TEK4090

Introduction to Modern Control

Lecture 4: Estimation

Mathias Hudoba de Badyn

October 2, 2025

Some notes on Norms

Some notes on Norms

Some notes on Norms

Optimal Control

Static Optimization

$$\begin{array}{ll} \underset{x \in \mathbb{R}^n}{min} & \frac{1}{2} x^T P x \\ \text{s.t.} & Ax = b \end{array}$$

Dynamic Optimization

$$\begin{array}{ll} \underset{x \in \mathbb{R}^n \times [0,T]}{\min} & \frac{1}{2} \sum_{k=1}^{T} \left[x_k^T P_k x_k + u_k^T R_k u_k \right] \\ \text{s.t.} & x_{k+1} = \bar{A} x_k + \bar{B} u_k \end{array}$$

Optimal **points** versus optimal **sequence**

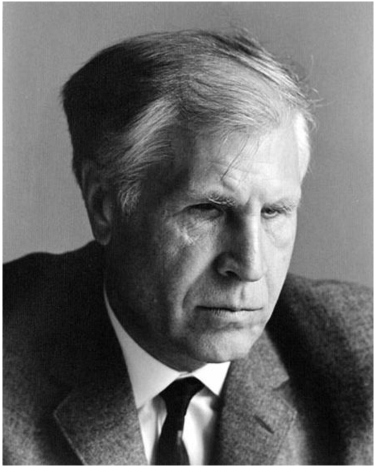
Discrete-Time LQR

Discrete-Time LQR

Discrete-Time LQR

Pontryagin Maximum Principle

What about the **KKT conditions** for **dynamic problems**?



Super general problem:

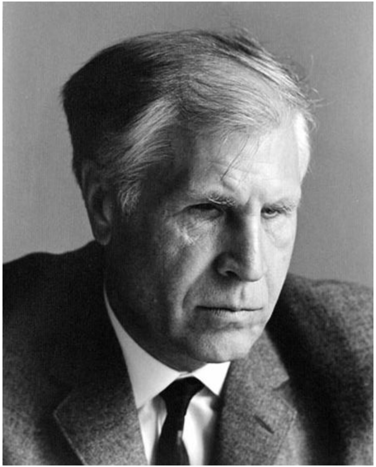
$$\begin{array}{ll} \min & \phi(x(T), T) + \int_0^T L(x, u, t)dt \\ \text{s.t.} & \psi(x(T), T) = 0 \\ & \dot{x} = f(x, u, t) \end{array}$$

Define the **Hamiltonian:**

$$H(x, u, \lambda, t) := L(x, u, \lambda, t) + \lambda(t)^T f(x, u, t)$$

Pontryagin Maximum Principle

What about the **KKT conditions** for **dynamic problems**?



$$H(x, u, \lambda, t) := L(x, u, \lambda, t) + \lambda(t)^T f(x, u, t)$$

Pontryagin's maximum principle:

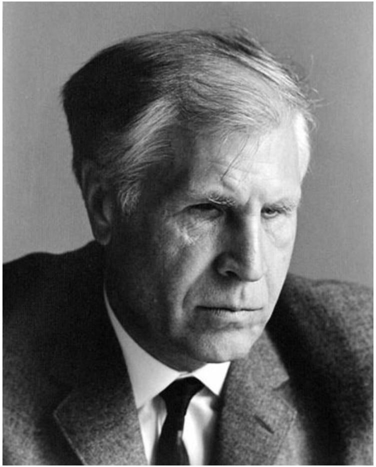
$$H(x^*, u^*, \lambda^*, t) \leq H(x^*, u, \lambda^*, t), \ \forall t \in [0, T]$$

Coupled DEs:

$$\dot{x} = \frac{\partial H}{\partial \lambda} = f$$
$$-\dot{\lambda} = \frac{\partial H}{\partial x} = \frac{\partial f}{\partial x}^T \lambda + \frac{\partial L}{\partial x}$$

Pontryagin Maximum Principle

What about the **KKT conditions** for **dynamic problems**?



Coupled DEs:

$$\dot{x} = \frac{\partial H}{\partial \lambda} = f$$

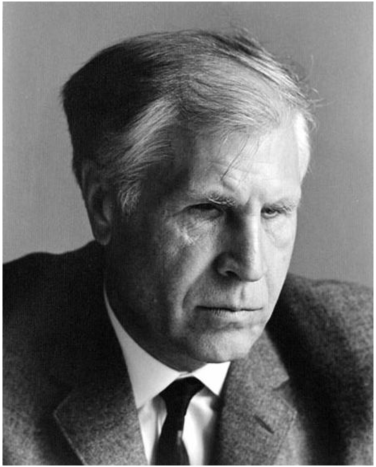
$$-\dot{\lambda} = \frac{\partial H}{\partial x} = \frac{\partial f}{\partial x}^T \lambda + \frac{\partial L}{\partial x}$$

Boundary conditions:

$$0 = \frac{\partial H}{\partial u} = \frac{\partial L}{\partial u} + \frac{\partial f}{\partial u}^T \lambda$$

Pontryagin Maximum Principle

What about the **KKT conditions** for **dynamic problems**?



Coupled DEs:

$$\dot{x} = \frac{\partial H}{\partial \lambda} = f$$
$$-\dot{\lambda} = \frac{\partial H}{\partial x} = \frac{\partial f}{\partial x}^T \lambda + \frac{\partial L}{\partial x}$$

Boundary conditions:

$$\left(\phi_x + \psi_x^T v - \lambda\right)^T \Big|_T dx(T) + \left(\phi_t + \psi_t^T v + H\right) \Big|_T dT = 0$$

Continuous-Time LQR vs Discrete-Time LQR

Discrete-Time LQR

$$\begin{aligned} \min_{x \in \mathbb{R}^n \times [0,T]} \quad & \frac{1}{2} \sum_{k=1}^{T} \left[x_k^T P_k x_k + u_k^T R_k u_k \right] \\ \text{s.t.} \quad & x_{k+1} = \bar{A} x_k + \bar{B} u_k \end{aligned}$$

Solution:

$$\begin{aligned} J_k^* &= \frac{1}{2} x_k^T S_k x_k \\ S_k &= \bar{A} T S_{k+1} \bar{A} - \bar{A}^T S_{k+1} \bar{B} K_k + P_k \\ K_k &= (B^T S_{k+1} B + R_k)^{-1} B^T S_{k+1} \bar{A} \\ u_k^* &= -K_k x_k \end{aligned}$$

Continuous-Time LQR

$$\begin{aligned} \min_{x \in \mathbb{R}^n \times \mathcal{T}} \quad & \frac{1}{2} \int_{\mathcal{T}} \left[x^T P x + u^T R u \right] dt \\ \text{s.t.} \quad & \dot{x} = Ax + Bu \end{aligned}$$

Solution:

$$\begin{aligned} J^*(t) &= \frac{1}{2} x(t)^T S(t) x(t) \\ -\dot{S} &= SA + A^T S - SBR^{-1} B^T S + Q \\ K &= -R^{-1} B^T S \\ u &= -Kx \end{aligned}$$

Motivation for MPC

In practice, LQR can tune gains really nicely

Some cons:

- Cannot handle constraints
- Computationally intensive (finite time-horizon)
- No guaranteed margins when noise in system and estimation loops

Guaranteed Margins for LQG Regulators

JOHN C. DOYLE

***Abstract*—There are none.**

Motivation for MPC

$$
\begin{aligned}
\min_{x \in \mathbb{R}^n \times [0,T]} \quad & \frac{1}{2} \sum_{k=1}^{T} \left[x_k^T P_k x_k + u_k^T R_k u_k \right] \\
\text{s.t.} \quad & x_{k+1} = \bar{A} x_k + \bar{B} u_k \\
& x_k \in \mathcal{X}_k \\
& u_k \in \mathcal{U}_k
\end{aligned}
$$

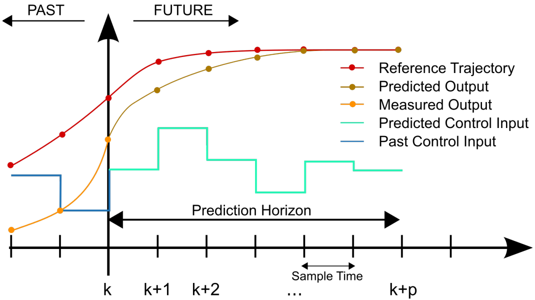
Model Predictive Control

MPC

- At each timestep k solve:

$$
\begin{aligned}
\min \quad & \sum_{t=k}^{k+p} c_k(x_k, u_k) \\
\text{s.t.} \quad & x_{t+1} = f(x_t, u_t) \\
& x_k = x(k) \\
& x_t \in \mathcal{X} \\
& u_t \in \mathcal{U}
\end{aligned}
$$

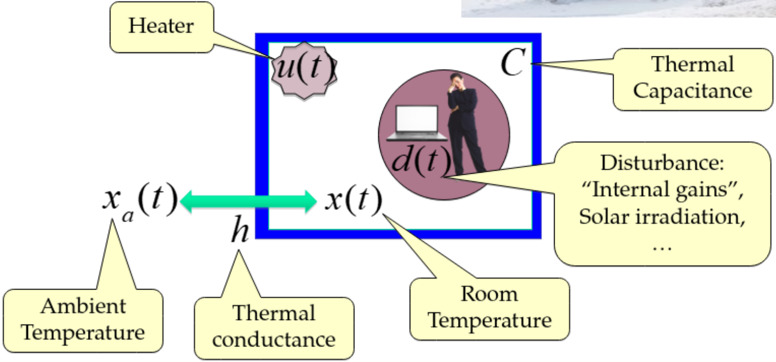
- Apply the first control input $u(k) = u_k$
- $k \mapsto k+1$
- Repeat



Building Control

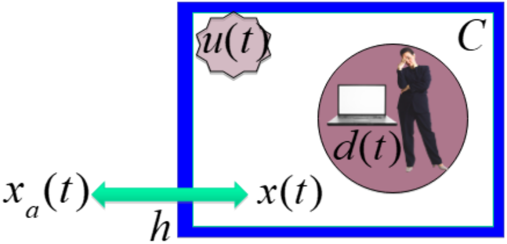
Building Control

Single room house



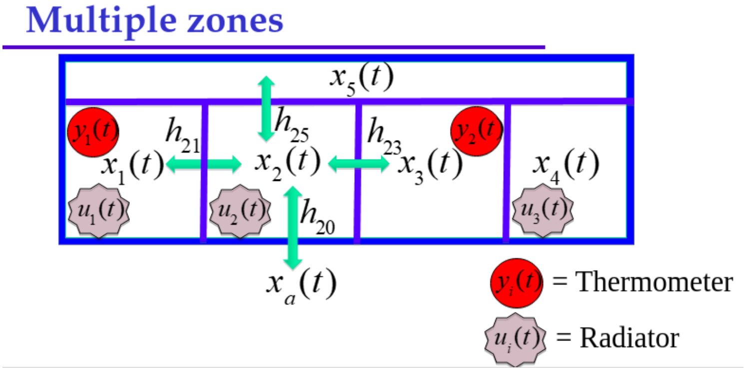
Modeling Buildings – Single Room

$$C\frac{d}{dt}x(t) = h\left[x_a(t) - x(t)\right] + u(t) + d(t)$$



- C: Thermal capacity of the room
- h: Insulation of the wall
- $u(t)$: Heat input at time t

Modeling Buildings – Multiple Zones



$$
\begin{aligned}
C_2\dot{x}_2(t) = \quad & h_{20}(x_a(t) - x_2(t)) + h_{21}(x_1(t) - x_2(t)) + \\
& h_{23}(x_3(t) - x_2(t)) + h_{25}(x_5(t) - x_2(t)) + u_2(t) + d_2(t)
\end{aligned}
$$

Modeling Buildings – Multiple Zones

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \end{bmatrix}, \quad u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{bmatrix}, \quad d(t) = \begin{bmatrix} d_1(t) \\ d_2(t) \\ d_3(t) \\ d_4(t) \\ d_5(t) \end{bmatrix} \tag{1}$$

$$H_{ij} = \begin{cases} \frac{h_{ij}}{C_i} & \text{if } i \text{ is a neighbour of } j \\ 0 & \text{else} \end{cases} \tag{2}$$

$$H_{ii} = \sum_{j=1,\ldots,5} H_{ij} \tag{3}$$

Modeling Buildings – Multiple Zones

$$\dot{x}(t) = \begin{bmatrix} -H_{11} & H_{12} & 0 & 0 & H_{15} \\ H_{21} & -H_{22} & H_{23} & 0 & H_{25} \\ 0 & H_{32} & H_{33} & H_{34} & H_{35} \\ 0 & 0 & H_{43} & -H_{44} & H_{45} \\ H_{51} & H_{52} & H_{53} & H_{54} & -H_{55} \end{bmatrix} x(t) + \begin{bmatrix} H_{10} \\ H_{20} \\ H_{30} \\ H_{40} \\ H_{50} \end{bmatrix} x_a(t) + \tag{4}$$

$$\begin{bmatrix} \frac{1}{C_1} & 0 & 0 \\ 0 & \frac{1}{C_2} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{C_4} \\ 0 & 0 & 0 \end{bmatrix} u(t) + \begin{bmatrix} \frac{1}{C_1} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{C_2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{C_3} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{C_4} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{C_5} \end{bmatrix} d(t) \tag{5}$$

Modeling Buildings – Multiple Zones

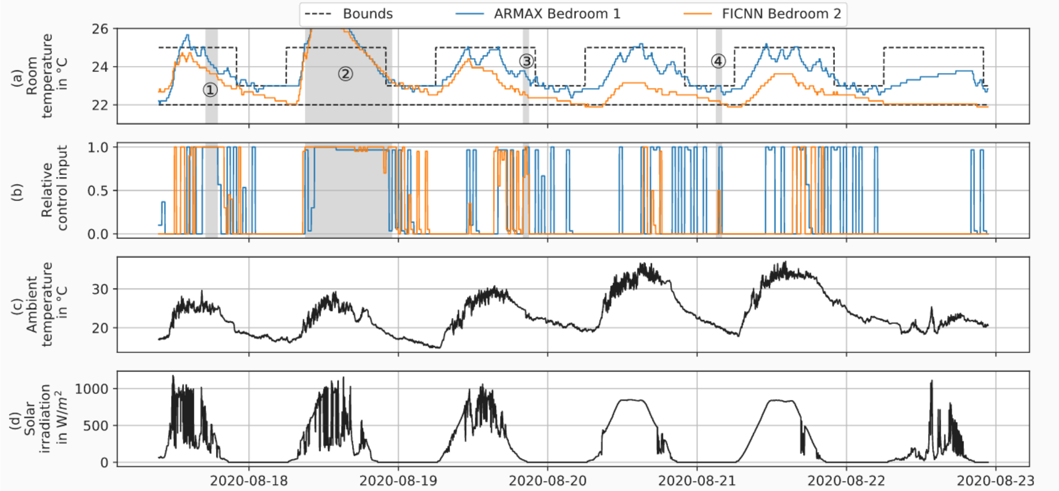
$$\dot{x} = Ax(t) + bx_a(t) + Bu(t) + Dd(t) \tag{6}$$

$$y(t) = Cx(t) + cx_a(t) \tag{7}$$

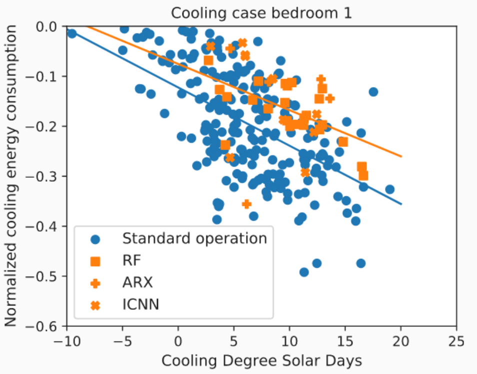
Example of Building MPC

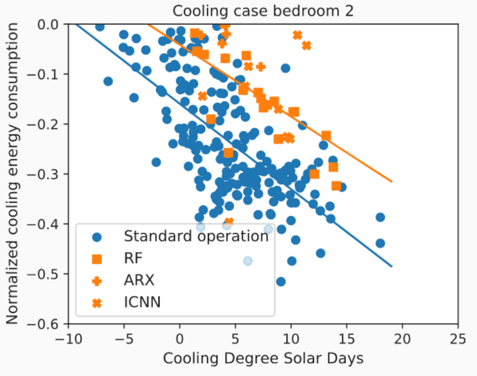
Empa
Materials Science and Technology

Performance of ICNNs In the Loop



Comparison with other ML Techniques + Hysteresis Control

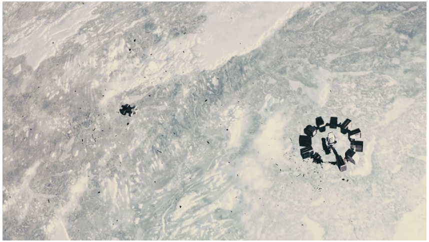




Punchline: MPC consumes 33% less cooling energy at 5 CDSD and 28% less at 15 CDSD.

[0] Felix Bünning, Benjamin Huber, Adrian Schalbetter, Ahmed Aboudonia, Mathias Hudoba de Badyn, Philipp Heer, Roy S Smith, John Lygeros. *Physics-informed linear regression is competitive with two Machine Learning methods in residential building MPC*. Applied Energy 310 (2022) 118491.

Next Time



- Observability ("Where am I?" from sensors)
- Kalman Filter (more black magic)
- Spacecraft GNC

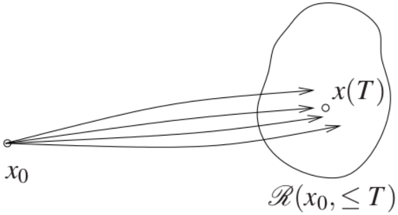
Recap from Last Time

- Optimal control
- Linear Quadratic Regulator
- MPC and Building Control

Reading Assignment

- **Åstrom & Murray:** Ch. 7
- **Crassidis & Junkins:**
  - Kalman filters: Ch. 3.1–3.3
  - Spacecraft Dynamics: Ch. 6.1, Appendix A.7-A.8
  - Probability: Appendix C

Reachability



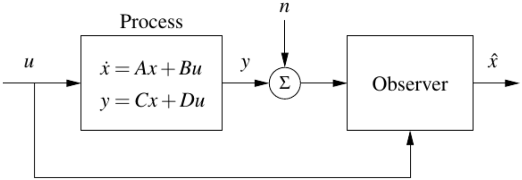
The **reachable set** $\mathcal{R}(x_0, T)$ is the set of all points x_f such that there exists an input $u(t)$ that steers the system from $x(0) = x_0$ to $x(T) = x_f$.

A system is **reachable** if for any $x_0, x_f \in \mathbb{R}^n$, there exists $T > 0$ and $u : [0, T] \mapsto \mathbb{R}$ such that $x(0) = x_0$ and $x(T) = x_f$.

Observability

Feedback requires a measurement of the system.
How do we use measurements of a system in the control loop?

- Consider the LTI system (A, B, C, D)
- Suppose y is corrupted by noise n
- We can construct an **observer** to produce the estimate $\hat{x}$



Observability

Challenges/Questions:

1. How ‘rich’ does our measurement need to be in order to be useful?
2. How do you design an observer?
3. How does the interconnection between the observer and the controller work?

Observability

Answer to **Challenge 1:**

Definition:

A system $\Sigma := (A, B, C, D)$ is said to be **observable** if, for any $T > 0$, it is possible to reconstruct the state $x(T)$ from measurements of $y(t)$ and $u(t)$ on $[0, T]$.

So, when is a system observable?

Observability

It's useful just to consider the autonomous system

$$\begin{aligned}\dot{x} &= Ax \\ y &= Cx.\end{aligned}$$

Immediately note: if C is invertible, then we can recover $x(t)$ from $x(t) = C^{-1}y(t)$.

In general, C is rank-deficient and can have fewer rows than x.

Observability

Observability

Observability

Theorem:
The system $\Sigma := (A, B, C, D)$ is observable if and only if the observability gramian W_o has full rank.

Observable Canonical Form

$$\dot{z} = \begin{bmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ -a_{n-1} & 0 & 0 & \cdots & 1 \\ -a_n & 0 & 0 & \cdots & 0 \end{bmatrix} z + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix} z + Du$$

Observable Canonical Form

$$W_o = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -a_1 & 1 & 0 & \cdots & 0 \\ -a_1^2 - a_2 & -a_1 & 1 & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ * & * & & \cdots & 1 \end{bmatrix} \qquad W_o^{-1} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ a_1 & 1 & 0 & \cdots & 0 \\ a_2 & a_1 & 1 & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ a_{n-1} & a_{n-2} & a_{n-3} & \cdots & 1 \end{bmatrix}$$

Observers & State Estimation

Answer to Challenge 2:

- Design a dynamical system for $\dot{\hat{x}}$
- Make $\hat{x}(t) \to x(t)$ as $t \to \infty$

Observers & State Estimation

Make a copy of the system and propagate the estimate with the right input

Original System

$$\dot{x} = Ax + Bu$$

Copy of System

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

Observers & State Estimation: Error Dynamics

Observers & State Estimation: Error Dynamics

Duality

Controller:

$$A - BK$$

Observer:

$$A - LC$$

$$A \leftrightarrow A^T$$

$$B \leftrightarrow C^T$$

$$L \leftrightarrow L^T$$

$$W_r \leftrightarrow W_o^T$$

(A, B) reachable iff (A^T, B^T) observable; (A, C) observable iff (A^T, C^T) reachable

Observer Design by Eigenvalue Assignment

$$\frac{dx}{dt} = Ax + Bu$$
$$y = Cx$$
$$\frac{d\hat{x}}{dt} = A\hat{x} + Bu + L(y - C\hat{x})$$

$$\lambda(s) = s^n + a_1 s^{n-1} + \cdots + a_n$$

$$L = W_o^{-1} \tilde{W}_o \begin{bmatrix} p_1 - a_1 \\ \cdots \\ p_n - a_n \end{bmatrix}$$

$$\tilde{W}_o = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ a_1 & 1 & \ddots & 0 \\ a_2 & a_1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-2} & a_{n-3} & \cdots & 0 \\ a_{n-1} & a_{n-2} & \cdots & 1 \end{bmatrix}^{-1}$$

$$p(s) = s^n + p_1 s^{n-1} + \cdots + p_n$$

Challenge 3:

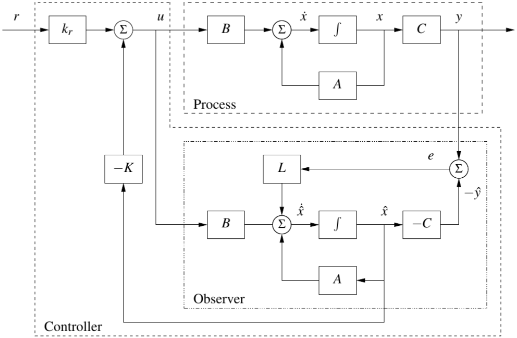
How does the controller work with estimated states?

Control with Estimated State

Control with Estimated State

Separation Principle

The controller and observer gains can be designed **independently** of each other



$$\frac{d}{dt}\begin{bmatrix} x \\ \tilde{x} \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix}\begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + \begin{bmatrix} Bk_r \\ 0 \end{bmatrix} r$$

$$\lambda(s) = \det(sI - A + BK)\det(sI - A + LC)$$

Kalman Decomposition

Distinct Eigenvalues

$$\frac{dx}{dt} = \begin{bmatrix} A_{ro} & 0 & 0 & 0 \\ 0 & A_{r\bar{o}} & 0 & 0 \\ 0 & 0 & A_{\bar{r}o} & 0 \\ 0 & 0 & 0 & A_{\bar{r}\bar{o}} \end{bmatrix} x + \begin{bmatrix} B_{ro} \\ B_{r\bar{o}} \\ 0 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} C_{ro} & 0 & C_{\bar{r}o} & 0 \end{bmatrix} x + Du$$

Non-Distinct Eigenvalues

$$\frac{dx}{dt} = \begin{bmatrix} A_{ro} & 0 & * & 0 \\ * & A_{r\bar{o}} & * & * \\ 0 & 0 & A_{\bar{r}o} & 0 \\ 0 & 0 & * & A_{\bar{r}\bar{o}} \end{bmatrix} x + \begin{bmatrix} B_{ro} \\ B_{r\bar{o}} \\ 0 \\ 0 \end{bmatrix} u$$

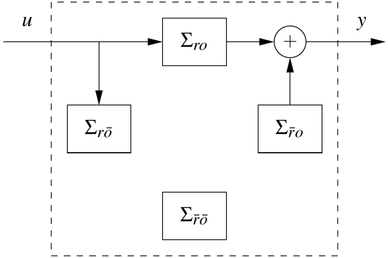
$$y = \begin{bmatrix} C_{ro} & 0 & C_{\bar{r}o} & 0 \end{bmatrix} x + Du$$

Kalman Decomposition

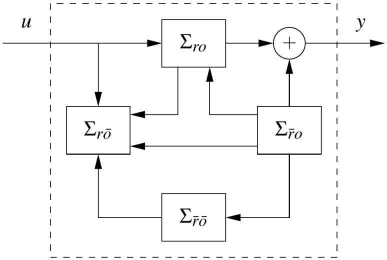
Distinct Eigenvalues

$$\frac{dx}{dt} = \begin{bmatrix} A_{ro} & 0 & 0 & 0 \\ 0 & A_{r\bar{o}} & 0 & 0 \\ 0 & 0 & A_{\bar{r}o} & 0 \\ 0 & 0 & 0 & A_{\bar{r}\bar{o}} \end{bmatrix} x + \begin{bmatrix} B_{ro} \\ B_{r\bar{o}} \\ 0 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} C_{ro} & 0 & C_{\bar{r}o} & 0 \end{bmatrix} x + Du$$



Kalman Decomposition



Non-Distinct Eigenvalues

$$\frac{dx}{dt} = \begin{bmatrix} A_{ro} & 0 & * & 0 \\ * & A_{r\bar{o}} & * & * \\ 0 & 0 & A_{\bar{r}o} & 0 \\ 0 & 0 & * & A_{\bar{r}\bar{o}} \end{bmatrix} x + \begin{bmatrix} B_{ro} \\ B_{r\bar{o}} \\ 0 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} C_{ro} & 0 & C_{\bar{r}o} & 0 \end{bmatrix} x + Du$$

Back to Challenge 2:

We saw optimal gains for control, what about observers?

How to choose the observer gain optimally?

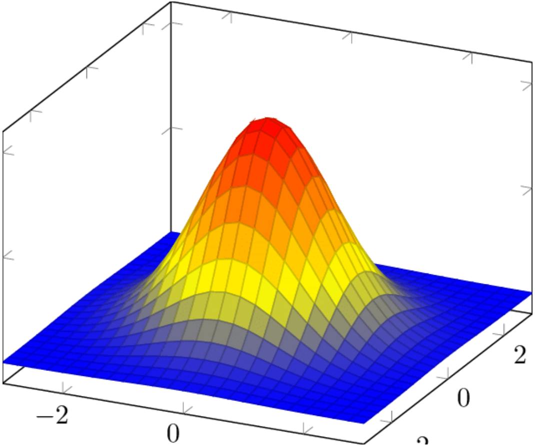
In general, measurements are **noisy**.

$$
\begin{aligned}
x_{k+1} &= Ax_k + Bu_k + Fv_k \\
y_k &= Cx_k + Du_k + w_k
\end{aligned}
$$

$$E[v_k] = 0, \;\; E[w_k] = 0$$

$$E[v_k v_{k+1}^T] = \begin{cases} 0 & k \neq j \\ Q_k & k = j \end{cases}$$

$$E[w_k w_{k+1}^T] = \begin{cases} 0 & k \neq j \\ R_k & k = j \end{cases}$$

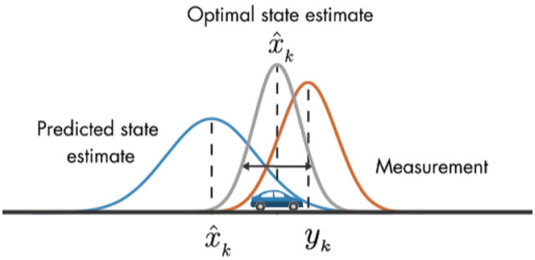


How to choose the observer gain?

With noisy measurements, the estimate is now a **random variable**.

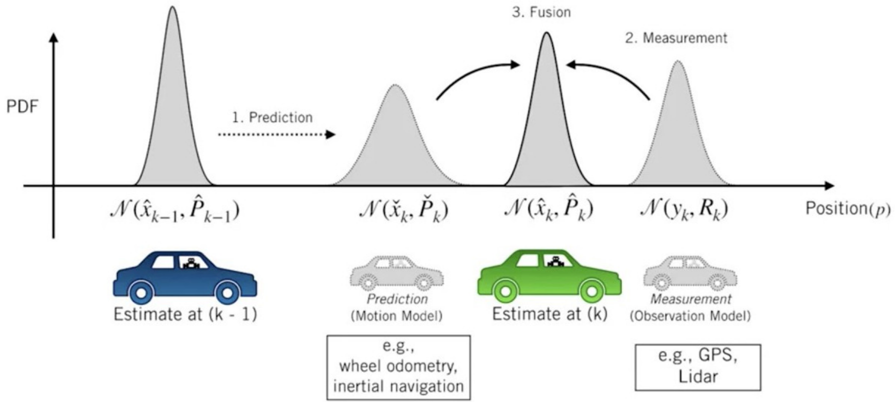
If L stabilizes the observer dynamics, then

$$E[e_k] = E[\hat{x}_k - x_k] \to 0.$$



Instead, we can optimize the 'width' (or, covariance) of the estimate

Optimal Observers



Kalman Filter Derivation

Kalman Filter Derivation

Kalman Filter Derivation

Kalman Filter Derivation

Kalman Filter Derivation

Kalman Filter Derivation

Kalman Filter Derivation

Kalman Filter

Model:
$$x_{k+1} = Ax_k + Bu_k + Fv_k$$
$$y_k = Cx_k + Du_k + w_k$$

Initialize:
$$\hat{x}_0 = x_0, \;\; P_0 = P$$

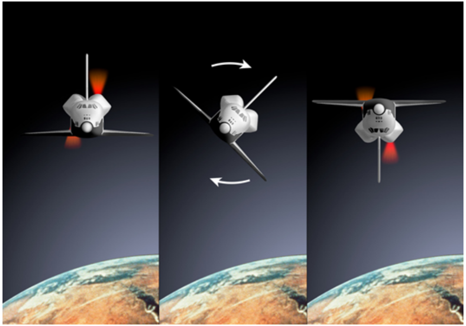
Gain:
$$K_k = P_k^- C^T \left(CP_k^- C^T + R_k \right)^{-1}$$

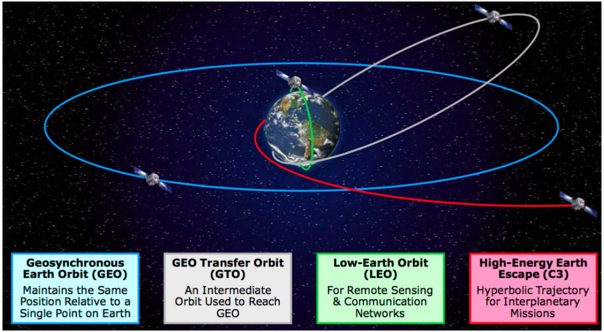
Update:
$$\hat{x}_k^+ = \hat{x}_k^- + K_k \left(\tilde{y}_k - C\hat{x}_k^- \right)$$
$$P_k^+ = (I - K_k C) P_k^-$$

Propagate:
$$\hat{x}_{k+1}^- = A\hat{x}_k^+ + Bu_k$$
$$P_{k+1}^- = AP_k^+ A^T + BQ_k B^T$$

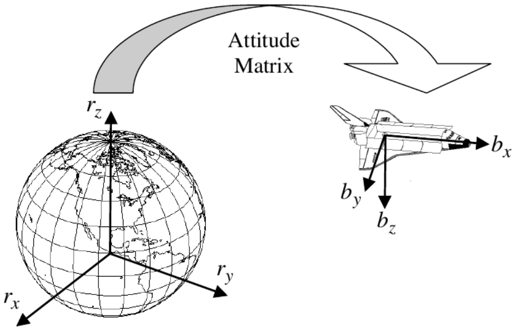
Spacecraft GNC

Spacecraft GNC





Attitude



Reference and body frames:

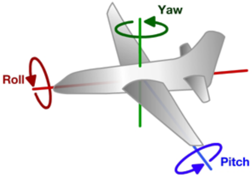
$$b = b_x \hat{b}_1 + b_y \hat{b}_2 + b_z \hat{b}_3$$
$$r = r_x \hat{r}_1 + r_y \hat{r}_2 + r_z \hat{r}_3$$

Transformation between the two:

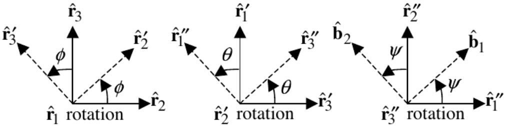
$$\begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \end{bmatrix} = A \begin{bmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \hat{r}_3 \end{bmatrix} \tag{8}$$

Euler Angles

One can represent a rotation via the **Euler angles:** roll ϕ, pitch θ, and yaw ψ.

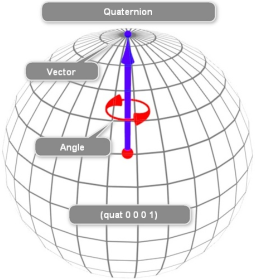


$$r' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} r \qquad r'' = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} r' \qquad b = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} r''$$



Quaternions

The Euler angles have a singularity when the attitude is pointed at a pole, so spacecraft attitude is represented as a **quaternion** instead.



$$q = \begin{bmatrix} \rho \\ q_4 \end{bmatrix} \in \mathbb{R}^4$$

$$q^T q = 1$$

$$A(q) = \Xi^T(q)\Psi(q)$$

$$\Xi(q) = \begin{bmatrix} q_4 I_{3\times 3} + [\rho\times] \\ -\rho^T \end{bmatrix}$$

$$\Psi(q) = \begin{bmatrix} q_4 I_{3\times 3} - [\rho\times] \\ -\rho^T \end{bmatrix}$$

$$[\rho\times] = \begin{bmatrix} 0 & -\rho_3 & \rho_2 \\ \rho_3 & 0 & -\rho_1 \\ -\rho_2 & \rho_1 & 0 \end{bmatrix}$$

Rigid Body Dynamics

- H: Angular momentum
- J: Moment of Inertia
- L: Applied torque
- ω: Angular velocity

Dynamics: Euler Angles

$$\begin{aligned}\dot{H} &= J\dot{\omega} \\ &= -[\omega\times]J\omega + L\end{aligned}$$

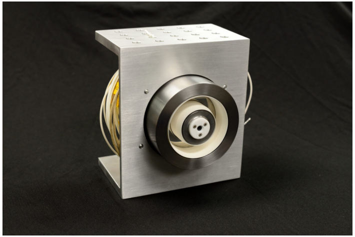
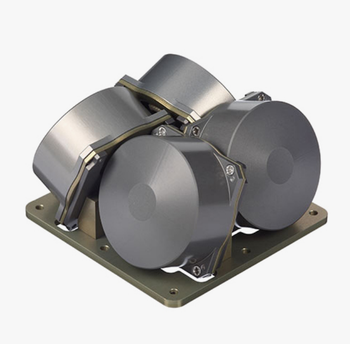
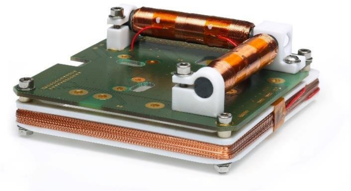
Kinematics: Quaternions

$$\dot{q} = \frac{1}{2}\Omega(\omega)q$$

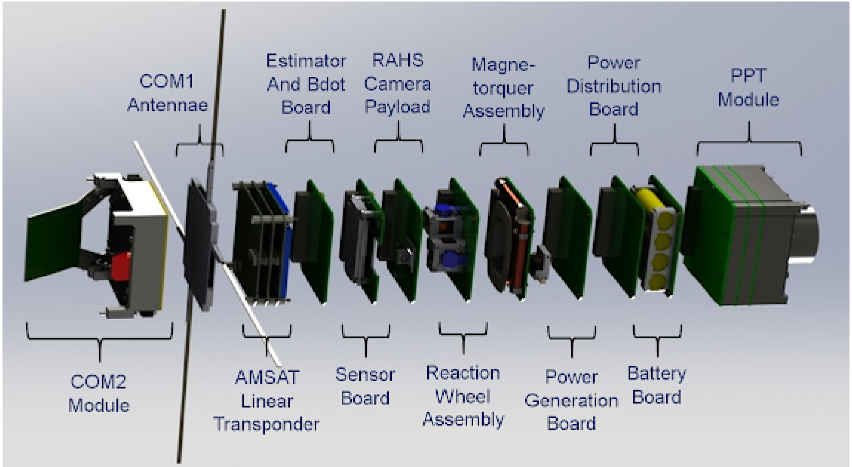
$$\Omega(\omega) = \begin{bmatrix} -[\omega\times] & \omega \\ -\omega^T & 0 \end{bmatrix}$$

The torque L comes from **thrusters**, **reaction wheels** or **magnetorquers**.

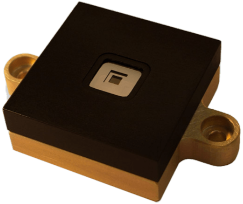
Spacecraft Actuators

Plasma Thruster

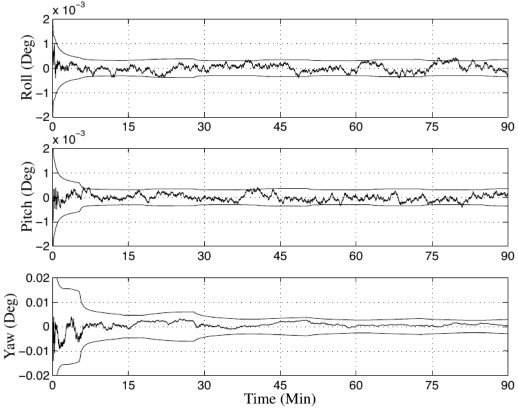


Measurement



Quaternion Extended Kalman Filter

Initialize	$\hat{\mathbf{q}}(t_0) = \hat{\mathbf{q}}_0, \quad \hat{\beta}(t_0) = \hat{\beta}_0$ $P(t_0) = P_0$
Gain	$K_k = P_k^- H_k^T(\hat{\mathbf{x}}_k^-)[H_k(\hat{\mathbf{x}}_k^-)P_k^- H_k^T(\hat{\mathbf{x}}_k^-) + R]^{-1}$ $H_k(\hat{\mathbf{x}}_k^-) = \left.\begin{bmatrix} [A(\hat{\mathbf{q}}^-)\mathbf{r}_1\times] & 0_{3\times3} \\ \vdots & \vdots \\ [A(\hat{\mathbf{q}}^-)\mathbf{r}_n\times] & 0_{3\times3} \end{bmatrix}\right\vert_{t_k}$
Update	$P_k^+ = [I - K_k H_k(\hat{\mathbf{x}}_k^-)]P_k^-$ $\Delta\hat{\tilde{\mathbf{x}}}_k^+ = K_k[\tilde{\mathbf{y}}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-)]$ $\Delta\hat{\tilde{\mathbf{x}}}_k^+ \equiv [\delta\hat{\alpha}_k^{+T} \ \Delta\hat{\beta}_k^{+T}]^T$ $\mathbf{h}_k(\hat{\mathbf{x}}_k^-) = \left.\begin{bmatrix} A(\hat{\mathbf{q}}^-)\mathbf{r}_1 \\ A(\hat{\mathbf{q}}^-)\mathbf{r}_2 \\ \vdots \\ A(\hat{\mathbf{q}}^-)\mathbf{r}_n \end{bmatrix}\right\vert_{t_k}$ $\hat{\mathbf{q}}_k^+ = \hat{\mathbf{q}}_k^- + \frac{1}{2}\Xi(\hat{\mathbf{q}}_k^-)\delta\hat{\alpha}_k^+$, re-normalize quaternion $\hat{\beta}_k^+ = \hat{\beta}_k^- + \Delta\hat{\beta}_k^+$
Propagation	$\hat{\omega}(t) = \tilde{\omega}(t) - \hat{\beta}(t)$ $\dot{\hat{\mathbf{q}}}(t) = \frac{1}{2}\Xi(\hat{\mathbf{q}}(t))\hat{\omega}(t)$ $\dot{P}(t) = F(t)P(t) + P(t)F^T(t) + G(t)Q(t)G^T(t)$ $F(t) = \begin{bmatrix} -[\hat{\omega}(t)\times] & -I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix}, \quad G(t) = \begin{bmatrix} -I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix}$



Matlab - simulating ODEs

```
clear all; close all; clc;

% simulation timespan
dt=0.1;
t_space = [0:dt:10];

% initial condition
x_0 = [10;10];

% sinusoidal input with zero-order-hold
u = cos(t_space);
x_array = zeros(2,length(t_space));

x_array(:,1) = x_0;

for ii = 1:length(t_space)-1

    t_int = [t_space(ii), t_space(ii+1)];

    %simulate ODE with input
    [t_out,y_out] = ode45(@fn_rhs,t_int,x_array(:,ii),[],u(ii));
    x_array(:,ii+1) = y_out(end,:)';

end

 plot(t_space, x_array(1,:)), hold on
 plot(t_space, x_array(2,:))
```

```
function [x_dot] = fn_rhs(t, x, u)
%fn_rhs: a `right-hand-side function' of the ode x_dot = f(t,x,u)

x_dot = exp(-t)*[cos(x(1)); sin(x(2))] + exp(-0.1*t)*[0;1]*u;

end
```