MultiWave: A Fully Customizable Function Generator for Wearable Electro- and Vibro-Tactile Arrays

Maximilian Schrapel[1] and Michael Rohs[2]

***Abstract*— Vibrotactile arrays require multiple periodic signals to drive their actuators in a coordinated fashion. Electrical muscle stimulation also relies on periodic signals to actuate one or more surface muscle. Typically the signals are produced using microcontrollers. However, as the number of channels becomes large, signal generation gets challenging. To overcome this problem we propose *MultiWave*, an open-source fully customizable FPGA-based function generator that provides 80 independent channels. Its small and compact design even allows the use in wearable applications. A serial protocol is used to configure each channel, either with a predefined or with a custom waveform. The frequency range is 0-512 Hz at a resolution of 10 bits. Higher frequencies are also possible by reducing the resolution and the relative phase at which the output signals can be controlled. We describe *MultiWave* in detail, review the signal-generation requirements of vibrotactile arrays from the literature and outline how they could be implemented with *MultiWave*.**

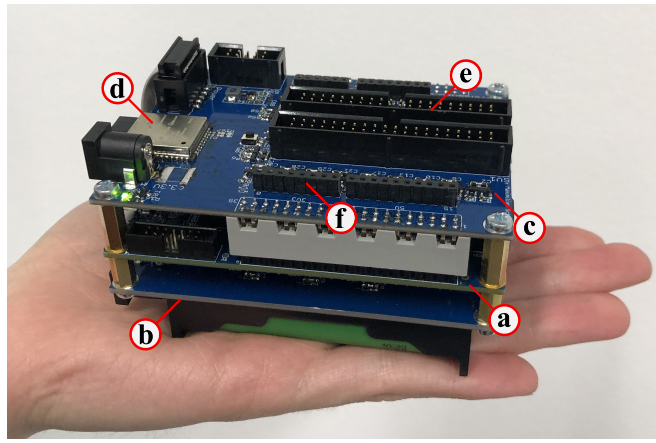

Fig. 1. Multiwave prototype with (a) DE0-Nano development board, (b) 10 A mobile power board, and (c) driver board. The driver board includes (d) a wireless connection, (e) an 80 port actuator socket, and (f) an additional Arduino header. Size (L×W×H): 10.5×6.9×6.5 cm, weight: 330 g.

I. Introduction

Haptic feedback is an important modality in human-computer interaction [1]. By attaching simple vibration motors, more advanced voice coil transducers, or electrical muscle stimulation (EMS) [2] electrodes to a wearable device, the user's attention can effectively be attained or feedback can be given. There are several stimulation parameters that influence the sensation [3]. First, the amplitude can be attenuated to vary effects of the perceived strength. Second, the waveform can cause various effects. For instance, a sine waveform may be modulated by a lower-frequency waveform to achieve a sensation of "roughness" [4]. Third, the basic frequency at which the signal is repeated can be adjusted. There has been a lot of investigation on the effects of these parameters at different positions on the human body for vibrotactile output and for providing force feedback with EMS.

A more complex approach is to use grids with a certain number of actuators at a certain distance between them to provide even richer types of feedback. In addition, waveforms can be output at any repetition rate (fundamental frequency) by several actuators simultaneously or with a specific relative phase offset. If the parameters are to be independently adjustable for each output channel, signal generation can become challenging. With analog circuits waveforms with high signal-to-noise ratios (SNR) can be realized, which, however, require a lot of space on the printed circuit board (PCB) and usually have to be modified when parameters change. To keep flexibility, microcontrollers with digital-to-analog converters or pulse width modulation (PWM) are thus a good alternative. However, these have the disadvantage of being limited to a small number of outputs with low accuracy on each port due to their limited computing power. Specialized integrated circuits, which can generate various signal forms, are also available. However, their possibilities are typically limited to a few outputs and a small number of periodic functions. If a larger number of outputs is required, more hardware components have to be placed on the PCB. In summary, flexible signal generation is a complex problem, especially when grids of actuators are involved.

In this paper we present *MultiWave*, a fully customizable open-source Field Programmable Gate Array (FPGA) function generator for grids of tactile actuators. Through the use of FPGAs the cost and space on the PCB can be minimized. *MultiWave* can generate arbitrary periodic signals at a resolution of 10 bits in the frequency range of 0-512 Hz, which covers the typical frequency of 250 Hz for vibrotactile output [5]. For the available 80 independent channels. The output signals can be attenuated and phase shifted in relation to other output channels. The maximum number of outputs is only limited by the internal resources of the FPGA and the number of GPIO pins. Via a simple serial protocol every output channel is fully configurable using predefined or custom waveforms. This paper describes the design of *MultiWave* in detail, provides a technical evaluation of a prototype implementation, and gives a number of application

[1]Maximilian Schrapel is with the Human-Computer-Interaction Group, Leibniz University Hannover, Hannover, Germany `schrapel@hci.uni-hannover.de`

[2]Michael Rohs is with the Human-Computer-Interaction Group, Leibniz University Hannover, Hannover, Germany `rohs@hci.uni-hannover.de`

examples for wearable electro- and vibrotactile arrays.

II. Related work

As *MultiWave* has been designed to generate signals for both vibrotactile and electrical muscle stimulation, we discuss both technologies separately. In addition, we summarize the requirements for signal generation.

A. Vibro-Tactile Displays

Vibrotactile displays have been used in a wide variety of applications. To name a few, Bark et al. [6] presented *StrokeSleeve* to teach kinesthetic arm movements with eight actuators. Two arm bands were placed around the biceps near the elbow joint and the wrist, respectively. Each band contained four equally spaced actuators (Precision Microdrive eccentric mass motors). An experiment showed that vibrotactile feedback improved the learning of movements compared to visual feedback. Measurements of the actuators showed that the vibration frequency increased approximately linearly with the applied voltage, up to 110 Hz at 3.5 V.

Spelmezan et al. [7] pursued a similar goal of assisting the correction of postures during physical activities. Vibrotactile patterns were created by 14 actuators placed on the shoulders, front and back, and the thighs. Cylindrical motors were pulsed at full intensity with a transmission delay of 39 ms.

There has also been research on combining motion capture and vibrotactile feedback for teaching violin playing. Van der Linden et al. [8] presented a system called *MusicJacket* consisting of seven motors placed on the elbows, arms, and ribs. Their experiments showed that the participants could reliably discriminate between five different sensation levels from *off* to *very strong*.

Kim et al. [9] designed a tactile glove with 20 actuators for immersive multimedia. They added a tactile scene descriptor to MPEG-4 movies. PWM was applied for varying the vibration intensity in eight levels corresponding to the scene descriptor. At 167 Hz vibration frequency there was a latency of 92 ms from sending the signal to reaching full intensity. They also mapped grayscale intensities of a 5×10 pixel video to an equivalent 5×10 actuator tactile display.

In virtual reality environments, one goal of providing haptic feedback is to increase immersion. Bloomfield and Badler [10] used tactile actuators (tactors) in four rings with four actuators in each ring, placed along the right forearm and upper arm. Their intention was to simulate collision awareness and to guide the body into a target pose. By driving the actuators with relays, it was not possible to attenuate the intensity of the signals. The participants did not notice the latency of approximately 68 ms.

Kaul and Rohs [11] used a spherical grid consisting of 24 vibrotactile actuators around the head to guide users towards virtual and real targets by varying intensity according to a guidance algorithm. The mounted actuators were updated at a rate of 75 Hz and operated at a maximum frequency as high as 150 Hz, as equally strong impulses at lower frequencies would have required bigger actuators. They included patterns from 1 to 50 Hz to indicate angular closeness to targets.

Lynette et al. [12] mounted a vibrotactile display consisting of 16 actuators on a waist band to stimulate the skin across the lower back. For use as a navigation aid, patterns with high recognition rates were developed and evaluated. The actuators were pulsed at a frequency of 2 Hz.

Minamizawa et al. [13] presented the "TECHTILE toolkit" for recording and replaying haptic interactions. As this device has only one input and output with amplification it cannot be applied to grids of actuators. A similar toolkit was presented by Israr et al. [14] to play audio signals on haptic actuators.

These toolkits can be used to evoke different sensations on the skin between two actuators such as apparent tactile motions [15], where a single moving stimulus is perceived, or tactile phantom sensations, where a stimulus is perceived at a static position [16]. Israr and Poupyrev [1] presented the *Tactile Brush* algorithm, which creates smooth, two-dimensional tactile motion sequences with a grid of 16 tactors mounted on a chair to enhance immersion while playing games. Their system operated at three frequencies (150, 200, and 270 Hz) and two durations (40 and 160 ms) with a static intensity above the perception threshold.

B. Electro-Tactile Displays

Electrical muscle stimulation is used in medical, fitness, and wellness applications [17]. In the recent years several applications and multichannel EMS toolkits have been developed in HCI.

Kono et al. [18] designed an easy to use four-channel EMS toolkit, which supports variable pulse widths, frequencies, voltages, and time durations for each channel. The toolkit generates monophasic waveforms based on simple integrated timer circuits. The pulse width can be varied between 40 µs and 240 µs. In theory the system can be extended up to 26 channels for frequencies from 50 Hz up to 150 Hz. However, all channels have to be driven at the same frequency.

Zap++ [17] is a wearable 20-channel EMS system for force feedback. Up to 26 channels with an update rate of 100 Hz are available by multiplexing a generated pulse signal with a variable pulse width. Higher frequencies are possible by combining channels. The intensity is controlled via a digital potentiometer.

Knibbe et al. [19] introduced an open-source 60-channel EMS switching board with spatial calibration based on electromyography (EMG). Pulse width, frequency, and amplitude can be controlled through digital potentiometers.

An even more massive approach was presented by Kajimoto et al. [20]. Their *HamsaTouch* system consists of 512 electrodes which are mapped to 512 photo transistors to generate a tactile vision substitution system. By multiplexing a current pulse the sensing and stimulation loop can achieve up to 100 Hz depending on the number of stimulation points.

Lopez et al. [21] used the medically compliant *HASOMED RehaMove* portable EMS stimulator with eight channels and pulse widths from 20 µs to 500 µs for a pen input/output system named *Muscle-Plotter*. Their system uses four of the

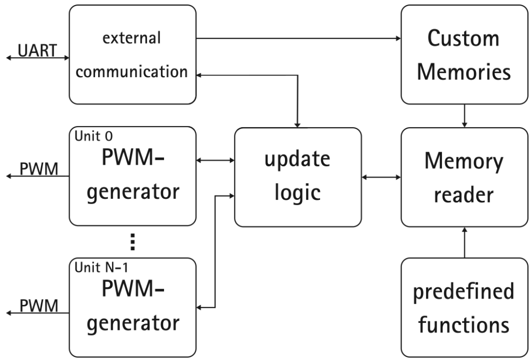

Fig. 2. Simplified block diagram of the FPGA implementation: The PWM generator requests the next PWM step and forwards it to the update logic, which is also responsible for external communication. Then the request is forwarded to the memory reader, which responds with the next PWM value from the corresponding memory element.

eight freely configurable channels. But as the authors mention, the prototype is only "merely portable" [21] because of the hardware size. Meanwhile, the four-channel *HASOMED RehaMove 3* has been introduced, which can be worn on a belt and allows generating arbitrary signal shapes in 16 intensity steps.

Pfeiffer et al. [2] investigated the combination of EMS and vibration feedback for hand pointing [22] and compared both techniques with regard to perceived strength. To enhance immersion, Lopez et al. [23] combined EMS with vibration actuators to render the haptic sensation of hitting and being hit in a virtual environment. The system latency was about 60 ms.

Compared to related work, MultiWave aims to increase the number of available channels, reduce latency, and allow for coordinated operation of actuators through precisely definable phase offsets. MultiWave also aims for untethered mobile use.

III. Hardware Design

FPGAs are integrated circuits that are programmable at the gate level in the hardware description language VHDL. They consist of programmable logic blocks that can be freely connected. The logic blocks contain logic gates and memory elements. In the future the described design could be implemented as an application-specific integrated circuit (ASIC), which is cheaper and more efficient. This section describes the functionality of the FPGA implementation and the expansion board for mobile applications.

The system operates at a clock rate of 100 MHz and therefore can generate a minimum pulse width of 5 ns. Fig. 2 shows a block diagram of the FPGA implementation. All major parameters such as PWM and clock frequency, baud rate, number of functions, and outputs can easily be set via top-level variables before synthesis and programming. The main functional components are described below.

A. Wave Memories

The wave memory units store discrete samples of the output signal. Each memory unit stores 1024 samples and each sample has a resolution of 10 bits. There are five predefined functions: sine, triangle, sawtooth, rectangle, and DC-output. The rectangle and DC signals do not rely on stored samples. The number of available memories of the FPGA defines how many functions can be stored simultaneously. On the DE0-Nano development board up to 393 additional functions are available at the same time.

The memory reader is responsible for reading the desired memory, scaling the stored values according to the set amplitude, and forwarding the values to the PWM generator. To speed up the calculation and to eliminate the most critical path through the logic gates, all divisions have been replaced by invariant integers [24]. Thus the total time for reading, calculating, and sending the values back to the PWM generator can be reduced to 8 clock ticks or 80 ns, respectively.

B. PWM Generators

A PWM generator produces the output signal for a particular channel. The PWM period is 10 ns and the PWM duty cycle is specified in 1024 steps (0 steps = 0% duty cycle = 0 ns on-time; 1023 steps = 100% duty cycle = 10 ns on-time). A PWM generator requests the step count (0-1023) and number of repetitions (0-97656 for 1 Hz) of this setting from the update logic.

Moreover the settings for amplitude, desired output function, and frequency are also stored in this unit. The PWM generator always stores the currently active and the next setting, so as to avoid delays during parameter update requests. When the next parameter pair (step count, repetition count) is requested, the update logic switches the port to the requested memory, computes the parameter pair, and sends it back to the PWM generator.

To generate PWM steps simple counters compare the number of clock ticks to the actual step. Depending on the output frequency the PWM signal with n bits has to be repeated several times.

C. Update Logic

The update logic handles the requests of all PWM generators. When more than one unit asks for new data, each is enabled one after the other in order to avoid concurrent reading of the memories. The update logic unit continuously scans the PWM generators for parameter requests. Hence, when a PWM generator requests the next parameter pair, external changes become effective at this point. Whenever the external communication unit received a valid command and the target unit is about to be processed, the parameters are changed before the request is handed over to the memory reader.

D. Communication

For external communication a UART was chosen because of its reliable and easy to use interface. Baud rates of more than 2 Mbaud are possible for updating the signal configurations, which corresponds to a throughput of 200 kbytes per second. In addition, via USB-UART converters it is easily possible connect the device to a PC. There is a simple

('M', port, 0, function)
A binary 0 corresponds to the target output function, which is followed by one byte specifying the desired output function. There are five predefined functions: 0 = sine, 1 = triangle, 2 = sawtooth, 3 = rectangle, and 4 = DC. Other values can be used for custom memories. Depending on the used FPGA a large number of custom wave memories could be utilized, but for most applications it should be sufficient to limit the number of functions to 256.

('M', port, 1, frequency high, frequency low)
A binary 1 after the target out-port indicates that the frequency is going to be set. Due to the range of 512 Hz two bytes are needed to specify the output frequency.

('M', port, 2, 0..255)
A binary 2 corresponds to the internal amplifier of the signal, which can be adjusted between 0 to 255 within one byte, which is interpreted as percentage value. Thus it is possible to scale the signals out of range to generate different signal forms. For example, a tuned-out triangle produces a trapezoid.

('M', port, 3, phase high, phase low, related port)
A binary 3 specifies the phase of the target output. It can be adjusted in degrees between 0 to 360 with 2 bytes, followed by one byte for the out-port relative to which the phase should be adjusted. The update logic then compares the next requested PWM step of both ports and sets the desired port in relation to the reference port.

('M', port, 4, parameter index 0..3)
Additionally the state of each port can be requested by sending a binary 4 followed by one byte with the requested type. As a result two bytes will be sent back with the desired internal value.

('C', memory index, addr. high, addr. low, data high, data low)
The wave memories can be filled by starting a command with a 'C' and then followed by one byte for the index of the custom wave memory and 2 bytes for the address and 2 bytes for the PWM value. A 9 bit PWM signal refers to a 512×9 bit memory and a 10-bit PWM signal refers to a 1024×10 bit memory. Fully updating an internal memory requires 6 bytes, multiplied by the 1024 or 512 PWM steps. In case of a 9-bit PWM signal the internal memory can be updated at a rate of 520 Hz. For a 10-bit PWM the update rate decreases to 260 Hz via UART at 2MBaud.

TABLE I

COMMANDS OF THE COMMUNICATION PROTOCOL, SPLIT INTO PORT COMMANDS ('M') AND MEMORY COMMANDS ('C').

byte-oriented communication protocol with port commands and memory commands. Port commands begin with an ASCII 'M' and memory commands with an ASCII 'C'. The commands are listed and described in Table I.

E. Expansion Board

For mobile and simple use of *MultiWave* we developed an expansion board, which uses all existing GPIO pins of the DE0-Nano to operate 80 independent actuators. An additional battery board on the bottom side provides a total of 10 A at 3.3 V, of which up to 700 mA are available per output. The three used 3400 mAh LR18650 li-ion batteries can operate all outputs at 100 % intensity and 100 mA current for more than one hour continuously. The ESP32 microcontroller on the expansion board can receive commands via Bluetooth, WiFi, and USB to communicate with the FPGA. The Arduino header can be used for further extension boards or other purposes.

IV. TECHNICAL EVALUATION

The technical evaluation the output signals is conducted with regard to using the function generator in tactile actuator

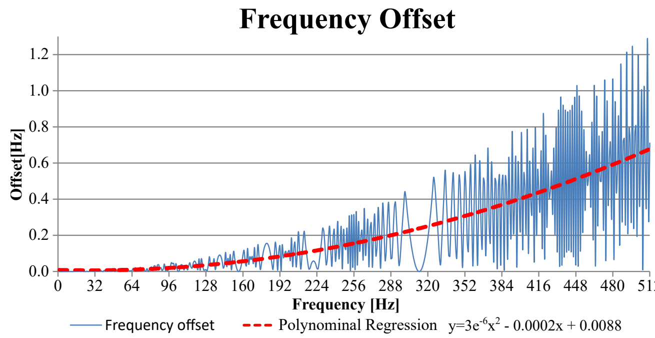

Fig. 3. Hardware-dependent static frequency offset with polynomial regression.

grids. For our tests we used the Altera/Intel DE0-Nano education board with a Cyclone V FPGA and a DE2 board with a Cyclone II FPGA (phased out). Both achieved the same performance on all tests. The configurations of the internal memories are not compatible with the FPGAs of other manufacturers such as Xilinx or Lattice and must be reconfigured accordingly. For portable applications we recommend using the DE0-Nano board because of its small size (68.6×96.0 mm) and low cost (99$). Other terasIC DE-boards, like the DE10-Nano, require a reconfiguration of the GPIO pins.

A. Resource Utilization

Each function generator requires internal hardware resources of the FPGA, which limits the maximum number of outputs. Table II gives an overview of the necessary resources for every PWM generator and custom memory. On the Cyclone V DE0-nano board 80 pins of the existing 314 can be used via the GPIO header. Theoretically, over 200 ports could be implemented based on the available resources, since the 80 function generators use less than 40 % of the available resources.

B. Frequency Offset

Because discrete clock ticks and samples are used to generate the output signals, a static offset occurs at a desired output frequency. This is due to the fact that an output signal is resolved in n steps of the same length, whereby smaller errors at each step are accumulated.

Fig. 3 shows the distribution of the error depending on the frequency for a 10 bit PWM signal. The maximum static error is 0.24 Hz at 250 Hz and 1.29 Hz at 510 Hz.

Resource utilization overview

Unit	Logic Cells (LC)	Additional LC per PWM generator	Memory Bits	Additional Bits per custom Memory
External communication	425	0	0	0
PWM generator	190	190	0	0
Update logic	114	35	3610	0
Memory reader	310	12	49664	10240
Summary	**1039**	**237**	**53274**	**10240**

TABLE II

OVERVIEW OF REQUIRED RESOURCES PER CUSTOM MEMORY UNIT AND PWM GENERATOR.

Lowering the precision of the PWM singal to 9 bit halves the number of steps, the offset is reduced by half as well. The distribution can be approximated by a second-degree polynomial regression.

V. Application scenarios

MultiWave enables the creation of mobile and wearable systems with advanced tactile output and allows evaluating a wide range of research questions. In the following we illustrate several application scenarios.

A. Schematics

In Fig. 4 basic schematics are given to drive different kinds of feedback devices.

Fig. 4a shows an EMS generator, which is divided into three parts. The NE555 (left) generates a periodic rectangle signal which is amplified by a MOSFET. The potentiometer attenuates the voltage, which is in then converted to a sine wave by a low-pass filter and a capacitor as a DC-blocker. A transformer (middle) separates the input voltage from the high voltage (HV), which is additionally multiplied in one or more cascades. A power MOSFET (right) switches the EMS signals through to the electrodes. The digital insulator separates the PWM control signals from the high voltage circuit. The outputs may be connected to the electrodes of a grid. The last part of the circuit has to be repeated depending on the required number of outputs. A disadvantage of this design is that the cascade that generates the high DC voltage also produces ripple noise. This ripple noise may be reduced by additional low-ESR electrolytic capacitors. To generate negative voltages the middle part of the EMS schematic could be repeated. An additional port would be required to switch between positive and negative voltage. The two ports then have to be driven with a 180° phase offset at the same frequency by an H-bridge. An alternative way to generate EMS signals is current sources as used for *HamsaTouch* [20]. To increase the number of outputs, multiplexers could be used, which are controlled by several outputs of *MultiWave*.

The mono audio amplifier (Fig. 4b) can be used for many different output devices such as bone conductors, voice coils, or speakers. At first a low-pass filter eliminates noise generated by the PWM signal. Then the signal is passed though the class D amplifier to the output. Depending on the amplifier and output device the gain must be set to the maximum and can further be attenuated by the PWM signal.

The schematic Fig. 4c can be used for brushed DC motors or stepper motors. In addition, two ports of the FPGA can be used to control the motor in forward and reverse direction. The reference voltage must be set according to the required output current. An additional port of the FPGA to generate a DC voltage can be used for dynamic current settings.

Fig. 4d shows a simple vibration motor driver schematic. The Schottky diode and the capacitor in parallel to the motor can optionally be used to suppress electromagnetic interference. A high pull-down resistor reduces the quiescent current and always keeps the MOSFET in a defined state. The gate resistor limits the output current of the FPGA. Each of the 80 out ports of *MultiWave* uses this circuit.

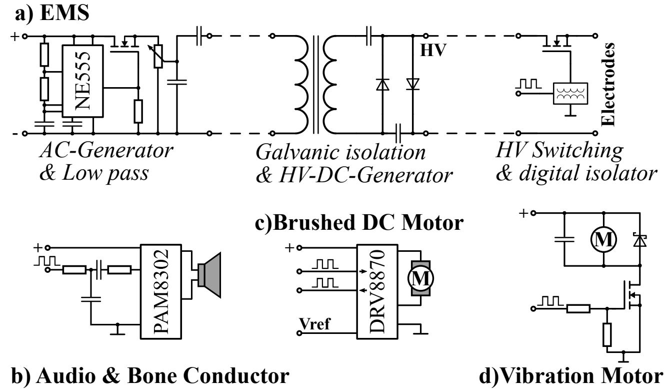

Fig. 4. Example schematics for different applications: (a) EMS high voltage generator with potentiometer for calibrating the maximum output; (b) audio amplifier for bone conductors, voice coils, and speakers; (c) brushed DC and stepper motor driver with forward and reverse control; and (d) vibration motor driver with electromagnetic interference suppression.

B. Replicating and Extending Related Work with MultiWave

MultiWave allows to freely configure the parameters of the generated signals. *MultiWave* can thus be used to easily reproduce and extend some systems described in related work.

1) Vibrotactile Displays: Vibrotactile grids can give feedback on different activities, such as teaching kinesthetic arm movements [6], learning a musical instrument [8], or doing sports like snowboarding [7]. This feedback can help to improve the performance of the user. For outdoor activities or for the use as a navigation aid [12] vibrotactile feedback systems have to be portable. If *MultiWave* is used to control vibration motors, each output can be driven individually by our extension board via Bluetooth or WiFi. In addition, variations of the *Tactile Brush* algorithm [1] in combination with more actuators can be used to investigate whether the error can be further reduced. Patterns or Tactons [4] can be stored in the custom memories of the FPGA and selected via UART from the microcontroller. To switch off the outputs, the amplitude has to be set to zero. In the area of immersive multimedia [9] and virtual reality [11], high-resolution grids can possibly provide an even more immersive experience. To avoid noise from vibration motors, bone conductors or voice coils and sine waves can be applied as shown in Fig. 4. This schematic can also be used to create a low frequency sound system. A limitation is that in contrast to [13] no direct playback of audio can easily be realized. It is possible but requires that a memory unit is continuously written to and read from by the PWM generators.

2) Electrotactile Displays: EMS toolkits that are used for research usually apply time multiplexing to distribute a generated pulse signal to all electrodes [17], [18], [19]. The medically compliant *HASOMED RehaMove* portable EMS stimulator, as used by [21], provides eight independent channels. *MultiWave* can multiplex a signal by connecting outports to the control inputs of the multiplexers and switching a selected signal. Then the control sequence has to be stored in the custom memories. One additional port can then generate the pulse signal. In that case every output requires an additional MOSFET for switching the channels on or off. With n EMS generators all n ports can be freely configured. *HamsaTouch* also uses time multiplexing for the electrodes of their tactile vision system [20]. By using *MultiWave* to generate differently pulsed current signals, other sensations may occur that might, for example, allow depth perception to be generated. If the grid is divided into areas and each one has its own current source, update rates can be increased and it might be possible to create several effects simultaneously. The size of the areas is then limited to the minimum pulse with, i.e., the maximum frequency. A combination of EMS and vibrotactile feedback [2], [22], [23] could be extended to a full-body suit to investigate immersion in virtual environments.

VI. Conclusion

This paper presented a fully customizable, portable, and low cost open source function generator for electro- and vibrotactile arrays. It substantially simplifies the control of such actuator arrays. It was shown that with a small frequency offset all outputs can be individually controlled with a simple external interface. Example schematics for applications involving EMS, audio or bone conductors, and different vibration motors were given. Existing systems from related work can be reproduced with *MultiWave* in a simple way and extended for future research. The technical evaluation shows that the proposed system exhibits a very low delay compared to other systems reported in the literature. *MultiWave* provides guaranteed phase synchronization and custom phase offsets between ports, which is not the case for most systems described in the literature. This allows the investigation of phantom sensation and other effects that rely on coordinating multiple actuators.

MultiWave is available as an open source project (see `https://hci.uni-hannover.de/research/multiwave`), including VHDL code, pre-compiled FPGA files, Arduino source code, schematics, housing for 3D printers, a Python API, and a GUI with additional features such as a pattern designer and video tutorials for an easy start. Moreover, we are planning to build a demonstrator for vibration feedback and use this prototype for our future work in electro- and vibrotactile displays.

References

[1] Ali Israr and Ivan Poupyrev. Tactile brush: Drawing on skin with a tactile grid display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2019–2028, New York, NY, USA, 2011. ACM.

[2] Max Pfeiffer, Stefan Schneegass, Florian Alt, and Michael Rohs. Let me grab this: A comparison of ems and vibration for haptic feedback in free-hand interaction. In *Proceedings of the 5th Augmented Human International Conference*, New York, NY, USA, 2014. ACM.

[3] G. W. Young, D. Murphy, and J. Weeter. Haptics in music: The effects of vibrotactile stimulus in low frequency auditory difference detection tasks. *IEEE Transactions on Haptics*, 10:135–139, 2017.

[4] Lorna M. Brown, Stephen A. Brewster, and Helen C. Purchase. A first investigation into the effectiveness of Tactons. In *Proceedings - 1st Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems; World Haptics Conference, WHC 2005*, pages 167–176. IEEE, 2005.

[5] Gunther Eric (Eric Louis). Skinscape: A tool for composition in the tactile modality. 05 2014.

[6] Karlin Bark, Emily Hyman, Frank Tan, Elizabeth Cha, Steven Jax, Laurel Buxbaum, and Katherine J Kuchenbecker. Effects of vibrotactile feedback on human learning of arm motions. 23, 06 2014.

[7] Daniel Spelmezan, Mareike Jacobs, Anke Hilgers, and Jan Borchers. Tactile motion instructions for physical activities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 2243–2252, New York, NY, USA, 2009. ACM.

[8] J. van der Linden, E. Schoonderwaldt, J. Bird, and R. Johnson. Musicjacket—combining motion capture and vibrotactile feedback to teach violin bowing. *IEEE Transactions on Instrumentation and Measurement*, 60(1):104–113, Jan 2011.

[9] Y. Kim, J. Cha, J. Ryu, and I. Oakley. A tactile glove design and authoring system for immersive multimedia. *IEEE MultiMedia*, 17(3):34–45, July 2010.

[10] A. Bloomfield and N. I. Badler. Virtual training via vibrotactile arrays. *Presence: Teleoper. Virtual Environ.*, 17(2):103–120, April 2008.

[11] Oliver Beren Kaul and Michael Rohs. Haptichead: A spherical vibrotactile grid around the head for 3d guidance in virtual and augmented reality. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '17*, CHI '17, 2017.

[12] Lynette A. Jones, Brett Lockyer, and Erin Piateski. Tactile display and vibrotactile pattern recognition on the torso. *Advanced Robotics*, 20(12):1359–1374, 2006.

[13] K. Minamizawa & Y. Kakehi & M. Nakatani & S. Mihara & S. Tachi. Techtile toolkit: A prototyping tool for design and education of haptic media. In *Proceedings of the 2012 Virtual Reality International Conference*, pages 26:1–26:2, New York, NY, USA, 2012. ACM.

[14] Ali Israr, Siyan Zhao, Kyna McIntosh, Zachary Schwemler, Adam Fritz, John Mars, Job Bedford, Christian Frisson, Ivan Huerta, Maggie Kosek, Babis Koniaris, and Kenny Mitchell. Stereohaptics: A haptic interaction toolkit for tangible virtual experiences. In *ACM SIGGRAPH 2016 Studio*, SIGGRAPH '16, pages 13:1–13:57, New York, NY, USA, 2016. ACM.

[15] Harold E. Burtt. Tactual illusions of movement. *Journal of Experimental Psychology*, 2:371–385, 10 1917.

[16] D. S. Alles. Information transmission by phantom sensations. *IEEE Transactions on Man-Machine Systems*, 11(1):85–91, March 1970.

[17] Tim Duente, Max Pfeiffer, and Michael Rohs. Zap++: A 20-channel electrical muscle stimulation system for fine-grained wearable force feedback. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '17, pages 1:1–1:13, 2017.

[18] Michinari Kono, Yoshio Ishiguro, Takashi Miyaki, and Jun Rekimoto. Design and study of a multi-channel electrical muscle stimulation toolkit for human augmentation. In *Proceedings of the 9th Augmented Human International Conference*, AH '18, pages 11:1–11:8, New York, NY, USA, 2018. ACM.

[19] Jarrod Knibbe, Paul Strohmeier, Sebastian Boring, and Kasper Hornbæk. Automatic calibration of high density electric muscle stimulation. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(3):68:1–68:17, September 2017.

[20] Hiroyuki Kajimoto, Masaki Suzuki, and Yonezo Kanno. Hamsatouch: Tactile vision substitution with smartphone and electro-tactile display. In *Proceedings of the Extended Abstracts of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI EA '14, pages 1273–1278, New York, NY, USA, 2014. ACM.

[21] Pedro Lopes, Doăa Yüksel, François Guimbretière, and Patrick Baudisch. Muscle-plotter: An interactive system based on electrical muscle stimulation that produces spatial output. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pages 207–217, New York, NY, USA, 2016. ACM.

[22] M. Pfeiffer and W. Stuerzlinger. 3d virtual hand pointing with ems and vibration feedback. In *2015 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 117–120, March 2015.

[23] Pedro Lopes, Alexandra Ion, and Patrick Baudisch. Impacto: Simulating physical impact by combining tactile stimulation with electrical muscle stimulation. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pages 11–19, New York, NY, USA, 2015. ACM.

[24] Torbjörn Granlund and Peter L. Montgomery. Division by invariant integers using multiplication. In *Proceedings of the ACM SIGPLAN 1994 Conference on Programming Language Design and Implementation*, PLDI '94, pages 61–72, New York, NY, USA, 1994. ACM.