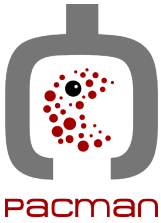

FP7-IST-60918
1 March 2013 (36months)

DR 3.1:
Control algorithms for haptic object exploration

C. Rosales, M. Bonilla, G. Santaera, E. Luberto, M. Gabiccini

Centro di Ricerca "E. Piaggio", Università di Pisa
⟨carlos.rosales@for.unipi.it⟩

Due date of deliverable:	2015-02-28
Actual submission date:	2014-02-28
Lead partner:	Università di Pisa
Revision:	draft
Dissemination level:	PU

This report describes activities related to the development of haptic object exploration methodologies. Object shape refinement and friction coefficient estimation require low-level controllers that allow contour tracing interaction as well as a sophisticated sensorization of the exploratory probe. This report presents the efforts on providing controllers to perform sliding and rolling of a fingertip over a surface, integrating controller strategy with vision, implementing a robust testbed to simulate results, and last but not least, specifying the designs on the sensorization of the Pisa/IIT SoftHand.

Executive Summary

This report describes the activities within the PaCMan consortium to define methodologies for *haptic object exploration*. The material included in this report shows the results of Task 3.1 (M 1-24). Aside, progress of Task 3.2 (M 1-36) is presented as well as the envision of a promising approach to tackle the problem within, particularly the information gathering for unknown objects or in cases where partial information is available.

Role of haptic object exploration in PaCMan

This deliverable reports the research done on finding a methodology to explore objects and gather their haptic properties such as the static and dynamic friction coefficient.

Contribution to the PaCMan scenario

The multi-modal object representation to be reported in WP2 requires information about the shape and haptic properties of an object. The active acquisition of such information is essential to build the representation of a particular object. The exploratory strategies which combine low-level control algorithms developed by the UNIPI team, high-level decision making strategies to be completed by UoB team, together with a sensorized adaptive hand as proposed from the UNIPI team, make this possible.

Additionally, the developed algorithms make no assumption on how the gathered information is encoded. This provides a versatile testbed to contrast the representation coming from WP2 with other approaches.

1 Tasks, objectives, results

1.1 Planned work

This report must show the results of Task 3.1. Particularly, it should describe reactive control strategies for haptic exploration of an object by a robotic hand, consisting of contour tracing of the object surface and finger rolling over the object surface, as well as strategies for extracting higher-order geometric features and frictional properties.

In Task 3.3 we planned to implement the active gaze model we presented in year 1 in the robot platform. This was intended to be used in the demonstration 5.2. The hypothesis to be tested is whether a reward based framework for active gaze control for grasping under positional uncertainty can be extended to work on real objects. The initial assumption was that the active gaze control model would work with a prior model of the object shape. The overall goal was to show that active gaze can improve the reliability of grasping and manipulation.

1.2 Actual work performed

We can proudly say that Task 3.1 has been accomplished, and Sec. 1.2.1 summarizes the results following the planned activities accurately.

The fact that the object shape uncertainty is reduced by making contact with the object, combined with the adaptability of the Pisa/IIT SoftHand to any kind of surface, shaped the idea of using the hand as sophisticated exploratory probe. Sec. 1.2.3 ans 1.2.2 provide glove-inspired solutions to read the hand configuration as well as the contact information to estimate accurately the object shape as the result of grasping the object. The solution can help greatly in situations where only partial visual information is available, as challenged in Task 3.2, thus we believe that this research line deserves further attention as seems beneficial to accomplish the project goals.

Work on grasping of novel objects in work package 4 progressed much faster than we expected. Thus this was taken as the grasping method to be combined with active gaze. The active gaze approach taken has been informed by the fact that experiments in our IJRR submission for WP4 have shown us that grasp failure is typically driven by the incompleteness of the point cloud near to suggested grasp points and along the final grasp trajectory. In addition tests with differing numbers of views of the test object prior to grasping showed that the more views one has from an object, the greater the probability of grasping success for this object. In addition we have used a wrist mounted depth camera for the active gaze method.

We are extending the reward based method from [1] to the case of incomplete point clouds. We require a measure of the proportion of the total possible coverage given by the incomplete object model. To this end we are

currently investigating the use of an octree representation, called octmap [2], for online object modelling and sensor data fusion. The octmap of a given object allows the representation of occupied (known voxels belonging to a segmented object), free (voxels that do not belong to an object), and unknown areas of the scene, which might belong to the object if they are near known and occupied voxels. We are now implementing a method for estimation of the safety of a grasp trajectory given this octmap. We are also testing different measures of the coverage of the object pertinent to a candidate grasp given the point cloud and the octmap representations.

Previously, in year one, we presented an algorithm for active gaze in the case where the object model is known, and the pose is uncertain. This was tested in simulation, and shown to fit human data. Relevant work on grasping under incomplete information is that by Bohg and Kragic [3]. There the approach is not active gaze to fill in information, but to use a symmetry prior to complete missing object parts. An active vision system for grasping is described in [**?**], but in this the main task of gaze control is to find and fixate on the object to support a visual servoing routine. A recent workshop at RSS 2014 on active information gathering for grasping was notable in that none of the papers considered active vision, but instead focussed on the type of approaches we study Task 4.4. Thus the area is underexplored.

1.2.1 Learning the haptic characteristics of objects by exploration in-hand

Without loosing generality, this report used fixture device as the supporting hand, and a 7 DOF robot as the exploratory finger. It is worth mentioning that a strategy for concurrent stable object grasping was demonstrated in DR5.1, and assumed to be given for the work presented here.

(a) Definition of control strategies for concurrent stable object grasping and object surface tracing for haptic exploration: An Arimoto-PD regulator provides the required torques to track a reference position. The position control law is

$$\boldsymbol{\tau_p} = G(\boldsymbol{q}) + K_{P_p}\left(\dot{\boldsymbol{q}}_{\boldsymbol{d}}\Delta T\right) + K_{D_p}\dot{\boldsymbol{q}}, \tag{1}$$

where $G(\boldsymbol{q})$ is the gravity load vector, $\boldsymbol{q}$ is the generalized joint variables, $\dot{\boldsymbol{q}}$ is the measured rate of joints variables, ΔT the control period, and K_{P_p},K_{D_p} two positive definite matrices.

A PID force controller is required to ensure contact during motion. The force control law is

$$\boldsymbol{\tau_f} = f_d J^T \boldsymbol{n_c} + K_{P_f}\boldsymbol{\tau_{e_n}} + K_{I_f}\int \boldsymbol{\tau_{e_n}} dt + K_{D_f}\dot{\boldsymbol{\tau}}_{e_n}, \tag{2}$$

where J is the robot Jacobian matrix, $\boldsymbol{e_n} = (f_d - f_n)\boldsymbol{n_c}$ is the normal force error, $\boldsymbol{\tau_{e_n}} = J^T(\boldsymbol{q})\boldsymbol{e_n}$ is the normal force error in terms of joints torques, $\boldsymbol{v_d}$ the desired twist of the end effector, and $\dot{\boldsymbol{q}}_{\boldsymbol{d}} = J^+\boldsymbol{v_d}$ the desired rate of angular joints.

Since the motion and force directions are orthogonal as seen in Fig. **??**, we can decouple the actions and sum up the contribution of each controller to form a hybrid force/position controller.

The strategy to slide a finger over a surface consist in assigning a reference velocity on the tangent plane at the contact point, of the same contact point. Let C be the contact point and $\{T\}$ the frame with C as origin, such that the y-z directions form the local tangent plane and the x direction is normal to the fingertip surface and pointing outwards, the desired twist of the fingertip with respect to C written in $\{T\}$ is

$$^{\boldsymbol{T}}\boldsymbol{v_C} = \begin{pmatrix} 0 \\ v_{dy} \\ v_{dz} \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{3}$$

where v_{dy} and v_{dz} are scalar components of the desired velocity in the local tangent plane, and $\sqrt{{v_{dy}}^2 + {v_{dx}}^2}$ the norm of the assigned velocity.

The desired twist of the fingertip center point, P, written in the end-effector frame $\{E\}$ is

$$^{\boldsymbol{E}}\boldsymbol{v_P} = \begin{pmatrix} \boldsymbol{I}_{3x3} & {}^{E}\hat{\boldsymbol{c}} \\ \boldsymbol{0}_{3x3} & \boldsymbol{I}_{3x3} \end{pmatrix} \begin{pmatrix} R_{ET} & \boldsymbol{0}_{3x3} \\ \boldsymbol{0}_{3x3} & R_{ET} \end{pmatrix} {}^{\boldsymbol{T}}\boldsymbol{v_C}, \tag{4}$$

where $\boldsymbol{c}$ is a position vector from the fingertip center to the contact point, and the $\hat{()}$ operator is the skew symmetric matrix extraction from a vector operator.

The Jacobian relative to the fingertip center is used to obtain the desired joint angle rates as

$$\dot{\boldsymbol{q}}_{\boldsymbol{d}} = J(\boldsymbol{q})^{\boldsymbol{E}}\boldsymbol{v_P} \tag{5}$$

The position controller receives as a set-point the joint angles updated as

$$\boldsymbol{q_d} = \dot{\boldsymbol{q}}_{\boldsymbol{d}}\Delta_T + \boldsymbol{q}, \tag{6}$$

where Δ_T is the control period.

(b) Definition of control strategies for concurrent stable object grasping and rolling manipulation to acquire second-order object surface properties Based on the previous control framework, rolling over a surface is achieved by changing reference velocity of the contact point for

a pure revolution about an axis on the tangent plane at the contact point. Similar to the previous section, the desired twist of the contact point C written with respect to $\{T\}$ is

$$^{\boldsymbol{T}}\boldsymbol{v_C} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \omega_{dy} \\ \omega_{dz} \end{pmatrix}, \tag{7}$$

where ω_{dy} and ω_{dz} are the scalar components of the desired rolling velocity, and $\sqrt{{\omega_{dy}}^2 + {\omega_{dx}}^2}$ is the norm of the assigned velocity.

The desired twist of the fingertip center and the desired joint angle rates rate of joint angles are obtained as above.

To estimate the curvature (a second-order property) of the unknown surface, it's possible to use the Montata's equations for two object in pure rolling contact, namely

$$M_f \dot{\alpha}_f = (K_f + \tilde{K}_0)^{-1} \begin{bmatrix} -\omega_y \\ \omega_x \end{bmatrix}, \tag{8}$$

where $\dot{\alpha}_f$ is the velocity of the contact point in the local surface chart [measured], M_f is the metric form of the fingertip surface [known], K_f is the curvature form of the fingertip surface [known], ω_x and ω_y] are the rolling component of the relative angular velocity of the Gauss frame [measured], and $\tilde{K}_0$ is the curvature form of the object surface [to be estimated].

For a finite rolling motion, it is possible to approximate (8) with

$$M_f \Delta\alpha_f = \bar{K}_r \begin{bmatrix} -\Delta\theta_y \\ \Delta\theta_x \end{bmatrix}, \tag{9}$$

where

$$\bar{K}_r = (K_f + \tilde{K}_0)^{-1} = \begin{pmatrix} r_1 & r_2 \\ r_2 & r_3 \end{pmatrix}, \tag{10}$$

or equivalently,

$$\boldsymbol{b} = A\boldsymbol{r}, \tag{11}$$

with

$$\boldsymbol{b} = M_f \Delta\alpha_f, \tag{12}$$

$$A = \begin{pmatrix} -\Delta\theta_y & \Delta\theta_x & 0 \\ 0 & -\Delta\theta_y & \Delta\theta_x \end{pmatrix}, \tag{13}$$

and

$$\boldsymbol{r} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}. \tag{14}$$

Therefore, using the generalized inverse as

$$\boldsymbol{r} = A^{+}\boldsymbol{b}, \tag{15}$$

it is possible to estimate the curvature form of the object surface,

$$\tilde{K}_0 = \bar{K}_r^{-1} - K_f, \tag{16}$$

and by diagonalization,

$$K_0 = R^T(\psi)\tilde{K}_0 R(\psi) \tag{17}$$

obtain the principal curvature directions given by ψ, and principal curvatures at the contact point, $\tilde{K}_0$.

(c) Definition of strategies for frictional coefficient estimation: Using the strategy described in (a), one can compensate for the tangential force generated due to the frictional property of the object. The tangential force can be extract from the measured contact force as

$$f_t = (I - n_c v_c^T) f_c, \tag{18}$$

and the dynamic friction coefficient at the contact point can be readily obtained as

$$\mu_d = \|f_t\|/\|f_n\|. \tag{19}$$

Using a look-up table strategy, and knowing the fingertip material, one can query a coefficient table with the pair material-dynamic coefficient and obtain the static as a well as the object material.

Finally, the compensation will be achieved by adding the term

$$\boldsymbol{\tau}_f = -J^T \mathbf{f}_t. \tag{20}$$

(d) Testing in simulation: The simulation framework is heavily based on Adams MSC©, a software for multi-body dynamic simulation and collision management. It uses several contact models based on a penalty formulation on the inter-penetration between meshes. The geometry engine is responsible for detecting contact between two geometries, locating the points of contact, and calculating the common normal at the contact points. Once the contact kinematics are known, contact forces, which are a function of the contact kinematics, are applied to the intersecting bodies. The model adopted is a Hertzian model, named *Impact* by Adams (see Fig. 1).

Another feature is the capability to export a Simulink block representing the articulated model with inputs (e.g. joint torques) and outputs (e.g. joint angles, contact wrench). Matlab and Simulink were used to implement the control laws and to manage the results.

joint	θ	d [m]	a [m]	α	Mass [Kg]	I_{zz} [Kgm]	CoG [m]
1	q_1	0	0.05	0	0.02	$5e^{-6}$	$(0, -0.0250, 0)^T$
2	q_2	0	0.05	0	0.02	$5e^{-6}$	$(0, -0.0250, 0)^T$
3	q_3	0	0.05	0	0.02	$5e^{-6}$	$(0 - 0.02500)^T$

Table 1: DH and dynamic parameters of the 3R finger.

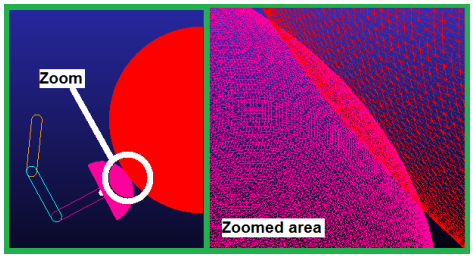

Figure 1: Contac model in Adams.

Parameter	**Value**
$K_{P_p}; K_{D_p}$	$100; 20$
$K_{P_f}; K_{D_f}; K_{I_f}$	$10; 1; 0.1$
Stiffness	$1000\frac{N}{m}$
Damping	$20\frac{Ns}{m}$

Table 2: Test parameters.

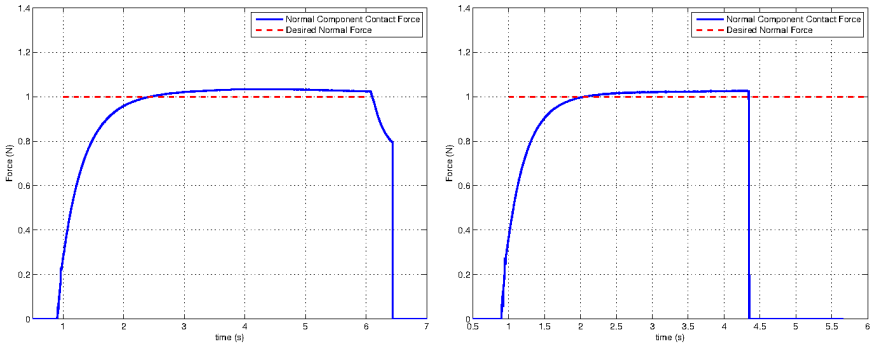

Figure 2: Normal component of the contact force when sliding over a plane (left) and a sphere (right).

Thus, a planar finger with three angular joints was modeled using the parameters in Table 1. The fingertip is a hemisphere of radius 3cm.

The controller gains for position and force regulation and contact parameters between the fingertip and the objects were set as in Table 2. The desired normal component contact force was set to 1N for all cases.

Fig. 2 shows the track of the normal component of the contact force for the sliding on a plane and a sphere surface. And Fig. 3 shows the track of the normal and tangential force during rolling over the a plane.

The tests were taken to a real scenario shown in Fig. 1.2.1, where the KUKA LWR was equipped with an ATI Nano 17 force torque sensor. The estimated principal curvatures are summarized in Table 3. Note that the sphere is a basketball, whose official radius is 11,92-12,07cm, as well as the large values for the plane and one of the pricnipal curvatures on the cylinder.

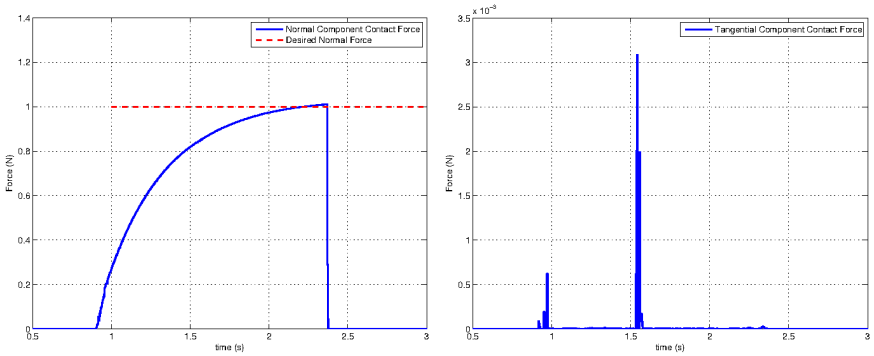

Figure 3: Normal (left) and tangential (right) component of the contact force when rolling over a plane.

Object / Direction	Real Curvature x (m)	Real Curvature y (m)	Measured Curvature x (m)	Measured Curvature y (m)
Plane	∞	∞	333	500
Cylinder	∞	.	500	.088
Sphere	.	.	0.121	0.127

Table 3: Experimental Results

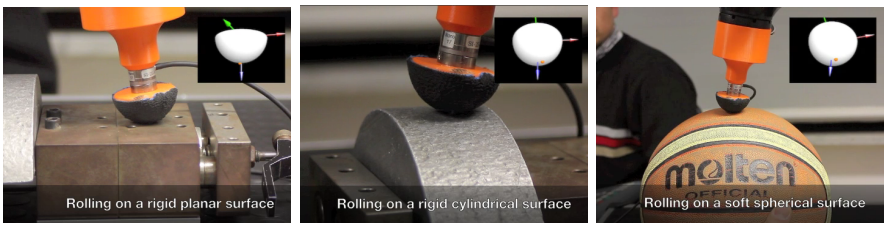

Figure 4: Test objects for curvature estimation test on a plane (left), a cylinder (middle), and a sphere (right).

(e) Integration of object/part model obtained from vision with the code for reactive haptic exploration strategies The setup overview is depicted in Fig. 5. The approach used Gaussian Processes to represent both, shape and fiction coefficient, and it was successfully exploited to generate exploration trajectories on object surface with arbitrary shapes as illustrated in Fig. 6. The normal contact force was used to discriminate the contact states from the non-contact state. In Fig. 7, the results from the box sliding operation are summarized.

More details on the whole methodology can be found in the attached paper [4] available at this link.

1.2.2 Endowing the Pisa/IIT SoftHand with the sense of touch

(a) The Madgwick Filter In [5] was used a modified version of the Mahony-Hamel passive complementary filter, to obtain the orientation of a generic frame $\{A\}$ with respect another one $\{B\}$ expressed by a rotation matrix.

In his works Sebastian Madgwick [6] studies a new algorithm able to better use measures read by sensors. Algorithm proposed is also able to tackle the singularities, associated for example with Euler angle representation, using quaternions to represent a general orientation. The Madgwick filter is similar to Mahony-Hamel filter, in each time a new orientation quaternion is estimated summing to its previously estimation a correction factor, which depends by data read from IMU.

¡¡¡¡¡¡¡ Updated upstream 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 ======= 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 ¿¿¿¿¿¿¿ Stashed changes

$$ {}^b\Omega = [0 \quad {}^b\omega_x \quad {}^b\omega_y \quad {}^b\omega_z], \tag{21} $$

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

$$ {}^b_a\dot{q} = \frac{1}{2} {}^b_a\bar{q} \otimes {}^b\Omega. \tag{22} $$

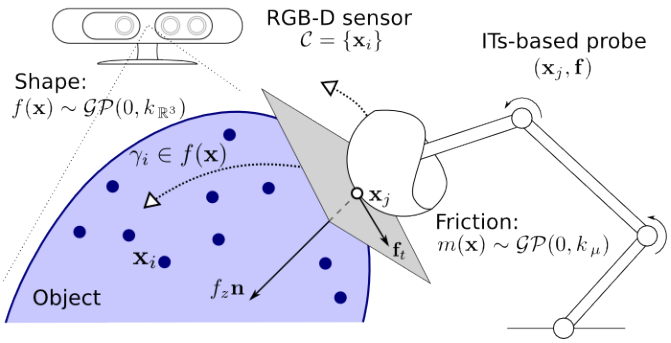

Figure 5: The shape is acquired by an RGBD sensor and the friction coefficients by the intrinsic tactile sensor, both shape $f(\mathbf{x})$ and friction $m(\mathbf{x})$ functions are represented as a Gaussian process $\mathcal{GP}(0, k(\cdot))$. The exploration follows geodesic flows γ_i on the surface of a fixed object using a compliant behavior.

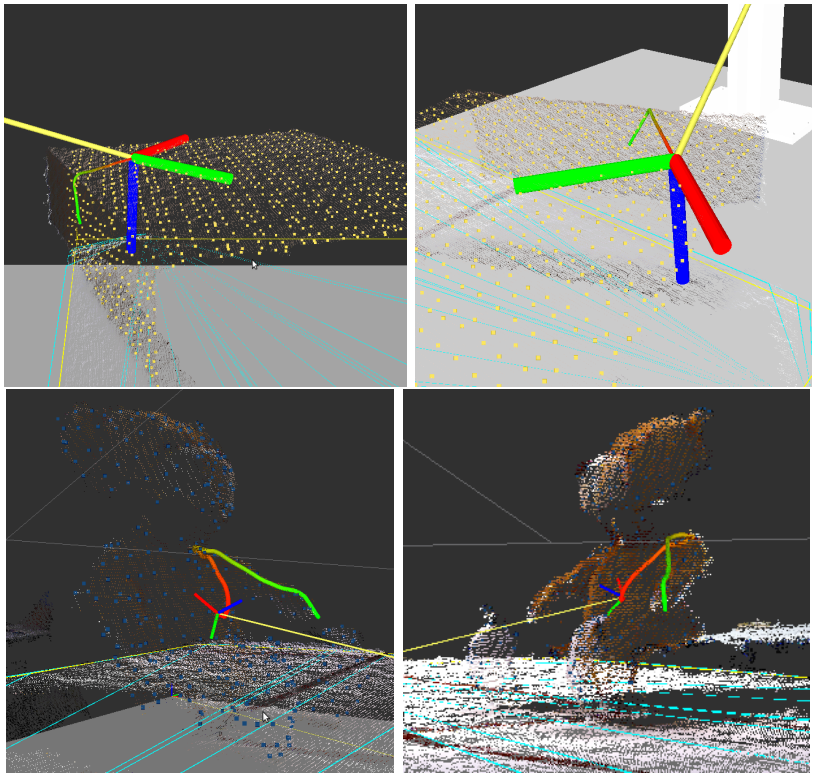

Figure 6: Geodesic flows on object surfaces captured by an RGBD sensor. The color of the flow goes from red, i.e. the initial contact point, to green, i.e. the final position, according to the predefined length of the curve. The box has flat surfaces, hence geodesics are straight lines (top). The teddy bear has a very irregular shape, but still geodesic flows are found (bottom).

Where $\otimes$ denotes a quaternion product, while $^-$ denotes the unity normalize operator. From eq. (22) trivially the orientation of the earth frame with respect to sensor frame at time t is given by eq. (23) and (24) as

$$ {}^b_a\dot{q}_{\Omega,t} = \frac{1}{2} {}^b_a\hat{\bar{q}}_{t-1} \otimes {}^b\,\Omega_t, \tag{23} $$

$$ {}^b_a q_{\Omega,t} = {}^b_a\hat{\bar{q}}_{t-1} + {}^b_a\dot{q}_{\Omega,t}\Delta t, \tag{24} $$

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

Reading now from a sensor a set of accelerometer and compass measures in a frame strap down to the sensor there are infinite earth frame orientation, according to measures read from the sensors. However using quaternions to express an orientation it is very easy to overcome to the singularities problem, obtaining from sensors meausures an unique orientation for the earth frame with respect the sensor one. From this considerations the orientation

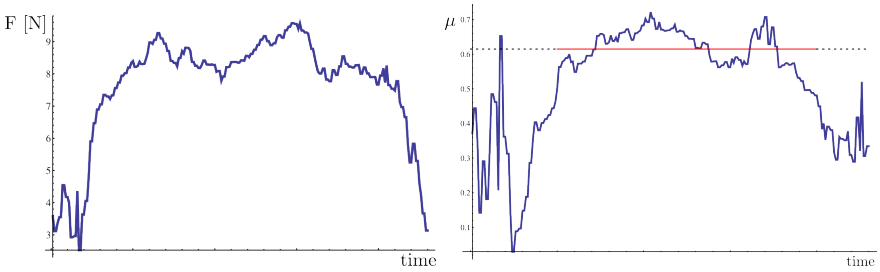

Figure 7: Relevant data recorded during a haptic exploration. Force along $\mathbf{z}_c$ used to detect the contact and non-contact states (top). Friction coefficient μ, the red line marks the mean value during the contact state (bottom). Material: Paperboard.

problem can be written as an optimisation 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 ${}^b\bar{s}$. So the quaternion ${}^b_a\bar{q}$ is given by the problem

$$\min_{{}^b_a\bar{q}\in\mathbb{R}^4} f({}^b_a\bar{q}, {}^a\bar{d}, {}^b\bar{s}), \tag{25}$$

with the objective function f, ${}^b_a\bar{q}$ and the vectors ${}^a\bar{d}$ and ${}^b\bar{s}$ defined as

$$f({}^b_a\bar{q}, {}^a\bar{d}, {}^b\bar{s}) = {}^b_a\bar{q}^* \otimes {}^a\bar{d} \otimes {}^b_a\bar{q} - {}^b\bar{s}, \tag{26}$$

$${}^b_a\bar{q} = [q_1 \quad q_2 \quad q_3 \quad q_4], \tag{27}$$

$${}^a\bar{d} = [0 \quad d_x \quad d_y \quad d_z], \tag{28}$$

$${}^b\bar{s} = [0 \quad s_x \quad s_y \quad s_z], \tag{29}$$

and * in (26) denotes the conjugate operator of a quaternion $q = [q_1 \quad q_2 \quad q_3 \quad q_4]$ as $q^* = [q_1 \quad -q_2 \quad -q_3 \quad -q_4]$. To solve problem in (25) many algorithms can be used, but in terms of simplicity the gradient descent algorithm should be the best choice. From this last choice and starting from an initial quaternion ${}^b_a\bar{q}_0$ it is possible to write the quarternion correction factor ${}^b_a\bar{q}_{k+1}$ linked to step $(k+1)^{th}$ as

$${}^b_a q_{k+1} = {}^b_a\bar{q}_k - \beta \frac{\nabla f({}^b_a\bar{q}_k, {}^a\bar{d}, {}^b\bar{s})}{||\nabla f({}^b_a\bar{q}_k, {}^a\bar{d}, {}^b\bar{s})||}, \quad k = 0, 1, 2 \ldots n, \tag{30}$$

where

$${}^b_a\bar{q}_k = \frac{1}{2} {}^b_a\bar{q}_{k-1} \otimes {}^b\Omega_k, \quad k = 1, 2, 3 \ldots n, \tag{31}$$

$$\nabla f({}^b_a\bar{q}_k, {}^a\bar{d}, {}^b\bar{s}) = J^T({}^b_a\bar{q}_k, {}^a\bar{d}) f({}^b_a\bar{q}_k, {}^a\bar{d}, {}^b\bar{s}), \tag{32}$$

$$f({}^b_a\bar{q}_k, {}^a\bar{d}, {}^b\bar{s}) = \begin{bmatrix} 2d_x(\frac{1}{2} - q_3^2 - q_4^2) + 2d_y(q_1q_4 + q_2q_3) + \\ 2d_x(q_2q_3 - q_1q_4) + 2d_y(\frac{1}{2} - q_2^2 - q_4^2) + \\ 2d_x(q_1q_3 + q_2q_4) + 2d_y(q_3q_4 - q_1q_2) + \\ 2d_z(q_2q_4 - q_1q_3) - s_x \\ 2d_z(q_1q_2 + q_3q_4) - s_y \\ 2d_z(\frac{1}{2} - q_2^2 - q_3^2) - s_z \end{bmatrix}, \tag{33}$$

$$J({}^b_a\bar{q}_k, {}^a\bar{d}) = \begin{bmatrix} 2d_yq_4 - 2d_zq_3 & 2d_yq_3 + 2d_zq_4 \\ -2d_xq_4 + 2d_zq_2 & 2d_xq_3 - 4d_yq_2 + 2d_zq1 \\ 2d_xq_3 - 2d_yq_2 & 2d_xq_4 - 2d_yq_1 - 4d_zq_2 \\ -4d_xq_32d_yq_2 - 2d_zq_1 & -4d_xq_42d_yq_1 + 2d_zq_2 \\ 2d_xq_2 + 2d_zq_4 & -2d_xq_1 - 4d_yq_4 + 2d_zq_3 \\ 2d_xq_1 + 2d_yq_4 - 4d_zq_3 & 2d_xq_2 + 2d_yq_3 \end{bmatrix}, \tag{34}$$

where β denotes the step size, while J^T denotes the transpose of the Jacobian matrix from the objective function f.

Problem described in eq. (25) is a general alignement problem, so considering measurements read from accelerometers and from magnetormeters, it is possible to write two different problems, the first one from accelerometers data $f_g({}^b_a\overline{q}_k, {}^a\overline{g}_a, {}^b\overline{g}_b)$ where ${}^a\overline{g}_a = \overline{g}_a$ is the gravity field measured in the inertial frame while ${}^b\overline{g}_b = \overline{g}_b$ is the gravity field in the sensor frame. In the same manner the second problem is $f_m({}^b_a\overline{q}_k, {}^a\overline{m}_a, {}^b\overline{m}_b)$ with ${}^a\overline{m}_a = \overline{m}_a$ magnetic field read in the inertial frame and ${}^s\overline{m}_b = \overline{m}_b$ magnetic field read in the sensor frame. Combining the two optimization problems as

$$f_{g,m}({}^b_a\overline{q}, \overline{g}_a, \overline{g}_b, \overline{m}_a, \overline{m}_b) = \begin{bmatrix} f_g({}^b_a\overline{q}, \overline{g}_a, \overline{g}_b) \\ f_m({}^b_a\overline{q}, \overline{m}_a, \overline{m}_b) \end{bmatrix}, \tag{35}$$

it is possible to find an unique quaternion which describes the orientation, at step $(k+1)^{th}$, of the inertial frame with respect the sensor frame as

$${}^b_aq_{k+1} = {}^b_a\overline{q}_k - \beta \frac{\nabla f_{g,m}({}^b_a\overline{q}_k, \overline{g}_a, \overline{g}_b, \overline{m}_a, \overline{m}_b)}{||\nabla f_{g,m}({}^b_a\overline{q}_k, \overline{g}_a, \overline{g}_b, \overline{m}_a, \overline{m}_b)||}, \quad k = 0, 1, 2 \dots n, \tag{36}$$

with

$$\nabla f_{g,m}({}^b_a\overline{q}_k, \overline{g}_a, \overline{g}_b, \overline{m}_a, \overline{m}_b) = J^T_{g,m}({}^b_a\overline{q}_k, \overline{g_a}, \overline{m_a}) \\ f_{g,m}({}^b_a\overline{q}_k, \overline{g}_a, \overline{g}_b, \overline{m}_a, \overline{m}_b), \tag{37}$$

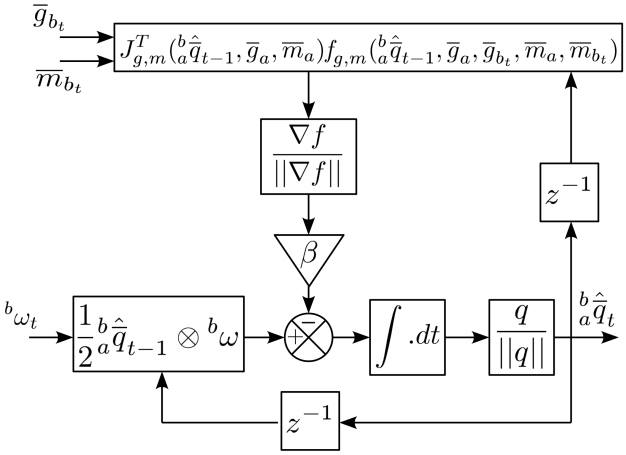

Figure 8: Block diagram of the Madgwick passive complementary filter

and

$$J^T_{g,m}({}^b_a\overline{q}_k, \overline{g_a}, \overline{m_a}) = \begin{bmatrix} J_g({}^b_a\overline{q}_k, \overline{g}_a) \\ J_m({}^b_a\overline{q}_k, \overline{m}_a) \end{bmatrix} \tag{38}$$

where J_g and J_m respectvily the Jacobians of the function f_g and f_m.

In figure 8 is shown the block diagram of the Madgwick filter, while algorithm 1 details the steps followed to obtain the orientation quaternion using its previous estimation and data read from IMU, using g_a and m_a to denote respectively the gravity and the magnetic field in the inertial frame. In particular the gravity field commonly is set as $g_a = [0 \quad 0 \quad 0 \quad 1]$, or rather using a North-East-Down convention. For the earth magnetic field it can be considered to have two components in one horizontal axis and the vertical axis, with its vertical component due to the inclination of the field depending by the latitude and it is about $1°$ to the horizontal in Pisa, so $m_a = [0 \quad m_{a_x} \quad 0 \quad m_{a_z}]$.

Orientation between two IMUs Using two different IMUs in a configuration as shown in figure 9 these will return two different set of measurements (r_1, r_2) referred to their body frame $\{B_1\}$ and $\{B_2\}$. Applying the Madgwick filter to r_1 this one will return ${}^{b_1}_a\hat{\overline{q}}$,or rather the orientation quaternion of the inertial frame with respect the frame $\{B_1\}$, while from r_2 the filter will return ${}^{b_2}_a\hat{\overline{q}}$, or rather the orientation quaternion of the inertial frame with respect the frame $\{B_2\}$. Trivially the orientation quaternion of the frame $\{B_2\}$ with respect the frame $\{B_1\}$ is given by

$${}^{b_1}_{b_2}\hat{\overline{q}} = {}^{b_2}_a\hat{\overline{q}}^* \otimes {}^{b_1}_a\hat{\overline{q}}. \tag{39}$$

However, as seen for the Mahony-Hamel filter, it is possible to apply in a

Algorithm 1: Madgwick Discrete Filter at n^{th} step

1: Reading the current values of the accelerometers (g_{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 $\overline{g}_{b_n} = \frac{g_{b_n}}{||g_{b_n}||}$, $\overline{m}_{b_n} \frac{m_{b_n}}{||m_{b_n}||}$
3: Computing the objective function value $f_{g,m}({}^{b}_{a}\hat{\bar{q}}_{n-1}, \overline{g}_a, \overline{g}_{b_n}, \overline{m}_a, \overline{m}_{b_n})$ using equations (33) and (35)
4: Computing the transpose of the Jacobian of the objective function $J^T_{g,m}({}^{b}_{a}\hat{\bar{q}}_{n-1}, \overline{g_a}, \overline{m}_a)$ using equations (34) and (38)
5: Computing the correction terms $c_n = \beta \frac{\nabla f}{||\nabla f||}$ using equation (37)
6: Computing the orientation quaternion time variation ${}^{b}_{a}\hat{q}_{d_n} = (\frac{1}{2}{}^{b}_{a}\hat{\bar{q}}_{n-1} \otimes \Omega_{b_n}) - c_n$
7: Computing the new orientation quaternion estimation given by its previouly one and its current time variation ${}^{b}_{a}\hat{q}_n = {}^{b}_{a}\hat{\bar{q}}_{n-1} + {}^{b}_{a}\hat{q}_{d_n}\Delta t$
8: Normalizing the new estimated quarternion ${}^{b}_{a}\hat{\bar{q}}_n = \frac{{}^{b}_{a}\hat{q}_n}{||{}^{b}_{a}\hat{q}_n||}$

suitable way the Madgwick filter to data read from the two IMUs, to obtain ${}^{b_1}_{b_2}\hat{\bar{q}}$.

Starting from the IMUs measurements where $\overline{g}_{b_1}$, ${}^{b_1}\Omega$ and $\overline{m}_{b_1}$ denotes respectively accelerometer, gyroscope and magnetometer measurements in $\{B_1\}$, while $\overline{g}_{b_2}$, ${}^{b_2}\Omega$ and $\overline{m}_{b_2}$ denotes respectively accelerometer, gyroscope and magnetometer measurements in $\{B_2\}$, the first term the correction factor (30), written in (31) become

$$ {}^{b_1}_{b_2}\bar{q}_k = \frac{1}{2}{}^{b_1}_{b_2}\bar{q}_{k-1} \otimes {}^{b_1}_{b_2}\Omega_k, \quad k = 1, 2, 3 \dots n, \tag{40} $$

where

$$ {}^{b_1}_{b_2}\Omega_k = {}^{b_2}\Omega_k - {}^{b_1}_{b_2}\bar{q}_{k-1} \otimes {}^{b_1}\Omega_k \otimes {}^{b_1}_{b_2}\bar{q}^*_{k-1}, \tag{41} $$

is the angular rate of the frame $\{B_2\}$ attached to the IMU_2 with respect the frame $\{B_1\}$ attached to IMU_1. The second term of the correction factor, in (37) become

$$ \begin{aligned} \nabla f_{g,m}({}^{b_1}_{b_2}\bar{q}_k, \bar{g}_{b_2}, \bar{g}_{b_1}, \bar{m}_{b_2}, \bar{m}_{b_1}) = J^T_{g,m}({}^{b_1}_{b_2}\bar{q}_k, \bar{g_{b_2}}, \bar{m_{b_2}}) \\ f_{g,m}({}^{b_1}_{b_2}\bar{q}_k, \bar{g}_{b_2}, \bar{g}_{b_1}, \bar{m}_{b_2}, \bar{m}_{b_1}). \end{aligned} \tag{42} $$

So the orientation quaternion of the frame $\{B_2\}$ with respect the frame $\{B_1\}$ at step $(k+1)^{th}$ is given by

$$ {}^{b_1}_{b_2}q_{k+1} = {}^{b_1}_{b_2}\bar{q}_k - \beta \frac{\nabla f_{g,m}({}^{b_1}_{b_2}\bar{q}_k, \bar{g}_{b_2}, \bar{g}_{b_1}, \bar{m}_{b_2}, \bar{m}_{b_1})}{||\nabla f_{g,m}({}^{b_1}_{b_2}\bar{q}_k, \bar{g}_{b_2}, \bar{g}_{b_1}, \bar{m}_{b_2}, \bar{m}_{b_1})||}, \quad k = 0, 1, 2 \dots n. \tag{43} $$

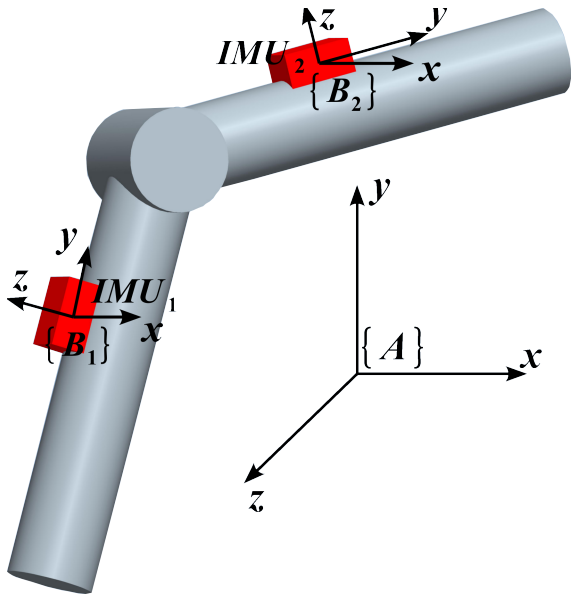

Figure 9: Simple structure with two link connected by a revolute joint

In algorithm 2 are shown steps followed to obtain the orientation quaternion of a frame $\{B_2\}$ with respect frame $\{B_1\}$.

(b) Hardware In this section are described the sensors used and how these ones are managed and used by the filter. Information about acceleration, angular rate and magnetic field are read by an IMU (Inertial Measurements Unit). The low power, light weight and potential for low cost manufacture of these sensors opens up a wide range of solution. To reconstruct the hand posture 17 IMU are rigidly constrain to a glove dressed by the hand. In particular is mounted one device for each phalanges of the fingers ($3x5 = 15$), one on the palm and the last one on the wrist of the glove.

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 Invensence [7].

MPU-9250 The MPU-9250 is a System in Package device (SiP) that combines two chip: the MPU-6500 device (used in the first feasibility study [5]) 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

Algorithm 2: Two IMUs Madgwick Discrete Filter at n^{th} step

1: Reading the current values of the accelerometers ($g_{b_1 n}$), magnetometers ($m_{b_1 n}$) and gyro rates ($\Omega_{b_1 n}$) in the local IMU_1 frame $\{B_1\}$
2: Normalizing gravity and magnetic field vector read from IMU_1 $\overline{g}_{b_1 n} = \frac{g_{b_1 n}}{||g_{b_1 n}||}$, $\overline{m}_{b_1 n} \frac{m_{b_1 n}}{||m_{b_1 n}||}$
3: Reading the current values of the accelerometers ($g_{b_2 n}$), magnetometers ($m_{b_2 n}$) and gyro rates ($\Omega_{b_2 n}$) in the local IMU_2 frame $\{B_2\}$
4: Normalizing gravity and magnetic field vector read from IMU_2 $\overline{g}_{b_2 n} = \frac{g_{b_2 n}}{||g_{b_2 n}||}$, $\overline{m}_{b_2 n} \frac{m_{b_2 n}}{||m_{b_2 n}||}$
5: Computing the objective function value $f_{g,m}({}^{b_1}_{b_2}\hat{\overline{q}}_{n-1}, \overline{g}_{b_2 n}, \overline{g}_{b_1 n}, \overline{m}_{b_2 n}, \overline{m}_{b_1 n})$ using equations (33) and (35)
6: Computing the transpose of the Jacobian of the objective function $J^T_{g,m}({}^{b_1}_{b_2}\hat{\overline{q}}_{n-1}, \overline{g}_{b_2 n}, \overline{m}_{b_2 n})$ using equations (34) and (38)
7: Computing the correction terms $c_n = \beta \frac{\nabla f}{||\nabla f||}$ using equation (37)
8: Computing the orientation quaternion time variation ${}^{b_1}_{b_2}\hat{q}_{d_n} = (\frac{1}{2} {}^{b_1}_{b_2}\hat{\overline{q}}_{n-1} \otimes {}^{b_1}_{b_2}\Omega_k) - c_n$ using (41)
9: Computing the new orientation quaternion estimation given by its previouly one and its current time variation ${}^{b_1}_{b_2}\hat{q}_n = {}^{b_1}_{b_2}\hat{\overline{q}}_{n-1} + {}^{b_1}_{b_2}\hat{q}_{d_n} \Delta t$
10: Normalizing the new estimated quarternion ${}^{b_1}_{b_2}\hat{\overline{q}}_n = \frac{{}^{b_1}_{b_2}\hat{q}_n}{||{}^{b_1}_{b_2}\hat{q}_n||}$

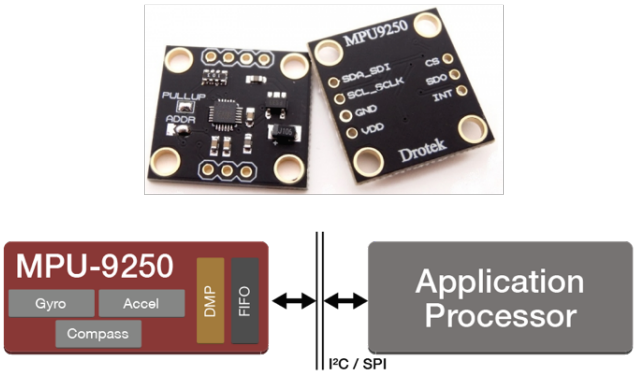

Figure 10: MPU-9250

proof mass, and capacitive sensors detect the displacement differentially. The 3-analog informations are digitized using individual on-chip 16-bit Analog-to-Digital Converers(ADCs) to sample each axis;

- *Gyroscope*: There are three indipendet vibratory rate gyroscopes, wich detect rotation about the X, Y and Z axes. When the gyroscope are rotated about any of sense axes, the Coriolis Effect 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, pass thorugh a ADC providing digital outputs.

- *Magnetometer*: The 3-axis magnetometer uses higly 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 MPU-9250's technical features are summarized in the table 4.

The output data from axes sensors can be read, in digital way, from a 8-bit register of the device. Each axis sensor presents two registers, up and low register. Thus a complete information from a specific sensor occupies 48-bit, considering for exaple the accelerometer there 16-bit for the x axis, 16-bit for the y axis and 16-bit for the z axis.

The figure 11 shows the orientation of the axes of sensitivity and the polarity of rotation.

As said a complete sensorization of the glove and then of the hand needs 17 IMU, so in terms of system time response the communication between a master processing unit and each IMU becomes a crucial aspect. The MPU-9250 supports two different type of digital communication; I^2C (Inter Integreted Circuit) and SPI (Serial Peripheral Interface). The I^2C maximum

Part	**Unit**	
Gyro Full Scale Range	(deg/sec)	±250 ±500 ±1000 **±2000**
Gyro Rate Noise	(dps/$\sqrt{\text{Hz}}$)	0.01
Accel Full Scale Range	(g)	±2 ±4 **±8** ±16
Compass Full Scale Range	(μT)	**±4800**
Digital Output		I^2C **SPI**
Logic Supply Voltage	(V)	1.7VtoVDD **VDD**
Package Size	(mm)	3x3x1

Table 4: MPU-9250 features

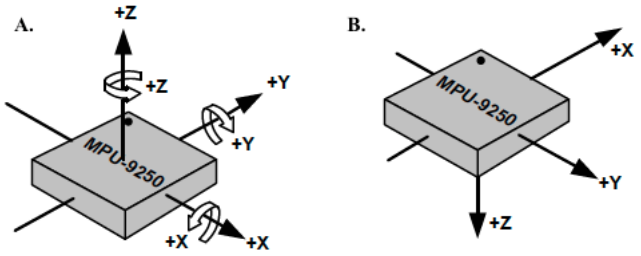

Figure 11: A. accelerometer and gyro orientation axes B. magnetometer oreintation axes

working frequency is 400Hz while the SPI works on 1Mhz. So the SPI allows a faster communication and using the a futher pin (Slave-Select pin) allows the user to use a single bus overcoming to the problem of set an unique address for each sensor.

In the figure 12 is showed the pin-out of a generic SPI cominication. In particular the SPI is a 4-wire synchronous serial interface that uses two control lines and two data lines:

- SCLK, clock - control line;
- MOSI, master-output slave-input - data line;
- MISO, master-input slave-output - data line;
- SS, slave select - control line.

In our work MPU-9250 always operates as a Slave device during standard Master-Slave SPI operation. To speed up communication between master and sensors in our works three SPI bus are used (reffig:firmwarepage1, 13).

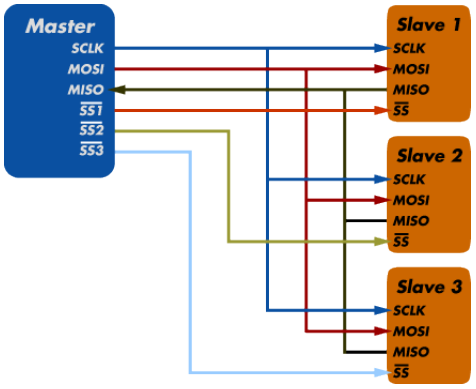

Figure 12: Scheme of SPI communication

In each bus the SCLK, the MISO and the MOSI pin are shared by all slaves devices, while each slave devices requires its own SS line to communicate with the master. The SS pin goes low, (active) when the transmission start and goes back high (inactive) at the end, so only one SS line is active at a time, ensuring that only one slave is selected at any given time.

Usually the master is a microcontroller or any processing unit that support SPI communication. To manage all 17 slave devices, it is necessary a microcontroller with sufficient number of pins. The microcontroller must be able not only to read the correct data from the IMUs but also it must send data packages to the PC, where the Magdwick Filter is implemented.

PSoC 5LP The processing unit used is a PSoC mounted on the *PSoC 5LP* board developed by Cypress Semiconductor [8]. 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 different from other microcontrollers, for this reason can be associated to the FPGA microcontroller family (Field-Programmable Gate Array).

By using configurable analog and digital blocks, it is possible to create and change mixed-signal embedded applications. These blocks are designed by PSoC-Creator, an Integreted Design Environment (IDE).

The first task of the microcontroller is to read data from the slave devices

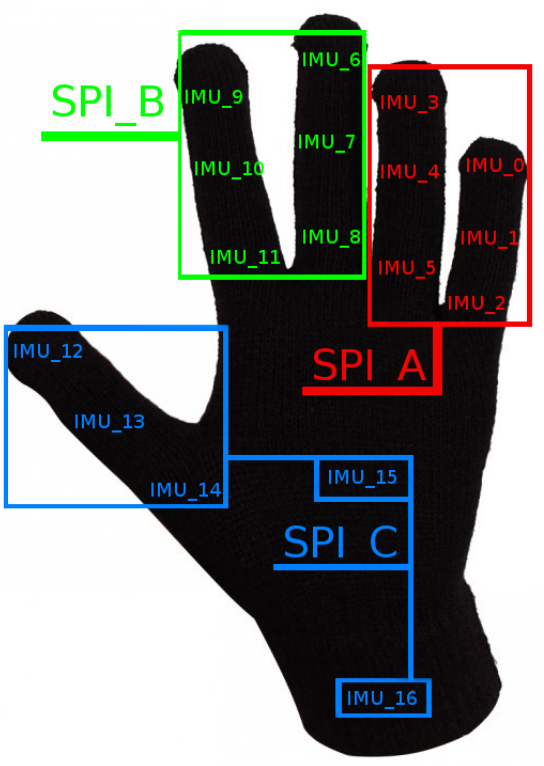

Figure 13: SPI bus used in the IMU Glove

allowing a SPI communication. The SPI bus could be only one, but were preferred three separated bus to have a greater control on the hardware system and to speed up the communication.

After read all IMUs data read these ones are stored in the EEPROM of the PSoC5 and are ready to be send to the PC. To communicate with the PC the PSoC5 has a dual-channel USB. A first channel *A* connected to the PSoC 5 in FIFO parallel mode to allow a fastest transfers between the USB host PC and the PSoC 5 and a channel *B* connected to the PSoC 5 in serial mode to allow a standard UART communication between the host PC and the PSoC5. However the channel A, commonly used to implement fast a communication, has a little communication buffer composed by 64 byte, while to send all IMU data to PC we need more than 450 byte. The channel B instead is a serial RS-232 channel and need a Serial-USB conveter to be connected to the PC. The PSoC 5LP mount on board an internal Serial-USB converter but its maximum speed is 115200 baud while we need more than 900000 baud. Starting from this considerations a further solution was implemented using on PSoC a standard serial communication and an external Serial-USB module to connected PSoC board to PC.

Figure 14: SPI buses on the IMU glove

Serial Communication The communication between PSocC5 and PC was created exploiting the serial adapter 990 004 [9]. It is optimum to connect microcontroller and logic circuit to a PC with an high rate trasfer data. The heart of the module is the chip FT232R distributed by FTDI [10] which works with a supply tension of 3.3V, thus it possible to connect the device to another TTL peripheral o microcontroller in the range of 3.3V-5V without the problem to convert the signal RS232 to the TTL. The size of the circuit are very small, 25x18 mm and two led visualize input-output

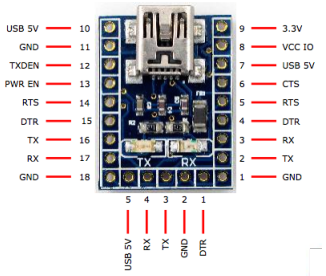

Figure 15: Serial adapter

data of serial port, very useful to control data stream, escaping possible software/hardware problems. The device is USB 2.0 compatible and it allows all velocity to 1Mb.

(c) Software To reconstruct the hand complete posture using the IMU glove two different software are written. The first one, written in C, running on the PSoC is its firmware and this one as described read all IMU measurements and then send data to a PC. The second software, written in C++, running on a PC reads data from PSoC and applies in a suitable way

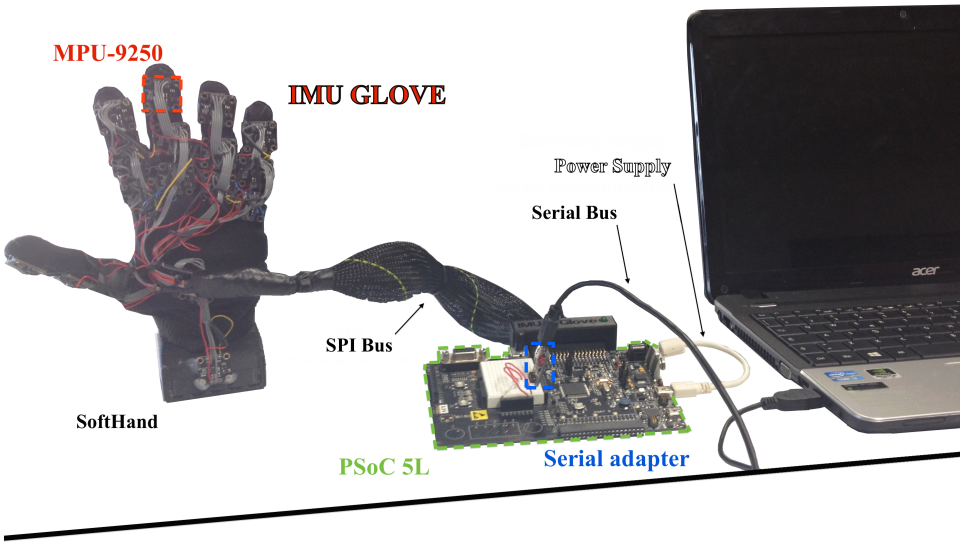

Figure 16: Full Hardware

the Madgwick filter to data from two subsequent IMU.

Firmware on PSoC The firmware is written using the PSoC creator IDE developed by Cypress. This one offers an intuitively GUI to managed 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.In figure 20 is showed how the three SPI bus are configurated and the relative pin used.

20

In figure 18 is showed how the firmware manages IMUs

In particular there is an initalization phase *Init*, who sets all IMU internal registers. This means configure e.g. internal clock, type of communication, accelerometer and gyro full scale range, ecc. After the initialization phase firmware could run in two different way. The *Magnetometer Config*, performed whenever there are new IMU, 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 IMU. So the manufacturer store in three different IMU 8-bit register three number, one for each axes, to uniform magnetic measures between two o 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

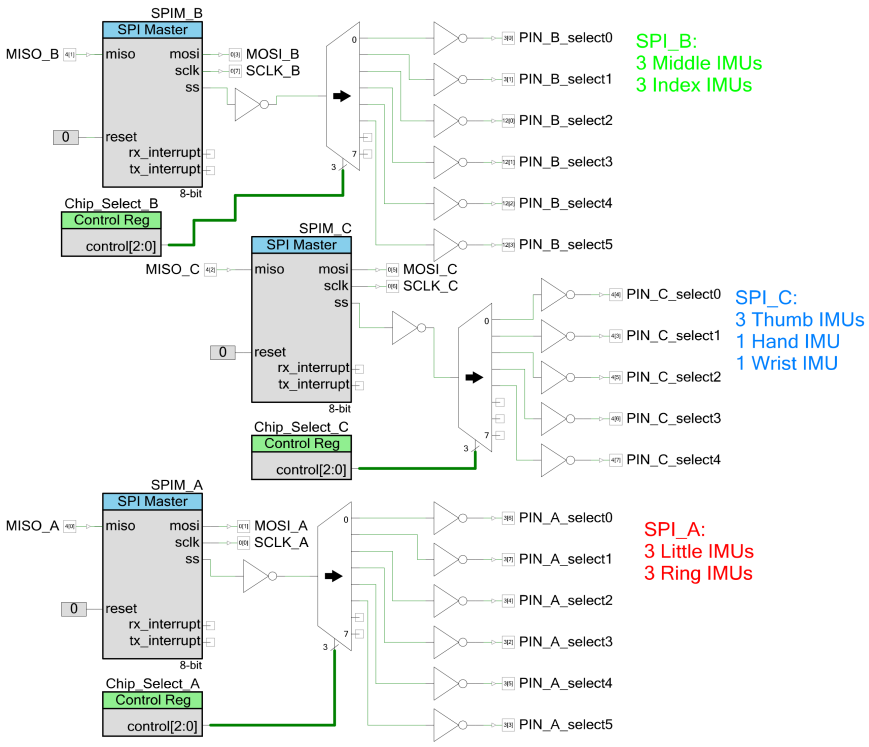

Figure 17: SPI blocks in the IDE

Read the three sensivity registers the sensitivity factor corrector s_a is given by

$$s_a = \frac{0.5(ASA_a - 128)}{128} + 1, \tag{44}$$

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

$$H_{adj_x} = H_x s_x, \tag{45}$$

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

If all magnetometer corrector factors are been computed after the initi phase, firmware will run *Infinite loop* phase. It is subdivided in two independent section

- *Read IMUs*: An internal counter each 20ms (50Hz) generates an interrupt where IMU are sequentially read and the output data stored in 3 different matrix

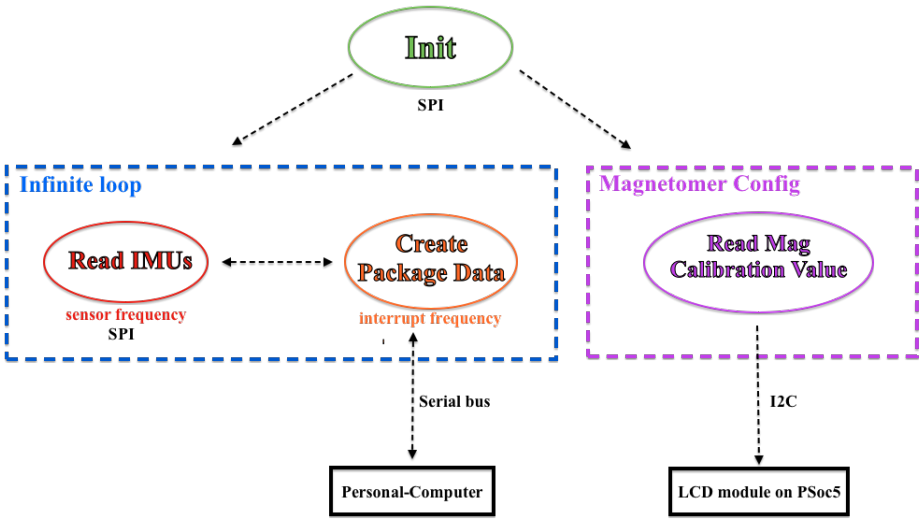

Figure 18: Scheme of Firmaware written in the PSoC5L

- Acc $\in \Re^{17x3}$
- Gyro $\in \Re^{17x3}$
- Mag $\in \Re^{17x3}$

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

· *Create Package Data*: Between two subsequent interrupt data stored in the three matrices are reorganized in a package ready to be sent to the PC. The package is summarized in the figure 19

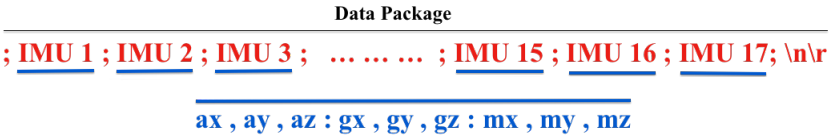

Figure 19: Data package sent to the PC

where its length is

$$\left.\begin{array}{llll} \texttt{int} & \texttt{ax}, \cdots, \texttt{mz} & = & 2\text{byte} \\ \texttt{char} & , & = & 1\text{byte} \\ \texttt{char} & : & = & 1\text{byte} \\ \texttt{char} & ; & = & 1\text{byte} \\ \texttt{char} & \backslash\texttt{r} & = & 1\text{byte} \\ \texttt{char} & \backslash\texttt{n} & = & 1\text{byte} \end{array}\right\} \begin{array}{l} \texttt{int} = 9 \cdot 17 = 306 \text{ byte} \\ \texttt{char} = 8 \cdot + \texttt{char} \cdot 20 = 156 \text{ byte} \\ \hline 462 \text{ byte} \end{array}$$

The interrupt who defines the IMU read and data send time is generated using an PWM block, while the communication with computer is implemented using an UART block, as shown in the figure 20

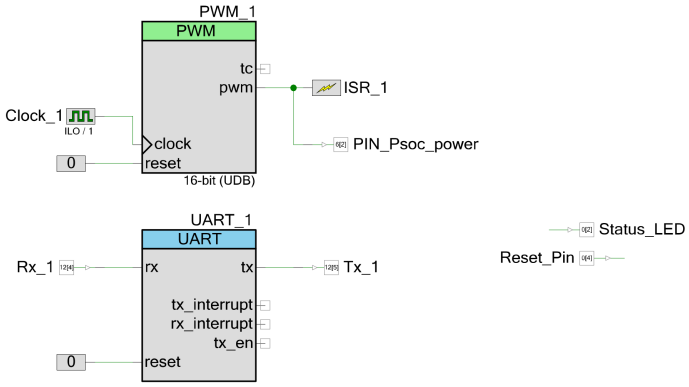

Figure 20: Interrupt and UART blocks in the IDE

C++ code on PC The code on PC implements, from data read by PSoC, the Madgwick algorithm between two IMUs, a scheme of code is shown in the figure 21.

During an initial phase the communication with the PSoC is configured, then user can decide whether to change the magnetometer calibration factors or no. If user decide to change the magnetometer calibration factors a special function to calibrate correctly magnetometer will run and the new compass correction parameter are stored in a file.txt to be load in the future. Magnetometer calibration spends about 50-60 seconds to compute all magnetometer corrector factors. If user decide to mantain the old magnetometer corrector factors, software applies the Madgwick filter to data read from the IMU to compute the offset angles between IMUs due to glove mounting inaccuracies. Then after these initiali stages software entries in a infinite loop where runs computing at each time the joint angles values from the current IMU measurements.

Once detected the offset quaternions, the infinite loop starts calculating the desired joint angles.

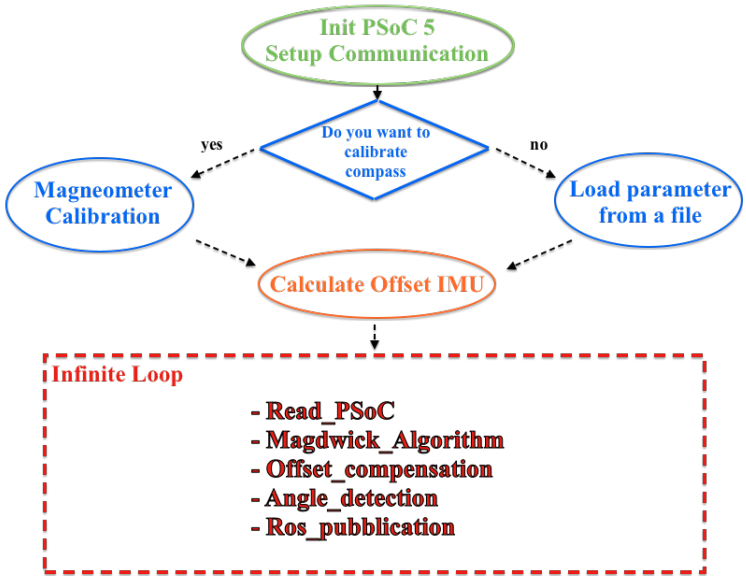

Figure 21: Code C++ implemented in the PC

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

Table 5: Probed Cups

(d) Validation In order to validate the hand posture, using the IMU glove, and to demonstrate the ability for a robot to recognize a grasped object from the knowledge of its hand posture some simple experiments were performed. To validate the hand posture, a Pisa/IIT SoftHand is dressed with the IMU glove. On line visualization of the hand posture is programmed wth ROS (Robot Operating System) [11]. Figs. 22, 23 and 24 show three different examples of hand posture reconstruction, while the two videos show the hand postrure reconstruction respectivily in the static and dynamic case.

To study the ability of a robot to recognize a grasped object knowing the posture of its hand, a set of five cups are used. Four of the five cups come from the PaCMan object database. All cups have different diameters as shown in table 5:

The object recognition experiments is composed by two different steps: *Learn* and *Recognize*. During the two phases the hand is used as a probe. The hand movements (open/close) are managed by a simple PID controller, so in this first phase the K_p, K_d and K_i parameters of the PID controller

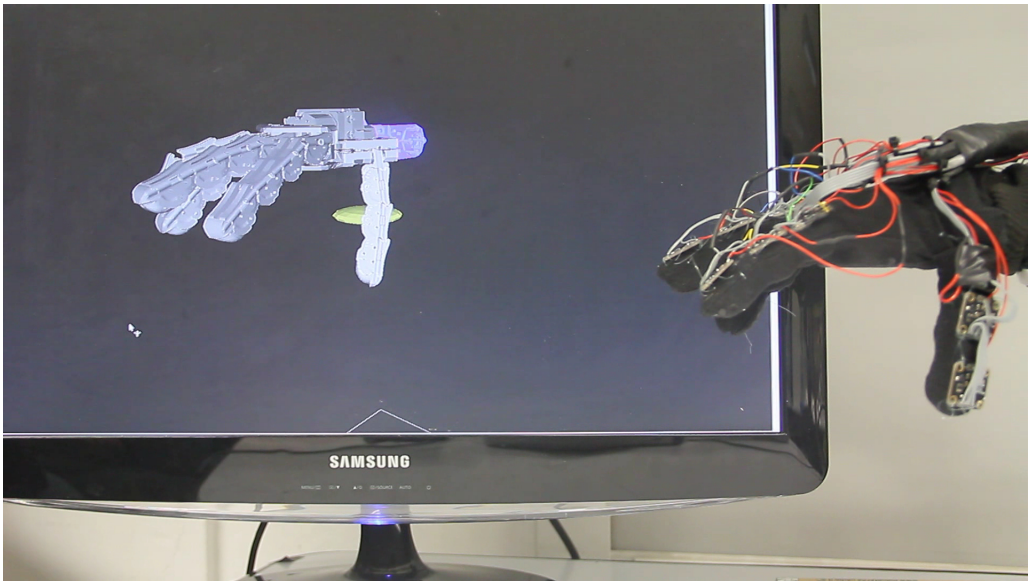

Figure 22: Hand Posture Reconstruction Example

are reduced in order to reduced the hand full torque. In these condictions the hand is not able to grasp an object and works as a probe robot.

In the learning phase when the hand probes an object, the software runnig on the PC appends in a matrix, saved in a text file, a twenty elements row. This row is composed by a first number which indentifies the object and by other nineteen elements which report the current 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 text file where the learning data are saved and creates a matrix $O_{bj} \in \mathbb{R}^{n\times 20}$, where n is the number of learned objects. Then the hand in closing probes the current object and software compares the current joint angles values with the values recorded during the learning phase.

To match the joint angles values an simple root mean squares was implemented, the software returns n different minimization numbers m_n and trivially the object in database nearest to the current one has the smallest m_n number. For each learned object m_n is given by

$$m_n = \sqrt{(C_{angle_1} - O_{n_{angle_1}})^2 + \cdots + (C_{angle_{19}} - O_{n_{angle_{19}}})^2}, \qquad (46)$$

where C_{angle_k} is the k^{th} hand current joint angles value, $O_{n_{angle_k}}$ is the k^{th} joint angles value of the object n and $k = 1, 2, \cdots, 19$.

With a large number of learned object (n) this object recognition strategy could require a large time to match current joints angles values with values recorded in the object database. In future works we study new strategies to better match the current joint angles values with the recorded ones

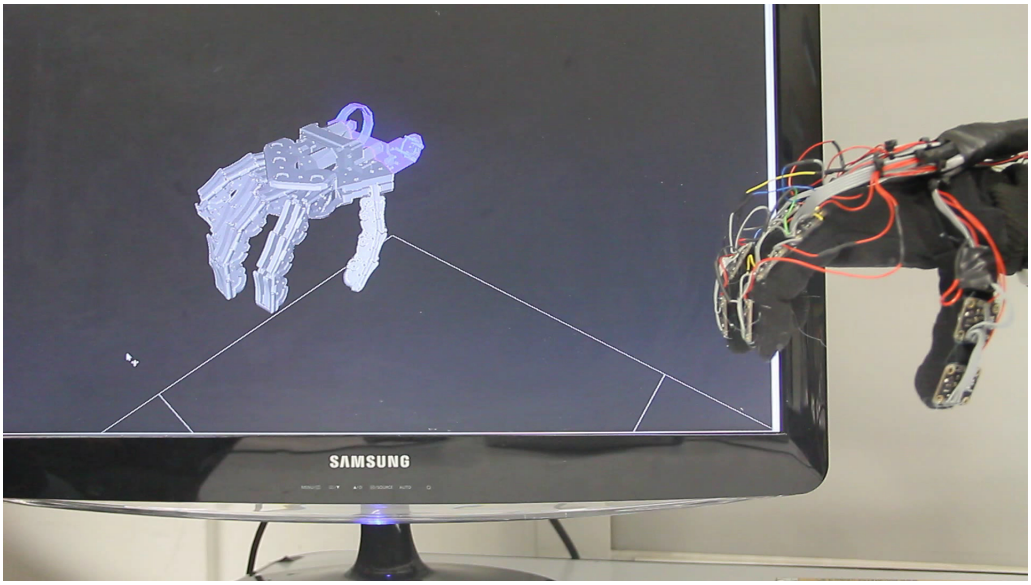

Figure 23: Hand Posture Reconstruction Example

and how to increase object database data, considering for example the grasp forces.

The Figs. 25, 26 and 27 show three different examples of objects recognition, while the videos show objects recognition procedure respectivily for the 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.

More details can be found in the attached paper [5] available at this link.

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

In this section, we describe our approach to reconstruct the posture of kinematic structures which do not easily lend themselves to the use of rotary encoders. This part is motivated by our need to measure the joint angles of the Pisa/IIT SofHand — its particular joint structure does not allow to fit encoders — after it has wrapped around an object (e.g., in an enveloping grasp), for the sake of employing it as a probe to explore and recognize objects.

The above discussed need, along with the mentioned constraints, brought us to consider the general problem of reconstructing the configurations of kinematic trees of rigid bodies without using measurements of relative angles, but employing absolute attitude 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

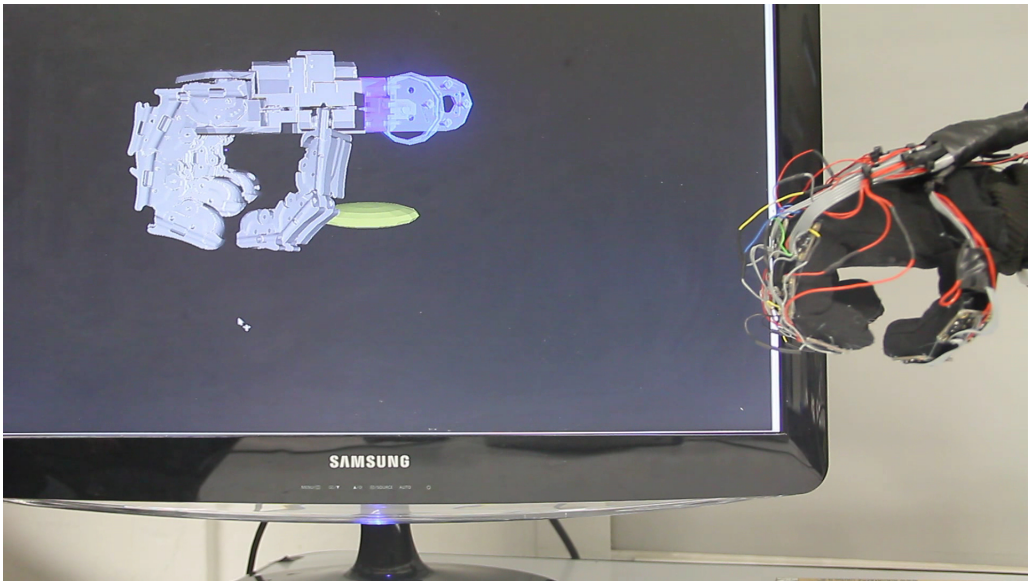

Figure 24: Hand Posture Reconstruction Example

as passive complementary filters exploiting the Mahony-Hamel formulation. The proposed method is applicable to general kinematic structures where measurements of relative angles is not feasible or convenient, or where, as in the case of the Pisa/IIT SoftHand, the joint kinematics are not lower pairs. In the accompanying paper [5] we present quantitative comparisons with ground truth data in grasping tests obtained for a two-fingered gripper: here, the fingers share the very same kinematic structure of the Pisa/IIT SoftHand. The comparisons validate the method employed, testifying that the resulting hardware design is mechanically robust, cheap and can be easily adapted to robotic hands with different topology, as well as to sensorizing gloves for studying human grasping strategies.

More details on this new approach and on the achieved results can be found in the attached paper [5] available at this link.

2 Annexes

Which papers / articles are included in the report? Mention titles, authors, publication info; abstract; and a one-liner relating the publication back to the discussion on actual work performed.

References

[1] J. Nunez-Varela and J. L. Wyatt, "Models of gaze control for manipulation tasks," *ACM Transactions on Applied Perception (TAP)*, vol. 10,

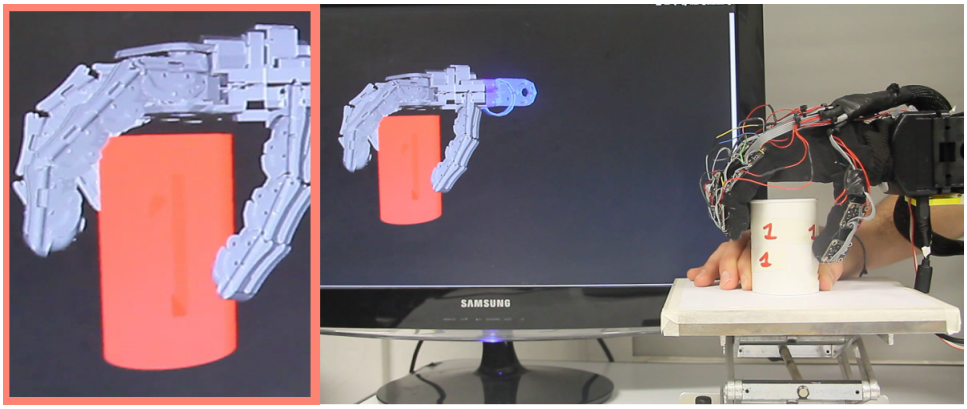

Figure 25: Object Recognition Example

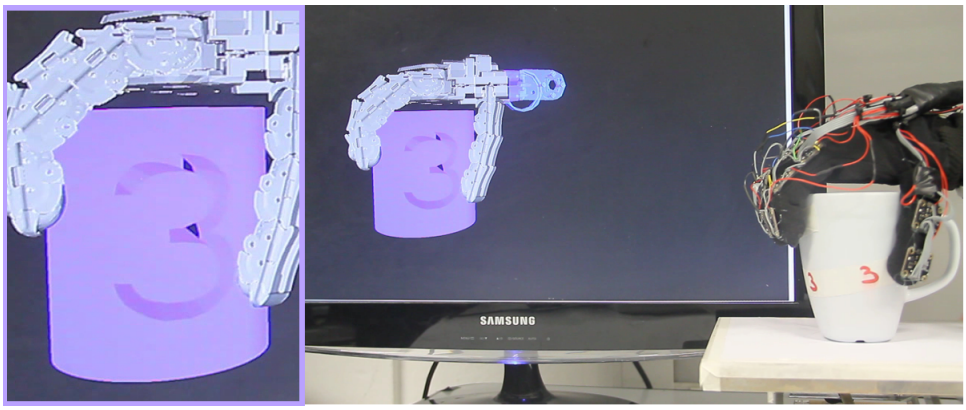

Figure 26: Object Recognition Example

no. 4, p. 20, 2013.

[2] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013, software available at http://octomap.github.com. [Online]. Available: http://octomap.github.com

[3] J. Bohg, M. Johnson-Roberson, B. León, J. Felip, X. Gratal, N. Bergström, D. Kragic, and A. Morales, "Mind the Gap - Robotic Grasping under Incomplete Observation," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, May 2011, to appear.

[4] C. Rosales, A. Ajoudani, M. Gabiccini, and A. Bicchi, "Active gathering of frictional properties from objects," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems*. Chicago, USA: IEEE, 2014, pp. 3982 – 3987.

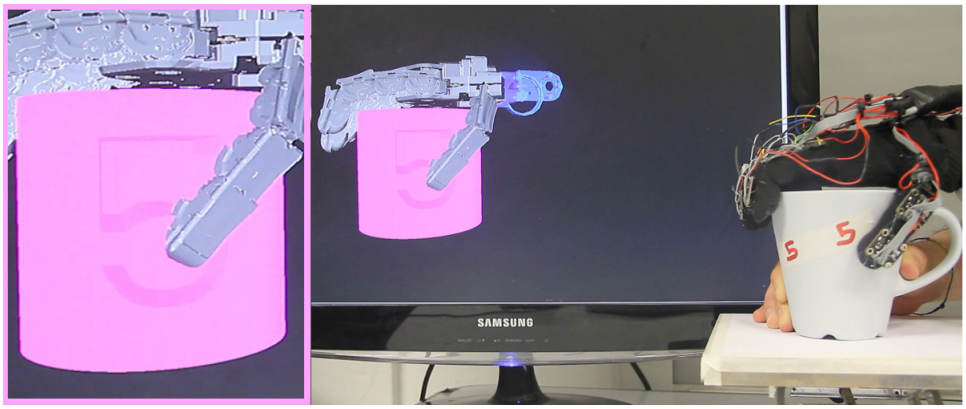

Figure 27: Object Recognition Example

[5] 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," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Seattle, USA, May 2015.

[6] 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.

[7] "Mpu9250 datasheet and register map page," online; accessed 23-February-2015. [Online]. Available: http://www.invensense.com/mems/gyro/mpu9250.html

[8] "Psoc 5lp development kit," online; accessed 23-February-2015. [Online]. Available: http://www.cypress.com/?rid=51577

[9] "Serial - usb converter datasheet," online; accessed 23-February-2015. [Online]. Available: http://www.droids.it/data_sheets/990.004%20datasheet.pdf

[10] "Ftdi chip home page," online; accessed 23-February-2015. [Online]. Available: http://www.ftdichip.com

[11] "Ros home page," online; accessed 23-February-2015. [Online]. Available: http://www.ros.org/