Low-cost, Fast and Accurate Reconstruction of Robotic and Human Postures via IMU Measurements

Gaspare Santaera [1], Emanuele Luberto[1], Alessandro Serio[1], Marco Gabiccini[2], and Antonio Bicchi[3]

***Abstract*— In this paper, we present a method to reconstruct the configurations of kinematic trees of rigid bodies not using measurements of relative angles (such as, e.g. rotary encoders at joints) but absolute posture sensors (such as IMUs) along with suitable filter algorithms. We argue that the relatively larger inaccuracies shown by absolute sensors can be compensated by suitable processing, such as a passive complementary filters exploiting the Mahony-Hamel formulation. The proposed method is applicable to systems where measurements of relative angles is not feasible or convenient, or where the joint kinematics are not lower pairs: for example, human body parts or soft robotic devices. In the paper, we make explicit reference to the reconstruction of posture of the compliant, underactuated Pisa/IIT SoftHand. Quantitative comparisons with ground truth data in grasping tests are used to validate the proposed method. The resulting hardware design is mechanically robust, cheap and can be easily adapted to robotic hands with different structures, as well as to sensorizing gloves for studying human grasping strategies.**

I. INTRODUCTION

In robotics textbooks and in the vast majority of robotics papers, forward kinematics of serial and tree-like structures is usually formulated as if perfect knowledge of the geometry of (technical) joints connecting the manipulator links is assumed, and the direct measurement of successive joint variables is always possible and can be realized with no effort [1]. Consider, with reference to Fig. 1, a kinematic chain composed only by revolute joints: in this case, like suggested from Fig. 1 (a), it is sufficient to measure the joint configuration $\mathbb{Q} = (\mathbb{S}^1 \times \cdots \times \mathbb{S}^1)$, provide each entry to a pre-defined $SE(3)$ template matrix (which varies according to the parametrization chosen, e.g., D.-H., P.O.E., and so on), and assemble the obtained matrices via standard matrix product to obtain the Cartesian configuration $\mathbb{C} = (SE(3) \times \cdots \times SE(3))$.

In most robotic structures, where the limb geometry is known with high accuracy and the joint kinematics are simple, this is without doubts the most effective approach — optical or magnetic encoders, or potentiometers are employed in these cases.

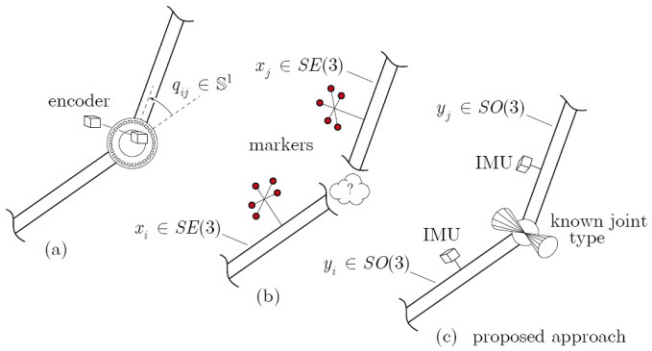


Fig. 1. Sensorization methods: (a) classical joint angle measurements, (b) direct measurements, (c) intermediate method (the one we propose).

A different approach with respect to the one previously described is depicted in Fig. 1 (b), and normally adopted in biomechanics [2]. Here, due to the *essential* non existence of technical joints that connect biological limbs, the only viable method is usually a direct sensorization of each limb (e.g., via optical markers and/or magnetic and inertial devices, "strap down" to the greatest extent to the underlying bone) in sufficient number so that the direct extraction of the $SE(3)$ pose of each limb, assumed as rigid, is performed. In this manner, the Cartesian configuration $\mathbb{C} = (SE(3) \times \cdots \times SE(3))$ is directly obtained. The goal is here, mostly, the definition of the simplest assembly of technical joints that best reproduces the animal or human joint motion [3], and correlate motion patterns to functional status levels for clinical studies [4]. However, high-costs, marker losses due to frequent occlusions (especially when tracking hands) and the need of working in a controlled environment (many cameras are needed) render this method inapplicable if a low-cost, non-invasive system has to be devised.

In this paper, we present an approach to reconstruct the pose of serial kinematic chain that aims at reconciling the two previously described, and could be defined as an intermediate way of tackling the problem, see Fig. 1 (c). Inspired by (b), we sensorize each link of the chain with an Inertial Measurement Unit (IMU), thus estimating its orientation via the corresponding $SO(3)$ attitude matrix. These measurements provide use with the global attitude configuration $\mathbb{C}_a = (SO(3) \times \cdots \times SO(3))$. Then, we exploit a *mild* knowledge about the underlying kinematic topology — mostly the number of DoFs at each joint and the link

[1] G. Santaera, E. Luberto and A. Serio are with Centro di Ricerca "E. Piaggio", Università di Pisa, Largo L. Lazzarino 1, 56122 Pisa, Italy. santaeragaspare at inwind.it, ema.lbt at hotmail.it, a.serio at centropiaggio.unipi.it

[2]M. Gabiccini is with Dipartimento di Ingegneria Civile e Industriale (DICI), Largo Lucio Lazzarino 1, 56122 Pisa, with Centro di Ricerca "E. Piaggio", Università di Pisa, Largo L. Lazzarino 1, 56122 Pisa, and with Department of Advanced Robotics (ADVR), Italian Institute of Technology, Via Morego 30, 16163 Genova, Italy. m.gabiccini at ing.unipi.it

[3]A. Bicchi is with Centro di Ricerca "E. Piaggio", Università di Pisa, Largo L. Lazzarino 1, 56122 Pisa and with Department of Advanced Robotics (ADVR), Italian Institute of Technology, Via Morego 30, 16163 Genova, Italy. bicchi at centropiaggio.unipi.it

Preprint submitted to 2015 IEEE International Conference on Robotics and Automation. Received October 1, 2014.

geometry, while the directions of the joint axes are negotiated with the data — to estimate (solving an inverse problem) $\mathbb{Q}_a = (\mathbb{S}^1 \times \cdots \times \mathbb{S}^1)$ and, finally, arrive at the global $SE(3)$ pose of each link, thus obtaining (like in (a)) the Cartesian configuration $\mathbb{C} = (SE(3) \times \cdots \times SE(3))$.

Even if the methodology presented in this work can be adapted to any tree-like structure, we refer, in the unfolding of the procedure, to the to the problem that motivated us towards this study: reconstructing the posture of a compliantly underactuated robotic hand and, in particular, the hand developed in our lab, the Pisa/IIT SoftHand [5].

In fully sensorized (and actuated) robotic hands, like the DLR-HIT Hand II [6], the UB Hand3 [7], and the iCub Hand [8], direct measurement of each joint angle is possible. However, high cost is paid to this sake in terms of large dimensions, heavy weight, weakness to external shocks or to magnetic fields — in those cases where encoders or potentiometers are employed.

In recent years, a novel design paradigm of compliantly underactuated robotic hands has been developed giving rise, among many others, to the Pisa/IIT Softhand [5]. Even if this hand is pretty robust and easily controllable, on the other side it has no joint position sensors, which could hinder it from widespread use when feedback control is requested. One exemplar case is during interaction with the environment in recognizing or reconstructing the shape of an object when robotic hand fingers are used as haptic probes: this allows to provide a complementary sensory modality to support vision in unstructured and cluttered environments.

In this hand, the complex kinematics (epicycloidal motion) of one phalanx w.r.t. the next, introduced via the modular design of roll-articular joints (directly realized via two geared sectors in direct mesh), and the presence of pulleys, tendons and elastic bands, does not allow to use encoders or similar devices to measure joint angles. This motivated us to find alternative ways to solve the problem.

The same approach can be applied also in the general setting of human hand posture reconstruction. In this case, gloves are widely employed as, for example, the CyberGlove [9], the Super Glove [10], the Humanglove [11], and Color Glove [12], to mention but a few (for a detailed survey see [13]). In gloves as well, many problems are related to the types of sensors employed, as, in many cases, they need a long calibration phase, they are fragile (both mechanically and in terms of output reliability) or, simply, because they are not cost effective.

In this work, to tackle the joint sensorization problem we present a general framework to measure the angular position of revolute joints (also valid for higher-pair joints and non-technical joints) which is suited both for robotic and human hands: the hardware ingredient are the inertial measurement units (IMUs), while the underlying algorithm is a tailored version of the passive complementary (Mahony) filter [14].

As well known, IMUs are micro electro-mechanical systems (MEMS) composed by one or more inertial sensors. IMUs are widely employed in general industry and aerospace industry for GPS guidance systems, in vehicles [15] and in particular in unmanned aerial vehicles (UAVs) [16]. Moreover, IMUs are also employed to stabilize camera devices [17], to reconstruct human gestures [18] or human walk [19], to reconstruct athletic movements [20], [21], [22] and in entertainment field to develop game controllers [23].

This document is organized as follows: section II states the attitude estimation problem via inertial measurements; section III briefly recalls the Mahony filter basics; section IV describes the method for reconstructing joint angle positions form IMU measurements, while section V reports on the experiments for the validation of the model presented. Finally, conclusions are drawn in section VI.

II. Inertial Measurements

Generally, IMUs are classified based on the number of measurement axes. Thus, for example, an IMU with 3 axis accelerometer, 3 axis magnetometer and 3 axis gyro has 9 axes, aka 9 DoF (Degrees of Freedom).

Each measured triplet (components of a three-component vector) is referred to a body frame $\{B\}$ "strap down" (attached) to the IMU, which is in generic motion w.r.t. an inertial frame $\{A\}$. In this work, we call $R = R_{ab}$ the rotation matrix describing the "displacement" from frame $\{A\}$ to frame $\{B\}$ (coordinate transformation from $\{B\}$ components to $\{A\}$ components).

Accelerometer: measures the instantaneous acceleration of $\{B\}$ w.r.t $\{A\}$. The measurement a is derived from the linear acceleration ($\dot{v}$) of the origin of $\{B\}$ minus the gravitational acceleration vector g_0, both expressed in $\{A\}$. The acceleration measurement a (in $\{B\}$) is therefore

$$a = R^T(\dot{v} - g_0) + b_a + \mu_a, \tag{1}$$

where b_a and μ_a model measurement bias and noise, respectively. In many robotic applications, the generic linear accelerations of the links are negligible w.r.t. the gravitational acceleration. Therefore, the linear acceleration can be considered influenced only by bias and noise by rewriting Eq. (1) as

$$a = R^T g_0 + b_a + \mu_a. \tag{2}$$

If only the direction of the gravity field is needed, it is possible to write

$$v_a = \frac{a}{|a|}. \tag{3}$$

Magnetometer: measures the magnetic field (m) in $\{B\}$. Therefore, the relation with the inertial frame components ${}^A m$ is given by

$$m = R^T({}^A m) + b_m + \mu_m \tag{4}$$

where ${}^A m$ is the Earth magnetic field in the inertial frame $\{A\}$, while b_m and μ_m are bias and noise on the measurements, respectively. In case of magnetic sources close to the IMU, noise can be very significant. As for the accelerometer, if only the direction of the magnetic field is needed, it is possible to write

Preprint submitted to 2015 IEEE International Conference on Robotics and Automation. Received October 1, 2014.

$$v_m = \frac{m}{|m|}. \tag{5}$$

Rate Gyro: measures angular velocity Ω_r of the frame $\{B\}$ w.r.t $\{A\}$ expressed in $\{B\}$,

$$\Omega_r = \Omega + b + \mu \tag{6}$$

where Ω is the true angular velocity (again in $\{B\}$), while b and μ are bias and noise, respectively.

III. The Mahony-Hamel Filter

In this section we briefly recall a discrete version of a passive complementary filter presented in [14]. Let $\{A\}$ be the inertial reference frame and $\{B\}$ the local IMU frame which has unknown orientation w.r.t. $\{A\}$. The purpose of the filter is to find the rotation matrix $R_{ab} = R$, or rather the matrix that describes the orientation of frame $\{B\}$ w.r.t. $\{A\}$ from the IMU istantaneous measurements.

The filter builds up an instantaneous algebraic measurement of the rotation matrix R_y as the solution of the following optimization problem

$$R_y = \arg\min_{R\in SO(3)} (\lambda_{\text{acc}}|v_a^* - Rv_a|^2 + \lambda_{\text{mag}}|v_m^* - Rv_m|^2) \approx R_{ab}, \tag{7}$$

where v_a^* and v_m^* are the measurements of gravity and magnetic field in $\{A\}$, respectively, while v_a and v_m are the IMU's instantaneous measurements in $\{B\}$. λ_{acc} and λ_{mag} are just weights (i.e., adjustable parameters) chosen on the basis of the relative confidence in sensor outputs. The optimization problem (7) has three degrees of freedom and, depending on the particular configuration, it cannot be easily solved. Furthermore, without the magnetometer, particular IMU configurations cannot be recovered (rotations around the gravity vector).

Starting from these considerations, the idea of the filter is to have an estimate of the rotation matrix based on previous estimates: in this manner it is always possible to estimate the rotation matrix, also in case of momentary loss of data from the IMU.

Calling $\hat{R}$ an estimate of the true rotation matrix R w.r.t. a reference frame denoted with $\{E\}$, it is possible to state that if $\hat{R} \cong R$, $\{A\}$ and $\{E\}$ coincide.

Now, defining an estimate of the error matrix as $\tilde{R} = \hat{R}^T R$, we can notice that if the estimation matrix is close to the true one, the matrix $\tilde{R}$ is close to the identity. The goal of the filter is to provide a set of dynamic equations for the estimated matrix $\hat{R}(t) \in SO(3)$ that drive the error rotation matrix closest to identity, i.e. $\tilde{R} \to I_3$.

Starting from the *correct* kinematics

$$\dot{R} = R\Omega_x, \tag{8}$$

where Ω is the rate gyros in $\{B\}$ while "$_x$" is the skew operator such that

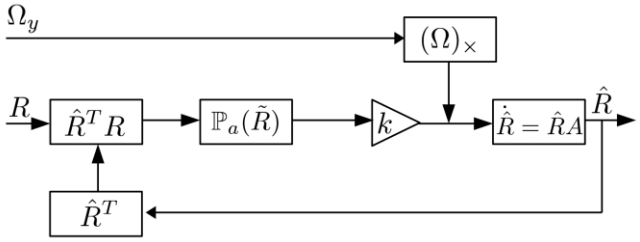


Fig. 2. Block diagram of the simplified form of the passive complementary filter

$$\Omega = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix}, \quad \Omega_x = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix}, \tag{9}$$

the filter kinematics is composed by: (i) a prediction term based on the measurement Ω, and (ii) a correction term $\omega := \omega(\tilde{R})$ derived from the error matrix $\tilde{R}$, which can be rewritten in the inertial frame becoming

$$\dot{\hat{R}} = (R\Omega + k_p\hat{R}\omega)_x\hat{R}, \quad \hat{R}(0) = \hat{R}_0, \tag{10}$$

where $k_p > 0$ is a positive gain to guarantee the filter convergence. The expression in Eq. (10) expresses the dynamics of a generic *complementary filter on SO(3)*. In this work, we use the *Passive Complementary Filter* (PCF), where also the prediction term depends on the filtered attitude $\hat{R}$, instead of R. In this manner, the filter dynamics become

$$\dot{\hat{R}} = (\hat{R}\Omega + k_p\hat{R}\omega)_x\hat{R}. \tag{11}$$

The ω in the correction term in (10) and (11) depends on $(\tilde{R})$ and it is defined as

$$\omega := vex(\mathbb{P}_a(\tilde{R})), \tag{12}$$

where *vex* is the anti-skew operator, while $\mathbb{P}$ is the anti-symmetric projection operator in the square matrix space. This last choice allows us to write the PCF dynamics in the IMU local frame $\{B\}$, resulting in

$$\dot{\hat{R}} = \hat{R}(\Omega_x + k_p\mathbb{P}_a(\tilde{R})). \tag{13}$$

Fig. 2 shows the block diagram of the passive complementary filter.

As in Bastelseiten work [24], we implemented a discrete version of the passive complementary filter considering the correction term in Eq. (12) as a PI controller. In algorithm 1, we detail the steps followed to apply a discrete version of the complementary filter to measurements coming from a 9 DoF IMU and obtain its instantaneous attitude matrix.

Here, K_p and K_i are the proportional and the integral coefficients of the PI regulator, respectively. In Algorithm 1, step no. 11, a new estimation of the rotation matrix is calculated as the sum of the previous one with a matrix depending on a correction term. This strategy allows to overcome to problem caused by loss of data from the IMU

Algorithm 1 Discrete Filter Version at n^{th} step

1: Reading the current values of the accelerometers (a'_{b_n}), magnetometers (m'_{b_n}) and gyro rates (Ω_{b_n}) in the local IMU frame $\{B\}$
2: Normalizing gravity and magnetic field vector read from IMU $a_{b_n} = \frac{a'_{b_n}}{|a'_{b_n}|}$, $m_{b_n} = \frac{m'_{b_n}}{|m'_{b_n}|}$
3: Determining the gravity vector in $\{B\}$ from the current attitude estimate and normalized gravity vector in the reference frame ($d_{acc_n} = R^T_{n-1} a_{a_n}$)
4: Determining the magnetic field vector in $\{B\}$ from the current attitude estimate and the normalized magnetic vector in the reference frame ($d_{mag_n} = R^T_{n-1} m_{a_n}$)
5: Calculating the gravity vector error $e_{acc_n} = a_{b_n} \times d_{acc_n}$
6: Calculating the magnetic vector error $e_{mag_n} = m_{b_n} \times d_{mag_n}$
7: Summing the two error vectors premultiplied by their weights $e_n = \lambda_{acc} e_{acc_n} + \lambda_{mag} e_{mag_n}$
8: Computing the contribution of the integrator $I_n = I_{n-1} + K_i \frac{1}{\Delta t} e_n$
9: Computing the correction term $\delta\omega_n = K_p e_n + I_n$
10: Computing the new gyro rates $\omega_n = \delta\omega_n + \Omega_{b_n}$ and its skew
11: Computing a first estimatation rotation matrix $R_{e_n} = R_{n-1} + R_{n-1}\omega_{n_x}\Delta t$
12: Computing the new rotation matrix R_n from the first estimation matrix R_{e_n}

but does not guarantee that the obtained matrix $R_{e_n} \in SO(3)$: therefore, a further step is required. At the step no. 12, as shown in [25], a new marix $R_n \in SO(3)$ closest to R_{e_n} (in Frobenius norm) is computed minimizing the following cost function:

$$||R_n - R_{e_n}||_F = \sqrt{\mathrm{Tr}((R_n - R_{e_n})^T (R_n - R_{e_n}))}, \qquad (14)$$

where Tr is the matrix trace operator. The solution of (14) is equivalent to maximizing the trace of the matrix $R_n^T R_{e_n}$. This leads to $R_n \in SO(3)$ as given by

$$R_n = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \sigma \end{bmatrix} V^T, \qquad (15)$$

where U and V^T are, respectively, left- and right- (orthogonal) eigenvector matrices of R_{e_n} (from its singular value decomposition $R_{e_n} = U\Sigma V^T$), while $\sigma = \det(UV^T)$. This guarentees that $\det(R_n) = 1$.

A. IMU Orientations

As previously described, each sensor on board the IMU returns a measurement in the IMU-fixed-frame, with respect to the inertial frame. Now, if we consider two different IMUs (IMU_1, IMU_2), along with their frames $\{B_1\}$ and $\{B_2\}$, they will return two sets of measurements (r_1,r_2) w.r.t. a common inertial frame. Applying PCF to r_1, we will obtain a rotation matrix $R_{ab_1} = R_1$. Analogously, from r_2 we will obtain $R_{ab_2} = R_2$. Trivially, the rotation matrix $R_{b_1 b_2} = R_{12}$ expressing attitude $\{B_2\}$ w.r.t. $\{B_1\}$ is given by

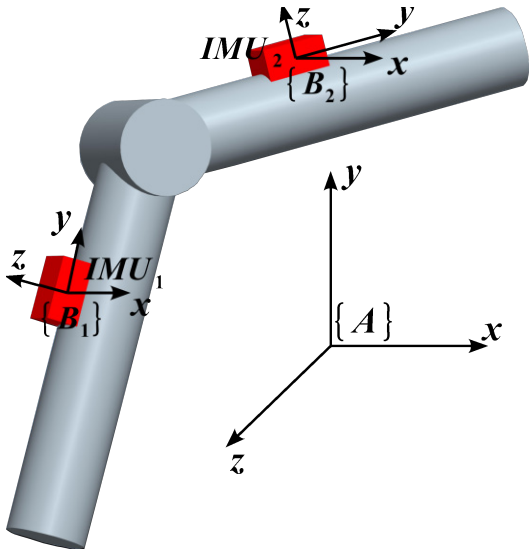


Fig. 3. Simple structure with two link connected by a revolute joint

$$R_2 = R_1 R_{12} \quad \Leftrightarrow \quad R_{12} = R_1^T R_2, \qquad (16)$$

It is possible to show that using the two sets of measures (r_1, r_2) in a suitable way, the PCF *directly* returns R_{12} — which is what we are interested in hand pose estimation.The PCF computes, in a stepwise fashion, the new estimate of the rotation matrix from the previous one and a new gyro rate correction term as

$$R_n = R_{n-1} + R_{n-1}\omega_{n_x}\Delta t; \quad \omega_n = \delta\omega_n + \Omega_n. \qquad (17)$$

Considering the orientation between the two IMUs, the rotation matrix R_n in (17) becomes R_{12_n}. In (17), the gyro rate correction term ω_n is composed by the gyro rates Ω_n at the step n^{th} and by the gyro correction term $\delta\omega_n$ that depends by the error in the previous rotation matrix estimation. The gyro rates read from the IMU its the angular velocity w.r.t. inertial frame: therefore, if we are interested in the orientation of the frame $\{B_2\}$ with respect the frame $\{B_1\}$, we will be interested in the angular velocity of the frame $\{B_2\}$ w.r.t. frame $\{B_1\}$. We will then employ Ω_{21_n} given by

$$\Omega_{21_n} = \Omega_{2_n} - \Omega_{1_n}. \qquad (18)$$

As for the gyro correction term, this is given by the error on the estimated rotation matrix at the n^{th} step from the values of the gravity and the magnetic field in the inertial frame. Now, if we are interested in the relative rotation matrix between two IMUs, this will depend on the gravity and magnetic fields read in the IMUs local frames $\{B_1\}$ and $\{B_2\}$ so that

$$e_{acc_n} = a_{b_{1_n}} \times R^T_{12_n} a_{b_{2_n}} \qquad (19)$$

$$e_{mag_n} = m_{b_{1_n}} \times R^T_{12_n} m_{b_{2_n}}. \qquad (20)$$

To wrap up, the PCF is able to return an estimation of the orientation between two frames in space knowing the gravity and the magnetic field values in the frames and the relative angular velocity between these ones. In Algorithm (2), we report a modified version of Algorithm (1) that directly outputs the rotation matrix between two local frames $\{B_1\}$ and $\{B_2\}$ "strap down" to two IMUs (IMU_1, IMU_2).

Algorithm 2 Two IMUs Orientation Filter Version at n^{th} step

1: Reading the current values of the accelerometers ($a'_{b_{1_n}}$, $a'_{b_{2_n}}$), magnetometers ($m'_{b_{1_n}}$, $m'_{b_{2_n}}$) and gyro rates ($\Omega_{b_{1_n}}$, $\Omega_{b_{2_n}}$) in the two IMUs frames ($\{B_1\}$, $\{B_2\}$).
2: Normalizing gravity and magnetic field vector read from IMUs
$$a_{b_{1_n}} = \frac{a'_{b_{1_n}}}{|a'_{b_{1_n}}|}, a_{b_{2_n}} = \frac{a'_{b_{2_n}}}{|a'_{b_{2_n}}|}, m_{b_{1_n}} = \frac{m'_{b_{1_n}}}{|m'_{b_{1_n}}|}, m_{b_{2_n}} = \frac{m'_{b_{2_n}}}{|m'_{b_{2_n}}|}$$
3: Determining the gravity vector in $\{B_2\}$ from the current attitude estimate and gravity vector read in the $\{B_1\}$ frame ($d_{acc_n} = R^T_{12_{n-1}} a_{b_{1_n}}$)
4: Determining the magnetic field vector in $\{B_2\}$ from the current attitude estimate and the magnetic vector read in the $\{B_1\}$ frame ($d_{mag_n} = R^T_{12_{n-1}} m_{b_{1_n}}$)
5: Calculating the gravity vector error $e_{acc_n} = a_{b_{2_n}} \times d_{acc_n}$
6: Calculating the magnetic vector error $e_{mag_n} = m_{b_{2_n}} \times d_{mag_n}$
7: Summing the two error vectors premultiplied by their weights $e_n = \lambda_{acc} e_{acc_n} + \lambda_{mag} e_{mag_n}$
8: Computing the contribution of the integrator $I_n = I_{n-1} + K_i \frac{1}{\Delta t} e_n$
9: Computing the correction term $\delta\omega_n = K_p e_n + I_n$
10: Computing the relative gyro rates $\Omega_n = \Omega_{b_{2_n}} - \Omega_{b_{1_n}}$
11: Computing the new gyro rates $\omega_n = \delta\omega_n + \Omega_n$
12: Computing a first estimation rotation matrix $R_{e_n} = R_{12_{n-1}} + R_{12_{n-1}} \omega_{n_x} \Delta t$
13: Computing the new rotation matrix R_{12_n} from the first estimation matrix R_{e_n}

IV. Joint Angles Values from Rotation Matrix

Considering that, in most cases, robotic and human revolute joints have maximum range less than 180°, (as an example, for the DLR-HIT Hand II the range is about 80°, for the Pisa/IIT Softhand less than 120°) it can be convenient to combine elementary rotations for representing general orientations. In this work, we use a Roll-Pitch-Yaw representation, thus allowing a generic rotation matrix R to be written as

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = R_z(\varphi)R_y(\theta)R_x(\psi) = \\ = \begin{bmatrix} c_\varphi c_\theta & -s_\varphi c_\psi + c_\varphi s_\theta s_\psi & s_\varphi s_\psi + c_\varphi s_\theta c_\psi \\ s_\varphi c_\theta & c_\varphi c_\psi + s_\varphi s_\theta s_\psi & -c_\varphi s_\psi + s_\varphi s_\theta c_\psi \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi \end{bmatrix}, \tag{21}$$

where $c_x = \cos(x)$ and $s_x = \sin(x)$. Considering $\theta \in (-\pi/2, \pi/2) \Rightarrow \cos(\theta) > 0$ and from (21) it is possible compute the three angle values as

$$\begin{cases} \varphi = \arctan\left(\frac{r_{21}}{r_{11}}\right) \\ \theta = \arctan\left(\frac{-r_{31}}{\sqrt{r_{32}^2 + r_{33}^2}}\right) \\ \psi = \arctan\left(\frac{r_{32}}{r_{33}}\right) \end{cases} \tag{22}$$

In many robotic structures, as an example structure shown in Fig. (3), a single revolute joint (one DoF) is present between two links. In a robotic or human hand, there is a simple revolute joint between distal and middle phalanx as well as between middle and proximal phalanx. In these particular cases, it is possible consider a simplified version of the rotation matrix with a further reduction of calculations complexity. With reference to the structure shown in Fig. 3, between the links, then between the two IMUs, one 1-DoF revolute joint is present and its revolute axis is parallel to the x axis of IMU_1 and IMU_2. Then, IMU_2 with respect to IMU_1 can only rotate about its x axis and the rotation matrix obtained from data read from $IMUs$ and PCF should be a simple rotation about the x axis, as follows

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & r_{22} & r_{23} \\ 0 & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\psi & -s_\psi \\ 0 & s_\psi & c_\psi \end{bmatrix}, \tag{23}$$

and the joint angle value should be trivially given by

$$\psi = \arctan\left(\frac{r_{32}}{r_{33}}\right), \tag{24}$$

Calculation complexity is reduced to solution of an arctangent function.

In real robot however, it is impossible to obtain a rotation matrix as in (23). In fact, due to tolerances in the IMU costruction process, or simply in mounting IMUs on the links, it is very difficult to perfectly align IMU frames. As an example, referring to the structure shown in Fig. (3), and considering $R = R_z(\varphi)R_y(\theta)R_x(\psi)$, it is not possible to obtain from PCF $\varphi = 0$ and $\theta = 0$.

Therefore, between two robotic links in the robot zero configuration (all joints being in zero position) there are always three offset angles $\varphi_o, \theta_o, \psi_o \neq 0$. Offset angles due to mechanical inaccuracies are time constant and independent from the robot configuration, so it is always possible compute an offset rotation matrix between two links and then consider this one in the further joint angles value calculations.

From these considerations, in order to obtain more accurate values in the joint angles reconstruction procedure, we consider for each couple of links two different phases: (i) *initialization* phase, perfomed once at starting conditions, where the robot is in its zero configuration, and PCF is applied to all IMUs couples to obtain all offset rotation matrix R_o; (ii) *operative* phase, perfomed every time step, where PCF is applied to all IMU pairs during free robot movements, to obtain the on-line rotation matrix estimate R_n.

Preprint submitted to 2015 IEEE International Conference on Robotics and Automation. Received October 1, 2014.

In the operative phase, the exact joint angles values are computed from the current rotation matrix estimate obtained from the PCF. This one is given by two matrices: one due to mechanical precision R_o, and a second one due to joint angles variations R_{j_n} as

$$R_n = R_o R_{j_n} \quad \text{and} \quad R_{j_n} = R_o^T R_n. \tag{25}$$

In this manner, R_{j_n} depends only on the joint angles values. By applying equations shown in (25), the computation of joint angle values is an easy process. If, on the one hand, management of offset angles increases the computational burden (a matrix product per IMU pair is required), one the other hand it guarantees higher precision and greater reliability of the calculations.

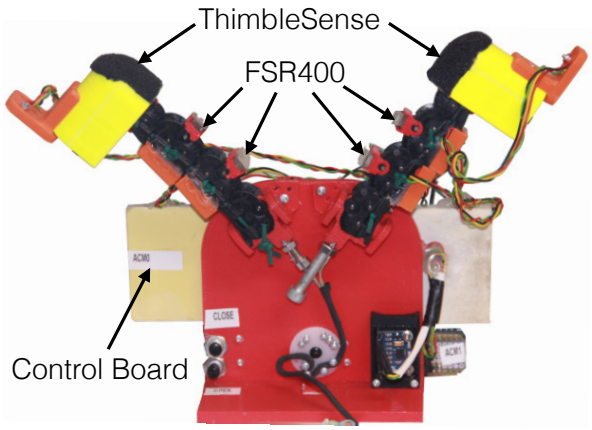
V. Experimental Results

To test the IMUs and PCF joint angles reconstruction, the gripper reported in Fig. 4 (a) was built. This is composed by a plastic frame and two fingers of the PISA/IIT SoftHand (see also [5] for more details). Fingers are composed by three revolute Hillberry's (roll-articular) joint (see Fig. 4 (c) and [26] for further details) while we consider that a spherical joint (virtual wrist) is connecting the palm to a fixed (laboratory) frame. Overall, the gripper configuration depends on nine joint angles. We model the rolling contact joint with two revolute joints, with same joint angles $q/2$, coupled by a virtual link.

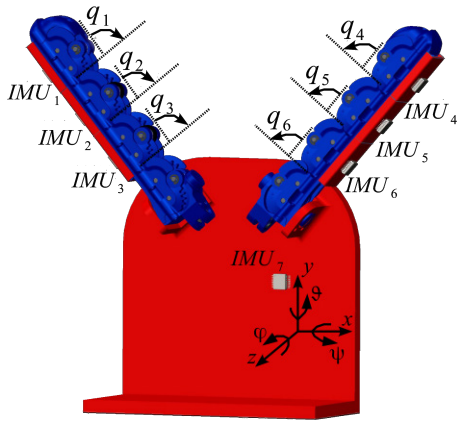
For the sensorization of the gripper we used seven IMUs: three for each finger, and one for the palm. Applying the PCF to data read by IMU_1 and IMU_2 we compute q_1 and from data of IMU_2 and IMU_3 we compute q_2. In the same manner, by applying the PCF to IMU_1 and IMU_3 we compute the angle $q_* = q_1 + q_2$.

In these tests we employed 6-DoF IMUs composed by three axis accelerometer and three axis gyroscope, without the 3-DoF magnetometer. Thus, it is not possible to measure the palm yaw angle ϕ and, in the case palm angle is not zero, ($\psi \neq 0$), it is not possible to compute the exact joint angles values for finger joints.

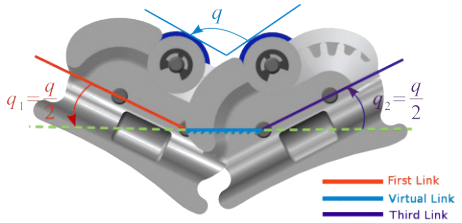
IMUs employed are MPU6050 sensors from InvenSense (see also [27] for more details), with refresh frequency of 1 kHz, power supply of 3.3 V and operating current of 3.7 mA. MPU6050's support digital serial communication in fast mode I^2C (400 kHz). Each IMU is connected and communicates with an Arduino Micro board [28]. On line visualization of the gripper posture is programmed with ROS (Robot Operating System) [29]. Figs. 5, 6, 7 and 8, show the real gripper during experiments and its resultant posture reconstruction in ROS. In particular, Fig. 5 shows the gripper rest configuration during the initialization phase. This is the zero position of each joint angle value, (i.e $q_n = 0$), and the PCFs are applied to obtain the offset rotation matrices. Figs.6 and 7 show the gripper and its ROS avatar in two different positions, while Fig. 8 show the roll and pitch reconstruction of the gripper with respect the inertial frame obtained applying the PCF to IMU_7.



(a) Gripper



(b) IMUs Positions



(c) Hillberry's Joint

Fig. 4. Gripper setup and its main features for the proposed experiments. In (a) we report the gripper used to test the joint angle calculation procedure while in (b) theCAD representation 2-fingered gripper built out of the Pisa/IIT SoftHand fingers (Details on the: joint angle taxonomy, positioning of the IMUs on the phalanxes and the frame palm, components). Finally, in (c) we point out the schematic illustration of the generic Pisa/IIT SoftHand finger roll-articular joint.

A. Posture Validation

In order to validate the gripper postures we employ FTR400 mono axial force sensors (from Interlink Electronics) to discriminate which link of the gripper is in contact with a grasped object. Moreover, in order to detect contact points and related forces on the gripper fingertips, we fix on the fingertips of the gripper the *ThimbleSense* (for more details see [30]), properly modified for fitting the current design (see also Fig. 4(a) for more details). In this manner, we are able to detect which link is in contact with the grasped

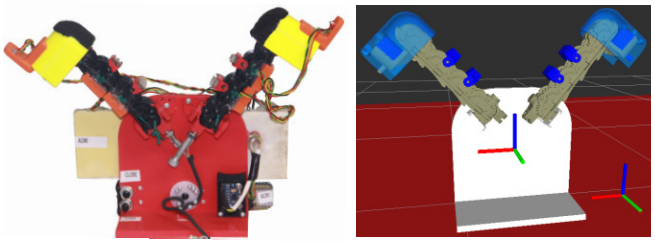


Fig. 5. On the left: gripper in its zero configuration; on the right: on-line reconstruction of the gripper pose in ROS.

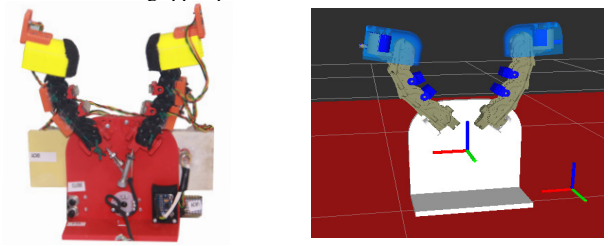


Fig. 6. On the left: gripper closing in free air; on the right: on-line reconstruction of the gripper pose in ROS.

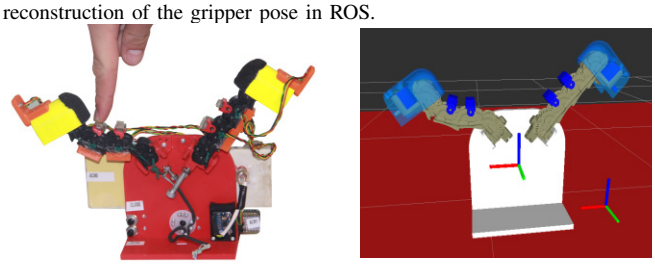


Fig. 7. On the left: gripper phalanx pushed by an external force; on the right: on-line reconstruction of the gripper pose and force applied in ROS (methodology for force measurement not described in this paper).

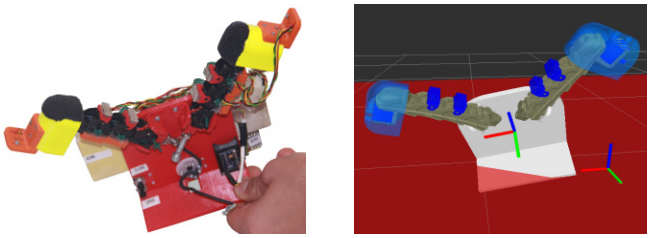


Fig. 8. On the left: gripper undergoing a roll and pitch angle rotations; on the right: on-line reconstruction of the gripper pose in ROS.

object and, in addition, to measure forces at contact points (see also Fig. 10).

Considering that the FTR400 and *ThimbleSense* sensors are rigidly connected to the phalanges, and knowing the gripper posture with the algorithm here presented, it is possible to indirectly validate the quality of posture reconstruction.

First, we compute the best fit circle to the contact points measured, and then we compare its radius with the ground truth value.

In Fig. 9, we report the validation experiments: the gripper grasps three circular object with radii of 24.5 mm, 32.5 mm and 41.5 mm, respectively. Reconstruction results are reported in Table I. It is possible to notice that the small error on the radius value implies good posture reconstruction.

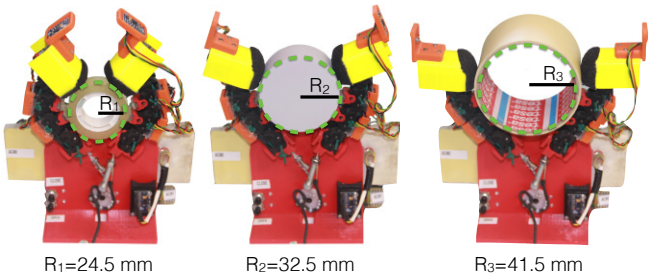


Fig. 9. Gripper during validation tests. From left to right: the gripper grasps three different circular objects with radii of 24.5 mm, 32.5 mm and 41.5 mm, respectively.

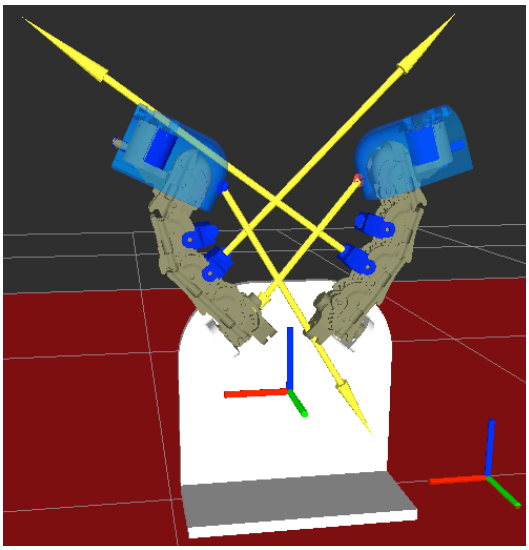


Fig. 10. Example of contact points and contact forces applied by the hand to the object during the grasp of circle with radius 41.5 mm.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we described a fast and accurate method to estimate joint angle values of a robotic hand, using low-cost IMU sensors and a passive complementary filter (Mahony's passive filter). Two versions of the estimation algorithm were presented that allow also to directly estimate the relative attitude of the IMUs attached to subsequent links of a serial kinematic chain. The overall architecture was tested on a gripper composed by roll-articular joints extracted from the Pisa/IIT SoftHand modular fingers. Satisfactory results were obtained in terms of time response, reconstruction accuracy and low computational cost. Future work will address the full sensorization of Pisa/IIT SoftHand: this will allow to exploit its compliance properties not only for grasping objects, but also to use it as an adaptive haptic probe to provide touch sensor capabilities in support for multi-modal perception of unstructured and cluttered environments.

VII. ACKNOWLEDGMENTS

This work is supported by the grant no. 600918 "PACMAN" - Probabilistic and Compositional Representations of Object for Robotic Manipulation - within the FP7-ICT-2011-9 program "Cognitive Systems", the European Re-

Object	R [mm]	R^* [mm]	% Error	C.P.
1	24.5	24.45	0.2	4
2	32.5	31.84	2	4
3	41.5	40.88	1.4	5

TABLE I

RECONSTRUCTION OF THE RADII OF THE TEST DISKS. R INDICATES THE RADIUS OF THE GRASPED OBJECT, R^* THE ESTIMATED RADIUS, % ERROR THE PERCENTAGE ERROR BETWEEN R AND R^*, RESPECTIVELY. C.P. (CONTACT POINTS) SHOWS THE NUMBERS OF ACTIVE CONTACTS BETWEEN THE GRIPPER AND THE OBJECTS IN THE GRASPS.

search Council under the ERC Advanced Grant no. 291166 SoftHands (A Theory of Soft Synergies for a New Generation of Artificial Hands), and under grant agreement no. 601165 Wearhap (Wearable Haptics for Humans and Robots), within the FP7/2007-2013 program: Cognitive Systems and Robotics.

REFERENCES

[1] R. M. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC Press, 1994.

[2] A. Cappozzo, F. Catani, U. Della Croce, and A. Leardini, "Position and orientation in space of bones during movement: Anatomical frame definition and determination," *Clinical Biomechanics*, vol. 10, no. 4, pp. 171–178, 1995.

[3] L. Lucchetti, A. Cappozzo, A. Cappello, and U. Della Croce, "Skin movement artefact assessment and compensation in the estimation of knee-joint kinematics," *Journal of Biomechanics*, vol. 31, no. 11, pp. 977–984, 1998.

[4] L. Balzini, L. Vannucchi, F. Benvenuti, M. Benucci, M. Monni, A. Cappozzo, and S. Stanhope, "Clinical characteristics of flexed posture in elderly women," *Journal of the American Geriatrics Society*, vol. 51, no. 10, pp. 1419–1426, 2003.

[5] M. G. Catalano, G. Grioli, E. Farnioli, A. Serio, C. Piazza, and A. Bicchi, "Adaptive synergies for the design and control of the pisa/iit softhand," *The International Journal of Robotics Research*, vol. 33, no. 5, pp. 768 – 782, April 2014.

[6] H. Liu, K. Wu, P. Meusel, N. Seitz, G. Hirzinger, M. H. Jin, Y. W. Liu, S. W. Fan, T. Lan, and Z. P. Chen, "Multisensory five-finger dexterous hand: The dlr/hit hand ii," in *Conf. on Intelligent Robots and Systems*. IEEE, 2008.

[7] F. Lotti, P. Tiezzi, G. Vassura, L. Biagiotti, G. Palli, and G. Melchiorri, "Develpment of ub hand 3: Early results," in *Conf. on Robotics and Automation*. IEEE, 2005.

[8] A. Schimitz, U. Pattaccini, N. Nori, L. Natale, G. Metta, and G. Sandini, "Design, realizatin and sensorization of the dexterous icub hand," in *Conf. on Humanoid Robots*. IEEE, 2010.

[9] "Cyberglove," online; accessed 25-September-2014. [Online]. Available: http://www.cyberglovesystems.com/

[10] J. LaViola, "A survey of hand posture and gesture recognition techniques and tecnology," Brown University, Tech. Rep., June 1999.

[11] "Humanware," online; accessed 25-September-2014. [Online]. Available: http://www.hmw.it/

[12] M. Schröder, C. Elbrechter, J. Maycock, R. Haschke, M. Botsch, and H. Ritter, "Real-time hand tracking with a color glove for the actuation of anthropomorphic robot hands," in *Conf. on Humanoid Robots*. IEEE, 2012.

[13] L. Dipietro, A. Sabatini, and P. Dario, "A survey of the glove-based system and their applications," *IEEE Transactions on System*, vol. 46, no. 4, pp. 461 – 482, July 2008.

[14] R. Mahony, T. Hamel, and J. Pflimlin, "Nonlinear complementary filter on the special orthogonal group," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203 – 1218, June 2008.

[15] Y. Wang, J. Mangnus, D. Kostic, H. Nijmeijer, and S. T. H. Jansen, "Vehicle state estimation using gps/imu integration," in *Conf. on Sensors*. IEEE, 2011.

[16] C. V. Angelino, V. R. Baraniello, and L. Cicala, "High altitude uav navigation using imu, gps and camera," in *Conf. on Information Fusion*. IEEE, 2013.

[17] G. Panahandeh and M. Jansson, "Imu-camera self-calibration using planar mirror reflection," in *Conf. on Indoor Positioning and Indoor Navigation*. IEEE, 2011.

[18] M. T. Wolf, C. Assad, M. T. Vernacchia, J. Fromm, and H. L. Jethani, "Gesture-based robot control with variable autonomy from the jpl biosleeve," in *Conf. on Robotics and Automation*. IEEE, 2013.

[19] S. K. Kim, S. Hong, and D. Kim, "A walking motion imitation framework of a humanoid robot by human walking recognition from imu motion data," in *Conf. on Humanoid Robots*. IEEE, 2009.

[20] A. Waegli, S. Guerrier, and J. Skaloud, "Redundant mems-imu integrated with gps for performance assessment in sports," in *Conf. on Position, Location and Navigation Symposium*. IEEE, 2008.

[21] M. Lapinski, M. Feldmeier, and J. A. Paradiso, "Wearable wireless sensing for sports and ubiquitous interactivity," in *Conf. on Sensors*. IEEE, 2011.

[22] Y. C. Huang, T. L. Chen, B. C. Chiu, C. W. Yi, C. W. Lin, Y. J. Yeh, and L. C. Kuo, "Calculate golf swing trajectories from imu sensing data," in *Conf. on Parallel Processing Workshops*. IEEE, 2012.

[23] D. Freeman, O. Hilliges, A. Sellen, K. O'Hara, S. Izadi, and K. Wood, "The role of physical controllers in motion video gaming," in *Conf. on Design Interactive Systems*, June 2012, pp. 701 – 710.

[24] O. Bastelseiten, "Personal blog," online; accessed 25-September-2014. [Online]. Available: http://www.olliw.eu/

[25] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modeling, Planning and Control*. Springer, 2010.

[26] B. Hillberry and A. J. Hall, *Rolling Contact Joint*, 1976, uS Patent 3,932,045.

[27] "Mpu6050 datasheet and register map page," online; accessed 28-September-2014. [Online]. Available: http://www.invensense.com/mems/gyro/mpu6050.html

[28] "Arduino micro home page," online; accessed 28-September-2014. [Online]. Available: http://arduino.cc/en/Main/ArduinoBoardMicro

[29] "Ros home page," online; accessed 25-September-2014. [Online]. Available: http://www.ros.org/

[30] E. Battaglia, G. Grioli, M. G. Catalano, M. Santello, and A. Bicchi, "Thimblesense: an individual-digit wearable tactile sensor for experimental grasp studies," in *IEEE International Conference on Robotics and Automation*, 2014.