CS 4495 Computer Vision
Tracking 1- Kalman,Gaussian

Aaron Bobick

School of Interactive Computing

Administrivia

- PS5 – will be out this Thurs
  - Due **Sun** Nov 10th 11:55pm

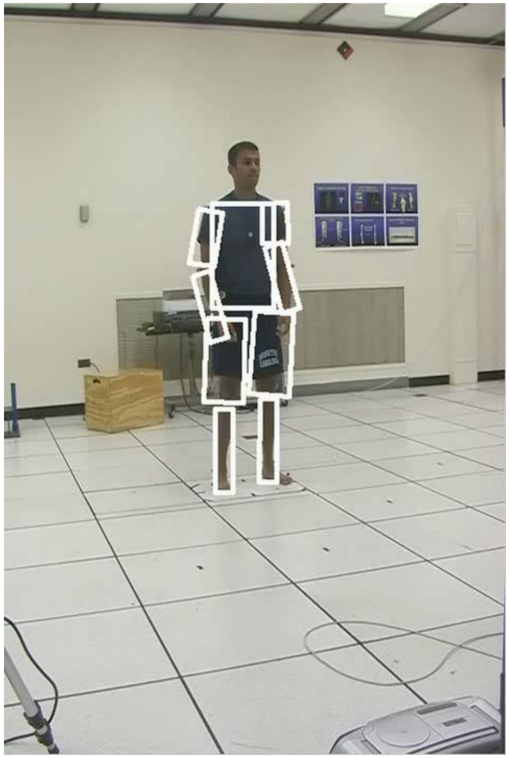
- Calendar (tentative) done for the year
  - PS6: 11/14, due 11/24   PS7: 11/26, due 12/5
  - EXAM: Tues before Thanksgiving. Covers concepts and basics
  - So no final on Dec 12….

Tracking

- Slides “adapted” from Kristen Grauman, Deva Ramanan, but mostly from Svetlana Lazebnik

Some examples

- Older examples:



- State of the art: http://www.youtube.com/watch?v=InqV34BcheM

Feature tracking

- So far, we have only considered optical flow estimation in a pair of images
- If we have more than two images, we can compute the optical flow from each frame to the next
- Given a point in the first image, we can in principle reconstruct its path by simply “following the arrows”

Tracking challenges

- Ambiguity of optical flow
  - Find good features to track
- Large motions
  - Discrete search instead of Lucas-Kanade
- Changes in shape, orientation, color
  - Allow some matching flexibility
- Occlusions, disocclusions
  - Need mechanism for deleting, adding new features

- Drift – errors may accumulate over time
  - Need to know when to terminate a track

Handling large displacements

- Define a small area around a pixel as the template
- Match the template against each pixel within a search area in next image – just like stereo matching!
- Use a match measure such as SSD or correlation
- After finding the best discrete location, can use Lucas-Kanade to get sub-pixel estimate (think of the template as the coarse level of the pyramid).

Tracking over many frames

- Select features in first frame
- For each frame:
  - Update positions of tracked features
    - Discrete search or Lucas-Kanade
  - Start new tracks if needed
  - Terminate inconsistent tracks
    - Compute similarity with corresponding feature in the previous frame or in the first frame where it's visible

- This is done by many companies and systems – often ad hoc rules tailored to the context.

Shi-Tomasi feature tracker

- Find good features using eigenvalues of second-moment matrix – *you've seen this now twice!*
  - Key idea: "good" features to track are the ones that can be tracked reliably
- From frame to frame, track with Lucas-Kanade and a pure translation model
  - More robust for small displacements, can be estimated from smaller neighborhoods
- Check consistency of tracks by affine registration to the first observed instance of the feature
  - Affine model is more accurate for larger displacements
  - Comparing to the first frame helps to minimize drift

J. Shi and C. Tomasi. Good Features to Track. CVPR 1994.

Tracking example

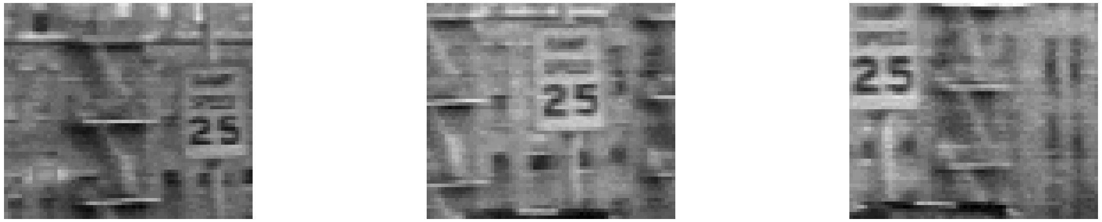


Figure 1: Three frame details from Woody Allen's *Manhattan*. The details are from the 1st, 11th, and 21st frames of a subsequence from the movie.

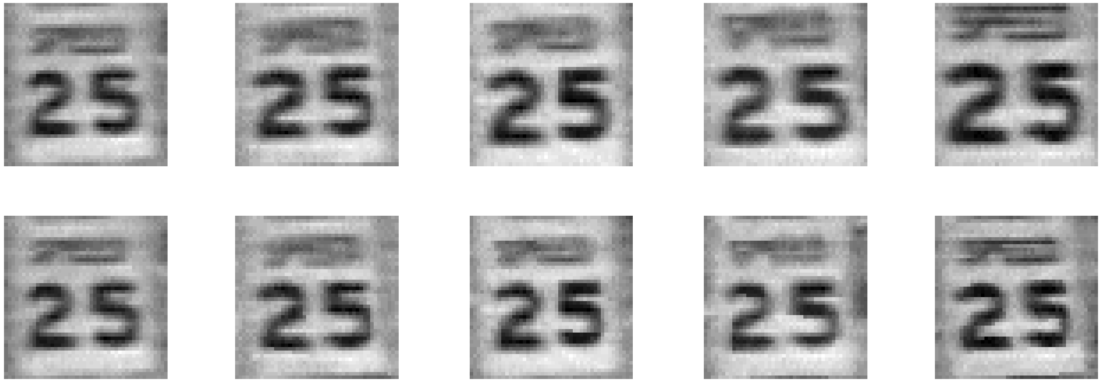


Figure 2: The traffic sign windows from frames 1,6,11,16,21 as tracked (top), and warped by the computed deformation matrices (bottom).

J. Shi and C. Tomasi. Good Features to Track. CVPR 1994.

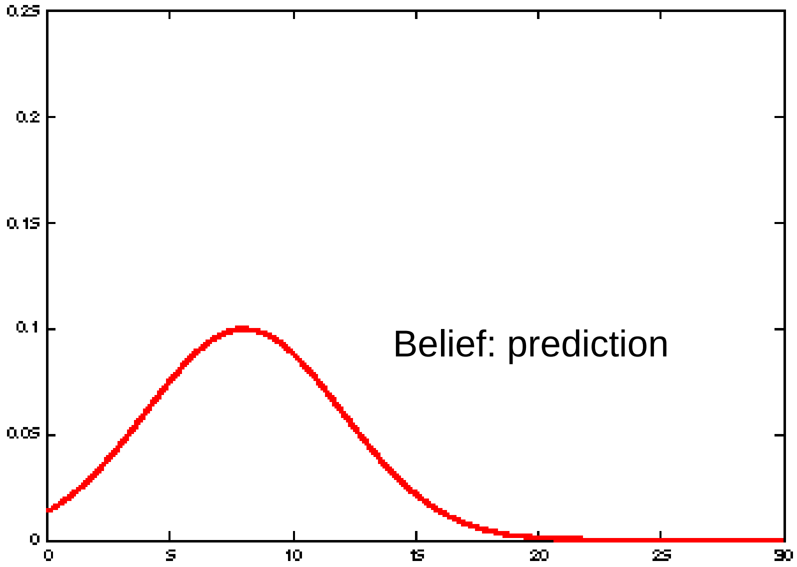
Tracking with dynamics

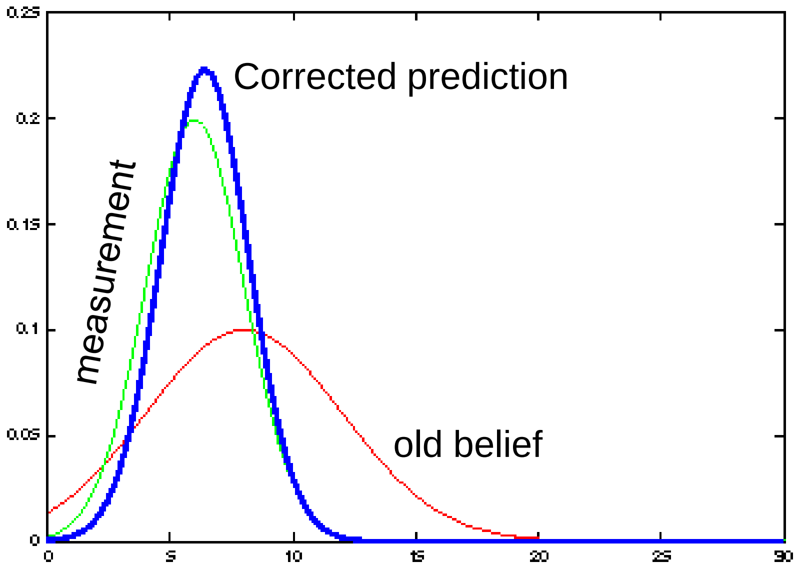
- Key idea: Given a model of expected motion, predict where objects will occur in next frame, even before seeing the image
  - Restrict search for the object
  - Improved estimates since measurement noise is reduced by trajectory smoothness

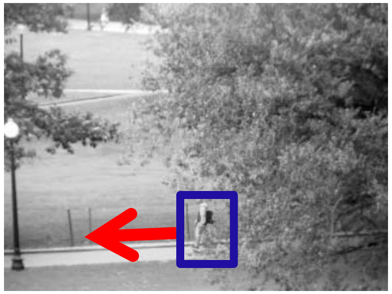
Tracking as inference

- The hidden state consists of the true parameters we care about, denoted X.
- The measurement is our noisy observation that results from the underlying state, denoted Y.
- At each time step, state changes (from X_{t-1} to X_t) and we get a new observation Y_t.
- Our goal: recover most likely state X_t given
  - All observations seen so far.
  - Knowledge about dynamics of state transitions.

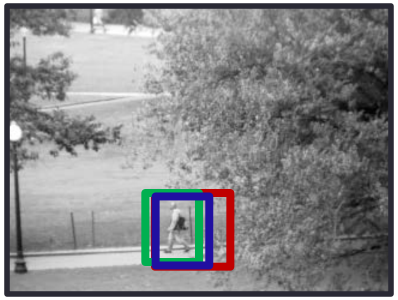
Tracking as inference: intuition







Time t



Time t+1

Steps of tracking

- **Prediction**: What is the next state of the object given past measurements?

$$P\left(X_t \middle| Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}\right)$$

Steps of tracking

- **Prediction**: What is the next state of the object given past measurements?

$$P\left(X_t \middle| Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}\right)$$

- **Correction**: Compute an updated estimate of the state from prediction and measurements

$$P\left(X_t \middle| Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, \boxed{Y_t = y_t}\right)$$

Steps of tracking

- Prediction: What is the next state of the object given past measurements?

$$P\left(X_t \middle| Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}\right)$$

- Correction: Compute an updated estimate of the state from prediction and measurements (posterior)

$$P\left(X_t \middle| Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, Y_t = y_t\right)$$

- *Tracking can be seen as the process of propagating the* ***posterior*** *distribution of state given measurements across time*

Simplifying assumptions

- Only the immediate past matters

$$P\left(X_t \middle| X_0, \ldots, X_{t-1}\right) = \boxed{P\left(X_t \middle| X_{t-1}\right)}$$

dynamics model

Simplifying assumptions

- Only the immediate past matters

$$P(X_t \mid X_0, \ldots, X_{t-1}) = \boxed{P(X_t \mid X_{t-1})}$$

dynamics model

- Measurements depend only on the current state

$$P(Y_t \mid X_0, Y_0 \ldots, X_{t-1}, Y_{t-1}, X_t) = \boxed{P(Y_t \mid X_t)}$$

observation model

Simplifying assumptions

- Only the immediate past matters

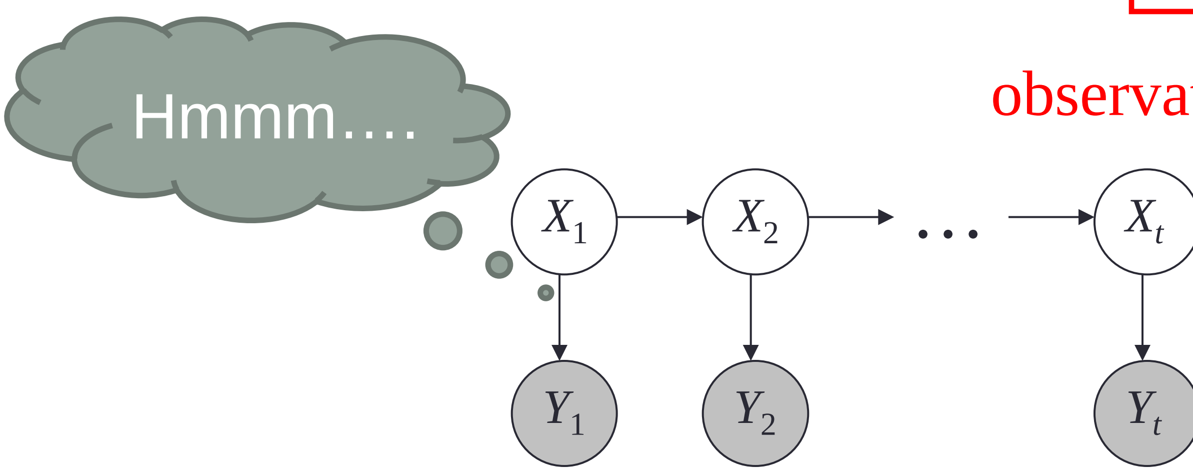
$$P\left(X_t \middle| X_0, \ldots, X_{t-1}\right) = \boxed{P\left(X_t \middle| X_{t-1}\right)}$$

dynamics model

- Measurements depend only on the current state

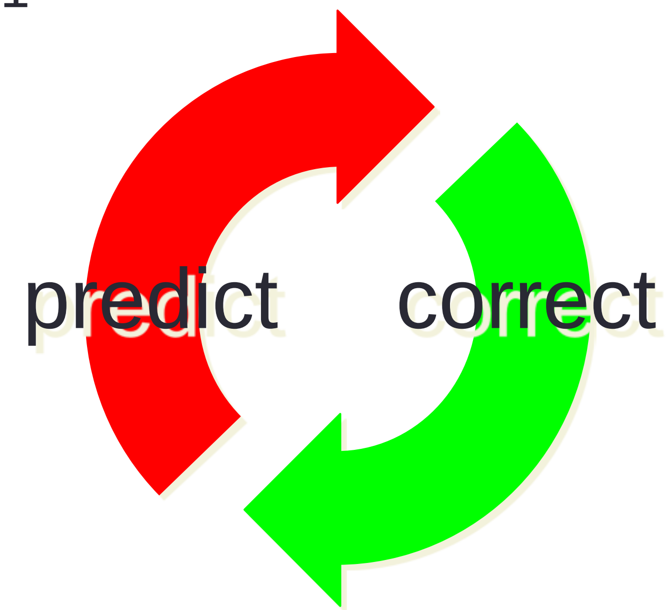
$$P\left(Y_t \middle| X_0, Y_0 \ldots, X_{t-1}, Y_{t-1}, X_t\right) = \boxed{P\left(Y_t \middle| X_t\right)}$$

observation model



Tracking as induction

- Base case:
  - Assume we have initial prior that predicts state in absence of any evidence: $P(X_0)$
  - At the first frame, *correct* this given the value of $Y_0=y_0$
- Given corrected estimate for frame t:
  - Predict for frame t+1
  - Correct for frame t+1



Tracking as induction

- Base case:
  - Assume we have initial prior that predicts state in absence of any evidence: $P(X_0)$
  - At the first frame, *correct* this given the value of $Y_0=y_0$

$$P(X_0 \mid Y_0 = y_0) = \frac{P(y_0 \mid X_0)P(X_0)}{P(y_0)} \propto P(y_0 \mid X_0)P(X_0)$$

Prediction

- Prediction involves guessing $P\left(X_t \middle| y_0, \ldots, y_{t-1}\right)$ given $P\left(X_{t-1} \middle| y_0, \ldots, y_{t-1}\right)$

Prediction

- Prediction involves guessing $P\left(X_t \middle| y_0, \ldots, y_{t-1}\right)$ given $P\left(X_{t-1} \middle| y_0, \ldots, y_{t-1}\right)$

$$P\left(X_t \middle| y_0, \ldots, y_{t-1}\right)$$

$$= \int P\left(X_t, X_{t-1} \middle| y_0, \ldots, y_{t-1}\right) dX_{t-1}$$

Law of total probability - Marginalization

Prediction

- Prediction involves guessing $P\left(X_t \middle| y_0,\ldots,y_{t-1}\right)$ given $P\left(X_{t-1} \middle| y_0,\ldots,y_{t-1}\right)$

$$P\left(X_t \middle| y_0,\ldots,y_{t-1}\right)$$

$$= \int P\left(X_t, X_{t-1} \middle| y_0,\ldots,y_{t-1}\right) dX_{t-1}$$

$$= \int P\left(X_t \mid X_{t-1}, y_0,\ldots,y_{t-1}\right) P\left(X_{t-1} \mid y_0,\ldots,y_{t-1}\right) dX_{t-1}$$

Conditioning on X_{t-1}

Prediction

- Prediction involves guessing $P\left(X_t \middle| y_0, \ldots, y_{t-1}\right)$ given $P\left(X_{t-1} \middle| y_0, \ldots, y_{t-1}\right)$

$$
\begin{aligned}
& P\left(X_t \middle| y_0, \ldots, y_{t-1}\right) \\
& = \int P\left(X_t, X_{t-1} \middle| y_0, \ldots, y_{t-1}\right) dX_{t-1} \\
& = \int P\left(X_t \mid X_{t-1}, y_0, \ldots, y_{t-1}\right) P\left(X_{t-1} \mid y_0, \ldots, y_{t-1}\right) dX_{t-1} \\
& = \int P\left(X_t \mid X_{t-1}\right) P\left(X_{t-1} \mid y_0, \ldots, y_{t-1}\right) dX_{t-1}
\end{aligned}
$$

Independence assumption

Correction

- Correction involves computing $P\left(X_t \middle| y_0, \ldots, y_t\right)$ given predicted value $P\left(X_t \middle| y_0, \ldots, y_{t-1}\right)$ and y_t

Correction

- Correction involves computing $P\left(X_t \middle| y_0, \ldots, y_t\right)$ given predicted value $P\left(X_t \middle| y_0, \ldots, y_{t-1}\right)$ and y_t

$$P\left(X_t \middle| y_0, \ldots, y_t\right)$$

$$= \frac{P\left(y_t \mid X_t, y_0, \ldots, y_{t-1}\right) P\left(X_t \mid y_0, \ldots, y_{t-1}\right)}{P\left(y_t \mid y_0, \ldots, y_{t-1}\right)}$$

Bayes rule

Correction

- Correction involves computing $P\left(X_t \middle| y_0, \ldots, y_t\right)$ given predicted value $P\left(X_t \middle| y_0, \ldots, y_{t-1}\right)$ and y_t

$$P\left(X_t \middle| y_0, \ldots, y_t\right)$$

$$= \frac{P\left(y_t \mid X_t, y_0, \ldots, y_{t-1}\right) P\left(X_t \mid y_0, \ldots, y_{t-1}\right)}{P\left(y_t \mid y_0, \ldots, y_{t-1}\right)}$$

$$= \frac{P\left(y_t \mid X_t\right) P\left(X_t \mid y_0, \ldots, y_{t-1}\right)}{P\left(y_t \mid y_0, \ldots, y_{t-1}\right)}$$

Independence assumption
(observation y_t depends only on state X_t)

Correction

- Correction involves computing $P(X_t | y_0, \ldots, y_t)$ given predicted value $P(X_t | y_0, \ldots, y_{t-1})$ and y_t

$$P(X_t | y_0, \ldots, y_t)$$

$$= \frac{P(y_t \mid X_t, y_0, \ldots, y_{t-1}) P(X_t \mid y_0, \ldots, y_{t-1})}{P(y_t \mid y_0, \ldots, y_{t-1})}$$

$$= \frac{P(y_t \mid X_t) P(X_t \mid y_0, \ldots, y_{t-1})}{P(y_t \mid y_0, \ldots, y_{t-1})}$$

$$= \frac{P(y_t \mid X_t) P(X_t \mid y_0, \ldots, y_{t-1})}{\int P(y_t \mid X_t) P(X_t \mid y_0, \ldots, y_{t-1}) dX_t}$$

Really a normalization

Conditioning on X_t

Summary: Prediction and correction

- Prediction:

$$P(X_t \mid y_0,\ldots,y_{t-1}) = \int \underbrace{P(X_t \mid X_{t-1})}_{\text{dynamics model}} \underbrace{P(X_{t-1} \mid y_0,\ldots,y_{t-1})}_{\text{corrected estimate from previous step}} dX_{t-1}$$

Summary: Prediction and correction

- Prediction:

$$P(X_t \mid y_0,\ldots,y_{t-1}) = \int \underbrace{P(X_t \mid X_{t-1})}_{\text{dynamics model}} \underbrace{P(X_{t-1} \mid y_0,\ldots,y_{t-1})}_{\text{corrected estimate from previous step}} dX_{t-1}$$

- Correction:

$$P(X_t \mid y_0,\ldots,y_t) = \frac{\overbrace{P(y_t \mid X_t)}^{\text{observation model}} \overbrace{P(X_t \mid y_0,\ldots,y_{t-1})}^{\text{predicted estimate}}}{\int P(y_t \mid X_t) P(X_t \mid y_0,\ldots,y_{t-1}) dX_t}$$

Linear Dynamic Models

- Dynamics model: state undergoes linear transformation plus Gaussian noise
- $$\mathbf{x}_t \sim N\left(D_t \mathbf{x}_{t-1}, \Sigma_{d_t}\right)$$
- Observation model: measurement is linearly transformed state plus Gaussian noise

$$\mathbf{y}_t \sim N\left(M_t \mathbf{x}_t, \Sigma_{m_t}\right)$$

Example: Constant velocity (1D)

- State vector is position and velocity

$$x_t = \begin{bmatrix} p_t \\ v_t \end{bmatrix} \qquad \begin{aligned} p_t &= p_{t-1} + (\Delta t) v_{t-1} + \varepsilon \\ v_t &= v_{t-1} + \xi \end{aligned}$$

(greek letters denote noise terms)

$$x_t = D_t x_{t-1} + noise = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_{t-1} \\ v_{t-1} \end{bmatrix} + noise$$

- Measurement is position only

$$y_t = M x_t + noise = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} p_t \\ v_t \end{bmatrix} + noise$$

Example: Constant acceleration (1D)

- State vector is position, velocity, and acceleration

$$x_t = \begin{bmatrix} p_t \\ v_t \\ a_t \end{bmatrix} \qquad \begin{aligned} p_t &= p_{t-1} + (\Delta t)v_{t-1} + \varepsilon \\ v_t &= v_{t-1} + (\Delta t)a_{t-1} + \xi \\ a_t &= a_{t-1} + \zeta \end{aligned}$$

(greek letters denote noise terms)

$$x_t = D_t x_{t-1} + noise = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_{t-1} \\ v_{t-1} \\ a_{t-1} \end{bmatrix} + noise$$

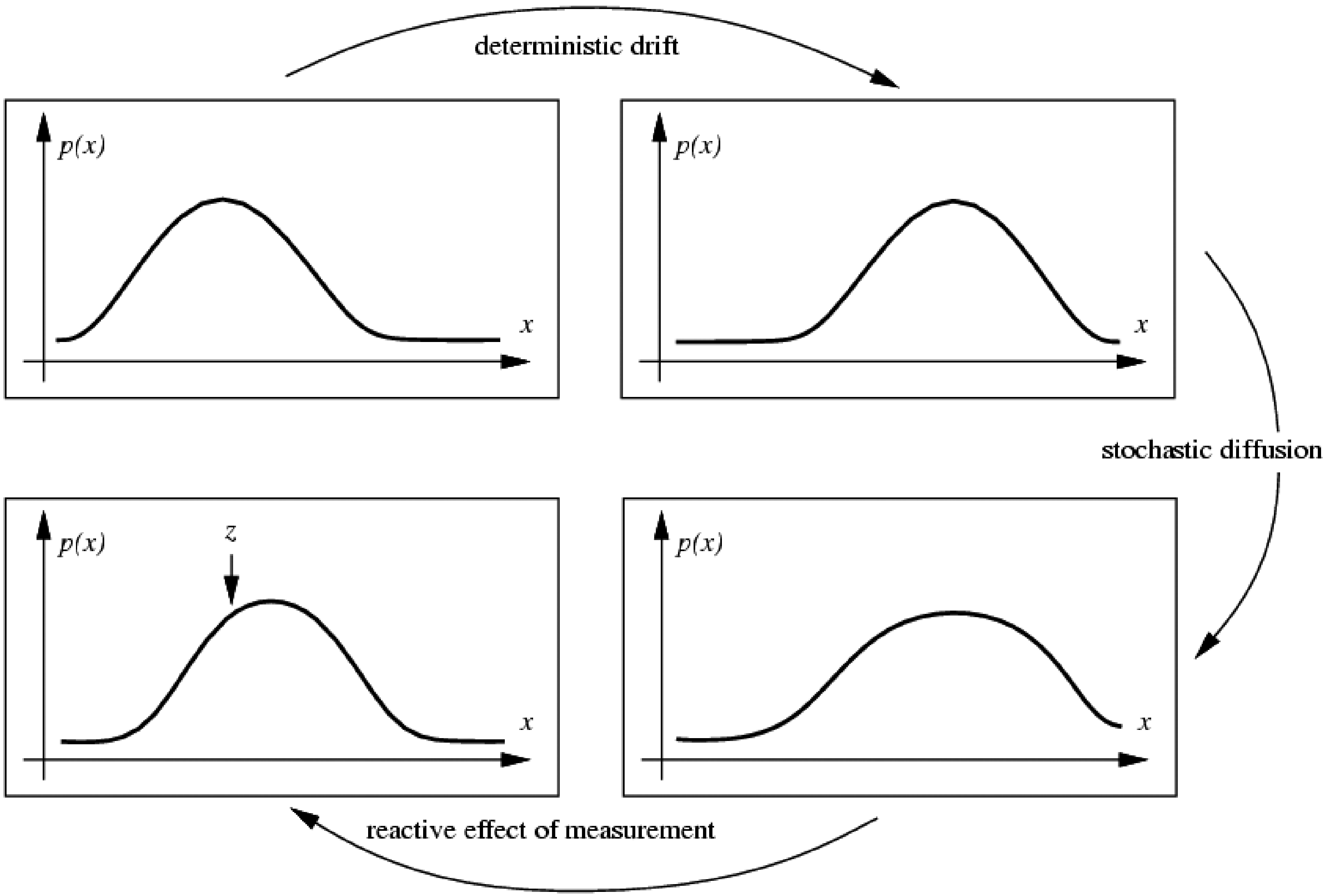
- Measurement is position only

$$y_t = Mx_t + noise = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_t \\ v_t \\ a_t \end{bmatrix} + noise$$

The Kalman filter

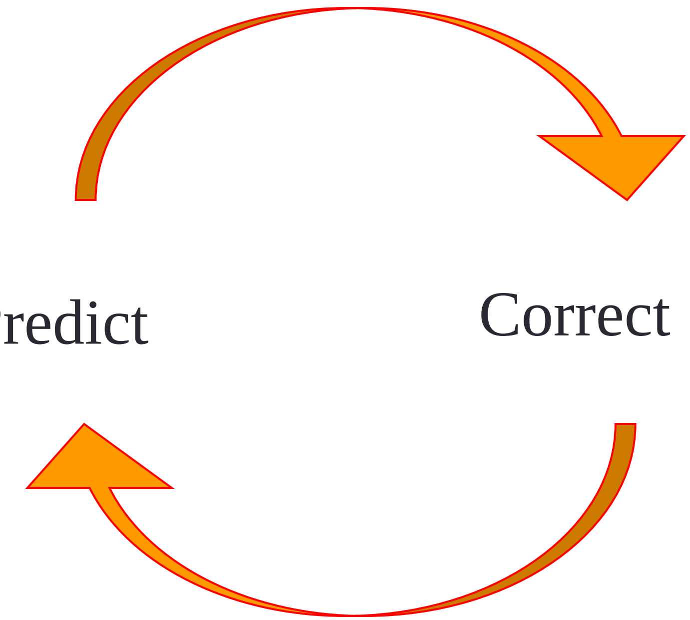
- Method for tracking linear dynamical models in Gaussian noise
- The predicted/corrected state distributions are Gaussian
  - You only need to maintain the mean and covariance
  - The calculations are easy (all the integrals can be done in closed form)

Propagation of Gaussian densities



The Kalman Filter: 1D state

Make measurement



Given corrected state from previous time step and all the measurements up to the current one, predict the distribution over the current step

Predict

Correct

Given prediction of state and current measurement, update prediction of state

$$P\left(X_t \mid y_0, \ldots, y_t\right)$$

$$P\left(X_t \mid y_0, \ldots, y_{t-1}\right)$$

Mean and std. dev. of predicted state:

$$\mu_t^-, \sigma_t^-$$

Time advances (from t–1 to t)

Mean and std. dev. of corrected state:

$$\mu_t^+, \sigma_t^+$$

1D Kalman filter: Prediction

- Linear dynamic model defines predicted state evolution, with noise
- $$X_t \sim N\left(dx_{t-1}, \sigma_d^2\right)$$
- Want to estimate distribution for next predicted state

$$P\left(X_t \mid y_0, \ldots, y_{t-1}\right) = \int P\left(X_t \mid X_{t-1}\right) P\left(X_{t-1} \mid y_0, \ldots, y_{t-1}\right) dX_{t-1}$$

1D Kalman filter: Prediction

- Linear dynamic model defines predicted state evolution, with noise

$$X_t \sim N\left(dx_{t-1}, \sigma_d^2\right)$$

- Want to estimate distribution for next predicted state

$$P\left(X_t \middle| y_0, \ldots, y_{t-1}\right) = N\left(\mu_t^-, (\sigma_t^-)^2\right)$$

- Update the mean: $\mu_t^- = d\mu_{t-1}^+$

- Update the variance: $(\sigma_t^-)^2 = \sigma_d^2 + (d\sigma_{t-1}^+)^2$

1D Kalman filter: Correction

- Mapping of state to measurements: $Y_t \sim N\left(mx_t, \sigma_m^2\right)$
- Predicted state: $P\left(X_t \middle| y_0, \ldots, y_{t-1}\right) = N\left(\mu_t^-, (\sigma_t^-)^2\right)$
- Want to estimate corrected distribution

$$P\left(X_t \mid y_0, \ldots, y_t\right) = \frac{P\left(y_t \mid X_t\right) P\left(X_t \mid y_0, \ldots, y_{t-1}\right)}{\int P\left(y_t \mid X_t\right) P\left(X_t \mid y_0, \ldots, y_{t-1}\right) dX_t}$$

1D Kalman filter: Correction

- Mapping of state to measurements: $Y_t \sim N\left(mx_t, \sigma_m^2\right)$
- Predicted state: $P\left(X_t \middle| y_0, \ldots, y_{t-1}\right) = N\left(\mu_t^-, (\sigma_t^-)^2\right)$
- We define the corrected distribution to be:

$$P\left(X_t \middle| y_0, \ldots, y_t\right) \equiv N\left(\mu_t^+, (\sigma_t^+)^2\right)$$

1D Kalman filter: Correction

- Mapping of state to measurements: $Y_t \sim N\left(mx_t, \sigma_m^2\right)$
- Predicted state: $P\left(X_t \middle| y_0, \ldots, y_{t-1}\right) = N\left(\mu_t^-, (\sigma_t^-)^2\right)$
- Want to estimate corrected distribution

$$P\left(X_t \middle| y_0, \ldots, y_t\right) = N\left(\mu_t^+, (\sigma_t^+)^2\right)$$

- Update the mean:

$$\mu_t^+ = \frac{\mu_t^- \sigma_m^2 + m y_t (\sigma_t^-)^2}{\sigma_m^2 + m^2 (\sigma_t^-)^2}$$

- Update the variance:

$$(\sigma_t^+)^2 = \frac{\sigma_m^2 (\sigma_t^-)^2}{\sigma_m^2 + m^2 (\sigma_t^-)^2}$$

1D Kalman filter: Correction

From:

$$\mu_t^+ = \frac{\mu_t^- \sigma_m^2 + m y_t (\sigma_t^-)^2}{\sigma_m^2 + m^2 (\sigma_t^-)^2}$$

Prediction of x

Measurement guess of x

Variance of prediction

Variance of x computed from the measurement

$$\mu_t^+ = \frac{\frac{\mu_t^- \sigma_m^2}{m^2} + \frac{y_t}{m} (\sigma_t^-)^2}{\frac{\sigma_m^2}{m^2} + (\sigma_t^-)^2}$$

- What is this?
- ***The weighted average of prediction and measurement based on variances!***

Prediction vs. correction

$$\mu_t^+ = \frac{\mu_t^- \sigma_m^2 + m y_t (\sigma_t^-)^2}{\sigma_m^2 + m^2 (\sigma_t^-)^2} \qquad (\sigma_t^+)^2 = \frac{\sigma_m^2 (\sigma_t^-)^2}{\sigma_m^2 + m^2 (\sigma_t^-)^2}$$

- What if there is no prediction uncertainty $(\sigma_t^- = 0)$?
- $$\mu_t^+ = \mu_t^- \qquad (\sigma_t^+)^2 = 0$$

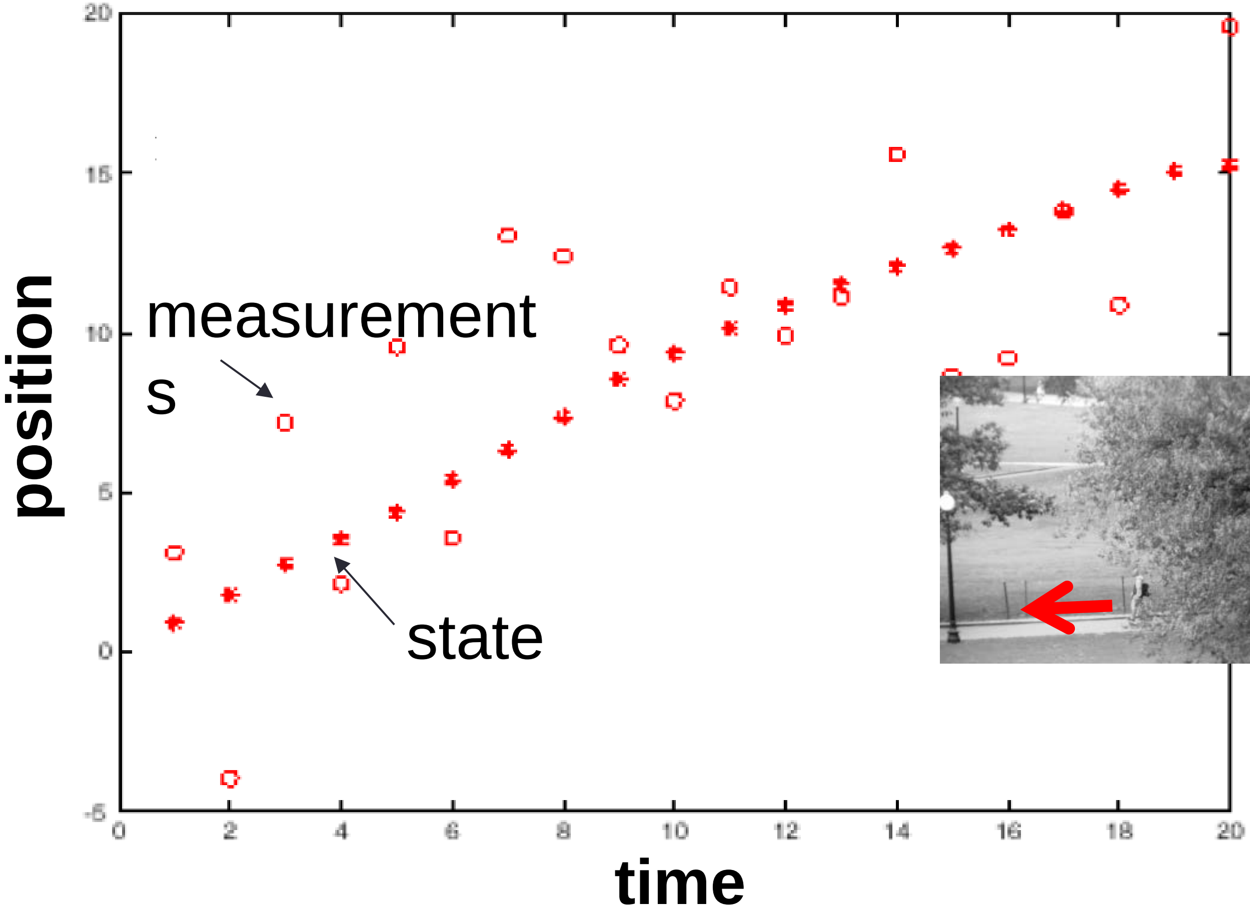
The measurement is ignored!

- What if there is no measurement uncertainty $(\sigma_m = 0)$?

$$\mu_t^+ = \frac{y_t}{m} \qquad (\sigma_t^+)^2 = 0$$

The prediction is ignored!

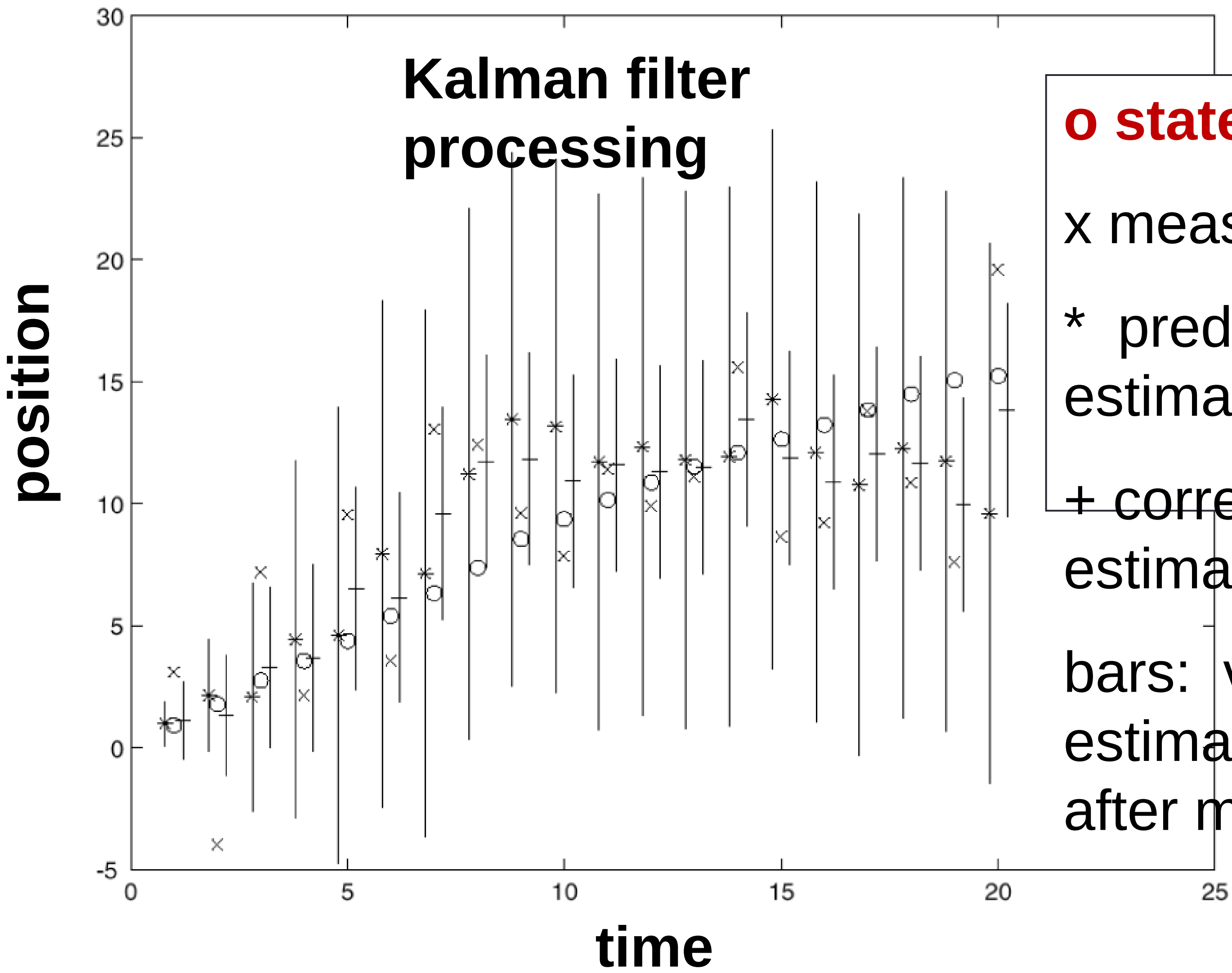
Recall: constant velocity example



State is 2d: position + velocity

Measurement is 1d: position

Constant velocity model



o state

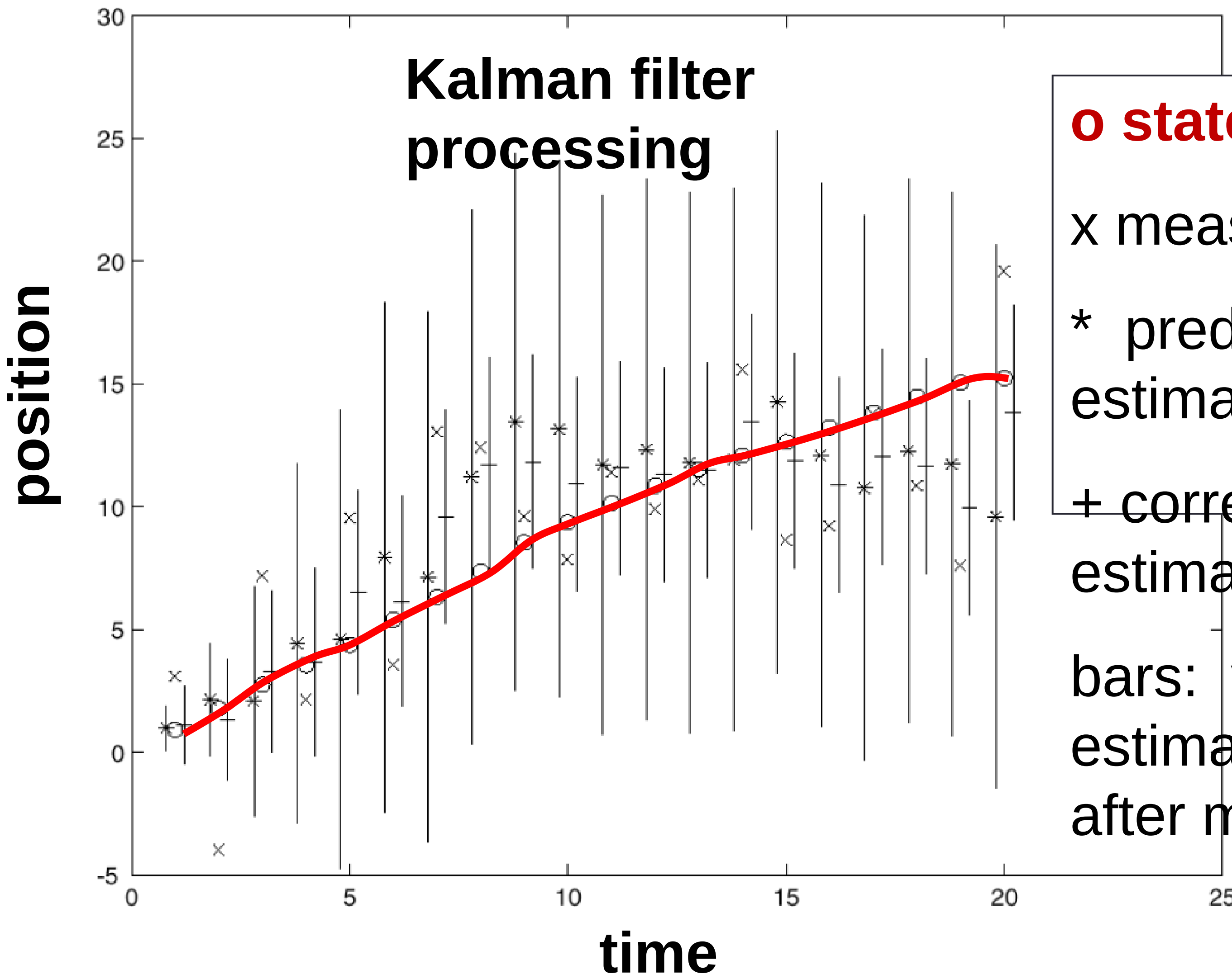
x measurement

* predicted mean estimate

+ corrected mean estimate

bars: variance estimates before and after measurements

Constant velocity model



o state

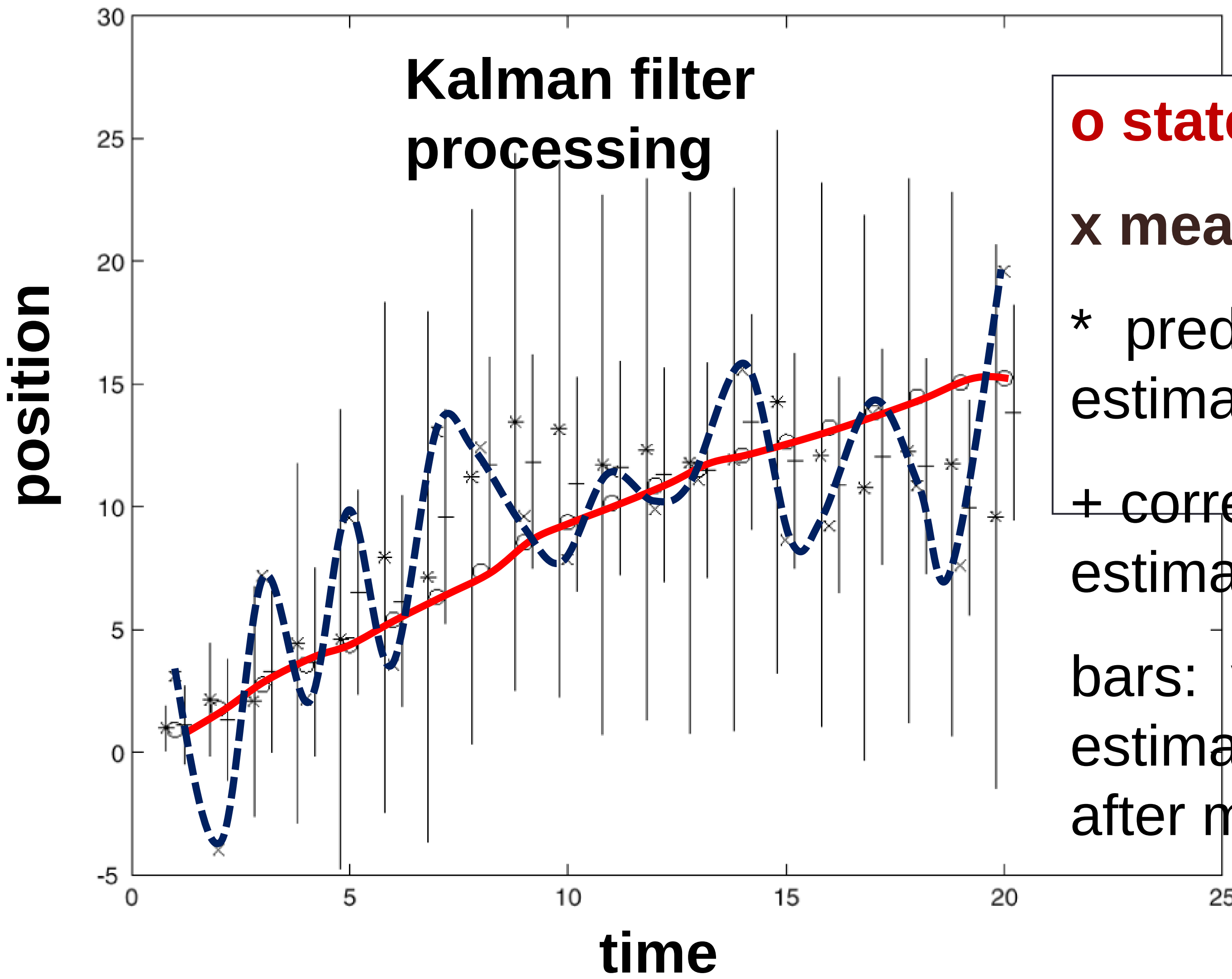
x measurement

* predicted mean estimate

+ corrected mean estimate

bars: variance estimates before and after measurements

Constant velocity model



o state

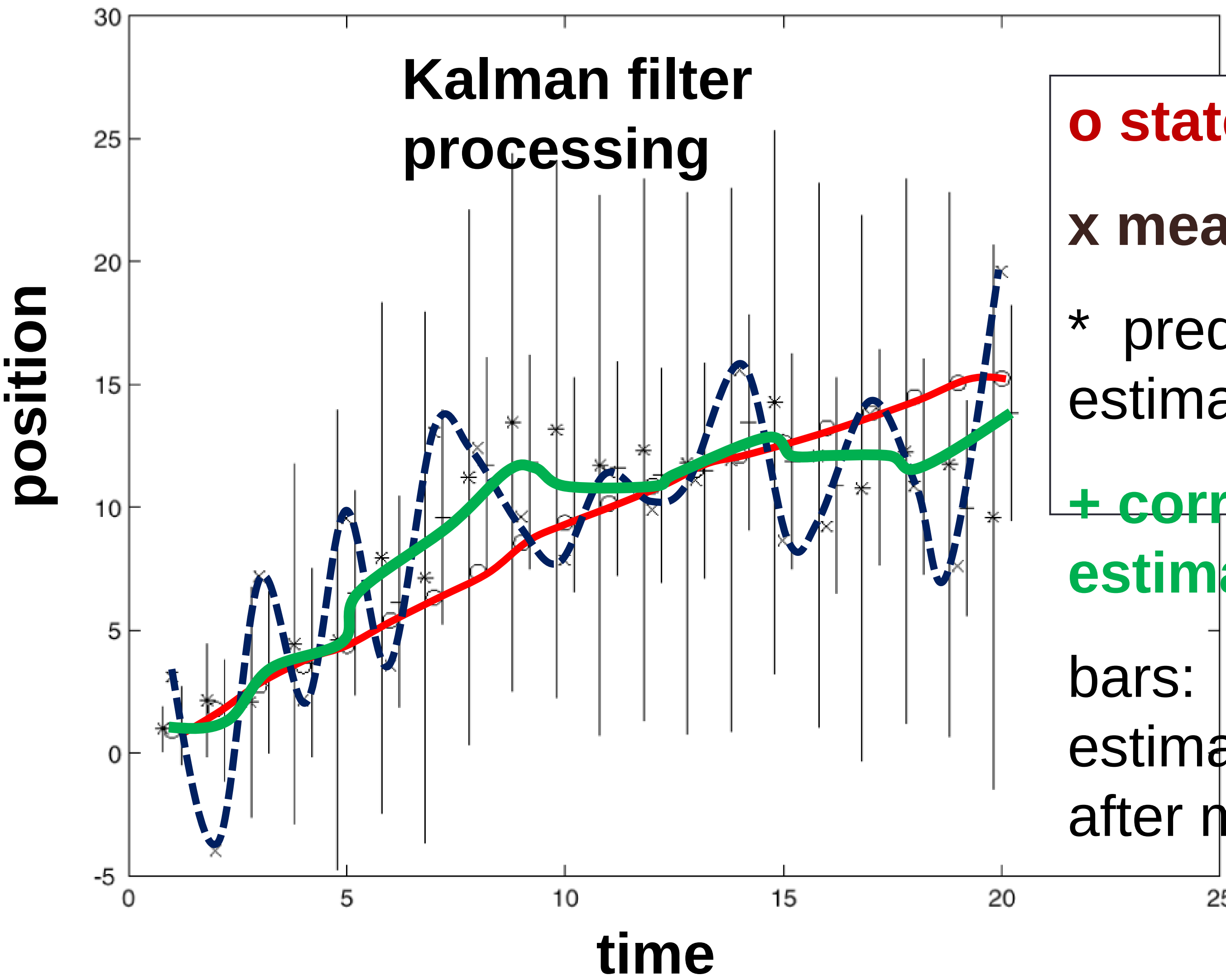
x measurement

* predicted mean estimate

+ corrected mean estimate

bars: variance estimates before and after measurements

Constant velocity model



o state

x measurement

* predicted mean estimate

+ corrected mean estimate

bars: variance estimates before and after measurements

Kalman filter: General case (> 1dim)

What if state vectors have more than one dimension?

PREDICT

$$x_t^- = D_t x_{t-1}^+$$

$$\Sigma_t^- = D_t \Sigma_{t-1}^+ D_t^T + \Sigma_{d_t}$$

CORRECT

$$K_t = \Sigma_t^- M_t^T \left(M_t \Sigma_t^- M_t^T + \Sigma_{m_t} \right)^{-1}$$

$$x_t^+ = x_t^- + K_t \boxed{\left(y_t - M_t x_t^- \right)}$$

$$\Sigma_t^+ = \left(I - K_t M_t \right) \Sigma_t^-$$

More weight on residual when measurement error covariance approaches 0.

Kalman filter pros and cons

- Pros
  - Simple updates, compact and efficient
- Cons
  - Unimodal distribution, only single hypothesis
  - Restricted class of motions defined by linear model
    - Extensions call “Extended Kalman Filtering”

- So what might we do if not Gaussian? Or even unimodal?

Find out next time! (Actually next Thurs)