0.1 Kalman Filtering Introduction

These notes will be used in my book but are based on George Kantor's spoon feeding to me his knowledge of Kalman filtering.

0.1.1 Observers

Newton has taught us that force equals mass multiplied by acceleration, i.e., $m\ddot{z} = F$ where m is the mass of the system, $z\colon \mathbb{R} \to \mathcal{WS}$ such that $z(t) \in \mathcal{WS}$ is the position of the system, and $F\colon \mathbb{R} \to T^*_{z(t)}\mathcal{WS}$ is the force acting on the mass. The solution to the equations of motion for a system ususally describe the trajectory that a mass follows. Equations of motion are derived from some sort of force balance. For example, a Hooke's Law tells us the the force induced by a spring is its spring constant k multiplied by the displacement of the spring. Therefore, $m\ddot{z} = -kz$ for a mass-spring system where z is the displacement from the equilibrium position. Assuming a starting velocity of zero, the solution to this differential equation is $z(t) = z(0)\cos(\sqrt{\frac{k}{m}})$ where $z(0)$ is the initial position of the mass. Given the mass and spring constant, i.e., the system parameters, the specifications of $z(0)$ and $\dot{z}(0)$ determine the solution to the equation of motion, i.e., the trajectory of the mass.

Typically, a mechanical system has damping, i.e., something that slows down the motion of the mass. This is modeled by a dashpot whose force contribution is proportional to the mass'velocity. This is modeled by a damping coefficient γ, i.e., $m\ddot{z} = -\gamma\dot{z}$. Therefore, a mechanical system with a dashpot and spring, as depicted in Figure 1, has an equation of motion $m\ddot{z} + \gamma\dot{z} + kz = 0$, whose solution is $z(t) = \exp^{-\gamma t} z(0)\cos(\sqrt{\frac{k}{m}})$.

This second order differential equation can be re-written as two first order differential equations or a first order linear vector equation as

$$\frac{d}{dt}\begin{bmatrix} z \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k & -\gamma \end{bmatrix}\begin{bmatrix} z \\ \dot{z} \end{bmatrix}. \tag{1}$$

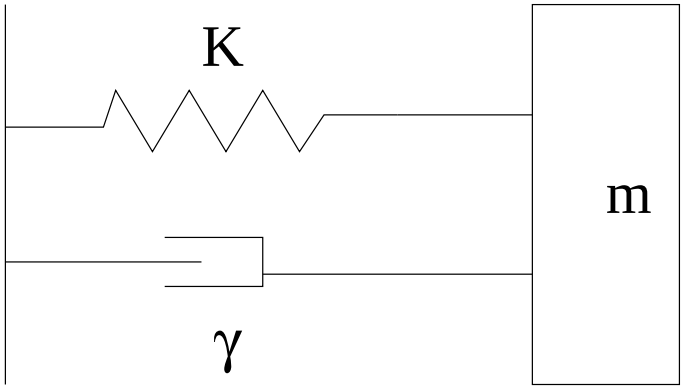

Figure 1. Mass-Spring-Dashpot System

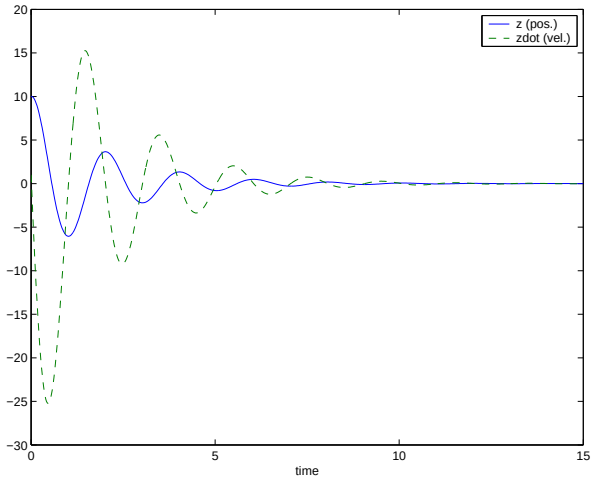

Figure 2. The position (solid line) and velocity (dashed line) with $m = 1, k = 10$, and $\gamma = 1$

This equation is generically written as

$$\dot{x} = Ax, \tag{2}$$

where

$$x = \begin{bmatrix} z \\ \dot{z} \end{bmatrix} \tag{3}$$

Typically, x is refered to as the *state* of the system. The matrix A describes how the state transitions when it experiences dynamics. Sometimes, A is refered to as the *plant.*

Now, lets introduce the notion of an input to our system. This input is also a force, which in the mass-spring-dashpot system, acts on the mass. The first order linear vector ordinary differential equation becomes

$$\frac{d}{dt}\begin{bmatrix} z \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k & -\gamma \end{bmatrix}\begin{bmatrix} z \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{4}$$

which in state space form can be written as

$$\dot{x} = Ax + bu, \tag{5}$$

Assume the mass is 1. The above system will naturally come to rest because of the damper. However, if the damper is removed and we want the system to come to rest, we can regulate the input $u \in \mathbb{R}$ to bring the mass to rest. We can "try" the feedback control law $u = Kx$ to bring the mass to rest, i.e., ensure the system to be exponentially stable. Subsituting the control law back into our equations of motion,

$$\begin{aligned} \dot{x} &= Ax + bu \\ &= Ax + bKx \\ &= (A + bK)x \end{aligned} \tag{6}$$

To make the above system stable, we want $A + bK$ to be negative definite.

$$A = \begin{bmatrix} 0 & 1 \\ -k & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \text{and} \quad K = [\alpha_1 \;\; \alpha_2] \tag{7}$$

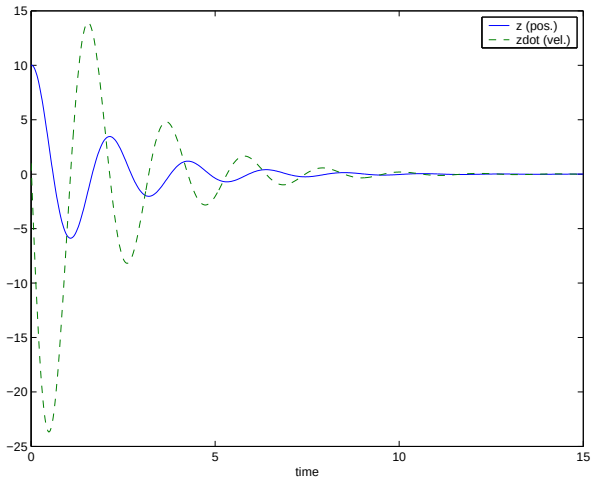

Figure 3. The position (solid line) and velocity (dashed line) with $m = 1, k = 10$, and $\gamma = 0$ with $\alpha_1 = 1$ and $\alpha_2 = -1$.

So, we want

$$\left[\begin{bmatrix} 0 & 1 \\ -k & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \alpha_1 & \alpha_2 \end{bmatrix}\right] = \begin{bmatrix} 0 & 1 \\ -k + \alpha_1 & \alpha_2 \end{bmatrix}$$

to be negative definite. So, we chose a K, i.e., an α_1 and α_2, to make the system stable. As an excercise for the reader, one can determine that $\alpha_1 < k$ and $\alpha_2 < 0$ for the above system to be stable. See Figure 3 for an example of a system stabilized by a feedback control law.

The feedback control law, stated above, depends upon the state of the system. The state, however, is internal to the plant and not directly observable. Therefore, we use an *observer* to estimate the state of the plant. We denote the estimate as $\hat{x}$. Since we know A (the plant model), b, and the control input u, we can look at

$$\dot{\hat{x}} = A\hat{x} + bu$$

to determine $\hat{x}$ via integration.

The question is: when does $\hat{x} = x$? If $\hat{x}(0) = x(0)$, then $\hat{x} = x$ from the state equation. So, an accurate determination of the initial conditions is all that is

required to estimate the state for all time. Unfortunately, we do not know $x(0)$. Therefore, consider

$$\dot{\hat{x}} = A\hat{x} + bu + K(C\hat{x} - y)$$

where A, $\hat{x}$, C, and y are known (or measurable). Note that $C\hat{x} - y$ is the error between the estimated output and the true output. This measure lies in a "sensor space." The K matrix weights this error and converts it to "state space." Without knowing the initial conditions $x(0)$, the goal is to chose a K matrix such that $\hat{x}$ converges onto x. Note that the K matrix here is different from the K matrix defined in the control law, stated above.

Let $e = \hat{x} - x$ be the error between the actual state and the observed state. Consider $\dot{e} = \dot{\hat{x}} - \dot{x}$, i.e.,

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + bu + K(C\hat{x} - y) \\ \dot{x} &= Ax + bu + K(Cx - y) \text{note that } y = Cx \text{ by definition} \\ \dot{e} &= A(\hat{x} - x) + KC(\hat{x} - x) \\ &= (A + KC)e \end{aligned} \tag{8}$$

In order to drive error to zero, $A + KC$ must be negative definite. Let $K = [\kappa_1 \ \kappa_2]$. As an excercise to the reader, derive the values of κ_1 and κ_2 that makes $A + KC$ to be negative definite. One will find that $\kappa_1 < 0$ and $\kappa_2 < k$. In Figure 5, we chose $K = [-11]^T$ and estimated the initial conditions at $[11]^T$ for system whose real initial conditions are $[10]$ (Figure 4. In other words, we guessed correctly at the starting location but incorrectly at the starting velocity. Note that the error between the observed state and actual state decays to zero as can be see in Figure 6.

0.1.2 Discrete Time

Many times, we take sensor readings and make state changes in discrete time steps. Therefore, lets consider a discrete version of Equation 5. So, instead of $\dot{x}$, consider at time step k over a length of time T

$$\dot{x}(k) \simeq \frac{x(k+1) - x(k)}{T}$$

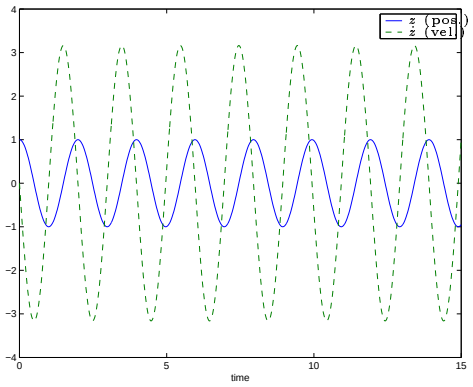

Figure 4. Actual system with initial conditions $x(0) = [10]^T$

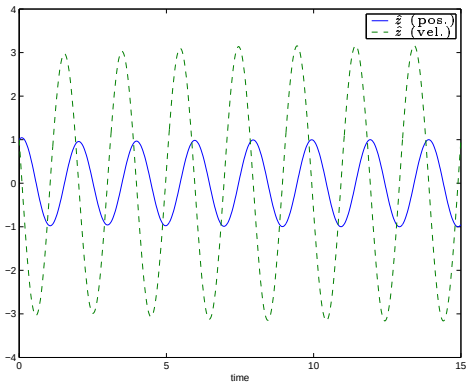

Figure 5. Estimated system with estimated initial conditions $\hat{x}(0) = [11]^T$

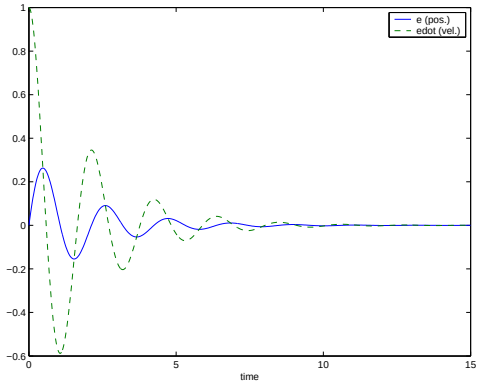

Figure 6. Error between the actual and esimate system.

Substitute this into the state equation

$$\frac{x(k+1)-x(k)}{T} = Ax(k) + Bu(k)$$

resulting in a new discrete state equation

$x(k+1) = Fx(k)+Gu(k)$, where $F = (I+TA)$, $G = TB$, and I is the identity matrix

With zero input (i.e., $u(k) = 0$ for all k), the above system is stable if F has eigenvalues of magnitude less than one. Note this is different than the continuous system where negative eigenvalues assured a solution to a differential equation with a decaying exponential. Here, the solution to the difference equation decays because we are constantly multiplying a value by another and in order to force it to decay, we want the magnitude of this value to be less than one.

For the above system with $u(k) = 0$ and $A = \begin{bmatrix} 0 & 1 \\ -k & 0 \end{bmatrix}$, $F = \begin{bmatrix} 1 & T \\ -kT & 1 \end{bmatrix}$ has eigenvalues $1 \pm T\sqrt{-k}$. Note that this system in discrete time is *not* stable because one of the eigvalues of F has magnitude greater than one (if $T \neq 0$, in which case we have a marginally stable continuous system, as above). However, stabilitiy is not the issue here; we still can construct an observer for this system.

The output equation $y = Cx$ becomes $y(k) = Hx(k)$. If C and H are constant for all time, then $C = H$. However, to distinguish between a continuous output and discrete output equation, we change notation here and use H. The question is: when does $\hat{x}(k) = x(k)$? If $\hat{x}_0 = x_0$, then you have $\hat{x}(k) = x(k)$ from the state equation. So, if you can accurately guess your initial conditions, all is good. Unfortunately, you do not know x_0. Therefore, consider

$$\hat{x}(k+1) = F\hat{x}(k) + Gu(k) + K(H\hat{x}(k) - y(k))$$

where we know F, $\hat{x}(k)$, C, and $y(k)$. We do not know the initial conditions and we are going to chose a K. Again, $H\hat{x}(k)$ is our estimated output and $y(k)$ is the actual output. The difference between the two, i.e., the error, is converted

to state space via the K matrix. So, the $K(H\hat{x}(k) - y(k))$ can be viewed as a correction in state space based on sensor readings.

Just as before, consider $e(k) = \hat{x}(k) - x(k)$.

$$\begin{aligned} e(k+1) &= \hat{x}(k+1) - x(k+1) \\ &= F\hat{x}(k) + Gu(k) + K(H\hat{x}(k) - y(k)) - (Fx(k) + Gu(k) + K(Hx(k) - y(k))) \\ &= F(\hat{x}(k) - Fx(k)) + KH(\hat{x}(k) - x(k)) \\ &= Fe(k) + KHe(k) \\ &= (F + KH)e(k) \end{aligned} \tag{9}$$

The error goes to zero when the eigenvalues of $(F + KH)$ have magnitude less than one. For the above example where $H = [1\ 0]$ (i.e., we measure position of the mass), $K = [\kappa_1\ \kappa_2]^T$, $(F + KH)$ takes the form $\begin{bmatrix} \kappa_1 + 1 & T \\ \kappa_2 - kT & 1 \end{bmatrix}$ whose eigenvalues are

$$\frac{\kappa_1}{2} + 1 \pm \frac{\sqrt{\kappa_1^2 - 2\kappa_1 + 4\kappa_2 T - 4kT^2}}{2}.$$

To maximize the stability of the system, chose κ_1 that minimizes the magnitude of the above eigenvalues. We can do that by making the real part zero and the imaginary part as small as possible. To make the real part of the eigenvalues zero, choose $\kappa_1 = -2$. After noting that the magnitude of the eigenvalues should be less than one for stability, we arrive at $1 + \kappa_2 T - kT^2 \leq 1$, i.e., $\kappa_2 \leq kT$. To maximize the stability, the imaginary part of the eigenvalues should also be set to zero, i.e., $\kappa_2 = kT - \frac{1}{T}$. ***** IS this previous paragraph right??? *****

We repeat the example from Figure 5, but for a discrete system. In Figure 8, we chose $K = [-1\ 1]^T$ and estimated the initial conditions at $[1\ 1]^T$ for system whose real initial conditions are $[1\ 0]$ (Figure 7. Note that the error between the observed state and actual state decays to zero as can be see in Figure 9. Here, $\kappa_1 = -2$ and $\kappa_2 = -49.8$ for maximal stability.

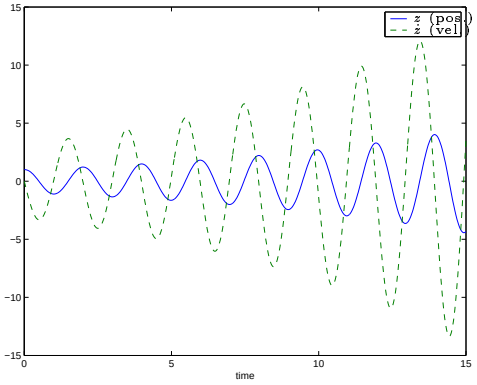

Figure 7. Actual system with initial conditions $x(0) = [1\ 0]^T$

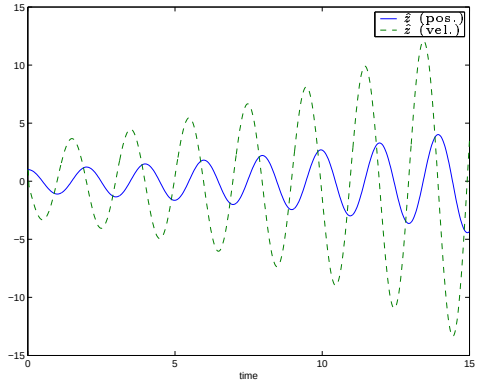

Figure 8. Estimated system with estimated initial conditions $\hat{x}(0) = [11]^T$

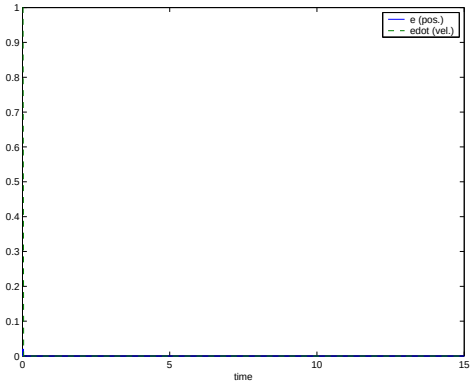

Figure 9. Error between the actual and esimate system.

0.1.3 Two-step Observer

We have real measurements and we have a model. For the moment, assume that there is no measurement error nor is there process noise. We use our linear model to estimate the state in the next time step ($x(k+1)$) and then we correct the model, i.e., our predicted state, based on real sensory measurements ($y(k)$). Therefore, we need to compare our model to actual measurements in order to generate the correction. Since we cannot directly measure the state of the system, we need to model how the state of the system is measured. In a sense, we work backwards. We start with a measurement and want to infer our state. We assume that our measurement is a linear function (H) of state. Note that our system and measurement model are both linear. For the dynamic system

$$x(k+1) = Fx(k) + Gu(k) \tag{10}$$

$$y(k) = Hx(k), \tag{11}$$

where $F \in \mathbb{R}^{m \times m}, G \in \mathbb{R}^{m \times n}$, and $H \in \mathbb{R}^{p \times m}$ are functions that model the system, consider a simple observer that employs the following two steps:

1. **Predict:** Use dynamic equation and last estimate to predict what the next state estimate.

2. **Update:** Correct predicted estimate so that the result is consistent with the observed output.

Now, lets introduce some notation to help make this idea more clear. Let $\hat{x}(k|k)$ be the current estimate at time step k. Let $\hat{x}(k+1|k)$ denote the predicted state estimate based on our plant model, i.e., $\hat{x}(k+1|k)$ is the estimate of $\hat{x}$ at time $k+1$ given the output $y(k)$ just before getting output $y(k+1)$. Let $\hat{x}(k+1|k+1)$ denote the corrected state estimate, i.e. $\hat{x}(k+1|k+1)$ is the estimate $\hat{x}$ at time $k+1$ given the output $y(k+1)$. Note that generally $\hat{x}(k+1|k+1)$ will be a better estimate than $\hat{x}(k+1|k)$ since $\hat{x}(k+1|k+1)$ incorporates more recent output data.

Using the new notation, the prediction step can be written as

$$\hat{x}(k+1|k) = F\hat{x}(k|k) + Gu(k),$$

which is the best estimate of x based at time $k+1$ based on everything known at time k. Note that the predicted estimate of the next state depends only upon the system model dynamics, F and G, and not any measurements.

We are now going to correct our predicted estimate $\hat{x}(k+1|k)$ using sensory data. A sensor reading is taken, i.e., $y(k+1) = H(k)x(k|k)$ is determined. All possible states that correspond to this sensor reading forms the set

$$\Omega(k+1) = \{\alpha \in \Re^n | H\alpha = y(k+1)\}.$$

To determine the next time step's estimate $\hat{x}(k+1|k+1)$, we pick the closest state in Ω to our current prediction $\hat{x}(k+1|k)$. In other words, $\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + \Delta x$ where $\Delta x = H^T(HH^T)^{-1}(y(k+1) - H\hat{x}(k+1|k))$. See Figure 10. The geometric intuition can be seen in Figures 10 and 11 which represents a system with two state variables. Note that $\hat{x}(k+1|k)$ is the best guess for the state. We assume that this estimate is good. Also, right now, we do not know anything about our error distribution, so all states that lie on a circle centered at $\hat{x}(k+1|k)$ are equally less likely to be correct than $\hat{x}(k+1|k)$, if $\hat{x}(k+1|k)$ is believed to be correct. So, the larger the circle, the less likely the estimate is. We want to pick the estimate that is "most" consistent with our predicted estimate and our output. The line $\Omega = \{x \,|\, Hx = y\}$ corresponds to the set of states that all correspond to the same output y. The state that is closest to $\hat{x}(k+1|k)$ on this line is the state that is most consistent with our predicted estimate and jibes with our sensor readings.

Consider the same state transition matrix A and sensor model H as above, but with initial conditions $x(0) = [10\ 4]^T$, but we guess initial conditions as $\hat{x}(0|0) = [0\ 0]^T$. See Figure 12. Using the above equations, the estimated state is depicted in Figure 13. Note that the error (Figure 14) goes to zero in one time step because we are dealing with a linear system with no noise.

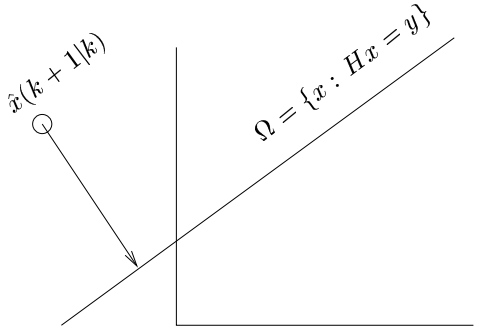

Figure 10. Ω is the set of states corresponding to the current sensor readings. The corrected state lies in this set and is the state which is closest to the predicted estimate.

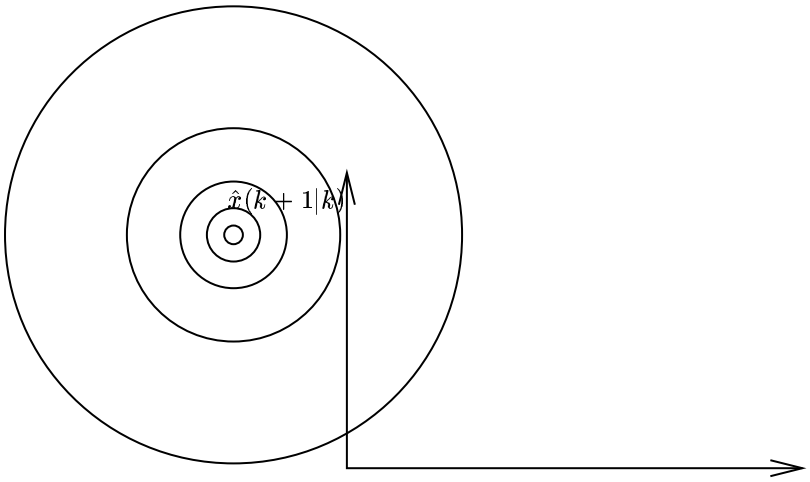

Figure 11. Our estimate $\hat{x}(k+1|k)$ is deemed to be good. All states surrounding this estimate are less likely to be correct.

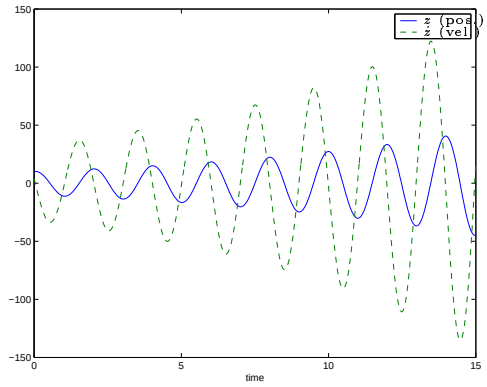

Figure 12. Actual system with initial conditions $x(0) = [10\ 4]^T$

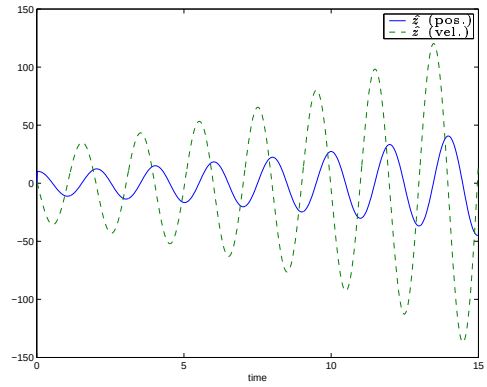

Figure 13. Estimated system with estimated initial conditions $\hat{x}(0|0) = [0\ 0]^T$

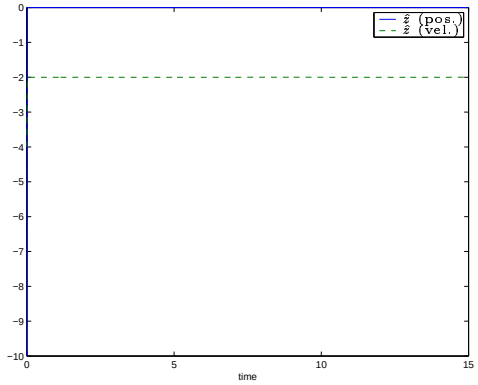

Figure 14. Error between the actual and esimate system.

0.1.4 A Three-Step Observer

Alas, the estimate of x is not exact. Instead, Kalman suggests that we estimate a distribution p whose mean is $\hat{x}$. The idea here is that we want the average value of the state estimate to equal the average value of the actual state. We also want the estimate to vary from the actual state as much as possible. The distribution used in Kalman filtering is a Guassian

$$p(x) = \frac{1}{(2\pi)^{\frac{n}{2}}|P|^{\frac{1}{2}}} \exp^{-\frac{1}{2}\left((x-\hat{x})^T P^{-1}(x-\hat{x})\right)}$$

where P is the co-variance matrix for p. We will use the co-variance matrix in the correction phase, but the prediction and the update remain the same. Before, we picked a value that was closest to our estimate and was consistent with our sensor readings, i.e., we picked the point on Ω that was closest to $\hat{x}(k+1|k)$. In other words, we picked a vector Δx that minimizes $\|\Delta x\| = \Delta x^T \Delta x$ which was the orthogonal projection of $\hat{x}(k+1|k)$ onto Ω. The Δx described the radius of a circle centered at $\hat{x}(k+1|k)$ and tangent to Ω.

Now, we are going to change our notion of "closest" and "orthogonal." Instead of considering a circle, we will consider an ellipse whose center is still $\hat{x}(k+1|k)$, but whose principal axes are defined by the eigenvectors of P. Directions associated with larger eigenvectors have larger uncertainties, i.e., have higher variances. Upon first inspection, a correction that parallels the largest eigenvector could yield a good choice for a correction because you want to move in the direction that you are least certain. However, moving exclusively in one direction is not ideal because there are other variables that have to be corrected based on the sensory readings. Therefore, chose the estimate where the ellipse is tangent to Ω. See Figure 15.

Therefore, a more powerful observer can be created by estimating the error covariance matrix as well as the state. The error covariance matrix P is defined to be

$$P(k|k) = E\left((x(k) - \hat{x}(k|k))(x(k) - \hat{x}(k|k))^T\right),$$

and

$$P(k+1|k) = E\left((x(k+1) - \hat{x}(k+1|k))(x(k+1) - \hat{x}(k+1|k))^T\right).$$

Now, the procedure uses three steps

- Predict:

  As before, the state prediction is given as $\hat{x}(k+1|k) = F\hat{x}(k|k) + Gu(k)$, but now there is a the prediction step for the error covariance matrix. which is

  $$P(k+1|k) = FP(k|k)F^T.$$

- Correct:

  Using sensor data, we compute a correction for both the covariance matrix and the state. These corrections are:

  $$K = P(k+1|k)H^T\left(HP(k+1|k)H^T\right)^{-1} \quad (12)$$
  $$\Delta x = K(y(k+1) - H\hat{x}(k+1|k)) \quad (13)$$
  $$(14)$$

- Update:

  Using the correction, the updates become

  $$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + \Delta x \quad (15)$$
  $$P(k+1|k+1) = P(k+1|k) - KHP(k+1|k)H^TK^T \quad (16)$$
  $$(17)$$

Essentially, we have redefined our notion of an inner product in $\mathbb{R}^m$ from $\langle x_1, x_2\rangle = x_1^T x_2$ to $\langle x_1, x_2\rangle = x_1^T P^{-1} x_2$. This induces a new norm $\|x\| = \langle x, x\rangle = x^T P^{-1} x$. The $\hat{x}$ in Ω that minimizes $\|\Delta X\|$ is again the orthogonal

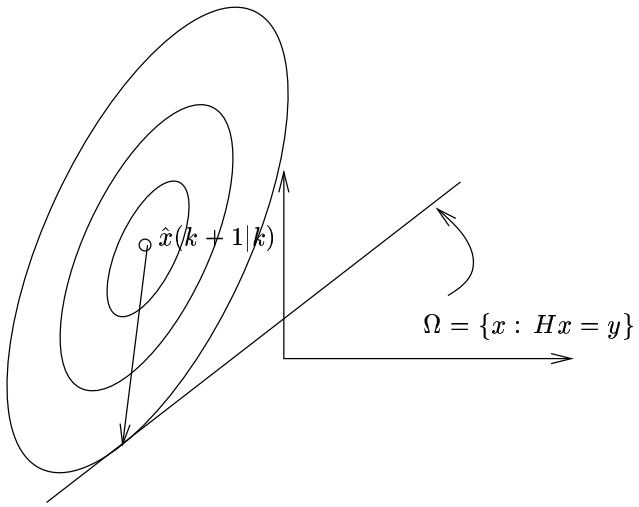

Figure 15. Correction determines the "closest" state on the set of states Ω but now closest is based on an ellipse.

projection of $\hat{x}(k+1|k)$ onto Ω, but our notion of orthogonal has changed. Since Δx is orthogonal to Ω, $\langle \omega, \Delta x \rangle = 0$ for all $\omega \in \Omega = (H)$. Therefore,

$$\langle \omega, \Delta x \rangle = \omega^T P^{-1} \Delta x = 0 \iff \Delta x \in \text{column}(PH^T).$$

Since the norm has changed, so has distance. Again, assume that $\hat{x}(k+1|k)$ is a "good" estimate. Therefore, all states that are the "same distance" away from $\hat{x}(k+1|k)$ are all equally "less good" from the predicted estimate. However, the notion of distance is now modified. Instead of using d: $\mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ with $d(x,y) = (x-y)^T(x-y)$, $d(x,y)$ becomes $d(x,y) = (x-y)^T M(x-y)$, which is called the Manhabilis distance funtion. The M weights which variables have a greater effect when measuring distance.

If M is the identity, then the Manhabilis distance function reduces to the standard Euclidean metric and values of equidistance are associated with circles. If M is a diagonal matrix (not equal to the identity), then values of equidistance are associated with ellipses where the eigenvalues correspond to the magnitude of the ellispe's principal axes and the eigenvectors to their diretion

Now, lets look at $K = P(k+1|k)H^T \left(HP(k+1|k)H^T\right)^{-1}$. From the cor-

rection equations, it can be seen that K maps from sensor space to state space. The mapping $\left(HP(k+1|k)H^T\right)^{-1}$ maps from sensor space to state space. H maps from state space back to sensor space and then P maps from sensor space to state space.

0.1.5 Kalman Equations (Linear)

Now, let there be noise: process noise v in the state transition equation and measurement noise w in the output equations, both modeled by Gaussian distributions with zero mean and covariance matrices V and W, respectively. Process noise comes from those unfortunate realities that screw up your system which should have been "perfectly" modeled. Examples of process noise for mobile robots include wheel slippage, bumps in the floor, and mismatch in wheel diameters which are supposed to be the same. Essentially, process noise can be considered as model mismatch. The other type of noise – measurement noise – is another fact of life; there is no avoiding it. Measurement noise is a result of imperfect sensors which must be accounted for in the measurement model.

Sensor noise prevents us from computing an exact correction term because sensor readings no longer correspond to the set Ω, but rather a distribution centered on Ω. See Figure 16. Determining an algrebraic correction is not possible, so this is now an optimization problem. Kalman determined the result of this optimzation as $P(k+1|k)H^T(HP(k+1|k)H^T+W)^{-1}(y(k+1)-H\hat{x}(k+1|k))$.

The system and observer equations now become:

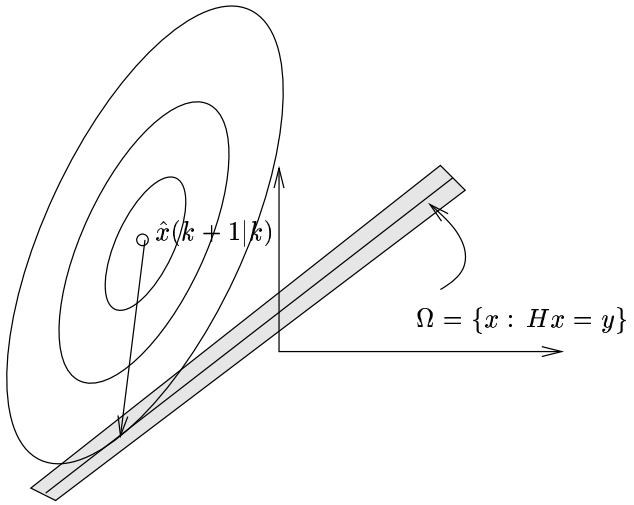

Figure 16. Correction determines the "closest" state on the set of states Ω but now closest is based on an ellipse.

System:	Dynamics	$x(k+1) = \overbrace{F(k)}^{\text{plant}}\overbrace{x(k)}^{\text{state}} + \overbrace{G}^{\text{How the inputs drive dynamics}} u(k) + \overbrace{v(k)}^{\text{process noise}}$
	Output	$y(k) = \underbrace{H(k)}_{\text{measurement model}} x(k) + \underbrace{w(k)}_{\text{measurement noise}}$

Observer:	Predict	$\hat{x}(k+1\|k) = F\hat{x}(k\|k) + Gu(k)$
		$P(k+1\|k) = FP(k\|k)F^T + V$
	Correct	$S = HP(k+1\|k)H^T + W$
		$K = P(k+1\|k)H^T S^{-1}$
		$\Delta x = K(y(k+1) - H\hat{x}(k+1\|k))$
	Update	$\hat{x}(k+1\|k+1) = \hat{x}(k+1\|k) + \Delta x$
		$P(k+1\|k+1) = P(k+1\|k) - KSK^T$

The difference $y(k+1) - H\hat{x}(k+1|k)$ is often termed the *innovation* and the K the *Kalman gain*, which determines how much to weigh the sensor readings versus the system model. If sensor noise is low, then measurements have high credibility, i.e., $\lim_{W(k)\to 0} K(k) = H(k)^{-1}$. The innovation gains more weight because sensor readings are trusted more. In other words, if the K is

"high," then the sensor readings are more believable than the model. If our model is quite good and sensor readings have a lot of error, then the predicted measurement carries more weight, i.e., $\lim_{P(k)\to 0} K(k) = 0$.

Lets look at S. This matrix is the covariance matrix for the innovation which represents the uncertainty of the innovation. The matrix S^{-1} mapped through H^T, i.e., $H^T S^{-1}$, can be viewed as an innovation "certainty" matrix in the state space. This certainty matrix gives rise to the ellipses in Figures 15 and 16. So, moving along in the major axis of the ellipse is equivalent to moving in a direction of most certainty. Likewise, moving in a direction along the minro axis is equivalent to moving in a direction of least certainty. Therefore, motion in that direction is desirable, but we do not want to exclusively move in that direction. Therefore, we move in a direction that trades off the most and least certain variables by looking for a point where the certainty ellipse becomes tangent to the set of states corresponding to the observed measurements. GEORGE, IS THIS LAST PARAGRAPH CORRECT?

It is important to note that the process noise V and the sensor noise W must be guassian with zero mean in order to guarantee that the average of the state estimate equals the average of the actual state with minimal variability.

0.1.6 Extended Kalman Filetering

Consider the following state equations

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) + v(k)(f, u(k), x(k), ...) \\ y(k) &= h(x(k)) + w(k) \end{aligned} \tag{18}$$

where $x(k) \in \mathbb{R}^m$ is the state at time k and $x(k+1) \in \mathbb{R}^m$ is the state at time $k+1$. The transition from state $x(k)$ to $x(k+1)$ is a function of the dynamics $f\colon \mathbb{R}^m \to \mathbb{R}^m$ of the system and process noise $v(k)$. Naturally, the dynamics depends upon the current state $x(k)$ and the controlled input $u(k) \in \mathbb{R}^n$. Moreover, the process noise v could be a function of the controlled input, the dynamics, state, etc.

Predicting the state based on the model is $x(k+1|k) = f(x(k|k), u(k))$.

The innovation is $y(k) - h(x(k+1|k))$. To determine the remaining quantities, linearize the non-linear Kalman equations, i.e., let

$$F_{ij} = \frac{\partial f_i}{\partial x_j}(\hat{x}(k), u(k)) \quad \text{and} \qquad H_{ij} = \frac{\partial h_i}{\partial x_j}(\hat{x}(k), u(k)).$$

0.2 Vehicle Navigation Example (This is taken Dan Simon's notes from the Web at embedded.com. It seems pretty basic and I changed it a bit)

Consider a robot moving along a straight line. Its state is position z and velocity $\dot{z}$. The input u is the value of acceleration and the output y is the position of the robot. If this sensor were perfect and if the robot perfectly moved along the straight line without any problems such as slippage, then the velocity of the robot at time step $k+1$ is $\dot{z}(k+1) = \dot{z}(k) + Tu(k)$ assuming that we can measure position and velocity in intervals of T. However, due to slippage and other unforeseen difficulties, we have to add an error term v_2 so the velocity becomes $\dot{z}(k+1) = \dot{z}(k) + Tu(k) + v_2$. Likewise, there is a position equation $z(k+1) = z(k) + Tv(v) + \frac{1}{2}T^2u(k) + v_1$ where v_1 is the position noise. Let x be the state vector and the state equations become

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} \frac{T^2}{2} & T \end{bmatrix} u(k) + v \\ y(k) &= \begin{bmatrix} 1 & 0 \end{bmatrix} + w \end{aligned}$$

where $v = [v_1 \; v_2]^T$ is the process noise and w is the sensor noise. If $T = .1$ seconds and the input acceleration is 1 meter/second^2, then the state equations become

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 1 & .1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} .005 & .1 \end{bmatrix} + v \\ y(k) &= \begin{bmatrix} 1 & 0 \end{bmatrix} + w \end{aligned}$$

In other words

$$F = \begin{bmatrix} 1 & .1 \\ 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} .005 & .1 \end{bmatrix}, \quad \text{and} \quad H = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Assume position is measured with an error to one standard deviation of 10 meters. The co-variance matrix of sensor noise is simply $W = 100$. The process noise co-variance matrix is a two by two matrix. Since the input noise has a standard deviation of .2 meter/second^2 and since the mapping from input space to state equals is .005 times the acceleration, the variance of position is $(.005 \times .2)^2 = 10^{-6}$. Likewise, the variance of velocity is $(.1 \times .2)^2 = 4.0 \times 10^{-4}$. Finally, the off-diagnal terms of co-variance matrix are the product of the standard deviations after they are mapped from input space to state space, i.e., $(.005 \times .2)(.1 \times .2) = 2 \times 10^{-5}$. So, the co-variance matrix is

$$V = \begin{bmatrix} 10^{-6} & 2 \times 10^{-5} \\ 2 \times 10^{-5} & 4.0 \times 10^{-4} \end{bmatrix}.$$

0.3 Hugh-Durrant Whyte/Leonard: Range/Bearing and Augmented State

Leonard and Durrant-Whyte are among the first to apply Kalman filtering to the mobile robot localization problem [**?**]. They use a *geometric beacon* which is a target that can be observed in successive sensor measurements. A beacon can be a piece of technology that broadcastes its absolute or relative position (or more generally state) to the robot. A beacon can also be a feature that is stable and naturally occurring in an environment, which could be stored in a map prior to deploying the robot. Here, a map is a list of beacon locations and not a roadmap or a pixel/grid structure. As the robot moves about the environment, it incurrs positioning error but uses the beacons, via Kalman filtering, to update its locations. Let b be the number of beacons. Since the beacon locations are known ahead of time, this process is called *localization.*

The state of the robot at time step k is $x(k) \in SE(2)$. The system equations

are

$$\begin{array}{rll} \text{plant} & x(k+1) & = f(x(k), u(k)) + v \\ \text{measurement} & y_j(k) = h_i(p_i, x(k)) + w_j & \end{array}$$

where $y_j(k)$ is the jth measurement. Note that this value probably does not correspond to the jth beacon. In other words, the robot receives a n sensor readings indexed by j without knowing the one-to-one mapping that corresponds measurements to beacons. The v and w are values from the process and measurement noise, respectively, and are samples from zero mean Guassians whose co-variance matrices are V and W_j, respectively.

The robot starts off with an estimate of its location and uncertainty, i.e., an initial state $\hat{x}(0)$ and $P(0)$. Their method has four steps: predict, observe, match, and estimate. The prediction stage is the same as before: use the plant or system model to predict what the next state is going to be, i.e., $\hat{x}(k+1|k) = f(\hat{x}(k|k), u(k))$ whose associated variance is $P(k+1|k) = F\,P(k|k)\,F^T + V$ where V is the co-variance of the process noise and F is the Jacobian of f. The prediction stage also uses the predicted state to determine a predicted output, i.e., $\hat{y}_i(k+1) = h_i(p_i, \hat{x}(k+1|k))$ for each geometric beacon p_i.

In the observation stage, n measurements are taken $y_j(k+1)$ and then compared to the predicted measurements $\hat{y}_i(k+1)$. Note that $n \leq b$. The innovation is denoted $\nu_{ij} = \hat{y}_i(k+1) - y_j(k+1)$. Likewise, the innovation co-variance matrix is $S_{ij} = H_i P(k+1|k) H_i^T + W_j$ (It bothers me that j really does not appear) where H_i is the Jacobian of h evaluated at $\hat{x}(k+1|k)$ and p_i.

The next step is to identify which observed measurement corresponds to a predicted measurement. To this, a thresholding scheme using the Manhabilis norm is used. The algorithm accepts all observations for which

$$\nu_{ij}(k+1) S_{ij}^{-1} \nu_{ij}^T(k+1) \leq g,$$

for some value g. We *hope* (and in fact force) there to be a one-to-one correspondance between the predicted and estimated measurements. In other words,

for each j there is only one i which satisfies the gate. Naturally, this assumption is naive and requires more thought.

The final step is estimation. Assume that n is now the number of measurements pass through the gate. We form three composite matrices by stacking matrices and measuresments associated with the n measurements that satisfy the gate equation. Note that the $\dim(y_j(k+1)) = m$ and thus $\dim(\nu_{ij}) = m$. Therefore, $S_{ij} \in \mathbb{R}^{m \times m}$.

We form a single $z(k+1) \in \mathbb{R}^{m \times n}$ by stacking the $z_j(k+1)$ readings that satisfy the gate equation. BUT I DO NOT SEE WHERE HE USES IT. Likewise, a composite innovation vector $\nu(k+1) \in \mathbb{R}^{n \times m}$ is formed by stacking the innovations which pass through the gate. Next, the Jacobians H_i are stacked to form a matrix $H \in \mathbb{R}^{(m \times n) \times 3}$ and $H^T \in \mathbb{R}^{3 \times (m \times n)}$. With this in hand, the Kalman gain is

$$K(k+1) = P(k+1|k)H^T S^{-1}(k+1)$$

which is the same as before. The updated vehicle state is

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + K(k+1)\nu(k+1)$$

CAN WE DO AN EXAMPLE

CAN WE MAKE THIS SLAM RELEVANT

0.4 Kantor: Range only

0.5 Bearing only

0.6 Skaff Example

0.7 Intuition behind the Covariance Matrix

0.7.1 Error Relationships

Let x be a random variable. Recall that a random variable can be viewed as a map from the set of possible events to a real number. For example, heads or tails are the events, but their corresponding values could be $x(\text{heads}) = 1$ and $x(\text{tails}) = 0$. Another example is a the classic urn with red and blue balls. A random variable can be the number of red balls selected from the urn. Many times, the event space of x can be real numbers or vectors, so for short hand we can view $x \in \mathbb{R}^m$.

The *expected value* for a discrete event is $E(x) = \sum_i p_i E_i$ where E_i is the ith value of the random variable x and p_i is the probability that this event occurs. Note that there is this temptation to think that the expected value is the outcome most likely to occur. Although this can be true many times – consider the expected value of sum value of two dice; it is seven. However, the expected value of one die is 3.5 which of course cannot occur.

The expected value of continuous random variable is $E(x) = \int_t f_x(t)E(t) \in \mathbb{R}^m$ where f_x is the probability density function. Sometimes $E(x)$ has μ as notation and other times it has $\bar{x}$. Note that expectation is a linear operator. So, $E(x - \bar{x}) = E(x) - E(\bar{x}) = \bar{x} - \bar{x} = 0$.

The *variance* of a random variable x is $E((x - \bar{x})(x - \bar{x})^T)$ which reduces to $E((x - \bar{x})^2)$ for scalar random variables. For a scalar random variable the variance is denoted σ^2 and for a vector random variable, the ith variance is denoted σ_i^2. Electrical Engineers can think of mean as the DC and variance

as the energy in the signal. Variance is like variability; the more energy in a system, the more variability. There is a theorem **(Ercan, MUST LOOK THIS UP)** that states if one acquires many many samples they will converge onto a normal distribution and have a mean which is the average of the samples.

Now we want to consider the effect of one variable on another. This is termed *correlation* between two random variable x_i and x_j. **(Ercan, GET A FORMAL DEFINITION HERE).** Let $\sigma_{ij} = E((x_i - \bar{x}_i)(x_j - \bar{x}_j)^T)$ Note that σ_{ii} is σ_i^2, the variance of x_i. For $i \neq j$, if $\sigma_{ij} = 0$ then x_i and x_j are independent of each other. When $\bar{x}_i$ and $\bar{x}_j$ are both zero for scalar independent random variables x_i and x_j, $\sigma_{ij} = E(x_i x_j) = E(x_i)E(x_j)$.

We will look at the effect of correlation on a two-dimensional density function. The ellipsoids in Figure **??** correspond to the set of random variables (really the images of random variables) with the same value froma two-dimensional density function. The ellipsoid on the left corresponds to a bi-variate Gaussian which was formed by two independent one-dimensional Gaussian distributions, N_1 and N_2, both with zero mean, but different variances. It is worth noting how we form this two-dimensional Gaussian. Assume the N_1 distribution has the x_1 axis as its domain and the N_2 distribution has the x_2 axis. The N_1 distribution is then extruded in the vertical direction (i.e., along the x_2 axis). So, any horizontal cut or slice of the extruded distribution is the original N_1 distribution. Likewise, the N_2 distribution is extruded in the horizontal direction and any vertical slice is the original N_2 distribution. A new function is formed where N_1 and N_2 are point-wise multiplied together. This function looks like a Guassian, having a big bump centered at the origin and will not integrate to one and thus must be rescaled, i.e., multiplied by a constant so that it integrates to one. A horizontal slice of this function is a not a Gaussian, but rather a scaled version of N_1 and likewise, a vertical slice of the resulting function is a scaled version of N_2.

The ellipsoid in the left hand side of Figure **??** corresponds to two independent variables because moving in the x_1 direction (along the dotted line) does not change the distribution in the x_2 direction. Here, $\sigma_{ij} = 0$ for $i \neq j$. In the

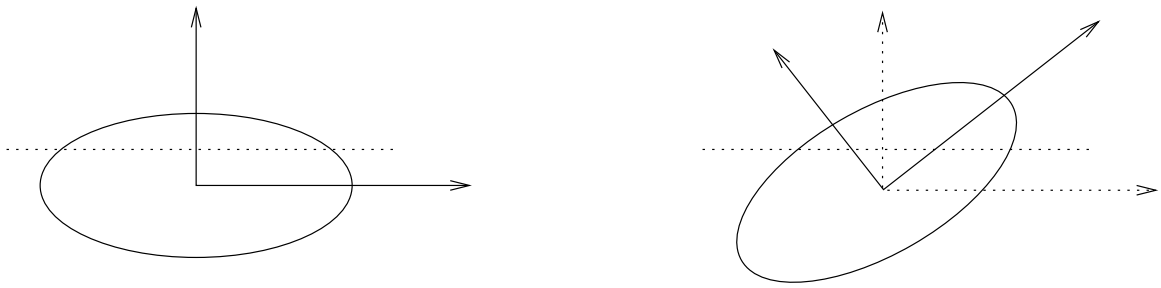

Figure 17. Two examples of joint distributions where the ellipsoids correspond to the set of events of equal probability, i.e., isoprobability lines

right hand side of Figure **??**, x_1 and x_2 depend on each other. Upon inspection, one can see how the mean of x_1 changes as x_2 varies. In other words, the mean of x_1 depends upon the mean of x_2 and vica versa.

With the σ_{ij}'s, we can form a matrix termed a *covariance matrix,* which encodes the correlation of random variables among each other. For the above example, the covariance matrix takes the form

$$\begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix}$$

The eigenvectors of the covariance matrix are the principal axes of the ellipse (ellipsoid in higher dimensions) with eigenvalues corresponding to the magnitude (eccentricity) of the ellipse. Note that if all the random variables are completely independent of each other, then all of the off-diagonal terms will be zero, i.e., the covariance matrix is diagonal and thus has eigenvectors parallel to the coordinate axes which means the ellipse is not rotated. If the covariance matrix has all non-zero terms then every variable depends upon every other variable.

0.7.2 Error Relationships – need to relook

Kalman's main contribution is to estimate a distribution as opposed to a value. The distribution can be modeled by its mean and covariance. The premise is that the estimate is "good," i.e., the average estimate is the average reading. Let $\hat{x}$ be the estimate for x and P be x's covariance matrix on the error from

the estimate, i.e.,

$$P = E\left(\left((x-\hat{x}) - \left(\bar{x}-\bar{\hat{x}}\right)\right)\left((x-\hat{x}) - \left(\bar{x}-\bar{\hat{x}}\right)\right)^T\right) \tag{19}$$

Recall, that we will assume that $\bar{x}$ is zero. Moreover, we will assume that our estimates are "good." In other words, we will assume that the average of the actual state is approximately equal to the average of the estimated state, therefore on average, $\bar{x} - \bar{\hat{x}}$ vanishes. Therefore, we are left with

$$P = E\left((x-\hat{x})(x-\hat{x})^T\right) \tag{20}$$

Lets investigate the geometry of the above. Then for some reason **??????**, George tells me the mean error $\hat{x} - x$ is zero for reasons that will become apparent, but they did not nor did I know why they were. Also, should it be $E(\hat{x} - x)$?

In Figure 18, the top set of ellipses is the density on the error, whose mean lies at the origin. The left ellipse is the top ellipse, but is centered on $\hat{x}$. This distribution models the possible locations of x given $\hat{x}$. The right set of ellipses is again the error density, but centered on x. This distribution models the possible locations of $\hat{x}$ given x. From here, we can see that $E\left(\left(\hat{x}-\bar{\hat{x}}\right)\left(\hat{x}-\bar{\hat{x}}\right)^T\right) = E\left((\hat{x}-\bar{x})(\hat{x}-\bar{x})^T\right)$ ****** Did not see where these were going? Did I get it right *****

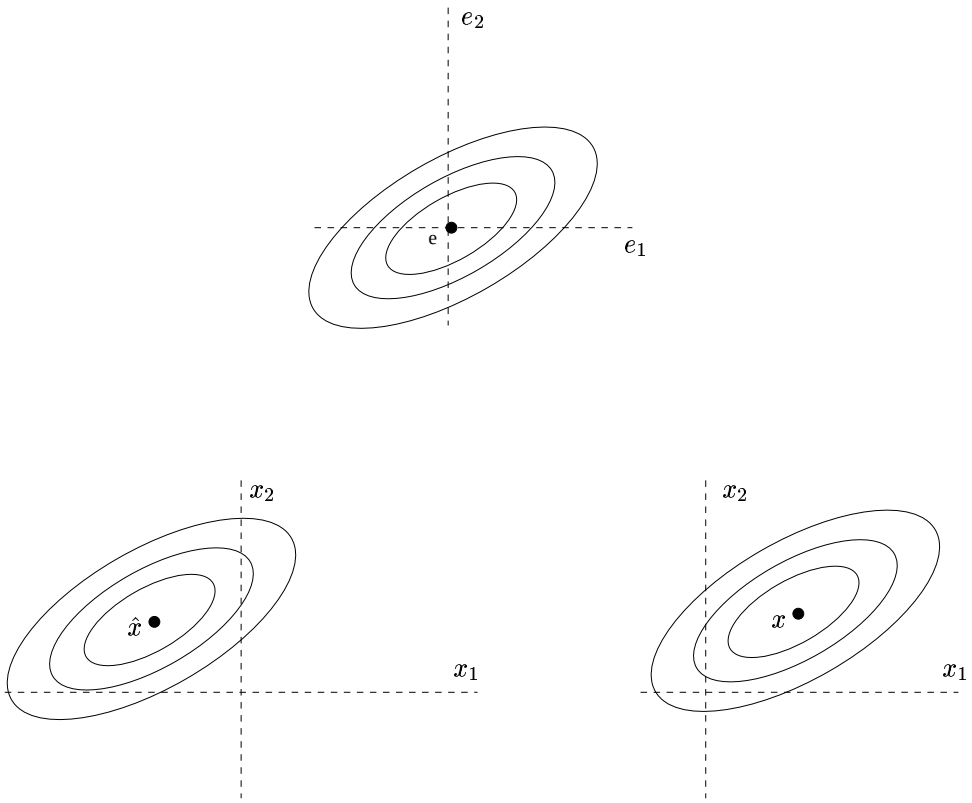

Figure 18. The top figure was the diagonal ellipses centered at zero on coordinate axes e_1 and e_2 because we are considering error. The bottom two figures are in the x_1 and x_2 axes. The first has a center in the upper left quadrant with $\hat{x}$ at the center. The second has a center in the upper right quadrant with x at the center.