IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 6 September 2023, accepted 30 October 2023, date of publication 2 November 2023, date of current version 8 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3329828

RESEARCH ARTICLE

ViolinTalk: Violin Robots as Internet of Things Devices

HSING-HSIN HUANG[1] AND YI-BING LIN[2,3,4,5,6,7], (Fellow, IEEE)
[1]Department of Mechanical Engineering, Chung Yuan Christian University, Taoyuan 320, Taiwan
[2]College of Humanities and Sciences, China Medical University, Taichung 406, Taiwan
[3]Miin Wu School of Computing, National Cheng Kung University, Tainan 701, Taiwan
[4]Department of Computer Science and Information Engineering, Asia University, Taichung 413, Taiwan
[5]Research Center for Information Technology Innovation, Academia Sinica, Taipei 115, Taiwan
[6]College of Artificial Intelligence, National Yang Ming Chiao Tung University, Tainan 711, Taiwan
[7]Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan

Corresponding author: Yi-Bing Lin (liny@nctu.edu.tw)

This work was supported in part by the National Science and Technology Council (NSTC) under Grant NSTC 112-2221-E-033-023, Grant NSTC 110-2622-8-A49-022, and Grant NSTC 112-2221-E-A49-049; and in part by the National Cheng Kung University (NCKU) Miin Wu School of Computing, Research Center for Information Technology Innovation, Academia Sinica.

ABSTRACT In recent years, the application of intelligent robots in the field of music has gradually emerged. Among string instruments, the violin stands as an unparalleled musical marvel, representing the soprano voice. Countless composers have penned masterpieces for this cherished instrument. This paper proposes ViolinTalk, a system for controlling XY-type violin robots as Internet of Things (IoT) devices. ViolinTalk enhances the performance of robots through calibration and testing. ViolinTalk has improved the software of violin robots with an IoT-based modular design, resulting in a better development environment. In this environment, we have made key contributions, including: 1) development of two control modes for the bowing servo motors, comprising displacement-based control and time-based control; 2) utilizing bowing speed control to introduce variations in volume and tone, enriching the performance effects; 3) analyzing errors in playing speed and establishing correction parameters for future use; and 4) analyzing the robot's mechanism to enhance its stability and durability. ViolinTalk also leverages IoT technology to facilitate seamless connections between different music instruments for ensemble performances. Multiple robots have been successfully integrated into a string orchestra, featuring two violins, one viola, and one cello, capable of performing standard chamber music.

INDEX TERMS Arts and humanities, bow speed control, Internet of Things (IoT), string control, violin robot.

I. INTRODUCTION

In its early stages, robots were primarily used for industrial applications [1]. For example, the study in [2] developed an IoT mobile network robot with mapping and location features for efficient route tracing, environment monitoring, and elderly care. With changing demographics and technological advancements, the use of robots has become increasingly diverse. In particular, the field of entertainment robotics has experienced rapid growth. Within the realm of entertainment robotics, music-oriented robots have gained prominence.

In recent years, the application of intelligent robots in the field of music has gradually emerged. Among string instrument robots, the violin has attracted a lot of attention. The violin stands as an unparalleled musical marvel, representing the soprano voice. Its remarkable versatility mirrors the human voice, weaving beauty and emotional resonance into every note. This instrument dazzles with its agility and intricate ornamentation, earning accolades as a pinnacle of craftsmanship. Originating in Italy, luminaries like Nicolò Amati, Giuseppe Guarneri, and Antonio Stradivari etched

The associate editor coordinating the review of this manuscript and approving it for publication was Jemal H. Abawajy.

their names in its storied history. Across the globe, the violin harmonizes with diverse cultures and musical expressions. Countless composers have penned masterpieces for this cherished instrument. As Albert Einstein once said: "I know that the most joy in my life has come to me from my violin."

Given the significant role that string instruments play in fields such as music, art, entertainment, and craftsmanship, they have attracted numerous scholars and experts for research over the years. Therefore, there are many musical robot studies and applications. For example, the flute-playing robot WF-4RIV developed by Waseda University in Japan since 1990 utilizes motors to control the flute's performance and can play in collaboration with humans. The robot's structure includes a lung and a mouth, allowing control over the airflow rate and quantity into the flute. The robot can interact with humans, and its eyebrows can move to express facial expressions during the process. In 2008, the Detroit Symphony Orchestra held a concert, where the humanoid robot ASIMO developed by Honda conducted the orchestra and performed "The Impossible Dream," a famous piece from the Broadway musical "Man of La Mancha," attracting widespread attention. The Georgia Institute of Technology developed the music-oriented robot Shimon, which can simultaneously play the xylophone and sing, going beyond the realm of traditional performance-based music robots and could be considered a creative music robot [3].

The playing technique of the violin is extremely complex, requiring not only coordination between bowing and fingering but also careful handling of various elements such as fingering changes, accurate timing, tone manipulation, volume control, and even adjustments in bowing speed and pressure, as well as the use of bowing techniques. Due to the highly integrated and intricate nature of these technical requirements, there has not been a systematic study in the development history of violin-playing robots. Toyota developed a violin-playing robot that not only stands and walks like a human but also uses its left hand to press the strings and its right hand to bow, producing melodious tunes. After its launch in 2007, the robot performed in exhibitions in Tokyo and Shanghai, garnering widespread attention [4].

Our team is dedicated to the research of string instrument robots. Starting from 2005, we began developing XY-type violin robots. Figure 1 shows a recent version of our violin robot. The advantage of these robots lies in simplifying the intricate and complex techniques of human performance. The bowing motion of the right hand is simplified by utilizing a linear slide driven by an X-axis servo motor, achieving precise bowing control. For the violin body, the Y-axis stepper motor is used to rotate the instrument body for string changing, and a pneumatic device is employed to control the finger mechanism for producing different pitches. As a result, the XY-type violin robot not only possesses precise positioning and high-speed movement capabilities but also comes at a lower cost, presenting a significant advantage in the development of music-oriented robots.

In the past, software control architectures for violin robots were not reported in the literature. This paper proposes ViolinTalk, an IoT-based control system for XY-type robots. By connecting the XY-type robot to the CATtalk IoT development platform [5], ViolinTalk provides a powerful environment for enhancing the performance of XY-type robots through calibration and testing.

This paper is organized as follows: Section II overviews the related studies; Section III describes the design and implementation of the XY-type violin robot; Section IV proposes ViolinTalk based on the CATtalk IoT development platform; Sections V-VII elaborate on the design and implementation of the bowing, string-changing, and finger-pressing software modules and their integration with the mechanical mechanisms; Section VIII shows how ViolinTalk effectively calibrates and plays the XY-type violin robots; Section IX discusses the interaction among multiple music instruments.

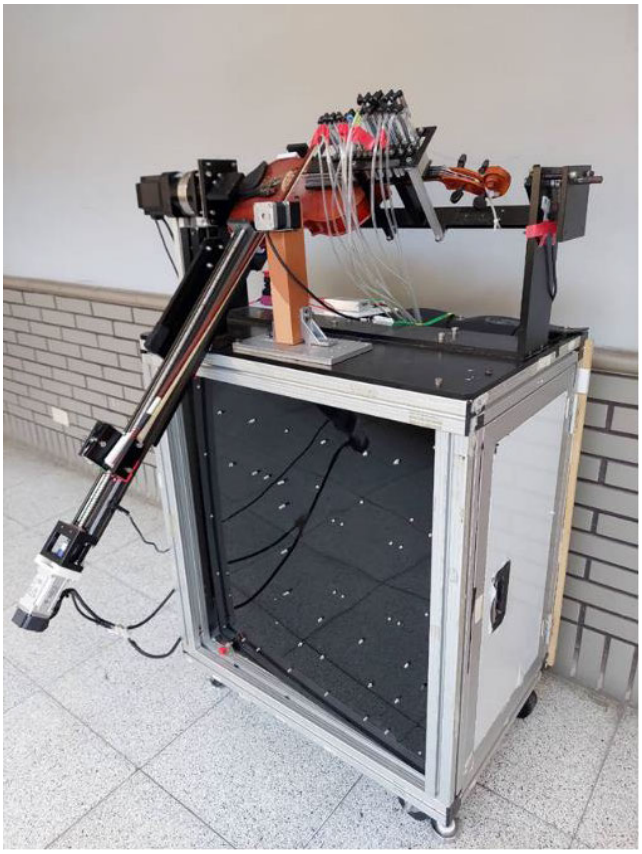


FIGURE 1. The XY-type violin robot.

II. RELATED WORKS

The research on violin-playing robots generally falls into three main directions. The first direction focuses on studying the tonal quality of violins and analyzing parameters that affect playing quality through vibration and noise analysis methods. The second direction emphasizes the analysis of playing techniques, including studies on fingering and bowing techniques. The third research direction centers around the development of automated mechanisms or robots for playing the violin.

In the first direction, the study in [6] utilized 1/3 octave frequency analysis and optical interference techniques to investigate the impact of violin soundbox vibrations on its tonal quality, thereby studying the design parameters of the f-hole positions and shapes on the soundboard. In [7], the authors compiled several papers related to violin acoustic research, categorizing and discussing the effects of factors such as the bow, bridge, soundbox, soundpost, wood, varnish, and others on violin vibrations and sound. The study in [8] employed spectrum analysis techniques to explore the influence of string vibrations on violin sound, aiming to identify parameters that affect playing quality.

In the second direction, the authors in [9] utilized artificial neural networks to analyze violin fingering during performance and established a set of optimized decision rules. These rules can offer performers references for shifting positions and finger placement. The frictional movement between the bow and strings determines the quality of the performance. The study in [10] specifically investigated the positioning, force, and speed of bowing to explore how they affect the presentation of volume and timbre.

The study in [11] developed an automated violin-playing robot capable of interpreting musical scores to determine performance nuances. The authors explored designing tempo variations within each bar and sound pressure for individual musical notes, aiming to convey bright and dark impressions. Initial analysis showcased a trained violinist's performance indicating faster tempo for bright timbres and notable sound pressure shifts for brightness. Building on this, the authors proposed a methodology for expressing bright and dark timbres based on these findings. Through experiments, an anthropomorphic violin-playing robot was employed to produce sounds, with wrist joint angle adjustments influencing sound pressure variations. The analysis confirmed the similarity between produced sound pressure patterns and the designed ones for bright performances. Ten participants evaluated the sounds, successfully distinguishing between bright and dark performances when variations in sound pressure and tempo were present. This research contributes to enhancing the authenticity of automated violin performances by integrating nuanced timbral and expressive elements.

The pitch estimation of a specific musical source in a multi-source polyphonic sound is vital for music performance analysis. One approach to extract the target source's pitch involves source separation followed by pitch estimation. However, as demonstrated in [12], this often yields unsatisfactory outcomes. Presenting an alternative, a timbre-sensitive pitch estimator (TAPE) was introduced. TAPE directly gauges the target source's pitch, eliminating the need for explicit source separation. Unlike existing methods presuming a dominant lead voice, TAPE relies solely on timbral attributes. In real violin-piano duet tests, TAPE, trained on synthetic blends, outperforms sequential source separation and pitch estimation. Notably, this holds true across various scenarios, even when the target source isn't dominant. This innovation showcased the effectiveness of accurately estimating pitch without prior preprocessing.

A crucial need exists for a system that accurately records and replicates violin playing movements to preserve expert skills. However, this demands interaction with the violin, which can be challenging when precise force, position control, or extensive freedom is required. In response, the study in [13] introduced a motion sensing method that avoids interference, effectively measuring violin playing skills. This method was applied to compare the abilities of skilled and novice violinists using data captured by the system. The study endeavors to highlight distinctions between the two player categories by utilizing wavelet transform on the recorded data, thus analyzing them within the frequency domain. This approach offers insights into skill disparities based on nuanced motion patterns during violin performance.

This paper proposes the ViolinTalk approach that will focus on the third direction. In this direction, the study in [14] developed a robotic device possessing six fingers and utilizing a system of cables to achieve the action of finger pressing on the strings. For changing strings and bowing, string replacement is accomplished through the rotation of a circular axle, while bowing is achieved by moving a cylindrical rod. This approach has two issues. The first one is that the tuning mechanism has only six gears and is limited to lower position (Figure 2), resulting in a smaller range of play. The second issue is that the bowing mechanism is driven by wheels, restricting operational flexibility, preventing high speeds, and making string changes prone to errors.

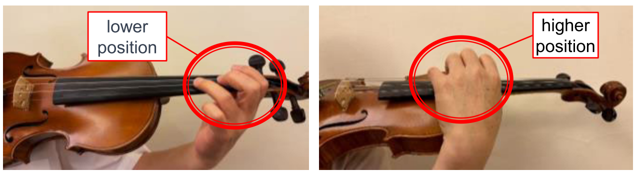


FIGURE 2. Positions on a violin.

Toyota developed a humanoid robot that possesses a human-like physiological structure, enabling it to walk upright and utilize both hands to play the violin, producing remarkably beautiful music [4]. While this robot demonstrates superior performance and high integration capabilities, it falls short in its ability to bow rapidly with its right hand and lacks the necessary dexterity in its left hand's finger movements on the strings. As a result, it can only play relatively simple musical compositions. Similar to the study in [14], the Toyota robot's finger placement on the strings is limited to lower positions on the fingerboard, restricting the range of play. When the tempo of the music is slightly faster or when there are larger pitch variations, the fingers and wrist must coordinate to enable rapid and extensive finger movement. Due to the considerable technical challenge involved, achieving this capability with humanoid finger mechanisms might not be easy at the moment.

In 2012, we published the XY-type violin robot implementation [15], which has two sets of systems (Figure 1): The first system employs a servo motor to execute bowing actions, enabling precise control of the bow's movement on the strings. The second system uses a stepper motor to rotate the body of the instrument, allowing the bow to switch between different strings for playing. Additionally, multiple sets of pneumatic devices are used to control the string pressing. Combined with the bowing design, this enhances the expression of tones.

In 2015, the author in [16] showcased Ro-Bow, a kinetic sculpture that employs robotic fingers (specifically, electromagnetic actuators) to play digital music files on a real violin. Similarly to the XY-type robot, this system employs a platform capable of rotating at various angles to achieve string replacement for the violin. The bow is affixed to a sliding track to execute back-and-forth movements for bowing during play. For finger pressing on the strings, a sliding track mechanism incorporates four circular blocks covered with rubber rings on the outer layer, acting as fingers. These can be moved laterally to perform the action of finger pressing while playing. This violin robot employs four finger mechanisms, all driven by levers, which press down when they reach a specific point. The entire mechanism is driven by a linear slide, expanding the range of play. A drawback is that the finger and tuning mechanisms are separate; the fingers must be released before adjusting the violin's position, resulting in reduced flexibility. Additionally, during string changes, there can be interruptions or delays between notes. Another downside of this design is its considerable size and poor mobility, making it less feasible for widespread adoption.

The study in [17] and [18] replicated the playing techniques of a violin expert and applied them to a robot. The research explored how the quality of the produced sound is influenced by factors such as bowing force, bowing velocity, and sound point when executed by a violin-playing robot. A vertical multi-joint robot arm controlled the bowing movement, and the sound quality analysis was conducted using Fast Fourier Transform (FFT) in a 32-bit microcontroller unit. Experimental outcomes revealed a proportional relationship between bowing velocity and sound magnitude. However, when the velocity exceeded a certain threshold, harsh noise emerged, degrading sound quality. The optimal bowing velocity for G string's natural frequency was determined to be 110 mm/sec, aligning with violin experts' assessment of optimal velocities between 90 mm/sec and 110 mm/sec. The study underscored the significance of bowing force and velocity in sound quality alteration, pinpointing an optimal velocity that corresponds to a specific bowing force. This sutdy only discussed the impact of bowing strength, speed, and contact point on sound quality, offering insights into bowing control as a reference. However, it did not cover the complete spectrum of robot control.

The study in [19] introduced a Matlab application to simulate the movements of a violin-playing robot's left hand. The application serves as a platform with features enabling users to evaluate motion effectiveness using performance indices applied to the hand's fingers. Additionally, the software provides tools for assessing the hand's overall ability to play a song. Users can visualize hand and finger movements through animation and hear simulated violin sounds. Motion programming is facilitated through an intuitive graphical user interface aligned with the violin's diapason. The software demonstrated basic functions for motion simulation and sound programming. The software's development stage focuses on refining motion synthesis, energy optimization, and expert evaluation for enhanced efficiency. This approach takes into account the motion of human-like fingers. From the results, it appears that currently, the motion is still limited to lower positions on the fingerboard. If development could extend to higher positions, its practical value would be significantly enhanced.

Developed by the Georgia Institute of Technology and utilizing AI learning, the system Shimon can learn performance techniques from over 50,000 song samples [3]. It possesses functions such as automatic lyric generation, composition, and singing. It can even play instruments like the marimba during performances. This robot has already gained experience in live rap battles alongside humans, and has even released multiple singles and albums. This work focuses on the functionality of automatic composition, which can serve as a reference for us to develop automatic sheet music transposition.

Based on our work in [15], we propose ViolinTalk to significantly improve the design and simplify the calibration and playing of the XY-type violin robot.

III. THE XY-TYPE VIOLIN HARDWARE

The first author of this paper has dedicated over 18 years to developing the XY-type violin robot. The robot comprises the following hardware components: Component X (X-axis) includes the violin bow (Figure 3 (1)) and bowing mechanism (Figure 3 (2)). Component Y (Y-axis) includes the violin body

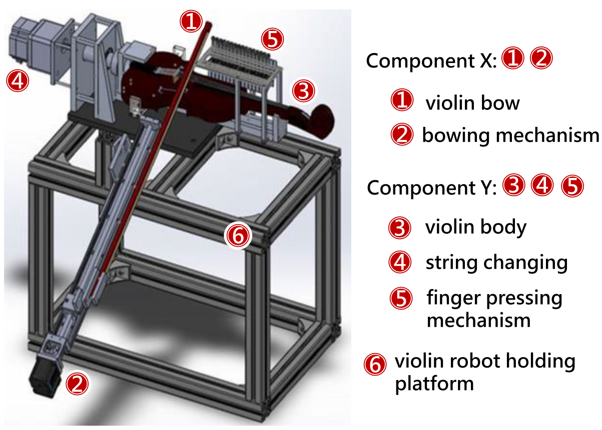


FIGURE 3. The hardware components of the XY-type robot.

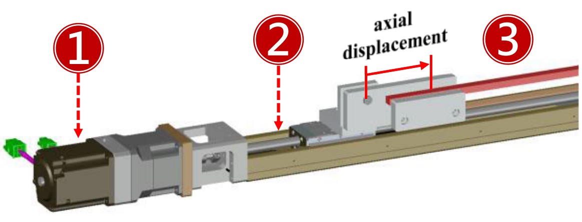


FIGURE 4. The bow-holding mechanism.

(Figure 3 (3)), string-changing mechanism (Figure 3 (4)), and finger-pressing mechanism (Figure 3 (5)).

The bowing mechanism utilizes a Panasonic MINAS A6 servo motor (100W; Figure 4 (1)) to drive a linear slide equipped with a ball screw (Figure 4 (2)). On the slide, a bowing apparatus is mounted, featuring a flexible interface pivot for holding the bow (Figure 4 (3)). This interface pivot also provides damping to ensure that when the bow is in contact with the strings, it can counteract the reactive force of the strings and reduce bouncing. The pivot employs a clamping fixture to secure the violin bow. Conventional clamping fixtures can limit the flexibility of the bow and may result in unusual sounds during play. To address this issue, our clamping fixture is designed to offer flexibility, allowing the bow to extend forward and avoiding any restrictions on the bowstring's tail caused by the mechanism. This design reduces bow wobbling, simplifies assembly and disassembly, and facilitates immediate post-use maintenance.

The string-changing mechanism utilizes a stepper motor with a gearbox to control the rotation of the violin body during string changes. This motor drives the instrument's body and ensures that the violin is positioned at the appropriate string location according to the performance requirements. We have two implementations for changing the strings. The first implementation uses a simple supported beam structure, relying on rotating axles at both endpoints to support and enable the rotational movement of the violin (Figure 5 (1)). In this implementation, the misalignment issue between the two bearings could affect the rotation, causing anything from slight disruptions to severe mechanical damage. Figure 5 (2) shows the deformed state of the structure, where a tendency towards red indicates a larger deformation, while a tendency towards blue indicates a smaller deformation. The figure indicates that this structure is essentially robust (no red areas). However, it requires more space and is, therefore, not very portable.

In the second implementation, a cantilevered mechanism was adopted. This mechanism relies on a single-sided axle for support and the rotational movement of the violin (Figure 5 (3)). This modification effectively eliminates the problem of misalignment at the center. While using a single-sided support could potentially lead to downward displacement of its end due to the weight (as indicated by the red area in Figure 5 (4)), our testing has shown that this design alteration does not pose any problems. This structure takes up less space and is more portable.

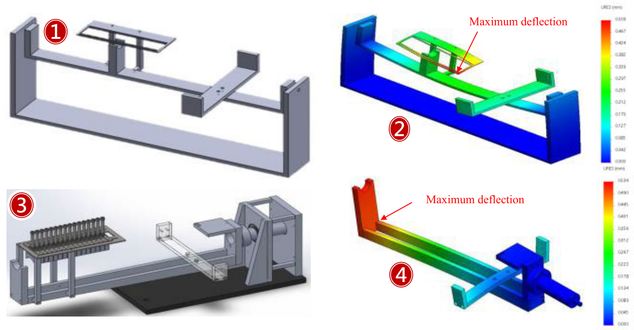


FIGURE 5. The string changing mechanism.

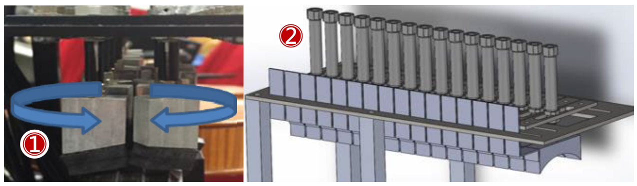


FIGURE 6. The finger pressing mechanism.

The finger-pressing mechanism utilizes multiple sets of pneumatic controls to adjust the position of the mechanical fingers on the strings. Initially, a rotary pneumatic cylinder was employed, which often resulted in deviation (Figure 6 (1)). To address this issue, a limit device was introduced to prevent rotational problems from occurring (Figure 6 (2)). We have implemented varying numbers of mechanical fingers, ranging from 5 to 20. This paper utilizes 12 fingers, as elaborated in Section V.

IV. THE ViolinTalk ARCHITECTURE

Based on the CATtalk IoT application creation platform [5], we develop the ViolinTalk server (Figure 7 (1)). The ViolinTalk server is installed on an industrial Raspberry Pi4 (Figure 8 (1)). With the standard Pi4, the ViolinTalk Python code can be easily compromised and illegally modified, leading to malfunctions. To resolve this security issue, we collaborated with Winbond to modify the Pi4 hardware using the W77Q TrustME®Secure Serial Flash memory (Figure 8 (2)). This modification helps us prevent unauthorized access.

ViolinTalk treats an XY-type robot as an IoT device, and we have developed the Calibration and Performing modules (Figure 7 (2) and (3)) to connect the XY-type robot (Figure 7 (7)) to the ViolinTalk server. A ViolinTalk module consists of two parts: the Device Application (DA) and the Sensor & Actuator Application (SA). The DA is responsible for connecting an IoT device to the ViolinTalk server, and the SA implements the functionality of the IoT device. To reduce communication delays, the Raspberry Pi4 for the ViolinTalk server is co-located with the robot, and they are connected through a wired Ethernet connection.

The SAs for the Calibration module include the Calibration SA, Bowing SA, String-Changing SA, and Finger-Pressing

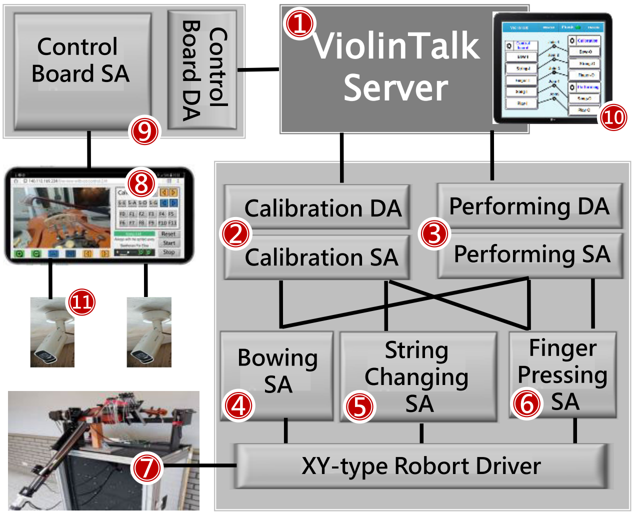


FIGURE 7. The ViolinTalk architecture.

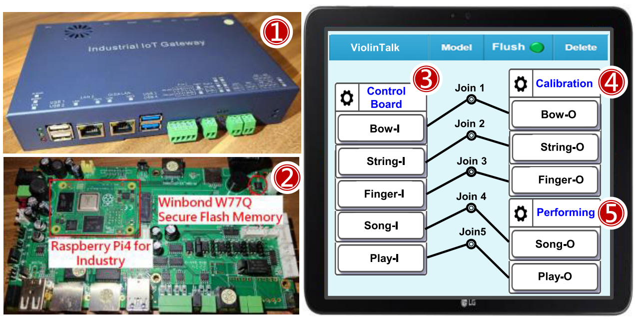


FIGURE 8. The ViolinTalk hardware and its GUI.

SA (Figure 7 (2), (4)-(6)). The Performing module shares the last three SAs (Figure 7 (4)-(6)) with the Calibration module.

To play a song, Performing SA requires converting the sheet music into a standardized format, structured into overall parameters and individual note parameters. The overall parameters include the total number of notes, the starting string position for the first note, motor reference speed, and prelude waiting time. The individual note parameters encompass rests and pitches (low/mid/high, natural/ raised/ lowered), string position, bowing direction, bowing time, bowing speed, and whether to trigger the vibrato function. The details will be addressed in a separate paper.

In Bowing, String-Changing and Finger-Pressing SAs, LabVIEW is used to write control programs and interfaces, which are paired with Advantech's motion control interface card PCI-1245 and USB-5830 IO module [20] to control the X and Y component mechanisms.

To flexibly control the violin robot, ViolinTalk has developed an IoT device called the Control Board (Figure 7 (9)). The DA of this device typically communicates with the ViolinTalk server through either wired or wireless (5G or Wi-Fi) connections. The SA provides a web-based Graphical User Interface (GUI) that allows users to control the robot using any mobile device with a web browser (Figure 7 (8)). The web-based GUI enables users to remotely watch the violin (Figure 9 (5)), calibrate the robot (Figure 9 (1)), and play songs (Figure 9 (7)).

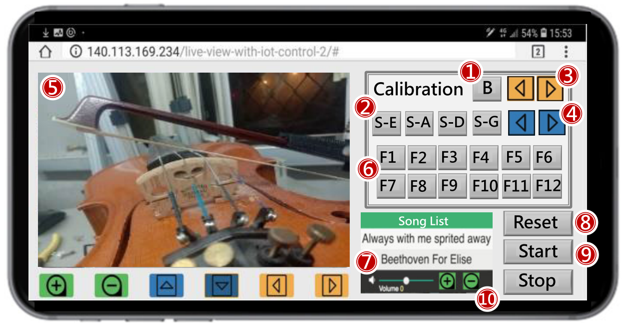


FIGURE 9. The web-based GUI for control board.

The connection between the Control Board module, the Calibration module, and the Performing module can be easily established through the ViolinTalk GUI (Figure 7 (10)). Each IoT device is represented by two icons in the ViolinTalk GUI. The sensor and control parts of the device, such as the Control Board module (Figure 7 (9)), are represented by icons placed on the left side of the GUI window (Figure 8 (3)). The actuator parts of the device, including the Calibration module (Figure 7 (2)) and the Performing module (Figure 7 (3)), are represented by icons placed on the right side of the window (Figure 8 (4) and (5)).

A sensor, button, or actuator is represented by a small icon within the device icon. For example, the bow calibration buttons (Figure 9 (1) and (3)) are represented by the "Bow-I" icon within the Control Board icon, while the bowing mechanism in the robot (Figure 7 (7)) is represented by the "Bow-O" icon within the Calibration icon.

To control the bowing mechanism using the bow calibration buttons, we simply drag a link labeled "Join 1" between the "Bow-I" and "Bow-O" icons. Therefore, configuring the IoT devices and their connections in Figure 7 can be easily accomplished in the ViolinTalk GUI using Joins 1-5 in Figure 8. For more details, please refer to [5]. ViolinTalk also installs multiple cameras (see Figure 7 (11)) to observe the robot's operation, with the video streams directed in real-time to the video viewer (see Figure 9 (5)).

Through the ViolinTalk GUI, users can create various application projects for musical instruments. The primary project is referred to as the ViolinTalk project (Figure 7 (6)). All IoT devices involved in the project, such as the Control Board, Calibration, and Performance components, can be selected from the "Model" dropdown list (Figure 7 (7)). More details can be found in [5].

V. FINGER-PRESSING SA AND THE MECHANISM

This section describes how the Finger-Pressing SA (Figure 7 (6)) generates pitches on a violin. To explain the pitch positions of the violin strings on the fingerboard, we use G, D, A, and E to represent the names of the four strings, and

0-12 to represent the relative positions of the pitches on each string. As shown in Figure 10, you can observe that on a single string, the range from 0 to 12 encompasses a complete octave scale.

Using the four fingers of the left hand to press the strings within the range of 0 to 7 on the fingerboard is referred to as the first position. As the pitch gradually rises, the player needs to move their left wrist. When the index finger shifts to the position of the 5th pitch, it is known as the third position fingering. Since the positions of the pitches often vary with the complexity of the music, in a piece intended for performance, the variations in pitch often span more than two octaves. Taking Pachelbel's Double Violin Concerto as an example, throughout the piece, the range of pitches spans as many as 18 pitches from the lowest to the highest note. Additionally, in the music, there are often instances where the difference between two adjacent pitches is greater than 10 degrees.

In ViolinTalk, when changing strings, overcoming the inertia of the mechanism itself can lead to speed delays, particularly when playing fast-paced musical passages. This is due to the inertia causing a delay in the response time. To address this issue, ViolinTalk employs an array-type finger mechanism, increasing the number of finger placements on the strings, allowing for a greater range of pitches to be played on the same string. Our finger pressing mechanism eliminates the need for shifts in finger positions, thus enabling finger pressing actions to be executed more efficiently.

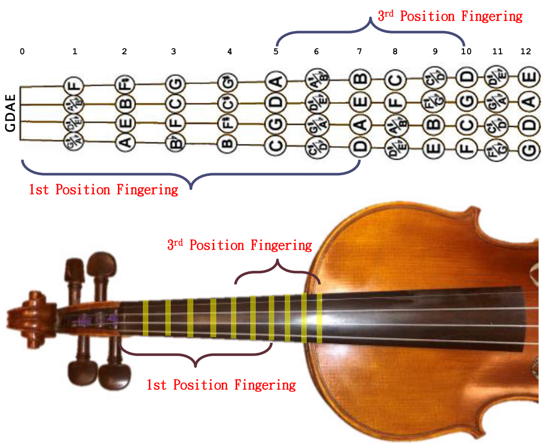


FIGURE 10. The pitch positions.

Finger-Pressing SA controls 12 mechanical fingers (Figure 11). Each finger can move to any of the four strings. When we designed Finger-Pressing SA, it's important to avoid changing strings in sections with fast tempos and short-duration notes. If there is a sequence of consecutive notes with significant pitch fluctuations that cannot be managed by a single string and occur in rapid succession, efforts should be made to confine the string actions to the two nearest strings. This approach helps alleviate the strain on the string changing mechanism.

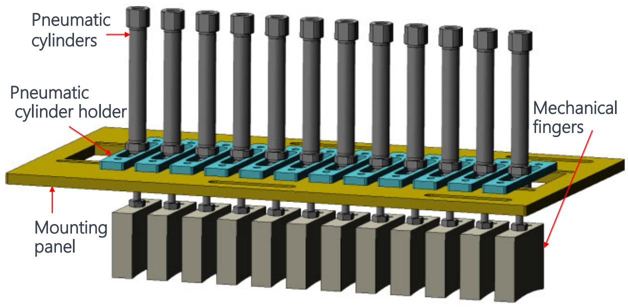


FIGURE 11. The pneumatic string pressing mechanism.

The pneumatic string pressing mechanism comprises a total of 12 sets of pneumatic cylinders covering 20 pitches. The pneumatic cylinders are initially attached to holders, which feature small holes for mobility on the mounting panel, facilitating position adjustments. A distinctive feature of this mechanism is arranging all pneumatic cylinders based on the violin's string positions, mounted onto the same panel. Notably, flexible pads are installed at the ends of the pneumatic cylinders to eliminate noise generated when the mechanism impacts the fingerboard. The violin fingerboard is narrow at the top and wider at the bottom, sharing an arc shape yet exhibiting irregularities in their design. Consequently, the design of the string pressing mechanism must be adjusted to accommodate the fingerboard's shape. After completing the structure, fine-tuning of positions and pitch testing remain necessary (to be elaborated in Section VIII).

VI. STRING-CHANGING SA AND THE MECHANISM

Based on the first note to be played, the string changing mechanism sets up the initial position for the stepper motor. The overall speed is determined according to the beat of the musical score. When converting the individual note parameters, various considerations come into play. This task is carried out by Performing SA (Figure 7 (3)). The specific conversion process should follow this flow:

1. Determine the scale of each played note and convert it into a code.
2. Due to the use of 12-finger mechanism covering an octave on a single string, the same pitch may appear on different string positions. Therefore, it needs to be determined based on the current performance situation whether a string change is necessary.
3. Identify the type of note and determine the duration of the note's performance.

Performing SA instructs String-Change SA (Figure 7 (5)) to move to the target string. We discovered that vibrations have minimal impact on the XY-type robot except for the fingerboard (neck) vibration of the violin, which is caused by the stepper motor of string changing mechanism (Figure 3 (4)). We employ a high-speed camera to identify vibrations in the top of the scroll, namely, the P1 point in Figure 12. We set point P1 as the center to understand the

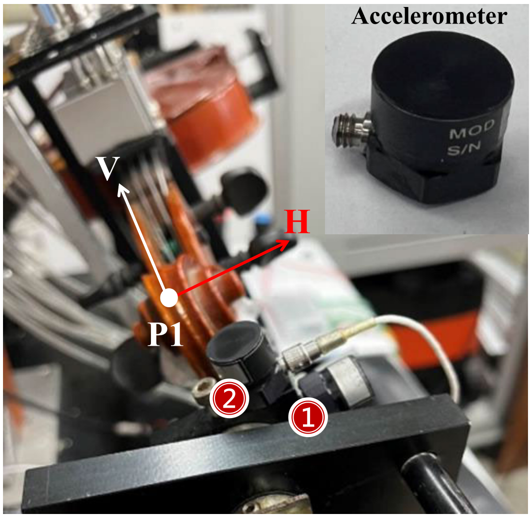


FIGURE 12. The vibration of the fingerboard.

oscillations caused at point P1 during the rotational string changing motion of the violin.

Through the web-based GUI of the Control Board, we can stream the high-speed camera feed to the video viewer (Figure 13 (1)) to observe the vibration of P1. During the vibration test, the user can control the string changes through the calibration buttons (Figure 13 (2)). Observing the images captured by the high-speed camera, we noticed that when the violin body rotates, the horizontal amplitude at the top is approximately 12mm, with a rightward offset of about 4mm from the center. The vertical amplitude at the top is approximately 7mm, with a downward offset of about 0.5mm from the center. Adjusting the angle of the violin body can help improve the offset condition.

Additionally, we have integrated two accelerometers onto the fingerboard: one positioned horizontally (Figure 12 (1)) and the other vertically (Figure 12 (2)) to measure the vibrations caused by the rotation and cessation of the violin body in different directions.

In the Calibration1 project (Figure 14 (5)), these two accelerometers function as an IoT device set (Figure 14 (4)), named Vibration-I1 for the horizontal accelerometer and Vibration-I2 for the vertical accelerometer. The measured samples are transmitted to the Control Board (Figure 14 (2)) via Joins 2 and 3. Based on the received samples, the Control Board (Figure 14 (1)) determines how to control the String-Changing SA and sends instructions to the Calibration SA (Figure 14 (3)) through Join 1 for subsequent tests.

The dashboard of the GUI (Figure 13 (3)) illustrates the time-series vibration charts of the fingerboard as measured by the accelerometers. Due to the greater magnitude of horizontal vibrations compared to vertical ones, we will only explain the detection and improvement results in the horizontal direction. The motor starts at Figure 13 (4) and stops at Figure 13 (5). Before adjustment, the horizontal vibration was quite significant. Upon inspection, it was found that the cause was the loose keyway coupling between the motor and the turning mechanism. The improvement method involved tightening the keyway with screws. Upon retesting, it was discovered that the vibration phenomenon had significantly reduced.

FIGURE 13. The vibrations measured by the high-speed camera and the accelerometers.

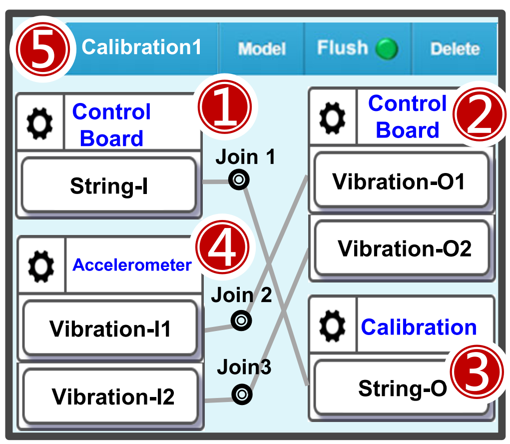


FIGURE 14. The calibration1 project.

VII. BOWING SA AND THE MECHANISM

Bowing SA (Figure 7 (4)) employs two bowing control methods: displacement-based and time-based. The displacement-based method involves calculating the displacement for each note based on the initial tempo of the music (measured in beats per minute or BPM). This facilitates the writing of control commands and ensures that the bow remains within a defined range during performance. The displacement-based method requires executing a sequence of startup, acceleration, constant speed, deceleration and stop processes for every single note. Furthermore, after issuing displacement control commands, confirmation of the motor reaching the intended position is required before Bowing SA issues the next control command. Consequently, this method often results in waiting delays between notes. While these delays are typically brief, in cases where the tempo is relatively fast, especially when playing legato passages, discontinuities in the performance can frequently arise due to these delays.

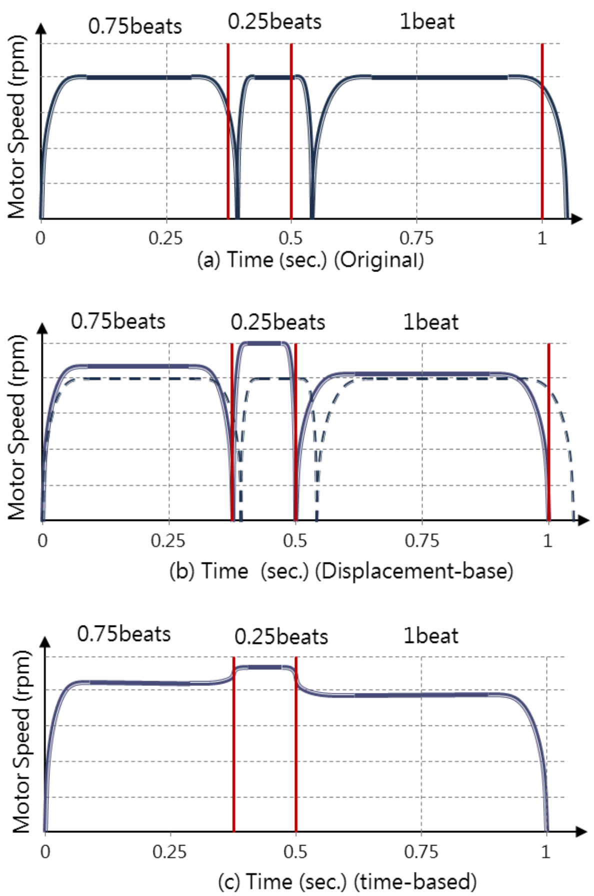


FIGURE 15. The speed and time variations in the displacement-based approach.

As depicted in Figure 15 (a), we also observed that achieving the necessary displacement for a particular note requires a significant amount of time for both acceleration and deceleration, resulting in the actual duration for that note being longer than intended. This phenomenon has a more pronounced impact on shorter note durations compared to longer ones. To mitigate this issue, adjustments must be made to the speed of each beat. As shown in Figure 15 (b), by capturing the bowing motion using a high-speed camera and subsequently fine-tuning the speed for each beat, real-time effects can be achieved. Through this method, Bowing SA can synchronize with musical accompaniment, enhancing the overall performance. However, the tuning process is time-consuming, especially for longer musical scores, often resulting in discrepancies that require manual correction. This becomes a major drawback of this method.

In the time-based method, Bowing SA first determines the tempo of the music, and sets the duration of a single beat. When issuing control commands, Bowing SA only needs to establish the bow's position and remaining bow length. Based on the note's duration and the beat's tempo, the control algorithm is executed. While keeping the time constant and the bowing distance within the allowable range, the time-based method allows Bowing SA to use varying speeds, facilitating changes in the performance style. Another advantage of this method is that when playing legato, the transition between notes no longer requires delays; instead, the speed changes directly to smoothly connect to the next note, as shown in Figure 15 (c). Therefore, the time-based method not only helps eliminate pauses but also achieves superior real-time control effects.

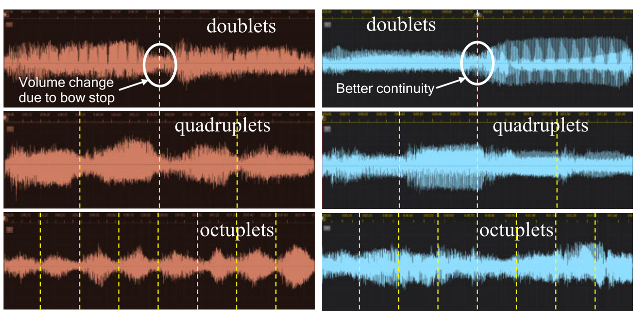
When programming the musical score for Bowing SA, speed and time are set as parameters to manage the beats and determine the duration based on the musical score. The main advantage is that during legato playing, the connection between notes no longer requires a sequence of positioning, complete stops, waiting for a command, and then starting again. Instead, speed control commands of the time-based method are used to manage the motor's motion. This not only eliminates pauses but also enhances real-time control.

The effective stroke of the linear slide (Figure 4 (2)) is approximately 55cm. It requires sensor-based origin calibration and confirmation of the distance to the upper and lower limits from the origin point. To avoid collision situations, the control algorithm for Bowing SA must consider the current position of the linear slide to prevent exceeding the effective control distance, which could result in damage due to collisions.

Multiplying the speed by time yields the bow length for each note (as elaborated in Eq. (1) later). Since the bow's stroke is constrained by the linear slide, programming the musical score requires consideration of the remaining distance based on the previous note's position. The decision whether to adjust the speed or perform a bow change depends on the desired performance effect. Furthermore, even though there are no speed drops to zero during the legato process, there is inevitably a sequence of motion-deceleration, stop, and reverse motion during bow changes. Therefore, during bow changes, there will be delays due to the deceleration and acceleration processes. These minor delays can contribute to errors in the bow's position during the direction change of the bow. If these errors are not corrected, they may affect the timing or result in insufficient bow length.

As mentioned earlier, while velocity control can address the delay caused by waiting for commands during legato playing, playing the violin inherently involves bow changes. Consequently, during these bow changes, there will still be delays due to the deceleration-stop-start process, leading to timing delays. When implementing the control algorithm for Bowing SA, it becomes necessary to make adjustments for this delay to ensure it doesn't affect the overall timing. To determine the correction amount, an analysis of velocity variations during the motion process is required.

We compared the legato performance effects of the displacement-based and the time-based methods using a noise meter. Figure 16 explains that the time mode is more suitable for legato bowing than the displacement mode. The figure reveals that in the displacement-based method, distinct pause points between the notes are evident during the performance of doublets, quadruplets, and octuplets, resulting in a discontinuous auditory perception. Conversely, in the time-based method, the transitions between the notes in the doublets, quadruplets, and octuplets are smoother. For doublets, displacement-based method causes stops, which are not found in the time-based method. As the tempo increases and the performance speed rises, the continuity of the time-based method remains better. In quadruplets, and octuplets, there are lumps in the displacement-based method, which are not found in the time-based method. For example, in the playing process of double, quadruple, and octuple stops, there is a noticeable volume change between notes due to the pause of the bow in the displacement mode, leading to a discontinuous auditory sensation. In the time mode, however, the connection points exhibit better continuity. This confirms that the time-based method enhances the legato performance effect.



(a) Displacement-based (b) Time-based

FIGURE 16. The results of Legato performance evaluation.

Since the bow control inherently involves three types of motion: acceleration, constant speed, and deceleration, the overall time must align with the beat duration. However, discrepancies in displacement compared to the ideal displacement can occur. These discrepancies can become apparent when playing longer notes or legato passages and may even result in the bow exceeding its designated range. To avoid such scenarios, ViolinTalk proposes the following resolution. Firstly, we execute continuous back-and-forth performances of various distinct notes. The outcomes are recorded and compared against the intended displacement, serving as corrective information for subsequent adjustments. Furthermore, we utilize feedback signals from the servo motors to determine the endpoint of each individual note. By comparing the required displacement for the upcoming note with the remaining bow length, real-time adjustments to the velocity of each note will be made, ensuring that the bow doesn't encounter overshooting.

Apart from mastering rhythm and pitch, controlling the volume is also an essential aspect of Bowing SA. By manipulating volume variations, performances can exhibit dynamics that effectively convey emotions. VoiceTalk initially establishes the following bowing principles:

Bow Rule 1(Bow Angle and Path): The bow's surface is slightly tilted during bowing, ensuring the bow's path is perpendicular to the strings while avoiding contact with other strings.

Bow Rule 2 (Initiation): Begin with gentle force to prevent unwanted noises caused by excessive friction. VoiceTalk does not consider techniques like spiccato or sautillé.

Bow Rule 3 (Mid-section): Control over pressure and speed is necessary; as pressure and speed increase, the volume will also increase.

Bow Rule 4 (Conclusion): The conclusion of a stroke should involve deceleration, aiding in controlling volume during bow changes while also generating lingering resonance.

When playing the violin, varying the volume can significantly enhance the emotional depth of the music. Human violinists achieve this by adjusting the contact force between the bow and the strings using their wrist, alongside controlling bowing speed to modulate volume changes in the performance. We also noticed instances of bouncing and detachment from the strings when moving the bow back, particularly when using short bow strokes. Furthermore, the process of short bowing noticeably resulted in weaker and more chaotic volume. Upon analysis, we identified that stopping the bow's movement during the backstroke led to bouncing due to the reactive force of the strings.

To regulate volume and prevent bouncing, VoiceTalk designs a pressure application mechanism (PAM) at the tip-side of the bowing structure (Figure 17), employing a spring to apply pressure to the pressure rod. The PAM applies direct force to the strings without affecting the motor's load, thus achieving superior force control. However, the friction between the pressure rod and the bow can lead to wear on the bow's surface over prolonged use. To address this, the pressure rod was transformed into a rotating roller covered with soft foam padding, preserving the bow's contact surface.

We create a project called "Calibration2" (Figure 18 (5)) to incorporate a microphone for recording violin sounds (Figure 18 (4)) and displaying volume data on time-series charts in the GUI dashboard (Figure 18 (6)). Through the

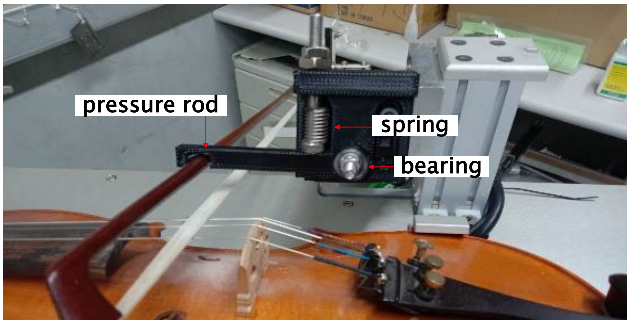


FIGURE 17. The pressure application mechanism at the end of the bowing mechanism.

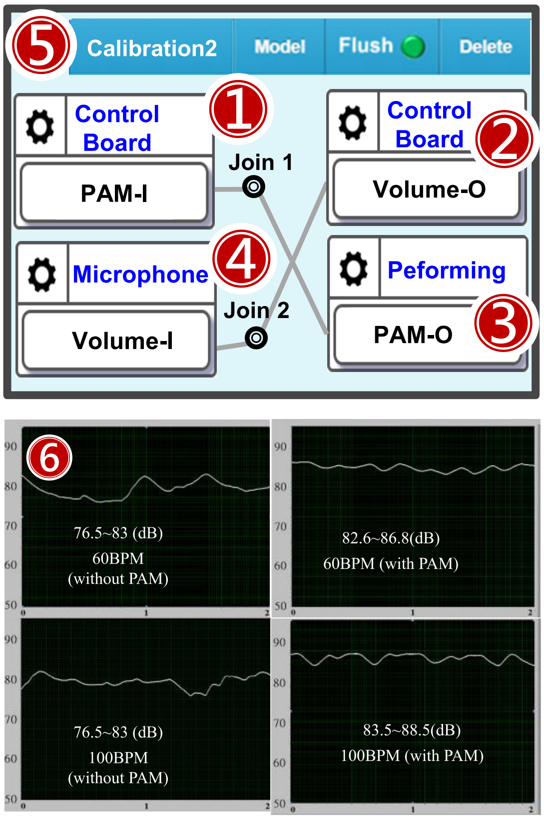


FIGURE 18. The effects of the pressure application mechanism on the volume.

analysis of the pressure application mechanism's effects on volume, we observed a significant increase in volume when using it, resulting in more stable sound variations. As the speed increased, so did the volume.

As shown in Figure 18 (6), both speed and pressure influence volume, with pressure having a more pronounced effect. During performances, achieving significant variations in speed is challenging due to the beat and bow length, making it difficult to create substantial volume fluctuations. PAM implemented in Bowing SA effectively addresses this issue.

ViolinTalk can utilize the microphone to control the PAM. When the Performing module (Figure 18 (3)) plays a song, the microphone records the sounds and sends the measured volumes to the Control Board (Figure 18 (2)) through Join 2. The Control Board (Figure 18 (1)) compares the received volumes with the target volumes to control the PAM in the String-Changing SA through Join 1.x

The timing of force application impacts the quality of performance. Generally, during the playing of single notes or between consecutive notes, one complete cycle of force control command must be executed. Bowing SA uses the four predefined bowing rules mentioned above to implement a force control algorithm and employs FPGA to construct a distributed force control system. The form of force control can maintain stable volume or create volume fluctuations. The main challenge lies in simulating human-like playing. In future work, Bowing SA aims to devise various control curves to create diverse volume fluctuations.

Bowing SA sets the speed of the servo motor (Figure 4 (1)) such that when playing the longest note in a song, the distance the bow moves cannot exceed the length of its hair. Use a 1/4 violin as an example (Figure 19). The hair length of the bow is 57.15cm.We set position 0 at the 6 cm from the frog of the bow, and the operation range is 47cm from position 0 toward the tip. We also provide a margin of 1.7cm for the bow to move from position 0 toward the frog and extra 2cm margin to move toward the tip. Suppose that ViolinTalk plays Pachelbel's Canon. The longest note is a half note, and the tempo is 66 beats per minute (BPM). The note speed v in beats per second is defined as

$$\text{note speed } v = \frac{60\alpha}{\text{tempo}} \tag{1}$$

where α is the speed ratio. The song slows down if $\alpha < 1$, and the song speeds up if $\alpha > 1$. In the normal case, $\alpha = 1$.

From Eq. (1), the period to play the longest note in Canon is 60(2/66)=1.818seconds. Therefore, the speed of the servo motor should be 47/1.818=25.853cm/s. Table 1 shows the

FIGURE 19. The operation range of the bow for a 1/4 Violin.

TABLE 1. The distances of bow movement for Pachelbel's Canon (tempo=66BPM).

Symbol	Note	Distance	Time
𝅗𝅥	half	47cm	1.818s
♩	quarter	26.5cm	0.909s
♪	eighth	13.25cm	0.454s
𝅘𝅥𝅯	sixteenth	6.625cm	0.2275s

distances of bow movement for the notes in Canon when the speed of the servo motor is 25.853cm/s.

VIII. CALIBRATION AND PERFORMING SAS

After completing the mechanical assembly and circuit configuration of an XY-type robot, ViolinTalk tests and calibrates the robot. During the X-component calibration, Calibration SA (Figure 7 (2)) assesses the performance of the bowing mechanism when the user presses the "B" button (Figure 9 (1)). In the process of calibrating the bow, Calibration SA instructs Bowing SA to move the linear slide (Figure 20 (1)) to the position of the sensor and set it as the zero point. It then identifies the front (L) and rear (R) extreme positions of the bow hair (linear slide) based on this zero point. Bowing SA saves the operation range of the bow to ensure there is no risk of collision.

When the user presses the "S-E" button (Figure 9 (2)), Calibration SA instructs String-Changing SA to perform a string-changing test to identify the rotational positioning point of the E string. String-Changing SA rotates the stepping motor (Figure 4 (1)) until the bow comes into contact with the E string. The SA then shields and defines this point as the origin of the string-changing reference point (zero point; Figure 20 (2)).

Then, Calibration SA instructs Bowing SA to perform a bowing test to ensure that the E string operation does not come into contact with other string positions. Specifically, Bowing SA tests the bowing effect on the E string and observes whether the bow moves forward and backward along the same angle without coming into contact with other strings (Figure 20 (3)). The user adjusts the angle of the bow (Figure 20 (4)) using the two "X-Adjust" buttons (Figure 9 (3)) and repeats the aforementioned steps to determine the angle that best suits the bowing operation. This process is referred to as X-axis adjustment.

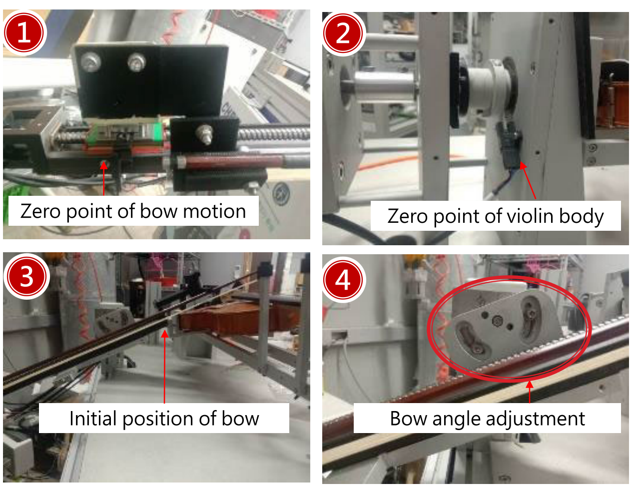


FIGURE 20. The Bow calibration ((1), (3), (4)) and string calibration (2).

If there is any contact with strings other than the E string, the user can further adjust the string angle using the two "Y-Adjust" buttons (Figure 9 (4)) until it operates correctly. The user can observe the contact of the bow with the E string through the video window (Figure 9 (5)). This process is referred to as Y-axis adjustment. The above calibration tasks can be achieved by a calibration project similar to Calibration1 (Figure 14 (5)) and Calibration2 (Figure 18 (6)), although the specific details are omitted.

String-Changing SA instructs the stepper motor to gradually rotate, sequentially determining the relative positions of the A, D, and G strings. This ensures that these positions won't interfere with other strings during the bowing process. Once verified, the relative positions of the strings are stored.

In the Y-component, the finger pressing mechanism is a crucial part for achieving accurate pitch during play. Calibration SA instructs Finger-Pressing SA to adjust the E, A, D, and G strings to their standard pitches while open. The user tests the fingers using buttons F1-F12 (Figure 9 (6)). Then, Calibration SA activates each mechanical finger and performs the bowing test. Specifically, Finger-Pressing SA moves the fingers to the correct pitch positions and securely tightens them to ensure acceptable pitch across different string positions and finger placements. Once this is achieved, the calibration of the violin robot is complete.

When the user selects a song from the song list (Figure 9 (7)), Performing SA reads the control parameters converted from the selected sheet music. Then, the user initializes the robot before playing the song. This is achieved by pressing the "Reset" button (Figure 9 (8)). In situations like collisions that require manual readjustment of the bow or string position, the user also clicks on "Reset" after turning off the servo of the robot (i.e., cutting off power to the motor). The "Reset" function works as follows: it releases the mechanical fingers, resets the string changing mechanism to its position at the first note in the song, and resets the bowing mechanism to its initial position for the first note.

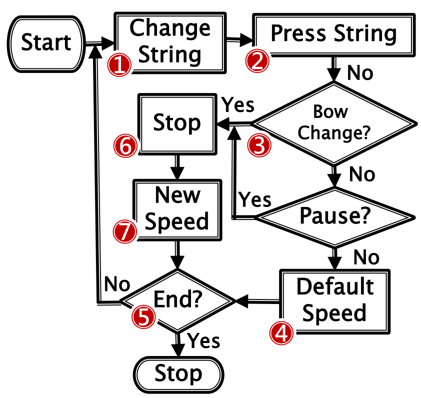


FIGURE 21. The flowchart of performing SA.

When the user presses the "Start" button (Figure 9 (9)), Performing SA plays the note based on the flowchart illustrated in Figure 21. In Step 1, Performing SA instructs String-Changing SA to move to the target string. In Step 2, Performing SA instructs Finger-Pressing SA to press the string with the target finger. In Step 3, Performing SA checks if a bow change or pause occurs. In normal situations, the answer is no, and Bowing SA is instructed to play at the default speed in Step 4. Step 5 checks if it is the end of the song. If it is not, control proceeds to Step 1; otherwise, the performance is terminated. If the answer is yes in Step 3, Step 6 instructs Bowing SA to stop and move the bow to the new position. Then, Bowing SA plays with a new speed in Step 7.

Step 7 merits further discussion. In situations where a note in the performance requires a bow change, and the tempo is fixed, adjustments can only be made to the speed ratio α of that specific note as defined in Eq. (1). This adjustment involves increasing the motor speed to compensate for any errors. The extent of these adjustments depends on the overall performance tempo and changes in the beat. When the overall tempo of the piece increases or the beats become shorter, the speed ratio must be increased accordingly.

In Figure 22, the desired displacement should ideally correspond to the area under the left-side blue curve ((A)-(B)-(C)-(D)), which is calculated as original bow velocity multiplied by the duration. However, in the real world, the effects of acceleration and deceleration lead to a reduction in the actual displacement, as shown by the area under the right-side yellow curve ((E)-(F)-(G)-(H)). As a result, the original velocity is multiplied by the speed ratio α to derive a corrective velocity that compensates for the displacement error ((F)-(G)-(I)-(J)). In the end, the combined result of actual displacement and corrective displacement equals the ideal displacement. Performing SA automatically computes the α value. The user can also manually adjusts the speed through the bow speed adjustment buttons (Figure 9 (10).

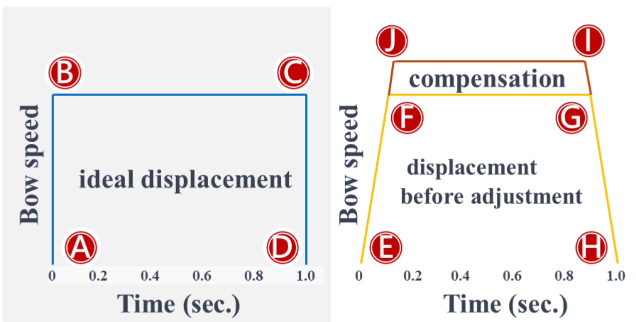


FIGURE 22. The Bow (motor) speed compensation.

IX. INTERACTION AMONG MULTIPLE MUSIC INSTRUMENTS

Through the ''Progress'' mechanism, musical instruments can easily collaborate in ViolinTalk. In this mechanism, an IoT device uses Progress-I to report its status and Progress-O to receive the statuses of other devices. In this case, a violin robot serves not only as an actuator for receiving instructions but also as a means to report the progress of its performance, specifically indicating which note it has reached. For example, in the ViolinTalk4 project, Canon in D violin duet is configured with the sensor/control functions of two IoT devices (Figure 23 (1) and (4)) and the actuator functions of three IoT devices (Figure 23 (2), (3), and (5)). Through Join 1, the Control Board employs Song-I to enable Song-O for both Violin1 (leader) and Violin2 (follower) to play Pachelbel's Canon. Then, the Control Board utilizes Play-I1 to instruct Violin1 to begin playing through Join 2. While playing, Violin1's Progress-I reports its status to the Control Board through Join 3. Once the Control Board detects that Violin1 has played up to the rest or any section of the song, it uses Play-I2 to initiate Violin2's follow-up performance through Join 4.

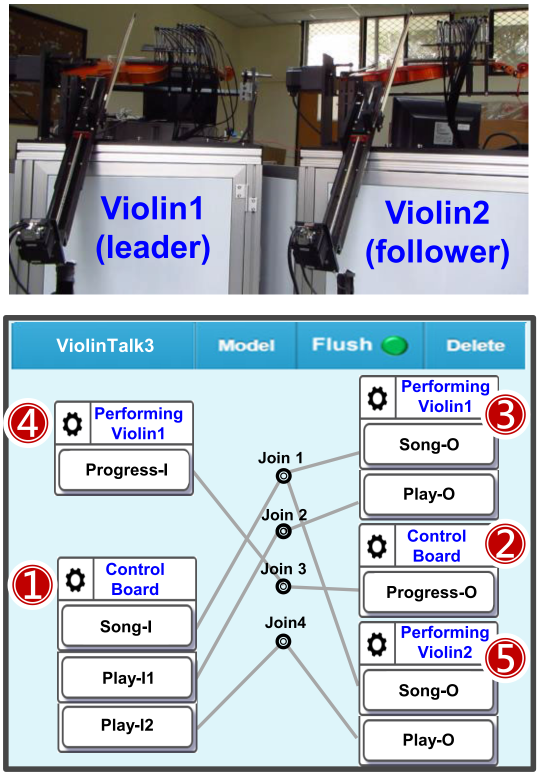


FIGURE 23. The canon in D violin duet.

All music instruments report their progress to the Control Board through Join 4. By analyzing the statuses of the music instruments, the Control Board issues next instructions to each instrument individually through Join 3. We have developed the first prototype for the Canon Quartet and are still in the fine-tuning stage.

ViolinTalk integrates multiple robots into a string orchestra, capable of performing standard chamber music. In the ''Canon in D violin duet'' example, the Control Board only sends one synchronization message to the violins. However, it is more common that musical instruments need to be synchronized several times during their performance. For instance, the music conductor may want to adjust the volume or the speed to express specific emotions. Figure 24 illustrates the Canon Quartet by ViolinTalk with frequent synchronization. The Canon Quartet includes two violins, one viola, and one cello. In the ViolinTalk configuration, every

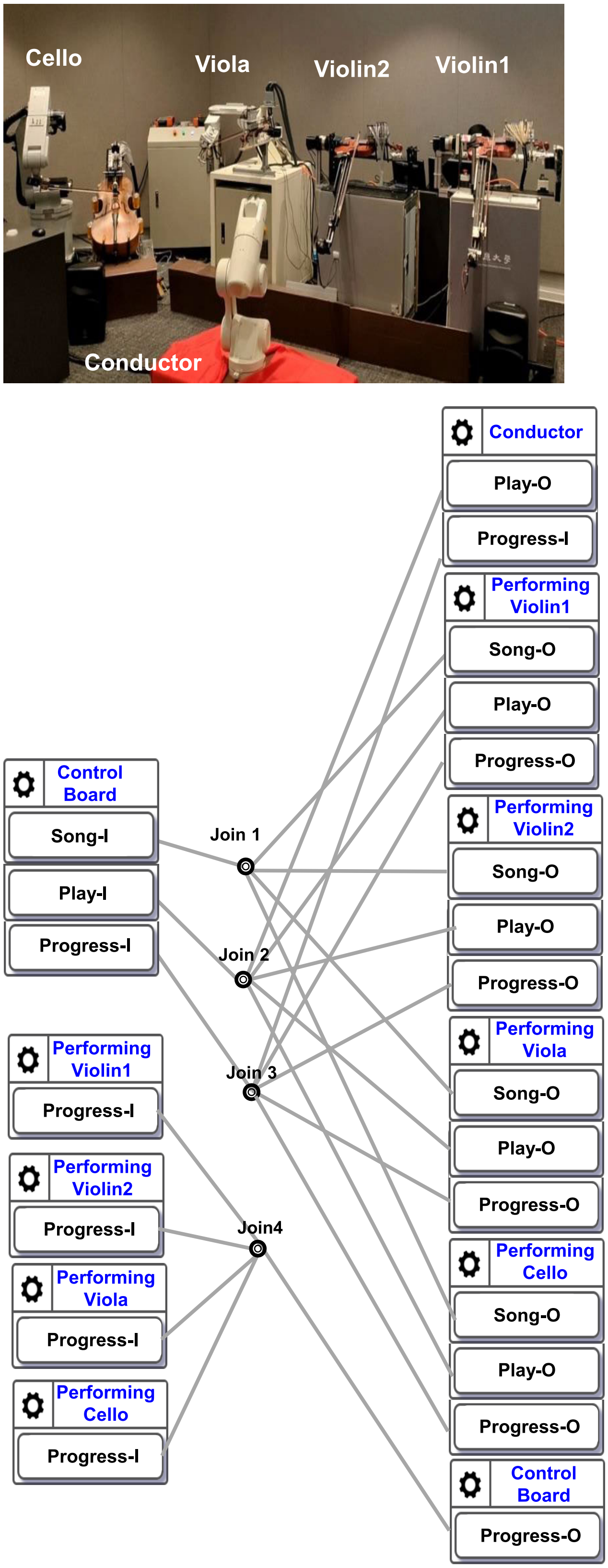


FIGURE 24. The canon quartet by ViolinTalk.

musical instrument (an IoT device) has one control (sensor) Progress-I and three actuators: Song-O, Play-O, and Progress-O. The Control Board also connects to a conductor robot. When the human conductor controls the musical instruments through the Control Board, it gives the same instructions to the conductor robot through Joins 1 and 2, and the conductor robot makes gestures to engage with the audience.

An important issue is the communication delays between the Control Board and the music instruments. Following CATtalk [5], the ViolinTalk server and the DA of each music instrument have installed timers that synchronize with the Network Time Protocol (NTP). This allows ViolinTalk to measure the one-way delay between the instrument and the ViolinTalk server.

We consider both the 4G and WiFi connections. Let t_{4G} be the remote 4G delay, and t_{WiFi} be the local WiFi delay. Figure 25 illustrates the histograms for t_{WiFi} and t_{4G} [5]. The average delay for local WiFi communication is $\mathrm{E}[t_{\mathrm{WiFi}}] = 9.96$ milliseconds (ms), which is less than 0.01 seconds, and the variance is $\mathrm{V}[t_{\mathrm{WiFi}}] = 0.03\ \mathrm{E}[t_{\mathrm{WiFi}}]^2$. The average delay for remote 4G communication is $\mathrm{E}[t_{4G}] = 47.76$ ms, which is less than 0.05 seconds, and the variance is $\mathrm{V}[t_{4G}] = 0.076\ \mathrm{E}[t_{4G}]^2$.

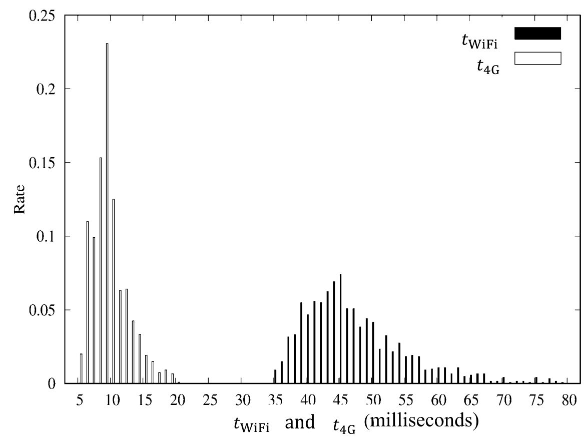


FIGURE 25. The histograms of local t_{WiFi} and remote t_{4G}.

X. CONCLUSION AND FUTURE WORK

We began developing violin robots in 2005, starting with the early XY-type violin robots and expanding to integrate various types of robots into a robot string orchestra. Drawing upon our experience with violin robots, this paper introduces ViolinTalk, a system for controlling the XY-type robot as an Internet of Things (IoT) device. By connecting the XY-type robot to an IoT development platform, ViolinTalk offers a powerful environment for enhancing the performance of the XY-type robot through calibration and testing.

ViolinTalk has improved the software modules of XY-type violin robots, enhancing their performance. The specific contributions include: (1) Developing two control modes for the bowing servo motors, including displacement-based control and time-based control; (2) Utilizing bowing speed control to create variations in volume and tone, enriching the performance effects; (3) Analyzing errors in playing speed and establishing correction parameters for subsequent use; (4) Analyzing the robot's mechanism to make it more stable and durable. ViolinTalk utilizes IoT technology, making it easy to connect different instruments for ensemble performances. We have already integrated multiple robots into a string orchestra, consisting of two violins, one viola, and one cello, capable of performing standard chamber music. This development has received a total of 8 patents listed in Table 2.

TABLE 2. The ViolinTalk robot patents.

No.	Patent Type	Title	Patent Number
1	Invention	Humanoid violin-playing robot	I321484
2	Invention	Automated mechanical apparatus for playing the violin	I323877
3	Invention	Multi-degree freedom mechanical finger with play function	I500022
4	Invention	Automatic performance apparatus	I680451
5	Utility Model	Palm-mimic mechanical music instrument play device	M424184
6	Utility Model	Both hands playing simulation device	M430690
7	Utility Model	Automatic cello playing device	M481474
8	Utility Model	Kneading system and fiddle mechanical musical apparatus	M482142

The ViolinTalk calibration projects proposed in this paper allow users to remotely calibrate the robot. In the future, we plan to design a semi-automatic calibration system that utilizes accelerometers (Sensors-I in Figure 26 (1)) similar to Calibration1, microphones (Volume-I in Figure 26 (2)) similar to Calibration2, and the addition of image detection (ImageAI-I in Figure 26 (3)). The current version of calibration projects require users to take action by pressing the buttons on the Control Board (Bow-I, String-I, and Finger-I in Figure 26 (4)). Through the cameras installed by ViolinTalk (Figure 26 (5)), we can analyze the captured images (Figure 26 (6)). A deep learning AI model such as Yolo-v8 (Figure 26 (7)) detects object movement displacement. Subsequently, ImageAI-I sends the detection results to ImageAI-O (Figure 26 (8)). Based on these detections, the Control Board can make decisions for the next action without human involvement. The AI models are packaged as IoT devices and can be easily integrated into ViolinTalk. Further details can be found in Section III in [21].

In order to make the XY-type violin robots perform like humans in terms of their playing, there is still significant room for improvement in both control technology and structural design. In the future, using the ViolinTalk development environment, we will challenge simulating human-like playing. We have proposed four rules for bowing, and in the future, will implement these rules in ViolinTalk, and devise various control curves to create diverse volume fluctuations.

Furthermore, ViolinTalk will continue to integrate other music robots to expand instrument ensemble capabilities. Table 3 lists the types of instrument robots that can be created and accommodated within ViolinTalk.

The single violin performance of Vivaldi's "Storm" in "The Four Seasons" can be found in [22]. The "Canon in D, String Ensemble" in Figure 23 can be found in [23]. The "Canon Quartet" in Figure 24 can be found in [24].

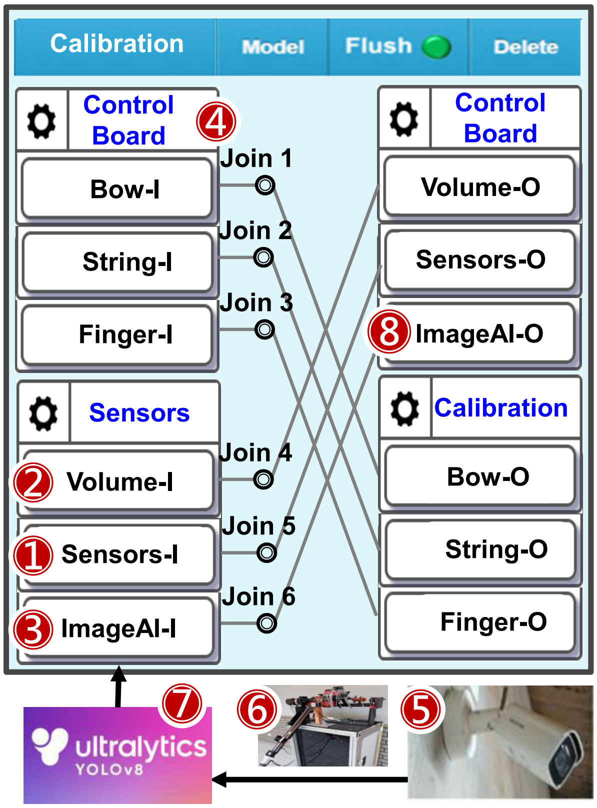


FIGURE 26. The template for semi-automatic calibration.

TABLE 3. The types of instrument robots that can be created and accommodated within ViolinTalk.

Instrument	Robot version	Technical difficulty	Portability	Mechanical drive method
Violin	Yes	High	Low	Pneumatic Electric
Viola	Yes	High	Low	Pneumatic Electric
Cello	Yes	High	Low	Pneumatic Electric
Mandolin	No	Medium	Low	Pneumatic Electric
Guitar	Yes	Medium	High	Electric
Electric guitar	Yes	Medium	High	Electric
Ukulele	No	Low	Low	Electric
Piano	Yes	Medium	High	Pneumatic Electric
Electronic keyboard	Yes	Medium	Low	Electric
Harp	No	Medium	High	Electric
Yangqin	No	Medium	Low	Electric
Erhu	No	High	Low	Electric
Gehu	No	High	High	Electric
Guzheng	No	Medium	Low	Electric
Lute(Pipa)	No	Medium	Low	Electric

ACKNOWLEDGMENT

The authors would like to thank the team members over the years, including Jing-Yun Hu, Chun-Han Wang, Yu-Cheng Lin, Hsin-Hsien Li, Yi-Hsiang Chen, Pei-De Lin, Hung-Wei Huang, Chiu-Yen Yu, Wun-Hong Li, and others, for making this research project successfully completed.

REFERENCES

[1] X. Wang, "Tennis robot design via Internet of Things and deep learning," *IEEE Access*, vol. 9, pp. 127460–127470, 2021, doi: 10.1109/ACCESS.2021.3111706.

[2] P. R. Pinheiro, P. G. C. D. Pinheiro, R. H. Filho, J. P. A. Barrozo, J. J. P. C. Rodrigues, L. I. C. C. Pinheiro, and M. L. D. Pereira, "Integration of the mobile robot and Internet of Things to monitor older people," *IEEE Access*, vol. 8, pp. 138922–138933, 2020, doi: 10.1109/ACCESS.2020.3009167.

[3] *It's Robot Versus Human as Shimon Performs Real-Time Rap Battles*. Accessed: Sep. 2, 2023. [Online]. Available: https://spectrum.ieee.org/robot-versus-human-shimon-real-time-rap-battles

[4] Toyota. (2010). *Toyota Mobiro New Robot*. [Online]. Available: http://www.dailymotion.com/swf/x3ojjn

[5] Y.-B. Lin, H. Luo, and C.-C. Liao, "CATtalk: An IoT-based interactive art development platform," *IEEE Access*, vol. 10, pp. 127754–127769, 2022, doi: 10.1109/ACCESS.2022.3227093.

[6] E. V. Jansson, "On the acoustics of musical instruments," in *Proc. MUSIC ROOM Acoust.*, vol. 17. Stockholm, Sweden: Royal Swedish Academy of Music, 1977, pp. 82–113.

[7] C. M. Hutchins and V. Benade, *Research Papers in Violin Acoustics*. New York, NY, USA: Acoustical Society of America, 1996, pp. 1975–1993.

[8] Y. Ando and D.-Y. Zheng, *Acoustics of Musical Instruments*. Chicago, IL, USA: Lion Publishing, 1989.

[9] J. Kim, F. Cordier, and N. Magnenat-Thalmann, "Neural network-based violinist's hand animation," in *Proc. Comput. Graph. Int.*, Jun. 2000, pp. 37–41.

[10] A. Askenfelt, "Measurement of the bowing parameters in violin playing. II: Bow–bridge distance, dynamic range, and limits of bow force," *J. Acoust. Soc. Amer.*, vol. 91, no. 6, pp. 3540–3550, 1992.

[11] K. Shibuya, K. Kosuga, and H. Fukuhara, "Bright and dark timbre expressions with sound pressure and tempo variations by violin-playing robot," in *Proc. 29th IEEE Int. Conf. Robot Human Interact. Commun. (RO-MAN)*, Aug. 2020, pp. 482–487.

[12] N. C. Tamer, Y. Özer, M. Müller, and X. Serra, "TAPE: An end-to-end timbre-aware pitch estimator," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Rhodes Island, Greece, Jun. 2023, pp. 1–5, doi: 10.1109/ICASSP49357.2023.10096762.

[13] K. Fujisaki and S. Katsura, "In-tool motion sensing for evaluation of violin performance," *IEEJ J. Ind. Appl.*, vol. 11, no. 2, pp. 291–298, 2021.

[14] Y. C. Chia, C. H. Lee, B. Y. Hong, and B. K. LIM, "RoboFiddler," Dept. Mech. Eng., Univ. Adelaide, Adelaide, SA, Australia, Level IV Project, Final Rep., 2006.

[15] H. H. Huang, "Automatic violin player," in *Proc. 10th World Congr. Intell. Control Automat.*, 2012.

[16] S. Goldstein. (2015). *Seth Goldstein Kinetic Sculpture Machines*. [Online]. Available: http://www.Sethgoldstein-kineticsculptures.com/

[17] W. Jo, H. Park, B. Lee, and D. Kim, "A study on improving sound quality of violin playing robot," in *Proc. 6th Int. Conf. Autom., Robot. Appl. (ICARA)*, Queenstown, New Zealand, Feb. 2015, pp. 185–191, doi: 10.1109/ICARA.2015.7081145.

[18] J. Yura, M. Oyun-Erdene, B.-E. Byambasuren, and D. H. Kim, "Modeling of violin playing robot arm with MATLAB/SIMULINK," in *Proc. 4th Int. Conf. Robot Intell. Technol. Appl.*, 2017.

[19] A. J. Meza-Perez and J. A. Pamanes, "Motion simulation of a violin-player robotic left hand," in *Proc. 20th Congreso Mexicano Robótica (COMRob)*, Ensenada, Mexico, Sep. 2018, pp. 1–6, doi: 10.1109/COMROB.2018.8689421.

[20] Advantech. *PCI-1245: PCI Card for 4-Axis Stepper and Servo Motor-Control Based on DSP*. Accessed: Sep. 2, 2023. [Online]. Available: https://www.ni.com/zh-tw/support/model.cdaq-9178.html

[21] Y.-B. Lin, C.-Y. Liu, W.-L. Chen, C.-H. Chang, F.-L. Ng, K. Yang, and J. Hsung, "IoT-based strawberry disease detection with wall-mounted monitoring cameras," *IEEE Internet Things J.*, early access, Jun. 22, 2023, doi: 10.1109/JIOT.2023.3288603.

[22] *'Storm' in 'The Four Seasons'*. Accessed: Sep. 2, 2023. [Online]. Available: https://youtu.be/Q7oyNYQiruY

[23] *Canon in D Violin Duet*. Accessed: Sep. 2, 2023. [Online]. Available: https://youtu.be/Q1P8_9HP1WI

[24] *Canon Quartet*. Accessed: Sep. 2, 2023. [Online]. Available: https://youtu.be/FGm8b-t0fHA

HSING-HSIN HUANG received the Ph.D. degree in mechanical engineering from the Florida Institute of Technology, Melbourne, FL, USA, in 1992. From 1992 to 2018, he was with the Minghsin University of Science and Technology (MUST), where he became the Vice President, in 2016. In 2018, he joined the Department of Mechanical Engineering, Chung Yuan Christian University (CYCU), as a Professor. His major is automation control. He is also the Host of the Intelligent Automation Industry-Academic Technology Alliance, National Science Council, committed to promoting the development of automation technology in the industry. He has presided over a number of industry-university cooperation projects and holds dozens of patents.

YI-BING LIN (Fellow, IEEE) received the Ph.D. degree in computer science from the University of Washington, Seattle, WA, USA, in 1990. From 1990 to 1995, he was a Research Scientist with Bellcore. Then, he joined National Chiao Tung University (NCTU), where he became the Senior Vice President, in 2011. From 2014 to 2016, he was the Deputy Minister of the Ministry of Science and Technology, Taiwan. He is currently a Winbond Chair Professor with National Yang Ming Chiao Tung University (NYCU), a Chair Professor with National Cheng Kung University and China Medical University, and an Adjunt Research Fellow of Academia Sinica. He is the coauthor of the books *Wireless and Mobile Network Architecture* (Wiley, 2001), *Wireless and Mobile All-IP Networks* (John Wiley, 2005), and *Charging for Mobile All-IP Telecommunications* (Wiley, 2008). He is a fellow of AAAS, ACM, and IET.