

Technical Description of Locate-US: an Ultrasonic Local Positioning System based on Encoded Beacons.

J. Ureña, J. M. Villadangos, D. Gualda, M. C. Pérez, A. Hernández, J. J. García, A. Jiménez, J. C. García, J. F. Arango, E. Díaz

Electronics Department, University of Alcalá
E-28805, Alcalá de Henares (Madrid), Spain
{urena, david.gualda, villa, carmen}@depeca.uah.es

Abstract—LOCATE-US is a low-cost ultrasonic local positioning system (U-LPS) for fine-grained indoor positioning of portable devices, such as smartphones or tablets, in order to offer Location Based Services (LBS) such as augmented reality, guidance or multimedia services. It uses encoded signals to obtain a more accurate determination of difference of times of arrival (DTOA), as well as high immunity to noise and robustness against multipath effect and near-far effect.

Keywords—LPS, indoor positioning, smartphones, ultrasonic signals.

I. GENERAL VIEW OF LOCATE-US SYSTEM

A portable ultrasonic beacon architecture has been proposed, so several U-LPS can be easily deployed to cover wide indoor areas. Every U-LPS is formed of five emitters that share roughly the same coverage area, where the receiver to be located is moving around. The emitters in each U-LPS use both, a code division multiple access (CDMA) and a time division multiple access (TDMA) protocols. Every emitter is codified with a different code, with good auto-correlation properties and low mutual interference properties with the others. The emitters for the U-LPS are controlled by a LPC1768 microcontroller [1], that allows the configuration of the ultrasonic transmissions in terms of sampling frequency, modulation schemes and code patterns to be emitted. It also provides a wireless link to an external PC from where the configuration can be easily carried out. At the reception, a non-limited number of receivers can compute its own position by measuring differences of times of arrival (DTOA) of the incoming ultrasonic signals. Thus, there is no need of synchronization between the beacons and the receivers. To overcome audible artifacts and poor resolution (low bandwidth) at the 20-22kHz range available in current smartphones [2], the U-LPS operates around 41kHz and an external hardware device has been included to digitize the incoming signals and send them to the portable device for their processing. It includes a MEMs microphone and a microcontroller that manages the signal digitization and the data sending to the mobile device by means of a USB connection. Then, an Android application at the portable device allows the correlation of the received signals with the code pattern of the emitted codes, computes the DTOAs and the position by means of a hyperbolic trilateration algorithm and represents the

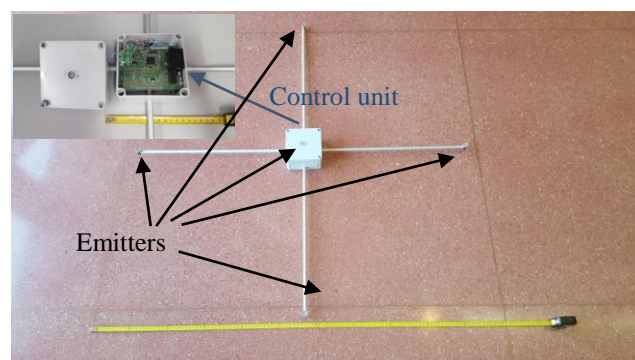


Fig 1.- General overview of a portable beacon unit for one U-LPS.

trajectory in the device screen in a time less than 0.5s, with a centimetric accuracy.

II. BASIC DESCRIPTION OF THE EMITTER ARCHITECTURE

Every U-LPS consists of five ultrasonic transmitters, distributed around a 70.7x70.7cm square structure (see Fig. 1). It covers an approximate area of 30m² when it is installed on the ceiling at a height of 3.5m. The ultrasonic transmitter is a Prowave 328ST160 [3]. The microcontroller for all the U-LPS provides a wireless link to an external PC, so as to easily configure the sampling frequency, encoding sequences and modulation scheme. These parameters can be modified in run-time, although in practical operation every U-LPS is configured only once in a set-up stage.

Specifically, in IPIN'2016 competition, the emitters have been configured to simultaneously transmit Kasami codes [4] with a length of 255 bits. Signals are encoded with the purpose of obtaining a good correlation peak that clearly identifies the time instant of arrival of the signal associated with every beacon, since no synchronization between emission and reception has been used. The encoded signals have been BPSK modulated with two periods of a sinusoidal carrier at 41.67kHz, which has been sampled at 500kHz. Hence, 24 samples (12 in each carrier period) are considered for each bit of the Kasami code. Since only one Digital to Analog Converter (DAC) is available in the microcontroller, a TDMA protocol has been used to transmit the

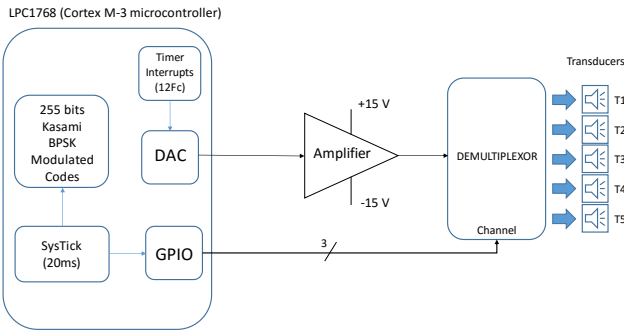


Fig 2.- Block diagram of the emitter control unit.



Fig 3.- Photo of the emitter control unit.

codes associated to every emitter at different times. A slot time of 20ms has been devoted for the transmission of every code. Hence, the time interval between two consecutive transmissions of the same code is of 100ms.

Considering the competition area in IPIN2016-Track4 of 12mx6m and the ceiling height of 3.56m, three U-LPS are enough to localize the mobile robot that will be moving around if we want good accuracy within all the robot trajectory. The selected Kasami family set provides 16 codes with low cross-correlation among them, which are sufficient to encode the five beacons of every U-LPS and to ensure low mutual interference and emitter identification even in cases where the robot receive signals from different U-LPSs. Anyway, LOCATE-US can be used in wider areas, as indicated in Fig. 4. To reduce costs, complete U-LPS can be installed only in critical zones, such as entrances or exits, and the inertial sensors of the mobile device together with individual ultrasonic sensors can be used in each zone where the positioning does not need to be so accurate. Bayesian filter algorithms can then be used to combine the information of the U-LPS signals with the inertial sensors of the mobile device.

III. BASIC DESCRIPTION OF THE RECEIVER ARCHITECTURE

LOCATE-US is thought to offer LBS to users through their mobile phones or tablets. Nevertheless, as in the case of IPIN-Track 4 competition, it can also be used to localize mobile robots. A non-limited number of portable receivers can compute their position autonomously by hyperbolic trilateration.

To avoid frequency limitations and restrictions due to the sampling rate of current smartphones (44.1kHz), a low cost ultrasonic signal acquisition module has been designed (see [5]

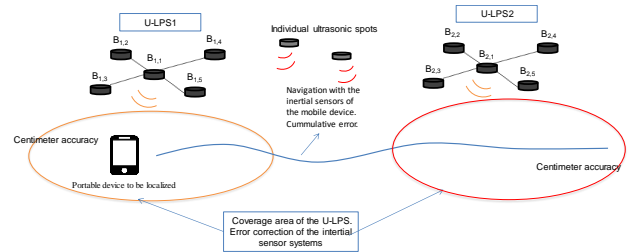


Fig 4.- Localization scheme proposal for an extensive area.

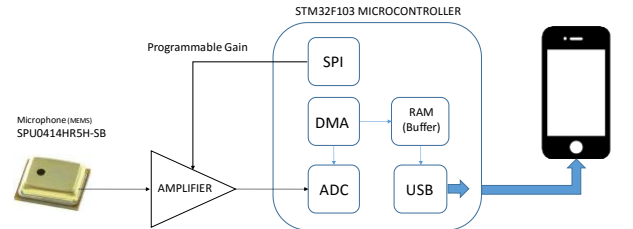


Fig 5.- Block diagram of the hardware adaptation module.



Fig 6.- Mobile device connected to the hardware

for further details). Fig. 5 shows the block diagram of the proposed adaptation hardware, while Fig. 6 shows a picture of a real device. Note its reduced dimensions and ease of connection to the portable device. The acquisition module includes a MEMS SPU0414HR5H-SB [6] microphone with an adequate bandwidth around the 41.67kHz. The incoming signals captured by the microphone are then high-pass filtered and amplified (or attenuated) through a programmable gain module controlled by a STM32F103. Thus, the gain can be adjusted depending on the application requirements and signal level. The microcontroller also digitizes the incoming signals at a sampling rate of 100kHz. A decimation by 5 is carried out to reduce the amount of data to be processed. Due to memory restrictions in the microcontroller, a buffer with 13.000 samples has been considered. Once the buffer is filled, it is send over a USB link to the portable device and a new acquisition starts. Hence 0.13s of the ultrasonic signal has to be saved to be processed. It assures that at least one complete period of every beacon code is received.

Once in the device, an Android application is run to obtain the location and depict the trajectory in the mobile screen. The application includes all the low level processing (signal demodulation and correlation, peak detection and TDOA computing algorithms), as well as the high level processing (position estimation through hyperbolic trilateration, map

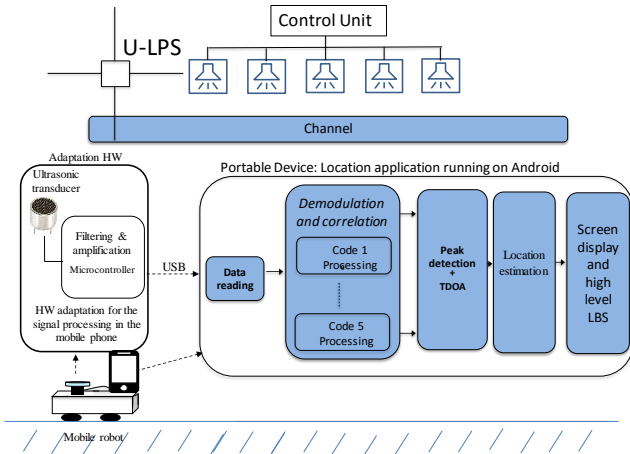


Fig 7.- Block diagram of the processing to be carried out in one portable device.

decoding and trajectory representation). Fig. 7 summarizes the proposed system and the task to be carried out in the Android Application. See [7] for more information regarding the developed Android application.

IV. POSITIONING ALGORITHM

LOCATE-US uses the well-known Gauss Newton hyperbolic trilateration algorithm [8] to obtain the mobile device position. The algorithm needs as inputs the difference of distances between a reference beacon and the others. These differences are obtained by means of correlation with the original emitted code patterns. Specifically, the correlation has been done by using the FFT. The complex conjugates of the FFT of the codes after the modulation process are loaded in a set-up stage of the Android application. On the other hand, the FFT of the data stored in the input buffer is performed. Then, both data sets are multiplied and the inverse transform of the product is carried out to obtain the correlation, see eq. (1). This is accomplished by means of two different libraries from Apache Commons, that provides well optimized and tested codes to manage Fourier transforms and complex numbers [9].

$$g_i[k] = IFFT(X \cdot H_i^*) \quad (1)$$

Where $g_i[k]$ represents the cross correlation between the input signal $x[k]$ and the template of the modulated codes $h_i[k]$, for every beacon i , $i=1, 2, \dots, N$. X and H_i are the discrete Fourier transforms of $x[k]$ and $h_i[k]$, and the asterisk denotes complex conjugation.

Then, the application looks for the maximum values of the correlation functions $g_i[k]$, that indicate when in the received signal is found the pattern associated to a specific beacon i . The beacon 1 is taken as a reference and the TDOA are obtained as follows: the number of sample corresponding to the correlation peak of the beacon 1 is subtracted from the others and the obtained values are aligned subtracting the differences in samples among their emissions (considering the TDMA emission scheme). Then, the difference in samples is translated into time by multiplying it with the inverse of the sampling frequency.

Once the TDOA are computed, the Gauss-Newton algorithm

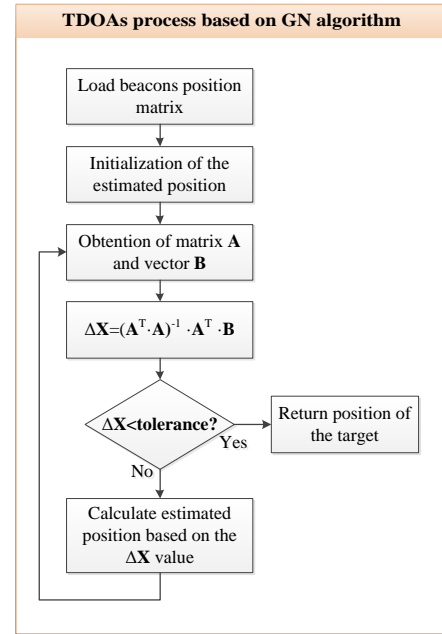


Fig. 8.- Block diagram of the Gauss-Newton (GN) processing for position estimation.

tries to obtain the coordinates $[\hat{x}, \hat{y}]$ that minimize the function represented by eq. (2):

$$F = \sum_{i=2}^l (\hat{r}_{i,1} - r_{i,1})^2 \quad (2)$$

Where $r_{i,1}$ is the difference of distances between the beacon i and the beacon 1 obtained from the TDOAs information and $\hat{r}_{i,1}$ is the estimated difference of distances between the i^{th} beacon and the reference one.

In order to find the optimum values, the algorithm executes eq. 3 until it is satisfied a minimum tolerance value of ΔX (or after a predetermined number of runs).

$$\Delta X = (A^T \cdot A)^{-1} \cdot A^T \cdot B \quad (3)$$

The vector B contains the differences between the estimated data and the measurements and A is the Jacobian matrix with respect to each component of the position of the target.

When the Android class associated with the algorithm is executed (see Fig. 8), the beacon's positions are loaded and it is reserved space to store the auxiliary matrix variables that allows to compute the position. It is an iterative algorithm. In the first execution a position close to the origin of coordinates of the beacons is considered, and the TDOA are included, so in every iteration the position is adjusted until it fulfils the quality criteria ΔX or it reaches a maximum number of iterations.

V. RESULTS WITH REAL SIGNALS

Basic experiments have been carried out with the hardware adaptation module connected to a laptop.



Fig 9.- Localization workspace for experimental tests.

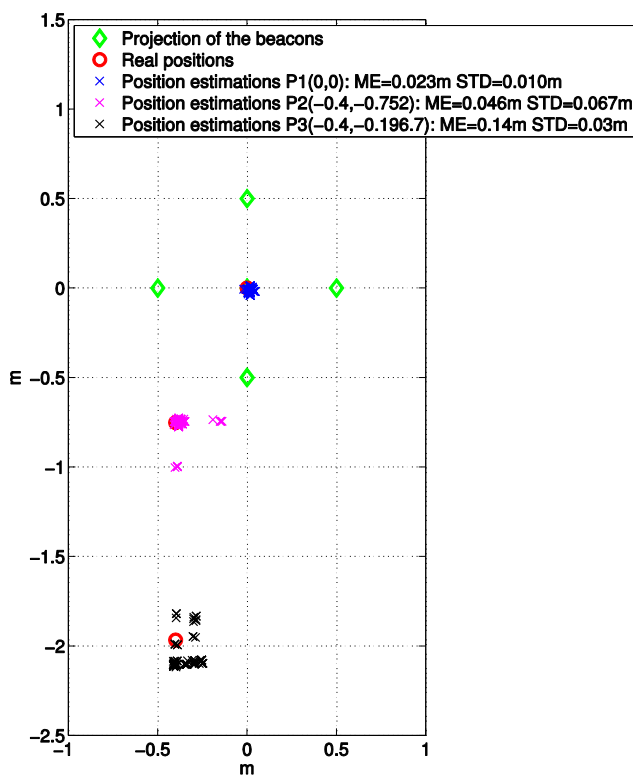


Fig. 10.- Visual results of the measured positions.

Fig. 9 shows the localization workspace for some experimental tests. It includes the proposed U-LPS and three test points inside the coverage area of the emitters. We have acquired 100 measures for each position in order to evaluate the performance of the system. Fig. 10 shows the visual results in 2D of the estimated positions, as well as the statistical information related to the Mean Error (ME) and the Standard Deviation (SD). Fig.

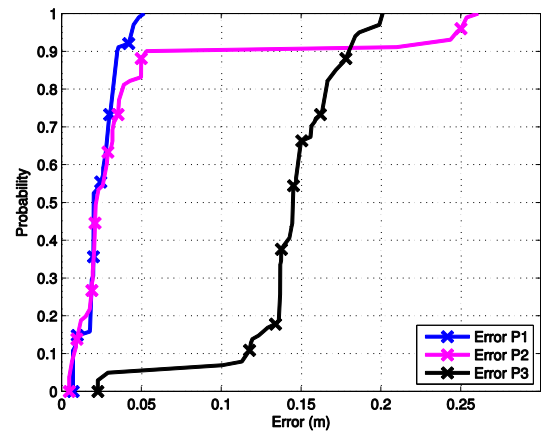


Fig. 11.- CDF of the distance error for each position.

11 shows a CDF of the errors in distance for each test position. The closest positions to the U-LPS presents an error of few centimeters (lower than 4cm in the 80% of cases) and the other positions show an error lower than 17cm in the 80% of cases. The error for position 3 is high because in this case the receiver is in the limit of the U-LPS coverage area. There is also a wall close to the point 3 that can deteriorate the measurements due to multipath effect.

ACKNOWLEDGMENT

This work has been possible thanks to the Spanish Ministry of Economy and Competitiveness (RETO-PLC project, TEC2015-64835-C3-2-R and TARSIVUS, TIN2015-71564-C4-1-R).

REFERENCES

- [1] LPC1768 Product Data Sheet, http://www.nxp.com/documents/data_sheet/LPC1769_68_67_66_65_64_63.pdf, last access September 2016.
- [2] S. Lopes, J. M. N. Viera, J. Reis, D. Albuquerque, N. B. Carvalho, "Accurate Smartphone indoor positioning using a WSN infrastructure and non-invasive audio for TDoA estimation", *Pervasive and Mobile Computing*, vol. 20, pp. 29-46, July, 2015.
- [3] Pro-Wave Electronics Corporation, Air Ultrasonic Ceramis Transducers 328ST/R160, Product Specification, 2014.
- [4] T. Kasami, "Weight distribution formula for some class of cyclic codes", *Combinatorial Mathematics and its Appl.*, Chapel Hill, N.C.: University of North Carolina Press, 1969.
- [5] A. Lindo, M. C. Pérez, J. Ureña, D. Gualda, E. García, J. M. Villadangos, "Proceedings of the 2014 IEEE Conference on Emerging Technologies & Factory Automation (ETFA)", pp. 1-4, Barcelona, Spain, 2014.
- [6] Knowles Acoustics Application Note, SiSonic Design Guide, http://media.digikey.com/pdf/Data%20Sheets/Knowles%20Acoustics%20PDFs/SiSonic_Design_Guide.pdf, last access September 2016.
- [7] M. C. Pérez, D. Gualda, J. M. Villadangos, J. Ureña, P. Pajuelo, E. Díaz, E. García, "Android application for indoor positioning of mobile devices using ultrasonic signals", *Proc. Of 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Alcalá de Henares, Spain, October 2016.
- [8] D. Ruiz, J. Ureña, J. C. García, M. C. Pérez, J. M. Villadangos, E. García, "Efficient Trilateration Algorithm using Time Differences of Arrival", *Sensors and Actuators A: Physical*, 2013.
- [9] Apache Commons, <https://commons.apache.org/>, accessed on May 2016.