Frequency + Phase Synchronization

- still: carrier only

Update rule: (phase synch)

$$\Theta[n+1] = \Theta[n] + \mu \cdot \frac{df}{d\theta} \bigg|_{\theta = \Theta[n]}$$

with:

$$\frac{df}{d\theta} \bigg|_{\theta = \Theta[n]} = LPR\{ s_p(t) \cdot (-\sin(2\pi f_d t + \Theta[n])) \}$$

For $$s_p(t) = \cos(2\pi f_d t + \phi)$$

$$\frac{df}{d\theta} \bigg|_{\theta = \Theta[n]} = \frac{1}{2} \cdot \sin(\phi - \Theta[n])$$

$$= \frac{1}{2} \cdot (\phi - \Theta[n])$$ if $$\phi - \Theta[n]$$ is small

$$\Rightarrow$$ Update relationship:

$$\Theta[n+1] = \Theta[n] + \mu \cdot \frac{1}{2} (\phi - \Theta[n])$$

$$= \Theta[n] \cdot (1 - \frac{\mu}{2}) + \phi \cdot \frac{\mu}{2}$$

$$\Rightarrow$$ converges to

$$\lim_{n \to \infty} \Theta[n] = \phi (\pm n \cdot 2\pi)$$
However, when there is a frequency offset:

\[
\cos(2\pi f_c t + \phi) = \cos(2\pi f t + \phi(t))
\]

\[
\text{incoming} \quad \phi(t)
\]

\Rightarrow \text{Frequency offset causes phase to grow over time.}

\[
\Rightarrow \phi[n] = \phi_0 + n \cdot \Delta \phi \quad \Delta \phi = (f_c - f) \cdot 2\pi \cdot T
\]

\[
\text{discrete time}
\]

Using this \( \phi[n] \) in update rule:

\[
\Theta[n+1] = \Theta[n] + \frac{\mu}{2} \cdot (\sin(\phi[n] - \Theta[n])
\]

\[
\approx \Theta[n] + \frac{\mu}{2} \cdot (\phi[n] - \Theta[n])
\]

In this case, \( \Theta[n] \) does not converge to \( \phi[n] \).

Can show:

\[
\lim_{{n \to \infty}} \Theta[n] - \phi[n] = -\frac{\Delta \phi}{1 - \frac{\mu}{2}}
\]

This residual phase "hurts" output signal:

\[
m(t) \cdot \cos(\Theta[n] - \phi[n])
\]

want this to be 1? 

-2-
Remedy: To close the "gap" between $\Theta_{\text{CuJ}}$ and $\phi_{\text{CuJ}}$:

- modify the update rule
  
  "penalize" persistent errors $\phi_{\text{CuJ}} - \Theta_{\text{CuJ}}$

- achieved by adding a term proportional to the sum (integral over all adjustments $\frac{dI}{d\Theta_{\text{CuJ}}}$

New update rule: $S[n] = \sum_{k=0}^{n} \frac{dI}{d\Theta_{\text{CuJ}}}

also: $S[n] = S[n-1] + \frac{dI}{d\Theta_{\text{CuJ}}}

$\Theta[n+1] = \Theta[n] + \mu_0 \left( \frac{dI}{d\Theta_{\text{CuJ}}} + \alpha S[n] \right)$

Small positive constant $\alpha$

MATLAB: Synch.NCOffreg.m
Allowing message signal:

When a message signal is present,

\[ \frac{df}{d\theta} = m(t) \cdot \sin(\phi - \theta) \]

when \( m(t) < 0 \), updates move \( \theta \) away from \( \phi \).

\[ \Rightarrow \text{ does not work} \]

Work around: square received signal

\[ s_p^2(t) = m(t) \cdot \cos(2\pi f t + \phi) \]

\[ \Rightarrow s_p^2(t) = m^2(t) \cdot \cos^2(2\pi f t + \phi) \]

\[ = \frac{1}{2} m^2(t) + \frac{1}{2} m^2(t) \cdot \cos(4\pi f t + 2\phi) \]

suitable for sync

\[ \Rightarrow \text{ control loop} \]

\[ s_p(t) \]

\[ \Theta[n+1] = \Theta[n] + m \cdot \frac{df}{d\theta} \]

or

\[ \Theta[n] + m \cdot \frac{df}{d\theta} + \text{bias} \]

\[ \text{SynchFreqMsg\&CO}_n \]

\[ \times \]

\[ \sin(4\pi f t + 2\phi) \]

\[ \text{LPF} \]

\[ \Rightarrow m^2(t) \cdot \sin(2\phi - 2\theta) \] can be used.