Trajectory-Based Shared Control with Integral Haptic Feedback

Firas Abi-Farraj, Riccardo Spica, Paolo Robuffo Giordano

Trajectory-Based Shared Control with Integral Haptic Feedback

Firas Abi-Farraj, Riccardo Spica, and Paolo Robuffo Giordano

I. INTRODUCTION

Arguably one of the main advantanges of using robots is the possiblility to execute tasks that would be impossible for a human because of the required forces/torques (e.g. heavy load manipulation), the desired accuracy (e.g. micromanipulation applications) or the accessibility of the operating environment (e.g. nuclear facilities, underwater, deep space and so on).

On the other hand, artificial intelligence has not yet reached the level of semantic understanding and complex reasoning of a human and many applications still require human intervention for high-level decision making. Even in presence of a human, however, some level of autonomy is still desirable for easing the task of the operator.

In order to take the best from both worlds many recent works have attempted to allow humans and robots to collaborate on a task [1], [2]. One effective strategy in this contex is telemanipulation [3], [4]. The goal of the RoMaNS project [5], for example, is to assist a human hoperator in remotely manipulating radioactive material. The RoMaNS robotic system (see Fig. 1) consists of two 6-dof serial manipulators. One of the robots is equipped with a gripper that can be used for manipulating objects. The other robot has an onboard camera that can be exploited to maintain a good visibility over the scene. Given the high number of degrees of freedom and the complexity of the system constraints (e.g. singularities, joint limits, field-of-view limitations and so on) partial autonomy must be introduced to facilitate the task.

In a typical shared control framework, a user specifies some *instantaneous* commands for (a subset of) the system degrees of freedom using an input device. A control algorithm executes these commands as accurately and efficiently (according to a proper performance index) as possible while also guaranteeing the safety of the system. *Instantaneous* force/torque cues are then exploited for informing the user about the system performance. A similar strategy was successfully applied to the RoMaNS scenario in [6].

The main limitation of this architecture, however, lies in its *local* nature: the operator cannot modify the *future* behavior of the robot, nor receive informative force cues about the *future* consequences of her/his actions. Moreover, if the system has some redundancy that can be exploited to maximize performance, a reactive/greedy optimization strategy can potentially result in suboptimal solutions.

The main novelty of our approach is that we apply the shared control architecture at a trajectory planning level: the user commands, the system actions and the force feedback are all defined over extended *trajectories* instead of instantaneous quantities (e.g. current system position, velocity and so on).

In the following paragraph we briefly summarize the proposed approach, which extends previous results presented in [6] and [7]. We then present some experimental results obtained in a simulation environment.

F. Abi-Farraj and P. Robuffo Giordano are with the CNRS at Irisa and Inria Rennes Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France firas.abi-farraj,prg@irisa.fr

R. Spica is with the Department of Aeronautics and Astronautics at Stanford University, 496 Lomita Mall, Stanford, CA 94305, USA rspica@stanford.edu

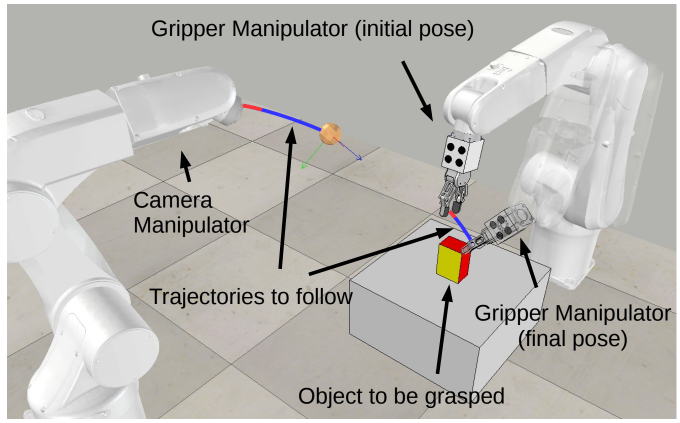

Fig. 1. the experimental setups showing the two 6-dof serial manipulator arms equipped with a camera and a gripper, respectively, together with the object to be grasped.

II. SHARED CONTROL ARCHITECTURE

Let us represent the trajectories of the gripper ($\mathcal{G}$) and the camera ($\mathcal{C}$) in a parametric form as

$$\boldsymbol{\eta}_{\mathcal{G}}(s|\boldsymbol{\theta}_{1,\mathcal{G}},\dots,\boldsymbol{\theta}_{l,\mathcal{G}}) \in \mathrm{SE}(3)$$
$$\boldsymbol{\eta}_{\mathcal{C}}(s|\boldsymbol{\theta}_{1,\mathcal{C}},\dots,\boldsymbol{\theta}_{l,\mathcal{C}}) \in \mathrm{SE}(3),$$

where $s \in [0,1]$ is a scalar line parameter and $\boldsymbol{\theta}_{i,\mathcal{G}}, \boldsymbol{\theta}_{i,\mathcal{C}} \in \mathrm{SE}(3)$ are the coefficients of the parametrization defining the *shape* of the trajectory. Let us stack these coefficients in a single vector $\boldsymbol{\theta}$. In our implementation, we exploited the classical B-splines [8] and the quaternion B-splines [9] to represent the trajectories, but our approach can also be applied to other parametrizations.

Figure 2 illustrates the proposed framework.

Given an initial trajectory for the gripper and the camera (i.e. an initial value of $\boldsymbol{\theta}$), the human operator can express the desired modification of the trajectory via an input device. The input device configuration $\boldsymbol{\lambda}$ is linearly mapped into a desired velocity for the control points $\dot{\boldsymbol{\theta}}_H = Q\boldsymbol{\lambda}$.

At the same time, an *autonomous corrector* generates an additional velocity term $\dot{\boldsymbol{\theta}}_A$ so as to minimize a potential function that goes to infinity if any of the system constraints is violated. In this work, in particular, we considered the following constraints: joint limits, singularities, and visual

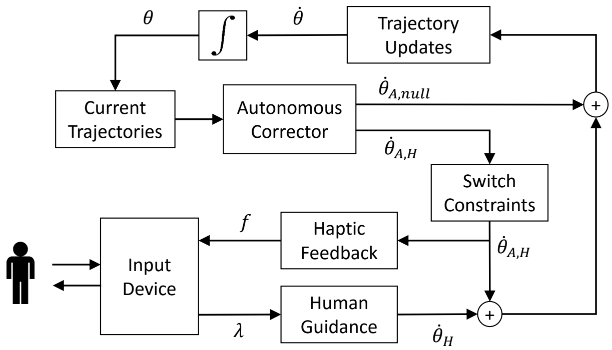

Fig. 2. An illustration of the proposed shared control framework.

constraints. The latter are designed so as to ensure that the object and the gripper remain in the field view of the camera while preventing one from obscuring the other. The potential can also encode additional performance metrics that should be optimized depending on the application. Finally note that the value of this potential is calculated as the integral of a pointwise cost along the entire trajectories, i.e. for $s \in [0, 1]$.

The velocity term $\dot{\boldsymbol{\theta}}_A$ generated by the autonomous corrector is divided into two terms:

- $\dot{\boldsymbol{\theta}}_{A,null}$ which is continuously active and acts in the null space of the human commands $\dot{\boldsymbol{\theta}}_H$
- $\dot{\boldsymbol{\theta}}_{A,H}$ which acts in the same space as the human and is activated only in the proximity of constraints.

This ensures that the autonomous corrector, while continuously commanding the free degrees of freedom of the trajectory to keep the system as far as possible from constraints, does not interfere with the human preference except when necessary to ensure the stability of the system and keep it away from 'dangerous' configurations.

In this latter case, the system generates force cues proportional to the discrepancy (due to the abovementioned autonomous factor $\dot{\boldsymbol{\theta}}_{A,H}$) between the commanded trajectory modifications and the actual ones. These cues are fed to the human through a haptic interface informing him about the performance of the system and guiding him away from undesired system configurations.

III. Experiments and Results

In the conducted experiment, the operator is commanding the system to steer the gripper towards a desired grasping pose. The gripper was constrained to be automatically oriented towards the object at its final pose (constraining two degrees of freedom) while the operator was given command over the remaining four degrees of freedom defining the pose. As the operator modifies the trajectory, an autonomous corrector is actively preventing him from hitting the system constraints as described in section II.

Figure 3 shows the results of the described experiment. Fig. 3 (b), top, depicts the user commands while Fig. 3 (b), bottom, shows the force cues he received. The operator commanded a chosen motion direction up until the system approached a constraint where he received a force feedback over the direction he was commanding and other directions along which the trajectory was adapted to keep the system away from the corresponding constraints. This was repeated three times for different motion directions with similar results.

Fig. 3 (a), on the other hand, shows the different constraints which the operator approached during the experiment. The measure is zero when the system is far from a constraint and starts to increase as it gets closer from a pre-defined threshold. The impact of this proximity to the constraints is reflected as force cues fed to the user. This can be significantly noticed at t=8, 20 and 35 sec.

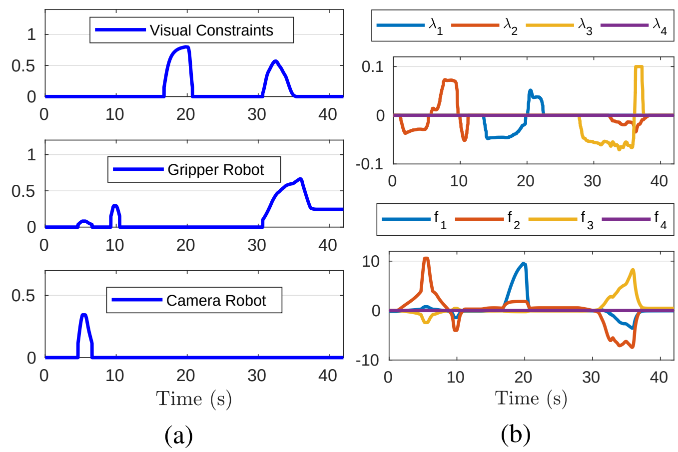

Fig. 3. The figure shows the results of the conducted experiments. (a) shows the different constraints approached while the operator was manipulating the trajectory while (b) shows his commands over the 4 different motion directions he was controlling (top) and the force cues he received over each (bottom).

Acknowledgments

This work was supported by the EU H2020 RoMaNS project 645582.

References

[1] T. Inagaki, "Adaptive automation: Sharing and trading of control," *Handbook of cognitive task design*, vol. 8, pp. 147–169, 2003.

[2] C. Masone, P. Robuffo Giordano, H. H. Bülthoff, and A. Franchi, "Semi-autonomous Trajectory Generation for Mobile Robots with Integral Haptic Shared Control," in *2014*, 2014, pp. 6468–6475.

[3] H. Boessenkool, D. A. Abbink, C. J. Heemskerk, F. C. van der Helm, and J. G. Wildenbeest, "A task-specific analysis of the benefit of haptic shared control during telemanipulation," *Haptics, IEEE Transactions on*, vol. 6, no. 1, pp. 2–12, 2013.

[4] F. Abi-Farraj, T. Osa, N. Pedemonte, J. Peters, G. Neumann, and P. G. Robuffo, "A learning-based shared control architecture for interactive task execution," in *IEEE Int. Conf. on Robotics and Automation, ICRA'17*, 2017.

[5] RoMaNS, "EU Collaborative Project H2020-ICT-645582," http://www.h2020romans.eu.

[6] F. Farraj, N. Pedemonte, and P. Robuffo Giordano, "A visual-based shared control architecture for remote telemanipulation," in *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Daejeon, Korea, Oct. 2016, pp. 4266– 4273.

[7] N. Pedemonte, F. Farraj, and P. Robuffo Giordano, "Visual-based shared control for remote telemanipulation with integral haptic feedback," in *IEEE Int. Conf. on Robotics and Automation*, 2017.

[8] L. Biagiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*. Springer, 2008.

[9] M.-J. Kim, M.-S. Kim, and S. Y. Shin, "A general construction scheme for unit quaternion curves with simple high order derivatives," in *22nd annual conf. on Computer graphics and interactive techniques*, Los Angeles, CA, USA, Aug. 1995, pp. 369–376.