6.3100 April 27, 2026 Lecture: State Estimation with Luenberger Observers

April 28, 2026

Last Lecture

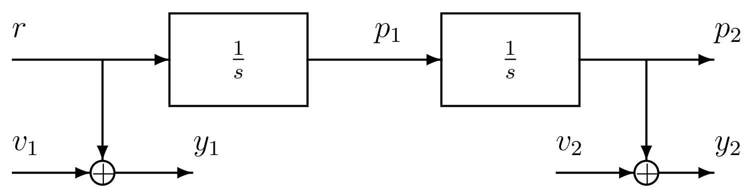
- Example: estimate the middle state of a double integrator chain

Outline for Today

- Example modified to match the maglev model
- State estimation with Luenberger observers

Example: Recovering the Middle Signal in a Chain of Integrators

Last lecture we considered the (rather common) scenario when both the input r and the output p_2 of a chain of two pure integrators are measured (with noises v_1 and v_2, respectively), while the output p_1 of the first integrator is not, as described by the block diagram

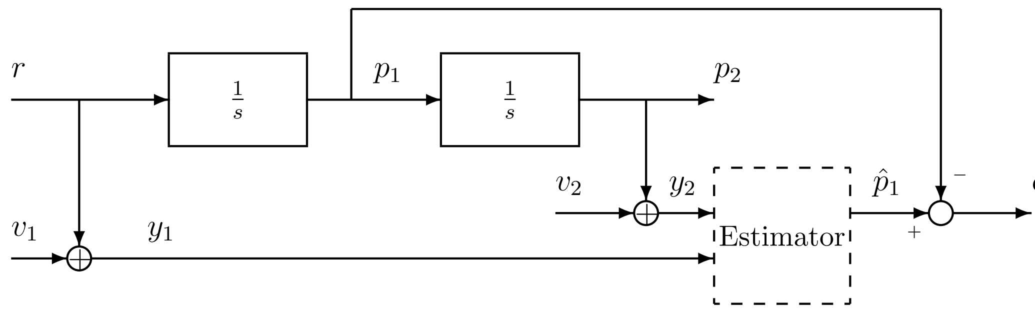


or, equivalently, equations

$$
\begin{aligned}
\dot{p}_1(t) &= r(t), \\
\dot{p}_2(t) &= p_1(t), \\
y_1(t) &= r(t) + v_1(t), \\
y_2(t) &= p_2(t) + v_2(t).
\end{aligned}
$$

We were designing a linear estimator with constant coefficients, which took y_1 and y_2 as inputs, and produced an estimate $\hat{p}_1$ of p_1, with the objective of making the estimation error $e_1 = \hat{p}_1 - p_1$ not depending on r, and minimizing its sensitivity of e_1 to noises v_1 and v_2, expressed by the "performance" functional

$$J = \max_\omega |H_{v_1 \to e_1}(j\omega)| + \max_\omega |H_{v_2 \to e_1}(j\omega)|.$$



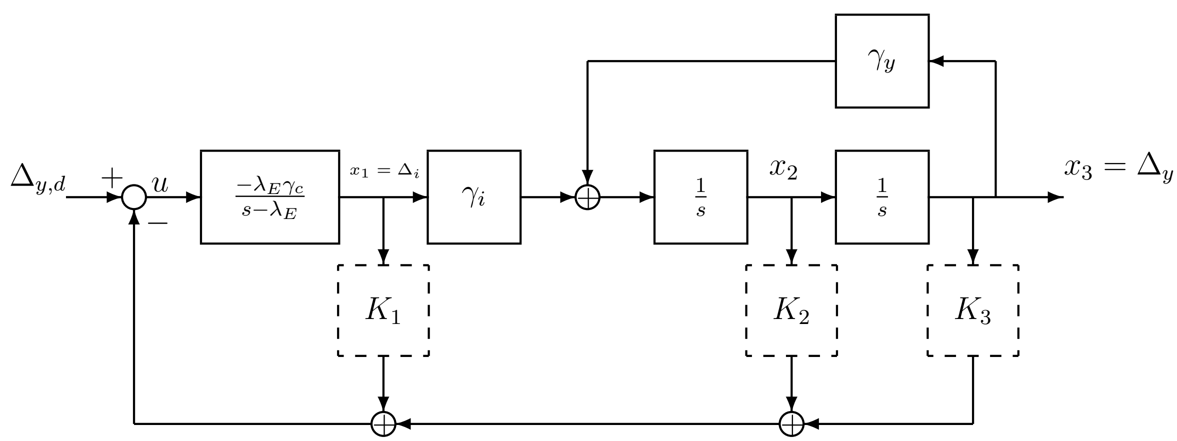
We came up with an estimator of the form

$$\hat{p}_1(t) = \hat{p}_*(t) + L y_2(t), \qquad \text{where} \quad \frac{d\hat{p}_*(t)}{dt} = -L\hat{p}_*(t) + y_1(t) - L^2 y_2(t),$$

where L is a positive constant. For this estimator, it turned out that $J = L + 1/L$, hence $L = 1$ was the optimal value of L.

A Better Noise Model

In applying the estimator from the example to our MAGLEV model



where, as shortcuts, γ_i, γ_y, and γ_c stand for $\gamma_{da/di}$, $\gamma_{da/dy}$, and $\gamma_{di/dc}$, respectively, we use it to estimate the unmeasured state x_2. To fit the double integrator model to the MAGLEV realities, we set

$$r(t) = \gamma_i x_1(t) + \gamma_y x_3(t), \quad p_1(t) = x_2(t), \quad p_2(t) = x_3(t),$$

and note that the actually measured variables are $x_1 = \Delta_i$ and $x_3 = \Delta_y$. We make an assumption that Δ_y is measured with noise $h_2 v_2$, and Δ_i is measured with noise $h_1 v_1$, where the constant h_1 and h_2 reflect the observed intensities of of the noises associated with measuring Δ_i and Δ_y, respectively. Accordingly, the measurement model has to be modified to

$$\begin{aligned} y_1(t) &= r(t) + \gamma_i h_1 v_1(t) + \gamma_y h_2 v_2(t), \\ y_2(t) &= p_2(t) + h_2 v_2(t). \end{aligned}$$

In turn, this leads to different expressions for $H_{v_1 \to e_1}(s)$ and $H_{v_2 \to e_1}(s)$:

$$H_{v_1 \to e_1}(s) = \frac{h_1 \gamma_i}{s + L}, \qquad H_{v_2 \to e_1} = h_2 \frac{Ls + \gamma_y}{s + L},$$

a modified performance functional:

$$J = \frac{h_1 \gamma_i}{L} + h_2 \max\{L,\ \gamma_y / L\},$$

and ultimately to the new optimal L:

$$L_{opt} = \max\left\{ \sqrt{\gamma_y},\ \sqrt{\gamma_i h_1 / h_2}\ \right\}.$$

Sensitivity to Modeling Errors

Assuming our knowledge of the MAGLEV coefficients γ_i, γ_y is perfect, the transfer function from $r = \gamma_i \Delta_i + \gamma_y \Delta_y$ to the resulting estimation error $e_1 = \hat{p}_1 - p_1$ remains zero. In practice, the estimator will use some *approximate* values $\hat{\gamma}_i \approx \gamma_i$, $\hat{\gamma}_y \approx \gamma_y$, in which case the estimator equations become

$$\begin{aligned} \hat{p}_1(t) &= \hat{p}_*(t) + L(\Delta_y(t) + h_2 v_2(t)), \\ \frac{d\hat{p}_*(t)}{dt} &= -L\hat{p}_*(t) + \hat{\gamma}_i(\Delta_i(t) + h_1 v_1(t)) + (\hat{\gamma}_y - L^2)(\Delta_y(t) + h_2 v_2(t)). \end{aligned}$$

Hence the estimation error $e_1(t) = \hat{p}_1(t) - \dot{\Delta}_y(t)$ will be given by

$$\begin{aligned} e_1(t) &= e_*(t) + L h_2 v_2(t), \\ \dot{e}_*(t) &= -L e_*(t) + h_1 \hat{\gamma}_i v_1(t) + (\hat{\gamma}_y - L^2) h_2 v_2(t) + (\hat{\gamma}_y - \gamma_y)\Delta_y(t) + (\hat{\gamma}_i - \gamma_i)\Delta_i(t). \end{aligned}$$

Accordingly, when figuring out a good value of $L > 0$, one has to take into account not just how strong the frequency responses to noises are, via

$$\max_\omega |H_{v_1 \to e_1}(j\omega)| = \frac{h_1 \hat{\gamma}_i}{L}, \qquad \max_\omega |H_{v_2 \to e_1}(j\omega)| = h_2 \max\{L, \hat{\gamma}_y / L\}$$

but also the impact of

$$\max_{\omega}\left|\frac{\hat{\gamma}_y-\gamma_y}{j\omega+L}\right|=\frac{|\hat{\gamma}_y-\gamma_y|}{L},\qquad \max_{\omega}\left|\frac{\hat{\gamma}_i-\gamma_i}{j\omega+L}\right|=\frac{|\hat{\gamma}_i-\gamma_i|}{L},$$

which quantify the sensitivity of estimation error to modeling error. In practice measurement intensity is relatively small so the modeling error is likely a deciding factor.

One extreme case of estimator design is to use $\hat{\gamma}_y=\hat{\gamma}_i=0$, which would be reasonable when there is no confidence in knowing the model coefficients. The resulting estimator ignores the y_1 (current) measurement completely:

$$\hat{p}_1(t)=\hat{p}_*(t)+Ly_2(t),\qquad \frac{d\hat{p}_*(t)}{dt}=-L\hat{p}_*(t)-L^2y_2(t),$$

which makes for the estimator transfer function

$$H_{y_2\to\hat{p}_1}(s)=L-\frac{L^2}{s+L}=\frac{Ls}{s+L}.$$

Since $H_{y_2\to\hat{p}_1}(s)\approx s$ for $|s|\ll L$, one can argue that transfer function $H_{y_2\to\hat{p}_1}(s)$ characterizes a system which acts as a differentiator for low-frequency signals.

The following Arduino plotter screenshots show the relative performance of four ways of estimating $\dot{\Delta}_y(t)$:

(a) Numerical differentiation, via something like

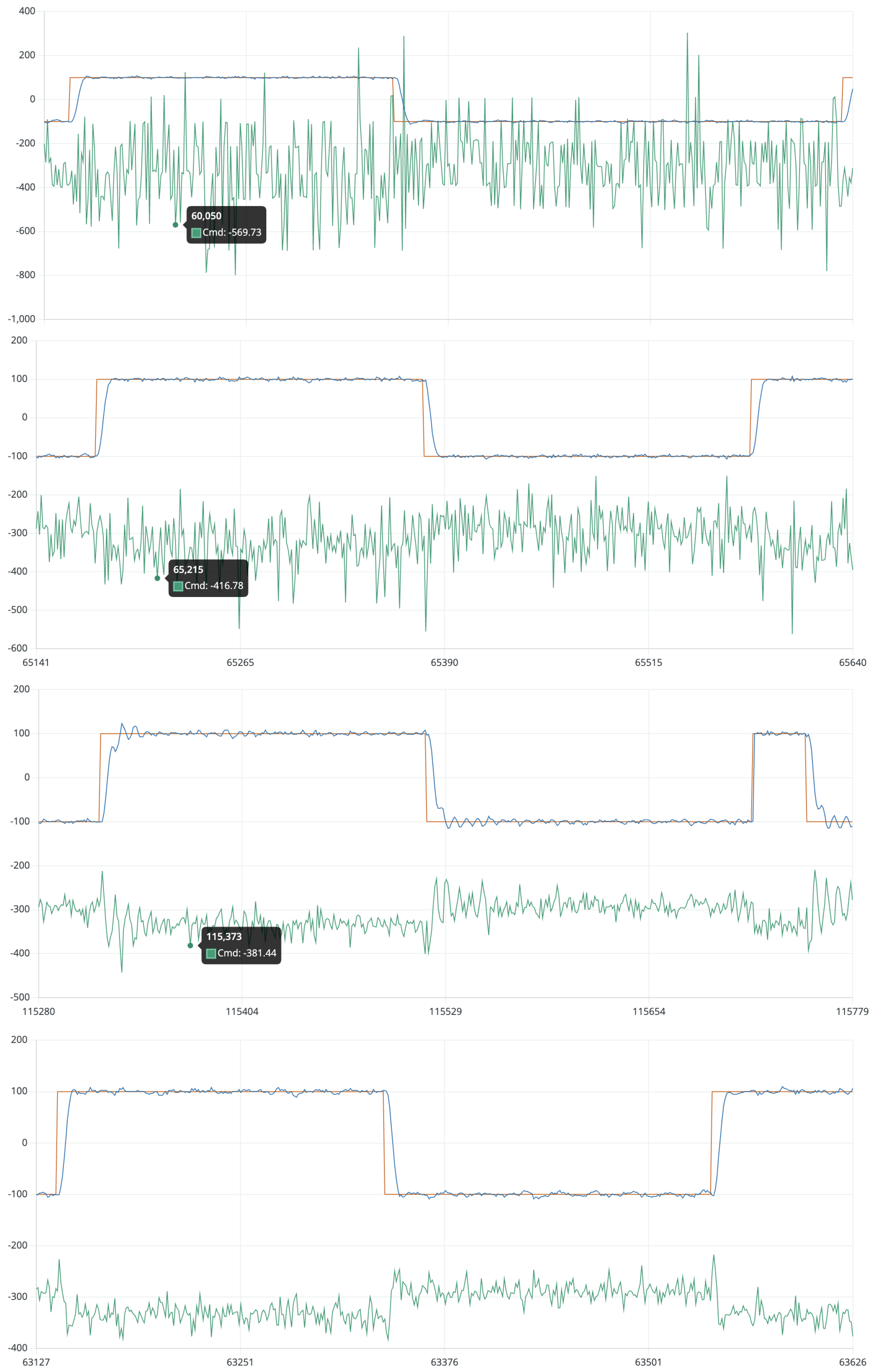
```
float delDeltaY = (deltaY - pastDeltaY[Mback]);
vel = my6310.scaleAndBound(delDeltaY, BV, mDeltaT);
```

in the Arduino code, with the default value of `Mback=4`.

(b) Limited bandwidth differentiation of $\Delta_y(t)$, i.e., using $\hat{\gamma}_y=\hat{\gamma}_i=0$ in the estimator, with $L=500$.

(c) Limited bandwidth differentiation of $\Delta_y(t)$, i.e., using $\hat{\gamma}_y=\hat{\gamma}_i=0$ in the estimator, with $L=100$ (reduced in an attempt to reduce noise sensitivity).

(d) The estimator with $\hat{\gamma}_y,\hat{\gamma}_i$ chosen as the best guesses at γ_y,γ_i, and $L=100$.



One can see that using a too-small value in (c) resulted in ringing response (though, on the positive side, it reduced noise, too). Estimator (d) performs better, as expected since it tries to use both measured signals, not just one of the two.

Linear State Estimation

In general, linear state estimation follows the basic outline of the example above:

- use a linear model of state dynamics which pays attention to noise insertion;
- design and optimize the estimator as a linear dynamical system;
- define performance in terms of the noise-to-estimation error transfer functions;

there are significant differences as well.

In general, the so-called linear-quadratic optimal state estimator is designed for a state space model of the form

$$\dot{x}(t) = Ax(t) + B_1 w(t) + B_2 u(t), \qquad y(t) = C_2 x(t) + D_{21} w(t) + D_{22} u(t),$$

where

- $x(t) \in \mathbb{R}^n$ is the state vector to estimate;
- $u(t) \in \mathbb{R}^m$ is the control command, assumed to be known at time t for every t;
- $y(t) \in \mathbb{R}^k$ is the sensor measurement (multi-dimensional when there are multiple sensors), assumed to be known at time t for every t;
- $w(t) \in \mathbb{R}^d$ is the noise vector, not known, but assumed to have independent components of equal intensity; mathematically, $w(\cdot)$ is assumed to be the so-called "normalized white noise", which is part reasonable (in particular, to assume that $w(t_1)$ and $w(t_2)$ are independent for $t_1 \neq t_2$), but part absolutely bonkers crazy (for example, to assume that, for every fixed t, the standard deviation of $w(t)$ is infinite);
- A, B_1, B_2, C_2, D_{21}, D_{22} are known constant real matrices of appropriate dimensions.

Luenberger Observer

The estimator itself is designed in the *Luenberger observer* (sometime referred to as simply *state observer*) form

$$\frac{d\hat{x}(t)}{dt} = A\hat{x}(t) + B_2 u(t) + L(y(t) - C_2\hat{x}(t)).$$

The idea of such observer extends the trick played in the earlier example: when all eigenvalues of A have negative real part, one can produce a stable estimate $\hat{x}(t)$ of $x(t)$ by simply running a copy of the original state space model, with the (unknown) $w(t)$ dropped out:

$$\frac{d\hat{x}(t)}{dt} = A\hat{x}(t) + B_2 u(t).$$

Since the estimation error $e(t) = \hat{x}(t) - x(t)$ will satisfy the differential equation

$$\dot{e}(t) = Ae(t) - B_1 w(t),$$

the estimation error will behave in a stable fashion. This, however, will not work when A has eigenvalues with non-negative real part (this will also work poorly when A has eigenvalues with real part that is negative, but still close to zero).

In order to be able to apply the idea, we can try to re-write the original state space equation in the equivalent form

$$\dot{x}(t) = Ax(t) + B_1 w(t) + B_2 u(t) + L(y(t) - C_2 x(t) - D_{21} w(t) - D_{22} u(t)),$$

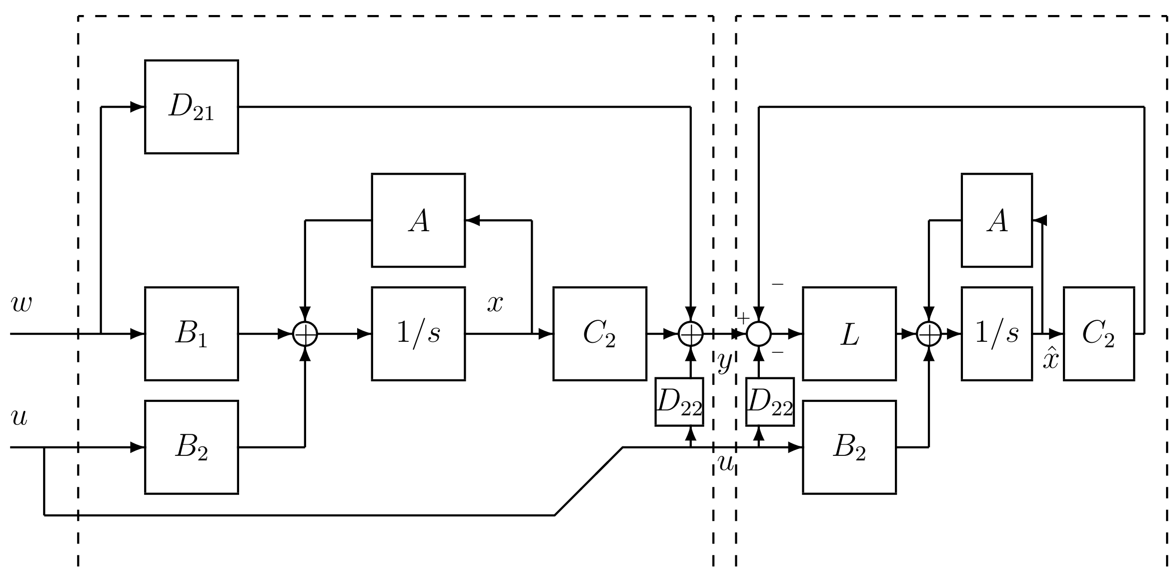
where L is an arbitrary matrix of appropriate dimensions (the equivalence follows from the fact that $y(t) - C_2 x(t) - D_{21} w(t) - D_{22} u$ equals to zero at all times). Now we can use the *modified* equation to convert to a state observer, by dropping the unknown $w(t)$:

$$\frac{d\hat{x}(t)}{dt} = A\hat{x}(t) + B_2 u(t) + L(y(t) - C_2 x_2(t) - D_{22} u(t)).$$

The resulting estimation error $e(t) = \hat{x}(t) - x(t)$ will satisfy the differential equation

$$\dot{e}(t) = (A - LC_2)e(t) + (LD_{21} - B_1)w(t),$$

which will be stable whenever all eigenvalues of $A - LC_2$ have negative real part.



Pole Placement for Luenberger Observer

Can we place the eigenvalues of $A - LC_2$ arbitrarily by selecting L? Since the eigenvalues of a real matrix are always the same as the eigenvalues of its transpose, the task is the same as the one of placing eigenvalues of $A^T - C_2^T L^T$, which can be accomplished by using the `place` function:

```
F = ct.place(A.T, C2.T, listOfDesiredEigs)
L = F.T
```