Hindawi
Journal of Electrical and Computer Engineering
Volume 2018, Article ID 6296013, 9 pages
https://doi.org/10.1155/2018/6296013

Research Article

Effective Inertial Hand Gesture Recognition Using Particle Filtering Based Trajectory Matching

Zuocai Wang,[1] Bin Chen,[1,2] and Jin Wu[3]

[1] *Chengdu Institute of Computer Applications, Chinese Academy of Science, University of Chinese Academy of Sciences, Chengdu 610041, China*
[2] *Guangzhou Electronic Science Inc. of Chinese Academy of Science, Guangzhou, China*
[3] *School of Aeronautics and Astronautics and School of Automation, University of Electronic Science and Technology of China (UESTC), Chengdu, China*

Correspondence should be addressed to Bin Chen; chenbin306@sohu.com and Jin Wu; jin_wu_uestc@hotmail.com

Received 2 May 2017; Revised 24 July 2017; Accepted 23 August 2017; Published 11 February 2018

Academic Editor: Tiancheng Li

Hand gesture recognition has become more and more popular in applications like intelligent sensing, robot control, smart guidance, and so on. In this paper, an inertial sensor based hand gesture recognition method is proposed. The proposed method obtains the trajectory of the hand by using a position estimator. The proposed method utilizes the attitude estimation to produce velocity and position estimation. A particle filter (PF) is employed to estimate the attitude quaternion from gyroscope, accelerometer, and magnetometer sensors. The improvement is based on the resampling method making the original filter much faster to converge. After smoothing, the trajectory is then converted to low-definition images which are further sent to a backpropagation neural network (BP-NN) based recognizer for matching. Experiments on real-world hardware are carried out to show the effectiveness and uniqueness of the proposed method. Compared with representative methods using accelerometer or vision sensors, the proposed method is proved to be fast, reliable, and accurate.

1. Introduction

With the development of mobile platforms, applications on hand gesture recognition have become more and more popular. For instance, Google Inc. released its Google Project Glass in 2012 which utilizes a very simple hand gesture recognition system. Basically, the hand gesture recognition systems can be classified into two groups: the vision-based and the inertia sensor based [1].

As has been proposed in [2, 3], the vision-based hand gesture recognition system tracks user's hand's feature, analyzes user's motion, and outputs the final interpreted command. The most widely used recognition methods are artificial-intelligence-based methods [4–6], for example, the neural network, supported vector machine, and hidden Markov model. This kind of recognition system is relatively reliable but it costs too much time consumption on the modeling and matching. According to stereo vision theory [7], at least one camera is needed to accurately estimate the three-dimensional attitude and velocity of a certain object. However, such configuration of the system will enlarge the economic cost of the system significantly, including the core processors and the data acquisition system. Also the time complexity of data processing and computation are high, which limits the development of applications on mobile platforms.

Other recognition methods are inertial sensor based [8]. These methods usually use low cost MEMS accelerometers to detect the motion of human hand [9–13]. Gyroscopes can also be used to measure the attitude of the user's hand in order to analyze the gesture more accurately [14, 15]. Combining the gyroscopes and accelerometers together the real-time velocity can be determined which is helpful for hand gesture recognition [16, 17]. However, the accuracy of the method is to be improved since by using only one sensor output the observability of the gesture may not be satisfactory.

The above methods are developed due to the advances in computer vision and integrated sensor fusion. In vision-based hand gesture recognition, the feature extraction, and learning are always important. There are many related techniques like the Scale-Invariant Feature Transform (SIFT [18]), Gradient Location and Orientation Histogram (GLOH [19]), Convolutional Neural Network (CNN [20]), and so on. Extracting motion from a picture requires the analysis of the correspondence between the current frame and last frame, for example, the RANdom Sample Consensus (RANSAC [21]). However, these techniques are time-costly. In fact, by using inertial sensors, the object's motion can be computed with much simpler frameworks. These years the attitude and position estimation based on MEMS sensors have been extensively studied [22], which provides us with new ways of obtaining hand's motion [23].

The main purpose of this research is to find a way that adopts inertial sensors only to determine the correct human hands gesture. According to existing papers published between 2012 and 2016, many of them use learning techniques to model the gesture via captured sensor data at one certain moment. At this point, it is not so reliable and accurate because a gesture can only be well determined if the history data is also included for analysis. To implement this, the trajectory determination is vital. Therefore, this paper first computes the trajectory of human hand by inertial integration of attitude, velocity, and position. What has to be pointed out is that the attitude estimation significantly influences the results of velocity and position; hence a PF is introduced for accurate and convergent estimates of attitude quaternion. In this paper, we mainly have the following contributions:

(1) We use inertial sensors including gyroscopes and accelerometers to estimate the position of the user's hand. The first step is to fuse these sensor outputs into the attitude quaternion using the PF. The velocity and position update is then performed using the ZUPT-aided equations.

(2) After recording the trajectory of the hand, the trace is saved as a low-definition image which will be further sent into a BP-NN based gesture recognition system proposed in [24].

(3) The proposed method is systematically verified by experiments showing its advantages in estimation accuracy and feasibility of implementation. It is also compared to a representative gesture recognition algorithm based on accelerometer, which gives proof of its superiority on the success rate.

Figure 1 shows the structure of the recognition system of our paper. This papers is briefly structured as follows: Section 2 introduces the trajectory estimation method. Section 3 includes the BP-NN based hand gesture recognition method. Experiment, simulations, and results are given in Section 4. Section 5 contains the concluding remarks.

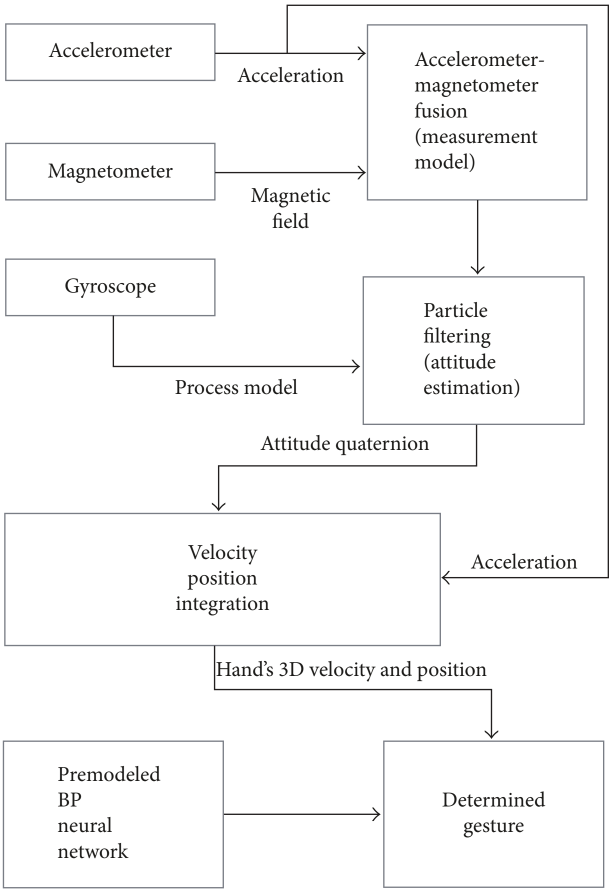

Accelerometer
Acceleration
Accelerometer-magnetometer fusion (measurement model)
Magnetometer
Magnetic field
Gyroscope
Process model
Particle filtering (attitude estimation)
Attitude quaternion
Velocity position integration
Acceleration
Hand's 3D velocity and position
Premodeled BP neural network
Determined gesture

Figure 1: Basic structure of the system.

2. Relative Position Estimation

2.1. Attitude Determination. In this paper, rotation vectors and quaternions are used for attitude determination. Rotation vector is an effective form for rotational representation. In fact rotation vector is the eigenvalue of the direction cosine matrix (DCM) and can be given by [25]

$$(\mathbf{C} - \mathbf{I})\,\mathbf{\Phi} = 0, \tag{1}$$

where $\mathbf{C}$ is the DCM, $\mathbf{I}$ is the identity matrix, and $\mathbf{\Phi}$ is the rotation vector which can be defined by

$$\mathbf{\Phi} = \begin{bmatrix}\Phi_x & \Phi_y & \Phi_z\end{bmatrix}^{\mathbf{T}}, \tag{2}$$

where Φ_x is the projection of $\mathbf{\Phi}$ on x-axis, Φ_y is the projection of $\mathbf{\Phi}$ on y-axis, and Φ_z is the projection of $\mathbf{\Phi}$ on z-axis. The differential equation of the rotation vector can be given by [25]

$$\dot{\mathbf{\Phi}} = \omega + \frac{1}{2}\mathbf{\Phi} \times \omega + \frac{1}{12}\mathbf{\Phi} \times (\mathbf{\Phi} \times \omega), \tag{3}$$

where ω is the real-time angular velocity which can be given by $\omega = (\omega_x, \omega_y, \omega_z)^T$. Generally speaking, (3) can be simplified as

$$\dot{\mathbf{\Phi}} = \omega + \frac{1}{2}\mathbf{\Phi} \times \omega \tag{4}$$

By integrating (4), the real-time rotation vector can be calculated. The most important characteristic of the rotation vector is that the rotation vector can compensate for noncommutative error effectively. In 1971, Bortz proposed an attitude estimation method that used rotation vector to avoid coning error [26]. Twelve years later, Miller proposed a discrete-rotation-vector based method with three samples in one integration period [27]. Savage summarized several compensation methods in [28, 29]. According to Savage's theory, the three-sample rotation vector method supposed that the angular velocity can be given by a two-order polynomial within one updating period:

$$\omega(t+\xi) = \mathbf{a} + 2\mathbf{b}\xi + 3\mathbf{c}\xi^2, \tag{5}$$

where **a**, **b**, **c** can be given by [25]

$$\begin{aligned} \mathbf{a} &= \frac{1}{2}\left(11\Delta\theta_1 - 7\Delta\theta_2 + 2\Delta\theta_3\right) \\ \mathbf{b} &= \frac{9}{2h^2}\left(-2\Delta\theta_1 + 3\Delta\theta_2 - \Delta\theta_3\right) \\ \mathbf{c} &= \frac{9}{2h^3}\left(\Delta\theta_1 - 2\Delta\theta_2 + \Delta\theta_3\right). \end{aligned} \tag{6}$$

Expanding (4) with Taylor Series, we have

$$\mathbf{\Phi}(t+h) = \mathbf{\Phi}(t) + \dot{\mathbf{\Phi}}(t)h + \frac{\ddot{\mathbf{\Phi}}(t)h^2}{2!} + \cdots. \tag{7}$$

Suppose $\mathbf{\Phi}(t) = 0$, multiorder derivatives of the rotation vector can be given by

$$\begin{aligned} \mathbf{\Phi}(t) &= 0, \\ \mathbf{\Phi}'(t) &= \mathbf{a}, \\ \mathbf{\Phi}''(t) &= 2\mathbf{b}, \\ \mathbf{\Phi}^{(3)}(t) &= 6\mathbf{c} + \mathbf{a}\times\mathbf{b}, \\ \mathbf{\Phi}^{(4)}(t) &= 6(\mathbf{a}\times\mathbf{c}), \\ \mathbf{\Phi}^{(5)}(t) &= 6(\mathbf{b}\times\mathbf{c}), \\ \mathbf{\Phi}^{(n)}(t) &= 0, \quad n \geq 6. \end{aligned} \tag{8}$$

Inserting (8) into (7), the discrete form of (4) can be obtained [28]:

$$\begin{aligned} \mathbf{\Phi}(t+h) &= \mathbf{a}h + \mathbf{b}h^2 + \mathbf{c}h^3 + \frac{h^3}{6}(\mathbf{a}\times\mathbf{b}) + \frac{h^4}{4}(\mathbf{a}\times\mathbf{c}) \\ &\quad + \frac{h^5}{10}(\mathbf{b}\times\mathbf{c}) \\ &= \Delta\theta + \frac{33}{80}(\Delta\theta_1\times\Delta\theta_3) + \frac{57}{80}\Delta\theta_2 \\ &\quad \times(\Delta\theta_3 - \Delta\theta_1), \end{aligned} \tag{9}$$

where $\Delta\theta = \Delta\theta_1 + \Delta\theta_2 + \Delta\theta_3$ is the vector sum of angular increments. Equation (9) is called the three-sample rotation vector algorithm. Once the rotation vector is calculated, it can be converted to a quaternion:

$$\mathbf{q}(h) = \left[\cos\frac{\phi}{2} \quad \frac{\Phi_x}{\phi}\sin\frac{\phi}{2} \quad \frac{\Phi_y}{\phi}\sin\frac{\phi}{2} \quad \frac{\Phi_z}{\phi}\sin\frac{\phi}{2}\right]^T, \tag{10}$$

where $\phi = \sqrt{\Phi_x^2 + \Phi_y^2 + \Phi_z^2}$ denotes the length of the rotation vector. Multiplying the quaternions from the initial time, the DCM from hand to geographical coordinate system can be given by [28]

$$\mathbf{C}_b^n = \begin{pmatrix} \left(q_0^2 + q_1^2 - q_2^2 - q_3^2\right) & 2\left(q_1q_2 - q_0q_3\right) & 2\left(q_1q_3 - q_0q_2\right) \\ 2\left(q_1q_2 - q_0q_3\right) & \left(q_0^2 - q_1^2 + q_2^2 - q_3^2\right) & 2\left(q_0q_1 + q_2q_3\right) \\ 2\left(q_1q_3 + q_0q_2\right) & 2\left(q_2q_3 - q_0q_1\right) & \left(q_0^2 - q_1^2 - q_2^2 + q_3^2\right) \end{pmatrix}, \tag{11}$$

where b represents hand while n represents North-East-Down (NED) Coordinate System. The DCM forms the mathematical platform of the Strapdown Inertial Navigation System (SINS).

The MEMS gyroscope usually has bias in the output which leads to the drift of the attitude's integral. In this way, accelerometer is usually used for compensating for the drift. In engineering practice, observers and optimal filtering techniques are usually adopted to improve the accuracy of the attitude estimation [23]. In the next subsection, we would like to introduce one Improved Unscented Particle Filter for such state estimation.

2.2. Particle Filtering for Attitude Estimation. The inertial attitude determination can be acquired from equations in last subsection. However, in real practice, other sensors like accelerometer and magnetometer are integrated with gyroscope for much more accurate and stable results. We now introduce the particle filtering algorithm which was proposed in [30]. Assume that the discrete state space has the following model:

$$\begin{aligned} \mathbf{X}_k &= f(\mathbf{X}_{k-1}) + \mathbf{W}_k, \\ \mathbf{L}_k &= h(\mathbf{X}_k) + \mathbf{V}_k, \end{aligned} \tag{12}$$

where $\mathbf{X}_k$, $\mathbf{L}_k$ represent the state vector and observation vector, respectively. $\mathbf{W}_k$, $\mathbf{V}_k$ are independent zero-mean white Gaussian noises (WGNs). Then the particle filtering has the following calculation procedure:

(1) *Initialization*: at $k = 0$, we extract N sample point $x_0^{(i)}$ and weight $w_0^{(i)} = 1/N$ from the importance function, where $i = 1, 2, \dots, N$. Using

$$w_{k+1}^{(i)} \propto w_k^i \frac{p\left(\mathbf{L}_{k+1} \mid \mathbf{X}_{k+1}^{(i)}\right) p\left(\mathbf{X}_{k+1}^{(i)} \mid \mathbf{X}_k^{(i)}\right)}{Q\left(\mathbf{X}_{k+1}^{(i)} \mid \mathbf{X}_k^{(i)}, \mathbf{L}_{k+1}\right)}, \tag{13}$$

where Q denotes the importance function and p is the probability density function, we may compute the particle's weight. It is also noted that here the former p denotes the likelihood function while the latter is the state transition one.

(2) *Forecast*: at time epoch k, we forecast N particles using the state model where the process noise is subjected to the probability density function $p(\mathbf{W}_k)$.

(3) *Update*: the importance of N particles is updated using $w_{k+1}^{(i)} = w_k^{(i)} p(\mathbf{L}_{k+1} \mid \mathbf{X}_{k+1}^{(i)})$ and then normalization and then using norm 1.

(4) *Resampling*: given the threshold number of the particles as N_{th}, we can calculate the effective number of particles by

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} \left[w_k^{(i)}\right]^2}. \tag{14}$$

When we get $N_{\text{eff}} < N_{\text{th}}$, we may resample the N particles and obtain their weights. Then using $\widehat{\mathbf{X}}_k = (1/N) \sum_{i=1}^{N} w_k^{(i)} \mathbf{X}_k^{(i)}$ the final estimated state can be computed. In this paper, the resampling technique is chosen as the residual resampling.

The presented scheme is a sequential Monte Carlo suboptimal method. In attitude estimation, the quaternion can be used as the state vector. The accelerometer-magnetometer combination can be used for measurement model. The measurement equation, that is,

$$\begin{aligned} \mathbf{A}^b &= \mathbf{C}\mathbf{A}^r \\ \mathbf{M}^b &= \mathbf{C}\mathbf{M}^r, \end{aligned} \tag{15}$$

can show that the direction cosine matrix is the quadratic function of the attitude quaternion. Hence there is nonlinearities inside the measurement model and it is very suitable to use particle filtering for state estimation.

With the above algorithm, we may improve the conventional nonlinear estimation results that mainly generated from Extended Kalman Filter (EKF [31]). Here we define the state variable as $\mathbf{X} = (q_0, q_1, q_2, q_3)^T$. The state propagation model is given in the last subsection and the variance information has been systematically derived. Related materials can be found in [32]. Then, with the presented approach, the filtering can be recursively continued.

2.3. Velocity and Position Determination. The differential equation of velocity can be given by [28]

$$\dot{\mathbf{v}}^n = \mathbf{C}_b^n \mathbf{f}^b - \left(2\omega_{ie}^n + \omega_{en}^n\right) \times \mathbf{v}^n + \mathbf{g}^n, \tag{16}$$

where $\mathbf{v}$ is the velocity of hand, $\mathbf{f}^b$ is the acceleration measured by the accelerometer, ω_{ie}^n is the rotational angular velocity of the Earth in NED, ω_{en}^n is the angular velocity of NED relative to the Earth-Centered, Earth-Fixed (ECEF) Coordinate system and $\mathbf{g}^n$ is the vector of gravity which can be written as

$$\mathbf{g}^n = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T, \tag{17}$$

where g is the local gravitational acceleration. Usually, MEMS gyroscopes cannot sense the rotational angular velocity of the Earth, so that ω_{en}^n is much smaller than ω_{ie}^n in most low-speed cases. So (13) can be simplified in this paper as follows:

$$\dot{\mathbf{v}}^n = \mathbf{C}_b^n \mathbf{f}^b + \mathbf{g}^n \tag{18}$$

In accordance with (15), the discrete form of the velocity update equation can be given by [29]

$$\mathbf{v}_k = \mathbf{v}_{k-1} + \mathbf{C}_{k-1} \int_{t_{k-1}}^{t_k} (\mathbf{f} + \Delta\theta \times \mathbf{f})\, dt + \int_{t_{k-1}}^{t_k} \mathbf{g}\, dt, \tag{19}$$

where $\mathbf{C}_{k-1}$ is the DCM at the moment of k and $\Delta\theta$ is the angular increment which is a function of time. Let

$$\begin{aligned} \Delta\mathbf{v}_{sf(k)} &= \int_{t_{k-1}}^{t_k} (\mathbf{f} + \Delta\theta \times \mathbf{f})\, dt \\ \Delta\mathbf{v}_{g(k)} &= \int_{t_{k-1}}^{t_k} \mathbf{g}\, dt \end{aligned} \tag{20}$$

and (16) can be written as

$$\mathbf{v}_k = \mathbf{v}_{k-1} + \mathbf{C}_{k-1} \Delta\mathbf{v}_{sf(k)} + \Delta\mathbf{v}_{g(k)} \tag{21}$$

Obviously, the most significant item of (17) is $\mathbf{C}_{k-1} \Delta\mathbf{v}_{sf(k)}$ which can be derived as follows:

$$\begin{aligned} \Delta\mathbf{v}_{sf(k)} = {} & \Delta\mathbf{v}_k + \frac{1}{2} \Delta\theta_k \times \Delta\mathbf{v}_k \\ & + \frac{1}{2} \int_{t_{k-1}}^{t_k} \left[\Delta\theta(t) \times \mathbf{f}(t) + \Delta\mathbf{v}(t) \times \omega(t)\right] dt. \end{aligned} \tag{22}$$

Let $\Delta\mathbf{v}_{\text{scul}(k)} = (1/2) \int_{t_{k-1}}^{t_k} [\Delta\theta(t) \times \mathbf{f}(t) + \Delta\mathbf{v}(t) \times \omega(t)] dt$ be the sculling motion item. According to [29], the three-sample-based sculling motion item can be given by

$$\begin{aligned} \Delta\mathbf{v}_{\text{scul}(k)} = {} & \frac{33}{80} \left(\Delta\theta_{k,1} \times \Delta\mathbf{v}_{k,3} + \Delta\mathbf{v}_{k,1} \times \Delta\theta_{k,3}\right) \\ & + \frac{57}{80} \left(\Delta\theta_{k,1} \times \Delta\mathbf{v}_{k,2} + \Delta\theta_{k,2} \times \Delta\mathbf{v}_{k,3} + \Delta\mathbf{v}_{k,1} \right. \\ & \left. \times \Delta\theta_{k,2} + \Delta\mathbf{v}_{k,2} \times \Delta\theta_{k,3}\right). \end{aligned} \tag{23}$$

With (16), (17), (18), (19), and (20), the real-time velocity can be calculated. The position of the hand in NED can be also calculated by integrating the velocity which can be given by

$$\mathbf{p}^n = \int_0^t \mathbf{v}^n d\tau. \tag{24}$$

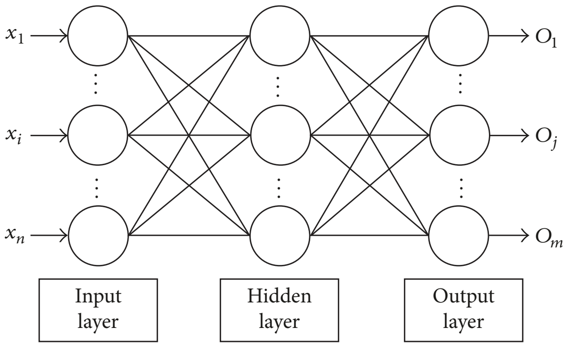


Figure 2: The structure of BPNN.

Here, we suppose that the initial position vector is $\mathbf{p_0}^n = \mathbf{0}$ and the real-time position of the hand will be calculated and recorded. The recorded trace will be projected to the vertical plane and the projected trace will be saved as a 64 × 32 image which can be recognized later using BP-NN.

In real applications, the velocity and position estimation may diverge due to the bias of the accelerometer. In this way, the zero-velocity update (ZUPT) is introduced to overcome such disadvantages [33]. That is to say, when the acceleration is measured to be less than a settled threshold (usually the absolute value of the accelerometer's bias), the acceleration would not be integrated, so as to the position.

3. Recognition of Gesture

3.1. The Structure of BP-NN. The backpropagation neural network is a Multilayer Neural Network (MNN) which owns at least three layers and each layer of the BP-NN consists of several neurons. Each neuron in the hidden layer is connected with all the neurons in both the front layer and the rear layer while there is no connection between neurons within a layer. When training the BP-NN, the activation values of the neurons will spread from the input layer to the output layer. To lower the error between expected output values and the feedback values, the backpropagation algorithm (BPA) will be used to adjust the weights between the neurons from the output layer to the input layer. Just like the feedback control in Modern Control Theory, the error will then be decreased time after time. When the error is less than a predetermined threshold, the training process stops and the trained BP-NN can be used for gesture recognition. In this paper, the three-layer BP-NN is used for recognition. A three-layer BP-NN can be illustrated in Figure 2.

3.2. The Training Algorithm of BP-NN. According to neural networks theory, the basic training methods can be given as follows.

Let $f(x)$ be the activation function of the input layer, hidden layer, and output layer. In this paper, the Sigmoid Transfer Function (STF) $f(x) = 1/(1+e^{-x})$, Linear Transfer Function (LTF) $f(x) = x$, and Hyperbolic Tangent Sigmoid Transfer Function (HTSTF) $f(x) = 2/(1+e^{-x}) - 1$ are used for representing activation functions.

The training process of BP-NN is actually an optimization problem. The training algorithm based on Gradient Descent Method (GDM) can be given by

$$\begin{aligned} w_{ij}(n+1) &= w_{ij}(n) + \Delta w_{ij} \\ \Delta w_{ij} &= \eta \delta_j O_i, \end{aligned} \tag{25}$$

where η is the training efficiency, O_i denotes the real output of the ith neuron, and δ_j is the error of the jth neuron. δ_j can be given by

$$\delta_j = \begin{cases} \left(y_j - O_j\right) O_j \left(1 - O_j\right), & N_j \in OutputLayer \\ O_j \left(1 - O_j\right) \sum_k \delta_k w_{jk}, & N_j \in HiddenLayer, \end{cases} \tag{26}$$

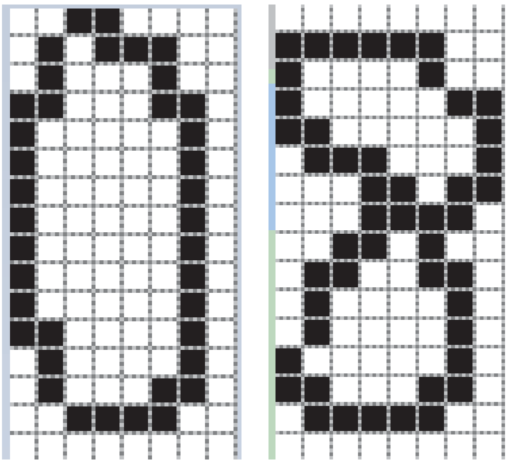
where y_j is the ideal output of the jth neuron, O_j is the real output of the jth neuron, and N_j denotes the jth neuron.

3.3. Grid-Based Feature Extraction. As has been utilized in [34], the grid statistical feature extraction method is very popular. In this paper, the image is divided into 16 × 8 = 128 grids. We use digits from 0 to 9 as the ideal hand gesture. The divided numbers 0 and 8 can be given in Figure 3.

4. Experiment

4.1. Platform Setup. The proposed algorithm fuses the inertial sensor data into attitude quaternion and then computes the velocity and position. Here the position's history can well describe the trajectory of the human hand. In this case, we especially design one experimental platform for such validation. In Figure 4, the tower development platform with NXP Kinetis MK60DN512 microcontroller is presented. The employed platform owns the core processing speed of 100 MHz along with the interfaces of SDIO, SPI, UART, WIFI, and CAN Bus. This allows for the data acquisition and logging of wearable inertial sensors. A miniature inertial sensor module including MPU6000 gyroscope, accelerometer combination, and HMC5983 magnetometer is attached to the designed platform using the RS232 cable free of electromagnetic interference (see Figure 5). The module is in the size of 3 cm × 3 cm, making it much more flexible to be mounted on human's hand. The data polling rate is set to 500 Hz for MPU6000 and 220 Hz for HMC5983. Apart from this, the miniature module can also produce the reference attitude outputs in quaternion. According to the reference manual of this product, the attitude estimation algorithms is effective and accomplished by EKF. In the later comparisons, the results are going to be compared with such reference Euler angles.

4.2. Attitude, Velocity, and Position Estimation. The initial attitude is set as $\mathbf{X}_0 = \mathbf{q} = (1, 0, 0, 0)^T$. The initial variance of the state vector is defined as $\Sigma_{\mathbf{W}_0} = 0.001\mathbf{I}_{4\times4}$. In the particle filtering design, we set $N = 50$ while the threshold is $N_{\text{th}} = N$. The raw sensor data is shown in Figure 6 while



(a) Trace of zero in grid (b) Trace of eight in grid

Figure 3: Illustration of grid-based feature extraction.

Figure 4: Designed hardware for data acquisition from hand gesture sensor.

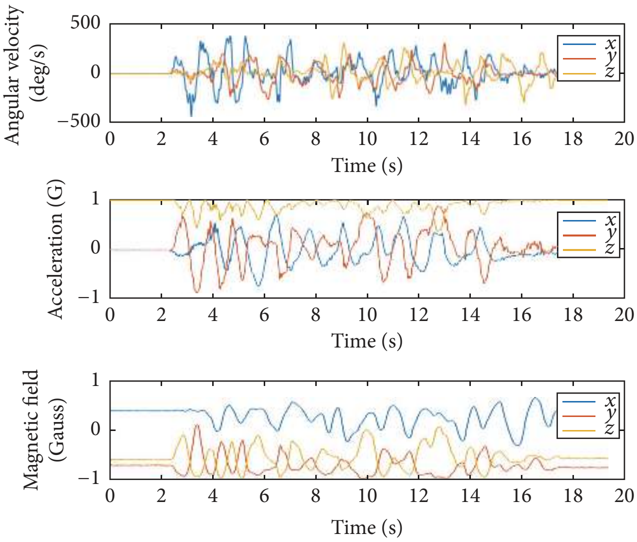


Figure 6: Inertial sensor data from miniature module.

Figure 5: Gesture sensor including 3-axis gyroscope, accelerometer, and magnetometer and reference Euler angles.

the generated attitude outputs are shown in Figure 7. We can see that the proposed attitude estimator has basically the same performance with reference system.

The calculated attitude is then used for velocity and position integral. After the position integral, the trajectory are saved as images. The first 30 trajectories are used as training samples. And the printing character is also added to the set of training samples which can be shown in Line 4 of Figure 8.

4.3. Neural Network. The Neural Network Toolbox of MATLAB is utilized for training the BP-NN. In this section, the Levenberg-Marquardt Method (LMM) and GDM are adopted as training algorithm, respectively. The Mean Square Error (MSE) was used as the evaluation standard of the performance of trained BP-NN. The performances of the BP-NNs are compared with different training algorithms and

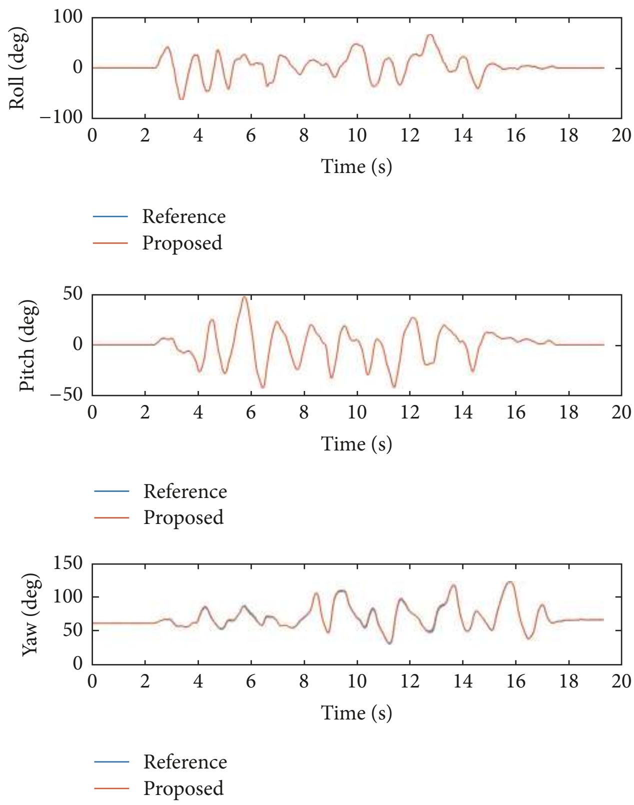


Figure 7: Attitude estimation results.

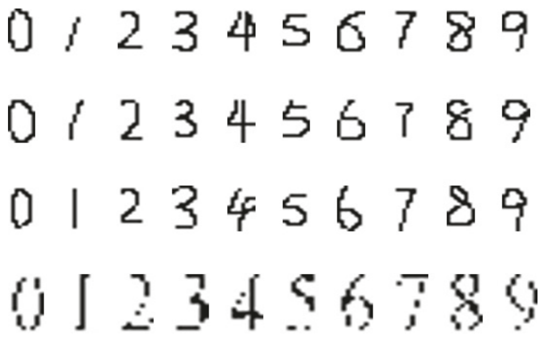


Figure 8: Training samples.

activation functions, which are shown in Figures 9 and 10 and Tables 1 and 2.

As can be seen in Figures 9 and 10, the performance of LMM is better than GDM in this case. MSEs from different combinations are grouped into Table 1.

MSEs of the test data can be given as in Table 2.

Obviously, GDM is not reliable in this paper at all because its success rate is too low for real applications. The STF-LMM combination shows the best performance among all the combinations. The STF-LMM combination can be used for recognition, which is a component of the proposed system.

4.4. Gesture Recognition Results. With the designed system shown above, we make several comparison experiments with a recent representative method. This method is proposed by Xu et al. in which an accelerometer is adopted for gesture recognition [13]. The advantage of the proposed method is that it logs the history gesture movements so that the hand gesture can be determined more accurately. We first generate several gestures with the designed platform and then use the aforementioned parameters and modeled BP-NN to verify the success rate of both methodologies. The general results are summarized in Table 3.

Table 1: Grouped MSEs.

Trn. func.	Act. func.		
	STF	LTF	HTSTF
LMM	$\text{MSE}_{1,1}$	$\text{MSE}_{1,2}$	$\text{MSE}_{1,3}$
GDM	$\text{MSE}_{2,1}$	$\text{MSE}_{2,2}$	$\text{MSE}_{2,3}$

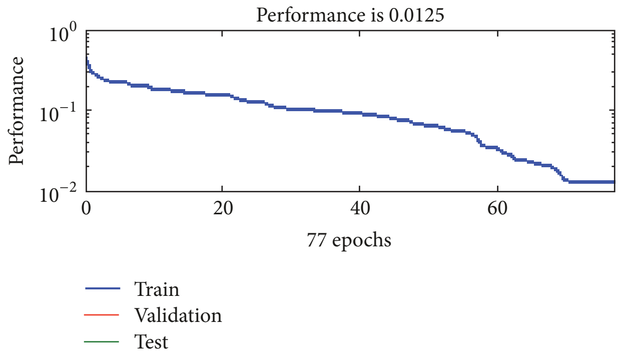


Figure 9: Performance of trained BP-NN using LMM.

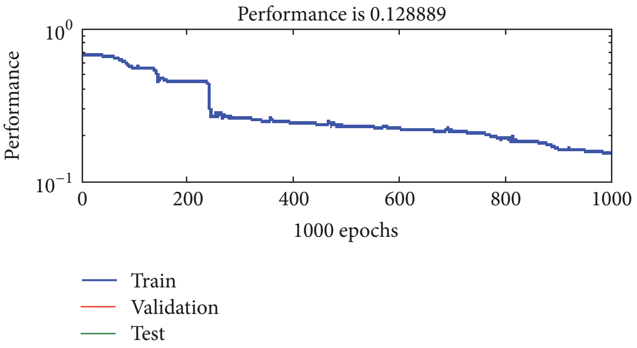


Figure 10: Performance of trained BP-NN using GDA.

We can see that, for some instant gestures, the two methods show not much macroscopic differences. However, when the gesture becomes slow, which relies on the history identification of itself, the proposed method shows much more superiority. This verifies the parameters and models described above and also proves the feasibility and efficiency of the proposed algorithm.

5. Conclusion

In this paper, we propose a hand gesture recognition system that combines inertial sensors and BP-NN together. Rotation

Table 2: Tested MSEs.

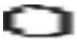
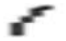
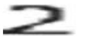
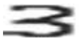
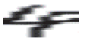
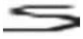
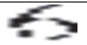
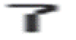
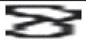
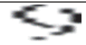
Gesture	0	1	2	3	4	5	6	7	8	9
Trace										
$\text{MSE}_{1,1}$	1.7*e* – 08	2.9*e* – 09	2.8*e* – 17	1.7*e* – 08	6.4*e* – 10	2.2*e* – 10	1.2*e* – 09	5.4*e* – 9	1.6*e* – 10	8.4*e* – 09
$\text{MSE}_{1,2}$	1.1*e* – 01	Wrong	4.8*e* – 02	Wrong	8.1*e* – 02	7.4*e* – 02	Wrong	Wrong	7.0*e* – 02	Wrong
$\text{MSE}_{1,3}$	6.3*e* – 04	1.3*e* – 03	1.0*e* – 03	4.5*e* – 04	1.6*e* – 05	5.0*e* – 03	3.3*e* – 05	1.0*e* – 03	1.5*e* – 03	6.4*e* – 06
$\text{MSE}_{2,1}$	7.8*e* – 02	Wrong	8.9*e* – 02	Wrong	3.7*e* – 02	Wrong	Wrong	Wrong	2.2*e* – 02	Wrong
$\text{MSE}_{2,2}$	9.8*e* – 02	Wrong	1.1*e* – 01	Wrong	Wrong	Wrong	Wrong	Wrong	9.9*e* – 02	Wrong
$\text{MSE}_{2,3}$	1.0*e* – 01	Wrong	Wrong	Wrong	9.7*e* – 02	Wrong	Wrong	Wrong	1.1*e* – 01	Wrong

Table 3: Gesture recognition results.

Algorithm	"One," fast	"One," slow	"Nine," fast	"Nine," slow
Proposed	98.76%	92.43%	89.96%	88.27%
Compared	97.45%	65.22%	90.13%	43.78%

vector method is used to compensate for the noncommutative errors in the determination of attitude, velocity, and position. Particle filtering is introduced to generate accurate attitude outputs from these sensors. Throughout the real-world experiments, the best activation function and training function of the BP-NN are determined successfully. The final tested performance of the BP-NN proved the effectiveness of our method. However, it should be noted that the robustness of our method has not been verified yet. In the future, we will combine different hand gesture recognition methods together in further researches to develop a new robust and more accurate recognition system.

Conflicts of Interest

The authors declare no conflicts of interest regarding this manuscript.

Acknowledgments

This research was supported by National Science Foundation of China (Grant no. 61450010).

References

[1] S. M. Mankar and S. A. Chhabria, "Review on hand gesture based mobile control application," in *Proceedings of the 2015 International Conference on Pervasive Computing (ICPC '15)*, IEEE, Pune, India, January 2015.

[2] V. Bhame, R. Sreemathy, and H. Dhumal, "Vision based hand gesture recognition using eccentric approach for human computer interaction," in *Proceedings of the 3rd International Conference on Advances in Computing, Communications and Informatics (ICACCI '14)*, pp. 949–953, IEEE, New Delhi, India, September 2014.

[3] Z. Xu and H. Zhu, "Vision-based detection of dynamic gesture," in *Proceedings of the 2009 International Conference on Test and Measurement (ICTM '09)*, pp. 223–226, IEEE, Hong Kong, December 2009.

[4] Q. Bai, Y. Zhang, J. Tan, L. Zhao, and Z. Qi, "Recognition of the numbers of numerical civilian instruments based on BP neural network," in *Proceedings of the International Conference on Industrial Machatronics and Automation*, pp. 105–108, IEEE, 2009.

[5] S. Gai, G. Yang, S. Zhang, and M. Wan, "New banknote number recognition algorithm based on support vector machine," in *Proceedings of the 2013 2nd IAPR Asian Conference on Pattern Recognition, ACPR 2013*, pp. 176–180, IEEE, Japan, November 2013.

[6] F. Li and S. Gao, "Character recognition system based on back-propagation neural network," in *Proceedings of the 2010 International Conference on Machine Vision and Human-Machine Interface (MVHI '10)*, pp. 393–396, IEEE, China, April 2010.

[7] R. Szeliski, *Computer Vision*, Springer, 2010.

[8] J. Wu, Z. Zhou, B. Gao, R. Li, Y. Cheng, and H. Fourati, "Fast linear quaternion attitude estimator using vector observations," *IEEE Transactions on Automation Science and Engineering*, 2017.

[9] A. Akl, C. Feng, and S. Valaee, "A novel accelerometer-based gesture recognition system," *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 6197–6205, 2011.

[10] E.-S. Choi, W.-C. Bang, S.-J. Cho, J. Yang, D.-Y. Kim, and S.-R. Kim, "Beatbox music phone: gesture-based interactive mobile phone using a tri-axis accelerometer," in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT '05)*, pp. 97–102, IEEE, Hong Kong, December 2005.

[11] J. Rao, T. Gao, Z. Gong, and Z. Jiang, "Low cost hand gesture learning and recognition system based on hidden markov model," in *Proceedings of the 2009 2nd International Symposium on Information Science and Engineering (ISISE '09)*, pp. 433–438, IEEE, Shanghai, China, December 2009.

[12] R. Xie, X. Sun, X. Xia, and J. Cao, "Similarity matching-based extensible hand gesture recognition," *IEEE Sensors Journal*, vol. 15, no. 6, pp. 3475–3483, 2015.

[13] R. Xu, S. Zhou, and W. J. Li, "MEMS accelerometer based nonspecific-user hand gesture recognition," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 1166–1173, 2012.

[14] A. I. Bhuyan and T. C. Mallick, "Gyro-accelerometer based control of a robotic arm using AVR microcontroller," in *Proceedings of the 9th International Forum on Strategic Technology (IFOST '14)*, pp. 409–413, IEEE, Cox's Bazar, Bangladesh, October 2014.

[15] Z. Lu, X. Chen, Q. Li, X. Zhang, and P. Zhou, "A hand gesture recognition framework and wearable gesture-based interaction

prototype for mobile devices," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 2, pp. 293–299, 2014.

[16] Y.-L. Hsu, C.-L. Chu, Y.-J. Tsai, and J.-S. Wang, "An inertial pen with dynamic time warping recognizer for handwriting and gesture recognition," *IEEE Sensors Journal*, vol. 15, no. 1, pp. 154–163, 2015.

[17] S. Zhou, F. Fei, G. Zhang et al., "2D human gesture tracking and recognition by the fusion of MEMS inertial and vision sensors," *IEEE Sensors Journal*, vol. 14, no. 4, pp. 1160–1170, 2014.

[18] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[19] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[20] M. Matsugu, K. Mori, Y. Mitari, and Y. Kaneda, "Subject independent facial expression recognition with robust face detection using a convolutional neural network," *Neural Networks*, vol. 16, no. 5-6, pp. 555–559, 2003.

[21] C. Ye, S. Hong, and A. Tamjidi, "6-DOF Pose Estimation of a Robotic Navigation Aid by Tracking Visual and Geometric Features," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1169–1180, 2015.

[22] Z. Zhou, Y. Li, J. Liu, and G. Li, "Equality constrained robust measurement fusion for adaptive kalman-filter-based heterogeneous multi-sensor navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2146–2157, 2013.

[23] J. Wu, Z. Zhou, J. Chen, H. Fourati, and R. Li, "Fast complementary filter for attitude estimation using low-cost MARG sensors," *IEEE Sensors Journal*, vol. 16, no. 18, pp. 6997–7007, 2016, http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7508437=0pt.

[24] Y. Shi, W. Fan, and G. Shi, "The research of printed character recognition based on neural network," in *Proceedings of the 2011 4th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP '11)*, pp. 119–122, China, December 2011.

[25] P. G. Savage, *Strapdown Analytics*, Strapdown Associates Inc., 1997.

[26] J. E. Bortz, "A new mathematical formulation for strapdown inertial navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-7, no. 1, pp. 61–66, 1971.

[27] R. B. Miller, "A new strapdown attitude algorithm," *Journal of Guidance, Control, and Dynamics*, vol. 6, no. 4, pp. 287–291, 1983.

[28] P. G. Savage, "Strapdown inertial navigation integration algorithm design. Part 1. Attitude algorithms," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 1, pp. 19–28, 1998.

[29] P. G. Savage, "Strapdown inertial navigation integration algorithm design part 2: velocity and position algorithms," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 208–221, 1998.

[30] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of the American Statistical Association*, vol. 93, no. 443, pp. 1032–1044, 1998.

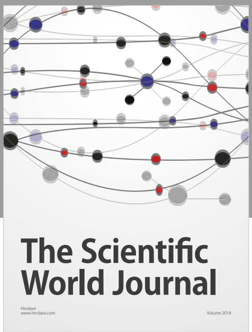
[31] A. M. Sabatini, "Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 7, pp. 1346–1356, 2006.

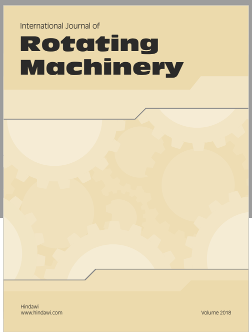
[32] R. G. Valenti, I. Dryanovski, and J. Xiao, "A linear Kalman filter for MARG orientation estimation using the algebraic quaternion algorithm," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 2, pp. 467–481, 2016.

[33] Z. Wang, H. Zhao, S. Qiu, and Q. Gao, "Stance-Phase Detection for ZUPT-Aided Foot-Mounted Pedestrian Navigation System," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 6, pp. 3170–3181, 2015.

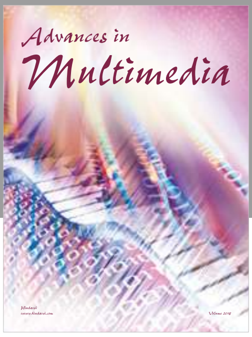
[34] Y. Zhang, S. Xie, and S. Wei, "Industrial character recognition based on grid feature and wavelet moment," in *Proceedings of the 2013 IEEE International Conference on Imaging Systems and Techniques (IST '13)*, pp. 56–59, IEEE, Beijing, China, October 2013.

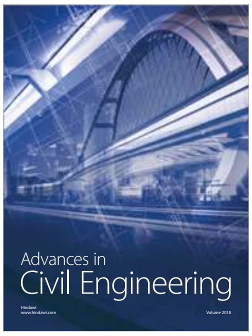
Journal of
Engineering

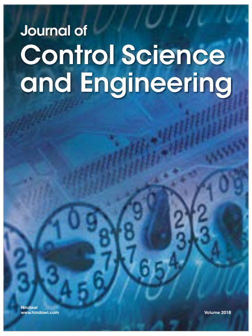

The Scientific
World Journal


International Journal of
Rotating
Machinery

Journal of
Sensors

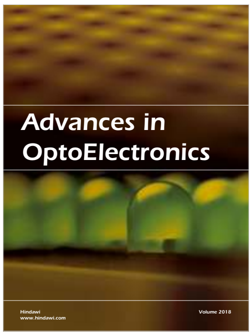

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

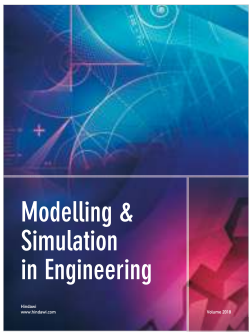
Journal of
Robotics

Journal of
Electrical and Computer
Engineering


Advances in
OptoElectronics

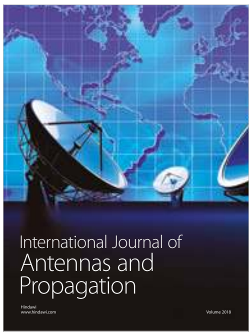
VLSI Design

International Journal of
Navigation and
Observation

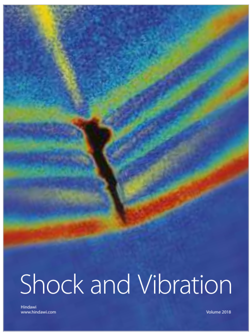

Modelling &
Simulation
in Engineering

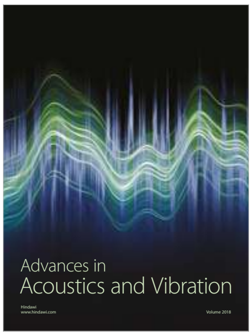
International Journal of
Aerospace
Engineering

International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration