"Tchaibotsky" Robotic Piano Composer and Performer

By

Matthew Graham

A thesis submitted to
the University of Birmingham
for the degree of
MASTER OF ENGINEERING

Student number: 1568466
Supervisor: Prof. Karl Dearn
Department of Mechanical Engineering
School of Engineering
College of Engineering and Physical Sciences
University of Birmingham
April 2020

ABSTRACT

This paper introduces Tchaibotsky, a player piano with musical compositional abilities. Although previous machine compositional models have been able to somewhat replicate human standard music compositions, none have been able to perform their works in a live setting; failing to capture the timbre and express the true articulation of their pieces. Nor has their work been fully tested. An adapted transformer algorithm trained on 187 hours of live piano MIDI recordings was used to compose music, this consists of 6 levels of trained encoder and decoder modules which follow a sequence to sequence setup. A linear memory scale was implemented, decreasing the parallel computation needed to run the algorithm, making it suitable for running on smaller devices as well as increasing long range motif reference through relative attention mechanisms. These compositions are composed on a Raspberry Pi 3+ through a serial connection to an Arduino Uno control 88 solenoids attached to additive manufactured finger mechanisms with one degree of freedom which each depress a piano key. The velocity and dynamics of the fingers were controlled through an 100Hz pulse width modulated signal; varying solenoid force profiles. This was managed through a series of shift registers connected to the 3 serial peripheral pins, SPI, Arduino pins allowing for 10ms precision control of any 88 outputs. The robot was fully capable of repeating any human sound on the piano. A binomial Turing test was performed with 39 people presented with recordings of human and Tchaibotsky's performances of classical pieces and its generated compositions with the task of determining which was which. 38.4% ($\pm$10.7%) of participants on average guessed correctly in both the compositional domains proving that the music captures human creative features better than expected. Tchaibotsky fared less well when performing, although still fooling some, 66.7%($\pm$14.5%) people guessed correctly with the performances of the human composed piece and 82.1%($\pm$12.4%) guessed correctly. Further research into more exact aptitude of this robot will be conducted as well as development on its physical capabilities.

ACKNOWLEDGMENTS

Thanks to Prof. Karl Dearn for letting me attempt this project, believing in it and supporting me, it has been really fun working on this as part of the course. This has definitely made the year worth while.

Thanks to Dr. Amir Hajiyavand for the brainstorming and reviews. Also thanks for the opportunities he has given me in the last few years.

Andrew Norman, Diego Perez and Lewis Lancaster have had to put up with me all year and have suffered too many ramblings and annoyances. Thanks for doing my shopping, scavenging materials and donating tools for this project. Also for the derby dash games to play in our down time, I could not ask for better quarantine buddies.

Thanks to Darcey Graham for playing the piano pieces composed and creating good benchmarks for comparison.

Table of Contents

1 Introduction

1.1 Preamble

"Writing about Music Is like Dancing about Architecture."

In order to fully appreciate and understand this paper's outputs, given that they are entirely auditory in nature, it is recommended to listen and watch the accompanying files with this paper. This will give needed context to the verbal descriptions of the paper. Furthermore, due to the size of this project a larger word count has been allowed.

1.2 Problem definition

A piano produces its sound through a series of levers that connect the keys to a hammer and string damper. Which when pressed, seen in Figure 1, the damper will be released from the strings and a hammer with strike them at a high velocity allowing them to resonate producing the note. For a successful note depression a white note needs to be lowered by 5.5mm and a black note by 4mm. This note continues until the key is released or, in case of the pedal being depressed, until the pedal is lowered. Both of these methods end the note by re-engaging the damper.

Then entire sound domain produced by this mechanism can therefore be controlled through the velocity and timing of a key press. Combining three attributes: articulation, rubato and the dynamics.

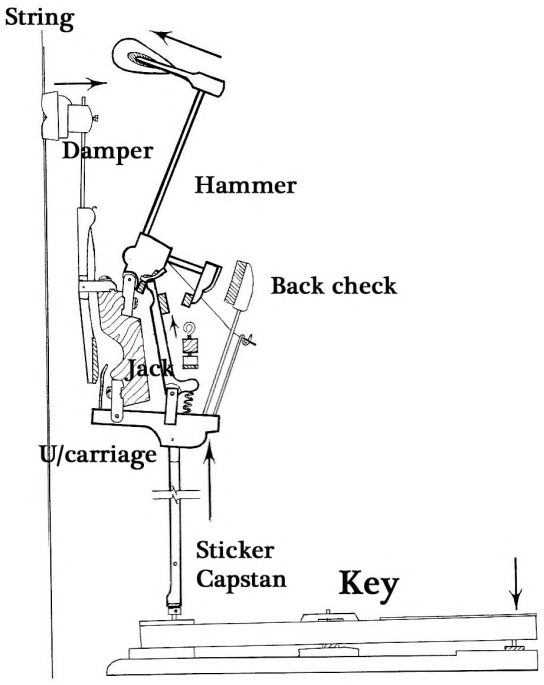

Figure 1 Diagram of working parts of an upright piano during keystroke [1].

- Articulation is defined in this paper as the length of a note relative to its written length. This needs to replicate key presses and pedal presses to a 15ms resolution to sound as technically identical to a human [2].
- The piano's note attack, decay and volume depend solely the hammer velocity as it strikes the string, the force required for this velocity is the note's dynamic. A striking

force of 4-10N is required for a full dynamic range of a standard key-weight piano [3] [4].

- The timing the notes are played at relative to the rhythm is defined as the rubato. This parameter needs to be replicated to 15ms precision.

The differential pitch created by striking different keys and the timbre of the piano describe the full possible soundscape. Written music does not capture the timbre. This can only be replicated authentically using a real piano. Polyphony is required to play piano music with a maximum of 10 notes needed to be played simultaneously. These 88 notes are of fixed pitches tuned to a twelve tone equal temperament in the domain, $f(n)$, described in Equation 1.

$$f(n) = (12\sqrt{2})^{(n-49)}440Hz \quad for \;\; 0 < n <= 88 \tag{1}$$

This paper presents a novel piano playing robot capable of reproducing a human piano performance following the parameters described as well as synthesising compositional AI technology to produce creative new music indefinitely and indistinguishable from a human composer. This makes Tchaibotsky the first machine to be able to reproduce all aspects of musical piano art creatively without any human intervention.

1.3 Literature Review

The idea of piano automata is not new. The first commercial player piano was invented in 1863. These utilised a hand cranked bellow pump which rotated a cardboard barrel, upon which were holes cut at specific positions allowing the air pass through depressing the piano key [5]. These simplistic mechanisms, although impressive for the time, do not have variation in articulation and required skilled operation. Today player piano technology has increased dramatically. The cutting edge of which have full range of recording and playback capabilities such as the Yamaha Disklavier, able to replicate the speed of a key stroke to 10mm/s resolution. These player piano systems require modification of the piano, becoming part of the instrument rather than a robot capable of playing a piano [6].

Robotic pianists have been in development since 1985. The first major project in this is the WABOT 2; a humanoid with cable actuated fingers and hands [7]. Keys further than the surrounding 7 white notes are impossible to play without extensive wait times for arm repositioning of the hand. Furthermore, it cannot reproduce dynamics due to its binary key depression. These robots have improved over time, a more recent robot made at Taiwan university has two hands articulated on a linear rail achieving 5m/s operation along the entire piano. [8] The hand cannot move during a note press limiting the speed notes outside the hand vicinity can reach. Although the hand can internally change it's finger positions to play black keys, the limits on the degrees of freedom the fingers reduces the note combinations

that can be played. Cambridge University has developed a soft skeletal piano playing hand. The anisotropic material structure enables the hand to perform in varying styles over a high dynamic range entirely passively [9]. Although the material technology is advanced the end effector has no actuation, limiting the articulation and note control.

Further research is needed to fully emulate the complexity of the human hand. A simpler control strategy is required to fit requirements of the project in the time frame. One of the most advanced robot pianists currently is the Teotronico. It relies on an array of 53 static electromagnets, one for each key limiting it's pitch range to 60% of the piano. The power supply the bottleneck for the number of simultaneous notes played, requiring 150A for all 53. The articulation is controlled with variable voltage input to the magnets. This enables 126 different velocities of key strike [10]. None of these robots have the capacity to perfectly replicate the human player nor possess any composition capabilities.

The foundation of musical harmony is based on the superposition of harmonic frequencies, with the western scale consisting of logarithmic divisions of harmonic frequencies split into 12 parts. Pythagoras was the first to argue that musical harmony was purely mathematical, correlating the harmoniousness of musical intervals to the simpleness of their numerical ratio [11]. Bach extended the ideas of musical mathematics to harmony in the form of counterpoint producing almost formulaic, yet interesting music. Schönburg argues that musical melody is an extension of this idea coincident with a balance of repetition and variation.[12] Modern mathematicians have developed these ideas into abstract mathematical concepts describing the entire musical domain through set theory, topos theory, transformational theory and abstract algebra [13] [14]. Because music can be described totally and analysed precisely form a mathematical perspective it is hypothesised that music can be entirely produced through these methods to the same standard as a human. The contextualisation and interpretation of musical ideas to tangible objects and events cannot be replicated organically through a music focused algorithm and present a theoretical shortcoming to the potential engagement with pure algorithmic music.

The earliest forms of automatic composition follow similar process to conventional computational design, an example this being Mozart's dice game. This creates waltzes from a database of short motif segments selected based off the value of dice [15]. This has been implemented commercially in modern computing compositions, where a vast database of motifs and chord progressions are accessed semi-randomly to create a piece [16]. Although this will create a unique piece it is not a creative endeavour with no new musical ideas. A different approach is needed that understands underlying musical styles in order to produce creative new music.

More powerful algorithms involve more complex Markov models. These are probabilistic models that alter their nodal design which enables a higher chance of producing similar patterns and musical features found in the data they learn from; whilst not mimicking it. These are capable of understanding immediate contextual patterns allowing musical phrasing, style and rhythm to be equal to that of humans [17] [18] [19]. These algorithms cannot produce macro scale features such as a strong melodic voice or overall compositional

structure, because they lack memory due to non linear manipulation of the previous musical notes.

Bachbot can produce short polyphonic four-part harmony in the style of Bach to a standard indistinguishable from the real work. This utilises a long short-term memory neural network, LSTM. It has neurons adapting themselves to manage specific fields of music theory without prior knowledge or supervision and exclusively uses linear manipulations in its input sequence allowing longer features to be incorporated. This enables melodic expression and full harmonic progression; however, this is still limited and requires manual assistance to composition[20]. Furthermore, it only produces notes, ignoring any rubato, articulation or dynamic contrast.

Recent developments in this field from OpenAI and Google resulted in Musenet and Magenta [21] [22]. They use a transformer neural network. They are trained off live performances and from this can reproduce expressive playing with long term structures. The models are made of 4.5 billion neurons and are too large to be ran on standard equipment. There have been no studies to recreate these types of models on a robotic platform, nor detailed analysis of human interaction with them.

There is a growing market for these devices to aid with composition and performance beyond the novelty of a robotic musician. Programs such as Jukedeck[23] can automate the creation of background music, a 1.3 billion dollar industry [2]. These could also be used as composition aids being able to automatically generate and play accompaniments to human composed melody lines, reducing work load, simplifying composition and acting as an educational tool. Furthermore, this could be developed for wider accessibility of high quality copywrite free unique music to be played at venues such as hotel lobbies or cruise ships.

2 Method

2.1 Electronics

Figure 2 shows the circuit schematic of the piano. A Raspberry Pi 3+ executes the machine composition algorithm and outputs the velocity, command type and time of each generated musical instrument digital interface, MIDI, command in a comma separated variable, CSV, format. This data is fed via the serial input into an Arduino Uno where it converts the output into a pulse width modulated, PWM, 88 bit output. Appendix 1 explains in detail this serial communication. This data is shifted into the memory of 11 daisy-chained shift registers which activate the solenoids needed to play the required keys powered by a 12V 30A power supply.

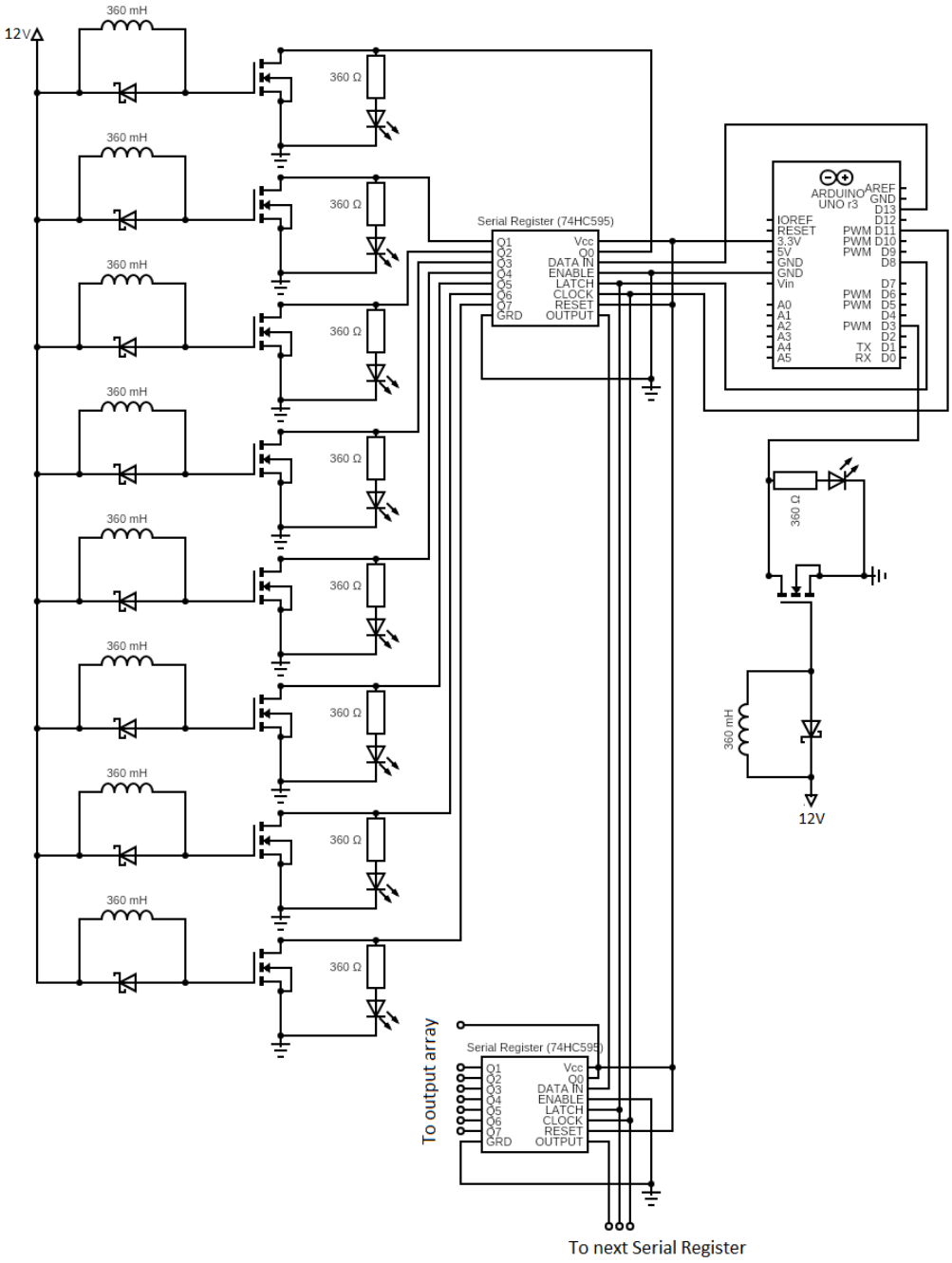

Figure 2 Circuit diagram of the player piano. Each inductor is a solenoid that represents a finger. The right most inductor is the pedal actuator. The repeated daisy chained shift registers and serial input to the Raspberry Pi are omitted for conciseness

2.1.1 Solenoids

Solenoids were chosen as the actuation mechanism due to their relatively quick binary stroke engagement and retraction (<0.3s) compared to other actuation systems. This enables fast repeated notes to be played required for conditions such as a trill. A PWM signal varies the time for over a duty cycle the circuit power is present. This gives the outcome of differential average voltage. This changes the average force of the magnetic field and therefore the acceleration of the steel armature when it strikes the key; changing its dynamic.

The solenoids were sized to depress the key optimally. Its minimum playable voltage must induce a force that can overcome the key-touch depression mass, m_k (50g), and finger mechanism friction, F_f, whilst at a stroke of 5.5mm and produce a striking force of 4N on the key. When playing with its maximum playable voltage this final force must be 10N on the key. The difference between the minimum and maximum voltages must be greater than a 50% duty cycle so that the 4-bit MIDI velocity range can be utilized within the 8-bit PWM signal.

The software FEMM v4.2 was used to perform a finite element analysis of the magnetic field to determine whether the solenoid would meet this specification as well as calculate the time it takes to depress the plunger at every duty cycle. The solenoid was modelled in a 2D axisymmetric homogeneous semicircular domain of a 100mm radius with the steel plunger and copper coil drawn to the dimensions and properties each solenoid. this set up as shown in Figure 3. A dirichlet boundary condition was used. A triangular mesh with a global size of 7mm, a growth rate ratio of 0.2 and a 0.4mm wall mesh. A mesh density convergence study was performed finding a change of <0.05N for finer meshes than these parameters, this mesh was therefore seen as converged.

The solenoid field strength was calculated from the number of coil turns, N, current draw, I, material permeability, μ, and the length, l. These values were found on the store solenoid data sheets. This was estimated numerically through Equation 2, and calculated discretely in FEMM 4.2 to calculate the

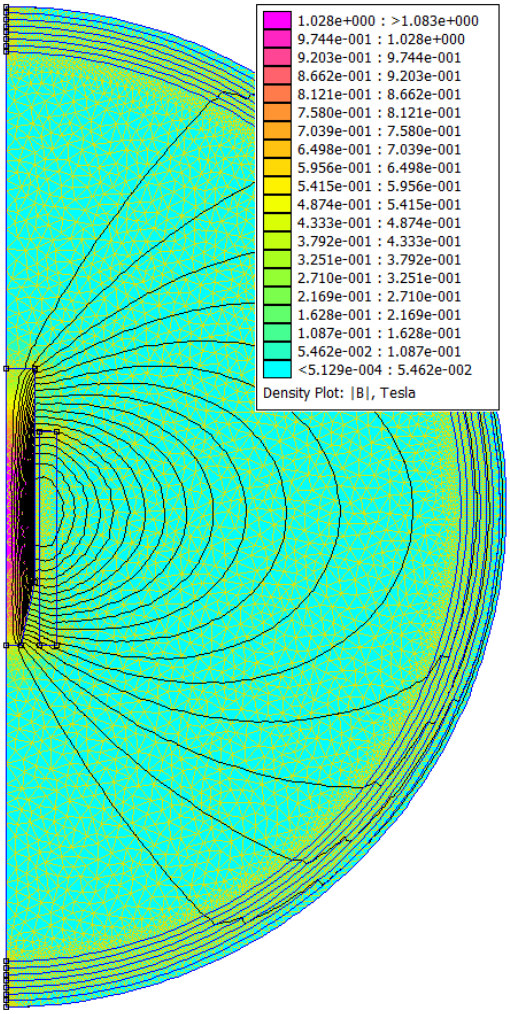

Figure 3 FEMM contour plot of the solenoid magnetic flux density, this set up was repeated to calculate the acceleration at every stroke position

required duty cycle to play at the right dynamic.

$$B = \frac{\mu N I}{l} \tag{2}$$

This output is not practical as it assumes a constant current and the solenoids will be driven through voltage. Solenoids are inductors and thus contain a reactive component of impedance causing the frequency of the duty cycle to become a factor in their output voltage. During a steady state DC current can this phenomena can be incorporated into the Ohm's law to Equation 3 to solve for voltage, V, where f is the frequency and L is the inductance.

$$V = 2\pi f L I \tag{3}$$

The magnetic force integral, F_s, over the plunger's length was calculated through a series of simulations in FEMM 4.2 for a plunger stroke extension from 0mm to 5.5mm with a 1mm resolution and a voltage duty cycle from 100% to 50% with a 5% resolution. This data was used to ensure minimum force requirements are met and to calculate the time delay in note activation from the transient motion of the plunger. The acceleration of the plunger was calculated through Equation 4 with m_p, being the mass of the plunger (42g).

$$a = \frac{F_s - F_f}{m_p + m_k} \tag{4}$$

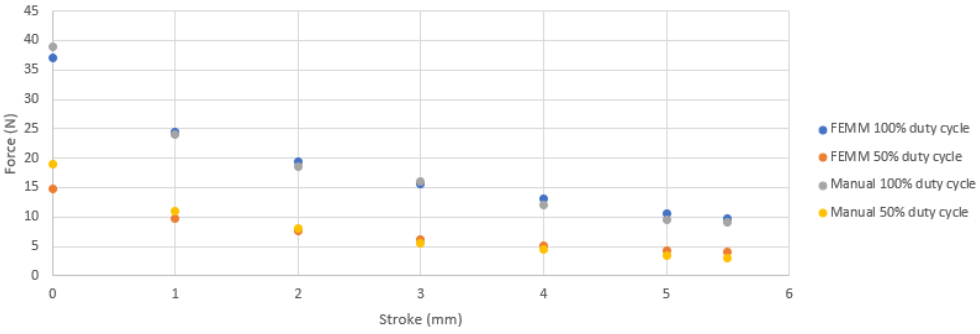

Figure 4 Force profiles of the solenoid at each stroke modelled in FEMM and from validation experiments for its maximum and minimum sufficient duty cycles.

The chosen solenoid was the Suzhou UE-3260T. Figure 4 shows the force of the solenoid plunger as it changes stroke at a 4-bit spread duty cycle. These match the literature's predicted optimum forces with an error of 0.23N. This deviation from the optimum is within the first standard deviation of human force spread found in the literature [3], this is therefore a satisfactory force profile that will give Tchaibotsky's music a specific style.

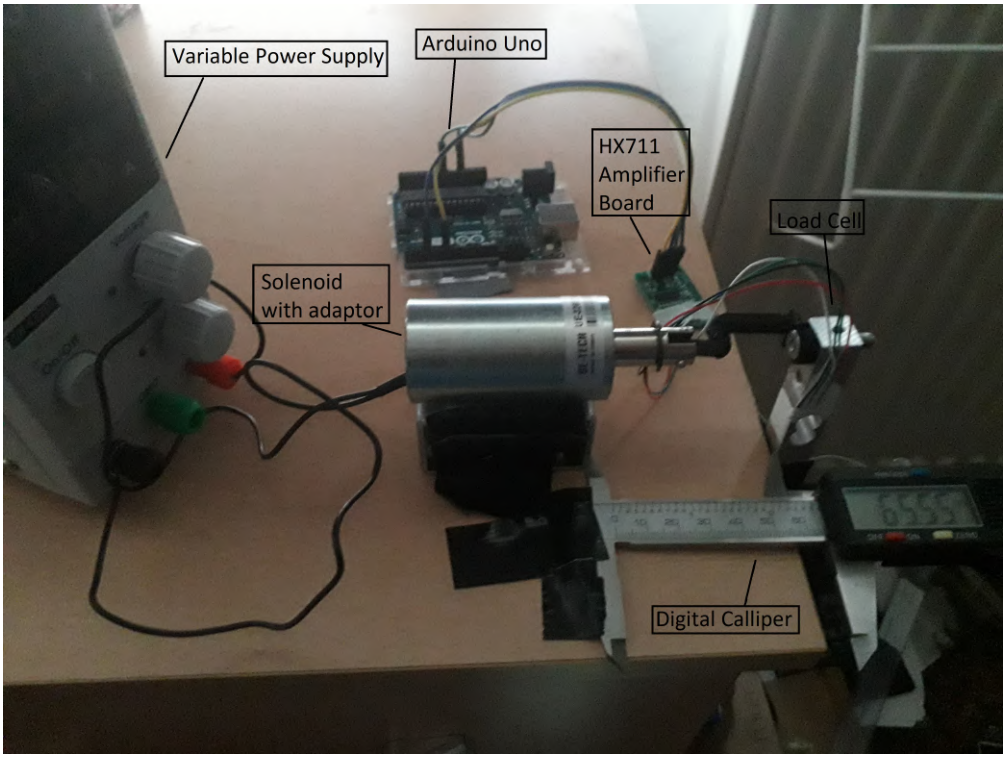

Figure 5 Experimental setup of the solenoid force verification test.

A second experiment using a load cell Figure 5 was undertaken to verify the forces predicted using the FEMM, the comparison of this result is incorporated in Figure 4. The same voltages and stroke distances were calculated as in the FEMM and were recorded as the average load over 3 separate experiments with different solenoids. The FEMM showed an average error of 0.26N with a maximum error of 2N occurring at the full depression of the stroke. This error could be caused by the elastic deformation of the 3D printed adaptor, measurement error in the distances and the deflection of the load cell base. This still validates the simulation for the needed precision of the piano timing delay and confirmation of the solenoid's acceptable force profile. This also confirmed the minimum duty cycle the solenoids could reliably depress the piano key. This was found to be at a 45% duty cycle equivalent to being driven at 5.4V.

Rubber dampers were added to the base of the solenoid plunger reducing the noise of movement to a negligible amount. Once the solenoid strikes the key, the PWM signal of that output is reduced to 50% output. This reduces the load on the power supply and, coupled with time offsetting of the PWM, in an ideal situation allows for a theoretical 120 solenoids to be completely depressed simultaneously. This control scheme is an improvement on prior art player pianos for polyphony capacity [6] [10].

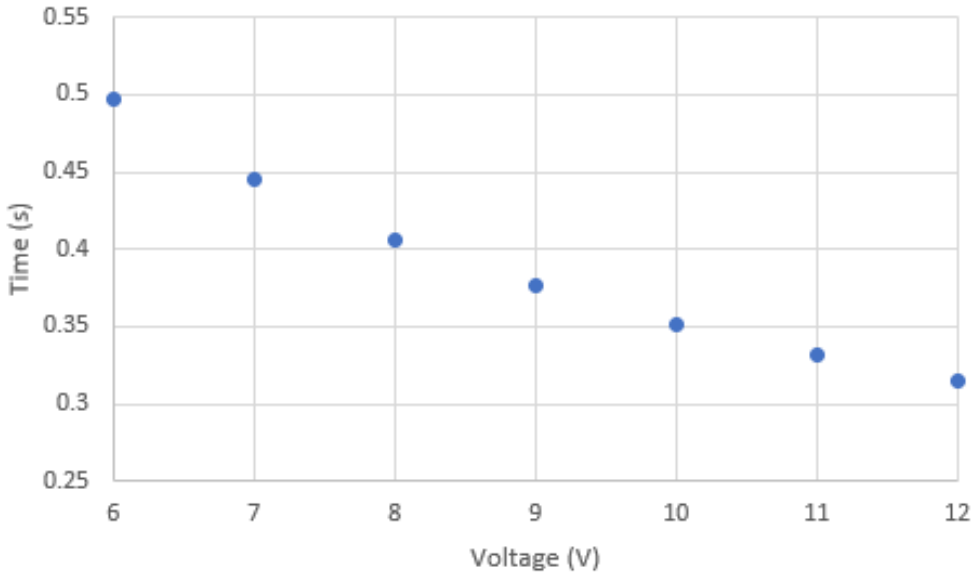

Figure 6 Time delay of a complete plunger stroke in the full duty cycle range of the solenoid

Using Figure 4's forces and Equation 4 the total time delay can be ascertained for every voltage of the solenoid through trivial Newtonian mechanics. Figure 6 shows this delay and is used as a lookup table to offset the time every note command is sent to the solenoids to ensure they are all played nominally at the same expected time.

2.1.2 Inductance Isolation

The solenoids require isolation from the microcontrollers to prevent damage to them from electromotive force spikes and the 12V operating voltage. This isolation was achieved using an IRLB8721 n-channel MOSFET. This has a safety factor of 6.6 with respect to the drain current from the solenoid. Its gate saturation region is maximum at 3V; thus the 5V output logic of the shift register would fully open the gate with its minimum channel resistance, 0.1Ω, R [24]. Under a full 1A load, I, through Equation 5, where m is the mass (5g) and c is specific heat (0.9), the temperature increase per second, ΔT, is calculated to be 11.1^0C. Equation 6 [25] is used to calculate the normal convective cooling energy rate from the initial ambient temperature, Q, accounting for the surface area of the MOSFET, A ($35mm^2$), and the ambient convective heat transfer rate of aluminium, h_c ($59W/m^2$)[26]. The cooling energy rate was calculated to be 0.18J initially, using Equation 7 the initial cooling rate can be calculated to be 40^0C/s allowing indefinite solenoid power use without overheating

$$\Delta T = \frac{I^2 R}{mc} \quad (5) \qquad Q = h_c A \Delta T \quad (6) \qquad \Delta T = \frac{Q}{mc} \quad (7)$$

A flyback diode was used to dampen the voltage spikes caused by the parasitic inductance from the solenoid magnetic field breakdown when the circuit is opened. A schottkey diode was used for its fast switching speed. When the MOSFET gate is open the inductor dissipates the energy stored in its electromagnetic field. Without a diode the solenoid generates electromotive force, emf, from this stored energy maintaining the amperage until it is completely dissipated. Due to the lack of power source this equilibrium can only be maintained through a voltage spike. This spike can be seen in the right image in Figure 7, this could instantaneously over-voltage the MOSFET; permanently damaging it. The diode, positioned in parallel with the solenoid, creates a new closed circuit with the solenoid when the MOSFET opens. The emf current flows through this circuit and decaying over a longer time span, resulting in the reduced voltage spike seen in the left image in Figure 7, consistently under the 30V breakdown voltage. The peak forward current of the diode has a safety factor of 2 [27]. Comparing the MOSFET activation timings from the oscilloscope readings with the timings of the Arduino power signal it can be determined that the addition of the diode reduces the magnetic force breakdown speed by a factor of ten. This reduce the maximum PWM frequency of the solenoid to approximately 600Hz, this disadvantage is of no consequence as the shift registers bottleneck the frequency at 120Hz.

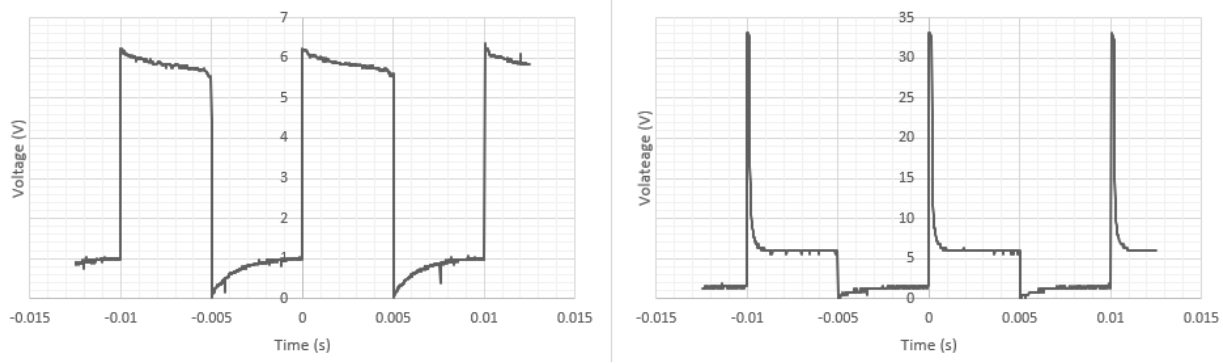

Figure 7 TDS 1002B Oscilloscope data from a 6V drain of a IRLB8721 N-channel MOSFET with an 100Hz 5V square wave signal. On the left is with a flyback diode and the right is without one. Note the voltage spike is potentially higher than shown as it's instantaneous peak is beyond the time resolution of the oscilloscope

A light emitting diode, LED, was used in conjunction with a 360Ω resistor in parallel with the MOSFET gate and the shift register output. This not only gives a visual aid in error handling the circuit, but also acts as a pull-down resistor. If the output from the serial register ever becomes high-impedance or held in reset, the gate voltage is pulled low and the MOSFET gate will open preventing the solenoid from parasitically draw power.

2.1.3 Output Management

11 8-bit 74HC595 shift registers were daisy chained together, providing control of the 88 solenoids from 3 Arduino output pins. Data is transmitted in 88-bit packets via synchronous serial communication. At every leading edge of a clock signal the shift register reads the state of the data pin. This state, along with the previous data states, is shifted as a 1 or 0 binary number through the consecutive internal memory registers; made of flip flop gates. The daisy chain connects the output data stream of the previous shift register to the input of the next one, as well as synchronising the latch and clock. This incrementally shifts a full 88 bit data packet, representing the current notes to be played, over 88 shift register clocks. After these 88 clock cycles the latch is set to high releasing the current data packet activating the solenoids. This process is performed at 100Hz.

A PWM signal is used to vary the velocity of the solenoids. This is limited by the processing speed of the Arduino. To maximise this speed the dedicated serial peripheral interface, SPI, port pins were used. This increases the speed by 2.5 times. The code reacts to an internal timer interrupt and from there takes 5 clock cycles to execute each output pin; ten times faster than using `digitalWrite` commands and is non blocking. Following Equation 8, with f being the PWM frequency, o_n being the output number, and c being the Arduino clock speed, the ratio of time between code execution and idle time can be calculated. The cpu load ratio was maintained under 0.8 at a 100Hz PWM signal, maintaining a rubato resolution of under 15ms and enough idle cpu time to detect Raspberry Pi serial input. Although this is in an audible frequency, the solenoids produce unnoticeable sound as evidenced by the recordings. Each consecutive solenoid activation was offset by a clock step to maintain a more constant and lower power draw.

$$cpu = \frac{1}{fo_nc} \tag{8}$$

The pedal output is handled on a separate pin from the Arduino, the output is identical in all ways to the note outputs, however it does not require a PWM signal. This simplification is can be made due to the binary nature of the pedal depression.

2.2 Hardware

2.2.1 Finger Design

Each piano key has its own finger mechanism. This simplified the design to one degree of rotational motion for each part. This is achieved through 88 lever mechanisms controlled by solenoids, seen in Figure 8. Due to the width of the solenoids being greater than the average key width, 12.3mm, the solenoids were staggered onto two trays: one for black keys and

one for white, requiring inextensible couplings of lengths, l_o, 110mm and 65mm to connect the solenoid plunger to the levers. The finger levers were designed with two versions, black and white keys. These black keys were longer reaching a lower platform on the stand, this maintains the same lever ratio of the white keys.

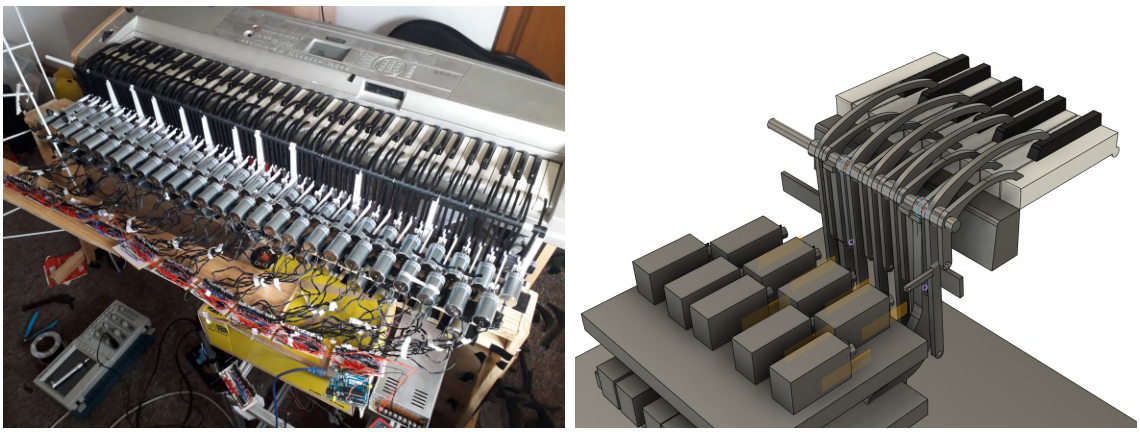

Figure 8 On the left is final image of the robot and the right is an octave drawn in CAD in Fusion 360.

The finger design was optimised to undertake a maximum solenoid force, F_s, of 10N, with a deformation, δ, under 0.3mm. The print time and cost was also minimised. A 5.5mm and 4mm arc travel was required to depress fully the white and black keys respectively. There is an exponential decrease in the solenoid force with stroke length. This creates a trade off between the mechanical advantage of the lever with respect to the solenoid force required to depress the key. After experimentation with varying lever ratios it was determined for the selected solenoid a ratio of 1:1 was optimum.

The designs were developed iteratively in Fusion 360. Due to Covid 19, only fused deposition additive manufacturing, FDM, processes were available for quick prototyping and production of the 88 similar parts. Polylactic acid, PLA, was used as it is the stiffest commonly available FDM material, with a Young modulus, E, of 3.86GPa [28]. A finite element analysis, FEA, of the black and white finger lever bodies were executed in the Fusion 360 toolkit. They were modelled as hollow 3D sections with homogeneous material properties. This simplification ignores the weaker axis caused by the fused plastic in the printing z-axis. There was no lateral force component allowing for this simplification. The internal geometry from infill patterns is too complex to be modelled and solved for each iteration. These were not modelled in order to create a worst case scenario.

Each design was iterated through the wall thicknesses at 0.75mm intervals (the PLA filament diameter), finger depth and width at 1mm intervals. These were iterated until there was deflection lower than 0.3mm. Low deflection is critical to reduce solenoid stroke and its required force output. A dynamic explicit model was used as the strain rate was above 0.01s. A mesh dependency study was performed revealing $<$0.01N changes for any tetrahedral mesh of a global size under 0.5mm, a growth factor ratio of 0.2 and a wall size of

0.1mm. A lateral fixture was added to the pivot and a static fixture was added to the finger tip. A 10N force was applied to the eye of the finger, where the solenoid will be pulling from. Figure 9 shows the final results of both fingers. A topologically optimised solution was not used as it increased the printing time due to the increased number of walls. The solution reached a maximum deflection of 0.26mm and 0.29mm respectively, with a von Mises safety factor of 12.4 and 11.3 respectively.

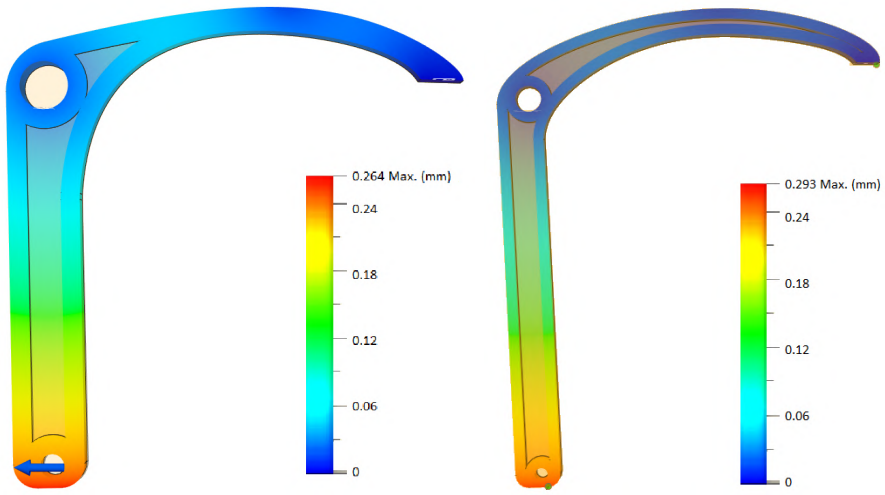

Figure 9 Cross section of the final design FEA for the white (left) and black (right) fingers

A wall thickness of 3mm was used with a gird infill of 10density. This was printed with a 3mm brim for bed adhesion on a modified Ender 3 printer with a heated carbon infused glass print bed at 60°C, a nozzle temperature of 196°C and a print speed of 70mm/s. The only difference geometrically between the two finger designs were the thickness. The white fingers were 5mm deep and the black 8mm with print times being 26 and 48 minutes respectively. The top and bottom walls were 3 layers thick with a layer height of 0.1mm. The fingers were unique for each note in the octave to compensate for the varying axial distance of each key. These differences were only in the heights of the spacer section in the finger and do not effect the deflection.

The key-finger interface was finished with a silicon tip for smooth silent depression of the keys. A rubber band was added on the lower lever arm connecting it to the back of the stand achieving a tension of around 5N. This decreases the retraction time of key by 0.08s.

$$h_c = \frac{F_s l_o}{\delta E w_c} \tag{9}$$

The coupling between the finger and the solenoid was also 3D printed, the only optimizable dimension in the design was the cross sectional area to ensure deflection was under 0.1mm. The width, w_c, of this section was limited by the 4mm wide plunger recess that can be seen in Figure 10. The minimum height, h_c, through Equation 9 was 0.014mm the minimum for the longer, worse case, coupling and therefore was not a limiting factor for design. The final height of the coupling was 5mm, to allow room for a 3mm steel dowel pin fixing it to the solenoid plunger and had a hook mechanism to be secured through the finger.

Figure 10 Close up view of the two coupling pieces from the top row of the solenoid tray.

2.2.2 Stand

The stand was constructed using recycled wood and manufactured by hand. The deviation in height between each side of the solenoid stacks is <0.5mm. A simply supported beam deflection calculation shown visually in Figure 11 was performed through Equation 10 [29] to ensure each shelf could hold the 260g solenoids without bowing. The top shelf holds 52 solenoids and the bottom 36. This weight was assumed to be spread homogeneously over a 2d plane of the shelf, only the top shelf was considered as it is the worst case scenario.

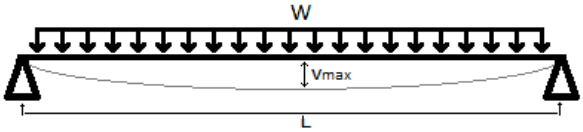

Figure 11 Free body diagram of the top stand shelf.

$$v_{\max} = \frac{5wL^4}{384EI} \tag{10}$$

Where L is the beam length (1.2m), w is the distributed force on the beam (132.6N), E is the elastic modulus along the grain of ash (11GPa)[26] and I is the moment of inertia (2.6x10$^-$5kgm^2). This gave a maximum deflection of 0.2mm.

The lower stack also carries 16 PLA 3D printed support struts where the finger array is mounted on. They are positioned at every neighbouring white-on-white key and also act as axial alignment for the fingers. A back wall is attached to these, limiting the finger extension to a resting position on the keys. An overhang seen in Figure 12 on the lower face of this strut which was used for location with the edge of the stand ensuring the finger array ran parallel with it. This was secured to the stand with Methacrylate epoxy. These are required to remain stable under the load from the weight of the fingers and bushing rod as well as, in a worst case scenario, the 10 surrounding fingers being depressed at full volume. An FEA was set up with the same mesh and solver parameters as the fingers shown in Figure 12. It had a load of -50N in the x-axis and a 16N force downwards accounting for each struts share of this maximum force. The design was iterated until it was under a 0.5mm deflection threshold, achieving a maximum deflection of 0.465mm. It was printed with the same parameters as the fingers other than a difference in wall thickness of 4 layers. The print time was 64 minutes.

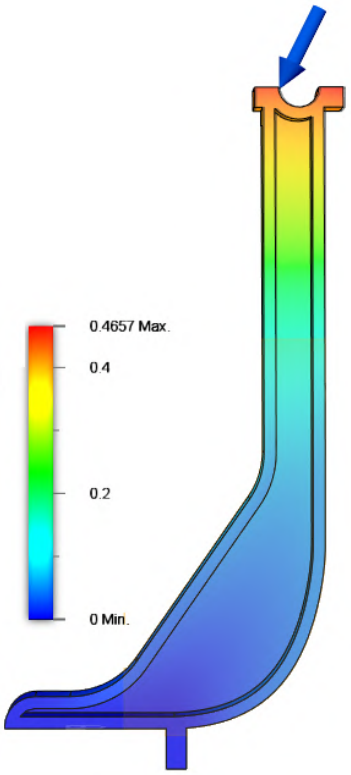

Figure 12 FEA of the strut holding the finger array performed on Fusion 360.

A compliant 3D printed clipped lid, Figure 13, is used to create an ISO H7 n6 interference fit with a 10mm PTFE bushing rod and the strut. This rod is where the fingers are mounted to providing a low friction mount, $\mu = 0.1$ [30]. The fingers are positioned axially to their correct key to a precision of the z-axis on the 3D printer, $\pm$ 0.05mm from being stacked on the shaft. Local error in this axis is not critical due to the relatively large key size and the lack of tolerance build up from the 0.2mm stand clearance. The longitudinal error was calibrated with feet shims until all the fingers rested on the keys. Due to 3D printing concentricity error, the 10mm mounting holes in the fingers featured slight deformity, this required hand finishing and therefore the friction on each finger was marginally different. Because of these unknown deformities it was impossible to accurately calculate the frictional force and thus its effects were nullified through manual calibration of the solenoid's PWM. The electronics are mounted beneath on the lower board for ease of access and assembly.

Figure 13 FEA of the strut holding the finger array performed on Fusion 360.

2.2.3 Pedal

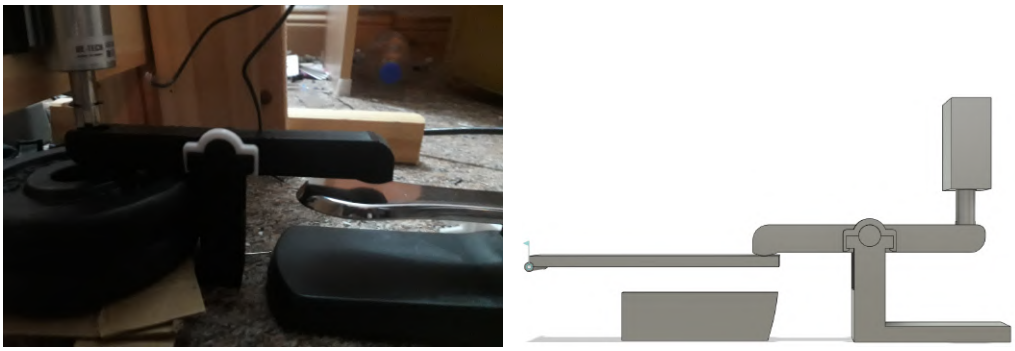

Figure 14 left, the CAD drawing of the pedal mechanism, right, the manufactured version

The pedal required only binary input from the solenoids and thus was on a separate circuit to the rest of the array. The pedal mechanism contains a spring that requires a 30N load to overcome. The design can be seen in Figure 14. The solenoids are large enough to be able to depress the 4mm needed to activate the pedal with a 1.1:1 ratio lever. An extended base of the lever mount was used to weigh and secure the mechanism to the ground. This also acts as a extension limit keeping the lever in contact with the pedal. The lever was couple with a cable tie to the solenoid which was mounted on the back board of the stand. A similar FEA iteration to the finger mechanism was performed, the results are shown in Figure 15. This model had a 30N load on the solenoid coupling, a fixed constraint on the pad and a rotational constraint on the axel. There was 0.37mm of deflection and a von-Mises minimum stress safety factor of 5.3.

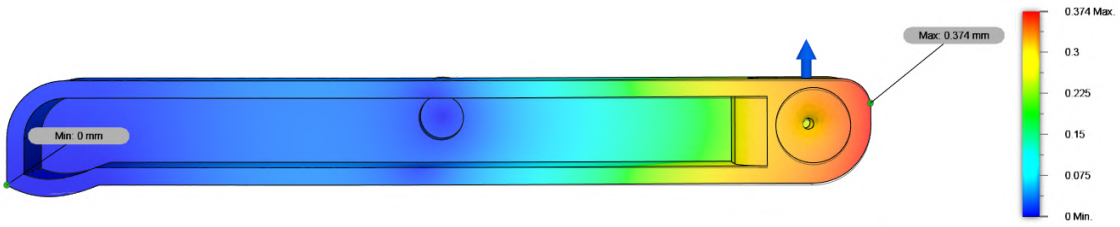

Figure 15 FEA of the pedal shaft.

2.3 Software

2.3.1 Background

Western classical music is built on harmonic rules and patterns, like all art, these are guidelines which can and should be broken and manipulated. The optimal solution for developing

new music must generate solutions that can violate constraints in order to optimise others, this type of relaxation scheme cannot be done using traditional programming and thus a machine learning approach is used [31]. For an output to be musical it also needs to reference and develop upon long and short term past motifs and patterns. This sort of problem is well suited for a transformer algorithm as it employs a self attention model that is able to reference specific relevant data from any section of the sequence at every generation step.

There is no objective optimal solution for music composition, this does not mean that it cannot be assessed to create pleasant music [21]. Similarly how great composers take inspiration and musical theory knowledge from previous composers, the algorithm will be trained off a dataset containing vast arrays of classical piano music and assessed on how close it predicts the next note to that of the composer. Over time with enough unique pieces this should train it on the average predictions not replication understanding the musical principles underlining the pieces.

2.3.2 Data

MIDI files have been shown to have the potential to encompass and replicate all the relevant data of a piano performance. It is important to note the distinction between MIDI score and MIDI performance file. A MIDI score represents sheet music precisely as it is written ignoring rubato, articulation and dynamics, this fails to capture the expression of music. MIDI performances are from human created recordings and thus are all encompassing in expression with the limiting factor being only the resolution of the recording.[32]

The MAESTRO data set was used. This data set contains 172.3 hours of virtuoso piano performances recorded on Yamaha CFX concert grand pianos with Disklavier Pro recording attachments in MIDI format. These were recorded over 10 years from the International Piano-e-Competition [33]. The note velocity is recorded with a resolution of 0-1023 at a sampling rate of 300Hz and includes pedal output [6]. All of these are beyond the resolution of MIDI input so is beyond the required quality. This is a useful dataset to use as the state of the art compositional algorithms, Musenet and Magenta, were trained on it [22] [21] allowing for more accurate comparisons between models. The dataset was increased in size by a factor of 12 by modulating each note in every MIDI file by every note in the scale. This created a version of each piece in every key signature. This not only increases the size of the training set but also evenly distributes the number of pieces in each key, reducing bias in the algorithm, a potential improvement on the data preparation from these prior models.

The MIDI sequence is broken down into a 1-dimensional array. Every 3 characters in the array represent one MIDI command built from the following in the order time, command, velocity:

- 89 Note on commands, for every key on the piano representing the start of a key press, this also includes the sustain pedal acting as down when its value is >=64.

- 89 Note off commands, representing the release of a key or the sustain pedal; when it is accompanied by a velocity value <=64.
- 125 time-shifts, represent each discrete change in the time domain between events ranging in increments from 8ms to 1s.
- 32 velocity commands, which change the velocity of all notes until the next velocity command.

The MIDI file's time domain representation was rounded to fit the time shift representation, and is now accurate to ±4ms making it an unnoticeable approximation. This dynamic time step allows for second long pauses and fast polyphonic sections of the music to be represented in one event. Rests longer than one second are represented by concatenated time shift events.

The MIDI velocity input only contains 127 discrete values, much less than the Disklavier resolution. For the sake of computational power this was reduced to a vocabulary of only 32 values, a quarter the resolution. As explored in Oore et al. [32] this quantization still captures the dynamic contrast expressed in the music and is a necessary trade off for dimensional minimisation of the input vector X.

This method compresses the music file dramatically. This gives the model a 335 word vocabulary and a short sequence distance between distant data points increasing the potential to understand long term sequences.

2.4 Model

The model used was a modified transformer algorithm built on Google's machine learning toolkit's: Tensorflow 1.15.2 and Magenta 1.2.2. An embedded input of a MIDI sequence up to the current note, $X = (x_0, ..., x_n)$ and then maps this sequence to a continuous representation in an encoder, vector $Z = (z_0, ..., z_n)$. This map is then used by the decoder to create an output $Y = (y_0, ..., y_n)$ to create the next note.

2.4.1 Positional Encoder

The sequence X is simultaneously inputted as a 335-dimensional one-hot encoded vector, x_n into the transformer and there is no convolution, with no notion of order in the input. In order to calculate relative position a positional encoding matrix is utilised at the input of the first encoder block created through Equation 11.

$$P_e(i) = \begin{cases} P_e(p, 2i) = \sin \frac{p}{10000^{\frac{2i}{d}}} \\ P_e(p, 2i+1) = \cos \frac{p}{10000^{\frac{2i}{d}}} \end{cases} \tag{11}$$

Where p is the time position, d is the total dimension size (335) and i is the current iterated dimension. Note that each dimension represents a different command such as a note on or off. This creates a matrix of positional sinusoids, R_e, that define the relative position of every feature in the time domain. Because of the infinite repeatable sequence of the sinusoids, these positions allow the model to extrapolate sequence lengths longer than it will have been trained on.

2.4.2 Encoder

This was composed of a stack of 6 identical encoder blocks connected via normalised residual connections, this is expressed as an overview in Equation 12. All of these layers have the same global input dimensions. Each encoder contains two sublayers, a multi-head self-attention layer and a feed-forward network, (FFN).

$$Z_{layer} = normalisedlayer(x + encoder(x)) \tag{12}$$

The self-attention layer enables the model to look at relevant data points in X and use this to encode with the current data point. This method is how the transformer incorporates time dependant contextual understanding into the data. This is done through the use of three vectors, a Query (Q), a Key (K) and a Value (V). These are abstractions of different attributes of attention and are multiplied by the input embedded vector, X, to create the weight matrices X_Q, W_K and W_V.

To calculate the output of x_1's self attention through Equation **??**, the softmax scores express how relevant each data point is to that position. This is done by taking multiple dot products of the respective data point's Q that is being scored and the K of each other data point. This score is then divided by the square root of K's dimension, this increases the stability of the model. A softmax function normalises this between 0 and 1. Each data point's V is multiplied by this score, this aims to just minimise irrelevant data points by multiplying them by very small numbers. These are summed to create an output vector, Z, for the initial data point. This process is continued for all data points.

$$Z_i = \sum(softmax(\frac{QK^t}{\sqrt{d}}V) \tag{13}$$

The transformer incorporates multi-headed attention of 8 sets of Q, K and V weight matrices. These expand the focus laterally, enabling more positions to be simultaneously and attention given to more abstract concepts. The weights on these matrices are trained and optimised to be specific for these jobs.

The feed-forward layer abstracts the data allowing for more features to be analysed. It is a three layered FFN connected with linear transformations, b, with a rectified linear unit

activation function at the nodes, W. The hidden central layer has a dimension of 1340, where the input and outputs have dimensions of 335. The transformation for each layer, n, is expressed in Equation 14.

$$FFN(n) = max(0, X_{n-1}W_{n-1} + b_{n-1})W_n + b_n \tag{14}$$

2.4.3 Decoder

The decoder's constituent parts are almost identical to the encoder and is made of the same sizes too. The only difference is the addition of an Encoder-Decoder attention layer. The input of this layer is a K and V vector. The weights of these vectors are developed through training that is formed from its multiplication with the final output of the encoder tower and thus represent, in a trained model, a highly sophisticated key to attaining relevant data points. The other inputs to the decoder are the output of the previous self attention layer and that layer's Q.

The previous outputs of the decoder tower are positionally embedded by calculating the closest matching phase of the sinusoids in the matrix P_e. This is fed into the input of the first decoder, similar to the input in the encoder tower. This cycle enables the decoder to produce an ordered output sequence until it produces a stop code. This therefore can enable it to self reference itself as well as produce sequences of music longer than it trains on, and in theory be just as musical.

In order to stop the decoder from using the future and present data values of the piece, the upper triangular section of the P_e is set to 0. This is the domain of all the relative future data and therefore should be masked.

The attention matrix was converted into 2D for all computation, this reduces the memory usage from an exponential to linear scale. This enables the algorithm to be ran on the Raspberry Pi 3+.

2.5 Execution

The Model was trained on BlueBEAR (a IBM Power 9) for 9 hours for 10000 parallelized training steps. An Adam optimizer was used with a varied learning rate following the Equation 15 where s_n is the current step, s_i is the initial step. The dropout rate, d, (the base ratio of neurons to be tweaked) was set to 0.1 for all layers of the neural networks. This variable learn rate ensures more volatile changes to the neural network at the start of the training and as it converges on a solution increases the plasticity of the neurons, lowering the modification frequency. This ensures important converged neurons are not modified too drastically decreasing the chances of divergence.

$$l_{\text{rate}} = d^{-0.5} min(s_n^{-0.5}, s_n s_i^{-1.5}) \tag{15}$$

The training was assessed to minimise the output entropy. The output of the decoder is a normalised 335x1 matrix, y_o. y_o contains the probability of every predicted output. This output is compared with the true output from the MIDI file, y_n, in a one-hot encoded form. Following Equation 16 the entropy, e, is defined as a measure of the model's certainty and accounts for all the outputs that were potentially incorrect.

$$e = \sqrt{y_n^2 - y_o^2} \tag{16}$$

This is then back propagated through the algorithm and the neuron weights are updated stochastically, emphasising the neurons responsible for the highest error. This produces gradient decent reducing the average entropy in the model. This is not a full measure of music aptitude and if the entropy is too low it suggests the model is over-fitted to the data-set. To reduce this it is trained and tested on an 80:20 split of the data. This means the entropy tests of the algorithm are performed on unseen data which would be impossible to replicate.

Model	Entropy
Performance RNN	2.495
LSTM	2.440
MuseNet	1.835
Magenta	1.840
Tchaibotsky	1.864

Table 1 Validation of other algorithms using the Piano-e-Competition dataset

Table 1 shows the previously attempted algorithms (Performance RNN and LSTM) as well as state-of-the-art algorithms for machine composition. Tchaibotsky produces near state of the art on entropy analysis. However, this is not a full measure of music quality so further analysis was done.

3 Turing Test Results

An online Turing test was set up using a google forms questionnaire. It started with a question on a self assessment of the user's skill level of their music:

- Professional (Graduate in a music related field)
- Expert (Grade 8 in an instrument or A level in Music)
- Amateur musician (GCSE in music or some experience with instrument)
- Non musician

The user was then presented with four 30 second mp3 music files and posed with a one-tailed head-to-head binomial test on whether they believed each was composed by a human or robot and whether they were performed by a robot or human. The human compositions were chosen to be an excerpt from Skarlatti's piano sonata in D, played by a human and Telemann's Fantasia in D, played by Tchaibotsky. This was chosen by a third party to remove bias and from a less well-known pieces to prevent participant prior knowledge. The participants were told there was one piece of each of the following:

- Computer composed and performed.
- Computer composed and human performed.
- Human performed and computer composed.
- Human performed and human composed.

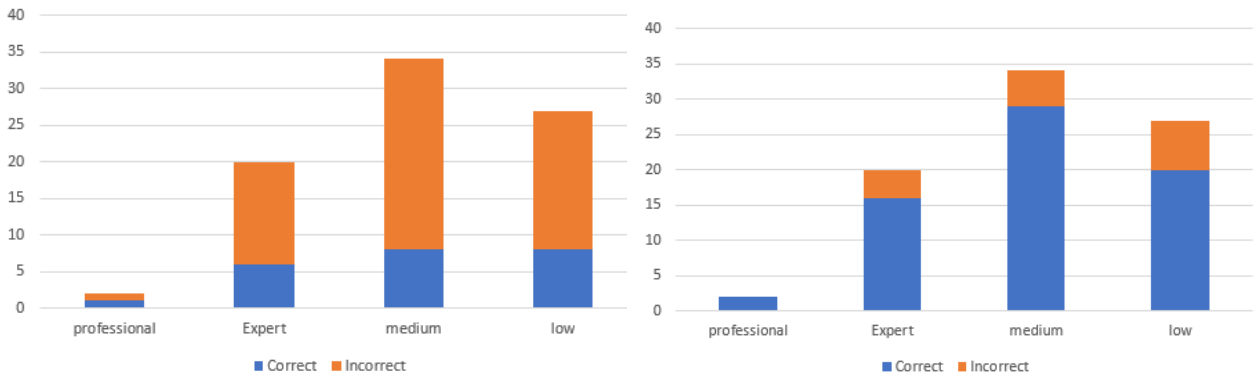

Figure 16 Distribution of participant's guesses on, Left: compositional selection, Right: performance selection.

The accuracy of the participants selections related to their musical ability on all questions is shown in Figure 16. A notes section was added to gather extra evaluation on how human's discriminated between the options. This was added to gain insight on any systematic flaws in the results. To increase the objectivity of the analysis both pieces were recorded on a Yamaha GGX series keyboard and played by the same semi-professional pianist. 39 people responded to the test with the data being shown in Table 2. The confidence intervals are all above 10% due to the lack of participants. Meaningful outcomes can still be extracted from these.

Music type (composition type/performer type)	Percentage of correct composition guesses	Percentage of correct performer guesses
Human/human	35.9% (± 14.7%)	82.1% (± 12.4%)
Human/Tchaibotsky	41.0% (± 15.0%)	25.6% (± 13.7%)
Tchaibotsky/human	69.2% (± 14.2%)	74.4% (± 13.6%)
Tchaibotsky/Tchaibotsky	56.4% (± 15.1%)	15.4% (±11.9%)

Table 2 Statistical data of the selections

4 Discussion

The only previous machine composition Turing test was performed as a head to head comparison against Bach's music. Humans were no better at determining the real compositions than from random chance [20]. This Turing test expands on that study being over the full range of musical domains and made of live performances, although there are significantly less participants.

Although relatively few people responded to the questionnaire, the results are very interesting showing with a high confidence that the composed pieces reliably sound human to a listener, passing the Turing test. This confirmation is statistically assured as the lower confidence interval is below 50%. This is particularly interesting as a 50/50 split was assumed to be the optimal possible outcome; as the algorithm became closer to human standard it would result in guesswork. This apparent 'more human than human' result must show systematic flaw in the testing.

The test has a comparison between too few samples which are of dissimilar genre of music. Some of the participants chose the Skarlatti to be the machine composed piece due to the pattern and rule rich style of the late Baroque era sounding more artificial than the generated romance style, which due to its genre is more expressive. If a non head to head format was incorporated or more complimentary genre pieces were considered this flaw could be minimised.This result also confirms the unexpected success of the algorithm. Participants appeared to assume that the free-flowing style of the machine composed pieces were not possible to have been replicated. Displaying real mastery of this genre over a 30 second interval.

The machine composed and performed piece performed the worst in fooling the participants, with only 6 being fooled. This is probably due to the lack of dynamic contrast and rubato in the performance compared with that in human performed machine composition. Although these features are present in the compositions, the human player exaggerated them excessively bringing relatively more flair to the piece. This, coupled with the lack of calibration on the solenoids reducing the consistency of note-to-note volume gave away the robot's performance. However, some people were still fooled by this, thus achieving some level human standard.

The head-to-head nature could have been the reason the human composed robot performed piece achieved more success. Participants seemed more certain that both of the human composed pieces were computer composed, prompting them to use this knowledge to influence the guess of the performers. The was a 0.7 correlation for more experienced musicians to detect the performer correctly. From the feedback it is determined that this could partly due to the human piano player participants hearing very unconventional playing styles. This was due to the multiple fingered nature of Tchaibotsky as it could play using different techniques than a human. A common technique Tchaibotsky did was to holding down more than ten notes simultaneously instead of using the sustain pedal. This non-human play style could have a positive effect on the musical outcome but does reveal its robotic nature.

The feedback from the test showed there was a general consensus that all pieces were enjoyable and no one claimed to be completely certain of their guess. 3 participants correctly noticed the slight loss of rhythmic pacing and melodic uncertainty from 24-28 seconds before resolving in the machine composed human performed piece. This was caused by a period in the sequence output with high entropy, where Tchaibotsky had multiple similarly high probability outcomes reducing the confidence in its next note and choice between conflicting ideas.

There is no correlation between musical ability and correct choice for the compositional choices, with a weak 0.33 correlation to the less musical participants, excluding those at the professional level as there was only one participant of that calibre. This suggests that similar selection regimes were being performed by all participants. Furthermore it shows that the human characteristics of Tchaibotsky's composition are not surface level or defy basic musical convention.

The algorithmic design involves no explicit musical theory teaching or understanding of prior music. Therefore these results prove that the robot has intuitively acquired the underlining musical theory concepts independently from its input data. Creativity in breaking the conventional rules of music theory is also experienced, for instance at the 23^{rd} second where the algorithm unexpectedly resolves to the perfect 5^{th} rather than the expected 6^{th}. This adds colour to the piece by defying what a human would subconsciously expect the notes the next chord would contain. This demonstrates the real creativity and unique output of the algorithm through its weak constraints. If it dogmatically followed music theory principles these more artistically interesting development would never have been produced.

More Turing tests will be performed after further development of Tchaibotsky. These will be done with a series of 8 questions. Each having a randomly distributed selection of pieces to prevent any head to head comparisons. This ensures each pieces is rated entirely on its own merits. These will be binomial tests questioning whether it is composed and performed by a robot as well as a likert scale to gather data on comparable enjoyment between human and robot music. Giving further data to analyse any correlation between perception of robotic influences in the creation of the music and their listening quality. The pieces will include a variety of different genres and performances by a range of different pianist skill levels, this will help create a more accurate gauge of Tchaibotsky's skill in both these domains. Further time will be taken on these tests with participant collection being performed until a 2% confidence interval.

5 Conclusion

The art of music, although dependant on multiple factors, has been shown to be fundamentally rooted in mathematics. If this was not true a robot could not have scored its successes in the Turing test, nor would it be able to replicate and correctly implement common musical forms and recognisable progressions. The mathematical formula is hidden inside the neural network in an abstract form, it is clear, however, that artificial intelligence has achieved levels of competence in artistic creation. Although the domain of art is pure abstract mathematics many pieces of music are only interesting due to their context and the story the music tells. The music is technically as good as a human, but this aspect of art cannot be reproduced.

Tchabotsky matches previous AI composition and player piano work blending the fields together becoming, to my knowledge, the first robotic artist to complete a Turing test to this level and has capabilities for much more. This prototype could be developed as a novelty, cheaper, alternative to live human professionals in hotel suites or cruise ships and be useful for composition aid, being able to quickly create and demonstrate interesting new music. Further research will be conducted into its own performance of the pieces it creates with improved and deeper testing done to ascertain its potential as a performer. The algorithm will also be updated to be trained on several different genres so Tchaibotsky can be compared more fairly and also help focus its compositions to one style.

APPENDICES

Figure 17 Visual representation of the MIDI file, the x-axis is the time domain and the y-axis is the note value

Figure 17 shows a graphical form of the outputted midi file from the compositional algorithm. This is converted into a 88-bit number for input to the shift registers, with each digit representing a solenoids power state. The steps for this are shown in Figure 18.

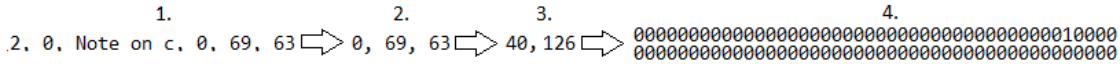

Figure 18 Walk through of a command to play the note A2 at a messo-piano volume occurring at the start of the piece.

1. Convert the midi to a comma seperated variable, CSV.
2. Reduce the unneeded data to a CSV command of time, note, velocity
3. Convert these MIDI pitches to the number of a piano key through Equation 17 and the velocity, v, to the solenoids equivalent duty cycle by Equation 18. These commands are called at an interrupt corresponding to the time from step 2 in ms.
4. Set the digit of the 88 bit number corresponding to the note for the time ratio prescribed in the duty cycle.

$$note_3 = (1 - \frac{note_2 - 21}{88}) * 88 \tag{17}$$

$$v_3 = round(v_2 \frac{(PWM_{max} - PWM_{min})}{127} + PWM_{min}) \tag{18}$$

References

[1] Pianoforte tuner's association. "Upright piano diagram". In: (2019). URL: http://www.aatuners.com/how-it-works.html.

[2] Reuters. "Global Background Music Market 2019, by Types, Top Players, Platforms, Industry Statistics, Trends and Growth Opportunities to 2025". In: (2019).

[3] Anders Thorin, Xavier Boutillon, and José Lozada. "Modelling the dynamics of the piano action: is apparent success real?" In: *Acta Acustica united with Acustica* (2014), p. 10. DOI: 10.3813/AAA.918795. URL: https://hal.archives-ouvertes.fr/hal-01193708.

[4] Hiroshi Kinoshita et al. "Characteristics of keystroke force in the piano". In: *Journal of Biomechanics - J BIOMECH* 40 (Dec. 2007). DOI: 10.1016/S0021-9290(07)70392-5.

[5] Alfred Dolge. "Pianos and their makers". In: (1972), pp. 139–152.

[6] Laurence G. Broadmoore. "Striker solenoid assembly for player and reproducing pianos". In: US5081893A (1990).

[7] Ichiro Kato. "The robot musician 'wabot-2' (waseda robot-2)". In: *Robotics* (1987), pp. 143–155.

[8] Chin-Shyurng Fahn. "Development of a Novel Two-Hand Playing Piano Robot". In: *International Conference of Control, Dynamic Systems, and Robotics (CDSR'17)* (2017), p. 106.

[9] J. A. E. Hughes. "An anthropomorphic soft skeleton hand exploiting conditional models for piano playing". In: *Sciences Robotics* (2018), p. 2.

[10] Matteo Suzzi. "Teotronico the Anthropomorphic Piano Playing Robot". In: *Sciences Robotics* (2019), p. 2. URL: http://www.teotronico.it/who/..

[11] C. Vamvacas. "The Founders of Western Thought – The Presocratics". In: *Springer* (2009).

[12] Arnold Schönburg. "The Theory of Harmony". In: (1948).

[13] M. Guerino. "The Topos of Music: Geometric Logic of Concepts, Theory, and Performance". In: *Birkhäuser* (2002).

[14] G. Wohl. "Algebra of Tonal Functions". In: *Springer* (2005).

[15] G Nierhaus. "Algorithmic Composition: Paradigms of Automated". In: *Acta Acustica united with Acustica* (2009).

[16] D.A. Levitt S.M. Schwanauer. "Machine Models of Music". In: *Cambridge : MIT Press* (1993).

[17] A. Willis T. Collins R. Laney and P. Garthwaithe. "Developing and evaluating computational models". In: *Cambridge University Press* (2016).

[18] S. Cherla. "Neural probabilistic models for melody prediction, sequence labelling and classification". In: *Semantic Scholar* (2016).

[19] P. Vincent N. Boulanger-Lewandowski Y. Bengio. "Modeling Temporal Dependencies in High-Dimensional Sequences". In: *ICML* (2012).

[20] Feynman T. Liang et al. "Automatic Stylistic Composition of Bach Chorales with Deep LSTM". In: *ISMIR*. 2017.

[21] Cheng-Zhi Anna Huang et al. "An Improved Relative Self-Attention Mechanism for Transformer with Application to Music Generation". In: *CoRR* abs/1809.04281 (2018). arXiv: 1809.04281. URL: http://arxiv.org/abs/1809.04281.

[22] OpenAI. "MuseNet - Overview". In: (2019). URL: https://openai.com/blog/musenet/#fn1.

[23] Crunchbase. "Jukedeck Overview". In: (Oct. 2012). URL: https://www.crunchbase.com/organization/jukedeck#section-overview.

[24] Radio Spares Components. "IRLB8721PBF| N-Channel MOSFET product datasheet". In: (2019). URL: https://uk.rs-online.com/web/p/mosfets/7259322/.

[25] H. Schlichting. "Boundary Layer Theory". In: *Gersten* (1951).

[26] Engineers toolbox. "Modulus of Materials Database". In: (2019). URL: https://www.engineeringtoolbox.com/young-modulus-d_417.html.

[27] Radio Spares Components. "SB240 Schottky diode product datasheet". In: (2019). URL: https://docs.rs-online.com/40dc/0900766b81300b24.pdf.

[28] Radio Spares Components. "PLA product datasheet". In: (2019). URL: https://docs.rs-online.com/7eed/0900766b8157ce85.pdf.

[29] Schaeffler. "Schaeffler Pocket Guide". In: (2015).

[30] Engineers toolbox. "Friction coefficient Database". In: (2019). URL: https://www.engineeringtoolbox.com/friction-coefficients-d_778.html.

[31] Andrew Blake and Andrew Zisserman. "Localising discontinuities using weak continuity constraints". In: *Pattern Recognition Letters* 6.1 (1987), pp. 51–59.

[32] Sageev Oore et al. "This Time with Feeling: Learning Expressive Musical Performance". In: *CoRR* abs/1808.03715 (2018). arXiv: 1808.03715. URL: http://arxiv.org/abs/1808.03715.

[33] Curtis Hawthorne et al. "Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=r1lYRjC9F7.