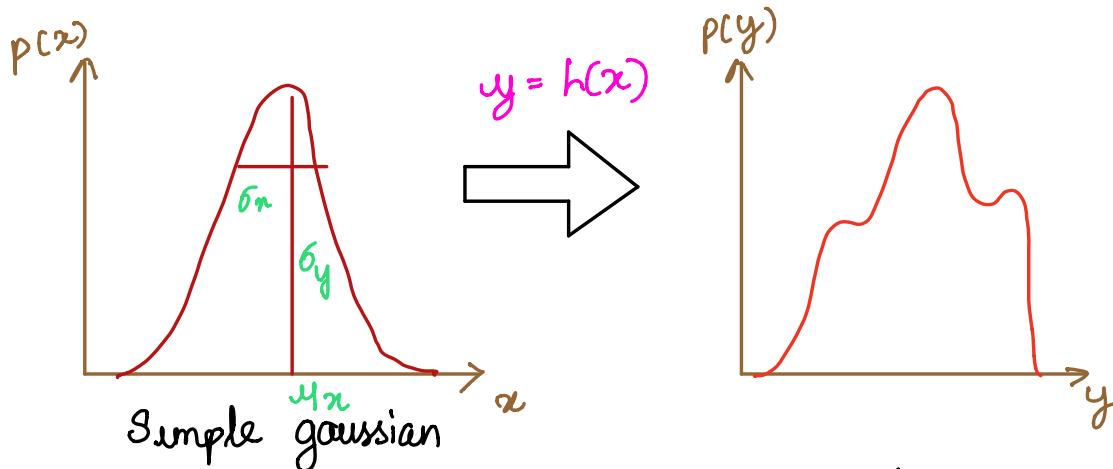


## Unscented Kalman Filter

→ Alternative approach for non linear functions, by using something called an unscented transform

"It is always easier to approximate a probability function rather than an arbitrary non-linear function"



Mean  $\rightarrow \mu_x$  if known  
Standard deviation  $\rightarrow \sigma_x, \sigma_y$

1D distribution  
after non-linear  
transformation.

We need to find the  
mean & S.D. of this  
distribution.

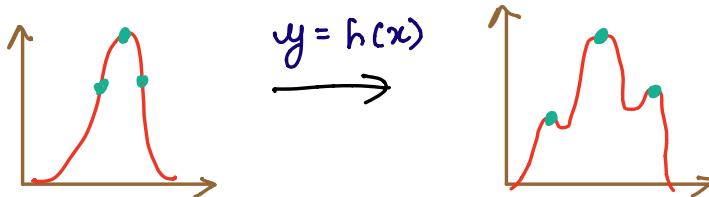
### Unscented Transform

- 1) we choose a set of sample sigma points from the input distribution, these samples are deterministic chosen certain standard deviations away from the mean

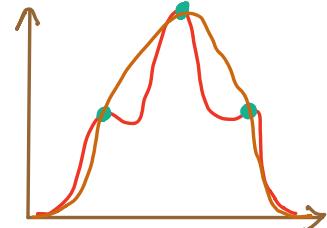
Unscented Transform  
is sometimes called  
sigma point transform



- 2) Pass each sigma point through the non-linear function to generate sigma points on the output distribution. (Transforming the sigma points)



- 3) Compute weighted mean & co-variance of transformed sigma points., this will give us a good approximation of the output distribution.

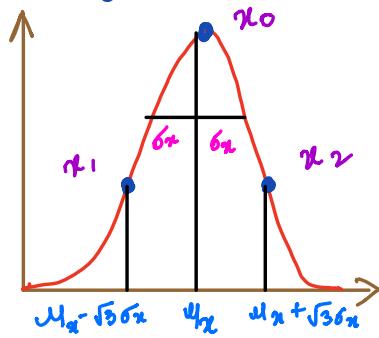


### 1) Choosing Sigma points

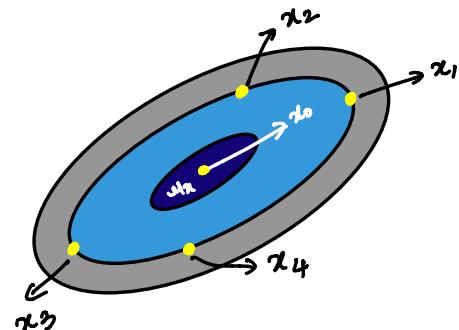
- a) How many sigma points do we need?
- b) Which of them are actually sigma points?

In general for a **N-dimensional** probability distribution we need  **$2N+1$**  sigma points, One for the mean & rest uniformly distributed around the mean.

For 1D gaussian.



For 2D gaussian



- 1) The first step to determine where the sigma points are is to compute the Cholesky decomposition of the co-variance matrix.

The Cholesky decomposition is basically a square root operation that acts on symmetric positive definite matrices such as the covariance matrices.

Note: If the input p.d.f is 1D then the Cholesky decomposition of the p.d.f is literally the square root of the variance which is the standard deviation of the p.d.f.

$$LL^T = \Sigma_{xx} \quad (L \text{ is lower triangular})$$

(use chol function in MATLAB  
cholesky function in Numpy)

- 2) Once the covariance matrix is decomposed we can choose the first sigma point to be the mean of the distribution and the other two sigma points will be mean  $\pm$  some factor multiplied by each col of the matrix 'L'

N - no. of dimensions  
of the p.d.f

$\kappa \rightarrow$  tuning param  
for gaussian p.d.f  
setting  $N + \kappa = 3$  is a  
good choice.

$$x_0 = \mu_x$$

$$x_i = \mu_x + \sqrt{N+\kappa} \text{ col } L$$

$$x_{i+N} = \mu_x + \sqrt{N+\kappa} \text{ col } L$$

$$i = 1, \dots, N$$

- 2) Transforming and recombining

Next we pass each of our  $2N+1$  sigma points through the non-linear function  $h(x)$

$$y_i = h(x_i) \quad i = 0, \dots, 2N$$

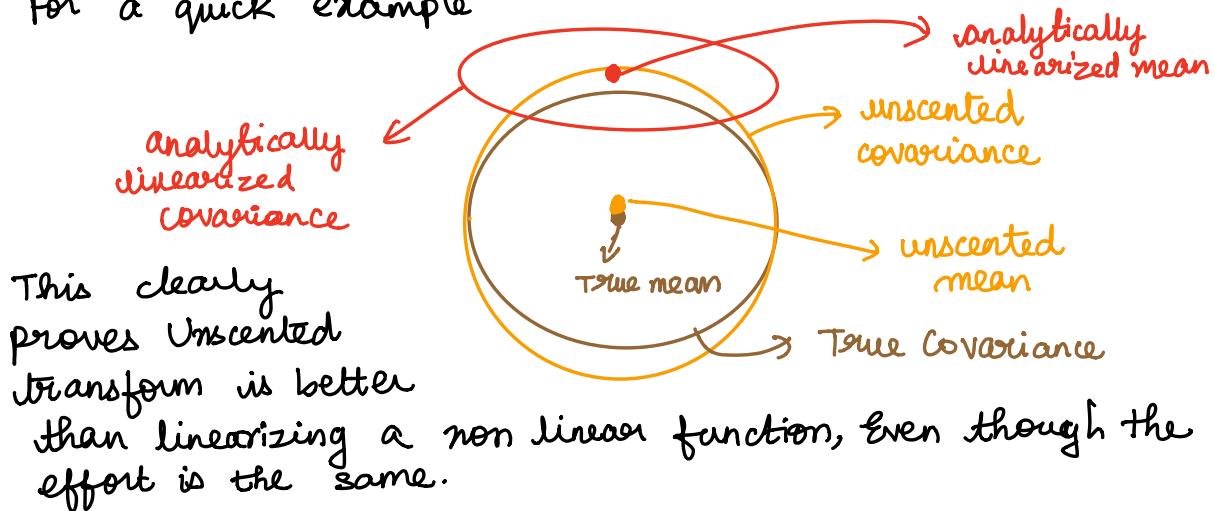
And finally compute the mean & co-variance of the output P.D.F.

$$\text{Mean : } \mu_y = \sum_{i=0}^{2N} d_i y_i$$

$$\text{Co-variance : } \Sigma_{yy} = \sum_{i=0}^{2N} d_i (y_i - \mu_y) \times (y_i - \mu_y)^T$$

$$\text{Weights : } d_i = \begin{cases} \frac{K}{N+K} & i=0 \\ \frac{1}{2} \frac{1}{N+K} & \text{otherwise} \end{cases}$$

For a quick example



We can now easily use the unscented transform in Kalman filtering framework with non-linear models.

### Non-linear Motion model

$$x_k = f_{k-1}(x_{k-1}, u_{k-1}, w_{k-1})$$

$$w_k \sim N(0, Q_k)$$

## Non-linear measurement model

$$y_k = h_k(x_k, v_k)$$

$$v_k \sim \mathcal{N}(0, R_k)$$

Instead of approximating the system equations by linearizing, we will calculate sigma points and use the Unscented Transform to approximate the PDF's directly.

### Prediction step

To propagate the state from time  $(k-1)$  to time  $k$ , apply the Unscented transform using the current best guesses for the mean & covariance.

1) Compute sigma points

$$\hat{x}_{k-1}^+ = \hat{x}_{k-1}$$

$$\hat{x}_{k-1}^{(0)} = \hat{x}_{k-1}$$

$$\hat{x}_{k-1}^{(i)} = \hat{x}_{k-1} + \sqrt{N+k} \text{ col } \hat{L}_{k-1} \quad i=1, \dots, N$$

$$\hat{x}_{k-1}^{(i+N)} = \hat{x}_{k-1} - \sqrt{N+k} \text{ col } \hat{L}_{k-1}$$

2) Propagate sigma points through non linear motion model to get predicted state at time  $k$

$$\hat{x}_k^{(i)} = f_{x-1}(\hat{x}_{k-1}^{(i)}, u_{k-1}, 0) \quad i=0, \dots, 2N$$

3) Compute predicted mean & covariance at time  $k$

It is important to add process noise

$$a^{(i)} = \begin{cases} \frac{1}{N+k}, & i=0 \\ \frac{1}{2(N+k)}, & \text{otherwise} \end{cases}$$

$$\hat{x}_k = \sum_0^{2N} a^{(i)} \hat{x}_k^{(i)}$$

$$\hat{x}_k = \sum_{i=0}^{2N} \alpha^{(i)} (\tilde{x}_k^{(i)} - \check{x}_k) (\tilde{x}_k^{(i)} - \check{x}_k)^T + \underbrace{Q_{k-1}}_{\substack{\text{process} \\ \text{noise}}} \quad (1)$$

(only applicable  
to additive noise)  
 $\underbrace{\quad}_{\substack{\text{process} \\ \text{noise}}} \quad \underbrace{Q_{k-1}}_{\text{(additive)}}$

## 2) Correction step

To correct the state estimate using measurements at time 'k', use the non linear measurement model and the sigma points from the prediction step to predict the measurements.

- Predict measurements from propagated sigma points

$$\hat{y}_k^{(i)} = h_k(\tilde{x}_k^{(i)}, 0) \quad i = 0, \dots, 2N$$

- Estimate the mean & the co-variance of the predicted measurements.

$$\hat{y}_k^{(i)} = \sum_{i=0}^{2N} \alpha^{(i)} \hat{y}_k^{(i)}$$

$$P_y = \sum_{i=0}^{2N} \alpha^{(i)} (\hat{y}_k^{(i)} - \hat{y}_k) (\hat{y}_k^{(i)} - \hat{y}_k)^T + R_k$$

(only applicable  
to additive  
noise)  
 $\underbrace{\quad}_{\substack{\text{additive} \\ \text{measurement} \\ \text{noise}}}$

- Compute the cross co-variance b/w the predicted states & predicted measurements for calculating the Kalman gain, This basically tells us how the measurements & states are co-related.

$$P_{xy} = \sum_{i=0}^{2N} \alpha^{(i)} (\tilde{x}_k^{(i)} - \check{x}_k) (\hat{y}_k^{(i)} - \hat{y}_k)^T$$

$$K_k = P_{xy} P_y^{-1} \rightarrow \text{Kalman gain}$$

4) Use the Kalman gain & correct the mean & covariance.

$$\hat{x}_k = \hat{x}_k + K_k (y_k - \hat{y}_k)$$

$$\hat{P}_k = P_k - K_k P_y K_k^T$$

### Conclusion

Since there is no linearization error, No computation of jacobian & higher accuracy & better speed

the UKF is always a better option compared to that of EKF. when the system is non linear.