

Quaternionic Attitude Estimation for Robotic and Human Motion Tracking Using Sequential Monte Carlo Methods With von Mises-Fisher and Nonuniform Densities Simulations

Gary To, *Member, IEEE*, and Mohamed R. Mahfouz*, *Senior Member, IEEE*

Abstract—In recent years, wireless positioning and tracking devices based on semiconductor micro electro-mechanical system (MEMS) sensors have successfully integrated into the consumer electronics market. Information from the sensors is processed by an attitude estimation program. Many of these algorithms were developed primarily for aeronautical applications. The parameters affecting the accuracy and stability of the system vary with the intended application. The performance of these algorithms occasionally destabilize during human motion tracking activities, which does not satisfy the reliability and high accuracy demand in biomedical application. A previous study accessed the feasibility of using semiconductor based inertial measurement units (IMUs) for human motion tracking. IMU hardware has been redesigned and an attitude estimation algorithm using sequential Monte Carlo (SMC) methods, or particle filter, for quaternions was developed. The method presented in this paper uses von Mises-Fisher and a nonuniform simulation to provide density estimation of the rotation group $SO(3)$. Synthetic signal simulation, robotics applications, and human applications have been investigated.

Index Terms—Motion measurement, sequential Monte Carlo (SMC) methods, wearable sensors.

I. INTRODUCTION

MOTION tracking is an important asset in various areas of medical studies and applications. It is one of the keys to understanding kinematics and kinetics of the human body. Diagnostic instrumentation with this technology can provide valuable information to physicians regarding patients' conditions. Medical device researchers and engineers also use motion analysis to study the biomechanics of the human joint and to design better implants and prostheses.

Optical tracking is one of the most accurate dynamic tracking systems and is commonly used as diagnosis devices for joint motion analysis. However, these systems are not typically avail-

able in a clinic or hospital. Second, these systems generally require trained technicians to setup and operate, which is inconvenient as a means of a day-to-day diagnostic tool. Additionally, optical systems have been used extensively in surgical robotics and computer-assisted surgical navigation for minimum invasive surgeries. The most notable limitation of these systems is that the target must remain in a clear line of sight to the camera.

In order to provide an alternative means for robotics motion tracking and joint kinematics assessment, researchers have been developing new tracking systems using different technologies. Inertial measurement units (IMUs) have a wide range of applications ranging from automobile to space flight navigation. In 1996, Veltink and Boom [1] published the initial concept of using uniaxial accelerometer for motion analysis. Bouten *et al.* [2] used triaxial accelerometer to analysis daily activities such as standing and walking. Clinical studies have been performed for posture estimation using accelerometers [3]–[5]. Mayagoita uses a combination of accelerometers and gyroscopes to track the lower extremities in two-dimensional (2-D) sagittal plane in 2002 [6]. Inertial motion tracking have also been used in kinematic studies, such as gait analysis [7]–[10]. In 2006, Roetenberg [11] used a set of triaxial accelerometers, gyroscopes, and magnetometers to monitor human motion in his dissertation. There are multiple studies using IMUs for motion tracking in virtual and augmented reality environment [12]–[14]. In recent years, optical and inertial hybrid systems have been used in various surgical navigation applications [14]–[17].

Quaternionic attitude estimation algorithms have been the focus of research for various navigation applications. There are multiple implementations and variations based on the extended Kalman filter (EKF) using different estimations and optimization approaches [17]–[21]. The main challenge for EKF is determining the error covariance of the quaternion, while maintaining the unit norm constraint. Shuster, Bar-Itzhack, and Choukroun demonstrated the integration of the quaternion estimation (QUEST) model with Kalman filter [21]–[24]. Crassida and Markley [25] developed a quaternionic unscented Kalman filter (UKF) by transforming the quaternion to modified Rodriguez parameters (MRPs) prior to unscented transform. The same transformation technique was employed to perform quaternionic attitude estimation with sequential Monte Carlo (SMC) method, or particle filter (PF), as demonstrated by Cheng and Crassidis [26]. Carmi [27] used a genetic algorithm for likelihood approximation to improve the performance of the quaternion

Manuscript received November 5, 2012; revised March 30, 2013; accepted May 6, 2013. Date of publication May 13, 2013; date of current version October 16, 2013. This work was supported in part by the Center for Musculoskeletal Research, in part by the University of Tennessee, and in part by JointVue LLC. Asterisk indicates corresponding author.

G. To is with the Institute of Biomedical Engineering, University of Tennessee, Knoxville, TN 37916 USA (e-mail: gto@utk.edu).

*M. Mahfouz is with the Institute of Biomedical Engineering, University of Tennessee, Knoxville, TN 37916 USA (e-mail: mmahfouz@utk.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TBME.2013.2262636

PF. Recently, an algorithm has been developed for the optical and inertial hybrid tracking system with a combination of Kanade Lucas tracker (KLT) [28] and EKF [14], [29].

In recent years, consumer electronics have employed a significant amount of semiconductor-based tracking systems to provide a different kind of interface control that uses body motion and gesture. However, these devices are not typically designed for medical applications that demand high accuracies and repeatability. In a previous study, off-the-shelf IMUs and an IMU prototype were tested [30]. The data were processed with a quaternionic EKF, and the outputs of the two designs were compared to an optical tracking system. It was concluded that both of the designs were insufficient for medical application due to hardware limitations and signal-processing-related issues. In this research, a modular design of the IMU system was implemented. The design introduces redundant sensor ports to provide flexibility to the tracker for different activities and to allow multiple duplicate sensor inputs. A novel PF was designed to address the stability issues observed in the previous experiment. This method addresses the statistical challenges of quaternion's unit norm constraint by using high-dimensional directional statistical distribution to sample and generate quaternions.

The primary goal for this research is to achieve a robust, accurate, and stable orientation tracking with IMU that can be used in surgical robotics and diagnostic instrumentations. These are demonstrated via experiments with robots and free-hand motion tracking. The hardware component is discussed in Section II. In Section III, the challenge of performing statistical analysis for quaternion is examined and a solution of using hyperdimensional statistical geometries is proposed. The formulation of the SMC algorithm is discussed in detail. The performance of the algorithm from simulation, robotic, and human motion tracking are presented in Section IV. In Section V, the potential and future designs are discussed.

II. SYSTEM ARCHITECTURE

The complexity of the kinematics of human body motion is one of the major concerns with inertial-based tracking systems. Physical limitation of the joint does not limit the degree of freedom experienced by the IMU in the inertial frame. Human joint segments have high dynamic range and are capable of instantaneously reversing or changing the direction of motion as a result of the complicity of multibody dynamics.

An IMU is composed of a set of triaxial accelerometers, gyroscopes, and magnetometers. There are many designs on these sensors, which offer different dynamic ranges and sensitivities. A modular design is implemented to allow sensors with different specifications to connect into the same system in order to increase the sensitivity of the system to handle different levels of activities in human motion. This is achieved by separating the sensing components from the electronics. The current design has three sensor ports, which can support up to six individual sensing components, as shown in Fig. 1.

There are two types of sensor strips. The first type consists of accelerometers (MMA7361 L, freescale technology) and gyroscopes sensors (LPY510AL, LPR510AL, ST-Microelectronics).

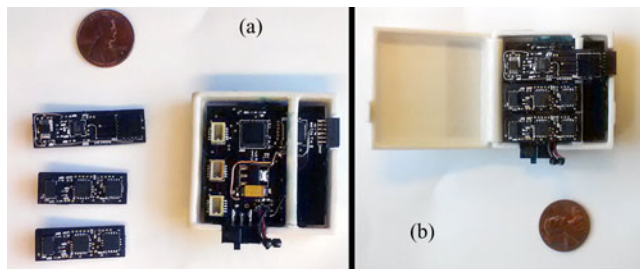


Fig. 1. (a) Different configurations of sensor strips and the electronics component. (b) Assembled IMU.

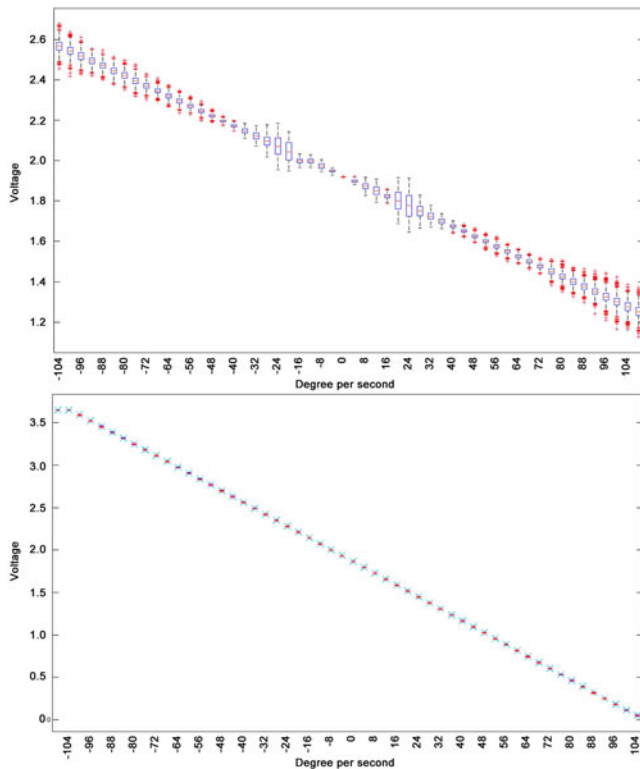


Fig. 2. Box plots of the X-axis of the gyroscopes from an (top) off-the-shelf system and (bottom) current system undergoing rotations at different angular velocities, where each box shows the mean and variance of the gyroscope output over the period of 5000 samples.

It has two configurations. The user can change the sensitivity of the gyroscopes between $\pm 100^\circ$ and $\pm 400^\circ$ per second (dps). The signal conditioning circuit is an offset amplifier that is designed to condition the sensor outputs to the input range of the analog-to-digital converter (ADC). The second type is a triaxial magnetometers (HMC1053, Honeywell). A multichannel 24-bit ADC was selected for this design (ADS1258, Texas Instrument), which provides 19.5 effective number of bits at the highest conversion speed setting. The power module was re-designed to provide a more stable reference. The current system has dramatically improved the digitization performance of the system, as shown in Fig. 2.

A low-power microcontroller (MSP430F2274, Texas Instrument) was used as the processor. A compact wireless transmitting module (A2500R24 A, Anaren) was used for communication. The receiver uses a predetermined device

identification number to process the received data from multiple IMUs. The data rate is approximately 90 Hz for a single IMU and decreases as more IMUs join the shared network. The current IMU system is powered by either two CR2032 coin cell batteries or a pack of 200 mA·h rechargeable LiFe batteries. It consumes, on average, 70 mA with the current sensors configuration and can be used continuously for approximately 2 hours.

III. ORIENTATION TRACKING

During the motion tracking activity, 15 sets of data from various sensors are sent from each IMU. The fundamental principle of the attitude estimation algorithm design is to utilize multiple types of sensors, which are insufficient for orientation tracking on their own, to produce accurate and reliable orientation estimation. There is extensive research using the Kalman family estimators. Many of them revolve around the optimization or the determination of error covariance for the quaternion via various projection methods or ad hoc techniques, such as modifying and restoring the quaternion to maintain the unit norm property. SMC method or PF does not use error covariance directly in either prediction or the update processes. It estimates an approximation of the posterior filtering density, and samples a finite number of “particles” for both processes instead. The challenge of formulating the PF for quaternion is to generate random quaternion with the desired statistical properties, which can be accomplished by using multidimensional Stiefel manifolds described in the following section.

A. Stiefel Manifold

The topological space of a collection of p -dimensional orthonormal vectors in N -dimensional space is considered as the basis of the Stiefel manifold. A quaternion (q), where $p = 4$ and $N = 3$, satisfies such condition and forms a unique case on the manifold (V_p)

$$V_p(\mathbb{R}^N) = \{q \in \mathbb{R}^{N \times p} : q^* \otimes q = I_p\} \\ \otimes : \text{quaternion multiplication}, \quad I_p : [1 \ 0 \ 0 \ 0] \quad (1)$$

where q^* is the complex conjugate of q and \mathbb{R} is the inner product space. Statistical densities reside on the Stiefel manifold encompasses any arbitrary dimensional objects in any dimensional space. Density that satisfies the condition of ($p < N$) can be considered. In this paper, the von Mises-Fisher (vMF) density and a nonuniform (NU) density based on Bingham density were investigated and were used for sampling the quaternion particles.

B. vMF Density

The vMF density is a generalized spherical distribution of a p -dimensional object in $(p - 1)$ dimensional sphere. The probability density function of a generalized vMF density is defined as [31]

$$f_{\text{vMF}}(X; F) = \frac{1}{a(F)} e^{\text{tr}(FX^T)} \quad (2)$$

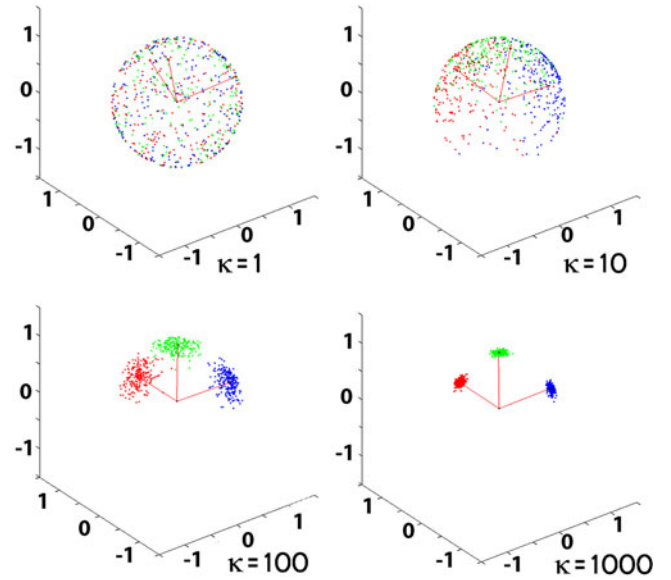


Fig. 3. Random quaternions sampled from different vMF densities via simulation Algorithm 1. The red lines represent the mean direction of the generated samples. As the dispersion factor increase, the sample concentration increased.

where X is a $p \times N$ matrix of orthogonal unit vectors, F is a $p \times N$ parameters matrix, and $1/a(F)$ is the normalizing constant, which can be expressed by the confluent hypergeometric limit function

$$a(F) = {}_0F_1\left(\frac{N}{2}, \frac{FF^T}{4}\right). \quad (3)$$

The density for a quaternion can be written as

$$f_{\text{vMF}}(x; \mu, \kappa) = C_4(\kappa) e^{(\kappa \mu^T x)}, \quad C_4 = \kappa / 2\pi^2 I_1(\kappa) \quad (4)$$

where x is a random quaternion, μ is the mean vector, κ is the dispersion factor, and I_1 is the first order of the modified Bessel function of the first kind.

It is often inefficient and impractical to sample directly from the density. Random samples that have the properties of the desired density can be simulated based on the parameters of the density. Random sampling from the vMF density is achieved by the simulation algorithm by Wood [32] (see Algorithm 1 in the Appendix). This simulation algorithm is an efficient method to create a set of quaternions based on the κ and μ . Fig. 3 shows four sets of random sampled quaternion from the vMF densities with different κ at $\mu = [1 \ 0 \ 0 \ 0]$. It is important to note that the distributions in Fig. 3 are not the full projection of the vMF densities. It is only one instance of the density projected to three dimensions with an identity matrix. It is analogous to view a 2-D “slice” of a 3-D distribution.

C. NU Distribution

The vMF density assumes the samples are uniformly dispersed around a rotation manifold. However, rotational symmetry and uniformity assumption is not always true in the physical system. It is useful to consider NU density for quaternions. Bingham density provides flexibility to create elliptic or girdle statistical geometries. The probability density function for

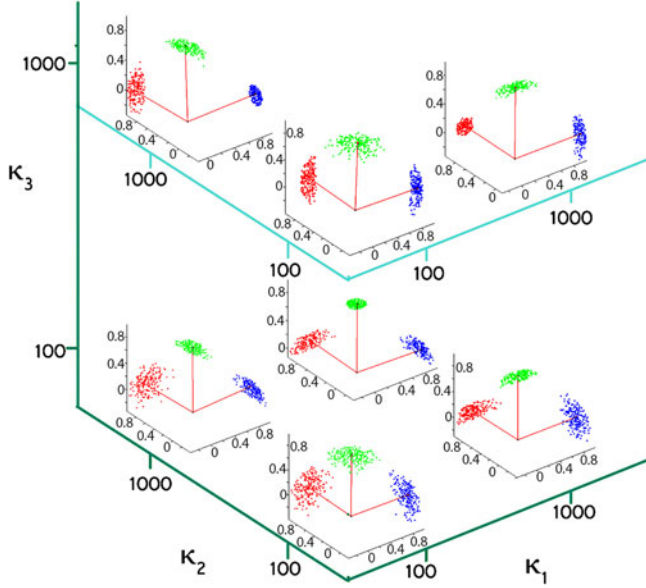


Fig. 4. Random quaternions sampled from NU density via simulating Algorithm 2. The red lines represent the mean direction of the samples. The K matrix controls the density proportions of the distribution.

Bingham density is

$$f_B(\pm q; K) = {}_1F_1\left(\frac{1}{2}, \frac{p}{2}, q\right)^{-1} e^{q^T U K U^T q} \quad (5)$$

where q is the quaternion describing the orientation, ${}_1F_1((1/2), (p/2), q)^{-1}$ is Kummer's function of the first kind as normalizing constant, U is an orthogonal matrix describing the orientation of the distribution, and K is a diagonal matrix that describes the dispersion axes of the distribution. An algorithm was designed to perform rejection sampling to create a set of random simulation samples from the distribution (see Algorithm 2 in the Appendix).

Due to quaternion's antipodal property, quaternion and its conjugate will result with the same probability density of the distribution on step 12 in Algorithm 2. Thus, sampling with complete random hyper-spheres (see steps 1–10 of Algorithm 2) must be done with care if the desired application is sensitive to such difference. In the formulation of PF for attitude estimation, random vMF quaternions with large dispersion were used as initialization of the sampling to create NU density (see step 11 of Algorithm 2). This reduces the time required to generate sample because it limits the seeking space of the density. Secondly, it eliminates the possibility of sampling the conjugates, which affects the expectation calculation. It should be noted that the samples calculated from this method are not strictly Bingham density as the antipodal properties has been removed. Fig. 4 shows the relationship between the different parametric ratios in the dispersion shape matrix K

$$K = \text{diag}[\kappa_s \kappa_1 \kappa_2 \kappa_3]. \quad (6)$$

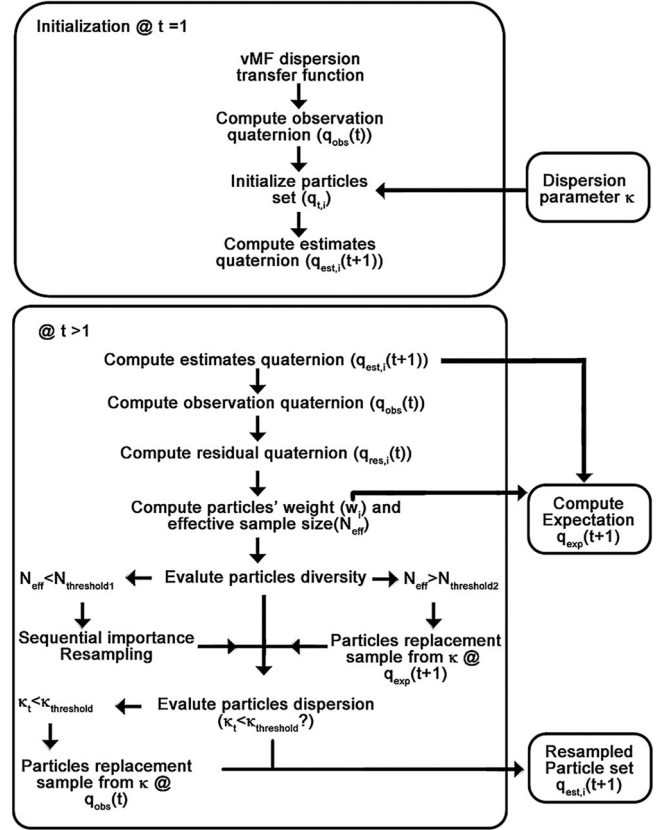


Fig. 5. Functional flow of the quaternionic PF algorithm using vMF density.

D. Particle Filtering (vMF Density)

The recursive Bayesian algorithm is a probabilistic model that computes the posterior probability density function of an unknown process and uses it in the estimation calculation. It generally involves two-stage processes of state prediction and state update to resolve the posterior density. The posterior probability $p(x_t|y_t)$ is defined as

$$p(x_t|y_t) = \frac{p(y_t|x_t)p(x_t|y_{t-1})}{p(y_t|y_{t-1})} \quad (7)$$

$$p(x_t|y_{t-1}) = \int^p (x_t|x_{t-1})p(x_{t-1}|y_{t-1})dx_{t-1} \quad (8)$$

$$p(y_t|y_{t-1}) = \int^p (y_t|x_t)p(x_t|y_{t-1})dx_t \quad (9)$$

where $p(y_t|x_t)$ is the likelihood, $p(y_t|x_{t-1})$ is the prior probability, $p(y_t|y_{t-1})$ is the normalizing constant, and $p(x_t|x_{t-1})$ is the transitional probability density.

The Monte Carlo method can be considered as a brute force approximation for an intractable inference problem with a large sum of random samples from the same probability density space. This technique forms the fundamental concept of particle filtering. In this section, the implementation of the PF with vMF density is discussed. The overall flow diagram is illustrated in Fig. 5.

Since PF overcomes the limitation of solving the intractable inference problem by using a set of random samples that

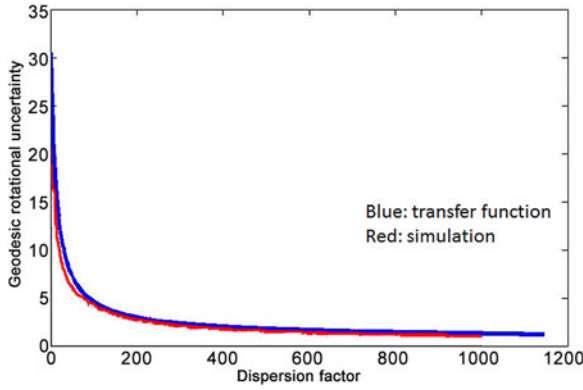


Fig. 6. Estimation state particle spread and dispersion parameter approximation function. Red line indicates the relationship established during simulation. The blue line is created by giving the particle spread to the function to calculate the dispersion parameter.

approximate the characteristics of the posterior filtering distribution density with associated weights for estimation, it is important to establish a method to evaluate and generate particles based on the posterior distribution density that reflect the state of the estimation. It can be achieved by first evaluating and establishing a correlation between the uncertainty of the random particles sampled at $q_0 = [1 \ 0 \ 0 \ 0]$ with different κ . The spread of the particles around q_0 can be evaluated by calculating the geodesic angle (δ) between two vectors as follows:

$$\delta_i = 2 * \text{acos}(|q_{x,i}^\kappa \cdot q_0|)$$

$$q_{x,i}^\kappa \sim \text{vMF}(\kappa), \quad i = 1 \rightarrow N, \quad \kappa = \kappa_{\min} \rightarrow \kappa_{\max}. \quad (10)$$

The root sum square of δ of N -particle rotational spread represents the particles' rotational uncertainty, which can be correlated to the dispersion parameter by simulating the distribution with a series of different κ . Least squares approximation is then used to realize the function between the uncertainty of the estimated particles to the posterior dispersion parameter [see (11)], as shown in Fig. 6

$$f \left(\sqrt{\sum_i^N \delta_i^2} \right) = \kappa. \quad (11)$$

During the initialization of the algorithm, the initial observation quaternion $q_{\text{obs}}(t)$ was first calculated from the outputs of the accelerometers and magnetometer sensors using Gauss Newton optimization method (see Algorithm 3 in the Appendix). Quaternionic particles $q_{\text{est},i}(t)$ are generated with the method described in Algorithm 1 with $q_{\text{obs}}(t)$ and an arbitrary user-defined κ as follows:

$$q_{\text{est},i}(t) \sim \text{vMF}(\kappa) \otimes q_{\text{obs}}(t), \quad i = 1, \dots, N. \quad (12)$$

The initialized particles have equal weights. In the subsequent cycles, the weights and the κ describing the posterior filtering density are used. The algorithm then computes the estimates of the particles for the next iteration $q_{\text{est},i}(t+1)$. The prediction of the particles is performed by integrating the quaternion derivative calculated from the outputs from the gyroscopes in

the IMU frame as follows:

$$q_{\text{est},i}(t+1) = q_{\text{est},i}^\kappa(t) + 0.5 \left(q_{\text{est},i}^\kappa(t) \otimes [0 \ \omega_x \ \omega_y \ \omega_z] \right) \Delta t, \quad i = 1, \dots, N \quad (13)$$

where ω are the angular rate measured at time t and Δt is the sampling period.

The recursive section of the algorithm is engaged after the initialization, which first determines $q_{\text{obs}}(t)$ at current time and computes the predictions of the next iteration $q_{\text{est},i}(t+1)$ using Algorithm 3 and (13), respectively. The posterior filtering distribution and the weights are determined as follows. The quaternion residuals $q_{\text{res},i}(t)$, which are the disparities between the estimations from the previous cycle and the current observation, is first calculated in (14). The objective of this algorithm is to minimize the residual using particles to approximate the posterior probability density of the system. For quaternionic geometry, the residual of the optimal estimation is equivalent to [1000], which is no error. Hence, the geodesic differences between the residual quaternion and q_0 can be used to infer the weights for the particles [see (15) and (16)]. With this method, the residual particle with less rotational estimation error receives higher weight, and vice versa

$$q_{\text{res},i}(t) = q_{\text{est},i}(t) \otimes \text{conj}(q_{\text{obs}}(t)), \quad i = 1, \dots, N \quad (14)$$

$$\delta_{\text{res},i} = 2 \cos(q_{\text{res},i}(t) \cdot q_0) \quad (15)$$

$$w_i = \frac{1/\delta_{\text{res},i}}{\sum_i^N (1/\delta_{\text{res},i})}. \quad (16)$$

The dispersion parameter for the posterior distribution density can be found with δ_i and the function illustrated in (11). The expectation of the filtered quaternion $q_{\text{exp}}(t+1)$ is calculated from the estimation particles $q_{\text{est},i}(t)$ and their associated weights (see Algorithm 4 in the Appendix). This algorithm uses spherical linear interpolation (SLERP) [33] in successive intervals. SLERP computes an intermediate quaternion from the two input quaternions and a weight parameter between 0 and 1. If the parameter is set at 0.5, it is equivalent to computing the average of the two rotations. In the presented algorithm, each particle has its associated weight. The weights are first normalized (see step 3 of Algorithm 4) before SLERP is applied. The weight of the output quaternion is determined from the ratio of the normalized weight (see step 6 of Algorithm 4). Each cycle (steps 2–8) reduces the total number of particles by half until only one output is left. This technique is limited to processing 2^n particles. Zero padding the particle estimates and weights are required for other particles size in this step.

In PF algorithm, particles maintenance is a method to prevent particles degeneracy and improve their effectiveness. Importance resampling improves the posterior density by resampling the particles based on their weights. The resampling scheme is to compare effective sample size (N_{eff}) to a predetermined threshold (N_{th}) to trigger resampling. If the condition is met, the algorithm resamples the particles based on the weights and the posterior probability density. The effective sample size is

approximated by [34]

$$N_{\text{eff}} = \frac{1}{\sum_i^N w_i^2}. \quad (17)$$

As the diversity of the particle diminishes over time with importance resampling, new particles must be introduced to replace the degenerated particles. There are two scenarios that require particle maintenance in this algorithm. The first instance is the importance resampling method mentioned earlier, which is triggered by predetermined thresholds. If N_{eff} is smaller than N_{th1} , importance resampling is performed with one of the following schemes: deterministic [35], residual [36], or auxiliary [37]. A second threshold (N_{th2}) was set to be triggered by low particles diversity. If N_{eff} is larger than N_{th2} , the particles population is resampled with the posterior density and the expectation quaternion as follows:

$$q_{rs,i}(t) \sim \mathbf{vMF} \left(f \left(\sqrt{\sum_i^N \delta_{\text{res},i}^2} \right) \right) \otimes (q_{\text{exp}(t+1)}). \quad (18)$$

The second scenario is relevant to the hardware system. There are several instances that cause temporary interruption or interference to the system, which leads to faulty outputs. For example, magnetic interference from an external source can cause a disruption to the magnetometers; motions temporary exceeding the operating range of the sensor results in signal saturation; or time lag during wireless transmission. As the estimation is temporal dependent, the differences between prediction and observation can be misrepresented, and the computation of the posterior density is affected. Since N_{eff} only uses normalized weights to estimate the diversity of the particles, it does not directly observe the conditions of the estimation and observation states. Events mentioned earlier may lead to wide particle dispersion in the subsequent states, which can either converge over time or destabilize the filter from diverging to uniformly distributed particles.

The posterior density κ infers the state of uncertainty of the residual particles. For instance, the posterior density increases the uncertainty of the dispersion in its estimation if a large disparity is observed between the residual particles and q_0 . Hence, a dispersion threshold κ_{th} was set up to prevent extreme divergence from faulty or corrupted signals. If the posterior density calculated from $\delta_{\text{res},i}$ is less than κ_{th} , a new set of particles are sampled using $q_{\text{obs}}(t)$ as the mean direction and the posterior density defined by the user in the initialization stage.

E. Particle Filtering (Nonuniform Distribution)

The overall process of quaternionic PF for NU density is similar to the vMF density as shown in Fig. 7. The primary difference is the rejection sampling mechanism. In this algorithm, the user defines the dispersion shape matrix, K , which governs the particle dispersion pattern and uses density bounds to control samples dispersion. This leads to a very different particle maintenance strategy. While the diversity of the particle can still be monitored with (17), the previous method cannot be used to evaluate particle dispersion. Unlike the vMF algorithm, the amount

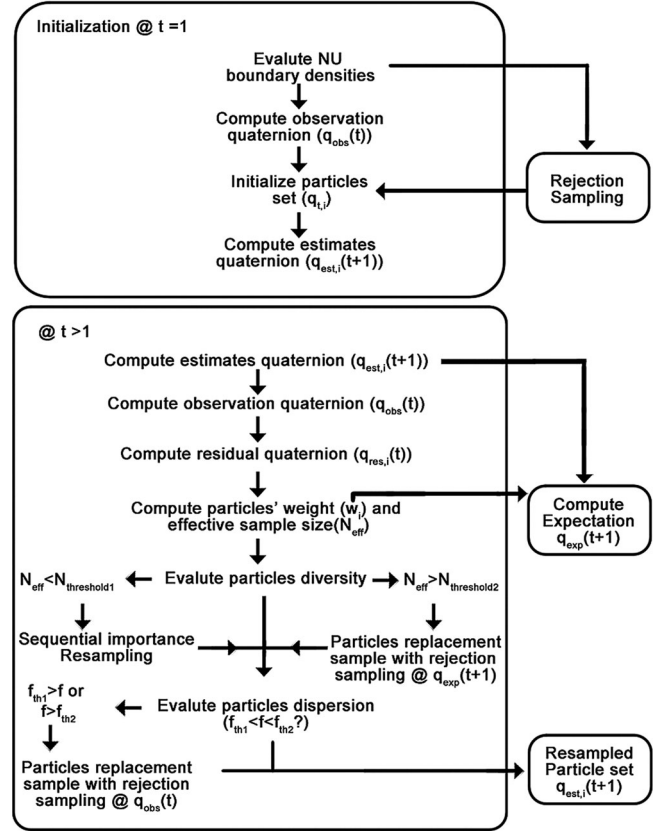


Fig. 7. Functional flow of the quaternionic PF algorithm using non uniform distribution.

of particle dispersion is controlled by the maximum and minimum rejection sampling density boundaries, f_{max} and f_{min} , respectively. They can be determined by evaluating the residual particle density using step 12 in Algorithm 2. The equation is approximated as the Kummer's function is constant in this configuration. The advantage of using residual particles calculated from (14) in evaluating the uncertainty state of the estimation is that the optimal μ or the attitude of the residual particles distribution (U) does not change. The posterior density of each particle is evaluated as follows:

$$f_i = \exp \left(q_{\text{res},i}(t)^T K q_{\text{res},i}(t) \right), \quad i = 1 \rightarrow N. \quad (19)$$

NU density may suffer from similar faulty hardware related problem discussed in previous PF design. Therefore, two predetermined densities (f_1 and f_2 , $f_1 < f_2$) boundaries thresholds were set to prevent distribution divergence. The boundary densities f_{max} and f_{min} must fall within f_1 and f_2 . Particles with f_i falling outside that range are resampled with new particles drawn from the posterior distribution within the initial boundary densities.

F. Particle Filtering and Bias Compensation

The major source of error in attitude estimation system is the null drift bias error of the gyroscopes caused by thermo variations. Filters designed from the Kalman family include the drift as part of the system model, and can estimate and

compensate the drift bias. Mahony [38] devised a simple and elegant approach in the complimentary filter by monitoring the error estimation feedback.

The PF designs discussed earlier monitor the null bias drift via the posterior distribution density as it is based on the residual error between the estimation and observation. However, if a large population of particles is resampled either to increase the diversity or correct the dispersion, the relationship between the posterior density and null drift estimation may be temporary shifted or lost. The bias correction method in [38] was used in the PF algorithms to provide stability and preserve the mean direction of the particle spread when a large sum of particles is resampled.

IV. EXPERIMENTAL RESULTS

The performance of the PFs was evaluated with four sets of experiments: verification studies with synthetic signals, a stationary experiment, robotic applications, and a validation study with human motion applications.

In the first experiment, synthetic signals simulating the output from the sensors were used to evaluate the algorithms. The system dynamics model simulates the IMU rotating around each of its axis at constant speed [see (20)–(22)]. Although the sensor noise model assumes the noise (η) to be Gaussian on each channel, it does not restrict either the mean or the variance to be identical constants on each axis. This enables the simulation of a nonlinear non-Gaussian quaternionic system with different functions and parameters

$$G = w_i + \eta_{g,i} \quad (20)$$

$$A = a_i + \eta_{a,i} \quad (21)$$

$$M = [\sin \theta \quad \cos \theta \quad 1] + \eta_{b,i} \quad (22)$$

$$i = x, y, z; \quad \eta_i \sim \prod (\mu_i, \sigma_i).$$

The dataset was set with noise several fold higher than the sensors' specifications to simulate and verify the performance of the algorithms with a noisy environment such as the musculoskeletal system. It is then processed by different algorithms with various particle populations and importance resampling methods, which are summarized as follows:

- 1) RAW: Raw noisy data
- 2) EKF: Extended Kalman Filter
- 3) CF: Complementary Filter
- 4) SMC: SMC method
 - a) N (Particle size): 128, 256, 512, 1024
 - b) Importance resampling:
 - i) Deterministic sampling (DET)
 - ii) Residual resampling (RES)
 - iii) Auxiliary resampling (AUX)
 - c) Method:
 - i) vMF density (vMF)
 - ii) Nonuniform density (NU)
 - d) With or without bias correction (BC)

The results are projected as Euler angles to compare with the intended noise free motion trajectory. Fig. 8 shows the error

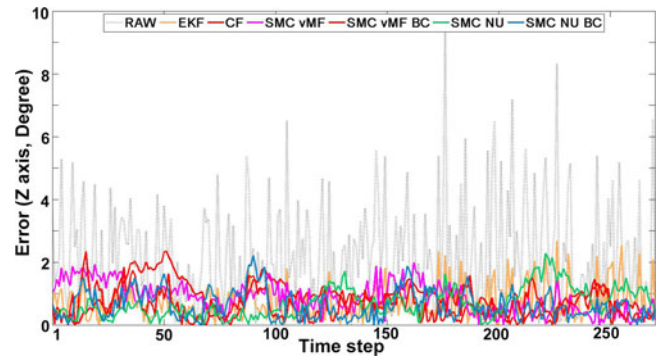


Fig. 8. Difference between the data output processed by different algorithms and the intended motion trajectory.

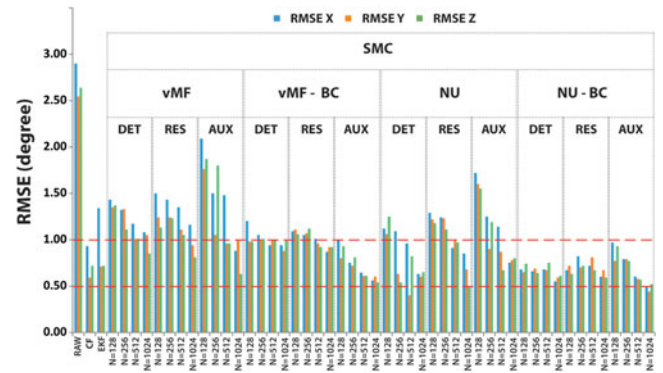


Fig. 9. RMSE of each axis processed by different algorithms and settings.

of the Z -axis among the data processed by 512 particles with auxiliary resampling method.

The averages of the root-mean-squared error (RMSE) of each axis over 200 simulations are shown in Fig. 9. The result shows that regardless of the importance resampling method, the RMSE decreases as the particles population increases. However, the auxiliary resampling method performs substantially better when compared to other importance resampling methods in the simulation study. The best strategy in this experiment is using PF with 1024 particle from the NU density with AUX and bias correction in its prediction state. The RMSEs are 0.49° , 0.44° , and 0.52° on X , Y , and Z axis, respectively. It achieves higher accuracy than EKF and CF benchmarking algorithms, which have RMSEs of 0.93° , 0.59° , 0.72° and 1.34° , 0.71° , and 0.72° on each axis, respectively. The algorithms were implemented with MATLAB (Mathworks, MA, USA). The average processing times of one recursion are shown in Fig. 10. The processing time increases with the size of the particles' population. PF using vMF density requires significantly less processing time comparing to those using NU. Bias correction reduces the processing time for the algorithm using NU posterior density. This is due to NU's rejection sampling method and bias correction limits the drift error, and reduces the search space for new particles. Hence, it reduces sampling time. PF using AUX has the fastest processing time compare to the other two sets.

The simulation provided invaluable insight to the capability of SMC based attitude estimation in handling extremely noisy data inputs. In the second experiment, the algorithm is tested with the modular IMU system. The IMU is attached to a

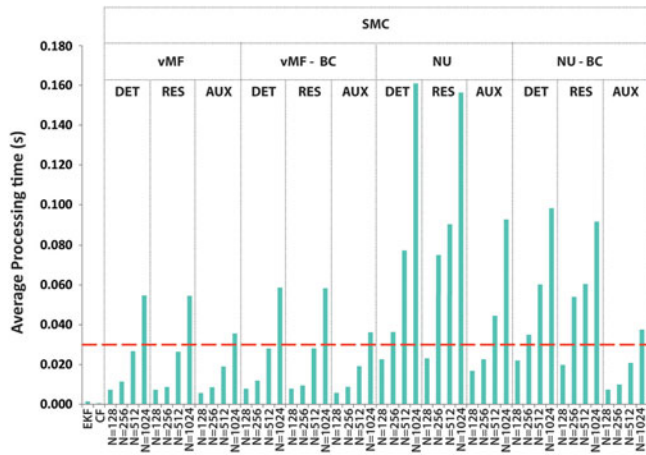


Fig. 10. Average processing time for each algorithm. The red dashed line indicates the time interval to achieve 30 fps.

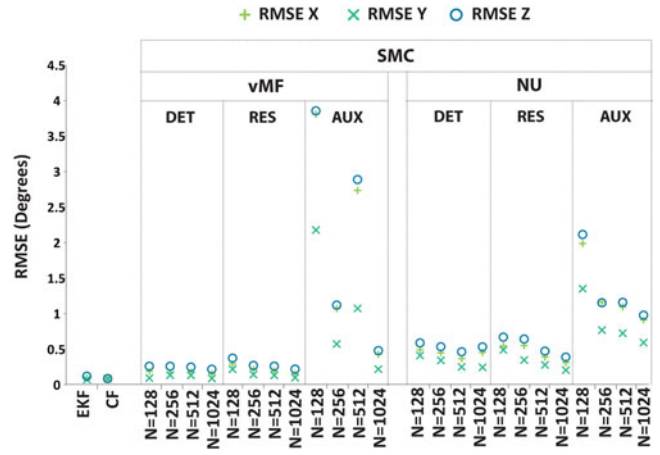


Fig. 12. RMSE of six different stationary poses processed by EKF, CF, and PF of various settings without bias correction.

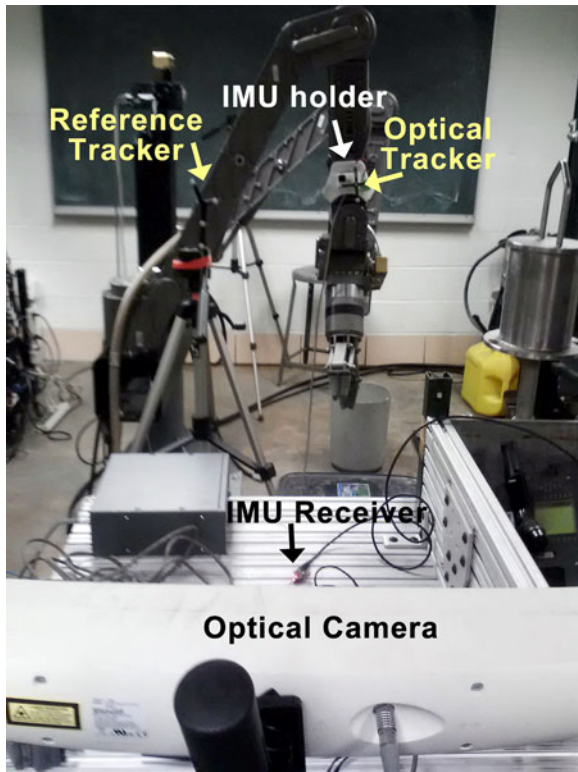


Fig. 11. Experimental setup with robotic application.

hydraulic robotic manipulator (TITAN II, Schilling Robotics) to monitor its motion (see Fig. 11). An optical tracker was attached to the IMU and was monitored by the optical camera (Polaris Spectra, Northern Digital Inc.). The data were collected in various stationary poses and dynamic motion. They were processed and compared against the optical tracking system.

The stationary experiment comprises of six different poses. Static pose is defined as the manipulator remaining in the same position for at least 300 samples. Four thousand samples were collected at the initial pose, which serves as the reference frame for subsequent poses. Each testing algorithm processes the reference dataset to create its own reference frame. Two thousand

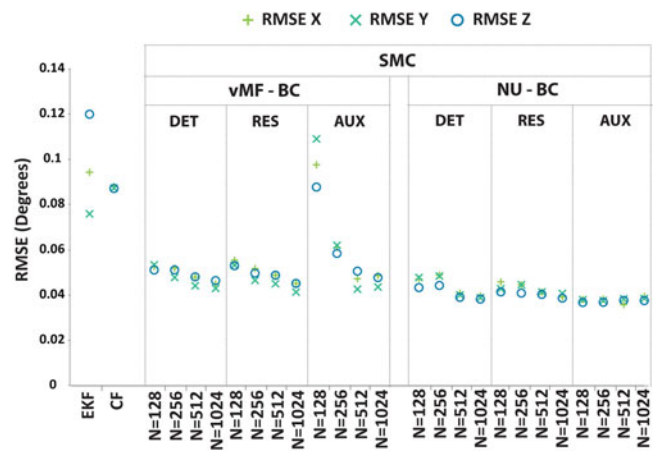


Fig. 13. RMSE of six different stationary poses processed by EKF, CF, and PF of various settings with bias correction.

samples were collected on each pose. The results from the PF are an average over 100 runs. The data are projected as Euler angles. The RMSEs are averaged among all the poses, and the results are shown in Figs. 12 and 13.

The RMSE decreases with the increasing particles size in all PF. There are significant differences on the performance of the importance resampling methods than the simulation study. Among the three methods examined, AUX has the highest error among the PF using both vMF and NU posterior densities without bias correction. However, extensive error reduction has been observed if bias correction is included in the prediction stage. PF using NU posterior density and bias correction gives the smallest RMSE.

For dynamic study, the robotic manipulator maneuvered the segment where the IMU tracker is attached to. The reference orientations of the optical and IMU system established in the static experiment are used to determine the relative transformation during the analysis. There are several instances during the experiment that the robotic manipulator remains stationary for an extended period of time. In order to minimize the influence of static data while assessing dynamic accuracy of the system, the

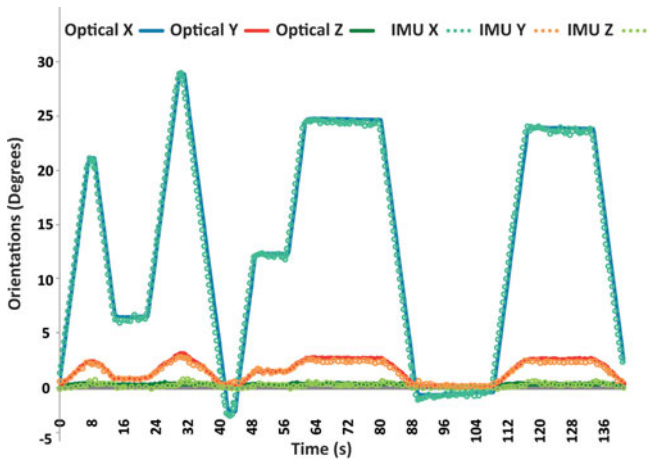


Fig. 14. IMU data processed by PF with bias correction (256 particles, NU posterior density) and compare to output to the optical tracker.

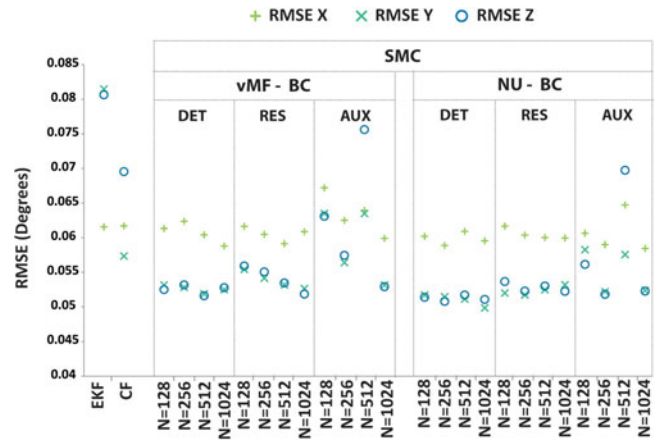


Fig. 16. RMSE in dynamic robotic application processed by EKF, CF, and PF of various settings with bias correction.

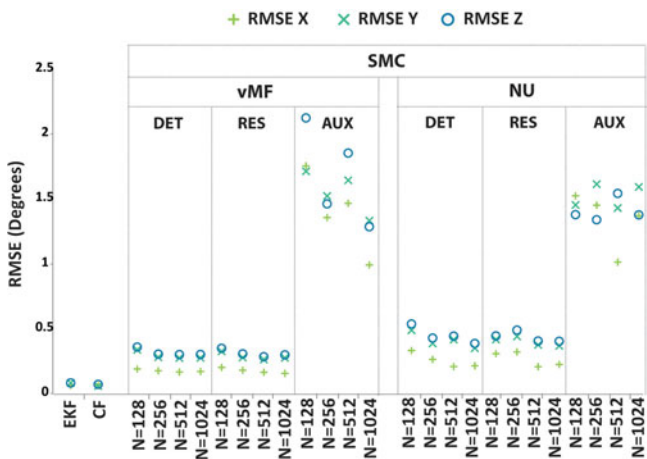


Fig. 15. RMSE in dynamic robotic application processed by EKF, CF, and PF of various settings without bias correction.

stationary data were removed manually from the processed data. The results from the PF algorithms are averaged over 100 runs. The data are projected as Euler angles. Fig. 14 shows the output of the optical tracker and the result of the IMU data processed by PF with 256 particles based on NU density with bias correction. The RMSE of each algorithm and settings are shown in Fig. 15 and 16.

The RMSE increases in all algorithms in the dynamic experiment compared to the static experiment. The data processed with PF using AUX without bias correction results in the highest RMSE. In general, PFs with bias correction yield smaller error than EKF and CF (see Fig. 16). The algorithm that gives the best result is PF with DET and bias correction.

The validation study for human motion tracking with IMU was determined with free-hand motion tracking. A container was made to fit the IMU and the optical tracking target as shown in Fig. 17(a). The container provides additional mounting spots for additional tracker to overcome the viewing limitation of the optical tracking system.

A total of 35 free-hand motion activities were performed in this experiment. Each activity began with establishing the refer-

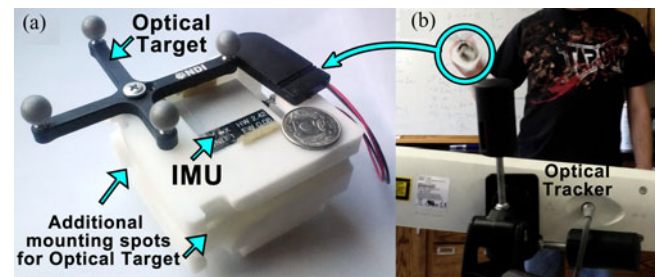


Fig. 17. (a) Container for the optical tracking target and the IMU system. (b) Free-hand human motion test of the IMU system.

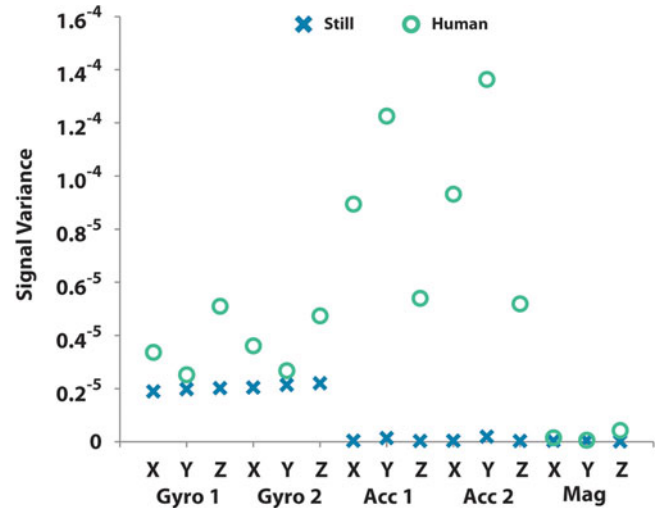


Fig. 18. Sensors' signal variance between undisturbed static pose and held by tester during initialization.

ence coordinate for both the IMU and optical systems with the tester holding the container stationary for a short period of time. The signal variances of the sensors during this procedure were compared to the static still pose from previous experiments, as shown in Fig. 18. The data revealed elevated variance of accelerometers and gyroscopes sensors as a result of the influence of the biofeedback mechanism from the musculoskeletal system during the free-hand stationary poses. As a result, the RMSE

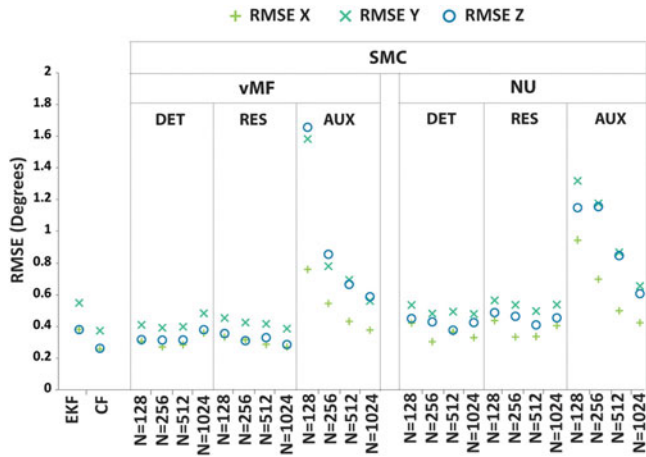


Fig. 19. RMSE of stationary poses processed by EKF, CF, and PF of various settings without bias correction.

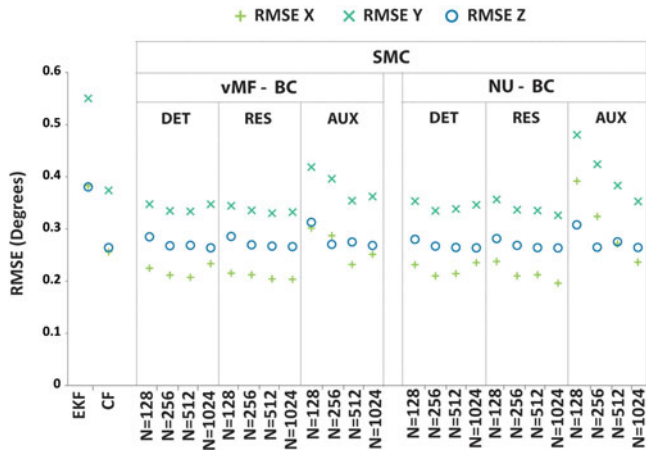


Fig. 20. RMSE of stationary poses processed by EKF, CF, and PF of various settings with bias correction.

during the stationary poses (see Figs. 19 and 20) have increased significantly in human motion tracking relative to the robotic experiment. Instability has been observed in several instances with EKF and CF algorithms, which will be discussed in the later sections. All of these instances have been omitted from error analysis.

The dynamic testing initiated after the references were established. The duration of the activities ranged from 1 to 20 min. The tester was free to maneuver the container in front of the optical tracker [see Fig. 17(b)]. The data were processed and analyzed by the testing algorithms. Instability was observed from activities processed by EKF and CF. There were two activities where instability occurred with EKF. In one instance, the filter was able to restabilize during a brief stall on the movement from the tester. In other case, the filter did not converge during the stationary period and remained unstable throughout the duration of the activity, as shown in Fig. 21(a). Fig. 21(b) shows the output from the PF processing the same activity. There were five instances where instability was observed with CF. In these cases, the filter failed to converge during the stationary period. The output from one of the cases is shown in Fig. 22(a) and the output from the PF processing the same activity is shown in

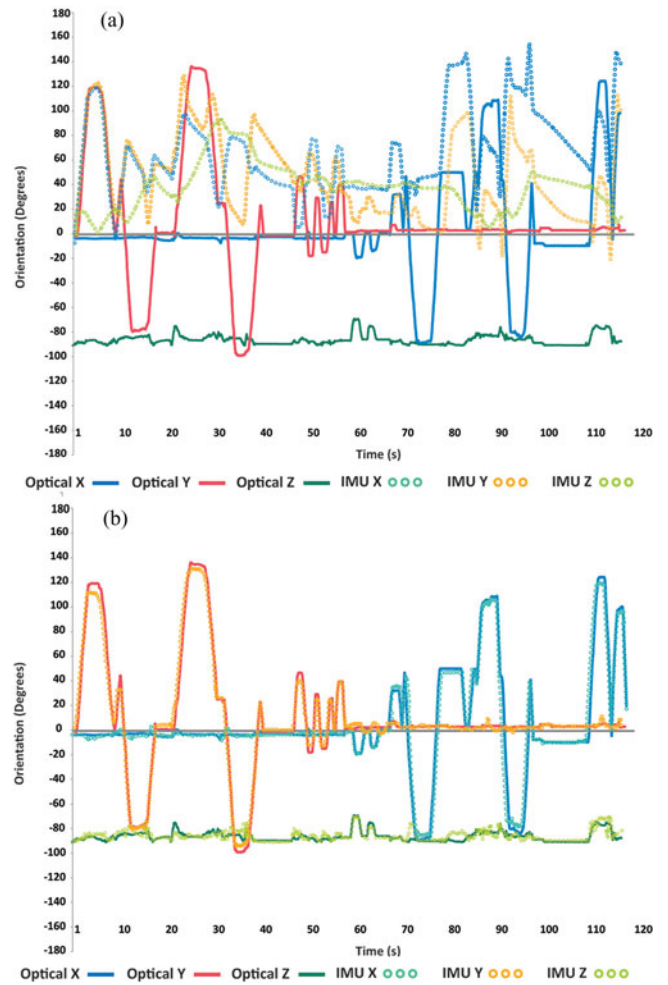


Fig. 21. Output of the same activity processed by: (a) EKF and (b) PF ($N = 256$, NU, RES).

Fig. 22(b). No divergence case was observed with the PF in any of the tested settings.

The instability issues can be contributed to the fundamental assumptions of the algorithms, which assume certain characteristics of the system, either linear/nonlinear Gaussian or the raw data properties such as variance of the signal. The advantage of PF over the benchmarking algorithm is its capability to deal with any system without compromising the integrity with assumptions. It allows detection and fast recovery if the solution begins to diverge.

Similar to the stationary poses analysis, the error analysis ignores all the cases where divergence was identified. The averages of the RMES from all the eligible testing activities are shown in Figs. 23 and 24.

For free-hand motion tracking, the RMSE have substantially increased by approximately tenfold compared to previous experiments with robotic manipulator. However, it was similar to the result from the simulation study. The performance of PF with bias correction yields better results in general. The performances of the PF without additional bias correction are less accurate than both EKF and CF. The effect of particle size is relatively subtle with DET and RES. AUX performs poorly without bias correction, yet it has the smallest RMSE among all

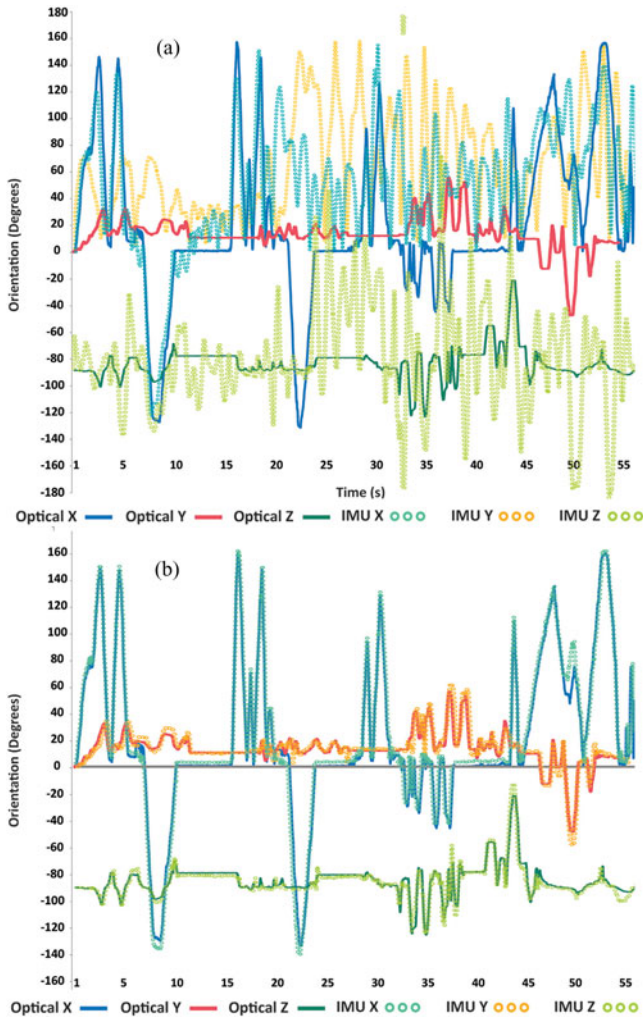


Fig. 22. Output of the same activity processed by: (A) CF and (B) PF ($N = 256$, NU, RES).

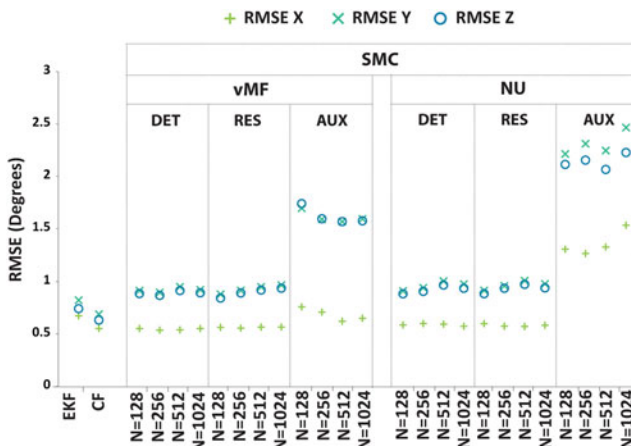


Fig. 23. RMSE of free-hand human motion tracking processed by EKF, CF, and PF of various settings without bias correction.

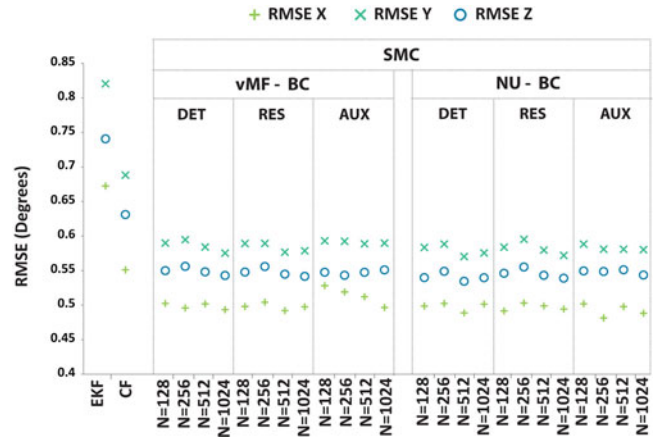


Fig. 24. RMSE of free-hand human motion tracking processed by EKF, CF, and PF of various settings with bias correction.

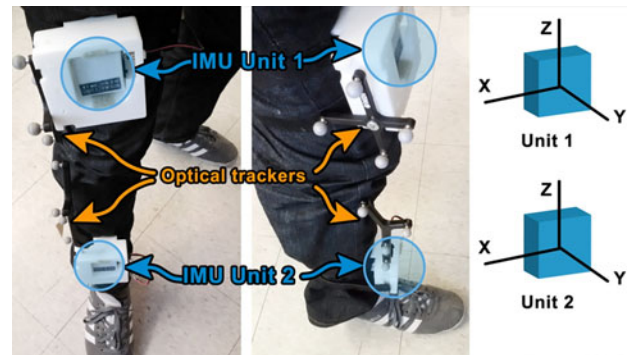


Fig. 25. Experimental setup for dynamic knee joint tracking.

tested algorithms if bias correction was used with NU density. The processing time for each algorithm is similar to the simulation study presented earlier. PFs using vMF density with more than 512 particles and NU density with more than 128 particles cannot be processed within the real-time threshold limit [i.e., 0.03 s or 30 frames per second (fps)]. One exception is the PFs using NU, BC, and AUX, which can be processed within the threshold up to 512 particles. The PF using AUX and NU density with 256 particles is best suited for real-time human motion tracking based on the processing time (0.0181 s or 55.1 fps) and the RMSE ($X: 0.45^\circ$, $Y: 0.53^\circ$, $Z: 0.51^\circ$).

The experiments from previous study are repeated with the current system to demonstrate its performance. The IMU system performed poorly under deep knee-bend and chair-rise activities in previous study [30]. These activities were performed and monitored with the modular inertial and optical tracking systems (see Fig. 25). The result is shown in Figs. 26 and 27. The PF with AUX without BC remains to be the least accurate among all the algorithms, while the PF using AUX and BC yields the best result.

The result of the current system is compared with the off-the-shelf system in the previous study. Table I shows the RMSE between the two systems. Both datasets were processed by the same EKF algorithm. The comparison demonstrates that the

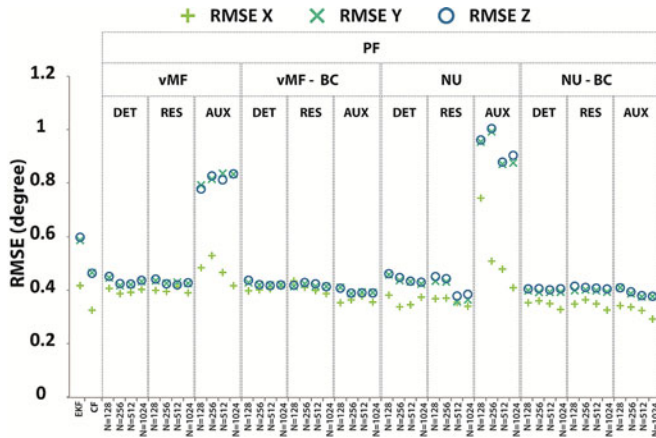


Fig. 26. Comparison between the RMSE of PF and the benchmark algorithms for IMU located at thigh during knee joint activities.

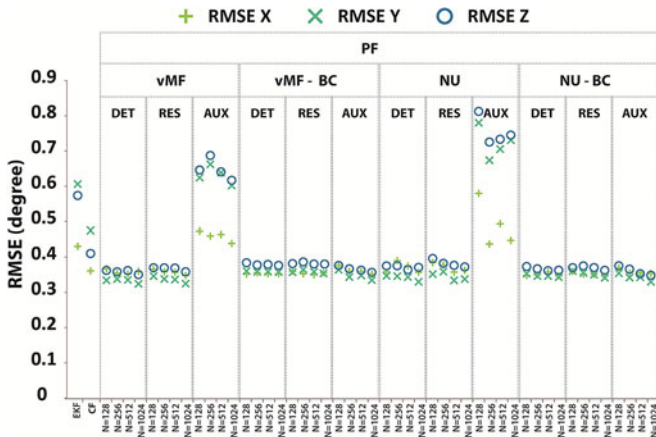


Fig. 27. Comparison between the RMSE of PF and the benchmark algorithms for shank during knee joint activities.

TABLE I
RMSE COMPARISON BETWEEN CURRENT AND PREVIOUS INERTIAL TRACKING SYSTEM ON DYNAMIC KNEE ACTIVITIES (EKF)

	X Axis	Y Axis	Z Axis
Thigh (Previous)	3.86°	3.84°	4.09°
Thigh (Current)	0.416°	0.588°	0.598°
Shank (Previous)	3.41°	4.51°	3.06°
Shank (Current)	0.431°	0.606°	0.573°

improvement on the hardware design had a positive impact on the accuracy of the system.

V. DISCUSSION

The current system shows highly accurate orientation tracking for robotic application, with RMSE ranging from 0.050° to 0.058° for dynamic activity in its optimal settings. However, tracking applications with surgical robotics must be designed carefully. The primary concern is the radius of operation of the robotic manipulator. The positioning accuracy of the robotic manipulator decreases as the length of unit increase. The current hardware offers the option to relocate only the sensor strips to the region of interest via cable connection to the sensor ports as a solution. For human motion tracking, the accuracy is generally

acceptable as a biomechanics assessment or clinical diagnostic tool.

The presented PF addresses the stability issue with inertial tracking of human motion. There are several cases of instability identified for both EKF and CF in free-hand motion tracking. The occurrence of CF's instability in free-hand activities is also statistically significant (Fisher's exact test, $p < 0.05$).

The primary drawback of this algorithm design is the exhaustive memory and computational cost, which are significantly higher than EKF and CF. Estimation algorithms from the Kalman or complementary filter families can easily be designed for microcontroller. On the other hand, the PF requires considerable amount of memory that scales with the amount of particles. The most memory exhaustive step in the presented algorithm is the weighted averaging of the quaternions. It requires a minimum of $2^{N+1} \times$ resolution allocated space, which most microcontrollers do not have. While it may still be possible to realize the PF algorithm with smaller particle size on some of the high-end microcontrollers with clever implementation, a programmable logic device is more suitable for this design. The current effort is to realize the PF algorithm onto an field programmable gate array (FPGA) module. This can substantially reduce the transmission bandwidth required for each IMU and subsequently allow more IMUs to enter the access point at a higher transmission rate.

The current algorithm has not utilized the full potential of the modular IMU design. There are multiple accelerometers and gyroscopes on the current unit, which can operate as multiple IMUs. The current processing method does not take advantage of this feature. Future expansion on the processing algorithm will utilize this feature to perform optimization on the inputs.

Other sampling methods and importance resampling techniques are currently being explored to provide a more efficient algorithm.

Future iterations will also expand the hardware inventory of available sensor strips to perform testing on more rigorous athletic activities such as running or jumping to assess the performance of the IMU in those conditions.

VI. CONCLUSION

This research presented implementation of a custom-built modular inertial tracking system. A novel integration and implementation of an SMC estimation algorithm using two hyperdimensional statistical geometries simulation techniques to sample and evaluates random quaternion particles. The dual particle states evaluation feature on the dispersion diversity and direction in the presented algorithm prevents extreme divergence from the estimating particles while maintaining necessary diversity for statistical estimation. It has also been shown that bias correction can significantly improve the accuracy of the PF. The capability of the algorithm has been demonstrated with multiple settings and resampling schemes in simulations, robotic, and human motion testing.

APPENDIX

This appendix demonstrates the implementation of various key components of the quaternionic PF algorithm. Random

quaternion simulations from vMF and NU distributions are shown in Algorithms 1 and 2, respectively. Algorithm 3 shows the estimation technique using Gauss Newton method. The weighted averaging for quaternion is shown in Algorithm 4.

Algorithm 1 – Random sampling from vMF density [32]

Input: μ (Mean vector),
 κ (Dispersion factor),
 N (Number of samples/particles)

1. $b = -\kappa + \sqrt{\kappa^2 + 1}$
2. $x_0 = \frac{1-b}{1+b}$
3. $c = \kappa(x_0) + 2\log(1 - x_0x_0)$
4. for $n = 1 \rightarrow N$
5. while $t \leq u$
6. while $s \leq 1$
7. $uu \sim \prod(-1,1)$, $vv \sim \prod(0,1)$
8. $s = uu + vv$
9. end
10. $z = \frac{1}{2} + uu * vv * \frac{\sqrt{1-s}}{s}$
11. $u \sim \prod(0,1)$
12. $w = \frac{1-z(1+b)}{1-z(1-b)}$
13. $t = \kappa(w) + 2\log(1 - x_0w) - c$
14. end
15. $\theta \sim \prod(0,2\pi)$, $u \sim \prod(-1,1)$
16. $v = \sqrt{1 - uu}$
17. $rand3DVec = [v * \cos(\theta) \quad v * \sin(\theta) \quad u]$
18. $q_r = w$
19. $q_{x,y,z} = \sqrt{1 - w^2} * rand3DVec$
20. $q = [q_r \quad q_x \quad q_y \quad q_z]$
21. $q_{vMF}(n) = q \otimes \mu$
22. End
23. Return q_{vMF}

Algorithm 2 – Random sampling from non-uniform density

Input: μ (Mean vector),
 K (Dispersion matrix),
 U (Orthogonal matrix of the distribution's symmetry axes)
 f_{max}, f_{min} (Maximum and minimum accepting densities)
 N (number of samples/particles)

1. $x_i = \infty, i = 1,2,3,4$
2. while $n < N$
3. if $x_1^2 + x_2^2 \geq 1$ or $x_3^2 + x_4^2 \geq 1$
4. $x_i \sim \prod(-1,1), i = 1,2,3,4$
5. else
6. $t = \frac{1-x_1^2+x_2^2}{x_3^2+x_4^2}$
7. $q_1 = x_4 \times t$
8. $q_2 = x_1$
9. $q_3 = x_2$
10. $q_4 = x_3 \times t$
11. end (go to 12.)
12. -- OR --
13. Use algorithm 1
14. $f = {}_1F_1\left(\frac{1}{2}, \frac{p}{2}, q\right)^{-1} e^{q^T U K U q}$
15. if $f > f_{min}$ & $f < f_{max}$
16. $q_{NU}(n) = q \otimes \mu$
17. $n = n + 1$
18. end
19. end
20. Return q_{NU}

Algorithm 3 – Observation quaternion (Gauss Newton method)

Input: $A_i, i = x, y, z$ (Accelerometers data),
 $M_i, i = x, y, z$ (Magnetometers data),
 N (Number of steps),
 q_{obv} (Observation quaternion from previous state)

1. for $n = 1 \rightarrow N$
2. $h_{1,2,3,4} = q_{obv} \otimes \{[0 \ M_x \ M_y \ M_z] \otimes conj(q_{obv})\}$
3. $b_{r,x,y,z} = \begin{bmatrix} 0 & \sqrt{h_2^2 + h_3^2} & 0 & h_4 \end{bmatrix}$
4. Compute Jacobian matrix
5. Compute rotational matrix R of q_{obv}
6. Perform Gauss Newton step
7. $qn_{x,y,z,r} = q_{obv,x,y,z,r} - \left(\frac{1}{J^T}\right) J^T (y_e - M(y_b))$ where
8. $y_e = \begin{bmatrix} 0 & 0 & 1 \\ b_x & b_y & b_z \end{bmatrix}, y_b = \begin{bmatrix} A_x & A_y & A_z \\ M_x & M_y & M_z \end{bmatrix}, M = \begin{bmatrix} R & \\ & R \end{bmatrix}$
9. Normalize $qn_{x,y,z,r}$
10. $q_{obv} = qn_{r,x,y,z}$
11. end
12. Return q_{obv}

Algorithm 4 – Expectation from weighted quaternion estimates

Input: $q_{est,i}, i = 1 \rightarrow N$ (Estimation data),
 $w_i, i = x, y, z$ (Data weights),
 N (Number of particles)

1. for $x = 1 \rightarrow \log(N)/\log(2)$
2. for $k = 1 \rightarrow \text{size}(q_{est,i})/2$
3. $w_n = w_{2k-1}/(w_{2k-1} + w_{2k})$
4. $\theta = \text{acos}(q_{est,2k-1} \cdot q_{est,2k})$
5. $q_{V,k} = q_{est,2k-1} \left(\frac{\sin((1-w_n)\theta)}{\sin\theta}\right) + q_{est,2k} \left(\frac{\sin((w_n)\theta)}{\sin\theta}\right)$
6. $w_{V,k} = w_{2k-1}(w_n) + w_{2k}(1 - w_n)$
7. end
8. $q_{V,k} \rightarrow q_{est,i}, i = 1 \rightarrow k$
9. $w_{V,k} \rightarrow w_i, i = 1 \rightarrow k$
10. end
11. Return q_V

REFERENCES

- [1] P. H. Veltink and H. B. K. Boom, "3D movement analysis using accelerometry theoretical concepts," in *Neuroprosthetics From Basic Research to Clinical Applications*, A. Pedotti, M. Ferrarin, J. Quintern, and R. Riener, Eds. Berlin, Germany: Springer-Verlag, 1996, pp. 317–326.
- [2] C. V. C. Bouten, K. T. M. Koekoek, M. Verduin, R. Kodde, and J. D. Janssen, "A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity," *IEEE Trans. Biomed. Eng.*, vol. 44, no. 3, pp. 136–147, Mar. 1997.
- [3] F. Forester, M. Smejia, and J. U. Fahrenberg, "Detection of posture and motion by accelerometry: A validation study in ambulatory monitoring," *Comput. Human Behav.*, vol. 15, no. 5, pp. 571–583, Sep. 1999.
- [4] M. J. Mathe, A. C. Coster, N. H. Lovell, and B. G. Celler, "Detection of daily physical activities using triaxial accelerometer," *Med. Biol. Eng. Comput.*, vol. 41, pp. 296–301, 2003.
- [5] G. Uswatte, "Objective measurement of functional upper extremity movement using accelerometer recordings transformed with a threshold filter," *Stroke*, vol. 31, pp. 662–667, 2000.
- [6] R. E. Mayagoitia, A. V. Nene, and P. H. Veltink, "Accelerometer and rate gyroscope measurement of kinematics: An inexpensive alternative to optical motion analysis systems," *J. Biomech.*, vol. 35, pp. 537–542, 2002.
- [7] R. Moe-Nilssen, "A new method for evaluating motor control in gait under real-life environmental conditions. Part 1: The instrument," *Clin. Biomech.*, vol. 13, pp. 328–335, 1998.
- [8] I. P. Pappas, M. R. Popovic, T. Keller, V. Dietz, and M. Morari, "A reliable gait phase detection system," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 9, no. 11, pp. 113–125, Oct. 2009.

- [9] K. O'Donovan, R. Yamnik, D. O'Keefe, and G. Lyons, "An inertial and magnetic sensor based techniques for joint angle measurement," *J. Biomech.*, vol. 41, no. 5, pp. 1029–1035.
- [10] R. Takeda, S. Tadano, M. Todoh, M. Morikawa, M. Nakayasu, and S. Yoshinari, "Gait analysis using gravitational acceleration measured by wearable sensors," *J. Biomech.*, vol. 42, no. 3, pp. 223–233, 2009.
- [11] D. Roetenberg, "Inertial and magnetic sensing of human motion," Ph.D. dissertation, Univ. Twente, Enschede, The Netherlands, 2006.
- [12] N. Al-Nadawi, S. Tedmori, E. Edirisinghe, and H. Bez, "An automated real-time people tracking system based on KLT features detection," *Int. Arab J. Inf. Technol.*, vol. 9, no. 1, pp. 100–107, 2012.
- [13] P. Jonker, S. Persa, J. Caarls, F. de Jong, and I. Lagendijk, "Philosophies and technologies for ambient aware devices in wearable computing grid," *Comput. Commun.*, vol. 26, no. 11, pp. 1145–1158, 2003.
- [14] E. Azimi, "Augmented reality goggles with an integrated tracking system for navigation in neurosurgery," in *Proc. IEEE VRW*, Mar. 2012, pp. 123–124.
- [15] C. D. W. Ward, "The WhaSP: A wireless hands-free surgical pointer for minimally invasive surgery," *IEEE Trans. Mechatronics*, vol. 17, no. 3, pp. 434–442, Mar. 2012.
- [16] H. L. Ren, "Multisensor data fusion in an integrated tracking system for endoscopic surgery," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 1, pp. 106–111, Jan. 2012.
- [17] A. Sororush, F. Farahmand, and H. Salarieh, "Design and implementation of an improved real-time tracking system for navigation surgery by fusion of optical and inertial tracking methods," *Appl. Mech. Mater.*, vol. 186, pp. 273–279, 2012.
- [18] E. G. Lefferts, F. L. Markley, and M. D. Shuster, "Kalman filtering for spacecraft attitude estimation," *J. Guid. Control Dyn.*, vol. 5, no. 5, pp. 417–429, Sep./Oct. 1982.
- [19] I. Y. Bar-Itzhack and Y. Oshman, "Attitude determination from vector observations: Quaternion estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-21, no. 1, pp. 128–136, Jan. 1985.
- [20] N. J. Kasdin, "Satellite quaternion estimation from vector measurements with the two-step optimal estimator," in *Proc. AAS Guid. Control Conf.*, Feb. 2002, pp. 02–002, AAS.
- [21] M. L. Psiaki, "Attitude determination filtering via extended quaternion estimation," *J. Guid. Control Dyn.*, vol. 23, no. 2, pp. 210–214, Mar./Apr. 2000.
- [22] M. D. Shuster, "Kalman filtering of spacecraft attitude and quest model," *J. Astronaut. Sci.*, vol. 38, no. 3, pp. 377–393, Sep. 1990.
- [23] I. Y. Bar-Itzhack, "Request: A recursive quest algorithm for sequential attitude determination," *J. Guid. Control Dyn.*, vol. 19, no. 5, pp. 1034–1038, Sep./Oct. 1996.
- [24] D. Choukroun, I. Y. Bar-Itzhack, and Y. Oshman, "Optimal request algorithm for attitude determination," *J. Guid. Control Dyn.*, vol. 27, no. 3, pp. 418–425, May/Jun. 2004.
- [25] J. L. Crassidis and F. L. Markley, "Unscented filtering for spacecraft attitude estimation," *J. Guid. Control Dyn.*, vol. 26, no. 4, pp. 536–542, 2003.
- [26] Y. Cheng and J. L. Crassidis, "Particle filtering for sequential spacecraft attitude estimation," in *Proc. AIAA Guid. Navigat. Control Conf.*, Rhode Island, VA, USA, Aug. 2004, pp. 2004–5337, AIAA.
- [27] A. Carmi, "Sequential Monte Carlo methods for spacecraft attitude and angular rate estimation from vector observations," M.S. thesis, Israel Inst. Technol., Haifa, Israel, 2008.
- [28] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artif. Intell.*, 1981, pp. 674–679.
- [29] P. Bagherpour, S. A. Cheraghi, and M. b. M. Mokji, "Upper body tracking using KLT and Kalman filter," in *Proc. INNS-WC*, 2012, vol. 12, pp. 185–191.
- [30] G. To and M. Mahfouz, "Design of wireless inertial trackers for human joint motion analysis," presented at IEEE Conf. BioWireless Tech., Santa Clara, CA, USA, 2012.
- [31] T. D. Downs, "Orientation statistics," *Biometrika*, vol. 59, pp. 665–676, 1972.
- [32] A. T. A. Wood, "Simulation of the von Mises distribution," *Commun. Stat. Simul. Comput.*, vol. 23, pp. 157–164, 1994.
- [33] K. Shoemake, "Animating rotation with quaternion curves," in *Proc. ACM SIGGRAPH*, 1985, pp. 245–254.
- [34] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Stat. Comput.*, vol. 10, pp. 197–208, 2000.
- [35] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *J. Comput. Graph. Stat.*, vol. 5, no. 1, pp. 1–25, 1996.
- [36] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamics systems," *J. Amer. Stat. Assoc.*, vol. 93, no. 443, pp. 1032–1044, 1998.
- [37] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *J. Amer. Stat. Assoc.*, vol. 94, no. 446, pp. 590–599, Jun. 1999.
- [38] R. Mahony, "Nonlinear complementary filters on the special orthogonal group," *IEEE Trans. Autom. Control*, vol. 53, no. 3, pp. 1203–1218, Jun. 2008.



Gary To (S'05–M'13) was born in Hong Kong, China, in 1982. He received the B.S., M.S., and Ph.D. degrees in biomedical engineering from the University of Tennessee, Knoxville, USA, in 2004, 2007, and 2012, respectively.

He is currently a Postdoctorate Research Associate at the Institute of Biomedical Engineering, University of Tennessee. His current research interests include bioinstrumentation and smart implant design.



Mohamed R. Mahfouz (S'98–M'01–SM'06) received the B.S.B.M.E. and M.S.B.M.E. degrees from Cairo University, Cairo, Egypt, in 1987 and 1992, respectively, the M.S.E.E. degree from the University of Denver, Denver, CO, USA, in 1997, and the Ph.D. degree from the Colorado School of Mines, Golden, USA, in 2002.

He was the Director of Research at Rocky Mountain Musculoskeletal Research Laboratory, Denver, in 1999. He was a Senior Research Scientist at Oak Ridge National Laboratory (ORNL) and the Co-

Director at the Center for Musculoskeletal Research, University of Tennessee (UT)/ORNL. He was the Program Director for BME at UT in 2007. He is currently the Chair of the Institute of Biomedical Engineering (IBME), University of Tennessee, Knoxville, USA.