der Bundeswehr
Universität München

Non-Contact Sensing with 77 GHz MIMO FMCW Radar: Vital Signs Monitoring and Hand Gesture Recognition

Keivan Alirezazad

Vollständiger Abdruck der von der Fakultät für Elektrische Energiesysteme und Informationstechnik (EIT) der Universität der Bundeswehr München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

angenommenen Dissertation.

Gutachter:

1. Prof. Dr. techn. Linus Maurer
2. Prof. Dr. techn. Mario Huemer

Die Dissertation wurde am 05.06.2025 bei der Universität der Bundeswehr München eingereicht und durch die Fakultät für Elektrische Energiesysteme und Informationstechnik (EIT) am 29.10.2025 angenommen. Die mündliche Prüfung fand am 05.11.2025 statt.

Declaration of Authenticity

I hereby declare under oath that I have written this dissertation independently and without unauthorized assistance. I have used only the resources listed in the bibliography, and all passages taken verbatim or paraphrased from other works have been clearly marked as such. I have not previously submitted this or a similar work as a dissertation at any other domestic or foreign academic institution, nor have I applied for admission to a doctoral procedure elsewhere.

Keivan Alirezazad
Neubiberg, 03.06.2025

To my wife, Dariana,
for her love, patience, and unwavering support.

To my parents, Mihan and Ourang,
for their unconditional sacrifices and encouragement throughout my life

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Prof. Dr. techn. Linus Maurer, for his invaluable guidance, encouragement, and mentorship throughout the course of this research. His insights, and constructive feedback have shaped not only this dissertation but also my development as a researcher. My sincere thanks go to my second examiner, Prof. Dr. techn. Mario Huemer, for generously dedicating his time to review my manuscript and for providing valuable suggestions.

I also wish to thank my colleagues, friends, and the Institute for Integrated Systems (EIT3) for providing a stimulating and supportive academic environment.

Finally, I am immensely grateful to my family-especially my wife and my parents-for their constant love, encouragement, and sacrifices that sustained me through this journey.

Keivan Alirezazad
Neubiberg, 03.06.2025

Abstract

This thesis examines the application of a 77 GHz Multiple-Input Multiple-Output (MIMO) Frequency Modulated Continuous Wave (FMCW) radar, which features 31 virtual receive elements enabled by Time Division Multiplexing (TDM) access, for non-invasive sensing in two key areas: vital signs assessment and Hand Gesture Recognition (HGR). The first section of this thesis targets the measurement of Breathing Rate (BR) and Heart Rate (HR) using Principal Component Analysis (PCA) applied to Range-Angle Maps (RAMs) to filter out noise from phase data over time. When tested against the EQ02+ LifeMonitor wearable device, this method yields a Mean Absolute Percentage Error (MAPE) of 1.06% for HR and 3.27% for BR, demonstrating its efficacy as a contactless health monitoring tool. The second section introduces a dual-stream neural network that integrates Convolutional Neural Networks (CNNs) and Gated Recurrent Units (GRUs) to analyze Range-Doppler Maps (RDMs) and RAMs for HGR. Evaluated through Leave-One-Out Cross-Validation (LOOCV), the CNN-GRU achieves a mean precision of $93.14\% \pm 5.50\%$, surpassing the performance of its Long Short-Term Memory (LSTM) counterpart, the CNN-LSTM, which has a mean precision of $90.90\% \pm 7.90\%$. Challenges arise with gestures such as Swipe Right (SR), Pinch Open (PO), and Pinch Close (PC), which indicate a reduced sensitivity of the radar to these specific gestures, thereby complicating classification. This is evidenced by a 8.33% misclassification rate from SR to Clockwise (CW) with the GRU. The study underscores the suitability of the developed algorithms for the recognition of vital parameters and gestures and thus forms a solid basis for the development of future radar-based systems for monitoring vital parameters (e.g. in motor vehicles) as well as combined systems for gesture and vital parameter recognition in healthcare.

Kurzfassung

Die in dieser Arbeit entwickelten Algorithmen ermöglichen nicht-invasive radarbasierte Messungen für zwei zentrale Anwendungen: die Erfassung von Vitalzeichen und die Handgestenerkennung (HGR). Dazu wurde ein 77-GHz-MIMO-FMCW-Radar (Multiple-Input Multiple-Output, Frequency Modulated Continuous Wave) mit 31 virtuellen Empfangselementen verwendet, die durch Time Division Multiplexing (TDM) realisiert werden. Der erste Teil der Arbeit fokussiert sich auf die Messung der Atemfrequenz (BR) und Herzfrequenz (HR) unter Verwendung der Hauptkomponentenanalyse (PCA), die auf Entfernungs-Winkel-Karten (RAMs) angewendet wird, um das Rauschen der Phasendaten zu minimieren. Im Vergleich zum tragbaren Messgerät EQ02+ LifeMonitor erzielt diese Methode einen mittleren absoluten prozentualen Fehler (MAPE) von $1{,}06\%$ für HR und $3{,}27\%$ für BR und zeigt damit überzeugend ihr Potenzial zur kontaktlosen Vitalzeichen-Erkennung. Der zweite Teil stellt ein duales neuronales Netzwerk vor, das Convolutional Neural Networks (CNNs) und Gated Recurrent Units (GRUs) kombiniert, um Entfernungs-Doppler-Karten (RDMs) und RAMs für die Handgestenerkennung zu analysieren. Mittels Leave-One-Out-Cross-Validation (LOOCV) erreicht das CNN-GRU-Netz eine mittlere Präzision von $93{,}14\% \pm 5{,}50\%$ und übertrifft damit etablierte Methoden wie das CNN-LSTM-Netz, welches eine mittlere Präzision von $90{,}90\% \pm 7{,}90\%$ erzielt. Herausforderungen treten bei Gesten wie Swipe Right (SR), Pinch Open (PO) und Pinch Close (PC) auf, was auf eine reduzierte Empfindlichkeit des Radars bei diesen speziellen Gesten hindeutet und die Klassifizierung erschwert. Dies zeigt sich in einer Fehlklassifizierungsrate von $8{,}33\%$ von SR zu Clockwise (CW) beim GRU-Netz. Die Studie unterstreicht die Eignung der entwickelten Algorithmen zur Erkennung von Vitalparametern und Gesten und bildet damit eine solide Basis, um zukünftige radarbasierte Systeme zur Überwachung von Vitalparametern (z.B. im KFZ) sowie kombinierte Systeme für Gesten- und Vitalparametererkennung im Gesundheitswesen zu entwickeln.

Contents

List of Acronyms vii

List of Symbols xi

1 Introduction 1
- 1.1 Context and Motivations . . . 1
- 1.2 Objectives of the Project . . . 1
  - 1.2.1 Vital Sign Detection . . . 2
  - 1.2.2 Hand Gesture Classification . . . 2
- 1.3 Thesis Structure . . . 3

2 Literature Review on Vital Sign Detection Using FMCW Radar 5
- 2.1 Introduction . . . 6
- 2.2 Multi-Target Detection and Beamforming Enhancements . . . 7
  - 2.2.1 Beamforming-Based Multi-Target Separation . . . 7
- 2.3 Signal Decomposition Techniques . . . 7
  - 2.3.1 Mathematical Representation of VMD . . . 7
  - 2.3.2 Adaptive Parameter Selection in IAPVMD . . . 9
  - 2.3.3 Wavelet-Based Signal Reconstruction . . . 11
- 2.4 Noise Suppression for Vital Sign Extraction . . . 11
  - 2.4.1 Singular Spectrum Analysis for Noise Reduction . . . 11
- 2.5 Sparse Signal Recovery for Multi-Person Monitoring . . . 12
  - 2.5.1 Sparse Signal Representation for Multi-Person Detection . . . 12
  - 2.5.2 Construction of the Dictionary Matrix . . . 13
  - 2.5.3 Sparse Signal Representation and Optimization . . . 13
  - 2.5.4 Empirical Results and Comparison . . . 14
- 2.6 MIMO Radar Configurations for Improved Accuracy . . . 14
- 2.7 Conclusion . . . 15

3 Vital Sign Detection **17**
3.1 Introduction . . . 18
3.2 Human Target Identification . . . 18
3.2.1 RAM Generation . . . 19
3.2.2 DC-value Suppression . . . 21
3.2.3 Experimental Setup . . . 22
3.3 Vital Sign Extraction . . . 30
3.3.1 FIFO Buffer for RAM Accumulation . . . 31
3.3.2 Peak-Finder Module . . . 31
3.3.3 Process of Index Selection . . . 33
3.3.4 Phase Computation . . . 35
3.3.5 Application of Principle Component Analysis . . . 36
3.3.6 Enhancing PCA Analysis with Spatial Information . . . 39
3.3.7 Extracting Breathing and Heart Signals from Chest-Wall Data 41
3.4 Empirical Investigation of Vital Sign Monitoring . . . 45
3.4.1 Procedure and Key Metrics . . . 48
3.4.2 Analysis of Metrics . . . 49
3.4.3 Data Analysis . . . 53
3.5 Conclusion . . . 55

4 Literature Review on Hand Gesture Recognition **57**
4.1 Introduction . . . 58
4.2 Overview of FMCW HGR Pipeline . . . 58
4.3 Data Representations in FMCW Gesture Recognition . . . 58
4.4 Radar-Deriven Hand Gesture Recognition Techniques . . . 59
4.4.1 Evolution of Feature Extraction Methods . . . 60
4.4.2 Comparative Analysis of Classification Techniques . . . 61
4.4.3 Multi-Hand and Multi-Target Gesture Recognition Challenges 62
4.4.4 Advanced Applications of Gesture Recognition . . . 63
4.5 Challenges and Research Gaps . . . 63
4.5.1 Emerging Trends and Future Directions . . . 64
4.6 Conclusion . . . 64

5 Hand Gesture Recognition **67**
5.1 Introduction . . . 69
5.2 Beat Signal Processing . . . 71
5.2.1 Creating Range-Doppler Maps . . . 71
5.2.2 Creating Range-Angle Maps . . . 72
5.2.3 Range, Velocity, and Angle Bin Spacing . . . 72
5.2.4 Enhancement of Range-Doppler and Range-Angle Maps . . . 73
5.3 Hand Activity Detection Using Energy Comparison . . . 74
5.3.1 Initialization . . . 75
5.3.2 Data Collection and Storage . . . 75
5.3.3 Energy Calculation . . . 75
5.3.4 Exponential Moving Average (EMA) . . . 76
5.3.5 Dynamic Thresholds . . . 76

5.3.6 Hand Activity Detection . . . 77
5.3.7 Incrementing Gesture Counters . . . 77
5.3.8 Multi-Cycle Gesture Confirmation . . . 77
5.3.9 Threshold Parameter Selection . . . 80
5.4 Gesture Design and Acquisition . . . 80
5.4.1 Normalization of Input Data . . . 81
5.5 Proposed Deep Neural Network Model . . . 82
5.5.1 Architectural Rationale . . . 83
5.5.2 CNN-GRU Model Architecture . . . 84
5.5.3 Necessity of Hyperparameter Tuning . . . 86
5.6 Hyperparameter Optimization Strategy . . . 86
5.6.1 Impact of Convolutional Layers on Spatial Feature Extraction 87
5.6.2 Effect of GRU Layers and GRU Units on Validation Accuracy . 88
5.6.3 Impact of Convolutional Layers and GRU Units . . . 89
5.6.4 Effect of Dense Layers and Dense Units on Accuracy . . . 89
5.6.5 Top-performing Hyperparameter Configurations . . . 92
5.7 Validation of Model Generalization through LOOCV . . . 95
5.7.1 Motivation for LOOCV and Comparative Analysis . . . 95
5.7.2 Methodology . . . 95
5.7.3 LOOCV Performance Results . . . 96
5.7.4 Per-Class Performance Analysis . . . 97
5.7.5 Subject-Wise Performance Analysis . . . 100
5.7.6 Training Dynamics and Generalization . . . 100
5.7.7 Future Directions . . . 102
5.7.8 Conclusion . . . 103

6 Conclusion and Outlook **105**
6.1 Conclusion . . . 105
6.2 Future Work . . . 106

Appendices **109**
A MIMO Radar Signal Processing . . . 111
B Radarbook 2 . . . 115
C Deep Learning Foundations . . . 119
D List of publications . . . 127

Bibliography **129**

List of Figures **137**

List of Tables **143**

List of Acronyms

AD Arctangent Demodulation. v, 14
ADC Analog-to-Digital Converter. v, 22
AoA Angle-of-Arrival. v, 61, 111, 113, 117
AR Augmented Reality. v, 69
ASM Angle-Slow-Time Map. v, 27, 59

BPM Beats per Minute. v, 10, 14, 15, 44
BR Breathing Rate. v, 1, 2, 6, 10, 13, 18, 30, 31, 41–45, 48, 49, 53, 55, 105–107

CEEMDAN Complete Ensemble Empirical Mode Decomposition with Adaptive Noise. v, 10, 11
CNN Convolutional Neural Network. v, 2, 3, 58–64, 71, 81, 82, 84, 86, 92, 94–97, 99–107, 119, 122
CTC Connectionist Temporal Classification. v, 61, 62, 64
CWT Continuous Wavelet Transform. v, 7, 11, 14
CZT Chirp-Z Transform. v, 7

DACM Differential and Cross-Multiply. v, 11
DIS Dynamic Interference Suppression. v, 63

ECG Electrocardiogram. v, 2
EEMD Ensemble Empirical Mode Decomposition. v, 10, 40, 63
EMA Exponential Moving Average. v, 21, 74, 76, 77

FC Fully Connected. v, 84, 85, 119, 123
FFT Fast Fourier Transform. v, 6, 11, 12, 14, 19, 24, 26, 27, 71, 72, 113
FIFO First-In-First-Out. v, 30, 31, 40

FMCW Frequency Modulated Continuous Wave. v, 1, 2, 6, 11, 12, 14, 15, 18, 19, 21, 22, 24, 35, 45, 48, 55, 58, 59, 61–64, 71, 103, 105
FoV Field of View. v, 18, 22
FPGA Field Programmable Gate Array. v, 115

GAN Generative Adversarial Network. v, 64
GRU Gated Recurrent Unit. v, 2, 3, 71, 82, 84, 86–89, 92, 94–107, 123, 124

HCI Human-Computer Interaction. v, 1, 3, 58, 59, 69, 104, 105, 108
HGR Hand Gesture Recognition. v, 2, 58, 59, 61–64, 69, 71, 72, 103, 105–107
HMM Hidden Markov Model. v, 59, 61, 64
HR Heart Rate. v, 1, 2, 6, 7, 10, 11, 13–15, 18, 30, 31, 41, 42, 44, 45, 48, 49, 53, 55, 105–107

IAPVMD Improved Adaptive Parameter Variational Mode Decomposition. v, 7, 9, 10, 15
ICA Independent Component Analysis. v, 63
IF Intermediate Frequency. v, 6
IIR Infinite Impulse Response. v, 41, 42
IMF Intrinsic Mode Function. v, 11
IMU Inertial Measurement Unit. v, 64, 107
IQR Interquartile Range. v, 89

JSR Joint Sparse Recovery. v, 12, 13, 15

LOOCV Leave-One-Out Cross-Validation. v, 94–96, 104, 105
LSTM Long Short-Term Memory. v, 61, 62, 64, 71, 94–98, 100–107, 122–124

MAE Mean Absolute Error. v, 10, 14, 15
MAPE Mean Absolute Percentage Error. v, 2, 3, 49, 53, 55, 105, 106
MIMO Multiple-Input Multiple-Output. v, 1, 6, 11, 12, 14, 15, 18, 19, 22, 63, 71, 103, 105, 111, 113, 115, 116
MLP Multilayer Perceptron. v, 71
mm-Wave Millimeter-Wave. v, 6, 14, 58, 59, 62, 69
MRC Maximal Ratio Combining. v, 14
MTI Moving Target Indicator. v, 21
MUSIC Multiple SIgnal Classification. v
MVDR Minimum Variance Distortionless Response. v, 7, 15

PCA Principal Component Analysis. v, 2, 6, 36, 37, 39, 40, 45, 55, 105–107
PCC Pearson correlation coefficient. v, 49, 53
PPG Photoplethysmography. v, 2
PSD Power Spectral Density. v, 42, 44, 45

RAM Range-Angle Map. v, 2, 6, 18, 19, 21, 22, 26, 28, 29, 31, 33–35, 58, 62, 64, 71–75, 80, 82, 84, 92, 95, 96, 98, 99, 101–106
RBK2 Radarbook2. v, 22, 25, 55, 72, 103, 105, 106, 115, 116
RCS Radar Cross Section. v, 11
RDM Range-Doppler Map. v, 2, 58, 60–62, 64, 71–75, 80, 82, 84, 92, 95, 96, 98, 99, 101–106, 119
ReLU Rectified Linear Unit. v, 121
RF Radio Frequency. v, 73
RL Reinforcement Learning. v, 64
RMSE Root-Mean-Square Error. v, 14, 15
RNN Recurrent Neural Network. v, 58, 61, 64, 71, 81, 122
RSM Range-Slow-Time Map. v, 26, 27, 59, 64

SEM Sensor Electronics Module. v, 47
SIS Static Interference Suppression. v, 63
SISO Single-Input Single-Output. v, 12
SNR Signal-to-Noise Ratio. v, 13–15, 26, 27, 106
SSA Singular Spectrum Analysis. v, 11, 12, 15
STFT Short-Time Fourier Transform. v, 58, 64
SVD Singular Value Decomposition. v, 11, 37, 113

TDM Time Division Multiplexing. v, 22, 103, 105, 116

ULA Uniform Linear Array. v, 22, 24

VISION VItal SIgns signals recONstruction. v, 11
VMD Variational Mode Decomposition. v, 7–10
VR Virtual Reality. v, 69
VSDR Vital Signs-based Dictionary Recovery. v, 13, 14

List of Symbols

A_R	Amplitude of the received chirp signal. v, 6
A_T	Amplitude of the transmitted chirp signal. v, 6
B	Chirp bandwidth of the FMCW radar. v
$C_{i,j}$	Cross-correlation between the center bin (p_c, q_c) and the offset bin $(p_c + i, q_c + j)$. v, 35
$E_A^{(i)}$	Total energy of the i-th Range-Angle Map $\mathbb{RA}^{(i)}$, computed as the squared magnitude sum over all bins. v, 75
$E_D^{(i)}$	Total energy of the i-th Range-Doppler Map $\mathbb{RD}^{(i)}$, computed as the squared magnitude sum over all bins. v, 75
$H_{\text{bp}}(f)$	Frequency response of the band-pass filter used to isolate the breathing signal. v, 41
$H_{\text{comb}}(f)$	Overall frequency response of the comb filter formed by multiplying the transfer functions of individual notch filters. v, 43
$H_{hf}(f)$	Frequency response of the band-pass filter used to isolate the heart signal. v, 42
$P_{bb}(f)$	Power spectral density (PSD) of the breathing signal at frequency f. v, 42
$P_{hh}(f)$	Power spectral density of the heart signal $h_{ff}(t)$. v, 44
$P_{i,j}$	Power level in dB at offset (i, j) from the center of the 9×9 RAM window. v, 34
Q_k	Quality factor of the k-th notch filter. v, 42
$S_{\text{IF}}(t)$	Intermediate frequency signal. v, 6
T_f	Fast-time sampling interval. v, 12
T_{End}	Number of consecutive no-gesture frames required to consider the gesture as finished. v, 77

T_{Start}	Minimum number of consecutive time frames required to confirm a gesture. v, 77
T_{up}	Duration of the up-chirp. v
$W(a, b)$	Wavelet coefficient at scale a and translation b. v, 11
$\Delta\theta_{\text{bin}}$	Angular bin spacing; angular separation between adjacent FFT bins in the angle spectrum (in degrees). v, 24
$\Delta\theta_{\text{res}}$	Angular resolution of the antenna array in degrees; minimum angular separation required to distinguish two targets. v, 24
Δr_{bin}	Range bin spacing; distance between adjacent FFT bins in the range profile. v, 24
Δr_{res}	Range resolution; the minimum distance required to distinguish two targets in range. v, 24
Δv_{bin}	Velocity bin spacing, determined by carrier frequency, size of Doppler-FFT, and chirp repetition interval. v, 73
$\Omega(t)$	Custom wavelet derived from a heartbeat template. v, 11
$\bar{\mathbf{\Pi}}$	Averaged magnitude matrix of dimensions $P \times Q$ obtained by taking the mean absolute value of $\mathbf{\Pi}^{(m)}$ over M_{FIFO} chirps. v, xi, 31
$\mathbf{\Psi}$	Range-Angle matrix (map) obtained after a Q-point FFT is applied to each row of $\mathbf{\Lambda}$; final output of the 2D FFT stage. v, 19
$\boldsymbol{\chi}$	Input chirp signal in FMCW radar systems. v, 19
γ	Regularization parameter ensuring sparsity. v, 13
$\hat{\phi}_{p,q}$	Unwrapped phase at range index p and angle index q. v, 36
λ	Wavelength of the radar signal. v, 19
$\mathbb{RA}$	Range-Angle Map obtained by taking the magnitude of the range-angle spectrum $\overline{\mathbf{RA}}$. v, 72
$\mathbb{RD}$	Range-Doppler Map computed by summing the magnitude spectra of all antennas for a single frame. v, 72
$\mathbf{C}$	2D matrix representing the complex cross-correlation between the central range-angle bin and all other bins in the 9×9 neighborhood. v, 34
$\mathbf{E}$	$K \times 1$ vector of mode energies, where $E(k)$ is the energy of the k-th mode. v, 9
$\mathbf{P}$	2D matrix representing signal power levels in a 9×9 window around the central range-angle bin. v, 33
$\mathbf{R}$	Covariance matrix of the received signal vectors. v, 7
$\mathbf{\Gamma}$	Input matrix containing time-domain signals across fast-time (rows) and slow-time (columns). v, xi, 19
$\mathbf{\Lambda}$	Range-profile matrix obtained by applying a P-point FFT to each column of $\mathbf{\Gamma}$. v, xi, 19

$\mathbf{\Omega}^{(m)}$	Sinusoidal component of the imaginary part of $\mathbf{\Psi}^{(m)}$. v, xi, 21
$\mathbf{\Phi}$	Element-wise mean of the imaginary parts of $\mathbf{\Psi}^{(m)}$ across all M chirps (DC component of imaginary part). v, 21
$\mathbf{\Pi}^{(m)}$	Clutter-free complex Range-Angle Map of dimensions $P \times Q$ for chirp m, formed by combining $\mathbf{\Theta}^{(m)}$ and $\mathbf{\Omega}^{(m)}$. v, xi, 22
$\mathbf{\Theta}^{(m)}$	Sinusoidal component of the real part of $\mathbf{\Psi}^{(m)}$. v, xi, 21
$\mathbf{\Upsilon}$	Element-wise mean of the real parts of $\mathbf{\Psi}^{(m)}$ across all M chirps (DC component of real part). v, 21
$\hat{\mathbf{\Phi}}$	Matrix of unwrapped phase values. v, 36
$\mathbf{a}(\theta)$	Steering vector at angle θ. v, 7
$\mathcal{F}$	Set of basis frequency components used to represent human vital signs, spanning the physiological bands $[0.1, 0.4]$ Hz (breathing) and $[0.9, 2.5]$ Hz (heart rate). v, 13
$\mathcal{H}$	Set of coordinates corresponding to the detected peaks in the $\bar{\mathbf{\Pi}}$. v, 31
$\mathcal{S}_1$	Set of indices within the 9×9 window whose power is within 5 dB of the central bin. v, xi, 35
$\mathcal{S}_2$	Set of indices within the 9×9 window whose cross-correlation exceeds 70% of the central autocorrelation. v, xi, 35
$\mathcal{S}_{\text{final}}$	Intersection of $\mathcal{S}_1$ and $\mathcal{S}_2$, containing the final set of valid indices for vital sign analysis. v, 35
$\phi(t)$	Phase variation of intermediate frequency signal. v, 6
$\psi_{K,\alpha}$	Energy loss rate criterion in IAPVMD based on number of modes K and penalty coefficient α. v, 10
ρ	Pearson correlation coefficient measuring the linear relationship between two variables. v, 49
$\text{EMA}_A^{(i)}$	Exponential Moving Average of the energy values from Range-Angle Maps up to index i. v, 76
$\text{EMA}_D^{(i)}$	Exponential Moving Average of the energy values from Range-Doppler Maps up to index i. v, 76
$b_f(t)$	Time-domain breathing signal obtained after band-pass filtering. v, 41
d_r	Antenna spacing between receive (RX) elements, expressed in units of wavelength λ. v, 116
d_t	Antenna spacing between transmit (TX) elements, expressed in units of wavelength λ. v, 116
f_{rr}	Slow-time frame repetition rate. v
f_b	Beat frequency. v, 6
f_c	Central operating frequency. v
f_s	Fast-time sampling frequency. v

g_L	Gesture length counter, incremented when a gesture is detected in a time frame. v, 77
g_N	No-gesture counter, incremented when no hand activity is detected. v, 77
$h_f(t)$	Time-domain heart signal obtained after band-pass filtering. v, 42
h_i	Channel coefficient for the i-th channel. v, 14
$h_{ff}(t)$	Heart signal $h_f(t)$ after comb filtering to suppress respiratory harmonics. v, 44
$r(t)$	Residual error remaining after decomposing the input signal into distinct modes. v, 7
$tr_A^{(i)}$	Dynamic threshold for the i-th Range-Angle Map, computed as the product of the smoothing factor γ and the exponential moving average $\text{EMA}_A^{(i)}$. v, 76
$tr_D^{(i)}$	Dynamic threshold for the i-th Range-Doppler Map, computed as the product of the smoothing factor γ and the exponential moving average $\text{EMA}_D^{(i)}$. v, 76
$u_k(t)$	k-th mode obtained from Variational Mode Decomposition. v, 7
$w[n, l]$	Additive noise at fast-time index n and slow-time index l. v, 12
z_{score}	Z-score indicating the statistical significance of the difference between two Fisher Z-transformed correlation coefficients. v, 53

CHAPTER 1

Introduction

1.1 Context and Motivations

Radar technology has evolved from its origins in defense to transformative applications in healthcare and Human-Computer Interaction (HCI) [1]. Among these, 77 GHz Frequency Modulated Continuous Wave (FMCW) radar excels due to its high sensitivity and robustness, enabling non-contact physiological monitoring and gesture recognition [2]. With aging populations and the growing demand for touchless interfaces post-COVID, these capabilities address the pressing needs for remote healthcare and intuitive device control. Unlike prior systems that relied on multiple sensors or lacked real-time performance [3], this thesis leverages a single 77 GHz Multiple-Input Multiple-Output (MIMO) FMCW radar to address two critical applications, as illustrated in Fig. 1.1:

1. **Vital sign detection**: Monitoring Breathing Rate (BR) and Heart Rate (HR) by capturing chest micro-movements.

2. **Hand gesture classification**: Recognizing predefined gestures like swipe or push for touchless control in smart homes, automotive interfaces, and assistive healthcare systems [4].

1.2 Objectives of the Project

This thesis focuses on **two distinct but complementary tasks**, addressing the limitations of intrusive monitoring devices [5, 6] and lighting-dependent gesture systems [7, 8]. It utilizes phase-based signal processing to enhance vital sign accuracy and

machine learning for gesture recognition, enabling real-time non-contact applications. Conducted in a controlled lab setting with healthy adults aged 20–50 under minimal motion conditions, the study lays the groundwork for scalable, real-world applications.

1.2.1 Vital Sign Detection

The first objective is to develop a real-time, non-contact system to estimate BR (0.1–0.4 Hz) and HR (0.9–2.5 Hz) using a 77 GHz FMCW radar, targeting less than 3% Mean Absolute Percentage Error (MAPE) compared to chest strap sensors (e.g., EQ02+ LifeMonitor or Polar H10). Traditional contact-based methods, such as Electrocardiogram (ECG), Photoplethysmography (PPG) or chest straps, cause discomfort, skin irritation and are impractical for continuous, long-term monitoring[5, 6]. These conventional methods also face challenges such as limited mobility and reduced patient compliance, especially in extended monitoring periods, highlighting the need for contactless alternatives. In contrast, radar-based sensing offers a seamless alternative by analyzing phase variations in reflected signals from chest micro-movements.

Key challenges include:

- Suppressing environmental noise and interference such as reflections from surrounding objects or electronic sources, which can obscure vital sign signals.
- Refining Principal Component Analysis (PCA) parameter selection to maximize feature extraction from slow-time phase data.
- Accurately isolating BR and HR after PCA, given their differing frequencies and the potential for cross-talk in the extracted phase features.
- Ensuring accuracy across diverse subjects and conditions.

The resulting algorithm aims to enable applications in telehealth, sleep monitoring, and remote patient care, matching chest strap sensors with high precision while improving user comfort.

1.2.2 Hand Gesture Classification

The second objective is to design a machine learning-based system for real-time Hand Gesture Recognition (HGR) using the same 77 GHz FMCW radar, achieving more than 92% classification accuracy for ten distinct gestures. Gesture-based interfaces are increasingly vital for touchless control in consumer electronics, automotive systems, and virtual reality [9]. Unlike camera-based systems, which falter in low light and raise privacy concerns, radar detects gestures through clothing or obstructions, ensuring robustness [10, 11].

The approach involves:

* Generating Range-Doppler Maps (RDMs) and Range-Angle Maps (RAMs) from radar signals.
* Extracting features like Doppler velocity and angle of arrival for gesture differentiation.

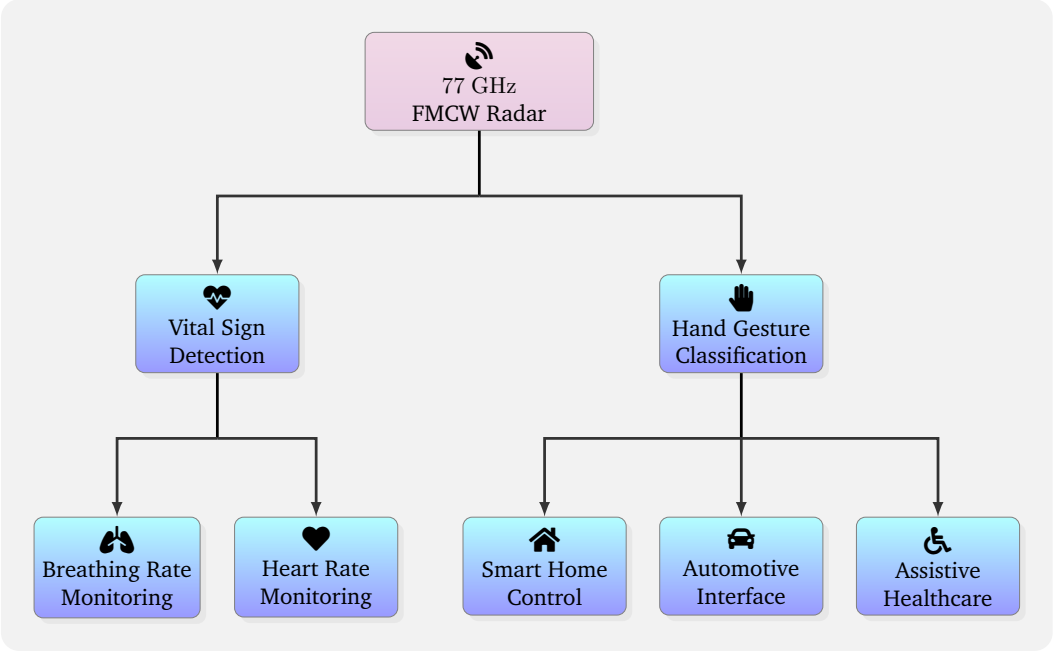


Fig. 1.1: Applications of a 77 GHz FMCW radar for vital sign detection and hand gesture classification.

* Training a Convolutional Neural Network (CNN) with Gated Recurrent Unit (GRU) layers (CNN-GRU) to classify gestures efficiently.

The challenge lies in balancing high accuracy with computational efficiency to support real-time applications, advancing touchless HCI.

1.3 Thesis Structure

This thesis is organized into five chapters, targeting radar engineers, signal processing experts, and HCI researchers:

Chapter 2 reviews literature on radar-based vital sign detection, synthesizing prior methods and identifying gaps in real-time accuracy.

Chapter 3 presents the proposed phase-based signal processing method for vital sign detection, validated with less than 3% MAPE against chest strap sensors, and discusses experimental results.

Chapter 4 reviews literature on radar-based gesture classification, highlighting limitations in computational efficiency and robustness.

Chapter 5 details the CNN-GRU model for classifying ten hand gestures with more than 92% accuracy, including data collection, feature extraction, and performance analysis.

Chapter 6 summarizes findings, discusses limitations, and proposes future research directions for real-world radar sensing.

CHAPTER 2

Literature Review on Vital Sign Detection Using FMCW Radar

Contents

2.1 Introduction 6
2.2 Multi-Target Detection and Beamforming Enhancements 7
2.2.1 Beamforming-Based Multi-Target Separation 7
2.3 Signal Decomposition Techniques 7
2.3.1 Mathematical Representation of VMD 7
2.3.2 Adaptive Parameter Selection in IAPVMD . 9
2.3.3 Wavelet-Based Signal Reconstruction 11
2.4 Noise Suppression for Vital Sign Extraction 11
2.4.1 Singular Spectrum Analysis for Noise Reduction 11
2.5 Sparse Signal Recovery for Multi-Person Monitoring . . 12
2.5.1 Sparse Signal Representation for Multi-Person Detection 12
2.5.2 Construction of the Dictionary Matrix . . . 13
2.5.3 Sparse Signal Representation and Optimization 13
2.5.4 Empirical Results and Comparison 14
2.6 MIMO Radar Configurations for Improved Accuracy . . . 14
2.7 Conclusion 15

2.1 Introduction

Millimeter-Wave (mm-Wave) FMCW radar functions by transmitting frequency-modulated chirp signals, where the frequency increases linearly over time, and these signals reflect off a target—such as a subject's chest in vital sign applications, or any moving object in broader contexts like automotive or surveillance systems. The radar measures the range and velocity of the target by analyzing the frequency difference between the transmitted and received signals, known as beat frequency, which is proportional to the target's distance, and the Doppler shift, which indicates the target's velocity. In the context of vital sign monitoring, the radar captures micro-movements of the chest caused by respiration and heartbeat, which manifest as subtle phase variations in the reflected signals. After mixing the transmitted and received signals, the resulting Intermediate Frequency (IF) signal $S_{\text{IF}}(t)$ encapsulates these phase variations and can be expressed as [3, 12]:

$$S_{\text{IF}}(t) = A_T A_R e^{j(2\pi f_b t + \phi(t))} \tag{2.1}$$

where:

- A_T and A_R represent the amplitudes of the transmitted and received signals, respectively.
- f_b is the beat frequency, which encodes the distance to the subject.
- $\phi(t)$ denotes the phase variation induced by chest motion, where periodic changes correspond to the frequencies of respiration (0.1–0.4 Hz) and heartbeat (0.9–2.5 Hz) .

The phase term $\phi(t)$ is particularly critical for vital sign detection, as its time-varying component captures the cyclic chest movements caused by rhythmic contraction and expansion of the cardiac muscles and breathing activity. By processing $\phi(t)$ through techniques such as Fast Fourier Transform (FFT) or signal decomposition, the distinct frequency components of BR and HR can be extracted, enabling non-contact monitoring [13, 14, 15]. Although FMCW radar's ability to measure range and Doppler makes it widely applicable for detecting any moving object [16], its sensitivity to micro-movements has made it increasingly valuable for biomedical applications like vital sign monitoring [2].

Recent research efforts have concentrated on improving detection accuracy [17, 18, 19], mitigating noise level [20], addressing multi-person scenarios [21], and refining signal decomposition techniques for better vital sign extraction [22]. This chapter reviews the latest developments in beamforming, signal decomposition, sparsity-based recovery, and MIMO processing, evaluating their strengths and limitations [21, 22, 23, 24, 25, 26]. Fig. 2.1 illustrates a mind map of these research areas. They provide a foundation for the proposed methodology presented in chapter 3, which leverages a 77 GHz MIMO FMCW radar with PCA on RAMs to enhance vital sign detection accuracy.

2.2 Multi-Target Detection and Beamforming Enhancements

One of the primary challenges in FMCW radar-based vital sign detection is simultaneously monitoring multiple individuals, particularly when they are positioned at similar distances from the radar. Conventional methods often struggle with signal separation due to overlapping reflections, resulting in interference that degrades detection accuracy [27].

2.2.1 Beamforming-Based Multi-Target Separation

To address this issue, Zhao et al. [23] proposed a Minimum Variance Distortionless Response (MVDR) beamforming approach to enhance signals from specific directions while suppressing interference from other angles. The power of the detected signal at a given angle θ, denoted as P_θ is calculated as:

$$P(\theta) = \frac{1}{\mathbf{a}^H(\theta)\mathbf{R}^{-1}\mathbf{a}(\theta)} \tag{2.2}$$

where $\mathbf{a}(\theta)$ is the steering vector for the desired angle, $\mathbf{a}^H(\theta)$ is its Hermitian transpose (the conjugate transpose, accounting for the complex phase and amplitude response of the antenna array) and $\mathbf{R}$ is the covariance matrix of the received signals.

By combining MVDR beamforming with Variational Mode Decomposition (VMD) and Chirp-Z Transform (CZT), Zhao et al. achieved significant improvements in isolating respiration and heartbeat signals from multiple subjects. Their experiments demonstrated a 97.56% HR accuracy, effectively separating targets at angles like 18° and 22° in controlled scenarios [23]. However, this approach has limited scalability for increasing numbers of targets, and the computationally intensive pipeline hinders real-time application in dynamic settings.

2.3 Signal Decomposition Techniques

A critical step in vital sign detection is separating respiration and heartbeat signals, which often overlap in frequency due to harmonics (e.g., the fourth harmonic of a 0.3 Hz respiration rate overlaps with a 1.2 Hz heart rate). This section explores advanced decomposition techniques to address this challenge, starting with the standard VMD and its enhanced version, the Improved Adaptive Parameter Variational Mode Decomposition (IAPVMD), by Qu et al. [22], followed by the Continuous Wavelet Transform (CWT) method by Vo Dai et al. [28].

2.3.1 Mathematical Representation of VMD

Standard VMD decomposes a signal $x(t)$ into K distinct modes $u_k(t)$, each representing a sub-signal in a specific frequency band [29]:

$$x(t) = \sum_{k=1}^{K} u_k(t) + r(t) \tag{2.3}$$

where $r(t)$ is the residual error. This can be expressed in vector-matrix form as:

$$\mathbf{x} = \mathbf{U}\mathbf{1}_K + \mathbf{r} \tag{2.4}$$

where:

- $\mathbf{x}$ is the $N \times 1$ input signal vector:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \tag{2.5}$$

- $\mathbf{U}$ is the $N \times K$ matrix of decomposed modes:

$$\mathbf{U} = \begin{bmatrix} u_{1,1} & u_{2,1} & \dots & u_{K,1} \\ u_{1,2} & u_{2,2} & \dots & u_{K,2} \\ \vdots & \vdots & \ddots & \vdots \\ u_{1,N} & u_{2,N} & \dots & u_{K,N} \end{bmatrix} \tag{2.6}$$

- $\mathbf{1}_K$ is a $K \times 1$ column vector of ones:

$$\mathbf{1}_K = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \tag{2.7}$$

- $\mathbf{r}$ is the residual noise vector.

The performance of VMD heavily depends on two key parameters:

1. The **mode number** K, which determines how many frequency components are extracted.
2. The **penalty coefficient** α, which controls mode smoothness and prevents excessive frequency overlap.

Limitations of Standard VMD

Standard VMD has several limitations [23, 24, 30]:

- The parameters K and α must be set manually, making it challenging to adapt to different types of signals.
- If K is **too low**, critical frequency components may be missed, leading to incomplete decomposition.
- If K is **too high**, it can result in over-decomposition, where a single frequency component is split across multiple modes or spurious modes capture noise, reducing the accuracy of the decomposition.

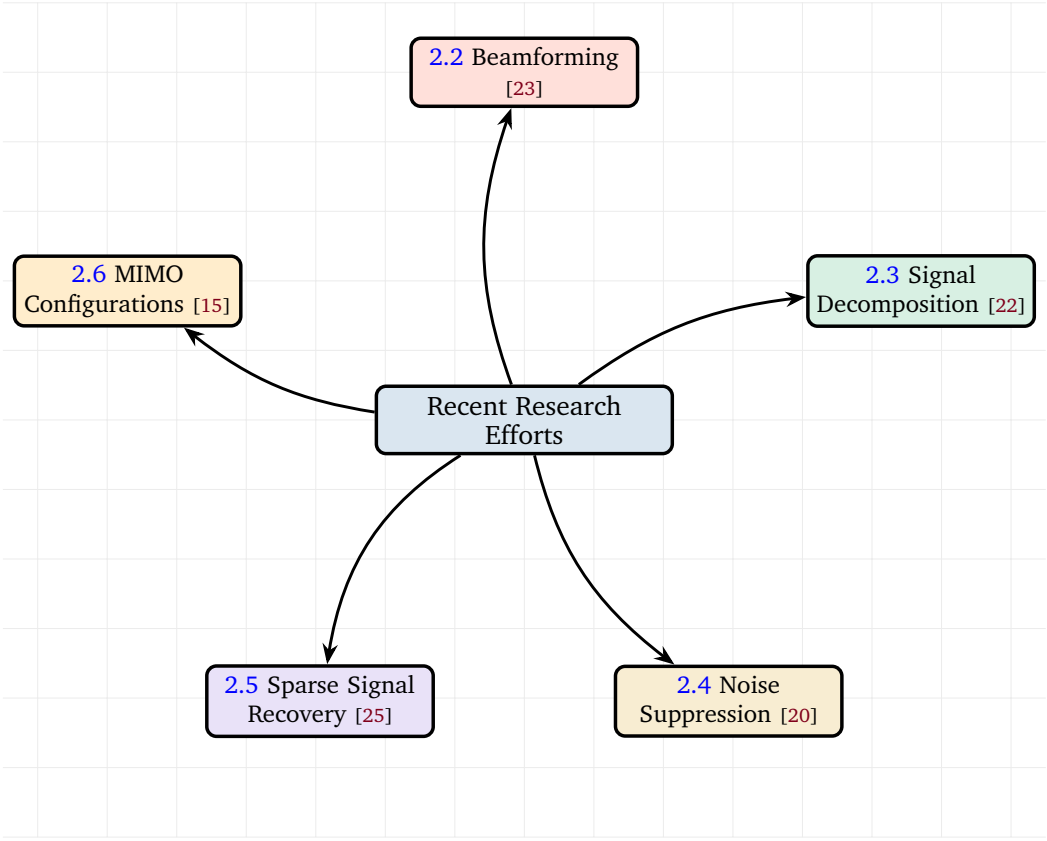


Fig. 2.1: Mind map outlining the structure and components of the recent research efforts in FMCW radar-based vital signs monitoring.

- Standard VMD lacks **iterative tuning** of α, making it suboptimal for real-world signals.

2.3.2 Adaptive Parameter Selection in IAPVMD

Unlike standard VMD, IAPVMD dynamically adjusts K and α based on an **energy loss rate criterion** to optimize decomposition [22, 30]:

$$\psi_{K,\alpha} = \frac{|\sum_{k=1}^{K} E(k) - E|}{E} \tag{2.8}$$

where:

- $E(k)$ is the energy of each extracted mode computed as:

$$E(k) = \sum_{n=1}^{N} |u_{k,n}|^2 \tag{2.9}$$

- E is the total energy of the input signal.

In matrix form, the energy loss criterion is:

$$\psi_{K,\alpha} = \frac{|\mathbf{1}_K^T \mathbf{E} - E|}{E} \tag{2.10}$$

Tab. 2.1: Grid search results for optimal K and α selection, highlighting $K = 3$ and $\alpha = 300$ with the lowest energy loss ($\psi_{K,\alpha} = 0.06$).

K	α	Energy Loss ($\psi_{K,\alpha}$)	Selected?
2	100	0.12	No
3	100	0.08	No
4	100	0.09	No
3	200	0.07	No
3	300	0.06	**Yes**

where $\mathbf{E}$ is the $K \times 1$ vector of mode energies:

$$\mathbf{E} = \begin{bmatrix} E(1) \\ E(2) \\ \vdots \\ E(K) \end{bmatrix} \tag{2.11}$$

Optimization of K and α

IAPVMD iteratively refines K and α by minimizing the energy loss rate. The process follows these steps:

1. Perform multiple VMD decompositions with various K and α combinations.
2. Calculate the energy loss rate $\psi_{K,\alpha}$ for each combination.
3. Select the (K, α) pair that yields the lowest energy loss.

Since α indirectly influences the bandwidth of each mode, IAPVMD adjusts it alongside K to balance mode separation and signal fidelity.

Grid Search Strategy for Optimal (K, α)

IAPVMD employs a **grid search** strategy to automate parameter selection, testing different K and α combinations and selecting the pair with the lowest energy loss [30]. An example of this process is shown in Table 2.1:

Impact of Adaptive Parameter Selection

By dynamically adjusting K and α, IAPVMD ensures that:

- Modes remain **compact** and **non-overlapping**.
- The noise and respiration harmonics are minimized, improving HR accuracy.
- The decomposition adapts dynamically to the varying signal conditions.

Qu et al. [22] reported that IAPVMD outperformed standard VMD and other methods like Ensemble Empirical Mode Decomposition (EEMD), Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN), achieving a Mean Absolute

Error (MAE) of 0.6 Beats per Minute (BPM) for BR and 2.1 BPM for HR, compared to 0.9 BPM and 3.2 BPM, respectively, with standard VMD. However, the experiments involve a single volunteer (e.g., one subject at 1 m distance with micromotions), and the optimal parameters ($K = 7$, $\alpha = 500$) are tailored to this individual, limiting scalability to multi-person environments.

2.3.3 Wavelet-Based Signal Reconstruction

Vo Dai et al. [28] proposed a 77 GHz FMCW radar system with MIMO channels, introducing a wavelet-based decomposition method using the CWT to enhance HR estimation. Unlike FFT, which provides only frequency-domain information and struggles with non-stationary signals due to spectral leakage [31], CWT offers time-frequency localization, making it better suited for separating overlapping respiration and heartbeat signals in multi-component vital signs.

The CWT processes the phase signal $x(t)$ using a custom wavelet $\Omega(t)$, which is derived from a subject-specific heartbeat template, as follows:

$$W(a,b) = \sum_{t=1}^{T} x(t)\Omega^* \left(\frac{t-b}{a}\right), \tag{2.12}$$

- $W(a,b)$ is the wavelet coefficient at scale a and translation b.
- $x(t)$ is the input signal, representing the phase at time t.
- $\Omega^*(t)$ is the complex conjugate of the wavelet function.

The HR is estimated by applying FFT to the wavelet coefficients $C_1(t) = W(1,t)$, which capture heartbeat-related oscillations at scale $a = 1$, as suggested by Vo Dai et al. [28]. Their approach achieved high accuracy (Pearson correlation coefficient greater than 0.9), but robustness is limited by distance and Radar Cross Section (RCS) variations, motivating other approaches with enhanced adaptability for multi-subject scenarios.

2.4 Noise Suppression for Vital Sign Extraction

While FMCW radar is highly sensitive to chest micro-movements, it is also susceptible to noise and environmental clutter, such as reflections from static objects or electronic interference [32]. Several advanced signal processing techniques have been developed to address these challenges.

2.4.1 Singular Spectrum Analysis for Noise Reduction

Wang et al. [20] proposed a three-stage vital signs measurement method using FMCW radar, combining Singular Spectrum Analysis (SSA) with an extended Differential and Cross-Multiply (DACM) phase extraction [33] and a VItal SIgns signals recONstruction (VISION) scheme to enhance weak signals and suppress noise. A key step in SSA-based processing is the Singular Value Decomposition (SVD) of the trajectory matrix, isolating periodic vital sign signals from noise [34, 35, 36]. The method uses

the top 1% eigenvalues for reconstruction, followed by an improved CEEMDAN to decompose signals into Intrinsic Mode Functions (IMFs) for respiration and heartbeat reconstruction. Rates are calculated via autocorrelation. Wang et al. reported 98.17% and 97.43% agreement for respiration and heartbeat compared to contact sensors, outperforming other methods by 2%.

However, the single-channel, offline-processed method limits scalability to multi-person scenarios, and its performance may degrade with distance or angle variations due to SSA's optimization for stable conditions.

2.5 Sparse Signal Recovery for Multi-Person Monitoring

A significant challenge in FMCW radar-based vital sign detection is handling cluttered environments and overlapping signals from multiple individuals. Traditional methods like beamforming or FFT-based frequency separation often fail in complex, real-world scenarios due to spectral leakage and localization errors in the presence of clutter. For instance, FFT-based methods suffer from spectral leakage, limiting frequency resolution even with zero-padding techniques [37]. In addition, localization approaches relying on spectral magnitudes struggle to distinguish human targets from static or vibrating objects like fans or pets [3, 38]. To address these limitations, Eder and Eldar [25] introduced a sparsity-based approach using a Single-Input Single-Output (SISO) FMCW radar, improving accuracy while maintaining low computational complexity.

2.5.1 Sparse Signal Representation for Multi-Person Detection

Eder and Eldar [25] employed Joint Sparse Recovery (JSR) to extract multiple respiration and heartbeat signals from cluttered environments. Unlike FFT-based methods, which suffer from spectral leakage when signals overlap[37], JSR formulates the problem as a sparse dictionary learning task, leveraging the fact that vital sign signals are sparse in the frequency domain (i.e., they consist of a few dominant frequency components).

Their generalized FMCW signal model for multiple subjects is defined over two time dimensions: the fast-time index n, which corresponds to range samples within each chirp, and the slow-time index l, which represents the chirp index. The model is given by [25, 39]:

$$y[n,l] = \sum_{m=1}^{M} x_m e^{j(2\pi f_m n T_f + \psi_m[l])} + w[n,l] \tag{2.13}$$

where:

- x_m is the amplitude of the signal from the m-th individual.
- f_m is the beat frequency, proportional to the subject's distance.
- T_f is the sampling interval in fast-time.
- $\psi_m[l]$ captures slow-time phase shift due to micro-motion.
- $w[n,l]$ is additive noise.

This formulation allows tracking multiple subjects without requiring multiple antennas or MIMO processing, reducing hardware complexity.

2.5.2 Construction of the Dictionary Matrix

JSR approach constructs a dictionary matrix $\mathbf{A}$ for the Vital Signs-based Dictionary Recovery (VSDR) method to represent respiration and heartbeat components in the frequency domain [25]. Human vital signs occupy distinct frequency bands. A set of basis functions that span these physiological frequency ranges, denoted as $\mathcal{F}$, is defined as follows:

$$\mathcal{F} = \{f_i \mid f_i \in [0.1, 0.4] \cup [0.9, 2.5]\} \tag{2.14}$$

where f_i are discrete frequencies within the BR and HR bands. The dictionary matrix $\mathbf{A}$ is constructed using sinusoidal basis functions for each frequency f_i:

$$A_{ij} = \cos(2\pi f_j t_i) \tag{2.15}$$

where:

- A_{ij} is the entry at time sample t_i and frequency f_j.
- t_i is the discrete time index, sampled at a given rate (100 Hz in [25]).
- f_j is a precomputed frequency from $\mathcal{F}$.

The dictionary matrix is explicitly represented as:

$$\mathbf{A} = \begin{bmatrix} \cos(2\pi f_1 t_1) & \cos(2\pi f_2 t_1) & \dots & \cos(2\pi f_N t_1) \\ \cos(2\pi f_1 t_2) & \cos(2\pi f_2 t_2) & \dots & \cos(2\pi f_N t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(2\pi f_1 t_M) & \cos(2\pi f_2 t_M) & \dots & \cos(2\pi f_N t_M) \end{bmatrix} \tag{2.16}$$

where M is the number of time samples, and N is the number of frequency components.

2.5.3 Sparse Signal Representation and Optimization

The measured signal is represented as a linear combination of dictionary elements in the VSDR method [25]:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}, \tag{2.17}$$

where:

- $\mathbf{y} \in \mathbb{R}^M$ is the observed signal vector containing M time-domain samples.
- $\mathbf{x} \in \mathbb{R}^N$ is the sparse coefficient vector representing the strength of each frequency component.
- $\mathbf{w} \in \mathbb{R}^M$ is the additive noise vector.

The optimization problem to estimate $\mathbf{x}$ is:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \gamma\|\mathbf{x}\|_1, \tag{2.18}$$

where γ is a regularization parameter ensuring sparsity. This approach enables accurate frequency estimation even in low Signal-to-Noise Ratio (SNR) conditions or multi-person scenarios.

2.5.4 Empirical Results and Comparison

Eder and Eldar [25] validated their sparsity-based approach using real physiological data from 30 individuals, showing that the VSDR method outperformed traditional techniques in multi-person scenarios at SNR $= 0$ dB. The median Root-Mean-Square Error (RMSE) across all subjects is:

- Respiration rate RMSE: 0.84 BPM (VSDR) vs. 1.03 BPM (FFT without zero-padding) [3, 38].
- Heart rate RMSE: 0.78 BPM (VSDR) vs. 1.28 BPM (FFT without zero-padding) [3, 38].

These results demonstrate the potential of sparsity-based methods for improving accuracy in cluttered environments with static and vibrating objects (e.g., fans). However, a trade-off exists between time window length and temporal localization, as longer windows (e.g., 60 s) improve frequency resolution but reduce temporal accuracy, posing a challenge for adapting the approach to environments with frequent movement.

2.6 MIMO Radar Configurations for Improved Accuracy

MIMO radar configurations enhance detection accuracy by exploiting multiple antennas to improve spatial resolution and SNR. A key method is Maximal Ratio Combining (MRC), which combines antenna signals based on channel quality to maximize SNR, improving the detection of weak vital signs such as HR. The combined signal is computed as follows:

$$y = \sum_{i=1}^{N} w_i x_i, \quad w_i = \frac{h_i^*}{\sum_{j=1}^{N} |h_j|^2} \tag{2.19}$$

x_i is the signal from the i-th channel, h_i is the channel coefficient, and w_i is the weight optimizing SNR across N channels.

Liu et al. [15] used a MIMO configuration with 2 transmit and 4 receive antennas, forming an 8-element virtual array, in their mmRH system. The system utilizes a combining technique like MRC to optimize SNR, improving HR estimation accuracy to a MAE of 0.27 BPM at 0.5 m compared to baseline methods [15].

Similarly, Vo Dai et al. [28] employed MRC with a 77 GHz MIMO FMCW radar using 192 channels, combined with Arctangent Demodulation (AD) and CWT. They reported an 18% improvement in HR estimation accuracy based on prior work [17, 18,

19]. The shared use of MIMO and MRC across these works highlights their potential to address mm-Wave challenges like propagation loss and multipath interference, making them particularly effective for non-contact vital sign monitoring in cluttered indoor environments.

While mmRH and related systems effectively handle indoor multipath interference, challenges remain in addressing random body movements and enabling multi-person detection, which are critical for broader practical deployment.

2.7 Conclusion

This chapter has reviewed key advancements in FMCW radar-based vital sign detection, covering beamforming, adaptive signal decomposition, noise suppression, sparse signal recovery, and MIMO processing. Techniques like MVDR beamforming and IAPVMD improve multi-target separation and signal decomposition, achieving MAE values as low as 2.1 BPM for HR [22]. Methods such as SSA, wavelet-based decomposition, and JSR enhance noise suppression and multi-person monitoring, with JSR reducing HR RMSE to 0.84 BPM in cluttered environments [25]. MIMO configurations further boost SNR and accuracy, though at the cost of increased complexity [15].

Despite these advancements, challenges remain, including handling motion artifacts, optimizing computational efficiency for real-time applications, and adapting to uncontrolled environments.

CHAPTER 3

Vital Sign Detection

Contents

3.1 Introduction 18
3.2 Human Target Identification 18
3.2.1 RAM Generation . . . 19
3.2.2 DC-value Suppression 21
3.2.3 Experimental Setup . 22
3.3 Vital Sign Extraction 30
3.3.1 FIFO Buffer for RAM Accumulation 31
3.3.2 Peak-Finder Module . 31
3.3.3 Process of Index Selection 33
3.3.4 Phase Computation . . 35
3.3.5 Application of Principle Component Analysis 36
3.3.6 Enhancing PCA Analysis with Spatial Information 39
3.3.7 Extracting Breathing and Heart Signals from Chest-Wall Data . 41
3.4 Empirical Investigation of Vital Sign Monitoring 45
3.4.1 Procedure and Key Metrics 48
3.4.2 Analysis of Metrics . . 49
3.4.3 Data Analysis 53
3.5 Conclusion 55

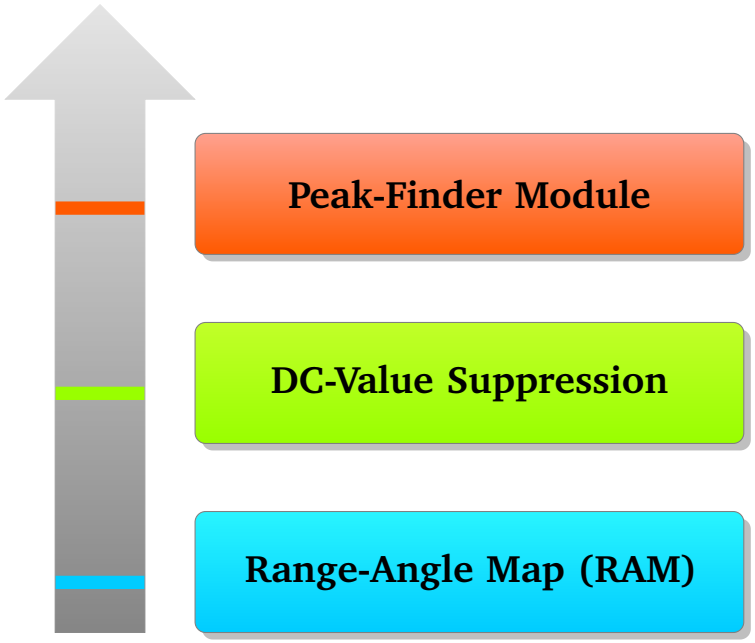


Fig. 3.1: Block diagram of the human target identification process.

3.1 Introduction

In this chapter, our focus is on the initial segment of the thesis, where we outline the proposed approach for computing vital signs using a MIMO FMCW radar, focusing on non-invasive detection of BR and HR. These physiological parameters are critical for assessing cardiovascular and respiratory health, making them essential in medical diagnostics and routine health monitoring [15, 40]. Unlike traditional contact-based sensors [5, 6], our radar-based method offers a non-invasive solution for tracking vital signs with high accuracy.

Our proposed module operates through two cascaded processing blocks. In the first block, the system identifies the coordinates of the subject in the experimental scene. The second block then tracks the reflected signals associated with the subject, facilitating the extraction and computation of BR and HR.

This chapter proceeds as follows: Section 3.2 describes the human target identification process, Section 3.3 details the vital sign extraction method, and Section 3.4 presents the experimental results.

3.2 Human Target Identification

The primary part of the vital sign detection algorithm, as illustrated in Fig. 3.1, involves the following steps:

1. Generating a RAM using MIMO FMCW radar.
2. Suppressing DC-values in the IQ signal.
3. Identifying local maxima in the refined RAM.

3.2.1 RAM Generation

The Range-Angle Map (RAM) is a fundamental component in radar signal processing, providing spatial information about detected objects within the radar's Field of View (FoV). The following algorithms outline the process of transforming a chirp signal into a RAM.

FFT-based:

The generation of RAM using FFT involves several key steps, as outlined below.

1. **Signal Formatting:** The input chirp signal, denoted as $\boldsymbol{\chi}$, is a real-valued vector of N samples. It is reshaped into a 2D matrix $\mathbf{\Gamma}$ of size $N \times L$, where L is the number of elements in the synthesized virtual antenna array formed by MIMO processing, as detailed in Appendix B.

2. **Column-wise FFT:** A P-point FFT is applied along each column of $\mathbf{\Gamma}$, resulting in a range-profile matrix $\mathbf{\Lambda}$ of size $P \times L$. Each row of this matrix corresponds to a specific range bin, capturing spatial attributes of objects at the given range.

3. **Row-wise FFT:** A Q-point FFT is applied along each row of $\mathbf{\Lambda}$, yielding the final RAM denoted as $\mathbf{\Psi}$ of size $P \times Q$.

To mitigate discretization effects, such as coarse frequency sampling, zero-padding is applied prior to both FFT computations [25, 37]. Specifically, zeros are appended to the columns of $\mathbf{\Gamma}$ and rows of $\mathbf{\Lambda}$, increasing their lengths to P and Q, respectively. This results in finer frequency bins and enhanced spectral resolution in the fast-time and angular dimensions.

In FMCW radar, a series of chirp signals is transmitted as frames. The above algorithm is applied to all M chirps within a measurement cycle, where $M = U \times W$, with U denoting the number of chirps per frame and W the number of frames in the measurement cycle. A visual representation of this process is provided in Fig. 3.2.

Beamforming:

Another method to generate a precise RAM is through beamforming. This technique, which offers improved angular resolution in MIMO FMCW radar, relies on the beam steering vector defined for an angle θ as follows [41].

$$\mathbf{a}(\theta) = \left[1, e^{j\frac{2\pi}{\lambda}d\sin(\theta)}, \ldots, e^{j(L-1)\frac{2\pi}{\lambda}d\sin(\theta)}\right]^T, \tag{3.1}$$

where λ is the free-space wavelength corresponding to the central frequency of the FMCW radar, and d is the inter-element spacing of the virtual antenna array.

The RAM is generated using Algorithm 1, which processes the range-profile matrix $\mathbf{\Lambda}$ from Section 3.2.1. The algorithm constructs a steering matrix $\mathbf{V}$ from steering vectors $\mathbf{a}(\theta_y)$ for a grid of Y angles $\theta_1, \ldots, \theta_Y$, uniformly sampled over a specified angular range (e.g., $[-90°, 90°]$). For each chirp, Algorithm 1 calculates $\mathbf{\Psi}[i, j]$ as the complex inner product between the i–th row of $\mathbf{\Lambda}$ (range bin) and the j–th column of $\mathbf{V}$ (angle bin). This operation quantifies the signal energy associated with a specific

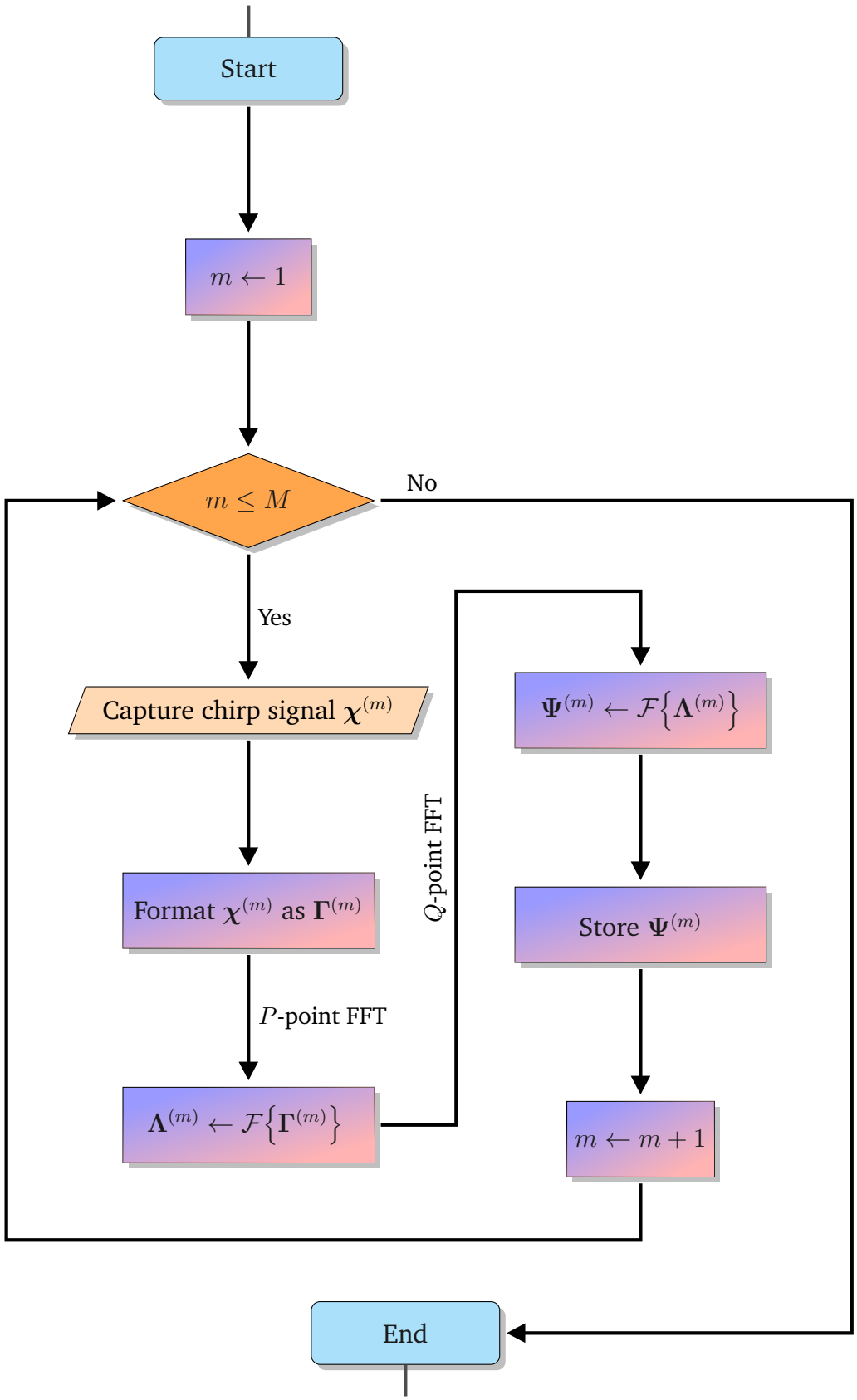


Fig. 3.2: Flowchart illustrating the RAM generation process for all chirp signals within a measurement cycle. The process involves iterative chirp capture, formatting, and Fourier transforms to produce RAMs.

Algorithm 1 Range-Angle Map using beamforming

procedure RAM_BEAM($\mathbf{\Lambda}$) ▷ $\mathbf{\Lambda}$: range-profile matrix
  Construct the steering matrix $\mathbf{V}$:

$$\mathbf{V} = \Big[\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \cdots \quad \mathbf{a}(\theta_Y)\Big]$$

  Initialize an empty RAM matrix $\mathbf{\Psi}$ of size $P \times Y$
  for $i \leftarrow 1$ to P **do**
    for $j \leftarrow 1$ to Y **do**
      $\mathbf{\Psi}[i, j] \leftarrow \mathbf{\Lambda}[i, :] \cdot \mathbf{V}[:, j]$
    end for
  end for
  return $\mathbf{\Psi}$
end procedure

range-angle pair, enabling precise localization of the human subject for vital sign extraction as detailed in Section 3.3.

3.2.2 DC-value Suppression

Static clutter must be attenuated in RAM to enable accurate discrimination between genuine human targets and false non-human targets in the experimental setup. The standard deviation [42, 43], and the Exponential Moving Average (EMA) algorithm, derived from the Moving Target Indicator (MTI) [44], are the two most commonly used techniques for suppressing stationary targets while preserving moving ones. Building on these methods, the study in [45] presents a modified EMA approach to effectively attenuate both stationary and undesired moving clutter indoors. Alternatively, [21] proposes a curve-length method that identifies human subjects by estimating the length of the I/Q signal trajectory in the experimental scene.

In this thesis, we introduce an innovative approach for mitigating stationary clutter in indoor human identification. Our proposed method focuses on monitoring the phase shift of the radar's raw data along the slow-time axis. Unlike stationary clutter, the human chest wall exhibits a pronounced phase shift between successive chirps in FMCW radars, attributed to respiration and heartbeat. By employing a slow-time DC-value suppression technique, we effectively diminish the impact of stationary clutter, facilitating precise localization of human subjects. The approach has minimal computing complexity as well as superior performance, making it an appealing alternative to existing schemes in this field.

Proposed Method

For each chirp $m \in \{1, \ldots, M\}$, the obtained RAM is a complex-valued matrix $\mathbf{\Psi}^{(m)}$ of dimensions $P \times Q$. Each matrix $\mathbf{\Psi}^{(m)}$ is characterized by four components:

- $\mathbf{\Upsilon}$ and $\mathbf{\Phi}$, the DC component matrices of dimensions $P \times Q$ for the real and imaginary parts, respectively, of the 2D Fourier transform.

- $\boldsymbol{\Theta}^{(m)}$ and $\boldsymbol{\Omega}^{(m)}$, the sinusoidal component matrices of dimensions $P \times Q$ for the real and imaginary parts, respectively, arising from phase shifts due to infinitesimal displacements.

Each matrix $\boldsymbol{\Psi}^{(m)}$ for chirp m is expressed as sum of DC and sinusoidal components for the real and imaginary parts:

$$\boldsymbol{\Psi}^{(m)} = \boldsymbol{\Theta}^{(m)} + \boldsymbol{\Upsilon} + j\Big(\boldsymbol{\Omega}^{(m)} + \boldsymbol{\Phi}\Big), \tag{3.2}$$

where operations are element-wise. The DC components are suppressed to estimate the sinusoidal components. The real part's sinusoidal component is:

$$\boldsymbol{\Theta}^{(m)} = \Re\Big(\boldsymbol{\Psi}^{(m)}\Big) - \underbrace{\frac{1}{M}\sum_{m'=1}^{M} \Re\Big(\boldsymbol{\Psi}^{(m')}\Big)}_{\boldsymbol{\Upsilon}}, \tag{3.3}$$

and the imaginary part's sinusoidal component is:

$$\boldsymbol{\Omega}^{(m)} = \Im\Big(\boldsymbol{\Psi}^{(m)}\Big) - \underbrace{\frac{1}{M}\sum_{m'=1}^{M} \Im\Big(\boldsymbol{\Psi}^{(m')}\Big)}_{\boldsymbol{\Phi}}, \tag{3.4}$$

where $\Re(\cdot)$ and $\Im(\cdot)$ denote the element-wise real and imaginary parts, respectively. A clutter-free RAM matrix $\boldsymbol{\Pi}^{(m)}$ of dimensions $P \times Q$ is generated as:

$$\boldsymbol{\Pi}^{(m)} = \boldsymbol{\Theta}^{(m)} + j\boldsymbol{\Omega}^{(m)}, \tag{3.5}$$

to facilitate human identification and vital sign detection within the radar's FoV. The process is illustrated in Fig. 3.3.

3.2.3 Experimental Setup

This thesis employs the Radarbook2 (RBK2), a 77 GHz MIMO FMCW radar developed by INRAS GmbH, for radar signal processing experiments (see Fig. B.1 in Appendix B for hardware details). The RBK2 communicates with the host computer via TCP/IP, supporting a maximum data rate of 550 MBit/s to ensure reliable data transfer [46]. The RF front-end's Analog-to-Digital Converters (ADCs) support clock frequencies of 20 MHz and 40 MHz for flexible signal sampling.

In the primary measurement mode, the timing unit of the radar, **SeqTrig**, which operates within the FPGA section of the system, generates a trigger signal for the RF synthesizer to initiate the predefined chirp sequence. Additionally, it signals the ADC data interface, **DatInt**, to capture data during the up-chirp. The radar utilizes a one-dimensional Uniform Linear Array (ULA) with 2 transmit (Tx) and 16 receive (Rx) antennas, employing Time Division Multiplexing (TDM) to synthesize a virtual array of 31 unique elements with an inter-element spacing of 0.5λ, where λ is the free-space wavelength at 77 GHz. Detailed hardware specifications and virtual array calculations are provided in Appendix B.

Table 3.1 summarizes the configuration parameters of the RBK2 radar for the proposed application. Based on these values, and assuming the speed of light

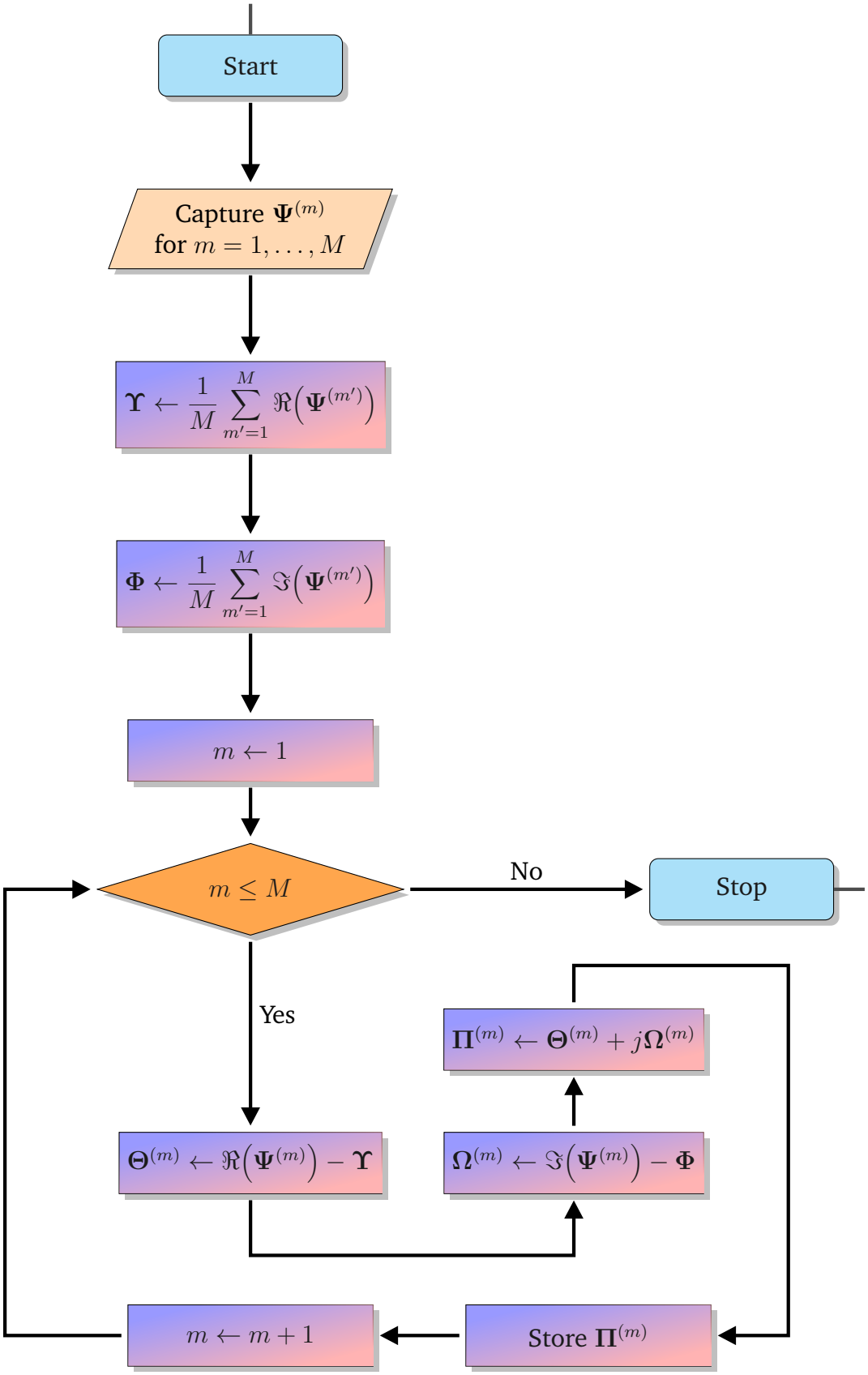


Fig. 3.3: Flowchart illustrating DC-value suppression concept for human identification. All matrix operations are element-wise.

$c = 3 \times 10^8$ m/s, the theoretical resolution limits and bin spacings in the range and angular (azimuthal) domains are calculated as follows.

Range Domain

The range resolution, which represents the minimum distance required to distinguish two targets in range, is defined by [3, 47]:

$$\Delta r_{\text{res}} = \frac{c}{2B} \tag{3.6}$$

where $B = 4$ GHz is the radar bandwidth. Substituting the value yields $\Delta r_{\text{res}} \approx 3.75$ cm.

The range bin spacing, which determines the distance between adjacent FFT bins in the range profile, is given by:

$$\Delta r_{\text{bin}} = \frac{cN}{2BP} \tag{3.7}$$

where $N = 256$ is the number of fast-time samples per chirp, and $P = 512$ is the range FFT size. Substituting these values yields a range bin spacing of approximately $\Delta r_{\text{bin}} \approx 1.87$ cm. This fine discretization indicates that the range profile is sampled more densely than the system's intrinsic range resolution.

Angular Domain

The angular resolution, determined by the aperture of the virtual antenna array, is defined as the 3-dB beamwidth of the array's radiation pattern, given by [47]:

$$\Delta\theta_{\text{res}} = \frac{\lambda}{2d(L-1)} \cdot \frac{180}{\pi} \tag{3.8}$$

Assuming a ULA geometry with half-wavelength spacing $d = \lambda/2$, the expression simplifies to:

$$\Delta\theta_{\text{res}} \approx \frac{1}{L-1} \cdot \frac{180}{\pi} \tag{3.9}$$

where $L = 31$ is the number of virtual antennas. This results in $\Delta\theta_{\text{res}} \approx 1.91°$.

The angular grid for angle estimation, obtained via FFT across the antenna array, is computed by sampling the $\sin(\theta)$ domain linearly. For an angular FFT size $Q = 512$, the $\sin(\theta)$ values are sampled from -1 to $+1$ with a resolution of:

$$\Delta \sin(\theta) = \frac{2}{Q} \tag{3.10}$$

The angle corresponding to the k-th FFT bin, where $k = -\frac{Q}{2}, \ldots, \frac{Q}{2} - 1$, is given by:

$$\theta_k = \arcsin\left(k \cdot \frac{2}{Q}\right) \cdot \frac{180}{\pi} \tag{3.11}$$

This results in a non-uniform angular grid in degrees, with finer spacing near $\theta = 0°$ and coarser spacing near $\theta = \pm 90°$. For small angles, the bin spacing approximates

Tab. 3.1: Simulation parameters for radar-based human identification on RBK2

Parameter	Description	Value/Unit
Radar Parameters		
B	Chirp bandwidth of the FMCW radar	4 GHz
f_c	Central operating frequency	75 GHz
Timing Parameters		
f_s	Fast-time sampling frequency	2 MHz
f_{rr}	Slow-time frame repetition rate	20 Hz
T_{up}	Duration of the up-chirp	120 μs
N	Number of samples per chirp	256
U	Number of chirps per frame	1
W	Number of frames in a measurement cycle	100
M	Number of chirps in a measurement cycle	100
Processing Parameters		
L	Number of virtual antennas	31
P	Size of the range-FFT	512
Q	Size of the angle-FFT	512

$\Delta\theta_{\text{bin}} \approx \frac{2}{Q} \cdot \frac{180}{\pi} \approx 0.223°$, which is finer than the system's angular resolution of $1.91°$, enabling accurate angle estimation and interpolation.

Experimental Procedure and Validation of Results

The experimental scenario shown in Fig. 3.4 is employed to demonstrate the effectiveness of the proposed technique for indoor human localization. In this experiment, alongside a single individual positioned in front of the RBK2, we created significant static clutter using two metallic corner cubes. To ensure stability, the radar system and corner cubes are mounted on tripods.



Fig. 3.4: Data acquisition setup with a single human subject: the RBK2 and corner cubes (CC1 and CC2) are placed on tripods for stability.

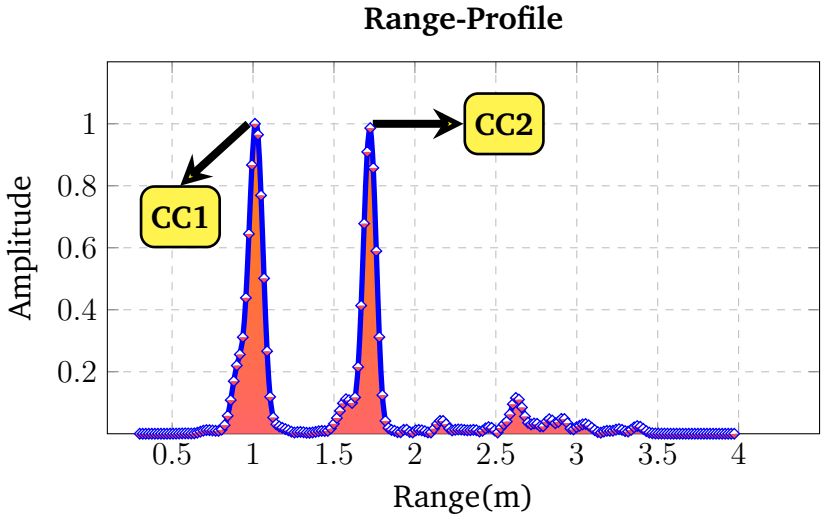


Fig. 3.5: Range-profile of the test scenario illustrated in Fig. 3.4, obtained from one of the receiving antennas. While the positions of the corner cubes (CC1 and CC2) are highlighted, the human subject's location remains unclear.

The normalized range-profile graph in Fig. 3.5 illustrates the specified test scenario, highlighting the positions of two corner cubes (CC1 and CC2) at approximately $1\ \mathrm{m}$ and $1.75\ \mathrm{m}$ from the radar, respectively. However, identifying the precise range of the human subject from the raw range-profile alone is challenging due to the dominant reflections from these metallic objects.

The RAM data, processed using FFT and beamforming methods, are shown in Fig. 3.6 and Fig. 3.7, respectively. In the raw RAM, the human subject's position is obscured because the transmitted chirps reflect more strongly off the pyramidal metallic corner cubes than off the human subject or other scene elements. Applying DC-value suppression mitigates this issue, enabling clear identification of the human subject as a distinct peak, well-separated from surrounding clutter.

Comparing Fig. 3.7a with Fig. 3.6a and Fig. 3.7b with Fig. 3.6b reveals that beamforming consistently yields sharper, more defined peaks, indicating superior range-angle resolution and higher SNR. In contrast, the FFT-based RAM shows broader peaks with greater noise interference, resulting in lower SNR. Given beamforming's enhanced performance in resolution and noise suppression, it is adopted exclusively for subsequent analyses in this thesis.

Fig. 3.8 presents supplementary graphs for the outlined experiment, illustrating key aspects of the test scenario. Fig. 3.8a displays the raw Range-Slow-Time Map (RSM), constructed by stacking range profiles from a single antenna element column-wise across 50 frames. This map reveals consistent peaks at the range bins corresponding to corner cubes CC1 and CC2, located at approximately $1\ \mathrm{m}$ and $1.75\ \mathrm{m}$ from the radar, respectively.

Fig. 3.8b shows the RSM after applying DC-value suppression along the slow-time axis, revealing amplitude fluctuations in the range bin of the human subject (HS) attributed to respiratory activity. This fluctuation is in stark contrast to the steady peaks observed in Fig. 3.8a. Corner cubes, designed to reflect radar signals strongly with minimal scattering, produce consistent returns, whereas the human

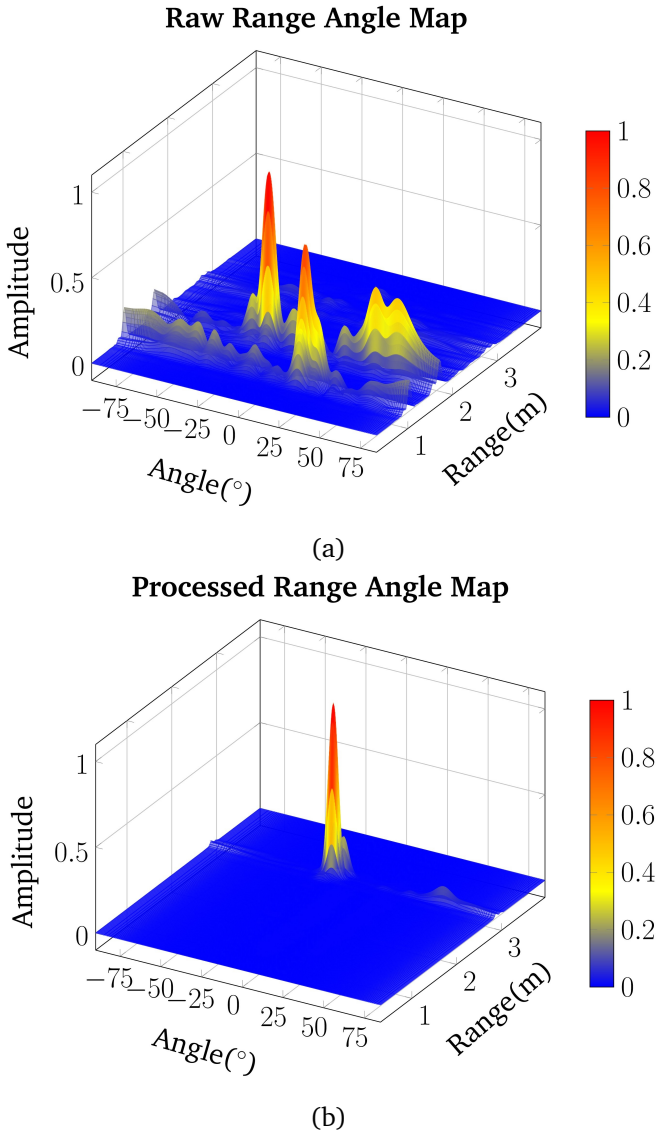


Fig. 3.6: RAM of the test scenario shown in Fig. 3.4, generated by the FFT method: (a) prior to applying the DC-value suppression technique; (b) following the DC-value suppression technique

body's irregular shape and motion scatter signals diffusely, hindering consistent peak detection.

Fig. 3.8c illustrates the Angle-Slow-Time Map (ASM) produced by the FFT method, enhanced through DC-value suppression. In contrast, Fig. 3.8d presents a similar ASM generated using the beamforming technique. This comparison highlights the effectiveness of DC-value suppression across different processing methods. Both figures clearly indicate the angular position of the detected human subject at $\theta = -16°$. Notably, the beamforming technique produces a narrower vertical strip compared to the FFT method. This difference suggests that beamforming effectively isolates the subject from background clutter, resulting in improved resolution and higher SNR.

We analyze the RSM by computing the curve-length of I/Q trajectories along the slow-time axis, as described in [21]. This metric estimates the curve-length of the In-phase (I) and Quadrature (Q) trajectory at a specified range-angle bin (p, q)

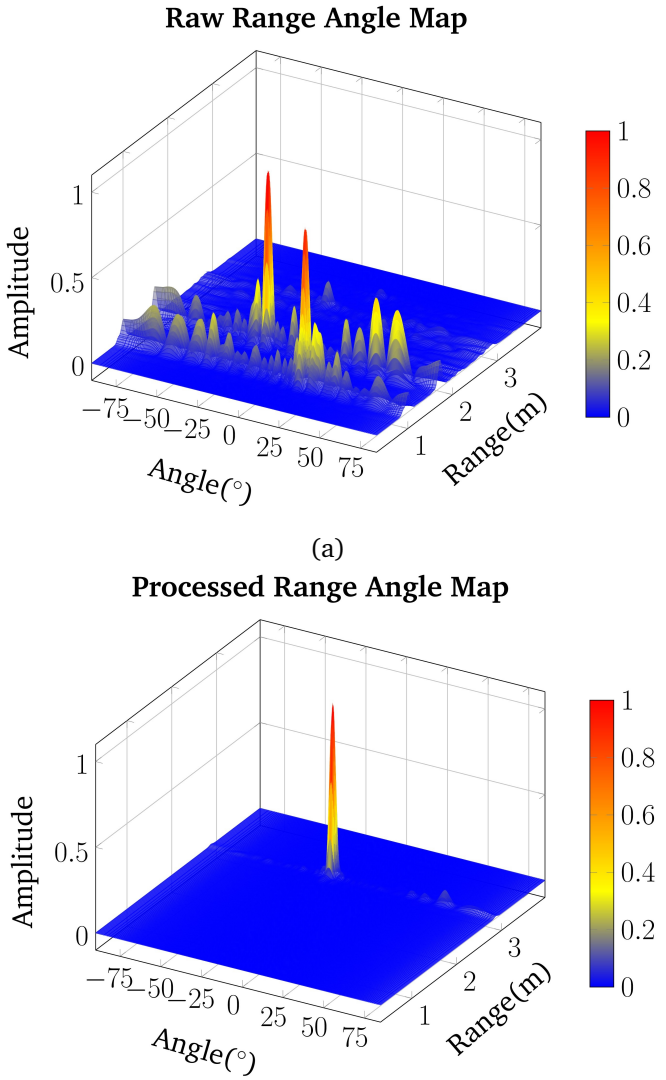


Fig. 3.7: RAM of the test scenario shown in Fig. 3.4, constructed using the beamforming method (Algorithm 1): (a) before applying the DC-value suppression technique; (b) after applying the DC-value suppression technique.

in relation to slow-time or its discrete analog, the chirp index m. Given that the I/Q components of $\Psi_{p,q}^{(m)}$ are represented as $I[m] = \Re\left(\Psi_{p,q}^{(m)}\right)$ and $Q[m] = \Im\left(\Psi_{p,q}^{(m)}\right)$, respectively, the curve-length metric is calculated as:

$$L(\Psi_{p,q}) = \sum_{m=2}^{M'} \sqrt{(I[m] - I[m-1])^2 + (Q[m] - Q[m-1])^2}, \tag{3.12}$$

where M' denotes the total number of chirps used in the computation. This equation computes the total curve-length by summing the Euclidean distances between consecutive samples in the I/Q plane. Because changes in the I/Q components reflect changes in phase, the curve-length metric quantifies the accumulated phase variation across chirp indices. The detected peak's range in Fig. 3.9 corresponds to the identified human's range in the processed RAM (Fig. 3.8b), which is 2.9 m. However, from

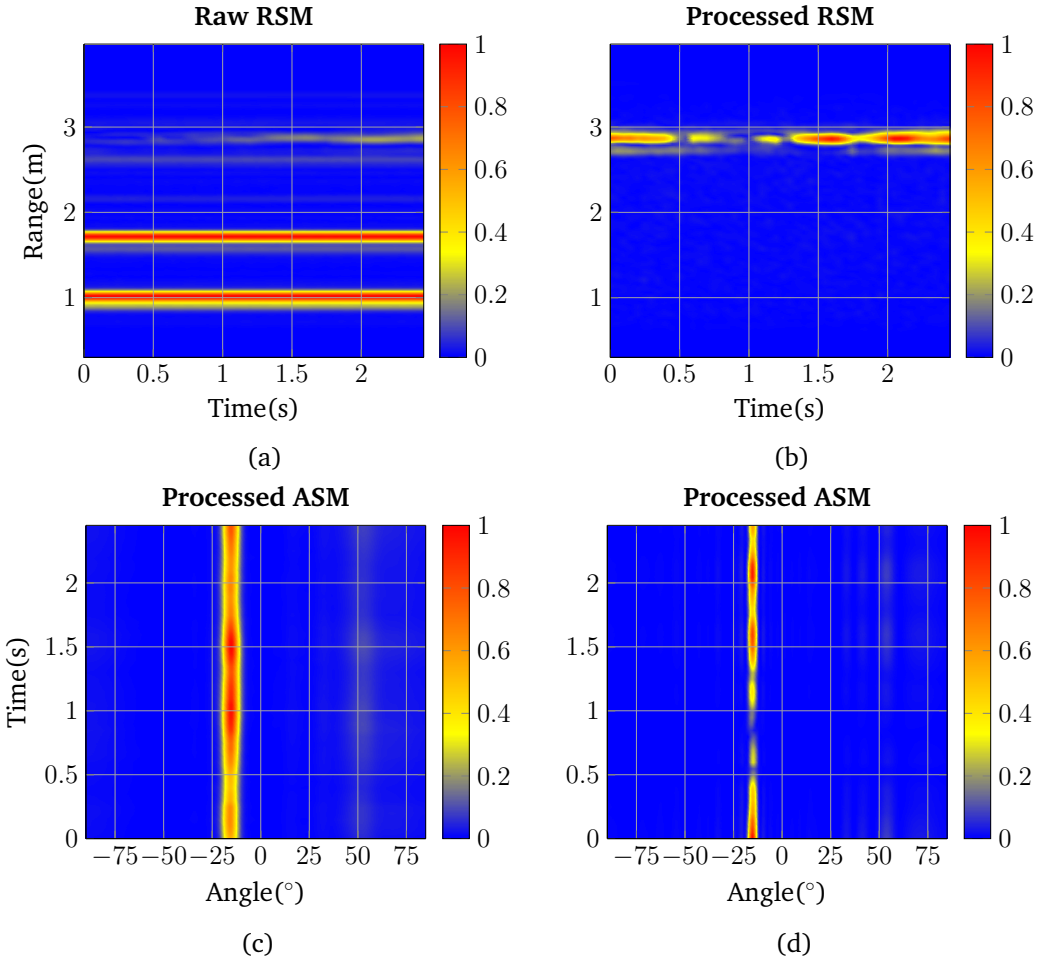


Fig. 3.8: Empirical results of the test scenario shown in Fig. 3.4: (a) the raw RSM; (b) the RSM after DC-value suppression along the slow-time axis; (c) the processed ASM constructed using the FFT-based approach; (d) the processed ASM generated via beamforming.

an algorithmic standpoint, DC-value suppression is simpler than Euclidean distance computation.

Fig. 3.10 presents a trajectory plot that highlights the effects of the DC-value suppression technique. This plot visualizes the real and imaginary components of the processed RAM obtained through the beamforming approach across a measurement cycle at two distinct positions. The green markers, which roughly outline a circular path, represent the range and angle bins related to the test participant's location, while the magenta point denotes one of the corner cubes used in the experiment (CC1 in Fig. 3.4).

This graphical representation provides valuable insights into our experiment. The circular trajectory formed by the green markers suggests a constant angular movement, potentially indicating the presence of a non-stationary object. This cyclic pattern is likely due to chest wall movements from respiratory or cardiovascular activities. In contrast, the magenta point remains fixed at the center of the coordinate system over all chrip signals, reflecting the stationary nature of the corner cube.

By analyzing these patterns, we can gain a deeper understanding of how the

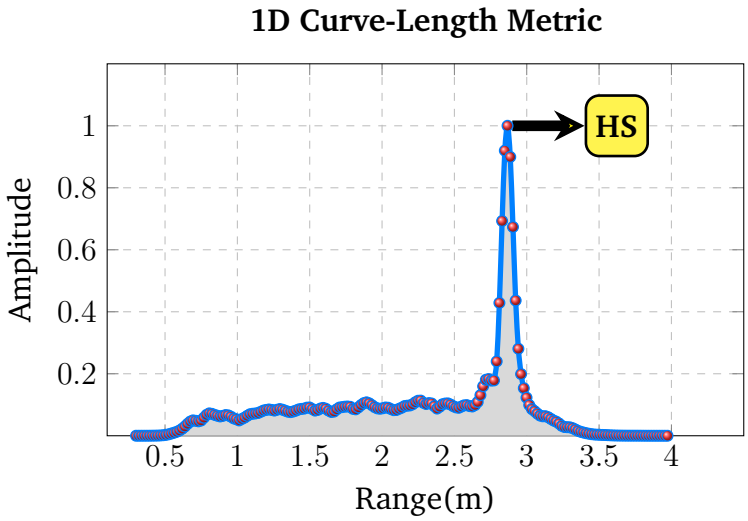


Fig. 3.9: One-dimensional version of the curve-length metric proposed in [21], demonstrating the range of the identified human subject as $2.9\ \mathrm{m}$, consistent with the range detected in Fig. 3.8b

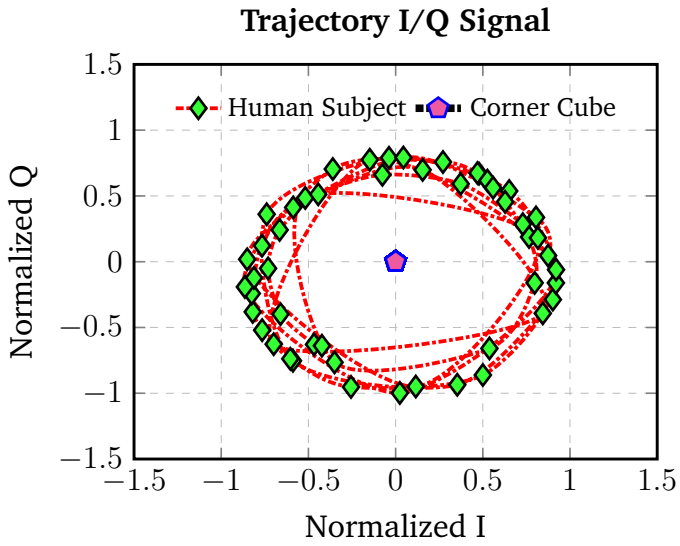


Fig. 3.10: Trajectory plot showing the effect of DC-value suppression. Green markers indicate the test participant's location, while the magenta point denotes CC1.

DC-value suppression technique influences the detection of moving objects while effectively eliminating stationary ones in our experimental setup.

3.3 Vital Sign Extraction

The schematic in Fig. 3.11 offers a visual representation of the components comprising our vital sign estimation block. The following sections provide further clarification on these components.

3.3.1 FIFO Buffer for RAM Accumulation

For vital sign detection, we implemented a First-In-First-Out (FIFO) buffer to enhance processing efficiency and responsiveness in real-time monitoring, as follows:

- **Swift Initialization**: The FIFO buffer collects four measurement cycles, each consisting of 100 frames (one chirp per frame). This results in a total of $M_{\text{FIFO}} = 400$ chirps before the analysis begins, ensuring sufficient data for accurate vital sign monitoring with an initial delay of $20\,\text{s}$ (at a frame rate of $20\,\text{Hz}$).
- **Continuous Updates**: After collecting 400 chirps, the FIFO buffer updates vital sign estimates with each new measurement cycle (100 chirps every $5\,\text{s}$), as illustrated in Fig. 3.12. It retains 300 previous chirps (three previous measurement cycles) and appends 100 new chirps (one measurement cycle), maintaining a sliding window of $M_{\text{FIFO}} = 400$ chirps to minimize latency for the subject.
- **Minimized Delays**: Immediate processing of each new measurement cycle reduces delays in updating vital sign metrics, such as BR and HR, ensuring timely detection of physiological changes.
- **Scalability**: The FIFO buffer size, determined by the number of frames per measurement cycle, is configurable to meet different system requirements.

Methods for Buffer Status Management

- `is_full()` **Method**: Checks if the FIFO buffer is full (i.e., four measurement cycles, or 400 chirps). When full, it triggers vital sign analysis.
- `is_empty()` **Method**: Determines if the FIFO buffer is empty, useful for initialization or when no frames are collected.

3.3.2 Peak-Finder Module

The peak-finder module processes the averaged magnitude matrix $\bar{\mathbf{\Pi}}$ with dimensions $P \times Q$. The elements of this matrix are defined as follows:

$$\bar{\Pi}_{p,q} = \frac{1}{M_{\text{FIFO}}} \sum_{m=1}^{M_{\text{FIFO}}} |\Pi_{p,q}^{(m)}|, \quad p = 1, \ldots, P, \quad q = 1, \ldots, Q, \tag{3.13}$$

where $\Pi_{p,q}^{(m)}$ is the clutter-free RAM element for chirp m at range-angle bin (p, q). To identify potential peaks, a fixed magnitude threshold p_t is applied. The algorithm iterates through each element $\bar{\Pi}_{p,q}$ using nested loops and flags it as a peak if $\bar{\Pi}_{p,q} \geq p_t$. If this condition is met, the algorithm further evaluates the local neighborhood of $\bar{\Pi}_{p,q}$ to detect significant deviations, indicating the presence of a peak. Identified peaks are stored in a set $\mathcal{H}$, with each entry containing the value $\bar{\Pi}_{p,q}$ and range-azimuth bin indices (p, q). The peaks in $\mathcal{H}$ are sorted by their values. A visualization is shown in Fig. 3.13, processing $\bar{\mathbf{\Pi}}$ as in (3.13).

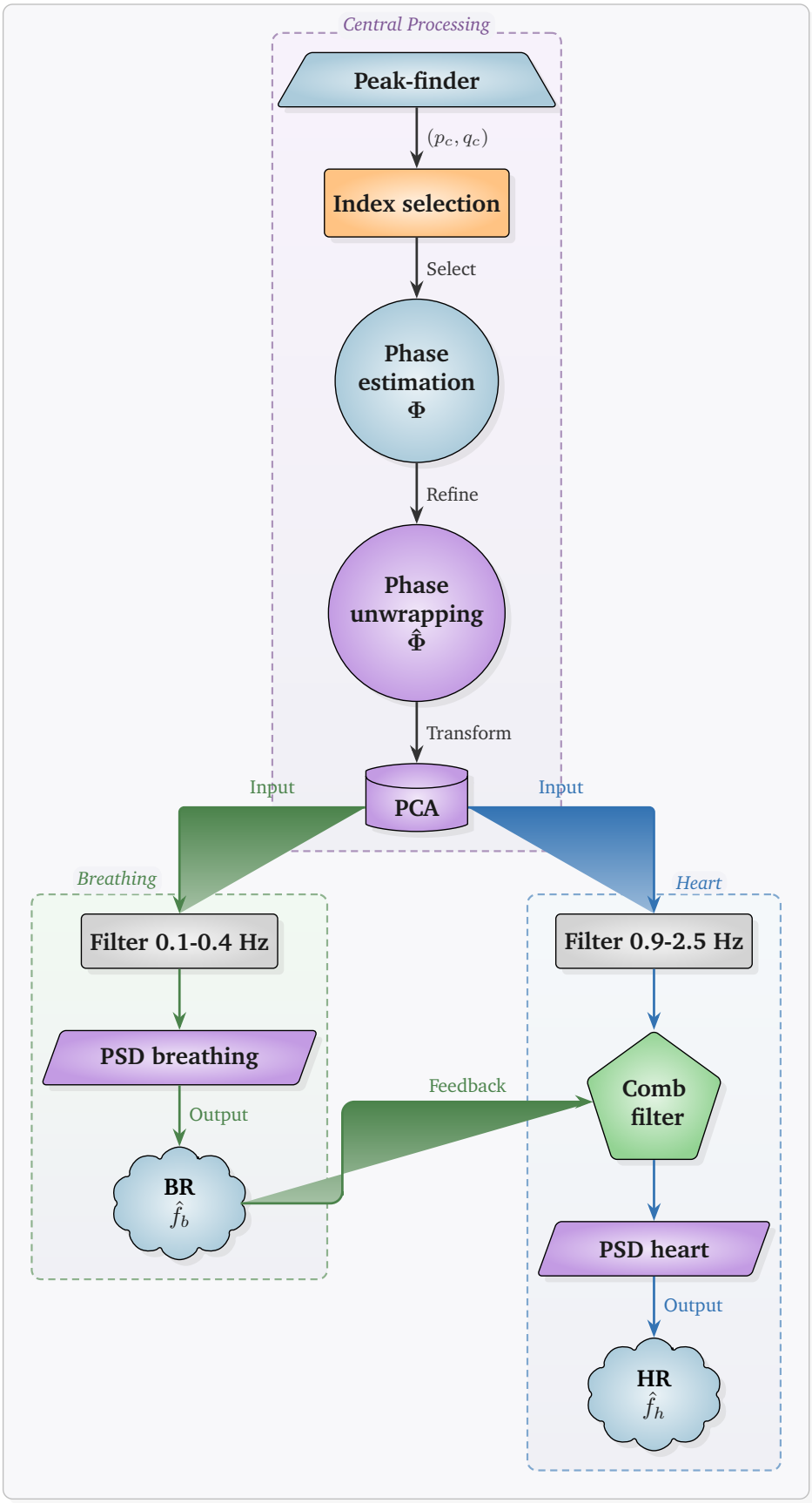


Fig. 3.11: Schematic illustration of the proposed vital sign estimation block.

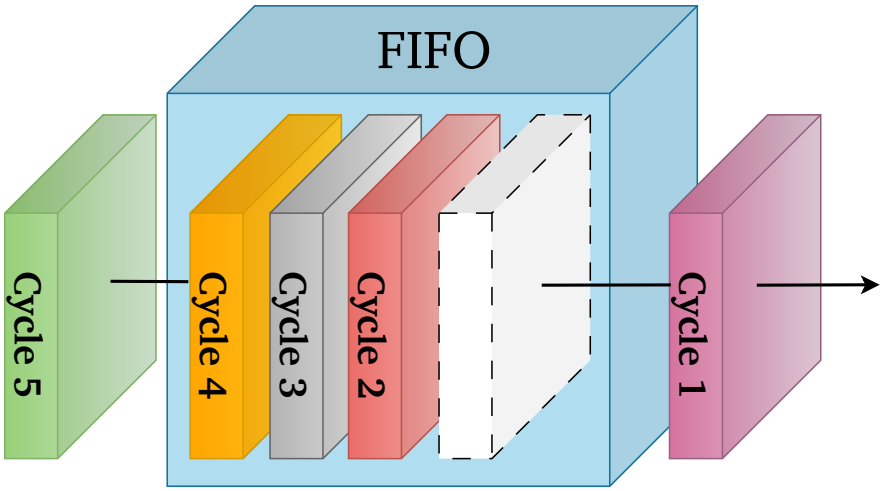


Fig. 3.12: FIFO buffer mechanism in our real-time vital sign monitoring system, using a sliding window to ensure timely detection of physiological changes.

3.3.3 Process of Index Selection

Unlike traditional methods, our approach acknowledges that targets on a RAM often appear as clusters of multiple points. This implies that an individual's vital signs are distributed across several points on the RAM, rather than concentrated at a single point. Each point in the cluster captures unique aspects of a person's vital signs. This distribution can be attributed to the varying interactions between the radar waves and the tissues surrounding the chest, as different tissues reflect radar waves differently.

Once the target is identified, we define a 9×9 window centered around the peak index, which is the point on the RAM with the highest signal strength. This window isolates the most likely points that represent the human target, allowing us to focus the analysis on the most relevant data.

The process of index selection in a 9×9 analysis window is based on two key criteria: **signal power level** and **cross-correlation**. These criteria ensure the selection of indices that exhibit robust signals and consistent patterns for further analysis.

Criterion 1: Signal Power Level

The signal power level criterion is used to identify indices with strong signals. The power of the clutter-free RAM is represented by a 2D matrix, $\mathbf{P}$, calculated across the 9×9 window. The layout of $\mathbf{P}$ is as follows:

$$\mathbf{P} = \begin{pmatrix} P_{-4,-4} & \dots & P_{-4,4} \\ \vdots & \ddots & \vdots \\ P_{4,-4} & \dots & P_{4,4} \end{pmatrix} \tag{3.14}$$

Each entry in this matrix represents the signal power level at a specific range-angle bin. The elements of this matrix are calculated using the following formula:

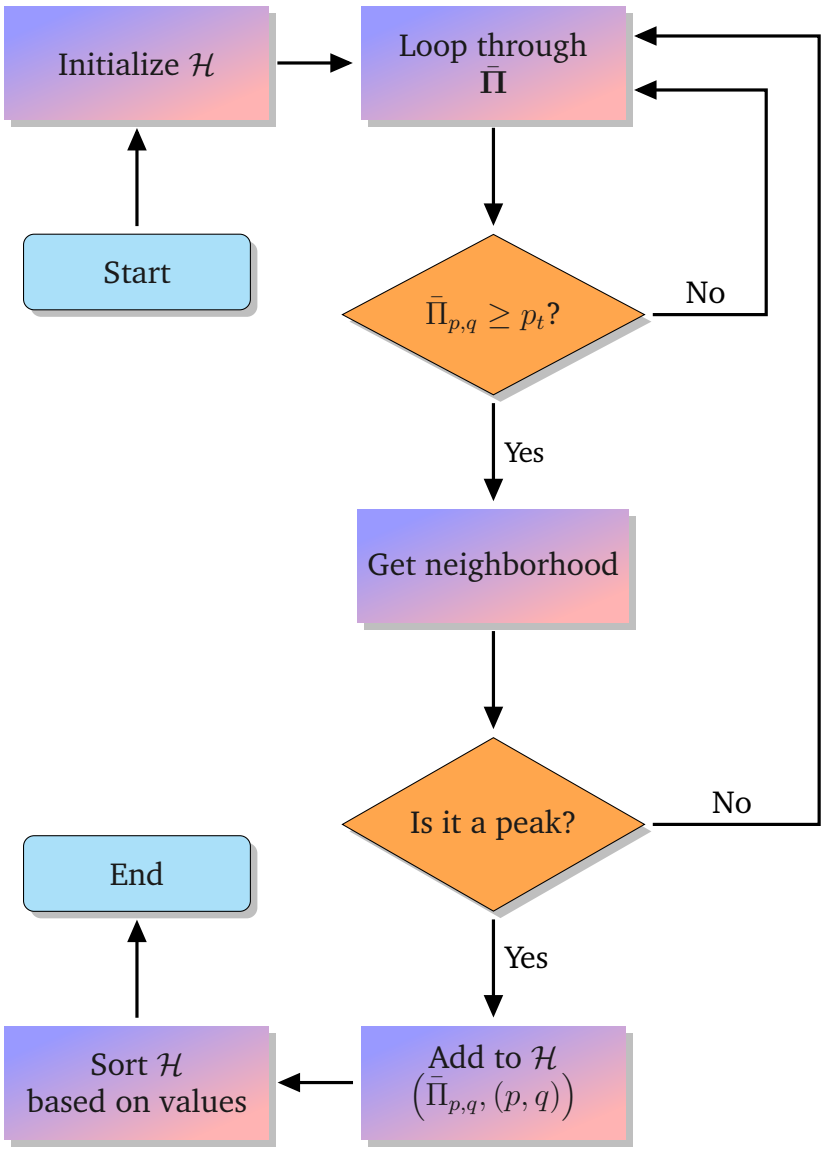


Fig. 3.13: Visualization of the peak-finder module, processing the averaged magnitude matrix $\bar{\mathbf{\Pi}}$ defined in Equation (3.13) to identify peaks, storing their values and range-angle bin indices (p, q) in $\mathcal{H}$.

$$P_{i,j} = 10 \log_{10} \left(\frac{1}{M_{\text{FIFO}}} \sum_{m=1}^{M_{\text{FIFO}}} \left| \Pi^{(m)}_{p_c+i, q_c+j} \right|^2 \right) \text{ (in dB)}, \tag{3.15}$$

where $P_{i,j}$ is the power in decibels, $(i, j) \in \{-4, -3, \ldots, 3, 4\}$ represents the offset from the central index (p_c, q_c) within the 9×9 window, $\Pi^{(m)}_{p_c+i, q_c+j}$ is the clutter-free RAM element for chirp m. This criterion helps minimize the influence of noise and irrelevant data.

Criterion 2: Cross-Correlation

The cross-correlation criterion is used to assess the consistency of signal patterns around the central index. The cross-correlation matrix, $\mathbf{C}$, compares each index in

the 9×9 window with the central index (p_c, q_c). The structure of $\mathbf{C}$ is as follows:

$$\mathbf{C} = \begin{pmatrix} C_{-4,-4} & \dots & C_{-4,4} \\ \vdots & \ddots & \vdots \\ C_{4,-4} & \dots & C_{4,4} \end{pmatrix}. \tag{3.16}$$

Each entry in this matrix represents the cross-correlation between the center of the window and another index within the 9×9 grid. The cross-correlation is calculated as:

$$C_{i,j} = \frac{1}{M_{\text{FIFO}}} \sum_{m=1}^{M_{\text{FIFO}}} \Pi_{p_c,q_c}^{(m)} \cdot \left(\Pi_{p_c+i,q_c+j}^{(m)}\right)^*, \tag{3.17}$$

where $(\cdot)^*$ denotes the complex conjugate [48].

Criteria Integration

The final selection of indices relies on both signal power level and cross-correlation within a 9×9 window. The following conditions are applied:

- **Condition 1:** Indices where the signal power $P_{i,j}$ is within 5 dB of the central index power $P_{0,0}$. This is defined as the set $\mathcal{S}_1$:

$$\mathcal{S}_1 = \{(p_c + i, q_c + j) \mid P_{i,j} \geq P_{0,0} - 5\,\text{dB}\}, \quad \forall i, j : -4 \leq i, j \leq 4. \tag{3.18}$$

  In short, indices (i, j) are selected if their power is not less than the central power minus 5 dB.

- **Condition 2:** Indices where the cross-correlation $C_{i,j}$ exceeds 70% of the central autocorrelation $C_{0,0}$. This is defined as the set $\mathcal{S}_2$:

$$\mathcal{S}_2 = \{(p_c + i, q_c + j) \mid C_{i,j} > 0.7 \times C_{0,0}\}, \quad \forall i, j : -4 \leq i, j \leq 4. \tag{3.19}$$

Both conditions must be met to ensure robust indices with strong power and consistent patterns, suitable for vital sign analysis. The final set of satisfying indices, denoted as $\mathcal{S}_{\text{final}}$, is defined as the intersection of $\mathcal{S}_1$ and $\mathcal{S}_2$ as follows:

$$\mathcal{S}_{\text{final}} = \mathcal{S}_1 \cap \mathcal{S}_2. \tag{3.20}$$

3.3.4 Phase Computation

Once the final set of indices $\mathcal{S}_{\text{final}}$ has been determined, the phase of the reflected radar signal is computed at these positions. This phase is critical, as it encodes minute displacements of the chest wall caused by respiratory and cardiac activity [3, 13, 47].

For each index $(p, q) \in \mathcal{S}_{\text{final}}$ and each chirp $m = 1, \dots, M_{\text{FIFO}}$, the raw phase is extracted from the complex-valued RAM signal $\Pi_{p,q}^{(m)}$ using the two-argument arctangent function:

$$\phi_{p,q}^{(m)} = \text{atan2}\left(\Im(\Pi_{p,q}^{(m)}), \Re(\Pi_{p,q}^{(m)})\right), \tag{3.21}$$

Algorithm 2 Phase Unwrapping Algorithm

Input: Wrapped phase signal $\boldsymbol{\phi} = [\phi_1, \phi_2, \ldots, \phi_{M_{\text{FIFO}}}]$ with values in $[-\pi, \pi]$
Output: Unwrapped phase signal $\hat{\boldsymbol{\phi}} = [\hat{\phi}_1, \hat{\phi}_2, \ldots, \hat{\phi}_{M_{\text{FIFO}}}]$
Initialize $\hat{\phi}_1 \leftarrow \phi_1$
for $i = 2$ **to** M_{FIFO} **do** ▷ Loop through the phase signal
    Compute phase difference: $\Delta\phi_i \leftarrow \phi_i - \phi_{i-1}$
    if $\Delta\phi_i > \pi$ **then** ▷ Identify forward wraps
        Correct: $\Delta\hat{\phi}_i \leftarrow \Delta\phi_i - 2\pi$
    else if $\Delta\phi_i < -\pi$ **then** ▷ Identify backward wraps
        Correct: $\Delta\hat{\phi}_i \leftarrow \Delta\phi_i + 2\pi$
    else
        No correction needed: $\Delta\hat{\phi}_i \leftarrow \Delta\phi_i$
    end if
    Update unwrapped phase: $\hat{\phi}_i \leftarrow \hat{\phi}_{i-1} + \Delta\hat{\phi}_i$
end for
return $\hat{\boldsymbol{\phi}}$

This computation yields phase values in the range $(-\pi, \pi]$, which may contain discontinuities due to phase wrapping. This phenomenon is explained by the general model of FMCW radar phase:

$$\phi(\tau) = \frac{4\pi f_c \left(R_0 + \xi(\tau)\right)}{c}, \tag{3.22}$$

where:

- R_0: Initial static distance between the radar and the chest wall,
- $\xi(\tau)$: Time-varying displacement due to physiological motion,

As $\xi(\tau)$ changes over time, the resulting phase $\phi(\tau)$ may cross the $(-\pi, \pi]$ boundary, causing abrupt phase jumps. To address this, a phase **phase unwrapping** is applied:

$$\hat{\phi}_{p,q} = \text{unwrap}\left(\boldsymbol{\phi}_{p,q}\right), \tag{3.23}$$

This process corrects discontinuities by detecting phase jumps greater than π and compensating with $\pm 2\pi$. The complete procedure is described in Algorithm 2, which outlines the logic for unwrapping the phase sequence within the FIFO buffer.

The resulting unwrapped values are organized into a two-dimensional matrix $\hat{\boldsymbol{\Phi}}$. In this matrix, each column corresponds to a spatial index $(p, q) \in \mathcal{S}_{\text{final}}$, while each row represents a chirp. As shown in Fig. 3.14, the resulting phase signals display a pseudo-sinusoidal pattern that is characteristic of physiological rhythms.

3.3.5 Application of Principle Component Analysis

The next step in our processing chain is to apply Principal Component Analysis (PCA) similar to [49] to the unwrapped phase array $\hat{\boldsymbol{\Phi}}$ to derive the so-called chest-wall displacement signal. PCA is a statistical technique that decomposes a large dataset

into a set of orthogonal components, allowing us to identify the most significant information from the original input data.

Foundations of PCA

PCA functions by identifying the axis that accounts for the maximum variance within the dataset, known as the first principal component (PC). Following the identification of the first PC, PCA then seeks a second axis that is orthogonal to the first, capturing the largest amount of the remaining variance. In datasets with more than two dimensions, PCA can identify additional principal components, each orthogonal to its predecessors. This process continues until the number of principal components equals the number of dimensions in the dataset.

Fig. 3.15 presents a simple 2D dataset along with two distinct axes, which can also be viewed as one-dimensional hyperplanes. The lower portion of the figure displays the projections of the dataset onto these axes. The projection onto the dotted line, where the vector PC1 lies, captures the maximum variance from the dataset. This is illustrated in Fig. 3.15b, which shows a wider spread of data points. The dotted line represents the first PC.

The dashed line, where the vector PC2 is located, is the only axis in this 2D example that is orthogonal to the first PC. The projection onto this line retains significantly less variance, as shown in Fig. 3.15c. This axis represents the second PC and captures much less information than the first.

It is generally advisable to choose the axis that preserves the most variance, as this axis is likely to retain more information compared to other projections. Additionally, this choice is justified by considering the mean squared distance: the first PC minimizes the distance between the original dataset and its projection onto the axis [50].

PCA commonly employs factorization methods such as Singular Value Decomposition (SVD) to identify principal components within a dataset. Given a data matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, SVD decomposes it into three distinct matrices [51, 52]:

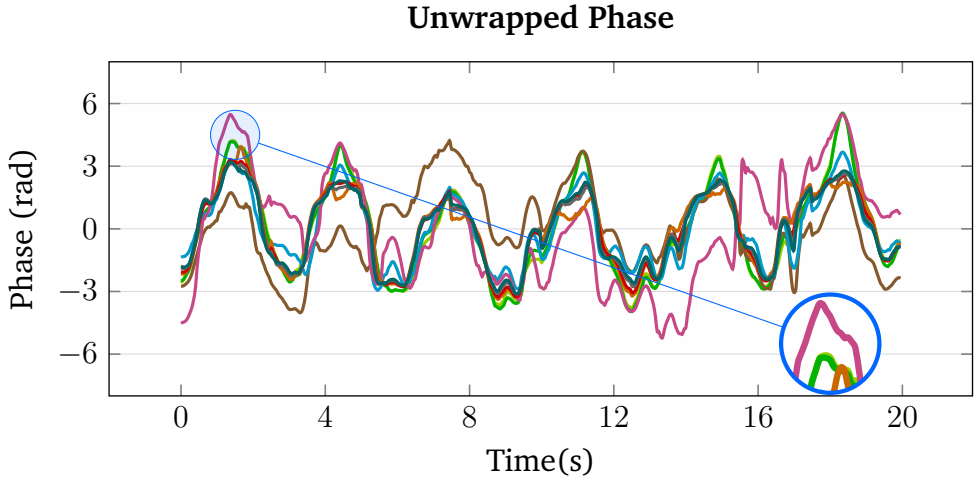
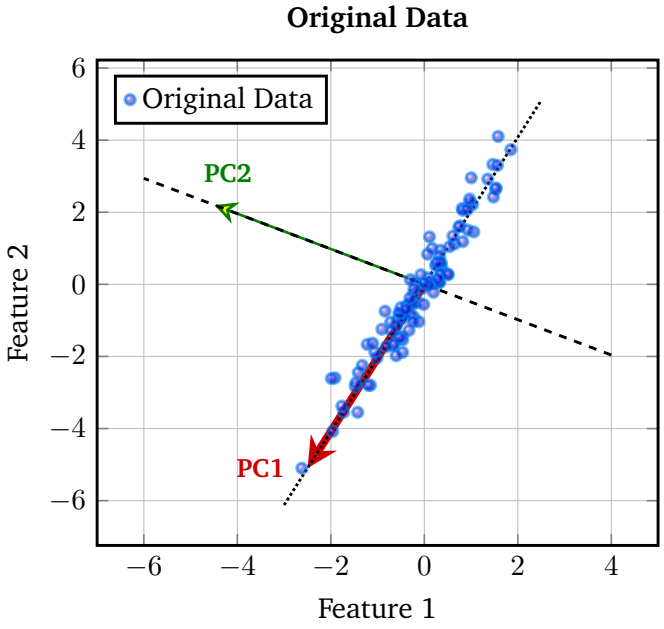
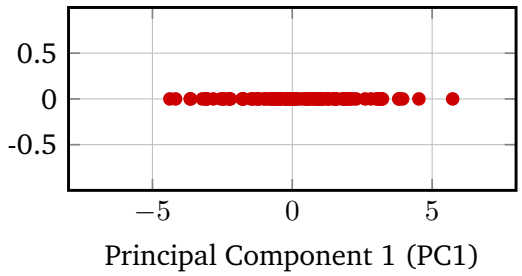


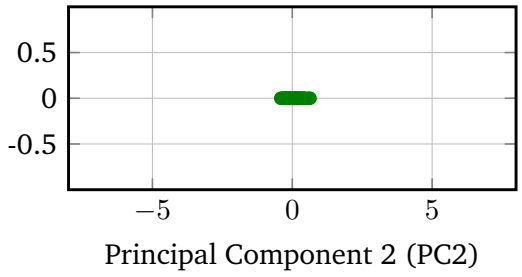
Fig. 3.14: A set of unwrapped phase values across all $M_{\text{FIFO}} = 400$ chirps of the FIFO buffer, computed for indices in $\mathcal{S}_{\text{final}}$.



(a)



(b)



(c)

Fig. 3.15: Visualization of PCA concept: (a) a 2D dataset with two axes representing principal components; (b) projection onto the first principal component (dotted line), capturing the maximum variance; (c) projection onto the second principal component (dashed line), retaining significantly less variance.

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{3.24}$$

where:

1. $\mathbf{U} \in \mathbb{R}^{m \times m}$ contains the left singular vectors, representing the principal directions in the row space of $\mathbf{A}$.
2. $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ is a diagonal matrix with non-negative singular values, indicating the relative importance of each principal component.
3. $\mathbf{V} \in \mathbb{R}^{n \times n}$ contains the right singular vectors; its transpose $\mathbf{V}^T$ captures the principal directions in the column space of $\mathbf{A}$.

Singular Values and Variance

The singular values in $\mathbf{\Sigma}$ are crucial in PCA because they indicate how much of the dataset's variance is explained by each principal component. If the singular values are ordered as follows:

$$\sigma_1 \geq \sigma_2 \ldots \geq \sigma_k, \tag{3.25}$$

σ_1^2 represents the variance captured by the first PC, while σ_2^2 indicates the variance explained by the second, and so on. By selecting the first k singular values, we choose the components that capture the most variance.

Projection Matrix

To project the dataset onto a lower-dimensional hyperplane with d dimensions, we extract the first d columns from $\mathbf{V} \in \mathbb{R}^{n \times n}$. The resulting projection matrix is defined as:

$$\mathbf{W} = \mathbf{V}[:, 0{:}d] \tag{3.26}$$

This projection matrix $\mathbf{W} \in \mathbb{R}^{n \times d}$ is used to create a reduced dataset by capturing the first d principal components.

To obtain the reduced dataset $\mathbf{A}_{\text{reduced}} \in \mathbb{R}^{m \times d}$, we multiply the original dataset $\mathbf{A}$ by the projection matrix $\mathbf{W}$ [50]:

$$\mathbf{A}_{\text{reduced}} = \mathbf{AW} \tag{3.27}$$

This matrix multiplication projects the dataset onto a lower-dimensional hyperplane defined by the first d principal components, resulting in a compact representation with reduced dimensionality. The reduced dataset $\mathbf{A}_{\text{reduced}}$ preserves the most informative components of the original data. This approach is widely used in machine learning and data analysis to reduce complexity, improve computational efficiency, and emphasize relevant information [53].

3.3.6 Enhancing PCA Analysis with Spatial Information

Previous studies, such as [49], have primarily focused on applying PCA to phase data derived from a single index. Although this approach provides useful insights, it overlooks the valuable spatial information that can be gained from analyzing neighboring indices. Focusing exclusively on one index may lead to missed broader

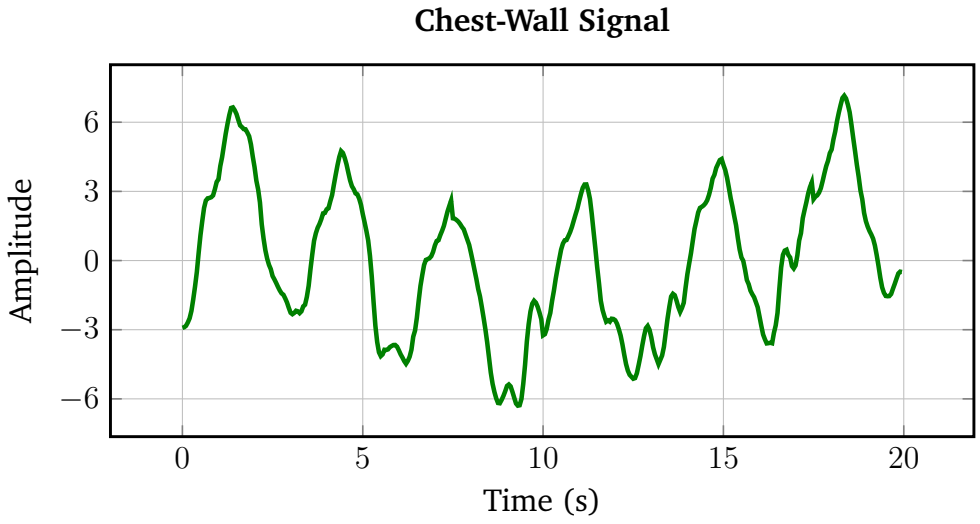


Fig. 3.16: Sample chest-wall displacement signal derived by scaling the combined phase signal from unwrapped phase data $\hat{\mathbf{\Phi}}$ across $M_{\text{FIFO}} = 400$ chirps.

spatial patterns, resulting in an incomplete understanding of unwrapped phase data $\hat{\mathbf{\Phi}}$.

Our research addresses this limitation by integrating spatial relationships into the PCA process, using phase data from the indices within $\mathcal{S}_{\text{final}}$. This captures richer features, enhancing the accuracy and robustness of respiratory and heart signal extraction across the chirps stored in FIFO buffer. The number of components was set to retain 95% of the variance, effectively capturing the majority of information while minimizing computational overhead.

Subsequently, a weighted combination of these principal components was used, with the explained variance ratio serving as the weighting factor, to extract a signal referred to as the "combined phase signal." This signal represents the primary features in the unwrapped phase data. According to Equation 3.22, the chest-wall signal is derived by scaling the "combined phase signal" by a constant factor ($\frac{c}{4\pi f_c}$). This scaling demonstrates how the unwrapped phase data $\hat{\mathbf{\Phi}}$ contributes to the generation of the chest-wall signal. A sample chest-wall signal is depicted in Fig. 3.16.

Fig. 3.17 presents the explained and cumulative variance for the combined phase signal. The red dashed line signifies the point at which 95% of the total variance is retained. Each bar represents the explained variance of an individual component, emphasizing its contribution to the overall data variability. The line plot illustrates the cumulative variance, showing how the total variance is progressively taken into account as more components are included in the analysis. This figure highlights the role of the principal components in capturing the core characteristics of spatially distributed phase data collected during a processing interval[1]. This interval comprises $M_{\text{FIFO}} = 400$ chirps.

This method significantly improves the accuracy and power of the extracted chest-wall signal. Moreover, it avoids the computational intensity associated with earlier studies that applied PCA to the output of EEMD, a process that can considerably slow

[1]The processing interval refers to the set of chirps stored in the FIFO buffer.

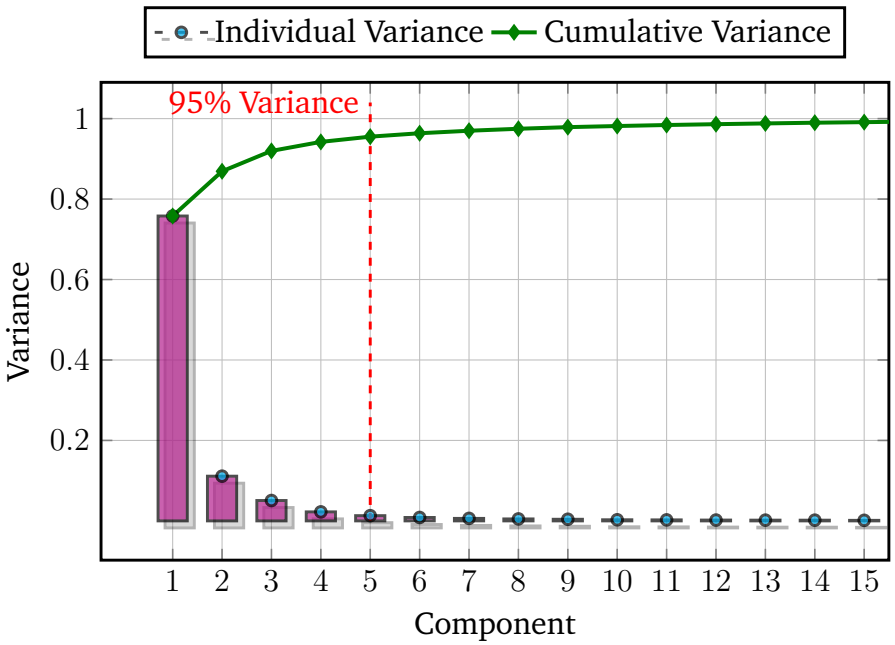


Fig. 3.17: The explained variance and cumulative variance for a combined phase signal. The red dashed line marks the point where 95% of the variance is retained.

down the data processing pipeline.

3.3.7 Extracting Breathing and Heart Signals from Chest-Wall Data

To extract vital signals from the chest-wall data, the signal is decomposed into two primary components: the breathing signal and the heart signal. This separation is achieved by applying two distinct Infinite Impulse Response (IIR) band-pass filters to the chest-wall signal. The key idea is that although these signals are intertwined in the time domain, their distinct intrinsic frequencies allow for effective separation in the frequency domain. Specifically, the BR typically falls within the range of 0.1 to 0.4 Hz, while the HR varies between 0.9 and 2.5 Hz.

The chest-wall signal, denoted as $x(t)$, can be expressed as a combination of the breathing signal $b(t)$ and the heart signal $h(t)$:

$$x(t) = b(t) + h(t) + n(t) \tag{3.28}$$

where $n(t)$ represents noise. The goal is to isolate $b(t)$ and $h(t)$ from $x(t)$.

Breathing Signal Isolation

To isolate the breathing signal, we apply a band-pass filter to $x(t)$, designed to pass frequencies in the range of 0.1 to 0.4 Hz while attenuating frequencies outside this range. The filtered signal $b_f(t)$ primarily contains the breathing component. Mathematically, the band-pass filtering operation can be expressed as:

$$b_f(t) = \mathcal{F}^{-1}\left\{H_{\text{bp}}(f) \cdot \mathcal{F}\{x(t)\}\right\} \tag{3.29}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier and inverse Fourier transforms, respectively, and $H_{\text{bp}}(f)$ is the frequency response of the band-pass filter used to isolate the breathing signal.

The next step is to analyze this filtered signal to determine the BR. This analysis involves computing the Power Spectral Density (PSD) of the breathing signal using the periodogram method. The PSD provides a representation of the power of each frequency component present in the signal, which is mathematically defined as:

$$P_{bb}(f) = \frac{1}{T}\left|\int_0^T b_f(t)e^{-j2\pi ft}dt\right|^2 \tag{3.30}$$

where T is the total observation time. In the context of our radar application, T is equivalent to $M_{\text{FIFO}} = 400$. The frequency component $\hat{f}_b$ with the highest amplitude in the PSD, $P_{bb}(f)$, corresponds to the BR:

$$\hat{f}_b = \underset{f\in[0.1,0.4]}{\arg\max}\, P_{bb}(f) \tag{3.31}$$

A sample breathing signal and its corresponding frequency response are shown in Fig. 3.18. Subfigure (a) illustrates the time-domain representation of the breathing signal after applying the band-pass filter. Subfigure (b) shows the frequency response of the filtered breathing signal, with the peak indicating the estimated breathing rate.

Heart Signal Isolation

To isolate the heart signal, we apply a band-pass filter $H_{hf}(f)$ to $x(t)$, designed to pass frequencies in the range of 0.9 to 2.5 Hz while attenuating frequencies outside this range. The filtered signal $h_f(t)$ primarily contains the heart component. Mathematically, this operation can be expressed as:

$$h_f(t) = \mathcal{F}^{-1}\left\{H_{hf}(f)\cdot\mathcal{F}\{x(t)\}\right\} \tag{3.32}$$

where $H_{hf}(f)$ is the frequency response of the band-pass filter for the heart signal.

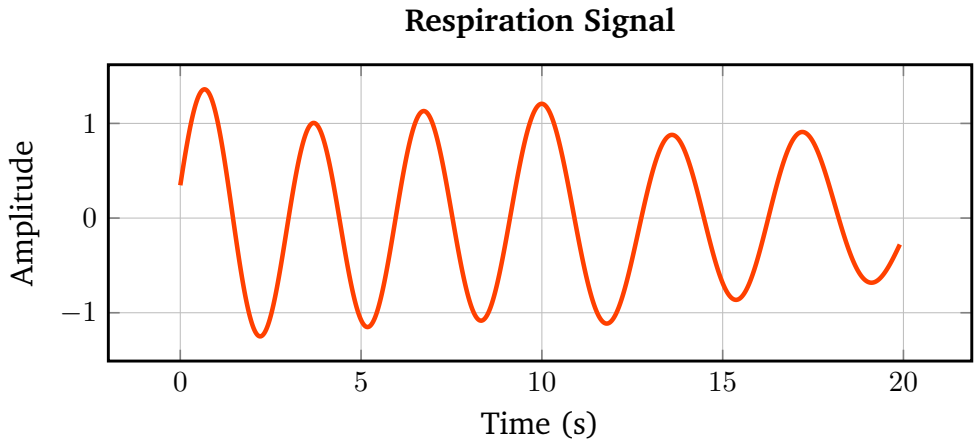
Next, to further enhance the heart signal, we introduce a comb filter. This filter suppresses the harmonics of the breathing signal that overlap with the frequency spectrum of the heart signal. This step is crucial because the strong harmonics of the breathing signal can introduce artifacts, potentially distorting the true frequency components of the heart signal in the frequency domain. Consequently, failing to mitigate these harmonics could lead to inaccuracies in the HR estimation, compromising the reliability and validity of vital sign monitoring outcomes.

To achieve this, a series of second-order IIR notch digital filters are employed. The transfer function of each notch filter $H_k(s)$ is represented as:

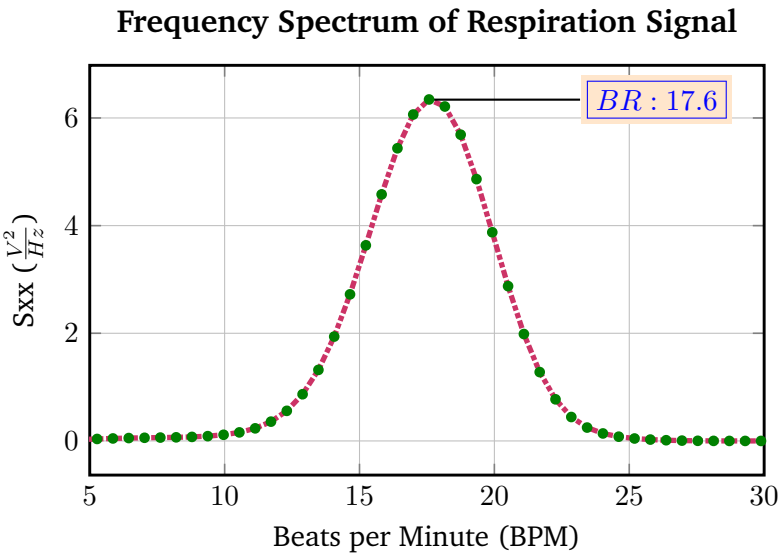
$$H_k(s) = \frac{s^2+\omega_{0k}^2}{s^2+\frac{\omega_{0k}}{Q_k}s+\omega_{0k}^2} \tag{3.33}$$

where:

- $\omega_{0k} = 2\pi f_{0k}$ is the angular frequency of the k-th notch.
- Q_k is the quality factor of the k-th notch filter.



(a)



(b)

Fig. 3.18: (a) Time-domain representation of the filtered breathing signal, showing the amplitude variations over time; (b) Frequency spectrum of the filtered breathing signal, displaying the PSD. The peak in the PSD indicates the estimated BR, marked at approximately 17.6 BPM.

- s is the complex frequency variable in the Laplace domain.

For practical implementation, the parameters are chosen as follows:

- **Notch Frequencies (f_{0k})**: These are the harmonics of the BR that fall within the 0.9 to 2.5 Hz range.
- **Quality Factor (Q_k)**: A value of 100 is chosen to ensure narrow notches, effectively attenuating the harmonic frequencies.

The overall frequency response of the comb filter, denoted as $H_{\text{comb}}(f)$, is achieved by multiplying the individual transfer functions of all notch filters:

$$H_{\text{comb}}(f) = \prod_{k=1}^{K} H_k(f) \tag{3.34}$$

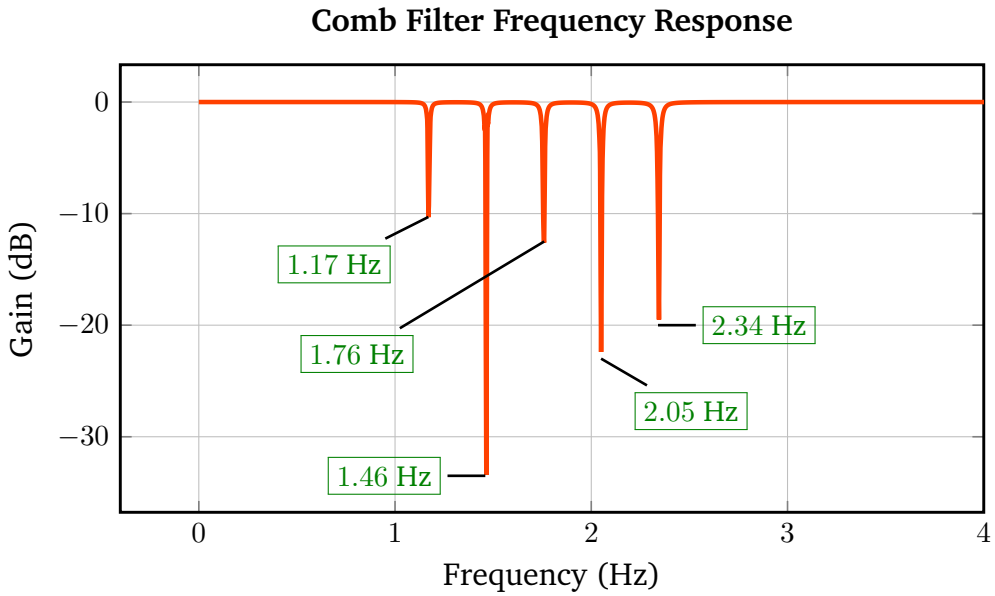


Fig. 3.19: Frequency response of the comb filter designed to attenuate harmonics of the breathing signal, based on a breathing rate of 17.6 BPM. The pins in the figure represent the first few harmonics of the BR, which fall within the range of 0.9 to 2.5 Hz.

Here, K represents the number of harmonics targeted for attenuation.

Considering a BR of 17.6 BPM($\hat{f}_b = 17.6/60 \approx 0.29$ Hz), we can calculate the corresponding notch frequencies f_{0k} as integer multiples of $\hat{f}_b$ within the specified range. The calculated frequencies are summarized in Table 3.2. The frequency response of the comb filter corresponding to this BR is illustrated in Fig. 3.19.

The final heart signal $h_{ff}(t)$ is then obtained by applying the comb filter to the output of the heart band-pass filter $h_f(t)$:

$$h_{ff}(t) = \mathcal{F}^{-1}\left\{H_{\text{comb}}(f) \cdot \mathcal{F}\{h_f(t)\}\right\} \tag{3.35}$$

The time-domain representation of a sample heart signal is depicted in Fig. 3.20, illustrating the filtered heart waveform following the application of the comb filter.

Subsequently, we compute the PSD of $h_{ff}(t)$. The frequency $\hat{f}_h$ corresponding to the highest peak in this PSD, denoted as $P_{hh}(f)$ is used to estimate the HR:

$$\hat{f}_h = \underset{f \in [0.9, 2.5]}{\arg\max}\, P_{hh}(f) \tag{3.36}$$

Tab. 3.2: Notch Frequencies for BR of 17.6 BPM

Harmonic (k)	Frequency (f_{0k}) (Hz)	Angular Frequency (ω_{0k}) (rad/s)
4	1.17	$2\pi \times 1.17$
5	1.46	$2\pi \times 1.46$
6	1.76	$2\pi \times 1.76$
7	2.05	$2\pi \times 2.05$
8	2.34	$2\pi \times 2.34$

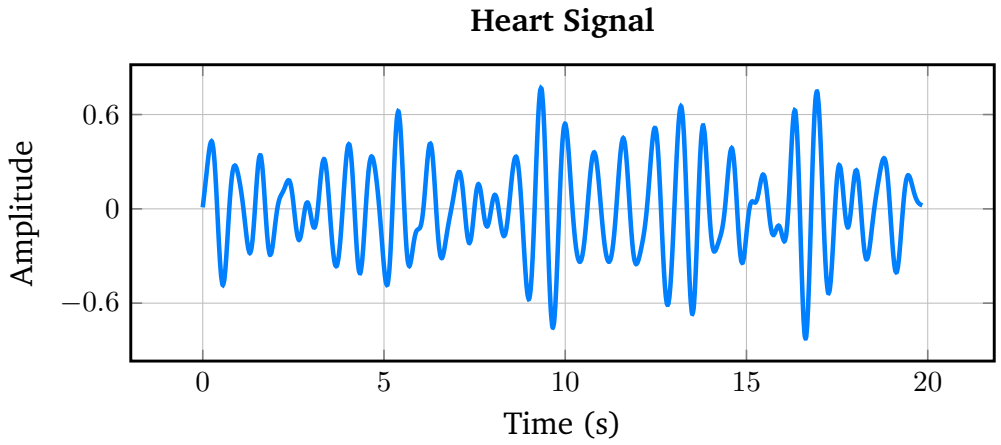


Fig. 3.20: Time-domain representation of the heart signal after applying the band-pass and comb filters.

This methodology ensures that the HR estimation remains both unaffected and undistorted by respiratory artifacts, thereby leading to more accurate and reliable vital sign monitoring.

Comparative Analysis of Heart Rate Estimation

To illustrate the effectiveness of our method, we compare the PSD of a test subject's heart signal across two consecutive processing intervals using two different approaches:

- The spectrum was computed using only the unwrapped phase at the peak index $\hat{\phi}^{(m)}_{p_c,q_c}$ for $(m = 1, \ldots, M_{\text{FIFO}})$.
- spectrum was computed through applying PCA to the unwrapped phase array $\hat{\mathbf{\Phi}}$.

Fig. 3.21 and 3.22 present the results of the comparative analysis. In the first processing interval, the single-point approach fails to accurately estimate the HR due to the presence of two dominant peaks with similar amplitudes in the PSD. This ambiguity makes it difficult for the algorithm to identify the correct peak, resulting in an incorrect estimation. While this method may yield correct results under certain conditions, it lacks consistent reliability.

In contrast, the PCA-based approach consistently provides accurate HR estimations across both processing intervals by analyzing a data slice around the peak. Its robustness stems from the ability to incorporate a broader set of data points, enabling it to effectively handle signal ambiguities and variations.

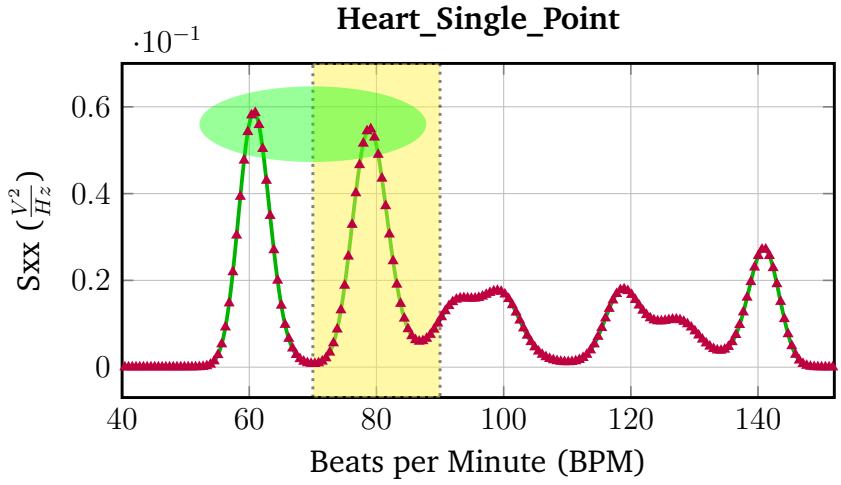
3.4 Empirical Investigation of Vital Sign Monitoring

This section describes the experimental setup, which is carefully designed to simulate a realistic automotive in-cabin environment. The aim is to evaluate the performance

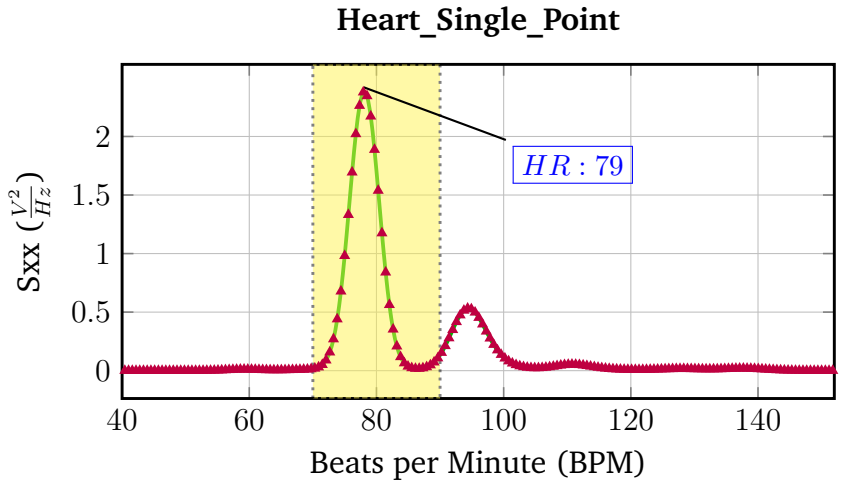
of the FMCW radar in detecting and monitoring a participant's vital signs, such as BR and HR, under conditions similar to those inside a vehicle.

The participant is positioned at three different angles θ relative to the radar: $-10°$, $0°$, and $+10°$. These angles are chosen to reflect the typical positions of occupants within a car, such as the driver and front passenger seats. This setup allows us to assess how the radar performs when the subject is oriented at different angles, simulating the natural movements and orientations of passengers in a vehicle.

In addition to varying the angle θ, the distance r between the participant and the radar is systematically adjusted to test the radar's range performance within a vehicle. Measurements are taken at three specific distances: 0.5 m, 0.75 m, and 1.0 m. These distances are representative of the close-range scenarios typical in a car cabin, where occupants are in close proximity to the radar sensors embedded in the vehicle's

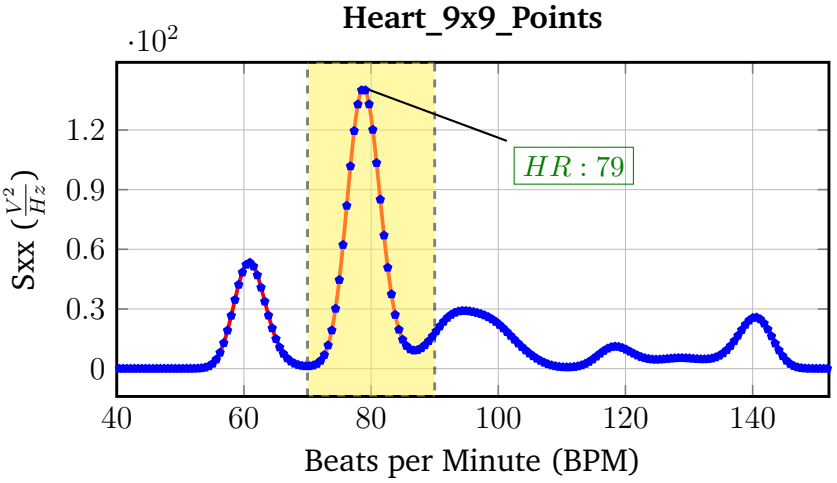


(a)

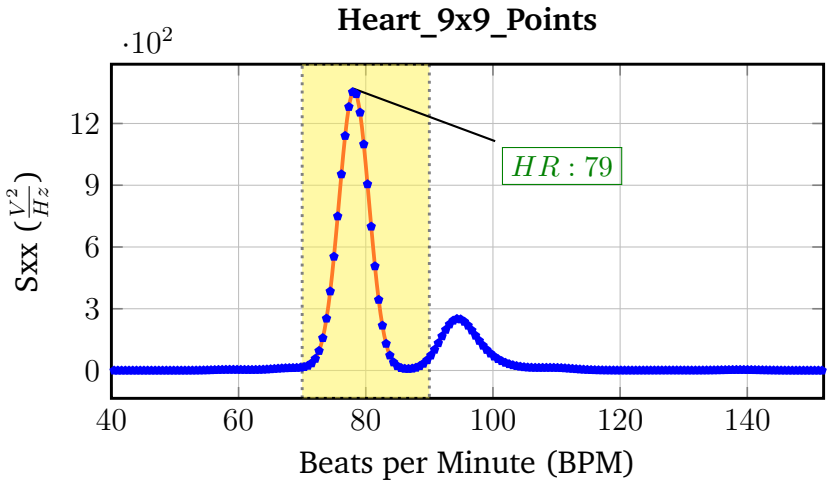


(b)

Fig. 3.21: HR estimation using the single-point method across two processing intervals: (a) PSD during the first interval; (b) PSD during the second interval. The single-point method initially fails to accurately estimate the HR in the first processing interval due to the presence of two dominant peaks. However, it effectively estimates it in the second interval.



(a)



(b)

Fig. 3.22: HR estimation using PCA across two processing intervals: (a) PSD computed using PCA in the first interval; (b) corresponding PSD in the second interval. The PCA-based approach consistently yields accurate estimations of HR across both processing intervals.

dashboard or interior panels.

The schematic of this empirical setup is illustrated in Fig. 3.23. The red circles indicate the various positions of the test subject relative to the radar, showcasing our systematic approach to evaluating the radar's performance across the spatial parameters of interest.

We conducted simultaneous measurements using the EQ02+ LifeMonitor chest strap sensor, manufactured by Equivital [54], to ensure the accuracy and reliability of our radar data. As depicted in Fig. 3.24, the LifeMonitor comprises two fundamental elements: the sensor belt and Sensor Electronics Module (SEM). Within the sensor belt, discreet low-profile sensors are seamlessly integrated into a biocompatible fabric. Meanwhile, the SEM securely fits into a cradle on the left-hand side of the sensor belt, managing the processing, storage, and transmission of the collected physiological data. The EQ02+ LifeMonitor records various physiological parameters,

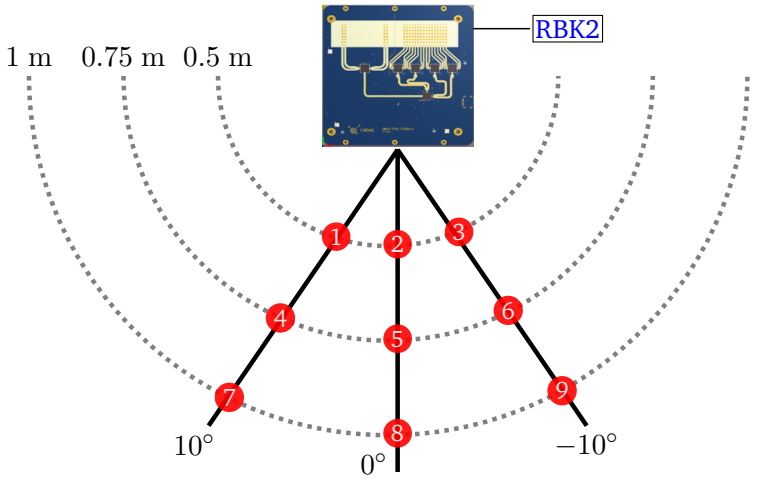


Fig. 3.23: Positions of the test subject for different combinations of range (r) and angle (θ), with red circles representing the test subject's locations



Fig. 3.24: EQ02+ LifeMonitor chest strap sensor by Equivital, depicting the sensor belt and the SEM for processing physiological data[54].

which can later be retrieved for analysis using the provided computer software. This additional data collection step enables us to cross-validate the radar data against precise ground-truth measurements.

3.4.1 Procedure and Key Metrics

This section outlines the experimental procedure and key metrics used to evaluate the performance of the FMCW radar system in monitoring vital signs.

Experimental Setup and Results

During each experimental repetition, the FMCW radar system captures and processes data in real-time, as outlined in Algorithm 3. Data is stored in `.csv` files to facilitate post-processing with various analysis tools.

Fig. 3.25 and Fig. 3.26 present temporal diagrams comparing BR and HR from three different methods: the proposed radar-based method, conventional radar-based methods, and the EQ02+ LifeMonitor sensor. These comparisons were conducted at a distance of $r = 0.75$ m and an angle of $\theta = 0°$, with each experiment repeated 150 times over 5 s intervals. The results show that the proposed method closely

Algorithm 3 Experimental Procedure

Input: Set of angles $\Theta = \{-10°, 0°, +10°\}$, set of distances $R = \{0.5 \text{ m}, 0.75 \text{ m}, 1.0 \text{ m}\}$
for each $\theta \in \Theta$ **do**
  for each $r \in R$ **do**
    Position the participant at angle θ and distance r from the radar
    Capture radar signal $x(t)$
    Process signal to estimate BR and HR
    Store estimated rates in .csv file
    Simultaneously capture ground-truth data using EQ02+ LifeMonitor
  end for
end for

tracks sensor measurements, indicating high consistency in capturing physiological parameters. In contrast, the conventional methods, which derive phase data from peak-index, exhibits notable inconsistencies, particularly in HR measurement.

In Fig. 3.27 and Fig. 3.28, scatter plots offer an additional perspective by directly comparing the proposed radar-based method with the ground-truth values for BR and HR, respectively. Each point in these plots represents a pair of measurements–one from the radar-based method and the other from the EQ02+ LifeMonitor sensor–facilitating a direct assessment of their agreement. The clustering of data points around the diagonal line ($x = y$) in both figures indicates a strong correlation between the BR and HR measurements obtained from our radar-based method and those from the EQ02+ LifeMonitor sensor. This finding underscores the accuracy and reliability of the proposed method in tracking these vital signs.

3.4.2 Analysis of Metrics

Before presenting specific results, it is important to define the metrics used for evaluation.

1. **Pearson Correlation Coefficient (ρ):** This metric measures the linear relationship between radar-derived vital signs (F_i) and ground-truth data (A_i). It is computed using Equation 3.37, which is defined as:

$$\rho = \frac{\sum_{i=1}^{n}(F_i - \bar{F})(A_i - \bar{A})}{\sqrt{\sum_{i=1}^{n}(F_i - \bar{F})^2 \sum_{i=1}^{n}(A_i - \bar{A})^2}} \tag{3.37}$$

   where F_i and A_i represent the radar-derived and ground-truth values, respectively, $\bar{F}$ and $\bar{A}$ are their respective means, and n is the number of observations.

2. **Mean Absolute Percentage Error (MAPE):** This metric calculates the average percentage error between radar-derived (F_i) and ground-truth (A_i) vital signs. It provides insight into the accuracy of estimation across different experimental conditions. The calculation for MAPE is expressed as:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{F_i - A_i}{A_i}\right| \times 100\% \tag{3.38}$$

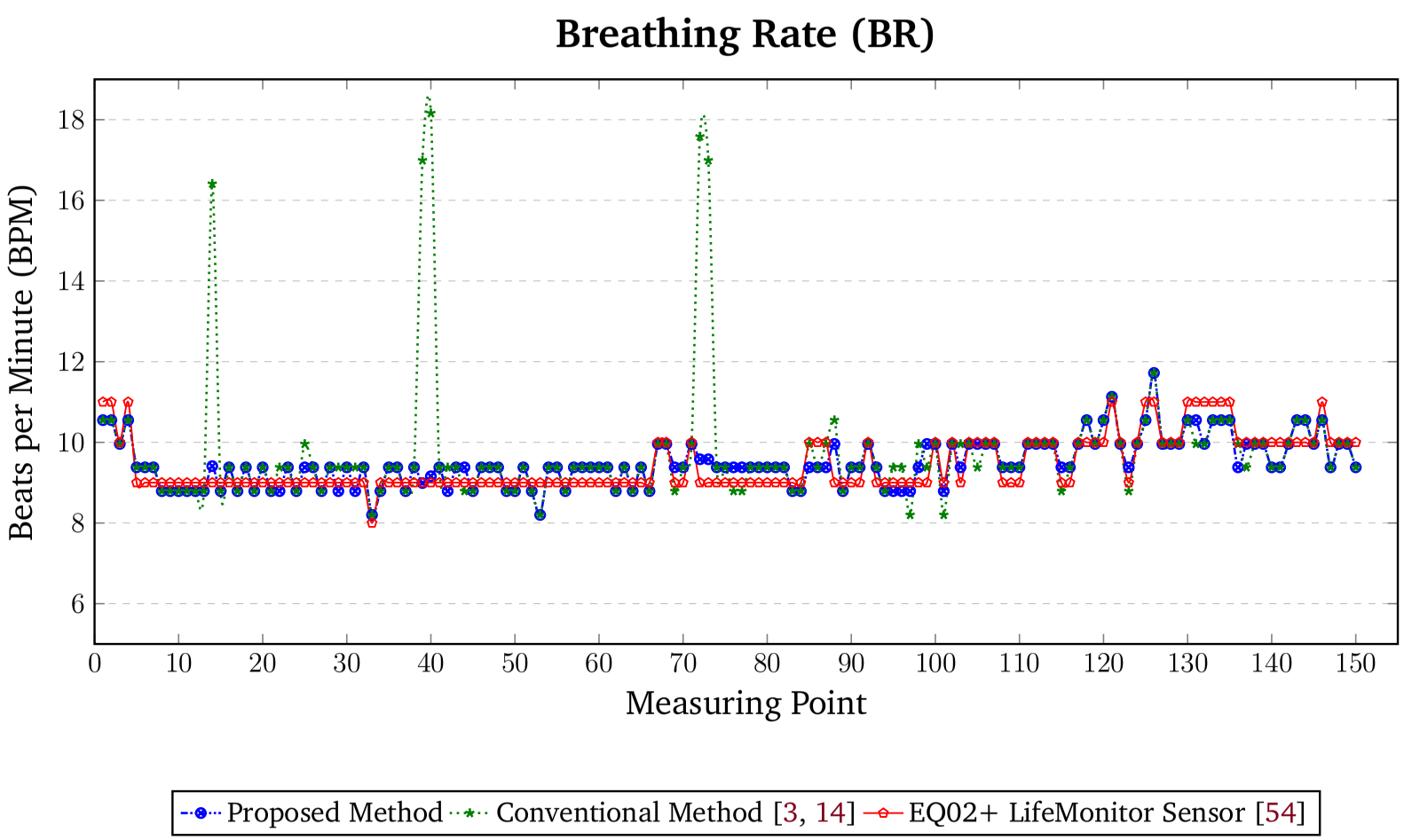


Fig. 3.25: Comparison of BRs obtained from the RBK2 and the EQ02+ LifeMonitor sensor measurements at a range of $r = 0.75$ m and an angle of $\theta = 0°$. The performance of the proposed method is highlighted against the methods presented in [3, 14].

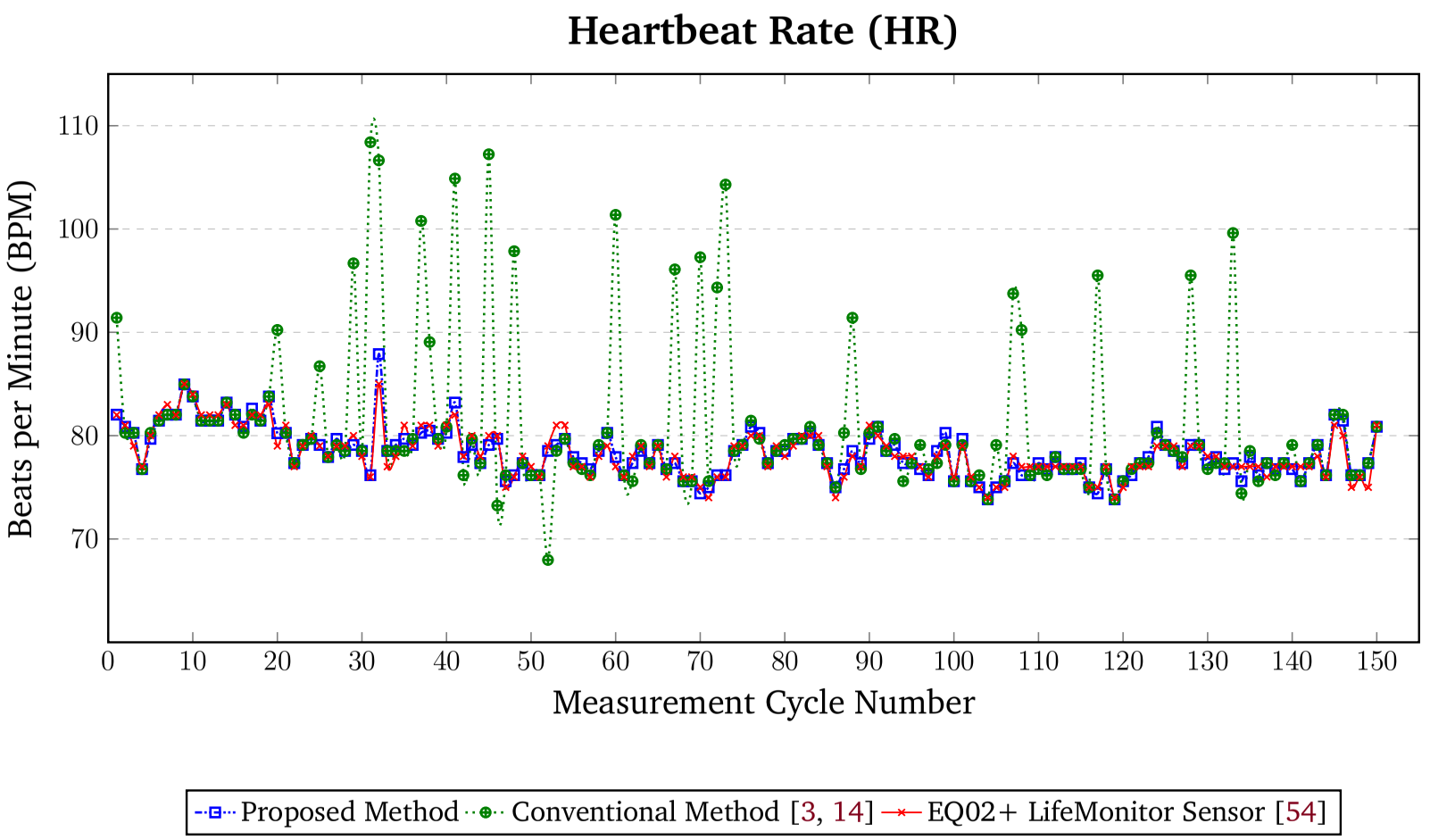


Fig. 3.26: Comparison of HRs obtained from the RBK2 and the EQ02+ LifeMonitor sensor measurements at a range of $r = 0.75$ m and an angle of $\theta = 0°$. The performance of the proposed method is highlighted against the methods presented in [3, 14].

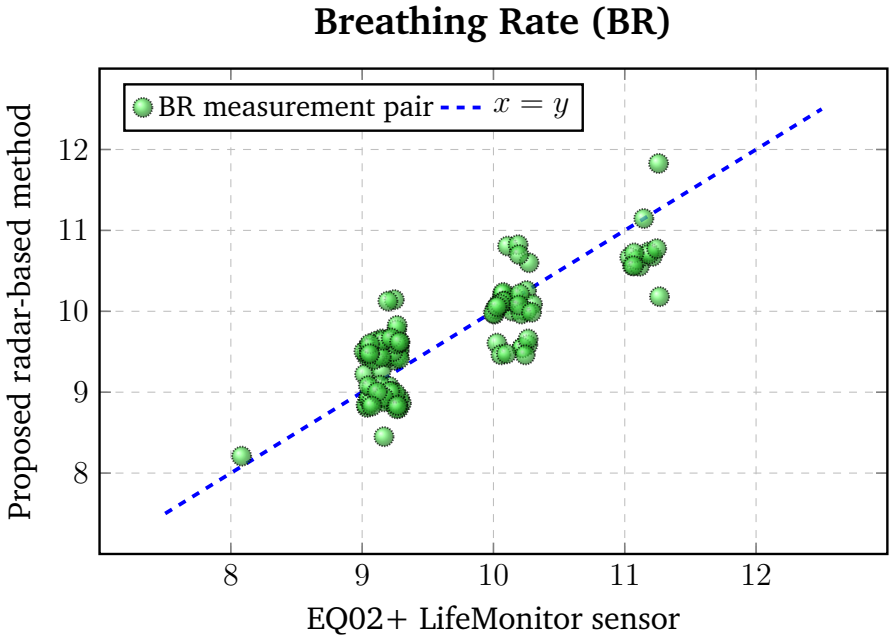


Fig. 3.27: Comparison of BR measurements between our radar-based proposed method and EQ02+ LifeMonitor sensor at $r = 0.75$ m and $\theta = 0°$. Each marker represents a data point from the experiment, with jitter added for better visualization. The diagonal line indicates perfect agreement ($x = y$).

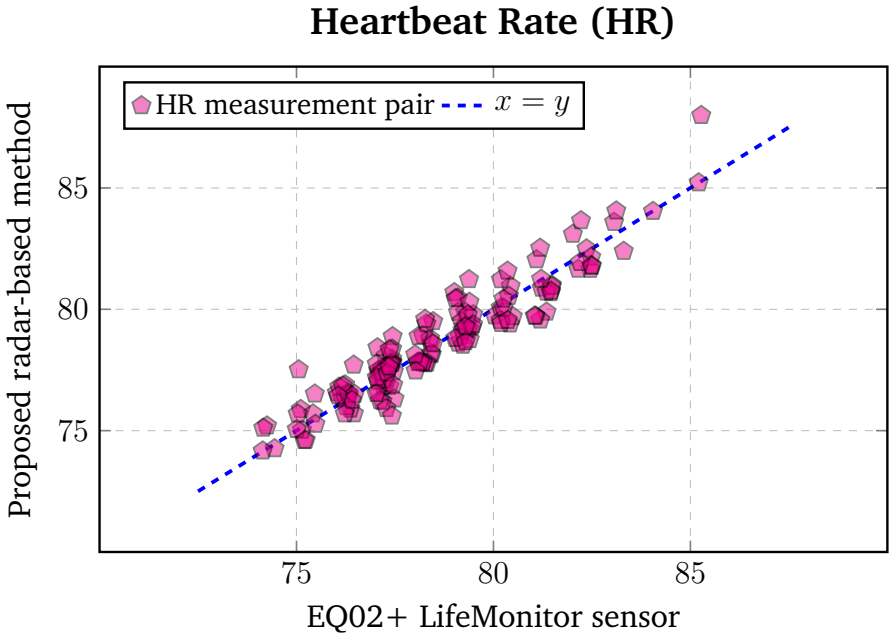


Fig. 3.28: Comparison of HR measurements between our radar-based proposed method and EQ02+ LifeMonitor sensor at $r = 0.75$ m and $\theta = 0°$. Each marker represents a data point from the experiment, with jitter added for better visualization. The diagonal line indicates perfect agreement ($x = y$).

The MAPE and Pearson correlation coefficient (PCC) values for BR and HR estimation, derived from the results presented in Fig. 3.25 and Fig. 3.26, are summarized in Table 3.3. The proposed method significantly reduced estimation error compared

Tab. 3.3: Comparison between the proposed and conventional methods [3, 14] in terms of MAPE, PCC (ρ), and statistical significance using Fisher's z-test for HR and BR. These results are associated with the data presented in Fig. 3.26 and Fig. 3.25.

Category	Metric	HR	BR
MAPE (%)	Proposed	**0.76**	**3.31**
	Conventional [3, 14]	4.50	6.44
	Improvement	-83.1%	-48.6%
Pearson ρ	Proposed	**0.947**	**0.833**
	Conventional [3, 14]	0.257	0.201
	Difference	$+0.690$	$+0.632$
Fisher z-test	z-score	13.17	8.52
	p-value	< 0.00001	< 0.00001

to conventional techniques [3, 14], with MAPE improvements of 83.1% for HR and 48.6% for BR.

Furthermore, the correlation with ground-truth measurements increased markedly. For HR, the PCC improved from 0.257 to 0.947, and for BR from 0.201 to 0.833. To statistically validate these improvements in correlation, we employed Fisher's z-test. Since PCC are not normally distributed, especially near their extreme values, direct comparisons are statistically unreliable. Fisher's z-transformation addresses this limitation by converting each correlation coefficient r into a normally distributed variable z, enabling formal hypothesis testing. The transformation is given by [55]:

$$z = \frac{1}{2} \ln \left(\frac{1+r}{1-r} \right) \tag{3.39}$$

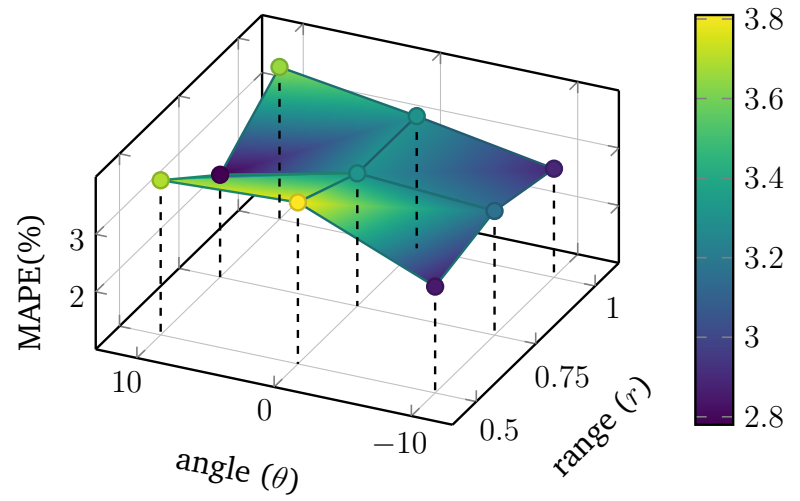
The significance of the difference between two transformed values z_1 and z_2 is then evaluated via:

$$z_{\text{score}} = \frac{z_1 - z_2}{\sqrt{\frac{1}{n_1 - 3} + \frac{1}{n_2 - 3}}} \tag{3.40}$$

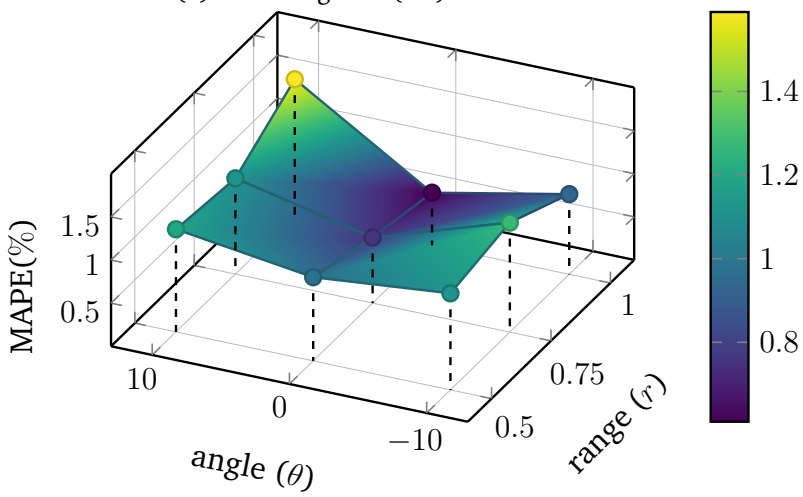
where n_1 and n_2 denote the sample sizes. In our case, both HR and BR comparisons yielded highly significant z-scores (13.17 and 8.52, respectively) with p-values below 0.00001, confirming the statistical superiority of the proposed method. These results demonstrate improved accuracy and stronger agreement with ground-truth mesurements, derived form the EQ02+ LifeMonitor sensor, underscoring the effectiveness of our proposed approach.

3.4.3 Data Analysis

The MAPE values for each participant's position are depicted in Fig. 3.29, corresponding to the layout shown in Fig. 3.23. At each position, physiological parameters were captured 150 times, and MAPE was calculated based on these measurements. Additionally, the MAPE values have been listed in Table 3.4 to facilitate detailed



(a) Breathing rate (BR) measurements



(b) Heartbeat rate (HR) measurements

Fig. 3.29: 3D visualization of MAPE calculated across all examined combinations of range and angle (r, θ), depicting the accuracy of vital sign measurements.

Tab. 3.4: MAPE for different angles and ranges associated with Fig. 3.29

Range (r)	**Angle (θ)**	**MAPE for HR (%)**	**MAPE for BR (%)**
0.5 m	$-10°$	1.12	2.85
0.75 m	$-10°$	1.27	3.16
1.0 m	$-10°$	0.93	2.89
0.5 m	$0°$	0.97	3.81
0.75 m	$0°$	0.76	3.31
1.0 m	$0°$	0.61	3.30
0.5 m	$+10°$	1.19	3.69
0.75 m	$+10°$	1.11	2.78
1 m	$+10°$	1.59	3.65
	Average	1.06	3.27

analysis.

The visualization in Fig. 3.29 reveals that deviations are generally more pronounced at oblique angles (e.g., $\theta = \pm 10^{\circ}$) compared to the frontal orientation ($\theta = 0^{\circ}$), which aligns with the expectation of stronger and more direct radar reflections at lower angles. Distance also plays a critical role; while the system maintains stable performance across the 0.5–$1.0\ \mathrm{m}$ range, certain combinations, such as $+10^{\circ}$ at $1.0\ \mathrm{m}$ exhibit higher error, particularly for HR estimation. These local variations may be attributed to reduced signal return or minor involuntary participant movements.

Despite these factors, the overall system performance remains robust. As summarized in Table 3.4, the average MAPE across all configurations is 1.06% for HR and 3.27% for BR, confirming the system's ability to deliver consistent estimation accuracy across a broad range of practical deployment scenarios.

3.5 Conclusion

The FMCW radar-based system, utilizing the RBK2, delivers precise and non-intrusive monitoring of vital signs. It achieved an average MAPE of 1.06% for HR and 3.27% for BR across various spatial configurations when compared to the EQ02+ LifeMonitor, even in the presence of participant motion and environmental challenges.

By applying PCA to a spatial window of unwrapped phase data centered on the target's strongest signal within the captured chirp signals, the system effectively extracts vital sign patterns. This approach outperforms single-point phase analysis, which yielded MAPE values of 4.50% for HR and 6.44% for BR in one of the test scenarios. When combined with harmonic suppression technique to isolate the heart signal from respiratory components, this method ensures reliable estimation of vital signs, particularly at closer ranges and smaller angles. These findings highlight the system's potential as a wearable-free solution for various applications, including healthcare and automotive environments.

CHAPTER 4

Literature Review on Hand Gesture Recognition

Contents

4.1 Introduction **58**
4.2 Overview of FMCW HGR Pipeline **58**
4.3 Data Representations in FMCW Gesture Recognition . **58**
4.4 Radar-Deriven Hand Gesture Recognition Techniques **59**
4.4.1 Evolution of Feature Extraction Methods . . 60
4.4.2 Comparative Analysis of Classification Techniques 61
4.4.3 Multi-Hand and Multi-Target Gesture Recognition Challenges 62
4.4.4 Advanced Applications of Gesture Recognition 63
4.5 Challenges and Research Gaps **63**
4.5.1 Emerging Trends and Future Directions . . . 64
4.6 Conclusion **64**

4.1 Introduction

Hand Gesture Recognition (HGR) has emerged as a transformative technology in HCI, enabling intuitive, contactless interfaces for a wide range of applications, including smart devices, automotive systems, healthcare, and assistive technologies. FMCW radar systems have become a preferred modality in this domain due to their ability to provide high-resolution spatial and temporal data, their robustness to environmental factors such as lighting and occlusion, and their privacy-preserving nature compared to vision-based approaches [10, 11, 56]. This chapter systematically reviews the state-of-the-art in HGR using FMCW radar, covering data representations, feature extraction, classification techniques, multi-hand gesture recognition, advanced applications, and emerging challenges. The review aims to identify key advancements, methodologies, and gaps, providing a foundation for proposing novel contributions in this field.

4.2 Overview of FMCW HGR Pipeline

A typical FMCW radar-based HGR system follows a multi-stage pipeline, as illustrated in Fig. 4.1. The process begins with gesture collection, where an FMCW radar captures raw radar signals reflected from hand movements. These signals are then processed to generate spectra, such as RDMs or RAMs, which represent the spatial and temporal characteristics of the gestures, as illustrated in Fig. 4.2. Deep learning models, often combining Convolutional Neural Network (CNN) and Recurrent Neural Networks (RNNs), are applied to extract features and model temporal dependencies. Finally, the system outputs a recognition result, classifying the gesture into a predefined category.

4.3 Data Representations in FMCW Gesture Recognition

FMCW radar systems provide rich data representations that capture the spatial and temporal dynamics of hand gestures. The most common representation is the RDM, which maps the range and velocity of targets. Choi et al. [57] utilized a 60 GHz FMCW radar to generate RDM sequences, applying clutter removal to isolate gesture signatures. Suh et al. [58, 59] constructed RDMs using a 24 GHz FMCW radar to recognize seven distinct gestures. RDMs are particularly effective for capturing the radial motion of gestures, such as push or pull, but they lack angular information, which can be critical for distinguishing gestures with similar radial profiles.

To address this limitation, RAMs have been explored to incorporate angular data, enhancing spatial context. Yu et al. [60] proposed a two-stream network integrating RDMs and RAMs from IWR1443BOOST mm-Wave radar to capture both radial and transversal gesture features [61]. This multi-dimensional approach has shown promise in improving recognition accuracy, particularly for gestures with significant angular variations, such as pointing or circular motions.

Micro-Doppler signatures, derived via Short-Time Fourier Transform (STFT), offer another representation for capturing fine hand movements. Kim and Toomajian [62]

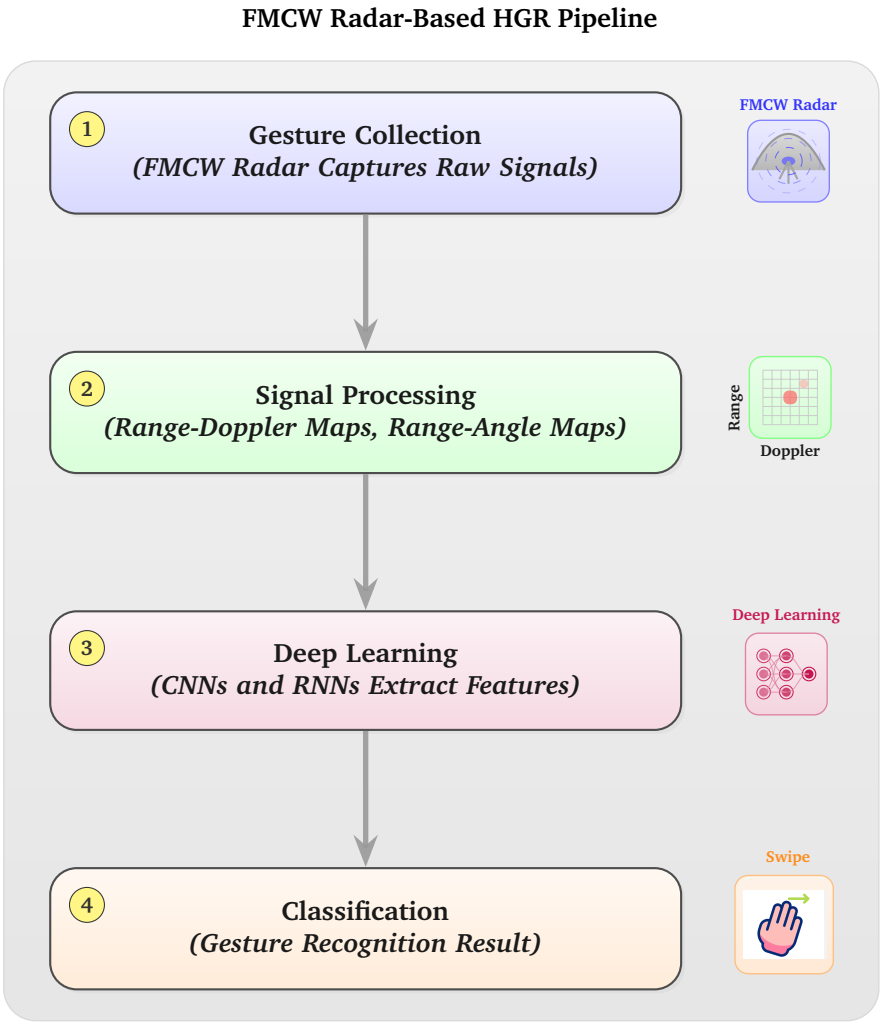


Fig. 4.1: A multi-stage pipeline for FMCW radar-based HGR, featuring gesture collection, signal processing, deep learning, and classification stages.

demonstrated the utility of micro-Doppler spectrograms, achieving an accuracy of 85.6% with a convolutional neural network. Micro-Doppler signatures are particularly useful for detecting subtle movements, such as finger flicks, but they require high temporal resolution, which can increase computational complexity.

Recent studies have explored additional representations, such as RSM and ASM, to capture the temporal evolution of gestures. Ahmed et al. [63] utilized a multi-stream CNN with RSMs, ASMs, and Doppler-time spectrograms for air writing recognition, highlighting the potential of combining multiple temporal representations. However, these representations often require careful synchronization and preprocessing to align the data streams effectively.

4.4 Radar-Deriven Hand Gesture Recognition Techniques

This section explores the evolution of feature extraction and classification methods in FMCW radar-based HGR, alongside advanced applications and challenges in multi-hand and multi-target scenarios.

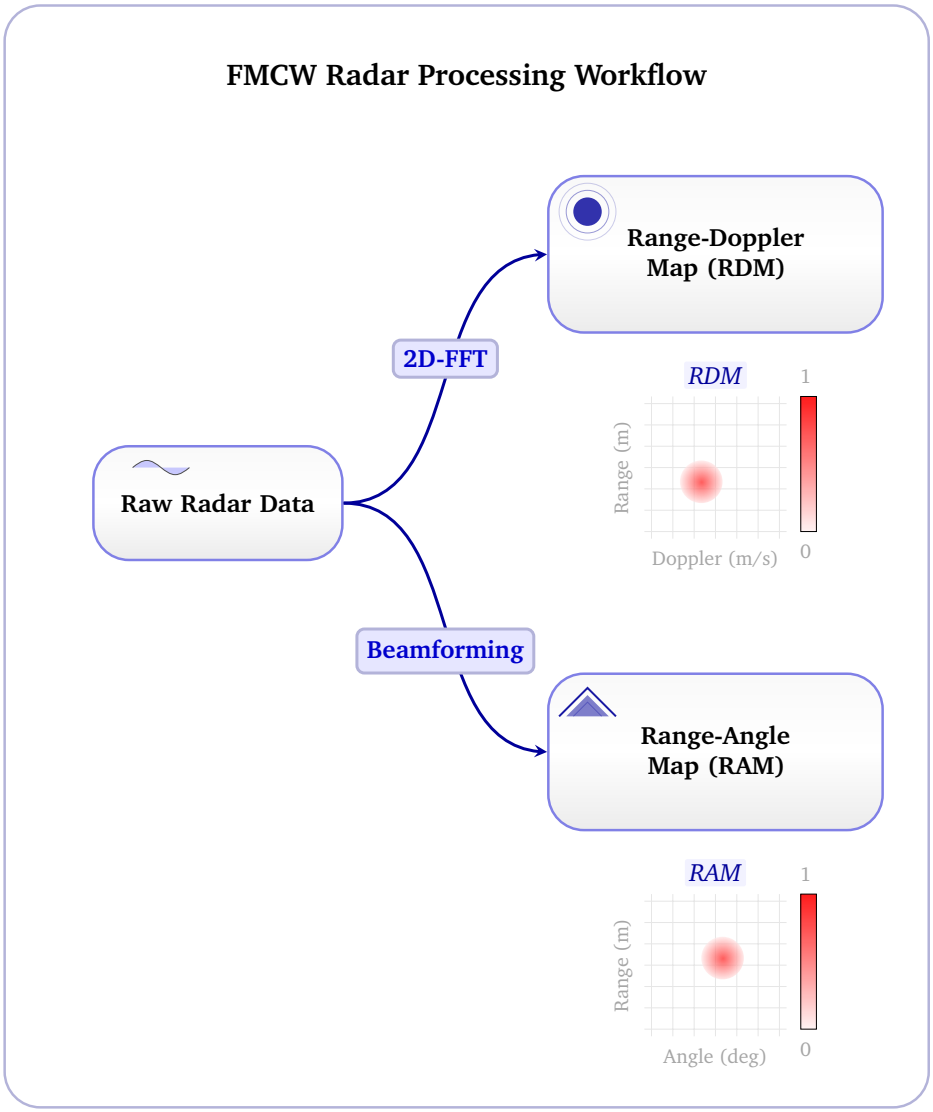


Fig. 4.2: Conceptual workflow of FMCW radar data processing for HGR. Raw radar data is transformed into a RDM using 2D-FFT and a RAM via beamforming.

4.4.1 Evolution of Feature Extraction Methods

Feature extraction in HGR using mm-Wave radar has evolved significantly, transitioning from handcrafted methods to deep learning approaches. Early studies relied on manual feature engineering. Malysa et al. [64] extracted low-dimensional motion features, such as velocity profiles and range trajectories, from a 77 GHz radar and applied Hidden Markov Models (HMMs) for classification, targeting HCI applications. While effective for simple gestures, these methods struggled with complex or dynamic gestures due to their limited expressive power.

The advent of deep learning has revolutionized feature extraction and classification. CNNs have become a cornerstone due to their ability to automatically extract spatial features from radar maps. Skaria et al. [65] utilized a deep CNN with a 24 GHz radar, processing micro-Doppler signatures to achieve robust gesture classification with an accuracy of 95% for a set of 14 gestures. The ability of CNN to learn hierarchical features directly from raw data representations, such as RDMs,

eliminates the need for manual feature engineering, making it highly adaptable to diverse gesture sets.

Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) units, have been employed to model temporal dependencies in gesture sequences. Choi et al. [57] combined an LSTM encoder with RDM sequences from a 60 GHz radar, achieving real-time HGR with an accuracy of 98.5% for a set of 10 gestures. Yang et al. [66] integrated a three-layer LSTM network with the IWR1443 radar, reporting a 97% accuracy for nine gestures.

Hybrid architectures combining CNNs and RNNs have gained prominence for their ability to capture both spatial and temporal features. Zhang et al. [67] proposed Latern, a recurrent 3D CNN with a Connectionist Temporal Classification (CTC) algorithm for continuous gesture recognition using a 24 GHz radar, achieving a 96% accuracy. This approach exemplifies the trend toward multi-dimensional feature fusion, where spatial features extracted by CNNs are fed into RNNs to model temporal dynamics.

Transfer learning has also been explored to address the challenge of limited radar data. Alnujaim et al. [68] applied transfer learning using a pre-trained VGG-16 model [69], originally intended for image classification tasks, on spectrogram images derived from antenna impedance. By fine-tuning the model for gesture classification, they achieved an accuracy of 94.6%. This approach demonstrates the potential of leveraging pre-trained models to reduce the need for large radar datasets, which are often labor-intensive to collect.

4.4.2 Comparative Analysis of Classification Techniques

Table 4.1 summarizes key studies on classification techniques for FMCW radar-based HGR, highlighting radar frequency, data representation, classification method, number of gestures, and reported accuracy. Below, we discuss the strengths, limitations, and emerging trends in these techniques to provide a deeper understanding of their applicability.

Traditional methods, such as HMMs used by Malysa et al. [64], rely on manually extracted motion features. While computationally lightweight, HMMs are limited by their dependence on predefined feature sets, resulting in lower accuracy (82.3% for four gestures) compared to deep learning methods. As a result, this approach is less effective for complex gestures, particularly those involving subtle spatial or temporal patterns.

Deep learning techniques, especially CNNs and LSTM networks, have significantly improved classification performance. Skaria et al. [65] achieved an accuracy of 95% using a CNN by leveraging two receiving antennas. They computed both micro-Doppler spectrograms and a time-frequency Angle-of-Arrival (AoA) map, which served as an additional input channel to the CNN, enhancing spatial resolution and demonstrating the power of automatic feature extraction. However, CNNs are primarily suited for spatial feature extraction and may require additional architectures to model temporal dynamics effectively.

In contrast LSTM-base methods are designed to model temporal dependencies in radar gesture sequences. Choi et al. [57] achieved a high accuracy of 99.1%

on a 10-gesture set using a 60 GHz Soli radar and motion profiles as input to their LSTM encoder. In a related study, Suh et al. [58] employed a 24 GHz radar system and used sequential projected RDM features as input to an LSTM network, achieving over 91% accuracy for seven gestures. While both systems leveraged LSTM's strength in temporal modeling, Choi's system benefited from richer input data and broader participant variation. Moreover, lower performance in Suh's work may stem from challenges related to out-of-plane hand movements and single-plane antenna constraints, which affect spatiotemporal representation quality.

Hybrid architectures, such as the 3D CNN with CTC proposed by Zhang et al. [67], combine spatial and temporal modeling, achieving a high accuracy of 96% for seven gestures. This approach is particularly effective for continuous gesture recognition, where gestures occur in sequences without clear boundaries, eliminating the need for manual gesture boundary detection. However, generalization to unseen users required additional training data. In cross-user evaluations, the model's performance declined notably–from 93.5% to 72.5%–when tested on an unseen participant. This highlights a key limitation.

Transfer learning, as demonstrated by Alnujaim et al. [68], offers a promising solution to the challenge of limited radar datasets. However, their approach has notable constraints. The system was evaluated on only three participants, raising concerns about generalization to a broader population. Furthermore, the need for network analyzer hardware and impedance measurements makes real-time deployment more complex compared to radar-based approaches.

Emerging trends in HGR techniques include the integration of multi-modal data representations (e.g., combining RDMs with micro-Doppler signatures) and the use of advanced architectures like attention mechanisms to focus on relevant features. Another important consideration is the choice of radar frequency, which has a direct impact on spatial and velocity resolution. Higher frequencies, e.g. 77 GHz, offer finer resolution and enable the detection of smaller or faster movements. However, they also pose a challenge: radar systems with higher frequencies often require a more precise calibration of the hardware.

4.4.3 Multi-Hand and Multi-Target Gesture Recognition Challenges

While most radar-based gesture recognition studies focus on single-hand gestures, recognizing complex gesture scenarios–such as simultaneous multi-hand gestures from one user or gestures from multiple targets-remains challenging and underexplored. Wang et al. [70] tackled simultaneous multi-hand gesture recognition using a 77 GHz FMCW radar [71], proposing a spatiotemporal path selection algorithm to separate overlapping signals from both hands in RDMs and RAMs, as shown in Fig. 4.3). Their dual 3D-CNN-based feature fusion network achieved a 93.12% accuracy for eight simultaneous multi-hand gestures, demonstrating the potential of multi-dimensional parameter utilization, though it struggles with range-overlapping or Doppler-similar gestures.

In a related context, Zhang et al. [72] explored gesture recognition involving two targets using a mm-Wave radar. They used a gesture separation network with dynamic RAMs and a CNN + LSTM model to classify gestures from two distinct sources. Their

system achieved an average accuracy of 93% for double-target scenarios across varied environments, addressing challenges such as environmental adaptability. However, it focuses on independent targets rather than simultaneous multi-hand gestures from a single user. These studies highlight distinct but complementary approaches to handling multiple gesture signals in radar-based systems.

4.4.4 Advanced Applications of Gesture Recognition

Beyond basic gesture recognition, FMCW radar-based HGR has been applied to several advanced use cases, such as air writing recognition, where users write characters or words in the air. Park et al. [73] explored air writing with a 60 GHz FMCW radar. They extracted 3D hand trajectories and converted them into 2D images, which were used to train a DenseNet modle for classifying 26 alphabet characters. Their approach achieved a classification accuracy of 91% for character recognition, which proves the potential of radar-based systems for fine-grained tasks. Air writing poses unique challenges, such as tracking continuous trajectories and distinguishing between similar characters (e.g., 'O' vs. 'Q'), which require high-resolution data and robust temporal modeling.

Few-shot learning has emerged as a solution to the challenge of data scarcity in radar-based HGR. Shen et al. [74] introduced ML-HGR-Net, a meta-learning network for few-shot HGR with a 77 GHz radar. Using a 3D CNN dual-channel fusion, their model adapts to unseen gestures with limited data, achieving an accuracy of 98.1% for a five-gesture set with only one or five training samples per gesture per episode. This approach is particularly valuable for personalized gesture recognition, where users may define custom gestures with minimal training data.

4.5 Challenges and Research Gaps

Despite significant progress, several challenges persist in HGR using radar systems. Environmental variability, including clutter and interference, complicates gesture isolation. Zhao et al. [75] noted that background reflections (e.g., from walls or furniture) can obscure gesture signatures, particularly in indoor settings. Techniques such as background noise suppression, Static Interference Suppression (SIS) and Dynamic Interference Suppression (DIS) have been proposed to suppress unwanted reflections and class-like interference. In these methods, static targets (e.g. walls) are removed based on zero Doppler shift and dynamic targets (e.g. arm movements) are filtered out based on Doppler velocity differences. Iida et al. [76] proposes a range-aware localization method, which leverages chest reflections to isolate hand signals, but they are not universally effective in highly cluttered environments.

The reliance on single-plane antenna arrays limits the detection of out-of-plane movements. Ryu et al. [59] reported a reduced accuracy for gestures with significant vertical motion, as their 24 GHz radar lacked sufficient angular resolution. MIMO radar systems, which provide better angular resolution, have been explored as a solution, but they increase hardware complexity and cost.

multi-hand and multi-target gesture recognition remains constrained by signal overlap and insufficient feature differentiation. Wang et al. [70] highlighted the

difficulty of separating gestures with similar Doppler profiles, such as two hands moving in the same direction. Advanced signal separation techniques, such as Independent Component Analysis (ICA) and EEMD, or deep clustering, may offer solutions but have not been widely applied in this context.

Data scarcity is a significant barrier, as radar data collection is labor-intensive and time-consuming. Few-shot learning approaches, such as those proposed by Shen et al. [74], mitigate this issue. In addition, synthetic data generation including Generative Adversarial Networks (GANs), has been explored as an alternative. For instance, Zhao et al. [77] used a GAN to generate synthetic RAMs for training, improving model robustness and achieving an accuracy ecceeding 94% for a set of nine gestures.

Computational efficiency is another concern, particularly for real-time applications on resource-constrained devices. Traditional RNNs, such as LSTMs, which are inherently sequential in nature, underutilize parallel computing capabilities compared to convolutional alternatives, leading to higher latency. Lightweight models, such as MobileNet or EfficientNet, have been explored in vision-based HGR but are underexplored in radar-based systems [78].

4.5.1 Emerging Trends and Future Directions

Several emerging trends are shaping the future of FMCW radar-based HGR. One trend is the integration of multi-modal sensing, combining radar with other modalities like Inertial Measurement Units (IMUs) or cameras [79]. This multi-modal approach improves robustness but introduces challenges in data fusion and synchronization.

Another trend is the use of Reinforcement Learning (RL) to optimize gesture recognition systems. RL can be used to dynamically adjust the model parameters or select optimal features based on environmental conditions [80]. This approach, in contrast to the supervised HGR studies covered in this chapter, does not require class labels and learns by receiving feedback in the form of rewards. However, its combination with FMCW radar for gesture recognition has not been adequately explored thus far.

4.6 Conclusion

FMCW radar-based HGR has advanced significantly, driven by sophisticated data representations, deep learning techniques, and innovative applications. RDMs, RAMs, and micro-Doppler signatures have enabled high accuracies, often exceeding 90%, for single-hand gestures in controlled settings. However, challenges such as multi-hand gesture separation, environmental robustness, data scarcity, and computational efficiency persist. Emerging trends, including multi-modal sensing, and RL offer promising directions for future research. This review provides a comprehensive foundation for proposing novel solutions to address these gaps, particularly in deploying advanced architectures to enhance feature extraction and classification capabilities.

Tab. 4.1: Comparison of classification techniques for FMCW radar-based HGR across studies

Study	Radar Frequency	Data Representation	Classification Method	No. of Gestures	Accuracy (%)
Malysa et al. [64]	77 GHz	Motion Features	HMM	5	82.3
Skaria et al. [65]	24 GHz	Micro-Doppler	CNN	14	95
Choi et al. [57]	60 GHz	RDM	LSTM	10	98.5
Suh et al. [58]	24 GHz	RDM	LSTM	7	91
Zhang et al. [67]	24 GHz	RSM	3D CNN + CTC	8	96.0
Alnujaim et al. [68]	2.4 GHz	STFT	Transfer Learning (VGG-16)	10	94.6

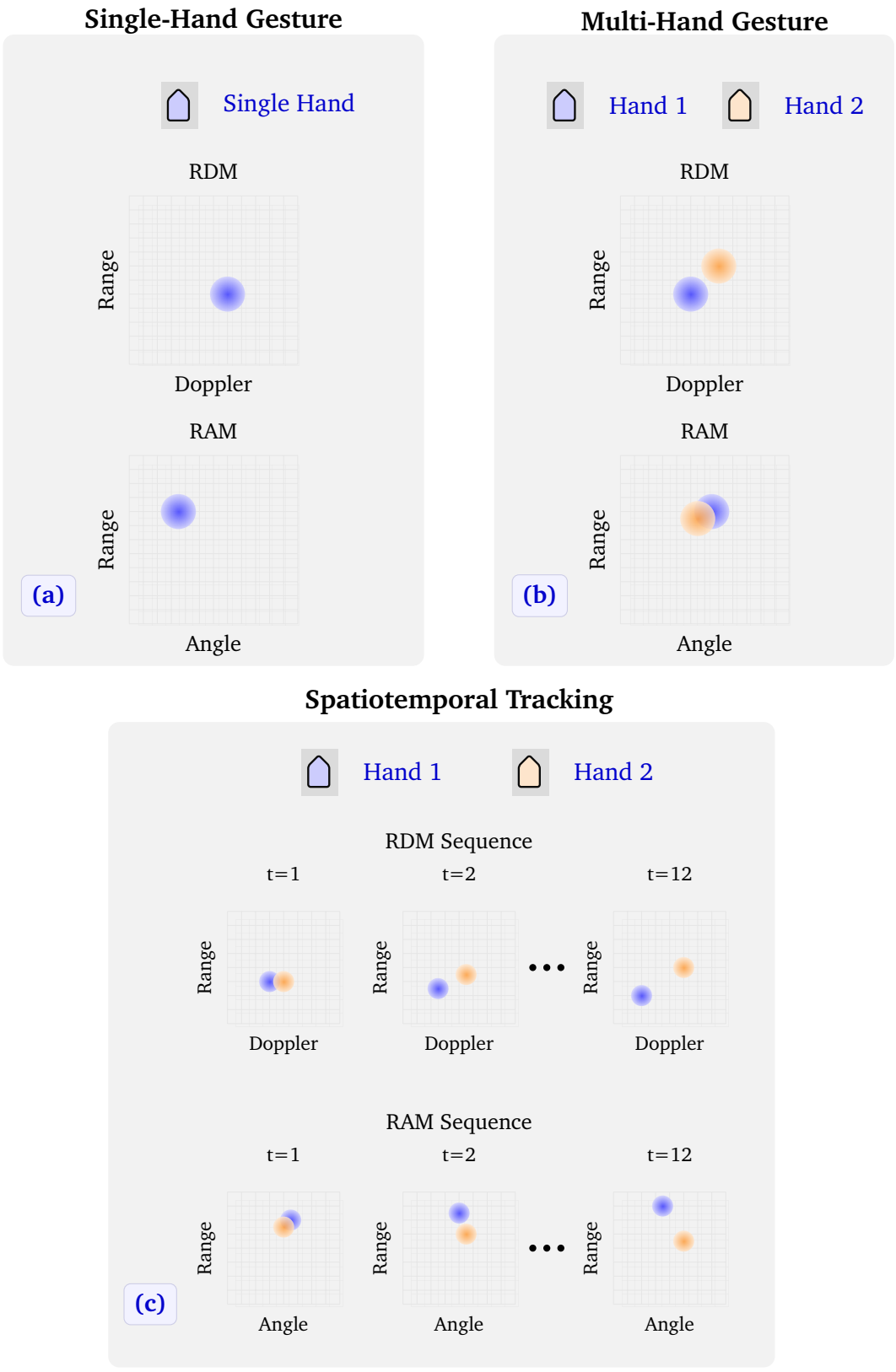


Fig. 4.3: Radar-based gesture recognition: (a) Single-hand gesture with corresponding RDM and RAM; (b) Multi-hand gesture; (c) Spatiotemporal tracking of two hands over time, demonstrating signal separation. The depicted RDM and RAM are illustrative examples, not derived from empirical data, and are intended to clarify the conceptual workflow.

CHAPTER 5

Hand Gesture Recognition

Contents

5.1 Introduction 69
5.2 Beat Signal Processing . . . 71
5.2.1 Creating Range-Doppler Maps . 71
5.2.2 Creating Range-Angle Maps 72
5.2.3 Range, Velocity, and Angle Bin Spacing . . 72
5.2.4 Enhancement of Range-Doppler and Range-Angle Maps . . 73
5.3 Hand Activity Detection Using Energy Comparison . 74
5.3.1 Initialization 75
5.3.2 Data Collection and Storage 75
5.3.3 Energy Calculation . . 75
5.3.4 Exponential Moving Average (EMA) 76
5.3.5 Dynamic Thresholds . 76
5.3.6 Hand Activity Detection 77
5.3.7 Incrementing Gesture Counters 77
5.3.8 Multi-Cycle Gesture Confirmation 77
5.3.9 Threshold Parameter Selection 80
5.4 Gesture Design and Acquisition 80
5.4.1 Normalization of Input Data 81
5.5 Proposed Deep Neural Network Model 82
5.5.1 Architectural Rationale 83
5.5.2 CNN-GRU Model Architecture 84
5.5.3 Necessity of Hyperparameter Tuning 86
5.6 Hyperparameter Optimization Strategy 86
5.6.1 Impact of Convolutional Layers on Spatial Feature Extraction 87
5.6.2 Effect of GRU Layers and GRU Units on Validation Accuracy . 88
5.6.3 Impact of Convolutional Layers and GRU Units 89

5.6.4 Effect of Dense Layers and Dense Units on Accuracy 89
5.6.5 Top-performing Hyperparameter Configurations 92
5.7 Validation of Model Generalization through LOOCV 95
5.7.1 Motivation for LOOCV and Comparative Analysis 95
5.7.2 Methodology 95
5.7.3 LOOCV Performance Results 96
5.7.4 Per-Class Performance Analysis 97
5.7.5 Subject-Wise Performance Analysis . 100
5.7.6 Training Dynamics and Generalization 100
5.7.7 Future Directions . . . 102
5.7.8 Conclusion 103

5.1 Introduction

Hand Gesture Recognition (HGR) is a branch of HCI that is gaining popularity in modern automotive control systems [81]. This technology enables a more intuitive and seamless interaction with the vehicle's infotainment system compared to the past, when drivers had to rely on pressing various buttons or touching the screen to perform basic tasks such as adjusting the radio volume or activating the air conditioner [81].

In recent years, advancements in HGR have significantly enhanced user experience by offering a more natural and efficient means of controlling these systems [81]. This evolution is especially important in the automotive industry, where minimizing distractions and maintaining focus on the road are crucial for safety [82]. Consequently, HGR technology is being integrated into a variety of vehicle models, from high-end luxury cars to more accessible mid-range vehicles, making it a standard feature in modern car design.

Beyond automotive applications, hand gesture recognition is also making significant strides in other emerging scientific areas, notably in Virtual Reality (VR) and Augmented Reality (AR) [9, 83]. These fields have experienced substantial advancements over the last decade, with hand gesture recognition playing a critical role in enhancing the immersive experience. In VR, users can interact with virtual environments more realistically, manipulating objects and navigating spaces with natural hand movements. Similarly, in AR, hand gesture recognition allows users to manipulate digital objects overlaid onto the real world, facilitating more interactive and engaging applications in fields such as gaming, and education.

Contactless hand gesture recognition methods are typically divided into two main categories: camera-based and radar-based techniques.

1. **Camera-Based Techniques:** Camera-based systems utilize visual data captured by cameras to interpret hand movements. These systems often employ complex algorithms to process the visual input and accurately recognize gestures. However, their performance can be adversely affected by ambient light conditions, such as low light or excessive glare, which can lead to inaccuracies in gesture recognition. Additionally, privacy concerns arise from the constant camera surveillance required for these systems to function effectively. The need for continuous video recording may make users uncomfortable, raising issues related to data security and privacy [56, 84].

2. **Radar-Based Techniques:** Radar-based techniques, particularly those using mm-Wave radars, offer a robust alternative to camera-based systems. These systems emit radio waves and analyze the reflected signals to detect and interpret hand movements. Radar-based systems are unaffected by varying illumination conditions, making them more reliable in diverse lighting environments. Furthermore, they provide a higher level of privacy, as they do not require capturing visual data. This characteristic makes radar-based techniques more appealing for users concerned about surveillance and data privacy.

Thanks to substantial progress in the semiconductor industry, mm-Wave radars can now be embedded in small and portable devices. This development makes

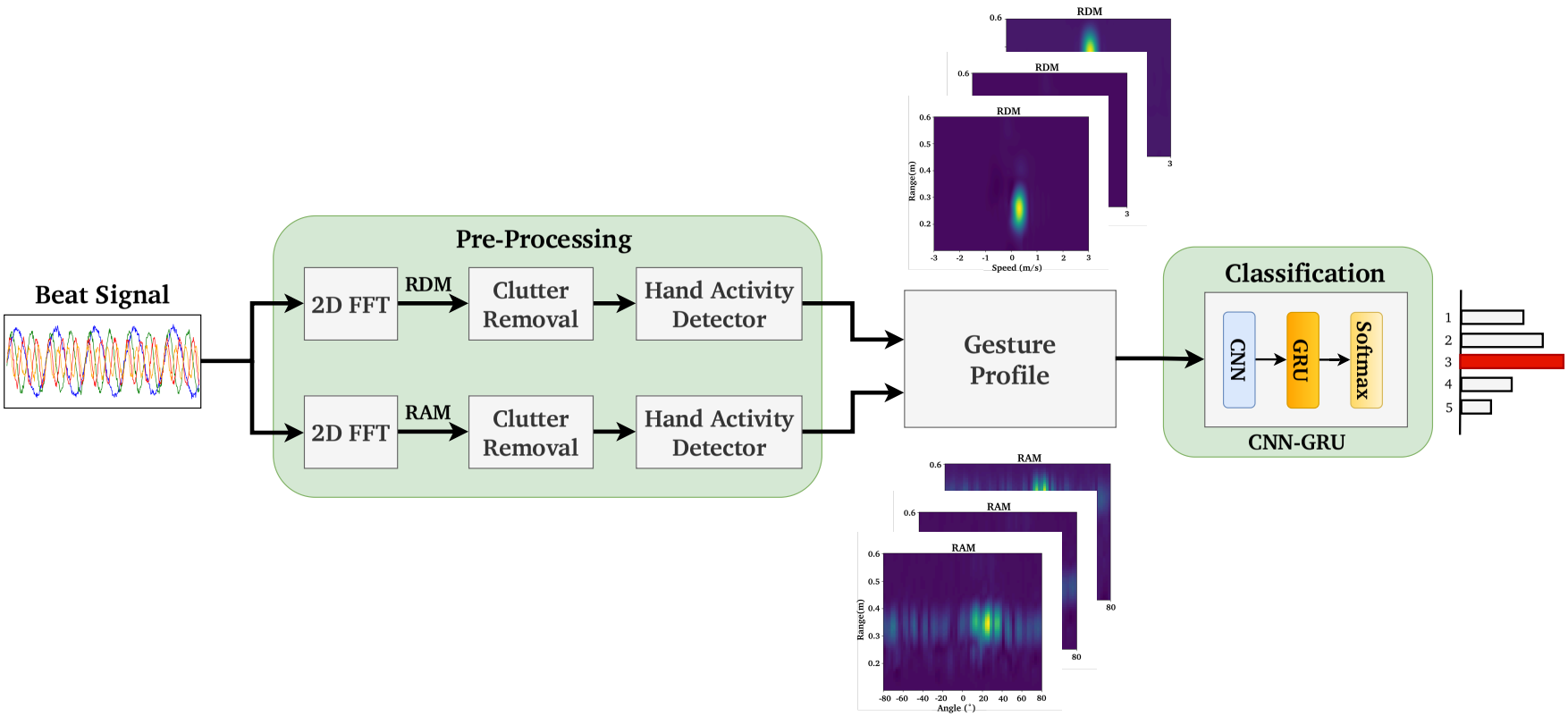


Fig. 5.1: Schematic illustration of the proposed radar-based hand gesture recognition system.

radar-based HGR systems more efficient than their camera-based counterparts in terms of power consumption.

Using micro-Doppler spectrograms [84, 85, 86] or RAMs [75, 87] is the most common approach in previous studies. However, HGR is facilitated when azimuth and elevation angles are also considered [56, 60, 88]. In [56, 87, 88], the radar's 3D data cube is reduced by calculating the weighted average of each parameter or by selecting a small subset of data points during each chirp frame. In contrast, [89] uses the entire collected RAMs for classification.

Various architectures of deep neural networks have already been explored for HGR systems, including Multilayer Perceptron (MLP) [87], CNN [90], RNN [57], or a cascade of the last two architectures as CNN-RNN [91]. Previous studies have excessively utilized LSTM units as RNN. Notably, the GRU, proposed in 2014 has demonstrated promising results comparable to those of its LSTM counterpart in video-based gesture identification [92].

To the authors' best knowledge, the feasibility of employing a dual-stream 2D CNN-GRU operating on pre-processed RDMs and RAMs has not been investigated thus far. Additionally, state-of-the-art works have employed a MIMO FMCW radar with a small antenna array limited to two transmit antennas (TX) and four receive antennas (RX) [56, 75, 87, 88, 93]. This study presents the first examination of a HGR system based on a 77 GHz MIMO FMCW radar consisting of 31 virtual receive elements.

The remaining sections of this chapter are organized as follows: we describe the processing of the radar's beat signal and the generation of clutter-free RDMs and RAMs. Next, we present the architecture of the proposed dual-stream CNN-GRU classifier, which incorporates the hyperparameter optimization. We also showcase the results achieved and compare them to those of state-of-the-art models. Finally, the chapter concludes with a summary of our findings.

5.2 Beat Signal Processing

Hand gestures, similar to other dynamic radar targets, are characterized by their range, Doppler velocity, and azimuth angle ($[r, v, \theta]$), which must be computed throughout the duration of each gesture across multiple chirp frames. The underlying trends in these features can be effectively tracked and analyzed by employing a 2D FFT to generate a pair of RDM and RAM for each frame [60].

5.2.1 Creating Range-Doppler Maps

The RDM for a single frame is generated as follows:

1. Arrange the beat signal of the k-th virtual antenna into a 2D matrix $\mathbf{S}_{(k)}$, where each column holds the fast-time samples of one chirp.
2. Apply a P-point FFT along the columns of $\mathbf{S}_{(k)}$ to produce the range-profile matrix $\mathbf{PR}_{(k)}$.
3. Perform a Z-point FFT across each row of $\mathbf{PR}_{(k)}$ to obtain the range-Doppler spectrum $\mathbf{RD}_{(k)}$.

4. Repeat steps 1-3 for all virtual antennas $(k = 1, \ldots, K)$.

5. To compute the final RDM for a single frame, the magnitude spectra of all antennas are summed, resulting in $\mathbb{RD}$.

Range-Doppler Map

$$\mathbb{RD} = \sum_{k=1}^{K} |\mathbf{PR}_{(k)}| \tag{5.1}$$

5.2.2 Creating Range-Angle Maps

The RAM for a single frame is computed through the following steps:

1. Represent the beat signal of the m-th chirp as a 2D matrix $\mathbf{V}^{(m)}$ [1], where each column contains samples from a virtual antenna.

2. Repeat the previous step for all chirps in the frame $(m = 1, \ldots, M)$.

3. Compute the **averaged virtual antenna signal**:

$$\overline{\mathbf{V}} = \frac{1}{M} \sum_{m=1}^{M} \mathbf{V}^{(m)} \tag{5.2}$$

4. Apply a P-point FFT along each column of $\overline{\mathbf{V}}$ to obtain the range profile matrix $\overline{\mathbf{RP}}$.

5. Perform a Q-point FFT across each row of $\overline{\mathbf{RP}}$ to compute the range-angle spectrum $\overline{\mathbf{RA}}$.

6. The magnitude of $\overline{\mathbf{RA}}$ represents the RAM:

Range-Angle Map

$$\mathbb{RA} = |\overline{\mathbf{RA}}| \tag{5.3}$$

5.2.3 Range, Velocity, and Angle Bin Spacing

The configuration parameters of the RBK2 radar used for the HGR application are summarized in Table 5.1. These include key values such as the chirp bandwidth $B = 3\,\text{GHz}$, center frequency $f_c = 75.5\,\text{GHz}$, sampling rate $f_s = 5\,\text{MHz}$, sweep duration $T_{\text{up}} = 64\ \mu\text{s}$, chirp repetition interval $T_{\text{p}} = 160\ \mu\text{s}$, and FFT sizes for range, Doppler, and angle axes: $P = 2048$, $Z = 512$, and $Q = 64$, respectively. The speed of light is assumed to be $c = 3 \times 10^8$ m/s.

Based on these parameters, the bin spacing in range, velocity, and angular domains can be calculated as follows:

[1] In this thesis, temporal indices, such as m, are represented as superscripts, while spatial indices, such as angle bin index or range bin index, are represented as subscripts.

Tab. 5.1: Simulation parameters for radar-based HGR

Parameter	Description	Value/Unit
Radar Parameters		
B	Chirp bandwidth	3 GHz
f_c	Central operating frequency	75.5 GHz
Timing Parameters		
f_s	Fast-time sampling frequency	5 MHz
T_{up}	Duration of each up-chirp	64 μs
T_{p}	Chirp repetition interval	160 μs
N	Number of samples per chirp	320
U	Number of chirps per frame	60
Processing Parameters		
Z	Size of the Doppler-FFT	512
P	Size of range-FFT	2048
Q	Size of the angle-FFT	64

Range Bin Spacing

$$\Delta r_{\text{bin}} = \frac{cN}{2BP} = \frac{3 \times 10^8 \cdot 320}{2 \cdot 3 \times 10^9 \cdot 2048} \approx 0.0078 \text{ m} = 0.78 \text{ cm} \tag{5.4}$$

Velocity Bin Spacing

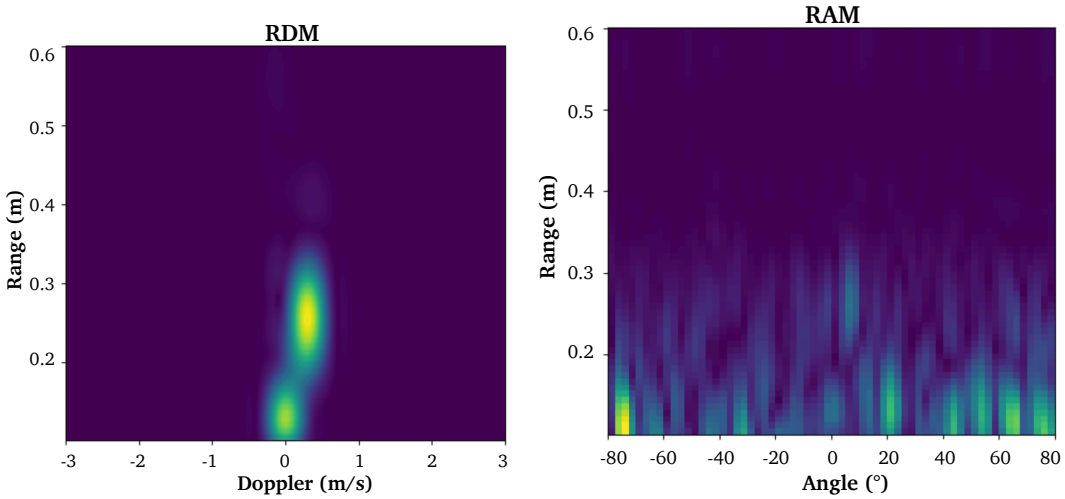
$$\Delta v_{\text{bin}} = \frac{c}{2 f_c Z T_p} = \frac{3 \times 10^8}{2 \cdot 75.5 \times 10^9 \cdot 512 \cdot 160 \times 10^{-6}} \approx 0.0243 \text{ m/s} = 2.43 \text{ cm/s} \tag{5.5}$$

Angle Bin Spacing

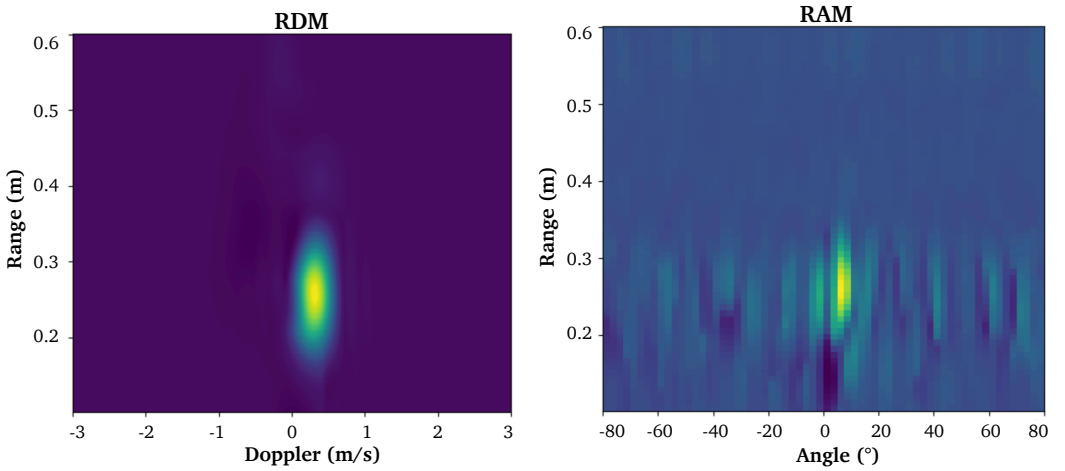
$$\Delta \theta_{\text{bin}} \approx \frac{2}{Q} \cdot \frac{180}{\pi} = \frac{2}{64} \cdot \frac{180}{\pi} \approx 1.79^\circ \tag{5.6}$$

5.2.4 Enhancement of Range-Doppler and Range-Angle Maps

According to our experiments, the generated RDMs and RAMs require a subsequent image enhancement stage to suppress clutter and interference arising from the Radio Frequency (RF) front-end. To achieve this, a pair of calibration matrices is computed as a background model. This background model represents the mean of the RDM and RAM over several chirp frames prior to the hand gesture. The difference between the realized model and the current frame produces a pair of RDM and RAM with minimized artifacts. Fig. 5.2 illustrates this process. It is important to discard unnecessary range and Doppler velocity bins that provide almost no information in



(a)



(b)

Fig. 5.2: The RDM and RAM before and after the clutter removal at the same chirp frame: (a) before; (b) after applying clutter removal technique

RDM and RAM about the hand gesture, also known as range and Doppler gating. This step significantly reduces computational time and improves performance during the classification stage. Therefore, the maximum range and velocity values in RDMs and RAMs are restricted to 0.6 m and 3 m/s, respectively. As a result, each frame is represented by an RDM and RAM with sizes of 65×125 and 65×64 pixels, respectively.

5.3 Hand Activity Detection Using Energy Comparison

The hand activity detection algorithm relies on comparing the energy levels of the RDM and RAM with adaptive thresholds. This approach uses an Exponential Moving Average (EMA) to smooth the energy values over time, capturing the underlying trends while remaining responsive to significant changes caused by hand gestures.

5.3.1 Initialization

Before each detection cycle, several key variables and buffers are initialized to ensure effective processing of hand activity data:

- **Circular Buffers**: These are used to retain both current and historical data for gesture analysis.
  - $\mathcal{D}$: A circular buffer of size M dedicated to storing the current and previous RDMs, facilitating the tracking of gesture dynamics over time.
  - $\mathcal{A}$: A circular buffer of size M designed to store the current and previous RAMs.
- **Counters**: These are used to track gesture length and the absence of gestures across multiple time frames.
  - g_L: The gesture length counter, which increments each time hand activity is detected, measuring the duration of a gesture.
  - g_N: The no-gesture counter, which increments when no hand activity is detected, serving as a baseline for gesture recognition.

5.3.2 Data Collection and Storage

Let i denote the index of the current chirp frame. During each frame, the current Range-Doppler Map $\mathbb{RD}^{(i)}$ and Range-Angle Map $\mathbb{RA}^{(i)}$ are captured and stored in their respective circular buffers:

$$\mathcal{D}^{(i)} = \mathbb{RD}^{(i)}, \quad \mathcal{A}^{(i)} = \mathbb{RA}^{(i)} \tag{5.7}$$

The index i is updated in a circular manner as follows:

$$i = (i + 1) \mod M \tag{5.8}$$

In this implementation, $M = 18$, meaning the system retains data from the last 18 frames for historical analysis.

5.3.3 Energy Calculation

At each frame i, the total energy in the Range-Doppler Map $\mathbb{RD}^{(i)}$ and Range-Angle Map $\mathbb{RA}^{(i)}$ is calculated as the sum of the squared magnitudes of all pixel values in the respective maps:

Total Energy Computation

$$E_D^{(i)} = \sum_{(x,y)\in\mathbb{RD}^{(i)}} \left|\mathbb{RD}_{(x,y)}^{(i)}\right|^2 \tag{5.9}$$

$$E_A^{(i)} = \sum_{(x,y)\in\mathbb{RA}^{(i)}} \left|\mathbb{RA}_{(x,y)}^{(i)}\right|^2 \tag{5.10}$$

These values, $E_D^{(i)}$ and $E_A^{(i)}$, represent the instantaneous energy detected in the respective maps at data frame i.

5.3.4 Exponential Moving Average (EMA)

To smooth out short-term fluctuations in energy values, an EMA is maintained for both $\mathbb{RD}$ and $\mathbb{RA}$. The EMA is updated at each time frame i using the following formulas:

Exponential Moving Average Computation

$$\text{EMA}_D^{(i)} = \alpha \cdot E_D^{(i)} + (1 - \alpha) \cdot \text{EMA}_D^{(i-1)} \tag{5.11}$$

$$\text{EMA}_A^{(i)} = \alpha \cdot E_A^{(i)} + (1 - \alpha) \cdot \text{EMA}_A^{(i-1)} \tag{5.12}$$

Here:

- α is the smoothing factor ($0 < \alpha \leq 1$), which controls the sensitivity of the EMA to recent changes.
- $\text{EMA}_D^{(i-1)}$ and $\text{EMA}_A^{(i-1)}$ are EMA of the energy values for $\mathbb{RD}^{(i-1)}$ and $\mathbb{RA}^{(i-1)}$.

The EMA values offer an adaptive baseline for comparison. In this implementation, we set the smoothing factor $\alpha = 0.2$, which balances responsiveness and stability in tracking energy fluctuations.

5.3.5 Dynamic Thresholds

The detection thresholds for both maps are dynamically computed based on the EMA values:

Dynamic Threshold

$$tr_D^{(i)} = \gamma \cdot \text{EMA}_D^{(i)} \tag{5.13}$$

$$tr_A^{(i)} = \gamma \cdot \text{EMA}_A^{(i)} \tag{5.14}$$

Where:

- $\gamma > 1$ is a scaling factor to increase the sensitivity of the threshold.
- $tr_D^{(i)}$ and $tr_A^{(i)}$ are the thresholds for detecting hand activity in $\mathbb{RD}^{(i)}$ and $\mathbb{RA}^{(i)}$, respectively.

Here, we use $\gamma = 1.2$ as the threshold scaling factor. These thresholds adapt over time, reflecting the overall energy levels observed during a detection cycle. Fig. 5.4 visually demonstrates the key stages of the hand activity detection process. It illustrates the initialization of circular buffers, the data capture procedure, and the subsequent calculation of energy parameters, all of which are critical for the gesture detection logic.

5.3.6 Hand Activity Detection

Hand activity is detected if the energy in $\mathbb{RD}$ or $\mathbb{RA}$ exceeds the corresponding dynamic threshold:

Gesture Detection Logic

$$\text{Gesture Detected} = \begin{cases} \text{True} & \text{if } E_D^{(i)} > tr_D^{(i)} \text{ or } E_A^{(t)} > tr_A^{(i)} \\ \text{False} & \text{otherwise} \end{cases} \tag{5.15}$$

This comparison ensures that the detection logic is adaptive to background energy and sensitive to abrupt changes caused by hand gestures.

To illustrate the functionality of the thresholding mechanism, Fig. 5.3 presents a one-dimensional representation of the energy signal from both $\mathbb{RD}$ and $\mathbb{RA}$ over a period of 150 time frames. The graph shows the raw energy signal, EMA and the dynamically calculated threshold. The shaded areas indicate the time frames in which the energy signal exceeds its corresponding threshold, suggesting possible detection of hand activity. This graph effectively demonstrates how the adaptive threshold follows energy trends while filtering out short-term fluctuations.

5.3.7 Incrementing Gesture Counters

When a gesture is detected, the gesture length counter g_L is incremented, and the no-gesture counter g_N is reset:

$$g_L += 1, \quad g_N = 0 \tag{5.16}$$

If no hand activity is detected, the no-gesture counter g_N is incremented instead:

$$g_N += 1 \tag{5.17}$$

5.3.8 Multi-Cycle Gesture Confirmation

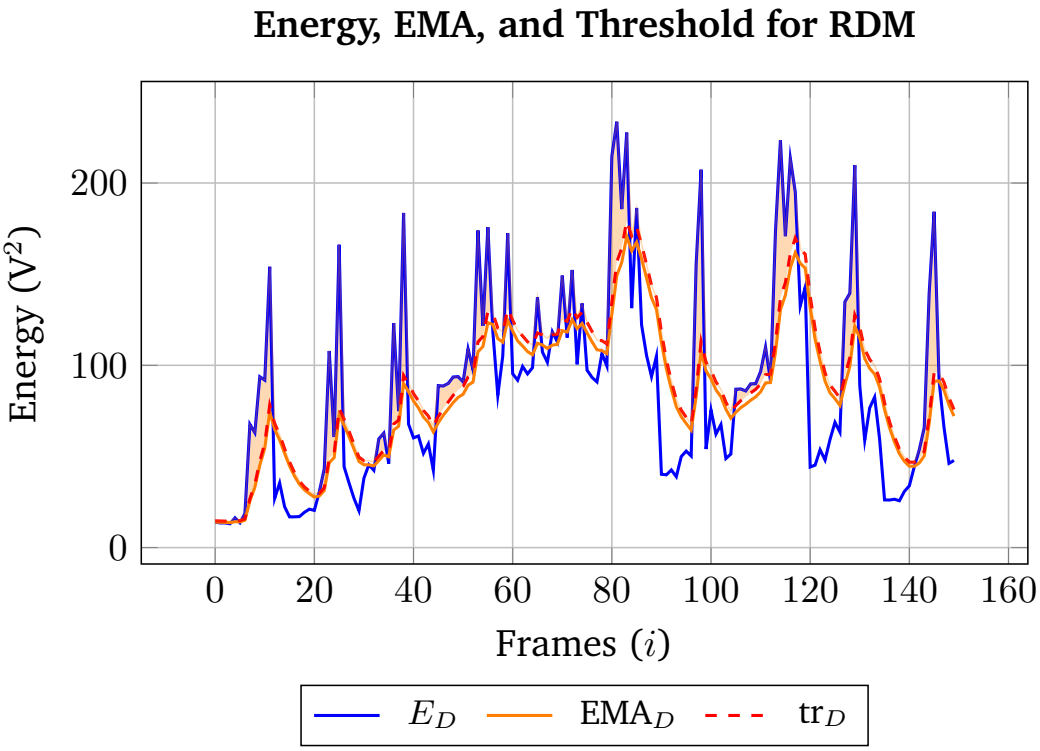
For a gesture to be validated, the detection conditions must persist for a minimum number of consecutive time frames, denoted as $T_{\text{Start}} = 5$. A gesture is confirmed if:

$$g_L \geq T_{\text{Start}} \tag{5.18}$$

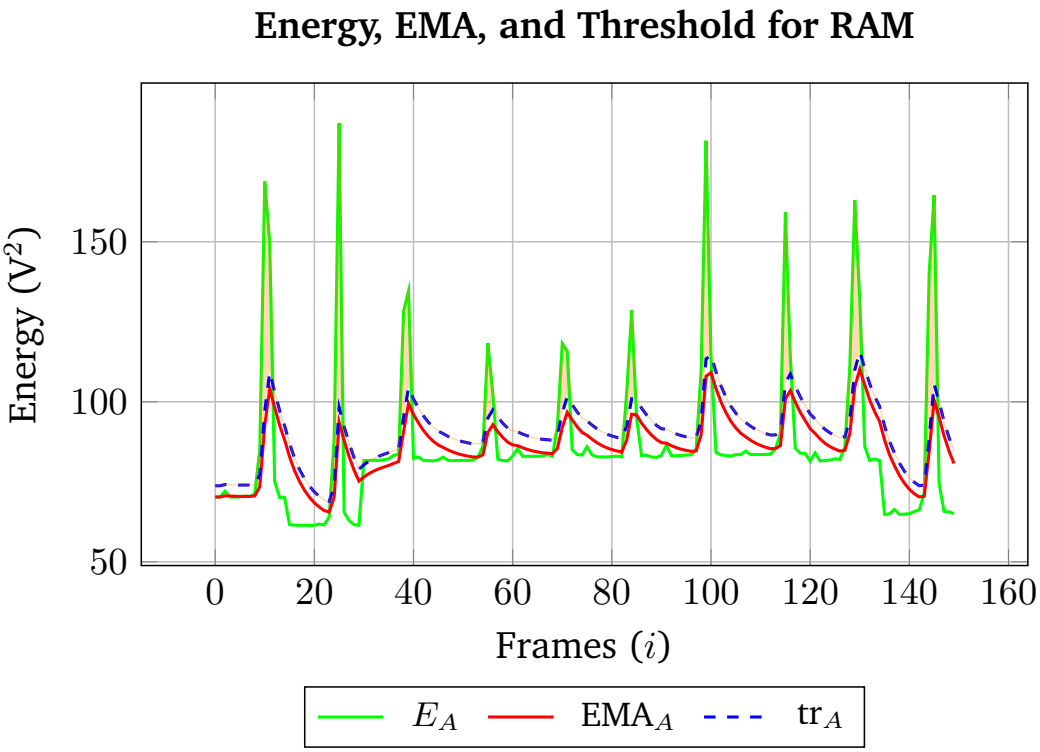
On the other hand, if no gesture is detected for more than $T_{\text{End}} = 3$ consecutive cycles, the gesture is considered finished:

$$g_N \geq T_{\text{End}} \tag{5.19}$$

Once these conditions are satisfied, the system extracts the **gesture profile** by selecting frames from the circular buffer. Since $M = 18$ and the gesture termination flag is activated after $T_{\text{End}} = 3$ consecutive non-gesture frames, the last three frames in the buffer do not contain valid gesture information. Therefore, the gesture profile consists of the 15 **frames preceding the first non-gesture frame**, ensuring that only



(a)



(b)

Fig. 5.3: Energy, EMA, and threshold calculations performed over 150 frames: (a) RDM; (b) RAM. The time frames where energy is above the threshold are highlighted.

frames containing actual hand activity are included. After extracting the gesture profile, the system resets the gesture counters to prepare for subsequent detections.

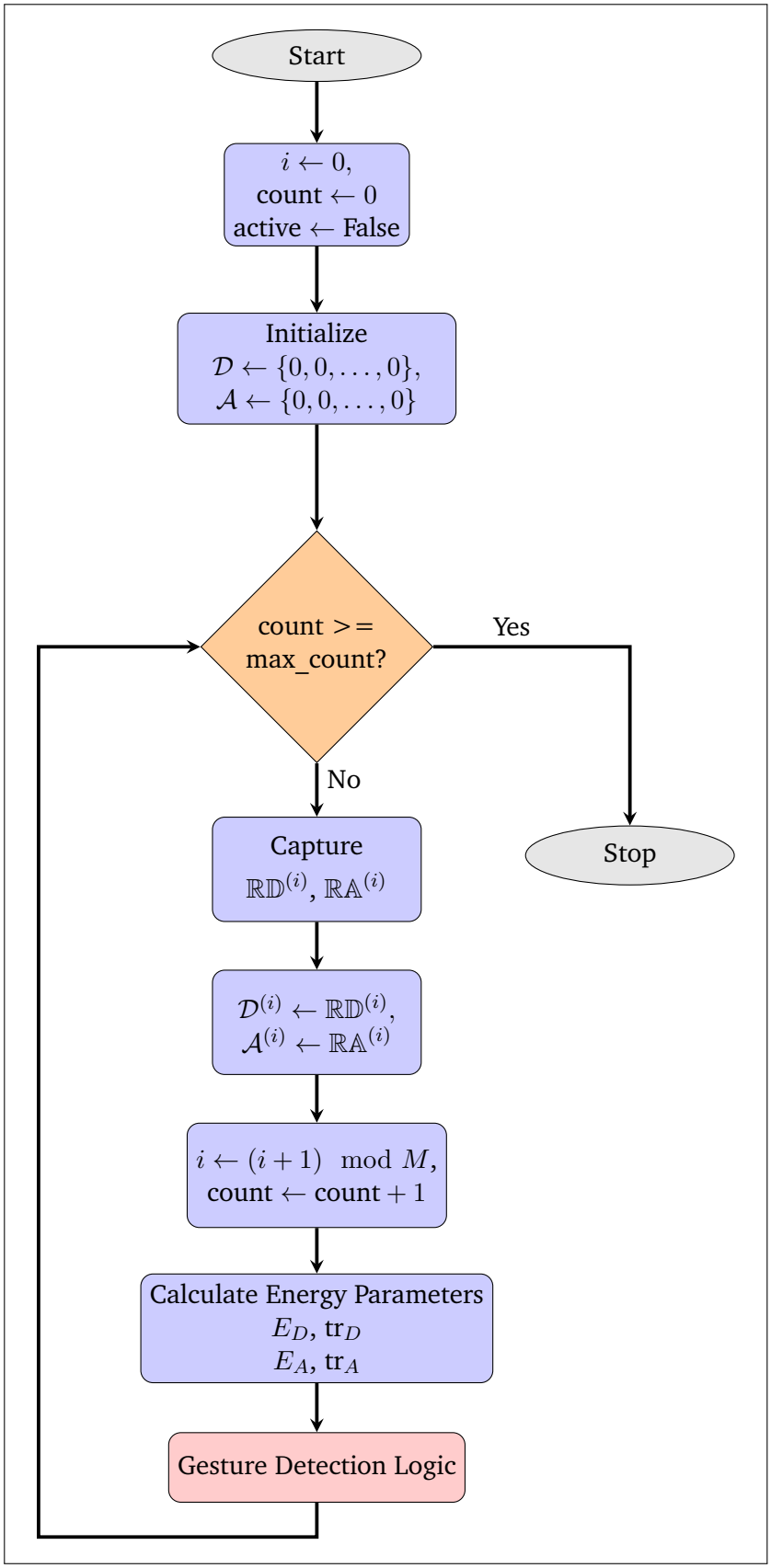


Fig. 5.4: Initialization and data collection stage of hand activity detector

5.3.9 Threshold Parameter Selection

The thresholds T_{Start} and T_{End} are determined through a series of preliminary experiments designed to balance sensitivity and accuracy:

- Longer values for T_{Start} minimize false starts but may overlook brief gestures.
- Conversely, shorter values for T_{End} enhance responsiveness but could lead to segmentation of longer gestures.

This careful selection ensures that the algorithm is both effective in detecting genuine gestures and capable of adapting to the fine-grained hand movements. Fig. 5.5 illustrates the gesture detection stage of the hand activity detector, where previously calculated energy parameters are utilized for gesture identification.

5.4 Gesture Design and Acquisition

The performance of the proposed hand gesture recognition system was evaluated using a set of ten distinct gestures, as illustrated in Fig. 5.6. These gestures were carefully selected to represent a broad range of movement characteristics, including variations in direction, duration, and complexity. By including a diverse range of gestures, we aimed to challenge the system's ability to accurately recognize and differentiate between subtle variations in hand movements.

The simplest gestures, such as push and pull motions, were effectively recognized due to their distinct and consistent movement signatures. This is particularly evident when analyzing the trajectory of the highest-magnitude point in both RDMs and RAMs, as shown in Fig. 5.7. These visual representations not only capture the movement patterns but also provide critical insights into the spatial and temporal characteristics of the gestures, thereby enhancing the system's ability to accurately recognize each one.

In the case of the Push-Pull (PSPL) and Pull-Push (PLPS) gestures, the system detected counter-clockwise circular trajectories for both. However, a key distinction lies in the spatial distribution of these trajectories. For PSPL, the first half of the movement occurs in the left half-plane, whereas for PLPS, it takes place in the right half-plane. Notably, push gestures are characterized by a **crescent-shaped path** in the left half-plane, while pull gestures exhibit a **crescent arc** in the right half-plane. This subtle yet significant difference enables the system to effectively differentiate between the two gestures, despite their overall similarity in pattern.

Additionally, for other gestures such as Swipe-to-Right (SR) and Swipe-to-Left (SL), the azimuth angle plays a more prominent role in defining the motion. In these cases, the system relies on the RAMs, which provide a clearer and more intuitive trajectory over successive time frames.

For the data acquisition process, eight volunteers performed each of the designed gestures 30 times, resulting in a total of $8 \times 30 \times 10 = 2400$ gesture samples. Participants were given brief instructions on how to perform the gestures, but the specific details of hand movement were not standardized to encourage variability in the dataset.

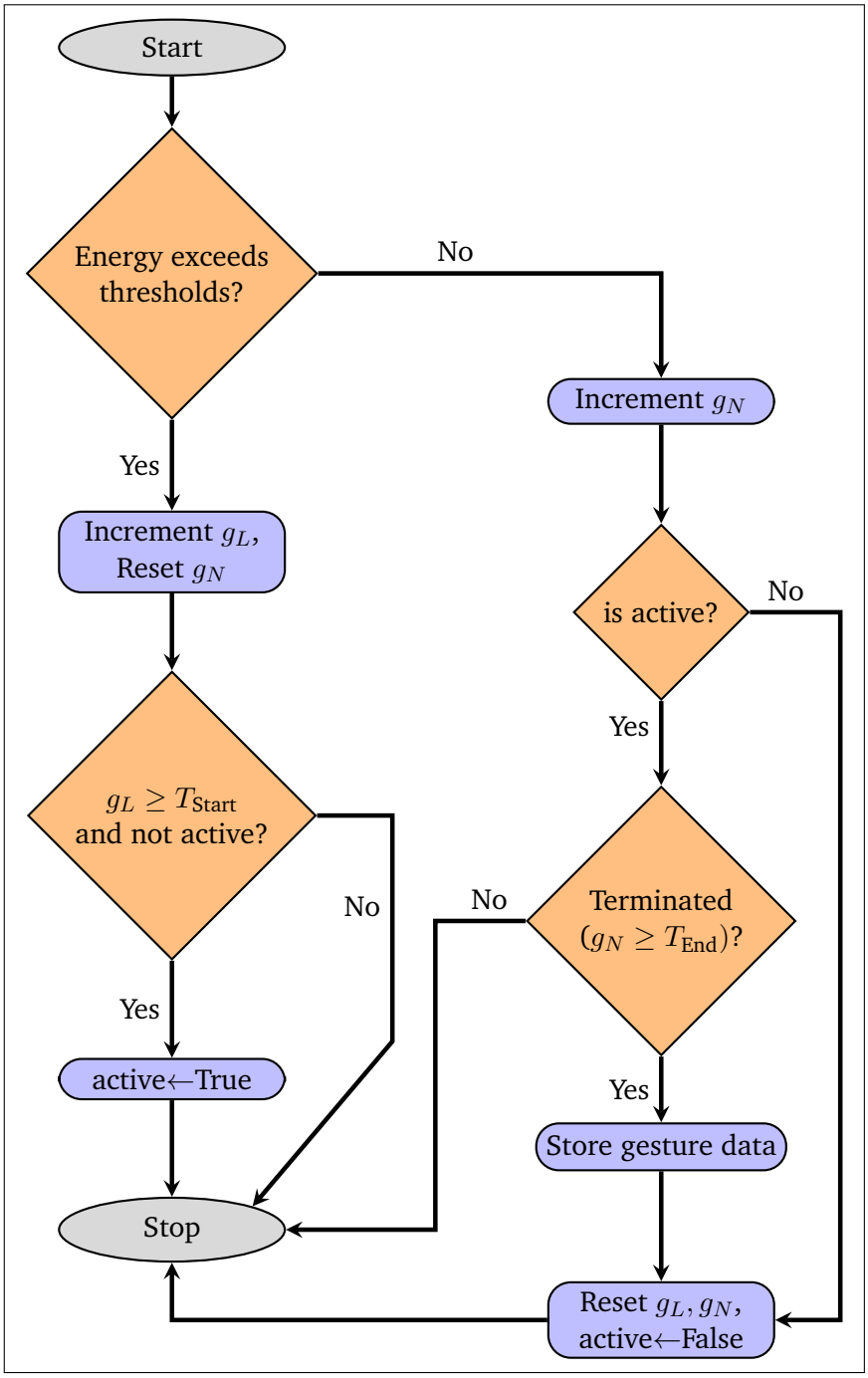


Fig. 5.5: Gesture detection logic of the hand activity detector illustrated in Fig. 5.4

5.4.1 Normalization of Input Data

Normalization is a crucial preprocessing step in deep learning models, particularly for input data such as RDMs and RAMs. By scaling input values to a standard range, typically between 0 and 1, normalization can significantly enhance model performance by reducing the impact of large variations in the data [50, 94, 95]. The normalization formula is given by:

$$I_{\text{Norm}} = \frac{I - V_{\text{Min}}}{V_{\text{Max}} - V_{\text{Min}}} \tag{5.20}$$

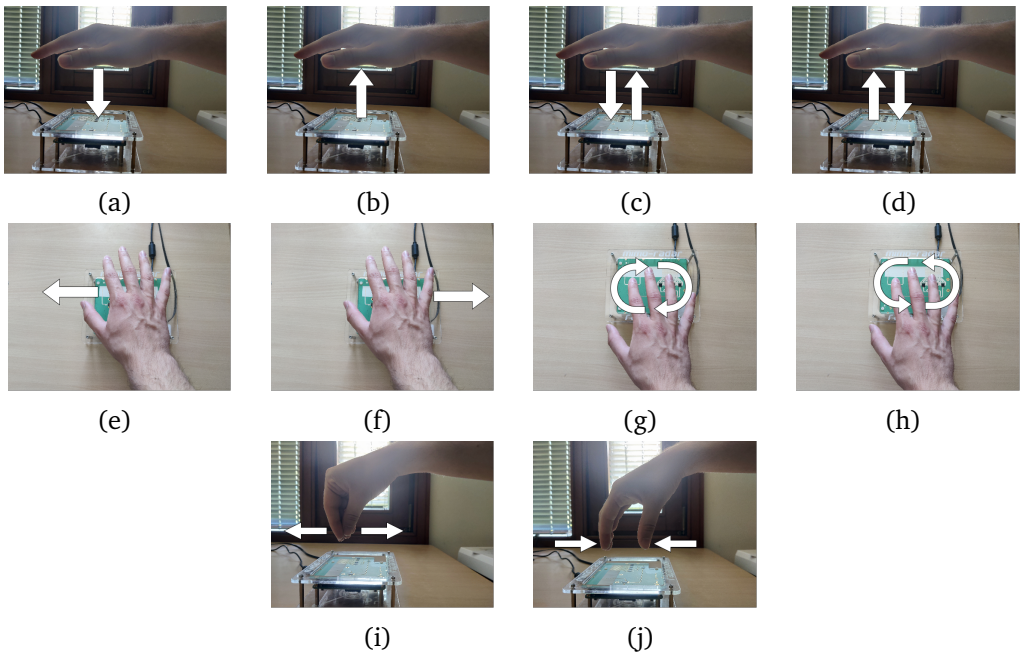


Fig. 5.6: The 10 hand gesture types used in the proposed HGR system: (a) Push (PS); (b) Pull (PL); (c) Push-Pull (PSPL); (d) Pull-Push (PLPS); (e) Swipe Left (SL); (f) Swipe Right (SR); (g) Clockwise Rotation (CW); (h) Counter-Clockwise Rotation (CCW); (i) Pinch Open (PO); (j) Pinch Close (PC)

where I is the pixel value, V_{Min} the minimum, and V_{Max} the maximum in the dataset. Normalization prevents certain features from dominating the learning process by bringing all features onto a similar scale. This also helps in faster convergence during gradient-based optimization and reduces the risk of vanishing or exploding gradients, common in deep networks like CNNs and RNNs [94, 95]. Furthermore, normalized inputs enhance the performance of activation functions such as ReLU and tanh, ensuring a smoother learning process. Consequently, normalization improves the model's ability to generalize, making it more robust to variations in real-world data, which is essential for accurate gesture recognition.

5.5 Proposed Deep Neural Network Model

Deep learning has emerged as a powerful tool for hand gesture recognition due to its ability to automatically learn spatial and temporal patterns from raw data. In this work, we propose a deep neural network architecture tailored for gesture classification, leveraging both convolutional layers for spatial feature extraction and GRUs for capturing temporal dependencies. A detailed introduction to these neural network layers can be found in Appendix C.

The motivation for selecting a **CNN-GRU** architecture arises from the need to effectively process radar-based gesture data, which encompasses both spatial and temporal information. By integrating convolutional layers with recurrent units, we aim to create a model that can accurately distinguish subtle variations in gestures.

5.5.1 Architectural Rationale

The proposed model is designed as a dual-stream deep neural network that separately processes two key radar representations. Each stream consists of an identical

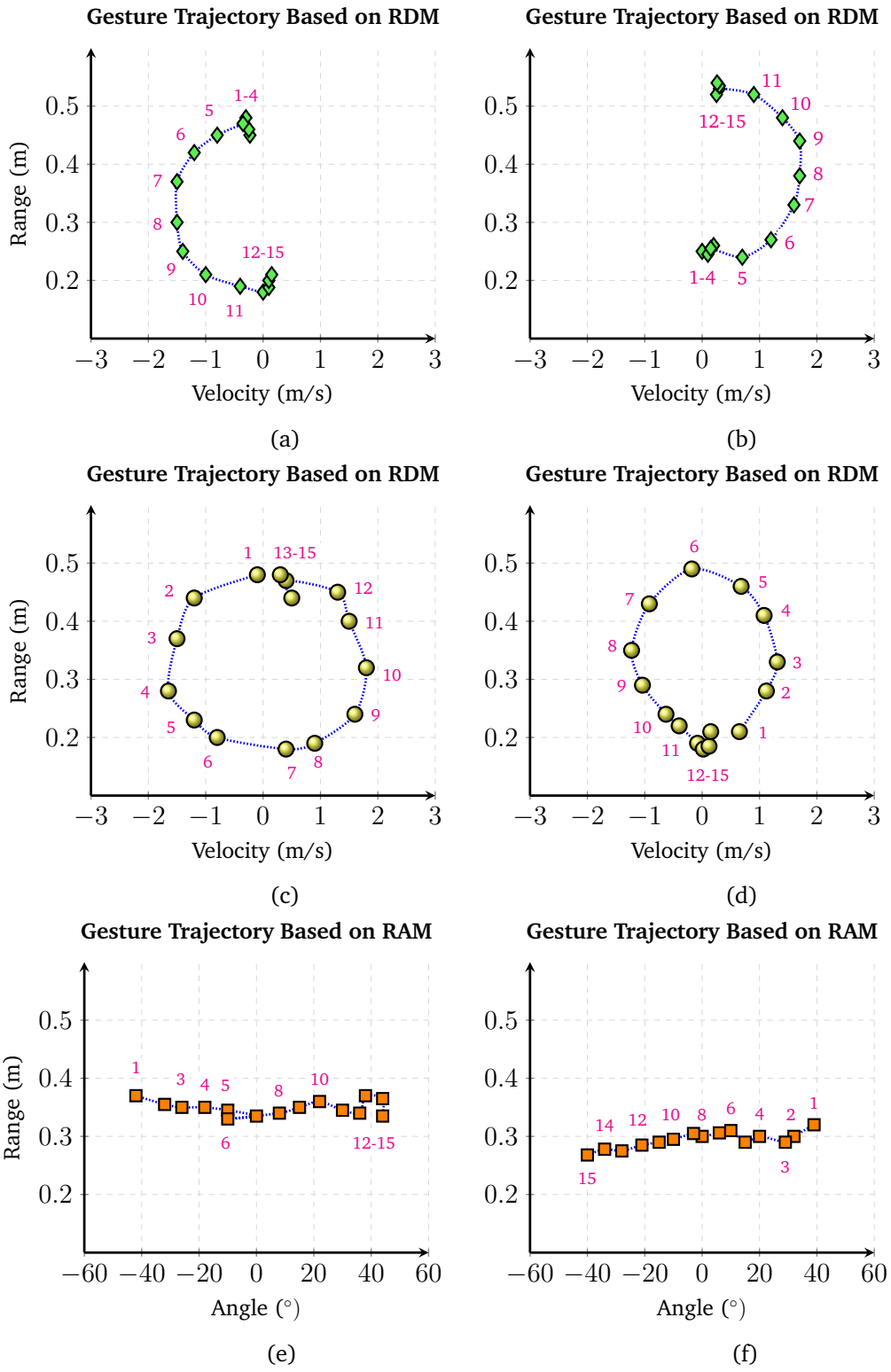


Fig. 5.7: Trajectory plot of PS, PL, PSPL, PLPS, SR, and SL gestures over 15 time frames: (a) PS; (b) PL; (c) PSPL; (d) PLPS; (e) SR, (f) SL

CNN-GRU structure to independently extract meaningful features from RDMs and RAMs. These two processed streams are then fused to make a final classification decision. The choice of a dual-stream approach is motivated by the following considerations:

- **Independent feature learning**: RDMs and RAMs capture distinct motion characteristics. Processing them separately enables the model to learn optimal representations for each modality.
- **Parallel processing efficiency**: Instead of combining inputs early, allowing each stream to learn independently before merging improves the network's ability to extract discriminative features.
- **Enhanced generalization**: By fusing spatial and motion-related features at a later stage, the model can generalize better across different gesture variations.

5.5.2 CNN-GRU Model Architecture

Each stream in the dual-stream model follows a structured pipeline that includes:

1. **Convolutional Layers** – These layers are responsible for learning spatial features from the input radar maps, extracting local patterns such as edges and motion-related features.
2. **Batch Normalization & Activation Layers** – These layers improve convergence and enable the network to learn more complex, nonlinear representations.
3. **GRU Layers** – Designed to capture temporal dependencies in the radar sequence data, these layers allow the model to understand gesture dynamics over time.
4. **Feature Fusion** – The outputs from both the RDM and RAM streams are concatenated to create a unified feature representation.
5. **Fully Connected Layers** – These layers further refine the learned features before producing the final classification output.

The overall structure of the proposed dual-stream CNN-GRU model is illustrated in Fig. 5.8, where each stream independently processes either RDMs or RAMs before their outputs are fused for final classification.

Role of the Fully Connected Layer in Final Classification

Before applying the softmax function to obtain class probabilities, the model requires a Fully Connected (FC) layer with 10 units (one per gesture class). The role of this layer is twofold:

- **Mapping high-dimensional features to logits**: The concatenated feature vector from the CNN-GRU pipeline is typically high-dimensional. The FC layer condenses this representation into a **logit vector** of size 10, where each value corresponds to an unnormalized confidence score for a specific gesture class.

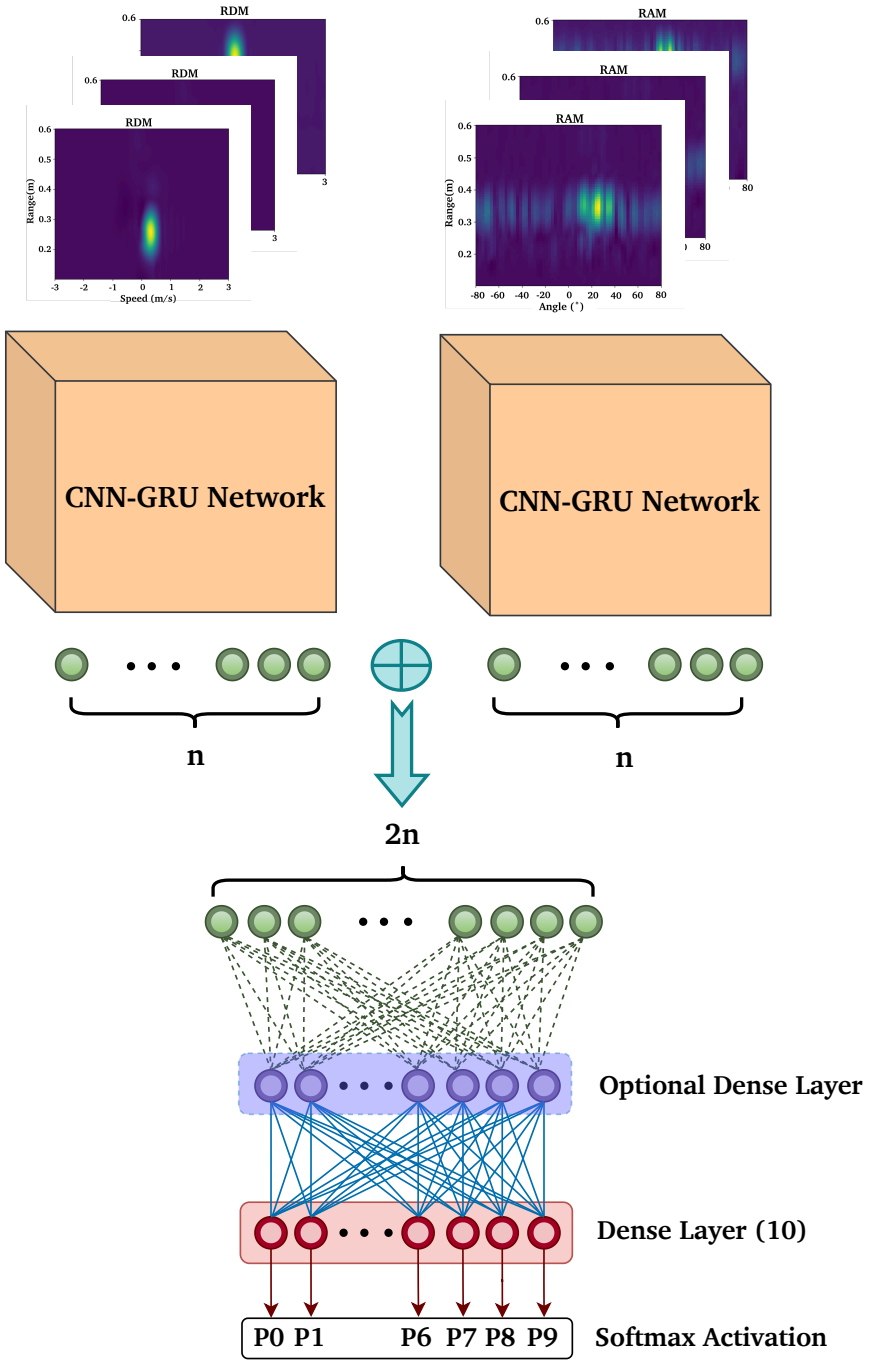


Fig. 5.8: Overview of the proposed dual-stream CNN-GRU model architecture for processing RDMs and RAMs. In this visualization, the CNN-GRU model is represented as a black box, where '**n**' signifies the number of output neurons from the final GRU layer.

- **Enhancing learnability**: The fully connected layer introduces trainable parameters that adaptively learn to weigh different features for improved classification performance.

The logits produced by the FC layer are then passed through the softmax function, which converts them into a probability distribution over the 10 gesture classes. The output probability for each class is given by [50]:

$$P(y_i) = \frac{e^{z_i}}{\sum_{j=1}^{10} e^{z_j}} \tag{5.21}$$

where z_i represents the logit value for class i. This ensures that all probabilities sum to 1, making the output interpretable as a classification confidence score.

5.5.3 Necessity of Hyperparameter Tuning

While the overall model architecture is well-defined, the exact number of convolutional layers, GRU layers, and GRU units significantly impacts performance. To determine the optimal configuration, hyperparameter tuning is necessary. The following section will detail the tuning process used to refine these architectural choices and enhance classification accuracy.

5.6 Hyperparameter Optimization Strategy

Selecting the right combination of hyperparameters is crucial for achieving optimal performance in the dual-stream CNN-GRU model. Although the general architecture is predefined, factors such as the number of convolutional layers, GRU layers, and GRU units directly impact the model's ability to capture spatial and temporal features. Furthermore, regularization strategies such as dropout, the number of dense layers, and batch size play a significant role in preventing overfitting and ensuring stable training.

To systematically explore these hyperparameters, random search was employed using **Keras Tuner**. Unlike exhaustive grid search, which evaluates all possible combinations, random search samples a subset of configurations, allowing efficient exploration within a limited number of trials. Specifically, 40 different configurations were tested, and each model was trained and evaluated based on validation accuracy. The tuning process was structured as follows:

- **Number of convolutional layers** (`num_conv_layers`): Varied between 3 and 5.
- **Number of GRU layers** (`num_gru_layers`): Tested values between 3 and 5.
- **GRU units per layer** (`gru_units`): Explored configurations of 32, 64, 96, and 128 units.
- **Base filter size** (`base_filters`): Chosen as 16 or 32, growing exponentially across layers.
- **Number of dense layers** (`num_dense_layers`): Tuned between 0 and 2 layers, with the second layer (if present) always having half the units of the first.
- **Dropout rate** (`drop_rate`): Examined values of 0.2, 0.4, and 0.6.
- **Batch size** (`batch_size`): Ranged from 16 to 64 to assess its impact on gradient updates and stability.

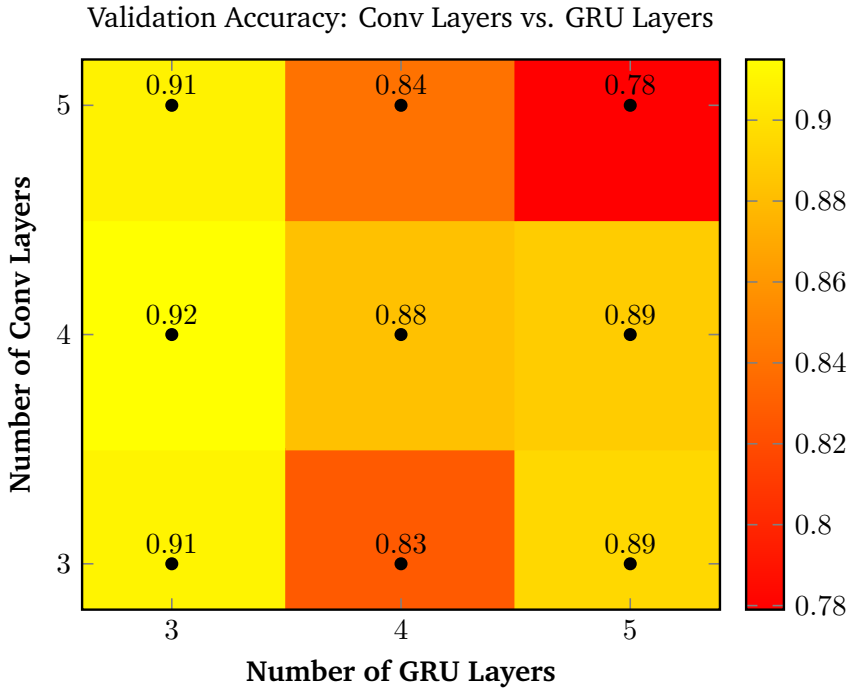


Fig. 5.9: Heatmap showing the validation accuracy as a function of the number of convolutional layers and GRU layers. The color intensity represents the accuracy, with brighter colors indicating higher accuracy.

Each trial instantiated a model with a different set of hyperparameters, trained it, and recorded its validation accuracy. At the end of the process, the best-performing configuration was identified based on its ability to generalize well.

The following subsections will delve into the effect of these hyperparameters individually, leading to a step-by-step reasoning behind the final optimal configuration.

5.6.1 Impact of Convolutional Layers on Spatial Feature Extraction

The number of convolutional layers directly influences the model's ability to extract spatial features from input data. A deeper network can capture more complex patterns, but excessive layers may lead to overfitting or increased computational overhead. To assess this tradeoff, different configurations were evaluated, with three, four, and five convolutional layers (`num_conv_layers`) tested alongside varying numbers of GRU layers (`num_gru_layers`).

Fig. 5.9 presents a heatmap illustrating the validation accuracy for different combinations of convolutional and GRU layers. From this visualization, several key observations emerge:

- **Increasing from three to four convolutional layers generally improves accuracy**, particularly when using three or four GRU layers.
- **A fifth convolutional layer does not provide consistent benefits** for some GRU configurations; in fact, accuracy may slightly decrease, suggesting diminishing returns or overfitting.

- **The highest validation accuracy is achieved with four convolutional layers and three GRU layers**.

- **Three GRU layers consistently yield the best accuracy** across different convolutional configurations, indicating that additional GRU layers may not provide further improvements and could even degrade performance.

This analysis establishes four convolutional layers as the optimal choice for spatial feature extraction. The following subsection will further examine the impact of GRU layers, analyzing their role in capturing temporal dependencies.

5.6.2 Effect of GRU Layers and GRU Units on Validation Accuracy

To further refine the architecture, we analyze the impact of both the number of GRU layers and the number of GRU units per layer. Fig. 5.10 illustrates validation accuracy for different configurations, providing critical insights into their influence on performance.

Several key observations emerge from this figure:

- Models with **three GRU layers** achieve the highest accuracy across different unit configurations. The performance declines for deeper networks (four and five layers), particularly when using a higher number of units, suggesting that excessive depth may lead to overfitting or vanishing gradient issues.

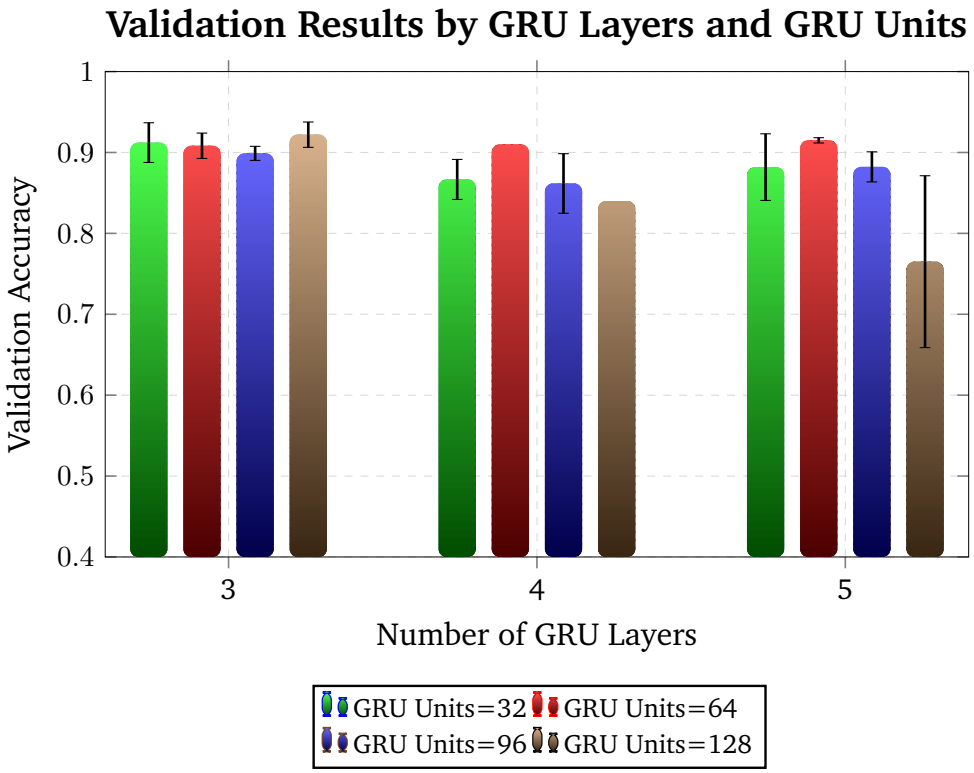


Fig. 5.10: Validation accuracy across different GRU layer depths and GRU unit configurations.

- The number of GRU units per layer significantly influences accuracy. Among all configurations, using **128 units with three layers** results in the highest validation accuracy. Additionally, **32 units with three layers** also demonstrate strong performance, offering a lower-complexity alternative while maintaining stable accuracy.
- Models with 64 GRU units exhibit relatively balanced performance across different layer depths. Unlike 32 and 128 units, which show clear peaks in performance, 64-unit models maintain stable accuracy across various layer depths, making them a flexible choice.
- Larger GRU models exhibit greater variance in accuracy. The error bars in Fig. 5.10 indicate that models with more units per layer tend to have higher fluctuations in validation accuracy. Notably, for five-layer configurations, the accuracy drops significantly with 128 units, suggesting potential overfitting.

These results indicate that a model with **three GRU layers** and either **128 or 32 units per layer** provides the best trade-off between accuracy and stability. This is consistent with the findings from the heatmap (Fig. 5.9), which further supports the idea that increasing the number of layers or units does not necessarily improve performance.

5.6.3 Impact of Convolutional Layers and GRU Units

To further refine our understanding of the optimal network architecture, we analyze the effect of different configurations of convolutional layers and GRU units. Fig. 5.11 presents a scatter plot illustrating the relationship between these two hyperparameters and the resulting validation accuracy.

Key observations from the figure include:

- Models with **four convolutional layers** consistently demonstrate high accuracy across various GRU unit configurations, reinforcing our earlier findings about the effectiveness of this depth.
- The best accuracy is achieved using 32 **GRU units** in combination with **three convolutional layers**, highlighting an efficient configuration that balances performance and model complexity.
- A slightly lower, yet still strong, performance is observed with 128 **GRU units** and **four convolutional layers**, supporting earlier findings while offering a more expressive alternative when complexity is less constrained.
- Increasing convolutional layers beyond four does not yield improvements, and in some cases, it leads to a decline in accuracy, indicating diminishing returns from deeper feature extraction.

5.6.4 Effect of Dense Layers and Dense Units on Accuracy

A separate box and scatter plot, shown in Fig. 5.12 was created to assess how the number of dense layers and dense units affect model performance. Different markers

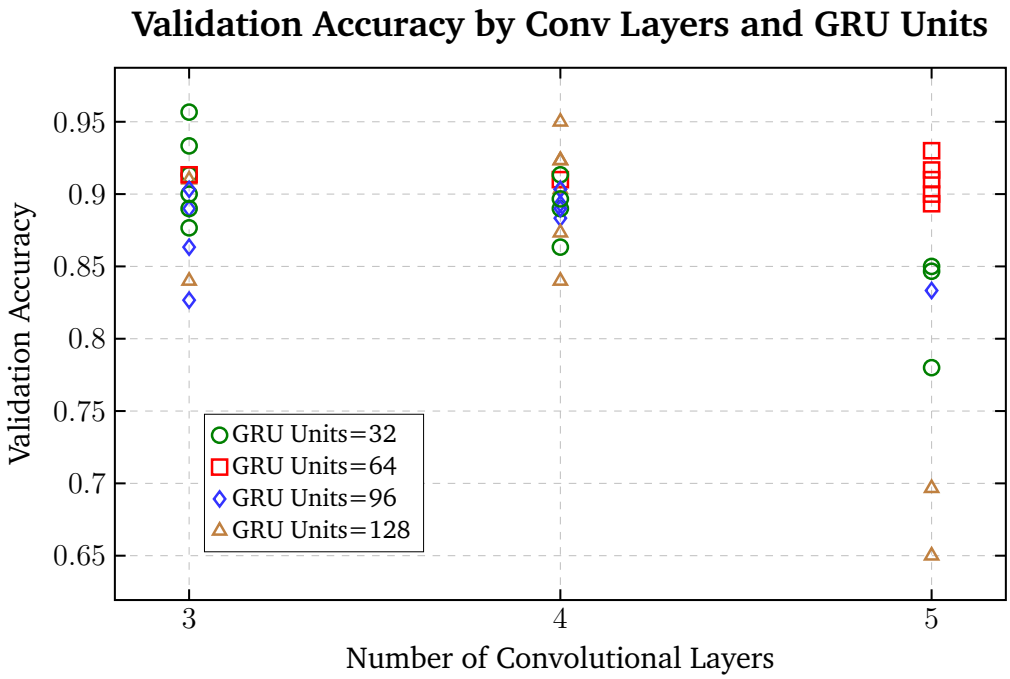


Fig. 5.11: Validation accuracy as a function of the number of convolutional layers and GRU units.

were used to represent varying numbers of dense units. Each boxplot in Fig. 5.12a summarizes the distribution of validation accuracies for a specific number of dense layers, indicating the median, Interquartile Range (IQR), and whiskers that represent the range excluding outliers.

Observations from the Box and Scatter Plot

✓ **Zero Dense Layers Performed Best**

- The highest validation accuracy (0.957) was achieved with **zero dense layers**. Although this configuration performed the best, it also demonstrated significant fluctuations across multiple tuning attempts, suggesting potential instability in generalization.

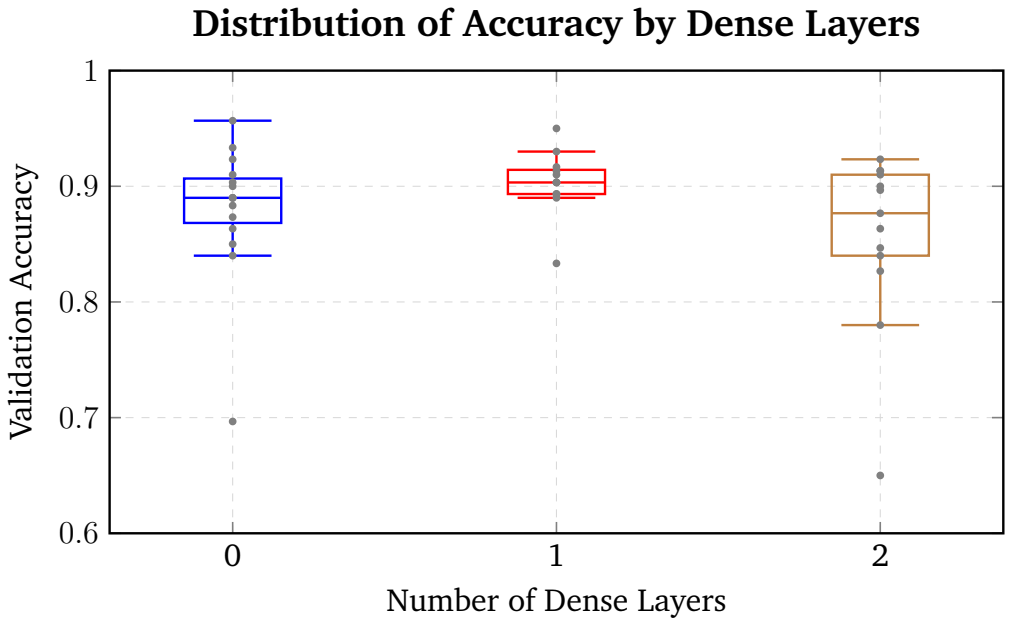
✓ **One Dense Layer Models Were Consistent**

- Most models with **one dense layer** achieved validation accuracies between 0.89 and 0.92, showing moderate benefits.

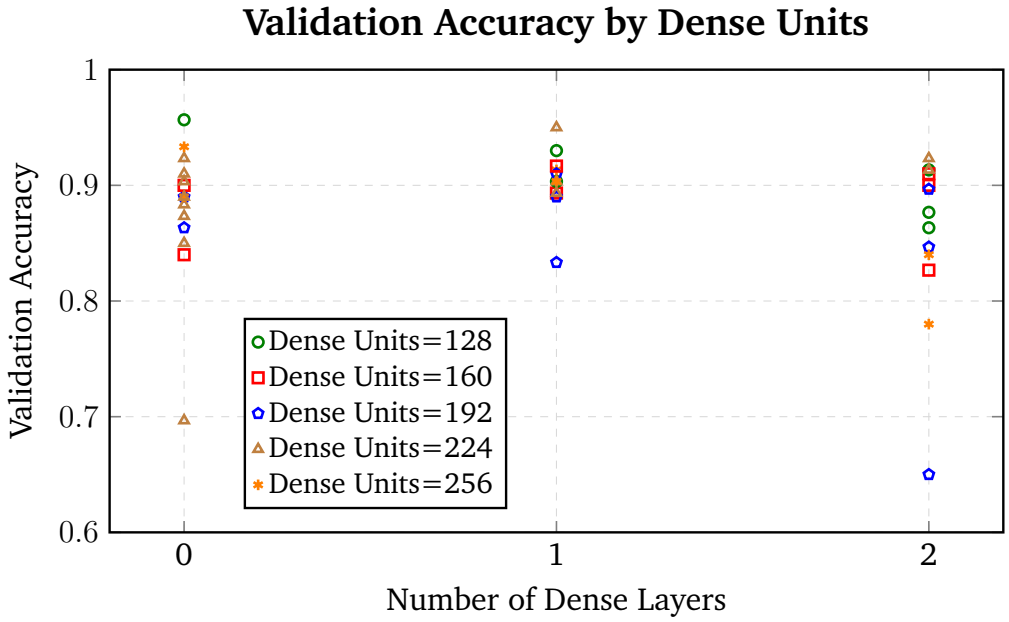
× **Two Dense Layers Introduced Variability**

- Some configurations with **two dense layers** performed well, but others suffered from lower accuracy and potential overfitting.

× **Impact of Dense Units**



(a)



(b)

Fig. 5.12: Validation performance visualized across models with different numbers of dense layers: (a) box plot summary; (b) individual trial results by dense units.

- Models with 128 and 224 dense units showed more consistent performance, whereas models with 256 dense units exhibited fluctuations, possibly due to **over-parameterization**.

These results suggest that **deep fully connected layers are not always necessary for optimal performance**. In fact, **using a single dense layer** yielded better generalization and more consistent accuracy compared to architectures without dense layers or multiple dense layers.

5.6.5 Top-performing Hyperparameter Configurations

To validate the insights from previous analyses, Table 5.2 presents the top-performing hyperparameter configurations. The highest-performing model (trial 00) achieves an accuracy of 0.957, reinforcing several key observations made earlier:

- **3 GRU layers are the dominant choice:** Four out of the five top-performing models utilize three GRU layers, confirming their effectiveness in capturing temporal dependencies. This supports our earlier findings, which suggested that three GRU layers provide an optimal balance between expressiveness and model stability.
- **The combination of 4 convolutional layers, 3 GRU layers, and 128 GRU units appears twice:** This specific configuration, identified earlier as a strong candidate, appears in two of the top five models, further supporting its reliability as a well-balanced setup.
- **3 convolutional layers with 5 GRU layers can also excel:** The highest-scoring model (trial 00) follows a different approach by using three convolutional layers and five GRU layers, demonstrating that deeper recurrent structures can be beneficial in certain cases.
- **Both 32 and 128 GRU units are viable choices:** The best model utilizes 32 GRU units, yet another strong performer (trial 35) employs 128 GRU units. This suggests that both configurations can yield high accuracy when paired with suitable layer depths.
- **Minimal dense layers:** Most top-performing models either avoid additional dense layers or use only one, suggesting that the CNN-GRU structure sufficiently extracts meaningful features without requiring extensive dense processing.

These results further validate the importance of selecting appropriate convolutional and GRU layer depths. The repeated success of **four convolutional, three GRU layers, and 128 GRU units** highlights their collective effectiveness for radar-based gesture classification. This optimal structure is illustrated in Fig. 5.13, which outlines the overall CNN-GRU architecture used in this thesis. Furthermore, the detailed

Tab. 5.2: Top hyperparameter configurations and their performance

	Trial 00	Trial 35	Trial 24	Trial 32	Trial 13
Score	0.957	0.950	0.933	0.930	0.923
No. Conv Layers	3	4	3	5	4
No. GRU Layers	5	3	3	3	3
No. GRU Units	32	128	32	64	128
No. Base Filters	32	16	16	32	32
No. Dense Layers	0	1	0	1	0
No. Dense Units	–	224	–	128	–
Dropout Rate	0.2	0.2	0.2	0.4	0.2
Batch Size	16	64	48	64	48

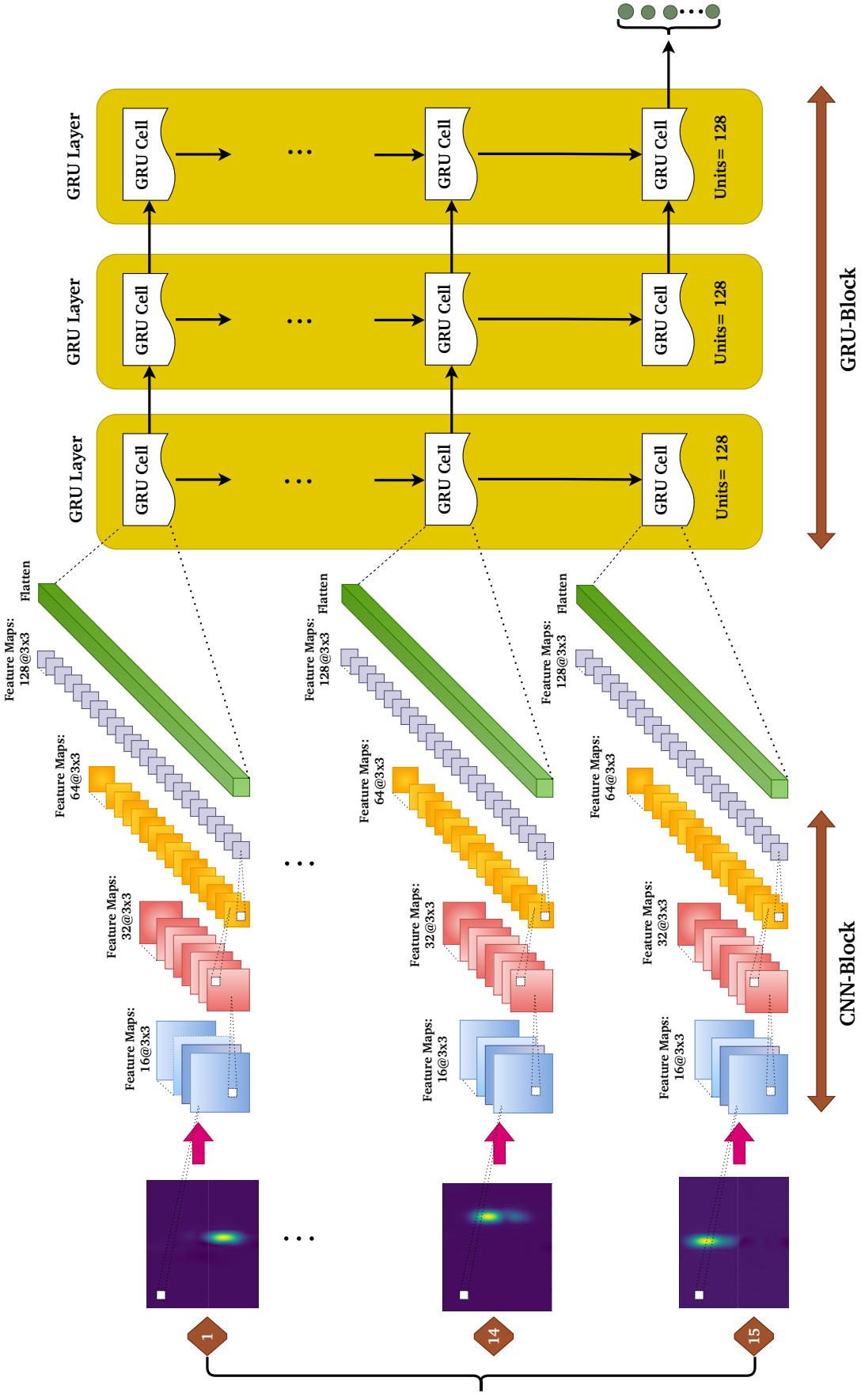


Fig. 5.13: Layer-wise architecture of the final CNN-GRU model, identified through hyperparameter optimization for the radar-based HGR.

Tab. 5.3: Tensor dimension changes through neural network layers for RDM

Layer	Filters	Kernel Size	Stride	Output Shape
Input	–	–	–	$15 \times 65 \times 125$
Conv2D 1	16	3×3	1	$15 \times 65 \times 125 \times 16$
MaxPool 1	–	2×2	2	$15 \times 32 \times 62 \times 16$
Dropout 1	0.2	–	–	$15 \times 32 \times 62 \times 16$
Conv2D 2	32	3×3	1	$15 \times 32 \times 62 \times 32$
MaxPool 2	–	2×2	2	$15 \times 16 \times 31 \times 32$
Dropout 2	0.2	–	–	$15 \times 16 \times 31 \times 32$
Conv2D 3	64	3×3	1	$15 \times 16 \times 31 \times 64$
MaxPool 3	–	2×2	2	$15 \times 8 \times 15 \times 64$
Dropout 3	0.2	–	–	$15 \times 8 \times 15 \times 64$
Conv2D 4	128	3×3	1	$15 \times 8 \times 15 \times 128$
MaxPool 4	–	2×2	2	$15 \times 4 \times 7 \times 128$
Dropout 4	0.2	–	–	$15 \times 4 \times 7 \times 128$
Flatten	–	–	–	15×3584
GRU 1	128	–	–	15×128
GRU 2	128	–	–	15×128
GRU 3	128	–	–	128

Tab. 5.4: Tensor dimension changes through neural network layers for RAM

Layer	Filters	Kernel Size	Stride	Output Shape
Input	–	–	–	$15 \times 65 \times 64$
Conv2D 1	16	3×3	1	$15 \times 65 \times 64 \times 16$
MaxPool 1	–	2×2	2	$15 \times 32 \times 32 \times 16$
Dropout 1	0.2	–	–	$15 \times 32 \times 32 \times 16$
Conv2D 2	32	3×3	1	$15 \times 32 \times 32 \times 32$
MaxPool 2	–	2×2	2	$15 \times 16 \times 16 \times 32$
Dropout 2	0.2	–	–	$15 \times 16 \times 16 \times 32$
Conv2D 3	64	3×3	1	$15 \times 16 \times 16 \times 64$
MaxPool 3	–	2×2	2	$15 \times 8 \times 8 \times 64$
Dropout 3	0.2	–	–	$15 \times 8 \times 8 \times 64$
Conv2D 4	128	3×3	1	$15 \times 8 \times 8 \times 128$
MaxPool 4	–	2×2	2	$15 \times 4 \times 4 \times 128$
Dropout 4	0.2	–	–	$15 \times 4 \times 4 \times 128$
Flatten	–	–	–	15×2048
GRU 1	128	–	–	15×128
GRU 2	128	–	–	15×128
GRU 3	128	–	–	128

progression of tensor dimensions through each layer is documented in Tables 5.3 and 5.4, corresponding to RDM and RAM branches, respectively.

5.7 Validation of Model Generalization through LOOCV

Following the hyperparameter optimization phase, which identified an optimal configuration of four convolutional layers, three GRU layers, and 128 GRU units for radar-based gesture classification, this section rigorously evaluates the generalization performance of the proposed dual-stream CNN-GRU model through Leave-One-Out Cross-Validation (LOOCV). The hyperparameters from trial 35, selected for their robust validation accuracy, were applied to both the CNN-GRU model and a comparative CNN-LSTM model to substantiate the architectural choice of GRUs over LSTMs. This comparative analysis not only validates the efficacy of the chosen configuration but also contributes to the understanding of recurrent neural architectures in the context of temporal sequence modeling for radar-based gesture recognition.

5.7.1 Motivation for LOOCV and Comparative Analysis

Given the limited number of subjects in the radar-based gesture dataset, LOOCV is employed to ensure robust generalization across diverse gesture variations. By iteratively designating each subject as the test set while training on the remaining subjects, LOOCV captures inter-subject variability-a critical factor for deploying gesture recognition systems in real-world scenarios where user-specific gesture patterns may differ significantly. The hyperparameters from trial 35, consisting of four convolutional layers, three recurrent layers (GRU or LSTM) with 128 units, 16 base filters, one dense layer with 224 units, a dropout rate of 0.2, and a batch size of 64, were applied to both the CNN-GRU and CNN-LSTM models. This comparison elucidates the theoretical and practical advantages of GRUs over LSTMs, particularly in terms of temporal modeling and generalization, thus reinforcing the architectural decisions made during the hyperparameter optimization phase.

5.7.2 Methodology

The LOOCV methodology was executed systematically as follows:

1. **Hyperparameter Retrieval**: The hyperparameters from trial 35 were retrieved from the results of a prior random search hyperparameter tuning experiment, ensuring consistency across both models.
2. **Data Splitting**: For each subject, the dataset was partitioned into a training set (comprising all subjects except the test subject) and a test set (the designated subject). RDMs and RAMs were reshaped to align with the input dimensions required by the model architecture.
3. **Model Configuration**: Both the CNN-GRU and CNN-LSTM models were constructed using the trial 35 hyperparameters.
4. **Training**: Each model was trained for a maximum of 60 epochs, employing several optimization strategies:
    - **Custom Model Checkpointing**: Models were saved whenever validation loss improved or test accuracy increased by at least 2%, ensuring the retention of optimal weights.

Tab. 5.5: Comparison of LOOCV performance metrics for CNN-GRU and CNN-LSTM models using trial 35 hyperparameters.

Metric (Mean ± Std. Dev.)	CNN-GRU	CNN-LSTM
Test Accuracy	$91.19\% \pm 7.57\%$	$89.31\% \pm 8.64\%$
Test Loss	0.423 ± 0.318	0.566 ± 0.422
Precision	$93.14\% \pm 5.50\%$	$90.90\% \pm 7.90\%$
F1-Score	$91.34\% \pm 7.80\%$	$88.93\% \pm 9.68\%$

- **Early Stopping**: Training was halted if validation loss failed to improve over a predefined number of epochs, with the best weights restored to prevent overfitting.
- **Learning Rate Reduction**: The learning rate was dynamically reduced on plateaus to facilitate convergence.

5. **Evaluation**: The best model for each fold was evaluated on the test set, computing metrics including test loss, accuracy, precision, and F1-score.

6. **Result Storage and Comparison**: Performance metrics for each fold were recorded. This approach ensured a fair and reliable comparison across the two architectures.

The sole architectural difference between the models was the type of recurrent layer (GRU vs. LSTM), allowing for a direct assessment of their respective impacts on performance.

5.7.3 LOOCV Performance Results

The LOOCV results for the CNN-GRU and CNN-LSTM models, evaluated across 8 subjects, are summarized in Table 5.5. The CNN-GRU model achieved higher overall performance, with better mean accuracy, precision, and F1-score, as well as lower test loss, compared to the CNN-LSTM counterpart.

The average confusion matrix for the CNN-GRU model, shown in Fig. 5.14, demonstrates strong classification performance with high diagonal dominance across most gesture classes. Several gestures achieved particularly high accuracy, such as PL with 97.08%, PLPS with 97.50%, and PSPL with 92.92%. Misclassifications were relatively minor and mostly occurred between gestures with overlapping spatial or temporal features. For instance, gesture PC was misclassified as PO 12.92% of the time and as CCW 1.67%, while gesture SR showed a 8.33% confusion rate with CW. These results highlight the CNN-GRU model's robustness, though subtle similarities in motion trajectories (as reflected in RDM and RAM features) occasionally led to errors.

The average confusion matrix for the CNN-LSTM model, depicted in Fig. 5.15, indicates a decline in classification accuracy for certain gestures compared to the CNN-GRU model. Gestures such as SR, PO, and PC achieved lower accuracies of 83.33%, 87.08%, and 82.50%, respectively, with notable misclassifications. Specifically, gesture PC was confused with PO 11.67% of the time, and gesture CW with SR at

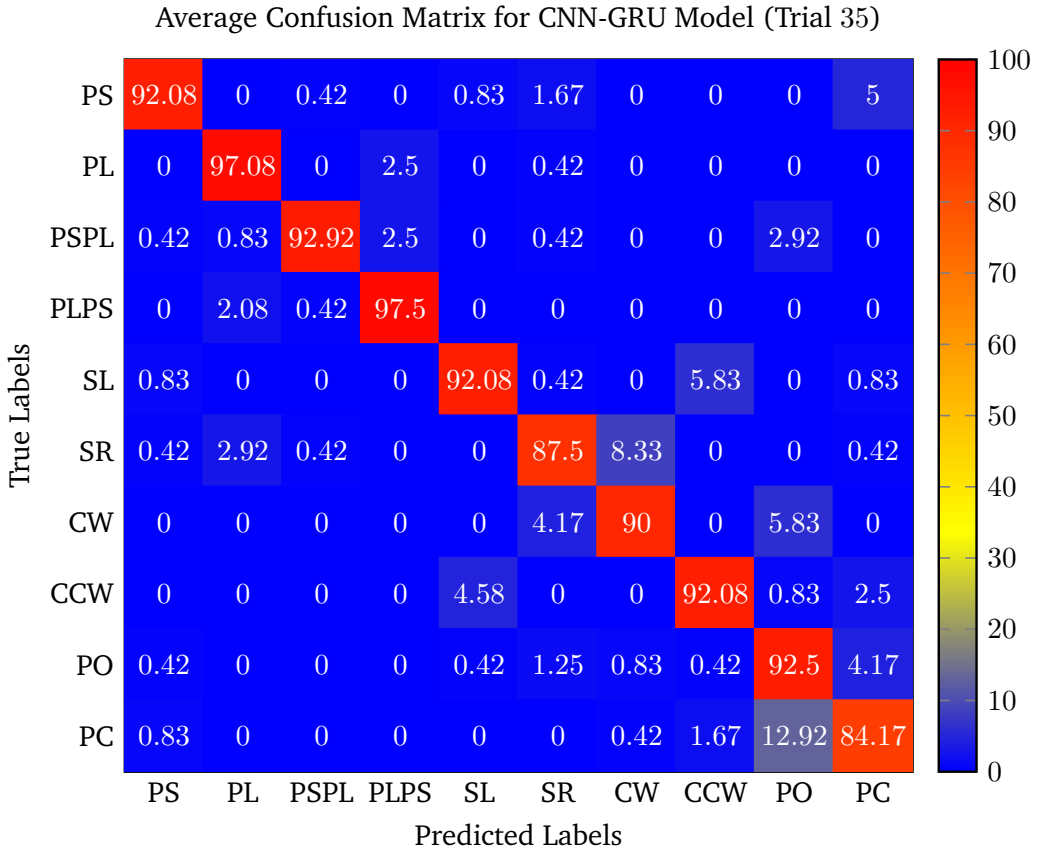


Fig. 5.14: Average confusion matrix for LOOCV with the CNN-GRU model using trial 35 hyperparameters (see Table 5.2), illustrating classification performance across 10 gesture classes.

10.83%. Although the model maintains solid performance for simpler gestures like PLPS (97.92%) and PL (95.79%), the increased off-diagonal values for complex gestures suggest limitations in temporal modeling capacity compared to the GRU-based approach.

5.7.4 Per-Class Performance Analysis

To provide deeper insights into model behavior, per-class accuracies derived from the confusion matrices are presented in Table 5.6 and visualized in Fig. 5.16. The CNN-GRU model consistently outperforms CNN-LSTM across nearly all gesture classes. Notable improvements are seen for gestures SR (87.50% vs. 83.33%), PO (92.50% vs. 87.08%), and CW (90.00% vs. 85.83%), indicating the GRU's superior ability to capture complex temporal dependencies.

Analysis of Challenging Gestures

The most challenging gestures for both models are SR, PO, and PC. While the CNN-GRU model yields higher accuracies-87.50%, 92.50%, and 84.17% for SR, PO, and PC, respectively-the CNN-LSTM model lags behind, achieving 83.33%, 87.08%, and 82.50% on the same classes. These gestures are difficult to classify for multiple

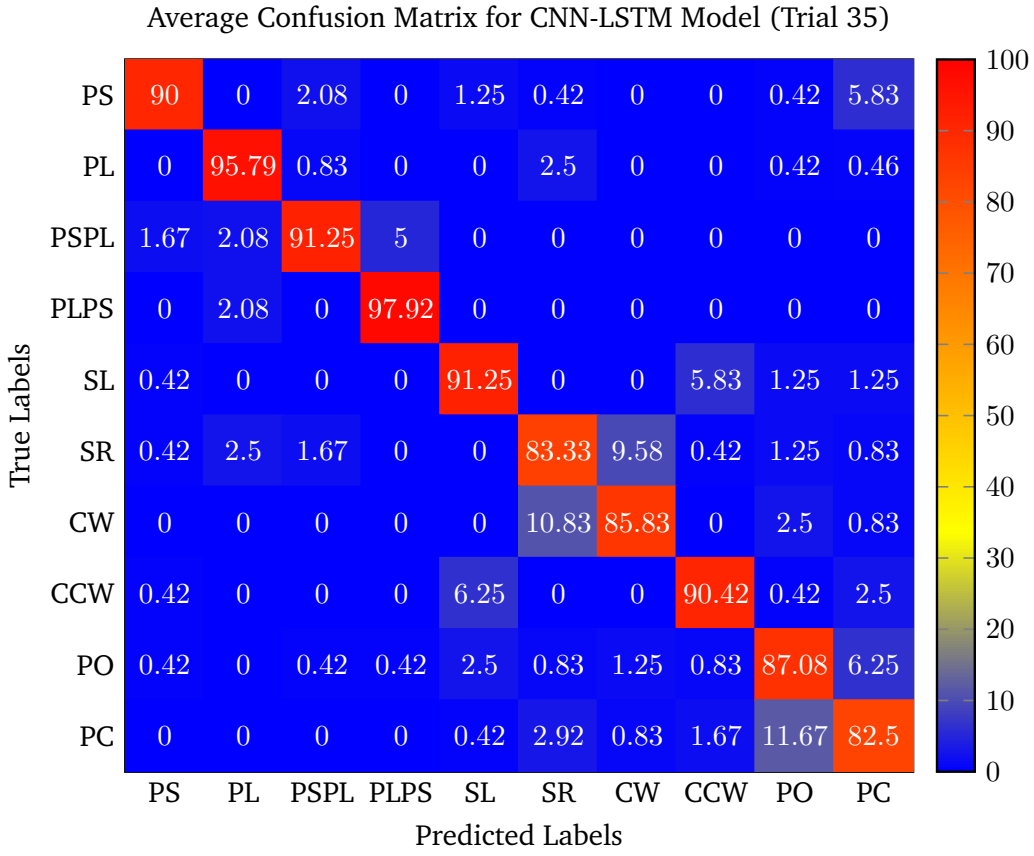


Fig. 5.15: Average confusion matrix for LOOCV with the CNN-LSTM model using trial 35 hyperparameters (see Table 5.2), illustrating classification performance across 10 gesture classes.

reasons. First, they involve subtle and complex kinematic patterns. SR (Swipe Right) involves rapid unidirectional hand movement, while PO (Pinch Open) and PC (Pinch Close) are fine-motor gestures primarily involving finger motion, with minimal displacement of the whole hand. These characteristics result in radar returns with limited inter-class variability, making them harder to distinguish using range-Doppler or range-angle features alone.

Second, the challenge is compounded by the radar system's sensing geometry. The radar is positioned horizontally on a tabletop, facing upward, with a 1D antenna array aligned along the X-axis. While this configuration enables effective estimation of Doppler shifts and range information, it lacks angular resolution along the Y-axis, which is the direction perpendicular to the antenna array, as depicted in Fig. 5.17. Consequently, it is highly sensitive to motion along the X-axis (left-right), but only weakly responsive to motion along the Y-axis (forward-backward from the radar's perspective)

This has important implications for gestures performed in the horizontal XY plane at a fixed height above the radar. In particular, circular gestures such as CW (Clockwise) and CCW (Counter-Clockwise), which involve motion along both X and Y axes, produce radar signatures dominated by the X-component. The Y-axis segments of the motion contribute weak Doppler shifts, resulting in incomplete or distorted temporal features. Consequently, such circular motions are often confused with linear

Tab. 5.6: Per-class accuracies derived from the average confusion matrices for CNN-GRU and CNN-LSTM models using trial 35 hyperparameters.

Gesture	CNN-GRU Accuracy (%)	CNN-LSTM Accuracy (%)
PS	92.08	90.00
PL	97.08	95.79
PSPL	92.92	91.25
PLPS	97.50	97.92
SL	92.08	91.25
SR	87.50	83.33
CW	90.00	85.83
CCW	92.08	90.42
PO	92.50	87.08
PC	84.17	82.50

swipes like SR (Swipe Right) or SL (Swipe Left), which produce strong, unidirectional Doppler responses. For example, the misclassification rate of SR to CW is 8.33% in GRU and 9.58% in LSTM. Similarly, misclassifications from PC to PO occur at rates of 12.92% and 11.67%, respectively, as these gestures involve fine finger motions that result in subtle variations in RDM and RAM.

Although both models struggle with these ambiguous gesture classes, the CNN-GRU demonstrates greater resilience. Its higher class-wise accuracies and reduced confusion rates suggest a more effective temporal modeling of subtle motion transitions

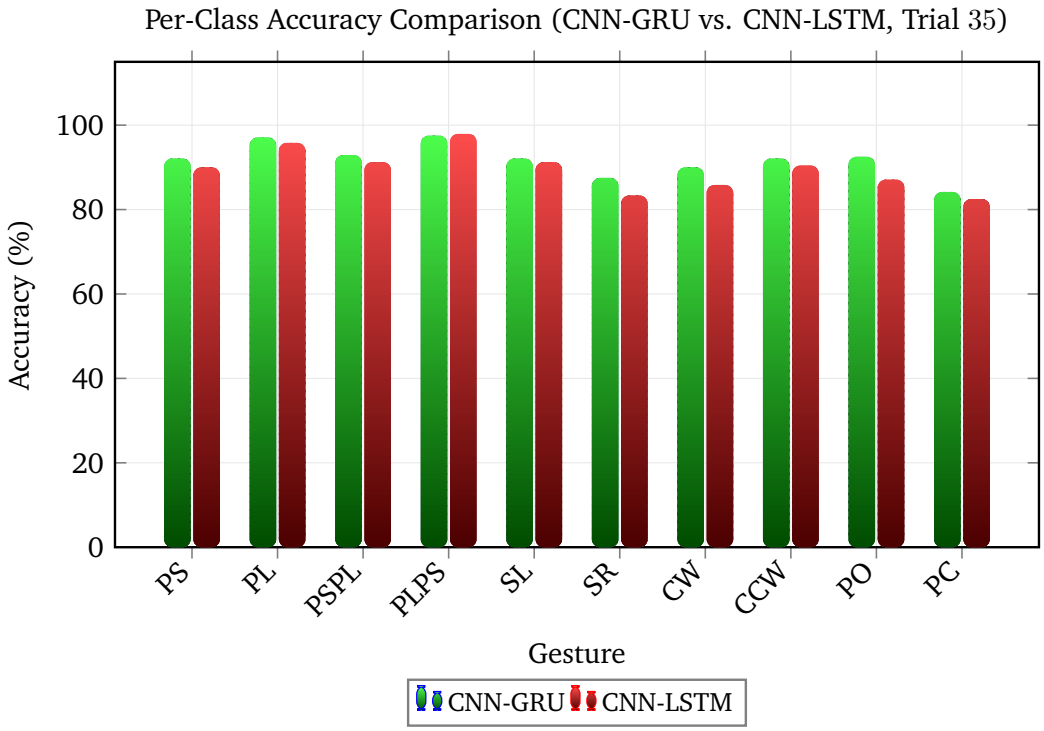


Fig. 5.16: Per-class accuracy comparison between CNN-GRU and CNN-LSTM models using trial 35 hyperparameters, highlighting CNN-GRU's superior performance across most gestures.

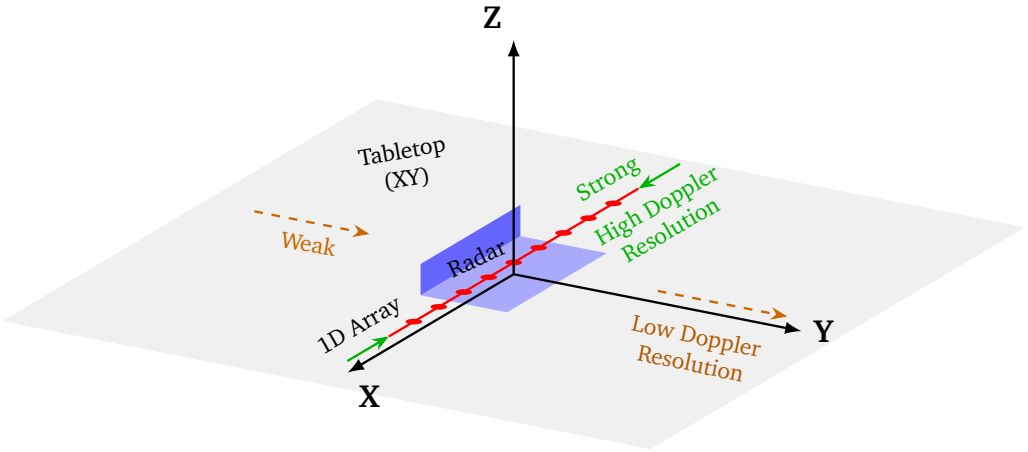


Fig. 5.17: Illustration of the radar system's sensing geometry and Doppler sensitivity. The radar is placed horizontally on a tabletop, facing upward (along the Z-axis), with a 1D antenna array aligned along the X-axis. Doppler sensitivity is strongest along the antenna array direction (green arrows) and weakest in the orthogonal direction (orange dashed arrows).

within the RDM and RAM sequences.

5.7.5 Subject-Wise Performance Analysis

Subject-wise evaluation, as shown in Fig. 5.18, indicates that the CNN-GRU model delivers consistently high test accuracy across all 8 subjects, with test accuracies predominantly above 90%. Notably, subject 6 achieved a perfect accuracy of 100%, while subject 0 recorded 98.30%. However, subjects 1 and 7 exhibited relatively lower accuracies of 77.30% and 82%, respectively. This variability suggests potential differences in gesture execution or radar signal quality among subjects. In contrast, CNN-LSTM exhibits greater variability, with accuracy dipping to 75.30% for subject 1 and only marginally outperforming CNN-GRU for subject 0 (99.30% vs. 98.30%). CNN-GRU's lower standard deviation (7.57%) compared to CNN-LSTM (8.64%) underscores its robustness to subject-specific variations in gesture performance or radar reflections.

5.7.6 Training Dynamics and Generalization

To further illustrate the superior performance of the CNN-GRU model, a detailed analysis of the training and validation metrics of the subject 6 is presented. Fig. 5.19 and Fig. 5.20 display the training and validation accuracy and loss for the CNN-GRU and CNN-LSTM models, respectively, over 60 epochs. The GRU model (Fig. 5.19) demonstrates superior performance compared to the LSTM model (Fig. 5.20) across all metrics over 60 epochs. GRU achieves a higher initial validation accuracy (73.67% vs. 47.33%) and consistently reaches 99–100% after epoch 20, peaking at 100% multiple times, while LSTM stabilizes at 97–99%, peaking at 99.33%. GRU's training accuracy also converges faster to $95\%+$ (epoch 5 vs. epoch 6) and peaks at 100%, compared to LSTM's 99.95%. Additionally, GRU exhibits lower training and validation losses, stabilizing at 0.09–0.14 and 0.10–0.20 respectively, compared to LSTM's

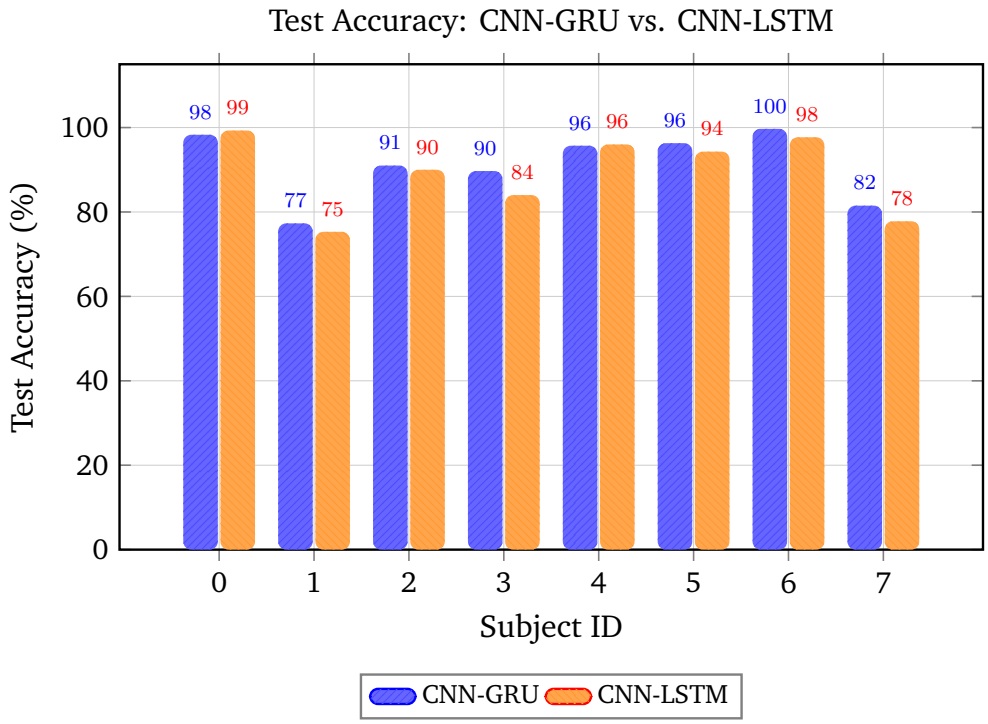


Fig. 5.18: Test accuracy comparison for CNN-GRU and CNN-LSTM models across subjects using LOOCV with trial 35 hyperparameters.

0.11–0.13 and 0.14–0.24, with final values of 0.088 and 0.108 for GRU versus 0.116 and 0.148 for LSTM, indicating better convergence and generalization for the GRU model.

Advantages of CNN-GRU over CNN-LSTM

The CNN-GRU's superior performance can be attributed to several theoretical and practical advantages over the CNN-LSTM model, particularly in the context of radar-based gesture recognition:

- **Computational Efficiency**: GRUs, with their simplified two-gate structure (update and reset gates), possess fewer parameters than LSTMs, which employ three gates (input, forget, and output gates)[2]. This reduction in complexity translates to faster training times within the constrained 60-epoch limit, a critical consideration for radar datasets where computational resources may be limited.

- **Effective Temporal Modeling**: GRUs mitigate the vanishing gradient problem more efficiently than LSTMs in this application, enabling better capture of subtle temporal dependencies in RDMs and RAMs. This is evidenced by the CNN-GRU's higher class-specific accuracies for complex gestures (e.g., 90% for CW compared to 85.83% for CNN-LSTM), where temporal dynamics are paramount.

[2] See Appendix C for architectural details.

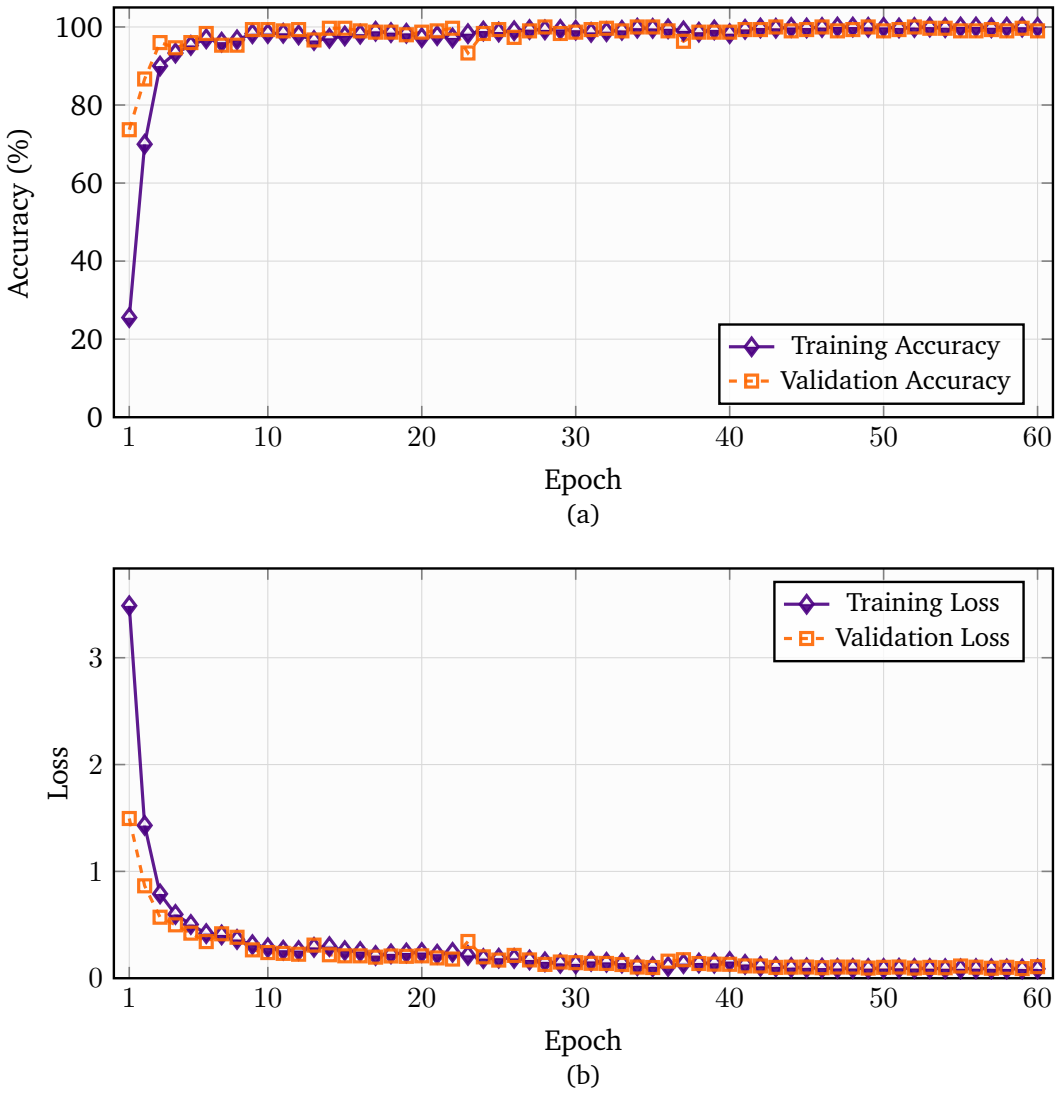


Fig. 5.19: Training and validation accuracy and loss metrics for the CNN-GRU model over 60 epochs, derived from a LOOCV iteration for subject 6.

- **Robust Generalization**: The CNN-GRU model exhibits slightly lower variability in performance metrics (e.g., 5.50% standard deviation in precision) compared to the CNN-LSTM model (7.90%). GRU also achieves a higher mean precision (93.14% vs. 90.90%) and a marginally lower mean test loss (0.423 vs. 0.566), as shown in Table 5.5. Overall, GRU demonstrates slightly better generalization and consistency, but the performance gap is not substantial.

These findings align with prior research on recurrent neural networks, where GRUs have been shown to offer a favorable trade-off between computational efficiency and modeling capability for sequence tasks with moderate complexity [96].

5.7.7 Future Directions

Several avenues for future research emerge from this analysis. First, the dropout rate (0.2) and base filters (16) could be further optimized to reduce variability for outlier subjects (e.g., subjects 1 and 7 in the CNN-GRU model). Techniques such as spatial

dropout or dynamic dropout [97] rates may enhance regularization. Second, data augmentation strategies, such as synthetic radar signal generation or perturbation of RDMs and RAMs , could address inter-subject variability and improve performance on challenging gestures [77]. Third, exploring hybrid architectures that combine GRUs and LSTMs may leverage the strengths of both, potentially mitigating the CNN-LSTM model's high variability while retaining the CNN-GRU's efficiency. Finally, incorporating multi-modal data (e.g., radar with inertial sensors) or transfer learning could improve the model's adaptability to new subjects or radar hardware, further enhancing its practical utility.

5.7.8 Conclusion

This chapter presented an evaluation of a dual-stream 2D CNN-GRU model for radar-based HGR, leveraging pre-processed RDMs and RAMs derived from INRAS RBK2, a 77 GHz MIMO FMCW radar with 31 virtual receive elements achieved throug TDM.

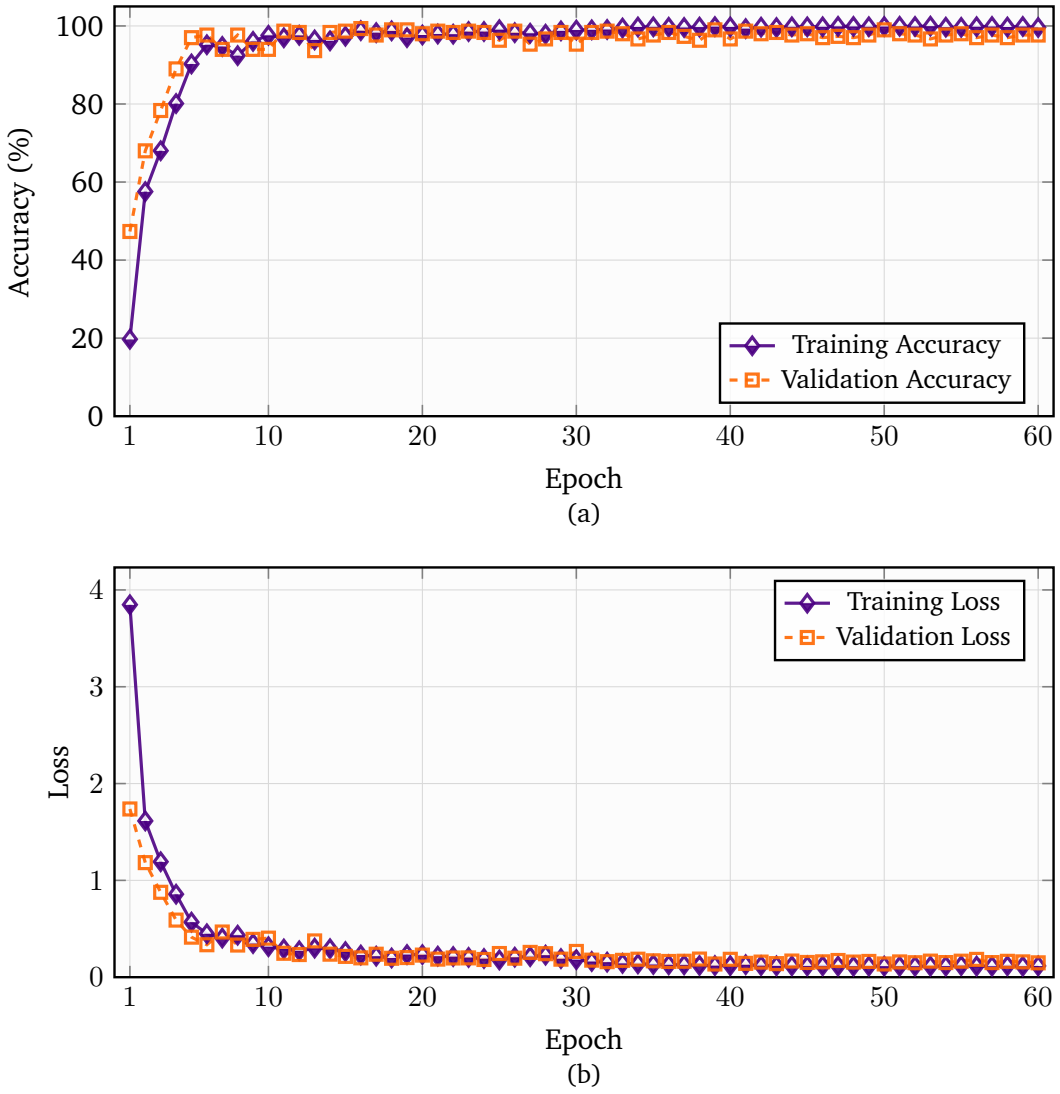


Fig. 5.20: Training and validation accuracy and loss metrics for the CNN-LSTM model over 60 epochs, derived from a LOOCV iteration for subject 6.

Addressing a significant gap in the literature, this study is the first to explore the feasibility of such an architecture compared to state-of-the-art works that relied on smaller antenna arrays (e.g., 2 TX and 4 RX). The primary objective was to compare the performance of the CNN-GRU model against a CNN-LSTM baseline, focusing on their ability to model temporal sequences, generalize across subjects, and achieve robust gesture recognition results.

The methodology employed LOOCV to assess the performace of the model across diverse subjects. This method offers a more rigorous evaluation than the k-fold cross-validation commonly used in the literature. The dual-stream CNN-GRU architecture, consisting of four convolutional layers, three GRU layers, and 128 GRU units, processes RDMs and RAMs to capture both temporal and spatial features of gestures. Regularization techniques, including standard dropout applied after convolutional and within recurrent layers, along with L2 regularization in the GRU layers, were utilized to enhance model robustness and prevent overfitting. Aggregated LOOCV results (Table 5.5) confirmed the CNN-GRU model's slight edge, achieving a mean test accuracy of $91.19\% \pm 7.57\%$ compared to the CNN-LSTM model's $89.31\% \pm 8.64\%$, a difference of 1.88%. The CNN-GRU model also exhibited lower variability in metrics such as precision (5.50% vs. 7.90% standard deviation) and test loss (0.423 vs. 0.566), underscoring its consistency and generalization capabilities. These findings position the CNN-GRU model as a robust and efficient solution for real-world gesture recognition systems, contributing to the advancement of non-contact HCI technologies.

CHAPTER 6

Conclusion and Outlook

6.1 Conclusion

This dissertation explored the transformative potential of radar-based sensing using Radarbook2 (RBK2), a 77 GHz Multiple-Input Multiple-Output (MIMO) Frequency Modulated Continuous Wave (FMCW) radar equipped with 31 virtual receive elements, achieved through Time Division Multiplexing (TDM) access technique. This research was conducted across two distinct yet complementary domains: vital signs monitoring and Hand Gesture Recognition (HGR). By leveraging the enhanced spatial resolution of this advanced radar system, the research addressed critical challenges in non-contact health monitoring and Human-Computer Interaction (HCI), contributing to the development of robust, efficient, and practical radar-based solutions.

The first section focused on monitoring vital signs, specifically Breathing Rate (BR) and Heart Rate (HR), from human participants. Employing Principal Component Analysis (PCA) on aggregated Range-Angle Map (RAM), the proposed algorithm effectively separated signal from noise in slow-time phase data. It achieved a low average Mean Absolute Percentage Error (MAPE) of 1.06% for HR and 3.27% for BR, significantly outperforming conventional techniques, which yielded MAPE values of 4.50% for HR and 6.44% for BR when compared to the EQ02+ LifeMonitor, a cutting-edge wearable sensor [54]. This high accuracy underscores the potential of radar-based systems as a non-invasive alternative to traditional wearable devices, offering significant advantages in scenarios where physical contact is impractical or undesirable, such as in medical settings or remote monitoring applications.

The second part of the dissertation evaluated a dual-stream architecture that integrates Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU) for radar-based HGR. It utilized pre-processed Range-Doppler Map (RDM) and

RAM to capture temporal and spatial features of gestures. Through rigorous Leave-One-Out Cross-Validation (LOOCV), the CNN-GRU model achieved a mean test accuracy of $91.19\% \pm 7.57\%$, slightly outperforming the Long Short-Term Memory (LSTM) model's $89.31\% \pm 8.64\%$ by 1.88%. The CNN-GRU model demonstrated faster convergence (reaching 95% training accuracy by epoch 5 compared to epoch 6 for CNN-LSTM), higher validation accuracy (99% vs. 97.67%), and lower validation loss (0.108 vs. 0.148). Per-subject analysis further confirmed its consistency, with GRU outperforming LSTM on 6 out of 8 subjects, and showing less variations in metrics such as precision (5.50% vs. 7.90% standard deviation) and test loss (0.423 vs. 0.566).

A notable difficulty in the HGR analysis involved distinguishing certain intricate gestures, notably Swipe Right (SR), Pinch Open (PO), and Pinch Close (PC), due to their nuanced movements and minimal differences in radar signatures. The CNN-GRU model achieved accuracy levels of 87.50%, 92.50%, and 84.17% for SR, PO, and PC, respectively, compared to the CNN-LSTM model's 83.33%, 87.08%, and 82.50%. These gestures pose challenges because of their delicate finger motions and rapid directional shifts, which generate radar data with subtle distinctions that are difficult to differentiate using RDMs or RAMs. Additionally, the radar's placement, mounted horizontally on a tabletop with a 1D antenna array oriented along the X-axis, enhances sensitivity to lateral movements but offers limited detection capability along the Y-axis. This setup leads to diminished signal strength for motions perpendicular to the array, causing overlap between circular motions (e.g., Clockwise [CW], Counter-Clockwise [CCW]) and straight swipes (e.g., SR, Swipe Left [SL]). For instance, errors in identifying SR as CW reached 8.33% with GRU and 9.58% with LSTM, while misclassifications between PC and PO were observed at 12.92% and 11.67%, respectively.

Together, these studies demonstrate the versatility of radar-based sensing in detecting fine-grained human activities, from physiological signals to complex gestures. The use of INRAS RBK2 enabled both applications to benefit from enhanced spatial resolution, allowing for precise detection of subtle movements and vital signs. By integrating advanced signal processing (PCA for vital signs) and machine learning (CNN-GRU for HGR), this research has advanced the state-of-the-art in non-contact sensing technologies, paving the way for innovative applications in healthcare, smart environments, and interactive systems.

6.2 Future Work

The findings from this dissertation open several avenues for future research, building on the potential improvements identified in both studies to further enhance the accuracy, reliability, and applicability of radar-based sensing systems. The following directions aim to address the limitations of the current work and explore new opportunities for integrating these technologies into practical, real-world scenarios.

- **Advanced Signal Processing and Noise Reduction:** Both vital signs monitoring and HGR systems could benefit from more sophisticated signal processing techniques to filter out noise and artifacts. For vital signs, techniques such as adaptive filtering or wavelet-based denoising could further improve the

Signal-to-Noise Ratio (SNR), potentially reducing the MAPE for BR and HR measurements below the current 3.27% and 1.06%. In HGR, enhanced preprocessing of RDMs and RAMs as well as a more sophisticated hand activity detector module could mitigate environmental noise, improving gesture recognition accuracy, especially in challenging settings with interference.

- **Advanced Machine Learning Techniques:** The application of more advanced machine learning algorithms presents significant potential for both domains. For vital signs monitoring, deep learning models, such as those used in HGR (e.g.,CNN-GRU or CNN-LSTM), could be adapted to improve the estimation of BR and HR by learning complex patterns in radar signals. In HGR, exploring hybrid architectures (e.g., combining GRUs and LSTMs) or advanced regularization techniques, such as spatial dropout (which drops entire feature maps to handle spatial correlations) or dynamic dropout (which adapts dropout rates during training), could enhance generalization and robustness, potentially addressing the modest performance gap between CNN-GRU and CNN-LSTM models.

- **Multi-Sensor Fusion:** Combining radar data with data from other sensors, such as cameras or Inertial Measurement Units (IMUs), could significantly improve overall system performance. In vital signs monitoring, multi-sensor fusion could provide complementary data (e.g., visual indicators of breathing patterns) to validate radar-based measurements, enhancing reliability in dynamic environments. For HGR, integrating radar with camera-based systems could enable more robust gesture detection by leveraging both spatial (radar) and visual (camera) information, particularly in scenarios with occlusions or complex backgrounds.

- **Scalability and Robustness Testing:** Both systems could benefit from scalability studies using larger and more diverse datasets. In vital signs monitoring, testing the PCA-based algorithm across a broader population (e.g., different age groups, health conditions) could validate its generalizability. For HGR, scaling the dataset to include more subjects, gestures, and environmental conditions could improve the CNN-GRU model's robustness. Additionally, investigating the impact of varying radar configurations (e.g., different numbers of virtual elements or frequencies) could provide insights into optimizing radar hardware for specific applications.

- **Hybrid Applications:** The complementary nature of vital signs monitoring and HGR suggests potential for hybrid applications. For instance, integrating gesture recognition with vital signs monitoring could enable smart healthcare systems, allowing hand movements to operate medical equipment while continuously monitoring BR and HR. Such systems could be particularly valuable in telemedicine, where non-contact interaction and monitoring are essential for patient comfort and safety. Additionally, this integration can extend to automotive contexts, where in-vehicle vital sign monitoring could improve driver safety by detecting stress or fatigue levels and potentially triggering alerts.

- **Practical Deployment:** Future work could focus on deploying these systems on resource-constrained devices, such as embedded radar platforms, and evaluating their performance in real-world settings like in-vehicle environments or smart homes. This will be crucial for widespread adoption.

By pursuing these directions, future research can build on the foundational work presented in this dissertation, further advancing the field of radar-based sensing and its applications in non-contact health monitoring and HCI.

Appendices

APPENDIX A

MIMO Radar Signal Processing

A.1 MIMO Automotive Radar Signal Notation

Multiple-Input Multiple-Output (MIMO) radar is a key technology in modern automotive radar systems, enhancing range, Doppler, and angle resolution by utilizing multiple transmit and receive antennas. This appendix establishes a formal notation for MIMO radar signals using a structured 3D tensor representation.

A.1.1 MIMO Radar System Model

A MIMO radar system consists of N_{TX} transmit antennas (TX) and N_{RX} receive antennas (RX), forming a virtual array of size:

$$L = N_{\text{TX}} \times N_{\text{RX}}. \tag{A.1}$$

Each TX antenna transmits a waveform, and the reflected signals are received by all RX antennas, forming a multi-dimensional received signal. The received signal is structured along three key dimensions:

- **Fast-time samples** (n): Used for range estimation.
- **Virtual channels** (l): TX-RX antenna pairs, used for Angle-of-Arrival (AoA) estimation.
- **Slow-time samples (chirps per frame)** (m): Used for Doppler estimation.

This naturally forms a **3D radar data cube**, represented as:

$$\mathbf{Y} \in \mathbb{C}^{N \times L \times M}, \tag{A.2}$$

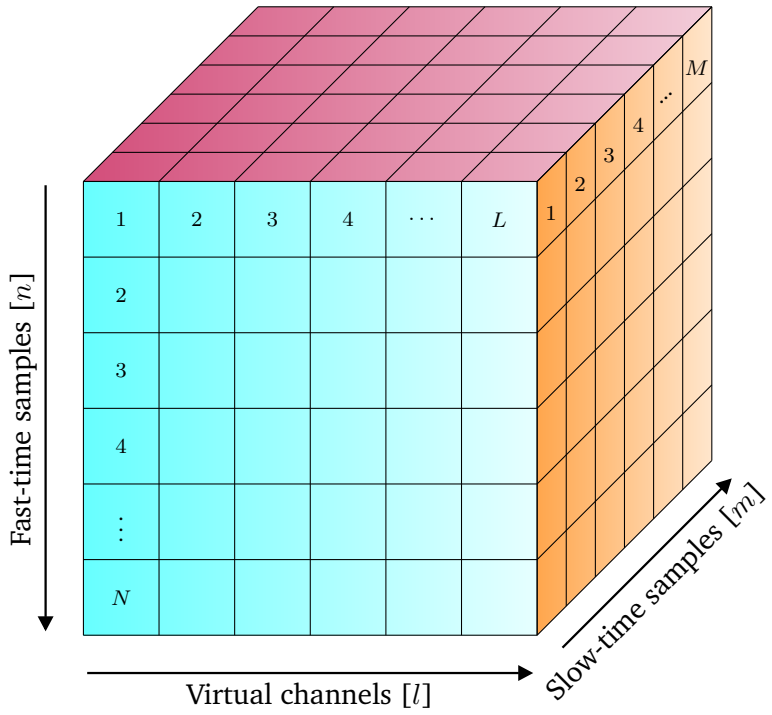


Fig. A.1: 3D Representation of MIMO radar data cube

where:

- N is the number of fast-time samples (range bins).
- L is the number of virtual channels ($L = N_{\text{TX}} \times N_{\text{RX}}$).
- M is the number of slow-time samples.

A visual representation of this **3D data cube structure** is provided in Fig. A.1, which illustrates the relationships between fast-time, virtual channels, and slow-time samples.

A.1.2 Mathematical Representation of the Received Signal

For a fixed chirp index m, the received signal matrix is given by:

$$\mathbf{Y}^{(m)} = \begin{bmatrix} y^{(m)}(1,1) & y^{(m)}(1,2) & \dots & y^{(m)}(1,L) \\ y^{(m)}(2,1) & y^{(m)}(2,2) & \dots & y^{(m)}(2,L) \\ \vdots & \vdots & \ddots & \vdots \\ y^{(m)}(N,1) & y^{(m)}(N,2) & \dots & y^{(m)}(N,L) \end{bmatrix} \tag{A.3}$$

where:

- Rows (n) correspond to fast-time samples (range bins).
- Columns (l) correspond to virtual channels (TX-RX pairs).
- m is fixed, representing a single chirp signal.

Since M chirps are transmitted, the full **3D data cube** consists of M such matrices stacked along the slow-time dimension:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}^{(1)} \\ \mathbf{Y}^{(2)} \\ \vdots \\ \mathbf{Y}^{(M)} \end{bmatrix} \tag{A.4}$$

Thus, the full data cube can be compactly written as:

$$\mathbf{Y}^{(n,l,m)} \in \mathbb{C}^{N \times L \times M}. \tag{A.5}$$

A.1.3 Processing Interpretation of the Data Cube

The 3D data cube structure aligns with the typical MIMO radar signal processing pipeline:

1. **Range Processing (FFT on N)**: Performs a Fast Fourier Transform (FFT) on fast-time samples to extract range information.
2. **Angle Estimation (Beamforming on L)**: Computes AoA using virtual array data.
3. **Doppler Processing (FFT on M)**: Applies an FFT on slow-time samples to estimate the radial velocity of targets.

This structured representation facilitates matrix-based algorithms such as:

- **Matrix Decomposition Techniques**: Such as Singular Value Decomposition (SVD) for noise reduction and target detection.
- **Beamforming Algorithms**: For enhanced angle estimation and spatial resolution.
- **Doppler Filtering**: To isolate and analyze target velocity profiles.

APPENDIX B

Radarbook 2

B.1 Hardware Description

The Radarbook2 (RBK2), manufactured by INRAS GmbH is a microwave radar evaluation platform utilized in this thesis for radar signal processing and experimental analysis. Designed for research and development, it features a high-speed Ethernet interface for efficient data transfer and a Field Programmable Gate Array (FPGA) for real-time signal processing. The inclusion of a dual-core ARM processor further enables the execution of custom algorithms directly on the platform, supporting rapid prototyping for experimental setups [46].

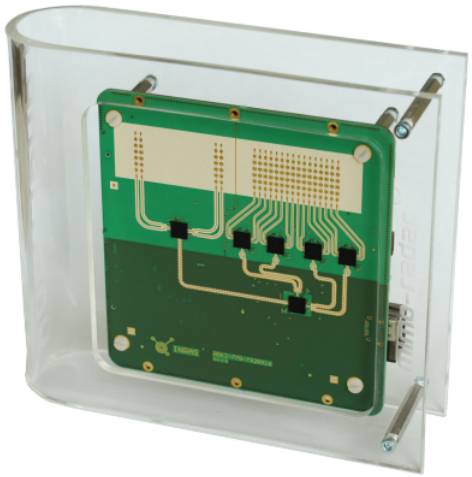


Fig. B.1: Radarbook2 evaluation platform [46]

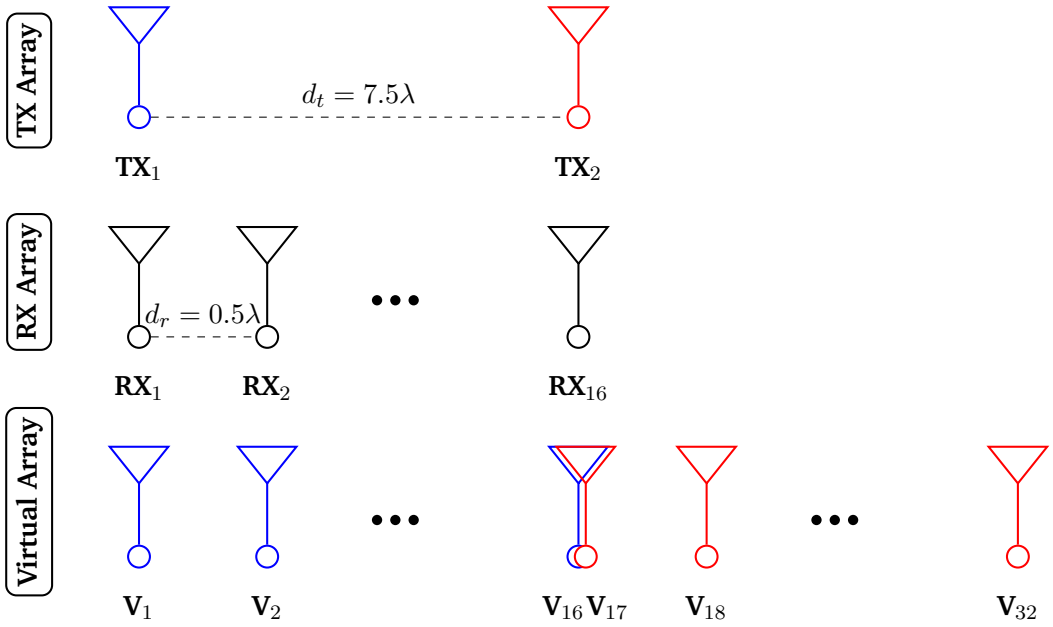


Fig. B.2: Array configuration showing TX array, RX array, and virtual array with inter-element spacing $d_t = 7.5\lambda$ and $d_r = 0.5\lambda$.

The RBK2 supports RF frontends operating at 24 GHz and 77 GHz, accommodating various radar configurations such as Doppler, range-Doppler, and multi-channel MIMO systems.

B.1.1 MIMO Antenna Configurations with TDMA

The RBK2's MIMO radar capabilities were leveraged in this research using a configuration with 2 Tx and 16 Rx channels, based on a Time Division Multiplexing (TDM) scheme. The antenna array was designed with an inter-element spacing of $d_t = 7.5\lambda$ for the TX antennas and $d_r = 0.5\lambda$ for the Rx antennas, where λ denotes the wavelength at the operating frequency. The layout of the antenna elements and their spacing is illustrated in Fig. B.2.

To enable multiple-input operation without interference, TDM was applied at the transmitter side. Specifically, the 2 Tx antennas transmit in separate, non-overlapping time slots, allowing the Rx channels to independently capture the responses from each transmitter. This TDM operation scheme is illustrated in Fig. B.3. By combining each transmit and receive antenna pair, a virtual array is formed, significantly increasing the effective aperture and angular resolution of the system. The positions of the virtual array are calculated as the sum of the Tx and Rx antenna positions. Specifically, the antenna positions are defined as follows:

- **Tx positions**: $T_m = (m-1) \cdot 7.5\lambda$, where $m = 1, 2$.
  - $T_1 = 0\lambda, T_2 = 7.5\lambda$.
- **Rx positions**: $R_n = (n-1) \cdot 0.5\lambda$, where $n = 1, 2, \ldots, 16$.
  - $R_1 = 0\lambda, R_2 = 0.5\lambda, \ldots, R_{16} = 7.5\lambda$.

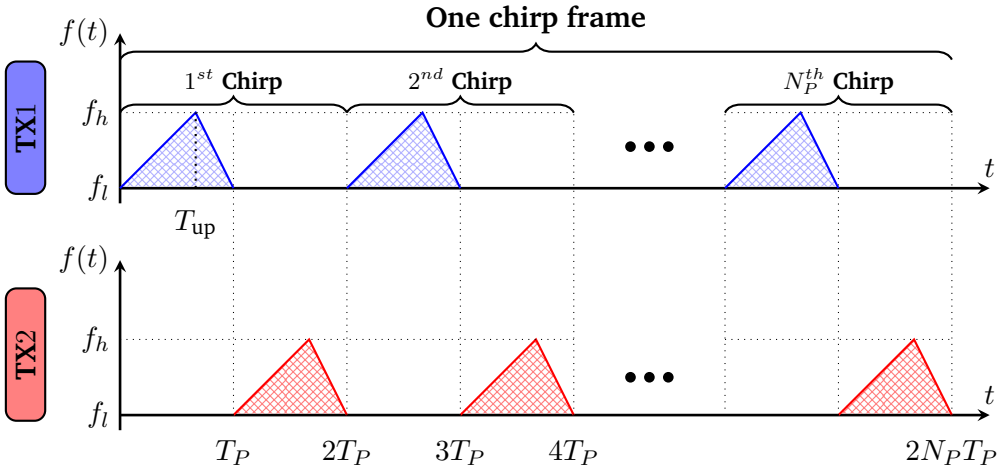


Fig. B.3: A time-division multiplexed waveform for the RBK2 MIMO configuration, illustrating alternating chirp sequences for TX1 (blue) and TX2 (red) over a time frame with N_P chirps, each separated by T_P, and a chirp duration of T_{up}.

The virtual array element positions $V_{m,n}$ are given by:

$$V_{m,n} = T_m + R_n = (m-1) \cdot 7.5\lambda + (n-1) \cdot 0.5\lambda, \tag{B.1}$$

- For $m = 1$ (Tx1):

$$V_{1,n} = 0 + (n-1) \cdot 0.5\lambda = (n-1) \cdot 0.5\lambda, \tag{B.2}$$

  yielding: $0, 0.5\lambda, 1\lambda, \ldots, 7.5\lambda$ (16 positions).

- For $m = 2$ (Tx2):

$$V_{2,n} = 7.5\lambda + (n-1) \cdot 0.5\lambda, \tag{B.3}$$

  yielding: $7.5\lambda, 8\lambda, 8.5\lambda, \ldots, 15\lambda$ (16 positions).

The total set of virtual positions is the union of $V_{1,n}$ and $V_{2,n}$, resulting in 32 elements. However, an overlap occurs at $V_{1,16} = 7.5\lambda$ (Tx1, Rx16) and $V_{2,1} = 7.5\lambda$ (Tx2, Rx1), reducing the number of unique positions. The distinct virtual array spans 0λ to 15λ with 0.5λ spacing:

- Number of unique positions = $\frac{15\lambda}{0.5\lambda} + 1 = 31$.

Thus, the maximum number of non-overlapped virtual elements is 31, with a total aperture of 15λ. In this thesis, this setup was utilized to perform high-resolution AoA estimation.

APPENDIX C

Deep Learning Foundations

C.1 Convolutional Neural Networks for Gesture Recognition

Convolutional Neural Network (CNN) excel at extracting spatial features from structured data, such as images, making them a powerful tool for tasks like gesture recognition. In this context, CNNs can process inputs like radar-based Range-Doppler Maps (RDMs) or spectrograms to identify patterns unique to hand movements, but their primary strength lies in handling image data [98]. Unlike traditional neural networks, which process each pixel independently, CNNs leverage local spatial relationships through convolutional operations, making them highly effective for detecting patterns like edges, textures, and shapes in images.

A CNN's architecture comprises several key components: **convolutional layers**, **activation functions**, **pooling layers**, and Fully Connected (FC) layers. Convolutional layers employ learnable filters to extract features such as edges, textures, and complex structures. These filters slide over the input, performing a 2D discrete convolution operation by computing dot products with local regions to produce feature maps that highlight key spatial characteristics. The 2D discrete convolution operation for a single output feature map can be mathematically expressed as [98]:

$$Y(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X(i+m, j+n) \cdot K(m,n) + b \tag{C.1}$$

where $Y(i,j)$ represents the value of the output feature map at position (i,j), X is the input (e.g., an image), K is the filter (or kernel) of size $M \times N$, and b is an optional bias term that shifts the output. The indices m and n iterate over the kernel dimensions, ensuring that the filter captures local spatial dependencies.

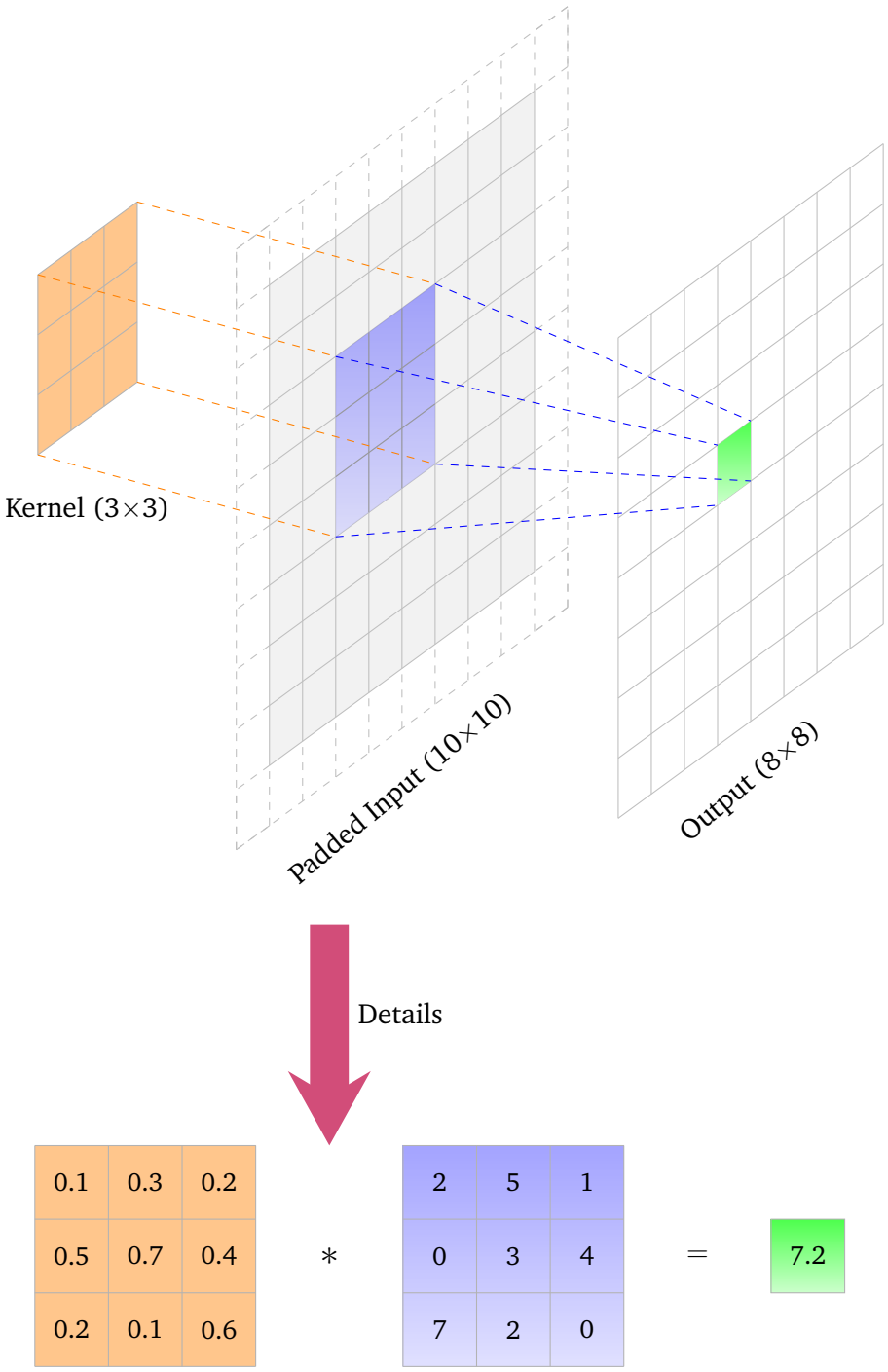


Fig. C.1: The output of a 2D convolution

To illustrate the convolution process, Fig. C.1 provides a detailed visualization. The top part of the figure depicts a 3D schematic of a 2D convolution operation: a 10×10 padded input (gray) is convolved with a 3×3 kernel (orange) to produce an 8×8 output feature map (green). The blue shaded region highlights the input window currently being processed, with dashed lines showing the mapping from the kernel and input to the corresponding output cell. The bottom part of the figure zooms into a specific convolution step, showing the element-wise multiplication of the 3×3 kernel and input window, resulting in a single output value (7.2).

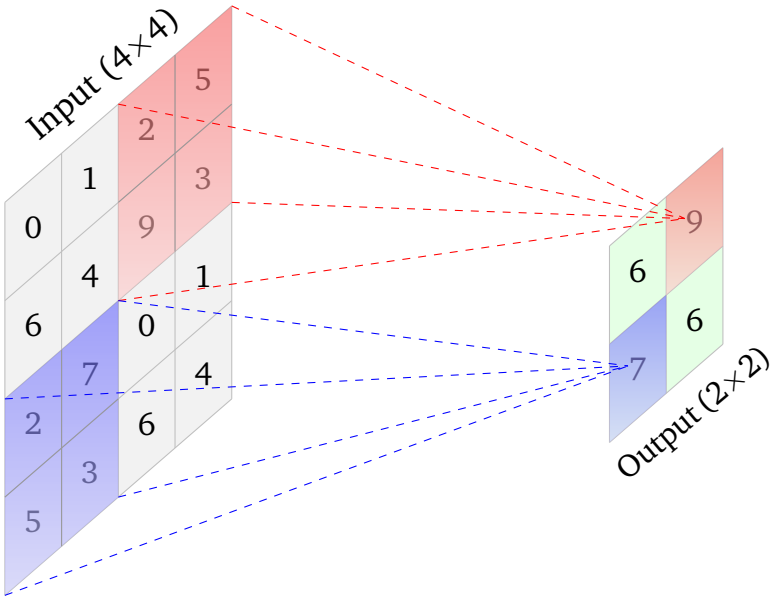


Fig. C.2: Max pooling operation with a 2×2 window and stride of 2, reducing a 4×4 feature map to a 2×2 output. Red dashed lines show the mapping for the top-right region, and blue dashed lines show the mapping for the bottom-left region.

Two critical parameters of the convolution operation are **stride** and **padding**, which significantly influence the output dimensions and the network's behavior. The stride defines the step size by which the filter moves across the input. A stride of 1 means the filter shifts by one unit at a time, producing a feature map that closely preserves spatial resolution, while a larger stride (e.g., 2) reduces the output size, effectively downsampling the feature map and increasing computational efficiency. Padding involves adding extra border pixels (typically zeros) around the input to control the output dimensions. Zero-padding, as depicted in Fig. C.1, ensures that the output size matches the input size when using a stride of 1, allowing the filter to process edge regions effectively. Without padding, the output size shrinks due to the filter's inability to extend beyond the input boundaries.

The Rectified Linear Unit (ReLU) activation function introduces non-linearity after convolution, enabling the network to model complex patterns:

$$f(x) = \max(0, x). \tag{C.2}$$

This widely adopted function enhances the network's ability to learn intricate features by allowing positive values to pass unchanged while setting negative values to zero, thus promoting sparsity and improving convergence [50].

Following convolution and activation, pooling layers reduce the dimensionality of feature maps while preserving essential information. Max pooling, a common technique, operates by sliding a window (e.g., 2×2) over the feature map and selecting the maximum value within each region. This process downsamples the data and mitigates overfitting by focusing on dominant features, a concept well-explained in [50]. For example, a 2×2 max pooling operation with a stride of 2 halves the spatial dimensions of the feature map, as illustrated in Fig. C.2. This figure shows a

4×4 input feature map being reduced to a 2×2 output, where each output value corresponds to the maximum within its respective 2×2 region.

At the network's end, fully connected layers transform the extracted features into classification probabilities, often using a softmax function to assign likelihoods to different gesture classes. In this thesis, CNN serve as feature extractors, processing inputs to distinguish patterns like hand movements before passing the results to recurrent models for temporal analysis.

C.2 Long Short-Term Memory Networks

Recurrent Neural Networks (RNNs) are essential for processing sequential data, such as radar frames in gesture recognition, where temporal relationships are crucial. However, traditional RNNs struggle with the vanishing gradient problem, which hinders their ability to learn long-term dependencies. Long Short-Term Memorys (LSTMs) address this by incorporating a memory cell and three gates–forget, input, and output—that regulate information flow, enabling effective modeling of extended temporal dependencies [50].

An LSTM updates its hidden state $\mathbf{h}_{(t)} \in \mathbb{R}^{d_h}$ and cell state $\mathbf{c}_{(t)} \in \mathbb{R}^{d_h}$ (long-term memory) at time t using the current input $\mathbf{x}_{(t)} \in \mathbb{R}^{d_x}$ and previous hidden state $\mathbf{h}_{(t-1)} \in \mathbb{R}^{d_h}$. This process involves four layers: forget gate, input gate, candidate cell state, and output gate. Fig. C.3 illustrates the structure of an LSTM cell, detailing the flow of information through its gates.

Each layer is governed by distinct weight matrices and bias vectors, enabling selective memory updates. The weight matrices $\mathbf{W}_{xf}, \mathbf{W}_{xi}, \mathbf{W}_{xg}, \mathbf{W}_{xo} \in \mathbb{R}^{d_x \times d_h}$ map the input vector $\mathbf{x}_t$ to the forget, input, candidate, and output gates, respectively, while $\mathbf{W}_{hf}, \mathbf{W}_{hi}, \mathbf{W}_{hg}, \mathbf{W}_{ho} \in \mathbb{R}^{d_h \times d_h}$ connect the previous hidden state $\mathbf{h}_{(t-1)}$. The bias vectors $\mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_g, \mathbf{b}_o \in \mathbb{R}^{d_h}$ are often initialized to ones in frameworks like TensorFlow to prevent premature forgetting during training. The operations of these layers, as detailed by Géron [50], are defined as follows:

The forget gate $\mathbf{f}_{(t)} \in \mathbb{R}^{d_h}$ controls the retention of the previous cell state $\mathbf{c}_{(t-1)}$:

$$\mathbf{f}_{(t)} = \sigma(\mathbf{W}_{xf}^{\mathrm{T}}\mathbf{x}_{(t)} + \mathbf{W}_{hf}^{\mathrm{T}}\mathbf{h}_{(t-1)} + \mathbf{b}_f) \tag{C.3}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function, producing values in $[0, 1]$. The input gate $\mathbf{i}_{(t)} \in \mathbb{R}^{d_h}$ determines which new information to store in the cell state:

$$\mathbf{i}_{(t)} = \sigma(\mathbf{W}_{xi}^{\mathrm{T}}\mathbf{x}_{(t)} + \mathbf{W}_{hi}^{\mathrm{T}}\mathbf{h}_{(t-1)} + \mathbf{b}_i). \tag{C.4}$$

The candidate gate $\mathbf{g}_{(t)} \in \mathbb{R}^{d_h}$ computes potential updates to the cell state:

$$\mathbf{g}_{(t)} = \tanh(\mathbf{W}_{xg}^{\mathrm{T}}\mathbf{x}_{(t)} + \mathbf{W}_{hg}^{\mathrm{T}}\mathbf{h}_{(t-1)} + \mathbf{b}_g) \tag{C.5}$$

where $\tanh$ is the hyperbolic tangent function, producing values in $[-1, 1]$. The cell state $\mathbf{c}_{(t)}$ combines retained and new information:

$$\mathbf{c}_{(t)} = f_{(t)} \otimes \mathbf{c}_{(t-1)} + \mathbf{i}_{(t)} \otimes \mathbf{g}_{(t)} \tag{C.6}$$

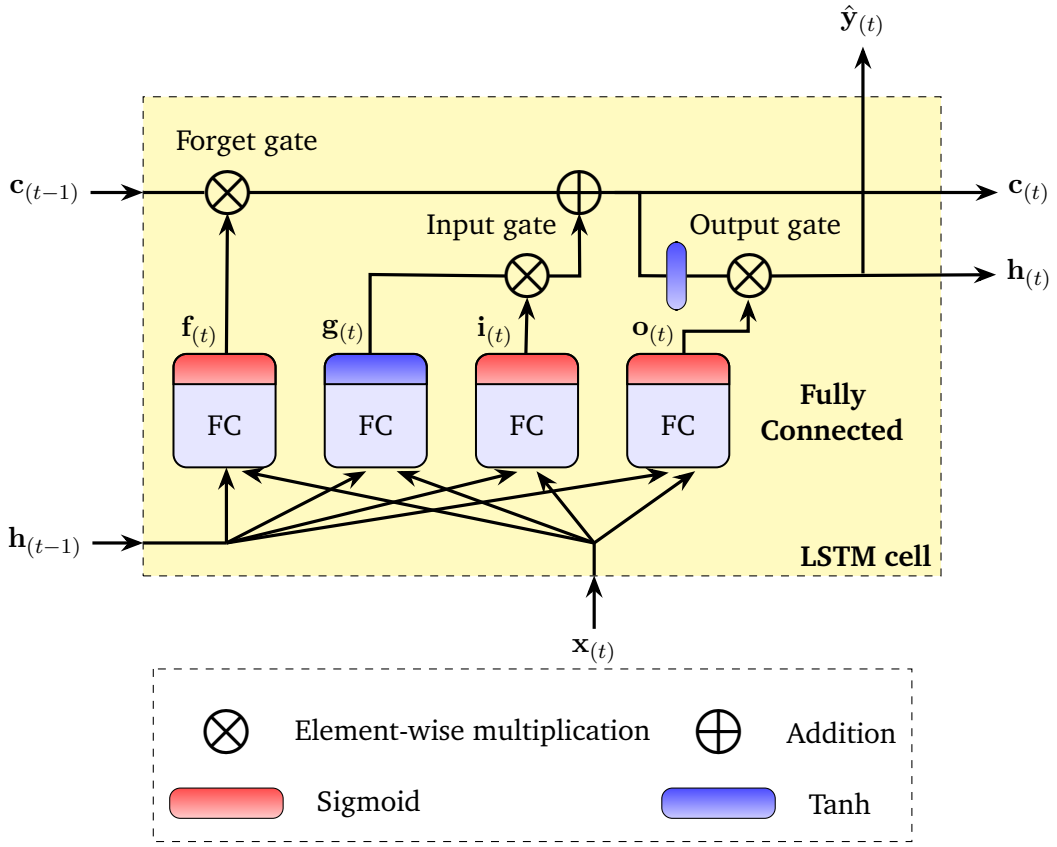


Fig. C.3: An LSTM cell

with $\otimes$ indicating element-wise multiplication. The output gate $\mathbf{o}_{(t)} \in \mathbb{R}^{d_h}$ controls the hidden state output:

$$\mathbf{o}_{(t)} = \sigma(\mathbf{W}_{xo}^{\mathrm{T}}\mathbf{x}_{(t)} + \mathbf{W}_{ho}^{\mathrm{T}}\mathbf{h}_{(t-1)} + \mathbf{b}_o). \tag{C.7}$$

The hidden state is then:

$$\mathbf{h}_{(t)} = \mathbf{o}_{(t)} \otimes \tanh(\mathbf{c}_{(t)}). \tag{C.8}$$

An output $\hat{\mathbf{y}}_{(t)}$ may be derived from $\mathbf{h}_{(t)}$ via additional layers (e.g., FCs), depending on the model. LSTM's ability to retain long-term dependencies makes them powerful, though their complexity can challenge real-time applications like gesture recognition.

C.3 Gated Recurrent Units

Gated Recurrent Units (GRUs) offer a optimized alternative to LSTMs, tackling the vanishing gradient problem with reduced computational overhead. Unlike LSTMs' three gates and separate cell state, GRUs use two gates—update and reset—and combine memory and hidden state into $\mathbf{h}_{(t)}$. This efficiency suits them for medium-length sequences, such as radar frame sequences in gesture recognition [50].

A GRU updates $\mathbf{h}_{(t)}$ using $\mathbf{x}_{(t)}$ and $\mathbf{h}_{(t-1)}$ across three layers: update gate, reset gate, and candidate gate. Fig. C.4 depicts the GRU cell architecture, highlighting its

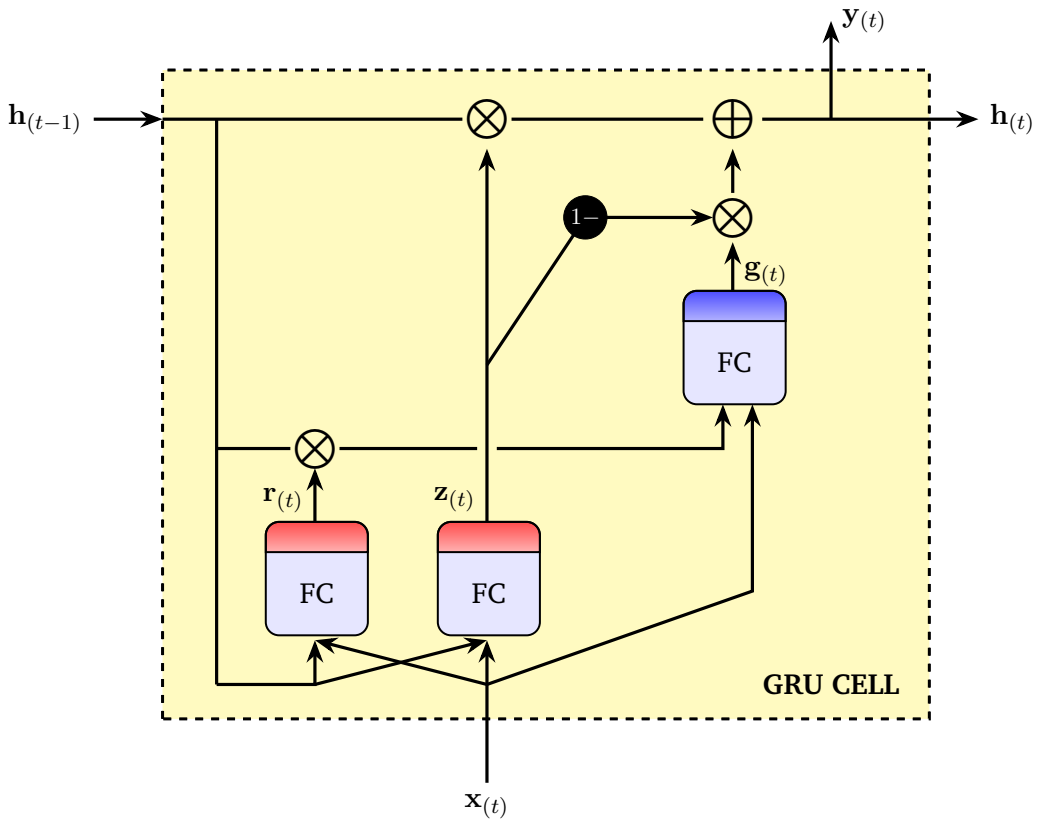


Fig. C.4: GRU cell

simplified gate structure.

These layers are controlled by weight matrices and bias vectors. The matrices $\mathbf{W}_{xz}, \mathbf{W}_{xr}, \mathbf{W}_{xg} \in \mathbb{R}^{d_x \times d_h}$ map $\mathbf{x}_t$ to the update, reset, and candidate gates, respectively, while $\mathbf{W}_{hz}, \mathbf{W}_{hr}, \mathbf{W}_{hg} \in \mathbb{R}^{d_h \times d_h}$ connect $\mathbf{h}_{(t-1)}$. Biases $\mathbf{b}_z, \mathbf{b}_r, \mathbf{b}_g$ are included, often initialized to stabilize training. These computations, following Géron's formulation [50], are expressed as follows.

The update gate $\mathbf{z}_{(t)} \in \mathbb{R}^{d_h}$ decides how much of $\mathbf{h}_{(t-1)}$ to retain:

$$\mathbf{z}_{(t)} = \sigma(\mathbf{W}_{xz}^{\mathrm{T}}\mathbf{x}_{(t)} + \mathbf{W}_{hz}^{\mathrm{T}}\mathbf{h}_{(t-1)} + \mathbf{b}_z) \tag{C.9}$$

where $\mathbf{z}_{(t)}$ is the update gate's activation. The reset gate $\mathbf{r}_{(t)} \in \mathbb{R}^{d_h}$ controls how much of $\mathbf{h}_{(t-1)}$ to forget:

$$\mathbf{r}_{(t)} = \sigma(\mathbf{W}_{xr}^{\mathrm{T}}\mathbf{x}_{(t)} + \mathbf{W}_{hr}^{\mathrm{T}}\mathbf{h}_{(t-1)} + \mathbf{b}_r) \tag{C.10}$$

where $\mathbf{r}_{(t)}$ is the reset gate's activation. A candidate hidden state $\mathbf{g}_{(t)} \in \mathbb{R}^{d_h}$ is computed, modulated by the reset gate:

$$\mathbf{g}_{(t)} = \tanh(\mathbf{W}_{xg}^{\mathrm{T}}\mathbf{x}_{(t)} + \mathbf{W}_{hg}^{\mathrm{T}}(\mathbf{r}_{(t)} \otimes \mathbf{h}_{(t-1)}) + \mathbf{b}_g) \tag{C.11}$$

The final hidden state $\mathbf{h}_{(t)}$ blends $\mathbf{h}_{(t-1)}$ and $\mathbf{g}_{(t)}$:

$$\mathbf{h}_{(t)} = \mathbf{z}_{(t)} \otimes \mathbf{h}_{(t-1)} + (1 - \mathbf{z}_{(t)}) \otimes \mathbf{g}_{(t)} \tag{C.12}$$

The simplicity of GRUs, which use two gates and one state, maintains temporal modeling capability while requiring less computational power than LSTMs. This balance enhances real-time applicability for gesture recognition with radar frame sequences.

APPENDIX D

List of publications

1. K. Alirezazad, J. Schmitt and L. Maurer, "Vital Sign Monitoring in Cluttered Environments: Leveraging FMCW Radar and PCA Analysis," 2024 25th International Microwave and Radar Conference (MIKON), Wroclaw, Poland, 2024, pp. 103108, doi: 10.23919/MIKON60251.2024.10633997.

2. K. Alirezazad, J. Schmitt and L. Maurer, "Enhanced Vital Sign Monitoring Using FMCW Radar and PCA Analysis," 2024 IEEE MTT-S International Microwave Biomedical Conference (IMBioC), Montreal, QC, Canada, 2024, pp. 2426, doi: 10.1109/IMBioC60287.2024.10590087.

3. K. Alirezazad and L. Maurer, "Human Target Identification via De-Cluttered RAI and MIMO FMCW Radar," 2023 IEEE Biomedical Circuits and Systems Conference (BioCAS), Toronto, ON, Canada, 2023, pp. 15, doi: 10.1109/BioCAS58349.2023.10388666.

4. K. Alirezazad and L. Maurer, "Human Target Recognition Using MIMO FMCW Radar and Slow-time DC-Value Suppression," 2023 IEEE International Radar Conference (RADAR), Sydney, Australia, 2023, pp. 16, doi: 10.1109/RADAR54928.2023.10371067.

5. K. Alirezazad and L. Maurer, "Detection of human targets using a MIMO FMCW radar with slow-time DC-Value suppression," in Technologies Enabling Future Mobile Connectivity and Sensing, River Publishers, 2023, pp.143-150.

6. K. Alirezazad and L. Maurer, "FMCW Radar-Based Hand Gesture Recognition Using Dual-Stream CNN-GRU Model," 2022 24th International Microwave and

Radar Conference (MIKON), Gdansk, Poland, 2022, pp. 15, doi: 10.23919/MIKON54314.2022.9924984.

7. K. Alirezazad, G. Rhiel and L. Maurer, "2D CNN-GRU Model for Multi-Hand Gesture Recognition System Using FMCW Radar," 2022 20th IEEE Interregional NEWCAS Conference (NEWCAS), Quebec City, QC, Canada, 2022, pp. 158162, doi: 10.1109/NEWCAS52662.2022.9841948.

Bibliography

[1] M. G. Amin. *Radar for Indoor Monitoring: Detection, Classification, and Assessment*. CRC Press, 2017 (cit. on p. 1).

[2] Y. Wang et al. "Remote Monitoring of Human Vital Signs Based on 77-GHz mm-Wave FMCW Radar". In: *Sensors* 20.10 (2020). DOI: 10.3390/s20102999 (cit. on pp. 1, 6).

[3] M. Alizadeh et al. "Remote monitoring of human vital signs using mm-wave FMCW radar". In: *IEEE Access* 7 (2019), pp. 54958–54968 (cit. on pp. 1, 6, 12, 14, 24, 35, 50, 51, 53).

[4] X. Qiu et al. "A Survey of Gesture Recognition Using Frequency Modulated Continuous Wave Radar". In: *Journal of Computer and Communications* 12.6 (2024), pp. 115–134 (cit. on p. 1).

[5] S. Yin et al. "Wearable Physiological Multi-Vital Sign Monitoring System With Medical Standard". In: *IEEE Sensors Journal* 21.23 (2021), pp. 27157–27167. DOI: 10.1109/JSEN.2021.3123084 (cit. on pp. 1, 2, 18).

[6] C. Botrugno et al. "Smart combination of ECG and PPG signals: an innovative approach towards an electronic device for vital signs monitoring". In: *2023 9th International Workshop on Advances in Sensors and Interfaces (IWASI)*. 2023, pp. 256–260. DOI: 10.1109/IWASI58316.2023.10164373 (cit. on pp. 1, 2, 18).

[7] W. Zhang, J. Wang, and F. Lan. "Dynamic hand gesture recognition based on short-term sampling neural networks". In: *IEEE/CAA Journal of Automatica Sinica* 8.1 (2021), pp. 110–120. DOI: 10.1109/JAS.2020.1003465 (cit. on p. 1).

[8] D. Fan et al. "Multi-Task and Multi-Modal Learning for RGB Dynamic Gesture Recognition". In: *IEEE Sensors Journal* 21.23 (2021), pp. 27026–27036. DOI: 10.1109/JSEN.2021.3123443 (cit. on p. 1).

[9] S. Zhao et al. “A novel interactive method of virtual reality system based on hand gesture recognition”. In: *2009 Chinese Control and Decision Conference*. 2009, pp. 5879–5882. DOI: 10.1109/CCDC.2009.5195252 (cit. on pp. 2, 69).

[10] Z. Peng et al. “An FMCW radar sensor for human gesture recognition in the presence of multiple targets”. In: *2017 First IEEE MTT-S International Microwave Bio Conference (IMBIOC)*. 2017, pp. 1–3. DOI: 10.1109/IMBIOC.2017.7965798 (cit. on pp. 2, 58).

[11] X. Dong et al. “FMCW Radar-Based Hand Gesture Recognition Using Spatiotemporal Deformable and Context-Aware Convolutional 5-D Feature Representation”. In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–11. DOI: 10.1109/TGRS.2021.3122332 (cit. on pp. 2, 58).

[12] K. Alirezazad and L. Maurer. “Human Target Recognition Using MIMO FMCW Radar and Slow-time DC-Value Suppression”. In: *2023 IEEE International Radar Conference (RADAR)*. 2023, pp. 1–6. DOI: 10.1109/RADAR54928.2023.10371067 (cit. on p. 6).

[13] E. Cardillo and A. Caddemi. “Radar Range-Breathing Separation for the Automatic Detection of Humans in Cluttered Environments”. In: *IEEE Sensors Journal* 21.13 (2021), pp. 14043–14050. DOI: 10.1109/JSEN.2020.3024961 (cit. on pp. 6, 35).

[14] W. Kang, F. Ke, and Y. C. Chang. “Non-Contact Realtime Vital Signs Monitoring System Based on Millimeter Wave FMCW Radar”. In: *2023 9th International Conference on Computer and Communications (ICCC)*. 2023, pp. 1594–1599. DOI: 10.1109/ICCC59590.2023.10507262 (cit. on pp. 6, 50, 51, 53).

[15] L. Liu et al. “mmRH: Noncontact Vital Sign Detection With an FMCW mmWave Radar”. In: *IEEE Sensors Journal* 23.8 (2023), pp. 8856–8866. DOI: 10.1109/JSEN.2023.3250500 (cit. on pp. 6, 9, 14, 15, 18).

[16] J. Gamba. *Radar Signal Processing for Autonomous Driving*. 1st. Singapore: Springer Nature Singapore Pte Ltd., 2020 (cit. on p. 6).

[17] M. Holtzman, R. Goubran, and F. Knoefel. “Maximal ratio combining for respiratory effort extraction from pressure sensor arrays”. In: *2011 IEEE International Symposium on Medical Measurements and Applications*. 2011, pp. 88–92. DOI: 10.1109/MeMeA.2011.5966674 (cit. on pp. 6, 14).

[18] T. Sakamoto. “Noncontact Measurement of Human Vital Signs during Sleep Using Low-power Millimeter-wave Ultrawideband MIMO Array Radar”. In: *2019 IEEE MTT-S International Microwave Biomedical Conference (IMBioC)*. Vol. 1. 2019, pp. 1–4. DOI: 10.1109/IMBIOC.2019.8777864 (cit. on pp. 6, 14).

[19] T. Sakamoto and K. Yamashita. “Noncontact Measurement of Autonomic Nervous System Activities Based on Heart Rate Variability Using Ultra-Wideband Array Radar”. In: *IEEE Journal of Electromagnetics, RF and Microwaves in Medicine and Biology* 4.3 (2020), pp. 208–215. DOI: 10.1109/JERM.2019.2948827 (cit. on pp. 6, 15).

[20] Y. Wang et al. “A Novel Non-Contact Respiration and Heartbeat Detection Method Using Frequency-Modulated Continuous Wave Radar”. In: *IEEE Sensors Journal* 24.7 (2024), pp. 10434–10446. DOI: 10.1109/JSEN.2024.3351274 (cit. on pp. 6, 9, 11).

[21] K. Han and S. Hong. “Detection and Localization of Multiple Humans Based on Curve Length of I/Q Signal Trajectory Using MIMO FMCW Radar”. In: *IEEE Microwave and Wireless Components Letters* 31.4 (2021), pp. 413–416. DOI: 10.1109/LMWC.2021.3057867 (cit. on pp. 6, 21, 27, 30).

[22] L. Qu et al. “Vital Sign Detection of FMCW Radar Based on Improved Adaptive Parameter Variational Mode Decomposition”. In: *IEEE Sensors Journal* 23.20 (2023), pp. 25048–25060. DOI: 10.1109/JSEN.2023.3312513 (cit. on pp. 6, 7, 9, 10, 15).

[23] M. Zhao, Z. Zhang, and W. Bao. “A High-precision Multi-target Vital Signs Detection Method Based on FMCW Radar”. In: *2024 Photonics & Electromagnetics Research Symposium (PIERS)*. 2024, pp. 1–8. DOI: 10.1109/PIERS62282.2024.10618777 (cit. on pp. 6–9).

[24] S. Ma et al. “Radar Vital Signs Detection Method Based on Variational Mode Decomposition and Wavelet Transform”. In: *2021 China Automation Congress (CAC)*. 2021, pp. 7469–7474. DOI: 10.1109/CAC53003.2021.9728129 (cit. on pp. 6, 8).

[25] Y. Eder and Y. C. Eldar. “Sparsity-Based Multi-Person Non-Contact Vital Signs Monitoring via FMCW Radar”. In: *IEEE Journal of Biomedical and Health Informatics* 27.6 (2023), pp. 2806–2817. DOI: 10.1109/JBHI.2023.3255740 (cit. on pp. 6, 9, 12–15, 19).

[26] A. Ahmad et al. “Vital signs monitoring of multiple people using a FMCW millimeter-wave sensor”. In: *2018 IEEE Radar Conference (RadarConf18)*. 2018, pp. 1450–1455. DOI: 10.1109/RADAR.2018.8378778 (cit. on p. 6).

[27] J.-M. Muñoz-Ferreras et al. “FMCW-Radar-Based Vital-Sign Monitoring of Multiple Patients”. In: *2019 IEEE MTT-S International Microwave Biomedical Conference (IMBioC)*. Vol. 1. 2019, pp. 1–3. DOI: 10.1109/IMBIOC.2019.8777845 (cit. on p. 7).

[28] T. K. V. Dai et al. “Enhancement of Remote Vital Sign Monitoring Detection Accuracy Using Multiple-Input Multiple-Output 77 GHz FMCW Radar”. In: *IEEE Journal of Electromagnetics, RF and Microwaves in Medicine and Biology* 6.1 (2022), pp. 111–122. DOI: 10.1109/JERM.2021.3082807 (cit. on pp. 7, 11, 14).

[29] K. Dragomiretskiy and D. Zosso. “Variational Mode Decomposition”. In: *IEEE Transactions on Signal Processing* 62.3 (2014), pp. 531–544. DOI: 10.1109/TSP.2013.2288675 (cit. on p. 7).

[30] X. Zhang et al. “Mutual Interference Suppression Using Signal Separation and Adaptive Mode Decomposition in Noncontact Vital Sign Measurements”. In: *IEEE Transactions on Instrumentation and Measurement* 71 (2022), pp. 1–15. DOI: 10.1109/TIM.2021.3132924 (cit. on pp. 8–10).

[31] M. Omachi and S. Omachi. “Fast calculation of continuous wavelet transform using polynomial”. In: *2007 International Conference on Wavelet Analysis and Pattern Recognition*. Vol. 4. 2007, pp. 1688–1691. DOI: 10.1109/ICWAPR.2007.4421725 (cit. on p. 11).

[32] Z. Liu et al. “Vital Sign Extraction in the Presence of Radar Mutual Interference”. In: *IEEE Signal Processing Letters* 27 (2020), pp. 1745–1749. DOI: 10.1109/LSP.2020.3026942 (cit. on p. 11).

[33] J. Wang et al. “Noncontact Distance and Amplitude-Independent Vibration Measurement Based on an Extended DACM Algorithm”. In: *IEEE Transactions on Instrumentation and Measurement* 63.1 (2014), pp. 145–153. DOI: 10.1109/TIM.2013.2277530 (cit. on p. 11).

[34] N. Golyandina and A. Zhigljavsky. *Singular Spectrum Analysis for Time Series*. 2nd ed. SpringerBriefs in Statistics. Berlin, Heidelberg: Springer, 2020. DOI: 10.1007/978-3-662-62436-4 (cit. on p. 11).

[35] J. B. Elsner and A. A. Tsonis. *Singular Spectrum Analysis: A New Tool in Time Series Analysis*. Berlin, Germany: Springer, 2013 (cit. on p. 11).

[36] R. Vautard, P. Yiou, and M. Ghil. “Singular-spectrum analysis: A toolkit for short, noisy chaotic signals”. In: *Physica D: Nonlinear Phenomena* 58.1 (1992), pp. 95–126. DOI: https://doi.org/10.1016/0167-2789(92)90103-T (cit. on p. 11).

[37] E. Turppa et al. “Vital Sign Monitoring Using FMCW Radar in Various Sleeping Scenarios”. In: *Sensors* 20.22 (2020). DOI: 10.3390/s20226505 (cit. on pp. 12, 19).

[38] M. Mercuri et al. “Vital-sign monitoring and spatial tracking of multiple people using a contactless radar-based sensor”. In: *Nature* (June 2019). DOI: 10.1038/s41928-019-0258-6 (cit. on pp. 12, 14).

[39] W.-H. Fang and L.-D. Fang. “Joint Angle and Range Estimation With Signal Clustering in FMCW Radar”. In: *IEEE Sensors Journal* 20.4 (2020), pp. 1882–1892. DOI: 10.1109/JSEN.2019.2949367 (cit. on p. 12).

[40] H. Zhao et al. “Noncontact Physiological Dynamics Detection Using Low-power Digital-IF Doppler Radar”. In: *IEEE Transactions on Instrumentation and Measurement* 66.7 (2017), pp. 1780–1788. DOI: 10.1109/TIM.2017.2669699 (cit. on p. 18).

[41] K. Han and S. Hong. “Vocal Signal Detection and Speaking-Human Localization With MIMO FMCW Radar”. In: *IEEE Transactions on Microwave Theory and Techniques* 69.11 (2021), pp. 4791–4802. DOI: 10.1109/TMTT.2021.3102233 (cit. on p. 19).

[42] Y.-H. Shen, Y.-R. Chien, and S.-H. Fang. “Poster Abstract: Human Detection with Weak Ranging Signal for FMCW Radar Systems”. In: *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 2020, pp. 343–344. DOI: 10.1109/IPSN48710.2020.00-15 (cit. on p. 21).

[43] Z. Peng et al. “A Portable FMCW Interferometry Radar With Programmable Low-IF Architecture for Localization, ISAR Imaging, and Vital Sign Tracking”. In: *IEEE Transactions on Microwave Theory and Techniques* 65.4 (2017), pp. 1334–1344. DOI: 10.1109/TMTT.2016.2633352 (cit. on p. 21).

[44] C. Will et al. “Human Target Detection, Tracking, and Classification Using 24-GHz FMCW Radar”. In: *IEEE Sensors Journal* 19.17 (2019), pp. 7283–7299. DOI: 10.1109/JSEN.2019.2914365 (cit. on p. 21).

[45] P. Nallabolu et al. “Human Presence Sensing and Gesture Recognition for Smart Home Applications With Moving and Stationary Clutter Suppression Using a 60-GHz Digital Beamforming FMCW Radar”. In: *IEEE Access* 9 (2021), pp. 72857–72866. DOI: 10.1109/ACCESS.2021.3080655 (cit. on p. 21).

[46] I. GmbH. *Radarbook2 Applications*. Datasheet. Retrieved from https://web.archive.org/web/20220516061714/http://inras.at/wp-content/uploads/2021/11/RBK2_Applications_02.pdf. 2020 (cit. on pp. 22, 115).

[47] G. Paterniani et al. “Radar-Based Monitoring of Vital Signs: A Tutorial Overview”. In: *Proceedings of the IEEE* 111.3 (2023), pp. 277–317. DOI: 10.1109/JPROC.2023.3244362 (cit. on pp. 24, 35).

[48] P. Stockel et al. “Interference Free Vital Sign Extraction with Radar Using a Signal Fusion Approach”. In: *2023 20th European Radar Conference (EuRAD)*. 2023, pp. 310–313. DOI: 10.23919/EuRAD58043.2023.10289569 (cit. on p. 35).

[49] C.-Y. Huang et al. “Clutter-Resistant Vital Sign Detection Using Amplitude-Based Demodulation by EEMD-PCA-Correlation Algorithm for FMCW Radar Systems”. In: *2019 49th European Microwave Conference (EuMC)*. 2019, pp. 928–931. DOI: 10.23919/EuMC.2019.8910730 (cit. on pp. 36, 39).

[50] A. Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media, 2022 (cit. on pp. 37, 39, 81, 85, 121–124).

[51] J. Shlens. “A Tutorial on Principal Component Analysis”. In: *arXiv preprint arXiv:1404.1100* (2014) (cit. on p. 37).

[52] A. Winursito et al. “Feature Data Reduction of MFCC Using PCA and SVD in Speech Recognition System”. In: *2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*. 2018, pp. 1–6. DOI: 10.1109/ICSCEE.2018.8538414 (cit. on p. 37).

[53] N. Singh and A. K. Bhandari. “Principal Component Analysis-Based Low-Light Image Enhancement Using Reflection Model”. In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–10. DOI: 10.1109/TIM.2021.3096266 (cit. on p. 39).

[54] E. Ltd. *EQ-000708-BR-3p0-eq02-LifeMonitor*. Datasheet. Retrieved from https://equivital.com/wp-content/uploads/2021/07/EQ-000708-BR-3-eq02-LifeMonitor-Datasheet.pdf. 2013 (cit. on pp. 47, 48, 50, 51, 105).

[55] M. R. Kazemi and A. A. Jafari. *Modified Signed Log-Likelihood Ratio Test for Comparing the Correlation Coefficients of Two Independent Bivariate Normal Distributions*. 2016. arXiv: 1605.09700 [stat.ME] (cit. on p. 53).

[56] Y. Sun et al. "Real-Time Radar-Based Gesture Detection and Recognition Built in an Edge-Computing Platform". In: *IEEE Sensors Journal* 20.18 (2020), pp. 10706–10716. DOI: 10.1109/JSEN.2020.2994292 (cit. on pp. 58, 69, 71).

[57] J.-W. Choi, S.-J. Ryu, and J.-H. Kim. "Short-Range Radar Based Real-Time Hand Gesture Recognition Using LSTM Encoder". In: *IEEE Access* 7 (2019), pp. 33610–33618. DOI: 10.1109/ACCESS.2019.2903586 (cit. on pp. 58, 61, 65, 71).

[58] J. S. Suh et al. "24 GHz FMCW Radar System for Real-Time Hand Gesture Recognition Using LSTM". In: *2018 Asia-Pacific Microwave Conference (APMC)*. 2018, pp. 860–862. DOI: 10.23919/APMC.2018.8617375 (cit. on pp. 58, 62, 65).

[59] S.-J. Ryu et al. "Feature-Based Hand Gesture Recognition Using an FMCW Radar and its Temporal Feature Analysis". In: *IEEE Sensors Journal* 18.18 (2018), pp. 7593–7602. DOI: 10.1109/JSEN.2018.2859815 (cit. on pp. 58, 63).

[60] J.-T. Yu, L. Yen, and P.-H. Tseng. "mmWave Radar-based Hand Gesture Recognition using Range-Angle Image". In: *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. 2020, pp. 1–5. DOI: 10.1109/VTC2020-Spring48590.2020.9128573 (cit. on pp. 58, 71).

[61] T. Instruments. *IWR1443BOOST Evaluation Module mmWave Sensing Solution*. https://www.ti.com/tool/IWR1443BOOST. Accessed: 2025-05-24. 2020 (cit. on p. 58).

[62] Y. Kim and B. Toomajian. "Hand Gesture Recognition Using Micro-Doppler Signatures With Convolutional Neural Network". In: *IEEE Access* 4 (2016), pp. 7125–7130 (cit. on p. 58).

[63] S. Ahmed et al. "Radar-Based Air-Writing Gesture Recognition Using a Novel Multistream CNN Approach". In: *IEEE Internet of Things Journal* 9.23 (2022), pp. 23869–23880. DOI: 10.1109/JIOT.2022.3189395 (cit. on p. 59).

[64] G. Malysa et al. "Hidden Markov model-based gesture recognition with FMCW radar". In: *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. 2016, pp. 1017–1021. DOI: 10.1109/GlobalSIP.2016.7905995 (cit. on pp. 60, 61, 65).

[65] S. Skaria et al. "Hand-Gesture Recognition Using Two-Antenna Doppler Radar With Deep Convolutional Neural Networks". In: *IEEE Sensors Journal* 19.8 (2019), pp. 3041–3048. DOI: 10.1109/JSEN.2019.2892073 (cit. on pp. 60, 61, 65).

[66] Z. Yang and X. Zheng. "Hand Gesture Recognition Based on Trajectories Features and Computation-Efficient Reused LSTM Network". In: *IEEE Sensors Journal* 21.15 (2021), pp. 16945–16960. DOI: 10.1109/JSEN.2021.3079564 (cit. on p. 61).

[67] S. Zhang et al. “Latern: Dynamic Continuous Hand Gesture Recognition Using FMCW Radar Sensor”. In: *IEEE Sensors Journal* 18.23 (2018), pp. 9708–9716 (cit. on pp. 61, 62, 65).

[68] I. Alnujaim et al. “Hand Gesture Recognition Using Input Impedance Variation of Two Antennas with Transfer Learning”. In: *IEEE Sensors Journal* 18.10 (2018), pp. 4129–4135. DOI: 10.1109/JSEN.2018.2820000 (cit. on pp. 61, 62, 65).

[69] K. Simonyan and A. Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV] (cit. on p. 61).

[70] Y. Wang et al. “Multi-Hand Gesture Recognition Using Automotive FMCW Radar Sensor”. In: *Remote Sensing* 14.10 (2022), p. 2374. DOI: 10.3390/rs14102374 (cit. on pp. 62, 63).

[71] Texas Instruments. *AWR1642: Single-Chip 76-GHz to 81-GHz Automotive Radar Sensor Integrating DSP and MCU*. https://www.ti.com/product/AWR1642. Accessed: 2022-04-19. 2022 (cit. on p. 62).

[72] H. Zhao et al. “DGSCR: Double-Target Gesture Separation and Classification Recognition Based on Deep Learning and Millimeter-Wave Radar”. In: *IEEE Sensors Journal* 23.21 (2023), pp. 26701–26711. DOI: 10.1109/JSEN.2023.3319339 (cit. on p. 62).

[73] J. B. Park et al. “Air Writing Gesture Recognition Using FMCW Radar and Deep Learning”. In: *2023 8th IEEE International Conference on Network Intelligence and Digital Content (IC-NIDC)*. 2023, pp. 369–373. DOI: 10.1109/IC-NIDC59918.2023.10390816 (cit. on p. 63).

[74] X. Shen et al. “ML-HGR-Net: A Meta-Learning Network for FMCW Radar Based Hand Gesture Recognition”. In: *IEEE Sensors Journal* 22.11 (2022), pp. 10808–10817. DOI: 10.1109/JSEN.2022.3169231 (cit. on pp. 63, 64).

[75] Z. Zhao et al. “Interference Suppression Based Gesture Recognition Method with FMCW Radar”. In: *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*. 2019, pp. 1–6. DOI: 10.1109/WCSP.2019.8928108 (cit. on pp. 63, 71).

[76] Y. Iida et al. “Range-Aware Hand Gesture Recognition Using FMCW Radar and Deep Learning”. In: *2023 IEEE 29th International Conference on Parallel and Distributed Systems (ICPADS)*. 2023, pp. 602–609. DOI: 10.1109/ICPADS60453.2023.00094 (cit. on p. 63).

[77] Y. Zhao et al. “Radar Dataset Synthesis Approach for Gesture Recognition”. In: *2023 20th European Radar Conference (EuRAD)*. 2023, pp. 193–196. DOI: 10.23919/EuRAD58043.2023.10289560 (cit. on pp. 64, 103).

[78] T. M P et al. “Hand Gesture Detection using Transfer Learning with Deep Neural Networks”. In: *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*. 2023, pp. 1–5. DOI: 10.1109/I2CT57861.2023.10126207 (cit. on p. 64).

[79] K. Liu et al. “Fusion of Inertial and Depth Sensor Data for Robust Hand Gesture Recognition”. In: *IEEE Sensors Journal* 14.6 (2014), pp. 1898–1903. DOI: 10.1109/JSEN.2014.2306094 (cit. on p. 64).

[80] W. Seok, Y. Kim, and C. Park. “Pattern recognition of human arm movement using deep reinforcement learning”. In: *2018 International Conference on Information Networking (ICOIN)*. 2018, pp. 917–919. DOI: 10.1109/ICOIN.2018.8343257 (cit. on p. 64).

[81] N. Zengeler, T. Kopinski, and U. Handmann. “Hand Gesture Recognition in Automotive Human–Machine Interaction Using Depth Cameras”. In: *Sensors* 19.1 (2019), p. 59. DOI: 10.3390/s19010059 (cit. on p. 69).

[82] Z. U. A. Akhtar and H. Wang. “WiFi-Based Gesture Recognition for Vehicular Infotainment System-An Integrated Approach”. In: *Applied Sciences* 9.24 (2019), p. 5268. DOI: 10.3390/app9245268 (cit. on p. 69).

[83] S. S. Rani, K. J. Dhrisya, and M. Ahalyadas. “Hand gesture control of virtual object in augmented reality”. In: *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 2017, pp. 1500–1505. DOI: 10.1109/ICACCI.2017.8126053 (cit. on p. 69).

[84] Y. Sun et al. “Gesture Classification with Handcrafted Micro-Doppler Features using a FMCW Radar”. In: *2018 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*. 2018, pp. 1–4. DOI: 10.1109/ICMIM.2018.8443507 (cit. on pp. 69, 71).

[85] M. Ritchie and A. M. Jones. “Micro-Doppler Gesture Recognition using Doppler, Time and Range Based Features”. In: *2019 IEEE Radar Conference (RadarConf)*. 2019, pp. 1–6. DOI: 10.1109/RADAR.2019.8835782 (cit. on p. 71).

[86] B. Dekker et al. “Gesture recognition with a low power FMCW radar and a deep convolutional neural network”. In: *2017 European Radar Conference (EURAD)*. 2017, pp. 163–166. DOI: 10.23919/EURAD.2017.8249172 (cit. on p. 71).

[87] P. Goswami et al. “Real-Time Multi-Gesture Recognition using 77 GHz FMCW MIMO Single Chip Radar”. In: *2019 IEEE International Conference on Consumer Electronics (ICCE)*. 2019, pp. 1–4. DOI: 10.1109/ICCE.2019.8662006 (cit. on p. 71).

[88] Y. Sun et al. “Multi-Feature Encoder for Radar-Based Gesture Recognition”. In: *2020 IEEE International Radar Conference (RADAR)*. 2020, pp. 351–356. DOI: 10.1109/RADAR42522.2020.9114664 (cit. on p. 71).

[89] W. Lei et al. “Continuous Gesture Recognition Based on Time Sequence Fusion Using MIMO Radar Sensor and Deep Learning”. In: *Electronics* 9.5 (2020). DOI: 10.3390/electronics9050869 (cit. on p. 71).

[90] Z. Chen et al. “Dynamic Hand Gesture Classification Based on Multistatic Radar Micro-Doppler Signatures Using Convolutional Neural Network”. In: *2019 IEEE Radar Conference (RadarConf)*. 2019, pp. 1–5. DOI: 10.1109/RADAR.2019.8835796 (cit. on p. 71).

[91] Y. Wang et al. “TS-I3D Based Hand Gesture Recognition Method With Radar Sensor”. In: *IEEE Access* 7 (2019), pp. 22902–22913. DOI: 10.1109/ACCESS.2019.2897060 (cit. on p. 71).

[92] M. Maghoumi and J. J. L. Jr. “DeepGRU: Deep Gesture Recognition Utility”. In: *CoRR* abs/1810.12514 (2018). arXiv: 1810.12514 (cit. on p. 71).

[93] S. Hazra and A. Santra. “Radar Gesture Recognition System in Presence of Interference using Self-Attention Neural Network”. In: *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. 2019, pp. 1409–1414. DOI: 10.1109/ICMLA.2019.00230 (cit. on p. 71).

[94] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. Cambridge, MA: MIT Press, 2016 (cit. on pp. 81, 82).

[95] F. Chollet. *Deep Learning with Python*. Shelter Island, NY: Manning Publications, 2017 (cit. on pp. 81, 82).

[96] R. Cahuantzi, X. Chen, and S. Güttel. “A Comparison of LSTM and GRU Networks for Learning Symbolic Sequences”. In: *Intelligent Computing*. Springer Nature Switzerland, 2023, pp. 771–785. DOI: 10.1007/978-3-031-37963-5_53 (cit. on p. 102).

[97] J. Tompson et al. “Efficient Object Localization Using Convolutional Networks”. In: *CoRR* abs/1411.4280 (2014). arXiv: 1411.4280 (cit. on p. 103).

[98] S. Raschka, Y. Liu, and V. Mirjalili. *Machine Learning with PyTorch and Scikit-Learn: Develop Machine Learning and Deep Learning Models with Python*. Packt Publishing, 2022 (cit. on p. 119).

List of Figures

1.1 Applications of a 77 GHz FMCW radar for vital sign detection and hand gesture classification. 3

2.1 Mind map outlining the structure and components of the recent research efforts in FMCW radar-based vital signs monitoring. 9

3.1 Block diagram of the human target identification process. 18
3.2 Flowchart illustrating the RAM generation process for all chirp signals within a measurement cycle. The process involves iterative chirp capture, formatting, and Fourier transforms to produce RAMs. 20
3.3 Flowchart illustrating DC-value suppression concept for human identification. All matrix operations are element-wise. 23
3.4 Data acquisition setup with a single human subject: the RBK2 and corner cubes (CC1 and CC2) are placed on tripods for stability. . . . 25
3.5 Range-profile of the test scenario illustrated in Fig. 3.4, obtained from one of the receiving antennas. While the positions of the corner cubes (CC1 and CC2) are highlighted, the human subject's location remains unclear. 26
3.6 RAM of the test scenario shown in Fig. 3.4, generated by the FFT method: (a) prior to applying the DC-value suppression technique; (b) following the DC-value suppression technique 27
3.7 RAM of the test scenario shown in Fig. 3.4, constructed using the beamforming method (Algorithm 1): (a) before applying the DC-value suppression technique; (b) after applying the DC-value suppression technique. 28
3.8 Empirical results of the test scenario shown in Fig. 3.4: (a) the raw RSM; (b) the RSM after DC-value suppression along the slow-time axis; (c) the processed ASM constructed using the FFT-based approach; (d) the processed ASM generated via beamforming. 29

3.9 One-dimensional version of the curve-length metric proposed in [21], demonstrating the range of the identified human subject as 2.9 m, consistent with the range detected in Fig. 3.8b 30
3.10 Trajectory plot showing the effect of DC-value suppression. Green markers indicate the test participant's location, while the magenta point denotes CC1. 30
3.11 Schematic illustration of the proposed vital sign estimation block. . . 32
3.12 FIFO buffer mechanism in our real-time vital sign monitoring system, using a sliding window to ensure timely detection of physiological changes. 33
3.13 Visualization of the peak-finder module, processing the averaged magnitude matrix $\bar{\mathbf{\Pi}}$ defined in Equation (3.13) to identify peaks, storing their values and range-angle bin indices (p, q) in $\mathcal{H}$. 34
3.14 A set of unwrapped phase values across all $M_{\text{FIFO}} = 400$ chirps of the FIFO buffer, computed for indices in $\mathcal{S}_{\text{final}}$. 37
3.15 Visualization of PCA concept: (a) a 2D dataset with two axes representing principal components; (b) projection onto the first principal component (dotted line), capturing the maximum variance; (c) projection onto the second principal component (dashed line), retaining significantly less variance. 38
3.16 Sample chest-wall displacement signal derived by scaling the combined phase signal from unwrapped phase data $\hat{\mathbf{\Phi}}$ across $M_{\text{FIFO}} = 400$ chirps. 40
3.17 The explained variance and cumulative variance for a combined phase signal. The red dashed line marks the point where 95% of the variance is retained. 41
3.18 (a) Time-domain representation of the filtered breathing signal, showing the amplitude variations over time; (b) Frequency spectrum of the filtered breathing signal, displaying the PSD. The peak in the PSD indicates the estimated BR, marked at approximately 17.6 BPM. . . . 43
3.19 Frequency response of the comb filter designed to attenuate harmonics of the breathing signal, based on a breathing rate of 17.6 BPM. The pins in the figure represent the first few harmonics of the BR, which fall within the range of 0.9 to 2.5 Hz. 44
3.20 Time-domain representation of the heart signal after applying the band-pass and comb filters. 45
3.21 HR estimation using the single-point method across two processing intervals: (a) PSD during the first interval; (b) PSD during the second interval. The single-point method initially fails to accurately estimate the HR in the first processing interval due to the presence of two dominant peaks. However, it effectively estimates it in the second interval. 46
3.22 HR estimation using PCA across two processing intervals: (a) PSD computed using PCA in the first interval; (b) corresponding PSD in the second interval. The PCA-based approach consistently yields accurate estimations of HR across both processing intervals. 47

3.23 Positions of the test subject for different combinations of range (r) and angle (θ), with red circles representing the test subject's locations 48
3.24 EQ02+ LifeMonitor chest strap sensor by Equivital, depicting the sensor belt and the SEM for processing physiological data[54]. . . . 48
3.25 Comparison of BRs obtained from the RBK2 and the EQ02+ LifeMonitor sensor measurements at a range of $r = 0.75$ m and an angle of $\theta = 0°$. The performance of the proposed method is highlighted against the methods presented in [3, 14]. 50
3.26 Comparison of HRs obtained from the RBK2 and the EQ02+ LifeMonitor sensor measurements at a range of $r = 0.75$ m and an angle of $\theta = 0°$. The performance of the proposed method is highlighted against the methods presented in [3, 14]. 51
3.27 Comparison of BR measurements between our radar-based proposed method and EQ02+ LifeMonitor sensor at $r = 0.75$ m and $\theta = 0°$. Each marker represents a data point from the experiment, with jitter added for better visualization. The diagonal line indicates perfect agreement ($x = y$). 52
3.28 Comparison of HR measurements between our radar-based proposed method and EQ02+ LifeMonitor sensor at $r = 0.75$ m and $\theta = 0°$. Each marker represents a data point from the experiment, with jitter added for better visualization. The diagonal line indicates perfect agreement ($x = y$). 52
3.29 3D visualization of MAPE calculated across all examined combinations of range and angle (r, θ), depicting the accuracy of vital sign measurements. 54

4.1 A multi-stage pipeline for FMCW radar-based HGR, featuring gesture collection, signal processing, deep learning, and classification stages. 59
4.2 Conceptual workflow of FMCW radar data processing for HGR. Raw radar data is transformed into a RDM using 2D-FFT and a RAM via beamforming. 60
4.3 Radar-based gesture recognition: (a) Single-hand gesture with corresponding RDM and RAM; (b) Multi-hand gesture; (c) Spatiotemporal tracking of two hands over time, demonstrating signal separation. The depicted RDM and RAM are illustrative examples, not derived from empirical data, and are intended to clarify the conceptual workflow. . 66

5.1 Schematic illustration of the proposed radar-based hand gesture recognition system. 70
5.2 The RDM and RAM before and after the clutter removal at the same chirp frame: (a) before; (b) after applying clutter removal technique 74
5.3 Energy, EMA, and threshold calculations performed over 150 frames: (a) RDM; (b) RAM. The time frames where energy is above the threshold are highlighted. 78
5.4 Initialization and data collection stage of hand activity detector . . . 79
5.5 Gesture detection logic of the hand activity detector illustrated in Fig. 5.4 81

5.6 The 10 hand gesture types used in the proposed HGR system: (a) Push (PS); (b) Pull (PL); (c) Push-Pull (PSPL); (d) Pull-Push (PLPS); (e) Swipe Left (SL); (f) Swipe Right (SR); (g) Clockwise Rotation (CW); (h) Counter-Clockwise Rotation (CCW); (i) Pinch Open (PO); (j) Pinch Close (PC) 82
5.7 Trajectory plot of PS, PL, PSPL, PLPS, SR, and SL gestures over 15 time frames: (a) PS; (b) PL; (c) PSPL; (d) PLPS; (e) SR, (f) SL 83
5.8 Overview of the proposed dual-stream CNN-GRU model architecture for processing RDMs and RAMs. In this visualization, the CNN-GRU model is represented as a black box, where '**n**' signifies the number of output neurons from the final GRU layer. 85
5.9 Heatmap showing the validation accuracy as a function of the number of convolutional layers and GRU layers. The color intensity represents the accuracy, with brighter colors indicating higher accuracy. 87
5.10 Validation accuracy across different GRU layer depths and GRU unit configurations. 88
5.11 Validation accuracy as a function of the number of convolutional layers and GRU units. 90
5.12 Validation performance visualized across models with different numbers of dense layers: (a) box plot summary; (b) individual trial results by dense units. 91
5.13 Layer-wise architecture of the final CNN-GRU model, identified through hyperparameter optimization for the radar-based HGR. 93
5.14 Average confusion matrix for LOOCV with the CNN-GRU model using trial 35 hyperparameters (see Table 5.2), illustrating classification performance across 10 gesture classes. 97
5.15 Average confusion matrix for LOOCV with the CNN-LSTM model using trial 35 hyperparameters (see Table 5.2), illustrating classification performance across 10 gesture classes. 98
5.16 Per-class accuracy comparison between CNN-GRU and CNN-LSTM models using trial 35 hyperparameters, highlighting CNN-GRU's superior performance across most gestures. 99
5.17 Illustration of the radar system's sensing geometry and Doppler sensitivity. The radar is placed horizontally on a tabletop, facing upward (along the Z-axis), with a 1D antenna array aligned along the X-axis. Doppler sensitivity is strongest along the antenna array direction (green arrows) and weakest in the orthogonal direction (orange dashed arrows). 100
5.18 Test accuracy comparison for CNN-GRU and CNN-LSTM models across subjects using LOOCV with trial 35 hyperparameters. 101
5.19 Training and validation accuracy and loss metrics for the CNN-GRU model over 60 epochs, derived from a LOOCV iteration for subject 6. 102
5.20 Training and validation accuracy and loss metrics for the CNN-LSTM model over 60 epochs, derived from a LOOCV iteration for subject 6. 103

A.1 3D Representation of MIMO radar data cube 112

B.1 Radarbook2 evaluation platform [46] 115
B.2 Array configuration showing TX array, RX array, and virtual array with inter-element spacing $d_t = 7.5\lambda$ and $d_r = 0.5\lambda$. 116
B.3 A time-division multiplexed waveform for the RBK2 MIMO configuration, illustrating alternating chirp sequences for TX1 (blue) and TX2 (red) over a time frame with N_P chirps, each separated by T_P, and a chirp duration of T_{up}. 117

C.1 The output of a 2D convolution 120
C.2 Max pooling operation with a 2×2 window and stride of 2, reducing a 4×4 feature map to a 2×2 output. Red dashed lines show the mapping for the top-right region, and blue dashed lines show the mapping for the bottom-left region. 121
C.3 An LSTM cell 123
C.4 GRU cell 124

List of Tables

2.1 Grid search results for optimal K and α selection, highlighting $K = 3$ and $\alpha = 300$ with the lowest energy loss ($\psi_{K,\alpha} = 0.06$). 10

3.1 Simulation parameters for radar-based human identification on RBK2 25
3.2 Notch Frequencies for BR of 17.6 BPM 44
3.3 Comparison between the proposed and conventional methods [3, 14] in terms of MAPE, PCC (ρ), and statistical significance using Fisher's z-test for HR and BR. These results are associated with the data presented in Fig. 3.26 and Fig. 3.25. 53
3.4 MAPE for different angles and ranges associated with Fig. 3.29 . . . 54

4.1 Comparison of classification techniques for FMCW radar-based HGR across studies 65

5.1 Simulation parameters for radar-based HGR 73
5.2 Top hyperparameter configurations and their performance 92
5.3 Tensor dimension changes through neural network layers for RDM . 94
5.4 Tensor dimension changes through neural network layers for RAM . 94
5.5 Comparison of LOOCV performance metrics for CNN-GRU and CNN-LSTM models using trial 35 hyperparameters. 96
5.6 Per-class accuracies derived from the average confusion matrices for CNN-GRU and CNN-LSTM models using trial 35 hyperparameters. . . 99