Autogaze - Guitar Pedal Automator

HITESH SONAWALA, University of California, San Diego
RONNIE LUSK, University of California, San Diego
SIVASANKAR PALANIAPPAN, University of California, San Diego
SNEH SHAH, University of California, San Diego

Guitarists use pedalboards to manipulate the sound of their guitars and tuning them is a difficult task. Autogaze, a pedalboard automator, alleviates this problem by automatically tuning the pedal knobs as per the guitarist's liking. Autogaze enables the user to select presets with the mobile application, which in turn controls a mechanical construct that tunes the knobs to the desired setting. The mobile application connects to a Raspberry Pi through Bluetooth while the mechanical construct consists of a detachable stand along with caps to be placed on the knobs.The knobs have servo motors connected on them which help them to be rotated. Model was tested by a guitarist who saved 2 to 3 mins using the app.

CCS Concepts: • **Computer systems organization → Embedded systems**.

Additional Key Words and Phrases: raspberry pi, android application

ACM Reference Format:
Hitesh Sonawala, Ronnie Lusk, Sivasankar Palaniappan, and Sneh Shah. 2018. Autogaze - Guitar Pedal Automator. *Proc. ACM Meas. Anal. Comput. Syst.* 37, 4, Article 111 (August 2018), 10 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Guitarists are an integral part of any band. They not only provide variety to the band with different tones and tunes but also pep up the crowd by actively interacting with them and take the entire performance to another level. Pedals are an important device which they connect to the guitar and play around with , by tuning large various knobs to different levels . Tuning them is a tedious task , as there will be large number of pedals and each of them will have many knobs. Tuning them between songs is an irritating task, and during the songs is even more challenging . Guitarists would find this nerve wracking as the process should be done quickly and correctly. Taking a long time would make the crowd agitated and not tuning correctly would lead to the song being spoiled and will lead to agitating the crowd anyways.

Fig.1 shows how a typical setup would be and tuning all of these pedals would be very hard. To solve this problem, there is a need for automation of the pedal knobs to make this process easier. Very few solutions are available which do the automation of the knobs. Some of these are the RigMaster,Autobot and the terrorBot.Automation should make the user do as less as possible.Autobot and terrorBot make the user rotate buttons or MIDI inputs which is easier than manually tuning the knobs but does not completely automate the process. RigMaster does

Authors' addresses: Hitesh Sonawala, hsonawala@ucsd.edu, University of California, San Diego; Ronnie Lusk, rlusk@ucsd.edu, University of California, San Diego; Sivasankar Palaniappan, spalania@ucsd.edu, University of California, San Diego; Sneh Shah, s5shah@ucsd.edu, University of California, San Diego.

2476-1249/2018/8-ART111 $15.00
https://doi.org/10.1145/1122445.1122456

Fig. 1. General setup of pedals in a concert

the same but provides an application, to rotate and set the level.There is no solution which will tune the knobs with just a click. Making users to click a button is the easiest from an user's perspective as it requires the least effort. Autogaze asks the user to provide minimum effort by providing an application with buttons with various presets. The user needs to click on the particular preset and the knobs are rotated automatically. This is done with the help of servo motors connected to the raspberry pi through bluetooth.

The article describes the following:

- Technical Material:
  - Block Diagram
  - Components
- Milestones
- Conclusion

Technical material talks about the design of the entire system , along with the integration of the various components to make the pedal. The components here are the mechanical design, the raspberry pi setup to control the servo motor and the android application to control the entire setup.

Milestones describes the plan created to design and execute the entire system , with changes to the plans and the failures which came about while executing all of the plans.

Conclusions is the description of the result, importance of this device and future work.

2 TECHNICAL DESCRIPTION

Design of Autogaze involved the following components:

1. Raspberry pi setup
2. Mechanical construct

3. Android Application

2.1 Design

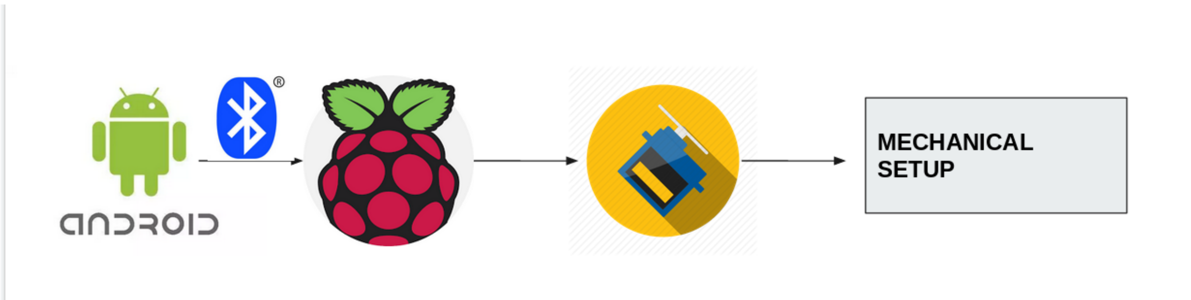

Fig. 2. Overall block design of Autogaze setup

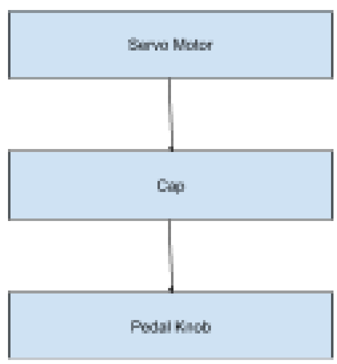

Fig. 3. Block design of mechanical setup

The 1st figure is the overall high level block design. The application communicates to the raspberry pi using bluetooth. The preset is communicated to the raspberry pi using bluetooth. Once the raspberry pi gets the preset, it rotates the servo motors according to the how the presets have been programmed. The servo motors are in turn connected to the mechanical setup.

The mechanical setup consists of a cap and a support block. The servo motors are attached on the mechanical support so that they do not fall off when the motors rotate at high speed.

2.2 Components

2.2.1 Raspberry Pi setup . The raspberry pi has GPIO ports to which the servo motors are connected. The raspberry pi 3B has 40 GPIO pins out of which VCC (3.3V), ground (0V) and four GPIO pins are used to connect the four servo motors. In this case pin number 11,12,13 and 15 were used. These pins can be chosen at random from any of the GPIO pins, only rule is that it should not be VCC or ground. Connecting the servo motors to the raspberry pi completes all connections to the pi.

Next is the coding of the raspberry pi. This involves four code snippets.

1. The 1st snippet is the controlling of the servo motor. The servo motor is controlled with the help of pulse width modulation, where pulses are to provided to the servo motor with different duty cycles to rotate the servos to different angles. The angle of rotation was decided based on the levels present on the knobs of the pedal. Initially the angle was given as 0°. Then the duty cycle was changed to different values, there by giving random angles, stopping when a particular intensity level of the knob was reached. These angles were stored in a list in the python program. To provide the pulses , custom GPIO libraries were used. The servoblaster kernel object was used, which runs the servos with the help of bash commands. To implement these in python the os library was used and bash commands were called on python itself.

2. The 2nd aspect is the programming of the Bluetooth channel to get value obtained from the application . This is done with the help of the bluetooth library in python. The bluetooth library helps in setting up of a bluetooth server socket. This allows us to listen to a particular port of the socket. Here bluetooth has been configured as socket 1. Once this has been done, the client address and name is obtained. Using the client's name ,the value(preset) can be obtained using the recv function. Once the value(preset) was obtained, the value was passed to the function designed in snippet 1 which will call the corresponding angles for the servo motors stored for that particular preset.

3. The 3rd snippet is the automatic pairing of the raspberry pi with the mobile phone using bluetooth. Since, it would not be possible for user to connect the rpi to ethernet or a montior screen and pair the bluetooth of the rpi with the phone 's bluetooth. This was done by creating a bash script using the bluetoothctl module.

4. The 4th snippet is combining the above three snippets. The 1st two snippets are combined into a single file as the value obtained from bluetooth is passed immediately to the function to set the duty cycle values for the servo motors to make them run. This program and the 3rd snippet must be running always when the pi is powered on and in a parallel manner. To do this the multicore architecture of the raspberry pi is used which helps to run both the scripts simultaneously.

2.2.2 Mechanical Construct. The mechanical construct has two parts: 1. The 1st part is the caps which act as the bridge between the knobs and the servo motors. The cap had two openings in either end. The opening on top is a circular one which is of a smaller diameter to fit the servo motor. The opening on the other end is of a decagonal opening in order to increase the friction between the knobs and the caps to make it more sturdy.

2. The 2nd part is the mechanical block or support on to which the servo motors are rested on. The servo motors are stuck on to the block , which will prevent them from popping out when the motors rotate. The block has been designed such that the inputs and the outputs from the pedals are not disturbed.

The pedals and the block have been designed using Google Sketchup . To design the block and the caps , measurement of the pedals were taken using vernier calipers. Once the measurements were taken, the designs were made in sketchup and then converted to .stl format in order to 3D print them.

Around 10 versions of the caps were made in order to get the perfect dimensions . Two versions of the block were made.

Table 1. Dimension of block

Segment	Dimension
Height	85.65 mm
Width to hold motor	22 mm
Thickness	6.6mm
Length	71.15 mm

Table 2. Dimension of cap

Segment	Dimension
Top half-Outer diameter of cylinder	10.85 mm
Top half-Inner diameter of cylinder	4.76 mm
Top half-Height of both cylinders	28.5 mm
Bottom half-Diameter of cylinder	13.9 mm
Bottom half-Length of segment of outer decagon	3.708 mm
Bottom half-Length of segment of inner decagon	3.399 mm
Bottom half-Diameter of outer decagon	12 mm
Bottom half - Diameter of inner decagon	11 mm
Bottom half- Height of cylinder	16mm

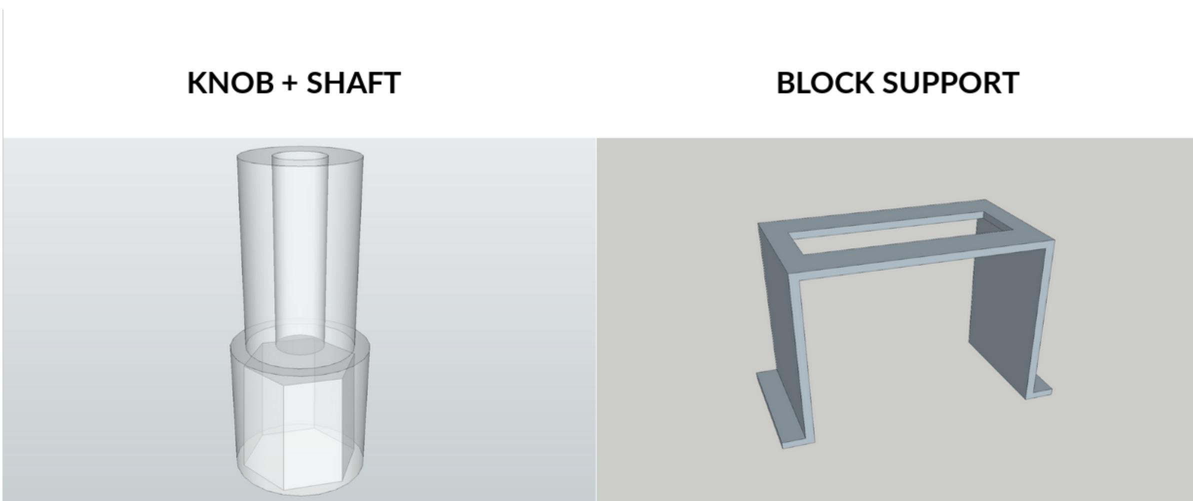

Fig. 4. 3D design of cap and block

2.2.3 Android Application. The android application is a simple one with 5 presets and an ability to pair device with raspberry pi through bluetooth. The application creates a socket on the client side and sends data to rpi through this socket.This application was created using two different software:

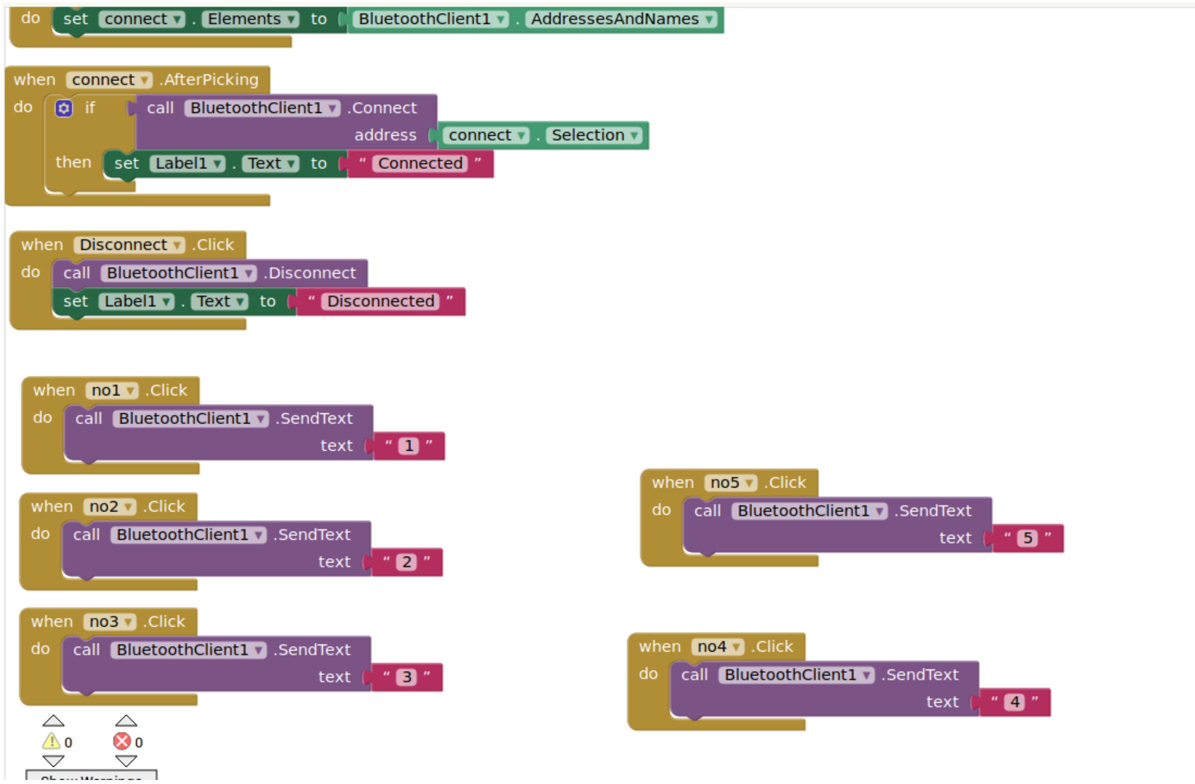

Fig. 5. Block design of application

1. MIT App Inventor : This helps to create android application easily by just dragging of the blocks.This version of the application has 5 preset buttons and 1 button to connect to the particular rpi and another to disconnect. Clicking connect would open a list of paired devices from which the user can choose the raspberry pi. Fig 5. shows the block design used for making this application.When connected it connects Elements to Bluetooth 's address and names.When one particular address is connected it displays the message connected.Similarly for the button disconnected. Once connected the text or value is sent according to the preset button which was selected.Fig 6. shows the user interface of the application.

2. Android Studio: This version does the same task as the previous i.e. sends data to the raspberry pi through bluetooth. However here there is no menu to select the raspberry pi from the paired devices. The UUID of the bluetooth has been hardcoded .The code is available on this page- https://github.com/codspalaniappan95/Autogaze

3 MILESTONES

3.1 Original Milestones

- Design pedal block:
  - Generate a rough blueprint of the pedal block by taking measurements of the pedal board setup
  - Create a 3D model for this in Solidworks

Fig. 6. GUI of app created using MIT App Inventor

- 3D printing
  - Print the 3D model generated in Solidworks.
  - This process will have to be repeated (correcting errors each time) until we get the perfect fit for the pedal board
- Design of servo motor with rod
  - Design a mechanism to connect the servo motor shaft to the knobs on the pedal
- Raspberry Pi to control the servo motors
  - Program and write efficient code in order to control the servo motor
  - Initially control the servo using command line interface
  - Control using an application
- Fixing knobs to motor and final product
  - Consolidate all components into the final hardware setup
- Software Application
  - Develop computer or mobile application using Android Studio
- Testing and corrections
  - Test and evaluate the final product in a real world setting
  - Make improvements according to feedback

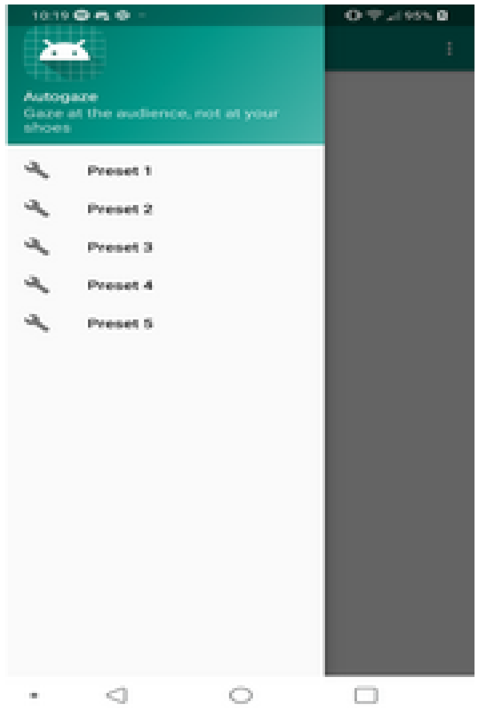

Fig. 7. GUI of app created using Android Studio

3.2 Final Milestones

- Raspberry Pi to control the servo motors (Sneh, Hitesh, Siva)
  - Program and write efficient code in order to control the servo motor
  - Initially we will try to control the servo using command line interface
  - Move on to application
- Software application (Ronnie)
  - Create application on MIT app inventor
  - Convert it to Android Studio
- Interface the Android Application with the RPi (Siva, Sneh)
  - Interface the android app with the rpi such that the app can send preset values to the rpi via bluetooth.
  - The rpi should be able to pair with any bluetooth device and should also run the servo motor program on boot.
- Mechanical Design (Hitesh)
  - Design and 3D print the knob fixtures that go on top of the guitar medal knobs and a mechanical support block which can stabilize the servo motors.
- Interface the Mechanical Design with the RPi setup (Siva, Hitesh, Sneh)

Fig. 8. Overall completed product

– Interface the support block and knob fixtures with the servos such that the servos can rotate the knobs with extreme precision.
– The mechanical block should be sturdy and stable enough so that it does not move when the motors rotate while the knob fixtures should have enough grip so that they can rotate the knobs without slipping.

Changes made to the milestones were the reordering and switching from sketchup from solid works. The reason for the reordering was that none of us had any experience in mechanical design and hence we decided to complete many tasks in knwown areas and did the remaining. The reason for switching from solidworks to sketchup is also the same , solidworks was difficult to work with due to lack of knowledge in mechanical design.All the milestones were completed successfully.

3.3 Issues Faced

- Running the RPi application and the Bluetooth script on start-up
  - Solved through parallel execution of both
- Making sure the RPi does not stop working when bluetooth is disconnected
  - Solved using a try - except block in the RPi application
- Removing jitter from the 180°servo motors
  - Solved using different GPIO library-using servoblaster instead of GPIO

4 CONCLUSION

Autogaze is a simple solution to the problems most guitarists face while tuning their pedals mid-performance. Autogaze consists of a simple mechanical setup that goes on top of the guitar pedal and a mobile phone application which controls the setup to tune the guitar pedal. It takes almost 2 to 3 minutes to properly tune a guitar pedal for a song, and it is impossible to tune it while playing a song. For this reason, guitarists use more than one pedal, and each pedal costs approximately 200 dollars. Autogaze allows the user to have predefined presets for each of the pedal knobs, and the mobile application will tune the pedal as per the selection of the user. This project mainly comprised of three parts - the raspberry pi and the servo motor setup, the mobile application, and the mechanical design. The mechanical block design had to be sturdy and stable enough so that it can handle the quickly rotating motors, and the knob fixtures had to be made such that they can grip the pedal knobs so that they can rotate with the servo motors and at the same time, it does not slip and cause errors in the rotation. The problem with the current design is that it works only on a guitar pedal with 4 knobs aligned. In the future, a more modular design for the mechanical block can be made so that it can fit on any kind of pedals. Moreover, the current mobile application has the presets defined in the raspberry pi. That can be changed such that the user can input and save the presets so that he can change it on the fly, without much trouble.

[1–3].

REFERENCES

[1] GigRig. 2016. AutoPot. https://www.musicradar.com/news/guitars/namm-2016-add-automated-hands-free-control-to-any-effects-pedal-with-thegigrigs-autopot-633238
[2] Micheal Karsay. [n.d.]. TerrorBot. http://trigonometrie.bplaced.net/blog/terrorbot/
[3] Novel-Tones. 2015. RigMaster. Retrieved November 5, 2015 from https://www.kickstarter.com/projects/rigmaster/rig-master-guitar-and-bass-analog-gear-control