

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, J. L. Wyatt

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 exploratory probe. This report presents the efforts on providing controllers to perform sliding and rolling of a fingertip over a surface, integrating controller strategies 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.

1 Tasks, objectives, results	3
1.1 Planned work	3
1.2 Actual work performed	3
1.2.1 Task 3.1	3
1.2.2 Task 3.2	3
1.2.3 Task 3.3	5
A Annexes	10
A.1 Article: Active Gathering of Frictional Properties from Objects	10
A.2 Article: Low-cost, Fast and Accurate Reconstruction of Robotic and Human Postures via IMU Measurements	17
A.3 Technical report: Endowing the Pisa/IIT SoftHand with the sense of touch	26

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). In addition, the progress is summarised on Task 3.2 (M 18-36) regarding the vision of a promising approach to tackle the problem of information gathering for unknown objects. Developments of Task 3.3 (M 1-30) and its relation with WP4 are also reported.

Role of haptic object exploration in PaCMan

This deliverable reports the research done on finding a methodology to explore objects by touch and vision, and gather their haptic properties such as the static and dynamic friction coefficients. We can state that the role of haptic object exploration is to PaCMan what the friction coefficient is to grasp planning and execution: an essential element to be considered for success. The importance is equivalent but in a different arena, that is they refer to high and low level issues, respectively. In the project, several sources of uncertainty are to be considered for grasping, and those include object pose and detection, as well as the estimation of mechanical properties. A common representation is to be found within the project to aid the performance of robust grasps, and this deliverable provides means to gather the complementary tactile information to feed and be integrated with visual information to feed this object representation.

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 flexible testbed to contrast the representation coming from WP2 with approaches external to the project.

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.

1.2 Actual work performed

1.2.1 Task 3.1

We can proudly say that Task 3.1 has been accomplished. The results from this task crystallized in [1] which is appended in Sec. A.1. Details on how the planned activities were followed are described in MS 4.2.

In previous approaches, in general, the friction coefficient is neglected during acquisition when being in contact with the object. For instance, [2] fuses the two sources of information, visual and tactile, by applying symmetry constraints to reconstruct the complete object shape, and [3] uses the contact points to improve the object pose within the hand, initially obtained by vision. Nonetheless, other object properties apart from the shape are hardly explored, perhaps due to the fact that one needs specific sensory systems to measure them and a proper representation to process the information. In our work, the hardware setup resembles that of [4], but instead of using tactile arrays at the tip of the exploratory probe, we use an intrinsic tactile sensor based on [5] due to its capability to retrieve the contact point and force [6], which is crucial to our approach. Thus, in the paper we propose a way to estimate the friction coefficient from an object that has been previously detected from vision. The visual information is encoded using a Gaussian Process [7]. This representation is exploited to generate trajectories over the object surface, which then is explored by means of sliding controllers implemented within the project (see MS 4.2 and Sec A.1). The friction coefficient is also encoded using a Gaussian Process using both visual and tactile shape measurements. This provides a common probabilistic framework for both representations, however, we plan to replace such codification with the multi-modal representation reported on DR 2.1 during year 3.

1.2.2 Task 3.2

Previous work on object detection assisted by touch typically comprise fully actuated hands, such as the work presented in [8]. To this end, the initial estimate provided by the visual input must be accurate enough in order to

place precisely the finger areas covered with tactile arrays over the object surface, or use proper compliant controllers (see MS 4.2, Sec. A.2). This implies solving the object recognition problem as well. Taking the extreme case where no visual input is available, the remarkable work by [9] propose a Bayesian approach termed Scaling Series, however, the testbed for information gathering was a robot arm equipped with a single force/torque sensor which allow to compute with high accuracy the contact point w.r.t. to the robot base, similarly to the one reported in the previous task. The inconvenient of such setup fall in the probing time. As the authors of [9] say, with that setup (recall they don't use vision input at all in contrast to our work [1]), each probe took approximately 10s, and in the meantime, the algorithm, which is not light, was allowed to do computations. They reported that at least 3 probes were required in order to get reasonable low uncertainty in the estimate, which sum up to 30s to have a good estimate of the object pose and detection.

The fact that the object shape and localization uncertainty is reduced by making contact, combined with the adaptability of the Pisa/IIT SoftHand to any object shape, influenced the idea of using the hand as sophisticated exploratory probe w.r.t. previous testbeds. The configuration is similar to the usual adopted in bio-mechanic systems [10], where direct sensorization of the limbs is advised via optical or magnetic markers, or inertial devices. In addition, such sensorization should "strap down" to the greatest extent to the underlying bone, while defining the simplest assembly of technical joints that best reproduces the animal or human joint motion [11]. Thus, we propose a glove-inspired inertial-based solution to read the hand configuration for object detection as the result of a grasping action. We strongly believe that the proposed solution will be of great help in situations where partial or none visual information is available, as challenged in Task 3.2. For this reason, we dedicated efforts to continue on this research as it will be beneficial to accomplish the project goals with an alternative methodology to the one initially proposed. The following two paragraphs show the preliminary and on-going positive results on this topic, respectively.

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

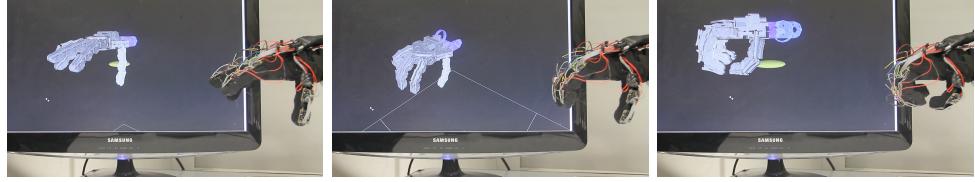


Figure 1: Hand posture reconstruction examples

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 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. We have found 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 initial approach and on the achieved results can be found in Sec. A.2.

Endowing the Pisa/IIT SoftHand with the sense of touch We successfully extended the proposed solution above to the case of the five-finger Pisa/IIT SoftHand. We validated the approach by visual inspection as shown in Fig. 1. Additional, we suggest watching the video sequences for static [static](#) and a [dynamic](#) configurations¹.

We used a naive recognizer to test the proposed solution, and we were able to discriminate similar cups in shape but in different sizes as shown in Fig. 2. 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](#). For more details on the implementation, refer to Sec. A.3.

1.2.3 Task 3.3

Concerning Task 3.3, we must say the the work on grasping of novel objects in WP4 progressed much faster than expected. The active gaze approach

¹External links are readily identified in the PDF and might not be visible in a black&white hard-copy, please, click on the PDF to go to the on-line site.

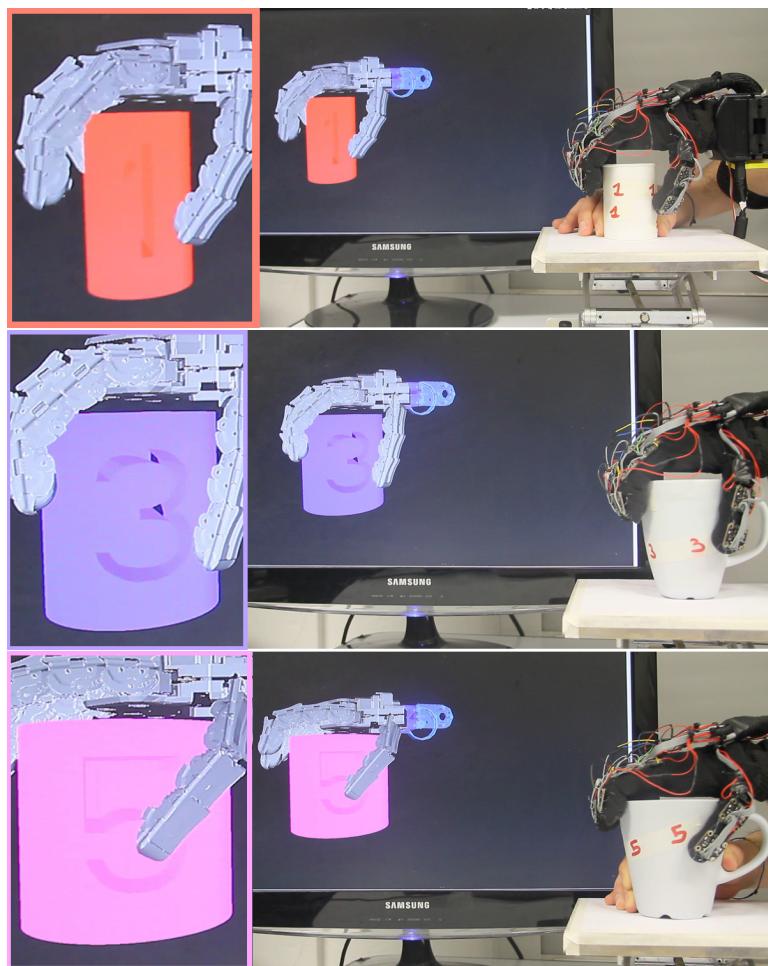


Figure 2: Object recognition examples using the same cup of different sizes.

taken has thus been informed by this progress. Specifically the fact that experiments in our IJRR submission for WP4 has 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 [12] 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 octomap [13], for online object modelling and sensor data fusion. The octomap 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 to estimate the safety of a grasp trajectory given this octomap. We are also testing different measures of the coverage of the object pertinent to a candidate grasp given the point cloud and the octomap representations.

Relevant work on grasping under incomplete information is that by Bohg and Kragic [14]. 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 [15], but in this the main goal of gaze control is to find and fixate on the object to support a visual servoing routine. There is relatively little published work that deals with active vision specifically in the service of manipulation. A more general but related problem is the use of active vision to recover the pose and complete shape of an object. Recently Krainin et al. [16] devised a next-best-view algorithm that autonomously acquires a complete shape model of an unknown object. The next-best-view is obtained by iterating over a grid of feasible viewpoints and selecting the (relative) camera pose with the highest score adjusted for the cost of rotating the object to the new viewpoint. The work also considers the volume occupied by the robotic manipulator (the configuration of which is known), to segment the object from the robot, and to plan new grasps in order to reveal object parts that were occluded by the manipulator. This is the closest to the approach we are currently working on. Finally 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.

References

- [1] 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*, 2014, pp. 3982 – 3987.
- [2] J. Ilonen, J. Bohg, and V. Kyrki, “Fusing visual and tactile sensing for 3-D object reconstruction while grasping,” in *Proc. of the IEEE Int. Conf. on Robotics & Automation*, 2013, pp. 3547–3554.
- [3] M. Chalon, J. Reinecke, and M. Pfanne, “Online in-hand object localization,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2013, pp. 2977–2984.
- [4] R. Bajcsy, “What can we learn from one finger experiments?” in *Int. Symp. on Robotics Research*, 1984.
- [5] A. Serio, M. Bianchi, M. Gabiccini, and A. Bicchi, “[d94] the tactile toolbox,” in *In Proc. of the 2014 IEEE Haptics Symposium*, 2014, p. 1.
- [6] A. Bicchi, J. K. Salisbury, and D. L. Brock, “Contact sensing from force measurements,” *The Int. J. of Robotics Research*, vol. 12, no. 3, pp. 249–262, 1993.
- [7] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
- [8] J. Bimbo, L. D. Seneviratne, K. Althoefer, and H. Liu, “Combining touch and vision for the estimation of an object’s pose during manipulation,” in *2013 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2013, pp. 4021–4026.
- [9] A. Petrovskaya and O. Khatib, “Global localization of objects via touch,” *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 569–585, 2011.
- [10] 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.
- [11] 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.
- [12] J. Nunez-Varela and J. L. Wyatt, “Models of gaze control for manipulation tasks,” *ACM Transactions on Applied Perception*, vol. 10, no. 4, p. 20, 2013.

- [13] 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>.
- [14] 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.
- [15] X. Gratal, J. Bohg, M. Björkman, and D. Kragic, “Scene representation and object grasping using active vision,” in *IROS’10 Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics*, 2010.
- [16] M. Krainin, B. Curless, and D. Fox, “Autonomous generation of complete 3D object models using next best view manipulation planning,” in *Proc. of the IEEE Int. Conf. on Robotics & Automation*, 2011.

A Annexes

A.1 Article: Active Gathering of Frictional Properties from Objects

Authors Carlos Rosales, Arash Ajoudani, Marco Gabiccini, and Antonio Bicchi

Info In proceedings of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, 2014.

Abstract This work proposes a representation that comprises both shape and friction, as well as the exploration strategy to gather them from an object. The representation is developed under a common probabilistic framework, particularly it uses a Gaussian Process to approximate the distribution of the friction coefficient over the surface, also represented as a Gaussian Process. The surface model is exploited to compute straight lines (geodesic flows) that guide the exploration. The exploration follows these flows by employing an impedance controller in pursuance of safety, shape accommodation and contact enforcement, while measuring the necessary data to estimate the friction coefficient. The exploratory probes consist of an RGBD camera and an Intrinsic Tactile sensor (ITs) mounted on a robotic arm. Experimental results give evidence for the effectiveness of the algorithm in the friction coefficient gathering and enrichment of the object representation.

Relation with the deliverable The article successfully integrate vision with the developed controller for the sliding primitive to refine the object shape and estimate the friction coefficient (static and dynamic), and concludes Task 3.1.

Attachment (following pages until next annex)

Active Gathering of Frictional Properties from Objects

Carlos Rosales¹, Arash Ajoudani², Marco Gabiccini³ and Antonio Bicchi⁴

Abstract—This work proposes a representation that comprises both shape and friction, as well as the exploration strategy to gather them from an object. The representation is developed under a common probabilistic framework, particularly it uses a Gaussian Process to approximate the distribution of the friction coefficient over the surface, also represented as a Gaussian Process. The surface model is exploited to compute straight lines (geodesic flows) that guide the exploration. The exploration follows these flows by employing an impedance controller in pursuance of safety, shape accommodation and contact enforcement, while measuring the necessary data to estimate the friction coefficient. The exploratory probes consist of an RGBD camera and an Intrinsic Tactile sensor (ITs) mounted on a robotic arm. Experimental results give evidence for the effectiveness of the algorithm in the friction coefficient gathering and enrichment of the object representation.

I. INTRODUCTION

Objects are perceived by robotic systems using non-contact or contact sensing, or a combination of both, for tasks such as recognition, and grasping and manipulation. Object properties which are of interest in robotics include shape, color, weight, texture, friction coefficients, inertial values, degrees of freedom, among many others. Shape and color are predominant due to the extensive use of cameras and the large computer vision field. However, other properties, such as the friction coefficients, are relevant to object grasping and manipulation and object recognition, by itself or in combination with visual information [1].

Visual and tactile array information has been successfully integrated into a single object shape model for different purposes. For instance, [2] uses the tactile information to discriminate object pose hypothesis coming from vision, [3] defines a hierarchical strategy to explore the object using the tactile information to refine the object shape only when necessary. In more recent works, [4] fuses the two sources of information by applying symmetry constraints to reconstruct the complete object shape, and [5] uses the contact points to improve the object pose within the hand, initially obtained

¹ C. Rosales is with Centro di Ricerca “E. Piaggio”, Università di Pisa, 1 Largo L. Lazzarino, 56122 Pisa, Italy. {carlos.rosales} at for.unipi.it

² A. Ajoudani is with the Department of Advanced Robotics (ADVR), Istituto Italiano di Tecnologia, Genova, Italy {arash.ajoudani} at iit.it

³ M. Gabiccini is with the Centro di Ricerca “E. Piaggio”, Università di Pisa, 1 Largo L. Lazzarino, 56122 Pisa, Department of Civil and Industrial Engineering (DICI), Largo Lucio Lazzarino 1, Pisa and Department of Advanced Robotics (ADVR), Italian Institute of Technology, 30 Via Morego, 16163 Genova, Italy. m.gabiccini at ing.unipi.it

⁴ A. Bicchi is with the Centro di Ricerca “E. Piaggio”, Università di Pisa, 1 Largo L. Lazzarino, 56122 Pisa and Department of Advanced Robotics (ADVR), Italian Institute of Technology, 30 Via Morego, 16163 Genova, Italy. bicchi at centropiaggio.unipi.it

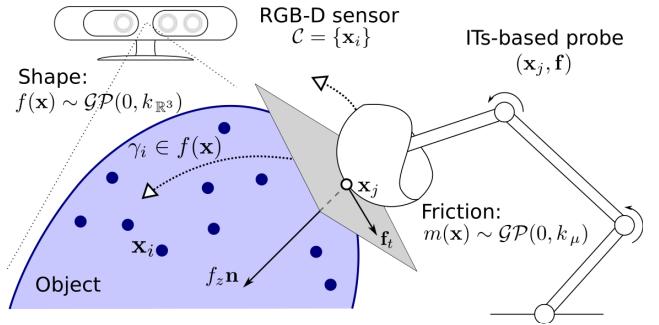


Fig. 1: The shape is acquired by an RGBD sensor and the friction coefficients by the intrinsic tactile sensor, both shape $f(\mathbf{x})$ and friction $m(\mathbf{x})$ functions are represented as a Gaussian process $\mathcal{GP}(0, k(\cdot))$. The exploration follows geodesic flows γ_i on the surface of a fixed object using a compliant behavior.

by vision. Nonetheless, other object properties apart from the shape are hardly explored, perhaps due to the fact that one needs specific sensory systems to measure them and a proper representation to process the information. In this preliminary work, the hardware setup resembles that of [6] (see Fig. 1), but instead of using tactile arrays at the tip of the exploratory probe, we use an ITs because of its capability to retrieve the contact force [7], which is necessary to our approach.

This work proposes a methodology to gather the kinetic coefficient of friction, a potential feature for object recognition [8], and certainly relevant to grasp determination [9], as well as its representation over the surface. The representation is developed under a probabilistic framework, particularly it uses Gaussian processes (GPs), which has proved to provide good results in object recognition [10] and grasp planning [11], [12]. The shape model allows to compute paths on the surface, here we use straight lines, hence the name geodesic flows, to guide the exploration. The actual exploration is executed by the ITs mounted on a 7 DoF arm. To this end, a controlled compliant behavior should provide a safe contact, shape accommodation and good data quality acquisition, assuming that the object is fixed to the table. The rest of the paper is organized as follows: Section II describes the devices and type of information they provide, Section III details the object representation and the exploration strategy, Section IV shows the experimental results, and, finally, Section V draws the conclusions and points for future developments.

II. SENSORY DATA

As shown in Fig. 1, the setup is composed of an RGBD sensor and an ITs mounted on a robot's end-effector, and a fixed object to be explored. Visual and tactile data acquired from them are described below.

A. Visual data

The object surface is acquired using a commercial RGBD sensor (Fig. 2a). The raw data comes in a point cloud format and it is pre-processed to isolate the point cloud of the object, $\mathcal{C} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, where $\mathbf{x}_i \in \mathbb{R}^3$ is a point on the surface. It is also possible to estimate the normal at the contact points which will be useful in the object representation learning (Subsection III-A). The extrinsic camera calibration is performed using objects as patterns and performing 3D pose estimation.

B. Tactile data

The ITs is composed of a force/torque (F/T) sensor, a tip and a compliant structure as a safety mechanism due to the high cost of the F/T sensor (Fig. 2b). The tactile data includes the contact point on the object surface, $\mathbf{x}_j \in \mathbb{R}^3$ w.r.t. the F/T sensor frame, and the contact force $\mathbf{f}_j \in \mathbb{R}^3$ expressed in the contact frame, such that we are able to estimate the kinetic coefficient of friction as

$$\mu = \frac{\|\mathbf{f}_t\|}{\|f_z\|}, \quad (1)$$

where f_z and \mathbf{f}_t are the components in the normal and in the tangent plane at the contact point, respectively [13]. There is no general correlation between the static and kinetic coefficients of friction [14], however, in general, tables are available for pairs of materials [13]. Given a set of materials to be encountered in the environment and the fingertip material, and assuming a fixed condition (humidity, temperature, etc.), one might query for the kinetic coefficient of friction and select the best matching pair and obtain the static coefficient of friction. In this work, we will present results with the the former only, since the procedure is equally applicable for both of them. The tip is 3D printed in ABS and with a hemi-spherical shape. The latter simplifies the contact point and force calculation, but other ellipsoidal surfaces can be used as well [7], or even arbitrary NURBS surfaces that require iterative optimization routines.

The ITs is mounted on a 7 DoF KUKA robotic arm to form the exploratory probe with high mobility. The compliant mechanism is an in-parallel passive compliant coupler (PPCC) with known Cartesian stiffness, and used to estimate the deformation using the F/T measurements. However, the stiffness is set with spring preloads to have negligible deformations on the range of forces applied during the exploration.

III. EXPLORATION STRATEGY

The proposed strategy is based on three main ingredients: the object representation for a complemented sensory data processing, exploration paths for an optimal information



(a) RGBD sensor. (b) Intrinsic Tactile sensor.

Fig. 2: Visual and tactile devices.

gain, and contour following for a successful data acquisition. Details on each of them are presented next.

A. Object Representation: Gaussian Process for implicit functions

The object shape is modeled by an implicit function, $f(\mathbf{x}) = 0$, with $\mathbf{x} \in \mathbb{R}^3$ being any point on the surface. This function is approximated using a Gaussian process (GP) [15] over the object point cloud, \mathcal{C} , as

$$f(\mathbf{x}) \sim \sum_{i=1}^N \alpha_i k_{\mathbb{R}_3}(\mathbf{x}, \mathbf{x}_i, \sigma), \quad (2)$$

where $k_{\mathbb{R}_3}(\cdot)$ is the covariance function centered at points $\mathbf{x}_i \in \mathcal{C}$, σ is a smoothing parameter (set according to the units of \mathbf{x} and enforced to have the same value for all terms), and α_i is the weighing factor for each term (to be learned from the measurements).

For 3D points, we use a triangular position kernel (see [10], [16] for more details), which has the form

$$k_{\mathbb{R}_3}(\mathbf{d}, \sigma) = \begin{cases} 1 - \frac{(\mathbf{d}^\top \mathbf{d})^{\frac{1}{2}}}{2\sigma} & \text{if } (\mathbf{d}^\top \mathbf{d})^{\frac{1}{2}} \leq 2\sigma \\ 0 & \text{if } (\mathbf{d}^\top \mathbf{d})^{\frac{1}{2}} > 2\sigma, \end{cases} \quad (3)$$

where we perform the change of variable $\mathbf{d} = \mathbf{x} - \mathbf{x}_i, \forall i$.

The friction coefficient value over the surface is also modeled by an implicit function, $m(\mathbf{d})$, and approximated as well by a GP, as

$$m(\mathbf{d}) \sim \sum_{i=1}^N \beta_i k_\mu(\mathbf{d}, \Sigma). \quad (4)$$

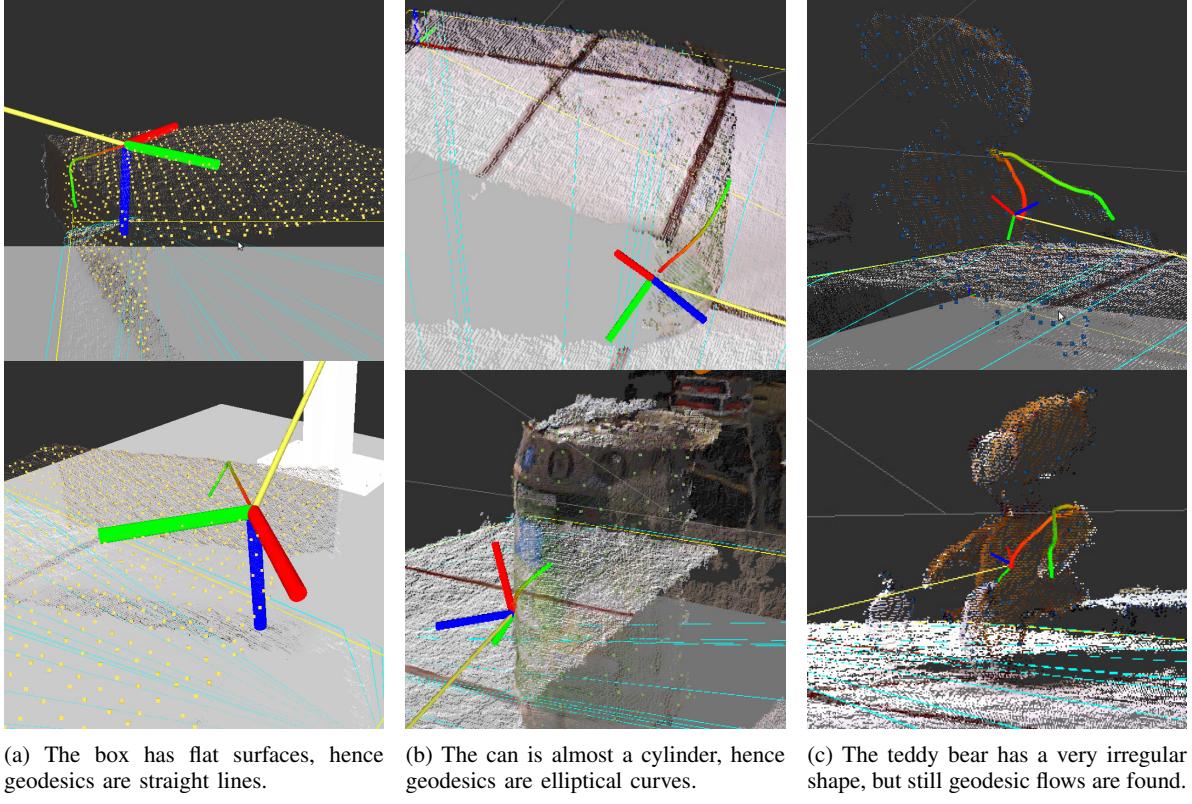
Due to the nature of the data, we decide to use a squared exponential as the covariance function instead, with the form

$$k_\mu(\mathbf{d}, \Sigma) = e^{-\frac{(\mathbf{d}^\top \mathbf{d})}{2\Sigma}}, \quad (5)$$

where Σ is a smoothing factor without units. It is worth noting that, the points on the surface in this case come from both the tactile and visual data (Section II).

The determination of the weights α_i and β_i depends on the function we want to represent to choose proper labels to the sensory data. In the case of the object surface, we would like to model

$$f(\mathbf{x}) \begin{cases} > 0, & \text{if } \mathbf{x} \text{ is outside the surface} \\ = 0, & \text{if } \mathbf{x} \text{ is on the surface} \\ < 0, & \text{if } \mathbf{x} \text{ is inside the surface,} \end{cases} \quad (6)$$



(a) The box has flat surfaces, hence geodesics are straight lines.

(b) The can is almost a cylinder, hence geodesics are elliptical curves.

(c) The teddy bear has a very irregular shape, but still geodesic flows are found.

Fig. 3: 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.

therefore, the labeling goes for a positive, null or zero value, e.g. 1, 0 and -1 , for points that are outside, on, or inside the surface, respectively. The set containing the points on the surfaces are directly in the visual data (Subsection II-A). The set containing points outside the surface is generated by estimating the outward normal over the previous set, and then, displacing the points along the normal. The set containing points inside could be created using a similar method, however, only one inside the surface proved to be enough. Thus, let the vector $\mathbf{y} = (y_1, \dots, y_i, \dots, y_s)^T$ contains the labels corresponding to the joined point set, and covariance matrix $K_{\mathbb{R}_3}$ with elements $k_{ij,\mathbb{R}_3} = k_{\mathbb{R}_3}(\mathbf{d}_{ij}, \sigma)$, with $\mathbf{d}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ for all points, the weights are approximated under simplifying assumptions as

$$(\alpha_1, \dots, \alpha_i, \dots, \alpha_N)^T = K_{\mathbb{R}_3}^{-1} \mathbf{y}. \quad (7)$$

In the case of the friction coefficient function, we would like to model

$$m(\mathbf{x}) \begin{cases} = \mu, & \text{if } \mathbf{x} \text{ is on the surface} \\ < 0, & \text{otherwise,} \end{cases} \quad (8)$$

therefore, the labels are the estimated friction coefficient and negative, e.g. -1 , for points from the tactile and visual data, respectively. Note that, a non-sense value for the friction coefficient is assigned for points that are not explored yet but are known to be on the surface. Like so, the weights are approximated again by arranging a proper label vector

$\mathbf{n} = (n_1, \dots, n_i, \dots, n_N)^T$ and covariance matrix K_μ with elements $k_{ij,\mu} = k_\mu(\mathbf{d}_{ij}, \Sigma)$ as

$$(\beta_1, \dots, \beta_i, \dots, \beta_N)^T = K_\mu^{-1} \mathbf{n}. \quad (9)$$

The contact points during the exploration could potentially be used to refine the object shape representation, but this is useful in the case that they provide radically new information, for instance, when exploring occluded parts of the object, which is clearly a point for future investigation.

B. Exploration paths: Geodesic flows on implicit surfaces

Paths for exploration should have a good ratio between information gain and exploration length (time or distance). A way to achieve this is to walk in straight lines over the surface to have minimal walking distance. Geodesics can be computed using the implicit function representing the surface and integrating from a point $(x, y, z)^{(0)}$ and a tangent vector $(p, q, r)^{(0)}$ to the surface, until the curve have a predefined length. Thus, given an implicit function $f(x, y, z)$, the geodesic curve equation can formulated in the following six first-order differential equations

$$\begin{cases} x' = p \\ y' = q \\ z' = r \\ e' = ((pf_z - rf_x)r + (pf_y - qf_x)q)L/D \\ f' = ((qf_z - rf_y)r + (qf_x - pf_y)p)L/D \\ g' = ((rf_y - qf_z)q + (rf_x - pf_z)p)L/D, \end{cases} \quad (10)$$

with

$$\begin{aligned} L &= f_{xx}p^2 + f_{yy}q^2 + f_{zz}r^2 + 2(f_{xy}pq + f_{yz}qr + f_{xz}pr) \\ D &= (rf_y - qf_z)(rf_y - qf_z) + (pf_z - rf_x)(pf_z - rf_x) + \\ &\quad + (qf_x - pf_y)(qf_x - pf_y). \end{aligned}$$

The first and second order derivatives of (2) required in (10) are

$$\frac{\delta f}{\delta \mathbf{x}} = \sum_{i=1}^N \alpha_i \frac{\delta k_{\mathbb{R}_3}(\mathbf{d}, \sigma)}{\delta \mathbf{d}} \frac{\delta \mathbf{d}}{\delta \mathbf{x}}, \quad (11)$$

and

$$\frac{\delta^2 f}{\delta \mathbf{x}^2} = \sum_{i=1}^N \alpha_i \frac{\delta^2 k_{\mathbb{R}_3}(\mathbf{d}, \sigma)}{\delta \mathbf{d}^2} \frac{\delta^2 \mathbf{d}}{\delta \mathbf{x}^2}, \quad (12)$$

where

$$\frac{\delta k_{\mathbb{R}_3}}{\delta \mathbf{d}} = \begin{cases} -\frac{1}{2\sigma} \frac{\mathbf{d}^\top}{(\mathbf{d}^\top \mathbf{d})^{\frac{1}{2}}} & \text{if } (\mathbf{d}^\top \mathbf{d})^{\frac{1}{2}} \leq 2\sigma \\ \mathbf{0} & \text{if } (\mathbf{d}^\top \mathbf{d})^{\frac{1}{2}} > 2\sigma, \end{cases} \quad (13)$$

and

$$\frac{\delta^2 k_{\mathbb{R}_3}}{\delta \mathbf{d}^2} = \begin{cases} -\frac{1}{2\sigma} \frac{\mathbf{I}_3 \mathbf{d}^\top \mathbf{d} - \mathbf{d} \mathbf{d}^\top}{(\mathbf{d}^\top \mathbf{d})^{\frac{3}{2}}} & \text{if } (\mathbf{d}^\top \mathbf{d})^{\frac{1}{2}} \leq 2\sigma \\ \mathbf{0}_3 & \text{if } (\mathbf{d}^\top \mathbf{d})^{\frac{1}{2}} > 2\sigma. \end{cases} \quad (14)$$

Thus, by integrating (10) using (2), (11), and (12), we obtain geodesic flows γ_i over the object surface, as shown in Fig. 3.

C. Contour following: Stiffness controller

The tactile data acquisition can only be successful when the contact with the object surface is guaranteed. To this end, force and position control is required during the exploration. Considering that there is no exact description of the object shape, executing a pure position control might be dangerous for the equipment. The presented ITs includes a compliant structure for safety, however, when the KUKA arm uses the stiffness controller, it behaves like a spring with defined stiffness and damping parameters. In such manner, we set the parameters to make the PPCC to seem infinitely stiff. The provided control law for the arm for this mode is

$$\boldsymbol{\tau} = \mathbf{J}^\top (K_c(X_d - X) + \mathbf{f}_d) + B(\mathbf{d}) + \boldsymbol{\tau}_{\text{dyn}}, \quad (15)$$

where $X_d \in SE(3)$ is the desired tool frame, $\mathbf{f}_d \in \mathbb{R}^6$ is the desired force at the tool, $K_c \in \mathbb{R}^6$ is the desired Cartesian stiffness of the tool frame specified along the three axes, and \mathbf{d} the desired damping behavior.

Concerning the contour following, the origin of the desired tool frame is taken directly from the integration of the geodesic curve. The orientation, instead, is corrected to keep the \mathbf{z}_r axis of the end-effector reference frame aligned with the normal at the contact point \mathbf{z}_c . This is due to the fact that the arm lacks the capability to render any desired Cartesian stiffness in this mode. Thus, since the tip of the ITs is spherical, the orientation correction is equivalent to have a fixed contact point w.r.t. to the sensor frame in a tactile-servoing fashion. The error rotation matrix can be readily obtained from angle-axis parameters, with angle

$$\theta = \cos^{-1}(\mathbf{z}_c \cdot \mathbf{z}_r),$$

Algorithm 1: Exploration logic.

```

Data: Scene point cloud  $\mathcal{S}$ , and parameters
Result: Object representation  $(f, m)$ 
 $\mathcal{C} = \text{SEGMENTOBJECT}(\mathcal{S});$ 
 $f = \text{COMPUTESHAPE}(\mathcal{C});$ 
while INFORMATIONGAIN do
     $\mathbf{x}_i = \text{SAMPLEPOINT}(f);$ 
     $\mathbf{t}_i = \text{SAMPLEDIRECTION}(f, \mathbf{x}_i);$ 
     $\gamma_i = \text{COMPUTEGEODESIC}(f, \mathbf{x}_i, \mathbf{t}_i);$ 
    while EXPLORING( $\gamma_i$ ) do
         $(\mathbf{x}_j^{(t)}, \mathbf{f}_j^{(t)}, \mu^{(t)}) = \text{RECORDTACTILEDATA}();$ 
         $m = \text{COMPUTEFRICTION}(\mathcal{C}, \mathbf{x}_j^{(\forall t)}, \mathbf{f}_j^{(\forall t)}, \mu^{(\forall t)});$ 
return  $(f(\mathbf{x}), m(\mathbf{x})) ;$ 

```

and axis

$$\mathbf{a} = \frac{\mathbf{z}_c \times \mathbf{z}_r}{\sin(\theta)}.$$

With this, and additionally setting a lower stiff value on the \mathbf{z}_r axis of the tool, as well as a non-zero desired contact force along it, a safe contact and shape accommodation is furnished during the haptic exploration.

D. Summary: Exploration logic

The three elements from the previous subsections are looped following the pseudo-code outlined in Algorithm 1, where the arguments of the implicit functions f, m are omitted for clarity. The SEGMENTOBJECT and RECORD-TACTILEDATA functions are related to the data acquisition process described in Section II. Functions COMPUTESHAPE and COMPUTEFRICTION are essentially the weight computations in (7) and (9). Function COMPUTEGEODESIC is the integration of (10). The EXPLORING(γ_i) loop is performed using the contour following procedure from Section III-C. A clear point for future discussion is how to systematically break the INFORMATIONGAIN loop.

IV. EXPERIMENTS

A. Implementation details

The object point cloud segmentation is done in four steps. First, we apply a pass-through filter over the complete point cloud to keep the table and object. Then, a dominant plane is estimated using RANSAC techniques on the filtered cloud. The table is removed, so only the object point cloud remains, which is downsampled to speed up computations. This process is aided by the PointCloud library [17] and the ROS system [18]. The numerical integration of the geodesic flows is performed using the boost C++ odeint library. The F/T sensor used in the ITs is the ATI Nano17 SI-25-0.25. A low-pass filter was applied to the F/T measurements to have a smoother contact estimation. The tip and the PPCC were designed in our lab. The arm where the ITs is mounted on is the 7 DoF KUKA LWR controlled through the Standford Fast Research Interface Library (FRIL). The data acquisition and synchronization interface between the KUKA impedance

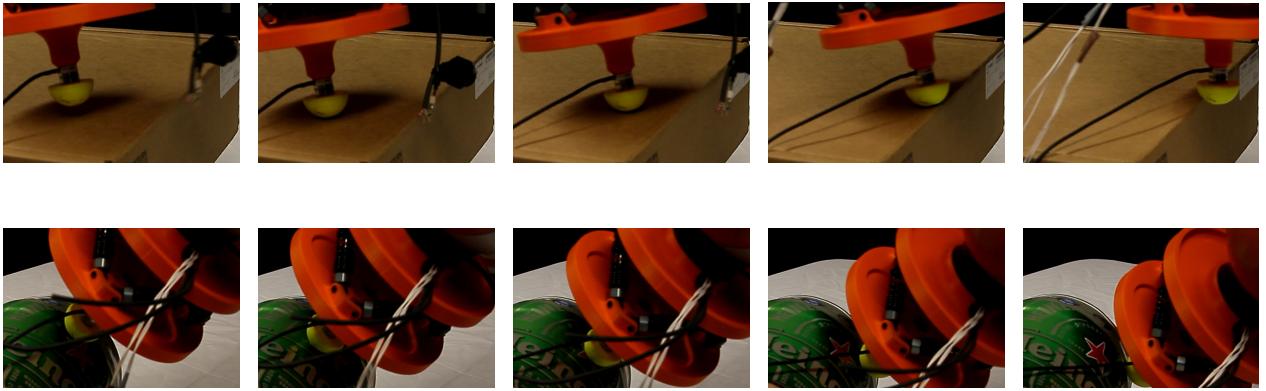


Fig. 4: Haptic exploration over objects. Top: a paperboard box. Bottom: a can.

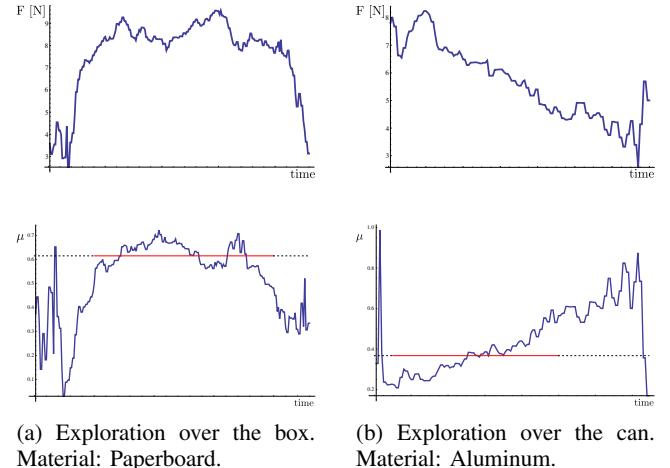
TABLE I: Parameters used for the experiments.

Parameter	Value
<i>RGBD sensor, object modeling, and global trajectories</i>	
Leaf size (downsampling)	0.02m
Min cluster size	500
Surface smoothing factor σ	100m
Friction smoothing factor Σ	5
Int. fixed-step Runge-Kutta	0.0001
Geodesic flow length	15cm
<i>KUKA lwr, ITs, and contour following</i>	
Robot cycle time	2ms
Cartesian stiffness ($\{\mathcal{E}\}$)	$(10, 10, 2, 5, 5, 5) \cdot 10^2$ [N/m, Nm/rad]
Desired force in z_E	5N
Orientation correction gain	0.001
ITs force threshold	0.1N
Tip radius	0.02m
F/T filt. low-pass cutoff freq.	5Hz

controller, the RGBD sensor and the 6-axes F/T sensor were developed in C++. The relevant parameters used for the experiments are shown in Table I.

B. Results and discussion

In Fig. 4, we show snapshots of the haptic exploration for two objects, a cardboard box and an aluminum can. The maximum distance between the planned and executed trajectories goes up to 2cm in the box, and 0.5cm in the can, caused, for instance, by the extrinsic camera calibration, object deformation, and F/T measurements errors. The force measured along z_c is used to determine the contact state of the trajectory to properly estimate the friction coefficient. Both are plotted in Fig. 5 for a single trajectory. We get a mean value of $\mu = 0.62$ and $\mu = 0.36$ for the box (ABS/Paperboard) and the can (ABS/Metal), respectively. In Fig. 6, the object shape with the friction coefficient estimation is shown after the haptic exploration along the geodesic flow. The points on the surface are obtained by sampling (2) and the color is obtained by evaluating (4), confirming that the methodology is capable of representing both visual and tactile data.



(a) Exploration over the box.
Material: Paperboard.

(b) Exploration over the can.
Material: Aluminum.

Fig. 5: Relevant data recorded during the haptic exploration. Top: Force along z_c used to detect the contact and non-contact states for. Bottom: Friction coefficient μ , the red line marks the mean value during the contact state.

V. CONCLUSIONS

This work proposed a novel object representation and the exploration strategy to instantiate it. The added value with respect to previous approaches is the consideration of the friction coefficient within the representation. This added information is essential to relevant robotic fields such as grasp planning and manipulation using mechanical hands, and potentially useful in object recognition, in a similar way as [19] recognizes textures. Whereas the visual data was acquired using a well-known RGBD sensor, the tactile data demanded a crude but effective Intrinsic Tactile sensor mounted on a robot's end-effector, acting as an haptic exploratory probe. The exploration is planned in minimal length paths over the object surface, i.e. geodesic flows, to looking to increase the information gain and exploration length ratio. The actual contour following used a stiffness controller for safety, and shape accommodation while ensuring a continuous contact during the exploration. The preliminary experimental results on a single-finger-like fashion revealed a valid approach to

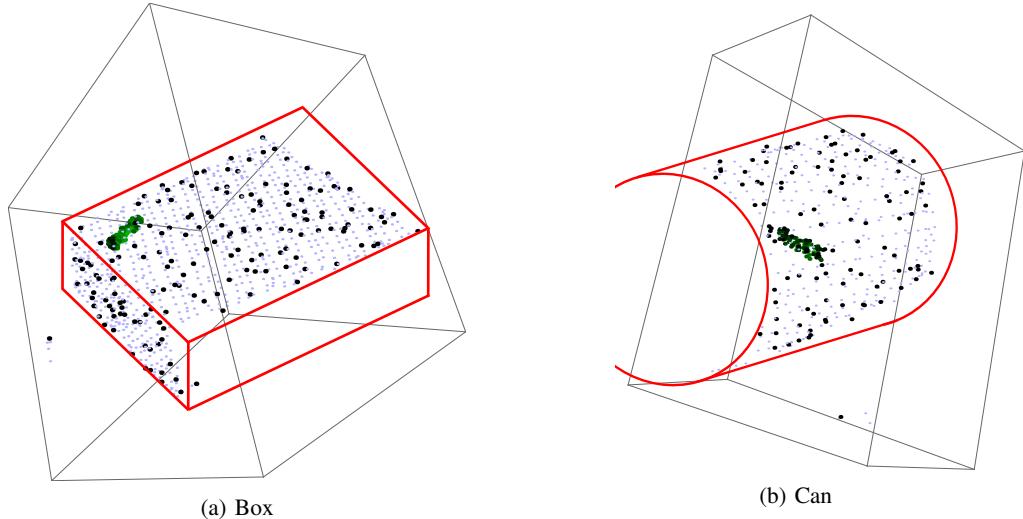


Fig. 6: Points on the object surface are colored with the estimated value of the friction coefficient, that is, the greener the higher, the darker the lower. Note that near the explored trajectory, the friction coefficient is non-zero even when the points are not exactly on the trajectory. For clarity, the actual surface points are smaller and shown in light blue, and an approximated wireframe of the object in red.

have both visual and tactile data in a single probabilistic framework using Gaussian processes.

Several points deserve further attention, such as how to generate points outside/inside the surface for a generic case, or the decision on when to end the exploration if no more information is gained, or how a different contour following controller can affect the quality of acquired data. Future work involves exploring the unseen part of the object, as well as the use of a dexterous hand equipped with ITs at the fingertips, to investigate the limitations due to a reduced mobility, and possibly, the benefits of a parallel tactile data acquisition system.

VI. ACKNOWLEDGEMENT

This work is supported by the grant no. 600918 “PAC-MAN” - Probabilistic and Compositional Representations of Object for Robotic Manipulation - within the FP7-ICT-2011-9 program “Cognitive Systems”. European Research Council and under the ERC Advanced Grant no. 291166 SoftHands (A Theory of Soft Synergies for a New Generation of Artificial Hands).

Authors would like to thank Manuel Bonilla for his support in the code implementation and the valuable comments of the reviewers.

REFERENCES

- [1] M. Cutkosky, R. Howe, and W. Provancher, “Force and tactile sensors,” in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer Berlin Heidelberg, 2008, pp. 455–476.
- [2] W. E. L. Grimson and T. Lozano-Perez, “Model-Based recognition and localization from sparse range or tactile data,” *The Int. J. of Rob. Res.*, vol. 3, no. 3, pp. 3–35, Sept. 1984.
- [3] P. Allen and R. Bajcsy, “Robotic object recognition using vision and touch,” in *Proc. of the 9th Int. Joint Conf. on Artif. Intel.*, 1987, pp. 1131–1137.
- [4] J. Ilonen, J. Bohg, and V. Kyrki, “Fusing visual and tactile sensing for 3-D object reconstruction while grasping,” in *Proc. of the IEEE Int. Conf. on Rob. and Aut.*, 2013, pp. 3547–3554.
- [5] M. Chalon, J. Reinecke, and M. Pfanne, “Online in-hand object localization,” in *Proc. of the IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.*, 2013, pp. 2977–2984.
- [6] R. Bajcsy, “What can we learn from one finger experiments?” in *International Symposium on Robotics Research*, 1984.
- [7] A. Bicchi, J. K. Salisbury, and D. L. Brock, “Contact sensing from force measurements,” *The Int. J. of Rob. Res.*, vol. 12, no. 3, pp. 249–262, June 1993.
- [8] X. Song, H. Liu, J. Bimbo, K. Althoefer, and L. D. Seneviratne, “Object surface classification based on friction properties for intelligent robotic hands,” in *World Automation Congress (WAC)*, 2012, 2012, pp. 1–5.
- [9] K. Hang, F. T. Pokorny, and D. Kragic, “Friction coefficients and grasp synthesis,” in *Proc. of the IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.* IEEE, Nov. 2013, pp. 3520–3526.
- [10] R. Detry and J. Piater, “Continuous Surface-Point distributions for 3D object pose estimation and recognition,” in *Computer Vision ACCV 2010*, ser. Lecture Notes in Computer Science, R. Kimmel, R. Klette, and A. Sugimoto, Eds., 2011, vol. 6494, ch. 45, pp. 572–585.
- [11] S. Dragiev, M. Toussaint, and M. Gienger, “Gaussian process implicit surfaces for shape estimation and grasping,” in *Proc. of the IEEE Int. Conf. on Rob. and Aut.*, May 2011, pp. 2845–2850.
- [12] S. El-Khoury, M. Li, and A. Billard, “On the generation of a variety of grasps,” *Rob. and Aut. Sys.*, vol. 61, no. 12, pp. 1335–1349, 2013.
- [13] CRC Handbook of Chemistry and Physics, 94th Edition, 94th ed. CRC Press, June 2013.
- [14] H. Yoshizawa, Y. L. Chen, and J. Israelachvili, “Fundamental mechanisms of interfacial friction. 1. relation between adhesion and friction,” *J. Phys. Chem.*, vol. 97, no. 16, pp. 4128–4140, Apr. 1993.
- [15] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
- [16] “NUKLEI: Kernel methods for SE(3) data (a c++ machine-learning library),” <http://nuklei.sourceforge.net/>.
- [17] R. B. Rusu and S. Cousins, “3D is here: Point cloud library (PCL),” in *Proc. of the IEEE Int. Conf. on Rob. and Aut.*, 2011, pp. 1–4.
- [18] “ROS: Robot operating system,” www.ros.org.
- [19] P. Dallaire, P. Giguere, D. Émond, and B. Chaib-draa, “Autonomous tactile perception: A combined improved sensing and bayesian nonparametric approach,” *Rob. and Aut. Sys.*, 2014.

A.2 Article: Low-cost, Fast and Accurate Reconstruction of Robotic and Human Postures via IMU Measurements

Authors Gaspare Santaera, Emanuele Luberto, Alessandro Serio, Marco Gabiccini, and Antonio Bicchi.

Info Accepted for publication in proceedings of the 2015 IEEE Intl. Conf. on Robotics and Automation

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.

Relation with the deliverable The proposed solution in this article is the preliminary result for the next annex.

Attachment (following pages until next annex)

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

Gaspare Santaera¹, Emanuele Luberto¹, Alessandro Serio¹, Marco Gabicci², and Antonio Bicchi³

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 $\mathbb{Q} = (\mathbb{S}^1 \times \dots \times \mathbb{S}^1)$, 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 $\mathbb{C} = (SE(3) \times \dots \times SE(3))$.

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 —

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. santaeragaspere at inwind.it, ema.lbt at hotmail.it, a.serio at centropiaggio.unipi.it

²M. Gabicci 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.gabicci at ing.unipi.it

³A. 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

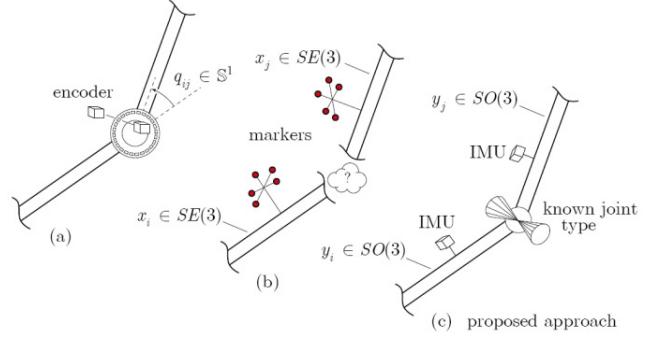


Fig. 1. Sensorization methods: (a) classical joint angle measurements, (b) direct measurements, (c) intermediate method (the one we propose).

optical or magnetic encoders, or potentiometers are employed in these cases.

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 $\mathbb{C} = (SE(3) \times \dots \times SE(3))$ 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 $\mathbb{C}_a = (SO(3) \times \dots \times SO(3))$. Then, we exploit a *mild* knowledge about the underlying kinematic topology — mostly the number of DoFs at each joint and the link

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 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

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

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

$$a = R^T g_0 + b_a + \mu_a. \quad (2)$$

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

$$v_a = \frac{a}{|a|}. \quad (3)$$

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

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

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

$$v_m = \frac{m}{|m|}. \quad (5)$$

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

$$\Omega_r = \Omega + b + \mu \quad (6)$$

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 instantaneous measurements.

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

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

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. $\tilde{R} \rightarrow I_3$.

Starting from the *correct* kinematics

$$\dot{\hat{R}} = R \Omega_x, \quad (8)$$

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

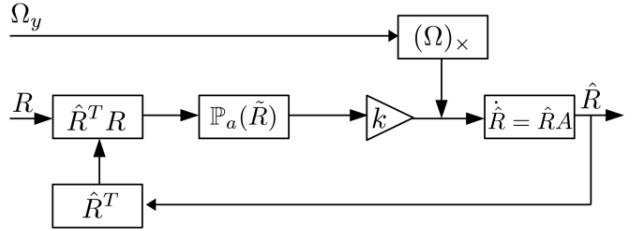


Fig. 2. Block diagram of the simplified form of the passive complementary filter

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

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

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

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

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

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

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

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

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

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 ($d_{acc_n} = R_{n-1}^T a_{b_n}$)
- 4: Determining the magnetic field vector in $\{B\}$ from the current attitude estimate and the normalized magnetic vector in the reference frame ($d_{mag_n} = R_{n-1}^T m_{b_n}$)
- 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 estimation 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 matrix $R_n \in SO(3)$ closest to R_{e_n} (in Frobenius norm) is computed minimizing the following cost function:

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

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

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

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 guarantees 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

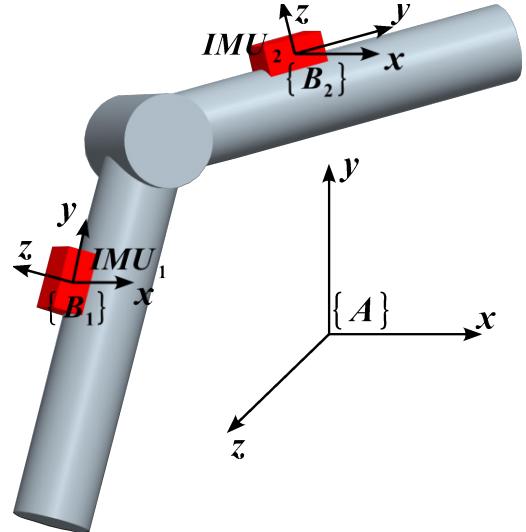


Fig. 3. Simple structure with two link connected by a revolute joint

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

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

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

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

Considering the orientation between the two IMUs, the rotation matrix R_n in (17) becomes R_{12n} . 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_{21n} given by

$$\Omega_{21n} = \Omega_{2n} - \Omega_{1n}. \quad (18)$$

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

$$e_{acc_n} = a_{b_{1n}} \times R_{12n}^T a_{b_{2n}} \quad (19)$$

$$e_{mag_n} = m_{b_{1n}} \times R_{12n}^T m_{b_{2n}}. \quad (20)$$

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 ($a'_{b_{1n}}, a'_{b_{2n}}$), magnetometers ($m'_{b_{1n}}, m'_{b_{2n}}$) and gyro rates ($\Omega_{b_{1n}}, \Omega_{b_{2n}}$) in the two IMUs frames ($\{B_1\}, \{B_2\}$).
 - 2: Normalizing gravity and magnetic field vector read from IMUs
 - $a_{b_{1n}} = \frac{a'_{b_{1n}}}{|a'_{b_{1n}}|}, a_{b_{2n}} = \frac{a'_{b_{2n}}}{|a'_{b_{2n}}|}, m_{b_{1n}} = \frac{m'_{b_{1n}}}{|m'_{b_{1n}}|}, m_{b_{2n}} = \frac{m'_{b_{2n}}}{|m'_{b_{2n}}|}$
 - 3: Determining the gravity vector in $\{B_2\}$ from the current attitude estimate and gravity vector read in the $\{B_1\}$ frame ($d_{acc_n} = R_{12_{n-1}}^T a_{b_{1n}}$)
 - 4: Determining the magnetic field vector in $\{B_2\}$ from the current attitude estimate and the magnetic vector read in the $\{B_1\}$ frame ($d_{mag_n} = R_{12_{n-1}}^T m_{b_{1n}}$)
 - 5: Calculating the gravity vector error $e_{acc_n} = a_{b_{2n}} \times d_{acc_n}$
 - 6: Calculating the magnetic vector error $e_{mag_n} = m_{b_{2n}} \times d_{mag_n}$
 - 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 $\Omega_n = \Omega_{b_{2n}} - \Omega_{b_{1n}}$
 - 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

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

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

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

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

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

and the joint angle value should be trivially given by

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

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 construction 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, performed 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, performed 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 .

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

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

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), on 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_4 and IMU_5 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 I²C (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 .

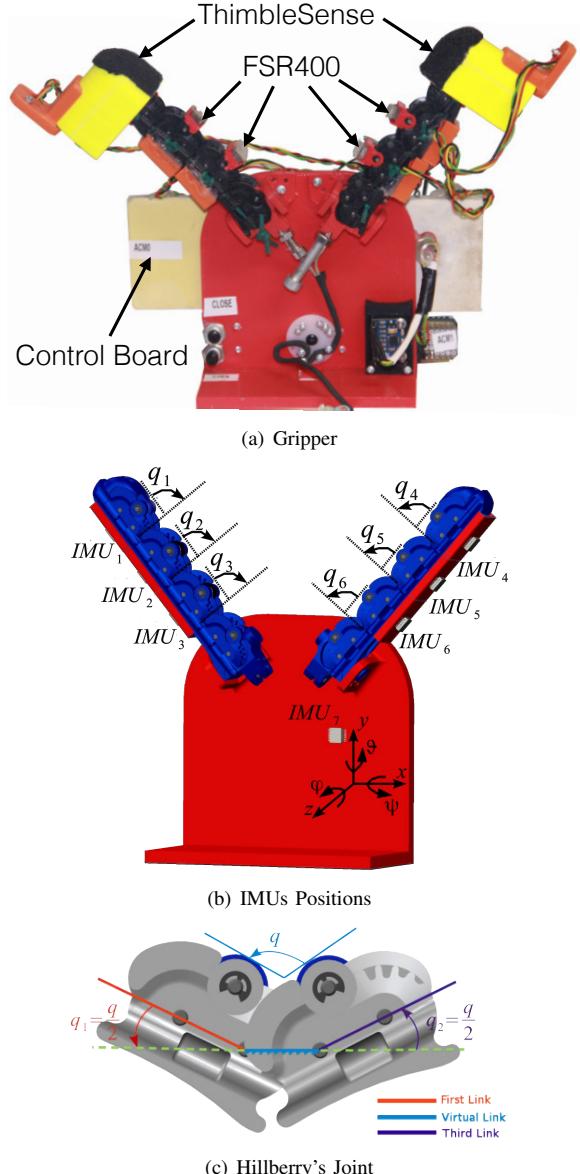


Fig. 4. Gripper setup and its main features for the proposed experiments. In (a) we report the gripper used to test the joint angle calculation procedure while in (b) the CAD representation 2-fingered gripper built out of the Pisa/IIT SoftHand fingers (Details on the: joint angle taxonomy, positioning of the IMUs on the phalanges 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

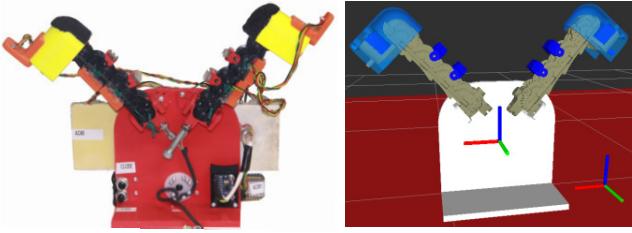


Fig. 5. On the left: gripper in its zero configuration; on the right: on-line reconstruction of the gripper pose in ROS.

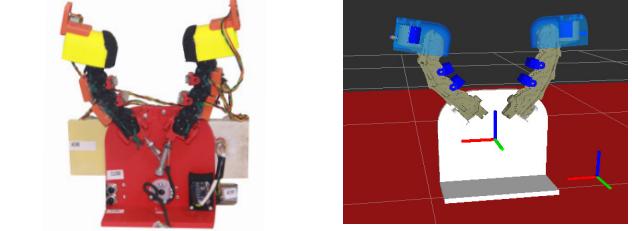


Fig. 6. On the left: gripper closing in free air; on the right: on-line reconstruction of the gripper pose in ROS.

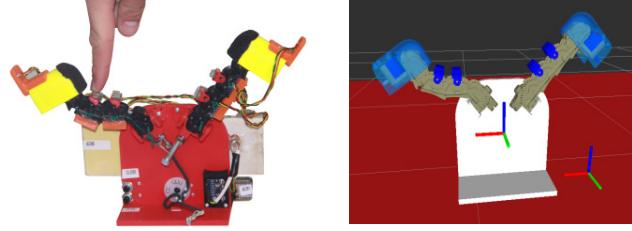


Fig. 7. On the left: gripper phalanx pushed by an external force; on the right: on-line reconstruction of the gripper pose and force applied in ROS (methodology for force measurement not described in this paper).

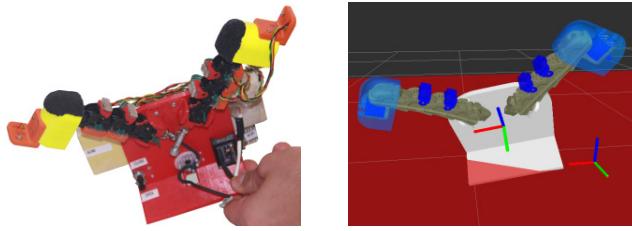


Fig. 8. On the left: gripper undergoing a roll and pitch angle rotations; on the right: on-line reconstruction of the gripper pose in ROS.

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.

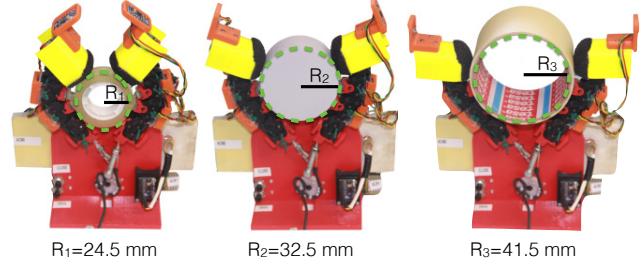


Fig. 9. Gripper during validation tests. From left to right: the gripper grasps three different circular objects with radii of 24.5 mm, 32.5 mm and 41.5 mm, respectively.

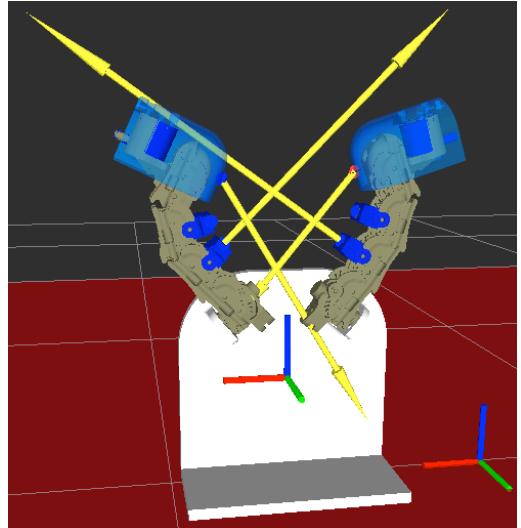


Fig. 10. Example of contact points and contact forces applied by the hand to the object during the grasp of circle with radius 41.5 mm.

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 "PAC-MAN" - 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/ii 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, "Development 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, realization 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 technology," 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. Magnus, 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.

A.3 Technical report: Endowing the Pisa/IIT SoftHand with the sense of touch

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

Info Internal report, in preparation for submission to a conference.

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. 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.

Relation with the deliverable The proposed solution in this report aims to tackle the problem of object exploration and recognition in the case where partial or none visual information is available. This scenario is challenged in Task 3.2.

Attachment (following pages)

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

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

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

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

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

where ${}^b\Omega_t$ is the angular rate measured at time t , Δt is the sampling period and ${}^b_a\hat{\bar{q}}_{t-1}$ 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 ${}^a\bar{d} = [0 \ d_x \ d_y \ d_z]$ and ${}^b\bar{s} = [0 \ s_x \ s_y \ s_z]$ 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

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

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 .

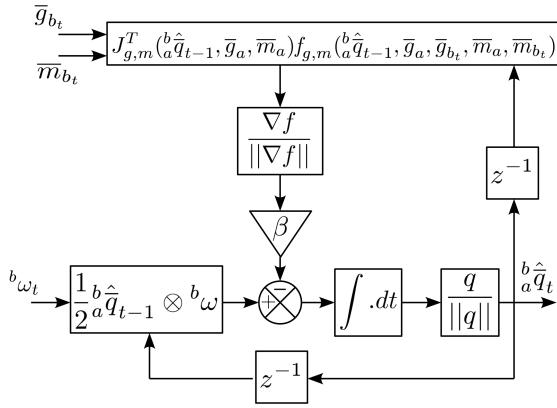


Fig. 1. Block diagram of the Madgwick passive complementary filter

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

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

where

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

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 $m_a = [0 \ m_{a_x} \ 0 \ m_{a_z}]$.

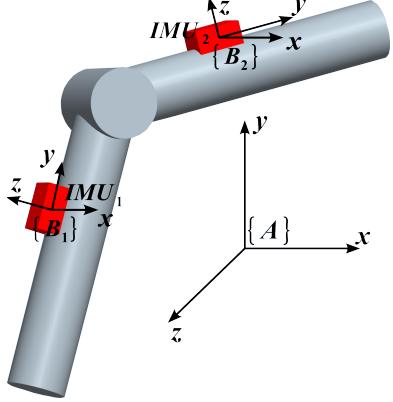


Fig. 2. Simple structure with two link connected by a revolute joint

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

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

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

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

where

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

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

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

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

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

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 the

Algorithm 1. 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 $\bar{g}_{b_1 n} = \frac{g_{b_1 n}}{\|g_{b_1 n}\|}$, $\bar{m}_{b_1 n} = \frac{m_{b_1 n}}{\|m_{b_1 n}\|}$
- 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 $\bar{g}_{b_2 n} = \frac{g_{b_2 n}}{\|g_{b_2 n}\|}$, $\bar{m}_{b_2 n} = \frac{m_{b_2 n}}{\|m_{b_2 n}\|}$
- 5: Computing the objective function value $f_{g,m}(^{b_1}\hat{q}_{n-1}, \bar{g}_{b_2 n}, \bar{g}_{b_1 n}, \bar{m}_{b_2 n}, \bar{m}_{b_1 n})$ using equations (10) and (12)
- 6: Computing the transpose of the Jacobian of the objective function $J_{g,m}^T(^{b_1}\hat{q}_{n-1}, \bar{g}_{b_2 n}, \bar{m}_{b_2 n})$ 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 $^{b_1}\hat{q}_{d_n} = (\frac{1}{2}^{b_1}\hat{q}_{n-1} \otimes {}^{b_1}\Omega_k) - c_n$ using (18)
- 9: Computing the new orientation quaternion estimation given by its previously one and its current time variation $^{b_1}\hat{q}_n = {}^{b_1}\hat{q}_{n-1} + {}^{b_1}\hat{q}_{d_n} \Delta t$
- 10: Normalizing the new estimated quaternion ${}^{b_1}\hat{q}_n = \frac{^{b_1}\hat{q}_n}{\|{}^{b_1}\hat{q}_n\|}$

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.

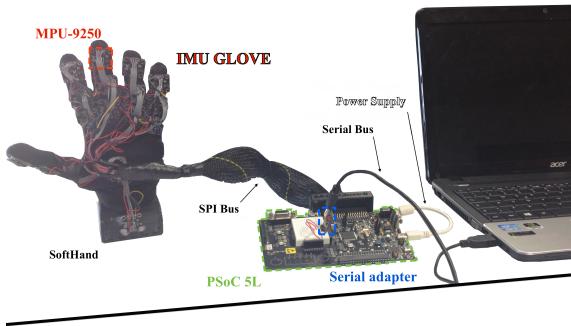


Fig. 3. Complete hardware setup of the IMU-based glove for the Pisa/IIT SoftHand

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

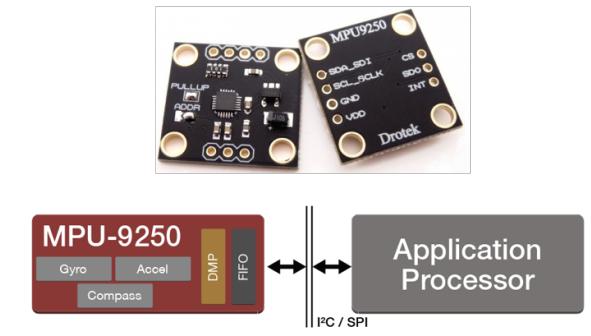


Fig. 4. MPU-9250

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

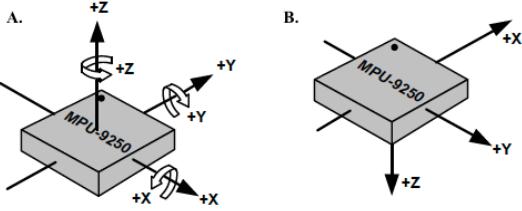


Fig. 5. A. Accelerometer and gyro orientation axes B. Magnetometer orientation axes

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].

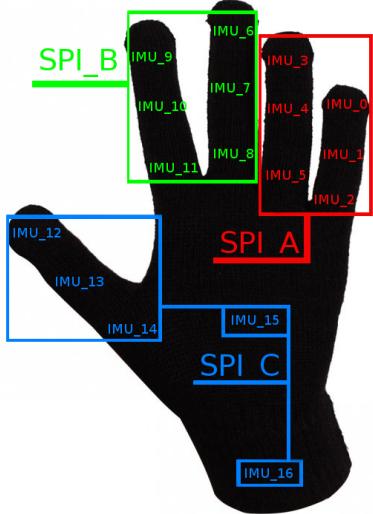


Fig. 6. SPI bus used in the IMU Glove

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 measurements.

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 initilization 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

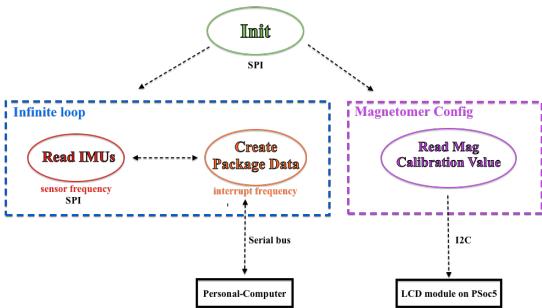


Fig. 7. Scheme of the firmware written in the PSoC5L

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

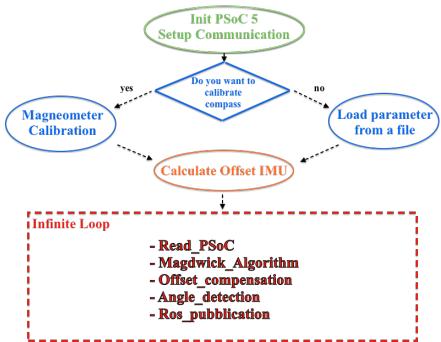


Fig. 9. Code C++ implemented in the PC

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 nothing 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 $O_{obj} \in \mathbb{R}^{n \times 20}$, 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

$$m_n = \sqrt{(C_{angle_1} - O_{nangle_1})^2 + \cdots + (C_{angle_{19}} - O_{nangle_{19}})^2}, \quad (23)$$

where C_{angle_k} is the k^{th} hand current joint angles value, O_{nangle_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.

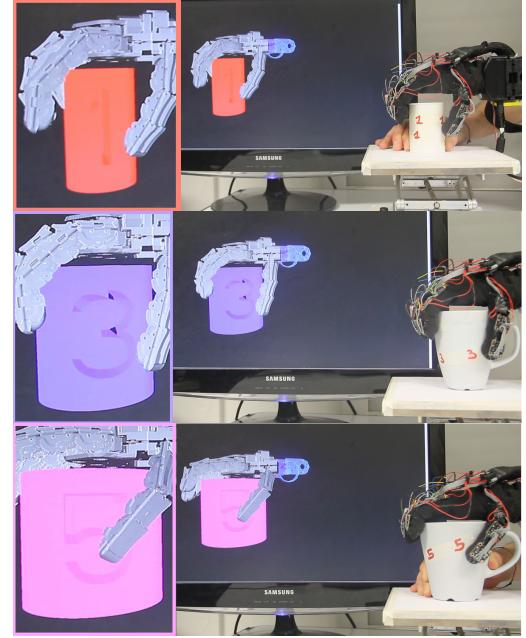


Fig. 11. Object recognition examples using the same cup of different sizes.

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

TABLE I
PROBED CUPS