# QAM Carrier Tracking for Software Defined Radio

SDR Forum Technical Conference 2008

*James Schreuder*

*SCHREUDER ENGINEERING*

*www.schreuder.com.au*

SCHREUDER ENGINEERING
SOFTWARE DESIGN AND ENGINEERING

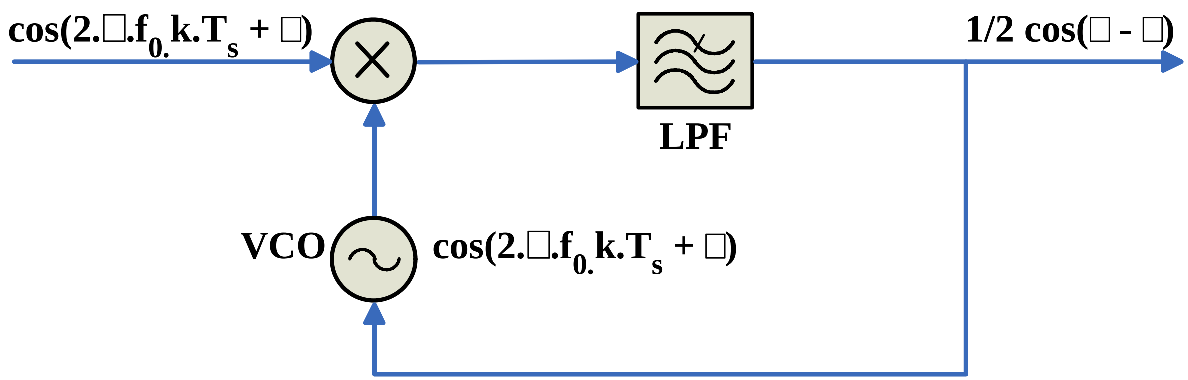
# Outline

1. Introduction
2. Analog versus Digital Phase Locked Loops (PLLs)
3. Adaptive Parameter Estimation
4. QAM Phase Recovery using PLLs
5. QAM Decision Directed Carrier Tracking
6. Summary

# Introduction

- Investigation into software based PLL techniques for tracking Quadrature Amplitude Modulation (QAM) carriers.
- Originally motivated by analysis of draft TIA Public Safety Radio protocol proposal *Scalable Adaptive Modulation* (TIA-902-BAAB)
- The protocol made use of a TDM/FDM structure incorporating 4/16/64 QAM channels
- The protocol used insertion of Pilot and Synchronisation symbols to aid receiver synchronisation (Pilot Symbol Assisted Modulation – PSAM).

# Analog versus Digital PLLs




**Analog PLL:**

• PLL adjusts the VCO phase, $\phi$, in order to match the input signal phase, $\theta$.

• When the PLL output is maximized, the PLL is locked and $\phi = \theta$.

**Digital PLL:**

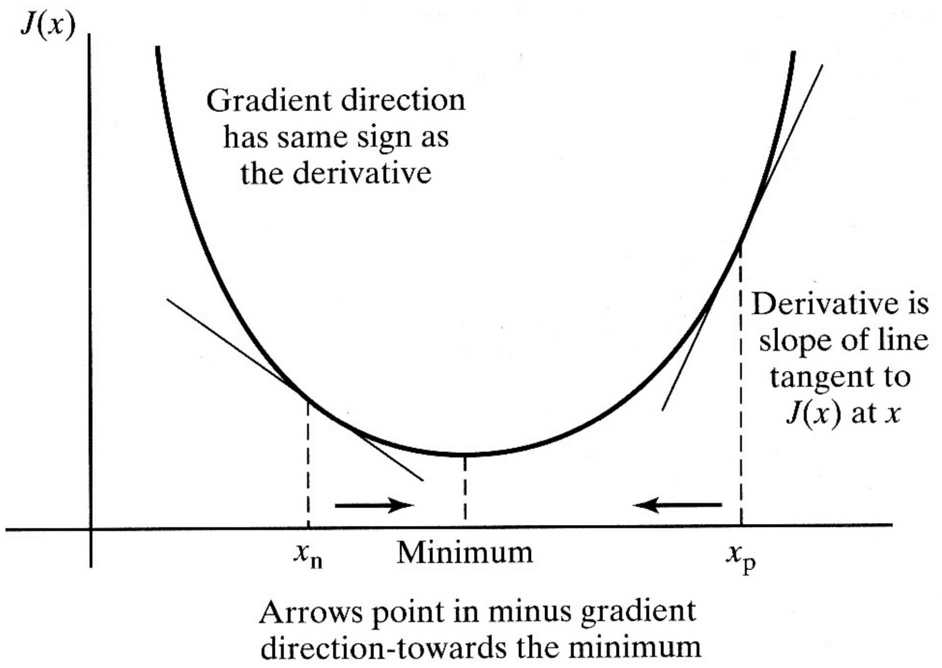
• VCO and loop filter are replaced by digital versions in a software loop

• On each loop iteration, VCO phase, $\phi$ [k], must be adjusted such that the PLL output steps closer and closer to a maximum value ( when $\phi$[k] = $\theta$[k] ).

• Optimization problem => we can implement using Adaptive Parameter Estimation.

# Adaptive Parameter Estimation

Estimation example:




Adaptive update equation:

$$x[k+1] = x[k] - \mu \frac{dJ(x)}{dx}$$

# Digital PLL using Adaptive Parameter Estimation

Performance function:

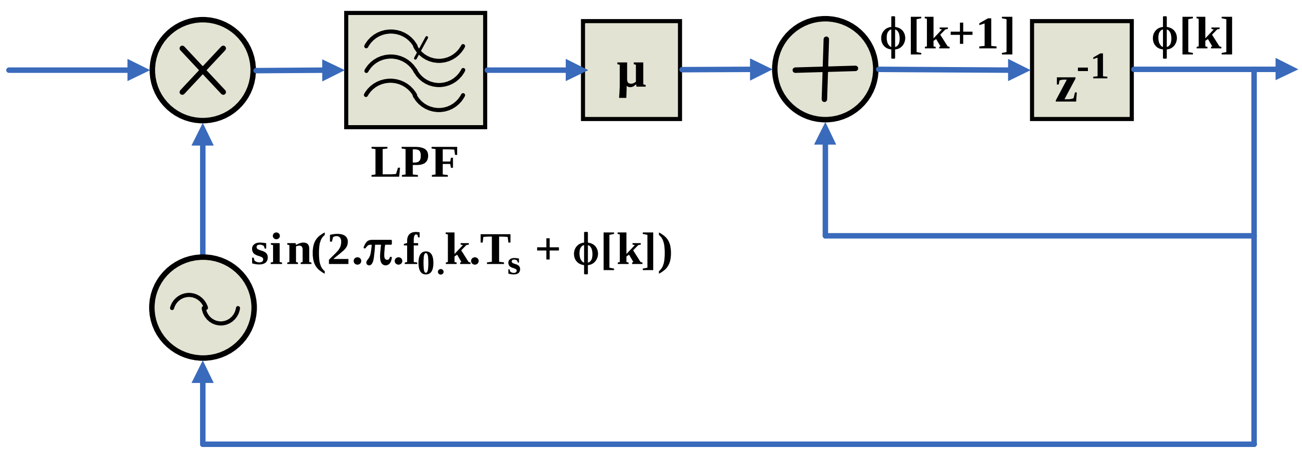
$$J_{PLL}(\phi) = LPF\left\{r[kT_s].\cos(2\pi.f_0 kT_s + \phi[k])\right\}$$

Approximate as:

$$\frac{dJ_{PLL}(\phi)}{d\phi} = LPF\left\{-r[kT_s]\sin(2\pi f_0 kT_s + \phi[k])\right\}$$

Final update equation:

$$\phi[k+1] = \phi[k] - \mu LPF\left\{r[kT_s]\sin(2\pi f_0 kT_s + \phi[k])\right\}$$



# MATLAB code of PLL

```
Ts=1/2000; time=1; t=0:Ts:time; % time vector
f0=200; phoff=pi/2;             % carrier freq. and phase
fc=200;                         % assumed freq. at receiver
rp=cos(2*pi*fc*t+phoff);        % simplified received signal
carrier = rp;

fl=100; ff=[0 .01 .02 1]; fa=[1 1 0 0];
h=firpm(fl,ff,fa);              % LPF design
mu=.01;                         % algorithm stepsize

theta=zeros(1,length(t));
theta(1)=0;                      % initialize vector for estimates
z=zeros(1,fl+1);                 % initialize buffer for LPF
for k=1:length(t)-1              % z contains past fl+1 inputs
  VCO(k) = sin(2*pi*f0*t(k)+theta(k));
  z=[z(2:fl+1), rp(k)*VCO(k)];
  update=fliplr(h)*z';           % new output of LPF
  theta(k+1)=theta(k)-mu*update; % algorithm update
end
```

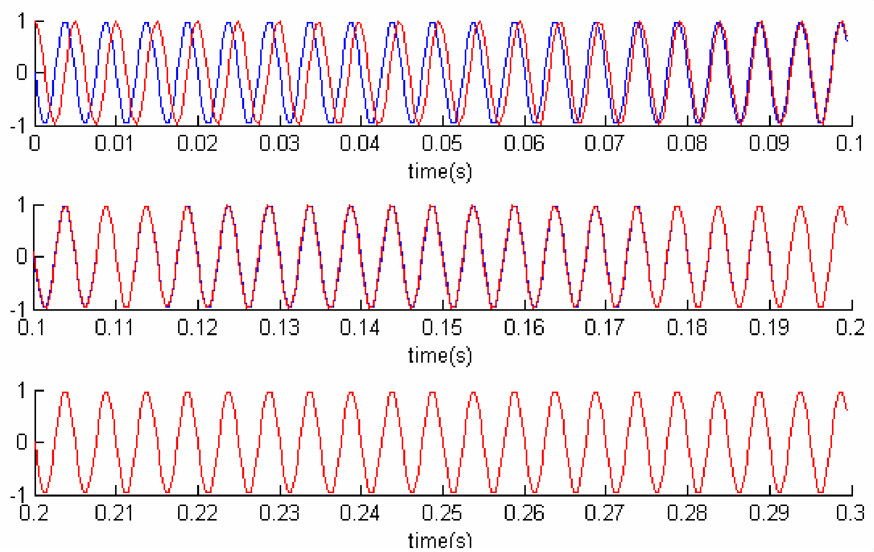
SCHREUDER ENGINEERING
SOFTWARE DESIGN AND ENGINEERING

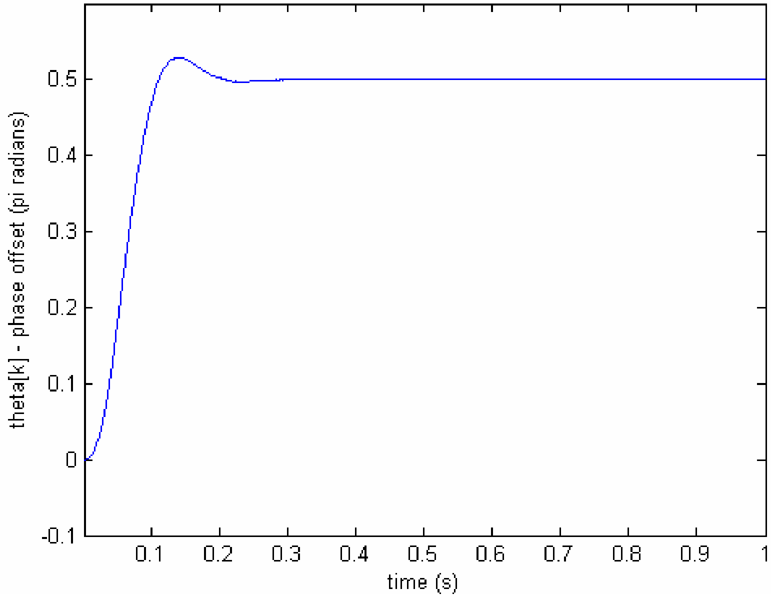


SDR forum™

# Digital PLL Phase Lock

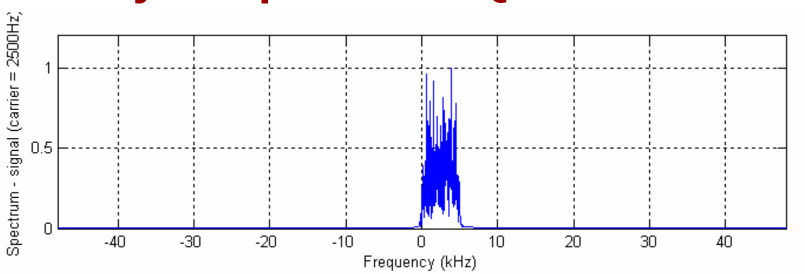
| | |
|---|---|
| • Input signal shown in blue: $\cos(2.\pi.200.t + \pi/2)$<br>• VCO phase shown in red | • VCO phase adaptation, $\mu$ = 0.01<br>• Phase Lock achieved after 0.3 seconds with VCO phase equal to $\pi/2$. |



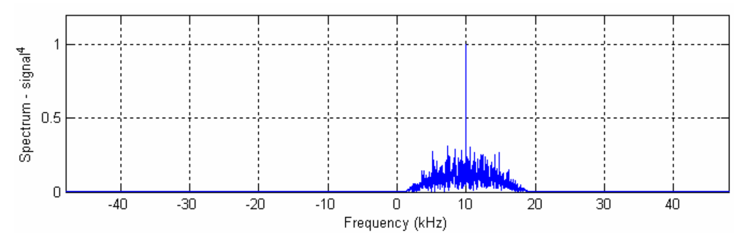



# QAM Phase Recovery using PLLs

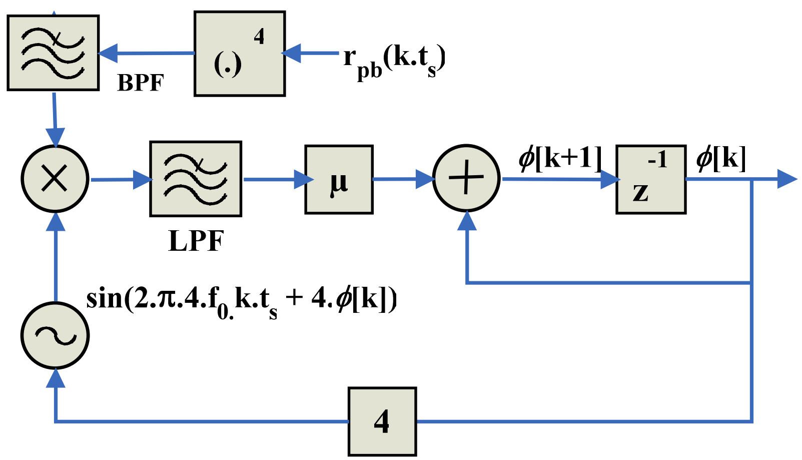
## Can you spot the 4-QAM carrier?




## 4-QAM signal raised to the 4th power

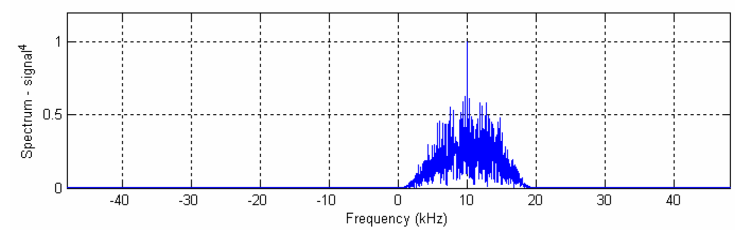



## Possible PLL design:

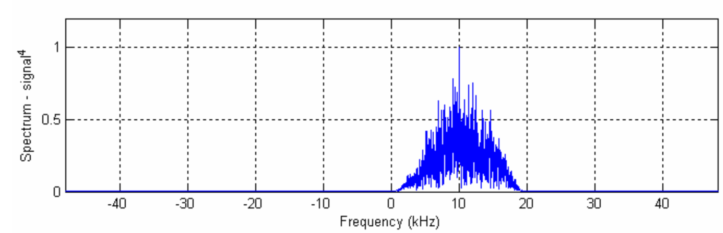



Where: $\phi[k+1] = \phi[k] + \mu \sin(4(\theta[k] - \phi[k]))$

## 16-QAM signal raised to the 4th power




## 64-QAM signal raised to the 4th power




SCHREUDER ENGINEERING
SOFTWARE DESIGN AND ENGINEERING



SDR forum

# QAM Decision Directed Carrier Tracking

- The algorithm generates a phase error signal by exploiting the phase and amplitude difference between each received symbol value and the nearest ideal QAM constellation symbol value.
- The receiver's carrier phase is adaptively adjusted to match the transmitter's phase by minimizing the mean-square of an error function.
- Minimization process performed using Adaptive Parameter Estimation.

DDCT error function is defined as:

$$e[kT_s] = c_{iq} - r_{bb}[kT_s]$$

The performance function is then:

$$J_{MSE} = E\{|e[kT_s]|^2\}$$

Giving an update equation as:

$$\phi[k+1] = \phi[k] + \frac{\mu \partial J_{MSE}}{\partial \phi}$$

Update equation is then derived as:

$$\phi[k+1] = \phi[k] + \mu \frac{\Im m\left[\overline{e[kT_s]}\, r_{bb}[kT_s]\right]}{|c_{iq}||r_{bb}|}$$

SCHREUDER ENGINEERING
SOFTWARE DESIGN AND ENGINEERING

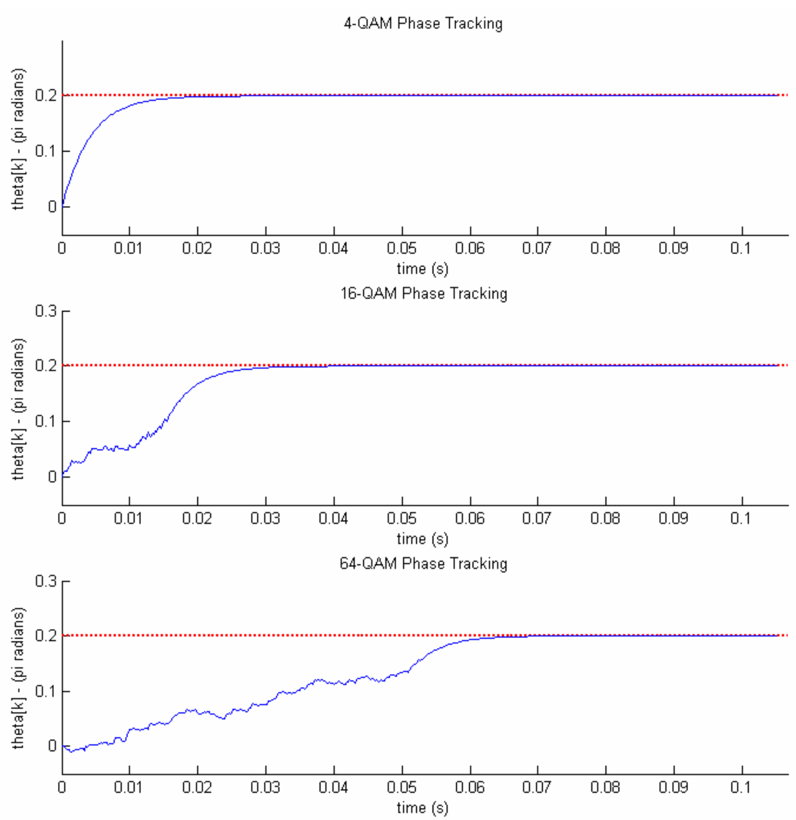
# MATLAB code of DDCT PLL

```
CARRIER = 1000;
k = 1; mu = 0.1; M = 16; Ts = 1 / 4800;
phaseNow = 0; phaseEst = phaseNow; phaseInc = 2*pi*CARRIER * Ts;

for s = pbSymbols(1:end) % An array of passband QAM symbols
  % Demodulate the passband symbol and store in array
  bbSymbols[k] = s .* exp(-j * phaseNow);
  % Find the nearest QAM constellation point to symbol s
  decisionSymbol = qamMatch(s, M);

  % Calculate the phase error
  decisionError = decisionSymbol - s;
  % Calculate the new phase estimate
  theta[k] = phaseEst;
  phaseEst = phaseEst + mu * (imag(conj(decisionError)*s)
                               / (abs(decisionSymbol)*abs(s)));
  % Calculate the next demodulation phase value
  phaseNow = phaseNow + phaseInc + phaseEst;
  k = k + 1;
end
```

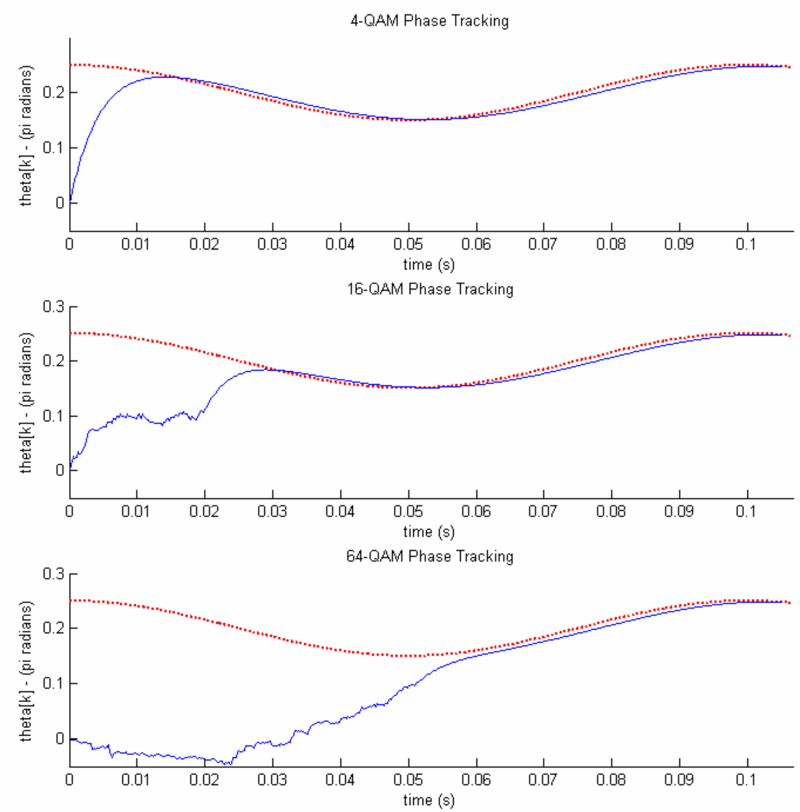
SCHREUDER ENGINEERING
SOFTWARE DESIGN AND ENGINEERING

# QAM DDCT Phase Lock

- Input 4/16/64 QAM signal with carrier = 1000Hz and transmitter phase offset = $0.2\pi$
- Expected phase offset of $0.2\pi$ radians plotted in red and $\phi[k]$ series in blue




- Same input QAM signals with varying transmitter phase offset



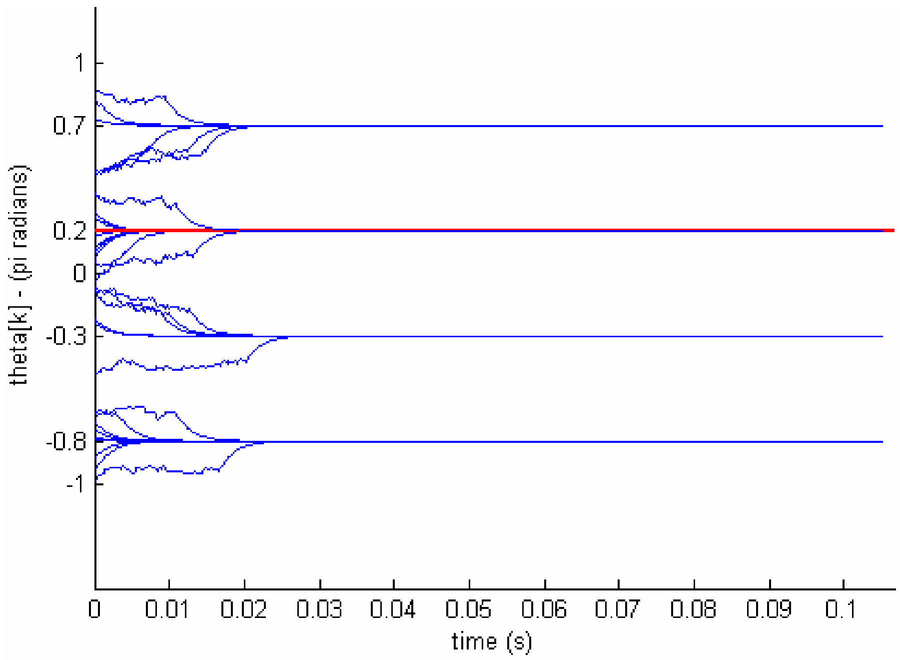

SCHREUDER ENGINEERING
SOFTWARE DESIGN AND ENGINEERING



SDR forum

# DDCT π/2 Phase Ambiguity

- The preceding DDCT PLL code was run 32 times on the a set of random 16-QAM symbols with a transmitter phase offset of 0.2π radians.
- On each execution the initial phase estimate is set to a random phase value in the interval [-π, π] radians.




- Expected phase offset of 0.2π radians plotted in red
- Plots of each executions $\phi[k]$ series in blue.
- The π/2 phase ambiguity inherent in DDCT is shown as each $\phi[k]$ plot converges to a value in the series:

$$0.2\pi + n.\pi/2,$$

$$\text{where } n = 0, \pm 1, \pm 2, \ldots$$

SCHREUDER ENGINEERING
SOFTWARE DESIGN AND ENGINEERING

# $\pi$ / 2 Phase Ambiguity Correction

There are several ways to resolve the phase ambiguity:

1. Differentially encode the message source so that the change in symbol value between each symbol is known
2. Let a trained equalizer automatically add a rotational phase to achieve a match to training symbols
3. Correlate the down-sampler output with a known/training signal
4. By insertion of known data symbols into the symbol stream

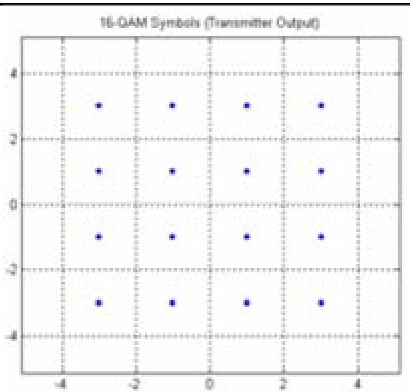
Since SAM was specified to use inserted Pilot/Sync symbols, these symbols were successfully incorporated into the DDCT algorithm to "kick" the receiver phase around to the correct $\pi/2$ phase orientation.

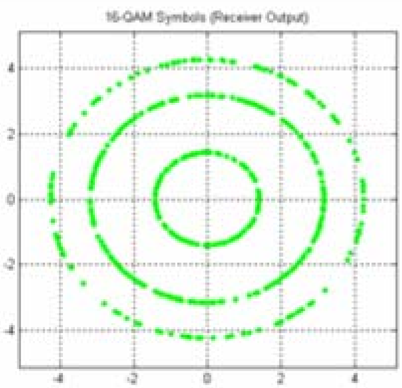
# DDCT Carrier Frequency Offsets

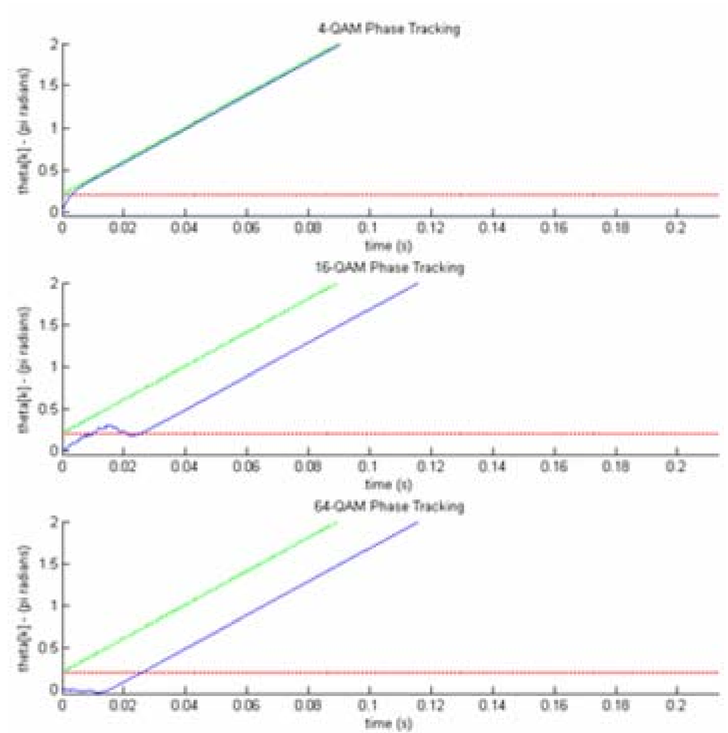
- Transmitter 16-QAM constellation in blue.
- Receiver constellation in green following demodulation with a 0.2π radians constant phase offset and 10Hz frequency offset. Constellation is spinning!

- Expected phase offset of 0.2π radians plotted in red and ϕ[k] series in blue.
- The green plot shows the expected ϕ[k] series given by the equation:

$$\theta[k] = 2\pi(f_t - f_r)kT_s + (\theta - \phi)$$



# DDCT Carrier Frequency Offsets

- Recall that the phase error signal to be tracked on each DDCT loop iteration is:

$$\Delta\phi = \frac{\Im m\left[\overline{e[kT_s]} \cdot r_{bb}[kT_s]\right]}{|c_{iq}||r_{bb}|}$$

- In order to track the additional phase change due to a carrier frequency offset, the following additional phase accumulation step (a second-order loop) is included in the PLL:

$$\psi[k+1] = \psi[k] + \mu_2 \Delta\phi[k]$$

- The final second-order adaptive update equation for DDCT is then:

$$\varphi[k+1] = \varphi[k] + 2\pi f_r T_s + \mu_1 \Delta\phi[k] + \psi[k]$$

SCHREUDER ENGINEERING
SOFTWARE DESIGN AND ENGINEERING

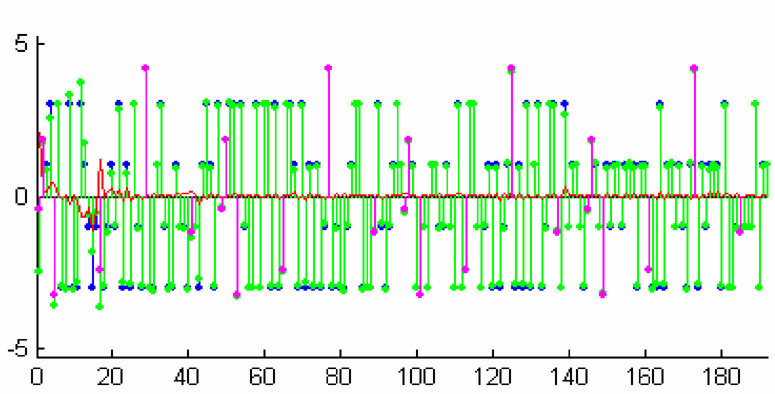


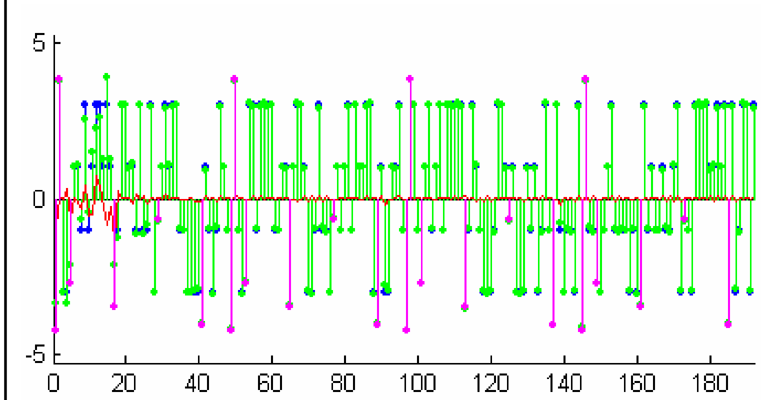
SDR forum™

# MATLAB code of 2nd Order DDCT

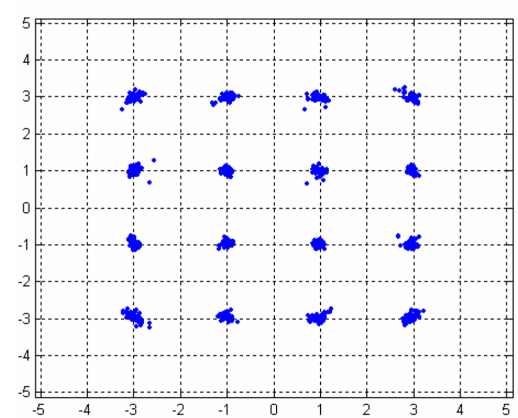
```
CARRIER = 1000;
k = 1; mu = 0.1; M = 16; Ts = 1 / 4800;
phaseNow = 0; psi = 0; phi = 0; phaseInc = 2 * pi * CARRIER * Ts;

for s = pbSymbols(1:end) % An array of passband QAM symbols

  % Demodulate the passband symbol and store in array
  bbSymbols[k] = s .* exp(-j * phaseNow);

  % Find the nearest QAM constellation point to symbol s
  decisionSymbol = qamMatch(s, M);

  % Calculate the phase error
  decisionError = decisionSymbol - s;

  % Calculate the new phase estimate
  theta[k] = phi;
  phaseError = (imag(conj(decisionError)*s))
                            / (abs(decisionSymbol)*abs(s));
  psi = psi + mu2 * phaseError;
  phi = mu1 * phaseError + psi;

  % Calculate the next demodulation phase value
  phaseNow = phaseNow + phaseInc + phi;
  k = k + 1;
end
```

# Putting it all together…



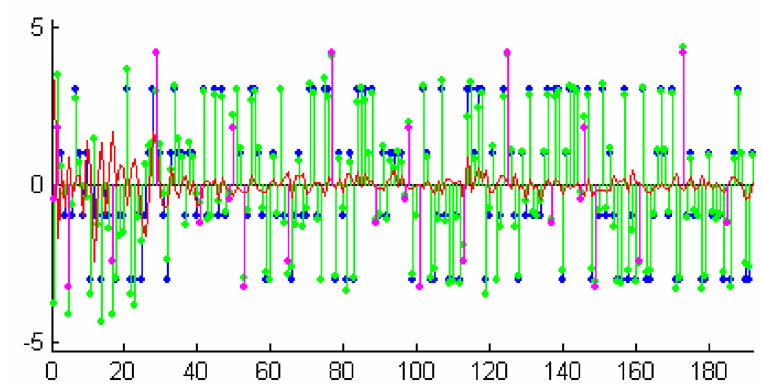

- 18900 Hz sub-carrier
- 70 Hz carrier offset
- 4800 symbols/second
- TX data symbols = blue
- TX pilot/sync symbols = pink
- Received symbols = green
- Error signal = red

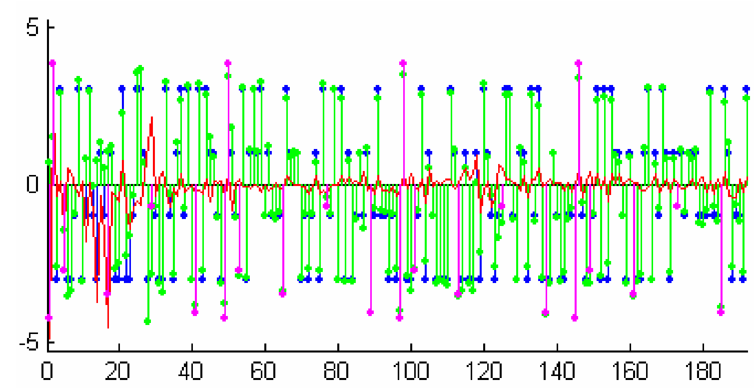



16-QAM RX Constellation

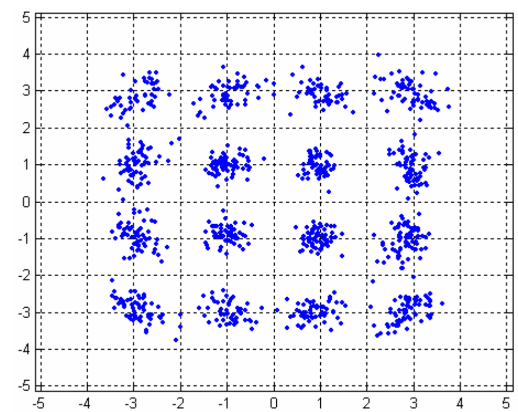
SCHREUDER ENGINEERING
SOFTWARE DESIGN AND ENGINEERING

# Putting it all together…with noise




- 18900 Hz sub-carrier
- 210 Hz carrier offset
- 4800 symbols/second
- SNR = 20dB




16-QAM RX Constellation

# Summary

- Adaptive Parameter Estimation can be used as a basis for many types of Digital PLLs
- DDCT provides a reliable method for tracking M-QAM carrier phase and frequency offsets
- DDCT operates on passband M-QAM symbol values rather than individual sample values
- DDCT has an inherent $\pi/2$ phase ambiguity that must be considered in the M-QAM tracking algorithm design
- DDCT is resilient to the introduction of channel noise

# References

[1] TIA "Wideband Air Interface Scalable Adaptive Modulation (SAM) Physical Layer Specification", TIA-902-BAAB, 2003.

[2] J.M. TORRANCE, L. HANZO, "Comparative Study of Pilot Symbol Assisted Modem Schemes". Sixth International Conference on Radio Receivers and Associated Systems, pp 36 – 41, 26 – 27 September 1995.

[3] R. E. BEST, Phase-Locked Loops: Design, Simulation and Applications, McGraw-Hill. 2003

[4] R. JOHNSON, W. A. SETHARES, W. A., Telecommunication Breakdown: Concepts of Communication Transmitted by Software-Defined Radio, Pearson Prentice Hall, 2004.

[5] R. JOHNSON, A Digital Quadrature Amplitude Modulation (QAM) Radio, Pearson Prentice Hall, 2003

[6] J.B.ANDERSON, Digital Transmission Engineering, Prentice Hall, 1999.

[7] J. A. C. BINGHAM, The Theory and Practice of Modem Design, Wiley Press, 1988.

[8] S. A. TRETTER, Communication system design using DSP algorithms: with laboratory experiments for the TMS320C6701 and TMS320C6711, Kluwer Academic/Plenum Publishers, New York, 2002.

[9] L. E. FRANKS, Carrier and Bit Synchronization in Data Communication - A Tutorial Review. IEEE Transactions on Communications, COM-28, 1980

# Questions?

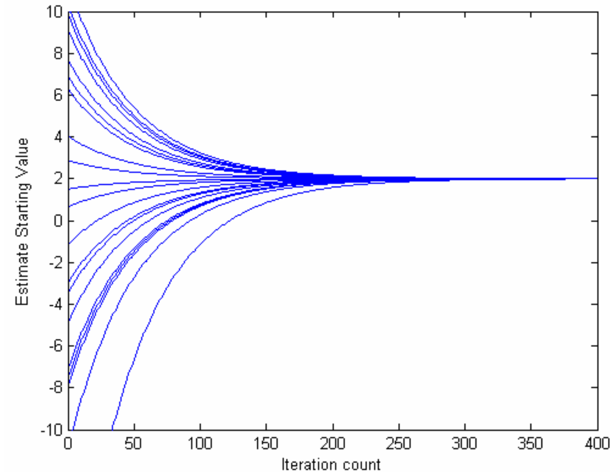
# Backup Slides

# Adaptive Parameter Estimation Example

Estimate the minimum value of: $J(x) = x^2 - 4x + 4$

Recall: $$x[k+1] = x[k] - \mu \frac{dJ(x)}{dx}$$ where: $$\frac{dJ(x)}{dx} = 2x[k] - 4$$

Therefore: $$x[k+1] = x[k] - \mu(2x[k] - 4)$$
$$= (1-2\mu)x[k] + 4\mu$$

| MATLAB code: | Estimation results: |
|---|---|
| (see code below) |  |

```
% Find the minimum of J(x)=x^2-4x+4 via steepest descent
N=50;                           % number of iterations
mu=.01;                         % algorithm stepsize
x=zeros(size(1,N));             % initialize x to zero
x(1)=3;                         % starting point x(1)
for k=1:N-1
  x(k+1)=(1-2*mu)*x(k)+4*mu;    % update equation
end
```