The Wand Chooses the IMU - Open Source Hardware for Synchronising Wearables using Magnetometer

Yuxuan Han*
Thomas J Gilbert*
yuxuan.han.22@ucl.ac.uk
thomas.j.gilbert@ucl.ac.uk
University College London
London, UK

Xiangyi Tan
University College London
London, UK
xiangyi.tan.23@ucl.ac.uk

Jamie A Ward
Goldsmiths, University of London
London, UK
jamie@jamieward.net

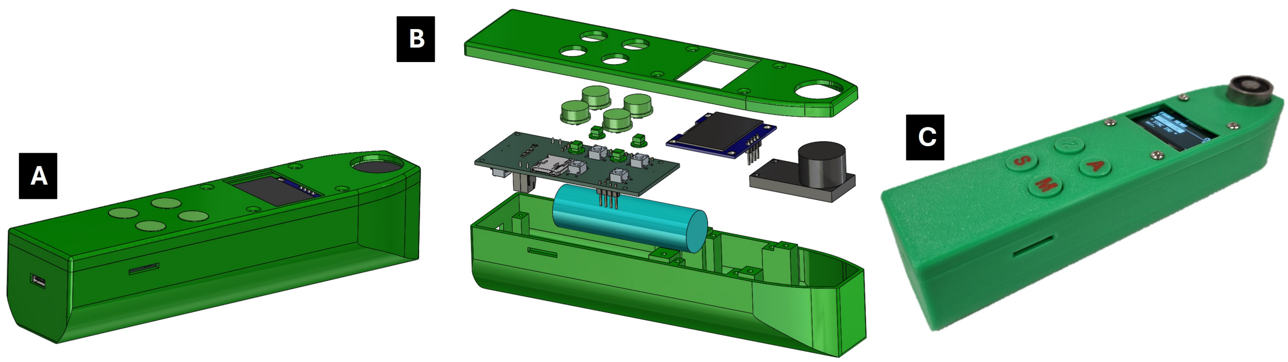

Figure 1: (A) Overview of the 3D CAD design of the case, (B) exploded view, (C) final fully-assembled device

Abstract

This paper presents the Synchronisation Wand, an open-source hardware solution which can be used to synchronise multiple inertial measurement unit sensors (IMU) using their onboard magnetometers. The wand is based on commercial off-the-shelf components with standard 4-layer PCB manufacturing and self-service assembly. The case can be 3D printed using any Fused Filament Fabrication (FFF) based printer. The system combines an ESP32-S3 micro-controller unit with an electromagnetic (EM) generator to create an encoded EM event which can be used to synchronise multiple IMUs. The hardware includes an onboard IMU allowing the user to track the motion of the wand as well as perform kinetic synchronising events. The device also includes an OLED display and 4 configurable tactile switches to enhance the usability of the system. We demonstrate the device's capability to generate an encoded EM pulse which can be used to reduce the maximum synchronisation error between two IMUs down to 10ms.

CCS Concepts

• **Hardware** → **PCB design and layout**; • **Human-centered computing** → *Ubiquitous and mobile computing systems and tools.*

*Both authors contributed equally to this research.

UbiComp Companion '24, October 5–9, 2024, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-1058-2/24/10
https://doi.org/10.1145/3675094.3678485

Keywords

wearable, synchronisation, imu, wand, open source

ACM Reference Format:
Yuxuan Han, Thomas J Gilbert, Xiangyi Tan, and Jamie A Ward. 2024. The Wand Chooses the IMU - Open Source Hardware for Synchronising Wearables using Magnetometer. In *Companion of the 2024 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp Companion '24), October 5–9, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3675094.3678485

1 Introduction and Background

Many devices in the domains of ubiquitous and wearable computing include onboard IMUs. Applications of IMUs stretch from the wider topics of human activity recognition [1] and multi-sensor fusion [14], to studies measuring social interaction and engagement in real-world settings, e.g. [8]. Many applications require precise synchronisation between separate IMU devices. Most commercial IMU devices include an onboard real-time clock (RTC) which can be used to synchronise multiple devices to a global clock. Unfortunately, standard RTCs tend to drift over time, leading to significant discrepancies between device clocks during extended recordings. Consequently, relying solely on RTC-based synchronization is infeasible for longer experiments, such as in [11], where recordings of multiple autistic children and actors interacting over several hours are analyzed to uncover fine-grained moments of motion synchrony.

Efforts have been made to overcome this desynchronisation issue. The most commonly used technique to synchronise multiple IMUs is to capture a common signal across different sensors and sensor types to facilitate temporal alignment. Kinetic events are most commonly used, requiring the experimenter or participant to perform a predefined movement, such as clapping, hitting the table [6, 9, 10], or even tapping the ear [5]. Synchronizing events can also be collected from various sensors, including sound and light sensors [2]. LED-sourced light has been used to update clock signals in IoT devices [4], and ECG sensors have been employed to synchronize across wearable devices [12, 13].

Recently, EM synchronising events have been explored, with [7] using inductors to create an EM event that is captured by Shimmer3 magnetometers. In our own work, upon which the current study is based, we introduced a technique to synchronise multiple IMU devices to a single RTC using EM events [3]. A problem of EM-based events is the low sample rate associated with magnetometers results in high maximum synchronisation errors. This problem has been addressed in our previous work where we introduced an encoded EM event which achieves sub-sample synchronising accuracies.

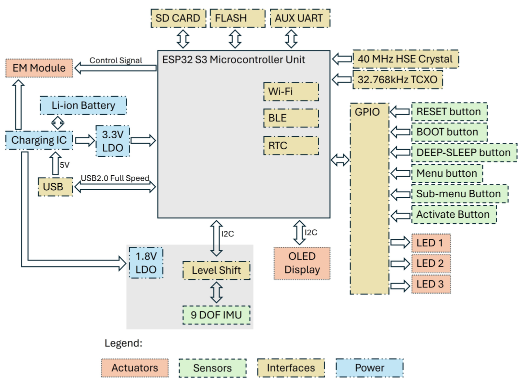

Figure 2: Overview of the Synchronising Wand architecture

2 Hardware

The Synchronisation Wand is based on our previously published description of a portable-EM pulse generator (EMPG) in [3]. Additional features of this version include the ability to generate an EM event as well as record a kinetic event. Figure 2 shows the system architecture of the wand. All resources required to make the device can be found in our repository [1].

2.1 Electronics

2.1.1 Component choice. Smaller ICs are used to reduce the PCB size and to make the device more portable. Newer models of components have been prioritised for better performance, availability and a reduced likelihood of being discontinued in the coming years.

2.1.2 PCB. A 4-layer PCB with a stack-up of Signal-GND-GND-Signal is used. This stack-up ensures each trace has a continuous reference ground to improve signal performance, which is especially vital for the USB 2.0 full-speed differential traces and the 2.4 GHz antenna. As required by the USB 2.0 specification, the full-speed differential pair is designed with 90Ω characteristic impedance. The antenna trace is designed to have 50Ω characteristic impedance for good RF performance.

[1]https://github.com/YuxuanHan0326/Synchronisation-Wand

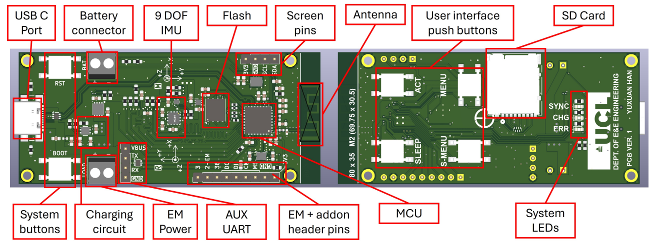

Figure 3: Top (left) and bottom (right) sides of 3D rendered, populated PCB showing the main components

2.1.3 Micro-controller Unit. The ESP32-S3 is used due to its low cost and high performance compared to other WiFi micro-controller units on the market. The ESP32-S3 features a dual-core 240 MHz CPU, offering low latency and high precision timing, which is crucial for this device. Its WiFi capability enables easy and accurate RTC synchronization. The SDMMC driver supports up to 8-bit SD at 40MHz for faster data logging and can be routed through the GPIO MUX for a simpler PCB layout.

2.1.4 Memory. The device includes an SD card and reader with support for a 4-bit SD protocol at 40 MHz, which is used to store timing information and IMU data. The PCB also includes a 256 Mbit flash (GD25Q256EWIGR), part of which can be configured as an extra drive to store data. This is used as it is the largest QSPI flash supported by the ESP32-S3.

2.1.5 General Purpose I/O. The design has 4 user-programmable tactile switches, acting as "Sleep", "Activate", "Menu" and "Submenu", with three user-programmable LEDs to indicate system charging, synchronisation, and error. The SD card detect-pin is used to enable hot plugging and unplugging of the SD. A GPIO is used to activate the EM module. The design also exposes six multi-function pins on the pin headers that can be freely configured and used as GPIOs.

2.1.6 EM Generator. The EMPG (Grove Electromagnet [2]), previously used in [3], was selected due to its low price and power consumption, wide availability, and requiring only one GPIO pin.

2.1.7 Power architecture. The power system of the device contains 4 distinct parts: battery, charger IC, 3.3V LDO and 1.8V LDO. The battery is a Li-ion 3.7 V, 2.6 Ah 18650 module from RS Pro. It includes an integrated battery protection IC and is more cost-effective than a capacity-equivalent LiPo battery.

The battery is charged by sourcing current from the USB-C connection via a charger IC. The charger selected is BQ25185DLHR, a newly released chip from Texas Instruments. It has a power path selection function, which enables the smooth switch of power source between battery and USB power (e.g. even if the battery is depleted, the user can connect it to a power bank and use it immediately while charging the battery at the same time). The chip also has 9 different integrated fault protections, which increase the safety of the device. It supports a wide variety of battery chemistry, configurable voltage, charging current and input current limit. This is adjustable simply by changing specific external resistors.

[2]Seeedstudio: https://wiki.seeedstudio.com/Grove-Electromagnet/

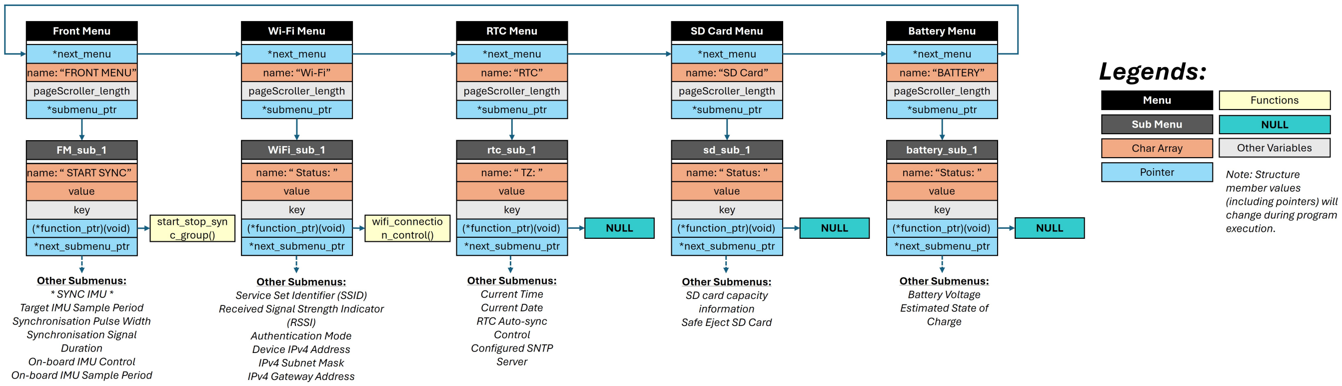

Figure 4: Overview of the firmware's data structure and interactive menu design

2.2 3D Printed Case

The CAD model for the case of the wand is composed of two parts: the upper cover and the lower shell, see figure 1. The PCB is secured to the lower shell using four M2 self-tapping screws. The SD card slot and USB-C port are precisely aligned with the openings on the side walls of the shell. The battery is housed in a compartment beneath the PCB. The OLED screen and EMPG are attached to the screw holes of the lower shell using #4 self-tapping screws. The four button caps are aligned with the buttons on the PCB and secured by the slots in the upper cover. The OLED screen and EMPG are visible through openings in the upper cover, which is then fastened to the lower shell with four #4 self-tapping screws.

3 Firmware

The firmware of the Wand is implemented in embedded C using the *Espressif* IoT Development Framework (ESP-IDF), due to its high flexibility and advanced feature support for ESP32S3. FreeRTOS is used to increase the robustness of the firmware and achieve more efficient resource utilisation. The firmware displays information and control options via the interactive menus on the OLED screen and can be user-controlled via tactile switches. For accurate system time, the device's RTC is synchronized using SNTP on configurable NTP servers via Wi-Fi. Synchronisation data and onboard IMU measurements are timestamped and written onto the SD card.

3.1 Interactive Menu

3.1.1 Display. A 128x64 OLED screen displays the interactive menu and is driven via i2c on a 400 kHz clock. The screen control and content display functionalities were implemented by porting the U8g2 Library [3] to the ESP-IDF development environment.

3.1.2 Data Structure. The data structure of menus and submenus are shown in *Figure 4*. A detailed explanation of each type of struct member can be found in *Figure 5*. Every time the menu tactile switch is triggered, the pointer **current_menu* will propagate to the address stored in the member **next_menu*, thus achieving the switch of menus. The switch of submenus can be achieved similarly, when the submenu tactile switch is triggered, the pointer **current_submenu* will propagate to the address stored in the member **next_submenu_ptr*, until it comes to a NULL, and points back to the address **submenu_ptr* under the current menu structure. When the activate tactile switch is triggered, if the function pointer in the current submenu structure is not NULL, it will be executed. The addresses of three submenus shown on the screen will be stored in pointers **sub1_disp* **sub2_disp* and **sub3_disp* correspondingly.

3.1.3 Screen Refresh Logic. The main screen refresh task is implemented using FreeRTOS notification. In order to reduce the CPU and i2c bus usage, the task will only be unblocked when a notification is received. If certain conditions are met, the screen will be refreshed. This will be explained below. Whenever an event causes a change to any value or name or icon shown on the screen, to ensure it is correctly updated on the screen, a notification to the screen refresh task has to be sent manually, with the proper event type identifier bit being set. Whenever the screen refresh task receives the notification, it will compare the received identifier with the key structure member values of three submenus currently displaying on the screen. If the following logic operation is non-zero: *received identifier & ((current submenu 1 key) | (current submenu 2 key) | (current submenu 3 key))*, then this indicates that there is a change on the screen, and a refresh event will be triggered.

3.2 Internal functionality

3.2.1 RTC Synchronisation. The device's RTC is automatically synchronised via SNTP service from pre-configured NTP servers periodically, provided that the Wi-Fi connection is successfully established. To ensure the continuity of the generated time stamp, the RTC synchronisation function is automatically disabled when the onboard IMU is on or when the user is synchronising the IMUs.

3.2.2 Synchronisation Signal. The system automatically calculates the parameters and determines the most suitable synchronisation signal according to the approach presented in[3]. In summary, the method works by transmitting an initial square pulse with width w, and $s/a + 1$ further pulses with a width of $w + a$, where a is the desired accuracy, and s is the sample period of the IMU device. The signal is then generated using an interrupt from the 54-bit general-purpose on-chip timer. Experimental results show almost zero inaccuracy of the generated signal compared to the theory.

[3] U8g2 Library: https://github.com/olikraus/u8g2

3.2.3 Deep Sleep. The device goes into deep sleep mode by pressing the *Sleep* switch for 3 seconds. This is implemented using the deep sleep function on the ESP32-S3. Only RTC and RTC peripherals will be powered on during deep sleep, which significantly reduces the device's power consumption. The device is woken when the user presses *Sleep* again.

3.2.4 SD Card. Data transfer from the SD card is achieved using the ESP-IDF's SDMMC driver and FAT filesystem support. Synchronisation data and onboard IMU data are logged automatically onto the SD card in CSV format.

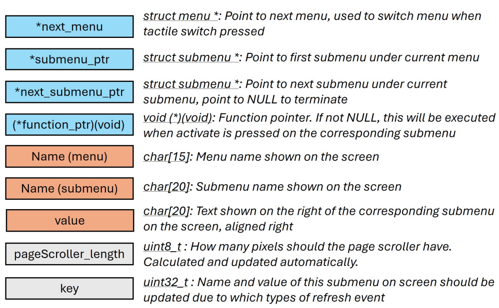

Figure 5: Explanation of the structure members in Figure 4

4 Application

To demonstrate the effectiveness of the Synchronisation Wand we have repeated the experiment described in [3]. Specifically, a commercial IMU device (MetaMotionR from Mbientlab [4]) is configured such that its accelerometer and magnetometer log data with sampling rates of 100Hz and 25 respectively. This is then placed with the wand on a table and a kinetic event is created by hitting the table. The wand is then configured with a=10 ms, brought within 11cm of the IMU device and activated creating an electromagnetic pulse event. Accelerometer and electromagnetic pulse data are collected from the wand's micro-SD card, while the accelerometer and magnetometer data from the IMUs are uploaded to a .csv file using the Mbientlab's MetaBase application.

Figure 6 shows an example of the magnetometer data recorded by one of the IMU devices synchronised by the Synchronisation Wand under the configuration of a= 10ms. The blue line shows the continuous magnetometer signal, with red dots marking IMU sampled data points. The signal pattern shows an initial pulse followed by a succession of synchronization pulses, creating a repeating up-and-down waveform. Note that the 4th pulse has 9 sample points while the other pulses have 8. This indicates that the EM event is offset by up to 40ms (4*10ms) and must be adjusted accordingly. Figure 7 shows the corresponding kinetic event with the 40ms adjustment.

5 Conclusion & future work

We present our Synchronisation Wand, an open hardware solution for synchronising multiple IMUs. This new device allows users to create both electromagnetic and kinetics events with timestamps logged by an RTC which is synchronized via a WiFi connection.

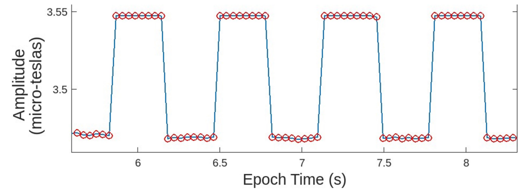

Figure 6: EM pulses sampled by an IMU. Note: the first 3 pulses have 8 samples, with the 4th having 9-samples

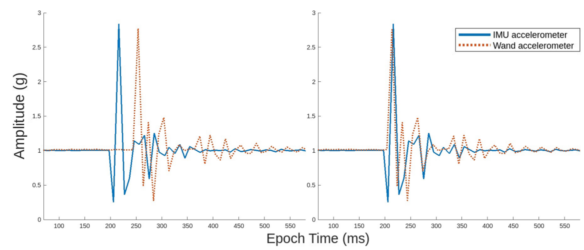

Figure 7: Kinetic event on the wand and the IMU device: (Left) before sync.; (Right) after 40 ms adjustment by the EM event.

These events can be recorded by commercial IMU devices using their onboard magnetometers and accelerometers. The wand also includes 4 programmable tactile switches and an OLED display allowing users to synchronise IMU devices during motion capture experiments without having to remove them from the participants. Featuring a large battery and SD card port users are able to use the wand in experiments lasting 40+ hours. We have validated the wand by repeating the main experiment described in [3].

Future versions of the Synchronisation Wand will include the implementation of BLE and a phone application allowing users to easily configure the settings and have a better overview of system status. WiFi aspects can be further developed to support more functionalities. For example, real-time data logging to the cloud, wireless file transfer from SD card to computer, real-time data visualisation, Over-The-Air (OTA) firmware update, etc. USB 2.0 full speed can be further developed for file transfer. The current version of the Synchronisation Wand uses the same electromagnetic generator module and power described in [3], which is limited to an active range of 11cm. A future version will look into expanding this active range.

Acknowledgments

This project would not have been possible without the generous financial support of UCL's Fleming Society.

References

[1] Manuel Abbas and Régine Le Bouquin Jeannès. 2022. Acceleration-based gait analysis for frailty assessment in older adults. *Pattern Recognit. Lett.* 161 (Sept. 2022), 45–51.

[2] David Bannach, Oliver Amft, and Paul Lukowicz. 2009. Automatic Event-Based Synchronization of Multimodal Data Streams from Wearable and Ambient Sensors. In *Smart Sensing and Context (Lecture Notes in Computer Science)*, Payam Barnaghi, Klaus Moessner, Mirko Presser, and Stefan Meissner (Eds.). Springer, Berlin, Heidelberg, 135–148. https://doi.org/10.1007/978-3-642-04471-7_11

[3] Thomas J Gilbert, Zexiao Lin, Sally Day, Antonia F de C Hamilton, and Jamie A Ward. 2024. A magnetometer-based method for in-situ syncing of wearable

[4] https://mbientlab.com

inertial measurement units. *Front. Comput. Sci.* 6 (April 2024).

[4] XiangFa Guo, Mobashir Mohammad, Sudipta Saha, Mun Choon Chan, Seth Gilbert, and Derek Leong. 2016. PSync: Visible light-based time synchronization for Internet of Things (IoT). In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. 1–9. https://doi.org/10.1109/INFOCOM.2016.7524358

[5] Alexander Hoelzemann, Henry Odoemelem, and Kristof Van Laerhoven. 2019. Using an in-ear wearable to annotate activity data across multiple inertial sensors. In *Proceedings of the 1st International Workshop on Earable Computing*. 14–19.

[6] Thomas Plotz, Chen Chen, Nils Y. Hammerla, and Gregory D. Abowd. 2012. Automatic Synchronization of Wearable Sensors and Video-Cameras for Ground Truth Annotation – A Practical Approach. In *2012 16th International Symposium on Wearable Computers*. 100–103. https://doi.org/10.1109/ISWC.2012.15 ISSN: 2376-8541.

[7] Andreas Spilz and Michael Munz. 2023. Synchronisation of wearable inertial measurement units based on magnetometer data. *Biomedical Engineering/Biomedizinische Technik* 0 (2023).

[8] Yanke Sun, Dwaynica A Greaves, Guido Orgs, Antonia F de C. Hamilton, Sally Day, and Jamie A Ward. 2023. Using Wearable Sensors to Measure Interpersonal Synchrony in Actors and Audience Members During a Live Theatre Performance. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 1 (2023), 1–29.

[9] Chaofan Wang, Zhanna Sarsenbayeva, Chu Luo, Jorge Goncalves, and Vassilis Kostakos. 2019. Improving wearable sensor data quality using context markers. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*. 598–601.

[10] Jamie A Ward, Gerald Pirkl, Peter Hevesi, and Paul Lukowicz. 2017. Detecting physical collaborations in a group task using body-worn microphones and accelerometers. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. 268–273. https://doi.org/10.1109/PERCOMW.2017.7917570

[11] Jamie A Ward, Daniel Richardson, Guido Orgs, Kelly Hunter, and Antonia Hamilton. 2018. Sensing Interpersonal Synchrony between Actors and Autistic Children in Theatre Using Wrist-Worn Accelerometers. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers (ISWC '18)*. Association for Computing Machinery, New York, NY, USA, 148–155. https://doi.org/10.1145/3267242.3267263 event-place: Singapore, Singapore.

[12] Florian Wolling, Cong Dat Huynh, and Kristof Van Laerhoven. 2021. IBSync: Intra-body synchronization of wearable devices using artificial ECG landmarks. In *Proceedings of the 2021 ACM International Symposium on Wearable Computers*. 102–107.

[13] Florian Wolling, Kristof van Laerhoven, Pekka Siirtola, and Juha Röning. 2021. PulSync: The Heart Rate Variability as a Unique Fingerprint for the Alignment of Sensor Data Across Multiple Wearable Devices. In *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. 188–193. https://doi.org/10.1109/PerComWorkshops51409.2021.9431015

[14] Jianfu Yang, Tzu-Hao Huang, Shuangyue Yu, Xiaolong Yang, Hao Su, Ann M Spungen, and Chung-Ying Tsai. 2019. Machine learning based adaptive gait phase estimation using inertial measurement sensors. In *2019 Design of Medical Devices Conference* (Minneapolis, Minnesota, USA). American Society of Mechanical Engineers.