# Oscillator Carrier phase compensation using Kalman tracking

Venkhat.V (123079008), Kumar Karunakaran (123079005)

For Course project in EE764 Wireless and Mobile communication

In practice, Oscillator won't produce carrier with accurate frequency fc, but within a small interval around fc. Hence, there is always a small frequency mismatch between carrier frequency from receiver's local oscillator and transmitter carrier frequency. This causes the received symbol to get shifted by some phase (not magnitude). This phase have to be compensated before decoding.

The system model and algorithm that we have implemented are based on [1].

The phase in each time 'n' is. Where $\Delta\emptyset$ is due to frequency mismatch $\Delta f$. (2*pi*$\Delta f$). Phase noise $\Delta\emptyset$ is modeled as Gaussian iid noise in each time n.

- mean=0
- variance= 4*pi*$\Delta f_{3dB}$ /fs
- fs- symbol frequency
- $\Delta f_{3dB}$ is the 3db half power frequency of local oscillator at receiver.

$$\phi(n) = \phi(n-1) + \Delta\phi.$$

The received symbol is (we are considering symbol level model. Without phase compensation in front -end of Rx, we get this symbol)

r(n)= $(s(n) \star h(n))e^{j\phi(n)} + \xi(n)$

After decoding data, we cancel out the data, channel part by dividing. Channel is known at Rx. This is the equation of our interest.

$$y(n) = r(n)/\hat{a}(n) \approx \underbrace{\exp\left(j\phi(n)\right)}_{D(\cdot)} + \frac{\xi(n)}{\hat{a}(n)}.$$

We try to estimate the phase, from this non linear equation. Kalman filter is used for estimation in linear functions only. So here Unscented Kalman is used, which transforms the non-linear into a some form of linear by weighted sum of different extrapolated samples around the input.

## Kalman Filter

**Parameter input:**

1. Phase noise variance  pn_var
2. Noise variance(for each iteration new noise variance is calculated by dividing by power of a  )
3. Observation function(y= exp(j*x(n))+noise)
4. Input state equation (x(n)=x(n-1)+ ΔØ)
5. Initial state variance(=pn_var )
6. Intial state estimate= E(ΔØ)
7. Unscented Kalman filter parameter 'L' , then 2L+1 samples
8. Unscented kalman realted values. Alpha_sq, beta. Calculate lambda, wc, wm

**Input**: y(n)

**State parameters updated everytime**: State variance, state estimate

**Output**: Estimated 'Phi (n)' value of present symbol

## Kalman Algorithm:

1. Find Predicted state from previous state estimate (xcap)
2. Find predicted variance from previous state variance (=prev state variance +phase noise variance). (state equation relation is used for above 2 steps)

Unscented Kalman peocedure.

- Find 2L+1 samples around predicted state
- Find Weighted sum of these samples
- Pass it through observation function relation
- Find variance of y
- Find cross covariance between phi and y.

3. Find kalman kain K.(from var_yy, var_xy, and observation noise variance)
4. Find the State estimate(weigted sum of predicted state and observed )
5. Find the state variance

(Note: Here, in our system we need predicted phi of next symbol)

## Simulation Parameters:

1. Alpha_sq=10^-3
2. Beta=2;

3. L=1 (unscented samples =3)
4. Delf/fs = 0.001
5. Noise variance =1;
6. Rayleigh Flat fading i.i.d channel with h~CN(0,1)
7. QPSK symbol modulation
8. SNR in db= 10 to 20

## Description:

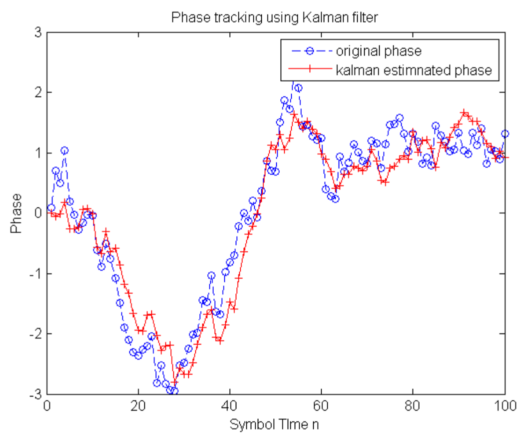
Calculate weights wc, wm and lambda. Compensate the phase using the predicted value. Then decode the data. Then Use the decoded data and uncompensated rx symbol to estimate the phase.

## 2 types of simulation done in Matlab:

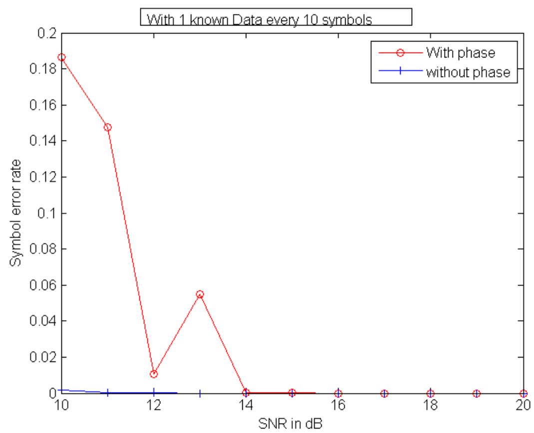
- Symbol error rate comparison between with phase and without phase
- Same as above , Along with known data every 10 or 20 samples

## Matlab Simulation Results:

1. Phase Tracking using the kalman filter. When Phase variance = 4*pi*0.01



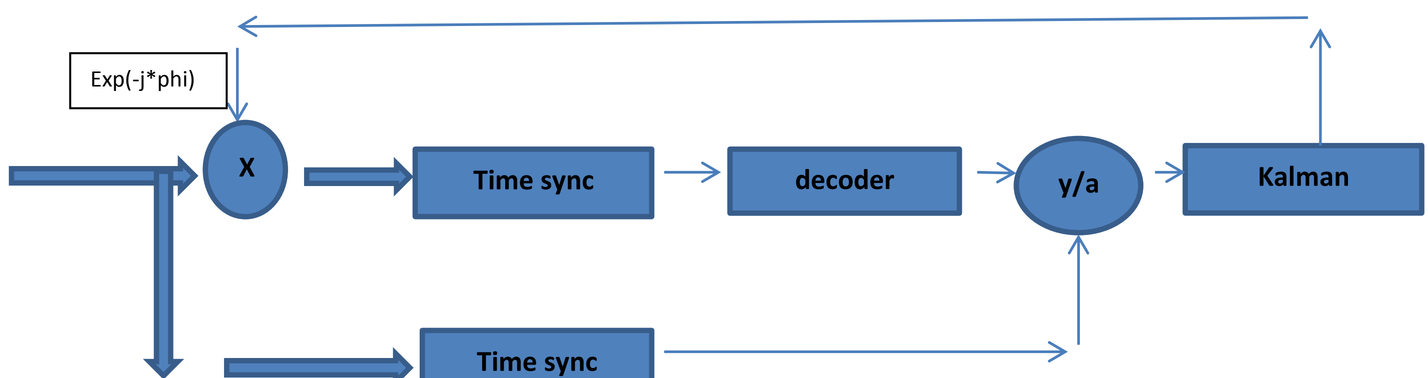

2. Symbol error rate comparison with phase error and without phase error. Also 1data is known every 10 symbols. SNR varies from 10dB to 20dB. Channel: Rayleigh flat fading i.i.d.




## Inference:

1. When iterations (iter) is less than 1000, except snr10, everything showed zero error.
2. When one error happens, it leads to lot of future symbols error, as the decoded data affects the phase estimate.
3. So once every 10 or 20 or 50 samples, if 1 data symbol is made known, then Pe remains with 0.3 goes to 0.0x values and then zero. Performance is better.
4. When delf=0.01, kalman performance was very bad. It couldn't track the phase.

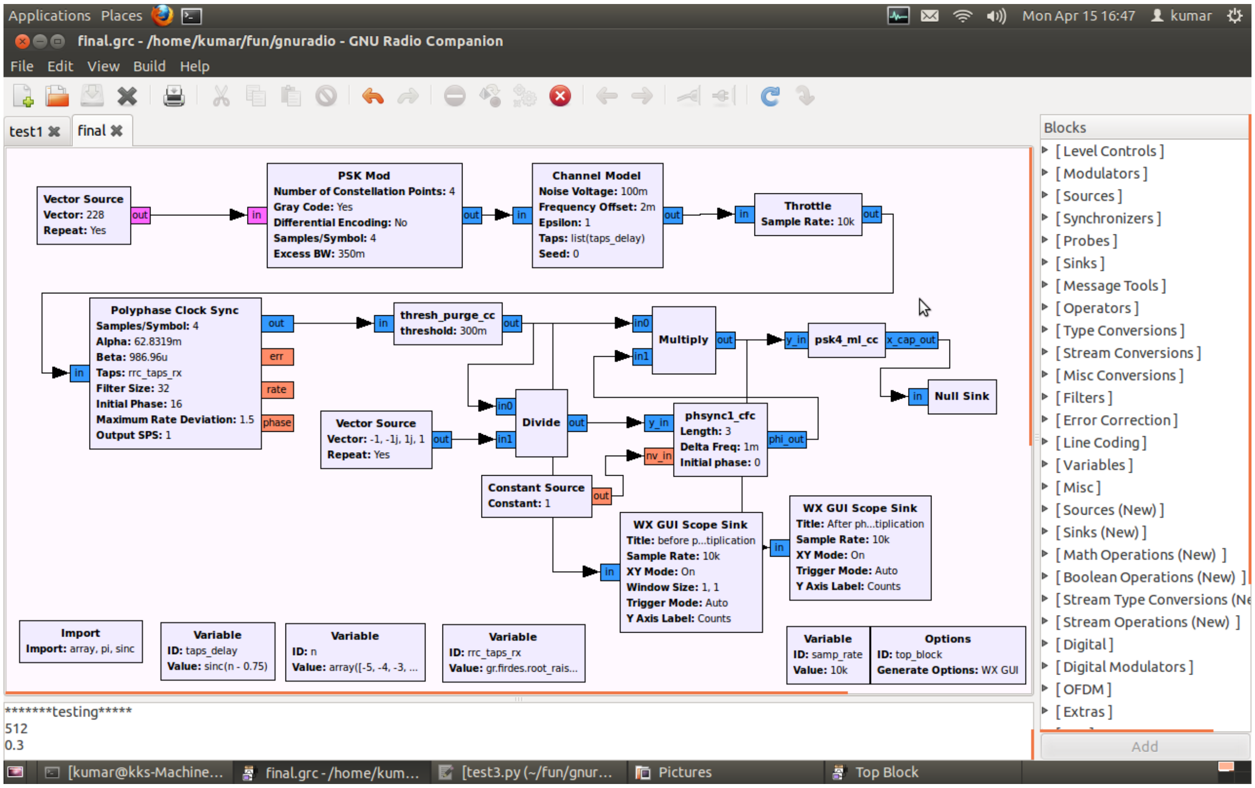
## Block diagram of the receiver:



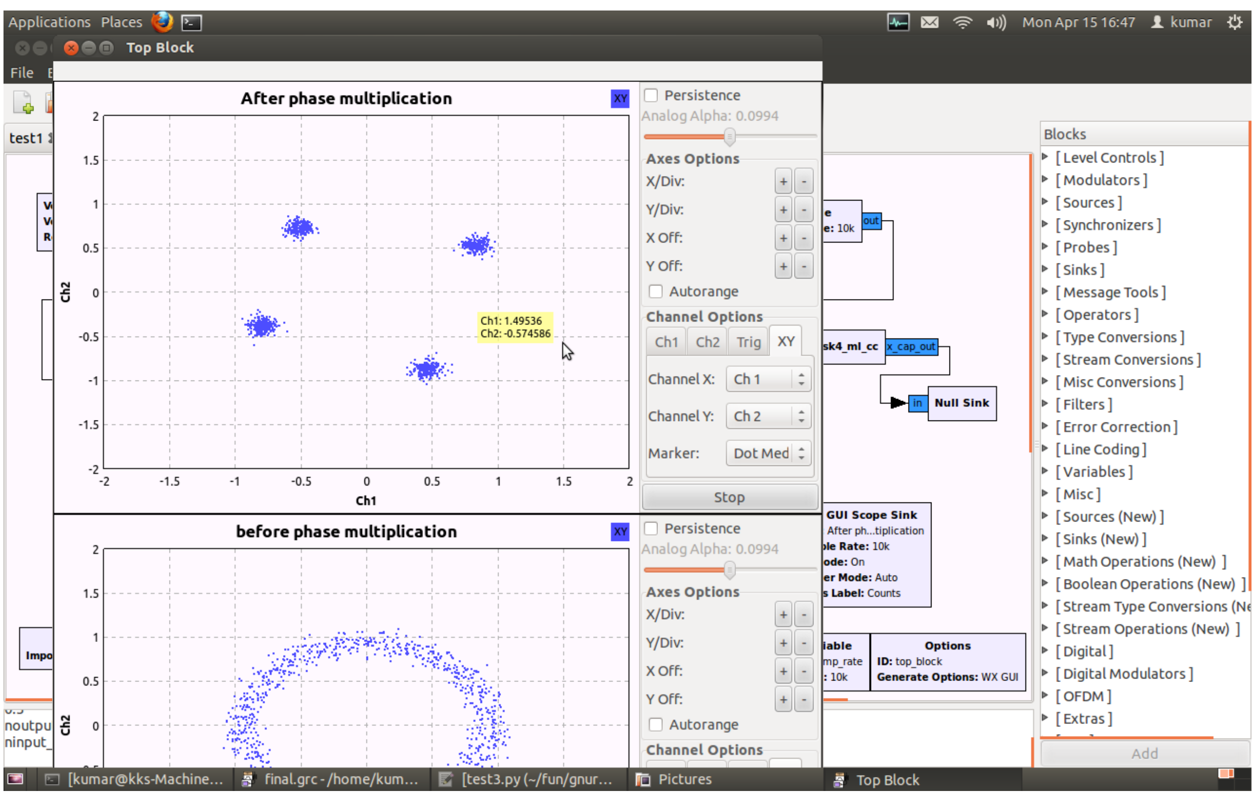

## GNU radio Implementation.

We coded in C++ and then converted to block.

## GNU Block diagram:




## Result:



## References


1. Bittner, S.; Frotzscher, A.; Fettweis, G.; Deng, E., "Oscillator Phase Noise compensation using Kalman tracking," *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on* , vol., no., pp.2529,2532, 19-24 April 2009

2. Wan, E.A.; Van der Merwe, R., "The unscented Kalman filter for nonlinear estimation," *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000* , vol., no., pp.153,158, 2000

3. http://gnuradio.org/redmine/projects/gnuradio/wiki/OutOfTreeModules