

Endowing the Pisa/IIT SoftHand with the sense of touch

Gaspare Santaera, Emanuele Luberto, Marco Gabiccini, Antonio Bicchi.

Abstract—A modified version of the Mahony-Hamel passive complementary filter is used to obtain the orientation of a frame $\{A\}$ with respect to another frame $\{B\}$, expressed by a rotation matrix and subject to kinematic constraints, to reconstruct a two-finger gripper. In this report, we use that filter to estimate all joint angles of the five-finger Pisa/IIT SoftHand and the potential application of object recognition with no visual input a priori and using a naive recognizer.

I. INTRODUCTION

In ??, a modified version of the Mahony-Hamel passive complementary filter was used to obtain the orientation of a frame $\{A\}$ with respect to another frame $\{B\}$, expressed by a rotation matrix and subject to kinematic constraints, to reconstruct a two-finger gripper. In this section, we extend that work to the five-finger Pisa/IIT SoftHand and the potential application of object recognition without vision and using a naive recognizer to tackle problems devised in Task 3.2. The following sections describe the components of the proposed solution.

II. THE MADGWICK FILTER

Sebastian Madgwick [1] studied a new algorithm to tackle the singularities associated with the Euler parametrization by employing quaternions. The Madgwick filter is similar to the Mahony-Hamel filter in the use of a correction factor that depends on the data read from an IMU at each time step. Let be ${}^b\omega_x$, ${}^b\omega_y$ and ${}^b\omega_z$ angular rate measures (in rad s $^{-1}$) with respect the IMU body frame $\{B\}$ respectively about x , y and z axis and ${}^b\Omega$ a vector containing these measures as

$${}^b\Omega = [0 \quad {}^b\omega_x \quad {}^b\omega_y \quad {}^b\omega_z], \quad (1)$$

the quaternion describing the rate of change of the earth frame $\{A\}$ with respect to the sensor frame $\{B\}$ can be written as

$${}^b\dot{q} = \frac{1}{2} {}^b_a\bar{q} \otimes {}^b\Omega, \quad (2)$$

where \otimes denotes the quaternion product, and $\bar{\cdot}$ denotes the normalization operator. From (2), the orientation of the earth frame with respect to sensor frame at time t is given by

$${}^b_a\dot{q}_{\Omega,t} = \frac{1}{2} {}^b_a\hat{q}_{t-1} \otimes {}^b\Omega_t, \quad (3)$$

$${}^b_aq_{\Omega,t} = {}^b_a\hat{q}_{t-1} + {}^b_a\dot{q}_{\Omega,t}\Delta t, \quad (4)$$

where ${}^b\Omega_t$ is the angular rate measured at time t , Δt is the sampling period and ${}^b_a\hat{q}_{t-1}$ is the previous estimate of the quaternion.

Now, by reading from a sensor a set of accelerometer and compass measurements in a frame strap down to the

sensor, one can find infinite earth frame orientations. Thus, the orientation problem can be written as an optimization problem, where quaternion ${}^b_a\bar{q}$ aligns a predefined reference direction of a field (gravity or magnetic) in the earth frame ${}^a\bar{d}$ with the measured direction of the field in the sensor frame ${}^b\bar{s}$, stated as

$$\min_{{}^b_a\bar{q} \in \mathbb{R}^4} f({}^b_a\bar{q}, {}^a\bar{d}, {}^b\bar{s}), \quad (5)$$

with the objective function defined as

$$f({}^b_a\bar{q}, {}^a\bar{d}, {}^b\bar{s}) = {}^b_a\bar{q}^* \otimes {}^a\bar{d} \otimes {}^b_a\bar{q} - {}^b\bar{s}, \quad (6)$$

where ${}^b_a\bar{q}$ is the optimal quaternion, and ${}^a\bar{d} = [0 \quad d_x \quad d_y \quad d_z]$ and ${}^b\bar{s} = [0 \quad s_x \quad s_y \quad s_z]$ are the directions to be aligned in their respective reference frames. Here, $*$ denotes the conjugate operator of a quaternion.

The solution to (5) can be found using a gradient descent strategy, which proved to be a satisfactory choice in terms of low computational burden and efficiency. Using an initial guess for the optimal quaternion ${}^b_a\bar{q}_0$, it is possible to write the correction factor ${}^b_a\bar{q}_{k+1}$ linked to step $(k+1)$ as follows

$${}^b_aq_{k+1} = {}^b_a\bar{q}_k - \beta \frac{\nabla f({}^b_a\bar{q}_k, {}^a\bar{d}, {}^b\bar{s})}{\|\nabla f({}^b_a\bar{q}_k, {}^a\bar{d}, {}^b\bar{s})\|}, \quad k = 0, 1, 2 \dots n, \quad (7)$$

where

$${}^b_a\bar{q}_k = \frac{1}{2} {}^b_a\bar{q}_{k-1} \otimes {}^b\Omega_k, \quad k = 0, 1, 2 \dots n, \quad (8)$$

$$\nabla f({}^b_a\bar{q}_k, {}^a\bar{d}, {}^b\bar{s}) = J^T({}^b_a\bar{q}_k, {}^a\bar{d}) f({}^b_a\bar{q}_k, {}^a\bar{d}, {}^b\bar{s}), \quad (9)$$

$$f({}^b_a\bar{q}_k, {}^a\bar{d}, {}^b\bar{s}) = \begin{bmatrix} 2d_x(\frac{1}{2} - q_3^2 - q_4^2) + 2d_y(q_1q_4 + q_2q_3) + \\ 2d_x(q_2q_3 - q_1q_4) + 2d_y(\frac{1}{2} - q_2^2 - q_4^2) + \\ 2d_x(q_1q_3 + q_2q_4) + 2d_y(q_3q_4 - q_1q_2) + \\ 2d_z(q_2q_4 - q_1q_3) - s_x \\ 2d_z(q_1q_2 + q_3q_4) - s_y \\ 2d_z(\frac{1}{2} - q_2^2 - q_3^2) - s_z \end{bmatrix}, \quad (10)$$

$$J({}^b_a\bar{q}_k, {}^a\bar{d}) = \begin{bmatrix} 2d_yq_4 - 2d_zq_3 & 2d_yq_3 + 2d_zq_4 \\ -2d_xq_4 + 2d_zq_2 & 2d_xq_3 - 4d_yq_2 + 2d_zq_1 \\ 2d_xq_3 - 2d_yq_2 & 2d_xq_4 - 2d_yq_1 - 4d_zq_2 \\ -4d_xq_32d_yq_2 - 2d_zq_1 & -4d_xq_42d_yq_1 + 2d_zq_2 \\ 2d_xq_2 + 2d_zq_4 & -2d_xq_1 - 4d_yq_4 + 2d_zq_3 \\ 2d_xq_1 + 2d_yq_4 - 4d_zq_3 & 2d_xq_2 + 2d_yq_3 \end{bmatrix}, \quad (11)$$

with β being the step size, and J^T denoting the transpose of the Jacobian matrix of the objective function f .

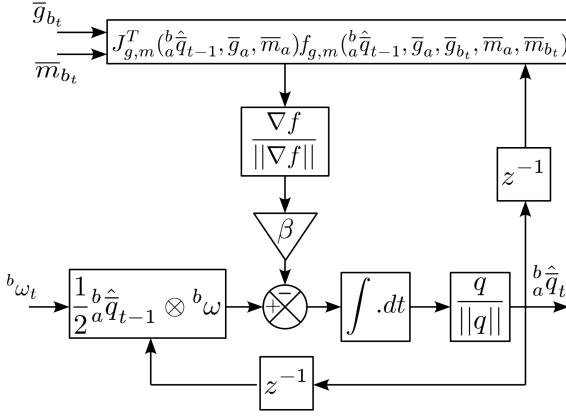


Fig. 1. Block diagram of the Madgwick passive complementary filter

Eq. (5) defines a general alignment problem. Therefore, by considering measurements from accelerometers and magnetometers, it is possible to write two different problems: the first one using accelerometers data $f_g(b_a \bar{q}_k, {}^a \bar{g}_a, {}^b \bar{g}_b)$ where ${}^a \bar{g}_a = \bar{g}_a$ is the gravity field measured in the inertial frame and ${}^b \bar{g}_b = \bar{g}_b$ is the gravity field in the sensor frame, and the second using the magnetometer data $f_m(b_a \bar{q}_k, {}^a \bar{m}_a, {}^b \bar{m}_b)$ where ${}^a \bar{m}_a = \bar{m}_a$ is the magnetic field read in the inertial frame and ${}^b \bar{m}_b = \bar{m}_b$ is the magnetic field in the sensor frame.

The two problems are combined to obtain a unique quaternion of the inertial frame with respect to the sensor frame as follows

$$f_{g,m}(b_a \bar{q}, \bar{g}_a, \bar{g}_b, \bar{m}_a, \bar{m}_b) = \begin{bmatrix} f_g(b_a \bar{q}, \bar{g}_a, \bar{g}_b) \\ f_m(b_a \bar{q}, \bar{m}_a, \bar{m}_b) \end{bmatrix}. \quad (12)$$

The expression of the optimal quaternion using again the gradient descent strategy at $(k+1)$ is

$${}^b q_{k+1} = {}^b \bar{q}_k - \beta \frac{\nabla f_{g,m}(b_a \bar{q}_k, \bar{g}_a, \bar{g}_b, \bar{m}_a, \bar{m}_b)}{\|\nabla f_{g,m}(b_a \bar{q}_k, \bar{g}_a, \bar{g}_b, \bar{m}_a, \bar{m}_b)\|}, \quad k = 0, 1, 2 \dots n, \quad (13)$$

where

$$\nabla f_{g,m}(b_a \bar{q}_k, \bar{g}_a, \bar{g}_b, \bar{m}_a, \bar{m}_b) = J_{g,m}^T(b_a \bar{q}_k, \bar{g}_a, \bar{m}_a) \quad (14)$$

and

$$J_{g,m}^T(b_a \bar{q}_k, \bar{g}_a, \bar{m}_a) = \begin{bmatrix} J_g(b_a \bar{q}_k, \bar{g}_a) \\ J_m(b_a \bar{q}_k, \bar{m}_a) \end{bmatrix}, \quad (15)$$

with J_g and J_m the Jacobians of f_g and f_m , respectively.

Fig. 1 shows the block diagram of the Madgwick filter. In particular, the gravity field in the inertial frame is typically set as $g_a = [0 \ 0 \ 0 \ 1]$, or rather, using a North-East-Down convention. The earth magnetic field in the inertial frame can be considered to have two components: one along the horizontal axis and one in the vertical axis, with its vertical component due to the inclination of the field depending on the latitude. In Pisa, for instance, it is about 1° w.r.t. the horizontal, so $m_a = [0 \ m_{ax} \ 0 \ m_{az}]$.

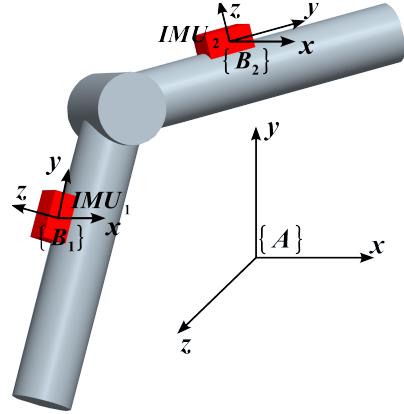


Fig. 2. Simple structure with two link connected by a revolute joint

A. Orientation between two IMU sensors

The use of two IMUs in a kinematic chain as shown in Fig. 2 returns two orientations, namely ${}^b_1 \hat{q}$ and ${}^b_2 \hat{q}$ in the inertial frames $\{B_1\}$ and $\{B_2\}$ of body 1 and 2, respectively, with respect to a common frame $\{A\}$. The relative orientation of the bodies can be readily obtained by

$${}^b_2 \hat{q} = {}^b_2 \bar{q}^* \otimes {}^b_1 \hat{q}. \quad (16)$$

However, it is also possible to apply the Madgwick filter to the measurements read from the two IMUs, to obtain ${}^b_2 \hat{q}$ directly, since it applies to the relative orientation of two bodies.

Thus, having the accelerometer, gyroscope and magnetometer measurements, \bar{g}_{b_i} , ${}^{b_i} \Omega$ and \bar{m}_{b_i} , in frame $\{B_i\}$, for $i = 1, 2$ bodies, the terms of the correction factor shown in (7) change correspondingly, that is, (8) becomes

$${}^{b_1} \bar{q}_k = \frac{1}{2} {}^{b_1} b_2 \bar{q}_{k-1} \otimes {}^{b_1} b_2 \Omega_k, \quad k = 1, 2, 3 \dots n, \quad (17)$$

where

$${}^{b_2} \Omega_k = {}^{b_2} \Omega_k - {}^{b_2} b_2 \bar{q}_{k-1} \otimes {}^{b_2} b_1 \Omega_k \otimes {}^{b_2} b_2 \bar{q}_{k-1}^*, \quad (18)$$

is the angular rate of frame $\{B_2\}$ with respect the frame $\{B_1\}$. Then, (14)), becomes

$$\begin{aligned} \nabla f_{g,m}({}^{b_1} \bar{q}_k, \bar{g}_{b_2}, \bar{g}_{b_1}, \bar{m}_{b_2}, \bar{m}_{b_1}) &= J_{g,m}^T({}^{b_1} \bar{q}_k, \bar{g}_{b_2}, \bar{m}_{b_1}) \\ &\quad f_{g,m}({}^{b_1} \bar{q}_k, \bar{g}_{b_2}, \bar{g}_{b_1}, \bar{m}_{b_2}, \bar{m}_{b_1}). \end{aligned} \quad (19)$$

Thus, the relative quaternion at step $(k+1)$, is given by

$${}^{b_2} q_{k+1} = {}^{b_2} b_2 \bar{q}_k - \beta \frac{\nabla f_{g,m}({}^{b_1} \bar{q}_k, \bar{g}_{b_2}, \bar{g}_{b_1}, \bar{m}_{b_2}, \bar{m}_{b_1})}{\|\nabla f_{g,m}({}^{b_1} \bar{q}_k, \bar{g}_{b_2}, \bar{g}_{b_1}, \bar{m}_{b_2}, \bar{m}_{b_1})\|}, \quad k = 0, 1, 2 \dots n. \quad (20)$$

Algorithm 1 summarizes the steps followed to obtain the relative orientation of the two bodies.

III. HARDWARE

In this section, the IMU (Inertial Measurements Unit) sensors are described along with their interrogation and management by the filter. To reconstruct the hand posture 17 IMUs are rigidly attached to a glove that is worn by

Algorithm 1. Two IMUs Madgwick Discrete Filter at n^{th} step

- 1: Reading the current values of the accelerometers (g_{b_1n}), magnetometers (m_{b_1n}) and gyro rates (Ω_{b_1n}) in the local IMU_1 frame $\{B_1\}$
- 2: Normalizing gravity and magnetic field vector read from IMU_1 $\bar{g}_{b_1n} = \frac{g_{b_1n}}{\|g_{b_1n}\|}$, $\bar{m}_{b_1n} = \frac{m_{b_1n}}{\|m_{b_1n}\|}$
- 3: Reading the current values of the accelerometers (g_{b_2n}), magnetometers (m_{b_2n}) and gyro rates (Ω_{b_2n}) in the local IMU_2 frame $\{B_2\}$
- 4: Normalizing gravity and magnetic field vector read from IMU_2 $\bar{g}_{b_2n} = \frac{g_{b_2n}}{\|g_{b_2n}\|}$, $\bar{m}_{b_2n} = \frac{m_{b_2n}}{\|m_{b_2n}\|}$
- 5: Computing the objective function value $f_{g,m}(\hat{q}_{n-1}, \bar{g}_{b_2n}, \bar{g}_{b_1n}, \bar{m}_{b_2n}, \bar{m}_{b_1n})$ using equations (10) and (12)
- 6: Computing the transpose of the Jacobian of the objective function $J_{g,m}^T(\hat{q}_{n-1}, \bar{g}_{b_2n}, \bar{m}_{b_2n})$ using equations (11) and (15)
- 7: Computing the correction terms $c_n = \beta \frac{\nabla f}{\|\nabla f\|}$ using equation (14)
- 8: Computing the orientation quaternion time variation $\dot{\hat{q}}_{d_n} = (\frac{1}{2} \hat{q}_{n-1} \otimes \Omega_k) - c_n$ using (18)
- 9: Computing the new orientation quaternion estimation given by its previously one and its current time variation $\hat{q}_n = \hat{q}_{n-1} + \dot{\hat{q}}_{d_n} \Delta t$
- 10: Normalizing the new estimated quaternion $\hat{q}_n = \frac{\hat{q}_n}{\|\hat{q}_n\|}$

hand. In particular, one device is mounted on each phalanx of each finger ($3 \times 5 = 15$), one is mounted on the hand palm, and the last one is attached to the wrist of the glove. Fig. 3 shows the complete hardware setup proposed as solution whose components are described below.

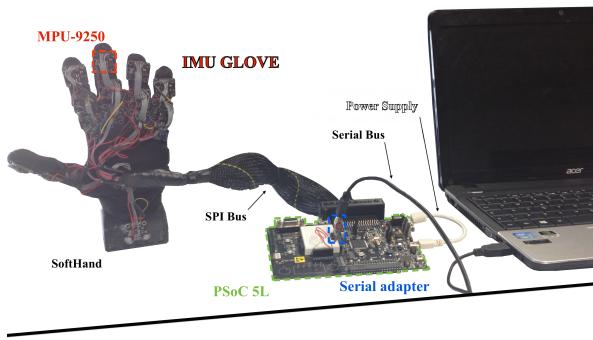


Fig. 3. Complete hardware setup of the IMU-based glove for the Pisa/IIT SoftHand

A. MPU-9250

Although the size of a generic IMU is very small, an accurate selection of the sensors to better assembly the glove led us to choose the device MPU-9250 by InvenSense [2].

This is a System in Package device (SiP) that combines two chips: the MPU-6500 device (used in the first feasibility

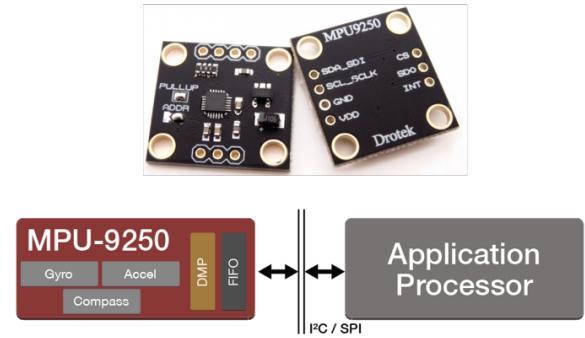


Fig. 4. MPU-9250

study [3]) containing a 3-axis gyroscope and a 3-axis accelerometer, and the AK8963, a market leading 3-axis digital compass. In particular:

- *Accelerometer*: Use separate proof masses for each axis. Acceleration along a particular axis induces displacement on the corresponding proof mass, and capacitive sensors detect the displacement differentially. The 3-analog information is digitized using individual on-chip 16-bit Analog-to-Digital Converters (ADCs) to sample each axis;
- *Gyroscope*: There are three independent vibratory rate gyroscopes, which detect rotation about the X, Y and Z axes. When the gyroscope are rotated about any of sense axes, the Coriolis force causes a vibration that is detected by a capacitive pickoff. The resulting signal is amplified, demodulated and filtered to produce a voltage proportional to the angular rate. This voltage, as for the accelerometer, passes through a ADC providing digital outputs.
- *Magnetometer*: The 3-axis magnetometer uses highly sensitive Hall sensor technology. It detects a magnetic field in the X, Y and Z axes. As the other, each ADC has 16-bit resolution.

The output data from axes sensors can be read from a 8-bit register. Each axis sensor presents two registers: the up and the low register. Thus, a complete information from a specific sensor occupies 48-bit. Considering, for example, the accelerometer there are 16-bit for the x axis, 16-bit for the y axis and 16-bit for the z axis.

Fig. 5 shows the orientation of the axes of sensitivity and the polarity of rotation.

As already mentioned, a complete sensorization of the glove and, therefore, of the hand needs 17 IMUs. This means that for keeping the time response of the system low, the communication between a master processing unit and each IMU becomes crucial. The MPU-9250 supports two different types of digital communication: I²C (Inter Integrated Circuit) and SPI (Serial Peripheral Interface). The I²C maximum working frequency is 400Hz, while the SPI works at 1Mhz. Therefore, the SPI allows a faster

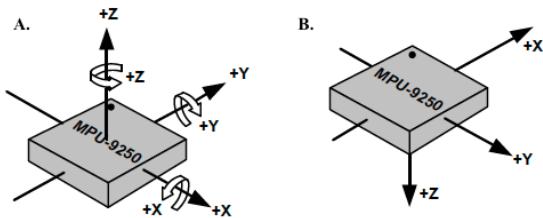


Fig. 5. A. Accelerometer and gyro orientation axes B. Magnetometer orientation axes

communication and using the Slave-Select pin, it allows the user to use a single bus to overcome the problem of setting a unique address per sensor.

In our work, the MPU-9250 always operates as a Slave device during standard Master-Slave SPI operation and to speed up communication between master and sensors, in our work three SPI bus are used [6].

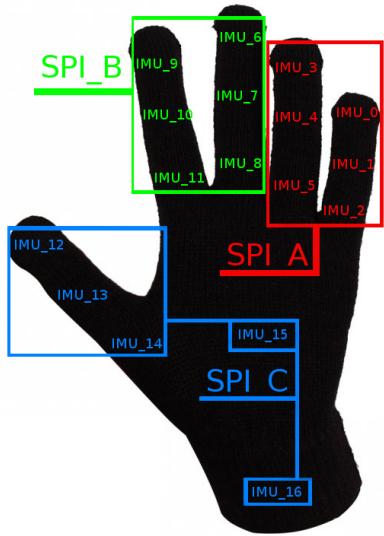


Fig. 6. SPI bus used in the IMU Glove

B. PSoC 5LP

This is master. It is a micro-controller that support SPI communication with more than 17 pins to manage all slave devices. The microcontroller is be able to read the IMUs as well as to send data to the PC through a serial adapter, where the Magdwick Filter is implemented. The processing unit used is a PSoC mounted on the *PSoC 5LP* board developed by Cypress Semiconductor [4]. The PSoC is a low-power ARM® Cortex - M3 based programmable system on chip devices offering unmatched high-precision analog and the flexibility to design custom system solutions. Combined with the free PSoC software development tools (PSoC Creator and PSoC Programmer) from Cypress, this board is a good solution for applications who need different hardware devices and good time response.

The PSoC integrated circuit is composed of a core, configurable analog and digital blocks, and programmable routing

and interconnect. The configurable blocks in a PSoC are the biggest difference from other micro controllers. For this reason it can be associated to the FPGA microcontroller family (Field-Programmable Gate Array).

To communicate with the PC, the PSoC5 has a dual-channel USB. A first channel *A* is connected to the PSoC 5 in FIFO parallel mode to allow a fastest transfers between the USB host PC and the PSoC 5. A second channel *B* is connected to the PSoC 5 in serial mode to allow a standard UART communication between the host PC and the PSoC5. However, channel *A*, commonly used to implement fast a communication, has a little communication buffer composed by 64 bytes, whereas to send all IMU data to the PC we would need more than 450 bytes. The channel *B* is a serial RS-232 channel and needs a Serial-USB converter to be connected to the PC. The PSoC 5LP mount on board presents an internal Serial-USB converter but its maximum baud rate is 115200, whereas we would need more than 900000. Starting from these considerations, a different solution was implemented using on PSoC a standard serial communication and an external Serial-USB module to connected PSoC board to PC as described next.

C. Serial Communication

The communication between PSocC5 and PC was created exploiting the serial adapter 990 004 [5]. It is beneficial to connect a micro-controller and a logic circuit to a PC with a high data transfer rate. The heart of the module is the chip FT232R distributed by FTDI [6], which works with a supply tension of 3.3V, thus it possible to connect the device to another TTL peripheral o micro-controller in the range of 3.3V-5V without the problem to convert the signal RS232 to the TTL. The circuit fits in a 25 × 18mm square. The device is USB 2.0 compatible and it allows full speed at 1Mb/s.

IV. SOFTWARE

To reconstruct the hand posture using the IMU glove two pieces of software are written. The first one is written in C and runs in the PSoC. This is called the firmware and it manages the IMU measurements as well as communication with the PC. The second one is written in C++ and runs in the PC. This is called the application and it manages communication with the firmware and applies the Madgwick filter to measurments.

A. Firmware on PSoC

The firmware is written using the PSoC creator IDE developed by Cypress. This one offers an intuitive GUI to manage the internal PSoC devices represented as code blocks, to configure the PSoC input/output, and to write the code necessary to connect and work PSoC internal device and pin.

Fig. 7 shows how the firmware manages the IMUs. In particular, there is an initialization phase *Init*, where all IMU internal registers are set. This means configure, e.g. internal clock, type of communication, accelerometer, gyro full width range, and so on. After the initialization, the

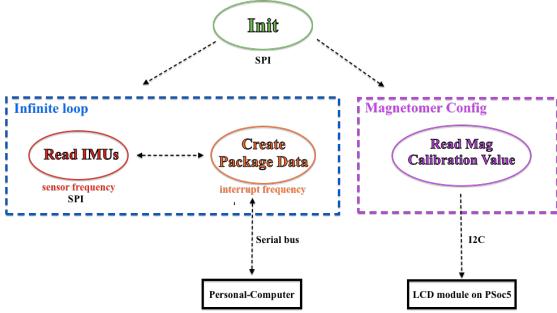


Fig. 7. Scheme of the firmware written in the PSoC5L

firmware can run in two different ways. The *Magnetometer Config*, performed whenever there are new IMUs, helps to know sensitivity adjustment data for each magnetometer axis. In fact, due to constructive inaccuracies during the magnetometer installation on the MEMS, the magnetometer axis pose or orientation can be different between two IMUs. Therefore, the manufacturer stores in each IMU 8-bit register, three numbers (one for each axis), to equalize magnetic measurements between two or more IMU. In particular, these registers are:

- $ASA_x[7 : 0]$: Magnetic sensor X-axis sensitivity adjustment value
- $ASA_y[7 : 0]$: Magnetic sensor Y-axis sensitivity adjustment value
- $ASA_z[7 : 0]$: Magnetic sensor Z-axis sensitivity adjustment value

The sensitivity correction factor s_a is given by

$$s_a = \frac{0.5(ASA_a - 128)}{128} + 1, \quad (21)$$

and for example the exact magnetic field on the x -axes is given by

$$H_{adj_x} = H_x s_x, \quad (22)$$

where H_x is the current data read from the measurement register, ASA_x is the sensitivity x -axes correction factor and H_{adj_x} is the real measurement. All magnetometer correction factors are computed, and then they are stored on the PC hard disk to correct the current data read from PSoC.

After the initialization phase, the firmware will run in an *Infinite loop* phase. This is subdivided in two independent sections

- *Read IMUs*: An internal counter each 20ms (50Hz) generates an interrupt where IMUs are sequentially read and the output data stored in 3 different matrices
 - Acc $\in \mathbb{R}^{17 \times 3}$
 - Gyro $\in \mathbb{R}^{17 \times 3}$
 - Mag $\in \mathbb{R}^{17 \times 3}$

The maximum frequency is the lowest one among the three sensors, and in particular the magnetometer has the lowest refresh frequency 100Hz.

• *Create Package Data*: Between two subsequent interrupt, the matrices are reorganized in a package following the communication protocol depicted in Fig. 8 with a total length of 462 bytes.

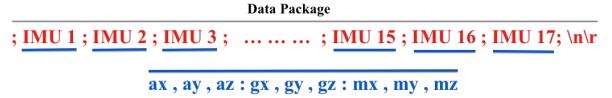


Fig. 8. Data package sent to the PC

The interrupt selecting the IMU reading and the data sending routines is generated using a PWM block. Similarly, the communication with the PC is implemented using an UART block.

B. C++ code on PC

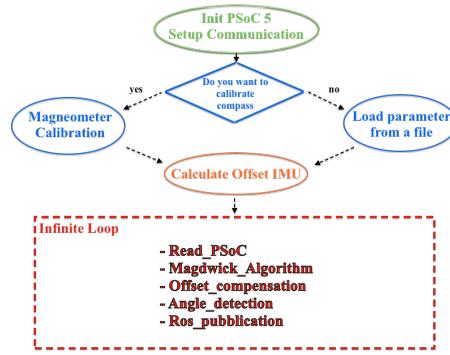


Fig. 9. Code C++ implemented in the PC

During an initial phase, the communication with the PSoC is configured. Then, the user decides whether to change or not the magnetometer calibration factors. If the user decides to keep the previous magnetometer corrector factors, the application applies the Madgwick filter to the IMU measurements to compute the offset angles between IMUs due to mounting inaccuracies. After these initial stages, the application enters in an infinite loop where it computes the relative orientation and the joint angles values from the current IMU measurements. Fig. 9 summarizes the tasks and phases performed by the application.

V. VALIDATION

In order to validate the hand posture reconstructed using the IMU glove, to demonstrate the potential of the proposed solution and its application for Task 3.3, some simple experiments have been performed on object recognition based on the hand configuration after grasping the object.

To validate the hand posture, a Pisa/IIT SoftHand is dressed with the IMU glove. Online visualization of the hand posture is programmed with ROS (Robot Operating System) [7]. Fig. 10 shows three different examples of hand posture reconstruction. A video was shot during a **static** and a **dynamic** configuration that validate the solution qualitatively.

Next, we challenge the proposed solution to recognize a grasped object based on the resulting hand configuration



Fig. 10. Hand posture reconstruction examples

and using a naive approach. To this end, a set of five cups are used, where actually four belong to the PaCMan object database [8]. The cup sizes are presented in Table I, and it is worth noting that, due their similarity, vision-based recognition techniques may present difficulties due to scaling issues.

The object recognition experiments is composed by two different steps: *Learn* and *Recognize*. The hand movements (open/close) are managed by a simple PID controller. The K_p , K_d and K_i parameters of the PID controller are set such that the hand squeezing force is low. In these conditions, the hand is not able to grasp an object, but behaves as a probe.

In the learning phase, the hand probes an object and the software running on the PC writes 20 row elements in a database. The row is composed by 1 number that identifies the object and 19 numbers that report the corresponding joint angles values (4 for little, 4 for ring, 4 for middle, 4 index and 3 for thumb).

In the recognition phase, the software reads the database and create in memory a matrix $O_{bj} \in \mathbb{R}^{n \times 20}$, where n is the number of learned objects. when the hand closes and probes the current object, the software compares the current joint angles values with the values recorded during the learning phase. The match uses a brute force root mean square error minimizer. Here, the software returns n different numbers m_n representing the distance of each of the n hand recorded poses to the current hand pose. The object in database whose hand pose signature is closest to the current one, has the smallest m_n number. Explicitly, the distance m_n is given by

$$m_n = \sqrt{(C_{angle_1} - O_{n_{angle_1}})^2 + \cdots + (C_{angle_{19}} - O_{n_{angle_{19}}})^2}, \quad (23)$$

where C_{angle_k} is the k^{th} hand current joint angles value, $O_{n_{angle_k}}$ is the k^{th} joint angles value corresponding to the i^{th} object, for $k = 1, 2, \dots, 19$ joints and $i = 1, 2, \dots, n$ objects.

Even with this naive recognizer, the proposed solution was successful in discriminating the cups as shown in Fig. 11. For more in-sight on the recognition examples, we suggest to see videos by clicking on [Object 1](#), [Object 2](#), [Object 3_a](#),

[Object 3_b](#), [Object 4_a](#), [Object 4_b](#), [Object 5_a](#) and [Object 5_b](#), where the subscripts a or b denote if the hand probes the cup handle or no, in this case the cup is recorded as two different objects.

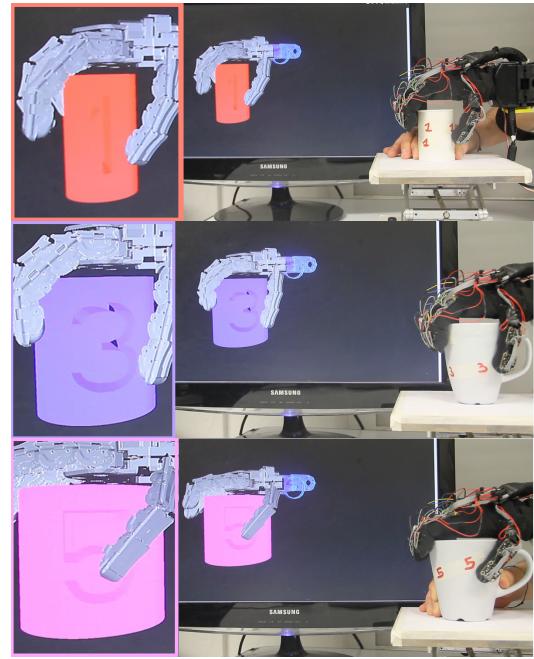


Fig. 11. Object recognition examples using the same cup of different sizes.

VI. CONCLUSIONS

We successfully provided the Pisa/IIT SoftHand with the sense of touch. The proposed solution was validated by visual inspection, as well as tested for object recognition with no visual input using a naive approach. Future work will comprise include tactile information to differentiate between contact states and complement the joint state information.

REFERENCES

- [1] S. Madgwick, A. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," in *Proc. IEEE/RSJ Intl Conf. on Rehabilitation Robotics*. Zurich, Switzerland: IEEE, 2011.
- [2] "Mpu9250 datasheet and register map page," accessed 23-February-2015.
- [3] G. Santaera, E. Luberto, A. Serio, M. Gabiccini, and A. Bicchi, "Low-cost, Fast and Accurate Reconstruction of Robotic and Human Postures via IMU Measurements," Seattle, USA, May 2015.
- [4] "Psoc 5lp development kit," accessed 23-February-2015.
- [5] "Serial - usb converter datasheet," accessed 23-February-2015.
- [6] "Ftdi chip home page," accessed 23-February-2015.
- [7] "Ros home page," accessed 23-February-2015.
- [8] "PaCMan object datasets," accessed 23-February-2015.

Cups Number	Diameter
#1	55mm
#2	76mm
#3	82mm
#4	90mm
#5	98mm.

TABLE I
PROBED CUPS