Exploration on Phase-Locked Loop

How possible is it to develop a phase-locked algorithm using high school knowledge only?

Table of Contents

Abstract

The mechanism of the phase-locked loop can be demystified with simple physics and mathematics. A model is built; an algorithm is developed, computationally simulated and mathematically verified; and eventually a design procedure is suggested. An application tool is also created to help with PLL design, providing users with the option among an industry standard, academic guideline, my proprietary suggestion, or users' pick. It's ultimately validated by practical experiments.

Introduction

Interest and Motivation

I am a baseball fan who loves to watch the live broadcast in my car. The video freezes sometimes when the car stops at traffic lights. Moving just a little further brings back video quality. It's allegedly caused by "phase difference" which, though curious, failed to understand until I learned sinusoidal functions and waves.

What is a phase?

If we imagine an ant crawling on an unit circle centered at origin on the X-Y plane, then phase, θ , is just its dynamic angle counterclockwise and the sine function value, $\sin(\theta)$, is assigned as the y-coordinate of the ant's shadow with light projection parallel to the x-axis. (*Figure.1-1*)

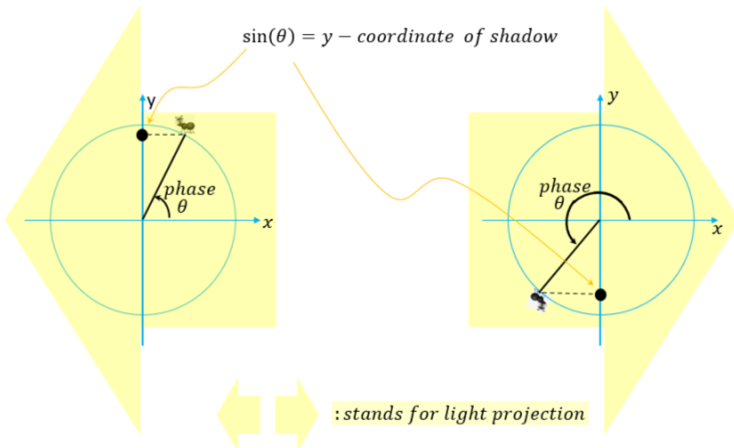


Figure.1-1 Phase and sine function

As the ant crawls in a circle, its phase is linearly proportional to time with the following linear relationship:

$\theta = 2\pi f t + \theta_0$ $\;\;, where\ \theta_0\ is\ the\ starting\ angle\ at\ t = 0\ \ and\ f\ is\ the\ number\ of\ cycle\ per\ unit\ time$

Therefore, the ***phase is just an expression of time*** in a different unit. (*Figure 1-2*)

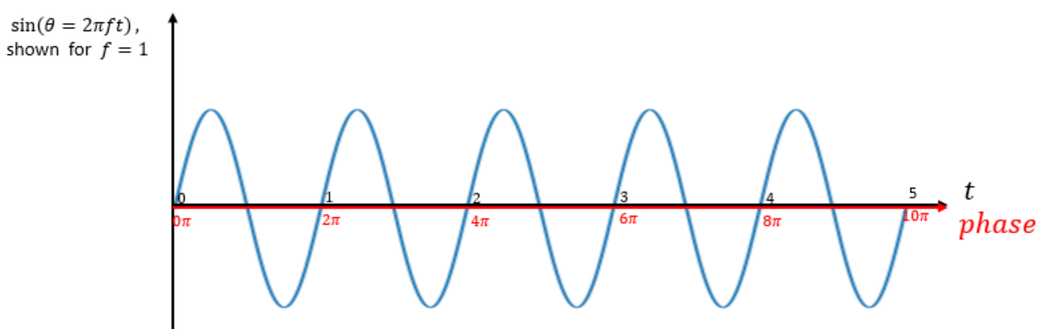


Figure 1-2 Sine function value vs time and phase

Why is there poor reception of broadcast?

For wireless communication, message $m(t)$ needs carrying by a sinusoidal wave $\sin(2\pi f t)$ to make the radio $m(t)\ \sin(2\pi f t)$ in the air. At the receiver end, the same sinusoidal wave shall be prepared to multiply the coming signal,

$$m(\,t)\sin(2\pi\, ft)\cdot\sin(2\pi ft)=\frac{1}{2}m(t)-\frac{1}{2}m(t)\cos(4\pi ft)$$

The message $m(t)$ could be retrieved as $\frac{1}{2}m(t)$ while the high frequency term is removed by special filters. However, if the received signal has some phase offset, θ_{offset} , to the locally prepared one, then the equation above becomes

$$m(t)\sin\left(2\pi ft+\theta_{offset}\right)\cdot\sin(2\pi ft)=\frac{1}{2}m(t)\cos\left(\theta_{offset}\right)-\frac{1}{2}m(t)\cos\left(4\pi ft+\theta_{offset}\right)$$

Again, the high frequency term could be filtered. But the retrieved message is in the form of $\frac{1}{2}m(t)\cos\left(\theta_{offset}\right)$, which means the retrieved message would diminish to nothing if θ_{offset} is $\frac{\pi}{2}$ and makes $\cos\left(\theta_{offset}\right)=0$.

Recalling that phase is equivalent to time, we can view θ_{offset} is the time for radio to travel from a station to a user. As the car is moving, θ_{offset} must be changing accordingly. It will make $\frac{1}{2}m(t)\cos\left(\theta_{offset}\right)$ vary all the time, close to zero once in a while, which explains why the video gets frozen because of nothing being updated during me watching the broadcast.

"Can the time-varying phase offset be removed?"

I searched for the answer to the above question on the internet and discovered phase difference can be eliminated by ***Phase-Locked Loop (PLL)*** technique adopted at the receiver end to adjust its sine wave. By digging deeper, I found the communication circumstances are even tougher--- not only is there phase offset for the incoming radio signal, but also its frequency suffers from *Doppler shift* since the car is moving. However, it's amazing that *PLL* still works to eliminate the frequency shift. This raises my interest and entrenches my intention to understand *PLL* in detail.

What does *PLL* do?

If the starts of each cycle of sine waves are marked as arrows, then the *PLL* function could be depicted as below to make all arrows aligned. (*Figure 1-3*)

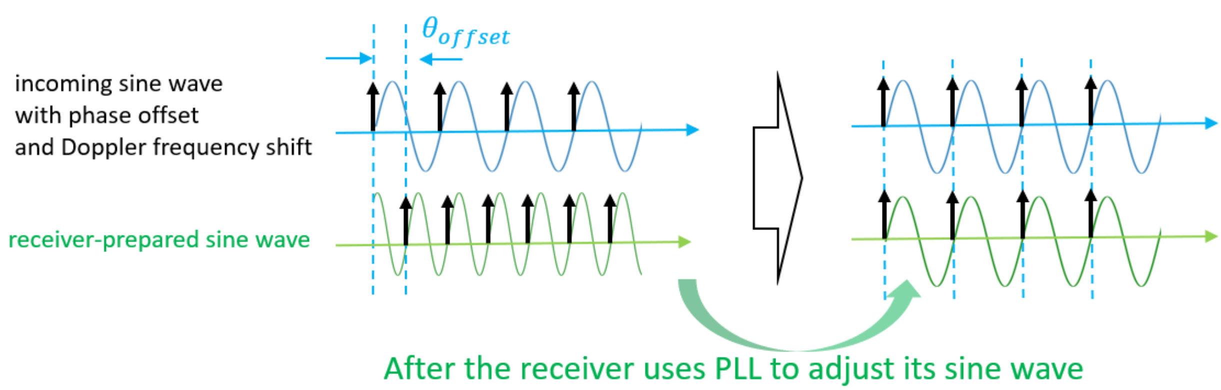


Figure 1-3 Function of Phase-Locked Loop (PLL)

There are many resources to introduce *PLL*, including open college classes (*19. Phased Locked Loops, 2013*) (*187N. Intro. To phase-locked loops (PLL) noise, 2019*) and industrial tutorials (*Analog Devices, 2009*). I could understand the functions of most individual components. The learning went smoothly until it came to the ***control algorithms***. *Laplace transform*, *Mason's loop theory* and *linear system analysis* are considered as prerequisites, all of which are way beyond the scope of high school study. However, the *TOK* of the *IB* program recommended the diverse paths of ways of knowing I should go on to discover with my acquired knowledge. It's very motivating to possibly share my exploration and help other high-school students understanding *PLL*.

Research of the PLL

The purpose of the *PLL* is to make a controllable signal source, after being divided by N times, to have the same phase as that of the reference signal. The basic functional blocks of the *PLL* go as *Figure 1-4.* (*187N. Intro. To phase-locked loop (PLL) noise, 2019*)

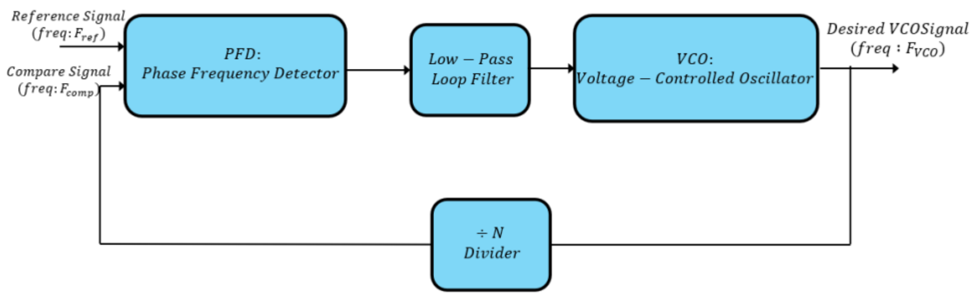


Figure 1-4 Architecture of PLL

Phase frequency detector (PFD): The phase frequency detector compares the timing of the rising edge (the arrow upright in *Figure 1.5*) of two clock signals. Equivalently, it compares when the cycles start.

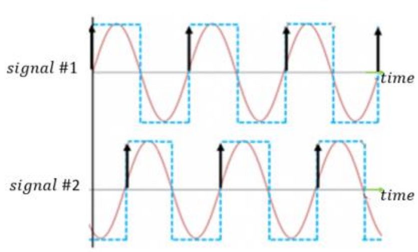


Figure. 1-5 Phase Frequency Detector (PFD) detects the starts of cycles for sinusoidal or pulse signal

PFD consists of the following logic devices (*Figure 1-6*).

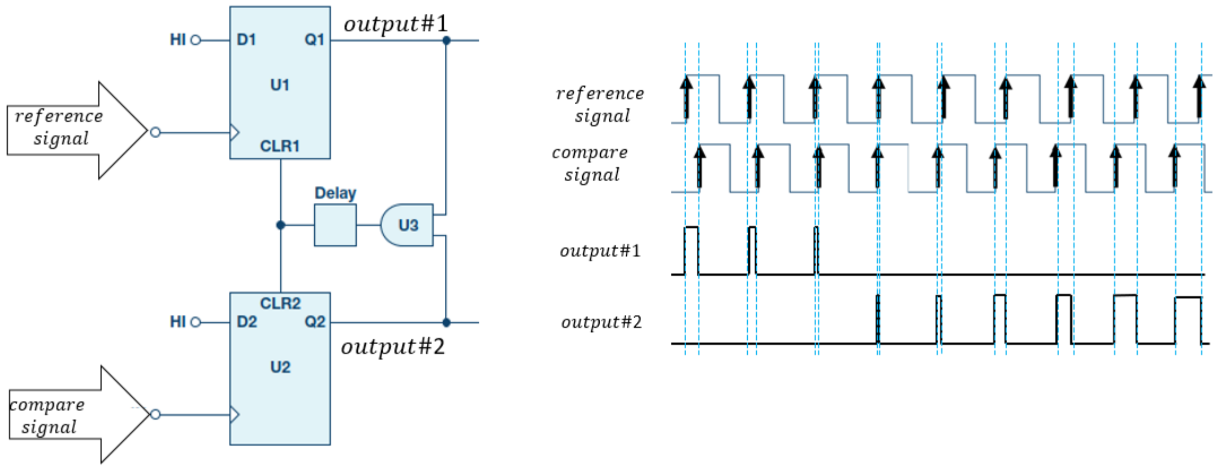


Figure 1-6. Architecture of Phase Frequency Detector (PFD)

U1 and U2 are D-flipflops. U1and U2 will set their outputs to "HI (high level) " every time they receive a rising edge, respectively. Once their outputs are both '1', U1 and U2 are reset to 0 and then wait for other rising edges to trigger them again.

Voltage-Controlled Oscillator (VCO): It outputs a sinusoidal wave of which the frequency is controllable by turning of control voltage. The ratio of frequency change, ΔF_{VCO} ,over control voltage change, $\Delta V_{control}$, is called the gain of the *VCO* and denoted as K_{VCO}. (Unit: Hz/V), i.e. $K_{VCO} = \frac{\Delta F_{VCO}}{\Delta V_{control}}$ *(Figure 1.7)*

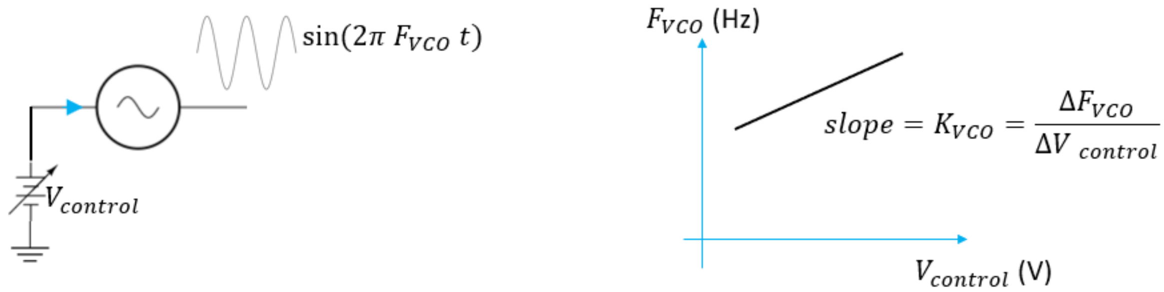


Figure 1.7 Voltage-Controlled Oscillator (VCO) and its Frequency-Voltage transfer function

N-divider : Most of the time, we are not simply replicating the reference signal but making a signal of higher frequency by N times. That is to say, we want this signal, when divided by N, to have the same phase (and the frequency accordingly) as that of the reference signal. *VCO*'s signal passing a divider produces a new signal, called a *comparison signal*, to have the frequency of $\frac{1}{N}$ times *VCO*'s original one. Accordingly, the comparison signal has frequency gain equal to $K_{comp} = \frac{K_{VCO}}{N}$. (Unit: *Hz/V*) (*Figure 1.8*)

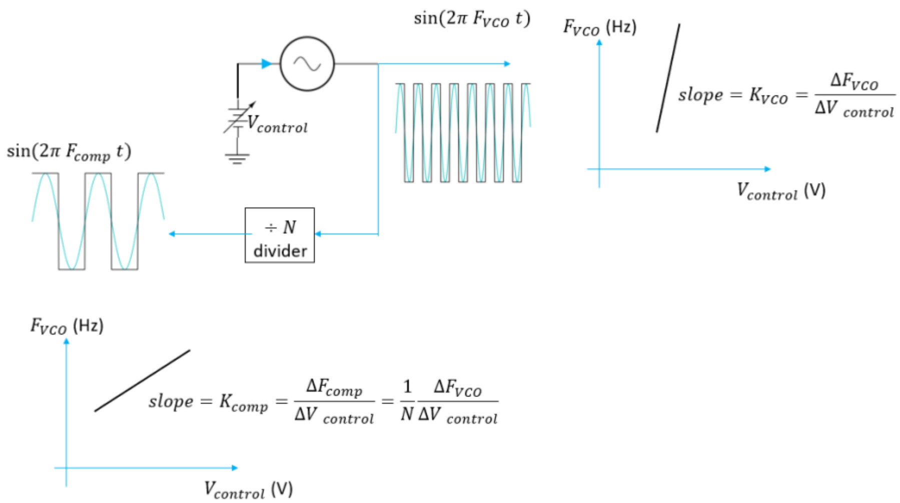


Figure 1.8 The gain of frequency over control voltage changes after N-divider

Loop filter: It's the very block to do the control work after receiving the fast/slow signaling from detectors. It takes action to adjust the VCO accordingly. From the appearance, it only consists of resistors and capacitors . However, it's the most difficult part where all the abtruse mathematical transforms and control theories are applied. Bypassing this block will leave my learning on *PLL* hollow. Therefore, I must figure out a way to crack this.

The Modeling and development of the algorithm

Most tutorials advise readers to understand *PLL* as an automobile racing. Based on my research, I believe the pace-keeping in the jogging scenario would be a better analogy since the *PLL* goal is not to race, but to keep the same tempo. My modeling goes as below.

Scenario:

Scenario description: A boy tries to keep pace with a girl who is jogging in a dark ground track. The only light is at the platform so that the boy could only see the girl pass platform but nowhere else. The boy could adjust his speed based on his awareness of whether he is late or early.

Correlation and mapping: I use a lap of the ground track (L) to illustrate a complete phase, 2π, of a periodic sine wave; therefore the location of the position on the track and full track length will be properly corresponding to phase and 2π. In a real *PLL* system, the *VCO*'s frequency is tunable; in my scenario, the speed of the boy is controllable. The boy's speed is adjusted when either the boy or girl passes the platform, just as *PFD* sets its output to affect *VCO*'s frequency when a rising edge of the two comparison signals arrives.

As for the loop filter, it corresponds to the ***control mechanism***, which is the very part I'm exploring in this essay. In my modeling, it is all about displacement, velocity, and acceleration which are well covered in physics. It looks promising for me to solve the mystery.

Intuitively, the boy shall increase or decrease his speed right away to mitigate any distance from the girl once he learns he is fast or slow. Here comes an initial strategy.

Algorithm Development:

Strategy #1: With speed jump only

The boy joins the jogging with normal speed V_B on a loop track of length L initially.

Upon seeing the girl passing the platform, he increases his speed by Δv_{fixed} to $V_B + \Delta v_{fixed}$, where Δv_{fixed} is a fixed speed jump. $V_B + \Delta v_{fixed}$ must be greater than the girl's speed V_G to make it possible for the boy to catch up with the girl. He keeps this high speed until he passes the platform as well; and then he returns back to his normal speed V_B.

Similarly, if the boy leads to pass the platform, he will reduce his speed by Δv_{fixed} to $V_B - \Delta v_{fixed}$ right away. He will keep this low speed until the girl passes the platform too; then the boy will return back to his original speed, V_B.

We can expect things will reach a steady state in the long run. It will be either of the two diagrams in *Figure 2-1*.

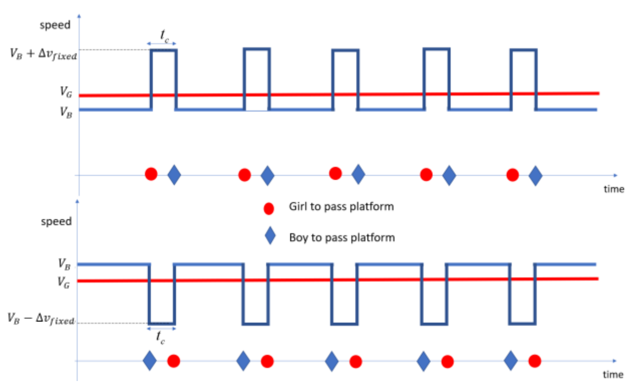


Figure 2-1. Speed vs Time in a long run (steady state)

The delay time (positive for lag, negative for ahead), t_c, will stay constant. In steady state, the boy and the girl will spend the same time, $\frac{L}{V_G}$, to finish a loop. Therefore, we get

$$\left(V_B + \Delta V_{fixed}\right) \times t_c + V_B \times \left(\frac{L}{V_G} - t_c\right) = L \text{ (The boy's normal speed } V_B \text{ is slower)}$$

Or

$$(V_B - \Delta V_{Fixed}) \times t_c + V_B \times \left(\frac{V}{V_G} - t_c\right) = L \text{ (The boy's normal speed } V_B \text{ is faster)}$$

In either case, it concludes $\rightarrow t_c = L \times \left|\frac{V_B - V_G}{V_G \times \Delta v_{fixed}}\right|$

Hence, I conclude, in steady state, the delay time reflects the speed difference. Besides, the boy and girl still never pass the platform at the same time (i.e. $t_c = 0$) unless V_B could be adjusted and equal to V_G.

Based on this, V_B shall be adjusted to approach V_G in order to make the synchronization. The boy cannot just return to the original speed after temporary velocity increase/decrease.

It seems trivial for this initial strategy to fail because the boy only switches between two speeds while the goal is for him to have a steady single speed equal to the girl's. This strategy is just intended to explore any clues out of foreseen failures.

Strategy #2: With speed jump and constant acceleration

The algorithm should be modified, following up on the initial strategy.

After carrying out the first strategy for a while to know the delay time which reflects speed difference, the boy should have additional follow-up below so that the more delay time, the more speed change the boy could make.

Upon seeing the girl pass the platform, the boy not only abruptly raises his speed from V_B to $V_B + \Delta v_{fixed}$ but also accelerates at constant acceleration, a_{fixed}, until he passes the platform; and then he reduces his speed by the fixed speed jump Δv_{fixed}.

Similarly, if the boy leads to pass the platform, not only does he abruptly lower down his speed form V_B to $V_B - \Delta v_{fixed}$ but also decelerates at a_{fixed} until the girl passes the platform as well; and then he increases his speed by the speed jump (fall), Δv_{fixed}.

Depicting the idea below shows the revised strategy is likely to work (*Figure 2-2*).

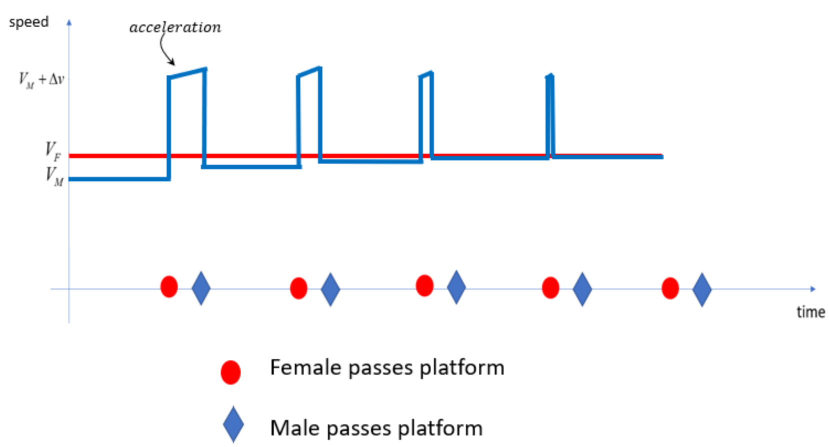


Figure 2-2. Intuition on boy's speed change in strategy #2.

The prerequisite for this strategy is that the boy should take constant acceleration as follow-up only after jogging for a while. However, if the acceleration is low enough, the transition of strategy #1 to #2 is negligible and I don't expect much difference between the results of *'strategy #1 and then #2'* and *'strategy #2 directly'*. It would be solid if I could validate the thought scientifically by writing a code to simulate (*Appendix A*).

The code is mainly to execute the following operations:

At each time[n]

1. *Check the current flags that reflect whether the boy, the girl or neither has passed the platform and decide the boy is in mode of constant acceleration, constant deacceleration or constant speed. Calculate the boy's next speed. Calculate the next positions for the boy and girl using the current speed.*
2. *If either of the boy's or girl's next positions exceeds ground track, change flags , modulo the position and further adjust the boys' calculated next speed*

My simulation results preliminarily prove that the modified algorithm does work (*Figure 2-4*). The distance between the boy and girl reaches zero at the end.

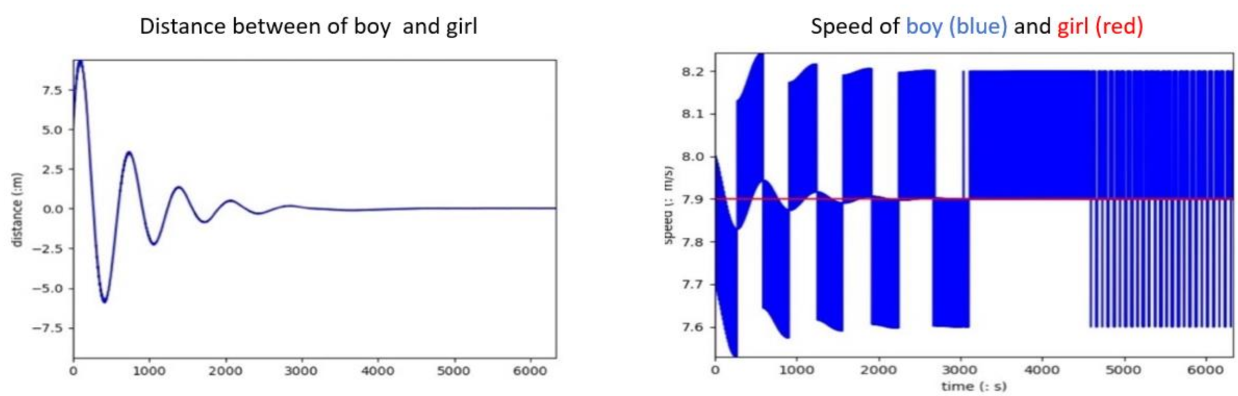


Figure.2-4 Result of jogging simulation.

Implementation of the Algorithm

The expected control signal on boy's speed seems quite odd (*Figure 3-1*). Can it be carried out in voltage to control the *Voltage-Controlled Oscillator (VCO)*?

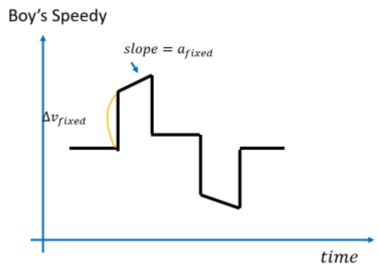


Figure 3-1 Waveform of boy's speed

It's trivial to get the voltage pulse train using a voltage source and a switch. It serves the same purpose if we replace the voltage source with a current one and attaching a resistor , according to *'V = IR'*. By adding in series another electrical component, capacitor, to accumulate the voltage, the waveform turns out exactly what is desired. (*Figure 3-2*)

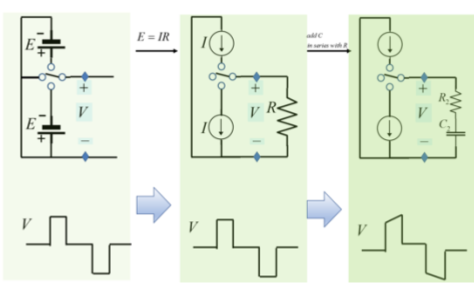


Figure.3-2 Current source together with peripheral components to make control signal.

Later I learned that the current source in my interference is called a ***charge pump***. It's usually integrated into ***PFD,*** together with the N-divider, as an integrated circuit (*IC*) called synthesizer. (*Figure 3-3*).

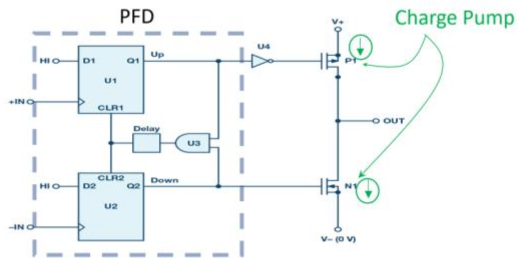


Figure 3-3 PFD IC to include charge pump.

VCO's frequency is controlled by voltage with frequency-to-voltage gain, K_{VCO}. The frequency gain will become $K_{comp} = \frac{K_{VCO}}{N}$ after the divider. Then we can come up with the following conversion. (*Figure 3-4*)

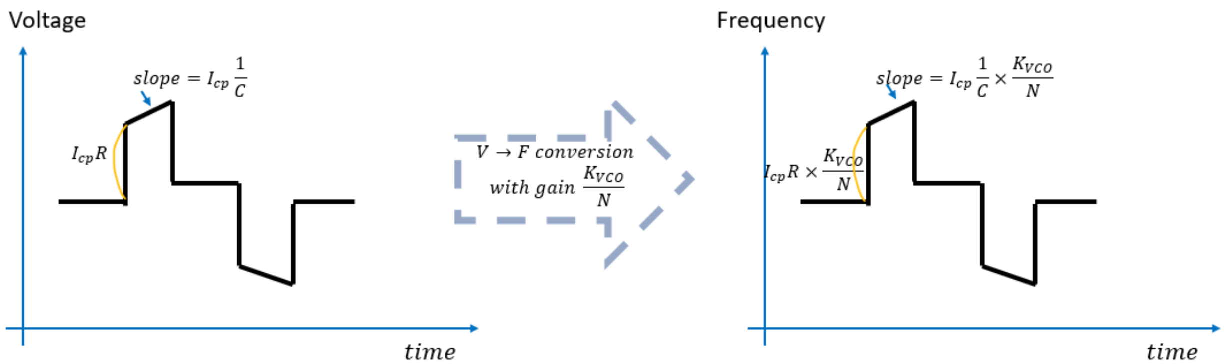


Figure. 3-4. Parameter conversion from controlled voltage to controlled frequency.

Therefore, it is feasible to implement my *PLL* algorithm with real circuitry (*Figure 3-5*)

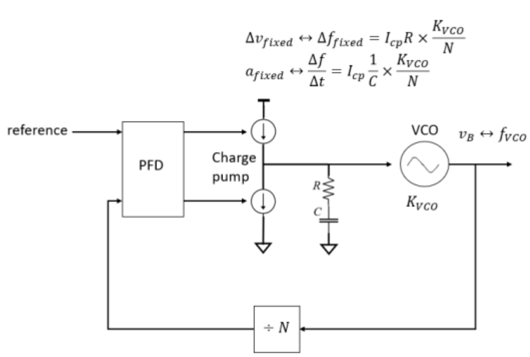


Figure 3-5. Circuitry to carry out Troy's PLL algorithm.

Theoretical verification and improvement

Comparison

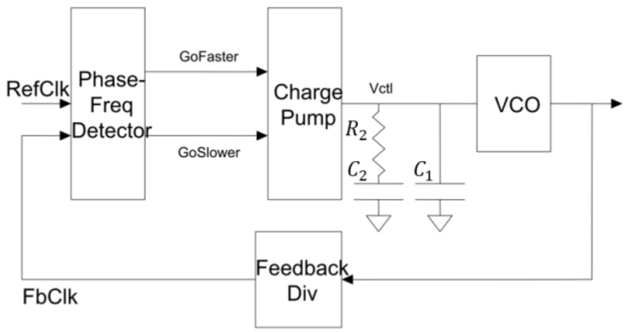


Figure 4-1 PLL reference paper from IEEE

Compared with a reference paper from *IEEE* (*Fischette, 2007*) (*Figure 4-1*), my design seems to have one capacitor missing (*C1 in the IEEE paper*) in the ***loop filter topology***.

(i) *Can I prove my algorithm still works analytically, in addition to the previous simulation on concept?*

(ii) *Can I figure out what C1 is for?*

Mathematical Analysis

Assuming V_G is constant, the girl will pass the platform at $t = 0, T_0, 2T_0, 3T_0, \ldots$, $where\ \ T_0 = \frac{L}{V_G}$. We will denote the boy's distance from the girl by $x[n]$ when the girl passes the platform at nT_0. The boy keeps constant acceleration after a velocity jump Δv_{fixed} to make up $x[n]$ in $t_c[n]$ until he passes the platform. (*Figure 4-2*)

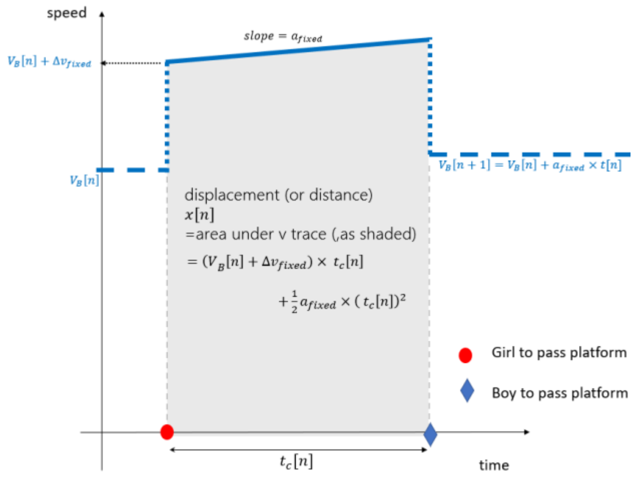


Figure 4-2 Boy's speed vs time plot for nth cycle

Either by the speed-time area calculation or the displacement formula for constant acceleration, we get

$$x[n] = \left(V_B[n] + \Delta v_{fixed}\right) \times t_c[n] + \frac{1}{2} a_{fixed} \times (t_c[n])^2$$

Not to be trapped by the complicated equation, I assume a simple case , $a_{fixed} \ll \Delta v_{fixed} \ll V_B[n] \cong V_G$, so that we get

$$t_c[n] \cong \frac{x[n]}{V_B[n] + \Delta v_{fixed}} \cong \frac{x[n]}{V_G}$$

Then $t_c[n]$ is linearly proportional to $x[n]$.

The boy's speed after n^{th} cycle is equal to the original speed $V_B[0]$ plus all the accumulated speed change.

$$V_B[n+1] = V_B[0] + \sum_{i=1}^{n} a_{fixed} \times t_c[i]$$

$$= V_B[0] + \sum_{i=1}^{n} a_{fixed} \times \frac{x[i]}{V_G}$$

$$= V_B[0] + \frac{a_{fixed}}{V_G} \sum_{i=1}^{n} x[i]$$

Next, we will calculate the distance between the boy and girl for nth cycle and plot their speed vs time below. (*Figure 4-3*)

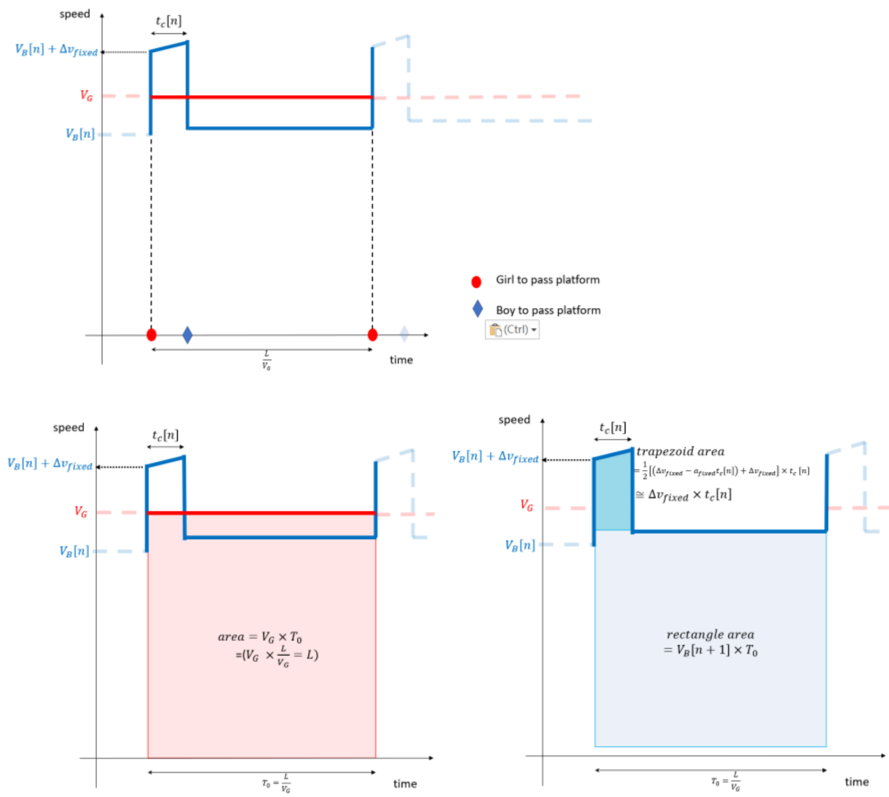


Figure 4-3 Speed-time plot to calculate respective distance run in nth cycle

The distance apart for the next cycle will be:

$x[n+1]$

$= x[n] + distance\ girl\ runs(red) - distance\ boy\ runs(blue)$

$= x[n] + V_G \times T_0 - (V_B[n+1] \times T_0 + \Delta v_{fixed} \times t_c[n])$

$= x[n] + (V_G - V_B[n+1]) \times T_0 + \Delta v_{fixed} \times t_c[n]$

$$= x[n] + (V_G - V_B[n+1]) \times T_0 - \frac{\Delta v_{fixed}}{V_G} \times x[n]$$

$$\rightarrow x[n+1] - x[n] = (V_G - V_B[n+1]) \times T_0 - \frac{\Delta v_{fixed}}{V_G} x[n] \ldots\ldots\ldots (2)$$

Then dividing T_0 on both sides, we get

$$\frac{x[n+1] - x[n]}{T_0} = (V_G - V_B[n+1]) - \frac{\Delta v_{fixed}}{V_G T_0} x[n]$$

$$= (V_G - V_B[0] - \frac{a_{fixed}}{V_G} \sum_{i=1}^{n} x[i]) - \frac{\Delta v_{fixed}}{V_G T_0} x[n]$$

$$= -(V_B[0] - V_G) - \frac{\Delta v_{fixed}}{V_G T_0} x[n] - \frac{a_{fixed}}{V_G} \sum_{i=1}^{n} x[i]$$

$$= -(V_B[0] - V_G) - \frac{\Delta v_{fixed}}{V_G T_0} x[n] - \frac{a_{fixed}}{V_G T_0} \sum_{i=1}^{n} x[i]\, T_0$$

$$\rightarrow \frac{x[n+1] - x[n]}{T_0} = -p - qx[n] - r \sum_{i=1}^{n} x[i] T_0$$

$$, where\ p = (V_B[0] - V_G),\ \ q = \frac{\Delta v_{fixed}}{V_G T_0}, r = \frac{a_{fixed}}{V_G T_0}$$

Since T_0 is small, we might as well replace it to be Δt

$$\frac{x[n+1] - x[n]}{\Delta t} = -p - qx[n] - r \sum_{i=1}^{n} x[i] \Delta t \qquad , where\ \Delta t\ is\ very\ small$$

The formula above is just an expression of differential/integral equation in series. I rewrite it in normal form.

$$\frac{dx(t)}{dt} = -p - qx(t) - r \int_0^t x(t) dt \qquad , where\ p = (V_B[0] - V_G), q = \frac{\Delta v_{fixed}}{V_G T_0}, r = \frac{a_{fixed}}{V_G T_0}$$

Differentiating both sides again,

$$\frac{d^2 x(t)}{dt^2} = -q \frac{dx(t)}{dt} - rx(t)$$

$$\frac{d^2x(t)}{dt^2} + q\frac{dx(t)}{dt} + rx(t) = 0 \qquad ,where\ q = \frac{\Delta v_{fixed}}{V_G T_0}, r = \frac{a_{fixed}}{V_G T_0}$$

The solution will be in the following form

$$x(t) = Ae^{-\left(\frac{q}{2}-\frac{\sqrt{q^2-4r}}{2}\right)t} + Be^{-\left(\frac{q}{2}+\frac{\sqrt{q^2-4r}}{2}\right)t}\ if\ q^2 - 4r \neq 0$$

From the above,

(i) If $q^2 - 4r > 0$, we know that the longer time constant will be

$$\tau = \frac{1}{\frac{q}{2} - \frac{\sqrt{q^2-4r}}{2}} = \frac{2}{q - \sqrt{q^2 - 4r}}$$

Choosing $q^2 - 4r \cong 0$ gets this dominant time constant to be smallest, equal to $\frac{2}{q}$

(ii) If $q^2 - 4r < 0$,

$$x(t) = Ae^{-\frac{q}{2}t}e^{-j\frac{\sqrt{4r-q^2}}{2}t} + Be^{-\frac{q}{2}t}e^{+j\frac{\sqrt{4r-q^2}}{2}t}$$

$x(t)$ will be decaying to $\frac{1}{e}$ with a fixed time constant $\frac{2}{q}$ while showing oscillation phenomena.

From (i) and (ii), it numerically proves that my algorithm works to make $x(t)$, the phase difference, decay to zero for sure. The best time constant for convergence is also derived as $\frac{2}{q}$ $(= \frac{2V_G T_0}{\Delta v_{fixed}})$

Decide Components Values for PLL Design

The derived mathematical equation not only shows my algorithm works but also provides a way to decide the values of those components used in *PLL* design.

I replace the speed, distance, fixed speed jump and acceleration with electrical counterparts in real *PLL* below. (*Figure 4-4*)

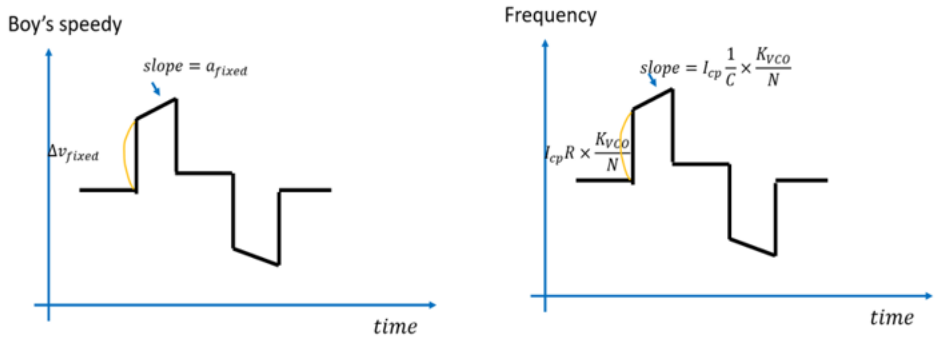


Figure 4-4 Mapping between jogging scenario and practical application

Because distance, $x(t)$, is mapped to phase difference, $phaseDifference(t)$, then

$$\frac{d^2x(t)}{dt^2}+q\frac{dx(t)}{dt}+rx(t)=0, \qquad where\ q=\frac{\Delta v_{fixed}}{V_G T_0}, r=\frac{a_{fixed}}{V_G T_0}$$

will turn out:

$$\frac{d^2[phaseDifference(t)]}{dt^2}+q\frac{d[phaseDifference(t)]}{dt}+r[phaseDifference(t)]=0$$

$$where\ q=\frac{I_{CP}R\frac{K_{VCO}}{N}}{F_{ref}T_0}=\frac{I_{CP}RK_{VCO}}{N}, r=\frac{I_{CP}C\frac{K_{VCO}}{N}}{F_{ref}T_0}=\frac{I_{CP}K_{VCO}}{CN}$$

Procedure to decide R and C

1. Usually specified is the lock time, t_{lock}, when the PLL shall reach the final value. We can choose $\tau=\frac{t_{lock}}{15}$ to make sure the phase difference has enough time margin to decay to zero.

According to the desire time constant and relation $\tau = \frac{2}{q} = \frac{2N}{I_{CP}RK_{VCO}}$, the R value to pick would be:

$$R = \frac{2N}{I_{CP}K_{VCO}\tau}, or\ alternatively, R = \frac{30N}{I_{CP}K_{VCO}t_{lock}}$$

2. The shortest time constant is made when $q^2 - 4r \leq 0$ is met:

$$\rightarrow (\frac{I_{CP}RK_{VCO}}{N})^2 - 4\frac{I_{CP}K_{VCO}}{CN} \leq 0$$

$$\rightarrow C \leq \frac{4I_{CP}K_{VCO}N}{(I_{CP}RK_{VCO})^2} = \frac{4N}{I_{CP}K_{VCO}R^2}$$

So, the capacitance would be picked by:

$$C = \frac{4N}{I_{CP}K_{VCO}R^2}$$

Is C1 necessary?

My algorithm is mathematically proven to work with practical implementation suggested. Then why C1?

I revisited the result of jogging simulation to find some room for improvement. The boy's speed still has large random jump even after his distance from the girl is almost made zero. That's because the boy abruptly changes his speed by Δv_{fixed} despite tiny time delay/ahead to pass the platform. It could be considered an over-reaction, which could be prevented if we allow the boy to "take time" to make the needed speed change Δv_{fixed} with finite acceleration. He could still make the change Δv_{fixed} in short time; but if the time delay/ahead is extremely small, he turns out to decrease or increase speed by a negligible amount. It will result in negligible speed fluctuation. (*Figure 4-5*)

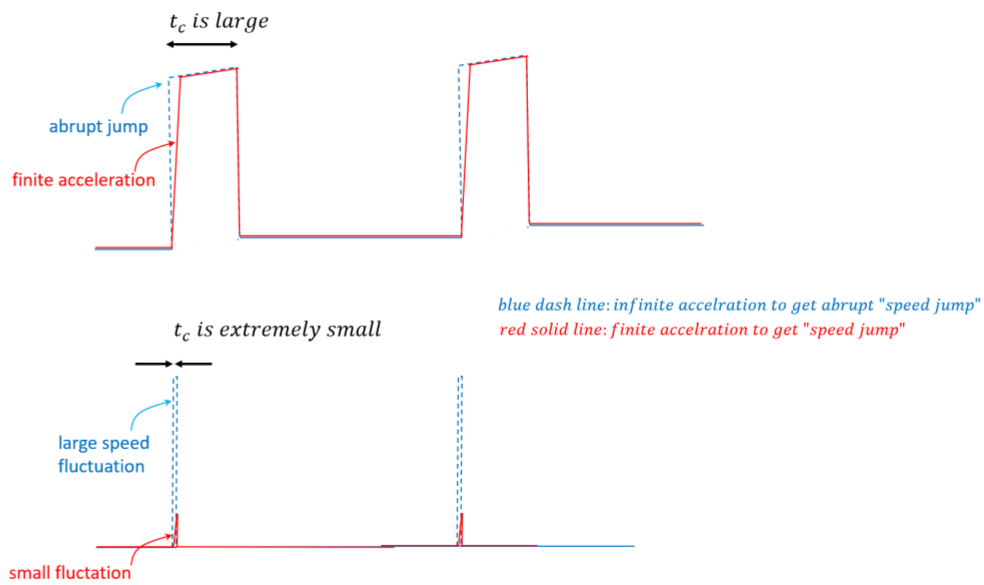


Figure 4-5 Depict finite acceleration cause negligible speed change for extremely small asynchronization.

This makes the strategy #3. (*Figure 4-6*)

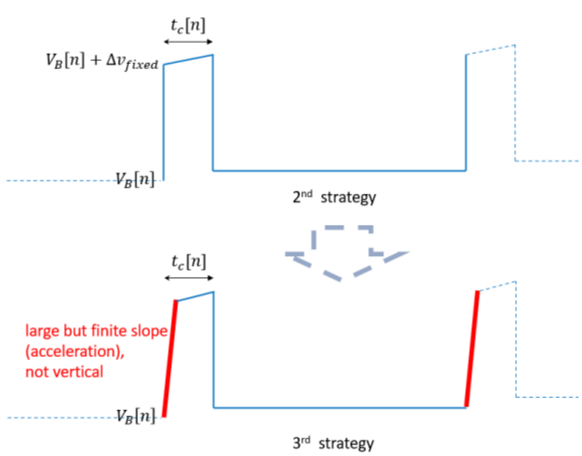


Figure 4-6 2nd strategy migrates to 3rd strategy

Strategy #3: With high acceleration to make speed jump and then with low constant accelerations

Upon seeing the girl passing the platform, the boy shall quickly increase his speed V_B to $V_B + \Delta v_{fixed}$ by high acceleration a_{fixed_high}. Then, he accelerates at constant low acceleration, a_{fixed_low} . After the boy passes the platform, he reduces his speed by Δv_{fixed}.

Similarly, if the boy leads to pass the platform, the boy shall quickly decrease his speed V_B to $V_B - \Delta v_{fixed}$ by high deceleration a_{fixed_high}.Then, he decelerates at a_{fixed_low} . After the girl passes the platform as well, the boy increases his speed by Δv_{fixed}

Implementation of Strategy #3

Adding a capacitor, C_1 in the following figure, makes the abrupt jump have finite slope because charging C_1 takes time (*Figure 4-7*).

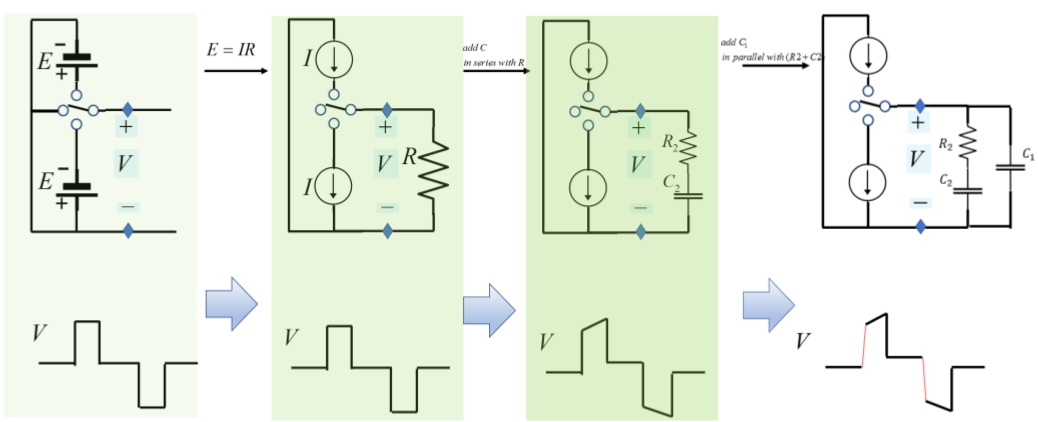


Figure 4-7 Adding C_1 makes an abrupt voltage jump of finite slope

The voltage to frequency conversion will goes as below (*Figure 4-8*)

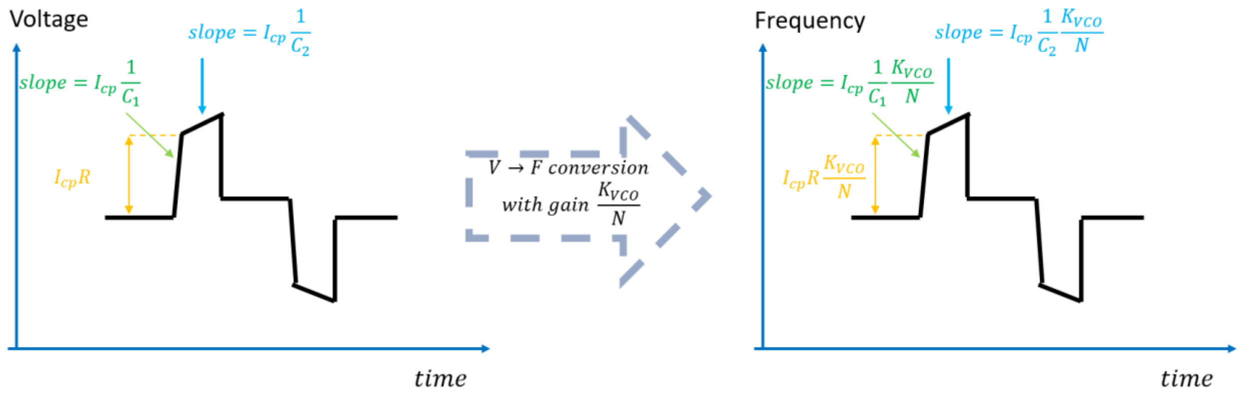


Figure 4-8 Parameter conversion from controlled voltage to controlled frequency.

If we let C_1 small enough, say $C_1 = \frac{C_2}{10}$, it won't substantially affect the previous formula and result.

Complete PLL Design Procedure

It's time to consolidate the exploration and algorithm and make simple the guideline of the complete *PLL* design. (*Figure 4-9*)

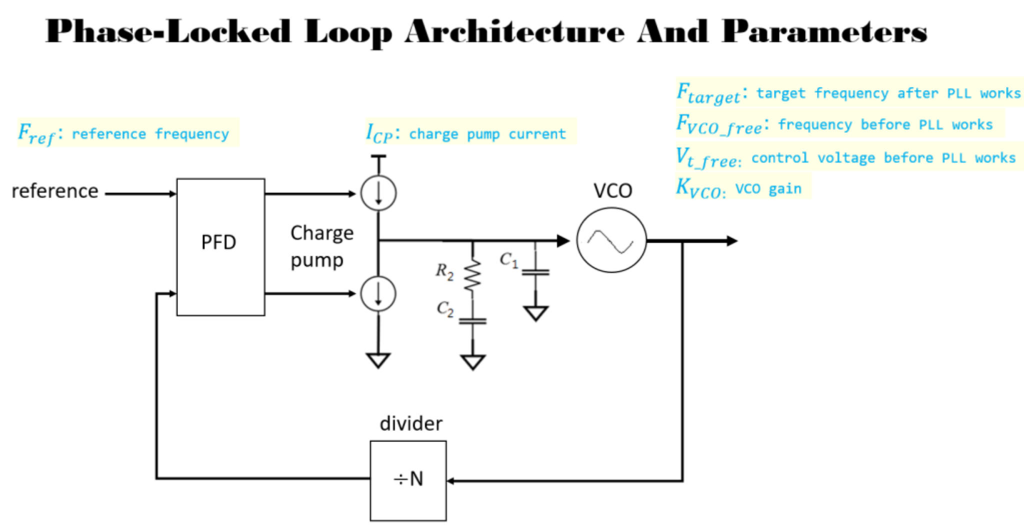


Figure 4-9 Complete design of PLL

Step #1: To learn VCO's V-F gain (K_{VCO}) and divide value (N)

Step #2: To learn PFD's charge pump current (I_{CP})

Step #3: To learn the lock time (t_{lock}) within which the phase difference shall decay to zero

Step #4: To decide R_2 *by* $R_2 = \frac{30N}{I_{CP}K_{VCO}t_{lock}}$

Step #5: To decide C_2 *by* $C_2 = \frac{4N}{I_{CP}K_{VCO}{R_2}^2}$

Step #6: To decide C_1 *by* $C_1 = \frac{C_2}{10}$

Experimental validation

I consulted several experts in industry for design references. *Richard Huang*, an advanced Electronic Engineer in Garmin, introduced me to two typical design guidelines, each from a textbook and application note from Philips (NXP now). I'll compare my design result with these two references, according to the practical parameters and specification for a PLL of GPS. (*Appendix B, C*). My practical design is described in the appendix D.

I'm pleased to learn my design values are within 10% difference from NXP's, and I notice that, even for the two recommended Rohde's and NXP's notes, there's significant difference for design values. There seems to be some design flexibility in general.

Virtual validation with simulation

Due to COVID-19, I failed to visit Garmin to conduct my experiment in the laboratory. Before I made my experiment in my bedroom, I developed an application tool to simulate the full phase locked loop. It basically follows my flow chart of the code simulating jogging model (Appendix A).

Simulation Challenge: The major challenge is to resolve differential equation set consisting of KCL for nodes and I-V characteristic of C1, R2 and C2. It's involved in differential-difference conversion and inverse matrix. (solution in *appendix E*).

Simulation Result (Video Demo): (Wu, 2020)

(1) The simulation confirms that both the phase and frequency differences could be eliminated in 1mS as expected. The behavior of my design is close to NXP's. *(Figure 5-3, Figure 5-4*)

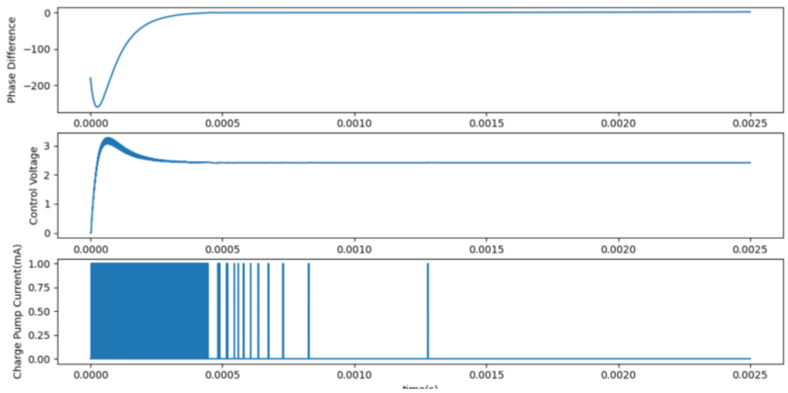


Figure5-3 Simulation result with Troy Wu's design guideline

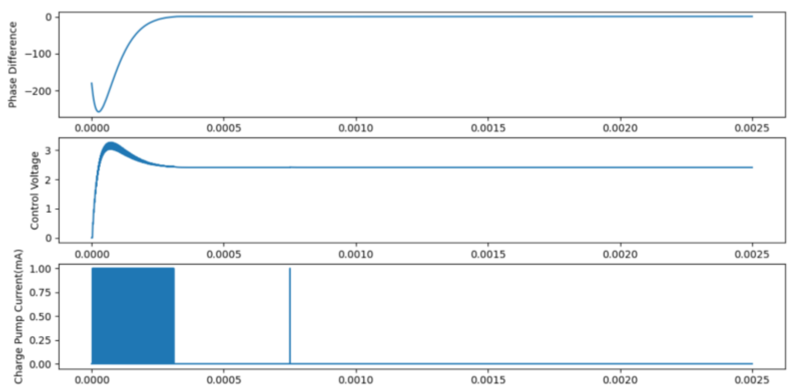


Figure5-4 Simulation result with NXP's design guideline

(2) Ulrich Rohde's solution with default ratio=10 is not good enough.(*Figure 5-5*)

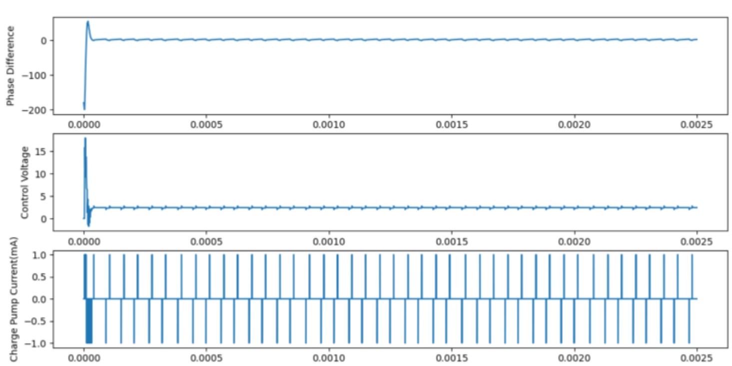


Figure5-5 Simulation result with Ulrich Rohde's design guideline with option=10

After tweaking ratio option to be 100, it performs way better. (*Figure 5-6*)

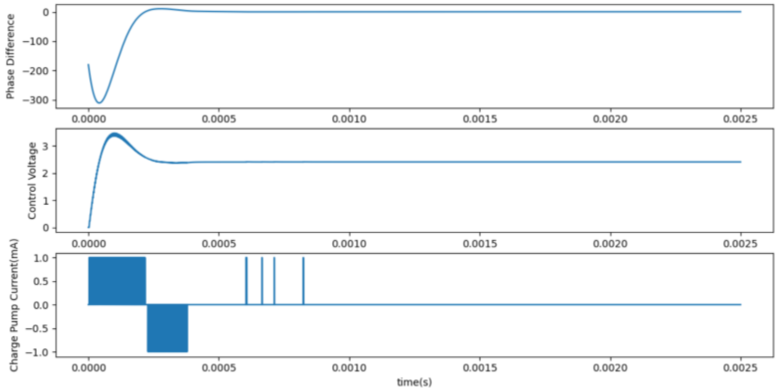


Figure 5-6 Simulation result with Ulrich Rohde's design guideline with option=100

(3) Removal of C1 still works. (*Figure5-7*). As expected, we can observe the control voltage is "spiny" if without C1.

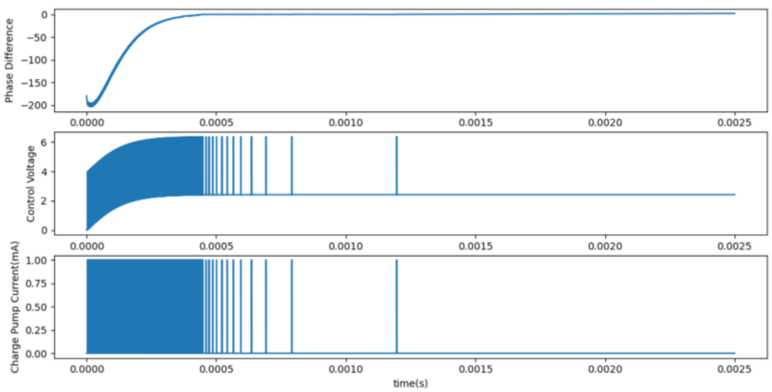


Figure5-7 Simulation result with Troy Wu's design guideline but without C1

Practical validation through experiment

I am grateful that Garmin supports me with *PLL* modules and the rework to my design values. (*Figure 5-8*). I bought an oscilloscope of bandwidth *100MHz* (Rigol DS1102Z-E) and prescalers (Fujitsu MB506 Evaluation Board) which divide the PLL outputs to fit in the oscilloscope operation range. Because the wiring is complicated (*Figure 5-9*), I will adopt abstract blocks to describe the setup.

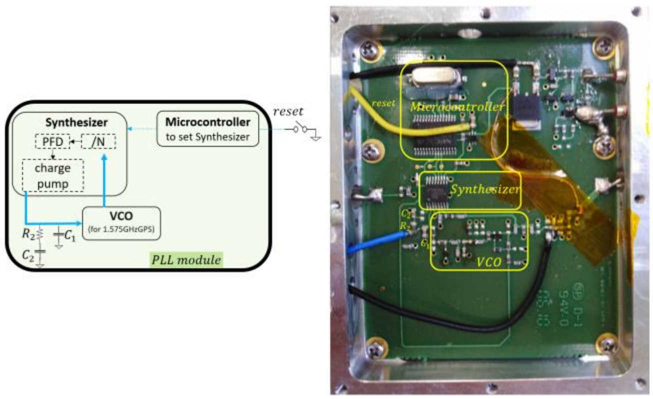


Figure 5-8 PLL module of GPS frequency 1.57542GHz (Courtesy of Garmin)

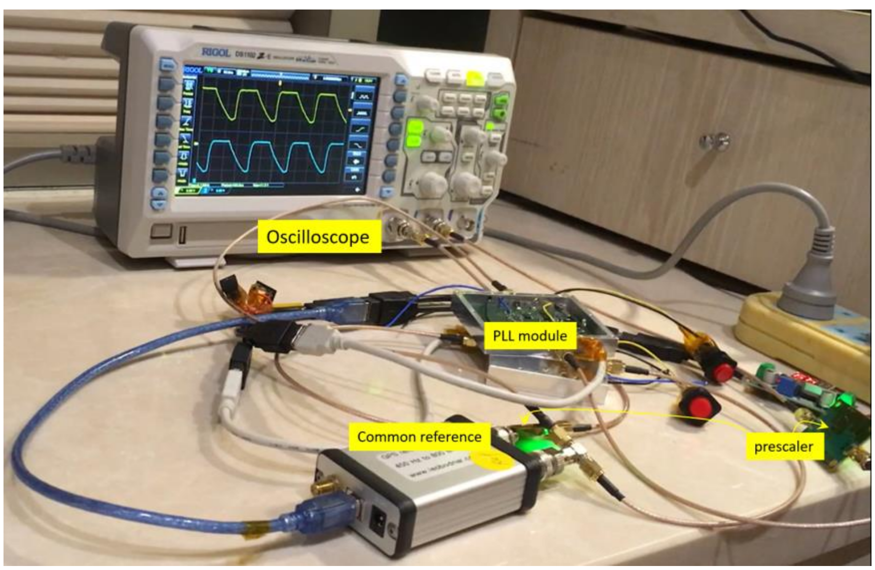


Figure 5-9 Picture of setup to learn whether two PLL modules work simultaneously.

1. Does control voltage go as simulated?

   Setup (Figure 5-10): (Extended Essay PLL Experiment 1, 2020)

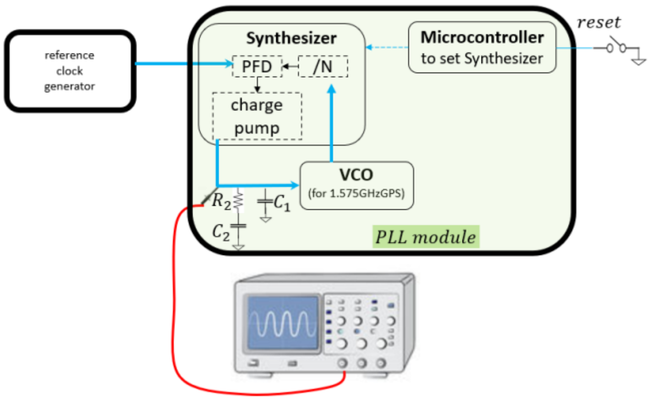
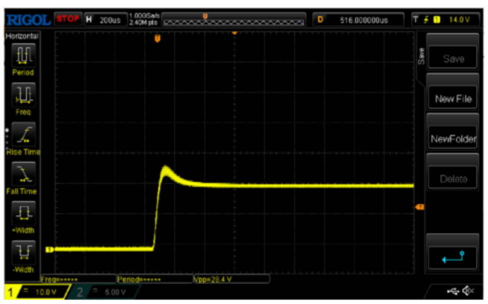
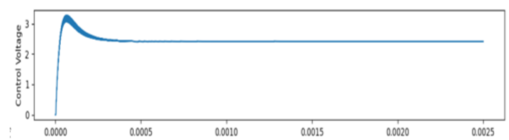


Figure 5-10 Setup for measurement of control voltage

Result (*Figure 5-11*, *Figure 5-12*):



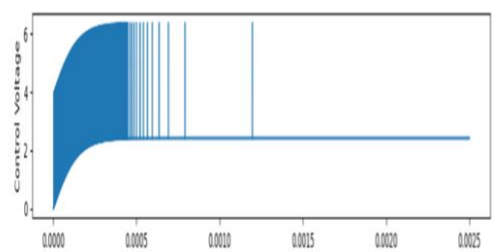
Measured control voltage



Simulated control voltage

Figure 5-11 Comparison between measured and simulated control voltages with loop filter consisting of C1, R2 and C2.

Measured control voltage



Simulated control voltage

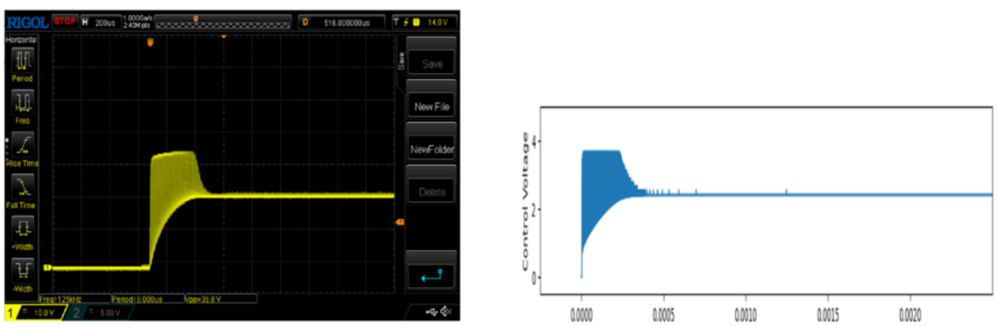
Figure 5-12 Comparison between measured and simulated control voltages with loop filter consisting of R2 and C2 only, without C1.

Analysis:

(i) The waveforms of control voltage just go as simulated by my application tool. It gets stable in 500us with C1 and 800us without C1, both meeting the target $t_{lock} \leq 1ms$

(ii) The waveform without C1 goes as spiny as simulated. However, the amplitude over 3.68V is truncated. It is reasonable because we supply the board with 5V only. To reflect this practical consideration, I modified my code to allow users to set limitation on control voltage.

The result of revised code goes as below.0.2nF capacitance is set to C1 to reflect parasitic capacitance everywhere. The simulation result matches the experiment well.(*Figure 5-13*)



Measured control voltage

Simulated control voltage

Figure 5-13 Comparison between measured and simulated control voltages of loop filter without C1, after voltage ceiling is added in simulation code.

2. Whether PLL works? (Measure the synchronization between PLL #1 and PLL #2, 2020)

I have two PLL modules on hand, each with C1 and without C1. I expect they are phase-locked to each other if they are phase-locked to a common reference. It means the other (CH2) shall be stationary to the triggered one (CH1 in this case).

Setup (*Figure 5-14*):

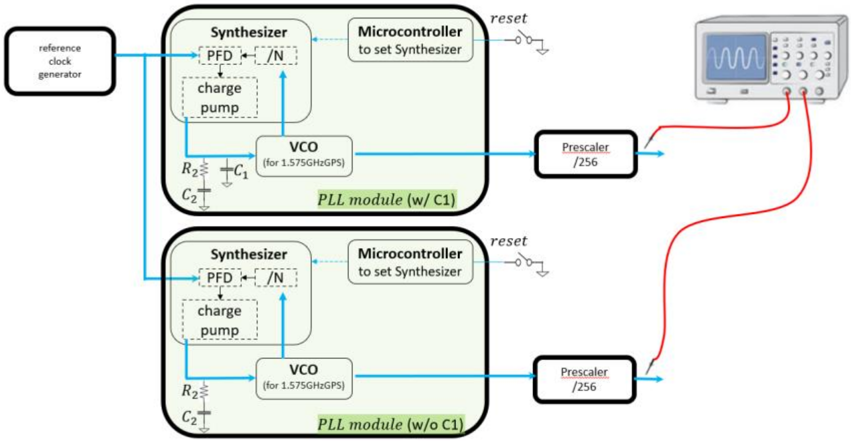


Figure 5-14 Setup to check out whether PLLs work

Result and Analysis:

The result shows before PLLs are enabled, CH2 is “running” with reference to CH1. After PLLs are enabled, CH2 becomes stationary to CH1, indicating their phases are locked to each other. (*Figure 5-15*)

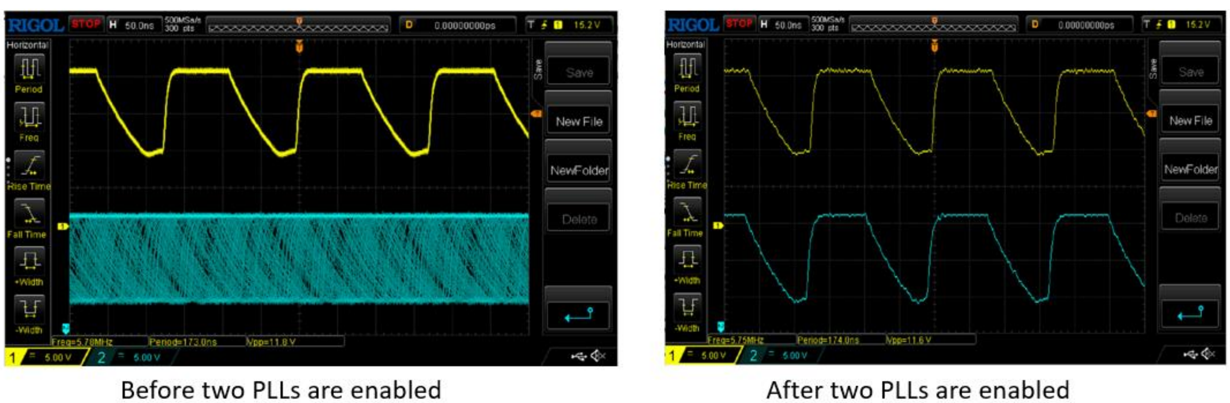


Figure 5-15 Two PLLs have phases locked to each other after enabled.

Conclusion

Goal Review

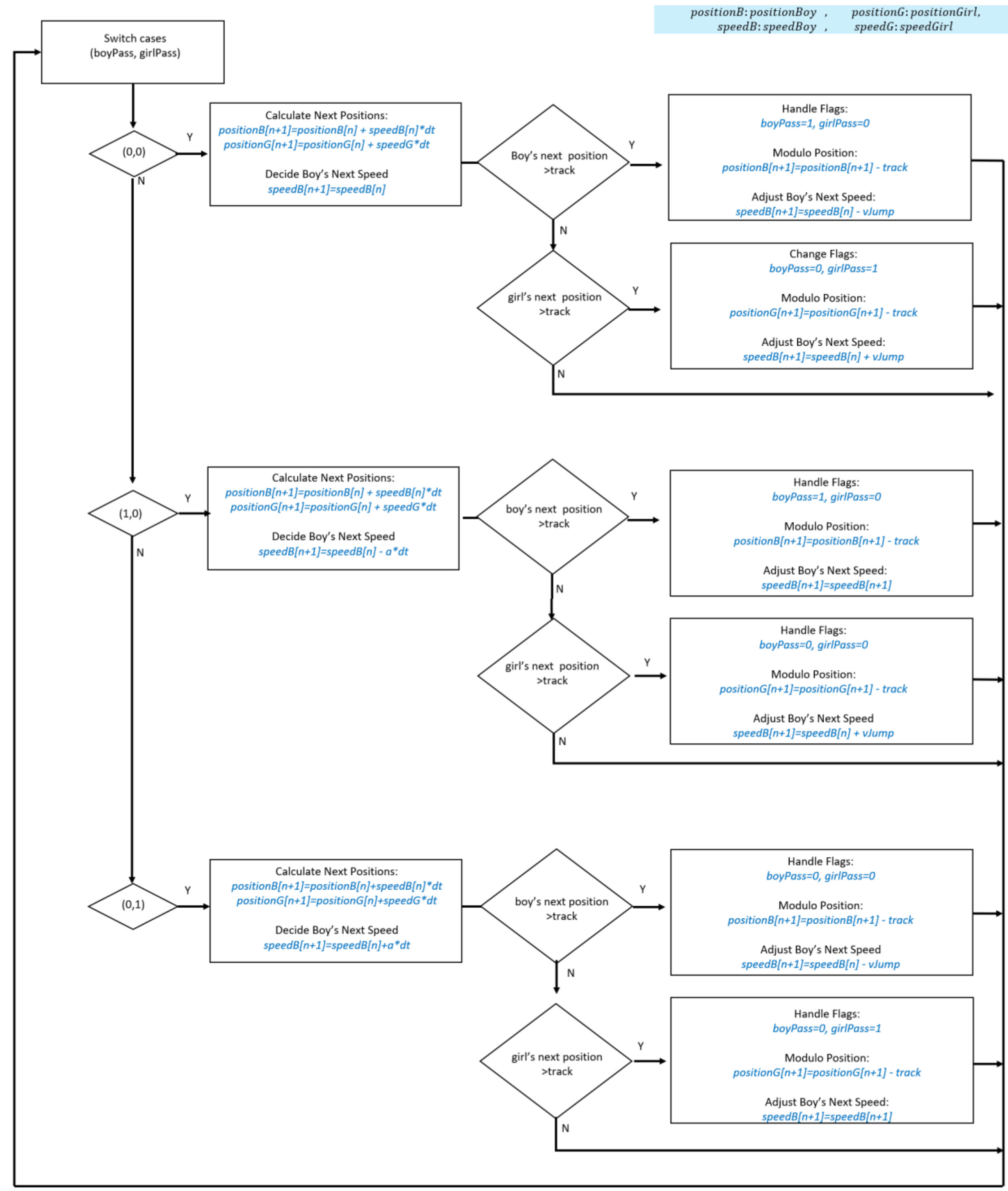
The theory of operation for phase-locked loop became crystal clear after building a model and developing an algorithm which is computationally simulated and mathematically verified in the positive. I feel excited to come up with a design procedure which consolidates my works to be useful for others. An application tool is also made to help with PLL design, providing users with the option to choose among an industry standard, an academic guideline, my proprietary suggestion, or users' own pick. My assertion is ultimately validated by practical experiments. I believe others could crack PLL easily through the sharing of this essay, the simulation code and the application tool.

Lesson learned

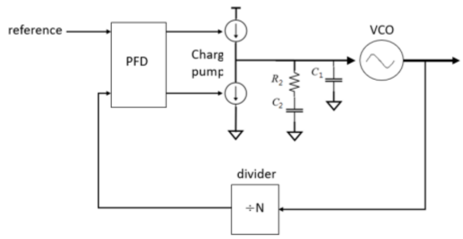
Exploring *PLL* was a long journey full of uncertainties and surprises. I'm glad that I was impulsive enough to dive in and see the spectacular view. Jumping out of a stereotype brings a life definable by myself. I am so grateful for the advices and help from my teachers, many experts in this field and Garmin Corporation. The experience entrenches a world's top company's slogan I've lived by --- "Just do it".

Appendices

Appendix A: *The flowchart of the jogging stimulated program*



Appendix B: *Design guideline from "NXP Application Note: Functional Description of the UMA1018M and UMA1020M Synthesizers"*



Study Notes
by Troy Wu

Loop Filter Design
by NXP UMA1018M Synthesizer Design Note

$$f_n = \frac{2.5}{t_{lock}}$$

(t_{lock} *is the expected time to get phase − locked*)

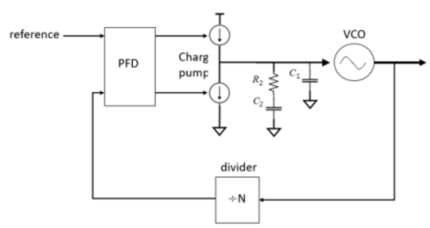
$$\omega_n = 2\pi f_n$$

$$C_2 = \frac{K_{VCO} I_{CP}}{{\omega_n}^2 N}$$

$$R_2 = 2\rho \sqrt{\frac{N}{K_{VCO} I_{CP} C_2}} \quad \text{with } \rho = 0.9 \text{ recommended}$$

$$C_1 = \frac{C_2}{10} \quad \text{where 10 is randomly picked from } (10{\sim}15)$$

Appendix C: *Figure5-2 Design guideline from "RF/Microwave Circuit Design for Wireless Applications by Ulrich L. Rohde and David P. Newkirk" page 872.*



Study Notes
by Troy Wu

Loop Filter Design
by Ulrich L. Rohde and David P. Newkirk

$$\omega_{LPF} = 2\pi \frac{f_{comp}}{ratio} \quad \text{with ratio} = 10{\sim}100 \text{ recommneded}$$

$$\tau_1 = \frac{sec\phi_p - tan\phi_p}{\omega_{LPF}}, \text{with } \phi_p = 45^o \text{recommnded}$$

$$\tau_2 = \frac{1}{{\omega_{LPF}}^2 \tau_1}$$

$$C_1 = \frac{\tau_1}{\tau_1} \frac{I_{CP} K_{VCO}}{{\omega_{LPF}}^2 N} \sqrt{\frac{1 + (\omega_{LPF}\tau_2)^2}{1 + (\omega_{LPF}\tau_1)^2}}$$

$$C_2 = C_1 (\frac{\tau_2}{\tau_1} - 1)$$

$$R_2 = \frac{\tau_2}{C_2}$$

Appendix D: Practical design

Design Parameters:
Target frequency: 1575.42MHz (for GPS)
$K_{VCO} = 35\ MHz/V$
Reference frequency: 341KHz (1.023MHz/3)
$N = \frac{1575.42}{0.341} = 4620$
Charge pump current of PLL IC: NXP LMX2326
$I_{Cp} = 1mA$
Expected time (switching time) to get done phase-locking:
$t_{lock} = 1mS$

Solutions by Ulrich L. Rohde and David P. Newkirk

$$\omega_{LPF} = 2\pi\frac{f_{comp}}{ratio} \quad with\ ratio: 10\ (Rohde's\ pick) \qquad = 2\pi\frac{341e3}{10} = 214000$$

$$\tau_1 = \frac{sec\phi_p - tan\phi_p}{\omega_{LPF}}\ with\ \phi_p = 45^o recommnded = \frac{\frac{2}{\sqrt{2}} - 1}{214000} = 1.93e-6$$

$$\tau_2 = \frac{1}{\omega_{LPF}{}^2\tau_1} = \frac{1}{21426^2 \times 1.933e-5} = 1.13e-3$$

$$C_1 = \frac{\tau_1}{\tau_2}\frac{I_{CP}K_{VCO}}{\omega_{LPF}{}^2 N}\sqrt{\frac{1+(\omega_{LPF}\tau_2)^2}{1+(\omega_{LPF}\tau_1)^2}} = \frac{1.93e-5}{1.13e-3} \times \frac{(1e-3)x35e6}{21400^2 \times 4620}\sqrt{\frac{1+(21400\times1.13e-3)^2}{1+(21400\times1.93e-5)^2}} = 6.84e-11\ \text{(0.068nF)}$$

$$C_2 = C_1\left(\frac{\tau_2}{\tau_1} - 1\right) = 6.84e-9 \times (\frac{1.13e-4}{1.93e-5} - 1)\text{=3.32e-10 (0.33nF)}$$

$$R_2 = \frac{\tau_2}{C_2} = \frac{1.13e-3}{3.32e-10} = 3.40e4 \quad (34k\ ohm)$$

Solutions by NXP Application Notes

$$f_n = \frac{2.5}{t_{lock}} = \frac{2.5}{1e-3} = 2.5e3$$

$$\omega_n = 2\pi f_n = 2\pi \times 2.5e3\text{=15700}$$

$$C_2 = \frac{K_{VCO}I_{CP}}{\omega_n{}^2 N} = \frac{35e6 \times 1e-3}{15700^2 \times 4620} = 3.07e-8 \quad (\ 30.7nF)$$

$$R_2 = 2\rho\sqrt{\frac{N}{K_{VCO}I_{CP}C_2}} \quad with\ \rho = 0.9 = 2 \times 0.9 \times \sqrt{\frac{4620}{35e6 \times 1e-3 \times 3.07e-8}} = 3.73e3 \quad (3.73k\ ohm)$$

$$C_1 = \frac{C_2}{10} = \frac{3.07e-8}{10} = 3.07e-09 \quad (3.07nF)$$

Solutions by Troy's algorithm

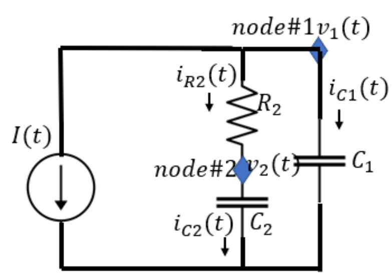
$R_2 = \frac{30N}{I_{CP}K_{VCO}t_{lock}} = \frac{30\times4620}{1e-3\times35e6\times1e-3} = 3960 \qquad (3.96\ Kohm)$

$C_2 = \frac{4N}{I_{CP}K_{VCO}{R_2}^2} = \frac{4\times4620}{1e-3\times35e6\times3960^2} = 3.37e-8 \qquad (33.7nF)$

$C_1 = \frac{C_2}{10} = \frac{3.37e-8}{10} = 3.37e-9 \qquad (3.37nF)$

Appendix E: Time-domain solution of a linear equation set

The major challenge to make my simulation code is to resolve the control voltage of circuitry consisting of current source, C1, R2, and C2. It is not difficult to list the I-V characteristic function of individual components and KCL equations for nodes. After applying the fundamental equation $\frac{dy(t)}{dt} = \lim_{\Delta t\to0}\frac{y(t)-y(t-\Delta t)}{\Delta t}$ $(\equiv \frac{y(t)-y(t-\Delta t)}{\Delta t}, \Delta t \to 0)$ for derivatives and replacing $y(t), y(t-\Delta t)$ with $y[k], y[k-1]$, the differential equation set is converted to be difference equation set.



$$I-V\ char.for\ C_1: \quad i_{C1}(t) = C_1\frac{dv_1(t)}{dt}$$

$$I-V\ char.for\ R_2: \quad i_{R2}(t) = \frac{1}{R_2}[v_1(t)-v_2(t)]$$

$$I-V\ char.for\ C_2: \quad i_{C2}(t) = C_2\frac{dv_2(t)}{dt}$$

$$KCL\ for\ node\#1: \quad i_{C1}(t)+i_{R2}(t) = I(t)$$

$$KCL\ for\ node\#2: \quad -i_{R2}(t)+i_{C2}(t) = 0$$

$$by\ \frac{dy(t)}{dt} = \lim_{\Delta t\to0}\frac{y(t)-y(t-\Delta t)}{\Delta t}\ \ and\ replacing\ y(t), y(t-\Delta t) with\ y[k], y[k-1]$$

$$I-V\ char.for\ C_1: \quad i_{C1}[k] = C_1\frac{v_1[k]-v_1[k-1]}{\Delta t}$$

$$I-V\ char.for\ R_2: \quad i_{R2}[k] = \frac{1}{R_2}(v_1[k]-v_2[k])$$

$$I-V\ char.for\ C_2: \quad i_{C2}[k] = C_2\frac{v_2[k]-v_2[k-1]}{\Delta t}$$

$$KCL\ for\ node\#1: \quad i_{C1}[k]+i_{R2}[k] = I[k]$$

$$KCL\ for\ node\#2: \quad -i_{R2}[k]+i_{C2}[k] = 0$$

Reorganizing the difference equation set into matrix form and applying the inverse matrix, I can iteratively calculate to learn variables[k] from variables[k-1] and I[k].

$$
\begin{aligned}
i_{C1}[k] &= C_1 \frac{v_1[k] - v_1[k-1]}{\Delta t} \\
i_{R2}[k] &= \frac{1}{R_2}(v_1[k] - v_2[k]) \\
i_{C2}[k] &= C_2 \frac{v_2[k] - v_2[k-1]}{\Delta t} \\
i_{C1}[k] &+ i_{R2}[k] = I[k] \\
-i_{R2}[k] &+ i_{C2}[k] = 0
\end{aligned}
$$

$$
\begin{aligned}
& i_{C1}[k] - \frac{C_1}{\Delta t} v_1[k] = -\frac{C_1}{\Delta t} v_1[k-1] \\
& i_{R2}[k] - \frac{1}{R_2} v_1[k] + \frac{1}{R_2} v_2[k] = 0 \\
& i_{C2}[k] - \frac{C_2}{\Delta t} v_2[k] = -\frac{C_2}{\Delta t} v_2[k-1] \\
& i_{C1}[k] + i_{R2}[k] = I[k] \\
& -i_{R2}[k] + i_{C2}[k] = 0
\end{aligned}
$$

$$
\begin{bmatrix} 1 & 0 & 0 & -\frac{C_1}{\Delta t} & 0 \\ 0 & 1 & 0 & -\frac{1}{R_2} & \frac{1}{R_2} \\ 0 & 0 & 1 & 0 & -\frac{C_2}{\Delta t} \\ 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \end{bmatrix}
\begin{bmatrix} i_{C_1}[k] \\ i_{R_2}[k] \\ i_{C_2}[k] \\ v_1[k] \\ v_2[k] \end{bmatrix}
=
\begin{bmatrix} -\frac{C_1}{\Delta t} v_1[k-1] \\ 0 \\ -\frac{C_2}{\Delta t} v_2[k-1 \\ +I[k] \\ 0 \end{bmatrix}
$$

$$
\begin{bmatrix} i_{C_1}[k] \\ i_{R_2}[k] \\ i_{C_2}[k] \\ v_1[k] \\ v_2[k] \end{bmatrix}
=
\begin{bmatrix} 1 & 0 & 0 & -\frac{C_1}{\Delta t} & 0 \\ 0 & 1 & 0 & -\frac{1}{R_2} & \frac{1}{R_2} \\ 0 & 0 & 1 & 0 & -\frac{C_2}{\Delta t} \\ 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \end{bmatrix}^{-1}
\begin{bmatrix} -\frac{C_1}{\Delta t} v_1[k-1] \\ 0 \\ -\frac{C_2}{\Delta t} v_2[k-1 \\ +I[k] \\ 0 \end{bmatrix}
$$

Bibliography

187N. Intro. to phase-locked loops (PLL) noise. (2019, July 18). [Video]. YouTube. https://www.youtube.com/watch?v=aVd7vAeJYzk

19. Phase-locked Loops. (2013, July 16). [Video]. YouTube. https://www.youtube.com/watch?v=PsUPRyatjxw

Analog Devices. (2009). Fundamentals of Phase Locked Loops (PLLs). https://www.analog.com/media/en/training-seminars/tutorials/MT-086.pdf

Fischette, D. (2007, May). First Time, Every Time– Practical Tips for Phase Practical Tips for Phase-Locked Loop Design (No. 120). Delroy. https://www.scribbr.co.uk/apa-reference-generator/new/report/

Extended Essay PLL Experiment 1: Measure the control voltage of PLL. (2020, September 21). [Video]. YouTube. https://www.youtube.com/watch?v=K38vr60DYLo

Measure the synchronization between PLL #1 and PLL #2. (2020, September 21). [Video]. YouTube. https://www.youtube.com/watch?v=ecPnGil_uP8

Rohde, U. L., & Rudolph, M. (2012). RF / Microwave Circuit Design for Wireless Applications (2nd ed.). Wiley.

Wu, T. (2020). *Troy Wu – 18-year-old enthusiastic Visionary*. Troy Wu.

https://troywuofficial.com/