



Sistema de Teleoperación a través de WiFi basado en Mediciones Inerciales y Filtro de Kalman

Bernardo Aceituno-C, 12-10764, y Luis Pérez-B, 10-10545

30 de Noviembre de 2016

Índice

1. Introducción	4
1.1. Teleoperación	4
1.2. El filtro de Kalman	5
1.3. IEEE 802.11 WiFi	6
2. Implementación	8
2.1. Adquisición	8
2.2. Procesamiento	10
2.3. Transmisión de Datos	12
2.4. Interfaz Gráfica	14
3. Resultados	16
3.1. Adquisición y Procesamiento	16
3.2. Transmisión	20
3.3. Teleoperación	22
4. Consideraciones Finales	23

Índice de figuras

1.	Sistema de teleoperación Maestro-Esclavo	4
2.	<i>WiFi</i> trademark	6
3.	Estructura del sistema	8
4.	diagrama de tiempos del sensor por ultrasonido	9
5.	diagrama de estados del sistema	12
6.	Architectura de WiFi como BSS	13
7.	pines expuestos del ESP8266-12	13
8.	pines expuestos del demoQE128	14
9.	Manipulador SCARA a simular	14
10.	circuitería final	16
11.	Posición inicial de Roll y Pitch	16
12.	Variaciones en Roll (1)	17
13.	Variaciones en Roll (2)	17
14.	Variaciones en Roll	17
15.	Variaciones completas en Roll	18
16.	Variaciones en Pitch (1)	18
17.	Variaciones en Pitch (2)	18
18.	Variaciones en Pitch - Ángulo Pitch	19
19.	Variaciones completas en Pitch	19
20.	Perturbaciones en el Ángulo Roll	19
21.	Perturbaciones en el Ángulo Pitch	20
22.	Verificando el énlace con la red	20
23.	Seleccionando interfaz de wlan	20
24.	Seleccionando protocolo y modo de análisis de red	21
25.	Paquetes UDP llegando a la interfaz de red	21
26.	Verificación de los valores transmitidos	21
27.	Teleoperación de un robot scara usando Drake y el sistema de teleoperaciones integrado	22

1. Introducción

1.1. Teleoperación

Hay situaciones en las cuales no existe la posibilidad de manipulación humana debido a la dificultad de acceso al medio o la necesidad de operación remota. Bajo estas situaciones surgen las alternativas de dispositivos teleoperados, controlados por el usuario en una estación remota. Estos sistemas acarrean beneficios desde el punto de vista de seguridad del usuario, evitando trabajos en ambientes inseguros.

Una plataforma de teleoperación típica consta de 3 componentes: estación de mando, sistema de comunicación y una o varias estaciones esclavas. Esta última consta de un robot móvil equipado con un manipulador en un entorno remoto. La estación de mando interactúa directamente con el usuario y es donde se encuentran la mayor cantidad de alternativas de diseño. El sistema de comunicación puede ser a través de redes de computadores o enlaces de RF, tomando en consideración elementos como latencia y consumo de energía. Un esquema de teleoperación básico es presentado en la figura 1.1

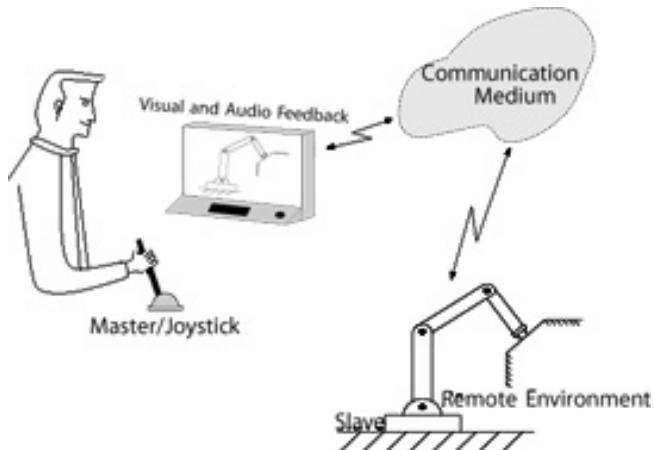


Figura 1: Sistema de teleoperación Maestro-Esclavo

Un ejemplo en concreto es la interfaz de operación *Director* [3] desarrollado por el MIT para el DARPA Robotics Challenge (DRC), desarrollado para navegar mediante *autonomía compartida* un complejo robot humanoide sobre un ambiente complejo, basado en fusión sensorial para la estimación de estado.

1.2. El filtro de Kalman

El filtro de Kalman es un algoritmo de estimación de estado desarrollado por Rudolf E. Kalman en 1960 que permite identificar el estado de un sistema dinámico lineal cuando el sistema está sometido a ruido blanco aditivo (AWGN). Este algoritmo se basa en dos etapas principales:

1. Predicción: Predice el estado del sistema en base a su modelo dinámico (lineal) y las entradas que este recibe del usuario.
2. Corrección: Toma una medida del estado y su respectivo error y de allí calcula el valor estimado.

Entonces dado un sistema lineal y discreto en el tiempo representado mediante la siguiente ecuación de espacio-stado:

$$x_{k+1} = A_k x_k + B_k u_k + w_k$$

$$z_k = H_k x_k + v_k$$

Donde w_k y v_k corresponden al ruido en la planta y al ruido por las perturbaciones, con covarianzas Q_k y R_k respectivamente. Dichas etapas de predicción y estimación para un sistema lineal y discreto en el tiempo se pueden desarrollar como:

1. Estimació a priori:

$$\hat{x}_{k|k-1} = \Phi_k x_{k-1|k-1}$$

2. Covarianza asociada a la estimación a priori:

$$P_{k|k-1} = \Phi_k P_{k-1|k-1} \Phi_k^T + Q_k$$

3. Actualización de la medida:

$$\bar{y}_k = z_k - H_k \hat{x}_{k|k-1}$$

4. Ganancia de Kalman:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1}$$

5. Estimación a posteriori:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \bar{y}_k$$

Donde Φ es la matriz de transición de estados y z_k es el vector de medidas. Tal relación puede interpretarse como una suma ponderada entre el valor estimado a priori y el valor medido en base a la ganancia de Kalman (la que varia en función de las covarianzas de la estimación).

1.3. IEEE 802.11 WiFi

El protocolo más importante en la implementación de una red de área local inalámbrica (WLAN) se rige por los lineamientos del estándar IEEE 802.11. Mejor conocido como *WiFi*, es una marca registrada de *Wi-Fi Alliance*, empresa registrada en Austin (Texas) en 1999. Esta restringe el uso de dispositivos a sólo aquellos que hayan pasado satisfactoriamente un conjunto de requerimientos.

Desde su creación, el estándar ha evolucionado para cumplir con una alta demanda de aplicaciones. De esta manera, ha sido imperativo cumplir con las necesidades de ofrecer tasas de bits más elevadas, mayor eficiencia energética, mejores protocolos de acceso al medio y mejor uso espectral. La figura 2 describe los estándares que existen actualmente en el mercado bajo el nombre de *WiFi*.



Figura 2: *WiFi* trademark

- IEEE 802.11b: sale al mercado en el año 2000 e implementa esquemas de multiplexación DSSS (espectro expandido por secuencia directa) y modulación QPSK. Se alcanza una tasa máxima de 11Mbps a través de un novedoso protocolo de acceso al medio. Trabaja en la banda de 2.4GHz.
- IEEE 802.11a: es anunciado en el 2001, aumentando la máxima tasa a nivel de capa física a 54Mbps. Hace uso de OFDM como esquema de multiplexación, estándar que se ha mantenido a lo largo del tiempo. Trabaja en la banda ISM de 5GHz e implementa esquemas de modulación BPSK, QPSK, 16-QAM y 64-QAM.
- IEEE 802.11g: sale al mercado en el año 2003 y es el resultado de pasar el esquema de 802.11a hacia la banda de 2.4GHz. La tasa de bits máxima no cambia.
- IEEE 802.11n: en el año 2009, Wi-Fi alliance anuncia la versión *n*. Utilizando OFDM con diversidad espacial (sistemas MIMO), junto a una nueva tasa mandatoria de codificación de canal (5/6) y canales de 20MHz o 40MHz (concatenación), se alcanzan tasas teóricas de 600Mbps. Trabaja en las bandas ISM de 2.4GHz y 5GHz. Es compatible con las versiones b/g/n
- IEEE 802.11ac: expande las características implementadas en la versión *n*, hace uso de concatenación de canales para llegar a anchos de banda

de hasta 160 MHz, aumenta el esquema de modulación hasta 256-QAM y utiliza esquemas MIMO de hasta 8 antenas para alcanzar tasas por encima de 1 Gbps. Trabaja en la banda de 5GHz.

- IEEE 802.11ad: supone un cambio de paradigma dentro del protocolo. Es llevado a cabo por parte de la *Wireless Gigabit Alliance* a través de un esfuerzo para llevar *multi-gigabit wireless internet* a los usuarios de WiFi. Hace uso de la banda ISM de 60GHz, sin dejar de lado las frecuencias de 2.4Ghz y 5Ghz utilizadas en protocolos anteriores. Desea ofrecer hasta 7Gbps con un rango de cobertura de hasta 10m.

2. Implementación

El sistema a implementar debe funcionar bajo el esquema base mostrado en la figura 3:

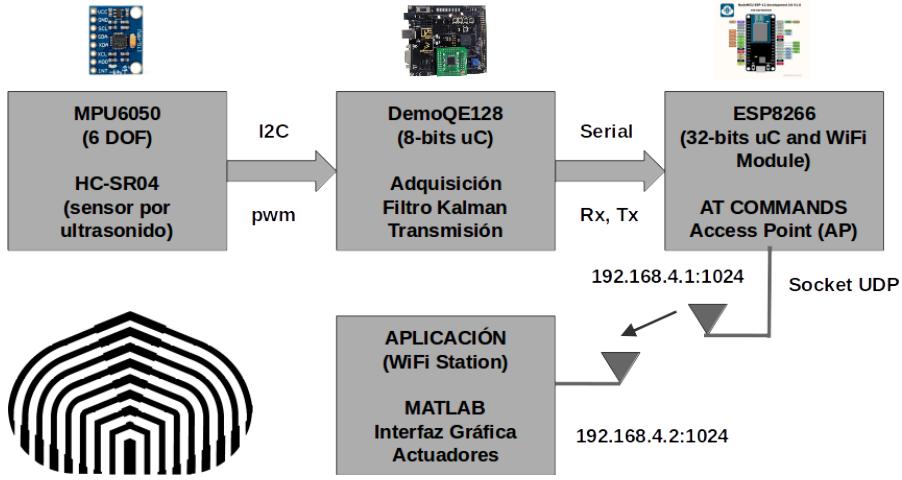


Figura 3: Estructura del sistema

El que se basa en las siguientes etapas:

- **Adquisición:** registra los valores de aceleración lineal y velocidad angular en cada eje, además de medir la distancia del sistema respecto al suelo a través de un sensor por ultrasonido.
- **Procesamiento:** estima el desplazamiento del dispositivo y su orientación (ángulos de roll y pitch) en base a las medidas realizadas y a través de un filtro de Kalman.
- **Transmisión:** se hace uso del estándar IEEE 802.11 para enviar los datos de posición y orientación a través de la red de forma inalámbrica. Cada dispositivo tiene asociada una dirección IP única y un puerto específico.
- **Simulación:** Para verificar el funcionamiento del sistema de teleoperación se implementa un software que simule el funcionamiento sobre un robot manipulador SCARA. El sistema puede ser utilizado sobre cualquier otro tipo de aplicación.

2.1. Adquisición

El sistema a implementar debe ser guiado en base a mediciones inerciales de la orientación del dispositivo generada por el usuario, por ello se opta por la estimación de las siguientes variables físicas:

- Orientación del dispositivo: medido por un acelerómetro y luego traducido a ángulos de Roll y Pitch.
- Tasas de rotación del dispositivo: medidas por un giróscopo y luego traducida a tasas de Euler (equivalente a tasas Roll y Pitch).
- Altura del dispositivo: medida por un sensor de ultrasonido y luego traducida a unidades métricas en un cierto rango de interés.

En base a dichos requerimientos el equipo selecciona los siguientes sensores digitales para realizar las medidas, basados en disponibilidad y complejidad de implementación:

- Acelerometro-Giroscopo Digital de 6 grados de libertad *MPU6050* [2]: con sensibilidades de $\pm 2g$, $\pm 4g$, $\pm 8g$ y $\pm 16g$ para el acelerómetro y de ± 250 , ± 500 , ± 1000 y ± 2000 deg/seg para el giroscopo. Rango alimentación desde 2.375V hasta 3.46V. Se utiliza I^2C como protocolo de comunicación maestro-esclavo. La dirección del dispositivo (siempre actuando como esclavo) es 110100x, donde el último bit se define arbitrariamente a través del pin AD0 (LSb Slave Address) del sensor.
- Medidor de distancia por ultrasonido *HC-SR04* [4]: trabaja a través del esquema de la figura 4 y utiliza un voltaje de alimentación nominal de 5V, con lo cual es necesario reducir la salida del sensor para acoplarlo con microcontroladores de 3V. La señal de trigger que inicia los 8 pulsos continuos de 40KHz debe durar al menos 10 μs . Se utiliza la siguiente ecuación para trasladar los valores a unidades métricas:

$$d(m) = \frac{pulseWidth(s) * 340(m/s)}{2}$$

$$d(cm) = \frac{pulseWidth(\mu s)}{58}$$

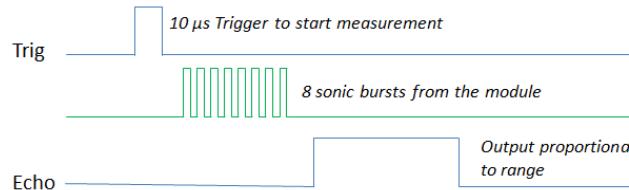


Figura 4: diagrama de tiempos del sensor por ultrasonido

2.2. Procesamiento

Debido a la disponibilidad en el laboratorio, se utiliza un HCS08QE128 [1] de *Freescale* como micro-controlador para este proyecto. El procesador viene integrado con un módulo de desarrollo *demoQE128*. Es una arquitectura de 8-bits con una frecuencia de reloj de hasta 25MHz. Posee los puertos ADC, I^2C y PWM necesarios para la implementación deseada.

Para el procesamiento de los datos se requiere que el μC sea capaz de realizar las siguientes operaciones:

1. Adquirir data de los sensores por interfaz I^2C y Capture.
2. Calcular angulos Roll y Pitch.
3. Calcular Tasas de Euler.
4. Calcular distancia en metros.
5. Filtrar ruido en las orientaciones mediante Filtro de Kalman.
6. Enviar data final a la etapa de transmisión.

Donde el procedimiento a seguir es detallado a continuación.

Desde el sensor MUPU6050 se adquieren mediante I^2C las orientaciones respecto a los ejes XYZ a_x , a_y y a_z , de allí calculamos los angulos roll (ϕ) y pitch (γ):

$$\phi = \text{atan}\left(\frac{a_y}{a_z}\right)$$

$$\gamma = \text{asin}(-a_x)$$

luego las tasas de euler se calculan como:

$$\omega_\phi = \frac{a_z w_y - a_y w_z}{\cos(\gamma)}$$

$$\omega_\gamma = w_x - a_x \frac{a_y w_y + a_z w_z}{1 - a_x^2}$$

Donde en todos los casos las funciones $\cos()$, $\sin()$ y $\tan()$ son implementadas mediante aproximaciones polinómicas de tercer orden, esto restringe la región de operación de los sensores a entradas entre $-\pi/2$ y $\pi/2$.

Luego de adquiridas las orientaciones se calcula la distancia entre el sensor ultrasonido y el suelo, para ello se usa el componente `capture()` de Processor Expert y se calcula la distancia mediante la relación:

$$d(m) = \frac{\text{pulseWidth}(s) * 340(m/s)}{2}$$

A partir de las medidas entonces se realizó el filtrado de cada ángulo por separado mediante filtro de Kalman, donde la matriz de transición de estados se escoge como:

$$\Phi = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}$$

Mientras que para el cálculo de las matrices Q y R se estiman usando MATLAB con el siguiente procedimiento:

1. Se leen el ángulo θ y su velocidad angular ω en una posición nominal.
2. Se deja de leer a los 30 segundos.
3. Se generan dos vectores $angle()$ y $rate()$ con el ángulo y su tasa durante la corrida.
4. se calcula la matriz de covarianza de ruido de planta Q con la función $cov()$ de MATLAB.
5. Se leen el ángulo θ y su velocidad angular ω en una posición nominal con perturbaciones fuertes en su entorno.
6. Se deja de leer a los 10 segundos.
7. Se generan dos vectores $angle()$ y $rate()$ con el ángulo y su tasa durante la corrida.
8. se calcula la matriz de covarianza de ruido de perturbaciones R con la función $cov()$ de MATLAB.

Luego ejecuta el algoritmo de filtro de Kalman como se presenta en la sección anterior. La implementación final del sistema funciona como se muestra en la figura 5.

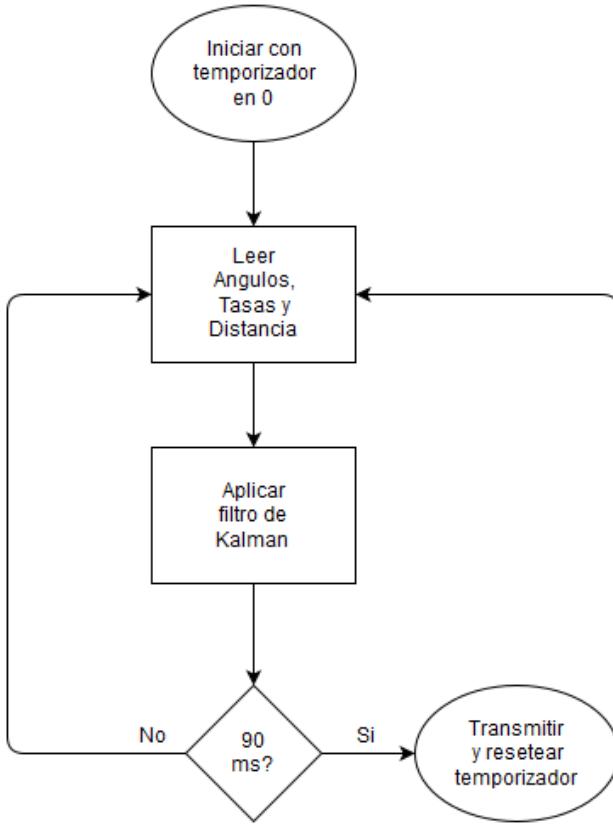


Figura 5: diagrama de estados del sistema

2.3. Transmisión de Datos

Luego de la estimación de los estados de los ángulos *Roll* y *Pitch* a través del Filtro de Kalman y la adquisición de la distancia a través del sensor por ultrasonido, se necesita transmitir los datos hacia la estación esclavo. Para ello se utiliza el μ C ESP8266 de *Espressif Systems* [5]. Consiste en una arquitectura de 32-bits con una frecuencia de reloj de hasta 80MHz. Contiene la implementación del protocolo de radio de 802.11b/g/n abarcando en torno al 20 % de la capacidad de procesamiento del chip, transmite con una potencia cercana a los 23dBm. De esta manera se asegura la compatibilidad con la pila de protocolos TCP/IP (internet).

Según la especificaciones del dispositivo, el ESP8266 puede funcionar en cualquiera de los modos descritos por un BSS (Basic Service Set) de WiFi:

1. Access-Point (AP)
2. Station (ST)

3. Modo dual AP+ST.

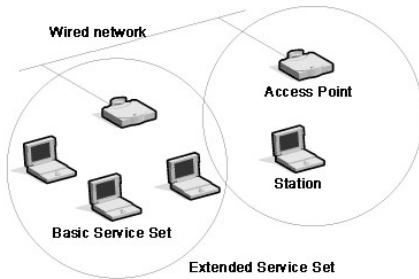


Figura 6: Arquitectura de WiFi como BSS

Para esta aplicación de teleoperación en particular, se utilizará el módulo WiFi como AP y el dispositivo esclavo como ST. Debido a la necesidad de soportar aplicaciones de baja latencia, se utiliza UDP como protocolo de transporte al momento de establecer el socket entre ambos puntos.

La comunicación entre el *demoQE128* y el *ESP8266* se hace a través del protocolo serial, utilizando los pines Rx (7) y Tx (5) del microcontrolador y sus respectivos pares en el módulo WiFi. Las figuras 7 y 8 ilustran esta conexión.

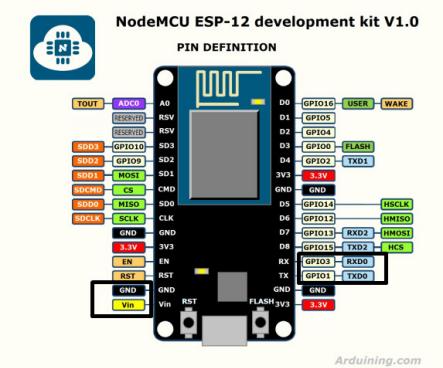


Figura 7: pines expuestos del ESP8266-12

VDD	1	2	PTA5/IRO/TPM1CLK /RESET
NRD	3	4	PTA5/IRO/TPM1CLK/RESET
PTB1/KB1/P5/TxD1/ADP5	5	6	PTA4/ACMP1/0/BG/DM
PTB0/KB1/P4/RxD1/ADP4	7	8	PTE7/TPM3CLK (n/c for 32 LQFP)
PTA2/KB1/P2/ADT1/ADP2	9	10	VREFL
PTA3/KB1/P3/SCL1/ADP3	11	12	VREFH
PTC0/TPM3CH0	13	14	PTA0/KB1/P0/TPM1CH0/ADP0/ACMP1+
PTC1/TPM3CH1	15	16	PTA1/KB1/P1/TPM2CH0/ADP1/ACMP1-
PTB3/BP1/P1/MOS1/ADP7	17	18	PTG0/ADP10 (n/c for 32 LQFP)
PTB4/TPM2CH1/MOS1	19	20	PTF1/ADP11 (n/c for 32 LQFP)
PTB2/KB1/P6/SPSCK1/ADP6	21	22	PTA8/TPM1CH2/ADP8
PTB6/TPM1CH1/SIS1	23	24	PTA7/TPM2CH2/ADP9
PTD1/KB2P1/MOS1Z	25	26	PTH8/SC1L2 (n/c for 32 LQFP)
PTD2/KB2P2/MOS2	27	28	PTI7/GDA2 (n/c for 32 LQFP)
PTD0/KB2P0/SPSCK2	29	30	PTD4/KB1P2/4 (n/c for 32 LQFP)
PTD3/KB2P3/SIS2	31	32	PTD5/KB1P2/5 (n/c for 32 LQFP)
PTC2/TPM3CH2	33	34	PTD8/KB1P2/6 (n/c for 32 LQFP)
PTC3/TPM3CH3	35	36	PTD7/KB1P2/7 (n/c for 32 LQFP)
PTC4/TPM3CH4/RST0	37	38	PTC7/TxD2/ACMP2-
PTC5/TPM3CH5/ACMPO	39	40	PTC6/RxD2/ACMP2+
(n/c for 32 LQFP) PTF2/A/DP12	41	42	PTB7/SC1/XTAL
(n/c for 32 LQFP) PTF3/A/DP13	43	44	PTB8/GDA1/XTAL
(n/c for 32 LQFP) PTF4/A/DP14	45	46	PTG3 (n/c for 32 LQFP)
(n/c for 32 LQFP) PTF5/A/DP15	47	48	PTG1 (n/c for 32 LQFP)
(n/c for 32 LQFP) PTF6/A/DP16	49	50	PTH0 (n/c for 32 LQFP)
(n/c for 32 LQFP) PTF7/A/DP17	51	52	PTH1 (n/c for 32 LQFP)
(n/c for 32 LQFP) PTG2/A/DP18	53	54	PTE6 (n/c for 32 LQFP)
(n/c for 32 LQFP) PTG3/A/DP19	55	56	NC

Figura 8: pines expuestos del demoQE128

2.4. Interfaz Gráfica

Para simular el sistema de teleoperación se opta por controlar un robot "SCARA" de tres enlaces (comunmente usado para operaciones de soldadura) en un ambiente de comunicación deteriorada como es mostrado en la figura 9.

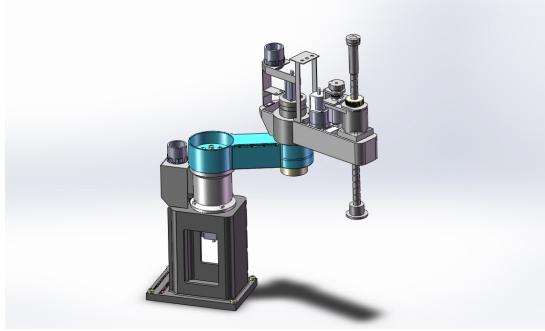


Figura 9: Manipulador SCARA a simular

En este caso la meta es controlar las articulaciones rotacionales del robot usando las mediciones inerciales y la velocidad de la articulación prismática es controlada mediante la velocidad del ultrasonido:

$$q_1 = \phi$$

$$q_2 = \gamma$$

$$\dot{q}_3 = d(m)$$

Pära el desarrollo de la interfaz gráfica se hace uso de MATLAB R2015 y el paquete de simulación de robótica Drake [6], desarrollado por el grupo de locomocón de robots (Robot Locomotion Group) del MIT en alianza con el Toyota Research Institute (TRI).

3. Resultados

En la siguiente figura se observa la circuitería final del proyecto.

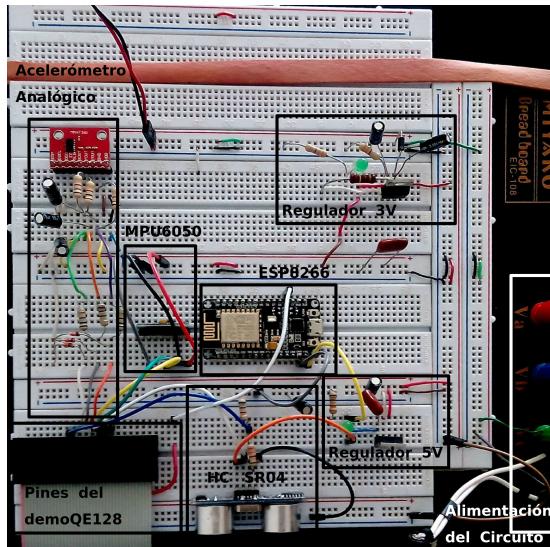


Figura 10: circuitería final

3.1. Adquisición y Procesamiento

A través de los sensores descritos anteriormente, el procesamiento a través del filtro de Kalman, la transmisión serial hacia el módulo ESP8266 y los tiempos asociados a la transmisión WiFi, se prueba el sistema a través de un socket UDP. Se leen los datos desde MatLab en la computadora que está actuando como estación esclava y se grafican respectivamente.

A partir de una posición inicial ($r=0$ y $p=0$) se verifica el funcionamiento del filtro Kalman para estimar los estados del sistema.

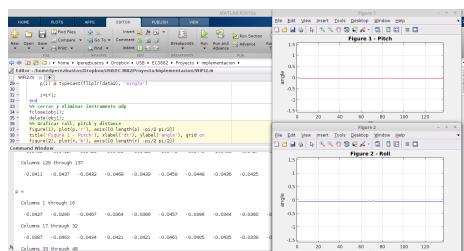


Figura 11: Posición inicial de Roll y Pitch

Se prosigue realizando movimientos entre $-\pi/2$ y $\pi/2$ para cada ángulo por separado. Al generar variaciones en el ángulo roll se obtienen los resultados de las figuras 14 y 15.

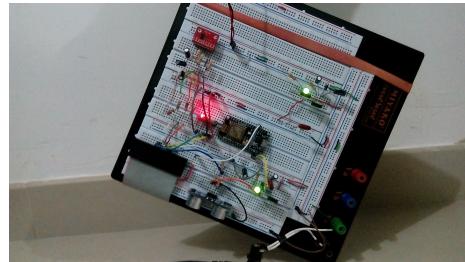


Figura 12: Variaciones en Roll (1)

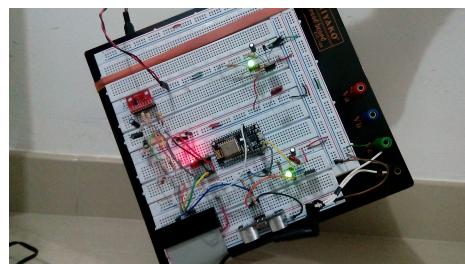


Figura 13: Variaciones en Roll (2)

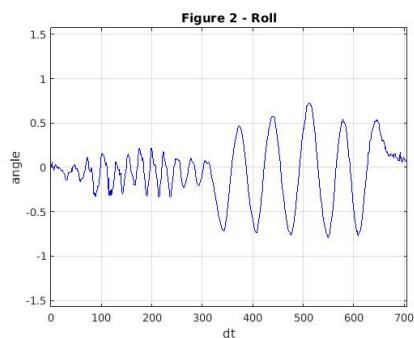


Figura 14: Variaciones en Roll

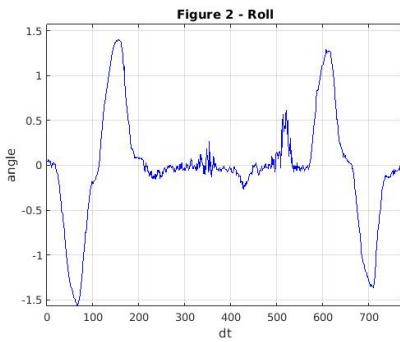


Figura 15: Variaciones completas en Roll

Se verifica, además, la adquisición de las variaciones ángulares en torno a pitch.

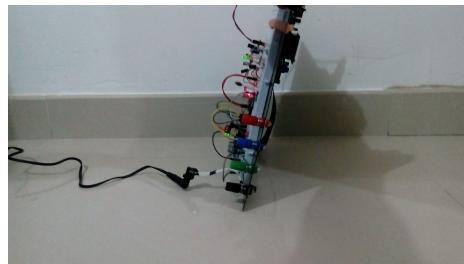


Figura 16: Variaciones en Pitch (1)

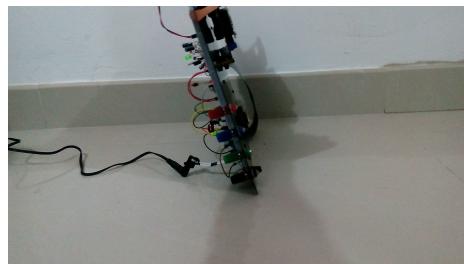


Figura 17: Variaciones en Pitch (2)

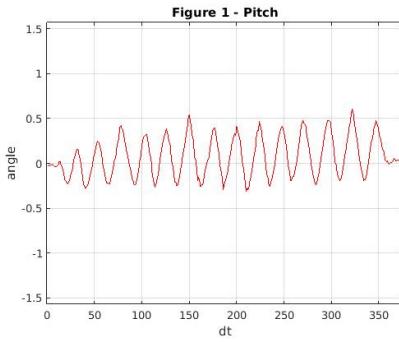


Figura 18: Variaciones en Pitch - Ángulo Pitch

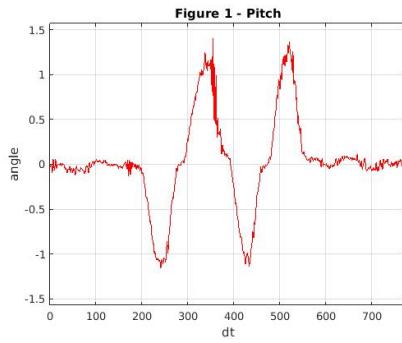


Figura 19: Variaciones completas en Pitch

Como figura de mérito, se observa como reacciona el sistema ante posibles perturbaciones.

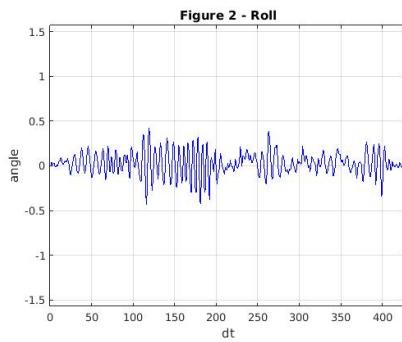


Figura 20: Perturbaciones en el Ángulo Roll

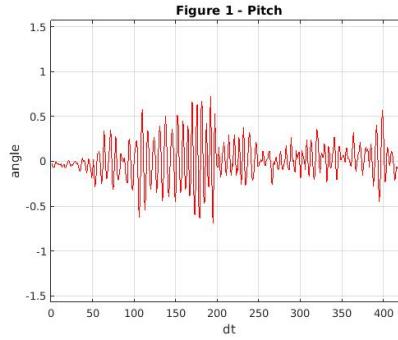


Figura 21: Perturbaciones en el Ángulo Pitch

3.2. Transmisión

Para verificar la conectividad en el sistema se utiliza un analizador de protocolos de red que guarde un registro de todos los paquetes que pasan por la interfaz de WiFi en el computador que funciona como ST. Para estos propósitos se utiliza *Wireshark*. Se coloca el sistema maestro en la posición inicial ($\text{pitch}=0$, $\text{roll}=0$) y comienza a transmitirse los paquetes de un nodo a otro. Las siguientes figuras ilustran el proceso de captura, visualización y verificación de los resultados.



Figura 22: Verificando el énlace con la red

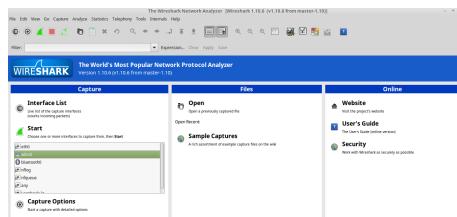


Figura 23: Seleccionando interfaz de wlan

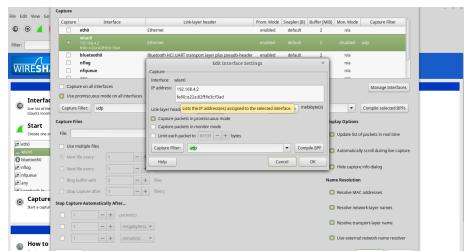


Figura 24: Seleccionando protocolo y modo de análisis de red

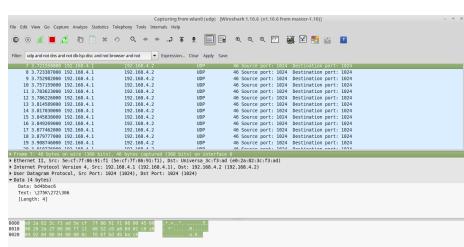


Figura 25: Paquetes UDP llegando a la interfaz de red

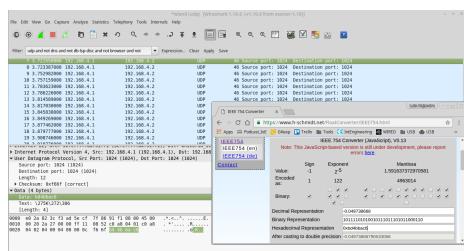


Figura 26: Verificación de los valores transmitidos

3.3. Teleoperación

Una vez implementada la interfaz gráfica con MATLAB se realizan pruebas en el visualizador de LIBBOT integrado con Drake y medidas realizadas cada 30 Hz. Un ejemplo de la teleoperación de un robot scara es mostrado en la figura 27

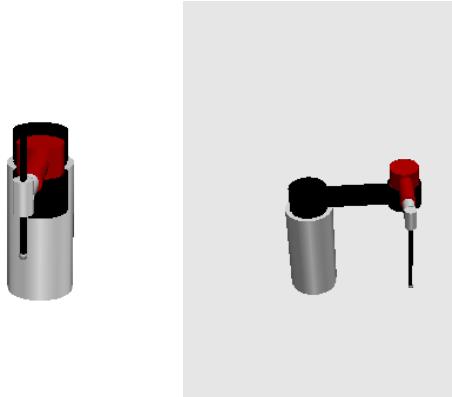


Figura 27: Teleoperación de un robot scara usando Drake y el sistema de teleoperaciones integrado

Los resultados efectivamente siguen las señales transmitidas por el usuario, aunque con un desfase introducido por el procesamiento del computador usado.

4. Consideraciones Finales

En este trabajo se ha desarrollado una unidad de teleoperación basada en mediciones iniciales con filtrado de Kalman y comunicación Wi-Fi. El sistema fue exitosamente integrado con MATLAB a una velocidad de 30 Hz. Se desarrolló una interfaz gráfica basada en Drake y se integró exitosamente con el proyecto.

Los resultados obtenidos muestran un funcionamiento correcto del sistema y del procesamiento realizado. La implementación de filtro de Kalman hacen al sistema robusto ante perturbaciones externas y el ruido de la planta. Sin embargo, esto trae sus desventajas al reducir la tasa de transmisión del sistema a 30 Hz (de alrededor de 300 Hz inicial).

La eficiencia del sistema sin embargo resulta mas que suficiente para el simulador implementado (que lee datos cada 90 Hz), por ello es importante resaltar que se limitó la tasa de transmisión a 90 Hz para evitar que se saturase el buffer del receptor debido a las diferencias de la tasa de transmisión y recepción del sistema.

En base a estos resultados a las experiencias obtenidas en el desarrollo del proyecto se realizan una serie de consideraciones para trabajos futuros en sistemas de este tipo:

- La implementación de la interfaz I^2C resulta bastante eficiente a la hora de configurarse mediante interrupciones, por ello se recomienda hacer uso de esta considerando los tiempos necesarios para la lectura y la escritura de los registros del sensor.
- El filtrado de Kalman resulta muy útil para sistemas lineales con ruido blanco aditivo, sin embargo es importante realizar las medidas correctas de las matrices de covarianza del sistema, pues en caso contrario el ruido no será filtrado de forma correcta pues la etapa de actualización fallará en realizar la fusión sensorial.
- La transmisión por Wi-Fi resulta simple, sin embargo es importante tomar en consideración los tiempos de transmisión y recepción de los paquetes de datos, pues de lo contrario el buffer del receptor puede saturarse generando errores en el sistema.
- El filtrado de la data no debe limitarse solo a filtro de Kalman, es posible realizar un preprocesamiento de la data haciendo uso de un filtro FIR el cual es fácil de implementar y no requiere muchos recursos del sistema.
- Las aproximaciones polinómicas de las funciones trigonométricas resultan de gran utilidad para aplicaciones embebidas, sin embargo estas limitan el dominio de las ecuaciones de forma significativa, por lo tanto en un futuro

puede resultar de mayor utilidad un sistema de aproximaciones mas exacto y menor restringido.

La implementación realizada cumple con los objetivos planteados al inicio del curso e integra elementos adicionales a los establecidos. Finalmente, el grupo concluye que los objetivos fueron cumplidos y los resultados obtenidos corresponden con los deseados, por lo tanto la implementación final es satisfactoria.

Los código fuente del proyecto se podrán encontrar próximamente en github a través de las cuentas @lperezbustos y @baceituno

Referencias

- [1] FreeScale. Microcontrolador de 8-bits de freescale. <http://www.nxp.com/assets/documents/data/en/datasheets/MC9S08QG8.pdf>.
- [2] InvenSense. Acelerómetro-gyroskop de 6 dof - mpu6050. https://www.cdiweb.com/datasheets/invensense/MPU-6050_DataSheetV3%204.pdf.
- [3] Pat Marion, Robin Deits, Andrés Valenzuela, Claudia Pérez D'Arpino, Greg Izatt, Lucas Manuelli, Matt Antone, Hongkai Dai, Twan Koolen, John Carter, et al. Director: A user interface designed for robot operation with shared autonomy.
- [4] micropik. Sensor de distancia hc-sr04. <http://www.micropik.com/PDF/HCSR04.pdf>.
- [5] Espressif Systems. Microcontrolador de 32-bits de espressif systems. http://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf.
- [6] Russ Tedrake and the Drake Development Team. Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems. <http://drake.mit.edu>, 2016.