

Copyright © 2012 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Electronic Assisting Violin Tuner

Zhen J. Wang

Department of Mechanical Engineering
Curtin University
Perth, Australia

Cesar Ortega-Sanchez

Department of Electrical and Computer Engineering
Curtin University
Perth, Australia
c.ortega@curtin.edu.au

Abstract— This paper presents the design concept, implementation and verification process of the Electronic Assisting Violin Tuner (EAVT). The analysis of the violin sound to extract the frequency information was done using a Goertzel filter which was optimised and embedded to a microcontroller. The resulting frequency information is used to control actuators to tune the violin. Safety issues have also been considered and appropriate actions have been taken to reduce the risk of accidental string breakages which could result in injuring the violinist.

The goal of this project was to develop a device to assist in tuning every string of the violin. This project was undertaken with intentions that it could be further developed to increase the stability and robustness to the level that the prototype is fit for the market. It is therefore expected that there is still many areas of this project to improvement upon.

Keywords-Fundamental Frequency Calculation; Design and Integration; Violin Tuning.

I. INTRODUCTION

Children normally do not have enough strength to tune the violin so the violin teacher has to tune the violin before every lesson [1]. When the violin teacher is absent, the child's parents will normally be the ones required to tune the violin; this may be fine for parents who has experience with the instrument however it would be much more difficult for parents who have no experiences with violins.

To keep the child interested in playing and practicing the instrument, the violin will need to be frequently tuned. This can be very demanding on the parent as they usually are busy and it would be inconvenient for them. In addition, if parents are not violinists themselves, there is a high risk that the tuning could be done incorrectly; this can cause the child to develop bad habits which will be difficult to correct.

The motivation of developing an Electronically Assisting Violin Tuner (EAVT) is to aid people in such scenarios to allow beginners to have increased practicing time by reducing the amount of string breakages and time spent to tune the instrument.

The concept of an electronically assisting tuner for stringed instrument has already been developed and they are commercially available in the market. One of the most commercially recognised and successful self tuning instruments is manufactured by Gibson, under the trademark Robot Tuners™[2].

II. BACKGROUND

A Violin, also known as a fiddle, is the smallest stringed instrument in the violin family. There are four instruments in the violin family; the violin, viola, violoncello (cello) and double bass [3].

There are two major classifications of the violin: electric violin and acoustic violin; Fig. 1 shows both electric and acoustic violin side by side. The acoustic violin produces sound by the strings transferring vibration to the bridge (shown in Fig. 2), which then transfer the vibration to the body. Notice that the acoustic violin has a larger surface area compared to the electric violin which means that there will be larger amount of displacement of the air, resulting in amplified sound. The electric violin produces sound by picking up vibrations from the bridge using a microphone or pick-up, which is amplified and then sent to speakers or headphones.

Violins are usually four stringed instrument with the open-string notes of G, D, A and E from lowest pitch to highest pitch. An open-string note is defined as the full string or lowest note for that particular string. The fundamental frequencies for each string in a tuned violin are shown in Table 1.



Figure 1. Electric violin (left), and acoustic violin (right).

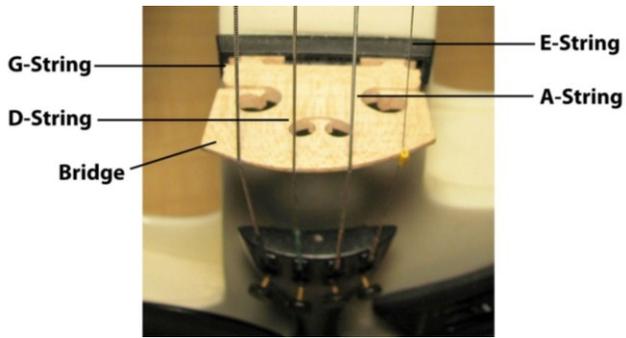


Figure 2. Violin parts- Middle section

Table 1. Standard Tuning Frequencies for Violin [4]

String	Fundamental frequency (Hz)
E	659.3Hz
A	440Hz
D	293.7Hz
G	196Hz

Violins are mostly made out of porous, unsealed wood, whose thickness and size is subject to fluctuation with changes in humidity and temperature. This is one of the major factors which leads to violins losing their ability to stay in tune for long periods of time [5].

III. PROJECT SCOPE

The goal of the project was to provide a solution which is able to tune a 4 stringed violin to a specified accuracy. By separating this goal into smaller components, the problems will be identifiable and treated as milestones as shown in Fig. 3.

Fig. 4 shows how the system is interconnected to give a good indication on the logic and concept behind the prototype.

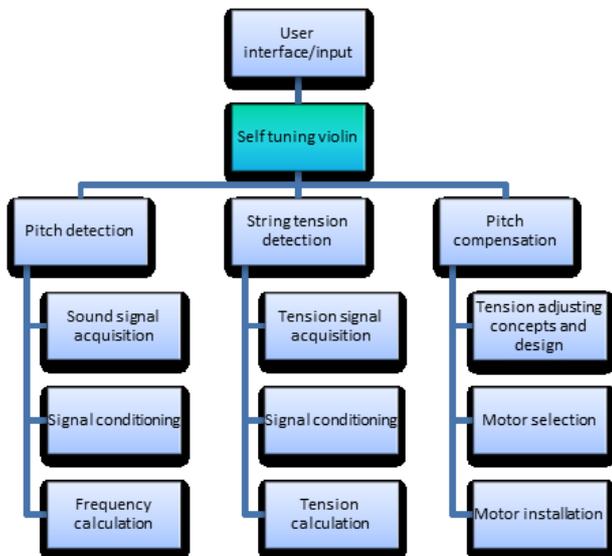


Figure 3. Breakdown of prototype concept.

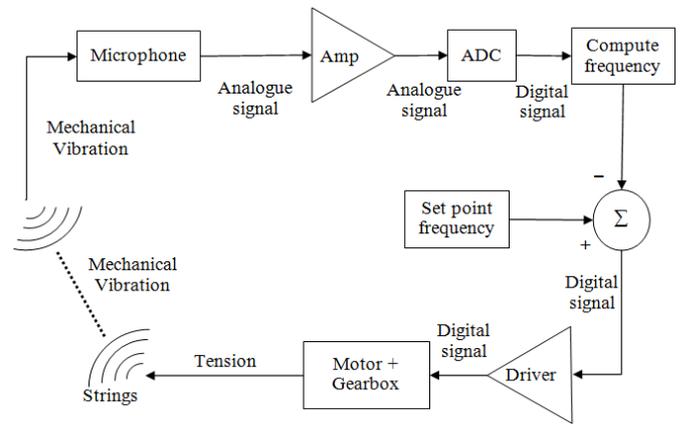


Figure 4. Block diagram of the system

A. Project Requirements

At the outset of the project it was determined that the prototype should meet the following requirements:

- Ensure that the violin strings will never reach the stage where it has been tightened to the point of snapping.
- The system must be powered by means of a portable power supply and
- Additional mechanical and electrical hardware must be relatively light compared to the violin.
- The mechanical and electrical hardware will need to be integrated to the violin such that the appearance of the instrument is not affected.
- Low cost (AUD 400)
- The acoustic properties of the violin must not be noticeably changed.
- The tuner will need to be reliable and robust
- Tuning should be completed reasonably quick

IV. FREQUENCY CALCULATION

A. Audio Acquisition – Part I (Microphone selection)

A microphone will be required to pick up the vibrations produced by the oscillating strings as the violin is played and convert these mechanical waves to an electrical signal. There are many types of microphones, most which have different behaviours and responses to sound waves. The piezoelectric (piezo) transducer was chosen as the microphone because it is very compact, light and has a very desirable frequency response for the project needs. Piezo microphones can also be configured to be not sensitive to ambient noise so it will not have issues with the audio feedback ‘squeal’ which is observed in other types of transducers such as the dynamic and capacitive microphones.

Fig. 5 shows the frequency response for the piezo. It is noticeable that when the piezo direction is 0 degrees to the sound source (i.e. when parallel to the sound source) the response of the microphone is very uniform when the frequency is within the operating range.

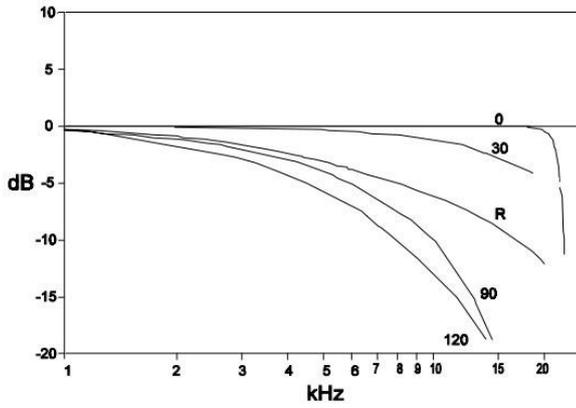


Figure 5. Frequency response for a particular PE microphone [6]

B. Audio Acquisition – Part II (Amplifier selection)

An amplifier is required as vibrations from the violin produce signals that are too low for direct feeding into most microcontrollers. The requirements for the amplifier were:

- Use commercial off-the-shelf (COTS) components.
- Have a variable gain.
- Receive unipolar input.
- Output can be biased by a variable amount, independent of the output gain (because most ADC in microcontrollers support unipolar inputs only).
- Accept a wide range of input voltages.
- Low distortion and noise output.
- Not sensitive to electromagnetic interference (EMI).
- High input impedance (to match the high impedance of the piezo element).

These requirements are very specific and therefore difficult to find an amplifier design which is suitable. An Op-Amp based piezoelectric amplifier designed by Richard [7] was modified to fulfil the requirements. The modification includes the capability to bias the output by a variable amount and adjusting the gain by a variable amount.

C. Microcontroller selection

The requirements for the microcontroller were:

- Low power consumption.
- Built-in ADC.
- ADC with sufficient sampling rate.
- Low voltage operation.
- Small package.
- Enough I/O for the purpose of the project.
- Powerful enough for the application.
- Enough memory for the application.

The Arduino Duemilanove (shown in Fig. 6) was chosen as it meets all of the requirements. It accepts a wide power supply range (5-12 V), has six 10-bit ADC and six PWM channels.

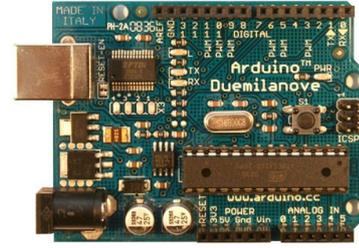


Figure 6. Arduino Duemilanove [8]

D. Pitch calculation – Part I (Accuracy requirements)

Human’s sound perception is not uniformly sensitive across the audible range; for example, human ears are more sensitive to small frequency changes in the low frequencies in comparison to the high frequencies. In music, the term ‘cent’ is used to describe the change in pitch where the difference between each semitone (in a twelve tone equal temperament [9]) is 100 cents. The use of ‘cent’ helps musicians compare the pitch of a note to another (relative to the human hearing) regardless of the frequency [10].

An experiment was conducted to determine the required accuracy. Adobe Audition™ was used to generate tones. Two different tones were produced, one on the left channel and a slightly different tone on the right channel. The test subject listened to the generated tones using high quality headphones to avoid production of beats. The subject needed to determine which channel had a higher pitch than the other. The frequency difference started off at 0.5Hz and was increased after every experiment and the two audio channels were swapped randomly to prevent the test subject guessing.

An additional test was done to see if the presence of harmonics would affect the perception of pitch change. To test this, the first five harmonics of different amplitudes were included in the tone. This experiment was conducted with subjects of different gender and age so that results would not be specific to a group of people. Table 2 shows the results.

Results showed that harmonics have a slight effect on the JND (Just Noticeable Difference) threshold; the E string (659.3Hz) is not changed which suggests that the additional harmonics will only make it harder to distinguish pitch at lower frequencies. The test with harmonics in the tone has a better representation of the violin acoustics because harmonics will always be present as they define the distinct violin sound. Hence, results with harmonics determined the JND thresholds that applied to this project, and an accuracy of ± 1 Hz was determined sufficient for this project.

Table 2. Experimental JND Threshold: Simultaneous Tones

Frequency (Hz)	Detectable change in frequency (Hz)	
	No harmonics	With five harmonics
659.3 (E)	3	3
440 (A)	1.5	1.6
293.7 (D)	0.8	1.3
196 (G)	0.5	1.5

E. Pitch calculation – Part II (Algorithm)

The Goertzel algorithm is a digital signal processing (DSP) technique which uses a different approach compared to the Fast Fourier Transform (FFT) for achieving the same goal; it utilises a Discrete Fourier Transform (DFT) which calculates a numerical value for a single frequency point of interest. This point of interest can be selected independently, whereas the frequency of interest for a FFT must be equally spaced. This is illustrated in Fig. 7. Because of this flexibility in the Goertzel algorithm, detecting a small number of tones over a large frequency range can be more efficient than computing the FFT for this particular application. The Goertzel algorithm is used in applications such as Dual Tone Multi-Frequency (DTMF) detection and frequency response measurements. It is increasingly popular due to the CPU friendly nature of the algorithm [11].

The frequency resolution (or bin width) of Goertzel filters is similar to the FFT's relationship. The value of N also controls the bin width. If the sampling rate is 8 kHz and N is 100 samples, the bin width is 80Hz [11]. The dissimilarity of the Goertzel from the FFT is that the value of N does not have to be a number of 2^n which is a big advantage for balancing between speed and performance of the algorithm.

The selected sampling rate was four times the fundamental frequency; therefore the highest sampling rate was 2640 samples per second (sa/s).

The Goertzel filter was first implemented in MATLAB to test the accuracy and reliability of the algorithm. Since the state of tuning is defined by the fundamental frequency of the violin strings, the phase information is not required. This means that it is possible to use a simplified Goertzel algorithm. Kevin Banks [11] explains the Goertzel filter in detail and also shows the simplified Goertzel algorithm. The simplification helps speed and processing time by reducing redundant and complicated calculations.

The computational requirements of a floating point implementation of the Goertzel filter far exceed the processing capability of the Arduino Duemilanove. For example, Arduino can perform a multiplication of two 8 bit integers in around 1.56 μ s; while a multiplication of two 8 bit floats takes 10.2 μ s. By using fixed point algorithm, the Goertzel filter is efficient enough to be running on the Arduino. To further speed up the calculation, all of the required trigonometric operations were solved using a look-up table.

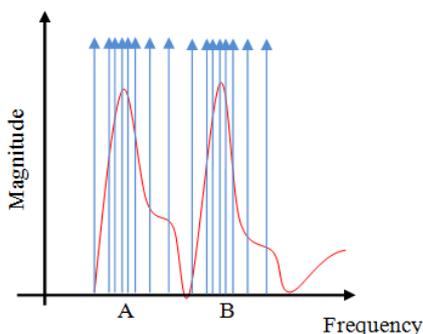


Figure 7. Goertzel filter scan

The output of the Goertzel algorithm is shown in Fig. 8. The Arduino needs to determine the global maximum to locate the fundamental frequency of the input signal. Fig. 9 shows the logic behind locating the global maximum. To make the scan more efficient, only frequencies near the frequency of interest (F_i) are scanned with maximum resolution, further frequencies are scanned with a resolution of 5Hz.

V. MECHANICAL DESIGNS

A. Motor Selection

The selected servo motor is a unit with a DC motor, position sensor and internal speed reduction gearboxes. Servomotors are manufactured such that they rotate only to a limited range. A modification was necessary to allow unlimited rotation in either directions of the output shaft. The modification is very simple and more details can be found at Embeddedtronics [12]. The selected servo motor was the VG 8.0g with dimensions of 25.2/8.6/29.3mm (L/W/H) and it is able to produce a maximum torque of 1.5kg.cm with a 6 V power supply.

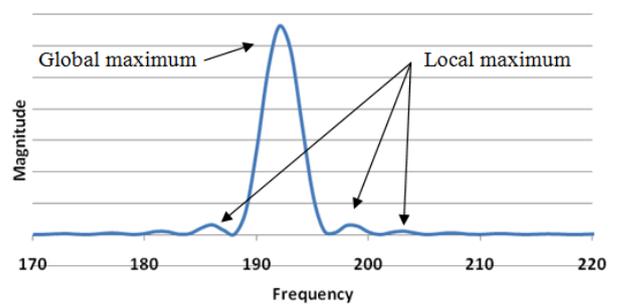


Figure 8. Goertzel filter output

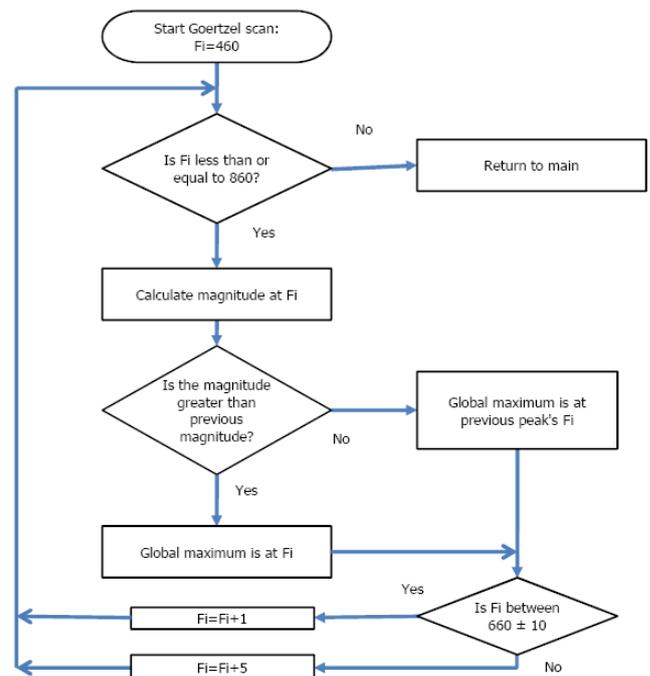


Figure 9. Global maximum determination flow chart

B. Transmission Design

A fine screw thread and nut combination was selected as the actuating mechanism. There are many advantages for using a fine screw thread and nut as the linear actuator configuration; the main reasons why are:

- High speed reduction ratio.
- The screw and nut will not slip a significant amount when the motor is switched off.
- Small and simple package.
- Reliable.
- Easy to manufacture.

One main disadvantage of the thread and nut drive is that the nut has a limited travel before reaching the end of the screw. This problem can be solved by selecting longer screws and installing limit switches at the end of the screw so that microcontroller knows that the limit has been reached. The configuration of the transmission is shown in Fig. 10.

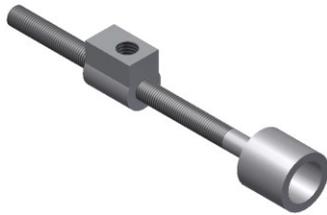


Figure 10. Motor transmission mechanism

C. Tuner housing design

The design requirements for the tuner housing were:

- The acoustic properties of the violin must not change a noticeable amount.
- Tuner will need to be reliable and robust.
- The tuner modification must not cause any irreversible changes to the violin body.
- The tuner can be removed and the violin can be reverted back to prior installing the tuner.

Any additional mass to the violin body would dampen the sounds produced. This would alter the acoustic properties of the violin and thereby impeding on the first design requirement. The tailpiece of the violin is suspended on top of the violin with the tailgut and the four violin strings holding it in place. There is no direct contact between the body of the violin and the tailpiece, therefore an added mass of the tailpiece does not contribute to acoustics of the violin. Hence it was decided that the tuner housing should be a replacement for the tailpiece.

The final design of the tailpiece and final assembly are shown in Fig. 11. The main objective of the design of this new tailpiece was to keep the shape and size very similar to the existing tailpiece and allow space for the existing tailgut, strain gauges and the four motors.



Figure 11. Assembled tailpiece tuner

D. Safety

As the tension increases in a string, more potential energy is stored. If this string breaks, all this potential energy will be released very quickly and could cause a violent whipping action. This is especially dangerous because the violin is held close to the face while playing; any injuries to the violinist's eyes could be very serious. This risk was managed by incorporating strain gauges and instrumentation amplifiers to allow the Arduino to monitor the strain levels of the string. If the string tension reaches over a threshold, the motors are stopped and reversed by a small amount to relieve the tension.

With the strain gauge monitoring the string tensions, it is also possible for the violinist to practice tuning the instrument manually, without worrying about the string snapping if they make mistakes; the motors will automatically reverse to relieve tension off the strings.

VI. TUNING THE INSTRUMENT

The controller for the Violin Tuner was implemented using a Fuzzy Logic Controller (FLC). A FLC takes in vague and ambiguous information and resolves a solution in a similar manner to how a human would make the decision. More information about Fuzzy Logic controllers can be found in Seattlerobotics [13] and the Imperial College of London [14].

The basic structure of a FLC differs from conventional controllers because rather than modelling the system, the FLC uses a rule-based "if X and Y then Z" approach which is easily implemented in a microcontroller. The implemented FLC for the A string (440 Hz) is shown in Fig. 12.

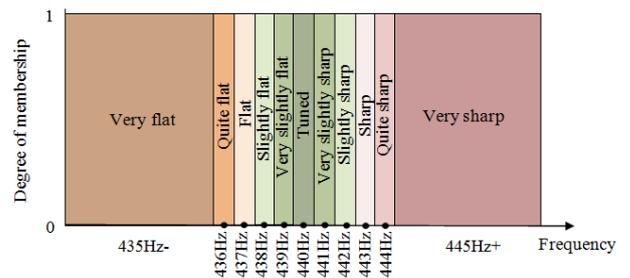


Figure 12. FLC for the A string (440 Hz)

From the calculated fundamental frequency, the microcontroller will know which direction and speed to turn the motor in order to adjust the pitch to the set point.

VII. RESULTS, CONCLUSIONS AND FUTURE WORK

The violin tuner presented in this paper is able to tune all four strings to the desired accuracy of $\pm 1\text{Hz}$. The time required to tune each string is dependent on the difference between set point and actual frequency. On average, it took approximately 5 seconds to tune a single string when it is approximately 50+cents out of pitch. During continuous testing, the tuner achieved the required frequency 90% of the time. The tuning reliability has been improved by adding a 4th order Inverse Chebyshev low pass filter before amplification.

While the ultimate goal has been achieved, there are many other aspects of the results that could be improved on, such as speed (make the tuning faster), robustness (make the tuner durable), size (make it smaller), weight (make it lighter) and reliability (make it more accurate every time).

There are also possibilities of additional future work to further improve the performance of the EVAT. One possibility for future work is to have the capability to simultaneously tune all four strings at the same time (polyphonic tuning) to dramatically decrease the time required for tuning the instrument.

Another future concept is to implement the YIN algorithm to compare the performance against the Goertzel. The advantage over Goertzel is that it can handle harmonics very well [15] and therefore the implementation of the low pass filter would not be required which would save cost in components and circuitry size.

REFERENCES

- [1] P. Zaret, "How to choose a violin for a child," [Online]. Available: <http://www.zaretandsonsviolins.com/choosingforchild.html>.
- [2] Gibson, "The Robot SG Guitar with Automatic Electric Guitar Tuning," [Online]. Available: <http://www.gibson.com/robotguitar>.
- [3] "Violin family (musical instrument)," 2011. [Online]. Available: <http://www.britannica.com/EBchecked/topic/569200/stringed-instrument/53726/The-violin-family>.
- [4] "Note Frequencies," [Online]. Available: <http://www.seventhstring.com/resources/notefrequencies.html>.
- [5] B. Buckingham, "Why is my violin always out of tune?," [Online]. Available: http://www.jcviolins.com/instrument_changes.html.
- [6] J. Makhair, "SOUND MEASURING INSTRUMENTS," [Online]. Available: http://www.who.int/occupational_health/publications/noise6.pdf.
- [7] M. Richard, "Piezo contact mic amplifier," June 2006. [Online]. Available: <http://www.megalithia.com/sounds/tech/piezo/opamp.html>.
- [8] "Arduino Duemilanove," [Online]. Available: <http://www.arduino.cc/en/Main/ArduinoBoardDuemilanove>.
- [9] Y. Nov, "Explaining the Equal Temperament," [Online]. Available: <http://yuvalnov.org/temperament/>.
- [10] R. Nave, "Cents," [Online]. Available: <http://hyperphysics.phy-astr.gsu.edu/hbase/music/cents.html>.
- [11] K. Banks, "The Goertzel Algorithm," August 2002. [Online]. Available: <http://www.eetimes.com/design/embedded/4024443/The-Goertzel-Algorithm>.
- [12] Embeddedtronics, "Servo modification for 360 degree rotation," [Online]. Available: <http://www.embeddedtronics.com/servo.html>.
- [13] S. D. Kaehler, "FUZZY LOGIC - AN INTRODUCTION," [Online]. Available: http://www.seattlerobotics.org/encoder/mar98/fuz/fl_part1.html#INTRODUCTION.
- [14] Imperial College London, "Fuzzy Logic and it's uses," [Online]. Available: http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol2/jp6/article2.html.
- [15] H. K. A. Cheveigne, "YIN, a fundamental frequency estimator for speech and music," June 2001. [Online]. Available: http://audition.ens.fr/adc/pdf/2002_JASA_YIN.pdf.