2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Human-Humanoid Joint Haptic Table Carrying Task with Height Stabilization using Vision

Don Joven Agravante[1], Andrea Cherubini[1], Antoine Bussy[1,2] and Abderrahmane Kheddar[1,2]

Abstract— **In this paper, a first step is taken towards using vision in human-humanoid haptic joint actions. Haptic joint actions are characterized by physical interaction throughout the execution of a common goal. Because of this, most of the focus is on the use of force/torque-based control. However, force/torque information is not rich enough for some tasks. Here, a particular case is shown: height stabilization during table carrying. To achieve this, a visual servoing controller is used to generate a reference trajectory for the impedance controller. The control law design is fully described along with important considerations for the vision algorithm and a framework to make pose estimation robust during the table carrying task of the humanoid robot. We then demonstrate all this by an experiment where a human and the HRP-2 humanoid jointly transport a beam using combined force and vision data to adjust the interaction impedance while at the same time keeping the inclination of the beam horizontal.**

Index Terms— **Physical Human-Robot Interaction, Human and humanoid skills/cognition/interaction.**

I. Introduction

Humanoid robotics focuses on the design of robots directly inspired by human capabilities. This design gives many advantages when working together with humans in performing tasks. Because of this, humanoids are ideal research platforms for physical Human-Robot Interaction (pHRI). Typically, humans have extensive experience in physically collaborating with each other. Humanoid robots simplify such interactions, since they possess a human-like range of motion and sensing capabilities. These can be used to create suitable behaviors, reducing the need for the human cooperator to learn how to interact with the robot. Although this goal is clear, many challenges are still present in the various research areas. The particular focus of this work is in the integration of visual servoing in pHRI, specifically in human-humanoid collaborative tasks.

Early work on human-humanoid collaboration was done in the Humanoid Robotics Project (HRP), where the HRP-2P humanoid cooperates with a human for a panel transportation task [1]. Here, vision was used to recognize the panel to be grasped. However, this is different from the topic of this paper. Grasping the panel is an interaction of the robot and the panel only. It does not physically involve the human. The focus of the work here is on using visual information during the carrying task where both human and robot already grasped an object on opposite ends and are moving it together. This task has two important aspects:

1) both robot and human are doing jointly the same task,
2) a haptic interaction exists.

A haptic interaction is said to exist since forces applied by one agent (either human or robot) are felt by the other (indirectly through the object). These two conditions characterize "human-robot haptic joint actions". In this research area, the main goal is to regulate the interaction forces for safety and make the robot proactive in helping the human [2], [3]. Since physical interaction is the focus, most works in this field use data from force/torque sensors only to do the task.

Recent advances on human-humanoid haptic joint actions are presented in [4], [5], where the HRP-2 carries a table/beam with a human. This was done by first studying human-human dyads to try and understand how they cooperate for such a task [4]. These observations were exploited to achieve a proactive behavior, using impedance control [6], which can enable the robot to be either leader or follower in the task [5]. Both works: [4], [5] are achieved using only haptic information. To build on these results, we propose the additional use of visual information during the task.

Vision brings a lot of new information that cannot be obtained from force/torque sensors. For example, information about object motion and human gestures/pose may be acquired. However, the integration of such information into human-robot haptic joint actions is not straightforward. The main issues are: what information would be helpful? how can this information be reliably obtained in the context of the task and platform? and how/where should this information be used? Since the priority should still be interaction force regulation, the last issue is particularly important and implies the need for combining vision with force data.

In this paper, a first step to incorporating vision is presented using a particular case of "human-humanoid haptic joint action": carrying the table/beam together while keeping it horizontal. In fact, in [4], [5] the height at which the HRP-2 holds the table is kept constant, with no regard for how the human holds the table at the other end. However, it is desirable to keep the table horizontal to avoid any objects on top from sliding. The other aim is that the robot should be the one to adjust to the human when s/he is leading the task. The strategy here is using vision to observe the table's inclination and then visually servoing the height in order to correct this. This is done while regulating the interaction forces by using impedance control.

Another interesting thing about the height control task is that it is observable by both vision and force/torque

[1]CNRS-UM2 LIRMM UMR 5506, Interactive Digital Human group, 161 Rue Ada, 34392 Montpellier, France {firstnames.lastname}@lirmm.fr.

[2]CNRS-AIST, JRL (Joint Robotics Laboratory), UMI 3218/CRT, Intelligent Systems Research Institute, AIST Central 2, Umezono 1-1-1, Tsukuba, 305-8568, Japan.

sensors. Although the topic here focuses on using vision, this opportunity is taken to compare the force and vision data observed (in the context of this task only).

To start detailing our contribution, the task of human-humanoid table carrying is formally defined in Section II. The same section also presents the simplified model to be used in controlling the height. In Section III, the important concepts of the controller framework developed in [4], [5] (which are still used here) are recalled. The main discussion of the vision algorithm used in estimating the table inclination is given in Section IV. Section V describes how this data is used for height control. The results on the actual humanoid (HRP-2) are then presented in Section VI also showing the comparison with force data. Finally, Section VII concludes and outlines some future works to be done.

II. Task Definition and Model

Fig. 1 illustrates the task, with the reference frames and naming convention used in the rest of this paper. The vectors composing the Cartesian frames are color coded: Red-Green-Blue corresponding to $(\vec{x}, \vec{y}, \vec{z})$ respectively.

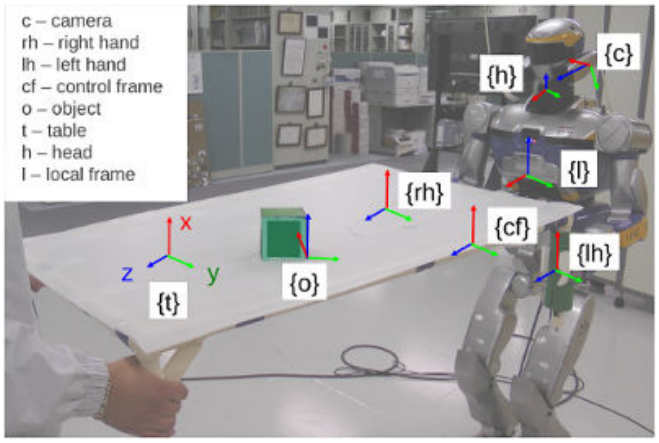

Fig. 1. Human-humanoid table/beam carrying task with the reference frames used in task modeling.

Six degrees of freedom (DOFs) of the table frame $\{t\}$ are available to be controlled. For example, a local frame $\{l\}$ can be used as a reference. In this work, $\{l\}$ is defined such that it moves (only in a planar direction) together with the robot as it walks. Its z-axis points the opposite direction of gravity and the x-axis is directed toward the table, as shown in Fig. 1. The table can then be controlled with the 3 translations ${}^{l}(X, Y, Z)_t$ and the rotation about each axis ${}^{l}(\phi_x, \phi_y, \phi_z)_t$. These are controlled using the humanoid's right and left hands $\{rh\}$ and $\{lh\}$. However, the task is constrained by the fact that human and robot collaborate in controlling the table from opposite ends. Another constraint is due to the robot's rigid grasp of the table. Therefore, the homogeneous transformation matrices ${}^{rh}\mathbf{T}_{lh}$, ${}^{t}\mathbf{T}_{lh}$ and ${}^{t}\mathbf{T}_{rh}$ are all assumed to be constant throughout the task. This allows the arbitrary definition of a "control frame" $\{cf\}$ from which the pose of the two hand frames can be derived.

Because grasping points are predefined, a convenient definition of the control frame origin is the midpoint of the two grasp poses such that the translation, ${}^{cf}\mathbf{t}_{lh} = -{}^{cf}\mathbf{t}_{rh} = [0\ t_y\ 0]^\top$, where t_y is half the distance between the hand frames origins. A further simplification can be done by setting identical orientations for the three frames (hands and control frame): ${}^{cf}\mathbf{R}_{rh} = {}^{cf}\mathbf{R}_{lh} = \mathbf{I}_3$. This also corresponds to the illustration in Fig. 1.

The works in [4], [5] present a method to control the pose (position and orientation) of the control frame. However, only ${}^{l}(X, Y, \phi_z)_{cf}$ are actively controlled (similar to an object moving in a plane) while leaving the remaining 3 DOFs compliant. The method for controlling ${}^{l}Z_{cf}$ is the main subject of this work and is left for the next sections. In fact, controlling the 2 remaining DOFs ${}^{l}(\phi_x, \phi_y)_{cf}$ does not fit with the task of keeping the table horizontal. For example, let the task include maintaining the rotations $\phi_x = 0$ and $\phi_y = 0$ (both are consistent with the task of keeping the table horizontal). If the human decides to tilt the table sideways (rotating ϕ_x), s/he will induce some positive angular velocity $\omega_z > 0$. To maintain the orientation, the robot needs to impose $-\omega_z$. In this case, the human and robot apply opposite torques at opposite ends of the table. This causes a torsion on the table, making it uncomfortable for both and, in the worse case, breaking the table and causing harm to both human and robot. Therefore, to avoid this ambiguity, ${}^{l}(\phi_x, \phi_y)_{cf}$ are left compliant, as in [4], [5]. For a cleaner notation, the reference frames may be left out in some cases. But it follows from the explanation here that the position of $\{cf\}$ is controlled, using $\{l\}$ as a reference.

Since control of ${}^{l}Z_{cf}$ is the main subject here, an appropriate model of the task is needed. A simplified model is preferred in this work and illustrated as follows.

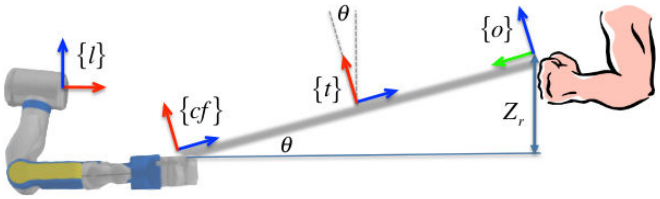

Fig. 2. A simplified "thin beam" model used to control the table height

Fig. 2 shows a simplified diagram of the task, along with the important variables to be used in the next sections. The variable to be controlled is Z_r, which is the height difference between how the robot holds the table and how the human holds the other side. It can be transformed into ${}^{l}Z_{cf}$ by taking into account the location of $\{l\}$. Ideally, $Z_r = 0$ needs to be imposed. However, it is difficult to observe Z_r directly. Instead, θ is used, which is the inclination of the table with length l_t. It is shown later that knowledge of l_t is not required in the final control law.

This simple model does not take into account either human or robot holding the table/beam. There are two conditions that make the simple model valid. Firstly, both the human and humanoid must be compliant (to some extent) when handling

the table together. Otherwise, rigid constraints need to be taken into account with a more complex model (for example: an arc motion where the table pivots at the point where the human holds it). On the humanoid side, the compliance is achieved with the impedance controller which is described in the next section. Secondly, it is assumed that the velocity of both human and robot is limited, so that the compliance can cope with the motion. In practice, this implies that a velocity limit is imposed on the humanoid motions.

III. Controller Framework

Before going into details, an overview of the methodology is presented. Fig. 3 shows a simplified diagram of the control framework along with the two main abstraction layers: an impedance controller and the Stack-of-Tasks. These are briefly described in the corresponding subsections. The innermost robot (motor) control is outside the scope, but it controls the robot through a desired joint control input $\mathbf{q}$.

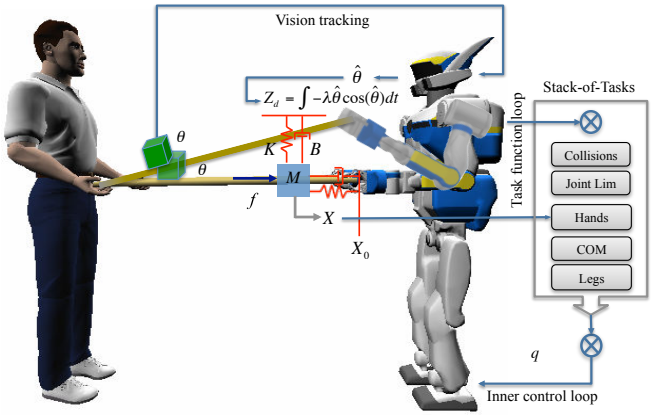

Fig. 3. A simplified block diagram of the control framework

A. Impedance Control

Impedance control is a method for regulating the contact interaction of a robot with its environment [6]. The general equation is:

$$\mathbf{f} = \mathbf{M}(\ddot{\mathbf{X}}_d - \ddot{\mathbf{X}}) + \mathbf{B}(\dot{\mathbf{X}}_d - \dot{\mathbf{X}}) + \mathbf{K}(\mathbf{X}_d - \mathbf{X}) \quad (1)$$

where $\mathbf{f}$ is the wrench (force-torque screw) composed of the force and torque vectors. The vectors $\mathbf{X}_d, \dot{\mathbf{X}}_d$ and $\ddot{\mathbf{X}}_d$ are the desired position, velocity and acceleration respectively. Correspondingly, vectors $\mathbf{X}, \dot{\mathbf{X}}$ and $\ddot{\mathbf{X}}$ represent the actual position, velocity and acceleration. Finally, matrices $\mathbf{M}, \mathbf{B}$ and $\mathbf{K}$ are the inertia, damping and stiffness parameters that define the desired virtual mass-spring-damper system [6].

In this work, we use an "admittance controller" which is also described by Eq.(1), but takes as inputs the "desired" values $\mathbf{X}_d, \dot{\mathbf{X}}_d, \ddot{\mathbf{X}}_d$ and $\mathbf{f}$ (from the sensors in the wrists). The robot is then controlled by $\mathbf{X}, \dot{\mathbf{X}}, \ddot{\mathbf{X}}$ – the "actual" values. Furthermore, $\mathbf{M}, \mathbf{B}, \mathbf{K}$ are tuned from experiments. As explained in the previous section all control is done on $\{cf\}$ with $\{l\}$ as the reference.

B. Stack-of-Tasks

The Stack-of-Tasks is a generalized inverse kinematics abstraction layer [7]. As its name implies, the main advantage it gives is the hierarchical organization of different tasks to be executed. This allows efficient integration and abstraction of the different humanoid robot tasks. The complete table carrying task described here can be roughly decomposed into the following: grasping and lifting the table, stable walking [8], maintaining a good posture (while walking and carrying the table) and avoiding joint limits.

IV. Inclination Estimation from Vision

As explained previously, with the simplified model of Fig. 2, vision is used to estimate θ – the inclination of the table. This is used as an error signal for the table height controller, described in Section V. Furthermore, in order to increase the vision processing algorithm, we use only one from the HRP-2's four IEEE1394 embedded cameras. Because of the viewpoint, θ cannot be directly observed on images taken from the HRP-2 and must be extracted from 3D data. Although a variety of ways to extract this 3D data exist, a fast visual tracker is preferred here, and briefly discussed in the first subsection. The following subsections describes the tracker, how it is made more robust to handle walking and finally how θ is derived from the tracker.

A. Visual Tracking

Visual tracking is used to obtain the pose of an object resting on the table (e.g., the cube of Fig. 1 with frame $\{o\}$ linked to it). The pose is represented by the homogeneous transformation matrix ${}^c\mathbf{T}_o$ where the camera frame $\{c\}$ is the reference. Frame $\{o\}$ is defined so that its z-axis corresponds to the vector normal to the table/beam (see Fig. 1 and 2). This vector forms angle θ with the z-axis of frame $\{l\}$.

A comprehensive review of monocular visual tracking methods is given in [9]. From the many available methods, the virtual visual servoing approach is used here for tracking and pose estimation [10]. This method relies on a model-based edge tracker, and is available as part of the visual servoing software library – ViSP [11]. It works by first initializing the projection of the object model onto the image. The edges are then tracked throughout the image, and a robust optimization process is used to obtain ${}^c\mathbf{T}_o$ from fitting the tracked edges onto the model [10]. A visualization of a typical result (${}^c\mathbf{T}_o$) is shown in Fig. 4 (left and middle images). Fig. 4 also shows, in the rightmost image, how edges are tracked in the normal direction [10].

B. Adding Robustness to the Visual Tracker

Reliability can be an issue for visual tracking and even state-of-the-art algorithms can fail [9]. This uncertainty is a problem, especially if its output is to be used for control. Therefore, precautions are taken here to prevent this. A known platform-specific problem in humanoid robots is the motion induced by walking. The effect on the image is a characteristic oscillatory motion [12], [13] (shown in our results video). Furthermore, impact forces resulting from

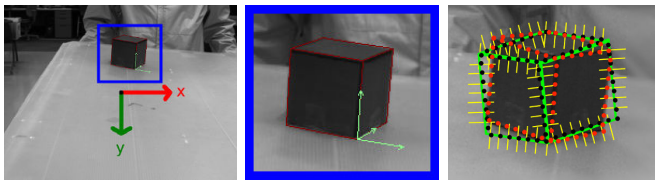

Fig. 4. Typical result of the visual tracker. The full image is at the left. The middle image is a zoomed-in portion bordered by blue with the projection of the cube's model in red, and the object frame in green. The right image shows how edges are tracked.

the walk can cause significant blur on some images. These problems make it necessary to add robustness to the tracker.

Although it is possible to model the cause of these problems (walking) and compensate for it directly [12], [13], a more general approach is taken here that can handle other unforeseen disturbances. More specifically, a fault detection and tracker reinitialization process is implemented. This method is also advocated in [9] as a possible future trend, since all trackers can fail given a difficult enough condition.

The first requirement is an ability to detect a fault. The covariance matrix of the tracker optimization process is used as a "measure of goodness". A "fault" condition arises if $\mathbf{var} > \mathbf{thr}$ where $\mathbf{var}$ is the variance vector (diagonal elements of the covariance matrix) and $\mathbf{thr}$ is a threshold vector that is manually tuned off-line by observing the tracker results from a typical image dataset. When 1 or more of the vector values is at fault, the tracker is reinitialized.

The next requirement is a reinitialization procedure using a good guess of ${}^c\mathbf{T}_o$. The reinitialization itself is trivial and a method is already provided in ViSP. The main problem is estimating ${}^c\mathbf{T}_o$. A tracker failure often indicates that the assumption of continuity between images is invalid. Therefore, the data for reinitialization must come mainly from the current image. Another important consideration is the runtime. Obtaining a guess for ${}^c\mathbf{T}_o$ should be fast enough, so that continuity towards the next images can be safely assumed. Therefore, speed is chosen over generality.

To start, the object needs to be detected in the current image. This is done quickly by thresholding and applying a sliding window for optimal detection. For thresholding, the hue space is used because the object used in this work has a distinct color. To speed up sliding window detection, the concept of image-pyramids is used, with "coarse detection" in a smaller scale version of the image. The result is used for successively "finer" detections up to the original image size. This results in a good localization in image space ${}^I(x, y)$ where x and y are normalized image locations such that:

$$x = {}^cX_o/{}^cZ_o \qquad y = {}^cY_o/{}^cZ_o,$$

with ${}^c(X, Y, Z)_o$ the Cartesian coordinates of the object $\{o\}$ in the camera frame $\{c\}$ (see Fig. 1). A correct pose at the previous iteration $t-\Delta t$ (Δt is the control time step) can be used as a guess for the object orientation ${}^c\mathbf{R}_o^{t-\Delta t}$ and depth ${}^cZ_o^{t-\Delta t}$, so that the new pose, at the current iteration t, is defined as:

$${}^c\mathbf{t}_o^t = \begin{bmatrix} x^t \cdot {}^cZ_o^{t-\Delta t} \\ y^t \cdot {}^cZ_o^{t-\Delta t} \\ {}^cZ_o^{t-\Delta t} \end{bmatrix} \qquad {}^c\mathbf{R}_o^t = {}^c\mathbf{R}_o^{t-\Delta t}.$$

Although this new pose is imperfect, it is a good guess for reinitializing the tracker. Furthermore, the assumptions used fit with the table carrying task done by a humanoid robot, namely: the depth to the object ${}^cZ_o$ is fairly constant, the rotation of the object is minimal and most of the perturbation from walking results in a perturbation in image space ${}^I(x, y)$.

Lastly, another covariance check is done after the tracker is reinitialized. In the event that even the reinitialization fails, a failure signal is produced such that the visual servo controller also stops thus preventing any erroneous motions. This is more of a safety measure, since the tracker reinitialization worked well throughout the experiments.

C. Calculating Inclination

Referring back to Fig. 2, θ is defined using $\{l\}$ as the reference. However, visual data gives ${}^c\mathbf{T}_o$ and as such a change of frame is needed:

$${}^l\mathbf{T}_o = {}^l\mathbf{T}_h\,{}^h\mathbf{T}_c\,{}^c\mathbf{T}_o. \tag{2}$$

${}^h\mathbf{T}_c$ is the pose of the camera in the robot's head frame. It is a constant matrix obtained from an off-line camera-robot calibration procedure. The pose of $\{h\}$ in the local frame (${}^l\mathbf{T}_h$) is available from proprioception. θ can then be extracted from the rotation matrix of ${}^l\mathbf{T}_o$, ${}^l\mathbf{R}_o$ by

$$\theta = \arctan(-r_{13}, r_{33}) \tag{3}$$

where r_{ab} is the element at row a column b of ${}^l\mathbf{R}_o$. Eq.(3) is obtained from the relationship between axes that a rotation matrix represents. The z-axis of $\{o\}$ is the column vector where $b = 3$, since $\{l\}$ is the reference, the important components are in the x-axis ($a = 1$) and z-axis ($a = 3$). The final result θ, is only dependent on the rotations of Eq. (2) and the program implementation can be optimized as such. Furthermore, only results where $\frac{-\pi}{2} < \theta < \frac{\pi}{2}$ are considered valid. The limits correspond to the table being fully vertical and provide another safety measure.

V. Visual Servoing of Inclination

Visual Servoing [14] enables the direct use of visual information in the controllers. It has also been successfully used together with impedance control achieving good results in [15], [16]. Thus, the approach here is to design a visual servo controller to generate the desired reference trajectory of the impedance controller: $\mathbf{X}_d$ in Eq. (1). The trajectory is properly generated since the impedance controller (200Hz) has a significantly faster loop rate than vision (15 fps).

To start the design, the error e needs to be defined. Fig. 2 shows that θ, the angle between the table normal and the vertical is suitable such that: $e = \theta - \theta^*$, where θ^* denotes the desired value of θ. To define a task that keeps the table horizontal, $\theta^* = 0$ making $e = \theta$. The vision algorithm to estimate θ was described in Section IV. The design of the

control law to drive e to zero is shown here. To continue, a suitable model of the task is needed. This model is obtained from Fig. 2 by trigonometry:

$$l_t \sin\theta = Z_r \tag{4}$$

Eq. (4) relates the observed angle θ to the height difference (Z_r) and the table length (l_t). Differentiating with respect to time and rearranging the terms results in

$$\dot{\theta} = \frac{\dot{Z}_r}{l_t \cos\theta} \tag{5}$$

Eq. (5) is the model of the system and the controller can be derived from this. If, for example, an exponential decrease of the error is desired, it must be $\dot{e} = \dot{\theta} = -\lambda\theta$. Since the table length l_t is constant, it can be considered as part of the gain parameter λ. The control law then becomes:

$$\dot{Z}_r = -\lambda\hat{\theta}\cos\hat{\theta}, \tag{6}$$

where $\hat{\theta}$ represents the estimate of θ. If the estimation is perfect ($\hat{\theta} = \theta$), plugging (6) into (5) yields: $\dot{\theta} = -\frac{\lambda}{l_t}\theta$. This shows that l_t contributes only to the convergence speed, and as mentioned it is not necessary to know its value. It only affects the tuning of the gain λ.

To use this result in the impedance control framework, $\dot{Z}_r$ is numerically integrated such that Z_r at the current digital time step is obtained as: $Z_r^t = Z_r^{t-\Delta t} + \dot{Z}_r^t \Delta t$. Lastly, a constant velocity is assumed throughout the time step such that $\ddot{Z}_r = 0$. The results here $(Z_r, \dot{Z}_r, \ddot{Z}_r)$ are then used as the Z part of $\mathbf{X}_d, \dot{\mathbf{X}}_d, \ddot{\mathbf{X}}_d$ in the impedance controller, taking into account the difference in reference frame (i.e. $\{l\}$ and $\{cf\}$ mentioned in Section II.

VI. Results

Several experiments were performed. Some of these are shown in the accompanying video, which also shows the tracker result (while the humanoid is walking) at the beginning. The first is the simplest involving just the arms of the robot standing in place. Next the HRP-2 is made to walk in place, introducing the perturbations from walking. Finally a full experiment is done showing that the work here integrates well to the previous works [4], [5]. Fig. 5 shows some qualitative results taken from the accompanying results video. The top pictures show the first test (standing in place). The bottom pictures show the 3rd test with the robot walking together with the human. Early experiments show that holding the table too high while walking can cause the robot to fall down because of the human's pulling force. This can be explained by the fact that more torque is applied on the humanoid since the lever arm is effectively increased the higher the table is held. So the experiments shown here are with a limit to the Z motion to prevent this.

To verify the controller design, a step response of the error (θ) is obtained from an experiment where the human holds his/her end steady with $Z_r \neq 0$ at the beginning (Fig. 5 top left). The controller is then activated and the robot corrects the height difference, ending at Fig. 5 top right. θ for this sequence is plotted in Fig. 6 (left). This result

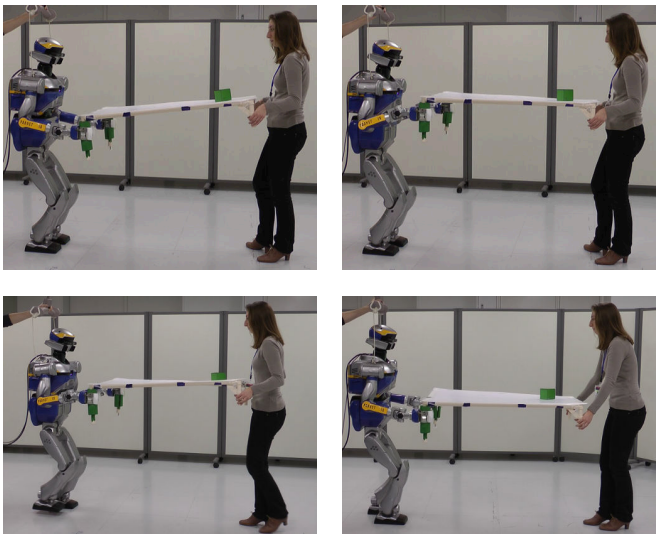

Fig. 5. Image results: top left - before vision control is activated, top right - result after the vision controller is activated, bottom left and right - results while walking with the robot and changing table height

shows an exponential decrease (although with some noise and noticeable latency of the vision algorithm), implying that the implementation follows the design.

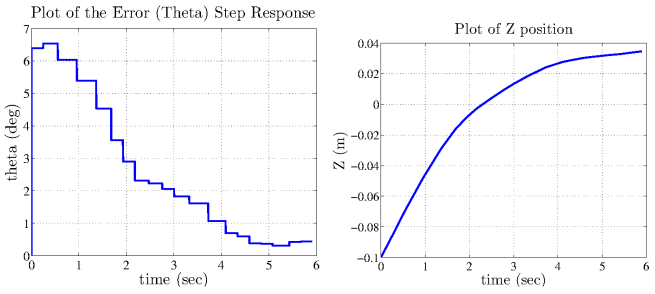

Fig. 6. Step response of the vision controller. Left: Plot of the error (θ). Right: Plot of ${}^{l}Z_{cf}$

As mentioned before, some correlation is observed on the force/torque sensor data and the task of controlling Z. Some preliminary experiments were done to examine the data that can be obtained: moving the table (on the human side) up and down, with some pause in between. The most correlated data are the force f_z and torque τ_y (subscripts referring to axes in $\{l\}$). This data is plotted in Fig. 7 (top). Red lines signify 2 of the transitions when the human moves the other end up/down. Although there is a clear correlation, the data is noisy and there is a bias that is difficult to account for. Finally, the force/torque data is compared with vision data (θ). In Fig. 7 (bottom) the up/down motion is apparent in the raw vision data, but not the raw force/torque data. The advantage of force/torque data is that it is obtained with a higher rate. However there is more noise. It should be noted that additional work can be done to process the raw data (i.e. filtering) before concluding which is better, but this is outside the scope of this paper and left for future works (along with possibly fusing both vision and force data). Although θ from

vision shows to be good from coarse measurements, a future work is to compare the vision data against a more accurate measuring device which can be considered as ground truth.

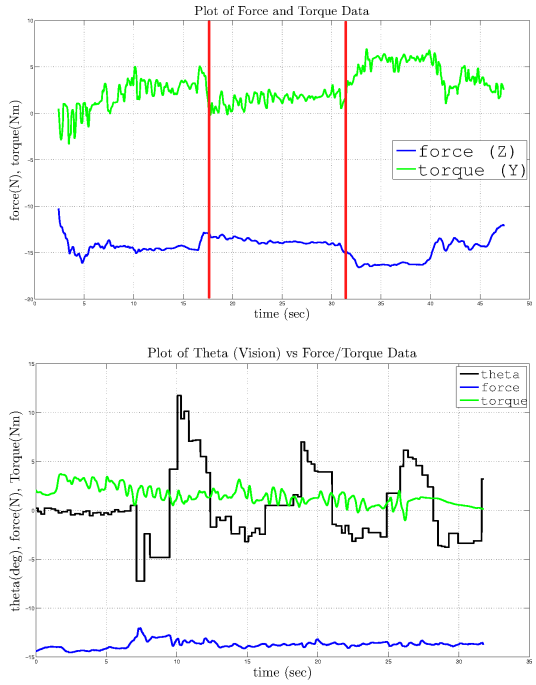

Fig. 7. Data from force/torque sensors. Top: Plot of f_z and τ_y showing correlation to table inclination from preliminary experiments. Bottom: Comparison of force/torque data with vision (θ)

VII. Conclusion and Future Work

In this paper, height stabilization for a human-humanoid table carrying task is presented. We decoupled the control into two orthogonal spaces. The first one is the main focus of this work: it uses visual servoing to define a reference trajectory for the impedance controller. This compensates for the table/beam's eventual inclination which is estimated from a visual pose estimate. The second one regulates the interaction between the human and the humanoid during the transportation of the table/beam. Real experiments conducted with the HRP-2 humanoid robot were successful and show the pertinence of our approach and its implementation.

Yet, our goal behind this early endeavor is to incorporate vision into human-humanoid haptic joint actions control and planning. This work shows results in the integration of visual servoing directly into the control level (although in a decoupled way). It might be interesting to actually merge vision and force in a unified way. However, the overall work is not limited solely to the control level. Adding vision into the cognition level will offer interesting perspectives in investigating the synergy with force/torque data and the incorporation of vision data in multiple levels of the task (i.e. low-level control and high-level cognition). This includes human posture interpretation that can offer a better understanding by the robot of its the human partner's intention and behavior. Moreover, our task will evolve in complexity along time: the cube will be replaced by a moving object or pot full of liquid that must be kept on the table during walking. This will certainly call for revisiting the walking pattern and the stabilizer to account for walking and equilibrium in human-humanoid haptic joint actions.

VIII. Acknowledgement

This work is supported in part by the FP7 IP RoboHow.Cog project (www.robohow.eu). FP7-ICT-2011-7 Contract No 288533. The authors would like to thank François Keith, Pierre Gergondet, Julie Dumora, Damien Petit and Eiichi Yoshida for their help with the experiments.

References

[1] K. Yokoyama, H. Handa, T. Isozumi, Y. Fukase, K. Kaneko, F. Kanehiro, Y. Kawai, F. Tomita, and H. Hirukawa, "Cooperative works by a human and a humanoid robot," in *Robotics and Automation (ICRA), 2003 IEEE International Conference on*, vol. 3, pp. 2985–2991, IEEE, 2003.

[2] Y. Maeda, T. Hara, and T. Arai, "Human-robot cooperative manipulation with motion estimation," in *Intelligent Robots and Systems (IROS), 2001 IEEE/RSJ International Conference on*, vol. 4, pp. 2240–2245, IEEE, 2001.

[3] E. Gribovskaya, A. Kheddar, and A. Billard, "Motion learning and adaptive impedance for robot control during physical interaction with humans," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 4326–4332, IEEE, 2011.

[4] A. Bussy, A. Kheddar, A. Crosnier, and F. Keith, "Human-humanoid haptic joint object transportation case study," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 3633–3638, IEEE, 2012.

[5] A. Bussy, P. Gergondet, A. Kheddar, F. Keith, and A. Crosnier, "Proactive behavior of a humanoid robot in a haptic transportation task with a human partner," in *RO-MAN, 2012 IEEE*, pp. 962–967, IEEE, 2012.

[6] N. Hogan, "Impedance control - An approach to manipulation. I - Theory. II - Implementation. III - Applications," *ASME Transactions Journal of Dynamic Systems and Measurement Control B*, vol. 107, pp. 1–24, Mar. 1985.

[7] N. Mansard, O. Stasse, P. Evrard, and A. Kheddar, "A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: The stack of tasks," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pp. 1–6, IEEE, 2009.

[8] A. Herdt, N. Perrin, P.-B. Wieber, *et al.*, "Walking without thinking about it," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010.

[9] V. Lepetit and P. Fua, *Monocular model-based 3D tracking of rigid objects*. Now Publishers Inc, 2005.

[10] A. I. Comport, E. Marchand, M. Pressigout, and F. Chaumette, "Real-time markerless tracking for augmented reality: the virtual visual servoing framework," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, no. 4, pp. 615–628, 2006.

[11] E. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills," *Robotics & Automation Magazine, IEEE*, vol. 12, no. 4, pp. 40–52, 2005.

[12] C. Dune, A. Herdt, E. Marchand, O. Stasse, P.-B. Wieber, E. Yoshida, *et al.*, "Vision based control for humanoid robots," in *IROS Workshop on Visual Control of Mobile Robots (ViCoMoR)*, pp. 19–26, 2011.

[13] S. Gay, A. Ijspeert, and J. Victor, "Predictive gaze stabilization during periodic locomotion based on adaptive frequency oscillators," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 271 –278, may 2012.

[14] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 4, pp. 82–90, 2006.

[15] G. Morel, E. Malis, and S. Boudet, "Impedance based combination of visual and force control," in *Robotics and Automation (ICRA), 1998 IEEE International Conference on*, vol. 2, pp. 1743–1748, IEEE, 1998.

[16] A. De Santis, V. Lippiello, B. Siciliano, and L. Villani, "Human-robot interaction control using force and vision," *Advances in Control Theory and Applications*, pp. 51–70, 2007.