Whole-Body Stabilization for Visual-based Box Lifting with the COMAN+ Robot

Arturo Laurenzi*, Dimitrios Kanoulas*, Enrico Mingo Hoffman*, Luca Muratore*, Nikos G. Tsagarakis

***Abstract*—In this paper, we present the integration of multiple components of a full-size humanoid robot system, including control, planning, and perception methods for manipulation tasks. In particular, we introduce a set of modules based on visual object localization, whole-body control, and real-time compliant stabilization on the robot. The introduced methodologies are demonstrated on a box lifting task, performed by our newly developed humanoid bipedal robot COMAN+.**

I. INTRODUCTION

In this paper, we integrate the preliminary work in [1] with visual perception and planning in order to autonomously detect and pick up a box from the ground, using the COMAN+ robot. Few works in the past have dealt with this challenging problem, as e.g. [2], [3], [4].

COMAN+ is an 1.6 meters tall, 67 kg, 29 degrees of freedom humanoid robot, which has been recently developed at Istituto Italiano di Tecnologia (IIT), targeting research in collaborative and developmental robotics. The robot is actuated by high power, torque controlled series elastic drives. This endows COMAN+ with high physical capacity, enabling the robot to manipulate high payloads (the payload capacity of each arm is 10 kg). This integration work uses software tools and libraries that were developed over the past few years in the Humanoids and Human-Center Mechatronics team at the Istituto Italiano di Tecnologia (IIT).

II. CONTROL, PLANNING AND PERCEPTION ARCHITECTURE

Figure 1 shows a general block scheme of the control and perception architecture running in the COMAN+ robot. It is possible to recognize two main sub-systems: the planning and perception sub-system (in red) and the control sub-system (in green).

The control sub-system is based on XBotCore [5] and runs in a dedicated real-time PC running a Xenomai micro-kernel. This subsystem is constituted by one non real-time thread, named *CommunicationHandler* and one real-time thread, named *PluginHandler* which communicates each other using `xddp` sockets.

A. The CartesI/O framework

In order to execute the actions that are planned by our perception-aware state machine, we developed a whole-body inverse kinematics engine called *CartesI/O*. Our engine is based on a *client-server* model: the *server* part is in charge of actually solving the inverse kinematics problem, whereas the *client* exposes a full ROS API in order to enable all components of a distributed control system to monitor the solver state, as well as to send appropriate references to it. More in detail, the *CartesI/O client* provides the ability to specify Cartesian references as both online-generated trajectories by continuously publishing to an appropriate ROS topic, as well as point-to-point motions (possibly via user-specified waypoints), which are commanded through ROS actions. On the server side, besides the actual IK solution, additional operations are carried out, as for instance the automatic pre-processing of all Cartesian references in order to enforce velocity and acceleration limits via the *Reflexxes* library, and point-to-point trajectory interpolation. The client and server components can run inside different processes or threads; for the present work, we choose to run our client component on our *CommunicationHandler* non-RT thread, whereas the server side is run inside a *XBot RT Plugin*. The server component solves the required control problem by using the *OpenSoT* framework [6], a RT-safe multi-priority whole-body IK engine.

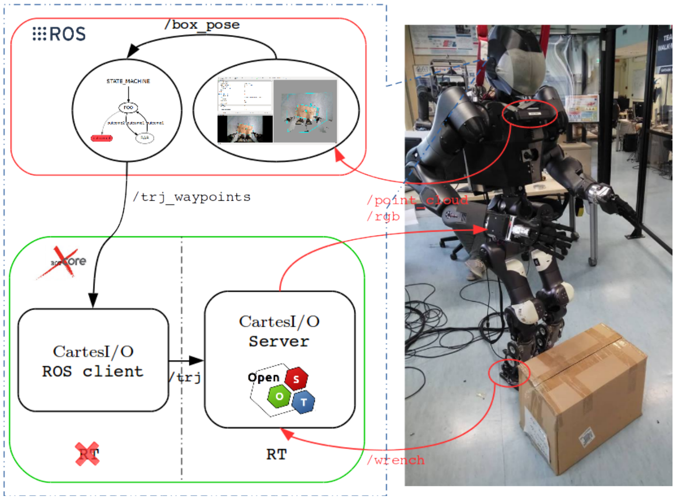


Fig. 1: Software architecture proposed in this work. The red color marks the planning subsystem, while the green color marks the motion control subsystem.

In order to achieve compliant disturbance rejection during the execution of our box-picking task, we employ a *compliant stabilizer* [7] module; by running it inside the server RT thread, we ensure that the whole feedback loop is executed with low jitter, and *deterministic loop frequency*, which is important

---

[1] Humanoids and Human-Centered Mechatronics Lab, Italian Institute of Technology (IIT), Italy, Genoa, {Arturo.Laurenzi, Dimitrios.Kanoulas, Enrico.Mingo, Luca.Muratore, Nikos.Tsagarakis}@iit.it

* all four authors contributed equally to this work

given that any delay contributes to weaken the closed loop stability.

B. Box localization

Perception is important in automatizing robotic tasks in uncertain environments, for a more reliable and faster performance. We use an ASUS Xtion PRO RGB-D sensor kinematically calibrated on the robot's body, providing 640 × 480 grid-based RGB and depth data at 30 Hz. This sub-module implemented in ROS, localizes in real-time the box on the floor in the following way. The floor is localized as the dominant plane in the scene [8], using the RANdom SAmple Consensus (RANSAC) algorithm [9]. For the remaining points, we run RANSAC again and assume that they represent the box's upper surface in front of the robot. The convex hull of this surface point cloud, defines the box size in the lateral direction. For localizing the bi-manual grasping point of the box, we localize its center as the mean point between the floor and the center of the box's upper surface center. This point and the box's lateral size, defines the symmetric box grasping position. The extracted point and the box's size are published into the system as stamped ROS messages in real-time, but used only before the grasping process in an open-loop fashion.

C. Reference generation

In order to script the robot behavior for the box-picking task, we coded a *SMACH*-based [10] state machine that runs inside a Python script. Notice that, thanks to the integration between *XBotCore* dual thread architecture and the *CartesI/O* client-server organization, we are able to easily command the robot in the Cartesian space from a remotely-executed script despite the fact that the *CartesI/O server* is running inside a hard-RT thread.

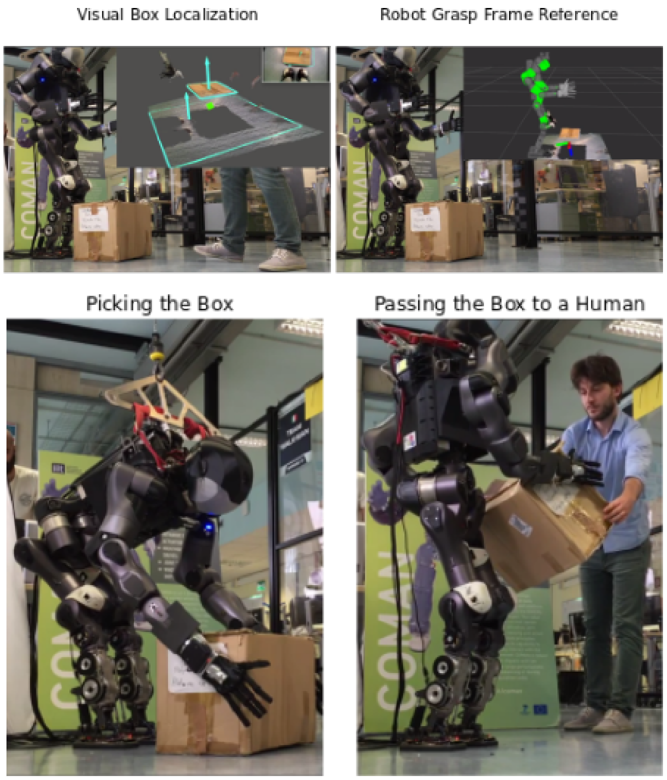
III. EXPERIMENTS AND CONCLUSIONS

To validate the outcome of our integration work, we have performed experiments on the real COMAN+ platform, which are depicted in Figure 2 and also available at https://youtu.be/lEkAKmkR9Rg. First, disturbance rejection capabilities are tested with pushes exerted by the human operator. Then, the perception module starts; it correctly recognizes the lab floor, as well as the box as soon as it is pushed inside the sensor field-of-view. Finally, we run our SMACH state machine, which commands a successful grasp-and-lift motion to the robot.

We further plan to perform similar task with the environment changing dynamically and the robot performing locomotion before and after the lifting task, with heavier objects and the robot being disturbed during the task. The whole framework will be used to assist humans in collaborative and developmental tasks.

ACKNOWLEDGMENT

This research was supported by European Union Horizon 2020 robotics program [ICT-23-2014], grant agreement n.644727 CogIMon.



Visual Box Localization

Robot Grasp Frame Reference

Picking the Box

Passing the Box to a Human

Fig. 2: Snapshots from an experiment with the real COMAN+ hardware.

REFERENCES

[1] E. M. Hoffman, B. Clement, C. Zhou, N. G. Tsagarakis, J.-B. Mouret, and S. Ivaldi, "Whole-Body Compliant Control of iCub: first results with OpenSoT," in *International Conference on Robotics and Automation*, Brisbane, Australia, 2018.

[2] K. Harada, S. Kajita, H. Saito, M. Morisawa, F. Kanehiro, K. Fujiwara, K. Kaneko, and H. Hirukawa, "A Humanoid Robot Carrying a Heavy Object," in *IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 1712–1717.

[3] H. Arisumi, J. Chardonnet, A. Kheddar, and K. Yokoi, "Dynamic Lifting Motion of Humanoid Robots," in *IEEE International Conference on Robotics and Automation*, April 2007, pp. 2661–2667.

[4] S. Nozawa, R. Ueda, Y. Kakiuchi, K. Okada, and M. Inaba, "A Full-Body Motion Control Method for a Humanoid Robot based on On-line Estimation of the Operational Force of an Object with an Unknown Weight," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010, pp. 2684–2691.

[5] L. Muratore, A. Laurenzi, E. M. Hoffman, A. Rocchi, D. G. Caldwell, and N. G. Tsagarakis, "XBotCore: A Real-Time Cross-Robot Software Platform," in *IEEE Int. Conf. on Robotic Computing*, 2017.

[6] E. M. Hoffman, A. Rocchi, A. Laurenzi, and N. G. Tsagarakis, "Robot control for dummies: Insights and examples using opensot," in *Humanoid Robotics, IEEE-RAS Int. Conf. on*, 2017, pp. 736–741.

[7] C. Zhou, Z. Li, X. Wang, N. Tsagarakis, and D. Caldwell, "Stabilization of bipedal walking based on compliance control," *Autonomous Robots*, vol. 40, no. 6, pp. 1041–1057, 2016.

[8] D. Kanoulas, J. Lee, D. G. Caldwell, and N. G. Tsagarakis, "Center-of-Mass-Based Grasp Pose Adaptation Using 3D Range and Force/Torque Sensing," *Int. Journal of Humanoid Robotics*, p. 1850013, 2018.

[9] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[10] J. Bohren and S. Cousins, "The SMACH High-Level Executive," *IEEE Robotics Automation Magazine*, vol. 17, no. 4, pp. 18–20, Dec 2010.