

Synthetic Radar Signal Generator for Human Motion Analysis

Evert I. Pocoma Copa¹, Hasan Can Yildirim², *Graduate Student Member, IEEE*,
Jean-François Determe³, and François Horlin⁴, *Member, IEEE*

Abstract—Synthetic generation of radar signals is an attractive solution to alleviate the lack of standardized datasets containing paired radar and human-motion data. Unfortunately, current approaches in the literature, such as SimHumalator, fail to closely resemble real measurements and thus cannot be used alone in data-driven applications that rely on large training sets. Consequently, we propose an empirical signal model that considers the human body as an ensemble of extended targets. Unlike SimHumalator, which uses a single-point scatterer, our approach locates a multiple-point scatterer on each body part. Our method does not rely on 3-D-meshes but leverages primitive shapes fit to each body part, thereby making it possible to take advantage of publicly available motion-capture (MoCap) datasets. By carefully selecting the parameters of the proposed empirical model, we can generate Doppler-time spectrograms (DTSs) that better resemble real measurements, thus reducing the gap between synthetic and real data. Finally, we show the applicability of our approach in two different application use cases that leverage artificial neural networks (ANNs) to address activity classification and skeleton-joint velocity estimation.

Index Terms—Activity classification, data generation, human motion, pose estimation, radar, synthetic data.

I. INTRODUCTION

RADAR sensors have become increasingly relevant in applications that leverage machine learning (ML) to solve challenging problems across many domains including automotive safety, surveillance, and health monitoring [1]. These solutions heavily rely on standardized datasets, which facilitates consistent training, evaluation, and reproducibility. However, collecting radar datasets is a time-consuming and labor-intensive process.

In the specific context of human-related applications, there is a notable lack of standardized datasets that provide both radar measurements and corresponding body-part trajectories. While publicly available motion capture (MoCap) datasets provide human motion for various activities (such as those provided by [2] and [3]), they do not include radar data. Consequently, generating synthetic radar signals using these MoCap (or Kinect) datasets directly alleviates the need for

extensive and standardized datasets for human motion analysis with radars.

In general, synthetic radar signals can be generated using either data-driven or signal-model-driven approaches. Data-driven methods rely on ML architectures, such as generative adversarial networks (GANs) and variational autoencoders (VAEs). They generate new training-like data by sampling from the learned data distribution, as shown in [4] and [5]. However, this approach is mainly used for data augmentation, since it still requires an initial training dataset. In contrast, signal-model-driven methods are required to accurately model the reflected signal received by the radar. This involves knowing the target trajectories in advance, which is easily met using external datasets, for example, MoCap. Therefore, this article focuses on the radar-signal-model-driven approaches.

The simplest radar-signal models for human-motion analysis are generally composed of two steps [6], [7], [8], [9], [10]. First, the human body is treated as an extended target, divided into multiple body parts. Second, a single-point scatterer is *fixed* on each body part. Thus, the final signal model is the contribution of all considered points scaled by the path gain [that is based on each body-part distance to the radar and radar cross section (RCS)]. Different studies locate these scatterers in different locations. For instance, [6] and [7] place scatterers at the body-part centers. Extra scatterers are added in [7] at the end of limbs (e.g., tips of hands and feet). Similarly, [8] and [9] place scatterers at the joints between body parts. Lastly, [10] uses arbitrary locations on the surface of each body part.

Unfortunately, derived maps [such as the Doppler-time spectrogram (DTS)] computed from these approaches fail to closely resemble the ones from real measurements. Thus, they cannot be used alone for training a artificial neural network (ANN). For instance, a pretrained ANN called the *denoising network* was used in [1] as a workaround to this issue. In short, it converts the input data from *measurement* to *synthetic* domain leveraging VAEs. Thus, the ANN's training and evaluation are always performed in the synthetic domain. Nevertheless, one still needs to collect a large-enough dataset to train the denoising network in the first place.

This resemblance gap is partially due to the absence of modeling multipaths and interbody-part occlusion, as the previously mentioned approaches [6], [7], [8], [9] only consider the line-of-sight (LOS) case. A second contributor to this gap is the body-part representation with a single scatterer. Consequently, more complex signal models have been proposed to address these issues [11], [12].

Received 3 July 2024; revised 29 September 2024 and 30 November 2024; accepted 12 December 2024. Date of publication 17 December 2024; date of current version 30 December 2024. (Corresponding author: Evert I. Pocoma Copa.)

The authors are with the OPERA-Wireless Communication Group (WCG), Université Libre de Bruxelles, 1050 Brussels, Belgium (e-mail: epocomac@ulb.be; hasan.can.yildirim@ulb.be; jean-francois.determe@ulb.be; francois.horlin@ulb.be).

Digital Object Identifier 10.1109/TRS.2024.3519138

For instance, 3-D-mesh vertices are used as scatterer locations in [11]. Although this approach increases the model accuracy, it requires the use of other software tools (e.g., Blender). More importantly, it requires working with many points (several thousand per 3-D-mesh), making them computationally impractical [13]. A simpler approach was proposed in [12], where additional samples are placed over the major axis of an ellipsoid fit to each body part. Both approaches [11], [12] assess whether the radar-point LOS path is occluded by other body parts or not. Nevertheless, only the approach using a 3-D-mesh also considers occlusion by the body part itself.

Therefore, in this article, we propose an alternative that does not rely on 3-D-meshes. Instead, we rely on the primitive shapes (ellipsoids) fit to each body part. Although ellipsoids are not an accurate representation of the human-body shape, they allow us to reproduce measurement-like radar data more accurately than current methods in the literature.

In summary, we introduce an empirical signal model for human motion analysis that can be used to synthetically generate radar data from MoCap or Kinect datasets. More precisely, the contributions of this article are threefold as follows.

- 1) First, we propose to consider each individual body part as an extended target, that is, we use multiple-point scatterers per body part. Different from [6] and [12], the ellipsoids are not only used to compute the RCS, but also to determine the normal direction of the reflective interface, and the location of point scatterers. We place all samples over the ellipsoid's surface, and around the specular point, which we define as the point whose reflected signal to the radar is purely due to specular reflection. In addition, we also take into account the interbody-part occlusion, as well as the multipath cases due to walls in the scene. The resulting approach captures better the human-activity signature in the DTS compared to [6] (that considers only the body-part centers).
- 2) The proposed empirical model introduces three parameter sets: number of scatterers, sampling spread, and strength coefficients. They define how numerous, concentrated, and relevant the scatterer's points are. Therefore, as a second contribution, we propose a two-step approach to properly select these parameter sets. In the first step, the number of scatterers is selected as the minimum needed to capture all body-part velocities, while the rest of the parameters are momentarily set to values that ensure uniform relevance and maximum spread. In the second step, we find the sampling spread and strength coefficients by minimizing the difference between synthetic and measurement DTSs. As a result, the final parameters make it possible to synthetically generate DTSs with similar characteristics as real measurement.
- 3) Lastly, we show the applicability of using the synthetic data generated with our approach in two application use cases that rely on ANN: activity classification and pose estimation. In the first, we train a vanilla convolutional neural network (CNN) model to classify four human

activities. In the second, we make use of the two-step architecture proposed in [1] to estimate 3-D velocities of all joints, later used in the pose estimation. In addition, we train all models with synthetically generated DTSs using the signal model SimHumalator [6] and our empirical model. As a result, we show that in both applications, the ANN trained with data generated with our approach, generalizes better to measurements, even when the synthetic equivalent of the measured DTS was not included in the training set.

In this article, we consider the following notation. $\langle \cdot \rangle$ and $\| \cdot \|$ denote the inner product and L2 norm, respectively. \odot denotes the Hadamard (elementwise) product.

II. FUNDAMENTALS

In this section, we progressively describe the signal model of an frequency-modulated continuous wave (FMCW) radar. We first address the case of a single-point scatterer. Subsequently, we show its application in modeling human movement, as introduced in [6] and [14].

A. Radar Signal and Single-Point Scatterer

The transmitted wave of an FMCW radar, also known as chirp, is represented at time instant t by $s_T(t)$. The chirp $s_T(t)$ interacts with the objects present in the scene, thus part of the transmitted energy is reflected back to the radar. As a result, the received signal $s_R(t)$ is a modified version, in amplitude and phase, of the transmitted signal $s_T(t)$. Notice that the radar sends a frame composed of a repeated chirp sequence.

The received signal $s_R(t)$ is then mixed with a replica of the transmitted signal $s_T(t)$ and passed through a low-pass filter. The resulting signal, known as the intermediate frequency (IF) signal $y(t)$, is defined for the simplest case of a moving point scatterer- b as [15]

$$y(t) = s_b(t, \mathbf{p}_b(t)) + n(t) \quad (1)$$

where $n(t)$ is the additive noise, $\mathbf{p}_b(t)$ is the scatterer- b position at time instant t expressed in Cartesian coordinates, and $s_b(t, \mathbf{p}_b(t))$ is defined as

$$s_b(t, \mathbf{p}_b(t)) = a_b(t, \mathbf{p}_b(t)) e^{2\pi j(f_0 \tau(\mathbf{p}_b(t)) + \mu t \tau(\mathbf{p}_b(t)))} \quad (2)$$

where μ and f_0 are the frequency slope and initial frequency of the chirp, respectively. The round-trip time delay is denoted by τ and the complex path gain by a_b . More precisely, they can be modeled as

$$\tau(\mathbf{p}_b(t)) = \frac{d_1(\mathbf{p}_b(t)) + d_2(\mathbf{p}_b(t))}{c} \quad (3a)$$

$$a_b(t, \mathbf{p}_b(t)) = G \frac{\sqrt{\sigma_b(t)}}{d_1(\mathbf{p}_b(t))d_2(\mathbf{p}_b(t))} \zeta_b(t) \quad (3b)$$

where c is the speed of light, $\sigma_b(t)$ is the RCS associated with the point scatterer- b , and G is a term that models various effects such processing gains [6]. The denominator in (3b) models the attenuation due to the path loss, where $d_1(\mathbf{p}_b(t))$ and $d_2(\mathbf{p}_b(t))$ are the propagation distances from radar-to-scatterer and for the returning path, respectively.¹

¹We make the distinction between d_1 and d_2 since the path trajectories might be different when considering multipaths. Moreover, $d_1 = d_2$ only for the LOS path case.

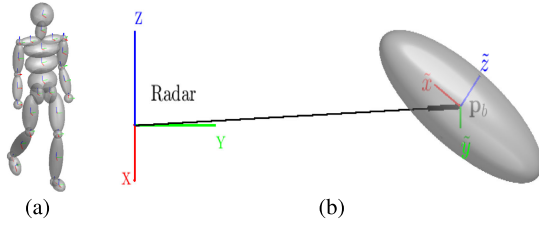


Fig. 1. Human model provided by MoCap datasets. (a) Human body divided into several body parts, each of them represented by a fit ellipsoid. (b) SimHumalator approach: one sample \mathbf{p} at the center of each body part.

Lastly, the effect of the antenna pattern is modeled by $\zeta_b(t) = \mathbf{F}^H(\text{AoA}_b)\mathbf{T}_b\mathbf{F}(\text{AoD}_b)$, where \mathbf{F} is the radar complex antenna field pattern, while AoA and AoD are, respectively, the angles of arrival and departure, potentially depending on the time. In addition, matrix \mathbf{T} describes the transformation that the radiated field undergoes until it reaches the receiving antenna [16]. We kindly refer the reader to [16] for an in-depth definition of matrix \mathbf{T} for different types of reflections. Alternatively, a simpler electromagnetic (EM) primer can be found in the documentation of project Sionna [17]. Moreover, in this article, we will only consider specular and diffuse reflections.

B. Primitive Shapes—Ellipsoids

To define the IF signal due to a person moving in the scene, the human body \mathcal{B} is considered to be composed of N_B body parts.² Following the approach described in [6], each body part is approximated in volume by a primitive shape, as shown in Fig. 1(a). They are usually chosen to be ellipsoids [6], [14]. Our work leverages these primitive shapes to compute multiple sampling points. Hence, we will shortly outline some useful properties and definitions.

The ellipsoid associated with body part- b is denoted as \mathcal{E}_b . A point \mathbf{p} is usually expressed in the global frame, whereas $\tilde{\mathbf{p}}$ denotes the same point in the local frame of some body part- b (see Appendix A-A). The conversion from local to global frames is defined by the transformation matrix $\mathbf{G}_{\mathcal{G},b}$ in the Appendix A [see (17)]. Notice that (\cdot) is used to denote that a vector or variable is expressed in this local frame. When that notation is not used, we express all vectors in the global frame, which is common to all objects in the scene.

The function $h_b(\tilde{\mathbf{p}})$ returns the normal $\tilde{\mathbf{n}}$ of a given point $\tilde{\mathbf{p}}$ over ellipsoid \mathcal{E}_b . Conversely, $h_b^{-1}(\tilde{\mathbf{n}})$ returns the point on the ellipsoid that has the normal $\tilde{\mathbf{n}}$. Both $h_b(\cdot)$ and $h_b^{-1}(\cdot)$ are defined in Appendix A-C. Moreover, we provide extended details of the concepts related to the ellipsoids in Appendix A.

C. State-of-the-Art: SimHumalator

We can now define the signal model introduced in [6]. Since the human body is considered to be composed of N_B body parts, a one-point scatterer is placed at the center of each body part, as illustrated in Fig. 1(b). As a result, the IF signal $y_S(t)$

is defined as the contribution of all point scatterers,³ that is,

$$y_S(t) = \sum_{b \in \mathcal{B}} s_b(t, \mathbf{p}_b(t)) + n(t). \quad (4)$$

Notice that the associated RCS $\sigma_b(t)$, needed to define the complex path gain $a_b(t, \mathbf{p}_b(t))$ in (3b), can now be derived from the primitive shape [14], and it is defined in (19).

To use (4) to synthetically generate $y(t)$, one needs to know the position of each body part at all times. To fulfill this condition, the body-part center $\mathbf{p}_b(t)$, expressed in the global frame, is usually computed from Kinect or any MoCap dataset, such as the one in [2]. Consequently, the synthetic signal $y(t)$ can later be used to compute derived information, such as the range-Doppler map (RDM) or DTS with traditional processing techniques, as the ones detailed in [18]. Notice that one frame of chirps can be postprocessed into one RDM, whereas several frames are needed to compute a single DTS.

III. EMPIRICAL SIGNAL MODEL FOR HUMAN-MOTION ANALYSIS

In this section, we introduce a new empirical model to generate synthetic radar signals that closely resemble measurements. Initially, we provide a general introduction to our model that takes into account multiple scatterer points and interbody-part occlusion. We then dive into more specific details regarding the scatterer placement and evaluation of occlusion. Lastly, we expand our model to incorporate multipath effects.

Unfortunately, the synthetic signal generated with (4) still presents nonnegligible differences compared to real measurements (we will show and analyze those differences in Section V). Subsequently, those synthetic signals cannot be used alone for training an ANN,⁴ or any other model requiring an extensive and quality dataset.

We argue that the main reason for this difference is the assumption that the transmitted chirp is reflected on a single point per body part, whereas, in reality, the signal is reflected from multiple points. Moreover, these reflective points might be different from the body-part center \mathbf{p}_b .

Consequently, we propose the empirical signal model $y_E(t)$

$$y_E(t) = S(t, \mathcal{Q}, \mathcal{R}) + n(t) \quad (5)$$

with

$$S(t, \mathcal{Q}, \mathcal{R}) = \sum_{b \in \mathcal{B}} w_b \left[I_b s_b(t, \mathbf{q}_b(t)) + \sum_{i=1}^{N_b} I_{b,i} s_b(t, \mathbf{r}_{b,i}(t)) \right] \quad (6)$$

where \mathbf{q}_b is referred to as the specular point over the body part- b , and $\mathbf{r}_{b,i}$ is the i th diffuse point over the body part- b sampled around \mathbf{q}_b (see Fig. 2). Hence, \mathcal{Q} and \mathcal{R} are the sets of all points \mathbf{q}_b and $\mathbf{r}_{b,i}$, respectively. Note that N_b additional diffuse points are drawn per body part.

³The signal model (4) was introduced first in [6] under the name of *SimHumalator*, hence the subscript $(\cdot)_S$ in $y_S(t)$.

⁴The gap difference was indirectly shown in [19, Figs. 3 and 5], where both synthetic and measurement DTSs are clearly different, for example, the synthetic DTS is highly concentrated compared to the ones from measurements.

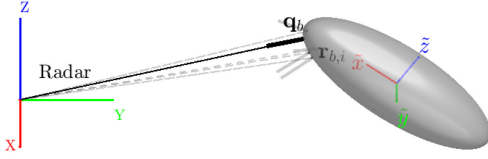


Fig. 2. Proposed sampling strategy: the specular point (in black) has a normal that points toward the radar. The diffuse points (in gray) are located around the specular point (radar located at the origin of the global frame). The normal direction is represented as thick lines over the points.

Notice that (6) makes use of \mathbf{q}_b rather than the body-part center \mathbf{p}_b . This change is mainly because the main reference point must be located in the illuminated area over the ellipsoid. However, this illuminated area changes with the ellipsoid orientation. While the most rigorous approach would be to model the scatterer center of the illuminated area, this task is difficult and time-consuming, and thus impractical. Therefore, we simply select a point that is always directly illuminated and surrounded by other illuminated points such as the specular point. Notice that we discuss and compare both sampling approaches, that is, sampling around \mathbf{q}_b or \mathbf{p}_b , in Section VI-B.

In addition, I_b and $I_{b,i}$ are 1 only when the whole propagation path is unobstructed, and zero otherwise. In addition, they are properly defined in Section III-B. The strength coefficients w_b are introduced to compensate for any overestimation or underestimation of the power reflected from each body part and satisfy $\sum w_b = 1$.

Notice that $s_b(\cdot)$ is defined in (2) where, to define the complex path gain a_b in (3b), the RCS associated with each diffuse point must be known. Current radar-signal models consider samples with equal-RCS [11], [12]. The ellipsoid RCS is evenly distributed to all samples in [12], whereas [11] manually sets a constant value. Therefore, we follow the same approach proposed in [12]. That is, we consider $\sigma_{b,i}^{(r)} = \sigma_b^{(q)} = (\sigma_b/(N_b + 1))$, where σ_b is the body-part RCS defined in (19), and $\sigma_b^{(q)}$ and $\sigma_{b,i}^{(r)}$ are the RCS associated with \mathbf{q}_b and $\mathbf{r}_{b,i}$, respectively. Since more samples are located around the specular point, our approach concentrates the radar-return power around this point, even when considering samples with equal-RCS.

Lastly, we describe the proposed approach to compute the specular point \mathbf{q}_b and diffuse points $\mathbf{r}_{b,i}$ in Sections III-A and III-B, as well as, the evaluation of obstructed paths.

A. Specular and Diffuse Points

We define the specular and diffuse points \mathbf{q}_b and $\mathbf{r}_{b,i}$ over the ellipsoid \mathcal{E}_b , as the points where the reflected chirp to the radar is only due to specular and diffuse reflection, respectively.

The specular point fulfills two conditions that can be independently used to locate it. The first condition identifies \mathbf{q}_b as the closest point over the ellipsoid \mathcal{E}_b to the radar, located at \mathbf{p}_R (in the global frame) and $\tilde{\mathbf{p}}_{R,b}(t)$ (in the b -body-part frame).⁵ For simplicity, we consider the average radar location

⁵Notice that the radar location is fixed in the global frame. However, it is not fixed in the frame of body part- b , hence the time dependence.

Algorithm 1 Specular and Sample Points

Input: $\tilde{\mathbf{p}}_{R,b}$, ρ_b , N_b , \mathcal{E}_b (for $b = 1, 2, \dots, N_B$), $\nu > 0$, N_K
Initialize: $\tilde{\mathbf{n}}_b = \tilde{\mathbf{m}}_R(\mathbf{0})$ and $\tilde{\mathbf{q}}_b = h_b^{-1}(\tilde{\mathbf{n}}_b)$

A. SPECULAR POINTS:

for all $b \in \mathcal{B}$ in parallel do

for $k = 1, 2, \dots, N_K$ do

1. Update $\tilde{\mathbf{n}}_b = \tilde{\mathbf{n}}_b + \nu[1 - \langle \tilde{\mathbf{n}}_b, \tilde{\mathbf{m}}_R(\tilde{\mathbf{q}}_b) \rangle] \tilde{\mathbf{m}}_R(\tilde{\mathbf{q}}_b)$

2. Normalize $\tilde{\mathbf{n}}_b = \frac{\tilde{\mathbf{n}}_b}{\|\tilde{\mathbf{n}}_b\|}$

3. Update $\tilde{\mathbf{q}}_b = h_b^{-1}(\tilde{\mathbf{n}}_b)$

end for

end for

B. SAMPLE POINTS:

for all $\tilde{\mathbf{q}}_b$ in parallel do

for $i = 1, 2, \dots, N_b$ in parallel do

1. Draw $\varphi_{b,i} \sim \mathcal{N}(0, \rho_b^2, -\frac{\pi}{2}, \frac{\pi}{2})$ and $\tilde{\mathbf{w}}_{b,i} \in \mathcal{T}_b$

2. Axis of rotation: $\tilde{\mathbf{w}}_{b,i} = \tilde{\mathbf{w}}_{b,i} - \tilde{\mathbf{q}}_b$

3. Rotate $\tilde{\mathbf{n}}_{b,i}^{rot} = \tilde{\mathbf{n}}_b \cos \varphi_{b,i} + (\tilde{\mathbf{w}}_{b,i} \times \tilde{\mathbf{n}}_b) \sin \varphi_{b,i}$

4. Compute $\tilde{\mathbf{r}}_{b,i} = h_b^{-1}(\tilde{\mathbf{n}}_{b,i}^{rot})$

end for

end for

$\tilde{\mathbf{p}}_{R,b}$ during a frame of chirps. Thus, the specular point in the frame of body part- b is defined as

$$\tilde{\mathbf{q}}_b = \arg \min_{\tilde{\mathbf{q}} \in \mathcal{E}_b} \|\tilde{\mathbf{p}}_{R,b} - \tilde{\mathbf{q}}\|. \quad (7)$$

This optimization problem is usually solved using the method of Lagrange multipliers [20].

Given that (7) must be solved N_B times for every frame of transmitted chirps, the computational cost can be significant. Therefore, we propose a simpler alternative to compute all N_B -specular points at once. This approach is based on the second condition of the specular point, that is, it is the point over the ellipsoid \mathcal{E}_b whose normal $\tilde{\mathbf{n}}_b = h_b(\tilde{\mathbf{q}}_b)$ points in the radar direction $\tilde{\mathbf{m}}_R(\tilde{\mathbf{q}}_b)$, that is,

$$\langle \tilde{\mathbf{n}}_b, \tilde{\mathbf{m}}_R(\tilde{\mathbf{q}}_b) \rangle = 1 \quad \text{with} \quad \tilde{\mathbf{m}}_R(\tilde{\mathbf{q}}_b) = \frac{\tilde{\mathbf{p}}_{R,b} - \tilde{\mathbf{q}}_b}{\|\tilde{\mathbf{p}}_{R,b} - \tilde{\mathbf{q}}_b\|}. \quad (8)$$

The point $\tilde{\mathbf{q}}_b$ that satisfies (8) can be found iteratively as detailed in Algorithm 1-Step-A. In short, adding a scaled version of $\tilde{\mathbf{m}}_R$ to $\tilde{\mathbf{n}}_b$, updates $\tilde{\mathbf{n}}_b$ closer to the direction of $\tilde{\mathbf{m}}_R$. Later, we update $\tilde{\mathbf{q}}_b$ as the point over the ellipsoid that has the same updated $\tilde{\mathbf{n}}_b$. As this process is repeated, $\tilde{\mathbf{n}}_b$ gradually stops being updated as both $\tilde{\mathbf{n}}_b$ and $\tilde{\mathbf{m}}_R$ become more and more aligned, as it can be seen in Step-A1. Notice that $\tilde{\mathbf{q}}_b$ can be found using noncomplex operations (element-wise sums and products), thus it can be implemented efficiently for all N_B body parts at once.

Regarding the sampling points $\tilde{\mathbf{r}}_{b,i}$, we propose to randomly select them based on the angle $\varphi_{b,i}$ between the normal vectors at points $\tilde{\mathbf{r}}_{b,i}$ and $\tilde{\mathbf{q}}_b$, that is,

$$\varphi_{b,i} \sim \mathcal{N}\left(0, \rho_b^2, -\frac{\pi}{2}, \frac{\pi}{2}\right), \quad \text{with } \angle h_b(\tilde{\mathbf{r}}_{b,i}), h_b(\tilde{\mathbf{q}}_b) = \varphi_{b,i}. \quad (9)$$

That is, $\varphi_{b,i} \in (-\pi/2, \pi/2)$ is drawn from a truncated zero-mean Gaussian distribution with variance ρ_b^2 . Notice that

as ρ_b increases, the diffuse points are more widely spread over the ellipsoid \mathcal{E}_b , hence we denote ρ_b as the sampling spread.

Note that all vectors fulfilling (9) lay over a cone of angle $\varphi_{b,i}$ and axis-of-symmetry $h_b(\tilde{\mathbf{q}}_b)$. Hence, we first draw a vector $\tilde{\omega}_{b,i}$ that lays over the tangential plane \mathcal{T}_b defined at the specular point. This vector $\tilde{\omega}_{b,i}$ is used to rotate the normal $\tilde{\mathbf{n}}_b$ by an angle $\varphi_{b,i}$.⁶ Finally, the rotated vector $\tilde{\mathbf{n}}_{b,i}^{\text{rot}}$ is used to find the diffuse point $\tilde{\mathbf{r}}_{b,i}$ that has the same normal, that is, $\tilde{\mathbf{r}}_{b,i} = h_b^{-1}(\tilde{\mathbf{n}}_{b,i}^{\text{rot}})$. This procedure is summarized in Algorithm 1-Step-B.

B. Occlusion

We consider that a path can only be occluded by another body part. Therefore, to decide if a path trajectory is obstructed or not, the path intersections with each body part must be evaluated. Consequently, we define the round-trip path trajectory \mathcal{P} as a sequence of $N_{\mathcal{P}}$ points \mathbf{v}_j (for $j = 0, 1, \dots, N_{\mathcal{P}}$). Notice that the initial and last point in the path trajectory is the same, for example, the radar position.

In addition, these points are located where the direction of propagation changed. The simplest path trajectory is for the LOS case, which contains only three points: radar, body part, and radar. However, this is different when considering first-order multipath (FOM) or second-order multipath (SOM). For instance, in the FOM case, the path trajectory \mathcal{P}_i contains four points: radar, wall reflective point, body part, and radar. Similarly, in the SOM case, the path trajectory \mathcal{P}_i contains five points: radar, first wall reflective point, body part, second wall reflective point, and radar.

Finally, the indicator function I_b in (6) is defined as

$$I_b = \prod_{j=1}^{N_{\mathcal{P}}} \prod_{a \in \mathcal{B} \setminus b} (1 - I_a(\mathbf{v}_{j-1}, \mathbf{v}_j)) \quad (10)$$

where $\mathcal{B} \setminus b$ denotes the set of all body parts except \mathcal{E}_b , and $I_a(\mathbf{v}_{j-1}, \mathbf{v}_j)$ evaluates if ellipsoid \mathcal{E}_a intersects the segment $\mathbf{v}_{j-1}\mathbf{v}_j$. Moreover, $I_a(\cdot)$ is defined in Appendix A.

Notice that the outer-level product in (10) ensures that the obstruction of all $N_{\mathcal{P}}$ segments is evaluated. In addition, the inner-level product evaluates the obstruction of a single segment by any body part except the one that owns the sample point, that is, it evaluates the interbody-part occlusion. In summary, as soon as any body part intersects one segment in the path \mathcal{P} , the indicator function $I_b = 0$.

C. Multipaths

Multipaths are unavoidable, since even in a large empty room, part of the energy is always reflected on the floor. Therefore, we extend the empirical signal model (5) to consider multipaths as

$$y_{\text{EM}}(t) = \sum_{m \in \mathcal{M}} \gamma_m S(t, \mathcal{Q}_m, \mathcal{R}_m) + n(t) \quad (11)$$

where \mathcal{M} is the set of considered path cases, such as LOS, FOM, and so on. The coefficients γ_m are introduced to

⁶Algorithm 1-Step-B3 makes use of the Rodriguez formula to perform the rotation [21]. Notice that the third term of the Rodriguez formula is omitted since it is zero when the vector and axis of rotation are orthogonal.

compensate for the relative strength between the path cases. Notice that \mathcal{Q}_m and \mathcal{R}_m denote that the specular and diffuse points are different for each path case because the direction of the incident ray on the body-part changes.

Considering multipaths can be an extremely difficult task when there is a high number of reflecting body parts in the scene. Therefore, we restrict our analysis in two ways. First, we consider paths that bounce over a single body part. For instance, we do not consider the FOM path given by: radar, first body part, second body part, and radar. Second, we consider paths that bounce over a single wall. Consequently, the incident ray on each body part is easily found using the image method. Note that these limitations might become relevant for specific human activities where multipaths between multiple body parts dominate.

Leveraging the image method, we require four steps to place the point scatterers for each specific wall. First, we find the image of all ellipsoids using the wall as the plane of symmetry. Second, the specular and diffuse points are found over the image ellipsoids as described in Section III-A. Third, the wall reflective point is found as the intersection of the radar-point segment and the wall. This reflective point is important since it is used to define the path trajectory and thus to evaluate the occlusion as detailed in Section III-B. Lastly, we compute the image of the generated points using again the wall as the plane of symmetry. Note that the final points lay over the real ellipsoids with incidence rays coming from the wall (see Fig. 3).

IV. PARAMETERS SELECTION

The empirical signal model introduced in Section III contains several adjustable parameters. More precisely, for each body-part- b , we list all of them, from (6) and (11), as follows.

- 1) N_b : number of diffuse points.
- 2) ρ_b : sampling spread.
- 3) w_b : body-part strength coefficient.
- 4) γ_m : multipath-case strength (one per element in \mathcal{M}).

In summary, taking $\mathcal{M} = \{\text{LOS}, \text{FOM}, \text{SOM}\}$, we need to select a total of $3(N_{\mathcal{B}} + 1)$ parameters, which is not an easy task.

The procedure introduced here helps to find the optimal hyperparameters for a specific scenario. We define a *scenario* based on various factors, such as radar position/orientation, disposition of walls, human activity type, and so on. Therefore, the hyperparameters found for one particular scenario might translate well or badly to another one, depending on the scenario similarities. Notice that if the objects in the scene are kept fixed, the portability/suitability of the optimized hyperparameters depends almost purely on the human-activity similarities. Consequently, the remainder of this section assumes that the parameters of a scenario are fixed, including the human-activity type [say e.g., walk-toward (WT)].

The number of diffuse points N_b has a direct impact on the computational cost, whereas the rest of the parameters ($\mathcal{Q}_b, w_b, \gamma_m$) do not affect it. For this reason, we propose a two-step approach to independently and properly select these two sets of parameters, as follows.

A. Step 1: Number of Samples

As stated before, increasing the number of samples increases the computational cost. On the one hand, a high number of samples N_b allows us to better represent points over the same body part that move with different radial velocities. On the other hand, there is an N_b^* value that optimally captures the range of radial velocities. That is, increasing N_b beyond N_b^* does not generate a significantly different DTS. Therefore, to minimize the computational cost without affecting the model accuracy, we choose this N_b^* value as the optimal hyperparameter.

We aggregate the body parts into groups based on symmetry (left and right, e.g., both arms, both knees, etc.) and mobility (trunk: head, neck, and upper/middle/low chest). For simplicity, we assume that all body parts in the same group share the same parameters: N_b , ρ_b , and w_b .

To select the appropriate N_b , we perform the following fourfold procedure for each body-part group.

- 1) Momentarily set the rest of the parameters to the nonoptimal but easily interpretable values in (12).
- 2) Using Kinect/MoCap data of a specific human activity, generate $y_E(t)$ using (5) (i.e., consider only LOS), and compute the DTS \mathcal{D} for incremental values of N_b .
- 3) Asses the similarity of the generated DTS \mathcal{D} with respect to the initial DTS \mathcal{D} (for $N_b = 1$) (e.g., monitoring the L2-norm, or multiscale structural similarity (MSSIM) index).
- 4) Select the lowest N_b as N_b^* for which the chosen similarity metric does not change significantly anymore. Notice that we provide a practical example in Section V-B based on the MSSIM index

$$\rho_b \rightarrow \infty; \quad w_b = \frac{1}{N_B}; \quad \sigma_b(t) = 1. \quad (12)$$

Notice that the first equation in (12) ensures to generate diffuse samples from all possible points over the ellipsoid \mathcal{E}_b . In addition, the two remaining expressions disable the effect due to unequal RCS on all body parts. This ensures that all samples contribute equally to the final result.

B. Step 2: Sampling Spread and Strength Coefficients

We propose to select (ρ_b, w_b, γ_m) based on a reference example of a human activity. More precisely, we assume that a few measurement pairs (MoCap/Kinect, radar) are available for the same experiment. The radar measurements are used to compute reference measurement DTS \mathcal{D}' . Similarly, the MoCap/Kinect data is used to recreate the DTS \mathcal{D} for the same experiment, in a synthetic way, using the signal model (11). Therefore, all parameters (ρ_b, w_b, γ_m) are found by solving the following optimization problem:

$$\rho^*, w^*, \gamma^* = \arg \min_{\rho, w, \gamma} f(\rho, w, \gamma) \quad (13)$$

where $f(\rho, w, \gamma) = \sum_k \|\mathcal{D}_k(\rho, w, \gamma) - \mathcal{D}'_k\|$, and k is the experiment index. In addition, vectors (ρ, w, γ) group all (ρ_b, w_b) (for $b = 1, \dots, N_B$), and γ_m for $m \in \mathcal{M}$.

It is important to notice that DTS $\mathcal{D}(\rho, w, \gamma)$ is computed based on multiple frames of chirps. Each of these frames

requires sampling many points to evaluate the obstruction of each path. Hence, the function $f(\rho, w, \gamma)$ is costly to evaluate, and the first/second-order derivatives are extremely difficult to compute.

Since the optimization problem (13) cannot be solved using traditional methods, we propose to make use of the Bayesian optimization (BO) method.

C. Bayesian Optimization

It is a statistical technique used for optimizing complex and expensive-to-evaluate functions, such as in hyperparameter tuning for ML models. It constructs a statistical model of $f(\cdot)$ based on previous evaluations. Using the constructed model, current, and past evaluations of $f(\cdot)$, select a new evaluation point. After repeating this process hundreds of times, the minimum is extracted from the final statistical model. In this article, we directly address the challenges of the use of BO for our specific problem, and we kindly refer the reader to [22] for a proper introduction on BO.

It is well known that BO struggles to perform well when the input dimensionality is high [22]. This is the main challenge when trying to solve (13). Consequently, we take the two following measures to address it.

First, we restructure (13) as a two-level optimization problem, as

$$\rho^* = \arg \min_{\rho} \left[\min_{w, \gamma} f(\rho, w, \gamma) \right]. \quad (14)$$

Here, the optimal values for (w, γ) are computed by solving $w^*, \gamma^* = \arg \min_{w, \gamma} f(\rho^*, w, \gamma)$. The outer level optimizes ρ , which dictates the generation of samples, whereas the inner level focuses on the strength coefficients (w, γ) which do not affect the sample generation process. Consequently, the sample generation (which is the most costly) is done only at the outer level. Thus, the inner level can be iterated quickly keeping the same samples generated in the outer-level iteration.

In addition, the new structure in (14) splits the input dimensionality into two groups, making it more suited for applying BO. Moreover, notice that we apply BO to each level.

As a second measure, we further reduce the dimensionality of ρ and w by assuming that body parts in the same group have the same parameters. In addition, we group all ρ for body part groups for which the value of the sampling spread seems to have little effect. For instance, small body parts (such as neck, elbows, etc.) generate the same DTS regardless of the value of ρ_b . The same is true for body parts with little micromotion (such as the neck, shoulders, and chest). This phenomenon will be better shown in Section V-C.

V. RESULTS

In this section, we first describe the experimental setup used to collect a small dataset of measurement pairs (Kinect, radar). Later, we show the final parameters of the signal model introduced in Section III. Next, we address a qualitative comparison between the synthetically generated DTS \mathcal{D} and the one from real measurements \mathcal{D}' . Lastly, we end this

TABLE I
WR1443—TI RADAR PARAMETERS

Frame rate	16.9 ms	Chirp Bandwidth	793.48 MHz
Carrier frequency	79 GHz	Samples per chirp	128
Chirps in frame	128		

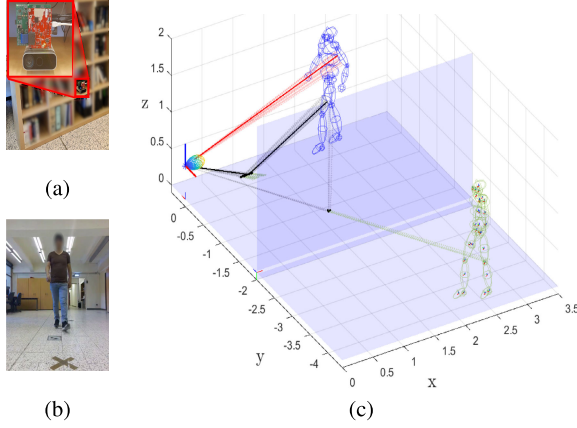


Fig. 3. Experimental setup and recreation in a simulated environment. (a) Kinect and radar setup located at $[0, 0, 45]^T$ cm. (b) Person walking toward the radar captured by the Kinect camera. (c) Simulated scene with only two visible reflective surfaces: floor ($z = 0$ m) and wall ($y = -2$ m). Blue ellipsoids represent the real person's position. Green ellipsoids represent the image person with respect to the vertical wall. The LOS-paths (in red) and SOM-paths (in gray) correspond to the left upper-arm and right femur, respectively. The highlighted/thicker paths are associated with the specular points.

section with a discussion on the effects that each intermediate contribution has over the resulting DTS \mathcal{D} .

A. Experimental Setup and Dataset

The experimental setup consists of a WR1443-radar sensor together with an Azure-Kinect camera co-located at a height of 45 cm, as shown in Fig. 3(a). Notice that both sensors are manually time-synchronized in postprocessing.

The radar is configured with the radar parameters given in Table I, which leads to range and velocity resolutions of 18.8 cm and 11.6 cm/s, respectively. The Azure-Kinect records data at 30 frames/s; each acquired frame is postprocessed to recover the position and axes-orientation of joints. We use this information to compute $\mathbf{G}_{\mathcal{G},b}$ for each body part. The synthetic micro-Doppler information in a radar frame can be computed either from the average frame positions and velocities or from the trajectories (one position per chirp) of the sample points. We use the second approach as it is more compatible with the Kinect data. Nevertheless, the Kinect data rate is lower than the radar-chirp rate. Hence, the Kinect data is filtered and interpolated with cubic splines, so that it can be retrieved at the radar-chirp rate (as commonly done in the literature [6]).

The dataset was collected in an empty room of 3 m height and walls located at $y = 3.5$ m (left), $y = -2$ m (right), and $x = 10$ m (front) with respect to the radar, as shown in Fig. 3(b). The dataset consists of ten people performing four activities with respect to the radar: WT, walk away (WA), lunge forward and recovery (LFR), and hop forward (HF). Notice that in the remainder of this section, we will only

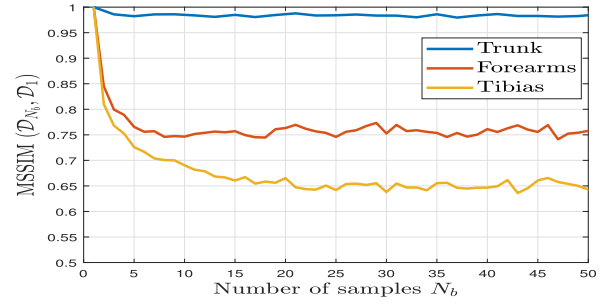


Fig. 4. MSSIM of DTS generated with different number of samples \mathcal{D}_{N_b} , compared to the DTS generated using a single sample \mathcal{D}_1 . To avoid cluttering the figure, we only show the MSSIM curves for three body-part groups: trunk, forearms, and tibias.

TABLE II
NUMBER OF DIFFUSE POINTS N_b PER BODY-PART GROUP

Trunk	1	Forearms	5	Upper-arms	5
Hands	1	Femurs	20	Knees	5
Tibias	20	Heels	5	Feet	10

focus on the walk toward (WT) activity to explain the results regarding the parameter selection (PS) of our empirical signal model.

B. PS—Step 1: Number of Samples

In this section, we show the results of the PS procedure proposed in Section IV. We leverage measurement pairs (Kinect, radar) for one specific activity (we consider WT activity).

For convenience, we present the analysis for three body-part groups: tibias, forearms, and trunk. We use the *multissim* function, provided by MATLAB, to compute the MSSIM index between two DTSs. A MSSIM value close to 1 indicates that the DTSs are similar, whereas a value close to 0 indicates that they are very different. Notice that as stated in Section IV-A, we use the nonoptimal parameters in (12).

The resulting curves are shown in Fig. 4. The three MSSIM curves are converging to 1 for $N_b = 1$ because the DTS generated with one sample is used as the reference. As N_b increases, the MSSIM decreases, since generating a DTS with more samples makes it different from the reference DTS. Each MSSIM curve presents a plateau that indicates the region where the generated DTS does not change for increasing values of N_b . Consequently, we choose the starting N_b values of these regions as selected hyperparameters. That is, we set $N_b = 5$ and $N_b = 20$ for the forearms and tibias, respectively. In addition, we set $N_b = 1$ for the trunk, since its MSSIM remains almost constant for all values of N_b . Notice that, we provide in Table II the final values for each body-part group.

To better analyze the impact of N_b , we show in Fig. 5 the synthetic DTS for $N_b = 5$ and $N_b = 20$ for both tibias [Fig. 5(a) and (b)] and forearms [Fig. 5(c) and (d)]. Fig. 5(a) shows that different points over the tibias move at different radial speeds at the same time instant. For instance, points closer to the feet move faster than points closer to the knee when the foot moves at maximum velocity (see Fig. 5(a) at $t = 2.7$ s). Moreover, Fig. 5(a) and (b) show that $N_b = 5$ diffuse points are not enough to properly capture those velocities.

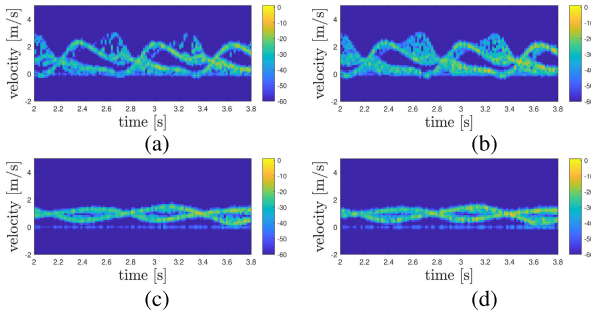


Fig. 5. DTSs for isolated body-part groups. A person walks to the radar located at knee height. We show three body-part groups. Both tibias sampled with (a) $N_b = 5$, (b) $N_b = 20$, (c) both forearms sampled with (c) $N_b = 5$, and (d) $N_b = 20$.

Fig. 5(c) and (d) shows synthetic DTSs for the forearms when $N_b = 5$ and $N_b = 20$. These DTSs are very similar which explains why the forearm-MSSIM curve is flat for $N_b > 5$. Notice that the forearm DTSs are very concentrated. This is because, from the radar perspective, the forearms contain little micromotion. That is, almost all points over the forearm (closer to the hand or elbow) move with almost the same radial speed.

To summarize this section, the more a body part moves in the radar radial direction, the higher the value N_b that is needed. Consequently, the chosen values of N_b might not perform well for a very different radar position, orientation, human activity style, and so on.

C. PS—Step 2: Sampling Spread and Strength Coefficients

On the one hand, when a body part has little micromotion, the sample locations are less relevant since almost all points over the body part move at the same velocity. That can be observed in Fig. 5(d), where for the maximum sampling spread the resulting DTS is highly concentrated in velocity. On the other hand, the most mobile body parts in the WT activity are feet, tibias, and femurs since the remaining body parts have little micromotion. Consequently, we optimize only the sampling spread for the three groups and gather the rest body parts in a common sampling spread.

To further simplify the optimization task, we define two reference values. According to the 3-sigma rule, we define the reference spread as $\rho_{\text{ref}} = (\pi/6)$, since $\varphi_{b,i} \in (-\pi/2, (\pi/2))$. We also define the reference strength coefficient $w_{\text{ref}} = (1/N_B)$, that is, each body part amplitude is mainly due to the RCS. Therefore, we normalize ρ_b and w_b as $\bar{\rho}_b = (\rho_b/\rho_{\text{ref}})$ and $\bar{w}_b = (w_b/w_{\text{ref}})$, respectively.

The two-level optimization in (14) is solved with the BO toolbox of MATLAB. We define the optimization-variable domains as $\bar{\rho}_b \in (0, 4]$, $\bar{w}_b \in (0, 2]$, and $\gamma_m \in (0, 10]$. The outer-level BO iterates for $N_{\text{out}} = 300$ iterations, whereas the inner-level one for $N_{\text{in}} = 200$.

We consider only two DTSs of 1 s duration since the whole optimization procedure is computationally demanding. Moreover, we repeat the process several times with different random seeds. As a result, Fig. 6(a) and (b) shows the mean of final parameters together with the 95% confidence interval.

Regarding the sampling spread result in Fig. 6(a), it is interesting to notice that the mobile parts (feet, tibias, and

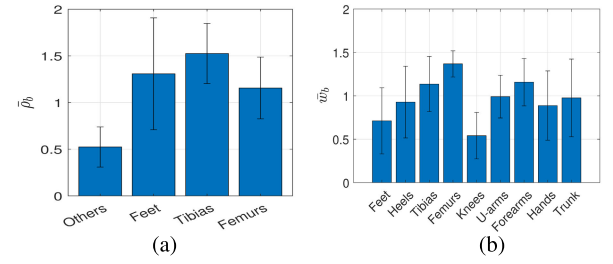


Fig. 6. BO results for (a) normalized sampling spread $\bar{\rho}_b$ and (b) normalized strength coefficients \bar{w}_b .

femurs) require sampling spreads higher than ρ_{ref} , whereas the remaining body parts are well represented with half ρ_{ref} . Notice that these results only hold for the WT activity.

Regarding the strength-coefficient results in Fig. 6(b), the RCS seems to be significantly overestimated only for the feet and knees, whereas it was underestimated for the femurs. The rest of the body parts oscillate around w_{ref} .

VI. QUALITATIVE AND QUANTITATIVE COMPARISON

With the parameters (N_b , ρ_b , w_b , γ_m) selected in Sections V-B and V-C, we can synthetically reproduce the experiments for the setup described in Section V-A, as shown in Fig. 3.

For visualization purposes, Fig. 3 only shows the LOS and SOM paths for the left upper-arm and right femur, respectively. Notice that there are more SOM paths reflected on the floor than on the right wall for the same right femur because some paths are obstructed by the left leg.

To have more realistic results, we implemented the antenna pattern $F(\cdot)$ needed in (3b) from [23, Table 7.3-1]. We also used a radar measurement (with no moving object present in the scene) as noise in the empirical model (11). Thus, realistic static clutter is added to the synthetic signal as part of the noise.⁷

A. Qualitative Comparison

Fig. 7(b) shows the computed DTS \mathcal{D}' based on radar measurements for a person walking toward the radar. Fig. 7(a) and (c) shows the synthetic DTS \mathcal{D} for the SimHumalator case [6] (one sample per body part) and our approach (multiple samples per body part). At first glance, our approach resembles better the measurement. More precisely, we would like to highlight the following aspects.

- 1) SimHumalator shows signatures in places where the measurements do not (e.g., points 1 and 2), whereas our approach agrees with the measurements. We attribute this effect to the location of samples, that is, SimHumalator takes only the body part center, and our approach takes points around the specular point. Notice that we provide a more detailed comparison of these two approaches in Section VI-B.
- 2) Our approach better captures the power distribution, thanks to the strength coefficients, that is, the feet (top

⁷Notice that we remove the average among the chirps before the slow-time Fourier transform, this removes most of the static clutter and leaves only the part due to leakage (horizontal lines around zero velocity in Fig. 7).

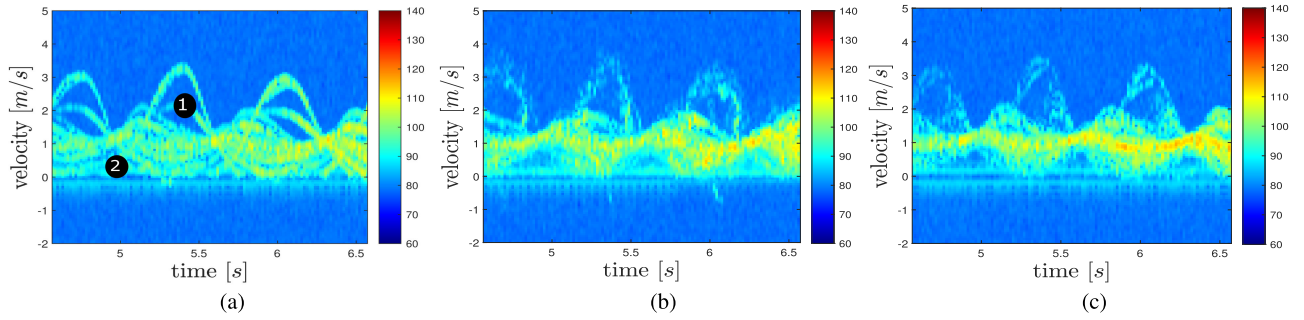


Fig. 7. Comparison of DTSs computed from real and synthetically generated signals. A person walks toward the radar where a co-located Kinect provides the body-part trajectories. (a) SimHumalator: a single-point scatterer per body part. (b) Spectrogram from real radar measurements. (c) Our approach: multiple-point scatterers per body part located around the specular point.

part of the signature) have less power compared to the trunk (central part). Conversely, the power seems to be uniformly distributed among the body parts of the SimHumalator.

- 3) Lastly, both SimHumalator and our approach slightly underestimate the maximum velocity. This is mainly because the Kinect-data was filtered.

B. Qualitative Effect of Intermediate Contributions

In Section VI-A, we directly show the initial and final synthetic DTSs in Fig. 7. Therefore, in this section, we present the intermediate effects that are needed to go from the initial SimHumalator DTS [Fig. 7(a)] to the final DTS [Fig. 7(b)].

The intermediate effects are mainly threefold: multiple samples, reference sampling point, and power distribution. We proceed to incrementally show these effects one after another as follows.

- 1) Fig. 8(a) shows the synthetic DTS extending the SimHumalator approach with multiple samples around the body part center. Notice that we kept fixed all other hyperparameters. It is clear that the addition of multiple samples makes the DTS signature blurry/diffused, which is closer to the measurement DTS [Fig. 7(b)]. Nevertheless, just adding multiple samples does not solve the issue of having DTS signatures in places where the measurements do not [e.g., point 1 in Fig. 7(a)].
- 2) Next, we show in Fig. 8(b) the effect of changing the reference sampling point to the specular point. Comparing Fig. 8(a) and (b), it can be seen that sampling around the specular point changes the shape of the synthetic DTS signature. Moreover, it solves the unwanted issue of having a DTS signature in places where the measurements do not.
- 3) Lastly, Fig. 8(b) is closer in shape to the measurements in Fig. 7(b). However, the power strength of some body parts needs to be adapted. We addressed this issue by using the strength coefficients shown in Fig. 6(b). The resulting and final synthetic DTS is shown in Fig. 7(c). Notice that the feet and trunk signatures have been, respectively, attenuated and strengthened in power.

In summary, all intermediate measures contribute in a different way to achieve the final synthetic DTS, closing the gap between synthetic and real measurement DTSs.

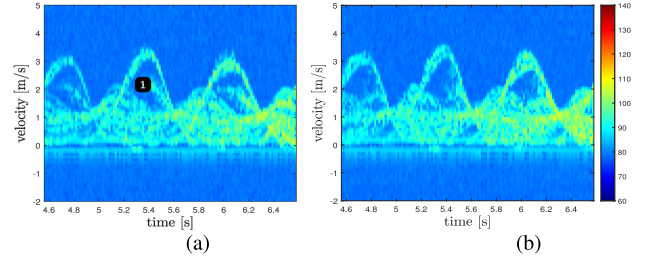


Fig. 8. Effect of considering multiple samples around different reference points. (a) Around body-part center (SimHumalator with multiple samples). (b) Around specular point. All strength coefficients are considered $w_b = (1/N_B)$.

C. Quantitative Comparison

To assess quantitatively the difference between the synthetic and measurement DTSs, we follow the same approach used in [24]. That is, we make use of the speeded-up robust features (SURF) algorithm to extract a robust feature vector of each WT-DTS. These feature vectors are later projected onto a 2-D space using t-distributed stochastic neighbor embedding (t-SNE) technique. Fig. 9 shows the projected feature vectors of all WT samples. It can be observed that our approach overlaps on almost the same domain as the measurement samples, whereas SimHumalator is less overlapped. Moreover, Fig. 9 also shows the centroids of each group. Therefore, we can take the L2 distance between centroids as similarly used in [24]. Consequently, the L2-centroid distances for ours-to-measurement and Simhumalator-to-measurement are 3.7 and 10.84, respectively. Namely, our approach, is in general, closer to the measurements compared to SimHumalator.

VII. APPLICATION USE CASES

We have shown in Section VI that our approach can synthetically generate DTSs that closely resemble the ones computed from real measurements. In this section, we push our approach one step further and show that these synthetically generated data can be used for training an ANN. Thus, we take two representative use cases for classification and regression problems, such as activity classification and pose estimation.

We do not focus on optimizing the ANN-model hyperparameters, but rather on showing how the model generalizes on real measurements when training purely on synthetic data. That is, we focus on assessing indirectly the domain gap between synthetic and real spectrograms. The domain gap can be measured in two different ways. Training on synthetic

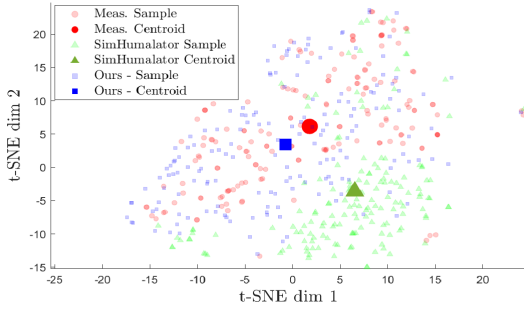


Fig. 9. 2-D projection of SURF features for the WT activity extracted from DTS measurements, simHumalator, and our approach.

TABLE III

DATASET-SPLIT FOR TRAINING, VALIDATION, AND TESTING

	Act. Classification	Vel. Estimation
Training	9 ppl.(90%)	9 ppl.(90%) + Synth. from MoCap
Validation	9 ppl.(10%)	9 ppl.(10%)
Testing	1 pers.	1 pers.

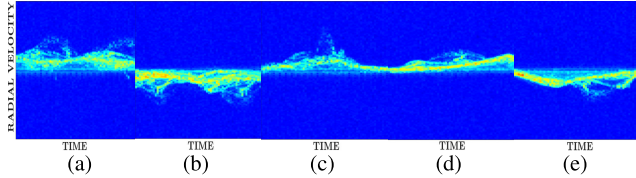


Fig. 10. 256×64 image samples of each class for activity classification (computed from radar measurements): (a) WT, (b) WA, (c) HF, (d) LFR (forward), and (e) LFR (recovery). Notice that (d) and (e) belong to the same class.

data and evaluating on measurement data, or the other way around, that is, training on measurement data and evaluating on synthetic data. In this article, we choose to adopt the first approach.

Consequently, we construct four datasets as follows.

- 1) *Dataset-M*: DTSs from radar measurements.
- 2) *Dataset-S*: synthetic DTSs using SimHumalator [6].
- 3) *Dataset-E*: synthetic DTSs using the empirical model.
- 4) *Dataset-EBC*: synthetic DTSs using the empirical model with sampling around body-part centers instead of specular points.

All four datasets are constructed from the collected (Kinect, radar) data described in Section V-A. Therefore, each example in the measurement dataset (dataset-M) has a corresponding equivalent example in the other three synthetically generated datasets (dataset-S, dataset-E, and dataset-EBC).

Since the collected dataset includes activities from ten people, we split the four datasets into training, validation, and testing sets as shown in Table III. Notice that in the velocity-estimation use case, we add synthetic data from the MoCap dataset [2] only to the training set. Lastly, we apply ($k = 10$)-fold cross-validation with validation and test sets. That is, we perform the training, validation, and testing of the model ten times and isolate the data of a different person as testing set in each realization.

A. Activity Classification

We make use of the collected (Kinect-radar) dataset, described in Section V-A, and thus our data contains four

TABLE IV

PRECISION (P) AND RECALL (R) FOR DIFFERENT MODELS EVALUATED WITH BOTH SYNTHETIC AND MEASUREMENT DATASETS

Class	Validation-set (9 ppl. - 10%)				Testing-set (1 pers.)	
	Synth.		Dataset M		Dataset M	
	P	R	P	R	P	R
Model trained with dataset-S (SimHumalator)						
WT	100	100	99.1	97.3	97.5	94.9
LFR	100	100	99.0	97.0	95.3	89.3
HF	100	100	93.4	99.5	81.4	95.0
WA	100	100	100	99.3	100	99.3
Model trained with dataset-E (our approach)						
WT	100	100	98.9	99.5	97.1	98.1
LFR	99.0	100	97.9	97.6	97.9	93.8
HF	100	98.5	99.8	97.2	93.0	95.6
WA	100	100	100	100	97.9	100
Model trained with dataset-EBC (Sampled around body-part center)						
WT	100	100	98.4	96.2	97.6	95.3
LFR	100	100	99.0	96.7	96.4	87.2
HF	100	99.0	94.3	98.9	85.3	93.7
WA	100	100	99.5	100	99.5	99.6
Model trained with dataset-M (Measurements) - Shown for reference						
WT	-	-	100	100	97.7	99.1
LFR	-	-	100	100	96.3	94.9
HF	-	-	100	100	92.4	96.5
WA	-	-	100	100	99.8	100

classes. Although the number of classes is reduced, this small classification example helps us to show the benefit of using the generated synthetic DTS instead of real measurements. Moreover, a more diverse and complex classification use case will be considered in future works. The amplitude of each DTS experiment is linearly scaled from [70, 140] to [0, 1] dB. In addition, we divide each experiment into images of 256×64 with 50% overlap on the time-axis. We show representative examples of each class in Fig. 10.

1) *Model Architecture and Training*: We make use of a vanilla CNN-architecture similar to the ones used in [25] and [26]. More precisely, the architecture is composed of four convolutional blocks (with output channels: 8, 16, 32, and 64) followed by three linear blocks (with output features: 1024, 512, and 4). The layers at each convolutional block are: conv2d, batchnorm, and relu, with 5×5 kernel size and 2×2 stride. The layers in each linear block are: linear, batchnorm, and relu. Except for the last linear block that outputs logits that are input to the softmax output layer. Lastly, the training process minimizes the cross-entropy loss using stochastic gradient descent (SGD) with a learning rate 0.001.

2) *Classification Results*: We make use of the metrics of *precision* P and *recall* R that are defined as follows:

$$P = \frac{tp}{tp + fp}; \quad R = \frac{tp}{tp + fn} \quad (15)$$

where tp, fp, fn stand for true-positive, false-positive, and false-negative, respectively. Intuitively, precision indicates how often the model correctly predicts the target class, whereas recall measures the model's ability to identify all instances of the target class.

As stated in the introduction of this section, we assess indirectly the domain gap by analyzing the classification results of a model trained on synthetic data and evaluated on measurement data. Consequently, in Table IV, we present the results for three models trained independently with purely

synthetic data: dataset-S, dataset-E, and dataset-EBC. In addition, and for reference only, we also provide the results when training and evaluating purely on measurement data (dataset-M), as it is commonly done when only measurement data is available.

We first show the results when the model is evaluated on *synthetic* validation sets (from dataset-S or dataset-E) that are the same as the training sets. It can be seen that both precision and recall are almost 100% for all cases when using the synthetic examples (two leftmost columns). When the models are evaluated on the validation set of dataset-M, the precision and recall decrease in all cases.⁸ This performance loss is an indirect measure of the difference gap between synthetic and measurement examples. Nevertheless, the gap is only notorious for the precision of HF class.

When evaluating the model on the testing set of dataset-M, the precision and recall fall even further. For instance, the precision falls 19% and 7% for the models trained with dataset-S (SimHumalator) and dataset-E (our approach), respectively. Stated differently, our approach improves the model's ability to generalize on measurement examples of the HF-class by 12% compared to the SimHumalator approach. Notice that it is challenging to correctly classify the HF-class, since it is similar to the LFR (forward) and WT (see Fig. 10).

Table IV presents also the results for a model trained with dataset-EBC. Although sampling around the body-part center makes DTS closer to measurements (Section VI-B), the associated classification results are, in general, just slightly better than those from training with dataset-S (SimHumalator). In addition, when training with dataset-EBC, the HF-class precision falls 9% when evaluating test measurements. Moreover, the final precision value is 8% lower than when using dataset E. The main difference between dataset-E and dataset-EBC is the location of the reference sampling point. Hence, this result indirectly shows the importance of sampling around a specular point rather than around the body-part center.

B. Pose Estimation

This application focuses on estimating the position of important points in the human body known as joints. Out of all model architectures proposed for pose estimation, we make use of the two-step approach introduced in [1], since it makes use of SimHumalator-generated data. Concretely, it is composed of two ANNs referred to as velocity estimation and pose-optimizing networks. The former network estimates the joint velocities from DTSs. Its output is then used together with the pose-optimizing network to estimate the joint positions. Since the DTSs are used only by the first network, we focus our analysis only on the velocity-estimation network.

1) *Model Architecture and Training*: We briefly detail the model architecture and kindly refer the reader to [1] for precise details. Our implementation is composed of: four 1-D-convolutional blocks, one gated recurrent unit (GRU) module, and three linear blocks. All convolutional and linear blocks

⁸From the pair (Kinect, radar) data, the radar data is used to generate the DTS of dataset-M, whereas Kinect data is used to generate the DTS of dataset-S and dataset-E.

TABLE V
AVERAGE DISTANCE ERROR [m/s] FOR SELECTED JOINTS

	Validation-set (9 ppl. 10%)		Testing-set (1 pers.)
Class	Synth.	Dataset-M.	Dataset-M.
Model trained with training-set of dataset-S (SimHumalator)			
Root	0.086	0.197	0.208
Knees	0.122	0.363	0.375
Ankles	0.141	0.564	0.554
Feet	0.155	0.609	0.601
Model trained with training-set of dataset-E (our approach)			
Root	0.077	0.118	0.162
Knees	0.127	0.249	0.274
Ankles	0.167	0.388	0.399
Feet	0.189	0.429	0.443

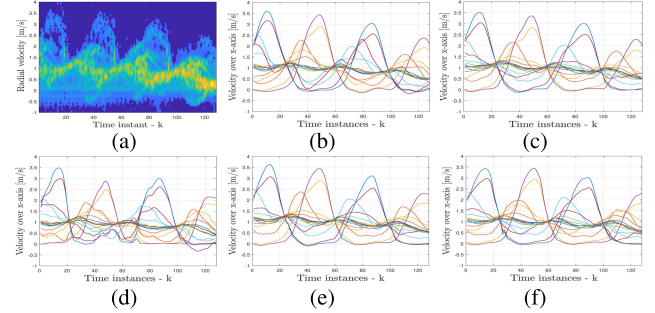


Fig. 11. Estimated x -component velocity of all joints. The person walks toward the radar over the x -axis. (a) DTS from measurement. (b) Target x -component velocity of joints from Kinect sensor. Estimation results when training with SimHumalator data and evaluated with (c) synthetic data and (d) corresponding measurement. Similarly, the results when training with our approach and evaluated with (e) synthetic data and (f) corresponding measurement. Notice that (c)–(f) correspond to a sample in the testing dataset. Each color represents a different joint.

are defined in the same way as in Section VII-A1, except that we implement 1-D-convolutions instead of 2-D-ones. In addition, the GRU has two hidden layers of 512, and the linear blocks have output features: 1024, 512, and 17×3 . Notice that, we track the velocity of 17 joints.

We only consider the WT activity, hence the collected (Kinect, radar) dataset might not be big enough for learning the sequences. Therefore, we leverage the MoCap dataset [2] to synthetically generate DTSs and enlarge the training dataset, as shown in Table III.

2) *Regression Results*: Similar to Section VII-A2, we train two models using purely synthetic data computed from SimHumalator (dataset-S) and our approach (dataset-E). We analyze its performance in two stages: validation and testing. For convenience, we show a detailed summary of results for only four joints in Table V. However, the results of all joints are shown in Fig. 11 for a particular example.

In the validation stage: we first observe that both models are capable of producing an output close to the target 3-D velocities. When evaluated on synthetic examples, the maximum average distance is 0.189 m/s which is of the same order as the radar-velocity resolution. As we use the equivalent validation set of dataset-M, the error increases due to the difference gap between synthetic and measurement examples. Nevertheless, the increment of distance error is lower for the model trained with the synthetic dataset-E (our model).

A similar phenomenon can be observed in the testing stage. Thus, our approach makes it possible to generalize better

dataset-M composed of real measurements. That is not only supported by the smaller increment of distance error but also by the quality of the output data. For instance, we show the input DTS and expected output (obtained from the Kinect sensor) in Fig. 11(a) and (b). When using the synthetic examples, both models yield output of similar qualities, as shown in Fig. 11(c) and (e). However, when using the measurement examples, only the model trained on dataset-E (our approach) maintains the expected quality [see Fig. 11(f)]. In contrast, the model trained with dataset-S (SimHumalator) yields noisy estimates, as shown in Fig. 11(d).

VIII. CONCLUSION AND FUTURE WORK

In this article, we have introduced an empirical radar-signal model to capture complex human movements. The proposed model contains numerous parameters that need to be chosen carefully. Consequently, we proposed a two-step procedure to appropriately select the right parameters for a specific targeted human activity. As a result, we showed that the synthetic DTS, computed from our empirical model, resembles the DTS computed from real measurements. Moreover, our approach yields better results than the current approaches. It is important to highlight that further analysis and experiments are needed in future works to address some limitations of the proposed methods, such as the hyperparameter portability from one scenario to another (changing radar positions and/or orientation, human activity type, etc.)

APPENDIX A ELLIPSOID—PRIMITIVE SHAPE

We define the associated ellipsoid \mathcal{E}_b to each body part- b as

$$\mathcal{E}_b : \left(\frac{\tilde{x}}{\alpha_{\tilde{x},b}} \right)^2 + \left(\frac{\tilde{y}}{\alpha_{\tilde{y},b}} \right)^2 + \left(\frac{\tilde{z}}{\alpha_{\tilde{z},b}} \right)^2 = 1 \quad (16)$$

where $\alpha_{\tilde{x},b}$ and $\alpha_{\tilde{y},b}, \alpha_{\tilde{z},b}$ are the ellipsoid-axis lengths along the \tilde{x} -, \tilde{y} -, and \tilde{z} -axes in the body-part frame of reference.

A. Transformation Matrix

The same 3-D-point can be expressed in the global or b -body part frame of reference, depending on what is more convenient. Therefore, we define the transformation matrix $\mathbf{G}_{g,b}(t)$ that makes the conversion (in homogeneous coordinates) from the b -body-part to the global frame of reference as

$$\mathbf{G}_{g,b}(t) = \begin{bmatrix} \mathbf{R}_{g,b}(t) & \mathbf{T}_{g,b}(t) \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (17)$$

where $\mathbf{0}_{1 \times 3}$ is the 1×3 -matrix of zeros, and $\mathbf{R}_{g,b}$ and $\mathbf{T}_{g,b}$ are the 3×3 -rotation and 3×1 -translation matrices, respectively.

Notice that a point $\tilde{\mathbf{p}}_b$ is always static in the b -body part frame. In fact, the trajectory of that point $\mathbf{p}_b(t)$ in the global frame is given by the matrix multiplication of $\mathbf{G}_{g,b}(t)$ and $\tilde{\mathbf{p}}_b$ in homogeneous coordinates, that is,

$$\begin{bmatrix} \mathbf{p}_b(t) \\ 1 \end{bmatrix} = \mathbf{G}_{g,b}(t) \begin{bmatrix} \tilde{\mathbf{p}}_b \\ 1 \end{bmatrix}. \quad (18)$$

Algorithm 2 Function $\tilde{\mathbf{p}} = h_b^{-1}(\tilde{\mathbf{n}})$

Input: $\tilde{\mathbf{n}}, \mathcal{E}_b, \tilde{\alpha}_b = [\alpha_{\tilde{x},b}, \alpha_{\tilde{y},b}, \alpha_{\tilde{z},b}]^T$
 Step 1: Compute $\tilde{\mu} = \tilde{\mathbf{n}} \circ \tilde{\alpha}_b \circ \tilde{\alpha}_b$
 Step 2: Define $\mathcal{E}_b^{(l)}(\cdot)$ as the left-side of (16).
 Step 3: return $\tilde{\mathbf{p}} = \beta \tilde{\mu}$ with $\beta = \frac{1}{\sqrt{\mathcal{E}_b^{(l)}(\tilde{\mu})}}$

We kindly refer the reader to [21], for a detailed description of the transformation matrix. Lastly, $\mathbf{R}_{g,b}(t)$ and $\mathbf{T}_{g,b}(t)$ are easily obtained from Kinect or MoCap datasets.

B. Radar Cross Section

The RCS can be derived from the primitive shape, that is, from the ellipsoid \mathcal{E}_b . Moreover, it is defined as [14]

$$\sigma_b(t) = \frac{\pi \alpha_{\tilde{x},b}^2 \alpha_{\tilde{y},b}^2 \alpha_{\tilde{z},b}^2}{\left(\alpha_{\tilde{x},b}^2 \beta_{\tilde{x},b} + \alpha_{\tilde{y},b}^2 \beta_{\tilde{y},b} + \alpha_{\tilde{z},b}^2 \cos^2 \tilde{\theta}_b(t) \right)^2} \quad (19)$$

with $\beta_{\tilde{x},b} = \sin^2 \tilde{\theta}_b(t) \cos^2 \tilde{\phi}_b(t)$ and $\beta_{\tilde{y},b} = \sin^2 \tilde{\theta}_b(t) \sin^2 \tilde{\phi}_b(t)$. The angles $\tilde{\theta}_b$ and $\tilde{\phi}_b$ are, respectively, the azimuth and zenith angles of the ray connecting the radar location $\tilde{\mathbf{p}}_R$ and the center of \mathcal{E}_b .

C. From Point to Normal Over the Ellipsoid and Vice Versa

For convenience, we define the function $h(\cdot)$ that computes the normal $\tilde{\mathbf{n}}$ at a given point $\tilde{\mathbf{p}}$ over the ellipsoid \mathcal{E}_b , that is,

$$\tilde{\mathbf{n}} = h_b(\tilde{\mathbf{p}}) = \frac{\nabla \mathcal{E}_b(\tilde{\mathbf{p}})}{\|\nabla \mathcal{E}_b(\tilde{\mathbf{p}})\|} \quad (20)$$

where $\nabla \mathcal{E}_b(\tilde{\mathbf{p}})$ is the gradient of ellipsoid \mathcal{E}_b , defined in (16), evaluated at point $\tilde{\mathbf{p}}$. The inverse function $h_b^{-1}(\cdot)$ is not straightforward and it involves several steps, as shown in Algorithm 2.

D. Segment Intersection With Ellipsoid

A segment is defined by two points $\tilde{\mathbf{p}}_0$ and $\tilde{\mathbf{p}}_1$. Moreover, the segment is over the line \mathcal{L} , defined as

$$\mathcal{L} : \tilde{\mathbf{p}}_0 + \delta \tilde{\mu}, \quad \text{with: } \tilde{\mu} = \frac{\tilde{\mathbf{p}}_1 - \tilde{\mathbf{p}}_0}{\|\tilde{\mathbf{p}}_1 - \tilde{\mathbf{p}}_0\|}. \quad (21)$$

Therefore, the intersection between line \mathcal{L} and the ellipsoid \mathcal{E}_b is found by replacing \mathcal{L} into (16) and solving for $\delta \in \mathbb{R}$.

To determine if the intersection happens over the segment $\tilde{\mathbf{p}}_0 \tilde{\mathbf{p}}_1$, we define function $I(\cdot)$ as follows:

$$I_b(\tilde{\mathbf{p}}_0, \tilde{\mathbf{p}}_1) = \begin{cases} 1 & \text{if } \delta^* < \|\tilde{\mathbf{p}}_1 - \tilde{\mathbf{p}}_0\| \text{ and } \delta^* \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

where $\delta^* = \min(\delta_1, \delta_2)$ and δ_1, δ_2 are the solutions replacing (21) into (16).

REFERENCES

- [1] C. Tang, W. Li, S. Vishwakarma, F. Shi, S. Julier, and K. Chetty, "MDPose: Human skeletal motion reconstruction using WiFi micro-Doppler signatures," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 60, no. 1, pp. 157–167, Feb. 2024.
- [2] C. M. Univ. *CMU Graphics Lab Motion Capture Database*. Accessed: Jun. 11, 2024. [Online]. Available: <http://mocap.cs.cmu.edu>
- [3] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "AMASS: Archive of motion capture as surface shapes," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 5442–5451.
- [4] Y. Zhao, V. Sark, M. Krstic, and E. Grass, "Radar dataset synthesis approach for gesture recognition," in *Proc. 20th Eur. Radar Conf. (EuRAD)*, Sep. 2023, pp. 193–196.
- [5] B. Erol, S. Z. Gurbuz, and M. G. Amin, "GAN-based synthetic radar micro-Doppler augmentations for improved human activity recognition," in *Proc. IEEE Radar Conf. (RadarConf)*, Apr. 2019, pp. 1–5.
- [6] S. Vishwakarma, W. Li, C. Tang, K. Woodbridge, R. Adve, and K. Chetty, "SimHumalator: An open-source end-to-end radar simulator for human activity recognition," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 37, no. 3, pp. 6–22, Mar. 2022.
- [7] S. Sundar Ram and H. Ling, "Simulation of human microDopplers using computer animation data," in *Proc. IEEE Radar Conf.*, May 2008, pp. 1–6.
- [8] Y. Zhao, V. Sark, M. Krstic, and E. Grass, "Low complexity radar gesture recognition using synthetic training data," *Sensors*, vol. 23, no. 1, p. 308, Dec. 2022.
- [9] B. Erol and S. Z. Gurbuz, "A Kinect-based human micro-Doppler simulator," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 30, no. 5, pp. 6–17, May 2015.
- [10] S. Waqar and M. Pätzold, "A simulation-based framework for the design of human activity recognition systems using radar sensors," *IEEE Internet Things J.*, vol. 11, no. 8, pp. 14494–14507, Apr. 2024.
- [11] A. Ninos, J. Hasch, M. E. P. Alvarez, and T. Zwick, "Synthetic radar dataset generator for macro-gesture recognition," *IEEE Access*, vol. 9, pp. 76576–76584, 2021.
- [12] N. Kern, J. Aguilar, T. Grebner, B. Meinecke, and C. Waldschmidt, "Learning on multistatic simulation data for radar-based automotive gesture recognition," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 11, pp. 5039–5050, Nov. 2022.
- [13] K. Bansal, G. Reddy, and D. Bharadia, "SHENRON—Scalable, high fidelity and Efficient radar Simulation," *IEEE Robot. Autom. Lett.*, vol. 9, no. 2, pp. 1644–1651, Feb. 2024.
- [14] M. Liu, F. Gao, Z. Cui, S. Pollin, and Q. Liu, "Sensing with OFDM waveform at mmWave band based on micro-Doppler analysis," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2023, pp. 1398–1403.
- [15] S. Peter and V. V. Reddy, "Extraction and analysis of micro-Doppler signature in FMCW radar," in *Proc. IEEE Radar Conf. (RadarConf21)*, May 2021, pp. 1–6.
- [16] N. Geng and W. Wiesbeck, *Planungsmethoden Für Die Mobilkommunikation* (Planning methods for mobile communication). Berlin, Germany: Springer, 1998.
- [17] Nvidia-Corporation. (2024). *Sionna—Primer on Electromagnetics*. [Online]. Available: https://nvlabs.github.io/sionna/em_primer.html
- [18] L. Senigaglia, G. Ciattaglia, A. De Santis, and E. Gambi, "People walking classification using automotive radar," *Electronics*, vol. 9, no. 4, p. 588, Mar. 2020. [Online]. Available: <https://www.mdpi.com/2079-9292/9/4/588>
- [19] D. Belgiovane and C.-C. Chen, "Micro-Doppler characteristics of pedestrians and bicycles for automotive radar sensors at 77 GHz," in *Proc. 11th Eur. Conf. Antennas Propag. (EUCAP)*, Mar. 2017, pp. 2912–2916.
- [20] S. Chen, S. Xin, Y. He, and G. Wang, "The closest and farthest points to an affine ellipse or ellipsoid," *Tsinghua Sci. Technol.*, vol. 17, no. 4, pp. 481–484, Aug. 2012.
- [21] G. Pons-Moll and B. Rosenhahn, "Model-based pose estimation," in *Visual Analysis of Humans*. London, U.K.: Springer, 2011, pp. 139–170, doi: [10.1007/978-0-85729-997-0_9](https://doi.org/10.1007/978-0-85729-997-0_9).
- [22] P. I. Frazier, "A tutorial on Bayesian optimization," 2018, *arXiv:1807.02811*.
- [23] 3rd Generation Partnership Project, "Study on channel model for frequencies from 0.5 to 100 GHz (release 16)," CEDEX, Sophia Antipolis, France, Tech. Rep. 38.901, Nov. 2020.
- [24] S. Vishwakarma, W. Li, C. Tang, K. Woodbridge, R. Adve, and K. Chetty, "Neural style transfer enhanced training support for human activity recognition," 2021, *arXiv:2107.12821*.
- [25] D. Cakoni, L. Storrer, P. D. Doncker, and F. Horlin, "Group counting using micro-Doppler signatures from a 77 GHz FMCW radar," in *Proc. IEEE Int. Radar Conf. (RADAR)*, Nov. 2023, pp. 1–6.
- [26] L. Storrer et al., "People flow estimation with a Wi-Fi-based passive radar," *people*, vol. 2, pp. 1–6, Apr. 2024.



Evert I. Pocoma Copa was born in 1990 in El Alto, Bolivia. He received the M.Sc. degree in electrical engineering from the Université Libre de Bruxelles (ULB), Brussels, Belgium, in 2019, and the Ph.D. degree from the OPERA Department, ULB, in 2024.

Currently, he is working on biomedical radar applications.

Dr. Pocoma received a scholarship from the Foundation Simón I. Patiño for the M.Sc. degree.



Hasan Can Yildirim (Graduate Student Member, IEEE) was born in Türkiye, in 1991. He received the M.Sc. degree in electrical engineering from Université Libre de Bruxelles (ULB), Brussels, Belgium, in 2018, and VUB, Brussels, and the Ph.D. degree from the OPERA Department, Université Libre de Bruxelles, in 2022, on the subject of "Design of Passive Radars Based on the Wi-Fi 802.11ax Signals."

Currently, he is working on passive radars, WLAN sensing, and BioMed radars.

Dr. Yildirim received a scholarship from F.N.R.S-F.R.I.A for the Ph.D. degree.



Jean-François Determe received the electrical engineering degree (Master en ingénieur civil électricien) from the Université libre de Bruxelles (ULB), Brussels, Belgium, in 2013, and the joint Ph.D. degree in engineering from ULB, and Université Catholique de Louvain, Louvain-la-Neuve, Belgium, in 2018.

From 2013 to 2017, he was an FNRS Research Fellow, funded by the Belgian FRS-FNRS. From 2018 to 2021, he was a Post-Doctoral Researcher with the OPERA-WCG Department,

ULB, where since 2021, he has been an Assistant Professor.



François Horlin (Member, IEEE) received the Ph.D. degree from the Université catholique de Louvain (UCLouvain), Louvain-la-Neuve, Belgium, in 2002. He specialized in the field of signal processing for digital communications.

He joined the Inter-University Micro-Electronics Center (IMEC), Leuven, Belgium, in 2006, as a Senior Scientist. He worked on the design of efficient transceivers that can cope with channel and hardware impairments in the context of 4G cellular systems. In 2007, he became a Professor at the

Université libre de Bruxelles (ULB), Brussels, Belgium. He is supervising a research team working on modern communication, positioning, and radar systems. Localization based on 5G signals and Wi-Fi-based passive radars for crowd monitoring are examples of investigated research topics.