

Sincronización en fase y frecuencia: conceptos principales

Pablo Belzarena

Comunicaciones inalámbricas

2023

Tabla de contenidos

- 1 Introducción
- 2 Phase Locked Loop : PLL
- 3 PLL en tiempo discreto
- 4 Costas Loop
- 5 Ambigüedad de fase : Codificación diferencial

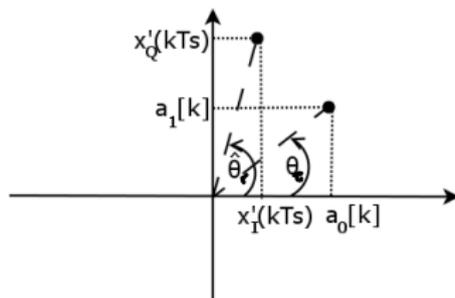
Sincronización en fase y frecuencia

Introducción

- La fase del oscilador en el receptor y el transmisor no es la misma. El canal de comunicación inalámbrico introduce también un diferencia de fase.
- Los osciladores del transmisor y receptor no pueden ser sintonizados exactamente en la misma frecuencia.

$$r(t) = s(t)e^{j\omega_0 t} \quad (1)$$

$$x(t) = s(t)e^{j\omega_0 t} e^{-j\omega_1 t + \Phi_0} = s(t)e^{j(\omega_0 - \omega_1)t + \Phi_0} \quad (2)$$



Sincronización en fase y frecuencia

Introducción

- Analizaremos como corregir el error en fase y errores en frecuencia que varíen lentamente.
- El problema entonces es el siguiente. Tengo una portadora, con una fase constante o variable que debo estimar.
- ¿Cómo lo hago?

PLL

Introducción

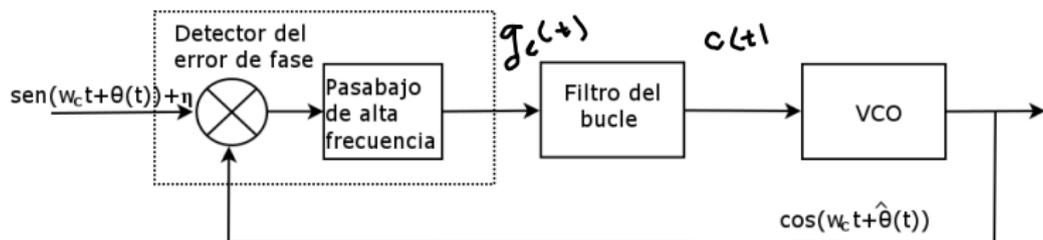


Figure – Esquema general de un PLL

La señal que ingresa : $r(t) = \sin(w_c t + \theta(t)) + \eta(t)$. Un oscilador controlado por voltaje VCO, generará una señal de la forma : $u(t) = \cos(w_c t + \hat{\theta}(t))$.

$$\dot{\hat{\theta}}(t) = k_1 c(t)$$

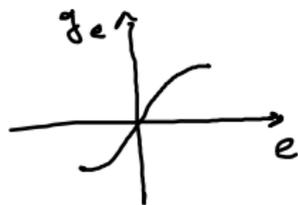
En este caso $\theta(t)$ es la fase que se desea seguir y $\hat{\theta}(t)$ es la estimación.

PLL

PLL no lineal

La señal de error en general en los PLL es una función no lineal del error :

$$g_e(t) = k_p \sin(\phi_e(t)) \quad (3)$$



La salida del filtro de bucle :

- PLL de primer orden : $c(t) = k_0 g_e(t)$
- PLL de segundo orden : $c(t) = k_0 (g_e(t) + a \int_0^t g_e(s) ds)$

El VCO genera un coseno de frecuencia w_c y la fase verifica la siguiente ecuación :

$$\dot{\phi}(t) = k_1 c(t) \quad (4)$$

siendo k_1 la constante de proporcionalidad.

Se asume que :

$$\phi(t) = \Delta w_c t + \phi_0 \quad (5)$$

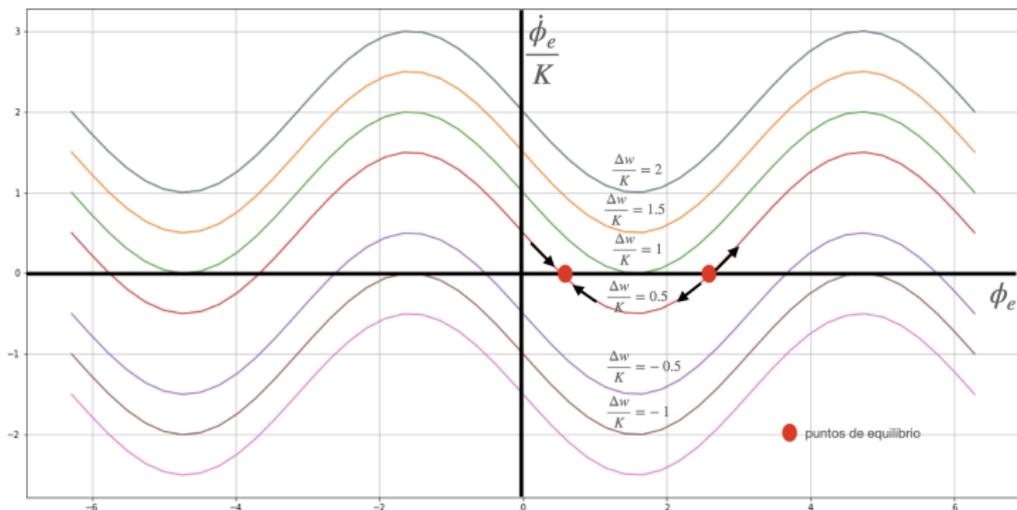
PLL

PLL no lineal de primer orden

Para el caso de primer orden operando se obtiene :

$$\dot{\phi}_e(t) = \Delta\omega_c - k_1 k_0 k_p \sin(\phi_e(t))$$

Plano de fase : Se grafica $\dot{\phi}_e(t)$ como función de $\phi_e(t)$.



PLL

PLL no lineal de primer orden

Los puntos de equilibrio ϕ_e^* son aquellos donde $\dot{\phi}_e = 0$

$$\dot{\phi}_e/K = \Delta w_c/K - \sin(\phi_e^*) = 0 \quad (6)$$

Para que existan puntos de equilibrio y logremos que el PLL no oscile y vaya a un punto de equilibrio es necesario que $|\Delta w_c/K| < 1$.

$$\sin(\phi_e^*) = \Delta w_c/K \quad (7)$$

Cuando el PLL de primer orden tiene un punto de equilibrio, siempre tiene un error constante $\phi_e^* \neq 0$.

Varios puntos de equilibrio. Estables o inestables?

PLL no lineal de segundo orden. Ver las notas.

PLL

Modelo lineal

Modelo no-lineal en casos simples se puede analizar, pero más en general es necesario linealizar : $g_e(t) = k_p \sin(\phi_e(t)) \approx k_p \phi_e(t)$

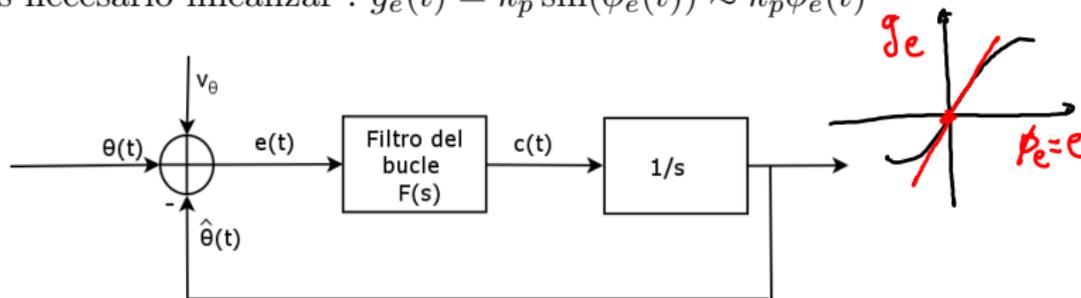


Figure – Esquema general de un PLL linealizado

$$\text{VCO} : \dot{\hat{\phi}}(t) = k_1 c(t), \quad \hat{\Phi}(s) = k_1 C(s) \frac{1}{s}$$

$$\text{Controlador} : c(t) = k_0 (k_p e(t) + a \int_0^t k_p e(s) ds),$$

$$C(s) = E(s) k_p (k_0 + \frac{k_0 a}{s})$$

PLL

Modelo lineal segundo orden

$$H(s) = \frac{\hat{\Theta}(s)}{\Theta(s)}$$

$$H(s) = k_T \frac{s + a}{k_T a + k_T s + s^2}$$

Se define frecuencia natural w_n y factor de amortiguamiento ζ :

$$w_n = \sqrt{k_T a} , \zeta = \frac{1}{2} \sqrt{\frac{k_T}{a}}$$

$$H(s) = \frac{2\zeta w_n s + w_n^2}{w_n^2 + 2\zeta w_n s + s^2}$$

PLL

Modelo lineal en tiempo continuo de segundo orden

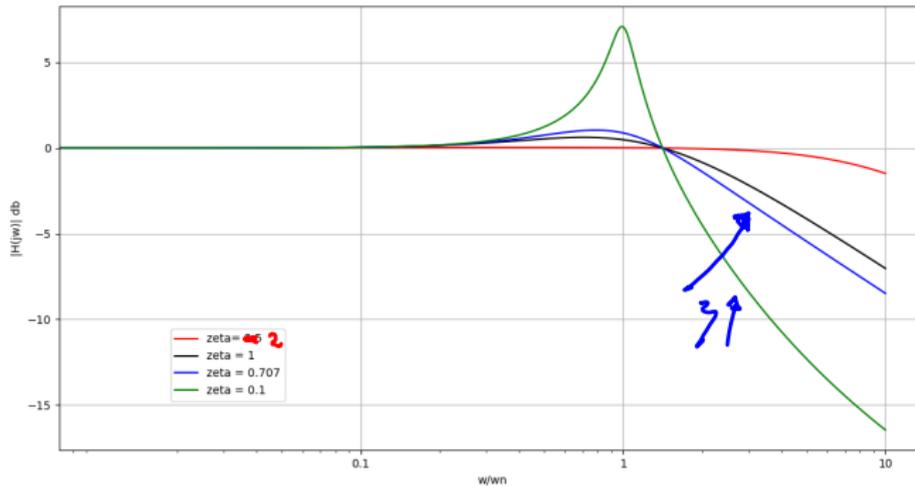
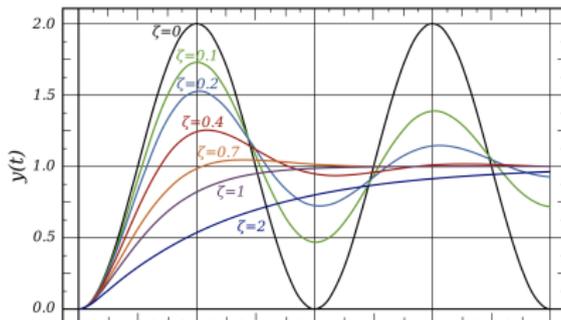


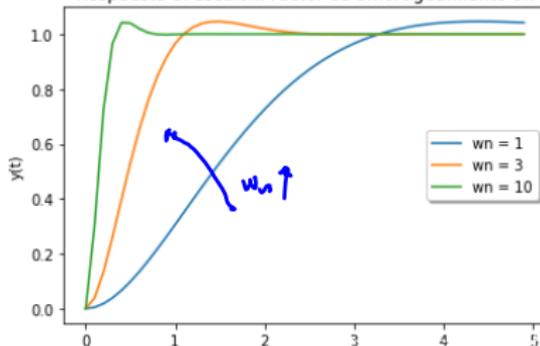
Figure – Respuesta en frecuencia de la transferencia $H(s)$

PLL

Modelo lineal en tiempo continuo de segundo orden



Respuesta al escalón. Factor de amortiguamiento 0.7



PLL

Modelo lineal en tiempo continuo de segundo orden

$$B_N = \frac{w_n}{2} \left(\zeta + \frac{1}{4\zeta} \right) \quad (11)$$

En la figura se muestra el gráfico de B_N/w_n como función de ζ . se puede observar que B_N es minimizado cuando $\zeta = 0.5$ y en ese caso $B_N = 0.5w_n$.

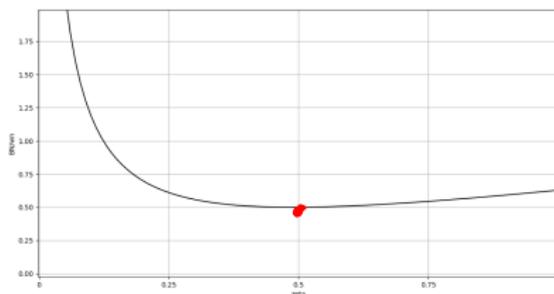


Figure – Ancho de banda equivalente de ruido en función de ζ

En aplicaciones prácticas se elige $\zeta = 0.707$. En este caso $B_N = 0.53w_n$ que es marginalmente superior al óptimo.

Sincronización en fase y frecuencia

PLL en tiempo discreto

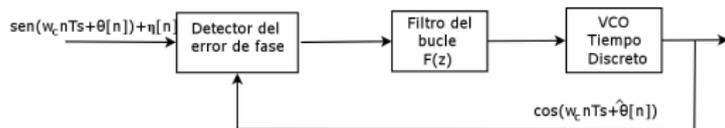


Figure – Esquema general de PLL en tiempo discreto

El detector de error de fase : $g(e[n])$, $e[n] = \theta[n] - \hat{\theta}[n]$. VCO también denominado DDS en tiempo discreto. La señal de entrada al DSS $c[n]$ y la fase del DSS $\hat{\theta}[n]$ estará dada por la siguiente ecuación :

$$\hat{\theta}[n + 1] = \hat{\theta}[n] + k_0 c[n] , c[n] = \frac{\hat{\theta}[n + 1] - \hat{\theta}[n]}{k_0}$$

donde k_0 es el paso de actualización. $c[n]$ puede verse como una derivada en tiempo discreto de $\hat{\theta}[n]$.

Sincronización en fase y frecuencia

Costas Loop

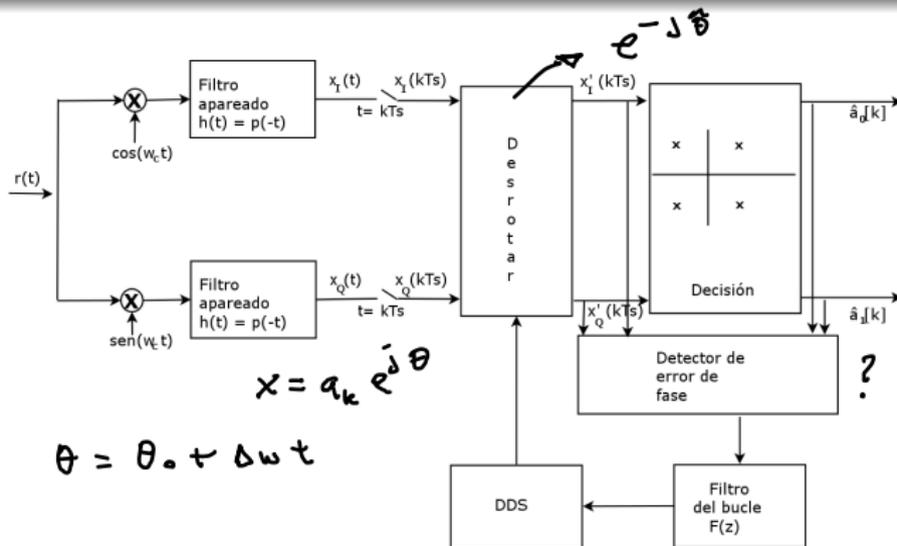


Figure – Esquema general de Costas Loop

Sincronización en fase y frecuencia

Costas Loop

$$e[k] = \theta_r - \theta_t = \arctan \frac{x'_Q[k]}{x'_I[k]} - \arctan \frac{a_1[k]}{a_0[k]} \quad (12)$$

θ_r y θ_t son los ángulos en la recepción y transmisión.

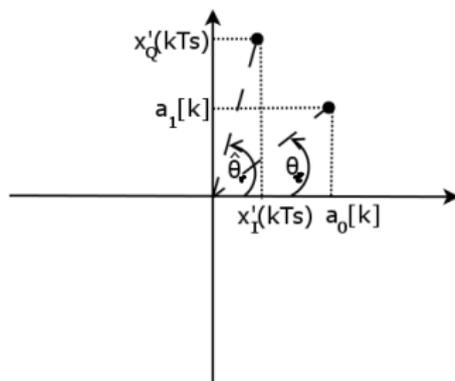


Figure – Fase transmitida y recibida

Sincronización en fase y frecuencia

Costas Loop

$$e[k] = \arctan \frac{x'_Q[k]}{x'_I[k]} - \arctan \frac{\hat{a}_1[k]}{\hat{a}_0[k]} \quad (13)$$

Costas utiliza una aproximación a este error por la siguiente función de error :

$$g(e[k]) = x'_Q(kT_s)a_0[k] - x'_I(kT_s)a_1[k] \quad (14)$$

y si no se cuenta con los símbolos enviados :

$$g(e[k]) = x'_Q(kT_s)\hat{a}_0[k] - x'_I(kT_s)\hat{a}_1[k] \quad (15)$$

Sincronización en fase y frecuencia

Costas Loop

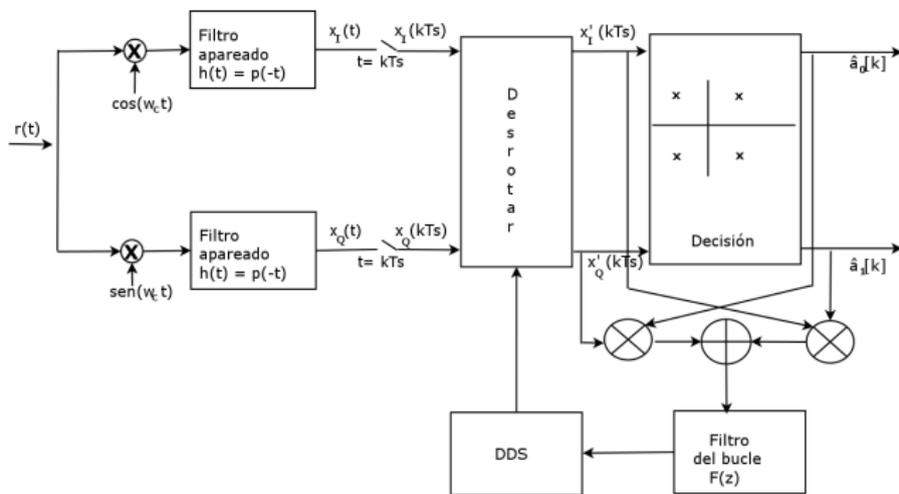
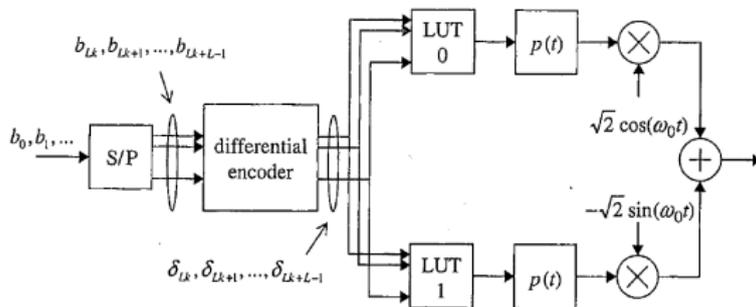


Figure – Costas Loop para QPSK

Sincronización en fase y frecuencia

Ambigüedad en la fase

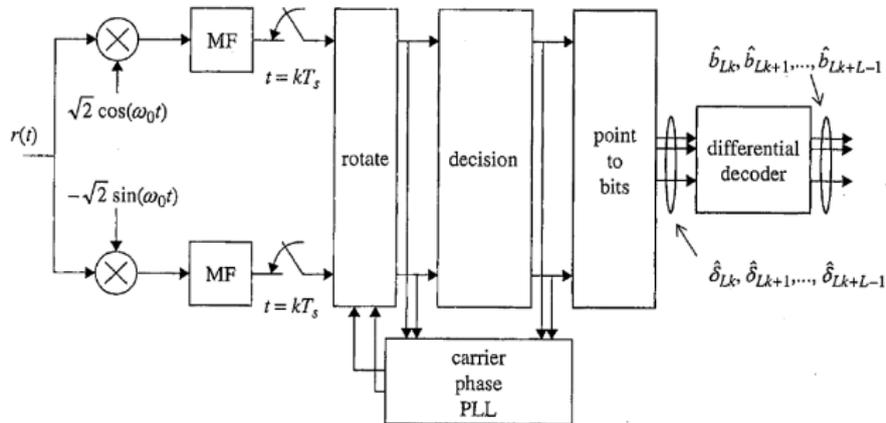
1. Codificación Diferencial



Sincronización en fase y frecuencia

Ambigüedad en la fase

Decodificador Diferencial



Sincronización en fase y frecuencia

Decodificador Diferencial. Ej BPSK

b_k codifico con dos bits δ_{k-1}, δ_k tal que haya o no cambio de fase siempre $\hat{b}_k = b_k$.

Una forma de hacer esto es por ejemplo codificar :

$$\delta_k = \delta_{k-1} \oplus b_k$$

y decodificar

$$b_k = \delta_{k-1} \oplus \delta_k$$

Sincronización en fase y frecuencia

Decodificador Diferencial. Ej BPSK

Ej. Si $b_k = 0$

- Si $\delta_{k-1} = 0$ se codificará $\delta_k = 0$
- Si $\delta_{k-1} = 1$ se codificará $\delta_k = 1$.

Si por ejemplo $\delta_{k-1} = 0, \delta_k = 0$.

- Si no hay error de fase se recibirá $\hat{\delta}_{k-1} = 0, \hat{\delta}_k = 0$
- Si hay error de fase π entonces $\hat{\delta}_{k-1} = 1, \hat{\delta}_k = 1$

Pero en ambos casos

$$\hat{b}_k = \delta_{k-1} \oplus \delta_k = 0 = b_k$$