ORIGINAL RESEARCH

Development and Evaluation of Wireless Interfaces to Monitor and Control Cycling Exercise During Home Telerehabilitation

Aref Smiley*, Te-Yi Tsai ⓘ*, Ihor Havrylchuk ⓘ*, Wanting Cui*, Irena Parvanova ⓘ*, Elena Zakashansky ⓘ*, Taulant Xhakli*, Hu Cui*, Joseph Finkelstein ⓘ*

Center for Biomedical and Population Health Informatics, Icahn School of Medicine at Mount Sinai, New York, NY, USA

*These authors contributed equally to this work

Correspondence: Aref Smiley, Center for Biomedical and Population Health Informatics, Icahn School of Medicine at Mount Sinai, 1770 Madison Avenue, New York, NY, 10035, USA, Tel +1 212-659-9596, Email Aref.smiley@gmail.com

Purpose: This paper focuses on developing and testing three versions of interactive bike (iBikE) interfaces for remote monitoring and control of cycling exercise sessions to promote upper and lower limb rehabilitation.

Methods: Two versions of the system, which consisted of a portable bike and a tablet PC, were designed to communicate through either Bluetooth low energy (BLE) or Wi-Fi interfaces for real-time monitoring of exercise progress by both the users and their clinical team. The third version of the iBikE system consisted of a motorized bike and a tablet PC. It utilized conventional Bluetooth to implement remote control of the motorized bike's speed during an exercise session as well as to provide real-time visualization of the exercise progress. We developed three customized tablet PC apps with similar user interfaces but different communication protocols for all the platforms to provide a graphical representation of exercise progress. The same microcontroller unit (MCU), ESP-32, was used in all the systems.

Results: Each system was tested in 1-minute exercise sessions at various speeds. To evaluate the accuracy of the measured data, in addition to reading speed values from the iBikE app, the cycling speed of the bikes was measured continuously using a tachometer. The mean differences of averaged RPMs for both data sets were calculated. The calculated values were 0.38 ± 0.03, 0.25 ± 0.27, and 6.7 ± 3.3 for the BLE system, the Wi-Fi system, and the conventional Bluetooth system, respectively.

Conclusion: All interfaces provided sufficient accuracy for use in telerehabilitation.

Keywords: telerehabilitation, cycling exercise, wireless interface, accuracy

Introduction

Advances in information technologies, sensor manufacturing, and cloud computing have greatly expanded the role of telecommunication technologies in the management of chronic health conditions.[17] Telemedicine evaluation studies demonstrated that this approach could lower healthcare costs and improve the quality of life of both patients and their caregivers.[18–20] Telerehabilitation utilizes telemedicine approaches to deliver rehabilitation services to patient homes which improves patient access to care, facilitates patient-provider communication, and allows remote supervision of patient performance.[21–24] The importance of using telerehabilitation has been especially recognized since the start of the COVID-19 pandemic, when its usage has surged in response to social distancing guidelines promulgated by the World Health Organization (WHO).[25] Recent studies have shown its effectiveness in monitoring patients' progress, ameliorating disease symptoms, and improving clinical outcomes.[2–6] In order to extend physical rehabilitation from a hospital or ambulatory settings to patient homes as a part of routine care of patients with chronic health conditions,[1] successful telerehabilitation programs should support scalable and cost-effective means for prescribing individualized exercise

1

programs, monitoring safety and adherence with exercise, two-way communication between patient and providers, and timely feedback based on patient performance.[26–28]

Training of the upper and lower extremities is often performed by using cycling exercise equipment, which is widely available in rehabilitation facilities, where patient exercise is supervised. Cycling was shown to improve clinical outcomes in patients with Parkinson's disease, chronic pulmonary disease, recovery stage after hip fracture, and patients with other chronic health conditions.[8–12] Telerehabilitation, with special attention to the remote control and monitoring of cycling exercise,[7] could broaden the use of home-based programs from expert settings to homes of people with chronic health conditions. In previous studies, we demonstrated that real-time monitoring of cycling exercise is well accepted by older adults, patients with chronic neurodegenerative conditions, and cancer.[29–31] In addition, we demonstrated that cycling exercise in which speed is controlled via the Internet in real-time significantly improves adherence with prescribed exercise speed trajectory.[32] The remotely controlled cycling exercise using a motorized bike can be especially instrumental for facilitating exercise in individuals with muscle weakness, paraplegia, or Parkinson's disease.[33,34] Thus, scalable approaches supporting accurate and cost-effective telemonitoring and remote control of cycling exercises at patient homes can greatly facilitate the implementation of successful telerehabilitation programs in patients with chronic health conditions. However, existing solutions for remote monitoring of cycling exercise are designed primarily for healthy athletes and may be cost-prohibitive for older adults or people with chronic health conditions. The current solutions are limited by an absence of low-cost wireless interfaces allowing to monitor exercise progress in real time using simple graphical and numerical representation, a lack of remote connectivity with a team of rehabilitation professionals, and an absence of an alert system that would prevent exertion levels exceeding those approved by the rehabilitation team.

The goal of this project was (1) to design and implement low-cost wireless interfaces for telemonitoring of cycling exercises using arm or leg-controlled bikes; (2) to design and implement wireless interfaces for remote control of cycling trajectory using a motorized bike; (3) to test accuracy and reliability of the resulting interfaces.

Materials and Methods
Remotely Monitored Interfaces

Two interactive biking systems (iBike) have been designed and implemented to support remote monitoring of users' exercise by communicating through either Wi-Fi or BLE protocols. We used the same hardware and user interface for both designs. Figure 1 shows an iBike system which consisted of a tablet PC and a low-cost portable bike. The cycling intervals were detected by a reed switch and then sent through wireless communications from built-in systems to tablet PC. When the telerehabilitation users were engaged in cycling exercises, the information was provided to them through an app developed for the tablet PC. To simplify the user interface, each of the apps and the iBikE had only one button to start/stop an exercise session.

Figure 1 System design for interface to monitor cycling exercise.

Tablet PC

The app was designed and implemented for the Windows operating system based on a touchscreen tablet PC, using C# (Figure 2). The first page of the app has two images of a pulse oximeter and the iBikE. To start a cycling exercise session, a user needed to push a button to connect a wrist-worn pulse oximeter. The device (WristOx2®, Model 3150) sends oxygen saturation level and pulse rate values to the tablet PC via Bluetooth connection. The information could be later monitored through the app when the user is engaged in a cycling exercise.

Once the oximeter was connected and the iBikE was turned on, the user needed to click the picture of the iBikE button on the app to connect the tablet to iBikE equipment (Figure 2). This triggered the connection of the app to the bike through either BLE or Wi-Fi. After making the connection, user could see the second page of the app, where the start of an exercise activates real-time visualization of the exercise progress via an app installed on a tablet PC. On this page, real-time cycling speed (RPM), pulse rate, oxygen saturation level, total session duration, and elapsed session time were shown in real-time (Figure 3). In addition, the user's entire exercise record was stored on a remote server and locally in the form of csv files.

iBikE Exercise Equipment

The design of the iBikE exercise interface was divided into two parts, the hardware, and the software. The hardware consisted of three modules (Figure 4A): 1) control and computing module (MCU), 2) magnetic switch module, and 3) either BLE or Wi-Fi communication module.

Hardware

We used FireBeetle ESP32 IoT Microcontroller with integrated Wi-Fi and dual-mode Bluetooth to design both iBikE systems. Cycling intervals were first detected by the magnetic switch and the interval times were calculated through the MCU. Each cycle was then sent to the tablet PC via either BLE or Wi-Fi. The control and computing module consisted of a single push button to turn on/off the MCU and a DC power source (Figure 5). We used the Adafruit push-button power switch to control the DC power source.[13] The power source had eight AAA rechargeable batteries (3.7 V, 6000 mAh). The push-button uses a latching analog circuit that is triggered by a push of the switch button. As it was shown in Figure 5, when the user pushes the switch button, the inputs of the first NAND gate (MC14093B, ON Semiconductor®) were LOW, causing its output to HIGH (C1 capacitor became open circuit) and, therefore, the inputs of the second NAND gate were HIGH. This resulted in a LOW voltage of the gate in P-channel MOSFET (DMG3415U, DIODES®) and caused a flow of current from the power source to the MCU. The MCU stays on by the feedback (100 K resistor) connected from the MOSFET gate to the input of the first NAND gate. If the user presses the switch button again, the
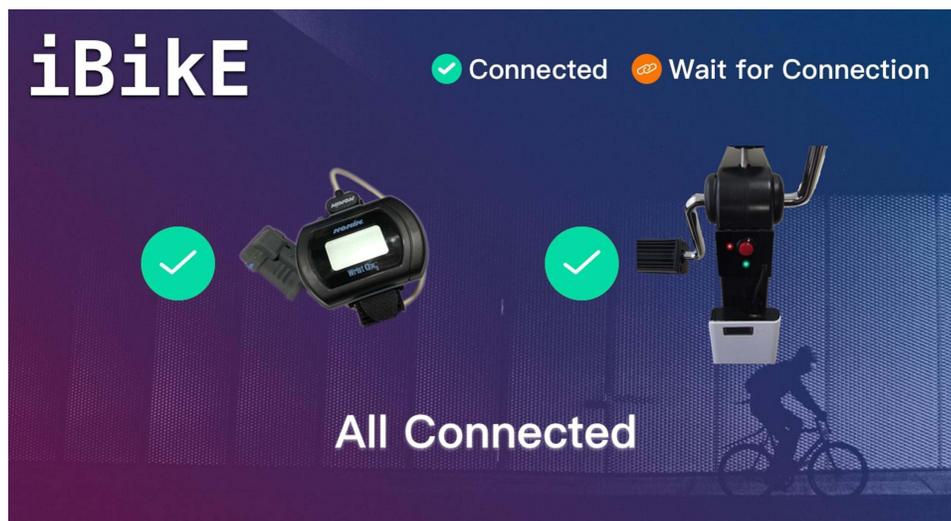
Figure 2 User interface. Users first need to push a button to pair the app with the oximeter and then push the button on the right to pair the app with iBikE equipment.
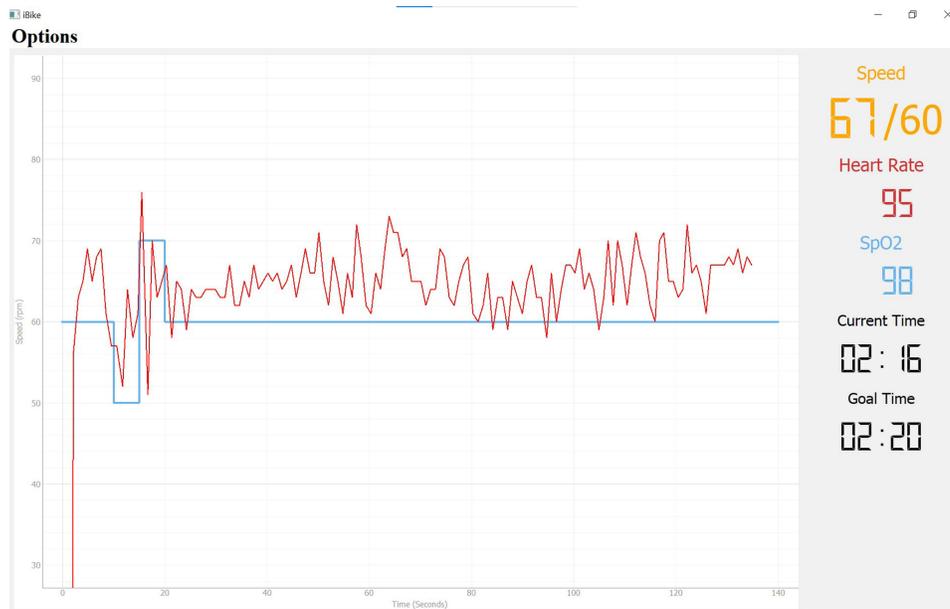
Figure 3 Real-time monitoring of the speed (red line, rpm), pulse rate, and oxygen saturation during cycling.
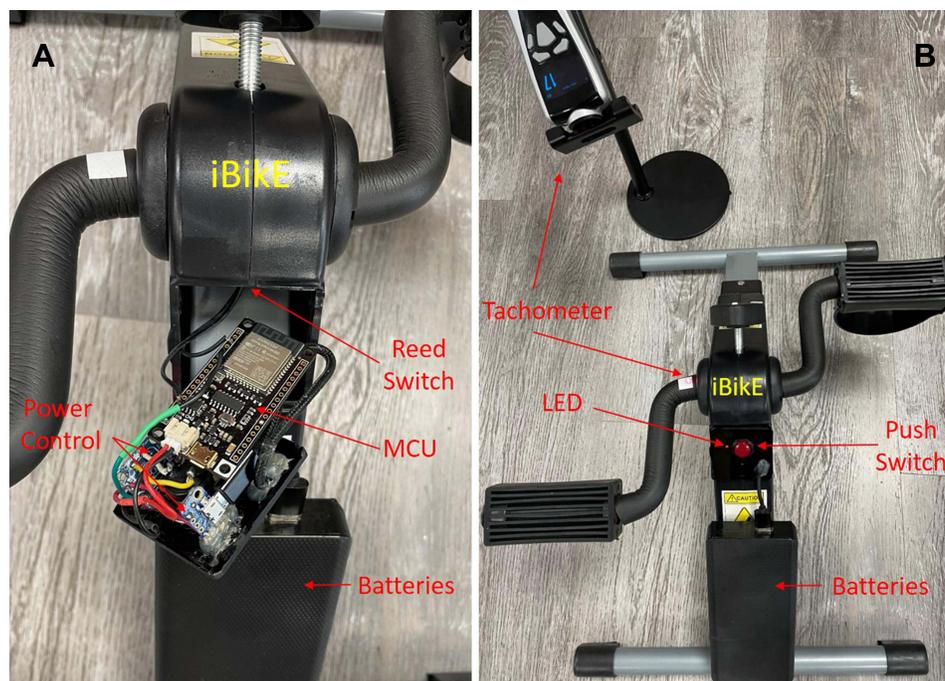
Figure 4 Exercise equipment. The bike is manufactured by AGM. (**A**) All the components, including the reed switch, push switch, MCU and BLE modules, and power control module are inside the iBikE equipment. (**B**) The user interface is only a physical button to make it easy to use the equipment. We used a laser tachometer, DT-2100, Nidec-SHIMPO,[14] in non-contact continuous measurement mode to detect the measured RPM in real-time to compare the results with data taken from iBikE equipment. Both Wi-Fi and BLE iBikE systems have the same hardware design.

HIGH voltage over C1 applies to the inputs of the first NAND gate and causes the gate of the MOSFET to turn HIGH. This results in turning off the MCU and saving power. If the user forgets to turn off the MCU, the algorithm in the MCU turns off the system after 90 seconds to save power. In this case, if the algorithm does not detect any new cycles from the reed switch for more than 90 seconds, the digital output pin (D18 in Figure 5) in MCU applies HIGH voltage to the Q1 transistor (2N2222), resulting in applying HIGH voltage to the MOSFET gate and shutting down the system. The reed
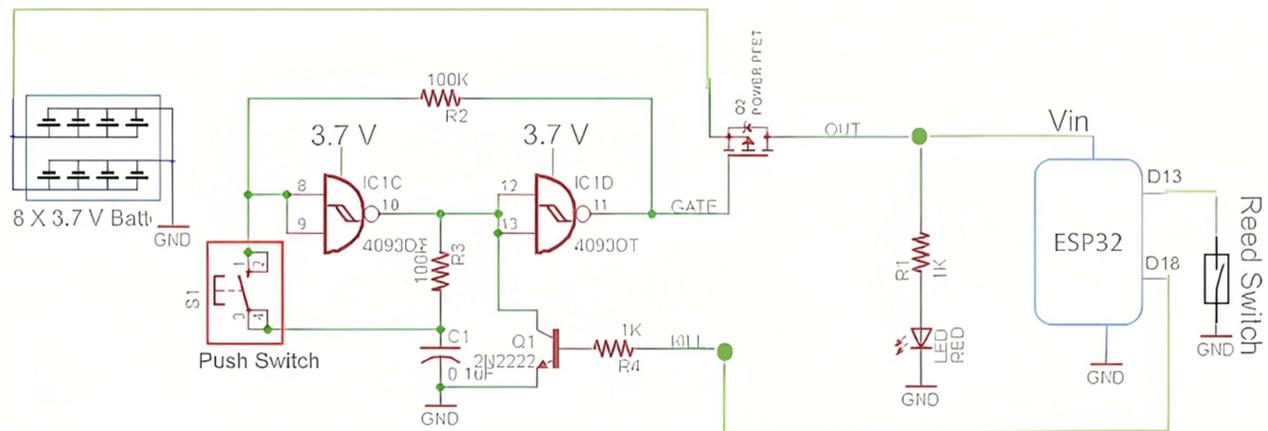
Figure 5 The control and computing module in iBikE exercise equipment.

switch (magnetic switch) converted magnetic field changes into electrical signals. It is normally an open circuit. By approaching the magnet to the reed switch, the reed switch detected the change in the magnetic field and the switch becomes a short circuit. When the magnetic field is removed, the magnetic switch detected another change and became an open circuit again. These changes (on/off signal) could be detected by MCU. In our application, the sensor was exposed to a magnetic field once during each cycle. By continuous detection of changes in rising and falling edges, we measured the cycle time intervals and prevented errors in on/off detection. We also identified the cessation of the iBikE exercise session by a user if there was no detection of on/off changes for more than 90 seconds.

Detected cycle intervals were sent from the MCU, through either BLE or Wi-Fi communication mode, to the app. For the MCU, we used FireBeetle ESP-32, a low-power consumption micro-controller designed for Internet of Things (IoT) projects. FireBeetle ESP-32 integrates a Dual-Core ESP-WROOM-32 module. The module supports both MCU and Wi-Fi & Bluetooth dual-mode communication. It could also work with a 3.7-Volt external lithium battery power supply. In this project, we designed and developed two iBikE systems to test and compare both Wi-Fi & BLE communication. The BLE system needs the tablet PC to be directly connected to the MCU. Cycle intervals were sent in 2-byte packets, a low byte, and a high byte. The low byte was the least significant part of an integer and the high byte was the most significant part of an integer. Wi-Fi system required both the tablet PC and the MCU to be connected to the same Wi-Fi network. We configured the tablet PC as a server and the MCU as a client. We used UDP protocol to communicate between the tablet PC and the bike.

To start an exercise session, the user first needs to press the red button on the iBikE exercise equipment to wake up the MCU and start Wi-Fi/Bluetooth communication with the tablet PC. On the app side, the user was required to press the button (Figure 2) to commence an exercise and allow the tablet PC to connect to the Wi-Fi/Bluetooth communication pairing with the iBikE exercise equipment. After the commencement of the communications between the tablet PC and iBikE the targeted speed trajectory appeared on the second page of the interface (Figure 3). During an exercise session, the users could monitor their actual speed value, heart rate, oxygen saturation level, and session time and compare these values to the prescribed targets in real time.

Software

The cycling interval calculation algorithm in MCU was developed to provide accurate real-time cycling information taken from the magnetic switch. The algorithm is the same for both BLE and Wi-Fi iBikE systems. A major difference is the communication protocol between the BLE and Wi-Fi systems. Each detected cycling interval is sent to the tablet PC through either Wi-Fi or BLE communication mode. Figure 6 shows the algorithm flowchart for the detection of the real-time cycling intervals using the MCU in the BLE communication mode. The algorithm is coded in C++ and Arduino IDE is used to program the MCU.
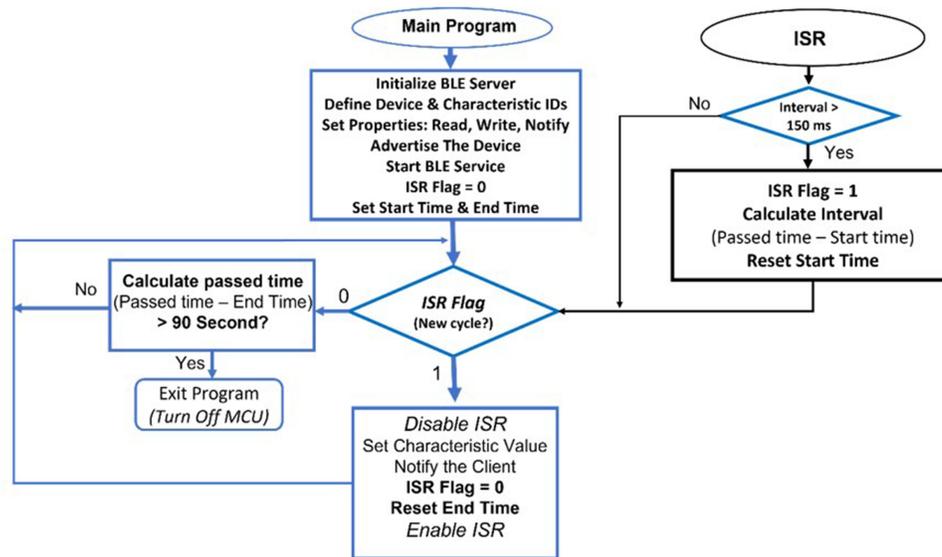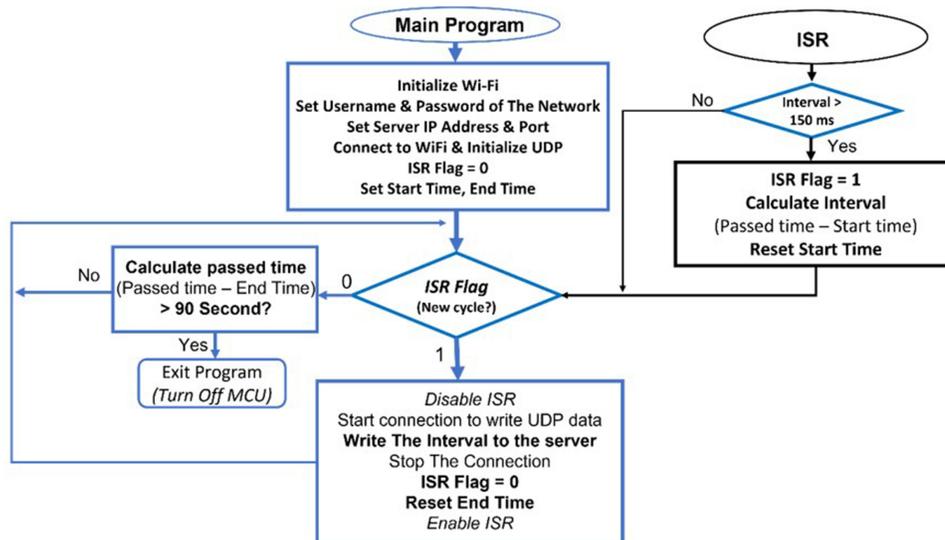
Figure 6 Algorithm flowchart for detection of the real-time cycling intervals in remotely monitored BLE iBikE.

In the software, we used the interrupt service routine (ISR) to detect the cycling intervals. An internal pull-up resistor was activated and the ISR was called whenever the MCU digital pin detected the falling edge (short circuit of the reed switch). Each detection of a falling edge means one completed cycling interval. The duration time of each interval is measured and then sent to the tablet PC via either BLE or Wi-Fi.

In the BLE communication mode (Figure 6), random device and characteristic IDs were defined, and then read, write, and notify properties were defined for the BLE server characteristic. When a cycling interval is detected by the MCU, ISR is deactivated and the characteristic value would be updated by two bytes, representing the interval value. Then the BLE server notifies the client. After updating the value, the BLE connection is closed and the ISR is activated to detect new cycling intervals.

Figure 7 shows the algorithm flowchart for the detection of the real-time cycling intervals using the MCU in the Wi-Fi communication mode. In Wi-Fi communication mode (Figure 7), both iBikE (client) and the tablet PC (server) need to be connected to the same router. To connect the MCU to the router, we need to have the username and password to be set in the code. The server IP address and the port number are set to later send cycling intervals to the server. When a cycling interval is

Figure 7 Algorithm flowchart for detection of the real-time cycling intervals in remotely monitored Wi-Fi iBikE.

detected by the MCU, the ISR is deactivated and the UDP connection from the client to the server is activated. Then data is sent to the server. Finally, the UDP connection is deactivated and the ISR is activated to detect the new cycling intervals.
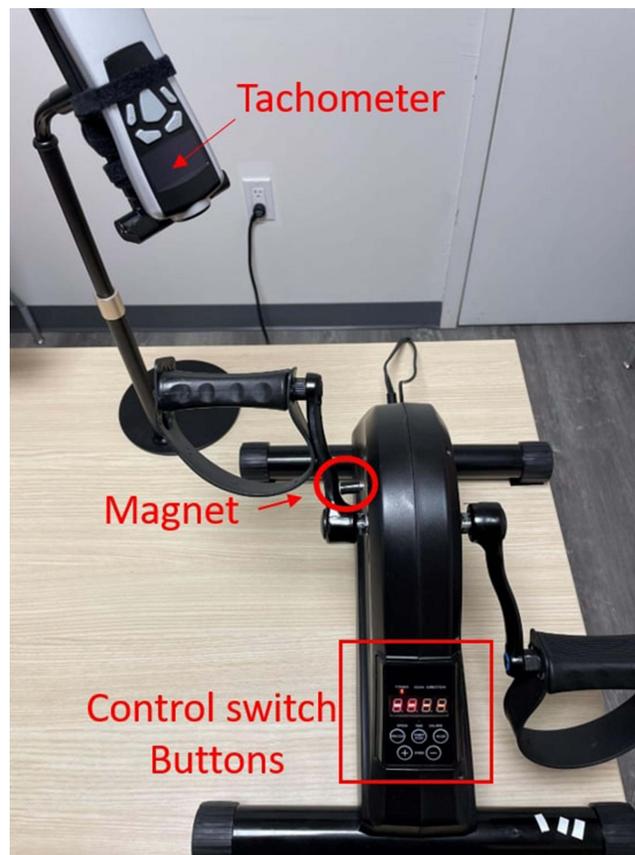
Remotely Controlled Interfaces

We modified the control board of a motorized bike to develop a system that gives a remote server ability to remotely control the bike's exercise speed according to a cycling speed trajectory prescribed by a clinical rehabilitation team. The new system also provides real-time visualization of both prescribed and actual speed trajectory through an app installed on a tablet PC screen, analogous to the remotely monitored iBikE systems (as explained above). In this design, the communication between the tablet PC and the motorized bike is carried out through conventional Bluetooth. An app was developed using C# to test the system. In the app, incoming real-time RPMs could be monitored when the exercise session started. In addition, it had three buttons to send the commands from the app to the bike and to control the speed (up and down) and direction of motor rotation. Like the app in the monitored iBikE system, the addition of an oximeter allows monitoring of oxygen saturation level as well as heart rate.

In the motorized iBikE exercise equipment, we used NPN transistors (2N2222) to replace the mechanical push button in its control circuit board (Figure 8). Then, the base gate source of each transistor was controlled by a digital output pin of the MCU (ESP-32). We also used a hall sensor (A3144) to measure cycling intervals. As shown in Figure 9, a magnet was mounted on the bike pedal. In the motorized bike, we utilized an approach that was analogous to the interval detection in the iBikE systems using a reed switch. In the motorized iBikE system, when a magnet passes by the hall sensor, which is mounted inside the bike frame, the time of cycling intervals was measured by the MCU and then sent to the tablet PC via Bluetooth. However, the cycling interval detection algorithm was different in the motorized iBikE. As shown in the algorithm flowchart, we had two main parts in the algorithm. The first part was measuring the cycling intervals and the second part was looking for the commands coming from the tablet PC to control the motor speed and direction. Whenever the user pressed any control buttons on the app, a character was sent to the MCU via Bluetooth. Based on the defined target of the character, the MCU was sending a pulse to its representative transistor (mechanical switch). In the first part of the program, we used a delay time of 20 milliseconds and a flag to make sure the detected interval was not false. When the hall sensor came near the magnet, the MCU read the sensor state multiple times. Therefore, we added a delay timer to read the sensor value every 20 milliseconds. However, if the magnet had stayed near the sensor for some time, the sensor could have been activated every 20 milliseconds. Therefore, we added a state flag to address this issue and make sure the resulting value is not false (Figure 10).
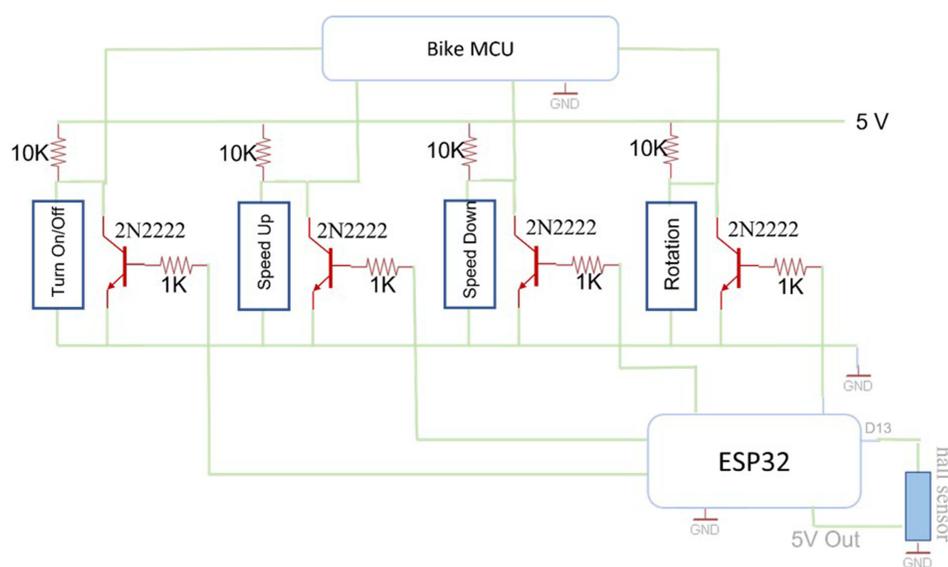
Figure 8 Control buttons in motorized iBikE were replaced by transistors to be controlled with ESP32. ESP32 communicates with the tablet PC through Bluetooth.

Figure 9 Remotely controlled and monitoring exercise equipment. The motorized bike is manufactured by HOMGIM.
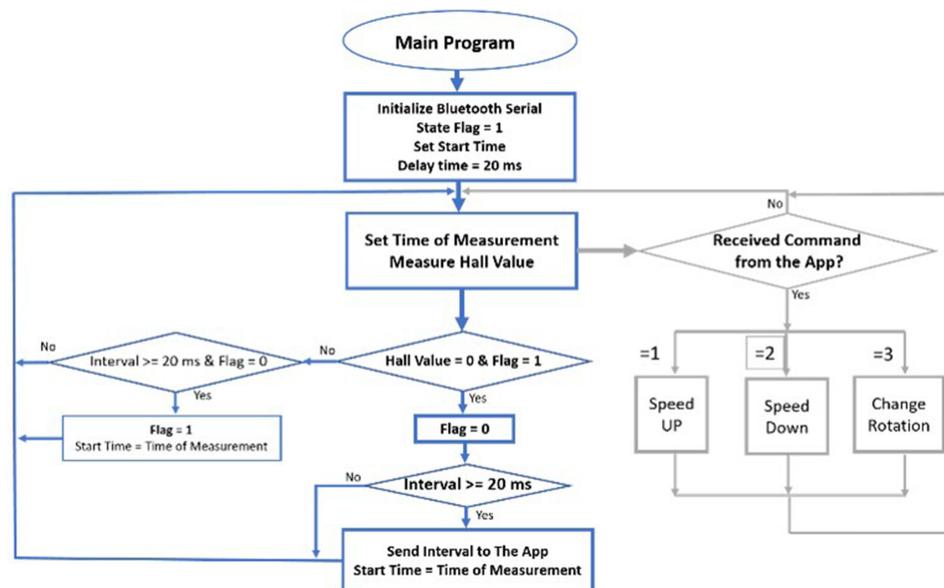
Figure 10 Algorithm flowchart on detection of the real-time cycling intervals in remotely controlled motorized iBikEs.

For BLE and Wi-Fi iBikE systems, two of the co-authors performed three separate 1-minute sessions using an arm bike at slow, medium, and fast. For the motorized iBikE system, the authors performed four consecutive sessions of hand-cycling with the cycling speed defined by the bike control system (from the slowest speed to the highest grade four speed). The speed was controlled by the developed app during each 1-minute cycling experiment. For all iBikE systems,

the recorded data taken from a laser tachometer, during the hand-cycling experiments, were compared to its representative collected data from the iBikE system.

Data Collection

For remotely monitored interfaces (BLE and Wi-Fi iBikE systems), a group of nine individuals performed 3 sessions of hand cycling for each iBikE system, with one-minute duration for each session. They started with one-minute hand-cycling at a slow pace, followed by one-minute rest. Then they proceeded with one-minute hand-cycling at a medium pace, followed by one-minute rest. Finally, they continued with one-minute hand-cycling at a fast pace. The meaning of slow, medium and fast pace was based on the user's interpretation. Participants in the study comprised nine healthy individuals aged between 21 and 61 years old, the average age was 37±14 years old, and 44% of participants were females. They performed the tests on two different days. On the first day, data was collected from the iBikE system with the BLE communication mode. On the second day, data was collected from the iBikE system with Wi-Fi communication mode. Users completed a total of 18 sessions (54 phases). Data was collected from both the iBikE system and tachometer in parallel during each session. The iBikE system recorded completed cycling intervals and sent the data to the tablet PC. Simultaneously, the tachometer collected RPM values and sent them to a PC at the sampling rate of 10 Hz.

For remotely controlled interfaces (conventional Bluetooth system), two healthy individuals performed four consecutive sessions of hand-cycle with the speed defined by the bike control system (from the slowest speed to grade four speed, 1 minute each). The speed was controlled by the developed app during the 1-minute cycling experiment.

The accuracy of all the iBikE systems was evaluated by collecting data from a laser tachometer as a reference, DT-2100, Nidec-SHIMPO.[14] The data from the laser tachometer was collected during each exercise in non-contact continuous measurement mode to detect the measured RPM in real-time (10 Hz sampling frequency).

In addition to cycling information, which was sent by the MCU to the app, other information, including heart rate, SpO2 (oxygen saturation level), and PAI (Pulse Amplitude Index), was sent from the pulse oximeter to the app.

Ethical Considerations

As there were no health risks and the participants were all authors of this paper, we did not require institutional review board approval. No protected health information was collected, and the resulting analytical data set was fully de-identified. No compensation was provided to the study participants.

Results

In order to evaluate the iBikE system's functionality and accuracy, the two collected data sets from each 1-minute session for each user (iBikE system and tachometer) were first averaged separately and then plotted (Figure 11). The plots illustrate the linear relationship between the two collected data sets.

For remotely monitored interfaces, the mean, standard deviation (SD), minimum, maximum, and 95% confidence intervals (CI) of the tachometer RPMs, were shown in Table 1. The results are calculated from all RPM values in 1-minute slow-paced, 1-minute medium-paced, and 1-minute fast-paced sessions for a total of 18 sessions for each system (nine experiments with BLE system + nine experiments with Wi-Fi system).

For remotely controlled interfaces, each 60-second session was divided into 4 sub-sessions, including 10-second, 20-second, 30-second, and 60-second sub-sessions, for measured data from the tachometer and iBikE systems. For example, in a 20-second sub-session, both collected data from the iBikE and tachometer in each 60-second session were divided into 3 bins of 20 seconds. Then the mean of the RPMs was calculated separately for each 20-second bin. Then the mean difference was calculated from each iBikE sub-session bin and its representative tachometer bin. Then the mean value of all the bins was calculated. Finally, the mean values of all the similar means for each experiment (9 experiments for each system) were averaged. Tables 2 and 3 summarize the calculated final averaged mean, standard deviation, and mean confidence interval range for all similar sub-sections for all the participants (total of 9 experiments).
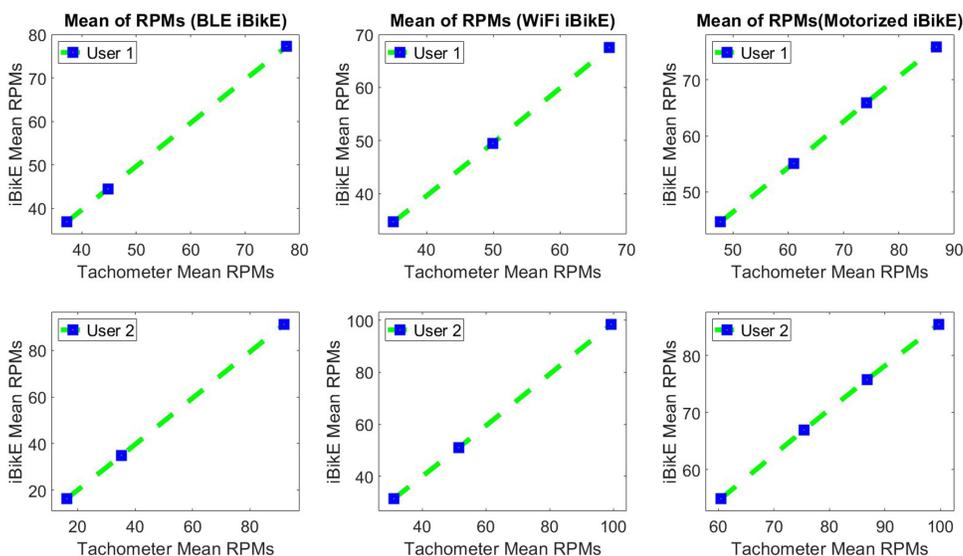
Figure 11 Collected data from BLE, Wi-Fi, and motorized iBikE systems and their representative collected data from the tachometer (in each 1- minute session) was averaged and compared. The results show the capability of our iBikE systems to measure the RPMs accurately.

Discussion

Our developed iBikE system interfaces provide remote control and monitoring of in-home cycling exercises for patients requiring a telerehabilitation exercise system. We designed the systems to be low-cost and support user-friendly interfaces for both the app and the iBikE. Test results showed that the implemented hardware solutions and software algorithms resulted in accurate measurements of cycling intervals and reliable transmission of the information from the iBikE to the app by the wireless interfaces. In the case of a motorized iBikE, our developed system was also capable of transmitting motor speed commands from the app to the motorized bike to control cycling speed in real-time.

Table 1 RPM Distribution at Different Cycling Paces

Pace	Mean	SD	Min	Max	95% CI
Slow	35.8	16.3	10.3	80.5	27.7–44.0
Medium	57.7	18.7	28.2	111.7	48.4–67.0
Fast	98.3	16.3	46.2	151.4	90.2–106.4

Table 2 Confidence Interval Range as Well as Mean and Standard Deviation (SD) of All the Collected Mean Differences Between Wireless Interface and Tachometer in Each Subsection with Similar Bins for Wi-Fi iBikE System

	Wi-Fi					
Session	Slow Pace		Medium Pace		Fast Pace	
Sub section	Mean±SD	CI	Mean±SD	CI	Mean±SD	CI
10 s	0.66±0.48	0.29–1.03	0.43±0.18	0.29–0.57	0.87±0.91	0.17–1.56
20 s	0.46±0.32	0.22–0.70	0.37±0.12	0.28–0.46	0.70±0.70	0.16–1.24
30 s	0.32±0.31	0.09–0.56	0.30±0.17	0.16–0.43	0.64±0.63	0.15–1.13
60 s	0.21±0.21	0.04–0.37	0.24±0.17	0.11–0.37	0.47±0.52	0.02–0.95

Table 3 Confidence Interval Range as Well as Mean and Standard Deviation (SD) of All the Collected Mean Differences Between Wireless Interface and Tachometer in Each Subsection with Similar Bins for BLE iBikE System

Session Sub section	BLE					
	Slow Pace		Medium Pace		Fast Pace	
	Mean±SD	CI	Mean±SD	CI	Mean±SD	CI
10 s	0.67±0.24	0.49–0.86	0.65±0.37	0.37–0.93	1.22±0.67	0.71–1.74
20 s	0.53±0.29	0.31–0.75	0.45±0.38	0.15–0.74	0.98±0.72	0.43–1.53
30 s	0.41±0.24	0.22–0.59	0.39±0.41	0.07–0.70	0.85±0.75	0.27–1.42
60 s	0.32±0.26	0.12–0.52	0.20±0.30	0.02–0.43	0.66±0.83	0.02–1.30

The app, hardware, and cycling intervals detection algorithm were the same for both monitoring iBikE systems. However, they used either BLE or Wi-Fi communication protocol to connect the app to the bike for sending information. In the cycling intervals detection algorithm, we used ISR to detect the cycling intervals from the magnetic switch. We noticed random "false" interruptions with small values that were detected by ISR. These values were less than 140 ms for the BLE system and less than 250 ms for the Wi-Fi system. These "false" interrupts, detected by MCU, could be due to the capacitive structure of the magnetic switch and external magnetic field (EMF) effects.[15,16] When the system battery went below its ~ 70% capacity, "false" interrupts were more frequent and longer, around 180 ms for BLE systems and around 300 ms for Wi-Fi systems. The effect was more frequent, and the durations were larger in Wi-Fi systems. It seems that enabling Wi-Fi in ESP-32 impacts the ISR functionality. In addition, Wi-Fi consumes more power compared to BLE in a similar session and this can directly impact the ISR functionality. Overall, in terms of battery usage and ISR performance, the BLE system was a better choice than the Wi-Fi system. To solve the ISR issue, a hold-off timer was added in the ISR to prevent the counter from increasing on the "false" interrupts (Figures 6 and 7). This did not affect cycling interval values as the highest RPM value measured during the fast pace recorded session was 142 RPM. For our application, both BLE and Wi-Fi systems were able to send data to the app fast enough without losing information. However, in applications with a higher frequency of data transmission, this issue needs to be considered.

We designed both BLE and Wi-Fi systems easy to operate by patients who had to use a single physical push button on the bike and a single push button on the app. The user was able to start/stop an exercise session just by pushing a button (Figures 2 and 4). There was no issue detected on starting/stopping communication in the systems. However, the Wi-Fi system needs to have the username and password of an active router in the area to be connected to Wi-Fi and to communicate with tablet PC. By changing the system location, where a new router is active, the MCU program needs to be updated with a new username and password. This requires special software (Arduino IDE, etc.) and knowledge of using the software to update the information. Such a restriction does not apply to the BLE system.

For a remotely controlled iBikE system, we used conventional Bluetooth to send/receive data between the app and a motorized bike system. We used a low-cost commercially available motorized bike in which we replaced its control buttons with transistors to control them with MCU (ESP-32). By sending commands from the app to the MCU, we could change the speed and cycling direction of the bike. In addition, we could send the real-time cycling intervals from the bike to the app. The bike performance and testing results demonstrated the reliability and accuracy of the system in remote control and monitoring of all cycling exercise sessions in all study participants.

As the next step, we are planning to assess both monitored and controlled iBikE systems in a broad spectrum of individuals with mobility impairment. The testing will include longer cycling time, different exercise speeds, physiological response assessment, and patient feedback. Patient adherence to regular use of the system for a prolonged period of time will be assessed.

For the iBikE monitoring system, the MCU algorithm could be further modified to lower the system power consumption by activating the sleep mode of the MCU and waking up the system whenever the data is ready to be

transmitted. In the current project, we used eight rechargeable batteries to power the unit for monitoring iBikE systems. Further improvement of the system power management could help us to optimize the power unit and get an estimation on how long the system will function accurately. In addition, the design could be further modified to make it easier for the patients to recharge the power unit.

In the motorized bike, we had to use the cycling speed increments which were pre-defined for its motor by the manufacturer. Further design and development of a motor drive for the system should overcome this limitation. The new design could also be used for all types of motorized bikes on the market, regardless of their control/drive design. In addition, compared to the monitoring iBikE system detection algorithm, the motorized iBikE showed less accuracy in cycling interval detection. Although the results were satisfactory for our application, further design modification in both hardware (eg, using different sensors), and software (eg, using ISR) may improve the system performance.

Conclusion

We developed and tested low-cost wireless interfaces to monitor and control cycling exercise during home-based telerehabilitation. The interfaces demonstrated sufficient accuracy and reliability for future use in telerehabilitation systems. Both monitored and controlled systems could be further developed to include virtual televisits by therapists and caregivers, social exergaming, extension of exercise options, support for treatment adherence, automated assessment of levels of exercise exertion, inclusion of cardiovascular monitoring, interactive health education and empowerment, and cuff-less measurement of blood pressure during exercise.[27,35–38] Patient preferences and needs should be fully reflected in the next generation of telerehabilitation systems to achieve high acceptance in the home environment.[39] Therefore, further testing of the home-based exercise cycling systems is warranted in randomized trials to evaluate the clinical impact in people with various chronic health conditions.

Acknowledgments

Disclosure

The authors report no conflicts of interest in this work.

References

1. Gibbons MC, Wilson RF, Samal L, et al. Consumer health informatics: results of a systematic evidence review and evidence based recommendations. *Transl Behav Med*. 2011;1(1):72–82. doi:10.1007/s13142-011-0016-4
2. Gehring K, Kloek CJ, Aaronson NK, et al. Feasibility of a home-based exercise intervention with remote guidance for patients with stable grade II and III gliomas: a pilot randomized controlled trial. *Clin Rehabil*. 2018;32(3):352–366. doi:10.1177/0269215517728326
3. Ariza-Garcia A, Lozano-Lozano M, Galiano-Castillo N, et al. A web-based exercise system (e-CuidateChemo) to counter the side effects of chemotherapy in patients with breast cancer: randomized controlled trial. *J Med Internet Res*. 2019;21(7):e14418. doi:10.2196/14418
4. Galiano-Castillo N, Cantarero-Villanueva I, Fernández-Lao C, et al. Telehealth system: a randomized controlled trial evaluating the impact of an internet-based exercise intervention on quality of life, pain, muscle strength, and fatigue in breast cancer survivors: telehealth System in Breast Cancer. *Cancer*. 2016;122(20):3166–3174. doi:10.1002/cncr.30172
5. Villaron C, Cury F, Eisinger F, Cappiello M-A, Marqueste T. Telehealth applied to physical activity during cancer treatment: a feasibility, acceptability, and randomized pilot study. *Support Care Cancer*. 2018;26(10):3413–3421. doi:10.1007/s00520-018-4191-4
6. Jeong IC, Karpatkin H, Stein J, et al. Relationship between exercise duration in multimodal telerehabilitation and quality of sleep in patients with multiple sclerosis. *Stud Health Technol Inform*. 2020;16(270):658–662.
7. Celesti A, Lay-Ekuakille A, Wan J, et al. Information management in IoT cloud-based tele-rehabilitation as a service for smart cities: comparison of NoSQL approaches. *Measurement*. 2020;151(107218):107218. doi:10.1016/j.measurement.2019.107218
8. Costi S, Crisafulli E, Degli Antoni F, Beneventi C, Fabbri LM, Clini EM. Effects of unsupported upper extremity exercise training in patients with COPD. *Chest*. 2009;136(2):387–395. doi:10.1378/chest.09-0165
9. McGough EL, Robinson CA, Nelson MD, et al. A tandem cycling program: feasibility and physical performance outcomes in people with Parkinson disease. *J Neurol Phys Ther*. 2016;40(4):223–229. doi:10.1097/NPT.0000000000000146
10. van Helvoort HA, de Boer RC, van de Broek L, Dekhuijzen R, Heijdra YF. Exercises commonly used in rehabilitation of patients with chronic obstructive pulmonary disease: cardiopulmonary responses and effect over time. *Arch Phys Med Rehabil*. 2011;92(1):111–117. doi:10.1016/j.apmr.2010.08.012
11. Mangione KK, Palombaro KM. Exercise prescription for a patient 3 months after Hip fracture. *Phys Ther*. 2005;85(7):676–687. doi:10.1093/ptj/85.7.676
12. Matta T, Simão R, de Salles BF, Spineti J, Oliveira LF. Strength training's chronic effects on muscle architecture parameters of different arm sites. *J Strength Cond Res*. 2011;25(6):1711–1717. doi:10.1519/JSC.0b013e3181dba162

13. Adafruit Industries. Adafruit push-button power switch breakout. Adafruit.com. [Online]. Available from: https://www.adafruit.com/product/1400. Accessed July 21, 2022.

14. Shimpo DT-2100 combination contact / non-contact tachometer with USB output. Shimpoinstruments.com. [Online]. Available from: http://shimpoinstruments.com/product/DT-2100. Accessed May 10, 2022.

15. Jain P. Reed switch: understanding specifications. [Online] Engineers Garage; 2022. Available from: https://www.engineersgarage.com/reed-switch-understanding-specifications/. Accessed May 10, 2022.

16. Teschler L. Designing with reed switches: what you need to know. [Online] Analog IC Tips; 2022. Available from: https://www.analogictips.com/designing-with-reed-switches-faq/. Accessed May 10, 2022.

17. Finkelstein J, Friedman RH. Potential role of telecommunication technologies in the management of chronic health conditions. *Dis Manag Health Outcomes*. 2000;8(2):57–63. doi:10.2165/00115677-200008020-00001

18. Joshi A, Amelung P, Arora M, et al. Clinical impact of home automated telemanagement in asthma. *AMIA Annu Symp Proc*. 2005;2005:1000.

19. Cross RK, Cheevers N, Rustgi A, et al. Randomized, controlled trial of home telemanagement in patients with ulcerative colitis (UC HAT). *Inflamm Bowel Dis*. 2012;18(6):1018–1025. doi:10.1002/ibd.21795

20. Jeong IC, Karpatkin H, Finkelstein J. Physical telerehabilitation improves quality of life in patients with multiple sclerosis. *Stud Health Technol Inform*. 2021;15(284):384–388.

21. Fleg JL, Keteyian SJ, Peterson PN, et al. Increasing use of cardiac and pulmonary rehabilitation in traditional and community settings: opportunities to reduce health care disparities. *J Cardiopulm Rehabilit Prevent*. 2020;40(6):350–355. doi:10.1097/HCR.0000000000000527

22. Rocco P, Finkelstein J. Telerehabilitation for patients with cancer: a scoping review. *Stud Health Technol Inform*. 2022;Jun(290):543–546.

23. Gibbons MC, Wilson RF, Samal L, et al. Impact of consumer health informatics applications. *Evid Rep Technol Assess*. 2009;188:1–546.

24. Finkelstein J, Jeong IC, Doerstling M, Shen Y, Wei C, Karpatkin H. Usability of remote assessment of exercise capacity for pulmonary telerehabilitation program. *Stud Health Technol Inform*. 2020;23(275):72–76.

25. Cui W, Finkelstein J. Impact of COVID-19 pandemic on use of telemedicine services in an academic medical center. *Stud Health Technol Inform*. 2021;281:407–411. doi:10.3233/SHTI210190

26. Shero ST, Benzo R, Cooper LS, et al. Update on RFA increasing use of cardiac and pulmonary rehabilitation in traditional and community settings NIH-funded trials: addressing clinical trial challenges presented by the COVID-19 pandemic. *J Cardiopulm Rehabil Prev*. 2022;42(1):10–14. doi:10.1097/HCR.0000000000000635

27. Jeong IC, Liu J, Finkelstein J. Factors affecting adherence with telerehabilitation in patients with multiple sclerosis. *Stud Health Technol Inform*. 2019;257:189–193.

28. Bedra M, Finkelstein J. Feasibility of post-acute hip fracture telerehabilitation in older adults. *Stud Health Technol Inform*. 2015;210:469–473.

29. Finkelstein J, Wood J, Cha E. Impact of physical telerehabilitation on functional outcomes in seniors with mobility limitations. *Annu Int Conf IEEE Eng Med Biol Soc*. 2012;2012:5827–5832. doi:10.1109/EMBC.2012.6347319

30. Jeong IC, Liu J, Finkelstein J. Association between system usage pattern and impact of web-based telerehabilitation in patients with multiple sclerosis. *Stud Health Technol Inform*. 2020;26(272):346–349.

31. Finkelstein J, Huo X, Parvanova I, Galsky M. Usability inspection of a mobile cancer telerehabilitation system. *Stud Health Technol Inform*. 2022;14(289):405–409.

32. Jeong IC, Finkelstein J. Remotely controlled biking is associated with improved adherence to prescribed cycling speed. *Technol Health Care*. 2015;23(Suppl 2):S543–9. doi:10.3233/THC-150992

33. Hunt KJ, Stone B, Negård NO, et al. Control strategies for integration of electric motor assist and functional electrical stimulation in paraplegic cycling: utility for exercise testing and mobile cycling. *IEEE Trans Neural Syst Rehabil Eng*. 2004;12(1):89–101. doi:10.1109/TNSRE.2003.819955

34. Alberts JL, Rosenfeldt AB, Lopez-Lennon C, et al. Effectiveness of a long-term, home-based aerobic exercise intervention on slowing the progression of Parkinson disease: design of the cyclical lower extremity exercise for Parkinson disease II (CYCLE-II) study. *Phys Ther*. 2021;101(11):pzab191. doi:10.1093/ptj/pzab191

35. Finkelstein J, Wood J. Designing pervasive telemedicine applications using various gaming platforms (2012) Proceedings of the IADIS International Conference e-Health 2012, EH 2012. Part of the IADIS Multi Conference on Computer Science and Information Systems 2012, MCCSIS 2012; 2012:97–104.

36. Jeong IC, Finkelstein J. Real-time classification of exercise exertion levels using discriminant analysis of HRV data. *Stud Health Technol Inform*. 2015;213:171–174.

37. Finkelstein J, Cha EM. Using a mobile app to promote smoking cessation in hospitalized patients. *JMIR mHealth UHealth*. 2016;4(2):e59. doi:10.2196/mhealth.5149

38. Jeong IC, Finkelstein J. Introducing contactless blood pressure assessment using a high speed video camera. *J Med Syst*. 2016;40(4):77. doi:10.1007/s10916-016-0439-z

39. Robins D, Brody R, Jeong IC, et al. Towards a highly usable, mobile electronic platform for patient recruitment and consent management. *Stud Health Technol Inform*. 2020;16(270):1066–1070.