Multimodal control for human-robot cooperation

Andrea Cherubini, Robin Passama, Arnaud Meline, André Crosnier and Philippe Fraisse

***Abstract*— For intuitive human-robot collaboration, the robot must quickly adapt to the human behavior. To this end, we propose a multimodal sensor-based control framework, enabling a robot to recognize human intention, and consequently adapt its control strategy. Our approach is marker-less, relies on a Kinect and on an on-board camera, and is based on a unified task formalism. Moreover, we validate it in a mock-up industrial scenario, where human and robot must collaborate to insert screws in a flank.**

Index Terms— **Human-Robot Interaction, Visual Servoing.**

I. Introduction

Flexible and reactive control of interaction between humans and robots will allow a closer cooperation in all service and industrial tasks, that require the adaptability skills of humans to be merged with the high performance of robots in terms of precision, speed and payload [1]. For this reason, recent research strives for intuitive human-robot cooperation, avoiding explicit clarification dialogue and commands. Humans are very good in mutual control of their interaction, by reading and interpreting the affective and social cues of each other. Hence, a robot that is able to read the user's (non-)verbal cues to infer the user's intention, will be able to interact more intuitively from the human perspective.

Pioneer work in the field of human-robot interaction (HRI) enabled an untrained user to intuitively interact with a lightweight robot just by touching its arm [2]. A prerequisite for this type of interaction is the capacity of the robot arm to sense the location and the strength of the human touch. In [3], a probability-based approach makes the robot adapt to the human behaviour and act proactively in ambiguous human intention scenarios. The robot can either wait for disambiguation of the intention, requiring extra human actions, or it can proactively act depending on his previous knowledge of the human behaviour. A finite state machine models the human intention. Similarly, in [4], proactive action selection is designed: the robot selects actions according to the human intention, without requiring an explicit user command.

All these scenarios require the robot to recognize the intentions of the human as early as possible, and to adapt to them in a reactive way. For this reason, we believe that sensor-based methods, such as visual servoing [5], provide better solutions, for intuitive HRI, than planning techniques requiring a priori models of the environment and agents [6]. To our knowledge, few works have merged these two robotic research fields, i.e., have explicitly dealt with HRI using sensor-based control approaches. One such work is [7], where force and vision based control are used to avoid collisions, while tracking human motion during interaction. Force sensing, along with minimum jerk based estimation of the human motion, is used by Maeda et al. [8] within a virtual compliance framework for cooperative manipulation. The authors of [9] present a system (including a wearable suit with 18 inertial motion capture sensors) for precise localization of a human operator in industrial environments. If the robot is realizing a task, and a human enters the safe area, the robot will pause until the human leaves.

Our objective is similar: we aim at enabling intuitive interaction between human and robot, in the context of an industrial scenario, where the two must collaborate to realize a task. The robot must be able to infer the human intentions during the task, using only sensed data. However, in contrast, with [9], we aim at doing this using low-cost sensors, and without structuring neither the environment nor the operator. Nowadays, this is possible thanks to the development of new low-cost depth sensors, such as the Microsoft Kinect TM [10], that ease human motion tracking and safe interaction [11]. The main contribution of this paper is the development of a multimodal sensor-based control scheme for intuitive human-robot collaboration. Apart from a Kinect, we rely on an on-board camera to enable human-robot collaboration in an industrial scenario. The approach is marker-less, and a unified task formalism is used for all the designed controllers.

The article is organized as follows. In Sect. II, all the relevant variables are defined. In Section III, we explain how our control framework is designed. Section IV is devoted to the specific explanation of the three control modes. Experimental results are reported in Section V, and summarized in the Conclusion.

II. General Definitions

The objective of this work is to enable a robot to aid a human operator in a screwing operation (see Fig. 1, top). Human and robot are operating on the opposite sides of a flank, where a series of screws must be inserted. The required operations are respectively:

- for the human: to insert the screws in the holes,
- for the robot: to tighten a bolt on each of the inserted screws, while the human maintains it on the flank.

To realize the proposed operation, we utilize two sensors: a Microsoft Kinect [10] that observes the work scene from a fixed pose, and a camera rigidly linked to the robot end effector. The Kinect outputs an RGB image, paired with a depth image (containing, for each pixel, the distance between the corresponding Cartesian point and the camera image plane). The camera, instead, outputs an RGB or gray image of the flank. These sensors are respectively dedicated to predicting the human intention, and to detecting new non-tightened screws. Specifically, the human intention is

The authours are with the Laboratory for Computer Science, Microelectronics and Robotics LIRMM - Université de Montpellier 2 CNRS, 161 Rue Ada, 34392 Montpellier, France. {firstname.lastname}@lirmm.fr

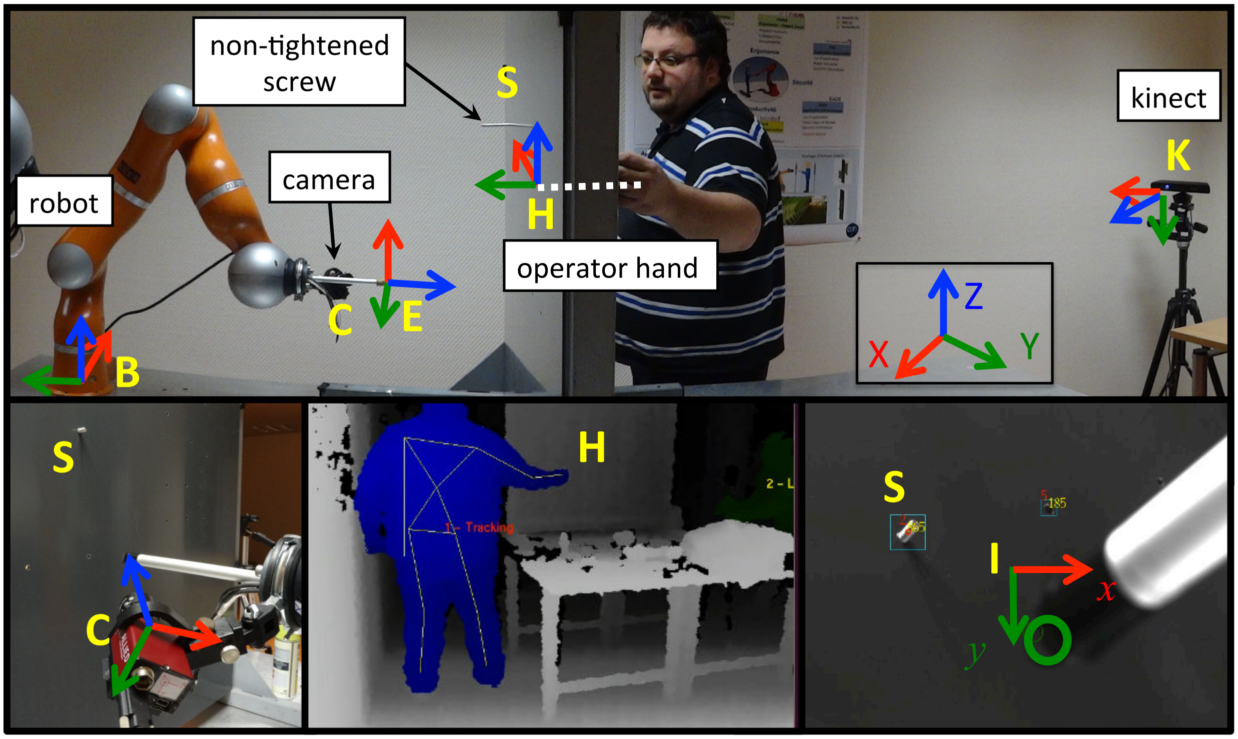

Fig. 1. Reference frames used in this work. Top: experimental setup. Bottom left: view of the camera and effector. Bottom center: Kinect image. Bottom right: camera image.

predicted by the Kinect through observation of the operating hand, which gives a rough estimate of the next screw position. Screw detection, instead, is realized by processing the images acquired by the on-board camera, as we will explain in Sect. IV.

The reference frames used in our work are (see Fig. 1): the robot base (B), camera (C), end effector (E), image (I), Kinect (K), and operating hand (H) frames. Reference frames B and K are fixed in the world, whereas C, E and I move with the robot. The origin of frame H is the orthogonal projection of the operator hand on the flank, while we fix its orientation, at all times, to that of B.

In this work, 3D points are represented in upper-case, and 2D image points in lower-case, using the homogeneous representation. For 3D points, coordinate frames are specified in superscript, such as ${}^{A}\mathbf{X}$, and the homogeneous transformation matrix ${}^{B}\mathbf{T}_{A}$ transforms points from frame A to B:

$$ {}^{B}\mathbf{X} = {}^{B}\mathbf{T}_{A}\, {}^{A}\mathbf{X}. \tag{1} $$

The transformation ${}^{B}\mathbf{T}_{A}$ is characterized by translation ${}^{B}\mathbf{X}_{A} = \left({}^{B}X_{A}, {}^{B}Y_{A}, {}^{B}Z_{A}\right)$, and by the angle/axis vector ${}^{B}\theta\mathbf{u}_{A}$ [12]. These constitute the pose of A in B: ${}^{B}\mathbf{P}_{A} = \left[{}^{B}\mathbf{X}_{A}, {}^{B}\theta\mathbf{u}_{A}\right]^{\top} \in \mathbb{SE}(3)$.

For the camera, we use the normalized perspective model. Since the origins of I and C are respectively the image center and the camera optical center, a 3D point with coordinates ${}^{C}\mathbf{X}$ in the camera frame, projects in the image as a 2D point with coordinates $\mathbf{x} = (x, y)$ such that:

$$ x = \frac{{}^{C}X}{{}^{C}Z}, \qquad y = \frac{{}^{C}Y}{{}^{C}Z}. \tag{2} $$

As an example, we show the non-tightened screw S both in the environment (Fig. 1, top and bottom left) and in the image (Fig. 1, bottom right).

To roughly determine the constant pose of the camera in the end effector frame, ${}^{E}\mathbf{T}_{C}$, we have utilized the Matlab camera calibration toolbox[1]. To avoid luminosity variations in the image, we maintain the camera orientation with respect to the flank constant throughout operation. Since ${}^{E}\mathbf{T}_{C}$ is constant, this is equivalent to keeping the effector tool perpendicular to the flank, with the E axes placed as in Fig. 1. Hence, throughout operation, we impose the desired rotation matrix from effector to base as:

$$ {}^{B}\mathbf{R}_{E}^{*} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}. \tag{3} $$

Our work assumptions are that the flank is perpendicular to the Y axis of the base frame, with distance ${}^{B}Y_{H}$ known, and that ${}^{B}\mathbf{T}_{K}$ is known through a coarse calibration between Kinect and robot base. In future work, we plan to relax these hypotheses.

III. Control framework

A. Control modes

To realize the proposed task of collaborative human-robot insertion and tightening of screws on a flank, we apply a multimodal strategy, where the four modes are each related to a subtask. Each subtask realizes a different phase of the operation, and the transitions are triggered by sensed data.

[1] www.vision.caltech.edu/bouguetj/calib_doc/

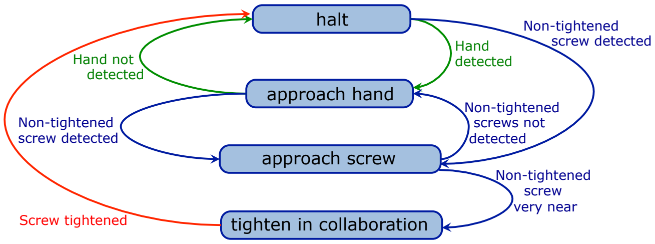
halt
Hand not detected
Hand detected
Non-tightened screw detected
approach hand
Non-tightened screw detected
Non-tightened screws not detected
approach screw
Non-tightened screw very near
Screw tightened
tighten in collaboration

Fig. 2. State Machine for selecting the appropriate control mode.

The four modes are:

- *Hand approaching*. If the human operating hand is detected by the Kinect, its position in the base frame is fed to a controller that moves the robot so that the camera has a good view of the area where the human is operating.
- *Screw approaching*. If a non-tightened inserted screw is detected in the image by the on-board camera (see Fig. 1, bottom right), its position is fed to a visual servo controller that moves the robot so that the end effector is placed in front of it.
- *Collaborative tightening*. Once the robot is sufficiently near, so that the screw is occluded by the tool in the image, the latter is placed on the screw to conclude the task. For the moment, this operation is executed in open-loop within our framework, since no sensed feedback is available at this stage[2]. This is a plausible assumption, since the occlusion occurs at a very short distance from the screw, and the subtask always proved successful in our experiments.
- *Halting*. If neither the hand nor the screws are detected by the Kinect and camera, or if the screw has been tightened, the robot is stopped until perceived data is again available.

To properly activate these modes, we utilize the simple state machine shown in Fig. 2. As the figure shows, the transitions between modes are activated by perceived information (either by the Kinect or by the camera) or by successful tightening. In the figure, we have depicted, respectively in green and blue, hand- and screw-related information.

B. General control law

Except for the halting mode, the three other modes are realized with a unified formalism, the task Jacobian controller [13]. We hereby recall the general formulation of this controller. Then, for each mode, we will detail the specific characteristics.

We name $\mathbf{s} \in \mathbb{R}^k$ the task vector, and $\dot{\mathbf{q}} \in \mathbb{R}^m$ the joint velocity, given as input to the robot controller. We assume that $m \geq k$, so that the task can be realized using the robot degrees of freedoms (dof). If $m > k$, redundancy exists, and one can minimize a cost function $H \in \mathbb{R}^{m-k}$, while concurrently realizing the task.

The task is related to the joint velocity by:

$$\dot{\mathbf{s}} = \mathbf{J}(\mathbf{q}, \mathbf{s})\,\dot{\mathbf{q}}, \tag{4}$$

where:

$$\mathbf{J}(\mathbf{q}, \mathbf{s}) = \frac{\partial \mathbf{s}}{\partial \mathbf{q}} \tag{5}$$

is the *task Jacobian*, of size $k \times m$, that depends on both the robot configuration and task. We assume it to have full rank during operation[3], and will denote it simply by $\mathbf{J}$ in the following.

The general solution of (4) is:

$$\begin{cases} \dot{\mathbf{q}} = -\lambda_1 \mathbf{J}^{-1}(\mathbf{s} - \mathbf{s}^*) & \textbf{when } m = k, \\ \dot{\mathbf{q}} = -\lambda_1 \mathbf{J}^{\dagger}(\mathbf{s} - \mathbf{s}^*) - \lambda_2 \left(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}\right) \bigtriangledown H & \textbf{otherwise}. \end{cases} \tag{6}$$

In the above equation:

- $\lambda_{1,2} > 0$ are two arbitrary positive scalar gains;
- $\mathbf{J}^{\dagger}$ is the $m \times k$ Moore-Penrose pseudoinverse of $\mathbf{J}$, i.e., a particular solution of: $\mathbf{J}\mathbf{J}^{\dagger}\mathbf{J} = \mathbf{J}$;
- $\mathbf{s} \in \mathbb{R}^k$ and $\mathbf{s}^* \in \mathbb{R}^k$ are respectively the current and desired task values, each of dimension k;

Control law (6) guarantees convergence of the task, since, replacing (6) in (4), yields linear differential equation:

$$\dot{\mathbf{s}} = -\lambda_1 (\mathbf{s} - \mathbf{s}^*), \tag{7}$$

for which, as desired, the pair $(\mathbf{s}^*, \dot{\mathbf{s}}^* = 0)$ is an exponentially stable equilibrium. Even in the presence of redundancy, minimization of H has no effect on the task, since $\bigtriangledown H$ is projected by $\mathbf{I} - \mathbf{J}^{+}\mathbf{J}$ onto the null space of $\mathbf{J}$.

In the following Section, we will detail, for each mode, the expressions of the current and desired tasks, $\mathbf{s}$ and $\mathbf{s}^*$, and of the Jacobian $\mathbf{J}$.

IV. Control Modes

As aforementioned, apart from the halting controller, which simply imposes:

$$\dot{\mathbf{q}} = 0, \tag{8}$$

for the three other modes $\dot{\mathbf{q}}$ is realized using (6). We will hereby detail each of these control modes.

A. Hand Approaching

The hand approach controller must use the operator intentions to drive the robot effector, so that the on-board camera has a good view of the zone of the flank where a new screw may be inserted.

The input to this controller is the Cartesian position of the hand in the base frame, ${}^{B}\mathbf{X}_H$. To derive this position, we rely on OpenNI (Open Natural Interaction[4]), a framework that tracks body motion in Kinect images. After a preliminary initialization step, OpenNI provides three-dimensional joint positions and limb orientations, by fitting a skeleton of the operator on the Kinect depth map. This data includes the operator hand Cartesian position in the Kinect frame, that we orthogonally project on the flank[5] to obtain ${}^{K}\mathbf{X}_H$, which can then be transformed in the robot base frame:

$${}^{B}\mathbf{X}_H = {}^{B}\mathbf{T}_K\,{}^{K}\mathbf{X}_H. \tag{9}$$

[2] In future work, we plan to integrate force control for this subtask.

[3] This will be discussed in Sect. V.

[4] http://www.openni.org

[5] It is trivial to derive the flank plane equation in K from ${}^{B}T_K$ and ${}^{B}Y_H$.

The OpenNI human tracker is sometimes erroneous. Hence, we validate the hand pose, to start the *Hand approaching* phase, only if it projects to a realistic position (defined by cartesian thresholds) in the robot base frame.

The hand pose in the robot frame will be used to define the desired control task $\mathbf{s}^*$. Specifically, the task is defined by the end effector pose in the base frame:

$$\mathbf{s} = {}^B\mathbf{P}_E \in \mathbb{SE}(3). \tag{10}$$

This can be estimated at each iteration, by applying the robot forward kinematics to the measured articular variables, $\mathbf{q}$.

Let us now explain the derivation of the desired task $\mathbf{s}^*$. For the translations, our aim is to place point H at a desired position

$${}^C\mathbf{X}^*_H = \left[{}^C X^*_H \; {}^C Y^*_H \; {}^C Z^*_H\right]^\top \tag{11}$$

in the camera frame, in order to visualize in the image the probable position of a future inserted screw. For rotations, we aim at servoing ${}^B\mathbf{R}^*_E$ according to (3). Then, since we set ${}^H\mathbf{R}_B = \mathbf{I}$, and since ${}^C\mathbf{R}_E$ is constant and known, the desired rotation from C to H can be calculated as:

$${}^H\mathbf{R}^*_C = {}^B\mathbf{R}^*_C = {}^B\mathbf{R}^*_E \, {}^E\mathbf{R}_C. \tag{12}$$

Combining (11) and (12), we can obtain the desired camera to hand transformation, ${}^H\mathbf{T}^*_C$. This can now be used to determine the desired effector to base transformation:

$${}^B\mathbf{T}^*_E = {}^B\mathbf{T}_H \, {}^H\mathbf{T}^*_C \, {}^C\mathbf{T}_E, \tag{13}$$

where ${}^B\mathbf{T}_H$ is estimated by the Kinect, and ${}^C\mathbf{T}_E$ is known from the camera calibration. This equation provides the value:

$$\mathbf{s}^* = {}^B\mathbf{P}^*_E \in \mathbb{SE}(3), \tag{14}$$

to be used in controller (6).

Finally, for this task, the Jacobian in (6) is simply:

$$\mathbf{J} = \frac{\partial {}^B\mathbf{P}_E}{\partial \mathbf{q}}. \tag{15}$$

To compute this Jacobian at run time, we apply the technique presented in [13].

B. Screw Approaching

The objective of the screw approaching controller is to drive the robot effector on the detected non-tightened screw, so that the bolt can be placed. To this end, we exploit the screw position as viewed from the on-board camera, along with the measures of the robot articular positions for forward kinematics.

Let us hereby detail the image processing algorithms used to detect and track the screws. Although we used the ViSP library [14] for various visualization utilities, the image processing algorithms for detecting and tracking the screws were developed from scratch. The detection of the non-tightened screw is decomposed in three steps. First, flank holes are detected, using a Sobel filter, followed by erosion, to suppress noise and detect coarse blobs. Then, the centroid of these blobs is projected, using the robot proprioception (i.e., the articular values $\mathbf{q}$), from image frame ${}^I\mathbf{X}$ to robot base frame ${}^B\mathbf{X}$. These projections in the base frame are used to build a history of all detected holes, necessary for matching holes from one image to the next. Finally, to detect if a screw has been inserted in a hole, we threshold the normalized correlation of the hole over two consecutive images. In fact, low correlation implies that the image of the hole has changed over the two images, and this occurs whenever the screw is inserted.

The current and desired tasks, and the Jacobian are defined using the two and one-half-dimensional (2 1/2 D) visual servo paradigm originally introduced in [15]. This method combines the advantages of image-based and position-based visual servoing schemes, while trying to avoid their shortcomings [5].

In fact, the task is defined by a combination of image features and 3D characteristics:

$$\mathbf{s} = \left[\begin{array}{cccc} x_S & y_S & \log {}^C Z_S & {}^{C^*}\theta\mathbf{u}_C \end{array}\right]^\top \in \mathbb{SE}(3). \tag{16}$$

In this equation, x_S and y_S are the image coordinates of the screw, ${}^C Z_S$ is the depth of the screw, and ${}^{C^*}\theta\mathbf{u}_C$ is the relative rotation between the camera current and desired poses.

The desired task

$$\mathbf{s}^* = \left[\begin{array}{cccc} x^*_S & y^*_S & \log {}^C Z^*_S & \mathbf{0} \end{array}\right]^\top \tag{17}$$

corresponds to driving the screw to image position (x^*_S, y^*_S) at the desired depth in the camera frame ${}^C Z^*_S$, while zeroing the orientation error between C and C^*. These values must be chosen so that end effector and screw are as near as possible, while avoiding that the effector occludes the screw in the image. In the bottom right of Fig. 1, the green circle indicates the image position (x^*_S, y^*_S) that we used in the experiments.

We set the desired effector Cartesian position to have a desired translation with respect to the screw:

$${}^E\mathbf{X}^*_S = \left[\begin{array}{ccc} 0 & 0 & {}^E Z^*_S \end{array}\right]^\top, \tag{18}$$

so that ${}^E Z^*_S > 0$ is as small as possible, without end effector occlusion. Then, from the known ${}^C\mathbf{T}_E$, we can derive:

$${}^C\mathbf{X}^*_S = {}^C\mathbf{T}_E \, {}^E\mathbf{X}^*_S, \tag{19}$$

hence ${}^C Z^*_S$, $x_S = {}^C X^*_S / {}^C Z^*_S$, and $y_S = {}^C Y^*_S / {}^C Z^*_S$. For rotations, as usual we aim at servoing ${}^B\mathbf{R}^*_E$ according to (3). Then, since ${}^C\mathbf{R}_E$ is known, ${}^{C^*}\theta\mathbf{u}_C$ can be calculated from:

$${}^{C^*}\mathbf{R}_C = {}^C\mathbf{R}_E \, {}^E\mathbf{R}^*_B. \tag{20}$$

The Jacobian corresponding to the 2 1/2 D task is [15]:

$$\mathbf{J} = \mathbf{L}_s \, {}^C\mathbf{V}_B \frac{\partial {}^B\mathbf{P}_C}{\partial \mathbf{q}}. \tag{21}$$

In this expression, $\mathbf{L}_s$ is the interaction matrix relating the task evolution to the camera velocity in frame C:

$$\mathbf{L}_s = \left[\begin{array}{cc} \mathbf{L}_{11}\left(x, y, {}^C Z_S\right) & \mathbf{L}_{12}\left(x, y\right) \\ 0 & \mathbf{L}_{22}\left({}^{C^*}\theta\mathbf{u}_C\right) \end{array}\right], \tag{22}$$

while ${}^C\mathbf{V}_B$ is the spatial motion transform matrix from frame B to frame C:

$${}^C\mathbf{V}_B = \left[\begin{array}{cc} {}^C\mathbf{R}_B & \left[{}^C\mathbf{t}_B\right]_\times {}^C\mathbf{R}_B \\ 0 & {}^C\mathbf{R}_B \end{array}\right]. \tag{23}$$

The complete expressions of $\mathbf{L}_{11}$, $\mathbf{L}_{12}$, and $\mathbf{L}_{22}$ are given in [5], and $[t]_{\times}$ is the skew symmetric matrix associated with vector t. Jacobian $\mathbf{J}$ can be calculated at each iteration, since: $\mathbf{L}_s$ depends on s, ${}^{C}\mathbf{V}_B$ on the pose of B in C (determined via forward kinematics ${}^{B}\mathbf{P}_E$ plus constant known ${}^{E}\mathbf{T}_C$), and $\partial^{B}\mathbf{P}_C/\partial\mathbf{q}$ can be calculated again using the technique presented in [13].

C. Collaborative tightening

For collaborative tightening, the effector must be displaced from the pose reached at the end of the screw approaching mode E^{sa} (when the screw is about to be occluded by the effector), to the desired pose E^{*}, that will place the effector on the screw for tightening. Since the 2 1/2 D visual servoing control used for screw approaching is highly precise even in the presence of camera calibration errors, this relative pose, that we denote with ${}^{E,sa}\mathbf{T}^{*}_{E}$, has proved constant throughout the experiments. In particular, since we always keep the effector orthogonal to the flank, it consists in a simple translation.

As for hand approaching, the task is defined here, by the end effector pose in the base frame:

$$\mathbf{s} = {}^{B}\mathbf{P}_E \in \mathbb{SE}(3). \tag{24}$$

The desired task:

$$\mathbf{s}^{*} = {}^{B}\mathbf{P}^{*}_E \in \mathbb{SE}(3), \tag{25}$$

will be derived from the homogeneous transformation matrix:

$${}^{B}\mathbf{T}^{*}_{E} = {}^{E}\mathbf{T}_{E,sa} \;\; {}^{E,sa}\mathbf{T}^{*}_{E} \tag{26}$$

Obviously, the Jacobian in (6) is also the same as for hand approaching:

$$\mathbf{J} = \frac{\partial^{B}\mathbf{P}_E}{\partial\mathbf{q}}. \tag{27}$$

V. EXPERIMENTS

For the experiments, we use a lightweight KUKA LWR IV robot [16] in the setup illustrated in Fig. 1. Since this robot has $m = 7$ degrees of freedom, and the tasks of all modes have dimension $k = 6$, the robot is redundant, and we use the remaining degree of freedom to guarantee joint limit avoidance. For this purpose, in (6), we use a scalar, configuration dependent, cost function [17]:

$$H(\mathbf{q}) = \frac{1}{2}\sum_{i=1}^{7}\left(\frac{q_i - q_{i,mid}}{q_{i,max} - q_{i,min}}\right)^2, \tag{28}$$

with $[q_{i,min}, q_{i,max}]$ the available range for joint i and $q_{i,mid}$ its midpoint.

The mounted camera is a black and white Stingray F201B from Allied Vision Technologies, with resolution 1024×768 pixels. Image processing for screw detection takes approximately 150 ms, so, although the skeleton processing on the Kinect is slightly faster, we fix the control loop rate at 8 Hz.

Since a tightening tool is not currently mounted on our end effector, we have validated our approach by verifying if the robot could successfully touch new, non-tightened screws with a cylindrical tool mounted on the end effector. This task requires high accuracy, since the screw and tool diameters are respectively 4 and 14 mm. We have run a series of experiments, where screws have been successfully touched by the effector, so we are confident that with a tightening tool the approach will also work.

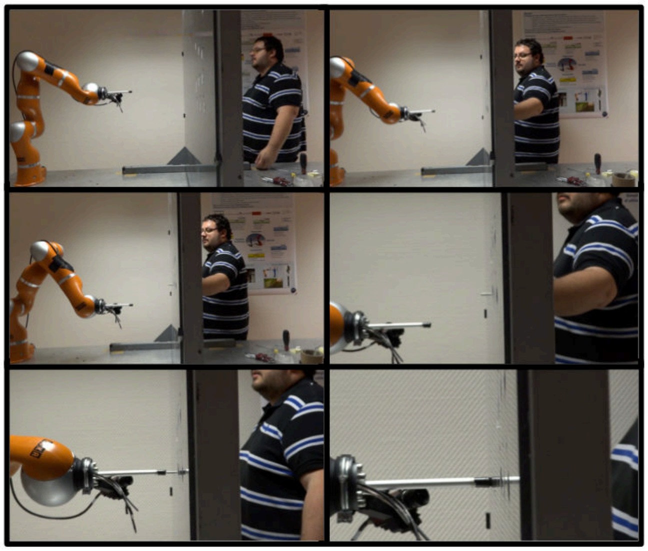

Fig. 4. Snapshots of the experiment of collaborative screwing.

An experiment where three screws are touched is shown in the video attached to this paper (the video only shows the first two screws, for duration issues), as well as in Fig. 4. In Fig. 3, we have plotted the components of the error $\mathbf{e} = \mathbf{s} - \mathbf{s}^{*}$ (top) and of the articular velocities $\dot{\mathbf{q}}$ (bottom) during the experiment. We have also indicated with the acronyms HA, SA and CT the different modes of the experiment. The numbers correspond to the inserted screws (1 to 3). It is clear from the curve that the transitions between modes are abrupt in terms of joint velocities. This is due to the fact that they are not yet managed in our approach (although we plan to do so in future work). Nevertheless, the behaviour of the robot is quite smooth, since the values of $\dot{\mathbf{q}}$ are fed to the Reflexxes online trajectory generation library[6] for smoothing. The shaky behaviour of the HA phases (as opposed to SA and CT) are due to the noisy signal of the Kinect. The strong third component of $\mathbf{e}$ during the SA phase, corresponds to the image depth. This is also the longest movement realized by the robot.

VI. CONCLUSIONS

In this paper, we present a multimodal control approach for human-robot cooperation. The scheme is based on a simple state machine, where all the modes are realized with the same control formalism. The contributions of this paper is a marker-less solution for human intention recognition and human-robot collaboration, and intuitive communication between the two agents, realized through action (specifically, screw inserting). The approach is validated in a mockup screw tightening experiment. This preliminary work opens numerous avenues for future research. In fact, we plan to

[6] www.reflexxes.com

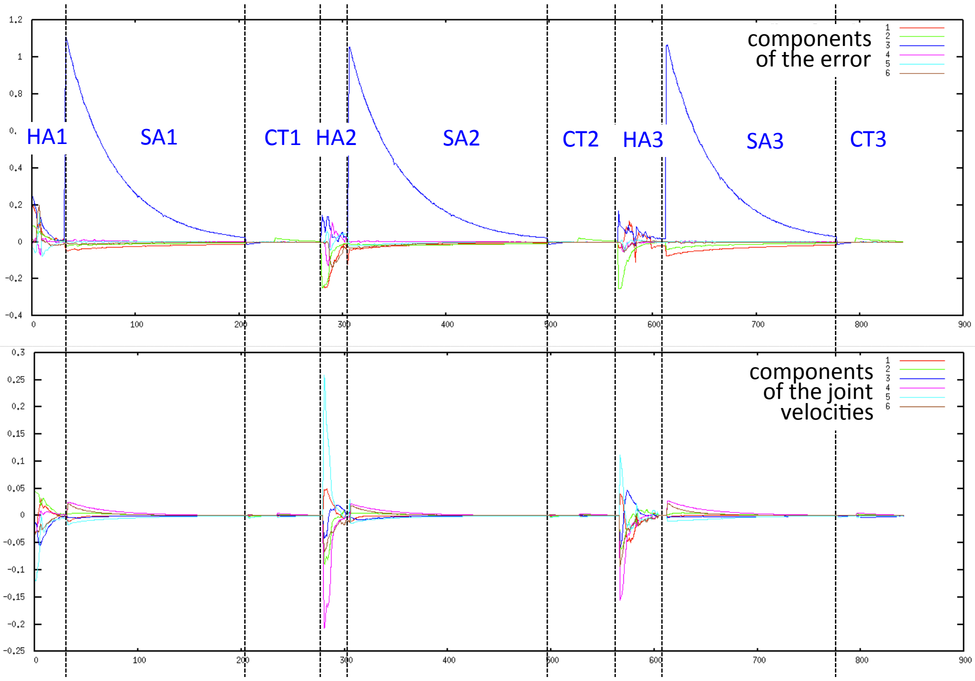

Fig. 3. The six components of the error $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$ (top) and the seven components of the articular joint velocities $\dot{\mathbf{q}}$ (bottom) during the experiment. The error components e_1, e_2, and e_3 for HA and CT are expressed in meters, e_3 for SA is in log meters, and e_4, e_5 and e_6 are in radians. The articular joint velocities are all expressed in rad/s.

utilize force sensing to complete the tightening action (e.g., by considering the forces applied by the human to the screw), to cleanly manage the transitions between the modes (e.g., through homotopy), and to relax the assumptions on the experimental setup. Finally, we plan to extend the framework to distinguish different intentions, involving different human body parts, and not only one hand.

ACKNOWLEDGMENTS

This work has been supported by ANR (French National Agency) ICARO project. The authors would like to thank F. Chaumette for the fruitful discussions on the visual servoing controller, and L. Kerleguer and E. Haug for their help during the experiments.

REFERENCES

[1] A. Bicchi, M. Peshkin and J. Colgate, “Safety for physical human-robot interaction”, in *Springer Handbook of Robotics*, B. Siciliano, O. Khatib (Eds.), 2008, Springer, pp. 1335-1348.

[2] G. Grunwald, G. Schreiber, A. Albu-Schaffer, G. Hirzinger, “Touch: The direct type of human interaction with a redundant service robot”, in *IEEE Int. Workshop on Robot and Human Interactive Communication, ROMAN*, 2001.

[3] M. Awais and D. Henrich, “Proactive premature intention estimation for intuitive human-robot collaboration”, in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012.

[4] A. J. Schmid, O. Weede and H. Worn, “Proactive Robot Task Selection Given a Human Intention Estimate”, in *IEEE Int. Workshop on Robot and Human Interactive Communication, ROMAN*, 2007.

[5] F. Chaumette and S. Hutchinson, “Visual servo control”, *IEEE Robotics and Automation Magazine*, 2006, Vol. 13, no. 4, pp. 82–90 and 2007, Vol. 14, no. 1, 2007, pp. 109–118.

[6] S. M. La Valle, “Planning Algorithms”, Cambridge University Press, 2006.

[7] A. De Santis, V. Lippiello, B. Siciliano and L. Villani, “Human-Robot Interaction Control Using Force and Vision”, *Advances in Control Theory and Applications*, 2007, Vol. 353, pp. 51–70.

[8] Y. Maeda, T. Hara and T. Arai, “Human-robot cooperative manipulation with motion estimation”, in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2001, Vol. 4, pp. 2240-2245.

[9] J. A. Corrales, G. J. Garcia Gomez, F. Torres Medina and V. Perdereau, “Cooperative Tasks between Humans and Robots in Industrial Environments”, *International Journal of Advanced Robotic Systems*, 2012, Vol. 9 No. 94, pp. 1-10.

[10] Microsoft Corporation, 1 Microsoft Way, Redmond, WA 98052-7329, USA, “Microsoft Kinect homepage. http://xbox.com/Kinect (accessed: Feb. 13, 2013)”, *Internet*, 2013.

[11] F. Flacco, T. Kröger, A. De Luca, O. Khatib, “A Depth Space Approach to Human-Robot Collision Avoidance”, in *IEEE Int. Conf. on Robotics and Automation*, 2012, pp. 338-345.

[12] K. Waldron and J. Schmiedeler, “Kinematics”, *Springer Handbook of Robotics, B. Siciliano, O. Khatib (Eds.)*, Springer, 2008, pp. 9-33.

[13] B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo, “Robotics: Modelling, Planning and Control”, Springer, 2009.

[14] E. Marchand, F. Spindler, F. Chaumette, “ViSP for visual servoing: a generic software platform with a wide class of robot control skills”, in *IEEE Robotics and Automation Magazine, Special Issue on “Software Packages for Vision-Based Control of Motion”*, 2005, Vol. 12, no. 4, pp. 40–52.

[15] E. Malis, F. Chaumette and S. Boudet, “2-1/2D visual servoing”, *IEEE Trans. Robot. Automat.*, 1999, Vol. 15, no. 2, pp. 238-250.

[16] KUKA Laboratories GmbH, Zugspitzstrae 140, D-86165 Augsburg, Germany, “Kuka Homepage. http://www.kuka-labs.com/en (accessed: Feb. 13, 2013), *Internet*, 2013.

[17] A. Liegeois, “Automatic supervisory control of configurations and behavior of multibody mechanisms”, in *IEEE Trans. on Systems, Man, and Cybernetics*, 1977, vol. 7, no. 6, pp. 868-871.