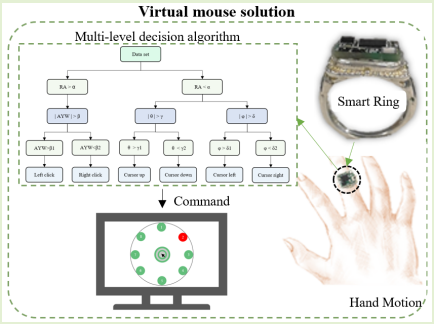
# Mouse on a Ring: A Mouse Action Scheme Based on IMU and Multi-Level Decision Algorithm

Yuliang Zhao, *Member, IEEE*, Xianshou Ren, Chao Lian, Kunyu Han, Liming Xin, and Wen J. Li, *Fellow, IEEE*

***Abstract*—The traditional mouse has been used as a main tool for human-computer interaction for more than 50 years. However, it has become unable to cater to people's need for mobile officing and all-weather use due to its reliance on the support of a two-dimensional plane, poor portability, weariness, and other problems. In this paper, we propose a portable ring-type wireless mouse scheme based on IMU sensors and a multi-level decision algorithm. The user only needs to operate in the air with a smart ring worn on the middle finger of their right hand to realize the interactive function of a mouse. The smart ring first captures changes in the finger's attitude angle to reflect how the cursor position changes. And, it captures the rapid rotation of the user's palm to the left and right to achieve mouse clicking. In addition, a multi-level decision algorithm is developed to improve the response speed and recognition accuracy of the virtual mouse. The experimental results show that the virtual mouse has a target selection accuracy of over 96%, which proves its practicability in real-world applications. This virtual mouse is expected to be used as a portable and reliable tool for multi-scenario human-computer interaction applications in the future.**




***Index Terms*—Wearable sensors, motion recognition, virtual mouse, multi-level decision algorithm.**

## I. Introduction

THE mouse has been the most widely used tool in the human-computer interaction field since it was first developed in 1964 [1]. A traditional optical mouse [2], [3] captures the mouse movement trajectory through an optical sensor, then controls the cursor movement accordingly, and finally realizes clicks through physical buttons. This scheme seems to be very natural and convenient and has been a popular choice for human-computer interaction. However, its limited compactness and portability makes it less suitable for use in scenarios such as mobile officing, business trips, and academic conferences. In addition, when a traditional mouse is used, the moving signal must be reflected by a two-dimensional (2D) plane. If the plane is not flat enough or has obstacles on it, the mouse may not work properly. This would limit the working space and mobility of the user. The wireless mouse, which transmits data using 2.4 GHz radio frequency technology, overcomes the limitation of distance, but it still requires the support of a 2D plane and cannot provide unrestricted human-computer interaction in multiple scenarios. Most mainstream multi-scenario human-computer interaction schemes today have the disadvantages of high costs, poor portability, and low recognition accuracy. It is still a technological challenge to achieve accurate multi-scenario human-computer interaction in a portable and responsive manner.

In recent years, a variety of multi-scenario unrestricted human-computer interaction schemes [4]–[21] have been proposed by researchers. For clarity purposes, these schemes

Manuscript received May 5, 2021; revised July 7, 2021; accepted July 8, 2021. Date of publication July 13, 2021; date of current version September 15, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61873307 and Grant 62073211, in part by the Hebei Natural Science Foundation under Grant F2020501040, in part by the Scientific Research Project of Colleges and Universities in Hebei Province under Grant ZD2019305, in part by the Science and Technology Planning Project of Hebei Province under Grant 206Z1702G, in part by the Fundamental Research Funds for the Central Universities under Grant N2023015, in part by the Science and Technology Planning Project of Qinhuangdao under Grant 201901B013, and in part by the State Key Laboratory of Robotics under Grant 2017-011. The associate editor coordinating the review of this article and approving it for publication was Dr. Ravibabu Mulaveesala. *(Corresponding authors: Liming Xin; Wen J. Li.)*

Yuliang Zhao, Xianshou Ren, Chao Lian, and Kunyu Han are with the Hebei Key Laboratory of Micro-Nano Precision Optical Sensing and Measurement Technology, School of Control Engineering, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China.

Liming Xin is with the School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China (e-mail: xin_liming@hotmail.com).

Wen J. Li. is with the CAS-CityU Joint Laboratory on Robotics, Department of Mechanical Engineering, City University of Hong Kong, Hong Kong, SAR, China (e-mail: wenjli@cityu.edu.hk).

Digital Object Identifier 10.1109/JSEN.2021.3096847

are collectively referred to as virtual mouse schemes in this paper. They are divided into two main categories: computer vision-based and sensor-based.

### A. Computer Vision-Based Virtual Mouse Schemes

Virtual mouse, based on computer vision technology, usually captures the user's motion trajectory through image processing, predicts the user's interaction intention, and realizes the interaction process. A computer vision-based virtual mouse usually realizes human-computer interaction by predicting the user's intention based on the user's motion trajectory it captures through image processing. Çapin *et al.* [7] realized the interaction process by processing and analyzing the user's interactive behavior in video streams. However, the camera viewing angle varies with the placement of the device, which in turn affects the reliability and accuracy of the interaction. Yousefi *et al.* [8] analyzed the user's finger clicking and sliding actions using a visual tracking algorithm. Changes in ambient light and the user's unconscious actions while resting will cause the system to produce incorrect recognition results. Virtual mouse schemes [7]–[12] based on computer vision technology usually require a camera to shoot the interaction process. The real-time video stream processing process would consume a lot of power. This requires installing large-capacity batteries, which would give the system a bloated feel. The reliability and accuracy of the system are also affected by the placement and viewing angle of the camera and the lighting of the external environment.

### B. Sensor-Based Virtual Mouse Schemes

In sensor-based virtual mouse schemes, the sensor detects human motion and identifies the user's intention, and the interaction is executed based on such information.

*1) Infrared Sensor Based Virtual Mouse Schemes:* Leblanc *et al.* [4] installed four infrared ranging sensors on the user's head and realized interaction by analyzing the user's distance with the target terminal. However, if an obstacle is present between the user and the target terminal, the reliability and accuracy of the interaction process will be largely reduced. Wada *et al.* [5] designed an infrared laser interactive device. First, the system recognizes and moves the cursor to the position of the laser spot; then the CMOS image sensor detects the intensity of the laser spot, based on which mouse clicks are realized. However, in a well-illuminated room, laser spot detection becomes difficult, which affects the stability of the system and limits the usage scenarios for this scheme. Moreover, additional image sensors increase the size of the system.

*2) Inertial Sensor Based Virtual Mouse Schemes:* In inertial sensor based virtual mouse schemes, the inertial sensor needs to be worn on the user's palm or wrist to detect hand movement [13]–[18]. Lee *et al.* [13] integrated acceleration data to obtain the moving distance of the user's hand, and then mapped the distance to control the movement of the cursor. However, the integral operation accumulates irregular errors caused by electromagnetic influences, which reduces the accuracy and reliability of the interaction process. Eom *et al.* [14] used gyroscopes to detect the angular velocity of the hand movement to control the moving speed of the cursor. However, it is difficult to accurately control the moving speed of the cursor, and the user needs to operate repeatedly to make the cursor reach the expected position on the screen. This leads to much less reliability and a poor human-computer interaction experience. Moreover, a neural network model is adopted in this scheme, which involves a large calculation workload and poor real-time performance. Some other researches have also been conducted in this respect [15]–[18], but the results are less than satisfactory.

*3) Multi-Sensor Based Virtual Mouse Schemes:* Researchers have also proposed virtual mouse schemes based on a combination of different types of sensors [19]–[21]. Ancans *et al.* [19] proposed such a scheme based on an IMU sensor and an EMG sensor. The IMU sensor controls the cursor position. The sensing electrode realizes mouse clicks by detecting changes in facial muscles caused by eye squeezing or mouth opening and closing. However, the additional EMG sensor causes the system to be bulky and awkward to carry. Moreover, long-term use of rigid sensing electrodes will cause discomfort to users. Computer *et al.* [20] combined inertial sensors and EMG signal sensors. Chen [21] used tilt sensors and touch switches. However, these composite systems generally have the disadvantages of poor portability, low reliability and low accuracy.

Based on the above analysis [4]–[21], we realize that it is necessary to develop a portable virtual mouse scheme to enable multi-scenario human-computer interaction in an accurate and responsive manner.

### C. Our Work

In this paper, low-cost inertial sensors and a stable multi-level decision motion recognition algorithm are used to develop a portable, highly accurate and responsive virtual mouse scheme for multi-scenario human-computer interaction. The scheme does not require the support of a 2D desktop. The user can realize the interactive function of a traditional mouse in the air by simply wearing a smart ring on the middle finger of their right hand. This scheme reduces the dependence on the environment and provides the possibility of unrestricted human-computer interaction in a wider range of scenarios.

The main contributions of our work can be summarized as follows:

1) A new virtual mouse scheme is developed based on the reflection relationship between attitude angle and cursor position. Specifically, we first capture the hand motion information, then translate this information into attitude angle, and finally establish the relationship between attitude angle and cursor position.
2) The smart ring captures the rapid rotation of the user's palm to the left and right to achieve mouse clicks.
3) To ensure the fast response and high tracking accuracy of the virtual mouse, a multi-level decision algorithm is developed for motion recognition.
4) Three indictors, i.e., target selection accuracy, target selection time, and path efficiency, are adopted to evaluate the performance of our virtual mouse. A host computer software is specially developed for this need.

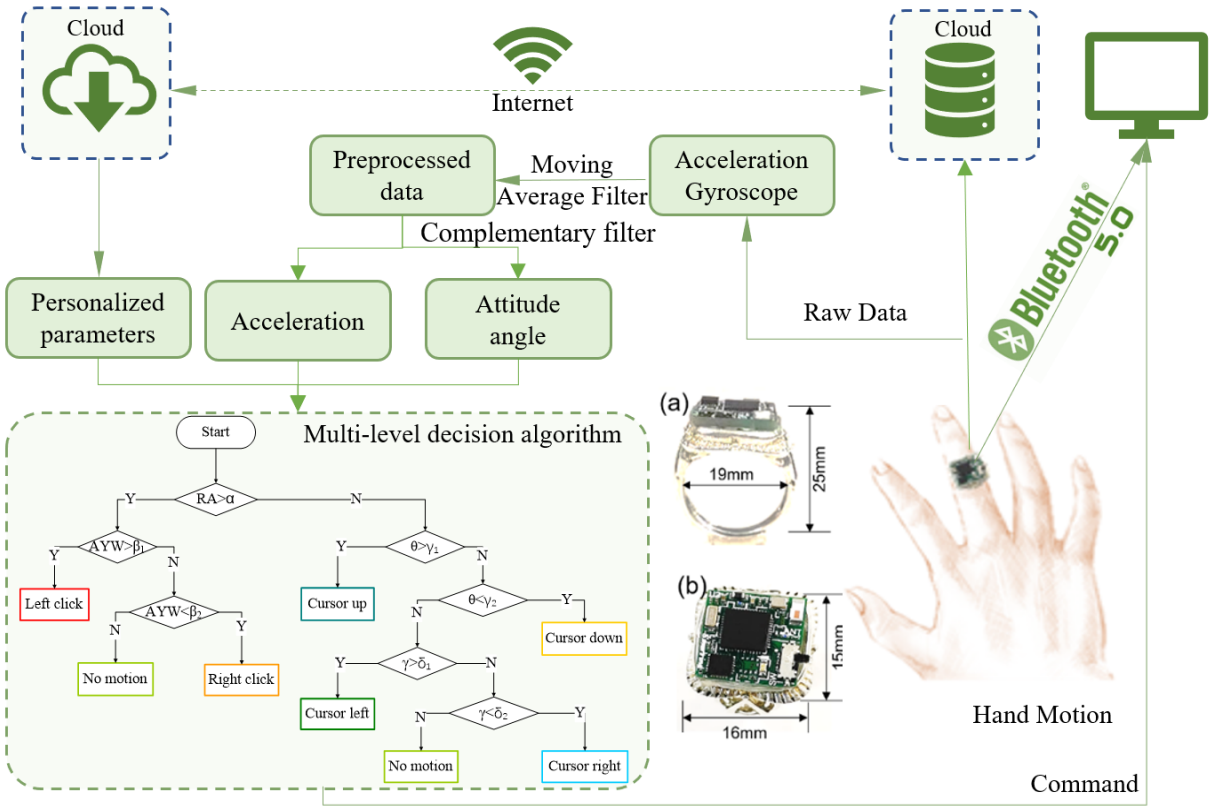



Fig. 1. Experimental setup. (a) The side view of the virtual mouse. (b) The top view of the virtual mouse.

Our virtual mouse solution is characterized by good portability, high precision, and fast response and can meet the needs of many special applications. For example, it allows users to quickly and accurately recognize hand or upper limb movements without feeling them. If it is applied to the rehabilitation training of upper limbs or joints [22]–[24], the boring and monotonous rehabilitation training can be transformed into a relaxing and interesting interaction, and the rehabilitation process can be quantitatively evaluated, thereby improving the performance of rehabilitation training.

## II. Method

In this section, we provide a detailed description of the virtual mouse scheme, including the experimental setup, how the motion data is preprocessed, the correspondence between mouse operations and hand motions, how the motion recognition process works, and how the multi-level decision algorithm is developed.

### A. Experimental Setup

In the experiment, the virtual mouse is designed based on a ring carrier platform [25]. The Bluetooth technology is used to transmit data, and a 48 mAh button battery is used to provide power, which greatly improves the portability. Fig. 1 (a-b) shows the side and top views of the virtual mouse. Compared to a traditional mouse, this virtual mouse has a significantly reduced size (19 mm × 16 mm × 25 mm) and weight (7.8 grams).

Inertial sensors, whose parameters are shown in Table I, are used to capture hand motions with the data sampling frequency set to 50 Hz. According to the Nyquist sampling theorem [26], the sensors can effectively collect frequency information within 20 Hz while the hand is moving. A sliding mean filter is used to remove high-frequency noise in the motion data without affecting motion recognition. The attitude angle, calculated by a complementary filter algorithm [27], is used to describe the hand motions. The personalized parameter settings are sent from a cloud platform. A multi-level decision algorithm is used to recognize user motion, based on which cursor movement and mouse clicking are achieved in multiple scenarios.

TABLE I
Parameters of Inertial Sensors

| Parameter | Accelerometer | Gyroscope |
|---|---|---|
| Direction | 3 axes | 3 axes |
| Power consumption | 500 uA | 5 mA |
| Temperature | -40~85 ℃ | -40~85 ℃ |
| Measuring range | ±2 g~±16 g | ±250°/s~±2000°/s |
| Communication protocol | $I^2C$ | $I^2C$ |
| ADC bits | 16 bits | 16 bits |

### B. Data Preprocessing

The collected motion data is preprocessed to remove the outliers and noise signals. First, the outliers are removed by the principle based on the $3\sigma$ criterion [28]. If the value satisfies formula (1), it is regarded as an outlier and corrected by formula (2).

$$|\mathrm{d(n)} - d_{mean}| > 3 \times d_{std} \quad (1)$$

$$\mathrm{d(n)} = \frac{d(\mathrm{n}-1) + d(\mathrm{n}+1)}{2} \quad (2)$$

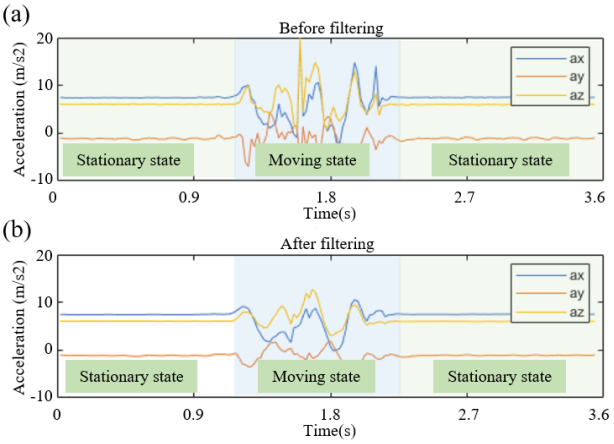



Fig. 2. (a) The acceleration data waveform before the moving average filter algorithm is used. (b) The acceleration data waveform after the moving average filter algorithm is used.

Here, d(n) is the value at time n, and $d_{mean}$ and $d_{std}$ are the mean and standard deviation of the motion data series, respectively.

Second, unpredictable noise signals in the captured motion data, caused by unconscious hand shaking and random noise generated by the sensor itself, may lead to incorrect motion recognition. To remove the noise from the original data without affecting motion recognition, a moving average filter algorithm [29], as described by formula (3), is used to preprocess the original data.

$$x(n) = \frac{1}{L}\sum_{i=0}^{\mathrm{L}-1} d(n-i) \tag{3}$$

Here, $x$(n) is the output value of the filter at time n, $d$(n) is the motion data at time n, and L is the window length of the filter algorithm, which is settled as 5 in the following experiments.

L is the window length of the filter algorithm, which determines the filtering performance. If the sliding window is too small, it would easily cause poor filtering. If it is too large, it is very likely that useful low-frequency information will be filtered out.

Fig. 2 describes the acceleration data before and after filtering by using the moving average filter algorithm. The data before filtering contains a few glitches. After filtering, the data waveform becomes smooth and has no obvious distortion. The filtering process also eliminate the noise signals, which improves and the stability and anti-interference performance of the system.

### C. Correspondence Between Mouse Operations and Hand Motions

The traditional mouse must be supported by a 2D desktop, which greatly limits its use in multiple scenarios. To solve this problem, an inertial sensor is used to capture the user's hand motion data and thereby achieve the interactive function of a traditional mouse. The inertial sensor is used to capture changes in the inclination of the finger to control cursor movement, and to capture the rapid rotation of the user's palm

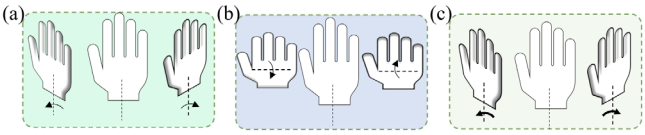



Fig. 3. Schematic of correspondence between mouse operations and hand motions. (a) The cursor moves to the left and right. (b) The cursor moves up and down. (c) Click the left and right buttons of the mouse.

to the left and right to achieve mouse clicking. This design eliminates the need for the support of a 2D plane and makes the scheme insusceptible to ambient light.

To clearly describe the scheme, six typical mouse operations, which are simplified from commonly found mouse operations in day-to-day human-computer interaction, are mapped by six corresponding hand motions. We call the position of the user's resting palm the "center area", which is parallel to the floor. Take moving the cursor to the left of the screen as an example. As shown in Fig. 3 (a), once the hand starts to tilt to the left from the "center area", the cursor would start to move to the left of the screen. The hand can stay at any position for a while, and the cursor would still move until the hand tilts back to the "center area". Take left clicking as another example. As shown in Fig. 3 (c), the motion of the user's hand quickly rotating to the left and then returning to the "center area" is regarded as a left click of the virtual mouse. Other operations of the virtual mouse are shown in Fig. 3, which are similar to the above examples.

### D. Principles of Virtual Mouse Operations

*1) Cursor Movement:* In this section, we discuss in detail the realization principles [16] of four typical cursor motions, as shown in Fig. 3 (a-b), controlled by the virtual mouse.

For the sake of simplicity, the geographic coordinate system is called the n system, and the body coordinate system of the virtual mouse is called the b system [30]. According to Euler's theorem [30], pitch ($\theta$), roll ($\gamma$), and yaw ($\psi$) are a set of attitude angles from the b system to the n system, and the attitude conversion matrix $c_b^n$, which is shown in formula (4), converts the motion data of the b system to the n system.

$$C_b^n = \begin{bmatrix} \cos\theta\cos\psi & f_{12} & f_{13} \\ \cos\theta\sin\psi & f_{22} & f_{23} \\ -\sin\theta & \sin\gamma\cos\theta & \cos\gamma\cos\theta \end{bmatrix} \tag{4}$$

Here, $f_{12}$, $f_{13}$, $f_{22}$, and $f_{23}$ are defined as follows:

$$\begin{cases} f_{12} = \sin\gamma\sin\theta\cos\psi - \cos\gamma\sin\psi \\ f_{13} = \cos\gamma\sin\theta\cos\psi + \sin\gamma\sin\psi \\ f_{22} = \sin\gamma\sin\theta\sin\psi + \cos\gamma\cos\psi \\ f_{23} = \cos\gamma\sin\theta\sin\psi - \sin\gamma\cos\psi \end{cases} \tag{5}$$

The quaternion, which is mathematically described in formula (6), contains the equivalent rotation information from the b system to the n system.

$$Q = q_0 + q_1 i + q_2 j + q_3 k \tag{6}$$

Here, Q is a quaternion; $q_0$, $q_1$, $q_2$, and $q_3$ are digital constants; i, j, and k are unit vectors.

As shown in formula (7), $c_b^n$ can be uniquely determined by the quaternion.

$$c_b^n = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_0q_1+q_2q_3) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \tag{7}$$

Formula (8) describes the relationship between the quaternion and the angular velocity. After the Picard successive approximation method is used to solve formula (8), the quaternion update formula, as shown in formula (9), can be obtained.

$$\frac{dQ}{dt} = \frac{1}{2} Q \otimes w_b^n \tag{8}$$

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}_k = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}_{k-1} + \frac{dt}{2} \begin{bmatrix} 0 & -w_x & -w_x & -w_x \\ w_x & 0 & w_z & w_y \\ w_y & -w_z & 0 & w_x \\ w_z & w_y & -w_x & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}_{k-1} \tag{9}$$

Here, $w_b^n$ is the angular velocity of the b system relative to the n system; $w_x$, $w_y$, $w_z$ are the angular velocities of the three axes, respectively; k is the serial number.

After formula (4) and formula (7) are combined, the relationship between attitude angle and quaternion can be obtained. The attitude matrix $c_b^n$ can be updated after the updated quaternion from formula (9) is obtained. As shown in formula (10), the quaternion can be converted into attitude angle.

$$\begin{cases} \theta = -\arcsin(2(q_1q_3 - q_0q_2)), & (-\frac{\pi}{2}, \frac{\pi}{2}) \\ \gamma = \arctan(\frac{2(q_0q_1 + q_2q_3)}{q_0^2 - q_1^2 - q_2^2 + q_3^2}), & (-\pi, \pi) \\ \psi = \arctan(\frac{2(q_1q_2 + q_0q_3)}{q_0^2 + q_1^2 - q_2^2 - q_3^2}), & (0, 2\pi) \end{cases} \tag{10}$$

In addition, the quaternion differential equation involves integration operations in formula (9). Due to the accumulation of low-frequency errors in the gyroscope itself, long-term operation will produce a large cumulative integration error. As a result, the final attitude angle error is relatively large. Calculating the attitude angle through the output value of the accelerometer does not produce any cumulative error, but it generates relatively large high-frequency noise for the device, which means poor reliability for short-term use. Therefore, as shown in formula (11), the attitude angle data output by the two inertial sensors can be processed together using the complementary filter algorithm [27].

$$\bar{\theta} = \frac{K}{s+K}\theta_a + \frac{s+K}{K}\theta_g \tag{11}$$

Here, K is the filter gain, which determines the cut-off frequency of the low-pass filter and the complementary weight of the angular velocity signal and the acceleration signal; $\bar{\theta}$ is the estimated value of the attitude angle; the input signals $\theta_a$ and $\theta_g$ are the attitude angles calculated by acceleration and angular velocity, respectively.

As shown in Fig. 4 (a), the IMU is integrated and placed on the ring carrier. The attitude angles [16] indicating rotations around the x-axis, y-axis and z-axis are called roll ($\gamma$), pitch ($\theta$) and yaw ($\psi$), respectively, and are used to reflect the tilt angle change of the virtual mouse. Fig. 4 (b-c) show the correspondence between the hand tilting motion and the pitch ($\theta$) change. Fig. 4 (d) shows the waveform change of pitch ($\theta$) and yaw ($\psi$) and the corresponding mouse operations.

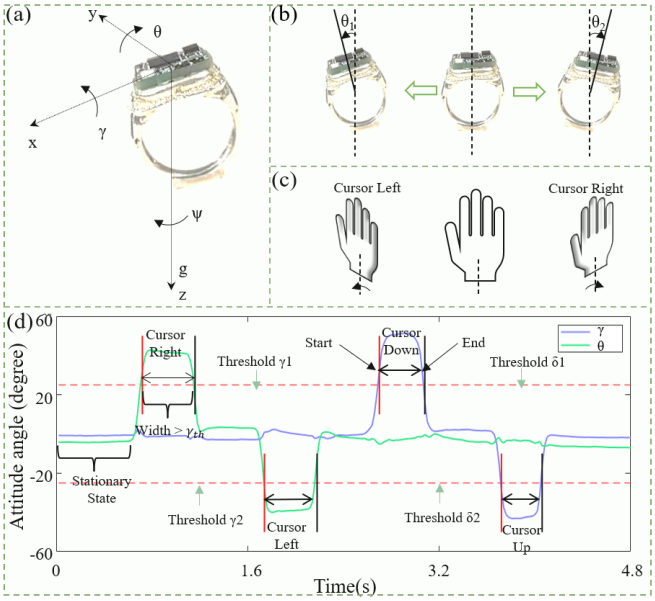



Fig. 4. (a) The relationship between attitude angle pitch ($\theta$), roll ($\gamma$), and yaw ($\psi$) with the virtual mouse coordinate system. (b) The change of the attitude angle pitch ($\theta$) when the hand is tilted. (c). The corresponding mouse operations when the cursor moves left and right.

$$Angle(k) < \gamma_1 \quad Angle(k+1) > \gamma_1 \tag{12}$$

$$Angle(k) > \gamma_2 \quad Angle(k+1) < \gamma_2 \tag{13}$$

$$k_2 - k_1 > \gamma_{th} \tag{14}$$

Here, Angle represents attitude angles pitch ($\theta$) and yaw ($\psi$), and thresholds $\gamma_1$, $\gamma_2$, $\delta_1$ and $\delta_2$ are used to detect the start and end points of the four tilting motions.

If pitch ($\theta$) satisfies formula (12), the start point of the right tilt motion can be obtained. If pitch ($\theta$) satisfies formula (13), the end point of the right tilt motion can be obtained. To avoid misrecognition caused by hand shaking, $\gamma_{th}$ in formula (14) is used to limit the holding time of the hand motion and the length of the data sampling window. Specifically, the starting point is defined as $k_1$, and the end point is defined as $k_2$. In this way, we can control the cursor on the screen to move to the desired position. By adjusting the threshold parameters using data visualization technology, the specific needs of different users can be met.

*2) Mouse Clicking:* As shown in Fig. 3 (c), the motion of the user's hand quickly rotating to the left or right and then returning to the "center area" is regarded as a click of the virtual mouse.

As shown in Fig. 5 (b), when the user's wrist rotates quickly, the resultant acceleration (RA) of the three axes, as described in formula (15), will change significantly, which

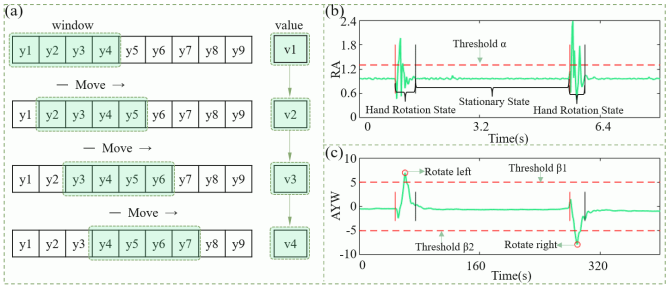



Fig. 5. The realization principle of left and right clicks of the virtual mouse. (a). Schematic of the sliding window algorithm. (b) The waveform change of RA. (c). The waveform change of AYW.

can distinguish the rotating motion from the other states.

$$RA(n) = \sqrt{a_x^2(n) + a_y^2(n) + a_z^2(n)} \tag{15}$$

Here, $a_x$, $a_y$, and $a_z$ are the preprocessed acceleration data of x-axis, y-axis, and z-axis, respectively. RA(n) is the resultant acceleration at time n.

In addition, the sliding window algorithm [31], as shown in Fig. 5 (a) and described in formula (16), is used to calculate the mean value of the y-axis acceleration (AYW) in the sliding window. Fig. 5 (c) shows the waveform change of AYW.

$$AYW(n) = \frac{1}{w}\sum_{i=0}^{w} a_y(n+i) \tag{16}$$

Here, AYW(n) is the average value of the y-axis acceleration in the sliding window obtained at time n, and w is the width of the sliding window.

As shown in Fig. 5 (b-c), under the premise that RA satisfies formula (17), if AYW satisfies formula (18), a left click will be executed; if AYW satisfies formula (19), a right click will be executed. In this way, the left and right clicking motions of the virtual mouse are recognized.

$$\mathrm{RA} > \alpha \tag{17}$$

$$AYW > \beta_1 \tag{18}$$

$$AYW < \beta_2 \tag{19}$$

### E. Multiple-Level Decision Algorithm

During the motion recognition process, there will be an increase in the types of user behaviors to be recognized, which has an effect on the recognition performance. To address this problem, we examined the hierarchy in the data set of motion recognition, based on which we proposed a multiple-level decision algorithm [32]–[34] to achieve the desired functionality as well as improved performance and accuracy for the virtual mouse. Table II gives the pseudo code of this algorithm.

It is worth noting that, when motion recognition is completed and cursor movement needs to be performed, the cursor will not move on the screen until the tilting motion is finished. In addition, the mouse will be hung up after a single click, instead of executing the click all the time. When using the virtual mouse, the user can see how the cursor moves on the screen in real time. Once any incorrect movement occurs, the user will notice it immediately, and can then adjust the mouse to re-execute the desired interaction.

TABLE II
THE PSEUDO CODE OF THE MULTIPLE-LEVEL DECISION ALGORITHM

**Algorithm** 1: Multiple-level Decision Algorithm

**Input:**
The resultant acceleration (RA) of the three axes; attitude angles pitch (θ) and yaw (ψ); The mean value of the y-axis acceleration in the sliding window (AYW).

**Output:**
Commands, including mouse left clicks and right clicks, and the cursor moving up, down, left, and right.

**Pseudo code:**

```
if RA > α
    if AYM > β1
        left click is performed
    else if AYM<β2
        right click is performed
    else
        no motion is performed
else if RA<α
    if |θ| > γ
        if θ > γ1
            cursor moves left
        else if θ < γ2
            cursor moves right
        else
            no motion is performed
    else if |φ| > δ
        if φ > δ1
            cursor moves up
        else if φ < δ2
            cursor moves down
        else
            no motion is performed
    else
        no motion is performed
else
    no motion is performed
```

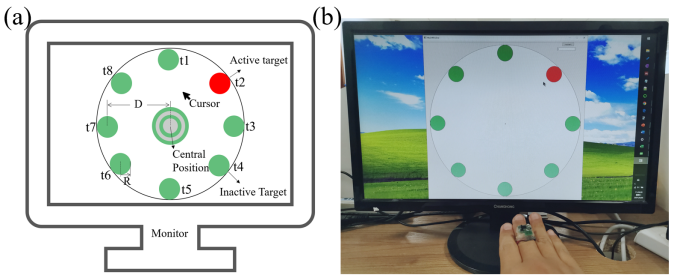



Fig. 6. The host computer used for collecting performance data of the standard and virtual mice.

## III. PERFORMANCE EVALUATION

In this section, target selection accuracy, target selection time, and path efficiency are used as the indicators to evaluate the performance difference between a standard mouse and a virtual mouse. The model of the standard mouse is Logitech G102. Six subjects who are aged between 22 and 26 and familiar with using the standard and virtual mouse participated in this experiment. Each subject was asked to wear a virtual mouse and use a standard wired mouse to collect 50 sets of data, respectively.

### A. Performance Data Acquisition

As shown in Fig. 6, a host computer, designed by QT software, is used to obtain the performance data of the standard

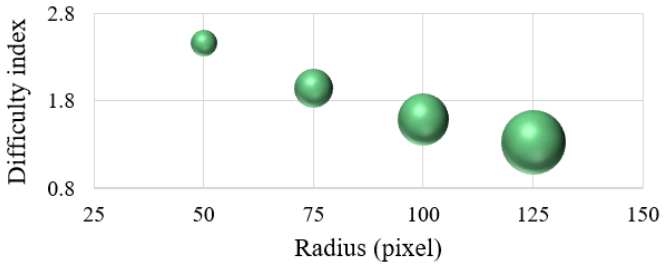



Fig. 7. Target width and distance (in pixels) and the corresponding difficulty degree.

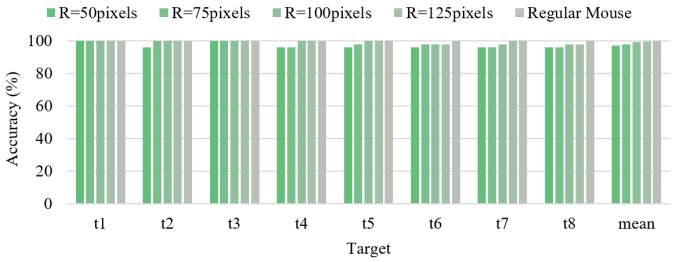



Fig. 8. Target selection accuracy results of the standard and virtual mice hitting different target circles with different radii.

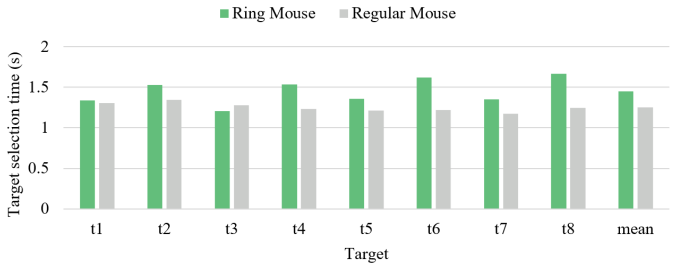



Fig. 9. Target selection time results of the standard and virtual mice hitting different target circles.

and virtual mice. The host computer runs on a 24-inch color screen with a resolution of 1920 × 1080 pixels [35].

Eight circular targets, which are uniformly distributed along the eight directions on the host computer interface, are named t1 to t8, respectively. They are of the same width so as to eliminate the influence of the approach angle [35]. In each round, one of the eight circular targets is randomly selected as the active target and turns red, and the other seven are inactive targets, which appear in green. The subject needs to control the cursor to move from the center of the screen to the position of the active target, and then perform a single click as quickly and accurately as possible. Relevant data information, including the serial number and position of the randomly assigned target, the time of target selection, whether the target is hit, the clicked target position, and the moving distance, will be automatically recorded in the log file. After hitting, the cursor will be automatically re-centered, and a new target will be randomly selected as the active target. One round of testing includes eight target selection tasks.

In the experiment, the target circle comes in four types of radii: 50, 75, 100, and 125 pixels. A small radius means a greatly increased difficulty degree [36] of subjects wearing a virtual mouse to control the cursor to hit the target. The difficulty degree is expressed by formula (20).

$$ID = \log_2\left(\frac{D}{R \times 2} + 1\right) \tag{20}$$

Here, ID is the difficulty degree; R and D, as marked in Fig. 6, are the radius of the target circle (in pixels) and the linear distance (in pixels) from the starting position of the screen center to the target point, respectively.

Fig. 7 describes the relationship between the difficulty degree, the target width, and the target distance. As the target width decreases, the difficulty degree will increase. If the virtual mouse can maintain good performance even at a high difficulty degree, it is safe to say that the virtual mouse can meet the expected applications.

### B. Performance Evaluation and Analysis

*1) Indicator 1: Target Selection Accuracy:* The target selection accuracy [14], as described in formula (21), is used to describe the accuracy of subjects wearing a virtual mouse to click on different target circles.

$$Accuracy(k) = \frac{TP + TN}{TP + FP + TN + FN} \tag{21}$$

Here, k represents the serial number of the target circles, from 1 to 8. Here, we take hitting the target circle "target-1" as an example to explain the meaning of the four parameters (TP, TN, FP and FN). TP (True Positive) means the target is hit correctly as "target-1". TN (True Negative) means the target is hit correctly as not "target-1". FP (false positive) means the target is incorrectly hit as "target-1". FN (false negative) means the target is incorrectly hit as not "target-1".

Fig. 8 depicts the target selection accuracy results of the experiment. The target selection accuracy of the standard mouse is 100%. For the virtual mouse, the accuracy varies with different targets and different radii, but it exceeds 96% for all the cases. The label "mean" represents the average accuracy for the eight target circles, which is 98.44%, with the minimum being 97% and the maximum 99.5%.

The target selection accuracy improves as the radius increases. In addition, the subject can see the real-time interaction process. Once an error occurs, the subject can control the mouse to re-execute the desired interaction. This satisfies the user's accuracy requirements for the virtual mouse in day-to-day usage.

*2) Indicator 2: Target Selection Time:* The target selection time [20], [37], which is mathematically described in formula (22), is defined as the time it takes for the subject to control the cursor to move from the screen center to the active target and hit the target. It reflects how much effort the user spends on the interaction and how fast the virtual mouse responds. Cases in which the subject fails to select the specified target is excluded from the target selection time data.

$$T = T_2 - T_1 \tag{22}$$

Here, T is the target selection time; $T_1$ denotes the time at which the subject starts moving the cursor from the screen center; $T_2$ represents the time at which the subject finishes selecting the target circle.


Fig. 10. Path efficiency results of the standard and virtual mice hitting different target circles.

TABLE III
PERFORMANCE COMPARISON BETWEEN STANDARD MOUSE AND VIRTUAL MOUSE

| Performance indicator | Standard mouse | Virtual mouse | Difference |
|---|---|---|---|
| Target selection accuracy | 100% | 98.44% | 1.56% |
| Target selection time | 1.25s | 1.45s | 0.2s |
| Path efficiency | 99.05% | 98.02% | 1.03% |

Fig. 9 compares the target selection times of the standard and virtual mice for the eight different target circles. We see that there is no significant difference in the times it takes for the standard mouse to select different target circles. When target circles 1, 3, 5, and 7 are selected, the target selection time of the virtual mouse is close to that of the standard mouse. When target circles 2, 4, 6, and 8 are selected, the target selection time of the virtual mouse is longer than that of the standard mouse.

*3) Indicator 3: Path Efficiency:* Path efficiency [37] is the measure of the linear distance of the cursor from the screen center to the target circle. Specifically, it is defined as the ratio of the ideal to the actual path that the cursor moves from the screen center to the target circle. Formula (23) shows the mathematical description of path efficiency.

$$\mathrm{PE} = \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{Distance}{\sqrt{(x_{i,j} - x_{0,0})^2 + (y_{i,j} - y_{0,0})^2}} / m/n \quad (23)$$

Here, PE is path efficiency; m is the number of target circles to click on in one round; n is the number of data collection rounds to perform. In this experiment, m is 8 and n is 50. Distance is the linear distance from the screen center to the target circle, $(x_{0,0}, y_{0,0})$ is the position of the screen center, and $(x_{i,j}, y_{i,j})$ is the position where the user clicks on the target circle.

Fig. 10 depicts the path efficiency results of the standard and virtual mice hitting different target circles. The closer the path efficiency is to 100%, the better the performance. It can be seen that the path efficiency of the virtual mouse is close to that of a standard mouse.

*4) Performance Comparison:* Table III compares the performance of the standard and virtual mice. According to the experimental results, the virtual mouse has a target selection accuracy of 98.44%, and its target selection time and path efficiency are very close to those of the standard mouse. These findings verify the practicability of the virtual mouse. It can be expected that this virtual mouse will provide a portable and reliable tool for multi-scenario human-computer interaction in the future.

## IV. CONCLUSION

In this paper, we propose a ring-type wireless mouse scheme based on IMU sensors and a multi-level decision algorithm. This scheme does not require the support of a 2D desktop. The user can realize the interactive function of a traditional mouse in the air by simply wearing a smart ring on the middle finger of their right hand. First, the smart ring captures changes in the finger's attitude angle to control how the cursor moves. This reduces the dependence on the environment and allows for unrestricted interaction in multiple scenarios. Second, the smart ring, which is integrated on the ring carrier without extra auxiliary sensors, captures the rapid rotation of the user's palm to the left and right to achieve mouse clicking. This design greatly reduces the virtual mouse's size and improves its portability. Finally, considering that the increase in user motion types affects the recognition performance, a multi-level decision algorithm is developed based on the hierarchy in the data set and used to improve the response speed and recognition accuracy of the virtual mouse. The experimental results show that the target selection accuracy of the virtual mouse is over 96%, and its target selection time and path efficiency are very close to those of a standard mouse. These findings demonstrate the effectiveness and usability of the virtual mouse in real-world applications. We believe that this virtual mouse will provide a portable, highly accurate, and responsive solution to support the needs for multi-scenario human-computer interaction in the future.

## REFERENCES

[1] H. Mccracken, "Douglas Engelbart inventor of the mouse- and more," *Time*, vol. 182, no. 4, p. 20, 2013.
[2] T. W. Ng and K. T. Ang, "The optical mouse for harmonic oscillator experimentation," *Amer. J. Phys.*, vol. 73, no. 8, pp. 793–795, Aug. 2005.
[3] T. W. Ng and M. Carne, "Optical mouse digital speckle correlation," *Opt. Commun.*, vol. 280, no. 2, pp. 435–437, Dec. 2007.
[4] D. LeBlanc, H. Hamam, and Y. Bouslimani, "Infrared-based human-machine interaction," in *Proc. 2nd Int. Conf. Inf. Commun. Technol.*, 2006, pp. 870–875.
[5] T. Wada, M. Takahashi, K. Kagawa, and J. Ohta, "Laser pointer as a mouse," in *Proc. SICE Annu. Conf.*, Sep. 2007, pp. 369–372.
[6] J.-F. Lapointe and G. Godin, "On-screen laser spot detection for large display interaction," in *Proc. IREE Int. Workshop Haptic Audio Vis. Environ. Their Appl.*, 2005, p. 5.
[7] T. Çapin, A. Haro, V. Setlur, and S. Wilkinson, "Camera-based virtual environment interaction on mobile devices," in *Proc. Int. Symp. Comput. Inf. Sci.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 4263, 2006, pp. 765–773.
[8] S. Yousefi, F. A. Kondori, and H. Li, "Camera-based gesture tracking for 3D interaction behind mobile devices," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 26, no. 8, Dec. 2012, Art. no. 1260008.
[9] Z. Ma and E. Wu, "Real-time and robust hand tracking with a single depth camera," *Vis. Comput.*, vol. 30, no. 10, pp. 1133–1144, Oct. 2014.
[10] A. Causo, E. Ueda, Y. Kurita, Y. Matsumoto, and T. Ogasawara, "Model-based hand pose estimation using multiple viewpoint silhouette images and unscented Kalman filter," in *Proc. 17th IEEE Int. Symp. Robot Hum. Interact. Commun. (RO-MAN)*, Aug. 2008, pp. 291–296.
[11] J. Cui and Z. Sun, "Model-based visual hand posture tracking for guiding a dexterous robotic hand," *Opt. Commun.*, vol. 235, nos. 4–6, pp. 311–318, May 2004.
[12] D. Wu *et al.*, "Deep dynamic neural networks for multimodal gesture segmentation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1583–1597, Aug. 2016.

[13] S. Lee, G.-J. Nam, J. Chae, H. Kim, and A. J. Drake, "Two-dimensional position detection system with MEMS accelerometers, readout circuitry, and microprocessor for padless mouse applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 10, pp. 1167–1178, Oct. 2005.

[14] G. Eom, K. Kim, C. Kim, J. Lee, and S. Chung, "Gyro-mouse for the disabled: 'Click' and 'position' control of the mouse cursor," *Int. J. Control, Automat., Syst.*, vol. 5, no. 2, pp. 147–154, Apr. 2007.

[15] J. Wu and R. Jafari, "Orientation independent activity/gesture recognition using wearable motion sensors," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1427–1437, Apr. 2019.

[16] Z. Wang *et al.*, "AirMouse: Turning a pair of glasses into a mouse in the air," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7473–7483, Oct. 2019.

[17] Y. W. Kim and J. H. Cho, "A novel development of head-set type computer mouse using gyro sensors for the handicapped," in *Proc. 2nd Annu. Int. IEEE-EMBS Special Topic Conf. Microtechnol. Med. Biol.*, May 2002, pp. 356–359.

[18] T. Zhang, L. Li, and H. Yan, "The HCI method for upper limb disabilities based on EMG and gyros," in *Proc. IEEE 13th Int. Workshop Adv. Motion Control (AMC)*, Mar. 2014, pp. 434–439.

[19] A. Ancans, A. Rozentals, K. Nesenbergs, and M. Greitans, "Inertial sensors and muscle electrical signals in human-computer interaction," in *Proc. 6th Int. Conf. Inf. Commun. Technol. Accessibility (ICTA)*, Dec. 2017, pp. 1–6.

[20] A. Computer, M. D. Groll, S. Hablani, J. M. Vojtech, and C. E. Stepp, "Cursor click modality in an accelerometer-based computer access device," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 7, pp. 1566–1572, Jul. 2020.

[21] Y.-L. Chen, "Application of tilt sensors in human-computer mouse interface for people with disabilities," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 9, no. 3, pp. 289–294, Sep. 2001.

[22] E. D. Oña, J. M. García-Haro, A. Jardón, and C. Balaguer, "Robotics in health care: Perspectives of robot-aided interventions in clinical practice for rehabilitation of upper limbs," *Appl. Sci.*, vol. 9, no. 13, p. 2586, Jun. 2019.

[23] J. Zheng, P. Shi, and H. Yu, "A virtual reality rehabilitation training system based on upper limb exoskeleton robot," in *Proc. 10th Int. Conf. Intell. Hum.-Mach. Syst. Cybern. (IHMSC)*, Aug. 2018, pp. 220–223.

[24] G. F. Muñoz, R. A. Mollineda, J. G. Casero, and F. Pla, "A RGBD-based interactive system for gaming-driven rehabilitation of upper limbs," *Sensors*, vol. 19, no. 16, p. 3478, Aug. 2019.

[25] C. Lian *et al.*, "Towards a virtual keyboard scheme based on wearing one motion sensor ring on each hand," *IEEE Sensors J.*, vol. 21, no. 3, pp. 3379–3387, Feb. 2021.

[26] P. P. Vaidyanathan, "Generalizations of the sampling theorem: Seven decades after Nyquist," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 48, no. 9, pp. 1094–1109, 2001.

[27] M. Euston, P. Coote, R. Mahony, J. Kim, and T. Hamel, "A complementary filter for attitude estimation of a fixed-wing UAV," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 340–345.

[28] Z. Ma and W. Liu, "Outlier correction method of telemetry data based on wavelet transformation and Wright criterion," *Multimedia Tools Appl.*, vol. 75, no. 22, pp. 1–13, 2016.

[29] D. Coltuc, "Mathematical complexity of running filters on semi-groups and related problems," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3191–3197, Jul. 2008.

[30] H. P. Horak, "Euler's theorem," *Amer. Math. Monthly*, vol. 101, no. 3, p. 260, 1994.

[31] H. Cohen, "Analysis of the sliding window powering algorithm," *J. Cryptol.*, vol. 18, no. 1, pp. 63–76, Jan. 2005.

[32] X.-G. Hu, P.-P. Li, X.-D. Wu, and G.-Q. Wu, "A semi-random multiple decision-tree algorithm for mining data streams," *J. Comput. Sci. Technol.*, vol. 22, no. 5, pp. 711–724, Sep. 2007.

[33] F. Duman, A. Erdamar, O. Erogul, Z. Telatar, and S. Yetkin, "Efficient sleep spindle detection algorithm with decision tree," *Expert Syst. Appl.*, vol. 36, no. 6, pp. 9980–9985, Aug. 2009.

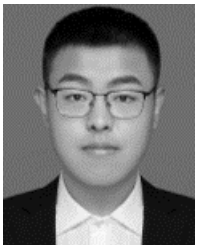
[34] S. P. Teeuwsen, I. Erlich, M. A. El-Sharkawi, and U. Bachmann, "Genetic algorithm and decision tree-based oscillatory stability assessment," *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 746–753, May 2006.

[35] R. G. Radwin, G. C. Vanderheiden, and M.-L. Lin, "A method for evaluating head-controlled computer input devices using Fitts' law," *Hum. Factors, J. Hum. Factors Ergonom. Soc.*, vol. 32, no. 4, pp. 423–438, Aug. 1990.

[36] T. Felzer, I. S. MacKenzie, and J. Magee, "Comparison of two methods to control the mouse using a keypad," in *Proc. ICCHP*, 2016, pp. 511–518.

[37] J. M. Vojtech, S. Hablani, G. J. Cler, and C. E. Stepp, "Integrated head-tilt and electromyographic cursor control," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 6, pp. 1442–1451, Jun. 2020.


**Yuliang Zhao** (Member, IEEE) received the B.S. degree in mechanical engineering from the Hubei University of Automotive Technology, the M.S. degree in mechanical engineering from Northeastern University at Qinhuangdao, Qinhuangdao, China, and the Ph.D. degree in mechanical and biomedical engineering from the City University of Hong Kong in 2016. He is currently an Assistant Professor with Northeastern University at Qinhuangdao. His research interests include intelligent sensors, machine learning, motion analytics, and big data analyses. His recent work involves applying these technologies to sports and biomechanical analyses.



**Xianshou Ren** received the B.E. degree from the University of Jinan in 2019. He is currently pursuing the M.S. degree with the School of Control Engineering, Northeastern University at Qinhuangdao, China. His current research interests are in the area of machine learning, data fusion, wearable device, and pattern recognition.

**Chao Lian** received the B.S. degree from Liaoning Technical University in 2017, and the M.S. degree in control engineering from Northeastern University at Qinhuangdao, China, in 2020. He is currently a Research Assistant with Northeastern University at Qinhuangdao. His research interests are in the area of wearable cyber physical devices, inertial measurement unit, motion analysis, machine learning, and artificial intelligence.

**Kunyu Han** received the B.E. degree from Northeast Electric Power University in 2019. He is currently pursuing the M.S. degree with the School of Control Engineering, Northeastern University at Qinhuangdao, China. His current research interests are in the area of machine learning, data fusion, wearable device, and pattern recognition.

**Liming Xin** received the B.Sc. (Eng.) degree from Jilin University in 2004, and the Ph.D. degree in mechatronics from the Chinese Academy of Science in 2009. He received the Postdoctoral Fellowship with the University of Toronto (2015–2019) before working with the University of Toronto as a Research Associate in 2019. He is currently working with Shanghai University. His research interest resides in robotics. In recent years, he has extended his research to energy harvesting and biomedical instruments.

**Wen J. Li** (Fellow, IEEE) received the B.S. and M.S. degrees in aerospace engineering from the University of Southern California (USC) in 1987 and 1989, respectively, and the Ph.D. degree in aerospace engineering from the University of California, Los Angeles (UCLA) in 1997. From September 1997 to October 2011, he was with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong. His industrial experience includes Aerospace Corporation, El Segundo, CA, USA, NASA Jet Propulsion Laboratory, Pasadena, CA, USA, and Silicon Microstructures, Inc., Fremont, CA, USA. He is currently the Chair Professor with the Department of Mechanical Engineering, and serving as an Associate Provost (Institutional Initiatives) with the City University of Hong Kong. His current research interests include intelligent cyber physical sensors, super-resolution microscopy, and nanoscale sensing and manipulation. Dr. Li served as the President for the IEEE Nanotechnology Council in 2016 and 2017.