Low Complexity Multi-directional In-Air Ultrasonic Gesture Recognition Using a TCN

Emad A. Ibrahim†, Marc Geilen†, Jos Huisken†, Min Li‡, and José Pineda de Gyvez†
Department of †Electrical Engineering and ‡Industrial Design, Eindhoven University of Technology, Eindhoven, Netherlands
e-mails: e.a.t.ibrahim@tue.nl, m.c.w.geilen@tue.nl, j.a.huisken@tue.nl, min.li@nxp.com, jose.pineda.de.gyvez@nxp.com

***Abstract*—On the trend of ultrasound-based gesture recognition, this study introduces the concept of time-sequence classification of ultrasonic patterns induced by hand movements on a microphone array. We refer to time-sequence ultrasound echoes as continuous frequency patterns being received in real-time at different steering angles. The ultrasound source is a single tone continuously being emitted from the center of the microphone array. In the interim, the array beamforms and locates an ultrasonic activity (induced echoes) after which a processing pipeline is initiated to extract band-limited frequency features. These beamformed features are organized in a 2D matrix of size 11×30 updated every 10ms on which a Temporal Convolutional Network (TCN) outputs continuous classification. Prior to that, the same TCN is trained to classify Doppler shift variability rate. Using this approach, we show that a user can easily achieve 49 gestures at different steering angles by means of sequence detection. To make it simple to users, we define two Doppler shift variability rates; very slow and very fast which the TCN detects 95-99% of the time. Not only a gesture can be performed at different directions but also the length of each performed gesture can be measured. This leverages the diversity of in-air ultrasonic gestures allowing more control capabilities. The process is designed under low-resource settings; that is, given the fact that this real-time process is always-on, the power and memory resources should be optimized. The proposed solution needs $6.2 - 10.2$ MMACs and a memory footprint of 6KB allowing such gesture recognition system to be hosted by energy-constrained edge devices such as smart-speakers.**

***Index Terms*—Gesture Recognition, Temporal Convolutional Networks (TCN), Human System Interaction (HSI), Edge Devices, Doppler shift**

I. Introduction

ULTRASOUND-based gesture recognition as a means for touchless control is currently being investigated in the literature. Recently, smartphone manufacturers such as OnePlus and Xiaomi are replacing the infrared sensor (IR); normally used for proximity detection, with an ultrasound-powered algorithm (https://www.ellipticlabs.com/home/). Leveraging such ultrasound technology allows notch-less smartphones offering the users a full screen display with lower energy consumption.

On a different trend, more studies utilize near-audio ultrasound; e.g. $18 - 24$kHZ, to classify a gesture based on ultrasound echoes created by a hand. We summarize the activity on this topic by showing the details in Table I. The table compares different studies in which near-audio ultrasound is employed to design and classify different gestures using commercial off-the-shelf (COTS) components. We highlight aspects such as the number of classified gestures, the performance of the system and the beamforming compatibility for each study. The fact that a COTS microphone is omnidirectional means that a recorded signal is received from all possible directions. To mitigate this, beamforming is applied to a microphone array to separate acoustic signals spatially. There is a limited number of studies involving the combination of beamforming and gesture recognition. Given a microphone array, beamforming refers to accepting signals from a specified steering angle while rejecting signals coming from other angles. In its simplest form, a microphone array can be modeled as a multi-input-single-output (MISO) system where the received signals at each microphone are processed to listen to a steering angle. In this endeavor, popular beamforming algorithms are deployed. Common approaches to produce such a MISO system are by means of Delay-and-Sum (DaS) beamforming and Minimum Variance Distortionless Response (MVDR). An ultrasonic gesture recognition system can be built in combination with beamforming as in [11] where an MVDR beamformer is applied to an 8-microphone array. This is done to collect features and train a CNN-LSTM classifier to distinguish 5 static gestures. Although such integration is promising, the classifier's accuracy is only 64.5%. On the other hand, some studies such as UltraGesture [4] involves the combination of multi-channel microphones input (without beamforming) to classify ultrasonic gestures based on difference channel impulse response (dCIR). While the accuracy is attractive (91.42% to 98.58%), the used classifier; a convolutional neural network (CNN), is compute intensive with 2.48M parameters. This triggers huge amount of Multiply-Accumulate (MAC) operations and memory accesses each inference hindering such proposals from deployment on energy/memory-constrained devices. Therefore this paper proposes a low-complexity ultrasonic system (Fig. 1) that receives ultrasound echoes from different steering angles and classifies them into gestures.

Fig. 1. System overview of ultrasonic patterns detection at the top of a microphone array.

This research has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737487 (SILENSE). This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation program.

TABLE I
ULTRASOUND-BASED STUDIES ON GESTURE RECOGNITION.

Studies	No. of Gestures	Performance	Tested Noise Conditions	TX Signal	RX Features	Hardware, Beamforming	Classifier
SoundWave [1]	5 (fixed)	$86.7-100\%$	Home, Café	18 kHz − 19 kHz	Doppler shift	(1 MIC, 1 SPKR), NA	NA
AudioGest [2]	6 (fixed)	95.1% (Placed at less than 45^o)	**1)** Loud music. **2)** Walking patterns.	19 kHz	Doppler shift	(1 MIC, 1 SPKR), NA	NA
Dolphin [3]	24 (fixed), using gravity sensor	93%	NA	21 kHz	Doppler shift + Gravity sensor	(1 MIC, 1 SPKR), NA	Linear
UltraGesture [4]	12 (fixed)	$91.42-98.58\%$	Music	Chirp: 18 kHz − 22 kHz	difference Channel Impulse Response (dCIR)	(1−4 MICs, 1−2 SPKRs), NA	CNN (2.48M par.)
[5]	8 (fixed)	$87-100\%$	NA	Tone: 21 kHz and 22 kHz	Doppler shift	(2 MICs, 2 SPKRs), NA	Decision Tree
[6]	4 (fixed)	77.5%	Noisy office	Tone: 21 kHz and 22.8 kHz	SNR	(2 MICs, 2 SPKRs), NA	SVM
[7]	NA	NA	NA	217 kHz	ToF	7 MICs (Zigzag), DaS	NA
[8]	NA	NA	NA	40 kHz	B-mode	7 MICs (Linear), DaS	NA
[9]	NA	NA	NA	40 kHz	ToF	16 MICs (Square), DaS	NA
[10]	8 (fixed)	$0-90\%$	NA	40 kHz	B-mode	64 MICs (Square), DaS	1NN
[11]	5 (fixed)	64.5%	NA	40 kHz	Depth and intensity of RX signal.	8 MICs (Square), MVDR	CNN-LSTM
This work	User defined gestures, (Flexible)	$95-99\%$	Dynamic environment: **1)** Loud music/speech. **2)** Microphone clipping.	19kHz − 23 kHz	Doppler Shift at different angles	7 MICs (Hexagon), DaS	TCN

The proposed solution first defines the building block of the gestures, in this context a real-time Doppler shift variability rate classifier. With such proposal, we aim to move the gesture design stage to post-training giving users on edge the ability to customize their own gestures. This is through the combination of stages of feature extraction, beamforming and classification performed under real-time constraints. Therefore, the contributions of this paper are:

- To integrate a MIMO system that samples ultrasound echoes from hand gestures at different angles and classifies them using a Temporal Convolutional Network (TCN) under low-compute/memory settings.
- To propose a generic method to design ultrasonic gestures through sequence labeling and detection.

II. BACKGROUND INFORMATION

When it comes to ultrasound feature extraction (a pre-classification stage), Doppler shift appears at the top of the list. Soundwave [1], AudioGest [2], Dolphin [3], [5] and [12] are examples of Doppler shift Ultrasound-based systems. Some studies measure features such as the transmission channel impulse response like in [4], or Signal-to-Noise Ratio (SNR) statistics as explained in [6]. Other explored features relate to Time-Of-Flight (TOF) of the ultrasound signal as in [7] [9] or B-mode images [8] [10]. In our study, we leverage continuous Doppler shift readings for sequence-to-sequence classification to distinguish time-dependent gestures. The amount of Doppler shift generated is described in (1) [3], where $f_{r(app)}$ and $f_{r(dep)}$ denote the received echoes as the hand approaches and departs,

$$f_{r(app)} = (\frac{v_s + v_h}{v_s - v_h}).f_{tr} \; and \; f_{r(dep)} = (\frac{v_s - v_h}{v_s + v_h}).f_{tr} \quad (1)$$

where f_{tr} (23 kHz) is the transmitted tone from the center of the array, v_s denotes the speed of sound (331.5 m/s) and v_h is the speed of the moving hand at the top of the array.

We represent this physical phenomenon in a bandwidth of $\sim$ 300 Hz and in a form of continuous spectrograms computed on overlapped frames. These features are collected at a steering angle where a hand gesture is being performed. In the next subsections, DaS and MVDR beamformers are briefly discussed and explained.

A. Delay-and-Sum

To understand DaS we revisit its basic principle. Let $x_i[n]$ be a discrete time ultrasound signal received at the ith microphone from hand reflections $s[n]$. Let a steering angle be represented by θ and β as the azimuth and elevation angles of the ultrasonic reflections. The reflections are received with a physical discrete time delay $\delta_{(\theta,\beta,i)}$ measured to the center of the array such that

$$x_i[n] = s[n - \delta_{(\theta,\beta,i)}]. \quad (2)$$

Then the signal recorded by each microphone $z_i[n]$ is balanced out with a calculated discrete time Δ_i as shown here,

$$z_i[n] = x_i[n + \Delta_i] \quad (3)$$

$$z_i[n] = s[n - \delta_{(\theta,\beta,i)} + \Delta_i] \quad (4)$$

and the beamformed output $y[n]$ is retrieved by summing the outputs of the whole set of P microphones,

$$y[n] = \frac{1}{P}\sum_{i=1}^{P} z_i[n] = \frac{1}{P}\sum_{i=1}^{P} s[n - \delta_{(\theta,\beta,i)} + \Delta_i]. \quad (5)$$

For proper DaS performance, the separation between the sensors must be at $\lambda/2$, where λ is the wavelength corresponding to the maximum frequency in the desired band; e.g. 18kHz-24kHz. Therefore a concentric hexagon of $P = 7$ microphones is built, where 6 microphones divide the azimuth span into steps of 60^o angles. This generates equilateral triangles with 6 microphones on the array circumference. In approaches like [7], such designs are taken as building blocks to produce cascaded hexagons fabricated ultrasonic array.

B. MVDR

Minimum Variance Distortionless Response (MVDR) is a beamformer that minimizes the effect of correlated noise. This beamformer assumes that a received signal $x_i[n]$ includes the desired signal $s[n]$, interference $i[n]$ and noise $n_o[n]$. The

desired signal $s[n]$ is assumed to be uncorrelated with both the noise and the interference while the received signal is assumed to be zero-mean and quasi-stationary. The solution to this beamformer is given by,

$$w = \frac{R_{i+n}^{-1} D}{D^H R_{i+n}^{-1} D} \tag{6}$$

where w is a vector of $1 \times M$ complex coefficients, R_{i+n} is the interference-plus-noise covariance matrix and vector D is the steering angle describing where the signal of interest is located. While an MVDR beamformer is effective in rejecting uncorrelated noise, it becomes inefficient with the existence of correlated noise. This is a known limitation of MVDR and other alike adaptive beamformers [10]. In the context of gesture recognition, this problem might appear with ultrasound reflections caused by e.g. room walls or nearby reflecting objects. Such interfering signals can be described as coherent signals (amplitude-scaled and phase-shifted signals). Therefore, a DaS beamformer will be used instead in this study.

C. Temporal Convolution Network

Some practices in the literature find the speed of the gesture using (1) to apply it as a feature to a classifier as in [13]. While this is possible in noiseless conditions, it can be very challenging with background noise e.g. human motion and loud music that can clip the microphone signal. We design a Temporal Convolution Network (TCN) that is trained on noisy Doppler shift readings to enable detection under noisy conditions. Unlike other sequence labeling classifiers such as the Long-Short-Term Memory (LSTM), TCNs extract features from flexible receptive field over input frames, while leveraging compute parallelism over input frames and intermediate channels. In a nutshell, a TCN is implemented by training a 1-D multi-channel fully-convolutional network (FCN) [14] where the convolutions are causal. In a multi-layer TCN, FCN is performed with zero padding at each layer to maintain the length of subsequent layers.

III. System Design

We start by defining the induced ultrasonic patterns as a function of Doppler shift ($f^{'}$) rate of change $\frac{d}{dt} f^{'}_{(t-T:t),\theta,\beta}$ on past T events at a steering angle (θ, β). Let $(hand)_{t,\theta,\beta}$ be the (hand) pattern at t event and at a steering angle (θ, β) defined as,

$$(hand)_{t,\theta,\beta} = \begin{cases} 1 : Slow, & \text{if } |\frac{d}{dt} f^{'}_{(t-T:t),\theta,\beta}| > 0 \\ 2 : Fast, & \text{if } |\frac{d}{dt} f^{'}_{(t-T:t),\theta,\beta}| \gg 0 \\ 3 : None, & \text{otherwise} \end{cases} \tag{7}$$

where '1' is a label indicating the *Slow-Variability* class and '2' is the second label indicating the *Fast-Variability* class, and finally '3'; an auxiliary class, indicating anything else. The microphone array (having $\lambda/2 = 0.7cm$) records Doppler shift readings of an ultrasound tone f_{tr} in the range $19-23$ kHz that is transmitted by a speaker. In this study we set $f_{tr} = 23$ kHz. Intuitively, the definition in (7) can be read as follows, if the hand is causing a slow rate of change of the Doppler shift, the classifier yields a *Slow-Variability* class. In case of a fast rate of change, the classifier declares a *Fast-Variability* class. To apply such definition, the classifier must preserve the type of the input on past T events, i.e fast/slow variability. Therefore we use the concept of causal convolution and pre-designed receptive field during training. The definition also supports instantaneous shifting between the classes; for instance, to tap slowly for 1 second then to tap fast for another second. For the above-mentioned task, Doppler shift readings are captured in a form of consecutive overlapped frames. Recall in (2) that $s[n]$ is the ultrasonic reflections recorded at $f_s = 96$ kHz and received by each microphone in the array. For each ith microphone, the samples in each frame (f) are represented by a column vector $x_{i,f}$, of length $N = 2880$ (equivalent to 30ms), such that $x_{i,f}$ is a delayed subset of the reflections $s[n]$ ($x_{i,f} \subset s[n]$). A shift of length $L = 960$ (equivalent to 10ms) guarantees 10ms of new data each frame. This shapes a matrix $\mathbf{X}_i$ for each ith microphone in (8) as shown in Fig. 2,

$$\mathbf{X}_i = [x_{i,1} \ \ldots \ x_{i,f} \ \ldots \ x_{i,F}] \tag{8}$$

where $\mathbf{X}_i$ is of size $N \times F$ and F is the number of the overlapped frames. Using (8), ultrasound echoes received by the array are $> 50\%$ overlapped maintaining a smooth transition between the consecutive frames.

Each of the frames is passed to a beamforming stage. Due to the nature of the coherent noise environment (ultrasound echoes), DaS instead of MVDR is used as the beamforming front-end. Therefore, a new sliding window denoted as $\mathbf{X}_{\theta,\beta}$ is formed.

The TCN classification will be processing on a predefined bandwidth. To define such bandwidth, we assume a maximum hand speed of $1m/s$. Assuming the speed of sound in air is $v_s = 331.5\ m/s$, and the transmitted ultrasound tone to be 23 kHz, the needed bandwidth BW is ~ 300 Hz as can be derived from (1). Therefore, we perform dimensionality reduction by mixing the received beamformed overlapped frames $\mathbf{X}_{\theta,\beta}$ with a cosine signal generated at $f_{mix} = f_{tr} - 0.5BW$. The frames are then passed through an anti-aliasing low pass filter (LPF) to finally be down-sampled by $DS = 100$. The 2880 samples in each of the frames in (2) reduces to only $\frac{2880}{DS} = 29$. Spectrograms are then formed by means of 32-point FFT. Recalling that the TCN is operating at ~ 300 Hz of bandwidth, the first 11 points in the generated spectrograms are taken as features. To remove the reflected tone and other unnecessary stationary noises, we preform background subtraction by means of *continuous-squared-frame-subtraction* to generate clean Doppler shift readings as shown in Fig. 2.

A. Building a TCN

The kernels in a TCN are convolved with the current frame sample and the previous ones with the option of dilating the convolution. This causal-dilated convolution [15] is defined as follows. Consider a 1-D sequence of an input channel Ch and

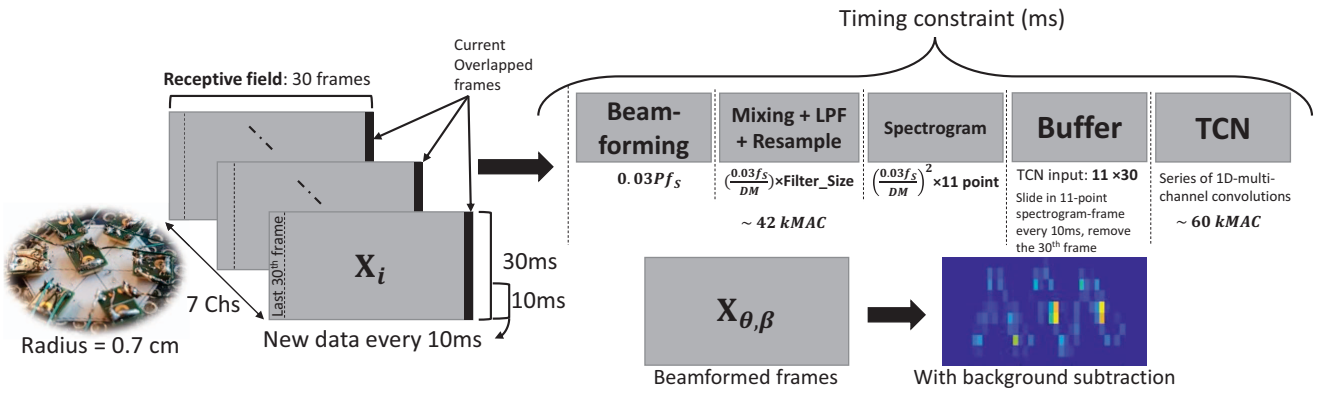

Fig. 2. Quantifying compute complexity from sensors to classification. The number of multiply-accumulate (MAC) operations is approximated for each described process in the data path including the TCN inference.

a 1-D filter *fil*, the dilated convolution operation is described as,

$$Conv_{dilated}(s) = \sum_{m=0}^{k-1} fil(m).Ch(s - d \times m) \qquad (9)$$

where d (the dilation factor) and k (the filter size) are used to control the receptive field. Given s as the current sample in real-time, $Ch(s - d \times m)$ is a dilated sample in the past. In simple terms, dilation is equivalent to including a gap of d steps every two adjacent filter entries. Bearing this definition, a normal convolution corresponds to having $d = 1$. The values of d and k help in defining the receptive field of the TCN, which is an essential step in designing a TCN.

We start by showing a 3-layer TCN in Fig. 3 that is fully connected across the channels in each layer. The dilation factor is changed as the features propagate inside the layers. Using $k = 3$ in all layers and a dilation ratio $d(j) = 2^{j-1}$ [15], where j is the index of each TCN layer, the Jth (final) receptive field RF_J of the final layer is written as,

$$\begin{aligned} RF_J &= RF_0 + (k-1)\sum_{j=1}^{J} d(j) \\ RF_3 &= RF_0 + (k-1)(d(1) + d(2) + d(3)) \\ &= 15 \text{ input frames} \end{aligned} \qquad (10)$$

where RF_0 is the first streamed 30ms frame and is equivalent to 1. The receptive field connections are shown in Fig. 3 and the number of the connections to yield one output feature frame (output receptive field before flattening) is verified to be 15. Practically, this means that each final output frame (before flattening) is derived based on the current frame (30ms) and the previous 14 frames (14×10)ms; i.e., an equivalent of 170ms causal input is covered.

At the first layer, the TCN starts with generating the first low-level 16 1D channels (features) from the 11 input channels. The second layer generates another set of 16-channel features from the first layer. In the third layer and before flattening, a deeper set of 16-channel features is generated. A hidden (classification) layer is triggered when the third layer

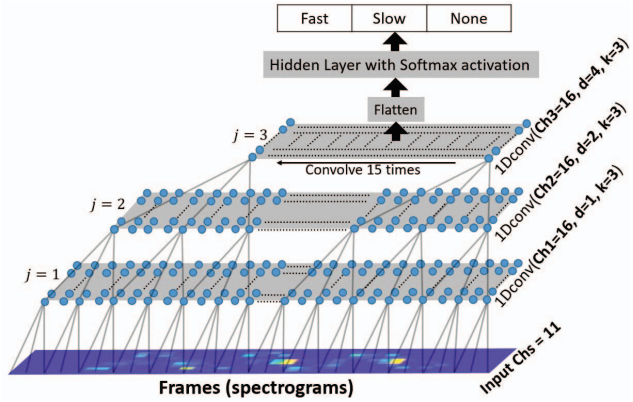

Fig. 3. The employed TCN structure to detect Doppler shift variability over a set of input frames.

is filled with 15 layer-3 frames such that the size of the layer is 16×15. Practically, this means that the TCN outputs a class on a 30-frame ($\sim$ 0.3s) of the beamformed input. A heuristic parametric search with memory/compute constraints has been performed to retrieve the TCN shown in Fig. 3. This is achieved by obtaining Pareto points between many TCNs in terms of multiply-accumulate operations, TCN parameters and their performance on a test-set.

IV. Results and Discussion

For prototyping and validation, 7 analog microphones of type Knowles (SPU0410LR5H−QB), having an operating frequency range 100Hz $\sim$ 80kHz are used in the experiment. Roland OCTA−CAPTURE (https://www.roland.com/); a sound card and an audio interface, is used for data acquisition and for analog-to-digital conversion (sampling at 96KHz and at 24-bit resolution). Playrec (http://www.playrec.co.uk/); a Multi-channel Matlab utility (MEX file) that provides simple yet versatile access, is used as an acquisition utility.

The robustness of data-driven algorithms heavily depends on dataset collection and training. We have justified the motivation of using a TCN and highlighted that a causal system, yielding decisions only on current and past input samples, is of interest. During the stage of data collection the

same data path in Fig. 2 is used to collect series of sliding windows through the shown buffer. The noise was collected by placing the microphone array in different environments, e.g. offices and houses. In these environments, dynamic noises were also introduced such as human activity and extreme noises generated by heavy music that causes microphone clipping. At a defined steering angle, we gather a dataset of variable ultrasonic Doppler shift rate (induced by a hand); such as $\sim$ 1 tap/s and 3 $\leq$ taps/s; as shown in the beamformed features in Fig. 2. This is created as per the definition shown in (7). A user can transition smoothly between these speed values while the classifier plots the transition pattern.

The data path shown in Fig. 2 summarizes the compute complexity at each processing stage in the pipeline. The complexity is represented in multiply-accumulate (MAC) operations. An inference decision has to be made every single stride (10ms) or multiple strides l. Having these design constraints one can determine the needed compute throughput. The MAC operations needed for each stage are estimated without compute-optimization. As shown and stated in the flow, a sum of $\sim$ 102 kMAC operations have to be finished in 10ms, yielding that a maximum of 10.2 million MAC operations per second (MMACs) is required. Another design choice is to trigger the TCN classification every l strides with which a further reduction in MACs is achieved. If the TCN has to make decisions between $10 - 30$ms then the needed MMACs is between,

$$6.2 \leq \frac{MAC_F \times l + MAC_{TCN}}{0.01l} \leq 10.2. \qquad (11)$$

where MAC_F stands for MAC operations for feature extractions (before the TCN) and MAC_{TCN} is the TCN MAC operations. This hardware/software co-design is valuable when it comes to deployment. In our study the learned TCN parameter only amounts to $\sim$ 2500 parameters. If compared to other similar studies such as UltraGesture [4] that requires 2.48M parameters, the saving amounts to $\sim$ 1000X.

Keras [16] a deep learning framework running in Python on a GPU was used to train the TCN. All experiments were performed on a Windows 10 machine with 16GB RAM, an Intel-i5 7300HQ, and a GTX1050 with 4GB VRAM. The intermediate activations were set to 'ReLU' while the final classes where activated using a 'softmax' layer. Adam: A Method for Stochastic Optimization [17] was used for learning with a batch size of 512. The confusion matrix in Fig. 4 shows the converged performance on a separate test set.

Now that the whole system is built, a user can map many gesture patterns at the top of the array. Given that the TCN detects fast/slow Doppler shift variability rate, the user can design different gestures with different speed and length, e.g. a slow tap for a three-second period is a gesture that can be defined. Table II shows seven designed gestures. On our built array, we beamform at 7 directions; 6 on the azimuth span at 60^o steps and one direction perpendicular to the array creating $7 \times 7 = 49$ different gestures.

Fig. 5 shows real-time recordings of the TCN when a user performs variable tapping patterns. The first panel shows the max-class confidence of the classifier each 10ms. The default state of the TCN is *None* which the TCN labels as class '3'. The user starts by tapping fast for $\sim$ 6s after which a slower tap is performed. When a tap transition happens (e.g. from fast tap to slow tap), a drop in the max confidence is observed. This happens because of the input buffer containing mixed patterns (a slow and fast tap features); accordingly, the TCN labels the transition period with *None* but with a lower confidence.

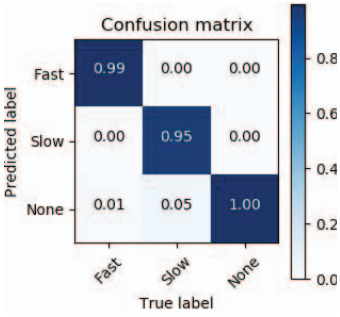

Fig. 4. The normalized confusion matrix computed on a test set.

TABLE II
3-S GESTURES AT (θ,β)

3-s gesture at (θ,β)	3-Fast	3-Slow	Fast-Slow-None	Slow-Fast-None	Slow-Fast-Slow	Slow-None-Fast	Fast-None-Slow
Sequence pattern	2-2-2	1-1-1	2-1-3	1-2-3	1-2-1	1-3-2	2-3-1

A more interesting feature of the designed TCN is that it can log a faster shifting between the classes; that is, a user can alternate between fast/slow taps at a higher pace. However, this introduces some miss-classified instances. To mitigate this issue, we use a low-cost post-processing stage; shown in Fig. 6, that holds the latest credible class. To show the effect of this denoising stage, Fig. 7 shows 3 panels (from top to bottom) **1)** the max-class confidence, **2)** the class instances before denoising, and **3)** the class instances after denoising. As described in Table II, the depicted real-time sequences demonstrate how the speed of the tapping hand changes over time.

As shown in Table I and due to the generic nature of gesture design, there is no unified gesture set/style with which an accurate benchmark can be created. Furthermore, most of the mentioned studies are not explicit about the real complexity of the proposed systems. Therefore, we qualitatively compare aspects such as gesture flexibility, noise-tolerance and performance under constraints. [1] and [2] use Doppler shift based analysis to classify the gestures. As observed this results in a relatively low number of gestures and some settings requirements such as certain device orientation for [2]. Therefore, data-driven classifiers should be considered [3] – [6], [10], [11]. While some referenced studies train a classifier on a fixed set of gestures, our approach is flexible with which the user can design the gestures on a pre-trained network at different steering angles. Our system takes into

consideration heavy external noise impact, which is dealt with during training. During the stage of data collection, we followed the approach of mimicking a real-life scenario. To give an example, a modern smart speaker is responsive almost all the time to a user even during the playback of heavy music or other noises. Accordingly, the idea of integrating such gesture recognition requires considering practical scenarios to possibly allow in-air ultrasonic control capabilities replace physical buttons on-board using a small microphone array.

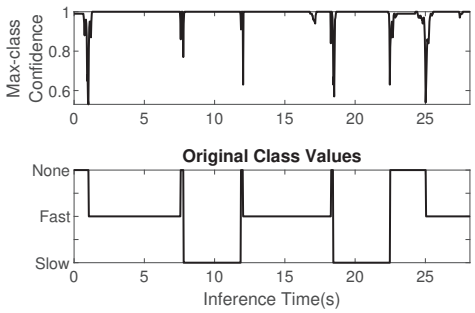

Fig. 5. A real-time plotting of a tapping hand alternating slowly between the two introduced speeds at (θ, β).

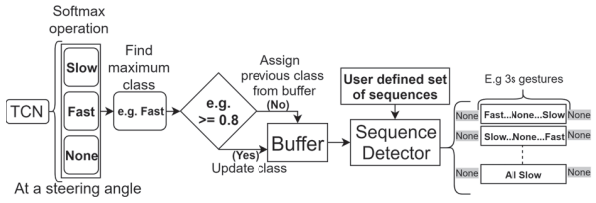

Fig. 6. A denoising scheme used to clean Doppler shift rate of change in case of fast shifting between patterns.

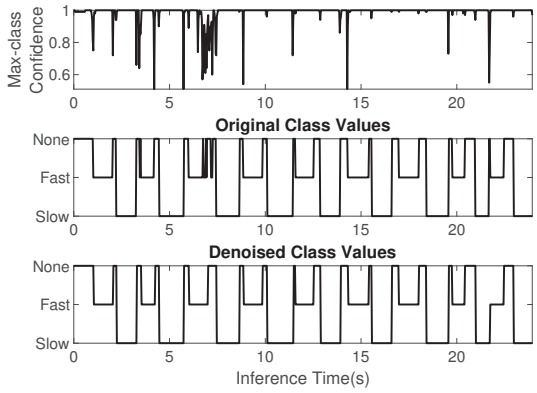

Fig. 7. A real-time plotting of a tapping hand alternating fast between the two introduced speeds at (θ, β) before and after denoising.

V. Conclusion

This work presented a low-complexity multi-steering angle ultrasonic Doppler shift variability classifier. Commercial off-the-shelf (COTS) microphones have been used to build a small form-factor array that receives ultrasound echoes from a hand. These ultrasound echoes are processed to retrieve unique downmixed background-subtracted spectrograms. The beamformer localizes the ultrasonic activity after which a TCN; with sequence detector, determines the type and length of the gesture. This paves the way for various in-air ultrasonic gestures that can be defined by the user after training. This work is an example of how a smart speaker or a conferencing system can be leveraged to create more interactive control functionalities. Future work will be carried out to expand the beamforming to many steering angles to realize more complex gestures.

References

[1] S. Gupta, D. Morris, S. Patel, and D. Tan, "SoundWave: Using the Doppler Effect to Sense Gestures," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '12. New York, NY, USA: ACM, 2012, pp. 1911–1914. [Online]. Available: http://doi.acm.org/10.1145/2207676.2208331

[2] W. Ruan, Q. Z. Sheng, L. Yang, T. Gu, P. Xu, and L. Shangguan, "AudioGest: Enabling Fine-grained Hand Gesture Detection by Decoding Echo Signal," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '16. New York, NY, USA: ACM, 2016, pp. 474–485. [Online]. Available: http://doi.acm.org/10.1145/2971648.2971736

[3] Y. Qifan, T. Hao, Z. Xuebing, L. Yin, and Z. Sanfeng, "Dolphin: Ultrasonic-Based Gesture Recognition on Smartphone Platform," in *2014 IEEE 17th International Conference on Computational Science and Engineering*, 2014, pp. 1461–1468.

[4] K. Ling, H. Dai, Y. Liu, and A. X. Liu, "UltraGesture: Fine-Grained Gesture Sensing and Recognition," in *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2018, pp. 1–9.

[5] B. Van Dam, Y. Murillo, M. Li, and S. Pollin, "In-air ultrasonic 3D-touchscreen with gesture recognition using existing hardware for smart devices," in *IEEE Workshop on Signal Processing Systems, SiPS: Design and Implementation*, 2016, pp. 74–79.

[6] C. Yiallourides and P. P. Parada, "Low Power Ultrasonic Gesture Recognition for Mobile Handsets," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2019, pp. 2697–2701.

[7] R. J. Przybyla, H. Y. Tang, S. E. Shelton, D. A. Horsley, and B. E. Boser, "3D ultrasonic gesture recognition," in *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, vol. 57, 2014, pp. 210–211.

[8] D. M. van Willigen, E. Mostert, and M. A. Pertijs, "In-air ultrasonic gesture sensing with MEMS microphones," *IEEE SENSORS 2014 Proceedings*, pp. 90–93, 2014.

[9] H. Furuhashi, Y. Kuzuya, Y. Uchida, and M. Shimizu, "Three-dimensional imaging sensor system using an ultrasonic array sensor and a camera," *Proceedings of IEEE Sensors*, pp. 713–718, 2010.

[10] H. Kim, *Ultrasound 3D Gesture Recognition (MSc thesis)*. Stellenbosch University, 2018.

[11] A. Das, I. Tashev, and S. Mohammed, "Ultrasound based gesture recognition," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 406–410, 2017.

[12] E. A. Ibrahim, M. Li, and J. Pineda de Gyvez, "PRESS/HOLD/RELEASE Ultrasonic Gestures and Low Complexity Recognition Based on TCN," in *IEEE International Workshop on Signal Processing Systems*, 2019.

[13] H. Ai, Y. Men, L. Han, Z. Li, and M. Liu, "High precision gesture sensing via quantitative characterization of the Doppler effect," in *Proceedings - International Conference on Pattern Recognition*, 2017, pp. 973–978.

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *CoRR*, vol. abs/1411.4, 2014. [Online]. Available: http://arxiv.org/abs/1411.4038

[15] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," *arXiv preprint arXiv:1803.01271*, 2018. [Online]. Available: http://arxiv.org/abs/1803.01271

[16] François Chollet, "Keras," 2015. [Online]. Available: https://github.com/keras-team/keras

[17] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *CoRR*, vol. abs/1412.6, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980