How to build pipe organ robots

Michael Krzyzaniak
mkrzyzaniak@protonmail.com
RITMO Centre for Interdisciplinary Studies in Rhythm, Time and Motion, University of Oslo
Oslo, Norway

Figure 1: Three miniature pipe organs consisting of three colorful Dr. Squiggles octopus-shaped robots, the three wind chests that they are on, 24 organ pipes, and one air supply in the rear.

ABSTRACT

In this paper I describe how I built three miniature computer-controlled pipe organs, using accessible digital fabrication techniques. I was motivated to build them as part of my ongoing research on musical robot swarms, which necessitated simplified, easy-to-build organs. Having done that, the goal of this paper is to present the average computer musician with the equations, software, information, and key materials that I used, so that they can build their own pipe organs with a low barrier to entry, minimal assembly time, and using standard digital fabrication equipment. Finally, I describe a few simple algorithms for imbuing the pipe organs with a small amount of self-awareness which will facilitate their use in common computer-music scenarios.

The completed pipe organs can be seen in the video below, and in several other videos referenced throughout this paper.

- https://www.youtube.com/watch?v=3dEJR9xg5nU

CCS CONCEPTS

• Applied computing → Sound and music computing; **Media arts**; **• Hardware**;

KEYWORDS

Pipe Organ, DIY, Computer Music, Dr. Squiggles, 3d-Printer, Laser Cutter, Digital Fabrication, Musical Robots, Musical Robot Swarms, Papers Written in the First Person

ACM Reference Format:
Michael Krzyzaniak. 2022. How to build pipe organ robots. In *Proceedings of 17th International Audio Mostly Conference (AM '22).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 MOTIVATION

In the field of robotics, *swarms* are the idea that if you want to accomplish something complex, sometimes it makes sense to use a large number of simple robots. Even if no individual robot is capable of doing anything complex on its own, together they might be able to accomplish more than the sum of the parts [19]. Swarms naturally have many nice properties that are also desirable in the context of computer-generated music, such as self-organization,

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
AM '22, September 06–09, 2022, St. Pölten, Austria
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

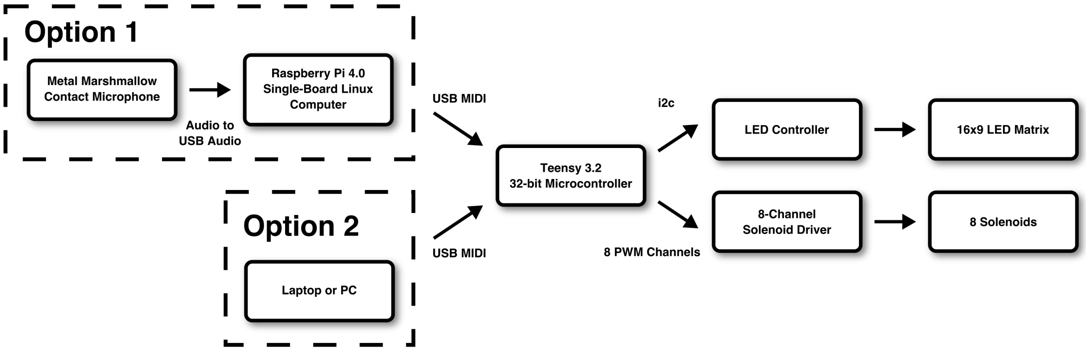


Figure 2: Block Diagram of the internal circuitry of Dr. Squiggles. It consists of a Teensy microcontroller that controls a) 8 solenoids and b) an LED panel that displays an eye. The Teensy receives USB MIDI commands from either the internal Raspberry Pi computer or, optionally, an external computer. An external microphone is intended to be connected to the Raspberry Pi.

emergence, aesthetic appeal, and interesting blends of structure and randomness. It is not then surprising that swarm, flocks, and other multi-agent systems have been the basis for many music synthesis algorithms [1, 4]. However, this work has focused largely on a) computer-simulated swarms e.g. the well-known *boids* algorithm [16], and b) spatial metaphors in which the behaviour of the swarm is determined by the position of the agents in Euclidian space, with musical features being extrinsically superimposed onto that space. By contrast, very little work has been done on musical robot swarms in which a) the agents are real physical autonomous robots, addressing particular strengths and challenges of this as opposed to simulation, and b) the swarm operates directly on the music, intrinsically, with no intervening spatial metaphor.

In order to address this, I previously built a swarm of 10 octopus-shaped rhythmic tapping robots called Dr. Squiggles, and studied their musical behaviour while playing unpitched rhythmic patterns [10]. Subsequently, I wanted to extend that work to include pitched music. Consequently, I decided to build miniature pipe organs for Dr. Squiggles to play, with each organ containing only eight organ pipes, and therefore eight pitches. This constraint means that no individual robot is able to play very interesting music on its own, but if there are many robots, each playing an organ with a different set of pitches, and if they work together as a swarm, then somehow they should be able to make interesting music.

In this paper I will first describe the design and construction of the pipe organs, including some open-source tools I made to facilitate the process for others. I will then describe software techniques for imbuing the robots with some degree of self-awareness, which will be a prerequisite for the future study of swarm behaviour in the robots. Note that whether a given collection of robots constitutes a swarm depends on their software; this paper describes the development of the robots considered as a hardware platform for the study of swarms, while their use as a swarm is left for future work.

2 PREVIOUS WORK

Pipe organs have been around for thousands of years [2], and have a prehistory that is intertwined with that of robotics and computer music [8, 9]. The earliest fully automated self-playing pipe-organ design I am aware of is in Kircher's 1650 treatise Musurgia [6][1], although it is not clear that this was ever built, and each component had been around since at least the 9th century [14].[2] Throughout the 19th century, orchestrions flourished; many of them still exist and have been retrofitted with circuitry that allows them to be controlled via MIDI. The 'Man and Machine Robot Orchestra' at the Logos Foundation, founded in 1968, contains a large number of organs and organ-like instruments [13]. Eric Singer, as part of Pat Metheny's Orchestrion project, built an organ-like robot consisting of transversally blown glass bottles [18]. In 1984, researchers at Tuksuba university built a humanoid robot that plays organ [17].

Despite this, self-playing pipe organs have not figured as prominently in computer music as, say, snare drum or guitar. This is in part because organs are not easily attainable, and are exceedingly difficult and time-consuming to build. Modern digital fabrication has the potential to drastically streamline the organ building process, and in fact there is a company that sells a laser-cut DIY hand-crank organ kits.[3] However, commercial solutions like this do not afford the musician any flexibility regarding the design of the organ and pipes. Furthermore, a review of the historical literature on organ building reveals precious few equations and other critical details of organ pipe design; presumably this information was guarded by builders as a trade secret. This means that even today, a computer musician and would-be organ builder is faced with a steep learning curve fraught with trial and error in order to even get started. The primary contribution of this paper then is to extend the long history of automated pipe-organs into the realm of modern DIY; to present the methods, tools, and materials that an average member of the computer-music community could use in order to build a simplified computer-controlled pipe organ, with minimal assembly time, easily attainable parts, and using standard digital

---

[1] Liber IX, Pars V, pp. 308 ff

[2] A tangent perhaps of interest to the computer music community: The earliest telematic instrument of which I am aware also involved pipe organ. In 1849, Innocenzo Manzetti had run pneumatic tubes from the harmonium in his workshop on Via Giocondo in Aosta, out the window, and over to the cathedral across the street, so that he could play the pipe organ there without leaving his laboratory [7, 15].

[3] https://castlewoodorgans.com

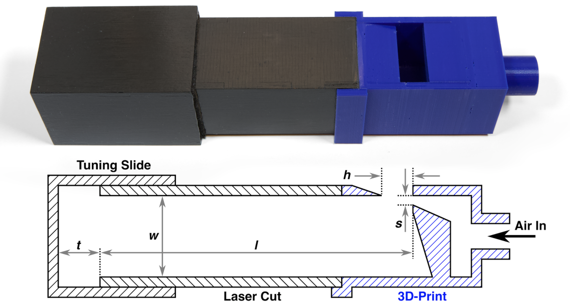


Figure 3: Top: One completed organ pipe, MIDI number 64 (E4, 330 Hz). Bottom: Schematic of the same showing the length l, width w, mouth height h, tuning headroom t, and wind-sheet thickness s.

fabrication equipment that can be found in most makerspaces and universities. Because these organs will be computer-controlled, the secondary contribution is to show how they can be imbued with a small amount of intelligence, as a pre-requisite for their use in a musical robot swarms or other interactive computer musical scenarios.

3 IMPLEMENTATION

I built three pipe organs, to be played by three Dr. Squiggles robots, and one shared air supply, which are all shown in Figure 1.[4] For the sake of this article, Dr. Squiggles is essentially a housing for a Raspberry Pi Linux computer that is connected to 8 solenoids for tapping rhythms, and to a microphone for listening. A more detailed block diagram of Dr. Squiggles' internal circuitry is in Figure 2. Further details of the construction are documented in a Make article [11] and will not be further elaborated here. The construction of the pipe organs and air supply follows.

3.1 Organ Pipes

I first built 24 organ pipes, eight of them for each of three robots. A photo of one completed pipe is shown in Figure 3, along with a schematic of the same.

To simplify construction, I chose to build square cross-section pipes based loosely on Figures CCLIX and CCLXX from the 1905 treatise The Art of Organ Building by George Ashdown Audsley [3]. The pitch and absolute dimensions of those diagrams are not given in the treatise and are impossible to infer; not all parts of an organ pipe scale at the same rate, and finding the correct size for certain parts is a mysterious art. So I supplemented the diagrams with a mix of empirical observations made while prototyping, and equations I found on the internet.[5]

The inner width (and depth) w of the pipe in millimeters is given by

$$w = 155.5/2^{\frac{m+\nu-36}{16}} \tag{1}$$

where m is the desired MIDI note number of the pipe, with $m = 69$ corresponding to A 440. ν is the 'normalmensur deviation' in semitones. Positive values of ν make the pipe narrower than it 'should be', negative wider. Negative numbers ostensibly have a more string-like sound, and positive have a more flute-like sound, and values near 0 are for principals. $\nu = +4$ means this pipe will be as wide as a pipe 4 semitones higher 'should' be. Pipes that are too wide will not speak at all, and pipes that are too narrow will sound a harmonic and not the fundamental. The value 16 in the equation means that this rank will have a so-called 17th halving ratio, where the width doubles every 17 semitones.

The length l from the flue (where air exits the pipe) to the top of the pipe, depends on the desired frequency f in Hz, which, for completeness, can be calculated from the MIDI note number μ as

$$f(\mu) = 440 \cdot 2^{\frac{\mu-69}{12}}, \tag{2}$$

and the length in millimeters is then

$$l = \frac{85750}{f(m+t)} - w \tag{3}$$

for a pipe that is closed at the top, or

$$l = \frac{171500}{f(m+t)} - 2w \tag{4}$$

for a pipe that is open at the top. t is the tuning headroom in semitones. The pipe will be sharp by approximately this amount so that it can be tuned down to the desired pitch with a tuning slide which fits over its end like a sleeve.

The mouth height h in millimeters, from the flue to the bottom of the upper lip, is

$$h = (3.018 - 0.233 \ln f(m))^5 \tag{5}$$

for a closed pipe and

$$h = \frac{550}{2^{\ln f(m)}} \tag{6}$$

for an open pipe, where $\ln x$ is the natural logarithm of x. In retrospect, after having observed several pipe organs in person and in images, I believe that this yields a mouth that is somewhat taller than typical, although it does work for the range of pipes that I made, and I have not had the opportunity to make detailed measurements of an exemplar.

The windsheet thickness, s, in millimeters, which is also the size of the flue opening, is given by

$$s = 2.409 \times 10^{-9} \frac{i^2 \cdot f(m)^2 \cdot h^3}{p} \tag{7}$$

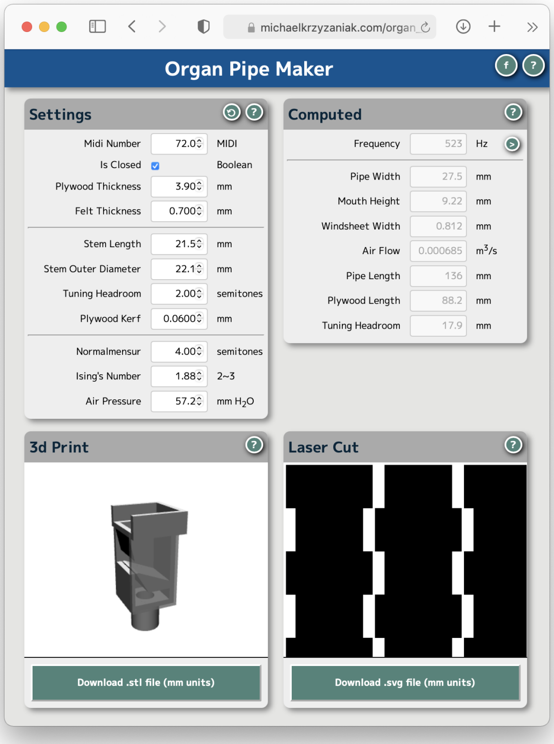
where i is the Ising number, which affects the timbre of the pipe and should normally be around 2. p is the static pressure of the air supply in inches of water (inH_2O) as measured by a u-tube manometer.[6] I will comment further on the pressure in Section 3.2 below.

---

[4] While three robots hardly constitutes a swarm, the more fundamental property of swarms is scalability; how it behaves when you add another agent, independent of the total number of agents, and I deemed three to be the minimum sufficient number for my purposes.

[5] This section relies heavily on information found here, http://www.rwgiangiulio.com/math/, which I will repeat for posterity along with my own observations and a few small modifications.

[6] This is the standard unit in this context; metric is used elsewhere throughout this paper.



Organ Pipe Maker

Settings

Midi Number	72.0	MIDI
Is Closed	☑	Boolean
Plywood Thickness	3.90	mm
Felt Thickness	0.700	mm
Stem Length	21.5	mm
Stem Outer Diameter	22.1	mm
Tuning Headroom	2.00	semitones
Plywood Kerf	0.0600	mm
Normalmensur	4.00	semitones
Ising's Number	1.88	2~3
Air Pressure	57.2	mm H_2O

Computed

Frequency	523	Hz
Pipe Width	27.5	mm
Mouth Height	9.22	mm
Windsheet Width	0.812	mm
Air Flow	0.000685	m^3/s
Pipe Length	136	mm
Plywood Length	88.2	mm
Tuning Headroom	17.9	mm

3d Print

Laser Cut

Figure 4: The organ pipe designer webpage that automatically generates organ pipe design files that can be 3d-printed and laser-cut.

The rest of the pipe scales linearly in its width. The dimensions of the air stem at the bottom of the pipe does not affect the sound, and can be made any size depending on the application.

Rather than solving the equations and individually designing each pipe, I built an interactive webpage that solves the equations and generates design files.[7] A screenshot of the webpage is in Figure 4. Users can enter the MIDI note number of the pipe they want, along with information about the timbre and the other free parameters from the equations above, and some information about the material they will be using to fabricate the pipe. The webpage uses this to generate two design files. The first is an STL file of the bottom 'mouth' part of the pipe, which can then be 3D printed. The second is an SVG file of the top 'resonator' and 'tuning slide' parts of the pipe which are intended to be laser-cut out of plywood and felt. The plywood parts need to be glued together and optionally painted, and can then be epoxied to the 'mouth' part to form a complete pipe. Complete instructions with detailed images are on the webpage itself.

I used this webpage and its default settings (shown in Figure 4) to make 24 pipes, ranging from MIDI note number 52 (E3, 164.8 Hz) to MIDI note number 75 (E♭5, 622 Hz). I made closed pipes, because to me they sound somewhat more 'cute', than open pipes, which is consistent with the general aesthetic of the project. In my opinion the pipes work well, they have the classic sound of a Gedackt or Bourdon stop on a church organ, and have consistent timbre across the range. The spectrum of a representative pipe is shown in Figure 5. The lower pipes speak somewhat more slowly than the higher ones, and in retrospect I suspect that designing the pipes with a rectangular cross section having a smaller width than depth would have solved this. It may also be related to the mouth height, and further prototyping may be necessary to produce very large pipes.

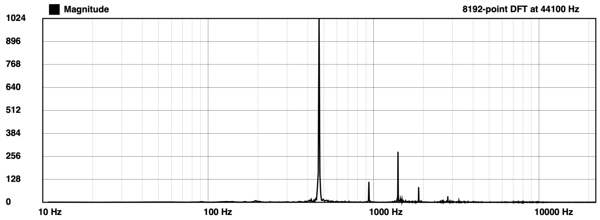


Figure 5: The spectrum of a representative pipe – the one with MIDI note number 70 (B♭4, 466.2 Hz).

The organ pipes can be heard reasonably unadorned in a video here.

- https://www.youtube.com/watch?v=SrkxFS3uiJE

3.2 Air Supply

I designed my pipes to operate at $p = 2.25\,\mathrm{inH_2O}$ ($57.15\,\mathrm{mmH_2O}$) of pressure. This is on the low side of typical, with Baroque organs often operating in the range of 2-3 $\mathrm{inH_2O}$, and modern organs around 5-10 $\mathrm{inH_2O}$. The air flow α consumed by each pipe, in $\mathrm{cm^3/s}$, is given by

$$\alpha = 20.45 \cdot w \cdot s \cdot \sqrt{p}, \tag{8}$$

which is calculated automatically by the organ-pipe designer webpage. My pipes ranged from 1670 $\mathrm{cm^3/s}$ for the lowest pipe and 574 $\mathrm{cm^3/s}$ for the highest, with the sum of all 24 pipes being about 26560 $\mathrm{cm^3/s}$. Virtually no obvious inexpensive source of wind, e.g. CPU fans, not even those marketed as 'high static pressure', produce a static pressure even as high as 1 inch of water. I did find one outlier, a counter-rotating server fan – the San Ace 60L 9CRLA Type from Sanyo Denki – which produces a nominal maximum static pressure of over 5.5 $\mathrm{inH_2O}$ (I measured it closer to 4.5), and, according to the datasheet, can produce a flow of about 29000 $\mathrm{cm^3/s}$ at 2.25 $\mathrm{inH_2O}$. This fan is therefore approximately powerful enough to drive all 24 of the organ pipes at once, and I used it in my design.

The fan is quite loud, so I enclosed it in a semi-soundproof box, depicted in Figure 6. The box is made of 22 mm plywood insulated on all inner surfaces with 20 mm thick mineral wool insulation and 3 mm thick felt. The area immediately surrounding the fan is double insulated. The ends of the box contain sound baffles that create a serpentine path which allows air to flow with no direct line-of-sight for sound to escape. Inside the box, halfway down its length, is a wall that separates the box into two sides; the fan is embedded in the wall and pressurizes one side, while the other side remains near atmospheric pressure. The wall itself is smaller than

---

[7] The webpage is at https://michaelkrzyzaniak.com/organ_pipe_maker/.

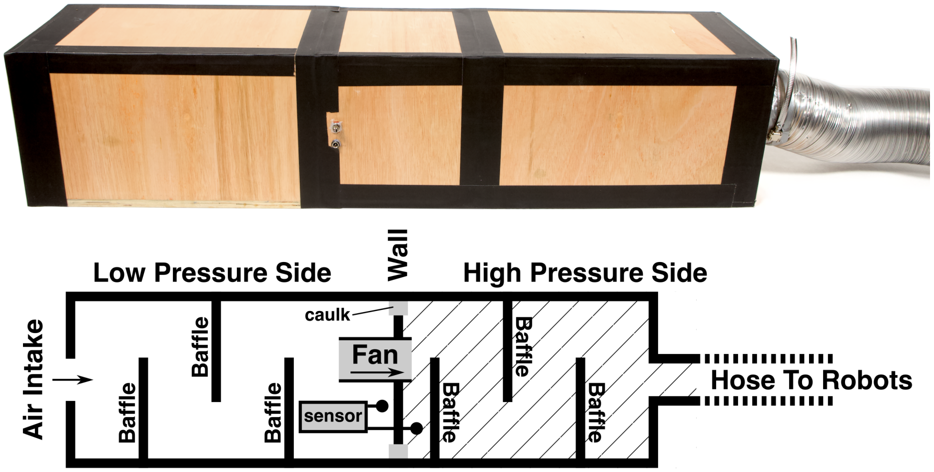


Figure 6: Top: Photograph of the organ air supply. Bottom: Schematic of the same. Air enters at the left, travels past sound baffles, is forced by the fan past a wall into the high-pressure side, passes more sound baffles, and then exits through a hose on its way to the wind chests.

the inner diameter of the box, leaving a gap all around; the gap is filled with silicone caulk, sealing the two sides of the box from one another while reducing the mechanical transmission of sound from the fan to the box. There is a pressure sensor that measures the difference between the two sides. The sensor is connected to an Arduino microcontroller which controls the fan, and uses a closed-loop PID controller with the sensor to maintain the correct pressure. This ensures that if several organ pipes are sounding at once, the fan will speed up to maintain pressure so the sounding pipes don't go flat. A large diameter outlet hose, which later splits into three smaller hoses, conducts air from the air supply to each of three the wind chests for the individual robots.

I wanted to know how effective the soundproof box is, so I set the robots up in a small room, in a configuration similar to that depicted in Figure 1, with an SPL meter 3 meters from the front of the air supply. The ambient sound in the room measured 32 dBA with the air supply off. Normally, in situ, the fan produces 2.25 $\mathrm{inH_2O}$ of pressure when the counter-rotating impellers spin at 17k and 19k RPM (inlet and outlet impellers, respectively, or 61% duty-cycle for each fan). With the box disassembled and the fan sitting on top of it spinning at this speed, the SPL meter reads 69 or 70 dBA. Then, with the box reassembled and the fan installed in it as intended and running at the same speed, the SPL meter reads 45 or 46 dBA. Therefore the box reduces the noise by 24 ± 1 dBA. It seems to me that most of the sound escapes from the walls of the box near the fan, and not from the air inlet or outlet, so in principle the box could be wrapped up in a second insulated box for further sound reduction. The approximate spectrum of the air supply is in Figure 8(c).

With the same setup, with the fan in the box and spinning as described, I turned three organ pipes on (MIDI note numbers 60, 64, and 67). The drop in pressure was negligible and the fans did not speed up to compensate. The SPL meter read 81 or 82 dBA, and therefore the pipes sound 36 ± 1 dBA above the air supply. To my ear, I find this sound level to be acceptable. Note that the robots, when listening through a microphone, can filter out the sound of the air supply, as described in Section 4.2, below. Note further that the air supply can be placed reasonably far from the robots with negligible loss of pressure owing to the large diameter outlet hose. This is because, according to Poiseuille's law, the pressure drop in the hose is linear in its length, but goes with $1/r^4$, so it is easy to choose a pipe with a large enough radius r to dominate the equation. I have thus on occasion placed the air supply in an adjacent room, thereby reducing the fan noise to effectively 0.

3.3 Wind Chests

The wind chests are the boxes that the organ pipes stick out of; assembled and disassembled ones are shown in Figure 7. They are simple wooden boxes that are pressurized by the air supply via a hose that connects to a hole at their rear. In the top of the wind chest there are 8 holes that accept the stems of the organ pipes. The holes have 3d-printed fittings that contain an o-ring so that the pipes seal well while also being removable.[8] Underneath each fitting is a pipe valve; this is a special electromechanical device that closes the hole with a felt pad, and opens it by pulling the pad away when a voltage is applied. I used the 12V Series II 50 Ohm model with 1 inch pallet from Peterson. The original Dr. Squiggles robot taps rhythms using 8 solenoids that have the same voltage and current requirements as these. So I simply removed the solenoids from Dr. Squiggles and wired the pipe valves in their place. Thus with no further hardware or firmware changes, Dr. Squiggles can open any of its valves to cause air to flow through the corresponding pipe.

---

[8] The organ-pipe designer webpage also has a utility for creating these fittings (by clicking the 'f' button on the top right).

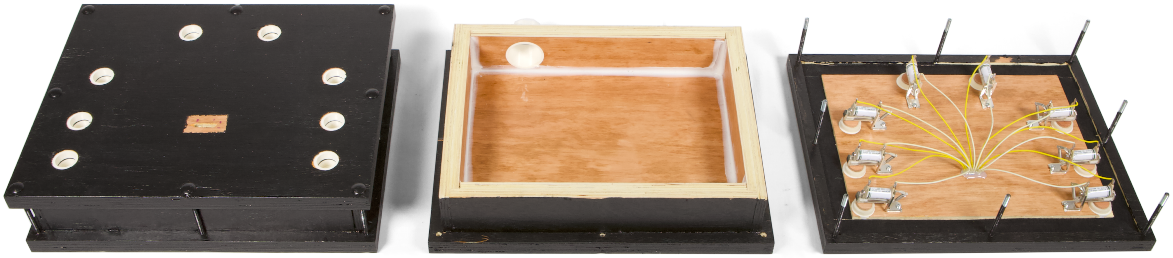


Figure 7: Left: One whole wind chest with 8 fittings for organ pipes; Center: Another with the lid removed showing the rear air inlet; Right: The inverted lid showing a pipe valve on each fitting.

The air-supply, wind-chests, and pipe designer webpage are further detailed in a video.

- https://www.youtube.com/watch?v=ATV3N4AFnjA

4 SELF AWARENESS

As a prerequisite for swarm behaviour or other interactive computer music applications, the robots should have some level of self-awareness [12]. This implies some way of a) discovering their own capabilities and b) examining the consequences of their own actions. Because Dr. Squiggles can listen through a microphone, it can figure out what notes its organ has, and what they sound like. Later, when it is playing music with other musicians, it can use this information to distinguish between its own sound and the sound of the other musicians. I will describe both in the following sections.

4.1 Note Discovery

The organ pipes can be easily removed from the wind chests and rearranged, so each robot needs to be able to figure out which note is controlled by each of its actuators. I implemented a calibration script in which each robot plays each of its notes in succession, listens to itself through an inexpensive USB lapel mic[9], and figures out the MIDI note number of each pipe. Based on Figure 5, one might suppose that the global maximum of the spectrum gives the fundamental frequency of the pipe. This is not always true, as I have observed circumstances in which, due to the acoustics of the room, some partial exceeds the fundamental. Consequently I implemented a simplified pitch-salience function σ as a function of frequency f, as

$$\sigma(f) = \sum_{j=1}^{5} \frac{1}{j} |DFT(\hat{jf})| \qquad (9)$$

where DFT refers to the discrete Fourier transform of a small recording (2^{13} audio samples at 44.1 kHz) of the organ pipe, and by $|DFT(\hat{jf})|$ I mean the amplitude of the bin whose frequency is closest to j times f. I evaluate this for every MIDI note number from 36 to 96 in 10 cent increments, converting MIDI numbers to frequencies using Equation 2. The MIDI note with the highest salience σ wins. I round the MIDI number to the nearest integer to obtain the presumed pitch of the pipe, and the rounding error is roughly the tuning error of the pipe in semitones. I have run this dozens of times using a variety of locations, microphones, and pipe arrangements, over the course of a year, and I have never observed it to misidentify the pitch.

4.2 Self Filtering

When a robot is playing music together with other human or robot musicians and listening through a microphone, it will hear a linear combination of its own sound, the sound of the other musicians, and whatever background noise is present in the environment such as the organ air supply fan. In the context of musical robot swarms and other intelligent music systems, it is often desirable for the robot to filter out the background noise and its own sound, so it only responds to the sound of the other musicians and does not become stuck in an undesirable feedback loop with itself. This is reasonably easy to accomplish in this context since the robots know what notes they are playing at all times, they know what each of their notes sounds like, and organ pipes produce a constant steady-state sound with negligible attack and decay.

I accomplish the desired filtering as follows. During the calibration described in Section 4.1 above, the robot makes a 2-second recording of the background noise in the room, and an additional 2-second recording for each of its 8 organ pipes. The recording of each pipe also contains the background noise, which needs to be removed so it can be handled separately. To do this, I break each recording into non-overlapping segments, apply a half-sine window to each segment, zero-pad each windowed segment to twice its length, calculate the DFT magnitude of each padded windowed segment, and average the DFT magnitudes across the segments to obtain the average spectrum for each recording. I then subtract the averaged background noise spectrum from each of the averaged organ pipe spectra. Later, during realtime interaction, as the robot is playing and listening through its microphone, I break the audio stream into segments with 50% overlap, apply the same padding and windowing, and subtract out the background noise spectrum and the de-noised spectrum of each of that robot's own currently

---

[9] The lapel mic takes the place of the contact mic used in the original Dr. Squiggles design.

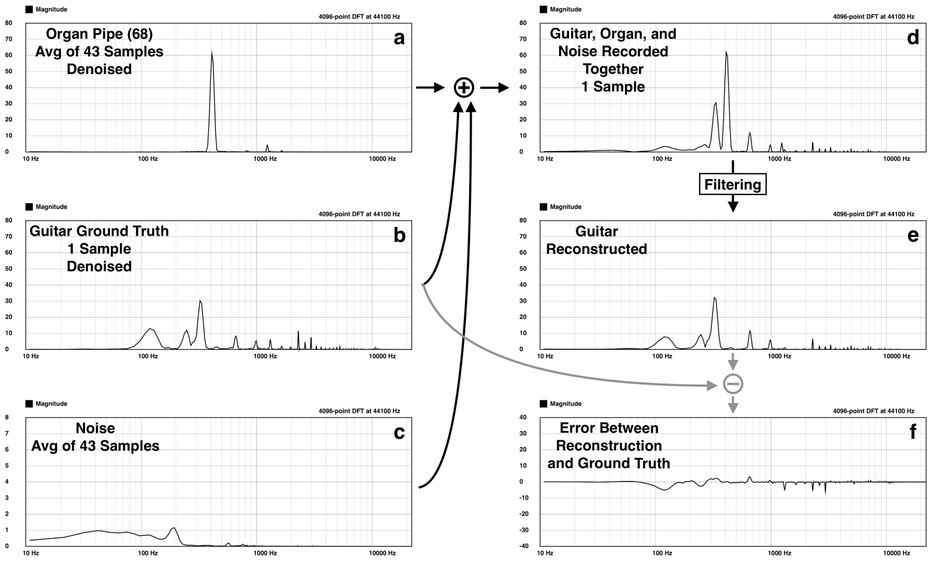


Figure 8: Operation of the self-listening filter. (a), (b), and (c) are three individual sound sources that are combined to produce (d). (e) is a reconstruction of (b) made by filtering (d), and (f) is the reconstruction error.

sounding pipes. This leaves just the sound of the other musicians, which can be re-synthesized into audio via the overlap-add method for further processing if necessary.

This process is depicted in Figure 8. The scenario is that the robot is playing one pipe (MIDI number 68, A♭4, 415 Hz), while a guitarist is also playing one note (the open high E string, MIDI number 64, E4, 330 Hz). On the left are the individual sound sources, consisting of

- (a) the averaged de-noised sound of the organ pipe obtained during calibration,
- (b) the sound of the guitar note recorded separately and de-noised for illustration, and
- (c) the averaged background noise in the room also obtained during calibration, and shown at 10x magnification on the Y axis for clarity.
- On the top right, (d) is from a recording of the robot and guitarist playing together, and is in principal the sum of (a), (b), and (c). Note that the most prominent feature is the spike at 415 Hz contributed by the organ pipe, which is about 6 dB louder than the most prominent spike contributed by the guitar.
- Subtracting (a) and (c) from (d) gives (e), the recovered sound of the guitar. Note the absence of the 415 Hz peak and the overall similarity to the ground truth (b).
- Subtracting (b) from (e) gives (f) the approximate reconstruction error, although note that the ground truth is a separate recording of the guitar note, and may be slightly different in amplitude and timbre despite being recorded under the same conditions, so one would not expect the error to be exactly 0 everywhere.

The note discovery algorithm and self-listening filter can be seen and heard in a video.

- https://www.youtube.com/watch?v=82ddn0kmpVQ

In my preliminary tests, this algorithm is sufficient for preventing the robots from becoming stuck in unwanted feedback loops with themselves.

5 APOTHEOSIS AND FUTURE WORK

In this paper I have shown how I constructed three miniature computer-controlled pipe organs using simple digital fabrication techniques. I have moreover shown some algorithms that imbue them with a small amount of self-awareness, which will aid in their use in intelligent interactive computer music systems. Hopefully this will make it easier for other computer musicians incorporate DIY pipe organs into their practice. In the future I will use these as the basis for the study of musical robot swarms, and human-swarm interaction. Swarms, in many ways, are defined by the constraints of the agents. Limiting the number of notes of each robots is an artificially imposed constraint that means that, in order to make music, each robot will have to know its own capabilities, listen to the other robots, and at each moment decide how it can use its particular capabilities to contribute to the overall mission of the swarm. This constraint will help lead the robots to the nice properties of swarms like structure and emergence. Some of my preliminary experiments in this direction are in a video.

- https://www.youtube.com/watch?v=rVeiIcR4YM4

6 ACKNOWLEDGEMENT

This work was partially supported by the Research Council of Norway through its Centres of Excellence scheme, project number 262762.

7 ETHICS STATEMENT

(1) It is important that new technologies are designed to benefit the users, not to exploit them, surveil upon them, or replace them. In my research, I am not seeking to replace human musicians with robots. I am seeking to make robot companions that humans could play music with in situations where they would otherwise play alone. The goal is that the robots would help people learn musical skills, or otherwise enrich their lives. The robots do not store any data in non-volatile memory, nor do they transmit data back to any server, nor attempt to identify the user or process any information that is not strictly necessary for moment-to-moment realtime musical interaction.

(2) Some of the items described in this paper are made of 3d-printed plastics. 3d printers are known to emit a large number of ultrafine plastic particles [20] which are toxic and can harm the environment. For example, these particles, when airborne, can enter your nose and travel up your olfactory nerve into your brain, where their long-term effects are unknown. I printed the parts using polylactic acid (PLA) which is derived from plant starch and is biodegradable with a median half life of 30 weeks in the human body [5]. I also used polyvinyl alcohol (PVA) support material, which likely more damaging, although the long-term effects of this material in the human body and environment are not well understood. In retrospect, the models could have been printed with far less support material, as there are only a few key layers that need support, and further work is needed to reduce or eliminate the support material from the designs for posterity.

(3) No robots were harmed in the execution of this research.

REFERENCES

[1] Aaron Thomas Albin. 2011. *Musical swarm robot simulation strategies.* Ph.D. Dissertation. Georgia Institute of Technology.

[2] Willi Apel. 1948. Early history of the organ. *Speculum* 23, 2 (1948), 191–216.

[3] George Ashdown Audsley. 1965. *The art of organ building.* Vol. 1. Dover Publications; Reprint edition.

[4] Tim Blackwell and Michael Young. 2004. Self-organised music. *Organised sound* 9, 2 (2004), 123–136.

[5] Dana Da Silva, Maya Kaduri, Maria Poley, Omer Adir, Nitzan Krinsky, Janna Shainsky-Roitman, and Avi Schroeder. 2018. Biocompatibility, biodegradation and excretion of polylactic acid (PLA) in medical implants and theranostic systems. *Chemical Engineering Journal* 340 (2018), 9–14.

[6] Athanasius Kircher. 1650. *Musurgia universalis sive ars magna consoni et dissoni.* Rome: Corbeletto.

[7] Michael Krzyzaniak. 2016. Manzetti’s Flute-Playing Automaton from Contemporary Sources. https://www.michaelkrzyzaniak.com/Research/Manzetti.php. (2016).

[8] Michael Krzyzaniak. 2016. Prehistory of Musical Robots. https://michaelkrzyzaniak.com/Research/PrehistoryMusicalRobots.pdf. (2016).

[9] Michael Krzyzaniak. 2016. *Timbral learning for musical robots.* Arizona State University.

[10] Michael Krzyżaniak. 2021. Musical robot swarms, timing, and equilibria. *Journal of New Music Research* (2021), 1–19.

[11] Michael Joseph Krzyzaniak. 2021. Dr. Squiggles AI Rhythm Robot. https://makezine.com/projects/dr-squiggles-an-ai-rhythm-robot/. *Make: Volume 76 (Behind New Eyes)* (2021), 88 – 97.

[12] Peter R Lewis, Arjun Chandra, Shaun Parsons, Edward Robinson, Kyrre Glette, Rami Bahsoon, Jim Torresen, and Xin Yao. 2011. A survey of self-awareness and its application in computing systems. In *2011 Fifth IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops.* IEEE, 102–107.

[13] Laura Maes, Godfried-Willem Raes, and Troy Rogers. 2011. The man and machine robot orchestra at logos. *Computer Music Journal* 35, 4 (2011), 28–48.

[14] Muhammad, Ahmad, and al-Hasan, sons of Musa. 1931. *The Instrument which Plays by Itself.* W. Reeves.

[15] Emile Quétànd. 1865. Curiositiés de la science. https://gallica.bnf.fr/ark:/12148/bpt6k589123j/f3.item.r=Manzetti.langEN.zoom. (22 novembre 1865), 3.

[16] Craig W Reynolds. 1987. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques.* 25–34.

[17] Curtis Roads. 1986. The Tsukuba musical robot. *Computer music journal* 10, 2 (1986), 39–43.

[18] Andy Rye. 2010. Hand Played by Robots. *hi-fi news & Record Review* (JUNE 2010), 14–17.

[19] Erol Şahin, Sertan Girgin, Levent Bayindir, and Ali Emre Turgut. 2008. Swarm robotics. In *Swarm intelligence.* Springer, 87–100.

[20] Brent Stephens, Parham Azimi, Zeineb El Orch, and Tiffanie Ramos. 2013. Ultrafine particle emissions from desktop 3D printers. *Atmospheric Environment* 79 (2013), 334–339.