A Low-Cost High-Fidelity Ultrasound Simulator with the Inertial Tracking of the Probe Pose

Saverio Farsoni[a,*], Marcello Bonfè[a], Luca Astolfi[b]

[a] *Department of Engineering, University of Ferrara, Italy*
[b] *Department of Morphology, Surgery and Experimental Medicine, Institution of Anesthesia and Intensive care, University of Ferrara, Italy*

Abstract

The authors developed a versatile ultrasound simulator. The proposed system achieves the main features of a high-fidelity device exploiting low-cost rapid prototyping hardware. The hand-guided ultrasound simulator probe includes a RFID reader, a 9-DOF inertial sensor unit, consisting of an accelerometer, a magnetometer and a gyroscope, and a microcontroller that performs the real-time data acquisition, the processing and the transmission of the estimated pose information to the visualization system, so that the proper ultrasound view can be generated. Since the probe orientation is the main information involved in the pose reconstruction, this work presents and investigates several tracking methods for the probe orientation, exploiting a sensor fusion technique to filter the noisy measurements coming from inertial sensors. The performances of a Kalman filter, a nonlinear complementary filter and a quaternion-based filter as inertial trackers have been tested by means of a robot manipulator, in terms of readiness, accuracy and stability of the estimated orientation signal. The results show that the nonlinear complementary filter and the quaternion-based filter match all the application requirements (RMSE < 3 deg, variance < 1 deg^2, settling time < 0.3 s), and they involve a lower computational time with respect to the Kalman filter.

Keywords: Inertial Tracking, Ultrasound Simulator, Sensor Fusion, Medical Training, Real-time Algorithms

[*] Corresponding author
Email address: saverio.farsoni@unife.it (Saverio Farsoni)

1. Introduction

In the last decades, lots of studies have demonstrated the importance and the usefulness of simulation in medical training and education, in particular in the field of emergency medicine, where operators need to practice on critical cases without endanger the patient's health (Parks et al., 2013). A novel valuable tool, for providing an early diagnosis on trauma, concerns the use of Point Of Care Ultrasonography (POCUS), consisting in the acquisition and the evaluation of a series of ultrasound (US) scans focused on particular landmark of interest, such as the lungs or the heart (Vignon, 2012). The standard procedures described in the Focus Assessment with Sonography for Trauma (FAST) protocol are specifically introduced to facilitate the detection of free fluids (Gillman et al., 2009). A consequence of the larger diffusion of US scanners is the fact that more and more emergency operators have to be trained on their use (Oxorn and Pearlman, 2012). Moreover, US scanners used on the field cannot be used for training, because of their cost. Therefore, the biomedical industry has recently marketed many devices, capable of simulating an ultrasonography scenario, with different characteristics and features, as reported in (Blum et al., 2013). Taking into account the requirement of low-cost, high-fidelity and user-friendliness the authors designed a versatile US simulator comprehensive of the main advantages of the currently available systems, as well as novel features regarding the visualization of US data sets, the graphical interfaces adapted for education sessions and the extensible database. A prototype of the simulator is described in (Farsoni et al., 2015). The crucial point for the proper behavior of the system is the tracking of the simulator probe: i.e. the device reproducing the US transducer. Indeed, the real-time information about its current pose (understood as the merging of position and orientation) relative to the phantom is exploited by the visualization system in order to display the appropriate US view.

This problem can be addressed as the reconstruction of the pose of a rigid body (the probe) relative to another rigid body (the phantom), both moving in the Earth Reference System. Indeed, also the phantom can be voluntarily moved, or even accidentally bumped during the simulation, causing the alteration of the relative reference system. The tracking issue interests many engineering domains including robotics and aerospace as well as simulation and virtual reality, and their applications involve different motion characteristics: the body acceleration can range from the low values of a human hand to several time the gravity acceleration in an aircraft. The tracking systems discussed in (Welch and Foxlin, 2002) are categorized depending on the exploited technology: optical, image-based, mechanical, magnetic, inertial, acoustic, and hybrid systems represent a wide range of opportunities to overcome the problem, with different costs, size and performances. The inertial sensor trackers relies on the data acquired by microelectromechanical system (MEMS) based accelerometers, magnetometers and gyroscopes and they meet the US simulators scope in terms of cheapness and size (Serrano and Ayazi, 2015). However, the electronic offsets and drifts of this kind of sensors yield significant tracking errors during the integration of linear acceleration and angular rate, so that a dynamic pose reconstruction exploiting the numerical integration of acceleration (for the position) and angular rate (for the orientation) is impossible unless introducing an external reference, as pointed out in (Luinge and Veltink, 2005). The common solution is a sensor fusion algorithm that combines together the information coming from different inertial sensors.

Considering the requirements of the POCUS simulation, the reconstruction of the probe position can be solved using the RFID technology, by discretizing the phantom body in several landmarks of interest, as described in Section 2. The tracking of the probe orientation involves the acquisition and filtering of noisy measurement from the accelerometer, the magnetometer, and the gyroscope and represents the challenging task to overcome. The proposed solutions, described in Section 3.2, exploit two inertial tracking systems, one fixed to the phantom and another one fixed to the probe. Both the tracking systems estimate the orientation relative to the Earth Reference System (ERS), so that the probe orientation relative to the phantom can be calculated as the rotation required to align the systems.

This paper extends (Farsoni et al., 2015) by briefly presenting the updated hardware and software implementation, which is described in Section 2. Furthermore, the novel contribution of the work is a complete study concerning the inertial tracking of the probe orientation. Indeed, the usage of the prototype induced further investigations about the proper tracking method, since tracking errors, as well as tracking delays, strongly depend on the algorithm that processes the sensor measurements. In Section 3, after a brief introduction about how inertial sensors such as gyroscope, accelerometer and magnetometer can produce the orientation estimation, three sensor fusion tracking algorithms are presented. Then, Section 4 addresses the comparative assessment of the considered algorithms. Section 5 makes

	Component	Features
Microcontroller	Teensy 3.1	MK20DX256 32 bit ARM Cortex-M4 processor, 64k RAM
Inertial Sensor Unit	Invensense MPU-9150	3-axis accelerometer, 3-axis gyroscope, 3-axis magnetometer, Fast Model I2C
RFID reader	Innovation ID-12LA	125KHz Read Frequency, EM4001 64-bit tags, 9600bps TTL output

Table 1: The hardware components of the Simulator Probe.

some considerations about the obtained results, highlighting the motivations that support the final choice of the algorithm implemented on the system. Finally, Section 6 summarizes the overall development of the simulator.

2. System components

In this section the hardware and software development of the simulator is briefly summarized, highlighting the improvement and the difference with respect to the previous prototype implementation.

2.1. Hardware

In order to create an economically sustainable system, all of the hardware components are chosen with the aim to increase the performance/cost ratio. The list of components can be found in Table 1.

The design of the probe involves a microcontroller that accomplishes different tasks: firstly, measurements are acquired from the RFID reader and from the inertial sensors, then the elaboration of the tracking algorithm is performed and the information on the probe pose is real-time communicated to the visualization system. The RFID reader is located at the bottom side of the probe. It acquires the codes of a set of markers, typically nine, positioned under the phantom skin, in agreement with the point of care described in FAST procedure. When the reader enters in the field of a marker, the identification (ID) code uniquely associated to the marker is read. Since for each marker its location with reference to the phantom is known, the ID code specifies also the position of the probe on the phantom skin. The distance between tags should be at least 3 cm, in order to avoid the acquisition of the wrong ID code. The chosen component is the Innovation ID-12LA, which differs from the prototype RFID reader of (Farsoni et al., 2015) in terms of a lower input voltage and a wider read range, although it maintains the same size.

Another important hardware upgrade concerns the microcontroller, the previous choice of Arduino Nano did not satisfy the requirement of the application in terms of stability, as one UART had to be shared between the RFID reader and the visualization system, often producing communication errors. The Teensy 3.1 is a recent low-cost microcontroller featuring three UARTs, suitable for implementing the TTL data communication up to three devices. Also the core, a MK20DX256 32 bit ARM, ensures better performances during the tracking processing, and a larger amount of memory allows the deployment of the sensor calibration code within the elaboration algorithm script.

The firmware running on the microcontroller executes repeatedly a main loop consisting of six successive acquisitions from inertial sensors, with the corresponding estimations and transmissions of the probe orientation to the visualization system. Afterwards, the microcontroller performs the acquisition and the transmission of the RFID code and another loop can start. The previous microcontroller mean loop time was 285 ms, while the Teensy mean loop time is decreased to 180 ms, allowing a faster response in the visualization of a US data set.

Finally, the inertial sensor Invensense MPU-9150 includes the three-axis MEMS sensors (the accelerometer, the gyroscope and the magnetometer) and communicates the measurements via I2C to the microcontroller. It is the same component exploited in the prototype, but the position inside the probe is different: it is located more distant from the bottom of the device in order to reduce the magnetic disturbances caused by the RFID reader and by the external sources. The shell of the probe can be built by a 3D printer, different shapes can be produced, based on user needing. The probe and the inner components are shown in Figure 1.

It is worth noting that the hardware required to track the phantom orientation has to be fixed to the phantom, and it consists of an inertial sensor unit and a microcontroller which acquires the measurements, executes the orientation estimation and transmits the information to the visualization system. The RFID reader is instead not included, because its information is only related to the position tracking and not to the orientation tracking. Once the current orientation of the probe and the current orientation of the phantom have been estimated, the rotation required for aligning the two systems can be calculated and the final information on the relative orientation is there-

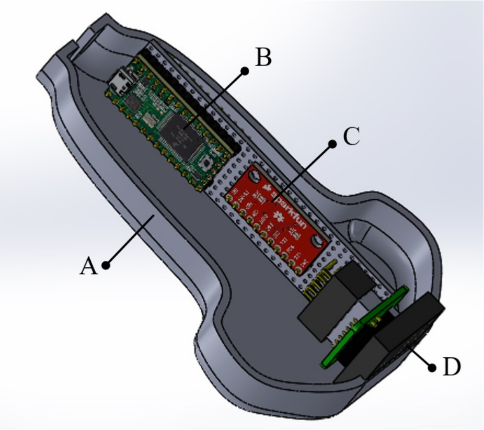

Figure 1: The 3D printed shell of the simulator probe (A) and the inner hardware components: from top to bottom: the microcontroller (B) , the inertial sensor unit (C), and the RFID reader (D).

fore provided. The overall system setup is depicted in the block diagram of Figure 2.

2.2. Visualization software

The software has been developed for Microsoft .NET Framework to permit the compatibility with the most common personal devices such as laptop, desktop or tablet. Two separated graphical user interfaces, namely one for the trainer and one for the trainee, running on different devices and communicating through a network socket, allow the execution of a standard training session, during which the instructor stays in its dedicated environment, selects the desired case from the database, and supervises the trainee's activity. The trainee, on the other hand, operates inside the simulation scenario which consists of the phantom, the probe, and the visualization system hosted in the trainee's application. He has to position and orientate the probe correctly and then he has to analyze the US view generated by the visualization system. The trainee's application allows the management of real US scans, in form of a single image, a video or a 3D volume, as explained more in detail in (Farsoni et al., 2015). Figure 3 shows the two graphical user interfaces and the operator during a simulation session.

Compensation of magnetic disturbances

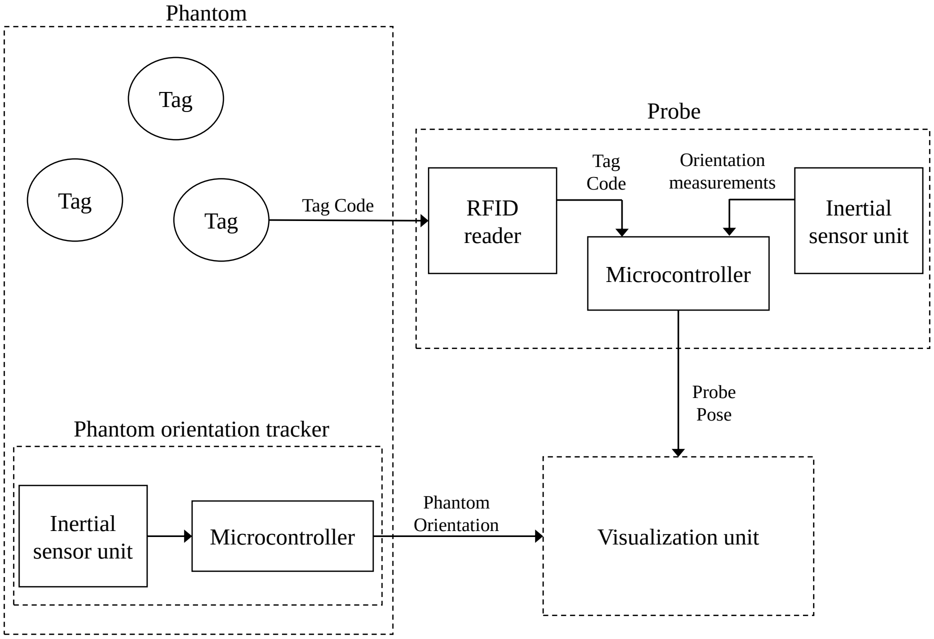

Figure 2: The scheme of the overall system setup.

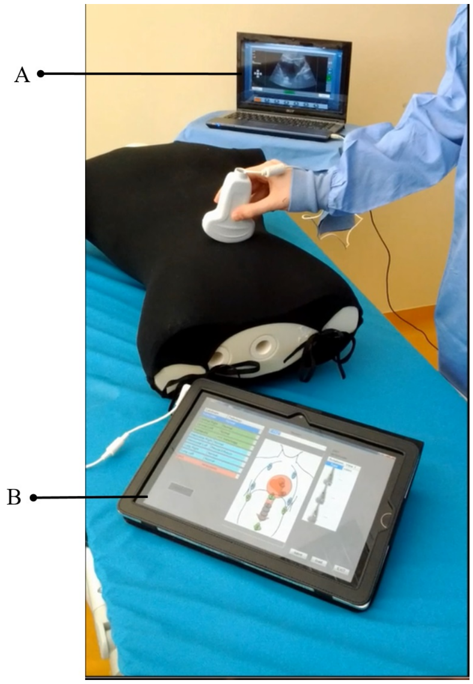

Figure 3: A simulation session with the trainee's (A) and the trainer's (B) applications.

An important feature implemented in the latest software version is the automatic disturbance compensation. Indeed, when a new data set is inserted into the database, the user has to specify the desired probe orientation, associated to that data set, in terms of elemental probe rotations. Unfortunately, often the measured orientation differs from the ideal one, because of the presence of magnetic disturbance sources located inside the phantom itself, as speakers or batteries. However, several experiments on magnetic disturbances showed that they can be considered as constant offsets around the target position marked by the RFID tags (see Section 4.3). Therefore, the user can acquire the actual target orientation exploiting the probe and a compensation rotation matrix can be created by comparing the measurements and the ideal values (it is the rotation required to align the measured and the desired orientation). The matrix is finally stored into the database as a landmark property, so that the visualization system can real-time perform the compensation by applying the constant rotation specified by the matrix to the current estimated orientation.

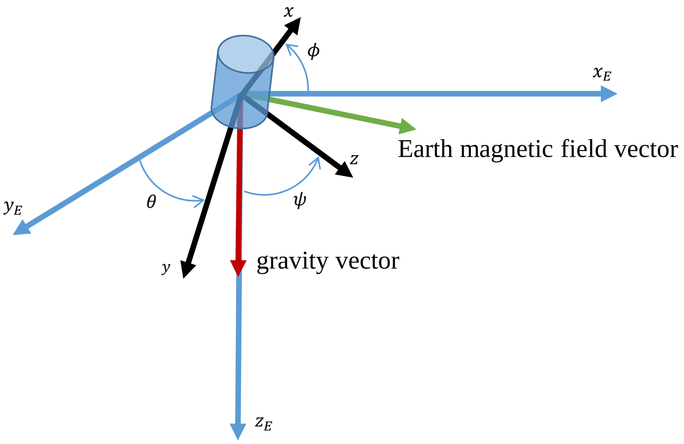

Figure 4: The representation of the orientation of a moving rigid body in the Earth Reference System.

3. Theory and calculation

3.1. Measuring the orientation from inertial sensors

In the context of low-g quasi-static applications the orientation can be measured by means of inertial sensors, using the accelerometer as a gravitometer and including the magnetometer data. In the following a brief summary on how to obtain nautical angles from inertial measurements is reported.

With reference to Figure 4, the Earth Reference System (ERS) is represented as a fixed left-handed triplet of orthogonal axes x_E, y_E, z_E pointing respectively to the North Pole, the East, and the gravity down direction. Similarly, the Body Reference System (BRS) is understood as a mobile triplet of left-handed orthogonal axes x, y, z. In this mainframe, the nautical angles roll (ϕ), pitch (θ), yaw (ψ), represent the rotation about each of the three BRS axes required to align that reference system to the ERS. An indirect measure of such angles can be calculated from accelerometers and magnetometers measurements, as follows:

$$\begin{aligned}\phi_{am} &= \arctan(\frac{a_y}{a_z}) \\ \theta_{am} &= \arctan(\frac{-a_x}{\sqrt{a_y^2 + a_z^2}}) \\ \psi_{am} &= \arctan(\frac{m_z \sin\phi - m_y \cos\phi}{m_x \cos\theta + m_y \sin\theta \sin\phi + m_z \cos\phi \sin\theta})\end{aligned} \tag{1}$$

Where a_x, a_y, a_z are the accelerations along the three axes of the BRS, under low-g motion hypothesis, while m_x, m_y, m_z are the Earth magnetic field vector components, measured in the BRS, pointing the North Pole when no disturbance affects the system. In order to return the appropriate angular value, the $\arctan(y/x)$ function has to be correct when $x < 0$ by adding the value of $\operatorname{sign}(y)\pi$.

It is worth observing that the yaw angle computation requires an alignment procedure that exploits the roll and pitch angles.

The obtained values represent a static (or low-g) estimation of the orientation of the rigid body and they are calculated by means of trigonometric functions applied to the accelerometer and the magnetometer data. However, this estimation can be too noisy for the requirement of the application and the dynamic behavior can be poorly tracked.

On the other hand, three-axis gyroscopes can provide a dynamic estimation of nautical angles ϕ_g, θ_g, ψ_g . Indeed, they measure the angular rates g_x, g_y, g_z of the three BRS axes. If the current nautical angles ϕ_g, θ_g, ψ_g are known, the gyroscope measurements can be transformed into the nautical angle rates $\dot{\phi}_g$, $\dot{\theta}_g$, $\dot{\psi}_g$ by means of the following expressions:

$$\begin{aligned}\dot{\phi}_g &= g_x + g_y \sin(\phi_g)\tan(\theta_g) + g_z \cos(\phi_g)\tan(\theta_g) \\ \dot{\theta}_g &= g_y \cos(\phi_g) g_z \sin(\phi_g) \\ \dot{\psi}_g &= g_y \sin(\phi_g)/\cos(\theta_g) + g_z \cos(\phi_g)/\cos(\theta_g)\end{aligned} \tag{2}$$

Then, assuming the knowledge of the initial conditions, Equation (2) can be integrated obtaining the time evolution of ϕ_g, θ_g, ψ_g. However, the numerical integration of signals with high noise levels and time varying additive biases is always characterized by drift errors, that prevent the correct estimation of the orientation, so that a corrective procedure is required.

Therefore, the main idea is to elaborate the static and the dynamic estimation of the orientation through a filter algorithm that fuses together the data provided by the inertial sensors.

3.2. Tracking algorithms

In the following three sensor fusion algorithms are presented. It will be shown how the acquired measurements are processed, and fused together, in order to provide the final estimation of the orientation.

The first algorithm is a Kalman Filter (KF), a well known technique to estimate the state of a dynamic system, taking into account the statistical information about the disturbances affecting the system and the measurements. The second solution deals with a Nonlinear Complementary Filter (NCF). It combines the static and the dynamic estimations followed by filtering procedures that operate at different frequencies. Finally, the third algorithm proposes a Quaternion-Based Filter (QBF) that involves the quaternion kinematic properties, described in detail in Appendix A.

Kalman filter

In recent years many studies proposed the use of KF and Extended Kalman Filter (EKF) in order to track the motion of a body, by fusing the acquisition from the inertial sensors (Ali and Hassan, 2014; Sabatini, 2006). The simple KF model adopted in this study is a suitable starting point for the tracking system comparison. The main idea is to integrate the gyroscope measurement over the time in the prediction step, and to update the estimation in the correction step by using the static measurements from the accelerometer and the magnetometer. The state $\mathbf{x}$ of the system collects, besides the three nautical angles, the bias affecting their angular rate, namely $b_{\dot{\phi}}$, $b_{\dot{\theta}}$, $b_{\dot{\psi}}$. These disturbances, due to the gyroscope noisy measurements, are difficult to estimate in an analytical way, because of their temperature dependence. The input $\mathbf{u}$ of the system contains the nautical angle rates $\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$ coming from the gyroscope measurements through the conversion of Equation (2). The KF measurements $\mathbf{y}$ is represented by ϕ, θ, ψ as expressed by Equation (3):

$$\mathbf{x} = \begin{bmatrix} \phi \\ \theta \\ \psi \\ b_{\dot{\phi}} \\ b_{\dot{\theta}} \\ b_{\dot{\psi}} \end{bmatrix} \mathbf{u} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \mathbf{y} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \tag{3}$$

The general KF scheme assumes the following model equations:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k \tag{4}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \tag{5}$$

Where:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & -\Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & -\Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & -\Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$\mathbf{w}$ is the process noise with covariance matrix $\mathbf{Q}$, $\mathbf{v}$ is the measurement noise with covariance matrix $\mathbf{R}$, Δt is the sample time and the subscript k indicates the current sample.

According to KF algorithm the final estimation of the state $\hat{\mathbf{x}}$ is performed by means of two steps, namely the prediction and the correction. The former exploits the state evolution law and produces the predicted estimation $\hat{\mathbf{x}}^-$ through the dynamic integration (6) of the angular rate acquired from the gyroscope, after its conversion into the nautical angle rate:

$$\begin{aligned} \hat{\mathbf{u}}_k &= \begin{bmatrix} \dot{\phi}_g & \dot{\theta}_g & \dot{\psi}_g \end{bmatrix}_k^T \\ \hat{\mathbf{x}}_{k+1}^- &= \mathbf{A}\hat{\mathbf{x}}_k + \mathbf{B}\hat{\mathbf{u}}_k \\ \mathbf{P}_{k+1}^- &= \mathbf{A}\mathbf{P}_k\mathbf{A}^T + \mathbf{Q} \end{aligned} \tag{6}$$

Where $\mathbf{P}$ is the error covariance matrix and $\mathbf{P}^-$ is its predicted value.

Afterwards, the correction step exploits the relations between the current state and the indirect measurements coming from the accelerometer and the magnetometer, as in (1). These relations can be expressed as:

$$\hat{\mathbf{y}}_k = \begin{bmatrix} \phi_{am} & \theta_{am} & \psi_{am} \end{bmatrix}_k^T \tag{7}$$

Finally, the correction is performed by means of the following equations:

$$\begin{aligned}\mathbf{K}_k &= \mathbf{P}^-_{k+1}\mathbf{C}^T(\mathbf{C}\mathbf{P}^-_{k+1}\mathbf{C}^T + \mathbf{R}^{-1})\\ \hat{\mathbf{x}}_{k+1} &= \hat{\mathbf{x}}^-_{k+1} + \mathbf{K}_k(\hat{\mathbf{y}}_k - \mathbf{C}\hat{\mathbf{x}}^-_{k+1})\\ \mathbf{P}_{k+1} &= (\mathbf{I} - \mathbf{K}_k\mathbf{C}\mathbf{P}^-_{k+1})\end{aligned} \tag{8}$$

Where **K** is the Kalman gain, **I** is the identity matrix.

It is worth observing that the singularity problems of nautical angles, that affect this algorithm and not the others described in the following, as they are based on different orientation representations, arises when $\theta = 90$ deg, and therefore when $\cos(\theta) = 0$. However, they can easily handled via software by checking, at each filter iteration, the current value of θ inside the estimated state vector $\hat{\mathbf{x}}$ and, in case it is equal to 90 deg, by adding a small value ϵ, not prominent in the context of the application as it is hidden by the measurement noise.

Further details regarding the setup of the filter parameters will be given in Section 4.

Nonlinear complementary filter

A Nonlinear Complementary Filter (NCF) is a sensor fusion algorithm based on two filters characterized by an overall sum of their transfer functions that is constantly equal to one across the whole frequency domain (Hua et al., 2004). It is commonly used in inertial tracking because of its intrinsic capability to merge together the information coming from different sources, operating on different frequency range. Indeed, as pointed out in (Mahony et al., 2008), the accelerometer can be low-pass filtered in order to preserve the static component of the body acceleration due to the gravity. On the other hand, the gyroscope measurements can be high-pass filtered so that the bias of the sensor is attenuated. The proper combination of the filtered signals can produce an increment of the signal to noise ratio. The NCF algorithm considered in this study follows the implementation of (Premerlani and Bizard, 2009) and involves the orthogonality properties of the rotation matrix. It represents a commonly adopted solution for inertial navigation of autonomous vehicles.

The block diagram in Figure 5 shows the elaboration of the data acquired from the inertial sensors.

The first step of the algorithm involves the evolution along time of the estimated rotation matrix $\hat{\mathbf{R}}$, which represents the current orientation of the rigid body. It depends on the angular rate and follows the law:

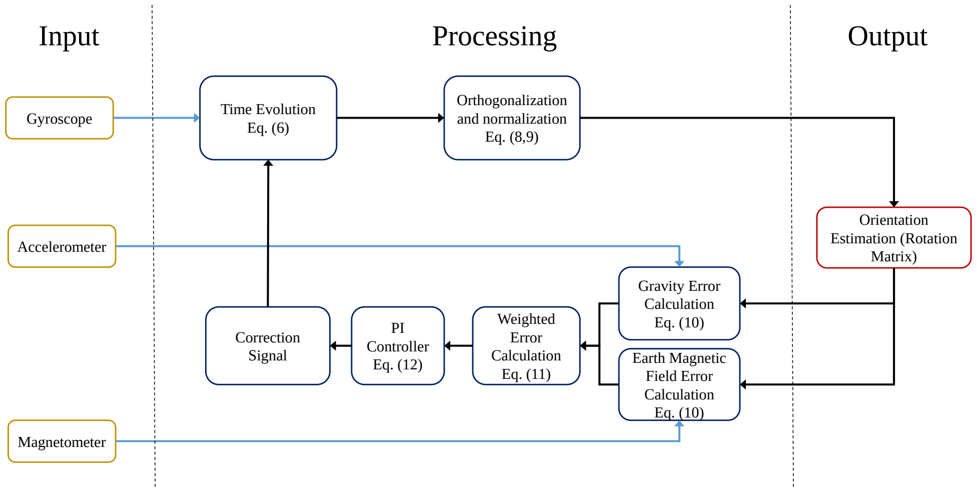

Figure 5: The block diagram of the NCF algorithm.

$$\hat{\mathbf{R}}_{k+1} = \hat{\mathbf{R}}_k \begin{bmatrix} 1 & -\omega_z \Delta t & \omega_y \Delta t \\ \omega_z \Delta t & 1 & -\omega_x \Delta t \\ -\omega_y \Delta t & \omega_x \Delta t & 1 \end{bmatrix} \tag{9}$$

Where ω_x, ω_y, ω_z are the components of the angular rate, after the correction procedure described in the following. Subscripts k are omitted where obvious. Equation (9) derives from the numerical integration of the moving axes of BRS and it is affected by numerical errors and measurement disturbances that are difficult to estimate and produce the drift of orientation estimation. Hence, a correction procedure is required and it involves the orthogonality property of the rotation matrix. Indeed, its three columns **X**,**Y**,**Z** should be normalized and orthogonal with respect to each other. The scalar product err_o between **X** and **Y** can be considered as the orthogonality error between the two columns, because it is zero only when they are orthogonal with respect to each other.

$$err_o = \mathbf{X} \cdot \mathbf{Y} \tag{10}$$

It can be demonstrated (Premerlani and Bizard, 2009) that, if the norm of each column is close to one, the orthogonality error is greatly reduced by equally dividing the error between $\mathbf{X}$ and $\mathbf{Y}$. Therefore, three approximately orthogonal columns $\mathbf{X}_o, \mathbf{Y}_o, \mathbf{Z}_o$ are obtained as follows:

$$\begin{aligned}\mathbf{X}_o &= \mathbf{X} - \frac{err_o}{2}\mathbf{Y} \\ \mathbf{Y}_o &= \mathbf{Y} - \frac{err_o}{2}\mathbf{X} \\ \mathbf{Z}_o &= \mathbf{X}_o \times \mathbf{Y}_o\end{aligned} \tag{11}$$

Where the third column $\mathbf{Z}_o$ is the cross product of $\mathbf{X}_o$ and $\mathbf{Y}_o$.

Afterwards, the resulting columns can be re-normalized in $\mathbf{X}_N, \mathbf{Y}_N, \mathbf{Z}_N$ exploiting the Taylor expansion, instead of dividing by the norm, so that a lower computational effort is required:

$$\begin{aligned}\mathbf{X}_N &= \frac{1}{2}(3 - \mathbf{X}_o \cdot \mathbf{X}_o)\mathbf{X}_o \\ \mathbf{Y}_N &= \frac{1}{2}(3 - \mathbf{Y}_o \cdot \mathbf{Y}_o)\mathbf{Y}_o \\ \mathbf{Z}_N &= \frac{1}{2}(3 - \mathbf{Z}_o \cdot \mathbf{Z}_o)\mathbf{Z}_o\end{aligned} \tag{12}$$

Indeed, considering that the norm of $\mathbf{X}_o$, $\mathbf{Y}_o$, $\mathbf{Z}_o$ is already close to one, the higher order terms of the expansion can be neglected (Premerlani and Bizard, 2009).

Once the current rotation matrix has been orthogonalized and normalized the orientation estimate can be extracted in form of nautical angles $\hat{\phi}$, $\hat{\theta}$, $\hat{\psi}$ from the rotation matrix. Afterwards, the feedback step can start from a comparison between the rotation matrix and the static estimation of the orientation provided by the accelerometer and the magnetometer. Indeed, the third row elements R_{31}, R_{32}, R_{33} of the matrix represent the three components of the gravity acceleration from the BRS point of view. This vector can be compared to the accelerometer acquisition by means of the cross product. Similarly, the elements R_{11}, R_{12} of the rotation matrix represent the components of the North Pole direction vector form the BRS point of view, after the alignment to the horizontal plane, and they can be compared to the magnetometer measurements after the roll and pitch compensation.

$$
\begin{aligned}
\mathbf{err_g} &= \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \times \begin{bmatrix} R_{31} \\ R_{32} \\ R_{33} \end{bmatrix} \\
\mathbf{err_m} &= \begin{bmatrix} m_x \cos\hat{\theta} + m_y \sin\hat{\theta}\sin\hat{\phi} + m_z \cos\hat{\phi}\sin\hat{\theta} \\ m_z \sin\hat{\phi} - m_y \cos\hat{\phi} \\ 0 \end{bmatrix} \times \begin{bmatrix} R_{11} \\ R_{12} \\ 0 \end{bmatrix}
\end{aligned}
\tag{13}
$$

$$
\mathbf{err}_w = w_g \mathbf{err}_g + w_m \mathbf{err}_m \tag{14}
$$

Then, the weighted sum in (14) resulting from the linear combination of the two error vectors (13), by means of the design parameters w_g, w_m, becomes the input of a PI controller which provides the final correction signal $\boldsymbol{\omega}_{\mathrm{cor}_k}$, at the k-th time step, to apply to the next gyroscope measurements:

$$
\begin{aligned}
\mathbf{P}_{\mathrm{cor}_k} &= K_p \mathbf{err}_{w_k} \\
\mathbf{I}_{\mathrm{cor}_k} &= \mathbf{I}_{\mathrm{cor}_{k-1}} + K_I \mathbf{err}_{w_k} \Delta t \\
\boldsymbol{\omega}_{\mathrm{cor}_k} &= \mathbf{P}_{\mathrm{cor}_k} + \mathbf{I}_{\mathrm{cor}_k}
\end{aligned}
\tag{15}
$$

Where K_p and K_I are the PI proportional and integral gains.

Finally, the current correction signal is subtracted to the new gyroscope acquisitions so that another iteration can start.

$$
\boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_{g_{k+1}} - \boldsymbol{\omega}_{\mathrm{cor}_k} \tag{16}
$$

Where $\boldsymbol{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ is the corrected angular rate vector and $\boldsymbol{\omega}_g = \begin{bmatrix} g_x & g_y & g_z \end{bmatrix}^T$ is the gyroscope measurement vector. Further details about the tuning of the filter will be given in Section 4.

Quaternion-based Filter

Alongside the rotation matrix and nautical angles, quaternions are a commonly used representation of the orientation of a rigid body. They consist of a vector of four elements understood as a three-dimensional axis and the rotation, along that axis, required to align the BRS to the ERS. They involve less parameters than those of the rotation matrix and they do not introduce the singularity problem of nautical angles. Therefore, in recent years, many quaternion-based tracking algorithms have been developed. In (Madgwick, 2010; Madgwick et al., 2011), the dynamic evolution of the orientation, elaborated by means of the gyroscope acquisitions, is fused with the static estimation, provided by the accelerometer-magnetometer measurements. The proposed solution exploits the weighted sum of the dynamic and the static quaternion estimations, namely $\hat{\mathbf{q}}_{d,k}$ and $\hat{\mathbf{q}}_{s,k}$:

$$\hat{\mathbf{q}}_k = w_d \hat{\mathbf{q}}_{d,k} + w_s \hat{\mathbf{q}}_{s,k} \tag{17}$$

Where the weight w_d,w_s are design parameter and $\hat{\mathbf{q}}_k = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T$ is the final quaternion estimation, at the k-th instant.

Note that Equation (17) indicates that also this algorithm can be considered a complementary filter, since low and high frequency measurements are combined into a weighted sum. The derivation of $\hat{\mathbf{q}}_{d,k}$ follows the quaternion evolution law, described by the equation:

$$\hat{\mathbf{q}}_{d,k} = \hat{\mathbf{q}}_{k-1} + \frac{1}{2}\Delta t(\hat{\mathbf{q}}_{k-1} \otimes \boldsymbol{\omega}_{g_k}) \tag{18}$$

Where $\boldsymbol{\omega}_g = \begin{bmatrix} 0 & g_x & g_y & g_z \end{bmatrix}^T$ is the angular rate quaternion, containing the gyroscope measurements, and the symbol $\otimes$ indicates the quaternion product.

On the other hand, the calculation of $\hat{\mathbf{q}}_{s,k}$ involves the solution of the optimization problem described by Equation (20). In this case, the error function $\mathbf{f}$, to be minimized, consists of two terms, namely $\mathbf{f}_a$ and $\mathbf{f}_m$. The former, $\mathbf{f}_a$, relies on the difference between the gravity vector $\mathbf{a}$ measured in the BRS and the same vector $\mathbf{g}$ viewed from the ERS, compensated through the quaternion alignment. Similarly, $\mathbf{f}_m$ is given by the difference between the BRS Earth magnetic field vector $\mathbf{m}$ and the ERS magnetic vector $\mathbf{M}$, after its alignment to BRS. These error functions are expressed as:

$$\begin{aligned} \mathbf{f}_a(\mathbf{q},\mathbf{a}) &= \mathbf{q}^* \otimes \mathbf{g} \otimes \mathbf{q} - \mathbf{a} \\ \mathbf{f}_m(\mathbf{q},\mathbf{m}) &= \mathbf{q}^* \otimes \mathbf{M} \otimes \mathbf{q} - \mathbf{m} \\ \mathbf{f} &= \begin{bmatrix} \mathbf{f}_a \\ \mathbf{f}_m \end{bmatrix} \end{aligned} \tag{19}$$

Where $\mathbf{g} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$ is the normalized gravity quaternion in ERS, $\mathbf{a} = \begin{bmatrix} 0 & a_x & a_y & a_z \end{bmatrix}^T$ contains the accelerometer acquisitions, $\mathbf{M} = \begin{bmatrix} 0 & b_x & 0 & b_z \end{bmatrix}^T$

is the normalized Earth magnetic field quaternion, $\mathbf{m} = \begin{bmatrix} 0 & m_x & m_y & m_z \end{bmatrix}^T$ contains the magnetometer acquisitions, and the superscripts * indicates the quaternion conjugation.

Then, the optimization problem, solved by exploiting the gradient-descent algorithm, can be formulated as:

$$\hat{\mathbf{q}}_{s,k} = \arg\min_{\mathbf{q}} |\mathbf{f}(\mathbf{q}, \mathbf{a}, \mathbf{m})| \tag{20}$$

The gradient-descent method commonly uses more iterations per step, however, in this case only one iteration per sample is executed, in order to save the computational time between two consecutive time steps. Hence, the resulting static estimation is:

$$\hat{\mathbf{q}}_{s,k} = \hat{\mathbf{q}}_{k-1} - \mu \frac{\nabla \mathbf{f}}{|\nabla \mathbf{f}|} \tag{21}$$

Where μ is a design parameter and $\nabla \mathbf{f} = \mathbf{J}^T(\hat{\mathbf{q}}_{k-1})\mathbf{f}$, with $\mathbf{J}$ the Jacobian of the error function $\mathbf{f}$.

4. Experimental setup and results

In order to evaluate and compare the performances of the three proposed tracking algorithms, the simulator probe has been fixed to the hand-tool of a PUMA-260 robot so that the inertial sensor reference system is aligned to the robot reference system. Figure 6 shows the setup of the platform.

The first experiment tests large probe movements. They are common when the user has to switch the visualization from a view to another one, e.g. the switching between a two-chambers and a four-chambers view of the heart. The trajectory assigned to the hand-tool of the robot performs three sequential movements along the three Cartesian axes. In particular, the first movement concerns the yaw angle: the probe has an initial orientation $\mathbf{\Theta}_0 = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$ deg, and reaches the orientation $\mathbf{\Theta}_1 = \begin{bmatrix} 0 & 0 & -45 \end{bmatrix}$ deg. Afterwards, the probe comes back to the initial orientation with a specular trajectory. Then, a rotation involving the roll angle is performed, in which the probe reaches the orientation $\mathbf{\Theta}_2 = \begin{bmatrix} 45 & 0 & 0 \end{bmatrix}$ deg, and comes back to $\mathbf{\Theta}_0$. The last movement is related to the pitch angle, as the probe rotates to $\mathbf{\Theta}_3 = \begin{bmatrix} 0 & 45 & 0 \end{bmatrix}$ deg and returns to $\mathbf{\Theta}_0$. The mean movement rate

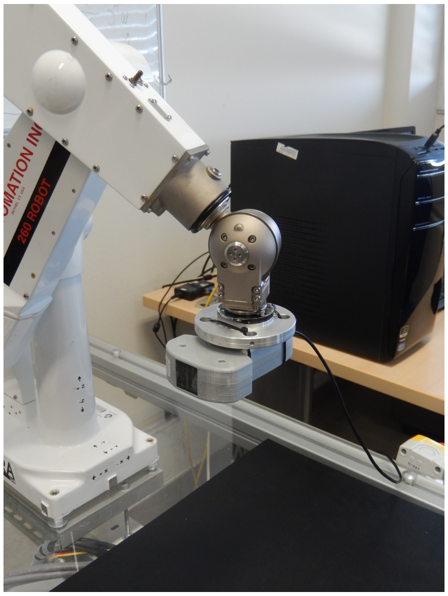

Figure 6: The simulator probe fixed to the robot.

is about 50 deg/s for the roll angle rotation, while it is about 90 deg/s for pitch and yaw rotations.

In the meanwhile the probe microcontroller acquires the noisy raw measurements from the inertial sensors and sends the data via serial bus to a laptop computer, where they are stored allowing the off-line elaboration and the evaluation of the performances. On the other hand, also the robot joint positions are acquired in order to reconstruct the actual orientation of the tool (and consequently of the probe) exploited as the target orientation for the tracking algorithm evaluation. The robot signal is acquired at a frequency of 1 kHz, while the inertial sensor unit provides data at 100 Hz. The synchronization between the two signals is achieved via software, by getting the absolute clock time at each acquisition.

The adopted design parameters for the three filters are reported in the following. Regarding the KF, the state and the output covariance matrices, namely $\mathbf{Q}$, $\mathbf{R}$ are difficult to estimate a priori because of the unknown statistics description of the system disturbances. Therefore they can be considered as design parameters. The values assumed by the matrices during the experiment are:

$$\mathbf{Q} = \begin{bmatrix} 30 & 0 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 0 & 0 & 0 \\ 0 & 0 & 30 & 0 & 0 & 0 \\ 0 & 0 & 0 & 40 & 0 & 0 \\ 0 & 0 & 0 & 0 & 40 & 0 \\ 0 & 0 & 0 & 0 & 0 & 40 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$$

Then, the tuning of the NCF involves the choice of the proportional and integral gains of the PI regulator and the error weights. The values used in the experiment are:

$$K_P = 15$$

$$K_I = 1$$

$$W_g = 10^{-4}$$

$$W_m = 10^{-3}$$

Finally, regarding the QBF, the gradient descent convergence rate value μ is chosen as:

$$\mu = 3$$

While the static and the dynamic quaternion weights are:

$$w_s = 0.6$$

$$w_m = 0.4$$

The second experiment involves small probe movements, as they are common when the user adjusts the probe to find the best visualization of a particular view. In this case, the hand-tool trajectory has been planned to perform the simultaneous rotation of the roll and the pitch angles of 5 deg in both the positive and the negative direction, at a rate of about 5 deg/s. The initial orientation has been set to $\mathbf{\Theta}'_0 = \begin{bmatrix} -45 & 45 & 0 \end{bmatrix}$ deg. Then, the probe passes sequentially through the first goal orientation $\mathbf{\Theta}'_1 = \begin{bmatrix} -40 & 40 & 0 \end{bmatrix}$ deg, the initial orientation, the second goal orientation $\boldsymbol{\Theta}_2' = \begin{bmatrix}-50 & 50 & 0\end{bmatrix}$ and, finally, the initial orientation. When the probe reaches a trajectory point, it pauses for a short time, in order to simulate a typical user movement while adjusting the probe to find the best visualization. The setup of the filters is the same of the first experiment.

4.1. Results of the large movement test

The first experiment involves large probe movements. For each of the three algorithms, and for each of the three nautical angles, three performance indices are calculated. They are the Root Mean Squared Error (RMSE) and the variance of the estimation error signal during the static phase, as they give an idea of the accuracy of the estimation, and the settling time during the transients, as it measures the readiness of the tracking. The settling time is defined as the time elapsed from the instant in which the reference reaches a steady-state value to the instant in which the considered filter signal reaches the 95% of the mean value of the subsequent static condition. Furthermore, the real-time mean computational time has been evaluated by implementing the algorithms on the microcontroller. This parameter indicates the computational effort required by the algorithms to the microprocessor. Table 2 summarizes the results of the experiments.

It is worth noting that RMSE and variance indices, for a particular nautical angle, are calculated when that estimation signal has reached the steady condition, after the movement relative to the considered angle has been performed. Similarly, the settling time, is calculated as the difference between the instant in which the target signal reaches the steady-state condition and the instant in which the estimation signal concludes the transient (the 95% of the steady condition has been considered).

Figures 7 to 9 show the tracking of the reference orientation signals performed through the three algorithms, that involves the transient of the roll, the pitch and the yaw angle estimations, respectively. Furthermore, Figure 10 depicts the tracking errors in steady condition associated to the roll angle estimation, for each of the three algorithms.

4.2. Results of the small movement test

The second experiment deals with small probe movements. The filter parameters are the same of the previous experiment. The considered performance indices (i.e. RMSE and variance) are reported in Table 3. They are the mean values of the indices that have been computed during the static

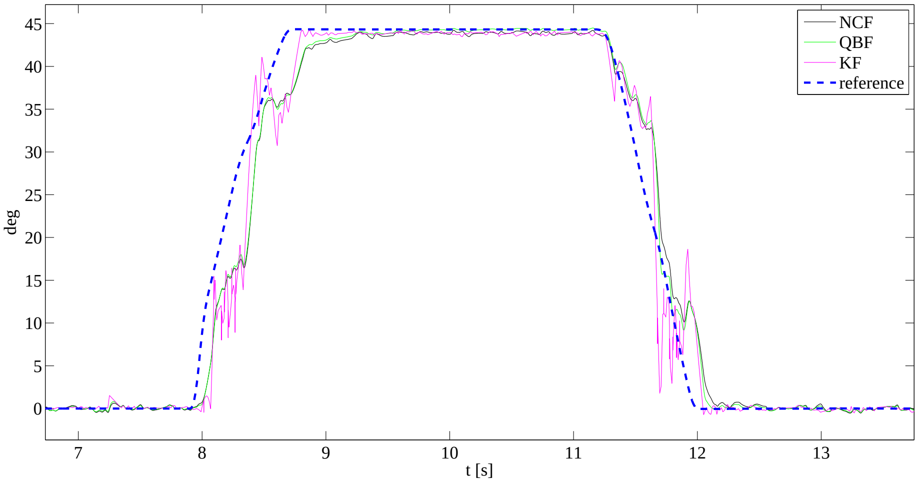

Figure 7: The comparison among algorithms considering the transient of the roll angle tracking.

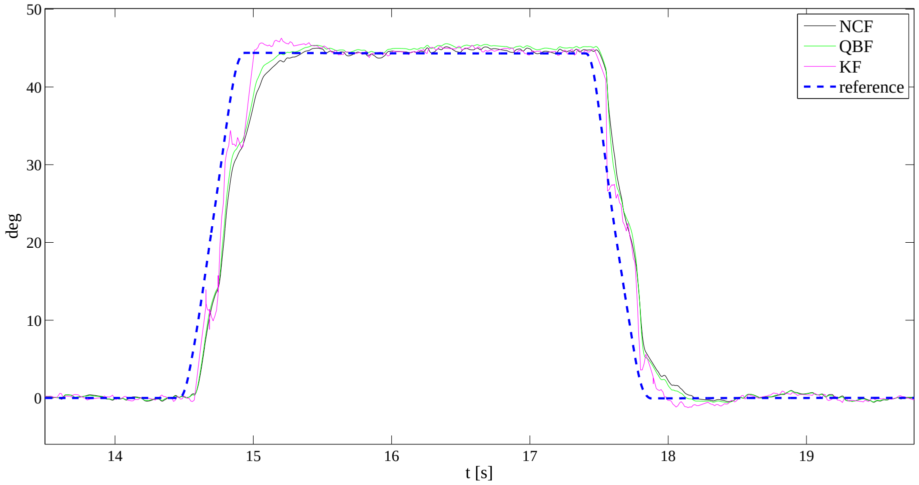

Figure 8: The comparison among algorithms considering the transient of the pitch angle tracking.

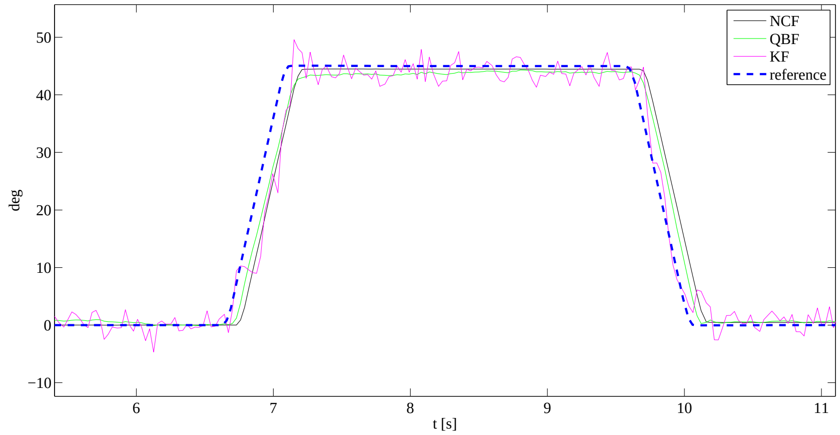

Figure 9: The comparison among algorithms considering the transient of the yaw angle tracking.

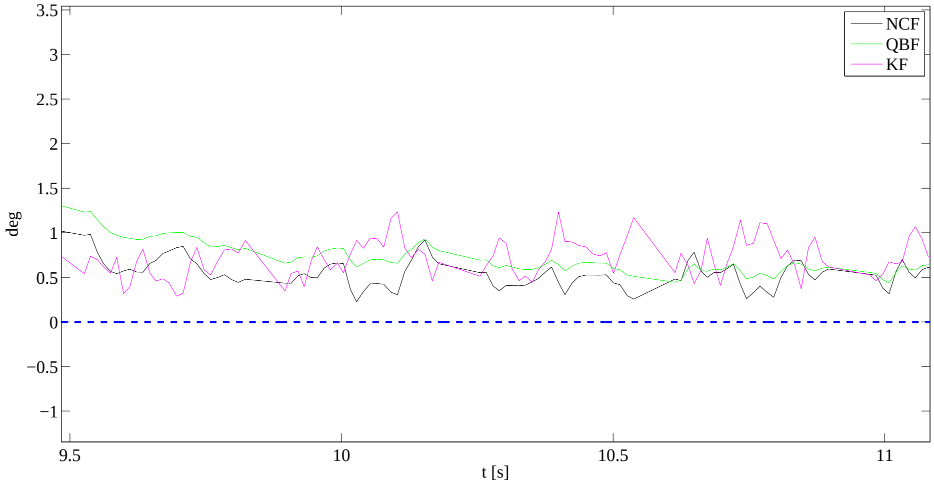

Figure 10: The tracking error after the transient of the roll angle estimation.

Parameter		KF	
	$\phi - \phi_{\text{ref}}$	$\theta - \theta_{\text{ref}}$	$\psi - \psi_{\text{ref}}$
RMSE (deg)	1.16	0.44	1.78
Variance (deg^2)	0.073	0.091	3.24
Settling time (s)	0.030	0.034	0.099
Computational time (μs)	6274 ± 12		
		NCF	
	$\phi - \phi_{\text{ref}}$	$\theta - \theta_{\text{ref}}$	$\psi - \psi_{\text{ref}}$
RMSE (deg)	1.11	0.56	0.64
Variance (deg^2)	0.014	0.049	0.078
Settling time (s)	0.078	0.078	0.169
Computational time (μs)	5438± 10		
		QBF	
	$\phi - \phi_{\text{ref}}$	$\theta - \theta_{\text{ref}}$	$\psi - \psi_{\text{ref}}$
RMSE (deg)	1.09	0.40	1.51
Variance (deg^2)	0.013	0.032	0.109
Settling time (s)	0.071	0.071	0.152
Computational time (μs)	5438± 15		

Table 2: The performance indices computed for the large movement experiment.

phases. Figure 11, 12 depict the tracking of the roll and pitch reference signals.

4.3. Analysis of magnetic disturbances

All of the proposed tracking algorithms rely on magnetometer reading, that could be affected by external sources of disturbance. Therefore the following experiment, carried out on a phantom that presents strong magnetic disturbances, investigates the influence of these disturbances on the considered tracking methods.

The probe has been positioned on the phantom thorax, and aligned to its reference system so that the actual relative orientation is equal to $\mathbf{\Theta}_{\text{ref}} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$ deg. Then three complete rotations involving the yaw angle have been carried out, stopping temporarily the movement when the probe reached the $\mathbf{\Theta}_{\text{ref}}$ orientation (marked on the phantom skin). In the meanwhile the raw sensor data have been acquired.

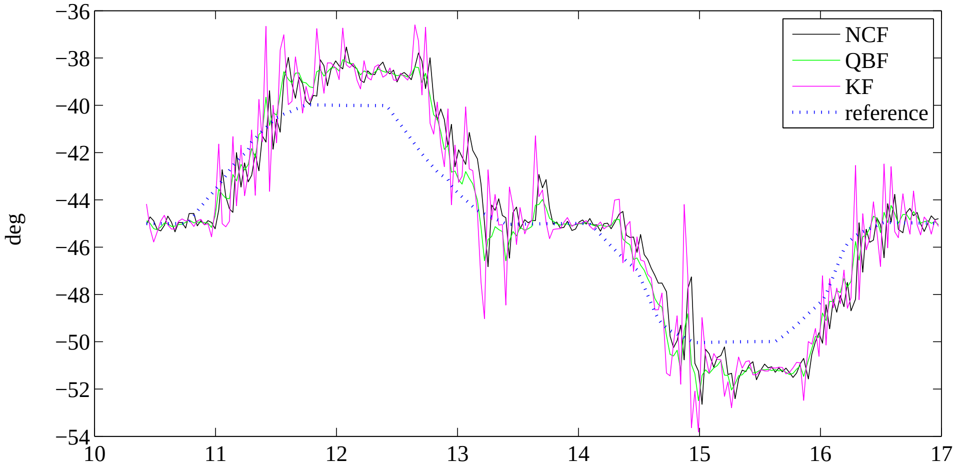

Figure 11: The comparison among algorithms considering small movements of the roll angle.

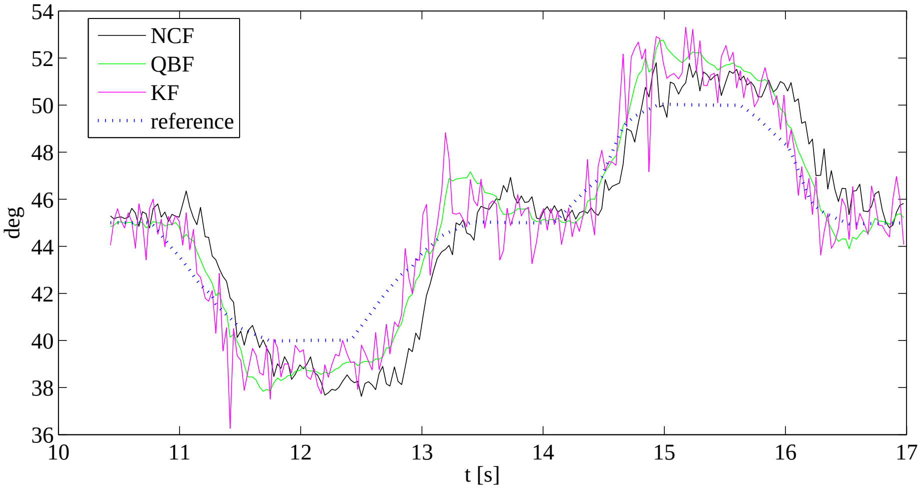

Figure 12: The comparison among algorithms considering small movements of the pitch angle.

Parameter	KF	
	$\phi - \phi_{\text{ref}}$	$\theta - \theta_{\text{ref}}$
RMSE (deg)	1.20	1.08
Variance (deg^2)	1.112	1.280
	NCF	
	$\phi - \phi_{\text{ref}}$	$\theta - \theta_{\text{ref}}$
RMSE (deg)	1.10	1.10
Variance (deg^2)	0.091	0.094
	QBF	
	$\phi - \phi_{\text{ref}}$	$\theta - \theta_{\text{ref}}$
RMSE (deg)	1.11	1.08
Variance (deg^2)	0.075	0.069

Table 3: The performance indices computed for the small movement experiment.

Then, the three algorithms have been off-line elaborated to provide the orientation estimations $\hat{\boldsymbol{\Theta}}$. Afterwards, the compensation procedure described in Section 2.2 has been applied to the reconstructed orientation, obtaining the compensated estimation $\hat{\boldsymbol{\Theta}}_c$. Table 4 show, for each algorithm, the RMSE and the variance of the yaw angle estimations, computed with and without the compensation procedure, during the motionless phase after each of the three rotations, when the estimations reach the steady condition. In the table only the yaw estimation has been reported, as it has been observed that the magnetic disturbances does not significantly affect the roll and pitch estimations. The obtained values show that the RMSE is significantly lower after the compensation procedure, supporting the assumption that magnetic disturbances act mainly as a constant offset, for all the proposed tracking algorithms.

Furthermore, it has been observed that the effect of magnetic disturbances does not significantly vary for small changes in probe position, so that the magnetic offset to be compensated remains constant in a sphere of 5 cm diameter, centered in the RFID tag target position. Outside that sphere the probe is no more in the tag read range and without that information the visualization software does not exploit the probe orientation estimate at all.

Parameter	KF		NCF		QBF	
	$\boldsymbol{\Theta}_{\mathrm{ref}} - \hat{\boldsymbol{\Theta}}$	$\boldsymbol{\Theta}_{\mathrm{ref}} - \hat{\boldsymbol{\Theta}}_c$	$\boldsymbol{\Theta}_{\mathrm{ref}} - \hat{\boldsymbol{\Theta}}$	$\boldsymbol{\Theta}_{\mathrm{ref}} - \hat{\boldsymbol{\Theta}}_c$	$\boldsymbol{\Theta}_{\mathrm{ref}} - \hat{\boldsymbol{\Theta}}$	$\boldsymbol{\Theta}_{\mathrm{ref}} - \hat{\boldsymbol{\Theta}}_c$
RMSE (deg)	75.55	2.78	74.09	1.94	74.30	1.59
Variance (deg^2)	3.10	3.19	0.091	0.089	0.12	0.12

Table 4: The results of the experiment on the influence of magnetic disturbance on the proposed tracking algorithms.

5. Discussion

The results described in the previous section yield some considerations about the choice of the algorithm to implement on the simulator.

The 3D views have the most strictly requirements in terms of tracking the movement of the simulator probe. Indeed, the accuracy of the estimated orientation should be high enough to permit the execution of the proper volume slice. RMSE of about 3 deg can be detected by users. Moreover, in steady state the signal coming from the probe, should be constant enough to generate a stable image. It has been observed that excessive noise of the orientation signal, i.e. with variance greater than 1 deg^2) can cause undesired image flicker. In addition, the readiness of the tracking should be fast, and should not introduce delays higher than 0.3 s, in order to guarantee a real-time simulation.

The results of the experiments have shown that all the considered filters provide an accurate estimation, measured in terms of RMSE in steady condition. The computed RMSE is about 1 deg for all the filters, considering both large and small movements.

The requirement of stability is achieved satisfactorily by NCF and QBF, indeed the variance of the error signal is always less than 0.1 deg^2, while KF produces a yaw estimation characterized by high noise. Indeed, both the experiments report a KF variance higher than 1 deg^2.

Finally, all the proposed algorithm achieve a fast tracking of the orientation, as the measured settling time during the transient is always less than 0.3 s. These considerations suggest the choice of NCF or QBF rather than KF, since they meet all the application requirements and because of their lower computational effort.

It is worth noting that the parameters of the filters have been empirically tuned taking into account the trade-off between the variance of the estimation and their settling time. In particular, regarding the NCF, the readiness of the response can be increased by exploiting higher values of K_P. On the other hand this increment leads to an undesired growth of the estimation noise. A more stable, but slower, response can be obtained by increasing the K_I parameter. The tuning of the convergence rate μ of the QBF involves considerations similar to those discussed for the K_p parameter of the NCF.

Furthermore, a large number of tests has been carried out regarding the tuning of the KF parameters, on the basis of the well known properties of the KF covariance matrices, see e.g. (Åkesson et al., 2008). The results did not show a significant improvement of the performance so that the accuracy and the noise on the yaw estimations never met the application requirement. In particular, further tuning tests show that a smoother Kalman filter signal, with a variance of about 1 deg^2 can be obtained by decreasing the state covariance matrix $\mathbf{Q}$ to a value of $\mathbf{Q}' = 10^{-2}\mathbf{Q}$ or equivalently by increasing the measurement covariance matrix $\mathbf{R}$ to a value of $\mathbf{R}' = 10^{2}\mathbf{R}$. However, in both cases, the settling time becomes higher than 0.5 s and does not meet the application requirements.

6. Conclusions

The developed US simulator represents a device suitable for medical training and education, particularly in Point of Care Ultrasonography. Hardware and software implementation fulfills the requirement of cheapness and the system is design to be used in high-fidelity simulation session, thanks to the two separated graphical user interfaces (instructor and trainee). The visualization system can manage 2D as well as 3D US data sets, and the database editor allows the user customization and the sharing of materials. The proper functioning of the simulator is based on the real-time inertial tracking of the hand-guided probe. Therefore, the main contribution of this work is the analysis of three tracking algorithms, i.e. a Kalman filter, a nonlinear complementary filter and a quaternion-based filter, that have been tested and compared each other in terms of accuracy, noise and response time during the tracking. The results show that the Kalman filter does not match the discussed requirements of the application, while the other algorithms provide satisfactory performances. Furthermore, since the magnetometer is involved in all the considered algorithms, the effect of magnetic disturbances, mainly due to phantom components, has been investigated. It has been demonstrated that they act as an offset on the orientation tracking, that can be compensated by applying a rotation to the current orientation estimation.

The overall simulator has been exploited during several simulation sessions, held at the University of Ferrara and the Maggiore Hospital of Bologna, Italy and it has obtained positive impression from physicians concerning the user-friendliness and fidelity of the system.

Appendix A. Quaternion properties

A *unit* quaternion $\mathbf{q}$ is a four-dimensional vector with unitary norm, consisting of two subcomponents: a scalar q_w and an inner vector $\mathbf{q}_v = \begin{bmatrix} q_x & q_y & q_z \end{bmatrix}^T$. Unit quaternions (Diebel, 2006) are related to the axis-angle representation of the orientation of a rigid body, in which two subcomponents denote respectively the angle θ and the axis versor $\mathbf{u}$ around which the orientation is defined:

$$\mathbf{q} = \begin{bmatrix} q_w \\ \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} \cos(\frac{\theta}{2}) \\ \mathbf{u}\sin(\frac{\theta}{2}) \end{bmatrix} \tag{A.1}$$

A set of useful operations applied to quaternions can be defined, and they are briefly recalled in the following:

- The product of two quaternion $\mathbf{q}$, $\mathbf{r}$ is given by:

$$\mathbf{q} \otimes \mathbf{r} = \begin{bmatrix} q_w r_w - \mathbf{q}_v \cdot \mathbf{r}_v \\ \mathbf{q}_v \times \mathbf{r}_v + q_w \mathbf{r}_v + r_w \mathbf{q}_v \end{bmatrix} \tag{A.2}$$

- The conjugate $\mathbf{q}^*$ of a quaternion $\mathbf{q}$ is:

$$\mathbf{q}^* = \begin{bmatrix} q_w \\ -\mathbf{q}_v \end{bmatrix} \tag{A.3}$$

- A vector $\mathbf{v} = \begin{bmatrix} 0 & v_x & v_y & v_z \end{bmatrix}^T$ can be rotated by a quaternion $\mathbf{q}$ obtaining the vector $\mathbf{v}'$ as:

$$\mathbf{v}' = \mathbf{q} \otimes \mathbf{v} \otimes \mathbf{q}^* \tag{A.4}$$

Furthermore, the time variation of the orientation is associated to the time derivative of the unit quaternion describing the orientation of the rigid body. The evolution of that quaternion along time is related to the body angular rate by means of the expression:

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \otimes \boldsymbol{\omega} \tag{A.5}$$

where the body angular rate $\boldsymbol{\omega}$ is seen from the body-fixed frame, and it is considered as $\boldsymbol{\omega} = \begin{bmatrix} 0 & \omega_x & \omega_y & \omega_z \end{bmatrix}^T$.

References

Åkesson, B. M., Jørgensen, J. B., Poulsen, N. K., Jørgensen, S. B., 2008. A generalized autocovariance least-squares method for kalman filter tuning. Journal of Process control 18 (7), 769–779.

Ali, N. H., Hassan, G. M., 2014. Kalman filter tracking. International Journal of Computer Applications 89 (9).

Blum, T., Rieger, A., Navab, N., Friess, H., Martignoni, M., 2013. A review of computer-based simulators for ultrasound training. Simulation in Healthcare 8 (2), 98–108.

Diebel, J., 2006. Representing attitude: Euler angles, unit quaternions, and rotation vectors. Matrix 58 (15-16), 1–35.

Farsoni, S., Astolfi, L., Bonfè, M., Spadaro, S., 2015. Design of an ultrasound simulator with probe pose tracking and medical dataset processing and visualization. IFAC-PapersOnLine 48 (20), 377–382.

Gillman, L. M., Ball, C. G., Panebianco, N., Al-Kadi, A., Kirkpatrick, A. W., 2009. Clinician performed resuscitative ultrasonography for the initial evaluation and resuscitation of trauma. Scandinavian journal of trauma, resuscitation and emergency medicine 17 (1), 34.

Hua, W., DeBra, D., Hardham, C. T., Lantz, B. T., Giaime, J. A., 2004. Polyphase fir complementary filters for control systems. In: in Proceedings of ASPE Spring Topical Meeting on Control of Precision Systems. pp. 109–114.

Luinge, H. J., Veltink, P. H., 2005. Measuring orientation of human body segments using miniature gyroscopes and accelerometers. Medical and Biological Engineering and computing 43 (2), 273–282.

Madgwick, S. O., 2010. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. Report x-io and University of Bristol (UK).

Madgwick, S. O., Harrison, A. J., Vaidyanathan, R., 2011. Estimation of imu and marg orientation using a gradient descent algorithm. In: Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on. IEEE, pp. 1–7.

Mahony, R., Hamel, T., Pflimlin, J.-M., 2008. Nonlinear complementary filters on the special orthogonal group. Automatic Control, IEEE Transactions on 53 (5), 1203–1218.

Oxorn, D., Pearlman, A., 2012. Con: physician-performed ultrasound: the time has come for routine use in acute care medicine. Anesthesia & Analgesia 115 (5), 1004–1006.

Parks, A. R., Atkinson, P., Verheul, G., LeBlanc-Duchin, D., 2013. Can medical learners achieve point-of-care ultrasound competency using a high-fidelity ultrasound simulator?: a pilot study. Crit Ultrasound J 5 (1), 9.

Premerlani, W., Bizard, P., 2009. Direction cosine matrix imu: Theory. DIY DRONE: USA, 13–15.

Sabatini, A. M., 2006. Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing. Biomedical Engineering, IEEE Transactions on 53 (7), 1346–1356.

Serrano, D. E., Ayazi, F., 2015. Mems inertial sensors. Resonant MEMS: Fundamentals, Implementation, and Application.

Vignon, P., 2012. Pro: physician-performed ultrasound: the time has come for routine use in acute care medicine. Anesthesia & Analgesia 115 (5), 999–1003.

Welch, G., Foxlin, E., 2002. Motion tracking survey. IEEE Computer graphics and Applications, 24–38.