):12571–12585. DOI 10.1021/acs.iecr.9b01314 (vendor-authored: examples "implemented in gPROMS ProcessBuilder, using PSE's Advanced Model Library for Fixed-Bed Catalytic Reactors (AML:FBCR)").
- "Digital Twin Approach toward the Design, Optimization, and Control of a Catalytic System for High-Purity Hydrogen Production from Biogas." *Ind. Eng. Chem. Res.* 64(45):21588.
- "Digital Twin of an Argon-Hydrogen Plasma Reactor." arXiv:2609.15250 (preprint).
- Chen, Y., Yang, O., Sampat, C., Bhalode, P., Ramachandran, R. & Ierapetritou, M. (2020). "Digital Twins in Pharmaceutical and Biopharmaceutical Manufacturing: A Literature Review." *Processes* 8(9):1088. DOI 10.3390/pr8091088
- "Transformative roles of digital twins from drug discovery to continuous manufacturing" (2025). <https://pmc.ncbi.nlm.nih.gov/articles/PMC12516570/>

- Bioprocess DT core models. *Fermentation* 10(9):463. DOI 10.3390/fermentation10090463
- Rolls-Royce company pages (2019); Microsoft customer story; Windpower Monthly on GE's Digital Wind Farm; Emerj on GE (secondary); PSE gPROMS Reactor (vendor).

Self-driving labs and catalysis data

- Burger, B. et al. (2020). "A mobile robotic chemist." *Nature* 583:237–241. [nature](#)
DOI 10.1038/s41586-020-2442-2
- Abolhasani, M. & Kumacheva, E. (2023). *Nature Synthesis* 2:483–492. [nature](#)
DOI 10.1038/s44160-022-00231-0
- Coley, C.W., Eyke, N.S. & Jensen, K.F. (2020). "Autonomous Discovery in the Chemical Sciences," Parts I and II. *Angew. Chem.* DOIs 10.1002/anie.201909987, 10.1002/anie.201909989
[mit](#)
- Winther, K.T. et al. (2019). Catalysis-Hub. *Sci. Data* 6:75. DOI 10.1038/s41597-019-0081-y
[researchgate](#)
- Hummelshøj, J.S. et al. (2012). CatApp. *Angew. Chem.* 51:272–274. DOI 10.1002/anie.201107947
[nih](#)
- NIST SRD 17: <https://kinetics.nist.gov/>

Standards and datasheets

- ASTM E1225-20: <https://store.astm.org/e1225-20.html>. Also ASTM D5470, C177, C518, E1461; ISO 8301, 8302, 22007-2.
- ISO 23247-1...4:2021. NIST AMS 400-2 use cases (free):
https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=932269
- ISO/IEC 30173:2023; ISO/IEC 30141.
- MAX31855: <https://www.analog.com/media/en/technical-documentation/data-sheets/max31855.pdf>
- DS18B20: <https://analog.com/media/en/technical-documentation/data-sheets/ds18b20.pdf>
- Zhao, D. et al. (2016). "Measurement Techniques for Thermal Conductivity and Interfacial Thermal Conductance..." arXiv:1605.08469

5.2 Recommended further reading

Heat-transfer textbooks

- Bergman, Lavine, Incropera & DeWitt, *Fundamentals of Heat and Mass Transfer*, 8th ed., Wiley 2017 (chapters 1–5).
- Çengel & Ghajar, *Heat and Mass Transfer*, 6th ed., McGraw-Hill 2020 (gentlest start).
- Lienhard & Lienhard, *A Heat Transfer Textbook* (free PDF).

Numerical methods

- LeVeque, *Finite Difference Methods for ODEs and PDEs*, SIAM 2007 (chapter 9).
- Patankar, *Numerical Heat Transfer and Fluid Flow*, 1980.
- Versteeg & Malalasekera, *An Introduction to CFD: The Finite Volume Method*, 2nd ed., 2007.

Inverse problems and estimation

- Özişik & Orlande, *Inverse Heat Transfer*, 2nd ed., CRC 2021.
- Beck, Blackwell & St. Clair, *Inverse Heat Conduction*, Wiley 1985.
- Beck & Arnold, *Parameter Estimation in Engineering and Science*, 1977.
- Kaipio & Somersalo, *Statistical and Computational Inverse Problems*, 2005.
- Simon, *Optimal State Estimation*, 2006.
- Labbe, *Kalman and Bayesian Filters in Python*: <https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python>
- Kennedy & O'Hagan (2001), *J. R. Stat. Soc. B* 63:425–464.
- Saltelli et al., *Global Sensitivity Analysis: The Primer*, 2008.

Library documentation

- SciPy (`solve_ivp` , `least_squares` , `solve_banded`).
- FiPy: <https://www.ctcms.nist.gov/fipy/>
- scikit-fem: <https://scikit-fem.readthedocs.io>
- FEniCSx tutorial: <https://jsdokken.com/dolfinx-tutorial/>
- FilterPy: <https://filterpy.readthedocs.io>
- PyMC: <https://www.pymc.io>; emcee: <https://emcee.readthedocs.io>; ArviZ: <https://python.arviz.org>; SALib: <https://salib.readthedocs.io>
- pyserial: <https://pyserial.readthedocs.io>; paho-mqtt: <https://eclipse.dev/paho/>; Mosquitto: <https://mosquitto.org>
- InfluxDB: <https://docs.influxdata.com>; Grafana: <https://grafana.com/docs>; Streamlit: <https://docs.streamlit.io>
- FMPy: <https://github.com/CATIA-Systems/FMPy>; FMI: <https://fmi-standard.org>; OpenModelica: <https://openmodelica.org>
- Eclipse Ditto: <https://eclipse.dev/ditto/>; BaSyx: <https://eclipse.dev/basyx/>; DTDL: <https://github.com/Azure/opendigitaltwins-dtdl>
- DeepXDE: <https://deepxde.readthedocs.io>; Cantera: <https://cantera.org>

Courses

- Willcox group, predictive digital twins: <https://kiwi.odn.utexas.edu/research/digital-twin>
- MIT OCW 2.51 (Intermediate Heat and Mass Transfer), 18.085/18.086 (Computational Science and Engineering), 2.29 (Numerical Fluid Mechanics).
- Aarhus University Centre for Digital Twin Technology publications.

Caveats

- **Industrial claims are not independently verified.** GE, Rolls-Royce, PSE/Siemens and Microsoft figures are company or vendor sources; cite them as claims. Academic cases are mostly single-lab demonstrations. The plasma-reactor twin is a recent, un-reviewed preprint.
- **Some sources are vendor-authored or conflicting.** The 2019 fixed-bed reactor paper is by Spatenka, Matzopoulos, Urban & Cano (as cited in MDPI *Processes* 13(6):1820), and its examples use PSE's own gPROMS AML:FBCR library, so it is a vendor-authored source. The 2020 pharma review is by Chen, Yang, Sampat, Bhalode, Ramachandran & Ierapetritou (*Processes* 8(9):1088). NIST database size figures conflict between sources.
- **Rule-of-thumb numbers are textbook figures,** not measurements of your materials: material k values, calibrated thermistor accuracy and diffusion times. Your own calibrated reference runs supersede them.
- **Check before adopting tools.** Library maintenance (e.g., FilterPy) and cloud-service availability change over time. Only the scope statements of the paywalled ASTM/ISO texts were checked; read the full standards via the library. Your rig will be "ASTM E1225-inspired", not compliant.
- **The timeline assumes the stated team and parts.** Procurement and steady-state test durations are the likeliest causes of slip, so order parts in week 1.