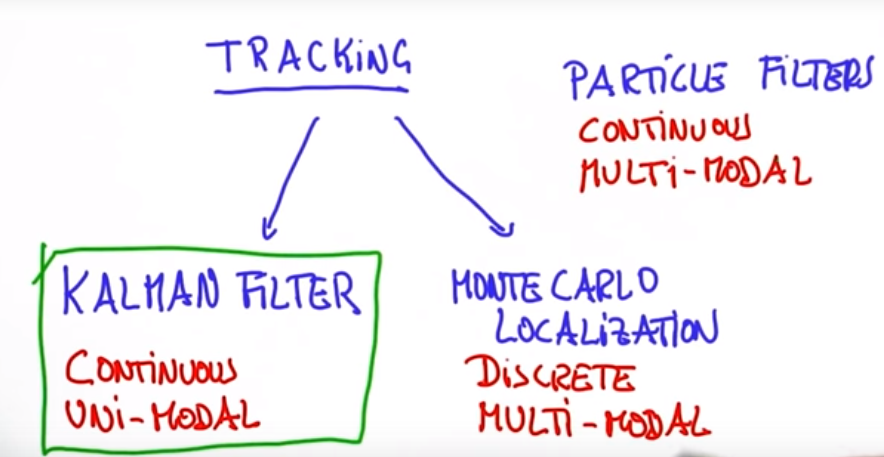
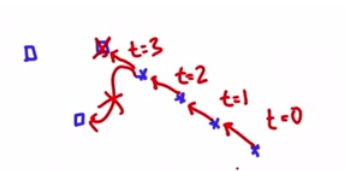
**Kalman Filters**

LIDAR gets distance measurements 10 times a second of about 1 million data points. This large amount of data is required to accurately place objects and your environment.

Gaining Data about where vehicles currently are is important for localization. It gets us information of other objects are so that there is no collision. However we also need to know how fast they are moving. That way we can use this information to predict where cars are going to be, in order to avoid collisions and hazardous situations. It’s important for all objects, whether it’s a car, a bike, or a pedestrian.

**Kalman Filters** are a very popular technique used to estimate the state of a system. Kalman Filters estimate a **continuous state**, and give us a **Uni-Modal distribution**. Finding out what state you are in in **discrete** states is done with **Monte Carlo Localization**.



If an object is being tracked at the given positions, and we need to estimate the position at t=4, we can assume that given the velocity of the data give, the prediction would follow in the same direction. These are the types of observations a Kalman Filter has. It estimates future positions, and velocities. Kalman filters are useful for these continuous situations.

**Gaussian**

In Kalman Filters, the probability distribution (of estimates) is given by a Gaussian. A Gaussian is a continuous function that has an area underneath that sums up to 1.

The Gaussian is characterized by the mean and the variance (In 1-D).

Kalman filters want to maintain a mean and variance as the best estimate of the location we are trying to find.



Gaussians have a single peak (Uni-modal), and are symmetric.

Variance is a good measure of uncertainty. The larger it is, the more uncertain we are about the actual state the Kalman Filter is estimating.

This means we want a Gaussian that is thin, meaning the uncertainty is low for a state.

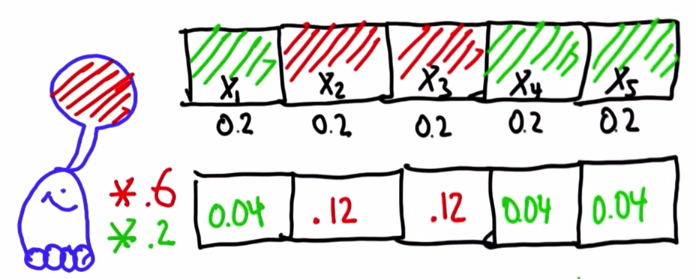
To find the max value of a Gaussian, choose the value of X equal to the mean, making the exponential value 1, and just leaving the magnitude.

**A Kalman Filter represents the distributions by Gaussians and iterates on two main cycles.**

**The First Cycle is the Measurement Update.**

It uses Bayes Rule, Which requires a product.

Example. Given an observation of red, and 5 possible states of which 3 are green and 2 are red. What is the probability that we are in each state given we see red. This can be solved with Bayes Rule. A much larger value (0.6) is chosen for red due to the measurement of Red being seen.



These values are then normalized to add up to 1. They are normalized by the sum of all the current values.

After they are normalized, we have a proper probability distribution.

**The Second Cycle is the Motion Update.**

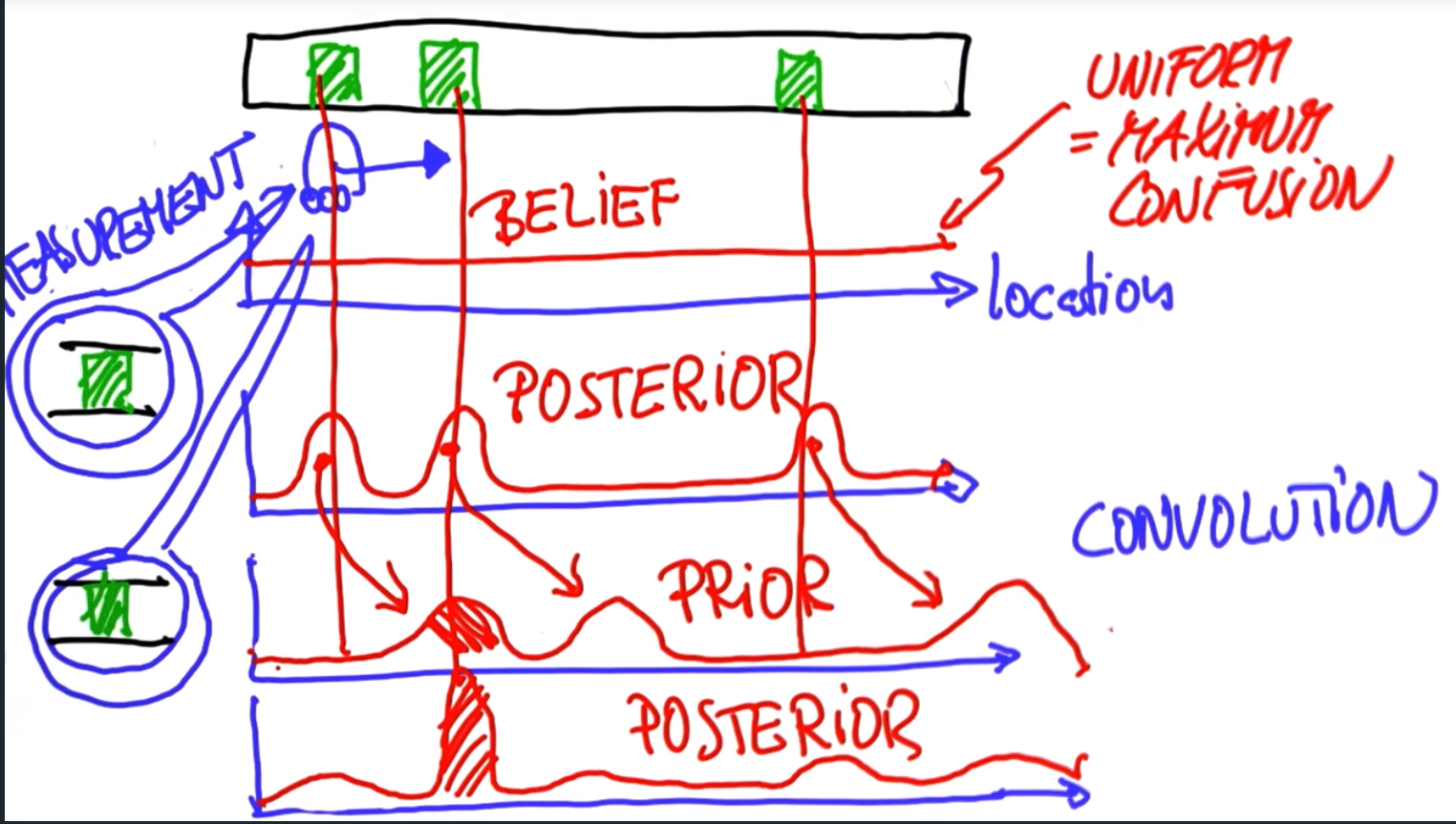
This uses Total Probability, which involves convolution.

Example.

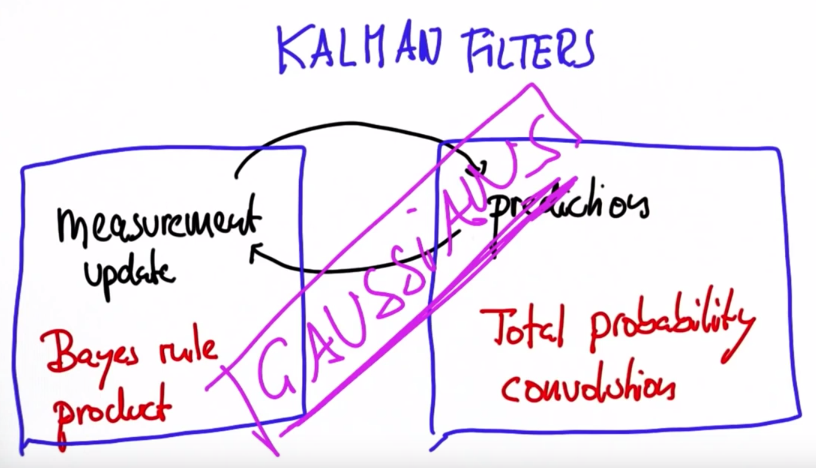
If we have a space where a robot could be, and the probability that the robot could be in any space is equal (Probability Distribution is Equal), it is called the State of Maximum Confusion.

To localize from this situation, a robot would look at landmarks. If a robot observes that it stands in front of a door changes the distribution of the possible locations. Locations near doors have an increased belief, meaning that locations without a door have a decreased belief.

As the robot moves and senses again, there is a shift in the belief distribution. This is called a Convolution. If the robot measures it is near a door a second time, while knowing how far it moved, Convolution of both belief distributions can be used to find the final localized position.

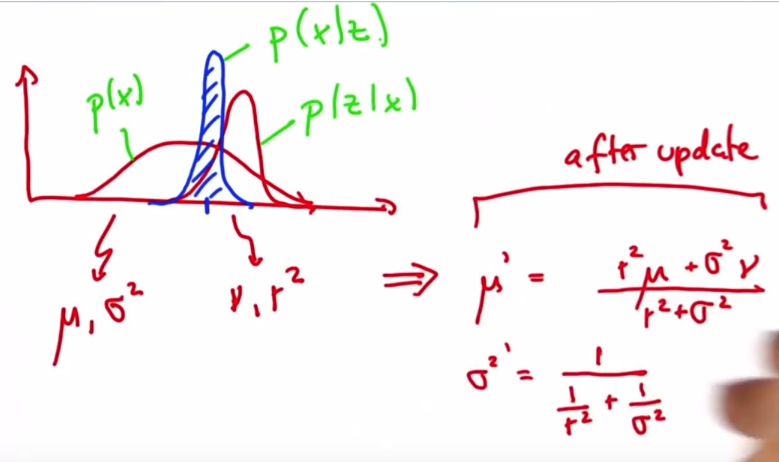


The final distribution is a good localization of where the robot is.

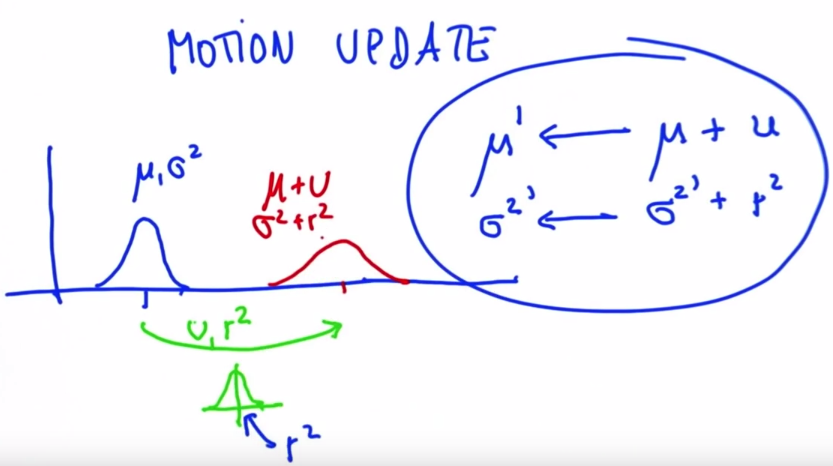


If we have a prior distribution of probability, and we have a measurement Gaussian, measurement where the mean is shifted, the new mean prediction (posterior) is somewhere in between the two, but closer to the distribution that has a smaller variance.

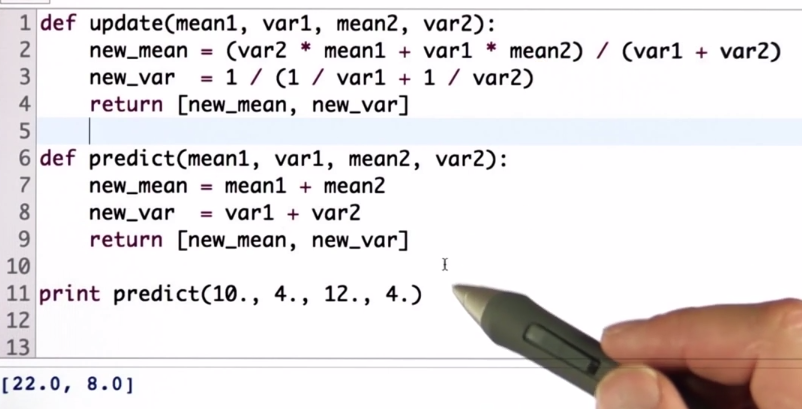
The peak of the predicted distribution is always higher than the measurement and the prior. This is because the more measurements we make, the more certain our new belief is. (Meaning the variance is smaller).



No matter the distribution, the more measurements, the smaller the variance of the prediction.



Example of a 1 dimensional Kalman Filter:



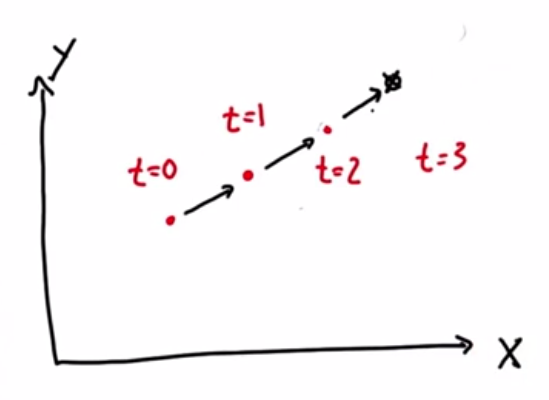
This example of a 1-D Kalman Filter is great but in reality, we need to have a filter that works at higher dimensions.

**2-D Example**

Example we have data from either a camera, or a radar that detects the location of a vehicle over time. The 2D Kalman Filter will let us estimate positions at future time steps. It lets estimate positions by implicitly figuring out the velocity of the object in order to have an accurate prediction. The sensor itself for a camera, only calculates velocities by taking differences between 2 measurements.

The fact that it can implicitly determine velocity as a part of a prediction is the most amazing part of using Kalman Filters for tracking applications.

This is why Kalman Filters are used in many applications from Control, to Artificial Intelligence.



**Higher Dimensional Gaussians**

These are called Multivariate Gaussians.

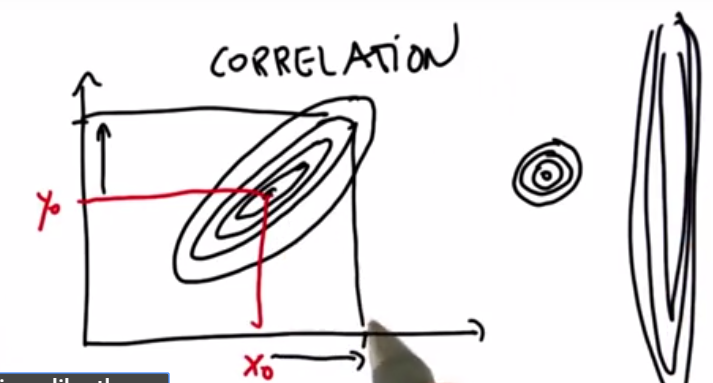
To represent multiple dimensions, values for the **mean is described as vectors with each element describing each dimension (Length of D).**

The **variance is described as Co-variance, which is represented as a Square Matrix with D rows and D columns.**

By thinking of a 2D Gaussian, we can draw the contour lines of Gaussian. The mean of the variance is the center (x0,y0), and the covariance defines the spread of the Gaussian described by the contour lines.

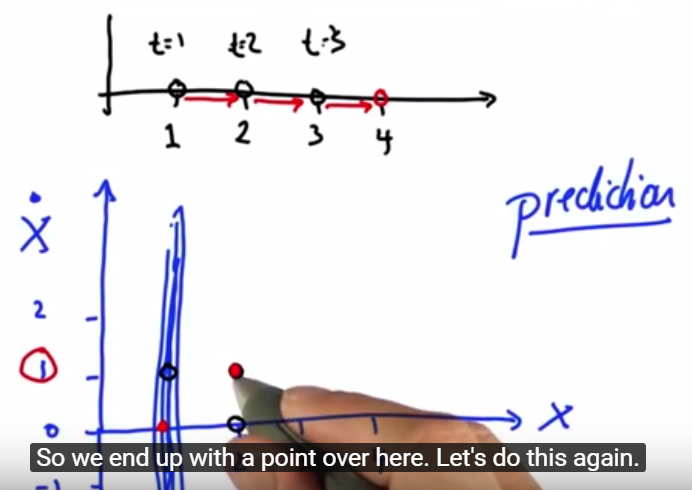
It is possible to have a fairly small uncertainty in 1 dimension and a high uncertainty in another.

When the Gaussian contours are angled, we can say that X and Y are correlated, so with just an x value, we can make an estimate of a y value.



For Example: Let’s say we are tracking an object that is uniformly moving in the example below. A Kalman Filter would predict that at t=4, the object is at location 4. What a Kalman Filter does is build a 2 dimensional estimate. 1 dimension of the estimate is for the location, and the other is for the velocity of the object.

If you know the initial position but not the initial velocity, the Gaussian that is predicted has a large variance in the velocity axis, and a small one in the position axis.



For the prediction step, a good prediction would be thee velocity staying the same as the previous time step, therefore making the position estimate from 1 to 2. Makes predictions of velocity making velocity observable.

**Variables of a Kalman Filter**

Variables of a Kalman Filter are often called States. This is because they represent states of the physical world like where an object is or how fast it is moving.

States are categorized as either **Observable** or **Hidden**.

An example of a Hidden variable is Velocity because the measurement of velocity is never directly observed.

With Kalman Filters, because both **Observable** and **Hidden** variables interact (are related), subsequent observations of the Observable variables give us information about the Hidden variables so that they can be estimated.

This means that from multiple observations of the position, we can estimate how fast a car is moving.

All Localization Filters actually do use information about Observable variables to gain information on Hidden variables, but because Kalman Filters are very efficient to calculate, problems that involve hidden variables that need to be estimated tend to use Kalman Filters.

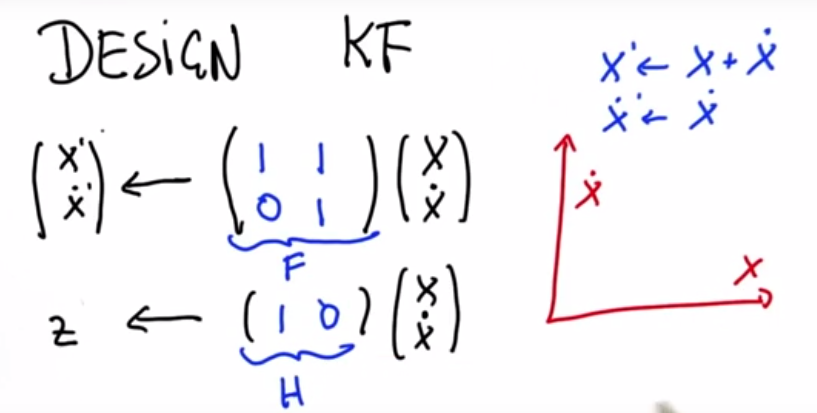
**Kalman Filter Design**

When designing a Kalman Filter, we need a State Transition Function (a Matrix) for the prediction step.

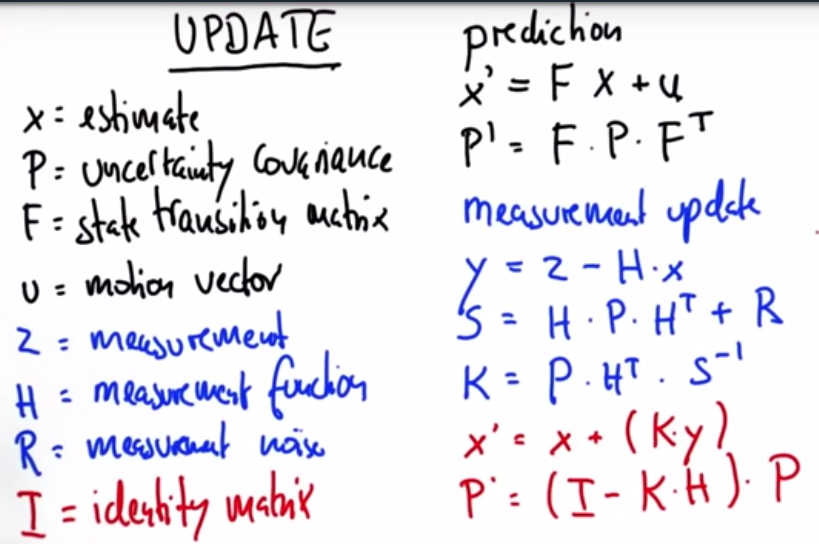
And a Measurement Function for the measurement step (Matrix).

Here is an Example for 1 Dimension where :





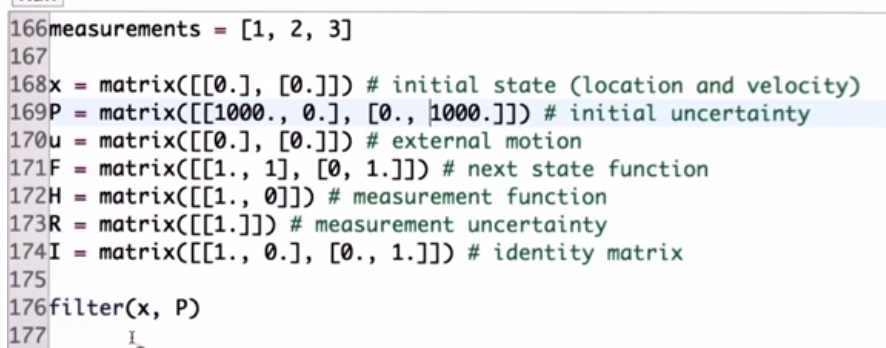
**Actual Formula:**

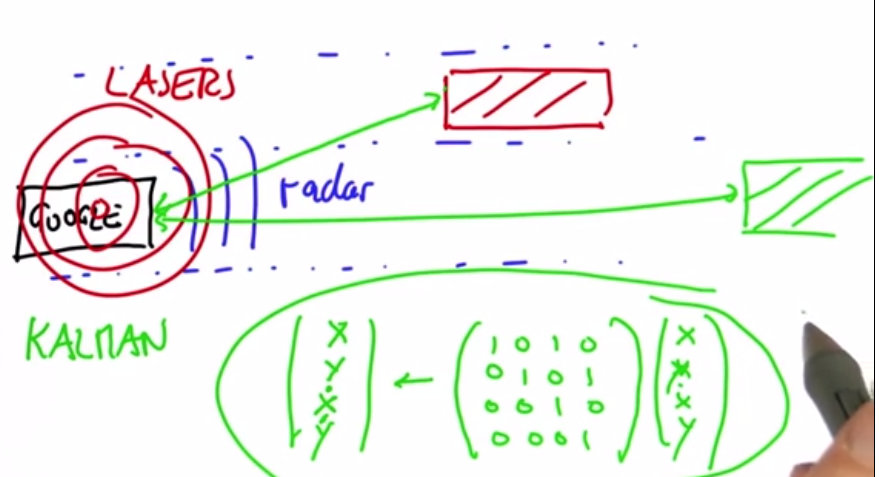


H maps states to measurements.

S is the projection of the System Uncertainty into the measurement Space

K is Kalman Gain





Particle Filters are another State Estimation Algorithm.

Particle Filters are easy to implement, and are very powerful. It is the most recent popular method of State Estimation.