Master Neuroengineering
Department of Electrical and Computer Engineering
Technical University of Munich

I can feel you sign

Bidirectional communication system between visually impaired and non-verbal individuals

Kunal Aggarwal, Katja Frey, Alexandra Samoylova, Oscar Soto Rivera, and Maria Zeller

Master Neuroengineering, Department of Electrical and Computer Engineering, Technical University of Munich

December 3, 2023

Abstract — Our bidirectional communication system addresses the challenges faced by visually impaired and non-verbal individuals, offering promising advancements in inclusive digital interactions. Through modular design and the integration of subsystems, our system enables successful unidirectional and bidirectional communication pathways, showcasing the potential for improved societal inclusion for individuals with disabilities. Although there remain some constraints and possibilities for improvement, the system's capacity for facilitating communication is already an important step in the direction of a more inclusive future.

1 Introduction

In recent years, there has been a notable surge in human-machine interface technologies designed to better integrate individuals with disabilities into society. From basic hand-operated wheelchairs to cutting-edge thought-controlled models, there has been remarkable progress in engineering solutions that prioritize human needs.

Among disabilities, visually impaired (blind) and non-verbal (mute and deaf) individuals confront significant barriers to societal inclusion. These challenges not only impact their professional lives but also impede social interactions, particularly in today's digital age. Visually impaired individuals heavily rely on Braille script, while non-verbal individuals communicate through hand gestures. Although effective in face-to-face communication, these methods are poorly integrated into digital systems for online networking.

As one possible solution, Reda et al. (2018), developed SVBiComm, a bidirectional desktop human-computer interface application facilitating communication between non-verbal individuals (via video) and visually impaired individuals (via speech). However, this solution has limitations. Firstly, video-to-speech conversion involves extensive video pre-processing, slowing down the system. Secondly, the system requires a substantial amount of training data for videos and voices to ensure accurate recognition.

To address these limitations, we have created an alternative system connecting video input to vibrotactile output and vice versa. Our system avoids heavy dataset training by utilizing the pre-built MediaPipe hand gesture recognition library and an array of vibrotactile motors (VTM), translating input numbers into Braille patterns. By incorporating an array of force-sensitive resistors (FSR), using which the user can send Braille-encoded numbers, we made our system suitable for bidirectional communication. This prototype aims to establish a user-friendly online communication system, bridging the gap between visually challenged and non-verbal individuals through video recognition and vibrotactile feedback.

2 Methods

Our bidirectional communication system comprises three modules: The MediaPipe subsystem to detect signed numbers, accompanied by VTMs that provide haptic feedback regarding the signed number, as well as FSRs that transmit a number back to the MediaPipe module. Accordingly, one individual engages solely with the MediaPipe subsystem whilst their partner interacts with both the VTMs and FSRs (Figure 1). Communication is achieved via UDP in both directions, i.e., from MediaPipe to the VTMs and from the FSRs back to MediaPipe.

2.1 Subsystem MediaPipe

To decode numbers signed by a non-verbal person, we imported the MediaPipe framework into Python. Google created the MediaPipe framework to provide developers with pre-made, modifiable AI solutions. With the help of open-source pre-trained models, developers can design applications which allow for the real-time observation of audio or video data.

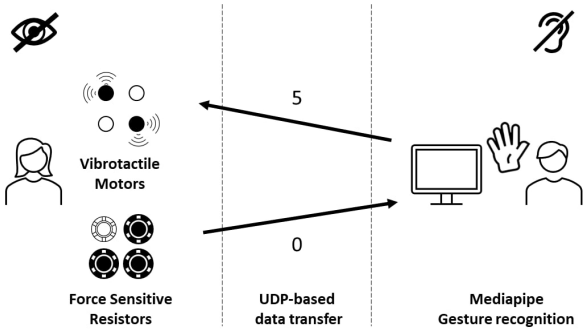


Figure 1 Schematic of biderectional communication system usage by a visually-impaired (left) and a non-verbal (right) person. The system comprises three modules: an array of vibrotactile motors (number reception), an array of force sensitive resistors (number sending) and MediaPipe gesture recognition module (number reception and sending on the non-verbal user side). Data transfer between subsystems happens through UDP protocol.

In our case, we used the framework to extract the landmarks / coordinates of both hands for two dimensions. For the hands, landmarks include the tips of the fingers, palms, and knuckles. In total, MediaPipe extracts 21 landmarks for each hand. These specified locations can be utilized to monitor hand movements. Standard identification values for particular landmarks are automatically included in MediaPipe such as 0 for the wrist, 8 for the tip of the index finger, and 4 for the tip of the thumb.

We developed a module that can determine if a finger is extended and automatically detects the total number of extended fingers at any time by using the tip IDs of the fingers which are provided by MediaPipe. To make the system more robust against wrong detections, we introduced a sampling counter. This counter makes sure, that we collect a certain number of samples before the final number is sent. From the array which contains all samples, the most frequent number is selected.

2.2 Subsystem Vibrotactile Motors

To facilitate data transmission through VTMs, we have assembled a system on a breadboard that includes one ESP32 microcontroller (programmed using the Arduino software IDE) and four VTMs arranged in a square grid. Each motor is controlled by a circuit consisting of a 1 µF ceramic capacitor, a 1N4001 diode, and a 1 KΩ. resistor, and a 2N2222 NPN transistor. This setup not only facilitates switching but also effectively isolates the low-power control logic from the high-power motor driving segment of the circuit. This isolation is achieved by utilizing the transistor as an electronic switch, controlled by a signal from a digital pin of the microcontroller. When activated, the transistor connects the power source to the motor, enabling operation in binary mode. The resistor limits the current from the ESP32 to the transistor's base, thereby protecting the microcontroller. Concurrently, the diode, being reverse-biased and parallel-connected to the motor, offers protection against voltage spikes. Additionally, the capacitor, situated parallel to the motor, serves to smooth the voltage supply, thereby mitigating circuit noise and ensuring a more stable operation.

The software implemented into the microcontroller is designed to assign a specific pin to each motor, configuring these pins as digital outputs. These outputs can switch between two states: '0' and '1'. In analogue terms, a '1' represents a voltage level of 3.3 V, while '0' corresponds to ground (GND). This setup effectively facilitates the control of the transistor base, allowing for precise management every time the motor activation is triggered by the software's control logic.

2.3 Subsystem Force-Sensitive Resistors

For data transmission via FSR, we used a breadboard, one ESP32 microcontroller (programmed using the Arduino software IDE), four FSRs placed in a square grid, four 10 KΩ resistors, connecting wires, a USB cable, and a laptop. The configuration of each FSR forms a voltage divider: one pin of the FSR is connected to the 3 V pin on the microcontroller for power, while the other pin connects to the microcontroller's GND pin through a 10 KΩ resistor, in series. To record the FSR reading, the output from the second FSR pin is directly connected to an analogue input pin of the microcontroller. The microcontroller is connected to a laptop via a USB cable. The voltage output from the voltage divider is directed into the Analog-to-Digital Converter (ADC) of the microcontroller. The onboard program in the ESP32 then evaluates whether the incoming analog values surpass a pre-set threshold, a condition interpreted as the user applying pressure to the sensor. Furthermore, this program also plays a crucial role in translating these sensory inputs into the corresponding Braille code, effectively decoding the user's input into recognizable numerical data. After

the decoding step, the final number is transmitted to a laptop via serial connection.

2.4 Braille Representation

Each VTM / FSR in our system represents an individual dot from the Braille system. As demonstrated in Figure 2, this approach allows for an effective translation of numeric data using a simple 4-point grid.

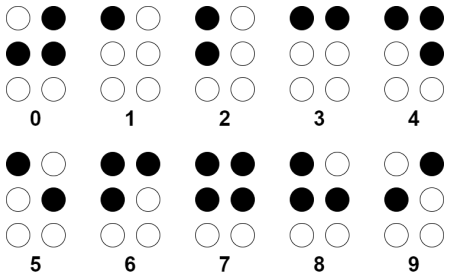


Figure 2 Number representation in Braille code. Notably, the arrangement demonstrates that just four VTMs / FSRs can effectively represent each dot, allowing for the accurate depiction of numbers as per Braille standards.

2.5 Bidirectional communication via Finite State Machine

To control the information flow between the subsystems, we suggest using a finite state machine (FSM). The implementation of this FSM, as illustrated in Figure 3, serves as an exemplar showcasing the potential structure for such flow control mechanisms. To effectively implement this FSM, a clear distinction is made between two communication subsystems: MediaPipe and VTM-FSR, each with an FSM integrated into its control logic. In the MediaPipe FSM, the "Read Input" state serves as the starting point, with the MediaPipe subsystem actively scanning for participant-initiated communication signals. Concurrently, the VTM-FSR subsystem enters a standby mode in the "Role" state, awaiting its role assignment. Once communication commences, the "Role" state randomly assigns the participants as either the sender or the receiver, thus initiating the interactive process. The role assignment is managed by the MediaPipe software, which relays role information internally or sends it to the VTM-FSR subsystem using UDP for external communication. When functioning as a sender, the subsequent state is "Send", involving the transmission of a number. The subsequent state entails "Wait for Confirmation," signifying acknowledgement of the signal or a resend request from the partner. If a resend request is received, the system reverts to the sending state; otherwise, the sender transitions to the receiver role in the next exchange. As a receiver, the system enters the "Read" state, where it reads from the UDP connection until a message is detected. Subsequently, the receiver proceeds to the "Confirmation" state to either confirm that the signal was understood or request a resend. Should a resend be requested, the system returns to the "Read" state. Otherwise, the receiver takes up the sender role for the subsequent communication exchange. To deliver feedback on the state machine's status, MediaPipe displayed easy-to-read messages like "Waiting message" or "Confirmed". In contrast, the VTM-FSR subsystem used numerical codes communicated through VTMs, challenging players to memorize the state machine's sequence for the next game step. This complexity highlights the potential for designing a more intuitive and user-friendly interface, especially for advanced systems utilizing our subsystem communication modules.

3 Evaluation and Results

3.1 Evaluation Metrics

As a means of comparison, we computed the Information Transfer Rate (ITR) for all experimental setups. The calculation of the ITR in bits per minute is given by the following equations:

$$ITR = \frac{B \cdot \text{trials}}{T} \tag{1}$$

$$B = \log_2(N) + P \cdot \log_2(P) + (1-P) \cdot \log_2\left(\frac{1-P}{N-1}\right) \tag{2}$$

, where B represents the information transferred in bits per trial, T the time in minutes, N the number of target symbols, and P the classification accuracy. Since we transmit digits between 0 and 9, the number of targets within our system is $N = 10$.

3.2 Learning Curves

We assessed the learning process for the two subsystems VTM and FSR by measuring the ITR across five discrete sessions: one pre-training, three training, and one post-training session. Each session lasted for 2 minutes and involved the participant and one person managing the experimental pipeline on a laptop.

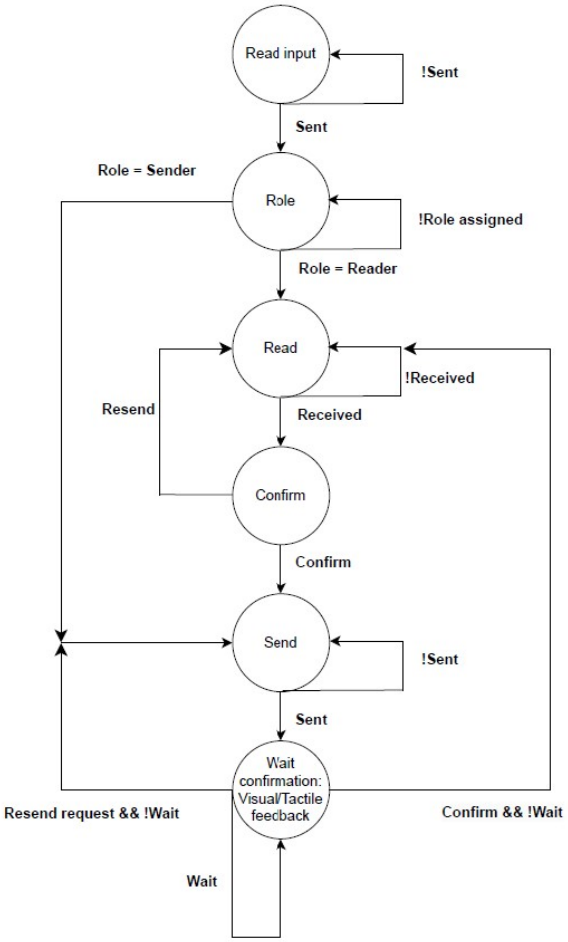


Figure 3 Finite State Machine outlining the control flow between the MediaPipe submodule, and VTM and FSR modules. This diagram exemplifies the system's modular design, highlighting the ease of integration for bidirectional communication and the potential for scalability to handle complex information exchange.

Each trial started with the experimenter pressing the 'n' key on the laptop keyboard, generating a random number between 0 and 9. For the VTM subsystem, the subject was instructed to decode the number via sensory feedback using Braille code generated by the VTMs, and to verbally communicate the decoded digit. Upon receiving the participant's input, the experimenter recorded the response in a results table structured to systematically track both the transmitted number and the number reported by the participant. For the FSR subsystem, the subject was instructed to input the Braille encoding by pressing the corresponding FSRs. The FSR Braille activation pattern was decoded into a digit, which was transmitted to the laptop through a serial connection. In both cases, verbal feedback was provided to the participant, indicating both correctly deciphered and mistaken numbers, following which a new trial was manually initiated by the operator on the laptop. Additionally, a table displaying Braille code encodings of numbers from 0 to 9 was provided during each session for the participant's convenience.

3.2.1 Subsystem Vibrotactile Motors

Two subjects participated in the study, each experiencing a different mode of vibrotactile stimulation: simultaneous and sequential. In the simultaneous mode, all motors vibrated concurrently for 1 s. In contrast, the sequential mode activated each motor individually for 250 ms, ensuring that the complete sequence for any number also spanned 1 s.

Figure 4 illustrates the learning process associated with the use of VTMs. In the initial pre-training session, we observed a lower performance for the simultaneous mode, indicating a greater initial difficulty in decoding simultaneous sensory feedback. However, this trend is reversed post three training sessions, with the simultaneous mode exhibiting a superior performance (ITR = 93.44 bpm, accuracy = 96.77 %) compared to the sequential mode (ITR = 84.52 bpm, accuracy = 98.15 %). The sequential mode exhibited a consistent, albeit modest, increase across sessions, though this does not reflect substantial gains in performance, indicating a gradual adaptation to the system rather than rapid learning. This contrast in learning dynamics highlights the differing cognitive loads imposed by each stimulation method. Comparison of accuracy and ITR over time reveals a similar trend for both curves, implying that the learning process primarily stems from reduced errors. In light of these findings, the simultaneous mode was selected for further optimization of the VTM vibration duration parameter.

3.2.2 Subsystem Force-Sensitive Resisitors

Similar to the VTM subsystem, we conducted the experiment with two subjects. However, both participants were assigned to the same experimental condition.

We used a 1000 mV (which corresponds to 0.25 N) threshold to detect whether force was applied to the FSR. After at least one of the FSRs detected above-threshold force being applied to it, we collected 15 samples and assigned "1" reading to a sensor if it was active in more than half of the total samples collected. Otherwise, we assigned a "0" value to it.

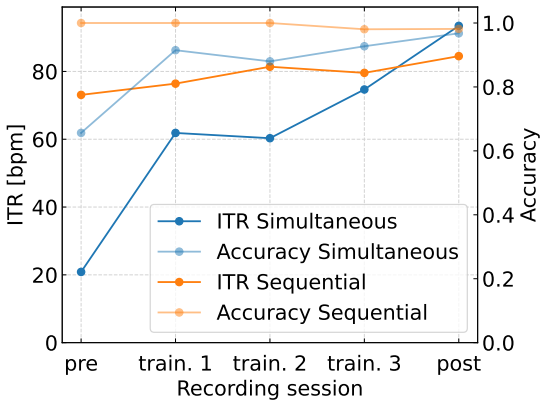


Figure 4 Learning curves in simoultaneous vs. sequential VTM stimulation modes. One participant was assigned to each condition. Initially, the simultaneous mode yielded a lower ITR and accuracy, indicating a more complex initial adjustment. Throughout the training, the simultaneous mode's ITR and accuracy improved, exceeding that of the sequential mode after three sessions. The sequential mode displayed gradual, consistent improvement, reflecting a steady, incremental learning process.

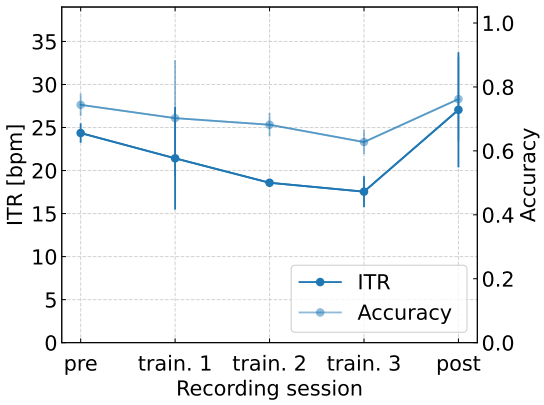


Figure 5 Learning curve for the FSR submodule. Mean and 95 % confidence intervals are shown for the ITR in bits per minute (opaque line) and accuracy (semi-transparent line), n = 2.

The learning curve for FSR-based data transmission module is presented in Figure 5. No gradual increase in ITR from the pre-training session to the post-training session is observed. Rather than that, there is a slight decline in accuracy and ITR from the pretraining session until training session 3, with a slight increase in both accuracy (76.19 %) and ITR (27.05 bpm) at the post-training session. We ran a paired t-test on ITR and accuracy values of pre- and post-training sessions. The difference in ITR and accuracy values between pre- and post-training sessions are not statistically significant ($p = 0.68$ and $p = 0.84$, respectively). Hence, we cannot state that learning occurred.

3.3 Parameter Finetuning

3.3.1 Subsystem MediaPipe

We examined four alternative sample sizes (5, 10, 15 and 20) with five participants to determine the ideal sample size for the MediaPipe subsystem. For each participant, the sample sizes were assigned in a randomized order. We conducted a two-minute experiment for each sample size. Thus, in those two minutes, and for all conditions, each participant had to sign as many numbers as possible. The numbers that had to be signed were obtained from an array containing 500 random numbers between 0 and 9. The currently detected number was displayed on the left side of the screen and on the right side the number which had to be signed by the participant was displayed. After all samples were collected, the most frequent number was stored in an array. To start a new recording session for the next number, the participant had to press the enter key manually.

As shown in Figure 6 results of the MediaPipe parameter optimization experiment showed that the ITR was best for a sample size of 15 (ITR = 60.34 bpm, accuracy = 86.20 %). Notably, even though the accuracy could have been improved with a sample size of 20 (ITR = 54.14, accuracy = 87.53 %), our optimization focus was specifically directed towards maximizing the ITR. Consequently, a sample size of 15 was selected for subsequent analyses.

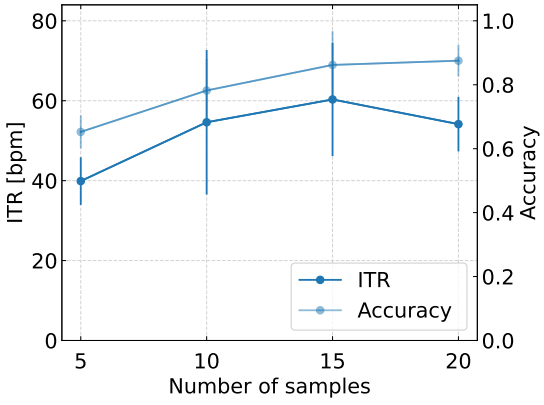


Figure 6 Finetuning of the parameter sample size in the MediaPipe subsystem. The parameter sample size indicates how many samples are collected before the signed number is classified. Displayed are mean and 95 % confidence intervals for the ITR (opaque line) and accuracy (semi-transparent line), n = 5 participants.

3.3.2 Subsystem Vibrotactile Motors

Optimization of the stimulation duration was conducted subsequent to the data collection phase for the learning curve, based on the premise that participants had become acclimated to the system and had successfully memorized the Braille code. The optimization process involved selecting the method (simultaneous vs. sequential) that yielded the highest ITR during the post-training sessions. As indicated before, the simultaneous method proved to be the more effective approach in terms of ITR. To finetune the parameter stimulation duration of the VTMs, we examined the performance for four different durations: 250 ms, 500 ms, 750 ms, and 1000 ms. The objective was to determine the optimal vibration time for the VTMs that would maximize the ITR.

The most effective ITR was achieved with a vibration duration set at 750 ms (ITR = 105.95 bpm), as evidenced in Figure 7. This parameter was subsequently used to evaluate the integrated system.

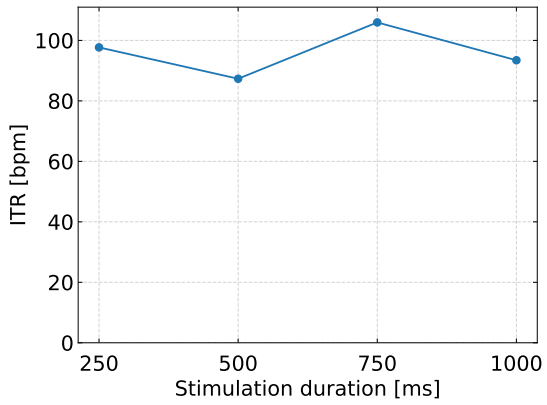
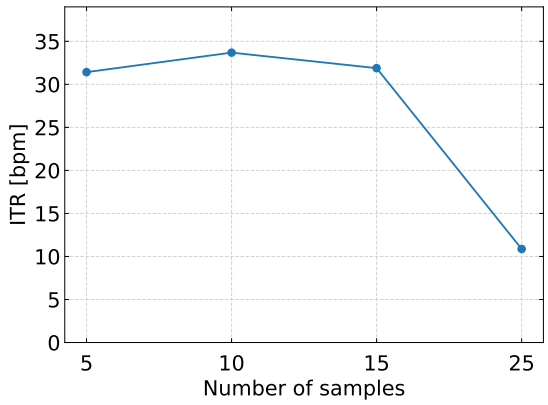


Figure 7 Vibration Duration Optimization for Simultaneous VTM Stimulation. Optimal ITR was identified at a 750 ms vibration duration, which was then utilized for comprehensive ITR evaluations of the VTM-based communication module; n = 1 subject.

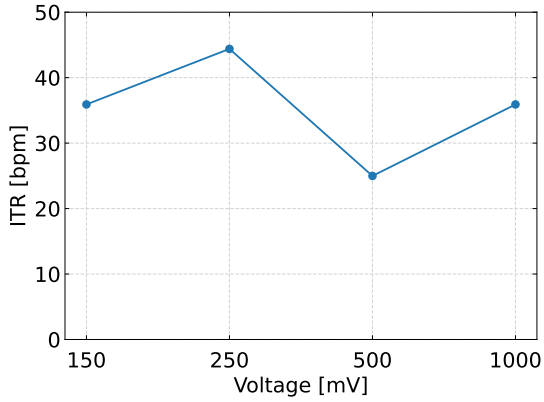
3.3.3 Subsystem Force-Sensitive Resistors

There are two parameters in the FSR-based communication module: the voltage threshold of the FSR and the number of samples collected after any of the FSRs detect the above-threshold voltage across the 10 KΩ resistor. Parameter optimization was performed after collecting the data for the learning curve, assuming that participants habituated to the system and memorized the Braille code. For each parameter, we tested four different values. For voltage thresholds, those were 150, 250, 500, and 1000 mV. For the number of samples, the values were 5, 10, 15, and 25 frames. We first chose the optimal number of samples (i.e., the one that maximizes ITR), keeping the threshold equal to 1000 mV. With the optimal number of frames, we next optimized the threshold value, also using ITR as a metric.

We obtained the highest ITR values for a sample size of 10 (ITR = 33.69 bpm, accuracy = 83.87 %) and a voltage threshold equal to 250 mV (ITR = 44.40 bpm, accuracy = 88.89 %) as shown in Figure 8. We used the trial with this parameter setting to report the ITR of the FSR-based communication module.



(a)



(b)

Figure 8 Parameter finetuning for the FSR submodule. Illustrated are the ITRs for the parameters sample size (a) and pressure threshold (b) for n = 1 (best-performing participant).

3.4 Unidirectional Integrated Systems

3.4.1 MediaPipe to Vibrotactile Motors

To assemble the MediaPipe-to-VTM integrated system, we connected the VTM-based communication

module to a laptop with the MediaPipe module running on it via UDP connection. We tested the subsystem using the optimal parameters we identified before. Two participants experienced with the subsystem and familiar with Braille code took part in this experiment. One of them transmitted self-selected numbers by performing hand gestures in front of a webcam. This allowed the MediaPipe submodule to encode these numbers into the Braille system. The encoded numbers were then transmitted to the VTM submodule via a UDP connection. The second participant received sensory feedback through the VTMs, enabling them to decode the number that had been sent. The experiment was video-recorded and the recording was manually analyzed to convert the results to a CSV file suitable for computer-aided analysis. The hand gestures were visible on the recording so that the numbers being sent could be inferred from the video. Once a number was received and decoded on the receiver's end, they said the number out loud, so that the received number could be easily inferred from the video recording.

The ITR for the MediaPipe-to-VTM subsystem has been estimated at 84.02 bpm with an accuracy of 90.77 %, based on one experimental trial. Figure 9 displays the percentage of errors for each number during transmission, with numbers 6 and 8 showing a higher error rate. These particular Braille codes require the simultaneous activation of three motors, which may lead to a higher likelihood of misinterpretation.

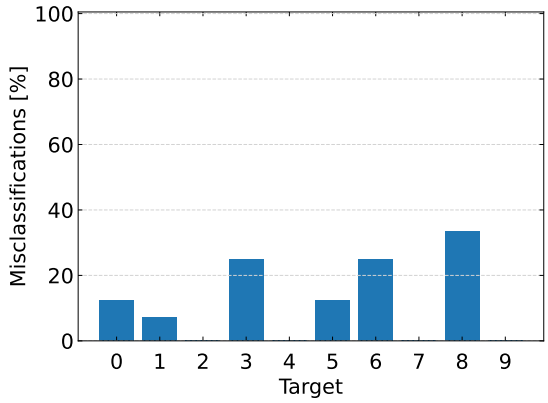


Figure 9 Missclassifications in the unidirectional integrated system MediaPipe-to-VTM. Displayed is the error rate per digit, n = 1 trial.

3.4.2 Force-Sensitive Resistors to MediaPipe

To assemble the FSR-to-MediaPipe subsystem, we connected the FSR-based communication module to a laptop with the MediaPipe module running on it via UDP connection. We tested the subsystem using the optimal parameters we determined before. Two participants experienced with the subsystem and familiar with Braille code took part in this experiment. One participant transmitted numbers that they chose themselves by pressing the corresponding encodings on the array of FSRs, and another person saw the transmitted number on the MediaPipe end. The experiment was video-recorded and the recording was manually analyzed to convert the results to a CSV file suitable for computer-aided analysis. The FSR array was visible on the recording so that the encoding being pressed could be inferred from the video. Once a number was displayed on the receiver's end, they said the number out loud, so that the received number could be easily inferred from the video recording.

The ITR of the FSR-to-MediaPipe subsystem was estimated to be 55.57 bpm with an accuracy of 80.36 %, based on 1 experimental trial. Figure 10 shows the proportion of mistakes made during the transmission of each number. The proportion of mistakes resembles a uniform distribution around all numbers except for 6 and 9 (no mistakes were made while sending these numbers using the FSR array). This shows that participants experienced difficulty with sending almost all numbers.

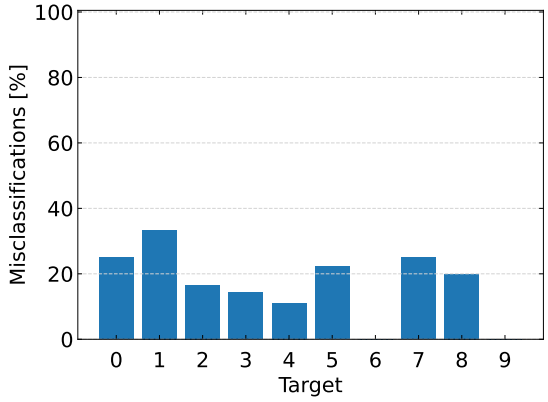
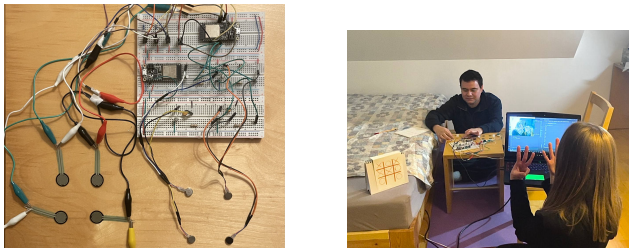


Figure 10 Missclassifications in the unidirectional integrated system FSR-to-MediaPipe. Displayed is the error rate per digit, n = 1 trial.

3.5 Bidirectional Integrated System

Finally, we integrated all subsystems to facilitate bidirectional communication. Moreover, we executed the proposed FSM and engaged two participants in a game of Tic-Tac-Toe. The setup is displayed in Figure 11.



(a) (b)

Figure 11 Setup of the fully integrated bidirectional communication system. (a) The hardware setup for the VTM-FSR module. (b) Two participants engaging in a game of Tic-Tac-Toe. The flow of bidirectional communication is controlled using a FSM as shown in Figure 3.

While MediaPipe presented easily understandable messages such as "waiting message" or "confirmed", the VTM-FSR subsystem conveyed numerical codes through VTMs, requiring players to memorize the sequence dictated by the state machine for the subsequent game step. This intricacy underscores the potential for crafting a more user-intuitive and accessible interface, particularly for sophisticated systems employing our subsystem communication modules.

4 Discussion

We engineered a bidirectional communication system with which individuals can communicate numbers via signing and Braille code, respectively. An individual signs numbers that are detected by MediaPipe and sent to the communication partner. The recipient interprets the transmitted numbers through VTMs, which vibrate according to the Braille code that represents the signed number. Conversely, the communication partner can transmit numbers back by pressing the Braille code corresponding to the desired number on the FSRs grid. The Braille code is then decoded and displayed on the MediaPipe screen, enabling reciprocal communication. We conducted a series of experiments aimed at independently characterizing each subsystem and the unidirectional integration of these subsystems with the receiving end to assess both one-way communication pathways.

For the VTM subsystem, we identified notable differences between the learning curves. While the ITR for the sequential stimulation pattern showed only minimal improvement, the ITR for the simultaneous stimulation pattern revealed steady improvement and even exceeded the sequential stimulation pattern after three training sessions. This suggests that the sequential arrangement is more intuitive, displaying higher accuracy even with an inexperienced user, resulting in a considerable ITR already in the pre-training session. In contrast, the user needs time to train and habituate with simultaneous stimulation, but the setting itself allows for a higher speed of data transmission: with 750 ms vibration duration, the individual can respond after 750 ms, while in the sequential setting, they would need to wait 1000 ms (4 * 250 ms). In the experimental protocol, we defined to only optimize the stimulation duration for the best-performing stimulation pattern due to time constraints. Consequently, the stimulation duration was only optimized for simultaneous stimulation. However, the data was only collected from one subject per stimulation pattern and therefore cannot be generalized. Furthermore, the difference in performance between the stimulation patterns after the training sessions was only marginal. Consequently, a rigorous optimization of the stimulation duration in the sequential arrangement might also improve the respective ITR. Altogether, this poses limitations to our work, necessitating further experiments involving multiple subjects to broaden the scope and validity of our conclusions. Additionally, including blind people as stakeholders would be beneficial to guide design decisions.

In contrast to this, we were unable to observe a learning effect for the FSR subsystem. This might be attributed to a suboptimal usability of the FSR subsystem during the learning curve experiments. For these experiments, the pressure threshold for the FSRs was not optimized (optimization only happened afterwards). Participants repeatedly expressed frustration regarding an FSR being pressed but no force being detected. This indicates that the choice of the threshold might have been too high, counteracting any potential learning effects.

When integrating MediaPipe with the VTMs for unidirectional communication from MediaPipe to the VTMs, we observed that the ITR for the integrated system surpassed the ITR of the MediaPipe subsystem. This is surprising, as one would expect a slight decrease in performance due to inherent transmission delays. However, it is plausible that a learning effect might have influenced the MediaPipe subsystem's performance within this integrated setup. During the characterization of the MediaPipe subsystem, we did not assess potential learning effects. Instead, we optimized the parameters for the MediaPipe subsystem

utilizing naive subjects. Thus while conducting the experiment, participants had several trials to become accustomed to the integrated system. Through these trials, some degree of learning might have taken place, enhancing the participant's interaction with the MediaPipe interface during experiments on the integrated system.

Similarly, we observed a higher ITR for the integrated system for unidirectional communication from the FSRs to MediaPipe compared to the ITR obtained from the standalone FSR subsystem. This may be attributed to the experimental arrangement. For the FSR module alone, participants received a number to encode into Braille before activating the corresponding FSRs, resulting in some number-to-Braille code encoding latency. In contrast, while testing the FSR-to-MediaPipe integrated system, the participant could send any number they wished, which made the data transmission faster. Despite the undesired latency introduced by the encoding process, we believe that our experimental setup was suitable to assess the learning curve and to choose the optimal parameter values of the FSR subsystem. However, the latency is detrimental during the estimation of the maximal achievable ITR of the subsystem.

Expanding our system by incorporating two additional VTMs and FSRs would emulate a comprehensive Braille system. This expansion would enable scalability, accommodating the entire alphabet within the system. Moreover, adjustments to the FSM could broaden the system's applicability, facilitating diverse use cases such as a communication platform for interactions between a visually impaired and a hearing-impaired individual, extending beyond simple gaming applications like Tic-Tac-Toe. The strength of our system stems from its modular design, where the operational logic is contained within each subsystem. This modularity facilitates the possibility of individual subsystem optimization concerning ITR or the potential substitution of one or more components with an alternative Brain-Computer Interface (BCI) tool if required.

5 Discussion on future applications

The future of brain-machine communication can and already is quite diverse. Today, its clinical applications range from assistive technology such as wheelchair control (Rebsamen et al., 2007), robotic arms (Chen et al., 2019; Hochberg et al., 2012), and even can help people with spinal cord injuries walk again (Lorach et al., 2023), to treating neurological disorders such as epilepsy and depression by modulating brain activity (Mayberg et al., 2005).
The long-term objective of the BCI system is to develop motion intention prediction systems that can comprehend and react to complicated human behaviours and intentions. These systems will have the adaptability and perception of natural systems. This will open the door to a future in which these technologies are deeply rooted in society, bringing with them autonomous exploratory robots, collaborative robots, and in-home assistive robots (Tang et al., 2023).
Different methods of recording signals are used for BCI systems. The most convenient to use is the IMU, as it can be easily placed embedded in a wearable device without causing discomfort and provides information about human motion. However, IMUs have a major limitation of information delay, as they become available after the movement has started. This limits the recording in real time and results in a less smooth user experience (Tang et al., 2023). A second option is the use of EMG, which directly reflects the actions of muscles and, therefore, is better for rehabilitative robotic systems. However, EMG signals acquired from paralyzed limbs are mostly weak, which negatively affects motion intention detection (Sarhan et al., 2023). Therefore, the most potential option is the EEG as it has the advantage of comfortability and low cost (Alzahab et al., 2021). It provides a method for analyzing the synchronization between muscle activity and motor cortex during movement. It is also possible to integrate it into neurofeedback systems for improving the cognitive, motor, and psychological functions (Sarhan et al., 2023). These methods were initially developed for medical purposes, but the focus has been moving toward nonmedical applications. This shift in focus does not reduce the importance of these applications in the medical domain but rather shows the wider potential of BCI applications (Värbu et al., 2022).

As BCI develops, the question of ethical issues also arises. The safety of the user and the privacy of sensitive data is required, which is covered by IEC 60601 and ISO 14971. Another concern is the growing use of AI in BCI systems as these methods are not fully understood and hence seen as a "black box". International communities are currently investigating and have therefore introduced Good Machine Learning Practices (GMLP) guidelines for BCI tools that need

to be "ticked" before reaching the end-use customer (Tang et al., 2023).

6 Conclusion

Our bidirectional communication system showcases promising advancements in bridging the gap between visually challenged and non-verbal individuals. Using our system people are enabled to convey numbers using Braille code and sign language. Through modular design and thoughtful integration of subsystems, we have demonstrated successful unidirectional and bidirectional communication pathways, emphasizing the potential for inclusive digital interactions. The results of our experiments provided valuable insights into the functioning of each subsystem. Specifically, they showed that the VTM subsystem's sequential stimulation pattern exhibited superior accuracy and intuitiveness. On the other hand, difficulties like FSR shortcomings in usability and encoding process latency were observed. Surprisingly, the ITR of the integrated system was higher than that of standalone subsystems, indicating possible learning effects. Our modular design emphasizes the adaptability of our system by laying the groundwork for scalability and diverse applications, despite limitations and the need for additional experiments.

References

Alzahab, N. A., Apollonio, L., Di Iorio, A., Alshalak, M., Iarlori, S., Ferracuti, F., Monteriù, A., & Porcaro, C. (2021). Hybrid deep learning (hdl)-based brain-computer interface (bci) systems: A systematic review. *Brain sciences*, *11*(1), 75.

Chen, X., Zhao, B., Wang, Y., & Gao, X. (2019). Combination of high-frequency ssvep-based bci and computer vision for controlling a robotic arm. *Journal of neural engineering*, *16*(2), 026012.

Hochberg, L. R., Bacher, D., Jarosiewicz, B., Masse, N. Y., Simeral, J. D., Vogel, J., Haddadin, S., Liu, J., Cash, S. S., Van Der Smagt, P., et al. (2012). Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, *485*(7398), 372–375.

Lorach, H., Galvez, A., Spagnolo, V., Martel, F., Karakas, S., Intering, N., Vat, M., Faivre, O., Harte, C., Komi, S., et al. (2023). Walking naturally after spinal cord injury using a brain–spine interface. *Nature*, 1–8.

Mayberg, H. S., Lozano, A. M., Voon, V., McNeely, H. E., Seminowicz, D., Hamani, C., Schwalb, J. M., & Kennedy, S. H. (2005). Deep brain stimulation for treatment-resistant depression. *Neuron*, *45*(5), 651–660.

Rebsamen, B., Burdet, E., Guan, C., Teo, C. L., Zeng, Q., Ang, M., & Laugier, C. (2007). Controlling a wheelchair using a bci with low information transfer rate. *2007 IEEE 10th International Conference on Rehabilitation Robotics*, 1003–1008.

Reda, M. M., Mohammed, N. G., & Seoud, R. A. A. A. A. (2018). Svbicomm: Sign-voice bidirectional communication system for normal,"deaf/dumb" and blind people based on machine learning. *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*, 1–8.

Sarhan, S. M., Al-Faiz, M. Z., & Takhakh, A. M. (2023). A review on emg/eeg based control scheme of upper limb rehabilitation robots for stroke patients. *Heliyon*.

Tang, C., Xu, Z., Occhipinti, E., Yi, W., Xu, M., Kumar, S., Virk, G. S., Gao, S., & Occhipinti, L. G. (2023). From brain to movement: Wearables-based motion intention prediction across the human nervous system. *Nano Energy*, 108712.

Värbu, K., Muhammad, N., & Muhammad, Y. (2022). Past, present, and future of eeg-based bci applications. *Sensors*, *22*(9), 3331.