

Muestreo y Adquisición Inteligente de Señales Sensoriales en Sistemas Embebidos

Gustavo Monte, Kector Kessel, Norberto Scarone, Cesar Almendra

Universidad Tecnológica Nacional - Facultad Regional del Neuquén – Plaza Huincul - Argentina
gmonte@frn.utn.edu.ar

Resumen- Hoy en día es muy común encontrar el prefijo inteligente en los sistemas que el hombre desarrolla. Encontramos desde fusibles inteligentes hasta autopistas inteligentes. La razón de esta “inteligencia” es que dada la complejidad de los sistemas actuales, los diseñadores la han trasladado a los dispositivos con el objetivo de simplificar la operación al usuario final. Un ejemplo típico es la filosofía Plug and Play. Aprovechando la capacidad actual de los sistemas embebidos, este trabajo presenta técnicas y algoritmos con el objetivo de sumar inteligencia al proceso de muestreo de una señal analógica. Empleando técnicas de sobremuestreo e interpolación adaptiva se logra analizar la señal, inferir estados, predecir comportamientos y validar la señal digitalizada. La señal a digitalizar es un proceso aleatorio, lo único que se conoce de antemano es su limitación en ancho de banda que le impide tomar valores totalmente arbitrarios entre muestras. Al aumentar la frecuencia de muestreo se restringe la aleatoriedad hasta quedar reducida a un espacio de escasas dimensiones que permite realizar un análisis certero sobre la señal. Se presentan resultados experimentales desarrollados en microcontroladores para validar los algoritmos propuestos.

Palabras Clave– Sensores Inteligentes, sobremuestreo, muestreo inteligente.

I. INTRODUCCIÓN

Los sistemas embebidos realizan sus decisiones en base a las señales de los sensores incluidos en él. Las señales provenientes de los sensores es la fuente de información disponible para cumplir el objetivo de diseño de un sistema de instrumentación y/o control. Estas señales proporcionan la abstracción del mundo real. Una señal es una entidad cuantificable de alguna forma de información. Más valiosa es la señal cuanto más información se encuentra embebida en ella. El concepto clave es la “información” que recibe el sistema. La Fig. 1 representa este concepto en forma gráfica. Esta información se encuentra inmersa en la señal, por lo tanto generalmente es necesario procesarla.

Aún en señales analógicas, no es necesario conocer el valor de la señal todo el tiempo. La máxima velocidad de cambio de una señal determina el intervalo de tiempo mínimo que se necesita conocer a la señal para poder inferir su comportamiento dentro él.

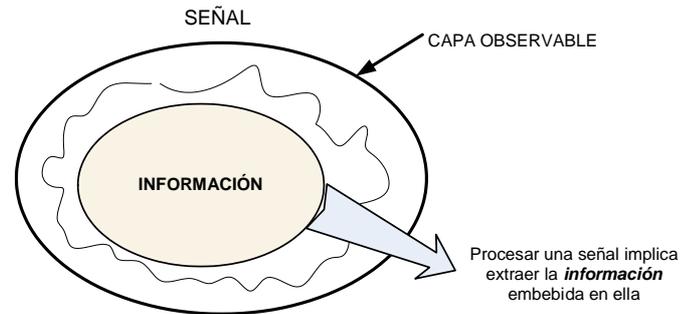


Figura 1. La información que transporta una señal generalmente se encuentra oculta. Procesar una señal implica extraer la información embebida en el núcleo de ella.

Si la señal en el mundo real es analógica, debe ser convertida en una representación digital que pueda ser entendida por el sistema embebido. La conversión de una señal en digital es realizada por el convertor analógico – digital que es el elemento principal de interfase entre el mundo real y el “mundo digital” como se observa en la Fig. 2. La forma más simple, pero no la única, de conversión es el muestreo uniforme. Otros tipos de muestreo incluyen muestreo por cruce de nivel [1] y muestreo no uniforme [2].

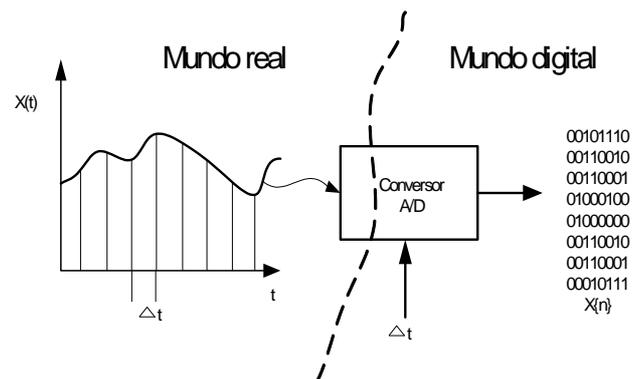


Figura 2. El convertor analógico digital es el elemento interfase que permite la representación de las señales reales en el mundo digital. En este caso,, muestreo uniforme, que transforma la señal en una *secuencia ordenada de números*.

En el caso de muestreo uniforme, a intervalos regulares Δt , se toman muestras de la señal y se convierten en un código correspondiente a un alfabeto finito, es decir una cantidad limitada de representaciones posibles. También el esquema más simple es el de dividir el rango dinámico del sensor en M divisiones de igual longitud, proceso que se conoce como cuantificación uniforme. Otros tipos de cuantificación se adaptan a la función densidad de probabilidad de la señal esperada. Es decir, para los valores más probables de la señal se disminuye la amplitud de la cuantificación.

El pasaje del mundo real al digital tiene dos particularidades importantes. Primero, la señal que existe para todo tiempo, es considerada solamente cada Δt segundos. Segundo la señal analógica, que es libre de tomar cualquier valor de amplitud, es forzada a una representación finita en el mundo digital, transformándose en una secuencia ordenada de números. El primer aspecto fue abordado por H. Nyquist y se conoce ampliamente como el teorema de muestreo. El teorema establece que no es necesario conocer la señal todo el tiempo para obtener una representación digital fidedigna siempre y cuando la señal posea un límite máximo de contenido espectral. El teorema determina que debemos tomar muestras a una frecuencia de por lo menos el doble de la máxima frecuencia presente en la señal. Es un límite teórico que necesita filtros reconstructores ideales para obtener sin error el valor de la señal entre las muestras. Si la señal se encuentra limitada en frecuencia hasta una f_{max} , las muestras, en un muestreo uniforme se deben realizar como mínimo a $2f_{max}$. Como resultado del proceso de muestreo uniforme se obtiene una secuencia ordenada de números que representan la señal, siempre y cuando se haya respetado el teorema de muestreo para evitar el "aliasing". Hasta aquí se ha descrito un proceso convencional de digitalización. La secuencia ordenada de números es la señal en el mundo digital a partir de la cual se deberán tomar todas las decisiones en el sistema.

Un sensor inteligente con capacidad de procesamiento embebida debería procesar el vector de muestras digitales con el fin de tomar conciencia de la señal que obtuvo.

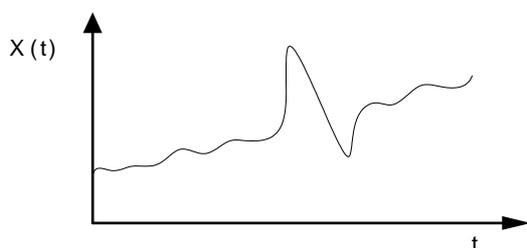


Figura 3. El muestreo inteligente debe "ver" a la señal completa, no solo una muestra a digitalizar.

Recordemos que el proceso de muestreo es el punto de conexión a través del cual se infiere el mundo real. En un sistema de control, todas las acciones estarán basadas en esta información. En la Fig. 3 se observa una señal típica proveniente de un sensor. **Un proceso de muestreo inteligente implica ver a la señal en su conjunto, no solo una muestra.** Si un ser humano observa la Fig. 3 podría sacar conclusiones tales como: la señal es ascendente, hay una forma tipo impulsiva que provoca un pico positivo y uno negativo. La señal proveniente del sensor es un proceso aleatorio. Los parámetros que se conocen previamente son la limitación en ancho de banda, impuesta por un filtro antialiasing, y su rango dinámico. En general, no se conoce la función densidad de probabilidad de la señal. La presencia de ruido, malfuncionamiento del elemento sensor o errores en la conversión deberían ser detectados. Las características deseables de un sensor inteligente con respecto a la señal que adquiere serían:

- Determinar la frecuencia de muestreo óptimo.
- Reconocer la forma de la señal.
- Predecir valores.
- Validar la señal.
- Marcar particularidades.
- Almacenar valores pasados en forma comprimida.
- Detectar patrones en forma automática y/o predeterminada.

En las secciones siguientes se desarrollan los algoritmos propuestos en busca de las anteriores características. Se mantiene el muestreo uniforme pero la señal se sobremuestra con respecto a la frecuencia de Nyquist. Se obtiene un vector redundante a partir del cual se disparan todos los algoritmos. El primero es una segmentación en tiempo real basada en interpolación lineal adaptiva. Luego los segmentos son etiquetados reflejando el comportamiento dentro de él. Con esta información se detectan secuencias de segmentos que identifican comportamientos globales. Estos tres procesos conforman la base para el análisis de la señal muestreada.

II. DESCRIPCIÓN DE LOS ALGORITMOS

De acuerdo al teorema de muestreo, debemos tomar dos muestras, como mínimo, en el tiempo correspondiente al periodo de la componente de frecuencia máxima de la señal. Si comenzamos a aumentar la frecuencia de muestreo, surgen dos preguntas: ¿Hasta que valor tiene sentido aumentarla? y

¿Que beneficios otorga este incremento? Las respuestas a estos interrogantes surgen del siguiente análisis.

Sea $x(t)$ la señal a ser muestreada, limitada en banda a f_{max} Hertz. Consideremos el caso particular que $x(t)$ es una señal sinusoidal de máxima amplitud y frecuencia.

$$x(t) = A \text{sen}(2\pi f_{max} t) \quad (1)$$

Donde A es la mitad de rango dinámico del conversor. Si transcurren Δt segundos tenemos:

$$x(t + \Delta t) = A \text{sen}(2\pi f_{max} (t + \Delta t)) \quad (2)$$

Por lo tanto:

$$x(t + \Delta t) - x(t) = A [\text{sen}(2\pi f_{max} (t + \Delta t)) - \text{sen}(2\pi f_{max} t)] \quad (3)$$

El lado izquierdo de (3) corresponde al cambio de la señal entre dos instantes de muestreo. Si fijamos este cambio a 1 bit menos significativo, un LSB, obtenemos:

$$\frac{2A}{2^N} = A [\text{sen}(2\pi f_{max} (t + \Delta t)) - \text{sen}(2\pi f_{max} t)] \quad (4)$$

Donde N es la cantidad de bits del conversor A/D. La máxima velocidad de cambio de la sinusoidal ocurre en los cruces por cero. Tomando $t=0$ para simplificar obtenemos de (4):

$$2^{-N+1} = \text{sen}(2\pi f_{max} \Delta t) \quad (5)$$

Dado que $1/\Delta t$ es la frecuencia de muestreo f_s , el argumento del seno es mucho menor que uno y teniendo en cuenta que $\text{sen}(x) \approx x$ para x pequeños obtenemos:

$$\frac{f_s}{f_{max}} = \pi 2^N \quad (6)$$

Por ejemplo, empleando (6), para N=8 bits obtenemos una frecuencia de muestreo de aproximadamente 400 veces la de Nyquist. Dada una resolución del conversor, (6) determina la máxima frecuencia de muestreo que tiene sentido aplicar ya que valores mayores darían como resultado la misma muestra. La ecuación (6) se determinó para una señal sinusoidal pura de máxima amplitud y frecuencia. Considerando una señal real y que el contenido espectral es en general decreciente con la frecuencia, un factor de 10% del máximo obtenido es razonable.

Entre los beneficios que brinda el sobremuestreo se encuentra una reducción efectiva del ruido de cuantificación, [3]. Es importante resaltar que la señal que podía tomar infinitos valores entre dos puntos en el mundo analógico, es forzada a un alfabeto finito. Más aun empleando la restricción del cambio de a solo un LSB, el espacio queda reducido a tres dimensiones, el mismo valor o mas menos un LSB.

Bajo condiciones de sobremuestreo se reduce la aleatoriedad de la señal. Visto de otra manera se genera redundancia entre muestras vecinas. Los algoritmos que describen el muestreo inteligente parten de los siguientes conceptos:

- No todas las muestras poseen la misma cantidad de información [4].
- En condiciones de sobremuestreo, las muestras pierden valor relativo si su valor puede ser obtenido mediante interpolación de sus muestras vecinas.

Las muestras que no pueden obtenidas mediante una combinación lineal de sus vecinas, transportan en algún sentido más información y las llamaremos muestras esenciales. Las muestras esenciales marcan límites naturales de segmentos de la señal con un comportamiento uniforme.

En [5] se describen ampliamente los algoritmos propuestos y aquí presentaremos solamente un resumen. El algoritmo de segmentación comienza interpolando la muestra 2 de 1 y 3. Se calcula el error como la diferencia entre la muestra 2 interpolada y la real. Si el error es menor que una cota de error, la muestra 2 no transporta información ya que su valor puede ser calculado mediante interpolación lineal de sus muestras vecinas. El próximo paso es interpolar la muestra 3 mediante las muestras 1 y 5 y así siguiendo. Cuando se supera el error de interpolación o un error acumulativo, se culmina un segmento determinado en sus extremos por dos muestras esenciales. Si la cota de error no se supera por un número determinado de muestras se culmina el segmento a ese máximo de muestras. De esta segmentación, que se ejecuta en tiempo real, se obtienen dos vectores que denominamos $mark(n)$ y $timepos(n)$ que contienen los valores de las muestras esenciales y los instantes de ocurrencia respectivamente.

En la Fig. 4 se observa el esquema del proceso completo de tratamiento de la señal sobremuestreada. Un tercer vector $clases(n)$ determina el comportamiento de la señal dentro del segmento, estableciendo ocho clases de comportamiento denominadas: a,b,c,d,e,f,g,h. Se basa en la diferencia entre las muestras interpoladas y las muestras reales. Por ejemplo si dentro de un segmento las muestras interpoladas exceden las reales, es un indicador de la forma de la señal dentro del segmento, como es el caso de los segmentos tipo "f" o "d". En la Fig. 5. se observan las ocho clases. Los segmentos "a", "b" y "c" son consecuencia de no haber alcanzado la cota

de error y el segmento se termina por alcanzar el límite máximo de segmento.

comportamiento determinado de la señal muestreada. En la Fig. 6 se observa una señal ejemplo. Considerando solamente el vector clases obtenido podemos inferir el comportamiento de la señal a través de los siguientes patrones:

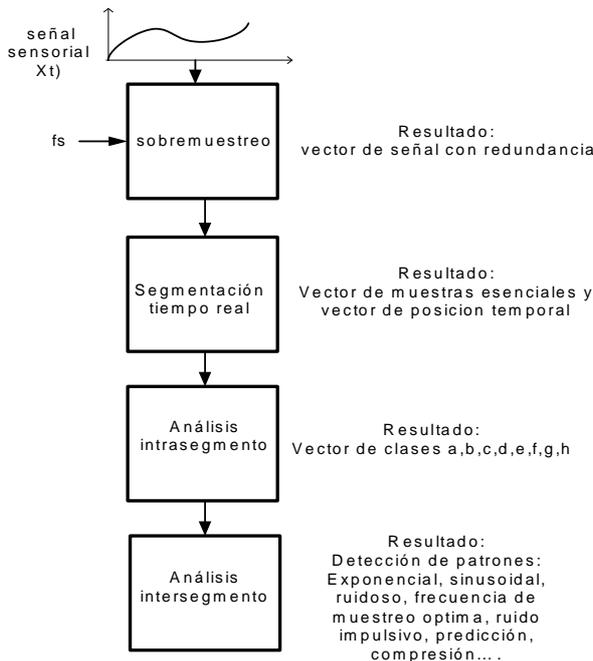


Figura 4. Esquema del proceso de análisis de la señal sobremuestreada.

- En la primera porción de la señal hay un crecimiento exponencial dado por la secuencia de segmentos “g”.
- Se alcanza un máximo que ocurre en la unión de segmentos “g” y “e”.
- A partir del máximo, la señal comienza a caer dada la ocurrencia de los segmentos “e”.
- Hay un cambio de tendencia dada la ocurrencia de un segmento “h”, luego del cual la señal decae en forma exponencial para alcanzar un mínimo que será localizado con precisión en la unión de los segmentos “f” y “d”.

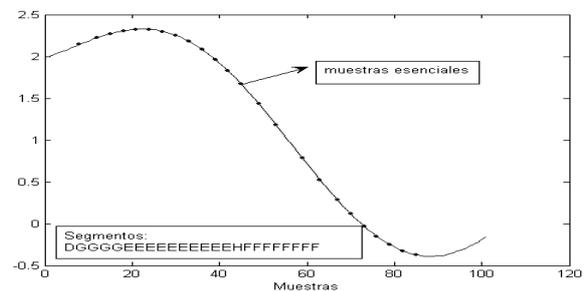


Figura 6. Vector clases sobre una señal testigo.

a	— — — — —	constante
b	↗	lineal +
c	↘	lineal (-)
d	↗ (curved)	+exp+
e	↘ (curved)	(-)exp+
f	↗ (curved)	c+exp(-)
g	↘ (curved)	c-exp(-)
h	~	ruidoso

Figura 5. Clasificación de segmentos en ocho clases.

A. Análisis inter-segmento

La información proporcionada por los tres vectores $mark(n)$, $timepos(n)$ y $clases(n)$ proporciona una plataforma de análisis para inferir comportamientos de la señal muestreada. Es importante recalcar que los tres vectores, en una estructura de de buffer circular, se obtienen en tiempo real a medida que ingresan las muestras. El patrón de secuencia de clases de segmentos especifica un

Mediante estas simples observaciones, el sistema embebido es conciente de la señal que esta digitalizando, prácticamente en tiempo real.

Es importante remarcar que si el error de interpolación tiende a un LSB, la probabilidad de ocurrencia de los segmentos “a”, “b”, “c” y “h” tienden a cero. Bajo esta condición, y con la señal suficientemente sobremuestreada, el comportamiento de la señal queda especificado solamente por los cuatro segmentos “d”, “e”, “f” y “g”.

El análisis del comportamiento macroscópico de la señal es posible mediante las secuencias de patrones de clases de segmentos. Por ejemplo, la ocurrencia consecutiva de segmentos tipo “g” indica que la señal se esta estabilizando a un valor de estado estacionario que podría incluso predecirse. Esta predicción es analizada en la siguiente sección.

Mediante el análisis de la cantidad de muestras de los segmentos obtenidos, el sistema embebido podría ajustar él mismo la frecuencia de muestreo. La frecuencia de muestreo óptima para una porción de señal es la frecuencia de sobremuestreo dividida por la cantidad de muestras del segmento más corto en dicha porción de la señal. La longitud de los segmentos es un parámetro importante ya que determina como se aparta la señal de un comportamiento lineal. La Fig 7. Detalla la detección de algunos comportamientos macroscópicos.

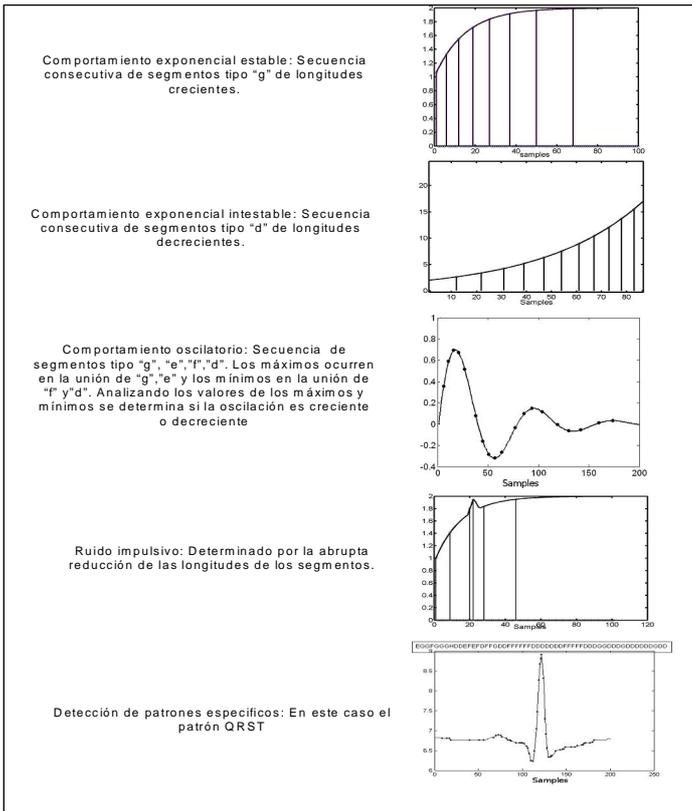


Figura 7. Detección de comportamiento macroscopico mediante analisis inter segmento.

III. RESULTADOS EXPERIMENTALES

Los algoritmos de segmentación y clasificación fueron implementados sobre microcontroladores. Se realizó un sensor de temperatura inteligente de bajo costo cuyo circuito se observa en la Fig 8. El microcontrolador es un 12F683 de Microchip que posee conversor A/D de 10 bits. La sonda de temperatura esta formada por tres diodos 1N4148 y posee interfase serie RS232 optoacoplada. El costo del sensor es inferior a los 4 USD.

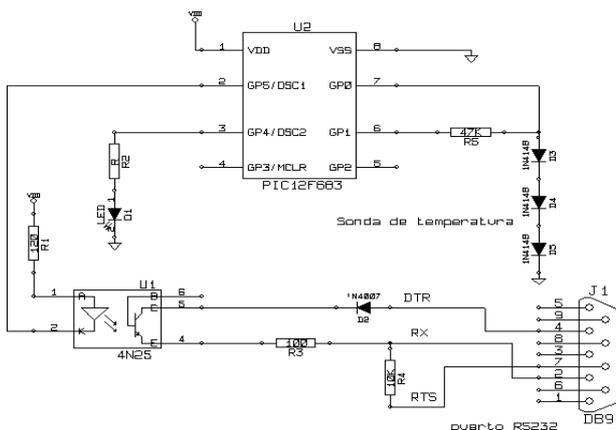


Figura 8 Implementación de un sensor inteligente de temperatura de bajo costo.

El sensor reconoce la forma que obtiene en el muestreo y cuando detecta tres segmentos consecutivos tipo “g” o tres segmentos consecutivos tipo “f” realiza el análisis de predicción empleando las muestras esenciales y la pendiente en esos instantes. Este tipo de predicción es importante en sistemas de control [6].

La expresión general de exponencial en la Fig. 9 es:

$$x(t) = X_{ss} - (X_{ss} - X_i) e^{-t/T} \quad (7)$$

Donde X_{ss} es el valor en estado estacionario y X_i el valor inicial. La derivada de $x(t)$ respecto de en $t=0$ es:

$$\left. \frac{dx}{dt} \right|_{t=0} = \frac{X_{ss} - X_i}{T} \quad (8)$$

La pendiente de los segmentos puede emplearse como la aproximación de la derivada en (8) aplicada en los puntos **a** y **b**. Si consideramos la derivada en el punto **a** y empleando este punto como valor inicial obtenemos:

$$\frac{X_{ss} - X_{ia}}{T} = Ma \quad (9)$$

Donde $Ma = Y_a / X_a$ es la aproximación de la pendiente en el punto **a**. Repitiendo el análisis ahora en el punto **b**, y empleando este punto como valor inicial obtenemos:

$$\frac{X_{ss} - X_{ib}}{T} = Mb \quad (10)$$

Donde $Mb = Y_b / X_b$ es ahora la aproximación de la pendiente en el punto **b**. Resolviendo (9) y (10) obtenemos (11).

$$X_{ss} = \frac{Ma X_{ib} - Mb X_{ia}}{Ma - Mb} \quad (11)$$

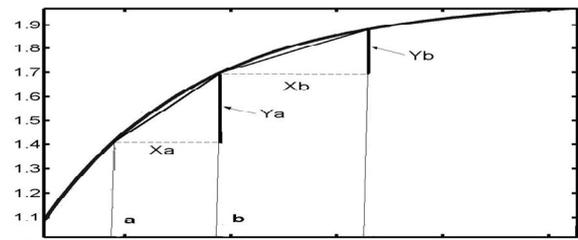


Figura 9 Predicción del valor estacionario de la señal basada en los cambios de pedndiente en los segmentos tipo “g”.

En la Fig. 10 se observa el algoritmo de segmentación y clasificación de segmento implementado en lenguaje “C” de CCS inc sobre el 12F683. Se diseño siguiendo la filosofía de programación Estado-Cooperativa. Por limitaciones de memoria RAM, 128 bytes, se limitó la longitud del segmento a 48 muestras.

Un microcontrolador adecuado para estas técnicas es el dsPIC33FJ12GP de Microchip, el cual fue diseñado para aplicaciones de sensores inteligentes, permitiendo digitalizar a 1.1 millones de muestras por segundo.

IV CONCLUSIONES

La potencia de los sistemas embebidos actuales ha cambiado no solo la topología de nuestros diseños, sino además la forma de concebirlos. Algo que en el pasado era inviable pasa a ser factible para luego transformarse en natural.

Naturalmente los sistemas embebidos deben muestrear en forma inteligente. Una propuesta de normalización de estos algoritmos fue presentada en [7]. El intercambio de información mediante un lenguaje común entre sensores de un sistema permite el aumento de confiabilidad, la predicción anticipada de estados anormales, la detección de patrones normales y anormales y el aprendizaje de patrones en forma automática.

Los algoritmos presentados son simples pero muy efectivos. Capturan la esencia de la señal muestreada, y representan la información presente en ella. Si el error de interpolación tiende a un LSB los cuatro segmentos “d”, “e”, “f” y “g” codifican la señal como si fuera el código genético de ella.

REFERENCIAS

- [1] N. Sayiner, H.V. Sorensen and T.R. Viswanathan, “A Level-Crossing Sampling Scheme for A/D Conversion”, IEEE Transactions on Circuits and Systems II, vol. 43, pp. 335-339, April 1996.
- [2] R. Haddad, T Parsons. “Digital Signal Processing, Theory, Applications and Hardware”. *Computer Science Press* 1991., pp 32-36.
- [3] J Murthy Madapura “Achieving Higher ADC Resolution Using Oversampling” AN1152, Microchip Technology Inc. 2008.
- [4] Marten D. van der Laan. “Signal sampling techniques for data acquisition in process Control”. Thesis Rijksuniversiteit Groningen. - 1995. ISBN 90-367-0502-9.
- [5] Monte, G. “Sensor Signal Preprocessing Techniques for Analysis and Prediction” Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE. pp 1788-1793. ISBN 978-114244-1766-7.
- [6] F. Bertling, S. Soter, “Real-time prediction of the steady state temperature of circuit components as a tool for power electronic circuit testing”. *PCIM Europe 2007 Conference, Nuremberg, Germany*.
- [7] Monte, G. “A proposal of a New Standard For Sensor Signal Analysis”. Industrial Technology (ICIT), 2010. Chile. IEEE International Conference. pp 1617-1622. ISBN 978-1-4244-5697-0/10.

```

//*****
void segmentacion()
{
  if(!bit_test(st_segmentacion,0))
  // inicializo la segmentacion
  senial[0]=convertir_ad();
  k=0;
  j=1; // es el indice para la muestra real que es interpolada
  cont_muestreo=0, // se incrementa en timer1 cada 40 milisegundos
  arriba=0;
  abajo=0;
  igual=0;
  bit_set(st_segmentacion,0); // siguiente estado
  return;
}
if(!bit_test(st_segmentacion,1))
{
  if(cont_muestreo>T_sampling) // pasaron xx ms?
  {
    k++;
    senial[k]=convertir_ad();
    bit_set(st_segmentacion,1); // siguiente estado
    cont_muestreo=0;
  }
  return;
}
if(!bit_test(st_segmentacion,2))
{
  if(cont_muestreo>T_sampling) // pasaron xx ms?
  {
    k++;
    if (k==48) // termino el segmento por limite sin superar el error de interpolacion
    {
      if (senial[0]<senial[k-1]) // miro los extremos para determinar clases A, B o C
        clase='b';
      if (senial[0]>senial[k-1])
        clase='c';
      if (senial[0]==senial[k-1])
        clase='a';
      muestra_izq=senial[0];
      muestra_der=senial[k-1];
      tiempos=k-1;
      bit_set(st_segmentacion,2); // siguiente estado paso los datos a la tarea clases
      return;
    }
    senial[k]=convertir_ad();
    dif=(int16)(senial[0]/2)+(int16)(senial[k]/2)-(int16)senial[j]; // tengo las muestras 1,2 y 3
    j++;
    if (dif>0) // la interpolacion esta arriba de la real -> candidato para
      arriba++; // segmento F o D
    if (dif<0) // la interpolacion esta abajo de la real -> candidato para
      abajo++; // segmento E o G
    if (dif==0) // la interpolacion es perfecta -> candidato para
      igual++; // segmento A, B o C

    if (dif< cota_error)
    {
      cont_muestreo=0;
      bit_clear(st_segmentacion,1); // vuelvo al estado anterior
    }
    else // termino el segmento por cota de error
    {
      muestra_izq=senial[0];
      muestra_der=senial[k]; // es la muestra que produjo el error muestra esencial
      if(arriba>abajo && muestra_izq>muestra_der) // es segmento tipo F
        clase='f';
      else if(arriba>abajo && muestra_izq<muestra_der) // es segmento tipo D
        clase='d';
      else if(abajo>arriba && muestra_izq>muestra_der) // es segmento tipo E
        clase='e';
      else if(abajo>arriba && muestra_izq<muestra_der) // es segmento tipo G
        clase='g';
      else
        clase='h';
      bit_set(st_segmentacion,2); // siguiente estado paso los datos a la tarea clases
      return;
    }
  }
  return;
}
if(!bit_test(st_segmentacion,3))
{
  long_segmento=k;
  st_segmentacion=0; // largo otra vez la tarea
  bit_set(st_prediccion,0); // hab para que entre a la tarea prediccion
  return;
}
}
}

```

Figura 10 Algoritmo de segmentación y clasificación de segmento implementado en el microcontrolador 12F683.