UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

EVALUACIÓN DEL EFECTO DE *VISUAL SERVOING* EN *SPEECH ENHANCEMENT* CON ARREGLO DE MICRÓFONOS LINEAL EN INTERACCIÓN HUMANO-ROBOT MOVIL.

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN ELÉCTRICA

ALEJANDRO PATRICIO DÍAZ ALBORNOZ

PROFESOR GUÍA:
Néstor Becerra Yoma

MIEMBROS DE LA COMISIÓN:
César Azurdia Meza
Miguel Carrasco Zambrano

SANTIAGO DE CHILE
2022

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE MAGÍSTER EN CIENCIAS
DE LA INGENIERÍA, MENCIÓN ELÉCTRICA.
POR: **ALEJANDRO PATRICIO DÍAZ ALBORNOZ**
FECHA: 2022
PROF. GUÍA: Néstor Becerra Yoma

EVALUACIÓN DEL EFECTO DE *VISUAL SERVOING* EN *SPEECH ENHANCEMENT* CON ARREGLO DE MICRÓFONOS LINEAL EN INTERACCIÓN HUMANO-ROBOT MOVIL.

Este trabajo de tesis estudia la comunicación por voz entre humano y robot en ambientes acústicos desafiantes en contexto de interacción humano robot. En un primer capítulo se estudia el efecto de usar *visual servoing* con *beamforming* para mejorar el reconocimiento de voz en un sistema de reconocimiento automático de voz. Se grabaron señales de audio con una plataforma robótica en un escenario real con diferentes condiciones de ruido adicional y de movimiento del robot para simular una interacción real. Mediante el uso de información visual, se controló parte de los movimientos de la plataforma robótica para obtener un filtrado espacial mejorado, usando *beamforming*. Esto se logró manteniendo un arreglo de micrófonos lineal cercano a la posición donde se obtiene la mejor respuesta. Luego en un siguiente capítulo, se implementó una red neuronal convolucional para enfrentar el problema de *speech enhancement* para obtener una estimación limpia de la voz a partir de señales voz ruidosas y reverberantes. Este capítulo está basado en el escenario del capítulo anterior donde se considera la plataforma robótica en un ambiente acústico dinámico, ruidoso y reverberante. En este capítulo se realizaron simulaciones y se crearon bases de datos representativas del problema. Utilizando una mezcla de señales obtenidas a través de *beamforming* se realizó *speech enhancement* a las señales para obtener una versión limpia de la señales reverberantes.

Le dedico este trabajo a mis padres, a mi hermano,
a mi hijo y a mi futura esposa.

Agradecimientos

Agradezco a mi familia, a mis amigos y a todas las personas que conocí durante de mis estudios.

Gracias por la fraternidad y sabiduría de los integrantes senior del laboratorio LPTV: colega Jorge Wuth y maestro Rodrigo Mahu. También agradezco la compañía y amistad de Diego Pincheira, Eduardo Alvarez, Nicolas Grajeda, Alejandro Luzanto, Javier Mosnaim y Franco Claverie.

Agradezco al Profesor Néstor Becerra por enseñarme con su ejemplo las virtudes necesarias para recorrer el hermoso mundo de la investigación.

Agradezco a la Universidad de Chile y todos los docentes que conocí y me entregaron su dedicación y conocimiento. Finalmente agradezco a ANID(Ex-Conicyt) por entregarme la posibilidad de estudiar gracias a la beca: CONICYT-PFCHA/MagísterNacional/2019-22191373.

Agradezco especialmente a los profesores César Azurdia y Miguel Carrasco por sus valiosas observaciones para mejorar este texto.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

La comunicación entre humanos tiene muchas dimensiones como el vocabulario, jerarquía, confianza, empatía y contexto. Dimensiones que deben ser consideradas para que la comunicación sea efectiva y natural. Como se menciona en [1]: "[las personas] deberían poder comunicarse con la tecnología tal como están acostumbrados a hacerlo con el mundo real todos los días, tal como la evolución y la educación les enseño a hacerlo". El habla es la forma más directa y natural de comunicarnos entre humanos, y lo debe ser entre humanos y robots.

La cooperación entre humanos y maquinas mejoraría sustancialmente si los robots estuviesen integrados socialmente. Esta integración social se está convirtiendo en una realidad y tiene múltiples aplicaciones en muchos escenarios. La colaboración entre humanos y robots será requerida[2], en escenarios como por ejemplo en desastres naturales, minería, industrias y otros ambientes hostiles. El área académica y tecnológica HRI (Human-robot interaction) es especialmente relevante para establecer una comunicación con los humanos que permita entregar instrucciones, información y también tomar decisiones, así establecer una simbiosis con comunicación natural y fluida entre humanos y robots [3, 4, 5, 6].

Naturalmente los robots humanoides están dotados de sensores que emulan nuestros sentidos humanos. Poseen cámaras, micrófonos, sensores táctiles y de posición entre otros. En los humanos, los sentidos operan de manera conjunta y la información de un fenómeno o evento se entiende de mejor manera si somos capaces de percibirla mediante varios sentidos a la vez. Así mismo los sensores de los que están dotados entregan información que es posible combinar y utilizar de manera conjunta, esto con el objetivo de mejorar el desempeño de funciones superiores que puede tener un robot, como, por ejemplo: reconocimiento automático de la voz (ASR - Automatic Speech Recognition), planificación del movimiento o planificación del dialogo entre muchos más.

1.1. Hipótesis

En esta investigación se establecen tres hipótesis en campo de HRI:

- Se puede mejorar el desempeño de un sistema ASR controlando la dirección hacia donde mira un robot mediante el uso de información multimodal.
- La directividad del *beamforming* depende de la dirección hacia donde mira un arreglo de micrófonos.

- La combinación unimodal de señales de *beamforming* puede mejorar el desempeño de un sistema de *speech enhancement*.

1.2. Objetivos

Para probar estas hipótesis, se ha establecido un objetivo general y tres objetivos específicos.

1.2.1. Objetivo General

Mejorar en términos de WERs el reconocimiento de voz en escenarios HRI usando información multimodal y unimodal.

1.2.2. Objetivos específicos

- Mejorar el desempeño de varios métodos de *beamforming* haciendo uso de *visual servoing* en una plataforma robótica real.
- Estudiar la dependencia de la directividad del *beamforming* con respecto al ángulo donde mira un arreglo de micrófonos
- Implementar un sistema de *deep-learning* capaz de combinar señales de voz unimodales, provenientes de múltiples *beamformings*

1.3. Estructura de tesis

En el capítulo 2 se presenta un estudio del efecto de *visual servoing* sobre el desempeño de un sistema ASR en un escenario acústico dinámico y móvil. Se implementó un sistema de *visual servoing*, empleando la imagen obtenida con una cámara Kinect sobre la cabeza de un robot PR2 y basándose en un reconocedor de objetos (YOLO) que utiliza redes neuronales convolucionales. El sistema de *visual servoing* es capaz de controlar la cabeza del robot y seguir un objeto específico con la mirada, esto implica mantener el arreglo de micrófonos en una posición óptima. Se montó un escenario experimental para la grabación de una base de datos, considerando una o dos fuentes de ruido y a la vez considerando el uso de *visual servoing* o no. La experimentación se realizó con escenarios de menor a mayor dificultad, y que permitiera el estudio del impacto de usar *visual servoing* sobre un arreglo de micrófonos.

En el capítulo 3 se presenta una red neuronal convolucional que se combina con *compact bilinear pooling* para resolver la tarea de *speech enhancement* en el contexto de interacción humano-robot mediante comunicación por voz. En este capítulo se realizaron simulaciones para obtener una base de datos de escenarios acústicos móviles y reverberantes. Las simulaciones se realizaron considerando un arreglo lineal de micrófonos, una fuente de ruido y una de voz, y así también movimientos aleatorios del arreglo de micrófonos. La red neuronal se implementó considerando como entradas dos señales de *beamforming* dirigidas hacia la fuente de voz y hacia la fuente de ruido. La salida de la red neuronal es una estimación de la señal de voz limpia sin ruido ni reverberación.

Finalmente, en el capítulo 4 se entregan conclusiones considerando los objetivos de esta tesis.

Capítulo 2

Visual Servoing con arreglo de micrófonos

2.1. Introducción

El ruido y la interferencia acústica afectan el desempeño de los sistemas de reconocimiento automático de la voz (ASR Automatic Speech Recognition), una forma de efectiva de filtrar espacialmente la interferencia y ruido es el *beamforming*. Los métodos de *beamforming Delay-and-sum* y MVDR (*Minimum Variance Distortionless Response*) sirven para mejorar la supresión espacial del ruido. MVDR es un método adaptativo que suprime el ruido correlacionado, ponderando específicamente las señales interferentes sin afectar la ganancia en la dirección de la fuente acústica de interés [7]. Y *Delay-and-sum* es un método que desfasa las señales según el tiempo de llegada para apuntar el *beamforming* hacia la fuente acústica de interés.

MVDR requiere una estimación de la matriz de covarianza y es adecuado para condiciones donde la fuente acústica de interés se encuentra estática, como en una sala de reuniones por ejemplo [8] donde los hablantes no se mueven y las fuentes de ruido también son estáticas [9, 10, 11], en consecuencia, la distancia y posición relativa entre los hablantes y las fuentes de ruido se mantiene constante. En aplicaciones de interacción humano-robot si existe movimiento relativo de los hablantes y fuentes de ruido, así MVDR y también otros métodos de *beamformig* no son directamente aplicables. Esto debido a que la matriz de covarianza del ruido será dependiente del tiempo si la posición y distancia relativa entre el hablante y el robot es variante en el tiempo.

Los métodos de *beamforming* requieren naturalmente un arreglo de micrófonos para funcionar, los arreglos de micrófonos lineales tienen una sensibilidad que depende de la posición angular entre la dirección del *beamforming* y el eje de mayor respuesta (MRA - *Main Response Axis*) [12], como se puede apreciar en la figura 2.1 la dirección del *beamforming* corresponde hacia donde está apuntando el lóbulo principal del *beamformig*. El MRA corresponde a la dirección donde la sensibilidad del arreglo de micrófonos es mayor [13]. Para el arreglo de micrófonos presentado en la figura 2.1, el MRA coincide con la dirección *broadside* [14], es decir, ortogonal al eje que cruza todos los micrófonos (el cual se conoce como *endfire*).

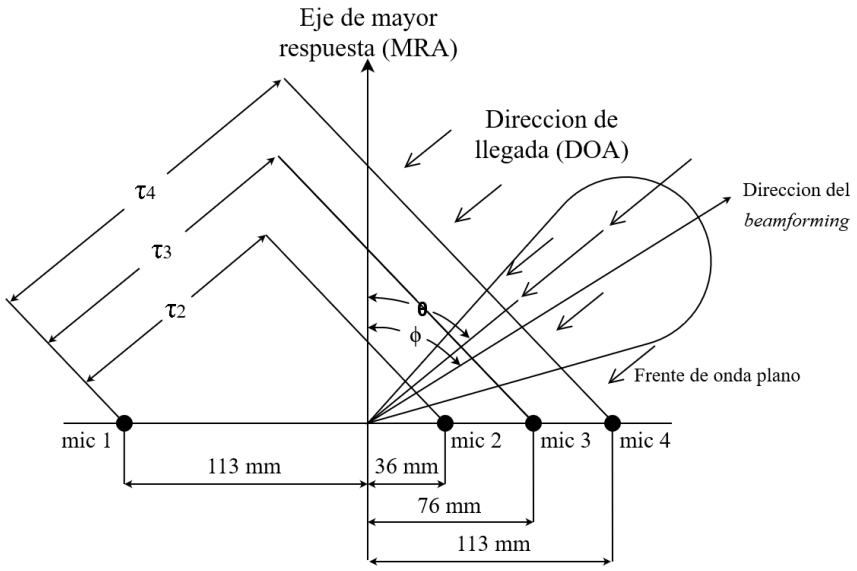


Figura 2.1: Geometría del arreglo de micrófonos del dispositivo Kinect de Microsoft, los desfases de cada canal, el MRA, el DOA, el AOI y un lóbulo de *beamforming* hipotético con su respectiva dirección θ. También se observa la posible diferencia entre el AOI y θ.

En escenarios de interacción humano robot, es posible utilizar el control de los servomotores de un robot humanoide para mover el MRA hacia la fuente de interés. Esto se puede hacer mediante el uso de información visual a través de un lazo de control cerrado, lo cual se conoce como *visual servoing* [15]. En este capítulo se estudia el efecto de minimizar el ángulo entre la dirección del *beamforming* y el MRA para mitigar el ruido espacialmente correlacionado. Para lograr esto, se utiliza un esquema de *visual servoing* para mover la cabeza de un robot y hacerlo seguir una fuente acústica de interés con su cámara y micrófonos. Considerando la relevancia de HRI y las aplicaciones de voz, es importante estudiar este efecto en una plataforma robótica real.

Para cuantificar las mejoras del sistema presentado aqui se utiliza la métrica WER (Word Error Rate), la cual es ampliamente utilizada como métrica de desempeño de sistemas ASR. A partir de una frase reconocida por un ASR, el calculo del WER consiste en el conteo de errores de inserción, substitución u omisión de palabras dividido por el numero de palabras que tiene la frase de referencia. Estos errores se detectan mediante la alineación de la frase reconocida con la frase de referencia. El WER se uso en este trabajo porque permite la comparación entre sistemas similares [16] y también que otros trabajos se comparen con este en el futuro debido a su uso generalizado en sistemas de ASR. Finalmente, en la siguiente sección se mencionan otras métricas que también se usan en sistemas ASR pero en menor medida y se discute sobre lo que significa usar estas métricas como medida del desempeño de la comunicación en HRI.

El objetivo principal de este capítulo es estudiar el efecto de *visual servoing* en el desempeño de un micrófono lineal en el contexto de ASR en un ambiente de HRI móvil, dinámico y no estacionario. También se estudia la dependencia de la directividad del *beamforming* con respecto a dirección donde mira el *beamforming* y su desviación del MRA. Estos objetivos se alcanzan haciendo uso de una plataforma robótica real y apropiada para realizar experi-

mentos de forma controlada. Con el fin de generar una base de datos adecuada, se montaron escenarios experimentales con el robot PR2 junto con fuentes de voz y ruido. Además, la Kinect de Microsoft se elijo como un caso de estudio de arreglo de micrófonos lineales y su respuesta al impulso fue medida en una cámara anecoica. Luego varios métodos de *beam-forming* fueron probados, específicamente *delay-and-sum*, *weighted delay-and-sum* y MVDR. *delay-and-sum* y MVDR fueron mejorados utilizando el esquema de *visual servoing* y también usando rastreo de objetos. Los resultados muestran una reducción máxima en WER de un 28.2 %, comparando el WER cuando se usa *visual servoing* y cuando no se usa.

2.2. Trabajos relacionados y marco teórico

2.2.1. Reconocimiento automático de la voz e interacción humano-robot

Para que la comunicación entre humanos y robot sea exitosa, es necesario que los robots se comuniquen de manera naturalmente humana [17, 18]. Considerando que los humanos nos logramos comunicar mediante el habla incluso en los ambientes acústicos más difíciles, es decir, nuestra capacidad de reconocimiento de la voz y selectividad auditiva es mas adaptable que los sistemas implementados para este propósito [19]. Así, HRI puede involucrar ASR en un escenario realista que se caracteriza por la presencia de ruido y un canal acústico dinámico, todo esto afecta la efectividad de un sistema ASR, incluso el ruido producido por los mismos motores del robot. Este problema ha sido estudiado en [20], donde los autores proponen un sistema ASR integrado para aplicaciones HRI donde se considera el canal acústico variable y el ruido producido por los motores de un robot.

Como se menciona en la introducción en este trabajo se utiliza la métrica WER para medir las mejoras sucesivas del sistema. Además de esta métrica también existen otras que son utilizadas en menor medida, como SER(Sentence Error Rate), y PER(Phoneme Error Rate). PER entrega una resolución mas fina para discriminar errores, por ejemplo si un sistema ASR predice la palabra "comida" y la referencia es "comino" el WER seria de 100 % mientras que el PER seria de 33 %, sin embargo una de mayores las dificultades de trabajar con PER es la alineación [21], lo cual puede inducir errores adicionales. Por otro lado el SER funciona a nivel de oraciones completas, por lo que se puede dar el caso de tener un SER de 100 % con un WER de 5 % que corresponde a errar una de cada veinte palabras por oración. La métrica SER apunta medir la consistencia semántica de lo que reconoce un sistema ASR, sin embargo también puede declarar como errónea dos oraciones distintas pero con la misma semántica como por ejemplo: "vamos a reunirnos hoy" y "vamos nos reuniremos hoy". Estas tres métricas son útiles en caso de querer comparar mejoras sucesivas sobre un mismo sistema, como es el caso del trabajo presentado aquí. No obstante, esta métrica y así también las otras tienen limitaciones ya que los fonemas, las palabras y las oraciones son solo una parte de la comunicación. Tampoco estas métricas entregan una medida de la capacidad de generalización que tiene un sistema ASR y cuales son sus limitaciones en un ambiente no controlado. Esto implica que un sistema con un WER reportado de 2 % a 3 % hace parecer que el problema de ASR esta casi resuelto pero ciertamente no es así [19].

Beamforming ha sido una técnica ampliamente utilizada y estudiada en la literatura para enfrentar escenario acústicos variables en tiempo. En [22], se propuso un método de *beamforming* para este tipo de escenarios, donde se modelan fuentes acústicas móviles con una convolución variable en el tiempo que requiere conocimiento de la respuesta al impulso de la sala en cada momento, el método fue probado usando simulaciones numéricas. Otro trabajo interesante es presentado en [23] donde se diseña un arreglo esférico de micrófonos que optimiza la ganancia de los lóbulos secundarios y maximiza la ganancia del lóbulo principal. En [24], se estudia el problema de las fuentes acústicas móviles mediante un método de *beamforming* que mejora la ganancia en una zona espacial, al contrario de un *beamforming* altamente directivo el esquema presentado considera una zona de interés más amplia para mejorar el desempeño en un escenario no estacionario. De todas formas, la técnica pierde

efectividad cuando hay fuentes de ruido dentro de la zona espacial de interés. Hasta donde conoce el autor de este trabajo, el uso de *beamforming* en conjunto con *visual servoing* no ha sido objeto de estudio en la literatura.

2.2.2. *Weighted delay-and-sum* y arreglos lineales de micrófonos

Un arreglo de micrófonos consiste en un numero arbitrario de micrófonos cuyas salidas pueden ser procesadas y combinadas para conseguir filtrado espacial a través de *beamforming*. El uso de arreglo de micrófonos para realizar *beamforming* puede reducir el efecto del ruido y la reverberación. Para el caso de la reverberación, el filtrado espacial ayuda a suprimir las señales interferentes que no provienen del camino directo entre la fuente y el arreglo de micrófonos [25]. Este trabajo se enfoca en un arreglo lineal de micrófonos y emplea la Kinect de Microsoft como un caso de estudio particular, la cual es usada ampliamente en aplicaciones de HRI y ASR [26]. La Kinect de Microsoft tiene un arreglo lineal de micrófonos de cuatro canales como se aprecia en la figura 2.1. El principio de la diferencia en el tiempo de llegada se refiere al tiempo que transcurre entre que el frente de onda pasa un micrófono y pasa por un punto de referencia. Este principio es aplicable a un arreglo arbitrario de micrófonos, así, este principio se ilustra a continuación.

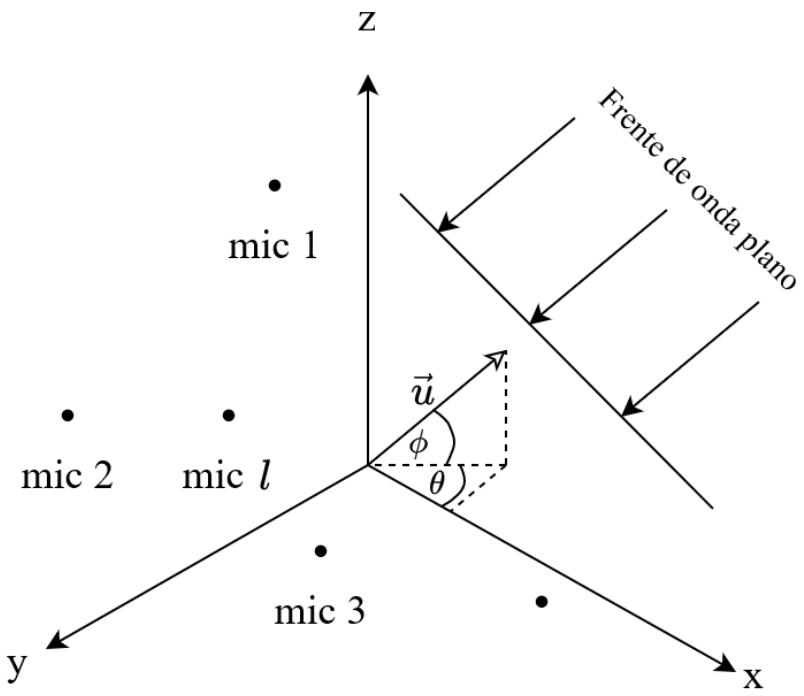


Figura 2.2: Arreglo de micrófonos arbitrario con frente de onda plano y caracterización de la dirección del frente de onda como un vector unitario descrito en el espacio en coordenadas polares

La figura 2.2 muestra un arreglo de L-micrófonos ubicados en el espacio, un frente de onda plano llega al arreglo en la misma dirección que el vector unitario $\vec{u}$ el cual es caracterizado por las coordenadas polares θ y ϕ. Se puede asumir un frente de onda plano si la distancia

entre el arreglo de micrófonos y la fuente acústica es mayor a 5-10 veces el largo del arreglo de micrófonos [13]. Se puede caracterizar el conjunto de señales que son captadas por el arreglo como una función dependiente del tiempo y la posición de cada micrófono. Así tenemos que:

$$\vec{f}(t,\vec{p}) = \begin{pmatrix} f(t-\tau_0) \\ f(t-\tau_1) \\ \vdots \\ f(t-\tau_{L-1}) \end{pmatrix} \tag{2.1}$$

donde τ_l corresponde al desfase de cada señal con respecto a un punto de referencia, c es la velocidad del sonido, y $\vec{p}_n$ es la posición del micrófono n. La onda plana se propaga en la misma dirección y sentido opuesto de $\vec{u}$, así, la dirección de propagación es $\vec{a}$ donde $\vec{a} = -\vec{u}$. Si presentamos el vector $\vec{a}$ en coordenadas cartesianas tenemos que:

$$\vec{a} = \begin{pmatrix} -cos(\phi)cos(\theta) \\ -cos(\phi)sen(\theta) \\ -sen(\phi) \end{pmatrix} \tag{2.2}$$

El retardo temporal para el elemento l con respecto al origen del arreglo esta dado por:

$$\begin{aligned} \tau_l &= -\tfrac{1}{c}[cos(\phi)cos(\theta)\cdot p_{x_l} + cos(\phi)sen(\theta)\cdot p_{y_l} + sen(\phi)\cdot p_{z_l}] \\ &= \tfrac{1}{c}[a_x\cdot p_{x_l} + a_y\cdot p_{y_l} + a_z\cdot p_{z_l}] \\ &= \tfrac{\vec{a}^T\vec{p}_n}{c} \end{aligned} \tag{2.3}$$

Para el caso de la Kinect los micrófonos están alineados en un eje lo que simplifica la eq. 2.3, ya que $p_{x_l} = p_{y_l} = 0$. Referenciado a la figura 2.1 el desfase para cada micrófono esta dado por:

$$\tau_l = \frac{\Delta_l \cdot \sin\phi}{c} \tag{2.4}$$

donde Δ_l es la distancia entre el micrófono l y el micrófono de referencia. ϕ es el ángulo de interés (AOI - *Angle of Interest*) el cual corresponde a el ángulo entre la dirección *broad-side* del arreglo o el MRA en el caso de la Kinect y el DOA de interés. Y finalmente c es la velocidad de propagación del sonido en el medio [14].

Delay-and-sum es una técnica de *beamforming* ampliamente estudiada cuyo flujo es el siguiente: dada una dirección de la fuente acústica (DOA - *Direction of Arrival* ver Fig. 2.1) y un tiempo de llegada de la onda acústica en cada canal, *delay-and-sum* desfasa las señales

según el tiempo de llegada para apuntar el *beamforming* hacia la fuente acústica de interés. Esto produce un efecto destructivo de ondas en todas las direcciones excepto en la dirección de interés en donde se produce un efecto constructivo. *Weigthed delay-and-sum* es una generalización de *delay-and-sum* donde se tiene que para la muestra $x_l(t)$ de cada señal de cada micrófono l se aplica un desfase de τ_l muestras multiplicado por un peso $w_l(t)$ y luego se suman. Así, la señal resultante en el dominio del tiempo corresponde a:

$$y(t) = \sum_{l=0}^{L-1} w_l \cdot x_l(t - \tau_l) \tag{2.5}$$

donde L es el número de micrófonos. Y para el caso particular de *delay-and-sum* se tiene que w_l es igual a $1/L$. Si el DOA de interés es desconocido, el AOI y los desfases temporales τ_l son estimados mediante correlación cruzada en una ventana de análisis [27]. Finalmente, el detalle de la implementación del algoritmo *delay-and-sum* usado en este trabajo se muestra al final de esta sección.

Al aplicar *beamforming* con un arreglo de micrófonos lineal, la sensibilidad y la posición angular de los lóbulos laterales depende del ángulo entre la dirección de interés y el MRA [28], lo que implica que la directividad del *beamforming* es dependiente del ángulo de interés. Generalmente, y particularmente en HRI este problema no es considerado en la literatura.

Para evaluar el método *weigthed-delay-and-sum*, se utilizó el *toolkit* BeamformIt [27] . Esta es una herramienta ampliamente validada para la tarea de *meeting diarization*, la cual implementa el método de *weigthed-delay-and-sum* en cuatro partes: primero se aplica un filtro de Weiner a cada canal, con lo cual se busca incrementar el SNR de todos los canales. Segundo, se extrae información de las señales de entrada para estimar algunos parámetros como el canal de referencia óptimo y los pesos de cada canal, así la segunda etapa hay tres sub-etapas: primero se determina el canal que mejor representa la acústica de la sala y es llamado canal de referencia. Esto se logra mediante el computo de las correlaciones cruzada de las señales enventanadas de los micrófonos. Esta ventana de análisis necesita proveer suficientes muestras para que las correlaciones sean confiables. Para el caso de un ambiente dinámico como un escenario típico de HRI una ventana muy larga es poco estacionaria por lo que se reduce la confiabilidad de las correlaciones estimadas. Por defecto, BeamformIt utiliza una ventana de análisis de 8000 muestras de largo con un traslape del 50 % con una tasa de muestro de 16Khz. La siguiente sub-etapa consiste en determinar los pesos asociados a cada canal para mejorar el rango dinámico de la señal resultante en el *beamforming*. La tercera sub-etapa estima N candidatos (por defecto 4) de retardos o desfases temporales para cada canal. Esto es implementado mediante la maximización de la correlación cruzada entre cada canal y el canal de referencia, específicamente se utiliza GCC-PHAT (*generalized cross-correlation with phase transform*). Después de obtener los N mejores candidatos para cada micrófono viene la tercera etapa la cual consiste en una selección entre los candidatos utilizando un umbral de ruido en primera instancia y luego se utiliza el algoritmo de Viterbi [27, 29, 30] para seleccionar la combinación óptima de retardos para ser usados en cada canal. La última etapa es la que aplica finalmente el algoritmo de *weigthed-delay-and-sum* con los pesos y retardos calculados previamente. Con los pesos se compensan posibles diferencias en el hardware de

los micrófonos y sus respuestas al impulso. Inicialmente estos pesos son uniformemente distribuidos y se adaptan en el tiempo, basándose en la correlación cruzada de cada canal con los otros. Así, las señales son sumadas para obtener la salida del *beamforming*. BeamformIt ha sido validado ampliamente en las tareas relativas a salas de reuniones con múltiples hablantes y donde el AOI debe ser estimado a partir de las señales, sin información visual [31, 32].

Considerando que: $x_l(t)$ es la muestra del l-ésimo canal al instante t, donde $1 \leq t \leq T$, T es el número total de muestras para una *utterance* particular, $1 \leq l \leq L$ y L es el numero total de micrófonos en el arreglo; $x_l(m,i)$ representa la muestra i-esima en el frame m del canal l, donde $1 \leq m \leq M$, M es el numero total de frames para una *utterance* particular, $1 \leq i \leq largoDelFrame$ y $largoDelFrame$ denota el largo de los frames en numero de muestras.

Pseudocódigo 1 D&S-AOI

Requiere $x_1(t), x_2(t), \ldots, x_l(t), \ldots x_L(t)$ { Muestras de la señal de voz para cada canal l}
Requiere $\phi = [\phi_1, \phi_2, \ldots, \phi_M]$ {Un AOI definido para cada frame}
Requiere G {Geometría del arreglo}

para $m = 1 : M$ **hacer**

$[\tau_1, \tau_2, \ldots, \tau_L] = CalcularRetardos(\phi_m, G)$ {Los retardos son estimados a partir del AOI y definidos para cada frame m}

para $i = 1 : largoDelFrame$ **hacer**

$y(m,i) = \sum_{l=1}^{L} x_l(m, i - \tau_l)$ {Se calcula la salida del *beamforming* $y(m,i)$ para la muestra i en el frame m }

terminar para
terminar para

2.2.3. MVDR

Como se menciona anteriormente, el algoritmo de *beamformig* MVDR es utilizado para suprimir adaptativamente el ruido espacialmente correlacionado disminuyendo la ganancia de las señales interferentes sin afectar la ganancia en la señal de interés. Si $x_l(m,i)$ representa la muestra $i-esima$ en el frame m del canal l, donde $1 \leq m \leq M$ y M es el número total de frames para una *utterance* particular, $1 \leq i \leq largoDelFrame$ donde $largoDelFrame$ es el largo de los frames en número de muestras, $X_l(m,\omega)$ se obtiene aplicando la DFT al frame $x_l(m,i)$ y representa el componente en la frecuencia discreta ω en el frame m para el canal l, donde $0 \leq \omega \leq numFreqBins$, $numFreqBins = (DFTsize)/2 + 1$ y $DFTsize$ corresponde a número de muestras usadas por la DFT. Si $N(\omega) = [N_1(\omega)N_2(\omega)...N_l(\omega)...N_L(\omega)]$ representa el ruido espacialmente correlacionado en cada micrófono en el dominio de la frecuencia, MVDR ajusta los pesos de un *beamforming* minimizando la varianza del ruido en la salida de acuerdo a $argmin_w\{w^H \sum_N(\omega)w\}$, donde $\sum_N \omega = E\{N(\omega)N^H(\omega)\}$ y $E\{\cdot\}$ es la esperanza, lo cual lleva a (Eq. 8 in [7]):

$$w^H(\omega) = \frac{(v^H(k,\omega)\sum_N^{-1}(\omega))}{(v^H(k,\omega)\sum_N^{-1}(\omega)v(k,\omega))} \tag{2.6}$$

donde $v(k,\omega) = [e^{-j\omega\tau_0}e^{-j\omega\tau_1}\dots e^{-j\omega\tau_(L-1)}]^T$ corresponde al vector *manifold* del arreglo de micrófonos; k representa el vector perpendicular al frente de onda plano en la dirección de propagación con magnitud $\omega/c = 2\pi/\lambda$ y λ es la longitud de onda que corresponde a ω. En este trabajo, MVDR fue aplicado utilizando el *toolkit* BTK2.0 [33].

Haciendo uso de las definiciones planteadas el final de la sección anterior y el párrafo anterior. Los detalles de la implementación son presentados en el pseudocódigo más adelante. Donde $X_l(m,\omega)$ se obtiene aplicando la DFT al frame $x_l(m,i)$ y representa el componente en la frecuencia discreta ω en el frame m para el canal l, donde $0 \leq \omega \leq numFreqBins$, $numFreqBins = (DFTsize)/2 + 1$ y $DFTsize$ corresponde a número de muestras usadas por la DFT.

2.2.4. Rastreo de objetos y *visual servoing*

Rastreo de objetos significa seguir la posición de un objeto detectado dentro del cuadro de una imagen mientras el objeto se mueve en cuadros de video continuos. Hay múltiples herramientas que pueden ser usadas para rastrear objetos. Una de ellas es YOLO (You Only Look Once), el cual es un detector de objetos en tiempo real. YOLO utiliza *deep-learning* y redes neuronales convolucionales, y a sido ampliamente utilizado para múltiples propósitos como detección y rastreo de organismos marinos [34], también fue utilizado por los ganadores de la Robocup 2017 en Nagoya Japón [35, 36]. La información extraída de la detección de objetos que realiza YOLO también puede ser utilizada para realizar *visual servoing*. Generalmente, utilizar información visual con cámaras para determinar la posición de un objeto es más preciso que estimar la posición espacial mediante señales acústicas con arreglos de micrófonos. El error en la estimación de la posición espacial de una fuente acústica degrada el desempeño de los algoritmos de *beamforming* [37], ya que implica un error entre la dirección hacia donde apunta el *beamforming* y el DOA (θ y ϕ en Fig. 2.1). Utilizar detección de objetos es una técnica interesante para eliminar este error. Incorporar información visual para estimar el DOA no es una técnica original de este trabajo [38, 39], pero estudiar el desempeño de un sistema ASR junto con el uso de rastreo de objetos para estimar el DOA y *visual servoing* para optimizar la ganancia del arreglo de micrófonos es un tema que no ha sido estudiado en la literatura.

2.3. Disposición del escenario de grabación

La disposición de los experimentos realizados en este trabajo está inspirada en el escenario HRI presentado en [20], donde el canal acústico es variable en el tiempo. La disposición experimental puede ser representativa de muchas situaciones en las cuales los humanos podrían interactuar con los robots de manera colaborativa, por ejemplo, en una sala de clases donde el robot puede estar enseñando; en un hospital donde el robot puede cuidar de un paciente,

o en un restaurante donde el robot puede servir a los clientes. En todas estas situaciones se tienen múltiples fuentes de ruido y un canal acústico variable.

Pseudocódigo 2 MVDR-AOI

Requiere $x_1(t), x_2(t), \ldots, x_l(t), \ldots x_L(t)$ { Muestras de la señal de voz para cada canal l}
Requiere $\phi = [\phi_1, \phi_2, \ldots, \phi_M]$ {Un AOI definido para cada frame}
Requiere G {Geometría del arreglo}
Requiere $nf(m) = \begin{cases} Verdadero \text{ si el frame es ruido} \\ Falso \text{ si el frame es voz} \end{cases}$ {Selección de frames de acuerdo a un VAD}

$x_l(m,i) \xrightarrow{\text{DFT}} X_l(m,\omega)$
$contadorFramesRuido = 0$ {Contador de *frames* de ruido}
$\sum_N(\omega) = 0$

para $m = 1 : M$ **hacer**
  si $nf(m) = True$ **entonces**
    para $\omega = 0 : numFreqBins - 1$ **hacer**

      $N(\omega) = [X_1(m,\omega), X_2(m,\omega), \ldots X_L(m,\omega)]$
      $\sum_N(\omega) += N(\omega)N^H(\omega)$

    terminar para

    $contadorFramesRuido += 1$

  terminar si
terminar para

$\sum_N(\omega) /= contadorFramesRuido$
para $m = 1 : M$ **hacer**

  $[\tau_1, \tau_2, \ldots, \tau_L] = CalcularRetardos(\phi_m, G)$ {Los retardos son estimados a partir del AOI y definidos para cada frame m}
  $v(k,\omega) = [e^{-j\omega\tau_1}, e^{-j\omega\tau_2}, \ldots, e^{-j\omega\tau_L}]^T$ {Vector *manifold* del arreglo}
  $w^H(\omega) = \frac{(v^H(k,\omega)\sum_N^{-1}(\omega))}{(v^H(k,\omega)\sum_N^{-1}(\omega)v(k,\omega))}$ {Pesos MVDR}
  $Y(m,\omega) = w^H(\omega)\begin{bmatrix}(X_1(m,\omega) \\ ... \\ X_L(m,\omega)\end{bmatrix}$ {Se aplica el filtro MVDR y se obtiene el frame de salida}

terminar para

$Y(m,\omega) \xrightarrow{\text{IDFT}} Y(m,i)$ {Se aplica la transformada inversa para volver al dominio del tiempo}

2.3.1. Movimientos de robot PR2

El escenario HRI propuesto en este trabajo se muestra en la Fig. 2.3. El robot realiza desplazamientos laterales periódicos entre las posiciones P1 y P3 a una velocidad máxima de $0.45m/s$, aplicando una aceleración y desaceleración constante al llegar y al partir de cada posición. Adicionalmente, la cabeza del robot puede realizar movimientos rotacionales para seguir a la fuente de voz de interés como se describe más adelante. Es importante mencionar que, aunque el desplazamiento del robot fue programado de manera rutinaria, fue necesario introducir correcciones menores para compensar el *drift* natural del movimiento del robot.

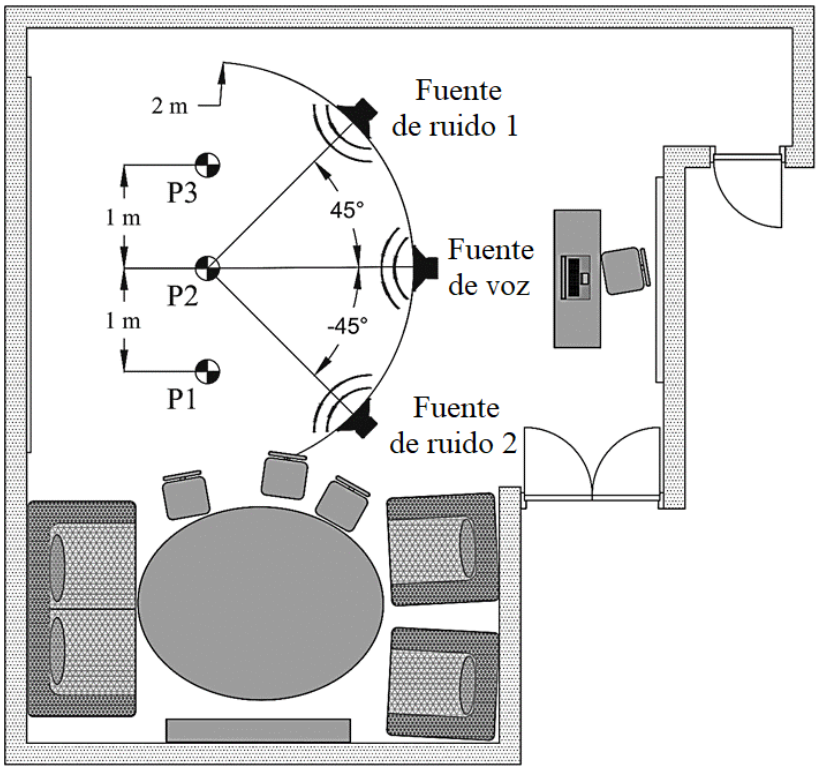


Figura 2.3: Disposición del escenario robótico móvil propuesto para la generación de la base de datos utilizada en este estudio. El robot PR2 realizó movimientos de traslación entre las posiciones P1 y P3 mientras la cabeza se mantenía en dos condiciones: a) estática, mirando hacia adelante; y, b) en movimiento mirando hacia la fuente de voz.

2.3.2. Fuentes de voz y de ruido

La fuente de voz fue ubicada a dos metros de la posición P2 en la Fig. 2.3. Se reprodujo la base de datos de prueba de Aurora-4 [40] con un monitor de estudio. Para evitar interferencia entre *utterances*, se introdujo una pausa de 5 segundos entre una *utterance* y la siguiente. También, se usaron dos condiciones de ruido. La primera utiliza solo una fuente de ruido (Fuente de Ruido 1) ubicada a dos metros de la posición P2 formando un ángulo de 45° con la fuente de voz como se muestra en la figura. La segunda condición de ruido emplea dos fuentes de ruido (Fuente de ruido 1 y Fuente de ruido 2). La Fuente de ruido 2 también fue ubicada a una distancia de dos metros de la posición P2, formando un ángulo

de -45° con la fuente de voz. En ambos casos la calibración de la relación señal a ruido fue realizada en dos pasos. Donde el robot se posicionó en P2 de frente a la fuente de voz en una posición estática. El único ruido que estaba presente en un comienzo era el ruido eléctrico y de los ventiladores del robot. Esto debido a que el robot debía mantenerse estático en cierta posición y para eso debía estar encendido. Primero, la base de datos de prueba limpia se reprodujo de manera completa por el parlante de estudio y fue capturada por el arreglo de micrófonos, no se reprodujo ruido por los demás parlantes en esta etapa. Luego se computó la potencia promedio de toda la grabación y así el SNR resultante fue de 11.8 dB dado el ruido eléctrico y de ventiladores de fondo. Luego, con el robot en la misma posición, un minuto de ruido se reprodujo por los parlantes (para ambas condiciones, una y dos fuentes de ruido) y nuevamente se calculó la potencia promedio de la grabación. En esta etapa no se reprodujo señales de voz. El SNR resultante se calculó con ambas potencias promedios obtenidas, de la señal de voz y de la señal de ruido. Este proceso se iteró variando el volumen de los parlantes hasta que el SNR fuera de 5dB.

2.3.3. Reconocimiento y rastreo de objetos

Para el reconocimiento de objetos rastreo de objetos se usó YOLO, con este se reconoció la fuente de voz y se estimó su posición angular dentro del cuadro con respecto al centro de la imagen (ver figura 2.4). Para detectar la fuente de voz se instaló una señalética personalizada similar al disco pare que pertenece a una de las muchas clases que YOLO puede reconocer con pesos pre-entrenados. YOLO entrega las coordenadas de un recuadro que contiene al objeto dentro de la imagen. El centro de este recuadro corresponde con la posición angular del objeto detectado, lo cual se considera la posición angular objetivo. Como se puede apreciar en la Fig. 2.4 la señalética está posicionada encima del parlante de estudio y esta verticalmente alineada con el centro de la fuente de voz. La posición horizontal se refirió siempre a la posición angular de la señalética θ en Fig. 2.1.

El sistema de reconocimiento de objetos funciona en todo el cuadro de video y no esta limitado al eje horizontal, si embargo la posición vertical del parlante se mantuvo fija durante toda la grabación. Esto por dos razones, primero porque al tener un arreglo lineal solo se tiene resolución en un plano (ver eq. 2.4) por lo que aunque se tenga una diferencia en la posición vertical del parlante no afecta el calculo del *beamforming*. Segundo, el objetivo de la disposición experimental es ser representativo de un escenario HRI, donde es poco común que un hablante cambie su posición vertical. Finalmente, el sistema de control de la cabeza del robot es capaz de realizar movimientos horizontales y verticales, por que podría fijar nuevamente el objetivo en la linea horizontal del recuadro de video.

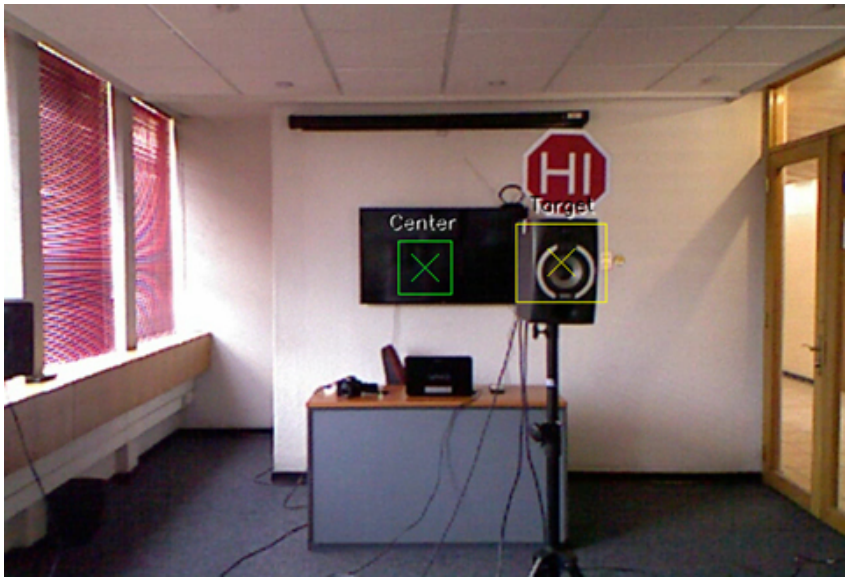


Figura 2.4: Cuadro de video capturado por la cámara Kinect montada sobre la cabeza del robot PR2. La X representa el centro del cuadro y el cuadrado amarillo representa la fuente de voz.

2.3.4. *Beamforming* y uso de información visual para mejorar su desempeño

En este capítulo, se estudia el efecto de usar *visual servoing* en conjunto con *beamforming* en una plataforma robótica como la descrita arriba. Dos esquemas de *beamforming* (i.e. delay-and-sum y MVDR) fueron integrados con detección de objetos y rastreo de objetos para mover la dirección del *beamforming* hacia el DOA mediante la actualización de los retardos (ver ecuaciones 2.5 y 2.4)[41]. Además, la cámara RGB que viene en la Kinect montada en la cabeza del robot PR2 fue también utilizada para implementar *visual servoing* (e.g. [15]). *Visual servoing* fue implementado de acuerdo al diagrama presentado en la figura 2.5.

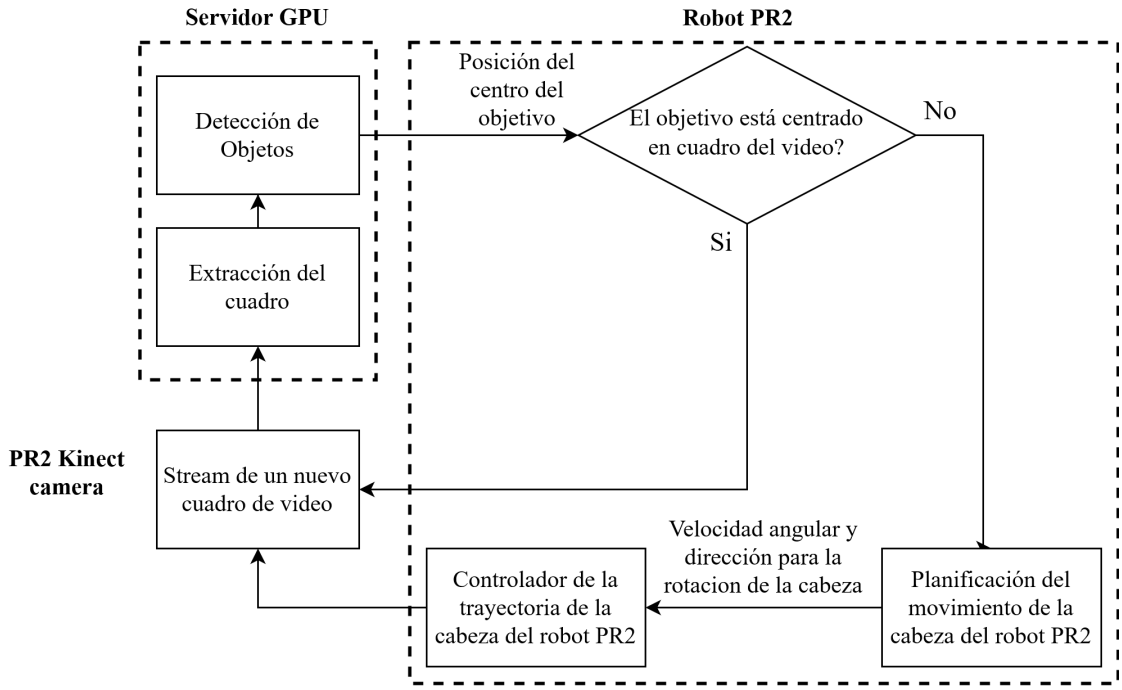


Figura 2.5: Diagrama de flujo detallado del esquema de *visual servoing* implementado en este estudio

Comenzando por el bloque "PR2 Kinect camera" de la figura, el robot envía video en tiempo real al servidor GPU para que aplique el procesamiento de imágenes. Cada cuadro se recibe y se procesa con YOLO para obtener la posición del objeto dentro del cuadro. Se aplican cálculos geométricos para determinar la posición angular del objeto con al centro de la imagen. Esta información es enviada al robot el cual calcula la diferencia en la posición angular actual de la cabeza y la posición objetivo, esto corresponde a la fase de *moving planning*, hay un bloque de decisión que determina si el objeto está centrado correctamente o no en cada cuadro. La señalética objetivo se considera centrada si se encuentra dentro de una región de tolerancia correspondiente a 1° alrededor del centro del cuadro de video. Dependiendo de esta condición se ejecuta o no un movimiento en la cabeza del robot. Después de eso, el sistema espera a que llegue un nuevo cuadro y el ciclo se repite nuevamente. Al hacer esto, la cabeza del robot esta continuamente orientándose hacia la fuente de voz, y si ya está centrada no se ejecuta movimiento alguno.

Como se mencionó anteriormente, la información es usada en un lazo cerrado para controlar la cabeza del robot PR2. Este control fue optimizado para realizar movimiento suaves y naturales, y que pudiese funcionar apropiadamente en tiempo real, con suficiente agilidad para cambiar la trayectoria en caso de que el objetivo cambie súbitamente su dirección y velocidad. El proceso mostrado en la figura 2.5, es realizado en un esquema cuadro-a-cuadro, y los cuadros son enviados por la Kinect a una tasa de muestreo (SR) que fue optimizada para evitar oscilaciones en el movimiento del robot. Así la tasa de muestreo determina la frecuencia con la que se actualiza la posición de la cabeza del robot y necesita ser lo suficientemente alta para mantener a la señalética objetivo centrada. Pero si la tasa de muestreo es muy alta, los cuadros no alcanzan a ser procesados por el servidor GPU y quedan guardados en un *buffer* lo que hace que el sistema no pueda responder a tiempo real. El tiempo para

procesar cada cuadro el servidor GPU es de 173 ms con un computador de escritorio con un procesador Intel i7 7700, 32GB RAM y una GPU GeForce 1080 de 8GB de VRAM, esto significa que los cuadros pueden ser procesados a una tasa de muestreo de 5.78 FPS. La Kinect puede transmitir video a una tasa de 30 FPS por defecto, y es posible controlar esa tasa mediante la omisión o *skip* de algunos cuadros intermedios. Si el parámetro de omisión o *skip* lo llamamos k, la tasa de muestreo por defecto de 30 FPS es reducida mediante la omisión de k cuadros después de enviar uno, obteniendo una tasa de muestreo equivalente a $\frac{30}{1+k}$ FPS. La tasa de muestreo óptima corresponde la tasa de muestreo máxima a la que el sistema puede operar el tiempo real, la cual fue obtenida haciendo $k = 4$ y la tasa de muestreo igual a 6.

El software del robot PR2 está escrito enteramente en ROS (Robot Operating System) [42]. El movimiento de la cabeza del robot es realizado usando el *head trajectory controller* el cual es uno de los controladores disponibles en ROS que corre en tiempo real en el PR2. El *head trajectory controller* recibe tres parámetros de entrada: p, la posición angular objetivo en radianes; ω, la velocidad angular en la posición objetivo; y t, el tiempo en segundo para ejecutar un movimiento. Los parámetros ω y t fueron elegidos empíricamente para obtener un movimiento suave de la cabeza del robot. La velocidad angular objetivo dada al *head trajectory controller* fue dinámicamente calculada para ser proporcional a la diferencia entre la posición actual de la cabeza del robot y la posición objetivo de la cabeza del robot. La velocidad angular ω fue definida por:

$$\omega = \alpha \cdot d \tag{2.7}$$

Donde α es la constante de proporcionalidad. Vale la pena mencionar que el parámetro ω corresponde a la velocidad angular a la cual el robot alcanza la posición p que se le entrega al controlador. El parámetro α fue elegido para minimizar el error absoluto promedio entre el centro del cuadro de video, lo cual corresponde al MRA, y la posición horizontal de la fuente de voz de interés, lo que corresponde al DOA. En otras palabras, α se estimó minimizando el AOI absoluto promedio o bien θ en la figura 2.1. El α óptimo fue determinado por una búsqueda discreta ente 0.05 y 0.12 con un paso igual a 0.01. Para cada valor de α, el AOI absoluto promedio fue medido mientras el robot PR2 realizaba desplazamientos laterales entre las posiciones P1 y P3, como se describe en la sección 2.3.1, por 80 segundos. La figura 2.6 muestra el AOI absoluto promedio resultante vs α. El mínimo AOI absoluto promedio resultante corresponde a 4.4° y se obtuvo con un α igual a 0.095.

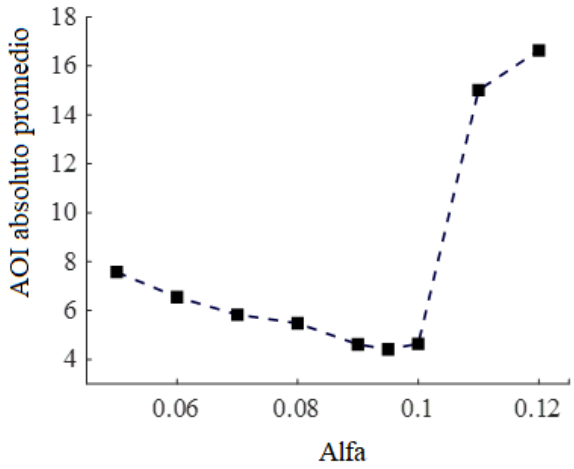


Figura 2.6: AOI absoluto promedio vs α como se define en la Ec. 1A.

Para resumir, una versión simplificada del algoritmo de *visual servoing* usado en este trabajo se presenta en la figura 2.7: comenzando por el bloque *PR2 Kinect camera*, el robot envía un cuadro de video en tiempo real al servidor GPU para aplicar procesamiento de imágenes con YOLO. El servidor GPU envía las coordenadas que describen hacia donde mover la cabeza del robot PR2, y con eso el movimiento del robot es planificado. Si el objetivo no está centrado en el cuadro de video, se realiza una acción en la cabeza del robot. Después de eso, el sistema espera la llegada de un nuevo frame y el ciclo se repite. Así, la cabeza del robot está siendo continuamente orientada hacia la fuente de voz. Esto mientras el robot realiza desplazamientos laterales entre las posiciones P1 y P3. Si no se utiliza *visual servoing*, el AOI varía entre -27° y 27° con un valor absoluto promedio de 15.7° (ver figura 2.8(a)). Cuando se aplica *visual servoing* el AOI varía entre -7° y 7° con un valor absoluto promedio de 4.5° (ver figura 2.8(b)). Consecuentemente, el esquema de visual servoing implementado aquí lleva a una reducción del AOI absolutos promedio de un 71 %.

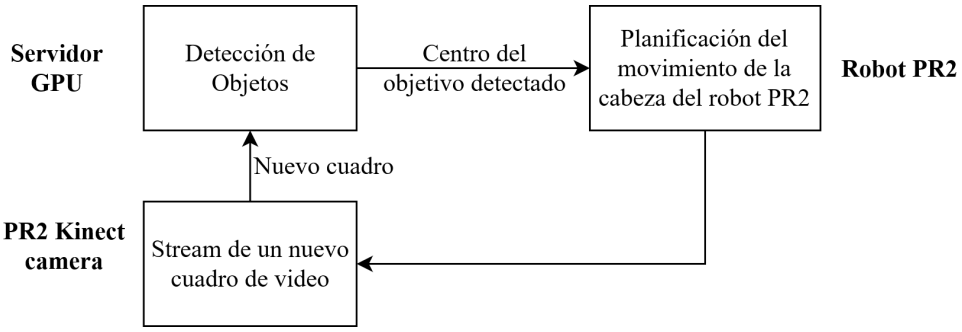


Figura 2.7: Diagrama de flujo simplificado del esquema de *visual servoing* empleado aquí

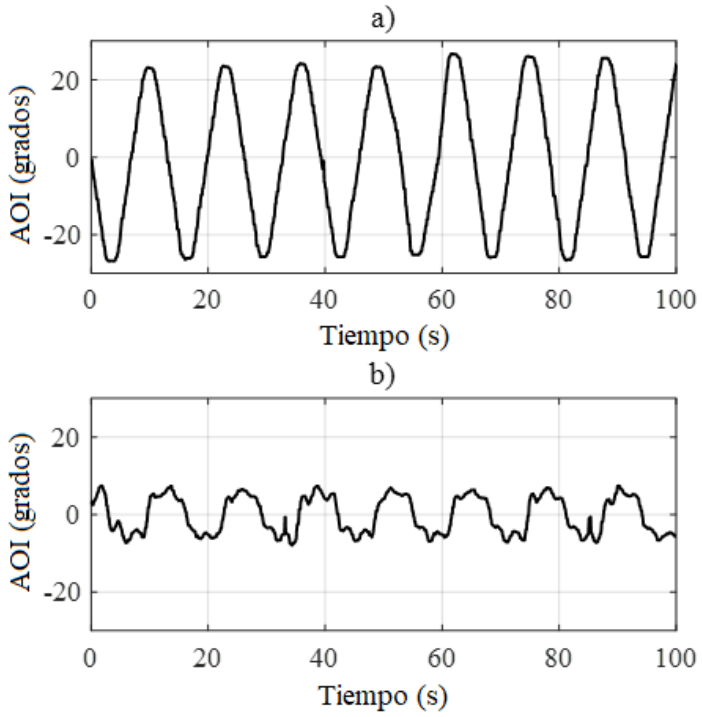


Figura 2.8: AOI vs tiempo cuando el robot PR2 se mueve desde P1 y P3 en la Fig. 2.3 sin (arriba) y con *visual servoing* (abajo).

Como se mencionó más arriba, la base de datos requerida para realizar este estudio es original. En esta sección, se presentan seis nuevas bases de datos con la Kinect montada encima de la cabeza del robot PR2. Para grabar el audio se usó la librería Kinect SDK disponible en sitio de Microsoft [43], se grabaron los cuatro canales independientes del arreglo de micrófonos de la Kinect. La base de datos fue grabada con el robot en una posición estática en P2 mirando hacia la fuente de voz o realizando desplazamientos laterales periódicos entre las posiciones P1 y P3 como se describe en la sección 2.3.1. El AOI de la fuente de voz fue obtenido mediante la cámara Kinect y sus valores fueron guardados. Dos condiciones fueron consideradas para el movimiento de la cabeza del robot, primero el robot permaneció con la cabeza estática a 0°, esto es, perpendicular a los movimientos de traslación horizontal que hace la base del robot; y la segunda condición, fue realizando *visual servoing* para seguir la fuente de voz de manera dinámica. Para cada condición de movimiento de la cabeza, se emplearon dos condiciones de fuentes de ruido externas, una usando una fuente de ruido y la otra usando dos fuentes de ruido. En la tabla 2.1 se resumen las condiciones de grabación del estudio reportado en este trabajo y en la figura 2.9 se aprecia una fotografía del escenario de grabación de las bases de datos.

Tabla 2.1: Condiciones de grabación de las bases de datos empleadas en este estudio.

Base de datos	Condiciones de movimiento del robot y su cabeza	Fuentes de ruido
Mov-1	Robot en movimiento de acuerdo a la sección 2.3.1. Cabeza fija a 0°	Fuente de ruido 1
Mov-2	Robot en movimiento de acuerdo a la sección 2.3.1. Cabeza fija a 0°	Fuentes de ruido 1 y 2
VS-Mov-1	Robot en movimiento de acuerdo a la sección 2.3.1. Cabeza siguiendo la fuente de voz	Fuente de ruido 1
VS-Mov-2	Robot en movimiento de acuerdo a la sección 2.3.1. Cabeza siguiendo la fuente de voz	Fuentes de ruido 1 y 2
ST-1	Robot estático, con la cabeza fija y mirando la fuente de voz en 0°	Fuente de ruido 1
ST-2	Robot estático, con la cabeza fija y mirando la fuente de voz en 0°	Fuentes de ruido 1 y 2

Figura 2.9: Escenario robótico móvil propuesto con el robot PR2, la fuente de voz y de ruido externo.

2.4. Estimación de la directividad de un arreglo de micrófonos lineal

Para tener una visión más precisa del efecto que tiene la dirección del *beamforming* con la directividad de este, se utilizó el dispositivo Kinect como un caso de estudio de micrófonos lineales. Este dispositivo a sido ampliamente usado por la comunidad HRI [44, 45] y también fue usado para la grabación de la base de datos del quinto *CHiME Challenge*[46]. La Kinect tiene un arreglo lineal de micrófonos compuesto de cuatro micrófonos cardioides apuntando hacia adelante del dispositivo[47], en la misma dirección y sentido que está la cámara que trae incorporada. Consecuentemente, este dispositivo tiene su MRA cercano a 90° con respecto al eje del arreglo de micrófonos. Se puede asumir que su MRA está en 90°, y los resultados que se muestran a continuación son consistentes con esta aproximación. Para estimar la directividad del arreglo de micrófonos de la Kinect, se grabaron tonos puros de 500 Hz, 1 KHz, 2 KHz y 4 KHz por cada uno de los cuatro canales del arreglo. Estas señales fueron grabadas con un AOI variable en un intervalo de 0° a 360° con un paso de 10°, mediante la rotación de la Kinect y manteniendo la fuente de sonido estática. La fuente de sonido corres-

ponde a un parlante de estudio (TANNOY 501a) ubicado a un metro del centro del arreglo de micrófonos. Las grabaciones fueron llevadas a cabo en una cámara anecoica y la presión de sonido fue medida con un sonómetro. De acuerdo a [48], la sensibilidad de un micrófono, M_p, puede ser definida como el voltaje que puede producir una determinada presión de sonido.

$$M_p = \frac{V}{P} \tag{2.8}$$

Donde V y P son el voltaje a la salida del micrófono y la presión de la onda de sonido respectivamente. Así, la sensibilidad de un *beamforming* puede ser definida como:

$$M_B = \frac{B(V_1, \dots V_n, \theta, \phi)}{P_B} \tag{2.9}$$

donde $V_1, \dots V_n$ son los voltajes de cada micrófono; θ y ϕ son el AOI y el ángulo donde apunta el *beamforming*, ambos con respecto al MRA ; y $B(.)$ es la salida del *beamforming* y P_B es la presión de sonido al centro del arreglo de micrófonos. Para obtener una mejor visión de la dependencia de la dirección del *beamforming* con la directividad ϕ, $B(V_1, \dots V_n, \theta, \phi)$ fue estimado usando el esquema *delay-and-sum*. Dada una dirección del *beamforming* ϕ, la ganancia directiva del *beamforming* se puede definir como la tasa entre la potencia dado un θ y la potencia de todos los demás θ [49]. La directividad es máxima cuando θ es igual a ϕ. Consecuentemente, la directividad del *beamforming* D(ϕ) se expresa como una función de la dirección del *beamforming*.

$$D(\phi) = \frac{|M_B(V_1, \dots V_n, \phi, \phi)|^2}{\frac{1}{N}\sum_{i=0}^{N-1} |M_B(V_1, \dots V_n, \theta_i, \phi)|^2} \tag{2.10}$$

Donde $\theta_i = 10° \cdot i$ y $N = 36$. En la plataforma robótica móvil implementado en este estudio, donde dos o una fuente de ruido fueron usadas. Como resultado, una directividad alternativa que está mejor relacionada con el SNR, $D_{snr}(\phi)$, se puede definir para dos o una fuente de ruido respectivamente.

$$D_{snr}(\phi) = \frac{\mid M_B(V_1, \dots V_n, \phi, \phi) \mid^2}{\mid M_B(V_1, \dots V_n, \theta_1, \phi) \mid^2} \tag{2.11}$$

$$D_{snr}(\phi) = \frac{\mid M_B(V_1, \dots V_n, \phi, \phi \mid^2}{0.5\cdot \mid M_B(V_1, \dots V_n, \theta_1, \phi) \mid^2 + 0.5\cdot \mid M_B(V_1, \dots V_n, \theta_2, \phi) \mid^2} \tag{2.12}$$

Donde θ_1 y θ_2 denota el AOI para la fuente 1 y la fuente 2 respectivamente. $D(\phi)$ y $D_{snr}(\phi)$, como se define en las ecuaciones de la 2.10 a la 2.12 respectivamente, fueron computadas con las sensibilidades medidas con los tonos de 500 Hz, 1kHz, 2kHz y 4kHz como se mencionó más arriba. Así, las directividades obtenidas con estos tonos fueron promediadas. La Figura 2.10(a) muestra el $D(\phi)$ promedio. las figuras 2.10(b) y 2.10(c) muestran el $D_{snr}(\phi)$ promediado con una y dos fuentes de ruido respectivamente. Como se puede ver en las figuras 2.10(a), 2.10(b) y 2.10(c) la directividad del *beamforming* es máxima cuando la dirección del *beamforming* ϕ es igual a 0° y tiende a degradarse cuando el valor absoluto de ϕ se acerca a 30°.

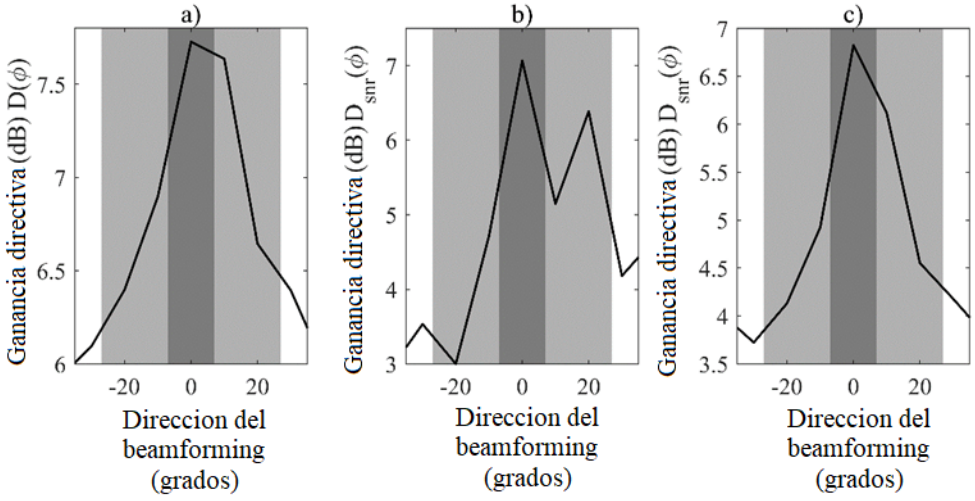


Figura 2.10: Ganancia directiva promedio vs dirección del *beamforming*, curvas estimadas con los arreglos de micrófonos lineales Kinect: izquierda, $D(\theta)$ como se define en la Ec. 6; y, centro y derecha, $D_snr(\theta)$ como se define en Eq. 7 y 8 con una y dos fuentes de ruido, respectivamente.

En el segundo peak derecho de la Figura 2.10(b) con una sola fuente de ruido se aprecia una excepción. En este caso la posición de la fuente de ruido se considera a la izquierda del MRA y esta asimetría en la Figura 2.10(b) causa que el denominador en la ecuación 2.11 disminuya más rápidamente que el numerador a medida que avanza ϕ. Es importante mencionar que las curvas en la Figura 2.10 no representan el lóbulo de un *beamforming*, y la que información entregada por las curvas de directividad del *beamforming* vs ϕ, no corresponden al ancho del *beamforming*. Finalmente se aprecia que la máxima directividad alrededor de $\phi = 0$ es una fuerte motivación para mantener la fuente de sonido de interés cerca de ese valor. Esto significa mantener la cabeza del robot apuntando hacia la fuente de interés mientras el robot de mueve en un ambiente dinámico. Como se muestra en la sección 2.3.4, *visual servoing* produce una reducción del rango en el que se mueve ϕ, este mismo rango se aprecia en la región gris claro y la región gris oscuro de la Figura 2.10.

2.5. Reconocimiento automático de la voz - ASR

Los experimentos de reconocimiento de voz fueron realizados con una DNN-HMM usando el Toolkit de reconocimiento de voz Kaldi [50]. Se entrenó una GMM-HMM de acuerdo a

la receta tri2b Kaldi Aurora-4 con los datos de entrenamiento descritos más adelante usando *features* MFCC, LDA (linear discriminant analysis) y MLLT (maximum likelihood linear transforms). Primero se entrenó un sistema mono-fonema, luego los alineamientos de ese sistema fueron utilizados para generar un sistema tri-fonema inicial. Finalmente, los alineamientos tri-fonema fueron utilizados para entrenar un sistema tri-fonema. La GMM en el sistema GMM-HMM entrenado fue reemplazado con una DNN compuesta de siete capas ocultas y 2048 unidades por capa, con una entrada con contexto de 11 frames. El número de unidades de la salida de la DNN es igual al número de Gaussianas en el sistema GMM-HMM correspondiente. La DNN fue entrenada utilizando características MelFB (Mel filter bank). La referencia para el entrenamiento del sistema DNN se obtuvo con la versión limpia -*clean*- de la base de datos completa y la GMM-HMM fue entrenada con la misma base de datos *clean*. Esto lleva a una mejor referencia para la DNN que usar la base de datos ruidosa directamente, [51, 52]. Primeramente, la DNN fue entrenada usando Cross-Entropy. Después el sistema DNN final fue reentrenado usando sMBR *discriminative training* [53]. Para decodificación, se usó el modelo de lenguaje de trigramas estándar 5K lexicón de la base de datos DARPA Wall Street Journal(WSJ).

El conjunto de datos de entrenamiento fue generado siguiendo el mismo procedimiento descrito en [20] para modelar el ambiente acústico y el canal acústico variable en el tiempo. En [20] se presentan resultados de ASR comparables con API's públicas, y estos resultados fueron obtenidos con limitados datos de entrenamientos. Primero, se estimaron 33 respuestas al impulso (IRs) haciendo uso de método de Farina's [54] con la Kinect orientada hacia 11 diferentes ángulos entre 150° y -150° con el arreglo de micrófonos ubicado a uno, dos, y tres metros de la fuente de voz. Se aplico convolución a un 25 % del conjunto de entrenamiento *clean* de la base de datos de Aurora-4 con la IR estimadas a un metro de la fuente de voz y apuntado hacia la fuente de voz. El restante 75 % se convolucionó con la restantes 32 IRs. Finalmente, se añadió ruido externo de restaurante y ruido de robot utilizando la herramienta FaNT [55] al restante 75 % de los datos con SNR variable entre 10 and 20 dB.

2.6. Resultados y discusión

En este trabajo se evaluaron tres esquemas de *beamforming*. Primero, *delay-and-sum* con el AOI estimado por rastreo de objetos en cual llamaremos D&S-AOI. Segundo, *weighted-delay-and-sum* tal como está implementado originalmente en el toolkit de BeamformIt, el cual llamaremos W-D&S. Y tercero, MVDR usando el toolkit BTK2.0 sin y con el AOI estimado por rastro de objetos, los cuales llamaremos MVDR-B y MVDR-AOI respectivamente. MVDR-B es también conocido como MMSE beamformer (minimum mean square error beamformer)[56, 49] y está implementado por defecto en el toolkit BTK2.0. También se implementó W-D&S con el uso de AOI estimado por rastreo de objetos, el cual para el mejor caso dio resultados similares a los obtenidos con D&S-AOI lo cual no aporta a discusión. Así solo se presentan los resultados obtenidos con W-D&S con su configuración por defecto. El AOI fue actualizado a una frecuencia de 11Hz, y fue usado para actualizar los retardos τ_1 de acuerdo a la 2.4 en el caso de D&S y actualizar los pesos w^H de acuerdo a la 2.6 en el caso de MVDR. Los resultados se presentan en las figuras 2.11, 2.12 y 2.13. El WER con la base de datos *clean* es igual a 3.21 %, el cual es competitivo con los publicados en la literatura para las condiciones de entrenamiento multi-condición con la base de datos Aurora-4 [19, 57]. La

figura 2.11 muestra los resultados en WER cuando el robot mantiene la cabeza estática y se mantiene fijo en la posición P2 mirando a la fuente de voz, utilizando una y dos fuentes de ruido. MVDR-B y MVDR-AOI entregan los WER más bajos. MVDR es muy efectivo para cancelar el ruido en condiciones estacionarias, MVDR-B y MVDR-AOI reducen en similar magnitud el WER cuando se comparan con D&S y W-D&S. El segundo WER más bajo lo entrega D&S-AOI. W-D&S entrega el WER más alto. Esto probablemente es porque W-D&S incorpora un error inherente a la estimación del DOA basándose solo en la señal acústica. En el escenario empleado existen fuentes de ruido interferentes y reverberación, lo cual dificulta muchas tareas relacionadas con procesamiento de voz, especialmente la estimación del DOA [58], dado que las reflexiones acústicas son estadísticamente similares a la señal que llega directo de la fuente y estas reflexiones vienen de distintas direcciones.

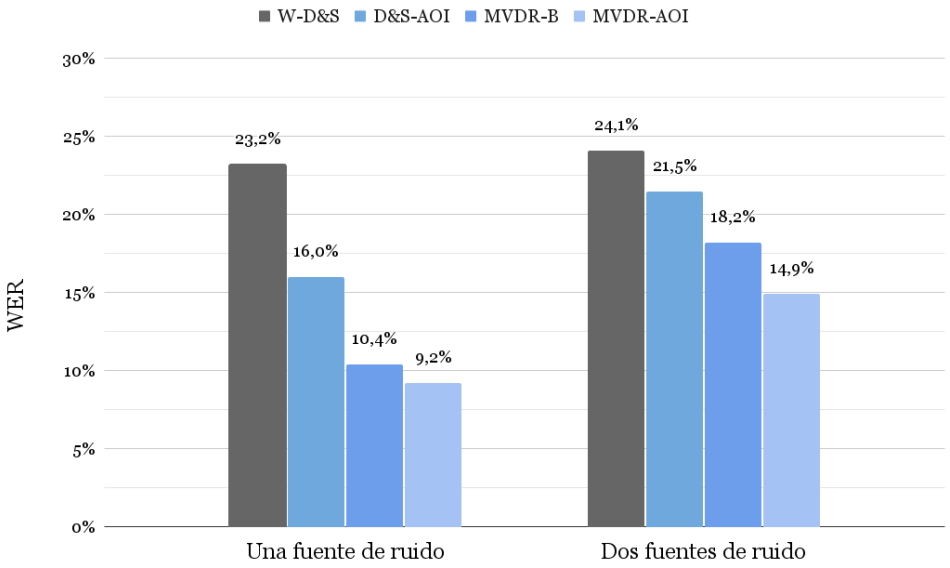


Figura 2.11: WERs obtenidos con las bases de datos ST-1 y ST-2, es decir, el robot y su cabeza están estáticos en la posición P2 mirando a la fuente de voz.

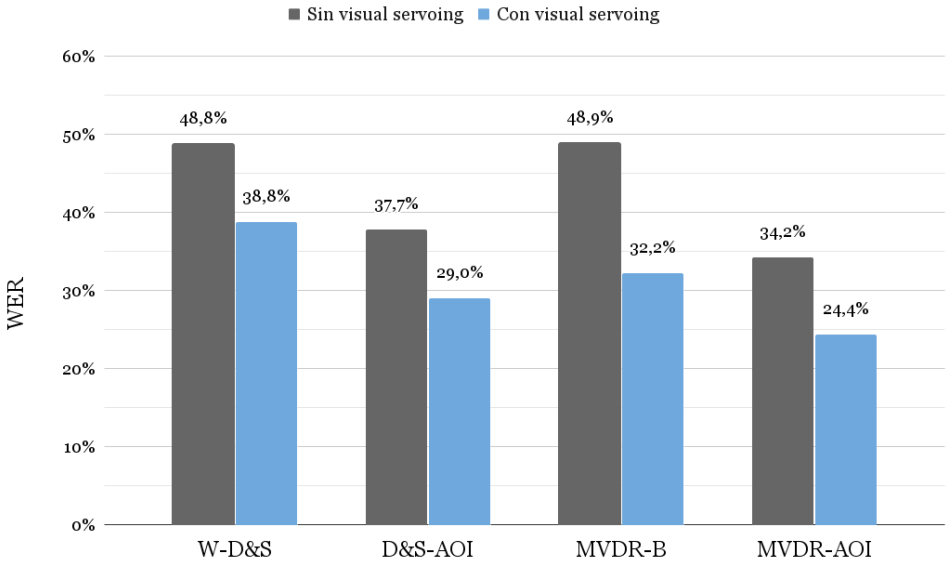


Figura 2.12: WERs obtenidos con las bases de datos Mov-1 y VS-Mov-1 y, es decir, el robot se mueve desde P1 y P3 sin y con *visual servoing*, respectivamente. Solo se utilizó una fuente de ruido interferente.

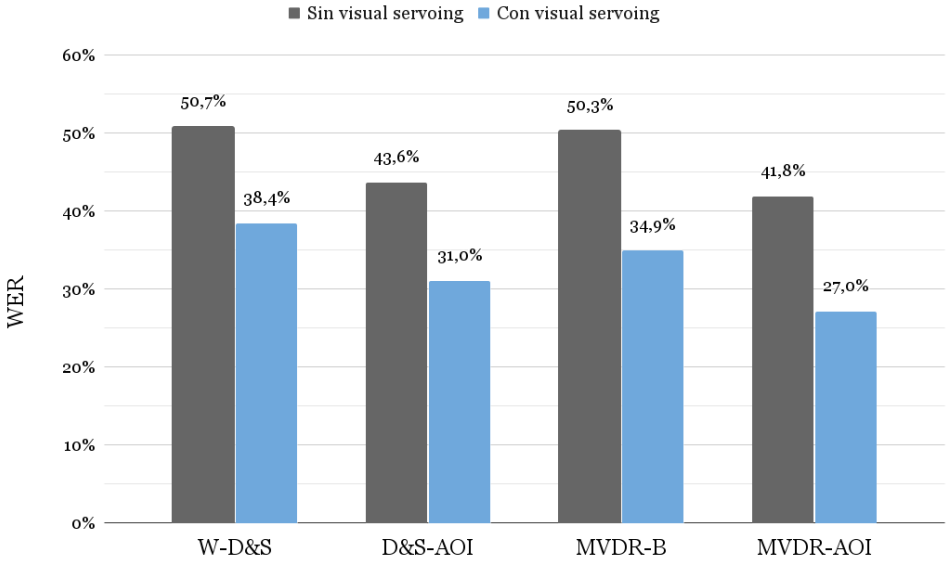


Figura 2.13: WERs obtenidos con las bases de datos Mov-2 y VS-Mov-2, es decir, el robot se mueve desde P1 y P3 sin y con *visual servoing*, respectivamente. Se emplearon dos fuentes de ruido interferentes.

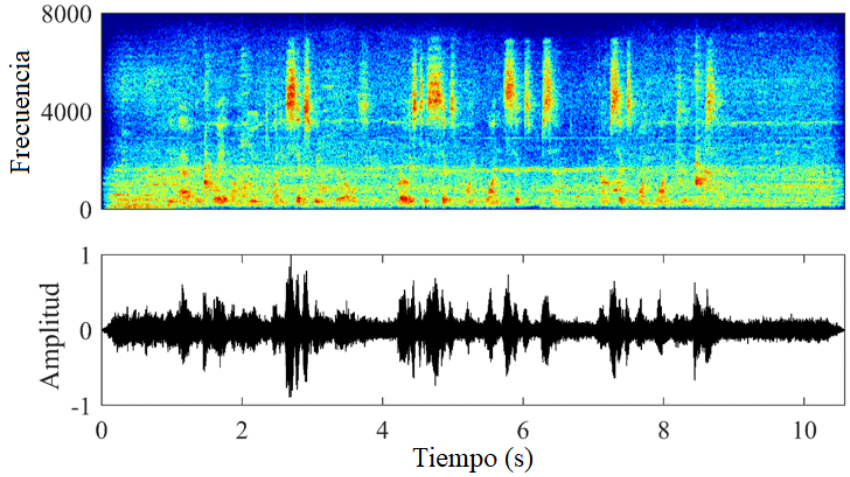
Los resultados obtenidos en condiciones dinámicas están resumidos en las figuras 2.12 y 2.13. Cuando el robot se mueve entre las posiciones P1 y P2 en la figura 2.3, el WER aumenta drásticamente si se compara al escenario estático en la figura 2.11, donde la cabeza del robot esta fija. Con una fuente de ruido la mejora relativa en WER fue igual a 110.3 %, 135.6 %, 393.9 % y 273.9 % con W-D&S, D&S-AOI, MVDR-B y MVDR-AOI, respectivamente. Cuan-

do se usan dos fuentes de ruido la mejora relativa en WER es igual a 110.4 %, 102.8 %, 242.2 % y 178.2 % con W-D&S, D&S-AOI, MVDR-B y MVDR-AOI, respectivamente. La variación relativa en WER fue estimada como $100 \times (finalWer - initialWer)/(initialWer)$. La degradación más grande del desempeño se observa entre MVDR-B y MVDR-AOI. Esto presumiblemente se debe al hecho que MVDR requiere que una estimación de la matriz de correlación espacial $\sum_N(\omega)$ en la Eq. 3, la cual asume estacionariedad la cual no se conserva en un escenario dinámico y móvil como el estudiado aquí. Se observa que la efectividad de MVDR se reduce por la falta de estacionariedad del ruido espacialmente correlacionado, lo cual afecta la estimación de $\sum_N(\omega)$. También, el incremento del WER con MVDR-B es aún mayor ya que se introduce un error en la estimación del AOI para este caso. Por defecto, la matriz de correlación fue estimada para toda la *utterance* haciendo uso de todos los intervalos de no-voz (ruido). Se intento obtener una estimación $\sum_N(\omega)$ dependiente del tiempo, obteniendo una nueva matriz de autocorrelación por cada intervalo de no-voz presente en la señal. Esta estrategia no llevo a buenos resultados ya que los intervalos de no-voz eran muy cortos y no entregaban suficientes muestras para estimar correctamente $\sum_N(\omega)$. Otra estrategia utilizada fue hacer una estimación inicial de la matriz de correlación con el primer intervalo de no-voz y luego actualizar $\sum_N(\omega)$ en los siguientes intervalos de no-voz. Esta aproximación tampoco dio mejores resultados comparados con la estimación original de la matriz de correlación. Esto sugiere que MVDR tiene limitaciones inherentes para ser aplicado en escenarios móviles de interacción humano-robot como el estudiado aquí, aun mas, particularmente tiene problemas cuando los intervalos de no-voz son escasos. Los WER más bajos obtenido corresponden a D&S-AOI y MVDR-AOI que hacen uso de estimación del AOI mediante procesamiento de imágenes.

Cuando se usa *visual servoing*, el WER se reduce significativamente cuando el robot se mueve entre las posiciones P1 y P3 como en la Fig. 2.3. Se puede observar que la distancia angular promedio entre el centro del *frame* y la fuente de voz objetivo (ver Figura 2.1) se reduce como se muestra en las figuras 2.8 y 2.10. De acuerdo a las figuras 2.12 y 2.13, el uso de *visual servoing* puede conducir a reducciones en WER de hasta 20.5 %, 23.1 %, 34.2 % y 29.1 % con WD & S, D & S-AOI, MVDR-B y MVDR-AOI, respectivamente, con una fuente de ruido. Con dos fuentes de ruido, las reducciones fueron similares, reducción de 24,3 %, 28,9 %, 30,6 % y 34,7 % con W-D & S, D & S-AOI, MVDR-B y MVDR-AOI, respectivamente. Este resultado corrobora claramente la suposición con respecto a la dependencia de la directividad con el ángulo de la dirección del *beamforming* como se muestra en la Fig. 2.10: cuanto menor es el ángulo de la dirección del *beamforming*, mayor es la directividad. Las mejoras en WER más altas se observaron con MVDR-AOI y MVDR-B, lo que presumiblemente se debe a un efecto de atenuación del ruido residual en estos escenarios dinámicos. Además, el WER más bajo se logró con MVDR-AOI que hace uso de la estimación del AOI mediante procesamiento de imágenes. D& S-AOI arrojo un mejor resultado que W-D&S principalmente debido al uso del procesamiento de imágenes para estimar el AOI. Además, el sistema de rastreo de objetos por imágenes proporciona una actualización de AOI a una velocidad de 11 Hz, que es casi tres veces más rápida que la proporcionada por BeamformIt con la ventana de análisis por defecto de 8000 muestras con una superposición del 50 %. Esta velocidad de actualización más alta del AOI es más adecuada para un escenario HRI dinámico como el que se aborda aquí. Finalmente, como se mencionó anteriormente, el *beamforming* con Kinect-SDK no se incluyó en el análisis porque sus técnicas o algoritmos no están disponibles públicamente. Sin embargo, aunque no se muestra en este documento, conduce a resultados comparables a W-D&S.

Como validación final, el efecto de *visual servoing* se evaluó en el dominio de tiempo y frecuencia. Las figuras 2.14-2.16 muestran las señales en el tiempo, espectrogramas y la transcripción ASR correspondiente para W-D&S, D&S-AOI y MVDR-AOI, respectivamente, sin y con *visual servoing*. Para una mejor comparación, las señales en el tiempo se normalizaron con respecto al valor absoluto máximo de la amplitud. Como puede verse en las Figs. 2.14-2.16, hubo una reducción notable en la potencia del ruido cuando se usó *visual servoing* con todos los esquemas *beamforming* probados aquí. Los espectrogramas se normalizaron con respecto al componente de potencia máxima. La representación de los espectrogramas se realizó con la escala de colores Matlab Jet donde el rojo oscuro denota 0db y el azul oscuro corresponde a -60db.

I HAVE NO WAY AROUND A TWENTY SIX WEEK BILLS ROSE TO SIX POINT ONE SIX PERCENT FROM SIX POINT ONE THREE PERCENT **A MONTH**



OF THE AVERAGE RATE ON NEW TWENTY SIX WEEK BILLS ROSE TO SIX POINT ONE SIX PERCENT FROM SIX POINT ONE TWO **<UNK>**

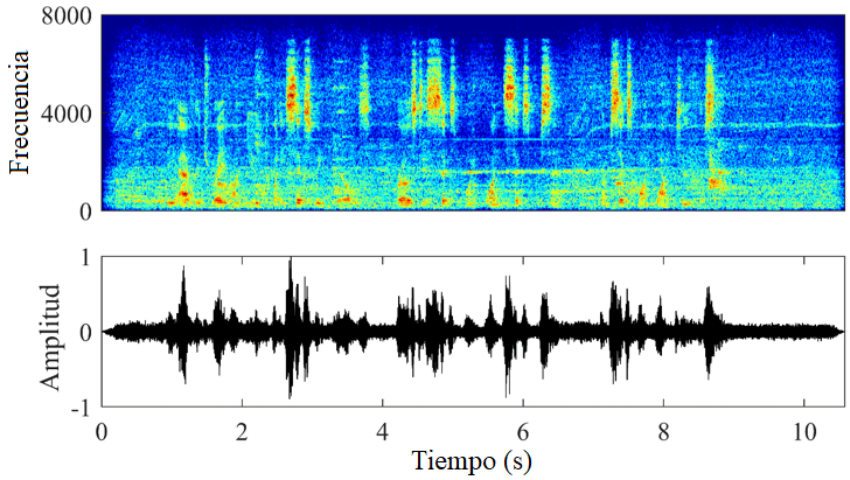
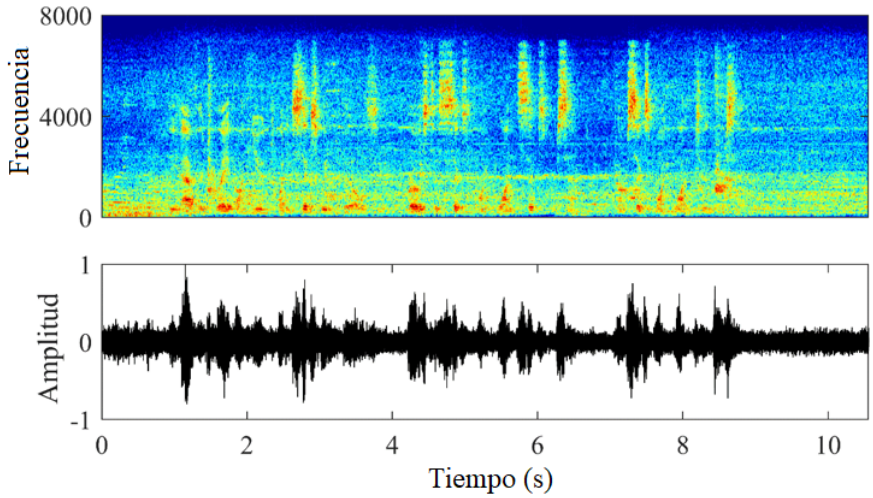


Figura 2.14: Transcripciones, espectrogramas y formas de onda obtenidas con W-D&S: arriba, sin *visual servoing*; e inferior, con *visual servoing*. La transcripción de referencia corresponde a: "THE AVERAGE RATE ON NEW TWENTY SIX WEEK BILLS ROSE TO SIX POINT ONE SIX PERCENT FROM SIX POINT ONE TWO PERCENT". Los errores de ASR con y sin *visual servoing* se resaltan en negrita en las transcripciones correspondientes.

<UNK> **OF** THE AVERAGE RATE ON NEW TWENTY SIX WEEK BILLS ROSE TO SIX POINT ONE SIX PERCENT **GROWTH** FROM SIX POINT ONE TWO PERCENT



OF THE AVERAGE RATE ON NEW TWENTY SIX WEEK BILLS ROSE TO SIX POINT ONE SIX PERCENT FROM SIX POINT ONE TWO <UNK>

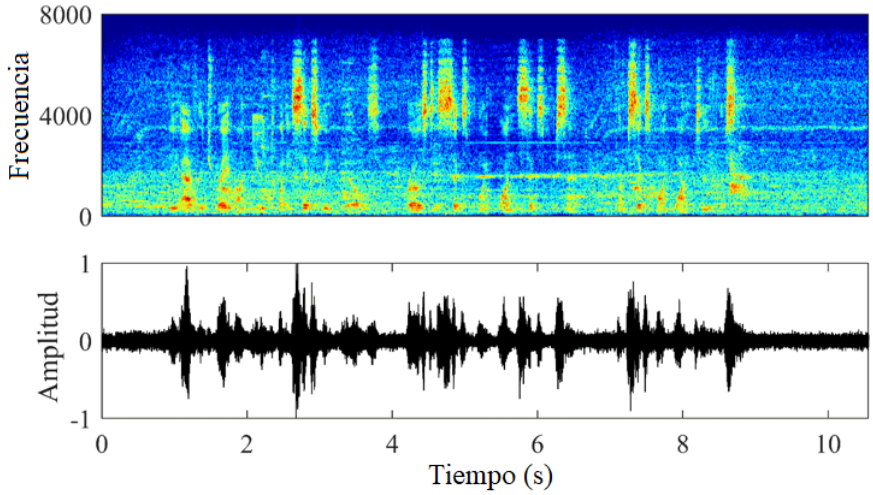


Figura 2.15: Transcripciones, espectrogramas y formas de onda obtenidas con D&S-AOI: arriba, sin *visual servoing*; e inferior, con *visual servoing*. La transcripción de referencia corresponde a: "THE AVERAGE RATE ON NEW TWENTY SIX WEEK BILLS ROSE TO SIX POINT ONE SIX PERCENT FROM SIX POINT ONE TWO PERCENT". Los errores de ASR con y sin *visual servoing* se resaltan en negrita en las transcripciones correspondientes.

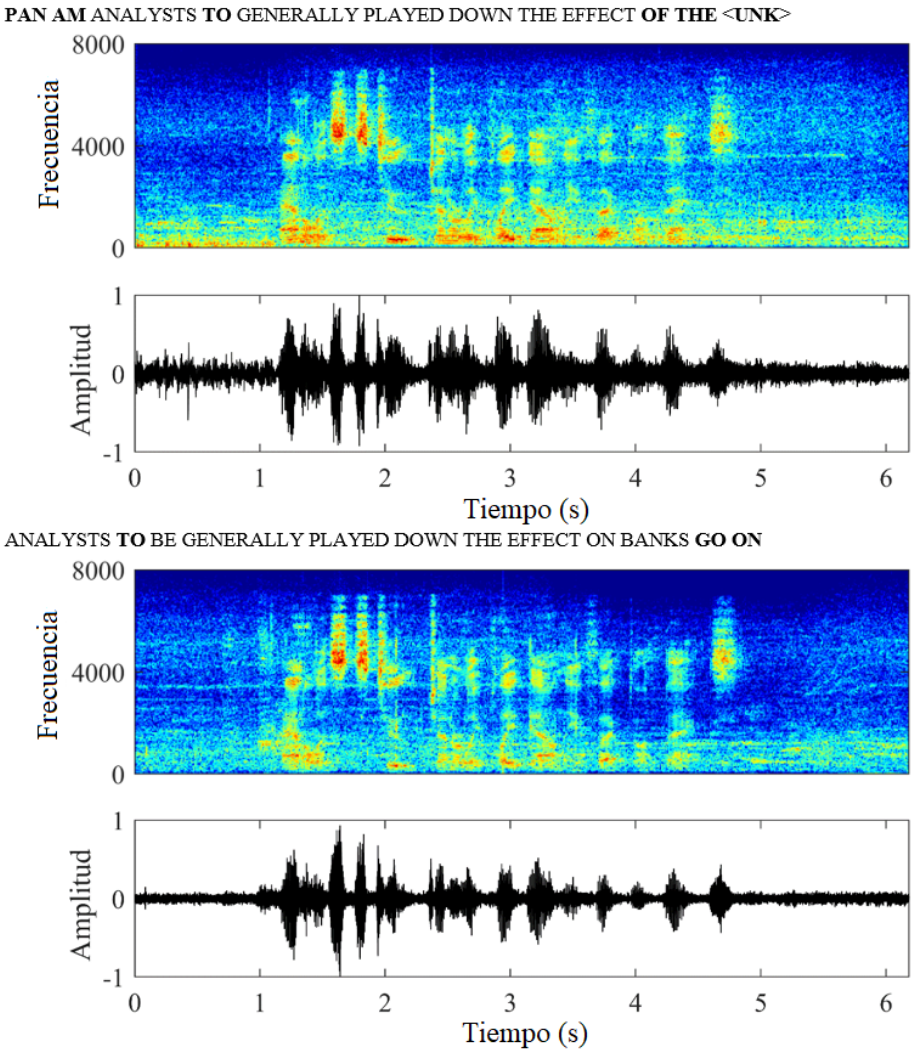


Figura 2.16: Transcripciones, espectrogramas y formas de onda obtenidas con MVDR-AOI: arriba, sin *visual servoing*; e inferior, con *visual servoing*. La transcripción de referencia corresponde a: .ᴬNALYSTS TOO GENERALLY PLAYED DOWN THE EFFECT ON BANKS". Los errores de ASR con y sin *visual servoing* se resaltan en negrita en las transcripciones correspondientes.

Como se mencionó anteriormente, el caso de estudio en este trabajo es la Kinect con su arreglo de micrófonos lineal. Por otro lado, los arreglos de micrófonos circulares también son comunes en HRI y merecen un poco de discusión con respecto a la aplicabilidad de *visual servoing* con este tipo de arreglo. La figura 2.17 representa un arreglo de micrófonos circular de radio r que se encuentra en el plano xy con su centro en el origen de un sistema de coordenadas polares y una onda plana que incide desde $\omega_s = (\Theta_s, \phi_s)$. La mayoría de los arreglos circulares de micrófonos se diseñan y construyen teniendo en cuenta la incidencia de onda normal con respecto al eje z, es decir, $\vartheta_s = \pi/2$ [59, 60, 61, 62, 63, 64, 65]. En principio, un *beamforming* se puede dirigir en cualquier dirección del plano xy sin cambiar significativamente su directividad [66]. Sin embargo, un arreglo circular con micrófonos omnidireccionales equiespaciados debería mostrar una periodicidad en la directividad con respecto a ϕ_s mientras cambia la dirección del *beamforming* con máximos en $\phi_s = 2\pi l/L$, donde $l = 0, 1, ..., L-1$

corresponde al índice del $l-esimo$ micrófono y L es el número de micrófonos. Si la distancia entre microfonos sucesivos es grande este efecto de periodicidad puede ser un problema que se puede mejorar usando *visual servoing*, girando el arreglo para alinear el maximo con el DOA. Además, en un escenario más general, el movimiento del usuario en relación con el robot no está restringido al plano xy definido por el arreglo circular y el uso de *visual servoing* puede mantener ϑ_s cerca de $\pi/2$. En este contexto, *visual servoing* es una estrategia mucho más factible que modificar la altura del arreglo de micrófonos. Además, existen aplicaciones en las que se puede emplear un arreglo plano no necesariamente circular, orientado hacia la fuente acústica, es decir, con un ángulo de elevación pequeño o ϑ_s igual a cero. Este es el caso, por ejemplo, cuando se colocan cámaras acústicas frente a un hablante humano para estimar las coordenadas espaciales de las fuentes de sonido para detectar de forma no invasiva tanto la nasalidad como la lateralidad de la voz [67, 68]. Este tipo de aplicaciones requiere que la fuente acústica de interés esté ubicada en ϑ_s, es decir, a lo largo o cerca del eje z. Esto se debe al hecho de que: "la resolución de los arreglos planos disminuye drásticamente para ángulos fuera del plano xy"[69]. También se puede usar *visual servoing* en estas aplicaciones para mantener la fuente acústica centrada.

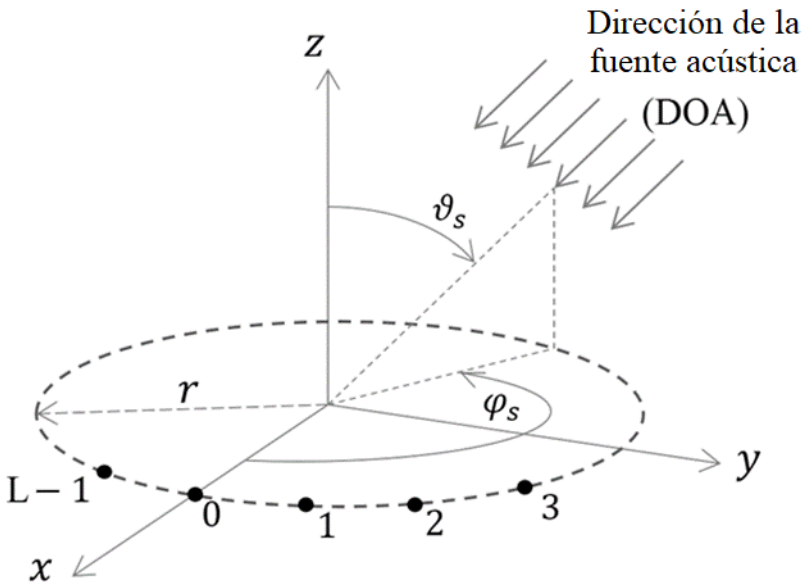


Figura 2.17: Diagrama de un arreglo circular de micrófonos de radio r que se encuentra en el plano xy con su centro en el origen de un sistema de coordenadas polares y una onda plana que incide desde $\omega_s = (\vartheta_s, \phi_s)$.

Cabe destacar que la tarea que aquí se abordó con un robot en movimiento y fuentes de ruido es una tarea desafiante. Como ejemplo, en [20] APIs públicas de ASR mostraron una degradación significativa en experimentos con una base de datos similar a la empleada aquí, pero sin fuentes de ruido adicionales al robot. En el caso aquí considerado, las fuentes de ruido aditivo representan una dificultad mayor. Desde la perspectiva del robot en movimiento, el ruido aditivo no es estacionario. En este trabajo, el AOI fue estimado mediante el uso de procesamiento de imágenes. En consecuencia, la precisión del método de rastreo de la fuente empleado aquí difícilmente podría mejorarse ya que las imágenes no se ven afectada por el entorno acústico y el canal acústico variable en el tiempo. Otra observación importan-

te es que MVDR-AOI logró los mejores resultados en posición estática. En consecuencia, la adaptación de la matriz de correlación de ruido en intervalos de ruidosos debería conducir a mejoras de precisión con MVDR-AOI. Además, también podría evaluarse la incorporación de más esquemas de cancelación de ruido aditivos después del *beamforming*. Finalmente, vale la pena mencionar que la precisión de ASR en aplicaciones de HRI no debería abordarse solo con la información acústica. De hecho, la información obtenida por el robot del usuario podría emplearse para reducir la perplejidad del modelo de lenguaje [20].

2.7. Conclusiones del capítulo

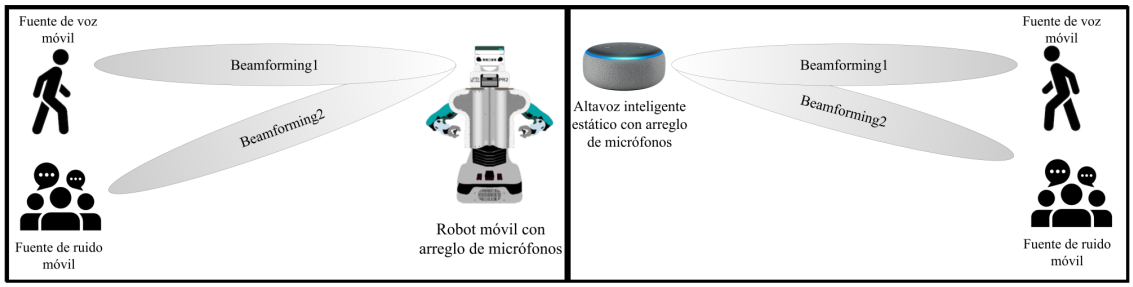
Se midió la directividad de un arreglo lineal de microgonos y se evaluaron los métodos estándar de *beamforming* en combinación con *visual servoing*: *delay-and-sum* combinado con rastreo de objetos; *weighted delay-and-sum* como se implementó en BeamformIt; y MVDR también combinado con rastreo de objetos. Los resultados presentados aquí sugieren que el rendimiento de los métodos de *beamforming* se degrada drásticamente en condiciones de movimiento y no estacionarias. Sin embargo, *visual servoing* en HRI puede mejorar significativamente el rendimiento de una matriz de micrófonos lineales con respecto a la precisión de ASR. La mejora media observada en la precisión de ASR cuando la cabeza del robot se dirigió hacia la fuente de voz puede llegar al 28,2 %. Cabe destacar que la metodología adoptada aquí es aplicable a cualquier arreglo de micrófonos, lineal o no, y también se presenta una discusión sobre arreglos circulares. El efecto positivo del *visual servoing* es generalizable a cualquier arreglo de micrófonos lineales, pero la mejora potencial en la precisión de ASR depende del caso. Tenga en cuenta que los conjuntos de datos necesarios para llevar a cabo el estudio aquí presentado no existían, y se tuvo que construir un escenario experimental robótico móvil con fuentes de voz y ruido para realizar el estudio.

Capítulo 3

Speech enhancement con aprendizaje profundo

3.1. Introducción

Las capacidades de los sistemas robóticos y autónomos para procesar y comprender el lenguaje hablado han ido creciendo en los últimos años con el uso de técnicas estadísticas y de aprendizaje profundo. Estas técnicas pueden completar numerosas tareas, como reconocimiento de voz, filtrado espacial y *speech enhancement*. La cancelación de ruido y reverberación -*speech enhancement*- es muy relevante para la interacción humano-robot sobre todo en escenarios acústicos complejos (ver Fig. 3.1(a)) y también para asistentes inteligentes, donde los comandos hablados a estos dispositivos pueden ser de corta duración, como " Alexa " o " Hey Siri " [70], y puede tener lugar en presencia de fuentes de ruido (ver Fig. 3.1(b)). Además, la complejidad del procesamiento de voz está parcialmente definida por el escenario donde ocurren [71]. Por ejemplo, los altavoces inteligentes son capaces de mejorar y reconocer la voz en un escenario estático (ver Fig. 3.1(b)), donde el dispositivo se encuentra en una sala de estar y las fuentes de voz suelen ser estáticas, pero también podrían estar en movimiento. Un escenario más desafiante se muestra en la Fig. 3.1(a) donde hay un robot en movimiento, así como fuentes de sonido en movimiento. Este escenario puede considerarse una generalización del altavoz inteligente y requiere técnicas más robustas que puedan capturar la dinámica del entorno acústico.



a) **b)**

Figura 3.1: (a) Fuentes de voz y de ruido móviles escenario HRI y (b) aplicación de altavoz inteligente.

3.1.1. El problema de separación de fuentes clásico

Para un arreglo de micrófonos, las señales capturadas pueden ser representadas como $Y = [y_0(t), y_1(t), \ldots, y_m(t), \ldots, y_{M-1}(t)]$, donde $0 \leq m < M$, M es el número total de micrófonos e Y es la matriz de señales capturadas. Considerando que hay K fuentes $s_k(t)$, donde $0 \leq k < K$. Cada micrófono m recibe la suma de todas las K fuentes, lo cual nos lleva a:

$$y_m(t) = s_{0,m}(t) + s_{1,m}(t) + \ldots + s_{K-1,m}(t) = \sum_{k=0}^{K-1} s_{k,m}(t)\,(1) \tag{3.1}$$

donde $s_{k,m}(t)$ es la señal recibida $s_k(t)$ en el micrófono m. Esta representación puede ser comprimida como una función $y_m(t) = f[s_{k,m}(t)]$ que depende de la posición de cada fuente k. Si asumimos que la función es lineal se puede obtener lo que se llama un modelo de mezclas: $Y = WS$. Como se definió arriba, Y es la matriz de señales capturadas de tamaño $M \times 1$ por cada muestra de tiempo t, donde $0 \leq t < T$ y T es el largo en muestras del intervalo de la señal que esta siendo analizada. Consecuentemente, Y puede ser interpretado como una matriz de tamaño $M \times T$, así W es la matriz de mezclas de tamaño $M \times K$ y S la matriz de fuentes de tamaño $K \times T$.

De manera general, los métodos que explotan las propiedades estadísticas fundamentales de las señales de voz (e.g. no-estacionariedad y no-gaussianidad) han sido muy populares para resolver tareas relacionada al procesamiento de voz [72]. ICA (Independen Component Analysis) y NMF (Non-negativa Matrix Factorizacion) son ejemplos típicos de este tipo de métodos. Estas técnicas también son conocidas como métodos de descomposición, y todos comparten el mismo principio donde una matriz V puede ser factorizada en otras dos matrices W y H. Sin embargo, cada método tiene sus propias restricciones: en ICA se asume que las filas de H son iid (independientes e idénticamente distribuidas) [73]; y, NMF tiene la restricción de que todas las matrices tienen que ser no-negativas. [74].

ICA se usa para resolver el problema de separación de fuentes o *speech enhancement* asumiendo que las fuentes son temporalmente iid y no-gaussianas. Bajos estas condiciones, ICA estima la matriz B también llamada matriz de separación o *demixing*, la cual lleva a la estimación de la señal deseada $\hat{S} = BY$. Dado el modelo de mezclas discutido arriba, $Y = WS$ y B corresponden a la inversa de W. De manera similar, NMF se usa para aproximar la matriz Y considerando las matrices W y H como matrices no-negativas [74]. Ambos métodos, ICA y NMF, han sido usado para resolver varias tareas relacionadas con el procesamiento de voz, y así se puede observar en la literatura. En [75], se uso NMF para realizar la tarea de *speech enhancement* en un ambiente ruidoso con un arreglo de micrófonos. Una solución similar se propone en [76] pero esta vez para la tarea de separación de fuentes, aquí los autores usaron *beamforming* para acelerar la convergencia del algoritmo ICA en condiciones estáticas y con dos niveles de reverberación (RT=150 y 300 ms). Los resultados mostraron una degradación del desempeño cuando la reverberación era mayor. En [77], se presenta un algoritmo multietapa basado en la versión temporal de ICA (TDICA - Time-Domain ICA) y la versión espectral (FDICA - Frequency-domain ICA). Los resultados muestran que FDICA se desempeña mejor que TDICA bajo condiciones reverberantes, pero aparece el problema

de la permutación de fuentes. Los autores además muestran que reducir el número de bines de la FFT mejora la robustez del problema de permutación, pero degrada el desempeño bajo reverberación. En otro interesante trabajo [78] se presenta una solución combinada utilizando NMF y cauterización por K-means para separación de fuentes, donde los autores probaron su método en múltiples bases de datos. El método muestra buen desempeño para mezclas lineales de voz y música, pero un desempeño deficiente para mezclas convolucionales. En [79], se presenta un método para separación de fuentes basado en NMF, donde los autores utilizan probabilidades bayesianas para modelar mutiles fuentes, y así resolver una de las restricciones de NMF, la cual es que el número de fuentes debe ser conocido. En [80] se utiliza IVA (independent vector analysis) el cual es una generalización de ICA para mejorar el desempeño en la tarea de separación de fuentes bajo escenarios no-estacionarios, esto se logra agregando un modelo gaussiano de mezclas para omitir la fase de preentrenamiento. Existe un problema significativo en los métodos clásicos o estadísticos para resolver la tarea de separación de fuentes, este problema es la dependencia en la cantidad de datos para optimizar sus funciones objetivos, esto impone una restricción cuando se aplican en ambientes dinámicos y variables en el tiempo como es el caso de HRI.

Speech enhancement y separación de fuentes son dos problemas fuertemente interrelacionados. Como se presenta en [81], los dos términos son intercambiables si resolvemos la tarea de extraer una fuente de voz y cancelar señales o ruidos interferentes. Simultáneamente, *Speech enhancement* es un término más general que se refiere a la extracción de una o más fuentes en presencia de reverberación, ruido o altavoces interferentes. Se han adoptado principalmente dos marcos algorítmicos para abordar el problema de *Speech enhancement* y la separación de fuentes, el uso de arreglos de micrófonos y la separación ciega de fuentes, estos dos marcos convergentes comparten teoría y conceptos y se utilizan conjuntamente en la literatura [82].

3.1.2. El problema de las fuentes múltiples y la reverberación en espacios interiores

En escenarios acústicos estáticos la reverberación se puede modelar como un sistema lineal variante en el tiempo. Con una respuesta de impulso reverberante (RIR), $h(t)$, la señal reverberada, $x(t)$, se puede modelar como $x(t) = \int_0^\infty s\left(t'\right) h\left(t - t'\right) dt'$, donde $s(t)$ es la señal de voz limpia. En el dominio de la frecuencia, la señal reverberada se puede expresar como $X(\omega) = H(\omega) S(\omega)$. La reverberación y el ruido aditivo reducen la inteligibilidad de las señales de voz y la precisión de los sistemas ASR, y los escenarios acústicos en interiores pueden ser muy reverberantes y ruidosos. Las técnicas clásicas de separación de señales de voz, como ICA y NMF, pierden eficacia con la reverberación [72]. Como se expresó anteriormente, la reverberación genera un proceso de mezcla convolutivo variable en el tiempo. También aumenta la autocorrelación de las señales de voz originales. La reverberación sigue siendo un desafío para los sistemas ASR en ambientes interiores, y se han propuesto muchos enfoques para reducir su efecto. En [83], se propuso el método Weighted-prediction-error (WPE), WPE consiste en un enfoque estadístico de desreverberación que ha demostrado ser muy eficaz y que modela el fenómeno de reverberación como un modelo autorregresivo (AR) [84]. Esta técnica se ha empleado ampliamente en otros trabajos e intenta dereverberar cada *bin* de frecuencia en la Transformada de Fourier (STFT) de una señal dada mediante el uso de un filtro lineal dependiente de la frecuencia.

El entrenamiento de los sistemas ASR bajo múltiples condiciones acústicas es un enfoque ampliamente adoptado que se utiliza en escenarios reverberantes y ruidosos. El entrenamiento multi condición de los sistemas ASR ha mostrado mejoras sobre el entrenamiento con señales limpias en comparación a cuando se usan señales ruidosas y reverberantes [85]. Además, ha mostrado mejoras cuando se usa en reconocimiento de voz con datos audio-visuales [86] y así también en detección de suplantación de identidad [87]. En [85], los autores implementaron una etapa de *speech enhancement* con un autoencoder (Denoising-Auto-Encoder - DAE) antes de ASR. El DAE fue entrenado usando señales de voz reverberantes como entrada y señales de voz limpia como objetivo. Los resultados mostraron que el uso de múltiples condiciones acústicas para el entrenamiento del ASR superó el uso de DAE con entrenamiento limpio en el ASR. Es decir, los autores obtienen un mejor resultado sin necesidad de realizar cancelación del ruido y reverberación. Sin embargo, limpiar o reverberar las señales de voz proporciona algunos beneficios como por ejemplo cuando se requiere obtener un perfil de la persona que emite la voz.

Para lograr una colaboración efectiva con las personas, los robots necesitan detectar y estimar un perfil a los usuarios con los que interactuarán para así modificar y adaptar su comportamiento de acuerdo a cada usuario. HRI requiere modelar y reconocer las acciones y capacidades humanas, para revelar las intenciones y metas detrás de tales acciones, y para determinar los parámetros que caracterizan la interacción social. La elaboración de perfiles de usuarios puede analizarse desde un punto de vista de interacción física, cognitiva y social [88].

Perfiles físicos − Este dominio comprende las características del usuario relacionadas con el cuerpo humano y los movimientos en el espacio. En consecuencia, perfilar a los usuarios de manera física corresponde a la detección de las capacidades de movimiento que están relacionadas con el proceso de interacción. Un ejemplo de esto sería una persona con movilidad reducida.

Perfiles cognitivos − Para que la comunicación sea efectiva existe la necesidad de predecir, detectar y reconocer las intenciones del agente observado [89] mas allá de las palabras. La capacidad de inferir y reconocer las intenciones, los deseos, las creencias, los estados internos, la personalidad y las emociones de los individuos. Un ejemplo de esto sería una persona que intente molestar al robot solicitándole tareas innecesarias o inapropiadas.

Perfiles sociales − Los robots deben reconocer e interpretar las señales sociales mostradas por un humano [90]. Las señales sociales se pueden definir como comportamientos observables que producen cambios de comportamiento durante la interacción. [91]. Un ejemplo de esto sería un momento social donde se deba guardar silencio por un determinado espacio de tiempo.

La creación de perfiles de usuario en los dominios físico, cognitivo y social es crucial para una comunicación natural entre humanos y robots, en este contexto, los robots sociales deben observar las entradas audio-visuales de los humanos. La voz transmite una gran cantidad de información lingüística y paralingüística (por ejemplo, prosodia). Más allá de los comandos de voz a los robots, el habla es una ventana a la condición psicológica, física y emocional de los humanos. No obstante, el análisis y el procesamiento del habla son muy sensibles a los

entornos de ruido, el canal acústico variable del tiempo y la reverberación en escenarios dinámicos. Los métodos de cancelación de ruido y reverberación permiten perfilar a los usuarios haciendo uso de su voz.

3.1.3. Aprendizaje profundo, separación de fuentes y *speech enhancement*

Las redes neuronales convolucionales (CNN) se han empleado en la literatura para extraer representaciones y características de señales de voz [92]. Además, las CNN se han utilizado en aplicaciones de voz en tiempo real [93]. Una de las arquitecturas recientes de CNNs más notable es ResNet [94] que soluciona el efecto negativo de apilar demasiadas capas en una red neuronal, lo que provoca una degradación de los resultados. La arquitectura ResNet se basa en una arquitectura CNN llamada VGG [95] y agrega una conexión residual a un bloque convolucional. Esta conexión encierra múltiples capas convolucionales y densas, lo que alivia el costo computacional sin perder rendimiento y mitiga el problema de degradación del gradiente [94]. Estas conexiones aseguran que una red a la que se le agrega una nueva capa funcione al menos tan bien como la red original. La justificación de esto se presenta a continuación, si $Q(x)$ es la función a la que una sección (es decir, una o más capas) de una CNN debe aproximar para producir buenos resultados, entonces puede aproximarse asintóticamente a una función residual $P(x) = Q(x) - x$. Si un bloque ResNet necesita aproximarse a la función de identidad, basta que todos los pesos sean cero. La conexión residual hace que sea trivial para la red no hacer nada a los datos de entrada si eso es lo mejor que se puede hacer. Esta arquitectura se ha utilizado ampliamente para múltiples tareas en el procesamiento de imágenes y otros campos.

Otra arquitectura notable de redes convolucionales es la FCN (*Fully Convolutional Network*) [96]. Esta arquitectura utiliza solo capas convolucionales, sin ninguna capa densa o *fully conected*, las capas densas se reemplazan por capas convolucionales con kernel de tamaño 1x1. La ventaja de usar solo capas convolucionales es que el tamaño de entrada puede ser arbitrario ya que la arquitectura solo tiene filtros de tamaño fijo que son independientes del tamaño de la imagen. Además, se espera que la correlación temporal de los frames se represente fácilmente en las capas más profundas, ya que los filtros mantienen la coherencia temporal. De manera diferente, en una capa densa o *fully conected*, todos los frames dentro de una ventana de análisis se procesan y se relacionan con todos los demás frames, lo que desordena la coherencia temporal. Un enfoque reciente basado en FCN es la TCN (*temporal convolutional network*) [97]. Esta arquitectura introduce filtros convolucionales unidimensionales con dilatación de los kernel, esta red puede reemplazar una red recurrente evitando el uso de mecanismos de activación. Una característica de TCN es el bajo número de parámetros necesarios para obtener resultados comparables a otras redes recurrentes [98] como LSTM.

Las tareas de separación de fuentes y *speech enhancement* han mejorado con el uso del aprendizaje profundo [75]. Se han empleado diferentes enfoques de aprendizaje profundo, como en [99], donde se presentó un modelo CNN-LSTM para separación de fuentes en escenarios con varios altavoces utilizando una extracción de características de varias etapas, mediante el uso de *auto-encoders* CNN y LSTM. Las características extraídas se concatenas antes de entrar a múltiples capas densas de salida. Los autores llegaron a la conclusión de que

el uso de ambos *auto-encoders* proporcionó mejores resultados que usando cada uno individualmente. Otro enfoque se presenta en [100], donde los autores describen una arquitectura LSTM que procesa información espacial y espectral. Como datos de entrada a la red, además del espectrograma se concatenaron características espaciales y direccionales relacionadas con la ubicación de las fuentes de sonido. Con el empleo de mecanismos de atención (*Attention Mechanism*) , los pesos fueron entrenados para optimizar el uso de características direccionales, espaciales y espectrales. En [101], se presentó un *auto-encoder* usando FCN que se evaluó en los datos del desafío REVERB. Este *auto-encoder* se usó para hacer *speech enhancement* procesando espectrogramas como imágenes con capas convolucionales bidimensionales. De manera similar, redes del tipo TCN que utilizan capas convolucionales unidimensionales también se han empleado para hacer *speech enhancement* en señales reverberantes [102].

3.1.4. *Compact bilinear pooling*

Las arquitecturas de aprendizaje profundo pueden interpretar combinaciones de múltiples señales o fuentes de información. Existen varias técnicas para realizar la fusión o combinación de esta información, como la concatenación, el producto punto y el producto externo o *bilinear pooling*. El rendimiento de *bilinear pooling* ha mostrado ventajas sobre otras técnicas de fusión, como se estudia en [103], donde los autores compararon diferentes técnicas para la fusión multimodal en una tarea de reconocimiento de emociones. Contrastaron *bilinear pooling* con la concatenación, la multiplicación elemento a elemento y la suma elemento a elemento, y *bilinear pooling* superó a todos ellos. Además, *bilinear pooling* usado con arquitecturas de aprendizaje profundo a sido utilizado para procesar eficazmente: texto y audio [103]; imagen y audio [104]; texto, audio y video [105]; y múltiples características o *features* de audio [106]. Sorprendentemente, *bilinear pooling* no se ha aplicado de forma exhaustiva a múltiples fuentes acústicas. En [106] se presenta un esquema de clasificación de escenas acústicas que utiliza *bilinear pooling* con información unimodal. Los autores extrajeron características armónicas y de percusión de espectrogramas de voz utilizando una CNN. Se empleó *bilinear pooling* para mezclar las características extraídas y clasificar las escenas acústicas.

Bilinear pooling es tan simple como un producto externo sobre dos vectores, estableciendo relaciones multiplicativas entre cada elemento de estos vectores. El problema más relevante de esta técnica es que produce una matriz de alta dimensionalidad. En [107] se presenta *Compact Bilinear pooling* que aborda ese problema de la matriz de alta dimensionalidad mediante una aproximación basada en muestras del *Bilinear pooling*. CBP se basa en Tensor Sketch Projection [108], que nos permite mapear un cierto vector en un espacio dimensional más bajo. Si u y w son dos vectores con la misma dimensionalidad, es posible omitir el cálculo directo del producto externo original $u \otimes w$ calculando la convolución de los vectores de características en un espacio de menor dimensión usando $\Psi\left(u \otimes w, h, s\right) = \Psi\left(u, h, s\right) * \Psi\left(w, h, s\right)$, en el que h y s son vectores de parámetros muestreados aleatoriamente.

3.1.5. Contribución de este capítulo

En este capítulo se aborda el problema de *speech enhancement* para mejorar señales de voz en entornos reverberantes interiores, esto se logra implementando una TCN con CBP.

El modelo adoptado incorpora la respuesta acústica para múltiples *beamformings* que filtran espacialmente las señales de voz y las fuentes de ruido. Teniendo en cuenta la aplicabilidad a entornos dinámicos variables en el tiempo, se explora y prioriza el uso de ventanas de análisis más cortas en el espectrograma de entrada. El objetivo final es permitir reconocimiento de voz en escenarios dinámicos variables en el tiempo como los que se encuentran en HRI con robots sociales y aplicaciones de altavoces inteligentes. Vale la pena enfatizar que la efectividad de los métodos de separación de fuentes ordinarios basados en modelos estadísticos como ICA y NMF depende del tamaño de la ventana de análisis y no pueden manejar ambientes con reverberación. El esquema de TCN/CBP propuesto es virtualmente independiente del tamaño de la ventana de análisis, al menos cuando es mayor de 1,6 s, lo que hace más abordable el problema de separación de fuentes en contextos variables en el tiempo. Además, se obtuvieron mejoras en WER de hasta 80 % sin y con WPE en comparación con ICA y NMF con un ASR entrenado en condiciones múltiples y pruebas reverberantes, y con experimentos de SNR variables en el tiempo para simular una fuente de voz objetivo en movimiento. Además, el experimento con la estimación de la señal limpia utilizando el esquema propuesto y el ASR entrenado con señales limpias proporcionó un WER 13 % menor que el obtenido con la señal corrupta y el ASR entrenado en condiciones múltiples. Este resultado desafía la práctica ampliamente adoptada de usar sistemas entrenados en condiciones múltiples y es consistente con el uso de métodos *speech enhancemet* que también pueden ser útiles para la elaboración de perfiles de usuarios en entornos HRI.

3.2. *speech enhancement* y escenarios reverberantes y variables en el tiempo

Un arreglo de micrófonos es un número arbitrario de micrófonos cuyas salidas se pueden procesar y combinar para obtener un filtrado espacial generando una única señal después de aplicar *beamforming*. El uso de un arreglo de micrófonos para realizar el *beamforming* puede reducir el efecto del ruido y la reverberación. En el caso de la reverberación, el filtrado espacial ayuda a suprimir las señales acústicas de que rebotan en las paredes del ambiente que también se llaman interferencia indirecta [109]. En el *beamforming delay-and-sum*, las muestras $y_m(t)$ de cada micrófono m son retrasadas en $\tau_{s_p,m}$ muestras y luego sumadas, donde s_p es la fuente objetivo. Lo cual entrega la señal única que entrega el *beamforming* $b_{s_p}(t)$ en el dominio de tiempo discreto, que se expresa a continuación.

$$b_{s_p}(t) = \sum_{m=0}^{M-1} y_m(t - \tau_{s_p,m}) \tag{3.2}$$

Se puede suponer un frente de onda plano si la distancia entre el arreglo de micrófonos y la fuente de sonido es mayor de $5 - 10$ veces la longitud del arreglo [110]. En consecuencia, el retardo de cada micrófono viene dado por:

$$\tau_{s_p,m} = \frac{\Delta_m \cdot sin\phi_{s_p}}{c} \tag{3.3}$$

donde Δ_m es la distancia entre el micrófono m y el micrófono de referencia. ϕ_{s_p} es el ángulo de incidencia (AOI) correspondiente a la fuente S_p, y c es la velocidad de propagación del sonido en el medio [111]. Reemplazando $y_m(t - \tau_{s_p,m})$ con (3.1) y cambiando el orden de la suma, $b_{s_p}(t)$ se obtiene la siguiente expresión.

$$b_{s_p}(t) = \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} s_{k,m}\left(t - \tau_{s_p,m}\right) \tag{3.4}$$

Considere que, dada una fuente k, $s_{k,m}$ tiene la misma energía para $0 \leq m < M$. Esta es una suposición razonable si los micrófonos son omnidireccionales y la distancia entre el arreglo de micrófonos y la fuente de sonido es mayor de $5 - 10$ veces la longitud del arreglo de micrófonos. El esquema de *beamforming delay-and-sum* condicionará la energía de la señal de la fuente k recibida con una ganancia que depende de la dirección a la que se apunta, que a su vez es una función de $\tau_{s_p,m}$:

$$b_{s_p}(t) = \sum_{k=0}^{K-1} G_{s_p,k} \cdot s_k(t) \tag{3.5}$$

donde $G_{s_p,k}$ denota la ganancia de la fuente k cuando el *beamforming* apunta a la fuente s_p. El retardo de tiempo debido a la propagación desde la fuente hasta el micrófono de referencia puede omitirse si se considera un escenario estático. Dada la Fig. 2 con fuentes s_0 y s_1, de la eq. (3.5) de obtienen las siguientes expresiones.

$$b_{s_0}(t) = G_{s_0,s_0} * s_0(t) + G_{s_0,s_1} * s_1(t) \tag{3.6}$$

$$b_{s_1}(t) = G_{s_1,s_0} * s_0(t) + G_{s_1,s_1} * s_1(t) \tag{3.7}$$

Según eqs. (3.6) y (3.7), $b_{s_0}(t)$ y $b_{s_1}(t)$ se pueden obtener agregando $s_0(t)$ y $s_1(t)$ a diferentes SNRs si $s_1(t)$ se considera ruido. En un ambiente interior real, el arreglo de micrófonos recibe la señal directa de cada fuente, pero también los reflejos correspondientes en las paredes, el suelo, el techo y otros objetos. En consecuencia, las ecs. (3.6) y (3.7) se pueden modificar para representar el problema de la Fig.3.2 con mayor precisión:

$$b_{s_0}(t) = h_{s_0,s_0} * s_0(t) + h_{s_0,s_1} * s_1(t) \tag{3.8}$$

$$b_{s_1}(t) = h_{s_1,s_0} * s_0(t) + h_{s_1,s_1} * s_1(t) \tag{3.9}$$

donde h_{s_0,s_0} y h_{s_0,s_1} son las RIRs observadas por el arreglo de micrófonos en la dirección de s_0 y s_1, respectivamente, cuando el *beamforming* apunta a s_0; h_{s_1,s_0} y h_{s_1,s_1} son las RIRs observadas por el arreglo de micrófonos en la dirección de s_0 y s_1, respectivamente, cuando el *beamforming* apunta a s_1. Observe que una RIR incorpora el retardo de ruta directa, por lo que h_{s_0,s_0}, h_{s_0,s_1}, h_{s_1,s_0} y h_{s_1,s_1} tienen en cuenta el retraso debido a la propagación desde la fuente hasta el micrófono de referencia.

Los métodos clásicos de separación de fuentes y *speech enhancement*, como ICA y NMF, no están diseñados para abordar el problema definido por las ecuaciones (3.8) y (3.9). Además, un modelo más genérico debería representar escenarios dinámicos variables en el tiempo con fuentes o arreglos de micrófonos en movimiento. En ese caso tendríamos que la respuesta al impulso queda descrita como $h(p', p_l, t)$, esta respuesta al impulso entrega una descripción espacio-temporal de la propagación del sonido desde la fuente en la posición p' al micrófono en la posición p_l. Esta RIR se le conoce como respuesta espacial al impulso o SRIR (Spatial Room Impulse Response) y es actualmente un problema vigente y en constante investigación. En [112] se presenta un método de estimación de la SRIR mediante la medición de una señal captada con un micrófono con trayectoria aleatoria y una fuente fija. Este método permite una representación visual bidimensional de la SRIR. En otro trabajo [113] se presenta un método de baja latencia para la construcción de la SRIR en un ambiente de realidad virtual, donde se tiene conocimiento preciso de la posición del emisor y el receptor del sonido. Trabajar con la construcción de SRIR esta fuera del objetivo de este trabajo, considerando que los escenarios dinámicos variables en el tiempo que se pueden caracterizar mediante una SRIR como los de las Figs. 3.1 y 3.2 todavía se pueden aproximar con las ecuaciones. (3.8) y (3.9) si asumimos las SRIRs como RIRs con emisores y receptores en posiciones fijas en intervalos de tiempo acotados, donde todo el sistema puede asumirse como cuasi-estático. Así tenemos:

$$b_{s_0}(t) \cong h_{s_0,s_0}(t) * s_0(t) + h_{s_0,s_1}(t) * s_1(t) \tag{3.10}$$

$$b_{s_1}(t) \cong h_{s_1,s_0}(t) * s_0(t) + h_{s_1,s_1}(t) * s_1(t) \tag{3.11}$$

Cuanto más corto sea la ventana de análisis, más precisa es la hipótesis cuasi-estática del entorno. No obstante, los métodos estadísticos pierden precisión cuando se reduce la ventana de análisis y no son buenos candidatos para abordar el problema de separación de fuentes como se define en las ecuaciones (3.10) y (3.11). Para contrarrestar esta limitación, en esta tesis se propone un método basado en el aprendizaje profundo que se compone de TCN y CBP que podrían entrenarse en múltiples condiciones y realizar la tarea de *speech enhancement* en ventanas de análisis más cortas. De ahora en adelante $b_{s_0}(t)$ y $b_{s_1}(t)$ se denotarán como $b_0(t)$ y $b_1(t)$, respectivamente. Observe que $b_0(t)$ y $b_1(t)$ indican los *beamformings* que apuntan

a las fuentes $s_0(t)$ y $s_1(t)$, respectivamente, donde $s_0(t)$ es la fuente de voz limpia y $s_1(t)$ corresponde a la fuente de ruido. En consecuencia, h_{s_0,s_0}, h_{s_0,s_1}, h_{s_1,s_0} y h_{s_1,s_1} se denotarán como h_{00}, h_{01}, h_{10} y h_{11}, respectivamente.

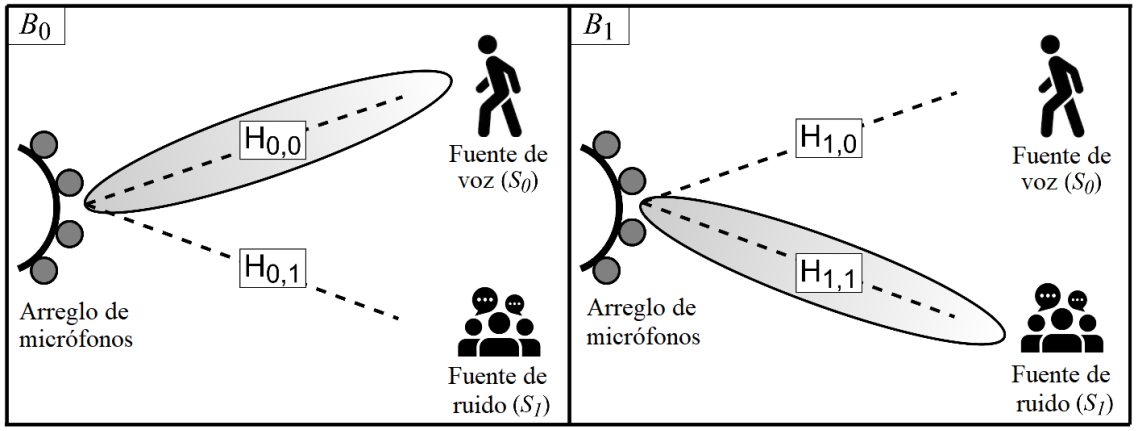


Figura 3.2: *Beamforming* para la fuente de voz(a) y *beamforming* para la fuente de ruido (b).

3.2.1. Solución propuesta

Speech enhancement en ambientes interiores reverberantes y dinámicos es un problema desafiante, que no se puede abordar con métodos convencionales como ICA y NMF porque el modelo de mezcla clásico no es adecuado. Considerando las dos fuentes que aparecen en la Fig. 3.2 (ruido y voz) y el correspondiente *beamforming*, con el modelo expresado en las ecuaciones (3.10) y (3.11), $B_0(\omega)$ y $B_1(\omega)$ corresponde al espectrograma de $b_0(t)$ y $b_1(t)$ respectivamente. Para enfrentar este desafío, en este capítulo se propone una arquitectura de *deep-learning* usando redes neuronales combinadas con CBP para obtener una versión desreverberada y sin ruido de la señal de voz objetivo (Ver Fig. 3.3). La solución propuesta hace frente a la restricción de un escenario acústico dinámico y variable en el tiempo.

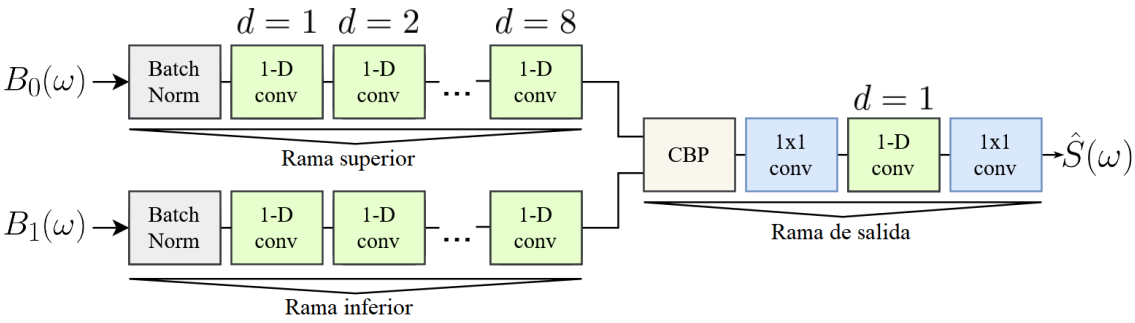


Figura 3.3: Arquitectura de aprendizaje profundo propuesta para la separación de fuentes de voz.

Se usaron bloques convolucionales unidimensionales con conexiones residuales y tasa de dilatación (d) creciente para extraer características de los espectrogramas (Fig. 3.3). Cada bloque convolucional tiene dos capas convolucionales, que se componen de un conjunto de múltiples filtros con parámetros entrenables que tienen una dimensión fija (Fig. 3.4). Estos filtros se convolucionan con la entrada para producir un nuevo mapa de características. Normalmente, en el procesamiento de imágenes, las capas convolucionales tienen filtros de dimensión 2 y se aplican horizontal y verticalmente. Los espectrogramas son representaciones temporales y espectrales de una señal que se puede visualizar como una imagen. Sin embargo, los ejes de tiempo y frecuencia no son simétricos, y no representan lo mismo como en el caso de una imagen donde los ejes verticales y horizontales son píxeles. Se usaron bloques convolucionales de dimensión 1 (llamados 1-D Conv en la Fig. 3.3) los cuales transitan a través de tiempo, para reducir la reverberación. La reverberación se puede modelar en cada bin de la Transformada de Fourier (STFT) de manera individual [83]. En este contexto, cada espectrograma se interpretó como una matriz de múltiples trayectorias o canales STFT. El número de trayectorias o canales corresponde al número de intervalos de frecuencia, es decir, 257 en el caso presentado en este trabajo, y la ventana de análisis o la longitud de la trayectoria de STFT se da en términos del número de frames.

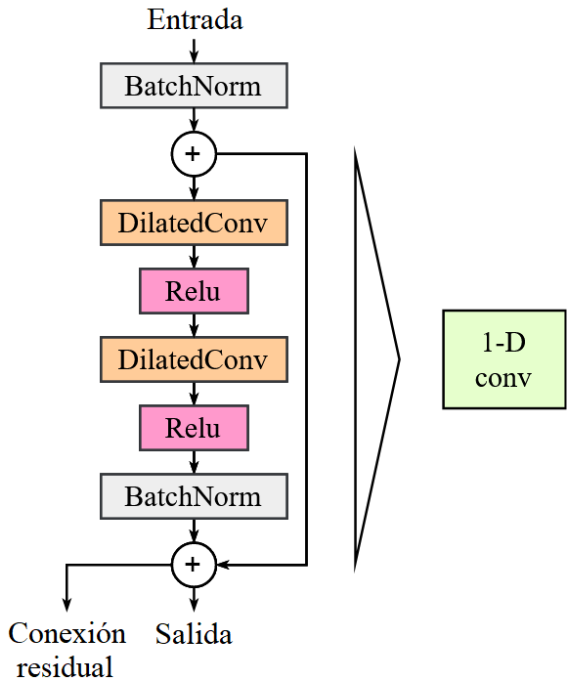


Figura 3.4: Detalle del bloque de convolucional unidimensional (en la Fig. 3.3) de la arquitectura de aprendizaje profundo propuesta.

En la Fig. 3.4, el bloque 1-D Conv de la figura 3.3 se muestra en detalle. Este, está compuesto por dos capas convolucionales con activación ReLu y normalización por *batch*, con una conexión residual desde la entrada del bloque hasta la salida de la normalización por *batch*. Ambas capas convolucionales tienen la misma tasa de dilatación y longitud de filtro. La normalización por *batch* se incorporó después de la activación de cada capa para acelerar la convergencia y evitar el sobreajuste. La salida de normalización por *batch* se suma a la conexión residual que da como resultado la entrada del siguiente bloque.

El tamaño de los filtros determina el campo receptivo de la entrada. Es deseable aumentar el campo receptivo para cancelar la reverberación. Sin embargo, existe un equilibrio entre el tamaño del filtro y la carga computacional. Dado que cada elemento de filtro es un parámetro entrenable, un tamaño de filtro más grande aumentará la carga computacional y la cantidad requerida de datos de entrenamiento. Una estrategia para suavizar esta restricción es utilizar un filtro convolucional dilatado [114]. Los bloques convolucionales 1-D Conv se apilan en la arquitectura con diferentes tasas de dilatación. En la Fig. 3.3, el primer bloque 1-D Conv consta de 257x257 filtros unidimensionales de longitud tres que se mueven a lo largo del eje del tiempo para cada convolución dilatada dentro del bloque. El segundo bloque consta del mismo número de filtros de longitud tres y tasa de dilatación 2; el tercer bloque 1-D es el mismo con la tasa de dilatación 4. Estos bloques 1-D Conv aumentan uniformemente en la tasa de dilatación hasta 8, que corresponde al último bloque 1-D Conv antes de CBP.

Después de procesar $B_0(\omega)$ y $B_1(\omega)$ con las ramas superior e inferior de la arquitectura propuesta en la Fig. 3.3, el bloque CBP se utiliza para obtener una representación conjunta de ambas salidas. Luego, las características resultantes se propagan a través de la rama de salida para obtener una estimación del espectrograma limpio $\hat{S}_0(\omega)$. La dimensión de la salida CBP es 257 canales x número de frames. La rama de salida se compone de una capa convolucional 1x1, un bloque convolucional 1-D y una capa convolucional 1x1. El primer bloque convolucional 1x1 usa activación ReLu, mientras que el bloque convolucional 1-D y la última capa convolucional 1x1 usa activación lineal.

3.3. Experimentos

El método propuesto se evaluó realizando experimentos con el sistema ASR limpio. Los resultados con el sistema ASR entrenado limpio son particularmente representativos porque el esquema TCN/CBP presentado ofrece una estimación de la fuente de voz original sin ruido ni reverberación. Sin embargo, también se obtuvieron resultados comparativos con el sistema ASR entrenado con condiciones múltiples. El esquema propuesto se comparó con ICA y NMF según la dependencia de la duración del intervalo de tiempo de análisis y la efectividad para llevar a cabo *speech enhancement* en presencia de reverberación. Los experimentos se llevaron a cabo con cuatro conjuntos de datos: primero, la base de datos que simula dos fuentes, es decir, la voz y ruido de restaurante, sin reverberación; en segundo lugar, uno que simula la fuente de voz y ruido en una condición reverberante, en donde el esquema TCN/CBP fue entrenado y probado con el mismo RIR; tercero, uno que emplea RIR de múltiples condiciones, es decir, el esquema TCN/CBP fue entrenado y probado con múltiples RIR; y cuarto, con multiples RIR's en varias condiciones y con SNR variable para simular una fuente de voz en movimiento. Los RIRs de condición múltiple significa que los RIRs de entrenamiento eran diferentes de los de prueba. ICA y NMF se aplicaron haciendo uso de los *toolkits* ICAmatlab [115] y FASST [116], respectivamente.

3.3.1. Generación de los conjuntos de datos

Se usaron los datos limpios de entrenamiento y validación del corpus AURORA-4, es decir, 7138 y 330 *utterances* limpias, respectivamente, para generar los conjuntos de entrenamiento y validación para el sistema TCN/CBP. Para el *test* o prueba, se usaron las 330 *utterances* limpias de la base de datos AURORA-4.

3.3.2. Generación de *beamforming* sin reverberación

Para generar $b_0(t)$ de acuerdo con (3.6), agregamos ruido de restaurante a cada *utterance* limpia de la base de datos de entrenamiento, validación y prueba con una SNR aleatoria entre 0dB y 15dB. La correspondiente señal $b_1(t)$, como se define en (3.7), se obtuvo agregando el mismo ruido de restaurante a una SNR igual a 3dB menor que la utilizada para $b_0(t)$.

3.3.3. Generación de *beamforming* con reverberación

Se simularon cuatro RIRs por cada *utterance* de entrenamiento, validación y prueba, utilizando Pyroomacoustics [117], es decir, h_{00}, h_{01}, h_{10} y h_{11} como se define en (3.8) y (3.9). Para generar $b_0(t)$ de acuerdo con (3.8), se convolucionó cada *utterance* limpia y una señal de ruido de restaurante con h_{00} y h_{01}, respectivamente. Luego, las señales convolucionadas resultantes se sumaron con un SNR aleatorio entre 0dB y 15dB. La *utterance* $b_1(t)$ como se define en (3.9) se obtuvo convolucionando la misma *utterance* limpia y la misma señal de ruido de restaurante con h_{10} y h_{11}, respectivamente. Las señales convolucionadas resultantes se sumaron con un SNR igual a 3dB menor que la utilizada para $b_0(t)$. En los experimentos con condiciones reverberantes coincidentes, es decir, RIRs coincidentes, todas las *utterances* de los conjuntos de entrenamiento, validación y prueba emplearon el mismo conjunto de h_{00}, h_{01}, h_{10} y h_{11}. En los experimentos multi-condición reverberante, es decir, RIR multi-condición, se generó un conjunto diferente de h_{00}, h_{01}, h_{10} y h_{11} para cada *utterance* limpia en el conjunto de datos de entrenamiento, validación y prueba. Para generar los RIR, se empleó el esquema presentado en la Fig. 3.5 en una habitación simulada de 2.5m x 6.0m x 6.0m (HxWxD) con una variación de distribución uniforme con un rango de $\pm 20\,\%$ en todas las dimensiones. El arreglo de micrófonos y las fuentes se colocaron en una ubicación aleatoria dentro de la habitación con una variación en el ángulo de $\pm\ 30°$ como se muestra en la Fig. 3.5. La distancia entre el micrófono y el altavoz fue obtenida a partir de una distribución uniforme aleatoria entre 1,6 y 2,4 m. El altavoz y el micrófono se colocaron en una ubicación aleatoria con la restricción de que estuvieran al menos a 1 m de cualquier pared, piso y techo. Se adopto un arreglo de micrófonos lineal de cuatro canales. Los micrófonos se colocaron en -11,3 cm, 3,6 cm, 7,6 cm y 11,3 cm con respecto al centro del arreglo. Estas dimensiones emulaban el arreglo de micrófonos del dispositivo Kinect [118]. Cada respuesta de micrófono se modeló como omnidireccional. Se utilizo *beamforming delay−and−sum* con retardos conocidos como se define en (3.2). En esta condición, h_{00} y h_{01} se obtuvieron apuntando el *beamforming* a la fuente objetivo S_0, y h_{10} y h_{11} se generaron apuntando el *beamforming* a la fuente S_1. Los cuatro RIR para la condición reverberante coincidentes se generaron con el valor medio de cada distribución empleada en las simulaciones.

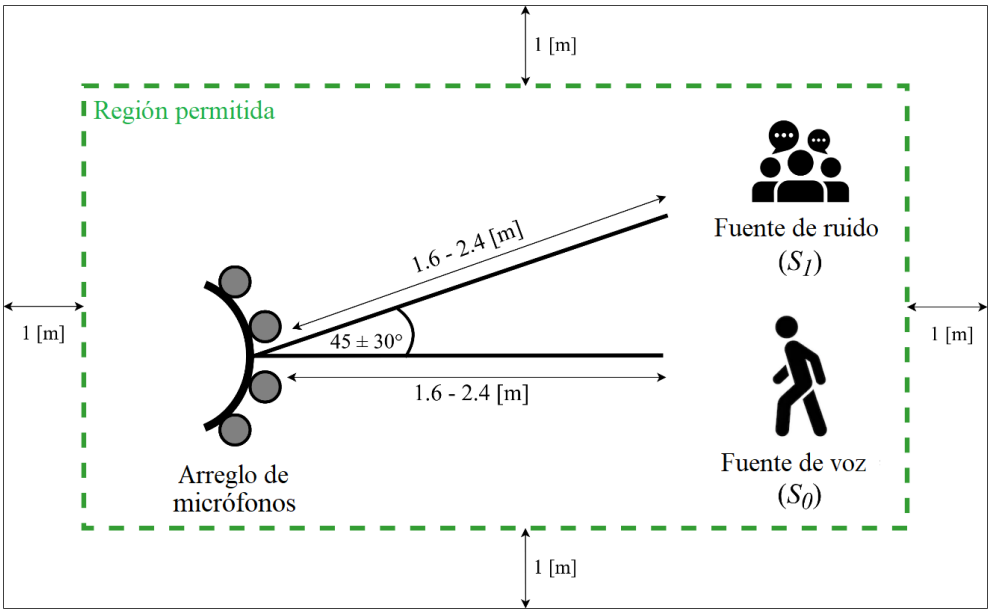


Figura 3.5: Escenario acústico simulado con Pyroomacoustics para generar los RIRs empleados en esta investigación.

3.3.4. Generación de *beamforming* variable en el tiempo

Basado en el mismo esquema presentado en la Fig. 3.5, se generó una base de datos para simular la variación SNR en el caso de fuentes de voz en movimiento. Se consideró un movimiento lineal entre 1.0m y 3.0 m, y viceversa, a una velocidad promedio de $5.0\frac{km}{h}$ con respecto al arreglo de micrófonos. La energía de la señal se modificó en cada muestra con una ganancia proporcional a $\frac{1}{x^2}$, donde x es la distancia entre la fuente de voz y el arreglo de micrófonos. La ganancia se aplicó de manera que la SNR a 2,0 m corresponda a la mencionada en la sección 3.3.2 para $(b_0(t))$. Esta base de datos se llama SNR variable en el tiempo.

3.3.5. Posible escenario experimental

Para realizar un escenario experimental real en base a este trabajo se requiere una disposición similar a la utilizada en el capítulo anterior. En principio se requiere una plataforma robótica real capaz de hacer *visual servoing*. Un arreglo de micrófonos sobre la plataforma robótica, fuentes de voz y ruido. La entrada de la red TCN/CBP son dos señales producidas mediante *beamforming*, por lo tanto se requiere conocimiento o estimación del TDOA. El TDOA se puede obtener mediante detección y rastreo de objetos como se explica en la sección 2.3.3, pero con la necesidad de detectar la fuente de voz y de ruido de manera simultanea, ya que se requieren estimar ambos TDOA para obtener ambos *beamformings*. Los movimientos de la plataforma robótica real pueden ser aleatorios y corresponder a las restricciones de la Fig. 3.5. Si los movimientos de la plataforma robótica suceden durante la reproducción de las utterance por la fuente de voz correspondería a un experimento de *beamforming* variable en el tiempo como en la sección 3.3.4. Si solo se cambia la posición del la plataforma robótica entre utterances y el robot no se mueve en la reproducción de estas correspondería a un experimento de *beamforming* no variable en el tiempo. En cualquier caso estaremos en

presencia de reverberación y ruido.

3.4. Entrenamiento del sistema TCN/CBP

La solución propuesta basada en *deep-learning* se entrenó con la API de Keras para TensorFlow [119]. Se uso Adam optimizer [120] para optimizar el modelo. Se adoptó el MSE como función de pérdida. El número de épocas se determinó mediante *early stoping* sobre el valor de la función de perdida en el conjunto de validación. Los hiperparámetros se ajustaron con experimentos ASR entrenados con señales limpias y los datos de prueba correspondientes a $b_0(t)$ sin reverberación. Las *utterances* de entrenamiento, validación y prueba se segmentaron en ventanas de análisis. Cada ventana de análisis se presenta al esquema de *deep-learning* como una unidad independiente y se compone de un número determinado de frames: 160, 320 o 640 frames. Cada frame corresponde a 25 ms con una superposición de 15 ms. Si el final de la última ventana de análisis no coincide con el final de la utterance correspondiente, entonces se aplica *reflect padding* [121]. Para mitigar el efecto de la variación aleatoria del parámetro inicial, la estructura propuesta TCN/CBP se entrenó tres veces para cada experimento aquí presentado. Se empleó una computadora de escritorio con un procesador Intel $i7-7700$, 32 GB de RAM y una GPU GeForce GTX1650 de 6 GB para los experimentos.

3.5. Sistema ASR

Se construyeron dos sistemas ASR basados en DNN-HMM utilizando la receta tri2b Kaldi para la base de datos AURORA-4. El primer sistema se entrenó con los datos limpios originales de la base de datos AURORA-4. El segundo fue entrenado con $b_0(t)$, donde los RIR's multi-condición fueron generados como en la sección 3.3.3. Excepto por los datos de entrenamiento, el procedimiento de entrenamiento empleado en ambos sistemas es el mismo. Primero, se construyó un GMM-HMM entrenando un sistema monofónico; luego, se emplearon las alineaciones de ese sistema para generar un sistema tri-fonema inicial; finalmente, se emplearon las alineaciones del sistema tri-fonema inicial para entrenar el sistema tri-fonema final. Esta receta también incluía coeficientes cepstrales de frecuencia Mel-(MFCC), análisis discriminante lineal (LDA) y transformadas lineales de máxima verosimilitud (MLLT). Una vez que se entrenó el sistema GMM-HMM, el GMM fue reemplazado por una DNN. La DNN estaba compuesta por siete capas ocultas y 2048 unidades por capa, y la entrada se consideraba una ventana de contexto de 11 frames. El número de unidades de la capa DNN de salida era igual al número de gaussianas en el sistema GMM-HMM correspondiente. La referencia para el entrenamiento de la DNN fue la alineación obtenida con el sistema GMM-HMM entrenado con los datos limpios y propagados sobre los mismos datos. Esto conduce a una mejor referencia para la DNN que usar datos de voz ruidosos o corruptos [122]. El vector de características constaba de 40 características del banco de filtros Mel (MelFB) y características dinámicas delta y delta-delta, utilizando una ventana de contexto de 11 frames. La DNN se entrenó inicialmente utilizando el criterio Cross-Entropy. Luego, el sistema final se obtuvo re-entrenando la DNN usando el entrenamiento discriminativo sMBR [123]. Para la decodificación, se utilizó el léxico estándar de 5K y el modelo de lenguaje de trigramas de WSJ [124]. Como resultado, el modelo de lenguaje depende de la tarea. El WER con los datos de prueba limpios arrojó 2.19 % que es competitivo con los publicados en otros lugares. El ASR

multi-condicion proporcionó un WER igual a 7.75 % en los datos de prueba correspondientes a $b_0(t)$ con RIR multi-condición. Se utilizó una computadora de escritorio con un procesador Intel i7-4790, 32 GB de RAM y una GPU GeForce GTX980 de 4 GB para el entrenamiento y decodificación de ASR.

3.6. Resultados y discusión

Cada resultado de reportado aquí corresponde a un WER promedio obtenido con tres redes neuronales TCN/CBP que fueron entrenadas para cada caso, esto se hizo para reducir el efecto de variación de las condiciones iniciales aleatorias como se discutió en la sección 3.4. En la Tabla 3.1, se muestra el WER de la estimación de la señal limpia original a partir de señales con ruido y sin reverberación mediante el uso de ICA, NMF y el esquema TCN/CBP propuesto. Comparado con la señal ruidosa $b_0(t)$, los tres métodos llevaron a reducciones de 77 %, 81 % y 78 %. NMF y TCN/CBP proporcionaron un WER que es solo 0.8 % y 1.33 % absoluto más alto que el WER obtenido con las señales limpias, es decir, 2.19 % (ver sección 4.3). Los resultados de los experimentos con diferentes tamaños de ventana de análisis se presentan en la Fig. 3.6. Se llevó a cabo *speech enhancement* para cada tamaño de ventana de análisis para ICA, NMF y el esquema TCN/CBP propuesto. Como se puede ver en la Fig. 3.6, el WER con ICA y NMF incrementó en un 81 % y 34 % relativo, respectivamente, cuando la ventana de análisis se redujo de 640 a 160 frames. Además, también hay degradación en WER cuando ICA y NMF se aplican en ventanas de 640 frames en lugar de la *utterance* completa. Este aumento en WER es igual a 21 % y 69 % con ICA y NMF, respectivamente. Por el contrario, TCN/CBP proporciona un WER que no se degrada cuando se acorta la ventana de análisis y también proporciona un WER ligeramente más alto cuando aumenta el tamaño de la ventana de análisis. Esto presumiblemente se debe al efecto del *padding* que se vuelve más abundante cuando aumenta el tamaño de la ventana de análisis. ICA y NMF se basan en métodos estadísticos y dependen en gran medida de la cantidad de datos disponibles. Por el contrario, los parámetros del método TCN/CBP propuesto se entrenan con la información proporcionada por las dos fuentes agregadas en dos SNR diferentes y no es necesario estimar ningún parámetro adicional en el proceso de prueba de *speech enhancement*.

Tabla 3.1: WER con experimentos de separación de fuentes sin reverberación. TCN/CBP usa una ventana de análisis igual a 160 frames, e ICA y NMF emplearon toda la *utterance*. Los experimentos se llevaron a cabo con un sistema ASR entrenado con señales limpias.

Metodo	WER
Baseline	15.90 %
ICA	3.74 %
NMF	2.99 %
TCN/CBP	3.52 %

Las figuras 3.7 y 3.8 muestran los resultados de *speech enhancement* en presencia de las condiciones de reverberación definidas en (3.8) y (3.9). En la Fig. 3.7, se probó el esquema propuesto bajo condiciones reverberantes coincidentes , es decir, con RIR coincidentes. Co-

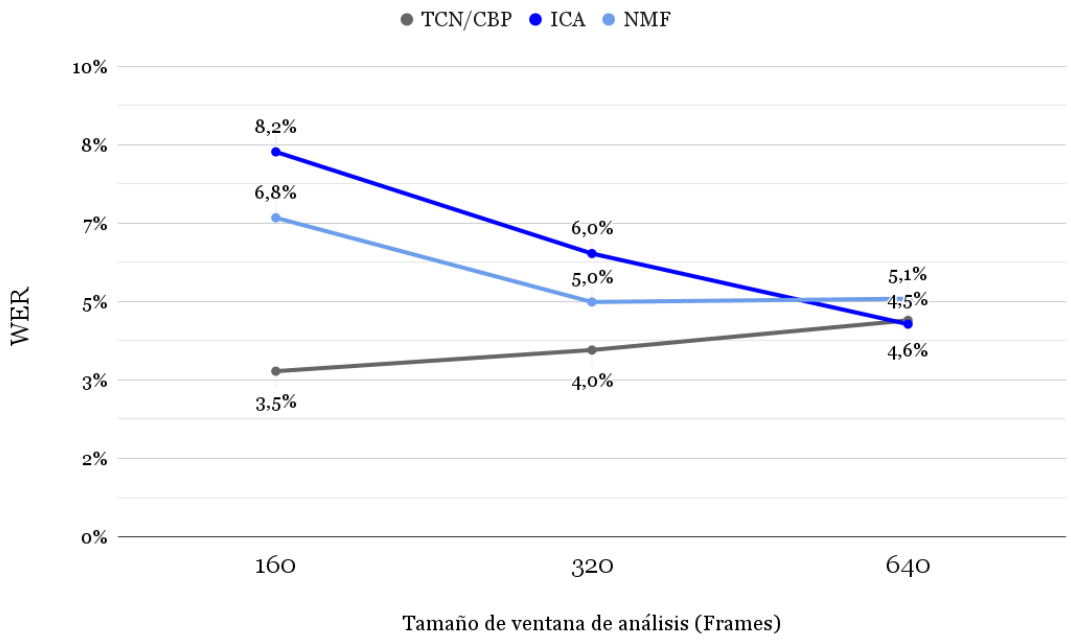


Figura 3.6: WERs en función del tamaño de la ventana de análisis en número frames. No se utilizó ninguna condición reverberante. Los experimentos se llevaron a cabo con el sistema ASR entrenado con señales limpias.

mo puede verse en la Fig. 3.7, ICA y NMF proporcionaron resultados incluso peores que el sistema base, es decir, con $b_0(t)$. En contraste, el sistema TCN/CBP propuesto proporcionó reducciones en WER tan altas como 91 % y 88 % sin y con WPE, respectivamente. Se pueden observar resultados similares en la Fig. 3.8 que presenta los WERs con múltiples RIR en el entrenamiento y prueba para el método TCN/CBP, es decir, RIRs multi-condición. Cabe destacar que los RIR empleados en los datos de prueba no se utilizaron en el procedimiento de entrenamiento. Como se puede ver en la Figura 8, ICA y NMF siguieron dando peores resultados que el sistema de referencia con $b_0(t)$ y TCN/CBP proporcionó reducciones en WER tan altas como 87 % y 82 % sin y con WPE, respectivamente. Observe que la mejora causada por WPE es bastante baja en comparación con la reducción total en WER causada por TCN/CBP. Además, de acuerdo con la Tabla 3.2, el rendimiento de TCN/CBP es independiente del tamaño de la ventana de análisis, excepto pequeñas variaciones que pueden ser causadas por la relación entre el número de parámetros entrenados y la cantidad de datos, y el efecto del *padding*. Observe que cuando se reduce el tamaño de la ventana de análisis, el número de frames de entrenamiento aumenta mientras que el número de parámetros entrenables libres se mantiene constante y el porcentaje de segmentos con *padding* también se reduce. En este punto, es interesante enfatizar que la independencia de TCN/CBP con respecto a la longitud de la ventana de análisis hace que el enfoque propuesto sea un candidato para abordar escenarios dinámicos variables en el tiempo más complejos como los modelados con (3.10) y (3.11).

La Figura 3.9 muestra los espectrogramas de la señal limpia original $s_0(t)$, las señales después de aplicar *beamforming* $b_0(t)$ y $b_1(t)$ y la señal limpia estimada, $\hat{s}(t)$. Como se puede ver en la Fig. 3.9, TCN/CBP efectivamente limpia la fuente de voz del ruido y elimina la

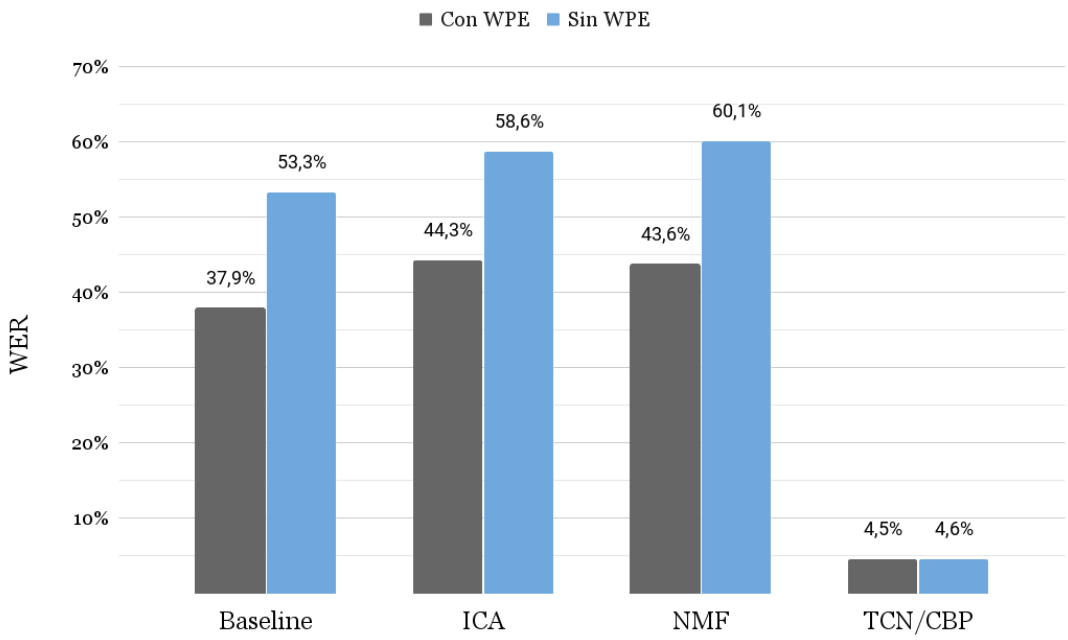


Figura 3.7: WERs de los experimentos de separación de fuentes en entornos reverberantes. Se utilizaron múltiples grupos de RIRs para entrenar y probar el esquema TCN/CBP propuesto, es decir, RIRs coincidentes. Se aplicó TCN/CBP 160 frames como ventana de análisis. Por el contrario, ICA y NMF se ejecutaron para toda la *utterance*. Los experimentos se llevaron a cabo con el sistema ASR entrenado con señales limpias.

Tabla 3.2: Resultados con separación de fuentes en condiciones reverberantes en función del tamaño de ventana de análisis. Los experimentos se llevaron a cabo con un sistema ASR entrenado con señales limpias.

Condiciones de prueba	Tamaño de ventana de análisis (frames)		
	160	320	640
RIRs coincidentes TCN/CBP	4.58 %	4.60 %	4.90 %
RIRs multi-condición TCN/CBP	6.76 %	6.73 %	7.08 %
RIRs coincidentes TCN/CBP + WPE	4.46 %	4.57 %	4.74 %
RIRs multi-condición TCN/CBP + WPE	6.35 %	6.10 %	6.75 %
SNR variable en el tiempo TCN/CBP	7.46 %	7.46 %	7.76 %

reverberación. De acuerdo con la Tabla 3.3, CBP conduce a una reducción en WER de 1.6 % relativo en comparación a la simple concatenación de *features* con el tamaño de la ventana de análisis igual a 160 cuadros. Este resultado sugiere que la red TCN en la Fig. 3.3 representa la mayor parte de la mejora observada en la disminución del WER. Además, la Tabla 3.3

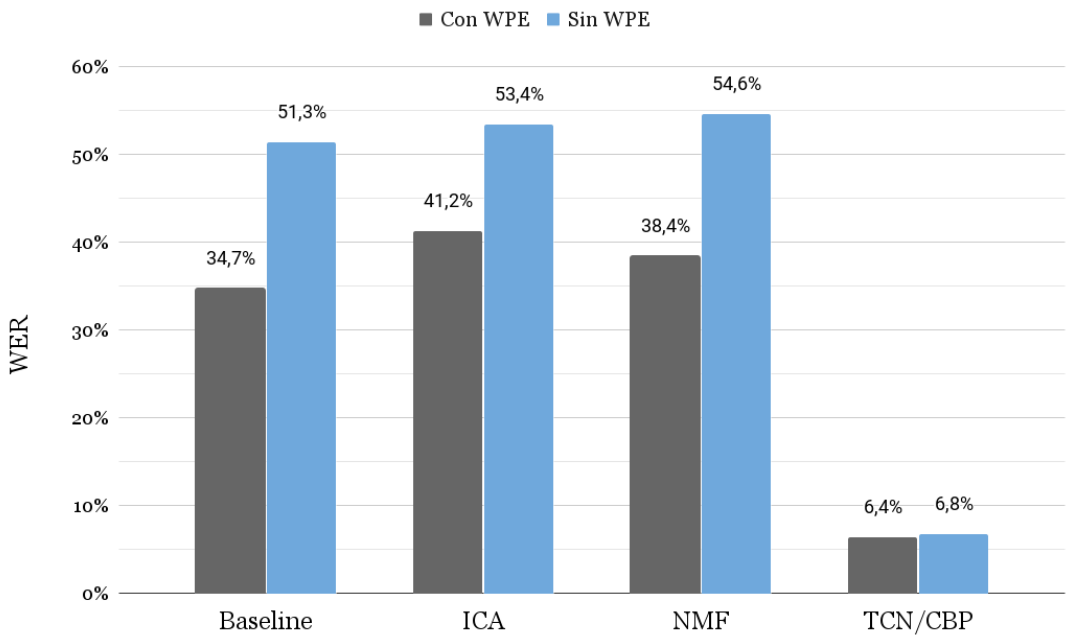


Figura 3.8: WERs de los experimentos de separación de fuentes en entornos reverberantes. Se utilizaron múltiples grupos de RIRs para entrenar y probar el esquema TCN/CBP propuesto, es decir, RIRs multi-condición. Se aplicó TCN/CBP 160 frames como ventana de análisis. Por el contrario, ICA y NMF se ejecutaron para toda la *utterance*. Los experimentos se llevaron a cabo con el sistema ASR entrenado con señales limpias.

también muestra que el esquema TCN/CBP aprovecha ambas entradas $b_0(t)$ y $b_1(t)$. Cuando solo está presente una señal, se obtienen peores resultados, es decir, se observa un aumento promedio de 6 % relativo en WER cuando el experimento se realiza solo con $b_0(t)$. Observe que $b_0(t)$ se generó con una SNR mayor que $b_1(t)$. De acuerdo con la Tabla 3.4, se lograron resultados similares con la base de datos SNR variable en el tiempo excepto por el hecho de que, en promedio, las WER en la Tabla 3.4 son 9 % más altas que en la Tabla 3.3. Esta degradación es relativamente baja si consideramos que la base de datos SNR variable en el tiempo es una tarea más compleja que la RIRs multi-condición, y este resultado confirma básicamente la pertinencia del enfoque propuesto de *speech enhancement* a corto plazo.

La Tabla 3.5 muestra una comparación entre los sistemas ASR entrenados con señales limpias y en condiciones múltiples con los datos RIRs multi-condición. Sorprendentemente, el experimento con la estimación de la señal limpia utilizando el esquema TCN/CBP propuesto, $\hat{s}(t)$, y el ASR entrenado limpio proporcionó un WER 13 % menor que el obtenido con $b_0(t)$ y ASR entrenado en condiciones múltiples. Este resultado contradice la práctica ampliamente adoptada de usar sistemas ASR entrenados en condiciones múltiples y respalda el uso de métodos de *speech enhancement* que también pueden ser útiles para la creación de perfiles de usuarios en entornos HRI. Finalmente, la Tabla 3.6 muestra las ventajas de TCN/CBP en comparación con ICA y NMF con respecto al tiempo de entrenamiento, el tiempo de prueba, la dependencia del tamaño de la ventana de análisis y la robustez a la reverberación.

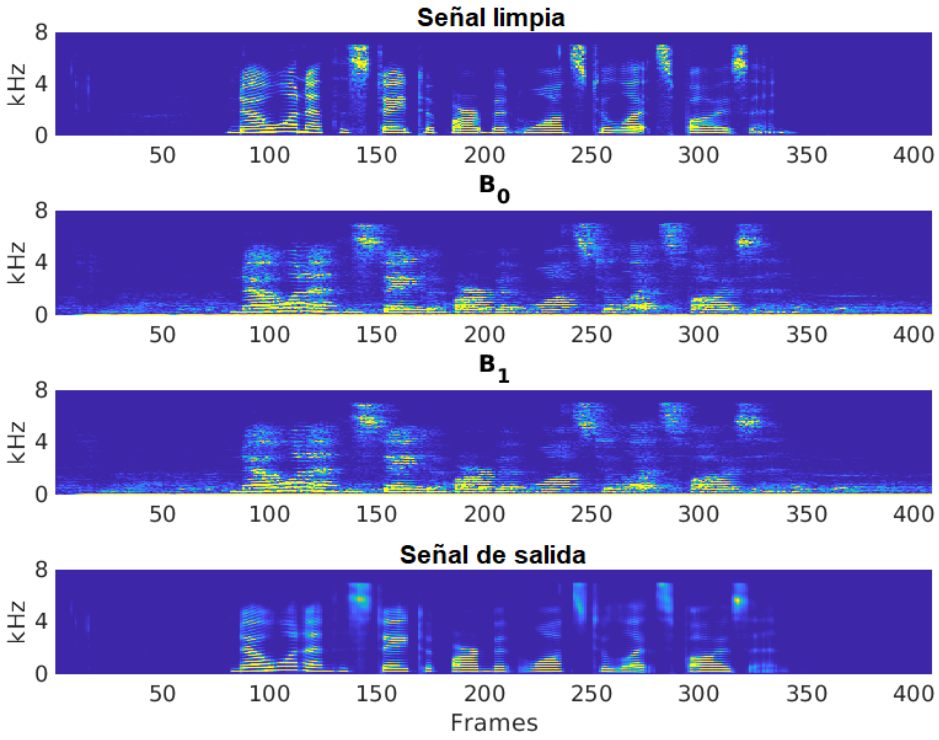


Figura 3.9: Espectrogramas de la señal limpia, las señales de *beamforming* $b_0(t)$ y $b_1(t)$, y la señal limpia estimada con el esquema TCN/CBP y la base de datos RIRs multi-condición.

Tabla 3.3: Comparación de CBP contra simple comparación de *features* en Fig. 3. También, se compara la el esquema de la Fig. 3.3 usando solo una señal de *beamforming*, es decir, solo la rama inferior o la rama superior, i.e. $B_0(\omega)$ o $B_1(\omega)$. Estos experimentos fueron llevados a cabo con la base de datos de RIRs multi-condición y el sistema ASR entrenado con señales limpias.

Condiciones de prueba	Tamaño de ventana de análisis (frames)		
	160	320	640
TCN/CBP	6.76 %	6.73 %	7.08 %
TCN/Concat	6.87 %	6.61 %	7.10 %
TCN solo con $B_0(\omega)$	7.34 %	6.76 %	7.68 %
TCN solo con $B_1(\omega)$	9.25 %	8.94 %	9.26 %

3.7. Conclusiones del capítulo

En este capítulo, se modelo el problema de *speech enhancement* con *beamformings* múltiples en entornos interiores reverberantes. Considerando que los modelos más genéricos deberían poder representar escenarios dinámicos variables en el tiempo con fuentes o arreglos de

Tabla 3.4: Comparación de CBP contra simple comparación de *features* en Fig. 3. También, se compara la el esquema de la Fig. 3 usando solo una señal de *beamforming*, es decir, solo la rama inferior o la rama superior, i.e. $B_0(\omega)$ o $B_1(\omega)$. Estos experimentos fueron llevados a cabo con la base de datos de SNR variable en el tiempo y el sistema ASR entrenado con señales limpias.

Condiciones de prueba	Tamaño de ventana de análisis (frames)		
	160	320	640
TCN/CBP	7,46 %	7,46 %	7,76 %
TCN/Concat	7,53 %	7,35 %	7,44 %
TCN solo con $B_0(\omega)$	7,95 %	7,42 %	8,36 %
TCN solo con $B_1(\omega)$	10,16 %	9,61 %	10,18 %

micrófonos en movimiento. Se pueden encontrar ejemplos típicos en HRI basados en voz con robots sociales o en aplicaciones de altavoces inteligentes. También se destaca que la efectividad de los métodos de *speech enhancement* ordinarios basados en modelos estadísticos como ICA y NMF depende del tamaño de la ventana de análisis y no pueden manejar entornos de reverberación. Para abordar estas limitaciones, se propone un método basado en una red convolucional temporal en combinación con CBP. El esquema propuesto es virtualmente independiente del tamaño de la ventana de análisis al menos cuando este es mayor que 1.6s, lo cual es muy interesante para abordar el problema de *speech enhancement* en escenarios variables en el tiempo. Además, se obtuvo mejoras en WER de hasta 80 % sin y con WPE en comparación con ICA y NMF con entrenamiento y pruebas reverberantes de múltiples condiciones. También se lograron resultados similares con experimentos de SNR variable en el tiempo para simular una fuente de voz en movimiento. Finalmente, el experimento con la estimación de la señal limpia empleando el esquema propuesto y el ASR entrenado limpio proporcionó un WER 13 % menor que el obtenido con la señal corrupta y el ASR entrenado en condiciones múltiples. Este resultado desafía la práctica ampliamente adoptada de usar

Tabla 3.5: Comparación de los WERs obtenidos con sistemas ASR entrenados limpio y multi-condición. Se evaluaron las siguientes condiciones de prueba: $b_0\,(t)$ y RIRs multi-condición; y la estimación de la señal limpia con TCN/CBP, $\hat{s}\,(t)$, también con RIRs multi-condición.

Condiciones de prueba	ASR entrenado limpio	ASR entrenado multi-condición
$b_0(t)$, RIRs Multi-condición	51.32 %	7.75 %
$\hat{s}_0(t)$, TCN/CBP RIR Multi-condición	6.76 %	20.29 %

Tabla 3.6: Comparación de método propuesto TCN/CBP con ICA y NMF como fueron implementados aquí.

	ICA	NMF	TCN/CBP
Tiempo de entrenamiento	N/A	N/A	1 hr.
Tiempo de inferencia (segundos por \textit{utterance}).	0.07s	55 s	0.03s
Dependencia en el tamaño de la ventana de análisis	Alta	Alta	Baja
Robustez a la reverberación	No	No	Si

sistemas entrenados en condiciones múltiples y fortalece el uso de métodos de mejora que también pueden ser útiles para la elaboración de perfiles de usuarios en entornos de HRI.

Capítulo 4

Conclusiones

En este trabajo de tesis se aborda el tema de la comunicación por voz en el contexto de interacción humano-robot en ambientes variables en el tiempo, ruidosos y reverberantes. Primero se estudia el efecto usar *visual servoing* con un arreglo de micrófonos en el desempeño de un sistema ASR. También se estudia la dependencia de la directividad de un *beamforming* con respecto al ángulo hacia donde apunta el arreglo de micrófonos. Y finalmente se muestra que a través de la combinación de varios *beamformings* se puede obtener un mejor desempeño en un sistema de *speech enhacement* basado en redes neuronales convolucionales.

En el capítulo 2 se evaluaron métodos estándar de *beamforming delay-and-sum*
y MVDR en combinación con *visual servoing*, además también se agregó rastreo de objetos para determinar el DOA en estos métodos de *beamforming*. Los resultados muestran que el uso de información visual para mejorar el desempeño los métodos de *beamforming* y que el mejor desempeño se obtiene con una combinación de *visual servoing* y rastreo de objetos. Estos resultados muestran una gran ventaja para escenario de HRI donde los robots naturalmente pueden mirar hacia las personas con quienes estén hablando y así apuntar un arreglo de micrófonos hacia la persona además de obtener la información visual para hacer rastreo de objetos. En el mismo capítulo también se estudió la dependencia de la directividad de un *beamforming* dependiendo de la orientación de un arreglo de micrófonos. Los resultados de este estudio muestran que la orientación de un arreglo de micrófono tiene una posición óptima que es necesaria controlar para hacer la comunicación en HRI lo más natural y efectiva posible.

En el capítulo 3 se implemento un sistema basado en redes neuronales convolucionales capaz de enfrentar el problema de *speech enhancement* con *beamformings* múltiples en entornos interiores reverberantes. Este sistema se comparó con dos metodos estadísticos tradicionales de *speech enhancement*, ICA y NMF. El método propuesto demostró baja dependencia al largo de la ventana de análisis lo cual permite enfrentar escenarios dinámicos y variables en el tiempo como lo pueden ser los escenarios típicos en HRI. Además de esto el sistema propuesto muestra que usar múltiples *beamformings* apuntando hacia la fuente de interés y la fuente de ruido entrega un mejor desempeño que solamente usando la información de la fuente de interés. Esto implica que la combinación de información unimodal entrega mejoras en el desempeño del sistema propuesto. La aplicabilidad de este esquema en HRI depende de si el robot posee una cámara para discriminar las posibles fuentes de ruido y de voz, lo cual es esperable y deseable.

Finalmente, este trabajo de tesis presenta dos aproximaciones para mejorar el desempeño en términos de WER de un sistema de reconocimiento automático de la voz en un escenario de interacción humano-robot. En ambos capítulos del cuerpo de esta tesis, se busca dar naturalidad a la comunicación en HRI ya que es lo esperable en futuro. Los robots deben mirar a las personas con quienes hablan para resultar natural y mostrar atención, y también para mejorar su capacidad de entender lo que les dicen. También, es necesario que los robots sean capaces de discriminar fuentes de ruido y fuentes de interés para mejorar la calidad de lo que escuchan, esto permitiría hacer estimaciones más precisas del contexto social donde el robot se desenvuelva y así aproximarse aún más a una comunicación totalmente natural.

Bibliografía

[1] A. Valli, "The design of natural interaction," in *Multimedia Tools and Applications*, vol. 38, pp. 295–305, Springer, 2008.

[2] M. A. Goodrich and A. C. Schultz, *Human-robot interaction: a survey*. Now Publishers Inc, 2008.

[3] L. S. Lopes and A. Teixeira, "Human-robot interaction through spoken language dialogue," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113)*, vol. 1, pp. 528–534, IEEE, 2000.

[4] G. Hoffman and K. Vanunu, "Effects of robotic companionship on music enjoyment and agent perception," in *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 317–324, IEEE, 2013.

[5] C.-Y. Lin, K.-T. Song, Y.-W. Chen, S.-C. Chien, S.-H. Chen, C.-Y. Chiang, J.-H. Yang, Y.-C. Wu, and T.-J. Liu, "User identification design by fusion of face recognition and speaker recognition," in *2012 12th International Conference on Control, Automation and Systems*, pp. 1480–1485, IEEE, 2012.

[6] K. Zheng, D. F. Glas, T. Kanda, H. Ishiguro, and N. Hagita, "Designing and implementing a human–robot team for social interactions," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, pp. 843–859, IEEE, 2013.

[7] K. Kumatani, T. Arakawa, K. Yamamoto, J. McDonough, B. Raj, R. Singh, and I. Tashev, "Microphone array processing for distant speech recognition: Towards real-world deployment," in *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, pp. 1–10, IEEE, 2012.

[8] S. Araki, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, T. Higuchi, T. Yoshioka, D. Tran, S. Karita, and T. Nakatani, "Online meeting recognition in noisy environments with time-frequency mask based mvdr beamforming," in *2017 Hands-free Speech Communications and Microphone Arrays (HSCMA)*, pp. 16–20, IEEE, 2017.

[9] S. Araki, N. Ono, K. Kinoshita, and M. Delcroix, "Meeting recognition with asynchronous distributed microphone array using block-wise refinement of mask-based mvdr beamformer," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5694–5698, IEEE, 2018.

[10] S. Araki, M. Okada, T. Higuchi, A. Ogawa, and T. Nakatani, "Spatial correlation model based observation vector clustering and mvdr beamforming for meeting recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 385–389, IEEE, 2016.

[11] T. Higuchi, N. Ito, S. Araki, T. Yoshioka, M. Delcroix, and T. Nakatani, "Online mvdr

beamformer based on complex gaussian mixture model with spatial prior for noise robust asr," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, pp. 780–793, IEEE, 2017.

[12] A. Díaz, R. Mahu, J. Novoa, J. Wuth, J. Datta, and N. B. Yoma, "Assessing the effect of visual servoing on the performance of linear microphone arrays in moving human-robot interaction scenarios," in *Computer Speech & Language*, vol. 65, p. 101136, Elsevier, 2021.

[13] I. J. Tashev, *Sound capture and processing: practical approaches*. John Wiley & Sons, 2009.

[14] M. Crocco and A. Trucco, "Stochastic and analytic optimization of sparse aperiodic arrays and broadband beamformers with robust superdirective patterns," in *IEEE Transactions on audio, speech, and language processing*, vol. 20, pp. 2433–2447, IEEE, 2012.

[15] F. Chaumette and S. Hutchinson, "Visual servoing and visual tracking," 2008.

[16] "Wer are we?." https://github.com/syhw/wer_are_we. Accessed: 2022-04-24.

[17] Y. Kondo, K. Takemura, J. Takamatsu, and T. Ogasawara, "A gesture-centric android system for multi-party human-robot interaction," in *Journal of Human-Robot Interaction*, vol. 2, pp. 133–151, Journal of Human-Robot Interaction Steering Committee, 2013.

[18] D. Wang, H. Leung, A. P. Kurian, H.-J. Kim, and H. Yoon, "A deconvolutive neural network for speech classification with applications to home service robot," in *IEEE Transactions on instrumentation and measurement*, vol. 59, pp. 3237–3243, IEEE, 2010.

[19] P. Szymański, P. Żelasko, M. Morzy, A. Szymczak, M. Żyła-Hoppe, J. Banaszczak, L. Augustyniak, J. Mizgajski, and Y. Carmiel, "Wer we are and wer we think we are," in *arXiv preprint arXiv:2010.03432*, 2020.

[20] J. Novoa, J. Wuth, J. P. Escudero, J. Fredes, R. Mahu, and N. B. Yoma, "Dnn-hmm based automatic speech recognition for hri scenarios," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 150–159, 2018.

[21] V. C. Mathad, T. J. Mahr, N. Scherer, K. Chapman, K. C. Hustad, J. Liss, and V. Berisha, "The impact of forced-alignment errors on automatic pronunciation evaluation," in *22nd Annual Conference of the International Speech Communication Association, INTERSPEECH 2021*, pp. 176–180, International Speech Communication Association, 2021.

[22] H. Nakajima, K. Nakadai, Y. Hasegawa, and H. Tsujino, "Moving sound source extraction by time-variant beamforming," in *Annual Conference of the Japanese Society for Artificial Intelligence*, pp. 47–53, Springer, 2007.

[23] Y. Sasaki, M. Kabasawa, S. Thompson, S. Kagami, and K. Oro, "Spherical microphone array for spatial sound localization for a mobile robot," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 713–718, IEEE, 2012.

[24] N. Tanaka, T. Ogawa, K. Akagiri, and T. Kobayashi, "Development of zonal beamformer and its application to robot audition," in *2010 18th European Signal Processing Conference*, pp. 1529–1533, IEEE, 2010.

[25] B. Kollmeier, T. Brand, and B. Meyer, "Perception of speech and sound," in *Springer handbook of speech processing*, pp. 61–82, Springer, 2008.

[26] D. Schnelle-Walka, S. Radeck-Arneth, C. Biemann, and S. Radomski, "An open source corpus and recording software for distant speech recognition with the microsoft kinect," in *Speech Communication; 11. ITG Symposium*, pp. 1–4, VDE, 2014.

[27] X. Anguera, C. Wooters, and J. Hernando, "Acoustic beamforming for speaker diarization of meetings," in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 2011–2022, IEEE, 2007.

[28] K. Niwa, Y. Hioka, S. Sakauchi, K. Furuya, and Y. Haneda, "Sharp directive beamforming using microphone array and planar reflector," in *Acoustical Science and Technology*, vol. 34, pp. 253–262, Acoustical Society of Japan, 2013.

[29] F. Jelinek, *Statistical methods for speech recognition*. MIT press, 1997.

[30] B. H. Juang and L. R. Rabiner, "Hidden markov models for speech recognition," in *Technometrics*, vol. 33, pp. 251–272, Taylor & Francis, 1991.

[31] H. Sun, T. L. Nwe, B. Ma, and H. Li, "Speaker diarization for meeting room audio," in *Tenth Annual Conference of the International Speech Communication Association*, 2009.

[32] D. Vijayasenan and F. Valente, "Speaker diarization of meetings based on large tdoa feature vectors," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4173–4176, IEEE, 2012.

[33] "Btk 2.0: Documentation." https://distantspeechrecognition.sourceforge.io/btk20_documentation/index.html. Accessed: 2021-02-24.

[34] H. Lu, T. Uemura, D. Wang, J. Zhu, Z. Huang, and H. Kim, "Deep-sea organisms tracking using dehazing and deep learning," in *Mobile Networks and Applications*, vol. 25, pp. 1008–1015, Springer, 2020.

[35] M. Matamoros, S. Viktor, and D. Paulus, "Trends, challenges and adopted strategies in robocup@ home," in *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 1–6, IEEE, 2019.

[36] S. Hori, Y. Ishida, Y. Kiyama, Y. Tanaka, Y. Kuroda, M. Hisano, Y. Imamura, T. Himaki, Y. Yoshimoto, Y. Aratani, *et al.*, "Hibikino-musashi@ home 2017 team description paper," in *arXiv preprint arXiv:1711.05457*, 2017.

[37] S. Argentieri, P. Danes, and P. Souères, "A survey on sound source localization in robotics: From binaural to array processing methods," in *Computer Speech & Language*, vol. 34, pp. 87–112, Elsevier, 2015.

[38] S. M. Naqvi, M. Yu, and J. A. Chambers, "A multimodal approach to blind source separation of moving sources," in *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, pp. 895–910, IEEE, 2010.

[39] S. M. Naqvi, W. Wang, M. S. Khan, M. Barnard, and J. A. Chambers, "Multimodal (audio–visual) source separation exploiting multi-speaker tracking, robust beamforming and time–frequency masking," in *IET Signal Processing*, vol. 6, pp. 466–477, IET, 2012.

[40] D. Pearce and J. Picone, "Aurora working group: Dsr front end lvcsr evaluation

au/384/02," 2002.

[41] M. Omologo, M. Matassoni, and P. Svaizer, "Speech recognition with microphone arrays," in *Microphone arrays*, pp. 331–353, Springer, 2001.

[42] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.

[43] "Kinect for windows sdk v1.8." https://www.microsoft.com/en-us/download/details.aspx?id=40278. Accessed: 2021-02-24.

[44] A. Sanna, F. Lamberti, G. Paravati, and F. Manuri, "A kinect-based natural interface for quadrotor control," in *Entertainment Computing*, vol. 4, pp. 179–186, Elsevier, 2013.

[45] L. Cheng, Q. Sun, H. Su, Y. Cong, and S. Zhao, "Design and implementation of human-robot interactive demonstration system based on kinect," in *2012 24th Chinese Control and Decision Conference (CCDC)*, pp. 971–975, IEEE, 2012.

[46] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'chime'speech separation and recognition challenge: Dataset, task and baselines," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 504–511, IEEE, 2015.

[47] I. J. Tashev, L. Le, V. Gopalakrishna, and A. Lovitt, "Cost function for sound source localization with arbitrary microphone arrays," in *2017 Hands-free Speech Communications and Microphone Arrays (HSCMA)*, pp. 76–80, IEEE, 2017.

[48] M. J. Crocker, "Theory of sound—predictions and measurement," in *Handbook of Noise and Vibration Control*, pp. 17–42, John Wiley & Sons, Inc. Hoboken, NJ, USA, 2007.

[49] H. L. Van Trees, "Modulation theory, part iv, optimum array processing," in *New York: Willey*, 2002.

[50] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF, IEEE Signal Processing Society, 2011.

[51] S. Sivasankaran, E. Vincent, and I. Illina, "A combined evaluation of established and new approaches for speech recognition in varied reverberation conditions," in *Computer Speech & Language*, vol. 46, pp. 444–460, Elsevier, 2017.

[52] P. Lin, D.-C. Lyu, F. Chen, S.-S. Wang, and Y. Tsao, "Multi-style learning with denoising autoencoders for acoustic modeling in the internet of things (iot)," in *Computer Speech & Language*, vol. 46, pp. 481–495, Elsevier, 2017.

[53] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks.," in *INTERSPEECH 2013*, vol. 2013, pp. 2345–2349, 2013.

[54] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept-sine technique," in *Audio Engineering Society Convention 108*, Audio Engineering Society, 2000.

[55] G. Hirsch, "Fant filtering and noise adding tool. software," 2005.

[56] M. Souden, J. Benesty, and S. Affes, "On optimal frequency-domain multichannel linear

filtering for noise reduction," in *IEEE Transactions on audio, speech, and language processing*, vol. 18, pp. 260–276, IEEE, 2009.

[57] J. Fredes, J. Novoa, S. King, R. M. Stern, and N. B. Yoma, "Locally normalized filter banks applied to deep neural-network-based robust speech recognition," in *IEEE Signal Processing Letters*, vol. 24, pp. 377–381, 2017.

[58] L. Madmoni and B. Rafaely, "Direction of arrival estimation for reverberant speech based on enhanced decomposition of the direct sound," in *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, pp. 131–142, 2019.

[59] S. Yan, "Optimal design of modal beamformers for circular arrays," in *The Journal of the Acoustical Society of America*, vol. 138, pp. 2140–2151, Acoustical Society of America, 2015.

[60] J. Benesty, J. Chen, and I. Cohen, *Design of circular differential microphone arrays*, vol. 12. Springer, 2015.

[61] A. Karbasi and A. Sugiyama, "A new doa estimation method using a circular microphone array," in *2007 15th European Signal Processing Conference*, pp. 778–782, IEEE, 2007.

[62] A. Parthy, N. Epain, A. van Schaik, and C. T. Jin, "Comparison of the measured and theoretical performance of a broadband circular microphone array," in *The Journal of the Acoustical Society of America*, vol. 130, pp. 3827–3837, Acoustical Society of America, 2011.

[63] G. Huang, J. Benesty, and J. Chen, "On the design of frequency-invariant beampatterns with uniform circular microphone arrays," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, pp. 1140–1153, IEEE, 2017.

[64] E. Tiana-Roig and F. Jacobsen, "Deconvolution for the localization of sound sources using a circular microphone array," in *The Journal of the Acoustical Society of America*, vol. 134, pp. 2078–2089, Acoustical Society of America, 2013.

[65] G. Huang, J. Chen, and J. Benesty, "On the design of robust steerable frequency-invariant beampatterns with concentric circular microphone arrays," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 506–510, IEEE, 2018.

[66] J. Meyer, "Beamforming for a circular microphone array mounted on spherically shaped objects," in *The Journal of the Acoustical Society of America*, vol. 109, pp. 185–193, Acoustical Society of America, 2001.

[67] D. Król, A. Lorenc, and R. Świcęiński, "Detecting laterality and nasality in speech with the use of a multi-channel recorder," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5147–5151, IEEE, 2015.

[68] A. Lorenc, D. Król, and K. Klessa, "An acoustic camera approach to studying nasality in speech: The case of polish nasalized vowels," in *The Journal of the Acoustical Society of America*, vol. 144, pp. 3603–3617, Acoustical Society of America, 2018.

[69] K. Haddad, J.-r. Hald, *et al.*, "3d localization of acoustic sources with a spherical array," in *Journal of the Acoustical Society of America*, vol. 123, p. 3311, 2008.

[70] G. Aleksei, V. Vladimir, A. Tseren, N. Sergey, L. Galina, V. Marina, G. Alice, S. Andrey, G. Artem, A. Anastasia, *et al.*, "Deep speaker embeddings for far−field speaker recognition on short utterances," in *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, pp. 179−−186, 2020.

[71] C. Danwei, Q. Xiaoyi, and L. Ming, "Multi−channel training for end−to−end speaker recognition under reverberant and noisy environment," in *INTERSPEECH 2019*, pp. 4365−−4369, 2019.

[72] W. Lin, D. Heping, and Y. Fuliang, "Speech separation and extraction by combining superdirective beamforming and blind source separation," in *Blind Source Separation*, pp. 323−−348, Springer, 2014.

[73] P. Bhanu and P. SR, *Speech, audio, image and biomedical signal processing using neural networks*. Springer, 2007.

[74] B. Hendrik, R. Klaus, K. Walter, B. Hendrik, R. Klaus, and K. Walter, *Informed Spatial Filtering Based on Constrained Independent Component Analysis*, pp. 237−−278. Springer, 2018.

[75] W. DeLiang and C. Jitong, "Supervised speech separation based on deep learning: An overview," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1702−−1726, 2018.

[76] S. Hiroshi, K. Toshiya, N. Tsuyoki, L. Akinobu, and S. Kiyohiro, "Blind source separation based on a fast−convergence algorithm combining ica and beamforming," in *IEEE Transactions on Audio, speech, and language processing*, 2006.

[77] N. Tsuyoki, S. Hiroshi, S. Kiyohiro, A. Shoko, and M. Shoji, "Multistage ica for blind source separation of real acoustic convolutive mixture," in *Proceedings of Fourth International Symposium on Independent Component Analysis and Blind Signal Separation*, 2003.

[78] O. Alexey and F. Cedric, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," in *IEEE Transactions on Audio, Speech, and Language Processing*, pp. 550−−563, 2009.

[79] N. Chaitanya, "A unified bayesian source modelling for determined blind source separation," in *INTERSPEECH 2019*, pp. 1343−−1347, 2019.

[80] G. Zhaoyi, L. Jing, and C. Kai, "Speech separation using independent vector analysis with an amplitude variable gaussian mixture model," in *INTERSPEECH 2019*, pp. 1358−−1362, 2019.

[81] G. Sharon, V. Emmanuel, M.-G. Shmulik, and O. Alexey, "A consolidated perspective on multimicrophone speech enhancement and source separation," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, pp. 692−−730, IEEE, 2017.

[82] V. Emmanuel, V. Tuomas, and G. Sharon, "Audio source separation and speech enhancement," John Wiley & Sons, 2018.

[83] N. Tomohiro, Y. Takuya, K. Keisuke, M. Masato, and J. Biing-Hwang, "Speech dereverberation based on variance−normalized delayed linear prediction," in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, pp. 1717−−1731, IEEE,

2010.

[84] L. Hao, Z. Xueliang, Z. Hui, and G. Guanglai, "Integrated speech enhancement method based on weighted prediction error and dnn for dereverberation and denoising," in *arXiv preprint arXiv:1708.08251*, 2017.

[85] M. Masato, S. Shinsuke, and K. Tatsuya, "Reverberant speech recognition combining deep neural networks and deep autoencoders augmented with a phone−class feature," in *EURASIP journal on Advances in Signal Processing*, vol. 2015, p. 62, Springer, 2015.

[86] Z. Shiliang, L. Ming, M. Bin, and X. Lei, "Robust audio−visual speech recognition using bimodal dfsmn with multi−condition training and dropout regularization," in *ICASSP 2019−2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6570−−6574, IEEE, 2019.

[87] Y. Hong, S. Achintya, T. Dennis, T. Zheng-Hua, M. Zhanyu, and G. Jun, "Effect of multi−condition training and speech enhancement methods on spoofing detection," in *2016 First International Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE)*, pp. 1−−5, IEEE, 2016.

[88] R. Silvia, F. Francois, and T. Adriana, "User profiling and behavioral adaptation for hri: a survey," in *Pattern Recognition Letters*, vol. 99, pp. 3−−12, Elsevier, 2017.

[89] D. Yiannis, "Prediction of intent in robotics and multi−agent systems," in *Cognitive processing*, vol. 8, pp. 151−−158, Springer, 2007.

[90] M. Derek, H. Alexander, H. Naoaki, N. Goldie, and B. Beno, "A survey of autonomous human affect detection methods for social robots engaged in natural hri," in *Journal of Intelligent & Robotic Systems*, vol. 82, pp. 101−−133, Springer, 2016.

[91] V. Alessandro and P. Alex, "New social signals in a new interaction world: the next frontier for social signal processing," in *IEEE Systems, Man, and Cybernetics Magazine*, vol. 2, pp. 10−−17, IEEE, 2015.

[92] van den Oord Aaron, D. Sander, Z. Heiga, S. Karen, V. Oriol, G. Alex, K. Nal, S. Andrew, and K. Koray, "Wavenet: A generative model for raw audio," in *9th ISCA Speech Synthesis Workshop*, pp. 125−−125, 2016.

[93] S. Abhishek and K. Nasser, "A convolutional neural network smartphone app for real−time voice activity detection," in *IEEE Access*, pp. 9017−−9026, IEEE, 2018.

[94] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770−−778, IEEE, 2016.

[95] S. Karen and Z. Andrew, "Very deep convolutional networks for large−scale image recognition," in *arXiv preprint arXiv:1409.1556*, 2014.

[96] L. Jonathan, S. Evan, and D. Trevor, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431−−3440, 2015.

[97] L. Colin, V. Rene, R. Austin, and H. Gregory, "Temporal convolutional networks: A unified approach to action segmentation," in *European Conference on Computer Vision*, pp. 47−−54, Springer, 2016.

[98] M. Matthew, W. Gordon, M. Emmett, and L. R. Jonathan, "Whamr!: Noisy and reverberant single−channel speech separation," in *ICASSP 2020− 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 696−−700, IEEE, 2020.

[99] Z. Jeroen *et al.*, "Cnn−lstm models for multi−speaker source separation using bayesian hyper parameter optimization," in *INTERSPEECH 2019*, 2019.

[100] G. Rongzhi, C. Lianwu, Z. Shi-Xiong, Z. Jimeng, X. Yong, Y. Meng, S. Dan, Z. Yuexian, and Y. Dong, "Neural spatial filter: Target speaker speech separation assisted with directional information," in *INTERSPEECH 2019*, pp. 4290−−4294, 2019.

[101] E. Ori, C. Shlomo, G. Sharon, and G. Jacob, "Speech dereverberation using fully convolutional networks," in *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 390−−394, IEEE, 2018.

[102] Z. Yan, W. DeLiang, X. Buye, and Z. Tao, "Monaural speech dereverberation using temporal convolutional networks with self attention," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, IEEE, 2020.

[103] A. Zakaria, D. Soheil, P. Dimitrios, and M. Emily, "Pooling acoustic and lexical features for the prediction of valence," in *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, pp. 68−−72, 2017.

[104] M. Saman and C. Israel, "Voice activity detection in presence of transient noise using spectral clustering," in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, pp. 1261−−1271, IEEE, 2013.

[105] S. Vivek, T. Makarand, and S. Rainer, "Deep multimodal feature encoding for video ordering," in *arXiv preprint arXiv:2004.02205*, 2020.

[106] K. Xing, C. Cheng, and L. Ye, "Acoustic scene classification using bilinear pooling on time−liked and frequency−liked convolution neural network," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 3189−−3194, IEEE, 2019.

[107] G. Yang, B. Oscar, Z. Ning, and D. Trevor, "Compact bilinear pooling," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 317−−326, IEEE, 2016.

[108] N. Pham and R. Pagh, "Fast and scalable polynomial kernels via explicit feature maps," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 239–247, 2013.

[109] K. Birger, B. Thomas, and M. Bernd, *Perception of speech and sound*, pp. 61−−82. Springer, 2008.

[110] T. Ivan, *Sound capture and processing: practical approaches*. John Wiley & Sons, 2009. John Wiley & Sons.

[111] C. Marco and T. Andrea, "Stochastic and analytic optimization of sparse aperiodic arrays and broadband beamformers with robust superdirective patterns," in *IEEE Transactions on audio, speech, and language processing*, pp. 2433−−2447, 2012.

[112] F. Katzberg, R. Mazur, M. Maass, P. Koch, and A. Mertins, "A compressed sensing framework for dynamic sound-field measurements," *IEEE/ACM Transactions on Audio,*

Speech, and Language Processing, vol. 26, no. 11, pp. 1962–1975, 2018.

[113] C. Schissler, P. Stirling, and R. Mehra, "Efficient construction of the spatial room impulse response," in *2017 IEEE Virtual Reality (VR)*, pp. 122–130, IEEE, 2017.

[114] Y. Fisher, K. Vladlen, and F. Thomas, "Dilated residual networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 472−−480, 2017.

[115] M. Brian, *PCA and ICA Package*. MATLAB Central File Exchange, 2020.

[116] O. Alexey and V. Emmanuel, *Using the FASST source separation toolbox for noise robust speech recognition*. Machine Listening in Multisource Environments, 2011.

[117] S. Robin, B. Eric, and D. Ivan, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 351−−355, 2018.

[118] W. Jarrett and A. James, "Beginning kinect programming with the microsoft kinect sdk," 2012.

[119] A. Martin, A. Ashish, B. Paul, B. Eugene, C. Zhifeng, C. Craig, C. Greg, D. Andy, D. Jeffrey, D. Matthieu, *et al.*, "Tensorflow: Large−scale machine learning on heterogeneous distributed systems," in *arXiv preprint arXiv:1603.04467*, 2016.

[120] K. Diederik and B. Jimmy, "Adam: A method for stochastic optimization," in *arXiv preprint arXiv:1412.6980*, 2014.

[121] "Tensorflow reflect padding documentation." https://www.tensorflow.org/api_docs/python/tf/pad. Accessed: 2021-02-24.

[122] S. Sunit, V. Emmanuel, and I. Irina, "A combined evaluation of established and new approaches for speech recognition in varied reverberation conditions," in *Computer Speech & Language*, vol. 46, pp. 444−−460, Elsevier, 2017.

[123] V. Karel, G. Arnab, B. Lukas, and P. Daniel, "Sequence−discriminative training of deep neural networks," in *INTERSPEECH 2013*, pp. 2345−−2349, 2013.

[124] G. Jean-Luc, L. Lori, and A.-D. Martine, "Developments in continuous speech dictation using the arpa wsj task," in *1995 International Conference on Acoustics, Speech, and Signal Processing*, 1995.