Robotics 2
Target Tracking

Giorgio Grisetti, Cyrill Stachniss,
Kai Arras, Wolfram Burgard

Linear Dynamical System (LDS)

- Stochastic process governed by

$$\boxed{\dot{x}(t) = A(t)x(t) + B(t)u(t) + \xi(t)}$$

  - $x \in \mathbb{R}^{n_x}$ is the state vector
  - $u \in \mathbb{R}^{n_u}$ is the input vector
  - $\xi \in \mathbb{R}^{n_x}$ is the process noise
  - $A \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$ is the system matrix
  - $B \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}$ is the input gain

- The system can be observed through

$$\boxed{z(t) = H(t)x(t) + \epsilon(t)}$$

  - $z \in \mathbb{R}^{n_z}$ is the measurement vector
  - $\epsilon \in \mathbb{R}^{n_z}$ is the measurement noise
  - $H \in \mathbb{R}^{n_z} \times \mathbb{R}^{n_x}$ is the measurement matrix

Discrete Time LDS

- Continuous model are difficult to realize
  - Algorithms work at discrete time steps
  - Measurements are acquired with certain rates

- In practice, **discrete models** are employed

- Discrete-time LDS are governed by

$$\boxed{x(k+1) = F(k)x(k) + G(k)u(k) + \xi(k)}$$

  - $F \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$ is the state transition matrix
  - $G \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}$ is the discrete-time input gain

- Same observation function of continuous models

Discrete Time LDS

- Continuous model are difficult to realize
  - Algorithms work at discrete time steps
  - Measurements are acquired with certain rates

- In practice, **discrete models** are employed

- Discrete-time LDS are governed by

In target tracking, the input is unknown!

$$\boxed{x(k+1) = F(k)x(k) + \cancel{G(k)u(k)} + \xi(k)}$$

  - $F \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$ is the state transition matrix
  - $G \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}$ is the discrete-time input gain

- Same observation function of continuous models

LDS Example – Throwing ball

- We want to throw a ball and **compute its trajectory**
- This can be **easily done with an LDS**
- The ball's **state** shall be represented as

$$\mathbf{x} = \left[\begin{array}{cccc} x & y & \dot{x} & \dot{y} \end{array}\right]^T$$

- We ignore winds but consider the **gravity force** g

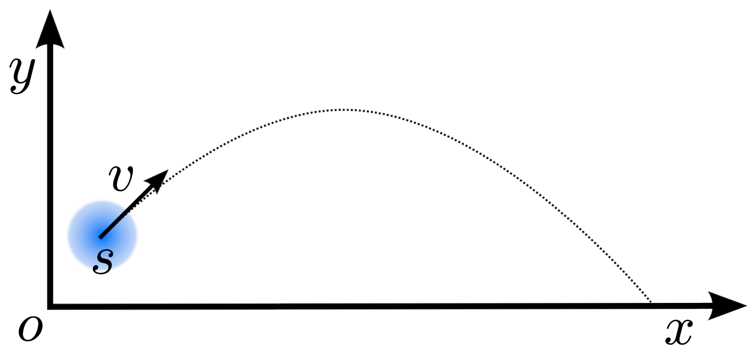
$$\mathbf{u} = -g$$

- No floor constraints
- We **observe** the ball with a noise-free position sensor

$$\mathbf{z} = \left[\begin{array}{cc} x & y \end{array}\right]^T$$

LDS Example – Throwing ball

- Throwing a ball from s with initial velocity v
- Consider only the gravity force, g, of the ball



- State vector
  $\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} \end{bmatrix}^T$
- Initial state
  $\mathbf{x}_0 = \begin{bmatrix} s_x & s_y & v_x & v_y \end{bmatrix}^T$
- Input vector (scalar)
  $\mathbf{u} = -g$
- Measurement vector
  $\mathbf{z} = \begin{bmatrix} x & y \end{bmatrix}^T$

- Process matrices

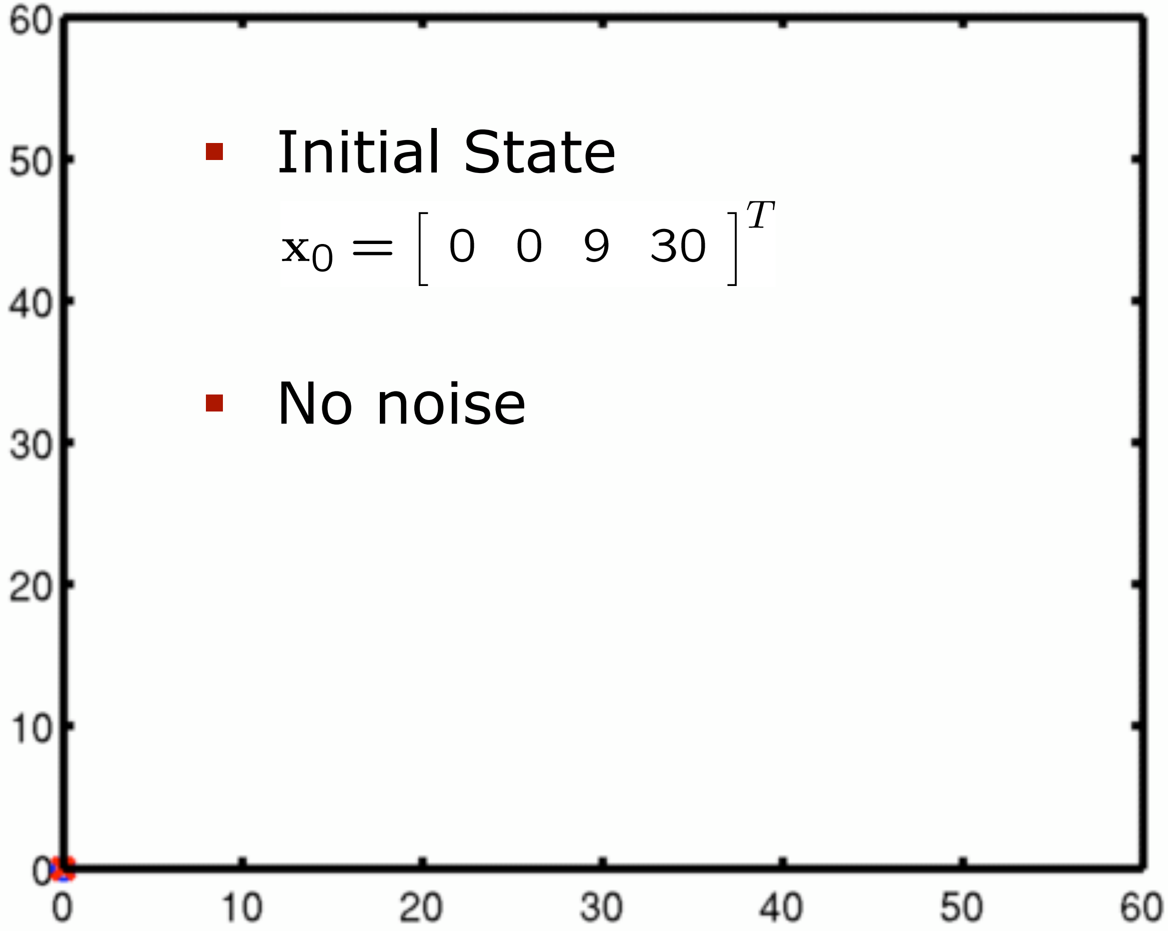
$$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$G = \begin{bmatrix} 0 & \frac{T^2}{2} & 0 & T \end{bmatrix}^T$$

- Measurement matrix

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

LDS Example – Throwing ball

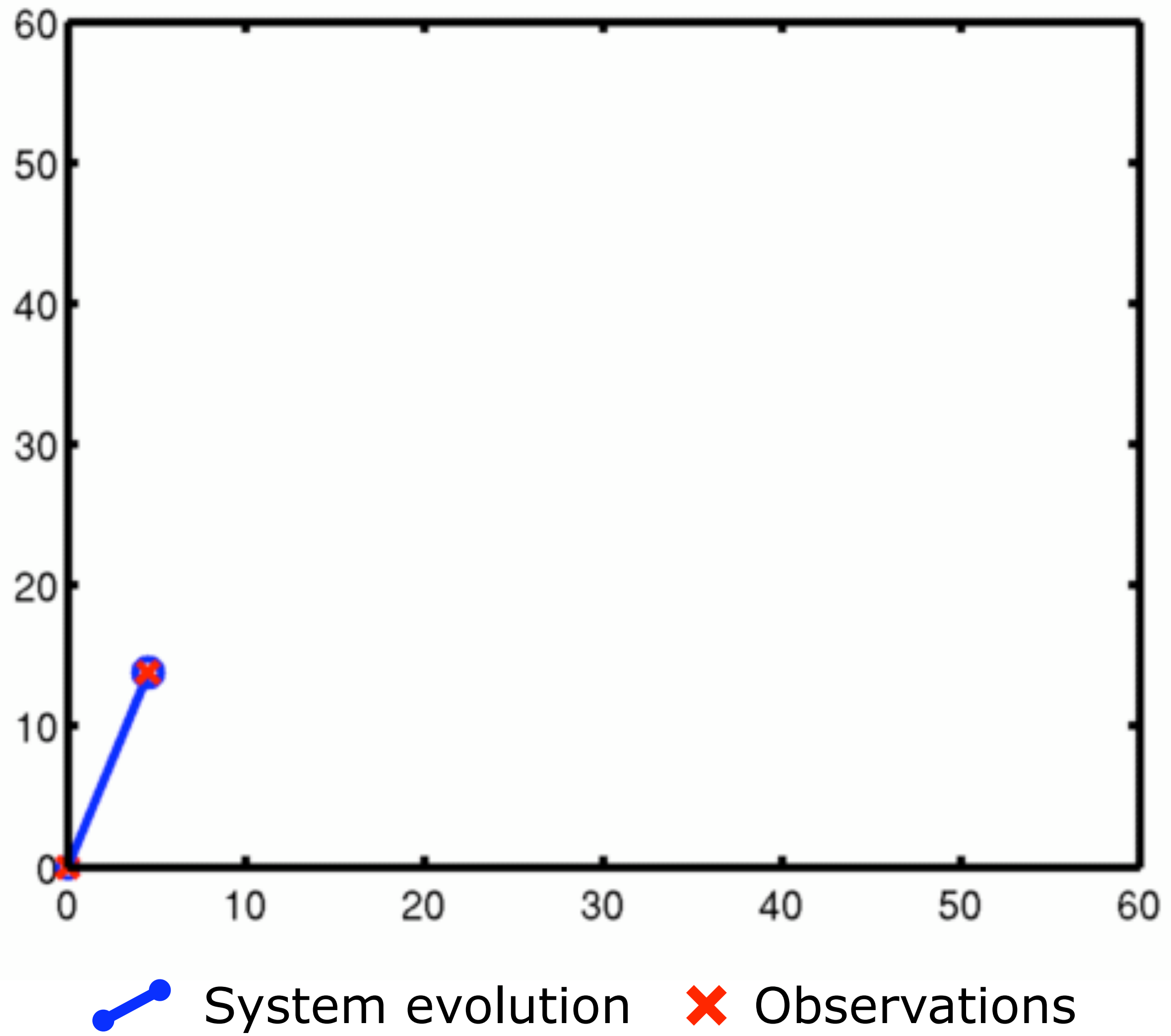


- Initial State

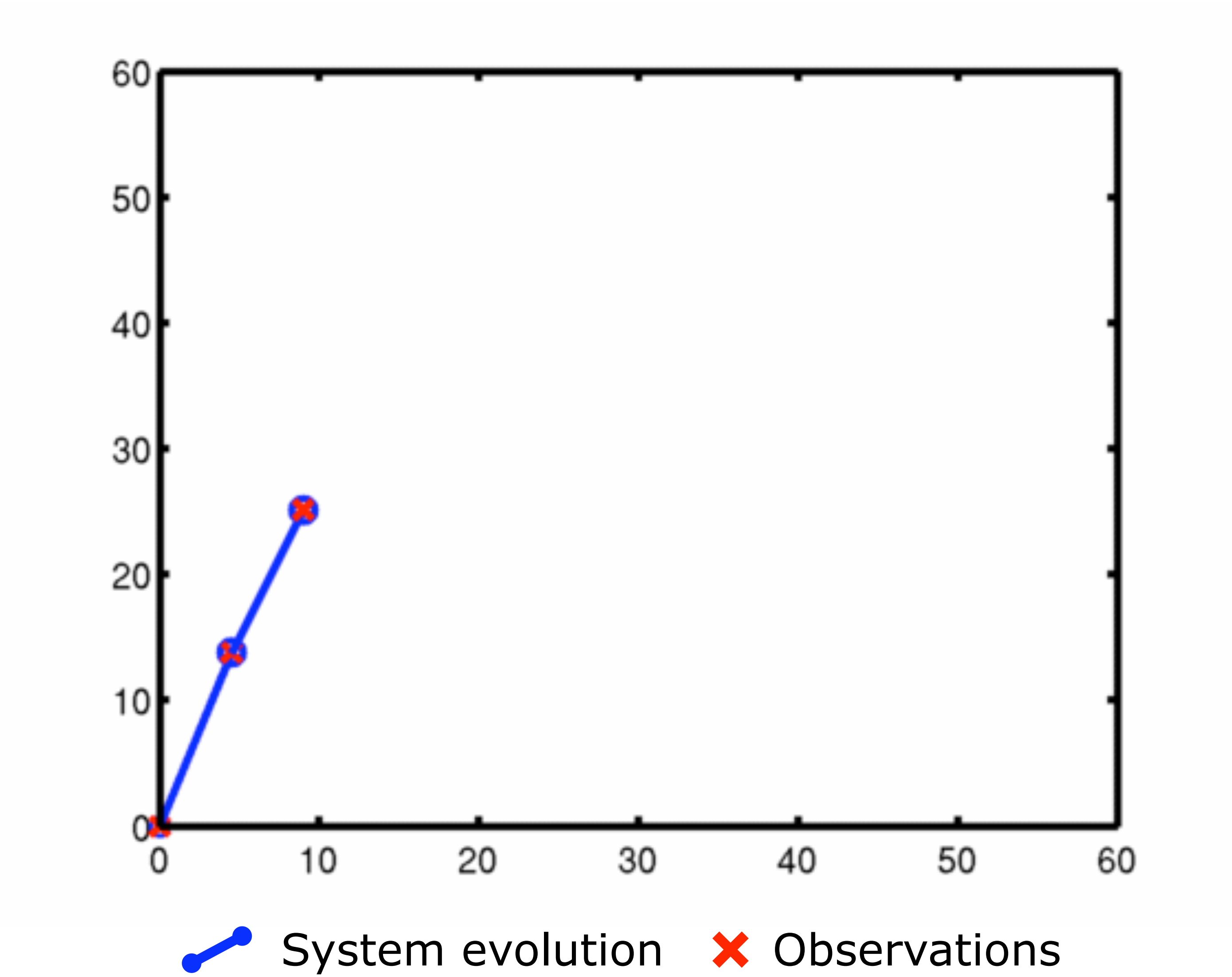
$$\mathbf{x}_0 = \begin{bmatrix} 0 & 0 & 9 & 30 \end{bmatrix}^T$$

- No noise

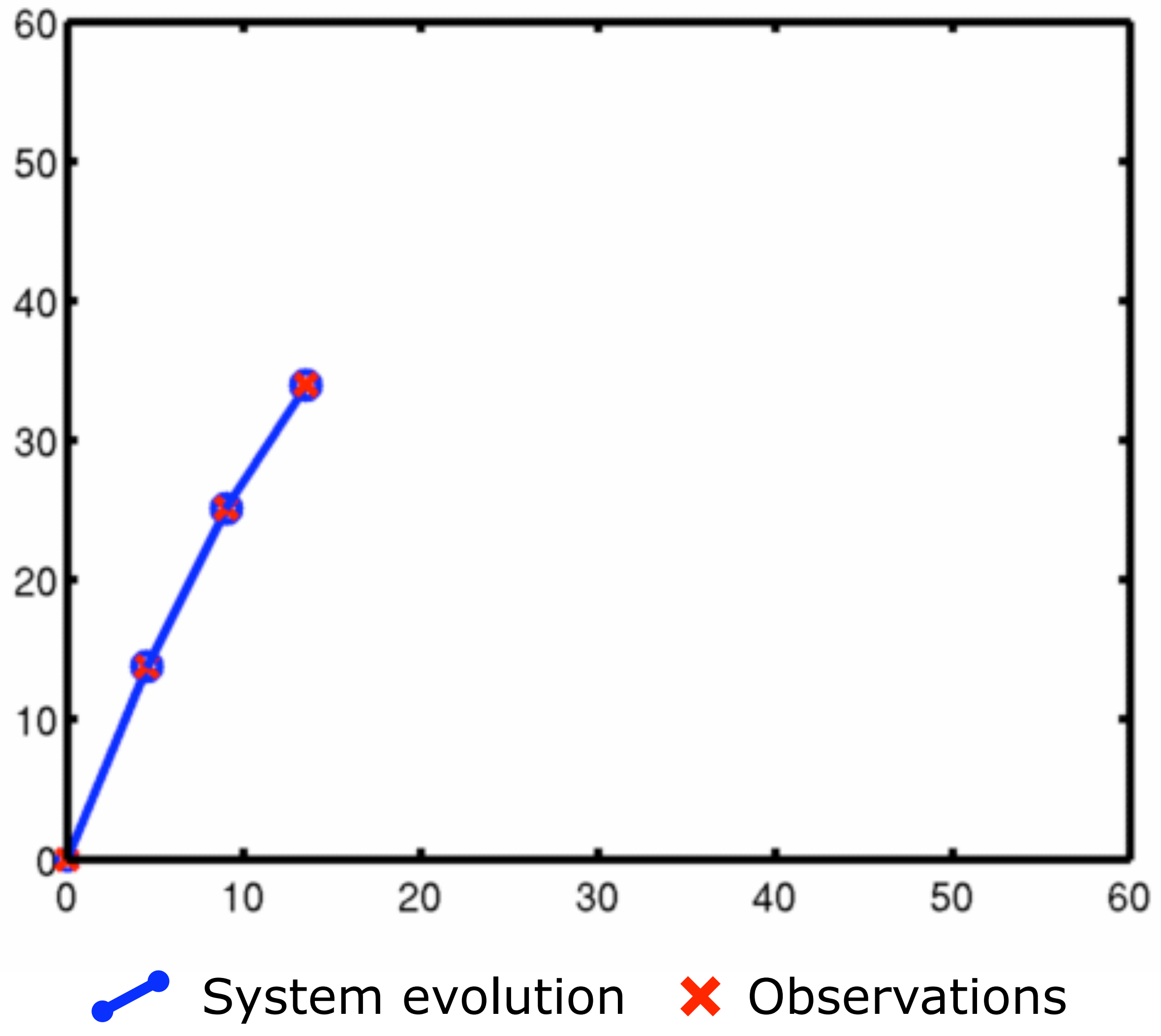
LDS Example – Throwing ball



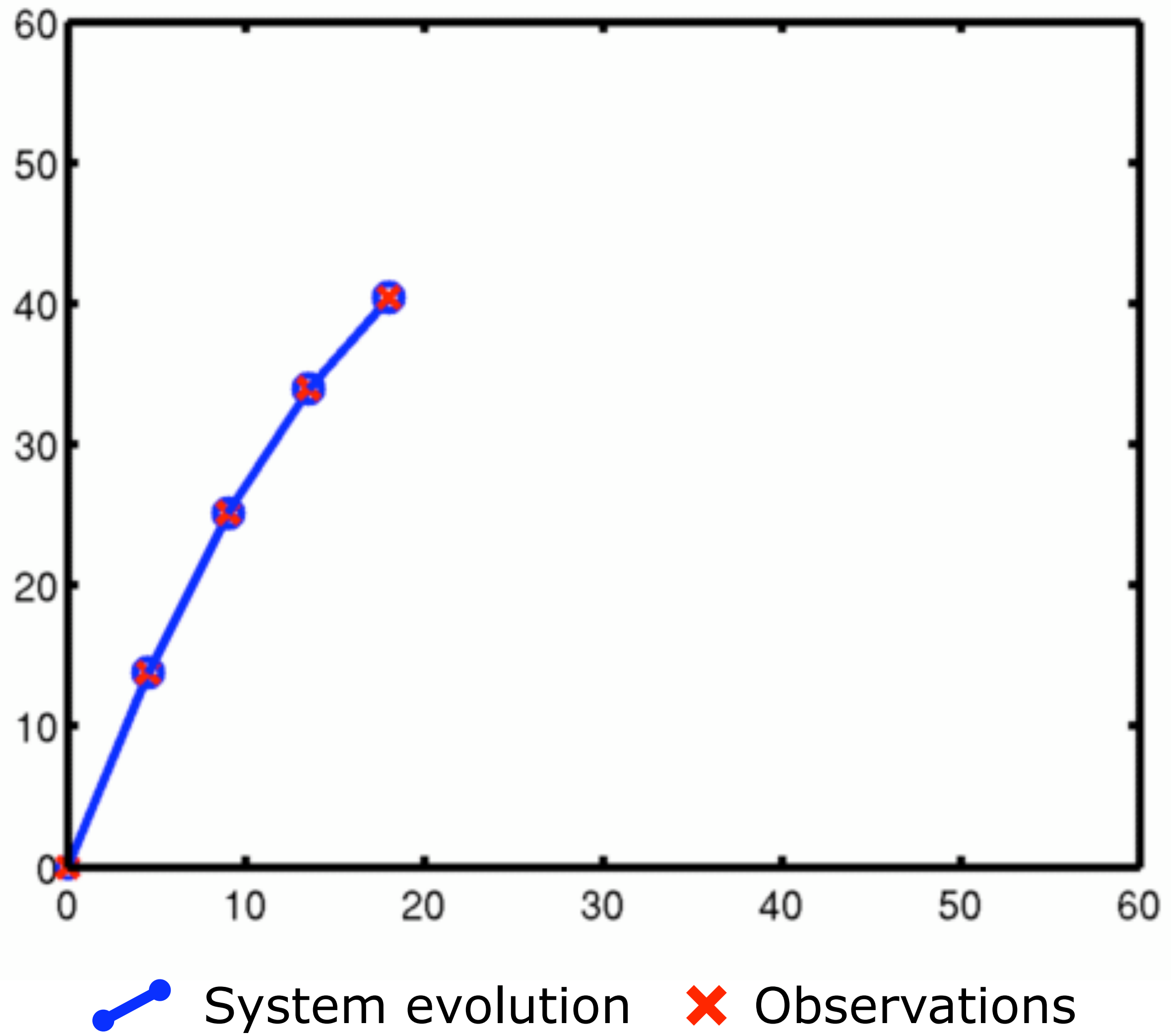
LDS Example – Throwing ball



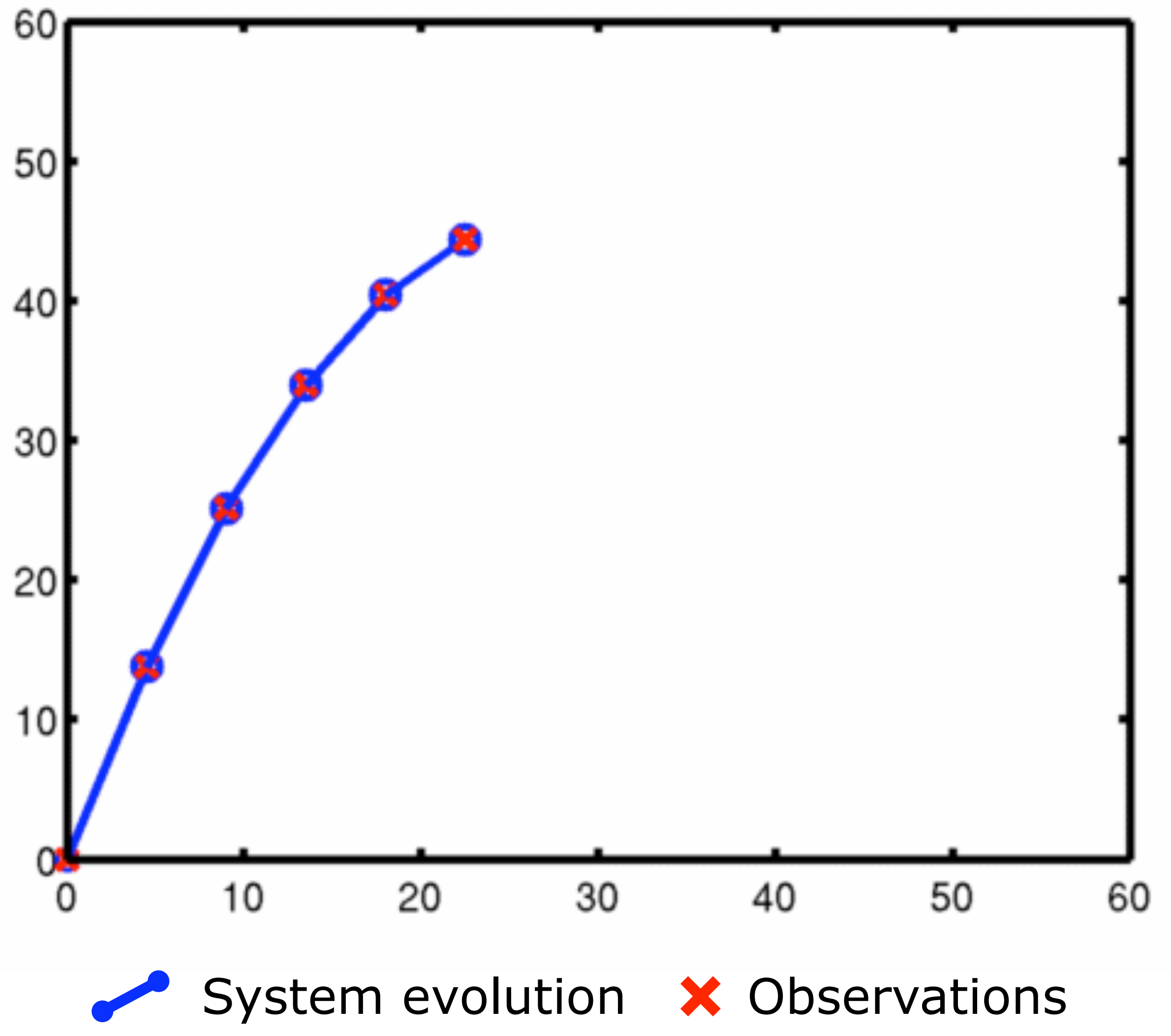
LDS Example – Throwing ball



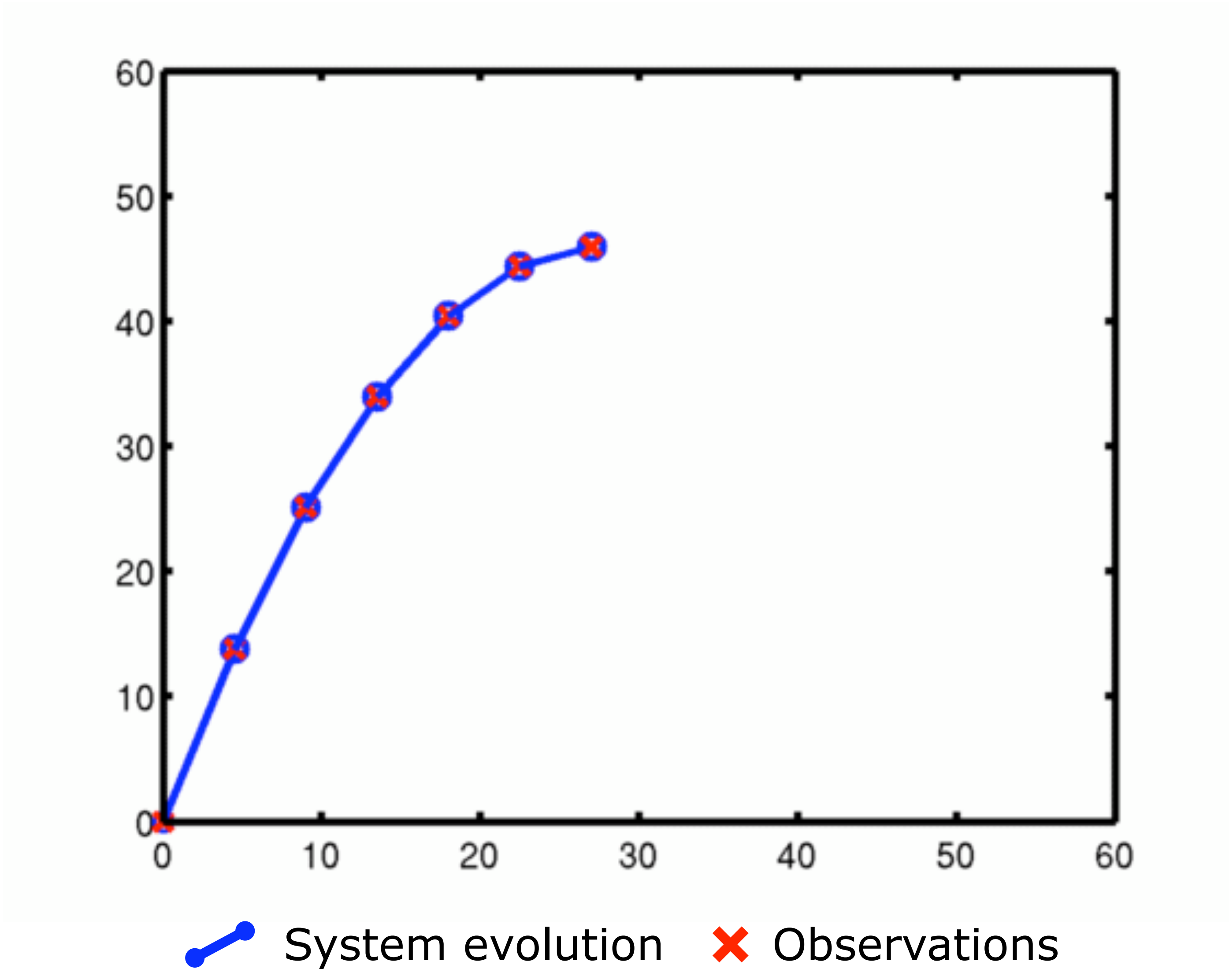
LDS Example – Throwing ball



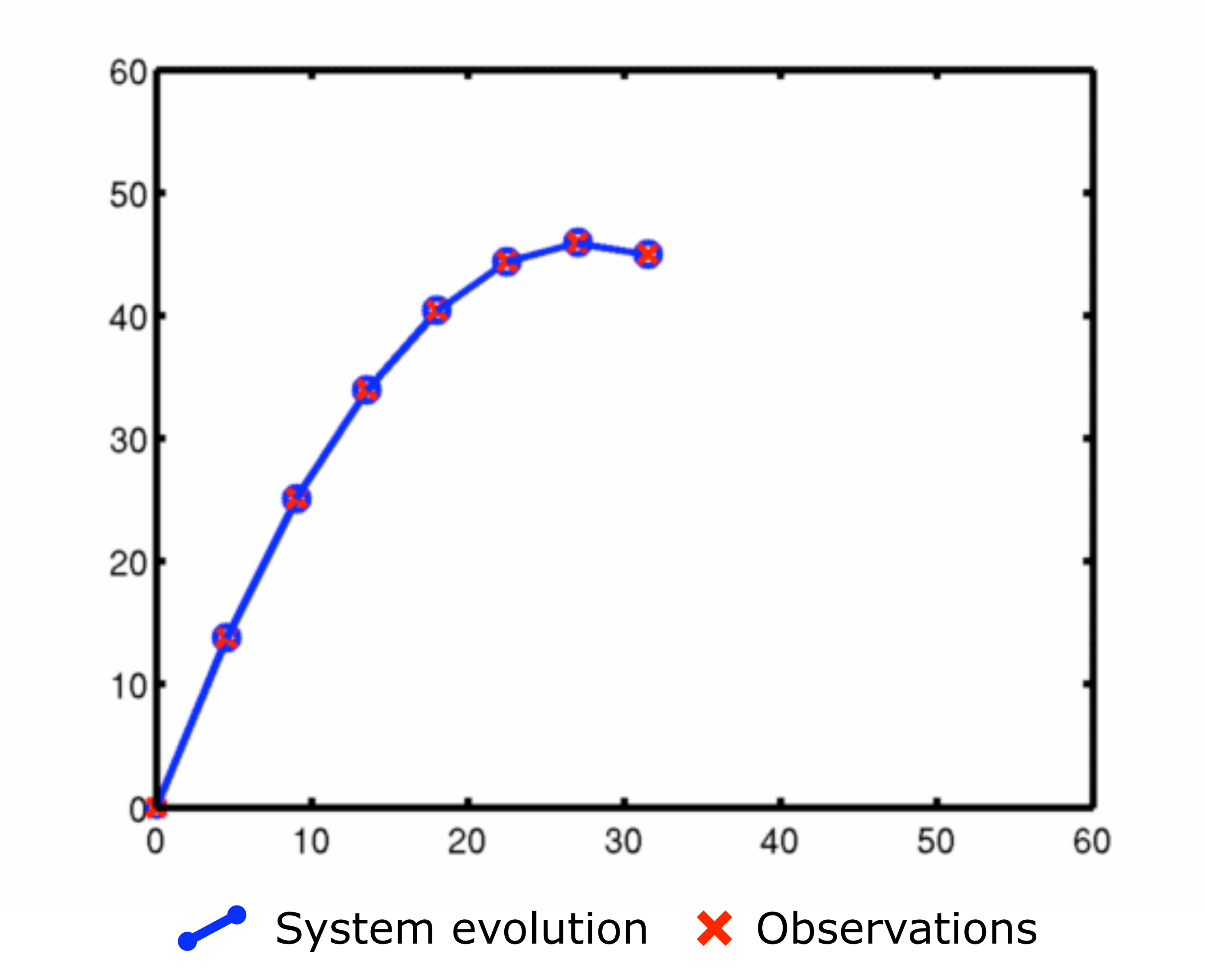
LDS Example – Throwing ball



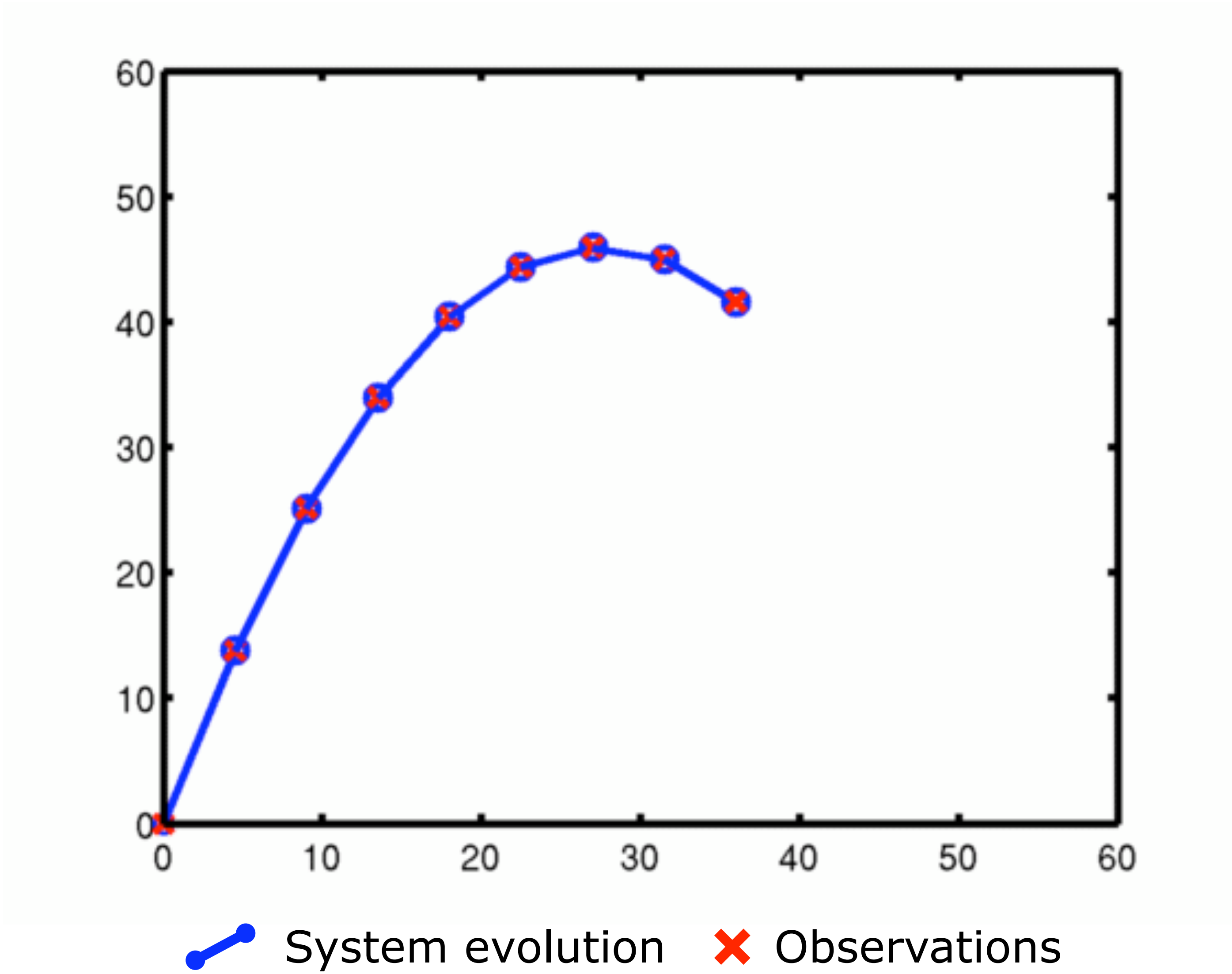
LDS Example – Throwing ball



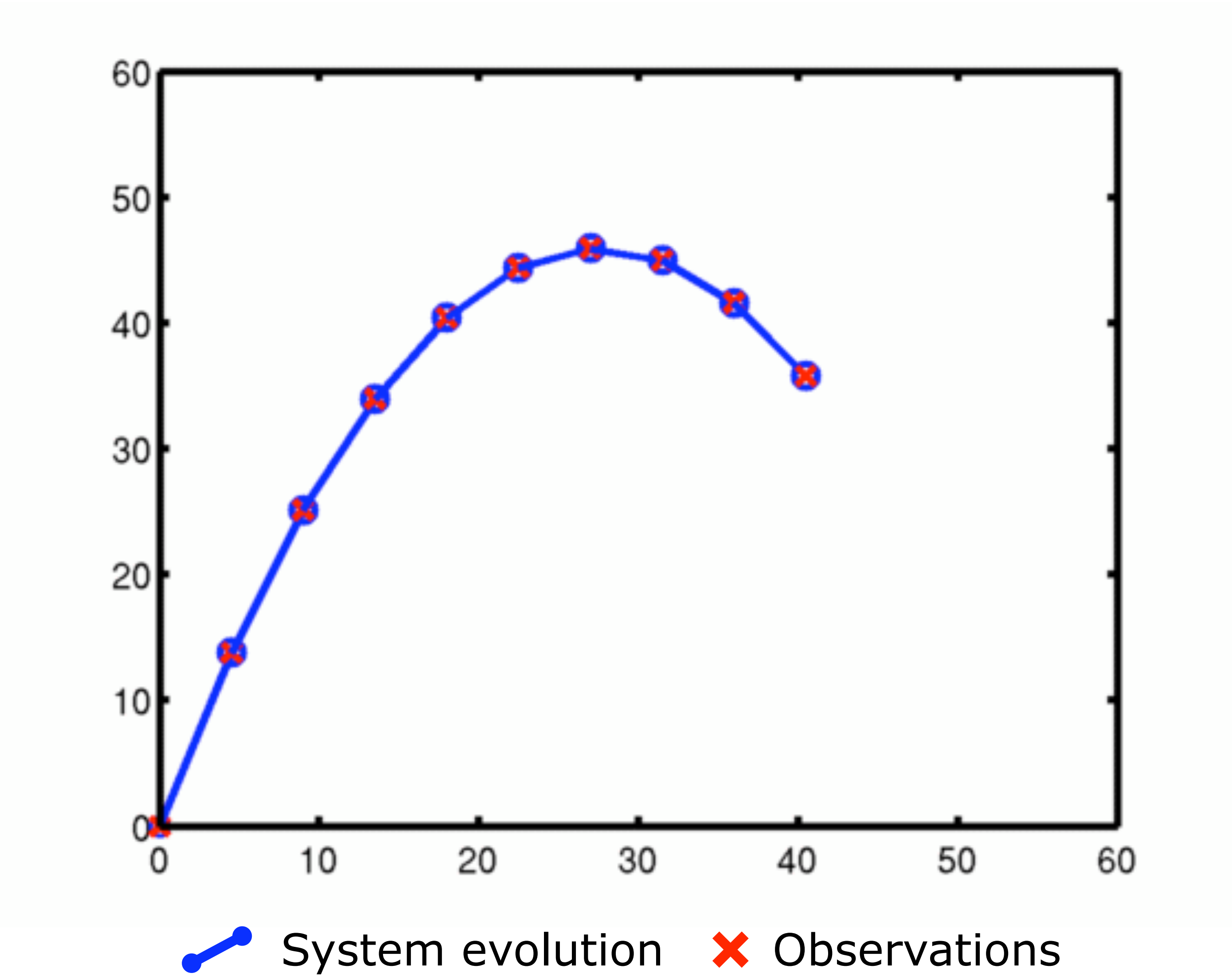
LDS Example – Throwing ball



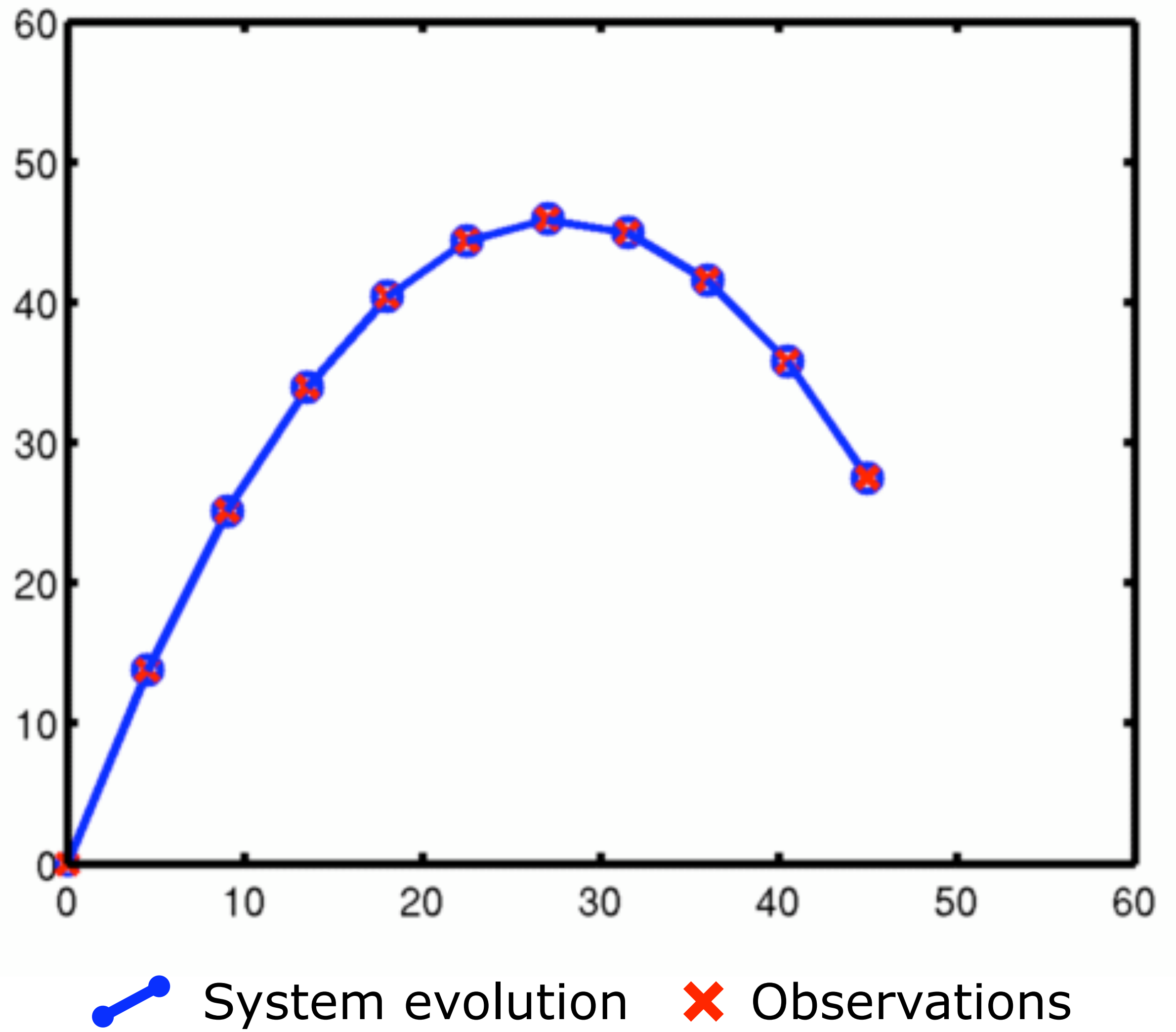
LDS Example – Throwing ball



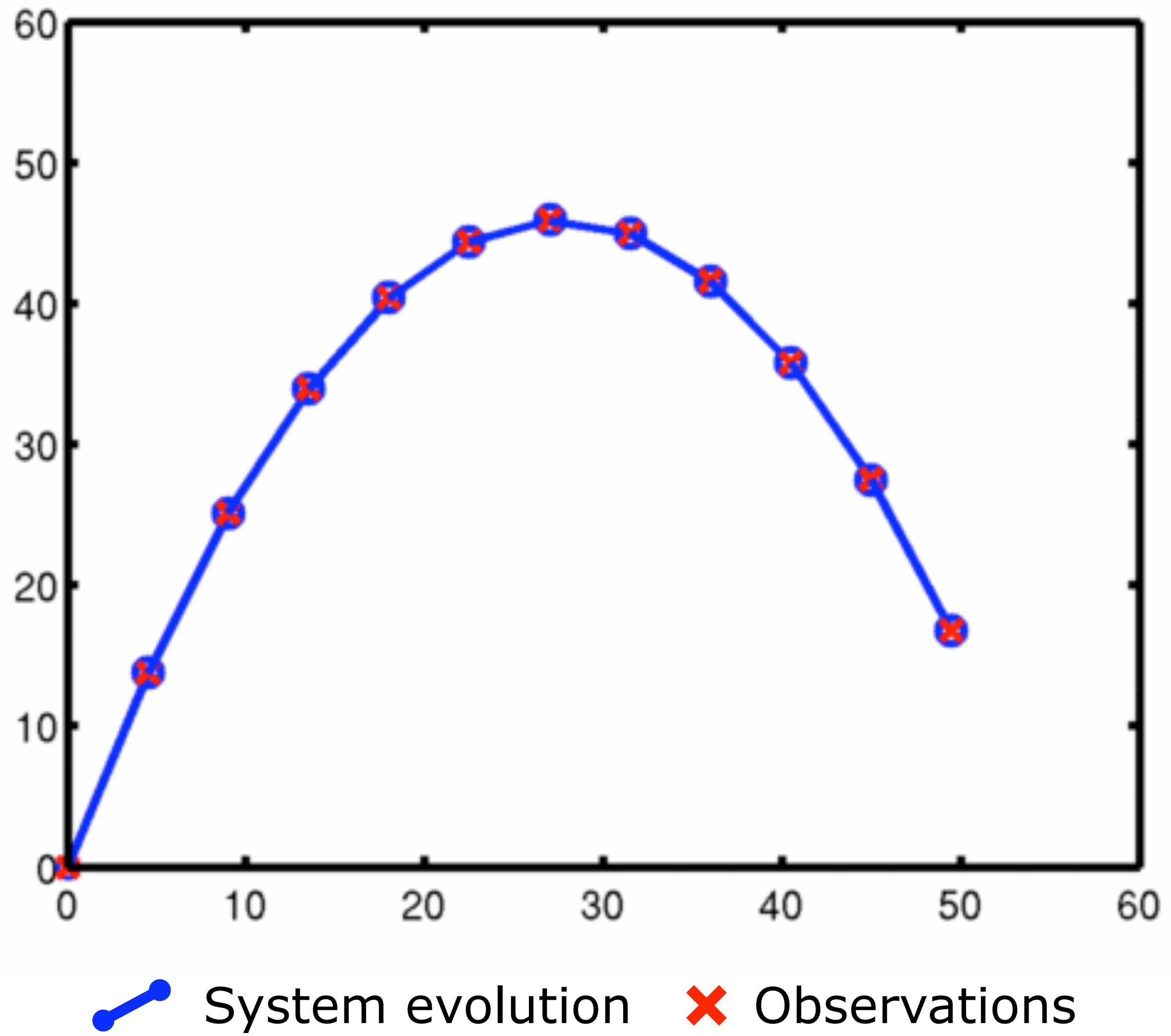
LDS Example – Throwing ball



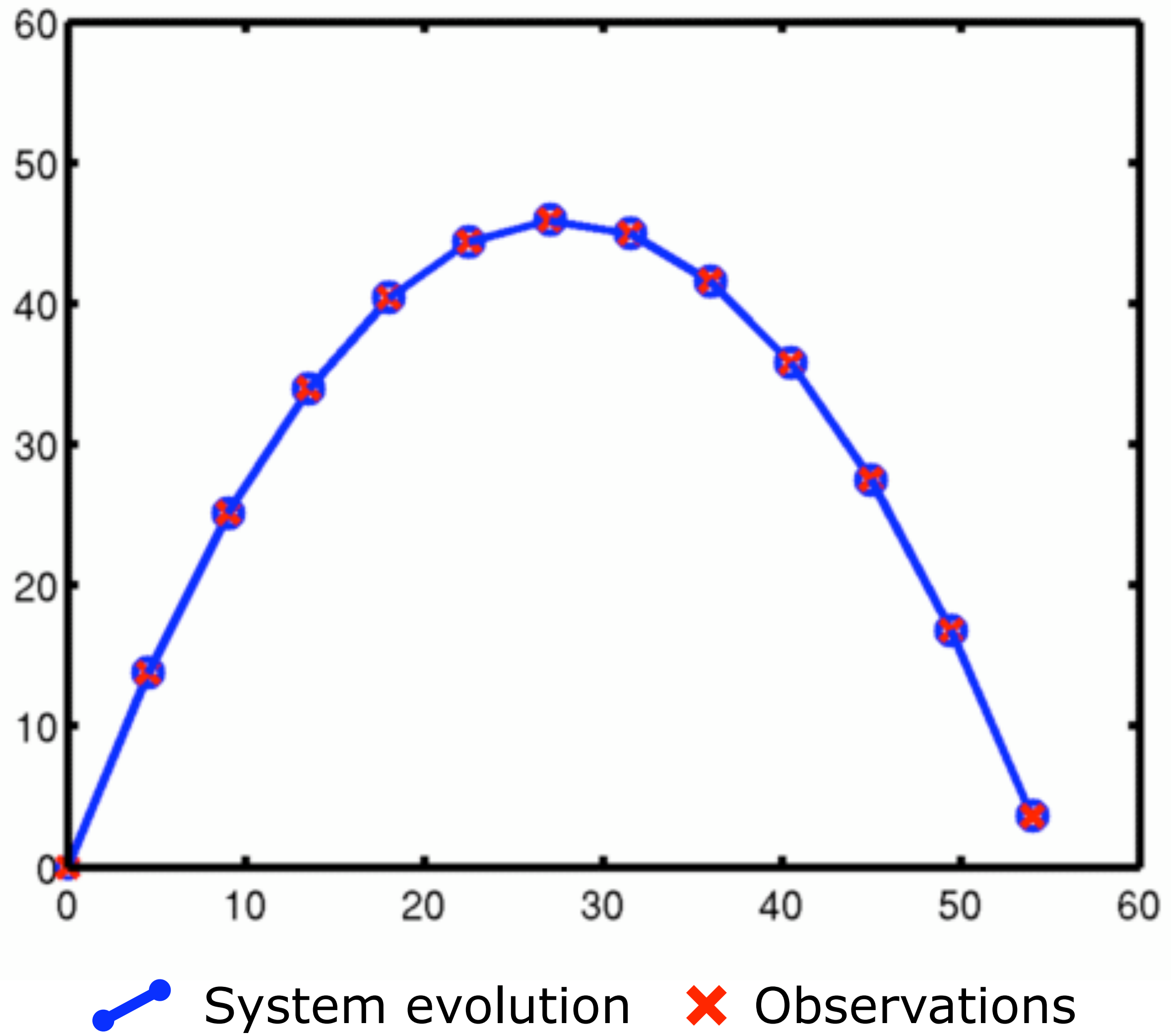
LDS Example – Throwing ball



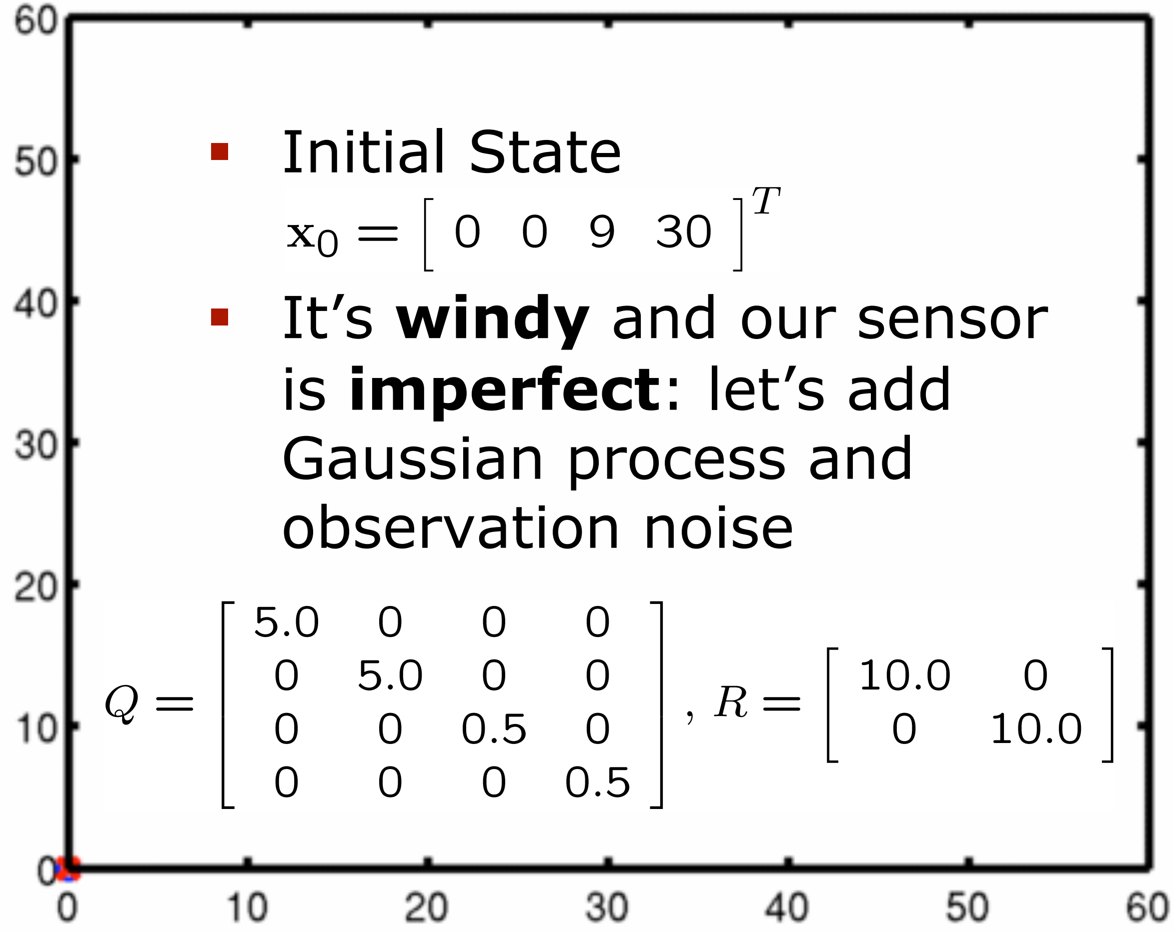
LDS Example – Throwing ball



LDS Example – Throwing ball



LDS Example – Throwing ball



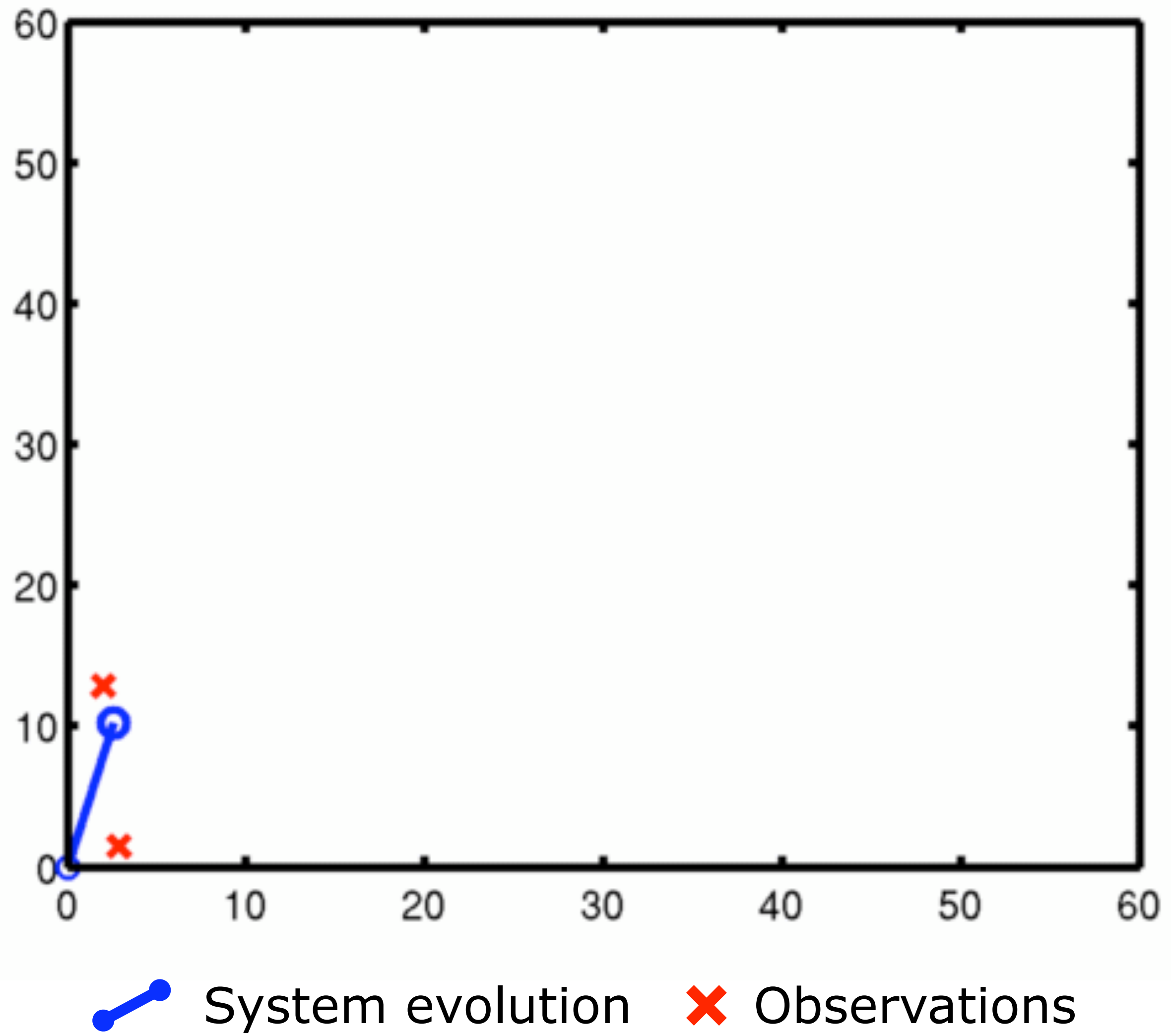
- Initial State

$$\mathbf{x}_0 = \begin{bmatrix} 0 & 0 & 9 & 30 \end{bmatrix}^T$$

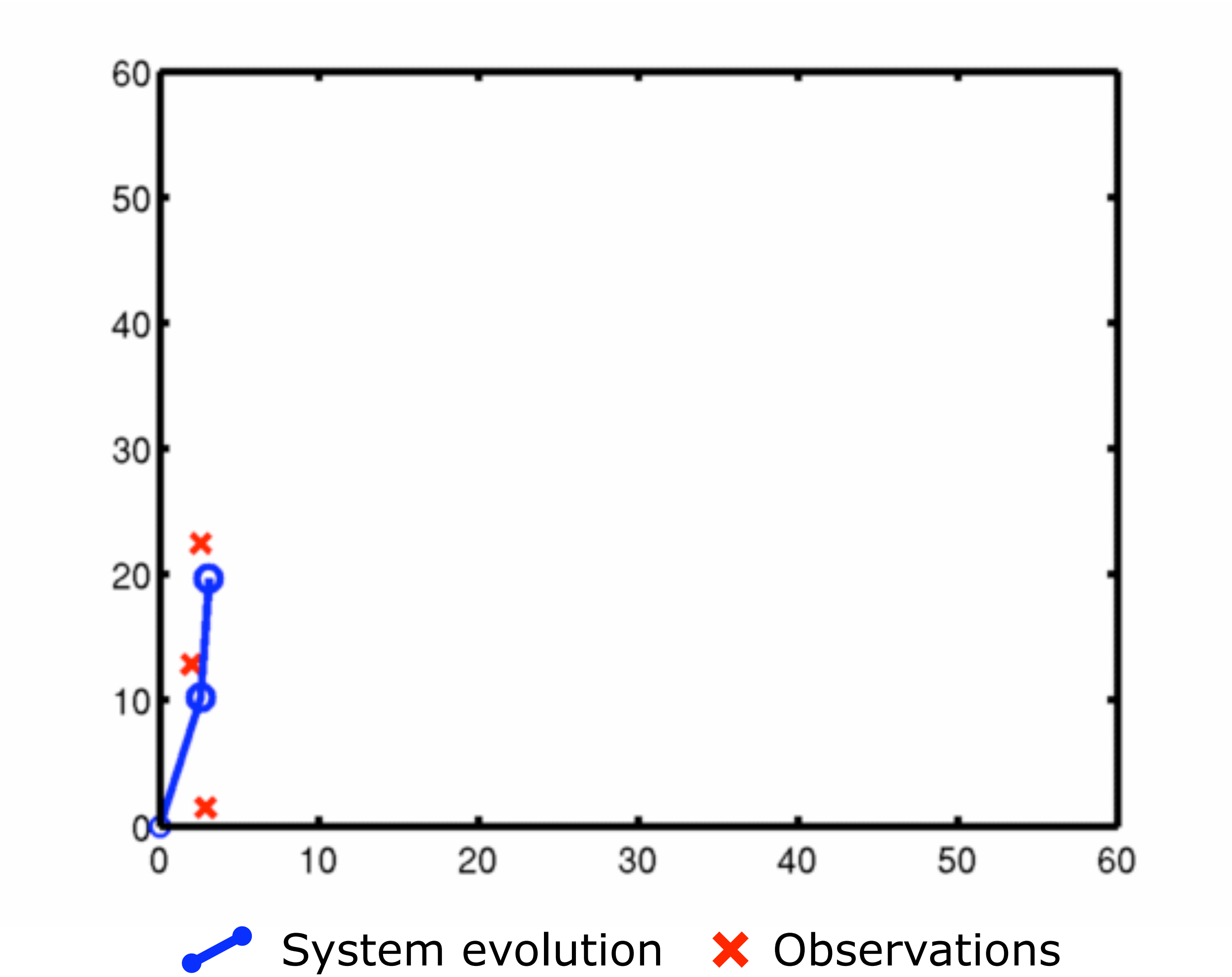
- It's **windy** and our sensor is **imperfect**: let's add Gaussian process and observation noise

$$Q = \begin{bmatrix} 5.0 & 0 & 0 & 0 \\ 0 & 5.0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}, \; R = \begin{bmatrix} 10.0 & 0 \\ 0 & 10.0 \end{bmatrix}$$

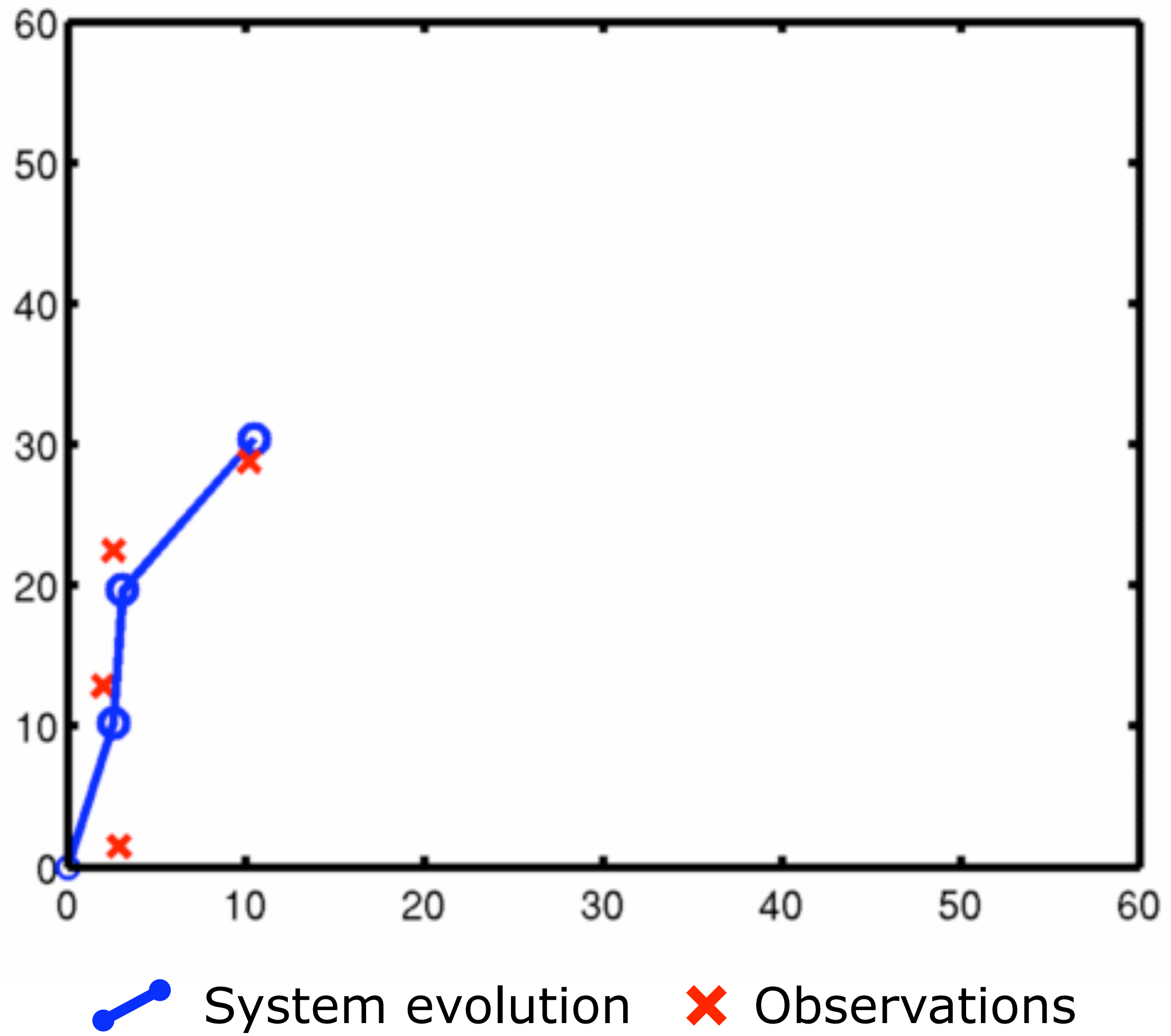
LDS Example – Throwing ball



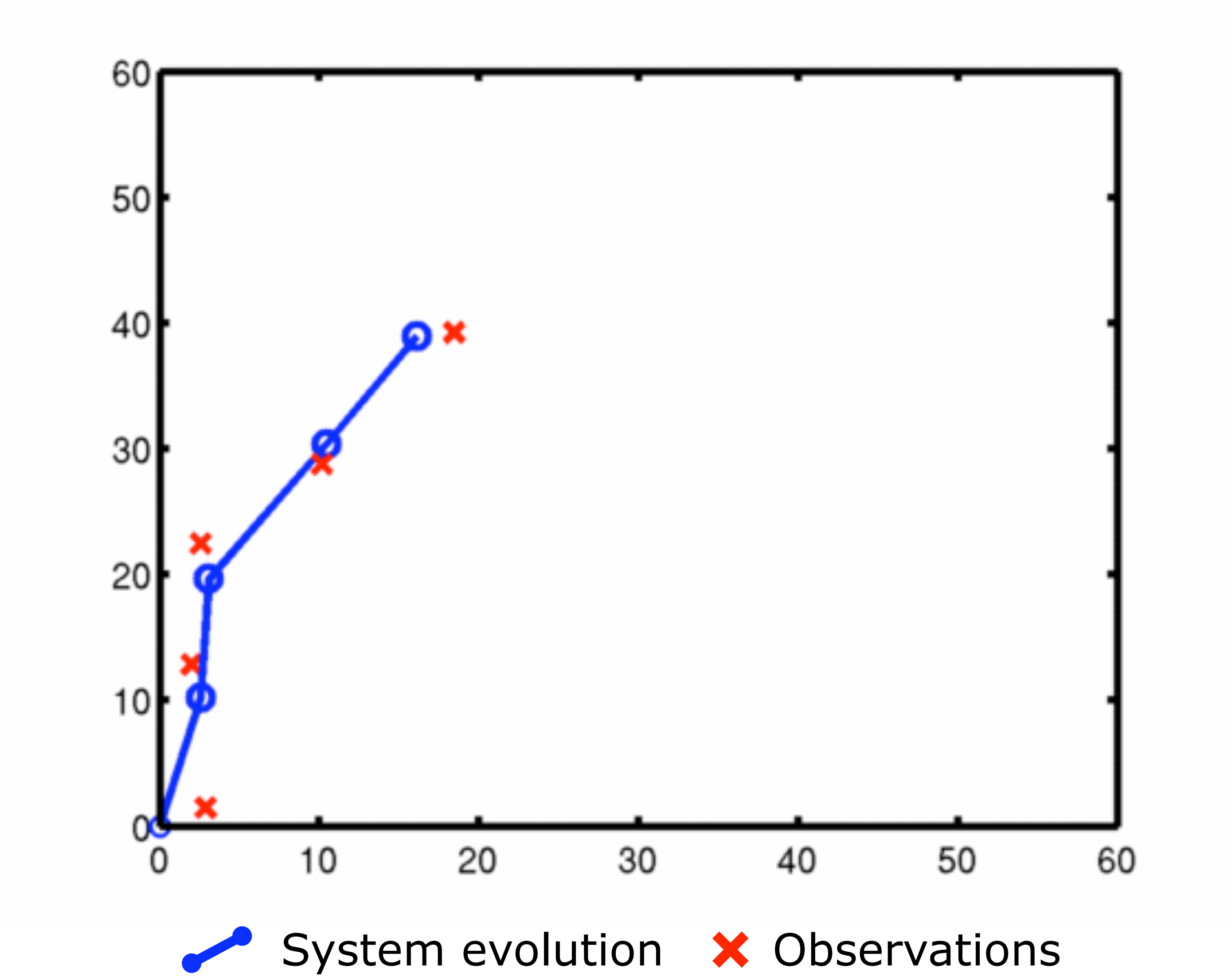
LDS Example – Throwing ball



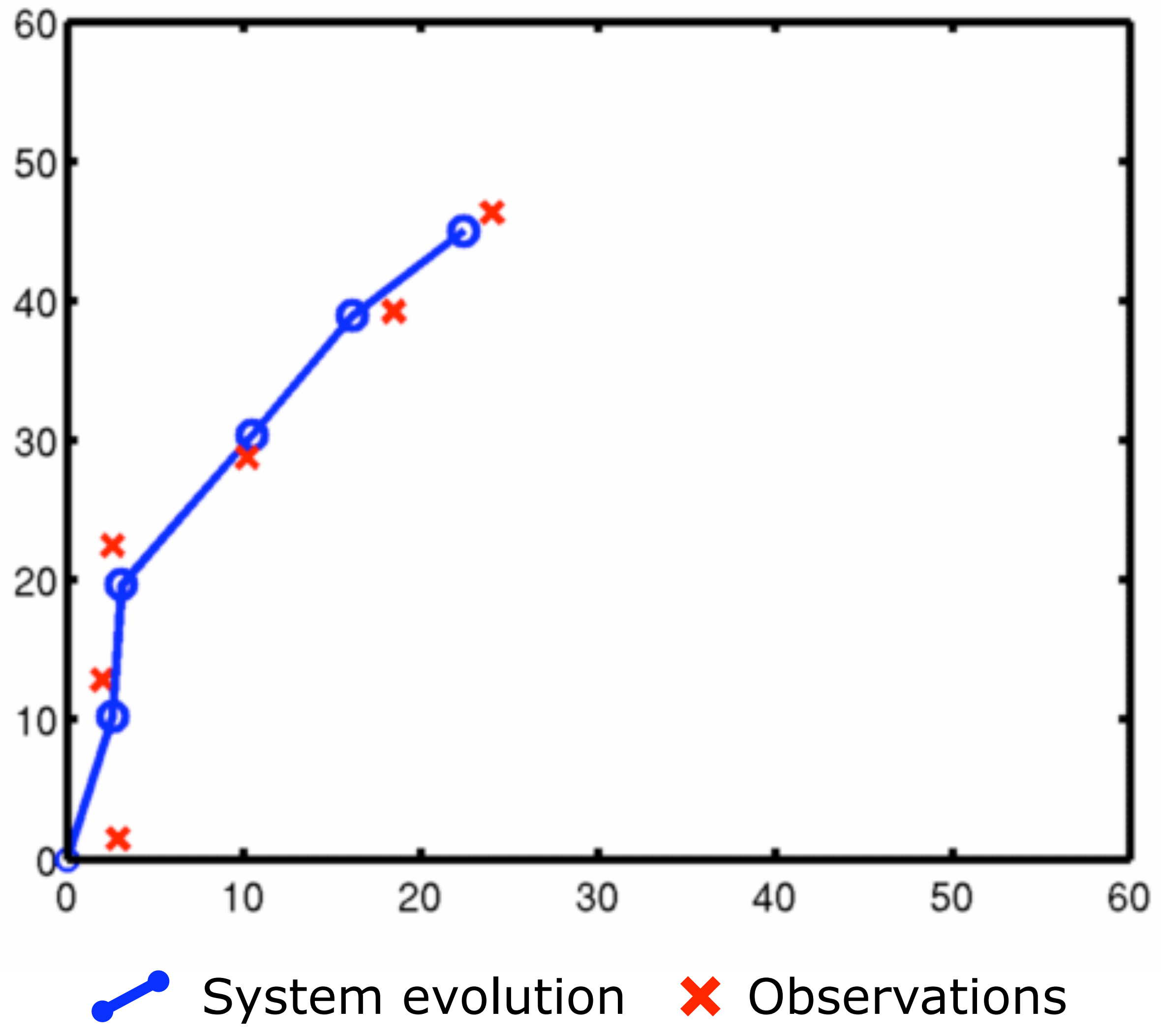
LDS Example – Throwing ball



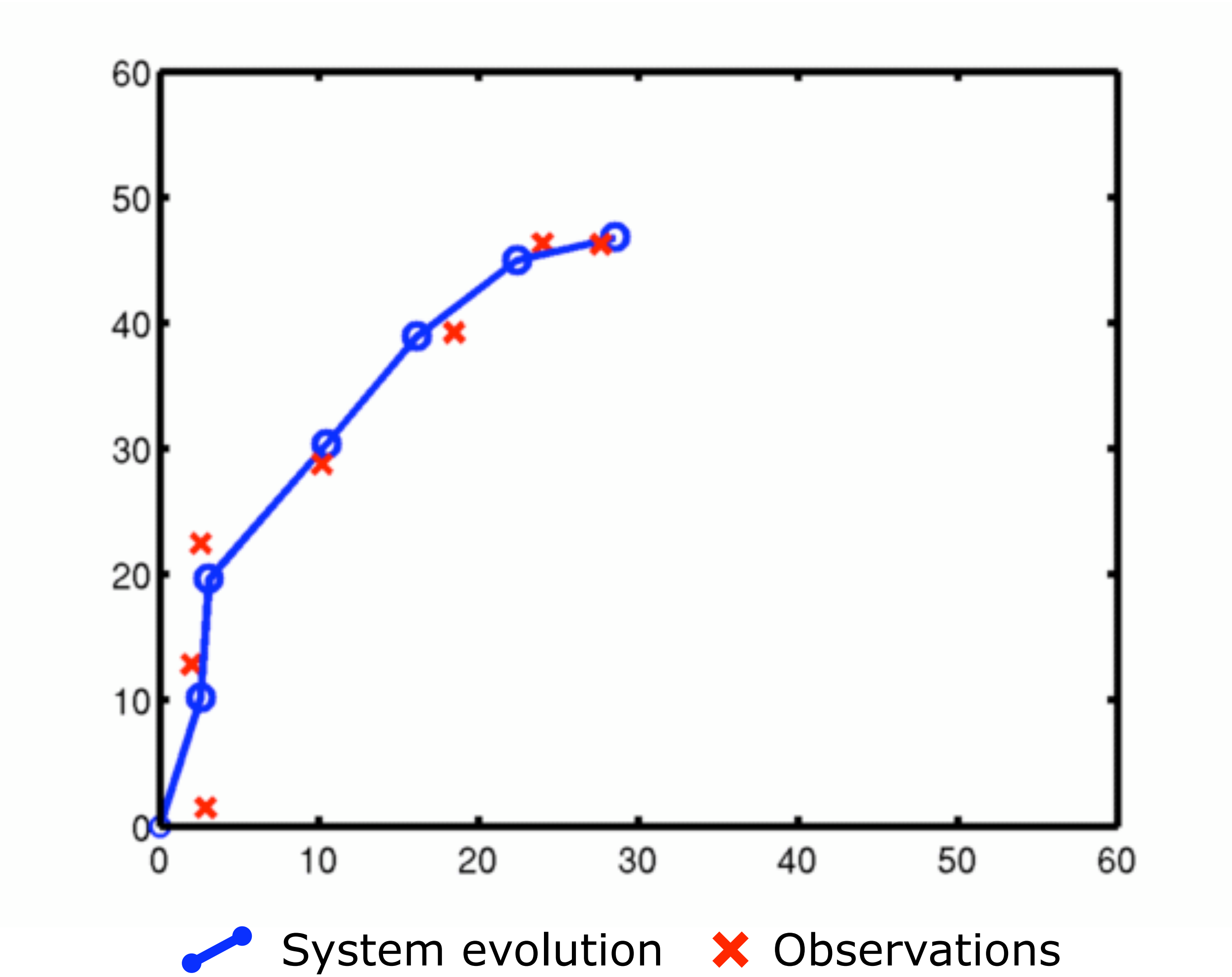
LDS Example – Throwing ball



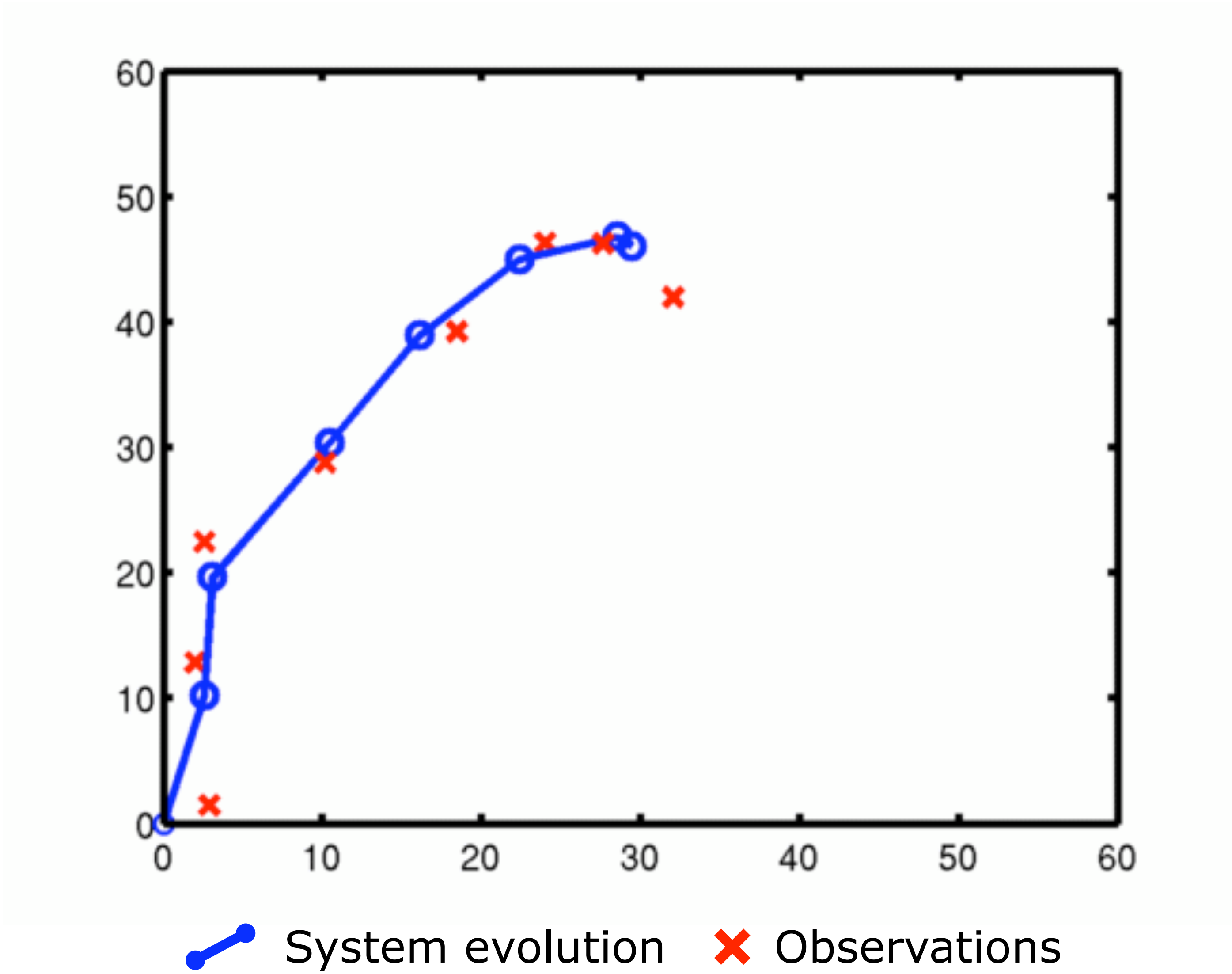
LDS Example – Throwing ball



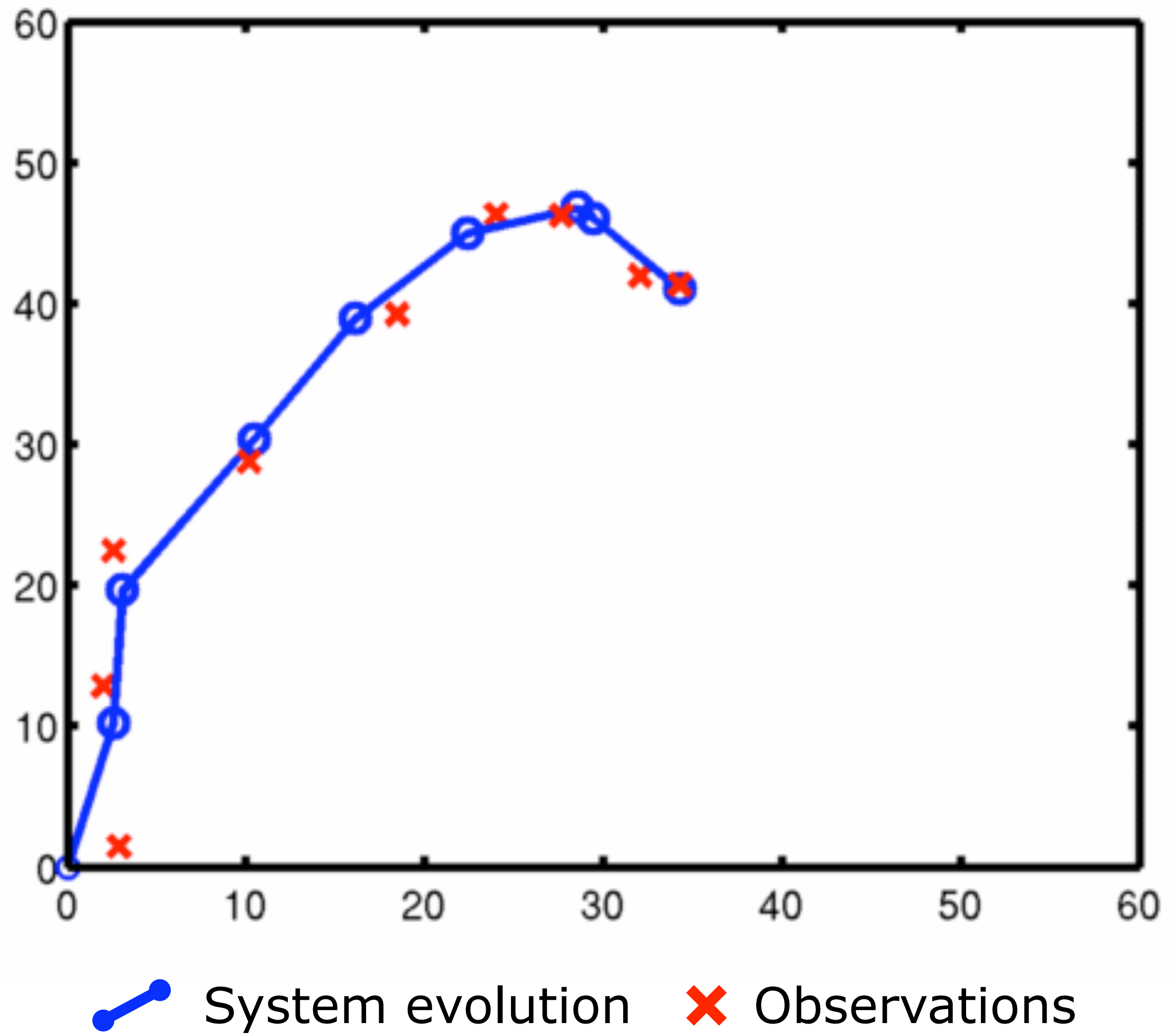
LDS Example – Throwing ball



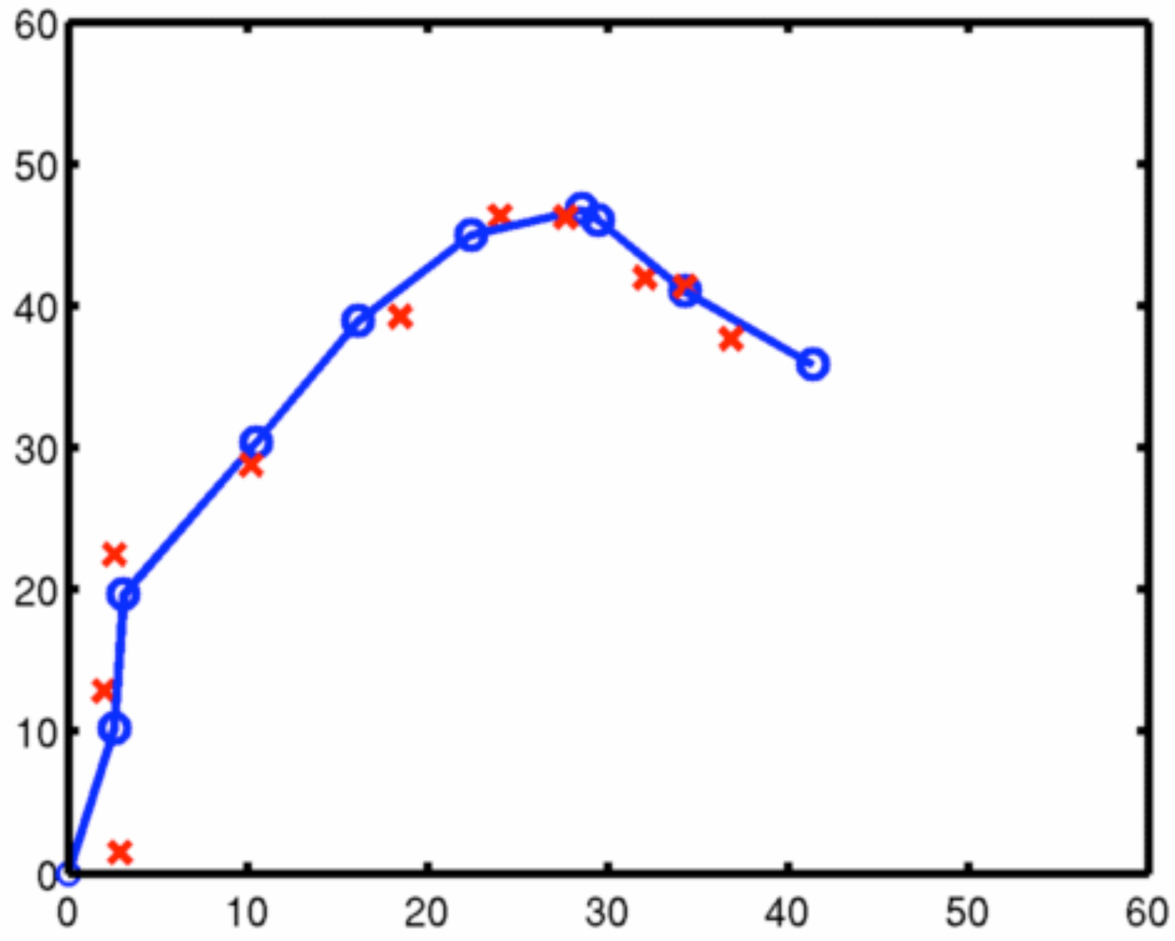
LDS Example – Throwing ball



LDS Example – Throwing ball

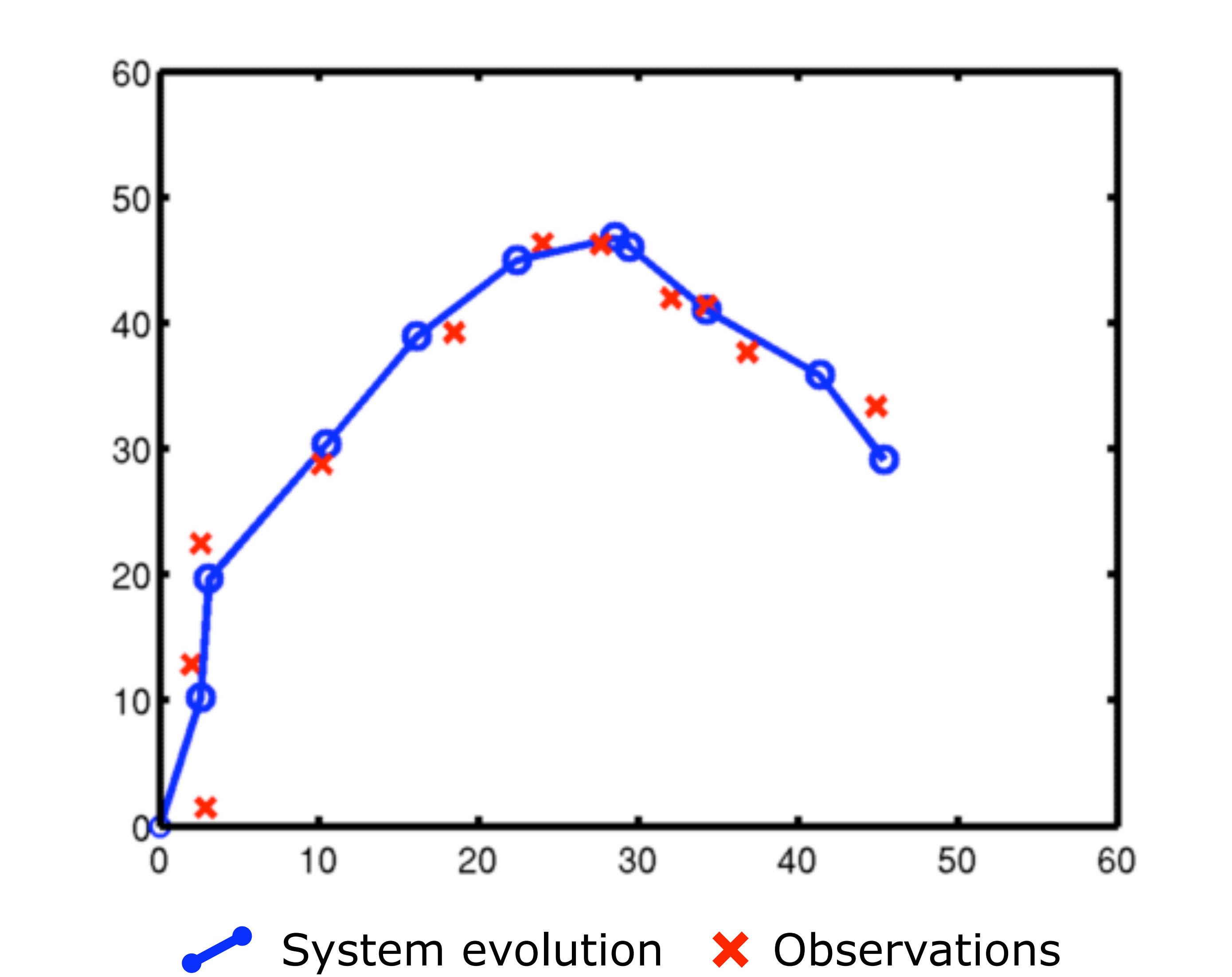


LDS Example – Throwing ball

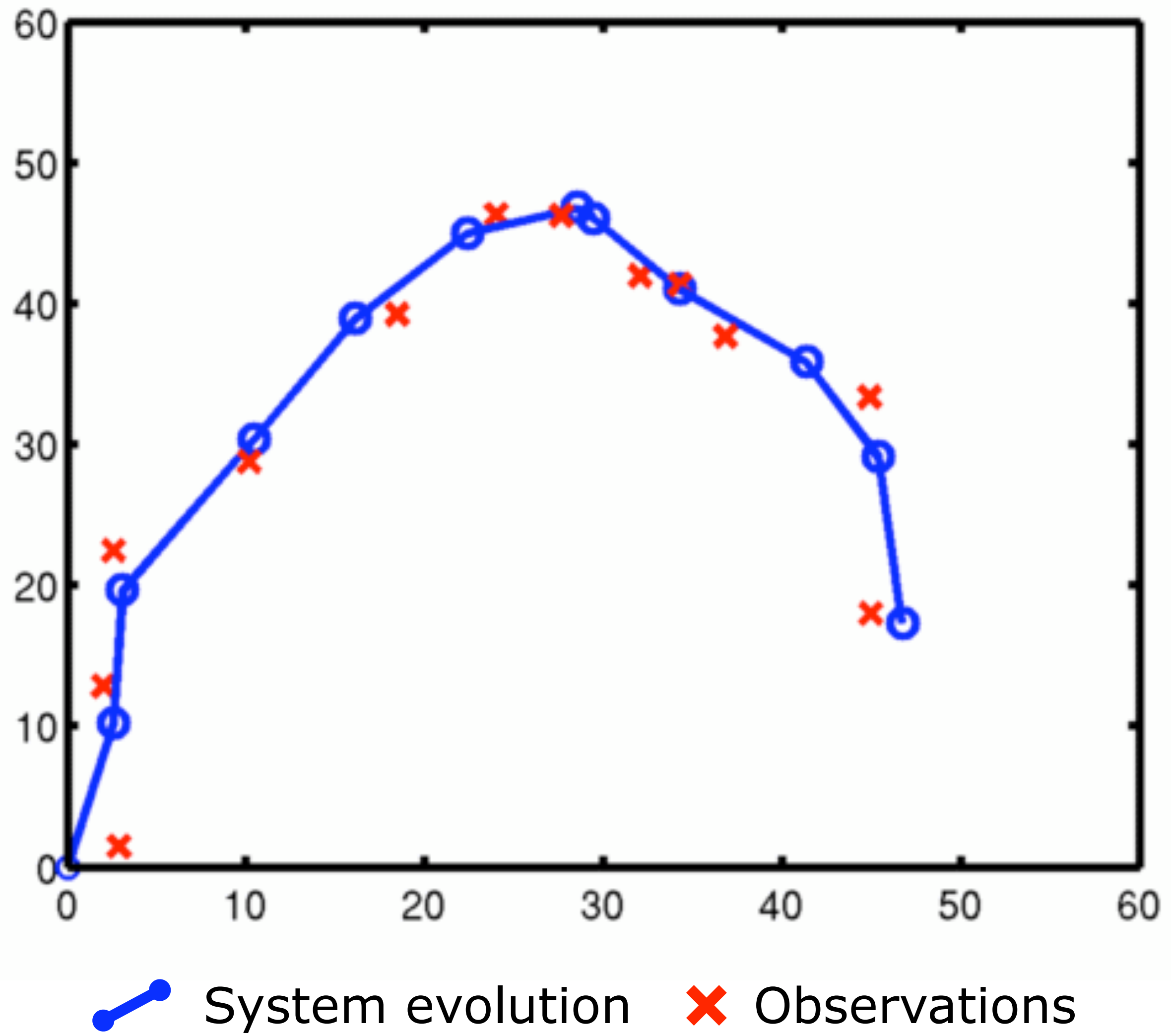


System evolution   Observations

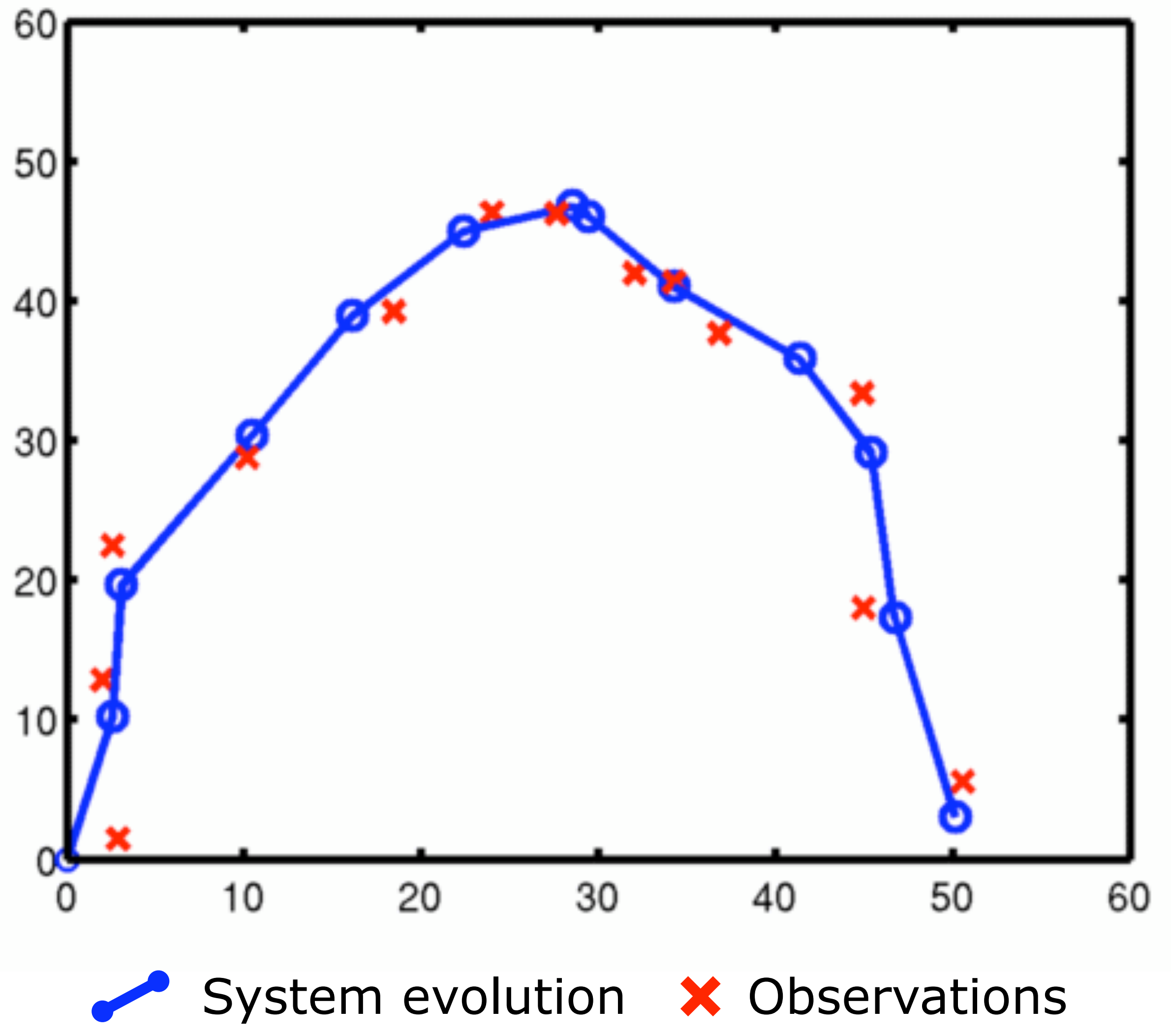
LDS Example – Throwing ball



LDS Example – Throwing ball



LDS Example – Throwing ball



Target Tracking

> "Tracking is the estimation of the state of a moving object based on remote measurements." [Bar-Shalom]

- **Detection is...**

  Knowing the presence of an object
- **Tracking is...**

  Maintaining a state of an object over time

> **Tracking** maintains the object's **state** and **identity** despite **detection errors** (false negatives, false alarms), **occlusions**, and in the presence of **other objects**

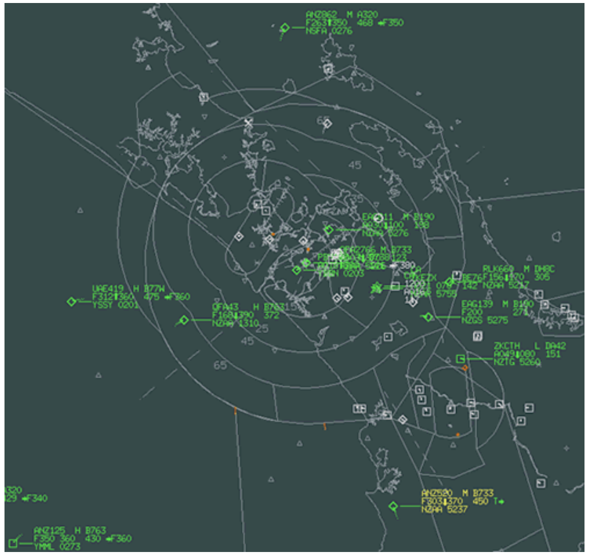
Target Tracking

Problem Statement

- **Given**
  - (Linear/nonlinear) dynamical system model
  - External measurements (from some sort of sensor)
- **Wanted**
  - System state estimate
    (e.g. position, velocity, acceleration, …)
- **Problems**
  - Track maintenance (i.e. creation, occlusion, deletion)
  - Multiple targets
  - Data association

Target Tracking

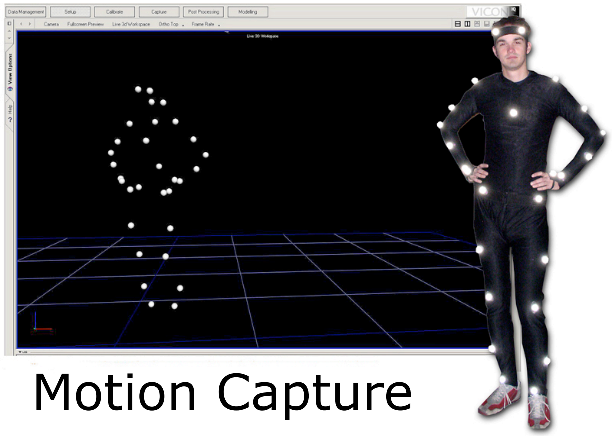
- **Long History, Many Applications**



Air Traffic Control

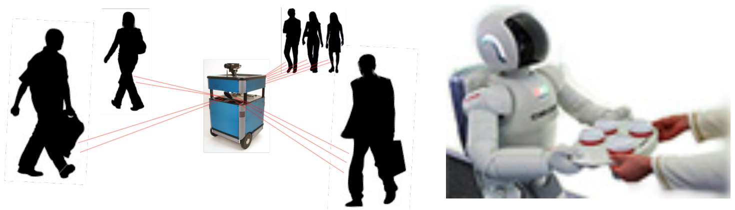
Fleet Management

Surveillance



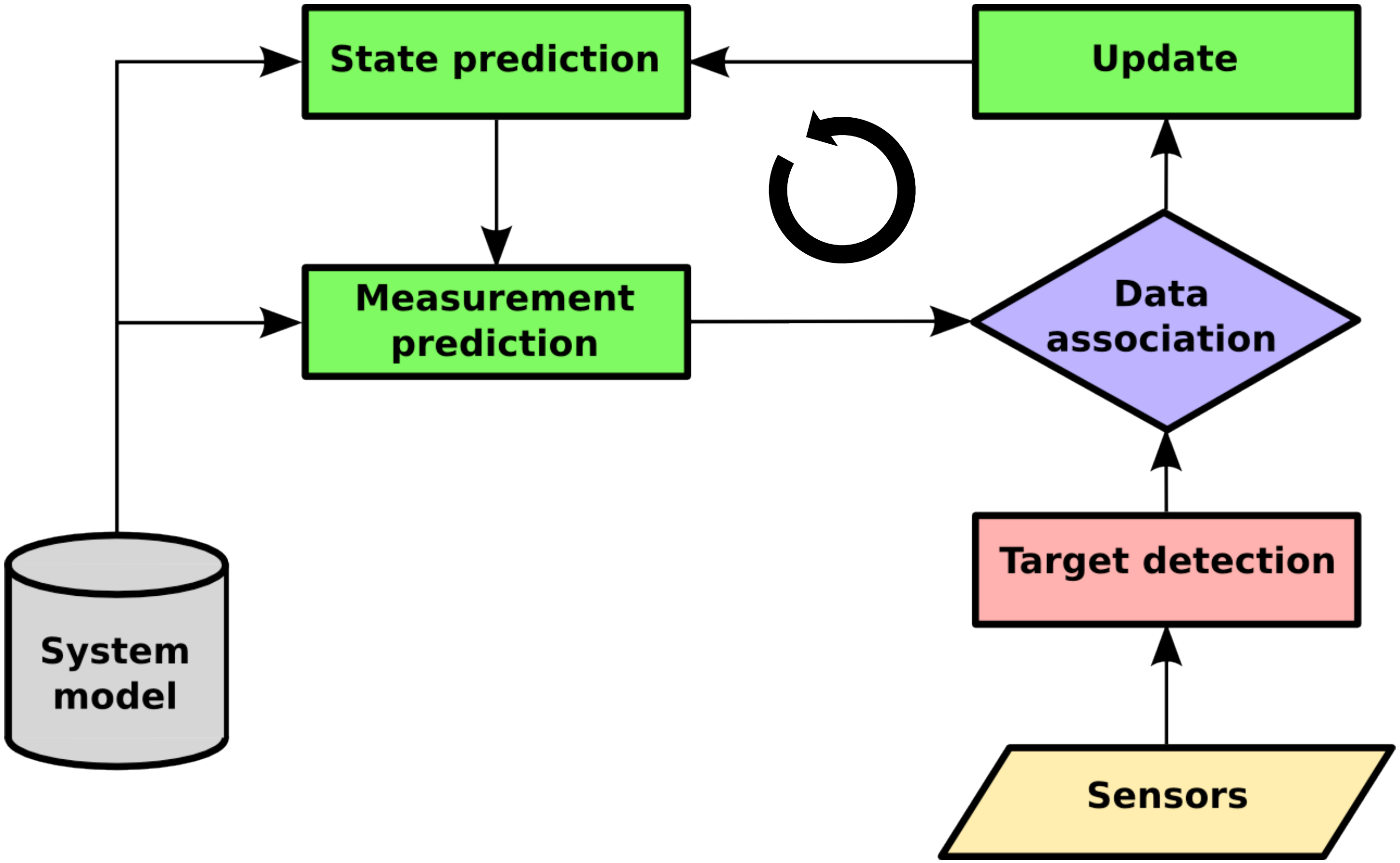
Motion Capture

Military Applications

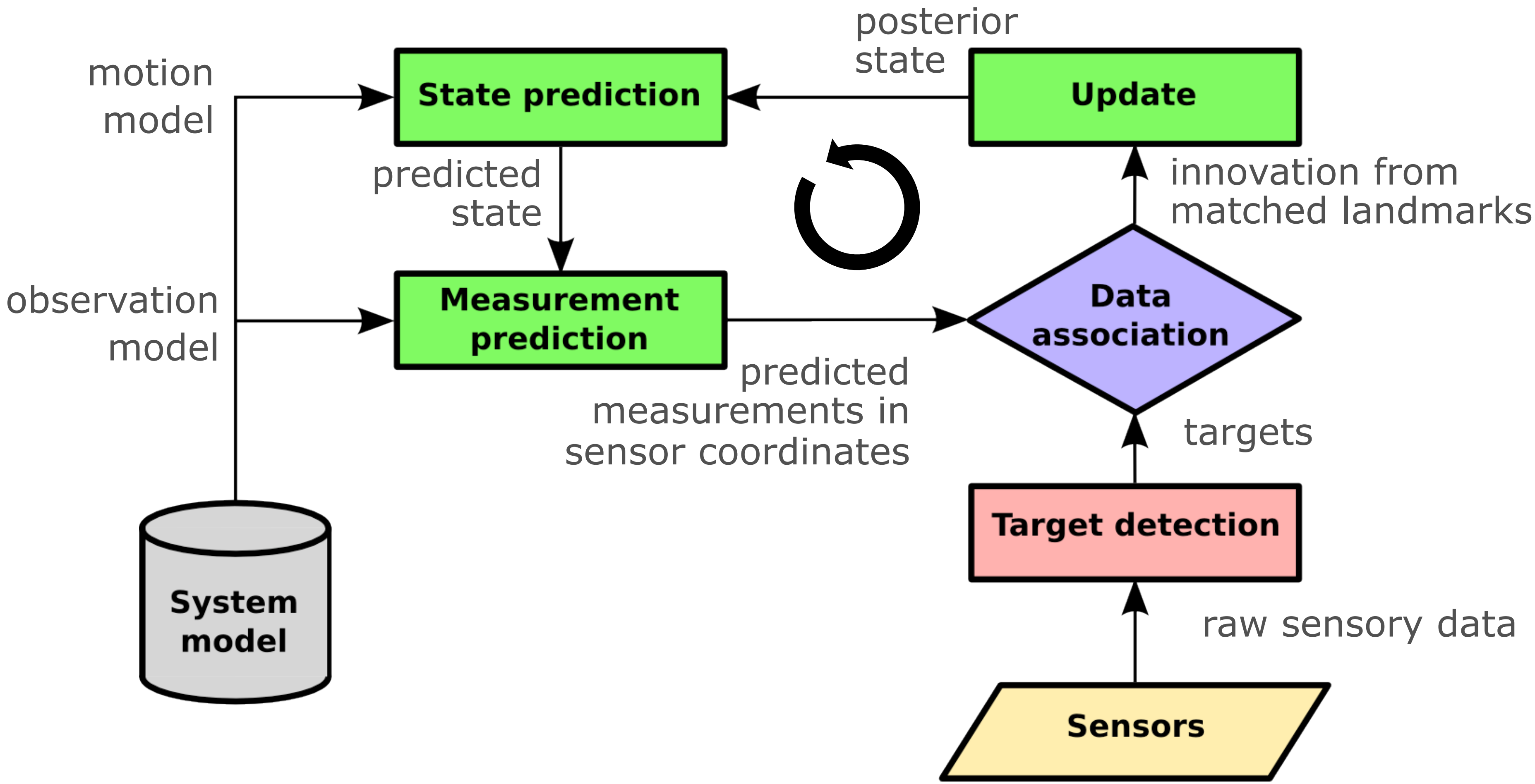


People Tracking, HRI

Tracking cycle



Tracking cycle – Kalman filter



Kalman Filter (KF)

- Consider a discrete time LDS with dynamic equation

$$x(k+1) = F(k)x(k) + \xi(k)$$

  where $\xi(k)$ is a process noise

$$\xi(k) \sim \mathcal{N}(0, Q(k))$$

- The measurement equation is

$$z(k) = H(k)x(k) + \epsilon(k)$$

  where $\epsilon(k)$ is a measurement noise

$$\epsilon(k) \sim \mathcal{N}(0, R(k))$$

- The initial state is generally unknown and modeled as a Gaussian random variable

$$\begin{aligned} \widehat{x}(0|0) &= x_0 \quad \textbf{State estimate} \\ \widehat{P}(0|0) &= P_0 \quad \textbf{Covariance estimate} \end{aligned}$$

KF Cycle: State prediction

- **State prediction**

$$\widehat{x}(k+1|k) = F(k)\widehat{x}(k|k)$$
$$\widehat{P}(k+1|k) = F(k)\widehat{P}(k|k)F^T(k) + Q(k)$$

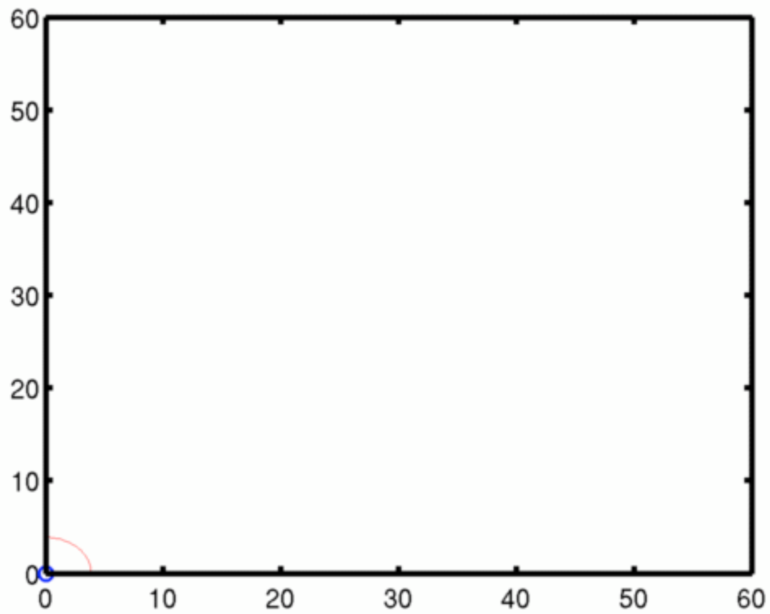
- In target tracking, **no a priori knowledge** of the dynamic equation is generally available
- Instead, different **target motion models** are used
  - Brownian motion model
  - Constant velocity model
  - Constant acceleration model
  - More advanced models (problem related)

Motion Models: Brownian

- **No motion assumption**
- Useful to describe stop-and-go motion behavior

- State representation
  $$\mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^T$$
- Initial state
  $$\mathbf{x}_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$$
- Transition matrix
  $$F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Ball example

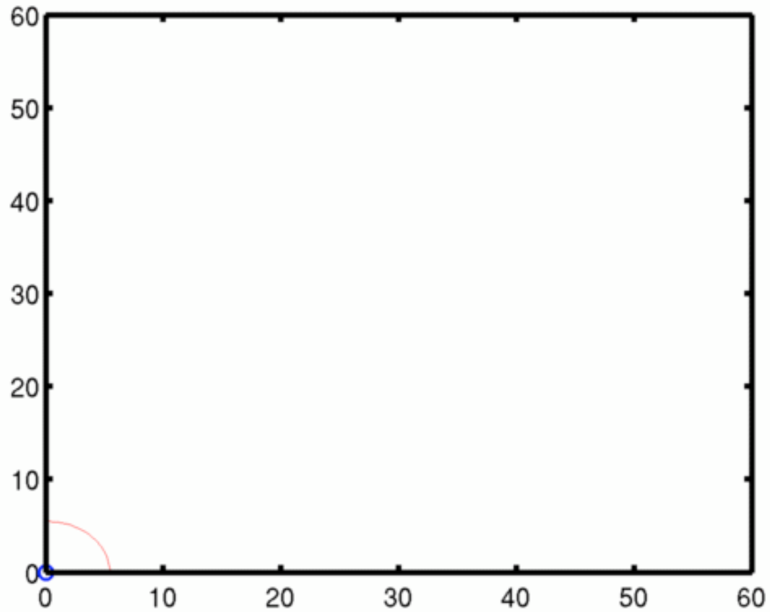


Motion Models: Brownian

- **No motion assumption**
- Useful to describe stop-and-go motion behavior

- State representation
  $$\mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^T$$
- Initial state
  $$\mathbf{x}_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$$
- Transition matrix
  $$F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Ball example



State prediction:

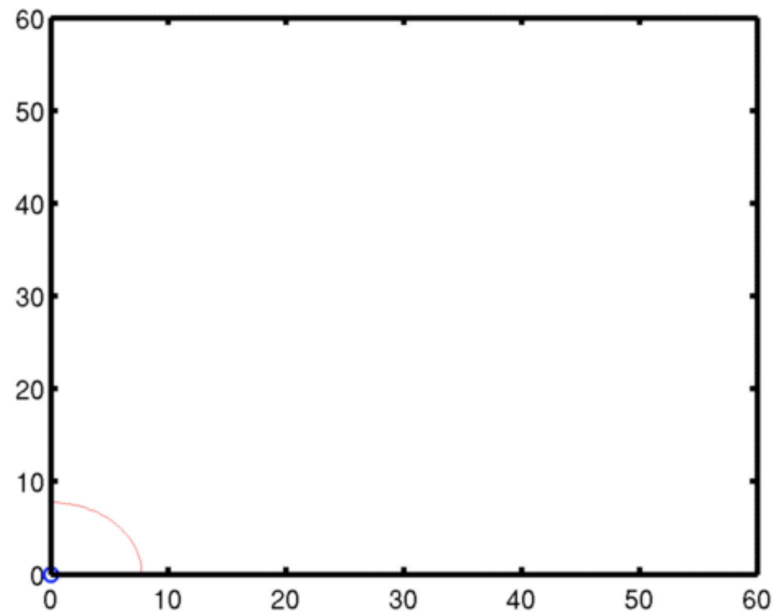
- **Uncertainty grows**

Motion Models: Brownian

- **No motion assumption**
- Useful to describe stop-and-go motion behavior

- State representation
  $$\mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^T$$
- Initial state
  $$\mathbf{x}_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$$
- Transition matrix
  $$F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Ball example



State prediction:

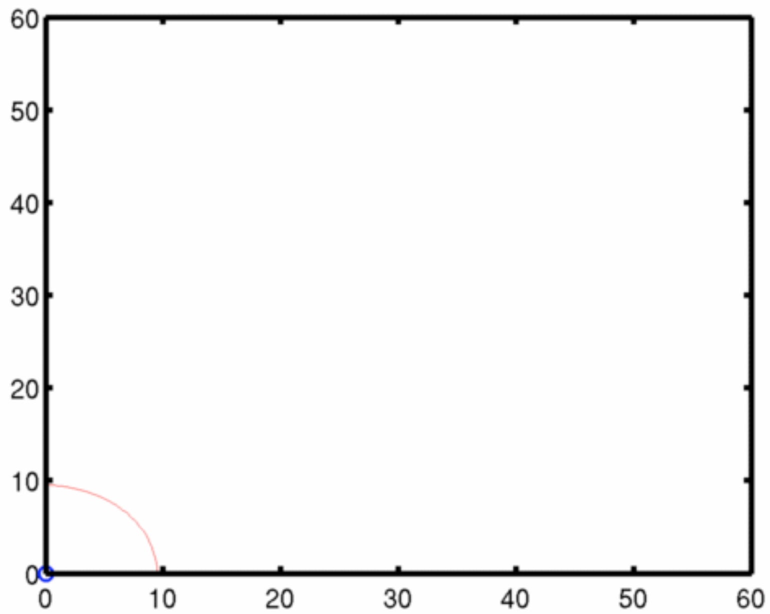
- **Uncertainty grows**

Motion Models: Brownian

- **No motion assumption**
- Useful to describe stop-and-go motion behavior

- State representation
  $\mathbf{x} = \left[\begin{array}{cc} x & y \end{array} \right]^T$
- Initial state
  $\mathbf{x}_0 = \left[\begin{array}{cc} 0 & 0 \end{array} \right]^T$
- Transition matrix
  $$F = \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right]$$

Ball example



State prediction:

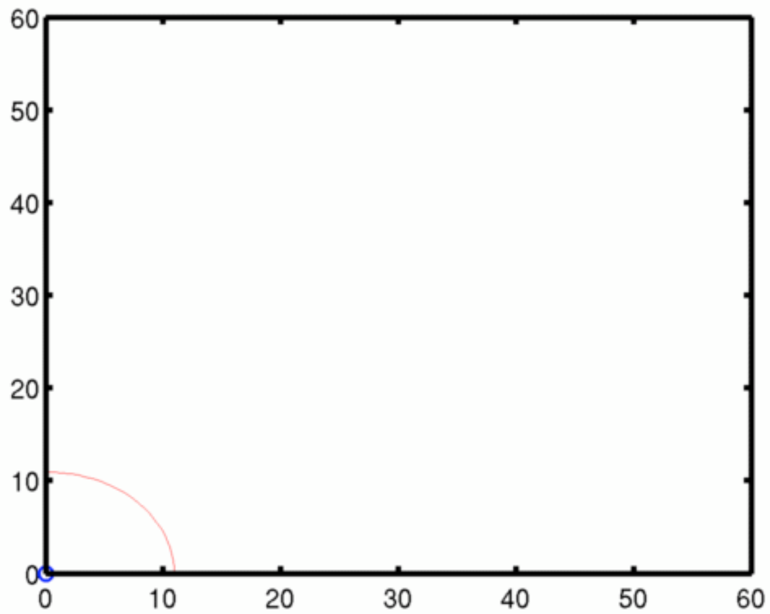
- **Uncertainty grows**

Motion Models: Brownian

- **No motion assumption**
- Useful to describe stop-and-go motion behavior

- State representation
  $$\mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^T$$
- Initial state
  $$\mathbf{x}_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$$
- Transition matrix
  $$F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Ball example



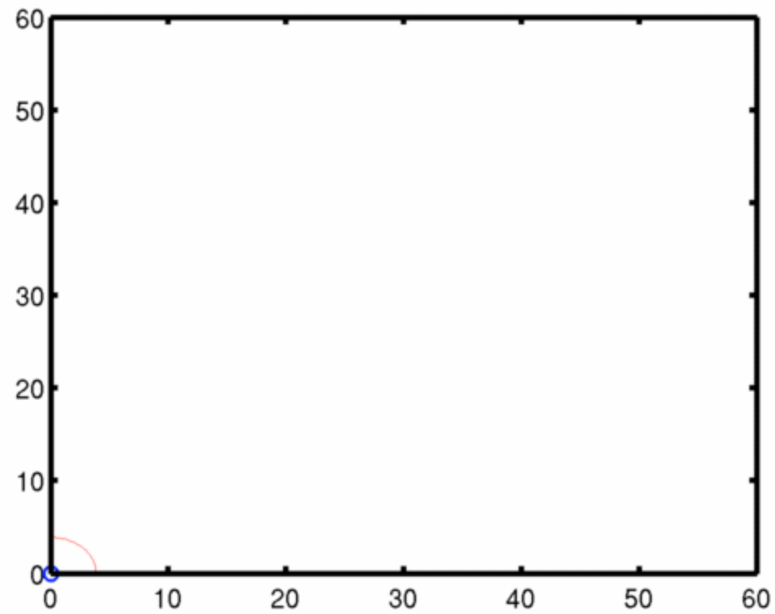
State prediction:
- **Uncertainty grows**

MMs: Constant Velocity

- **Constant target velocity** assumption
- Useful to model smooth target motion

- State representation
$$\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} \end{bmatrix}^T$$
- Initial state
$$\mathbf{x} = \begin{bmatrix} 0 & 0 & 9 & 30 \end{bmatrix}^T$$
- Transition matrix
$$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ball example



MMs: Constant Velocity

- **Constant target velocity** assumption
- Useful to model smooth target motion

- State representation

  $$\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} \end{bmatrix}^T$$

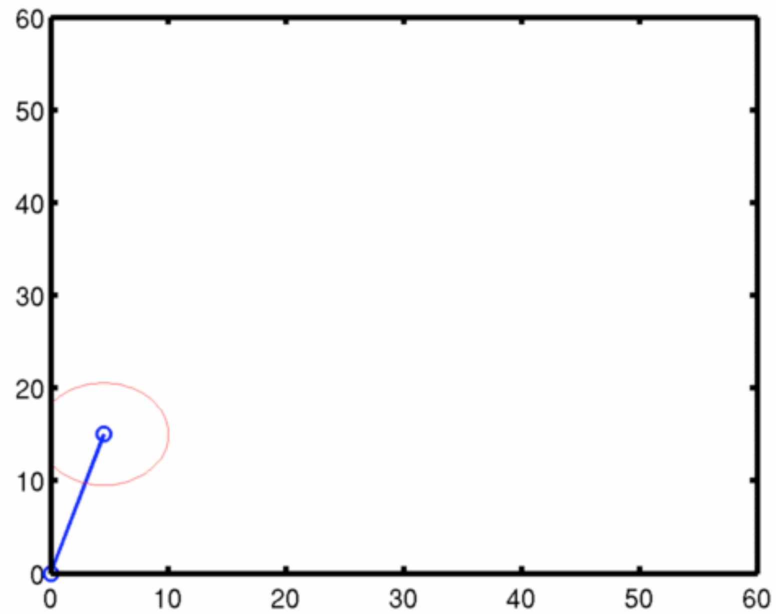
- Initial state

  $$\mathbf{x} = \begin{bmatrix} 0 & 0 & 9 & 30 \end{bmatrix}^T$$

- Transition matrix

$$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ball example



State prediction:

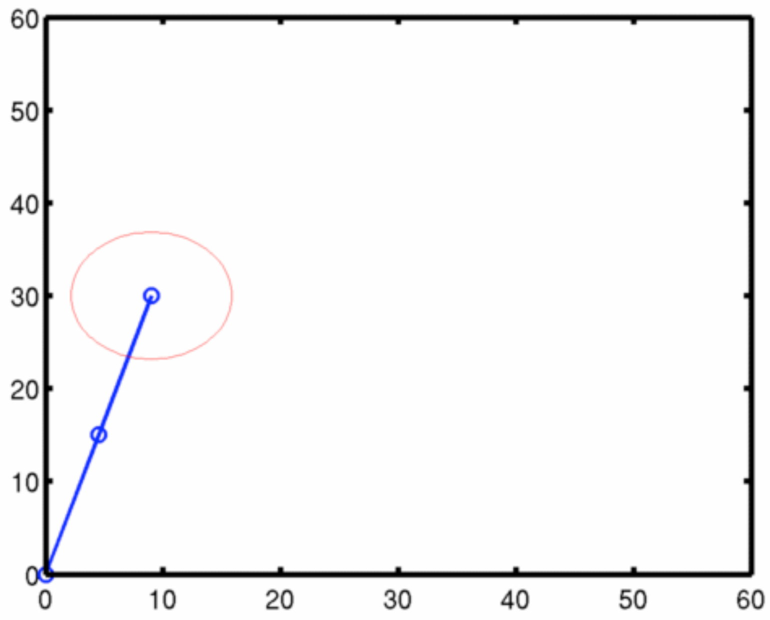
- **Linear target motion**
- **Uncertainty grows**

MMs: Constant Velocity

- **Constant target velocity** assumption
- Useful to model smooth target motion

- State representation
  $$\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} \end{bmatrix}^T$$
- Initial state
  $$\mathbf{x} = \begin{bmatrix} 0 & 0 & 9 & 30 \end{bmatrix}^T$$
- Transition matrix
  $$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ball example



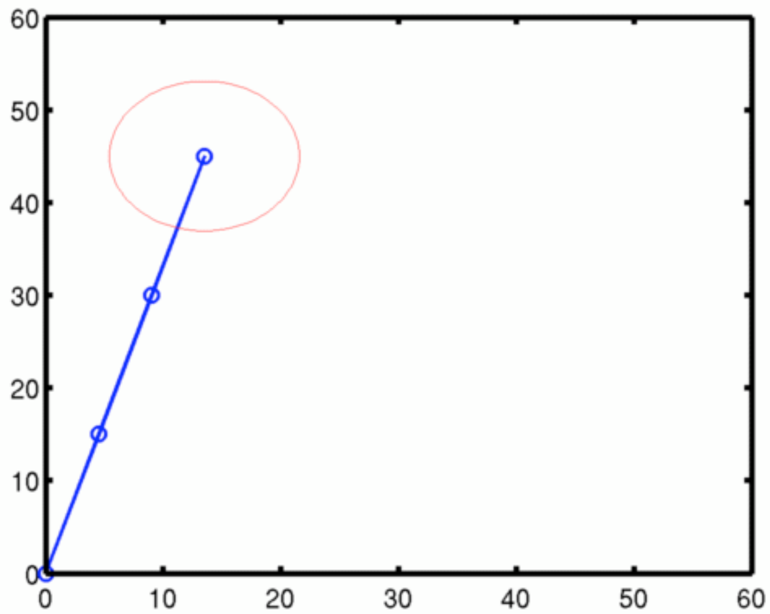
State prediction:
- **Linear target motion**
- **Uncertainty grows**

MMs: Constant Velocity

- **Constant target velocity** assumption
- Useful to model smooth target motion

- State representation
  $$\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} \end{bmatrix}^T$$
- Initial state
  $$\mathbf{x} = \begin{bmatrix} 0 & 0 & 9 & 30 \end{bmatrix}^T$$
- Transition matrix
  $$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ball example



State prediction:
- **Linear target motion**
- **Uncertainty grows**

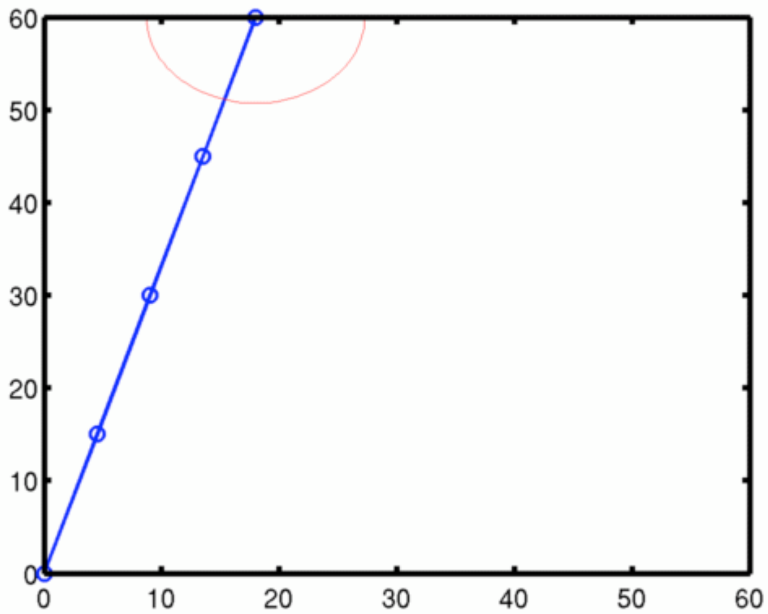
MMs: Constant Velocity

- **Constant target velocity** assumption
- Useful to model smooth target motion

- State representation
  $$\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} \end{bmatrix}^T$$
- Initial state
  $$\mathbf{x} = \begin{bmatrix} 0 & 0 & 9 & 30 \end{bmatrix}^T$$
- Transition matrix

$$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ball example

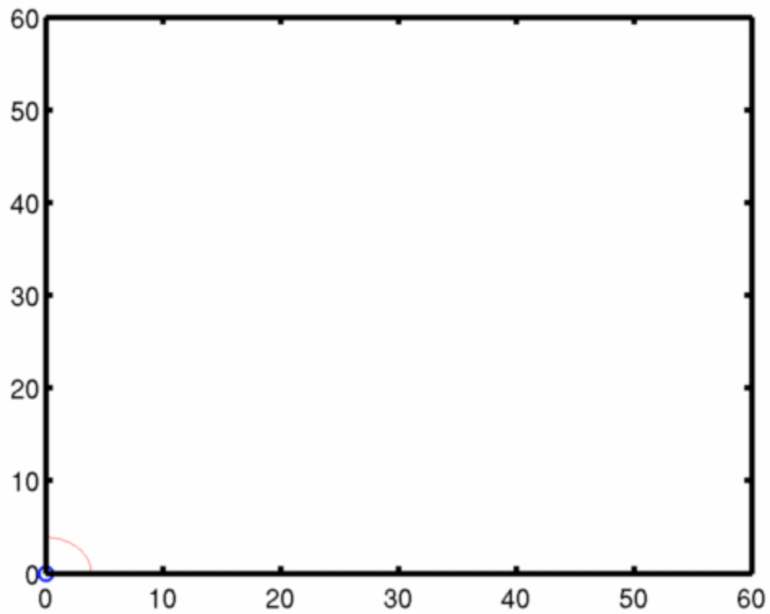


State prediction:
- **Linear target motion**
- **Uncertainty grows**

MMs: Constant Acceleration

- **Constant target acceleration** assumed
- Useful to model target motion that is smooth in position and velocity changes

Ball example



- State representation

$$\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} & \ddot{x} & \ddot{y} \end{bmatrix}^T$$

- Initial state

$$\mathbf{x} = \begin{bmatrix} 0 & 0 & 9 & 30 & 0 & -g \end{bmatrix}^T$$

- Transition matrix

$$F = \begin{bmatrix} 1 & 0 & T & 0 & \frac{T^2}{2} & 0 \\ 0 & 1 & 0 & T & 0 & \frac{T^2}{2} \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

MMs: Constant Acceleration

- **Constant target acceleration** assumed
- Useful to model target motion that is smooth in position and velocity changes

- State representation

$$\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} & \ddot{x} & \ddot{y} \end{bmatrix}^T$$

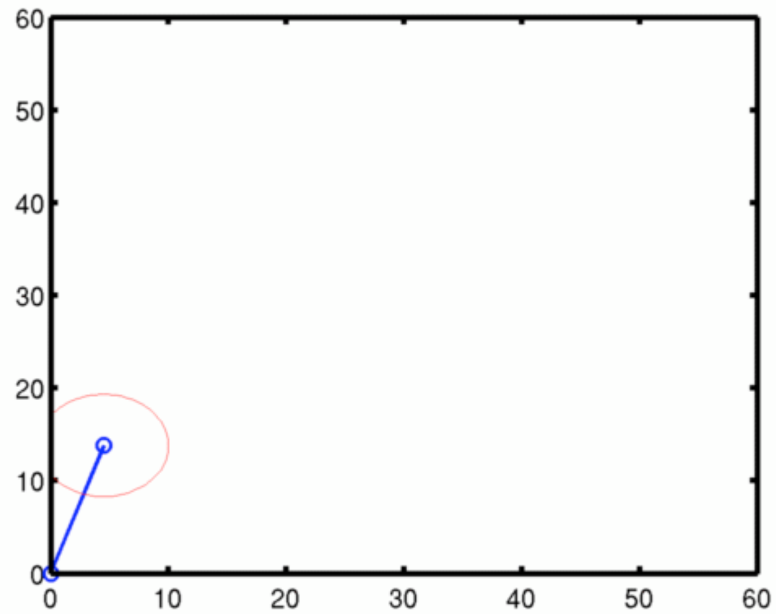
- Initial state

$$\mathbf{x} = \begin{bmatrix} 0 & 0 & 9 & 30 & 0 & -g \end{bmatrix}^T$$

- Transition matrix

$$F = \begin{bmatrix} 1 & 0 & T & 0 & \frac{T^2}{2} & 0 \\ 0 & 1 & 0 & T & 0 & \frac{T^2}{2} \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Ball example



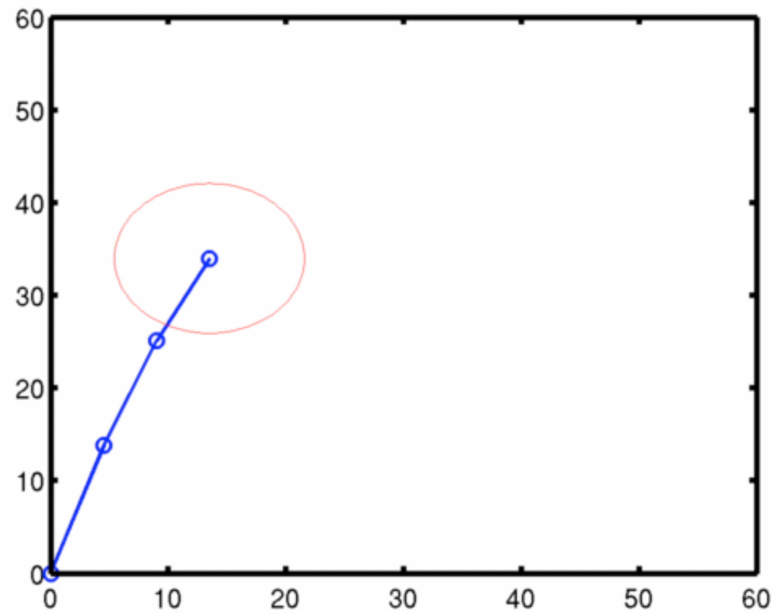
State prediction:

- **Non-linear motion**
- **Uncertainty grows**

MMs: Constant Acceleration

- **Constant target acceleration** assumed
- Useful to model target motion that is smooth in position and velocity changes

Ball example



- State representation

$$\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} & \ddot{x} & \ddot{y} \end{bmatrix}^T$$

- Initial state

$$\mathbf{x} = \begin{bmatrix} 0 & 0 & 9 & 30 & 0 & -g \end{bmatrix}^T$$

- Transition matrix

$$F = \begin{bmatrix} 1 & 0 & T & 0 & \frac{T^2}{2} & 0 \\ 0 & 1 & 0 & T & 0 & \frac{T^2}{2} \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

State prediction:

- **Non-linear motion**
- **Uncertainty grows**

MMs: Constant Acceleration

- **Constant target acceleration** assumed
- Useful to model target motion that is smooth in position and velocity changes

- State representation

$$\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} & \ddot{x} & \ddot{y} \end{bmatrix}^T$$

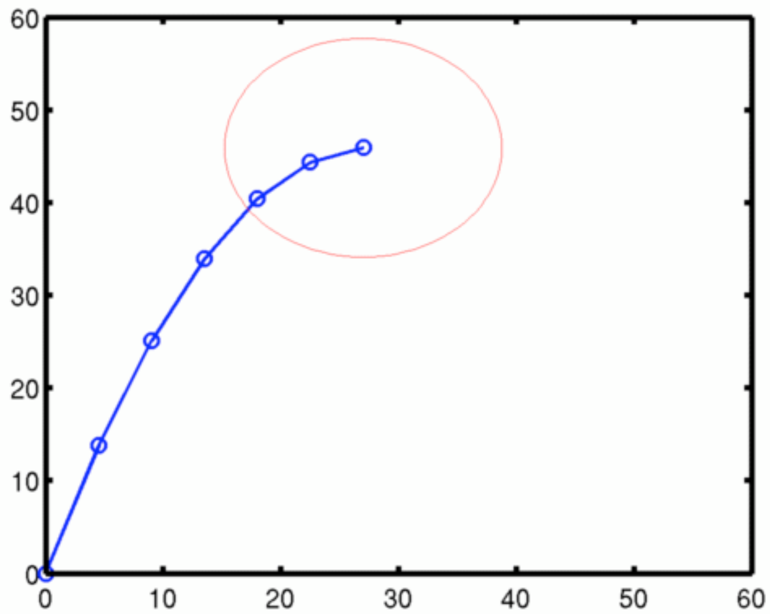
- Initial state

$$\mathbf{x} = \begin{bmatrix} 0 & 0 & 9 & 30 & 0 & -g \end{bmatrix}^T$$

- Transition matrix

$$F = \begin{bmatrix} 1 & 0 & T & 0 & \frac{T^2}{2} & 0 \\ 0 & 1 & 0 & T & 0 & \frac{T^2}{2} \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Ball example



State prediction:

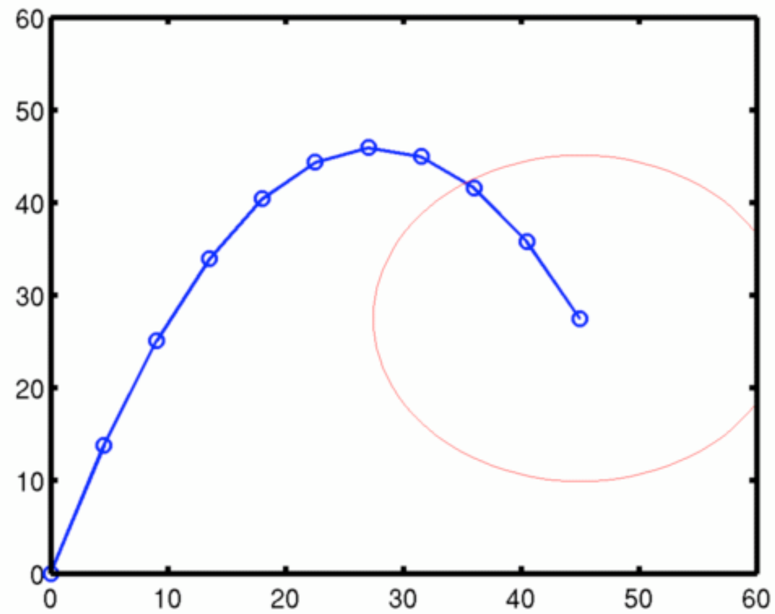
- **Non-linear motion**
- **Uncertainty grows**

MMs: Constant Acceleration

- **Constant target acceleration** assumed
- Useful to model target motion that is smooth in position and velocity changes

- State representation
$$\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} & \ddot{x} & \ddot{y} \end{bmatrix}^T$$
- Initial state
$$\mathbf{x} = \begin{bmatrix} 0 & 0 & 9 & 30 & 0 & -g \end{bmatrix}^T$$
- Transition matrix
$$F = \begin{bmatrix} 1 & 0 & T & 0 & \frac{T^2}{2} & 0 \\ 0 & 1 & 0 & T & 0 & \frac{T^2}{2} \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Ball example

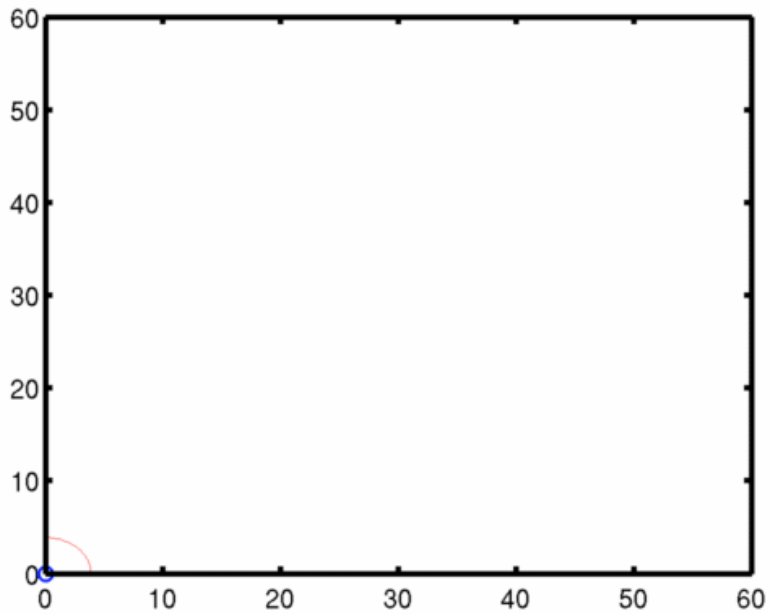


State prediction:
- **Non-linear motion**
- **Uncertainty grows**

MMs: Constant Acceleration

- **Constant target acceleration** assumed
- Useful to model target motion that is smooth in position and velocity changes

Ball example



- State representation

$$\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} & \ddot{x} & \ddot{y} \end{bmatrix}^T$$

- Initial state

$$\mathbf{x} = \begin{bmatrix} 0 & 0 & 30 & 30 & -20 & -12 \end{bmatrix}^T$$

- Transition matrix

$$F = \begin{bmatrix} 1 & 0 & T & 0 & \frac{T^2}{2} & 0 \\ 0 & 1 & 0 & T & 0 & \frac{T^2}{2} \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

MMs: Constant Acceleration

- **Constant target acceleration** assumed
- Useful to model target motion that is smooth in position and velocity changes
- State representation

$$\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} & \ddot{x} & \ddot{y} \end{bmatrix}^T$$

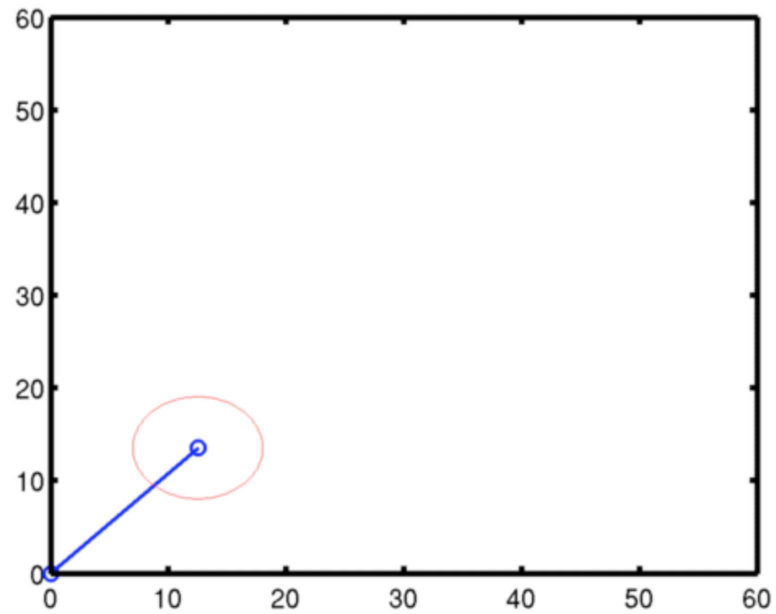
- Initial state

$$\mathbf{x} = \begin{bmatrix} 0 & 0 & 30 & 30 & -20 & -12 \end{bmatrix}^T$$

- Transition matrix

$$F = \begin{bmatrix} 1 & 0 & T & 0 & \frac{T^2}{2} & 0 \\ 0 & 1 & 0 & T & 0 & \frac{T^2}{2} \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Ball example



State prediction:

- **Non-linear motion**
- **Uncertainty grows**

MMs: Constant Acceleration

- **Constant target acceleration** assumed
- Useful to model target motion that is smooth in position and velocity changes

- State representation

$$\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} & \ddot{x} & \ddot{y} \end{bmatrix}^T$$

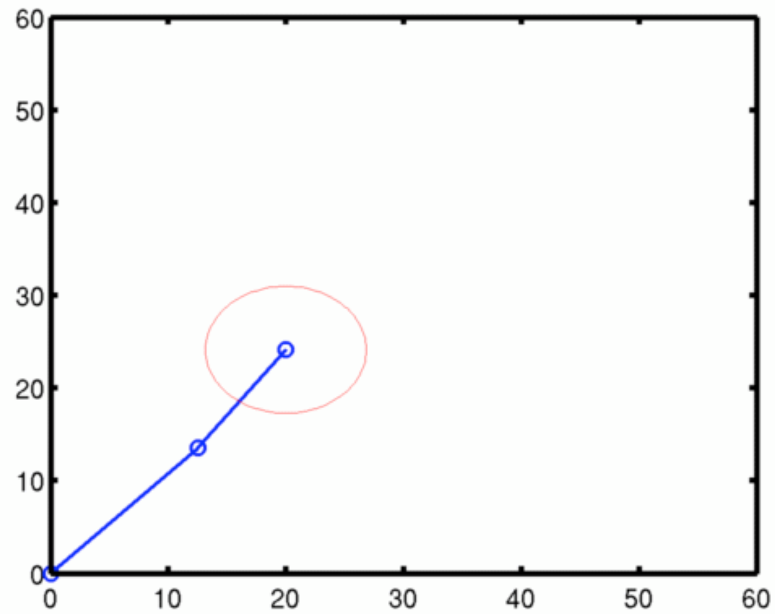
- Initial state

$$\mathbf{x} = \begin{bmatrix} 0 & 0 & 30 & 30 & -20 & -12 \end{bmatrix}^T$$

- Transition matrix

$$F = \begin{bmatrix} 1 & 0 & T & 0 & \frac{T^2}{2} & 0 \\ 0 & 1 & 0 & T & 0 & \frac{T^2}{2} \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Ball example



State prediction:

- **Non-linear motion**
- **Uncertainty grows**

MMs: Constant Acceleration

- **Constant target acceleration** assumed
- Useful to model target motion that is smooth in position and velocity changes

- State representation

$$\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} & \ddot{x} & \ddot{y} \end{bmatrix}^T$$

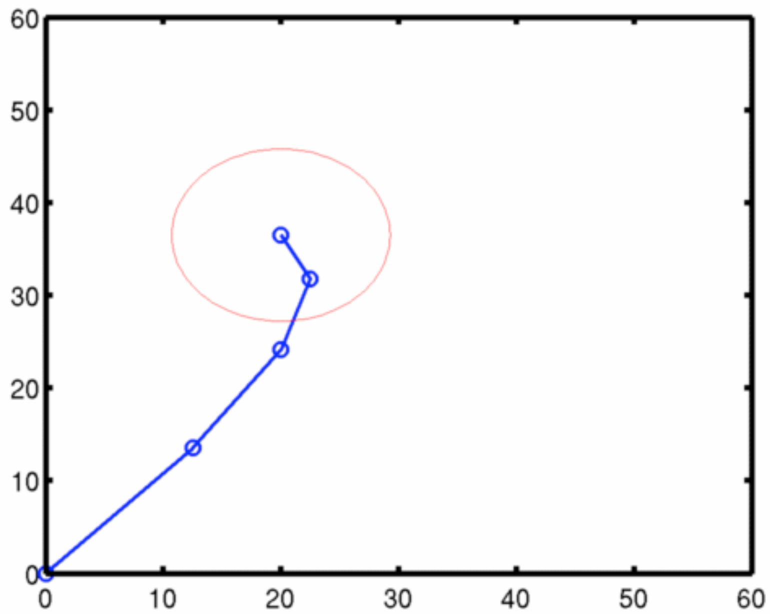
- Initial state

$$\mathbf{x} = \begin{bmatrix} 0 & 0 & 30 & 30 & -20 & -12 \end{bmatrix}^T$$

- Transition matrix

$$F = \begin{bmatrix} 1 & 0 & T & 0 & \frac{T^2}{2} & 0 \\ 0 & 1 & 0 & T & 0 & \frac{T^2}{2} \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Ball example



State prediction:

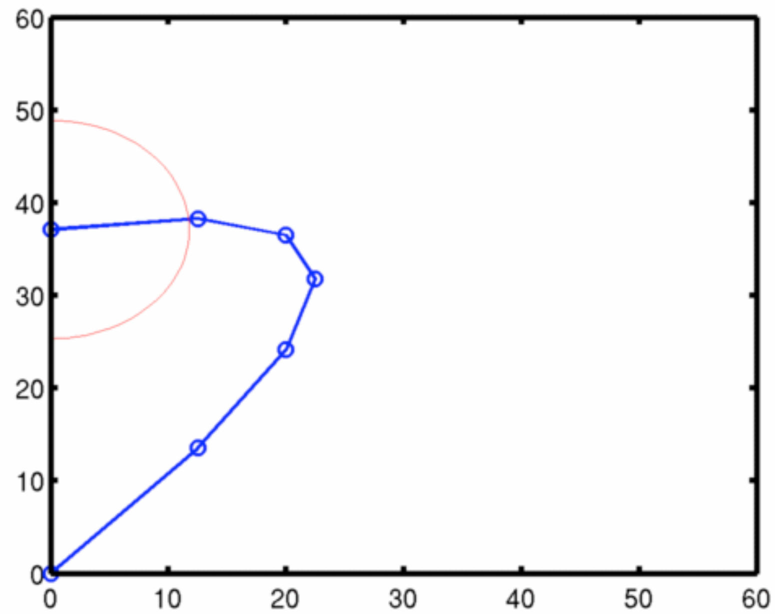
- **Non-linear motion**
- **Uncertainty grows**

MMs: Constant Acceleration

- **Constant target acceleration** assumed
- Useful to model target motion that is smooth in position and velocity changes

- State representation
$$\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} & \ddot{x} & \ddot{y} \end{bmatrix}^T$$
- Initial state
$$\mathbf{x} = \begin{bmatrix} 0 & 0 & 30 & 30 & -20 & -12 \end{bmatrix}^T$$
- Transition matrix
$$F = \begin{bmatrix} 1 & 0 & T & 0 & \frac{T^2}{2} & 0 \\ 0 & 1 & 0 & T & 0 & \frac{T^2}{2} \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Ball example



State prediction:
- **Non-linear motion**
- **Uncertainty grows**

KF Cycle: Measurement Predict.

- **Measurement prediction**

$$\widehat{z}(k) = H(k)\widehat{x}(k+1|k)$$

$$\widehat{S}(k) = H(k)\widehat{P}(k+1|k)H^T(k) + R(k)$$

- **Observation**
  Typically, only the target **position** is observed.
  The measurement matrix is then

$$\mathbf{z} = \left[\begin{array}{cc} x & y \end{array}\right]^T \qquad H = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right]$$

  Note: One can also observe (not in this course)
  - Velocity (Doppler radar)
  - Acceleration (accelerometers)

KF Cycle: Data Association

- Once measurements are predicted and observed, we have to **associate them with each other**
- This is basically resolving the **origin uncertainty** of observations
- Data association is typically done in the **sensor reference frame**
- Data association can be a **hard problem** and many advanced techniques exist

➡ More on this later in this course

KF Cycle: Data Association

Step 1: Compute the pairing difference and its associated uncertainty

- The difference between predicted measurement and observation is called **innovation**

$$\nu_{ij}(k) \;=\; z_i(k) - \hat{z}_j(k)$$

- The associated covariance estimate is called the **innovation covariance**

$$\hat{S}_{ij}(k) \;=\; H(k)\hat{P}_j(k+1|k)H^T(k) + R_i(k)$$

- The prediction-observation pair is often called **pairing**

KF Cycle: Data Association

Step 2: Check if the pairing is statistically compatible

- Compute the **Mahalanobis distance**

$$d_{ij}^2 \;=\; \nu_{ij}(k)^T \widehat{S}_{ij}(k)^{-1} \nu_{ij}(k)$$

- Compare it against the proper threshold from an cumulative χ^2 ("**chi square**") **distribution**

$$d_{ij} \;\leq\; \chi^2_{n,\alpha}$$

Significance level

Degrees of freedom

Compatibility on level α is finally given if this is true

KF Cycle: Data Association

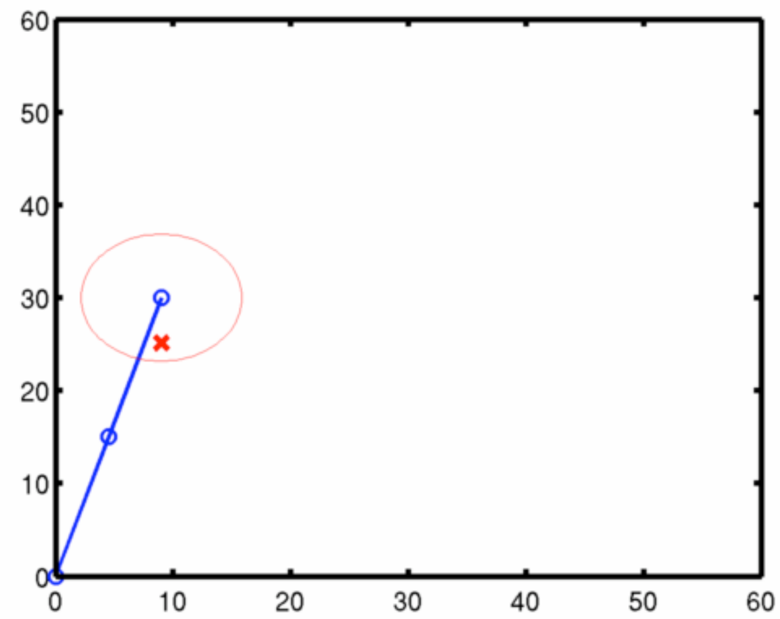
- Constant velocity model
- Process noise

$$Q = \begin{bmatrix} 5.0 & 0 & 0 & 0 \\ 0 & 5.0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}$$

- Measurement noise

$$R = \begin{bmatrix} 10.0 & 0 \\ 0 & 10.0 \end{bmatrix}$$

- No false alarm



➔ **No problem**

KF Cycle: Data Association

- Constant velocity model
- Process noise

$$Q = \begin{bmatrix} 5.0 & 0 & 0 & 0 \\ 0 & 5.0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}$$

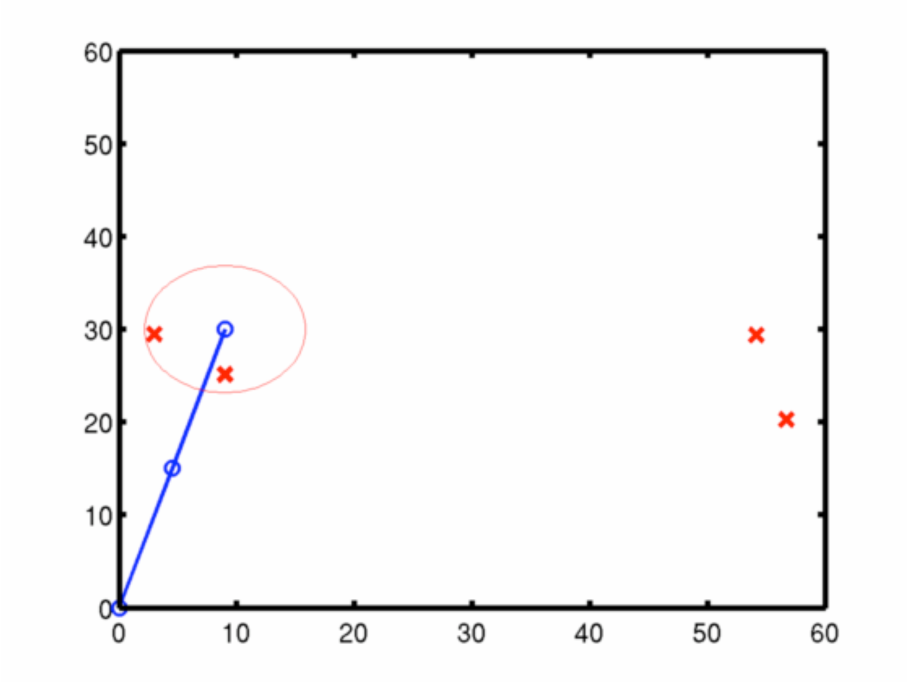
- Measurement noise

$$R = \begin{bmatrix} 10.0 & 0 \\ 0 & 10.0 \end{bmatrix}$$

- Uniform false alarm

$$x \sim \mathcal{U}(0, 60), \quad y \sim \mathcal{U}(0, 60)$$

- False alarm rate = 3



➔ **Ambiguity**: several observations in the validation gate

KF Cycle: Data Association

- Constant velocity model
- Process noise

$$Q = \begin{bmatrix} 5.0 & 0 & 0 & 0 \\ 0 & 5.0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}$$

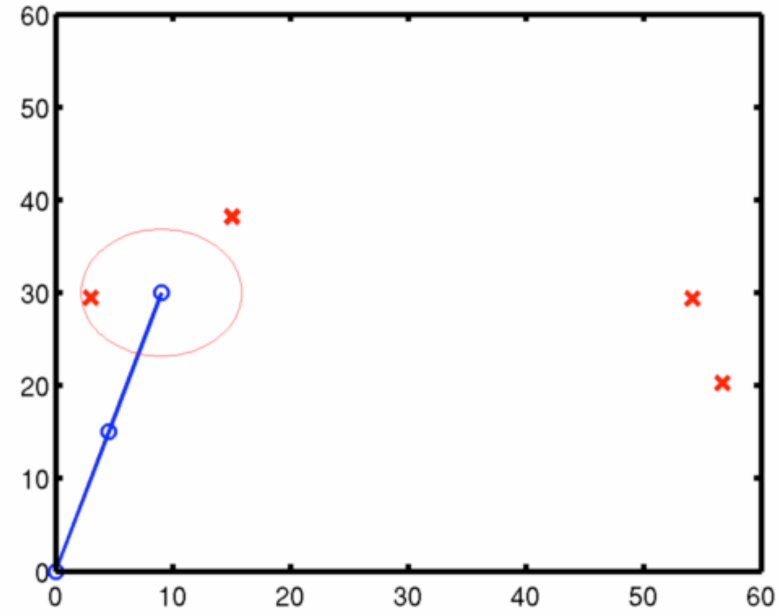
- Measurement noise

$$R = \begin{bmatrix} 50.0 & 0 \\ 0 & 50.0 \end{bmatrix}$$

- Uniform false alarm

$$x \sim \mathcal{U}(0,60), \quad y \sim \mathcal{U}(0,60)$$

- False alarm rate = 3



➔ **Wrong association** as closest observation is false alarm

KF Cycle: Update

- Computation of the Kalman gain

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

- State and state covariance update

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

$$\widehat{P}(k+1|k+1) = (I - K(k)H(k))\widehat{P}(k+1|k)$$

Track management: Naïve

Creation

- When to create a new track?
- What is the initial state?

Two heuristics

- Greedy initialization
  - Every observation not associated is a new track
  - Initialize only position
- Lazy initialization
  - Accumulate several unassociated observations
  - Initialize position & velocity

Occlusion/deletion

- When to delete a track?
- Is it just occluded?

Two heuristics

- Greedy deletion
  - Delete if no observation can be associated
  - No occlusion handling
- Lazy deletion
  - Delete if no observation can be associated for several time
  - Implicit occlusion handling

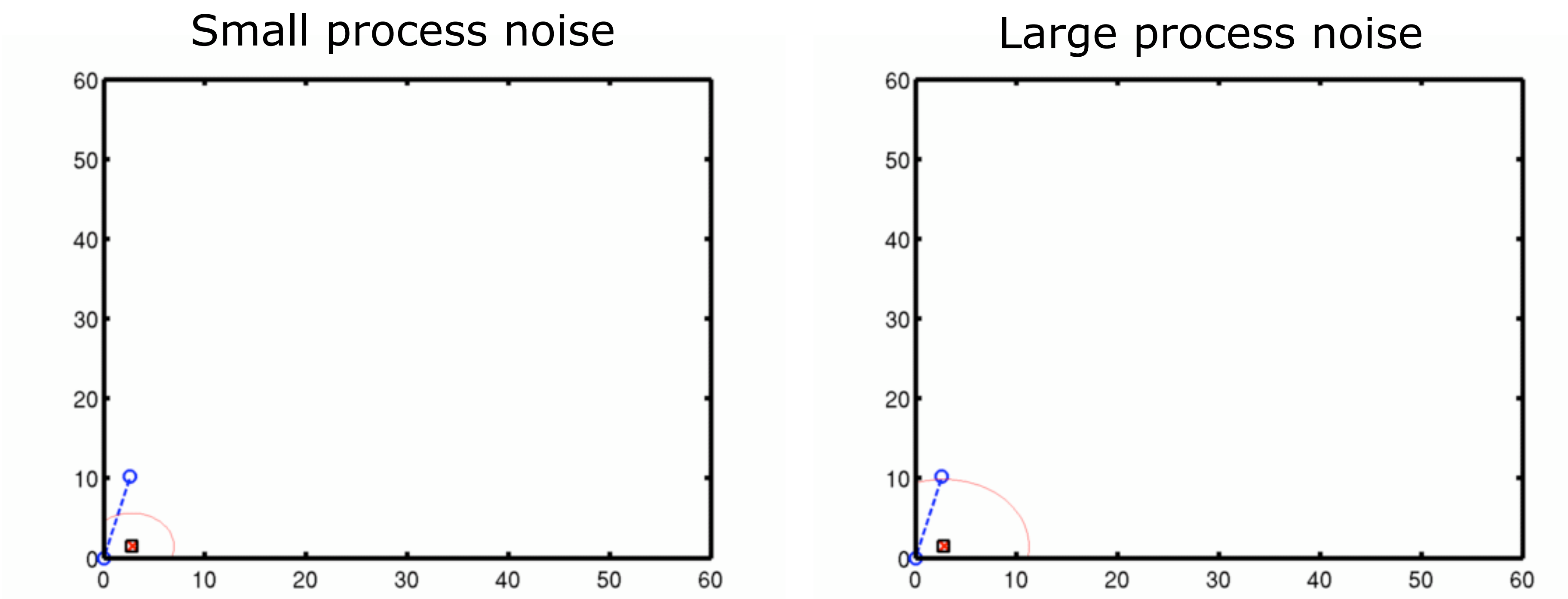
Example: Tracking the Ball

- Unlike the previous experiment in which we had a model of the ball's trajectory and just observed it, **we now want to track the ball**

- Comparison: small versus large process noise Q and the effect of the three different motion models

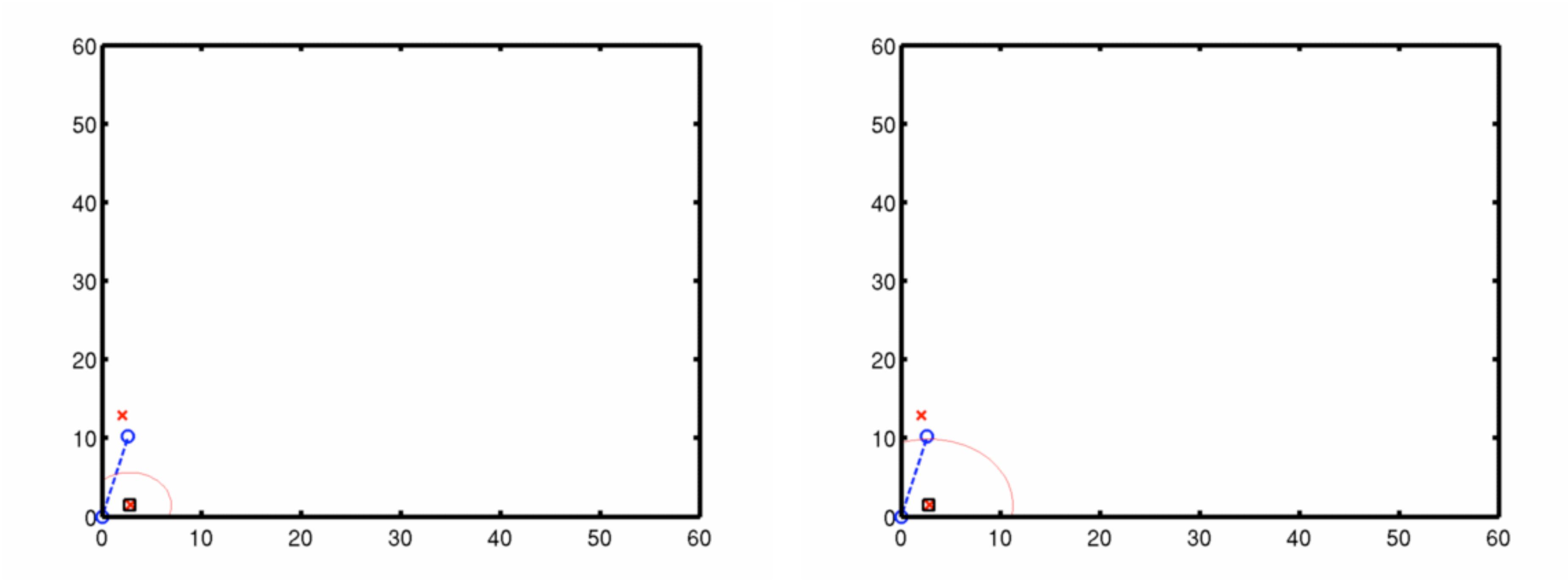
- For simplicity, we perform **no gaiting** (i.e. no Mahalanobis test) but accept the pairing every time

Ball Tracking: Brownian



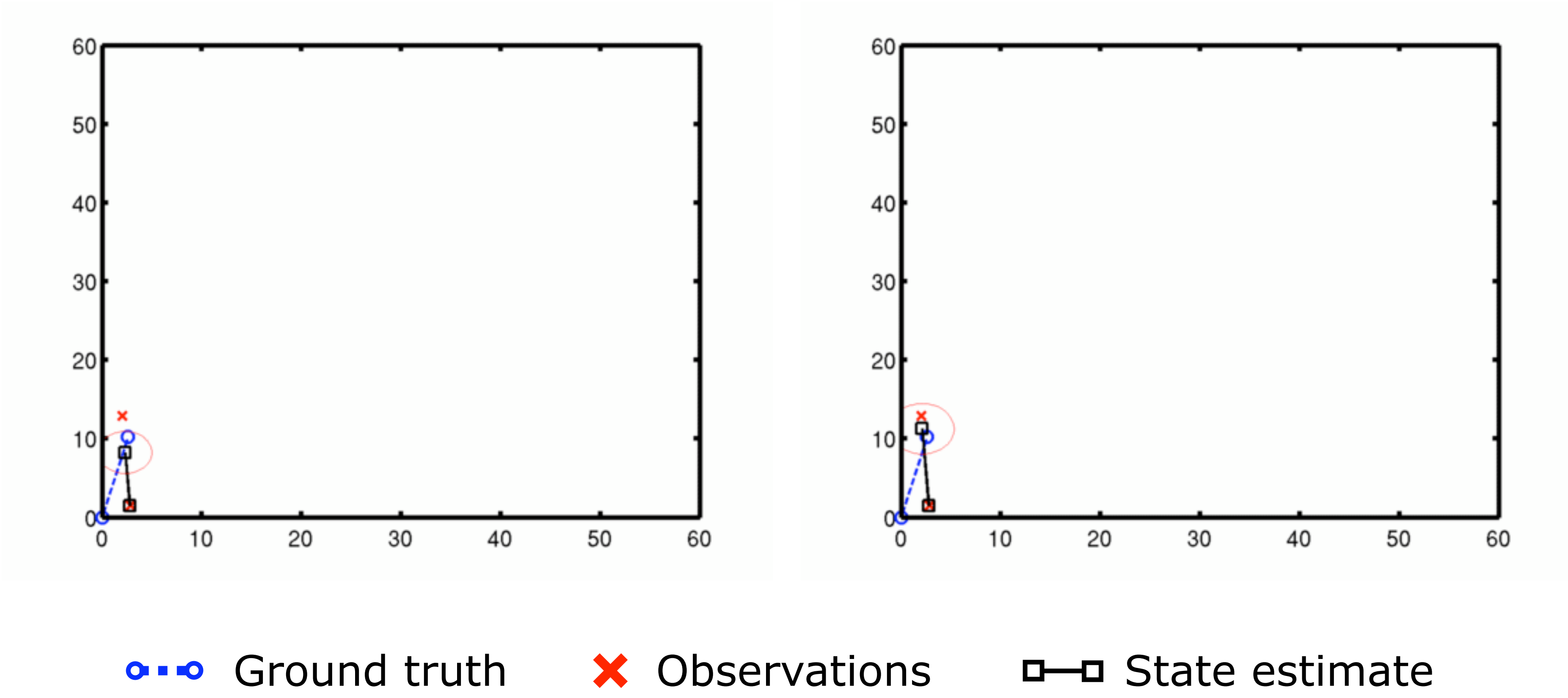
Ground truth   Observations   State estimate

Ball Tracking: Brownian

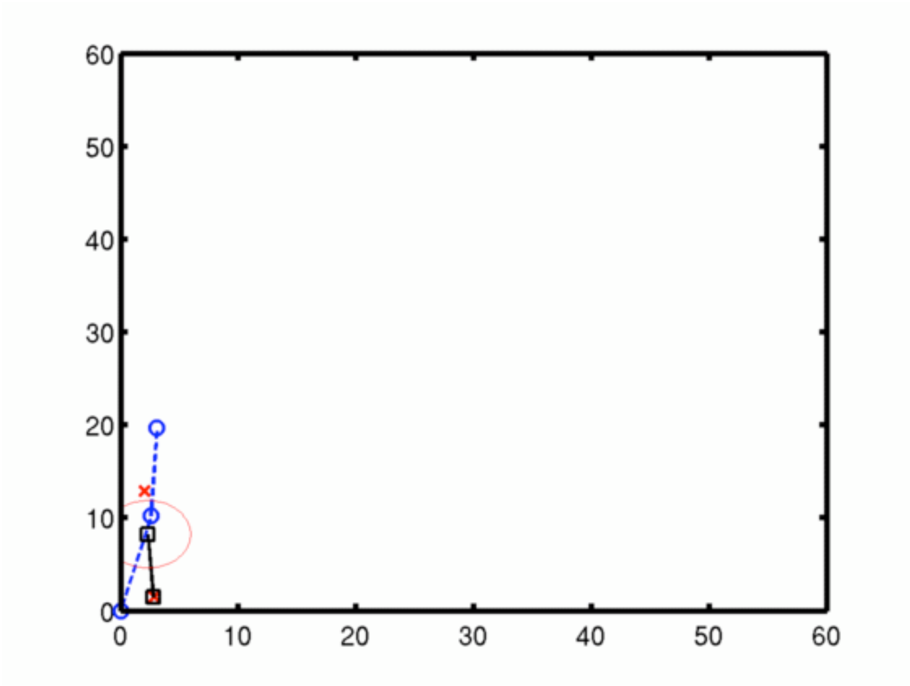


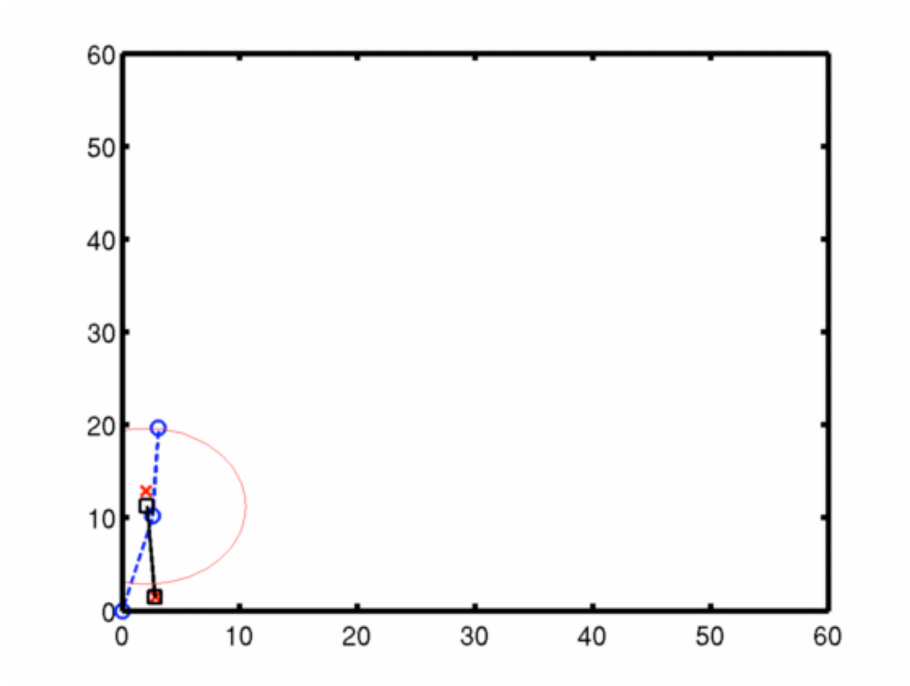
Ground truth Observations State estimate

Ball Tracking: Brownian



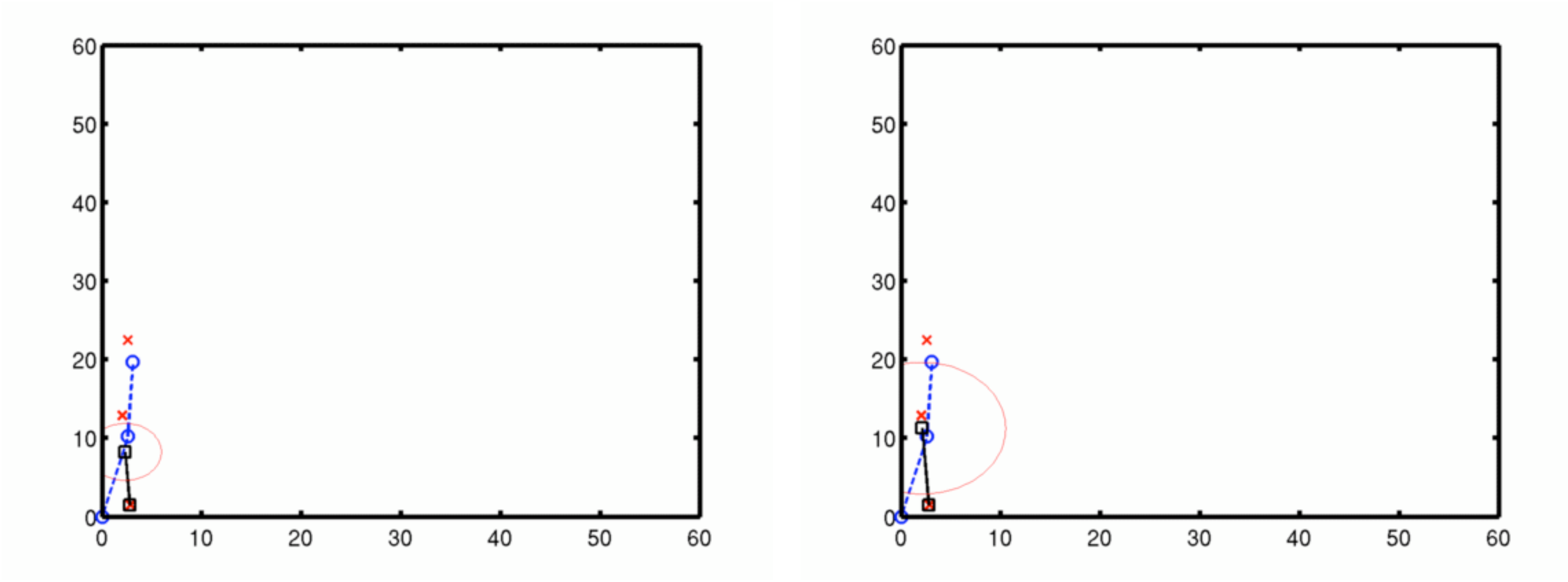
Ball Tracking: Brownian





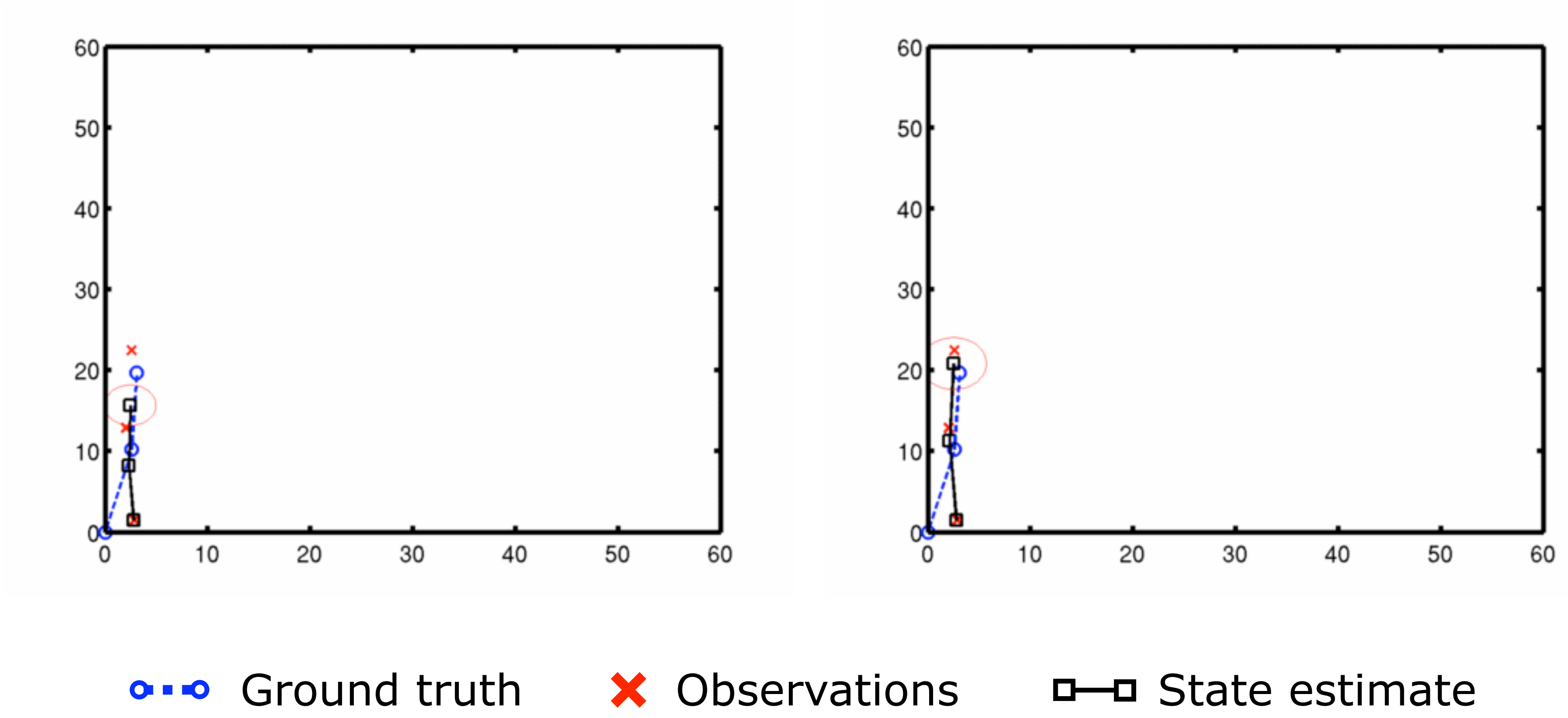
Ground truth Observations State estimate

Ball Tracking: Brownian

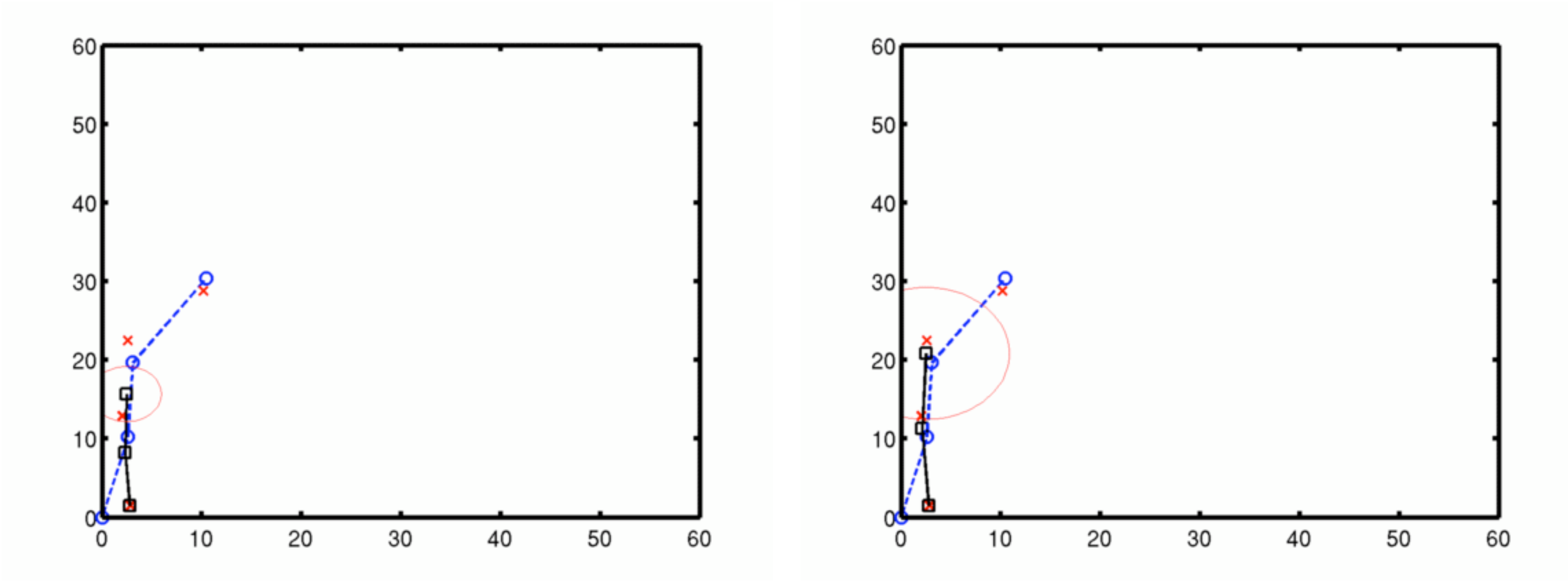


Ground truth   Observations   State estimate

Ball Tracking: Brownian

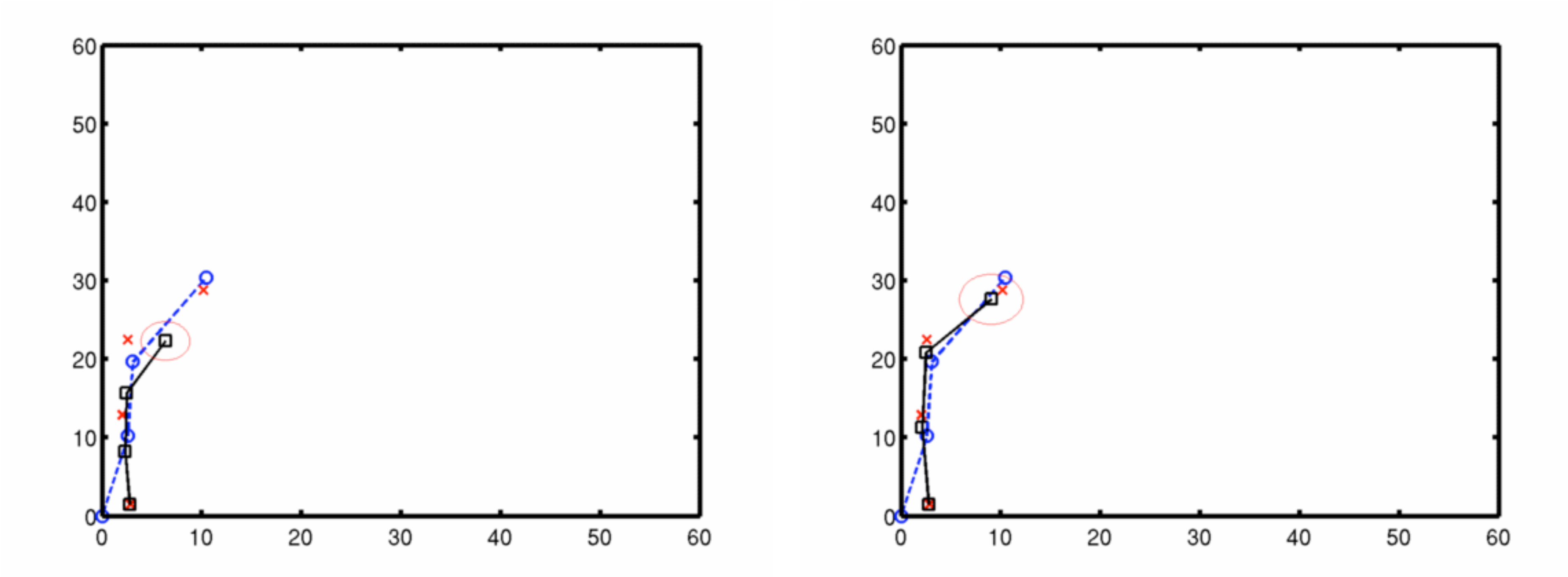


Ball Tracking: Brownian



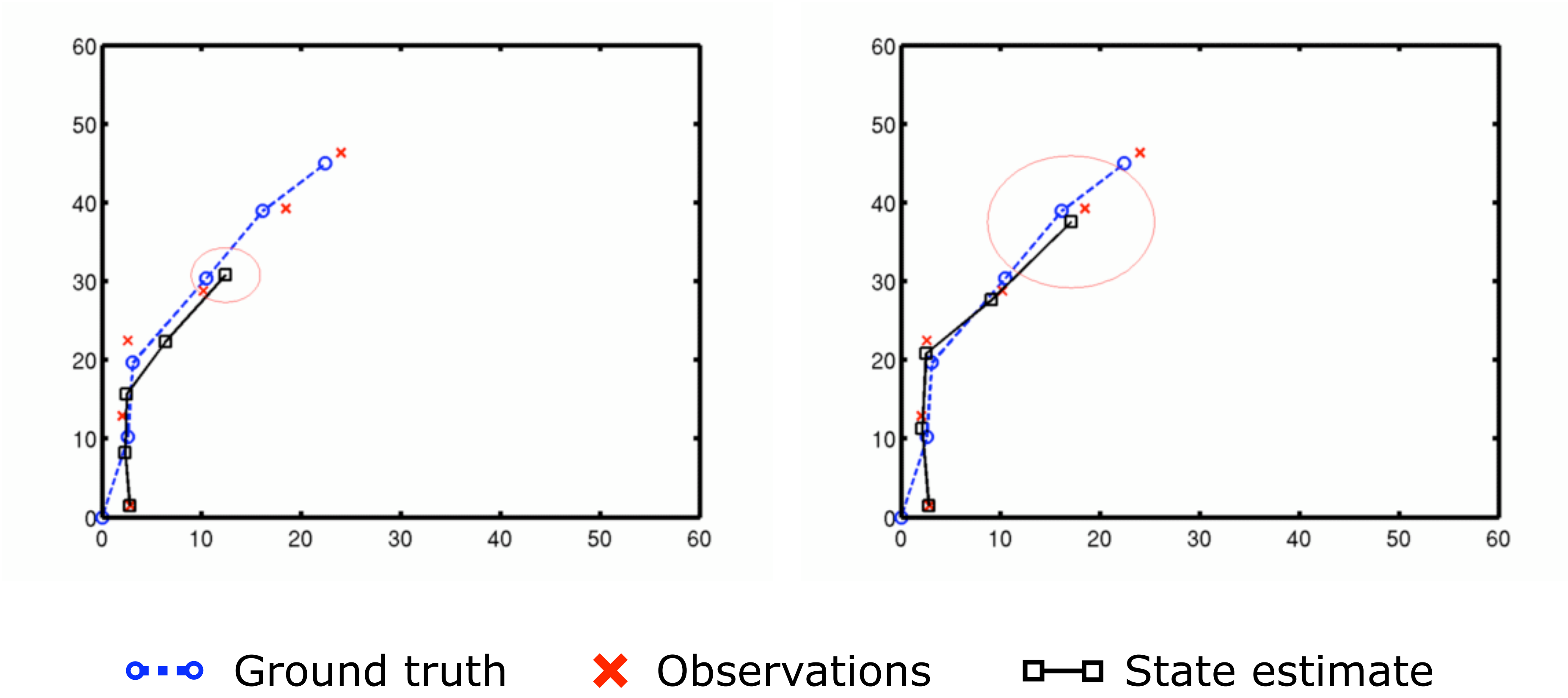
Ground truth   Observations   State estimate

Ball Tracking: Brownian

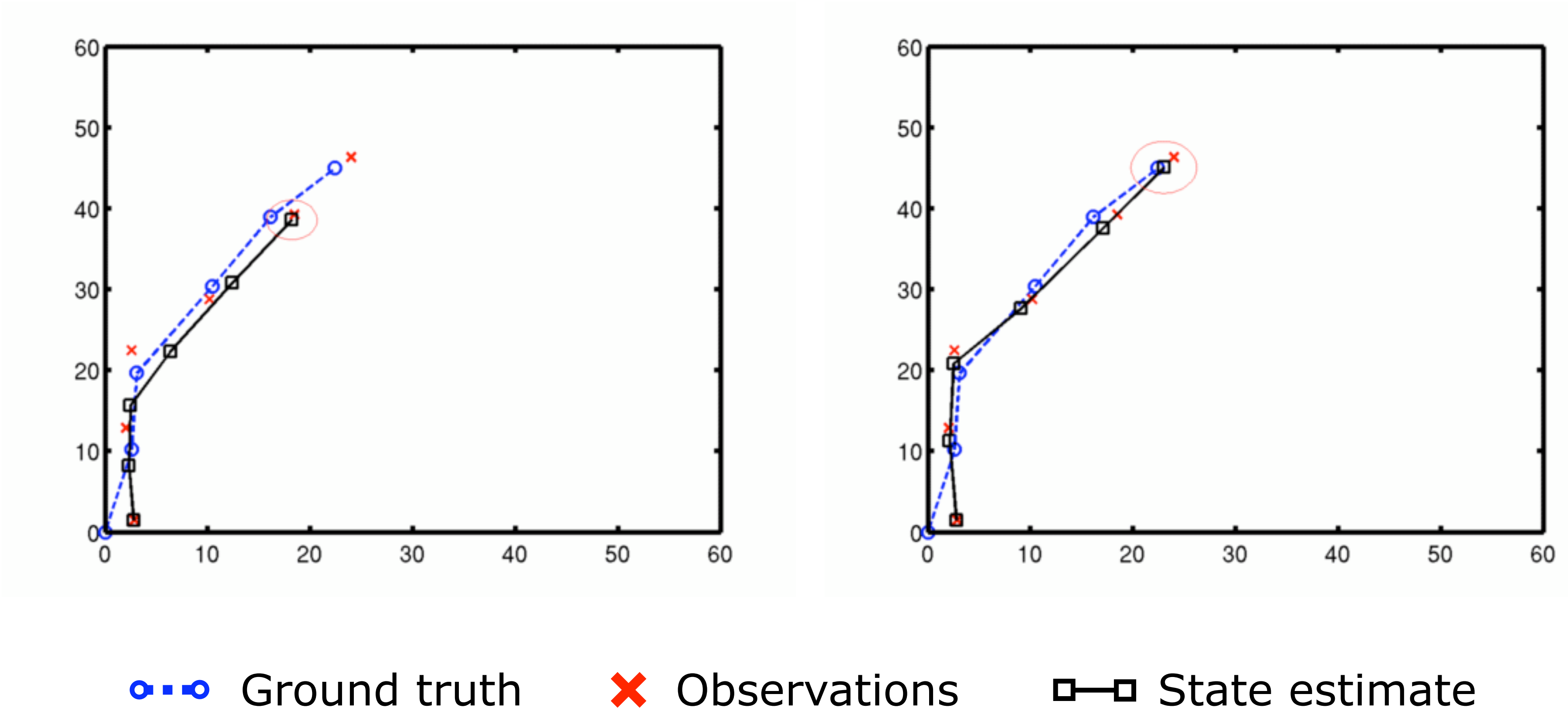


Ground truth　Observations　State estimate

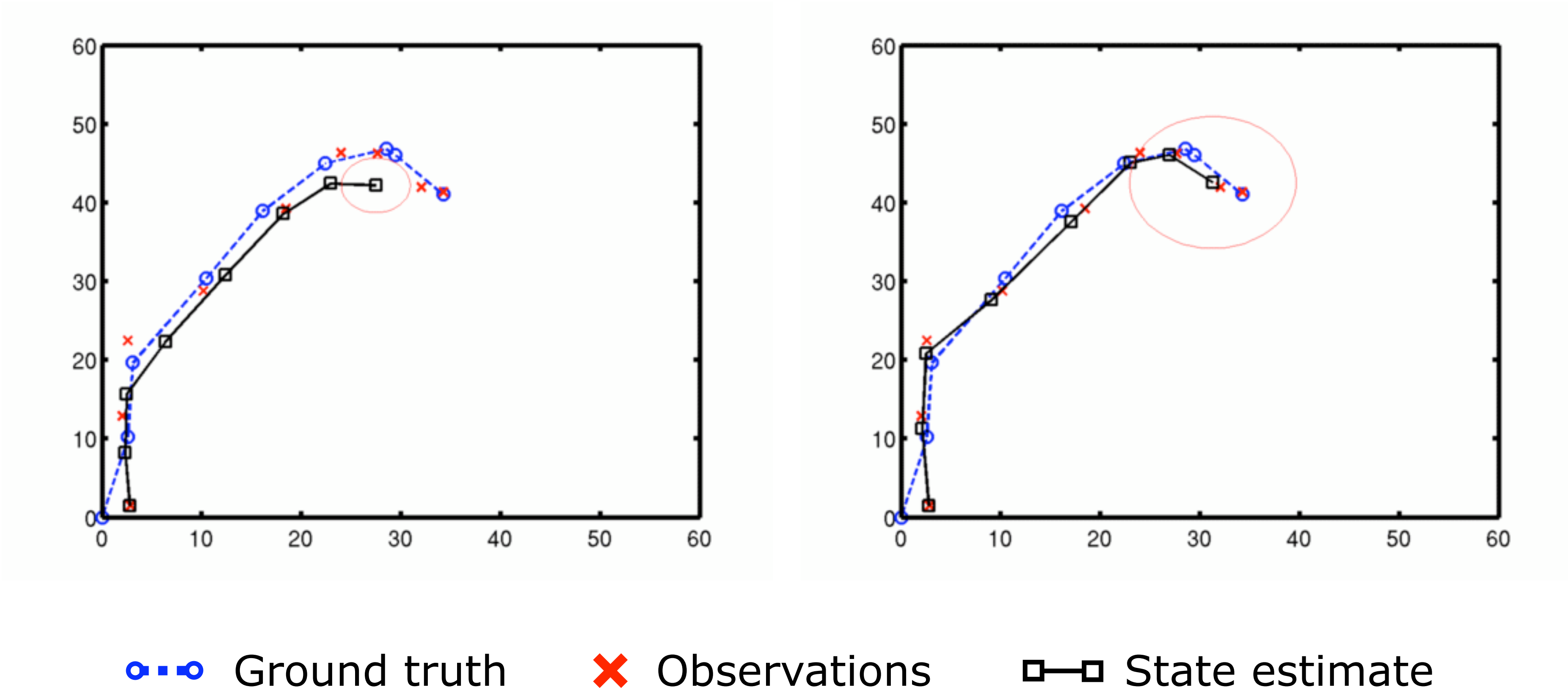
Ball Tracking: Brownian



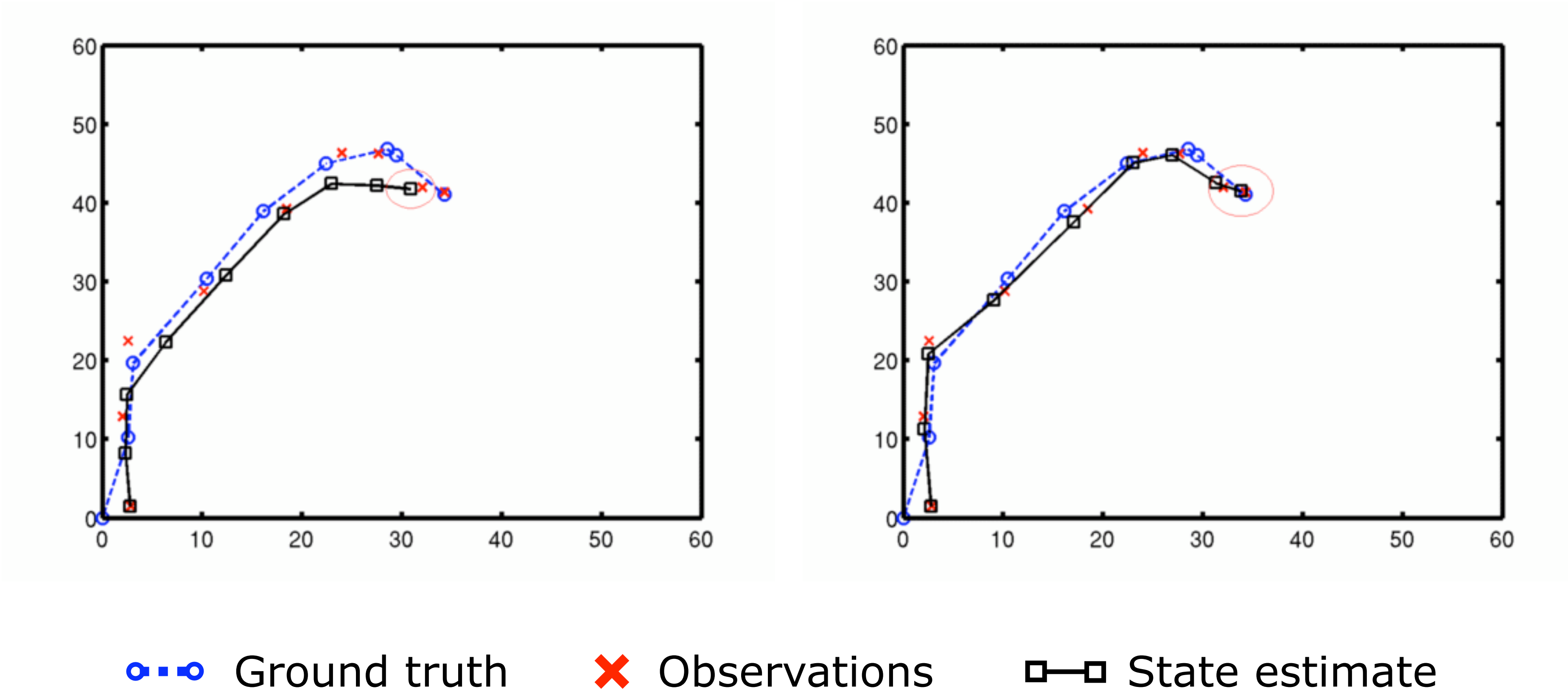
Ball Tracking: Brownian



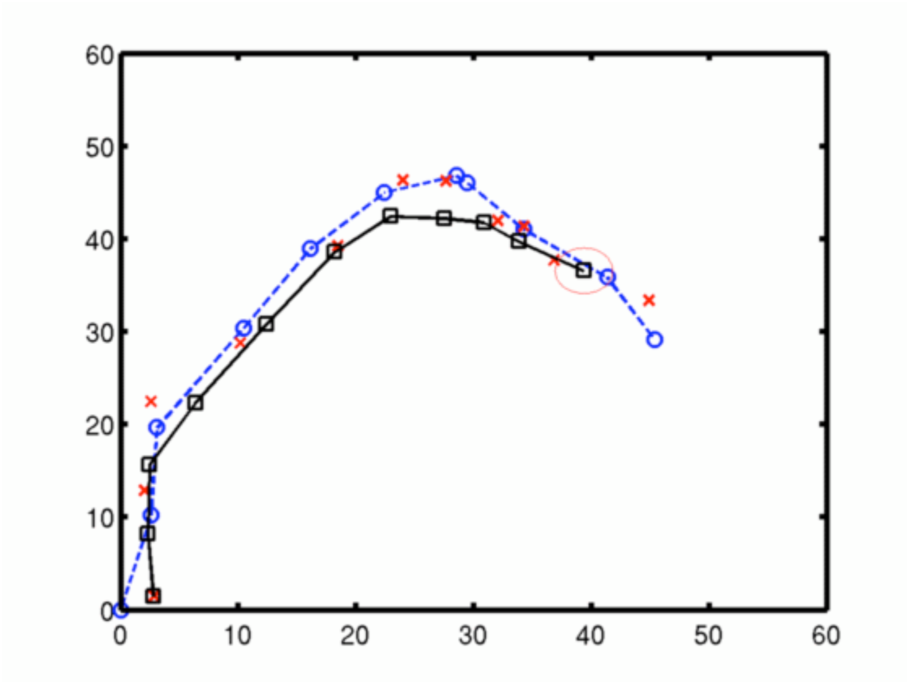
Ball Tracking: Brownian

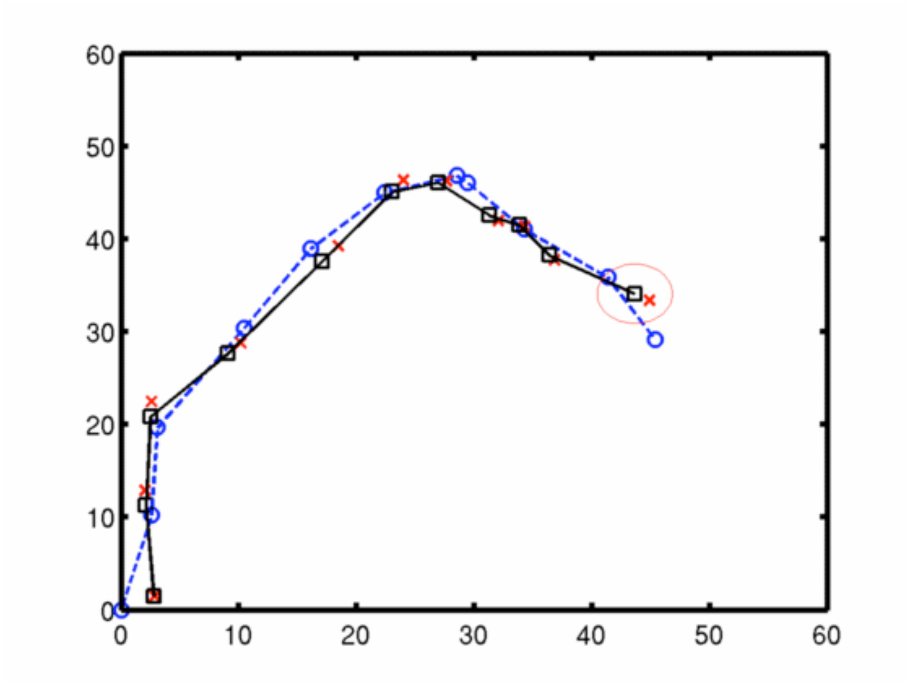


Ball Tracking: Brownian



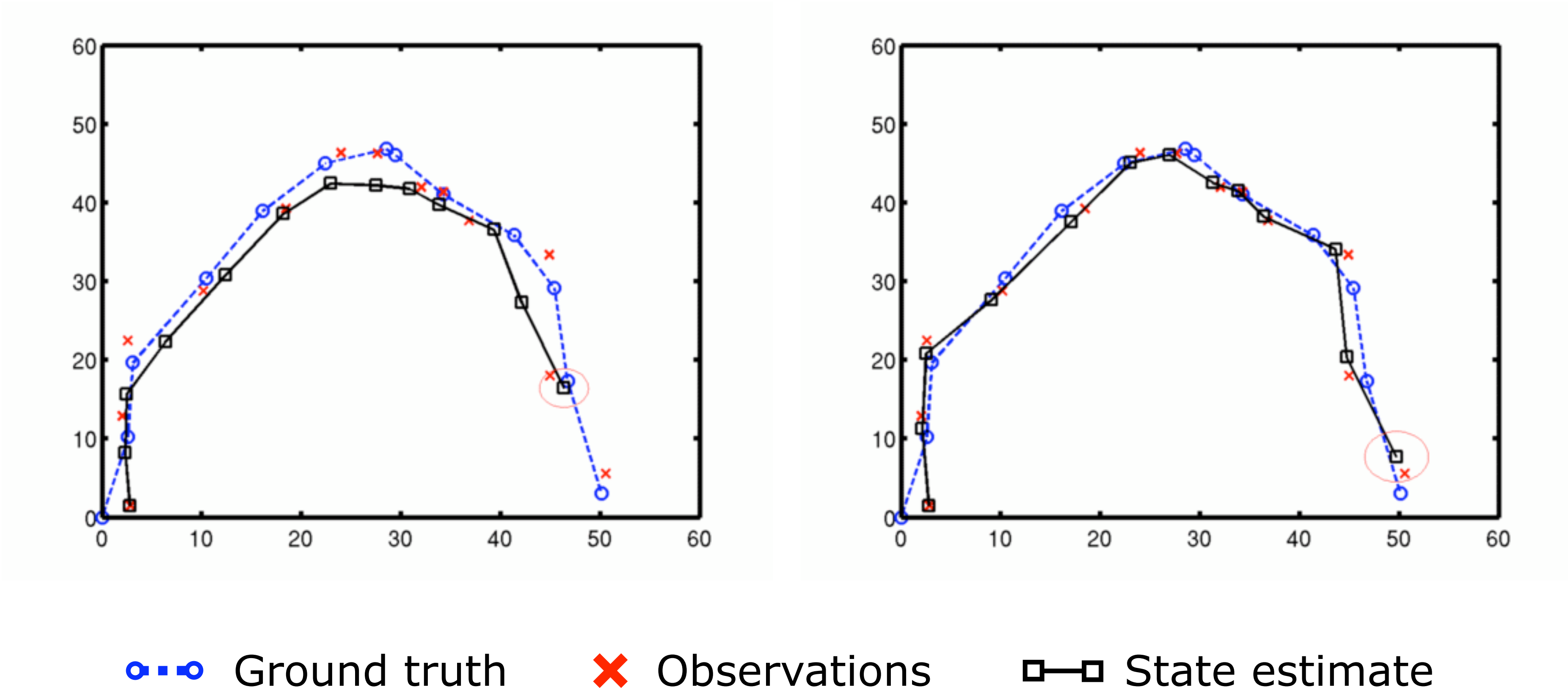
Ball Tracking: Brownian



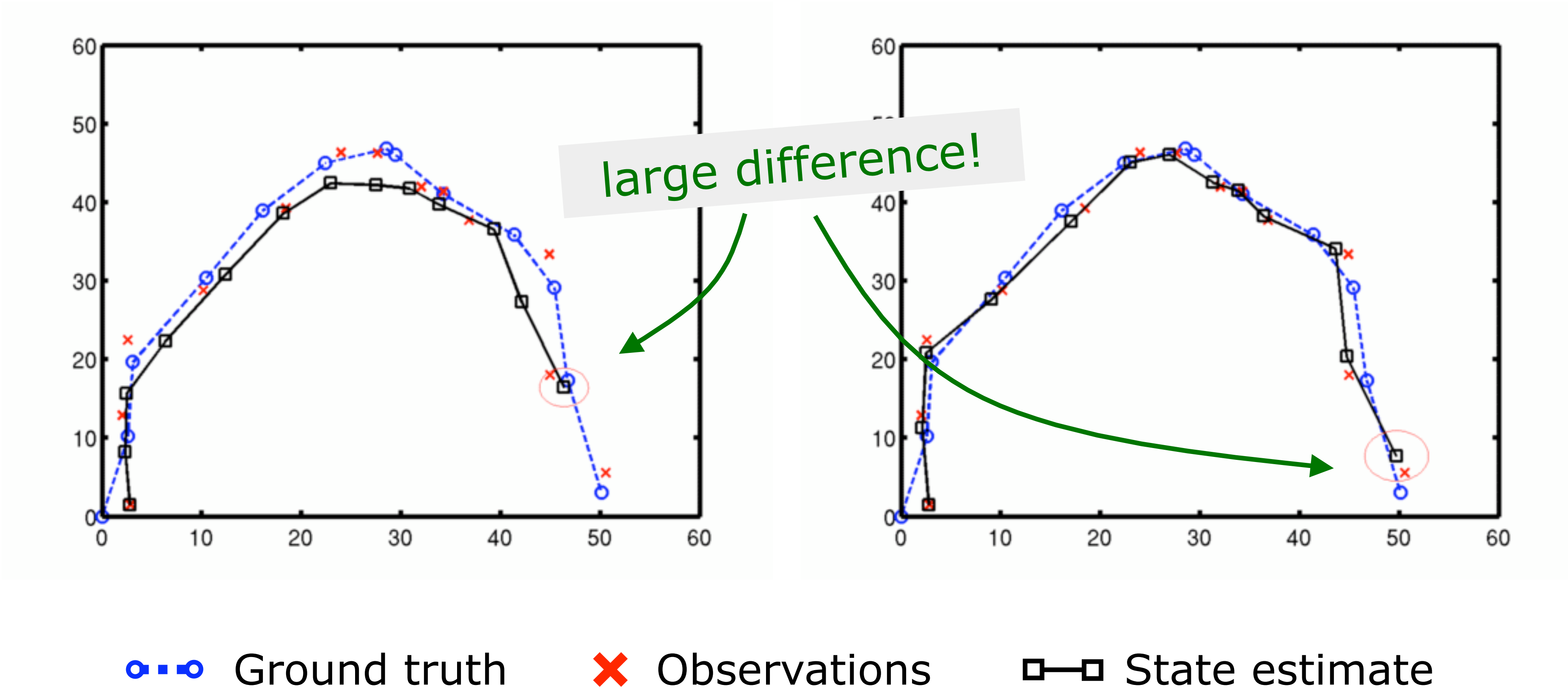


Ground truth   Observations   State estimate

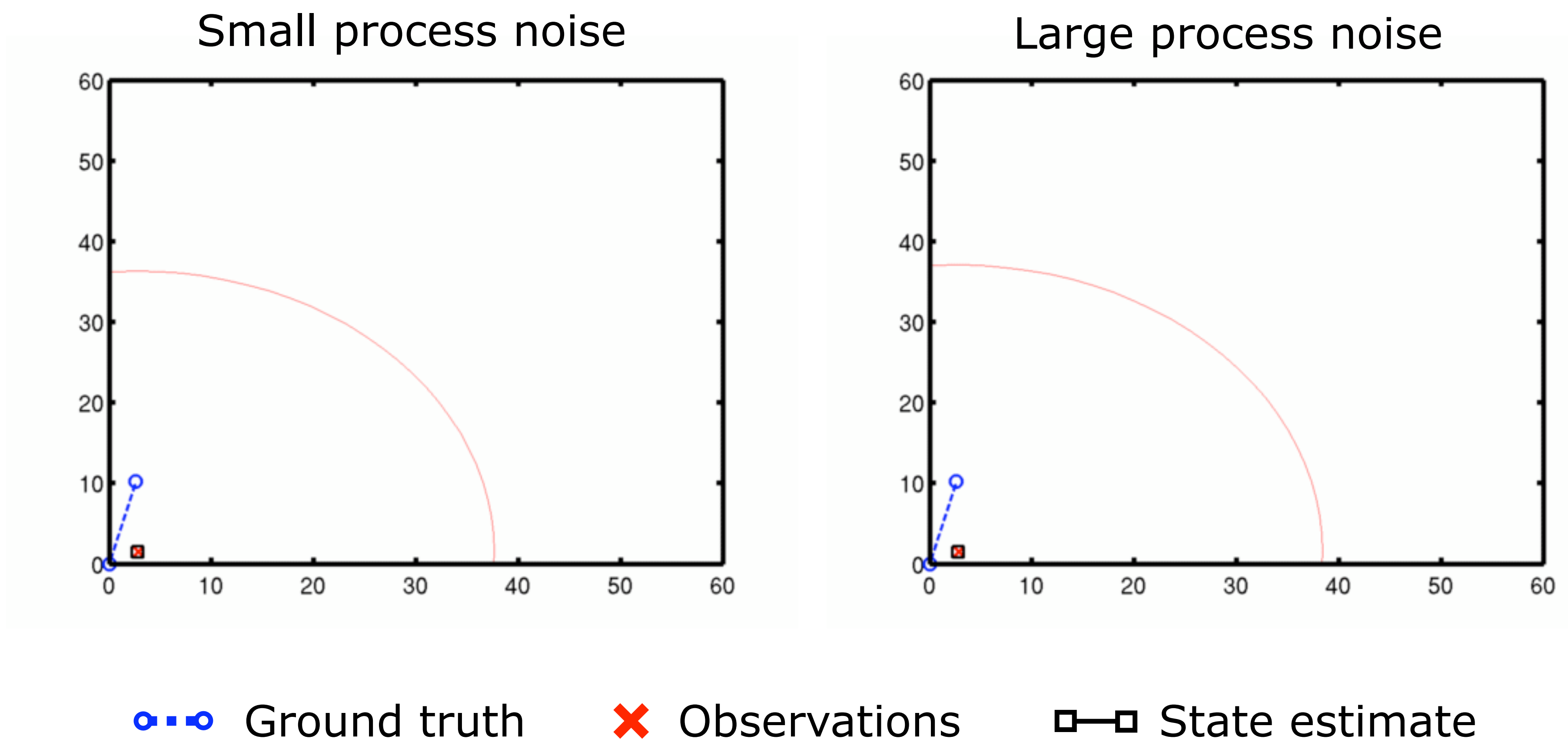
Ball Tracking: Brownian



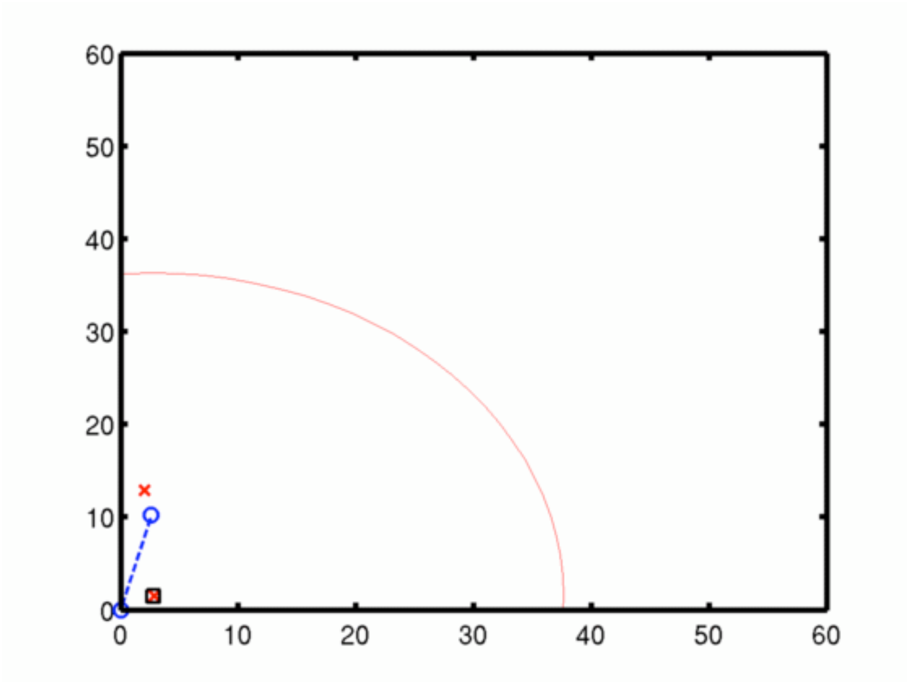
Ball Tracking: Brownian

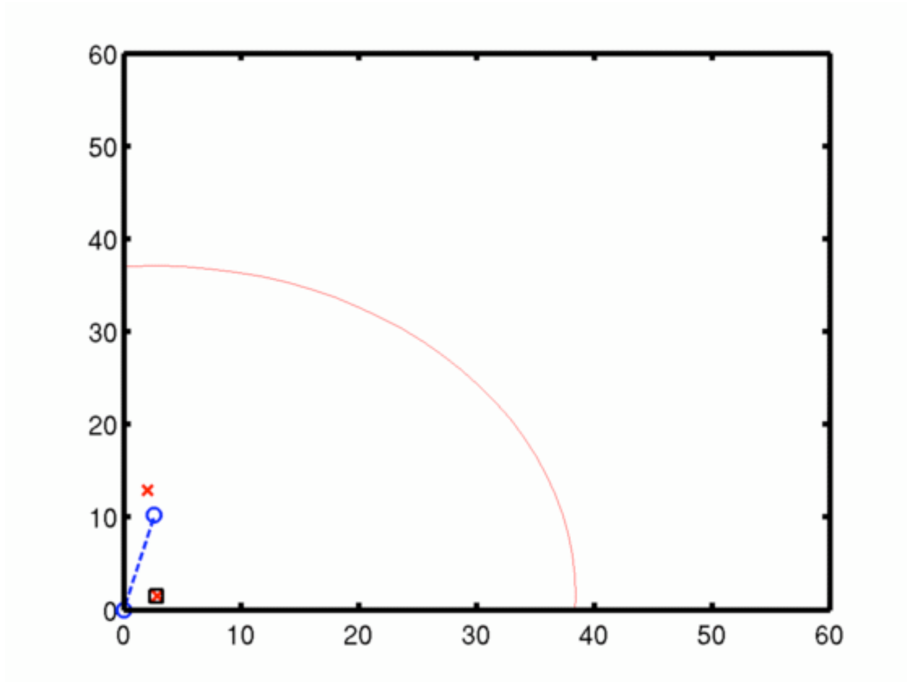


Ball Tracking: Constant Velocity



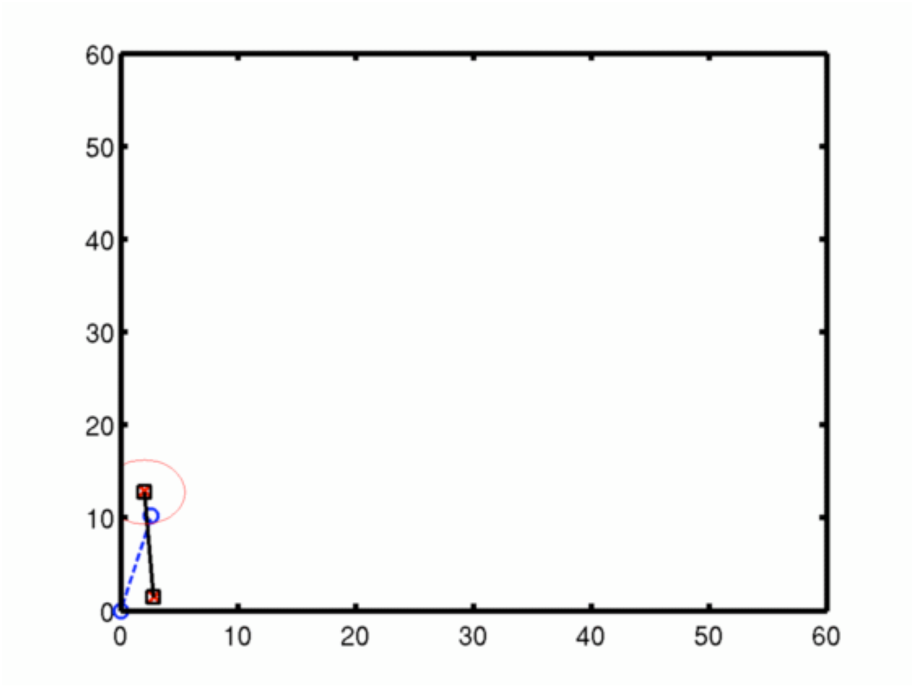
Ball Tracking: Constant Velocity

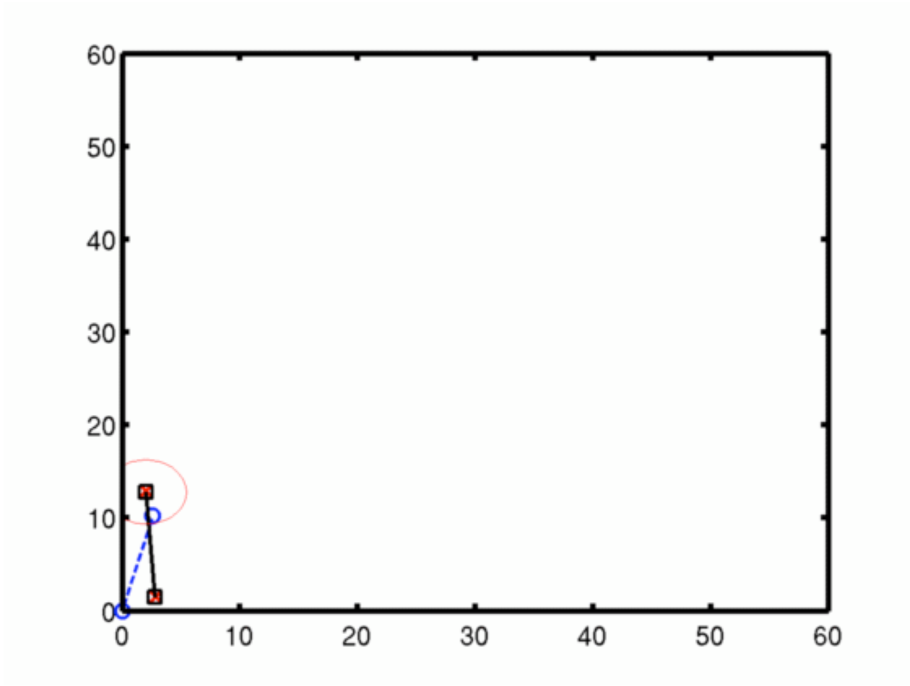




Ground truth Observations State estimate

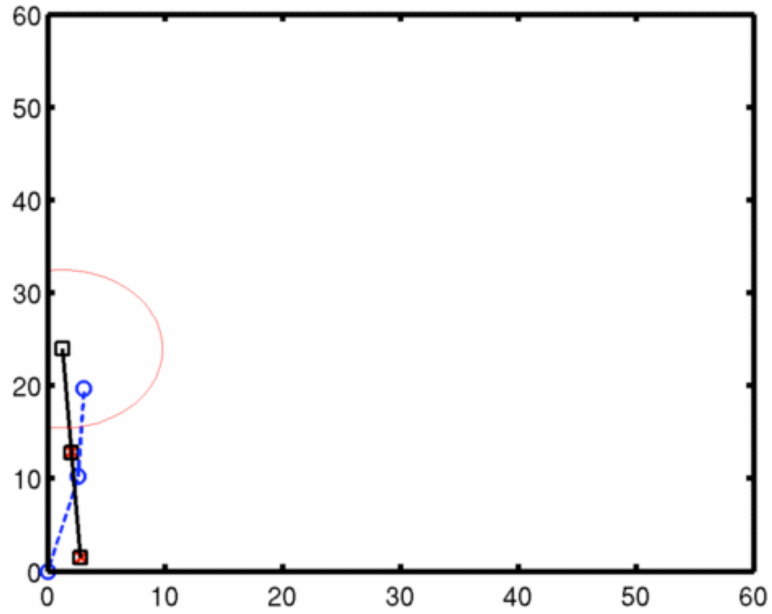
Ball Tracking: Constant Velocity

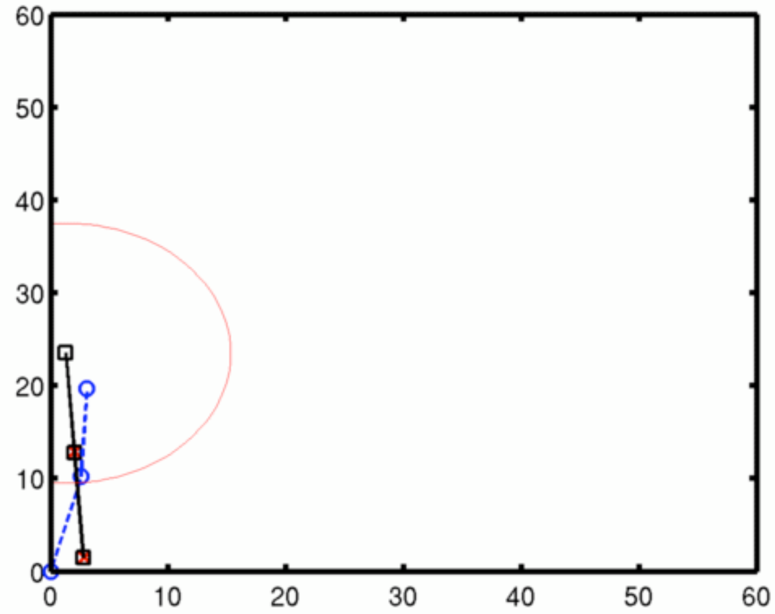




Ground truth   Observations   State estimate

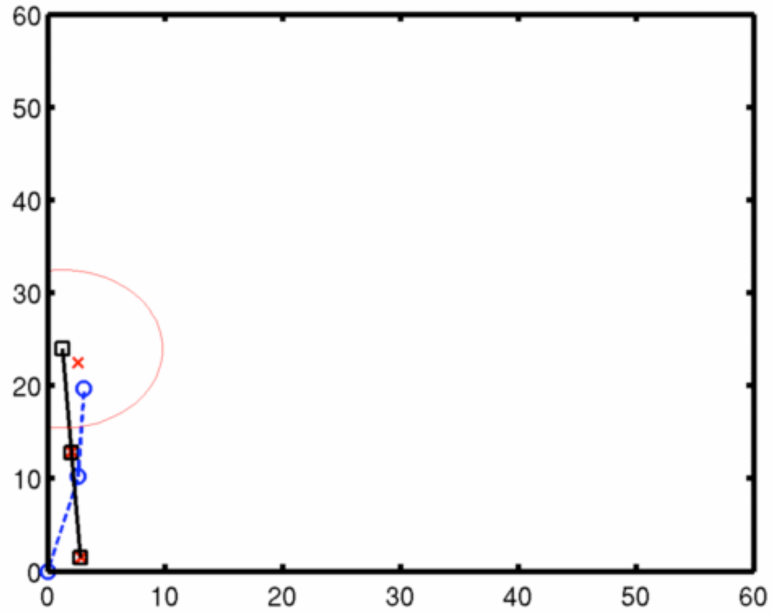
Ball Tracking: Constant Velocity

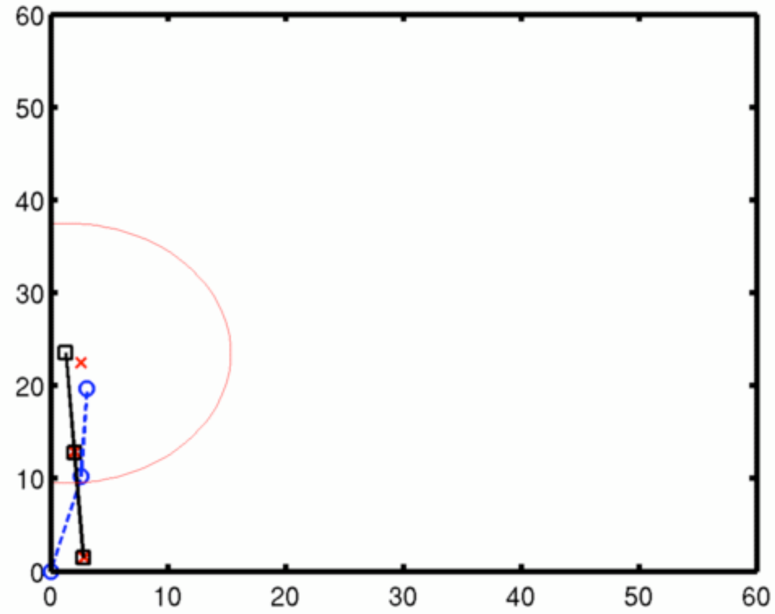




Ground truth Observations State estimate

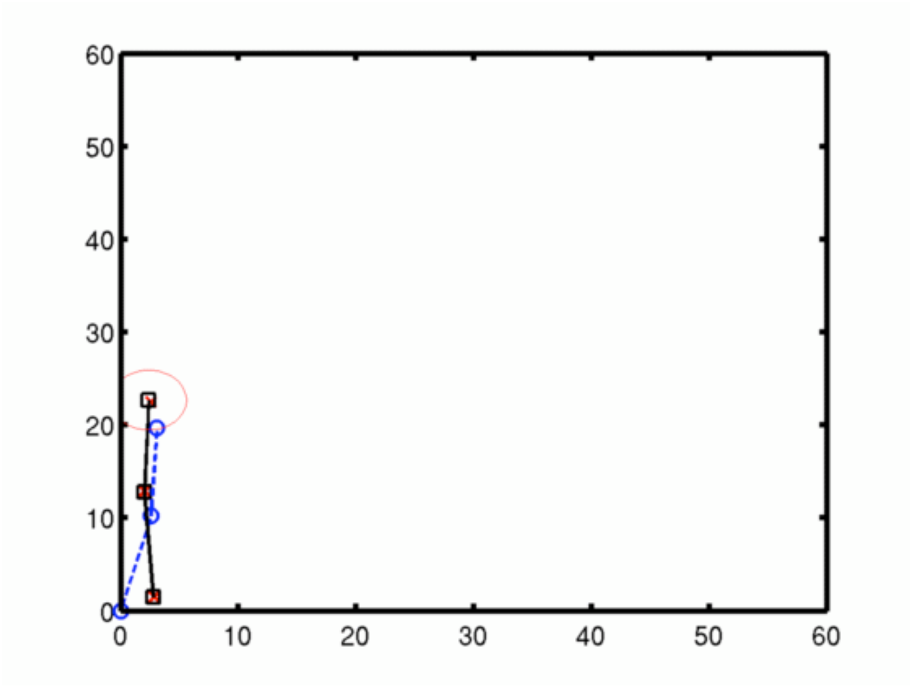
Ball Tracking: Constant Velocity

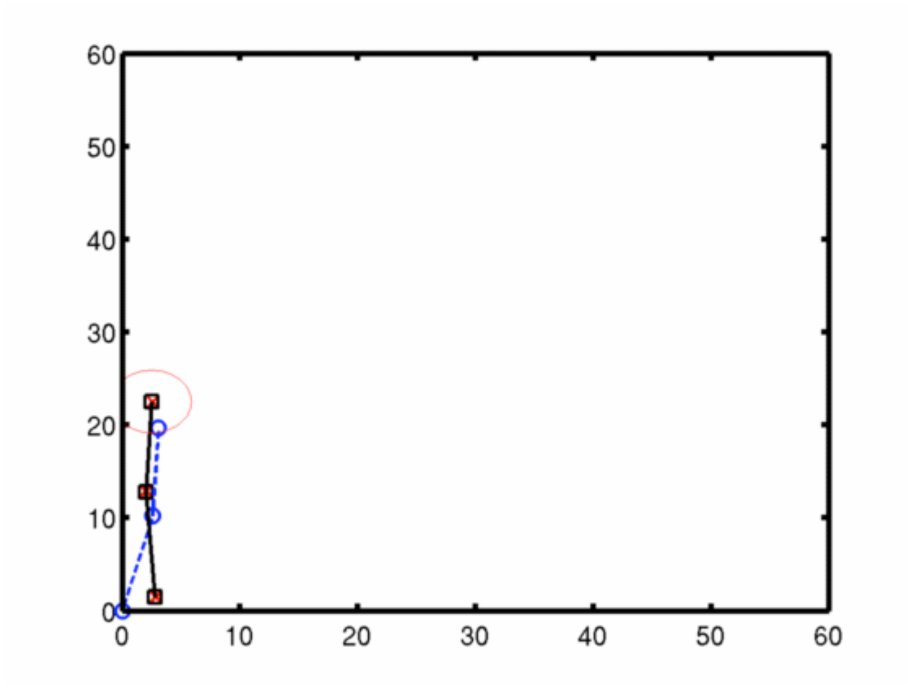




Ground truth   Observations   State estimate

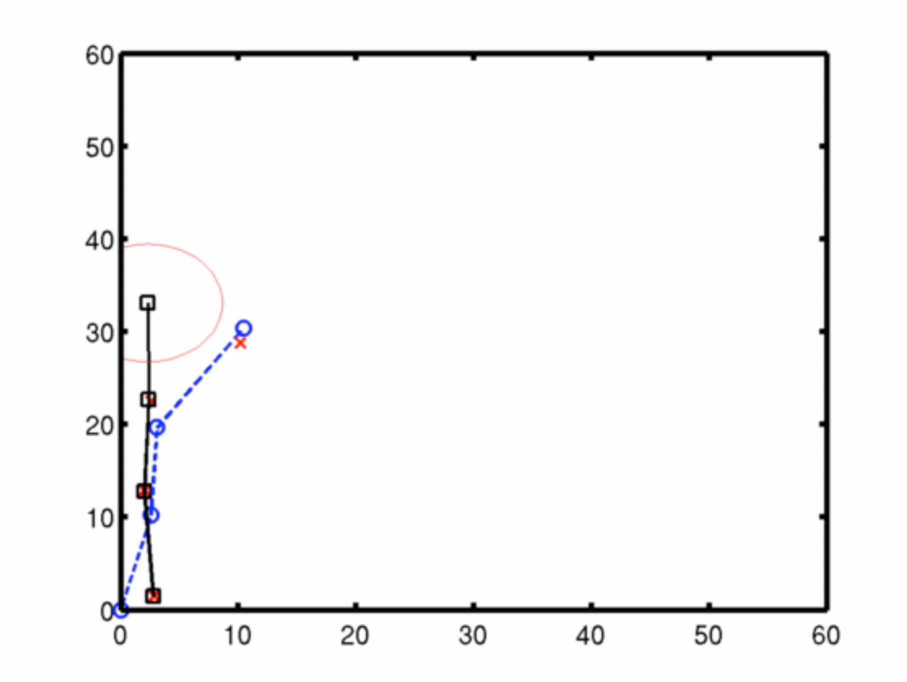
Ball Tracking: Constant Velocity

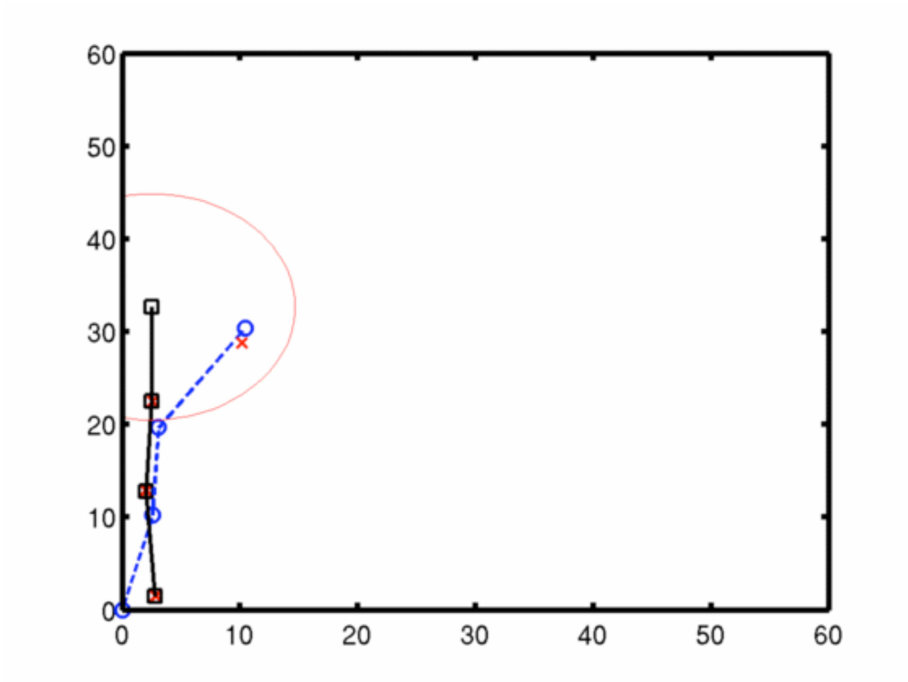




Ground truth Observations State estimate

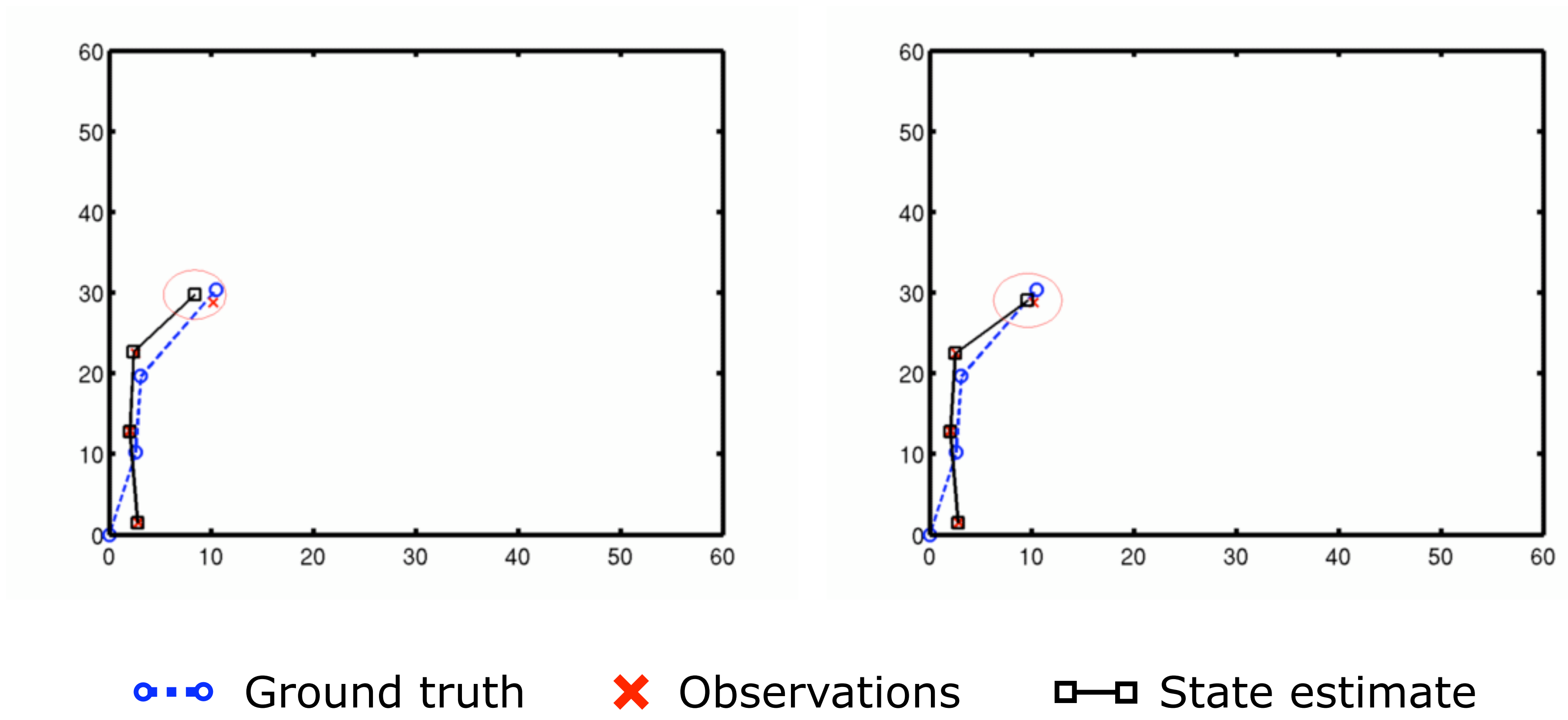
Ball Tracking: Constant Velocity



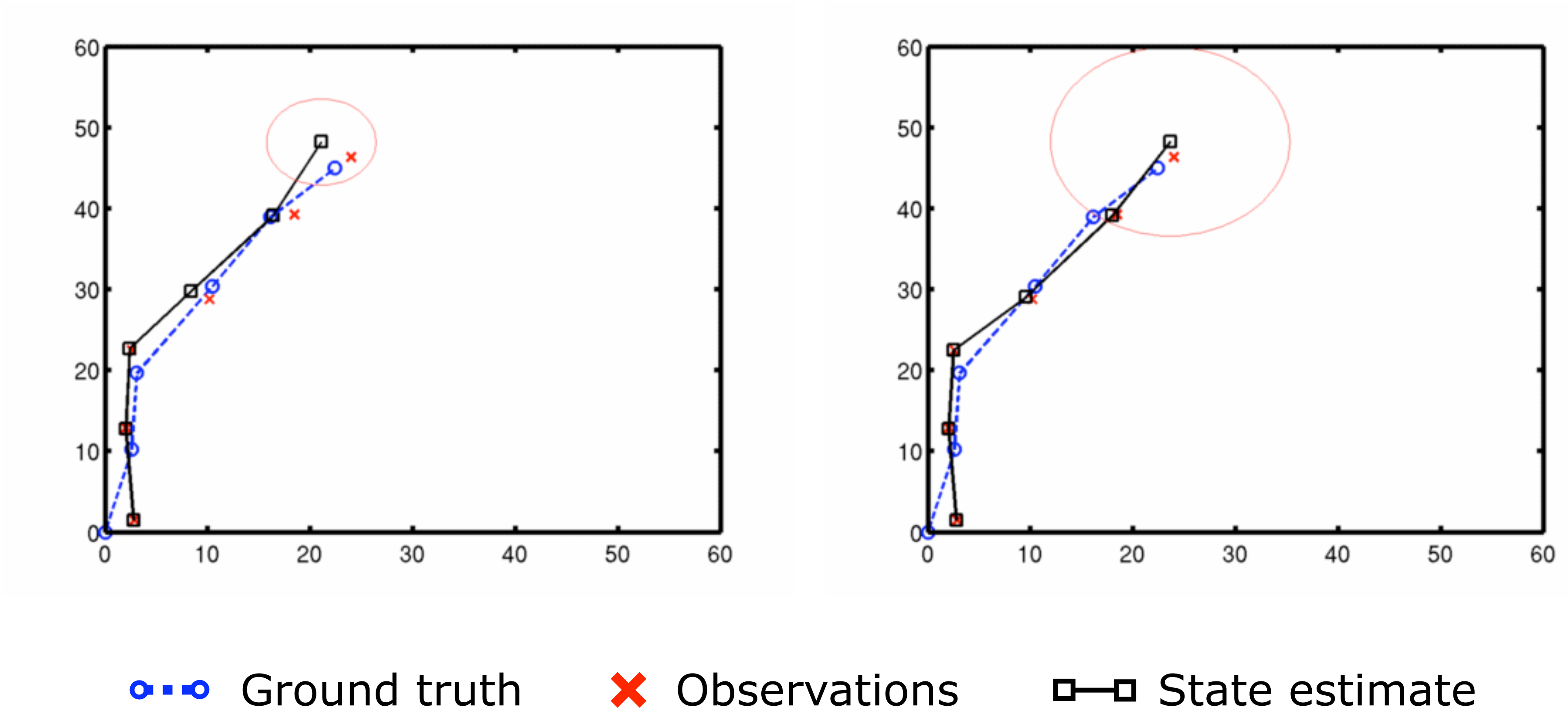


Ground truth Observations State estimate

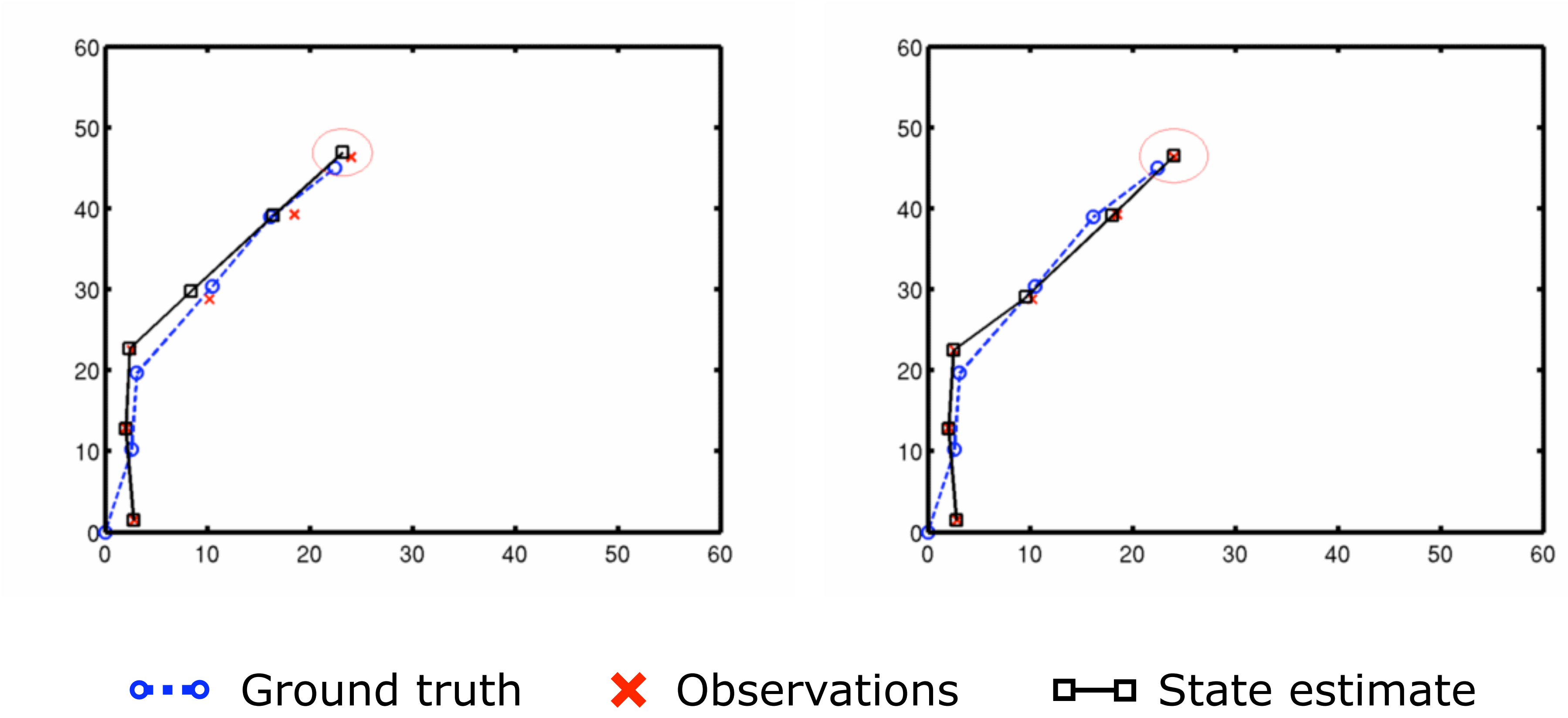
Ball Tracking: Constant Velocity



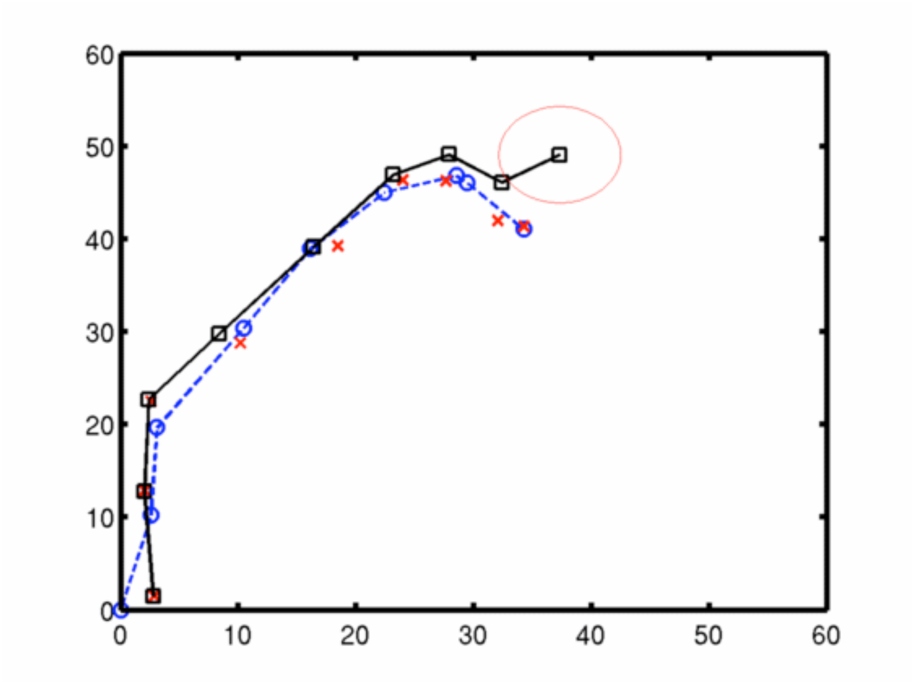
Ball Tracking: Constant Velocity

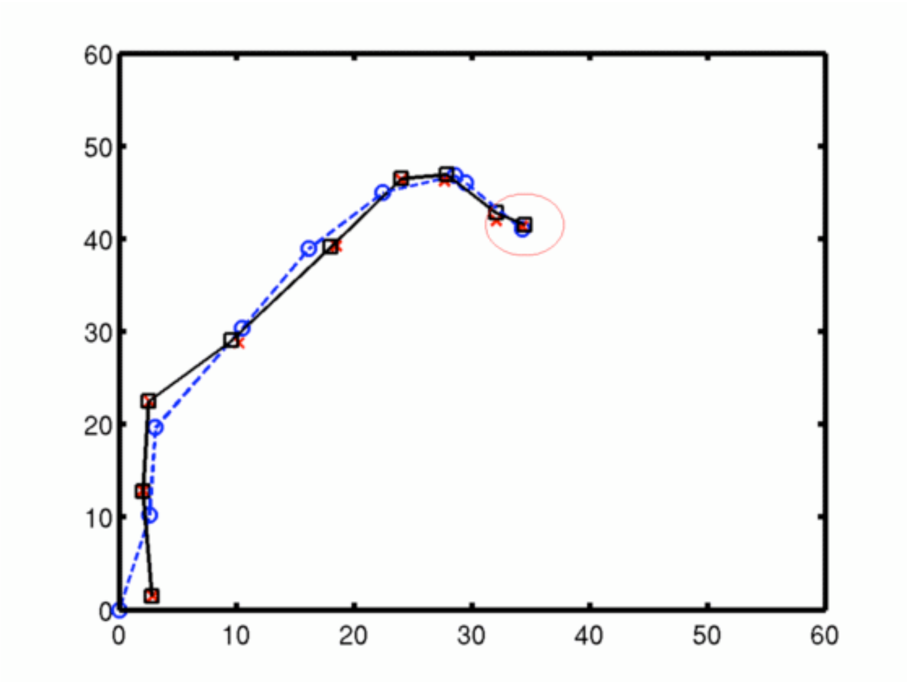


Ball Tracking: Constant Velocity



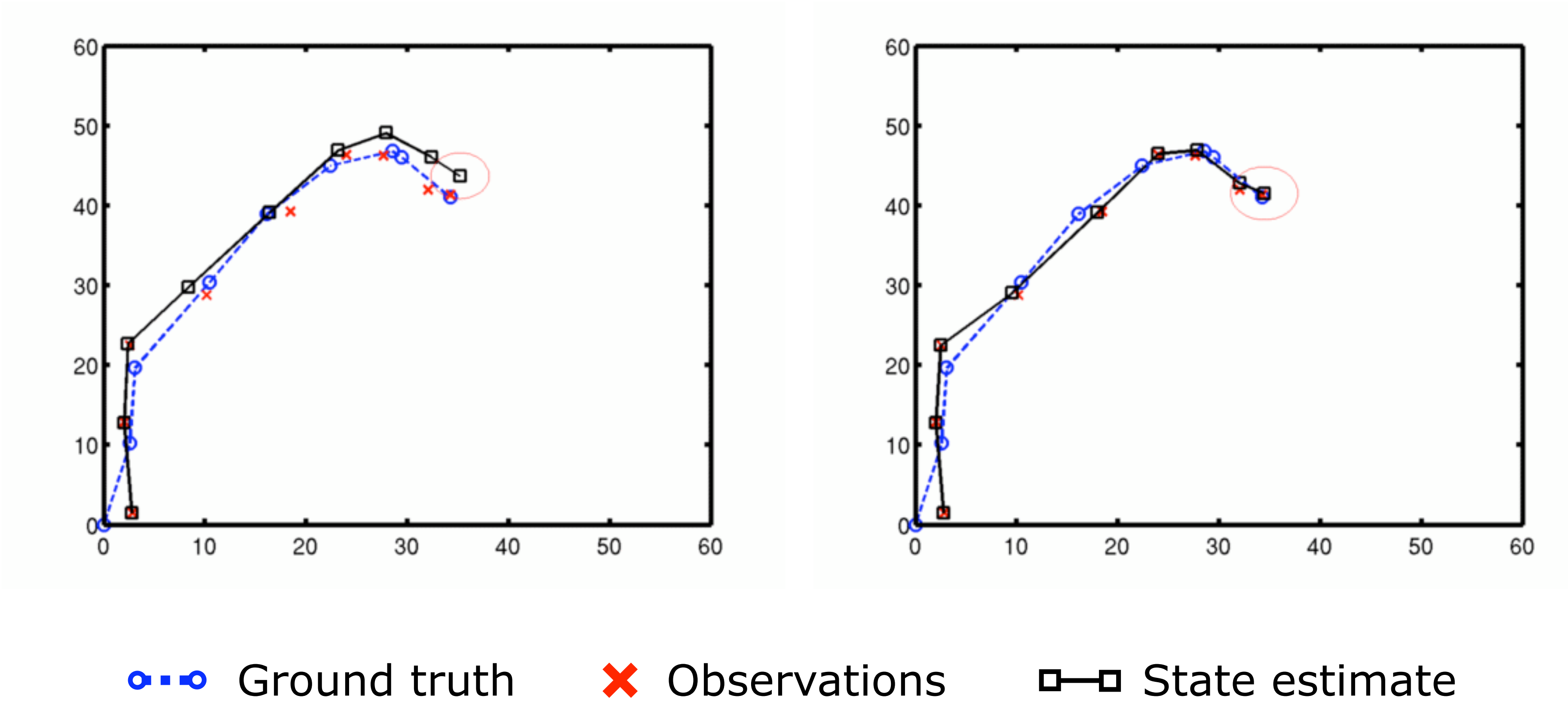
Ball Tracking: Constant Velocity



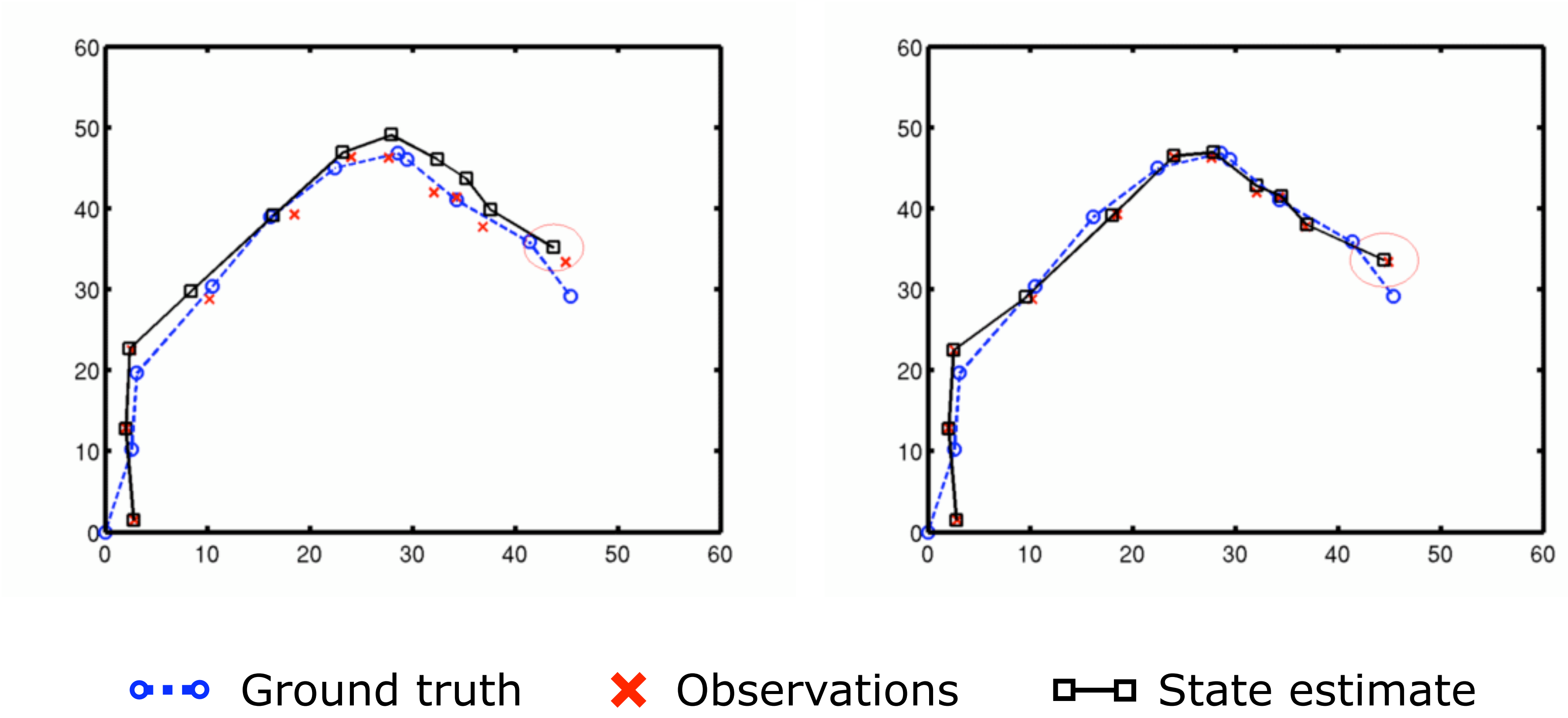


Ground truth Observations State estimate

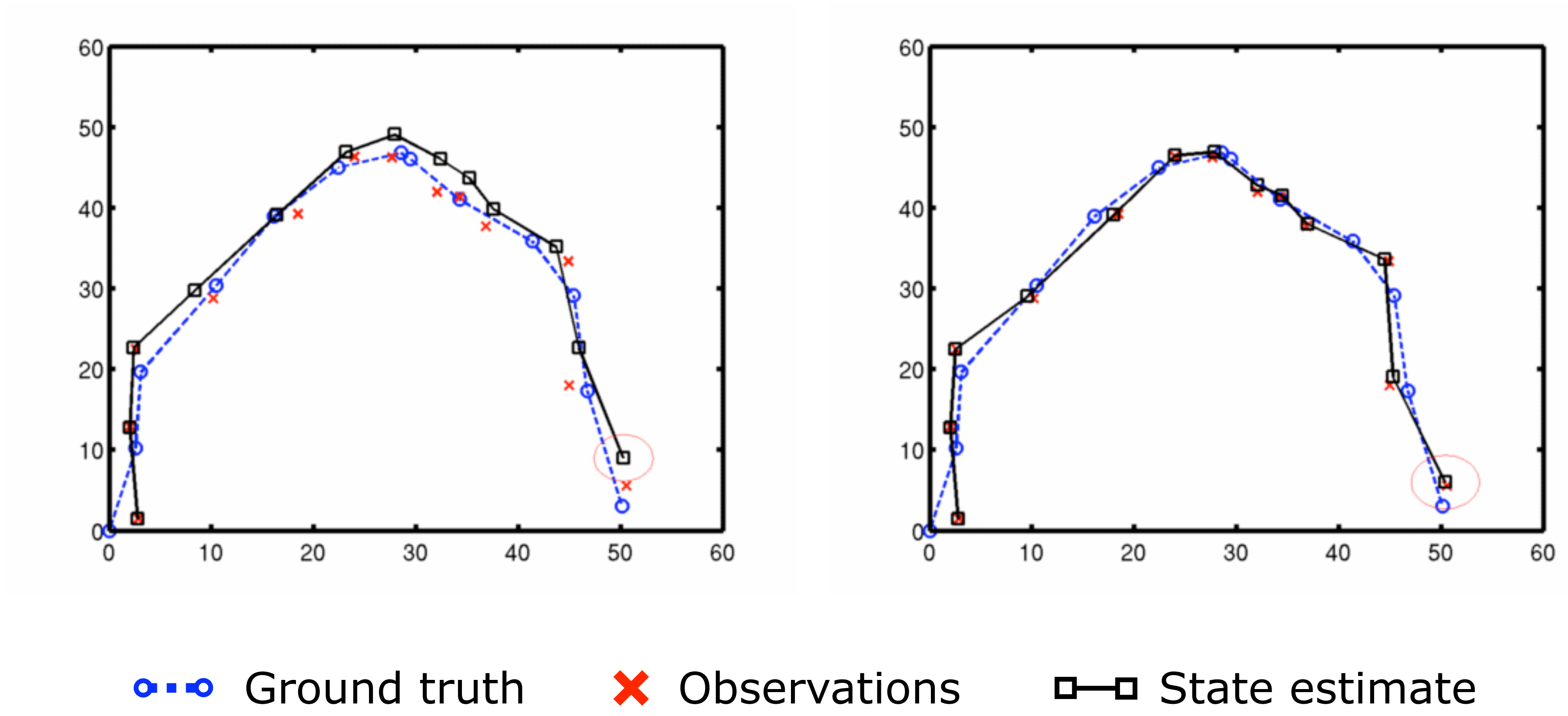
Ball Tracking: Constant Velocity



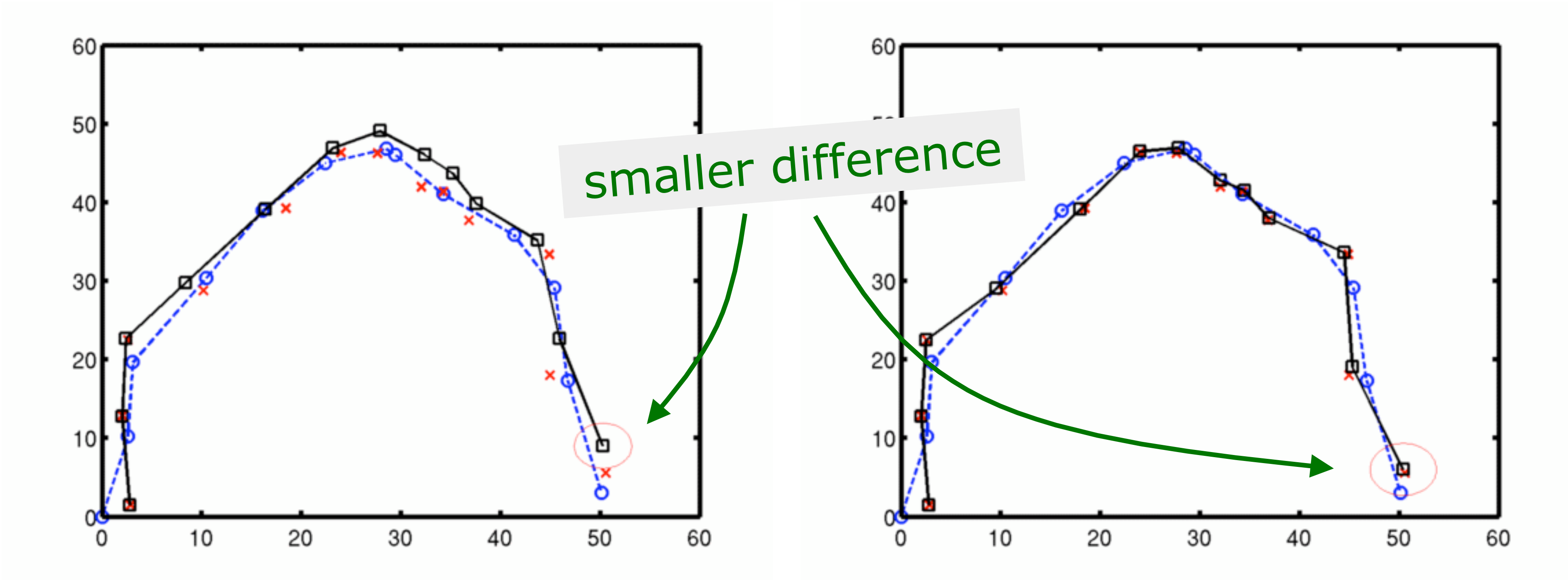
Ball Tracking: Constant Velocity



Ball Tracking: Constant Velocity

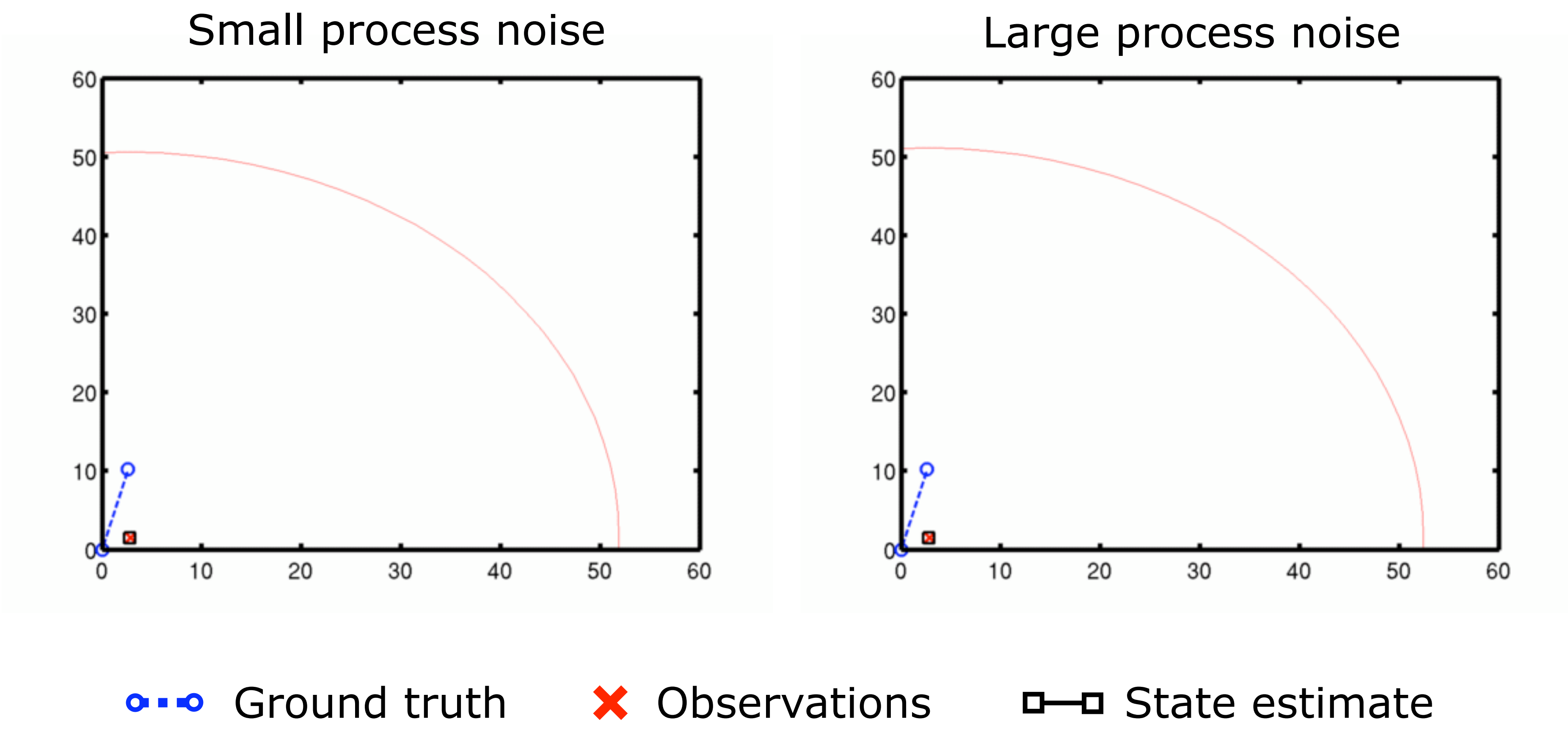


Ball Tracking: Constant Velocity

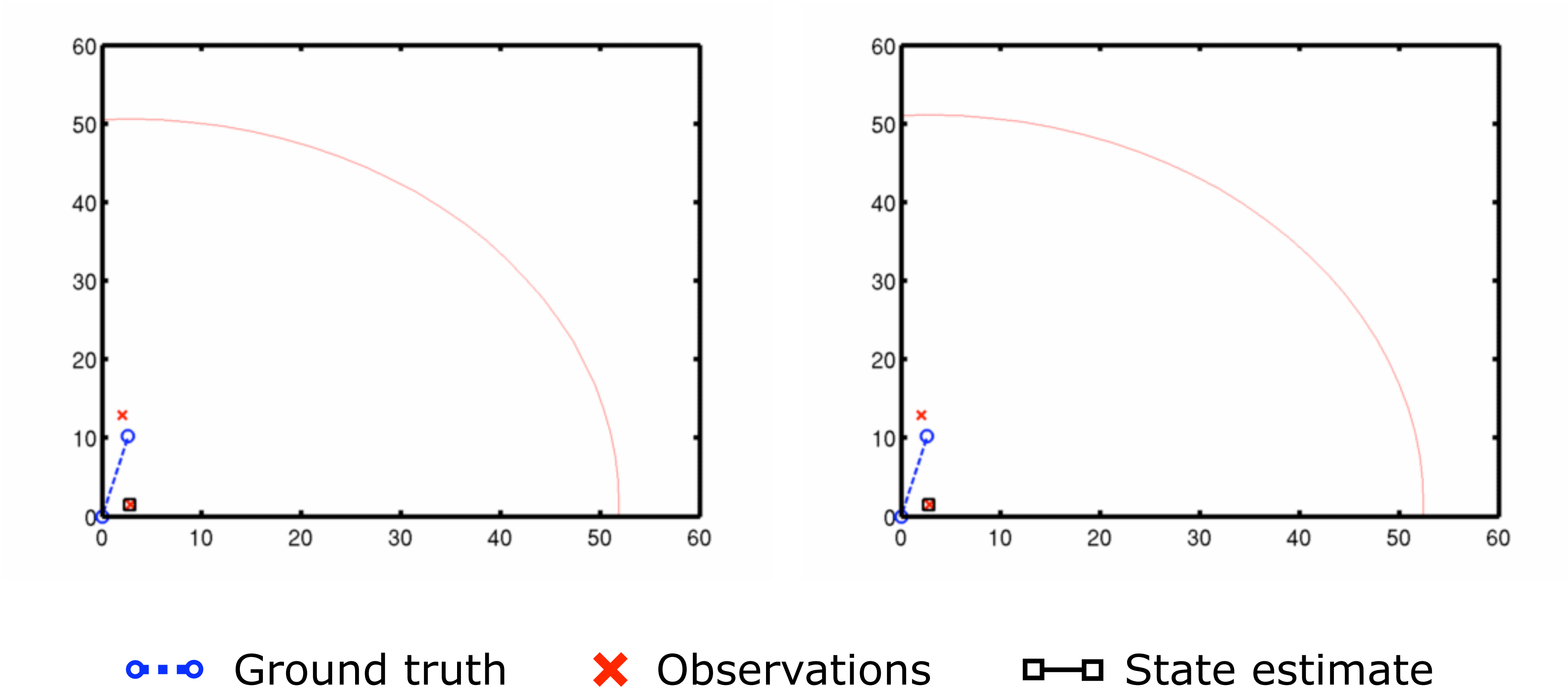


Ground truth　　Observations　　State estimate

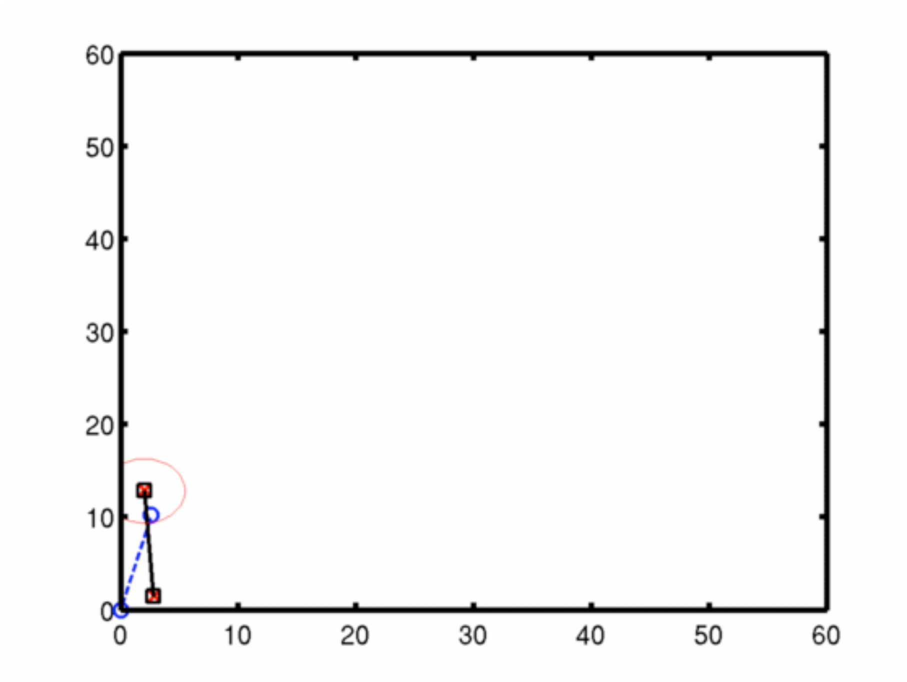
Ball Tracking: Const. Acceleration

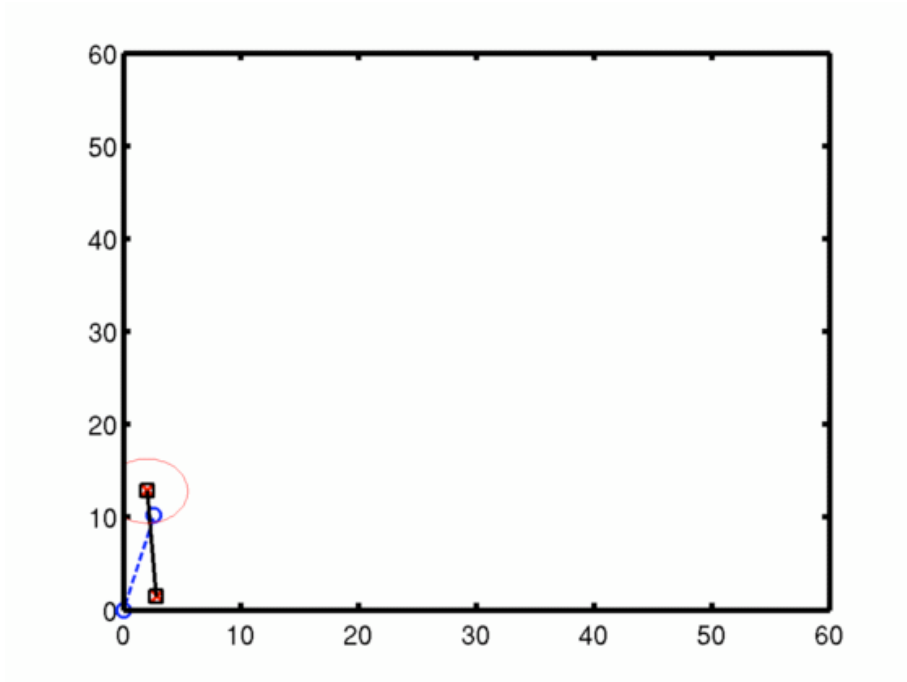


Ball Tracking: Const. Acceleration



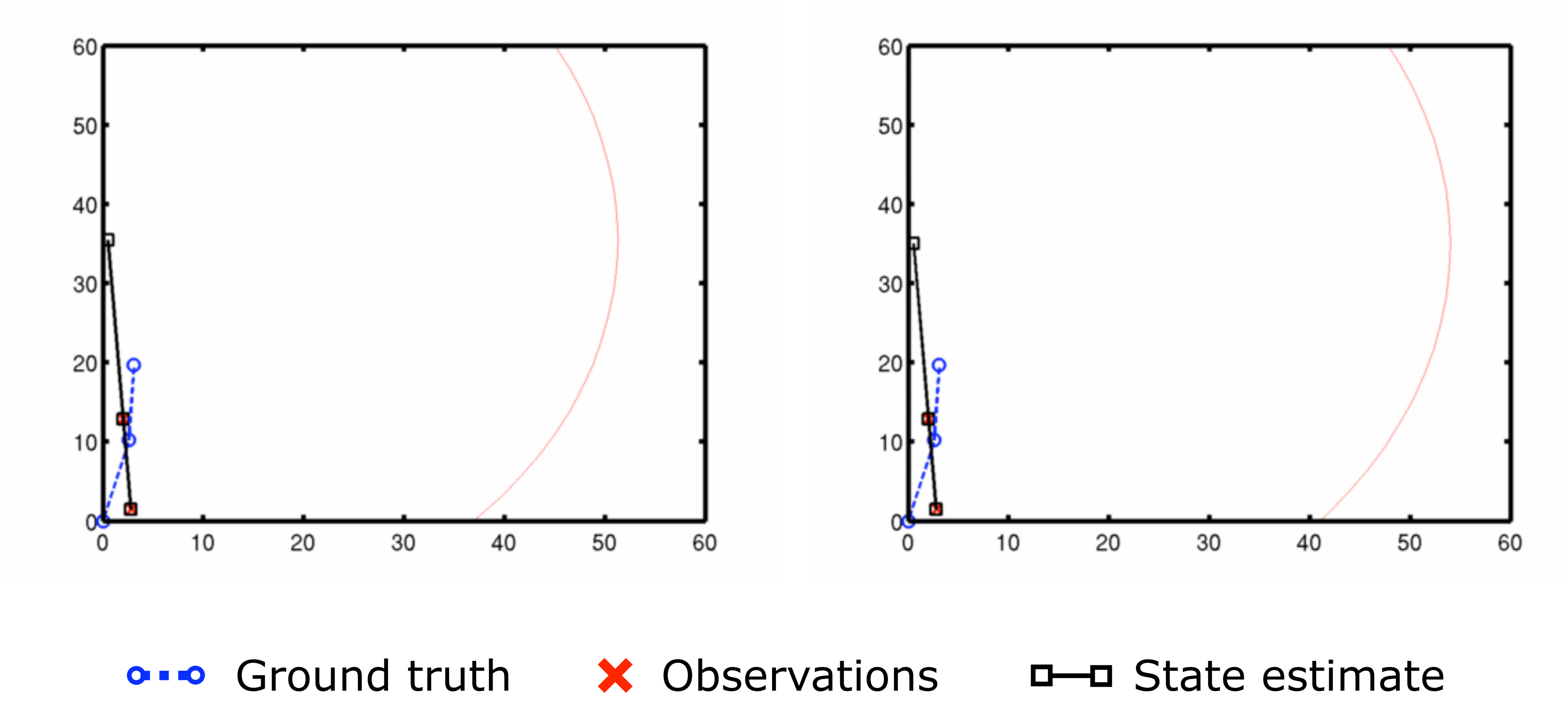
Ball Tracking: Const. Acceleration



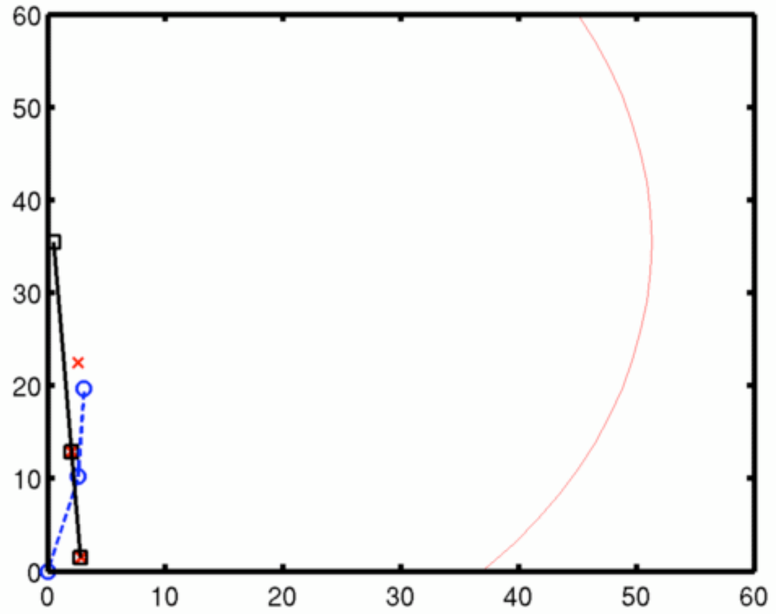


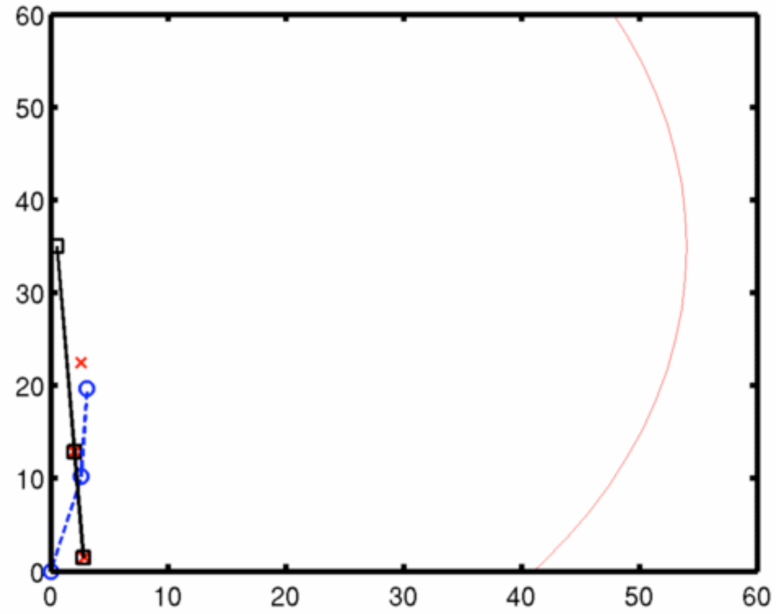
Ground truth Observations State estimate

Ball Tracking: Const. Acceleration



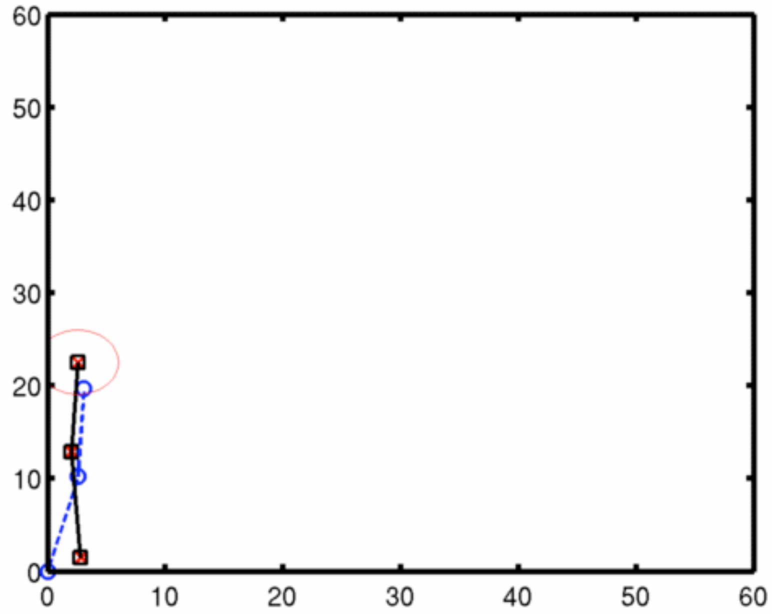
Ball Tracking: Const. Acceleration

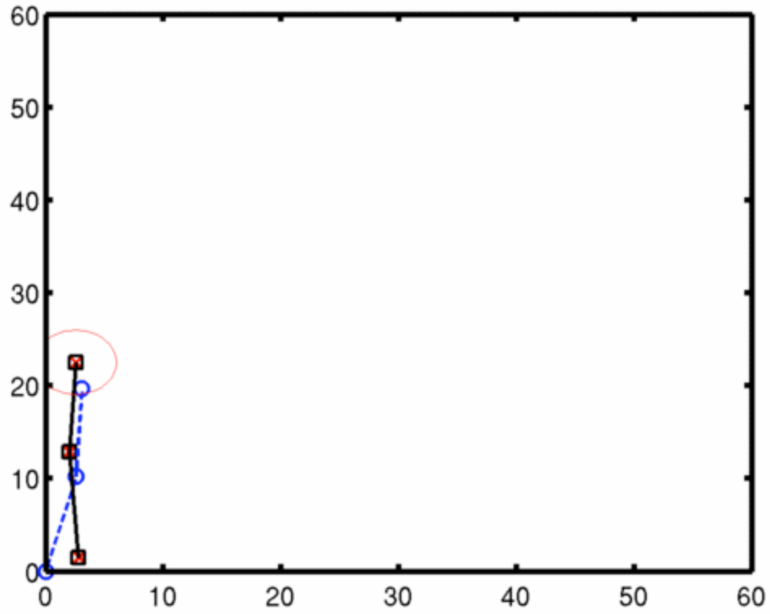




Ground truth Observations State estimate

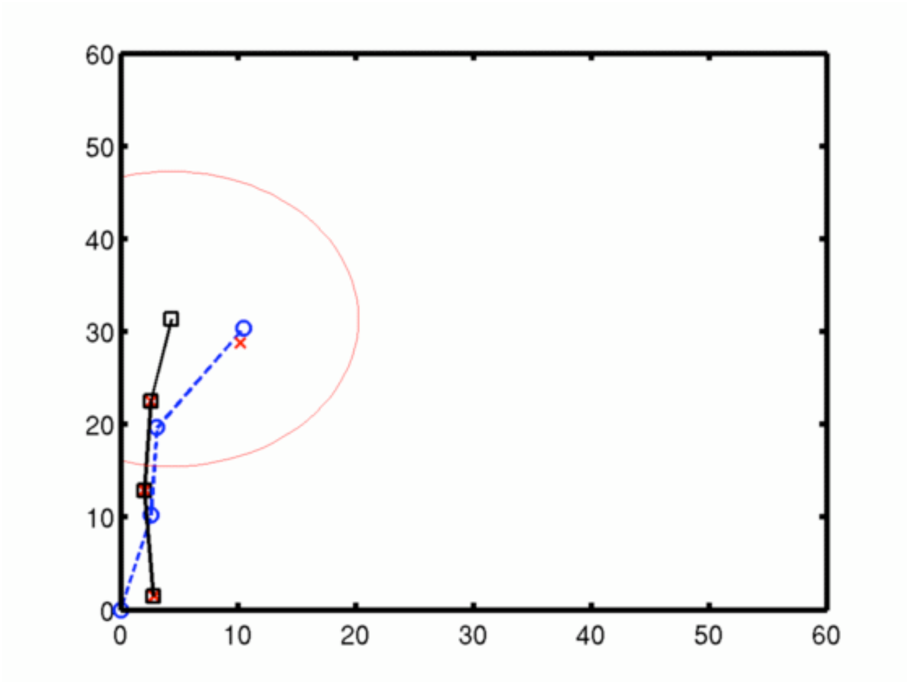
Ball Tracking: Const. Acceleration

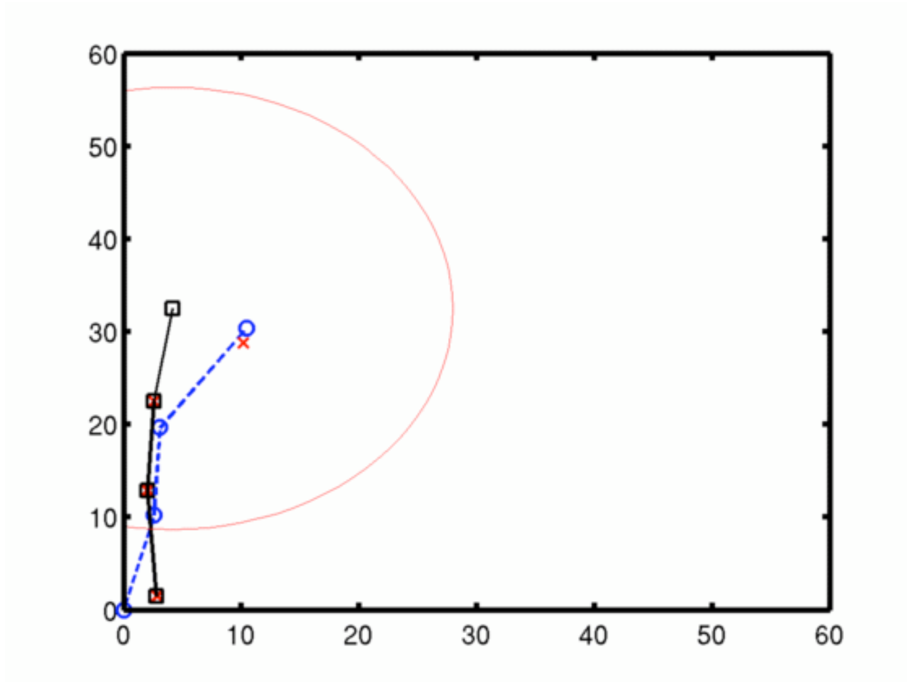




Ground truth Observations State estimate

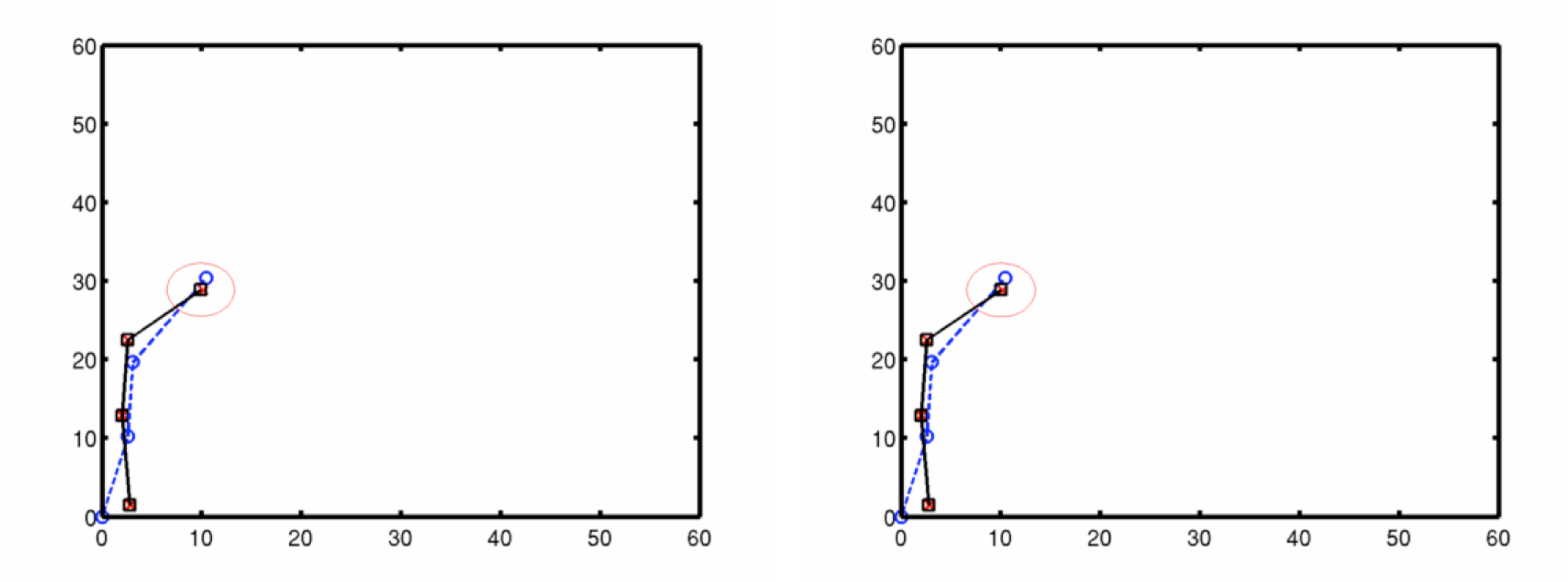
Ball Tracking: Const. Acceleration





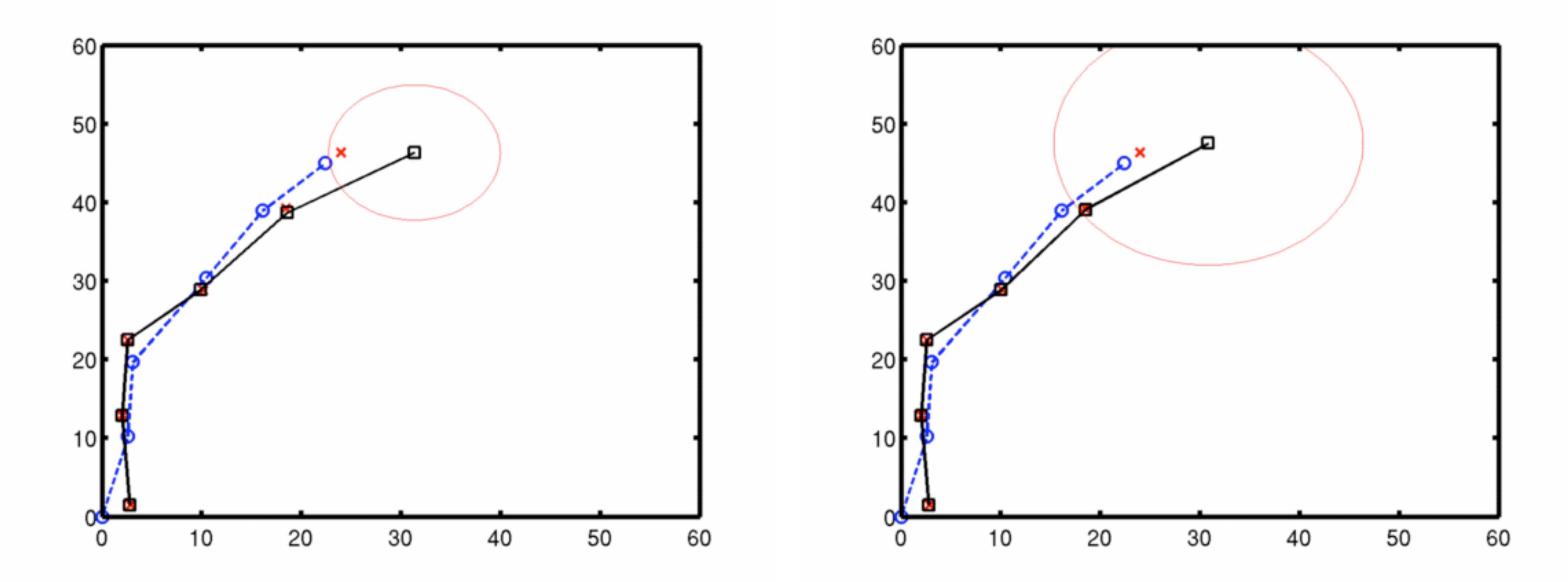
Ground truth Observations State estimate

Ball Tracking: Const. Acceleration



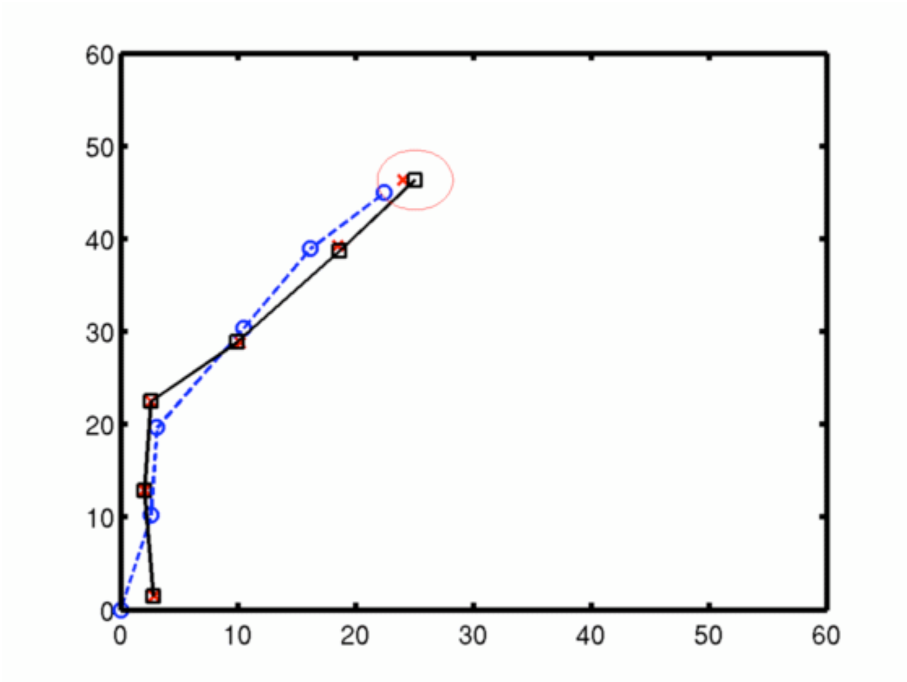
Ground truth   Observations   State estimate

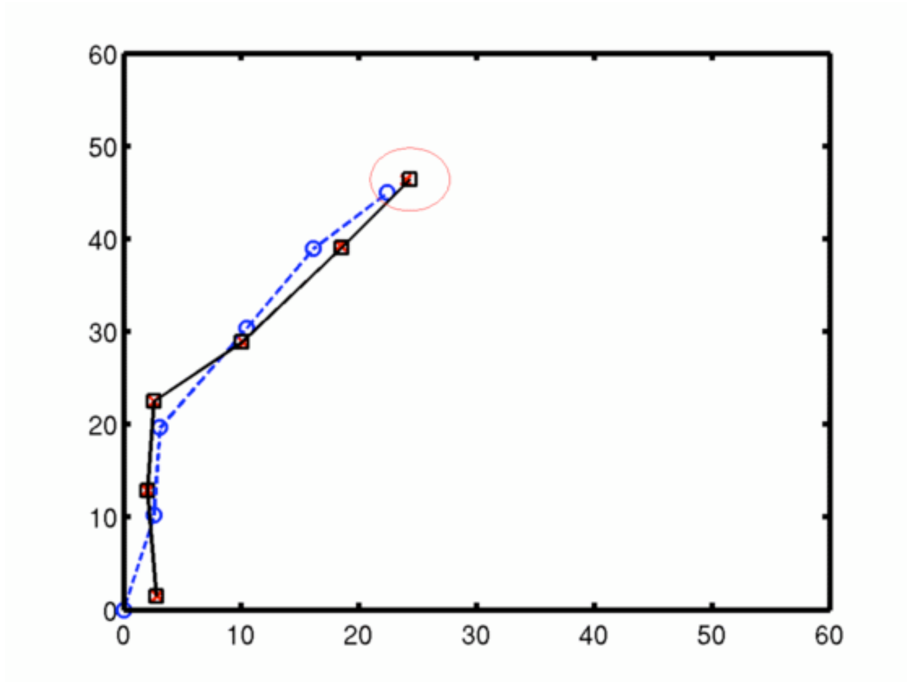
Ball Tracking: Const. Acceleration



Ground truth   Observations   State estimate

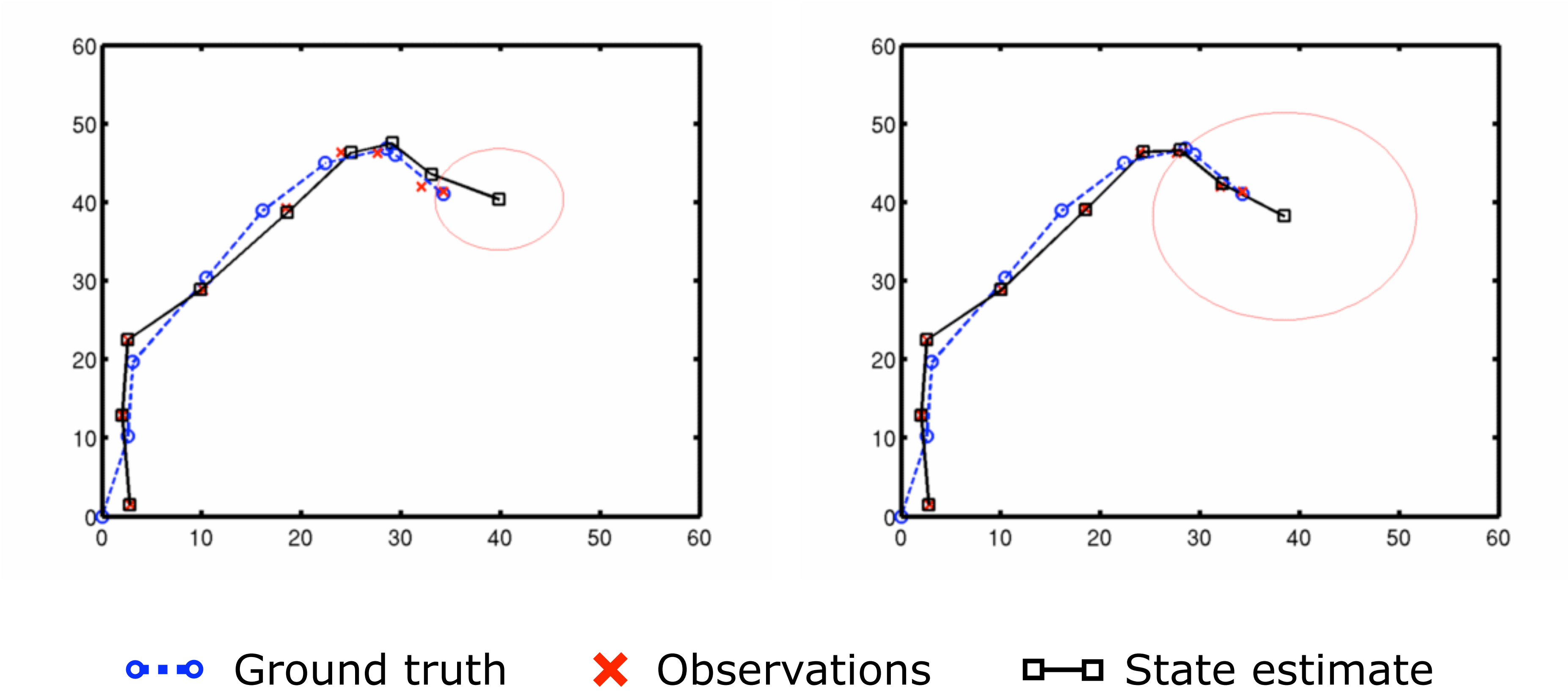
Ball Tracking: Const. Acceleration



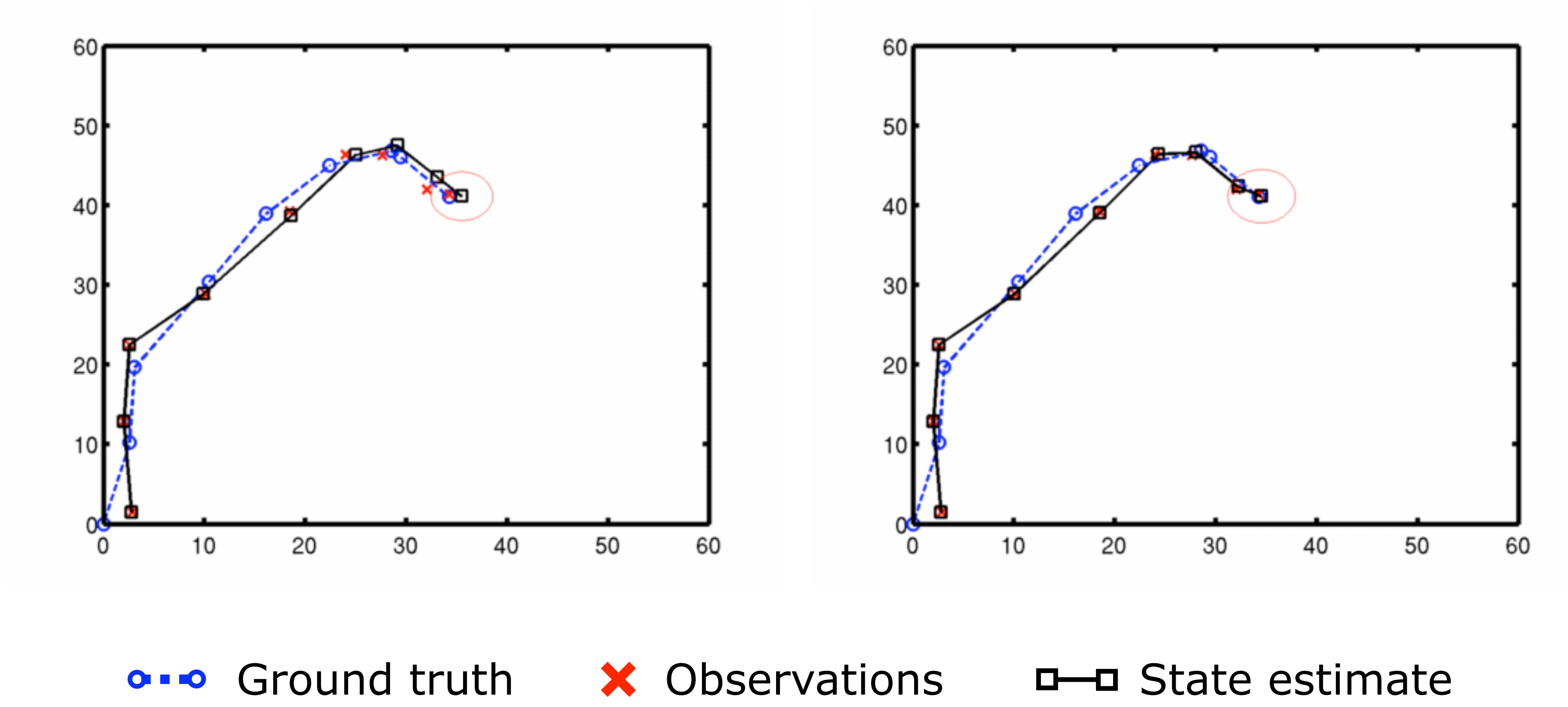


Ground truth   Observations   State estimate

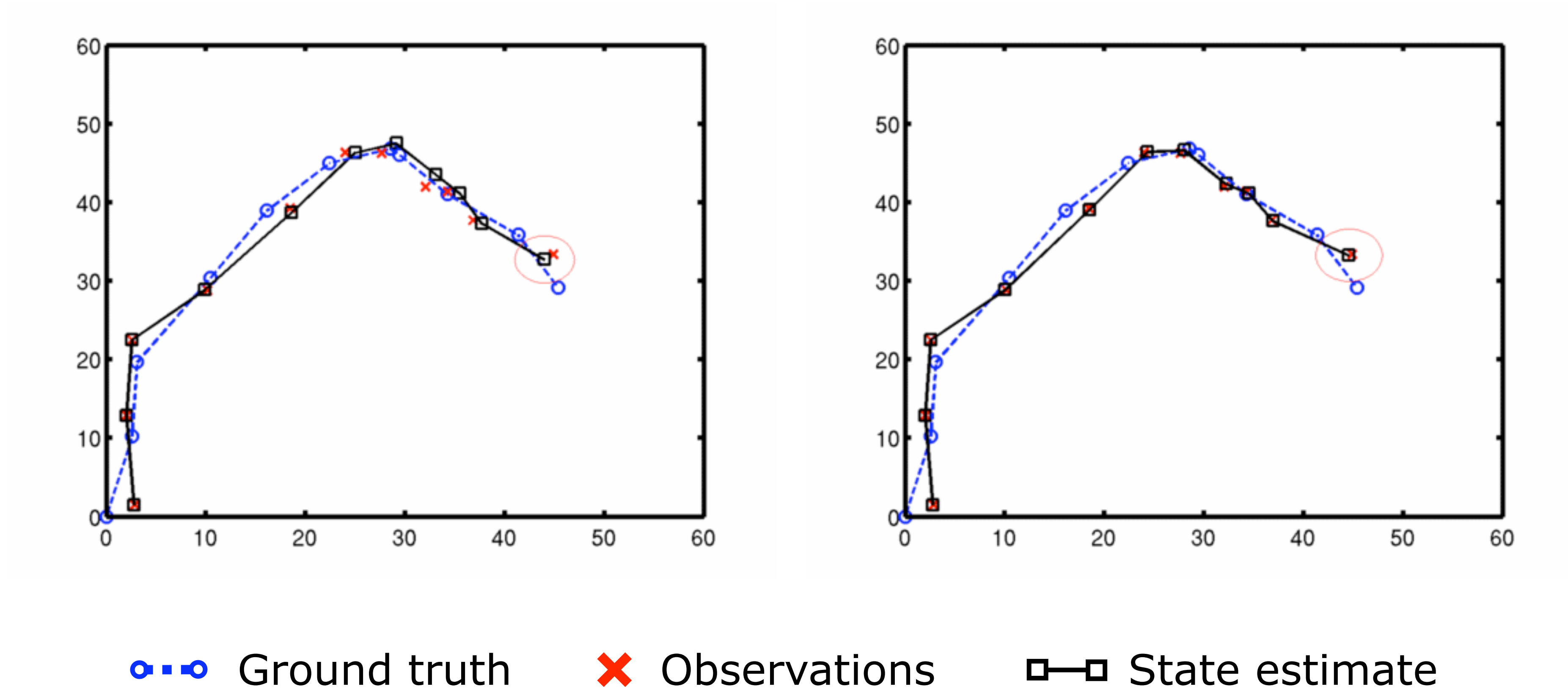
Ball Tracking: Const. Acceleration



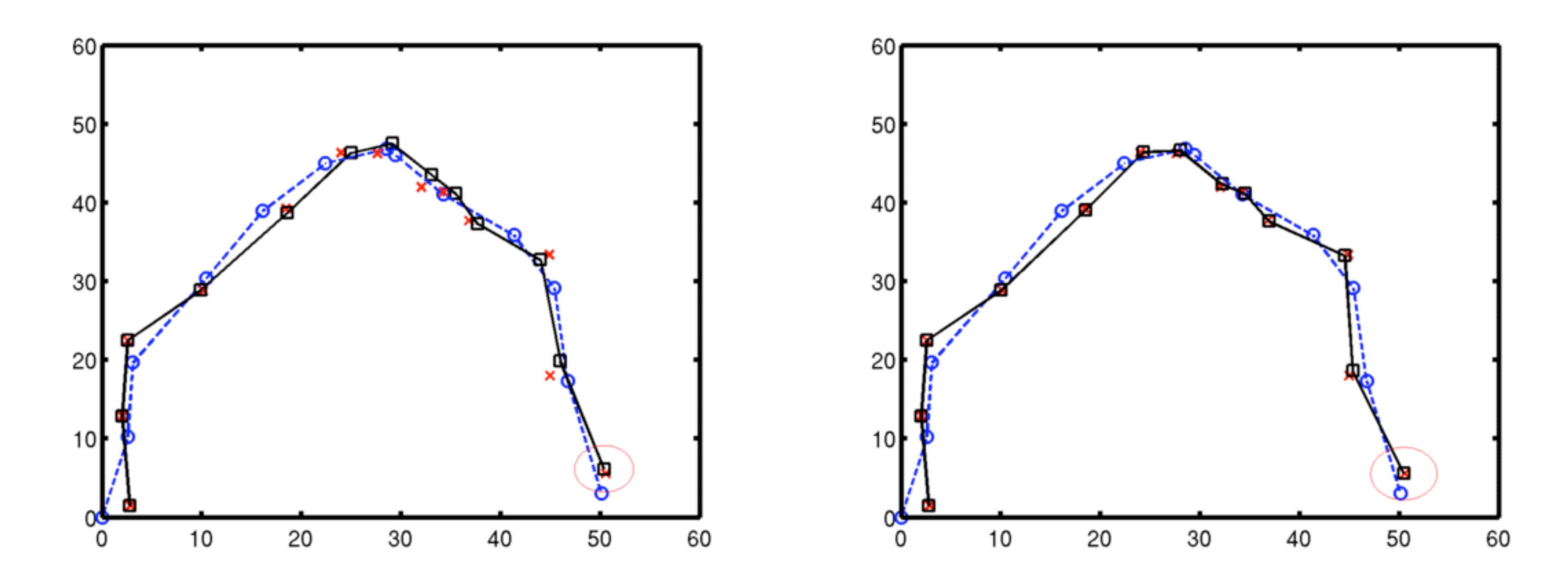
Ball Tracking: Const. Acceleration



Ball Tracking: Const. Acceleration

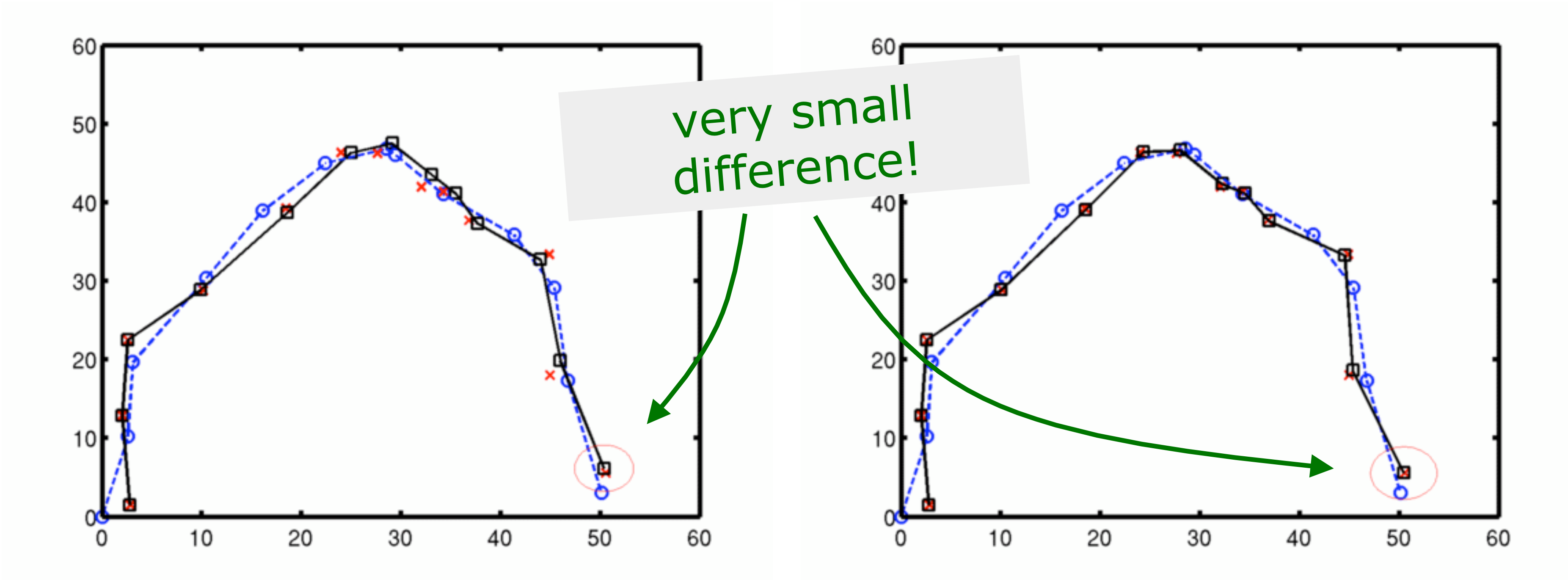


Ball Tracking: Const. Acceleration



Ground truth   Observations   State estimate

Ball Tracking: Const. Acceleration

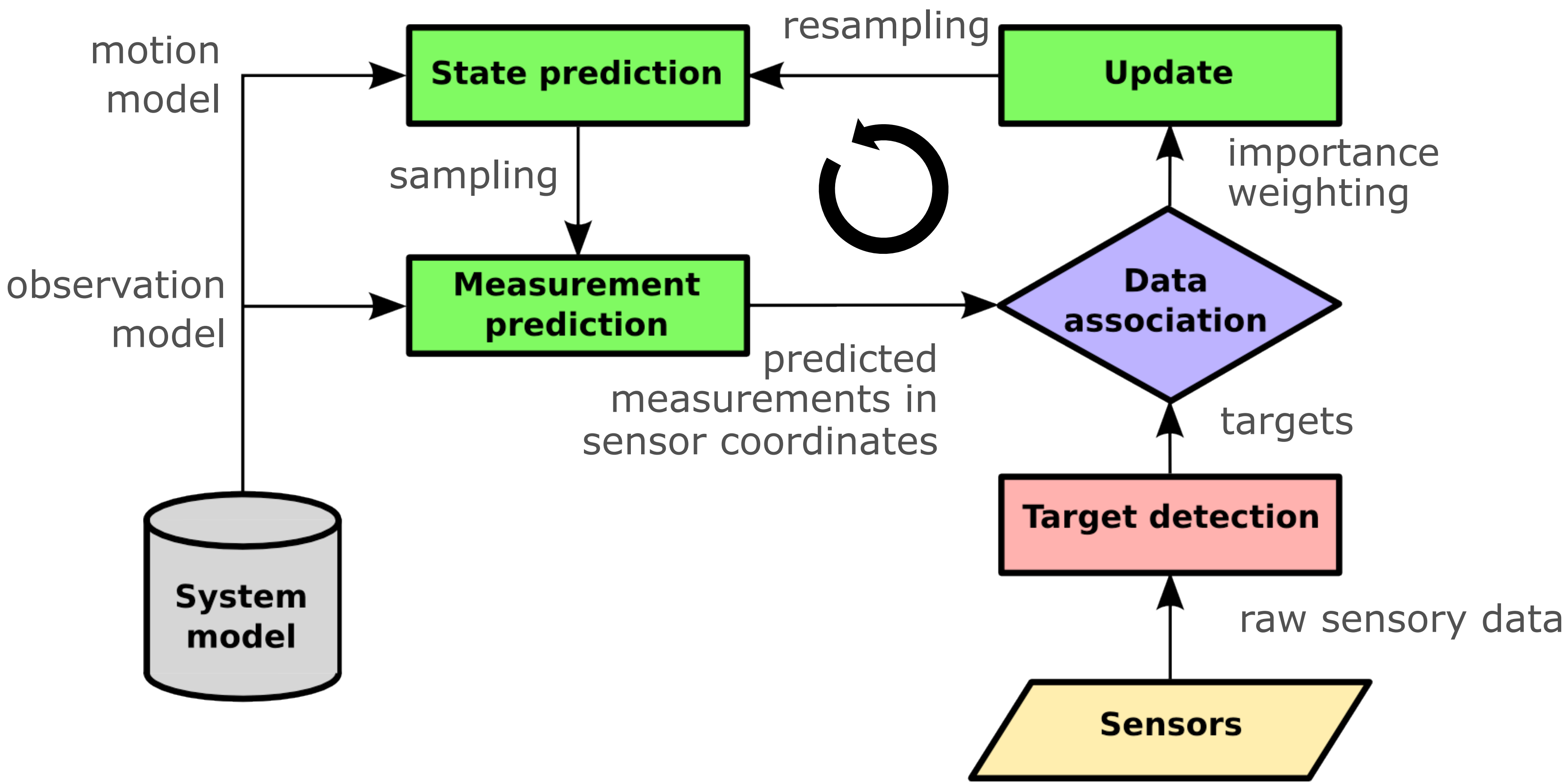


Ground truth　Observations　State estimate

Ball Tracking: Wrap Up

- The better the motion model, the better the tracker is able to **follow the target**
- A large process noise covariance can partly **compensate a poor motion model**
- But: large process noise covariances cause the validation gates to be large, which in turn, creates a **high level of ambiguity for data association** in case of multiple targets
- In other words: the tracker can’t tell which is which target anymore because they are all statistically compatible with the observations

Tracking cycle – Particle filter



KF Cycle: Measurement Predict.

- **Measurement prediction**

$$\widehat{z}(k) = H(k)\widehat{x}(k+1|k)$$

$$\widehat{S}(k) = H(k)\widehat{P}(k+1|k)H^T(k) + R(k)$$

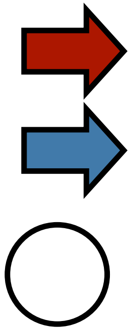
- **Observation**
  Typically, only the target **position** is observed.
  The measurement matrix is then

$$\mathbf{x} = \begin{bmatrix} x & y & \mathbf{s}^T \end{bmatrix}^T \qquad H = \begin{bmatrix} 1 & 0 & \mathbf{0} \\ 0 & 1 & \mathbf{0} \end{bmatrix}$$

  Note: One can also observe (not in this course)
  - Velocity (Doppler radar)
  - Acceleration (accelerometers)

Some Shapes & Numbers

- Blue is RGB: 51, 102, 153
- Red is RGB: 154, 0, 0



test