Kalman Filter

Kalman Filter

Predict:

$$x_{k|k-1} = F_k x_{k-1|k-1} + B_k u_k$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q$$

Update:

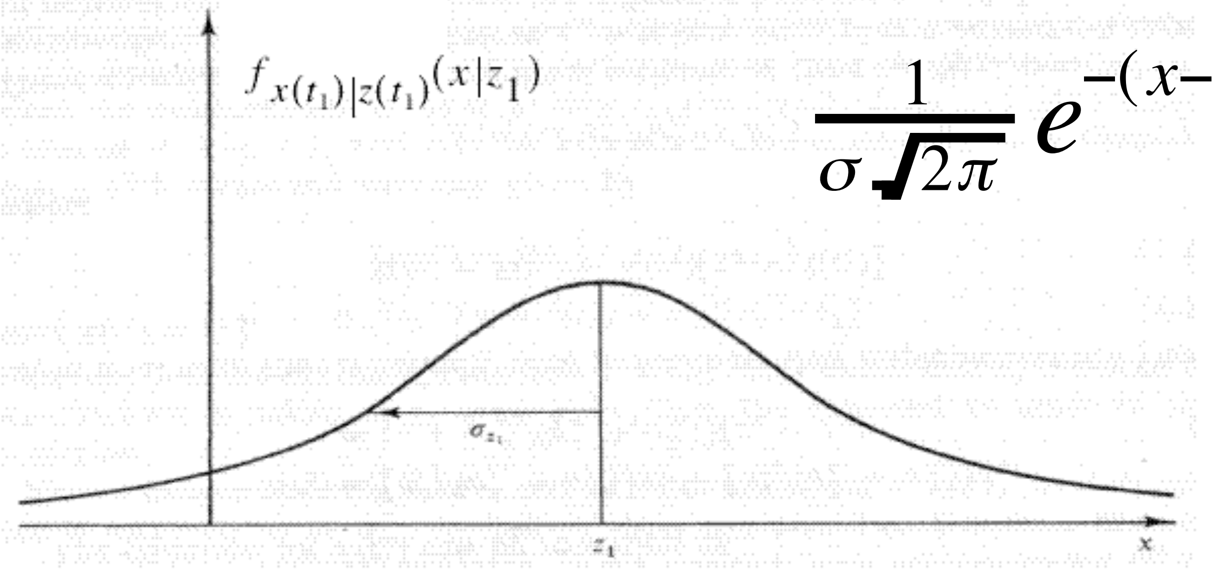
$$K = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R)^{-1}$$

$$x_{k|k} = x_{k|k-1} + K(z_k - H_k x_{k|k-1})$$

$$P_{k|k} = (I - K H_k) P_{k|k-1}$$

Gaussian (Normal) Distribution

- Completely described by $N(\mu,\sigma)$
  - Mean μ
  - Standard deviation σ, variance σ^2



$$\frac{1}{\sigma\sqrt{2\pi}}\, e^{-(x-\mu)^2/2\sigma^2}$$

The Central Limit Theorem

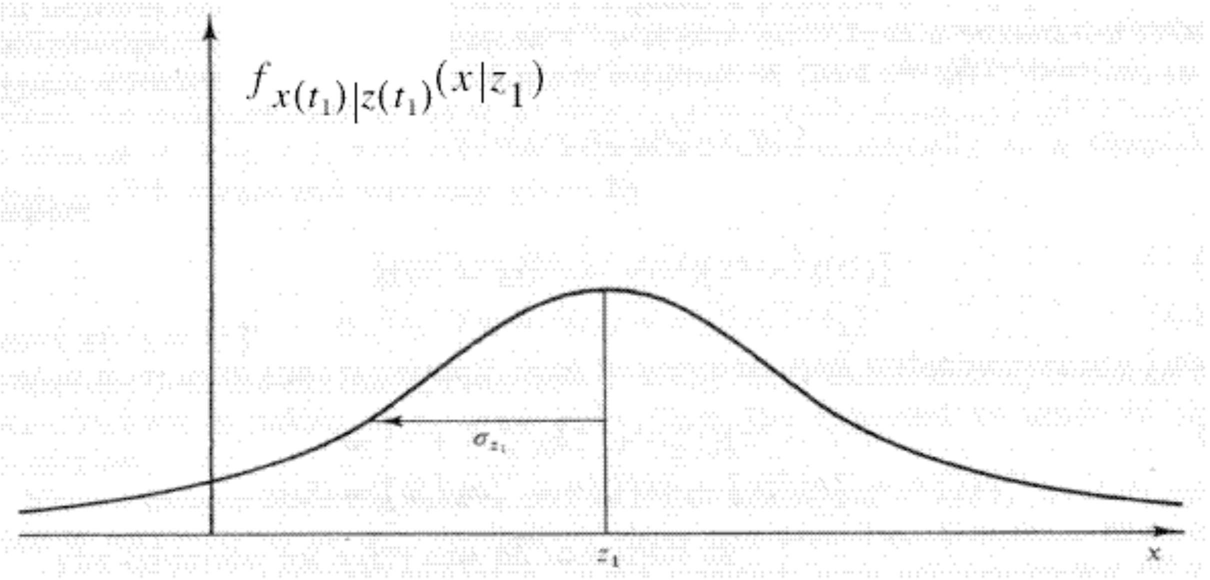
- The sum of many random variables
  - with the same mean, but
  - with arbitrary conditional density functions,

  converges to a Gaussian density function.

- If a model omits many small unmodeled effects, then the resulting error should converge to a Gaussian density function.

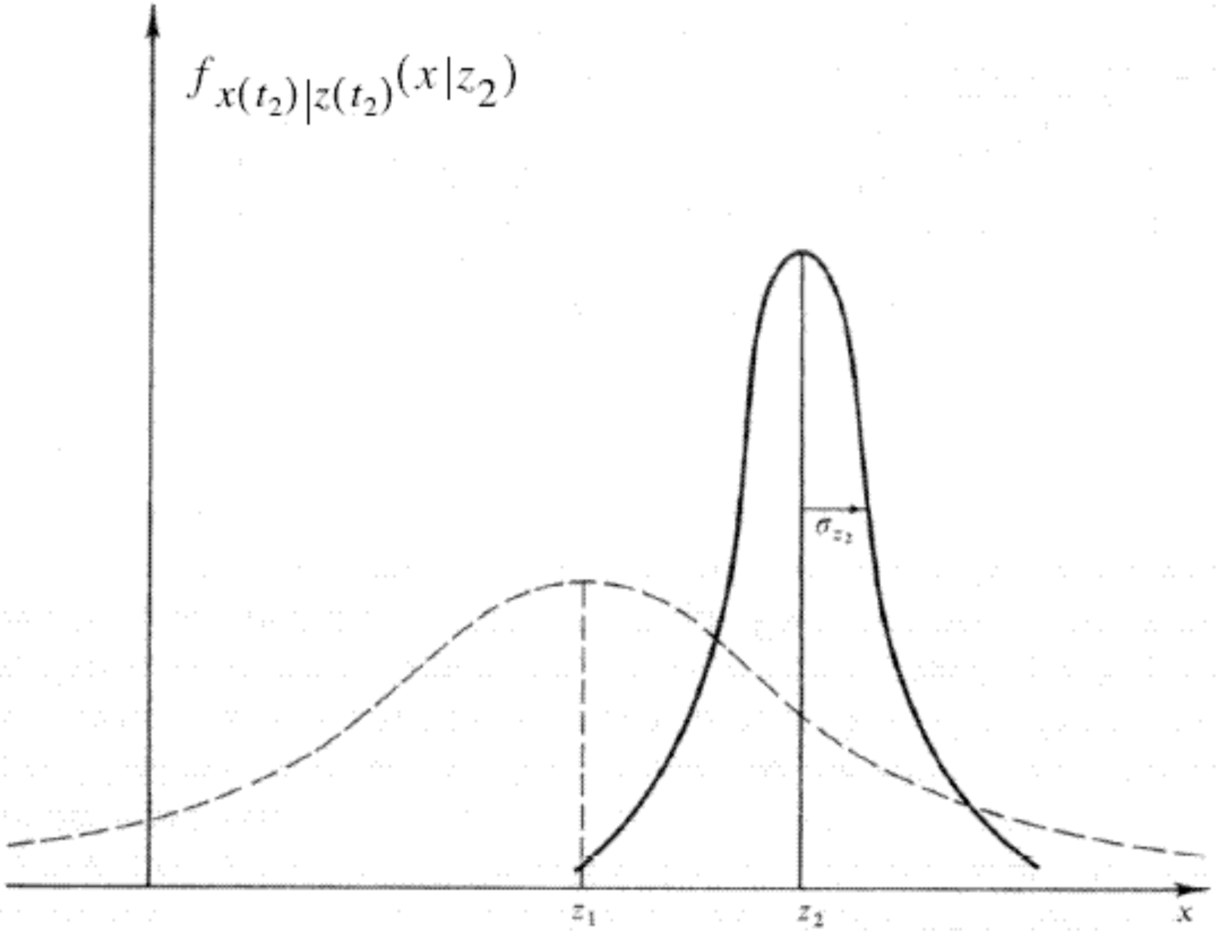
Estimating a Value

- Suppose there is a *constant* value *x*.
  - Distance to wall; angle to wall; etc.
- At time t_1, observe value z_1 with variance σ_1^2
- The optimal estimate is $\hat{x}(t_1) = z_1$ with variance σ_1^2

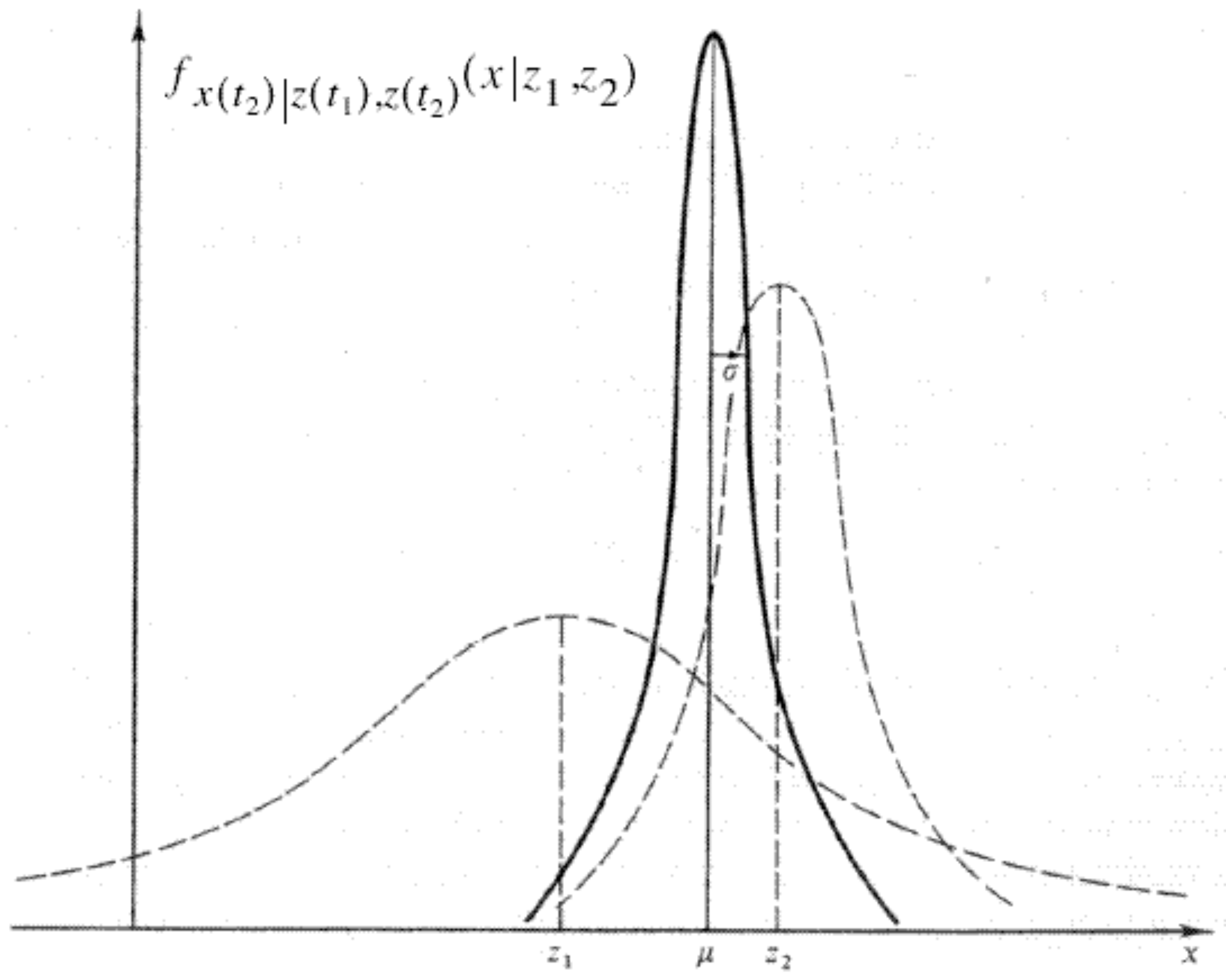


A Second Observation

- At time t_2, observe value z_2 with variance σ_2^2



Merged Evidence



Update Mean and Variance

- Weighted average of estimates.

$$\hat{x}(t_2) = Az_1 + Bz_2 \qquad A + B = 1$$

- The weights come from the variances.
  - Smaller variance = more certainty

$$\hat{x}(t_2) = \left[\frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right] z_1 + \left[\frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\right] z_2$$

$$\frac{1}{\sigma^2(t_2)} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}$$

From Weighted Average to Predictor-Corrector

- Weighted average:

$$\hat{x}(t_2) = Az_1 + Bz_2 = (1-K)z_1 + Kz_2$$

- Predictor-corrector:

$$\begin{aligned}\hat{x}(t_2) &= z_1 + K(z_2 - z_1)\\ &= \hat{x}(t_1) + K(z_2 - \hat{x}(t_1))\end{aligned}$$

- This version can be applied “recursively”.

Predictor-Corrector

- Update best estimate given new data

$$\hat{x}(t_2) = \hat{x}(t_1) + K(t_2)(z_2 - \hat{x}(t_1))$$

$$K(t_2) = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}$$

- Update variance:

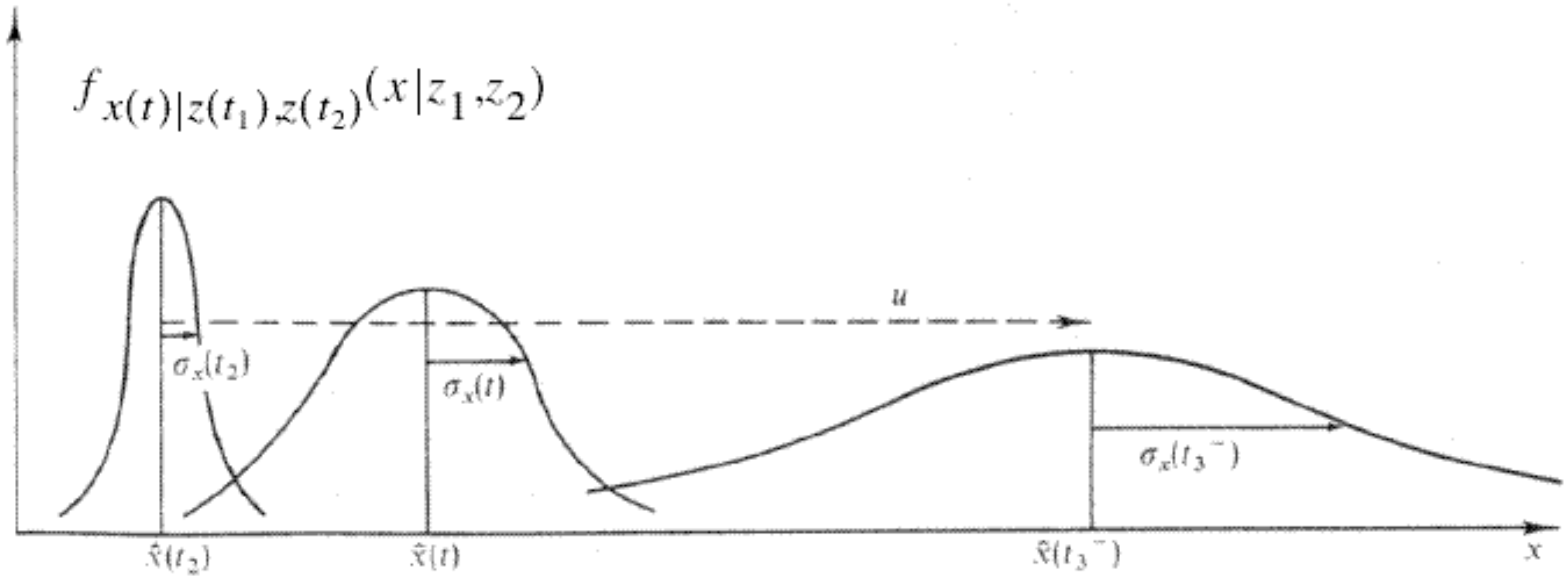
$$\sigma^2(t_2) = \sigma^2(t_1) - K(t_2)\sigma^2(t_1)$$

$$= (1 - K(t_2))\,\sigma^2(t_1)$$

Static to Dynamic

- Now suppose *x* changes according to

$$\dot{x} = F(x,u,\varepsilon) = u + \varepsilon \qquad (N(0,\sigma_\varepsilon))$$



Dynamic Prediction

- At t_2 we know $\hat{x}(t_2)$ $\sigma^2(t_2)$
- At t_3 after the change, before an observation.

$$\hat{x}(t_3^-) = \hat{x}(t_2) + u[t_3 - t_2]$$

$$\sigma^2(t_3^-) = \sigma^2(t_2) + \sigma_\varepsilon^2[t_3 - t_2]$$

- Next, we correct this prediction with the observation at time t_3.

Dynamic Correction

- At time t_3 we observe z_3 with variance σ_3^2
- Combine prediction with observation.

$$\hat{x}(t_3) = \hat{x}(t_3^-) + K(t_3)(z_3 - \hat{x}(t_3^-))$$

$$\sigma^2(t_3) = (1 - K(t_3))\sigma^2(t_3^-)$$

$$K(t_3) = \frac{\sigma^2(t_3^-)}{\sigma^2(t_3^-) + \sigma_3^2}$$

Qualitative Properties

$$\hat{x}(t_3) = \hat{x}(t_3^-) + K(t_3)(z_3 - \hat{x}(t_3^-))$$

$$K(t_3) = \frac{\sigma^2(t_3^-)}{\sigma^2(t_3^-) + \sigma_3^2}$$

- Suppose measurement noise σ_3^2 is large.
  - Then $K(t_3)$ approaches 0, and the measurement will be mostly ignored.
- Suppose prediction noise $\sigma^2(t_3^-)$ is large.
  - Then $K(t_3)$ approaches 1, and the measurement will dominate the estimate.

Kalman Filter

- Takes a stream of observations, and a dynamical model.
- At each step, a weighted average between
  - prediction from the dynamical model
  - correction from the observation.
- The Kalman gain $K(t)$ is the weighting,
  - based on the variances $\sigma^2(t)$ and σ_ε^2
- With time, $K(t)$ and $\sigma^2(t)$ tend to stabilize.

Simplifications

- We have only discussed a one-dimensional system.
  - Most applications are higher dimensional.
- We have assumed the state variable is observable.
  - In general, sense data give indirect evidence.

$$\dot{x} = F(x, u, \varepsilon_1) = u + \varepsilon_1$$
$$z = G(x, \varepsilon_2) = x + \varepsilon_2$$

- We will discuss the more complex case next.

Up To Higher Dimensions

- Our previous Kalman Filter discussion was of a simple one-dimensional model.
- Now we go up to higher dimensions:
  - State vector: $\mathbf{x} \in \Re^n$
  - Sense vector: $\mathbf{z} \in \Re^m$
  - Motor vector: $\mathbf{u} \in \Re^l$
- First, a little statistics.

Expectations

- Let *x* be a random variable.
- The expected value $E[x]$ is the mean:

$$E[x] = \int x\ p(x)\ dx \approx \bar{x} = \frac{1}{N}\sum_{1}^{N} x_i$$

  - The probability-weighted mean of all possible values. The sample mean approaches it.
- Expected value of a vector **x** is by component.

$$E[\mathbf{x}] = \bar{\mathbf{x}} = [\bar{x}_1, \cdots \bar{x}_n]^T$$

Variance and Covariance

- The variance is $E[\,(x\text{-}E[x])^2\,]$

$$\sigma^2 = E[(x-\bar{x})^2] = \frac{1}{N}\sum_{1}^{N}(x_i-\bar{x})^2$$

- Covariance matrix is $E[\,(\mathbf{x}\text{-}E[\mathbf{x}])(\mathbf{x}\text{-}E[\mathbf{x}])^T\,]$

$$C_{ij} = \frac{1}{N}\sum_{k=1}^{N}(x_{ik}-\bar{x}_i)(x_{jk}-\bar{x}_j)$$

  - Divide by $N-1$ to make the sample variance an *unbiased estimator* for the population variance.

Biased and Unbiased Estimators

- Strictly speaking, the sample variance

$$\sigma^2 = E[(x - \overline{x})^2] = \frac{1}{N}\sum_1^N (x_i - \overline{x})^2$$

is a biased estimate of the population variance. An unbiased estimator is:

$$s^2 = \frac{1}{N-1}\sum_1^N (x_i - \overline{x})^2$$

- **But**: *"If the difference between N and N−1 ever matters to you, then you are probably up to no good anyway …"* [Press, et al]

Covariance Matrix

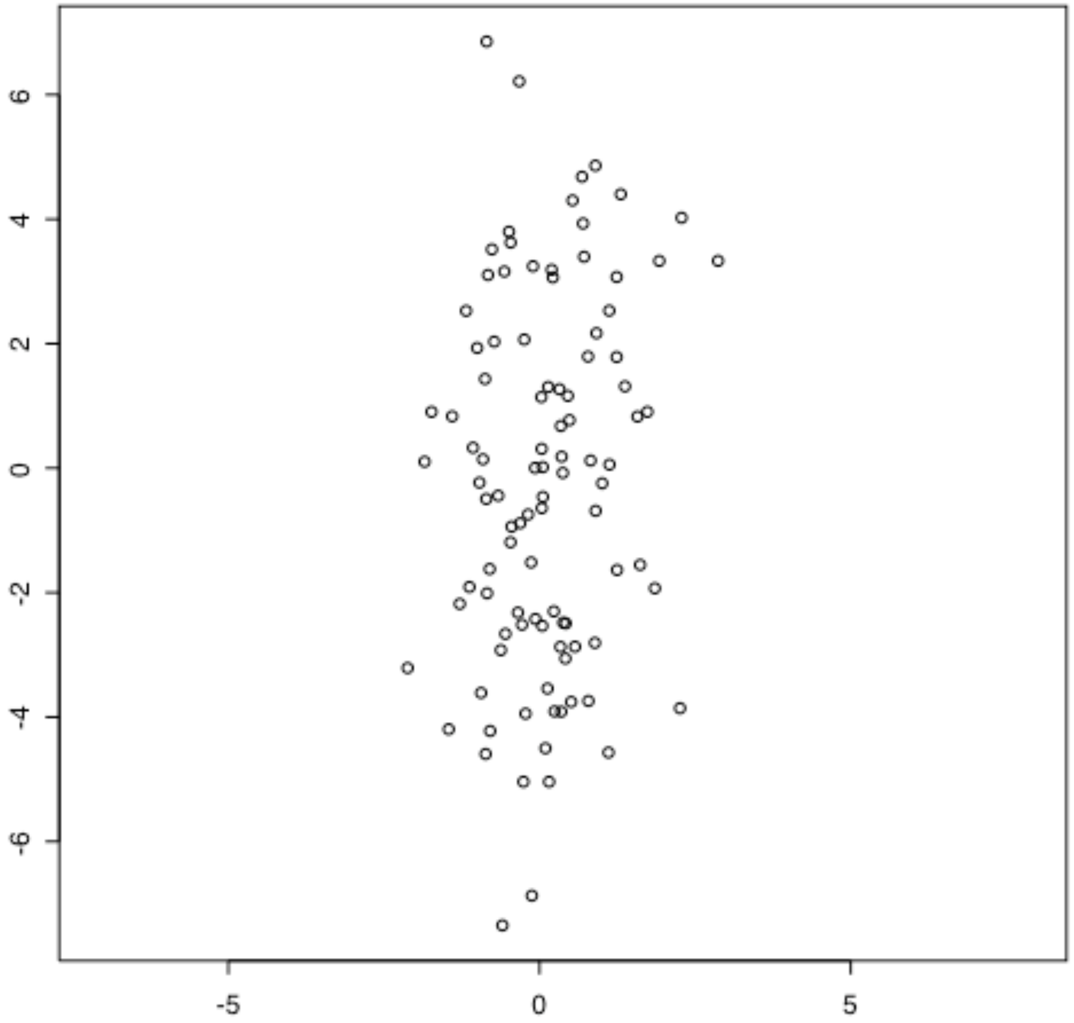
- Along the diagonal, C_{ii} are variances.
- Off-diagonal C_{ij} are essentially correlations.

$$\begin{bmatrix} C_{1,1} = \sigma_1^2 & C_{1,2} & & C_{1,N} \\ C_{2,1} & C_{2,2} = \sigma_2^2 & & \\ & & \ddots & \vdots \\ C_{N,1} & & \cdots & C_{N,N} = \sigma_N^2 \end{bmatrix}$$

Independent Variation

- x and y are Gaussian random variables (N=100)
- Generated with $\sigma_x=1 \quad \sigma_y=3$
- Covariance matrix:

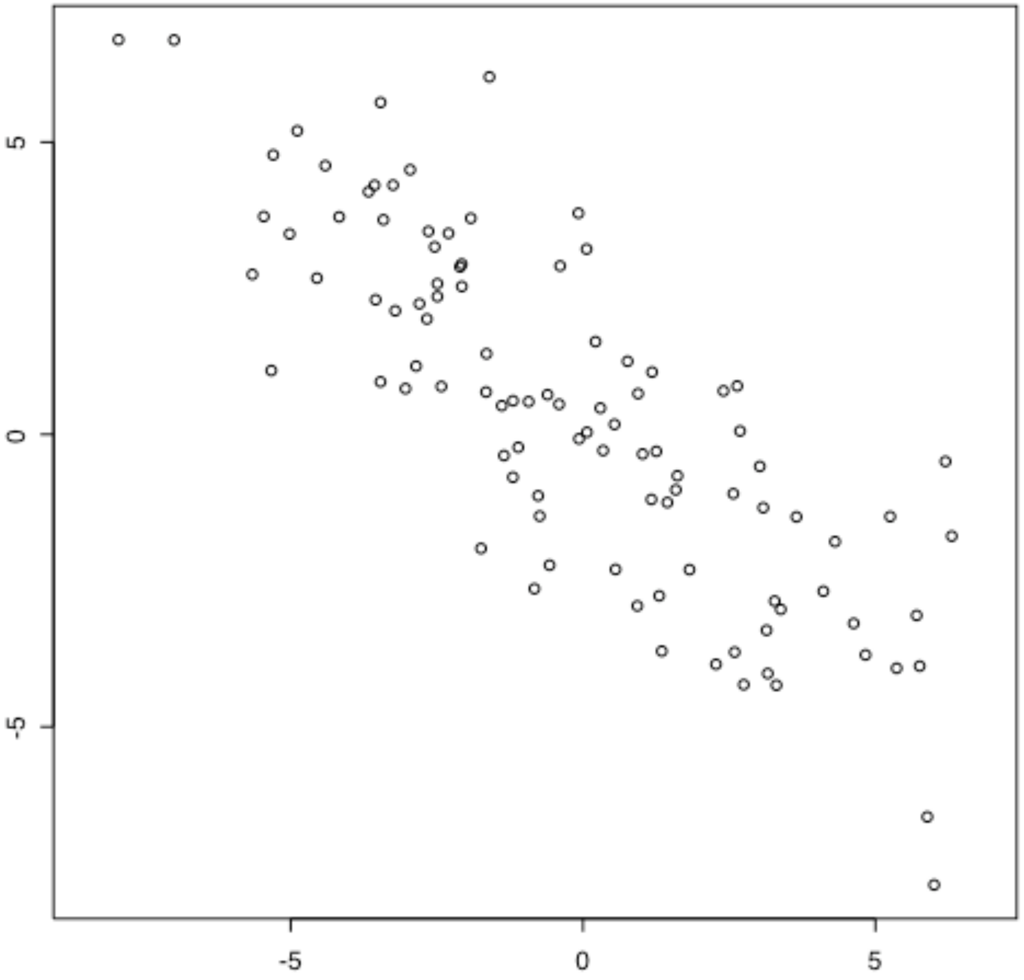
$$C_{xy} = \begin{bmatrix} 0.90 & 0.44 \\ 0.44 & 8.82 \end{bmatrix}$$



Dependent Variation

- *c* and *d* are random variables.
- Generated with $c=x+y \quad d=x-y$
- Covariance matrix:

$$C_{cd} = \begin{bmatrix} 10.62 & -7.93 \\ -7.93 & 8.84 \end{bmatrix}$$



Discrete Kalman Filter

- Estimate the state $\mathbf{x} \in \Re^n$ of a linear stochastic difference equation

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k + \mathbf{w}_{k-1}$$

  - process noise **w** is drawn from *N*(0,**Q**), with covariance matrix **Q**.

- with a measurement $\mathbf{z} \in \Re^m$

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k$$

  - measurement noise **v** is drawn from *N*(0,**R**), with covariance matrix **R**.

- **A, Q** are *n*x*n*. **B** is *n*x*l*. **R** is *m*x*m*. **H** is *m*x*n*.

Estimates and Errors

- $\hat{\mathbf{x}}_k \in \Re^n$ is the estimated state at time-step *k*.
- $\hat{\mathbf{x}}_k^- \in \Re^n$ after prediction, before observation.
- Errors:

$$\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^-$$

$$\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k$$

- Error covariance matrices:

$$\mathbf{P}_k^- = E[\mathbf{e}_k^- \mathbf{e}_k^{-T}]$$

$$\mathbf{P}_k = E[\mathbf{e}_k \mathbf{e}_k^T]$$

- Kalman Filter's task is to update $\hat{\mathbf{x}}_k \quad \mathbf{P}_k$

Time Update (Predictor)

- Update expected value of **x**

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_k$$

- Update error covariance matrix **P**

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}$$

- Previous statements were simplified versions of the same idea:

$$\hat{x}(t_3^-) = \hat{x}(t_2) + u[t_3 - t_2]$$

$$\sigma^2(t_3^-) = \sigma^2(t_2) + \sigma_\varepsilon^2[t_3 - t_2]$$

Measurement Update (Corrector)

- Update expected value

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-)$$

  - *innovation* is $\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-$
- Update error covariance matrix

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^-$$

- Compare with previous form

$$\hat{x}(t_3) = \hat{x}(t_3^-) + K(t_3)(z_3 - \hat{x}(t_3^-))$$

$$\sigma^2(t_3) = (1 - K(t_3))\sigma^2(t_3^-)$$

The Kalman Gain

- The optimal Kalman gain $\mathbf{K}_k$ is

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1}$$

$$= \frac{\mathbf{P}_k^- \mathbf{H}^T}{\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R}}$$

- Compare with previous form

$$K(t_3) = \frac{\sigma^2(t_3^-)}{\sigma^2(t_3^-) + \sigma_3^2}$$

Extended Kalman Filter

- Suppose the state-evolution and measurement equations are non-linear:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_{k-1}$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k$$

  - process noise **w** is drawn from *N*(0,**Q**), with covariance matrix **Q**.
  - measurement noise **v** is drawn from *N*(0,**R**), with covariance matrix **R**.

The Jacobian Matrix

- For a scalar function $y=f(x)$,

$$\Delta y = f'(x)\Delta x$$

- For a vector function $\mathbf{y}=f(\mathbf{x})$,

$$\Delta \mathbf{y} = \mathbf{J}\Delta \mathbf{x} = \begin{bmatrix} \Delta y_1 \\ \vdots \\ \Delta y_n \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}) \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial f_n}{\partial x_n}(\mathbf{x}) \end{bmatrix} \cdot \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_n \end{bmatrix}$$

Linearize the Non-Linear

- Let **A** be the Jacobian of *f* with respect to **x**.

$$\mathbf{A}_{ij} = \frac{\partial f_i}{\partial x_j}(\mathbf{x}_{k-1}, \mathbf{u}_k)$$

- Let **H** be the Jacobian of *h* with respect to **x**.

$$\mathbf{H}_{ij} = \frac{\partial h_i}{\partial x_j}(\mathbf{x}_k)$$

- Then the Kalman Filter equations are almost the same as before!

EKF Update Equations

- Predictor step: $\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k)$

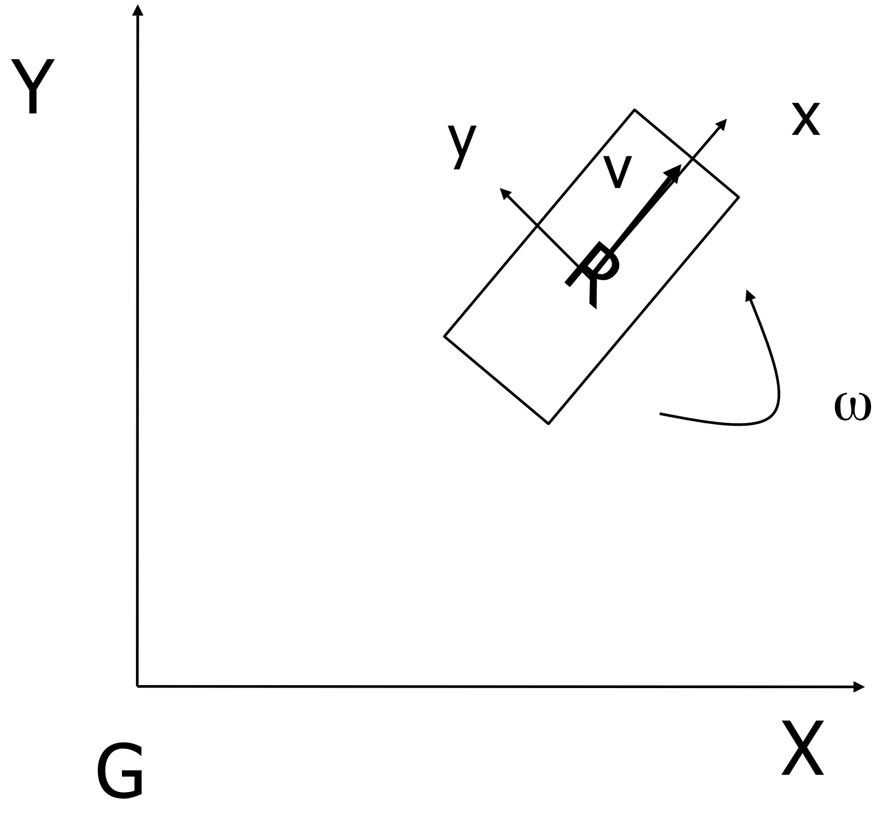
$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}$$

- Kalman gain: $\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1}$

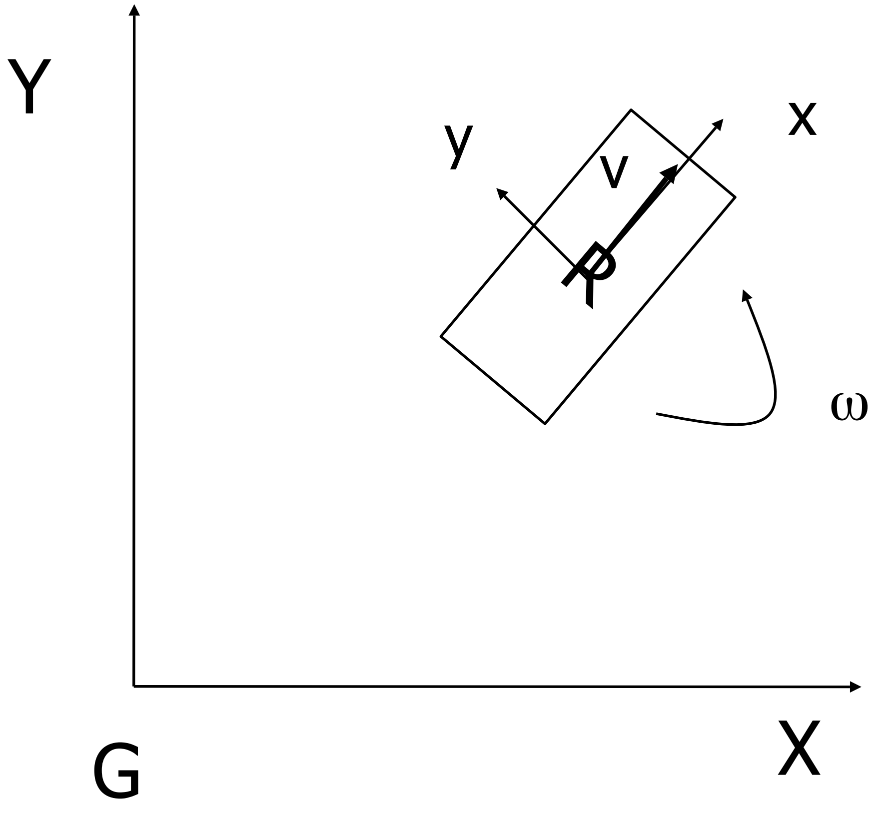
- Corrector step: $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - h(\hat{\mathbf{x}}_k^-))$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^-$$

Linearized Motion Model for a Robot



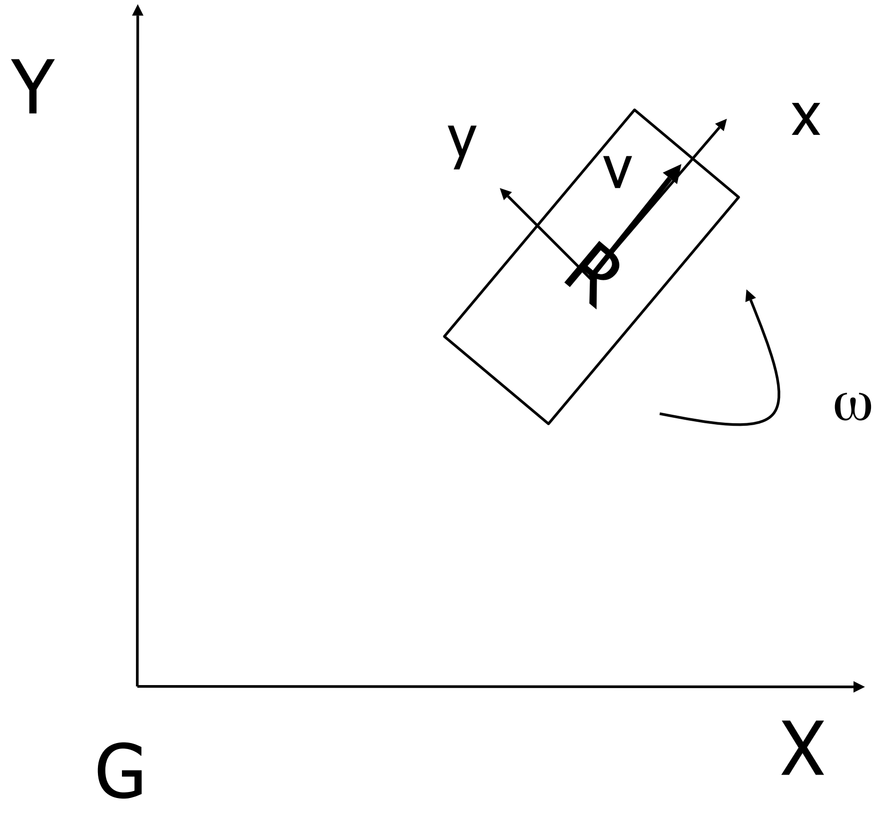
Linearized Motion Model for a Robot



From a robot-centric perspective, the velocities look like this:

$$\dot{x}_t = V_t$$
$$\dot{y}_t = 0$$
$$\dot{\phi}_t = \omega_t$$

Linearized Motion Model for a Robot



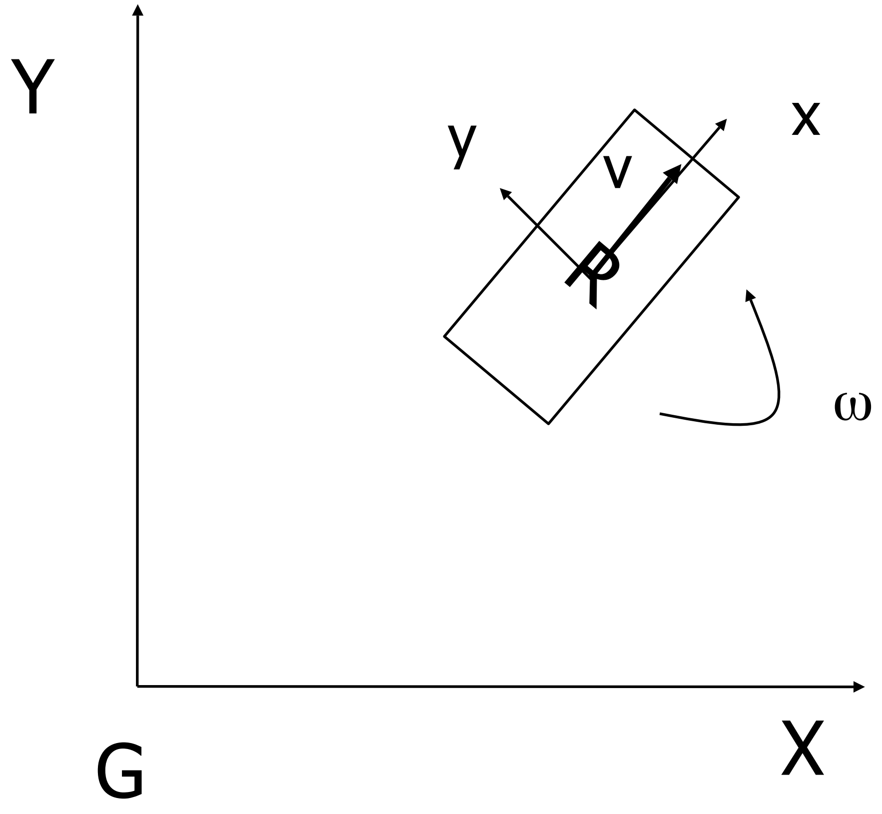
From a robot-centric perspective, the velocities look like this:

$$\dot{x}_t = V_t$$
$$\dot{y}_t = 0$$
$$\dot{\phi}_t = \omega_t$$

From the global perspective, the velocities look like this:

$$\dot{x}_t = V_t \cos\phi_t$$
$$\dot{y}_t = V_t \sin\phi_t$$
$$\dot{\phi}_t = \omega_t$$

Linearized Motion Model for a Robot



From a robot-centric perspective, the velocities look like this:

$$\dot{x}_t = V_t$$
$$\dot{y}_t = 0$$
$$\dot{\phi}_t = \omega_t$$

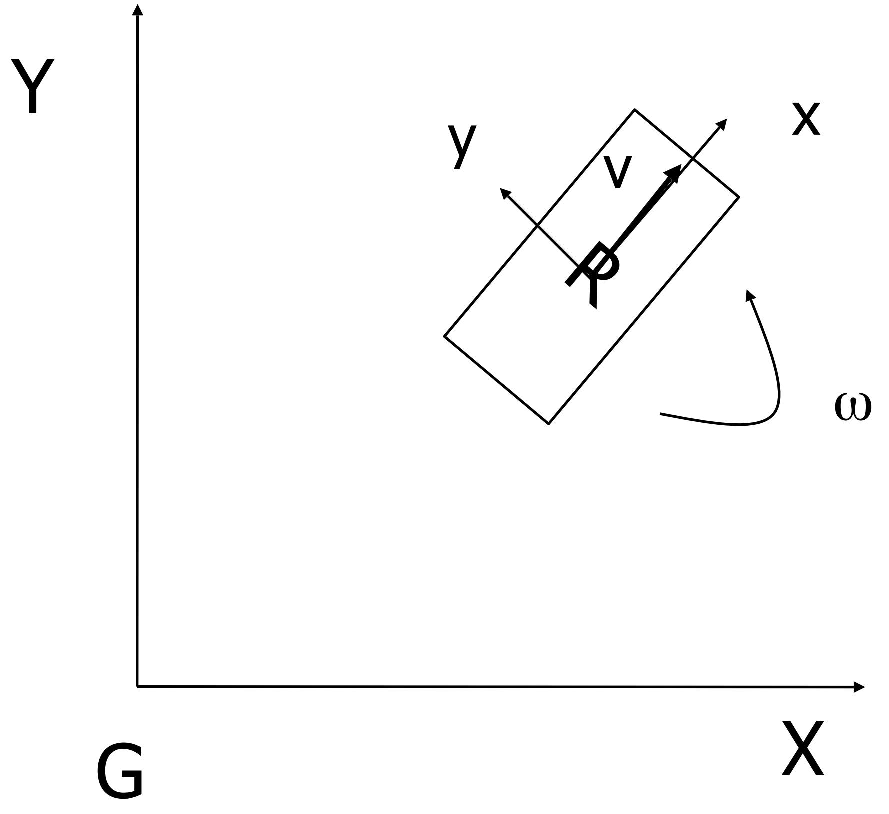
From the global perspective, the velocities look like this:

$$\dot{x}_t = V_t \cos\phi_t$$
$$\dot{y}_t = V_t \sin\phi_t$$
$$\dot{\phi}_t = \omega_t$$

The discrete time state estimate (including noise) looks like this:

$$\hat{x}_{t+1} = \hat{x}_t + (V_t + w_{V_t})\delta t \cos\hat{\phi}_t$$
$$\hat{y}_{t+1} = \hat{y}_t + (V_t + w_{V_t})\delta t \sin\hat{\phi}_t$$
$$\hat{\phi}_{t+1} = \hat{\phi}_t + (\omega_t + w_{\omega_t})\delta t$$

Linearized Motion Model for a Robot



From a robot-centric perspective, the velocities look like this:

$$\dot{x}_t = V_t$$
$$\dot{y}_t = 0$$
$$\dot{\phi}_t = \omega_t$$

From the global perspective, the velocities look like this:

$$\dot{x}_t = V_t \cos\phi_t$$
$$\dot{y}_t = V_t \sin\phi_t$$
$$\dot{\phi}_t = \omega_t$$

The discrete time state estimate (including noise) looks like this:

$$\hat{x}_{t+1} = \hat{x}_t + (V_t + w_{V_t})\delta t \cos\hat{\phi}_t$$
$$\hat{y}_{t+1} = \hat{y}_t + (V_t + w_{V_t})\delta t \sin\hat{\phi}_t$$
$$\hat{\phi}_{t+1} = \hat{\phi}_t + (\omega_t + w_{\omega_t})\delta t$$

Problem! We don't know linear and rotational velocity errors. The state estimate will rapidly diverge if this is the only source of information!

Linearized Motion Model for a Robot

Now, we have to compute the covariance matrix propagation equations.

Linearized Motion Model for a Robot

Now, we have to compute the covariance matrix propagation equations.

The indirect Kalman filter derives the pose equations from the estimated error of the state:

$$x_{t+1} - \hat{x}_{t+1} = \tilde{x}_{t+1}$$

$$y_{t+1} - \hat{y}_{t+1} = \tilde{y}_{t+1}$$

$$\phi_{t+1} - \hat{\phi}_{t+1} = \tilde{\phi}_{t+1}$$

Linearized Motion Model for a Robot

Now, we have to compute the covariance matrix propagation equations.

The indirect Kalman filter derives the pose equations from the estimated error of the state:

$$
\begin{aligned}
x_{t+1} - \hat{x}_{t+1} &= \tilde{x}_{t+1} \\
y_{t+1} - \hat{y}_{t+1} &= \tilde{y}_{t+1} \\
\phi_{t+1} - \hat{\phi}_{t+1} &= \tilde{\phi}_{t+1}
\end{aligned}
$$

In order to linearize the system, the following small-angle assumptions are made:

$$
\begin{aligned}
\cos\tilde{\phi} &\cong 1 \\
\sin\tilde{\phi} &\cong \tilde{\phi}
\end{aligned}
$$

Linearized Motion Model for a Robot

From the error-state propagation equation, we can obtain the State propagation and noise input functions F and G :

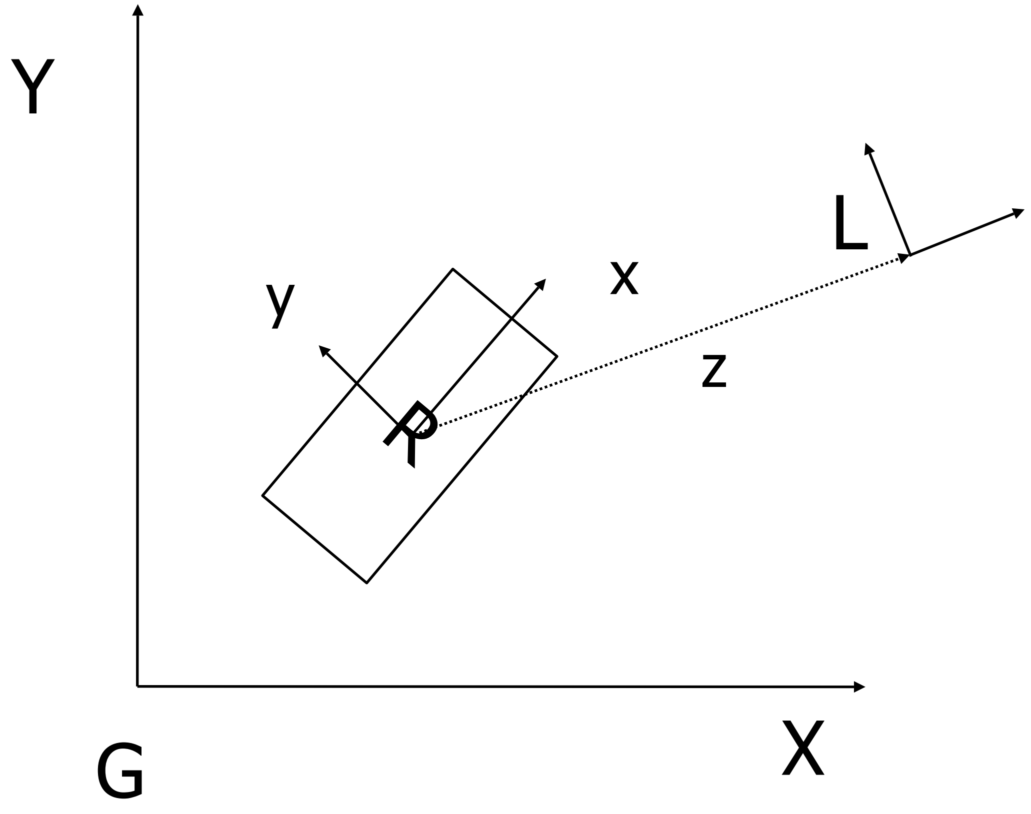
$$\begin{bmatrix} \tilde{x}_{t+1} \\ \tilde{y}_{t+1} \\ \tilde{\phi}_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -V_m \delta t \sin\hat{\phi} \\ 0 & 1 & V_m \delta t \cos\hat{\phi} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x}_t \\ \tilde{y}_t \\ \tilde{\phi}_t \end{bmatrix} + \begin{bmatrix} -\delta t \cos\phi_R & 0 \\ -\delta t \sin\phi_R & 0 \\ 0 & -\delta t \end{bmatrix} \begin{bmatrix} w_{V_t} \\ w_{\omega_t} \end{bmatrix}$$

$$\tilde{X}_{t+1} = F_t \tilde{X}_t + G_t W_t$$

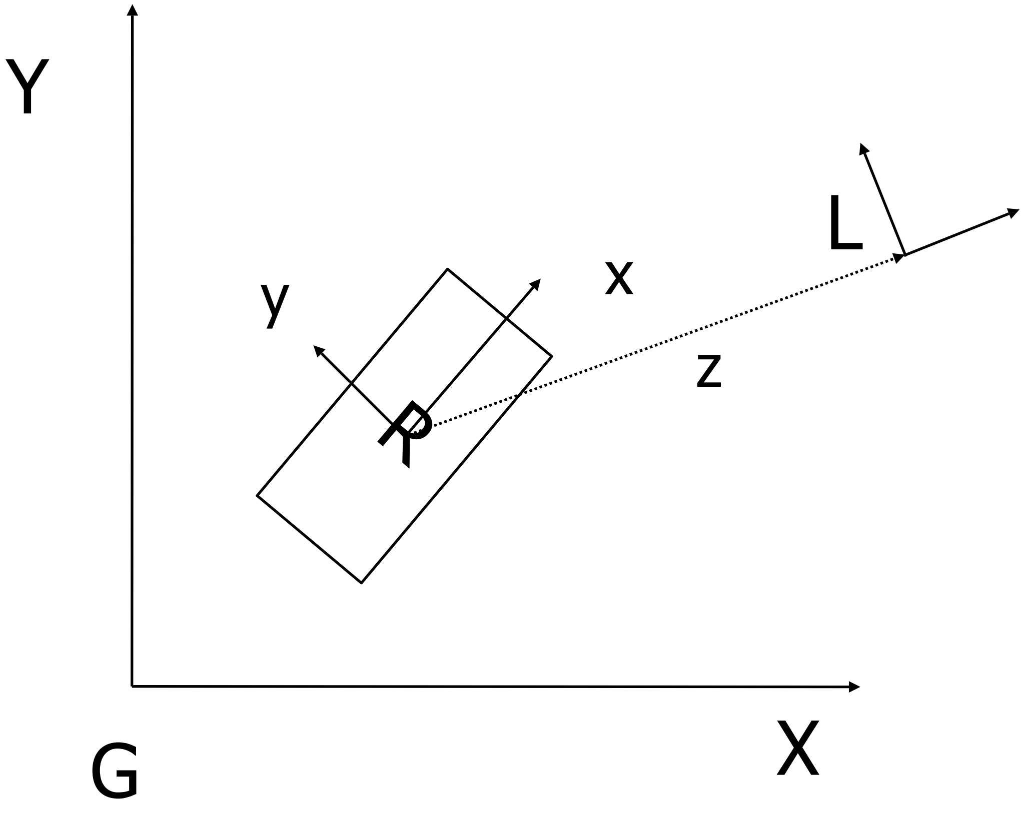
From these values, we can easily compute the standard covariance propagation equation:

$$P_{t+1/t} = F_t P_{t/t} F_t^T + G_t Q_t G_t^T$$

Sensor Model for a Robot with a Perfect Map



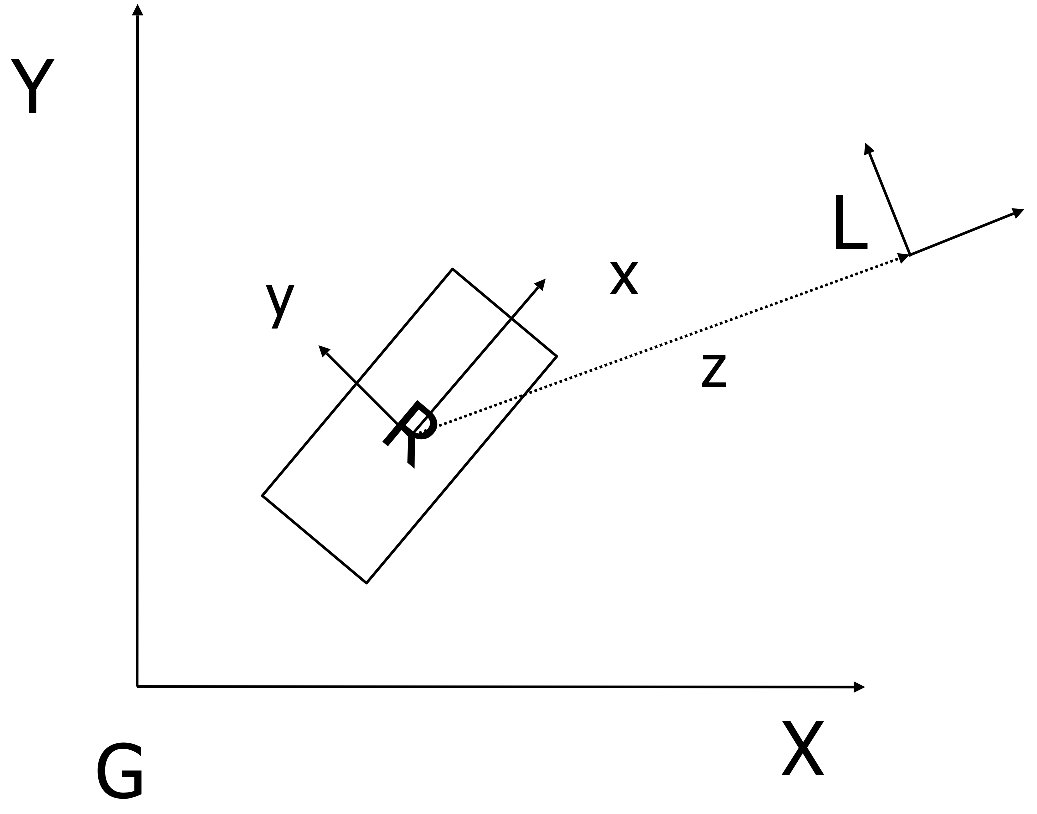
Sensor Model for a Robot with a Perfect Map



From the robot, the measurement looks like this:

$$z_{t+1} = \begin{bmatrix} x_{L_{t+1}} \\ y_{L_{t+1}} \\ \phi_{L_{t+1}} \end{bmatrix} + \begin{bmatrix} n_x \\ n_y \\ n_\phi \end{bmatrix}$$

Sensor Model for a Robot with a Perfect Map



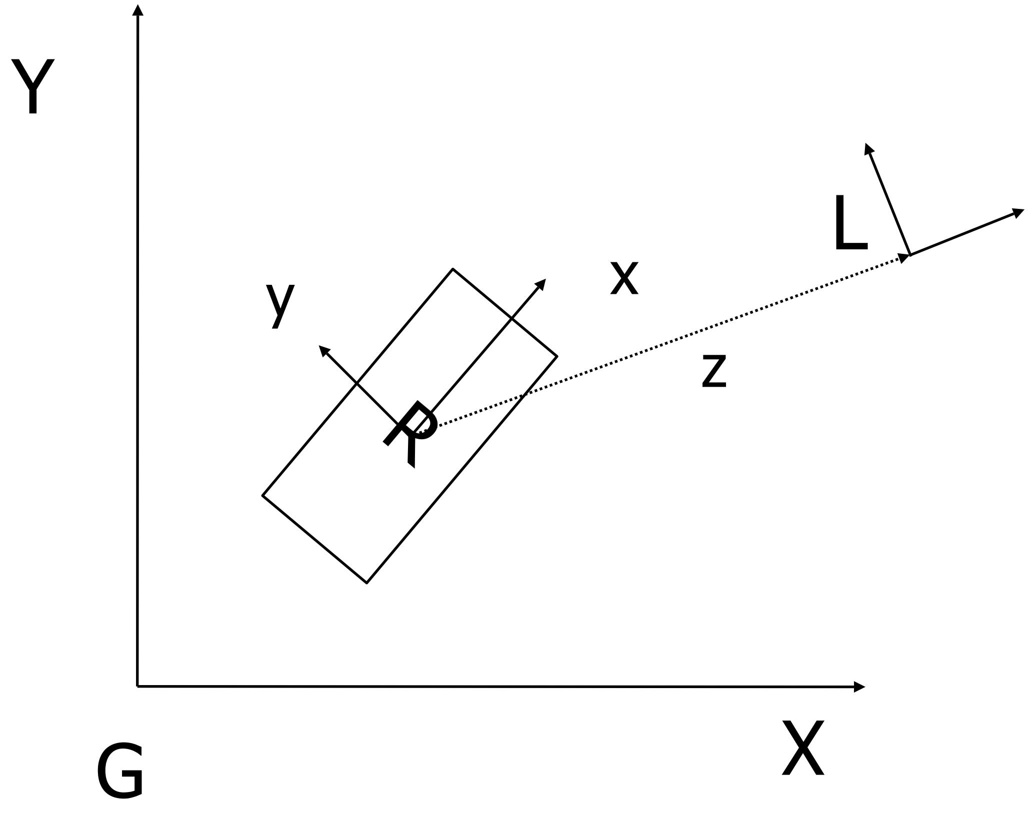
From the robot, the measurement looks like this:

$$z_{t+1} = \begin{bmatrix} x_{L_{t+1}} \\ y_{L_{t+1}} \\ \phi_{L_{t+1}} \end{bmatrix} + \begin{bmatrix} n_x \\ n_y \\ n_\phi \end{bmatrix}$$

From a global perspective, the measurement looks like:

$$z_{t+1} = \begin{bmatrix} \cos\phi_{t+1} & -\sin\phi_{t+1} & 0 \\ \sin\phi_{t+1} & \cos\phi_{t+1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{L_{t+1}} - x_{t+1} \\ y_{L_{t+1}} - y_{t+1} \\ \phi_{L_{t+1}} - \phi_{t+1} \end{bmatrix} + \begin{bmatrix} n_x \\ n_y \\ n_\phi \end{bmatrix}$$

Sensor Model for a Robot with a Perfect Map



From the robot, the measurement looks like this:

$$z_{t+1} = \begin{bmatrix} x_{L_{t+1}} \\ y_{L_{t+1}} \\ \phi_{L_{t+1}} \end{bmatrix} + \begin{bmatrix} n_x \\ n_y \\ n_\phi \end{bmatrix}$$

From a global perspective, the measurement looks like:

$$z_{t+1} = \begin{bmatrix} \cos\phi_{t+1} & -\sin\phi_{t+1} & 0 \\ \sin\phi_{t+1} & \cos\phi_{t+1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{L_{t+1}} - x_{t+1} \\ y_{L_{t+1}} - y_{t+1} \\ \phi_{L_{t+1}} - \phi_{t+1} \end{bmatrix} + \begin{bmatrix} n_x \\ n_y \\ n_\phi \end{bmatrix}$$

The measurement equation is nonlinear and must also be linearized!

Sensor Model for a Robot with a Perfect Map

Now, we have to compute the linearized sensor function. Once again, we make use of the indirect Kalman filter where the error in the reading must be estimated.

Sensor Model for a Robot with a Perfect Map

Now, we have to compute the linearized sensor function. Once again, we make use of the indirect Kalman filter where the error in the reading must be estimated.

In order to linearize the system, the following small-angle assumptions are made:

$$\cos\tilde{\phi} \cong 1$$

$$\sin\tilde{\phi} \cong \tilde{\phi}$$

Sensor Model for a Robot with a Perfect Map

Now, we have to compute the linearized sensor function. Once again, we make use of the indirect Kalman filter where the error in the reading must be estimated.

In order to linearize the system, the following small-angle assumptions are made:

$$\cos\tilde{\phi} \cong 1$$
$$\sin\tilde{\phi} \cong \tilde{\phi}$$

The final expression for the error in the sensor reading is:

$$\begin{bmatrix} \tilde{x}_{L_{t+1}} \\ \tilde{y}_{L_{t+1}} \\ \tilde{\phi}_{L_{t+1}} \end{bmatrix} = \begin{bmatrix} -\cos\hat{\phi}_{t+1} & -\sin\hat{\phi}_{t+1} & -\sin\hat{\phi}_{t+1}(x_L - \hat{x}_{t+1}) + \cos\hat{\phi}_t(y_L - \hat{y}_{t+1}) \\ \sin\hat{\phi}_{t+1} & -\cos\hat{\phi}_{t+1} & -\cos\hat{\phi}_{t+1}(x_L - \hat{x}_{t+1}) - \sin\hat{\phi}_t(y_L - \hat{y}_{t+1}) \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \tilde{x}_{t+1} \\ \tilde{y}_{t+1} \\ \tilde{\phi}_{t+1} \end{bmatrix} + \begin{bmatrix} n_x \\ n_y \\ n_\phi \end{bmatrix}$$

Slides

- Slides were taken from:
    - http://www.cs.utexas.edu/~pstone/Courses/395Tfall05/resources/
    - www.cs.cmu.edu/~robosoccer/cmrobobits/lectures/**Kalman**.ppt