The Telephonist:
a communication system for telematic performance

Iain Mott [i]
Universidade de Brasília - UnB, Brasília/DF, Brasil [ii]

Abstract - The Telephonist: a communication system for telematic performance
This article describes a communications system for telematic performance developed during the COVID-19 pandemic of 2020. Taking the form of an antique telephone switchboard, the system is designed specifically for a solo performer to interact with voice recordings, music and sound synthesis mechanisms and engage with local and remote audiences via telephone.
Keywords: Telematic art. Performance. Voice. Pandemic. Interactivity.

Resumo - A Telefonista: um sistema de comunicação para performance telemática
Este artigo descreve um sistema de comunicação para performance telemática desenvolvido durante a pandemia de COVID-19 de 2020. Na forma de uma central telefônica antiga, o sistema é concebido especificamente para uma performance solo para interagir com gravações de voz, músicas e mecanismos de síntese de som e engajamento, por telefone, com audiências remotas e locais.
Palavras-chave: Arte telemática. Performance. Voz. Pandemia. Interatividade.

Resumen - La Telefonista: un sistema de comunicación para la performance telemática
Este artículo describe un sistema de comunicación para performance telemática desarrollado durante la pandemia de COVID-19 de 2020. En forma de central telefónica antigua, el sistema está diseñado específicamente para una performance en solitario para interactuar con grabaciones de voz, música y mecanismos de síntesis de sonido e interacción con audiencias locales y remotas por teléfono.
Palabras clave: Arte telemático. Performance. Voz. Pandemia. Interactividad.

Introduction

This article deals with the design of an extended telephony system for live performance. The project has seen a long period of gestation and the initial idea was born in conversation with performer Simone Reis, well over ten years ago. The desire was to have a device that enabled Simone to record her voice and play it back, along with other recordings, as part of a performance. Also, to change the quality of the voice in real-time. I initially thought of using an audio patchbay, of the type used in recording studios, linked via a microcontroller board to software to perform these tasks. Different cord patchings would be registered by the microcontroler and used to launch different effects as well as triggering both sound capture and reproduction. The idea slept. It later re-emerged when a relation was made between the studio patchbay and its origins in telephony. An idea was forged of the performer as telephone operator and Simone Reis and I gave the project the name *A Telefonista* (*The Telephonist*, in the feminine gender).

The character of telephonist would express a form of omnipotence, able to receive calls and make connections, as well as intercept conversations, messages, ghosts and memories from the ether with the aid of a switchboard. An antique hotel switchboard was purchased.

The Telephonist was conceived as telematic theatre performed by Simone Reis, in front of a live audience, involving communication with and between audience members and with other people or telephone services beyond the performance space. This would be achieved by way of analogue telephones distributed amongst the audience and connected to the performer's switchboard. The switchboard would itself be connected to the external telephone network and act as a hub for communications in the performance, mediating connections and transfers between all parties. Conversations would audible to the audience via loudspeakers in the performance space.

The switchboard would also serve to connect to virtual extensions with, for example, recorded messages or music and as such, could free the performer to leave her station, perform movements or interact with the audience. In the early conception it was never considered that the performance itself might be transmitted and with a remote audience capable of engagement with the performer. This possibility and indeed potential necessity, called attention to itself during the pandemic.

After abandoning an earlier idea of restoring the electromechanical components of the switchboard to function as it would have done when it was built in the 1940s or 50s, I began, in March 2020 and at the beginning pandemic in Brazil, to research how the switchboard might be used as an interface to a computerised telephony system.

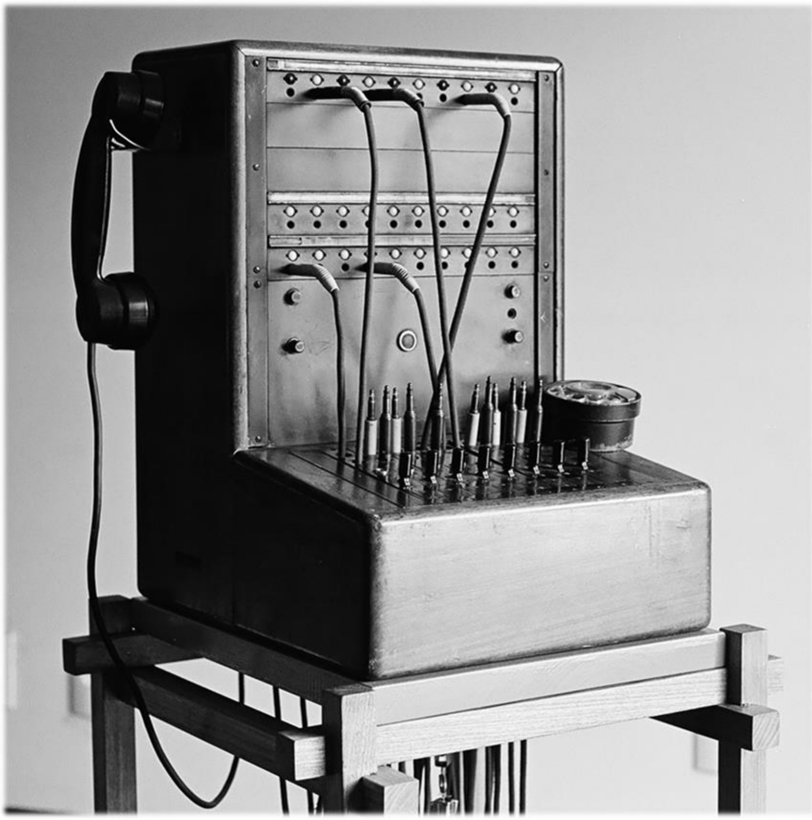

Figure 1: Restored Ericsson switchboard and stand

The open-source and widely used *Asterisk* software system was identified as an ideal choice for the telephony requirements of the project along with the *SuperCollider* audio language to perform high-fidelity audio processing and sound synthesis. The Arduino Mega microcontroller was chosen for the interface between switchboard and software.

Once these elements were defined and in the context of the pandemic, it became clear that audience involvement need not be restricted to local interactions but could be open to a wider remote audience via internet video streaming and public telephone numbers assigned to the switchboard. The project has taken a path that embodies both options of performance. A third option exists for the system where it can act as an autonomous agent, for example receiving calls and delivering content via interactive mechanisms such as voice recognition or caller selection of options. Further, the system could potentially take a yet more aggressive and active role, making its own scheduled calls to individuals.

Figure 2: Cord plugs

This article describes the methods used to realise these goals developed between March and August, 2020. The performative content to be delivered and engaged with by the system as well as the aesthetics of the project its design, will be the subject of future texts.

The switchboard and its original use

The switchboard used in *The Telephonist* (Figure 1)[1] is stamped with the make *Ericsson Brasil* and came from a hotel in the city of Maringá in the Brazilian state of Paraná. It is part of a line of manual switchboards produced by Ericsson from 1946 into the 1950s and the models are documented in the L. M Ericsson review (Engqvist, 1946, 1950; Siewert, 1953). The switchboard was designed to follow a standardised operational procedure ("General Switching Discussions", 2020; War Department, 1944). In Figure 1 above, one can see on the vertical board, three rows of sockets or *jacks* with a lamp above each as well as two buttons and two rotary switches beneath them. The top row of plugs would have been used to connect to with external exchanges or *trunks*. Only five of the ten sockets in the upper row

[1] All photos by Iain Mott.

were connected to circuitry and consequently there are only five lamps in the upper row. The two rows of jacks below were apparently connected to local telephones in the rooms of the hotel and many of the room number labels remain. The other items on the panel include a jack for headset use, a buzzer to register incoming calls and two switches and buttons for power management and the operation of a *night bell* to alert a sleeping telephonist. On the more horizontal board, two rows of cords and their terminal plugs may be seen, some inserted into jacks. The upper cord plugs are pale in shade and the lower plugs are dark. The upper and lower rows form eight *cord-pairs* and the lower cord of each was known as the *answer cord* and upper cord was known as the calling cord. Immediately beneath each of the eight cord-pairs is a three-way *key* with a black Bakelite handle (see also Figure 3) and the three positions of which were used to ring the bell on local extensions and to speak and listen to connected parties.

When an incoming call was received, a buzzer would sound and the *supervisory* lamp above the jack of the incoming call was illuminated. The telephonist would take the answer cord of an unoccupied cord-pair, insert it into the jack and shift the corresponding key of the cord-pair to the forward position to speak with the caller. The answering of the call would have the effect of switching off the lamp. If asked by the caller to make a transfer, the telephonist would take the calling cord of the pair and insert it into the jack of the desired local extension or a jack connected an external trunk. To gain the attention of a person on a local extension, the telephonist would shift the key to its spring-loaded back position. This would have the effect of sending an alternating current down the line to sound the bell on the telephone of the extension. A crank generator is also included on the right-hand side of the Ericsson switchboard to perform this task if no such suitable power supply were available. When the desired transfer was to an external number and the trunk was a manual exchange, the insertion of the calling cord into the jack of the trunk would sound a buzzer and lamp at the external exchange. The local operator would move the key forward to speak with the attendant operator at the exchange and read the number of the transfer to the operator to initiate the transfer. Once the remote operator confirmed that the transfer request was accepted, the local telephonist would return the key to the central, *do nothing*, position to allow the call to proceed. If the external exchange were automatic, there would be no need to speak with an external operator. The local telephonist would simply dial the desired number after having inserted the calling cord into the trunk jack. As with the assisted trunk call, the operator would return the key to the central position if the transfer was accepted to complete the transfer. Once the call between the two parties ended, the lamps above the connected jacks would switch on, signalling to the telephonist that the cords could be returned to their resting position on the horizontal panel of the switchboard and in doing so, the cords would be retracted by cord-weights under the switchboard (Figures 9 and 10).

The performing switchboard - a system overview

As mentioned in the introduction, the *Asterisk* (Sangoma Technologies, 2020) communications software system was chosen to perform telephonic managements tasks and the keys, jacks and cords of the Ericsson switchboard serve as an interface to Asterisk, by way of an Arduino Mega microcontroller board (Arduino, 2020), rather than channelling and switching analogue audio signals directly as was done in its original use. Asterisk is open-source software and is used to develop internet-based PBX systems. PBX stands for Private Branch Exchange and such systems allow for the management of internal telephone networks and their connection to external networks achieved via Voice over IP, ISDN[2] and analogue channels, among others. Asterisk, which runs as a server on a networked computer, may also be configured to deploy additional services such as internal voice mail, menu-driven interactive mechanisms to process incoming calls or to run surveys, for example, as well as audio and video-conferencing. The compact raspberry pi 4 (Raspberry Pi Foundation, 2020) with its standard Debian-based Linux operating system was chosen to serve as the onboard switchboard computer and run Asterisk and all other software besides that used in the Arduino interface.

[2] Integrated Services Digital Network.

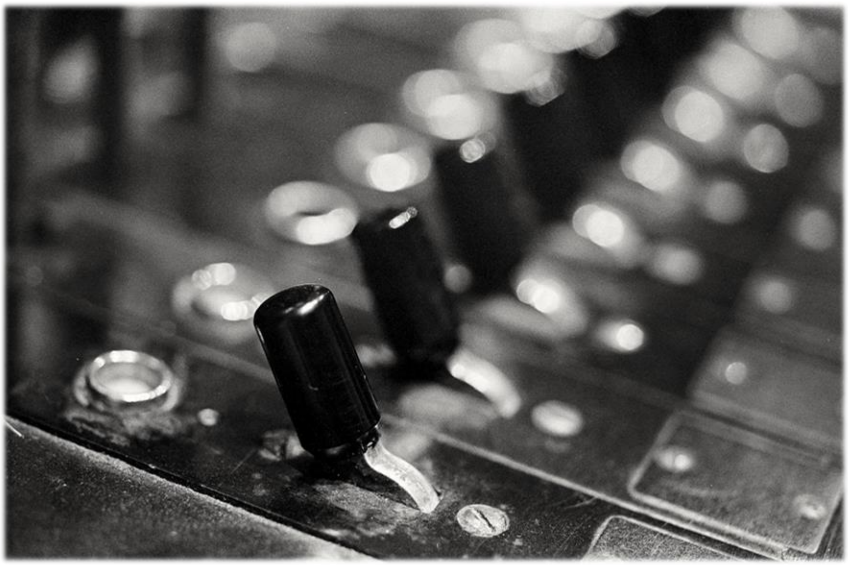

Figure 3: Channel keys in the forward talk position

While there was a desire to modernise the Ericsson switchboard, it is not meant to function solely as a PBX. It has been developed as a performance device, fulfilling the duties of a PBX while presenting audible material—pre-recorded and synthesised voice, live conversations, music and sound effects—to an audience, both in attendance and online.

Asterisk does not itself reproduce audio, although it can deliver and channel it. In order for audio *passing through* the switchboard during the performance to be heard, Asterisk must direct the audio to a telephone; either a physical telephone or a virtual one. A number of *Linphone-daemon*[3] software-telephones, or *soft-phones*, have been configured to operate as *services* on the computer running Asterisk. The audio received by the soft-phones is to be channelled to an amplifier and speakers at the performance and over the internet to a remote audience online. In total, eight separate soft-phones are used, one for each of the eight cord-pair channels on the switchboard and in this way, up to eight separate conversations or other audio sources may auditioned and controlled by the switchboard at any one moment.

Because Asterisk audio is usually destined to be sent over internet and telephone networks, it is passed through a choice of data compression codecs for efficiency of transmission and voice over IP telephones or interfaces for analogue phones receiving such audio must have the same codecs available to decode the signal. This is the case with the Linphone-daemon. If such sound were exclusively played to the local audience over

[3] Console soft-phone server of the Linphone software by Belledonne Communications (2020)

loudspeakers, its quality and frequency bandwidth would be limited by those codecs. For this reason, among others that will be discussed below, the audio processing and multimedia language Supercollider (Wilson; Cottle; Collins, 2011) was chosen to deliver full bandwidth audio in communication with Asterisk. SuperCollider, like Asterisk and the Linphone-daemons, runs as a server and Asterisk has been configured to communicate with the SuperCollider server by sending messages over the local network in response to telephony and switchboard events. Details of this mechanism will be discussed below.

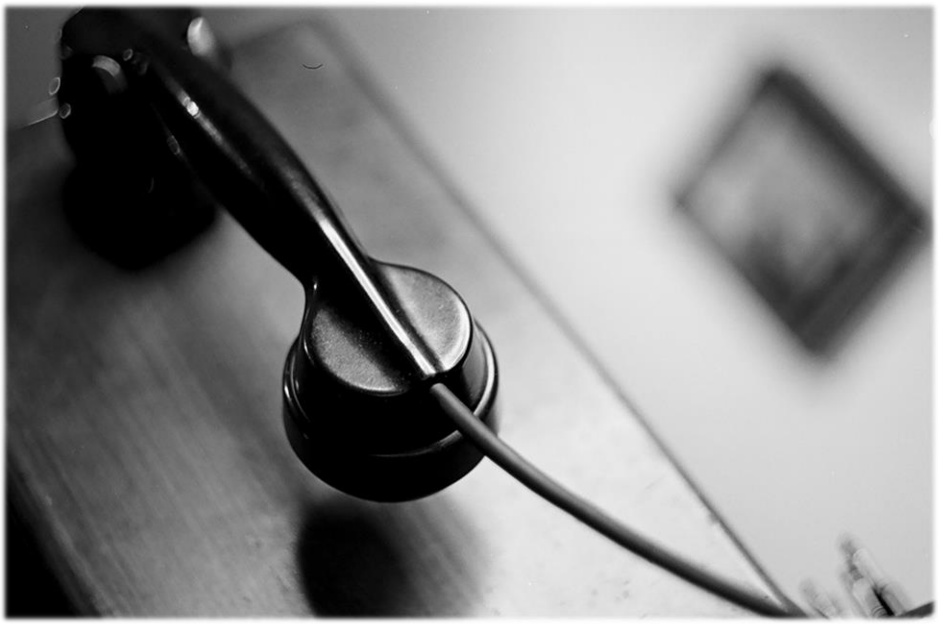

Figure 4: Switchboard handset

The switchboard is not the only telephony device to be used in the live performances. Analogue telephones, including the one shown in Figure 5, are to be distributed amongst the audience. These phones need a networked interface to connect with Asterisk and for the development process a *SPA2102 Phone Adapter with Router* (CISCO, 2020) has been used and to which two analogue telephones can connect. In order for the rotary dial of the telephone in Figure 5 to be operable with modern networks, a low-cost pulse to dual-tone multi-frequency (DTMF) converter was purchased.[4]

[4] The device counts the number of pulse signals generated when a digit is dialled and converts that digit to a two-tone signal used by all modern analogue fixed-line telephones to transmit the individual digits of telephone numbers.

Figure 5: Analogue rotary telephone

To enable an external audience to interact with the performer and contribute their own voices to the performance, a so-called *virtual* or *Direct Inward Dialling* (DID) number has been leased from a service provider. Such numbers may be leased for a large number of cities internationally and are used to grant low-cost remote telephone access. For example, the central office of a business based in a major city may lease individual DID numbers in all regional capitals of a country or in various international locations.

People wishing to contact the company may do so by using the local number for the cost of a local call. This is achieved by forwarding the audio of the call over the internet. In the case of *The Telephonist*, this forwarding is established by the DID service between the calling telephone and a certain extension configured in Asterisk. Asterisk, must thus be made available on the internet to external connections from the DID service on a fixed IP address and by opening specific ports its local router.[5] To facilitate interaction with remote audiences, the performer/telephonist must either inform the audience of the DID number or numbers verbally or display them on a card or panel during the performance.

[5] For development, a single DID number for the Brazilian Federal District has been leased from the service provider Carry My Number (https://www.carrymynumber.com).

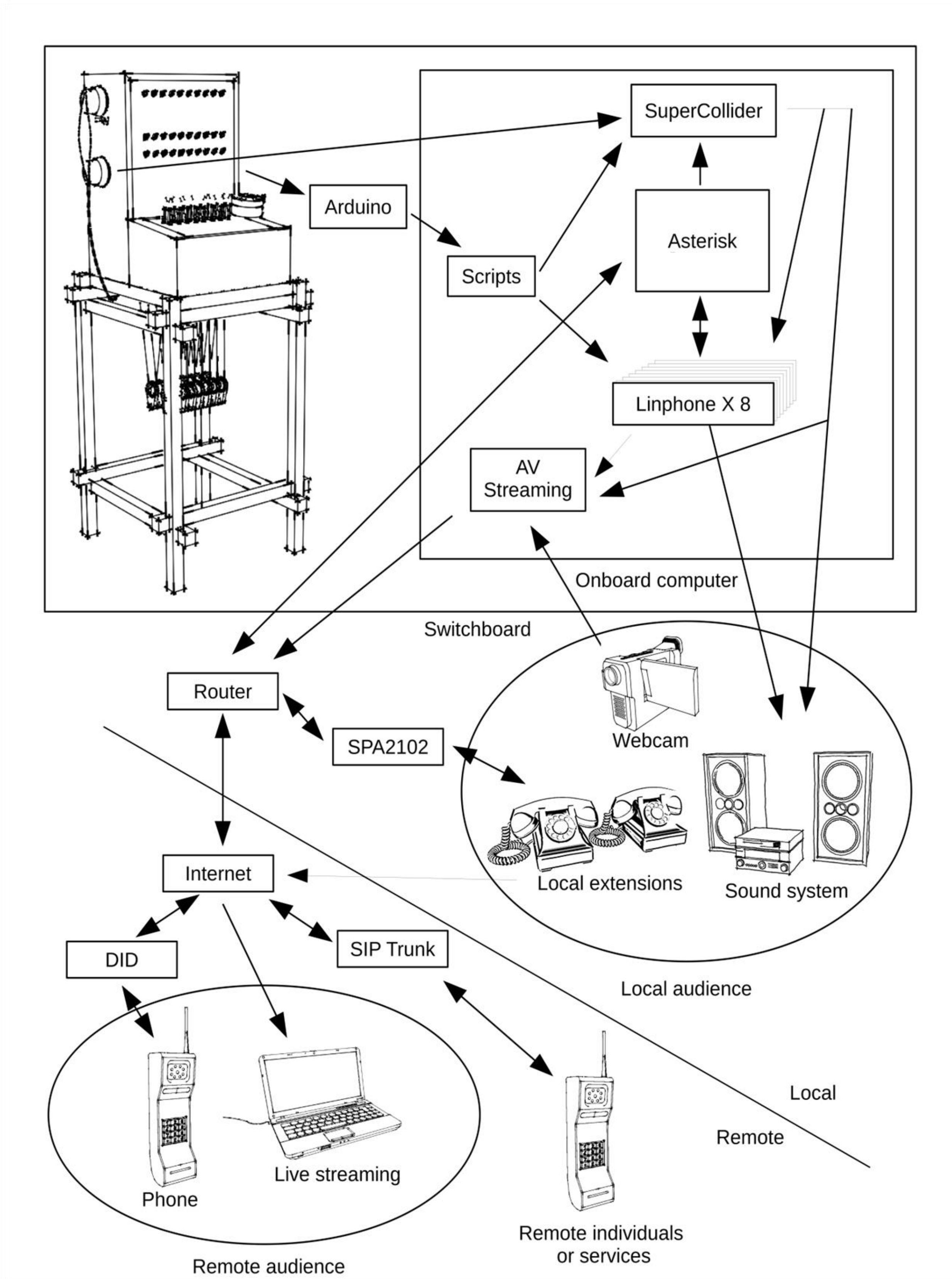

Figure 6: System schematic

To make outgoing calls to external telephones or make transfers to them, a *SIP trunk* service is used. No physical telephone line is involved, only an internet connection. SIP stands for Session Initiation Protocol and is used in voice and video calls as well as messaging over the internet between SIP equipped devices. The eight Linphone-daemons and the SPA2102 analogue telephone interface used in the project as well as Asterisk itself are such *devices*. While SIP device to SIP device communications may be made over the internet for no charge, a *SIP trunk* service can be used to connect a SIP device with any analogue telephone on fixed or mobile networks and this may be done over large distances at economical rates. The Linphone-daemons and SPA2102 used in the *Telephonist* do not connect with the SIP trunk directly, although they could if wished. Rather, they have SIP connections with Asterisk which in turn has a SIP connection to the trunk. When the telephonist/performer dials a number from the switchboard, she in effect uses one of the eight Linphone-daemons operating on the switchboard's computer, facilitated by the Arduino interface. The SIP connection between the Linphone-daemons and Asterisk allows Asterisk to process the number. If the number is characterised as not to be that of a local extension or other private process such as a sound synthesis request on SuperCollider, Asterisk directs the call to the trunk over the internet with a certain IP address and password. The SIP trunk then facilitates the connection between caller and external telephone.[6]

The telephonist's voice is captured by a dynamic microphone capsule of in the switchboard hand-piece, of the type used in stage microphones. The capsule is balanced which allows for a long cable length and it is envisaged that the telephonist may require the use of the hand-piece at a extended distance from the switchboard. A balanced audio external interface is also required for the Raspberry Pi 4 to receive the signal.

The audio captured by the Raspberry Pi 4 is first channelled to SuperCollider, which may process the audio with, for instance, a band pass filter to achieve a "telephone voice" effect or a fast-fourier transform process to change the pitch of the voice, before passing the signal on the front-of-house loudspeakers and to any one of the eight Linphone-daemons in use by the telephonist. As mentioned, via the switchboard, the telephonist can answer or make calls to individuals both within and outside of the performance space, dial pre-recorded

[6] The Integravoip SIP trunk service has been used during development to make outgoing local and national calls (Integravoip Tecnologia, 2020).

voice messages and dial either pre-composed audio or synthesis algorithms. The way SuperCollider receives communications and performs its audio tasks will be discussed further below.

All of the audio that is reproduced by the front-of-house loudspeakers and the video from a camera used to capture to performance, will be channelled to a streaming mechanism. This aspect of the project has yet to be developed however it will be very straight forward to realise using streaming capacities of the Linux command line utility *FFmpeg*. It is planned to use FFmpeg to unite the audio channels with the video from a webcam and stream directly to YouTube or other distribution service to reach a remote audience (FFmpeg, 2020). Figure 6 shows a schematic diagram of the system described in this section. Subsequent sections will further elucidate its functioning.

The Switchboard interface

The design of the circuit board for the Arduino Mega interface to the switchboard was done in communication with Jim Sosnin in Melbourne, Australia, via Skype and Jim was responsible for the circuit design and the great majority of the C programming of the Arduino to which the circuit is attached. I tested and repaired the dial and the various plugs, jacks and keys and relayed information to Jim about the wiring and contacts. He in turn sent sketches of the circuits and component lists for me to build. Most aspects of the switchboard are connected to the interface including the dial, the 25 jacks originally used for trunk and internal lines, the 16 cords, the 8 keys and the mechanical crank. The compiled C program loaded on the Arduino registers actions on the switchboard and transmits the actions as serial data to the onboard computer via a USB link. Communication is unidirectional from the interface to the computer. In the future it would possible to control the lamps of the switchboard with the Arduino in two-way communication, however this was not considered. Open-source C code to register the switchboard dial was obtained online[7] and integrated with Jim's code.

Serial data transmitted from the interface includes four main message types: *crank*, *dial*, *key* and *patch*. Crank messages are a single character *r* and are sent once the crank generator voltage rises above a certain mark. Dial messages are designated by the letter *d* followed by

[7] https://www.instructables.com/id/Interface-a-rotary-phone-dial-to-an-Arduino

the digit dialled (0-9). Key messages may be one of three letters, *f* (forward), *c* (centre) and *b* (back) for the three positions of each key, followed by its cord-pair number (0-8). Patch messages communicate connections and disconnection made with specific cords and jacks. They start with the letter *p* followed by a plus or minus sign and the jack number (1-25) and then the cord number (1-16) after a comma delimiter. *Positive* messages are connections made between a particular cord and a jack. *Negative* messages are disconnections.

Figure 7 shows the Arduino and circuit boards in an unclosed box during installation in the switchboard. The outside of the box is covered screw-contact strips for ease of connection with the switchboard hardware.

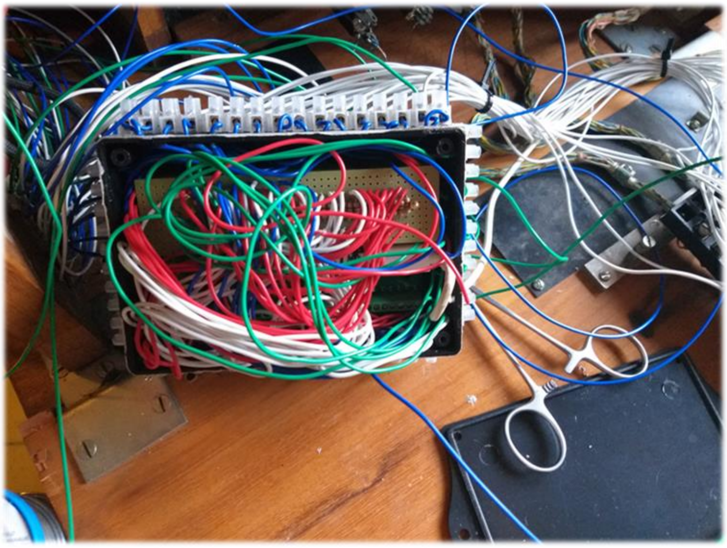

Figure 7: Installation of interface

Switchboard operation

As mentioned, there is no actual audio routed via the patch cords of the switchboard and only the events of operation are transmitted to the onboard computer. Consequently there is a lot of flexibility in the way that actions are interpreted. The original operation of the switchboard, detailed above, has been simplified to some extent and made a little more *noisy*, for the sake of performance. The concept of cord pairs, for example, has been retained, however there is no need to designate one as an answer cord or the other as calling cord as a script running on the onboard computer keep track of cord usage. Either cord of a pair may be used to answer when the pair is not engaged and either may be used to make calls or transfers.

Since there are no supervisory lamps in operation in the current usage, an acoustic means of alerting the telephonist to incoming calls has been implemented. When Asterisk detects an incoming call it first plays a voice-synthesised message stating that there is a call on a certain channel. This is done with the aid of SuperCollider and the communication method will be explained below. Immediately after the message is heard, Asterisk dials, in addition to one of the Linphone-daemons for the telephonist to answer, a particular extension to sound an electromagnetic bell inside the switchboard connected to Asterisk via the SPA2102 interface.[8]

Another divergence from the original operation is that all of the jacks have the same functionality and are not connected to any local extension or external exchange. When the switchboard notifies the telephonist of an incoming call on a particular cord-pair channel,[9] the telephonist simply takes one of the cords of the cord-pair and inserts it in to any available jack to receive the call.

Because the jacks are not physically connected to any particular telephone or exchange, a number must be dialled for Asterisk to know what extension or external telephone to call. To make a call or transfer, the telephonist inserts the appropriate cord into any free jack before dialling the number. For the call to initiate, she must either pull the key for the cord-pair to back position or operate the crank generator. In the case of the generator and purely for dramatic effect, it rings a second electromagnetic bell inside the switchboard.[10] This is a non-virtual operation. The switchboard's original crank generator still works and generates an alternating current of just over 100V to ring the bell.

Finally, as well as using the back-position of the cord-pair keys to initiate calls, the keys are used to control call transfers. When the telephonist receives or first makes a call, the key for the channel used should be in the central position. She speaks with the person on the line with the key remaining in the central position and if a transfer is requested, inserts the

[8] For simplicity, this use of the SPA2102 is not shown in Figure 6. Essentially the bell operates as a physical telephone inside the switchboard, with only the alarm in use as the call is taken by one of the Linphone-daemons and not a physical phone when the telephonist attends the call.

[9] When receiving a call, Asterisk will first check to see if the first cord-pair channel is free to use. If not, it will test the consecutive channels until a free cord-pair is identified. If all pairs are in use, it will keep the caller on hold until a channel becomes available.

[10] While it would be possible to use the incoming call bell for this task, a second bell was installed to avoid the necessity of voltage protection circuitry for the Arduino interface.

spare cord of the pair into an unoccupied jack before operating the spring-loaded back position of the key or the crank to initiate the transfer. She then speaks with the transferee and with the original caller or called individual placed on hold. Then, if the transfer is accepted, she pushes the key to forward position to finalise the transfer. As well as making the connection, the telephonists voice is muted in the established connection. As mentioned however, the ensuing conversion is audible to both local and remote audiences. If the telephonist wishes to rejoin the conversation in a three-way connection, she shifts the key back to its central position. Any call or transfer may be terminated at any moment by simply removing the connective cords.

Audio mechanisms and routing

Serial data sent from the Arduino interface is read by a *shell script*[11] named *serialtel.sh* which, in addition to keeping track of cord and jack usage, has the function of controlling Linphone-daemons to receive calls or make calls and transfers via secondary scripts. It is also used to send messages to SuperCollider using the OSC protocol (Schmeder; Freed; Wessel, 2010). These direct communications between serialtel.sh and SuperCollider are however secondary to commands sent to SuperCollider via telephone numbers mediated by Asterisk. The telephonist may, for instance, dial a particular synthesis process using a particular extension. The serialtel.sh script first sends the extension number to an available Linphone-daemon and by way the daemon's SIP connection to Asterisk, Asterisk references the desired extension. If the extension is of a certain class belonging to SuperCollider algorithms, in this case an extension beginning with the digit 3, Asterisk then launches a specific OSC message associated with the extension to initiate the synthesis process. The pitch shifting and *telephone-voice* effects mentioned above work in this way. Other SuperCollider processes are interactive and after initially dialled, subsequent jack insertions with the other cord of the used pair, modulate the synthesis algorithm.[12] In this way the switchboard emulates modular synthesisers of the past.

[11] A type of program written in a specific scripting language and interpreted by the *shell* (in this project the *Bash* shell and scripting language is used), a program that acts as interface to the operating system.
[12] Specifically, *node proxy definitions* in SuperCollider are used to modify and transition between algorithm states (Rohrhuber; de Campo, 2011).

Neither Asterisk, which can reference and channel sound, nor the Linphone-daemons, which capture, channel and ultimately reproduce sound, offer much in the way of control mechanisms for audio levels. It was therefore necessary to implement a system to control all aspects of calls played over the front-of-house loudspeakers or streamed over the internet, for example, to control the levels of dial tones, ring tones, voice synthesisers, the speaker's voices, pre-recorded announcements and even to correct individual differences in volume between local extensions. This task was complicated by several factors. The first being that SuperCollider on Linux requires the JACK ("JACK Audio Connection Kit", 2020) to operate and not PulseAudio (Debian, 2020) which is the default sound server on many contemporary Linux operating systems, including that used on Raspberry Pi 4. The Linphone-daemons however use PulseAudio exclusively and not JACK. Further, the Asterisk server is run by the virtual user *asterisk* which has no access to PulseAudio as ordinarily, it would be expected that client software such as Linphone softphones do the interactions with the sound server and in a very rudimentary way. To resolve the problem of the separate user spaces of Asterisk and that of the audio reproduction and control software, the solution was to send all control messages from Asterisk as OSC commands on the local network and to which the SuperCollider audio server listens on the chosen port. In this way, not only can SuperCollider synthesis be controlled, the server can also be used to make system calls to launch scripts controlling PulseAudio volumes and other tasks as the principal user—that is, as the user *iain* rather than the user *asterisk*. Additionally, further control messages are launched by serialtel.sh as the user user *iain* in response to switchboard actions.

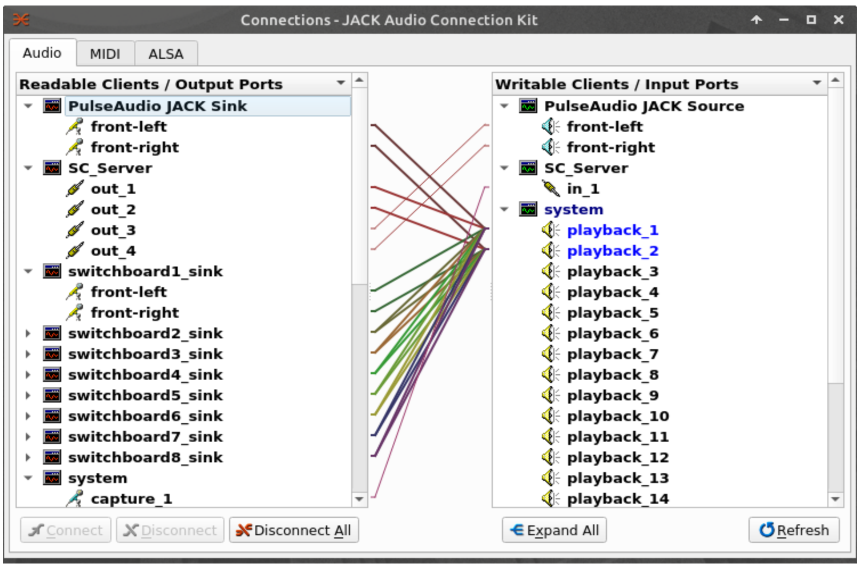

Figure 8: JACK routing

To perform audio routing between PulseAudio and JACK complaint software, the Linux package *pulseaudio-module-jack* was installed. The package enables the PulseAudio utility *pactl* to create any number of JACK *sinks* or *sources*. Eight such sinks are created on initialisation and these are used individually as audio outputs for the eight Linphone-daemons. As such, each Linphone-daemon has a means of independent volume adjustment, again with the aid of pactl, operated by scripts. The default *PulseAudio JACK Source* is used and is accessed by all Linphone-daemons as the audio input source. The current JACK routing configuration used in the project is displayed in Figure 8 (Debian, 2020; "JACK Audio Connection Kit", 2020).

Additional switchboard elements

In order for the switchboard to be used it needs to be either wall-mounted or placed onto a special stand. This is to allow cords to retract when not in use by way of a simple weight and pulley system. Wall-mounting is not practical for performance and I resolved to design and build my own stand out of wood. The stand, shown in figure 10, uses *half lapping joints* and is loosely based on the furniture design of Gerrit Thomas Rietveld (Daniele Baroni, 1978). The stand was given a natural lacquer finish with the baseplate painted with orange spray paint. Fishing sinkers and washing line pulleys were used for the cord-weight mechanism (Figure 9).

Figure 9: Cord-weights and pulleys

Figure 10: Switchboard on stand

Conclusion

The project *The Telephonist*, at the time of writing this article, is poised to commence development of the creative content of the performance with the involvement of Simone Reis. The process of developing the script for performance, recorded spoken word material and sound compositions, will be facilitated by the switchboard itself. Simone will soon commence improvising an experimenting with the device and the requirements of the performance will become manifest.

It is profoundly hoped that by the time the project is performed, the COVID-19 pandemic will be over. The development of certain telematic aspects of the project in response to the pandemic, will however, no matter the outcome, serve the project well in reaching audiences both in Brazil and abroad.

Acknowledgements

I would firstly like thank Jim Sosnin for his excellent assistance and good virtual company, all along the development process. Jim's work on the switchboard interface has been essential to the project, as has his general advice and suggestions on all things audio. I would also like to thank members of the *General Switching Discussions* board of the *Classic Rotary Phones* online forum for sharing their knowledge on the functioning of the Ericsson switchboard and the sourcing of parts. The project gratefully acknowledges the assistance of the *Fundo de Apoio à Cultura* of the Brazilian Federal District (FAC DF).

Bibliography

ARDUINO. **Arduino Mega**. Available at: https://store.arduino.cc/usa/mega-2560-r3 . Accessed on: 14 set. 2020.

BELLEDONNE COMMUNICATIONS. **Linphone**. Available at: https://www.linphone.org/products . Accessed on: 14 set. 2020.

CISCO. **SPA2102 Phone Adapter with Router**. Available at: https://www.cisco.com/c/en/us/products/unified-communications/spa2102-phone-adapter-router/index.html . Accessed on: 14 set. 2020.

DANIELE BARONI. **The Furniture of Gerrit Thomas Rietveld**. New York: B E S Pub Co, 1978.

DEBIAN. **PulseAudio**. Disponível em: https://wiki.debian.org/PulseAudio . Accessed on: 4 set. 2020.

ENGQVIST, Erik I. **Manual Private Branch Switchboards with Cords**. L. M. Ericsson Review, v. 27, n. 2, pp. 36-45, 1950.

ENGQVIST, Erik I. **New Series of Manual LB Switchboards with Cords**. L. M. Ericsson Review, v. 23, n. 3, pp. 250-258, 1946.

FFMPEG. **Encoding for Streaming Sites**. Available at: https://trac.ffmpeg.org/wiki/EncodingForStreamingSites . Accessed on: 14 set. 2020.

General Switching Discussions. Classic Rotary Phones. Available at: http://www.classicrotaryphones.com/forum/index.php . Accessed on: 2 set. 2020.

INTEGRAVOIP TECNOLOGIA. **IntegraVoIP**. Available at: https://www.integravoip.com.br . Accessed on: 14 set. 2020.

JACK Audio Connection Kit. Available at: https://jackaudio.org/ . Accessed on: 14 set. 2020.

RASPBERRY PI FOUNDATION. **Raspberry Pi 4**. Available at: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/ . Accessed on: 14 set. 2020.

SANGOMA TECHNOLOGIES. **Asterisk: Open Source Software**. Available at: https://www.asterisk.org . Accessed on: 14 set. 2020.

SCHMEDER, Andrew; FREED, Adrian; WESSEL, David. **Best Practices for Open Sound Control**. In: LINUX AUDIO CONFERENCE, 2010.

SIEWERT, O. **Manual CB-switchhoard ADE 1210**. L. M. Ericsson Review, v. 30, n. 1, pp. 17-18, 1953.

WAR DEPARTMENT. **Procedure Common to all Switchboards**. FM 24-75 Telephone Switchboard Operating Procedure. Washington, D.C.: The United States War Department, 1944. pp. 6-22. Disponível em: https://archive.org/details/FM24-75/page/n3/mode/2up.

WILSON, Scott; COTTLE, David; COLLINS, Nick (Org.). **The SuperCollider Book**. Cambridge, Massachusetts & London: MIT Press, 2011.

Artigo recebido em 14/09/2020 e aprovado em 18/11/2020.

DOI: https://doi.org/10.26512/vozcen.v1i02.34032

Para submeter um manuscrito, acesse https://periodicos.unb.br/index.php/vozecena/

[i] Iain Mott - Artista sonoro inglês-australiano e professor adjunto no Departamento de Artes Cênicas da Universidade de Brasília (UnB) na área de sonoplastia teatral e voz e performance. Possui doutorado pela University of Wollongong, Austrália (2010) intitulado Sound Installation and Self-listening e graduação, Bachelor of Science (BSc) em microbiologia, da University of Queensland (1986). iainmott@unb.br
Lattes: http://lattes.cnpq.br/5981318071840816
ORCID: https://orcid.org/0000-0001-9452-5999