

# PROTECCIONES NUMÉRICAS



A-II.

Curso: CAPSEP

Conceptos Avanzados sobre Protección de  
Sistemas Eléctricos de Potencia,  
IIE-FING-UdelaR 2019.

# **PROTECCIONES NUMÉRICAS**

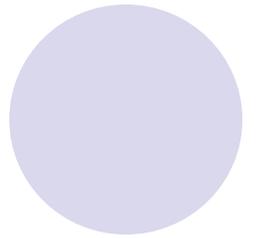
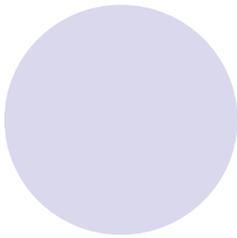
## **BIBLIOGRAFÍA**

- **Understanding microprocessor-based technology applied to relaying.** Power System Relaying Committee. Report of Working Group I-01 of the Relaying Practices Subcommittee (IEEE) January 2009
- **Procesamiento Digital de la Información para Sistemas Eléctricos de Potencia.** Jorge L. Alonso. IIE-FING-UdelaR.
- **Computer Relaying for Power Systems.** Arun G. Phadke, James S. Thorp.
- **Manuales, papers y cursos** de fabricantes de protecciones.
- **Fundamentos de la medición de sincrofasores.** Roberto Cimadevilla, ZIV P+C (XIII ERIAC, CIGRE)
- **Advancements in Microprocessor Based Protection and Communication.** IEEE Tutorial Course 97TP120-0.
- **Network Protection & Automation Guide** (NPAG), ALSTOM (2002 y 2011).
- **Protective Relaying Theory and Applications**, W. A. Elmore.
- **Power System Relaying.** Stanley H. Horowitz, Arun G. Phadke.
- **Wikipedia** <http://en.wikipedia.org/> <http://es.wikipedia.org/>

# PROTECCIONES NUMÉRICAS

Relés microprocesados que mediante

- el muestreo de las magnitudes eléctricas medidas varias veces por ciclo,
- la retención de dichos valores instantáneos (sample & hold o S/H)
- y su posterior conversión analógica digital (A/D),  
desarrollan sofisticados algoritmos de protección mediante técnicas numéricas; por ej.:
  - calcular fasores de las magnitudes medidas
  - y a partir de ellos desarrollar algoritmos de protección.
- Son un tipo de IED.
- Las funciones de protección numérica se desarrollan en IEDs.



# CARACTERÍSTICAS DE LOS RELÉS NUMÉRICOS

- Flexibilidad
- Integración
  - Varias funciones de protección – relés multifunción
    - funciones principales y de respaldo (e incluso de monitoreo)
    - amplios rangos de ajuste
  - Múltiples grupos de ajustes (seleccionables o conmutables por hardware y software)
  - Lógicas y configuraciones lógicas definidas por el usuario (de protección, control, mando)
  - Las magnitudes se miden una vez y las utilizan varias funciones de protección
  - Funciones de mando y control (protección y control integrado, *protection and automation*, muy difundido en MT),
  - No son necesarios TIs auxiliares,
  - Disminución de espacio en los paneles

# CARACTERÍSTICAS DE LOS RELÉS NUMÉRICOS

- Protección adaptiva y lógicas adaptivas
- Señalización y alarmas visuales (propias y salidas a otros equipos por protocolos de comunicación y/o cableado)
- Interfase humano-máquina en el relé (HMI) (display y teclado)
- Comunicación para su gestión (local y a distancia) desde PC,
- Localización de faltas (protección de líneas y cables)
- Sincronización horaria (GPS, IRIG-B o IEEE 1588)
- Registro oscilográfico de faltas y condiciones anormales.
- Registro Cronológico de Eventos o RCE (*Sequence Of Events* o SOE, *Sequence Events Report* o SER)
  - funciones de protección (arranques, disparos, bloqueos, etc.)
  - falla o defecto interno
  - otros eventos (cambio de ajustes, pérdida de referencia temporal, reinicio, etc.)

# CARACTERÍSTICAS DE LOS RELÉS NUMÉRICOS

- Medida de magnitudes de servicio on-line (local y remoto)
- Baja carga a los circuitos de medida de corriente y tensión.
- Las magnitudes residuales pueden calcularse en vez de medirse, o además de medirse
- Chequeos
  - Supervisión de coherencia de  $I_s$  y  $V_s$
  - Detección de saturación de TIs.
  - Supervisión de circuitos disparo,
  - Supervisión de posición 52 y 89,
  - Supervisión del estado de interruptores (cantidad de aperturas,  $I^2t$  acumulado, etc.)
  - Auto-supervisión muy exhaustiva y alarma de falla interna,
  - Supervisión de la calidad de la continua de alimentación.

# CARACTERÍSTICAS DE LOS RELÉS NUMÉRICOS

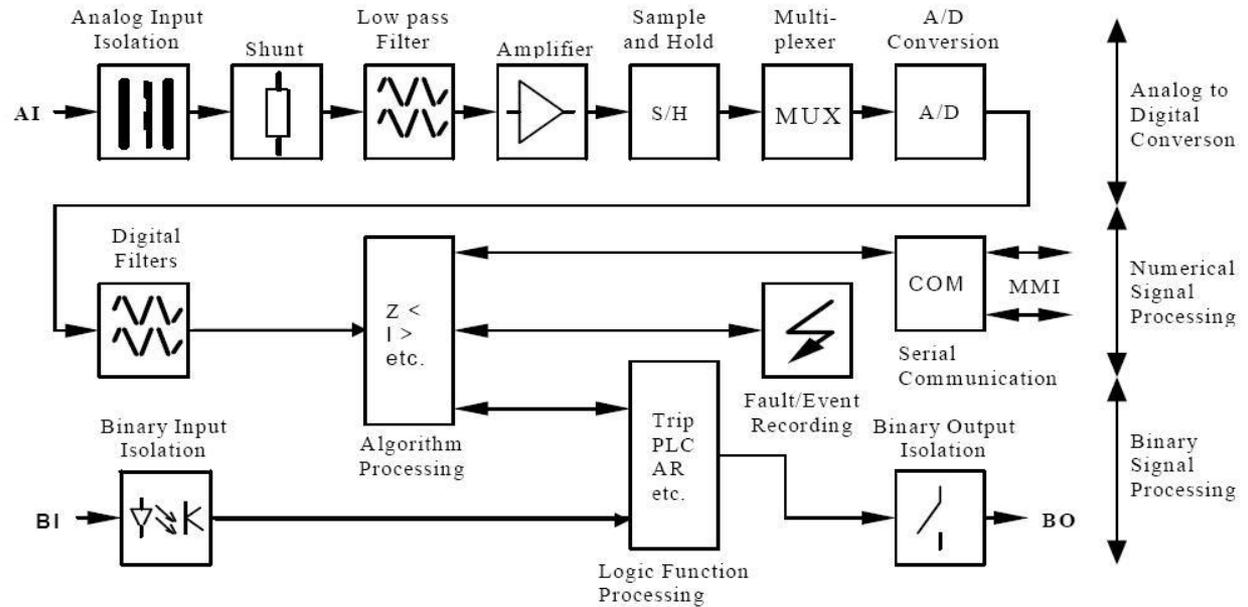
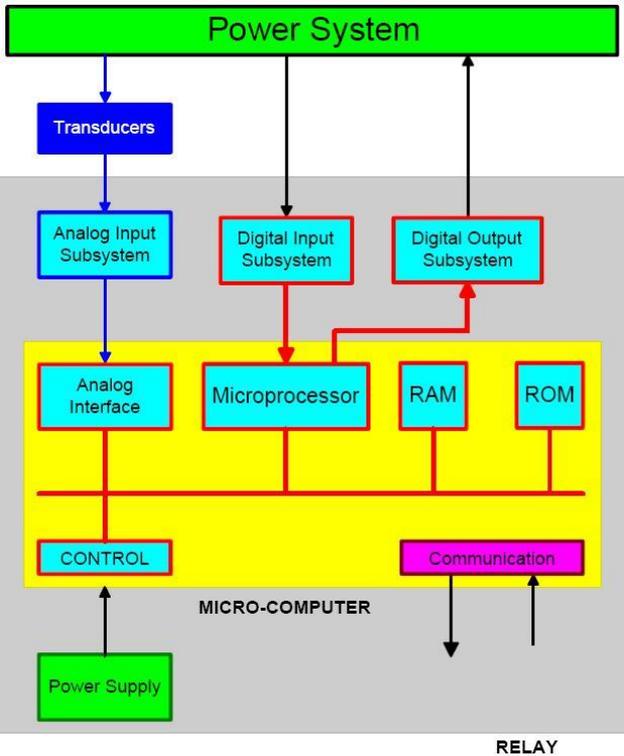
- Funciones de ensayo
  - forzado de salidas,
  - forzado de señales internas,
  - verificación y forzado de entradas digitales,
  - debug de lógicas,
  - modo ensayo para habilitar y deshabilitar funciones,
  - etc.
- Posibilidad de actualizar el firmware

# CARACTERÍSTICAS DE LOS RELÉS NUMÉRICOS

Dificultades o inconvenientes:

- Enorme cantidad de datos (¿información?)
- Gran complejidad (para entender funcionamiento y relaciones, implementar ajustes, implementar lógicas, lógicas y funciones adaptivas, IEC61850, ensayo, etc.)
- Vida útil limitada.
  - Depende de la obsolescencia tecnológica (nuevas necesidades, funcionalidades o compatibilidades no cubiertas)
  - La vida de los componentes limita la confiabilidad y disponibilidad.
- La integración de funciones, lógicas, protección y control integrados, aumenta la susceptibilidad a las fallas de modo común y afecta la confiabilidad.
- Susceptibilidad a interferencia de radio (RFI) y electromagnética (EMI). El rechazo es muy bueno, pero menor que algunas tecnologías previas.

# BLOQUES FUNCIONALES PRINCIPALES



# ALIMENTACIÓN. E/S.

- Fuente de alimentación
  - Duplicada ¿?
  - Alarma falla DC
  - Pila
  - De rango amplio de trabajo, o no
- Entradas digitales (umbrales seleccionables)
- Salidas digitales
  - Contactos
  - SCR (Silicon Controlled Rectifier), SSR (Solid State Relay)
- E/S por protocolos de comunicación (IEC 61850)

# ENTRADAS ANALÓGICAS

- Aíslan al relé de los circuitos secundarios de medida,
- Reducen el nivel de las tensiones,
- Convierten las corrientes en tensiones proporcionales,
- Eliminan las componentes de alta frecuencia. Filtros analógicos pasabajo (antialiasing o no).

# PREFILTRADO Y MUESTREO

- Muestreo es el proceso de conversión de una señal continua en el tiempo, en una señal discreta en el tiempo.
- Teorema de muestreo, de Shannon o de Nyquist  $f_s > 2 f_{MÁX}$  debe cumplirse para que no haya aliasing.

$f_{MÁX}$  es la del armónico más alto de interés en la aplicación (la fundamental, el 5° arm., etc.)

# PREFILTRADO Y MUESTREO

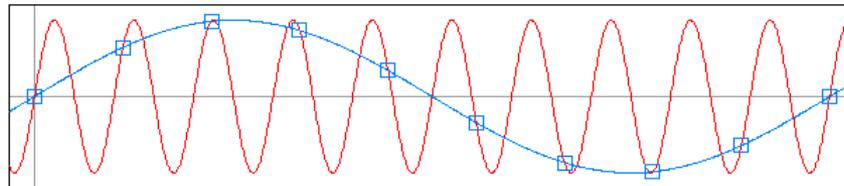
## Aliasing (alias)

- durante el procesamiento digital (muestreo periódico a cierta frecuencia  $f_s$ )
- una señal sinusoidal (analógica periódica) de una cierta frecuencia  $f$ ,
- puede llegar a parecer una señal de menor frecuencia si:
  - no se la filtra previa y adecuadamente
  - no se cumple el teorema del muestreo

# PREFILTRADO Y MUESTREO

## Aliasing (alias)

- Ambas señales de frecuencias  $f$  y  $(f - f_s)$ 
  - tienen las mismas muestras,
  - pero una señal es real (original) y la otra no
- Una senoide es un “alias” de la otra y viceversa



$$f_{ROJO} = 500Hz$$

$$f_s = 450 \text{ mps}$$

$$f_{AZUL} = f_{ROJO} - f_s = 50Hz$$

$$f_s \leq 2f$$

No se cumple el teorema de Nyquist

# PREFILTRADO Y MUESTREO

## Aliasing (alias)

### Solución

1. Elegir frecuencia de muestreo suficientemente alta para que se cumpla

$$f_s > 2 f_{MÁX}$$

2. Aplicar a la señal (antes de digitalizarla) un filtro analógico pasabajo (antialiasing) que
  - deje pasar las frecuencias por debajo de la mitad de la frecuencia de muestreo y
  - le quite energía a las señales de alta frecuencia.

# PREFILTRADO Y MUESTREO

## Aliasing (alias)

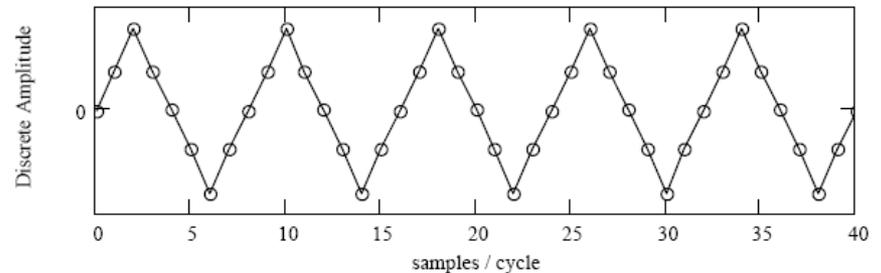
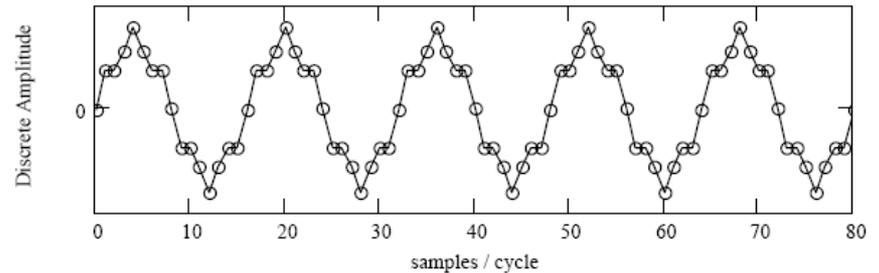
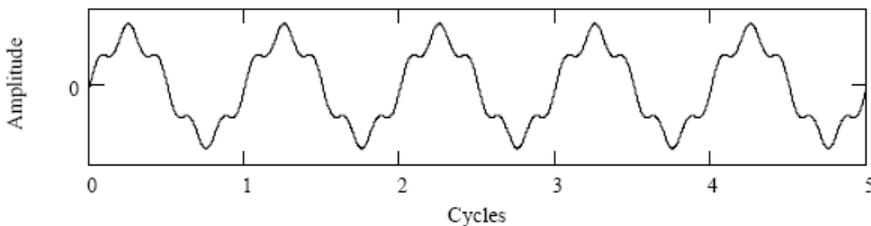
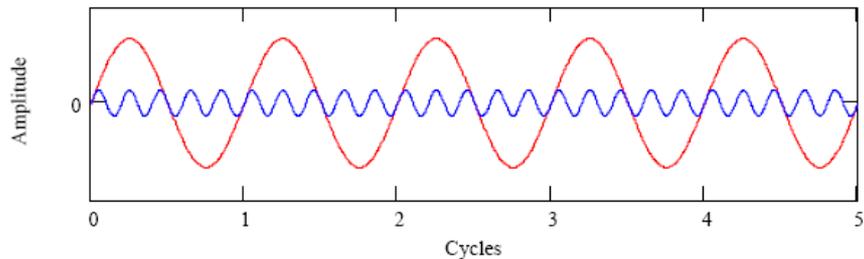
- Los filtros antialiasing pasivos o activos.
  - Por ej. doble filtro RC, atenuación de 12 dB /octava

$f_s$

- primeros relés numéricos
  - 4 muestras/ciclo de la fundamental  $200 \text{ mps} > 2 * 50 \text{ Hz}$
- en la actualidad
  - 8, 16, 20, 32 muestras/ciclo de la fundamental
  - pueden llegar a 96 muestras/ciclo de la fundamental
- Ejemplo de la filmación de las ruedas de un auto
  - [https://www.youtube.com/watch?v=9l\\_C2aahVUk](https://www.youtube.com/watch?v=9l_C2aahVUk)
  - <http://cienciotec.blogspot.com.uy/2013/04/efecto-estroboscopico-y-aliasing.html>

# PREFILTRADO Y MUESTREO

## Aliasing (alias)



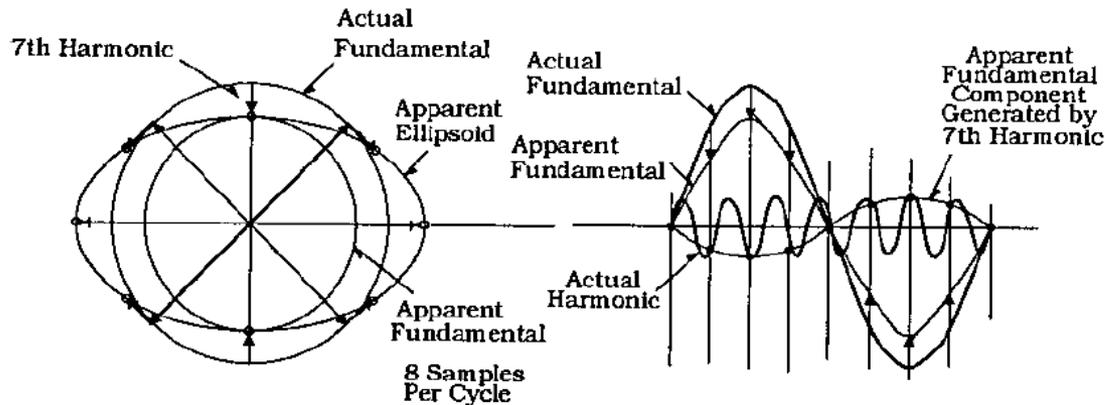
- Señal compuesta por la fundamental (50 Hz) y un 20% de 5<sup>o</sup> armónico (250 Hz)
- 16 muestras por período (800 mps). Cumple teorema del muestreo (800 mps > 500 Hz)
- 8 muestras por período (400 mps). NO cumple teorema del muestreo (400 mps < 500 Hz)

Este efecto del muestreo a 400 mps es indeseado por dos razones:

- La información del 5<sup>o</sup> armónico se perdió.
- La energía de la componente de 5<sup>o</sup> armónico corrompió la interpretación de la onda.

# PREFILTRADO Y MUESTREO

## Aliasing (alias)



Armónico 7 en una señal muestreada 8 veces por ciclo de la fundamental

$$f_s = 400 \text{ mps} \quad f_{MÁX} = 50 \times 7 = 350 \text{ Hz} \quad \text{NO se cumple } f_s > 2 f_{MÁX}$$

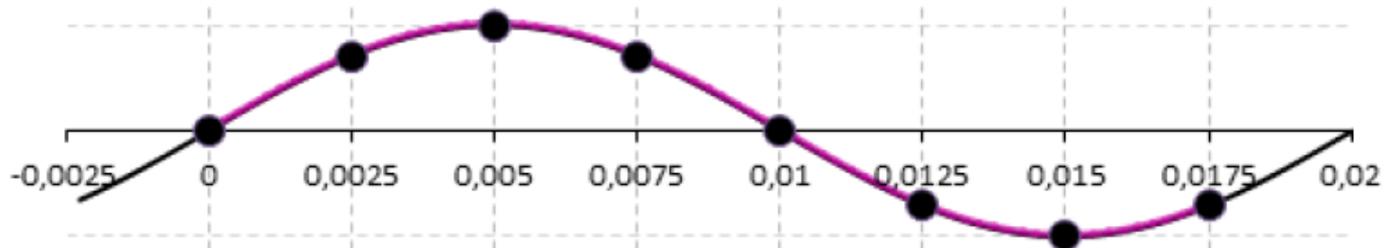
$$(f - f_s) = 400 - 350 = 50 \text{ Hz}$$

El relé ve un fasor de frecuencia fundamental pero de menor amplitud  
(*Apparent Fundamental*)

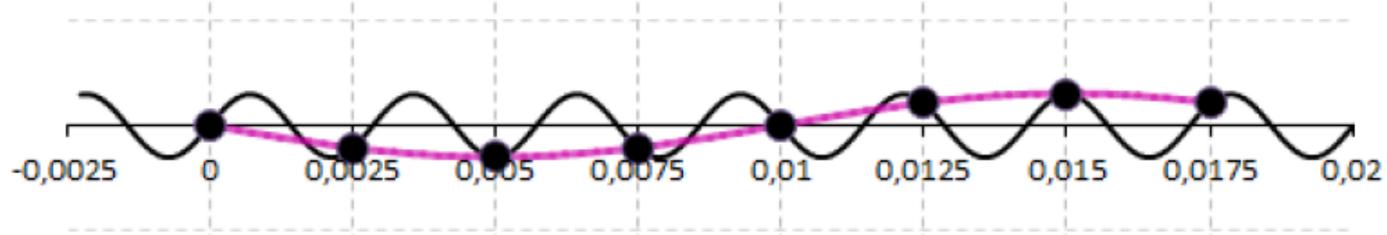
# PREFILTRADO Y MUESTREO

## Aliasing (alias)

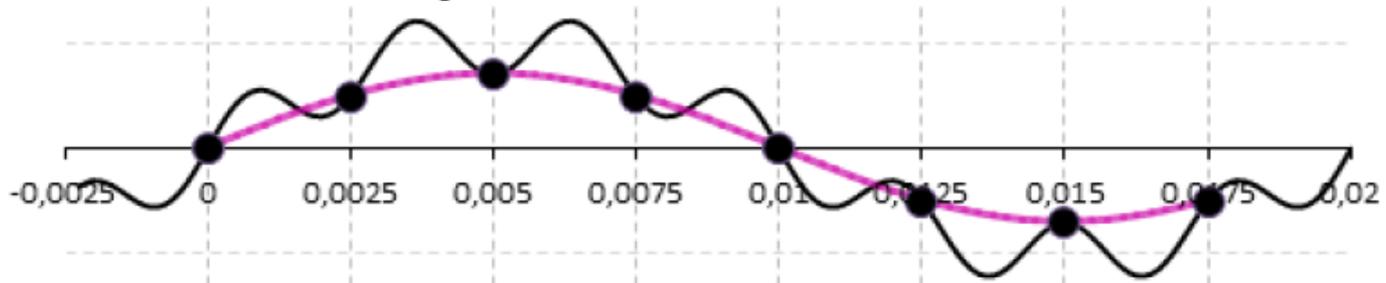
- Señal sólo fundamental (*Actual Fundamental*) y su muestreo (reconstrucción) que coincide con la señal.



- Señal sólo 7º armónico y su muestreo (reconstruye el alias = frecuencia fundamental).



- Señal con fundamental y 7º armónico. Su muestreo (*Apparent Fundamental*) reconstruye la suma de la fundamental original más el alias de frec. fundamental.



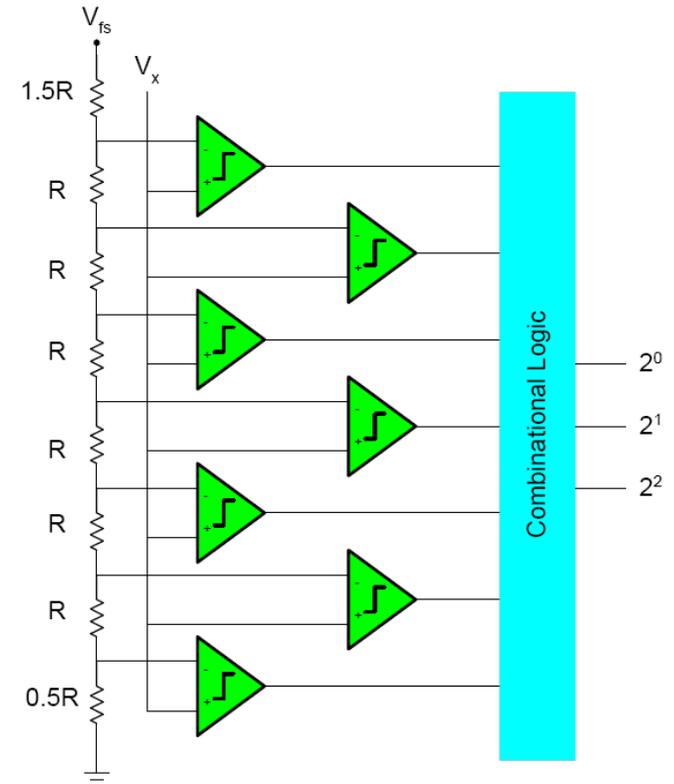
# CONVERSIÓN ANALÓGICA A DIGITAL (A/D o ADC)

## Conversor A/D de n+1 bits

Salidas  $x_i = 0$  o  $1$

$$x_n 2^n + x_{n-1} 2^{n-1} + x_{n-2} 2^{n-2} + \dots + x_1 2^1 + x_0 2^0$$

La “palabra binaria” de salida  
corresponde al  
valor instantáneo muestreado



Ej. de conversor flash A/D de 3 bits

# CONVERSIÓN ANALÓGICA A DIGITAL (A/D o ADC)

## Ejemplo. Conversor A/D de 8 bits, entrada +- 5V

- La salida tiene  $2^8 = 256$  valores discretos (0 a 255)
  - -5V corresponde a 00000000 binario
  - 0V a 10000000 = 127 decimal
  - +5V a 11111111 = 255
- Resolución. LSB

$$\frac{1}{255} \times 10V = 39mV \qquad \frac{1}{255/2} \times 100 = 0,78\%$$

## Ejemplo. Conversor A/D de 16 bits, entrada +- 5V

- $2^{16} = 65536$
- Resolución. LSB  $0,15mV$   $0,0031\%$

# CONVERSIÓN ANALÓGICA A DIGITAL (A/D o ADC)

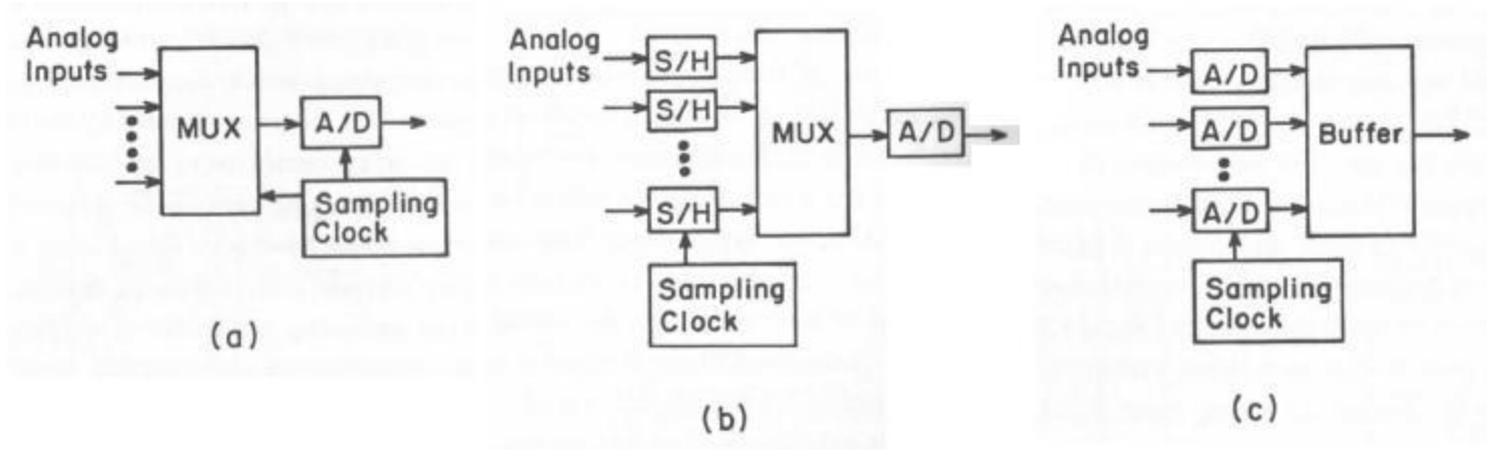
En el período entre la muestra  $i$  e  $i+1$  es  $\frac{1}{f_s}$

El(los) microprocesador(es) debe(n) procesar completamente:

- las muestras de todas las señales de corrientes y tensiones medidas y
- los algoritmos de protección

Esta es una de las causas para limitar la frecuencia de muestreo.

# CONVERSIÓN ANALÓGICA A DIGITAL (A/D o ADC)

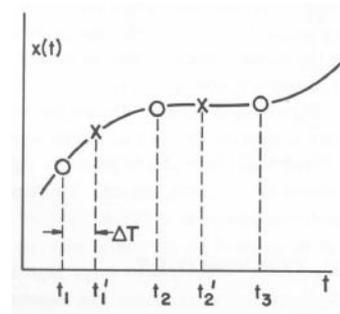


(a)

- $\sum_{k=1}^N \Delta T^k < t_{i+1} - t_i$

sincronizar las muestras por intrapolación.

- $\sum_{k=1}^N \Delta T^k \ll t_{i+1} - t_i$  no es necesario sincronizar.



(b) Se agrega S/H a cada entrada.

Se elimina el problema anterior

(c) Conversores A/D individuales por cada señal

# CONVERSIÓN ANALÓGICA A DIGITAL (A/D o ADC)

Los conversores A/D pueden:

- Truncar
- Redondear

Tipos de conversores A/D:

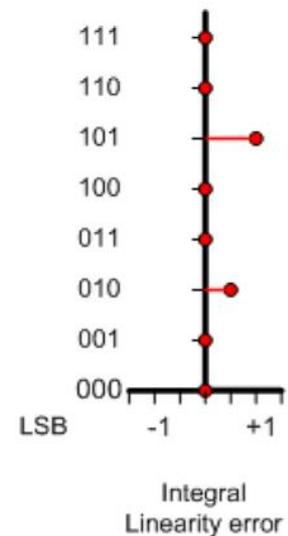
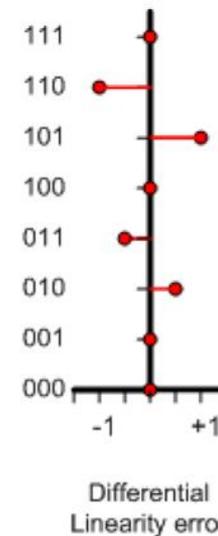
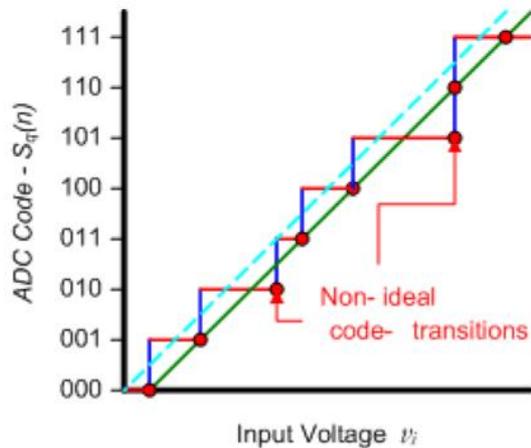
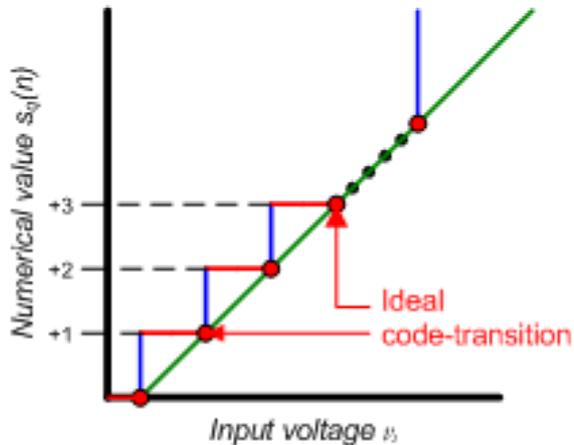
- Contadores de rampa
- De rampa simple o doble
- “Tracking”
- De aproximaciones sucesivas
- “Flash”

La conversión A/D tiene dos etapas:

- Muestrear (sampling)
- Cuantificar (generar el valor numérico binario)

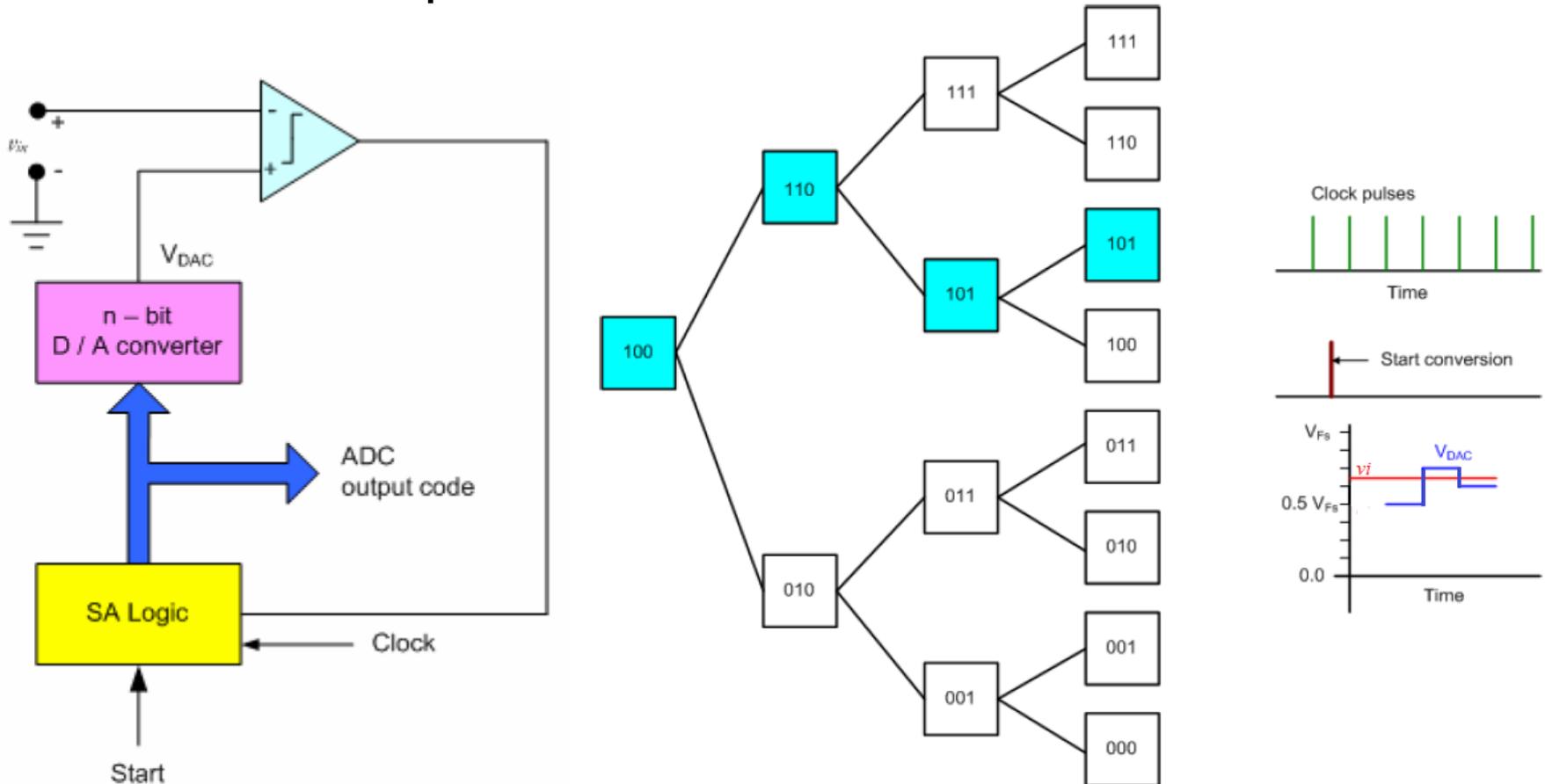
# CONVERSIÓN ANALÓGICA A DIGITAL (A/D o ADC)

- Transiciones ideales conversor A/D con redondeo
- Codificación no ideal, errores



# CONVERSIÓN ANALÓGICA A DIGITAL (A/D o ADC)

- Conversor A/D aproximaciones sucesivas



# ALGORITMOS DE PROTECCIÓN

## Filtros FIR

Un filtro FIR (*Finite Impulse Response*)

es un filtro digital en que:

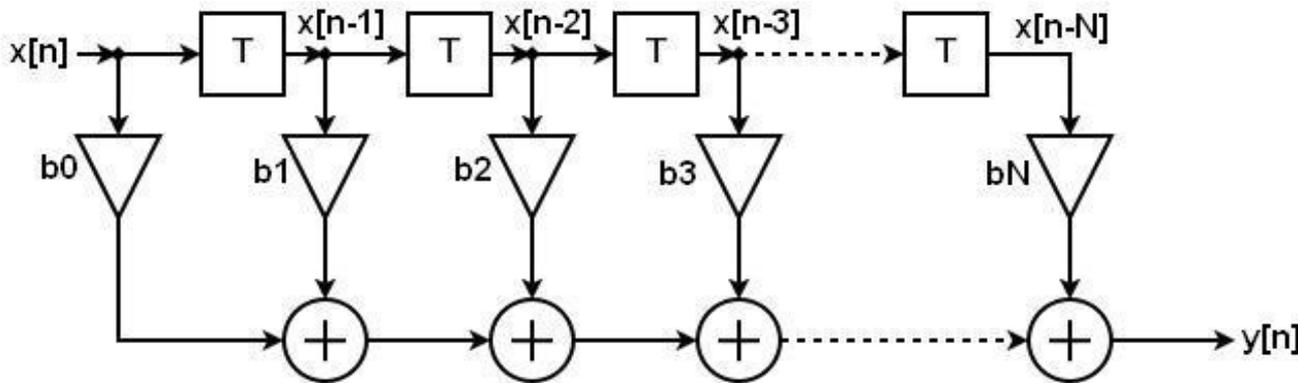
- si la entrada es un impulso de amplitud 1 (delta de Kronecker),
- la salida tendrá un número finito de términos no nulos (la salida será cero luego de un tiempo finito máximo)

# ALGORITMOS DE PROTECCIÓN

## Filtros FIR

La salida de un FIR de orden N en el instante n es:

$$y[n] = b_0x[n] + b_1x[n-1] + \dots + b_Nx[n-N] = \sum_{i=0}^N b_i x[n-i]$$



Maneja una historia de N muestras (N+1 términos)

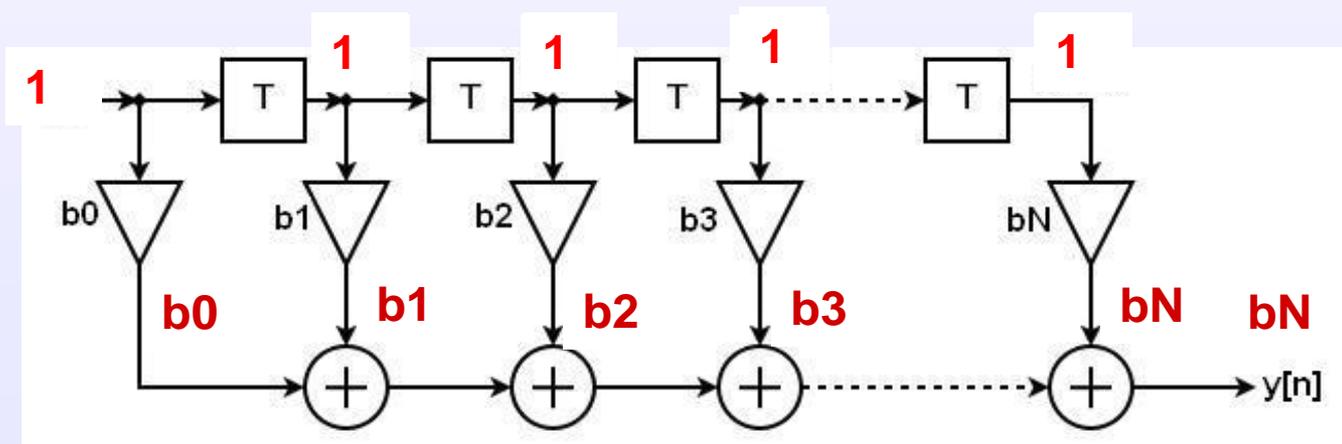
# ALGORITMOS DE PROTECCIÓN

## Filtros FIR

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$

$$n = 3 \quad t = t_0 + 3T \quad \Gamma$$

$$x[n] = \delta[n]$$



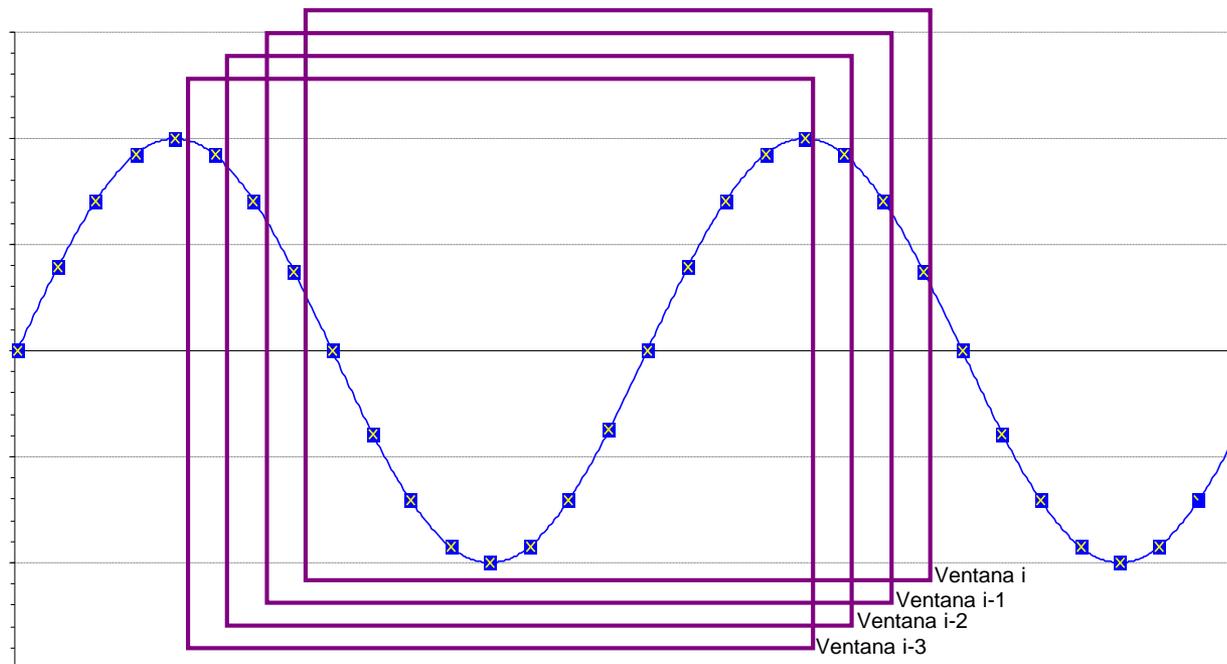
$$h[n] = \sum_{i=0}^N b_i \delta[n-i]$$

$$h[n] = \begin{cases} b_i, & n = i = 0 \dots N \\ 0, & \forall \text{ otro } n \end{cases}$$

# ALGORITMOS DE PROTECCIÓN

## Ventana móvil o deslizante

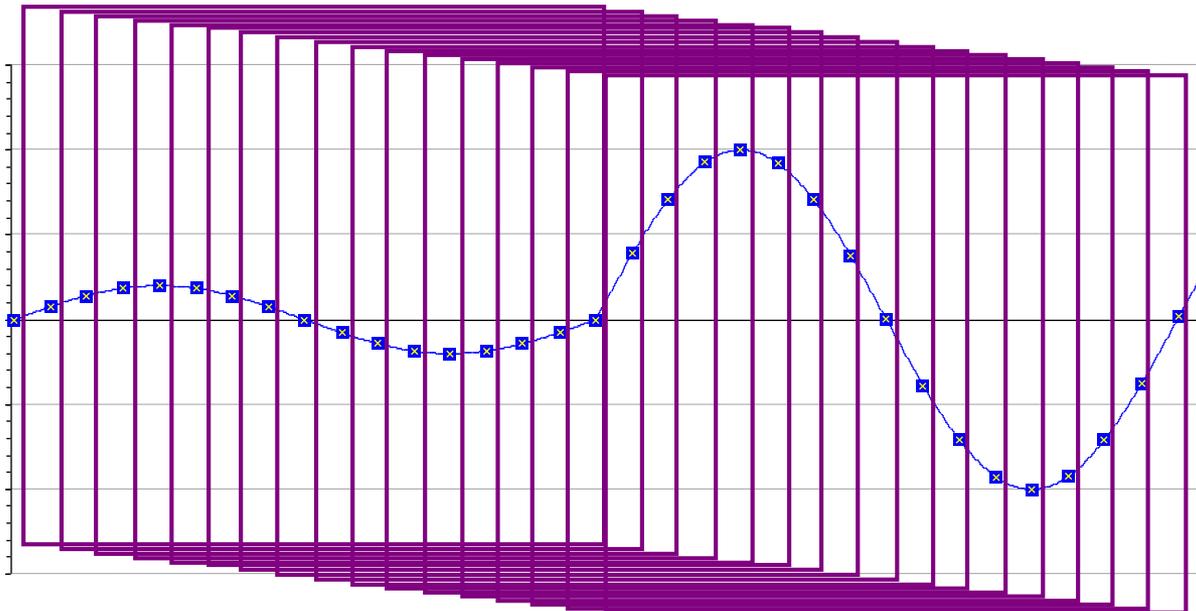
- La ventana es el conjunto de muestras que utiliza el filtro FIR para calcular el fasor en un instante dado.
- La ventana de tamaño  $N$ , consiste en la muestra actual y las  $N-1$  muestras anteriores (es la historia que “ve”).



# ALGORITMOS DE PROTECCIÓN

## Tiempo de operación y tamaño de la ventana

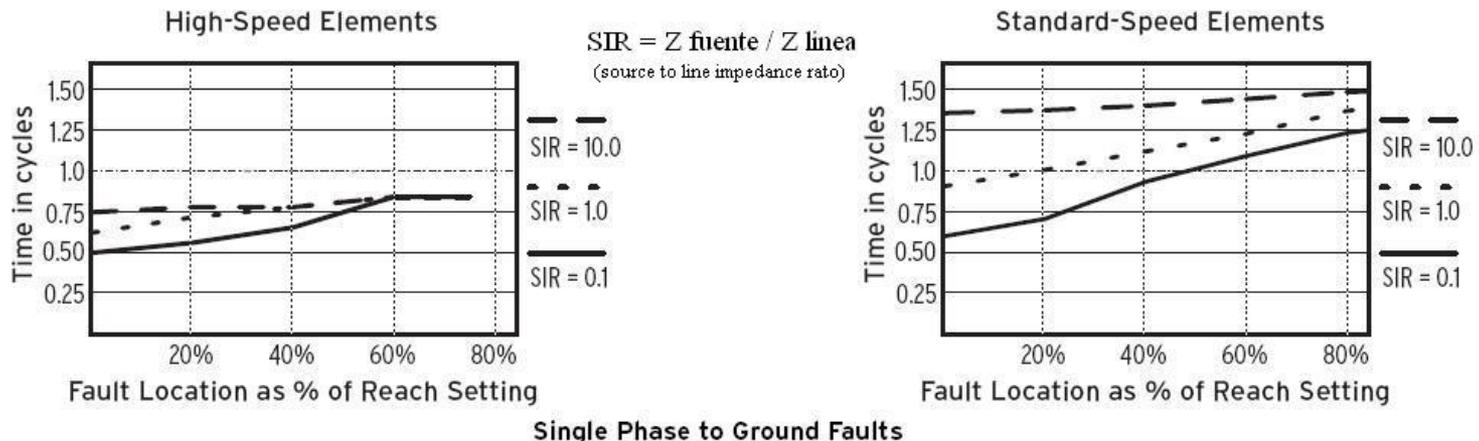
- La ventana móvil “avanza” e incluye más muestras de falta y menos muestras de pre-falta.
- Dos tareas:
  1. Detectar la falta
  2. Estimar satisfactoriamente el fasor  
(ventana con muestras pre-falta y falta no es confiable)



# ALGORITMOS DE PROTECCIÓN

## Tiempo de operación y tamaño de la ventana

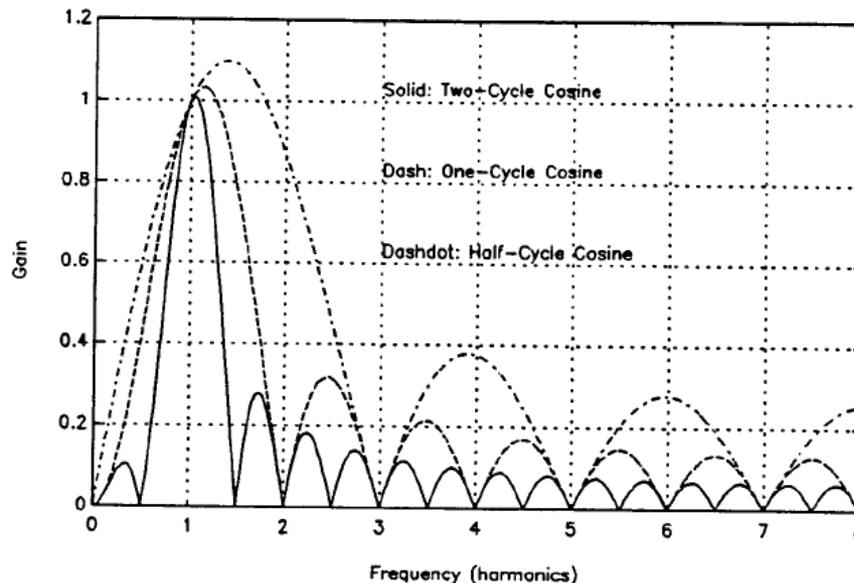
- Cuanto más pequeña la ventana, más rápida es la decisión pero, también más imprecisa (se verá)
- Ventana de 1 ciclo
  - tiempos de operación por ej. 30 o 40 ms.
- Ventanas de ½ ciclo
  - tiempos de operación por ej. 14 o 20 ms



# ALGORITMOS DE PROTECCIÓN

## Ventana y armónicos

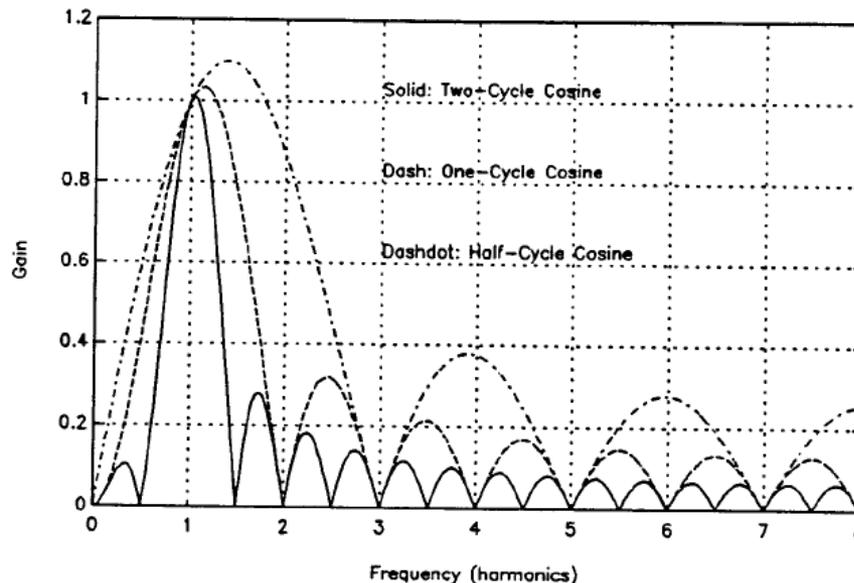
- Señal: senoide pura.
  - ventana de 1 ciclo de la fundamental es adecuada.
  - la ventana puede ser también de múltiplos del período.
- Señal: con armónicos, pero interesa la fundamental
  - la ventana tiene que ser de 1 ciclo
  - de esta manera se rechazan los armónicos (p. ej. DFT, filtro coseno)



# ALGORITMOS DE PROTECCIÓN

## Ventana y armónicos

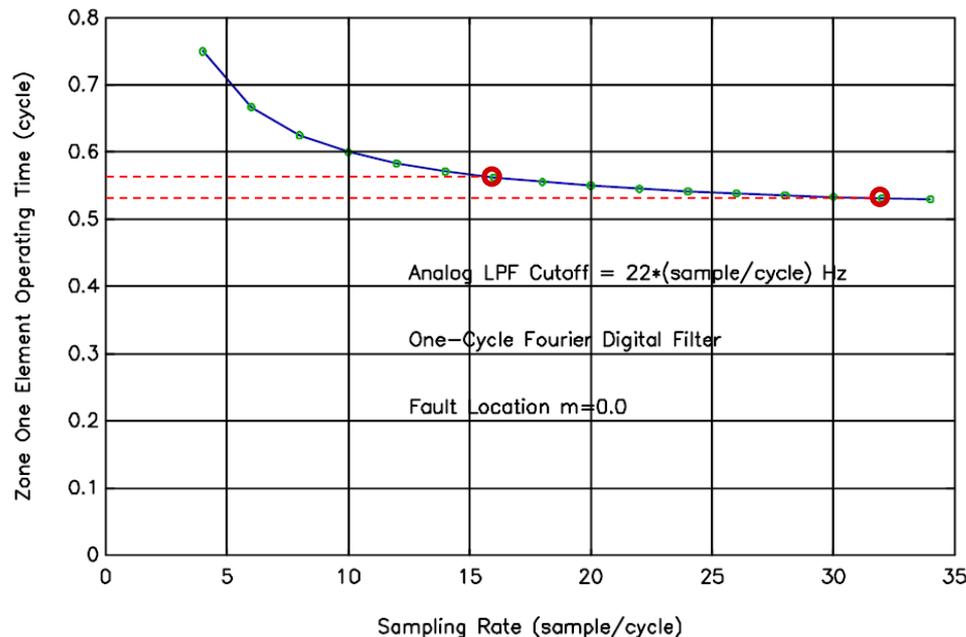
- Un FIR de menos de 1 ciclo no puede rechazar todos los armónicos
- Fasores de los armónicos
  - Ventana de 1 ciclo de la fundamental (o múltiplos)
  - Los pesos de las muestras se calculan considerando el período del armónico



# ALGORITMOS DE PROTECCIÓN

## Tiempo de operación vs. frecuencia de muestreo

- Filtro FIR (DFT) de 1 ciclo
  - pasar de 16 a 32 mpc no mejora demasiado
  - reduce  $\approx 4\%$  ( $\approx 0,04$  ciclos  $\approx 0,8$  ms) el tiempo de disparo, que ya era bajo ( $\approx 20 \cdot 0,57 \approx 12$  ms)



- Por eso las  $f_s$  usadas para protección no son muy elevadas.
- Oscilografías pueden requerir frecuencias más altas.

# ALGORITMOS DE PROTECCIÓN

## Tipos de Algoritmos Numéricos

- De estimación de fasores (Filtros FIR)
- En el dominio del tiempo
- De onda viajera

# ALGORITMOS DE PROTECCIÓN

## Algoritmos de estimación de fasores

- DFT (*Discret Fourier Transform*)
- Coseno
- Mínimos cuadrados
- Filtro de Kalman
- Transformada Wavelet

# ALGORITMOS DE PROTECCIÓN

## Algoritmos de estimación de fasores

### DFT (*Discret Fourier Transform*)

- Señal compuesta por frecuencia fundamental ( $h=1$ ) y armónicos (DC  $h=0$ ,  $h=2$ ,  $h=3$ , etc.)
- Señal periódica y la ventana corresponde al período (o un múltiplo del período)
- El fasor (de cualquier  $h$ ) se estima por la fórmula:

$$V_h = \frac{2}{N} \sum_{n=0}^{N-1} v_n e^{-jn \frac{2\pi h}{N}} = \underbrace{\frac{2}{N} \sum_{n=0}^{N-1} v_n \cos\left(\frac{2\pi n h}{N}\right)}_{P_R(V_h)} - j \underbrace{\frac{2}{N} \sum_{n=0}^{N-1} v_n \sin\left(\frac{2\pi n h}{N}\right)}_{j P_I(V_h)}$$

# ALGORITMOS DE PROTECCIÓN

## Algoritmos de estimación de fasores

### DFT (*Discret Fourier Transform*)

de 1 ciclo (ventana = período)

$$V_h = \frac{2}{N} \sum_{n=0}^{N-1} v_n e^{-jn \frac{2\pi h}{N}} = \underbrace{\frac{2}{N} \sum_{n=0}^{N-1} v_n \cos\left(\frac{2\pi n h}{N}\right)}_{P_R(V_h)} - j \underbrace{\frac{2}{N} \sum_{n=0}^{N-1} v_n \sin\left(\frac{2\pi n h}{N}\right)}_{j P_I(V_h)}$$

La componente  $h$  de la señal  $v(t)$  se podría reconstruir a partir de la DFT como:

- $v_h(t) = |V_h| \cos(\omega_h t) + \phi_h$  o sea
- $v_h(t) = P_R(V_h) * \cos(\omega_h t) - P_I(V_h) * \sin(\omega_h t)$

$$|V_h| = \sqrt{P_R(V_h)^2 + P_I(V_h)^2}$$

$$\phi_h = \arctg \frac{P_I(V_h)}{P_R(V_h)}$$

$$\omega_h = \frac{2\pi h}{T}$$

# ALGORITMOS DE PROTECCIÓN

## Algoritmos de estimación de fasores

### ¿Por qué DFT (*Discret Fourier Transform*)?

- **Serie de Fourier (análisis armónico) (\*)**

$$v(t) \approx \frac{a_0}{2} + \sum_{h=1}^{\infty} a_h \cos\left(\frac{2\pi h}{T} t\right) + \sum_{h=1}^{\infty} b_h \sin\left(\frac{2\pi h}{T} t\right) \quad a_h = \frac{2}{T} \int_0^T v(t) \cos\left(\frac{2\pi h}{T} t\right) dt \quad b_h = \frac{2}{T} \int_0^T v(t) \sin\left(\frac{2\pi h}{T} t\right) dt$$

- **Fórmulas de Bessel (análisis armónico aproximado) (\*)**

$$v(t) \approx \frac{a_0}{2} + \sum_{h=1}^r a_h \cos\left(\frac{2\pi h}{T} t\right) + \sum_{h=1}^r b_h \sin\left(\frac{2\pi h}{T} t\right) \quad a_h = \frac{2}{N} \sum_{n=0}^{N-1} v_n \cos\left(\frac{2\pi n h}{N}\right) \quad b_h = \frac{2}{N} \sum_{n=0}^{N-1} v_n \sin\left(\frac{2\pi n h}{N}\right)$$

$$v_h(t) \approx a_h \cos\left(\frac{2\pi h}{T} t\right) + b_h \sin\left(\frac{2\pi h}{T} t\right)$$

(\*) *Es la mejor aproximación de una función periódica e integrable en el período (menor error cuadrático medio; método de los mínimos cuadrados)*

- **DFT**

$$v_h(t) = P_R(V_h) \cos \omega_h t - P_I(V_h) \sin \omega_h t \quad P_R(V_h) = \frac{2}{N} \sum_{n=0}^{N-1} v_n \cos\left(\frac{2\pi n h}{N}\right) \quad P_I(V_h) = -\frac{2}{N} \sum_{n=0}^{N-1} v_n \sin\left(\frac{2\pi n h}{N}\right)$$

- **entonces**

$$a_h = P_R(V_h)$$

$$b_h = -P_I(V_h)$$



# ALGORITMOS DE PROTECCIÓN

## Algoritmos de estimación de fasores

### DFT (Discret Fourier Transform) NO Recursivo

$$V_h = \frac{2}{N} \sum_{n=0}^{N-1} v_n e^{-jn \frac{2\pi h}{N}} = \frac{2}{N} \sum_{n=0}^{N-1} v_n \cos\left(\frac{2\pi n h}{N}\right) - j \frac{2}{N} \sum_{n=0}^{N-1} v_n \sin\left(\frac{2\pi n h}{N}\right)$$

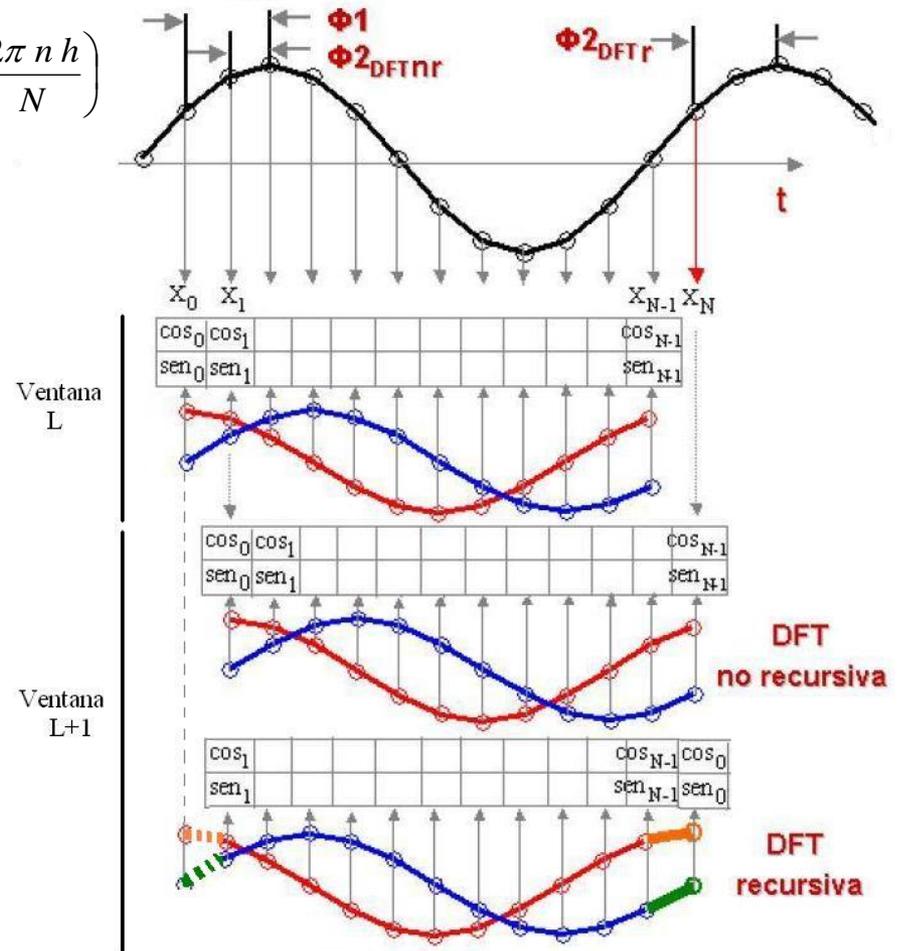
$\Phi$  cambia en c/ventana  
 (adelanta  $\frac{2\pi h}{N}$ ) pues cambia el origen

No afecta si calculo  
 cocientes  $\bar{Z} = \frac{\bar{V}}{\bar{I}}$

$\cos_n$  y  $\sin_n$  precalculados

Calcula en c/ventana:

- $2N \times$
- $2(N-1) +$



# ALGORITMOS DE PROTECCIÓN

## Algoritmos de estimación de fasores

### DFT (Discret Fourier Transform) RECURSIVO

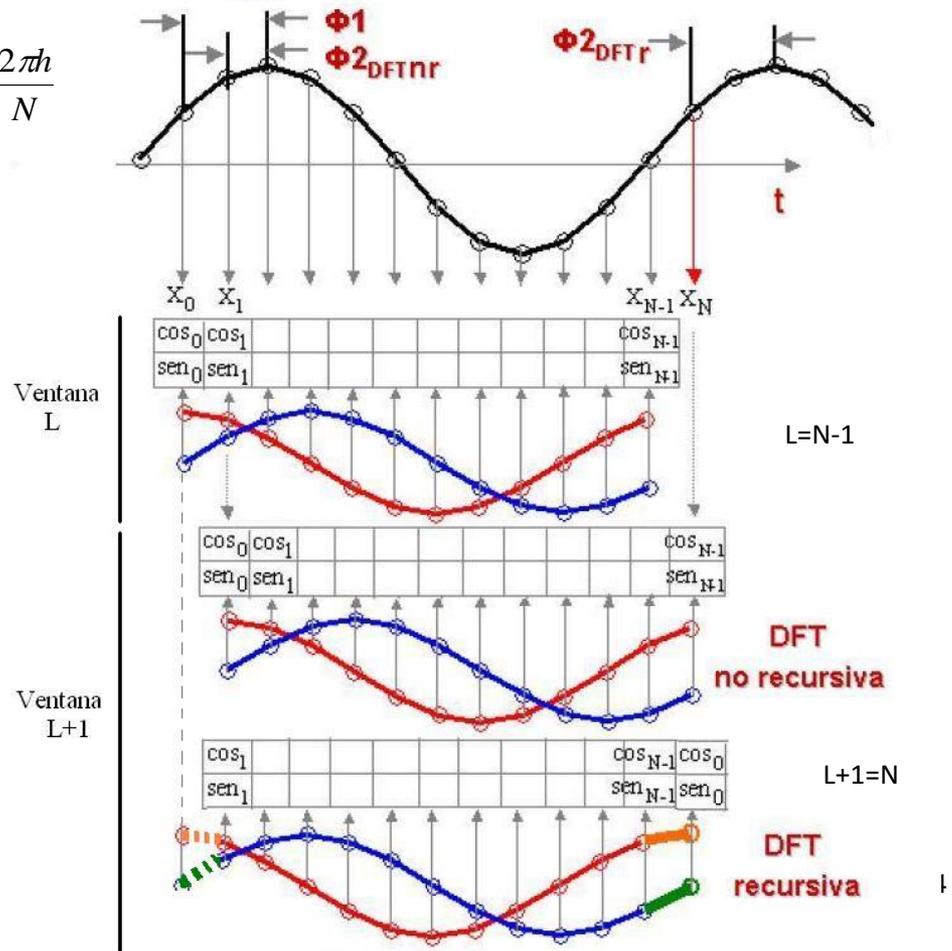
$$\tilde{V}_h^{L+1} = \tilde{V}_h^L + \frac{2}{N} (v_{L+1} - v_{L-N+1}) e^{-jL \frac{2\pi h}{N}}$$

$\Phi$  fijo (así se definió)

Calcula en c/ventana:

- 3 +
- 2 x

Más eficiente



# ALGORITMOS DE PROTECCIÓN

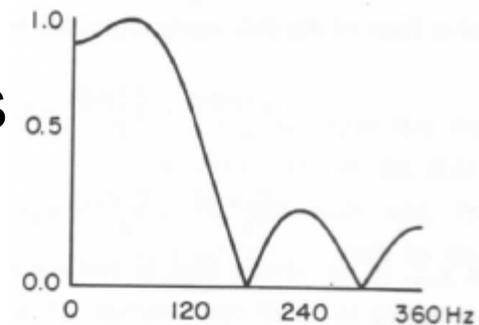
## Algoritmos de estimación de fasores

### “DFT” de 1/2 ciclo

- La ventana no abarca todo el período de la señal (DFT asume señal periódica)
- No puede rechazar todos los armónicos (no rechaza los pares)

$$\tilde{V}_h^{L+1} = \tilde{V}_h^L + \frac{2}{N} (v_{L+1} + v_{L-N+1}) e^{-jL \frac{\pi h}{N}} \quad \text{para } h \text{ impar}$$

$$\tilde{V}_h^{L+1} = \tilde{V}_h^L + \frac{2}{N} (v_{L+1} - v_{L-N+1}) e^{-jL \frac{\pi h}{N}} \quad \text{para } h \text{ par}$$



- Más impreciso. Más rápido (entra antes en el defecto).

# ALGORITMOS DE PROTECCIÓN

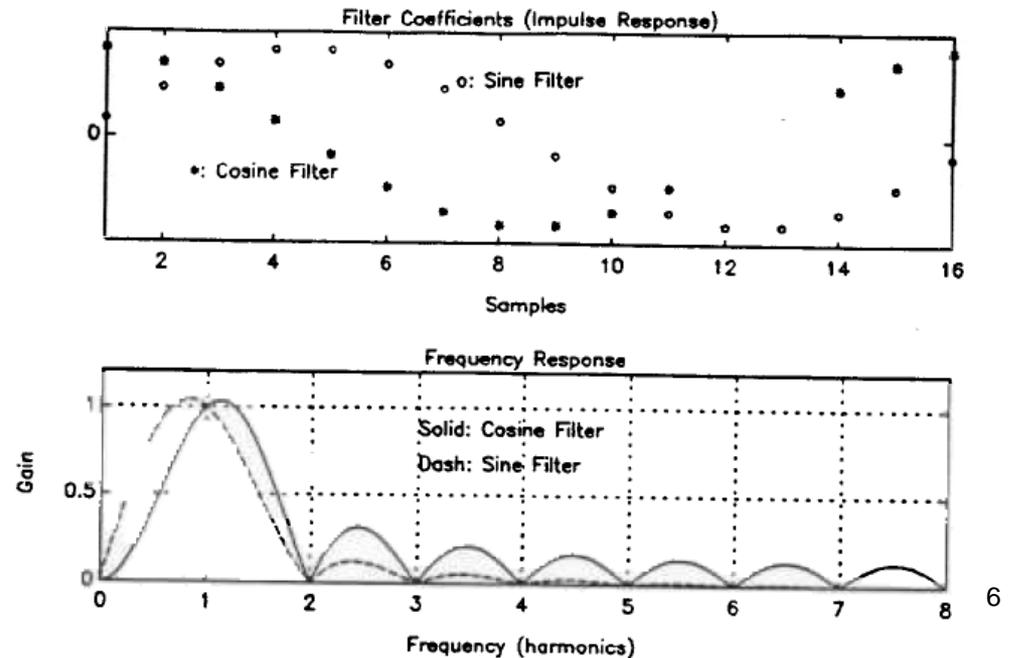
## Algoritmos de estimación de fasores

### Filtro coseno

- Filtros seno y coseno son los usados en DFT (ortogonales).

$$V_h = \frac{2}{N} \sum_{n=0}^{N-1} v_n e^{-jn\frac{2\pi h}{N}} = \frac{2}{N} \sum_{n=0}^{N-1} v_n \cos\left(\frac{2\pi n h}{N}\right) - j \frac{2}{N} \sum_{n=0}^{N-1} v_n \sin\left(\frac{2\pi n h}{N}\right)$$

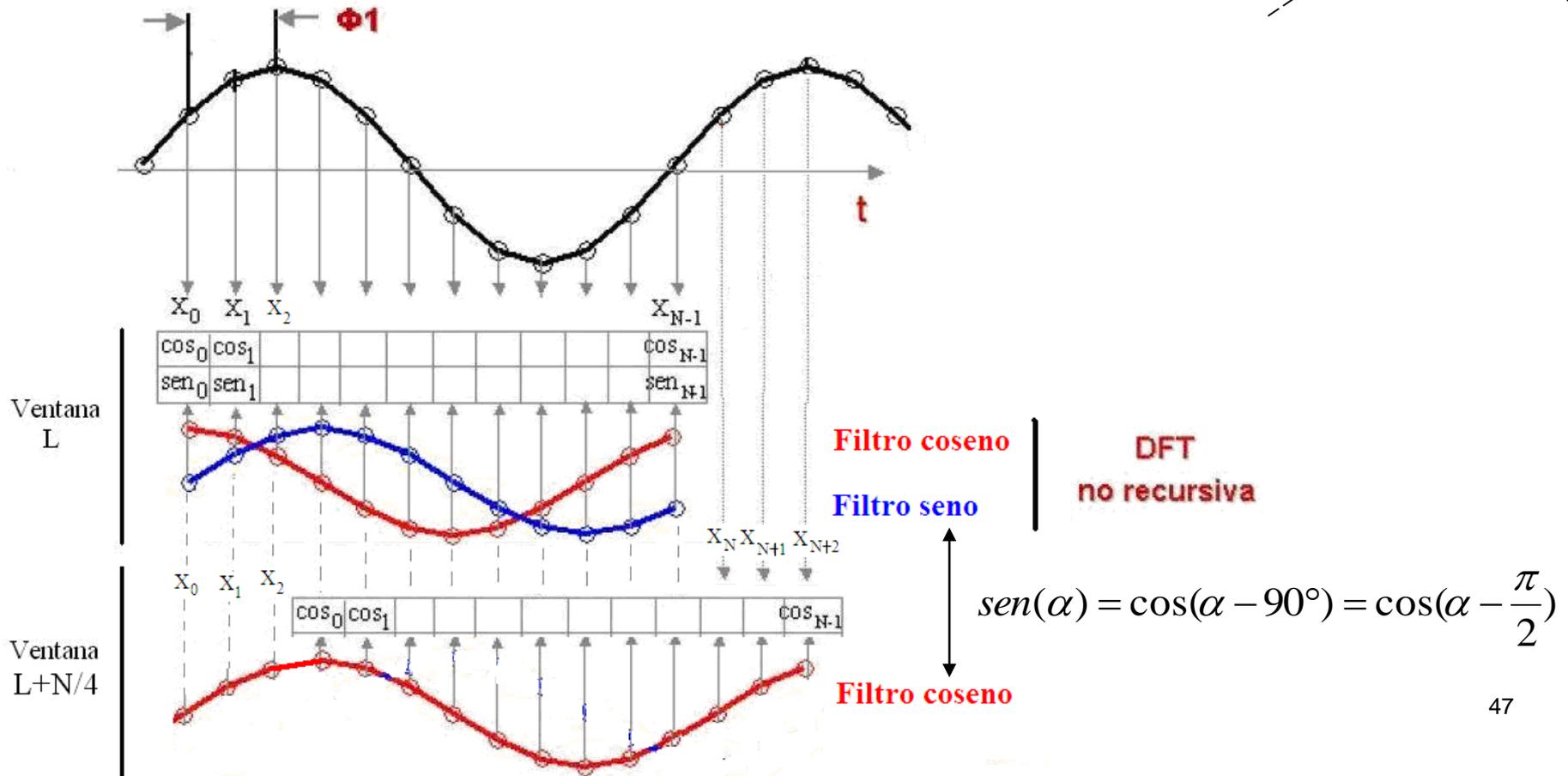
- Rechazan armónicos y DC



# ALGORITMOS DE PROTECCIÓN

## Algoritmos de estimación de fasores

**Filtro coseno**  $V_h = \frac{2}{N} \sum_{n=0}^{N-1} v_n e^{-jn \frac{2\pi h}{N}} = \frac{2}{N} \sum_{n=0}^{N-1} v_n \cos\left(\frac{2\pi n h}{N}\right) - j \frac{2}{N} \sum_{n=0}^{N-1} v_n \sin\left(\frac{2\pi n h}{N}\right)$



# ALGORITMOS DE PROTECCIÓN

## Algoritmos de estimación de fasores

### Filtro coseno

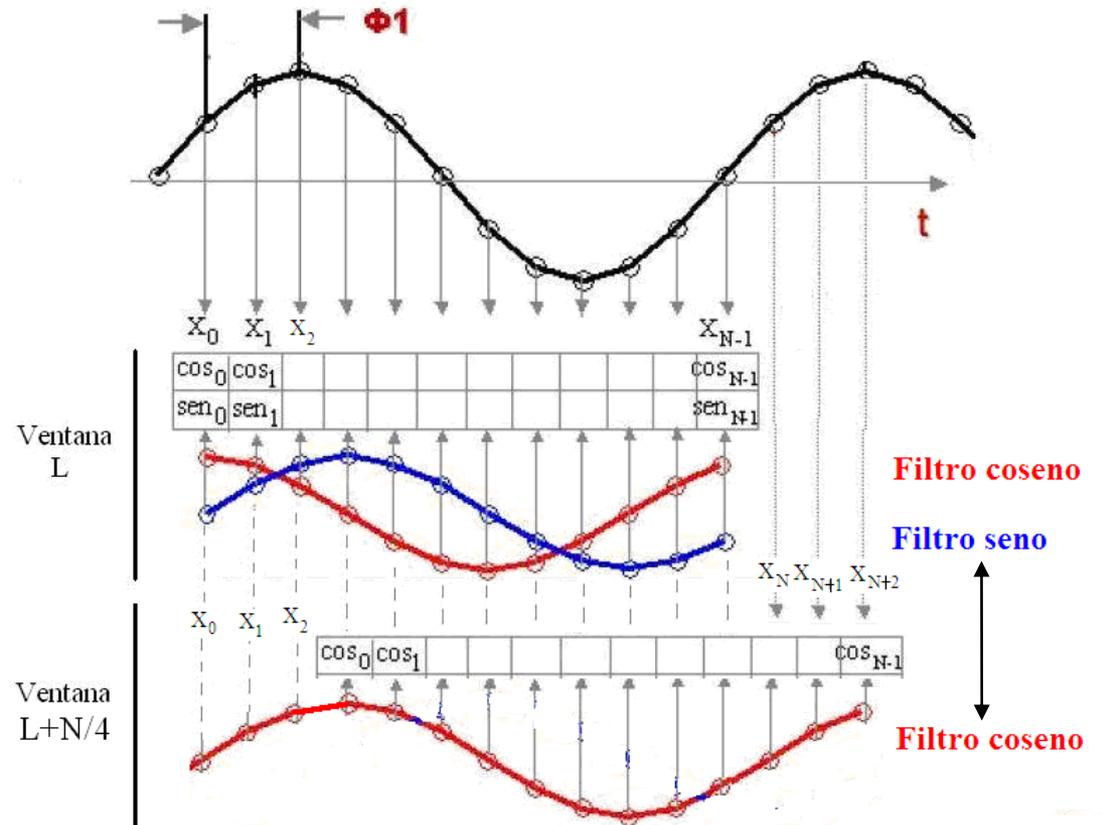
- Se puede usar sólo un filtro coseno para estimar un fasor.

Veamos por qué

$$P_I^L(V_h) \approx P_R^{L+N/4}(V_h)$$

En el ejemplo:  $N/4 = 3$

Ventana L		Ventana L+3
$X_0$	=	$X_N$
$\text{sen}_0$	=	$\text{cos}_{N-3}$
$X_0 \text{sen}_0$	=	$X_N \text{cos}_{N-3}$
$X_1 \text{sen}_1$	=	$X_{N+1} \text{cos}_{N-2}$
$X_2 \text{sen}_2$	=	$X_{N+2} \text{cos}_{N-1}$
$X_3 \text{sen}_3$	=	$X_3 \text{cos}_0$
$X_4 \text{sen}_4$	=	$X_4 \text{cos}_1$
$X_{N-1} \text{sen}_{N-1}$	=	$X_{N-1} \text{cos}_{N-4}$



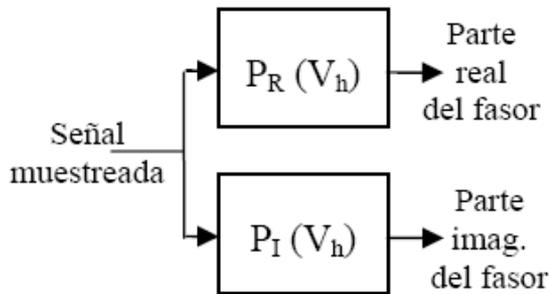
# ALGORITMOS DE PROTECCIÓN

## Algoritmos de estimación de fasores

### Filtro coseno

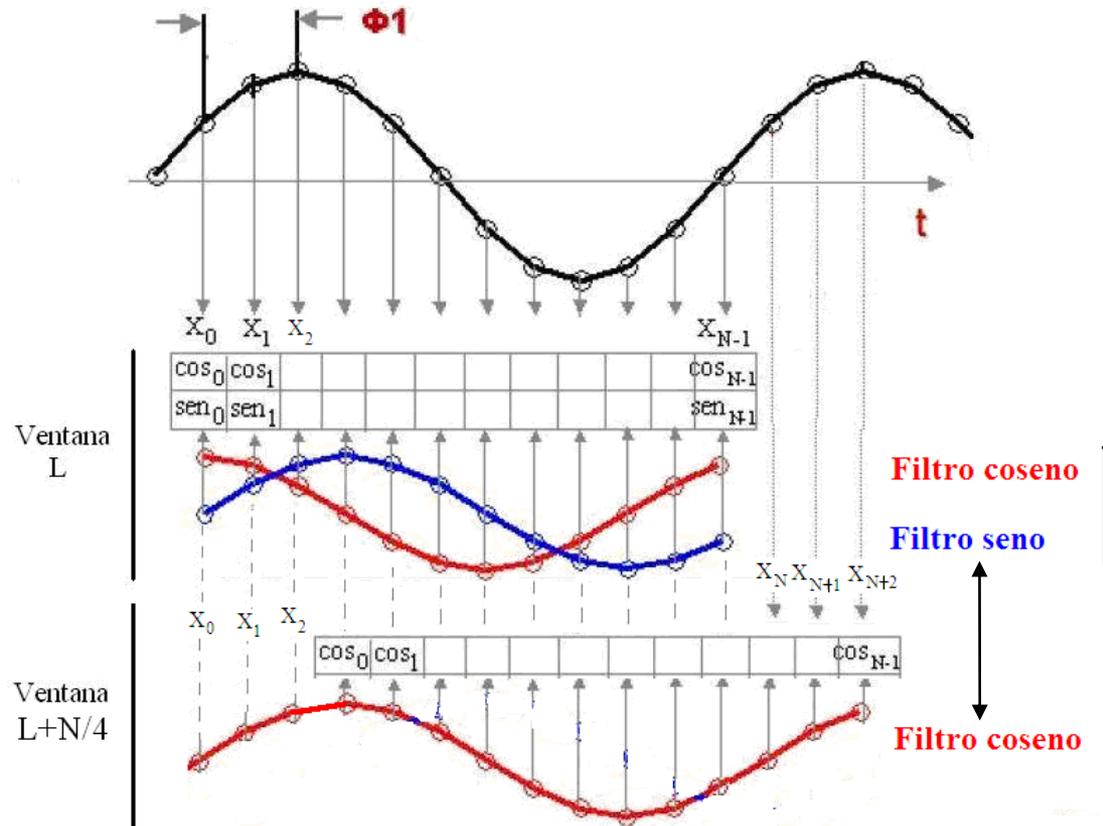
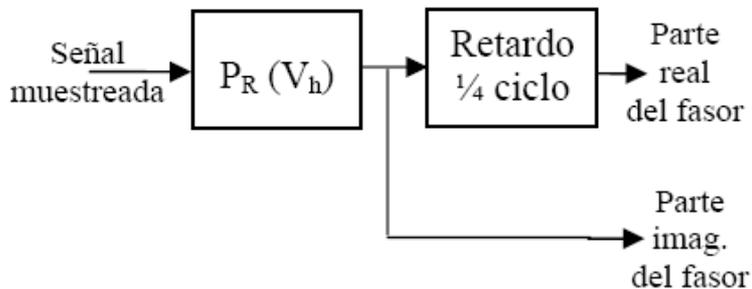
- Se puede usar sólo un filtro coseno para estimar un fasor.

#### Filtro DFT



≈

#### Filtro coseno

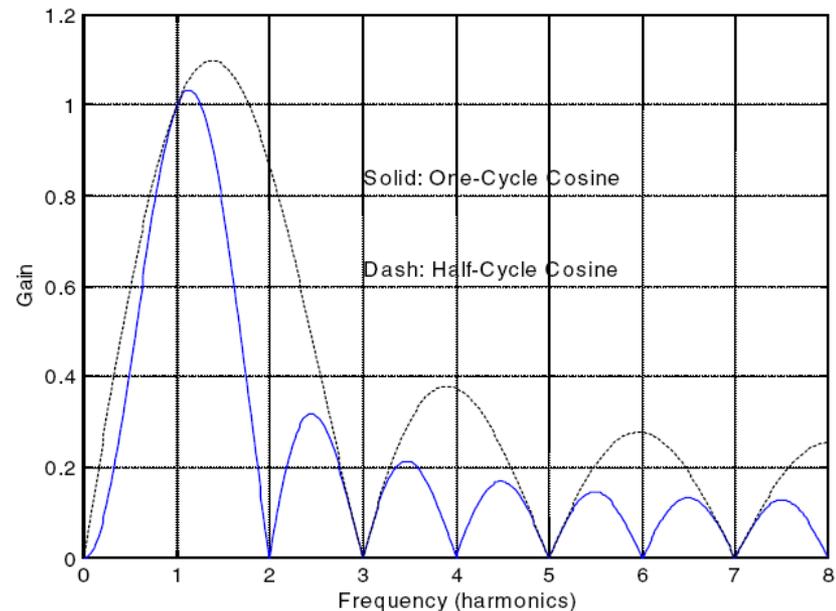


# ALGORITMOS DE PROTECCIÓN

## Algoritmos de estimación de fasores

### Filtro coseno

- Respecto de DFT retrasa (alarga) el cálculo  $\frac{1}{4}$  de ciclo.
- Respuesta en frecuencia de filtro coseno de 1 ciclo y de  $\frac{1}{2}$  ciclo



# ALGORITMOS DE PROTECCIÓN

## Algoritmos de estimación de fasores

### Resumen

Frecuencias de muestreo elevadas requieren procesadores potentes para rápidamente:

- Realizar conversión A/D de todas las señales
- Computar algoritmos y decisiones.

### Ventana

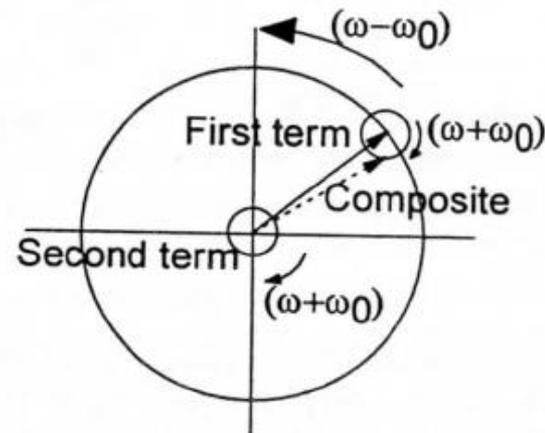
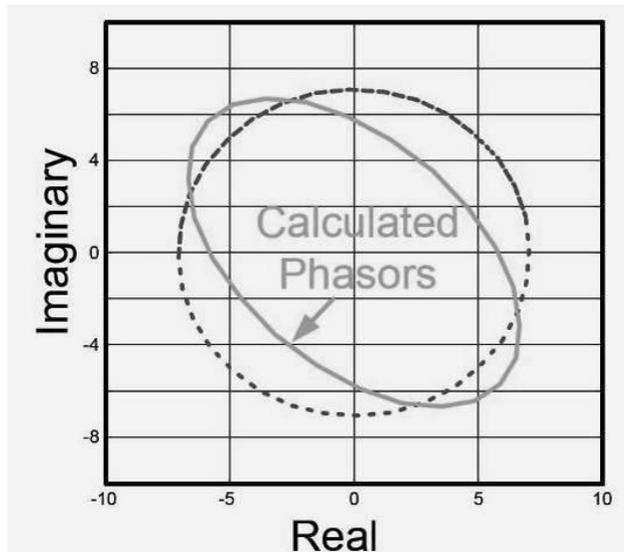
- De 1 ciclo filtra todos los armónicos
- Más pequeña (1/2 ciclo) hace más rápido al algoritmo, pero más impreciso

Aumentar demasiado la frec. de muestreo no mejora sensiblemente el tiempo de operación

# ALGORITMOS DE PROTECCIÓN

## Frecuencias distintas de la nominal

- Muchos relés numéricos muestrean las señales a frecuencia constante  $f_s$
- Si la frecuencia del sistema eléctrico no es la frecuencia nominal, se producen distorsiones

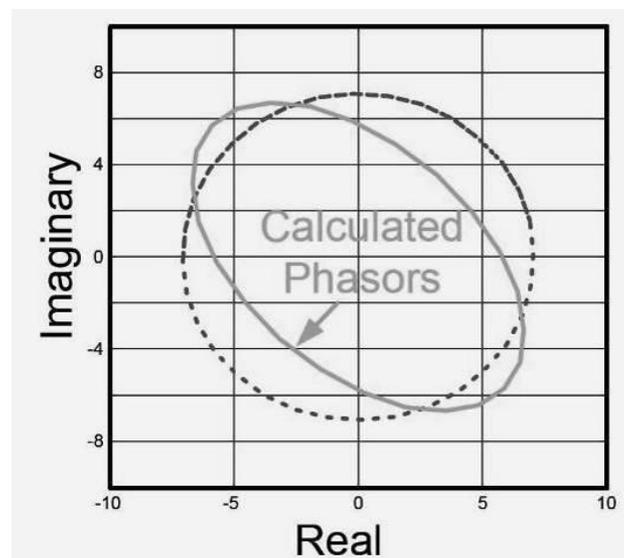


# ALGORITMOS DE PROTECCIÓN

## Frecuencias distintas de la nominal

DFT no recursiva sin corrección de fase

- Un fasor de 50 Hz (frec. nominal) describe una circunferencia de período 20 ms



- Un fasor de  $f \neq f_0$  describe una curva parecida a una elipse, en que la amplitud y fase del fasor tiene cierto error.

# ALGORITMOS DE PROTECCIÓN

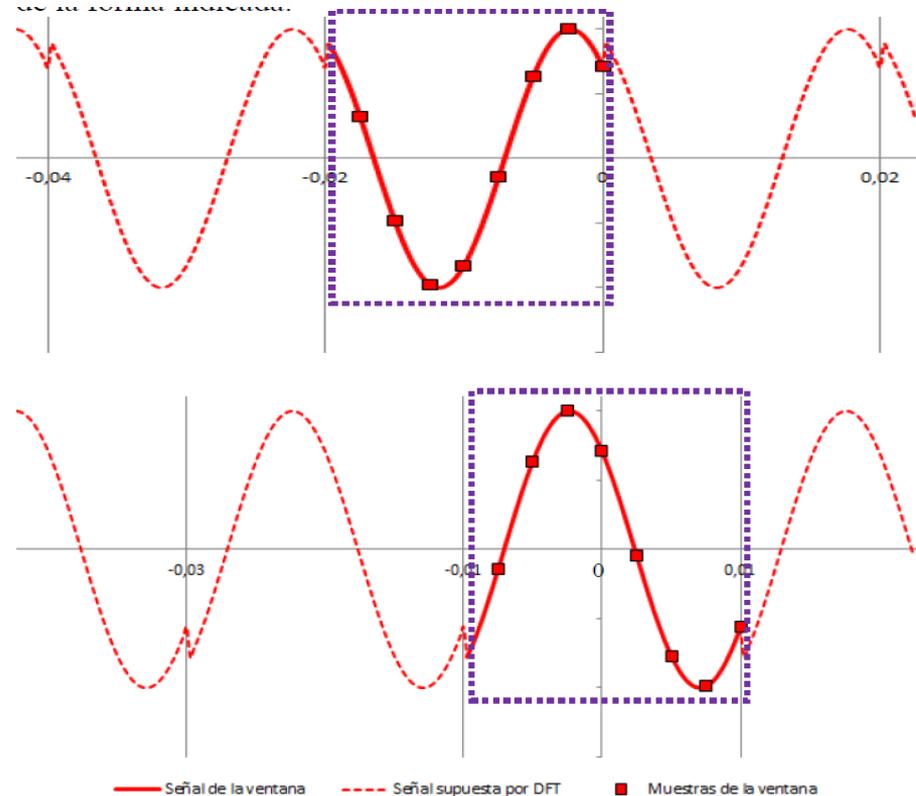
## Frecuencias distintas de la nominal

DFT no recursiva sin corrección de fase

- Una señal de **53 Hz** se ve de la siguiente manera en dos ventanas distintas de tamaño fijo 1 ciclo de la fundamental nominal (20 ms).
- ¿Cuál es la frecuencia fundamental de las señales de ambas figuras?

**¡50 Hz!**

- La DFT de ambas ventanas estima fasores distintos (ambos de 50Hz) pues la forma de onda es distinta

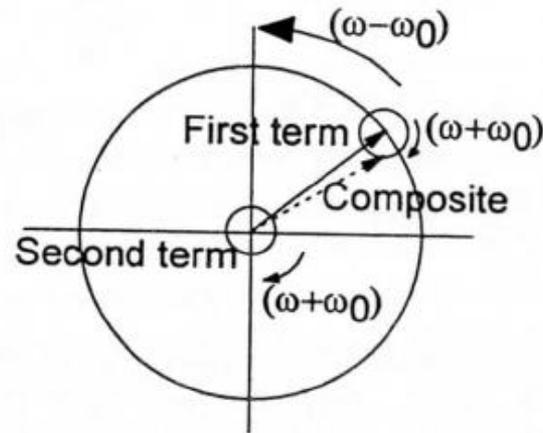


# ALGORITMOS DE PROTECCIÓN

## Frecuencias distintas de la nominal

DFT no recursiva con corrección de fase, o DFT recursiva

- El fasor de 50 Hz (frec. nominal) queda fijo.

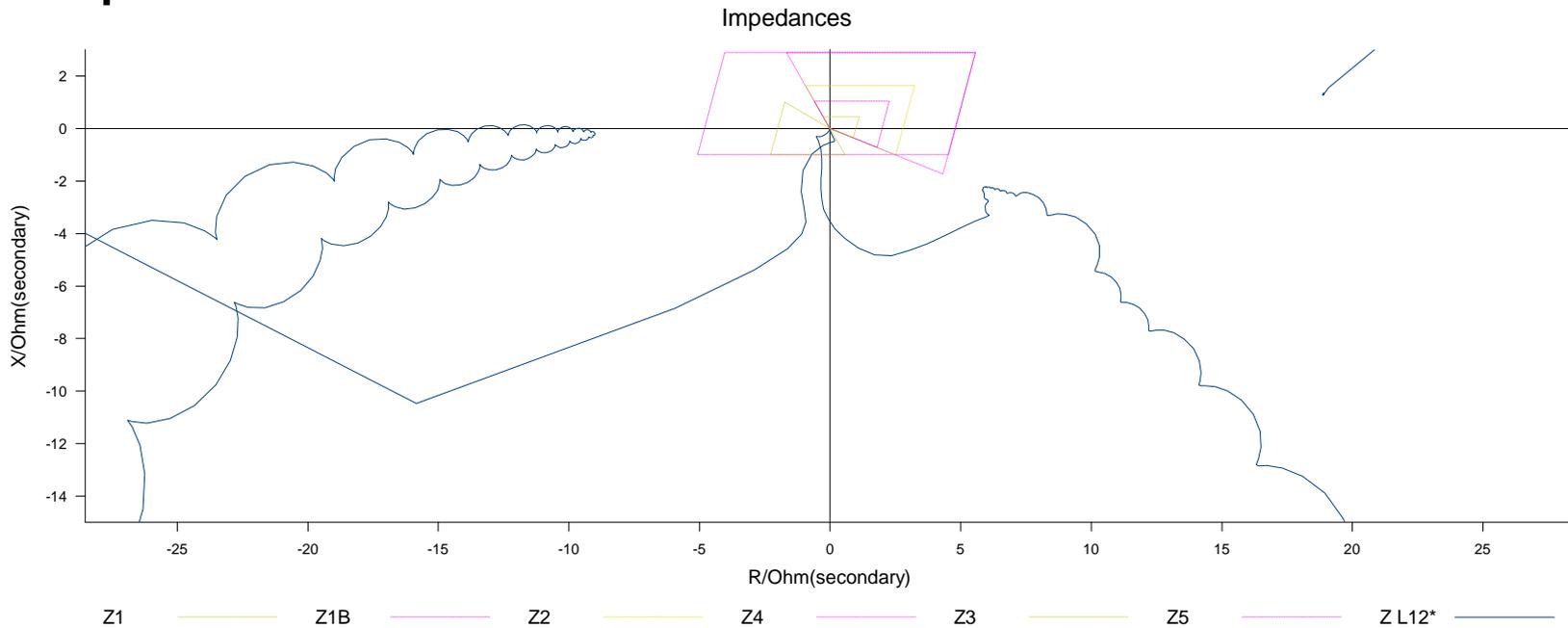


- Un fasor de  $f \neq f_0$  describe un “rulo” con componentes de velocidad  $\omega - \omega_0$  y  $\omega + \omega_0$

# ALGORITMOS DE PROTECCIÓN

## Frecuencias distintas de la nominal

- Una oscilación de potencia vista en el plano de impedancias

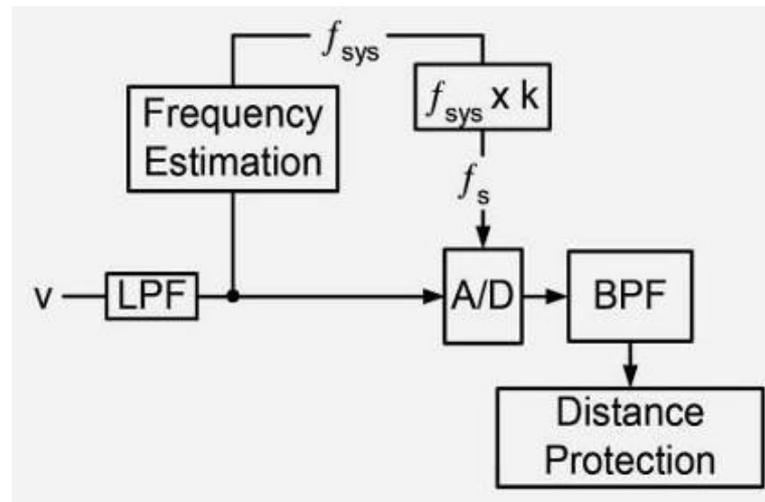


- Cada “rulo” corresponde a 10 ms aprox.

# ALGORITMOS DE PROTECCIÓN

## Frecuencias distintas de la nominal

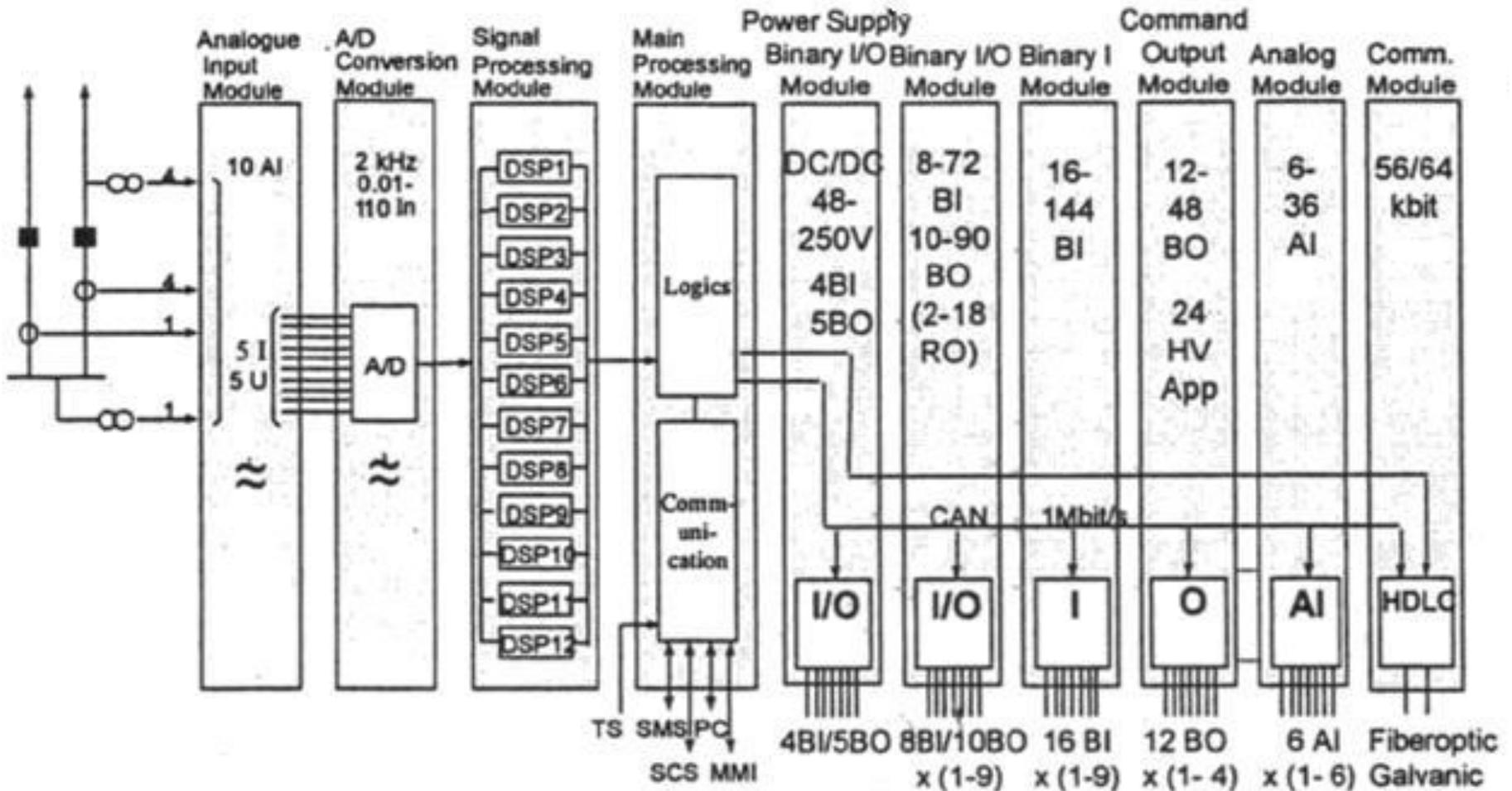
- Muestreo a una tasa de muestras por ciclo constante
- También se debe corregir la frecuencia de corte del filtro antialiasing



- Otras técnicas: filtrado de módulo y argumento, Smart DFT (SDFT), muestreo asíncrono

# Un relé comercial (1)

## RELÉ DE LÍNEA (21/87L/67, ETC.) DE 1999



# Un relé comercial (1)

## RELÉ DE LÍNEA (21/87L/67, ETC.) DE 1999

Filtro antialiasing:

- de primer orden.  $f_c = 500Hz$

Conversor A/D:

- 10 entradas analógicas muestreadas a  $f_s = 2kHz$  (40 muestras por ciclo)
- Ancho de banda: 250 Hz
- Resolución: conversor A/D de 12 bits, aumentado a 16 bits en las corrientes mediante manejo dinámico del rango (2 shunts/canal).

DSPs:

- 12 DSPs
- Desarrollan los algoritmos de protección trabajando en paralelo.
- Se baja el muestreo a  $f_s = 1kHz$  (20 muestras por ciclo, o sea muestreado cada 1 ms).
- Incorporan un filtro numérico (otro antialiasing) con  $f_c = 250Hz$

# Un relé comercial (1)

## RELÉ DE LÍNEA (21/87L/67, ETC.) DE 1999

Procesador central:

- Procesador de 32 bits.
- Recibe los resultados de los algoritmos de los DSPs cada 1 ms.
- Da las órdenes, maneja las comunicaciones, E/S, etc.

Lógicas:

- De protección: ciclo de 6 ms.
- De automatización (control): ciclo de 200 ms.
- Otro modelo de la misma época y fabricante tiene lógicas cuyo refresco puede ser cada 5, 10 o 20 ms (seleccionable).

# Un relé comercial (2)

## RELÉ DE LÍNEA (21/67, ETC.) DE 2005

Filtro antialiasing:

- Pre-filtrado  $f_c = 3kHz$  para la oscilografía.
- Antialiasing digital para muestreo de protección

Muestreo:

- Vs y Is muestreadas a  $f_s = 2kHz$  para oscilografía (40 muestras por ciclo, o sea cada 0,5 ms aprox.)
- Bajada a 8 muestras por ciclo para protección  $f_s = 400Hz$
- Muestreo adaptado a la frecuencia real de las tensiones medidas (8 muestras por ciclo).
- Señales digitales muestreadas a  $f_s = 8kHz$  para oscilografía (160 muestras por ciclo, o sea cada 0,125 ms aprox.).

# Un relé comercial (2)

## RELÉ DE LÍNEA (21/67, ETC.) DE 2005

Conversor A/D:

- Resolución de 16 bits

Lógicas:

- De protección: 8 veces por ciclo (ciclo de 2,5 ms aprox.).
- De automatización (control): ciclo de entre 8 y 20 ms (baja prioridad; el ciclo depende de las actividades más prioritarias).

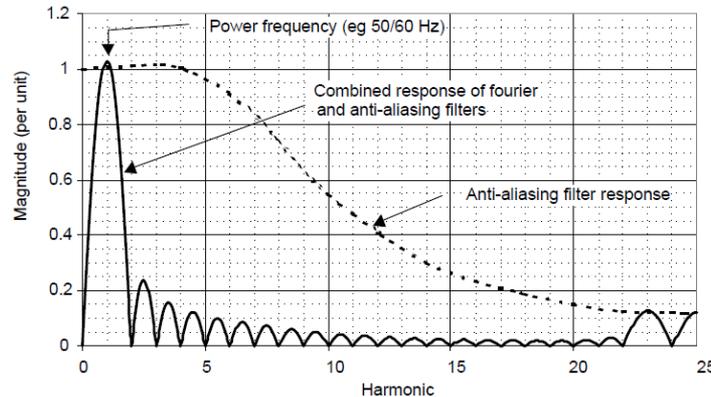
Algoritmos:

- Filtro coseno de 1 ciclo. Funciona bien pues la frecuencia de muestreo está adaptada a la frecuencia de la señal (8 muestras por ciclo).
- DFT (Digital Fourier Transform) de  $\frac{1}{2}$  ciclo (responde entre 3 y 10 ms; menos preciso)

# Un relé comercial (3)

## RELÉ DE GENERADOR DE 2014

Filtro antialiasing:



Muestreo, multiplexado y conversión A/D:

- Multiplexado de 16 canales
- Muestreo a 24 muestras / ciclo de la frecuencia real (frequency tracking).
- Resolución: conversor A/D de 16 bits

Procesador central:

- DSP de 32 bits y 150 MHz (trabajando a 75 MHz).

Algoritmos:

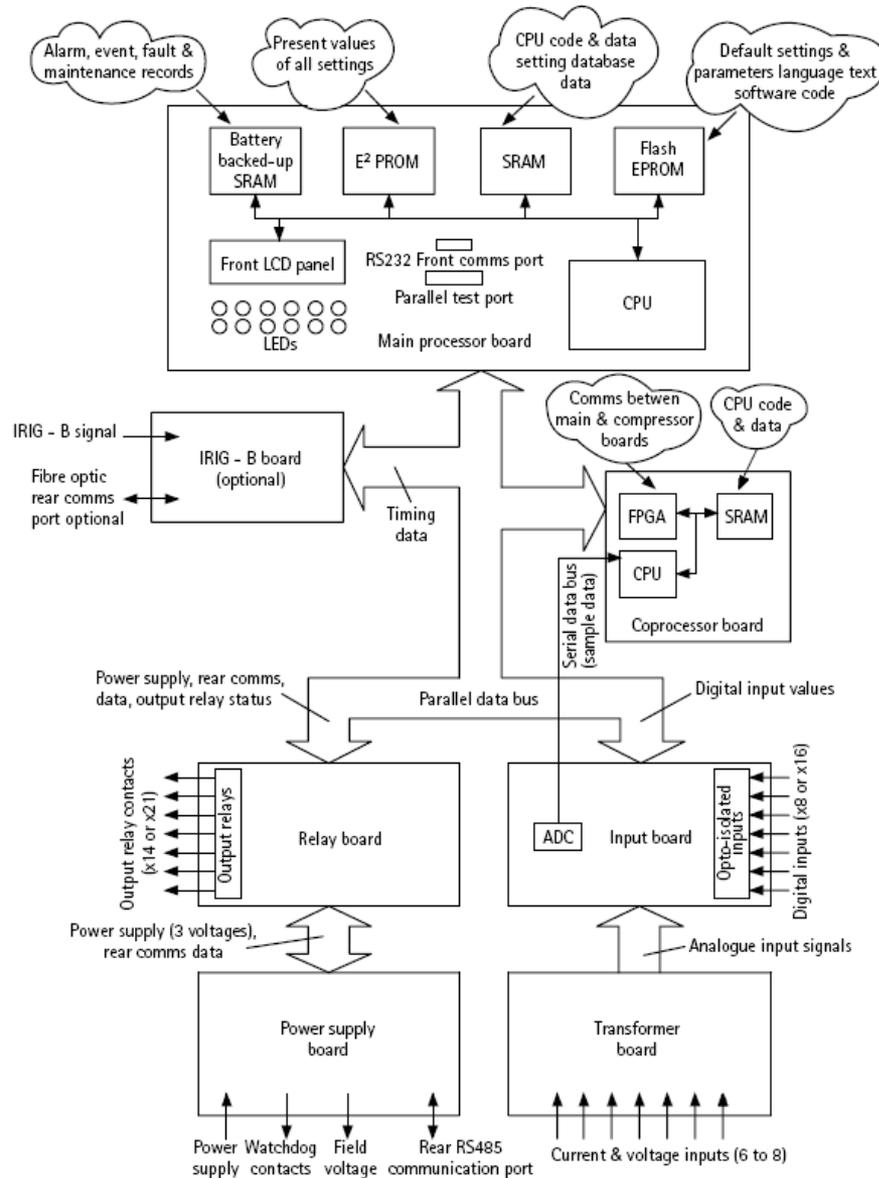
- DFT (Digital Fourier Transform) de 1 ciclo
- ¿y de  $\frac{1}{2}$  ciclo y  $\frac{1}{4}$  ciclo si la señal es “limpia”?

# Un relé comercial (3)

## RELÉ DE GENERADOR DE 2014

### Estructura de hardware

- Módulos
- Flujos de Información
- Usos de memorias



Legend:  
 SRAM - Static Read Only Memory  
 CPU - Central Processing Unit  
 IRIG-B - Time Synchronisation Signal  
 FPGA - Field Programmable Logic Array  
 ADC - Analog to Digital Converter  
 E<sup>2</sup>PROM - Electrically Erasable Programmable Read Only Memory  
 EPROM - Electrically Programmable Read Only Memory  
 LCD - Liquid Crystal Display

# Un relé comercial (3)

## RELÉ DE GENERADOR DE 2014

### Estructura de software

