2017 IEEE World Haptics Conference (WHC)
Fürstenfeldbruck (Munich), Germany
6–9 June 2017

Towards a Self-Collision Aware Teleoperation Framework for Compound Robots

Mario Selvaggio[1], *Student Member, IEEE*, Stanislao Grazioso[2], *Student Member, IEEE*,
Gennaro Notomista[3], *Student Member, IEEE*, Fei Chen[4], *Member, IEEE*

***Abstract*— This work lays the foundations of a self-collision aware teleoperation framework for compound robots. The need of an haptic enabled system which guarantees self-collision and joint limits avoidance for complex robots is the main motivation behind this paper. The objective of the proposed system is to constrain the user to teleoperate a slave robot inside its safe workspace region through the application of force cues on the master side of the bilateral teleoperation system. A series of simulated experiments have been performed on the Kuka KMRiiwa mobile robot; however, due to its generality, the framework is prone to be easily extended to other robots. The experiments have shown the applicability of the proposed approach to ordinary teleoperation systems without altering their stability properties. The benefits introduced by this framework enable the user to safely teleoperate whichever complex robotic system without worrying about self-collision and joint limitations.**

Keywords: self-collision, joint limits, haptic rendering

I. INTRODUCTION

Compound robots are robotic systems composed by multiple modules such as mobile bases, arms or even legs. As such systems consist in more than one robotic subsystem, workspace related analysis (*e.g.* definition of non-colliding zones) can be carried out only once the subsystems are assembled together. Among other important issues that guarantee the safe usage of such robots, self-collision, which is the collision that may occur between any two parts of the overall robotic system, is of great importance especially when the robot is teleoperated using a dissimilar mechanism. Some robotic arms may already have safety bounds at joints level which do not allow the user to specify position which are outside their joint ranges. As these bounds could be opportunely enforced for taking into account the robot self-collision, this limitation may not be sufficient when, for example, the arm is placed a mobile platform. Hence, the necessity to carry out workspace analysis for the integrated robotic system. Autonomous compound robots, especially humanoids, require implementing real-time reactive strategies to cope with possible self-collisions while interacting with humans [1]. This approach can be easily extended and applied to the bilateral teleoperation framework in order to safely teleoperate any compound robot while receiving intuitive force cues through the haptic device. This may constitute an important augmentation, particularly for operations performed in partially or completely occluded regions of the workspace, when dissimilar mechanisms are used as master and slave robots.

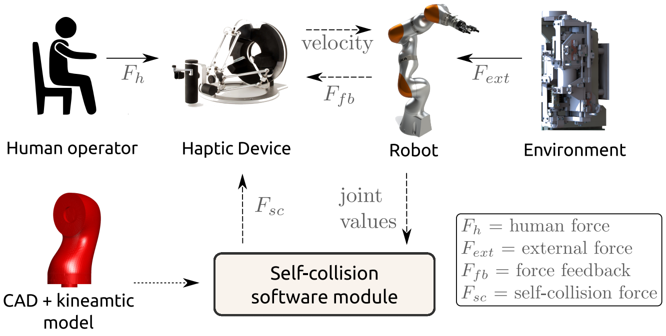


Fig. 1. A qualitative picture of the proposed self-collision aware framework architecture. At the top of the figure the conventional bilateral teleoperation setup can be recognized.

This work aims to lay the foundations of a self-collision aware teleoperation framework for compound robots which will led to a safer and more intuitive teleoperation of complex robotic systems. The proposed framework, despite its generality, solely relies on the robot CAD and kinematic models and on a real-time minimum distance computation and collision detection algorithm. For this purpose the Gilbert-Johnson-Keerthi (commonly abbraviated as GJK) algorithm [2] has been used to calculate the minimum distance between robot shapes pairs. The algorithm has attracted our attention due to its simplicity and generality to work with convex shapes making our framework easily adaptable to other robotic systems. A qualitative picture of the framework is shown in Fig. 1.

For the validation of our method, a series of experiments have been performed using a simulated mobile robot, namely the Kuka KMRiiwa robot[1], composed by an omnidirectional mobile base plus a lightweight seven degrees of freedom arm and equipped with two stereo-cameras and a laser scanning system. Since it is mobile and multi-link, the considered robot is representative of a wide range of compound robots. However, the framework is intended to be easily applicable to any other type of robot, for instance humanoid robots.

---

[1] The author is with Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione, Università degli Studi di Napoli "Federico II", via Claudio, 21, 80125 Napoli, Italy. Email: `mario.selvaggio@unina.it`

[2,3] The authors are with Dipartimento di Ingegneria Industriale, Università degli Studi di Napoli "Federico II", P.le Tecchio, 80, 80125 Napoli, Italy. Email: `stanislao.grazioso@unina.it`, `g.notomista@studenti.unina.it`

[4] The author is with Advanced Robotics Department, Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genova, Italy. Email: `fei.chen@iit.it`

[1] `https://www.kuka.com/`

978-1-5090-1425-5/17/$31.00 ©2017 IEEE

The rest of the paper is organized as follows: Section II contains related works in the field of haptic feedback teleoperation and self-collision; Section III introduces the mathematical theory behind this paper; Section IV contains the overall framework description; in Section V experiments and results are shown and discussed in detail; Section VI concludes the paper.

II. RELATED WORK

Historically, robotic teleoperation systems have made use of virtual fixtures as a perceptual overlay to enhance the human experience in performing remote manipulation tasks. With the virtual fixtures at the master side, the operator is forced to occupy a certain workspace region which is retained to be safe. Rosemberg has pioneered virtual fixtures in his works [3], [4]. Interactive generation of active constraints has been previously investigated by Bettini et al. in [5]. They extended the use of virtual fixtures using computer vision to provide reference trajectory to the control algorithm. More recently, many authors have dealt with the virtual fixtures use in shared control teleoperation. Ferraguti et al. [6] proved a passivity preserving condition for redirection of virtual fixtures forces in surgical robotics. In Boessenkool's work [7] a comparison between shared control and conventional teleoperation in terms of task performances, control effort and cognitive operator workload can be found. In addition, Vozar et al. [8] analyzed the improvement caused by the use of non-holonomic constraints in time delayed space teleoperation, whereas Smisek et al. [9] experimentally quantified the effect of the guidance inaccuracy during a peg-in-hole insertion task. In our previous works [10], [11] we have shown the benefits introduced by an online virtual fixtures generation procedure for teleoperated robotic manipulation tasks. The reduced time spent in programming the system and the level of assistance, which is given to the operator, stressed the importance of this research line. Virtual fixtures can be efficiently used to avoid self-collision and joint limitations. However, the nonlinear mapping between manipulator joint and Cartesian space makes this approach not very attractive. Furthermore, as the slave robot changes so does the sets of forbidden regions which need to be opportunely re-programmed leading to tedious and time consuming interventions.

Collision detection and minimum distance computation are problems embracing a wide range of disciplines, from robotics and control to computer graphics. Several authors have developed/used algorithmic solution to efficiently solve collision problems in robotics [12], [13], [14], [15]. Self-collision is often closely related to humanoid robots [16] in which the generation of collision free movements is essential. On the other hand, haptic rendering of virtual environments, which makes use of minimum distance queries, has been previously investigated by Johnson in [17]. However, a deep investigation of these topics is beyond the scope of this paper. Our primary focus is to transfer this previous knowledge into a shared control telerobotic framework in order to improve the performances of bilateral teleoperation systems for industrial applications.

Telerobotic systems have been widely investigated in the past in terms of performances. Transparency, intended as the capability of transferring the task impedance to the human operator located at the master side, is mainly limited by stability issues. The destabilizing effect of the closed loop system increases with communication delay, unavoidable in any teleoperation architecture. Lawrence [18] showed that it is possible to find a trade-off between transparency and stability regardless of the used teleoperation scheme. Passive behavior can be obtained using a passivity preserving controller which prevents the released energy from being greater than the injected one. One possible choice is to use the two-layer approach from Franken et al. [19].

The contribution of this paper lies in laying the foundations of a teleoperation framework which embeds self-collision avoidance capabilities and gives intuitive feedback to the users. This helps her/him to preserve the safety and stability of the entire teleoperation system. Fast adaptation of the proposed framework to whatever master/slave robotic system makes it very attractive for the use in real industrial scenarios where avoiding failure and/or interruptions is a crucial aspect.

III. BACKGROUND THEORY

The core of proposed teleoperation framework is constituted by a collision detection and minimum distance computation algorithm which is commonly known in robotics as the GJK algorithm [2], [13]. Minimum distance between two bodies is here used to generate the haptic force which guides the operator away from possible self-collisions and joint bounds. In this section we give an overview of the GJK algorithm (Section III-A) and present the mathematics behind the force computation (Section III-B).

A. GJK Algorithm Review

The original GJK algorithm was proposed for the minimum distance computation between two general convex polytopes. The algorithm can be extended to non convex objects by subdividing them into multiple convex shapes. Briefly, the GJK algorithm makes use of a support function defined on a compact set X as $h_X : \mathbb{R}^m \rightarrow \mathbb{R}$ in which the farthest point on a shape in a given direction is calculated, namely

$$h_X(\boldsymbol{\eta}) = \max\left(\boldsymbol{x} \cdot \boldsymbol{\eta} \; : \; \boldsymbol{x} \in X\right) \tag{1}$$

where $\boldsymbol{\eta}$ is a given direction. The solution to Eq. (1) can be expressed as

$$h_X(\boldsymbol{\eta}) = s_X(\boldsymbol{\eta}) \cdot \boldsymbol{\eta} \qquad s_X(\boldsymbol{\eta}) \in X \tag{2}$$

where s_X is the point giving the maximum inner product with $\boldsymbol{\eta}$. When minimum distance between two shapes is of interest, the algorithm only requires the evaluation of h_k and s_k for K_1 and K_2 being compact and convex sets representing the region occupied by the bodies. Thus, the computational effort is proportional to the sum of the vertices

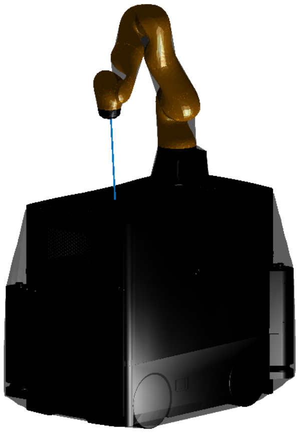


Fig. 2. Example of minimum distance computation and display (blue line) using the GJK algorithm. The involved shapes, or more properly their convex hulls, are the robot mobile base and the arm flange.

in the polytopes M_1+M_2. For simplicity, the calculations can be carried out on the Minkovsky difference of the two regions using the following two equations

$$h_K(\boldsymbol{\eta}) = h_{K_1}(\boldsymbol{\eta}) + h_{k_2}(-\boldsymbol{\eta}) \tag{3}$$

$$s_K(\boldsymbol{\eta}) = s_{K_1}(\boldsymbol{\eta}) - s_{k_2}(-\boldsymbol{\eta}) \tag{4}$$

It is trivial to prove that, if the Minkowski difference contains the origin, the two regions penetrate each other; otherwise the shortest distance between them is the shortest distance between the Minkowsky difference and the origin.

The GJK algorithm iteratively picks simplexes inside the Minkowsky difference closer to the origin until the simplex ceases to change. The termination condition is a function $g_k(\boldsymbol{x}) : \mathbb{R}^m \to \mathbb{R}$ defined as follows

$$g_k(\boldsymbol{x}) = |\boldsymbol{x}|^2 + h_K(-\boldsymbol{x}) \tag{5}$$

If $g_k(x) < 0$ there are no other points $z \in K$ satisfying $|z| < |x|$ and the algorithm returns the latest computed distance; otherwise it constructs the next simplex $V_{k+1} = V_k \cup s_k(-x)$ where V_k has $m \leq 3$ elements.

When the algorithm terminates, the minimum distance between the two initial shapes is the distance between the latest picked simplex and the origin. Geometric approaches can be exploited to calculate the minimum distance between the origin point and lower order simplexes, for instance, line segments, triangles or tetrahedrons. An illustrative example of the minimum distance computation between two robot links is shown in Fig. 2.

B. Force Computation

The haptic information is the second essential part of the proposed framework. It helps the teleoperator to command desired poses for the slave robot keeping it away from both its joint limits and self-collision among its links. In order to obtain the force starting from a generic robot configuration, we need to determine a function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^3$, with n being the dimension of the robot joint space. In this work we take inspiration from performance criteria used in Gradient Projection Method [20] and Weighted Least-Norm solution [21] to handle kinematic constraints (namely joint limits and self-collision constraints) into the inverse kinematic formulation for redundant manipulators.

A performance criterion $h(\boldsymbol{q})$ for joint limit avoidance can be defined as

$$h(\boldsymbol{q}) = \sum_{i=1}^{n} h_i(q_i) = \sum_{i=1}^{n} \frac{1}{\gamma} \frac{(q_{i,max} - q_{i,min})^2}{(q_{i,max} - q_i)(q_i - q_{i,min})} \tag{6}$$

where n is the number of joints, γ is a scalar value, q_i is the $i-$th joint coordinate, $q_{i,max}$ and $q_{i,min}$ are the upper and the lower limits, respectively (see Fig. 3). Its gradient ∇h can be derived analytically and each component ∇h_i will have the following form

$$\frac{\partial h(\boldsymbol{q})}{\partial q_i} = \frac{1}{\gamma} \frac{(q_{i,max} - q_{i,min})^2 (2q_i - q_{i,max} - q_{i,min})}{(q_{i,max} - q_i)^2 (q_i - q_{i,min})^2} \tag{7}$$

It can be noted that $\partial h(\boldsymbol{q})/\partial q_i$ attains zero at the middle of the $i-$th joint and goes to infinity to either limits.

Analogously, a performance criterion for the minimum distance $d \geq 0$ between two non adjacent links may be obtained with a function $c(\boldsymbol{q}, d)$ which has a maximum value at $d = 0$ and decays exponentially toward zero as d increases. Its gradient ∇c may be described as

$$\frac{\partial c(\boldsymbol{q}, d)}{\partial \boldsymbol{q}} = \frac{\partial c(\boldsymbol{q})}{\partial d} \frac{\partial d}{\partial \boldsymbol{q}} \tag{8}$$

In case of self collisions, the second term in (8) may be computed as

$$\frac{\partial d}{\partial \boldsymbol{q}} = \frac{1}{d} [\boldsymbol{J}_A^T (\boldsymbol{p}_A - \boldsymbol{p}_B) + \boldsymbol{J}_B^T (\boldsymbol{p}_B - \boldsymbol{p}_A)]^T \tag{9}$$

where $\boldsymbol{p}_A$ and $\boldsymbol{p}_B$ are the position vectors of the two collision points in the base frame obtained as explained previously, and $\boldsymbol{J}_A$ and $\boldsymbol{J}_B$ are the associated Jacobian matrices. The elements of ∇c represent how each joint angle influences the distance to collision. It is appropriate to select the function c such that its gradient is zero when d is large and infinity when d approaches zero. One possible candidate function is given by

$$c = \rho e^{-\alpha d} d^{-\beta} \tag{10}$$

where α and β control the rate of decay and ρ controls the amplitude (see Fig. 3). Taking its partial derivatives as

$$\frac{\partial c(\boldsymbol{q})}{\partial d} = -\rho e^{-\alpha d} d^{-\beta} (\beta d^{-1} + \alpha) \tag{11}$$

it follows that ∇c may be computed using (8), (9) and (11). Taking into account the kinematic constraints, the haptic feedback force can be finally designed as follows

$$\boldsymbol{f} = -\boldsymbol{J}^{-\mathrm{T}} (\nabla h + \nabla c) \tag{12}$$

where $\boldsymbol{J}^{-\mathrm{T}}$ is the pseudo$-$inverse of the manipulator Jacobian transpose.

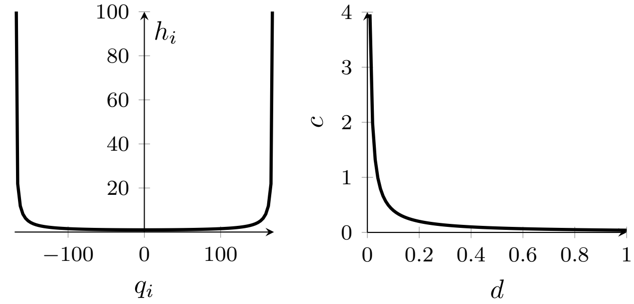


Fig. 3. Plot of the performance criteria used for joint limits (left) and minimum distance (right). In this example, $q_{i,max}$=170, $q_{i,min}$=-170, γ=4, ρ=1/25, α=0, β=25.

IV. FRAMEWORK DESCRIPTION

The overall framework is intended to be applied upon a conventional bilateral teleoperation system so as to endow it with self-collisions avoidance ability (see Fig. 1). A self-collision module is in charge of monitoring the state of the slave robot and output the haptic force to augment the user teleoperation capabilities. The requirements to this block are the CAD and the kinematic models of the teleoperated slave robot, which are essential in order to carry out the computation of the minimum distance. As sometimes the slave robot may be composed by non-convex shapes, some pre-processing is required. Starting from any shape a convex hull construction algorithm should be invoked to construct the smallest convex solid containing all the vertices of the shape. Numerous algorithms exist in computational geometry with various computational cost. However, since the convex hull construction is a process being made offline, its computational complexity does not affect the performance of our framework. Additionally, a self-collision matrix can be built accounting for links which may potentially collide. Assuming adjacent links do not collide thanks to joint limits, the total number of pairs to be checked for collision are $P = \left[n\left(n-3\right)+2\right]/2$ with n being the number of links. The self-collision matrix stores information regarding a pre-process run in which a given number of samples configuration have been generated and collision among the links have been checked. The main purpose of this matrix is to reduce the algorithm complexity at runtime. The self-collision matrix is offline generated for any slave robot and the process complexity does not affect the framework performances. All the pre-processing software has been implemented using MATLAB. Once the convex polygons of the slave robot have been built, the self-collision controller of the teleoperation system can efficiently compute the minimum distance among them at each time step. Algorithm 1 summarizes the explained procedure.

It is worth to note that, despite the fact that the computation may be opportunely carried out by the master side controller, the shapes are updated in accordance with slave joint values. The unavoidable delay in the communication makes the teleoperation system potentially unstable and a passivity controller must be exploited in order to keep the stability. In this work we use the two layers approach from Franken [19].

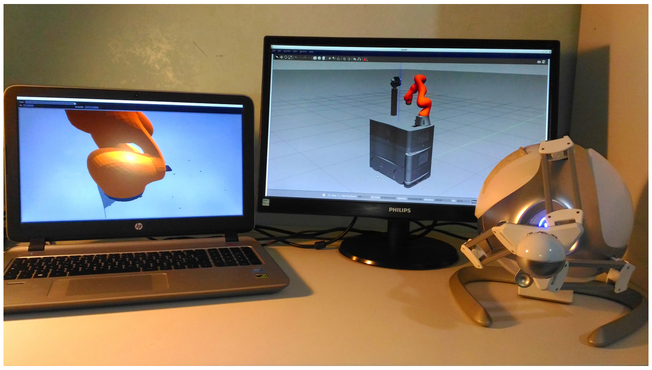


Fig. 4. The setup of the experiments comprising the simulated Kuka KMRiiwa robot and the Novint Falcon haptic device.

V. EXPERIMENTS AND RESULTS

In this section we report the software setup and the experiments conducted on the simulated model of the Kuka KMRiiwa robot. In Section V-A we briefly describe the simulated scenario in which the experiments have been performed, while in Section V-B we extensively analyze the recorded experimental data.

A. *Experimental setup*

The evaluation procedure adopted in this work to demonstrate the validity of the proposed framework consists in showing the haptic force that the user experiences while she/he is teleoperating the slave robot. Two different experiments are reported to isolate the effects of each self-collision force component due to the closest distance (Section V-A.1) and the joint limits (Section V-A.2).

The robot model has been integrated in ROS[2] and simulated through the Gazebo[3] simulator. The experimental setup is shown in Fig.4. The master side consists of a Novint Falcon[4] haptic device. Its three DOFs were used to command the operational space velocities of the slave robot, while its orientation has been left fixed. In the startup phase the offset between the robot and the haptic device is computed, allowing to start from generic initial conditions. The operator monitors the robot arm through the pan-tilt stereo camera

Algorithm 1 Self-collision force feedback computation

```
Require: Convex Shapes, Kinematics, JntValues, CollMatrix
  $S \leftarrow$ load(ConvexShapes)
  $S \leftarrow$ updateShapes(Kinematics,JntValues)
  while true do
    for all $S_i$ in S do
      $d \leftarrow$ GJKalgorithm($S_i$, CollMatrix)
    end for
    $F \leftarrow$ computeForce($d$)
    $S \leftarrow$ updateShapes(Kinematics,JntValues)
  end while
```

---

[2] `http://www.ros.com/`

[3] `http://gazebosim.org/`

[4] `http://www.novint.com/`

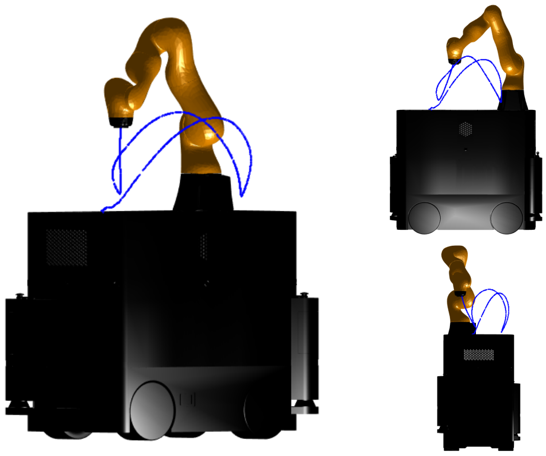


Fig. 5. Robot trajectory recorded during the execution of the *self-collision* experimental test with minimum distance calculation between robot mobile base and arm flange.

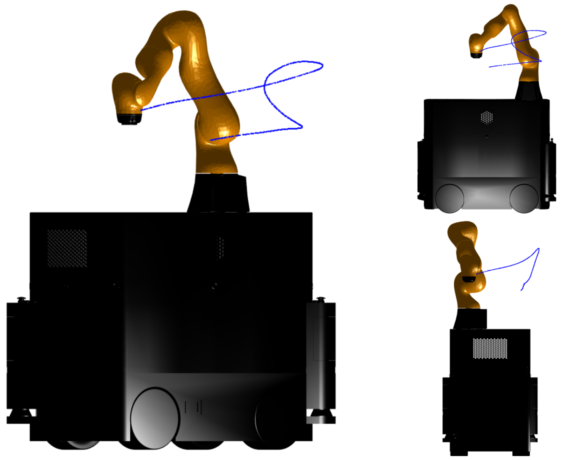


Fig. 7. Robot trajectory recorded during the execution of the *joint limits* experimental test with self-collision checking enabled only on the first joint of the robot arm.

system fixed to the robot mobile base. The commanded velocities of the end effector, as well as the haptic forces, have been opportunely mapped to the stereo-camera reference frame in order to intuitively teleoperate the robot.

1) Self-collision experiment: For the sake of clarity we refer to a simplified self-collision experiment performed computing the minimum distance between solely two bodies of the mobile manipulator, namely, the robot base (12 vertices) and the arm flange (38 vertices). This choice was made in order to present clearer and more intuitive results of the framework application. Specifically, the slave robot starts in the pose shown in Fig. 5, with joint values $jnt = [1.26, 0.4, 0.38, 1.93, 2.95, -0.84, -1.4]$, and it is guided towards the base more than once, in order to emulate a pick and place operation. The user experiences a repulsive force which prevent self-collision (see Fig. 6). With the considered shapes, the non-optimized self-collision routine takes 1 ms on average to perform its calculations on a Intel Core i7-5500U 2,4 GHz CPU 16Gb RAM 1600MHz DDR3L machine, running Ubuntu 14.04. However, the computational time can be further reduced by resorting to a real-time implementation.

2) Joint limits experiment: This experiment is intended to show the capability of the framework to handle joint limitations. For clearness, we show the results when only the first joint of the robot arm is implementing the self-collision functionality. The starting pose of the robot is the same as in the previous experiment and is shown in Fig. 7 together with the robot end effector trajectory. As it can be seen, the robot is guided in directions that push the first joint towards its upper limit being $q_{1,max} = 2.96$ [rad]. The time history of the recorded positions and forces, as well as the first joint angular position, are shown in Fig. 8.

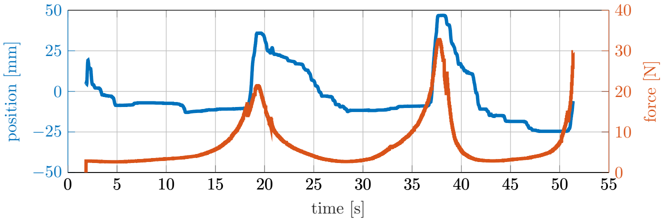


Fig. 6. Haptic device position and relative force recorded during the execution of the *self-collision* experimental test (the data are relative to the z axis which is the axis reporting the most significant component of motion/force experienced by the user).

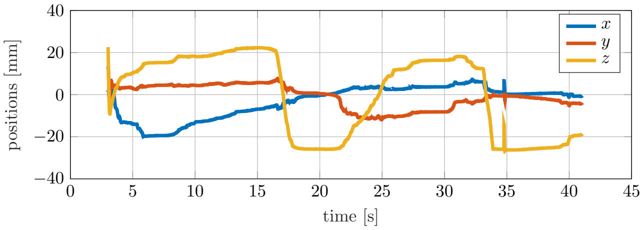
B. Outcomes and Discussion

In this section we analyze and discuss the results obtained from both the performed experiments.

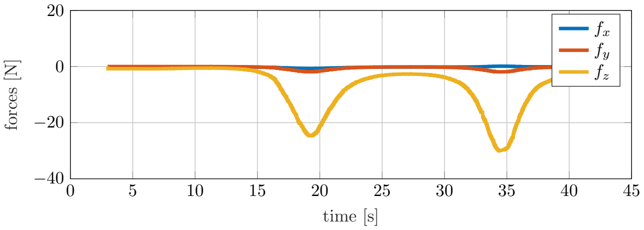
Regarding the *self-collision* experiment we report the commanded incremental pose, as well as the force due to the self-collision algorithm of the haptic device in Fig. 6. As it can be noticed, the user tries to command a velocity in the direction of the base and the repulsive force brings her/him away from the self-collision point. In more detail, the graph contains important information: when the operator commands a velocity to the end effector in the direction of the base, the force starts increasing steadily (between 0 and 18 seconds). At the second 18 the force reaches a significant value so as to move the operator in the counter direction. The damping action of both the human operator and the master device let the overshooting being bounded. Thus, the force decreases till a value close to zero and the sequence is repeated again (in the interval 30-50 seconds).

The *joint limits* experiment shows the same behavior for the system: the haptic force, here intended as the magnitude of the three components shown in Fig. 8, reaches the peak twice (at 19 and 34 seconds) as the joint angle comes close to its upper bound. When the operator feels the guidance force, she/he moves in the counter direction making the force steadily decrease.

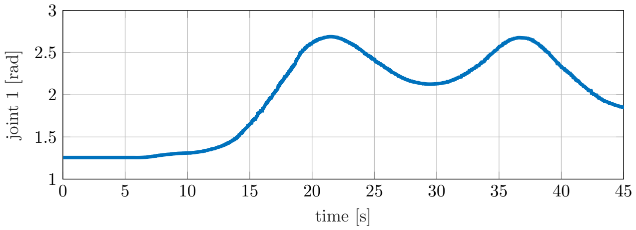
What is important to point out is that the proposed architecture inherently prevents self-collision to happen while giving the operator an intuitive feedback. Our aim is to extend the proposed framework with other important force cues, such as, proximity to singularities and/or collisions with the surrounding environment.



(a) Haptic device recorded positions during the *joint limits* experiment.



(b) Haptic forces due to the movement towards the first joint limitation.



(c) Robot manipulator first joint recorded angle.

Fig. 8. Haptic device recorded data during the *joint limit* experiment. The bottom graph depicts the time history of the joint angle causing the haptic force.

VI. CONCLUSIONS

In this paper we introduced a framework for self-collision aware teleoperation of compound robots. The framework constitutes a general architecture which is easily applicable to a vast variety of robots. By making use of CAD and kinematic models of the slave robot, the framework allows to teleoperate it safely and intuitively. In wider terms, this permits the safe teleoperation of any robot even by an inexperienced user, with no prior knowledge of the system. The haptic cues that the operator receives may be very useful for operations in partially or completely occluded regions of the workspace. The proposed system has been proved to be a valid framework for augmented teleoperation. Furthermore, the proposed framework does not affect the stability properties of the underlying teleoperation system. The performed experiments confirm what has been stated above and provide us with guidelines towards further developments.

REFERENCES

[1] A. D. Santis, A. Albu-Schaffer, C. Ott, B. Siciliano, and G. Hirzinger, "The skeleton algorithm for self-collision avoidance of a humanoid manipulator," in *2007 IEEE/ASME international conference on advanced intelligent mechatronics*, Sept 2007, pp. 1–6.

[2] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE Journal on Robotics and Automation*, vol. 4, no. 2, pp. 193–203, Apr 1988.

[3] L. B. Rosenberg, "The use of virtual fixtures as perceptual overlays to enhance operator performance in remote environments." DTIC Document, Tech. Rep., 1992.

[4] ——, "Virtual fixtures: Perceptual tools for telerobotic manipulation," in *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*. IEEE, 1993, pp. 76–82.

[5] A. Bettini, S. Lang, A. M. Okamura, and G. D. Hager, "Vision assisted control for manipulation using virtual fixtures," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Maui, HI, USA, October 29 - November 3, 2001*, pp. 1171–1176.

[6] F. Ferraguti, N. Preda, M. Bonfè, and C. Secchi, "Bilateral teleoperation of a dual arms surgical robot with passive virtual fixtures generation," in *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, September 2015.

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

[8] S. Vozar, Z. Chen, P. Kazanzides, and L. L. Whitcomb, "Preliminary study of virtual nonholonomic constraints for time-delayed teleoperation," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, Hamburg, Germany*, 2015, pp. 4244–4250.

[9] J. Smisek, M. van Paassen, and A. Schiele, "Haptic guidance in bilateral teleoperation: Effects of guidance inaccuracy," in *World Haptics Conference (WHC), 2015 IEEE*, June 2015, pp. 500–505.

[10] M. Selvaggio, G. Notomista, F. Chen, B. Gao, F. Trapani, and D. Caldwell, "Enhancing bilateral teleoperation using camera-based online virtual fixtures generation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 1483–1488.

[11] M. Selvaggio, F. Chen, B. Gao, G. Notomista, F. Trapani, and D. Caldwell, "Vision based virtual fixture generation for teleoperated robotic manipulation," in *2016 International Conference on Advanced Robotics and Mechatronics*, Aug 2016, pp. 190–195.

[12] R. O. Barr, "An efficient computational procedure for a generalized quadratic programming problem," *SIAM Journal on Control*, vol. 7, no. 3, pp. 415–429, 1969.

[13] E. G. Gilbert and C. P. Foo, "Computing the distance between general convex objects in three-dimensional space," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 1, pp. 53–61, Feb 1990.

[14] D. E. Johnson and E. Cohen, "A framework for efficient minimum distance computations," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 4. IEEE, 1998, pp. 3678–3684.

[15] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Rob. Res.*, vol. 5, no. 1, pp. 90–98, Apr. 1986.

[16] J. Kuffner, K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue, "Self-collision detection and prevention for humanoid robots," in *Robotics and Automation, 2002. International Conference on*, vol. 3, 2002, pp. 2265–2270.

[17] D. E. Johnson, "Minimum distance queries for haptic rendering," Ph.D. dissertation, The University of Utah, 2005.

[18] D. A. Lawrence, "Stability and transparency in bilateral teleoperation," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, pp. 624–637, Oct 1993.

[19] M. Franken, S. Stramigioli, S. Misra, C. Secchi, and A. Macchelli, "Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 741–756, 2011.

[20] B. Dariush, G. B. Hammam, and D. Orin, "Constrained resolved acceleration control for humanoids," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2010, pp. 710–717.

[21] T. F. Chan and R. V. Dubey, "A weighted least-norm solution based scheme for avoiding joint limits for redundant manipulators," in *1993 IEEE International Conference on Robotics and Automation*, May 1993, pp. 395–402 vol.3.

[22] F. Chen, B. Gao, M. Selvaggio, Z. Li, D. Caldwell, K. Kershaw, A. Masi, M. D. Castro, and R. Losito, "A framework of teleoperated and stereo vision guided mobile manipulation for industrial automation," in *2016 IEEE International Conference on Mechatronics and Automation*, Aug 2016, pp. 1641–1648.

[23] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.