

# Sensor de RF de Bajo Coste para el Procesado Remoto de Señales GNSS

Joaquim Gáñez-Fernández, Alejandro Pérez-Conesa, Gonzalo Seco-Granados, José A. López-Salcedo

Grupo de Signal Processing for Communications and Navigation (SPCOMNAV)

Departamento de Telecomunicación e Ingeniería de Sistemas, IEEC-CERES

Universitat Autònoma de Barcelona (UAB), 08193 Bellaterra, Barcelona, Spain

Email: joaquim.ganez@e-campus.uab.cat, {alejandro.perez, gonzalo.seco, jose.salcedo}@uab.cat

**Abstract**—Este trabajo presenta el desarrollo de un prototipo de sensor de radio-frecuencia (RF) de bajo coste. Su objetivo es capturar fragmentos de señal de los satélites de GNSS (*Global Navigation Satellite Systems*), como es el caso de GPS o Galileo. Estos fragmentos son posteriormente enviados a un servidor remoto en donde se calcula la posición del sensor. De esta forma se libera al sensor de toda la carga computacional y se convierte en un mero acondicionador de señal, el cual puede ser implementado de forma sencilla y económica utilizando componentes de Software-Defined Radio (SDR) existentes en el mercado. La motivación de procesar señales de forma remota sigue el paradigma emergente de procesado de señal en la nube, que se espera que en los próximos años impere en multitud de los sistemas existentes.

## I. INTRODUCCIÓN

En los últimos años se ha experimentado una vertiginosa evolución en los dispositivos de radio-frecuencia (RF) capaces de acondicionar y digitalizar señales de onda centimétrica. Este es el caso de señales que ocupan la banda UHF que se extiende desde los 300 MHz hasta los 3 GHz, donde por ejemplo se encuentran los servicios de telefonía móvil, televisión digital y sistemas de posicionamiento como GPS o Galileo. A éstos últimos se los conoce también por sus siglas en inglés como sistemas GNSS (*Global Navigation Satellite Systems*). De entre las diferentes aplicaciones que se ha mencionado, el caso de la televisión digital merece una mención especial ya que se trata de un servicio de radiodifusión en abierto accesible a millones de personas y que tiene una gran penetración en el mercado de consumo. Es en esta aplicación en concreto donde recientemente han surgido dispositivos de RF que, gracias a las economías de escala, han reducido drásticamente sus costes [1] y los han hecho accesibles al gran público [2]. Además, su filosofía de *Software-Defined Radio* (SDR) permite reconfigurarlos de forma flexible por software haciendo aún más versátil su uso en multitud de aplicaciones [3], incluyendo las de instrumentación y medida [4].

Esta gran flexibilidad permite implementar sistemas de forma sencilla utilizando dispositivos SDR como cabezales de RF de bajo coste. Aunque las prestaciones no son comparables con dispositivos diseñados y fabricados a medida, sí que son

suficientes para una gran cantidad de aplicaciones de gran consumo. Las más populares corresponden a la recepción de televisión digital o de señales de radio amateur. Sin embargo, las posibilidades de los dispositivos de RF basados en el concepto de SDR van mucho más allá. En concreto, este trabajo presenta el caso de utilizar dispositivos de RF de bajo coste para implementar un sensor de señales de sistemas de posicionamiento por satélite GNSS, como es el caso de GPS o Galileo. El sensor diseñado es capaz de captar fragmentos de señal de pocos milisegundos de duración de forma autónoma, para posteriormente enviarlos a un servidor remoto para su análisis. De esta forma el sensor actúa como mero acondicionador de las señales recibidas, y el procesado de las mismas para la obtención de la posición del usuario se lleva a cabo remotamente. Ello permite una drástica reducción del consumo del sensor, puesto que el grueso del cálculo computacional se lleva a cabo fuera del sensor, permitiendo una mayor duración de la batería. Esta particularidad abre la puerta a su uso en aplicaciones de redes de sensores en el ámbito de *Internet of Things* (IoT) y Ciudades Inteligentes.

En este contexto, el objetivo del presente trabajo es presentar el diseño, implementación y test de un prototipo de sensor de RF para la captación de señales de GNSS que de forma autónoma, las envíe a una plataforma de procesamiento en la nube. Esta plataforma, denominada `cloudGNSSrx` [5], permite procesar archivos binarios con muestras de señales de GPS/Galileo para calcular la posición del usuario y está siendo utilizada comercialmente por la empresa Loctio [6].

## II. ELEMENTOS DEL SENSOR DE RF DE BAJO COSTE

Antes de presentar propiamente el diseño del prototipo de sensor RF de bajo coste, es importante mencionar brevemente su interacción con la plataforma de procesamiento de señales en la nube. La arquitectura del sistema se presenta de forma resumida en la Fig. 1, donde puede observarse cómo el sensor se encarga de enviar los datos en crudo (muestras de señales RF) a la plataforma, donde un receptor software se encarga de procesarlas y guardar los resultados en una base de datos.

Por lo que respecta al sensor propiamente, la Fig. 2 muestra de forma sintetizada los elementos principales que lo componen y que consisten en:

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness projects TEC2017-89925-R and TEC2017-90808-REDT.

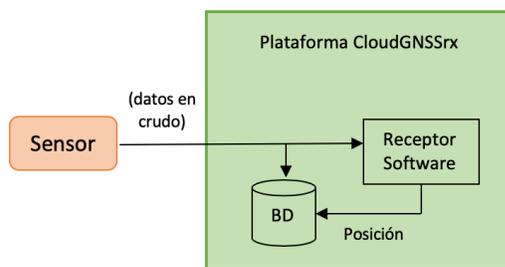


Fig. 1: Esquema de funcionamiento del sensor y la plataforma para el procesamiento remoto de señales GNSS.

- Antena: Elemento que actúa para captar las señales en la banda de interés, en particular la banda L1 donde se encuentran las señales de GPS L1 y Galileo E1. Al ser señales extremadamente débiles, se utiliza una antena activa para poder aplicar una ganancia a la señal recibida y facilitar el posterior análisis.
- Cabezal de RF por SDR: Es el encargado de acondicionar la señal captada por la antena, convertirla a banda base y digitalizarla para proporcionar a su salida un flujo de datos binarios. Todo ello configurable por software.
- Placa de control: Formada por una Raspberry Pi Zero W que permite implementar las operaciones básicas de control del cabezal de RF, captura de datos, almacenamiento y comunicación inalámbrica con la plataforma remota.
- Batería: Encargada de proporcionar alimentación a la placa de control y al cabezal de RF. Éste último se encarga a su vez de alimentar la antena a través de un bias-tee interno.

Durante el desarrollo del sensor se ha considerado dos modelos de SDR para actuar como cabezal de RF. Se trata del RTL-SDR v3 [7] y del Airspy Mini [8], cuyas características se presentan con más detalle en la siguiente sección.

### III. CABEZALES DE RF BASADOS EN SDR

Tal y como se ha comentado anteriormente, los cabezales de RF escogidos para el desarrollo del sensor son el RTL-SDR v3 y el Airspy Mini. Ambos presentan una fisonomía parecida contando con un puerto SMA de entrada para la conexión de antena, y un puerto USB de salida a través del cual se reciben las muestras digitalizadas de la señal de entrada. Ambos dispositivos están basados en el concepto de Software Defined Radio (SDR), y por tanto son configurables. A continuación se describe sus características más importantes.

#### A. RTL-SDRv3

El dongle RTL-SDR v3 (RTL2832) se muestra en la Fig. 3 y consiste en un cabezal de RF basado en SDR de bajo coste (~ 25€) que está formado por dos elementos principales:

- Cabezal de RF: basado en el chipset Rafael R820T2 con un rango de frecuencias de entrada desde 24 hasta 1766 MHz, y un ancho de banda máximo de 2.8 MHz (2.4 MHz de forma estable). El oscilador local es un TCXO

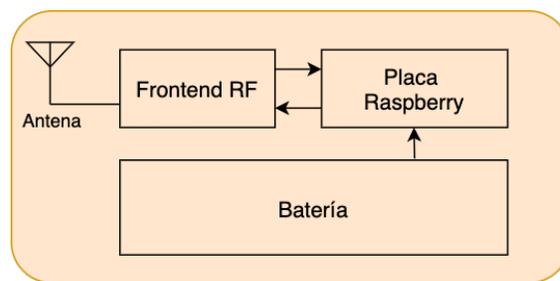


Fig. 2: Esquema de componentes del prototipo de sensor GNSS desarrollado.

con una estabilidad en frecuencia de 1 ppm y una deriva con la temperatura de entre 0.5 y 1 ppm.

- ADC: convertor analógico-digital (ADC) de 8 bits por muestra IQ, las cuales son proporcionadas por el cabezal R802T2 a una frecuencia de muestreo de 2.8 Msps (o 2.4 Msps de forma estable).

Como funcionalidades adicionales cabe mencionar que este SDR incluye un bias-tee seleccionable por software que proporciona un voltaje de 4.5 V en el puerto SMA. Esta funcionalidad es de suma utilidad ya que permite alimentar el LNA de la antena activa que conectaremos al puerto SMA, y así beneficiarnos de los 25-35 dB de ganancia que típicamente proporciona una antena de este tipo en la banda L1.



Fig. 3: Cabezal de RF basado en SDR, RTL-SDR.

#### B. Airspy mini

El cabezal de RF Airspy Mini se muestra en la Fig. 4 y está formado también por dos elementos principales:

- Cabezal de RF: basado en el chipset Rafael R820T2 con un rango de frecuencias que se extiende desde 24 hasta 1700 MHz. Su ancho de banda es superior al del RTL-SDR llegando hasta los 6 MHz, lo cual lo hace adecuado para captar señales de Galileo. El oscilador local es un TCXO con una estabilidad en frecuencia de 0.5 ppm.
- ADC: convertor analógico-digital (ADC) de 16 bits por muestra IQ que pueden obtenerse a una frecuencia de muestreo de 3 o 6 Msps. De estos 16 bits, realmente sólo son útiles los 12 más significativos.

Tal y como ocurre en el caso del RTL-SDR v3, el Airspy Mini también dispone de un bias-tee interno de 4.5V seleccionable por software. La principal ventaja del Airspy Mini

	RTL-SDRv3	Airspy Mini
Chipset RF	Rafael R820T2	Rafael R820T2
Rango de frecuencias	24-1766MHz	24-1700MHz
ADC	8 bits	12 bits
Ancho de banda	2.8 MHz (2.4 MHz estables)	6 MHz
Deriva oscilador local	$\pm 1$ ppm	$\pm 1$ ppm
Bias-Tee interno	Sí	Sí
Precio	20€	140€

Tab. I: Resumen de los parámetros de los SDR RTL-SDRv3 y Airspy Mini.

es el mayor ancho de banda que ofrece respecto al RTL-SDR, aunque su precio es mayor, alrededor de los  $\sim 140$ €. Se hace necesario valorar, por tanto, la necesidad o no de incorporarlo según la aplicación final a la que se destine el sensor.



Fig. 4: Cabezal de RF basado en SDR, Airspy Mini.

En la tabla I puede observarse una comparación de los parámetros más relevantes de los modelos de SDR RTL-SDRv3 y Airspy Mini.

#### IV. CALIBRACIÓN AUTOMÁTICA DEL OSCILADOR DEL CABEZAL DE RF

Tanto el RTL-SDR como el Airspy Mini disponen de un oscilador local basado en un Temperature Controlled Crystal Oscillator (TCXO) con una estabilidad en frecuencia considerable a pesar de ser dispositivos de bajo coste (sobretudo el RTL-SDR). Sin embargo, para las frecuencias de operación de las señales de GNSS, alrededor de 1.5 GHz, apenas unos pocos ppm de error ocasionan errores de frecuencia de varios kHz. Ello dificulta la adquisición de los satélites visibles al no coincidir la frecuencia esperada a la que se ha sintonizado el cabezal con la frecuencia real a la que se encuentra centrada la señal. Por ello se hace necesario un mecanismo de calibración para corregir este offset del oscilador local.

Para ello se ha hecho uso del efecto Doppler que presentan los satélites de GNSS. Por ejemplo, los satélites de GPS completan una órbita sobre la Tierra cada 12 horas aproximadamente, lo cual supone que viajan a una velocidad constante de cerca de 14.000 km/h. Esta velocidad genera un efecto Doppler que el usuario ve a través de su componente radial sobre la línea de visión directa entre el satélite y el usuario. Puesto que las trayectorias que siguen los satélites son todas diferentes vistas desde el receptor de usuario, también son distintos los Dopplers con los que el receptor recibe cada una de sus señales. Estos Dopplers son conocidos ya que

dependen de la órbita de los satélites y de la posición del usuario, aunque no es necesario que ésta última sea muy precisa para el propósito que se persigue.

Para calcular los Dopplers se ha utilizado el simulador orbital open-source SGP4 [9] que, a partir de un fichero de ephemerides orbitales TLE (*Two Line Element*) disponible por ejemplo en [10], y una posición aproximada del usuario, proporciona los Dopplers esperados en esa posición para una fecha y hora determinadas. Una vez se conocen los Dopplers esperados se puede configurar el receptor del servidor remoto para que lleve a cabo la búsqueda de los satélites para diferentes valores tentativos de Doppler. Acabada la búsqueda, el offset del reloj del sensor acaba siendo la diferencia entre el Doppler encontrado por el receptor remoto y el Doppler esperado. Esta calibración se puede realizar periódicamente en caso de que el sensor esté sujeto a variaciones de temperatura que hagan variar el offset del oscilador local. Mientras tanto, en el receptor remoto se puede desactivar la búsqueda de Doppler y utilizar directamente los valores que proporciona el simulador SGP4 más el offset calibrado.

De este modo, es posible calibrar periódicamente cualquier SDR, algo que es muy útil debido a que algunos modelos pueden variar considerablemente su error de reloj inducido dependiendo de la temperatura a la que se encuentren.

#### V. FUNCIONAMIENTO DEL SENSOR

Por lo que respecta a la implementación hardware del sensor, la arquitectura sigue el esquema mostrado anteriormente en la Fig. 2. El resultado final se muestra en la Fig. 5 donde se presenta una vista superior del sensor, cuyos componentes están contenidos en una carcasa Rubicon del fabricante OpenH [11]. Se trata de una carcasa color naranja con la parte superior transparente certificada IP67. En primer nivel justo debajo de la tapa transparente se puede observar la antena GNSS modelo 580 del fabricante Cirocomm. Se trata de una antena activa de tipo parche, de 5cm de lado, con una ganancia de 32 dB, 1.5 dB de figura de ruido, alimentación entre 3 y 5V y un consumo de 10 mA. La antena está conectada al puerto SMA de un RTL-SDR actuando como cabezal de RF, el cual aparece en la Fig. 5 desprovisto de su carcasa metálica por razones de espacio. La salida USB del RTL-SDR se transporta mediante un cable USB plano flexible hasta una placa Raspberri Pi Zero W situada en la parte inferior del sensor. Finalmente, el elemento cuadrado de color negro y grandes dimensiones que se observa debajo de la antena es un powerbank compacto de 10,000mAh que proporciona varios días de autonomía.

Por lo que respecta al funcionamiento lógico del sensor, éste funciona de forma autónoma respondiendo a las peticiones que llegan desde el servidor remoto. Para ello efectúa una consulta periódica para comprobar si existen trabajos pendientes para, en su caso, llevarlos a cabo, o de lo contrario, permanecer en reposo. A continuación se describe con más detalle este procedimiento.



Fig. 5: Imagen del prototipo de sensor de RF para la captación de señales GNSS que se ha desarrollado en este trabajo.

### A. Inicialización del sensor

Para comunicarse con la plataforma, el sensor debe estar registrado por un usuario previamente validado. Este proceso puede realizarse desde el mismo sensor una vez completado el registro. En ese caso la plataforma provee al sensor de la configuración y credenciales necesarias para poder trabajar.

### B. Rutina de escucha

Los sensores correctamente registrados e inicializados funcionan automáticamente una vez arrancan. Su funcionamiento está basado en una rutina de escucha según se indica en la Fig. 6 compuesta por 6 pasos:

- 1) Comprobar si existe una solicitud de cambio de configuración pendiente. En caso afirmativo, el sensor se reconfigura automáticamente. En caso negativo la configuración local del sensor no se modifica.
- 2) Comprobar si existe una solicitud de captura de señal pendiente. En caso afirmativo, realiza los pasos de 3 a 6. En caso negativo, salta directamente al paso 6.
- 3) Capturar señal usando la configuración local.
- 4) Aplicar compresión MSB (*Most Significant Bit*) al fichero obtenido en el paso anterior. Este paso es opcional, seleccionable por el usuario, y se describe con más detalle en la sub-sección a continuación.
- 5) Enviar el fichero de datos a la plataforma.
- 6) Esperar un periodo de tiempo predefinido en la configuración local y volver al paso 1.

### C. Compresión de muestras

A la salida de ambos SDR se obtienen muestras digitales complejas IQ. En el caso del RTL-SDR son en formato `uint8` y en el caso del Airspy Mini existen varias opciones, la escogida en este trabajo es `int16`. Esto significa que cada muestra compleja está codificada por 16 y 32 bits, respectivamente. Si bien para muchas aplicaciones este rango dinámico es adecuado, la propia naturaleza de espectro ensanchado de las señales GNSS hace que prácticamente sea suficiente con 1 bit por muestra IQ (2 bits por muestra compleja) [12]. Desde este

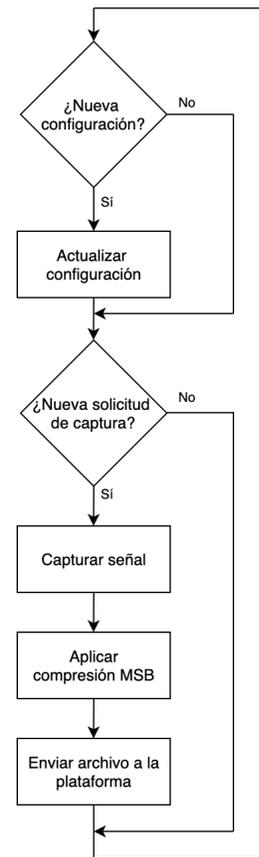


Fig. 6: Rutina de escucha del sensor desarrollado.

punto de vista, y en ausencia de otros fenómenos que pudieran sugerir el uso de más bits por muestra (e.g. en el caso de que hubiera interferencias presentes), utilizar más de 1 bit por muestra IQ no aporta grandes beneficios. Al contrario, para el sensor supone un esfuerzo adicional puesto que los ficheros binarios que debe transmitir vía radio al servidor remoto tienen un tamaño entre 8 y 16 veces superior al necesario.

Para mitigar este problema, el sensor desarrollado incorpora la funcionalidad de comprimir los ficheros de muestras antes de enviarlos a la plataforma. Se realiza una compresión MSB que reduce el tamaño de los ficheros en un factor de 8:1 en el caso del RTL-SDR y de 16:1 en el caso del Airspy Mini. Ello da lugar a un considerable ahorro de batería en el dispositivo.

## VI. RESULTADOS EXPERIMENTALES

### A. Descripción del estudio

El objetivo de esta sección es validar el concepto y analizar las prestaciones que se obtienen al procesar muestras de GNSS captadas con un sensor RF de bajo coste como el desarrollado. Para ello se ha hecho uso de dos unidades del prototipo mostrado en la Fig. 5 las cuales incorporan el cabezal RF RTL-SDR y el Airspy Mini, respectivamente. El estudio consiste en 50 capturas de señal en banda L1 sobre la frecuencia central de 1575.42MHz donde se encuentran las señales de GPS L1

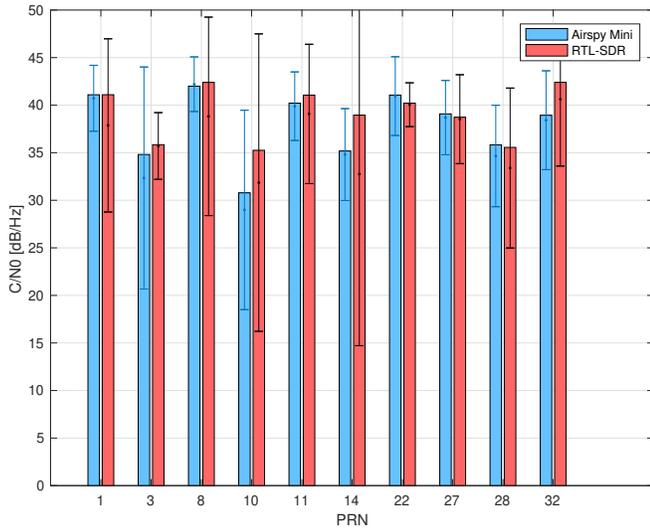


Fig. 7: Nivel de señal recibido para cada satélite de GPS adquirido por la plataforma utilizando el sensor desarrollado con el cabezal RTL-SDR o Airspy Mini.

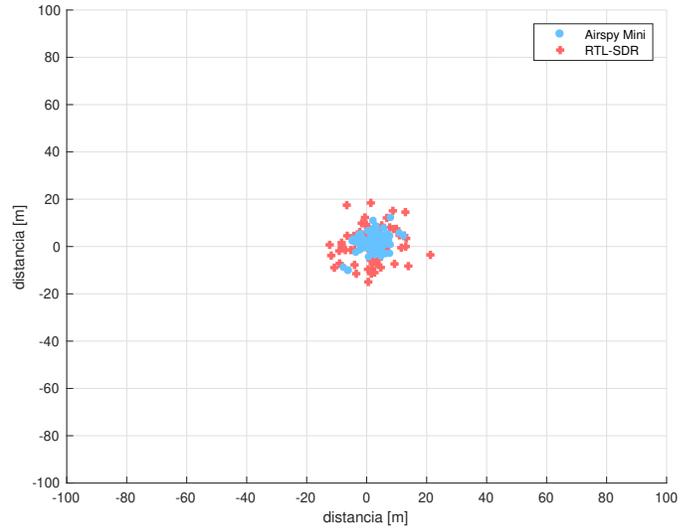


Fig. 8: Errores de posición para GPS L1 C/A obtenidos por la plataforma utilizando las capturas de 100 ms enviadas por el sensor que se ha desarrollado.

y Galileo E1. Las capturas o *snapshots* de señal tienen una longitud de 100 ms tanto para el caso de GPS como para el caso de Galileo. Éstas son enviadas por el sensor a la plataforma `cloudGNSSrx` donde son procesadas y se obtiene la posición del usuario para cada captura. Los dos sensores se sitúan a una distancia de un metro el uno al respecto del otro, de forma estacionaria. En ambos casos se utiliza compresión a 1 bit en las muestras que se envían a la plataforma.

A continuación se muestra los resultados obtenidos para el caso de GPS y de Galileo utilizando el sensor desarrollado como cabezal de RF. En ambos casos se muestran dos figuras. La primera de ellas presenta la calidad de la señal recibida, a través de la relación entre potencia de portadora y densidad espectral de ruido,  $C/N_0$ . La segunda figura muestra los errores en la estimación de la posición del sensor que se han obtenido en la plataforma al utilizar como cabezal de RF las dos variantes desarrolladas con el RTL-SDR y el Airspy Mini.

### B. Resultados al procesar señales de GPS L1 C/A

Los valores promedio de  $C/N_0$  obtenidos con las dos variantes del sensor se muestran en la Fig. 7 para los 10 satélites adquiridos. El eje horizontal muestra el identificador del satélite a través de su número de código pseudoaleatorio (PRN). Las barras azules corresponden a los datos obtenidos con el Airspy mini y las rojas al RTL-SDR. En cada barra se indica además el rango de valores obtenidos.

Como puede observarse, ambos sensores adquieren razonablemente bien la mayoría de satélites visibles. Las capturas con el RTL-SDR resultan en valores de  $C/N_0$  ligeramente superiores a los del Airspy Mini, con una diferencia de hasta 5dB a pesar de que las ganancias del Airspy Mini se configuraron prácticamente al máximo de su valor. Nótese que en condiciones ideales y señales captadas en exterior con buena

visibilidad del cielo, sería esperable un valor de  $C/N_0 \sim 45$  (dBHz). Los valores de  $C/N_0$  que se obtienen para los dos sensores oscilan entre los 35 y 45 dBHz, ligeramente por debajo de lo esperado debido esencialmente a las pérdidas de implementación (e.g. cuantificación de 1 bit, ancho de banda limitado, etc.), pero asumible para calcular la posición con capturas de 100 ms. Para GPS L1 C/A, el receptor en el cloud se configura de forma que cada captura de 100 ms se procesa utilizando  $N_1 = 5$  correlaciones coherentes de  $T_{\text{coh}} = 20$  ms cada una, las cuales se acumulan de forma no-coherente.

Los errores de posición que se obtienen con cada sensor se muestran en la Fig. 8 en forma de diagrama de puntos. Como puede observarse, la gran mayoría de puntos están comprendidos en una zona de dimensiones 40 por 40 metros centrada en el origen. Los errores obtenidos con el Airspy Mini están alrededor de 3 metros ( $1\text{-}\sigma$ ) mientras que para el RTL-SDR están alrededor de 4.5 metros ( $1\text{-}\sigma$ ). La ventaja del Airspy Mini, a similares valores de  $C/N_0$ , viene dada por su mayor ancho de banda respecto el RTL-SDR.

### C. Resultados al procesar señales de Galileo E1C

Los valores promedio de  $C/N_0$  al procesar Galileo E1C se muestran en la Fig. 9 para los 10 satélites adquiridos, 9 en el caso del RTL-SDR. Las capturas con Airspy Mini resultan en valores superiores a los del RTL-SDR. Ello es debido a que el ancho de banda del RTL-SDR es inferior al necesario para procesar adecuadamente la señal de Galileo E1C (mínimo  $\sim 4$  MHz), con lo cual el espectro a la salida del RTL-SDR queda muy recortado, implicando pérdida de potencia y distorsión de señal. Por ello se espera que los resultados del RTL-SDR con Galileo sean mucho peores que con el Airspy Mini. Este hecho se aprecia claramente en la Fig. 10.

En la misma figura se observa que los resultados para el

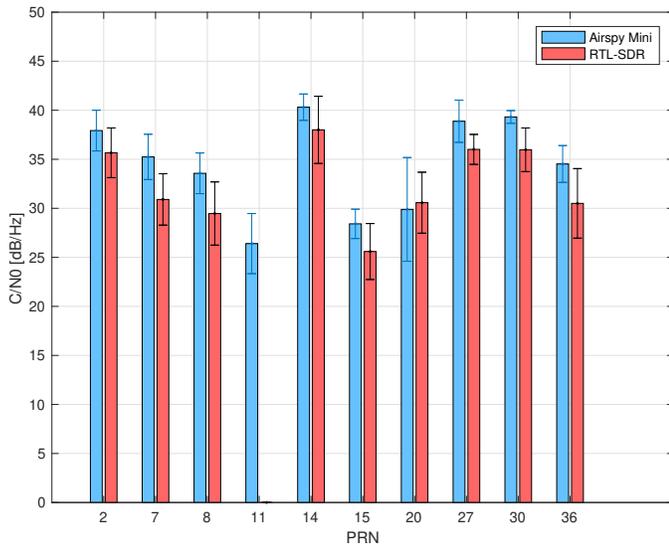


Fig. 9: Nivel de señal recibido para cada satélite de Galileo adquirido por la plataforma utilizando el sensor desarrollado con el cabezal RTL-SDR o Airspy Mini.

Airspy Mini están alrededor de 6 metros ( $1\sigma$ ), algo peores que los obtenidos con el mismo cabezal para el caso de GPS L1 C/A. Aunque las condiciones de visibilidad de los satélites GPS eran mejores en el momento de la captura (más satélites sobre el zenith y con mejor geometría), la principal degradación es debida a la antena, cuyo ancho de banda está más pensado para señales GPS que no para señales Galileo. Ello convierte a la antena en el cuello de botella de toda la cadena, haciendo que el espectro de la señal Galileo quede recortado y distorsionado de origen, con una degradación significativa según se observa al comparar las barras azules en las Fig. 7 y 9. Este efecto se ve amplificado en el caso del sensor con RTL-SDR, cuyos resultados para Galileo empeoran hasta los 35 metros ( $1\sigma$ ).

Para Galileo E1C, los 100 ms de la captura se procesan totalmente de forma coherente debido a la ausencia de bits de información en esta señal (i.e. se trata de una señal piloto), con lo cual  $T_{\text{coh}} = 100$  ms, y por tanto  $N_I = 1$ .

En resumen, debido a la naturaleza de las señales GPS L1, un cabezal de RF con  $\sim 2$  MHz de ancho de banda como el del RTL-SDR es suficiente para obtener posiciones de usuario con un error aceptable. Sin embargo, para el caso de Galileo E1 se hace necesario una antena y un cabezal con por lo menos el doble de ancho de banda que para el caso de GPS L1.

## VII. CONCLUSIONES

En este trabajo se ha presentado el desarrollo de un sensor de bajo coste para capturar fragmentos de señales GNSS y enviarlos a un servidor remoto para su procesamiento y cálculo de la posición del sensor. Por un lado se ha mostrado que es posible implementar un dispositivo de bajo coste que ofrezca resultados de posicionamiento con un error aceptable

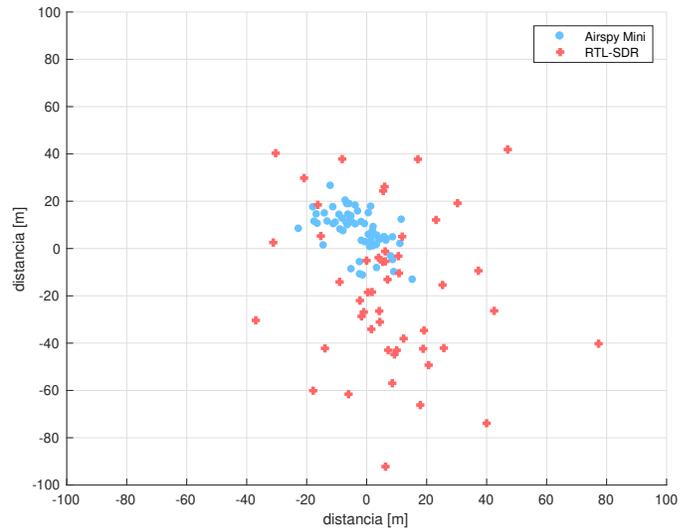


Fig. 10: Errores de posición para Galileo E1C obtenidos por la plataforma utilizando las capturas de 100 ms enviadas por el sensor que se ha desarrollado.

para muchas aplicaciones (e.g. control de flotas de vehículos, seguimiento de contenedores, localización de animales, etc.). Por otro lado se ha comprobado que para el caso de Galileo es conveniente utilizar una antena y un cabezal de RF de mayor ancho de banda, y por tanto de mayor coste, para poder aprovechar las prestaciones que ofrece Galileo. El prototipo aquí presentado puede servir de punto de inicio para el desarrollo de sensores más avanzados y para explotar la multitud de aplicaciones que presentan los sistemas GNSS en la actualidad.

## REFERENCIAS

- [1] S. Cass, "A \$40 software defined radio," *IEEE Spectrum*, vol. 3, no. July, pp. 22–23, 2013.
- [2] R. Stewart *et al.*, "A low-cost desktop software defined radio design environment using MATLAB, Simulink, and the RTL-SDR," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 64–71, 2015.
- [3] T. Ulversoy, "Software defined radio: challenges and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 4, pp. 531–550, 2010.
- [4] F. Harris and W. Lowdermilk, "Software defined radio: Part 22 in a series of tutorials on instrumentation and measurement," *IEEE Instrumentation & Measurement Magazine*, vol. 13, no. 1, pp. 23–32, 2010.
- [5] *CloudGNSSrx platform*, (accedido el 10/03/2020). [Online]. Available: <http://cloudGNSSrx.com>
- [6] *Loctio*, (accedido el 10/03/2020). [Online]. Available: <http://loctio.com/>
- [7] *About the RTL-SDR*, (accedido el 10/03/2020). [Online]. Available: <https://www.rtl-sdr.com/about-rtl-sdr/>
- [8] *Airspy Mini*, (accedido el 10/03/2020). [Online]. Available: <https://airspy.com/airspy-mini/>
- [9] *Simplified General Perturbation model orbital simulator (SGP4)*, (accedido el 10 de marzo de 2020). [Online]. Available: <https://pypi.org/project/sgp4/>
- [10] *Celestrak, NORAD Two-Line Element Sets*, (accedido el 10/03/2020). [Online]. Available: <https://www.celestrak.com/NORAD/elements/>
- [11] *RUBICON modular enclosure*, (accedido el 10/03/2020). [Online]. Available: <http://openh.io/rubicon/>
- [12] C. J. Hegarty, "Analytical model for GNSS receiver implementation losses," *Navigation*, vol. 58, no. 1, pp. 29–44, 2011.