remarks on linear quadratic regulator

- computing *K* requires solving "algebraic riccati equation"
  - tricky to solve & requires numerical iteration
    ⇒ best to use software from experts

    $$K = -R^{-1}B^TP$$
    $$0 = PA + A^TP - PBR^{-1}B^TP + Q$$

    `ct.lqr(A,B,Q,R)` or MATLAB `lqr(A,B,Q,R)`
- LQR is a special case of general optimization problem: find ***u*** that minimizes given cost function and satisfies constraints (e.g. max throttle). This can be used to guide system to desired state/trajectory
- sketch for how LQR is solved:
  - use pontryagin's maximum principle (variational calculus)
  - in special case of linear, time-invariant system, quadratic cost, and infinite time horizon, result is that input is a linear function of state: ***u***=-*K**q***
- for control far away from equilibrium (e.g. aggressive maneuvers), need full nonlinear *trajectory tracking*
  - common "engineering" approach is receding horizon control (a.k.a. "model predictive control"): repeatedly calculate optimal ***u*** over a short horizon
  - biology-inspired: explore the solution space, reward if success (reinforcement learning). parameterize controller/"policy" with a neural network

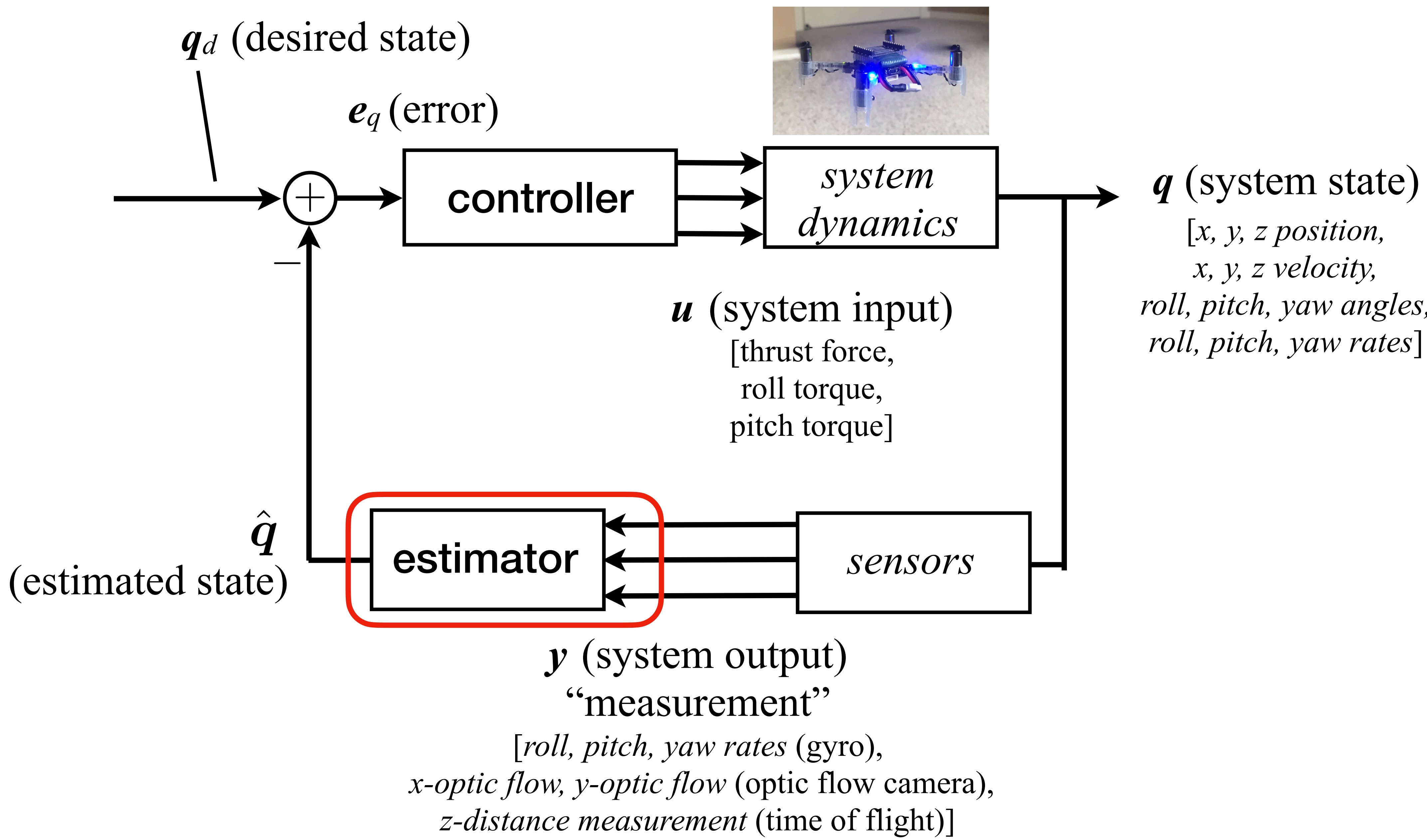
more remarks

- Full 3D flight control requires two separate, parallel 2D controllers. Also required are coordinate rotations between inner and outer loops.
  - See Fuller2019, "Four Wings: An Insect-Sized Aerial Robot With Steering ability and payload capacity for autonomy", *Robotics and Automation Letters* (2019) on the course web page for one approach.
- Simulating $\dot{\mathbf{R}} = \mathbf{R}\boldsymbol{\omega}'^{\times}$ in 3D leads to ill-formed R matrices because of numerical inaccuracy. Better to parameterize R with Euler Angles or quaternions
  - Euler Angles: see Mellinger2012: "Trajectory generation and control for precise aggressive maneuvers with quadrotors" *Int. J. Robot. Res.* on course we page
- To compensate for steady-state disturbances, e.g. torque bias that we must correct for, do *integral action* by adding a state z:

$$\frac{d}{dt}\begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} Ax + Bu \\ y - r \end{bmatrix} = \begin{bmatrix} Ax + Bu \\ Cx - r \end{bmatrix}$$ ← integral of (output) error

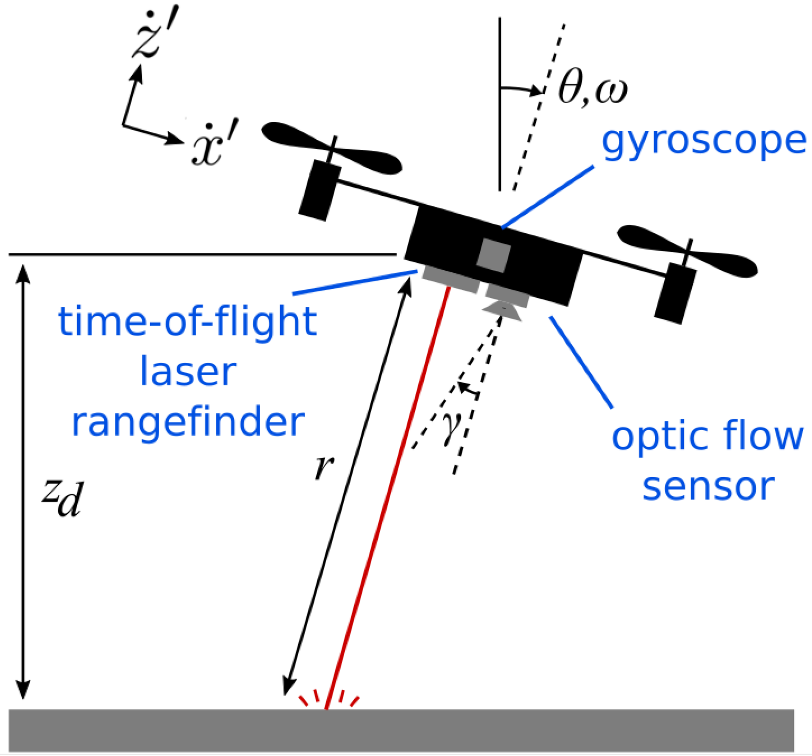
$$\boldsymbol{u} = -\boldsymbol{K}\boldsymbol{x} - \boldsymbol{K}_i\boldsymbol{z}$$

the control task

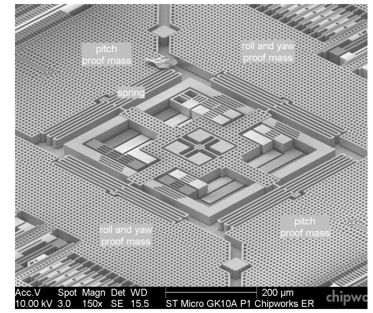


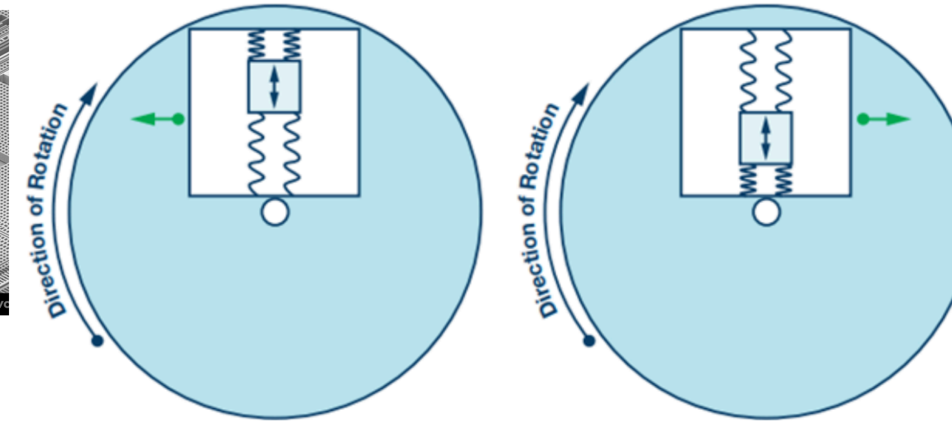
- estimator must reconstruct state vector from limited sensor information (number of sensors is typically < number of states)
- *separation principle* states that controller and estimator can be designed independently

sensors



Gyroscope: Bosch BMI088

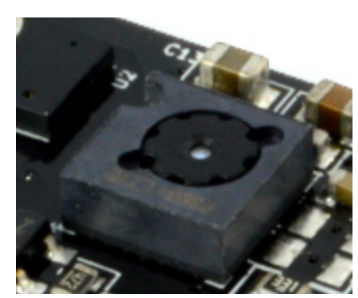




principle: sense coriolis forces using a vibrating proof mass

$$\omega_m = \omega + n_g$$

Optic flow sensor: Pixart PMW3901

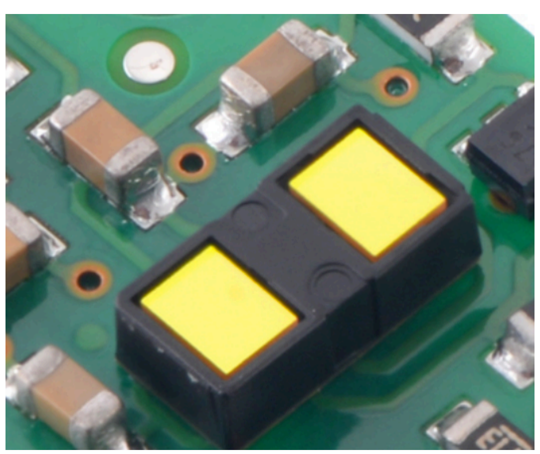


principle: measure speed of motion of visual scenery directly below to estimate lateral velocity

$$\Omega_m = \omega_y' - \frac{\dot{x}'}{r} + n_o$$

(will derive on board)

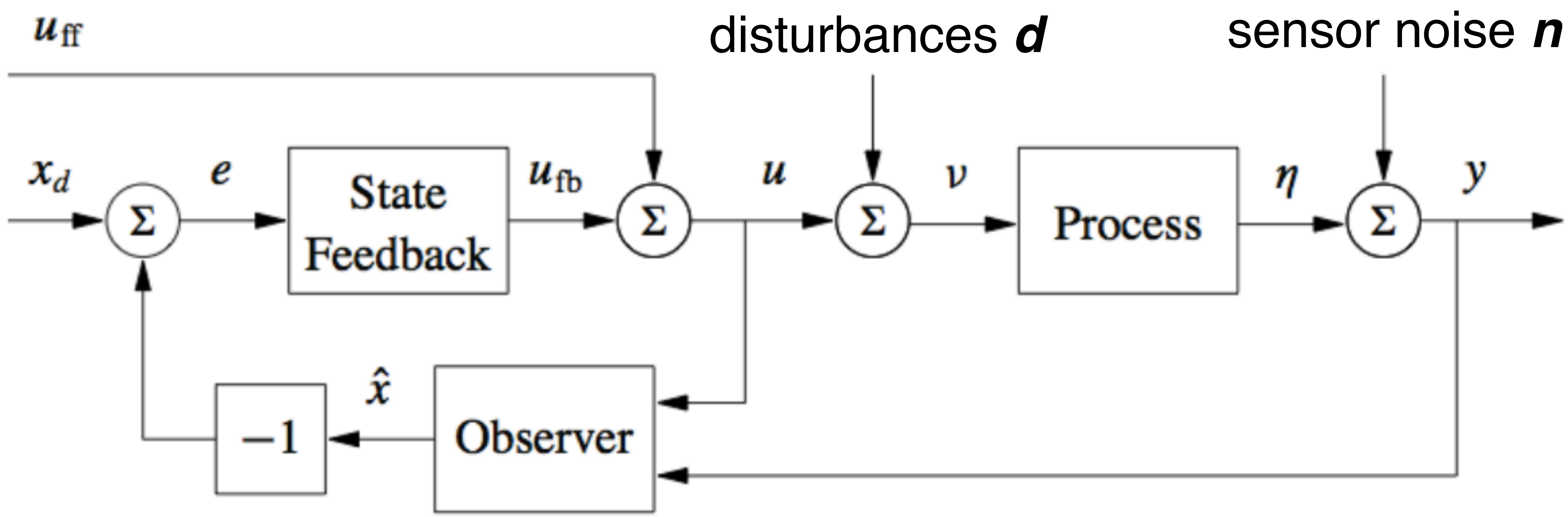
Time-of-flight laser rangefinder: ST VL53L1



principle: measure time taken for laser light to reflect

model for sensor: $r_m = r + n_t$ ← **noise**

State estimation for control



Problem Setup

- Given a dynamical system with noise and uncertainty, estimate the state

$$\begin{aligned} \dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\boldsymbol{u} + G\boldsymbol{d} \\ \boldsymbol{y} &= C\boldsymbol{x} + \boldsymbol{n} \end{aligned} \longrightarrow \begin{aligned} &\dot{\hat{x}} = \alpha(\hat{x}, y, u) \\ &\lim_{t\to\infty} E(x - \hat{x}) = 0 \end{aligned}$$

estimator

expected value

- $\hat{x}$ is called the *estimate* of x

Remarks

- Several sources of uncertainty: noise, disturbances, process, initial condition
- Uncertainties are unknown, except through their effect on measured output
- First question: when is this even *possible?*

Observability

Defn A dynamical system of the form

$$\dot{x} = f(x, u)$$
$$y = h(x, u)$$

is *observable* if for any $T > 0$ it is possible to determine the state of the system $x(T)$ through measurements of $y(t)$ and $u(t)$ on the interval $[0,T]$

Remarks

- Observability must respect *causality*: only get to look at past measurements
- We have ignored noise, disturbances for now ⇒ estimate exact state
- Intuitive way to check observability:

$$\dot{x} = Ax + Bu \qquad y = Cx$$

$$\begin{aligned} y &= \underline{Cx} \\ \dot{y} &= C\dot{x} = \underline{CAx} + CBu \\ \ddot{y} &= \underline{CA^2x} + CABu + CB\dot{u} \\ &\vdots \end{aligned}$$

$$W_o = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

Thm A linear system is observable if and only if the observability matrix W_o is full rank

State estimation: observer

Given that a system is observable, how do we actually estimate the state?

- Key insight: if current estimate is correct, follow the dynamics of the system

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

$$y = Cx$$

correction (based on output error)

prediction (copy of dynamics)

- Modify the dynamics to correct for error based on a linear feedback term
- L is the *observer gain matrix*; determines how to adjust the state due to error
- Look at the error dynamics for $\tilde{x} = x - \hat{x}$ to determine how to choose L:

$$\dot{\tilde{x}} = \dot{x} - \dot{\hat{x}} = Ax + Bu - (A\hat{x} + Bu + LC(x - \hat{x})) = (A - LC)\tilde{x}$$

Thm If the pair (A, C) is observable (associated W_o is full rank), then we can place the eigenvalues of $A\text{-}LC$ arbitrarily through appropriate choice of L.

How to choose gain *L*?

- "Kalman Filter" formulation: given system

$$\begin{aligned} \dot{\boldsymbol{q}} &= A\boldsymbol{q} + B\boldsymbol{u} + G\boldsymbol{d} \\ \boldsymbol{y} &= C\boldsymbol{q} + \boldsymbol{n} \end{aligned}$$

where ***d*** is process noise ("disturbance"), ***n*** is sensor noise.

d and ***n*** are zero-mean white Gaussian noise (eg for scalar *d*, $p(d) = \frac{1}{\sqrt{2\pi\sigma_d^2}} e^{-\frac{1}{2}\left(\frac{d}{\sigma_d}\right)^2}$)

and $E\{\boldsymbol{d}\boldsymbol{d}^T\} = Q_N = Q_N^T \geq 0 \quad E\{\boldsymbol{n}\boldsymbol{n}^T\} = R_N = R_N^T > 0$

- if noise is "stationary" (not changing with time) then the Kalman gain *L* minimizes expected squared error of the state estimate

$$\hat{\boldsymbol{q}} = A\hat{\boldsymbol{q}} + B\boldsymbol{u} + L(\boldsymbol{y} - C\hat{\boldsymbol{q}})$$

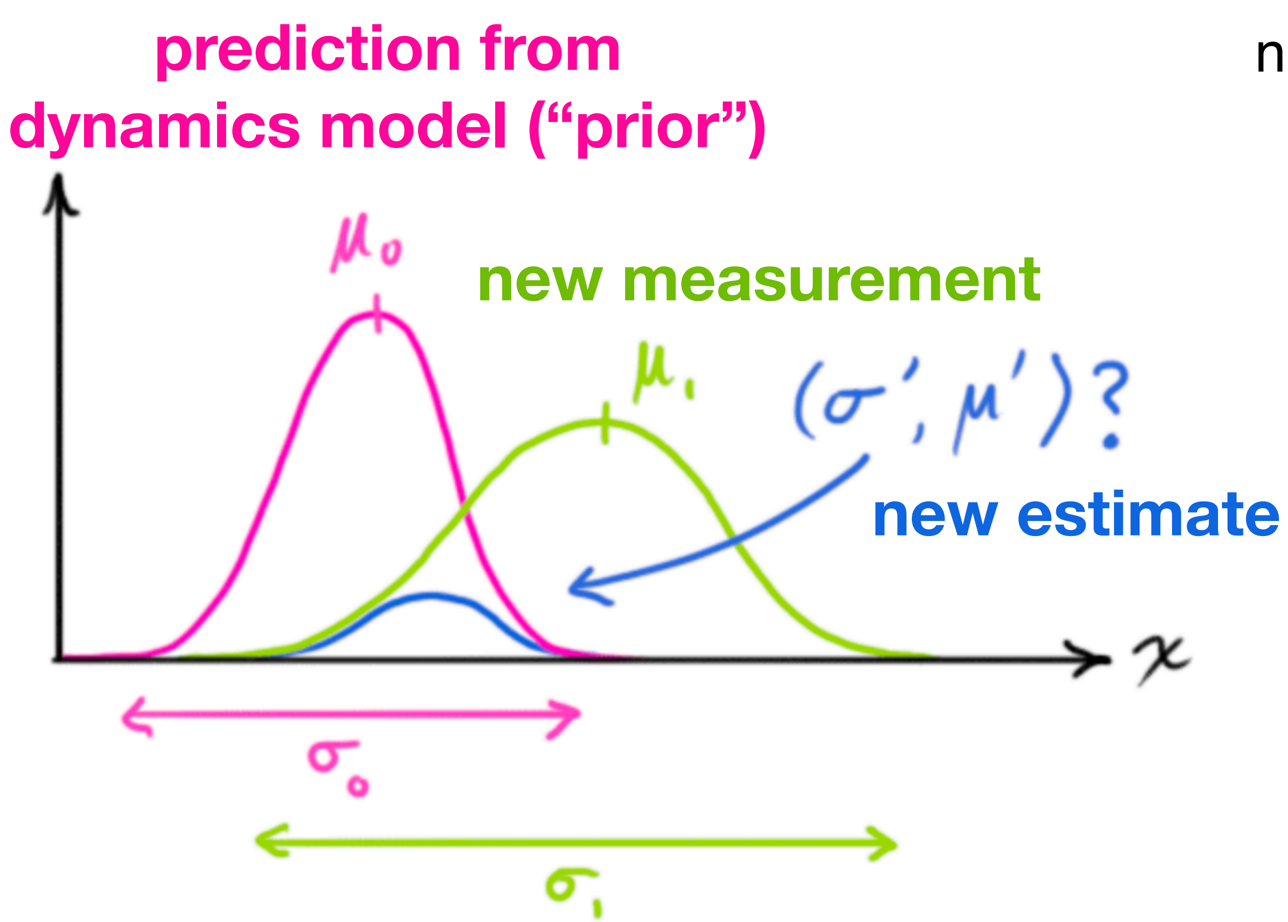
Remarks

- *L* is *also* the solution to an algebraic Riccati equation
  - use `ct.lqe(A,B,G,QN,RN)` or MATLAB `lqe(A,B,Q,R)`
- Can choose other *L*'s, but Kalman *L* minimizes error size

intuition

- Kalman Filter combines information from dynamics prediction with information sensor measurements using a “bayesian update”
  - multiply the probability density function (PDF) of the state estimate by the PDF of the new measurement

1D case



Bayesian inference:
new PDF = prior PDF * measurement PDF

$$\mu' = \mu_0 + \frac{\sigma_0^2(\mu_1 - \mu_0)}{\sigma_0^2 + \sigma_1^2}$$

$$\sigma'^2 = \sigma_0^2 - \frac{\sigma_0^4}{\sigma_0^2 + \sigma_1^2}$$

(KF does this for *n* dimensions)

remarks

- matrices Q_N and R_N are usually diagonal, meaning noise is not correlated
- sensor noise matrix R_N can come from datasheet or can be estimated:

$$R_N = \begin{bmatrix} \sigma_{n1}^2 & 0 & \cdots \\ 0 & \sigma_{n2}^2 & \\ \vdots & & \ddots \end{bmatrix} \quad \sigma_n = \sqrt{\frac{1}{N-1}\Sigma_i (y_i - y_{i,m})^2}$$

$y_{i,m}$ is sensor's measurement
y_i is ground truth measurement

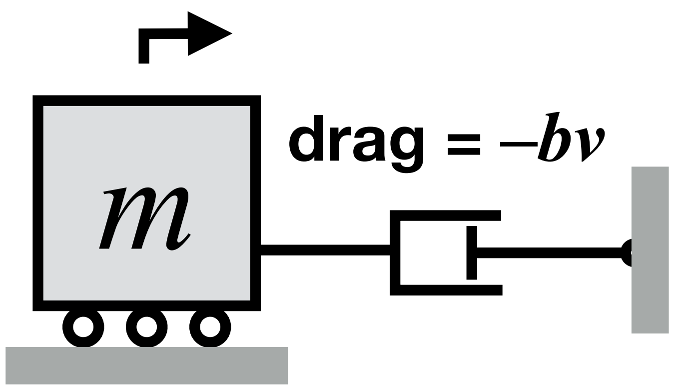
= `numpy.linalg.std(`$y_m - y$`)` if y's are arrays of data

- disturbance noise Q_N is harder to measure. Perspective: is tuning knob
  - large disturbance $Q_N \Rightarrow$ trust sensors more than prediction $\Rightarrow$ large L
  - small disturbance $Q_N \Rightarrow$ trust prediction more than sensors $\Rightarrow$ small L
- linear KF requires very little computation, just a few matrix multiply operations
  - rose to prominence on the Moon Lander in the 1960's (!)
- important variants:
  - sensors that do not update at equal intervals: use "information form" that separates prediction from correction step, using different L for each sensor
  - for nonlinear system, use extended KF ("EKF") (see Murray, Optimization-Based Control) or unscented KF ("UKF") (more computation needed)
    - crazyflie uses an extended KF to enable more aggressive maneuvers (Greif2017 on course website)

Example: me586_example_kalman_estimator.ipynb

A) Kalman Filter to estimate velocity from this dynamical system:

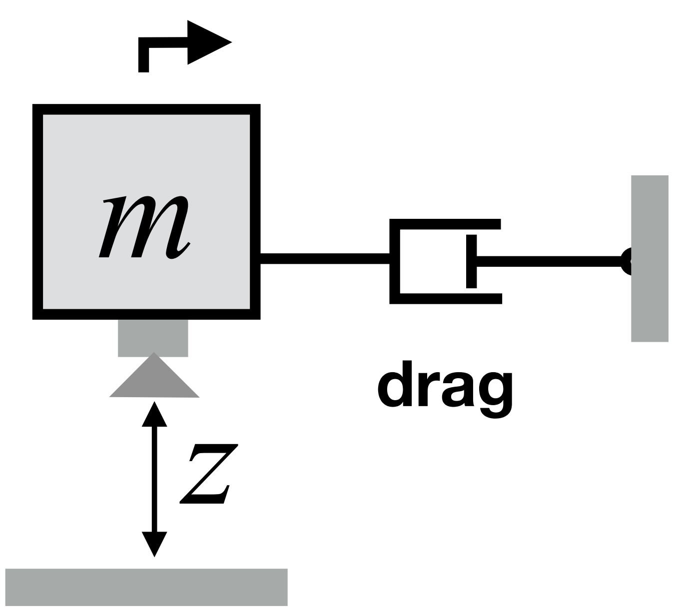
Velocity measurement is $v_m = v + n$ (true value + noise)



B) Vary tuning knob Q_N (magnitude of disturbance noise)

C) helicopter-based optic flow (must linearize at desired height $z=z_d$)

$$v_m = -\Omega_m z + n$$



compared to a low-pass filter, the Kalman Filter:

- can estimate "hidden" but observable states
- can perform sensor fusion between different sensors at different update rates
- can accommodate effect of known inputs
- reduces estimate lag time, if the quantity you are interested in behaves as a dynamical system
- minimizes expected squared estimate error
- but needs a model of dynamics

well-suited to a dynamical system such as an aircraft with a good model (eg rigid body equations) and states that are not directly measured by sensors (e.g. orientation)

The separation principle

Feedback the estimated state: $u = -K\hat{x} + k_r r$ **driving the output *y* to the value *r* is another way to do trajectory tracking**

- **Analysis:** Again, let $\tilde{x} = x - \hat{x}$ denote the error in the state estimate. The dynamics of the controlled system under this feedback are:

$$\dot{x} = Ax + Bu = Ax - BK\hat{x} - Bk_r r = Ax - BK(x - \tilde{x}) + Bk_r r$$
$$= (A - BK)x + BK\tilde{x} + Bk_r r$$

  - Introduce a new *augmented* state: $q = [x \quad \tilde{x}]^T$. The dynamics of the system defined by this state is:

$$\begin{bmatrix} \dot{x} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} (A - BK) & BK \\ 0 & (A - LC) \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + \begin{bmatrix} Bk_r \\ 0 \end{bmatrix} r \equiv Mq + B_M r$$

  The characteristic polynomial of M is:

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

- If the system is *observable* and *reachable*, then the poles of $(A - BK)$ and $(A - LC)$ can be set *arbitrarily* and *independently*
- If *K* is an LQR controller and *L* is a Kalman Filter, then is a “Linear Quadratic Gaussian” (LQG) controller