Analog And Telecommunication Electronics – Mini-Project

Politecnico Di Torino

06/06/2016

JAVA APPLICATION OF A PHASE LOCK LOOP

Authors:
Jurgen Kello
Luigi Zaffarana

ABSTRACT:

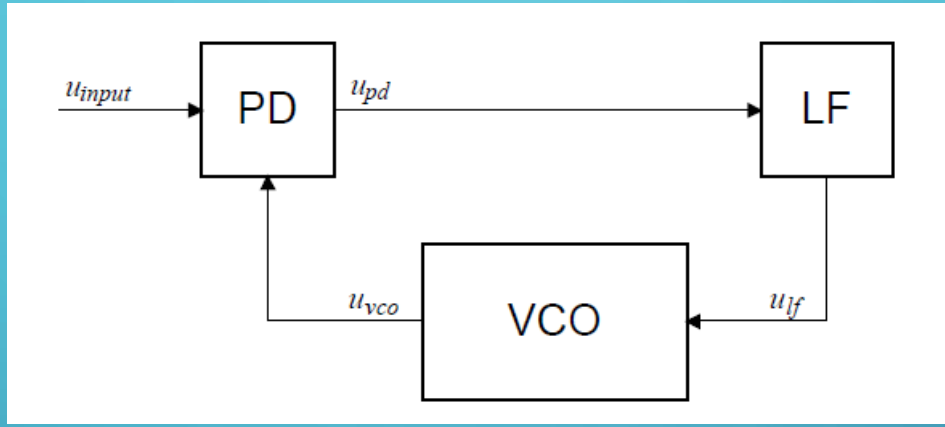
PHASE-LOCK LOOPS ARE USUALLY IMPLEMENTED TO EVALUATE THE FREQUENCY OF A SINUSOID IN THE PRESENCE OF NOISE AND TO TRACK THE FREQUENCY OF THIS SIGNAL AS IT CHANGES. THIS DOCUMENT DESCRIBES THE DEVELOPMENT OF A JAVA SOFTWARE TO EMULATE THE BEHAVIOUR OF A PHASE-LOCK LOOP.

OUTLINE:

- **Overview of the Phase-Locked Loop**
- **Software realization of each block**
- **Graphical Interface**
- DEMO

OVERVIEW OF THE PHASE-LOCKED LOOP

The three basic components of a standard PLL are:



As a phase detector a simple multiplier can be used, which multiplies the input with the output of the voltage controlled oscillator and the result will be the sum of a DC with a time varying component.

The sum beat on the other hand will be cut off by the loop filter.
The mean value of the filter output will determine the operating frequency of the VCO

So due to this feedback loop it is possible to have the output of the VCO with the same frequency as the input and with approximately 90 degree of phase shift.

PHASE DETECTOR

A simple analog phase detector will be just a multiplier :

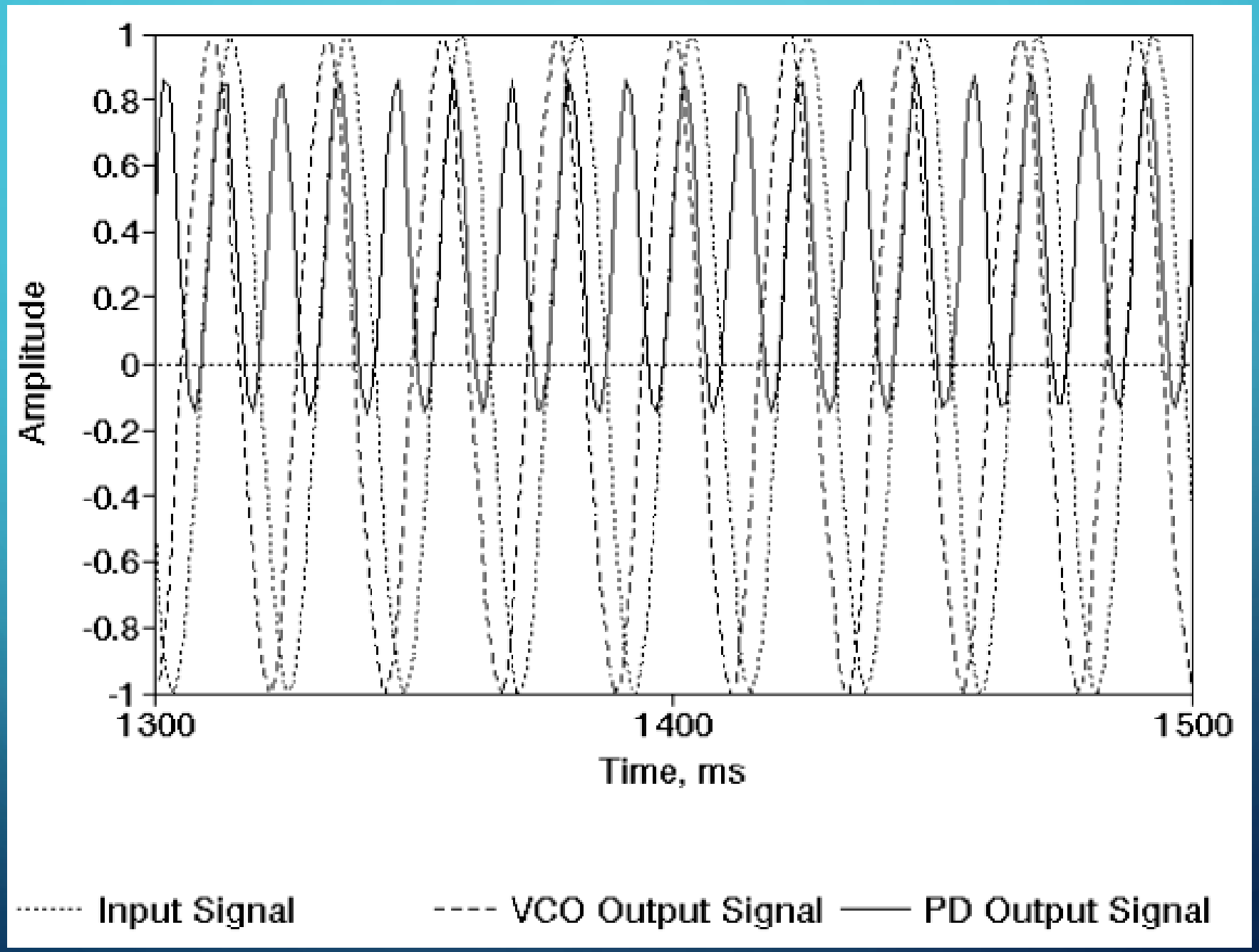
$$u_{input}(t) = A\sin(\omega t + \theta_1)$$

$$u_{vco}(t) = B\sin(\omega t + \theta_2)$$

The output of the phase detector will be:

$$u_{pd}(t) = K_d \frac{A\,B}{2}\left[\cos(\theta_1 - \theta_2) + \cos(2\omega t + \theta_1 + \theta_2)\right] \quad K_d \rightarrow Phase\ detector\ Gain$$

So the DC component mentioned before depends on the sine of the phase difference between input and VCO output.

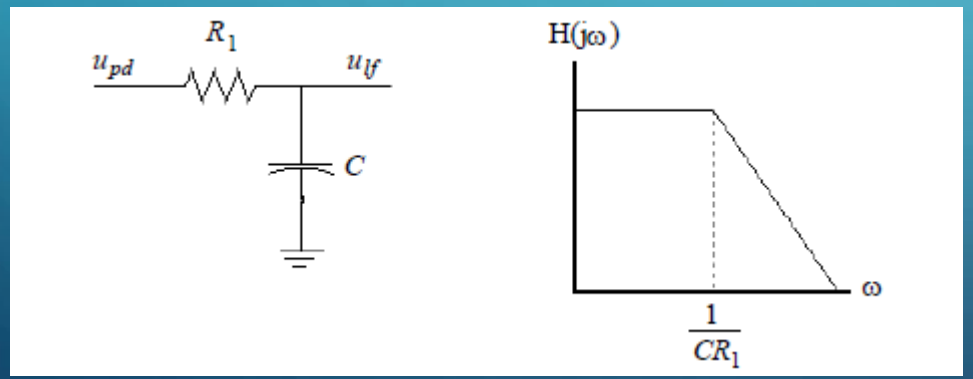

1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
Amplitude
1300
1400
1500
Time, ms
Input Signal
VCO Output Signal
PD Output Signal

LOOP FILTER

The AC components of the phase detector output are attenuated by this filter.

$$BW = \frac{1}{\tau}$$

Having a small bandwidth, meaning cutting off most of the AC components, would result in having a large time constant, which means to have a slow response, then a slow response may reduce the ability of the PLL to remain locked.



The transfer function for the filter will be given by:

$$H(j\omega) = \frac{1}{j\omega C R_1 + 1}$$

VCO

The output signal frequency of the VCO is linearly dependent on an input signal

For this application, the VCO signal at the output is given by the following expression:

$$u_{vco}(t) = \sin(\,(K_0\, lpf_{out}(t) + \,\omega_{or}\,)\,t\,) \quad where:$$

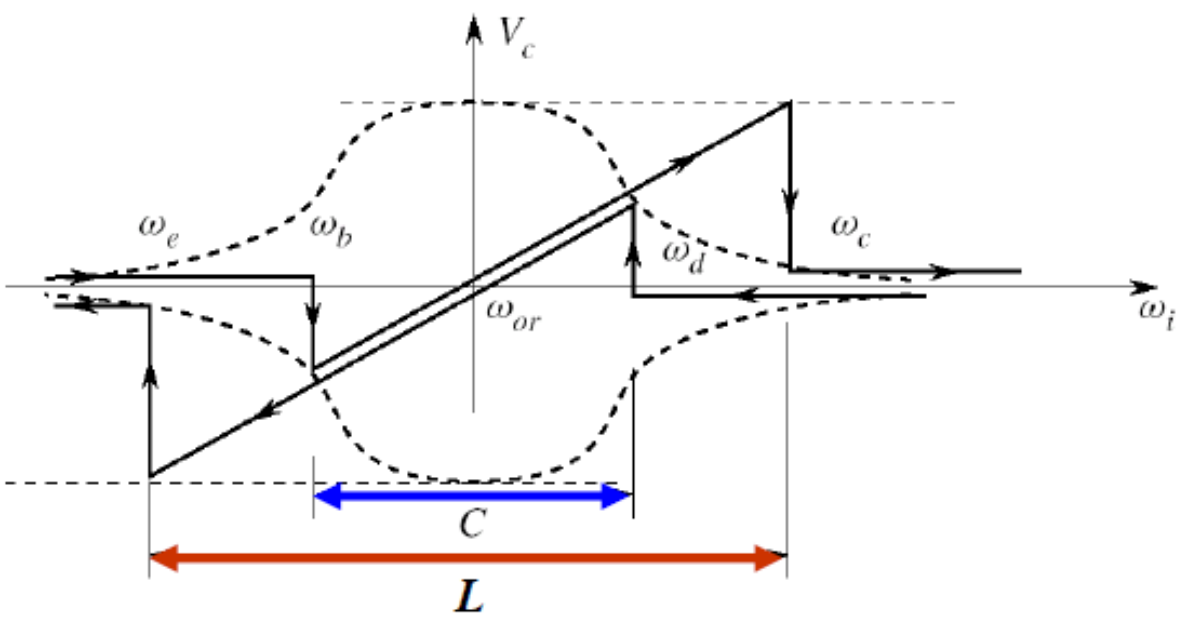
$$-lpf_{out}(t)\ is\ the\ loop\ Filter\ Output$$

$$-K_0\ is\ the\ gain\ of\ the\ VCO$$

$$-\omega_{or}\ is\ the\ Centre\ Frequency\ of\ the\ VCO$$

BUTTERFLY CHARACTERISTIC

As we know from theory, to determine the behaviour of the PLL, is possible to use the Butterfly Characteristic :



In our Software realization we will just use the first quadrant

Assuming 2 signals with amplitude of $1V$ the V_c maximum value at the output will be of $0.5V$.

SOFTWARE REALIZATION OF EACH BLOCK

THE PHASE DETECTOR

As described before, an analog phase detector for a PLL will consist of a multiplication of the input signal with the VCO output signal.

```
phase_detector = k_phd * u_out * u_in;
```

THE LOOP FILTER

To realize a software version of the Filter, transform in the z-Domain should be applied to arrive at the Digital version of it.

For this implementation of the software, we preferred to not use a system described in the z-domain since that would imply to use vectors, which would increase the computational cost of the system

```
s_delu_dt = s_delu_dt + (phase_detector - lpf_out) * (dt_r);

lpf_out = (float) ((float) s_delu_dt / (cr));
```

(see reference [3] for more details)

VOLTAGE CONTROLLED OSCILLATOR

A discrete-time software implementation of the VCO was developed by incrementing a phase angle $\theta(t)$ using a circle with unitary radius (because the input signal will always have a unitary amplitude).

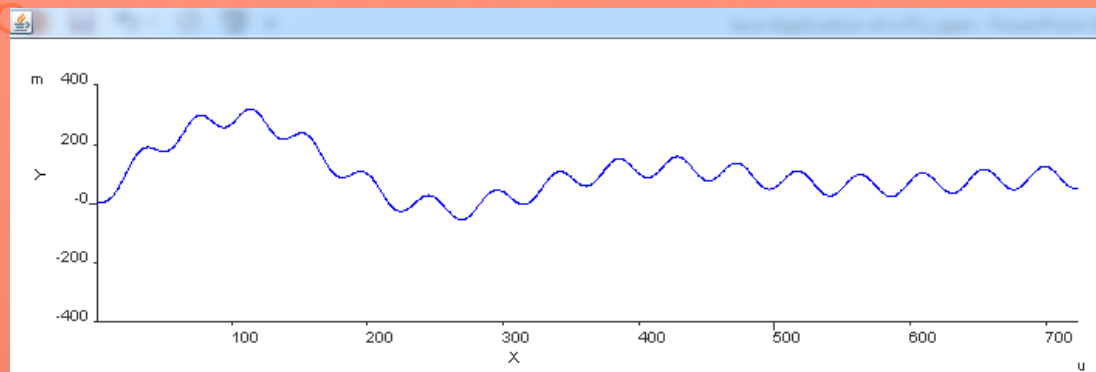
$$\theta(t) = \theta(t-1) + (\omega_0 + u_{lf(t)})\, T_S$$

$$u_{vco}(t) = \sin\big(\theta(t)\big)$$

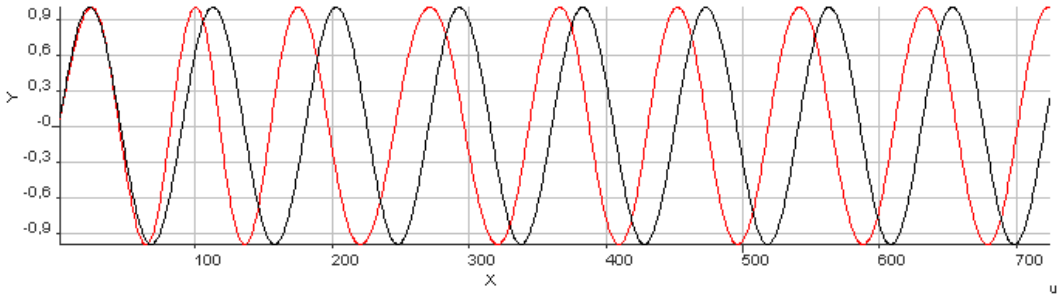
The software realization of these relations will be straightforward:

```
phase_u_out = phase_u_out + ((float)(2*dt_r*Math.PI) *(f0 +(k_vco*lpf_out)));

u_out = (float) ((float)A * Math.sin( phase_u_out ));
```



Low Pass filter Output [x: Seconds, y: Value]



Output Signal [x: Seconds, y: Value] Input Signal [x: Seconds, y: Value]

Input Signal Amplitude = 1

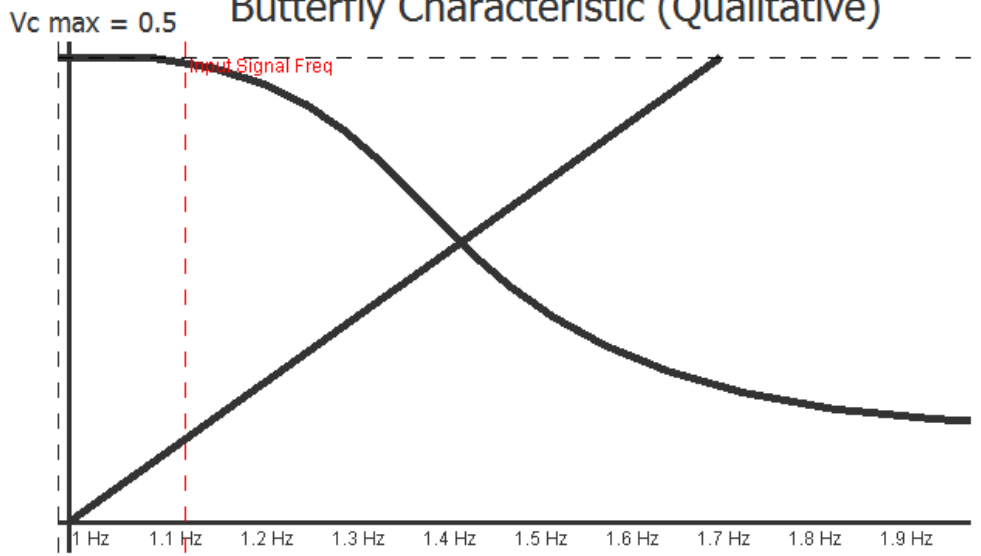
Sine Stop | Input Signal Frequency | 1.11 | Hz

VCO Central Frequency = 1 Hz

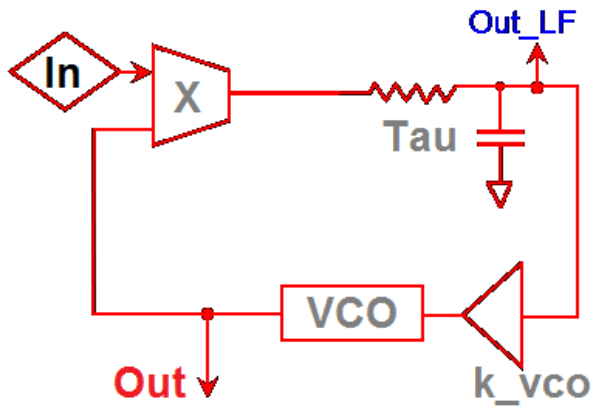
VCO Gain | VCO Gain value: 1.45

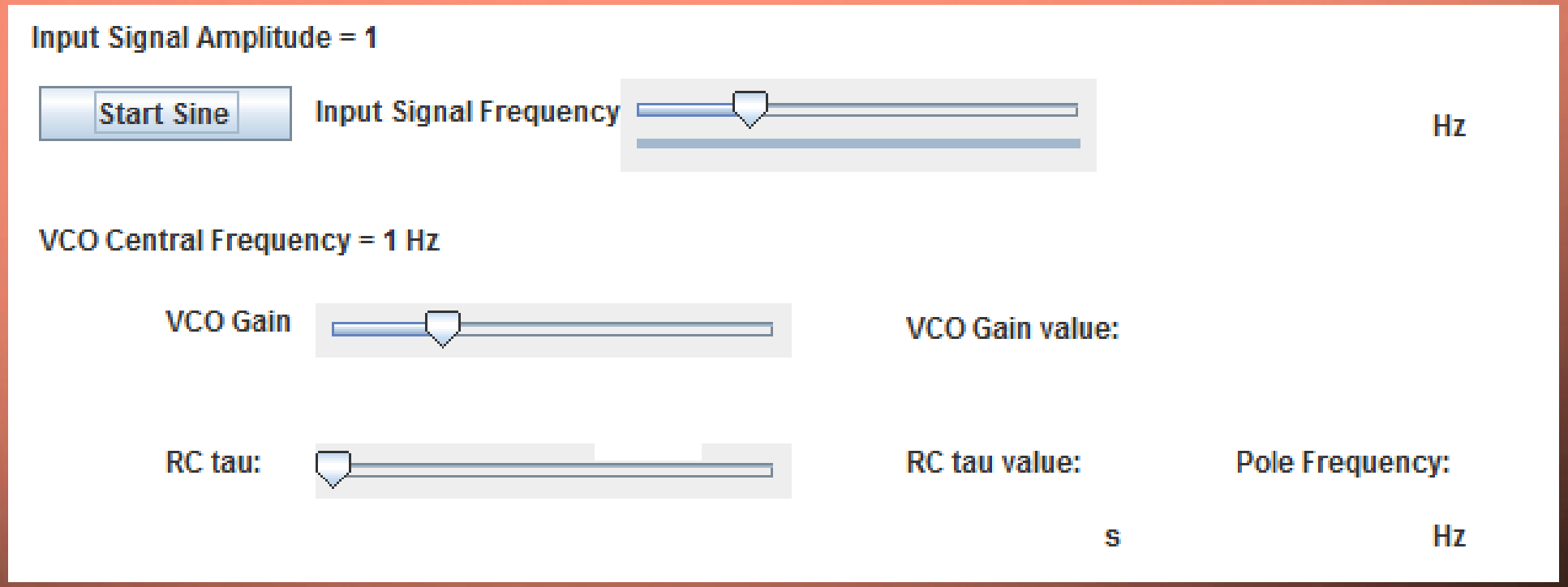
RC tau: | RC tau value: 0.11849641 s | Pole Frequency: 1.3431203 Hz

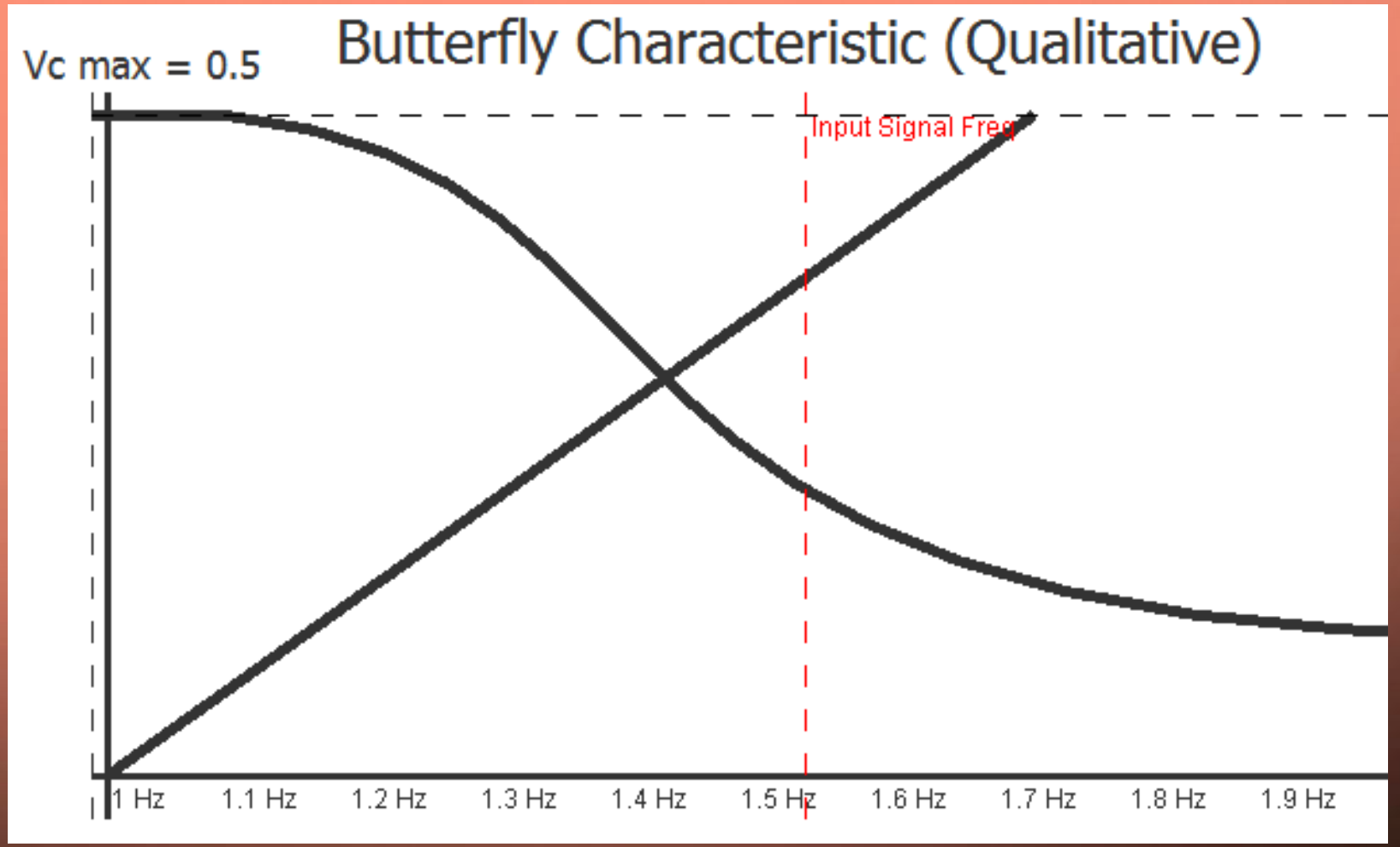
Butterfly Characteristic (Qualitative)



PLL Block Diagram:




Input Signal Amplitude = 1
Start Sine
Input Signal Frequency
Hz
VCO Central Frequency = 1 Hz
VCO Gain
VCO Gain value:
RC tau:
RC tau value:
Pole Frequency:
s
Hz


Butterfly Characteristic (Qualitative)
Vc max = 0.5
Input Signal Freq
1 Hz
1.1 Hz
1.2 Hz
1.3 Hz
1.4 Hz
1.5 Hz
1.6 Hz
1.7 Hz
1.8 Hz
1.9 Hz

REFERENCES:

1. **D. Del Corso: Elettronica per Telecomunicazioni**
   McGraw Hill, Settembre 2002 [Cap3]

2. **"The Automated Software phase locked loop and the exploration of ad adaptive algorithm for the adjustment of PLL Parameters" by Kevin Rolfes thesis** [Theory about realization of the Software PLL and Images related to results]
   Source:www.**rolfes**.org/**kevin_rolfes**_msee_thesis.pdf

3. **Roland E. Best: Phase-Locked Loops Theory, Design, and Applications**
   McGraw-Hill, 1993, ISBN 0-07-911386-9 [PLL Parameters and Theory related to PLL]