2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Event-based device-behavior switching in surgical human-robot interaction*

Mirko Daniele Comparetti[1], Elisa Beretta[1], Mirko Kunze[2], Elena De Momi[3], Jörg Raczkowsky[2] and Giancarlo Ferrigno[1]

***Abstract*— In present days, the number of application in which robots and users share the same workspace is increasing, as long as the need of cooperation between them. To achieve a smooth cooperation, in particular in surgical applications, the robot needs to timely change its behavior to adapt to the needs of the user.**

In this work, a simplified scenario for neurosurgery was defined in which the user interacts with the robot through a Graphical User Interface (GUI) and by touching the robot links and, based to those events and on the current status, different control modes are enabled in the high level controller we developed, such as autonomous, cooperative and teleoperation.

Experiments were performed to measure the performances and safety of the developed high level controller in handling the transitions between two states by checking the continuity of data from the robot and from an external measurement system.

Results proved that the trajectories of the end effector and links during the switching phase are continuous and thus the modular high level controller developed switches control safely without undesired deviation from desired course.

I. Introduction

In present days, in several applications, such as manufacturing and surgical interventions [1], the robot and the user share the same workspace. In general, robots can operate in different control modes, depending on the application, i.e. "Autonomous": the robot automatically drives the tool on a predefined target; "Hands-on": the user manually drives the robot in the space; "Tele-operated": the user movement on the master is replicated by the robot though a suitable mapping. In some applications, it is necessary to switch the control mode of the robot among the different modes according to the particular needs of the user. In [2], the robot interacts with the environment in order to get in contact with a target object and thus the robot needs then to switch from an unconstrained motion in space to a constrained motion with force control after the contact. For doing so, sensors information can be used to trigger the instant in time when to perform the transition between the two controllers; in particular the authors used proximity and force sensors to get the information about the contact with the target and thus to trigger the change of control mode.

Robots in surgery are used in all the three aforementioned control modes: in the autonomous mode the robot automatically positions and directs its flange in order to reach the correct pose, according to the pre-operative plan, e.g. the Neuromate (Renishaw ltd., UK) commercial device [3]. In hands-on control the robot is compliant and the user drives it via a direct contact or through handles [4], [5]. The tele-operation mode can be used to filter out the hand tremor and to increase the dexterity of the surgeon; commercial systems, as the neuroArm [6], the Da Vinci robot [7] and the MiroSurge [8], already use this control mode.

During the execution of a surgical procedure, the surgeon may need to switch from one of the control modes previously mentioned to another one, thus requiring the safety of this procedure via a proper handling of the transition phase. I.e., in the surgical workflow of the ROSA™ (MedTech, France) system [9], [5], the surgeon first performs the registration of the patient via an hands-on robot motion and a laser pointer, then the robot automatically directs the laser to point in the access spot on the patient and then, the robot is guided hands-on to with virtual constraints on the tool motion; the surgeon triggers the switch from one phase to the following one via a Graphical User Interface (GUI).

The signal that triggers the switching between steps can be an event detected by sensors, that, based on the current step, can make the control architecture to react accordingly. The possibility to react in different ways to events leads to the possibility of a flexible event driven scenario, where the following status for the devices is selected from a finite set of possible statuses and is chosen according to the event itself; i.e., an event like the user approaching or touching the robot can be considered as a collision in case the robot is moving autonomously, while it is accepted in case the robot is in hands-on mode and it is manually guided by the user.

In a previous paper [10] we presented the proof of concepts about the high level control in surgery, while in the present paper we present a high level controller, called System Behaviour Supervisor (SBS), which handles parameters for the robot control during the transition among control modes, as events are detected. The events here considered (interaction of the user with the GUI and contact of the user with the robot) depend on the scenario and current step of the workflow.

In Section II the scenario is presented as long as the

*This work was supported by the EU Project Grant FP7-ICT-2009-6-270 460 ACTIVE, Scuola Interpolitecnica di Dottorato and Associazione Laureati del Politecnico

[1]Mirko Daniele Comparetti, Elisa Beretta and Giancarlo Ferrigno are with Politecnico di Milano, Department of Electronics, Informatics and Bioengineerings, NearLab, Milan, Italy

[2]Mirko Kunze and Jörg Raczkowsky are with Karlsruher Institut für Technologie, Institut für Prozessrechentechnik, Automation und Robotik, Karlsruhe, Germany

[3]Elena De Momi is with Politecnico di Milano, Department of Electronics, Informatics and Bioengineerings, NearLab, Milan, Italy and Consiglio Nazionale delle Ricerche, Istituto di Tecnologie Industriali e Automazione, Milan, Italy

devices used, the control architecture and the experimental protocols, while in Section III the results are presented and discussed in Section IV.

II. Scenario, Material and Methods

A. Scenario

The application for which this work has been done is neurosurgery. The control modes that are required are the following:

- Autonomous for brain electrodes insertion in Stereo-Electro-Encephalography (epilepsy invasive diagnostics);
- Cooperative with active constraints (limiting the operative space for safety purpose) for cortical stimulation or resection (epilepsy surgery, tumor removal);
- Tele-operated via master console for remote resection or disconnection (epilepsy surgery, tumor ablation).

All the experiments described in this paper were performed in the "Neuro Engineering and medicAl Robotics Lab – NearLab" of the Department of Electronics, Information and Bioengineering (DEIB) at Politecnico di Milano.

B. The NearLab neurosurgical suite

The set-up used in this paper is showed in Fig. 1.

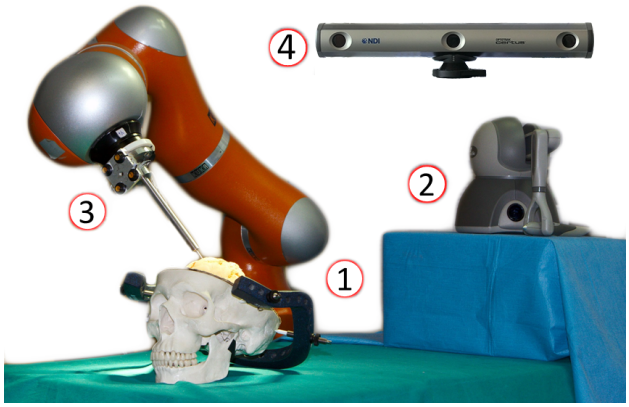


Fig. 1. The setup of the experiment: 1 is the LWR actuator, 2 is the tele-operation master device and 3 are the IR markers for the optical tracking system and 4 is the optical tracker.

It encompasses the LightWeight Robot (LWR) 4+ (KUKA Laboratories, Augsburg, DE) [11], a robot which features 7 Degrees of Freedom (DoFs) an accuracy of 1 mm [12] and a repeatability of 0.05 mm (from specs), as actuator and that can be remotely controlled via a proprietary interface, the Fast Research Interface (FRI), that allows the user to get the internal sensors information and to provide motion commands with control frequencies up to 1 kHz, which is the frequency of the internal controller; the LWR also features torque sensors on each joint, which allows to measure the external torque on the joint itself, compensating the weight of the tool attached to the robot flange. As a master devices, we used the Geomagic Touch (formerly Sensable Phantom Omni – Geomagic, Morrisville, North Carolina), a small serial robot arm which can measure the motion of the handle in 6 DoFs and can provide force feedback in 3 DoFs (cartesian translations).

The OptoTrack Certus optical tracking system (NDI, Ontario, Canada) was used as a supervisor of the system in order to measure and check the performances of the control architecture; its stated accuracy is 0.15 mm in a pyramidal working volume of about 25 m^3.

C. Control architecture

The control architecture described in this paper was developed using different frameworks and middlewares:

- *Common Object Request Broker Architecture (CORBA)*:[1] for the communication between the SBS, GUI and robot controller;
- *Open RObot COntrol Software (OROCOS)* [13]: used to control the robot movements and behavior through modules that run as periodic activity;
- *Robot Operating System (ROS)* [14]: used to provide the CORBA interface between the SBS and OROCOS, and for the Local Area Network (LAN) communication.

The use of middlewares for the communications enables the possibility to have a modular and scalable architecture: in Fig. 2, the different modules are represented with their connections. In particular, the core modules are:

- *GUI*: Provides an interface for the user to select the desired behavior for the robot system, written using C++ and the Qt[2] framework;
- *SBS*: when a switching event is reached the module retrieves the desired parameters and behavior for the devices and forwards that information to the controllers;
- *CORBA wrapper*: the module wraps the information from the SBS and delivers them on a ROS topic to the robot controller module;
- *Robot controller*: manages the transition phase from a control mode to the following one and enables the different functionalities of the other components;
- *Joint admittance*: it implements an admittance controller in joint space [15];
- *Pose provider*: it provides the cartesian pose in the robot Coordinate Frame (CF) in the different modalities;
- *Kinematics*: analytic version of the forward and inverse kinematics [16];
- *Configuration optimizer*: a module to define the constraints on the solution of the inverse kinematics for reduntant robots;
- *Interpolator*: given a target in the task space, it calculates the via points from the source to the target using the trapezoidal velocity profile;
- *FRI modules*: components used to handle the communication of the commands and sensors data to and from the robot through the FRI.

In detail, during the Hands-on mode (cooperative) the *Joint Admittance* module takes as input the current pose of the robot and the current external torque measured from the

[1] `http://www.corba.org`
[2] `http://qt.digia.com`

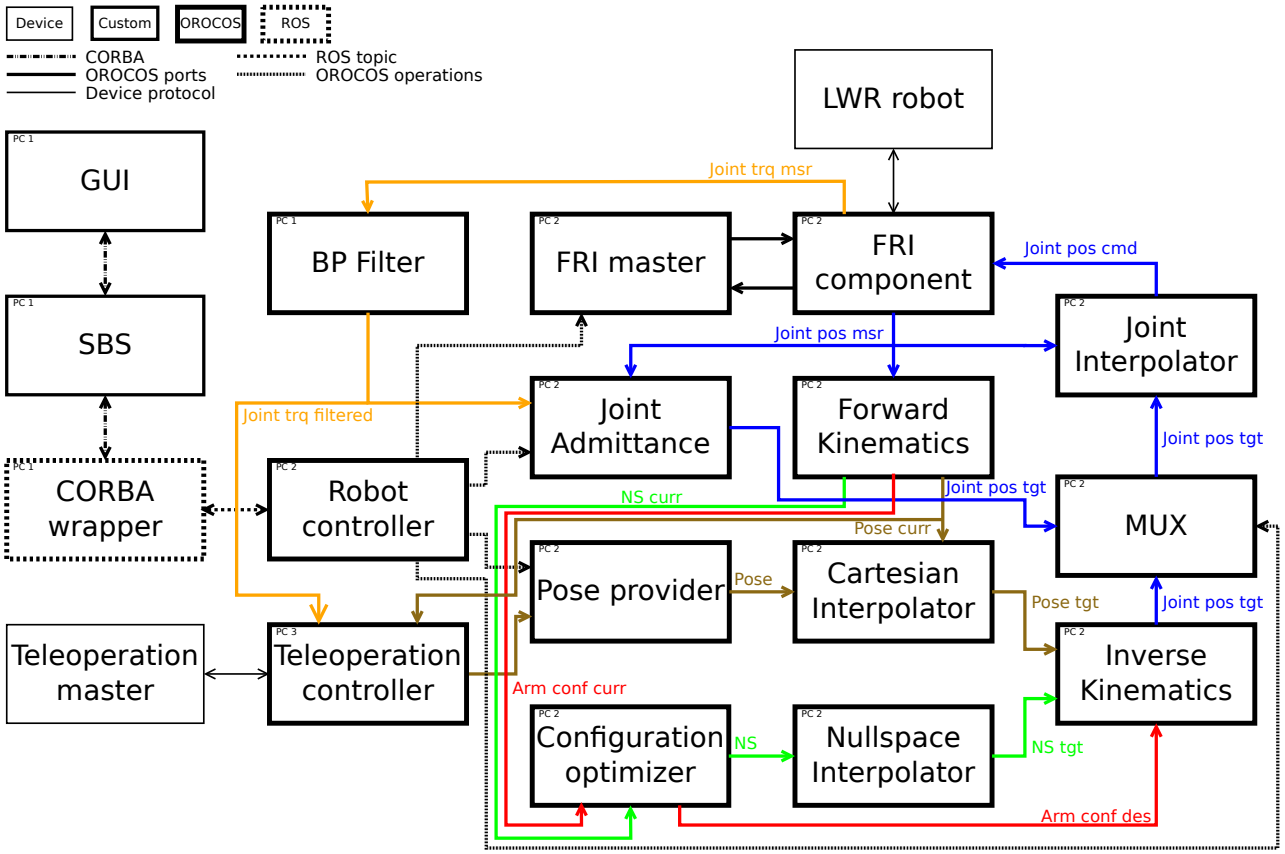


Fig. 2. The control schema: the figure shows the devices as blocks with a thin line and the different components, written using OROCOS for the blocks with thick solid line, ROS for the blocks with dashed line, and standard C++ applications for the blocks with medium thickness borders. The communication lines among the components are described in the legend in top left corner while the colored lines shares the same data type identified by the label on the line itself.

robot joint torque sensors, both expressed in joint space, and, as long as the external torque of at least one joint exceed its joint-specific threshold (meaning that the user is pushing on the robot body), an admittance controller [15] as in eq. 1

$$j_i^{t+1} = j_i^t + G \cdot \tau_i^t \tag{1}$$

controls the robot, where i is the joint index (in $[1, 7]$), t is the time, τ is the external torque, G is the constant gain of the controller, v is the computed velocity given the torque τ, j_i^t is the current joint position from the encoder readings and j_i^{t+1} is the desired value for the joint position.

During Autonomous, Homing (an autonomous movement towards an 'home' pose) and Tele-operation, the *Pose Provider* module gives the target in cartesian space (in the robot base CF), which is then interpolated to create a trapezoidal velocity profile sampled at the robot controller frequency. This data is then fed to the analytic inverse kinematics [16] which, given the constraints on the robot configuration, computes the joint angles. The block *Configuration optimizer* takes care of handling the redundancy of the robot by constraining the configuration of the robot to the one assumed by the robot itself at startup (this does not cause any loss of generality in this analysis). During Tele-operation, the *Pose provider* forwards the data from the *Teleoperation controller* which, given the current robot pose and the desired command from the master side, calculates the desired pose of the robot in the robot base CF.

D. Workflow, states and transitions

A surgical procedure encompasses different steps in which the surgical staff executes different tasks: a robot which has to act as a surgeon assistant has to change its behavior according to the needs of the current step of the intervention.

A simplified scenario was drawn for the current paper, and it foresees the following five steps for the robot:

1) Go to home position (*Homing*);
2) Autonomously approach the patient (*Autonomous*);
3) Hand-guided to the target (*Hands-on*);
4) Move according to a master device (*Tele-operation*);
5) Keep the current pose and stop the procedure (*Steady*).

The different steps require different behaviors for the devices in the environment, thus it is fundamental to switch from one behavior to another during the intervention without stopping the devices or having unpredictable movements due to a mis-handled transition, i.e. bad settings in the initial parameters of the following controller.

The trigger to switch between steps is based on events like pressing buttons or touching the devices, and, for doing so, the surgeon is provided with an interface that allows

the control of the devices: as long as the surgeon requires to switch to the following step, (s)he creates an event by pressing a button on the GUI or by touching the robot, as shown in Fig. 3.

In the GUI is present a button that goes to the next step in the list; in case of the event towards the *Hands-on* step, the trigger event is raised by the contact between surgeon and robot: in fact, the user is allowed to touch the robot only during the cooperative mode, while in all the other steps this contact has to be considered as a fault, the user has to be warned and the robot has to reach a safe state eventually moving it away.

When one of the described events is detected, the SBS is triggered and it loads the parameters for the devices which are stored in a dedicated DataBase (DB), and it streams them through a ROS topic; the *Robot Controller* module, depending on the new set of parameters, enables different features on the other modules, such as the *Joint Admittance* and the *Pose Provider*, using OROCOS operations that are executed in the caller thread. In particular, in the control mode Hands-on, the *Joint Admittance* is enabled, while in the other three modes the *Pose provider* module is enabled; the Multiplexer (MUX) component then connects its output to the correct input depending which is the current active control mode.

E. Experiments

In order to test the behavior of the devices during the transition between different states, the suite described in § II-B was used during the execution of the procedure described in § II-D. During the tests, the Optotrack Certus was used as a supervisor, to acquire through a ROS interface, the pose of a reference frame placed on the robot flange, in order to detect eventual residual movements that can happen during the switching phase. All the devices and modules in the control architecture were sharing the same clock in order to have consistent timestamps among all of them; the components were running at $100\,\mathrm{Hz}$ and the Band Pass (BP) Finite Impulse Response (FIR) filter for the external torques has a bandwidth from $0.5\,\mathrm{Hz}$ to $3\,\mathrm{Hz}$ with an attenuation, in the stopped band, of $-40\,\mathrm{dB}$.

An experiment is divided in six slots with five transitions among the following ordered list of states: Steady, Homing, Autonomous, Hands-on, Tele-operation, Steady. All the transitions are triggered via the GUI, apart from the transition from Autonomous to Hands-on which is triggered by the contact of the user with the robot as in Fig. 3. This experiment was repeated for 19 times.

During the experiment, the following data were acquired:

- Pose of a CF fixed on the robot flange from the optical tracking system;
- Cartesian pose of the LWR from the forward kinematics.

All those data were analyzed to check the continuity over time of the data and their derivative computed using a First-Order Adaptive Window (FOAW) derivative filter [17]; the derivatives were analysed up to the third order to show the behavior of the robot also including the accelerations and its variations.

For each event, the derivatives of the signal in the $0.1\,\mathrm{s}$ before, after and during the event itself were averaged and the population made of these data in the 19 trials were statistically compared using the One Way ANOVA test with Bonferroni adjustment [18] and $\alpha = 0.05$, in order to check if the differences among them are statistically significant.

III. Results

In Fig. 4 an example of the raw data from robot and optical tracking system is shown, along with the instant in which a transition happened and the contact between human and robot was detected. It can be seen that, after the third transition the *Hands-on* mode is enforced and the motion of the LWR is following the torques provided by the user.

Fig. 5 shows an example of the three derivatives of the signal recorded from the optical tracking system, using a time window of $0.1\,\mathrm{s}$. In Fig. 5a it can be noted that the first derivative is close to 0, while the derivatives of higher order present a bigger order of magnitude due to numerical differentiation. Statistical difference was found in the data of the cartesian pose of the LWR, as shown in Fig. 6.

IV. Discussions and conclusions

In this paper, we presented a flexible, scalar and modular high level architecture to manage the switching among different control modes for a robot manipulator. This kind of architecture can be used in applications in which the user is closely interacting with the robot, changing the scenario dynamically at runtime.

The approach chosen in this paper is the definition of a workflow for a simplified surgical procedure in which the user is provided with an interface to the developed control architecture that, based on GUI and sensors input, defines the correct behavior for the devices in the scenario.

In [2] the robot was equipped with external proximity sensors and force sensors to obtain a smooth transition from free motion to contact with the target in a four-stages transition towards the contact; during this transitions, the initial conditions of the following controller are adapted to obtain continuity from the previous state, thus obtaining a reduced impact force with the target.

In a complex and dynamic environment like the operating room, it is important to have the possibility modify and adapt the behavior of the devices according to user input, events and sensor data: during the surgical procedure, the tasks for the robot change and thus the control mode needs to be adapted to cope with the new situation.

The target of this work is the continuity of the robot trajectory during the transition between states, without unpredictable shaking of the robot flange. Results showed that the robot flange and the tool connected to it, are kept in a steady position during the transitions: this is mandatory for application in which the robot is in contact, throg its tool, with the patient organs and tissues, in order to prevent damages to the patient. During the transition from Autonomous

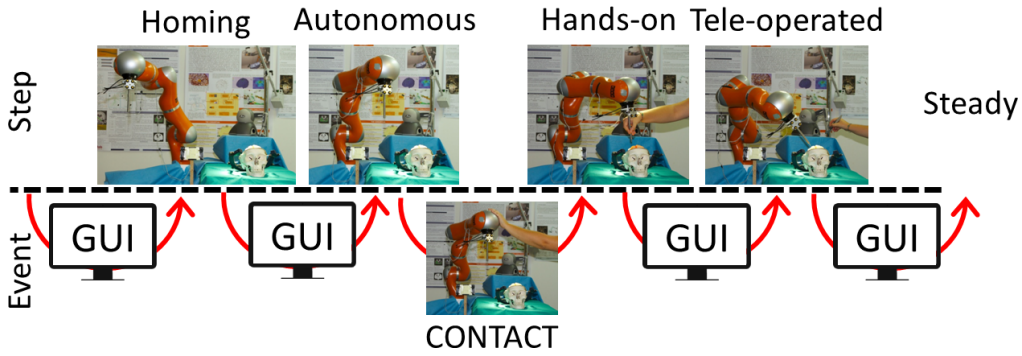


Fig. 3. Animated schema of the possible steps and transitions event: as long as an event is detected, the step goes from the current to the following one, as indicated by the red arrow.

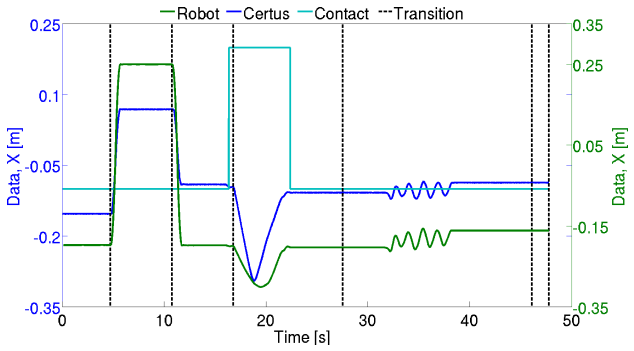
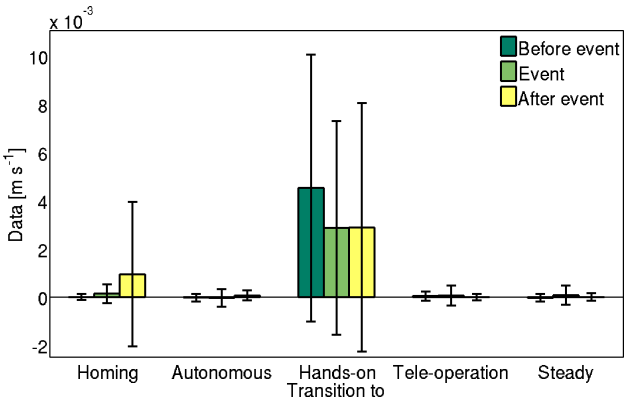


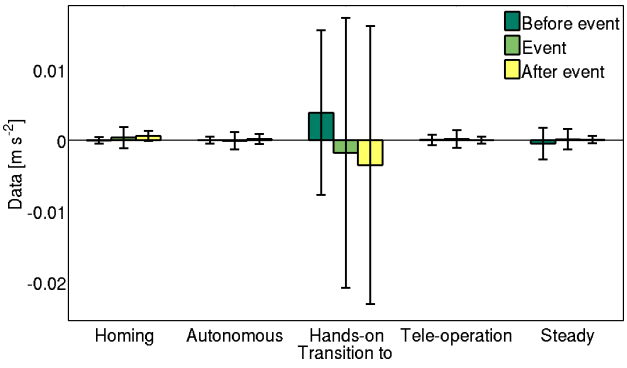
Fig. 4. Example of the recorded data: the blue and green solid lines are the data of the absolute X axis measured in the CF the optical tracking system and robot respectively, the cyan line is the detection of the contact between human and robot (low for no contact, high for contact) and the dashed vertical lines are the instant in which the transition occurred.

mode to Hands-on mode, an higher variability is present in all the data because at the beginning of the contact the user is acting against the robot, which is commanded to keep its current position, causing a small bending of the structure due to the mechanical compliance in the geometry of this kind of serial robots; this, however, doesn't affect the performance of the schema because this transition happens far from the outside of the patient and it is under the control of the user.

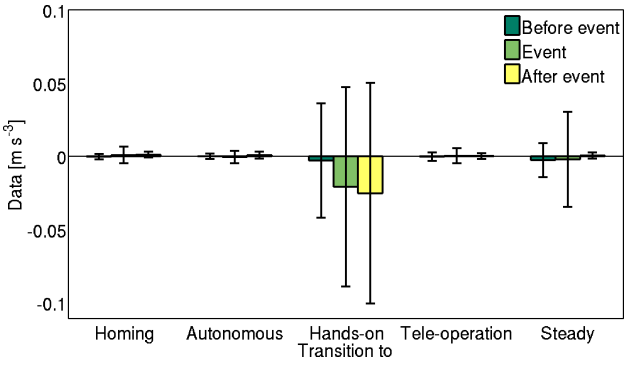
When switching to *Homing* and *Autonomous* mode, the robot starts its movement as long as it receives the event, accelerating at the maximum acceleration allowed by the interpolator in cartesian space. A short time-window on the one hand reduces the effects of the motion of the robot, after the transition has been triggered, but, on the other the contribution of the noise is more important. In the data here showed, the statistical difference found in the robot data during the switching to *Autonomous* mode with a time-window of $0.1\,\mathrm{s}$ is due to the fact that, within that time frame, the robot has already began its movement. The same statistical difference was not revealed by the tracking system because the noise on this data is higher, compared to the one in the forward kinematics based on the encoder of the robot, and this variability masks the statistical difference due to the beginning of the motion. The use of the optical localizer allows to measure the motion of the tip of the tool, carried



(a) First derivative

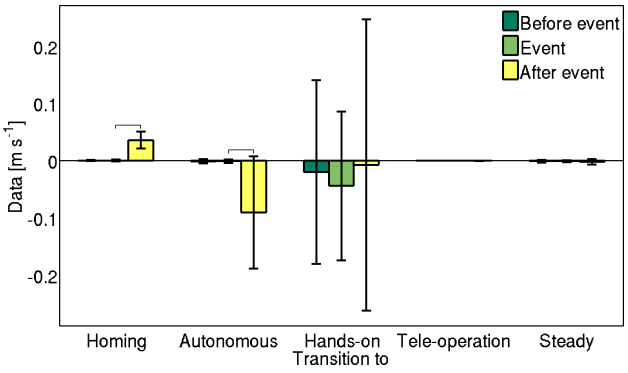


(b) Second derivative

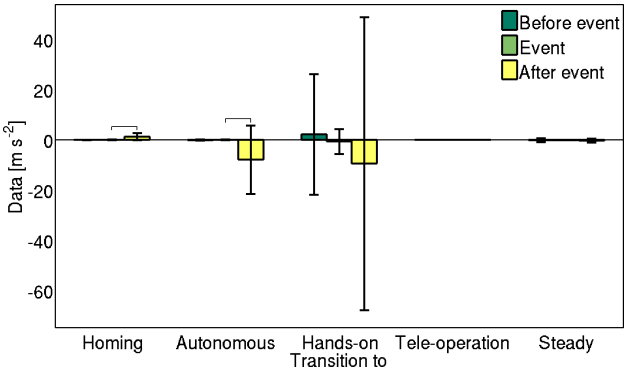


(c) Third derivative

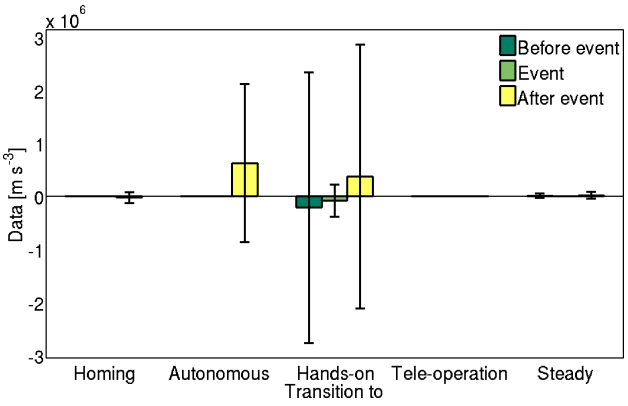
Fig. 5. Data of the optical tracking system measured on the absolute X axis, with a time window of $0.1\,\mathrm{s}$: the mean value and the standard deviation are showed in the interval around the event.



(a) First derivative



(b) Second derivative



(c) Third derivative

Fig. 6. Data of the X axis of the robot pose, with a time window of 0.1 s: the mean value and the standard deviation are showed in the interval around the event and horiziontal lines indicate a statistical difference among the populations.

by the robot and also the possible movements due to the bending of the structure during the contact of the operator with the robot arm.

The presented schema can be expanded by using other type of sensors to detect possible events in the scenario from other sources, to increase the safety of the system. Environmental cameras, such as the Microsoft Kinect™ camera, can be used to foresee and detect possible collisions among the user and the robots, thus providing important information for event detection.

REFERENCES

[1] J. Lew, Y.-T. Jou, and H. Pasic, "Interactive control of human/robot sharing same workspace," in *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1. IEEE, 2000, pp. 535–540.

[2] Y. F. Li, "A sensor-based robot transition control strategy," *The International journal of robotics research*, vol. 15, no. 2, pp. 128–136, 1996.

[3] T. R. K. Varma and P. Eldridge, "Use of the Neuromate stereotactic robot in a frameless mode for functional neurosurgery," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 2, no. 2, pp. 107–113, 2006.

[4] M. Jakopec, F. Rodriguez y Baena, S. J. Harris, P. Gomes, J. Cobb, and B. L. Davies, "The hands-on orthopaedic robot "acrobot": Early clinical trials of total knee replacement surgery," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 5, pp. 902–911, 2003.

[5] G. Hughes, S. Vadera, J. Bulacio, and J. Gonzalez-Martinez, "Robotic placement of intracranial depth electrodes for long-term monitoring: Utility and efficacy," in *Cleveland Clinic*. Cleveland Clinic, 2013.

[6] G. Sutherland, I. Latour, and A. Greer, "Integrating an image-guided robot with intraoperative MRI: a review of the design and construction of NeuroArm." *IEEE engineering in medicine and biology magazine: the quarterly magazine of the Engineering in Medicine & Biology Society*, vol. 27, no. 3, p. 59, 2008.

[7] G. S. Guthart and J. K. Salisbury Jr, "The Intuitive™ telesurgery system: overview and application," in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA, San Francisco)*, vol. 1. IEEE, 2000, pp. 618–621.

[8] U. Hagn, R. Konietschke, A. Tobergte, M. Nickl, S. Jörg, B. Kübler, G. Passig, M. Gröger, F. Fröhlich, U. Seibold, L. Le-Tien, A. Albu-Schäffer, A. Nothhelfer, F. Hacker, M. Grebenstein, and G. Hirzinger, "DLR MiroSurge: a versatile system for research in endoscopic telesurgery," *International journal of computer assisted radiology and surgery*, vol. 5, no. 2, pp. 183–193, 2010.

[9] S. Ferrand-Sorbets, D. Taussig, M. Fohlen, C. Bulteau, G. Dorfmuller, and O. Delalande, "Frameless stereotactic robot-guided placement of depth electrodes for stereo-electroencephalography in the presurgical evaluation of children with drug-resistant focal epilepsy," in *CNS Annual Meeting*, 2010.

[10] M. D. Comparetti, E. De Momi, D. De Lorenzo, T. Beyl, J. Raczkowsky, and G. Ferrigno, "Safe surgical robotic system and workflow design in the active project for awake neurosurgery," in *Proceedings of the 2012 IEEE International Conference on Intelligent Robots and Systems (IROS, Vilamoura), Workshop on Safety in Human-Robot Coexistence & Interaction: How Can Standardization and Research benefit from each other?*, Oct. 2012.

[11] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schäffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, *et al.*, "The KUKA-DLR lightweight robot arm. a new reference platform for robotics research and manufacturing," in *41st International Symposium on Robotics (ISR) and 6th German Conference on Robotics (ROBOTIK)*, 2010, pp. 1–8.

[12] D. Stein, H. Monnich, J. Raczkowsky, and H. Worn, "Visual servoing with an optical tracking system and a lightweight robot for laser osteotomy," in *IEEE International Conference on Control and Automation, 2009*. IEEE, 2009, pp. 1896–1900.

[13] H. Bruyninckx, "OROCOS: design and implementation of a robot control software framework," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) - Tutorial*, 2002.

[14] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *IEEE International Conference on Robotics and Automation, Workshop on Open Source Software*, vol. 3, no. 3.2, 2009.

[15] G. Zeng and A. Hemami, "An overview of robot force control," *Robotica*, vol. 15, no. 5, pp. 473–482, 1997.

[16] M. Shimizu, H. Kakuya, W.-K. Yoon, K. Kitagaki, and K. Kosuge, "Analytical inverse kinematic computation for 7-dof redundant manipulators with joint limits and its application to redundancy resolution," *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 1131–1142, 2008.

[17] F. Janabi-Sharifi, V. Hayward, and C.-S. Chen, "Discrete-time adaptive windowing for velocity estimation," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 6, pp. 1003–1009, 2000.

[18] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian journal of statistics*, pp. 65–70, 1979.