

平成 30 年度 特別研究報告書

距離センサを利用した高齢者との電話の
すれ違いに対応するシステムの検討

龍谷大学 理工学部 情報メディア学科

T150515 國友芽夢

指導教員 三好 力 教授

内容梗概

コミュニケーションツールとして、固定電話を利用する 65 歳以上の高齢者(以下高齢者)は多く存在すると考える。しかし聴力や身体能力の低下により応答に時間がかかり、相手が高齢者をいないと勘違いし電話を切ってしまう様な相手とのすれ違いが起きる問題も存在する。この問題に対し着信音増幅器等高齢者に配慮したシステムは複数存在するが、電話をかける人の精神的負担に配慮をしたシステム、デバイスは存在しない。そこで、本研究では電話をかける人に向けたシステムを提案する。手法として電話をかける前に遠隔から距離センサを起動し、電話と高齢者の距離情報を通知する事で高齢者の位置、動きが分かるシステムを提案する。またセンサの起動及び通知は Twitter を採用した。本実験では、提案システムと同様の動作が可能なプロトタイプを作成し検証を行なった。

目次

第1章 はじめに.....	4
第2章 既存技術.....	6
2.1 高齢者が聞き取りやすい着信音に変更する.....	6
2.1.1 自立コム テレアンプ III 電話音量増幅器 UA-45.....	6
2.1.2 携帯電話向け着信音配信サイト「ポケメロ JOYSOUND」.....	7
2.2 音以外の方法で着信を伝える.....	7
2.2.1 NZT-380 呼出音フラッシュコール.....	7
2.2.2 シルウォッチ 充電器式 腕時計受信機セット.....	8
2.3 問題点.....	8
第3章 提案手法.....	9
3.1 距離センサを利用した高齢者の位置の特定.....	9
3.2 Twitter の利用によるセンサの遠隔起動や距離の通知.....	9
3.3 提案システムの動作.....	10
第4章 実験.....	11
4.1 実験の準備.....	11
4.1.1 Raspberry Pi (Raspberry Pi3 Model B V1.2).....	11
4.1.2 Twitter API (Twitter API 4.0).....	11
4.1.3 Tweepy (Tweepy 3.7.0).....	11
4.1.4 超音波距離センサモジュール HC-SR04.....	11
4.2 システムの流れ.....	11
4.3 プロトタイプの詳細.....	13
4.4 固定電話の配置パターン.....	14
4.5 センサの観測可能角度を求める計算式.....	15
4.6 Twitter との連携の確認実験.....	16
4.6.1 目的.....	16
4.6.2 実験の概要.....	16
4.6.3 実験結果.....	16
4.6.4 考察.....	18
4.7 センサの認識可能範囲の確認実験.....	19
4.7.1 目的.....	19

4.7.2 実験概要	19
4.7.3 実験結果	20
4.7.4 考察	21
4.8 動く対象に対するセンサの確認実験	22
4.8.1 目的	22
4.8.2 実験概要	22
4.8.3 実験結果	22
4.8.4 考察	26
第5章 まとめ	27

第1章 はじめに

近年では高齢化社会が進み、65歳以上の高齢者（以下「高齢者」と記す。）人口は、平成29年9月15日現在の推計では3514万人となっている。同時期の総人口は1億2671万人となっており、3.6人に1人が高齢者となっている。又それに伴い一人暮らしの高齢者も増加しているのが現状である。図1.1に示すように内閣府の調査によると、65歳以上の一人暮らし高齢者は、昭和55（1980）年には男性約19万人、女性約69万人、高齢者人口に占める割合は男性4.3%、女性11.2%であったが、平成27（2015）年には男性約192万人、女性約400万人、高齢者人口に占める割合は男性13.3%、女性21.1%と大きく増加している。

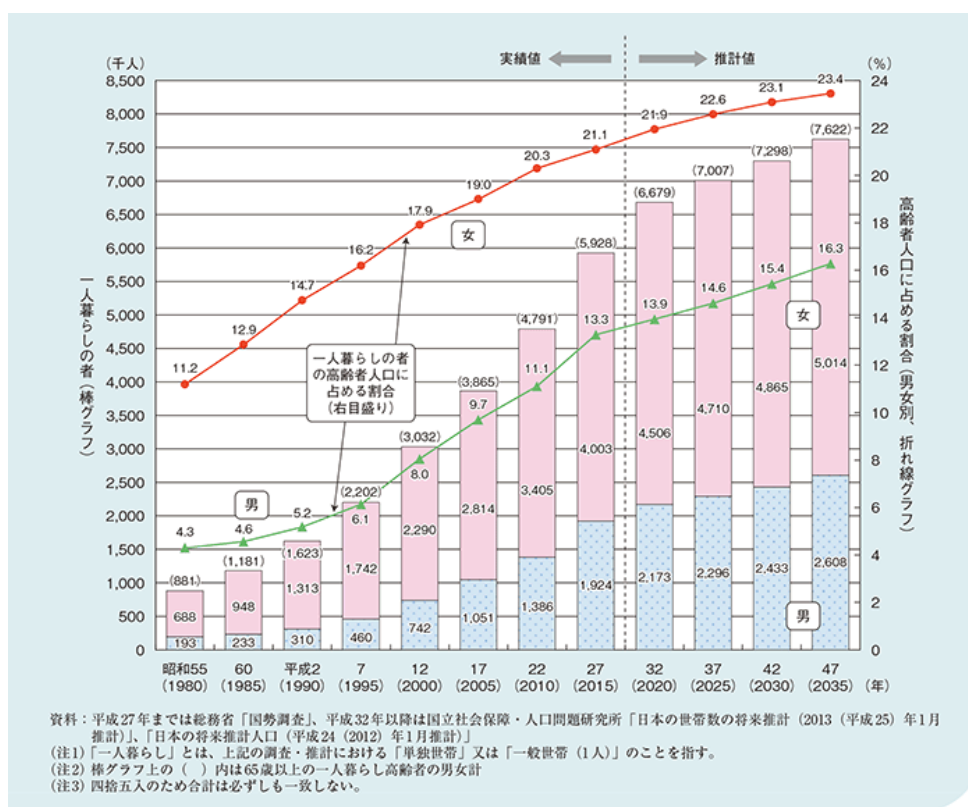


図1.1：65歳以上の一人暮らし高齢者の動向

その様な中で出てくる問題の1つとして、高齢者の子供や孫など家族とのコミュニケーション不足が挙げられる。近所に家族が住んでいない限り中々会ってコミュニケーションを取るとことは難しい。直接会う以外でのコミュニケーションの方法として、高齢者の大部分が固定電話を利用しているのではないかと考えられる。図1.2に示すように約9割の高齢者が固定電話を保持していること、電話番号を機器に入力すれば後は受話器を通して話すだけという操作の簡易性、彼らが幼少期の頃から電話は存在しており他の電子機器より馴染みがある、というメリットがあり、会話の次に使用率が高いと考えられる。一方で、聴力の低下による着信音や会話の音声がかええないという問題や、身体能力の低下によって受話器を取り応答するまでの時間がかかってしまい、電話をかけている相手の人が高齢者をいないと勘違いする事で電話を切ってしまう、相手とのすれ違い問題という様なデメリットも存在する。

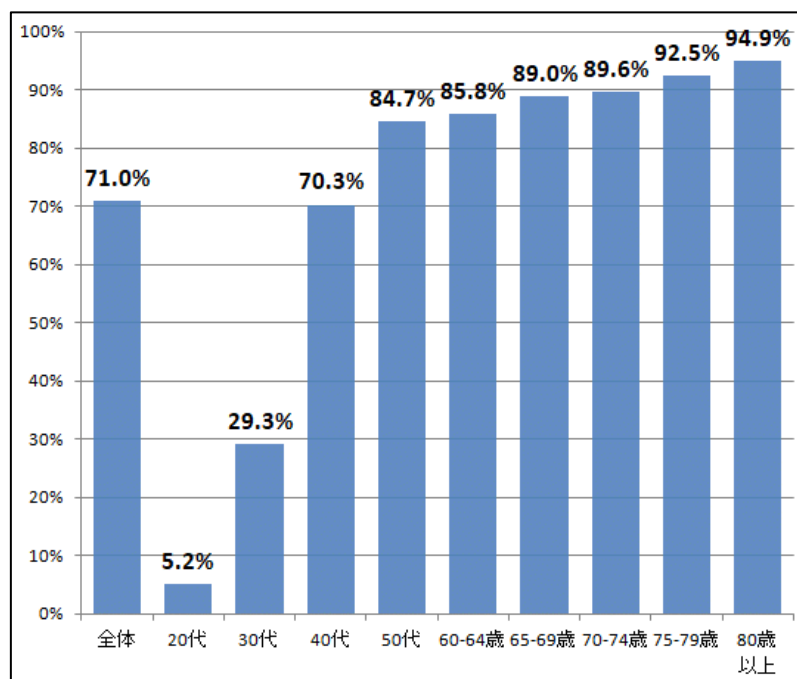


図 1.2：固定電話の保有状況(世帯単位、世帯主年齢階層別)(2017年)

着信音が聞き取りづらいという問題に関しては、音量増幅器など多く販売されており、比較的簡単に解消される。しかし、電話に出るまでの時間は短縮されない。電話をかけている側の人は相手の姿が見えない為に心配をしたり、不安になったりと精神的負担がかかるが、それに対する配慮をしたシステム、デバイスというものは存在しない。そこで、本論文では高齢者が電話に気がついており、応答しようとしているのかどうかについて電話をかけている側、家族などの若者の方が分かるシステムを提案する。

第2章 既存技術

電話の着信音が聞こえないという問題に対しての解決策として、大きく分けて2つの方法があった。

2.1 高齢者が聞き取りやすい着信音に変更する

2.1.1 自立コム テレアンプ III 電話音量増幅器 UA-45

この電話機着信音増幅器は電話機本体と電話機ケーブルの間にセットし、着信音をクリアで大きくすることによって高齢者でも聞きやすくするようにしている。

図 2.1.1.1 に電話機着信音増幅器の図を示す。



図 2.1.1.1 : 電話機着信音増音器

2.1.2 携帯電話向け着信音配信サイト「ポケメロ JOYSOUND」

音量をあげる以外の方法として、聞き取りやすい周波数で着信音を流すという方法もある。普段の生活で耳にするレンジなど家電の音や電話の着信音は、3000Hz 前後が多いとのことで、可聴域の低下している高齢者にはなかなか聞き取りづらくなっている。この「ポケメロ JOYSOUND」は、200Hz～2000Hz という周波数と低めの音で着信音を作成し、高齢者にとって聞き取りやすい音にした。また、音が低域から広域に変化するよう加工しており、雑音の中でも着信音と分かるよう工夫している。

2.1 の問題点は、どちらも着信音を加工してより聞きやすいようにしているが、確実に高齢者が気がつくとは限らないという問題点がある。

2.2 音以外の方法で着信を伝える

2.2.1 NZT-380 呼出音フラッシュコール

音以外の方法で着信を伝える方法として、目で分かるようパトランプや LED ライトを利用した機械も存在する。この「NZT-380 呼出音フラッシュコール」は電話着信音を音センサーで感知して、LED フラッシュライトが点滅し、着信音が聞こえづらい高齢者でも分かるようになっている。図 2.2.1.1 に呼出音フラッシュコールの図を示す。



図 2.2.1.1：呼出音フラッシュコール

2.2.2 シルウォッチ 充電器式 腕時計受信機セット

着信をバイブレーション（振動）と文字表示で知らせる。着信を振動で知らせることで、このデバイスを身につけている限り最も確実に着信を伝えられる。図 2.2.2.1 に腕時計受信機の図を示す。



図 2.2.2.1 シルウォッチ 充電器式 腕時計受信機セット

2.3 問題点

既存技術を調べた結果、着信をどのように高齢者に伝えるか、という手法はいくつか見つかったが、高齢者が応答するまでに時間がかかってしまった場合結局すれ違い問題が生じてしまう為、解決手法には不十分に感じた。

また、どのデバイスも高齢者に対してのものであり、電話をかけている側の不安、心配等の精神的負担を解消しているものは存在しなかった。

第3章 提案手法

2.3 で示した問題点を解決する為に、電話を受信した側の高齢者と、電話をかけている側の人ですれ違いを起こさず、更に電話をかけている側の精神的負担を軽減するシステムを提案する。

3.1 距離センサを利用した高齢者の位置の特定

高齢者が電話に出ようとしているのかを把握する為には、高齢者が電話に対してどのような位置、動きをしているのか把握が必要である。今回のシステムは高齢者の見守りではなく、電話のすれ違い問題を防ぎ、電話をかけた際にかけている側の人に高齢者がどのような状態であるのかを知らせるシステムである。その為大掛かりなシステムである必要はないと思われる。つまりある一定の条件時のみ起動する安価な機器を使用することが適切と考えた。また人を感知するだけでなく、電話に気が付き近づいてきているのか、気が付かず通り過ぎただけなのかをわかるようにしておく必要がある。その為距離を測定できる距離センサを採用した。この距離センサを利用することで電話と高齢者の距離を測定し、一定以上近づいたことが確認できればその情報を通知するというシステムを提案する。

3.2 Twitter の利用によるセンサの遠隔起動や距離の通知

センサは常に起動させた状態にする、電話の着信に反応し起動する、電話の直前に起動させるといくつか方法があるが、電話をかけるという特定の状況のみでしか使用しないということから、一つ目の常に起動をする必要はないと判断した。また二つ目の電話の着信に反応させるという方法に関しては、電話線を通しての受信確認は法律等から厳しく、着信音で判断するという方法では様々な音楽が存在し、また人物によって音楽を変えることが可能である着信音に対し正確に判別することは困難と判断した。その為三つ目の電話の直前に起動させるという方法を採用した。この方法で起動させる合図を送る手段として、電話をかける人が Twitter にて特定のツイートをするという手段を提案する。Twitter は手軽に投稿を行えるという点、またアカウント名に実名を使用する必要がない事や複数アカウントを作成しやすい事から、プライバシーを気にする必要がない点、以上よりセンサ起動の合図を送る、通知を受け取るツールとして適していると判断した。

3.3 提案システムの動作

本研究では電話をかける側の人には Twitter にて特定のワードをツイートしてもらい、それによって距離センサを起動させ、測定結果を Twitter にて通知するという方法を提案する。

具体的には、まず電話をかける側の人に Twitter にて「tell」という特定のワードを含んだツイートをしてもらう。するとその単語をきっかけに距離センサが起動し、1秒毎に測定が行われる。その測定値は Twitter にて通知されるが、今回は距離によって表示内容及び通知頻度を変える様にした。具体的には、センサが対象を認識でき、測定できた距離が 50cm 以上であった場合、1回目以降 2秒毎に「距離：～cm」と数値をツイートする。また対象とセンサの距離が 50cm 以下になった場合「50cm 以内に反応がありました ～cm」とツイートしプログラムを終了する。センサが正常に対象を検知できない状態が続く場合は 10秒毎に「測定中…」とツイートする、という様に各場合で表示を変更した。また Twitter の仕様である同じツイートが出来ないという課題に関しては各ツイートの最後に「～番目のツイート」と記載することで対応した。また高齢者が電話に気が付いていない、家にいない場合にも対応する為 150 回目の測定でプログラムを終了する。これは主な電話会社の呼び出し時間が最大でも 120 秒ほどであった為、電話をかける時間も考慮して 150 回とした。

第4章 実験

提案システムと同様の動作が可能なプロトタイプを作成し、提案システムの有効性を検証する。

4.1 実験の準備

今回実験を行うにあたって以下の機器、システムを使用した。

4.1.1 Raspberry Pi (Raspberry Pi3 Model B V1.2)

Raspberry Pi (ラズベリー パイ) は、ARM プロセッサを搭載したシングルボードコンピュータ。本実験ではこの Raspberry Pi と距離センサをつなげ距離を測定した。また Twitter と連携し、つぶやきの取得、データの送信等のやり取りを行った。

4.1.2 Twitter API (Twitter API 4.0)

Twitter API とは、Twitter の機能を外部のプログラムから利用する為のインターフェースである[11]。今回はこの Twitter API を利用し、Twitter の基本機能を自身が作成したプログラムでも使用できるよう OAuth 認証を用いた。

4.1.3 Tweepy (Tweepy 3.7.0)

Twitter API にアクセスする為の Python ライブラリ。Python を通してツイートやタイムライン取得、検索など幅広く利用することができ[12]、今回はこのライブラリを使って Twitter で自身のツイートの取得、ワードの検索、Raspberry Pi から自身のアカウントにアクセスし、ツイートを行う際に利用した。

4.1.4 超音波距離センサモジュール HC-SR04

本実験にて利用する超音波距離センサで、超音波を出しその反射を受信して対象物までの距離を測る。また、測定可能距離は約 2cm-450cm となっている[13]。距離を測定するセンサは赤外線センサ、レーザセンサ、3次元センサ、超音波センサなど様々な種類があるが、今回は長距離の測定が可能である点、距離の精度は低くて良い点、安価な点からこの「超音波距離センサモジュール HC-SR04」を選択した。

4.2 システムの流れ

Twitter のタイムラインを読み込み、センサを起動、測定し通知を行った後、終了するまでのフローチャートを図 4.2.1 に示す。

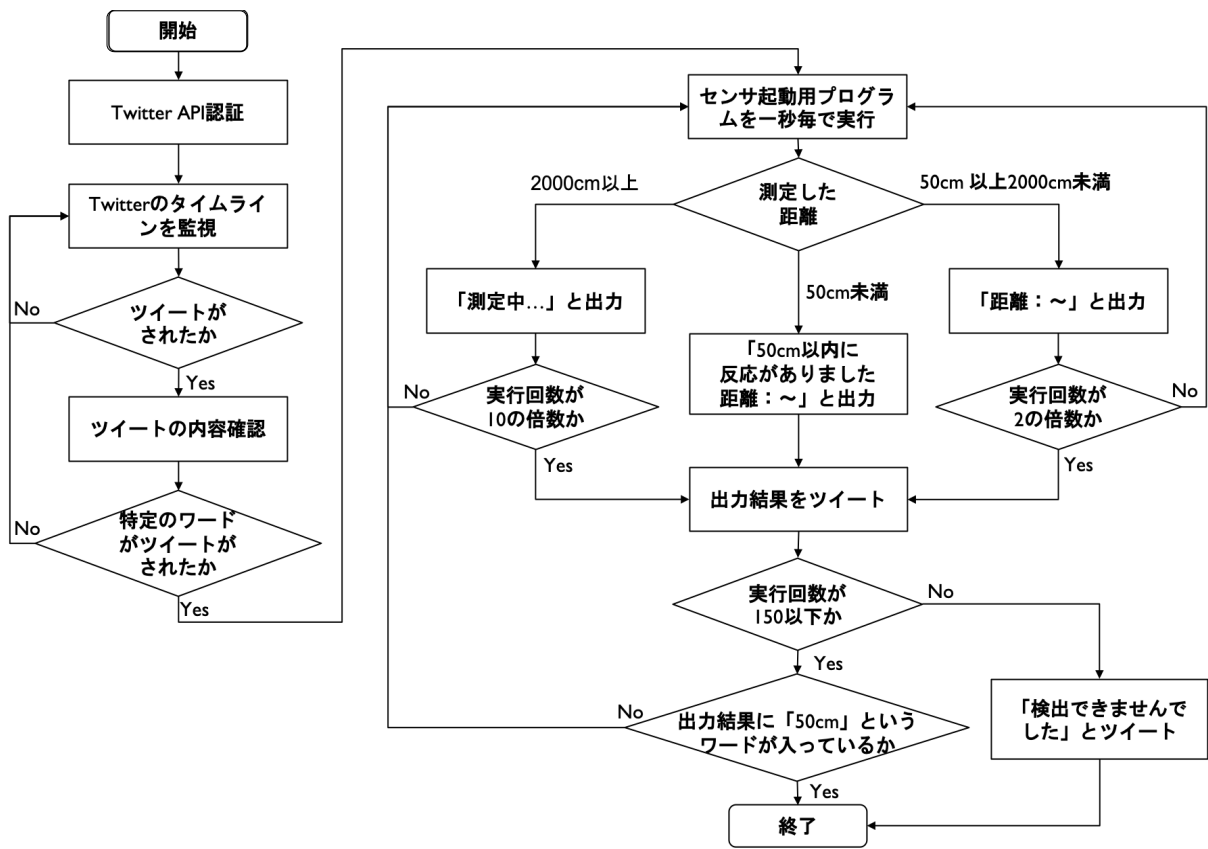


図 4.2.1：フローチャート

4.3 プロトタイプの概要

今回の実験で使用したシステムの図を図 4.3.1 に示す。

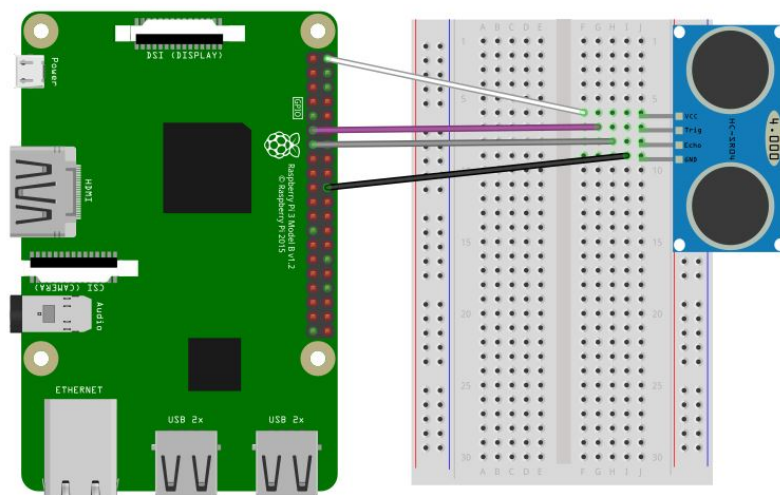


図 4.3.1：プロトタイプとして採用した Raspberry Pi と
超音波距離センサモジュール HC-SR04

4.4 固定電話の配置パターン

インターネット上に掲載してある固定電話の配置場所に関する文章[14]を参考に、主な固定電話の配置場所は図 4.4.1、図 4.4.2、図 4.4.3 の 3 種類とした。

なお図の矢印は人が固定電話に向かって移動する際の動き方を表しており、最終的には「立ち位置」と書かれた場所に移ると想定した。

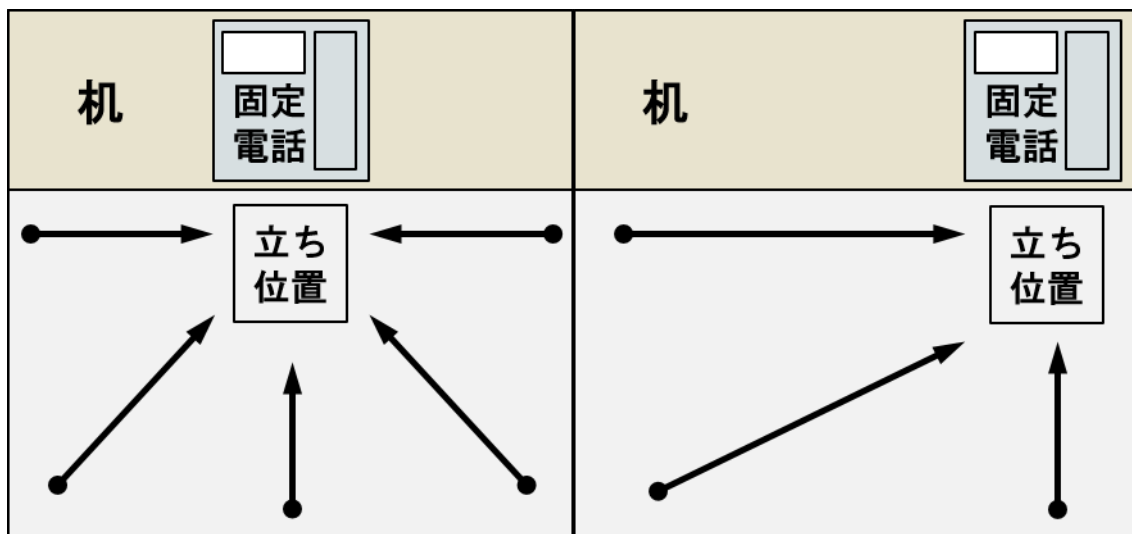


図 4.4.1：固定電話の位置(中央に配置)

図 4.4.2：固定電話の位置(隅に配置)

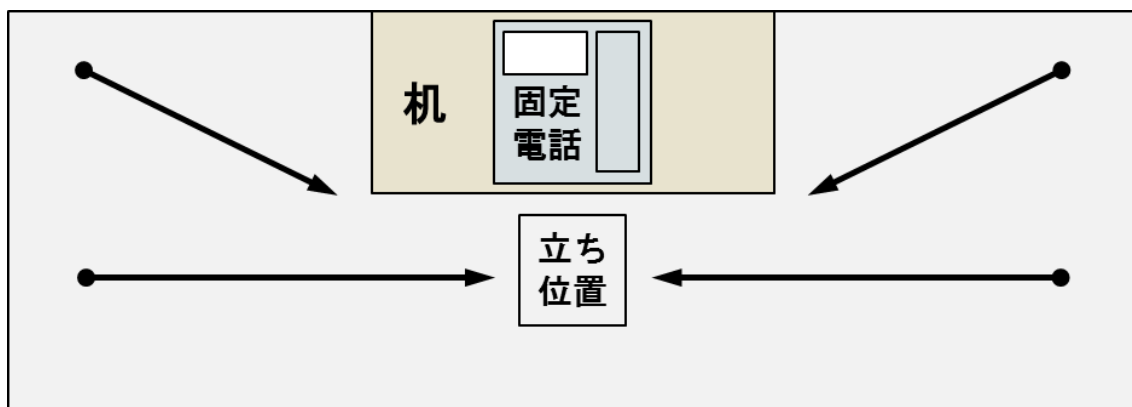


図 4.4.3：固定電話の位置(廊下に配置)

今回は以上の3パターンの配置にてセンサが人を検知する場合、どれくらいの個数でセンサを配置する必要があるのかの確認を行う。その為センサの指向性を調べる実験を行なった。

また今回は距離の精密さはある程度で良い点、広域を測定できる点、安価である点から超音波距離センサモジュールを採用したが、高齢者と電話の距離を測定するセンサとして適切なのかも確認する必要があると考えた。その為高齢者が電話へと向かうという想定動きで実際に実験を行う。

4.5 センサの観測可能角度を求める計算式

センサの測定できる範囲の角度を求めるにあたり式 4.5.1 を使用する。図 4.5.1 の様にセンサとの垂直な距離を a とし、その地点からセンサと平行に移動した距離を b とし角度 θ を求めた。また観測可能角度は図 4.5.1 の様に2倍になる為、 2θ として考察を行なった。

$$\theta = \tan^{-1}\left(\frac{b}{a}\right) \quad (\text{式 4.5.1})$$

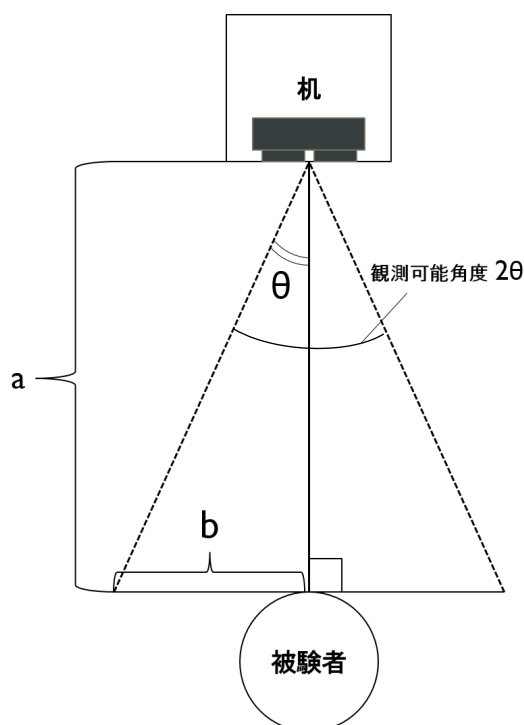


図 4.5.1 : 観測可能角度

4.6 Twitter との連携の確認実験

4.6.1 目的

特定のワードをツイートすることで、正常にセンサが動作し、測定値が通知されるかの確認を行う為、実際にツイートをする事で動作確認を行った。

4.6.2 実験の概要

まず Twitter にてセンサを起動するツイートをし、ツイート後センサ認識可能範囲外の 5m 地点(センサの真正面の位置)からセンサに向かって歩いた。センサは測定範囲の広い腰辺りを認識させる為、地面から 75 cm の位置に配置し、正常に起動され距離データがツイートされているかの確認を行った。また家に高齢者がいなかった場合を想定し、一定時間でセンサが停止するかも確認を行った。

4.6.3 実験結果

図 4.6.3.1 に 11 月 20 日 17 時 59 分に実際にセンサを起動し、通知を受け取った Twitter 画面を示す。まずセンサを起動させるツイートが表示され、その後再度タイムラインを読み込むと最初の距離情報が表示された。その後も図 4.6.3.1 を見るとセンサが被験者を認識している限り、タイムラインを読み込む度に距離情報が表示され、また一定以上被験者がセンサに近づいた際には通知がなくなったことが分かる。

また図 4.6.3.2 は高齢者が認識できなかった場合の Twitter 画面を示す。11 月 19 日 18 時 52 分にツイートした画面の図であり、この図からツイート後 15 回目の「測定中…」のツイートが表示されると「検出できませんでした」というツイートが表示されたことが分かる。またそれと同時に通知のツイートが停止し、センサも停止した。



図 4.6.3.1 : 11月20日17時59分のツイート



図 4.6.3.2 : 11月19日18時52分のツイート

4.6.4 考察

特定のワードをツイート後、測定値がツイートにて通知されていたことが確認できた。また被験者を一定時間認識しなかった場合は自動で通知及びセンサを停止していたことも確認ができた。これより、今回の提案しているシステムのプロトタイプは正常に動作していることが分かった。

また Twitter のタイムラインはページを再度読み込まない限り自動で更新することではなく、最新のデータを受け取るには何度もページを読み込む必要があることが分かった。その為、リアルタイムでどんどんと数値が表示されていく様な環境で通知をする必要があると考えられる。

4.7 センサの認識可能範囲の確認実験

4.7.1 目的

センサの性能の確認及び大まかな認識可能範囲を確認する為、立ち止まった状態での距離測定の実験を行なった。

4.7.2 実験概要

図 4.7.2.1 のようにセンサから距離 5.0m、横幅左右 1.5m ずつの範囲を 0.5m 間隔で 10 回ずつ測定した。なお、正面以外では電話に向かっていることを想定し体はセンサに向けて確認を行った。

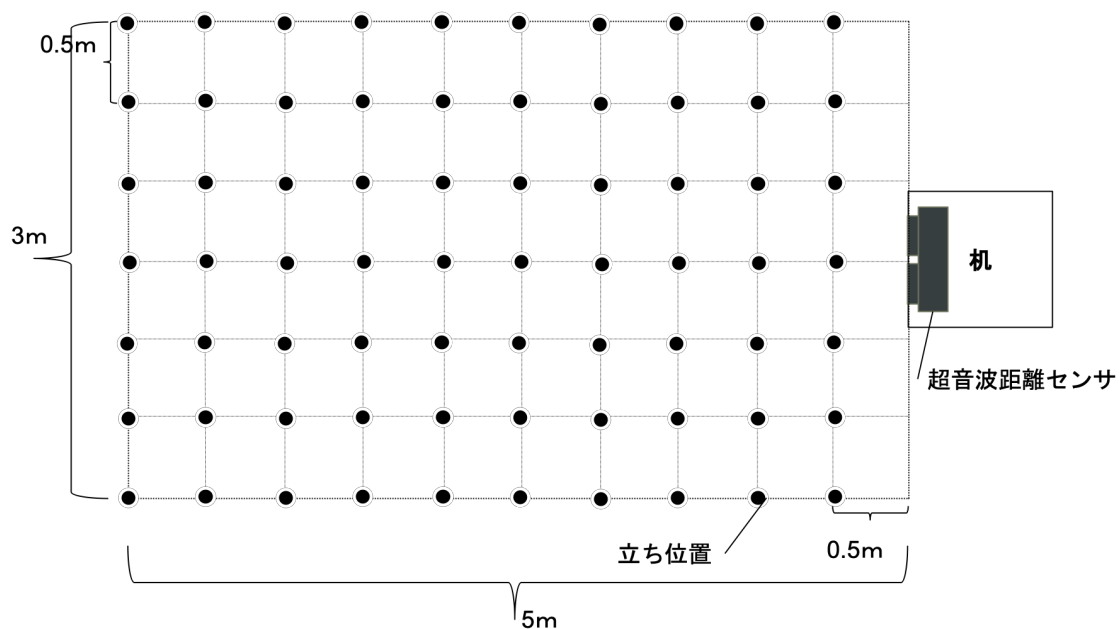


図 4.7.2.1 : センサの認識可能範囲の確認実験

4.7.3 実験結果

測定したデータを表 4.7.3.1~4.7.3.5 に示す。

正面からの測定では測定可能距離の 4.5m まで距離の測定が可能であった。しかし右又は左に 0.5~1.0m ずれた状態で測定すると 1.0m 又は 1.5m~2.5m の間の距離しか測定することは不可能であった。また、1.5m 以降横にずれた場合には全ての距離で測定不能であった。

表 4.7.3.1：正面からの測定データ

回数(回)	距離(cm)	50	100	150	200	250	300	350	400	450	500
1		51.0	99.3	148.3	202.6	248.4	299.8	345.9	405.0	446.1	測定不可
2		51.3	99.0	149.3	199.8	248.4	295.2	346.1	401.7	441.7	測定不可
3		51.0	100.4	149.2	198.2	247.8	296.1	346.1	402.6	446.1	測定不可
4		52.2	98.8	147.2	199.9	247.9	297.0	347.5	402.9	443.5	489.8
5		49.9	99.0	146.7	199.6	247.6	299.0	346.8	402.3	442.0	測定不可
6		50.4	98.6	147.2	198.1	247.2	298.3	344.7	403.0	445.8	測定不可
7		51.4	98.7	147.8	198.2	249.4	297.1	343.8	401.1	444.2	測定不可
8		50.4	101.4	148.4	198.2	250.2	298.6	343.5	403.1	443.4	測定不可
9		50.8	101.0	147.5	198.1	248.5	299.3	346.5	399.1	445.1	測定不可
10		48.6	100.6	148.4	197.1	246.9	298.6	344.0	398.9	442.8	測定不可

表 4.7.3.2：右に 0.5m ずれた位置からの測定データ

回数(回)	距離(cm)	50	100	150	200	250	300	350	400	450	500
1		測定不可	103.2	148.4	199.7	測定不可	測定不可	測定不可	測定不可	測定不可	測定不可
2		測定不可	104.0	151.3	199.4	255.8	測定不可	測定不可	測定不可	測定不可	測定不可
3		測定不可	測定不可	150.1	199.2	測定不可	測定不可	測定不可	測定不可	測定不可	測定不可
4		測定不可	102.5	152.6	198.0	251.6	測定不可	測定不可	測定不可	測定不可	測定不可
5		測定不可	101.1	151.5	198.5	247.1	測定不可	測定不可	測定不可	測定不可	測定不可
6		測定不可	100.1	測定不可	198.3	測定不可	測定不可	測定不可	測定不可	測定不可	測定不可
7		測定不可	100.8	149.6	198.8	249.5	289.1	測定不可	測定不可	測定不可	測定不可
8		測定不可	100.5	150.3	198.4	249.0	測定不可	測定不可	測定不可	測定不可	測定不可
9		測定不可	98.6	150.6	197.9	253.4	測定不可	測定不可	測定不可	測定不可	測定不可
10		測定不可	100.0	149.5	196.5	249.6	測定不可	測定不可	測定不可	測定不可	測定不可

表 4.7.3.3：左に 0.5m ずれた位置からの測定データ

回数(回)	距離(cm)	50	100	150	200	250	300	350	400	450	500
1		測定不可	104.6	156.3	197.2	測定不可	測定不可	測定不可	測定不可	測定不可	測定不可
2		測定不可	103.9	154.8	203.9	測定不可	測定不可	測定不可	測定不可	測定不可	測定不可
3		測定不可	103.7	154.6	203.9	250.7	測定不可	測定不可	測定不可	測定不可	測定不可
4		測定不可	103.7	154.9	203.9	250.8	測定不可	測定不可	測定不可	測定不可	測定不可
5		測定不可	102.8	154.1	203.6	測定不可	測定不可	測定不可	測定不可	測定不可	測定不可
6		測定不可	102.7	154.6	202.0	250.0	測定不可	測定不可	測定不可	測定不可	測定不可
7		測定不可	102.0	155.0	202.0	248.6	測定不可	測定不可	測定不可	測定不可	測定不可
8		測定不可	102.8	155.0	203.7	247.6	測定不可	測定不可	測定不可	測定不可	測定不可
9		測定不可	103.0	測定不可	205.1	251.8	測定不可	測定不可	測定不可	測定不可	測定不可
10		測定不可	103.3	155.3	204.0	250.9	測定不可	測定不可	測定不可	測定不可	測定不可

表 4.7.3.4：右に 1.0m ずれた位置からの測定データ

	距離(cm)									
回数(回)	50	100	150	200	250	300	350	400	450	500
1	測定不可	測定不可	176.7	213.7	263.7	307.6	測定不可	測定不可	測定不可	測定不可
2	測定不可	測定不可	測定不可	211.9	262.3	測定不可	測定不可	測定不可	測定不可	測定不可
3	測定不可	測定不可	178.0	測定不可	265.1	測定不可	測定不可	測定不可	測定不可	測定不可
4	測定不可	測定不可	178.0	212.3	264.6	測定不可	測定不可	測定不可	測定不可	測定不可
5	測定不可	測定不可	177.0	213.7	265.5	測定不可	測定不可	測定不可	測定不可	測定不可
6	測定不可	測定不可	175.1	213.2	264.6	312.4	測定不可	測定不可	測定不可	測定不可
7	測定不可	測定不可	176.5	215.8	測定不可	測定不可	測定不可	測定不可	測定不可	測定不可
8	測定不可	測定不可	176.2	測定不可	262.9	測定不可	測定不可	測定不可	測定不可	測定不可
9	測定不可	測定不可	174.9	211.3	263.5	測定不可	測定不可	測定不可	測定不可	測定不可
10	測定不可	測定不可	178.6	測定不可	264.6	測定不可	測定不可	測定不可	測定不可	測定不可

表 4.7.3.5 : 右に 1.0m ずれた位置からの測定データ

	距離(cm)									
回数(回)	50	100	150	200	250	300	350	400	450	500
1	測定不可	測定不可	168.9	測定不可	260.6	測定不可	測定不可	測定不可	測定不可	測定不可
2	測定不可	測定不可	168.2	220.4	測定不可	測定不可	測定不可	測定不可	測定不可	測定不可
3	測定不可	測定不可	169.4	220.2	測定不可	299.4	測定不可	測定不可	測定不可	測定不可
4	測定不可	測定不可	168.5	測定不可	269.9	測定不可	測定不可	測定不可	測定不可	測定不可
5	測定不可	測定不可	169.8	220.1	261.1	測定不可	測定不可	測定不可	測定不可	測定不可
6	測定不可	測定不可	170.0	218.4	265.6	測定不可	測定不可	測定不可	測定不可	測定不可
7	測定不可	測定不可	168.5	217.7	測定不可	測定不可	測定不可	測定不可	測定不可	測定不可
8	測定不可	測定不可	168.2	218.6	268.2	測定不可	測定不可	測定不可	測定不可	測定不可
9	測定不可	測定不可	169.6	218.4	266.2	測定不可	測定不可	測定不可	測定不可	測定不可
10	測定不可	測定不可	168.6	217.6	268.5	測定不可	測定不可	測定不可	測定不可	測定不可

4.7.4 考察

止まった状態で正面に立った場合、センサの測定可能範囲である 4.5m まで測定できており、人を測定対象としても問題なく動作している事がわかる。また、1.5m 以上横にずれた場合は測定できないが、正面のみではなく、ある程度の横のずれであれば問題なく測定できるという事がわかった。

4.8 動く対象に対するセンサの確認実験

4.8.1 目的

本実験では歩いている人を認識し距離が測定できるかの確認を、一定の位置からセンサに向かって歩き距離を測定する事で行った。またこの実験からセンサの指向性を調べ、図 4.4.1、図 4.4.2、図 4.4.3 の固定電話の配置パターンに対し、センサが何個必要なのかを調べた。

4.8.2 実験概要

図 4.8.2.1 の様にセンサの測定可能範囲外の 5.0m 地点から歩いてセンサに近づいた。正面からの経路は 5 秒間と 7 秒間の 2 種類の時間をかけて、0.3m ずつ横にずれた経路での測定は 5 秒間かけて移動した。この時間は山崎昌廣、佐藤陽彦が 1990 年に書いた論文「ヒトの歩行」を参考に、高齢者が普段歩いている速度と、電話を受け移動するとなるとやや初速度は落ちると考え、5 秒間と 7 秒間にした。

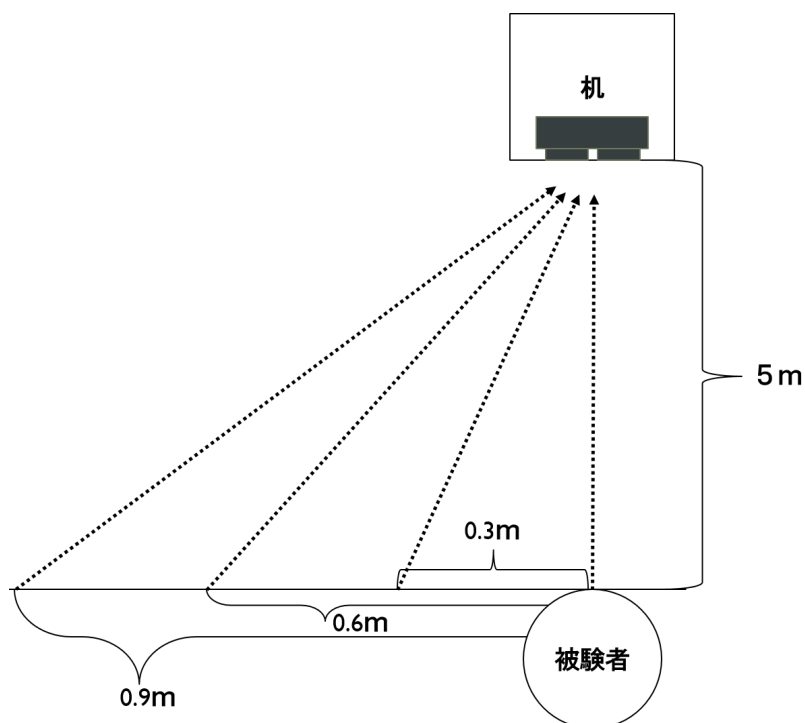


図 4.8.2.1 : 動く対象に対するセンサの確認実験

4.8.3 実験結果

測定したデータを表 4.8.3.1~4.8.3.5 に示す

これらの図から、歩行速度に関わらず 3.0m 以遠の距離はあまり測定できなかったことが分かる。その為か、センサにたどり着く 2 秒ほど前に対象との距離が測定可能になることが多かった。また経路が横にずれると共に測定不可能になることが多くなり、0.9m 横にずれた経路ではセンサとの距離が約 1m にならない限りほとんど測定不可能になった。

表 4.8.3.1：正面からの経路で 5 秒間かけての移動

実験回数	1 秒後	2 秒後	3 秒後	4 秒後	5 秒後
1	測定不可	測定不可	測定不可	241.2	54.2
2	測定不可	測定不可	278.5	241.4	64.1
3	測定不可	測定不可	測定不可	248.5	72
4	測定不可	測定不可	測定不可	265.5	91.4
5	測定不可	測定不可	測定不可	226.6	74.6
6	測定不可	測定不可	374	254.1	57.2
7	測定不可	測定不可	361.9	195.5	50.6
8	測定不可	測定不可	測定不可	測定不可	88.1
9	測定不可	測定不可	339.1	176.9	67.9
10	測定不可	測定不可	測定不可	273.6	88
11	測定不可	測定不可	測定不可	220.3	50.1
12	測定不可	測定不可	測定不可	278.7	72.4
13	測定不可	測定不可	測定不可	243.6	65.8
14	測定不可	測定不可	測定不可	234.3	85.9
15	測定不可	測定不可	測定不可	283.1	89.6
16	測定不可	測定不可	測定不可	283.2	92.4
17	測定不可	測定不可	測定不可	303.4	85.6
18	測定不可	測定不可	測定不可	277.8	85.4
19	測定不可	測定不可	測定不可	測定不可	98.5
20	測定不可	測定不可	測定不可	274.4	78.5
21	測定不可	測定不可	測定不可	測定不可	104.3
22	測定不可	測定不可	測定不可	279.7	94.9
23	測定不可	測定不可	測定不可	248.1	58.3
24	測定不可	測定不可	測定不可	229	64.7
25	測定不可	測定不可	388.5	209.6	74.3
26	測定不可	測定不可	測定不可	測定不可	118.9
27	測定不可	測定不可	391.5	246.9	81
28	測定不可	測定不可	測定不可	263.4	92.3
29	測定不可	測定不可	測定不可	232.7	62.1
30	測定不可	測定不可	445.5	269.1	110.1

表 4.8.3.2 : 正面からの経路で 7 秒間かけての移動

	1 秒後	2 秒後	3 秒後	4 秒後	5 秒後	6 秒後	7 秒後
1	測定不可	測定不可	測定不可	測定不可	測定不可	223.2	79
2	測定不可	測定不可	測定不可	測定不可	測定不可	190.4	52.5
3	測定不可	測定不可	測定不可	測定不可	測定不可	206.3	59
4	測定不可	測定不可	測定不可	測定不可	測定不可	213.5	65
5	測定不可	測定不可	測定不可	測定不可	測定不可	174.1	44.4
6	測定不可	測定不可	測定不可	測定不可	測定不可	182.1	58.5
7	測定不可	測定不可	測定不可	測定不可	測定不可	190.1	57.3
8	測定不可	測定不可	測定不可	測定不可	測定不可	測定不可	89.1
9	測定不可	測定不可	測定不可	測定不可	測定不可	193.8	60.3
10	測定不可	測定不可	測定不可	測定不可	302.9	169.4	69.2
11	測定不可	測定不可	測定不可	測定不可	測定不可	197.5	52.9
12	測定不可	測定不可	測定不可	測定不可	測定不可	測定不可	52.2
13	測定不可	測定不可	測定不可	測定不可	318.8	172.5	52.2
14	測定不可	測定不可	測定不可	測定不可	294.7	172.1	64.6
15	測定不可	測定不可	測定不可	測定不可	302.5	155.8	47.5
16	測定不可	測定不可	測定不可	測定不可	測定不可	219.7	81.4
17	測定不可	測定不可	測定不可	測定不可	測定不可	208.3	51.9
18	測定不可	測定不可	測定不可	測定不可	測定不可	191.8	73.9
19	測定不可	測定不可	測定不可	測定不可	292.9	179.5	52.1
20	測定不可	測定不可	測定不可	435.9	測定不可	217.9	79.6
21	測定不可	測定不可	測定不可	測定不可	測定不可	測定不可	89.6
22	測定不可	測定不可	測定不可	測定不可	327.5	175.5	49.7
23	測定不可	測定不可	測定不可	測定不可	測定不可	測定不可	83.2
24	測定不可	測定不可	測定不可	測定不可	297.9	197.2	55.8
25	測定不可	測定不可	測定不可	測定不可	測定不可	208.9	55.3
26	測定不可	測定不可	測定不可	測定不可	318.2	179.3	59.3
27	測定不可	測定不可	測定不可	測定不可	306	182.7	77.5
28	測定不可	測定不可	測定不可	測定不可	測定不可	測定不可	83.1
29	測定不可	測定不可	測定不可	測定不可	測定不可	194.9	59.4
30	測定不可	測定不可	測定不可	測定不可	301.2	192.4	56.4

表 4.8.3.3 : 0.3m 横からの経路で 5 秒間かけての移動

実験回数	1 秒後	2 秒後	3 秒後	4 秒後	5 秒後
1	測定不可	496.5	測定不可	191.4	61.4
2	測定不可	測定不可	測定不可	213.8	63.7
3	測定不可	測定不可	測定不可	172.3	62.4
4	測定不可	測定不可	測定不可	211	58.1
5	測定不可	測定不可	測定不可	測定不可	53.5
6	測定不可	測定不可	測定不可	測定不可	58.9
7	測定不可	測定不可	測定不可	測定不可	76.6
8	測定不可	測定不可	417.9	213.1	59.5
9	測定不可	測定不可	測定不可	測定不可	73.7
10	測定不可	500.3	測定不可	192.9	79.6
11	測定不可	測定不可	測定不可	測定不可	117.5
12	測定不可	測定不可	測定不可	239.3	55.2
13	測定不可	測定不可	366.3	218.9	68
14	測定不可	測定不可	測定不可	226.5	49.5
15	測定不可	測定不可	測定不可	245.6	52.8
16	測定不可	466.4	367.3	238.6	56
17	測定不可	測定不可	396.7	測定不可	65.8
18	測定不可	測定不可	測定不可	223.8	57.1
19	測定不可	測定不可	381.3	225.7	57.9
20	測定不可	測定不可	測定不可	229.7	51.5
21	測定不可	測定不可	測定不可	測定不可	74.5
22	測定不可	測定不可	測定不可	199.6	54
23	測定不可	測定不可	測定不可	測定不可	72.5
24	測定不可	測定不可	測定不可	測定不可	76
25	測定不可	測定不可	373.9	248.9	67.9
26	測定不可	測定不可	396.2	197.6	57.2
27	測定不可	測定不可	測定不可	229.7	61.2
28	測定不可	測定不可	測定不可	204.1	69.3
29	測定不可	測定不可	測定不可	230.7	65.8
30	測定不可	504.5	374.6	205.9	62

表 4.8.3.4 : 0.6m 横からの経路で 5 秒間かけての移動

実験回数	1 秒後	2 秒後	3 秒後	4 秒後	5 秒後
1	測定不可	測定不可	測定不可	258.6	73
2	測定不可	測定不可	測定不可	226.6	53
3	測定不可	測定不可	測定不可	217.8	59.8
4	測定不可	測定不可	測定不可	測定不可	101
5	測定不可	測定不可	測定不可	242.5	56.7
6	測定不可	測定不可	測定不可	238.5	58
7	測定不可	測定不可	測定不可	測定不可	87.6
8	測定不可	測定不可	測定不可	243.7	58
9	測定不可	測定不可	測定不可	238.8	67.3
10	測定不可	測定不可	測定不可	225.5	84.5
11	測定不可	測定不可	測定不可	測定不可	91.9
12	測定不可	測定不可	測定不可	265	66.6
13	測定不可	測定不可	369.2	測定不可	84.1
14	測定不可	測定不可	測定不可	250.9	72.2
15	測定不可	測定不可	測定不可	測定不可	56
16	測定不可	測定不可	測定不可	測定不可	60.4
17	測定不可	測定不可	400.4	測定不可	86.9
18	測定不可	測定不可	測定不可	測定不可	72.1
19	測定不可	測定不可	測定不可	測定不可	86.7
20	測定不可	測定不可	測定不可	測定不可	97.6
21	測定不可	測定不可	測定不可	227.2	77.1
22	測定不可	測定不可	測定不可	248.6	51.6
23	測定不可	測定不可	測定不可	測定不可	82.1
24	測定不可	測定不可	測定不可	測定不可	82.9
25	測定不可	測定不可	測定不可	233.8	75.8
26	測定不可	測定不可	測定不可	測定不可	92.3
27	測定不可	測定不可	測定不可	測定不可	54.3
28	測定不可	測定不可	測定不可	267.8	79.1
29	測定不可	測定不可	測定不可	測定不可	91.2
30	測定不可	測定不可	測定不可	測定不可	67.3

表 4.8.3.5 : 0.9m 横からの経路で 5 秒間かけての移動

実験回数	1 秒後	2 秒後	3 秒後	4 秒後	5 秒後
1	測定不可	測定不可	測定不可	298.1	124.9
2	測定不可	測定不可	測定不可	測定不可	99.3
3	測定不可	測定不可	測定不可	測定不可	105.2
4	測定不可	測定不可	測定不可	301.6	123.1
5	測定不可	測定不可	測定不可	測定不可	99.2
6	測定不可	測定不可	測定不可	測定不可	94.4
7	測定不可	測定不可	測定不可	測定不可	98.7
8	測定不可	測定不可	測定不可	測定不可	109.9
9	測定不可	測定不可	測定不可	測定不可	88.6
10	測定不可	測定不可	測定不可	測定不可	104.4
11	測定不可	測定不可	測定不可	測定不可	106.3
12	測定不可	測定不可	測定不可	測定不可	88.1
13	測定不可	測定不可	測定不可	測定不可	101.3
14	測定不可	測定不可	測定不可	271.8	104.9
15	測定不可	測定不可	測定不可	測定不可	110.4
16	測定不可	測定不可	測定不可	測定不可	86.9
17	測定不可	測定不可	測定不可	測定不可	121.6
18	測定不可	測定不可	測定不可	測定不可	96.6
19	測定不可	測定不可	測定不可	測定不可	97.6
20	測定不可	測定不可	測定不可	測定不可	117.2
21	測定不可	測定不可	測定不可	測定不可	83.3
22	測定不可	測定不可	測定不可	測定不可	82.4
23	測定不可	測定不可	測定不可	測定不可	94.8
24	測定不可	測定不可	測定不可	266.9	97.3
25	測定不可	測定不可	測定不可	測定不可	91.2
26	測定不可	測定不可	測定不可	測定不可	92.3
27	測定不可	測定不可	測定不可	測定不可	102.8
28	測定不可	測定不可	測定不可	測定不可	107.3
29	測定不可	測定不可	測定不可	測定不可	84.8
30	測定不可	測定不可	測定不可	309.3	95.3

4.8.4 考察

実験 4.7 の時に比べて、歩いた場合の測定であるとかかなり測定できる範囲が減少した。これは超音波を出しその反射を受信して対象物までの距離を測る超音波距離センサの特性上、超音波の反射がセンサに安定して返ってこない対象の測定は厳しいものであったと考える。

また今回の超音波距離センサは指向性が高いものであったと考え、ほとんど計測が出来なかった 0.9m 地点からの経路ではセンサは認識出来ないと考えた。また測定できた回数が正面からの経路の時に比べ少ない 0.3m~0.6m の経路の範囲は複数のセンサで距離を測定することでより確実に測定できるようになるのではないかと考えた。

5.0m 地点で 0.3m 横にずれた場合の角度は式 4.5.1 より約 3.4 度である。また 0.6m ずれた場合は約 6.8 度となる。これにより、図 4.8.4.1 の様に 0.3m から 0.6m の範囲を重ねて配置した場合 90 度の範囲では 9 個のセンサを用意することでほぼ確実に対象を感知できる事が分かった。よって図 4.4.1、図 4.4.2、図 4.4.3 の 3 パターンに対し、図 4.4.2 の様に 90 度の範囲で測定する場合は 9 個、図 4.4.1、図 4.4.3 の様に 180 度の範囲で測定する場合はその倍の 18 個必要であることが分かった。

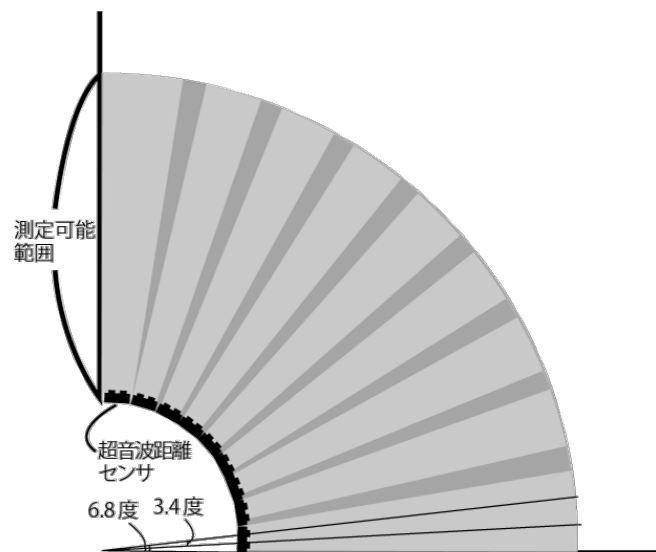


図 4.8.4.1 : 90 度の範囲で必要になるセンサの数と測定可能範囲

第5章 まとめ

本実験では提案システムと同様の動作が可能なプロトタイプを作成し検証を行なった。その結果、Twitter 及び多数のセンサを利用する事で電話のすれ違いに対応するシステムを実現することができたと考えている。今後の課題としては、今回使用した超音波距離センサは指向性が高かった為、他の種類のセンサでも試す必要がある事と、Twitter のタイムラインはページを何度も再読み込みしない限り最新の情報を受け取ることが難しい為、吹き出しでポップアップをしてくれるダイレクトメッセージという Twitter の機能を利用したセンサの起動、通知が行えるようにしたいと考えている。

謝辞

本研究を進めるにあたり多くの方々のご指導、ご支援をいただきました。心より深くお礼申し上げます。

龍谷大学工学部情報メディア学科三好力教授には本研究の指導教員を担当して頂きました。研究内容の御指導、御助力頂き、心より御礼申し上げます。

また三好研究室の皆様には、研究に対する多くの御意見、御助力、御指導を頂きました。心より御礼申し上げます。

参考文献

- [1] 高齢者の人口 | 総務相統計局, <http://www.stat.go.jp/data/topics/topi1031.html>
- [2] 平成 29 年版高齢社会白書 (全体版) | 内閣府,
http://www8.cao.go.jp/kourei/whitepaper/w-2017/html/zenbun/s1_2_1.html
- [3] 通信利用動向調査 | 総務省 情報通信行政・郵政行政審議会,
<http://www.soumu.go.jp/johotsusintokei/statistics/statistics05a.html>
- [4] テレアンプ III 自立コム, <http://www.jiritsu.com/products/detail.php?id=79>
- [5] JOYSOUND 着メロ, https://www.joysound.com/web/s/mobile/js_melo
- [6] 呼出音フラッシュコール (NZT-380) 大切な電話を逃さない | なんず,
<https://www.nanzu.jp/syohin/nzt-380.htm>
- [7] 聴覚障害者用屋内信号装置 | 株式会社東京信友,
<http://www.shinyu.co.jp/product/main/watch1.html>
- [8] Twitter, <https://twitter.com>
- [9] アンドロイドの留守番電話:呼び出し時間を変更するには? 設定方法/スマホ,Android,ドコモ,docomo,au,ソフトバンク,softbank,
https://andropp.jp/cat_tel/004021478/
- [10] Raspberry Pi-Teach, Learn, and Make with Raspberry Pi,
<https://www.raspberrypi.org/>
- [11] Twitter の API について - Twitter Help Center,
<https://help.twitter.com/ja/rules-and-policies/twitter-api>
- [12] tweepy, <http://www.tweepy.org>
- [13] 超音波距離センサモジュール HC-SR04,
<https://www.switch-science.com/catalog/1606/>
- [14] 一戸建て何でも質問掲示板, <https://www.e-kodate.com/bbs/thread/95418/>
- [15] 山崎昌廣, 佐藤陽彦“ヒトの歩行-歩幅,歩調,速度及びエネルギー代謝の観点から“,
https://www.jstage.jst.go.jp/article/ase1911/98/4/98_4_385/_pdf/-char/ja
- [16] Raspberry Pi に超音波距離計”HC-SR04”つけてみた,
<http://arkouji.cocolog-nifty.com/blog/2014/05/raspberry-pihc-.html>
- [17] Tweepy で雑に Twitter bot を作ってみた,
<https://qiita.com/tsc343/items/e51f412480ea8bf5619a>

付録

プログラムリスト

1. tweet.py

tweet.py は Twitter のタイムラインから特定のつぶやきを発見すると、距離センサを起動し、測定した値を Twitter に投稿するプログラムである。以下に tweet.py のプログラムリストを示す。

```
coding: utf-8
import sys
import time
import tweepy
from subprocess import check_output
# 取得したトークンを入れる
consumer_key = 'XXXXXX'
consumer_secret = 'XXXXXX'
oauth_token = 'XXXXXX'
oauth_token_secret = 'XXXXXX'
auth = tweepy.OAuthHandler(consumer_key,
consumer_secret)
auth.set_access_token(oauth_token,
oauth_token_secret)
api = tweepy.API(auth)
i = 0
t = 1;
while 400:
    statuses = api.user_timeline(count=1)
    for status in statuses:
        if "tell" in status.text:
            print("ok")
            num = 1
            while 400:
                retcode = check_output(["sudo","-
E","python","sensor.py"])
                print retcode
                time.sleep(1)
                num += 1
                if num % 2 == 0:
                    if "距離:" in retcode :
                        api.update_status(retcode +
str(t) + "回目のツイート")
                        t += 1
                    if num % 10 == 0 :
                        if "測定中" in retcode :
                            api.update_status(retcode +
```

```
str(t) + "回目のツイート")
                        t += 1
                    if "50cm" in retcode :
                        api.update_status(retcode +
str(t) + "回目のツイート")
                        sys.exit()
                if num == 150 :
                    print ("I can't search")
                    api.update_status(" 検出で
きませんでした¥n" + str(t) + "回目のツイート")
                    sys.exit()
            else:
                i += 1
```

2. sensor.py

sensor.py は距離センサが数値を測定する為のプログラムである。以下にsensor.py のプログラムリストを示す。

```
#!/usr/bin/python
# coding:utf-8
# remember to change the GPIO values below to match
your sensors
# GPIO output = the pin that's connected to "Trig" on
the sensor
# GPIO input = the pin that's connected to "Echo" on
the sensor
from subprocess import check_output
def reading(sensor):
    import time
    import RPi.GPIO as GPIO
    GPIO.setwarnings(False)
    GPIO.setmode(GPIO.BCM)
    if sensor == 0:
        GPIO.setup(17,GPIO.OUT)
        GPIO.setup(27,GPIO.IN)
        GPIO.output(17, GPIO.LOW)
        time.sleep(0.3)
        GPIO.output(17, True)
        time.sleep(0.00001)
        GPIO.output(17, False)
        while GPIO.input(27) == 0:
            signaloff = time.time()
            while GPIO.input(27) == 1:
                signalon = time.time()
                timepassed = signalon - signaloff
            distance = timepassed * 17000
            distance = int(distance)
            return distance
        GPIO.cleanup()
    else:
        print "Incorrect usonic() function variable."
        print "End"
if reading(0) > 2000:
    print ("測定中...")
elif reading(0) > 250:
    print ("距離:" + str(reading(0)) + "cm")
elif reading(0) > 50:
    print ("距離:" + str(reading(0)) + "cm")
else:
    print ("50cm 以内に反応がありました")
    print (str(reading(0)) + "cm")
```