Stochastic Optimal Control: LQG, Kalman Filtering, Robust and Unscented Methods

Jouke van Westrenen

Georgia Tech

October 1, 2025

What is Stochastic Optimal Control?

- **Definition:** Choosing control actions u_t to optimize a performance criterion when **dynamics and observations are noisy**.
- **Key Ingredients:**
  - **Stochastic dynamics:**
  $$x_{t+1} = Ax_t + Bu_t + w_t, \quad w_t \sim \mathcal{N}(0, Q_w)$$
  - **Noisy measurements:**
  $$y_t = Cx_t + v_t, \quad v_t \sim \mathcal{N}(0, R_v)$$
  - **Objective:** minimize expected cost
  $$J = \mathbb{E}\left[\sum_{t=0}^{T} x_t^\top Q x_t + u_t^\top R u_t\right]$$
- **Goal:** Balance *performance* (cost minimization) and *robustness* (uncertainty handling).

Applications of Stochastic Optimal Control

- **Aerospace:** stabilize flight under turbulence and wind gusts
- **Robotics:** reliable navigation with uncertain sensors
- **Finance:** portfolio optimization under stochastic returns
- **Process Engineering:** control of chemical and energy systems under measurement noise and disturbances

Focus of This Talk

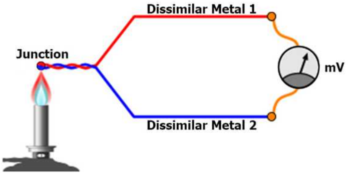
We will use **process control** as a running example to illustrate stochastic optimal control methods.

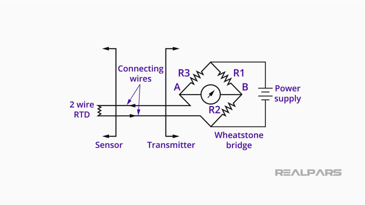
Why Stochastic Optimal Control?

- **Uncertainty is everywhere:**
  - **Sensor noise** $y_t = Cx_t + v_t$
  - **Model mismatch** A, B not perfectly known
  - **External disturbances** w_t
- **Limitations of deterministic control:**
  - Assumes perfect state knowledge
  - Ignores variability $\Rightarrow$ fragile performance
- **Advantage of stochastic optimal control:**
  - Explicitly accounts for noise and uncertainty
  - Balances performance and robustness
  - Produces *principled*, reliable decisions

How Do We Measure Temperature?

- **Thermocouples**
  - Voltage from junction of two metals
  - **Pros:** cheap, wide temperature range
  - **Cons:** lower accuracy, noise-sensitive
- **Resistance Temperature Detectors (RTDs)**
  - Resistance varies with temperature
  - **Pros:** accurate, stable
  - **Cons:** slower response, more costly
- **Infrared Sensors**
  - Detect emitted radiation (non-contact)
  - **Pros:** fast, non-invasive
  - **Cons:** emissivity issues, line-of-sight





Challenges in Temperature Measurement

- **Sensor-related**
  - Electronic noise in signals
  - Calibration drift over time
  - Limited bandwidth / lag in response
- **Environmental**
  - Airflow or convection currents
  - Heat radiation from surroundings
  - Ambient temperature fluctuations
- **Process-related**
  - Thermal inertia: slow heating/cooling response
  - Spatial gradients: one sensor cannot capture entire system
- **Implication:** Controllers operate on noisy, delayed, and incomplete temperature information.

How We Measure Things (and Why It Matters)

- Many ways to measure the same physical quantity (e.g., temperature, position, flow rate)
- Each method has **trade-offs**:
  - Accuracy vs. cost
  - Speed vs. stability
  - Robustness vs. sensitivity
- Choosing a measurement method defines the **errors and limitations** that will enter the control system
- The type of information available directly affects which control strategies are feasible

Why Different Stochastic Methods?

- No single stochastic control method handles all types of uncertainty equally well
- Examples:
  - **LQG:** optimal for linear systems with Gaussian noise
  - **Robust control:** ensures safety under worst-case model mismatch or bounded disturbances
  - **Unscented/iLQG:** deals with nonlinear dynamics and non-Gaussian uncertainty
- Selecting a method depends on:
  - System linearity
  - Noise distribution
  - Magnitude and type of uncertainties
  - Performance vs. safety requirements
- $\rightarrow$ These considerations will guide our discussion of each stochastic method

Lecture Roadmap

1. **Linear Quadratic Gaussian (LQG) Control**
   What is it, why it works, and the separation principle between estimation and control
2. **Kalman Filtering**
   Recursive state estimation, prediction and update steps, and why it is optimal for linear-Gaussian systems
3. **Robust Control under Uncertainty**
   Stochastic vs. worst-case approaches, $H\infty$ methods, handling model mismatch and adversarial disturbances
4. **Unscented Optimal Control**
   Advantages of sigma-point propagation, and iterative LQG (iLQG) for nonlinear stochastic systems

Linear Quadratic Gaussian (LQG) Control

Linear Quadratic Gaussian (LQG) Control: Overview

- **Purpose:** Compute optimal control for linear systems under Gaussian noise
- **Components:**
  - **Linear Quadratic Regulator (LQR):** finds control law minimizing quadratic cost
  - **Kalman Filter:** estimates true system states from noisy measurements
- **Separation Principle:** estimation and control can be designed independently
- **Cost function:** balances *performance* (tracking, regulation) and *control effort*:

$$J = \mathbb{E}\left[\sum_{t=0}^{T} x_t^\top Q x_t + u_t^\top R u_t\right]$$

What Makes a System Linear and Noise Gaussian?

- **Linear System:**
  - State evolves according to a linear relationship:

$$x_{t+1} = Ax_t + Bu_t + w_t$$

  - Output is a linear function of the state:

$$y_t = Cx_t + v_t$$

  - Key properties:
    - **Superposition:** response to a sum of inputs = sum of responses
    - **Scaling:** response scales proportionally with input

- **Gaussian Noise:**
  - Random disturbances with a **normal distribution**:

$$w_t \sim \mathcal{N}(0, Q), \quad v_t \sim \mathcal{N}(0, R)$$

  - Fully described by **mean** and **variance** (or covariance)
  - Common assumption because of the **Central Limit Theorem**: many small independent disturbances sum to an approximately Gaussian distribution

Linear Quadratic Regulator (LQR)

- **Purpose:** Optimal control for a **deterministic linear system** with full state knowledge
- **System:**
$$\dot{x}(t) = Ax(t) + Bu(t)$$
- **Quadratic cost to minimize:**
$$J = \int_0^\infty \left(x^T Q x + u^T R u\right) dt$$
- **Optimal control law (state feedback):**
$$u(t) = -Kx(t)$$
- **Gain matrix** K computed from the continuous-time **Riccati equation:**
$$A^T P + PA - PBR^{-1}B^T P + Q = 0$$
- Minimizes cost while balancing **performance** $(x^T Q x)$ and **control effort** $(u^T R u)$

Kalman Filter: State Estimation for LQG

- In LQG, the controller needs the **true system state** $x(t)$ to compute $u(t) = -Kx(t)$
- Often we only measure $y(t)$, which is **noisy**:

$$y(t) = Cx(t) + v(t)$$

- **Kalman filter role:** produce an optimal estimate $\hat{x}(t)$ from noisy measurements
- State estimate dynamics:

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L\big(y(t) - C\hat{x}(t)\big)$$

  - L = Kalman gain, chosen to **minimize estimation error variance**
  - $(y - C\hat{x})$ = innovation (new information from measurements)
- **Takeaway:** Kalman filter + LQR = LQG; the controller acts as if it knows the true state, using $\hat{x}(t)$

Separation Principle in LQG

- Key insight of LQG: **control design and state estimation can be done separately**.
- Two parts:
  1. **Controller (LQR):** Computes optimal control law assuming full state is known.
  2. **Estimator (Kalman Filter):** Reconstructs state from noisy measurements.
- **Separation Principle:**

$$u(t) = -K\hat{x}(t)$$

  where K comes from LQR and $\hat{x}(t)$ comes from the Kalman filter.
- This means:
  - We can design the controller and estimator *independently*.
  - Together, they form the optimal LQG controller.

Cost Function in LQG

- Quadratic cost to balance performance and effort:

$$J = \mathbb{E}\left[\int_0^\infty \Big(x(t)^T Q x(t) + u(t)^T R u(t)\Big)\, dt\right]$$

- Parameters:
  - Q: penalizes deviation from desired state
  - R: penalizes control effort (energy, actuator usage)
- How tuning affects the controller:
  - Large Q, small R $\rightarrow$ aggressive control (fast correction, high energy use)
  - Small Q, large R $\rightarrow$ conservative control (slower response, low energy use)
  - Balanced Q, R $\rightarrow$ trade-off between performance and efficiency
- Interpretation: The cost function **shapes controller behavior** by deciding which error is more costly.

Kalman Filtering

Kalman Filtering: Overview

- **Goal:** Reconstruct the hidden system state $x(t)$ from noisy, indirect measurements $y(t)$.
- Operates in a **recursive estimation**:
  1. **Prediction:** Use the system model to forecast the next state and its uncertainty.
  2. **Update:** Refine this forecast with the latest measurement.
- Key properties:
  - Recursive — no need to store all past data
  - Real-time capable — fast and efficient
  - **Optimal** under: linear dynamics + Gaussian noise
- **Why important for LQG:** Supplies the controller with the best possible state estimate.

Kalman Filter: Prediction Step

- **Goal:** Forecast the system state before seeing the next measurement
- State prediction:

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_{k-1}$$

  - Uses the **system model** (A, B) and last estimate
  - Think of it as: "Where should the system be now?"

- Uncertainty (covariance) prediction:

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q$$

  - Propagates uncertainty from last step
  - Q: extra uncertainty from process noise
  - Larger P = less confidence in prediction

- **Interpretation:** Prediction is like a forecast with error bars — good guess, but not yet corrected by real data.

Kalman Filter: Update Step (part 1)

- **Goal:** Correct the prediction using the latest measurement
- Measurement model:

$$y_k = Cx_k + v_k$$

  - y_k: measured output
  - C: links true state to measurement
  - v_k: measurement noise
- Compute Kalman gain:

$$K_k = P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}$$

  - Balances **trust in measurement vs. prediction**
  - R: measurement noise covariance

Kalman Filter: Update Step (part 2)

- Update with correction:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - C\hat{x}_{k|k-1})$$

  - Prediction + adjustment (“innovation” term)
- Update confidence:

$$P_{k|k} = (I - K_k C)P_{k|k-1}$$

  - Uncertainty shrinks after using the measurement
  - More accurate estimate than prediction alone

Example: LQR vs LQG Temperature Tracking

- **System:** Simplified thermal model with state $x_{k+1} = Ax_k + Bu_k$
- **Control goal:** Track step changes in temperature setpoint
- **Key elements:**
  - State: temperature (°C)
  - Control input: heating power
  - Measurement noise: random fluctuations in thermometer readings
  - Step disturbances: setpoint changes from 60°C $\rightarrow$ 80°C $\rightarrow$ 70°C
- **Controllers compared:**
  - **Naive LQR:** Feedback directly on noisy measurements
  - **LQG:** LQR + Kalman filter state estimation
- **Outcome:**
  - Naive LQR reacts to measurement noise (jittery response)
  - LQG achieves smooth tracking by filtering noise

Robust Control

Robust Control: Overview

- **Core idea:** Ensure stability and performance even when the model is uncertain or disturbances are stronger than expected.
- **Two main philosophies:**
  - *Stochastic approaches:* Assume uncertainties are random (e.g. noise with known statistics).
  - *Worst-case approaches:* Plan for the most adverse model mismatch or disturbance.
- **Key methods:**
  - H_∞ control – balance performance against worst-case disturbances.
  - μ-synthesis – structured uncertainty modeling.
  - Robust MPC – predictive control under bounded uncertainty.
- **Applications:** Systems with uncertain parameters, unmodeled dynamics, or adversarial environments.

Stochastic vs. Worst-Case Approaches

- **Stochastic view:**
  - Uncertainty modeled as random noise with known distribution.
  - Example equation: $x_{k+1} = Ax_k + Bu_k + w_k, \;\; w_k \sim \mathcal{N}(0, Q)$
  - In the code example: measurement noise (thermocouple error) $\Rightarrow$ handled well by Kalman filtering (LQG).

- **Worst-case view:**
  - Uncertainty treated as an unknown but bounded disturbance.
  - Example equation: $x_{k+1} = Ax_k + Bu_k + d_k, \;\; \|d_k\| \leq \delta$
  - In the code example: process noise could be seen as an adversarial disturbance (e.g. sudden feed composition change) $\Rightarrow$ worst-case design (e.g. H_∞) ensures stability even under the largest possible d_k.

- **Key difference:** Stochastic = *optimize average performance under probabilistic noise*, Worst-case = *guarantee performance under all admissible disturbances*.

H∞ Control in State-Space

- Goal: Design $u(t)$ such that the effect of worst-case disturbances $w(t)$ on critical outputs $z(t)$ is minimized.
- Standard setup:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_u u(t) + B_w w(t), \\ z(t) &= C_z x(t) + D_{zu} u(t) + D_{zw} w(t), \\ y(t) &= C_y x(t) + D_{yu} u(t) + D_{yw} w(t). \end{aligned}$$

- Robustness condition: energy gain from $w(t)$ to $z(t)$ is bounded

$$\|T_{zw}(s)\|_\infty < \gamma.$$

- Interpretation: **limit the worst-case amplification of disturbances through the state dynamics**.

μ-Synthesis: Structured Uncertainty Handling

- **Structured Uncertainties:**
  - Represent specific, known forms of uncertainty in the system.
- **Concept:**
  - The *structured singular value* μ quantifies the smallest uncertainty that can destabilize the system.
  - If $\mu < 1$ across frequencies, the system is robust to all allowed uncertainties.
  - Provides a frequency-dependent measure of robustness.
- **Method (D-K Iteration):**
  1. **D-step:** Find a scaling matrix D that bounds the effect of uncertainties on the system.
  2. **K-step:** Design a controller K that minimizes the closed-loop μ (maximizes robustness) under the current scaling.
  3. Repeat D-K iteration until convergence.
- **Key idea:** Iteratively shape the controller and scale uncertainties to ensure robust stability and performance under all structured uncertainties.

Robust MPC: Predictive Control under Bounded Uncertainty

- **Bounded Uncertainties:**
  - Uncertainties are assumed *unknown but constrained* within known bounds.
- **Concept:**
  - MPC predicts future system behavior over a finite horizon using the model.
- **Method:**
  1. Define a cost function (tracking + control effort) over the prediction horizon.
  2. Formulate constraints to hold *robustly* for all $w_k \in \mathcal{W}$.
  3. Solve an optimization problem (often convex) to find control inputs.
  4. Apply the first input, measure new state, and repeat.
- **Key idea:** Anticipate all possible disturbances within bounds and ensure constraints are satisfied, achieving robust stability and performance in a predictive framework.

LQG vs H∞ Control: Persistent Disturbance Example

Scenario: Compare performance of LQG and H∞ controllers under a structured, persistent disturbance with measurement noise.

- **System:** Single-state process with dynamics

$$\dot{x} = -a(x - x_{\text{set}}) + bu + w_{\text{process}}, \quad y = x + v_{\text{meas}}$$

  where $w_{\text{process}} = 2\sin(0.5t) +$ Gaussian noise and $v_{\text{meas}} \sim \mathcal{N}(0, 15^2)$.

- **Setpoint:** Step changes:

$$x_{\text{set}}(t) = \begin{cases} 0, & t < 0.1T \\ 50, & 0.1T \leq t < 0.4T \\ 20, & 0.4T \leq t < 0.6T \\ 50, & t \geq 0.6T \end{cases}$$

- **Controller Design:**
  - **LQG:** Optimal state-feedback + Kalman filter
  - **H∞:** Aggressive gain ($K_\infty \approx 2K_{\text{LQR}}$) to attenuate worst-case disturbances

Unscented Control

Why Unscented Optimal Control?

Motivation: Choosing UOC over LQG or Robust Control

Compared to LQG:

- LQG assumes linear dynamics or relies on linearization (EKF)
- Linearization can introduce significant errors for strongly nonlinear chemical processes
- UOC propagates a set of *sigma points*, capturing nonlinear effects accurately up to second order

Compared to Robust / H∞ Control:

- H∞ focuses on worst-case disturbances, often leading to overly conservative control
- UOC optimizes expected performance under stochastic noise, balancing control effort and robustness

Sigma Points and Iterative LQG (iLQG)

Sigma-Point Propagation:

- Approximates how the mean and covariance of the state evolve through nonlinear dynamics
- The new mean and covariance are reconstructed from sigma points, giving a more accurate estimate than linearization

Iterative LQG (iLQG):

- Solves nonlinear stochastic optimal control by iteratively linearizing around a nominal trajectory
- Computes a locally optimal feedback policy at each iteration
- Integrates naturally with sigma-point propagation to account for stochastic uncertainty

Takeaway:

- "Unscented" = captures nonlinear stochastic effects without linearization
- Sigma points + iLQG = practical method for near-optimal control of nonlinear processes with uncertainty

Sigma-Point Propagation in Unscented Control

How Sigma Points Work:

1. Start with the current state mean $\bar{x}$ and covariance P_x.
2. Generate a set of $2n+1$ sigma points (for an n-dimensional state):

$$\chi_0 = \bar{x}, \quad \chi_i = \bar{x} + (\sqrt{(n+\lambda)P_x})_i, \quad \chi_{i+n} = \bar{x} - (\sqrt{(n+\lambda)P_x})_i$$

   where $(\sqrt{(n+\lambda)P_x})_i$ is the i-th column of the matrix square root of $(n+\lambda)P_x$.
3. Propagate each sigma point through the nonlinear function $f(\cdot)$:

$$\chi_i' = f(\chi_i)$$

4. Recompute the predicted mean and covariance:

$$\bar{x}' = \sum_{i=0}^{2n} W_i^m \chi_i', \quad P_x' = \sum_{i=0}^{2n} W_i^c (\chi_i' - \bar{x}')(\chi_i' - \bar{x}')^T$$

Iterative LQG (iLQG) for Nonlinear Stochastic Control

How iLQG Works:

1. Start with a nominal trajectory $\{x_k^0, u_k^0\}$.
2. Linearize the nonlinear dynamics $x_{k+1} = f(x_k, u_k)$ around the trajectory:
$$\delta x_{k+1} \approx A_k \delta x_k + B_k \delta u_k$$
where $A_k = \frac{\partial f}{\partial x}|_{x_k^0, u_k^0}, \quad B_k = \frac{\partial f}{\partial u}|_{x_k^0, u_k^0}$.
3. Quadratic approximation of the cost function around the nominal trajectory:
$$J \approx \sum_k \frac{1}{2}\delta x_k^T Q_k \delta x_k + \frac{1}{2}\delta u_k^T R_k \delta u_k$$
4. Solve the linear-quadratic problem to compute feedback gains K_k and feedforward updates δu_k.
5. Update the nominal trajectory and iterate until convergence.

Example: Unscented vs. Robust Control in a Nonlinear Reactor

- **System:** Continuous stirred-tank reactor (CSTR) with strongly nonlinear Arrhenius heat release.
- **Challenge:** Maintain reactor temperature under exothermic reaction with uncertain kinetics and noisy sensors.
- **Comparison:**
  - **Robust control (H∞):** Designed from a fixed linear model; struggles as reaction nonlinearity grows.
  - **Unscented control (UOC):** Uses sigma points to propagate full nonlinear dynamics.
- **Unscented control loop:**
  1. Generate sigma points around temperature and concentration states.
  2. Propagate sigma points through nonlinear reactor model (Arrhenius kinetics).
  3. Fuse with noisy temperature measurements to update state estimate.
  4. Compute optimal heater input for temperature tracking.