A SystemC-based Simulator for Design Space Exploration of Smart Wireless Systems

Gabriele Miorandi[1], Francesco Stefanni[2], Federico Fraccaroli[3], Davide Quaglia[1]
[1]Department of Computer Science, University of Verona, Verona, Italy
[2]EDALab Srl, Verona, Italy; [3]Wagoo LLC, Irving, Texas, USA
davide.quaglia@univr.it

***Abstract*—Smart wireless techniques are at the core of many today's telecommunication and networked embedded systems where performance are enhanced by intertwining radio frequency (RF) and digital aspects. Therefore their design requires to focus on both domains. Traditional approaches for their simulation rely either on different domain-specific tools or on analog-mixed-signal modeling languages. In the former case, the simulation of the whole platform in the same session is not possible while in the latter case, simulation performance is limited by the computationally most intensive domain (usually RF). We present an extension of the SystemC Network Simulation Library that allows to simulate antenna details and node position together with digital hardware and software. The validation on a real wearable system shows that the proposed simulation approach achieves a good trade-off between accuracy and speed thus allowing fast exploration of various configurations in the early phase of the design flow without recurring to the expensive and time-consuming creation of physical prototypes.**

***Index Terms*—Antenna, radio propagation, reconfigurable antenna, beam-forming, MIMO, networked embedded system, wearable system, SystemC**

I. Introduction

Smart wireless systems are networked embedded systems in which performance of wireless transmission is enhanced by using antenna array management (e.g., for beam-forming and MIMO) and antenna reconfiguration to change transmission parameters such as beam direction, frequency and polarization. These systems are at the basis of novel network architectures, e.g., WiFi and 5G [1] as well as future wearable systems [2].

The essential feature of smart wireless systems is the close interaction between the antenna sub-system and the digital sub-system (embedded hardware and software). Algorithms implemented as software and running on the digital platform reconfigure one or more antennas or use received signal to extract context information (not just the transmitted message as usually done).

To clarify the discussion, we introduce a real case study concerning the implementation of a smart eyewear system. The purpose is to determine when user's gaze aims at a Bluetooth radio source, associated, e.g., to a retail item or to a mobile phone, as depicted in Figure 1.a. The patented methodology is based on the comparison of received signal strength on different antennas [3]. The well-known iBeacon standard is used to link source ID with augmented information, e.g., sale information or a Linkedin/Facebook page [4]. Both ad-hoc beacon and smartphones can be used to generate iBeacon advertisements. The architecture is depicted in Figure 1.b. It consists of three dipole antennas, one placed in front of the glasses and others on each lateral temple. When a radio beacon is roughly located in front of the user, the front antenna receives a higher amount of radio power as compared to the left and right antennas because of the donut shape of the dipole radiation pattern. Each antenna is associated with a Bluetooth receiver which continuously scans radio channels looking for nearby advertising beacons and collect their Received Signal Strength Indication (RSSI). The RSSI values are processed by each receiver module and the output value is offloaded to the master module through an I2C bus. The master executes

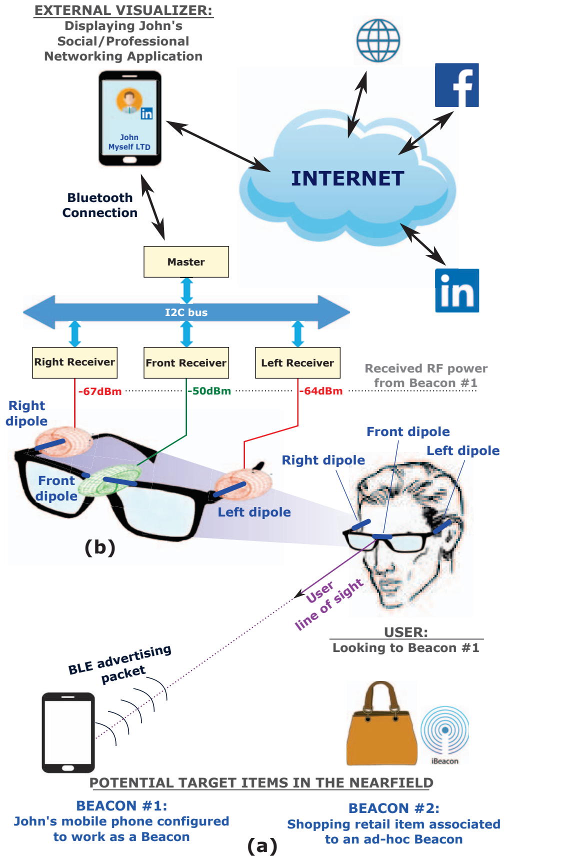

Fig. 1. Description of the case study: (a) behavior (b) system architecture.

The work was partially funded by Regione Veneto "POR CRO, parte FESR, 2014-2020 Azione 1.1.1 bando 2016".

978-3-9819263-0-9/DATE18/©2018 EDAA

the direction estimation algorithm and sends the result to an external device ("visualizer").

There are various design aspects that need to be addressed to implement an effective and efficient solution:

- antenna type: directionality is essential to increase the detection precision and therefore the antenna with the most directive radiation pattern should be chosen;
- antenna position: reception is affected by antenna orientation w.r.t. signal polarization, presence of user's head, relative distance and speed between the user and the Bluetooth source;
- algorithm for RSSI computation: it is implemented as software executed by the baseband receiver module and aims at precisely estimating the amount of electromagnetic energy received on the antenna by taking into account noise and peculiarities of the Bluetooth protocol;
- algorithm for direction estimation: it is implemented as software executed by a CPU and aims at detecting if the beacon is in front of the user with the desired trade off between precision and response time;
- parameters of the digital architecture, e.g., bus transfer rate that may affect size, cost and energy consumption of the wearable device.

The crucial role of functional simulation in electronic system design is witnessed by well-known tools [5] and hardware description languages [6]. Realistic simulations reduce time and cost needed for the creation of a physical prototype during the early design phase when several choices are still open.

An efficient exploration of solutions for smart wireless systems requires the simulation of:

- reception of radio signals as a function of antenna type, distance, direction, polarization and user movements;
- wireless protocols in terms of channel contention, packet size and transmission interval;
- digital hardware such as processing elements and buses;
- software implementation.

In literature, these simulation aspects are addresses by heterogeneous tools and the simulation of the whole system is not possible. Tools for electromagnetic simulation work in the physical or frequency domain [7] and cannot address protocol simulation. Well-known packet-based network simulators [8] do not provide accurate models for digital hardware. Solutions to address multi-domain simulations range from analog-mixed-signal extensions of hardware description languages [9] to co-simulation [10], [11]. Recently, it has been shown that the speed of multi-domain simulation is limited by the computationally most intensive simulation domain, that is often the physical/analog domain [12].

SystemC Network Simulation Library (SCNSL) [13] has been developed as an open source project and provides some interesting features such as primitives to model channels and packet-based protocols with the possibility to simulate computation and communication aspects in a single environment relying on the flexibility of SystemC [14]. In [15] SCNSL is used to simulate efficiently a wireless body area network. Instead of physical-level details, their effects are modeled through a database of performance values based on physical measurements on a human phantom. Such information is embedded into the system-level simulation thus combining speed and fidelity and allowing cross-domain design space exploration.

The paper follows this approach and aims at extending SCNSL to model:

- the reception of radio signals as a function of antenna type, distance, direction and polarization;
- the motion of network nodes (transmitters and receivers).

The paper is organized as follows. Section II presents the modeling problems to be solved for a holistic approach to the simulation of smart wireless systems. Section III describes the proposed extensions. Section IV reports experimental results and finally Section V draws some conclusions.

II. Modeling Aspects

A. Signal Propagation

RF signal propagating between a transmitting and a receiving node is subject to *delay* and *attenuation*. Delay t depends on signal speed (approximately equal to light speed c) and distance d according to (1):

$$t = \frac{d}{c} \tag{1}$$

Signal attenuation depends on various causes; among them the so-called *path loss* is always present and due to the spread of energy on a large volume around the transmitter. The path loss between a transmitter T and a receiving antenna R can be described by the well known Friis equation [16] in (2):

$$P_R = PLF \cdot \frac{P_T G_T G_R c^2}{(4\pi d f)^2} \tag{2}$$

This equation computes the received power (P_R) as a function of parameters related to the transmitter (i.e., transmitted power P_T, frequency f, and transmitter antenna gain G_T), the channel (i.e., propagation speed c and distance d between T and R) and the receiver (i.e., receiver antenna gain G_R).

Even if the total amount of energy radiated in the space by the antenna is constant for different antenna designs having the same input power, for each antenna type the local radiated energy, and thus the gain, is a function of the radiation angle of arrival (AoA). The 3D representation of antenna gain as a function of direction is called *radiation pattern*. The shape of a radiation pattern depends on the type of antenna. For instance, Figure 4 shows the toroidal, or donut-shaped, radiation pattern of an ideal dipole antenna. In this case, along the z-axis, which would correspond to the radiation directly overhead the antenna, there is very little power transmitted/received. In the x-y plane (perpendicular to the z-axis), the radiation is maximum. Typically, radiation patterns are plotted as 2D "slices". Standard spherical coordinates are used, where polar angle θ is the angle measured from the z-axis, and ϕ, is the azimuth angle, i.e., the angle measured counter-clockwise from the x-axis. Since these plots are useful to visualize the principal directions for the antenna signal radiation/reception, they are usually employed to properly place static antennas; however their evaluation in dynamic scenarios is interesting for the optimal design of mobile device.

Relative misalignment between the oscillation planes of T and R antennas generates further energy loss due to *polarization*. Modeling the polarization loss is particularly important in a dynamic scenario, where nodes move and rotate. In the Friis equation this is performed by the Polarization Loss Factor ($PLF \in [1, 0]$). Given a pair of linearly polarized antennas (e.g., dipoles) the received power is maximized only if the oscillation planes of the electric vector for the two antennas are parallel, while exchanged energy is theoretically null if the planes are perpendicular. Being γ the angle between the oscillation planes of two linearly-polarized antennas, $PLF = cos^2(\gamma)$ is a good loss model for them.

B. Smart Antennas

A radiation pattern can be associated either to a single antenna or to a set of antennas which are managed together as in case of arrays. In this paper we use the term *antenna* to denote a physical system consisting of one or more radiating elements connected together by non-digital components such as power splitters, RF switches or delay elements. We aim at representing the *resulting radiation pattern* as determined by using electro-magnetic simulators or experimental measurements.

If such radiating elements are digitally reconfigured, it is possible to change antenna properties (e.g., gain, radiation pattern and polarization) at run time. In *reconfigurable antennas* the change of the primary propagation direction (also called "beam") can be useful to avoid interference with other transmissions thus increasing the channel capacity by adding a spatial dimension to the communication multiplexing concept. Furthermore, the beam can even follow the interacting node in a mobile scenario. Also polarization can be dynamically adjusted either to reduce interference and increase multiplexing or to maximize the energy transfer towards the receiver when it rotates because of user's movement.

Reconfigurability is the core of smart wireless systems and, from the modeling point of view, it requires to simulate together digital and RF sub-systems. In the proposed simulator, the antenna gain, radiation pattern and polarization can be changed at run time as a result of actions made by the digital part of the model or by the software.

C. Node Movements

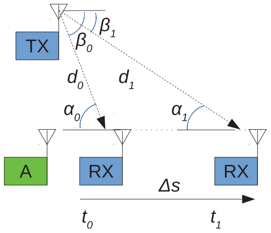

Fig. 2. Example of scenario changes due to node movement.

Figure 2 shows a mobile scenario in which the transmitter is static while a receiver translates towards the right. Considering that the speed of node is much lower than signal speed (i.e., light speed), the main effect of node movement is the change of distance and of the angles of the radio signal w.r.t. antennas. Transmission delay changes proportionally to distance while received power changes according to Friis equation in (2) where the parameters d, G_T, G_R, and PLF become time varying. Delay and received power variations affect packet reception and collision events (i.e., when two or more signals overlap at the receiver thus invalidating packets). In fact a receiver may be listening in a given time interval only and, because of the delay, the packet may arrive at receiver's antenna when the node is not in the listening state. Also collisions may happen according to the specific time in which the packet arrives. Since receiving equipments are not able to detect signals whose power is below a given value (named *sensitivity threshold*) a change in the received power due to movements can affect the possibility to receive a packet or to cause a collision. With reference to the example in Figure 2, if A is already transmitting, in t_0 a collision arises while in t_1 a collision is not possible provided that A's signal power falls below the sensitivity threshold of RX.

All the phenomena described in this section should be taken into account in the simulator to reproduce the effect of RF propagation and node movement.

III. Extended Simulator

As described in the previous section, the effects of RF propagation and node movements are quite intertwined and cannot be addressed separately. SystemC Network Simulation Library has been adopted since it already allows to simulate network protocols and digital hardware [14]. The extensions regards the definition of geometrical coordinate systems for node position and antenna orientation, the modeling of the radiation pattern and the management of movements. They are described in the following sections.

A. Coordinate Systems

As far as antenna movements concern, a three-dimensional vector is used to move the node in the Euclidean space. The spatial point defined by this vector is the origin of three orthogonal unit vectors that define a local coordinate system for the node, by which it can be rotated around the position point in order to model different antenna orientation. From now on we refer to those coordinate systems as Global Coordinate System (GCS) (standard $[x, y, z]$ Cartesian coordinate) and Antenna Coordinate System (ACS), respectively.

Accordingly to the theory discussed in Section II, part of the Friis equation (2) has been inserted into the SCNSL channel object (such as distance information and propagation speed), while a new class of node properties called *RadiationPattern* has been introduced to model antenna operation, its polarization and gain function. Antenna polarization and gain function

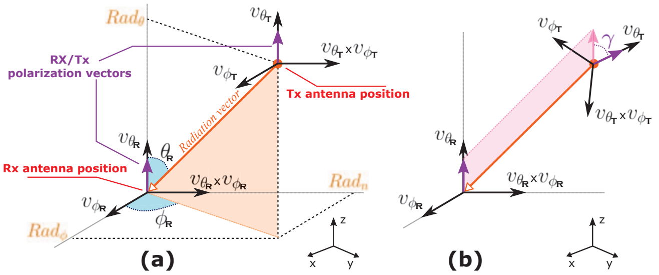

Fig. 3. Possible configurations for equal node position: antenna rotation with (a) aligned and (b) unaligned polarization.

are anchored to the ACS. There is an ACS for each antenna, its origin corresponds to the antenna position and its rotation its defined by the orthogonal base $[v_\theta, v_\phi, v_\theta \times v_\phi]$. AoA with respect to ACS is also is defined in spherical coordinate $[\theta, \phi]$ where θ is the angle measured from v_θ and ϕ is the angle measured from v_ϕ. Any modification of any ACS will affect the orientation antenna polarization and the antenna gain to be considered in the estimation of the total received power as described in the example below.

Figure 3 depicts two possible working scenarios. In both cases the Rx/Tx antennas have the same position $P_{Rx} = [x_{Rx}, y_{Rx}, z_{Rx}]$ and $P_{Tx} = [x_{Tx}, y_{Tx}, z_{Tx}]$. For each antenna, the polarization vector matches the direction of versor v_θ. In Figure 3.a the radiation emitted by Tx is aligned with Rx polarization; vice versa in Figure 3.b the polarization is unaligned because there is an angle $\gamma \neq 0$ between Rx and Tx polarization vectors. In this latter case a polarization loss will be experienced since $PLF < 1$.

To estimate the antenna gain (function of the AoA) the simulator starts calculating the radiation vector $Rad = P_{Rx} - P_{Tx}$, that represents the direction by which the Rx antenna is receiving the radiation. Its AoA at the receiver side is found by changing base of Rad from $[x, y, z]$ to $[v_{\theta R}, v_{\phi R}, v_{nR}]$ (where $v_{nR} = v_{\theta R} \times v_{\phi R}$ is the missing orthogonal versor of the ACS) and then calculating $[\theta_R, \phi_R]$ angles by applying simple trigonometric equations:

$$\phi_R = atan\left(\frac{Rad_n}{Rad_\phi}\right) \qquad \theta_R = atan\left(\frac{\sqrt{Rad_\phi^2 + Rad_n^2}}{Rad_\theta}\right)$$

.

After θ_R and ϕ_R have been calculated, the antenna gain parameter G_R can be estimated. Because of reciprocity, G_T is similarly computed at the Tx side. Rx and Tx antenna gains, together with PLF, complete the Friis equation and allow a proper estimation of the received power P_R in mobile scenarios.

B. Radiation Patterns

The proposed simulator allows to model antenna gains either analytically or numerically. For example, considering the ideal dipole radiation pattern in Figure 4, the analytical model is $G = sin^2(\theta)$. In this case the polarization vector lays on v_θ. Actual antenna radiation patterns are usually expressed only for a subset of all AoA values. In this case the designer can provide the numerical pattern shape as an input CSV file where any line represents a radiation point $[\theta_i; \phi_i; Gain]$. During simulation the gain for the specific AoA is computed either by using the analytical expression or by interpolating numerical values according to one of the following approaches:

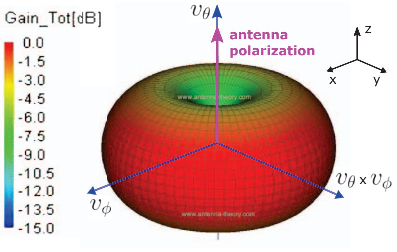

Fig. 4. Ideal radiation pattern of a dipole antenna.

Complete Pattern: a gain value is defined for a grid of AoA with $\theta = [\theta_0, ..., \theta_i]$ and $\phi = [\phi_0, ..., \phi_j]$. Using this specification any AoA value falls into a quadrant. Interpolated gain is calculated as the average gain of the four surrounding corners delimiting the quadrant weighted by the angular distance between the radiation vector and the corner $[\theta, \phi]$ vectors (Figure 5.a).

Incomplete Pattern: this model includes just a subset of gain values. Given the AoA, finding the nearest input points may be a computationally intensive task; therefore gain estimator is the weighted average among a set of input points in the proximity of the AoA (Figure 5.b)).

Orthogonal Planes Pattern: this includes a reduced number of gain values, strictly belonging to the three orthogonal planes $v_\theta - v_\phi$, $v_\theta - n$ and $v_\phi - n$. This pattern format is quite common in commercial antenna datasheets. In this case, the AoA vector is projected on all the three orthogonal planes, then antenna gain is estimated considering the closest input values on each plane and weighting by the distance between the radiation vector and the orthogonal planes (Figure 5.c).

IV. Experimental Results

The smart glasses system described in Figure 1 has been used to validate SCNSL extensions and to show their potential in design space exploration. This case study has been chosen because it is the subject of a real design effort such that we know all the details to build the models and simulate them.

A. Case Study

Figure 6 shows the simulation scenario. A Bluetooth iBeacon rotates around smart glasses equipped with three quarter-wave dipole antennas whose radiation pattern is depicted in Figure 4. The presence of the user's head leads to a -8 dB attenuation of antenna beams towards the head. The angular speed of the iBeacon is 4.5 degree/s.

The purpose of a Bluetooth iBeacon is to "advertise" its presence to the neighbor Bluetooth "scanners". iBeacons advertise on three specific channels, i.e., 37, 38 and 39. The

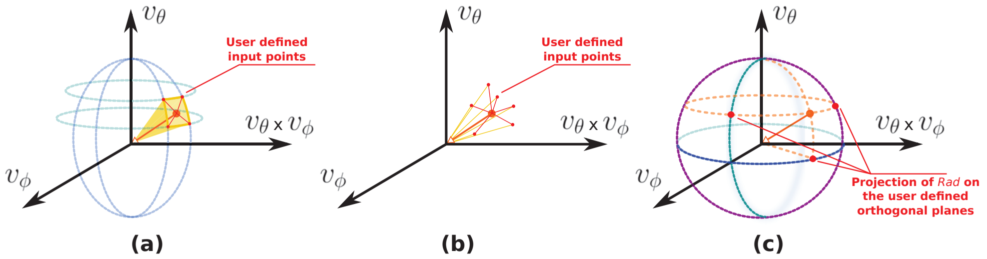

Fig. 5. Interpolation approaches for numerical radiation patterns.

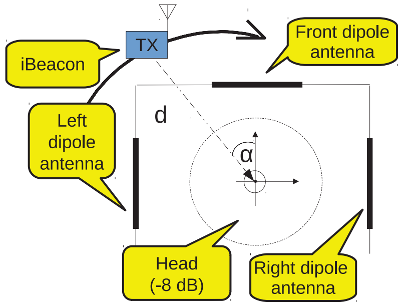

Fig. 6. Simulation scenario.

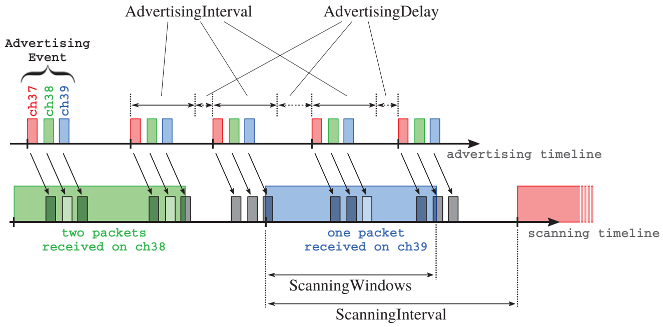

Fig. 7. Bluetooth iBeacon protocol.

advertising rate depends on the power budget of the iBeacon, e.g., ultra-low power “coin” device vs. mobile phone. Figure 7 shows the timing diagram for the Bluetooth transmitter performing advertising (top) and the receivers performing scanning (bottom). An advertising event consists in sending a packet through channels 37, 38 and 39. The time interval between consecutive advertising events is the sum of a fixed *AdvertisingInterval* and a random *AdvertisingDelay* to reduce packet collisions. The beacon vendors currently span the power/responsiveness trade-off by enlarging or shortening this interval. The receivers consecutively scan the three channels. Smart glasses power efficiency depends on setting *ScanningWindow* and *ScanningInterval* delays. For instance, by setting ScanningWindow=ScanningInterval the receiver works in “continuous scanning mode” with high responsiveness and energy consumption.

B. Modeling and Performance

Figure 8 shows how the SCNSL primitives have been used to model the architecture depicted in Figure 1.b. The iBeacon is modeled by using a `Node` associated to a `Task` implementing the iBeacon advertising protocol and a model of omni-directional antenna. The antenna is bound to three `SharedChannels` to model channels 37, 38 and 39. Each Bluetooth receiver (front, left, and right) is modeled as a `Node` associated to an analytical model of quarter-wave dipole. Each dipole model is bound to all Bluetooth channels. Each receiving node host two `Tasks` to implement the iBeacon scanning protocol and the I2C slave protocol, respectively. Also the I2C bus is modeled as a `SharedChannel` but without antenna models. It is worth noting that the SCNSL `SharedChannel` models a multi-point channel which can be wired or wireless. The I2C master is modeled as a `Node` with a `Task` implementing the I2C polling protocol and the algorithm that detects if the iBeacon is in front of the user.

The RSSI power value measured by the front receiver as a function of the angle is reported in Figure 9 and compared with the experimental value found on a real setup in the anechoic chamber. Values for the other two receivers are equal except for 90 degrees rotation.

The simulation of 1,000 rounds with 1 degree granularity (i.e., 360,000 movements) took 12.62 s on an Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz with 8GB RAM and Linux Ubuntu 16.04 LTS.

C. Demonstration

To show the potential of the simulation framework, we used it to evaluate the direction estimation algorithm which runs on the master node. The algorithm uses the average RSSI values from the three receivers (i.e., front, left and right) and it determines whether the beacon is placed in front of the user by checking:

$$I_{front} = RSSI_{front} - max(RSSI_{left}, RSSI_{right}) > Thres$$

The algorithm is independent from distance between the user and the beacon because absolute RSSI values decrease with distance but their ranking order is preserved.

Figure 10 shows simulation results for different power-related metrics as a function of time and beacon angle (rotation speed is constant). First, RSSI measured by the receiver and sampled by the I2C master are shifted in time because of digital hardware latency. The evaluation of this delay as a function of architectural choices is important for the design. Second, the comparison between $RSSI_{front}$ and I_{front} shows that the latter allows a higher angular sensitivity. In this plot,

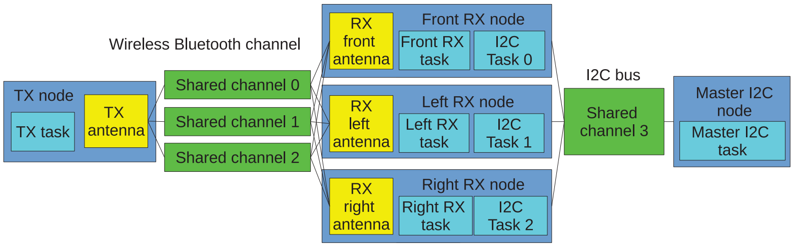

Fig. 8. Model of smart glasses in the extended SCNSL.

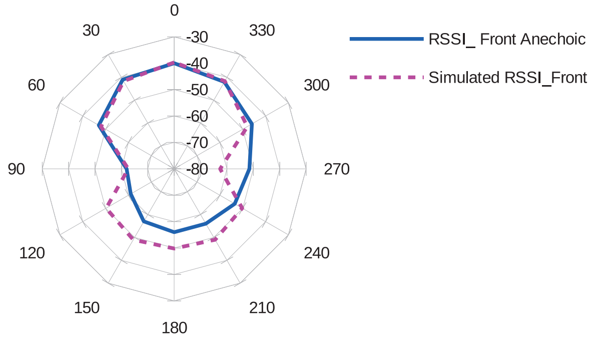

Fig. 9. Simulation of received signal power compared with real values experimentally found in the anechoic chamber.

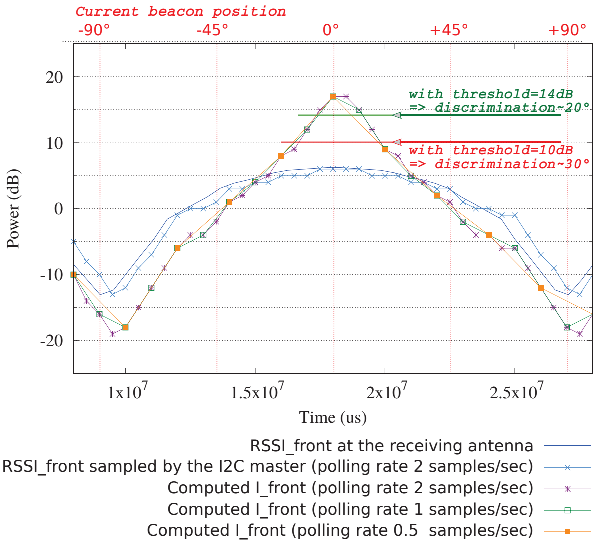

Fig. 10. $RSSI_{front}$ and I_{front} as a function of the angle of arrival and the I2C bus polling rate.

the choice of the antenna type and the effect of software can be jointly evaluated. Third, the plot shows that the application developer can restrict (or expand) the angle under which the beacon is considered centered by increasing (or decreasing) the value of $Thres$. Finally, the plot shows the relationship between detection speed and I2C polling rate which, in turn, affects the energy consumption of the wearable device. Even this simple experiment shows the contribution of the tool for the joint evaluation of very different aspects.

V. Conclusions

We presented an extension of SystemC Network Simulation Library to address the simulation of smart wireless systems where RF and digital modeling are strictly related. We introduced the possibility to reproduce the effect of RF signal propagation with emphasis on antenna modeling in mobile scenarios. The proposed approach allows the modeling of radiation patterns that can be expressed both analytically and numerically. Additionally, polarization is taken into account. The application to a real case study shows that the proposed simulation approach allows to perform effective design space exploration by covering RF aspects, network protocols, movements, digital hardware and software.

References

[1] E. J. Black *et al.*, “Software defined apertures for 5G wireless network communications,” in *2015 IEEE Conference on Standards for Communications and Networking (CSCN)*, Oct 2015, pp. 124–129.

[2] H. Wong *et al.*, “Multi-polarization reconfigurable antenna for wireless biomedical system,” *IEEE Trans. on Biomedical Circuits and Systems*, vol. 11, no. 3, pp. 652–660, June 2017.

[3] F. Fraccaroli and B. Bochicco, “Method, system and apparatus for the augmentation of radio emissions,” Jul. 28 2015, US Patent 9,092,898.

[4] Apple, “iBeacons for developers,” 2017, URL: https://developer.apple.com/ibeacon/.

[5] Cadence, “Virtual System Platform,” www.cadence.com.

[6] OSCI and IEEE, “IEEE Std 1666 - 2005 IEEE Standard SystemC Language Reference Manual,” *IEEE Std 1666-2005*, pp. 1–423, 2006.

[7] J. Hoffmann *et al.*, “Comparison of electromagnetic field solvers for the 3d analysis of plasmonic nanoantennas,” *Proc.SPIE*, vol. 7390, 2009.

[8] H. Wu *et al.*, “An Energy Framework for the Network Simulator 3 (NS-3),” in *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, 2011, pp. 222–230.

[9] M. Vasilevski, F. Pecheux, N. Beilleau, H. Aboushady, and K. Einwich, “Modeling and Refining Heterogeneous Systems With SystemC-AMS: Application to WSN,” in *IEEE/ACM DATE*, 2008, pp. 134–139.

[10] Mentor Graphics, “Questa Advanced Simulator,” www.mentor.com/products/fv/questa.

[11] D. Quaglia, R. Muradore, R. Bragantini, and P. Fiorini, “A SystemC/Matlab co-simulation tool for networked control systems,” *Simulation Modelling Practice and Theory*, vol. 23, pp. 71–86, 2012.

[12] M. Lora, S. Vinco, E. Fraccaroli, D. Quaglia, and F. Fummi, “Analog models manipulation for effective integration in smart system virtual platforms,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. PP, no. 99, pp. 1–1, 2017.

[13] D. Quaglia and F. Stefanni, “SystemC Network Simulation Library – version 2,” 2013, URL: http://sourceforge.net/projects/scnsl.

[14] P. Sayyah *et al.*, “Virtual platform-based design space exploration of power-efficient distributed embedded applications,” *ACM Trans. Embed. Comput. Syst.*, vol. 14, no. 3, pp. 49:1–49:25, Apr. 2015.

[15] M. Crepaldi *et al.*, “A physical-aware abstraction flow for efficient design-space exploration of a wireless body area network application,” in *Euromicro Conf. on Digital System Design*, Sept 2013, pp. 1005–1012.

[16] S. R. Bullock, *Transceiver and System Design for Digital Communications*, 2nd ed. Noble Publishing Corporation, 2000.