UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Complemento al Corpus de Evaluación Acoustic Interactions for Robot Audition con Fuentes Móviles

TESINA

Que para obtener el título de

Ingeniero en Computación

P R E S E N T A

Aldo Jesús Millán González

DIRECTOR DE TESINA

Dr. Caleb Antonio Rascón Estebané

Ciudad Universitaria, Cd. Mx., 2019

A mis padres Gabriela y Joel por su soporte incondicional y su gran amor, por todos los consejos que me brindan ya que sin ellos yo no sería nada.

A Joel, que más que un hermano es mi amigo, aunque no lo expresa sé que respalda todos mis sueños.

A Crystal, por impulsarme a concluir este ciclo y motivarme a seguir persiguiendo mis sueños, gracias por aparecer en el momento indicado y dejarme caminar a tu lado. Nunca olvidaré lo que has hecho por mí.

A aquellas personas que en su momento estuvieron presentes en el transcurso de mi vida personal y profesional, les estaré eternamente agradecido.

A todos mis profesores que han estado presentes a lo largo de mi trayectoria académica, pero en especial al Dr. Caleb Antonio Rascón Estebané por ser tan paciente y por la gran ayuda que me brindó al realizar esta tesina, pero en especial por brindarme un consejo que jamás olvidaré.

Reconocimientos

Agradecimientos a personas/entidades no humanas.

También quisiera reconocer a CONACYT por el financiamiento de los proyectos 81965, 178673 y 251319, a PAPIIT-UNAM por el proyecto IN107513, y a ICYTDF por el proyecto PICCO12-024.

Declaración de autenticidad

Por la presente declaro que, salvo cuando se haga referencia específica al trabajo de otras personas, el contenido de esta tesina es original y no se ha presentado total o parcialmente para su consideración para cualquier otro título o grado en esta o cualquier otra Universidad. Esta tesina es resultado de mi propio trabajo y no incluye nada que sea el resultado de algún trabajo realizado en colaboración, salvo que se indique específicamente en el texto.

Aldo Jesús Millán González. México, D.F., 2019

Resumen

Dentro de la Universidad Nacional Autónoma de México existen grupos enfocados a la investigación y desarrollo tecnológico como es el caso de grupo Golem, encargado de buscar la interacción entre humanos y sistemas computacionales apoyando proyectos que logren cumplir con este objetivo. El principal proyecto en el cual se encuentran trabajando hoy en día, es la creación de un robot de servicio que interactúe con el humano de la manera más natural posible, creando diversos módulos los cuales se enfocan a las distintas actividades que realiza el robot. La tesina que se presenta a continuación se especializa en la parte de audición del robot de servicio, logrando la ubicación exacta y la trayectoria que realizan fuentes de sonido que pueden permanecer en un estado estático o en movimiento.

Actualmente, existe el corpus que lleva por nombre *Acoustic Interactions for Robot Audition* (AIRA, por sus siglas en inglés) que es usado por el grupo para la evaluación de sus técnicas de Audición Robótica. Dentro de este corpus, se tiene las grabaciones de varias fuentes sonoras con posición exacta, y en algunas escenarios, dichas fuentes son móviles. Desgraciadamente, la trayectoria exacta de las fuentes móviles no fue capturada, sólo sus posiciones iniciales y finales. En este documento se describe el esfuerzo de complementar dicho corpus con nuevas grabaciones de fuentes móviles, anidado con su posición exacta mediante el uso de un seguidor de pies basada en un láser. Este seguidor fue codificado en lenguaje de programación c, para el análisis de los datos obtenidos. Finalmente, se llevó a cabo la evaluación de fuentes móviles con estos nuevos datos, y se identificaron áreas de oportunidad para el sistema del robot servicial Golem-II+, las cuales fueron resueltas. Su segunda evaluación confirma una mejora en el sistema de localización de fuentes sonoras que no hubiera sido posible identificar con los datos anteriores del corpus.

Cabe mencionar que parte de esta tesina fue publicado en la revista “The Journal of the Acoustical Society of America” (Vol.144, No.5), el articulo lleva por nombre “Acoustic interactions for robot audition: A corpus of real auditory scenes” [18]

Índice general

Índice de figuras

Capítulo 1

Introducción

1.1. Objetivo

El proyecto que a continuación se presenta en esta tesina, se ha enfocado en apoyar la labor de darle el sentido auditivo a un robot de servicio (es decir Audición Robótica), por medio de reforzar parte de la investigación ya realizada por el grupo Golem. En esta tesina se explicará a grandes rasgos, el cómo se llevó acabo la detección de fuentes móviles mediante un láser para poder identificar con exactitud la posición y así complementar el corpus de evaluación de la localización de fuentes sonoras que se encuentran en movimiento. El corpus con el que se evalúan los algoritmos actuales de localización se enfoca mayormente a fuentes estáticas, y la pequeña sección de fuentes móviles presenta la posición *estimada* de la fuente, no la posición real.

Se elaboró un programa para localizar fuentes móviles que se encuentran alrededor del robot de servicio Golem-II+, e identificar cual es la trayectoria que realizan por medio de un sistema de localización basado en láser. Mediante los datos obtenidos, se evalúa el algoritmo de localización actual para lograr una mayor calibración y precisión de éste.

1.2. Definición del Problema

Pensar en un robot que realice todo tipo de servicios no cuesta mucho ya que en alguna etapa de nuestra vida hemos llegado a imaginar un robot que pueda ayudar a realizar nuestras actividades diarias. Un robot de servicio cuenta con varios módulos de trabajo a nivel software y hardware que en su conjunto logran la movilidad e interacción del robot con un cliente y de esta forma lograr que sea lo más parecido a un ser humano. Para implementar un sistema encargado de tomar ordenes, se necesita de investigación constante.

La manera de entender y aprender algo en la vida diaria comienza por la vista y el escuchar. Estos son los sentidos esenciales para comenzar con algún tipo de aprendizaje,

pero también cabe mencionar que hay personas con discapacidad que sustituyen la manera de aprender mediante algún otro tipo de sentido. Por tal motivo, el tratar de hacer que un robot aprenda y tome decisiones por cuenta propia es algo muy complejo. Para poder hacer que interprete algunos mensajes o actividades humanas, es necesario dar al robot algún sentido semejante al del humano.

Por tal motivo el grupo Golem se dedica a investigar, implementar y actualizar cada una de las nuevas versiones del robot de servicio Golem-II+. Todas las aportaciones a la Audición Robótica del Grupo Golem son evaluadas con un corpus llamado *Acoustic Interactions for Robot Audition* (AIRA) recolectado anteriormente. Actualmente, este corpus sólo estima la trayectoria de fuentes sonoras móviles, ya que no se contó con una estimador de posición de personas durante la recaudación del corpus. El proyecto que se presenta en esta tesina complementa al corpus de evaluación con fuentes móviles con la ubicación exacta de cada uno de los usuarios (fuentes móviles).

Además de contribuir al artículo "*Acoustic interactions for robot audition: A corpus of real auditory scenes*" [18].

1.3. Motivación

Hoy en día, avances tecnológicos han sido de suma influencia en nuestras vidas. Han simplificado nuestras actividades diarias y nos han facilitado acciones que en su tiempo llegaron a tener una gran complejidad, como encontrar una dirección cuando estamos perdidos o comunicarnos por medios inalámbricos. Este proceso, normalmente, es motivado por el interés de que los dispositivos se comporten tan inteligentemente como lo hace un ser humano.

La IA es fundamental para lograr que los dispositivos se asemejen lo más posible a la capacidad mental del hombre y de que tomen sus propias decisiones. Sin embargo, ésta requiere de una investigación constante que se lleva a cabo con el paso del tiempo y la misma evolución humana. Aunque hablar de inteligencia artificial es llegar a pensar en un robot con una apariencia humanoide, ésta no es simplemente eso. También la encontramos reflejada en los dispositivos electrónicos que en ciertos casos en conjunto forman parte de un robot.

En un robot servicial, como es el caso de Golem-II+, es necesario que reciba las ordenes y logre ubicar a cada uno de los usuarios que están solicitando algún servicio dentro de un espacio. Para obtener las ordenes de los usuarios, en Golem-II+ se ocupa la funcionalidad de escucha. Los datos que son obtenidos mediante algoritmos de reconocimiento de voz y localización de fuentes sonoras deben ser frecuentemente evaluados para su mejora continua.

En la actualidad existen pocos recursos para la evaluación de algoritmos de Audición Robótica. Complementar el corpus ya existente en Golem-II+, y lograr la ubicación de las fuentes de sonido tanto estáticas como en movimiento, ayudarán a perfeccionar la funcionalidad de escucha del robot servicial Golem-II+. Esto aportará al grupo Golem un corpus más robusto y las herramientas necesarias para poder continuar con

investigaciones futuras.

1.4. Resultados Esperados

Los resultados que se pretenden conseguir con el desarrollo de este proyecto es complementar el corpus AIRA, para tener la posibilidad de no solo evaluar el desempeño de localización de fuentes sonoras estáticas, sino también de la localización de fuentes sonoras móviles así como su trayectoria. Aunque actualmente AIRA provee dicha información, ésta es basada en una estimación de la ubicación de la fuente sonora a partir de información conocida. La ubicación exacta en que se encuentra actualmente no es conocida en los registros de AIRA. Por lo tanto, este complemento proveerá dicha información de datos reales de localización y no solo de la estimación de los mimos. También se espera que con estos datos los algoritmos de Audición Robótica actualmente desarrollados sean evaluados de una manera más cercana a la realidad. Con esto, se pretende que dichos algoritmos sean mejorados a partir de las evaluaciones realizadas, lo cual impulsaría a la investigación y desarrollo de algoritmos de Audición Robótica.

1.5. Antecedentes

Existen distintas universidades enfocadas en la producción de dispositivos inteligentes como es el caso de la Universidad Nacional Autónoma de México y el Grupo Golem. Éste es un grupo de investigación y desarrollo tecnológico donde se busca la interacción entre humanos y sistemas computacionales y su proyecto principal es un robot de servicio que interactúe con el humano de la manera más natural posible.

1.6. Publicación

Este proyecto ha sido evaluado y publicado en la revista académica "*The Journal of the Acoustical Society of America*" con el título *Acoustic interactions for robot audition: a corpus of real auditory scenes* [18]. Para mayor información consultar la página: https://aira.iimas.unam.mx/.

1.7. Estructura del Documento

Capítulo 2. Marco teórico En este capítulo se presenta la teoría que fundamenta el proyecto y logra cumplir con los objetivos del problema a resolver de este proyecto.

Capítulo 3. Metodología Dentro de este capítulo se encuentra la lógica de los métodos que se siguieron para cumplir con las expectativas de los algoritmos de Audición Robótica del robot de servicio Golem-II+. Se describe un conjunto de programas

que logran la captura de información de fuentes sonoras tanto en movimiento como estáticas.

Capítulo 4. Evaluación En este capítulo se presenta la evaluación de un algoritmo de Audición Robótica previamente desarrollado en el grupo Golem-II+. Esta evaluación utiliza los datos recaudados, que contienen la ubicación y trayecto que realizan las fuentes sonoras. Se presentan los resultados de una evaluación en el que dicho algoritmo presenta áreas de oportunidad que no se habían presentado con los datos anteriores del corpus. También se presentan los resultados de una segunda evaluación, tras haber recalibrado dicho algoritmo, el cual muestra una mejora que no hubiera sido posible con los datos anteriores.

Capítulo 5. Conclusión En este capítulo se presentan las conclusiones del proceso de desarrollo, así como de los resultados obtenidos.

Capítulo 2

Marco teórico

En este capítulo se explican las definiciones esenciales por la parte de software y hardware. Específicamente, se describirán los conceptos con los que se estableció la conexión para poder capturar el sonido y localizar de manera precisa los usuarios que harán un recorrido alrededor del robot de servicio Golem-II+, capturando tanto su posición como los enunciados que reciten.

2.1. Captura de Sonido

Para poder resolver la problemática de audición e interpretación de un robot como Golem-II+, es necesario poder localizar las fuentes de sonido que encontramos en nuestro entorno, y a su vez lograr la separación de las mismas. En esta sección se presentarán una serie de conceptos relacionados con la audición y todo lo involucrado con sistemas computacionales y tecnológicos que han sido indispensables para lograr la interacción humano-robot en el contexto de audición.

De igual manera uno de los objetivos principales que se desea lograr con Golem-II+ es que la interacción con el humano sea lo más natural posible. Para esto la comprensión de las órdenes de los usuarios debe ser precisa para nuestro robot. Si entráramos en un momento de reflexión, nos daríamos cuenta que el proceso de audición que realiza nuestro cuerpo es muy complejo. Es sorprendente la manera con la que trabaja el oído humano y como procesa las diversas señales. Es decir, el cuerpo humano es una máquina muy difícil de igualar, pero no imposible de llegar a su semejanza.

Es clara la complejidad que trae consigo el desarrollo de un robot que procese las señales y comprenda a la perfección las ordenes expresadas por algún usuario.

2.1.1. Concepto de audio/sonido

El audio es una señal analógica que ocurre en un cierto intervalo de frecuencias. Así mismo, se entiende por señal analógica aquella que su amplitud puede tomar un valor cualquiera dentro de un conjunto continuo de valores [3]. Para que una señal analógica

pueda ser procesada por nuestro sistema computacional es necesario convertirla a una señal discreta y luego cuantizarla o representarla en un formato binario de L bits. Es decir, es necesario convertir dicha señal eléctrica a una representación binaria que un ordenador pueda analizar. Obsérvese en la Figura 2.1 la forma de las señales [1].

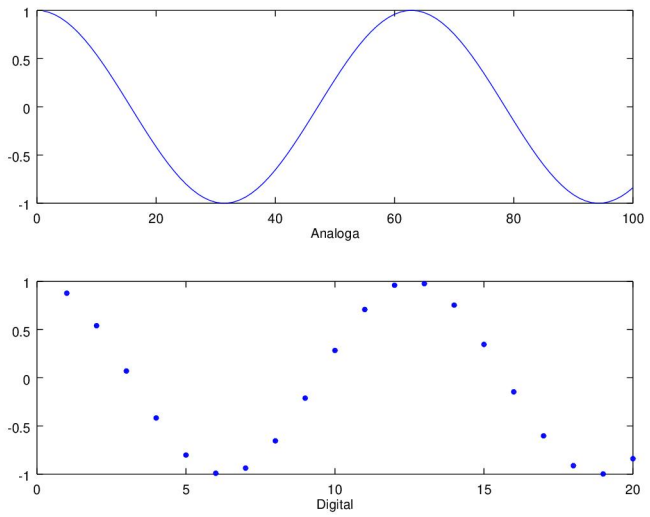

Figura 2.1: Descripción de señales análogas y digitales.

El concepto de sonido se puede definir como la interpretación que realiza nuestro cerebro de las variaciones de presión que genera un objeto vibrante en un medio como el aire, el agua y otros medios materiales. A esta señal se le conoce una onda sonora. Los objetos vibrantes pueden ser instrumentos de cuerdas o las cuerdas bucales de alguna persona. Éstos producen movimientos vibratorios que desplazarán a las moléculas que se encuentran a su alrededor, presentando el mismo comportamiento de vibración el cual será detectado por diafragmas móviles tales como un micrófono o el tímpano del ser humano [5].

> "La ingeniería del sonido o acústica, trata de los problemas de la transmisión y de la reproducción del sonido, así como del control y supresión del ruido."
> – A. H. Cromer, 1986. [5]

2.1.2. Ruido e Interferencia

Los sistemas eléctricos producen o reciben señales analógicas que pueden estar acompañados de señales no deseadas, contaminando los valores que pueden provocar errores y variación en la precisión [8]. Dichas señales son capturadas de fuentes internas o externas que pueden estar presentes en cada una de las capturas realizadas y pueden

ocurrir de manera momentánea, intermitente o periódicamente. Las señales no deseadas pueden ser eliminadas con diversos métodos matemáticos, pero es importante saber su comportamiento y cuál fuente es la que la esta produciendo dicho error (interno o externo).

Existen diversos efectos que producen la alteración de nuestras señales como es el caso de la atenuación, la distorsión, la interferencia, el ruido, etc. En esta sección sólo se explica qué es la interferencia y el ruido, no porque los demás efectos carezcan de importancia, sino porque éstas dos son las más relevantes a este proyecto.

2.1.2.1. Ruido

El ruido es una señal impredecible y aleatoria que se puede encontrar dentro o fuera del sistema de una forma natural y de tipo eléctrico [11]. Este tipo de señales se "ocultan" al fusionarse con la señal de la fuente de interés, ya que sus componentes de tiempo-frecuencia normalmente se empalman con la fuente de interés. Es importante mencionar que, justamente por dicho empalme, no se puede eliminar completamente de la señal capturada, lo cual lo hace diferente a todas los demás tipos de señales contaminantes.

Como se explicó anteriormente hay una gran variedad de ruido que afectarán a la señal capturada: tanto en su origen o por la naturaleza del entorno en el que se estén realizando las pruebas. Algunas de estas señales pueden ser eliminadas parcialmente por medio de filtrado.

2.1.2.2. Interferencia

Las interferencias son las señales que presentan similitud con nuestra señal original, originadas generalmente de una manera artificial [13]. El receptor puede captar dos o más señales de manera simultánea y la solución a este tipo de problema es eliminar las señales interferenciales o la fuente que la produce. La interferencia se presenta como fuentes que se encuentran a nuestro alrededor al realizar las prácticas necesarias. Ejemplos son las personas que están a nuestro alrededor, los coches que circulan por las calles aledañas o el simple canto de un pájaro. La presencia de interferencias es un obstáculo a vencer durante la separación de fuentes sonoras, ya que se requiere procesar los datos auditivos de sólo la fuente de interés para poder establecer una comunicación robusta con un robot de servicio como Golem-II+. [15]

2.1.3. Sonido digital

Como ya se mencionó anteriormente, el sonido es considerado en términos matemáticos como una señal ya sea analógica o digital. Es bien sabido que si un sonido es almacenado en algún repositorio, éste se puede reproducir, copiar, transferir y manipular las veces que sean necesarias. Para realizar estas actividades por medio del uso de alguna computadora, las señales almacenadas deben ser digitales. [14]

Por un lado, una señal continua se define como una función en el tiempo, expresada matemáticamente como $x(t)$. Es decir, una señal en tiempo continuo está definida en todo tiempo t, y su amplitud varía continuamente en el tiempo. Este tipo de señales surgen cuando una señal física (como una onda acústica) se convierte en una señal eléctrica. Para poder realizar la conversión de una onda acústica a una señal eléctrica es necesario un micrófono. Éste es un transductor que convierte la variación de presión del sonido en las correspondientes variaciones de voltaje o corriente. Se entra en más detalle en la sección 2.3.

Por otra parte, una señal en tiempo discreto se define sólo en instantes de tiempo discretos [7]. Esto quiere decir, que la variable independiente (el tiempo) toma valores discretos, espaciados de manera uniforme. Las señales en tiempo discreto se pueden derivar de una señal en tiempo continuo muestreándola a una tasa uniforme.

El muestreo de una señal que está en función de un tiempo continuo $x(t)$ se discretiza tomando al tiempo como el periodo de muestreo multiplicado por nuestro valor entero, dícese:

$$t = np \tag{2.1}$$

donde p es el periodo de muestreo y n es un número entero que pueda tomar valores positivos. De tal manera, obtenemos que:

$$x[n] = x(np) \quad \text{para} \quad n = 0, +1, +2, ... \tag{2.2}$$

Así una señal en tiempo discreto se representa por medio de una secuencia de números $x[0], x[1], x[2], ...$, llamada serie de tiempo [7], luego se cuantizan para representarse en forma binaria y ser tratada digitalmente.

2.2. Hardware para captura de sonido digital

Se considera el hardware como la parte física que conforma a un sistema informático. En el caso de la implementación de un robot de servicio, hay diversos tipos de hardware que tienen funcionalidades diferentes, pero todas estas partes van relacionadas a un objetivo principal: la interacción humano-robot. En el caso de Audición Robótica, el hardware relevante es aquél que lleva a cabo la captura de las diversas fuentes sonoras que están alrededor del robot de servicio Golem-II+.

Este proceso de captura requiere la conexión de diversos dispositivos por los cuales circula la señal portadora de información auditivo. La calidad de estos dispositivos es en lo que depende la calidad de la grabación y la fidelidad del resultado final [2]. Dicho conjunto de conexiones son resumidas en el esquema presentado en la Figura 2.2.

En esta sección, se describe las partes de hardware que son necesarias para lograr la captura de audio. Se toma como referencia al esquema presentado en la Figura 2.2, explicando de izquierda a derecha. Es importante mencionar que el hardware utilizado para este proyecto ya había sido seleccionado con anterioridad por el grupo Golem.

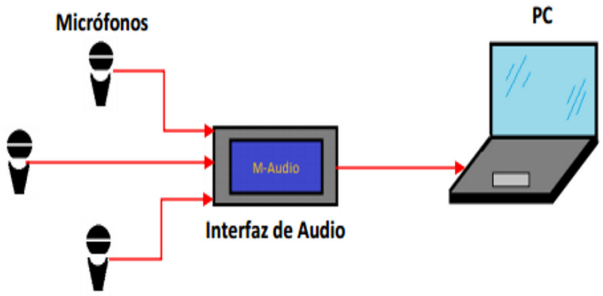

Figura 2.2: Conexión para detección de fuentes.

2.2.1. Micrófono

Un micrófonos convierte las variaciones de la presión de aire en variaciones análogas de corriente eléctrica. De esta forma, el voltaje de la señal eléctrica capturada es proporcional a las variaciones de presión de aire.

Se utilizó un micrófono marca SHURE MX391, mostrado en la Figura 2.3, el cual fue seleccionado por el grupo Golem. Las características de este micrófono son:

1. Una respuesta de frecuencia plana en el intervalo vocal, lo cual produce un sonido sin coloración.

2. Es omnidireccional: tiene una captura de sonido de 360° en el plano horizontal.

3. Su sensibilidad es constante: no hace filtrado de sonidos dependiendo de su amplitud. Aunque esto es de interés para la percepción completa del ambiente, esto provoca la inconveniente captura de interferencias y ruido.

4. Requiere de potencia adicional, conocida como *Phantom Power*, para entregar una señal limpia y sin ruido de línea eléctrica. El *Phantom Power* es una forma de proporcionar corriente continua a los dispositivos de audio para alimentar el circuito interno de los micrófonos. De manera más formal, los micrófonos necesitan amplificar la señal generada a través de un pre-amplificador de audio que se encuentran en la parte interna del gabinete del micrófono.

2.2.2. Interfaz de Audio

La interfaz de audio es la encargada de realizar la conexión de audio entre la computadora y los dispositivos de captura, como instrumentos musicales y micrófonos [20]. Dependiendo de la interfaz de audio, varía la cantidad de salidas y entradas disponibles

Figura 2.3: Micrófono de superficie SHURE MX391.

para poder realizar la conexión de los dispositivos. La conexión de la interfaz de audio a la computadora puede ser mediante USB.

La interfaz utilizado es una M-Audio Fast Track Ultra, presentada en la Figura 2.4.

Figura 2.4: Interfaz de audio marca M-Audio, modelo Fast Track Ultra [10].

Las características de esta interfaz [10] son:

1. Se le pueden conectar hasta cuatro micrófonos que requieran *Phantom Power*, pero requiere de una fuente externa de voltaje para darle energía a más de dos micrófonos.
2. Provee un amplificador por cada micrófono.
3. Se conecta a la computadora por USB.

2.3. Conceptos de sonido digital

Como ya se ha mencionado, aunque diferentes dispositivos en conjunto logran la captura y conversión del sonido, es la interfaz de audio la que realiza la conexión entre el sonido en el mundo físico y la computadora. Hace esto convirtiendo la señal en un formato digital.

Por medio de un sistema de conversión analógico-digital el sonido es convertido a una representación digital mediante la codificación de la señal analógica en una secuencia numérica [7]. Se puede remplazar la señal original por una secuencia de números binarios que serán interpretados por una computadora. Estos números binarios representaran la forma de onda de sonido.

La fidelidad del sonido se determina mediante la resolución y la frecuencia de muestreo. En el caso de esté proyecto se maneja una frecuencia de muestreo de 48000 Hz y una resolución de 16 bits.

2.3.1. Frecuencia de muestreo

La frecuencia de muestreo dictamina la cantidad de muestras de amplitud de sonido que se capturan en cada segundo. Es calculada de la siguiente manera:

$$f_m = \frac{1}{p} \tag{2.3}$$

donde f_m es la frecuencia de muestreo y p es el periodo de muestreo.

2.3.2. Resolución

Toda información que se almacena en una computadora requiere de una cantidad de L bits para almacenar cada muestra. Dicho de otra manera, se requiere de un número de bits para la representación de la amplitud de cada muestra capturada. Este número de bits define la resolución de dicha representación.

2.4. Formato de sonido/audio utilizado

Existe una gran variedad de formatos de audio digital que son utilizados para guardar archivos de audio. Cada uno es guardado con su correspondiente extensión especificada en el nombre del archivo. Diferentes formatos tienen diferentes objetivos por los cuales fueron diseñados como: altos factores de comprensión o no comprometer la calidad del sonido. Cada formato tienen sus ventajas y desventajas, y la decisión de cuál utilizar en este proyecto fue determinado por la necesidad de una alta calidad (frecuencia de muestreo y resolución) sin importar el espacio que ocupan los archivos de audio en la memoria de la computadora.

En el proyecto se utiliza el formato WAV. Éste almacena el sonido sin compresión lo cual resulta en necesitar una gran cantidad de espacio de memoria para almacenamiento. Se ocupa este formato de audio cuando no se quiere tener pérdida en la calidad del sonido por compresión, por lo que es mejor para el propósito de este proyecto.

2.5. JACK Audio Connection Toolkit

JACK Audio Connection Toolkit es una biblioteca multi-plataforma que permite compartir dispositivos entre módulos de procesamiento, así como crear conexiones entre éstos. Adicionalmente, presume de poder llevar a cabo dicho procesamiento en tiempo

real [19]. “Tiempo real” en este contexto implica ser tan rápido como un humano habla y se escucha a sí mismo. JACK es una de las bibliotecas más populares para captura de audio en tiempo real, utilizado en software como: Audacity, AvLinux, Swami, etc.

Esta biblioteca provee el servidor de JACK, el cual se conecta a la interfaz de audio la cual, a su vez, tiene conectado los dipositivos de entrada de audio así como de salida de audio. Adicionalmente, esta biblioteca provee la facilidad de crear agentes que se conectan al servidor de JACK para tener acceso a los dispositivos de audio, así como a las salidas de otros agentes de JACK. En la Figura 2.5 se puede observar de una manera gráfica esta filosofía.

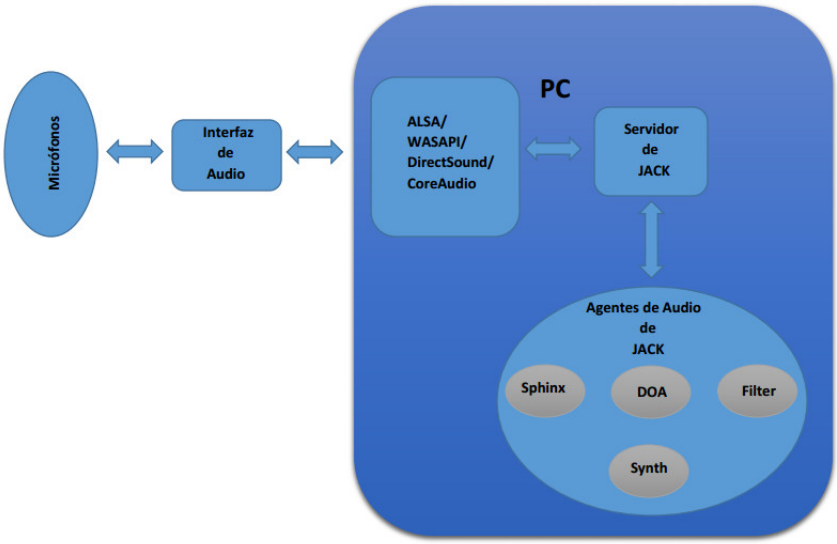

Figura 2.5: Esquema de Servidor de Sonido.

Los agentes de JACK se comunican por medio de la escritura y lectura de valores de energía en arreglos de datos. Mientras los arreglos de datos sean escritos o leídos adecuadamente, el servidor de JACK se encarga de entregar los datos en tiempo real a los dipositivos de audio así como al resto de los agentes.

2.6. Bases de Láser

En este proyecto, los datos de localización de la fuente sonora son recavados mediante un láser. Éste no queda exento de señales contaminantes, que debido a su funcionalidad algunas veces son provocadas por el entorno que lo rodea: partículas de polvo en el aire, objetos no deseados que obstaculicen el paso del láser, objetos personales (vestimenta) de los usuarios. En las siguientes secciones explicaremos esto de manera más detallada.

2.6.1. Hardware (Hokuyo UTM-30LX/LN)

Al querer detectar a una persona, se pudiera pensar una gran variedad de dispositivos que están creados para cumplir con esa funcionalidad. En el caso de Golem-II+, se utilizó el láser que ya trae implementado, el cual es de marca Hokuyo, modelo UTM-30LX/LN [12]. Se puede observar en la Figura 2.6. Cuenta con diversas características a las cuales se fue necesario adaptar para poder cumplir con el objetivo de localizar a las fuentes sonoras.

Figura 2.6: Láser marca Hokuyo, modelo UTM-30LX/LN.

2.6.2. Operaciones principales

El láser UTM-30LX/LN realiza un escaneo de un campo semicircular, midiendo la distancia de los objetos por medio de medir la energía con la que regresan sus emisiones direccionadas. El sensor de medición junto con el láser realizan la transferencia de datos mediante el canal *Laser Safety Class 1*.

Este sensor tiene dos modos de operación, dependiendo del tipo de salida:

- **UTM-30LX**: la señal de salida va a ser síncrona y se obtiene en cada exploración que realice el láser. Esté tipo de transmisión de datos es utilizada principalmente para aplicaciones robóticas, y este proyecto no es la excepción.
- **UTM-30LN**: genera señales de aviso cuando se encuentra algún objeto en el intervalo de visibilidad del láser y es utilizado principalmente en la aplicaciones de seguridad.

2.6.3. Posicionamiento y rango del láser

En la Figura 2.7 se muestra el intervalo de visibilidad que alcanza el láser UTM-30LX/LN, el cual es de 270°. Fue necesario limitar el radio de alcance del láser para la detección de usuarios, como se muestra en la Figura 3.13, Capítulo 3.

Adicionalmente, el láser está ubicado en la parte inferior del robot servicial Golem-II+, de manera centrada.

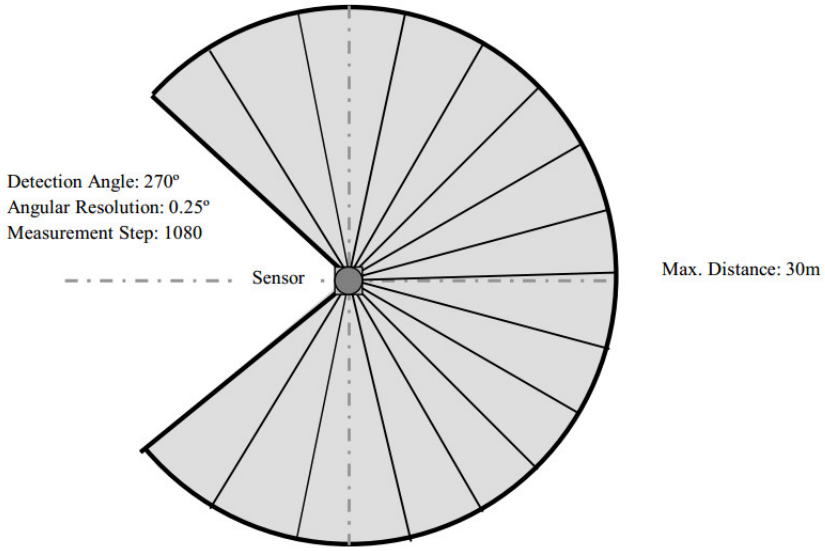

Figura 2.7: Intervalo de visibilidad del láser.

2.6.4. Restricciones

El láser UTM-LX/LN proyecta un haz de luz que al ser interceptado por algún objeto, presenta la distancia o la presencia de dicho objeto. Claro está, si el objeto es transparente, el haz de luz no detectará a dicho objeto [12]. Ésta es una de las principales restricciones.

Adicionalmente, el intervalo de visibilidad que maneja el láser limita el intervalo en donde se puede detectar usuarios. Es decir, no se puede cubrir los 360° alrededor del Golem-II+. Y, por la ubicación en la parte inferior del robot servicial, si algunos de los usuarios brincara o se saliera del intervalo de visibilidad asignado para esté proyecto, los datos que nos proporcionaría el láser serian erróneos.

Otra restricción es que por detectar a los usuarios a la altura de las pantorrillas, existe un espacio entre una pierna y la otra. Ya que el el haz de luz sigue su curso a lo largo del plano horizontal, si detecta dos piernas, estaría detectando a dos usuarios, con lo que se obtendrían resultados erróneos. La solución a este problema es crear un mecanismo que se coloca en las piernas de cada usuario para evitar que el láser logre pasar por en medio de las mismas. En las Figuras 3.14 y 3.15, Capítulo 3, se muestra los esquemas de este complemento.

2.7. Player y Stage

Para poder controlar o establecer una conexión con los dispositivos de Golem-II+ (como el láser) se necesita un software capaz de interpretar lenguaje de código que va aunado con el funcionamiento de dispositivos o sensores. Es analógico en el humano cuando el cerebro le ordena al resto de los órganos.

Player [22] es un servidor que controla y obtiene la información proporcionada por dispositivos o sensores del robot. De esta manera hace de cada parte del hardware tenga un funcionamiento individual. En este programa los clientes de Player son los algoritmos que controlan al robot.

Este software contiene librerías que realizan las funciones necesarias, enviando mensajes que controlan al robot siguiendo el protocolo de comunicación de Player. Estas librerías son compatibles con diversos lenguajes de programación como C++, Java, Python o Lisp.

Stage en conjunto con Player funciona como un simulador en 2D de robots móviles, facilitando el trabajo si no se cuenta con un robot físico o para hacer pruebas de algoritmos antes de llevarlo a cabo sobre el robot físico. Una de las ventajas de trabajar con Player y Stage es que son de código abierto por lo que se pueden modificar a la conveniencia del proyecto.

2.7.1. Sockets TCP/IP

Los mensajes de Player se realizan a través de un socket TCP/IP, los cual permite el intercambio de datos entre dos programas que posiblemente se encuentran situados en dispositivos o computadoras diferentes [4]. Para poder realizar esta conexión es necesario un puerto que contienen un número de 16 bits, además de una dirección IP la cual es un número de 32 bits. Por decirlo de otra manera, es una clave única que tendrá cada una de las computadoras participantes, y facilita la conexión entre dispositivos para poder intercambiar una trama de datos. Para realizar la conexión entre los algoritmos y los dispositivos de Golem-II+ se utiliza el puerto 6665 utilizado por defecto por Player y la IP local de la PC interna de Golem-II+.

2.7.2. Instalación y Configuración del Servidor

Para realizar la instalación serán necesarios dos paquetes que deberán ser descargados: Player y Stage. Se distribuyen en forma de código fuente, lo cual significa que deben ser compilados. En el caso del proyecto, el desarrollo de los algoritmos fue llevado a cabo en el sistema operativo Linux, versión Ubuntu.

La instalación de Player normalmente se realiza en el directorio:

```
/usr/local
```

Esto implica que se debe contar con permiso de súper usuario.

Para compilar a Stage es necesario seguir los pasos típicos de compilación de software en Linux:

```
./configure
make
make install
```

Mayores detalles de la compilación no son proveídos aquí por ser irrelevantes, pero pueden ser consultados en la documentación de Player/Stage [6].

2.7.2.1. Librerías

Para que un módulo de Player acceda a las funcionalidades del servidor Player y la simulación de Stage, es necesario hacer uso de las librerías *libplayer.so* y *libstage.a* que se encuentran en la dirección:

```
/usr/local/lib
```

De tal forma es necesario poner este directorio en la variable de entorno PLAYERPATH, por medio del siguiente comando:

```
export PLAYERPATH= /usr/local/lib
```

2.7.2.2. Levantar el servidor Player y simulador de Stage

Player es el encargado de controlar la comunicación entre dispositivos del robot. Para ejecutarlo, es necesario pasarle un fichero de configuración con extensión *cfg*, donde se describe la configuración de los dispositivos. Dicha configuración es única para cada robot, por lo que se omite dicha información en este documento. Es importante mencionar que uno de los dispositivos que se puede configurar en dicho fichero es Stage, que a su vez simula otros dispositivos.

Para levantar el servidor es necesario teclear:

```
player archivo.cfg
```

Si se realizan correctamente los pasos de la instalación se pondrá en marcha Player, y es así como se podrá observar una ventana con una simulación en 2D gracias a Stage. Si se requiere que el servidor Player se conecte con dispositivos reales, se le tiene que pasar un fichero de configuración *.cfg* apropiado. En el caso de Golem-II+, dicho fichero fue creado antes de comenzar a llevar a cabo el proyecto, el cual configura el acceso al láser UTM-30LX/LN.

2.7.3. Visualización del Láser con Player Stage

Para poder visualizar los datos que está arrojando nuestro láser es necesario utilizar PlayerViewer, el cual se puede levantar con el siguiente comando:

```
playerv
```

Es necesario entrar en el menú *Devices* para localizar los diferentes dispositivos que contiene nuestro robot. Para poder interactuar con este dispositivo, PlayerViewer se debe suscribir a éste. En nuestro caso, se debe suscribir al dispositivo láser. Acto seguido, se comienza a hacer un barrido del espacio que se encuentra frente al robot servicial, devolviendo la distancia que hay entre el láser y los objetos en su intervalo visual.

2.7.4. Programación con Player

Existen librerías para distintos lenguajes (C++, Python, Java, etc.) que contienen el protocolo de comunicación de Player, proveyendo una capa de transparencia para el manejo de envío y recepción de datos.

Las librerías están desarrolladas siguiendo un modelo de programación basado en *proxies*. Estos son los objetos locales que actúan como intermediarios para que el cliente tenga un manejo de los dispositivos del robot, u obtener información del mismo. Esto se realiza llamando a los métodos de los objetos a través de los comandos dentro de las librerías. Es necesario mencionar que se puede obtener más información sobre todas las clases de proxies existentes, junto con sus métodos, en la documentación de Player [6].

2.7.4.1. Código ejemplo para conexión al servidor

En el caso de Golem-II+, la conexión se establece a través de la clase PlayerClient, mediante la dirección IP 10.10.10.3 que pertenece al computadora interna de Golem-II+, y al puerto 6665, como se muestra en el codigo 2.1:

Listing 2.1: Código para iniciar cliente Player.

```
client = playerc_client_create(NULL, "10.10.10.3", 6665);
if (0 != playerc_client_connect(client))
  return -1;
```

Se crea un proxy para la información de posición 2D del robot, para después suscribirse al dispositivo que se quiera controlar, en este caso: los motores del robot. Obsérvese el código 2.2:

Listing 2.2: Código para controlar motores del robot.

```
position2d = playerc_position2d_create(client, 0);
```

```
if (playerc_position2d_subscribe(position2d,
  PLAYER_OPEN_MODE))
      return -1;

if (0 != playerc_position2d_set_cmd_vel(position2d,
  1, 0, DTOR(0.0), 1))
      return -1;
```

En este código se indica que los motores de Golem-II+ deberán estar apagados en todo momento, ya que es necesario que el robot servicial se encuentre estático para realizar las pruebas.

2.7.4.2. Código ejemplo de conexión con el láser

Con el código 2.3 se crea un proxy para el láser.

Listing 2.3: Proxy para láser.

```
device = playerc_laser_create(client, 0);
```

Ahora es necesario suscribirse al proxy creado. Obsérvese el código 2.4:

Listing 2.4: Inscripción al proxy.

```
if (playerc_laser_subscribe(device, 1) != 0)
      return -1;
```

Después, se puede mandar a llamar este mismo proxy y poder realizar la comunicación con el láser. Es necesario conocer la posición y la orientación del láser mediante el código 2.5, para escribir los resultados en el proxy.

Listing 2.5: Posición y Orientación del láser.

```
if (playerc_laser_get_geom(device) == 0)
    printf("laser geom: [%6.3f %6.3f %6.3f] [%6.3f %6.3f]\n",
    device->pose[0], device->pose[1], device->pose[2],
    device->size[0], device->size[1]);
```

2.7.4.3. Código ejemplo de desconexión del servidor

Para desconectarnos de Player, es necesario desconectarse primero de los proxies y luego desconectar al cliente del servidor Player. Obsérvese el código 2.6

Listing 2.6: Desconexión del láser y del cliente.

```
playerc_position2d_unsubscribe(position2d);
```

```
    playerc_position2d_destroy(position2d);
    playerc_laser_unsubscribe(device);
    playerc_laser_destroy(device);

    playerc_client_disconnect(client);
    playerc_client_destroy(client);

    return 0;
```

2.8. Corpus para Audición Robótica

Son muy pocos los corpus de Audición Robótica que existen y cada uno tiene sus propias características. El conocerlos nos ayuda a poner en contexto el proyecto. Estos corpus comparten algunas características semejantes en cuanto a los dispositivos electrónicos que son utilizados, pero existe una diferencia en la cantidad de estos dispositivos y los modelos utilizados, además de los algoritmos que implementan cada corpus.

2.8.1. AV16.3 (An Audio-Visual Corpus for Speaker Localization and Tracking)

Es un corpus de datos audiovisuales [9], los cuales son recaudados en un sala de reuniones. Se enfoca en la colocación de cámaras y micrófonos calibrados de tal manera que se tiene un arreglo de datos en tres dimensiones. De esta manera, se pueden hacer pruebas con la localización de audio mediante algoritmos de Audición Robótica.

- Presencia hasta de tres fuentes a la vez.
- Presencia de fuentes móviles.
- Se recaudo sólo información auditiva y visual, no de texto.
- Se utilizaron tres cámaras para obtener la localización de los usuarios.
- Se utilizaron 16 micrófonos en una oficina para grabarlos: dos arreglos circulares de 8 micrófonos cada uno.

2.8.2. DIRHA_AEC (A Multi.Channel Corpus for Distant-Speech Interaction in Presence of Known Interferences)

Esté corpus [21] se enfoca en el reconocimiento de voz para el control de servicios y dispositivos dentro de un departamento, mediante la separación de fuentes de sonido

como es la del habla y técnicas para la separación de ruido.

- Presencia hasta de tres fuentes a las vez.
- No hubo fuentes móviles.
- Se recaudo información auditiva y textual, pero no la posición del usuario.
- Se utilizaron micrófonos alrededor del cuarto: tres pares en las paredes, un arreglo de tres en la cuarta pared, y otro arreglo de seis en el techo.
- Los micrófonos que utilizaron fueron de marca SHURE MX- 391 (mismos que se utilizaron en este proyecto).

2.8.3. AIRA

Este es el nombre del corpus que fue implementado para la realización de este proyecto [17]. La recaudación de la información se explicará en el capítulo de la metodología. Existen grabaciones previas en este corpus que fueron realizadas con Golem-II+ y la presencia de fuentes estáticas.

- Presencia hasta cuatro fuentes a la vez.
- Presencia de fuentes móviles, pero sin una posición exacta.
- Se utilizaron tres micrófonos puestos arriba de un robot de servicio.
- Se recaudo información auditiva, textual y de posición.

La aportación que se presenta en este proyecto es que dichas fuentes se encuentran en movimiento, estableciendo una nueva manera de ubicación de los usuarios y volviendo más completa la información utilizado para la evaluación de algoritmos de Audición Robótica.

Capítulo 3

Metodología

3.1. Audio

Al ser éste un proyecto que está enfocado en complementar un recurso de evaluación de algoritmos de ubicación de fuentes sonoras, el principal punto a cubrir fue el audio, utilizando los micrófonos que se ubican en la parte superior del robot de servicio Golem-II+.

3.1.1. Conexiones de los micrófonos de superficie SHURE MX391

La conexión de los micrófonos toma un papel importante dentro del proyecto, a pesar de que el proyecto pareciera que está enfocado a la detección física de las fuentes. Es necesario conocer la dirección en que se ubican así como el sonido que éstas producen, y esto se puede lograr obteniendo la mayor cantidad de información que se pueda recopilar.

Los micrófonos SHURE MX391 son conectados mediante un cable XLR al canal de la interfaz de audio. Es necesario activar el *Phantom Power* como se muestra en la Figura 3.1 para que los micrófonos funcionen.

3.1.1.1. Instalación a los canales correspondientes a la interfaz de audio

Los micrófonos SHURE MX391 cuentan con cápsulas de condensador intercambiables para una gran variedad de aplicaciones. Estas cápsulas funcionan como un convertidor para poder conectar el cable XLR del micrófono a los canales de la interfaz de audio. En la Figura 3.2 se puede observar la conexión al canal 1 de la interfaz de audio.

Figura 3.1: Interruptor del *Phantom Power*.

Figura 3.2: Conexión a canal 1.

3.1.2. Topología de los micrófonos de superficie SHURE MX391 ya establecida

Cabe mencionar que la posición de estos micrófonos de superficie fue establecida por el grupo Golem como se muestra en la Figura 3.3. Los micrófonos están situados de tal manera que forman un triangulo equilátero y al ser omnidireccionales logran una captura de sonido de 360° por lo que el alcance de registrar los sonidos que se encuentran en nuestro entorno es completo en el plano horizontal.

Figura 3.3: Topología de los micrófonos de superficie.

3.1.3. Funcionamiento de la interfaz M-Audio

La interfaz de audio es la encargada de convertir las señales de analógicas a digitales. Se utilizan tres de los canales de la interfaz para poder conectar los micrófonos y mediante un cable USB se conectó la interfaz a la computadora como se muestra en la siguiente Figura 3.4.

Figura 3.4: Conexión de cable USB a interfaz de audio.

Una vez que los cables de los micrófonos de superficie, se conectan a los canales de la interfaz de audio. Se debe realizar la siguiente serie de pasos para poder lograr que

la interfaz de audio funcione correctamente:

- Verificar que el switch se encuentra posicionado en el voltaje de 48 V conocido como *Phantom Power*, como es mostrado en la Figura 3.1.
- Conectar el eliminador a la interfaz de audio y al enchufe de la corriente, como es mostrado en las Figuras 3.5 y 3.6.

Figura 3.5: Conexión del eliminador a la interfaz de audio.

Figura 3.6: Conexión de eliminador a corriente.

- Activar los canales frontales de la interfaz de audio mediante el botón que se muestra en la Figura 3.7.

Figura 3.7: Botón para activación del canal frontal.

Una manera de verificar que nuestros micrófonos están conectados correctamente es saturándolos por medio de hacer un ligero movimiento parecido al rascar sobre el capuchón del micrófono. Si el micrófono está conectado correctamente, el LED arriba del conector se convierte de color verde a color rojo al hacer esto.

Por último, es necesario que la perilla de la amplificación se encuentre ubicada en 0 dB para no saturar innecesariamente la interfaz y, a la vez, mantener un buen nivel de intensidad en las grabaciones.

3.1.3.1. Configurar opciones de M- Audio

Se necesita hacer una serie de configuraciones dentro de la computadora para poder lograr la conexión con la interfaz y así mismo con el servidor de JACK, ya que cada computadora funciona de una manera diferente.

3.1.3.2. Configuración de software Jack Audio

Para realizar la configuración del servidor de JACK es necesario abrir la interfaz QJAckCtl, mostrada en la Figura 3.8. Esta interfaz viene como parte del paquete de instalación de JACK.

En esta ventana, se debe oprimir el botón que lleva por nombre “Setup...”, lo cual abrirá la ventana de configuración de JACK, mostrada en la Figura 3.9:

Se necesitan seleccionar las siguientes casillas:

– Tiempo real: Esta opción activa la posibilidad de correr el servidor de JACK en tiempo real, lo cual es importante para mantener la integridad de los datos de

Figura 3.8: Ventana del software QJAckCtl.

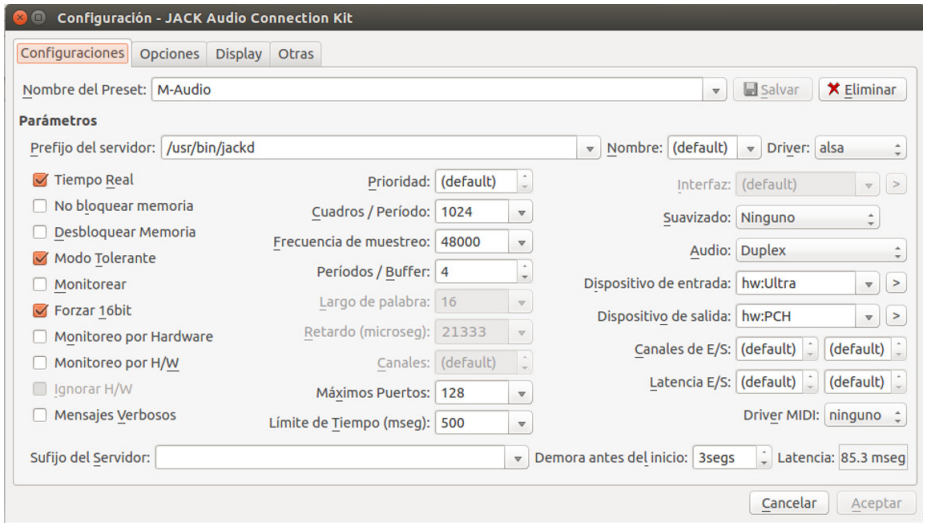

Figura 3.9: Ventana de configuración de software JACK Audio.

audio que están capturados y que posteriormente van a ser utilizados para simular el ambiente que se capturó.

– Frecuencia de muestreo: Es la frecuencia a la cual se están capturando los datos de audio. Este valor se decidió anteriormente en el Grupo Golem para obtener una alta calidad de los datos capturados.

– Periodos/Buffer: Es el tamaño del buffer interior del servidor de JACK para mantener las conexiones entre los agentes de JACK vivas. Este valor, si es grande, da una mayor flexibilidad temporal a los agentes pero presenta desfases en el procesamiento; si es pequeño, el desfase se minimiza, pero los agentes lentos perderán datos. El valor presentado la Figura 3.9 otorga un buen balance entre estos datos.

3.2. Posición

3.2.1. Colocación de Golem-II+ para registro de fuentes sonoras

La posición del robot de servicio Golem-II+ es de suma importancia para que los datos no presenten errores a la hora de capturarlos. Dicha ubicación fue dentro del laboratorio del Departamento de Ciencias de la Computación del IIMAS-UNAM. El robot fue ubicado dentro del laboratorio, como es mostrado en la Figura 3.10:

Figura 3.10: Ubicación del robot servicial Golem-II+.

Se hicieron pequeñas marcas en el piso del aula como se muestra en la Figura 3.11 en rojo. Estas marcas sirven como referencia en el momento de la captura y se redujera el margen de error entre una prueba y la otra. Asímismo, si el robot servicial Golem-II+ era cambiado de la posición, estas marcas facilitaron el re-ubicarlo a su posición original.

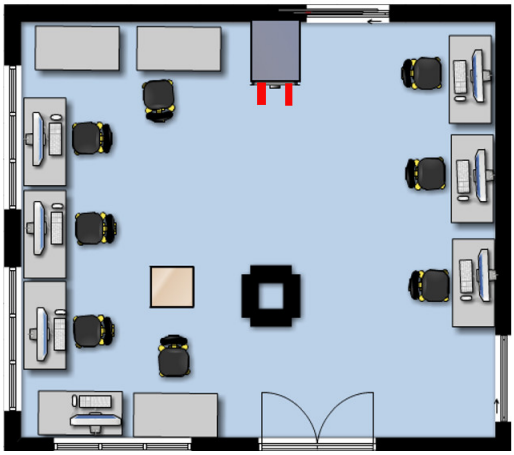

Figura 3.11: Marcas para ubicación del robot servicial Golem-II+.

Esta ubicación fue decidida en base a los restricciones impuestas por el láser instalado en Golem-II+.

3.2.2. Intervalo angular para detección de fuentes sonoras

El intervalo del láser fue reducido a un intervalo de -90 a 90 grados, ya que las demás direcciones son bloqueadas por el cuerpo del robot. Se marcó media circunferencia como se muestra en la Figura 3.12, el cual es el intervalo establecido para que los usuarios caminen.

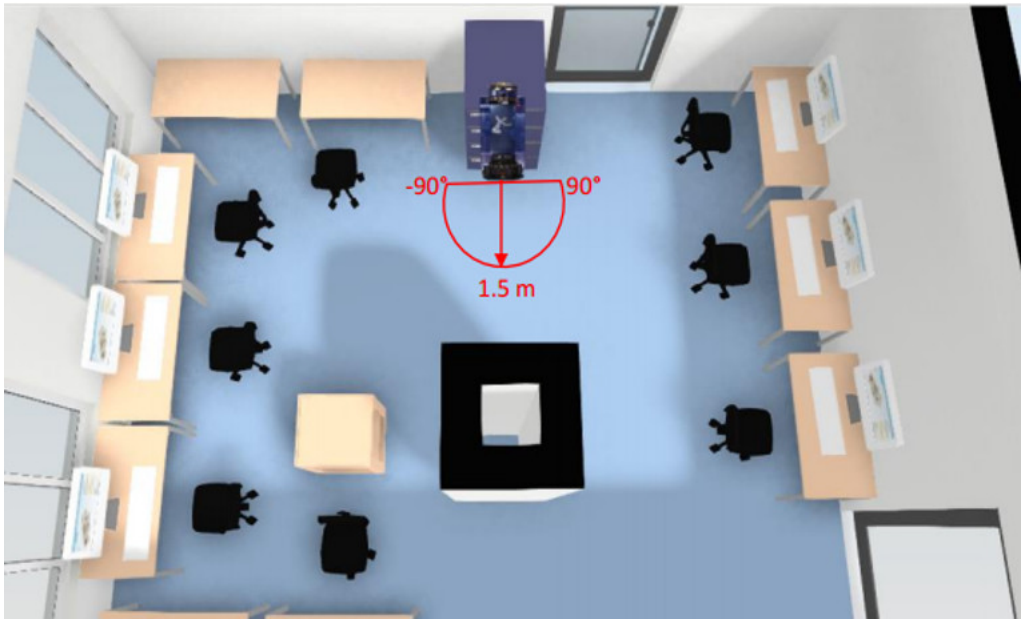

Figura 3.12: Rango donde caminaran usuarios.

3.2.3. Distancia máxima para registro de fuentes sonoras

Es necesario limitar el alcance en distancia del láser para poder tomar esta medida durante el desarrollo del proyecto, y marcarla en el espacio de prueba. Se marcó desde el punto central del robot servicial Golem-II+ donde se encuentra el láser hasta 1.5 m como radio de un semi-círculo. Obsérvese la Figura 3.13.

3.2.4. Programación para detección de personas

Una vez marcadas y registradas los intervalos en ángulo y distancias, la programación para la detección de fuentes se llevó a cabo y descrito aquí. Se desarrolló en el lenguaje C ya que ambos Player y JACK lo soportan.

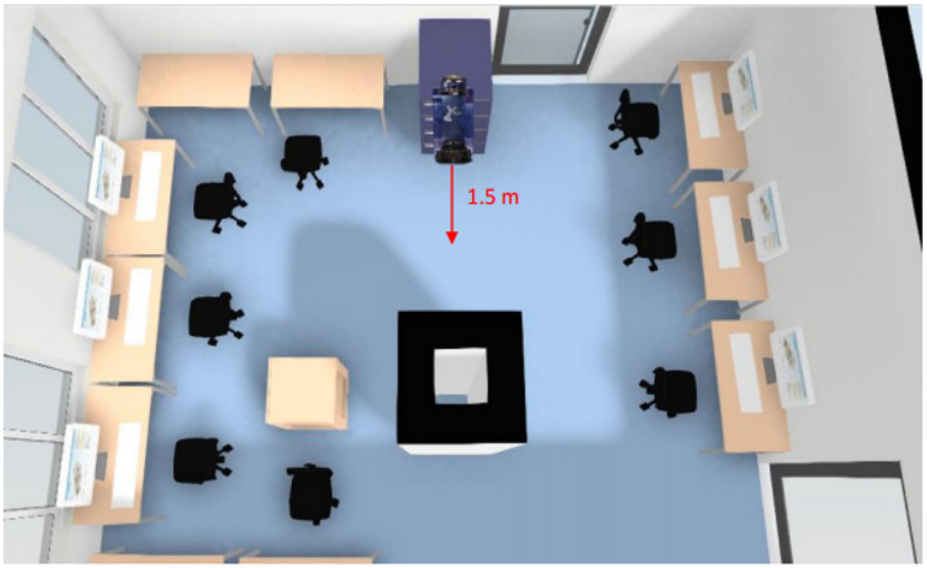

Figura 3.13: Alcance de Láser.

En el código 3.1 se muestra la función "posicion", donde se lleva a cabo la localización de los usuarios por medio del uso del láser. A lo largo de esta sección se describirán los diferentes componentes de esta función.

Listing 3.1: Código para localizar usuarios por medio de láser.

```
std::vector<Loc> posicion(float a[], float g[], int n){
    std::vector<Loc> Personas;
    int i,contador=0,bandera=0;
    float maxDistancia=1.5,promGra=0,sumaGra=0;
    for(i=n-1; i>=0; i--){
        if(a[i]<maxDistancia){
            bandera=1;
            sumaGra=sumaGra+g[i];
            contador=contador+1;
        }else if(bandera==1){
            promGra=sumaGra/contador;
            sumaGra = 0; contador = 0; bandera = 0;
            Loc Persona;
            Persona.posicion=(-1.0)*promGra;
            Personas.push_back(Persona);
        }
    }
    return Personas;
}
```

El algoritmo hace un recorrido del arreglo que contiene todos los datos capturados mediante el láser. Se asume que el usuario trae un aditamento en las piernas y que está dentro del semicírculo que tiene un radio igual al valor de la variable "maxDistancia" a la cual se le ha asignado 1.5. El algoritmo recorre los arreglos de los disparos del láser, buscando aquellos que sean menores a "maxDistancia". Cabe mencionar que el valor de 1.5 es el que se asignó previamente como el radio de semi-círculo que limita el área por donde podrán hacer su recorrido nuestros usuarios.

En el código 3.2 se muestran las operaciones que se realizan siempre y cuando cumpla con la condicional de "a[i]¡maxDistancia", donde el arreglo "a" contiene las distancias de los disparos del láser. Se asigna una variable con el nombre de "sumaGra" en la que se guardan la suma de todos los valores angulares (contenidos en el arreglo "g") que tengan una distancia menor a "maxDistancia".

Listing 3.2: Operaciones del programa cuando se cumple la condicional.

```
bandera=1;
sumaGra=sumaGra+g[i];
contador=contador+1;
```

La variable "contador" se incrementa para que al final se sepa la cantidad de disparos de láser que se ubican dentro del sub-arreglo que corresponde al usuario, y así poder sacar el promedio de los valores angulares de éste. Se guarda en la variable "promGra", la cual es dividida entre "contador" para este efecto. Este promedio es el punto medio de cada objeto que se atraviese frente al haz de luz del láser, así logramos tener su ubicación.

Es necesario que a cada una de las variables donde se guardaron previamente los datos las igualemos a cero como se muestra en el código 3.3, esto con la finalidad de reinicializar cada una de ellas para posteriormente grabar nuevos datos y con esto nuestros resultados no se vean erróneos.

Listing 3.3: Código para resetear variables.

```
sumaGra = 0; contador = 0; bandera = 0;
```

En el código 3.4 se crea el objeto "Persona", tipo "Loc" que contiene la variable "posicion" en donde se guarda el valor promedio angular de la persona localizada. Debe notarse que en la variable "Persona.posicion" se realiza una multiplicación por -1.0 con la finalidad de cambiar la orientación de los grados que nos proporcionan los datos del láser. Esto se realizó ya que al leer el arreglo de datos se hizo de izquierda a derecha y los ángulos están asignados por el láser en orden contrario.

El método "Personas.push_back(Persona);" añade un elemento al final del vector "Personas" con lo que se guardan la posición de todas las personas localizadas.

Listing 3.4: Código de Loc Persona para guardar información.

```
Loc Persona;
Persona.posicion=(-1.0)*promGra;
Personas.push_back(Persona);
```

Así, al terminar el recorrido de los disparos del láser, el vector "Personas" es formado por la localización de todas las personas localizadas en este momento en el tiempo. La información de dicho vector es después añadido a un archivo de texto que es compatible con el resto del corpus.

3.3. Texto

Los textos utilizados para ser leídos por los usuarios fueron obtenidos de aquellos utilizados por el corpus DIMEx100 Pineda et al. [16]. Estos enunciados fueron seleccionados por tener un nivel bajo de perplejidad. Es decir, por tener una variación importante entre diferentes unidades léxicas (fonemas y alofonemas) que sean continuas entre sí mismas. Esto resulta en tener una gran variedad de sonidos del español mexicano, lo cual resulta en una representación balanceada de dicho lenguaje.

3.4. Sensor

3.4.1. Funcionamiento del láser UTM-30LX/LN

El intervalo angular del láser UTM-30LX/LN es de 270° y cuenta con un rango de medida de hasta 30 metros, por lo que su alcance es bastante amplio. Se pueden tomar una cantidad de 1080 disparos en un tiempo de 25ms con una alta velocidad de rotación del motor del láser, 2400rpm. Gracias a estas características se forma una especie de abanico el cual es de gran utilidad para la detección de fuentes sonoras del robot de servicio Golem-II+.

3.4.2. Programación para establecer conexión con el código de detección de personas

En la sección 2.7.4.1 se hace mención de cada una de las partes que realizan la conexión del código detección de personas con el robot de servicio Golem-II+.

3.4.3. Mecanismo para evitar escape del haz de luz del láser

Una importante limitante es que el haz de luz del láser se puede filtrar por en medio de las piernas de los usuarios, provocando con esto que el programa identificara a una

sola persona como si fueran dos (una persona por cada pierna). Para evitar, esto se utilizó un mecanismo de madera donde se hizo una especie de cajón como se observa en las Figuras 3.14 y 3.15 con sus medidas correspondientes:

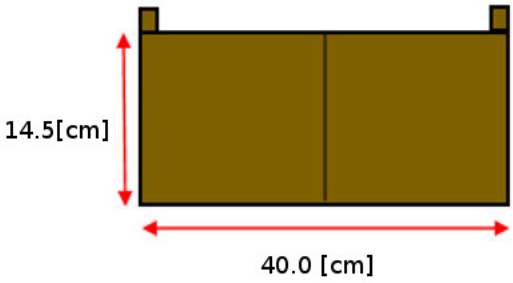

Figura 3.14: Medidas de mecanismo cerrado.

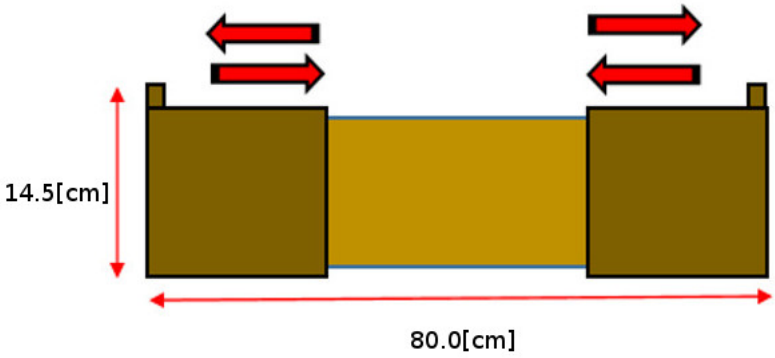

Figura 3.15: Medidas de mecanismo abierto.

Cabe mencionar que el mecanismo fue creado con madera para reducir costos, pero es un buen boceto para en un futuro hacerlo con un material más resistente y ligero como aluminio.

3.5. Corpus de Golem-II+

El corpus *Acoustic Interactions for Robot Audition* (AIRA), el cual se quiere complementar en este proyecto, fue presentado como un trabajo del Grupo Golem. Este corpus se centra en la fuente de sonidos Dirección-de-Arribo (DOA), Separación de Fuentes de Voz y Reconocimiento de Voz en los siguientes escenarios:

1. "Cámara Anecoica", con casi ninguna presencia de ruido y reverberación, con hasta cuatro fuentes de voz simultáneas en posiciones conocidas fijas.

2. “Oficina A”, con presencia de ruido y reverberación moderada, con el mismo tipo de fuentes de habla como en la cámara anecoica.

3. “Oficina B”, con una mayor presencia de ruido y reverberación, con hasta dos fuentes móviles de voz.

4. “Pasillo”, con una estación móvil de grabación que añade ruidos de movimiento (principalmente procedente de los motores del robot), además de la cantidad moderada de ruido ambiental y reverberación, y una combinación de fuentes de voz fijas y móviles.

La estación de captura de audio consta de tres micrófonos en forma de triángulo equilátero, que se encuentra aproximadamente a la misma altura que las fuentes del habla. Las fuentes de voz fijas eran bocinas de respuesta de frecuencia plana (monitores de estudio de grabación) colocadas en diferentes direcciones y a la misma distancia de la estación de captura, con diversas distancias angulares entre ellos. Las fuentes móviles de voz eran hablantes humanos que se mueven a un ritmo aproximadamente constante alrededor de la estación de captura, siempre manteniendo la misma distancia a la estación de registro. En el caso del escenario “Pasillo”, la estación de captura se montó sobre el robot de servicio Golem-II+ que se mueve en una línea recta. Esto significa que, desde el punto de vista del robot, una fuente fija de voz parece estar en movimiento, y, debido a que las fuentes de discurso en movimiento en este escenario están acompañando el robot, éstas parecen estar fijas.

Todas las fuentes de voz se dirigían a la estación de registro mediante la lectura de frases del corpus DIMEx100 Pineda et al. [16], que está en español, con una separación pequeña o ninguna entre las frases, en intervalos de 30 segundos. Teniendo en cuenta todas las variaciones de escenarios posibles (las diferentes posiciones de las fuentes, el silencio entre las frases, etc.), el corpus consta de 49 diferentes configuraciones de evaluación. En cada configuración, se capturaron diez grabaciones de 30 segundos, para proporcionar varios intentos para cada configuración de evaluación. Cada intervalo de 30 segundos proporcionó tres archivos WAV estándar, uno para cada micrófono, grabados a 48 kHz, y van acompañados de un archivo generado automáticamente que contiene la dirección de arribo real de cada fuente de voz, así como las transcripciones ortográficas de lo que se dice por cada fuente de voz. El corpus tiene alrededor de 735 minutos de grabaciones de voz (245 minutos de voz del usuario, multiplicado por 3 micrófonos), por un total de alrededor de 4 GB de espacio.

Integrado al corpus se proporciona un marco de trabajo que utiliza las grabaciones para simular el entorno en el que se grabaron. Esta simulación podría alimentarse a los sistemas de Audición Robótica que estiman múltiples direcciones de arribo y/o hacer separación de fuentes de voz en tiempo real, para efectos de su evaluación. Se propuso una evaluación de la estimación de dirección de arribo basado en ventanas y en la tasa de error de palabra para el reconocimiento de voz de las fuentes del habla separadas. Es importante tener en cuenta que cada escenario proporciona un conjunto diferente de desafíos con dificultad creciente, por lo que estas evaluaciones pueden proporcionar

puntos de referencia de la capacidad de audición del robot, así como señalar sus áreas de oportunidad.

Las grabaciones capturadas en este proyecto se integraron a este sistema como el escenario "Oficina C", siendo compatibles con su árbol de directorios, el cual es:

```
Multi-DOA Corpus
Office C Recordings
    1 Source
        90_to_-90
            Evaluation_1
                goldstandard.mdoa
                speech.txt
                wav_mic1.wav
                wav_mic2.wav
                wav_mic3.wav
            Evaluation_2
                goldstandard.mdoa
                speech.txt
                wav_mic1.wav
                wav_mic2.wav
                wav_mic3.wav
```

Capítulo 4

Evaluación

4.1. Encendido del robot de servicio Golem-II+.

4.1.1. Conexión del robot de servicio Golem-II+.

1. El robot de servicio Golem-II+ cuenta con un cable de red que debe ser conectado a nuestra PC para poder realizar la comunicación con la computadora interna. Obsérvese la Figura 4.1.

Figura 4.1: Cable de red para conectar PC a Golem.

2. Se conectan los micrófonos y la interfaz de audio como se describe en las secciones 3.1.1, 3.1.2 y 3.1.3.

4.1.2. Encendido del robot de servicio Golem-II+.

Para encender el robot de servicio Golem-II+ es necesario seguir los siguientes pasos:

1. Encender al robot de servicio Golem-II+ activando el interruptor de encendido que se encuentra en la parte de abajo del robot.
2. Los botones AUX1 y AUX2 deben estar oprimidos, presentados en la Figura 4.2

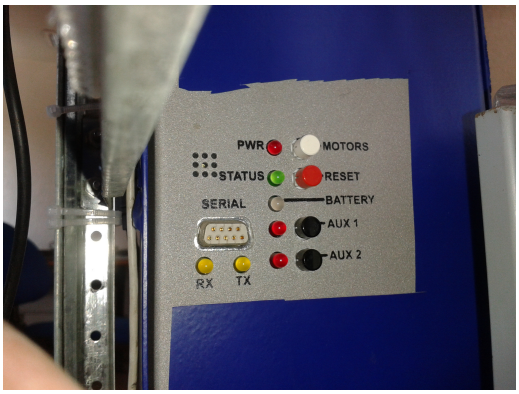

Figura 4.2: Interruptor de encendido y botones auxiliares.

3. Encender la computadora interna de Golem-II+ activando el interruptor de encendido que se encuentra en la parte media del robot, presentado en la Figura 4.3.

Figura 4.3: Interruptor de la computadora principal.

4. Para comprobar la conexión de la computadora interna con la computadora de desarrollo: abrir una terminal y hacer “ping 10.10.10.3”. Una vez que responda la computadora interna, se continúa con los siguientes pasos.

4.1.3. Comandos para conectarse a servidor Player

1. Abrir una terminal remota y correr: "ssh golem@10.10.10.3". Con esto el usuario accede de forma remota al sistema Linux.
2. Corre el comando "player nd_robot.cfg" para levantar el servidor de Player y así controlar los dispositivos del robot y obtener la información del láser.
3. Correr los algoritmos desarrollados que se conectan al servidor Player. Hecho esto, se puede comenzar con la captura de datos.

4.2. Escenarios Capturados

Se capturaron tres escenarios de evaluación, clasificados de la siguiente manera:

1 Fuente. El usuario realizó movimientos de 90º a 0º dando lectura a su correspondiente texto. El usuario repitió este movimiento 5 veces. Para cada repetición, se guardó la información posición del usuario en un archivo llamado "goldstandard.mdoa". La información guardada se puede observar en la Figura 4.4.

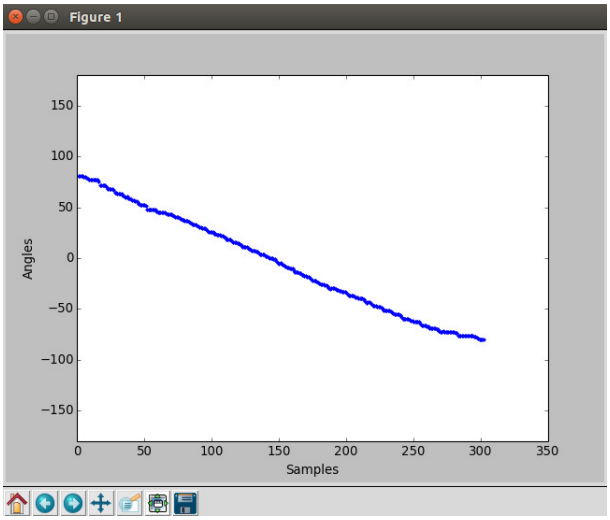

Figura 4.4: Gráfica con una fuente

2 Fuentes. La primera fuente realizó una trayectoria de 90º a 0º y la segunda fuente se traslado de 0º a -90º, dando lectura a su correspondiente texto. Los usuarios repitieron este movimiento 10 veces. Para cada repetición, se guardó la información posición del usuario en un archivo llamado "goldstandard.mdoa". La información guardada se puede observar en la Figura 4.5.

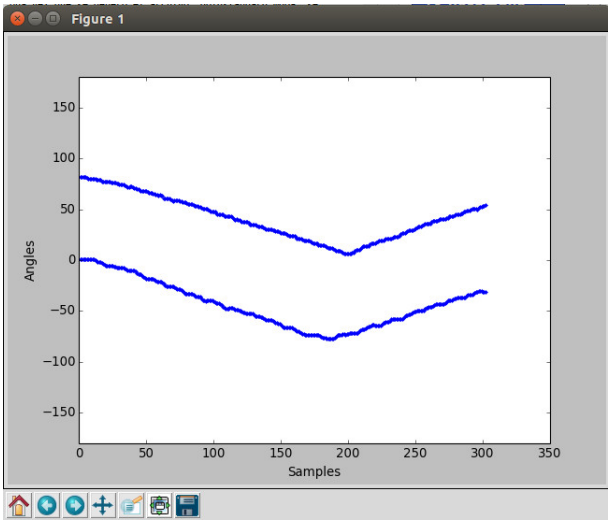

Figura 4.5: Gráfica con dos fuentes

3 Fuentes. Una de las fuentes se mantuvo estática en 0º, mientras que las dos fuentes restantes realizaban movimientos simultáneos. Una fuente de 90º a 0º y la otra de -90º a 0º, cada una de ellas haciendo lectura de su correspondiente texto.Los usuarios repitieron este movimiento 10 veces. Para cada repetición, se guardó la información posición del usuario en un archivo llamado "goldstandard.mdoa". La información guardada se puede observar en la Figura 4.6.

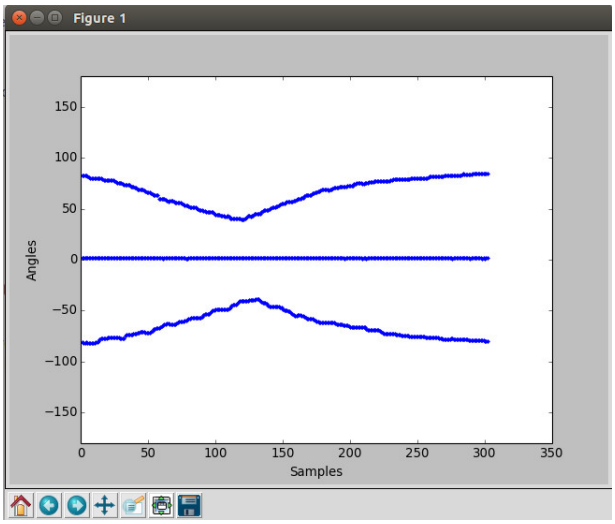

Figura 4.6: Gráfica con tres fuentes

4.3. Evaluación de Algoritmo y Resultados Esperados

Se utilizó la información recaudada para evaluar el algoritmo de localización de fuentes sonoras de Golem-II+ [17]. El sistema de evaluación desarrollado por el Grupo Golem utiliza estos datos para calcular la métrica de evaluación llamada $F1$, la cual consiste en ser una medida de exactitud considerando la *precisión* (P) y la *recuperación* (R).

La precisión es el número de resultados correctos positivos dividido entre el número total de resultados. La recuperación se obtiene dividiendo el número de resultados correctos positivos entre la suma del número de resultados correctos positivos y del número de resultados incorrectos negativos. Para obtener la métrica $F1$ se utiliza la siguiente ecuación:

$$F_1 = 2(\frac{P * R}{P + R}) \tag{4.1}$$

La métrica $F1$ se puede considerar como el promedio de la precisión y la recuperación. En esta métrica mientras más alto sea el resultado mejor.

Se corrió el sistema de localización de Golem-II+, con su configuración por defecto y se obtuvieron los resultados mostrados en la Tabla 4.1.

Fuentes	Precisión	Recall	F1score
1	31.1 %	21.3 %	25.3 %
2	75.8 %	45.4 %	56.8 %
3	87.3 %	41.4 %	56.1 %

Tabla 4.1: Resultados de la evaluación del localizador de fuentes del Grupo Golem.

Se puede observar que el desempeño con una fuente se encuentra en un 25 % lo cual es un valor muy bajo. En cuanto a las demás fuentes los valores se encuentran más altos pero no cumplen con los datos que se reportaron en el sistema de evaluación en el artículo original [17]. Por tal motivo la evaluación no estaba siendo congruente con la realidad, es posible que esto se deba a que el grupo Golem estaba utilizando estimaciones de la posición del usuario.

Dados estos resultados, el supervisor de audio del grupo Golem, el Dr. Caleb Antonio Rascón Estebané, recalibró el algoritmo por medio de hacer cambios a la configuración del algoritmo para que fuera más tolerante a un ambiente reverberativo y fuera más responsivo. Esta versión re-calibrada del algoritmo obtuvo los resultados mostrados en la Tabla 4.2.

Como se puede observar, el algoritmo re-calibrado subió su desempeño considerablemente con una fuente. También obtuvo incrementos de desempeño observables en

los otros dos escenarios.

Fuentes	Precisión	Recall	F1score
1	42.5 %	51.9 %	46.7 %
2	71.9 %	52.4 %	60.6 %
3	89.3 %	44.6 %	59.4 %

Tabla 4.2: Resultados de la evaluación del localizador de fuentes del Grupo Golem después de la re-calibración.

Gracias al complemento del corpus AIRA desarrollado en esta tesina, se sabe con exactitud donde está el usuario y la trayectoria que está realizando. Esto logra una evaluación más cercana a la realidad, lo cual ayudó a incrementar el desempeño del sistema de localización de fuentes sonoras del grupo Golem. Aún así, se ha identificado que el sistema de Golem-II+ requiere mejoras para ambientes reales y se dio solución a la problemática de localización de fuentes móviles.

Capítulo 5

Conclusión

El corpus *Acoustic Interactions for Robot Audition* (AIRA) contiene todas las aportaciones en Audición Robótica que el grupo Golem realiza en el robot de servicio. De igual manera las actualizaciones a sus versiones son de suma importancia para lograr que el robot se mantenga siempre un paso adelante. Dar solución a la problemática que presenta el corpus en la actualidad, proveyendo una medición exacta (no solo una estimación) de la ubicación de cada una de las fuentes sonoras móviles, es de gran ayuda para las futuras generaciones que deseen realizar investigaciones en este tipo de proyectos.

La solución al problema antes descrito necesitó de un código de programación que lograra cumplir con los objetivos para el seguimiento y la localización de fuentes sonoras, haciendo uso de los dispositivos electrónicos con los que contaba el robot de servicio Golem-II+. De igual manera, se necesitó hacer uso de los códigos de programación que ya se encontraban dentro del corpus para establecer las conexiones necesarias. Los resultados obtenidos mediante de este proyecto cumplen y dan solución a la problemática de una evaluación más cercana a la realidad, lo cual ayudó a lograr un incremento en el desempeño del sistema e identificando que es necesaria una mejora para ambientes reales.

Es de gran satisfacción participar en un proyecto que forma parte de un grupo enfocado a la investigación y desarrollo tecnológico. De esta manera se logra un interés por la investigación aplicando los conocimiento adquiridos en la carrera de Ingeniería. Este proyecto es la base para dar paso a un trabajo a futuro de mejorar el algoritmo de seguimiento con láser, consiguiendo una solución a la problemática que se presenta al cruzarse dos personas sobre un mismo punto o lograr detectar a alguna persona que se encuentre saltando frente al robot de servicio Golem-II+. Parte de estas problemáticas pueden tener solución utilizando nuevas tecnologías o dispositivos electrónicos más avanzados como el Kinect el cual es parte de la infraestructura del robot de servicio Golem-II+. Otra forma de solucionar estas problemáticas es creando un sistema que se encargue de grabar por separado a las personas que se encuentren frente a éste por medio de vídeo, y utilizando micrófonos inalámbricos para grabar el audio de cada usuario por separado.

Saber que parte de esta tesina se encuentre publicada en el diario “The Journal of the Acoustical Society of America”, es de gran satisfacción y motivación, pero compartir créditos junto a un equipo de investigadores de grupo Golem es algo que jamás hubiera imaginado. [18]

Capítulo 6

Glosario

- Corpus: Conjunto estructurado de textos, archivos, muestras orales.
- Laser: Light Amplified by Stimulated Emission of Radiation (Luz amplificada por emisión estimulada de radiación), es un dispositivo que emite un haz de luz muy intenso generado mediante la estimulación eléctrica o térmica de átomos, moléculas o iones.
- Robot: Sistema eléctrico/mecánico programable capaz de realizar funciones de manera autómata.
- Software: Es la parte intangible que conforma la computadora, ejemplo de ello programas y sistemas operativos.
- Hardware: Son los componentes físicos que conforman a la computadora, ejemplo de ello monitor, teclado, mouse, etc.
- Inteligencia Artificial (IA): Es una rama de las ciencias computacionales, donde la inteligencia se aplica a maquinas que intentan asemejarse a la mente humana.
- Algoritmo: Serie de instrucciones que llevan a la solución de un problema.
- Señal Analógica: Es una señal continua con variaciones a lo largo del tiempo, estas señales representan magnitudes físicas que toman todos los valores posibles en un intervalo.
- Señal Digital: Es una señal con variaciones discontinuas en el tiempo y toman ciertos valores discretos
- Atenuación: Es la perdida de potencia de una señal.
- Distorsión: Es la alteración de una señal provocada por la naturaleza del algoritmo o filtro, que modifican la forma de la señal.
- Variaciones analógicas de corriente eléctrica (duda)

- Números Binarios: Sistema numérico representado por unos y ceros utilizado en sistemas computacionales, es una señal eléctrica que representa dos estados: encendido o apagado.
- Cable XLR: Estándar de cableado para aplicaciones de sonido que involucra tres componentes: voltaje, referencia y tierra.
- Micrófonos Omnidireccionales: micrófonos que no atenúan la señal capturada dependiendo de su dirección de arribo.

Bibliografía

[1] Albertí, E. B. (2006). *Procesado digital de señales: Fundamentos para comunicaciones y control - II*. Edicions UPC. 6

[2] Ashour, G., Brackenridge, B., Tirosh, O., Todd, C., Zimmermann, R., and Knapen, G. (1998). *Universal Serial Bus Device Class Definition for Audio Devices*. Audio Devices. http://www.usb.org/developers/docs/devclass_docs/audio10.pdf. 8

[3] Boquera, M.-C. E. (2003). *Servicios avanzados de telecomunicación*. Madrid : Díaz de Santos, D.L. 2003. 5

[4] Comer, D. E. (1996). *Redes globales de información con Internet y TCP/IP*. Pearson. 15

[5] Cromer, A. H. (1986). *Física en la ciencia y en la industria*. Reverte. 6

[6] Gerkey, B. (2011). Player documentation. http://playerstage.sourceforge.net/index.php?src=doc. 16, 17

[7] Haykin, V. V. (1986). *Señales y Sistemas*. Limusa Wiley. 8, 10

[8] Hyde, J., Regué, J., and Cuspinera, A. (1997). *Control electroneumático y electrónico*. Barcelona Norgren Marcombo D.L. 1997. 6

[9] Lathoud, G., Odobez, J.-M., and Gatica-Perez., D. (2004). Av16.3: an audio-visual corpus for speaker localization and tracking. In *Proceedings of the MLMI'04 Workshop*. 19

[10] M-Audio (2007). *M-Audio Fast Track Ultra English User Guide*. Avid Technology, Inc. Sitio Espejo: http://www.strumentimusicali.net/manuali/MAUDIO_FTULTRA_ENG.pdf. XIII, 10

[11] Miyarač, F. (2000). ¿ruido o señal? la otra información. en defensa del registro digital del ruido urbano. Laboratorio de Acústica y Electroacústica; Facultad de Ciencias Exactas, Ingeniería y Agrimensura; Universidad Nacional de Rosario (UNR). 7

[12] Mori, Kamitani, Kamon, and Hino (2012). *Scanning Laser Range Finder UTM-30LX/LN*. Hokuyo Automatic Co., LTD. http://www.hokuyo-aut.jp/02sensor/07scanner/download/pdf/UTM-30LX_spec_en.pdf. 13, 14

[13] Moyanoč, J. M. D. (2005). Ruidos e interferencias: Técnicas de reducción. Dpto. de Electrónica y Computadores, Santander, 2005. 7

[14] Nashelsky, L. (1993). *Fundamentos de tecnología digital*. Limusa. 7

[15] Olarteč, E. R. A. D. (2005). *Estudio y análisis de las fuentes de interferencia y ruido en el radar atmosférico de Piura*. PhD thesis, Universidad de Piura, Facultad de Ingeniería. 7

[16] Pineda, L. A., Castellanos, H., Cuétara, J., Galescu, L., Juárez, J., Llisterri, J., Pérez, P., and Villaseñor, L. (2010). The Corpus DIMEx100: Transcription and Evaluation. *Language Resources and Evaluation*, 44:347–370. 31, 33

[17] Rascon, C., Fuentes, G., and Meza, I. (2015). Lightweight multi-doa tracking of mobile speech sources. *EURASIP Journal on Audio, Speech, and Music Processing 2015*, 11(2):1–15. 20, 39

[18] Rascon, C., Meza, I., Millán, A., Vélez, I., and Fuentes, G. (2018). Acoustic interactions for robot audition: A corpus of real auditory scenes. *The Journal of the Acoustical Society of America, 144 (5). ISSN 0001-4966. doi: 10.1121/1.5078769.*, 144(12):1–5. VII, 2, 3, 42

[19] Rascón, C. (2015). Material de curso "audición robótica" del posgrado de ciencias e ingeniería de la computación de la universidad nacional autónoma de méxico: "instalación de jack". http://calebrascon.info/AR/Topic3/03_JACK.pdf. 12

[20] Vicente Bresó Flores, M. J. I. O. (2007). *Trabajar con sonido digital en un PC*. PUV. 9

[21] Zwyssig, E., Ravanelli, M., Svaizer, P., and Omologo., M. (2015). A multi-channel corpus for distant-speech interaction in presence of known interferences. In *Proceedings of ICASSP 2015*. 19

[22] Álvarez, J. A. (2014). *Desarrollo de un controlador en Player/Stage para el seguimiento de trayectorias de robots móviles*. PhD thesis, Universidad de Alcalá Escuela Politécnica Superior. 15