

Prototype development of effective pass judgment device in Epee Fencing

| | |
|-------|--|
| メタデータ | 言語: jpn 出版者: 公開日: 2021-04-05 キーワード (Ja): キーワード (En): 作成者: 栗國, 晴楽, 西川, 哲夫 メールアドレス: 所属: |
| URL | https://mu.repo.nii.ac.jp/records/1497 |

フェンシング競技エペ種目における有効得点判定装置の試作

Prototype development of effective pass judgment device in Epee Fencing

栗 國 晴 楽¹

Aguni Seira

西 川 哲 夫²

Nishikawa Tetsuo

概要

フェンシング、特にエペ種目の「突き」の瞬間を認識しやすくするための可視化の方法、及びエペ種目にける「同時突き」の判定を、電気審判機を用いることなしに実施可能な方法の提案を目的として、「突き」の瞬間を LED によって可視化し、「同時突き」の判定を、電子ブザーの音の伝達で実現する試作装置を開発した。さらに、同時突きの要件である 40 ミリ秒以内の判定を実現するために、音読み取りのバッファサイズの最適化を測定時間とスペクトル波形の 2 つの観点から実施した結果、バッファサイズとして 128 が最適であることを見出だした。また、ラズベリーパイゼロを CPU として用いた同時突き判定システムを構築し、音取得のバッファサイズを 128 にして、ラズベリーパイゼロからの電圧を直接 LED に印加することで、同時突き判定が、 40 ± 10 ミリ秒程度の誤差で可能であることを、試作機による実験によって明らかにした。

1. はじめに

フェンシングの試合は、動きが高速かつ複雑なため、「突き」の様子が良く見えない場合が多く、それは観戦者、審判、選手の 3 者にとって問題であり、フェンシングの普及を妨げる大きな要素となっていると考えられる。フェンシングの中でもエペ競技は、日本ではあまり普及していないが、体中のどこをついてもよく、フルール競技の「攻撃権」のような複雑なルールがなく最も単純でわかりやすいため、世界的にみればフェンシングの 3 種目の中で最も普及している。海外では競技をエペからスタートさせる選手も多いといわれており、日本でもフェンシング競技の普及のために、分かりやすいエペを普及させていこうという動きがある[1]。

エペは全身が的であり、マスクや腕、足など、他の種目に比べて的が小さくかつ平坦でない部分があるため、突く位置や角度が少しでもずれば、エペ競技での得点に必要なとなる剣先への 750 g の圧力が不十分となり、「突き」自体が成立しない場合が生じやす

¹ 武蔵野大学工学部数理工学科

² 武蔵野大学工学部数理工学科特任教授 / 武蔵野大学数理工学科センター員

いと考えられる。電気審判機を設置すれば、有効な剣先の圧力がかかっているかを電気審判機で確認しながら競技を行うことができるが、電気審判機の設置には手間がかかるので、通常の練習時には電気審判機を用いないことも多い。そのため、「750gの圧力で剣先を押す」ということを確認できずに、練習を行っている場合が多いと考えられる。これはフェンシングにおいて最も重要な「突き」の有効性を高めるための練習が十分にできていないことを意味すると考えられる。

「突き」の有効性を高める練習には、「突き」の有効性を剣の近傍で可視化できることが有効であり、通常の電気審判機を用いることなしにそれができると望ましいと考えられる。「剣」の可視化や、「突き」の可視化については、これまでいくつか試みがある。「剣」の可視化については、特表 2019-522551 [2] において、剣のブレードに沿って光らせる試みがあるが、突いた時点で光るようにはなっていない。また、映像技術を駆使して剣の軌跡や剣が当たった瞬間を可視化する試み[3]もあるが、大掛かりな準備が必要であり、通常の練習に使えるものではない。

また、エペには、一定の時間内（40 ミリ秒）に両者の「突き」が有効であった場合、「同時突き」が認められるという独特のルールがあり[4]、通常のエペの試合において頻繁に用いられ、重要な戦術の一つとなっている。通常は、電気審判機を使って「同時突き」の判定を行っているが、より有効なエペの練習のためには、電気審判機を使わないで「同時突き」の判定もできることが望ましい。電気審判機を使わないで「同時突き」の判定を行う方法として、特開平 7-51425[5]に剣の間で超音波を用いて信号のやり取りを行う方法が開示されている。このような方法では信号のやり取りに要する時間を40 ミリ秒以内に収める必要があるが、この方法では、信号のやり取りに要する時間に関しての記述は見当たらない。

以上説明したように、本研究の目的は、視覚的に認識しにくいフェンシング、特にエペ種目の「突き」の瞬間を、審判や観戦者、選手にとって認識しやすくするための可視化の方法を提案し、またエペ種目において重要な戦術である「同時突き」の判定を、剣の間で信号をやり取りすることで、電気審判機を用いることなしに実施できる方法を提案し、エペ種目の手軽で有効な練習方法を実現することである。

この目的のために、「突き」の瞬間を LED によって可視化し、「同時突き」の判定を、電子ブザーの音によって伝えることで実現する装置の試作を行った。さらに、同時突きの要件である 40 ミリ秒の判定の実現のために、剣の間での信号伝達に関わるパラメータ、特に音の読み取り時のバッファサイズの最適化を実施した。判定誤差としては、練習用という用途から、 $\pm 2 \sim 3$ 割程度、ここでは、 ± 10 ミリ秒以内の精度を目指した。

2. 試作装置の開発

2. 1 試作装置の構成と動作

本試作装置は、図 1 に示すように、フェンサー 1 が持つ剣 1 のブレード及び「つば」の内側に装着されたシステム 1 と、フェンサー 2 が持つ剣 2 のブレード及び「つば」の内側に装着されたシステム 2 の 2 つのシステム 2 からなる。システム 1 は、「突き」に伴う LED 発光、及びブザー発音システム (S11)、及び「同時突き」判定のための「突き」

フェンシング競技エペ種目における有効得点判定装置の試作（栗國、西川）
 情報処理システム（S12）から構成され、システム2は、フェンサー2が持つ剣2のブレード及び「つば」の内側に装着された、「突き」に伴うLED発光、及びブザー発音システム（S21）、及び「同時突き」判定のための「突き」情報処理システム（S22）から構成される。

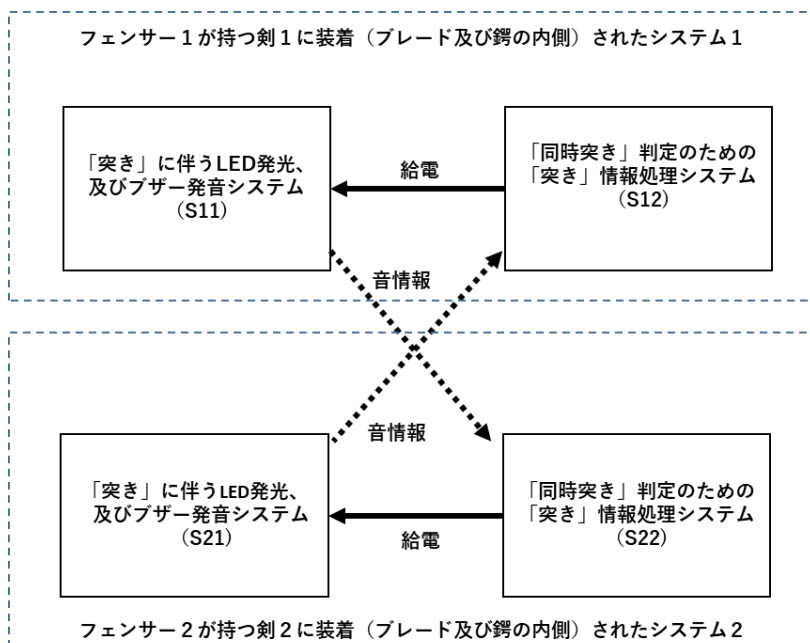
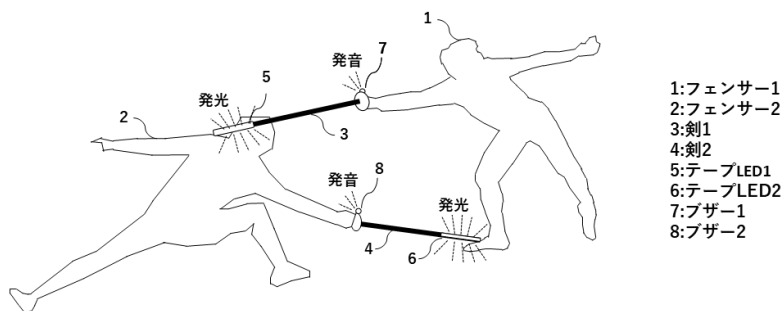


図1 試作装置の概要図

A) 「突き」の時間間隔が40ミリ秒未満の場合（「同時突き」）



B) 「突き」の時間間隔が40ミリ秒以上の場合（「同時突き」ではない）

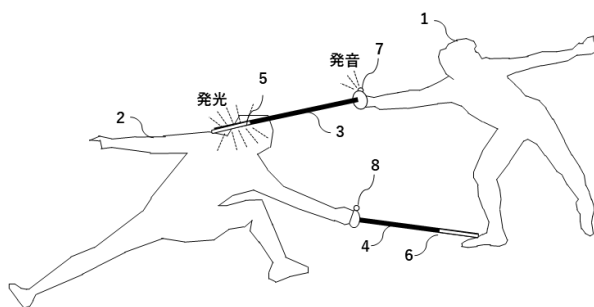


図2 フェンシングエペ種目における「突き」の可視化と「同時突き」の判定

図1のシステムの動作の概要を示す。まず、フェンサー1、フェンサー2それぞれのシステムにおいて、初期状態では、S11、S22内のLEDとブザーに対して、それぞれS12、S21から給電されており、「突き」が発生すると、突いた剣のLEDが発光しブザーが発音する。フェンサー1の剣の突きが発生すると、その際のS11からのブザーの発音は、もう一方（フェンサー2）のS22内のマイクで検出され、それから40ミリ秒経過するとS22からのS21への給電が停止される。

すなわち、図2A)に示すように、フェンサー1の剣の「突き」から40ミリ秒未満の間は、フェンサー2の剣の「突き」でも発光と発音が生じるが、図2B)に示すように、40ミリ秒以上が経過するとフェンサー2の剣の「突き」は発光と発音を生じなくなる。フェンサー2の「突き」が入った場合も同様に動作する。このようにして、「同時突き」の可視化による判定が、電気審判機無しでかつ剣の近傍で可能になる。

2.2 「突き」に伴うLED発光、及びブザー発音システム

1) システムの構成と動作

図3に、「突き」に伴うLED発光、及びブザー発音システムの構成図を示す。図3に示すように、「つば」の内側に設置されたLED・ブザー電源からの電圧は、剣先スイッチを介して、剣先に張られたテープLEDと剣の「つば」の内側に設置されたブザーに並列に印加される。

テープLEDとブザーのマイナス極からの導線は、「同時突き」判定のための「突き」情報処理システム内のリレーを介してLED・ブザー電源のマイナス極に戻る。この通常の状態では、リレーは閉じているので、この状態で「剣」が突かれ剣先スイッチがONになると、同じ電圧がテープLEDとブザーに印加され、テープLEDは発光し、ブザーは発音する。

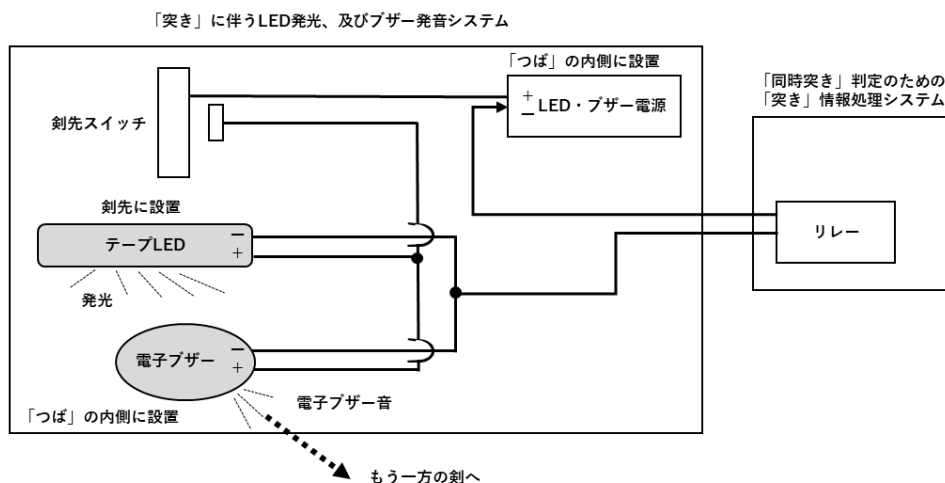


図3 「突き」に伴うLED発光、及びブザー発音システム

以上の方法では、剣で突いているあいだのみ発光と発音が生じるが、これは生徒がレッスンを受けている状況に適した方法である。レッスンの際には、有効な突きがはいったかどうか重要であるから、突いた瞬間のみの発光と発音で十分である。一方、練習試合にこの剣を用いる場合は、「突き」の後数秒間、発光と発音が持続するようにすれば、審判や観衆・及び選手自身にとって、どちらが突いたかがより分かりやすくなると考えられる。ここでは示さないが、タイマー回路とリレー回路を組み合わせることで実現可能である。

「同時突き」判定のための「突き」情報処理システムで、もう一方の剣からの発音を受信して40ミリ秒経過すると、リレーが閉じてLED・ブザー電源の電圧がテープLEDとブザーに印加されなくなる。すなわち、剣が突かれてもテープLEDの発光とブザーの発音が生じなくなる。

2) システムの実装

図4に「突き」に伴うLED発光、及びブザー発音システム内の各構成要素の剣への実装の様子を示す。図4A)に示すように、つば内に電子ブザーとLED・ブザー電源を設置する。電子ブザーとしては、秋月電子通商の電子ブザー(24mm P K B 2 4 S P C H 3 6 0 1)、LED・ブザー電源として12Vの電池を使用した。図4B)に示すように、エペ競技用の剣の剣先に60cmの長さの青色テープLEDを両面テープによって固定した。電気剣としては、エペ用電気剣(フレンチ標準タイプ右利き用 ウクライナ)を用い、テープLEDとしては、エーモン トップビューテープLED 青・60cmを使用した。

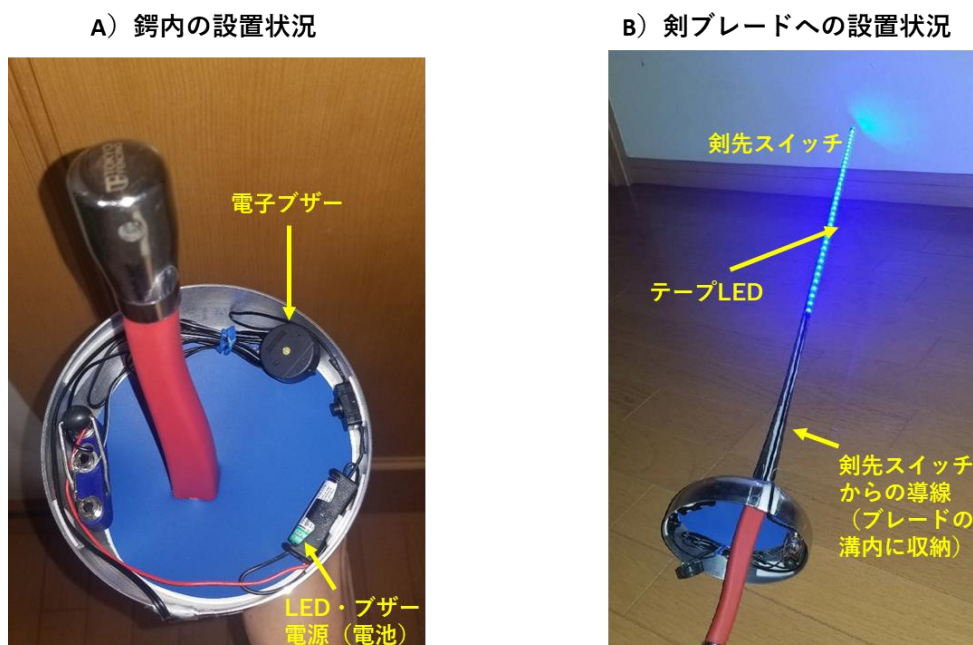


図4 「突き」に伴うLED発光、及びブザー発音システムの実装

2. 3 「同時突き」判定のための「突き」情報処理システム

1) システムの構成と動作

図5に、「同時突き」判定のための「突き」情報処理システムの構成図を示す。図5に示すように、本システムに、もう一方の剣からの電子ブザー音の音情報が届くと、システム内のUSBマイクで音が検出されデジタル信号に変換される。その後、CPU内で音入力が行われた後、高速フーリエ変換され、波形判断ユニットによって電子ブザーの音かどうかが判断される。電子ブザーの音以外のノイズによる波形判断の誤りをなくすために、波形判断に加えて音の大きさを判断の指標に加えることが考えられる。このことによって、隣のピストからの音による誤判定をなくすことも可能である。

波形が電子ブザーの音であると判断されたら、タイマーをスタートさせる。タイマースタート後一定時間が経ったら、リレーを作動させて、もう一方の剣内にあるLED・ブザー電源回路を遮断する。これによって、もう一方の剣は、突きが入っても発光と発音が生じなくなる。タイマー時間の設定は、測定開始からの時間が40ミリ秒になるように調整される。その詳細は、次節で詳しく述べる。

以上のように、一方の剣の「突き」から40ミリ秒未満の間は、もう一方の剣の「突き」でも発光と発音が生じるが、40ミリ秒以上が経過するともう一方の剣を突いても発光と発音が生じなくなる。このようにして、剣どうしで音の情報を伝達することで、「同時突き」の判定が、電気審判機無しでかつ剣先を見ながら可能になる。

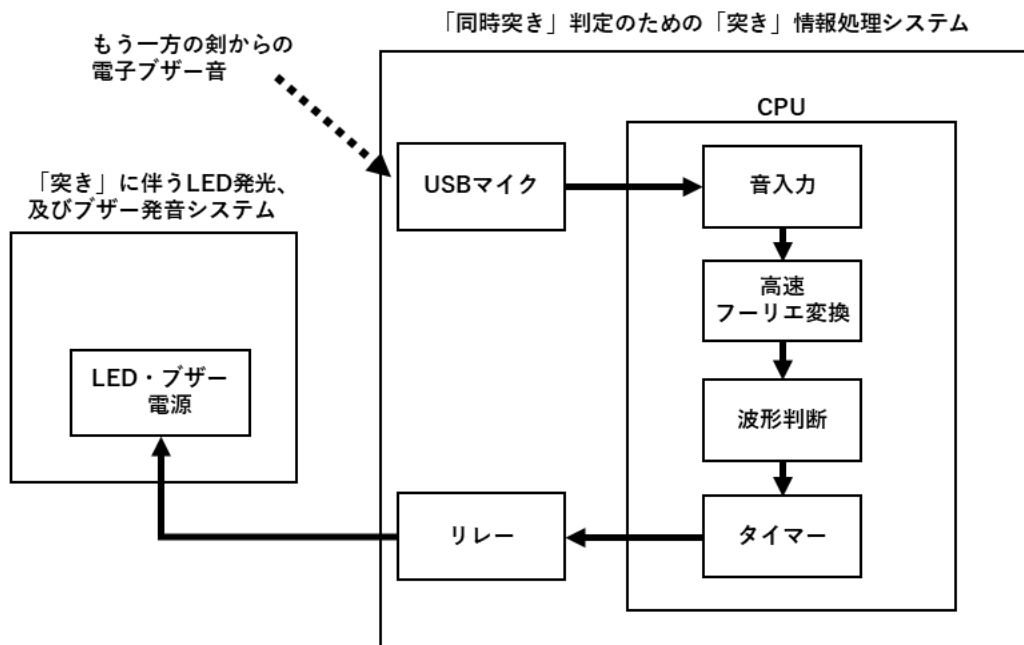


図5 「同時突き」判定のための「突き」情報処理システム

2) システムの実装

図6に「同時突き」判定のための「突き」情報処理システム内の各構成要素を実装し、回路を構成した例を示す。図6には、CPUとしては、シングルボードコンピュータであるRaspberry Pi Zero（ラズベリーパイゼロ）を用いて、USBマイク（Buauty

PC マイク世界最小 USB マイク）とリレー（WayinTop リレーモジュール）を接続して、リレーから赤色のテープ LED に接続した様子を示す。図 6 の下部には剣に設置された青色 LED を置いている。

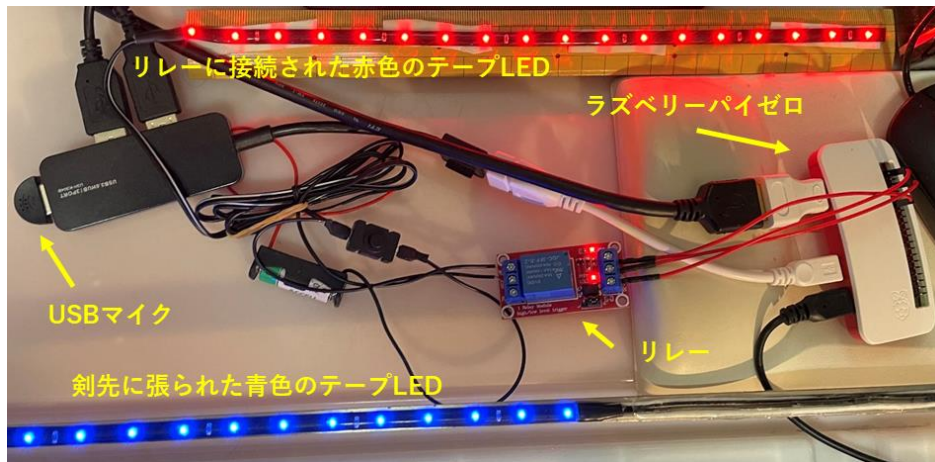


図 6 「同時突き」判定のための「突き」情報処理システムの実装

図 7A) に、システムの構成要素（ラズベリーパイゼロ、USB マイク、電子ブザー）を鍔の内側に収納した様子を示す。それぞれは、両面テープでとめており、ラズベリーパイゼロのサイズと重量は、79mm×38mm×12mm、25g である。このように、充電電池を除いて鍔の内側に無理なく収納が可能である。図 7B) に、試作装置を手を持った様子を示す。人差し指を最も深い位置におくことができ、グリップを握った指のどこも設置物に接触することなく、重量も充電電池を除いて 30g 程度であり（エペ用の剣の重量は最大 770g）、通常通りの剣さばきが可能であることがわかった。充電電池については、現状では鍔の内側には収納できないので、十分長い USB コードを用意し、袖を通して腰のあたりに装着することを考えている。充電電池の小型化は今後の課題である。

A) 試作装置の鍔への収納



B) 試作装置を持った様子



図 7 試作装置の鍔の内側への収納と手に持った様子

3) 音情報処理のアルゴリズム

図8に、「同時突き」判定のための「突き」情報処理システムの処理の流れ図を示す。図8は、図5におけるUSBマイクによる音入力から、リレーの動作までの処理を流れ図で示したものである。OSはLINUXのRasverianでプログラミング言語Python3.7を用い、音情報の処理にはPyaudioライブラリーを用いた。

図8に示すように、まず、音入力の前に①時刻 t_1 を取得する。次に、②電子ブザーで音を入力し、デジタル化を行い、③入力音データのリストへの格納をバッファ単位で行う。音を取り込む際のパラメータとして、バッファサイズをまず1024に設定し、サンプリングレートを44,100と設定した。サンプリングレートは、Windows PC上では約300,000にまで高い値が設定できたが、Raspberry Pi Zeroではそれができず、44,100固定の値で測定を行った。バッファサイズは測定時間と波形の質に大きく関係するので、この後で最適化の検討を行う。

その後、④高速フーリエ変換によって、音のフーリエスペクトルを取得した。図9に、電子ブザーを鳴らしたときに得られたフーリエスペクトルを示す。このように、約3445Hzに単独のピークが現れることが分かる。このフーリエスペクトルを用いて⑤ブザー音判定を行う。このピークから電子ブザーの音であることを判定するためには、スペクトルの最大値を与える周波数を検出して、それが3445Hz付近にあることを示せば良いと考えられる。また、前述したように、電子ブザーの音以外のノイズによる波形判断の誤りをなくすために、波形判断に加えて音の大きさを判断の指標に加えた方がよい。

そこでまず、⑤-1 フーリエパワースペクトラムの最大値 P_m と、 P_m を与える周波数 F_m を取得し、⑤-2 ($F_m > 3450$) and ($F_m < 3460$) and ($P_m \geq P_t$) の条件で波形判断を行う。この条件を満たさない場合は、①時刻 t_1 の取得にもどり、満たした場合は、⑤-4 時刻 t_2 を取得し、⑤-5 計測時間 $t_m = t_2 - t_1$ を取得する。

次に、⑥同時突き判定のための待ち処理を行う。ここでは、⑥-1 待ち時間 $t_w = t_s - t_m - t_r$ を取得する。ここで、 t_s は、同時判定時間 (=40ミリ秒)、 t_r は、リレー動作時間であるが、この時点では未知であり5ミリ秒と仮定した。得られた t_w が正の数であれば、次に⑥-2 t_w 秒間の待ち処理を行う。 t_w が負の数になると同時判定時間の40ミリ秒を超えてしまうことになるので、待ち処理は行わない。この後、リレーを作動させることで、⑦剣先設置LEDへの給電を停止する。

t_m の値が実際どの程度であるのかによって、ここの処理に大きく影響すると考えられるので、次の節では、 t_m に関するバッファサイズの最適化の検討を行った。図10に、同時突き判定プログラムのPythonによる実装プログラムを示す。

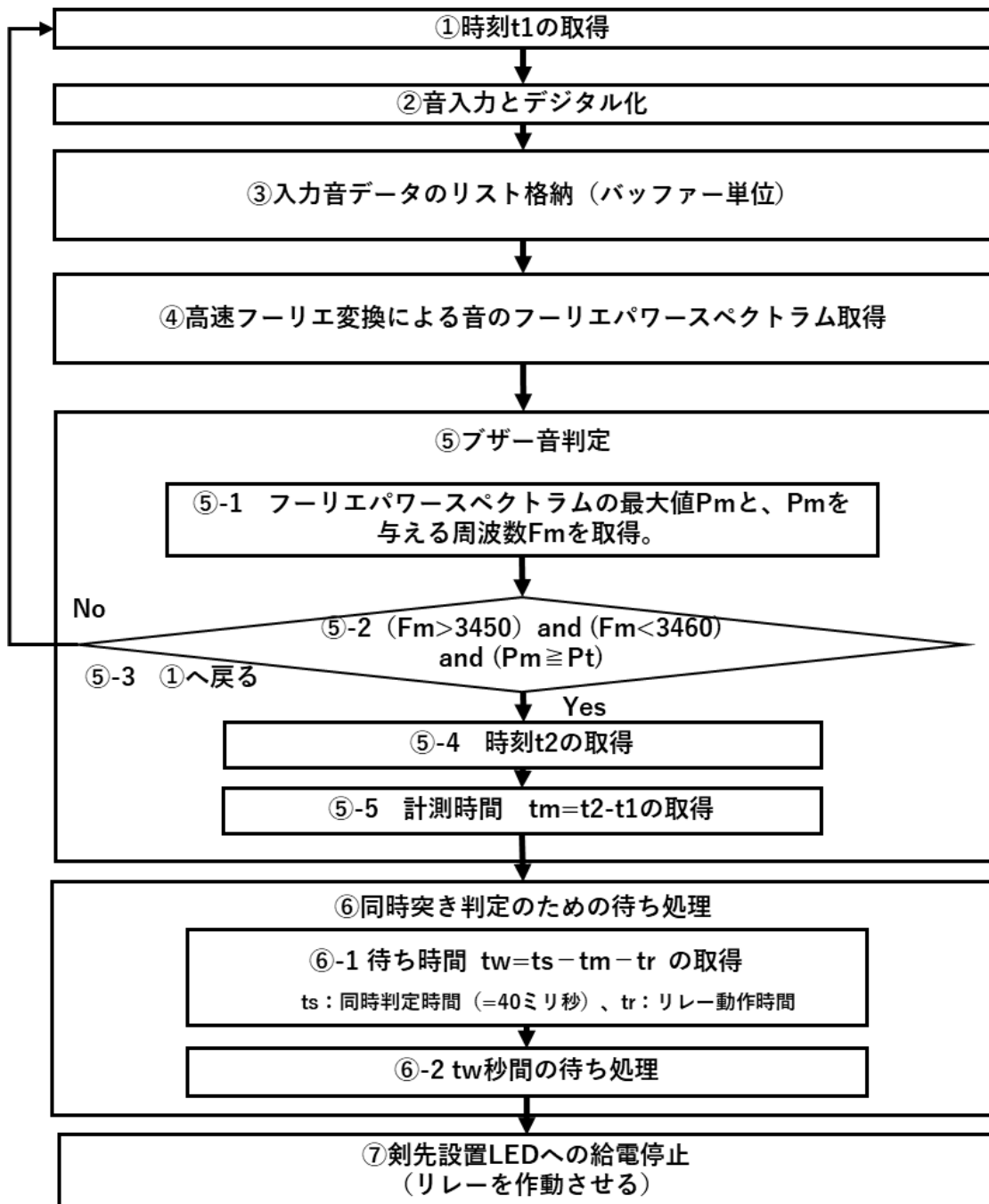


図8 「同時突き」判定のための「突き」情報処理システムの処理の流れを表す図

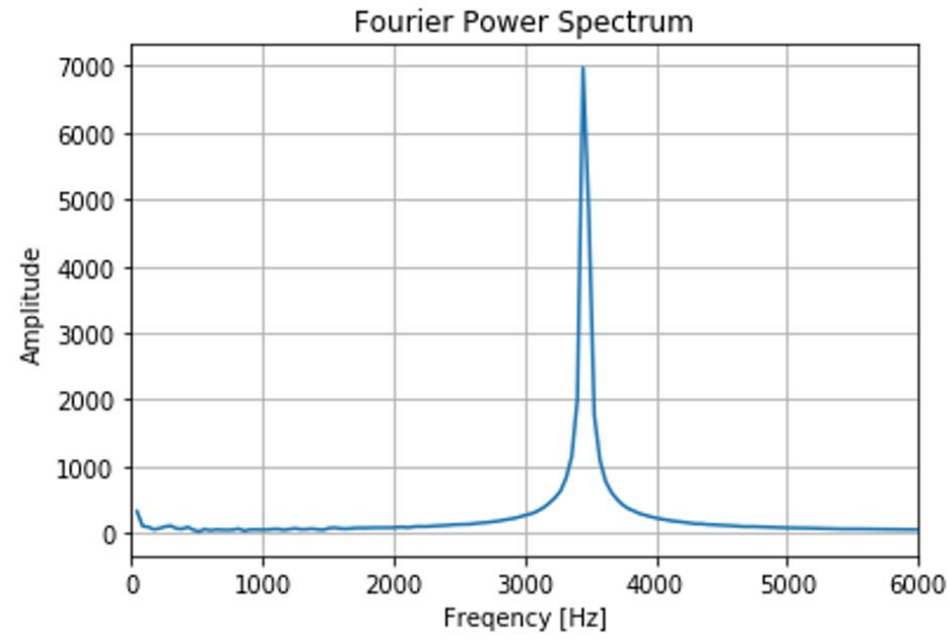


図9 電子ブザーを鳴らしたときに得られたフーリエスペクトル

```

#実音の音声受信プログラム-ラズパイゼロ-L.py
# 2021-01-16-1
# プログラム (例の電子ブザーの音を検出して、一定時間後にリレーを動作させ、LEDを点灯させる。)
#
#音声関係のライブラリ
import pyaudio
import struct
#プロット関係のライブラリ
import matplotlib.pyplot as plt
import numpy as np
import sys
#時間のライブラリ
import time
from time import sleep
#数学・統計のライブラリ
import statistics
import math
import RPi.GPIO as GPIO

#*****
RELAY_PIN = 17
GPIO.setmode(GPIO.BCM)
GPIO.setup(RELAY_PIN, GPIO.OUT)

# GPIO.output(RELAY_PIN, False)

chunk = 128
FORMAT = pyaudio.paInt16
CHANNELS = 1
RATE = 44100
RECORD_SECONDS = 1
p = pyaudio.PyAudio()

#入力ストリームのオープン
stream = p.open(format = FORMAT,
                channels = CHANNELS,
                rate = RATE,
                input = True,
                frames_per_buffer = chunk)

#測定開始
start_time1 = time.perf_counter()
elapsed_time2=[]
elapsed_timefft=[]
mx = 0
n = 0
while stream.is_active():
    n += 1
    start_time2 = time.perf_counter()
    #inputにバッファ単位で入力オーバーフローしない限り
    if n == 1:
        input = stream.read(chunk, exception_on_overflow=False)
        #測定開始時刻を取得する
        ndarray = np.frombuffer(input, dtype='int16')
        #inputから、16bitずつndarray(配列)に変換
        start_timefft = time.perf_counter()
        #測定開始時刻を取得する
        f = np.fft.fftn(ndarray)
        #ndarrayを高次元フーリエ変換して時間領域から周波数領域にする()
        N = int(chunk)
        dt = 1/RATE
        #フーリエ変換に差するデータ数(=サンプルサイズ)
        #サンプリング間隔(秒)
        ffreq = np.fft.ffreq(N, dt)
        #周波数(リスト)
        Amp = np.abs(f/(N/2))
        #振幅(リスト)
        mx = np.max(Amp)
        #振幅の最大値
        index = np.argmax(Amp)
        #振幅の最大値を与える添え字番号 f = np.fft.fftn(ndarray)
    end_time = time.perf_counter()
    #サンプル取り終った時刻を取得する
    
```

```

elapsed_time1 = end_time - start_time1
elapsed_time2 = end_time - start_time2
elapsed_timefft = end_time - start_timefft
#トータル実行時間を計算する
#1読み取り処理(1バッファ)読み取り出たりの実行時間を計算する
#1出たりの実行時間を計算する
#読み取り処理時間リストに追加する
#時刻リストに追加する
max.append(mx)
#スペクトル振幅最大値リストに追加する
print("最大振幅がブザーの周波数に一致した！", "index=", index, "最大振幅=", mx, "elapsed_time1=", elapsed_time1, "elapsed_time2=", elapsed_time2, "elapsed_timefft=", elapsed_timefft)
print("スペクトル振幅最大値=", index, "elapsed_time1=", elapsed_time1, "elapsed_time2=", elapsed_time2, "elapsed_timefft=", elapsed_timefft)
break

GPIO.output(RELAY_PIN, True)
sleep(1)
GPIO.output(RELAY_PIN, False) # off
tm = elapsed_time2
ts = 0.04
tr = 0.005
tw = ts - tr - tm

print("tm=", tm, "tw=", tw)

elapsed_time2_m = statistics.median(elapsed_time2)
elapsed_timefft_m = statistics.median(elapsed_timefft)
elapsed_time2_s = statistics.mean(elapsed_time2)
elapsed_timefft_s = statistics.mean(elapsed_timefft)

print("Measurements finished.")
print("RATE=", RATE)
print("chunk=", chunk)

#計測時間の出力
print("Number of Measurements=", 20)
print("elapsed_time2_s=", elapsed_time2_s)
print("elapsed_timefft_s=", elapsed_timefft_s)
print("Number of Measurements=", 20)
print("elapsed_time2_m=", elapsed_time2_m)
print("elapsed_timefft_m=", elapsed_timefft_m)

t2_max = np.max(elapsed_time2) * 1.1
plt.title("Measurement Time Distribution")
plt.xlabel("Time [sec]")
plt.ylabel("Frequency")
binwidth = 0.001
bins = np.arange(0, max(elapsed_time2) + binwidth, binwidth)
plt.hist(elapsed_time2, range=(0, t2_max), bins=bins, color='red')
plt.grid(True)
plt.show()

plt.title("Measurement Time Cumulative and Normed Distribution")
plt.xlabel("Time [sec]")
plt.ylabel("Frequency")
plt.yticks(np.arange(0, 1, 0.05))
plt.hist(elapsed_time2, range=(0, t2_max), bins=bins, color='orange', cumulative=True, density=True)
plt.grid(True)
plt.show()

stream.stop_stream()
stream.close()
p.terminate()
#入力終了
#入力開始
#pyaudio終了
    
```

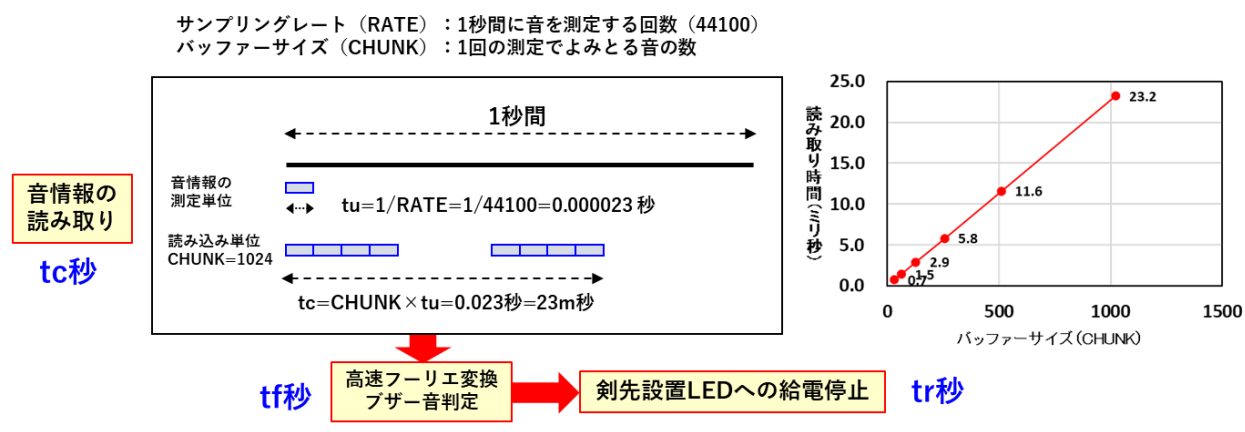
図10 同時突き判定プログラム (Pythonによる実装プログラム)

3. 音情報測定方法の最適化

3. 1 音情報測定の必要条件

先述したように、バッファサイズは測定時間と波形の質に大きく関係する。その理由とバッファサイズ最適化の重要性を、図 11 を用いて説明し、バッファサイズ最適化をいかに行うかを述べる。図 11 に示すように、音測定のサンプリングレートを RATE (=44100)、音読み込み時のバッファサイズを CHUNK (=1024) とすれば、音情報の測定単位時間 t_u は、 $t_u=1/\text{RATE}=0.000023$ 秒となり、バッファサイズ分の読み込み単位時間 t_c は、 $t_c= \text{CHUNK} \times t_u=0.023$ 秒=23 ミリ秒となる。音を入力してから給電停止までのトータル処理時間 t_p は、 $t_p =t_c + t_f + t_r$ で与えられる。ここで、 t_f は高速フーリエ変換とブザー音判定の時間であり、 t_r は、剣先設置 LED への給電停止に要する時間である。リレーを用いる場合は、 t_r はリレー反応時間となる。

本試作装置が、同時突き判定を実現できるためには、少なくとも $t_p < t_s$ (=40 ミリ秒) が成り立つ必要がある。また、「はじめに」で述べたように、判定時間の誤差の精度として ± 10 ミリ秒程度を達成するためには、トータル処理時間自体が 10 ミリ秒程度以内 ($t_p \leq 10$ ミリ秒) である必要があると考えられる。先ほど得られたバッファサイズ=1024 における $t_c=23$ ミリ秒では、この条件を満たしていない。



トータル処理時間 $t_p = t_c + t_f + t_r$ への要請

必要条件 $t_p < 40\text{m秒} (=t_s)$
 誤差 (± 10 ミリ秒以内) のためには、 $t_p \leq 10\text{m秒}$

- 右上のグラフより、理論的には、バッファサイズを256程度にすれば t_c を5m秒程度にできると考えられる。これが本当に成り立つのかを確認する必要がある。
- しかし、バッファサイズを小さくすると、高速フーリエ変換後のスペクトルの周波数分解能と S/N 比が小さくなり、ブザー音の誤判定が増加すると考えられる。

- 実施事項
- 1) 音測定時間 $t_m (=t_c + t_f)$ 分布のバッファサイズ依存性の測定
 - 2) ブザー音ピークスペクトル形状のバッファサイズ依存性の確認

図 11 音情報測定時のバッファサイズの最適化

図 11 内のグラフに示すように、 t_c はバッファサイズに比例しているため、バッファサイズを 256 程度にすれば t_c を 5 ミリ秒程度にできると考えられ、 $t_p \leq 10$ ミリ秒を満たす可能性もあると考えられる。このように、バッファサイズを小さくすれば、音の読み取り平均時間を小さくできると期待されるが、読み取り時間のばらつきがあると考えられるので、どの程度あるかを実験によって確認する必要がある。

また、バッファサイズを小さくすると、高速フーリエ変換に用いるデータの数が少なくなるので、得られたスペクトルの周波数分解能と S/N 比が小さくなり、ブザー音の判定制度が悪化するものと考えられる。そこで、測定時間とスペクトル測定の 2 つの観点から以下の検討を行い、バッファサイズの最適化を行った。

- 1) 音測定時間 $t_m (=t_c+t_f)$ の分布についてのバッファサイズ依存性の測定
- 2) ブザー音フーリエパワースペクトル形状のバッファサイズ依存性の確認

3. 2 バッファサイズの最適化

1) 音測定時間 $t_m (=t_c+t_f)$ の分布のバッファサイズ依存性の測定

① t_f の分布の検討

まず、高速フーリエ変換の時間 t_f の分布を測定した。図 12 に 1000 回の音の測定を繰り返した際の t_f のヒストグラムを、バッファサイズを 1024、512、256、128、64 と変えて示した。図 12 からバッファサイズ=1024 を除いて最頻区間は 3 ミリ秒の付近にあるが、非常に少数ではあるが 15 ミリ秒以上の区間にも存在することがわかる。

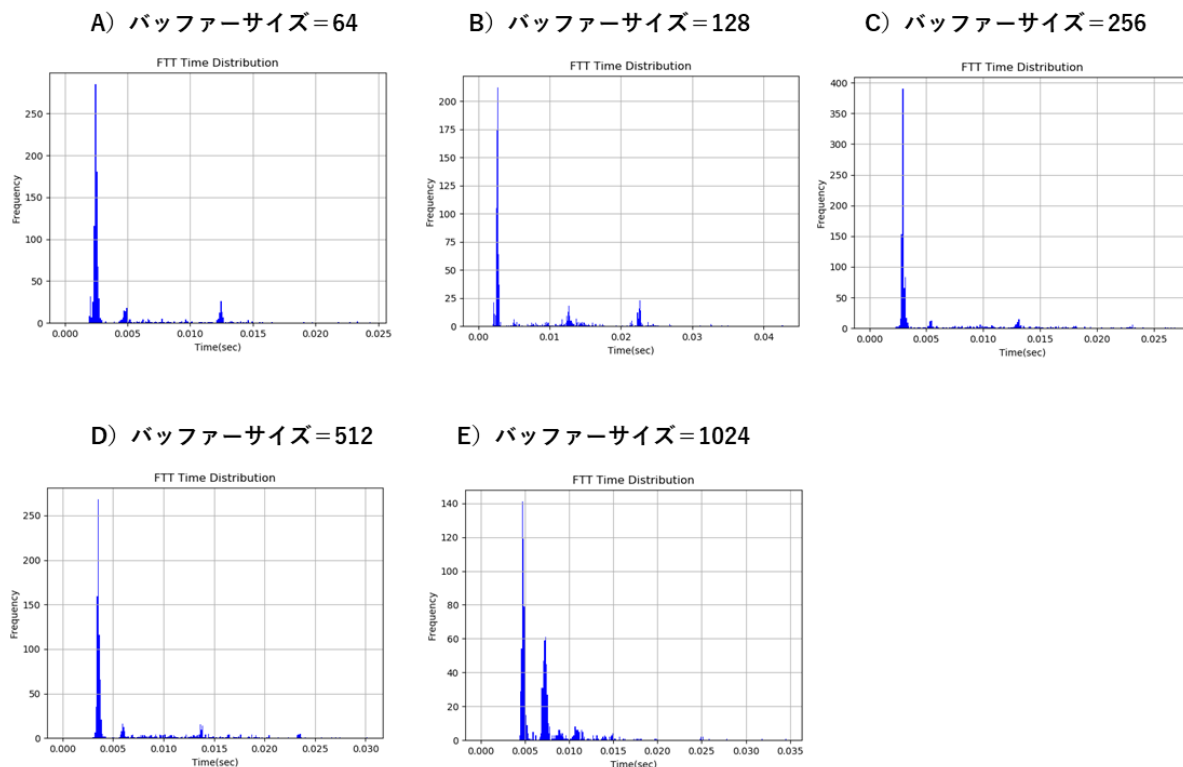


図 12 高速フーリエ変換時間のヒストグラムのバッファサイズ依存性

②tc の分布の検討

次に、音測定時間 t_c の分布を測定した。図 13 に 1000 回の音の測定を繰り返した際の t_c のヒストグラムを、バッファサイズを 1024、512、256、128、64 と変えて示した。図 13 より、 t_f よりも、 t_c のばらつきが大きいことがわかる。 t_c の平均値でみると数ミリ秒～20 ミリ秒ぐらいであり、理論的な値とオーダー的に等しいが、ばらつきが相当大きいことがわかる。

ばらつきの分布は、バッファサイズが小さくなると、次第に値の小さいほうに分布がずれてはくる。しかし、ばらつきは均等なガウス分布的なばらつきではなく、とびとびのピークがどのバッファサイズでも観察される。最頻値はどのバッファサイズでも 5 ミリ秒のあたりにあるが、13 ミリ秒や 23 ミリ秒、45 ミリ秒、63 ミリ秒の付近にもピークが存在する。

このばらつきの特徴は、統計的なばらつきではなく何らかの系統的なばらつきではないかと考えられる。その原因は明らかではないが、汎用的なコンピュータを CPU として用いているため、同時に実行している job が影響している可能性もあると考えられる。いずれにせよ、このように分布の幅が大きいと、平均値のみで議論することはできない。

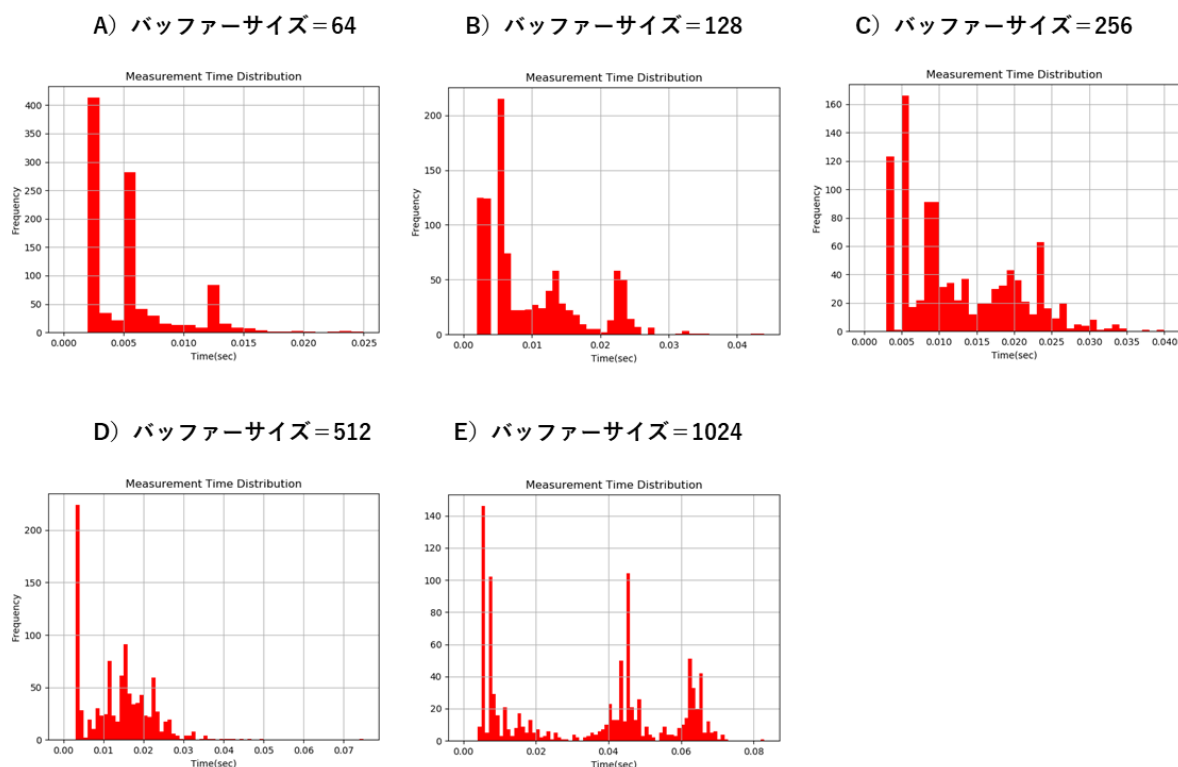


図 13 音取得時間のヒストグラムのバッファサイズ依存性

③ t_c の積算比率分布の検討

そこで、音測定時間のばらつきを統計的に扱うために、図 14 のように、音取得時間の累積比率ヒストグラムを求め、そのバッファサイズ依存性を調べた。このヒスト

グラムを用いると、例えば、C) バッファサイズ=256 において、1000 回の測定で、100%の測定をある時間内に収めようとする、その時間は 35 ミリ秒となるが、95%の測定でよしとすれば、その時間は 25 ミリ秒となることがわかる。

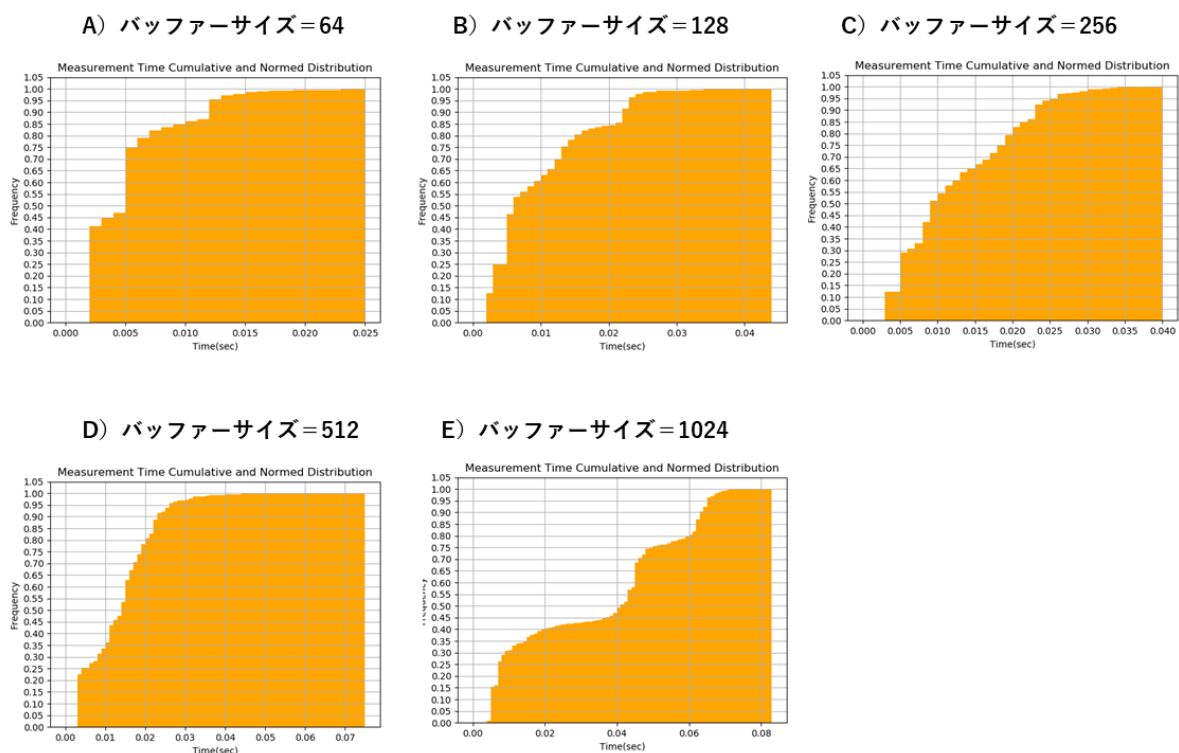


図 14 音取得時間の累積比率ヒストグラムのバッファサイズ依存性

この見方を 1 枚のグラフでわかりやすく表現するために、各バッファサイズで、許容する比率 (%) を 80%、85%、90%、95%、100%と変えて、その時の測定時間を読み取った情報を、図 15 に示すように、比率 (%) 毎の、測定時間のバッファサイズ依存性の形のグラフで表現した。

図 15 から、比率が 100%の場合は、測定時間が 25 ミリ秒以内である条件は、バッファサイズが 64 の場合しかないとわかる。この場合は、比率が 95%以下で測定時間が 12 ミリ秒、80%では 7 ミリ秒になる。従ってもシスプレットの波形上問題がなければ、バッファサイズ 64 が最適である。

バッファサイズ 64 がもし波形上問題があれば、バッファサイズ 128、256、512 で考えるが、比率 85%~95%では、バッファサイズが変わっても測定時間はあまり変わらない。比率 80%で見ると、バッファサイズ 128 だけが 15 ミリ秒となり、他のサイズよりも優位に小さくなっている。以上のことから、バッファサイズ 128、256、512 の中では、バッファサイズ 128 が最適であると考えられる。

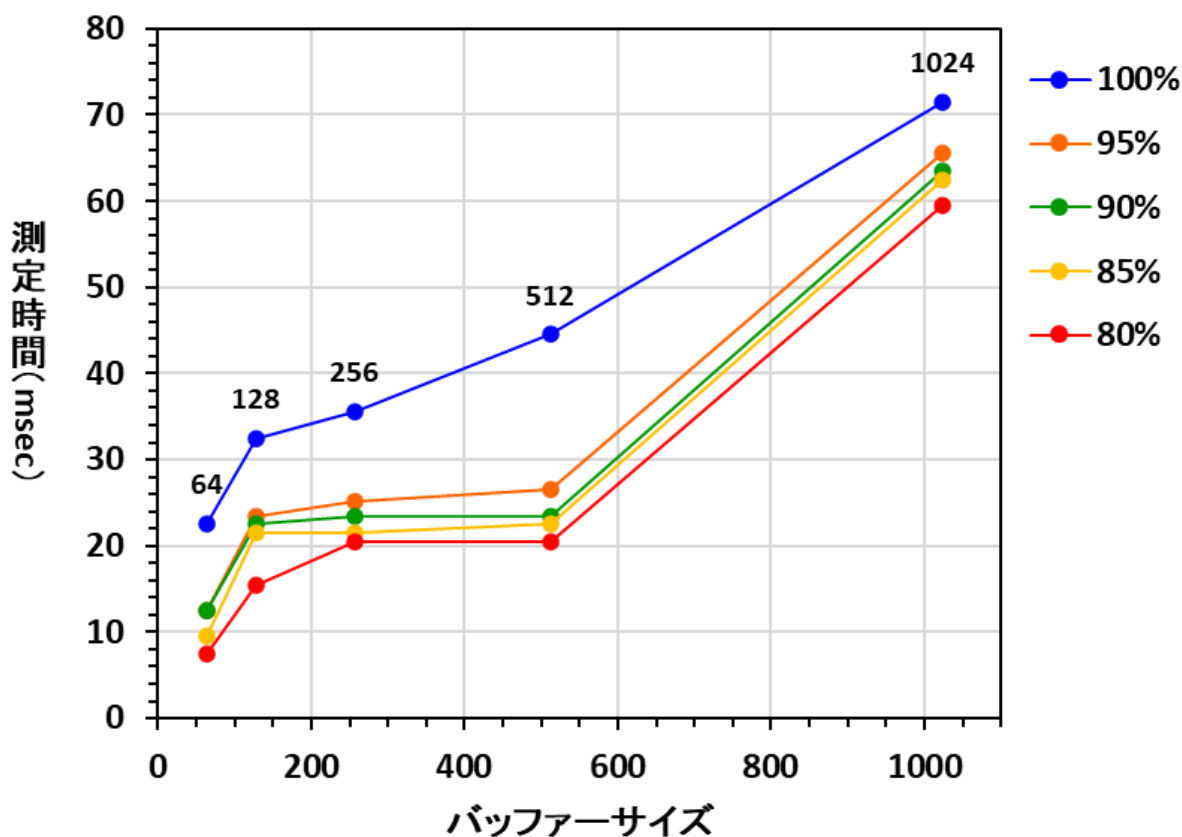


図 15 音取得時間累積比率毎の測定時間のバッファサイズ依存性

2) ブザー音フーリエスペクトル形状のバッファサイズ依存性

図 16 に、電子ブザー音のフーリエスペクトルを、バッファサイズ毎に示した。フーリエスペクトルの形状は、バッファサイズが大きいほどシャープであり、バッファサイズが小さくなると次第にピーク幅が広がっていきることがわかる。ピーク最大値の検出には、ピークがシャープであり、ノイズが小さいことが必要であるが、バッファサイズ 1024 から 128 までは、ピークは十分シャープであり、誤検出なく再現よくピーク検出が可能であった。

一方、バッファサイズ 64 では、ピーク幅の拡大が大きく S/N も小さくなっており、時折誤検出も見られたことから、バッファサイズは 128 以上にすることが望ましいと判断した。このことと、前節での結論「バッファサイズ 128、256、512 の中では、バッファサイズ 128 が最適」から、トータルでの最適なバッファサイズは 128 であると結論づけられた。

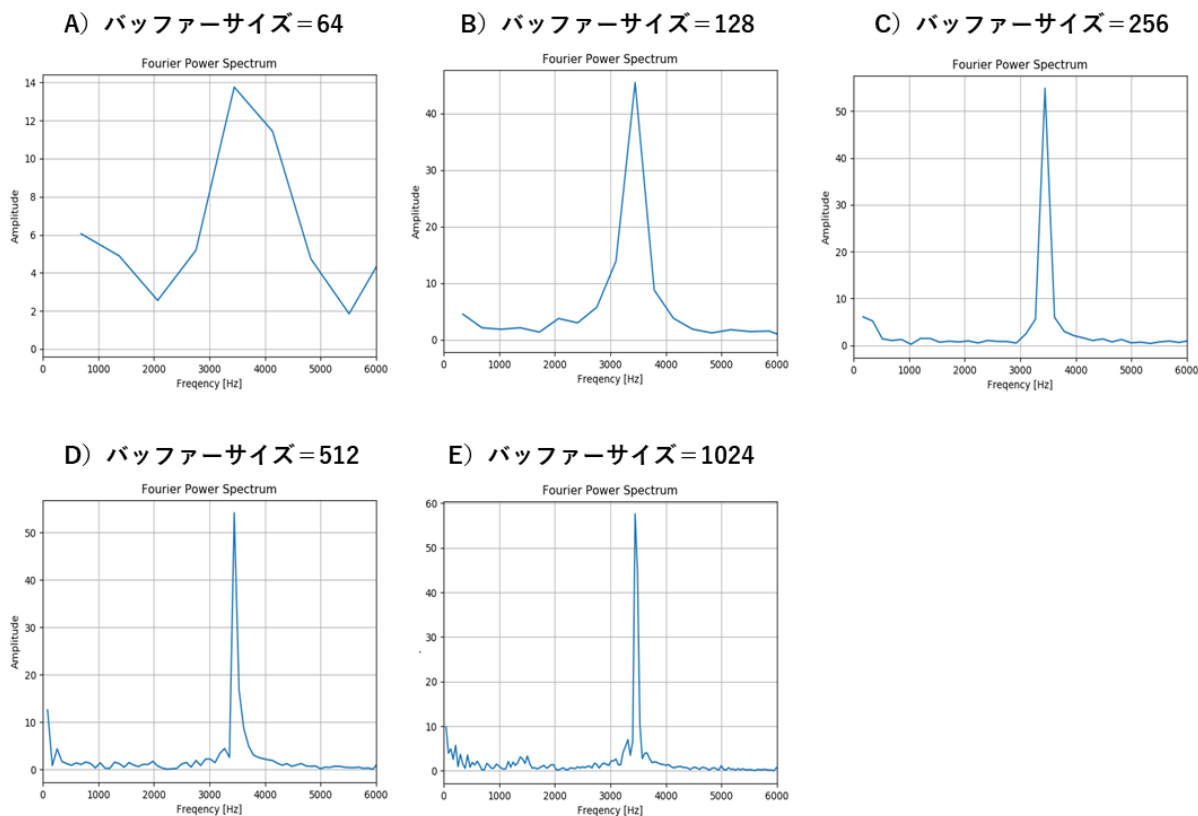


図 16 フーリエスペクトルのバッファサイズ依存性

3. 3 音情報測定時間の実測による評価

前節で得られたバッファサイズとして最適値である 128 を用いて試作機を動作させ、同時突き判定時間の評価を行った。

1) リレーを介して LED を発光させた場合

図 17 に、ラズベリーパイゼロからリレーを介して赤い LED を発光させた場合の同時突き判定時間の評価結果を示す。評価は、以下のように行った。ラズベリーパイゼロからリレーを介して赤い LED に接続し、フーリエ変換後のブザー音判定の後、待ち処理を実施せずにリレーを ON にした。青いテープ LED を剣に張り、剣先スイッチを ON にすると、青いテープ LED が発光すると共にブザーが発音する。ブザーの音が検知されて、その後のリレーによって赤い LED が発光する。

この様子(剣先スイッチを ON にする前から、赤いテープ LED が光った時点まで)を、iPhone 付属のカメラのスローモード (240fps) で撮影した。図 17 には、A) 青いテープ LED が光る直前、B) 青いテープ LED が光った時点、c) 赤いテープ LED が光った時点とそれぞれの時刻を示している。B) と C) の時間差を求めてみると、0.084 秒=84 ミリ秒であった。この場合の t_m は約 5 ミリ秒であったことが分かっているので、これは、リレーの反応時間が約 80 ミリ秒であったことが推定される。

今回、用いたリレーは有接点リレーであるが、ラズベリーパイゼロからの 3.3V の信号電圧の供給で動作する小型のリレーとして、使用例がサイトに記載されていたこともあり、最初の試用として用いることとした。有接点リレーの反応時間は、あるサイトでは 3ms~5ms [6] とあったが、用いたリレーの反応時間の仕様は記載されていなかったため、このような長い反応時間となったことの原因は明らかではない。さらに各種の有接点リレーと共に無接点リレーを用いた反応時間の検討が必要であると考えられる。ここでは、有接点リレーを用いずに、ラズベリーパイゼロからの電圧を直接 LED に印加する方法の評価を行った。



図 17 試作機による同時突き判定時間の評価
 (ラズベリーパイゼロからリレーを介して LED を発光させた場合)

2) 直接 LED 接続して発光させた場合

図 18 に、ラズベリーパイゼロから直接 LED 接続して発光させた場合の同時突き判定時間の評価結果を示す。今回は、ラズベリーパイゼロの電源からの電圧を直接 LED に印加して、点状の LED を発光させた (図中の赤い丸で囲ったところ)。図 18 には、A) 青いテープ LED が光る直前、B) 青いテープ LED が光った時点、C) 点状 LED が光った時点、D) 点状 LED の発光が高輝度になった時点、及びそれぞれの時刻を示している。

B)とC)の時間差を求めると、0.011 秒=11 ミリ秒であった。従って、この場合の t_p は約 10 ミリ秒であり、図 11 で指摘した誤差の条件 $t_p \leq 10$ ミリ秒が満たされており、同時突き判定時間として 40 ± 10 ミリ秒の測定が達成されていると考えられる。

以上から、ラズベリーパイゼロを CPU として用いた同時突き判定システムにおいて、音取得のバッファサイズを 128 にして、ラズベリーパイゼロからの電圧を直接 LED に印加することで、同時突き判定が、 40 ± 10 ミリ秒程度の誤差で、可能であることが明らかになった。

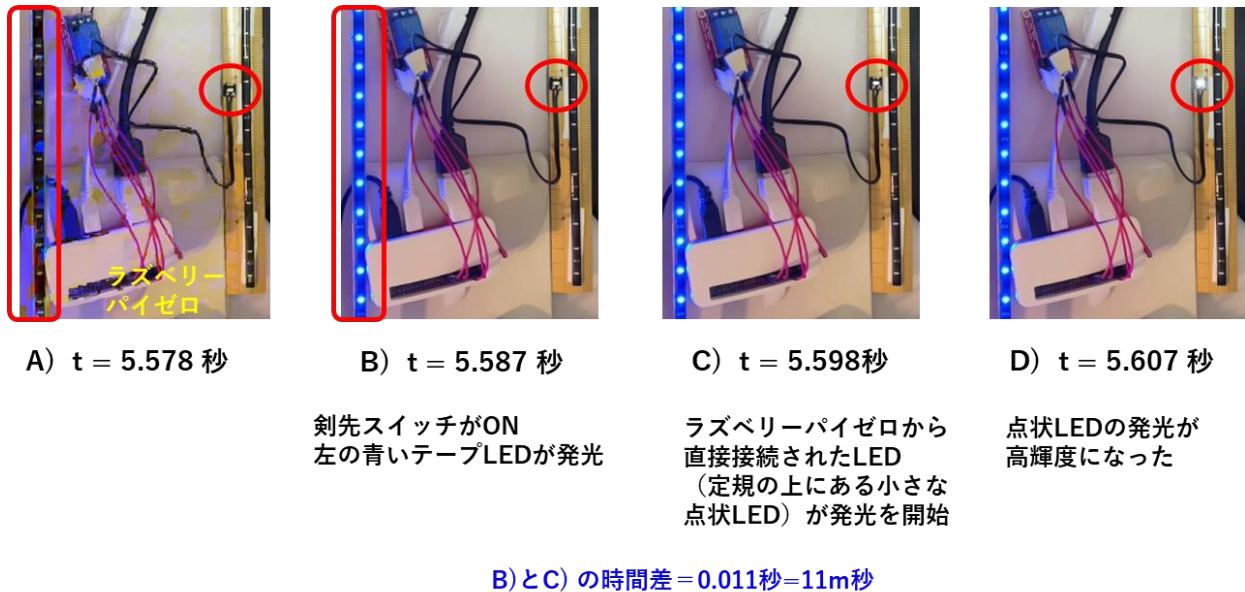


図 18 試作機による同時突き判定時間の評価
(ラズベリーパイゼロから直接 LED 接続して発光させた場合)

4. まとめと今後の課題

4. 1 まとめ

視覚的に認識しにくいフェンシング、特にエペ種目の「突き」の瞬間を認識しやすくするための可視化の方法、及びエペ種目において重要な「同時突き」の判定を、電気審判機を用いることなしに実施できる方法を提案することを目的として、「突き」の瞬間を LED によって可視化し、「同時突き」の判定を、電子ブザーの音によって伝えることで実現する試作装置を開発した。さらに、同時突きの要件である 40 ミリ秒以内の判定を実現するために、剣の間での信号のやり取りに関わるパラメータ、特に音読み取りのバッファサイズの最適化を実施した結果、以下のことが得られた。

- 1) 音読み取りのバッファサイズの最適化を、測定時間とスペクトル波形の 2 つの観点から行い、バッファサイズ=128 が最適であることを見出した。
- 2) ラズベリーパイゼロを CPU として用いた同時突き判定システムを構築し、音取得のバッファサイズを 128 にして、ラズベリーパイゼロからの電圧を直接 LED に印加することで、同時突き判定が、 40 ± 10 ミリ秒程度の誤差で可能であることが明らかになった。

4. 2 今後の課題

同時突き判定のための剣間の信号のやり取りとして、電磁波を用いる方法の検討が挙げられる。電磁波を用いることで、会場の雑音が大きい場合の誤作動を防ぐことが可能であると考えられる。また、電気審判機に縛られない特徴を持つ本システムを、フェンシング第 4 の競技として複数人対複数人の対戦などに展開していった場合などに、複数の選手に正確に情報を伝えることが正確にできるようになると考えられる。

さらに、試作装置を、充電池を除いて鍔の内側に収納することができ、剣さばきにも影響を与えないことが確認できたが、充電池については現状で鍔の内側に収納できず、小型化の検討が必要である。

5. 文献

- [1] 田中夕子、「アジア大会での喜びと悔しさを力に変えて明暗分かれた日本フェンシングが進む未来」 2018年8月26日
(<https://2020.yahoo.co.jp/column/detail/201808260003-spnavi>)
- [2] 特表 2019-522551
- [3] 「フェンシングの動きを AI で可視化、VR や 4K を駆使したハイテク観戦」 | 日経 xTECH (クロステック)
(<https://tech.nikkeibp.co.jp/atcl/nxt/column/18/00129/021600013/>)
- [4] 「FIE 競技規則(2020年11月更新版)_用具規定(m)」 p65.
<https://fencing-jpn.jp/cms/wp-content/uploads/2020/12/4ce8fb9b51aaf9da1254aa205cb3503f.pdf>
- [5] 特開平 7-51425
- [6] リレーの基礎知識 1-1 基礎編 | オムロン電子部品情報サイト - Japan
(<https://www.omron.co.jp/ecb/product-info/basic-knowledge-series/basic-knowledge-of-relays/part1/basics>)

6. 謝辞

本研究を遂行するにあたり、試作装置の設計やデータ分析の際に議論して頂いた武蔵野大学工学部数理工学科高石武史教授、武蔵野大学工学部数理工学科森竜樹助教に感謝いたします。

(原稿提出: 2021年1月20日; 修正稿提出: 2021年2月12日)