Multimodal Movement Sensing using Motion Capture and Inertial Sensors for Mixed-Reality Rehabilitation

by

Yangzi Liu

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science in Technology

Approved November 2010 by the
Graduate Supervisory Committee:

Gang Qian, Chair
Loren Olson
Jennie Si

ARIZONA STATE UNIVERSITY

November 2010

ABSTRACT

This thesis presents a multi-modal motion tracking system for stroke patient rehabilitation. This system deploys two sensor modules: marker-based motion capture system and inertial measurement unit (IMU). The integrated system provides real-time measurement of the right arm and trunk movement, even in the presence of marker occlusion. The information from the two sensors is fused through quaternion-based recursive filters to promise robust detection of torso compensation (undesired body motion). Since this algorithm allows flexible sensor configurations, it presents a framework for fusing the IMU data and vision data that can adapt to various sensor selection scenarios. The proposed system consequently has the potential to improve both the robustness and flexibility of the sensing process.

Through comparison between the complementary filter, the extended Kalman filter (EKF), the unscented Kalman filter (UKF) and the particle filter (PF), the experimental part evaluated the performance of the quaternion-based complementary filter for 10 sensor combination scenarios. Experimental results demonstrate the favorable performance of the proposed system in case of occlusion. Such investigation also provides valuable information for filtering algorithm and strategy selection in specific sensor applications.

ACKNOWLEDGEMENTS

I sincerely appreciate Prof. Gang Qian for being my advisor during the past 2 years. His patient guidance and illuminating teaching laid a strong basis for my academic progress.

I would like to express my thanks to Prof. Jennie Si and Prof. Loren Olson for being my committee members and providing valuable advice.

I appreciate the hard work of my co-workers, Jeff Boyd and Michael Baran, who cooperated with me in this project.

I would send gratitude to Assegid Kidané for his selfless help in solving hardware and software issues.

TABLE OF CONTENTS

Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 Mixed-Reality Rehabilitation

Mixed-reality rehabilitation (MRR) [1, 2] has recently attracted much attention in movement training and therapy. An MRR system is a user-friendly, easily adjustable, interactive environment that is more responsive and more engaging than traditional rehabilitation. In MRR, a training patient receives real-time computer-generated visual and audio feedback responding to the quality of their movement during the training. An MRR system is fully adjustable according to the specific physical and psychological status of the training subject to help the subject perform therapeutic rehabilitation with enhanced self-confidence. Furthermore, in MRR training patients implicitly learn the training movement patterns and accumulate self-correct experience according to the system feedback, instead of relying on the explicit instruction from the therapists. Implicit learning has been shown to be more effective in promoting motor learning than explicit instructions. These advantages of MRR contribute to its increased training efficiency. In our research, we have developed a mixed-reality stroke rehabilitation system [3] to assist stroke patients in finishing tasks such as reaching and grasping by tracking the movement of the affected arm and the torso of the training subject. This system has been tested in a pilot study with three patient participants for six 75-minute sessions each subject over two weeks [3]. Results show that after this short period of time, the participating patients all showed significant motor function improvement such as faster reaching speed, better joint coordination and less torso/shoulder compensation [3].

1.2 Sensor Fusion Approach

Effective MRR requires real-time and accurate movement tracking. In addition, such movement tracking also needs to be noise-resilient in challenging tracking scenarios, e.g. during occlusions caused by the physical therapist walking around the patients while

taking care of the training session. These demands can be satisfied through the application of sensor fusion approaches.

Sensor fusion aims to combine sensory data to provide more accurate, complete or dependent measurement than that from individual sensors. When observations are corrupted by noise (e.g. occlusion) for some sensors, they might still be visible to others. Multi-sensor fusion is also important since it can mitigate the effects of errors in measurement. Popular sensor fusion algorithms include least mean square (LMS) methods, Kalman filters (KF), and particle filters (PF). Proper sensor selection and fusion strategy can enable different sensors to compensate for the limitation of each other. For example, orientation tracking can be realized by the fusion of vision data and inertial data.

Vision-based sensing is an intuitive solution to movement tracking for MRR [1, 2]. By tracking reflective markers or other identifiers attached to the subjects, visual sensing provides accurate measurements of the positions of human bony landmarks such as body joints [4], from which the movement of the training subject such as body orientation and joint angles can be derived. Furthermore, vision-based marker-less tracking systems [4, 5, 6, 7, 8] have been developed. However major obstacles such as depth ambiguities, kinematic singularities, high computational cost, and limited tracking accuracy still need to be overcome for marker-less motion capture to be useful in MRR [9]. For camera-based visual sensing, occlusion is a major source of tracking failure and error. To tackle this challenge, inertial sensors such as accelerometer, gyroscope, and magnetometer have been widely used in movement tracking. The Kalman filter and the complementary filter have been adopted to integrate sensor observations from magnetometer and accelerometer with the gyro data to obtain reliable movement tracking [10, 11, 13, 14]. Such techniques have also been used in MRR [12]. The current trend in inertial sensing is to integrate the gyro, accelerometer and magnetometer into a single inertial measure unit

(IMU). Existing off-the-shelf IMUs include the Xsens MT9 [15], the MicroStrain G-link [16], and the Sparkfun 6DOF V3/V4 IMU sensors [17]. These IMUs are low-cost, compact in size, and capable of wireless data transmission. Nevertheless, compared to marker-based visual sensing, existing IMUs suffer from inferior tracking accuracy due to noticeable measurement noise. A combination of the camera and the IMU will potentially compensate their drawbacks for each other while remain their advantages, leading to an occlusion-free and highly accurate orientation tracking solution

In view of the complementary movement tracking strengths of the camera and the inertial sensor, integrated solutions using both visual and inertial sensors have recently attracted much attention [18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]. The vision-based mark-less tracking techniques have been mainly used in these integrated frameworks. Such integrated tracking has also been used in MRR. For example, in [29] a multimodal arm movement tracking system for MRR is introduced using static camera and wearable inertial sensors. The static video camera detects skin color to capture arm movement, while the inertial sensor is attached to the wrist joint of the subject's arm to provide additional motion sensing measurement. This method assumes the shoulder joint of the training subject is fixed, and it does not track the torso orientation.

1.3 Challenges and Motivation

Although much effort has been devoted to developing integrated visual and inertial tracking systems, a number of pressing challenges are yet to be addressed for such systems to be useful in MRR. First of all, existing multimodal movement tracking methods are usually tailored to specific sensor configurations. Tracking failure can easily occur when the targeted sensor configuration is violated. Developing an integrated tracking solution that can cope with various sensor configurations according to specific sensing conditions is a challenge. Secondly, despite significant progress in multimodal

tracking, to our knowledge, there has been no systematic evaluation of tracking performance between various sensor configurations based on the same benchmark testing data. Such comparative performance study is necessary and extremely useful for sensor selection and system design. Furthermore, the reported tracking accuracy of the existing tracking systems is low (e.g. center meter accuracy in joint position tracking [29]) and can hardly satisfy the needs of precise feedback control in an immersive MRR environment. Securing accurate real-time tracking is also a challenge.

Confronting these challenges, it is highly desired to develop a multimodal sensing system for MMR with both satisfying precision and flexible sensor configuration through the fusion of the vision and the IMU data. Based on this system, the tracking performance of different sensor configurations can be compared.

1.4 System Overview

This thesis presents a novel integrated tracking framework for MRR using marker-based motion capture and inertial sensing (Figures 1 and 2). As illustrated in Figure 2, the motion capture cameras distributed around patients track the markers attached on the patients, while an IMU composed of gyro, accelerometer, and magnetometer is attached to the back-marker board and aligned with the back markers (Figure 3) to track the torso orientation. The complementary filter [14], the extended Kalman filter (EKF), the unscented Kalman filter (UKF) and the particle filter (PF) are deployed to integrate the multimodal movement data and provide quaternion orientation estimation. In the proposed approach, along with the magnetometer and accelerometer readings, vectors formed by markers also comprise an input to update torso orientation. In the presence of occlusion, the filtering algorithms could still provide satisfying measurement of torso orientation solely relying on the IMU data.

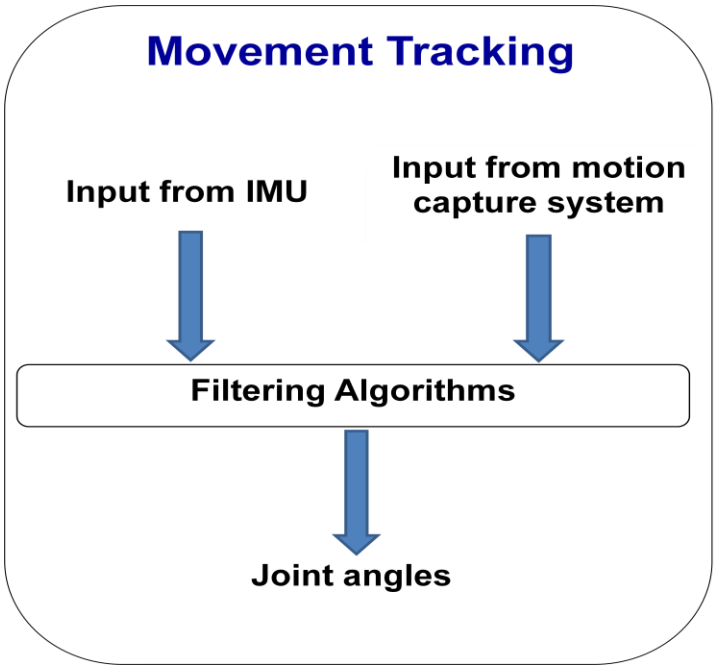

Fig. 1. The Proposed Integrated Movement Tracking Framework

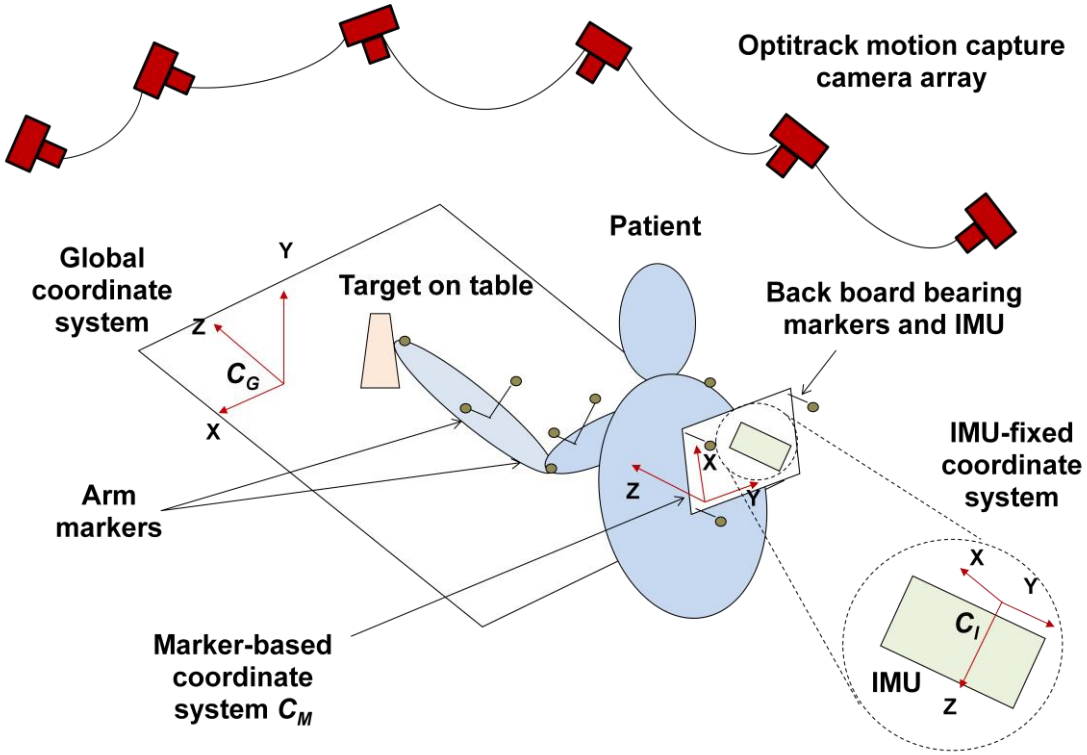

Fig. 2. System Setup and Coordinate Systems

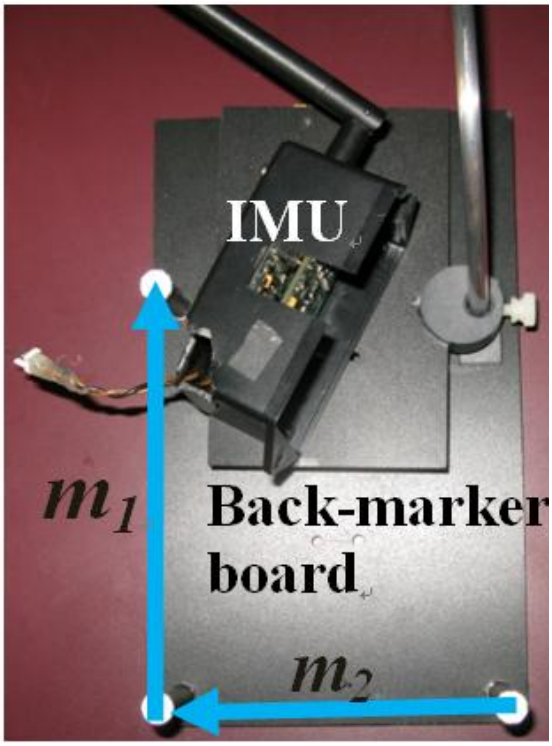

Fig. 3. IMU, Back-marker Board and Marker Reference Vectors

1.5 Contributions

The major contributions of this thesis are as follows. First of all, the proposed integrated tracking based on filtering approaches offers a general sensor fusion framework that can cope with various sensing scenarios according to the availability and reliability of different sensory data. Through dropping noise-corrupted sensory data and switching between different sensor configurations, the system is able to handle challenging sensing situations such as occlusion within therapeutic trials. In our experiments, a total of 10 typical sensor configurations cases have been investigated and evaluated. The results show that the proposed tracking system can work reasonably well in all 10 sensing scenarios, and that when the sensing scenarios become challenging, e.g., with less sensors, the performance of the proposed tracking approach degrades elegantly. Especially, an algorithm has been proposed to improving tracking stability when only one reference vector is available. Secondly, a comparative study between different sensing

scenarios has been carried out based on the tracking results of the proposed approach in 10 typical sensor scenarios using the same set of benchmark testing data. Also, the performance of four popular sensor fusion filters including the complementary filter, the extended Kalman filter (EKF), the unscented Kalman filter (UKF) and the particle filter (PF) are compared for the application of quaternion-based orientation tracking. Such study is valuable in sensor selection and system design for a particular MRR application. Finally, compared to existing work, the marker-based motion capture used in the proposed multimodal tracking system secures precise movement tracking for effective MRR.

1.6 Organization

The outline of this thesis is as follows. Chapter 2 introduces the calibration of the motion capture and inertial sensors. In Chapter 3, we introduce the temporal and spatial alignment between different sensors. The quaternion orientation representation is introduced in Chapter 4. Then the sensor fusion algorithms are described in Chapter 5. Chapter 6 clarifies certain implementation details. Experimental results and performance analysis are given in Chapter 7. Finally in Chapter 8, the research is concluded.

2. SENSOR CALIBRATION

System calibration is a critical component in information and sensor fusion. In our research on multimodal movement tracking, as illustrated by Figure 4, the system calibration includes the calibration of individual movement sensors and the temporal and spatial alignment across different sensors. In this section, we discuss the calibration of individual sensors, and the sensor alignment is discussed in the next section.

2.1 Motion Capture System Calibration

Most marker-based motion capture systems come with their custom-made calibration modules. Users can easily obtain reliable system calibration through a step-

by-step calibration procedure. Millimeter accuracy of the 3-D marker location can be achieved after a successful calibration.

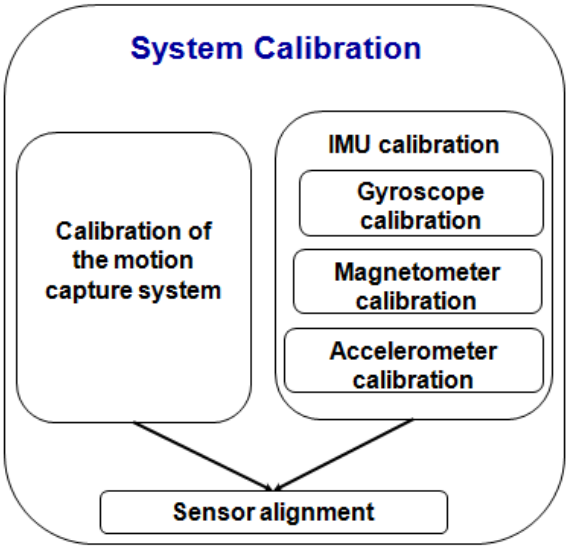

Fig. 4. System Calibration

2.2 IMU Calibration

In our research, the Sparkfun 6DOF V3 IMU has been used. This IMU contains a three-axis accelerometer, a three-axis gyroscope and a two-axis magnetometer. The raw readings from the IMU need be converted to engineering values according to equations obtained through corresponding calibration procedures. The sampling rate of the IMU was set to 201 Hz. Some cyclical noise is introduced in the data when sampling at 200Hz. The 1Hz increase in the sample rate eliminated this noise. The manufacturer's technical support forums verified this error. By collecting data from different sensor placement or known rotation rate, calibration equations can be determined through linear regressions, which provide both sensitivity and offset. For each sensor placement, the sensor is tested on a leveled surface and the edges of the sensor board are assumed to be in line with the printed reference axes on the sensor board as shown in Figure 5.

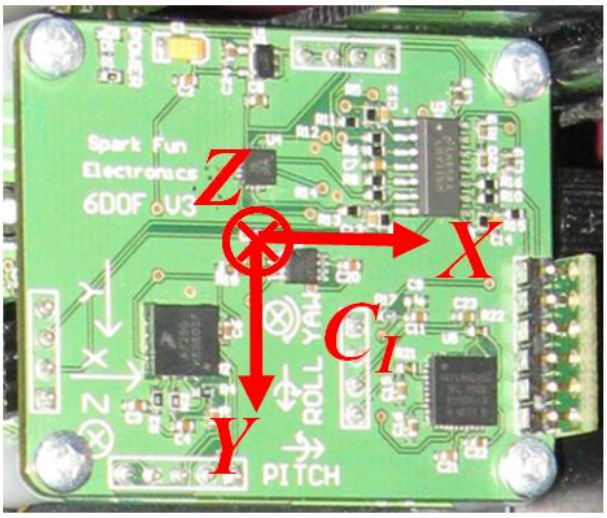

Fig. 5. Sparkfun 6DOF V3 IMU and the IMU-fixed Coordinate System C_I

Simultaneously, the inclination angle α of the local magnetic field is measured. In Tempe, AZ, where the testing was done, the National Geophysical Data Center reports that the magnetic field points downward at an inclination angle of 59.72°. Therefore, a significant vertical component of the magnetic field is expected in the testing. According to our calibration data, α = 63.34°, which in general agrees with the records. Slight difference exists due to environmental ferromagnetic materials. The inclination angle needs to be reexamined once the sensor is moved to a new space. In this case the magnetometer calibration also needs to be redone.

2.3 Cross Axis Effect

Cross axis effect, which refers to the asymmetry of the ADC values when the magnetometer is rotating around the vertical direction, is observed during calibration. The cross axis effect may occur to magnetometers with multiple sensing axes on a single chip. In case of the magnetometer on the Sparkfun IMU used for our research, the measurement for the X-axis and the Y-axis originate from the same chip. Therefore, there is a chance that each axis is sensitive to magnetic fields orthogonal to their principal sensing direction. Ideally, the ADC value along certain axis from the magnetometer

responds to the absolute angle between the axis and north. Thus, the reading of a 45° clockwise rotation from north should be the same as that of a 315° clockwise rotation. Somehow, influenced by the cross axis effect, the raw values for X and Y axis illustrated in Figure 6 are skewed and asymmetry about the vertical axis, which indicates that correction is necessary for accurate measurement of magnetic field to be made. To compensate for the cross axis effect, the method developed by Honeywell [30] has been implemented in our research. The corrected values are also shown in Figure 4. It can be seen that these corrected values show valid symmetry about the vertical axis, and these values are used for calibrating the magnetometer.

3.

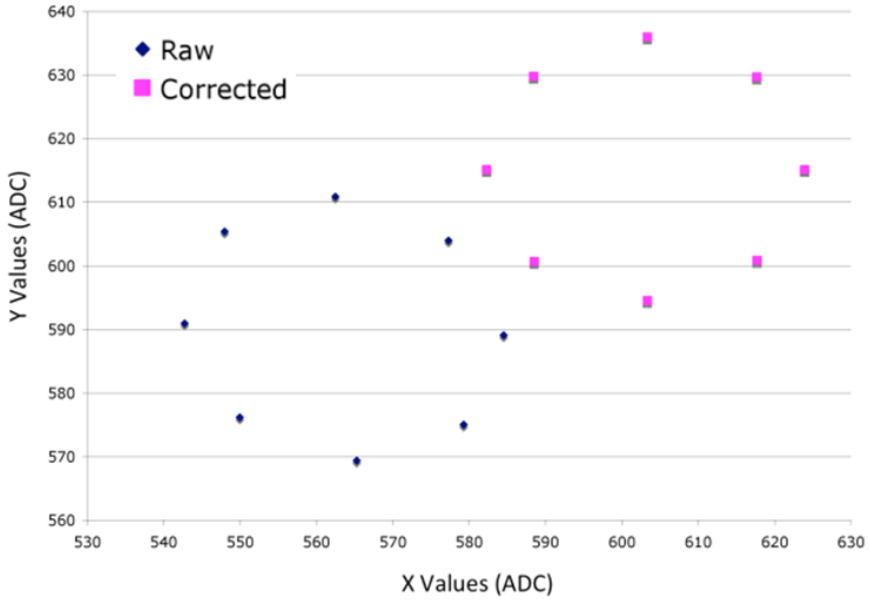

Fig. 6. Comparison Before and After Cross Axis Correction

SENSOR ALIGNMENT

In a sensor fusion approach, it is often required that the sensors are aligned temporally and spatially so there observations from different sensors can be effectively integrated.

3.1 Temporal Alignment

The temporal alignment ensures that measurements from different sensors are aligned in the same global time frame. In our proposed approach, the motion capture system and the IMU run at high sampling rates, 100Hz for the motion capture system and 201Hz for the IMU. In addition, the latencies of both systems are also very small. Therefore, the observations from the motion capture system and the IMU are simply aligned in time according to their arrival time at the processing computer. Specifically, whenever a new frame of motion capture data is available, the latest IMU observation is then taken and integrated with the motion caption data. Consequently, the integrated movement tracking system runs at an average frame rate of 100 Hz.

3.2 Spatial Alignment

To achieve spatial sensor alignment in our proposed approach, four different coordinate systems are used, including a fixed global coordinate system $\boldsymbol{C_G}$, and three local coordinate systems $\boldsymbol{C_I}$, $\boldsymbol{C_M}$ and $\boldsymbol{C_S}$, which are respectively attached to and defined on the IMU, the back-marker board, and torso of the subject, respectively. These coordinate systems are illustrated in Figure 2. Detailed definitions and setup of these coordinate systems are given in Table 1. The IMU is mounted on the back-marker board, as shown in Figures 2 and 3. The spatial relationships among these three local coordinate systems remain fixed throughout a training session.

In our proposed approach, it is necessary to first align $\boldsymbol{C_I}$ and $\boldsymbol{C_M}$ by finding the rotation matrix R_{IM} from $\boldsymbol{C_I}$ to $\boldsymbol{C_M}$ so that the inertial observations can be represented in $\boldsymbol{C_M}$. This alignment is done as part of the system calibration when the marker board and the IMU are static. To obtain R_{IM}, we first compute R_{IG}, the rotation matrix from $\boldsymbol{C_I}$ to $\boldsymbol{C_G}$, and then R_{GM}, the rotation matrix from $\boldsymbol{C_G}$ to $\boldsymbol{C_M}$, and finally R_{IM} is given by

$$R_{IM} = R_{GM} R_{IG} \tag{1}$$

Table 1. Coordinate System

Coordinate Systems	Description
Global coordinate system $\boldsymbol{C_G}$	Determined during the calibration of the motion capture system
Marker-based coordinate system $\boldsymbol{C_M}$	Determined by the three markers on the board attached to on the back of the subject
IMU coordinate system $\boldsymbol{C_I}$	The local coordinate system used by the IMU. The IMU readings are measured in $\boldsymbol{C_I}$
Segmental coordinate system $\boldsymbol{C_S}$	A virtual coordinate system attached to the back of the subject so that the Euler angles (pitch, yaw, roll angles) encoding the orientation of $\boldsymbol{C_S}$ in $\boldsymbol{C_G}$ actually correspond to the anatomical angles caused by torso joint actions such as leaning forward/backward /left/right, and twisting.

The rotation matrices R_{GM} and R_{IG} can be found based on the shared reference vectors between $\boldsymbol{C_G}$ and $\boldsymbol{C_M}$, and $\boldsymbol{C_I}$ and $\boldsymbol{C_G}$. In general, given a pair of shared reference column vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ in two coordinate systems $\boldsymbol{C_X}$ and $\boldsymbol{C_Y}$, the rotation matrix R_{XY} from $\boldsymbol{C_X}$ to $\boldsymbol{C_Y}$ can be found. Let $\boldsymbol{v}_{Xi}$ $(i = 1, 2)$ be the reference vectors measured in $\boldsymbol{C_X}$ and $\boldsymbol{v}_{Yi}$ $(i = 1, 2)$ in $\boldsymbol{C_Y}$. A third vector in both coordinate systems can be found as the cross-product of the measurement vectors, i.e.,

$$\begin{cases} \boldsymbol{v}_{X3} = \boldsymbol{v}_{X1} \times \boldsymbol{v}_{X2} \\ \boldsymbol{v}_{Y3} = \boldsymbol{v}_{Y1} \times \boldsymbol{v}_{Y2} \end{cases} \tag{2}$$

Let

$$\begin{cases} A_X = [\boldsymbol{v}_{X1}, \boldsymbol{v}_{X2}, \boldsymbol{v}_{X3}] \\ A_Y = [\boldsymbol{v}_{Y1}, \boldsymbol{v}_{Y2}, \boldsymbol{v}_{Y3}] \end{cases} \tag{3}$$

Then according to the definition of R_{XY}

$$A_Y = R_{XY} A_X \tag{4}$$

and R_{XY} can be easily found by solving (4).

Using this approach, both R_{IG} and R_{GM} can be obtained. To find R_{IG}, the normalized gravity vector $\boldsymbol{n}$ and local magnetic field $\boldsymbol{b}$ are taken as the two shared reference vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ between $\boldsymbol{C_G}$ and $\boldsymbol{C_I}$. The measurements of $\boldsymbol{n}$ and $\boldsymbol{b}$ in $\boldsymbol{C_I}$ are obtained from the readings of the accelerometer and the magnetometer. The measurements of these two

reference vectors in $\boldsymbol{C_G}$ are determined according to the way $\boldsymbol{C_G}$ is defined. In our approach, the Y-axis of $\boldsymbol{C_G}$ is to the opposite direction of the gravity vector. Thus $\boldsymbol{n}$ in $\boldsymbol{C_G}$ is

$$\boldsymbol{n_G} = [0, -1, 0] \tag{5}$$

The angle β between the Z-axis of $\boldsymbol{C_G}$ and the magnetic north can be found using a compass. Based on the knowledge of the inclination angle α described in Section 2.2, the local magnetic field $\boldsymbol{b}$ in $\boldsymbol{C_G}$ can be represented by a normalized constant vector

$$\boldsymbol{b_G} = [\cos(\alpha)\sin(\beta), -\sin(\alpha), \cos(\alpha)\cos(\beta)] \tag{6}$$

To find R_{GM}, as illustrated in Figure 3, two marker vectors $\boldsymbol{m_1}$ and $\boldsymbol{m_2}$formed by two pairs of markers are taken as the shared reference vectors between $\boldsymbol{C_G}$ and $\boldsymbol{C_M}$. The measurements for computing R_{IG} and R_{GM} are taken simultaneously when the marker board and IMU are static. Then R_{IM} can be derived from (1).

Furthermore, in our proposed approach, we would like to obtain the joint angle tracking results in terms of the kinesiology-meaningful anatomical joint actions. For this reason, we have introduced the segmental coordinate system $\boldsymbol{C_S}$ in which the joint action angles correspond to the Euler angles. In our approach, after obtaining the angle tracking results in terms of the orientation of $\boldsymbol{C_M}$ in $\boldsymbol{C_G}$, we need to further obtain the orientation of $\boldsymbol{C_S}$ in $\boldsymbol{C_G}$. Therefore, $\boldsymbol{C_S}$ and $\boldsymbol{C_M}$ need to be aligned. This alignment is carried out in the reference position, which refers to a specific initial attitude of the subject so that $\boldsymbol{C_S}$ is considered to be aligned with $\boldsymbol{C_G}$ in this reference position. Therefore, finding the rotation matrix R_{SM} is equivalent to finding the rotation matrix R_{GM} in the reference position. This can be easily done using the procedure presented above.

Nomenclature

$\boldsymbol{C_G}$	Global coordinate	$\boldsymbol{W}$	Weight matrix
$\boldsymbol{C_M}$	Marker-based coordinate	$\boldsymbol{\varepsilon(q)}$	Predicted error
$\boldsymbol{C_I}$	IMU coordinate system	X	Jacobi matrix of $\boldsymbol{y(q)}$
$\boldsymbol{C_S}$	Segmental coordinate	$\boldsymbol{x}$	State vector
$\boldsymbol{q}$	Quaternion orientation	P	Covariance matrix
θ	Rotation angle	$\boldsymbol{z}$	Sensor observation
$\boldsymbol{a}$	Rotation axis	$f(.)$	State transition function
$\boldsymbol{e}$	Rotation vector	$h(.)$	Measurement function
R_{IM}	Rotation matrix from $\boldsymbol{C_I}$ to $\boldsymbol{C_M}$	$\boldsymbol{w}$	Process noise
R_{SM}	Rotation matrix from $\boldsymbol{C_S}$ to $\boldsymbol{C_M}$	$\boldsymbol{v}$	Measurement noise
$\boldsymbol{n}$	Gravity vector	Q	Process noise covariance
$\boldsymbol{b}$	Magnetic field vector	R	Measurement noise covariance
$\boldsymbol{m}$	Marker vector	K	Kalman gain
$\boldsymbol{\omega}$	Angular velocity	$\boldsymbol{\Upsilon}$	Disturbance vector
$d(.)$	Prediction function	$\boldsymbol{\chi}$	Sigma Point
$\tilde{\boldsymbol{y}}$	Observed reference vector	$\boldsymbol{x^{(i)}}$	Particle state vector
$\boldsymbol{y(q)}$	Predicted reference vector	$\boldsymbol{W^{(i)}}$	Particle weights

4. QUATERNION BASICS

4.1 Quaternion Definition

Various rotation representation methods, such as Euler angles, rotation matrix, rotation vector, unit quaternion, etc., have been developed to express the orientation of an object (or coordinate frame) relative to a reference coordinate system. Among these methods, the quaternion provides a singularity-free description (as opposed to Euler angles) of the relative rotation and achieves compactness (as opposed to rotation matrices). There are three common ways to define a quaternion. A quaternion representation of orientation can be defined as a unit four-dimensional vector

$$\boldsymbol{q} = [q_0, q_1, q_2, q_3]^T \tag{7}$$

in which

$$q_0{}^2 + q_1{}^2 + q_2{}^2 + q_3{}^2 = 1 \tag{8}$$

Quaternion does not suffer from the singularity problem like Euler angles since they express rotation in terms of a single rotation about an inclined axis. A rotation of angle θ ($\theta \in [-\pi, \pi]$) about unit rotation axis $\boldsymbol{a}$ can be expressed as a normalized quaternion vector

$$\boldsymbol{q} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \boldsymbol{a} \cos\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{9}$$

According to equation (9), $\boldsymbol{q}$ and $-\boldsymbol{q}$ generate the same rotation.

A quaternion can also be expressed as the sum of four elements

$$\boldsymbol{q} = q_0 + q_1 i + q_2 j + q_3 k \tag{10}$$

where *i, j* and *k* are the hypercomplex numbers that satisfy

$$i^2 = -1, j^2 = -1, k^2 = -1 \tag{11}$$

$$ij = -ji = -1 \tag{12}$$

$$jk = -kj = -1 \tag{13}$$

$$ki = -ik = -1 \tag{14}$$

By the definition given by equation (10), a quaternion can be divided as the real part q_0 and the virtual part $[q_1, q_2, q_3]^T$. The virtual part is also called the vector part as it is corresponding to the rotation axis in the quaternion definition given by equation (9). Based on (10) – (14), a quaternion is defined on a nonlinear four dimensional unit sphere, instead of a linear vector space. Such definition leads to a distinct algebra of treating quaternions.

4.2 Quaternion Algebra

By analogy to normal complex numbers that also comprises of the real and virtual part, the conjugate of a quaternion $\boldsymbol{q}$, denoted as $\boldsymbol{q}^*$ is obtained by negating its virtual part as

$$\boldsymbol{q}^* = q_0 - q_1 i - q_2 j - q_3 k \tag{15}$$

When a quaternion $\boldsymbol{q}$ is multiplied by a real number c, every element of $\boldsymbol{q}$ is multiplied by c

$$\boldsymbol{q}c = c\boldsymbol{q} = cq_0 + cq_1 i + cq_2 j + cq_3 k \tag{16}$$

Based on the quaternion definition given by equation (10), the product of two quaternions $\boldsymbol{q_A}$ and $\boldsymbol{q_B}$ generates a new quaternion $\boldsymbol{q_C}$.

$$\begin{aligned}\boldsymbol{q_C} = \boldsymbol{q_A} \otimes \boldsymbol{q_B} &= (q_{A0} + q_{A1} i + q_{A2} j + q_{A3} k)(q_{B0} + q_{B1} i + q_{B2} j + q_{B3} k) \\ &= (q_{A0} q_{B0} - q_{A1} q_{B1} - q_{A2} q_{B2} - q_{A3} q_{B3}) \\ &+ (q_{A0} q_{B1} + q_{A1} q_{B0} + q_{A2} q_{B3} - q_{A3} q_{B2}) i \\ &+ (q_{A0} q_{B2} + q_{A2} q_{B0} + q_{A3} q_{B1} - q_{A1} q_{B3}) j \\ &+ (q_{A0} q_{B3} + q_{A3} q_{B0} + q_{A1} q_{B2} - q_{A2} q_{B1}) k\end{aligned} \tag{17}$$

where "$\otimes$" denotes quaternion multiplication. Assuming $\boldsymbol{q_A}$ represents the rotation from the coordinate system $\boldsymbol{C_1}$ to the coordinate system $\boldsymbol{C_2}$, and $\boldsymbol{q_B}$ represents the rotation from the coordinate system $\boldsymbol{C_2}$ to the coordinate system $\boldsymbol{C_3}$, then the quaternion

multiplication $\boldsymbol{q_A} \otimes \boldsymbol{q_B}$ implies the rotation from $\boldsymbol{C_1}$ to $\boldsymbol{C_3}$. From this point, the quaternion multiplication is not communicative, while quaternion multiplication still fulfills the associative law and distributive property of multiplication.

In contrast to the quaternion multiplication described, the dot product of the two quaternions results in a number, like the dot product of vectors

$$\boldsymbol{q_A} \cdot \boldsymbol{q_B} = q_{A0}q_{B0} + q_{A1}q_{B1} + q_{A2}q_{B2} + q_{A3}q_{B3} \tag{18}$$

If $\boldsymbol{q_A} \cdot \boldsymbol{q_B} = 0$, $\boldsymbol{q_A}$ and $\boldsymbol{q_B}$ are orthogonal to each other. Comparing equation (18) to the definition of quaternion multiplication shown by equation (17), $\boldsymbol{q_A} \cdot \boldsymbol{q_B}$ is the real part of the quaternion multiplication $\boldsymbol{q_A} \otimes \boldsymbol{q_B}^*$ or $\boldsymbol{q_A}^* \otimes \boldsymbol{q_B}$.

The norm or length of a quaternion $\boldsymbol{q}$ is defined as the square root of the quaternion multiplication of $\boldsymbol{q}$ and its conjugate $\boldsymbol{q}^*$

$$|\boldsymbol{q}| = \sqrt{\boldsymbol{q} \otimes \boldsymbol{q}^*} = \sqrt{{q_0}^2 + {q_1}^2 + {q_2}^2 + {q_3}^2} \tag{19}$$

If $\boldsymbol{q_A} \otimes \boldsymbol{q_B} = 1$, $\boldsymbol{q_B}$ is the inverse of $\boldsymbol{q_A}$, denoted as $\boldsymbol{q_A}^{-1}$. Generally the inverse of a quaternion $\boldsymbol{q}$ can be calculated by

$$\boldsymbol{q}^{-1} = \boldsymbol{q}^*/|\boldsymbol{q}| \tag{20}$$

For a unit quaternion $\boldsymbol{q}$, its inverse $\boldsymbol{q}^{-1}$ equals its conjugate $\boldsymbol{q}^*$

$$\boldsymbol{q}^{-1} = \boldsymbol{q}^* = q_0 - q_1 i - q_2 j - q_3 k \tag{21}$$

in this case, from (9), $\boldsymbol{q}^{-1}$ produces the reversed rotation of $\boldsymbol{q}$.

The addition of quaternions still follows the component-wise addition rule, like the addition of vectors.

$$\boldsymbol{q_A} + \boldsymbol{q_B} = (q_{A0} + q_{B0}) + (q_{A1} + q_{B1})i + (q_{A2} + q_{B2})j + (q_{A3} + q_{B3})k \tag{22}$$

Note that the addition of the quaternions might break the unit length constraint. Thus when quaternion addition happens, the result should be normalized by dividing the non-zero quaternion norm $|\boldsymbol{q}|$ to derive a new unit quaternion as a represent of rotation.

4.3 Rotation using Quaternion

A 3-elements vector $\boldsymbol{a} = [\, a_1, a_2, a_3]^T$ can be rotated by a unit quaternion $\boldsymbol{q}$ to produce a rotated 3-D vector $\boldsymbol{a}'$. The rotation calculation starts with concatenating a zero to the head of $\boldsymbol{a}$, which presents its quaternion form $\boldsymbol{q_a}$

$$\boldsymbol{q_a} = [\, 0, a_1, a_2, a_3]^T \tag{23}$$

Then $\boldsymbol{q_a}$ rotated by $\boldsymbol{q}$ is given by a quaternion product $\boldsymbol{q_{a'}}$ derived as

$$\boldsymbol{q_{a'}} = \boldsymbol{q}^{-1} \otimes \boldsymbol{q_a} \otimes \boldsymbol{q} \tag{24}$$

And the virtual part of $\boldsymbol{q_{a'}}$ presents the rotated 3-D vector $\boldsymbol{a}'$. For simplicity, in the discussion of the following chapters, the rotation of vector $\boldsymbol{a}$ using quaternion $\boldsymbol{q}$ will be denoted by

$$\boldsymbol{a}' = \boldsymbol{q}^{-1} \otimes \boldsymbol{a} \otimes \boldsymbol{q} \tag{25}$$

5. SENSOR FUSION USING FILTERING APPROACHES

Sensor fusion can happen in three levels: data-level fusion, feature-level fusion and decision-level fusion. Data-level fusion directly combines the data from multiple sensor modules. Feature-level fusion firstly parses the sensor data to generate corresponding features and then integrate these features. Decision-level fusion is a high-level approach that simply takes the computation results from each sensor module and makes choices based on these results. This project applies the data-level fusion method using recursive filters. The filters take the motion capture and IMU data as input, and provide the orientation estimation based on the latest observation.

Common sensor fusion algorithms include the least mean square filters, the Kalman filters and the particles filters. All these filters are recursive estimators. This means that only the estimated state from the previous time step and the current measurement, instead of the whole batch of observation and estimate history, are needed to compute the estimate for the current state. Generally the filtering process can be conceptualized as two

distinct phases: prediction and update. In the prediction phase, the state of the current time step is estimated based on the previous state. And in the update step, the current observation is employed to refine the result derived from the prediction phase. In this work, four filtering algorithms, the complementary filter, the extended Kalman filter, the unscented Kalman filter and the particle filter, are implemented and compared when fusing the observation from the motion capture system and the IMU for the quaternion-based orientation tracking task.

5.1 Quaternion-based Complementary Filter

In the proposed framework, the orientation of $\boldsymbol{C_M}$ is tracked in $\boldsymbol{C_G}$, and we have adopted the quaternion-based rotation representation. A quaternion-based complementary filter has been adopted for orientation estimation through sensor fusion. Figure 7 illustrates the key elements the proposed multimodal tracking algorithm using complementary filter. The complementary filter consists of the prediction and update steps. In the prediction step, the orientation at the current time is predicted according to the previous estimate and the current rotation rate observation from the gyro. When no gyro data is available, the previous orientation estimate is just used as the prediction. In the update step, the predicted orientation is refined in an iterative Gauss-Newton framework by minimizing the squared observation error. The Gauss–Newton algorithm is used to solve the non-linear least squares problem that aims to minimize a sum of squared function values. A resulting update vector is then obtained to adjust the prediction to get the final quaternion estimate. The sensor observations used for update are called "complementary data". In our research, the marker data and the readings from the accelerometer and the magnetometer have been used as the complementary data.

Table 2. Sensor Fusion Scenario Selections

Sensor Fusion Scenario	1	2	3	4	5	6	7	8	9	10
Motion Capture	√	×	√	×	√	×	√	×	√	×
Magnetometer	√	√	×	×	√	√	×	×	√	√
Accelerometer	√	√	√	√	×	×	×	×	√	√
Gyroscope	√	√	√	√	√	√	√	√	×	×

In practice, it is possible that not all of such complementary data is available in orientation tracking. Sometimes, a sensor is not selected in system design due to specific reasons. For example, in applications requiring low-cost sensing, the motion capture system can be optioned out to reduce the cost. In some other cases, certain data is unavailable due to sensor failure and large observation noise. For instance, marker occlusion can cause tracking failure for the motion capture system. In our research, we have systematically studied and evaluated the performance of the complementary filter for 10 typical sensing scenarios (Table 2) in multimodal orientation tracking according to the availability of complementary sensing data.

In Table 2, sensing scenarios 1, 3, 5, 7, 9 involve motion capture data and additional data from the IMU. In general, commercial motion capture systems provide precise tracking results. However, when simple motion capture systems with relatively low resolution and precision are employed, integrating the IMU data with the motion capture data can improve orientation tracking.

The sensing scenarios 2, 4, 6, 8, 10 in Table 2 correspond to the cases when the motion capture data is unavailable due to system design or marker occlusion. In these scenarios, the orientation tracking is performed by using only the IMU data. In Scenario 2, all the IMU data is available. In environments where ferromagnetic materials such as iron, nickel, or various alloys that exhibit extremely high magnetic exist, the magnetic field may be distorted and lack unification over space. Under such circumstances it is not wise

to trust the output from the magnetometer, which leads to Scenario 4, including only accelerometer and the gyroscope observations. In practice, when large accelerating movement occurs frequently, the accelerometer reading becomes unreliable. In such cases, Scenario 6 needs to be adopted using only the magnetometer and the gyroscope. Furthermore, Scenario 8 corresponds to the case when both the ferromagnetic materials and large accelerating movement are present simultaneously. In this scenario, only the gyro is used in tracking. Finally, in scenario 10 the magnetometer and the accelerometer are used in tracking, corresponding to cases when the gyro is not used.

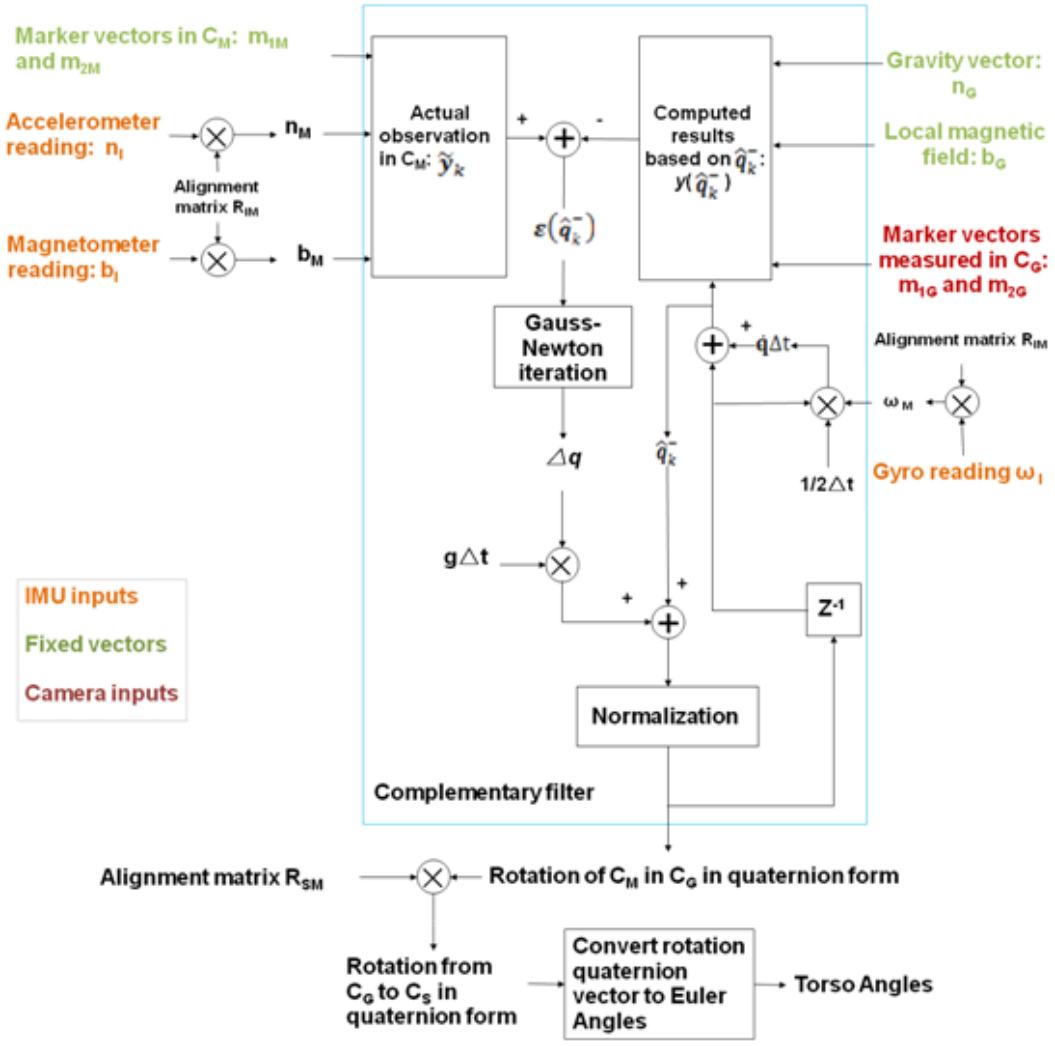

Fig. 7. Quaternion-based Complementary Filter for Orientation Tracking

5.1.1 Prediction of the Quaternion-based Complementary Filter

When the gyro data is available, it can be used in quaternion prediction. Let $\boldsymbol{\omega_I}$ be the rotation rate provided by the gyro, representing the rotation rate of $\boldsymbol{C_I}$ in $\boldsymbol{C_G}$:

$$\boldsymbol{\omega_I} = \left[\omega_{Ix}, \omega_{Iy}, \omega_{Iz}\right]^T \tag{26}$$

Using the alignment matrix R_{IM} from $\boldsymbol{C_I}$ to $\boldsymbol{C_M}$, the rotation rate of $\boldsymbol{C_M}$ can also be found as

$$\boldsymbol{\omega_M} = R_{IM}\boldsymbol{\omega_I} = [\omega_{Mx}, \omega_{My}, \omega_{Mz}]^T \tag{27}$$

Using $\boldsymbol{\omega_M}$, the time derivative of the quaternion representing the orientation of $\boldsymbol{C_M}$ in $\boldsymbol{C_G}$ is given by

$$\dot{\boldsymbol{q}}_k = \frac{1}{2}\hat{\boldsymbol{q}}_{k-1} \otimes \boldsymbol{\omega_M} \tag{28}$$

where $\hat{\boldsymbol{q}}_{k-1}$ is the quaternion estimate at time *k - 1*. Then the predicted quaternion $\tilde{\boldsymbol{q}}_{k+1}$ is

$$\hat{\boldsymbol{q}}_k^- = \hat{\boldsymbol{q}}_{k-1} + \dot{\boldsymbol{q}}_k \Delta t \tag{29}$$

where Δt is the sampling period. In our implementation, Δt=10ms. The derivation of $\hat{\boldsymbol{q}}_k^-$ is summarized from equation (28) – (29) as function

$$\hat{\boldsymbol{q}}_k^- = d(\hat{\boldsymbol{q}}_{k-1}, \boldsymbol{\omega_M}) = \hat{\boldsymbol{q}}_{k-1} + \frac{\Delta t}{2}\hat{\boldsymbol{q}}_{k-1} \otimes \boldsymbol{\omega_M} \tag{30}$$

When the gyro data is unavailable, the current quaternion estimate is directly used as the prediction for the next time instant:

$$\hat{\boldsymbol{q}}_k^- = \hat{\boldsymbol{q}}_{k-1} \tag{31}$$

5.1.2 Update of the Quaternion-based Complementary Filter

When complementary data are available, they are used to further refine $\hat{\boldsymbol{q}}_k^-$. An update vector $\Delta\boldsymbol{q}$ is found by minimizing the squared error between the observed and the predicted complementary reference vectors. When the marker data is available, two vectors formed by three markers are used as the maker reference vectors. When the

accelerometer is used, the inertial measurement is used as the inertial reference vector. When the magnetometer is used, the magnetic measurement is used as the magnetic reference vector.

During tracking, the inertial and magnetic reference vectors are measured in $\boldsymbol{C_I}$. Let $\tilde{\boldsymbol{n}}_I$ and $\tilde{\boldsymbol{b}}_I$ be these inertial and magnetic reference vectors measured in $\boldsymbol{C_I}$, and $\tilde{\boldsymbol{n}}_{\boldsymbol{M}}$ and $\tilde{\boldsymbol{b}}_{\boldsymbol{M}}$ the same vectors measured in $\boldsymbol{C_I}$. Given $\tilde{\boldsymbol{n}}_I$ and $\tilde{\boldsymbol{b}}_I$, $\tilde{\boldsymbol{n}}_{\boldsymbol{M}}$ and $\tilde{\boldsymbol{b}}_{\boldsymbol{M}}$ can be found using R_{IM} as

$$\tilde{\boldsymbol{n}}_{\boldsymbol{M}} = R_{IM}\tilde{\boldsymbol{n}}_I\,, \quad \tilde{\boldsymbol{b}}_{\boldsymbol{M}} = R_{IM}\tilde{\boldsymbol{b}}_I \tag{32}$$

During tracking, the marker reference vectors are measured in $\boldsymbol{C_G}$ as $\tilde{\boldsymbol{m}}_{\boldsymbol{1G}}$ and $\tilde{\boldsymbol{m}}_{\boldsymbol{2G}}$. For a rotation quaternion prediction $\hat{\boldsymbol{q}}_k^-$, the predicted reference vectors are

$$\begin{cases} \boldsymbol{n}(\hat{\boldsymbol{q}}_k^-) = \hat{\boldsymbol{q}}_k^{-\boldsymbol{1}} \otimes \boldsymbol{n_G} \otimes \hat{\boldsymbol{q}}_k^- \\ \boldsymbol{b}(\hat{\boldsymbol{q}}_k^-) = \hat{\boldsymbol{q}}_k^{-\boldsymbol{1}} \otimes \boldsymbol{b_G} \otimes \hat{\boldsymbol{q}}_k^- \\ \boldsymbol{m_1}(\hat{\boldsymbol{q}}_k^-) = \hat{\boldsymbol{q}}_k^- \otimes \boldsymbol{m_{1M}} \otimes \hat{\boldsymbol{q}}_k^{-\,-1} \\ \boldsymbol{m_2}(\hat{\boldsymbol{q}}_k^-) = \hat{\boldsymbol{q}}_k^- \otimes \boldsymbol{m_{2M}} \otimes \hat{\boldsymbol{q}}_k^{-\,-1} \end{cases} \tag{33}$$

where $\boldsymbol{n_G}$ and $\boldsymbol{b_G}$ are the gravity vector and the magnetic field vector in $\boldsymbol{C_G}$ as given in (5) and (6). Both $\boldsymbol{n_G}$ and $\boldsymbol{b_G}$ are obtained during system calibration. In (33), $\boldsymbol{m_{1M}}$ and $\boldsymbol{m_{2M}}$ are the marker reference vectors in $\boldsymbol{C_M}$ as shown in Figure 3. These two reference vectors are constant for a fixed back marker set.

According to the sensor availability, at time *k* multiple combinations exist to formulate the observation reference vector $\boldsymbol{y}_k$ and the corresponding predicted reference vector $(\hat{\boldsymbol{q}}_k^-)$. Table 3 lists the integrated reference vectors and their predictions for all the 10 typical scenarios (Table 2) discussed in our research. At time *k*, given $\tilde{\boldsymbol{y}}_k$ and the quaternion prediction $\hat{\boldsymbol{q}}_k^-$, the weighted measurement prediction error $\boldsymbol{\varepsilon}(\hat{\boldsymbol{q}}_k^-)$ is

$$\boldsymbol{\varepsilon}(\hat{\boldsymbol{q}}_k^-) = \boldsymbol{W} \cdot [\,\tilde{\boldsymbol{y}}_k - \boldsymbol{y}(\hat{\boldsymbol{q}}_k^-)] \tag{34}$$

where $\boldsymbol{W}$ is a diagonal weighting matrix and its elements are computed as the inverse of the standard deviations of the components in $\tilde{\boldsymbol{n}}_{\boldsymbol{M}}$, $\tilde{\boldsymbol{b}}_{\boldsymbol{M}}$ and $\tilde{\boldsymbol{m}}_{\boldsymbol{1G}}$ and $\tilde{\boldsymbol{m}}_{\boldsymbol{2G}}$when the sensor board is static. The squared error function is given by

$$\boldsymbol{\varphi}(\hat{\boldsymbol{q}}_k^-) = \boldsymbol{\varepsilon}(\hat{\boldsymbol{q}}_k^-)^{\boldsymbol{T}}\boldsymbol{\varepsilon}(\hat{\boldsymbol{q}}_k^-) \tag{35}$$

Table 3. Complementary Data in Typical Sensing Scenarios

Scenarios	Prediction	Update using complementary data	
		$\tilde{\boldsymbol{y}}_k$	$\boldsymbol{y}(\hat{\boldsymbol{q}}_k^-)$
1	√	$[\tilde{\boldsymbol{n}}_{\boldsymbol{M}}{}^T, \tilde{\boldsymbol{b}}_{\boldsymbol{M}}{}^T, \tilde{\boldsymbol{m}}_{\boldsymbol{1G}}{}^T, \tilde{\boldsymbol{m}}_{\boldsymbol{2G}}{}^T]^T$	$[\boldsymbol{n}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{b}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{m}_1(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{m}_2(\hat{\boldsymbol{q}}_k^-)^T]^T$
2	√	$[\tilde{\boldsymbol{n}}_{\boldsymbol{M}}{}^T, \tilde{\boldsymbol{b}}_{\boldsymbol{M}}{}^T]^T$	$[\boldsymbol{n}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{b}(\hat{\boldsymbol{q}}_k^-)^T]^T$
3	√	$[\tilde{\boldsymbol{n}}_{\boldsymbol{M}}{}^T, \tilde{\boldsymbol{m}}_{\boldsymbol{1G}}{}^T, \tilde{\boldsymbol{m}}_{\boldsymbol{2G}}{}^T]^T$	$[\boldsymbol{n}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{m}_1(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{m}_2(\hat{\boldsymbol{q}}_k^-)^T]^T$
4	√	$\tilde{\boldsymbol{n}}_{\boldsymbol{M}}$	$\boldsymbol{n}(\hat{\boldsymbol{q}}_k^-)$
5	√	$[\tilde{\boldsymbol{b}}_{\boldsymbol{M}}{}^T, \tilde{\boldsymbol{m}}_{\boldsymbol{1G}}{}^T, \tilde{\boldsymbol{m}}_{\boldsymbol{2G}}{}^T]^T$	$[\boldsymbol{b}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{m}_1(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{m}_2(\hat{\boldsymbol{q}}_k^-)^T]^T$
6	√	$\tilde{\boldsymbol{b}}_{\boldsymbol{M}}$	$\boldsymbol{b}(\hat{\boldsymbol{q}}_k^-)$
7	√	$[\tilde{\boldsymbol{m}}_{\boldsymbol{1G}}{}^T, \tilde{\boldsymbol{m}}_{\boldsymbol{2G}}{}^T]^T$	$[\boldsymbol{m}_1(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{m}_2(\hat{\boldsymbol{q}}_k^-)^T]^T$
8	√	N/A	N/A
9	×	$[\tilde{\boldsymbol{n}}_{\boldsymbol{M}}{}^T, \tilde{\boldsymbol{b}}_{\boldsymbol{M}}{}^T, \tilde{\boldsymbol{m}}_{\boldsymbol{1G}}{}^T, \tilde{\boldsymbol{m}}_{\boldsymbol{2G}}{}^T]^T$	$[\boldsymbol{n}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{b}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{m}_1(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{m}_2(\hat{\boldsymbol{q}}_k^-)^T]^T$
10	×	$[\tilde{\boldsymbol{n}}_{\boldsymbol{M}}{}^T, \tilde{\boldsymbol{b}}_{\boldsymbol{M}}{}^T]^T$	$[\boldsymbol{n}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{b}(\hat{\boldsymbol{q}}_k^-)^T]^T$

The squared error function $\boldsymbol{\varphi}(\hat{\boldsymbol{q}}_k^-)$ is minimized by the Gauss-Newton iteration [31]. The Gauss–Newton algorithm solves non-linear least squares problems by forcing the gradient (vector derivative) of the squared error function to zero. In this technique, the prediction measurement function $\boldsymbol{y}(\boldsymbol{q})$ is approximated by the first two terms of its Taylor series expansion at $\hat{\boldsymbol{q}}_k^-$

$$\boldsymbol{y}(\hat{\boldsymbol{q}}_k^- + \Delta\boldsymbol{q}) = \boldsymbol{y}(\hat{\boldsymbol{q}}_k^-) + \boldsymbol{X}\Delta\boldsymbol{q} + \boldsymbol{o}(\Delta\boldsymbol{q}^2) \tag{36}$$

where $\boldsymbol{X}$ is the Jacobi matrix of $\boldsymbol{y}(\boldsymbol{q})$ evaluated at $\hat{\boldsymbol{q}}_k^-$

$$\boldsymbol{X}_{\boldsymbol{ij}} = \left.\frac{\partial \mathrm{y_i}}{\partial \boldsymbol{q}_\mathrm{j}}\right|_{\boldsymbol{q}=\hat{\boldsymbol{q}}_k^-} \tag{37}$$

According to the method given in Appendix A, the Jacobi matrix of the predicted measurement $\boldsymbol{n}(\hat{\boldsymbol{q}}_k^-)$, $\boldsymbol{b}(\hat{\boldsymbol{q}}_k^-)$, $\boldsymbol{m}_1(\hat{\boldsymbol{q}}_k^-)$ and $\boldsymbol{m}_2(\hat{\boldsymbol{q}}_k^-)$, noted as $\boldsymbol{J}_{\boldsymbol{n}}$, $\boldsymbol{J}_{\boldsymbol{b}}$, $\boldsymbol{J}_{\boldsymbol{m1}}$ and $\boldsymbol{J}_{\boldsymbol{m2}}$, can be derived. The combination of these Jacobi matrices forms $\boldsymbol{X}$, as shown in Table 4.

Table 4. Jacobi Matrix *X* for Different Scenarios

Scenarios	$\boldsymbol{y}(\hat{\boldsymbol{q}}_k^-)$	$\boldsymbol{X}$
1	$[\boldsymbol{n}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{b}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{m_1}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{m_2}(\hat{\boldsymbol{q}}_k^-)^T]^T$	$[\boldsymbol{J_n}^T, \boldsymbol{J_b}^T, \boldsymbol{J_{m1}}^T, \boldsymbol{J_{m2}}^T]^T$
2	$[\boldsymbol{n}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{b}(\hat{\boldsymbol{q}}_k^-)^T]^T$	$[\boldsymbol{J_n}^T, \boldsymbol{J_b}^T]^T$
3	$[\boldsymbol{n}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{m_1}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{m_2}(\hat{\boldsymbol{q}}_k^-)^T]^T$	$[\boldsymbol{J_n}^T, \boldsymbol{J_{m1}}^T, \boldsymbol{J_{m2}}^T]^T$
4	$\boldsymbol{n}(\hat{\boldsymbol{q}}_k^-)$	$\boldsymbol{J_n}$
5	$[\boldsymbol{b}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{m_1}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{m_2}(\hat{\boldsymbol{q}}_k^-)^T]^T$	$[\boldsymbol{J_b}^T, \boldsymbol{J_{m1}}^T, \boldsymbol{J_{m2}}^T]^T$
6	$\boldsymbol{b}(\hat{\boldsymbol{q}}_k^-)$	$\boldsymbol{J_b}$
7	$[\boldsymbol{m_1}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{m_2}(\hat{\boldsymbol{q}}_k^-)^T]^T$	$[\boldsymbol{J_{m1}}^T, \boldsymbol{J_{m2}}^T]^T$
8	N/A	N/A
9	$[\boldsymbol{n}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{b}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{m_1}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{m_2}(\hat{\boldsymbol{q}}_k^-)^T]^T$	$[\boldsymbol{J_n}^T, \boldsymbol{J_b}^T, \boldsymbol{J_{m1}}^T, \boldsymbol{J_{m2}}^T]^T$
10	$[\boldsymbol{n}(\hat{\boldsymbol{q}}_k^-)^T, \boldsymbol{b}(\hat{\boldsymbol{q}}_k^-)^T]^T$	$[\boldsymbol{J_n}^T, \boldsymbol{J_b}^T]^T$

Using the linear part of (36), the weighted measurement prediction error $\boldsymbol{\varepsilon}(\hat{\boldsymbol{q}}_k^-)$ can be approximated as

$$\begin{aligned}\boldsymbol{\varepsilon}(\hat{\boldsymbol{q}}_k^- + \Delta\boldsymbol{q}) &= \boldsymbol{W}[\tilde{\boldsymbol{y}}_k - \boldsymbol{y}(\hat{\boldsymbol{q}}_k^-) - \boldsymbol{X}\Delta\boldsymbol{q}] \\ &= \boldsymbol{\varepsilon}(\hat{\boldsymbol{q}}_k^-) - \boldsymbol{WX}\Delta\boldsymbol{q}]\end{aligned} \tag{38}$$

According to the inverse law of transposed matrices

$$\boldsymbol{\varepsilon}(\hat{\boldsymbol{q}}_k^- + \Delta\boldsymbol{q})^{\boldsymbol{T}} = \boldsymbol{\varepsilon}(\hat{\boldsymbol{q}}_k^-)^{\boldsymbol{T}} - \Delta\boldsymbol{q}^{\boldsymbol{T}}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}} \tag{39}$$

Substitute (38) and (39) into the squared error function (35)

$$\begin{aligned}\boldsymbol{\varphi}(\hat{\boldsymbol{q}}_k^- + \Delta\boldsymbol{q}) &= \boldsymbol{\varepsilon}(\hat{\boldsymbol{q}}_k^- + \Delta\boldsymbol{q})^{\boldsymbol{T}}\boldsymbol{\varepsilon}(\hat{\boldsymbol{q}}_k^- + \Delta\boldsymbol{q}) \\ &= \boldsymbol{\varepsilon}(\hat{\boldsymbol{q}}_k^-)^{\boldsymbol{T}}\boldsymbol{\varepsilon}(\hat{\boldsymbol{q}}_k^-) - \boldsymbol{\varepsilon}(\hat{\boldsymbol{q}}_k^-)^{\boldsymbol{T}}\boldsymbol{WX}\Delta\boldsymbol{q} \\ &\quad - \Delta\boldsymbol{q}^{\boldsymbol{T}}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{\varepsilon}(\hat{\boldsymbol{q}}_k^-) + \Delta\boldsymbol{q}^{\boldsymbol{T}}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{WX}\Delta\boldsymbol{q}\end{aligned} \tag{40}$$

Collect terms by making use of the inverse law of transposed matrices again, which provides

$$\boldsymbol{\varepsilon}(\hat{\boldsymbol{q}}_k^-)^{\boldsymbol{T}}\boldsymbol{WX}\Delta\boldsymbol{q} = \Delta\boldsymbol{q}^{\boldsymbol{T}}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{\varepsilon}(\hat{\boldsymbol{q}}_k^-) \tag{41}$$

Then equation (40) can be simplified to

$$\boldsymbol{\varphi}\left(\hat{\boldsymbol{q}}_k^- + \Delta\boldsymbol{q}\right) = \boldsymbol{\varepsilon}\left(\hat{\boldsymbol{q}}_k^- + \Delta\boldsymbol{q}\right)^{\boldsymbol{T}}\boldsymbol{\varepsilon}\left(\hat{\boldsymbol{q}}_k^- + \Delta\boldsymbol{q}\right)$$

$$= \boldsymbol{\varepsilon}\left(\hat{\boldsymbol{q}}_k^-\right)^{\boldsymbol{T}}\boldsymbol{\varepsilon}\left(\hat{\boldsymbol{q}}_k^-\right) - 2\Delta\boldsymbol{q}^{\boldsymbol{T}}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{\varepsilon}\left(\hat{\boldsymbol{q}}_k^-\right) + \Delta\boldsymbol{q}^{\boldsymbol{T}}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{W}\boldsymbol{X}\Delta\boldsymbol{q} \tag{42}$$

Based on vector calculus, the gradient (vector derivative) of $\boldsymbol{\varphi}(\boldsymbol{q})$ at $\widetilde{\boldsymbol{q}}_{\boldsymbol{k+1}}$is expressed as

$$\frac{\mathrm{d}\boldsymbol{\varphi}(\boldsymbol{q})}{\mathrm{d}\boldsymbol{q}}\Big|_{\boldsymbol{q}=\hat{\boldsymbol{q}}_k^-} = -2\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{\varepsilon}\left(\hat{\boldsymbol{q}}_k^-\right) - 2\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{W}\boldsymbol{X}\Delta\boldsymbol{q} \tag{43}$$

It can be proved that when $\boldsymbol{X}$ is of full rank, the squared error function $\boldsymbol{\varphi}$ given by (35) is follows a positive definite quadratic form. And the unique minimum of the positive definite $\boldsymbol{\varphi}(\boldsymbol{q})$ can be found when the gradient of $\boldsymbol{\varphi}(\boldsymbol{q})$ at $\hat{\boldsymbol{q}}_k^-$ equals zero, which yields the result

$$\Delta\boldsymbol{q} = \left[\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{W}\boldsymbol{X}\right]^{-1}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{\varepsilon}\left(\hat{\boldsymbol{q}}_k^-\right) \tag{44}$$

Then the refined quaternion estimate at time n+1 is given by

$$\widehat{\boldsymbol{q}}_k = \hat{\boldsymbol{q}}_k^- + g\Delta t\Delta\boldsymbol{q} = \hat{\boldsymbol{q}}_k^- + \Delta\boldsymbol{q}_k \tag{45}$$

where g is a scalar filter gain.

5.1.3 Reduced Order Filter

Note that $\widehat{\boldsymbol{q}}_k$ has to remain as a unit quaternion, so that the calculation of the Jacobi matrix $\boldsymbol{X}$ is valid. Moreover, such constraint promises the solution $\Delta\boldsymbol{q}_k$ is unique. Or else, $\boldsymbol{X}$ will not be of full rank, and $\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}$ turns to be singular.

The reduced order filter promises the updated $\widehat{\boldsymbol{q}}_k$ is still a unit quaternion when it is updated by (45) and considerably simplifies the matrix computation. In (45), $\widehat{\boldsymbol{q}}_k^-$ is a unit quaternion. The norm of the updated quaternion orientation $\widehat{\boldsymbol{q}}_k$ is

$$|\widehat{\boldsymbol{q}}_k| = \left|\hat{\boldsymbol{q}}_k^- + \Delta\boldsymbol{q}_k\right|$$

$$= \sqrt{\left|\hat{\boldsymbol{q}}_k^-\right|^2 + \left|\Delta\boldsymbol{q}_k\right|^2 + 2\hat{\boldsymbol{q}}_k^- \cdot \Delta\boldsymbol{q}_k}$$

$$\approx \sqrt{\left|\hat{\boldsymbol{q}}_k^-\right|^2 + 2\hat{\boldsymbol{q}}_k^- \cdot \Delta\boldsymbol{q}_k}$$

$$= \sqrt{1 + 2\hat{\boldsymbol{q}}_k^- \cdot \Delta \boldsymbol{q}_k}$$

$$= \sqrt{1 + 2k\Delta t \hat{\boldsymbol{q}}_k^- \cdot \Delta \boldsymbol{q}} \tag{46}$$

Thus only when $\Delta \boldsymbol{q}$ is orthogonal to $\hat{\boldsymbol{q}}_k^-$ ($\hat{\boldsymbol{q}}_k^- \cdot \Delta \boldsymbol{q} = 0$), the length of the updated orientation quaternion $\hat{\boldsymbol{q}}_k$ remains to be 1. According to the Orthogonal Quaternion Theorem of [15], if $\Delta \boldsymbol{q}$ is orthogonal to $\hat{\boldsymbol{q}}_k^-$, it can be written into the form

$$\Delta \boldsymbol{q} = \hat{\boldsymbol{q}}_k^- \otimes \boldsymbol{p} \tag{47}$$

where $\boldsymbol{p}$ is a vector given by

$$\boldsymbol{p} = p_1 i + p_2 j + p_3 k \tag{48}$$

Consequently, only a 3-element vector $\boldsymbol{p}$, instead of a 4-element quaternion $\Delta \boldsymbol{q}$, needs to be solved to update $\hat{\boldsymbol{q}}$. The vector can still be solved using the Gauss-Newton method as follows.

Replace the $\Delta \boldsymbol{q}$ in (38) with (48), the linearization of the predicted measurement becomes

$$\boldsymbol{y}(\hat{\boldsymbol{q}}_k^- + \Delta \boldsymbol{q}) \approx \boldsymbol{y}(\hat{\boldsymbol{q}}_k^-) + \boldsymbol{X}(\hat{\boldsymbol{q}}_k^- \otimes \boldsymbol{p})$$

$$= \boldsymbol{y}(\hat{\boldsymbol{q}}_k^-) + \boldsymbol{X}(\hat{\boldsymbol{q}}_k^- \otimes [0, p_1, p_2, p_3]) \tag{49}$$

And the partial differential calculus of $\boldsymbol{y}(\boldsymbol{q})$ with respect to $\boldsymbol{p}$ is given by

$$\frac{\partial \mathbf{y}}{\partial p_1} = \boldsymbol{X}(\hat{\boldsymbol{q}}_k^- \otimes i) \tag{50}$$

$$\frac{\partial \mathbf{y}}{\partial p_2} = \boldsymbol{X}(\hat{\boldsymbol{q}}_k^- \otimes j) \tag{51}$$

$$\frac{\partial \mathbf{y}}{\partial p_3} = \boldsymbol{X}(\hat{\boldsymbol{q}}_k^- \otimes k) \tag{52}$$

Now build a new Jacobi matrix $\boldsymbol{X}_{\boldsymbol{p}}$ by combining (50) – (52)

$$\boldsymbol{X}_{\boldsymbol{p}} = [\frac{\partial \mathbf{y}}{\partial p_1} | \frac{\partial \mathbf{y}}{\partial p_2} | \frac{\partial \mathbf{y}}{\partial p_3}] \tag{53}$$

Consequently the predicted measurement $\boldsymbol{y}(\boldsymbol{q})$ at $\hat{\boldsymbol{q}}_k^-$ can be approximated by

$$\boldsymbol{y}\left(\hat{\boldsymbol{q}}_k^- + (\hat{\boldsymbol{q}}_k^- \otimes \boldsymbol{p})\right) \approx \boldsymbol{y}\left(\hat{\boldsymbol{q}}_k^-\right) + \boldsymbol{X_p p} \tag{54}$$

Then the minimum of the positive definite $\boldsymbol{\varphi}(\boldsymbol{q})$ can be found using the method specified by equation (34) – (44), which yields the result

$$\boldsymbol{p} = \left[\boldsymbol{X_p}^T \boldsymbol{W}^T \boldsymbol{W} \boldsymbol{X_p}\right]^{-1} \boldsymbol{X_p}^T \boldsymbol{W}^T \boldsymbol{\varepsilon}\left(\hat{\boldsymbol{q}}_k^-\right) \tag{55}$$

And the final update quaternion $\Delta\boldsymbol{q}$ can be derived by equation (47). Since $\boldsymbol{X_p}$ is a 3 by 3 matrix while $\boldsymbol{X}$ is a 4 by 4 matrix, the calculation of (55) is more efficient that given by (44).

5.1.4 Tracking with Single Reference Vector

The complementary filter-based tracking algorithm described in previous section necessitates at least two reference vectors to decide the rotation between two coordinate systems. In practice, it is possible that the magnetometer or the accelerometer is the only available complementary sensor (Scenarios 4 and 6 in Table 1). In these cases, extra care needs to be taken to secure the tracking stability. When only one reference vector is available, the rotation about the direction of the reference vector should not be updated for such rotation does not affect the measurement prediction error.

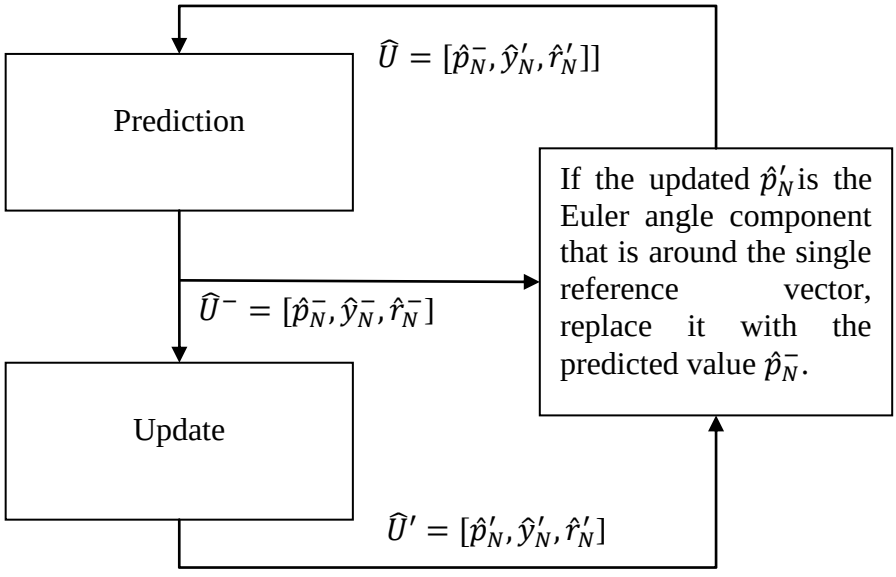

Fig. 8. Filtering Strategy with Single Reference Vector

To track the orientation with only one reference vector, the rotation about the reference vector obtained in the Gauss-Newton iteration needs to be excluded. To this end, we have developed a method to accomplish such constrained update. Assume that the local magnetic field is the only complementary data available. Define a new global coordinate system $\boldsymbol{C_N}$ and one of its axes (the X-axis in our research) is aligned with the local magnetic field. The spatial alignment between $\boldsymbol{C_N}$ and $\boldsymbol{C_G}$ is predetermined by a rotation matrixR_{NG}.

During the orientation tracking, before the Gauss-Newton iteration, the predicted quaternion $\hat{\boldsymbol{q}}_k^-$ is first converted to a rotation matrix R_{GM}^-, indicating the predicted rotation from $\boldsymbol{C_G}$ to $\boldsymbol{C_M}$. Using R_{NG}, the predicted rotation matrix from $\boldsymbol{C_N}$ to $\boldsymbol{C_M}$ is given by

$$R_{NM}^- = R_{GM}^- R_{NG} \tag{56}$$

And the corresponding Euler angles (in X-Y-Z convention) representing the rotation from $\boldsymbol{C_N}$ to $\boldsymbol{C_M}$ are

$$\hat{U}^- = [\hat{p}_N^-, \hat{y}_N^-, \hat{r}_N^-] \tag{57}$$

where $\hat{p}_N^-$ is the Euler angle around the X-axis of $\boldsymbol{C_N}$, parallel to the local magnetic field. After the Gauss-Newton process as introduced in the previous section, the above quaternion-to-Euler angles procedure is carried out again to infer the Euler angles from $\boldsymbol{C_N}$ to $\boldsymbol{C_M}$

$$\hat{U}' = [\hat{p}_N', \hat{y}_N', \hat{r}_N'] \tag{58}$$

using the updated quaternion. Since $p_N'$is unobservable from only the local magnetic field, it should not be updated in the Gauss-Newton process. Therefore, a refined set of Euler angles $\hat{U}$ is obtained by replacing $\hat{p}_N^-$ in $\hat{U}'$with $\hat{p}_N^-$ in$\hat{U}^-$:

$$\hat{U} = [\hat{p}_N^-, \hat{y}_N', \hat{r}_N'] \tag{59}$$

$\hat{U}$ is then taken as the updated Euler angle from $\boldsymbol{C_N}$ to $\boldsymbol{C_M}$, as illustrated by Figure 8. The corresponding quaternion $\hat{\boldsymbol{q}}_{\boldsymbol{n+1}}$from $\boldsymbol{C_G}$ to $\boldsymbol{C_M}$ can be found by first obtaining the rotation

matrix from $\boldsymbol{C_G}$ to $\boldsymbol{C_M}$ as $R_{GM} = R_{NM}R_{NG}^T$ and then inferring $\widehat{\boldsymbol{q}}_k$ from R_{GM}. The same approach can be adopted when only the inertial data is available. Such strategy can also be applied to other filtering algorithms described in the following sections about the Kalman filters (KF) and the particle filter (PF). Like the complementary filter, KF and PF compose of the prediction and update steps, thus the process illustrated by Figure 8 can be implemented on them in a similar measure. As shown by the experimental results, this proposed algorithm is effective in improving tracking stability when only one reference vector is available.

5.2 Quaternion-based Extended Kalman Filter

The Kalman filters aims to use measurements $\boldsymbol{z}_k$ undermined by noise or other inaccuracies to reconstruct values that tend to be closer to the true values of the measurements and their associated calculated values represented by a state vector $\boldsymbol{x}_k$. Classical Kalman is a recursive estimator based on linear dynamical systems discretized in the time domain described by the process model (predict model, state transition model) and the measurement model (observation model). The process model predicts the revolution of the state vector and describes the influence of the process noise based on the state transition function $f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k, \boldsymbol{w}_k)$

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k, \boldsymbol{w}_k) = F_k \boldsymbol{x}_{k-1} + B_k \boldsymbol{u}_k + \boldsymbol{w}_k \tag{60}$$

where F is the state transition matrix applied to the estimated state vector of previous iteration $\boldsymbol{x}_{k-1}$, B is the input-control matrix applied to input $\boldsymbol{u}_k$, and $\boldsymbol{w}_k$ is the process noise (system noise) that follows a zero mean multivariate normal distribution with the covariance matrix Q. For systems with no control input, like the proposed orientation tracking system, the state transition function can be simplified to be $f(\boldsymbol{x}_{k-1}, \boldsymbol{w}_k)$

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{w}_k) = F_k \boldsymbol{x}_{k-1} + \boldsymbol{w}_k \tag{61}$$

The measurement model generates the measurement (or observation) $\boldsymbol{z}_k$ of the true state $\boldsymbol{x}_k$ according to the measurement function $h(\ \boldsymbol{x}_k\ ,\ \boldsymbol{v}_k)$

$$\boldsymbol{z}_k = h(\ \boldsymbol{x}_k\ ,\ \boldsymbol{v}_k) = H_k\ \boldsymbol{x}_k + \ \boldsymbol{v}_k \tag{62}$$

where H is the measurement matrix multiplied to the state vector of the current time step $\boldsymbol{x}_k$, and $\boldsymbol{v}_k$ is the measurement noise (observation noise) that follows a zero mean multivariate normal distribution with the covariance matrix R. Note that both the measurement function $f(\ \boldsymbol{x}_{k-1}\ ,\ \boldsymbol{w}_k)$ and the measurement function $h(\ \boldsymbol{x}_k\ ,\ \boldsymbol{v}_k)$ are of linear form, which is the underlying precondition for implementing the basic Kalman filter.

The filtering process of the Kalman filters contains two phases: prediction and update. The prediction phase uses the state estimate from the previous time step to produce an estimate of the state at the current time step. This predicted state estimate is also known as the a priori state estimate because, although it is an estimate of the state at the current time step, it does not include observation information at the current time step. In the update phase, the priori prediction is combined with the current observation information to refine the state estimate. This improved estimate is termed as the a posteriori state estimate. Given the previous posteriori state estimate $\widehat{\boldsymbol{x}}_{k-1}$, the prediction step calculates a priori state estimate $\widehat{\boldsymbol{x}}_k^-$ according to

$$\widehat{\boldsymbol{x}}_k^- = f(\ \widehat{\boldsymbol{x}}_{k-1}\ ,\boldsymbol{0}) \tag{63}$$

In addition, a priori covariance matrix P_k^- is produced by

$$P_k^- = F_k P_{k-1} {F_k}^T + Q \tag{64}$$

where P_{k-1} is an posteriori estimate covariance matrix as a measure of the accuracy of the previous state estimate. In the update step, $\widehat{\boldsymbol{x}}_k^-$ and P_k^- are further refined by

$$\widehat{\boldsymbol{x}}_k = \widehat{\boldsymbol{x}}_k^- + \ K_k\big(\ \boldsymbol{z}_k - \ h(\ \widehat{\boldsymbol{x}}_k^-\ ,\boldsymbol{0})\big) \tag{65}$$

$$P_k\ = (I - \ K_k H_k) P_k^- \tag{66}$$

where K_k is the Kalman gain defined as

$$K_k = P_k^- H_k{}^T (H_k P_k^- H_k{}^T + R)^{-1} \tag{67}$$

It can be proved that Kalman filter yields MMSE (minimum mean-square error) estimate when applied to systems that can be modeled by the linear measurement function $f(\boldsymbol{x}_{k-1}, \boldsymbol{w}_k)$ and the measurement function $h(\boldsymbol{x}_k, \boldsymbol{v}_k)$. Somehow such conditions are highly restrictive. The difficulty of filtering unit quaternion data stems from the nonlinear nature of unit quaternion space. Since the implementation of classical Kalman filter necessitates linear non-linear process and measurement models, it is not capable of estimating the quaternion state. The extended Kalman Filter (EKF) is an expansion of the Kalman filter to the nonlinear system simply by linearizing the nonlinear models so that the traditional linear Kalman filter equations can be applied.

5.2.1 State Vector

The state vector $\boldsymbol{x}$ of the quaternion-based EKF has two forms depending on whether the gyroscope data is involved for the orientation tracking. When using the gyroscope measurement, the state vector $\boldsymbol{x}$ comprises of two parts: the orientation in the form of a unit quaternion $\boldsymbol{q} = [q_0, q_1, q_2, q_3]^T$ and the rotation rate $\boldsymbol{\omega_M} = [\omega_{Mx}, \omega_{My}, \omega_{Mz}]^T$

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{\omega_M} \end{bmatrix} \tag{68}$$

Thus the length of the state vector is 7. The quaternion part $\boldsymbol{q}$ holds the same meaning as in section 5.1, representing the orientation of $\boldsymbol{C_M}$ in $\boldsymbol{C_G}$. And $\boldsymbol{\omega_M}$ is the angular velocity of $\boldsymbol{C_M}$ in $\boldsymbol{C_G}$ derived from the measurement of the gyroscope by (27). When the gyroscope measurement is not utilized, the state vector $\boldsymbol{x}$ is simplified to the 4-element quaternion

$$\boldsymbol{x} = \boldsymbol{q} \tag{69}$$

5.2.2 Process Model

If gyroscope data is not involved, the priori state estimate $\boldsymbol{x}_k$ is simply previous posteriori state estimate $\boldsymbol{x}_{k-1}$ plus noise. Then

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{w}_k) = \boldsymbol{x}_{k-1} + \boldsymbol{w}_k = \boldsymbol{q}_{k-1} + [\boldsymbol{w}_q]_k \tag{70}$$

where $\boldsymbol{w}_q$ is a 4-element vector denoting the process noise of the quaternion.

When gyroscope data is involved, it can contribute to the prediction of the quaternion orientation estimate within the process model according to a nonlinear state transition function

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{w}_k) = f\left(\begin{bmatrix}\boldsymbol{q}\\ \boldsymbol{\omega_M}\end{bmatrix}_{k-1}, \begin{bmatrix}\boldsymbol{w_q}\\ \boldsymbol{w_\omega}\end{bmatrix}_k\right) = \begin{bmatrix} d(\boldsymbol{q}_{k-1}, \boldsymbol{\omega_M}_{k-1}) \\ \boldsymbol{\omega_M}_{k-1}\end{bmatrix} + \begin{bmatrix}\boldsymbol{w_q}\\ \boldsymbol{w_\omega}\end{bmatrix}_k \tag{71}$$

where $\boldsymbol{w_\omega}$ is a 3-element vector denoting the process noise of the quaternion. The angular velocity integration function $d(\boldsymbol{q}_{k-1}, \boldsymbol{\omega_M}_{k-1})$is defined by equation (30) in section 5.1.1.

Table 5. Measurement Noise v in Different Sensor Fusion Scenarios

Scenarios	v
1	$[v_n^{\boldsymbol{T}}, v_b^{\boldsymbol{T}}, v_m^{\boldsymbol{T}}, v_\omega^{\boldsymbol{T}}]^{\boldsymbol{T}}$
2	$[v_n^{\boldsymbol{T}}, v_b^{\boldsymbol{T}}, v_\omega^{\boldsymbol{T}}]^{\boldsymbol{T}}$
3	$[v_n^{\boldsymbol{T}}, v_m^{\boldsymbol{T}}, v_\omega^{\boldsymbol{T}}]^{\boldsymbol{T}}$
4	$[v_n^{\boldsymbol{T}}, v_\omega^{\boldsymbol{T}}]^{\boldsymbol{T}}$
5	$[v_b^{\boldsymbol{T}}, v_m^{\boldsymbol{T}}, v_\omega^{\boldsymbol{T}}]^{\boldsymbol{T}}$
6	$[v_b^{\boldsymbol{T}}, v_\omega^{\boldsymbol{T}}]^{\boldsymbol{T}}$
7	$[v_m, v_\omega^{\boldsymbol{T}}]^{\boldsymbol{T}}$
8	N/A
9	$[v_n^{\boldsymbol{T}}, v_b^{\boldsymbol{T}}, v_m, v_\omega^{\boldsymbol{T}}]^{\boldsymbol{T}}$
10	$[v_n^{\boldsymbol{T}}, v_b^{\boldsymbol{T}}, v_\omega^{\boldsymbol{T}}]^{\boldsymbol{T}}$

5.2.3 Measurement Model

The measurement model relates the measurement value $\boldsymbol{z}_k$ to the state vector $\boldsymbol{x}_k$ through a nonlinear measurement function

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k, \boldsymbol{v}_k) = h\left(\begin{bmatrix}\boldsymbol{q}\\ \boldsymbol{\omega_M}\end{bmatrix}_k, \boldsymbol{v}_k\right) = \left(\begin{bmatrix}\boldsymbol{y(q)}\\ \boldsymbol{\omega_M}\end{bmatrix}_k + \boldsymbol{v}_k\right) \tag{72}$$

where $\boldsymbol{v}$ is the combination of the motion capture camera measurement noise $\boldsymbol{v_m}$, the gyroscope measurement noise $\boldsymbol{v_\omega}$, the accelerometer measurement noise $\boldsymbol{v_n}$, and the magnetometer measurement noise $\boldsymbol{v_b}$, depending on different sensor fusion scenario as shown in Table 5. The nonlinear quaternion rotation function $\boldsymbol{y}(\boldsymbol{q})$ also varies in sensor selection situations as defined previously in Table 3.

5.2.4 Linearization of the Nonlinear Models

Since it necessitate linear model to employ the standard Kalman filter loops, the nonlinear process model $f(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{w}_k)$ and measurement model $h(\boldsymbol{x}_k, \boldsymbol{v}_k)$ have to be linearized. A straight-forward way to achieve this goal is to adopt a first order Taylor expansion of the nonlinear models to provide local approximation within the current state as below

$$F_k = \frac{\partial f}{\partial x}|_{\hat{\boldsymbol{x}}_{k-1}} \tag{73}$$

$$H_k = \frac{\partial h}{\partial x}|_{\hat{\boldsymbol{x}}_k^-} \tag{74}$$

where F_k is the Jacobi matrix of $f(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{w}_k)$ evaluated at $\hat{\boldsymbol{x}}_{k-1}$

$$F_{ij} = \frac{\partial f}{\partial x}|_{\hat{\boldsymbol{x}}_{k-1}} \tag{75}$$

and H_k is the Jacobi matrix of $h(\boldsymbol{x}_k, \boldsymbol{v}_k)$ evaluated at $\hat{\boldsymbol{x}}_k^-$

$$H_{ij} = \frac{\partial h}{\partial x}|_{\hat{\boldsymbol{x}}_k^-} \tag{76}$$

The calculation of the Jacobi matrices H_k is similar to the derivation of $\boldsymbol{X}$ in section 5.1.2. H_k in different scenarios are listed in Table 6, where $\boldsymbol{J_n}, \boldsymbol{J_b}, \boldsymbol{J_{m1}}$ and $\boldsymbol{J_{m2}}$ are the Jacobi matrices for quaternion rotation, as defined in Table 4 . The calculation of F_k is expanded in Appendix C. Substitute F_k and H_k into equation (63) – (67), then the traditional Kalman filter procedure will work for the nonlinear system. Since the unit quaternion constraint needs to be maintained, the quaternion part of the state vector is normalized after equation (63) and equation (66).

Table 6. The Jacobi Matrices H_k for Different Scenarios

Scenarios	H_k
1	$[\boldsymbol{J_n}^T, \boldsymbol{J_b}^T, \boldsymbol{J_{m1}}^T, \boldsymbol{J_{m2}}^T, \boldsymbol{I}]^T$
2	$[\boldsymbol{J_n}^T, \boldsymbol{J_b}^T, \boldsymbol{I}]^T$
3	$[\boldsymbol{J_n}^T, \boldsymbol{J_{m1}}^T, \boldsymbol{J_{m2}}^T, \boldsymbol{I}]^T$
4	$[\boldsymbol{J_n}^T, \boldsymbol{I}]^T$
5	$[\boldsymbol{J_b}^T, \boldsymbol{J_{m1}}^T, \boldsymbol{J_{m2}}^T, \boldsymbol{I}]^T$
6	$[\boldsymbol{J_b}^T, \boldsymbol{I}]^T$
7	$[\boldsymbol{J_{m1}}^T, \boldsymbol{J_{m2}}^T, \boldsymbol{I}]^T$
8	N/A
9	$[\boldsymbol{J_n}^T, \boldsymbol{J_b}^T, \boldsymbol{J_{m1}}^T, \boldsymbol{J_{m2}}^T]^T$
10	$[\boldsymbol{J_n}^T, \boldsymbol{J_b}^T]^T$

When only one reference vector is available, similar to the approach taken for the quaternion-based complementary filter: the rotation about the direction of the reference vector is excluded in the update step according to the method clarified in section 5.1.4.

5.3 Quaternion-based Unscented Kalman Filter

When the system defined by the process models and measurement models are highly nonlinear, the extended Kalman filter may give poor performance. The Unscented Kalman Filter (UKF) employs unscented transform to select a set of points called sigma points around the mean of the estimate. The mean and covariance of the estimate can later be reconstructed by propagating these sigma points through the nonlinear functions. Such measure allows a better performance than a standard EKF since it approximates the nonlinearity more precisely.

5.3.1 System Modeling

The quaternion-based UKF for orientation tracking share the same state vectors and measurement model with the EKF described above. Note that since the orientation estimate $\boldsymbol{q}$ is a unit quaternion, its four elements are no longer independent with each other. Such constraint leads to a different strategy of sampling and treating noise in the UKF.

The UKF add disturbance to the orientation estimate $\boldsymbol{q}$ through quaternion multiplication. The process noise of the quaternion $\boldsymbol{w_q}$ is a 3-element vector, which can be regarded as a rotation vector to represent a random rotation with the angle

$$\theta_{\boldsymbol{w}} = |\boldsymbol{w_q}| \tag{77}$$

and the unit rotation axis

$$a_{\boldsymbol{w}} = \frac{\boldsymbol{w_q}}{|\boldsymbol{w_q}|} \tag{78}$$

The quaternion representation of this disturbance rotation is

$$\boldsymbol{q_w} = \begin{bmatrix} \cos\left(\frac{\theta_{\boldsymbol{w}}}{2}\right) \\ a_{\boldsymbol{w}} \cos\left(\frac{\theta_{\boldsymbol{w}}}{2}\right) \end{bmatrix} \tag{79}$$

The process noise is applied to the original quaternion component $\boldsymbol{q}$ through quaternion multiplication. When no gyro data is involved

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} \otimes \boldsymbol{q_w} = \boldsymbol{q}_{k-1} \otimes \boldsymbol{q_w} \tag{80}$$

This method of dealing quaternion noise leads to the change of the size of the block of the process noise covariance Q that is corresponding to the quaternion noise. Since the process noise of the quaternion $\boldsymbol{w_q}$ is interpreted as a 3-element rotation vector, its covariance Q becomes a 3 by 3 matrix, unlike the 4 by 4 matrix adopted in the EKF. Accordingly, the covariance matrix P measures the accuracy of the estimated orientation in the rotation vector form, and agrees with the size of Q.

When gyro data contributes to the orientation prediction

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{w}_k) = f\left(\begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{\omega_M} \end{bmatrix}_{k-1}, \begin{bmatrix} \boldsymbol{w_q} \\ \boldsymbol{w_\omega} \end{bmatrix}_k\right) = \begin{bmatrix} d(\boldsymbol{q}_{k-1}, \boldsymbol{\omega}_{\boldsymbol{M}_{k-1}}) \otimes \boldsymbol{q}_{\boldsymbol{w}_k} \\ \boldsymbol{\omega}_{\boldsymbol{M}_{k-1}} + \boldsymbol{w}_{\boldsymbol{\omega}_k} \end{bmatrix} \tag{81}$$

where $d(\boldsymbol{q}_{k-1}, \boldsymbol{\omega}_{\boldsymbol{M}_{k-1}})$ is the angular velocity integration function defined by equation (30) in section 5.1.1. Consequently, the process noise covariance Q and the estimate

covariance *P* are expanded to be 6 by 6 matrices. Note it is also valid to introduce the process noise before integrating the angular velocity, which presents

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{w}_k) = \begin{bmatrix} d(\boldsymbol{q}_{k-1} \otimes \boldsymbol{q}_{\boldsymbol{w}_k}, \boldsymbol{\omega}_{\boldsymbol{M}_{k-1}}) \\ \boldsymbol{\omega}_{\boldsymbol{M}_{k-1}} + \boldsymbol{w}_{\omega_k} \end{bmatrix} \tag{82}$$

Since the model defined by (82) facilitates the generation of the sigma points, it is employed by the quaternion-based UKF.

5.3.2 Sigma Points Generation

The filtering procedure of the quaternion-based UKF contains three major steps. Prior to the prediction step and the update step, the sigma points are selected in a proper way so that they can recover the mean and covariance of the state estimate. Provided the previous state estimate $\widehat{\boldsymbol{x}}_{k-1}$and the N by N estimate covariance matrix P_{k-1}, the loop of unscented Kalman filter starts with generating a set of sigma points $\{\boldsymbol{\chi}_i\}$ $(i = 1, \dots, 2N)$ near $\widehat{\boldsymbol{x}}_{k-1}$. This sampling process resembles the way of introducing noise to the state vector described previously in the process model part, because the sigma points can be interpreted as a set of disturbed state vectors formed by imposing known disturbances on the original state vector. Let $\boldsymbol{\Upsilon}_i$ be the disturbance vectors

$$\boldsymbol{\Upsilon}_i = (\sqrt{2N}\sqrt{P_{k-1}})_i = \boldsymbol{w}_{\boldsymbol{\Upsilon}_i} \tag{83}$$

where $(\sqrt{P_{k-1}})_i$ is the i-th (i = 1, ⋯, N) column of the matrix resulted from the Cholesky decomposition of P_{k-1}. When the gyroscope data is not involved, P_{k-1} is a 3 by 3 matrix. Thus $\boldsymbol{w}_{\boldsymbol{\Upsilon}_i}$ is a 3-element rotation vector to represent the rotation disturbance. Using the method described by equation (77) – (79), the rotation vector $\boldsymbol{w}_{\boldsymbol{\Upsilon}_i}$ can be converted to its quaternion form $\boldsymbol{q}_{\boldsymbol{\Upsilon}_i}$. Then the 2N sigma points are created by

$$\begin{cases} \boldsymbol{\chi}_i = \widehat{\boldsymbol{x}}_{k-1} \otimes \boldsymbol{q}_{\boldsymbol{\Upsilon}_i} = \widehat{\boldsymbol{q}}_{k-1} \otimes \boldsymbol{q}_{\boldsymbol{\Upsilon}_i} & (\mathrm{i} = 1, \dots, \mathrm{N}) \\ \boldsymbol{\chi}_i = \widehat{\boldsymbol{x}}_{k-1} \otimes \boldsymbol{q}_{\boldsymbol{\Upsilon}_i}{}^{-1} = \widehat{\boldsymbol{q}}_{k-1} \otimes \boldsymbol{q}_{\boldsymbol{\Upsilon}_i}{}^{-1} & (\mathrm{i} = \mathrm{N}+1, \dots, 2\mathrm{N}) \end{cases} \tag{84}$$

In view of the fact that the procedure of dealing with the process noise (with 3 by 3 covariance Q) and that for producing sigma points are very similar, the process noise can be included in the sigma points through

$$\boldsymbol{\Upsilon}_i = (\sqrt{2N}\sqrt{P_{k-1} + Q})_i \tag{85}$$

When the angular velocity is part of the state vector, P_{k-1} and Q are 6 by 6 matrices. As described in the process model defined by equation (83), the process noise is applied to the state vector in advance of the prediction with gyroscope data. Thus it is still valid to derive $\boldsymbol{\Upsilon}_i$ from $P_{k-1} + Q$.

$$\boldsymbol{\Upsilon}_i = (\sqrt{2N}\sqrt{P_{k-1} + Q})_i = \begin{bmatrix} \boldsymbol{w}_{\boldsymbol{\Upsilon}_i} \\ \boldsymbol{\omega}_{\boldsymbol{\Upsilon}_i} \end{bmatrix} \tag{86}$$

where $\boldsymbol{\omega}_{\boldsymbol{\Upsilon}_i}$ is the disturbance to the angular velocity estimate that contains 3 components. Then the 2N sigma points are created by

$$\begin{cases} \boldsymbol{\chi}_i = \begin{bmatrix} \widehat{\boldsymbol{q}}_{k-1} \otimes \boldsymbol{q}_{\boldsymbol{\Upsilon}_i} \\ \widehat{\boldsymbol{\omega}}_{\boldsymbol{M}_{k-1}} + \boldsymbol{\omega}_{\boldsymbol{\Upsilon}_i} \end{bmatrix} & (i = 1, \dots, N) \\ \boldsymbol{\chi}_i = \begin{bmatrix} \widehat{\boldsymbol{q}}_{k-1} \otimes \boldsymbol{q}_{\boldsymbol{\Upsilon}_i}{}^{-1} \\ \widehat{\boldsymbol{\omega}}_{\boldsymbol{M}_{k-1}} - \boldsymbol{\omega}_{\boldsymbol{\Upsilon}_i} \end{bmatrix} & (i = N+1, \dots, 2N) \end{cases} \tag{87}$$

5.3.3 Prediction of the Quaternion-based Unscented Kalman Filter

Within the prediction step, the nonlinear state transition function will take these sigma points $\{\boldsymbol{\chi}_i\}$ as the input to produce a new set of predicted sigma points $\{\boldsymbol{\chi}_i^-\}$ $(i = 1, \dots, 2N)$. Firstly consider the case without involving gyroscope: the output of the state transition function is a quaternion

$$\boldsymbol{\chi}_i^- = f(\boldsymbol{\chi}_i, \mathbf{0}) = \boldsymbol{q}_i \tag{88}$$

The mean and covariance of $\{\boldsymbol{\chi}_i^-\}$ respectively provides the priori state estimate $\widehat{\boldsymbol{x}}_k^-$ and priori covariance matrix P_k^-. The computation of the mean and covariance of quaternion vectors will be elaborated in this section.

The orientations represented by the unit quaternion vectors belong to a nonlinear manifold (a four dimensional unit sphere), instead of the linear vector space. Constrained by the nonlinearity and the fixed unit length, the mean and covariance of quaternion vectors cannot be computed in the ways designed for the vector space. Take an extreme case for example, the weighted sum of the quaternion vectors $[1, 0, 0, 0]^T$ and $[-1, 0, 0, 0]^T$ yields a zero vector, while as stated in section 4.1, they produce the same rotation.

The intrinsic gradient descent algorithm proposed by [33] provides a solution for computing the mean of quaternion orientations. This method aims to find a quaternion vector, the sum of whose errors with the members of the quaternion set $\{\boldsymbol{q}_i\}$ $(i = 1, \dots, 2N)$ reaches the minimum. Then this quaternion is a good estimate of the mean of this quaternion set $\{\boldsymbol{q}_i\}$. The error (rotation) between two quaternion vectors is quantified by a rotation vector $\boldsymbol{e}_i$, which can be summed in the vector space. Ideally the error sum equals zero when the true mean of the quaternion vector $\bar{\boldsymbol{q}}$ is located. The search of the quaternion mean is realized in a recursive manner. The quaternion mean estimate $\bar{\boldsymbol{q}}_n$ can be initialized with a random quaternion vector. During every loop, the errors between $\bar{\boldsymbol{q}}_n$ and each quaternion member $\boldsymbol{q}_i$, represented by a quaternion, is updated as

$$\boldsymbol{q}_{e_i} = \boldsymbol{q}_i \otimes \bar{\boldsymbol{q}}_n{}^{-1} \tag{89}$$

Following equation (77) – (79), $\boldsymbol{q}_{e_i}$ is translated to the rotation vector $\boldsymbol{e}_i$. Then, $\bar{\boldsymbol{e}}$, the mean of the error rotation vector set $\{\boldsymbol{e}_i\}$ is be calculated as

$$\bar{\boldsymbol{e}} = \frac{1}{2N}\sum_1^{2N} \boldsymbol{e}_i \tag{90}$$

Let $\boldsymbol{q}_{\bar{\boldsymbol{e}}}$ be the corresponding quaternion form of $\bar{\boldsymbol{e}}$. The estimate of the quaternion mean $\bar{\boldsymbol{q}}_n$ is adjusted for next round of iteration by

$$\bar{\boldsymbol{q}}_{n+1} = \boldsymbol{q}_{\bar{\boldsymbol{e}}} \otimes \bar{\boldsymbol{q}}_n \tag{91}$$

When the length of $\bar{\boldsymbol{e}}$ is a smaller than a pre-decided threshold, implying $\bar{\boldsymbol{q}}_n$ is close to the real quaternion mean $\bar{\boldsymbol{q}}$, the loop can be terminated with

$$\bar{\boldsymbol{q}} = \boldsymbol{E}[\boldsymbol{q}_i] = \bar{\boldsymbol{q}}_n \tag{92}$$

Since $\boldsymbol{E}[\boldsymbol{\chi}_i^-]$ presents the predicted priori state estimate $\hat{\boldsymbol{x}}_k^-$

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{E}[\boldsymbol{\chi}_i^-] = \boldsymbol{E}[\boldsymbol{q}_i] = \bar{\boldsymbol{q}}_n \tag{93}$$

Note at the end of the recursion the error rotation vector set $\{\boldsymbol{e}_i\}$ ends up to be the errors between the quaternion mean $\bar{\boldsymbol{q}}$ and the members of $\{\boldsymbol{q}_i\}$. According to the definition of the covariance, the covariance of the orientation $\{\boldsymbol{q}_i\}$ is

$$cov(\boldsymbol{q}_i) = \frac{1}{2N}\sum_1^{2N} \boldsymbol{e}_i \boldsymbol{e}_i{}^T \tag{94}$$

Because $\boldsymbol{e}_i$ are a rotation vector, the covariance matrix given by (94) is 3 by 3. Since $cov(\boldsymbol{\chi}_i^-)$ presents the predicted priori estimate covariance P_k^-

$$P_k^- = cov(\boldsymbol{\chi}_i^-) = cov(\boldsymbol{q}_i) = \frac{1}{2N}\sum_1^{2N} \boldsymbol{e}_i \boldsymbol{e}_i{}^T \tag{95}$$

Now consider the case involving the gyroscope data for prediction, each predicted sigma point $\boldsymbol{\chi}_i^-$ is extended to 7 elements

$$\boldsymbol{\chi}_i^- = f(\boldsymbol{\chi}_i, \boldsymbol{0}) = \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{\omega}_i \end{bmatrix} \tag{96}$$

The calculation of the mean of the angular velocity vectors $\boldsymbol{\omega}_i$ can follow the standard procedure for multivariate variables in the vector space.

$$\bar{\boldsymbol{\omega}} = \frac{1}{2N}\sum_1^{2N} \boldsymbol{\omega}_i \tag{97}$$

Consequently

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{E}[\boldsymbol{\chi}_i^-] = \begin{bmatrix} \bar{\boldsymbol{q}} \\ \bar{\boldsymbol{\omega}} \end{bmatrix} \tag{98}$$

$$P_k^- = cov(\boldsymbol{\chi}_i^-) = \frac{1}{2N}\sum_1^{2N} \begin{bmatrix} \boldsymbol{e}_i \\ \boldsymbol{\omega}_i - \bar{\boldsymbol{\omega}} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_i \\ \boldsymbol{\omega}_i - \bar{\boldsymbol{\omega}} \end{bmatrix}^T \tag{99}$$

5.3.4 Update of the Quaternion-based Unscented Kalman Filter

The update step calculates the Kalman gain K_k to indicate the extent of adjustment that needs to be made based on the difference between the real observation and that derived from the state vector. The update begins with propagating the predicted sigma points $\{\boldsymbol{\chi_i^-}\}$ through the nonlinear measurement function to produce a new set of measurement sigma points $\{\boldsymbol{Z_i^-}\}$ $(i = 1, \dots, 2N)$.

$$\boldsymbol{Z_i^-} = h(\boldsymbol{\chi_i^-}, \mathbf{0}) \tag{100}$$

Then the predicted measurement $\boldsymbol{z_k^-}$ is defined as the mean of the measurement sigma points $\{\boldsymbol{Z_i^-}\}$

$$\boldsymbol{z_k^-} = \boldsymbol{E}[\boldsymbol{Z_i^-}] = \frac{1}{2N}\sum_1^{2N} \boldsymbol{Z_i^-} \tag{101}$$

And the predicted measurement covariance P_{zz} is defined as the covariance of $\{\boldsymbol{Z_i^-}\}$

$$P_{zz} = cov(\boldsymbol{Z_i^-}) = \frac{1}{2N}\sum_1^{2N}[\boldsymbol{Z_i^-} - \boldsymbol{z_k^-}][\boldsymbol{Z_i^-} - \boldsymbol{z_k^-}]^T \tag{102}$$

Depending on whether the angular velocity $\boldsymbol{\omega}_i$ is included in the predicted sigma points $\boldsymbol{\chi_i^-}$, the state-measurement cross-covariance matrix is defined as

$$P_{xz} = \frac{1}{2N}\sum_1^{2N} \boldsymbol{e}_i[\boldsymbol{Z_i^-} - \boldsymbol{z_k^-}]^T \tag{103}$$

or

$$P_{xz} = \frac{1}{2N}\sum_1^{2N} \begin{bmatrix} \boldsymbol{e}_i \\ \boldsymbol{\omega}_i - \overline{\boldsymbol{\omega}} \end{bmatrix} [\boldsymbol{Z_i^-} - \boldsymbol{z_k^-}]^T \tag{104}$$

where $\boldsymbol{e}_i$ is the by-product of the recursive quaternion mean calculation – the differences between the quaternion mean $\overline{\boldsymbol{q}}$ and the members of $\{\boldsymbol{q}_i\}$ contained by the predicted sigma points $\{\boldsymbol{\chi_i^-}\}$, as described in the previous section.

Provided the covariance matrices defined by equation (102) – (104), the Kalman gain K_k can be calculated as

$$K_k = P_{xz}(P_{zz} + R)^{-1} \tag{105}$$

Eventually the updated state vector $\hat{\boldsymbol{x}}_k$ and estimate covariance P_k can be obtained through the following equations

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k-1} + K_k(\boldsymbol{z}_k - \boldsymbol{z}_k^-) \tag{106}$$

$$P_k = (I - K_k H_k)P_k^- \tag{107}$$

When only one reference vector is available, similar to the approach taken for the quaternion-based complementary filter: the rotation about the direction of the reference vector is excluded in the update step according to the method clarified in section 5.1.4.

5.4 Quaternion-based Particle Filter

The particle filters (PF) approximates the posterior probability distribution function (pdf) of the state vector $\boldsymbol{x}_k$ given all available observations $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_k\}$. A set of samples $\{\boldsymbol{x}_k{}^{(i)}\}$ termed as the particles with associated weights $\{W_k{}^{(i)}\}$ $(i = 1, \dots, L)$ are used to characterize this required pdf. The particles and their weights are updated sequentially along with the state evolution when new observations become available. When the particle number *L* is sufficiently large, the particles propagated through the nonlinear system can precisely capture the posterior mean and the posterior covariance for any nonlinearity. Such manner allows the particle filters to approach the Bayesian optimal estimate, so they can potentially outperform the EKF or UKF.

5.4.1 System Modeling

The state vector $\boldsymbol{x}_k$, process model $f(\boldsymbol{x}_{k-1}, \boldsymbol{w}_k)$ and measurement model $h(\boldsymbol{x}_k, \boldsymbol{v}_k)$ of the particle filter are identical with those of the EKF. Since in the system described by

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{w}_k) \tag{108}$$

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k, \boldsymbol{v}_k) \tag{109}$$

the noise vectors $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are both Gaussian, the conditional pdf of $\boldsymbol{x}_k$ and $\boldsymbol{z}_k$ comply with the normal distribution as

$$\boldsymbol{p}(\boldsymbol{x}_k|\,\boldsymbol{x}_{k-1}) = N(\,f(\,\boldsymbol{x}_{k-1}\,,\,\mathbf{0}), Q) \tag{110}$$

$$\boldsymbol{p}(\boldsymbol{z}_k|\,\boldsymbol{x}_k) = N(\,h(\,\boldsymbol{x}_k\,,\mathbf{0}), R) \tag{111}$$

5.4.2 Particle Propagation

The chosen particle filter for the quaternion-based orientation tracking is the sequential importance resampling (SIR, also named as the bootstrap filter) algorithm proposed by N. J. Gordon etc. During every iteration of the SIR, a set of particles $\{\boldsymbol{x}_k{}^{(i)}\}(i = 1, \dots, L)$ are drawn from the predicted state estimate $\widehat{\boldsymbol{x}}_k^-$ according to the Gaussian distribution (110). The weights $\{W_k{}^{(i)}\}(i = 1, \dots, L)$ (importance) of these renewed particles are evaluated by $\boldsymbol{p}(\boldsymbol{z}_k|\boldsymbol{x}_k{}^{(i)})$ given by equation (111). Besides updating the particles and their weights, resampling is executed at the end of every cycle to pick up the particles with larger weights. The selected particles end up with equal weights and their mean provides the state estimate $\widehat{\boldsymbol{x}}_k$.

The prediction step generates the predicted state estimate $\widehat{\boldsymbol{x}}_k^-$ according to equation (109). The particles $\{\boldsymbol{x}_k{}^{(i)}\}\,(i = 1, \dots, L)$ scattered around $\widehat{\boldsymbol{x}}_k^-$ follows the Gaussian distribution

$$\boldsymbol{p}(\boldsymbol{x}_k{}^{(i)}|\widehat{\boldsymbol{x}}_k^-) = N(\,\widehat{\boldsymbol{x}}_k^-, Q) \tag{112}$$

Thus the particle set $\{\boldsymbol{x}_k{}^{(i)}\}\,(i = 1, \dots, L)$ can be viewed as L predicted state estimate vectors $\widehat{\boldsymbol{x}}_k^-$ perturbed by the zero-mean Gaussian noise set $\{\,\boldsymbol{w}_k{}^{(i)}\}$ $(i = 1, \dots, L)$ with covariance Q. The method of introducing noise into the state vector is discussed section 5.3.1 when presenting the process model of the UKF.

The weight $W_k{}^{(i)}$ of the drawn particle $\boldsymbol{x}_k{}^{(i)}$ is proportional to the product of the weight of $\boldsymbol{x}_{k-1}{}^{(i)}$ and the conditional pdf of the current observation $\boldsymbol{z}_k$

$$W_k^{(i)} \propto W_{k-1}^{(i)} \boldsymbol{p}(\boldsymbol{z}_k|\boldsymbol{x}_k^{(i)}) = W_{k-1}^{(i)} N(h(\boldsymbol{x}_k^{(i)}, \mathbf{0}), R) \quad (113)$$

Considering that the weights needs to be normalized, $W_k^{(i)}$ is given by

$$\widehat{W}_k^{(i)} = W_{k-1}^{(i)} \boldsymbol{p}(\boldsymbol{z}_k|\boldsymbol{x}_k^{(i)}) = W_{k-1}^{(i)} N(h(\boldsymbol{x}_k^{(i)}, \mathbf{0}), R) \quad (114)$$

$$W_k^{(i)} = \widehat{W}_k^{(i)} / \sum_{i=1}^{L} \widehat{W}_k^{(i)} \quad (115)$$

The major bulk of the resulted weights may fall on only a handful of or even just one particle. In such scenario, the weights of most particles are close to zero. Consequently these particles can hardly contribute to the state estimate. Thus this phenomenon, termed as the particle degeneracy, is highly undesired. The particle degeneracy can be overcome through particle resampling. Particle resampling screens particles with negligible weights and replace them with particles with large weights, which will possess the same weights $W_k^{(i)} = 1/L$. The resampling schemes can be implemented in every cycle, like in the original SIR, or when the degeneracy phenomenon has been detected. In our experiments, the stratified resampling method (proposed by Kitagawa 1996) once the weight set $\{W_k^{(i)}\}$ $(i = 1, \dots, L)$ has been update. Note that, since at the end of the loop the resampled particles have the same weights, equation (114) can be simplified to

$$\widehat{W}_k^{(i)} = \boldsymbol{p}(\boldsymbol{z}_k|\boldsymbol{x}_k^{(i)}) = N(h(\boldsymbol{x}_k^{(i)}, \mathbf{0}), R) \quad (116)$$

The updated state estimate $\widehat{\boldsymbol{x}}_k$ is given by the mean of the resampled particles as

$$\widehat{\boldsymbol{x}}_k = \boldsymbol{E}[\boldsymbol{x}_k^{(i)}] = \frac{1}{L} \sum_1^L \boldsymbol{x}_k^{(i)} \quad (117)$$

When only one reference vector is available, similar to the approach taken for the quaternion-based complementary filter: the rotation about the direction of the reference vector is excluded in the update step. These can be achieved by excluding the rotation about the direction of the reference vector when generating the particle set $\{\boldsymbol{x}_k^{(i)}\}$ $(i = 1, \dots, L)$ according to the method clarified in section 5.1.4. This promises that the

particles always share the same rotation around the reference vector, which is only watched by the gyroscope.

6. IMPLEMENTATION ISSUES

6.1 Data Smoothing

Smoothing the IMU data is often necessary to reduce observation noise. In our research, we have conducted a comparative study to examine the performance of the moving average (MA) filter and the 2nd order Savitzky-Golay (SG) filter. Let $u(n)$ be the original data to be smoothed. The MA filter is defined as

$$v(n) = \frac{1}{S}\sum_{i=1}^{S} u(n+i) \tag{118}$$

where S is the size of the moving window and $v(n)$ is the smoothed data. The SG filter is defined as

$$v(n) = \sum_{i=-n_L}^{n_R} c_i u(n+i) \tag{119}$$

where n_L represents the number of data points used "to the left" of the data point n, or, coming before the data point n; n_R is the number of points "to the right" of it, or, coming after that data point; c_i's are the SG coefficients pre-computed according to the filter order, n_L and n_R.

Table 7. Performance Comparison of the SG Filter and the MA Filter

Low-pass filter parameters		STD (°) in the static case				STD (°) and latency in the dynamic movement case				
		Pitch	Yaw	Roll	Sum	Pitch	Yaw	Roll	Sum	Latency (ms)
SG	nL = 38, nR = 10	0.105	0.738	0.065	0.908	0.895	1.693	0.619	3.207	100
	nL = 43, nR = 5	0.137	0.970	0.090	1.198	0.871	1.957	0.670	3.498	50
	nL = 45, nR = 3	0.161	1.137	0.105	1.403	1.002	2.322	0.734	4.058	30
MA	S = 20	0.115	0.805	0.072	0.993	0. 947	1.786	0.651	3.384	100
	S = 10	0.158	1.222	0.105	1.485	0.996	2.304	0.722	4.022	50

In our research, we have conducted systematic study to evaluate and compare these two data smoothing approaches. The performance of the smoothing methods was analyzed based on the orientation tracking results obtained from the smoothed IMU data. Data collected from both static IMU and moving IMU has been used. As shown in Figure 9, in the dynamic case the IMU sensor board was manually rotated by consecutively cycling the sensor through the angles of pitch, yaw and roll between –30° and 30°. Table 7 lists the standard deviations of the tracking results for both static and dynamic cases using the two data smoothing methods.

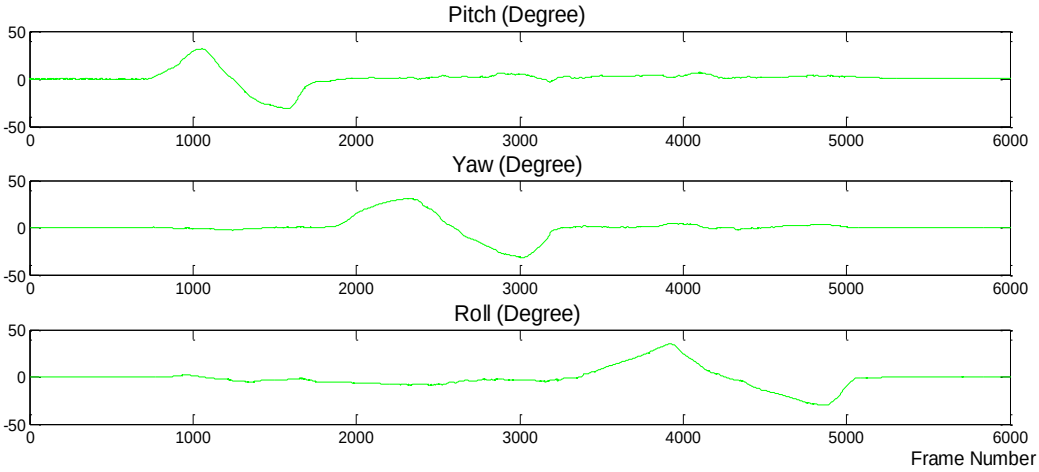

Fig. 9. Motion Composed of Consecutive Cycling the Sensor through Angles of Pitch, Yaw and Roll between -30° and 30°.

For the dynamic case, the latency was also obtained by reading the peak position of the correlation between the results from the IMU and that from the motion capture system. Let $h(n)$ and $g(n)$, n=0,1,...,N–1, be the tracking results obtained using the IMU data and that from the motion capture data, respectively. Their discrete correlation is given by

$$Corr_{g,h}(j) = \sum_{k=0}^{N-1} \tilde{g}_{j+k}\, h_k \qquad j=-\mathrm{N}+1,\ldots,\mathrm{N}\text{-}1 \quad (120)$$

where $\tilde{g}(\cdot)$ is the periodic signal by repeating $g(\cdot)$ with a period of N. If the highest value appears at frame p, it indicates that due to the smoothing technique, $h(n)$ lags behind p frames comparing to $g(n)$. For a sample rate of 100Hz, the latency is p×10ms.

As shown in Table 4, when using the MA filter, the latency is about half of S, the moving average window length. When using the SG filter, the latency is around the length of n_R. The maximum overall latency allowed by MRR system is 30ms, which implies that the S must be smaller than 6 frames/60ms for the MA filter or n_R must be lower than 3 frames/30ms for the SG filter. As shown in Table 4, when n_R = 3 frames, the SG filter reached the same accuracy of the MA filter with S=10 frames and in this case, the latency of the SG filter is 30% lower than that of the MA filter. Furthermore, given comparable latency, for instance, when SG filter with n_R= 5 frames and MA filter with S = 10 frames, the SG filter leads to more accurate tracking than the MA filter. These experimental results indicate that in our application of orientation tracking, the SG filter provides superior accuracy to the MA filter when mitigating the latency to the same level. Thus the SG filter has been used to smooth IMU data in our experiments with n_L = 45, n_R = 3 to meet the system requirements.

6.2 Estimating the Z Axis Reading of the Magnetometer

Since magnetometer integrated in the Sparkfun 6DOF v3 IMU only provides measurements on the X-Y plane, the magnetic reading in the Z axis needs to be estimated to provide three dimensional measurements as the input of the filters. In our research, we have developed a robust procedure to secure the estimation of the Z component of the magnetic vector.

The estimation process composes of two stages. First, from given the magnitude of the local magnetic field and the observed X and Y components, the length of Z component can be calculated. Then the sign of the Z component needs to be found. Determining the sign of the Z component is challenging, especially when its length is close to zero. To obtain reliable estimate for the sign of the Z component, in our research, we have placed the IMU in a specific configuration so that the Z component is always

positive in the range of torso motion. This is achieved by properly adjusting the orientation of the IMU on the back-marker-board of the subject so that at the initial neutral position of the subject the local magnetic field is aligned with the Z axis of $\boldsymbol{C_I}$. In other word, Z component of the magnetic vector reaches its maximum at the initial position (Fig. 10 (a)). Consequently, within the range of motion (usually between –30° and 30°), there will always be a large positive portion of the local magnetic field projected onto the Z axis of $\boldsymbol{C_I}$ (Fig. 10 (b)). Considering the local magnetic field may vary in different rehabilitation environments due to the change of the facing direction of the patient and the distribution of ferromagnetic objects, a hinged connector is made to connect the IMU and the marker board and to make the orientation of the IMU easily adjustable for various system configurations.

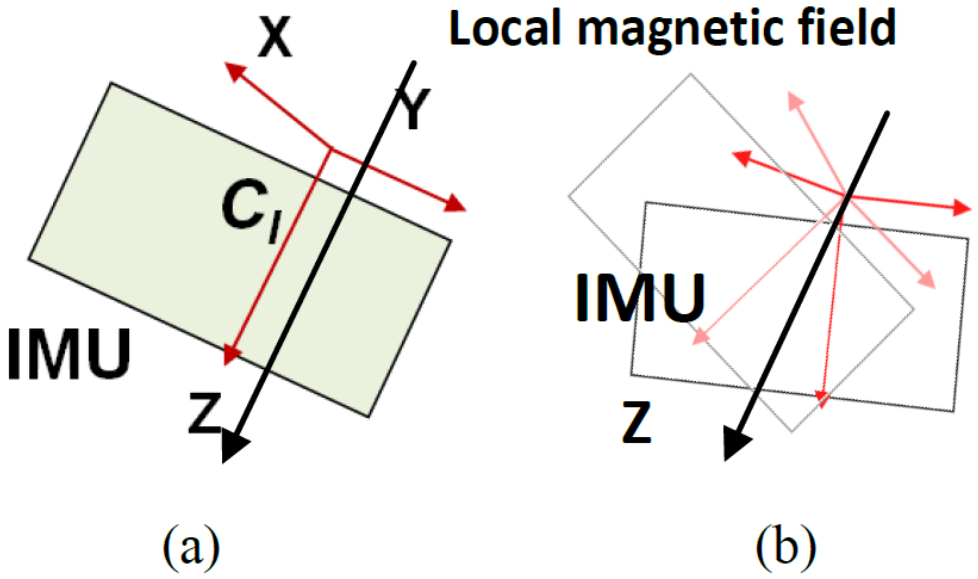

Fig. 10. (a) At the initial position, the local magnetic field is paralleled with Z-axis of the IMU's coordinate system. (b) During the orientation tracking, the angles between the Z-axis and the local magnetic field are smaller than 90 degree. Thus the the local magnetic field is always positive.

7. EXPERIMENTAL RESULTS AND ANALYSIS

Experiments have been conducted to evaluate and compare the tracking performance of the 10 typical sensing scenarios using the complementary filter, the extended Kalman filter (EKF), the unscented Kalman filter (UKF), the particle filter (PF). All the sensing scenarios and filtering algorithms have been tested using the same set of benchmark data. In our experiments, we have recorded data containing random rotations composed of repeatedly cycling the sensor through various angles of pitch yaw, and roll performed on the inertial sensor mounted on the marker board. Figure 11 shows a typical movement trial. A total of 10 trials of such random motion have been recorded as the benchmark testing dataset. The accuracy of all filtering algorithms in each sensor fusion scenario has been evaluated using the tracking results derived from the motion capture system as the reference in view of its high accuracy.

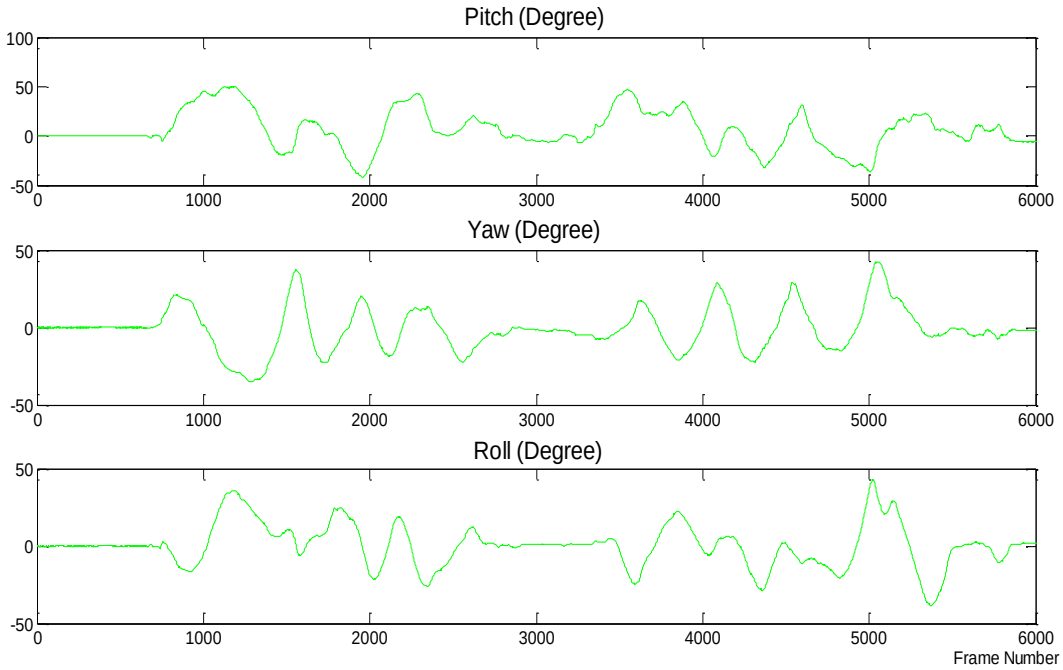

Fig. 11. An Exemplar Random Motion Trial

Out of the 10 sensing scenarios, seven of them, except Scenarios 4, 6, and 8, fall into the case of updating with two reference vectors. Thus these seven scenarios are able to

provide drifting-free measurements. The tracking results, including the averaged accuracy, measured by the root mean square difference (RMSD), and the latency, are shown in Table 8. Table 8 (b) – (d) gives the results from the proposed complementary filter framework, the extended Kalman filter (EKF), the unscented Kalman filter (UKF) and the particle filter (PF) respectively.

Generally, a number of observations can be made based on these results. First of all, when the motion capture system is functional (Scenarios 1, 3, 5, 7, and 9), the tracking results from shown in Table 8 (a) – (d) are all close to the reference data using only the motion capture system, implying all four sensor fusion methods can present precise orientation tracking with motion capture data. It is mainly because of the higher weights, or smaller observation matrix, applied to the motion capture data in the fusion frameworks in light of its high tracking accuracy. Figure 12 – 15 shows the RMSDs of the four methods for the scenarios when the motion capture data is available (Scenarios 1, 3, 5, 7, and 9) from one testing trial.

Then, the latency of the four filters shown by Table 8 (a) – (d) are comparable with each other. Thus the processing time of the filter algorithms does not affect the lag a lot. The latency mainly relies on specific smoothing techniques and sensor selection. When the motion capture data are present, since no smoothing measures are applied to the vision data, no lag is observed comparing to the reference, which is solely from the motion capture data. Without the motion capture data, smoothing the IMU readings causes a lag around 26ms for all four filtering algorithms.

When it comes to the accuracy of the four chosen filters, for odd scenarios in Table 8 (scenario 1, 3, 5, 7 and 9, these scenarios involve motion capture data), the RMSD of proposed complementary filter is smaller than that from the Kalman filters and the particle filter. This makes sense considering that the Gauss-Newton method applied in the

complementary filter is a least square algorithm that aims to minimize the square error between the fitted parameters and the observation in solving every single iteration. Since when the majority of the weights are assigned to the motion capture data, the Gauss-Newton iteration tends to give results mainly depending on the recently available motion capture data. Simultaneously, the motion capture data is used to generate references. This makes the complementary filter agree with the reference very well for every single frame. Instead of pursuing a minimum least square error for every frame, the Bayesian approaches (i.e. the Kalman filters and the particle filter) aim to minimize the mean square error over time by constructing the probability density function (PDF) of the state vectors. Though the KFs and the PFs are optimal or suboptimal estimators in the sense that they yields minimum mean square error (MMSE), their estimation might not follow the reference (tracking results derived from the motion capture observation, without considering noise) as closely as the least square methods. For example, the least square filter can respond to peak-shaped noise in observation, and make corresponding changes in final results instantly, while the Bayesian approaches might ignore the peak in observation and result in smoothed results. Among the Bayesian methods, the UKF outperformed the EKF as expected, since the former better approximate the nonlinearity. Somehow, the particle filter, which is usually considered a better estimator than the Kalman filters, provides the largest RMSD when motion capture data is involved. A possible explanation is that the particle number (2000) might be insufficient to precisely approximate the nonlinear model. Since sampling method is like a random search, its accuracy heavily depends on the sample volume. Experiments do show an improvement of the PF when the particle number increases.

When the motion capture data is unavailable, the complementary filter, the Kalman filters and the particle filter still produces long-lasting valid angles when two reference

vector are used for orientation tracking (i.e., both the inertial and magnetic field vectors are available) such as in Scenarios 2 and 10. The key difference between Scenarios 2 and 10 is that the former consists of all three IMU sensors and the latter only uses accelerometer and magnetometer without the gyro. By comparing the performances of these two scenarios as shown by Table 8 (a) – (d), it can be seen that adding the gyro only slightly improves the tracking accuracy and almost makes no impact on the latency.

The rank of the RMSDs of the four filtering algorithms for fusion within the IMU (Scenario 2 and 10) varies from the odd scenarios. The particle filter clearly outperforms the complementary filter and the Kalman filters, which demonstrates the advantage of the particle filter when handling noisy IMU data. The complementary filter provides the second smallest RMSD, closely followed by the UKF. The EKF presents the worst result. Theoretically, the UKF and EKF should be superior tracking solutions to the complementary filter based on the least square method. Somehow, the Bayesian approaches desire a proper system model to fully unleash their potential. In our methods Gaussian model are imposed on both the process model and the observation model, which may not be the exact case for our tracking task. This might undermine the performance of the Bayesian methods.

In our experiments, we have observed drifting from the gyro (i.e., tracking using only gyro in Scenario 8) or when only one reference vector is used for tracking (e.g., Scenarios 4 and 6). Table 9 lists the average drifting rates for these three scenarios over 20 seconds. Figure 16 – 19 shows the RMSDs for Scenarios 2, 4, 6 8, and 10 using the same testing trial data as Figure 12 – 15. From Figure 16 – 19, it can be seen that in scenario 2 and 10, the complementary filter, the EKF, the UKF and the PF all succeed in correcting the drifting with 2 reference vectors.

Table 8. Tracking Performance Comparison (without Drifting)

(a) The Complementary Filter

Sensor Fusion Options	Root Mean Square Difference (°)			Latency (ms)
	Pitch	Yaw	Roll	
1	2.4514×10^{-2}	1.3850×10^{-2}	0.1532	0
2	1.7683	3.4194	1.7398	27
3	1.1889×10^{-2}	1.3202×10^{-2}	0.1622	0
5	1.9048×10^{-2}	3.0501×10^{-2}	0.1815	0
7	5.873×10^{-3}	2.671×10^{-3}	0.1617	0
9	2.6149×10^{-2}	2.1346×10^{-2}	0.1828	0
10	1.8417	3.4946	1.7864	27

(b) EKF

Sensor Fusion Options	Root Mean Square Difference (°)			Latency (ms)
	Pitch	Yaw	Roll	
1	0.5042	0.6282	0.4234	0
2	2.2509	4.0259	2.4480	26
3	0.5042	0.6282	0.4234	0
5	0.5041	0.6287	0.4238	0
7	0.5041	0.6287	0.4238	0
9	0.5033	0.6297	0.4233	0
10	2.4001	4.2036	2.8624	27

(c) UKF

Sensor Fusion Options	Root Mean Square Difference (°)			Latency (ms)
	Pitch	Yaw	Roll	
1	0.1392	0.1175	0.2070	0
2	1. 6509	3.6503	1. 6480	26
3	0. 1721	0. 1106	0. 1685	0
5	0. 1722	0. 1106	0. 1686	0
7	0. 2237	0. 1703	0. 1806	0
9	0. 0846	0. 1811	0. 1556	0
10	1.8729	4.0383	1.9223	27

(d) PF

Sensor Fusion Options	Root Mean Square Difference (°)			Latency (ms)
	Pitch	Yaw	Roll	
1	0.3968	0.5937	1.0801	0
2	1.1000	2.3543	0.8501	26
3	0.4232	0.5067	0.8820	0
5	0.4188	0.5407	0.9285	0
7	0.4789	0.2496	0.3088	0
9	0.8907	1.7507	1.0289	0
10	1.1423	2.7504	1.1704	26

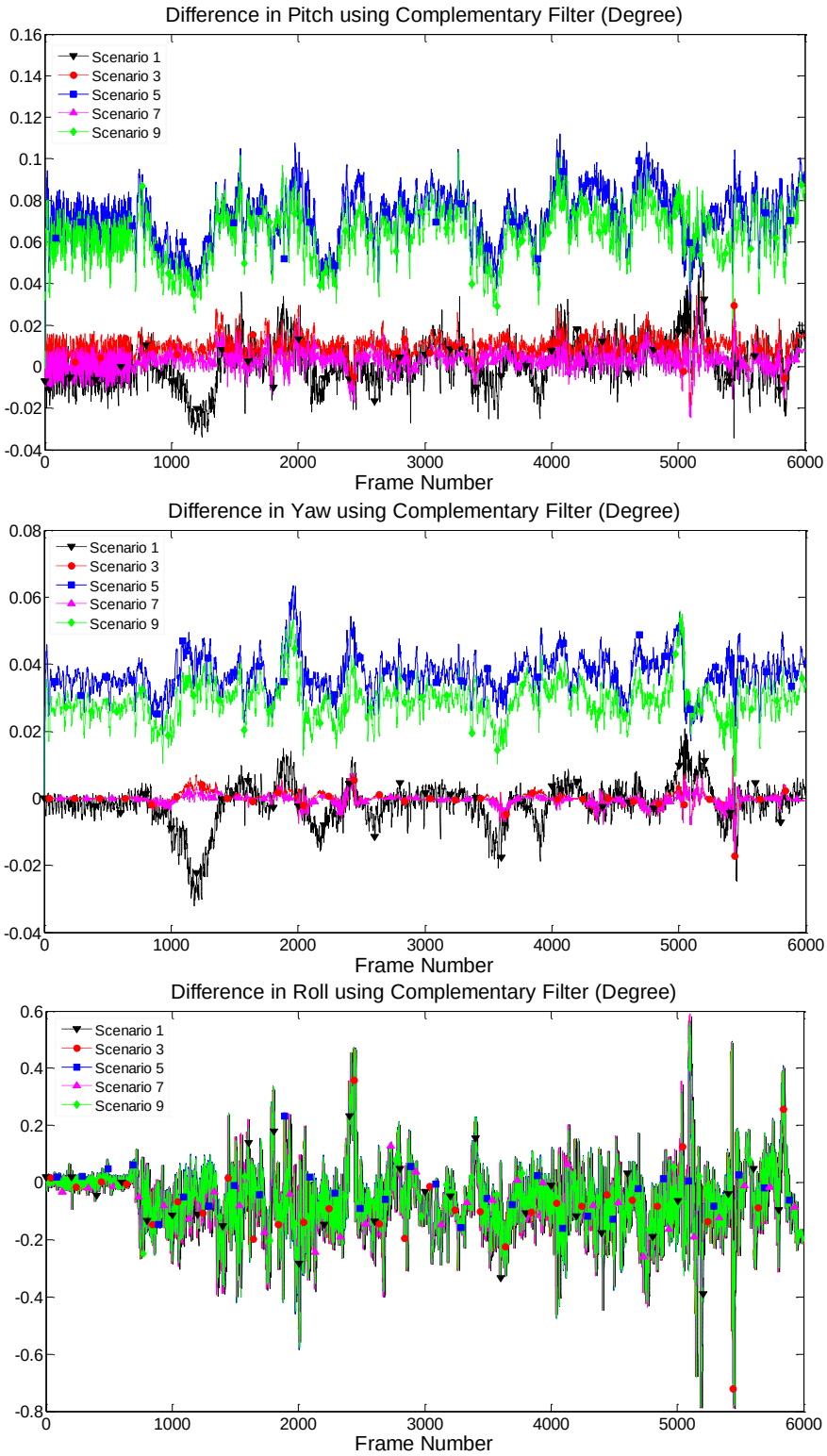

Fig. 12. Difference between the results from the motion capture system and those from the quaternion-based complementary filter when motion capture information is used.

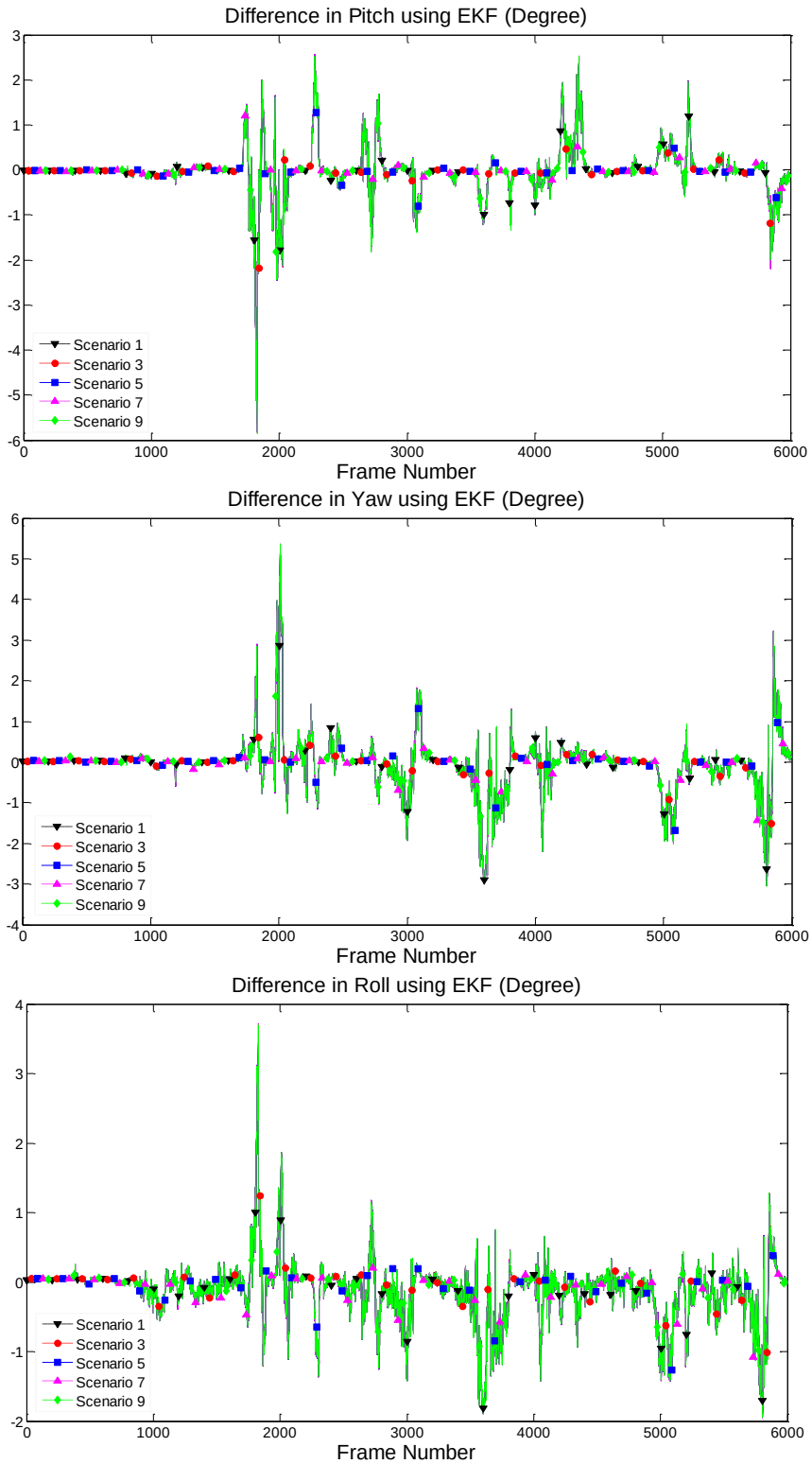

Fig. 13. Difference between the results from the motion capture system and those from the quaternion-based EKF when motion capture information is used.

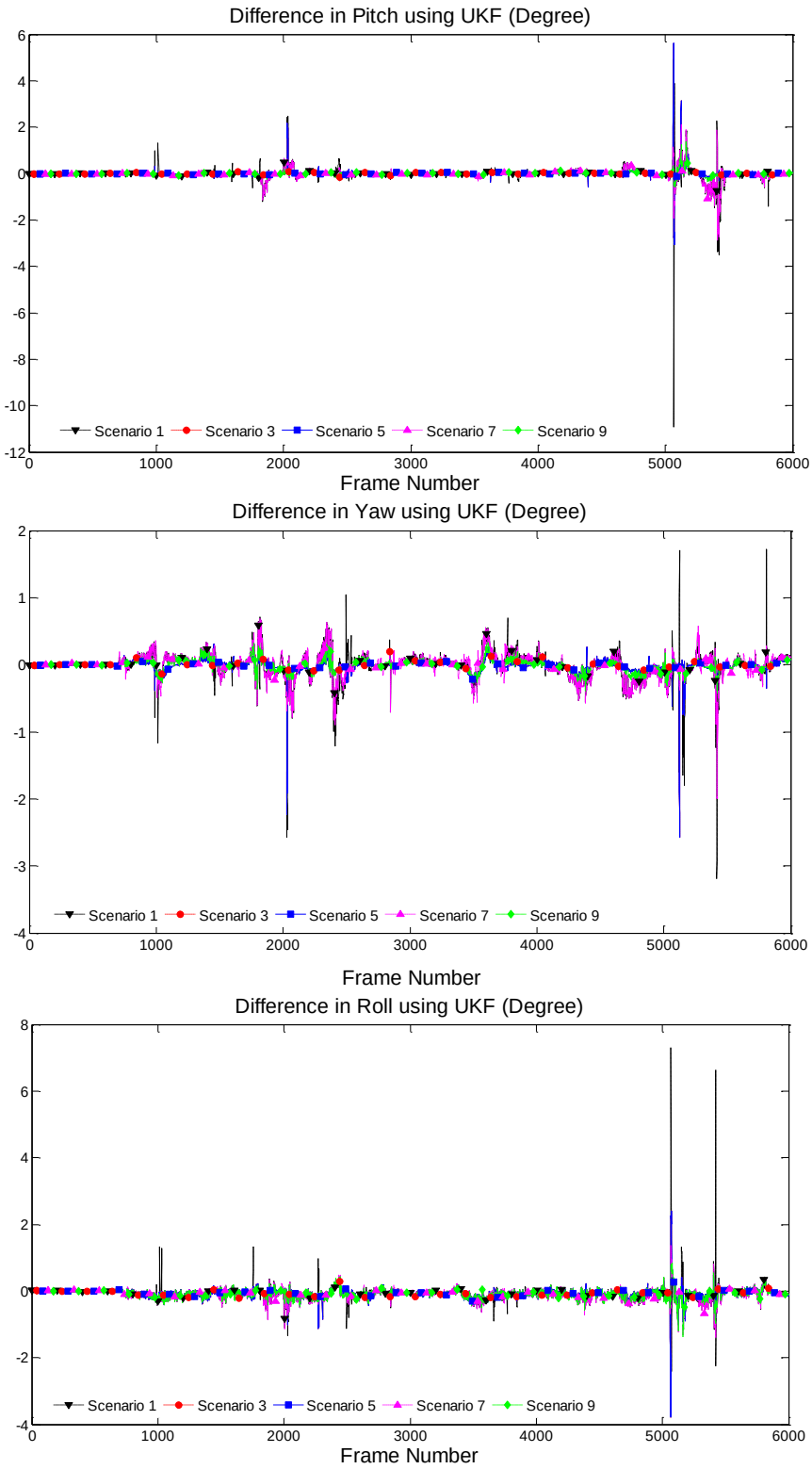

Fig. 14. Difference between the results from the motion capture system and those from the quaternion-based UKF when motion capture information is used.

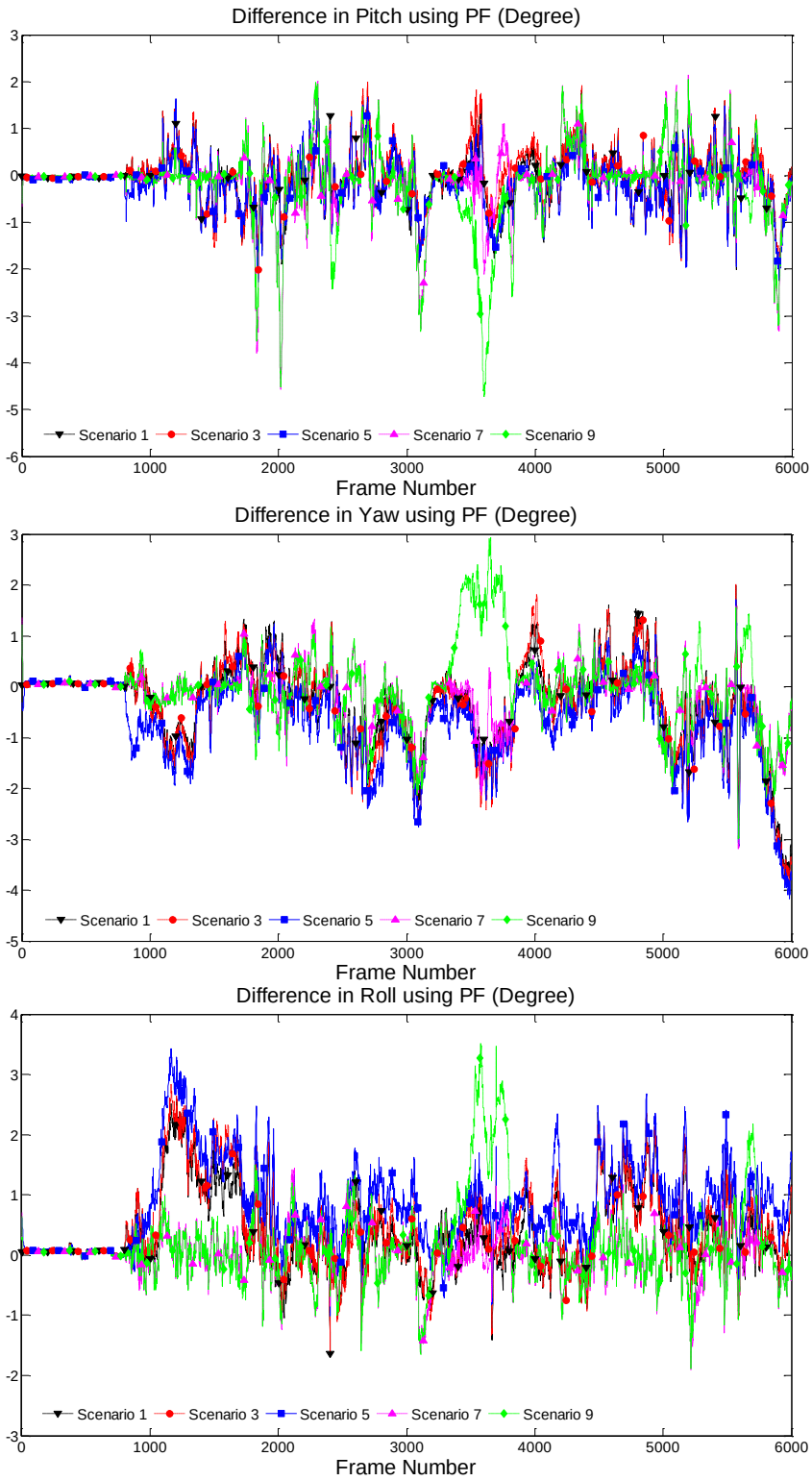

Fig. 15. Difference between the results from the motion capture system and those from the quaternion-based particle filter when motion capture information is used.

When no sufficient source can be used to remove the drifting, like the cases with only one reference vector, we have further validated the necessity of using the method introduced in Section 5.1.4. Figure 20 shows the tracking results with and without using the proposed algorithm that handles the tracking with only one reference vector in Scenario 6 when only the magnetometer reading is used in the update step. It can be seen that the tracking accuracy has been largely improved by using the proposed method for fusing the information from the gyro and that about the single reference vector.

Besides, it can be seen from Table 9 (a) – (d) that, compared to using the gyro alone in Scenario 8, deploying an extra sensor, such as the accelerometer in Scenario 4 and the magnetometer in Scenario 6, helps mitigating the drifting. A 10% improvement in the drift rate has been obtained by Scenario 4 for the complementary filter in Table 9 (a). Table 9 (c) – (d) show that this strategy also works for other filters that composes of the prediction and the update phases. Note that from Table 9 it is clear that Scenario 4 using the accelerometer is better than Scenario 6 using the magnetometer. This is because in our experiments, the readings from the accelerometer are less noisy than those of the magnetometer. As illustrated by Table 10, when the sensors are static, the standard deviations of the inertial vector are less than those of the magnetic field vector.

Finally, the computational costs of the four adopted filters are compared. The averaged running time (in Matlab code) for processing a 60s trial using the complementary filter, the EKF, the UKF and the PF are listed in Table 11. From Table 11 we can see that, the computational complexity of the PF is much higher. Such high computation load results from the resampling of the PF. The time complexity of the resampling step is proportional to the particle number. Such operation introduces a loop into every iteration of the PF, which dramatically amplifies the running time in Matlab. The complementary filter requires the least time to finish the processing.

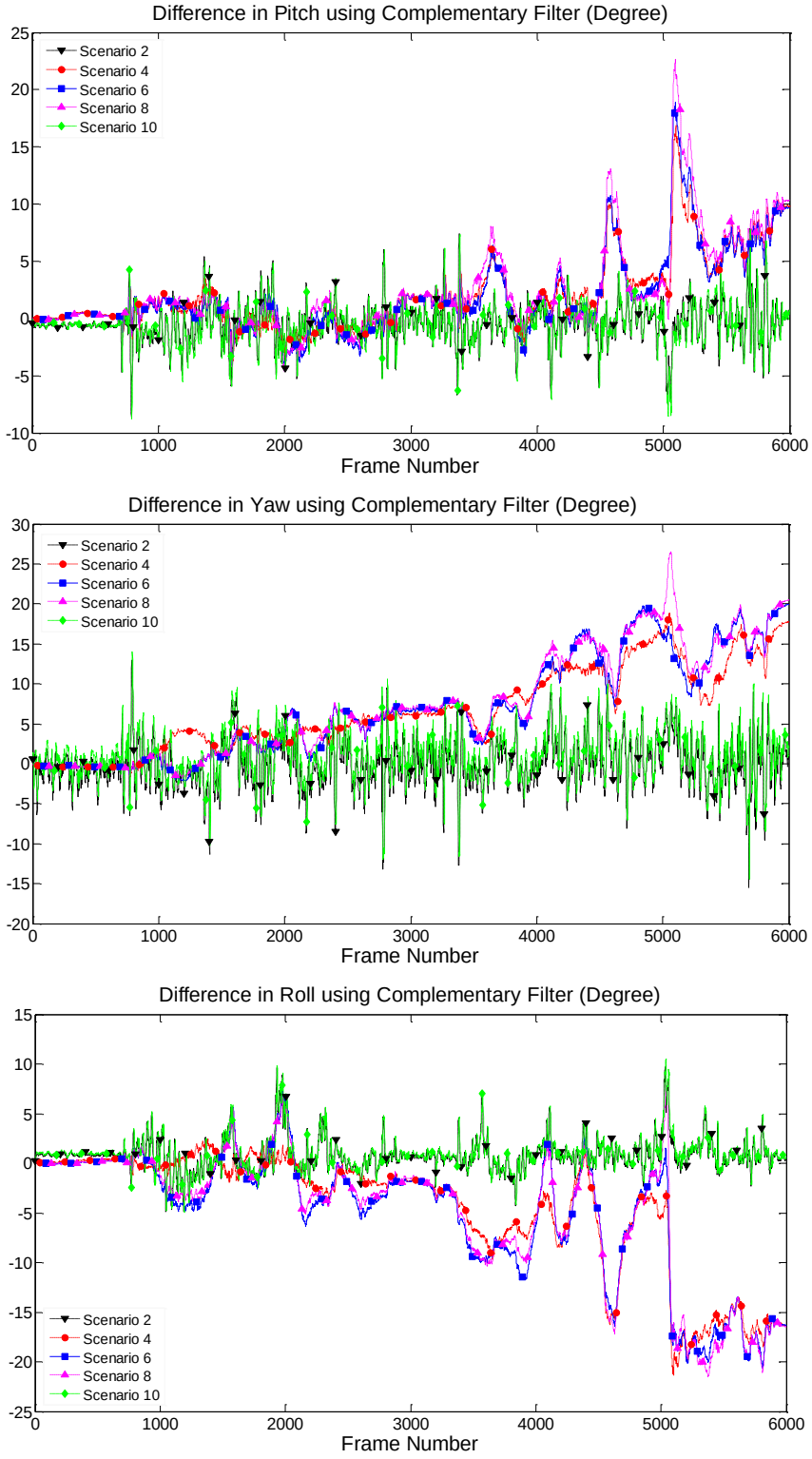

Fig. 16. Difference between the results from the motion capture system and those from the quaternion-based complementary filter without using motion capture data.

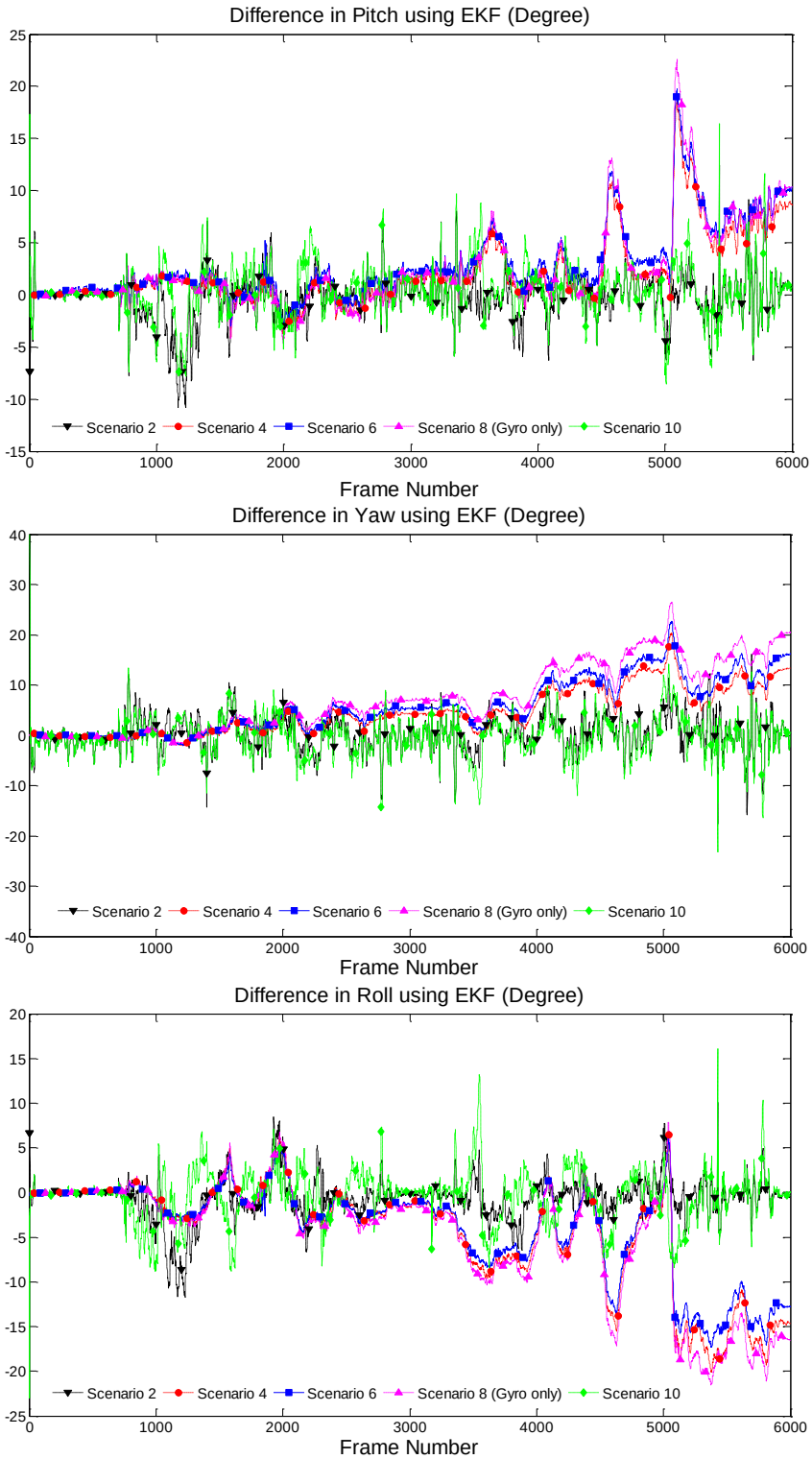

Fig. 17. Difference between the results from the motion capture system and those from the quaternion-based EKF without using motion capture data.

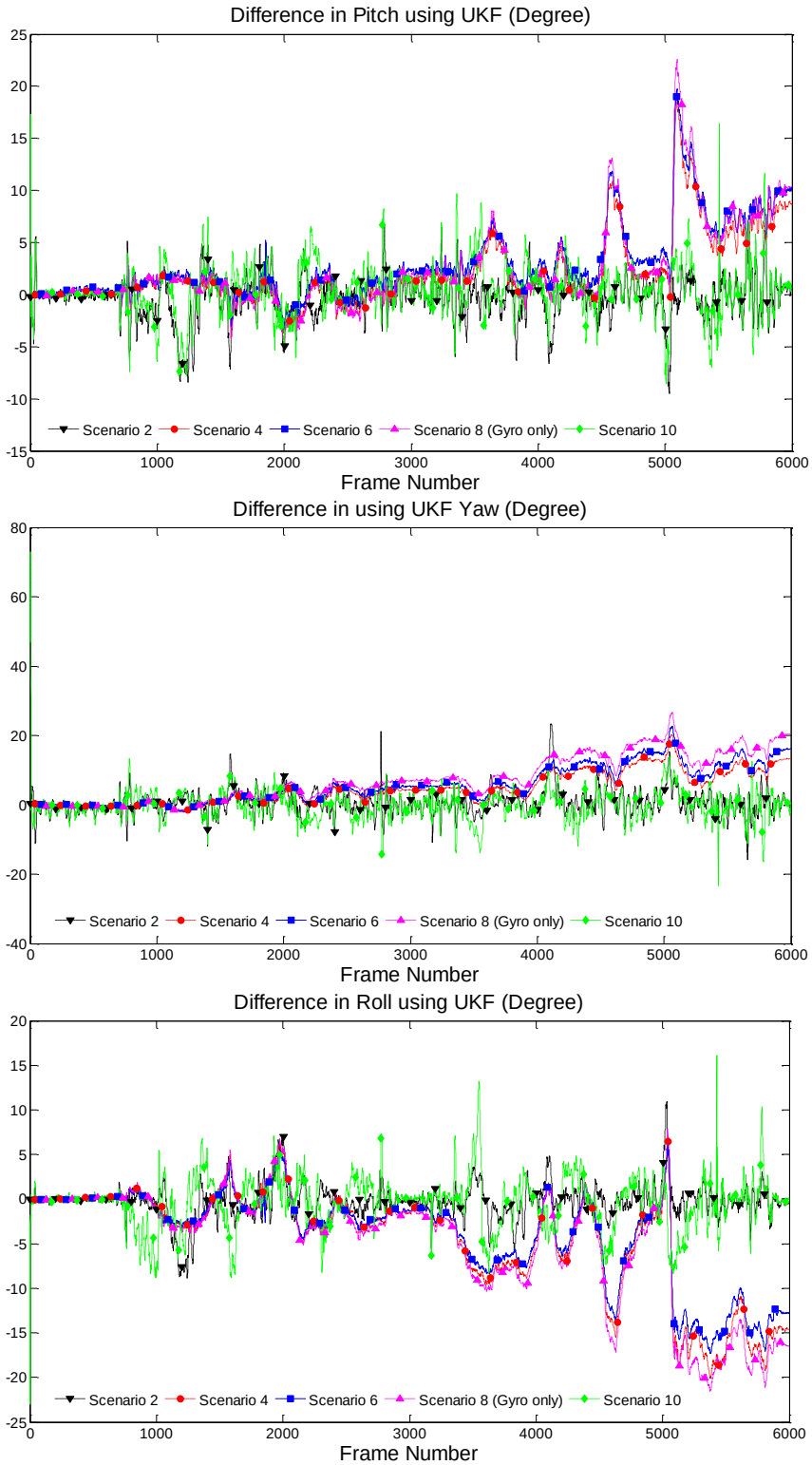

Fig. 18. Difference between the results from the motion capture system and those from the quaternion-based UKF without using motion capture data.

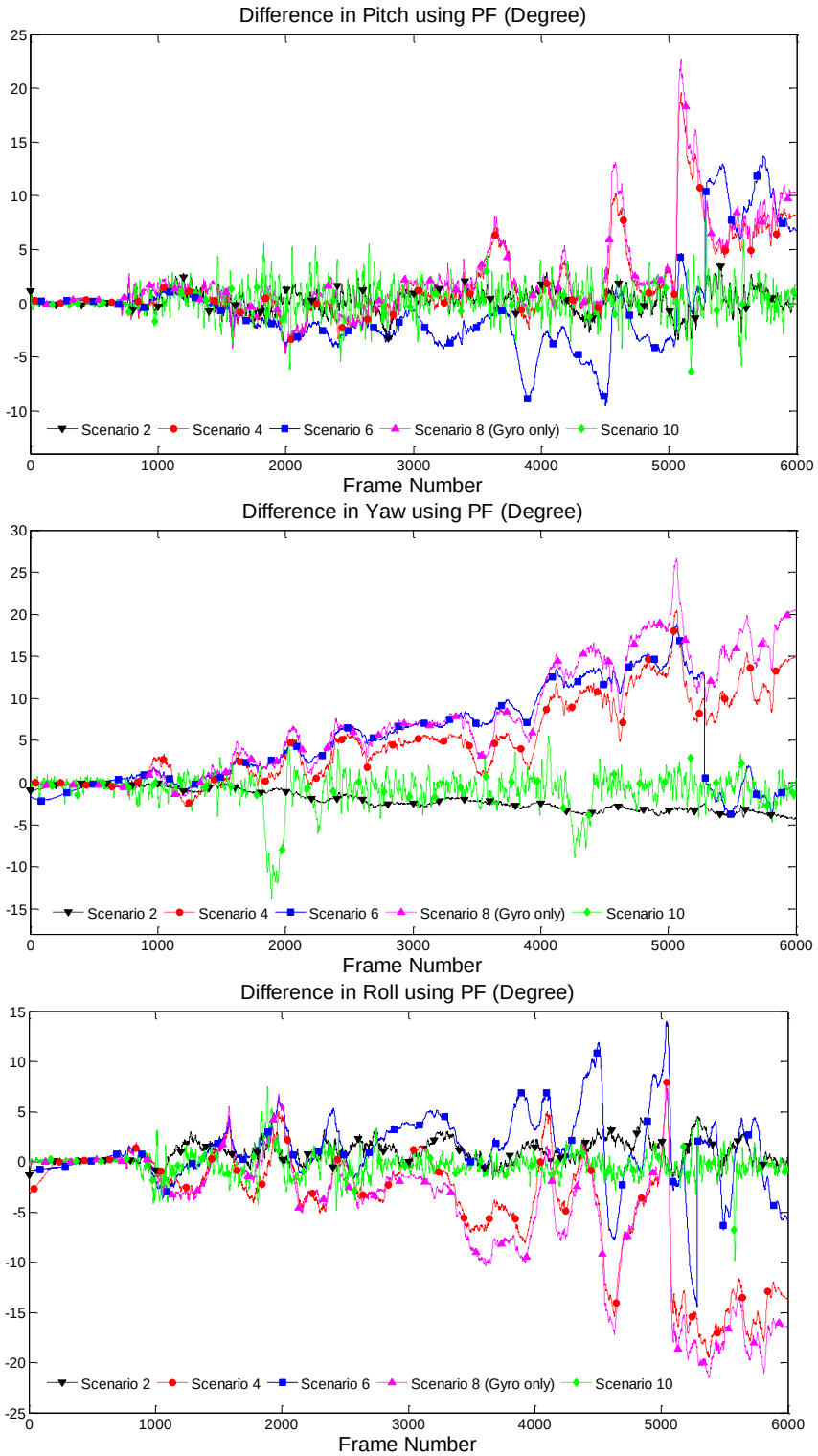

Fig. 19. Difference between the results from the motion capture system and those from the quaternion-based particle filter without using motion capture data.

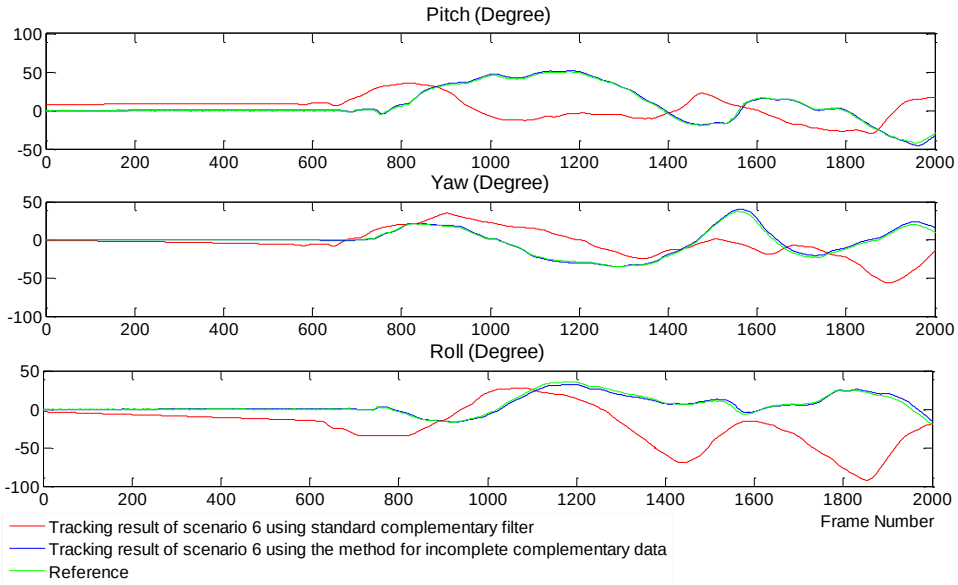

Fig. 20. Comparison between tracking results with (blue curves) and without (red) using the single reference vector handling algorithm for Scenario 6 when only the magnetometer reading is available as the complementary data

Table 9. Average Drifting Rates using Single Reference Vector in 20s

(a) Complementary Filter

Sensing scenarios	Average drifting rates in 20s (°/s)		
	Pitch	Yaw	Roll
4	0.0871	0.1775	0.3575
6	0.0940	0.2208	0.3708
8	0.0988	0.2484	0.3957

(b) EKF

Sensing scenarios	Average drifting rates in 20s (°/s)		
	Pitch	Yaw	Roll
4	0.0976	0.1892	0.3890
6	0.1123	0.2459	0.3827

(c) UKF

Sensing scenarios	Average drifting rates in 20s (°/s)		
	Pitch	Yaw	Roll
4	0.0627	0.1858	0.3826
6	0.1065	0.2479	0.3810

(d) PF

Sensing scenarios	Average drifting rates in 20s (°/s)		
	Pitch	Yaw	Roll
4	0.0839	0.1951	0.3421
6	0.0963	0.1928	0.3433

Table 10. Standard Deviations of IMU Reference Vectors in Static Case

Reference vector	X	Y	Z
Inertial vector	0.0022	0.00082	0.00037
Magnetic field vector	0.0023	0.0025	0.0010

Table 11. Averaged Time Cost for Processing a 60s Trial

Filtering algorithms	Complementary filter	EKF	UKF	PF
Running time (s)	2.9175	3.7203	4.4018	2166.1245

In summary, with respect to the tracking accuracy in the proposed system, the complementary filter provides smaller RMSD in all scenarios than the Kalman filters. Such observation implies the complementary filter is quite satisfying in describing the nonlinearity for the quaternion-based orientation tracking task. Also, the complementary filter does not require extensive computation. Thus it is a proper choice for fusing data from the motion capture system and that from the IMU. The particle filter is promising for handling noisy sensory data for systems with sufficient computational sources.

8. CONCLUSION

The proposed multimodal movement tracking framework using the quaternion-based filtering algorithms can work reasonably well in a wide range of sensing scenarios in terms of sensor combination. When the sensing scenarios become challenging, e.g., using less sensors, the performance of the proposed tracking approach degrades elegantly. The four filters implemented for this project: the complementary filter, the extended Kalman filter, the unscented Kalman filter and the particle filter, all demonstrate their capability in the fusion of motion capture data and IMU data. Through the comparison between the Bayesian approaches (the Kalman filters and the particle filter) and the least square approach (the complementary filter), it can be concluded that the complementary filter is a more effective method for integrating the IMU output and the vision information. When handling sensor fusion within the IMU, the particle filter provides superior performance.

REFERENCES

[1] S. Zhang, H. Hu, H. Zhou, An interactive Internet-based system for tracking upper limb motion in home-based rehabilitation, Medical Biological Engineering and Computing. (2008) 241-249.

[2] Y. Chen, W. Xu, H. Sundaram, T. Rikakis, S. Liu, Media Adaptation Framework in Biofeedback System for Stroke Patient Rehabilitation, Proceedings of the 15th international conference on Multimedia, September 25-29, 2007, Augsburg, Germany. (2007).

[3] M. Duff, Y. Chen, S. Attygalle, J. Herman, H. Sundaram, G. Qian, J. He, T. Rikakis, An Adaptive Mixed Reality Training System for Stroke Rehabilitation, in press, IEEE Transactions on Neural Systems and Rehabilitation Engineering (2010).

[4] H. Zhou, H. Hu, A survey - human movement tracking and stroke rehabilitation, University of Essex, UK. (2004).

[5] J. Aggarwal, Q. Cai, Human motion analysis: a review, Proceedings IEEE Nonrigid And Articulated Motion Workshop. (1999) 90-102.

[6] D. Gavrila, The Visual Analysis of Human Movement: A Survey, Computer Vision And Image Understanding. 73 (1999) 82-98.

[7] T. Moeslund, A Survey of Computer Vision-Based Human Motion Capture, Computer Vision And Image Understanding. 81 (2001) 231-268.

[8] L. Wang, W. Hu, T. Tan, Recent Developments in Human Motion Analysis, Pattern Recognition. 36 (2003) 585-601.

[9] C. Sminchisescu, Estimation Algorithms for Ambiguous Visual Models 3D Human Modeling & Motion Reconstruction in Monocular Video Sequences, PhD Thesis, Institute National Politechnique de Grenoble INRIA, France. (2002).

[10] E.R. Bachmann, R.B. McGhee, X. Yun, M.J. Zyda, Inertial and magnetic posture tracking for inserting humans into networked virtual environments, Proceedings of the ACM Symposium On Virtual Reality Software And Technology (2001) 9-16.

[11] D. Roetenberg, H.J. Luinge, C.T. Baten, P.H. Veltink, Compensation of Magnetic Disturbances Improves Inertial and Magnetic Sensing of Human Body Segment Orientation, Rehabilitation. 13 (2005) 395-405.

[12] H. Zhou, H. Hu, Inertial motion tracking of human arm movements in stroke rehabilitation, Proceedings of the IEEE International Conference on Mechatronics & Automation, July 2005, Niagara Falls, Canada. (2005) 1306-1311.

[13] A.D. Young, M.J. Ling, D.K. Arvind, Orient-2: A Realtime Wireless Posture Tracking System Using Local Orientation Estimation, Proceedings of the 4th workshop on Embedded networked sensors table of contents, Cork, Ireland. (2007) 53-57.

[14] E. R. Bachmann, Inertial and magnetic tracking of limb segment orientation for inserting humans into synthetic environments, Ph.D. dissertation, Naval Postgraduate School, Monterey, California, USA. (2000).

[15] http://www.xsens.com/

[16] http://www.microstrain.com/

[17] http://www.sparkfun.com/

[18] J. D. Hol, T. B. Schon, H. Luinge, P. J. Slycke, F. Gustafsson, Robust real-time tracking by fusing measurements from inertial and vision sensors, J Real-Time Image Proc, Rochester, New York. (2007) 149-160.

[19] S. You, U. Neumann, Fusion of vision and gyro tracking for robust augmented reality registration, Proceedings IEEE Virtual Reality. (2001) 71-78.

[20] P. Lang, M. Ribo, A. Pinz, A new Combination of Vision-Based and Inertial Tracking for Fully Mobile, Wearable, and Real-Time Operation, Proceedings of 26th Workshop of the Austrian Association for Pattern Recognition, Graz, Austria. (2002) 141-148.

[21] J. Chen, A. Pinz, Structure and Motion by Fusion of Inertial and Vision-Based Tracking, Proceedings of the 28th OAGM/AAPR Conference. 179 (2004) 55-62.

[22] L. Chai, 3-D Motion and Structure Estimation Using Inertial Sensors and Computer Vision for Augmented Reality, Tele-operators and Virtual Environments. 11 (2002) 474-492.

[23] D. Strelow, S. Singh, Online Motion Estimation from Image and Inertial Measurements, Workshop on Integration of Vision and Inertial Sensors (INERVIS), Coimbra, Portugal. (2003).

[24] A. Huster, Relative position sensing by fusing monocular vision and inertial rate sensors, PhD thesis. Department of Electrical Engineering, Stanford University, USA. (2003).

[25] G.S. Klein, T.W. Drummond, Tightly integrated sensor fusion for robust visual tracking, Image Vis. Comput. 22 (2004) 769-776.

[26] J. Lobo, J. Dias, Inertial sensed ego-motion for 3D vision, J. Robot. Syst. 21 (2004) 3-12.

[27] L. Armesto, J. Tornero, M. Vincze, Fast ego-motion estimation with multi-rate fusion of inertial and vision. Int. J. Robot. Res. 26 (2007) 577–589.

[28] M. Ribo, M. Brandner, A. Pinz, A Flexible Software Architecture for Hybrid Tracking, Journal Of Robotic Systems. 21 (2004) 53-62.

[29] Y. Tao, H. Hu, H. Zhou, Integration of Vision and Inertial Sensors for 3D Arm Motion Tracking in Home-based Rehabilitation, The . 26 (2007) 607-624.

[30]Honeywell, Cross Axis Effect for AMR Magnetic Sensors.

[31] R. McGhee, Some Parameter-Optimization Techniques, Digital Computer User's Handbook, McGraw-Hill. (1967) 234-253.

[32] R. McGhee, E. Bachmann, X. Yun, M. Zyda, Real-Time Tracking and Display of Human Limb Segment Motions Using Sourceless Sensors and a Quaternion-Based Filtering Algorithm - Part I: Theory, MOVES Academic Group Technical Report NPS-MV-01-001, Naval Postgraduate School, Monterey, CA. (2000).

[33] X. Pennec, Computing the mean of geometric features – Application to the mean rotation, Institut National de Recherche en Informatique et en Automatique (INRIA), Le Chesnay, France (1998).

APPENDIX A

THE JACOBI MATRIX FOR QUATERNION ROTATION

Let $\boldsymbol{u}(\boldsymbol{q})$ represents a quaternion rotation function that rotates a fixed vector $\boldsymbol{a}$ to $\boldsymbol{a}'$ with quaternion $\boldsymbol{q}$ in the pattern given by

$$\boldsymbol{a}' = \boldsymbol{u}(\boldsymbol{q}) = \boldsymbol{q}^{-1} \otimes \boldsymbol{a} \otimes \boldsymbol{q} \tag{A-1}$$

Its Jacobi matrix J, whose j-th column J_j is defined as partial differential calculus

$$J_j = \frac{\partial \boldsymbol{u}(\boldsymbol{q})}{\partial q_{j-1}} = \frac{\partial (\boldsymbol{q}^{-1} \otimes \boldsymbol{a} \otimes \boldsymbol{q})}{\partial q_{j-1}} \tag{A-2}$$

where q_j is the j-th ccomponent of quaternion $\boldsymbol{q}$ (j = 1, 2, 3, 4). Note that, since in equation (A-1) $\boldsymbol{a}'$ is only the virtual part of the quaternion product, $\boldsymbol{u}(\boldsymbol{q})$ is a 3-element vector. Thus the length of J_j is 3. Consequently J is a 3 by 4 matrix.

The calculation of J_j takes use of the product rule of differential calculus

$$J_j = \frac{\partial (\boldsymbol{q}^{-1} \otimes \boldsymbol{a} \otimes \boldsymbol{q})}{\partial q_{j-1}} = \frac{\partial \boldsymbol{q}^{-1}}{\partial q_{j-1}} \otimes \boldsymbol{a} \otimes \boldsymbol{q} + \boldsymbol{q}^{-1} \otimes \boldsymbol{a} \otimes \frac{\partial \boldsymbol{q}}{\partial q_{j-1}} \tag{A-3}$$

According to the definition of quaternion given by (10), $\frac{\partial \boldsymbol{q}}{\partial q_j}$ is derived by

$$\frac{\partial \boldsymbol{q}}{\partial q_0} = [1, 0, 0, 0]^T = 1 \tag{A-4}$$

$$\frac{\partial \boldsymbol{q}}{\partial q_1} = [0,1, 0, 0]^T = i \tag{A-5}$$

$$\frac{\partial \boldsymbol{q}}{\partial q_2} = [0, 0, 1, 0]^T = j \tag{A-6}$$

$$\frac{\partial \boldsymbol{q}}{\partial q_3} = [0, 0, 0, 1]^T = k \tag{A-7}$$

Since $\boldsymbol{q}$ is a unit quaternion, its inverse $\boldsymbol{q}^{-1}$ equals its conjugate $\boldsymbol{q}^*$, as demonstrated in equation (21)

$$\boldsymbol{q}^{-1} = \boldsymbol{q}^* = q_0 - q_1 i - q_2 j - q_3 k \tag{A-8}$$

Thus, $\frac{\partial \boldsymbol{q}^{-1}}{\partial q_{j-1}}$ can be calculated as

$$\frac{\partial \boldsymbol{q}^{-1}}{\partial q_0} = [1, 0, 0, 0]^T = 1 \tag{A-9}$$

$$\frac{\partial \boldsymbol{q}^{-1}}{\partial q_1} = [0, -1, 0, 0]^T = -i \tag{A-10}$$

$$\frac{\partial \boldsymbol{q}^{-1}}{\partial q_2} = [0, 0, -1, 0]^T = -j \tag{A-11}$$

$$\frac{\partial \boldsymbol{q}^{-1}}{\partial q_3} = [0, 0, 0, -1]^T = -k \tag{A-12}$$

Substitute (A-4) – (A-12) into (A-3) produces the four columns of J

$$J_1 = \frac{\partial(\boldsymbol{q}^{-1} \otimes \boldsymbol{a} \otimes \boldsymbol{q})}{\partial q_0} = \boldsymbol{a} \otimes \boldsymbol{q} + \boldsymbol{q}^{-1} \otimes \boldsymbol{a} \tag{A-13}$$

$$J_2 = \frac{\partial(\boldsymbol{q}^{-1} \otimes \boldsymbol{a} \otimes \boldsymbol{q})}{\partial q_1} = -i \otimes \boldsymbol{a} \otimes \boldsymbol{q} + \boldsymbol{q}^{-1} \otimes \boldsymbol{a} \otimes i \tag{A-14}$$

$$J_3 = \frac{\partial(\boldsymbol{q}^{-1} \otimes \boldsymbol{a} \otimes \boldsymbol{q})}{\partial q_2} = -j \otimes \boldsymbol{a} \otimes \boldsymbol{q} + \boldsymbol{q}^{-1} \otimes \boldsymbol{a} \otimes j \tag{A-15}$$

$$J_4 = \frac{\partial(\boldsymbol{q}^{-1} \otimes \boldsymbol{a} \otimes \boldsymbol{q})}{\partial q_3} = -k \otimes \boldsymbol{a} \otimes \boldsymbol{q} + \boldsymbol{q}^{-1} \otimes \boldsymbol{a} \otimes k \tag{A-16}$$

Since $\boldsymbol{u}(\boldsymbol{q})$ returns a 3-element rotated vector, J_j only takes the virtual part of the quaternions in equation (A-13) through (A-16). Combining (A-13) – (A-16) presents the 3 by 4 Jacobi matrix J as

$$J = [J_1 | J_2 | J_3 | J_4] \tag{A-17}$$

APPENDIX B

PROOF OF THE ORTHOGONAL QUATERNION THEOREM

The orthogonal quaternion theorem: If two quaternions $\boldsymbol{q_A}$ and $\boldsymbol{q_B}$ are orthogonal to each other, $\boldsymbol{q_A}$ can be written into the form

$$\boldsymbol{q_A} = \boldsymbol{q_B} \otimes \boldsymbol{p} \tag{B-1}$$

where $\boldsymbol{p}$ is unique vector resulting from

$$\boldsymbol{p} = \boldsymbol{q_B}^{-1} \otimes \boldsymbol{q_A} \tag{B-2}$$

Proof: Using the quaternion notation given by equation (1)

$$\boldsymbol{q_B} = q_{B0} + q_{B1}i + q_{B2}j + q_{B3}k \tag{B-3}$$

$$\boldsymbol{p} = p_1 i + p_2 j + p_3 k \tag{B-4}$$

The quaternion multiplication $\boldsymbol{q_B} \otimes \boldsymbol{p}$ produces $\boldsymbol{q_A}$

$$\boldsymbol{q_A} = \boldsymbol{q_B} \otimes \boldsymbol{p} = (-q_{B1}p_1 - q_{B2}p_2 - q_{B3}p_3) + (q_{B0}p_1 + q_{B2}p_3 - q_{B3}p_2)i$$
$$+(q_{B0}p_2 + q_{B3}p_1 - q_{B1}p_3)j + (q_{B0}p_3 + q_{B1}p_2 - q_{B2}p_1)k \tag{B-5}$$

Then the dot product between $\boldsymbol{q_B}$ and $\boldsymbol{q_A}$ is

$$\boldsymbol{q_A} \cdot \boldsymbol{q_B} = (-q_{B1}p_1 - q_{B2}p_2 - q_{B3}p_3)q_{B0} + (q_{B0}p_1 + q_{B2}p_3 - q_{B3}p_2)q_{B1}$$
$$+(q_{B0}p_2 + q_{B3}p_1 - q_{B1}p_3)q_{B2} + (q_{B0}p_3 + q_{B1}p_2 - q_{B2}p_1)q_{B3} = 0 \tag{B-6}$$

which shows that $\boldsymbol{q_A}$ and $\boldsymbol{q_B}$ are orthogonal to each other.

Substitute (B-5) into (B-2) presents

$$\boldsymbol{p} = \boldsymbol{q_B}^{-1} \otimes \boldsymbol{q_A} = \boldsymbol{q_B}^{-1} \otimes \boldsymbol{q_B} \otimes \boldsymbol{p} = \boldsymbol{p} \tag{B-7}$$

which proves uniqueness. Thus the orthogonal quaternion theorem holds true.

APPENDIX C

DERIVATION OF THE JACOBI MATRIX F FOR $f(\hat{\boldsymbol{x}}_k, \mathbf{0})$

The state transition function $f(\hat{\boldsymbol{x}}_k, \mathbf{0})$ returns a 4-element predicted quaternion orientation $\boldsymbol{x}_k^-$ as

$$\boldsymbol{x}_k^- = f(\hat{\boldsymbol{x}}_k, \mathbf{0}) = \hat{\boldsymbol{x}}_k \tag{C-1}$$

when gyroscope data is not involved in the filtering. Then F is a 4 by 4 identity matrix.

When gyroscope data is introduced into the prediction step, $\boldsymbol{x}_k^-$ is a 7-element vector as $\hat{\boldsymbol{x}}_k$ as

$$\boldsymbol{x}_k^- = f(\hat{\boldsymbol{x}}_k, 0) = \begin{bmatrix} d(\hat{\boldsymbol{q}}_{k-1}, \hat{\boldsymbol{\omega}}_{\boldsymbol{M}_{k-1}}) \\ \hat{\boldsymbol{\omega}}_{\boldsymbol{M}_{k-1}} \end{bmatrix} \tag{C-2}$$

where $d(\hat{\boldsymbol{q}}_k, \hat{\boldsymbol{\omega}}_{\boldsymbol{M}_{k-1}})$ is defined as

$$d(\hat{\boldsymbol{q}}_k, \hat{\boldsymbol{\omega}}_{\boldsymbol{M}_{k-1}}) = \hat{\boldsymbol{q}}_k + \frac{\Delta t}{2}\hat{\boldsymbol{q}}_k \otimes \hat{\boldsymbol{\omega}}_{\boldsymbol{M}_{k-1}} = \hat{\boldsymbol{q}}_k + \frac{\Delta t}{2}\hat{\boldsymbol{q}}_k \otimes [0, \omega_{Mx}, \omega_{My}, \omega_{Mz}]^{\mathrm{T}} \tag{C-3}$$

Note that in (C-3), $\hat{\boldsymbol{\omega}}_{\boldsymbol{M}_{k-1}}$ is in its quaternion form, whose virtual part equals the vector $\hat{\boldsymbol{\omega}}_{\boldsymbol{M}_{k-1}}$. F turns to be a 7 by 7 matrix, which can be defined in the form

$$F = \frac{\partial f}{\partial x} = \frac{\partial \begin{bmatrix} d(\hat{\boldsymbol{q}}_{k-1}, \hat{\boldsymbol{\omega}}_{\boldsymbol{M}_{k-1}}) \\ \hat{\boldsymbol{\omega}}_{\boldsymbol{M}_{k-1}} \end{bmatrix}}{\partial x} = \begin{bmatrix} \frac{\partial d}{\partial q} & \frac{\partial d}{\partial \omega} \\ 0_{3\times 4} & I_3 \end{bmatrix} \tag{C-4}$$

where the i-th column of the 4 by 4 matrix $\frac{\partial d}{\partial q}$ is

$$\left(\frac{\partial d}{\partial q}\right)_i = \frac{\boldsymbol{\partial q}}{\boldsymbol{\partial} q_i} + \frac{\Delta t}{2}\frac{\boldsymbol{\partial q}}{\boldsymbol{\partial} q_{i-1}} \otimes \hat{\boldsymbol{\omega}}_{\boldsymbol{M}_{k-1}} \; (i = 1, 2, 3, 4) \tag{C-5}$$

and the j-th column of the 4 by 3 matrix $\frac{\partial d}{\partial q}$ is

$$\left(\frac{\partial d}{\partial \omega}\right)_j = \frac{\Delta t}{2}\hat{\boldsymbol{q}}_k \otimes \frac{\partial \boldsymbol{\omega}}{\partial \omega_j} = \quad (j = 1, 2, 3) \tag{C-6}$$

Since in $d(\hat{\boldsymbol{q}}_k, \hat{\boldsymbol{\omega}}_{\boldsymbol{M}_{k-1}})$ given by (C-3), $\hat{\boldsymbol{\omega}}_{\boldsymbol{M}_{k-1}}$ is also used a quaternion. Thus

$$\frac{\boldsymbol{\partial \omega}}{\partial \omega_j} = \frac{\boldsymbol{\partial q}}{\boldsymbol{\partial} q_j} \; (j = 1, 2, 3) \tag{C-7}$$

Consequently

$$\left(\frac{\partial d}{\partial \omega}\right)_j = \frac{\Delta t}{2}\widehat{\boldsymbol{q}}_k \otimes \frac{\boldsymbol{\partial q}}{\boldsymbol{\partial} q_j} = \quad (j = 1, 2, 3) \tag{C-8}$$

The calculation of $\frac{\boldsymbol{\partial q}}{\boldsymbol{\partial} q_{i-1}}$ $(i = 1, 2, 3, 4)$ has been clarified by (A-4) – (A-7).