APPPHYS 217 Thursday 15 April 2010

We begin by illustrating some of the concepts we examined last time (linear observers).

Consider, as our plant, a simple harmonic oscillator:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u,$$

$$\frac{d}{dt}\vec{x} = A\vec{x} + Bu,$$

where as usual $x_1 \leftrightarrow q,\ x_2 \leftrightarrow \dot{q}.$ The dynamics as given fix A and $B,$ and let us consider a general output signal related linearly to the state:

$$y = C\vec{x} = \begin{bmatrix} C_1 & C_2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

We have seen that the observability criterion is that we have full rank for the matrix

$$W_o = \begin{bmatrix} C \\ CA \end{bmatrix},$$

and since

$$CA = \begin{bmatrix} C_1 & C_2 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix} = \begin{bmatrix} -\omega_0^2 C_2 & C_1 \end{bmatrix},$$

we have

$$\det W_0 = \det\begin{bmatrix} C_1 & C_2 \\ -\omega_0^2 C_2 & C_1 \end{bmatrix} = C_1^2 + \omega_0^2 C_2^2.$$

Hence as long as $\omega_0 \neq 0,$ the system is observable as long as C is nonzero.

For general $C,$ our linear (Luenberger) observer structure is

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

which induces the following dynamics for the estimation error:

$$\frac{d}{dt}\tilde{x} = (A - LC)\tilde{x}.$$

We thus want to design L to make the eigenvalues of $A - LC$ have negative real part. Last time we noted that we could do this by using Matlab's pole-placement routine, **`place`**, with

$$A \leftrightarrow A^T, \quad B \leftrightarrow C^T, \quad K \leftrightarrow L^T.$$

We try the following examples, setting $\omega_0 = 1$ and designing for eigenvalues $\{-1,-2\}$:

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} : L = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \quad L(y - C\hat{x}) = \begin{bmatrix} 3 \\ 1 \end{bmatrix}(x_1 - \hat{x}_1),$$

$$C = \begin{bmatrix} 0 & 1 \end{bmatrix} : L = \begin{bmatrix} -1 \\ 3 \end{bmatrix}, \quad L(y - C\hat{x}) = \begin{bmatrix} -1 \\ 3 \end{bmatrix}(x_2 - \hat{x}_2),$$

$$C = \begin{bmatrix} 1 & 1 \end{bmatrix} : L = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad L(y - C\hat{x}) = \begin{bmatrix} 1 \\ 2 \end{bmatrix}(x_1 - \hat{x}_1 + x_2 - \hat{x}_2).$$

To examine how the observers work, we perform some numerical integrations. Setting $u = 0$ and

$$\vec{x}(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

we have for the plant evolution

$$\vec{x}(t) = \exp(At)\vec{x}(0) \mapsto \begin{bmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos t + \sin t \\ \cos t - \sin t \end{bmatrix}.$$

For the observer we assume no knowledge of the initial state, and thus set

$$\hat{x}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

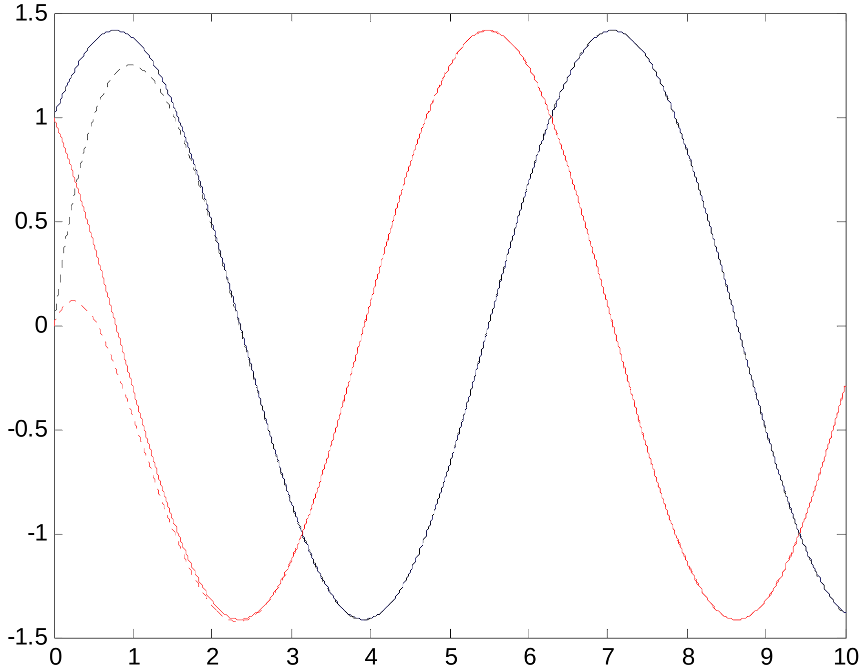
The dynamics of the state estimate is

$$\begin{aligned} \frac{d}{dt}\hat{x} &= \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} + Ly - LC\hat{x} \\ &= \left(\begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix} - LC \right) \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} + Ly. \end{aligned}$$

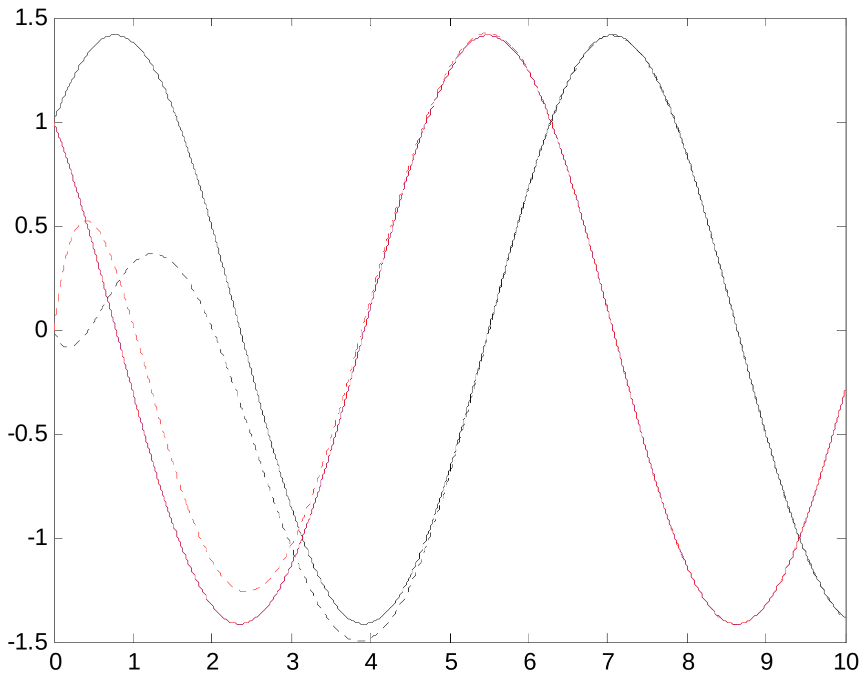
For the purposes of this example we could actually integrate this analytically, treating y as a driving term. However in the spirit of recursive state estimation we instead integrate numerically (Matlab example).

```
Ttot=10; Nsteps=5000;
t=linspace(0,Ttot,Nsteps); dt=t(2)-t(1);
x=[cos(t)+sin(t);cos(t)-sin(t)];
xhat=zeros(2,Nsteps);
for ii=2:Nsteps,
 y=C*x(:,ii);
 xhat(:,ii) = xhat(:,ii-1) +
dt*(A*xhat(:,ii-1)+L*(y-C*xhat(:,ii-1)));
end;
```

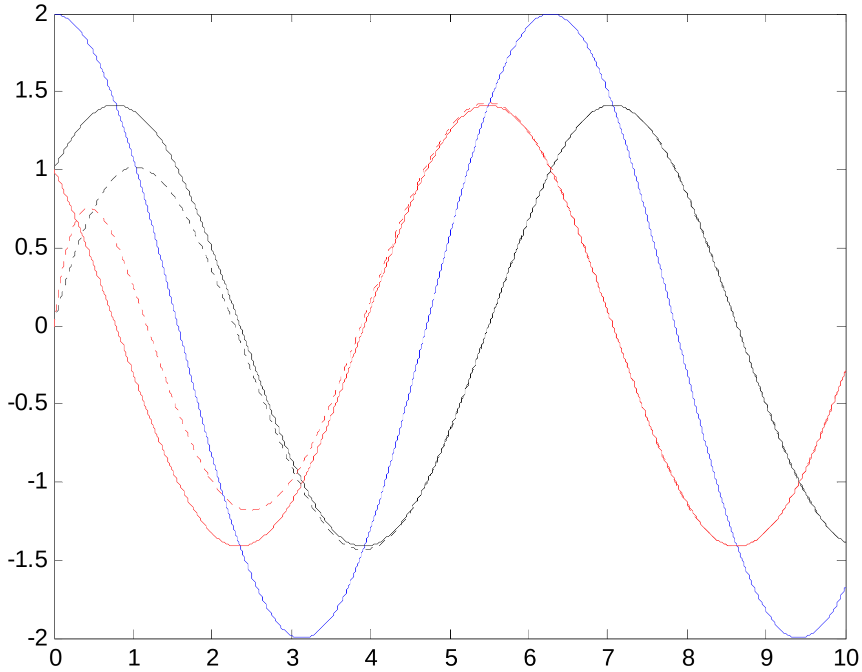
In the following we plot the results, with x_1 as solid black, x_2 as solid red, $\hat{x}_1$ as dashed black, and $\hat{x}_2$ as dashed red. The results for $C = \begin{bmatrix} 1 & 0 \end{bmatrix}$:



The results for $C = \begin{bmatrix} 0 & 1 \end{bmatrix}$:



The results for $C = \begin{bmatrix} 1 & 1 \end{bmatrix}$, with y in blue:



Could we have guessed the forms of these observers?

For $C = \begin{bmatrix} 1 & 0 \end{bmatrix}$, note that we can construct an "intuitive" observer by setting $\hat{x}_1 = y$, $\hat{x}_2 = \dot{y}$, which should work well as long as there is negligible measurement noise.

For $C = \begin{bmatrix} 0 & 1 \end{bmatrix}$ we obviously have $\hat{x}_2 = y$, and we can guess

$$\hat{x}_1 = \int_0^t ds\, y(s),$$

but it is not entirely clear how we should choose $\hat{x}_1(0)$. Note that the Luenberger observer does something more complicated, as it integrates

$$\frac{d}{dt}\hat{x} = \left(\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} - LC\right)\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} + Ly$$

$$\equiv A'\hat{x} + Ly, \quad A' = \begin{bmatrix} 0 & 2 \\ -1 & -3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -1 & -\frac{1}{2} \end{bmatrix}\begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} -1 & -2 \\ 2 & 2 \end{bmatrix},$$

$$\exp(A't) = \begin{bmatrix} 1 & 1 \\ -1 & -\frac{1}{2} \end{bmatrix}\begin{bmatrix} e^{-2t} & 0 \\ 0 & e^{-t} \end{bmatrix}\begin{bmatrix} -1 & -2 \\ 2 & 2 \end{bmatrix} = \begin{bmatrix} 2e^{-t} - e^{-2t} & 2e^{-t} - 2e^{-2t} \\ e^{-2t} - e^{-t} & 2e^{-2t} - e^{-t} \end{bmatrix},$$

$$\hat{x}(t) = \exp(A't)\left\{\hat{x}(0) + \int_0^t ds\, \exp(-A's)Ly\right\}$$

$$= \begin{bmatrix} 2e^{-t} - e^{-2t} & 2e^{-t} - 2e^{-2t} \\ e^{-2t} - e^{-t} & 2e^{-2t} - e^{-t} \end{bmatrix}\left\{\hat{x}(0) + \int_0^t ds \begin{bmatrix} 4e^{-s} - 5e^{-2s} \\ 5e^{-2s} - 2e^{-s} \end{bmatrix} y\right\},$$

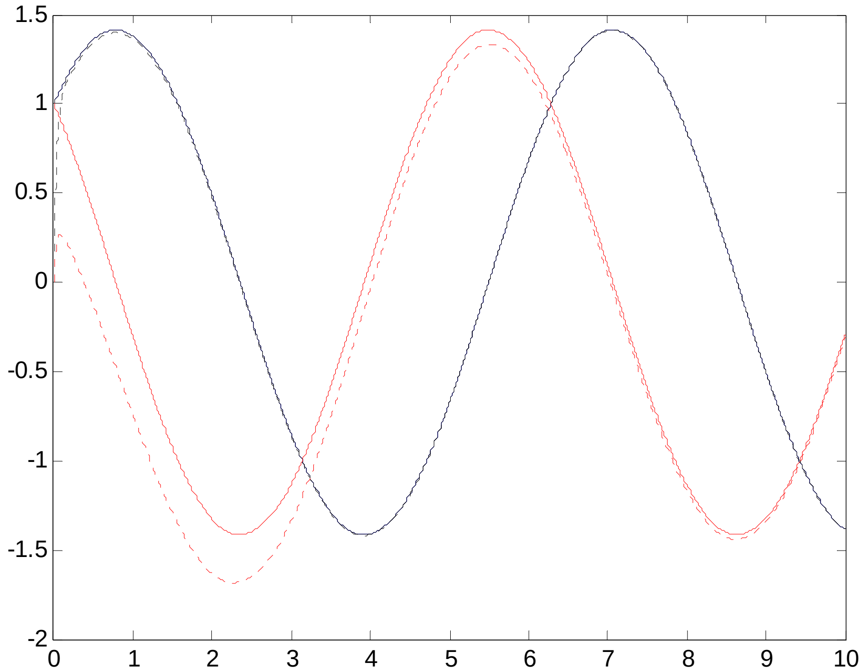
with $\hat{x}(0)$ arbitrary.

There doesn’t seem to be an obvious intuitive strategy for $C = \begin{bmatrix} 1 & 1 \end{bmatrix}$.

Before turning to consider noisy observation scenarios, we take a brief look at the behavior of the Luenberger observer for $C = \begin{bmatrix} 1 & 0 \end{bmatrix}$ and varying gain. Recall that when we designed for eigenvalues $\{-1, -2\}$ we obtained

$$L \to \begin{bmatrix} 3 \\ 1 \end{bmatrix}.$$

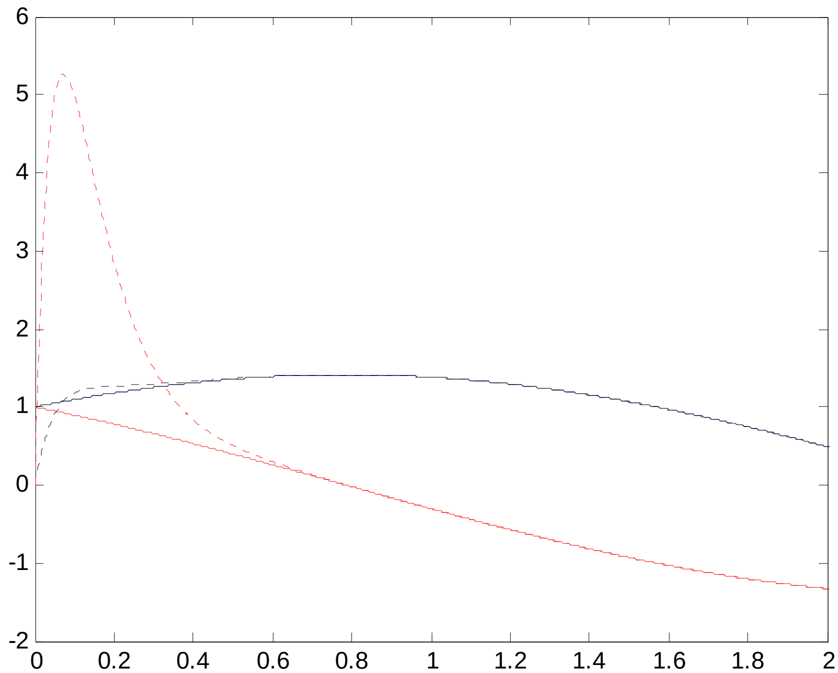
What happens if we try simply multiplying this gain by 10?



If instead we design for eigenvalues $\{-10,-20\}$ we get

$$L \to \begin{bmatrix} 30 \\ 199 \end{bmatrix},$$

and the performance is transiently bad, but does indeed settle quite quickly:



Stochastic models (notation)

We will normally write linear stochastic control models in the form

$$dx_t = Ax_t\,dt + Bu_t\,dt + FdV_t,$$
$$dy_t = Cx_t\,dt + G\,dW_t.$$

Here the subscripts serve to remind us of things that depend on time, and the vector nature of x and/or y is implicit. The stochastic increments dV_t and dW_t satisfy

$$\langle dV_t\rangle = \langle dW_t\rangle = 0,$$
$$dV_t^2 = dW_t^2 = dt,$$
$$dV_t dt = dW_t dt = 0,$$

and for $s \neq t$ we have $\langle dV_s dV_t\rangle = \langle dW_s dW_t\rangle = 0.$ We can informally think of dV_t/dt and dW_t/dt as gaussian white noises with zero mean and unit variance. It is conventional to refer to dV_t as process noise and to dW_t as measurement noise or observation noise.

It is important to be aware of the fact that stochastic differential equations (SDE's) of the type we have written are, rigorously speaking, a sort of shorthand notation for stochastic integrals. There is an important distinction between Itô and Stratonovich stochastic integrals, and therefore between Itô and Stratonovich SDE's. In control theory one normally works with Itô SDE's, and in any case there is a straightforward recipe for converting a model between Itô and Stratonovich forms.

The Stratonovich form is sometimes preferred (especially in physics) because Stratonovich SDE's can be manipulated using standard calculus. For Itô SDE's, however, one must in general be careful to observe the Itô Rule, which says that if x_t obeys the Itô SDE

$$dx_t = A(x_t)dt + B(x_t)dW_t,$$

then a variable y_t related to x_t via

$$y_t = U(x_t)$$

evolves according to

$$dy_t = \left[A(x_t)\frac{\partial U}{\partial x} + \frac{1}{2}B^2(x_t)\frac{\partial^2 U}{\partial x^2}\right]dt + B(x_t)\frac{\partial U}{\partial x}dW_t,$$

where the second-derivative term in the square brackets is known as the Itô correction. Note that if U is a linear function the Itô correction vanishes and we recover the prediction of normal calculus.

An important advantage of working with Itô SDE's is that if x_t obeys the Itô SDE

$$dx_t = A(x_t)dt + B(x_t)dW_t,$$

then x_t is uncorrelated with dW_t. This considerably simplifies the computation of statistical moments.

For example consider the linear SDE model

$$dx_t = Ax_t\,dt + FdV_t,$$

with x_t a scalar and $A < 0$ (the Ornstein-Uhlenbeck model). We then have

$$\begin{aligned} d\langle x_t\rangle &= A\langle x_t\rangle dt + F\langle dV_t\rangle \\ &= A\langle x_t\rangle dt, \\ \langle x_t\rangle &= \langle x_0\rangle \exp(At), \end{aligned}$$

and if $y_t = x_t^2$, so that $\langle y_t\rangle$ is the variance of x_t,

$$\begin{aligned} dy_t &= [2Ax_t^2 + F^2]dt + 2Fx_t\,dV_t, \\ d\langle y_t\rangle &= [2A\langle y_t\rangle + F^2]dt + 2F\langle x_t\,dV_t\rangle \\ &= [2A\langle y_t\rangle + F^2]dt + 2F\langle x_t\rangle\langle dV_t\rangle \\ &= [2A\langle y_t\rangle + F^2]dt, \\ \langle y_t\rangle &= \exp(2At)\left\{\langle y_0\rangle + \int_0^t ds\,\exp(-2As)F^2\right\} \\ &= \exp(2At)\left\{\langle y_0\rangle + F^2\int_0^t ds\,\exp(-2As)\right\}. \end{aligned}$$

If we assume that x_t evolves from a known value x_0 at $t = 0$, then $\langle x_0\rangle = x_0$ and $\langle y_0\rangle = x_0^2$, and the mean-square uncertainty in x_t is

$$\begin{aligned} \langle x_t^2\rangle - \langle x_t\rangle^2 &= \langle y_t\rangle - \langle x_t\rangle^2 \\ &= \exp(2At)F^2\int_0^t ds\,\exp(-2As) \\ &= \exp(2At)F^2\left(-\frac{1}{2A}\right)(\exp(-2At) - 1) \\ &= -\frac{F^2}{2A}(1 - \exp(2At)). \end{aligned}$$

The mean-square uncertainty thus has a steady-state value as $t \to \infty$,

$$\langle x_t^2 \rangle - \langle x_t \rangle^2 \to \frac{F^2}{2|A|}.$$

In numerical simulations, we can simply update x_t according to

$$x_{t+dt} = x_t + A x_t \, dt + B u_t \, dt + F \, dV_t,$$

$$dy_t = C x_t \, dt + G \, dW_t,$$

where dV_t and dW_t are independent normal random variables with variance dt. In Matlab, if **`dt`** is a variable with some assigned numerical value,

```
dVt=sqrt(dt)*randn(1); dWt=sqrt(dt)*randn(1);
```

This simple procedure is known as the Itô-Euler stochastic integration routine, which is easy to implement but has the disadvantage that it only converges to order $(dt)^{1/2}$. Higher-order integrators can be found in various computer packages (including SDE toolboxes for Matlab), and are described in textbooks.

State observers - performance with noise

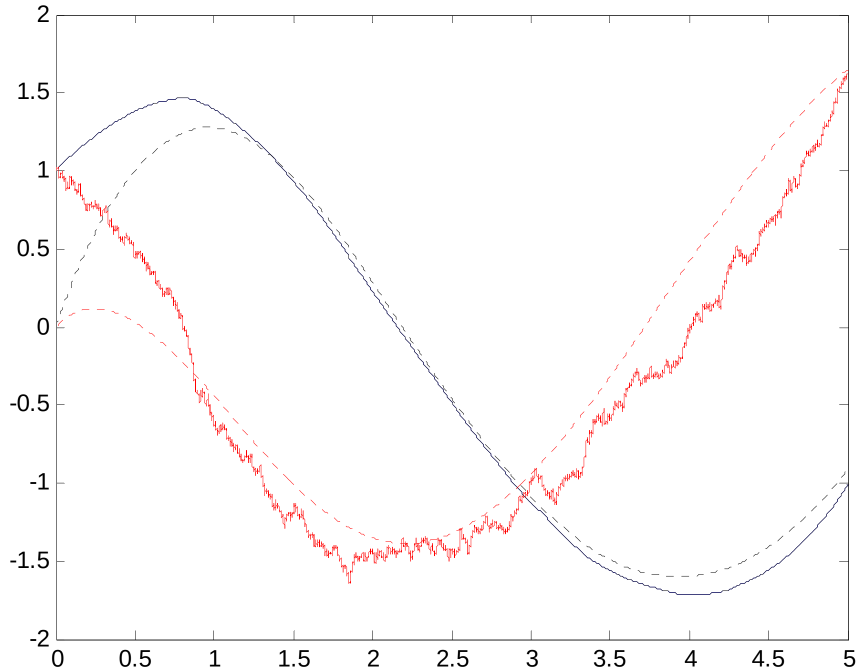
First we consider the case of process noise only. Returning to our simple harmonic oscillator with $C = \begin{bmatrix} 1 & 0 \end{bmatrix}$, we can add a noisy force acting on the oscillator by setting

$$F = \begin{bmatrix} 0 \\ 0.3 \end{bmatrix}.$$

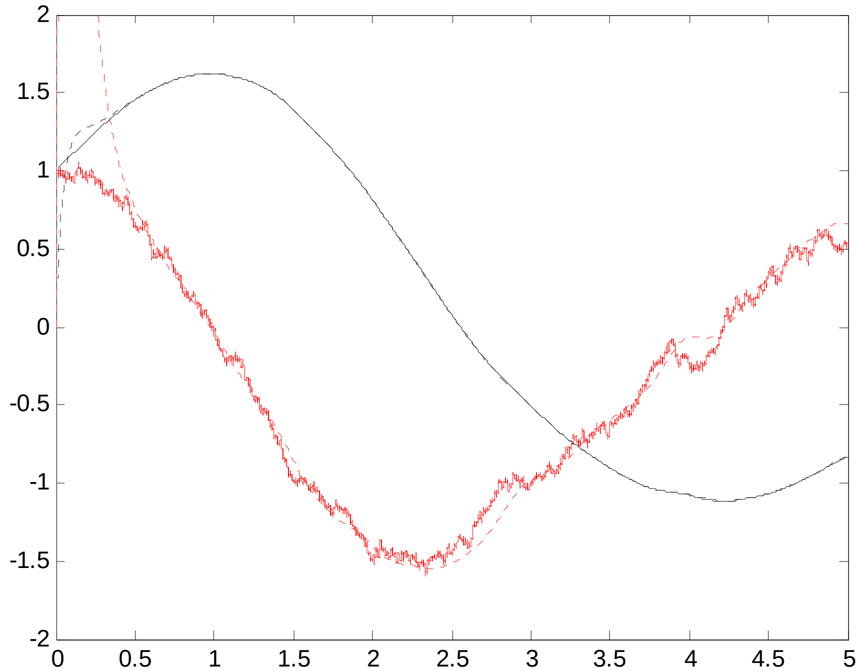
Simulating both the plant and the response of the Luenberger observer, which we now write as

$$\hat{x}_{t+dt} = \hat{x}_t + (A - LC) \hat{x}_t \, dt + B u_t \, dt + L \, dy_t,$$

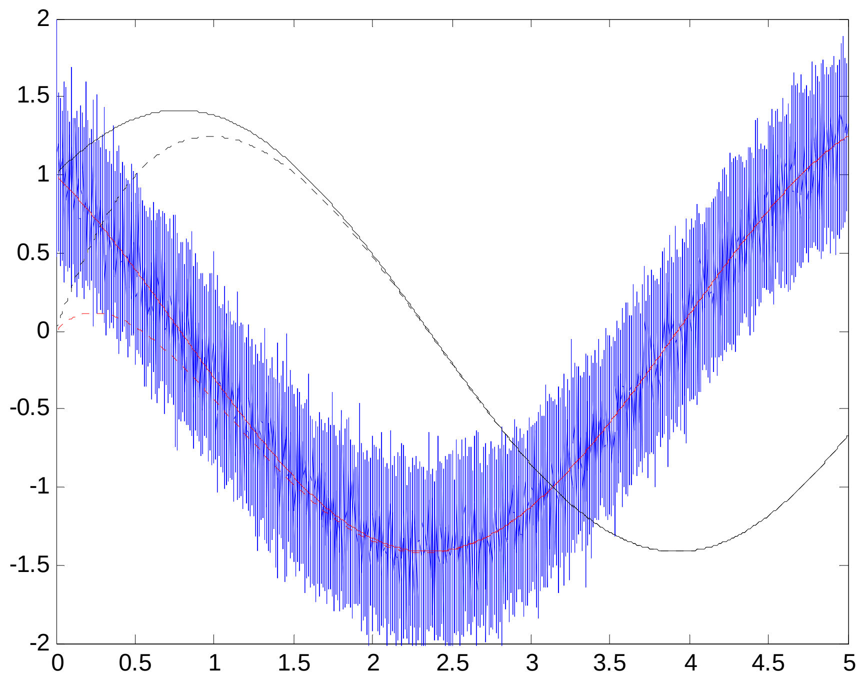
we obtain:



Here we used the L computed for target eigenvalues $\{-1,-2\}$. It is clear that the observer still tracks the state but with degraded performance due to the process noise. If we try turning up the observer gain by using values computed for target eigenvalues $\{-10,-20\}$, we do much better:

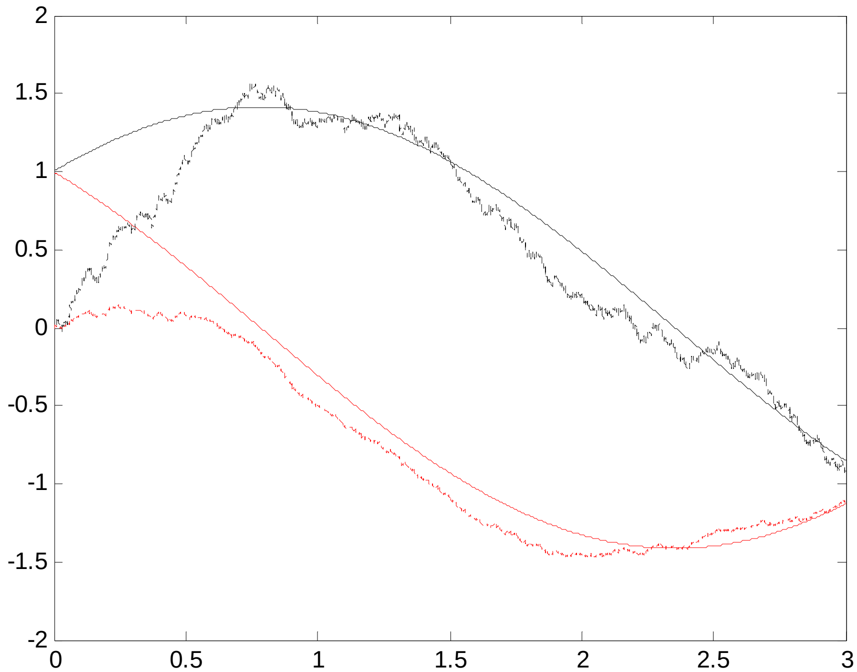


Next we set F to zero but $G = 0.001$. The results are:

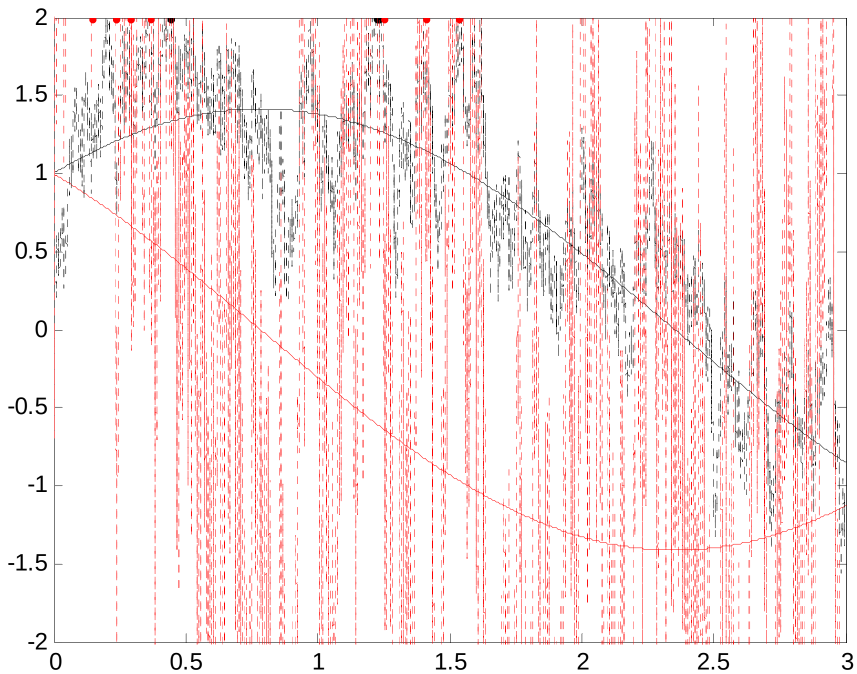


Here the blue shows the performance of a naive velocity estimator, $\hat{x}_2 = \dot{y}$ (note that there is some aliasing). The Luenberger observer does much better, as expected.

In order to bother the Luenberger observer we turn G all the way up to 0.1:



Now if we try to get greedy with this much observation noise, by turning up L to the values that would achieve eigenvalues $\{-10,-20\}$ in the noiseless system,



and we see that the estimation of velocity becomes very poor. So apparently too much observer gain is a bad thing, when there is noise. Is there an optimal value of the Luenberger gain? This would seem to be an especially important question when there is both process noise and measurement noise.

The Kalman-Bucy filter

To answer this sort of question we first have to state how we judge the observer's performance quantitatively. It is most common to adopt a minimum-least-squares framework, in which our objective is to design the estimator (method of generating $\hat{x}_t$ from knowledge of y_s with $s \leq t$) that achieves the lowest possible value of $\langle (x_t - \hat{x}_t)(x_t - \hat{x}_t)^T \rangle$. As discussed in A&M section 7.4, for the plant and observation model

$$dx_t = Ax_t\,dt + Bu_t\,dt + F\,dV_t,$$
$$dy_t = Cx_t\,dt + G\,dW_t,$$

we have the following Theorem:

(Kalman-Bucy, 1961) The optimal estimator has the form of a linear observer

$$d\hat{x}_t = (A\hat{x}_t + Bu_t)dt + L_t(dy_t - C\hat{x}_t dt), \quad \hat{x}_0 = \langle x_0 \rangle,$$

where $L_t = P_t C^T [GG^T]^{-1}$ and $P_t = \langle (x_t - \hat{x}_t)(x_t - \hat{x}_t)^T \rangle$ is the (symmetric and positive-definite) estimation error covariance matrix that satisfies the following matrix Riccati equation:

$$\frac{d}{dt}P_t = FF^T + AP_t + P_t A^T - P_t C^T [GG^T]^{-1} CP_t, \quad P_0 = \langle x_0 x_0^T \rangle.$$

It is important to note that the Kalman filter provides both a point estimate of the evolving system state and a computation of the estimation error covariance matrix - it gives you its best guess *and* a numerical uncertainty. When the system is stationary and if P_t converges, the observer gain settles to a constant:

$$L = PC^T[GG^T]^{-1}, \quad FF^T + AP + PA^T - PC^T[GG^T]^{-1}CP = 0.$$

The second equation is called the *algebraic Riccati equation*, and may be solved using Matlab's **`lqe`** function.

We see that the essence of Kalman filtering is an optimal choice of the observer gain, which may be time-dependent in a way that reflects our evolving degree of confidence in our state estimate. The general structure is to apply high observer gain when we have large uncertainty, and to reduce it when our uncertainty approaches a limiting value set by the process and measurement noises.

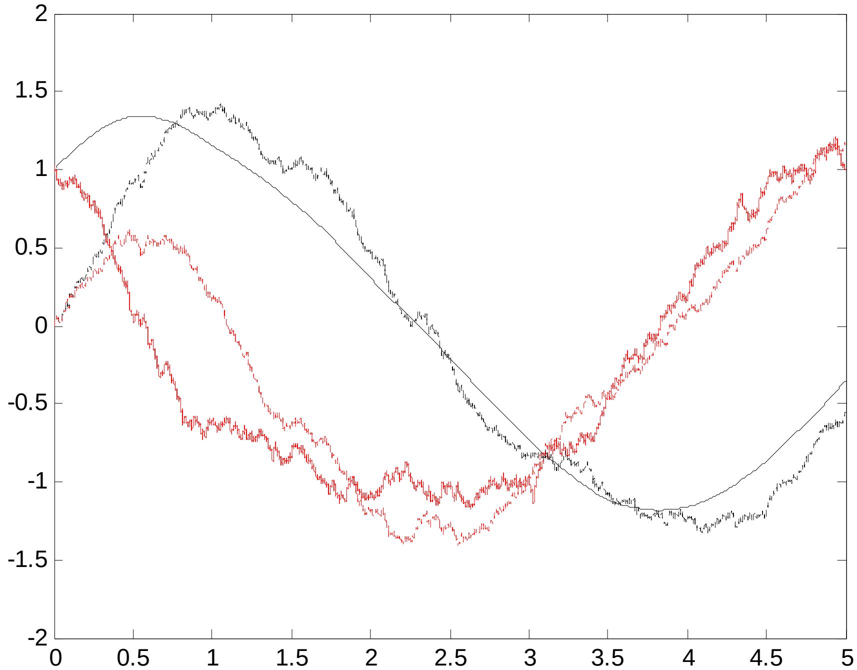
As an example let us compute the Kalman gain for our simple harmonic oscillator example with

$$F = \begin{bmatrix} 0 \\ 0.3 \end{bmatrix}, \quad G = 0.1.$$

This results in

$$L \approx \begin{bmatrix} 2.08 \\ 2.16 \end{bmatrix},$$

and a simulation looks as follows:



It is interesting to note that, as a consequence of its least-squares optimality, the Kalman-Bucy filter achieves what is known as "whitening" of the innovations process $dy_t - C\hat{x}_t dt$. That is, if $\hat{x}_t$ is propagated by the Kalman-Bucy filter then $(dy_t - C\hat{x}_t dt)$ becomes a completely random signal (Gaussian white noise); roughly we can think that $\hat{x}_t$ becomes good enough that subtracting $C\hat{x}_t\,dt$ from dy_t removes all the information from the observed signal. The notions of least-squares optimal state estimation, the innovations process, and whitening all carry over to nonlinear scenarios.