the control task

q_d (desired state)

e_q (error)

controller

system dynamics

q (system state)

[*x, y, z position,
x, y, z velocity,
roll, pitch, yaw angles,
roll, pitch, yaw rates*]

$-$

u (system input)
[thrust force,
roll torque,
pitch torque]

$\hat{q}$
(estimated state)

estimator

sensors

y (system output)
"measurement"
[*roll, pitch, yaw rates* (gyro),
x-optic flow, y-optic flow (optic flow camera),
z-distance measurement (time of flight)]
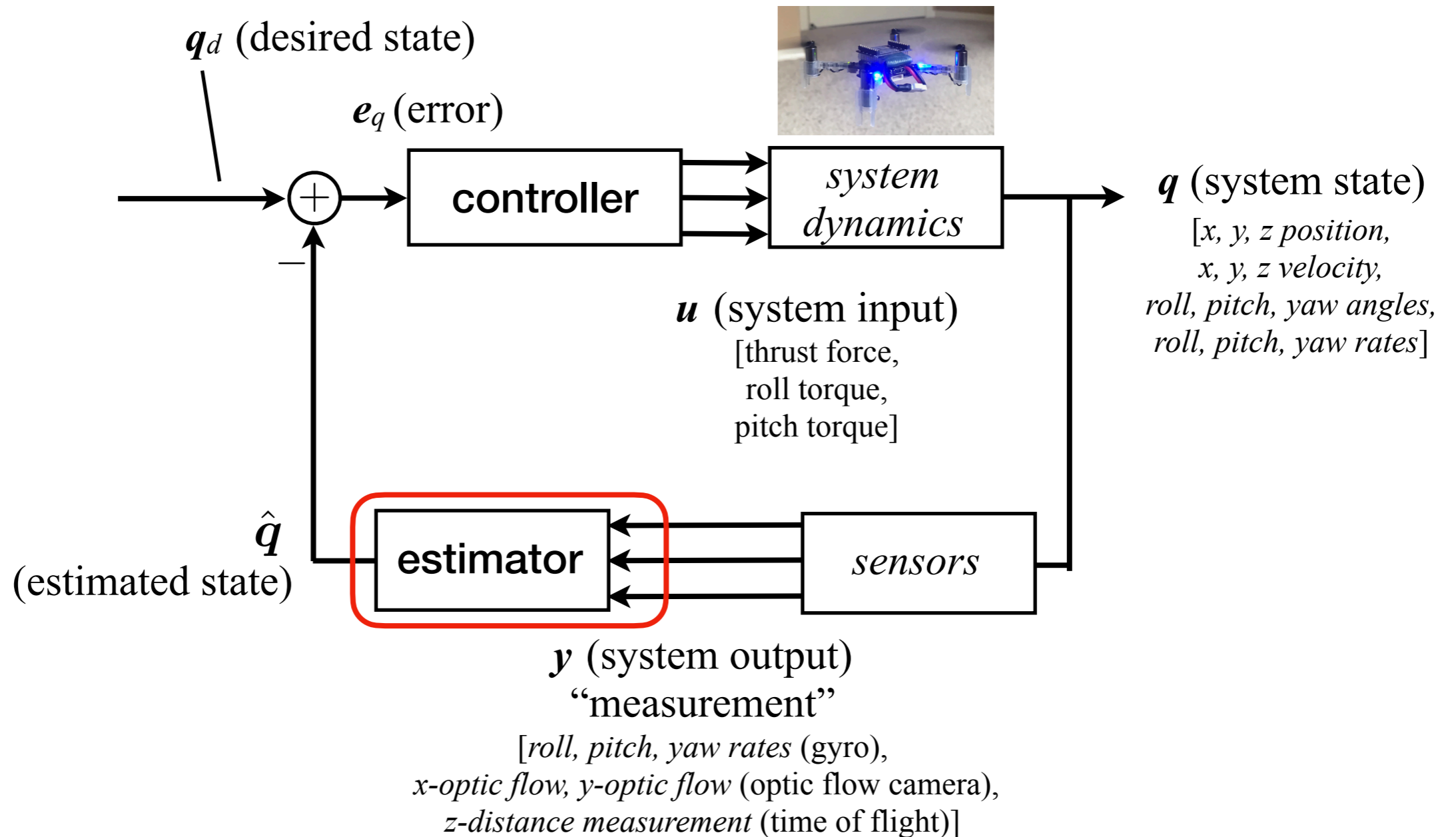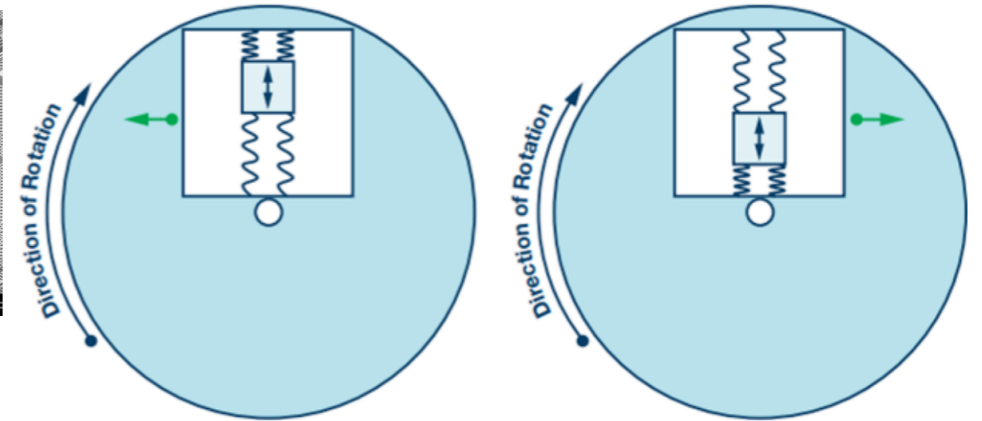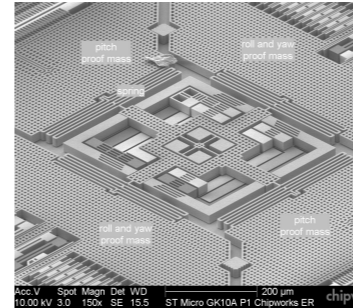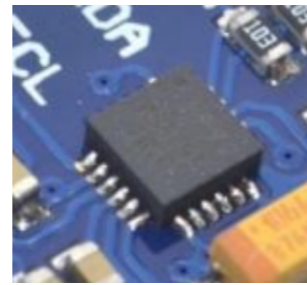
- estimator must reconstruct state vector from limited sensor information (number of sensors is typically < number of states)

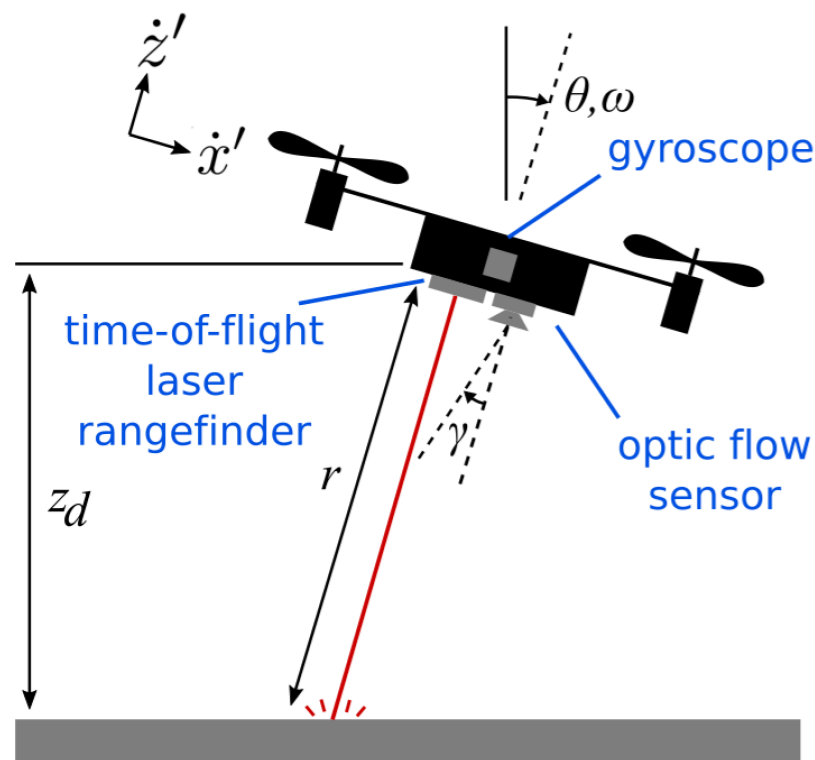- *separation principle* states that controller and estimator can be designed independently
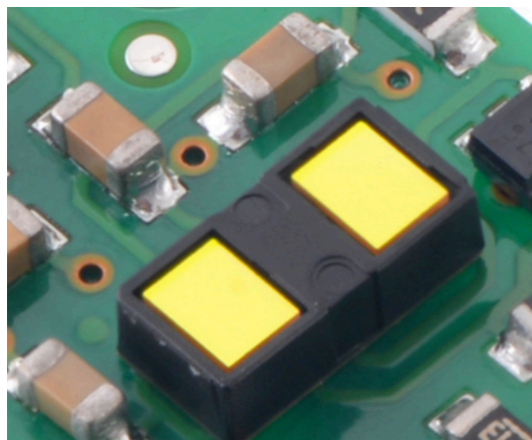
sensors

Gyroscope: Bosch BMI088

principle: sense coriolis forces using a vibrating proof mass

$$\omega_m = \omega + n_g$$

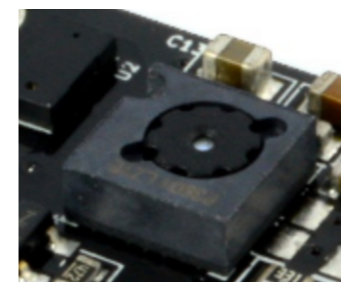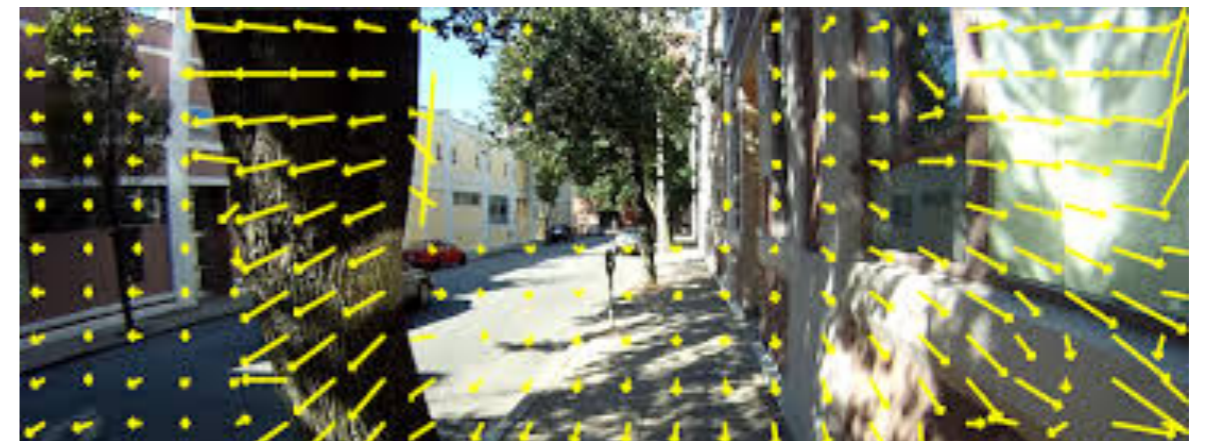## Time-of-flight laser rangefinder: ST VL53L1

principle: measure time taken for laser light to reflect

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

Optic flow sensor: Pixart PMW3901
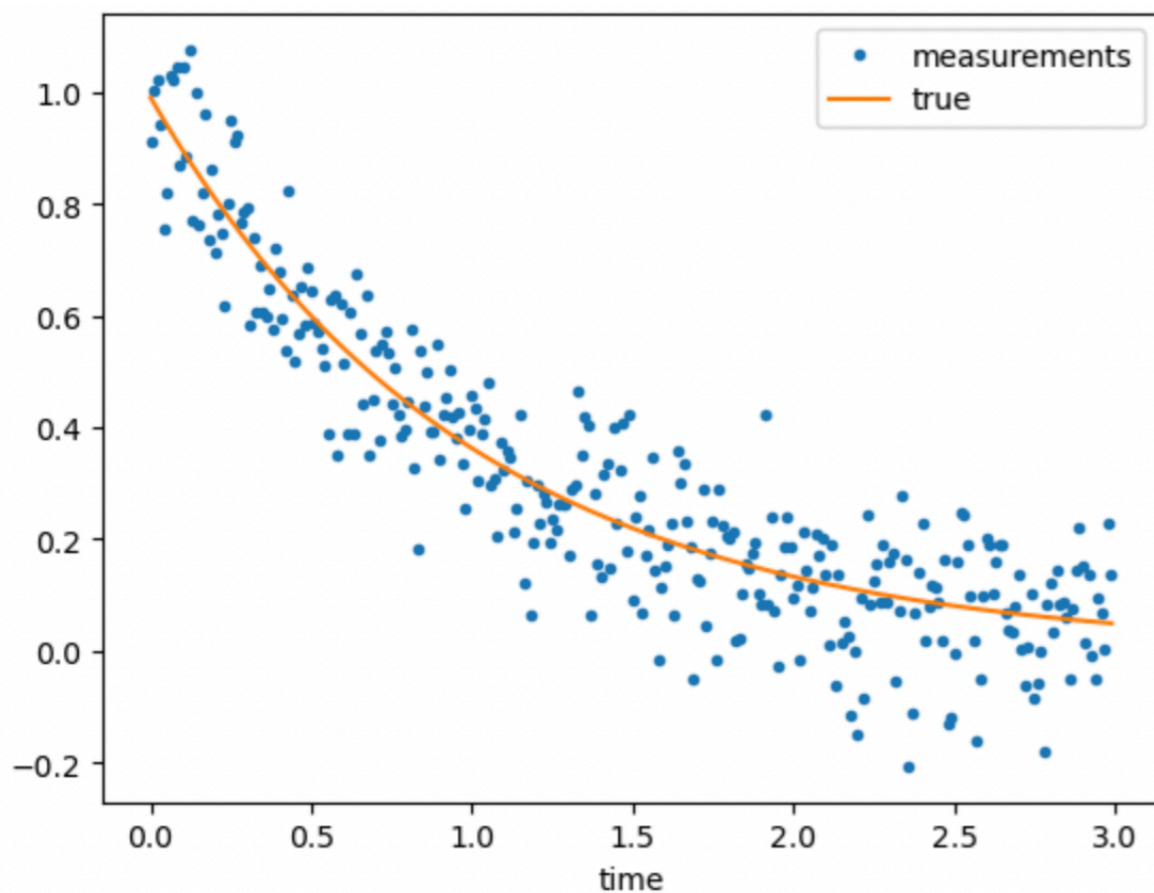

principle: measure speed of motion of visual scenery directly below to estimate lateral velocity

$$\Omega_m = \omega'_y - \frac{\dot{x}'}{r} + n_o$$
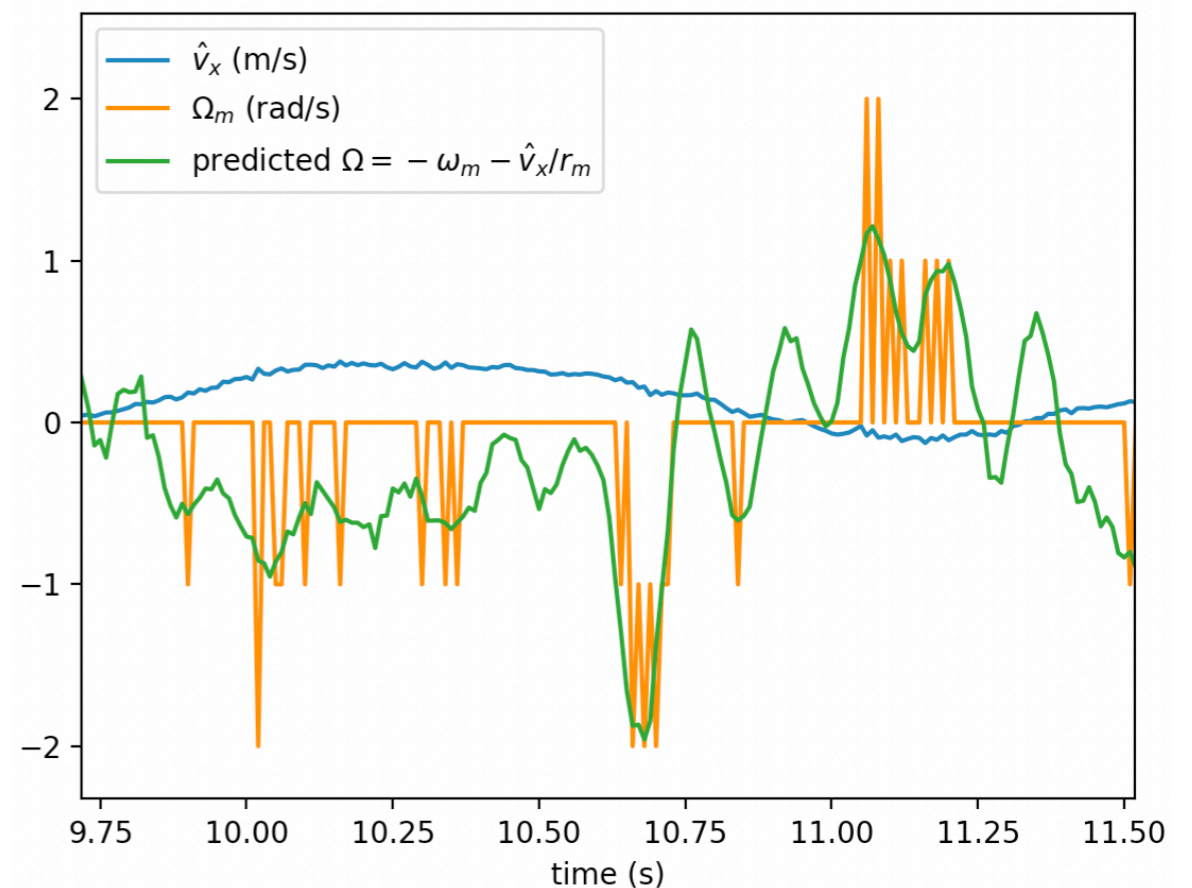
(will derive on board)

Measurement noise

idealization: Gaussian noise added to true signal
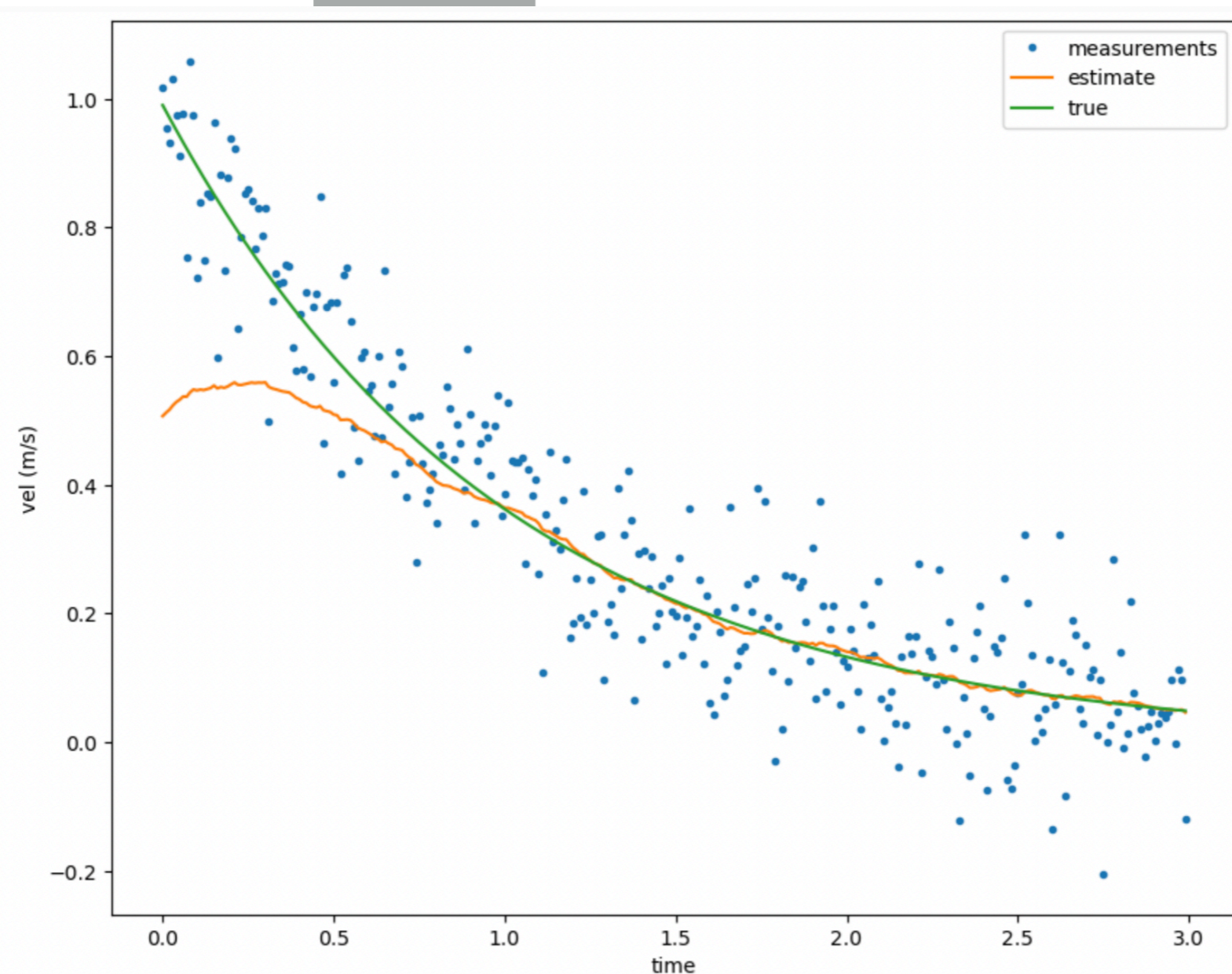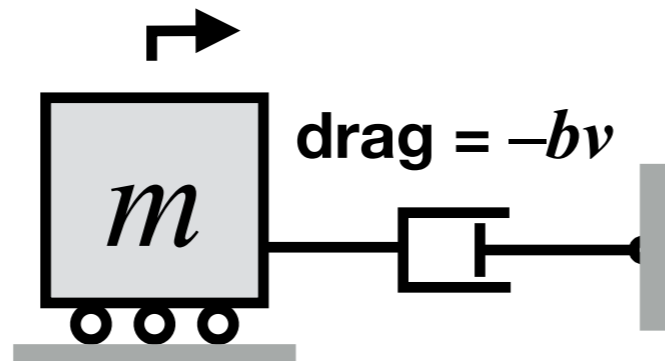
real signals may look different!

Crazyflie measured optic flow (and predicted based on Kaman filter) during a forward maneuver

3

Example: estimate velocity of a dynamical system

(In me586_example_kalman_estimator.ipynb)

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

State estimation for control
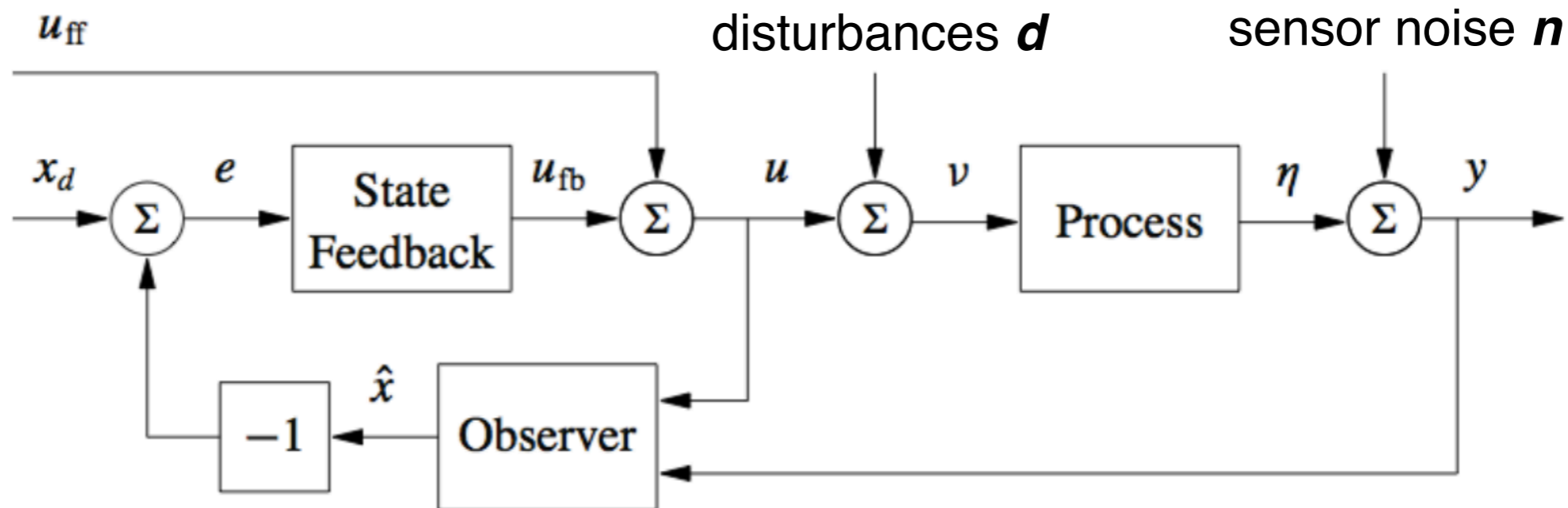
Problem Setup

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

$$
\begin{aligned}
\dot{x} &= A x + B u + G d \\
y &= C x + n
\end{aligned}
\qquad\longrightarrow\qquad
\begin{aligned}
&\dot{\widehat{x}} = \alpha(\widehat{x}, y, u) \quad \longleftarrow \text{ estimator} \\
&\lim_{t\to\infty} E(x - \widehat{x}) = 0 \\
&\phantom{\lim_{t\to\infty}} \longleftarrow \text{expected value}
\end{aligned}
$$

- $\widehat{x}$ 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
(General, nonlinear case)

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

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 $\Rightarrow$ estimate exact state
- Intuitive way to check observability:

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

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

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

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

$$[y, \dot{y}, \ddot{y}, \ldots]^T = W_o x \Rightarrow x = (W_o^T W_o)^{-1} W_o^T [y, \dot{y}, \ldots]^T$$

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

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

$$\dot{\hat{x}} = \underbrace{A\hat{x} + Bu}_{\text{prediction (copy of dynamics)}} + L(y - C\hat{x}) \longleftarrow \text{correction (based on output error)}$$

- 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-LC arbitrarily through appropriate choice of L.

How to choose gain *L*?

- "Kalman Filter" formulation: given system

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

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

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

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

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

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

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

- 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

prediction from
dynamics model ("prior")

μ_0

new measurement

μ_1

$(\sigma', \mu')?$

new estimate

σ_0

σ_1

x
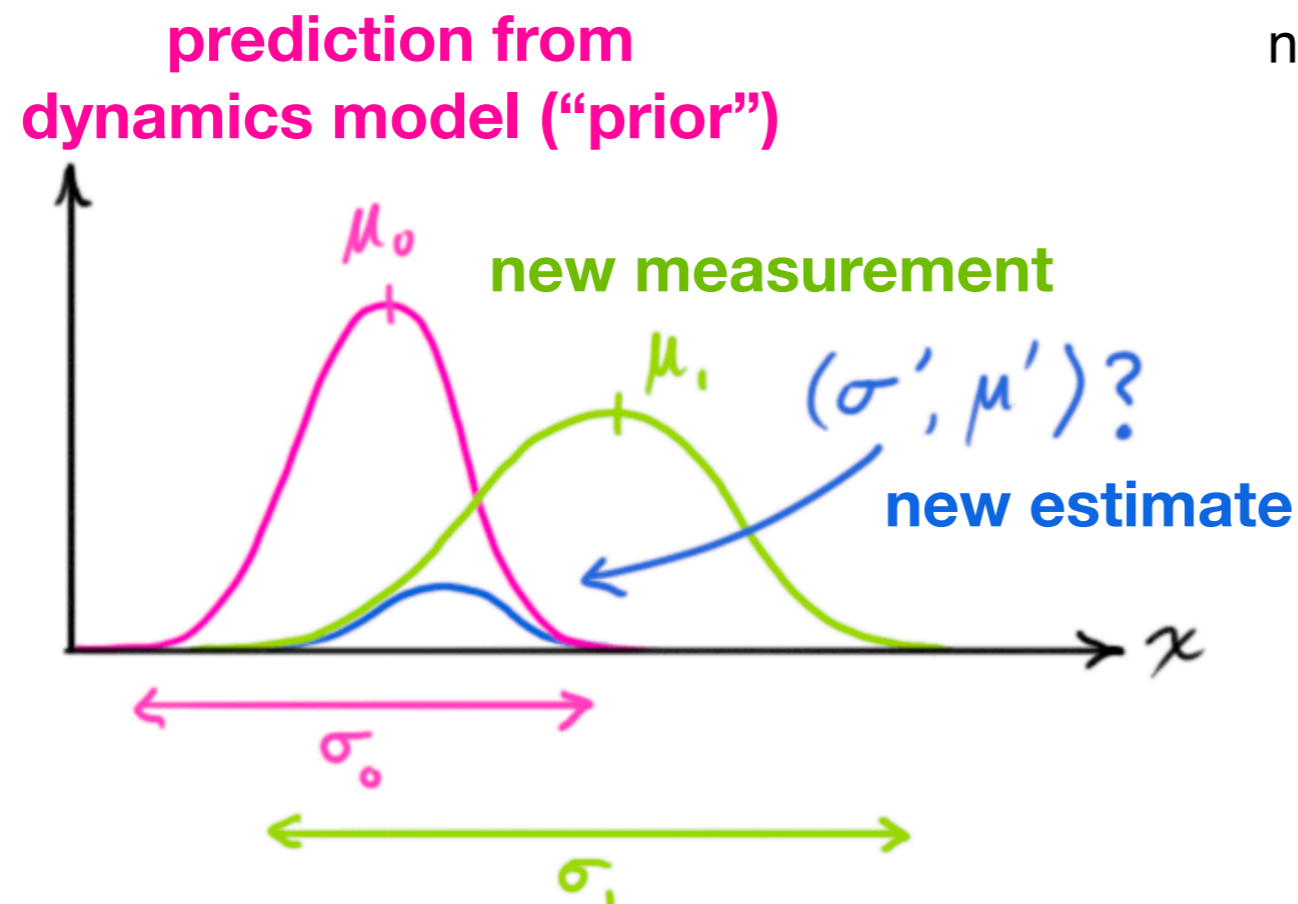
Bayesian inference:
new PDF = prior PDF * measurement PDF

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

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

(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:

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

$y_{i,m}$ 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

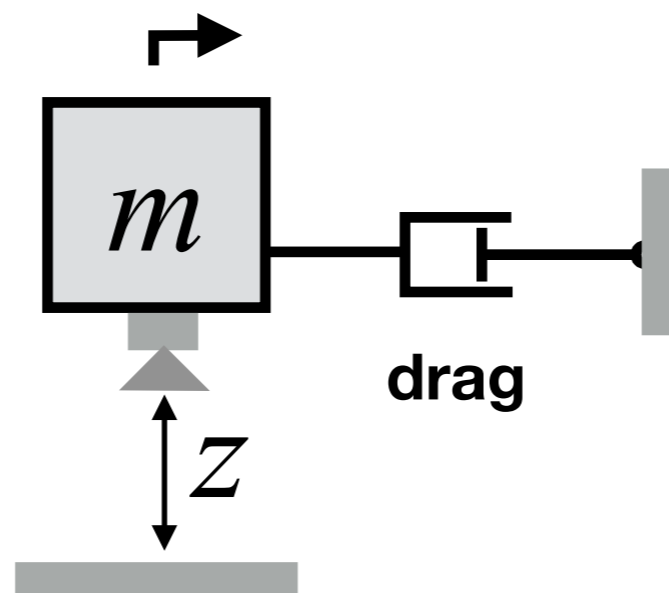A) Kalman Filter to estimate velocity from this dynamical system:

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

drag $= -bv$
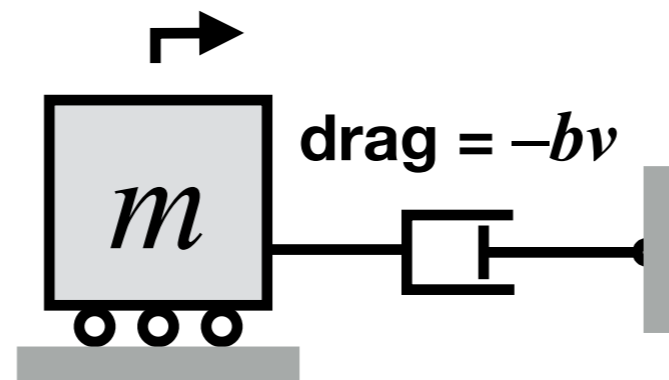
B) Vary tuning knob Q_N (magnitude of disturbance noise)

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

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

- **not directly measuring v**
- **Effect of not being at linearized altitude**

drag

z

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

- can estimate "hidden" but observable states, not just directly-measured 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: $u = -K\hat{x} + k_r r$

- **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