Bar, Rope, String: An Exploratory Study into Relationships between Physical Interfaces and Digital Sound Synthesis

Olaf Wisselink (s2690233)
Graduation Thesis
Media Technology MSc program, Leiden University
Completed 25-08-2023
Primary supervisor: Edwin van der Heide
Secondary supervisor: Maarten Lamers

Abstract

In digital musical instruments, there is a clear separation between sound production: the sound synthesis, and control: the physical interface, typically consisting of a MIDI keyboard or controller. Where acoustic instruments often have inherent relationships between control and sound production, we have to explicitly design these relationships for digital instruments. For this research we are interested in exploring what happens when we blur the separation between the physical interface and digital sound synthesis and give the control interface also sonic properties. We develop three physical interfaces that vary in their ability of control and actual sound production, and in their similarity to the synthesis model. For the sound synthesis we use a plucked string algorithm. For each of the interfaces, we perform different mapping experiments in which we create different relations between the output of the interface and the input of the synthesis model. From the experiments we observe several advantages of a blurred approach towards sound production and control, such as a tight relation between gesture, motion and sound, and parameter coupling. We make use of parameter coupling in hardware, by physically coupling the sensors, and in software by mapping interface parameters to multiple synthesis parameters and by obtaining control parameters from audio via signal analysis. Parameter coupling makes it possible to create complex relations, such as controlling many parameters at the same time with one gesture. However, we note that our approach of blurring also has disadvantages. We observe that obtaining control parameters from audio can cause analysis errors. Moreover, signal analysis adds layers of abstraction which can make the interface-synthesis relation less direct. We note that it can be beneficial to have some degree of separation between the sonic properties and control properties of the interface, by for example using different sensors to measure them separately. Regarding the similarity between the interface and the synthesis, we observe that less similarity allows for more freedom to make different relations. More similarity between the interface and the synthesis conditions, and thereby limits, their possible relationships.

1. Introduction

In most acoustical musical instruments, there is an inherent connection between the way the instrument is controlled and the sound which the instrument produces. The instrument is both a source of sound and an interface: a means of controlling the sound. For example, a string on a violin is both the sound source and a means for the player to control the sound. Furthermore, because of the physical properties of the instrument, complex interactions can exist between different musical parameters. Bowing the string on a violin faster or stronger causes not only an increase in volume, but also produces a brighter sound. In this way amplitude and brightness are connected to each other and are controlled simultaneously by the same action. In digital music instruments there is no such connection, but rather a clear separation between the sound source and the interface. Their relations must be explicitly designed. In a digital musical instrument, the sound source is a synthesis algorithm. The interface consists of a physical control device. Typically this is a MIDI keyboard, or a controller with buttons, sliders and knobs. Making relations between them consist of connecting the output parameters of the interface to input parameters of the synthesis algorithm in a particular way. Making these connections is also known as mapping (Miranda & Wanderley, 2006). For this research we define the separation between the interface and the sound synthesis as follows. Firstly, the typical keyboard or controller interface does not produce any sound itself, but only control data, meaning its output parameters do not consist of continuous audio signals, but event data that are meant to control audio signals generated by the synthesis algorithm. This data could for example be a keypress, or the value of a potentiometer. Secondly, there is no inherent connection between individual interface parameters. For example, two knobs could separately control sound parameters like pitch and amplitude, without influencing each other. Thirdly, there is no inherent similarity between the physical qualities of the interface and the parameters of the synthesis. For example, a MIDI keyboard could potentially be used to play any model of a physical instrument such as a clarinet or a drum kit, or a completely synthesized sound.

For this research we are interested in the interface-synthesis relationship specifically regarding this described concept of separation. We ask the following question: how does the relationship between the physical interface and the sound synthesis change when their separation is blurred? The research is done within the following context. We focus our work around one type of sound synthesis and three different physical interfaces. The sound synthesis consists of a plucked string algorithm. The reason for using physical modeling synthesis is to allow for a varying degree of connection between the parameters of the model and the parameters of the interfaces. The three interfaces vary in their capacity of being both a sound source and controller, and in their similarity with the synthesis. The research question can therefore be made more specific by asking what kind of physical interfaces are possible for a plucked string synthesis algorithm, based on the described concept of separation. Additionally, the goal of this research is to observe more general principles of the relationship between physical interfaces and sound synthesis that go beyond the specific interfaces and synthesis methods used in this research. The next section will describe in more detail the practical research setup, describing and motivating the sound synthesis and the three physical interfaces.

2. Research Method

2.1. Research Setup

To observe the relationship between the physical interface and digital sound synthesis we create and compare three similar physical interfaces that are connected to the same plucked string synthesis algorithm, but that gradually vary in their amount of separation with the synthesis algorithm. For each interface we will perform experiments, in which we try different ways of interpreting the interface as both a source of sound and control, and try out possible mapping strategies. We then compare the interfaces by their potential similar and different qualities in relationship to the sound synthesis. Additionally, we base our work on related digital musical instruments and interfaces, and on research done more broadly on instrument interface design, mapping and signal analysis. This is described in section five.

The interfaces and the sound synthesis are designed around two general principles. Firstly, they are both designed to be relatively simple, in terms of their number of parameters, the diversity of sounds that they can produce and in the possible ways they can be played. The goal is not to produce highly specific and virtuosic musical instruments, but to observe more general principles of the relationship between physical interfaces and sound synthesis by making similar, but gradually varying, basic instruments.
Secondly, as mentioned, because of the inherent disconnection between the interface and sound source in a digital musical instrument, the number of possible mappings between their parameters is great, but it is not always immediately evident which mappings are more successful than others. Additionally, (Miranda & Wanderley, 2006) note that interfaces that closely model existing acoustic instruments, or ‘instrument-like controllers’, often inherit the same limitations and are therefore limited in the amount of sounds that they can successfully produce. They mention as an example that on the typical MIDI keyboard it is very difficult to produce a convincing sound of a woodwind instrument, because the complex attack is difficult to replicate on a keyboard. This is of course only a limitation if the goal is to closely reproduce the sound of an existing instrument. Other goals are possible, such as producing sounds that are not reproductions of existing instruments, or that bear little relation to the physical world. In our mapping experiments with the interfaces we will initially focus on relationships that are known from acoustic instruments. From there we will look at other types of relationships specifically shaped by the interface and sound synthesis. The goal is not to try to completely reproduce existing instruments. Similarly, the reason for using a physically informed model is to allow for a varying degree of connection between the parameters of the model and the parameters of the interfaces. The goal is however not to solely make relations that are physically informed. The next section will describe each interface in more detail. The sound synthesis will be further described in section four.

3. Physical Interfaces

3.1. BAR - STRING Interface

The first interface is called ‘BAR - STRING’. It consists of a solid wooden bar of approximately 1.2 meters in length. It is played by hitting or scraping the bar. These two actions are defined as the main gestures to interact with the interface. It is therefore mainly a percussive interface. The interface is designed to be played with two mallets, but any object could be used, including the hands. Two contact microphones placed on the left and right side of the bar pickup the vibrations of the bar. The two signals from the microphones are the direct outputs of the interface. They are sampled at a sampling rate of 44100 Hz with an audio interface. Figure 1 shows a picture of the BAR - STRING interface.

Figure 1: the BAR - STRING interface

This BAR - STRING interface is designed to be the most separated from the sound synthesis. Firstly, this separation regards the interface's capacity of being both a controller and sound source. On the one hand the interface can be considered to be only a sound source, and not a controller, since its outputs consist of only the audio signals from the contact microphones. On the other hand, it is also possible to think of these signals more in terms of control, rather than as a sound source. The signals from the contact microphones generally consist of two types, namely a short impulse when the bar is hit, or a continuous noisy signal when the bar is scraped. Because the interface is kept very basic, without complex acoustic properties, such as a resonating chamber, these signals are acoustically not very complex, and do not significantly contain properties such as a decay, resonance or pitch. In this sense they can be regarded as more complex control triggers, such as electronic drum triggers, or a trigger from a keypress or touchpad.
Secondly, regarding the similarity with the sound synthesis, the interface is designed to have two analogous properties. The first property relates to how the bar is played and the input to the string model. Typically, a plucked string model typically is injected with noise, either a short burst of noise resembling a plucking gesture, or a continuous noise resembling a bowing

gesture. The type and timbre of the input influences the timbre of the string sound. The two main gestures of the interface are similar to these types of input. Hitting the bar produces a short impulse and scraping the bar produces a continuous, noisy signal. The timbre of the gesture is determined by the amount of energy in the strike and with what kind of object is used. The second shared property relates to the spatial properties of the interface and the spatial parameters of the plucked string model. The plucked string algorithm is spatially a one-dimensional model. Its spatial parameters consist of a pluck and listening position along the string. The physical shape of the interface approximates this one-dimensional shape in the real world, with its length being the most significant dimension. It therefore mainly affords playing along its length. Because the interface has two spatially separated microphones, it is possible to obtain an approximation of where the bar is played along this length via analysis of the microphone signals, providing a parameter similar to the pluck position of the model. We define such parameters derived from signal analysis as indirect output parameters of the interface. This is described in more detail in the experiments in section four.

3.2. ROPE - STRING Interface

The second interface is called 'ROPE - STRING'. It consists of a rope of approximately 1.5 meters in length, tied over bridges on both ends of a wooden bar. It is played by plucking and pulling the rope with the hands. We define these actions as the two main gestures of the interface. An MPU6050 motion sensor, containing an accelerometer and gyroscope are placed on either side of the rope, measuring the movements of the rope in three dimensions as it is being plucked and pulled. In addition, the accelerometer and gyroscope measurements together provide information regarding the orientation of the rope, for example at what angle the rope is pulled up or down. Each motion sensor is sampled at a sampling rate of 1500 Hz, by an ESP32 connected to a computer through a serial connection. Figure 2 shows a picture of the ROPE - STRING interface.

The interface is designed to be less separated with the sound synthesis compared to the first interface. Regarding the overlap between control and sound production, we regard this interface as being both a controller and a generator of sound. Unlike the first interface, which mostly produces discrete triggers with no properties such as pitch or decay, this interface does produce signals that vary continuously over time and which do have properties such as a pitch and decay, because they are inherent to the behavior of the rope. Conversely, it also produces signals that can be seen as potential control variables, such as the vertical orientation of the rope. Such a signal is under complete control of the player. Regarding the similarity with the sound synthesis model, the rope can be said to behave as a 'proto string'. Similar to a string on for example a guitar, plucking the rope causes it to vibrate across a series of harmonically related frequencies, together producing a standing wave in the rope. The resulting pitch of the string is dependent on the length, thickness, and tension on the rope. However, the applied tension is not as high as to produce a definite pitch in the audible range, but rather a very low frequency. In addition, because of the physical nature of the rope and irregularities in construction it does not vibrate completely periodically. It therefore does not behave in completely the same way as a string. It rather behaves as a semi-periodic, dampened low frequency oscillator. This makes the rope potentially both a source of sound and control similar to how a low frequency oscillator (LFO) is a commonly used synthesis technique as a source of modulation for a certain parameter of a sound.

Similarly to the first interface, the sensors on the ROPE-STRING interface are spatially separated, which allows for an approximation of where the rope is played along its length via analysis of the sensor signals, providing a parameter similar to the pluck position of the model.

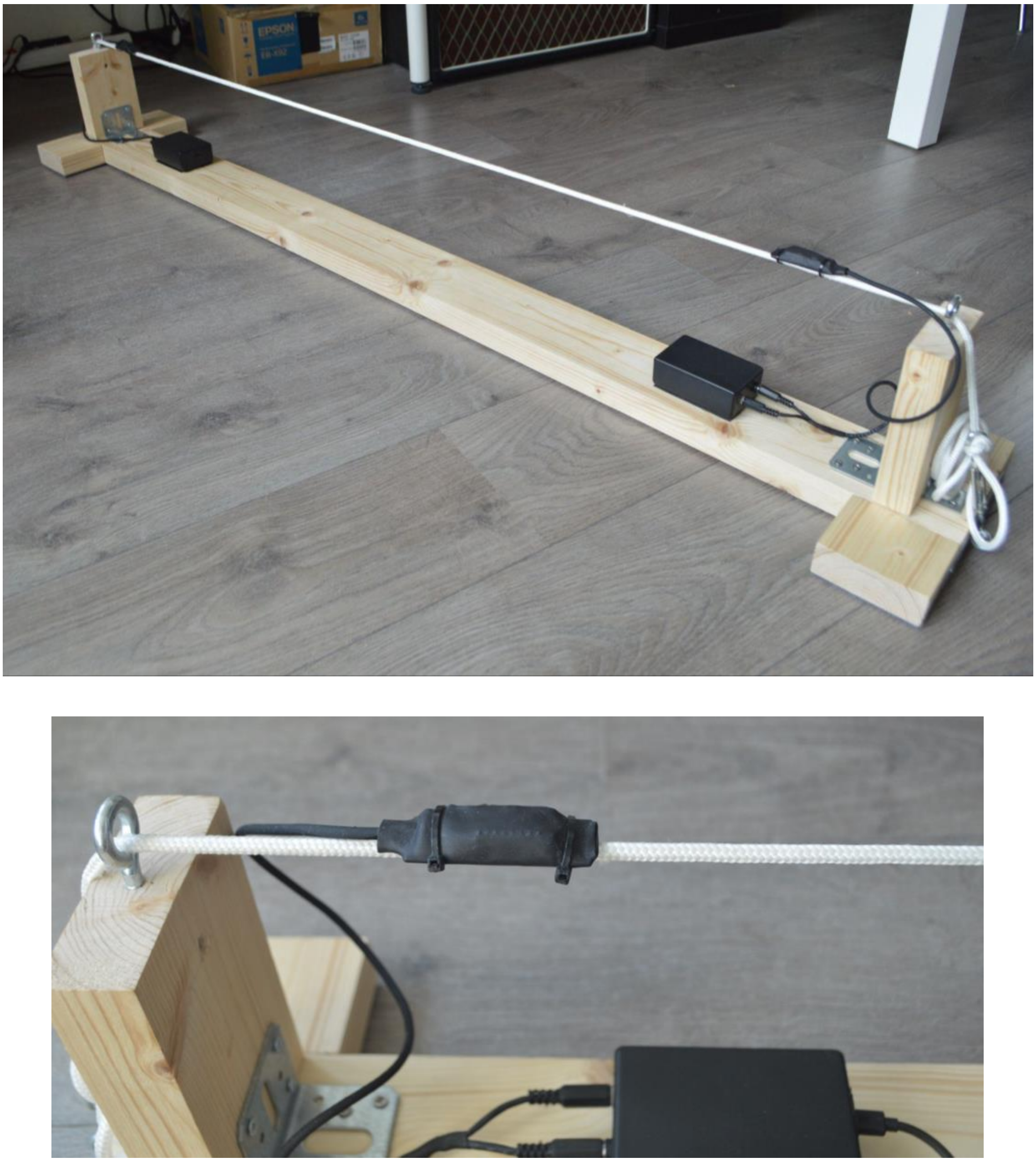

Figure 2: the ROPE - STRING interface, showing below a detail of the motion sensor on one side of the rope.

3.3. STRING - STRING Interface

The third interface is called STRING - STRING. It consists of a steel string with a diameter of 2 millimeters, approximately 1.5 meters in length, tied over bridges on both ends of a wooden bar. It is played by plucking and pulling the string with the hands, or with a bow. Similar to a monochord, it has a third moveable bridge allowing for the shortening of the string to play different pitches. The string can be tuned using a wire tensioner. Because the string is relatively thick, its fundamental frequency is low, or even in the sub audio range, depending on the tuning. Two bass guitar pickups placed parallel to the string on either side are used to pick up the vibrations of the string. Additionally, two small neodymium magnets are attached to the ends of the string, both with a hall sensor placed underneath. The output of the hall sensor is proportional to the strength of the magnetic field, which in turn is inversely proportional to the distance between the sensor and the magnet. The hall sensors therefore measure how far the string is pulled up or down. This information is not represented in the pickups, because their output corresponds to velocity of the string, which does not include static position information, because this is a DC component. The pickups are sampled at a sampling rate of 44100 Hz with an audio interface. The hall sensors are sampled at a sampling rate of 1170 Hz by an Arduino Nano connected to a computer through a serial connection. Figure 3 shows pictures of the STRING - STRING interface, with a detail of the sensors and magnet.

The STRING - STRING interface is designed to be a variation of the ROPE - STRING that is even more similar to the sound synthesis. This similarity makes it potentially straightforward to connect the corresponding parameters of the interface and the sound synthesis to each other. This could be meaningful if the goal is to produce an 'instrument-like interface' (Miranda & Wanderley, 2006). However, the goal of this interface is not so much to only reproduce relations known from acoustic instruments, but also to try to produce other possible mapping strategies that are not physically informed. Similar to the ROPE - STRING, we also regard this interface as being both a controller and a generator of sound. It produces signals with certain properties related to the behavior of the string, which vary continuously at either audio or sub audio rate. Conversely, it produces signals that can be seen as potential control variables, such as the vertical orientation of the string, which is under complete control of the player.

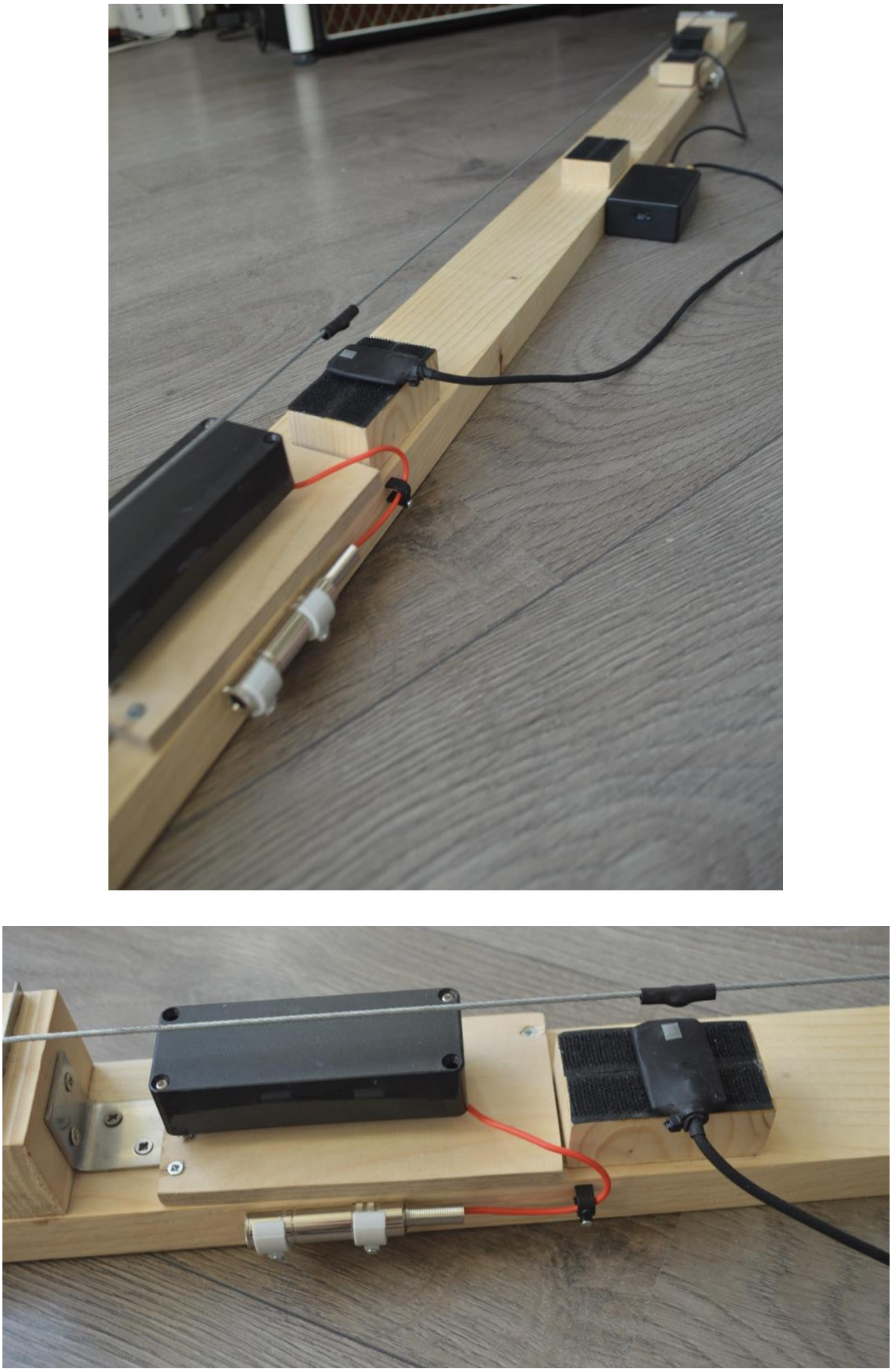

Figure 3: the STRING - STRING interface with below a detail of the pickup (left), the hall sensor (right) with the magnet above it on the string, on one side of the string.

4. Sound Synthesis Algorithm

4.1. Plucked String

The digital sound synthesis algorithm consists of a physical model of a plucked string, implemented by means of a digital waveguide, based on the model by (Smith, 1992; Berdahl & Smith, 2008). A digital waveguide is a physically informed computational model of how waves travel through a physical medium. For example, it can model how waves travel across a one-dimensional string, on the surface of a drum skin or through the area of a tube.
A digital waveguide is a generalization and extension of the well-known Karplus-Strong algorithm which was originally developed to simulate string-like sounds (Karjalainen et al., 1998; Karplus & Strong, 1983). A digital waveguide model can be very complex with many parameters. For the goals and scope of this research we limit ourselves to a basic implementation of the model. It has six main parameters which can be controlled. Similarly, we choose to work with only one instance of the model. It is therefore monophonic, meaning that we cannot play multiple strings at the same time. Polyphony and other possible extensions are addressed in the discussion in section seven.

Table 1 below shows a summary of all six parameters, their amount and type. Every parameter is of type signal, meaning it can vary continuously, at audio rate. The parameters will each be described shortly. The first parameter is the input to the string. The waveguide model needs to be injected with energy to produce sound. Typically this is done with an impulse, or a burst of noise to resemble the short, high energy excitation signal caused by the plucking of the string. The input could also be a continuous noisy signal, resembling the bowing of the string. The use of a burst of noise was originally proposed in the Karplus-Strong algorithm, but in fact any signal could be used as input. The second parameter is the pitch of the string, or its fundamental frequency. The third parameter is the decay time, meaning the time that the string continues to sound after it has been struck. The fourth parameter is the amount of low pass filtering, causing higher frequencies decay to faster, making the string sound less bright. The fifth parameter is the pluck position. This refers to a point along the string at which it is excited. Plucking the string at a certain location changes the timbre, because only harmonics with an antinode at that location will be excited, harmonics with a node will not. For example, plucking the string exactly at the center will only excite the uneven harmonics because they have an antinode at this location. The even harmonics have a node at the center. Finally, the sixth parameter is the pickup position, referring to a point along the string at which we listen to the output. The pickup position affects the timbre in a way similar to the pluck position, reinforcing or decreasing harmonics depending on the listening position. This is similar to how in an electric guitar the bridge pickup sounds brighter than the neck pickup because it is closer to the nodes of the lower harmonics.

	1	2	3	4	5	6
Synthesis Parameter	String input (2x)	Pitch	Decay time	Brightness	Pluck position	Pickup position (2x)
Type	Signal Continuous					

Table 1: Sound synthesis parameters

Practically, the plucked string model is implemented in the Max/MSP programming language, specifically using the gen~ environment. The gen~ environment operates on a time resolution of one sample, which makes it possible to work with very small delay lines. This allows for accurate control and greater range over the pitch of the string. Additionally, we choose to use two inputs and two pickup positions in the model. The reason for doing so is to make use of the fact that each interface has two (or more) spatially separated sensors. This makes it possible for the string to be 'stereophonic' in the sense that we can potentially play the string with two different signals and play back the output from two different pickup positions going to the left and right speakers in a stereo setup.

4.2. Parameter Coupling and Behavior

The described parameters are part of the affordances of the string model because it is only these parameters that it allows control over. But as mentioned, the string model also has a certain behavior, with which we mean that some parameters are inherently coupled to each other: a change in one parameter can cause a change in another parameter. Therefore, we say that the affordances of the string model not only relate to its parameters, but also to how these parameters are connected. Practically, this mainly concerns the connections to the decay time of the string. Firstly, the pitch of the string affects the decay time, such that a higher pitch results in a shorter decay time. Secondly, the amount of low pass filtering affects the decay time of high frequencies, but it also affects the overall decay time of the sound. If the amount of low pass filtering is high, the overall decay time is shorter. These connections can be seen as inherent relations, or mappings, that reproduce the behavior of a real physical string. Importantly, these inherent connections can be also decoupled again to produce mappings that are not physically informed., for example by proportionally adjusting the decay time depending on the string frequency.
Additionally, the affordances of the string also refer to how it allows control over its parameters. This especially regards the input parameter of the string. The input can be any type of signal. For example, the input can be discrete, similar to a key press on a piano, which has a clear beginning and end without any variation in between. The input can also be continuously varying, similar to how a held note on a wind instrument can continue to change. This affordance makes the plucked string algorithm very flexible and suited to many types of input from the interfaces. For example, it is possible to feed the signals of the sensors of any of the interfaces directly into string, or a signal derived from the sensors, such as an impulse when an onset is detected. Or it can be a signal that is also used to control other parameters of the model, such as the pitch. Because of this flexibility, the plucked string algorithm does not make

a clear distinction between signals that are intended for the control of parameters and signals that are intended to generate sound. It is therefore possible to have an overlap between control and input signals, which is very similar to how an acoustic instrument behaves both as a source of sound and as an interface for control over the sound.

5. Related Work

5.1. Conceptual Framework

The following section will address theory and techniques related to our work. Together they are meant to provide a conceptual framework to describe and compare our interfaces, and to map out common techniques and approaches that we can use in our experiments. We discuss three different concept categories, firstly mapping and parameter coupling, secondly the distinction between signal and control, and thirdly affordances.

5.1.1. Mapping and Parameter Coupling

As mentioned in the introduction, in the traditional MIDI keyboard or controller interface each interface parameter is mapped separately to each synthesis parameter. This is unlike an acoustic instrument in which parameters of the sound are tightly coupled to each other. Regarding mapping and parameter coupling, (Hunt & Kirk, 2000) propose two desired characteristics of digital musical instrument interfaces, based on the workings of acoustic musical instruments. Firstly, they note the interface should be multi-parametric, meaning it is capable of continuous control over multiple synthesis parameters at the same time. Secondly, the parameters of the interface should be coupled together, meaning that (some) interface parameters should be mapped to multiple synthesis parameters. Hunt & Kirk note several common mapping strategies, which are also mentioned in (Baalman, 2022; Miranda & Wanderley, 2006). The simplest strategy is a one-to-one mapping, which means that one output parameter of the interface is connected to one input parameter of the sound synthesis. A one-to-one mapping is per definition not a form of parameter coupling. The following mapping strategies are forms of parameter coupling. Firstly, it is possible to connect one interface parameter to multiple synthesis parameters. This is a one-to-many mapping, divergent mapping. Vice versa, connecting the sum of multiple interface parameters to one synthesis parameter creates a many-to-one, or a convergent mapping. Combining both, creates a many-to-many mapping, in which the sums of multiple interface parameters are mapped to multiple synthesis parameters. Moreover, these summed mappings can be weighed so that each parameter contributes with a different amount. Additionally, Hunt & Kirk describe the concept of biasing, in which a parameter needs to be above a certain threshold for it to have an effect. Multiple one-to-one, convergent and divergent mappings, weights and biases can be made in parallel, together resulting in a complex, nonlinear interaction, similar to the behavior of acoustic instruments.

In the experiments of this research, we make use of the described mapping strategies and concepts to make relations between the interfaces and the sound synthesis. Additionally, in order to describe these mappings, we adopt a visual mapping language developed by (Baalman, 2022). The language consists of diagrams, showing the connections and operations between the interface outputs and the synthesis inputs. For each interface, we will make the diagrams to highlight and demonstrate parts of the mapping experiments.

5.1.2. Signal and Control

(Baalman, 2022) categorizes amongst others two main types of data. Firstly, they define the data stream as a continuous stream of values without a beginning or end. Our definition of audio-rate signal can be seen as a subcategory of the data stream. Namely, the data stream could be an audio-rate signal from a microphone, but also data from different sensors at different rates. Secondly, they define an event as a moment in time with a specific beginning, a possible duration, an end and possible other characteristics such as loudness or pitch. If an event has no duration, it is instantaneous. Our use of the term control data with regard to the typical MIDI keyboard interface can be seen as a part of the event data type. If we regard the MIDI keyboard and controller interface both in terms of data streams and events, we can say that these interfaces do not generate data streams, but only event data meant to control data streams (audio-rate signals) produced by the sound synthesis. In this way data streams and events are clearly separated and independent of each other. In our interfaces we try to interpret the output parameters both as continuous data streams and as events, without such clear separation. Baalman defines this interpretation as transmutation. Transmutation can occur in two directions. Firstly, it is possible to translate a continuous data stream into a discrete event by analysis of the data stream. For example, by looking at when the data stream passes a certain threshold, or by (periodically) sampling and holding a datastream. In the sample and hold example the datastream is discretized not in value, but only in time. It is also possible to discretize the datastream in value, but keep its continuous nature, by for example rounding the values of a datastream. A combination of both is of course also possible. Vice versa, it is possible to translate an event in a data stream by synthesis of multiple events over time. For example, by continuously integrating the value of a potentiometer over time. Moreover, Baalman mentions that data streams and events can occur within other events. An example of multiple events within an event is the typical Attack, Sustain, Decay, Release (ADSR) envelope. The envelope is an entire event containing four distinct sub events.

5.1.3. Affordances

A term that is commonly used in interface design in Human Computer Interaction (HCI) is "affordance". We think the concept of an affordance is also useful for our purposes in thinking about musical interface design. In the context of HCI, concerning specifically the typical computer interface with a keyboard, mouse, and screen, an affordance is defined by (Norman, 1999) as a perceived actionable property of the interface by the user. Norman specifically uses the term "perceived" to distinguish between the actual possible actions and what the user considers to be possible actions. For example, the actual physical affordances of a computer mouse are the abilities to click and move it. The perceived affordances of the mouse relate to how the actual affordances are communicated to and interpreted by the user.

Typically they are communicated through graphics on the screen, such as icons, representing where the user can click, and cursors, representing the state and position of the mouse. For our purposes in this research, we use the concept of affordances not in terms of actual and perceived affordances, but in how the physical affordances of the interface are represented in the output parameters of the interface. The physical affordances relate to what kinds of actions, or gestures are possible the interface allows for. For example, the bar interface affords hitting and scraping gestures at a certain location and with a certain loudness and timbre. We consider the way in which these physical affordances are represented in the output parameters as part of the affordances of the interface. Namely, a gesture could be physically possible on the interface, but not captured by the sensors (or captured differently), or it could be hard to measure via signal analysis. This is mostly a technical description of affordances, without considering the perceived affordances by the player, but it is already an important factor in shaping what the interface can and cannot do, how it is played and what sounds can be made. The described concept of affordances will be practically addressed in the experiments section in which we obtain the output parameters of each interface.

5.2. Related Interfaces and Musical Instruments

In recent years many new digital musical instruments with novel interfaces and ways of interaction have been developed that provide alternatives to the traditional MIDI keyboard and controller-based interfaces. Noteworthy examples can be seen in the annual conference on New Interfaces for Musical Expression (NIMEconference, n.d.), or in the relatively recent history of new digital musical instruments by (E. R. Miranda & Wanderley, 2006). Here we will discuss five such physical interfaces that are related to our three interfaces.

An early example of an interface related to the BAR - STRING interface is The Sequential Drum, developed by Max Matthews and Curtis Abbot (Mathews & Abbott, 1980). The Sequential Drum consists of a rectangular surface which can be played with the hands or a drumstick. The interface has four output parameters that are sent to a synthesizer program on a computer. The parameters are all signals. They consist of a trigger from a contact microphone that occurs when the drum is hit, a measure of the loudness of the hit, and two signals from a grid of grounding wires corresponding to the horizontal and vertical location of the hit on the surface. Matthews and Abbot note that the signals could be used in various ways. Their main approach is to use them to traverse sequentially through events of a predefined score. Typically an event consists of a note with a predefined pitch. The trigger output starts a new note. The amplitude signal controls the loudness, and the position signals control the timbre of the note. The player therefore controls all parameters, including time, of the music except pitch. The Sequential Drum is very similar to the BAR - STRING interface in terms of its percussive way of playing, and its types of output parameters. Both interfaces make use of contact microphones and perform analysis on their signals to obtain loudness and onset (trigger) information. Both interfaces also output position information. However, the Sequential Drum does not use the signal of the microphones in the synthesis directly. This relates to the main difference between the interfaces. The BAR - STRING is used both as a controller and as a direct source of sound, while the Sequential Drum is used more as a controller and conductor to traverse a score.

An instrument related to the ROPE - STRING interface is The Web, by Michel Waisviz (Waisvisz, n.d.). The Web consists of an interconnected web of wires that can be manipulated with the hands. Sensors containing a magnet and hall sensor detect the movement and tension of the wires (Bongers, 2000). Because the wires are interconnected, pulling one wire causes a change in tension in multiple other wires. This makes it possible to get many different, but interrelated output parameters from one hand gesture, which in turn can get mapped to many parameters of the sound synthesis. The concept behind The Web very much relates to the concept of mapping and parameter coupling. Although now the coupling does not have to be done in software because it is already happening physically in the interface. One could map the output of each sensor separately to a sound parameter and still obtain parameter coupling, because a change in wire tension in one place causes a change in all the connected wires, causing a change in all the sound parameters mapped to the related sensors.

Another instrument related to ROPE-STRING is Soundnet by Sensorband (Bongers & Sensorband, 1998; Sensorband, n.d.). Inspired by the Web, Soundnet is a life sized version of the same concept. It consists of a web of 11 x 11 meter of interconnected ropes. At the end of the ropes, eleven sensors detect the stretching and movement caused by the players climbing the net. Similar to the Web, because the ropes are interconnected, the sensors are physically coupled together. Additionally, the instrument is designed to be played by multiple performers at the same time. One could therefore say that the performers are also coupled together, because the players physically influence each other through their movements on the net. The output from the sensors directly controls the processing of recordings of natural sounds, such as filtering and waveshaping. Both The Web and Soundnet are related to STRING-STRING because they all make use of the behavior and manipulation of ropes or string-like material to control sound. They all also make use of parameter coupling through physically making the sensors interdependent, although in The Web and the Soundet this principle is explored and amplified more than in our rope interface.

An instrument related to both the ROPE-STRING and STRING-STRING interfaces is the STRIMIDILATOR by Marije Baalman (Baalman, 2003). The STRIMIDLATOR is a string controlled MIDI instrument, consisting of four strings tied over a frame. The concept behind the instrument is to use the deviation and vibration of strings to control synthesized sound. The deviation of the strings is captured with linear transducers attached directly to the string. The vibration of the strings is captured using electric guitar coil pickups. The output of both sensors is translated into MIDI signals, either directly, or through an envelope follower. Additionally, the output of the coils is used as a direct audio output. The output signals from the interface are used to control parameters of various sounds, for example using the direct coil output as an LFO to control the vowel frequencies of a vocal filter. The ROPE-STRING and STRING-STRING are directly inspired by the STRIMIDILATOR. Their underlying concepts are therefore very similar. For example, we also try to capture both the deviation and vibration of the strings, and use the sensors both as audio data and as control data. However the implementation, and therefore the qualities of the interface are different. Most notably, the STRIMIDILATOR makes use of multiple strings and the sensing of deviation and vibration happens on separate strings. Also, Baalman makes use of various synthesis techniques, rather than only plucked string synthesis. Our rope and string interfaces could be regarded as a simplified one-stringed variation on the STRIMIDILATOR. In the discussion in section seven we note similar extensions such as using multiple strings and sensors.

Lastly, an instrument related to the rope and string interfaces is the Global String, developed by Atau Tanaka and Kasper Toeplitz (Tanaka, 2001). The concept behind Global String is to create a networked instrument which is physically separated between two locations, but connected virtually through a shared sound synthesis. It consists of two steel strings each placed at a separate location. Each string is equipped with piezo and hall sensors that capture respectively the fast and slower movements of the string, and actuators to physically move the string based on the input from the sensors of the other string half at the other location. The sound synthesis consists of a physical model of a string. The instrument allows for players to interact with each other through the string at two different locations at the same time. The physical interface of Global String, the sensors and sound synthesis are similar to our rope and string interfaces, and also to the STRIMIDILATOR by Baalman. Global String also makes use of similar mapping strategies as in the STRIMIDLATOR. This mainly concerns the use of the sensors as both audio and control data. For example, in Global String the fast changing signals from the piezo sensors are used as direct input for the physical model. The slower changing data from the hall sensor is used to control the qualities of the sound. We experiment with similar mapping strategies in the experiments for the rope and string interfaces.

6. Mapping Experiments

6.1. Interface 1: BAR - STRING

6.1.1. Output Parameters and Signal Analysis

As mentioned in section two, the BAR - STRING interface is designed to be a percussive interface that generates signals that are similar to the type of input signals typically used in plucked string synthesis, namely a short impulse or a continuous noise. These signals are represented directly by the two contact microphone signals, meaning they could be used as direct input into the string without making any explicit analysis or mapping. Therefore we define the two microphone signals as direct outputs of the interface. Additionally, we decide to obtain other information regarding when and how and the bar is played from analysis of these signals. Rather than being part of the sound production of the interface, these parameters can be regarded as control parameters to control the synthesis parameters. From the signal analysis we obtain six additional parameters. Table 2 shows a summary of all output parameters.

Firstly, we use an amplitude follower on both the contact microphone signals. The envelope follower has a fast attack and a slow decay, to capture both a quick increase in amplitude of a hit and to slowly follow the amplitude during a scrape gesture. Secondly, we use an amplitude based onset detection algorithm to determine the exact moment the bar is played. We do this independently for both microphones. A detected onset for both microphones defines a hit gesture. A scrape gesture is defined if the amplitude after the onset stays above a threshold for some time after the onset. Thirdly, in order to obtain a parameter similar to the brightness parameter of the synthesis, we determine the brightness of a strike from spectral analysis. From the spectrum the spectral centroid and spectral spread are calculated, using the Zsa Descriptors library in Max/MSP (Malt & Jourdan, 2008). The spectral centroid is the weighted frequency mean of the spectrum. The spectral spread is the variance around the spectral centroid and describes how energy is spread out around the spectral centroid (Malt & Jourdan, 2008). The spectral centroid and spread give an indication of the energy distribution of the strike, corresponding to the perceived brightness of the sound. Fourthly, this same analysis is used to distinguish between two different gestures, namely if the bar was struck or scraped with either a soft or a hard object, like the felt or wooden side of the mallet respectively. A hit with a soft object has a lower brightness than a hard object. Fifthly and lastly, the onset information from both microphones is used to determine the location of a hit gesture, using a Time Difference of Arrival (TDOA) method similar as (Novello & Raijekoff, 2015). Because the microphones are spaced apart there will be a time difference between the arrival of a signal to both microphones. The time difference of arrival is calculated as the time difference in samples between the moments of onset for both microphones. This difference gives an approximation of where the bar was struck along its length at the moment of a hit gesture. This output parameter is chosen to obtain a parameter similar to the plucking position of the sound synthesis.

	1	**2**	**3**	**4**	**5**	**6**
Interface Parameter	Contact microphone signals (2x)	Envelope Follower	Onset	Brightness	Gesture type: mallet hard of soft side	Playing position
Type	Sound	Control				
	Direct	Indirect (signal analysis)				
	Continuous	Continuous	Discrete	Continuous in value Discrete in time	Discrete	Continuous in value Discrete in time

Table 2: BAR - STRING output parameters

6.1.2. Mapping Experiments

The following section describes the mapping experiments with the BAR - STRING interface. During the experiments we observe that two general mapping approaches can be made for multiple parameters, namely to either make a continuous or a discrete mapping in time. Both relate to the two main gestures, respectively hitting and scraping the bar. This means that a parameter can either only change at the moment of a hit gesture, or continuously change during a scraping gesture. This allows for different kinds of relations and playing techniques. This will be further explained for each sound synthesis parameter.

In the first experiment we try different approaches for the string input. As mentioned, one mapping approach is to use the two contact microphone signals directly as the inputs into the string. In this way all the inherent qualities of the input, such the type of gesture, its loudness and timbre are directly audible in the output of the string model without explicitly making use of analysis or parameter mapping. In another experiment we try to reproduce the signals typically used for plucked string synthesis. We use the envelope follower on the contact microphones to control the amplitude of a pink noise source. Depending on the decay of the envelope follower, it is possible to have the string sound for a longer time than when using the direct microphones signals. Using this mapping, the timbre of the contact microphone signals is not included. We try to include timbre in the following experiment by lowpass filtering the noise source. This can be done discreetly in time, by setting the lowpass filter frequency according to the brightness during an onset. It can also be done continuously, by using the envelope follower to control the lowpass frequency, allowing to also control the brightness during a scraping gesture. The latter mapping is shown in diagram 1. Both input mapping approaches have their own quality. Using the contact microphone signals results in a very direct relation between the player's input and the sounding result. The synthesized noise source is an abstraction of the microphone signal and therefore it does not reproduce it completely. However, it allows for other timbres and different ways of playing. For example, if the decay of the envelope follower is long, the string can still sound even when the player has stopped hitting the bar.

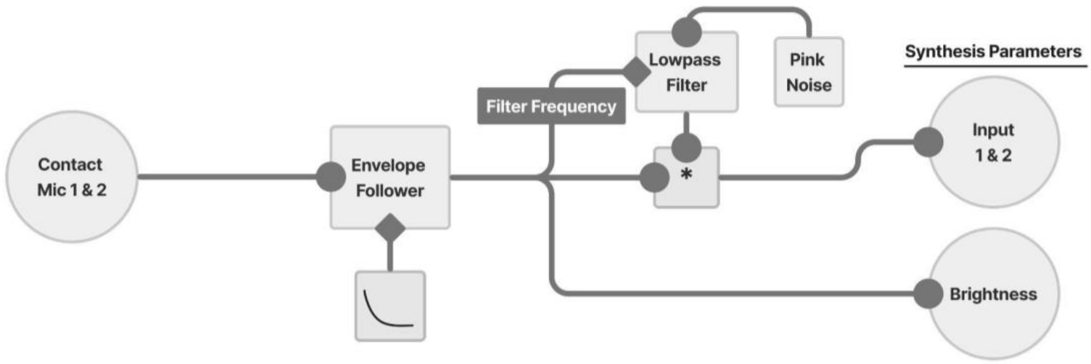

Diagram 1: BAR - STRING Input and brightness mapping

The input lowpass filter on the noise source has a similar effect as the brightness parameter of the string. Therefore, we decide to also map the envelope follower to this brightness parameter (shown in diagram 1), meaning that the brightness is coupled to the loudness of the contact microphone signals. If the decay of the envelope follower is very short, it is only possible to play very short percussive sounds, because the brightness of the string also affects its decay. If it is longer, it is possible to play longer sounding tones. In this way, the amplitude envelope continuously controls both the brightness and the decay. An expressive feature of this mapping is that it is possible to control the brightness during a scraping gesture. Directly and continuously controlling the brightness by the speed and force of the gesture feels very direct and intuitive. In another experiment we map the string brightness discreetly in time, by mapping it to the brightness analysis of the contact microphones. In this way it is possible to set the brightness during a hit gesture, and then continue to play with this brightness during a scraping gesture.

As mentioned, the brightness parameter of the string affects the decay time of the string. And because we map the brightness to loudness, the loudness is coupled both to brightness and decay. In our experiments with mapping the decay parameter, we experiment with either enforcing or partially reverse this coupling. To enforce the coupling, we map the brightness of the hit to the decay, meaning that a hit with the hard side of the mallet results in a longer decay than a hit with the soft side. Inherently, a loud hit contains more high frequency information, so this is also coupled to loudness. We make the mapping range large, so that it is possible to play very long, loud and bright tones, and very short, soft and lowpass-filtered tones. To partially reverse the coupling, we first classify either a soft-side or hard-side mallet hit using a threshold on the hit brightness. We then reverse the loudness and brightness mapping for a hard-side hit, meaning that the decay becomes shorter the louder you hit with the hard side of the mallet. This mapping allows for the possibility to play loud and bright notes, but with a fast decay. This is not possible in the other mapping in which loudness, brightness and decay are coupled to each other. This mapping is shown in diagram 2.

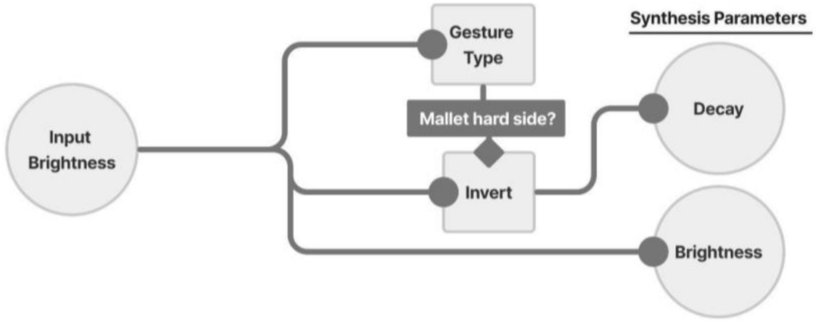

Diagram 2: Inverse decay and brightness mapping

Next, we experiment with mapping the pitch of the string. One approach that we find successful is to interpret the playing position of a hit as pitch information, with the left to right position corresponding to a low to high pitch. This mapping is based on the analogous spatial pitch relationship of instruments like the piano or vibraphone. Because this is a well-established pitch mapping it feels quite intuitive. The TDOA method has a spatial resolution to discern between roughly 8 different positions, allowing to play 8 different pitches. The pitches are mapped to a scale in one octave. We use a major scale, but any scale can be used. Because the playing position varies discreetly at the moment of an onset, the pitch can also only vary at that time. It is in theory also possible to map pitch in a continuous way to create glissandi and pitch variations during a scraping gesture. However, this is not possible to achieve with the Time Difference of Arrival method, because it specifically uses onset information. Another method, such as calculating the maximum value of the cross-correlation signal could be a useful alternative method.

In the final experiment, we map the pluck position and pickup positions. Since the interface has a location parameter, it is intuitive to make a direct mapping between these. We map the pluck position to the horizontal playing position on the bar. Additionally, we also do this with the pickup positions, such that one pickup is always positioned at the pluck position while the other is mirrored, meaning it is positioned at the other half of the string at the same distance from the end. This reinforces the effect of the pluck position on the timbre. This mapping couples timbre and pitch to each other. Similarly to pitch, it would be interesting to be able to vary the pluck position continuously. However, this is not possible given the affordances of the interface and the method used in determining the playing position.

6.2. Interface 2: ROPE - STRING

6.2.1. Output Parameters and Signal Analysis

The physical interaction with the ROPE - STRING interface, plucking and moving it, and observing its response feels very direct and intuitive. The interface has a clear, string-like physical behavior. Therefore, in deciding the output parameters and possible mapping approaches, we focus on closely representing this physical behavior of the interface, in order to make a close coupling between physical motion and sound. We note three key motions and affordances of the interface that we intend to represent in the output parameters of the interface. They are firstly the response of the rope to a fast plucking motion, secondly the string-like vibration of the rope, and thirdly the manual movement of the rope by the player. The motion of the rope is captured by the motion sensors, which are the direct output parameters of the interface. We regard these parameters as part of the sound production of the interface. We obtain additional parameters from analysis of the motion sensor data. These parameters are chosen to provide information on how and when the rope is played. We regard these derived parameters as the controller part of the interface. All the parameters of the interface are now further described.

The direct output parameters of the ROPE - STRING interface consist of the magnitude of the gyroscope and the accelerometer and the orientation of the rope. The magnitude of the gyroscope consists of the sum of the absolute values of the angular velocity in three dimensions. The magnitude of the accelerometer consists of the sum of the absolute values of the acceleration in three dimensions. The orientation of the rope consists of rotations around the three axes, namely the x-axis (pitch), the y-axis (roll) and the z-axis (yaw). Although the rotations are calculated from integration of the gyroscope and fusion with the accelerometer, and therefore are technically not direct output parameters, we do regard them as such because they are an integral part of the output of the motion sensors. For the angle calculations we make use of an MPU6050 library by (Fetick, 2020/2023). From analysis of the sensor outputs we obtain four additional parameters. They are summarized in table 3 below.

Firstly, we use an amplitude-based detection method on the gyroscope magnitude to detect the exact moment the rope is struck: the onset, and the moment the rope is not being played and has decayed in amplitude by a certain amount: the offset. We use this information to conditionally control other parameters. We do this in order to achieve what we define as ‘parameter uncoupling’. This will be explained further in the mapping section. The second parameter is the loudness of the onset event. It is calculated from the gyroscope magnitude at the moment of the onset. It therefore can gradually vary in value, but only discreetly in time. The third parameter consists of an envelope follower on the accelerometer magnitude. The envelope follower has a fixed fast attack and a slow, variable decay, so that it follows both the motion of a fast pluck and the general decay of the rope afterwards. The fourth parameter consists of the horizontal playing position on the rope. This position is calculated from the difference of the pitch angles. Holding the string down or up from a certain position along the string creates an angle between this position and its ends. Because the motion sensors are spaced apart they register different angles depending on the position. The difference of the angles therefore gives an approximation of the playing position. Furthermore, this position information is continuous, meaning it can be gradually changed over time by for example sliding across the rope.

	1	2	3	4	5	6
Interface Parameter	Gyroscope & accelerometer magnitude	Orientation (x, y, z)	Onset & Offset	Onset Loudness	Envelope Follower	Playing Position
Type	Sound	Sound & Control	Control			
	Direct		Indirect (signal analysis)			
	Continuous		Discrete	Continuous in value Discrete in time	Continuous	Continuous

Table 3: ROPE - STRING output parameters

6.2.2. Mapping Experiments

The following section describes the mapping experiments with the ROPE - STRING interface. We discuss different mapping experiments per synthesis parameter. Diagram 3 shows the main mapping approach for all parameters together.

First, we make a general observation from the experiments regarding parameter coupling and the separation between signal and control parameters. Regarding coupling, the data from the motion sensor is already coupled together. A change in the sensor's orientation can cause a change in both the gyroscope and accelerometer, in one or more dimensions. It is very difficult to change one of these dimensions without affecting the others. This coupling is also reflected in data derived from the sensors. Since all parameters of the interface are derived from the sensors, they therefore are all inherently coupled together. In other words, there is no clear separation between sound production and control. For instance, in one mapping experiment we map the horizontal playing position to pitch. This creates the possibility for continuous changes in pitch, which is a difference in affordance between the BAR - STRING interface, which does not allow for continuous pitch changes. However, plucking or moving the rope in any way in order to create sound inevitably causes a change in the measured angles, and therefore a change in the measured horizontal position. Particularly a plucking motion creates a large change. This results in a pitch change every time the rope is struck. This can be either seen as an undesirable effect, or rather a distinct quality of the interface. Either way, we find it interesting that the interface causes us to want to partially separate parameters again, since the initial goal was to create an interface in which the sound production and control part is tightly coupled. We call this partial parameter separation ‘parameter uncoupling’. An example of this is the use of the onset and offset information to conditionally control the parameters derived from angle measurements. In the mapping experiment as depicted in diagram 3 we directly map the horizontal playing position to the pluck and pick up positions. The playing position can continuously control these parameters until an onset occurs. The position is then sampled and held as depicted by the gate operator until the offset has occurred, after which it can continuously change again. The result is that the onset event is separated from the position measurement, which would otherwise cause a large fluctuation in value, and a potential undesirable change in pluck and pickup position.

The further mapping experiments are now described for each synthesis parameter. Firstly, we discuss the input parameter, for which we perform two different experiments, which relate to the two main gestures of the interface, namely plucking and pulling. In the first experiment we try to closely connect the input parameter to the motion of the rope. We use either the gyroscope or accelerometer magnitudes as direct inputs to the string. We add a slight amount of noise to increase the amount of high frequency content. Because we use the data from the motion sensor directly as input, the plucked string directly responds to every movement from the player. And because the motion sensors are very sensitive, they pick up both very small motions and very fast and bigger motions. Opposite to inputting energy, the player can also dampen the sound by dampening the rope. This creates a tight relation between the player's input, the rope’s movement, and the sounding output, resulting in direct and intuitive interaction. Using the rope’s movement as input requires constant input by the player. In the second experiment we also use the position of the rope, by using the angles of rotation around the x axis (pitch) to control the volume of a noise source. This makes it possible to pull and hold the rope up or down to produce a sound indefinitely, allowing for an ‘infinite sustain’. It also allows for slow changes in amplitude rather than only fast changes caused by a plucking motion. It is also possible to have both mappings at the same time, so we can produce sound by both holding the rope down and plucking it. In total, the input parameter consists of the sum of the accelerometer magnitude and the rotations around the x and z axes, multiplying each with an amount of noise. This is depicted in diagram 3.

For the pluck and pick up positions we make one mapping. We directly map the horizontal playing position to both the pluck and pick up positions. The pickup positions are mirrored to enhance both the timbral effect of the playing position in the same way as in the BAR - STRING interface. Only now the position can vary continuously rather than discreetly at an onset event. The horizontal playing position is gated by the onset event as previously mentioned.

For the pitch parameter of the string, we make three different mapping experiments. We already described the first approach, in which we map the horizontal playing position to pitch directly, without quantization or gating. It is therefore mapped both continuously in time and in value. The pitch is logarithmically scaled such that the position from left to right corresponds to a pitch from low to high. This mapping creates a very direct relation and intuitive between pitch and the position and movement of the rope. For example, the vibration of the rope is directly heard as a vibrato. However, as mentioned, it is hard to have precise and stable control over pitch, separated from movement, because the position calculation is inherently coupled to the moment of the rope.
In the second mapping experiment we try to create more precise and stable control over pitch, by using a mapping that is both discrete in both value and time. We use the same position to pitch mapping as before, but we gate the pitch so that it can only change at the moment of an onset. Otherwise the pitch remains constant. We also quantize the position to a pitch scale. We choose a harmonic scale, but any scale can be used. This approach in theory does allow for more stable and precise control over pitch, because we partially separate the pitch from the movement of the rope. However, the onset detection does not always work: sometimes very soft plucking motions are not detected, or there are false detections rapidly after an onset. But more importantly, the position measurement works best if the player first moves the string up or down at a certain position. Measuring the position from only a fast-plucking motion is not reliable, because the fast movement creates a large fluctuation in measurement.

The resulting pitch is often incorrect if the player is plucking without additionally holding the rope up or down. Therefore this mapping in combination with the used position measurement does not allow for stable and precise control over pitch. In the third experiment we decide to keep the base pitch constant, but to allow the player to influence pitch only slightly by moving the rope left and right. We use the amount of change in the yaw angles to offset the pitch a small amount. We use the change in rotation instead of the rotation itself, because the rotation around the z-axis is very sensitive to drift. Moving the rope left to right creates a vibrato effect. This can be with the hand that is also used to pluck the rope, or with the other hand. The amount of vibrato can be controlled by moving further or closer to either one of the motion sensors. Moving close to the sensors creates a larger vibrato. This mapping allows for only a very limited control over pitch, but it does feel quite direct and intuitive because the mapping itself is very direct. It is also less sensitive to measurement errors, because it does not use the position measurement. We do use the gating mechanism, so that the pitch is not affected by plucking motion. Diagram 3 shows both the second and third pitch mapping approaches together.

For the lowpass filter we make experiments that are similar to the input parameter mappings, shown in diagram 3. The goal is to closely reflect the motion of the rope in changes in the filter frequency. In making these mappings, we decide to keep the decay parameter of the string constant, because the filter frequency also influences the decay. We combine the direct output of the accelerometer and the envelope follower on the accelerometer to reflect the oscillatory motion of the rope, a fast pluck and the overall decay of the rope. The oscillatory motion opens and closes the filter like an LFO, and the envelope follower opens and closes the filter based on fast plucking motions and the slow decay of the rope. Additionally, we use the loudness of the onset event to control the decay time of the envelope follower, so that a louder onset results in a longer decay time. This mapping, in combination with the input parameter mapping creates a coupling between loudness, brightness and decay, just as in the BAR - STRING interface. Because this coupling is inherently tied to the physical motion of the rope, we believe this mapping is more intuitive and direct than for instance an opposite mapping, in which plucking the string with more energy results in a lower brightness and shorter decay. Similar to the input parameter, we also map the brightness to pull gestures. We map the absolute value of the rotation around the x-axis to the filter frequency. In combination with the input mapping, pulling the string up or down controls the volume and brightness of a noise source, and the plucking position. Interestingly, because of this combined mapping, it is possible to obtain not a string-like, but a flute-like sound by holding the rope slightly up or down from the center. You hear a noisy, low-pass filtered tone, which only contains uneven harmonics because the even harmonics have an antinode at the center. This creates a square wave. Together these mappings give the impression of a noisy flute-like tone. This is an interesting timbral possibility of the interface. This is not possible in the bar interface, because it does not afford to create and finely control such a sustained sound.

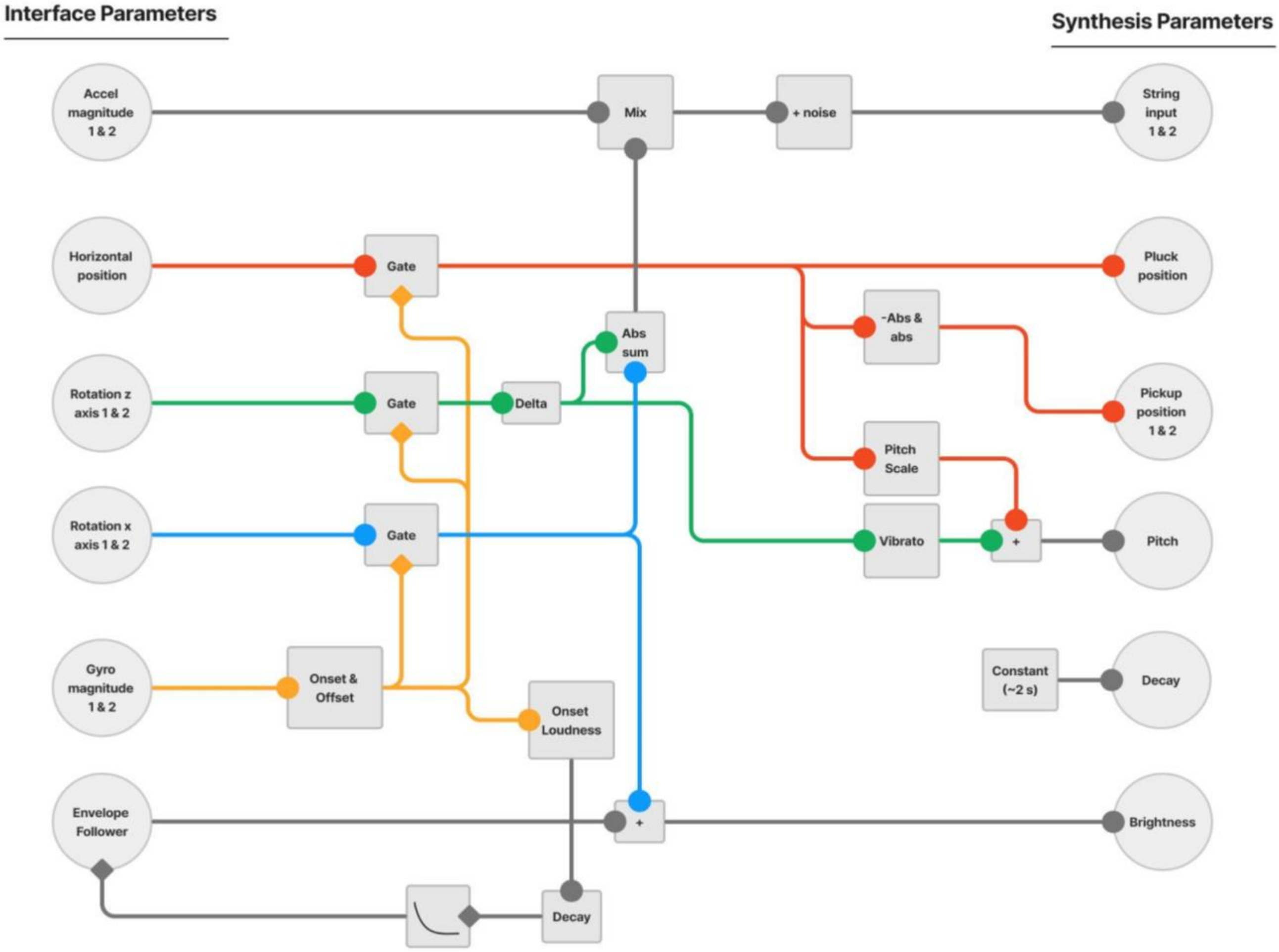

Diagram 3: Main mapping approach of the ROPE - STRING interface

6.3. Interface 3: STRING - STRING

6.3.1. Output Parameters and Signal Analysis

The STRING - STRING interface is very similar to the ROPE - STRING interface. It therefore has similar output parameters, which are chosen to closely represent the physical behavior of the interface, in order to make a close coupling between physical motion and sound. There are also differences between the interfaces. Firstly, the output parameters are partly separated between sensors. We regard the guitar pickups as being part of the sound production of the interface and the hall sensors as being more a part of the control part of the interface. Although this separation is intentionally not very clear, the hall sensors are intended to mostly capture slow string movements and gesture information, while the pickups are intended to capture faster movements of the string. Because we partially separate this on the sensors, the STRING - STRING deals differently with the coupling of signal and control data as the ROPE - STRING interface. The second difference is that the ROPE - STRING uses an actual physical string that has different physical properties as the rope. The main difference is that it has an actual pitch, at either sub-audio rate range or in the low frequency range. The fundamental frequency is roughly in the range of 10 to 100 Hz.

We now describe the output parameters of the interface. The direct output parameters consist of the two guitar pickups and the two hall sensors. From analysis of these outputs we obtain three indirect parameters that we use as control parameters. They are summarized in table 4 below. These parameters are similar to the parameters of the ROPE-STRING interface, except for the pitch parameter. The first parameter derived from analysis is an envelope follower on the two pickup signals. It has a fast attack and a long decay, to quickly follow fast plucking motions and to follow the slow decay of the physical string. The second parameter is the horizontal playing position, calculated using the signal difference between the hall sensors. The third parameter is the pitch of the physical string. We analyze the pitch using an FFT-based method, using the Yin algorithm that is part of the Fluid Corpus Manipulation Toolbox (Tremblay et al., 2022).

<table>
<tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr>
<tr><td>Interface Parameter</td><td>Pickup 1 & 2</td><td>Hall sensor 1 & 2</td><td>Pickup Envelope Follower</td><td>Playing position</td><td>Pitch</td></tr>
<tr><td rowspan="3">Type</td><td>Sound</td><td>Sound & Control</td><td colspan="3">Control</td></tr>
<tr><td colspan="2">Direct</td><td colspan="3">Indirect (signal analysis)</td></tr>
<tr><td colspan="5">Continuous</td></tr>
</table>

Table 4: STRING - STRING output parameters

6.3.2. Mapping Experiments

In our mapping experiments, we found that the physical design and the setup of the STRING - STRING interface has a large impact on the possible ways of playing and types of mappings. This mostly relates to the tuning of the physical string. If the string is tuned to a very low, sub audio frequency, it can be described as vibrating in terms of rhythm rather than vibrating with a definite pitch. In this way, it behaves, just as the rope interface, like a complex LFO, allowing for slow parameter modulations. Additionally, the string is very easy to play and manipulate because the string tension is low. Conversely, if the string is tuned to an audible pitch, it is harder to pull the rope up and down, making plucking and bowing the dominant ways of playing. In this case, it also becomes apparent that the physical design of the interface influences the pitch of the string. The magnets placed on the string influence the behavior of the string such that it vibrates slightly inharmonically, making the fundamental pitch of the string more unclear. This results in a bell-like sound, similar to the sound of a prepared piano string. This is a behavior that we initially did not intend. It does however provide different and interesting timbral qualities in combination with the sound synthesis. Additionally, we found that while the movable bridge works to change the pitch of the string, it does not allow the player to do so flexibly and fast, making it hard to play a melody like one would on a guitar. This is of course also due to the physical design of the interface in its entirety. These constraints together make the interface more suited to control timbral qualities of the sound synthesis, rather than being a more note-based controller to play melodies. Because of the LFO-like possibilities of the low tuning and the limitations of the higher tuning, we perform our mapping experiments mainly with the string tuned to a low pitch.

In the first experiment, we make a similar mapping approach for the input and brightness parameters as done in the rope interface to see if this indeed gives similar, or different results. We do not however implement a gating method, because the hall sensors are less sensitive to plucking motions than the motion sensors, especially the position measurement that is derived from both. The string input and the brightness are mapped in the same way as in diagram 3. We use the signals from the pickups and the hall sensors as direct input to the plucked string, multiplied with a small amount of noise to add high frequency content. Similarly, we map the filter frequency to the direct pickup signals, the envelope follower on the pickups and the hall sensors. These three signals correspond to the three different types of string motion, respectively the harmonic motion, the overall decay and the manual pulling of the string. Because of these mappings, both the input and brightness are closely coupled to every motion of the string. When making this mapping, we notice a difference with the rope interface. We observe that the slow vibrating motion of the string is more pronounced than in the rope. It vibrates more easily, consistently and it has a longer decay. This motion is heard very clearly as low frequency modulation of the amplitude and the filter frequency. It is also visually very apparent. We try to emphasize this motion more, by compressing the pickup signals and the AC part of the hall sensor signals. In this way we amplify the decay stage of the string which contains the low frequency motion, while keeping the amplitude of the initial motion caused by a pluck within bounds. This has as result that the amplitude and filter modulation is more pronounced for a longer amount of time.

In the experiments with mapping pitch, we first keep the pitch parameter constant, but we tune the pitch of the physical string to a higher, audible pitch. We can change the physical pitch with the hands or with the movable bridge. The goal of this experiment is to observe the resulting sound depending on the relationship between the two pitches, and if it is possible to still hear the physical string through the plucked string model if they are not tuned to the same pitch. The idea is to use the synthesized string as a fixed resonator through which we play the physical string. If the pitches are the same or harmonically related, then we hear this mostly as an increase in amplitude because they share a resonant frequency. In this scenario, feedback can occur because the physical string is increasingly excited by the output from the speakers. If the pitches are not the same, the pitch of the synthesis dominates over the pitch of the physical string, meaning that we mostly hear the synthesis and not the physical string. This is partly due to the fact that the signal from the pickups does not contain very much high frequency information.

In the next pitch mapping experiments, we ask the question if the motion from the string could also be reflected in pitch just as in the input and filter mappings. We do so by setting the string pitch to a constant base frequency, and add to it the signal from one hall sensor, and the pickups through an envelope follower, each multiplied by a certain amount. This has multiple effects. It is now possible to slightly and slowly change the pitch of the string manually by pulling and holding the string, or to rhythmically modulate the pitch by the movement of the string after it has been plucked. The pitch of the physical string and the speed of the envelope follower determines the frequency of the modulation. If the envelope follower is slow, the modulation follows the general decay of the string, creating a decaying modulation without oscillation. The sensor amplification factors determine the modulation amount, making it more or less pronounced. If the pitch of the physical string and amplification are low, then the modulation creates a vibrato effect, just as in the rope interface. If the amplification is large, the envelope follower is fast, and the pitch of the physical string is relatively high (>20 Hz), the vibrato changes into a timbral effect. These mappings are an example of the well-known synthesis technique called Frequency Modulation (FM) synthesis. It is depicted in diagram 4. In FM Synthesis, one signal, the modulator, modulates the frequency of another signal, the carrier, with a certain amount. In the mapping experiment, the modulator is the physical string, and the carrier is the string sound synthesis. The modulation amount is inherent to the decay of the string, multiplied by a constant. It therefore continuously varies. It is also possible to reverse the carrier and modulator. The carrier then becomes the signal from the pickups which is modulated by the plucked string synthesis. However, we mainly experimented with the former. Depending on the frequency relation of the carrier and the modulator, and the modulation amount, different timbres can be created. If the carrier and modulator frequency are harmonically related, then the resulting spectrum is harmonic. If they are not, the resulting timbre is inharmonic. We try to both make a harmonic and inharmonic relation between the carrier and modulator frequency using the pitch analysis. In one experiment we set the frequency of the pitch parameter (the carrier frequency) to a multiple of the physical string pitch (the modulator), giving a harmonic timbre. However, it was hard to obtain a stable pitch from the pitch analysis. The analysis is inherently less precise for low frequencies, requiring a large FFT size, which introduces latency. And as mentioned, the pitch of the physical pitch is affected by the magnets which also reduces the accuracy of the analysis. In other experiments we set the carrier frequency manually and keep it constant, but vary the pitch of the physical string with the hands or with the movable bridge. Using this FM technique we find that we can create a large variety of both harmonic sounds, in which you still hear the sound of the plucked string synthesis, and very different, inharmonic and noisy sounds in which the typical timbre of the plucked string is not clearly heard anymore.

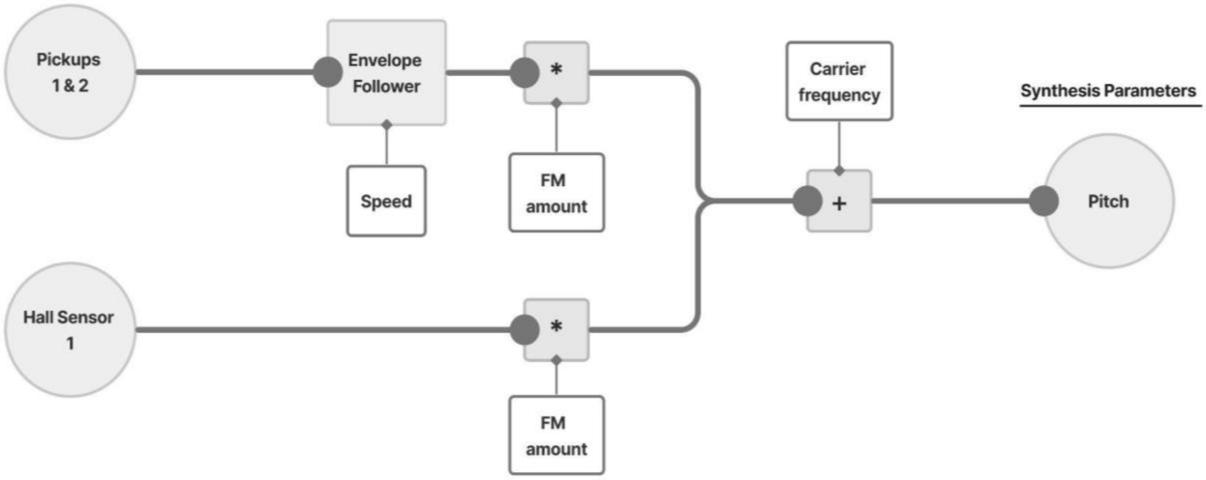

Diagram 4: STRING - STRING FM mapping

7. Discussion

For this research we asked how the relationship between the physical interface and the digital sound synthesis changes when we blur the separation between control and sound production in a digital musical instrument. We made three physical interfaces that vary in their capacities of sound production and control, and in their similarity with the synthesis.
In the following section we discuss the results and observations from the experiments. We compare the interfaces, describing their similarities and differences according to the framework discussed in section four, and how this relates to their relationship with the sound synthesis. We describe for each interface what worked well and what did not. We address principles and observations that could apply to the interface-synthesis relationship more generally. Finally, we propose further improvements and extensions that could be made to our work.

7.1. Interface - Synthesis Relationship

From the experiments we observe the following changes in the relationship between the interfaces and the synthesis. These changes mainly exist between the bar interface on the one hand, and the rope and string interfaces on the other, because the latter two are very similar.
Firstly, an evident, but important difference between the interfaces is a difference in physical affordances and how this affects possible ways of playing and control. The bar interface is mostly a percussive interface. This is reflected in the possible ways of playing and control by the fact that the control parameters, such as pitch and position, are discrete in time; they can only vary during a hit gesture. The rope and string interfaces on the other hand also afford continuous ways of control. This allows for very different types of playing and different sounds, such as for example the described flute-like sound.
Importantly, the difference in affordances is also due to the different sensors. The bar interface is more a sound source than a controller, because its sensors consist of audio signals, which can be used directly as input into the synthesis, but they do not directly contain control information such as playing position. The control parameters are derived indirectly from the audio. The rope and string interfaces do have sensors in which both audio control parameters are directly represented in the sensors, such as respectively the gyroscope magnitude and the angle measurements. We observe from this that in general it can be beneficial to have both audio and control parameters already be physically represented by the sensors, because it can allow for direct measurement, requiring less complex signal analysis. Additionally, it can be beneficial if the audio and control parameters are (partly) represented by different sensors, because it can resolve the mentioned problems of ‘too much’ parameter coupling as mentioned in the rope interface.

This relates to where the parameter coupling is happening: physically, in software, or in both. In the bar and rope interface, parameter coupling is inherently happening physically, because they both have two of the same sensors coupled through the bar or the rope. In the string interface, audio and control parameters are partially separated between the pickup and hall sensors, but they are also intrinsically coupled through the string. A practical example and improvement regarding parameter coupling for the bar interface, would be to add a sensor that measures only the playing position, such as a ribbon sensor placed along the length of the bar. This makes the position measurement independent of the sound production. In addition, a ribbon sensor would make it possible to measure position continuously. Of course, the goal was to create parameter coupling in the first place, but we observe that the lack of separate control parameters in the bar interface can be limiting.

Secondly, we observe a change in the interface-synthesis relationship which is related to the similarity of their parameters. As mentioned, the bar interface is the most separated from the synthesis in this regard. We find that this separation allows for more diverse mapping approaches than in the other two interfaces. This is partly due to a difference in technical complexity in the signal analysis. It was easier to obtain information from the contact microphones on the bar, making the mapping process less prone to error. For example, it was easier to detect clear onsets in the bar interface than in the rope interface, because the bar is not a vibrating mechanical system in the same pronounced way as the rope. But more importantly, the separation made it possible to interpret the parameters of the interface in different ways, while still allowing for an intuitive relationship with the synthesis. For example, mapping the playing position to pitch worked both technically and intuitively better in the bar interface than in the other two, simply because the interface itself does not physically afford something like pitch (such as a resonant frequency), which could conflict with or limit the relation. Conversely, the string interface has a physical, controllable pitch. This initially prompted us to create a direct relation with the pitch of the synthesis, replicating the control of an acoustic string instrument, but it was difficult to perform stable pitch tracking with low latency. And the lack of physical control, like one has on a guitar, limited the amount of control and playability. Another example regarding parameter similarity which worked well in the bar interface, and not the others, were inverse mappings between decay and loudness, such that hitting the bar louder results in a shorter decay. This worked because, again the bar does not have a clear decay parameter itself. When we tried this inverted mapping with the other two interfaces, this did not feel intuitive, because plucking with more energy makes the rope or string move for a longer amount of time, which sets the expectation that we also hear this as an increase in loudness and decay in the sound.

In other words, the tight connection between the interface and the synthesis conditions the possible relationships between the two. More separation between the interface and synthesis gives more freedom to make different relations, including ones that are not physically informed. This is similar to what (Miranda & Wanderley, 2006) describe as a potential limitation of “instrument-like” controllers, as already mentioned in the introduction. They similarly note that the interface conditions the possible relations with the synthesis and that “some controllers are simply not adapted to control certain sound events”. However, they mention this specifically in the case of interfaces closely modeled on existing instruments, that control a sound which tries to closely model another instrument, such as playing a (synthesized) clarinet on a MIDI keyboard. This is only partly the case in our interfaces, since we do not exclusively try to closely model existing instruments.

Therefore, it is important to note that the described effect is not a 'hard rule', or always present for every interface-synthesis relation that we made. As mentioned, the mapping experiments that we made with the string interface using FM synthesis worked intuitively and were expressive in terms of the variability and directness of sound production, although they did not so much make use of a physically informed relation between the pitches of the interface and the synthesis.

7.2. Interface Strengths

We note the following common strengths in the interfaces. Firstly, what worked well in each interface is a clear relation between motion and sound. In the bar interface, the motion is mostly fast and percussive. In the other two interfaces, the motion can be fast and percussive (plucking motions), slow and continuous (pulling motions), and harmonic (the motion from the rope/string). In every interface, energy from these motions is required to produce sound. This energy is used to directly 'excite' the string synthesis. It is then 'used' and 'dissipated', so that we continuously need to inject energy in order to continually produce sound. The concept of injecting energy works well because we work with physical modeling synthesis, in which this concept is inherently represented in the input parameter. But this could also work for other synthesis methods, by for example controlling the amplitude of an oscillator with an envelope follower on the contact microphones in the case of the bar interface.

Secondly, what worked well is parameter coupling. For every interface, we made use of parameter coupling both physically in the interface itself, and in software via mapping. In the interface this relates to the coupling of sensors. In for example the rope interface, a change in one axis of one motion sensor is almost always met with a change in all the axes of both motion sensors. In the mapping we made use of parameter coupling mostly by using interface parameters both as an excitation signal into the synthesis as well as to control synthesis parameters. For example in the rope interface, by using all the outputs of the motion sensors as input into the string, as well as mapping it to brightness. And as mentioned previously, the parameter couplings were successful if they reproduced relations in the physical world, but also other (such as inverse) relations, specifically in the case of the bar interface. This combined use of parameters as excitation source and as control source relates to the interface's capacity of being both a source of sound and a controller. This combination results in a tight coupling of sound generation and control, just as in an acoustic instrument. In the mapping experiments we found that this tight coupling makes each interface feel very direct, responsive and intuitive. For example, the above-described mapping for the rope interface made it feel as if the synthesized string is really present and directly playable in the hands, because every movement of the rope is directly represented in the resulting sound.

Thirdly, what worked well and what contributes to the feeling of directness is the fact the interfaces are very sensitive. With sensitive we mean that the interfaces afford very small and large, slow and fast gestures, and that the sensors have a high dynamic range, capturing these as soft and loud signals, at high sampling rates. The string and rope interfaces use relatively low sample rates compared to standard audio sampling rate, because the data was not captured with an audio interface, but with microcontrollers through a serial connection. However, the sample rates were high enough to capture fast motions such as plucking and the pitch of the vibrations of the rope and string.

7.3. Interface Weaknesses

Two common things did not work well in the interfaces. They are parameter coupling and the effect of adding many analysis layers. Firstly, although we previously mentioned parameter coupling as a strength, it can also be a weakness. Especially in the rope interface we experienced such entanglement of the motion sensor data that it was hard to make a parameter mapping without too much influence of other parameters. This especially concerns the discrete control of pitch. This can be seen as a limitation of the interface. It is mostly an interface to continuously control sound from motion, but it is not suited to discreetly and reliably control pitch. This is also partially the case for the string interface. We think it is interesting that the rope interface made us want to partially uncouple parameters again, by making the described gating mechanism. However, this gating mechanism introduced problems related to the second main weak point of the interfaces. Namely, we observed that adding many analysis layers between the interface and synthesis parameters can hinder their relation. This is due to two factors. Firstly, adding more analysis increases the chance of analysis errors, such as missed or falsely detected onsets and offsets. Secondly and more importantly, adding analysis layers creates a more abstract and interpretative relationship, potentially making it less direct.

For example, for the bar interface, an extremely indirect and abstract mapping would be to map pitch to the time interval between measured onsets, such that hitting the bar twice with a fast tempo corresponds to a high pitch. Such a mapping is inherently less direct than the currently used pitch mappings, because it requires the player to set the pitch indirectly via an abstract and discrete control gesture over some amount of time. Such a control gesture is more common in traditional Human Computer Interaction, such as fast or long press on a touchscreen. But for musical instruments, we note from our experiments that it is desirable to have direct and continuous control, as also mentioned by (Hunt & Kirk, 2000). Similarly, the use of the gating mechanism in the rope interface introduced a layer of abstraction in the interface and made the total interaction less continuous and more discrete: some parameters could only change during a plucking motion while some were held for some time. It required the player to adjust to this behavior, rather than providing a set of inputs that could be continuously changed at any time.

7.4. Extensions and Future Work

Several improvements and extensions are possible for our interfaces. We note the following improvements on the current setup. Most notably, pitch control can be improved on the rope and string interfaces. With pitch control we mean the ability to play discrete pitches (in a scale) and not vibrato, because vibrato already worked quite well. In the experiments for the rope interface we mapped pitch to the measured playing position discreetly at the moment of an onset. However, as described, the measurement was not accurate. A different measurement of the playing position is needed, such as an onset based TDOA method as used in the bar interface. For the string interface, pitch control of the physical string is limited due to the physical design of the interface, namely the influence of the hall sensors on the string and the limitations of the movable bridge. In order to physically control pitch the interface's design could incorporate a neck similar to a violin and sensors that do not influence the behavior of the string itself too much. In the software, the pitch tracking algorithm could be improved in terms of latency. Now the latency is large due to a large FFT size. The FFT size could be made smaller if we increase the lowest possible pitch of the physical string. Another interesting way to improve pitch control is to add a separate interface, such as a MIDI-keyboard or a capacitive touch strip, that solely controls pitch. Our interface then still controls all the other parameters of the sound. This could of course also be interesting for separately controlling other parameters than pitch. As previously mentioned, the bar interface could be improved by making it possible to continuously track the playing position by adding another sensor, such as a ribbon sensor.

An interesting extension on the current interfaces would be polyphony: For the and string interfaces this could involve the addition of strings, both in software and in hardware. For example, the rope interface could have two or more ropes, each connected to one instance of the synthesis algorithm, each with its own set pitch. The interface then resembles a stringed instrument such as a harp, in which the pitch of each string is fixed, while still allowing control over vibrato, and the other parameters of each string individually. This extension would also solve the current problem of pitch control, because such control would simply not be needed anymore. For the bar interface a limited form of polyphony could be added in software in the same way, with one instance of the synthesis algorithm for each pitch. It is then still only possible to play one pitch at the same time, but they can however sound together. To actually be able to play more pitches at the same time, a different sensing mechanism would be needed. Again, something like an array of grounding wires that could work in parallel could be useful.

Another extension could be to expand the possibilities of the plucked string model. We use a relatively simple model with few parameters, but other more complex models exist, modeling additional parameters, such as the coupling of the string to a resonant body, the sympathetic coupling of multiple strings, or the impact of a bow on the string (Karjalainen et al., 1998). It would also be interesting to experiment with completely other types of synthesis. For example, it would be interesting to see in what ways the rope interface could control for example a granular synthesizer, or an additive synthesizer. Different types of synthesis that bear little or no relation to the parameters of the interface could open new possibilities for relationships and mapping strategies that we currently did not think of.

8. References

Baalman, M. (2003). *The STRIMIDILATOR: a String Controlled MIDI-Instrument*.

Baalman, M. (2022). *Composing interactions: An artist's guide to creating expressive interactive systems*. V2_Publishing.

Berdahl, E. J., & Smith, J. O. (2008). *Plucked String Digital Waveguide Model*. 14.

Bongers, B. (2000). *Physical Interfaces in the Electronic Arts*. 30.

Bongers, B. & Sensorband. (1998). An Interview with Sensorband. *Computer Music Journal*, *22*(1), 13. https://doi.org/10.2307/3681041

Fetick, R. J. (2023). *MPU6050_light* [Computer software]. https://github.com/rfetick/MPU6050_light (Original work published 2020)

Hunt, A., & Kirk, R. (2000). *Mapping Strategies for Musical Performance*. 29.

Karjalainen, M., Valimaki, V., & Tolonen, T. (1998). Plucked-String Models: From the Karplus-Strong Algorithm to Digital Waveguides and beyond. *Computer Music Journal*, *22*(3), 17. https://doi.org/10.2307/3681155

Karplus, K., & Strong, A. (1983). Digital Synthesis of Plucked-String and Drum Timbres. *Computer Music Journal*, *7*(2), 43.

Malt, M., & Jourdan, E. (2008). Zsa.Descriptors: A library for real-time descriptors analysis. *5th Sound and Music Computing Conference, Berlin, Germany*, 134–137. https://hal.archives-ouvertes.fr/hal-01580326

Mathews, M. V., & Abbott, C. (1980). The Sequential Drum. *Computer Music Journal*, *4*(4), 45–59. https://doi.org/10.2307/3679465

Miranda, E. R., & Wanderley, M. M. (2006). *New digital musical instruments: Control and interaction beyond the keyboard*. A-R Editions.

Norman, D. A. (1999). Affordance, conventions, and design. *Interactions*, *6*(3), 38–43. https://doi.org/10.1145/301153.301168

Novello, A., & Raijekoff, A. (2015). *A prototype for pitched gestural sonification of surfaces using two contact microphones*.

Sensorband. (n.d.). *Sensorband*. Retrieved February 13, 2023, from https://www.evdh.net/sensorband/soundnet/

Smith, J. O. (1992). Physical Modeling Using Digital Waveguides. *Computer Music Journal*, *16*(4), 74. https://doi.org/10.2307/3680470

Tanaka, A. (2001). *Global String A Musical Instrument for Hybrid Space*. 7.

Tremblay, P. A., Green, O., Roma, G., Bradbury, J., Moore, T., Hart, J., & Harker, A. (2022). *Fluid Corpus Manipulation Toolbox*. https://doi.org/10.5281/zenodo.6834643

Waisvisz, M. (n.d.). *Crackle—Instruments*. Retrieved May 8, 2023, from https://web.archive.org/web/20210615230606/https://www.crackle.org/instruments.html