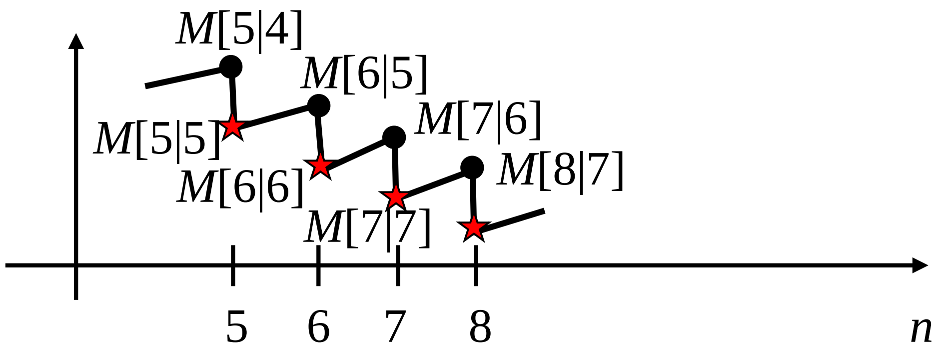
<u>Important Properties of the KF</u>

1. Kalman filter is an extension of the sequential MMSE estimator
   - Sequential MMSE is for a fixed parameter
   - Kalman is for time-varying parameter, but <u>must have a known dynamical model</u>
   - Block diagrams are nearly identical except for the $\mathbf{A}z^{-1}$ feedback box in the Kalman filter… just a z^{-1} box in seq. MMSE… the $\mathbf{A}$ is the dynamical model's state-transition matrix
2. Inversion is only needed for the vector observation case
3. Kalman filter is a time-varying filter
   - Due to two time-varying blocks: gain $\mathbf{K}[n]$ & Observation Matrix $\mathbf{H}[n]$
   - Note: $\mathbf{K}[n]$ changes constantly to adjust the balance between "info from the data" (the innovation) vs. "info from the model" (the prediction)
4. Kalman filter computes (and uses!) its own performance measure $\mathbf{M}[n|n]$ (which is the MMSE matrix)
   - Used to help balance between innovation and prediction

5. There is a natural up-down progression in the error

- The Prediction Stage increases the error
- The Update Stage decreases the error $M[n|n-1] > M[n|n]$
- This is OK… prediction is just a natural, intermediate step in the Optimal processing



6. Prediction is an integral part of the KF

- And it is based entirely on the Dynamical Model!!!

7. After a "long" time (as $n \to \infty$) the KF reaches "steady-state" operation… and the KF becomes a Linear Time-Invariant filter

- $M[n|n]$ and $M[n|n-1]$ both become constant
- … but still have $M[n|n-1] > M[n|n]$
- Thus, the gain $k[n]$ becomes constant, too.

8. The KF creates an uncorrelated sequence… the innovations.
   - Can view the innovations as “an equivalent input sequence”
   - Or… if we view the innovations as the output, then the steady-state KF is a LTI whitening filter (need state-state to get constant-power innovations)

9. The KF is optimal for the Gaussian Case (minimizes MSE)
   - If not Gaussian… the KF is still the optimal Linear MMSE estimator!!!

10. $\mathbf{M}[n|n-1]$, $\mathbf{M}[n|n]$, and $\mathbf{K}[n]$ can be computed ahead of time (“off-line”)
   - As long as the expected measurement variance σ^2_n is known
   - This allows off-line data-independent assessment of KF performance

13.5 Kalman Filters vs. Wiener Filters

They are hard to directly compare… They have <u>different models</u>

- <u>Wiener</u> assumes <u>WSS signal</u> + Noise
- <u>Kalman</u> assumes <u>Dynamical Model</u> w/ <u>Observation Model</u>

So… to compare we need to put them in the same context:

If we let:

1. Consider only after much time has elapsed (as $n \to \infty$)
   - Gives IIR Wiener case
   - Gives steady-state Kalman & Dynamic model becomes AR
2. For Kalman Filter, let σ^2_n be constant
   - Observation noise becomes WSS

Then… Kalman = Wiener!!! See book for more details

13.7 Extended Kalman Filter

The dynamical and observation models we assumed when developing the Kalman filter were **Linear** models:

Dynamics: $$\boxed{\mathbf{s}[n] = \mathbf{A}\mathbf{s}[n-1] + \mathbf{B}\mathbf{u}[n]}$$

Observations: $$\boxed{\mathbf{x}[n] = \mathbf{H}[n]\mathbf{s}[n] + \mathbf{w}[n]}$$

(A matrix is a linear operator)

However, many (most?) applications have a

- Nonlinear State Equation

  and/or

- Nonlinear Observation Equation

Solving for the Optimal Kalman filter for the nonlinear model case is generally intractable!!!

The "Extended Kalman Filter" is a sub-optimal approach that linearizes the model(s) and then applies the standard KF

EKF Motivation: A/C Tracking with Radar

Case #1: Dynamics are Linear but Observations are Nonlinear

Recall the constant-velocity model for an aircraft:

Define the state in rectangular coordinates:

$$\mathbf{s}[n] = \begin{bmatrix} r_x[n] \\ r_y[n] \\ v_x[n] \\ v_y[n] \end{bmatrix} \begin{array}{l} \left.\begin{array}{l} \\ \\ \end{array}\right\} \text{A/C positions (m)} \\ \left.\begin{array}{l} \\ \\ \end{array}\right\} \text{A/C velocities (m/s)} \end{array}$$

Dynamics Model

$$\boxed{\mathbf{s}[n] = \mathbf{A}\mathbf{s}[n-1] + \mathbf{B}\mathbf{u}[n]}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

For rectangular coordinates the state equation is linear

But… the choice of rectangular coordinates makes the radar's observations nonlinearly related to the state:

A radar can observe range and bearing (i.e., angle to target)

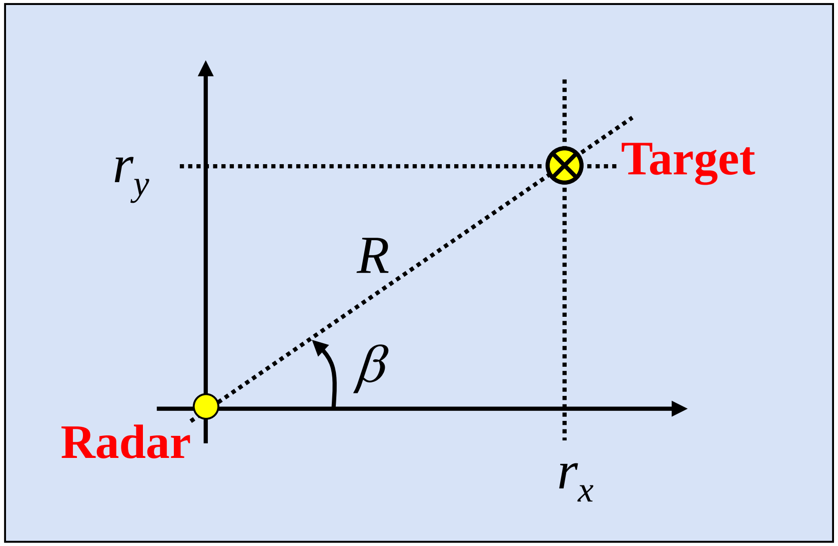
(and radial and angular velocities, which we will ignore here)

So the observations equations – relating the observation to the state – are given by:

For rectangular state coordinates the observation equation is <u>Non-Linear</u>

Observation Model

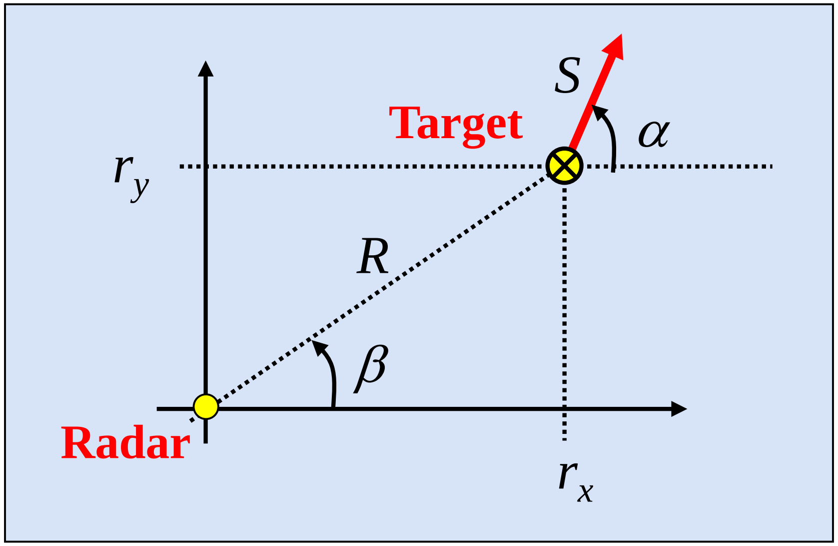
$$\mathbf{x}[n] = \begin{bmatrix} \sqrt{r_x^2[n] + r_y^2[n]} \\ \tan^{-1}\left[\dfrac{r_y[n]}{r_x[n]}\right] \end{bmatrix} + \begin{bmatrix} w_R[n] \\ w_\beta[n] \end{bmatrix}$$



Case #2: Observations are Linear but Dynamics are Nonlinear

If we choose the state to be in polar form then the observations will be linear functions of the state… so maybe then we won't have a problem??? **WRONG!!!**

$$\mathbf{s}[n] = \begin{bmatrix} R[n] \\ \beta[n] \\ S[n] \\ \alpha[n] \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} R \\ \beta \end{matrix}}\right\} \text{A/C Range \& Bearing} \\ \left.\vphantom{\begin{matrix} S \\ \alpha \end{matrix}}\right\} \text{A/C Speed \& Heading} \end{matrix}$$



Observation Model

The observation is linear:

$$\mathbf{x}[n] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} R[n] \\ \beta[n] \\ S[n] \\ \alpha[n] \end{bmatrix} + \begin{bmatrix} w_R[n] \\ w_\beta[n] \end{bmatrix}$$

But… The Dynamics Model is now <u>Non</u>-Linear:

$$\mathbf{s}[n] = \begin{bmatrix} R[n] \\ \beta[n] \\ S[n] \\ \alpha[n] \end{bmatrix} = \begin{bmatrix} \sqrt{\left(R[n-1]\cos(\beta[n-1]) + \Delta S[n-1]\cos(\alpha[n-1])\right)^2 + \left(R[n-1]\sin(\beta[n-1]) + \Delta S[n-1]\sin(\alpha[n-1])\right)^2} \\ \tan^{-1}\left[\left(R[n-1]\sin(\beta[n-1]) + \Delta S[n-1]\sin(\alpha[n-1])\right) / \left(R[n-1]\cos(\beta[n-1]) + \Delta S[n-1]\cos(\alpha[n-1])\right)\right] \\ \sqrt{\left(S[n-1]\cos(\alpha[n-1]) + u_x[n]\right)^2 + \left(S[n-1]\sin(\alpha[n-1]) + u_y[n]\right)^2} \\ \tan^{-1}\left[\left(S[n-1]\sin(\alpha[n-1]) + u_y[n]\right) / \left(S[n-1]\cos(\alpha[n-1]) + u_x[n]\right)\right] \end{bmatrix}$$

In each of these cases…

We can't apply the standard KF because it relies on the assumption of <u>linear</u> state and observation models!!!

Nonlinear Models

We state here the case where both the state and observation equations are nonlinear…

$$\boxed{\mathbf{s}[n] = \mathbf{a}(\mathbf{s}[n-1]) + \mathbf{B}\mathbf{u}[n]}$$

$$\boxed{\mathbf{x}[n] = \mathbf{h}_n(\mathbf{s}[n]) + \mathbf{w}[n]}$$

where $\mathbf{a}(.)$ and $\mathbf{h}_n(.)$ are both <u>non</u>linear functions mapping a vector to a vector

<u>What To Do When Facing a Non-Linear Model?</u>

1. Go back and re-derive the MMSE estimator for the the nonlinear case to develop the “your-last-name-here filter”??
   - Nonlinearities don’t preserve Gaussian so it will be hard to derive…
   - There *has* been some recent progress in this area: “particle filters”
2. Give up and try to convince your company’s executives and the FAA (Federal Aviation Administration) that tracking airplanes is not that important??
   - Probably not a good career move!!! ☺
3. Argue that you should use an <u>extremely dense</u> grid of radars networked together??
   - Would be extremely expensive… although with today’s efforts in sensor networks this may not be so far-fetched!!!
4. Linearize each nonlinear model using a 1^{st} order Taylor series?
   - Yes!!!
   - Of course, it won’t be optimal… but it might give the <u>required</u> performance!

Linearization of Models

State:

$$\mathbf{a}(\mathbf{s}[n-1]) \approx \mathbf{a}(\hat{\mathbf{s}}[n-1|n-1]) + \underbrace{\left[\left. \frac{\partial \mathbf{a}}{\partial \mathbf{s}[n-1]} \right|_{\mathbf{s}[n-1]=\hat{\mathbf{s}}[n-1|n-1]} \right]}_{\triangleq \mathbf{A}[n-1]} \left(\mathbf{s}[n-1] - \hat{\mathbf{s}}[n-1|n-1] \right)$$

Observation:

$$\mathbf{h}_n(\mathbf{s}[n-1]) \approx \mathbf{h}_n(\hat{\mathbf{s}}[n|n-1]) + \underbrace{\left[\left. \frac{\partial \mathbf{h}_n}{\partial \mathbf{s}[n]} \right|_{\mathbf{s}[n]=\hat{\mathbf{s}}[n|n-1]} \right]}_{\triangleq \mathbf{H}[n]} \left(\mathbf{s}[n] - \hat{\mathbf{s}}[n|n-1] \right)$$

Error: should be $\mathbf{s}[n]$

Using the Linearized Models

$$\mathbf{s}[n] = \mathbf{A}[n-1]\mathbf{s}[n-1] + \mathbf{B}\mathbf{u}[n] + \left[\mathbf{a}(\hat{\mathbf{s}}[n-1|n-1]) - \mathbf{A}[n-1]\hat{\mathbf{s}}[n-1|n-1]\right]$$

Just like what we did in the linear case except now have a time-varying A matrix

New additive term… But it is *known* at each step. So… in terms of *development* we can imagine that we just subtract off this known part… ⇒ Result: This part has no real impact!

$$\mathbf{x}[n] = \mathbf{H}[n]\mathbf{s}[n] + \mathbf{w}[n] + \left[\mathbf{h}_n(\hat{\mathbf{s}}[n|n-1]) - \mathbf{H}[n]\hat{\mathbf{s}}[n|n-1]\right]$$

1. Resulting EKF iteration is virtually the same – except there is a "linearizations" step
2. We no longer can do data-free, off-line performance iteration
   - **H**[*n*] and **A**[*n*-1] are computed on each iteration using the data-dependent estimate and prediction

Extended Kalman Filter (Vector-Vector)

Initialization: $\hat{\mathbf{s}}[-1\,|\,-1] = \boldsymbol{\mu}_s \qquad \mathbf{M}[-1\,|\,-1] = \mathbf{C}_s$

Prediction: $\hat{\mathbf{s}}[n\,|\,n-1] = \mathbf{a}\left(\hat{\mathbf{s}}[n-1\,|\,n-1]\right)$

Linearizations : $\mathbf{A}[n-1] = \left[\dfrac{\partial \mathbf{a}}{\partial \mathbf{s}(n-1)}\right]_{\mathbf{s}(n-1)=\hat{\mathbf{s}}(n-1|n-1)}$ $\qquad \mathbf{H}[n] = \left[\dfrac{\partial \mathbf{h}_n}{\partial \mathbf{s}(n)}\right]_{\mathbf{s}(n)=\hat{\mathbf{s}}(n|n-1)}$

Pred. MSE: $\mathbf{M}[n\,|\,n-1] = \mathbf{A}[n-1]\mathbf{M}[n-1\,|\,n-1]\mathbf{A}^T[n-1] + \mathbf{BQB}^T$

Kalman Gain: $\mathbf{K}[n] = \mathbf{M}[n\,|\,n-1]\mathbf{H}^T[n]\left(\mathbf{C}[n] + \mathbf{H}[n]\mathbf{M}[n\,|\,n-1]\mathbf{H}^T[n]\right)^{-1}$

Update: $\hat{\mathbf{s}}[n\,|\,n] = \hat{\mathbf{s}}[n\,|\,n-1] + \mathbf{K}[n]\left(\mathbf{x}[n] - \mathbf{h}_n\left(\hat{\mathbf{s}}[n\,|\,n-1]\right)\right)$

Est. MSE: $\mathbf{M}[n\,|\,n] = \left(\mathbf{I} - \mathbf{K}[n]\mathbf{H}[n]\right)\mathbf{M}[n\,|\,n-1]$