DTMF Audio Communication for NAO Robots

Kyle Poore, Joseph Masterjohn, Andreas Seekircher, Pedro Peña, Ubbo Visser
University of Miami
Department of Computer Science
1365 Memorial Drive, Coral Gables, FL, 33146, USA
{kyle—joe—aseek—pedro—visser}@cs.miami.edu

Abstract

We propose an alternative to Wi-Fi for robotic communication, as its increased use in a competition environment has lead to highly overlapping and interfering networks. This interference often causes unreliable transmission of data, which affects teams' ability to coordinate complex behaviors. Our method uses fixed length Dual Tone Multi Frequency (DTMF) messages and uses a basic packet structure designed to reduce data corruption as a result of noise. We conducted twelve different experiments varying the distance between robots and message format, as well as whether the robots are walking or sitting silently. The results show that while this method appears to be sensitive to room reverberation and multipath effects, it has very low data corruption rates, which makes it suitable for use in some applications.

1 Introduction

For most autonomous robotics tasks involving multiple robots, some level of communication is crucial for coordinating complex actions to achieve a desired goal. A popular method of communication between robots (especially in RoboCup) is Wi-Fi. With hundreds of robots communicating on several independent overlapping Wi-Fi networks, interference is rampant, often causing network delays of several seconds, and in the worst cases, dropping entire connections altogether. It is therefore desirable to have an alternative method of communication, which, while possibly inferior to a strong Wi-Fi connection, is useful in situations where Wi-Fi has become unusable. Of course, the usefulness of a alternative communication method extends beyond just Wi-Fi. Any system which might encounter a catastrophic level of interference could benefit from having a backup communication scheme.

There are a few potential candidates to consider when looking for ways to transmit data from one robot to another. Since we are working on a standard platform, we cannot modify or enhance our robots' hardware to accommodate alternative communication equipment, and are limited to using the robots' existing hardware for transmitting and receiving messages. One possibility is to use visual communication, controlling LEDs on the robot to send data through the visible or infrared spectrum, or perhaps even using the robot's physical movements to encode data in some way. A major caveat to visual communication is that it is line-of-sight only; if one robot moves in front of another or turns so that its LEDs are no longer visible, the channel is interrupted. Another problem with visual communication schemes is that they would require complex vision algorithms for detecting signals, which are often not suitable when processing power is limited. Another, more favorable possibility for an alternative communication system is to encode data as audio, and broadcast it via the robot's loudspeakers. This does not suffer from the same problems as visual communication because sound waves travel around obstacles, and analyzing audio samples can be done efficiently. Our research shows that the Dual-Tone Multi-Frequency (DTMF) method, while unreliable in certain conditions, can be used to broadcast messages with low probability of message corruption.

Much research has been devoted to audio signals featuring humanoid robots, especially in the past decade. Audio signals can be important sensor information as they can be used for various purposes, whether for the communication between multiple robots, the detection of audio cues in the environment or game events such as whistles, or using the audio signals to improve self-localization. A demonstration within the RoboCup Standard Platform League (SPL) in 2013 in Eindhoven by the team RoboEireann revealed how difficult it is to communicate between NAOs, humanoid robots engineered by Aldebaran (SoftBank) robotics, on the soccer field in a noisy environment.

The paper is organized as follows: we discuss relevant work in the next section and describe our approach in Section 3. Our experimental set-up and the conducted robot tests is explained in Section 4. We discuss the pros and cons of our results in Section 5, and conclude and outline future work in the remaining Section 6.

2 Related Work

When consulting the literature, one finds a number of research papers that relate to our work. We include work that is not only related to communication, but also work that develops audio processing techniques for sensing the environment, since communication and sensing are inherently related. Saxena and Ng (Saxena and Ng 2009) present a learn-

ing approach for the problem of estimating the incident angle of a sound using just one microphone not connected to a mobile robot. The experimental results show that their approach is able to accurately localize a wide range of sounds, such as human speech, dog barking, or a waterfall. Most existing research investigates sound-source localization using static microphone arrays, particularly for the purpose of navigation. These methods are used by Sun et al. for a audio-based robot navigation system for a rescue robot. It is developed using tetrahedral microphone array to guide a robot finding the target shouting for help in a rescue scenario (Sun et al. 2011). The approach uses speech recognition technology and a Time DOA (TDOA) method. The authors claim that the system meets the desired outcome.

Another recent application of audio based communication has been developed by Sauer et al. (Sauer, Dickel, and Lotter 2014) for the purpose of facilitating control and adjustment of hearing aids by using high frequency audio signals sent from a smart phone. Their technique involves fully redundant transmission by sending the same control signals across multiple different frequencies, so that in the case of environmental noise masking one frequency, there is a higher chance that the control codes can still be recovered by the earpiece.

ASIMO, the remarkable humanoid developed by HONDA also uses the auditory system for its tasks. An early paper from 2002 introduces the use of a commercial speech recognition and synthesis system on ASIMO. The authors state that the audio quality and intonation of voice need more work and that they are not yet satisfactory for use on the robot (Sakagami et al. 2002). Okuno et al. (Okuno, Nakadai, and Kim 2011) present a later version of ASIMO's ability to use the auditory system for tasks at hand. They use the HARK open-source robot audition software (Nakadai et al. 2010) and made experiments with speech and music. The authors claim that the active audition improves the localization of the robot with regard to the periphery.

Carrara et al. (Carrara and Adams 2014) have shown that audio communication between machines is possible and practical for covert transmission of data in an office environment between computers in a manner which is imperceptible to humans. By using frequencies just above the human hearing range of about 20kHz - 20.5kHz they were able to transmit data at a rate of 140 bps, and were able to achieve 6.7 kbps when using audible frequencies between 500Hz and 18kHz.

Latest research such as the paper by Jayagopi et al. (Jayagopi et al. 2013) suggest that significant background noise presented in a real HRI setting makes auditory tasks challenging. The authors introduced a conversational HRI dataset with a real-behaving robot inducing interactive behavior with and between humans. The paper however does not discuss the auditory methods used in detail. We assume that the authors use the standard auditory recognition that comes with the NAO.

All mentioned approaches and techniques so far differ from our approach (a) in the method used, (b) in the application of the audio recognition, and (c) the RoboCanes framework, a robotics framework developed by the RoboCanes robotic soccer team at the University of Miami. Here, all audio modules have been implemented from scratch and run within the framework's system loop.

Nguyen and Bushnell (Nguyen and Bushnell 2004) have suggested that acoustic communication using DTMF is, in general not, recommended for mobile robot applications due to the unreliability in acoustical integrity of the signal during transmission. While their transmission methods are similar, there are key differences in the recognition methods used: the frequencies used in their experiments are the generic set of frequencies used in telecommunications (which lie in a range prone to environmental noise). We sought to experiment with different sets of frequencies, chosen for specific empirical reasons, and to overcome signal degradation through robust filters.

Other uses of DTMF technology for robotic communication have been explored apart from acoustical environments. Srivastava et al. (Srivastava et al. 2014) and Aswath et al. (Aswath et al. 2013) have used DTMF with mobile phones for long range control of robots. Each of these works uses a mobile phone directly connected to embedded hardware and the signal is transmitted through RF, not acoustically.

3 Approach

There are many methods for transferring data over analog media, but most are not suitable for communicating over the open air waves with a moving, noisy robot. Some of the challenges presented by this domain are relatively high noise levels (which can be unpredictable, especially in a robotics competition environment), interference from the internal vibrations of the robot (such as motors, fans, and stressed plastic), and unknown/changing distance between communicating robots. The latter of the above challenges means that using a Phase Shift Keying (PSK) method would likely perform poorly, since the data is embedded in the phase of the carrier wave. As a robot moves closer or further away from the sender, the distance – and thus the phase of a signal – will drift. This effect would be particularly prominent in systems which use high frequency carrier waves, as their wavelengths are very short and thus sensitive to small changes in distance. Frequency Shift Keying (FSK) handles the problem of the robots moving around since humanoid robots rarely reach speeds that would affect the frequencies via the Doppler effect, but they have a different weakness since single tones are modulated between two or more frequencies. Interference on one of those frequencies is likely to cause a data corruption error.

The Dual-Tone Multi-Frequency (DTMF) method uses eight different frequencies, divided equally into two groups: low tones and high tones. Symbols are transmitted by combining one frequency from the set of low frequency tones and one frequency from the set of high frequency tones using additive synthesis followed by a short period of silence and playing the resulting signal through a loudspeaker. The number of bits which can be transmitted through one symbol for a *generalized* DTMF scheme with a frequency groups, each with b frequencies is $log_2 b^a$. Since we use the typical two groups of four frequencies each, we can send $log_2 4^2 = 4$ bits at a time. Sending arbitrary bytes of data is convenient,

since eight bit bytes can be broken into four bit codes, which allow a direct mapping to the sixteen possible symbols of DTMF.

This method is less prone to errors than FSK in noisy environments because if the probability of random noise coinciding with a chosen frequency f is $P(f)$, then probability of random noise emulating two chosen frequencies f_1 and f_2 simultaneously is $P(f_1) \cdot P(f_2)$. This assumes that the probability of random noise producing each of the two frequencies is independent of each other. However, frequencies in open-air environments are often not independent; while many sounds have a fundamental frequency, they are often accompanied by harmonic frequencies. Usually, the higher the multiple of the fundamental frequency a harmonic is, the lower the amplitude of that harmonic. This means that the most highly correlated frequencies in any noisy environment tend to be ratios whose numerator and denominator are small integers; since either f is the fundamental frequency, or f is some harmonic f_n of some other fundamental frequency $f_1 = \frac{1}{n} f_n$, and is related to the other harmonics of f_1 by $f_k = \frac{k}{n} f_n$. Since the frequencies present in general noisy environments are not known beforehand, we cannot choose frequencies which avoid the harmonics present in the signals. We can, however choose frequencies whose ratios are not fractions with small a and b – that is – frequencies whose least common multiple are large. To achieve this, we divide our window length w_s by prime numbers p in the range of $\frac{1}{4} w_s \leq p \leq \frac{1}{2} w_s$ in order to determine the appropriate wavelengths for the low and high tones. The corresponding frequencies are calculated according to $f = s/\lambda$, where s is the speed of sound. This method of selecting frequencies ensures that they share only distant harmonics while making sure that there are at least two periods of each frequency per window.

3.1 DTMF Transmitter

The transmitter sends encoded data as DTMF symbols through open-air sound waves. The encoded data can easily be converted to playable waveforms by isolating four bit segments of the message (conveniently represented as a hexadecimal-digit) and then using Table 1 to determine the appropriate frequency pair for that symbol. The two frequencies are combined using additive synthesis. Each waveform is pre-computed and simply indexed by each four bit section of the message. The gain of each DTMF tone is gently faded at the end as to avoid discontinuities which result in undesirable pops and clicks in the resulting audio signal resulting from waveforms ending with non-zero samples.

3.2 DTMF Receiver

The receiver works very differently from the transmitter. This is mostly due to the fact that everything has to work in reverse: going from audio samples to decoded digital data. The process is illustrated in Fig 3. Recorded audio is passed in windows of samples through a series of Goertzel filters to isolate the desired frequency magnitudes. This is done continuously on the new recorded samples, which generates a stream of vectors of frequency magnitudes (see Fig. 1). This

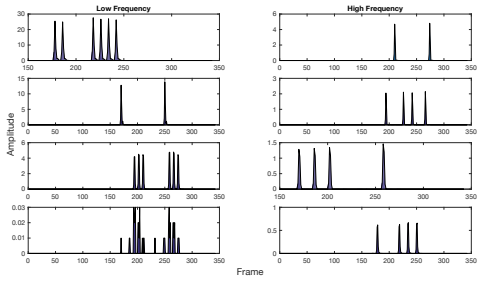

Figure 1: Magnitudes for individual frequencies

stream of Goertzel responses is then analyzed in two more steps. First, we need to identify sequences in the magnitudes that have enough similarity with a message. If a sequence has been identified as message, the symbols are decoded and the robot has received a message.

The first step is to capture audio data from the robot's microphones. We acquire our audio samples from the ALSA library, and use a 2,400 sample sliding window, advancing the window by half of its length after each analysis step. With a sample rate of 48,000, this means we are effectively performing 40 analysis steps per second of audio data. The window length was chosen such that it would not overlap two separate DTMF symbols. The part of the waveform corresponding to a symbol is referred to as the *mark*, the other part of the waveform is called the *space*. We have chosen the lengths of the mark and space to each have durations of 100ms, so both take up 4,800 samples. Thus, the window lies either completely in *mark*, completely in *space*, or some combination of the two, but will never span the gap and include samples from unrelated symbols. Since the window advances by half of its width after each step, we are guaranteed that at least one window will be completely filled with samples from the *mark*.

The next step is to perform frequency analysis on the windowed data. It would be sufficient to perform a Fast Fourier Transform (FFT) on the sample window, but there are only eight frequencies used in the messages. Therefore, it is more efficient to use multiple passes of the Goertzel algorithm (Goertzel 1958), one for each frequency. This leaves us with a vector of magnitudes for the eight frequencies.

We keep a history of the last several seconds of magnitudes and also keep a record of the sum of the magnitudes of each frequency over time. We use these frequency sums in order to detect the presence of a message before using

	h_1	h_2	h_3	h_4
l_1	0	1	2	3
l_2	4	5	6	7
l_3	8	9	A	B
l_4	C	D	E	F

Table 1: Frequency/Hexadecimal Encoding/Decoding Table l_i and h_i indicate the low and high frequency groups.

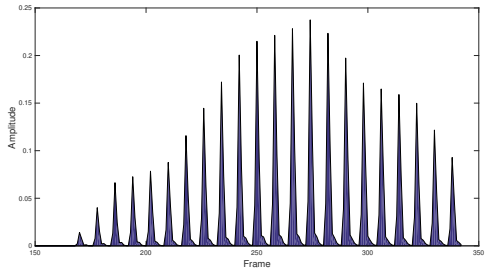

Figure 2: Peaks corresponding to output of the comb filter

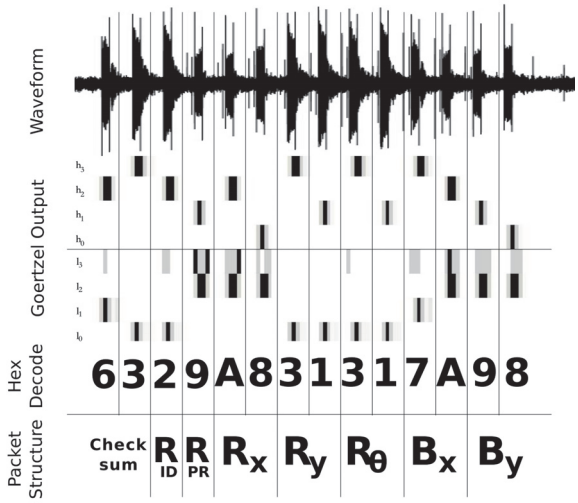

Figure 3: Illustration of the decoding process

the individual frequency magnitudes to decode the message. A comb filter with width equal to the separation between DTMF symbols and length equal to the number of symbols per message is used to detect when a message is heard. This is a filter that has the strongest response if the sum of the magnitudes of the eight frequencies are high in exactly the positions selected by the comb filter.

Since the comb filter accumulates the magnitudes of the tones until the end of the message, when the comb filter begins to discard older tones, the sums develop a triangular pattern consisting of distinctive peaks in ascending and descending magnitudes as in Fig. 2. The overall shape formed by the peaks can be evaluated for symmetry. If one peak exceeds a defined threshold and the surrounding peaks are close enough to the triangular shape, the message is accepted.

The position of the maximum peak is exactly the end position of the message. We can find all measured frequency magnitudes at the message *marks* at the corresponding positions in the buffer relative to the end of the message. The appropriate magnitude vectors can be revisited and evaluated for the maximum low and high frequencies to decode the message.

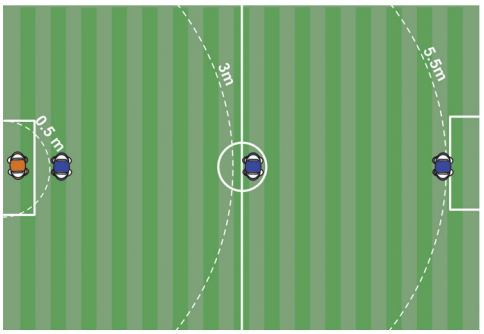

Figure 4: Robot placement

4 Experiments & Results

We have conducted our experiments to determine the effectiveness of DTMF as a communication method between robots that are different distances apart (Fig. 4), while also measuring the effect of internal noise while the receiving robot is walking. For each of the stationary experiments, 500 fixed length messages were sent, 7 bytes each. For the experiments involving a walking robot, only 200 messages were sent, due to limitations of battery and motor temperature. For each of the distances separating the robots and the different activities of the robots, two kinds of messages were tested. The first method uses random blocks of data that span the entire message length, and the second method uses a short header, including a checksum of the message, the transmitting robot's ID, and the ID of the robot from which the transmitter has last received, followed by random data.

For both block and packet messages, random data were generated by the transmitting robot and sent via the robot's loudspeakers. The same data were then saved to a file for later comparison. The receiving robot, upon receiving a block message, records it in a file, however, upon receiving a packet message, the packet is verified using its checksum. If the packet is valid, only the data portion of the packet is written to a file; otherwise the entire packet is discarded. After each test, the message files are copied from the robots for analysis. This experimental set-up is shown in Fig. 5. The analysis commonly performed on communication channels is the Hamming distance between the sent data and the received data. However, this metric is inappropriate in this case; it does not account for errors involving insertions or deletions, only substitutions. If the Hamming distance were to be used, the data would become misaligned after the first message drop, resulting in incorrect error rates. Besides that, the Hamming distance is not defined for data of different length. For these reasons we use the Levenshtein distance (edit distance). This metric allows for measuring error due to the insertion, deletion, and substitution of symbols, as it counts the minimum number of such edits to transform one string into the other.

For the first round of experiments, both robots were kept inactive to prevent the internal noise of motors from interfering with the signals. The transmitting robot was placed in the keeper's position on the field (between the goal posts). The receiving robot was then placed 0.5 meters (on the penalty

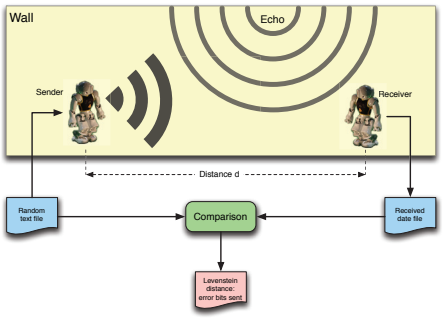

Figure 5: Experimental setup

box) in front of the transmitting robot, so that the two robots were facing each other. In this position, 500 fixed length randomized messages were sent from the transmitting robot to the receiving robot twice; once for the random block method and once for the random packet method (note that for the random packet method, the header is not randomized). The above procedure was then repeated for distances of 3 meters (midfield) and 5.5 meters (opponent's penalty box) as shown in figure 4. The results of these tests can be seen in table 2.

For the second round of experiments, in order to simulate game-like conditions, the receiving robot was made to walk in such a way that it's average position remains at distance d from the transmitting robot, while all other variables were kept the same as in the above experiments. These results can be seen in Table 3.

We had the opportunity to test our communication method in a live game at RoboCup 2016 in Leipzig, Germany to verify that it could indeed handle the challenges that a noisy competition environment poses for audio recognition. We collected the transmitted and received messages from three robots, and calculated the transmission accuracy for each pair by comparing the number of correctly recieved messages to the number of messages which were sent by each robot. Table 4 shows the ratio of correctly received messages per message sent between each pair of robots, where T_i and R_i are the transmitting and receiving robots.

5 Discussion

While the DTMF communication method seems to work relatively well between short range, quiet robots, the performance deteriorates drastically as the distance between the robots is increased. That said, we should expect that the error rates would be proportional to the inverse-square of the

Set	Distance	Mode	Bits Sent	Error Bits	Error Rate
1	0.5	block	28,000	2,392	0.0854
2	0.5	packet	20,000	880	0.0440
3	3.0	block	28,392	5,948	0.2094
4	3.0	packet	20,000	10,200	0.5100
5	5.5	block	28,000	5,759	0.2056
6	5.5	packet	20,000	2,338	0.1169

Table 2: Results for Silent Robots

Set	Distance	Mode	Bits Sent	Error Bits	Error Rate
7	0.5	block	11,200	6,440	0.5750
8	0.5	packet	8,080	4,680	0.5792
9	3.0	block	11,200	7,056	0.6300
10	3.0	packet	8,000	5,080	0.6350
11	5.5	block	11,200	6,552	0.5850
12	5.5	packet	8,000	4,400	0.5500

Table 3: Results for Walking Robots

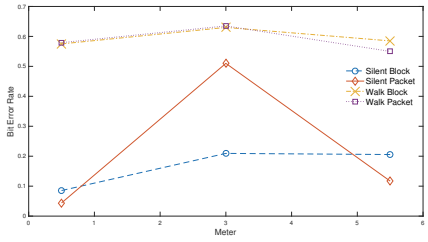

Figure 6: Bits of error per bits sent vs. transmission distance

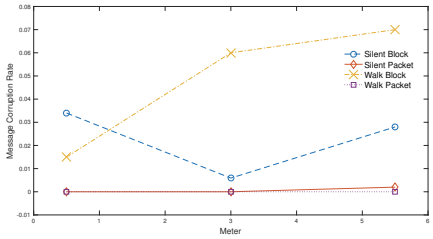

Figure 7: Corrupted messages per message sent vs. transmission distance

distance between the robots. Our best explanation for this is room reverberation. The experiments were performed in a room with hard, parallel walls, resulting in a significant echo. As these reverberations propagate and reflect about the room, the waves undergo constructive and destructive interference, contributing to the non-uniformity of the error curve.

Another factor which seemed to adversely affect the communication performance was the robot's walking. The actuation of motors within the body of the robot contribute greatly to the internal noise level of the robot, much of which is inaudible to an observer. Other physical sources of noise on the robot stem from the creaking of the plastic covers as they deform as a result of movement and stress. These sources of noise are far from random, and it is certainly possible that

	R_1	R_2	R_3
T_1	-	0.425	0.297
T_2	0.145	-	0.473
T_3	0.370	0.352	-

Table 4: Transmission Rates from a Live RoboCup Game

these internal noises interfere with one or more of the chosen DTMF frequencies. One possible way to improve this method is to automate the selection of frequencies for a particular environment with the robot actively moving, in order to choose the frequencies with least interference.

In Fig. 6, the bit error rate is relatively high for all but the short range, silent transmission experiments, but this can be misleading. Particularly in the domain of robotic soccer, message accuracy takes priority over reliable transmission. In fact it is for this reason that we do not bother to retransmit missed messages; it is simply better to wait for the next message from that robot. Most of the error bits are bits that have been dropped due to either failure to recognize a message, or due to packet rejection because the checksum failed. As shown in Fig. 7, all of the received messages contained no bit corruption errors. We consider a message corrupted if as little as one bit of that message has been flipped.

Some of our previous work has shown excellent recognition of whistle signals, which used a logistic regression classifier on the result of an FFT, however, this method is not suitable for this application because it is too computationally expensive. For this system to work properly, the receiver must be able to demodulate the signal just as fast as the transmitter can generate it. Furthermore, the mentioned approach is less appropriate for detecting/recognizing DTMF tones because it takes advantage of the presence of the many harmonic components of whistle-like signals, whereas for DTMF we specifically choose frequencies to mitigate harmonics for the reduction of interference between frequencies.

6 Conclusion

We have presented an approach for audio based broadcast communication between robots, using different states of activity and different message styles over multiple distances. The approach is based on fixed length DTMF messages. The results show that for short ranges, and robots with low activity levels, the method works well; however, with increased separation or activity levels, the message reliability rapidly deteriorates. Automatic frequency configuration may improve these results. Message corruption rates seem to stay relatively low, especially for the packet mode of operation, where the corruption rate is virtually zero. It is suspected that in the case of increased distance, the performance decline is more closely related to room reverberation; in a larger room, or outside, the method is expected to perform better.

References

Aswath, S.; Tilak, C. K.; Sengar, A.; and Udupa, G. 2013. Design and development of mobile operated control system for humanoid robot. *Advances in Computing* 3(3):50–56.

Carrara, B., and Adams, C. 2014. On acoustic covert channels between air-gapped systems. In *International Symposium on Foundations and Practice of Security*, 3–16. Springer.

Goertzel, G. 1958. An algorithm for the evaluation of finite trigonometric series. *The American Mathematical Monthly* 65(1):34–35.

Jayagopi, D. B.; Sheiki, S.; Klotz, D.; Wienke, J.; Odobez, J.-M.; Wrede, S.; Khalidov, V.; Nyugen, L.; Wrede, B.; and Gatica-Perez, D. 2013. The vernissage corpus: A conversational human-robot-interaction dataset. In *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, 149–150. IEEE Press.

Nakadai, K.; Takahashi, T.; Okuno, H. G.; Nakajima, H.; Hasegawa, Y.; and Tsujino, H. 2010. Design and implementation of robot audition system'hark'—open source software for listening to three simultaneous speakers. *Advanced Robotics* 24(5-6):739–761.

Nguyen, T., and Bushnell, L. G. 2004. Feasibility study of dtmf communications for robots. *Dept of EE, University of Washington Seattle WA* 98195–2500.

Okuno, H. G.; Nakadai, K.; and Kim, H.-D. 2011. Robot audition: Missing feature theory approach and active audition. In *Robotics Research*. Springer. 227–244.

Sakagami, Y.; Watanabe, R.; Aoyama, C.; Matsunaga, S.; Higaki, N.; and Fujimura, K. 2002. The intelligent asimo: System overview and integration. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, 2478–2483. IEEE.

Sauer, G.; Dickel, T.; and Lotter, T. 2014. Acoustic wireless control–connecting smart phones to hearing instruments.

Saxena, A., and Ng, A. Y. 2009. Learning sound location from a single microphone. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, 1737–1742. IEEE.

Srivastava, A.; Vijay, S.; Negi, A.; Shrivastava, P.; and Singh, A. 2014. Dtmf based intelligent farming robotic vehicle: An ease to farmers. In *Embedded Systems (ICES), 2014 International Conference on*, 206–210. IEEE.

Sun, H.; Yang, P.; Liu, Z.; Zu, L.; and Xu, Q. 2011. Microphone array based auditory localization for rescue robot. In *Control and Decision Conference (CCDC), 2011 Chinese*, 606–609. IEEE.