

Medición indirecta del nivel de líquido en tanques de almacenamiento

Melchor Nuñez Guerrero, Gustavo Cerda Villafañá¹

¹Universidad de Guanajuato
gcerdav@ugto.mx¹

Resumen

La medición del nivel de líquidos en tanques de almacenamiento se realiza mediante una amplia gama de técnicas, desde visuales como tubos de vidrio externos, hasta sensores ultrasónicos y de radio frecuencia. Pero cuando se quiere automatizar la medición, es obligatorio el uso de sensores. Y, cuando se quiere medir nivel de líquidos corrosivos, se requieren medidas especiales, como los materiales con que son fabricados dichos sensores o sensores que no entren en contacto con los líquidos o atmósferas dentro de los tanques, como algunos sensores radioactivos que incrementan el costo de la automatización. Con la propuesta que se presenta en este trabajo se busca evitar el uso de sensores dentro del tanque de almacenamiento y que sea una opción simple y económica de implementar. El objetivo es determinar si se puede medir la variación en corriente por parte de la bomba que se utiliza para extraer el líquido del tanque de almacenamiento y, utilizar dicha variación para determinar el nivel de líquido que se encuentra dentro del tanque. Es un método indirecto y debe estar sujeto a tres condiciones principales: conocer el flujo con el que se extrae el líquido por parte de la bomba, que dicho flujo sea uniforme y que sea extraído en un punto donde las condiciones de salida sean idénticas en todo momento. Cumpliéndose dichas condiciones se puede utilizar el método que se describe en este trabajo.

Palabras clave: tanque de almacenamiento; sensor de nivel, automatización.

Introducción

La necesidad de medir el nivel de líquidos en tanques de almacenamiento ha llevado al desarrollo de diversas técnicas ya probadas y en operación en todo el mundo [1 - 5]. La medición se realiza de forma directa mediante sensores [1, 2], mediante técnicas que buscan reducir el ruido presente en las mediciones [3] o de forma indirecta como en el procesamiento de imágenes [5]. El objetivo de este trabajo es determinar la posibilidad de desarrollar un método alternativo, utilizando la medición de la corriente consumida por el motor de la bomba utilizada para extraer el líquido de un tanque y, mediante la variación de la misma, determinar el nivel del líquido en el interior del mismo.

Metodología

Para analizar la viabilidad del método que se interesa desarrollar, primero se considera que una bomba va a extraer el líquido de un tanque de almacenamiento y lo va a llevar a una altura determinada donde el líquido será extraído a una atmósfera de presión constante, como lo muestra la Figura 1. La diferencia entre la altura a la que se encuentra el líquido y la altura a la que será llevado el mismo Δh variará con el tiempo, conforme se vaya vaciando el líquido del tanque. Esta variación será utilizada para relacionarla con el incremento en el consumo de corriente por parte del motor de la bomba de extracción.

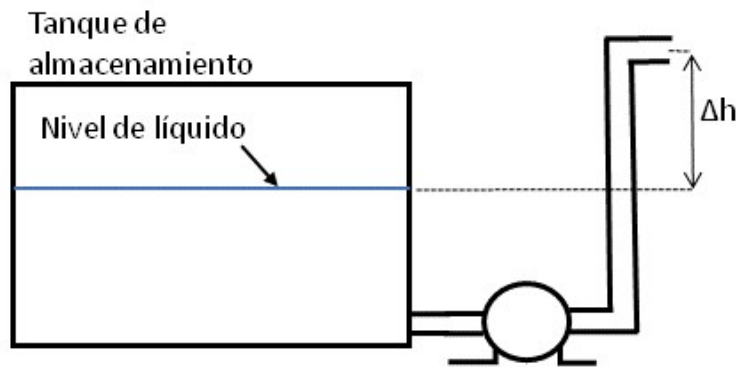


Figura 1. Diagrama del trabajo de una bomba de descarga en un tanque de almacenamiento.

El desarrollo de este trabajo comienza con el análisis matemático para determinar la relación entre el consumo de corriente y la variación en el nivel del líquido dentro de un tanque de almacenamiento. Para ello se consideran las siguientes variables:

- E Energía
- ρ Densidad del líquido (kilogramos por metro cúbico (kg/m³))
- V Volumen de líquido desplazado
- g Aceleración debido a la gravedad de la tierra (9.8 metros por segundo al cuadrado (m/s²))
- h Cambio neto en altura.

La energía necesaria para extraer una cierta cantidad de líquido a una altura específica es

$$E_p = \rho V g h$$

La variación en energía para extraer cierta cantidad de líquido de dos alturas diferentes es:

$$\Delta E_p = \rho V g h_2 - \rho g h_1$$

ó

$$\Delta E_p = \rho g V (h_2 - h_1)$$

Por otro lado, la potencia utilizada por el motor de la bomba se calcula mediante la siguiente ecuación:

$$P = EI$$

donde

- E Voltaje (Volts)
- I Corriente (Amperes)
- P Potencia (Watts)

Sabiendo que la potencia está relacionada con la energía mediante la relación $P = E/t$, entonces se puede expresar la relación $(h_2 - h_1)$ mediante el incremento de la corriente utilizando la siguiente ecuación:

$$(h_2 - h_1) = \frac{EIt}{\rho gV}$$

ó

$$(h_2 - h_1) = \frac{EI}{\rho g V/t} = \frac{EI}{\rho g Q}$$

Si la densidad, el voltaje y el flujo se consideran constantes, entonces la variación en $(h_2 - h_1)$ generará una variación en el consumo de corriente y, si es posible determinar esa variación de la corriente, entonces se podrá determinar la variación de la altura del líquido dentro del tanque de almacenamiento.

El análisis dimensional muestra que:

$$\frac{EI}{\rho g Q} = \frac{\text{vatio}}{\frac{\text{kg m m}^3}{\text{m}^3 \text{s}^2 \text{ s}}} = \frac{\frac{\text{Joule}}{\text{s}}}{\frac{\text{kg m}}{\text{s}^3}} = \frac{\frac{\text{N m}}{\text{s}}}{\frac{\text{kg m}}{\text{s}^3}} = \frac{\frac{\text{kg m/s}^2 \text{ m}}{\text{s}}}{\frac{\text{kg m}}{\text{s}^3}} = \text{m}$$

Para realizar las mediciones de la corriente en una bomba de extracción de líquidos se desarrolló un sistema basado en un microcontrolador que tomara las muestras provenientes de un sensor de corriente y las guardara en una memoria SD. La tarjeta de desarrollo utilizada se muestra en la figura 2.

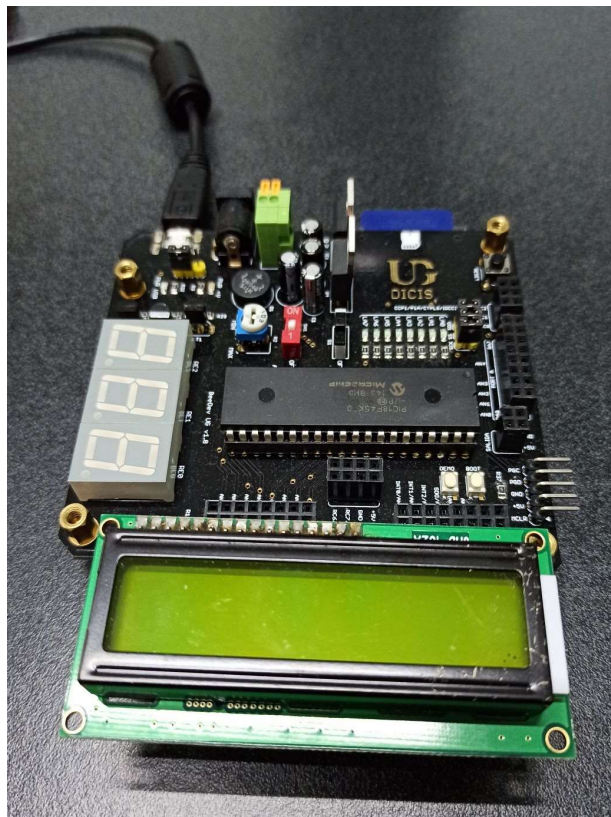


Figura 2. Tarjeta de desarrollo utilizada en el proyecto.

El sensor utilizado para medir la corriente consumida por la bomba se muestra en la figura 3.



Figura 3. Sensor de corriente.

Los componentes electrónicos utilizados tanto para la simulación como para la implementación son principalmente una pantalla LCD, una memoria SD, el sensor de corriente, un amplificador operacional para acondicionar la señal proveniente del sensor y la tarjeta de desarrollo basada en el microcontrolador PIC18F45K50.

El código implementado consta de 14 archivos de los cuales seis son archivos con extensión “.c” y ocho son archivos con extensión “.h”. La pantalla LCD utiliza para su funcionamiento el archivo “lcd_lib.h”, la memoria SD utiliza los archivos “diskio.h”, “diskio.c”, “ff.h”, “ff.c”, “ffconf.h” e “integer.h”, por último, para la configuración y funcionamiento del microcontrolador 18F45K50 se utilizan los archivos “mcc.h”, “mcc.c”, “pin_manager.h”, “pin_manager.c”, “spi1.h”, “spi1.c” y el archivo principal “main.c”.

Descripción de archivos/código implementado:

- lcd_lib.h

Esta librería realiza la función de configurar los pines del puerto D conectados a la pantalla LCD y también contiene funciones que sirven para inicializar la pantalla LCD, limpiar el contenido de la pantalla, posicionar el cursor, escribir en la pantalla ya sea un dato tipo “string” o solo un dato tipo “char”. Todas estas funciones mencionadas son importantes para que por medio de la pantalla LCD estemos informados del proceso de muestreo de datos.

- diskio.h

El contenido de este archivo está orientado al módulo de interfaz del disco de bajo nivel, el cual consta de configuración y definición de variables y estructuras, definición de prototipos para funciones de control del disco que serán llamadas por el archivo “ff.c” para ser utilizadas por otras funciones.

- diskio.c

Este archivo representa el módulo de control SD en modo SPI el cual contiene principalmente todos los prototipos declarados en el archivo “diskio.h”. Estas funciones son muy utilizadas debido a la comunicación SPI, son funciones un poco extensas que en algunos casos requieren de diversos datos para realizar sus operaciones y también internamente mandan llamar a otras funciones lo que las vuelve más extensas.

- ff.h

Esta librería es utilizada para poder manipular la memoria SD en el modo SPI, ya que este archivo es el módulo de sistema de archivo FAT. Este archivo contiene declaración de estructura para objeto de sistema de archivos, para objeto de archivo, para objeto de directorio, para objeto de información de archivo, definición de valor de retorno de funciones, definición de prototipos para la interfaz de aplicación del módulo FatFs y definición de valor de banderas para variables de proceso.

- ff.c

Este archivo es el más extenso de todos, también desde el punto de vista de elementos de programación es el más complejo. Utiliza herramientas de programación hacen que este archivo sea el más complejo y difícil de implementar. Este código contiene primeramente definición y redefinición de variables para la respuesta de las funciones, también definición de banderas para el estado del proceso. Contiene también funciones privadas que son utilizadas por las funciones públicas. Las funciones privadas abarcan la mitad del contenido del código y algunas de estas no están habilitadas debido a la configuración elegida. Las funciones públicas que se encuentran definidas en el archivo "ff.h" abarcan la segunda mitad del código aunque algunas de ellas no están habilitadas.

- ffconf.h

El contenido de este archivo únicamente es configuración del módulo del sistema de archivos FAT. Con esta configuración se habilitan o deshabilitan funciones dentro del archivo "ff.c" de acuerdo con las necesidades del proyecto, por ejemplo, si únicamente se quiere utilizar la memoria SD para lectura se puede seleccionar que así sea.

- integer.h

Este código es exclusivamente para definir la longitud en bits de algunos tipos de variables utilizadas en el módulo FatFs.

- mcc.h

Este código contiene los prototipos de las funciones para la configuración del microcontrolador y la selección de la frecuencia del microcontrolador.

- mcc.c

El contenido de este código es la función de inicialización del sistema la cual manda llamar a otras funciones que configuran el microcontrolador y también este código contiene la configuración del oscilador con la definición de sus respectivos registros.

- pin_manager.h

El contenido del código es redefinición de variables para que su escritura sea más intuitiva, y contiene también el prototipo para inicializar los pines del microcontrolador.

- pin_manager.c

Este archivo únicamente contiene la configuración de los pines, donde se establece el valor de cada puerto, si son pines de entrada o de salida, se habilitan y deshabilitan los canales analógicos, se habilita el weak pull up del puerto B y se configura los registros para el ADC.

- spi1.h

Este archivo solo contiene los prototipos de las funciones para el módulo SPI.

- spi1.c

El contenido del archivo son la configuración de los registros para el módulo SPI y funciones que son utilizadas para indicar que el buffer está lleno, si ocurrió una colisión, para limpiar el estatus de colisión y para intercambiar datos con la memoria SD.

- main.c

Este archivo contiene la configuración del microcontrolador, dos funciones secundarias, una utilizada para crear retardos y otra utilizada para realizar la conversión del ADC. Dentro de la función main las operaciones principales son primero revisar por medio de la función f_mount que la tarjeta SD está montada por medio del siguiente código

```
if (f_mount(&FatFs, "", 1) != FR_OK) {      /* Montaje SD */
Lcd_Set_Cursor(1,1);
Lcd_Write_String("Inicio: ERROR");
    __delay_sec(2);

    while(f_mount(&FatFs, "", 1) != FR_OK) {      //Permanece en el while
        ;
    }
}
```

Si una tarjeta SD está montada, la función f_mount regresara el valor de FR_OK lo que hace que las operaciones dentro del condicional if no se ejecuten y continua a la siguiente operación. En caso de que no esté montada la tarjeta SD, se indicara por medio de la pantalla LCD el estado de "Inicio: ERROR"

Después de revisar que la tarjeta SD está montada se abre o crea un archivo en la tarjeta SD por medio de la función f_open y se posiciona el cursor en la parte final del archivo para comenzar a escribir en este los datos de las mediciones muestreadas por medio de siguiente código

```
f_write(&Fil, "En este archivo TXT se guardarán los datos. \r\n", 46, &bw); /*Escribe*/
Lcd_Set_Cursor(1,1);
Lcd_Write_String("Archivo EDITADO");
    __delay_sec(2);

    for(int i = 0; i<500; i++){      //Realiza un ciclo de 500 iteraciones
        dato = Conversion();      //Guarda el dato recibido por el canal analogico AN0
        sprintf(sdato, "%i", dato); //Guarda la medicion del ADC en un string
        f_write(&Fil, sdato, 4, &bw); /* ]Escribe en el archivo el dato muestreado*/
        f_write(&Fil, "\n", 1, &bw); /* ]Escribe en el archivo un salto de linea*/
    }
}
```

Se escribe una primera línea para indicar el inicio de la escritura, para despues guardar 500 mediciones dando un salto de línea despues de cada conversión.

Para terminar con la función f_close para cerrar el archivo e indicar en la pantalla LCD el estado de "Extraer SD".

Resultados

Las pruebas se realizaron utilizando una bomba de agua que sube el líquido de una cisterna de 1.5m de diámetro por 1.6 m de profundidad y 2500 litros de capacidad. Por el tiempo que tarda en llenar un tinaco que está a 3.3 m de altura se calculó un flujo de 32 l/min. Las mediciones que se obtuvieron mediante el sistema implementado se capturaron con mucho ruido y se presentan en la figura 4. En la parte superior de la imagen estás los valores de las mediciones sin filtro y en la parte inferior después de filtrarlas utilizando un filtro IIR Butterworth implementado en Matlab. En estas gráficas se muestra el valor del amperaje medido con relación a las muestras realizadas. Se realizaron tres conjuntos de mediciones diferentes, cada una de 1500 muestras de duración y 10 minutos de diferencia entre ellas. Las muestra se realizaron cada 33 microsegundos aproximadamente.

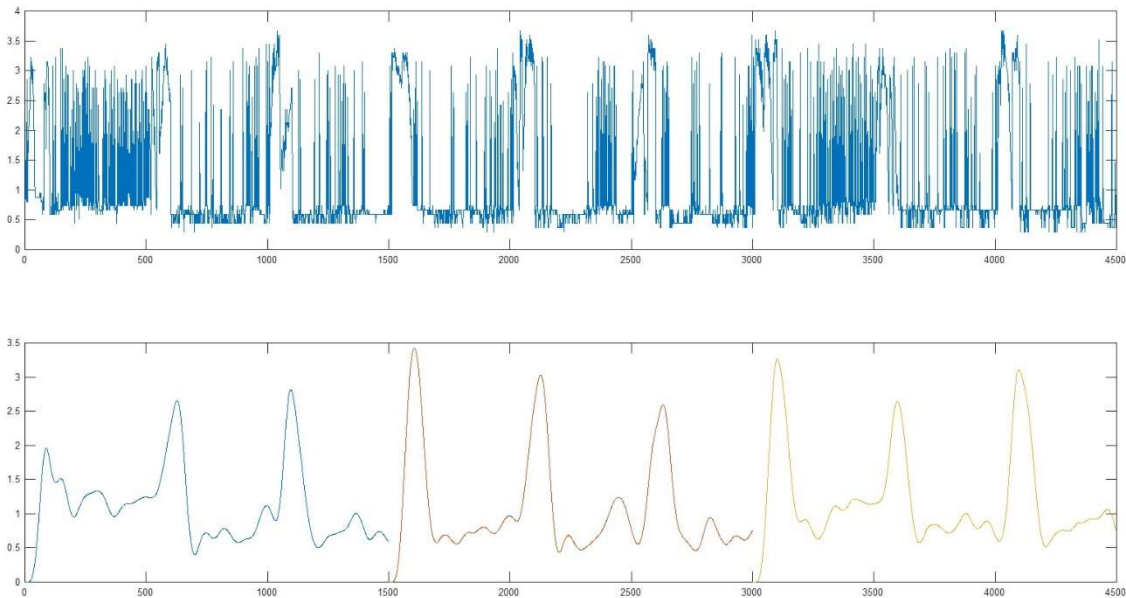


Figura 4. Parte superior, datos crudos. Parte inferior, datos filtrados

Los valores rms de corriente obtenidos de las muestras filtradas utilizando la función rms() de Matlab para cada conjunto de datos se presentan en la tabla 1.

Tabla 1. valores rms de cada conjunto de muestras

Conjunto de datos	Valor rms (amperes)
Conjunto 1	1.2298
Conjunto 2	1.3018
Conjunto 3	1.3781

Utilizando la ecuación obtenida con los siguientes valores:

$E = 110V$

$\rho = 1000kg/m^3$

$g = 9.81m/s^2$

$Q = 32 \text{ l/min} = 0.0002$

Con estos valores se obtiene una Δh entre el primer conjunto de muestras y el siguiente de 1.51m y entre el segundo y el tercer conjunto de muestras de 1.6m, pero las mediciones hechas directamente en la cisterna indicaron una reducción del nivel del agua de 25 cm entre cada conjunto de muestras. Esta disparidad se analizará para determinar la causa. Hay varios factores que se deben considerar: la calibración del sensor, el ajuste del amplificador operacional que se utilizó para agregar un offset a la señal de voltaje que proporciona el sensor de corriente y el ruido introducido en la señal.

Bibliografía/Referencias

- [1] Ameen, O. F., Younus, M. H., Aziz, M. S., & Raja Ibrahim, R. (2016). TEMPERATURE AND WATER LEVEL MEASUREMENT OF LIQUID IN A TANK USING FIBER BRAGG GRATING. *Jurnal Teknologi*, 78(3). <https://doi.org/10.11113/jt.v78.7529>
- [2] D. Brumbi, "Measuring process and storage tank level with radar technology" *Proceedings International Radar Conference*, Alexandria, VA, USA, 1995, pp. 256-260, doi: 10.1109/RADAR.1995.522555.
- [3] D. Borg, G. Serpa Sestito, M. Martins da Silva, "Machine-learning classification of environmental conditions inside a tank by analyzing radar curves in industrial level measurements" *Flow Measurement and Instrumentation*, v. 79, p. 1-9, 2021
- [4] K. Loizou, E. Koutroulis, D. Zalikas and G. Liontas, "A low-cost capacitive sensor for water level monitoring in large-scale storage tanks," *2015 IEEE International Conference on Industrial Technology (ICIT)*, Seville, Spain, 2015, pp. 1416-1421, doi: 10.1109/ICIT.2015.7125295.
- [5] C. Zhao and Y. Chen, "A New Liquid Level Measuring System of Standard Metal Tank Based on Sub-pixel Edge Detection," *2007 IEEE International Conference on Control and Automation*, Guangzhou, China, 2007, pp. 3207-3212, doi: 10.1109/ICCA.2007.4376954.