# Monitoring Companion for Industrial Robotic Processes

Enrico Sita

A thesis submitted for the degree of Doctor of Philosophy

Funded by PPM Robotics AS, Norway
And the Norwegian Research Council

Bristol Robotics Laboratory
Faculty of Environment and Technology
University of the West of England, Bristol

January 2021

# Abstract

For system integrators, optimizing complex industrial robotic applications (e.g. robotised welding) is a difficult and time-consuming task. This procedure is rendered tedious and often very hard to achieve when the operator cannot access the robotic system once in operation, perhaps because the installation is far away or because of the operational environment. In these circumstances, as an alternative to physically visiting the installation site, the system integrator may rely on additional nearby sensors to remotely acquire the necessary process information. While it is hard to completely replace this trial and error approach, it is possible to provide a way to gather process information more effectively that can be used in several robotic installations. This thesis investigates the use of a "monitoring robot" in addition to the task robot(s) that belong to the industrial process to be optimized. The monitoring robot can be equipped with several different sensors and can be moved into close proximity of any installed task robot so that it can be used to collect information from that process during and/or after the operation without interfering. The thesis reviews related work in the industry and in the field of teleoperation to identify the most important challenges in remote monitoring and teleoperation. From the background investigation it is clear that two very important issues are: i) the nature of the teleoperator's interface and; ii) the efficiency of the shared control between the human operator and the monitoring system. In order to investigate these two issues efficiently it was necessary to create experimental scenarios that operate independently from any application scenario, so an abstract problem domain is created. This way the monitoring system's control and interface can be evaluated in a context that presents challenges that are typical of a remote monitoring task but are not application domain specific. Therefore the validity of the proposed approach can be assessed from a generic and, therefore, more powerful and widely applicable perspective. The monitoring framework developed in this thesis is described, both in the shared control design choices based on virtual fixtures (VF) and the implementation in a 3D visualization environment. The monitoring system developed is evaluated with a usability study with user participants. The usability study aims at assessing the system's performance along with its acceptance and ease of use in a static monitoring task, accompanied by user-filled TLX questionnaires. Since future work will apply this system in real robotic welding scenarios, this thesis finally reports some preliminary work in such an application.

# Acknowledgements

I would like to thank the Norwegian Research Council for sponsoring this research, and for providing support throughout the entire PhD path. Special thanks to PPM Robotics and in particular to Trygve, for his unending enthusiasm and for his supervision and support even during hard times in my life.

To my supervisory team at BRL, Tony Pipe, Farid Dailami and Matthew Studley for providing invaluable feedback, guiding me and challenging me with questions I didn't even think of.

My progression reviewer Dr Martin J. Pearson has provided very important feedback to this research, and has helped me refining this work throughout the whole PhD period.

A sincere thank you to all the people at PPM, who have always been so nice to work with and that have made my PhD life in Trondheim so much sweeter: Martin, Balázs, Bálint, Mattias, Csongor and Swetlana. Thank you. To my friends in Bristol, in particular to Eugenio and Salua at BRL. Thank you for being such good friends even if you were seeing me so little.

Thank you to my best friend Matteo Todeschini, for all the good laughs, the talks and the psychological support during these long years, even at this distance. It is very hard for me to express how grateful and blessed I am for having a friend like you by my side. I will forever treasure every single moment, as we go on together and continue to be "bombers".

To all my friends from my university years, for the amazing time spent together back then and even now as we are all scattered across the world. Thank you Enrico Merlo, Matteo Ciocca, Leonardo Urbano, Alberto Garatti, Luca Pirotta and Dario Nava.

Finally, I want to dedicate this thesis to my family: my mother Maria Chiara, my father Stefano and my sister Sofia. Thank you for always making sure that I was happy and had the resources to pursue my wildest ambitions.

# Contents

Page

# Glossary

**ANOVA** Analysis of variance.. 108, 110

**CT** Completion Time. It is the time to complete a single task with respect to the experiments of the current work, from the start event to the end event.. 92, 107, 110, 114

**DOF** Degree of freedom. In the case of a robot manipulator it refers to the number of independent relative motions between its links. DOFs or dofs is used to indicate plural.. 40, 48, 75, 94, 95, 120

**FD-HS** FD controller high speed interface. It is a custom-made interface based on two mini PCs that can read the robot's joint values and control its position via a position control loop.. 67, 72, 78, 79, 83, 84

**NC** Number of Commands. The number of control commands that a user inputted during a single task. It is representing the number of movements issued to the monitoring robot.. 93

**NCA** Number of Corrective Actions. The number of times the system with virtual fixtures actively filters a user movement that would otherwise lead to entering a forbidden region. Corrective actions can be registered only when redirection is enabled.. 93, 103–105, 107, 112

**NPV** Number of Constraint Pseudo-violations. The number of times that the system has collided with a virtual fixture.. 93, 103–105, 107, 112

**TCP** Tool centre point: is the point used for robot positioning in any robot program that involves targets defined in the Cartesian space. The TCP is defined as a transformation from the robot flange.. 87

**TLX** Task Load Index. In this document TLX will be used to refer to the NASA-TLX.. 91, 95, 100–102, 105–110, 112

**TRL** Technology readiness level. It's a method for estimating the maturity of technologies. It was developed by NASA.. 12, 13

**VF** Virtual Fixture. 27, 50, 52, 53, 59, 60, 62, 64, 66, 68, 71, 76, 78, 80–82, 90–93, 103, 104, 106–108, 111–114, 116–118, 120

# 1 Introduction

Generally speaking, since the advent of numerically controlled (NC) machines in the 1950s, the manufacturing industry has become more and more heavily automated, with major impact on the volume of production and on the accuracy in fulfilling the specifics (Gasparetto and Scalera, 2019). Then, after the first wave of modern robotised automation was introduced into industrial processes during the middle of the 1970s, industrial robotic applications have only increased and diversified further (Gasparetto and Scalera, 2019).

Since 2010 there has been a further acceleration in demand due to the continued innovative development and improvement of industrial robots. By 2014, there was a 29% increase in robot sales across the globe (IFR, 2018). With robot sales increasing over the years, the business opportunities increased for both robot manufacturers - new companies were created - and system integrators which were often required to design robotic cells to meet the customers' needs. However, the challenges and needs for robot manufacturers and system integrators are sometimes surprisingly different. Generally speaking, robot manufacturers are interested in the overall performance of their robots, and aim at providing the capabilities to solve an increasingly larger set of industrial tasks. On the other hand, system integrator typically work with a robotic system after the basic requirements have been met, meaning that the robot is surely capable of executing a certain task given a set of requirements. What system integrators need instead are the "tools" and methods to efficiently optimize and set up the system for the customer’s application. The "tools" can be, for example, a more user friendly interface for programming a palletizing task, or turn-key solutions to perform auxiliary tasks (e.g calibration, machine vision). If the tuning and optimization of the final system is in the interest of system integrators, this can easily fall outside of the scope of robot manufacturers for marketing reasons.

More specifically, for complex robotic applications (e.g robotised welding), one of the most time-consuming phases of a project is the actual process optimization, in contrast to the path programming. Continuous process tuning can take up to 80% of the overall optimization phase of a project (Zimber *et al.*, 2016). This information is also in line with the fact that making path programming faster generally falls in line with the robot manufacturers’ intent to improve their products: path programming is less application dependent and makes robots appealing to a broader range of users.

This is usually due to discrepancies between the models and the actual behaviour of complex systems, and the system integrator needs to fine tune the final installation by trial and error to obtain the desired quality. This procedure is even more tedious when the operator cannot access the robotic system once in operation and must rely on additional sensors to acquire the necessary process information. However, it is often difficult to find a permanent placement for the sensors to be able to fully monitor the process at any given time during the trials, and this would also be a very expensive and potentially unreliable approach, if applied to all of the robot installations. While it is hard to completely replace this trial and error approach, it should be possible to provide a way to gather process information more effectively that can be used in several robotic installations.

Research into systems that improve the efficiency and overall cost of the process tuning phase can have a significant impact on the advancement of complex robotised application such as robotised welding, as well as design of the processes themselves.

## 1.1 Aim and Objectives

The aim of this thesis and the objectives that delivered that aim are described in this section. In Section 1.1.1 below, some terms that are used throughout the thesis are introduced and explained to aid the reader of this thesis, and their definition is followed by a series of research questions that the remainder of the thesis attempts to answer adequately.

However, before we come to the contents of Section 1.1.1, an effort is here made to describe as clearly as possible the underlying aim of this thesis, before the introduction of those additional definitions.

> *The aim of this thesis is to present a platform comprised of hardware equipment and software that enables a system integrator to remotely fine tune the operation of an industrial task oriented robotic system by using an additional robot used to monitor the operational behaviour of such robotic system.*

The aim of the thesis can be then delivered if the following core objectives can be achieved:

- Identify a list of core challenges in the field of teleoperation and relate them to an industrial application scenario of remote monitoring.
- Provide a method to enable remote navigation of a monitoring robotic unit that allows both teleoperation and autonomous navigation.
- Provide a solution that, by using a set of feedback modalities, assists the operator during remote control of the monitoring robot by avoiding sensory overload.
- Design a set of experiments for the evaluation of the navigation method and the feedback modalities selected.
- Provide an operational solution that addresses the challenges of telerobotics, viewpoint control and the camera placement problem.

The process of achieving those objective is also summarised in Section 1.1.2, where the most important steps during the development of this research are described.

### 1.1.1 Research Questions

The main body of this thesis, although often described with an automated welding application in mind, is general in nature and applicable to a wide range of shared-control teleoperated application domains. In fact, the next chapter investigates a set of such application domains and links them to common real world challenges. Such challenges come from unsolved issues that still have a very wide ranging significance across the field of teleoperation in general. From there, the discussion is gradually focused on two of the most important challenges arising from these unsolved issues:

- Efficient shared control between the remote human operator and the robot's semi-autonomous but locally situated AI.
- The nature of the teleoperator's interface to the remote machine that allows maximum data flow to the user whilst introducing minimal cognitive loading.

In order to investigate these two main issues effectively, it is necessary to disambiguate them from each other, and from potentially confusing other details of any one particular application scenario.

As such, it is clear that the content of this thesis was initially motivated by a set of unsolved high Technology Readiness Level (TRL) challenges that robot system integrators currently face. In order to investigate those issues effectively it was necessary to do so from the perspective of experiments devised at a significantly lower TRL. Nonetheless, at the end of the thesis, the TRL is again raised, so as to present an example of a practical demonstration of the application of the ideas investigated earlier in the thesis, thus giving the reader a sense of the potential for future industrial impact of the developed ideas.

To aid the reader of this thesis further, a distinction is made between the two "types" of robots discussed in later chapters, based on their role in the workspace. These two terms will be used frequently throughout the thesis:

- *Task Robot*: a robotic unit which is executing a, normally pre-programmed, application domain task. Throughout this thesis, robotised welding will be the most frequent example for the task carried out by the task robot.
- *Monitoring Robot*: this additional robotic unit would normally be temporarily moved to be adjacent to a task robot, and its role would be to enable a user to inspect such a task remotely, in an efficient shared control manner. The monitoring robot is then typically equipped with the sensors that are deemed necessary to accomplish the monitoring job.

The purpose of this thesis' research is then to investigate and develop a framework for the control of a monitoring robot, with the industrial robotic application domain in mind. A 3D representation of such a scenario is shown in Figure 1.1.

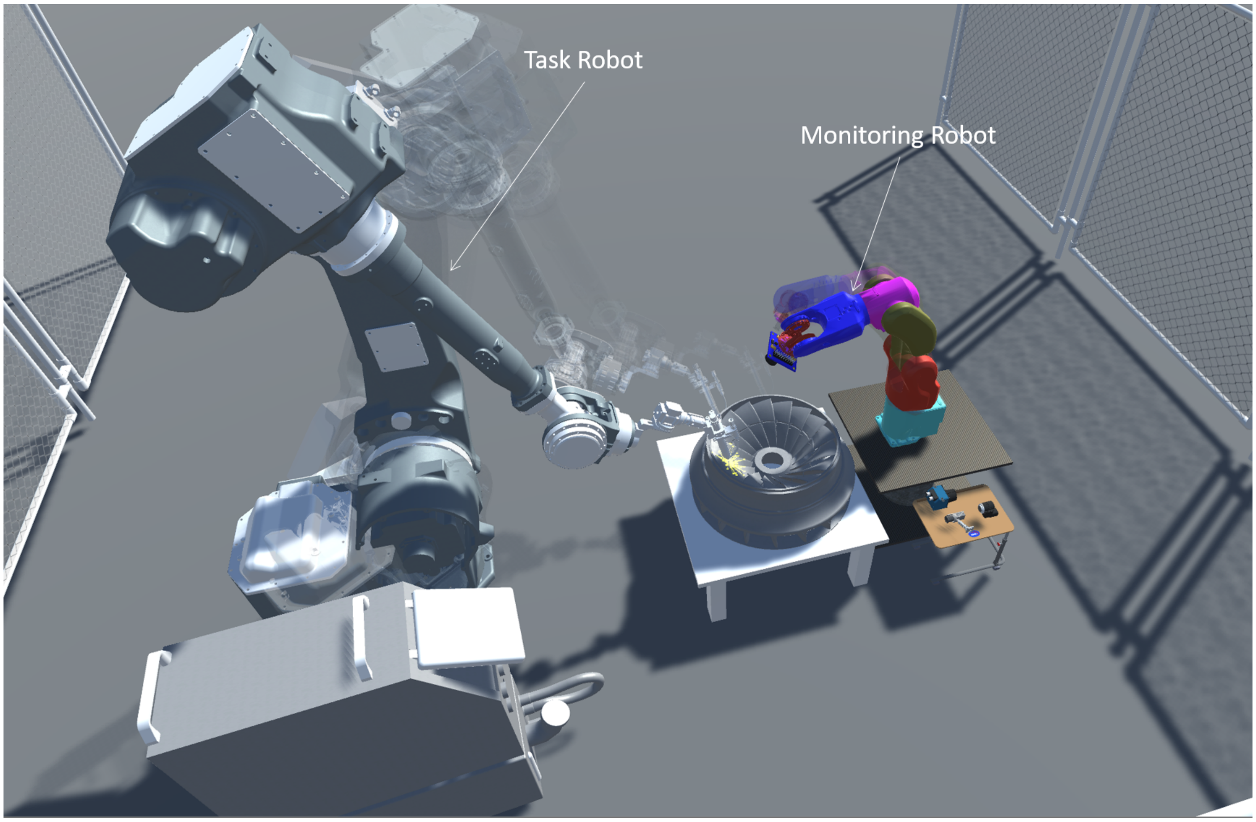


Figure 1.1: 3D representation of the monitoring robot and the task robot in a welding application. ©2020 IEEE

In the process of abstracting the challenges faced in teleoperation from the details of any one application domain, the following questions have been posed. Answering the following questions throughout the thesis will highlight the contribution of this thesis in the investigation of the two main issues previously mentioned.

I) What are the challenges related to monitoring a system with an external monitoring robot?

II) Is it possible to provide remote flexible viewpoint selection independently of the task that is being performed?

III) What are the design choices to have a seamless transition between an autonomous mode of operation for a monitoring robot and a mode that allows for user input?

### 1.1.2 Methodology

In order to answer the three questions posed in the previous section, the research can be summarised in the following steps:

1. Description of the issues for industrial robotic processes and of what could benefit the system integrators' job during the set up of a robotic cell.
2. List of the challenges related to telerobotics, viewpoint control and the camera placement problem according to what has been identified during the initial investigation.
3. Development of the first concept of a monitoring framework based on Unity3D and ROS which is capable of controlling a robot in real time for remote inspection.
4. Development of a virtual fixtures based approach for the control of the monitoring robot, which allows for both manual adjustments and autonomous motion.
5. Usability study with untrained participants with the monitoring framework on a static inspection task.

6. Selection of an extra modality to convey information about obstacles outside the field of view or singularity configuration that has been assumed by the monitoring robot.

7. Tests of the second iteration of the monitoring framework on a static inspection task.

   These steps were necessary in order to achieve the latest version of the monitoring framework. During the development, a further process of trial and error was sometimes needed in order to achieve an acceptable level of intuitiveness and usability of the system and its component.

## 1.2 Publications

Part of the work throughout the research period has been published in relevant conferences. In particular, two papers have been published in conference proceedings, another publication has been accepted for publication in the conference proceedings. The abstracts can be found on page 122.

- Edited versions of Sections 3.2.2, 3.2 have been published in:

  Sita, E., Horváth, C. M., Thomessen, T., Korondi, P., and Pipe, A. G. (2017). ROS-Unity3D based system for monitoring of an industrial robotic process. In the IEEE/SICE International Symposium on System Integration proceedings, pages 1047-1052.

- Edited versions of Sections 3.1, 3.1.1 have been published in: Sita, E., Thomessen, T., Pipe, A. G., Dailami, F., and Studley, M. (2018). Robot Companion for Industrial Process Monitoring Based on Virtual Fixtures. In the 44th Annual Conference of the IEEE Industrial Electronics Society proceedings, pages 6051-6056.
- Edited version of Chapter 4 has been published in: Sita, E., Thomessen, T., Pipe, A. G., Studley, M., and Dailami, F. (2020). Usability Study of a Robot Companion for Monitoring Industrial Processes. In the 5th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS) proceedings, pages 37-42.

## 1.3 Thesis Outline

In Chapter 2 the type of robotic applications that can commonly occur in an industrial scenario are briefly treated and then the challenges related to the optimization of certain robotic applications are discussed. More specifically, the chapter introduces the concept of an external monitoring robot for the inspection of industrial processes and then describes the fields that are closely related to the research activity that has been conducted. The specific application of robotised welding is used as the reference case for the discussion throughout both Chapter 2 and the rest of this thesis.

In Chapter 3 the system that enables control of the monitoring robot and interaction with the human operator is described. The algorithms to control the monitoring robot are developed in Unity3D (Technologies, 2017) and the framework is bridged to ROS to allow the communication with ROS-compatible machines (robots and sensors) and third-party algorithms. Moreover, the approach adopted (virtual fixtures) to allow for both autonomous and manual motion of the monitoring robot is also described. The preliminary work that has been carried out for the study of the system in an autonomous task is also described briefly and a demonstration of its potential is presented.

In Chapter 4 the monitoring system is evaluated in a problem domain that is strongly linked but critically abstracted from the details of any one application domain. In this chapter, the control of the monitoring robot based on virtual fixtures is compared to a constraint-free control mode during a monitoring task with a static workpiece (Section 4.2), with additional obstacles and eventually with additional feedback to improve navigation (Section 4.4). The behaviour of the monitoring robot and the control interface is then discussed in light of a real industrial application, robotised welding, and the additional challenges are discussed.

In Chapter 5 the main contributions of this research activity in the field of industrial robotics are discussed. The use of a monitoring robot and its control system are critically analysed in their advantages and limitations in the context of optimizing industrial processes. Finally, part of the limitations of the system are used to identify future directions for further improvements and research activities.

# 2 Background

In general, the aim of this research is to improve a remote user's perception of the distal environment in order to facilitate the monitoring/controlling task. There are two important topics that this research is spanning across: telerobotics, and multi-modal interaction. On one hand, this research looks at telerobotics, and the work within this area, to compare our approach in perception with current work, in the context of reaching a good trade off between transparency and stability - which is one of the main challenges of telerobotics. On the other hand, multi-modal interaction is a very important research field for the current work due to the effects that different input/output modalities can have on the overall performance of the telepresence communication.

These topics will be broken down into sub-themes that provide the most relevant context with which to compare the contribution of this research.

## 2.1 Telerobotics

### 2.1.1 Mobile and Non-mobile

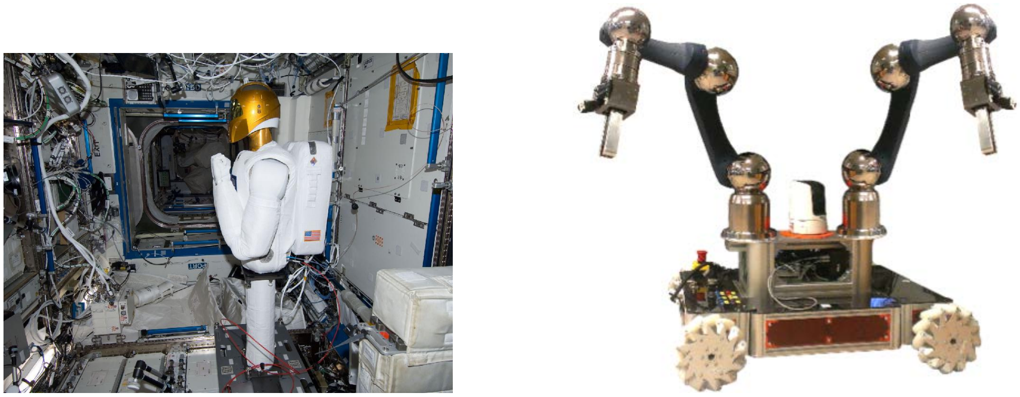
The first distinction that should be made is between mobile and non-mobile telerobotics. The first assumes that the remote robotic system is capable of large-scale movements in the remote environment. Therefore mobile telerobotics deals with sometimes slightly different challenges than non-mobile telerobotics, like the navigation control and interface, or the perception of the surroundings. In space exploration for example, teleoperated robots and autonomous robots can cooperate with astronauts to facilitate their tasks and to perform auxiliary functions during space missions (Landis, 2004; Fong *et al.*, 2012). Schmidt, Landis, and Oleson (2012) discuss a project where mobile telerobotics can decrease or entirely remove the risk of user-related elements in space missions, while at the same time offering benefits on the decision making process and perception of the remote environment. Navigation for mobile telereobotics is naturally a very important challenge. In particular, researchers are focusing on ways to pass task-related information to the operator in order to improve the navigation. Stone *et al.* (2009) presented a study in which an augmented reality interface could aid the user in a land-based navigation task. A completely different perspective in the navigation problem was taken by Salmeròn-Garcìa *et al.* (2015) where they proposed a cloud-based computational architecture to reduce the amount of processing that the robotic system has to perform in order to improve the communication with the operator. In their work instead, Saltaren *et al.* (2007) proposed a new underwater parallel robot (UPR) and the challenges related to its control to perform complex underwater tasks (Figure 2.1 and 2.2).

In mobile telerobotics, rescue robotics is also a very active research field. Birk,

(a) ©2007 IEEE (Saltaren *et al.*, 2007)  (b) ©2012 IEEE (Fong *et al.*, 2012)

Figure 2.1: Examples of mobile telerobotics for remote inspection and manipulaiton tasks: (a) REMO robot, (b) K10 Rover.

Schwertfeger, and Pathak (2009) argue that teleoperation of safe, security rescue robotics (SSRR) needs to move beyond simple mapping between user-inputs and robots' motor actuators. For such systems the control needs to reach the behaviour and mission-level, so that the user would be capable of inputting high-level commands and supervise the whole network of sub-systems and agents. In rescue robotics navigation is understandably only a sub-challenge, while many other researchers are focusing on the human-robot interactions and interfaces to execute the complex task once a target location has been reached and they won't be mentioned here. Kristoffersson, Coradeschi, and Loutfi (2013) provide on the other hand a review of the applications and approaches in mobile telerobotics and telepresence.



(a) Robonaut 2, (Fong *et al.*, 2012)
©2012 IEEE

(b) CERNbot mobile unit, (Di Castro, Ferre, and Masi, 2018)
©2018 IEEE

Figure 2.2: Examples of mobile telerobotics for remote inspection and manipulaiton tasks.

What will be treated more in depth however is *non-mobile* telerobotics, where relatively small-scale movement of an end-effector and the links connecting that end-effector to a fixed base is all that takes place; here problems concomitant with self-localisation and navigation do not occur. It is important to notice how the challenges in this area are relevant also for mobile telerobotics, but for the purpose of this research the focus is in the control strategies and approaches for the control of a robotic manipulator, on a stationary base. According to the presence or absence of time delays in the communication and the task to accomplish, many different control schemes and solutions have been explored in order to achieve a suitable trade-off between transparency and stability. The control strategies and their differences will be analysed in more detail in the following paragraph. The second part of this paragraph will highlight the different application domains which benefit from telerobotics solutions. According to the task's details, the particular application takes a more appropriate name such as *tele-manipulation* - when the task is to manipulate objects in the environment (Mitra and Niemeyer, 2008; Xia *et al.*, 2012; Niemeyer and Slotine, 2004) - or in the healthcare sector *telesurgery* (Angelini and Papaspyropoulos, 2000; Hagn *et al.*, 2010), where the tasks consist of a surgical operation performed with a telerobotic system. Regardless of the particular application, these research areas all share the fact that an operator is in total or partial control of the remote system and can send control commands through a user interface. Moreover, they all face the challenge of communicating effectively information to the operator, and the challenge of adopting a suitable set of control inputs and control strategy that allows for low-error rate operations (and possibly also short execution time). The next section will give an overview of relevant approaches to the problem of finding a suitable control strategy, and it will try to highlight the particular assumptions and constraints that characterise them.

### 2.1.2 Control Modalities

Although by definition in telerobotics the controlled system and the operator are physically apart, that doesn't necessarily mean that there will be significant delays in the communication that will interfere with the overall success-rate and execution time. For example, in telesurgery the robotic system and the surgeon's interface are in some cases in the same network, with nearly zero delays. In a telemanipulation task performed in space instead, the extreme distance between the user (on earth) and the robotic system in space is source of significant delays and flaws in the communication. It is then natural to see researchers focusing on different control strategies depending on the different constraints that need to be dealt with. The work mentioned in this literature review regards bilateral control, that is a teleoperation mode where the *slave* can reflect back to the *master* reaction or software generated forces from the task being performed. Although reflecting forces back to the operator can increase his/her perception of the environment and potentially improve performances, it may cause instability if there are delays in the communication. The trade off between a more *transparent* communication and a stable system has been the main focus of researchers in bilateral teleoperation and an historical survey has been produced by Hokayem and Spong (Hokayem and Spong, 2006). The authors highlighted how a more transparent system can provide the user a sense of the remote environment at the expense of the stability of the remote communication and is mostly considered when the system needs to interact with the remote environment. On the other hand, the authors have also indicated how force-based methods have explored the concept of passivity to ensure the stability of the communication, partly sacrificing the amount of information that can be reflected back to the user when interacting with the remote environment. The work in bilateral operation can be divided into passive and non-passive approaches, where passivity can be briefly described as a condition between control inputs and feedbacks that ensures the stability of the system. Passivity makes the whole telerobotic system stable and more robust to time delays but at the expense of transparency. In fact, passivity is only a sufficient but not necessary condition for the stability of the communication, and therefore some researchers have suggested also non-passive control strategies (Polushin, Liu, and Lung 2007; Hua and Liu 2011; Hua and Liu 2009). Passive control approaches instead can be further divided into two categories, depending whether they deal with constant or time-varying delays. As for the latter case, research contributions have been made by Chopra *et al.* (2003) and Pan, Canudas-de-Wit, and Sename (2006), Wu *et al.* (2012) and Soleimani *et al.* (2014), and Xia *et al.* (2012).

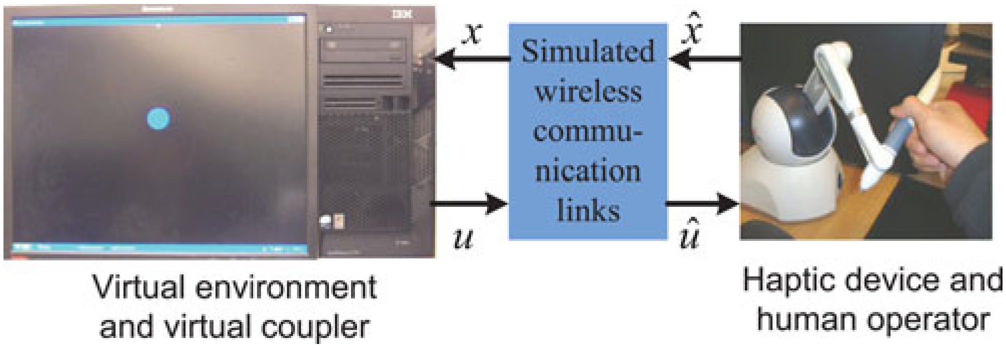


Figure 2.3: System example from (Huang, Shi, and Wu, 2012) ©2012 IEEE

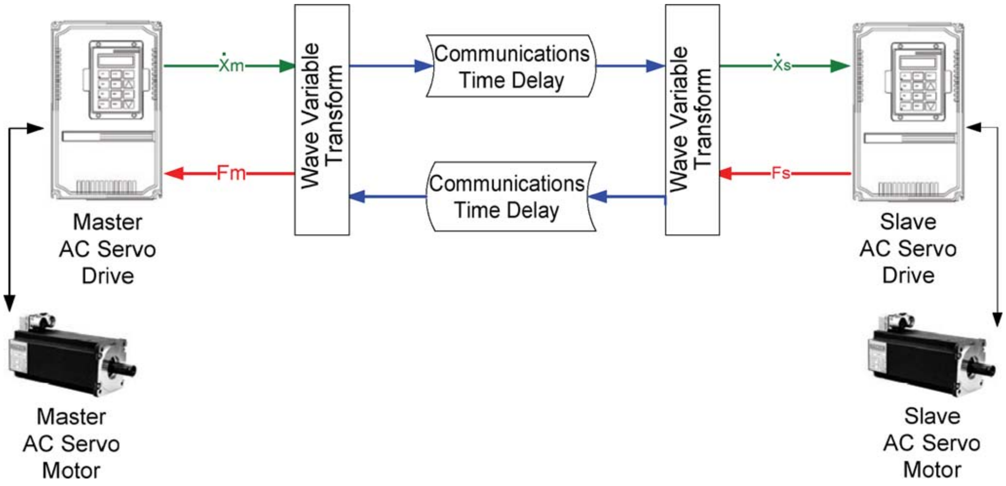


Figure 2.4: Teleoperator Layout by (Bate, Cook, and Li, 2011) ©2011 IEEE

A comparative study of different passive teleoperation control schemes with respect of time delays has been performed in (Arcara and Melchiorri, 2002) and more recent contributions have been proposed by (Bate, Cook, and Li 2011; Ye and Liu 2010; Huang, Shi, and Wu 2012) (Figure 2.3 and 2.4). The major limitation of the aforementioned contributions is the assumption that the time delays in the communication are symmetrical in nature, meaning that the data flowing from remote to local and from local to remote experience the same delay. However, in practice this assumption does not always stand due to the unpredictable instability of the communication channel, especially true in extreme conditions such as telemanipulation in space. That is why some recent contributions (Tumerdem 2017; Yang *et al.* 2017) have been focusing on bilateral teleoperation control schemes under asymmetrical time-varying delays in order to achieve stability under such unpredictable conditions. One peculiarity of the approach of Yang *et al.* (2017) is the adoption of a neural network to approximate the uncertain model of the teleoperator and the external disturbances. The passivity assumption is dropped and the stability criterion is developed taking into account non-passive human operator actions and remote environment insertion forces (Yang *et al.*, 2017).

When time delays are not a major issue in the communication instead, the trade-off transparency/stability can be changed in favour of more transparency of the system. In particular it is interesting to investigate methods that are applicable to the control of a monitoring robot, which won't exert forces on the environment. The robot instead will be able to move almost freely in the environment with the exception of some forbidden areas or trajectories, for which the master device will reflect certain reactive forces. Even though the forces replicated at the master side are not physically caused but rather software generated (the robot's end effector is not interacting with any environmental part), this kind of teleoperation control can still be considered bilateral. The monitoring robot's movements need to be biased whenever necessary, without overloading the operator, and a significantly relevant approach for that, also used in bilateral teleoperation, involves the so called *virtual fixtures* (VF) or *active constraints*. The generation of such constraints at run time is an essential element in the control of the monitoring robot that will be the subject of this research. More specifically, the constraints will serve the purpose of allowing end effector orientation adjustment (and minor position alignment) during the automatic tracking of the industrial task being monitored. The constraints will be used to convey information about the limits of the monitoring robot pose and to guide the user at realigning the view with the original trajectory (more on this topic in the next section). In bilateral teleoperation, one of the direct advantages of VF is the fact that the constraints can be changed dynamically, added or removed (inhibited) during the operation since they are software generated. Virtual fixtures can then assist the user at following a certain planned trajectory, or can prevent the tool tip from colliding with nearby surfaces. One example application in surgical telerobotics is the use of virtual fixtures to prevent the surgeon from damaging healthy tissue during the operation by imposing a dynamic active constraint on the cutting depth (Li, Ishii, and Taylor, 2007). A thorough analysis of the different techniques in the generation of active constraints is provided by Bowyer, Davies, and Rodriguez y Baena (2014) and their description of a generalized active constraint implementation is shown in Figure 2.5. In their work the authors described how more complex geometries and representations can be chosen in the definition of active constraints, however at the expense of additional computational power required to evaluate the constraint geometry during contact and partial penetration. On the other hand, complex geometries often allow the generated constraints to have greater use during a task, since they can be adapted to a specific task space configuration and not tied to a pre-defined shape.

The last part of this section will focus on some advanced techniques treated by Bowyer, Davies, and Rodriguez y Baena (2014), for the generation of virtual fixtures:

- Adaptive constraints
- Dynamic active constraints
- Multi-handed active constraints

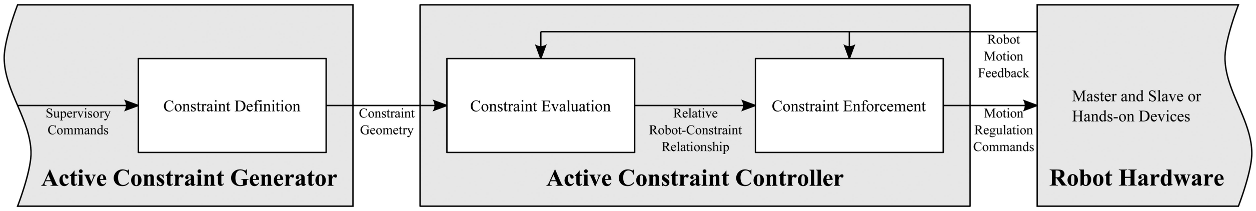


Figure 2.5: Generalized active constraint implementation as summarized by (Bowyer, Davies, and Rodriguez y Baena, 2014) ©2014 IEEE.

In the adaptive constraints techniques, the controller (software) decides when and how to apply the constraints based on some knowledge of the task, hardware or users. The way to enforce such limits is using one of the well established approaches, such as potential fields (Kuang *et al.* 2004; Chung, Lam, and Xu 2014), non-energy storing constraints (Kikuuwe, Takesue, and Fujimoto 2008; Hennekens, Constantinescu, and Steinbuch 2008), proxy and linkage simulation (Zilles and Salisbury 1995; Abbott and Okamura 2006 in Figure 2.6, Pezzementi, Okamura, and Hager 2007; Bowyer and Baena 2016) and reference direction fixtures (Bettini *et al.* 2004; Prada and Payandeh 2009) among others. In (Weede *et al.*, 2011) and (Nolin, Stemniski, and Okamura, 2003), for example, selective activation of constraints has been achieved through the use of Hidden Markov Models (HMM), which also provide a way of imposing virtual fixtures in situations which were not specifically programmed. HMMs have been used to interpret the user's control inputs and infer the current state of the system in order to apply a suitable active constraint. In (Srimathveeravalli, Gourishankar, and Kesavadas, 2007) a complex task is divided into sub-tasks with simple linear active constraints in each one. In (Yu *et al.*, 2005) the same methodology was used, but with a more complicated (curved) active constraint enforced during each sub-task to also consider obstacle avoidance. However, when many constraints are in place and the user needs to adjust the path taken due to unplanned obstacles, the robot might become immobilized. Marayong and Okamura in (Marayong and Okamura, 2004) suggest a possible approach to solving such problem. The authors researched the effects of variable admittance ratio depending on the nature of the task. The authors concluded that when many constraints and obstacles are present in the task space, automatic admittance ratio tuning is the recommended approach and developed an algorithm to select an appropriate admittance ratio based on the type of task to execute. What the authors refer to as admittance ratio is described in the current work as the admittance compliance and, as it will be described in more detail in chapter 3, has been designed to dynamically change during the execution of the remote task in order to better assist the user during teleoperation.

Another use of adaptive constraints has been proposed in (Li and Okamura, 2003) and (Kragic *et al.*, 2005) in order to understand when the user was intentionally leaving the path suggested by the constraints and therefore the software needed to inhibit the correction to allow smooth free movements. Passenberg *et al.* (2011) suggested an approach to the same problem by measuring the interactive forces between user and robot.

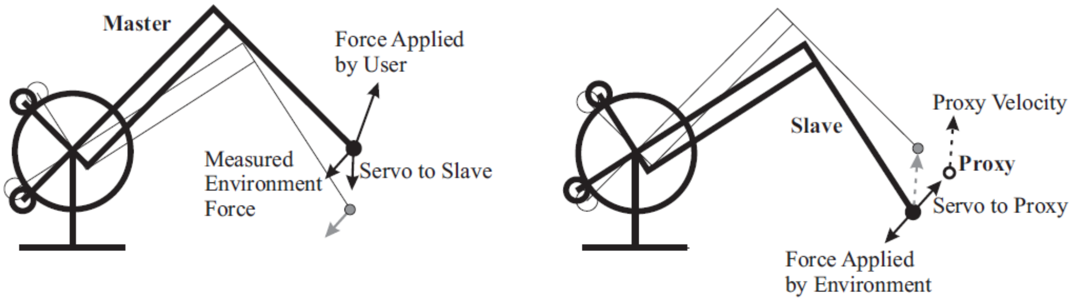


Figure 2.6: Pseudo-admittance manipulator scheme, by (Abbott and Okamura, 2006) ©2006 IEEE

Dynamic active constraints instead are those where the constraint geometry moves continuously due to changes in the environment (unstructured) or due to the task at hand (Bowyer, Davies, and Rodriguez y Baena, 2014). Most of the research has been focusing on dynamic active constraints in telesurgery. For example Navkar et al. used such technique based on a proximity function to constrain a robot end effector to follow a human beating heart (Navkar *et al.*, 2012) and showed that it brings significant improvements by lowering the off-path errors during the operation. In (Ryden and Chizeck, 2012) the authors proposed a method to constrain the robot to adapt to the heart's external surface, with the flexibility to adapt to changes in the heart's motion thanks to real-time acquisition of the heart's model through a range camera. In (Bowyer and Rodriguez y Baena, 2013) instead the authors investigated the problem of active tool motion when a dynamic constraint passed past a static tool, causing it to switch from unconstrained to constrained region.

Finally, multi-handed active constraints involve the use of multiple manipulators and deal with the problem of imposing constraints on the relative positions of the robots while keeping the individual environmental and physical constraints (Kapoor and Taylor, 2008). In (Xia *et al.*, 2011) the authors showed that by removing the user guidance from one of two constrained robots, it interestingly became autonomous and reached the constrained optimum point relative to the other.

Now that a quick description of these three approaches has been given, it is important to underline why such methods have been deemed to be particularly relevant to this research. The characteristics of the control of the monitoring robot will have to cope with a desired free camera movement (the user can choose the most preferred point of view) while still imposing constraints in the motion around the robot performing the task to monitor and the work piece. More particularly, this research can be related to the previously mentioned approaches for the following properties that the monitoring strategy needs to possess:

1. Constrained movement around the task robot, and the task path. Certain directions will be non-compliant taking also collision avoidance into account. This element relates to the research problem of multiple active constraints in the task space. The monitoring strategy developed in the current work will take into account the possibility of having multiple obstacles influencing the motion of the monitoring robot at the same time. As it was predicted by Marayong and Okamura (2004) as the advised approach to such a problem, a dynamic admittance ratio technique has been developed in order to ensure the smooth behaviour of the monitoring robot when multiple constraints are enforced.

2. Constraints enforced also when the automatic tracking of the task is active. As the monitoring robot moves in the workspace, the active constraints need to take into account its position relatively to the task robot. This property of the monitoring system is related the choice of having a dynamic active

constraint approach, as described in (Bowyer and Rodriguez y Baena, 2013). For the monitoring system developed in this thesis, the dynamic active constraints have been implemented for an admittance type of control with virtual fixtures, to achieve stability and better positional accuracy during free motion compared to impedance control approaches.

3. Fully compliant movements when departing from restricted areas, independently of the tracking.Such a property becomes necessary whenever partial penetration of an active constraint occurs and the system needs to ensure the safety of the operation and allow the user to move the monitoring robot and exit the collision state. This challenge is addressed in the current work by choosing a passive approach for the monitoring system, that is achieved through the admittance control approach and by considering the direction of motion during a collision state (directions that *exit* a virtual fixture will be more compliant than the ones that increase the partial penetration).

## 2.2 Sensory Substitution

It is very important to observe how the control mechanism and control strategy are influenced by the different feedback that are used in the communication (for example force feedback or cutaneous feedback). Although visual feedbacks are always present in all applications mentioned, solutions vary in the way they support visual stimuli to improve the user's performance (which may be measured as the completion time, number of corrections, number of attempts etc.). As stated above, one of the goals in telerobotics, is to achieve the optimal trade-off between transparency and stability. Transparency refers to how accurate it is possible to reproduce the sensory information collected at the remote site, like for example the forces exerted by the end effector onto objects, their textures, the audio of the environment and so on. Stability refers instead to the communication with the remote system: since the user needs to process such information and act accordingly, delays and other factors can worsen his/her performance and potentially destabilise the communication. To avoid such instability one could simply reduce the amount of information that is replicated at the user's site, at the cost of reducing the overall system's transparency. The degree with which a control scheme applies restrictions on the feedbacks modalities in order to preserve stability is referred to as *conservativeness*. Focusing now on the feedback provided during the communication, it has been shown that sensory substitution can bring benefits to telerobotics applications. Although it is not possible to mention all relevant papers in the area, it is provided here context with some contributions that belong to different fields and explore different feedback modalities. In general it has to be noted that the main tool to obtain a transparent communication between the operator and the telerobotic system is by exchanging both forces and positions between master and slave (Hashtrudi-Zaad and Salcudean, 2002). In general, this aspect explains why a big part of the research in sensory substitution is focused on haptic feedback (Visell 2009; Westebring van der Putten *et al.* 2008; Massimino and Sheridan 1994; Moody, Baber, and Arvanitis 2002; Wagner, Stylopoulos, and Howe 2002). The kinesthetic part of the haptic interaction can bring instability in the communication. Therefore in applications where safety is imperative like telesurgery, some researchers have concentrated only on the cutaneous component of the haptic feedback (see Figure 2.7) (Meli, Pacchierotti, and Prattichizzo 2014; Prattichizzo, Pacchierotti, and Rosati 2012; Prattichizzo *et al.* 2010).

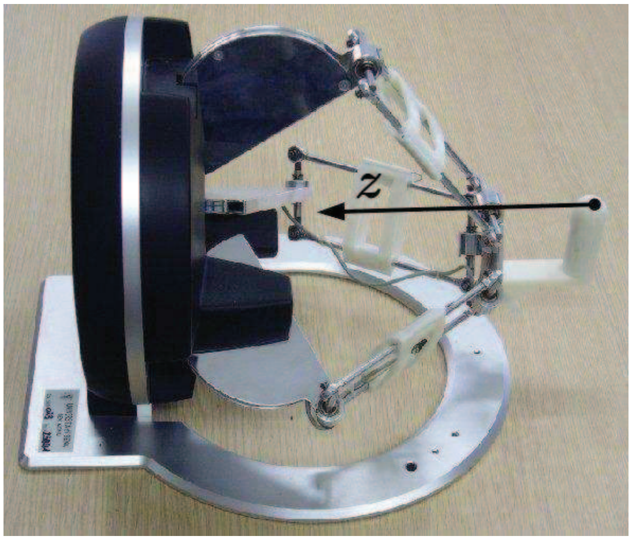
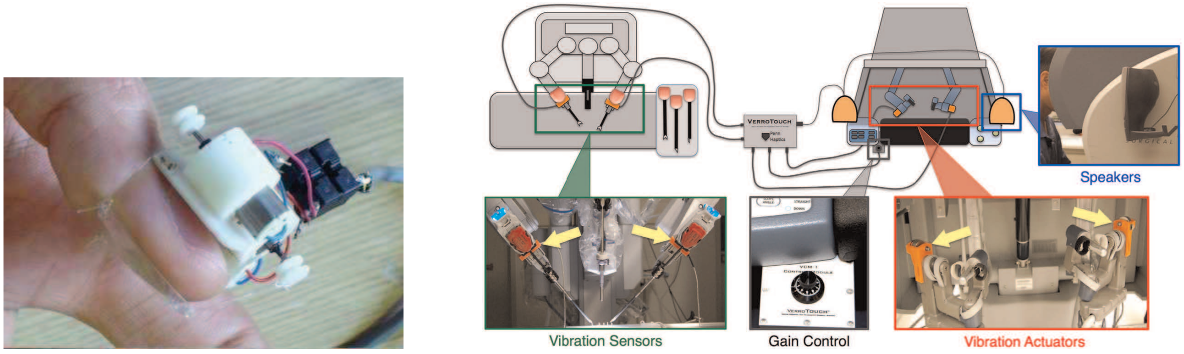


Figure 2.7: (Prattichizzo, Pacchierotti, and Rosati, 2012) ©2012 IEEE

Quek *et al.* (2015) suggested how fingerpad skin deformation could be a promising form of tactile feedback to convey force and torque information in teleoperation systems such as robot-assisted surgery, where force feedback may be undesirable due to stability and safety concerns. Other works have proposed an alternative to kinaesthetic in the form of vibrotactile feedbacks (Schoonmaker and Cao, 2006), and audio/visual stimuli (Kitagawa *et al.* 2005, McMahan *et al.*

2011) (Figure 2.8). In the same sub-area of audio-visual sensory substitution, researchers have explored such stimuli to enhance navigation during teleoperation (Liu and Wang, 2012). Benz and Nitsch recently presented in (Benz and Nitsch, 2017) a comparison of the effects of auditory and haptic feedback compared to visual feedback on target localization accuracy. This contribution is relevant because it highlights how multimodal stimuli don't always improve the information processing done by the operator. Although this effect is not new in telerobotics (the cognitive load of the operator has been investigated before for example in (Prewett *et al.*, 2010) and (Weber *et al.*, 2013)), the authors presented a systematic evaluation with particular attention to the loss of accuracy that too many sensory cues can cause. In their work they observed how the visual sensory modality is oftentimes saturated with information (Yanco, Drury, and Scholtz, 2004).



(a) (Prattichizzo, Pacchierotti, and Rosati, 2012) ©2012 IEEE

(b) (McMahan *et al.*, 2011) ©2011 IEEE

Figure 2.8: Alternatives to kinaesthetic force feedback: vibrotactile feedback in (a) and combination of audio and visual stimuli in (b).

Consequently, the auditory and haptic information channels may be suitable alternatives for the problem of direction information. Eventually they discovered that combining modalities does not necessarily lead to a superior accuracy. One interesting finding is that adding auditory to haptic feedback decreases performance compared to unimodal haptic feedback (for direction information).

All the approaches and contributions mentioned above can be seen as attempting to solve a multi-objective goal comprised of the following:

1 - Increase the transparency of the system: more information during the communication can turn into increased operator's performance

2 - Increase the stability of the system and its robustness to communication delays and sensor failures

3 - Decrease the cognitive load of the operator, which translates into lower error-rate and more stable performance during prolonged teleoperation sessions

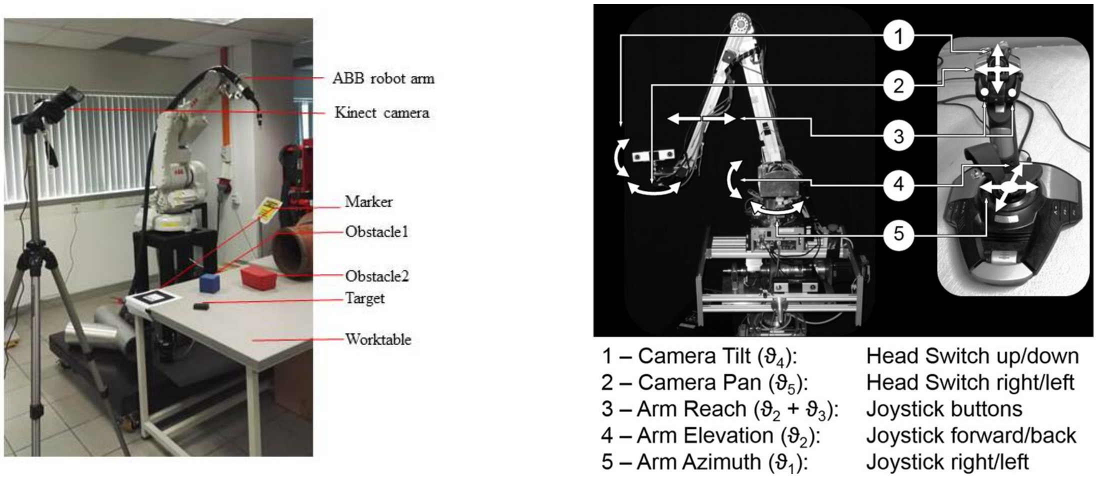
One way to involve sensory substitution approaches in the monitoring solution of this research is in the way spatial information is communicated to the user without forcing him/her to change the viewpoint of the monitoring robot. With tactile feedback it is possible to communicate additional information on the pose of the monitoring robot and on the possible movement in a particular moment. As mentioned in the previous subsection different virtual fixtures will be used to achieve a stable behaviour of the monitoring robot, and without additional feedback it would be hard or impossible for the user to pinpoint the reason why the movements are constrained in a particular situation. Understanding the nature of the active constraint that is being enforced is important because the user could then adjust the control movements to avoid unwanted reactions from the monitoring robot (e.g. if the robot needs to change the viewpoint automatically to avoid an obstacle without providing feedback to the user).

A small observation to point (1.) in the list is that the notion of performance can be evaluated with different metrics depending on the aspect that is investigated in the experiments. However, one can adopt (Steinfeld *et al.*, 2006) definition: the operator's performance in the navigation task could be rated across three dimensions, namely effectiveness, efficiency, and effort. *Effectiveness* refers to how "well" the task is completed, and can be represented as a percentage of tasks successfully completed. *Efficiency* instead refers to the time and/or resources needed to complete the task. Finally *effort*, regards the workload of the operator during the task, and could be represented as the amount of errors/mistakes made during the operation. The next section delves deeper into the visual sensory modality to relate the challenge of conveying more information from the telerobotic system to the problem of the operator's limited field of view and constrained viewpoint.

## 2.3 Vision and Camera Viewpoint Control

It is clear that stability is necessary in order to successfully perform a certain task during remote communication. On the other hand however, it also important to have an accurate representation of the state of the remote site in order to efficiently accomplish our task. In telerobotics, most of the information is conveyed through the visual channel. Visual perception can be seen as the main contributor to the overall understanding of the remote site as well as inefficiencies whenever visual stimuli are insufficient. As it was mentioned earlier, many approaches have explored how to augment such visual stimuli to increase the user's perception by exploiting multi-modal stimuli. Other contributions instead have investigated how to improve the vision systems control in telerobotics applications. (Ni *et al.*, 2016) (Figure 2.9) have exploited additional visual stimuli, overlaying information on top a camera feed. In (Kamezaki *et al.*, 2014) (Figure 2.10) and (Yang *et al.*, 2015) instead the authors involved the selection of the most appropriate viewpoint among many cameras according to the active phase of the teleoperation. However, these approaches still face the problem of a limited camera viewpoint even in the case of multiple cameras positioned in the remote environment. Furthermore, in the case of adjustable viewpoint, previous solutions have mostly covered zoom pan and tilt camera movements (Zhu, Gedeon, and Taylor, 2011).

One important contribution that is aligned with the approach taken in this research is the one of Wilde, Chua, and Fleischner (2014) (Figure 2.9). In their paper the authors address the problem of limited perception during teleoperation of spacecraft docking. They adopted a robotic arm directly controlled by the operator with a camera mounted on the end effector in order to improve the operator's awareness during the task. The experiments showed how the partial error rates decreased for almost all cases, with the exception of highly complex situations due to the workload in operating both the camera view and the docking spacecraft.



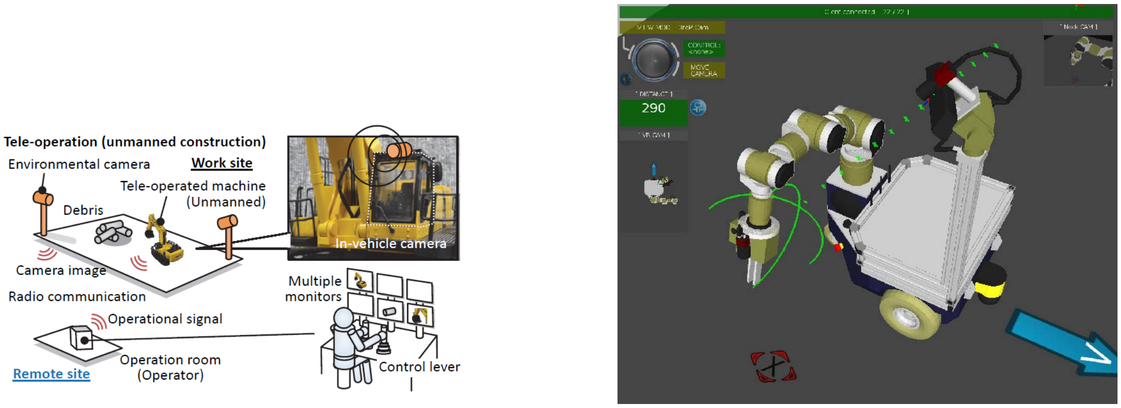
(a) (Ni *et al.*, 2016)
©2016 IEEE

(b) (Wilde, Chua, and Fleischner, 2014)
©2014 IEEE

Figure 2.9

Such system however had the limitation of not having any position/orientation support and guidance in the case of collision avoidance. Moreover, the authors observed that "more intuitive control of the camera arm, and more pre-set positions and viewing angles, would better support the operator" (Wilde, Chua, and Fleischner, 2014). In this particular context is placed the main contribution of this PhD. In fact, in order to improve the user's perception of the environment and the task during telepresence, the use of a 6 DOF external monitoring robot will be investigated, that the user can control to obtain the most appropriate camera view of the operation at hand. With a combination of automatic tracking and user-controlled adjustments, the goal is to efficiently monitor a remote industrial robotic task.

However, remote control of a telerobotic system can essentially take any shape between a completely manual approach and a fully autonomous one. The selection of the camera viewpoint is no exception. More precisely, a fully autonomous solution in this context refers to a monitoring system capable of autonomously select the optimal viewpoint *for* the user in order to facilitate the teleoperation. Researchers that have been focusing on fully autonomous monitoring systems for telerobotics operations are for example (Yang *et al.*, 2015), (Pandya *et al.*, 2014) or (Notheis, Hein, and Worn, 2014) (Figure 2.10), while Pandya et al. in (Pandya *et al.*, 2014) present a survey on viewpoint control in telerobotics and laparoscopic telesurgery.



(a) (Kamezaki *et al.*, 2014)
©2014 IEEE

(b) (Notheis, Hein, and Worn, 2014)
©2014 IEEE

Figure 2.10

In (McKee, Brooks, and Schenker, 2003) McKee et al. have proposed the so called "visual acts" approach to assist the operator by automatically identifying the best point of view during the task. However such a method is heavily dependent on the task's geometrical relationships and definition, therefore only suitable for well defined operations, which contain almost no uncertainty nor possibility of replanning. In (Pandya *et al.*, 2014), the authors discuss potential avenues towards fully autonomous vision systems for teleopreation, identifying the following research topics: improved tracking methods; algorithms for skills acquisition; knowledge representation; user intent modelling; and methods for evaluation and testing.

With an attempt of merging the discussion of Pandya et al. with other contributions in research, the four main challenges that are being faced in this research area are:

- When possible, select the optimal viewpoint for the user according to the state of task/system
- Increase the perception and understanding of the task without overloading the operator
- How to efficiently map the user control inputs when the camera viewpoint changes without increasing the operator workload
- Design a human-machine interface that allows for natural and efficient control of both camera viewpoint and main robot (which performs the task)

It is however important to highlight one particular observation about the first challenge listed above. Regardless of the probably too generic notion of "optimal viewpoint", the fully autonomous selection is currently only feasible for a certain range of situations. More particularly, one key aspect that enables such an approach is the repetitiveness of the task and the possibility to divide it into clearly defined subtasks. Moreover, automatic selection of the optimal viewpoint in research usually assumes that the user is primarily engaged in the execution of the task and does not want to have workload related to the control of the camera view. This very last assumption is what differentiates the subject of this research from the majority of the reviewed papers. The operator will be controlling a monitoring robot with the primary objective of *inspecting* the task, which will be performed by another robot. Once again, since the focus is not on the control of the industrial robot which performs the task, the monitored system can be chosen as fully autonomous if it is appropriate to do so. The choice of a fully autonomous system that accomplishes the industrial task is not a limitation since the operator's job is to inspect and monitor such process (there is no longer an optimal viewpoint/s since the user should be able to monitor any location around the workpiece and/or the industrial robot). Finally, the autonomous viewpoint selection usually requires significant information about the ongoing process, such as geometrical constraints, relationships and task objectives. Such information is assumed to be unknown by our monitoring system, which has only access to the industrial robot and workpiece 3D models, but has no access to the process data. The work of this research will verge on the second challenge presented above, with repercussions on the third and fourth challenges. The correlation with these other two challenges is due to the fact that the control of the monitored industrial robot could be then enabled from the viewpoint of the robot used for the inspection/supervision. The characteristic of the suggested approach will be described in detail in Chapter 3 along with the implications for the the control of the industrial robot and the mapping between user inputs and robot movements.

## 2.4 Remote Monitoring

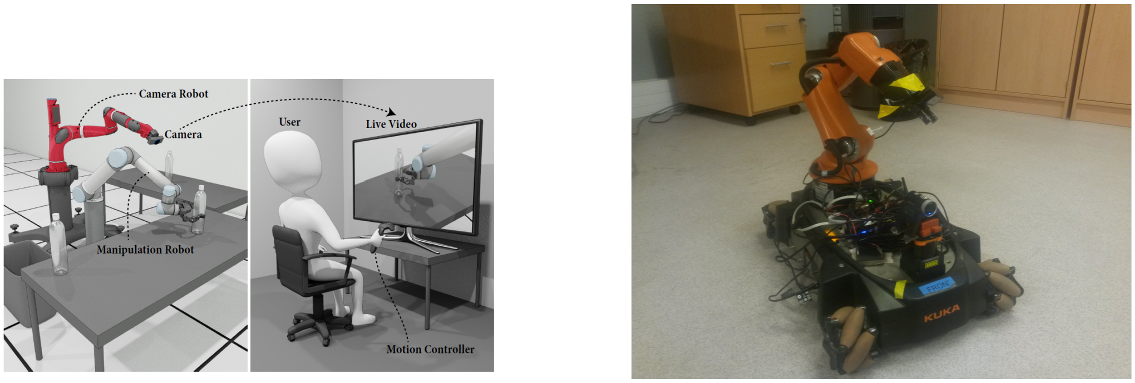
If on one hand there has been a considerable amount of research to explore the control of robot systems for teleoperation, on the other hand there has been less work related to how to provide an effective view to improve the telerobotic task. It is important to re-state the current limitations of existing approaches commonly used in telerobotics such as arrays of static cameras or a camera attached to the task robot's end-effector. When static cameras are used the task robot may occlude vision from one or multiple viewpoints at the same time, requiring the monitoring system to piece together information from the different cameras, however potentially increasing the cognitive load for the operator or the task complexity. When an end-effector camera is used instead, it is often the case that either the remote task is interfering with the camera view (e.g. in grasping or manipulation tasks) or that the camera placement does not provide sufficient context for the effective execution.

The idea of adopting a secondary robot with the function of a monitoring unit has been explored by an increasing number of papers in recent years, with some important contributions published during the time of writing of this thesis. In particular, the work from (Di Castro, Ferre, and Masi, 2018) discussed a modular robot for autonomous inspection and maintenance of hazardous industrial scenarios, to be deployed in sites such the CERN laboratory where maintenance of the extensive equipment is paramount to the experiments preparation and execution. The focus in their work is mostly on how to make the navigation autonomous and enable manipulation skills in such delicate scenarios and present the information through an adaptive graphical user interface, presented in (Lunghi., Prades., and Castro., 2016).

Another important work recently published is the paper from (Rakita, Mutlu, and Gleicher, 2018) (Figure 2.11), where the authors propose a method to assist an operator during a teleoperation task. The approach involves an external monitoring unit that is autonomously following the manipulator robot that is controlled by the user. In order to provide the appropriate viewpoint, the monitoring system has to use motion prediction and concepts from animation and graphics in order to evaluate which pose is the best as the user control the manipulator.

It is also worth mentioning a slightly more dated, but nonetheless pertinent contribution from (Bjerkeng *et al.*, 2011) where they also proposed to used a monitoring robot for an industrial task to be carried offshore.

Although work could be done to integrate subparts of the autonomous behaviours that have been presented in these works, this research focuses on the challenge of having a user in charge of the *monitoring* task, and not for example



(a) (Rakita, Mutlu, and Gleicher, 2018)

(b) (Lunghi., Prades., and Castro., 2016)

Figure 2.11

of the manipulation part as in the work of (Rakita, Mutlu, and Gleicher, 2018). The challenge brought from a user in charge of the monitoring is in the way the system handles the navigation of the monitoring robot with respect to the task robot and the surroundings. Whenever the user stops the manual operation, the system could resume its motion according to the behaviour specified by (Rakita, Mutlu, and Gleicher, 2018).

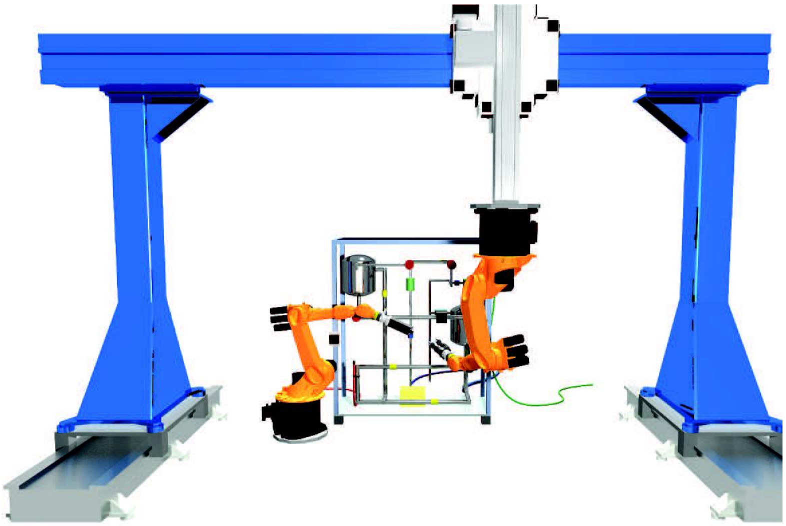


Figure 2.12: (Bjerkeng *et al.*, 2011) ©2011 IEEE

## 2.5 Virtual Fixtures

In the literature, telesurgery is one of the most prolific research fields for advancements in virtual fixtures/active constraints techniques. Virtual fixtures approaches can be generally used in any teleoperation scenario. Depending on the hardware, the slave device might implement an admittance-type or impedance-type control, influencing the way the virtual fixtures are enforced along with other implications on transparency, stiffness and precision.

### 2.5.1 Impedance and Admittance

Robots of the impedance type are typically backdrivable with low inertia, low friction and force-source actuators, as is typical of haptic devices (Abbott and Okamura, 2006).

As highlighted by (Lawrence, 1988), (Ott, Mukherjee, and Nakamura, 2010), (Ott, Mukherjee, and Nakamura, 2015) and most recently by (Cavenago, Voli, and Massari, 2018), impedance and admittance control have opposite stability and performance characteristics. Generally, admittance control can assure better performances in the interaction with soft environment and free motion thanks to its stiff feature. On the other hand, impedance control is robust to uncertainties in the model parameters and can guarantee very good performance and stability in stiff environments thanks to its soft behaviour.

In their work (Cavenago, Voli, and Massari, 2018) discuss an approach to combine the two control types together to achieve better performance in time-varying environments. Their hybrid system has been applied to a 2-DOF system in simulation, although without experimental validation. Moreover, the system has not been observed when transitioning from free to constraint motion

## 2.6 Summary

After reviewing the techniques and challenges related to remote inspection and monitoring and the control of telerobotics systems, it is clear that there is room for research and improvement. When remote monitoring is needed in industrial scenarios where humans cannot access the workspace due to its danger or lack of space, it is difficult to provide sufficient viewpoint control to the users.

Part of the challenge is the control of the *monitoring robot* in a fashion that does not interfere with the industrial task, but that at the same time allows to almost freely choose the point of view in the environment. To address the problem, research works have proposed several different approaches depending on the kind of sub-challenges they aimed to address. The current work has been focusing on an admittance control approach, based on the use of virtual fixtures. Each element possessed by the monitoring framework developed in the current work has been carefully designed to build on previous work and improve it, and to address the particular challenges related to a human-in-the-loop application scenario.

The first element summarized here, is the degree of shared control of the monitoring framework developed in the current work. Compared to the work of similar monitoring implementation like the on proposed by Bjerkeng *et al.* (2011) and the work of Rakita, Mutlu, and Gleicher (2018), the current work focuses rather on the teleoperation characteristics than the autonmous behaviour of the monitoring system. While manual navigation has still been developed together with an autonomous navigation system that can track the motion of the task robot, the focus of this thesis has been on the properties of the monitoring robot's behaviour during teleoperation: free motion of the robot whenever possible and stability of the monitoring task.

With the use of an additional robotic unit that can provide a dynamic viewpoint of the task, the user can perform the monitoring task with more awareness of the process itself thanks to not being constrained to one single point of view and without being overwhelmed with information provided by an array of cameras.

Furthermore, an admittance type of control has been developed for the monitoring framework in order to achieve better positional accuracy and stability during free motion. Compared to its dual impedance-type approach (and highlighted in their work by Cavenago, Voli, and Massari (2018)) admittance control allows the monitoring robot to have a more stable motion when not in contact with obstacles. Since the monitoring robot must not interact with the environment, admittance control is the preferred approach if stability is the utmost priority during teleoperation. Finally, in this thesis, the admittance control has been paired with a passive approach in order to ensure the safety of the system even in the case of partial penetration of virtual fixtures. In this thesis, the VF implementation builds on the work of Marayong *et al.* (2003) for the geometric approach, adopted in the current work as well, and improves its performance with the implementation of a variable admittance ratio during operation to assist the user when multiple constraints are being enforced on the monitoring robot (the reader is referred to Chapter 3 for the implementation details).

Such redirection techniques could also be employed for the autonomous behaviour for the monitoring robot as soon as the operator hands over the control of the system, and a brief demonstration of such a scenario is presented in Section 3.3 in the next chapter.

It is, however, important to re-state that the actual control of the industrial robot performing the operation that is being monitored is not within the scope of the PhD. Instead, this research will focus solely on the monitoring robot, with the additional simplification of having a fully automated industrial task. In this way, the contribution of this research can be well identified as the control of a monitoring robot independently from the task robot.

# 3 System Description

In this chapter the software choices are discussed in detail, along with a description of the functioning of the monitoring software and its building blocks. In this chapter the focus is on the local virtual fixture paradigm and its application for a monitoring robot rather than on the user interface component. Since the monitoring system has a human-centred design, part of this chapter is dedicated to the features that have been developed to assist the user navigation, and also to assist the operator in the process of viewpoint selection (for a simplified diagram of the monitoring system that will be described in this chapter, see Figure 3.1).

## 3.1 Admittance Virtual Fixtures

For system integrators, optimizing complex industrial robotic applications (e.g. robotised welding) is a difficult and time-consuming task. This is usually due to discrepancies between the models and the actual behaviour of complex systems, and the system integrator needs to fine tune the final installation by trial and error to obtain the desired quality. This procedure is even more tedious when the operator cannot access the robotic system once in operation and must rely on additional sensors to acquire the necessary process information. However, it is often difficult

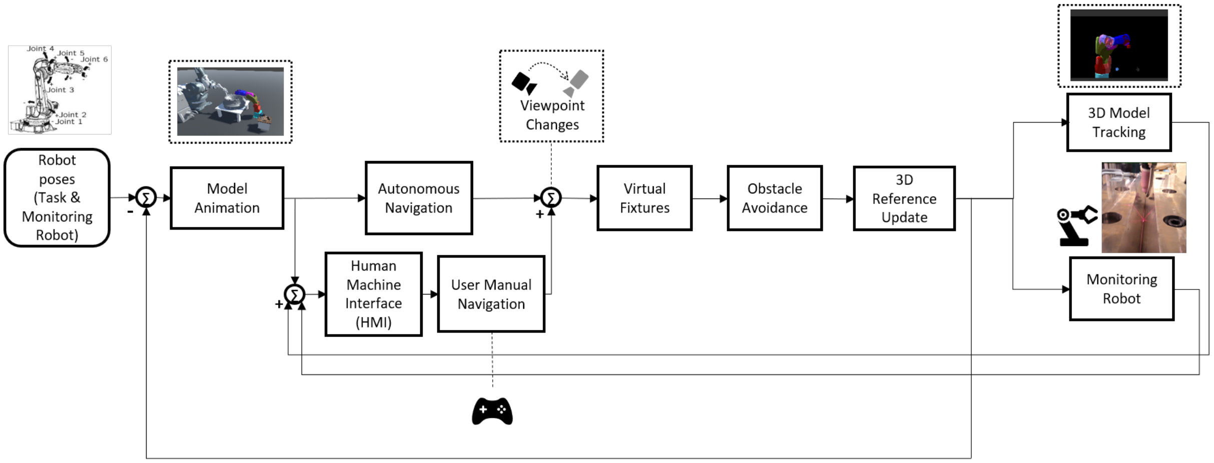


Figure 3.1: The simplified control diagram of the monitoring software. Autonomous navigation (or motion) input are in parallel with the user manual navigation. ©2018 IEEE

to find a permanent placement for the sensors to be able to fully monitor the process at any given time during the trials, and this would also be a very expensive and potentially unreliable approach, if applied to all robot installations. While it is hard to completely remove this trial and error fashion, it is possible to provide a way to gather process information more effectively that can be used in several robotic installations.

Virtual fixtures (VF) (also called active constraints) are a concept introduced by Rosenberg, 1992 as a way to anisotropically influence robot movements. Active constraints are a very important concept for many telesurgery applications, and have been thoroughly surveyed in this light by Bowyer, Davies, and Rodriguez y Baena, 2014. For conciseness, the term virtual fixtures will be used instead of active constraints from now on. The current work adopts the geometric approach discussed by Marayong *et al.*, 2003: the virtual fixtures are defined in the task space as geometric entities (e.g. spheres or planes) that impose constraints on the monitoring robot's motion when in contact with it. The main advantage for which the geometric approach has been chosen, is the visualization of the active constraints. Since the VF are defined as geometrical shapes, it is trivial to visualize them in a 3D representation of the remote workspace with a corresponding virtual object. Moreover, it is also easier for the user to understand what restrictions they impose on the monitoring robot's motion compared to an approach that would define the constraints in the joint space. However, it is important to observe that it is more difficult to define effective geometrical shapes and therefore constraints to assist the user when the monitoring robot is close to a singular configuration.

In this thesis the virtual fixtures are therefore represented by a set of preferred and non-preferred directions of motion which can be designed to be an abstract surface that the robot's end-effector cannot penetrate. The fact that some directions are identified as non-preferred means that the end-effector motion will be less compliant along such directions, as if the end-effector were experiencing some resistance. It is important to notice that the use of virtual fixtures is independent of the type of control scheme used on the robot, which could be either admittance or impedance control. With a very brief description, it could be said that impedance control imposes a force on the robot (spring-mass-damper behaviour), while admittance control imposes a position. With impedance control, forces can be applied to the robot in response to its interaction with the environment. In admittance control, the robot motion is purely decided by the control software and the robot tends to be stiffer when it gets in contact with external objects. In particular contexts, admittance control can be seen as a "safer" approach compared to impedance control due the absence of motion if the user input drops to zero. However, admittance and impedance control could be interchanged in many applications. The type that has been chosen here is admittance control, meaning that the control software will filter the user input and apply the filtered motion to the master reference (which in this case is a 3D model of the monitoring robot).

The robot is modelled as a purely kinematic body with the position of the end effector as $x \in SE(3)$ and its velocity defined as a vector $v = \dot{x} \in \mathbb{R}^6$. The definition that will be used in this formulation has been discussed in detail in Marayong *et al.*, 2003, and here only the most important concepts will be discussed. By looking at the relation:

$$\boldsymbol{v} = c\mathbf{f} \tag{3.1}$$

the velocity of the end effector is linked to the input force via an admittance control law, with all vectors referred with respect to the robot base. In Equation 3.1, $c$ is called the *compliance* factor. If $c$ is a scalar, then the system is equally compliant in all directions of movement.

The value of the compliance factor traditionally goes from zero to one, with one meaning fully compliant movements and zero non-compliant. If the compliance is instead a diagonal matrix $C$ of order $n$, where $n$ is the number of degrees of freedom of the robot, it is possible to select which direction should be more or less compliant (for example, if only the first two elements of the diagonal of $C$ are non-zero, namely $c_{11}$ and $c_{22}$, the system will only move along the $xy$ plane). Depending on the needs, certain directions of movement my be preferred to others, and then being assigned a closer-to-one value compared to other directions. This kind of anisotropic compliance has been termed *guidance virtual fixture*. As observed in Marayong *et al.*, 2003, the virtual fixtures approach could be formulated entirely in terms of compliances, but it is also possible to have a geometric approach and identify at each moment the preferred and non-preferred directions of motion.

Let $D$ be a $6 \times n$ matrix $D = D(t)$, $1 < n < 6$ containing the preferred directions of motion. The dimension of the $D$ matrix is determined by the type of constraint imposed on the end effector. For example, if $n$ is 1, the preferred motion is along a curve in $SE(3)$, or along a surface if $n$ is 2 and so on. As described by Marayong *et al.*, 2003 and in Hager, 2002 the input vector can then be decomposed along preferred and non-preferred directions with the Kernel and Span operators:

$$v_D \equiv [D]\boldsymbol{v}_{in} \quad \text{and} \quad \boldsymbol{v}_\tau \equiv \langle D \rangle \boldsymbol{v}_{in} \tag{3.2}$$

Since it is possible to write $v_D + v_\tau = \boldsymbol{v}$ due to the properties of the Kernel and Span operator (for more details consult the appendix), the end effector velocity can be rewritten as:

$$\boldsymbol{v} = c([D] + c_\tau \langle D \rangle)\boldsymbol{v}_{in} \tag{3.3}$$

where $c_\tau \in [0, 1]$ is the compliance factor for the non-preferred directions. The smaller the value of $c_\tau$, the smaller the compliance along the non-preferred directions of motion. If $c_\tau$ is chosen equal to 0, it provides a *hard* virtual fixture, as opposed to any other value which instead would still permit motion along the non preferred directions.

Until now the virtual fixture approach has been described in the context of pseudo-admittance control (there are no real input forces). Local virtual fixtures are introduced as a way to redefine the mapping between user input and end-effector motion. After that, the concept of *varying compliance* is described as a way to influence the end-effector motion when approaching certain forbidden regions.

In the chosen scenario, the robot motion will be dictated by inputs coming from two different sources: the user and the autonomous navigation. The latter refers to motions that are decided by a software and which do not have anything to do with the user commands. All motion commands will be summed to create a resultant direction for the robot's end-effector.

This is a very important preliminary step, since it changes the original formulation in Equation (3.3). The velocity of the end effector $v$ is made of the sum of two different vectors, one is the user velocity and another which is the velocity calculated by the autonomous navigation part, described in Section 3.2.1:

$$\boldsymbol{v} = \boldsymbol{v}_u + c([D] + c_\tau \langle D \rangle)\boldsymbol{v}_a \quad \text{and} \quad \boldsymbol{v}_u = c([D] + c_\tau \langle D \rangle)\boldsymbol{v}_{in} \tag{3.4}$$

where $\boldsymbol{v}_a$ is the velocity vector of the autonomous navigation part. In Equation (3.4) $\boldsymbol{v}_u$ is the user velocity vector that in Equation (3.3) was simply called $\boldsymbol{v}$.

The user must be able to move the monitoring robot inside the workspace, so that he/she can inspect the workpiece from different viewpoints. The monitoring robot's main purpose is therefore to inspect the workspace and the workpiece, and the mapping between user input and end-effector's motion can be designed to better reflect this functionality. Let $D$ define a local virtual fixture which constrains the end effector on its surface. The user input will navigate the end-effector along such surface while autonomously maintaining the orientation toward the centre of such virtual fixture. Although there is no requirement for the virtual fixture's shape, from now on it will be considered a *spherical* surface (virtual sphere) for the explanation as that is what has been used also in the actual implementation.

If the end-effector is constrained on the surface of a virtual sphere, it will always be oriented towards the centre of such a sphere. Therefore, placing the sphere on the workpiece makes it target of the inspection. The user navigation commands will move the end-effector along the surface. Such a virtual fixture is referred to as *local* because it is defined with respect of a specific point in the workspace which serves as the centre of the sphere. Moreover, since the fixture is locally defined, if the point in the workspace to which the virtual sphere is anchored moves, the whole virtual fixture moves as well. If the end-effector is constrained to the surface of the virtual sphere while this is moving in the workspace, the end-effector will experience the same relative movement as it is constrained to remain on the same point on the sphere's surface.

The local virtual fixture can be expressed in terms of preferred directions, by defining the matrix $D(t)$ which contains the preferred directions of motion. For a motion along a surface in $SE(3)$ the $D$ matrix must be defined as a $6 \times 2$ matrix:

$$\mathbf{D} = \begin{bmatrix} t_{x_1} & t_{y_1} & t_{z_1} & 1 & 0 & 1 \\ t_{x_2} & t_{y_2} & t_{z_2} & 1 & 0 & 1 \end{bmatrix}^{\mathbf{T}} \tag{3.5}$$

where $(t_{x_1}, t_{y_1}, t_{z_1})$ and $(t_{x_2}, t_{y_2}, t_{z_2})$ are the vectors that identify the tangent plane to the sphere's surface at any given time. The tangent vectors will change as the end-effector moves along the sphere's surface, making then the $D$ matrix time dependent as expected. The elements of $D$ equal to one indicate the freedom to rotate along the $Z$ and $X$ axis in order to maintain the focus toward the centre of the virtual fixture.

According to the previously given definition, the $D$ matrix contains the preferred directions of motion (that describe a plane) and the end-effector can be imagined to move on the "instantaneous" tangent plane at every iteration. It is not hard to observe however that with the current formulation, the end-effector will drift away from the centre of the sphere, as it will move on a parallel tangent plane at every iteration while it accumulates the linearisation error. That is why an *attraction* motion at every iteration could be applied to ensure that the distance from the sphere's centre is respected. This approach has been used earlier in this research when the end effector was in contact with the forbidden region, and when such a local virtual fixture was set to be non-compliant, meaning that the $c_\tau$ factor was always 0 for the fixture. If the attraction motion is used, a *hard* virtual fixture can also be seen as a way to redefine the input mapping. The manual motion of the end-effector is dependent on the local virtual fixture. More specifically, if the user is given the ability to move the end-effector "forward", "backward", "up", "down", "left" and "right", these navigation commands can be assigned to different *actual* motions depending on the type of virtual fixture chosen once the TCP is in contact with a virtual fixture.

However, using an external attraction force to remain on the surface of the VF introduces energy into the system, and also results in motion - apart from the autonomous navigation motion - which the user has no control over.

In the current implementation of the monitoring system the attraction component has been removed, and the $D$ matrix is generated when the monitoring robot TCP enters in contact with a VF. The vectors $t_1$ and $t_2$ used to generate $D, \langle D \rangle$ and $[D]$ are two tangents of the VF surface at the point of contact with the TCP and the

VF itself. Figure 3.2 shows an example of the construction of the tangent plane given the contact point between TCP and virtual fixture.

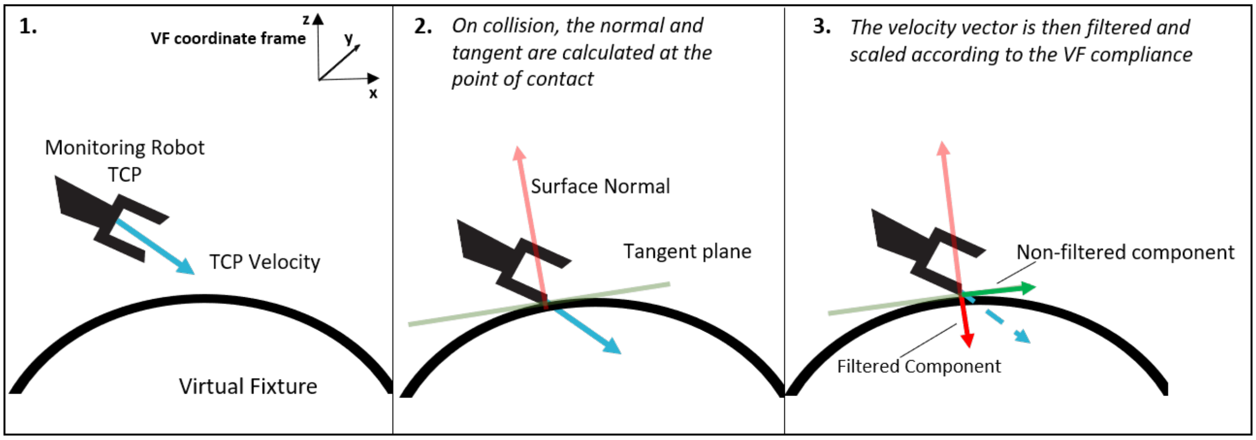


Figure 3.2: Construction of the D matrix and the tangent plane during manual motion. The *D* matrix is only generated when the TCP is in contact with the VF. The reference frame shown in the picture is the VF coordinate frame, where the *xy*-plane is tangent to the VF surface. ©2020 IEEE

The normal is used to find two tangent vectors, which are then used for the *D* matrix. Although the compliance for each VF can be changed at runtime (see also Section 3.1.1), the *default* behaviour of the virtual fixtures is with a non-zero $c_\tau$ factor. Therefore, the user can penetrate the virtual constraints during manual motion although with more resistance ($c_\tau << 1$). In Equation 3.2 it can be noticed that if the TCP is inside a virtual fixture and the user wants to exit the constraint, the velocity vector will still be scaled down by $c_\tau$ making it harder for the user to exit the violation for example. An additional term is then added to Equation 3.3 that allows the user to easily exit from the VF when the motion is not *worsening* constraint violation. Equation 3.3 is then reformulated with the additional term:

$$\boldsymbol{v} = c([D] + \lambda + c_\tau \langle D \rangle)\boldsymbol{v}_{in} \quad \text{with} \quad \lambda = \langle D \rangle \hat{v}_n \tag{3.6}$$

The term $\hat{v}_n$ is equal to zero when the dot product between $\langle D \rangle \boldsymbol{v}_{in}$ and $\hat{n}_c$ is negative, where $\hat{n}_c$ is the normal of the virtual fixture at the penetration or collision point. If the dot product is positive instead, $\hat{v}_n$ equals to one :

$$\hat{v}_n = \begin{cases} 0 & \langle D \rangle \boldsymbol{v}_{in} \bullet \hat{n}_c \leq 0 \\ 1 & \langle D \rangle \boldsymbol{v}_{in} \bullet \hat{n}_c > 0 \end{cases} \tag{3.7}$$

If the virtual fixtures are used to define specific targets for the monitoring robot, e.g. a specific type of motion or a pose, informal rules have been defined by Marayong *et al.*, 2003 to determine a virtual fixture control law:

1) A surface $S \subseteq SE(3)$ which is the motion objective

2) A control law $u = f(x, S)$ that, if $v = u$, moves the end-effector into the surface $S$, which formally corresponds to:

$$\lim_{t \to \infty} x \in S$$

3) A rule for computing the matrix $D = D(t)$ relative to $S$, where $\langle D \rangle u = 0$ *iff* $u = 0$. That is to say that the end-effector motion is "unfiltered" only when the monitoring robot is not moving.

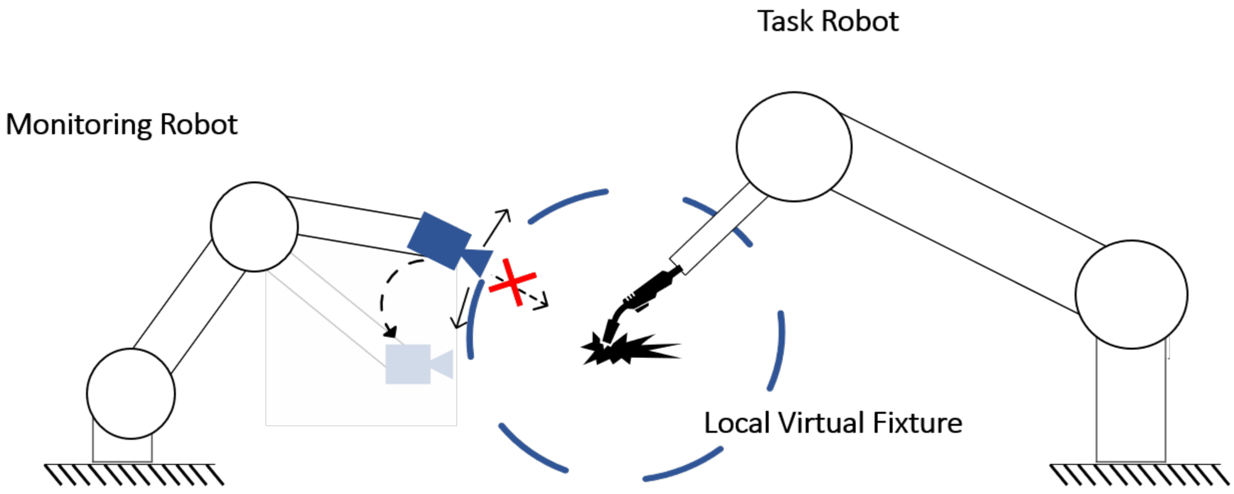


Figure 3.3: Representation of the forbidden region around the task robot. The monitoring robot moves only on the surface of the sphere. ©2018 IEEE

However, these informal laws refer to the case where the VFs are used as motion reference. In the case of the monitoring robot and task robot, the virtual fixtures define the forbidden regions that shouldn't be violated by the monitoring robot. Therefore during manual operation, the monitoring robot motion switches between *free* and *constrained* movement depending on whether the monitoring TCP is in contact with a forbidden region.

An additional observation is that such virtual fixtures are not static in the 3D environment where the monitoring robot is being controlled. Instead, the virtual fixtures can move, typically together with the end-effector of the task robot. In the literature they are called dynamic virtual fixtures or dynamic active constraints Bowyer, Davies, and Rodriguez y Baena, 2014. The virtual fixture is initialized with respect to a point or object in the 3D environment, such as the task robot tool tip or for example the workpiece. If the point to which the local virtual fixture is anchored moves inside the workspace, then the whole fixture moves along as well. This behaviour is the motion that is generated by the autonomous tracking part.

### 3.1.1 Varying Compliance

Depending on the application scenario, a noncompliant virtual fixture can bring advantages as well as disadvantages. In this particular case, the zero compliance property of the local virtual fixture conveniently allows to fulfil the monitoring job's implicit objectives. These objectives are the orientation of the monitoring robot end-effector (i.e. the camera view) towards the anchor point (the object that is being inspected) and a minimum distance from the workpiece or task robot for example.

However, the local virtual fixture does not give any constraint about how the motion should be influenced when approaching other obstacles. Currently obstacles (or forbidden region that can be regarded as obstacles) are expressed in the 3D environment as virtual fixtures so to prevent their contact with the monitoring robot. In the current implementation, however, it is possible that the local virtual fixture partially intersects one or more obstacles' forbidden regions if the user decides to change the inspection target by moving the local virtual fixture inside the workspace.

Whenever this happens, it is possible to provide more than just haptic feedback when the monitoring robot comes in contact with the obstacles virtual fixtures (as it moves along the LVF surface). The compliance of the system can be modified to simulate an increase in friction as the robot approaches an obstacle, even though the robot still only moves along the *preferred* directions of motion.

It is important to notice that in the case of a partly compliant VF, a similar result could be achieved by gradually changing $D = D(t)$ to exclude the direction of motion that would move the robot toward the obstacle.

It is assumed that the LVF is spherical, and that it intersects a certain obstacle represented by its own VF, $S_{obst} \in SE(3)$. It is also assumed that the set of point of the intersection is known, and that for any position of the end effector $\boldsymbol{x}_{tcp}$ it is possible to determine the point $\boldsymbol{P} \in S_{obst}$. Let us introduce the varying compliance $c = c(\boldsymbol{x}_{tcp}(t))$, function of the end-effector position $\boldsymbol{x}(t)$ :

$$c(\boldsymbol{x(t)}) = \begin{cases} 1 & dist(\boldsymbol{x}_{tcp}, \boldsymbol{P}) > h, \quad 0 < h \leq R_{LVF} \\ w dist(\boldsymbol{x}_{tcp}, \boldsymbol{P}) & dist(\boldsymbol{x}_{tcp}, \boldsymbol{P}) \leq h, \quad 0 < w \leq 1 \\ 0 & dist(\boldsymbol{x}_{tcp}, \boldsymbol{P}) = 0 \end{cases} \tag{3.8}$$

where $h$ is a threshold that determines at what distance the system starts having less compliance on the preferred directions, while the term $w$ is a scaling factor. If the LVF is spherical, $dist(\boldsymbol{x}_{tcp}, \boldsymbol{P})$ corresponds to the spherical distance. If the LVF is not of canonical shape (e.g. is a sensor generated 3D surface) the distance between two points is computed as the distance between two nodes on a graph, where the graph is obtained by sub-sampling the 3D surface. More specifically, once the sensor-generated 3D surface has been sampled, it is possible to approximate the distance of two points (A and B) on its surface as the distance between the two nodes of the graph that are closest to the points A and B on the corresponding 3D surface (this approach has been described in the work of Sintorn and Borgefors (2004) for reconstructed 3D images and sampled with grids).

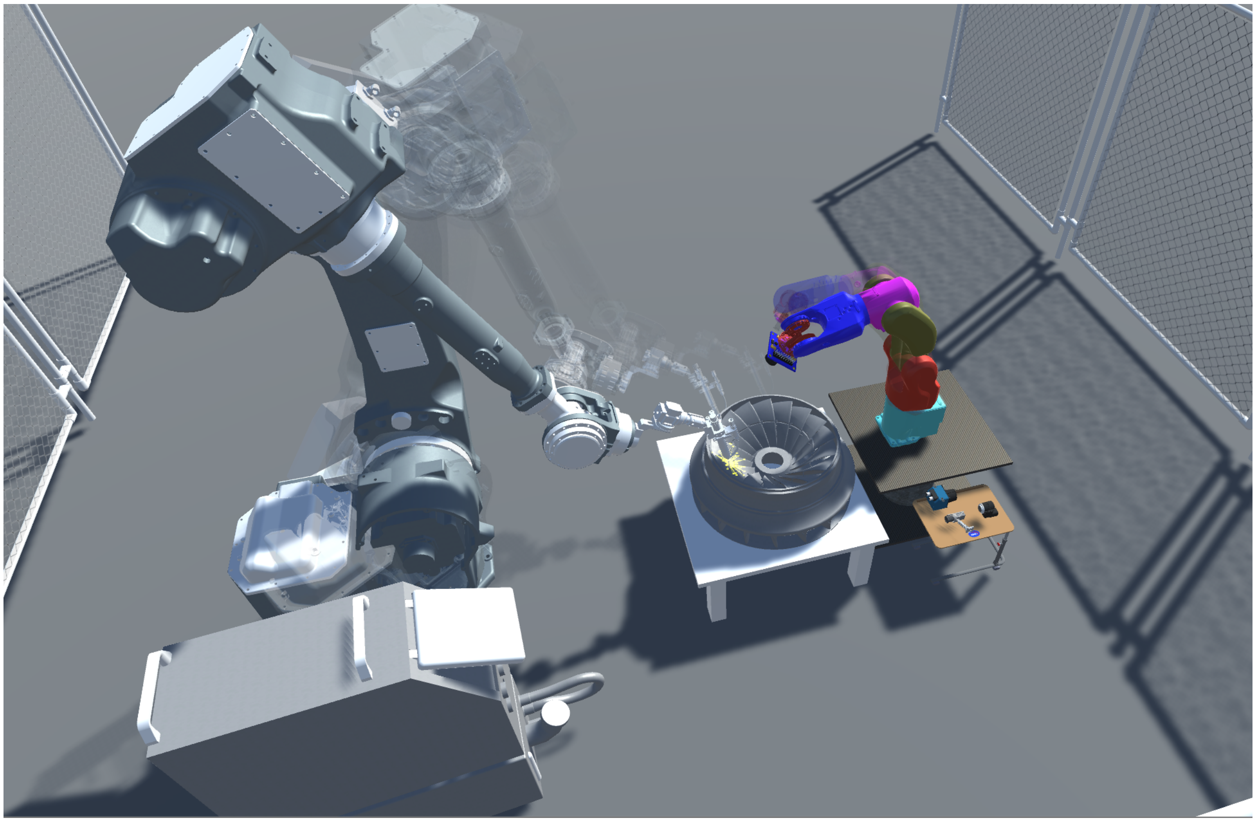


Figure 3.4: Scene view of the monitoring robot environment.

## 3.2 Monitoring Robot Control

Most of this section is dedicated to the description of the scripts and features that have been implemented in the Unity3D environment. As it is described further down in this section, Unity3D is used for the visualization of the monitoring robot and task robot's models and it is in this 3D environment that the VF redirection is calculated and applied.

It is important to clarify the distinction between a system as the one chosen and a simulation software alternative as for example Gazebo that is commonly used in ROS simulations. While Gazebo and similar simulation software are used to create replicas that are as close as possible to the real cell in terms of hardware and sensor behaviour, Unity3D has been chosen as the engine for the visualization and plays a key role regarding the remote user-interface design and implementation. The main purpose of such engine for the monitoring interface is that it allows for both a generic supervision of the robotic cell and a tele-operation inspection.

Moreover, the use of Unity3D lays the foundation for the use of more sophisticated visualization techniques such as virtual reality and mixed reality and the use of the monitoring system for configurations with multiple robot cells. However, part of the work towards the use of different visualization techniques has been omitted in this thesis since it is not within the main scope of this research.

### 3.2.1 Tele-operated and Autonomous Motion

The task robot and its operations must be monitored from different points of view, and that is accomplished by the tele-operation features. The monitoring robot can in fact be controlled and moved around a target without colliding with the task robot. However, the process that is being monitored is not static and in the majority of the cases the task robot is moving along a certain path and performing operations in different positions. The control system in Unity3D tracks the motions of the task robot and applies the same relative motion to the monitoring robot. In this way, the monitoring and task robot's relative positioning is only influenced by the initial state and the manual adjustments made by the user.

The monitoring framework in Unity3D makes use of the joint values of the task robot, obtained through the communication scripts that can provide Unity3D with such information coming from either the custom made high speed interfaces for Nachi FD controllers (FD-HS) (see Section 3.2.7) or from the ROS environment. Since the planned trajectory of the task robot is not known, the monitoring system needs to use the information coming at runtime, and make the monitoring robot track the task robot's path in a leader-follower fashion. The autonomous motion generated when the task robot is moving assists the user in focusing solely on the inspection of the workpiece and process itself, by acquiring in real time the task robot's pose and using it to control the task and monitoring robot's relative position. If the user wants to inspect the process in tele-operated mode he/she can do so while having the monitoring robot and task robot relative position constant unless he/she intervenes.

Regarding the tele-operated mode alone, it is important to describe how the monitoring robot is represented in the Unity3D scene and in what way the user controls the real hardware. The monitoring robot is represented in the Unity3D environment with two identical 3D models:

- *Shadow robot*: the 3D model of the monitoring robot that is synchronized with the real hardware pose in real time. This model is used in the Unity3D environment to assess the actual pose of the monitoring robot in the workspace.
- *Target model*: the 3D model of the monitoring robot that can be controlled directly by the user during tele-operation mode. The target model is also directly affected by the VF control and redirection.

The shadow robot and the target model, along with the scripts that orchestrate their interactions, have been designed and developed anew during the course of this research. The 2-models design has been to some extent inspired by the concept of a virtual mechanism, introduced for the first time by Joly and Andriot (1995). In their approach, the motion of the tele-operated robot is controlled by simulating the kinematics of an inertia free mechanism, which is attached via abstract equations, to both the master and slave device.

To control the real hardware, the monitoring framework keeps track of the distance between the shadow robot and the target model controlled by the user to send the appropriate position reference to the real counterpart (described as well in Section 3.2.7). The system also continuously performs a safety check to ensure that the distance between the shadow robot and the tele-operated one is never exceeding a certain value, which also ensures the maximum speed for the monitoring robot during tele-operation. In the following section the Unity3D-ROS structure will be described in more detail, along with the main script's functionalities and variables.

### 3.2.2 The Unity3D-ROS Structure

Unity3D (Technologies, 2017) is a game engine, and the 3D environment contains scripts assigned to what are called game objects. Game objects can essentially be any sort of entity, from empty placeholders to the monitoring robot model, or the camera view. Objects can be duplicates, along with the scripts assigned to them. Each frame, the engine executes all relevant methods from all scripts and updates the appropriate variables. Unity3D is not a real time system, and there is no way to guarantee a specific execution time or amount of frames per second during the execution. However, Unity3D allows to specify certain scripts to be executed in a pre-determined order to ensure that certain interactions are performed correctly every frame. This Unity3D feature has been used to ensure for example that the communication scripts are executed only after the message data has been processed by the other scripts.

The monitoring companion consists of an industrial manipulator and a Unity-ROS framework for controlling it. In Section 3.1 the approach based on admittance virtual fixtures was described and how such abstraction is used to track a moving target (it could be the end-effector of the task robot for example). The features that allow the solution to be re-used in different industrial application thanks to its compatibility with ROS will be discussed in this section. The control of the monitoring robot is performed in Unity3D, as well as the calculations for the virtual fixture control and the processing of user inputs. The information belonging to the task robot and the industrial process to be monitored instead are communicated to the Unity3D environment from ROS. The task robot communicates its position in the workspace to ROS, which is then forwarded to Unity3D via web-socket communication. More than just the joints data could be passed from ROS to Unity3D however, the information available from the task robot is the minimum amount necessary to perform an autonomous tracking and enable the monitoring of the industrial process. The pose of the task robot could, for example, be estimated by a camera vision algorithm and then passed to the Unity3D environment without changing the scope of the algorithms of the monitoring robot. In the current implementation, the task robot's joint values are obtained via the Nachi high speed interface, explained in more detail in Section 3.2.7. Unity3D is therefore not only used for visualization and for handling the user input coming from the joystick, but is also used for the most important calculations of virtual fixture constraint enforcement and for calculating the monitoring robot velocity that is dictated by the end effector position relative to the virtual fixtures.

In Figure 3.5 the Unity3D code structure developed for the monitoring framework is summarized. The monitoring framework software architecture has been specifically designed and developed during this research work to achieve the functionalities required for the monitoring robot. On one hand, the top and bottom layer handle user inputs and robot and sensors communication respectively; on the other hand, the core layer of scripts is in charge of handling VF collisions, robot motion and VF redirection among the main tasks.

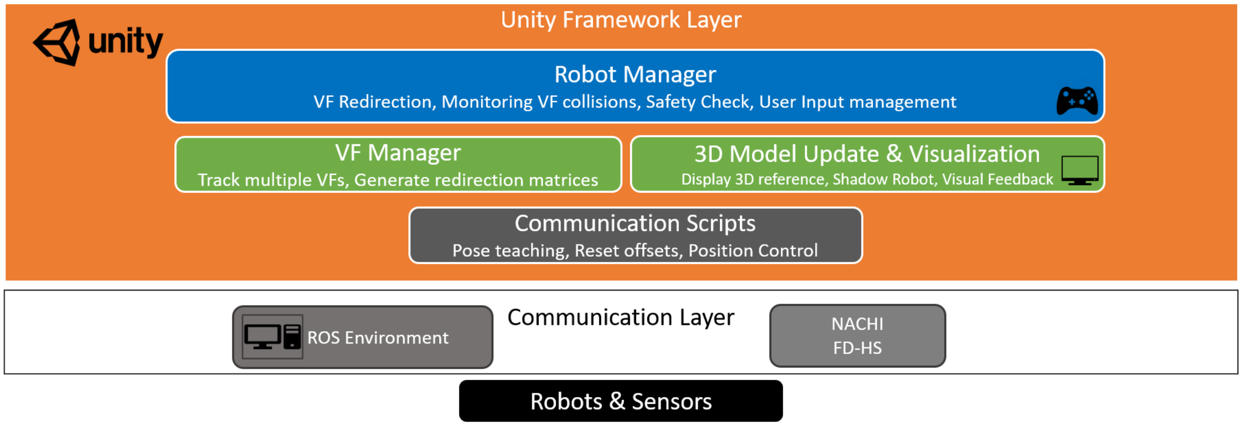


Figure 3.5: Unity3D software structure for the monitoring framework.

The most important functions are performed by the following scripts:

- *Robot Manager*: stores the robot information and performs motion tasks, safety checks and dispatches to other scripts events about the robot's state.
- *VF Event Manager*: is responsible for the compliance factors of the VFs, stores their locations and calculates the filtering matrices when the robot collides with a forbidden region.
- *Communication scripts*: send low level information and heartbeat messages

to the robot and sensors, like the FD-HS. The communication scripts are linked to the ROS communication script in order to receive information from ROS topics when either the task robot or monitoring robot are configured in the ROS environment.

An additional script which is used for the experiment scenarios is the *Experiment Manager*, which is in charge of recording, storing and processing data during the experiments and produces the metrics of interest. In addition to the data required to produce the metrics (described in Chapter 4), the experiment manager collects information about the robot pose and its motion during tele-operation so that it is also possible to analyze joint angles and position in the workspace after the experiment is over.

### 3.2.3 Robot Manager

The robot manager script is in charge of storing the information about a robotic unit that has been set up in the 3D environment. A robot manager script is assigned to both the monitoring robot and the task robot, although the robot manager script on the task robot only stores pose information and cannot control the task robot motion or auxiliary functions. Unless specified, the subject of this section will be the robot manager script of the *monitoring robot*.

In general, the robot manager receives user inputs from the communication layer, and translates them into motion of the target 3D model. If such motion needs to be filtered, this will be performed as a middle step from the VF manager before forwarding the data to the robot manager. The 3D target model pose of the monitoring robot is then updated according to the motion vector calculated in the current frame, and the distance between the shadow robot and the target model is used to send the properly scaled velocity commands to the hardware.

Along with motion control, the robot manager also performs an additional safety check to ensure that communication with the monitoring robot is stable and that the motion commands are not sent anymore if the monitoring robot cannot execute them. The flowchart of the update cycle of the robot manager script is shown in Figure 3.6.

Moreover, the most important variables and information stored into the robot manager are described here:

- *Joint angles for target model & shadow robot:* each joint angle is stored for both the target robot model and the shadow robot. The information is stored an additional time but already converted in the format accepted by the hardware to save time;

- *Cartesian position of target & shadow TCP:* the position in the workspace is stored for both the target robot's TCP and the shadow robot's TCP, with respect to the robot base.

- *Current velocity vector:* the velocity vector that is applied to the target robot's

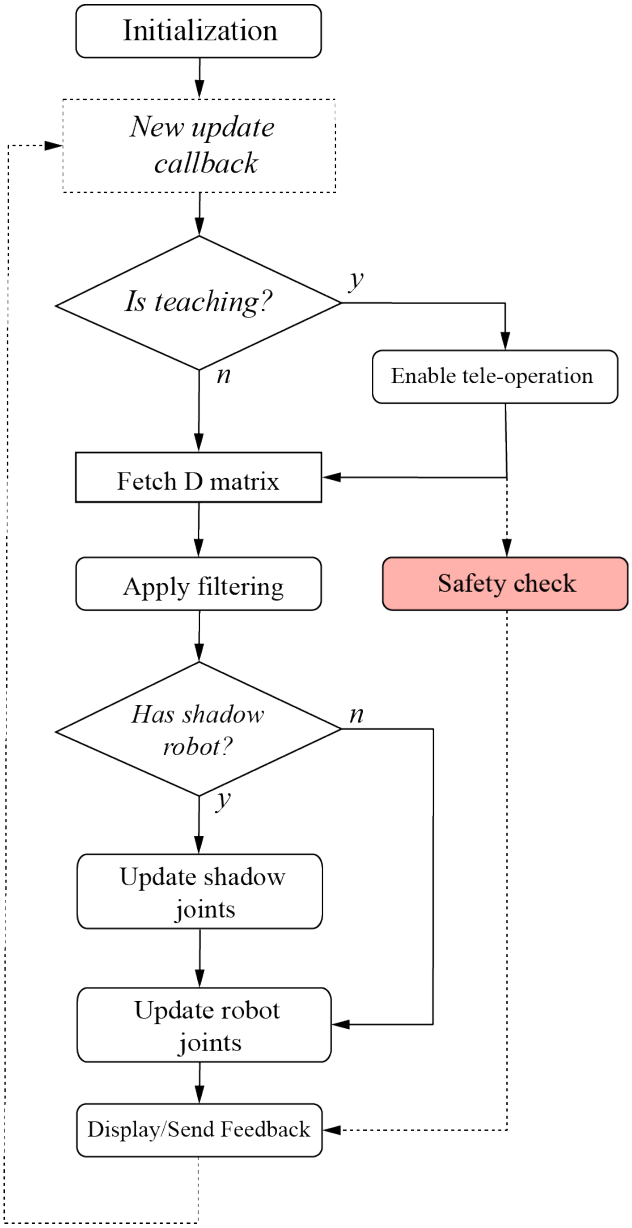


Figure 3.6: Robot Manager script: flow chart of the update cycle.

TCP is stored in the robot manager. The vector stored has already been filtered by the VF manager if the robot is in a collision state;

- *Control type:* this variable allows the control type to switch from cartesian to joint space. It is not used in the experiment scenario and is typically used by the autonomous agent if needed. In joint space the velocity vector is replaced with speed and delta motion for each joint;
- *Safety check active:* if a communication or hardware error causes the monitoring robot to not move, the robot manager can detect the discrepancy between the target model and the shadow model and stop any motion function;
- *Collision state:* in the case of a collision with a forbidden region, the robot manager stores that information and passes it to the experiment manager for example to store that information when collecting data for the metrics;
- *Lock Rotation Of Camera:* auxiliary variable that allows to lock the rotation of the camera view, which translate in limiting the DOFs of the monitoring TCP.

### 3.2.4 Virtual Fixtures Scripts

The VF event manager passes collision information and the filtered velocity whenever collision events are fired. This information about the redirection and the collision are used by the robot manager and the experiment manager scripts. The event manager itself is simply a collection of events to which all the VF Model scripts subscribe and that will fire if they are part of a VF collision.

On the other hand, each VF object in the Unity3D environment is given a *VF Model* script, which implements the basic methods that each virtual fixture should have: a method to fire the collision events, to produce the $D$ matrix and to filter the velocity vector. In addition, each VF Model script stores their own value of the compliance factor $c_\tau$. This choice allows to have forbidden region or obstacles with different compliance/stiffness when the monitoring robot's TCP is in contact with them. It is also worth noticing that the compliance can be also globally set by the experiment manager script by sending a common value through the VF event manager to all VF in the scene.

The reader is referred to Algorithm 1 for the generalized description of the script behaviour, while the main actions performed by the VF model script are described here in more details:

1. Initialization: consists in subscribing to the collision events and free motion allowed event from the VF event manager;
2. Continuous loop waiting for a collision event: at the beginning of script and at the end of a collision, the "free motion allowed" event is triggered to signal that no redirection is needed and no collision is in place.
3. Collision detected: when a collision with the monitoring robot and a VF is

detected, the VF model calculates the normal and tangent plane at the point of contact and generates both the kernel and span matrices. Finally, it collects this information and triggers the collision event. The robot manager - as can other scripts subscribed to the event - uses the information in the collision event to filter the velocity vector before apply it to the robot's TCP.

**Algorithm 1:** *VF Model* update cycle

```
Initialization;
while Monitoring Robot is active do
    while Collision do
        $CP$ = collision point with VF and monitoring robot;
        $collisionNorm$ = surface normal at CP;
        $tan1, tan2$ ← Build tangent plane to VF at CP;
        $D$ = CreateDMatrix($CP, collisionNorm, tan1, tan2$);
        $span$ = [$D$];
        $kernel$ = $\langle D \rangle$;
        $VFcollisionInfo$ = GenerateCollisionInfo($CP, span, kernel$);
        SpanMatrixReady(VFCollisionInfo);
    end
end
```

Therefore, whenever the robot manager receives new collision info from a virtual fixture it will apply the filtering according to the span and kernel matrices contained among the event variables. In the case of multiple VF contacts at the same time, the monitoring robot will simply filter the velocity vector for each unique collision detected.

This section described the main features of the virtual fixtures scripts, that are in charge of sending collision information to the robot manager and for example also the experiment manager so that other functions can make use of the redirection matrices, contact normal and tangent plane. In the next section the communication script will be described briefly together with the main functions used to exchange information betwwen the ROS environment and the monitoring framework in Unity3D.

### 3.2.5 Communication Scripts and ROS# Library

The communication scripts, as briefly introduced in Section 3.2.2, are in charge of passing the monitoring and task robots information to their respective robot managers and other scripts in the Unity3D scene.

Generally, the most important information is the value of each robot joint which in the case of the FD-HS implementation for the Nachi robot is passed in the form of encoder values. For other robot interfaces this can be directly an angle in degrees or radians, as is the case for example for simulated robots in the ROS environment.

In the implementation used for this research work, the monitoring robot manager first receives from the communication layer - connected directly to the FD-HS - the robot's parameters necessary for the encoder-to-angle conversion and passes them to an auxiliary script that will take care of the conversion and variable handling.

In normal operation conditions, even if the user is not tele-operating the monitoring robot, the robot manager each frame issues a *reading* (i.e. synchronization command to the communication layer in order to update the pose of the robot. A reading command consists of the robot manager receiving back from the communication layer the current pose of the monitoring robot to then synchronize with it. As a background check, the reading operation is also sent by the robot's communication script in the case its robot manager script has not issued the command for a number of consecutive frames. This safety check is performed to ensure that the connection with the monitoring robot hardware is stable and that no commands are buffered in the case of an error in the robot manager execution.

Moreover, the communication layer of the monitoring framework is also tasked to communicate with the ROS environment if the task robot and the process information are stored and updated there. To establish the communication and subscribe to the topics that contain the information needed in Unity3D, the monitoring framework makes use of the ROS# library (Siemens, 2020).

In general, if the task robot is being simulated or simply updated in the ROS environment, the ROS# library provides scripts to connect (TCP connection to RosBridge) and retrieve information from topics in ROS. The messages contained in the ROS topics are stored in variables in Unity3D that are then used to update the task robot's pose.

### 3.2.6 Visual and Haptic Feedback in Unity3D

While the previous paragraphs described the scripts that are always running in the background as part of the monitoring framework, this subsection focuses on the visual and haptic feedback that are provided when the system is tele-operated and the user needs assistance during the navigation. More specifically, the feedback is intended to help the user locate obstacles that are outside the field of view of the camera and better understand when the monitoring robot is being redirected and when it has collided with a VF.

A schematic example of the behaviours of both the visual and haptic feedback during tele-operated mode is provided in Figure 3.7. An image taken from the feed of the monitoring robot's camera is shown in Figure 3.8 to provide a better visualization of the 3D arrow used as visual feedback.

The visual feedback consists of a 3D arrow that is displayed in the bottom right corner of the camera view. The script handling the arrow's behaviour subscribes to the collision events so that the arrow icon is pointing at the closest VF during tele-operation. The icon is coloured in green if there is no collision in place, and

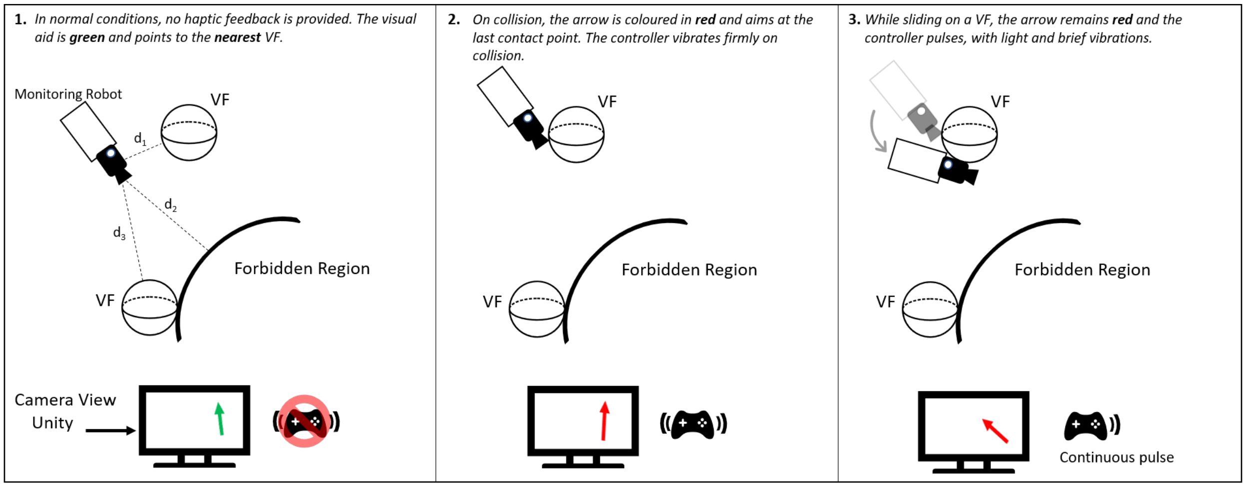


Figure 3.7: Schematic representation of the two different vibration types of feedback, on collision and while in contact with a VF. The arrow icon points at the closest VF or at the last point of contact detected, coloured in red or green depending on the collision state. For a better visualization of the 3D arrow indication in the camera view, the reader is referred to Figure 3.8.

changes to red when the monitoring robot is in contact with a forbidden region. When the monitoring robot is in a collision state, the arrow is pointing at the last contact with a VF detected by the system.

The intent is to provide the user with a visual indication of the contact point with the obstacle that triggered the system's VF redirection. The user can then attempt to move the robot in the opposite direction if his/her goal is to resolve the collision state as quickly as possible and resume unfiltered navigation.

As for the haptic feedback provided through joystick vibrations, it is also briefly described in Figure 3.7 and can be summarized as below:

- *On collision:* in the frame where collision has been detected, the joystick vibrates with medium-high intensity for a short period, to signal that the monitoring robot has entered a collision state and cannot navigate as freely as

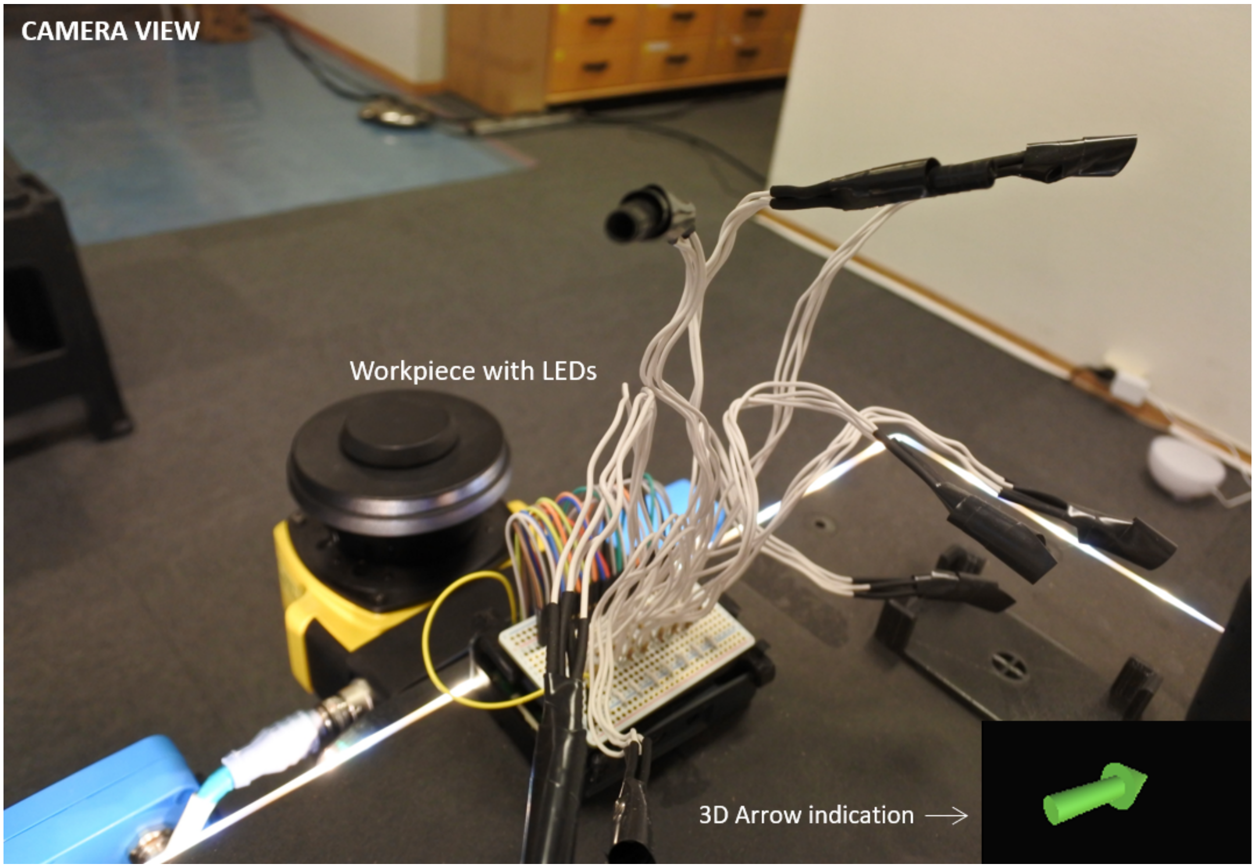


Figure 3.8: Camera view of the workspace from the monitoring robot's end-effector. The 3D arrow indication is visualized at the bottom right corner of the view. The arrow points at the point in the workspace where the last collision has detected by the system. In this figure there is, however, no collision occurring since the arrow is coloured in green.

before.

- *During redirection:* if the user is navigating the monitoring robot while in contact with one or more VFs, the joystick vibrates in a heartbeat fashion with lower vibration intensity. This is to signal the user that redirection is still active as the collision state has not been resolved.
- No vibration is provided in all other occurrences.

Together with the visual indication, the goal of the haptic feedback is to provide the user with additional information regarding the surroundings of the monitoring robot, that may be outside the camera view. The effects of the implemented feedback modalities on the users' navigation and their performance are discussed in Chapter 4.

### 3.2.7 Nachi High Speed Interface

As described in Section 3.2.1, in the Unity3D scene there are two identical 3D models for the monitoring robot, one being the "slave" and one being the "master". The "slave" model is the shadow robot and is always in the same configuration of the real hardware counterpart. In fact, the shadow robot model in Unity3D is synchronized with the real hardware thanks to the encoder values provided by the Nachi high speed interface, also called FD-HS (due to the robot controller name being "FD series"). The FD-HS operates with a position control loop with the reference position coming from the monitoring system, and the setup is sketched in Figure 3.9.

The update frequency of the encoder values has a limit of 5ms (both for writing and reading), which limits the Unity3D maximum frame rate to 200 fps. However, such a limit is more than enough for real time application and does not constitutes a bottle neck in our system.

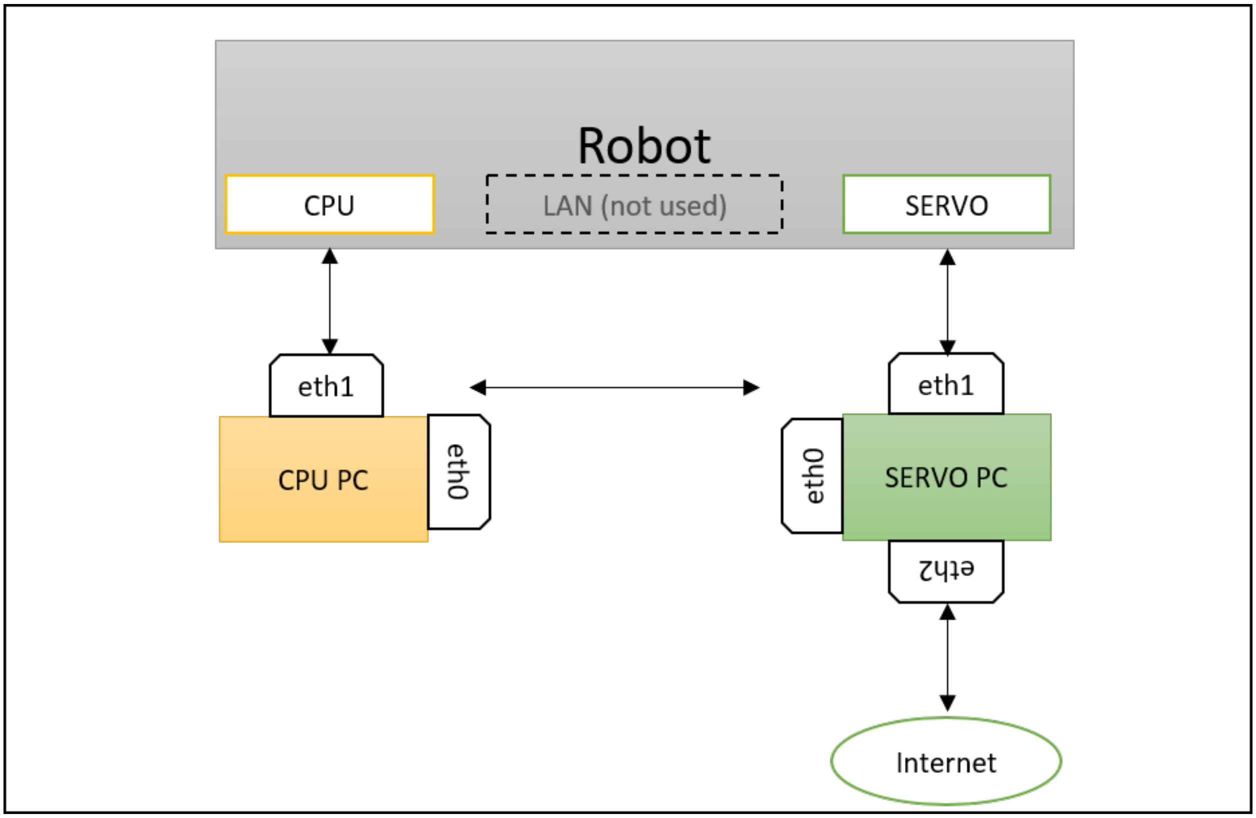


Figure 3.9: The diagram of the structure of the FD-HS. The current implementation uses two Raspberry Pi3 to communicate with the CPU and servo board of the Robot Controller.

## 3.3 Preliminary Work on an Automated Task

As introduced in Chapter 1, the work in this thesis is often described with a robotised welding application in mind, but through the chapters an abstract application domain has been created in order to focus on the two main issues identified in Chapter 2: the shared control between the human operator and the monitoring robot; and the nature of the monitoring framework's interface. These issues will be investigated further with a usability study described in Chapter 4.

To conclude this chapter, in this section the initial promise made in Chapter 1 is kept and the abstraction that was made in the introduction is here put aside.

In particular, a demonstration of the potential use of the monitoring robot for an automated task is shown in Figure 3.10 and is described in further detail in the rest of this section. The work described below, however, represents an initial study in one specific application designed to indicate the promise for further more comprehensive application-domain specific research and development that lies beyond the scope of the current thesis. Even though such a demonstration has not been the subject of the usability evaluation in this thesis, it still provides a valuable contribution in highlighting this thesis achievements as well as future potential.

With respect to the task robot and the testing of its path during the integration, the monitoring companion can be used mainly in two ways: the first is the approach described throughout this chapter until now and it consists in controlling the monitoring robot to inspect the task robot as it runs autonomously; however a second approach that requires the monitoring robot to make use of autonomous navigation is when the user is programming the task robot and would like the monitoring robot to follow along a new trajectory as it is being defined by the system integrator.

Both cases are shown in Figure 3.11 and Figure 3.12 respectively. Although they might appear similar in setup, Figure 3.11 shows the operator holding the *monitoring* robot joystick, so being in tele-operation mode and the monitoring framework's autonomous navigation is taking care of tracking the task robot movement.

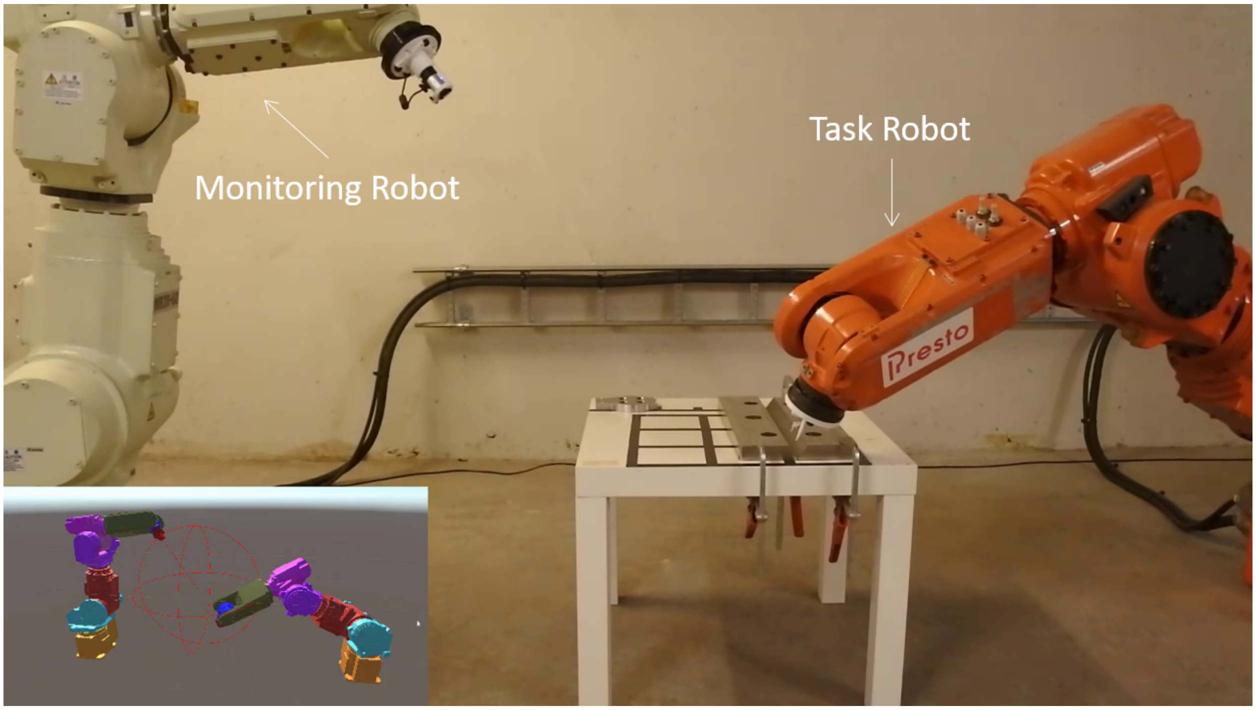


Figure 3.10: Setup with two industrial robots. The monitoring robot is equipped with an RGB camera sensor.

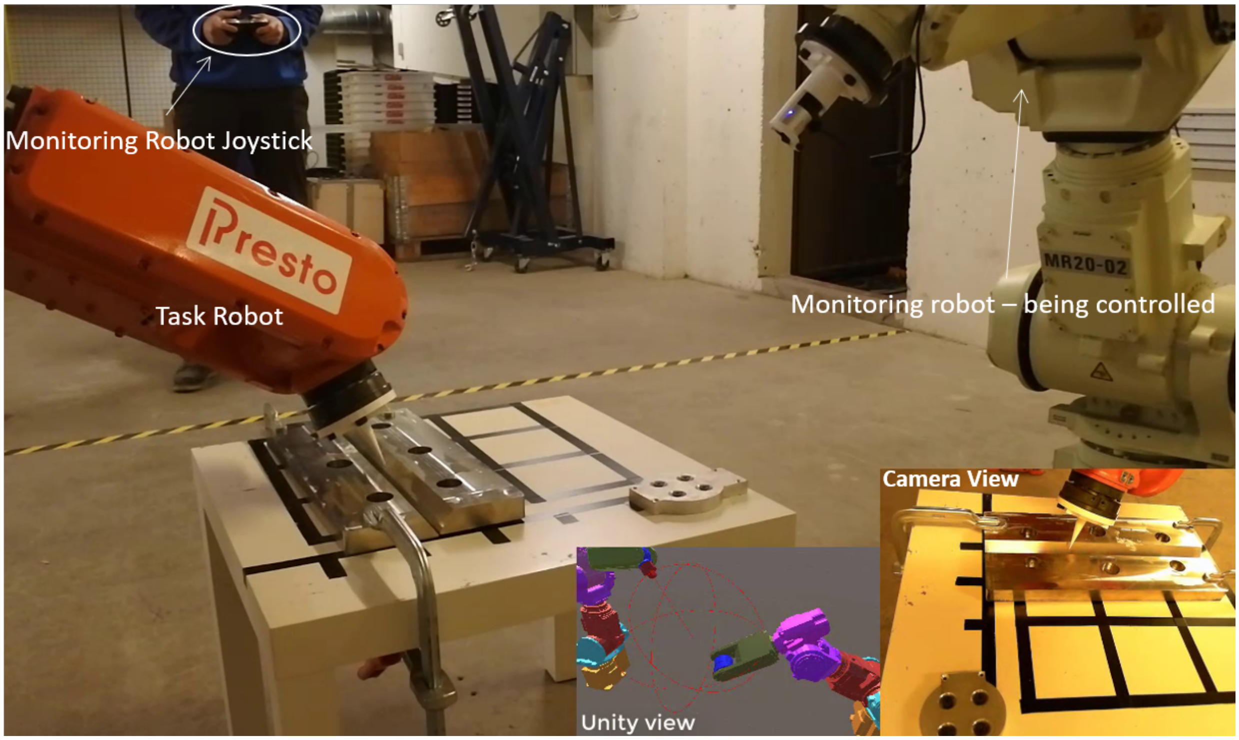


Figure 3.11: The operator is controlling the monitoring robot with the joystick as described in Chapter 3. The task robot is executing a pre-programmed path and the autonomous navigation can keep the relative position constant.

Figure 3.12 shows the same operator holding the *task* robot's teach pendant instead, that is used to jog the task robot itself. The motion of the task robot is therefore not planned or pre-programmed, but it can, however, be tracked by the monitoring robot since the framework synchronizes it with the task robot's pose in real time. In this configuration, when the operator moves the task robot, the monitoring robot can autonomously maintain a constant distance from the task robot's TCP and also the orientation specified beforehand.

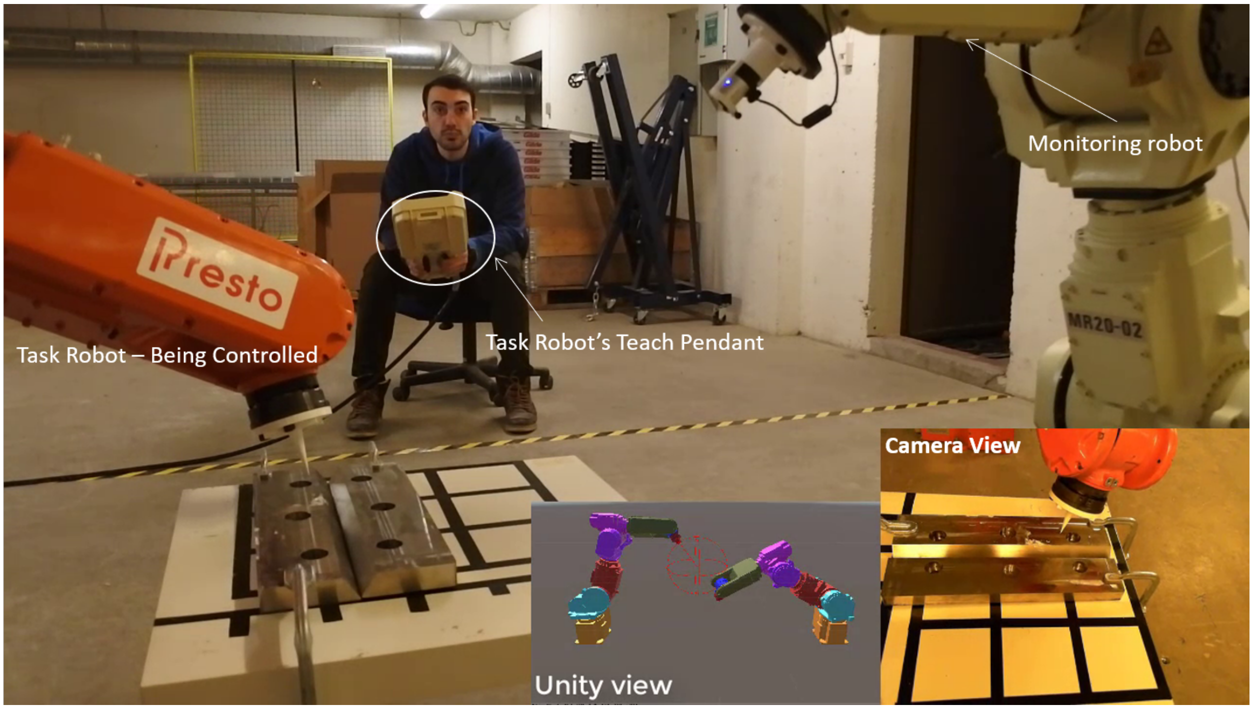


Figure 3.12: The operator is controlling the task robot by using its teach pendant. The monitoring robot is in the meantime maintaining its relative distance to the task robot's TCP constant.

These two scenarios do, indeed, show promise for applying the work carried out in the body of this thesis in a real robot task setting and have been investigated in preparation for the use of the monitoring robot in an actual robotised welding task. However, continuation of the current work is more fitting for an industry-biased and closer-to-market design development study and is, therefore, considered to lie beyond the scope of this PhD.

The next chapter will continue with the evaluation of the monitoring framework in an abstract application domain, and will analyze the benefits of the features that have been described in this chapter and their effect on tele-operation.

# 4 Usability Evaluation

Chapter 3 treated in more detail the technical implementation of the monitoring framework and the choices made during its development. The abstraction that was initiated in Chapter 2 in order to obtain a set of challenges not specific to any one application domain is continued in this chapter. The performance of the monitoring framework is going to be evaluated in a tailored experiment setup. Doing so makes it possible to find the answer to the research questions posed both here and in Chapter 1 without ambiguity that would result from influences from details specific to industrial applications (e.g. the case of robotised welding, which will be returned to later in the thesis as an example).

The purpose of the monitoring robot is to assist the system integrator in the tuning phase of an industrial process, which is the main task for the system integrator. It is therefore important that the monitoring robot can be controlled with as little effort as possible not to increase the workload and the time needed during the optimization phase. It is known that when manual input is allowed for robotic systems, both the control scheme and the user interface play a key role in the overall complexity for the human operator. This chapter describes a usability evaluation on a set of test tasks performed with the monitoring robot on an experiment test-piece, shown in Figure 4.1.

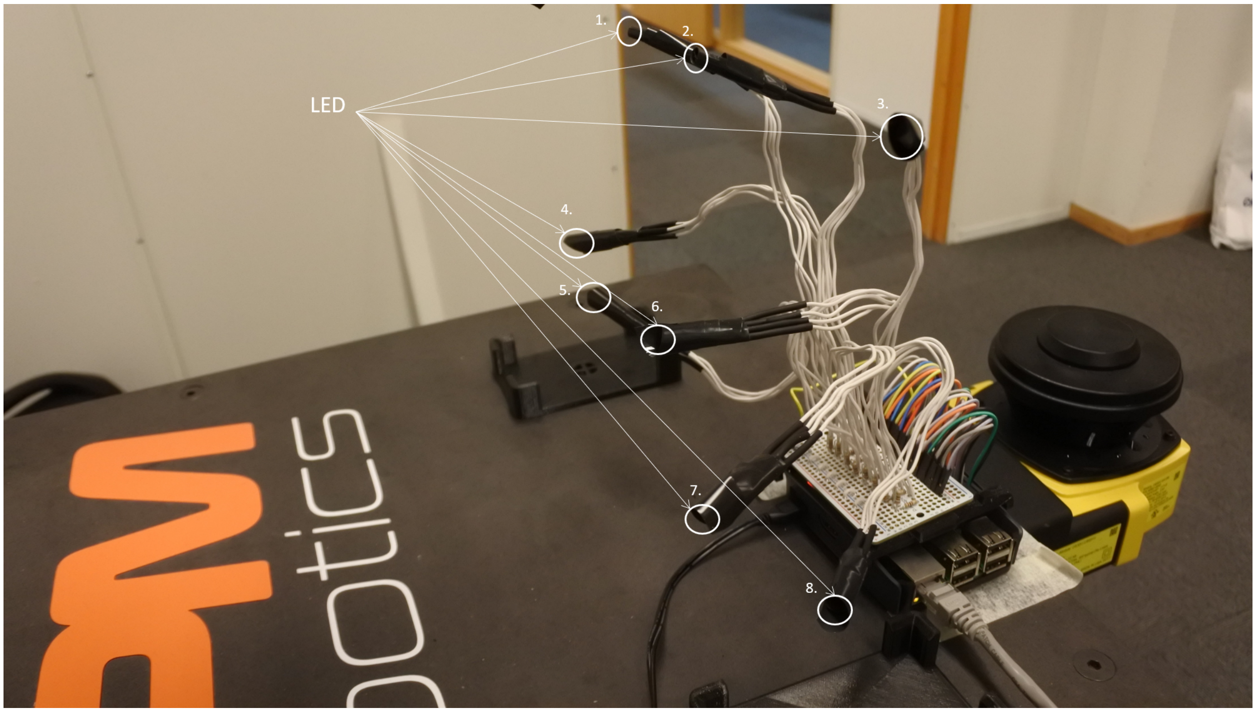


Figure 4.1: Experiment piece used for the static experiment. The LEDs are triggered via a Python script on the Raspberry Pi. ©2020 IEEE

## 4.1 Test Cases and Motivation

A usability evaluation is conducted to provide insights on the efficacy and benefits of the VF control approach and of the monitoring framework in general. It is important to evaluate in particular whether the control of the monitoring robot does not increase too much the mental and cognitive load of the operator. In fact, the goal of the operator is to inspect the workspace and the process carried out in it and not to control the monitoring robot per se.

The main questions that are being investigated in this section are the following:

- *Does the VF control framework allow the operator to control the monitoring* *robot without overloading him/her?*

This question will be answered by comparing TLX indexes among the scenarios where VF redirection was not enabled to the ones where it was active.

- *What are the effects on performance introduced by the VF control framework?*

  This question will be answered by comparing the average completion time among the different experiment scenarios. Moreover, observations will be made on the effects that VF redirection has on the number of collisions and corrective actions.

- *What are the effects of an extra feedback modality on the user TLX score and performance?*

  The final question will investigate more in detail if an extra feedback modality can improve performance in this particular task, and what are the effects on the users' TLX scores when this feedback is introduced, with particular focus on the users' cognitive load.

### 4.1.1 Control Modes

The monitoring robot is controlled by the user via the control framework and a joystick. The user controls will not change, regardless of the presence of constraints in the workspace. However, for the first experiment scenario the control mode will be *unconstrained*, while for the other scenarios will be said to be *constrained*. In the unconstrained mode, the camera view (the robot's end effector essentially) can be freely moved inside the workspace with a simple mapping of rotation and translation to the joystick's buttons. In the constrained mode, the camera view (position and orientation) is influenced by virtual surfaces defined via software. The user can still control the position of the robot in the workspace but subject to these virtual constraints. The inputs mapping for the constrained mode is redefined to consider these constraints (in certain situation a straight-line movement may no longer be permitted, for example when constrained to move on the surface of a virtual sphere): when constraints are in place it will simply be said that virtual fixture redirection is enabled.

### 4.1.2 Choice of Metrics

The experiments aim at investigating whether untrained users can more efficiently accomplish the task of acquiring information from a process with the help of a virtual fixture based control. Moreover, the VF based control should assist the user in respecting secondary objectives such as minimum distance from specific objects or from the workpiece without requiring additional effort from the operator.

The metrics that are collected for both the static and dynamic scenarios are the following:

- Completion time (CT): this is the time to complete a single task, from the start event to the end event. The start of a task is identified by a LED lit and followed by a notification to the user. The task ends when the user finds and correctly classifies the lit LED's colour.

- Number of commands (NC): the number of control commands that a user inputted during a single task. It is representing the number of movements issued to the monitoring robot.
- Number of corrective actions (NCA): the number of times the system with VF actively filters a user movement that would otherwise lead to entering a forbidden region. Corrective actions can be registered only when redirection is enabled.
- Number of constraint pseudo-violations (NPV): the number of times that the system has collided with a VF.

### 4.1.3 Number of Trials

Each participant is asked to complete a task four (4) times, with the task consisting in locating and classifying one lit LED with the monitoring robot. For each time the aforementioned metrics are collected, leading to four measurements per metric for a single user trial. Each user performs exactly one trial, so as to avoid "learning trends" as much as possible since a decrease of completion time and the improvement of the metrics over successive trials it is not the focus of this study.

Although it is to be expected that all users will improve regardless of the control mode, the focus is rather on the differences between the groups of users and their performance with the different control modes.

### 4.1.4 Participant Organization

The majority of the participants is made of students from the last year of BSc and first year of MSc studies, with no restrictions on the type of background. There are no restrictions of age nor of technical background for taking part in the experiment, as it is not a variable of interest. The average age is $24.6 \pm 1.87$, and the participant were coming mainly from a healthcare studies and engineering background, although no statistics were made of that.

## 4.2 Experiment Setup

The static experiment setup consists in a 6-DOF manipulator mounted on a base as shown in 4.2. The experiment workpiece will be placed in front of the robot and will be an electronic board with eight (8) multicolour LEDs that can be lit remotely via script (see Figure 4.1). The monitoring robot will have a camera mounted on the end-effector and will be the primary source of information for the user.

The user's task is to identify which LED is lit at a certain time and what is its colour. To accomplish the task, the users must control the camera view via a

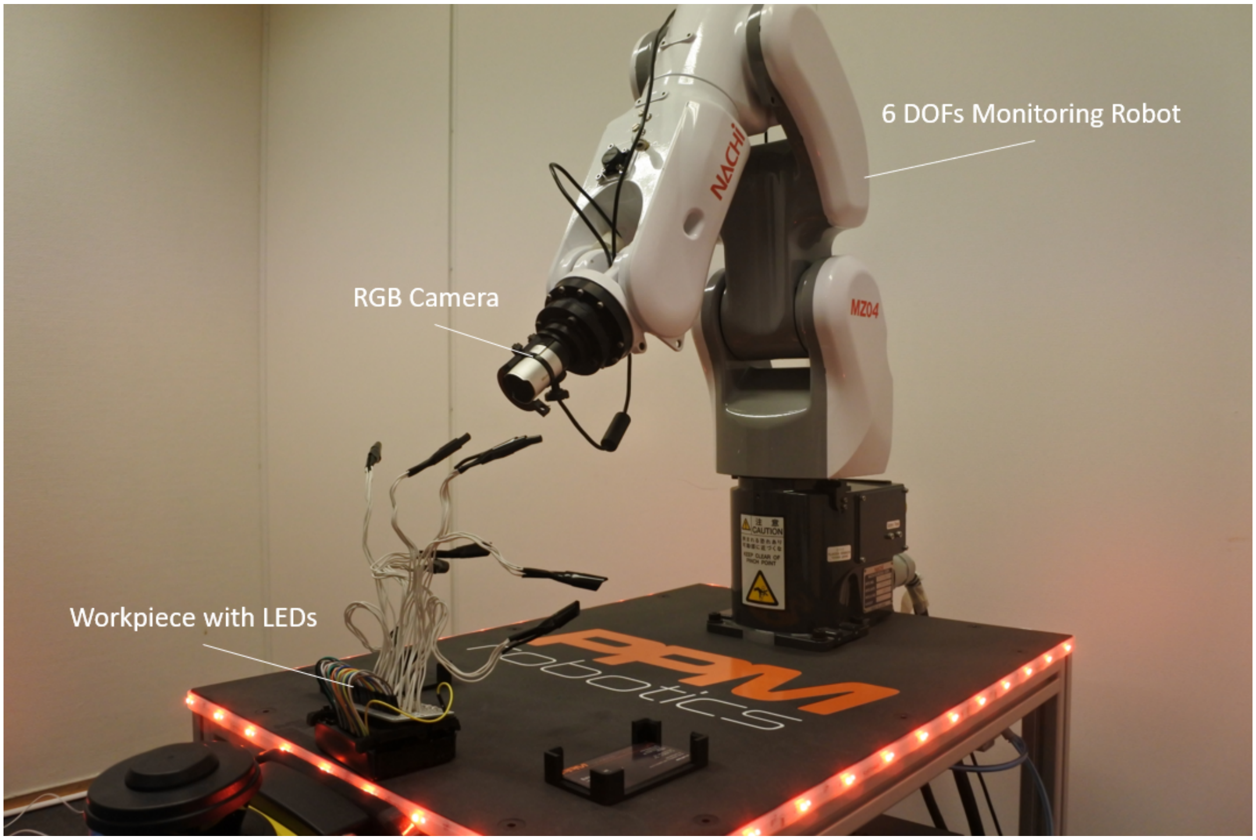
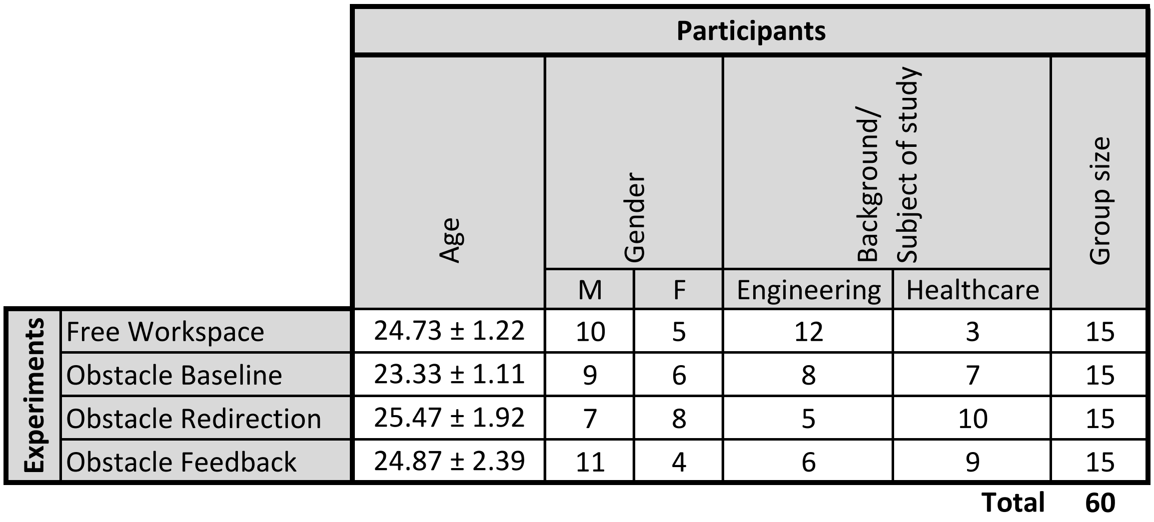


Figure 4.2: The setup for the experiment. A 6-DOF monitoring robot is equipped with a RGB camera. The workpiece is placed in front within the robot's workspace.

joystick (that is, they control the monitoring robot) to be able to look at the correct LED and identify the LED's colour.

Each user is asked to "find" the coloured LED 4 times in a single session. Once these four (4) attempts have been accomplished the user is asked to fill a NASA-TLX form (the empty form is in the appendix at page 125) and the experiment is then over. The experiment is conducted in a "Wizard of Oz" fashion, with an external supervisor in control of which LEDs trigger and with which colour during the experiment. Once the supervisor decides which LED to trigger, the user is notified and can begin the task. A single task is considered completed when the user correctly reports the colour of the LED that is ON. Currently, failing a task can only happen if the user gives up in trying to adjust the camera view correctly or if, in controlling the monitoring robot, the monitoring robot has to be stopped so as to avoid hitting something in the workplace (this can happen only for the unconstrained mode).



| | **Participants** | | | | | |
|---|---|---|---|---|---|---|
| | Age | Gender | | Background/ Subject of study | | Group size |
| | | M | F | Engineering | Healthcare | |
| **Experiments** Free Workspace | 24.73 ± 1.22 | 10 | 5 | 12 | 3 | 15 |
| Obstacle Baseline | 23.33 ± 1.11 | 9 | 6 | 8 | 7 | 15 |
| Obstacle Redirection | 25.47 ± 1.92 | 7 | 8 | 5 | 10 | 15 |
| Obstacle Feedback | 24.87 ± 2.39 | 11 | 4 | 6 | 9 | 15 |
| | | | | | **Total** | **60** |

Figure 4.3: Summary of the experiment participants' age, gender, background and their total number per each experiment scenario.

A third failing condition consists in the user wrongly identifying the colour of the LED from the camera view. This eventuality has been foreseen during the design phase as well. As a mitigation factor, the available colours for the LED are only red, blue or green so that they can more easily be distinguished. Moreover, a preliminary check of the camera feed has been carried out to ensure that the colours do not appear too similar to each other.

Four (4) experiments have been conducted, each with fifteen (15) participants,

| | Obstacles | VF Redirection | Visual and Haptic |
|---|---|---|---|
| **Free Workspace** | - | - | - |
| **Obstacle Baseline** | X | - | - |
| **Obstacle Redirection** | X | X | - |
| **Obstacle Feedback** | X | X | X |

Table 4.1: Overview of the difficulty elements in the experiment scenarios

for a total of sixty individual users (i.e. $n = 60$). The participants' age, gender and background/subject of study is shown in figure 4.3 As it is shown in table 4.1, the number of elements that could influence the users' performance was increased from one experiment to the next. The added obstacles are invisible to the users, and have one additional property in addition to their position and shape: the compliance. As described in more detail in section 3.1.1, the compliance changes how "hard" an obstacle is in response to the user motion that would penetrate it. For all the experiments, spherical virtual fixtures were used. However, in some cases, overlapping so that to the user the obstacles didn't always have a spherical shape.

The first experiment involves simple free motion without VF redirection, with no obstacles in the workspace. In the first experiment, the only virtual fixtures that have been used were to ensure the safety of the workspace and the robot. That means a forbidden region is still defined around the worktable and the workpiece, even if no virtual fixture redirection is present.

The second experiment introduces obstacles in the workspace. The users' goal is unchanged but the robot will have constrained motion every time it will come in contact with a virtual fixture. In the second experiment, whenever the robot is colliding with a virtual fixture the only motion allowed is in a direction that resolves completely the collision state, generally along the normal of the VF on the point of contact. The user motion is completely filtered as long as the collision state is not resolved. The second experiment provides a baseline for the future comparison of completion times and the average number of actions to complete the task.

The third experiment introduces virtual fixture redirection as described in 3.1, which facilitate the robot motion when it comes in contact with an obstacle. For as long as the collision state is not resolved, the user motion is filtered not to violate the constraint further. In the third experiment as in the previous two experiments, the users are not provided feedback on whether they are in a collision state or not.

The fourth experiment adds a visual and haptic feedback component to the experiment scenario. The feedback is provided when the user is "about to" collide with an obstacle, and also during the collision state with a slightly more intense haptic feedback. The visual feedback consists into a visual 3D arrow that points to the nearer obstacle from the user’s point of view (see figure 4.4).

Completion times, number of actions to accomplish the task and number of collisions during the operation, can all be compared across the sets and against the baseline experiment.

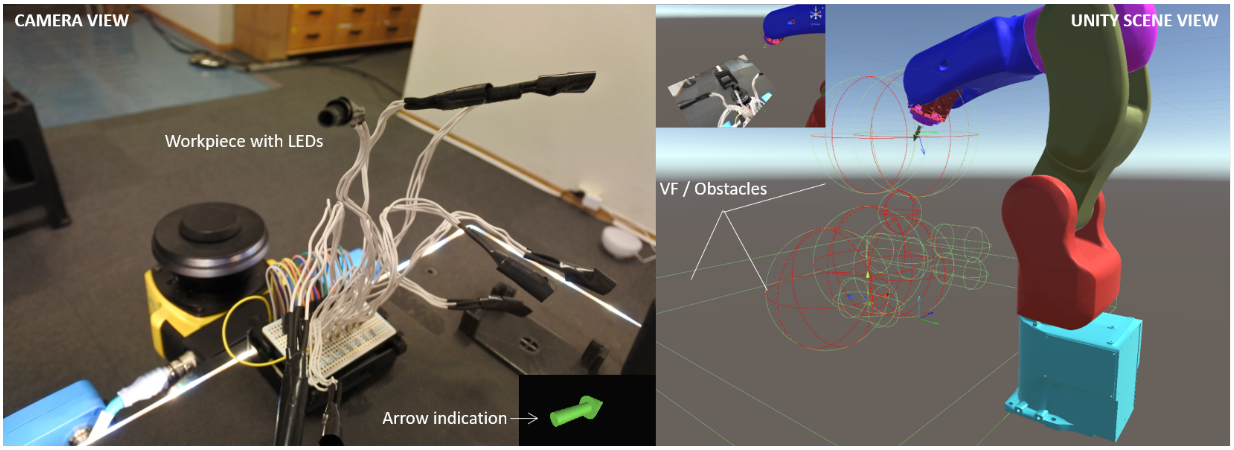


Figure 4.4: The 3D arrow used for for the visual feedback in the fourth experiment scenario. ©2020 IEEE

## 4.3 Scenario A: Free Workspace

As described in the previous section, the first experiment aims at providing a baseline of operation in order to compare users' performance against the scenario without obstacles and VF redirection. When there are no obstacles in the workspace, the users can concentrate on the way they control the robot camera view to complete the task. Although the users are not provided with the information about which LED will be lit at the beginning of each task, they have the time to evaluate the best way to inspect the workpiece. However, finding the quickest route to inspect the workpiece to find the lit LED is not the goal of this experiment, as it is not the goal to observe eventual learning trends in the users performance as they repeat the experiment task. Therefore, the number of task that each user had to perform has been limited to a maximum of five (5) and each user has been only involved in one experiment scenario so to avoid the learning effect as much as possible.

### 4.3.1 Observations

In section 4.7 a more detailed analysis of the experiment results will be provided, comparing results from all the experiment sets.

However, from the TLX reports, it can be noticed that the overall performance of the users throughout the first experiment has been evaluated towards the "easy" side of the spectrum. The average scores for frustration and effort from Figure 4.5 appear to be below the middle line and it is a positive sign of the base system not being too difficult to control even for an inexperienced user. This observation is still in line with what was hoped with the design of the first experiment, where no obstacles are present that can influence the motion of the user and the only unknown system is the base navigation of the robot. Moreover, this result has been probably achieved also thanks to the choice of a joystick and game-like interface, with the users being somehow familiar to such an environment thanks to their age.

## 4.4 Scenario B: Obstacles and No Virtual Fixtures Redirection

In the second experiment, obstacles were introduced in the workspace. All added obstacles had spherical shape, but some of them were partially overlapping so that the resulting forbidden region wasn't perfectly spherical. In this set up, the virtual fixtures used for the obstacles were stiff (that is, with compliance close to zero)

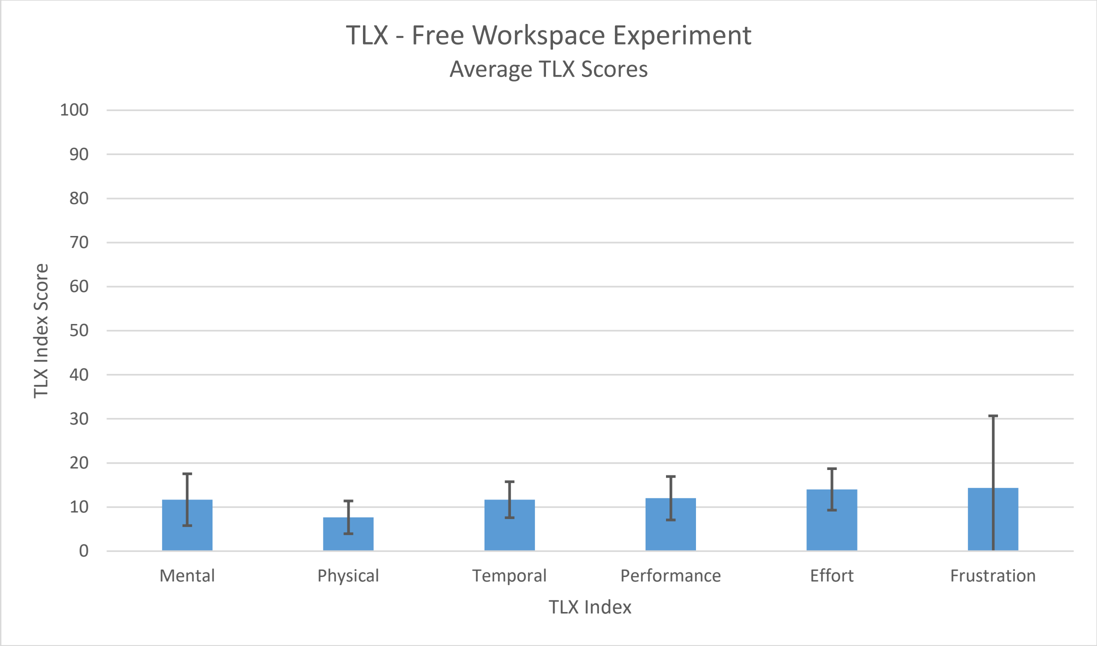


Figure 4.5: Average TLX scores for the first experiment

and no redirection was provided in the case of collision.

The main objective for this experiment set up was to provide, together with the first experiment, a baseline for comparison. The main interest in this particular scenario was the number of times the users would have entered a forbidden region and how that would affect their completion times and experience of the task. Even though the experiment task remains the same, the presence of invisible obstacles the users cannot avoid intentionally is a very frustrating element, and it has been confirmed by the TLX responses and personal interviews, shown in Figure 4.6. The frustration mainly comes from the system changing suddenly from normal operation to unresponsiveness due to the absence of redirection. Once the robot TCP enters in contact with a virtual fixture, the only directions allowed by the systems are those that resolve the collision state and do not have velocity components that would further penetrate the forbidden region being in contact with. This situation makes the forbidden region normal at the point of contact the only viable direction of movement to resume normal operating conditions. Since no feedback are provided from the system, the users have adopted the strategy of "sensing" the shape of the obstacles by moving the robot into collision multiple times and slowly trying to move past an obstacle.

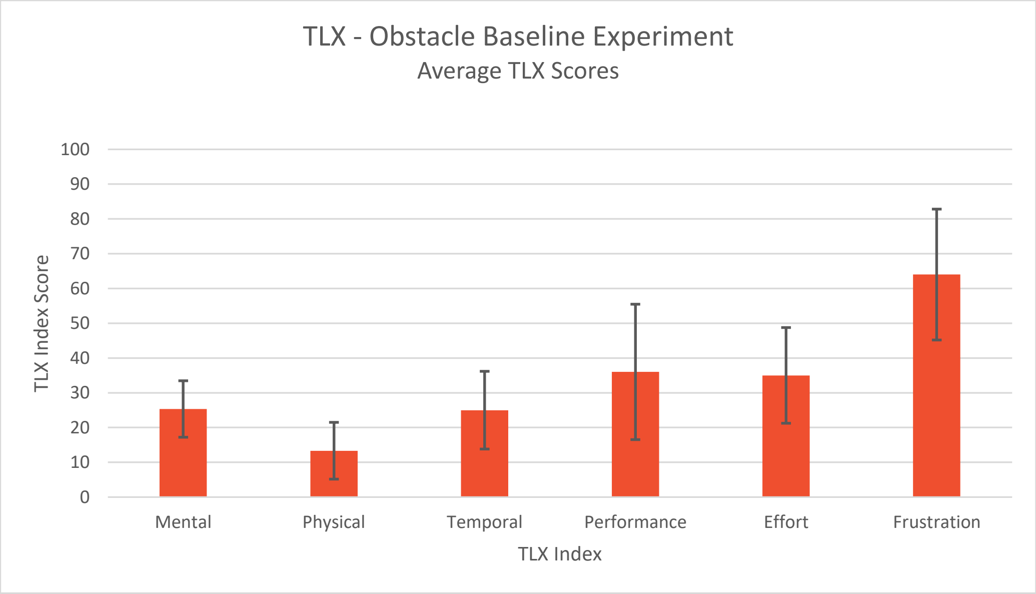


Figure 4.6: Average TLX scores for the experiment with obstacles only

## 4.5 Scenario C: Obstacles and Virtual Fixtures Redirection

The third experiment setup introduces virtual fixtures redirection, to smoothen the robot motion when it comes in contact with a forbidden region. The average completion time decreases in this experiment scenario mostly due to the fact that the robot motion now is not stopped if an obstacle is touched. Instead, the redirection filters the user velocity input and makes the robot "slide" along the obstacle surface. The resulting motion is not the one the user expects, but the discrepancy between user intent and robot motion is less compared to the previous scenario thanks to the redirection factor.

It is also interesting to see how the redirection affects the number of pseudo-violations (NPV) and corrective actions (NCA). By looking at the graph in Figure 4.7 it can be noticed the difference between the experiment with obstacles, when redirection is enabled and when it is disabled.

The reader may expect to see the number of pseudo-violations NPVs reduce to zero after the introduction of the VF redirection, but as it is shown in the graph that is not the case. It is true that every time the robot is in a collision state, every change of direction detected by the system, that is filtered by the control framework, will count to the total number of corrective actions. This means that the system will only increase the NCA count by one if the user, while being in a collision state, does not change the direction of motion of the monitoring robot.

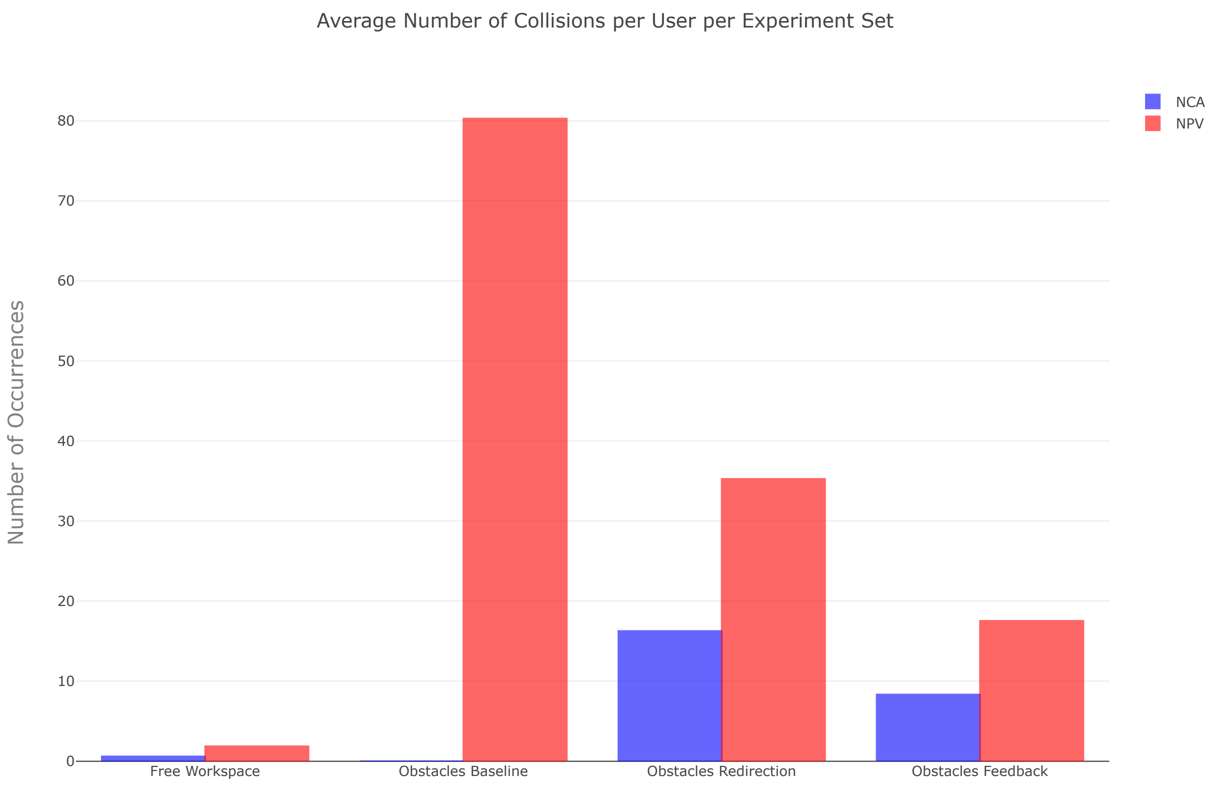


Figure 4.7: Comparison of the number of corrective actions (NCA) and pseudo-violations (NPV) between the different experiments. ©2020 IEEE

Every time the user decides to move in a different direction and the system has to redirect the monitoring robot velocity due to a collision state in place, then the total number of NCA will increase by exactly one per change of direction. The system can detect any change of direction that is slower than 5ms, as this limit is set by the Unity3D frame-rate update frequency of 200 fps. However, it is still possible to "collide" with an obstacle and not trigger the VF redirection if the user motion is already aligned with the tangent plane at the point of collision. This event occurs for example every time the user experiences the "sliding" motion along an obstacle surface, as simplified in Figure 4.8: in the last frame before leaving the obstacle surface, the user motion vector lies in the obstacle's tangent plane and therefore the action does not add to the NCA count, but it does add to the NPV count.

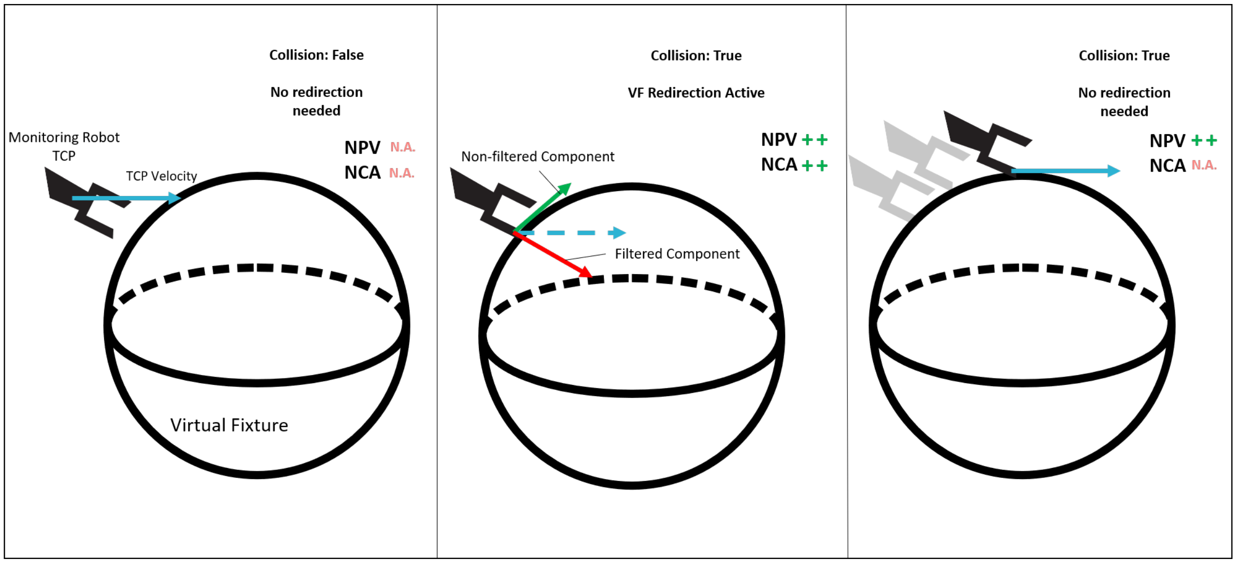


Figure 4.8: Representation of the sliding effect caused by the redirection and its effect on the metrics of NCA and NPV

It is possible to notice in the experiment's TLX scores, shown in Figure 4.9, that the frustration index has decreased significantly compared to the scenario where redirection was not enabled. The other indexes also appear to be on the lower part of the possible range, which is a general indication that users felt they accomplished the task successfully with relatively low effort and frustration.

## 4.6 Scenario D: Obstacles, Virtual Fixtures Redirection, Visual and Haptic Feedback

In the fourth experiment scenario a visual haptic component is introduced to signal the presence of nearby obstacles and when the user is colliding with them.

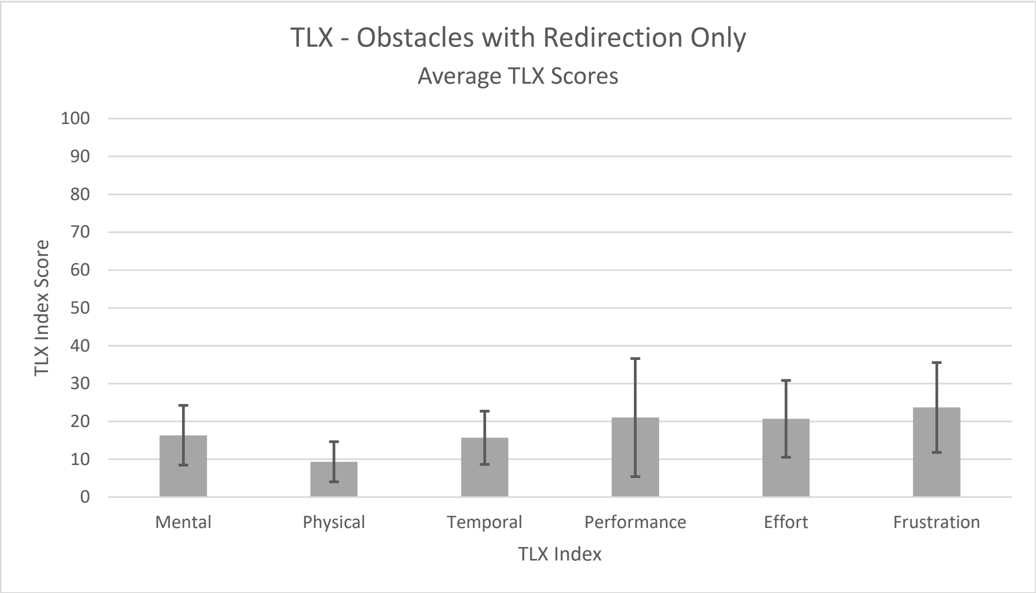


Figure 4.9: Average TLX scores for the experiment setup with obstacles and VF redirection only

The feedback is provided together with VF redirection and the goal is to assess whether the feedback helps improve the user performance and decrease the frustration factor that characterized the third experiment. The visual feedback consists in a 3D arrow icon displayed in the bottom right corner of the user interface, which points at the nearest obstacle along the direction of movement, even if not in the user's field of view. If the user collides with an obstacle, the arrow turns red and points at the point of contact, so that the user understand where the collision took place and which direction he or she should try to take. At first glance, the extra modality seems to not provide sensible effects in terms of average completion time, neither positive or negative. However, by observing the average collisions during the experiment from Figure 4.7 it can be noticed that the extra feedback modality contributes to decreasing the combined number of NCAs and NPVs (this will discussed further in 4.7).

The result of the TLX questionnaires for this scenario are shown in Figure 4.10: the overall feedback provided by the users is similar to the scenario C experiment, with relatively low effort required to accomplish the task and limited frustration associated with the obstacles, redirection and feedback combination. A more detailed comparison of the TLX scores is treated in section 4.7 where it will be discussed the effect of the extra feedback modality on cognitive load and what can be concluded from the statistical analysis.

## 4.7 Comparison Among Experiments

Now that the experiment scenarios have been individually described, it is important to compare the metrics and TLX scores to investigate what are the effects of the VF redirection and additional feedback on the users performance and difficulty evaluation.

The first graph that is worth introducing is the average completion time CT per experiment scenario, shown in Figure 4.11: it is expected to notice an increase in the average CT when obstacles are introduced in the workspace. However, a decrease in the average CT can be observed in the scenarios where redirection is introduced and also where haptic and extra visual feedback is provided.

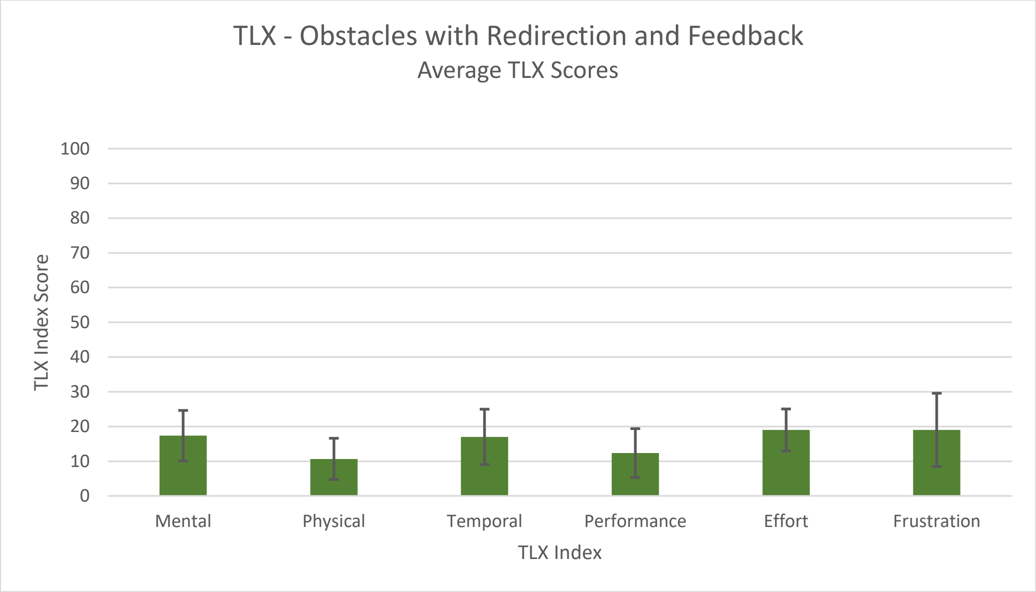


Figure 4.10: Average TLX scores for the experiment setup with obstacles, VF redirection and feedback

The one-way analysis of variance (ANOVA) determined that there is a statistically significant difference between the groups ($F(3,237) = 4.74$, $p = 0.0031$). The subsequent post hoc tests were performed on the pairs: (obstacle baseline, obstacle redirection) and (obstacle baseline, obstacle feedback). The first post hoc test does not highlights a statistically significant difference between the scenario where redirection was disabled and when it was active ($P = 0.36$), however there is a statistically significant difference between the former scenario and when both redirection and visual feedback are introduced ($P = 0.0039$). It is true that even with redirection and visual and haptic feedback to overcome the difficulty of

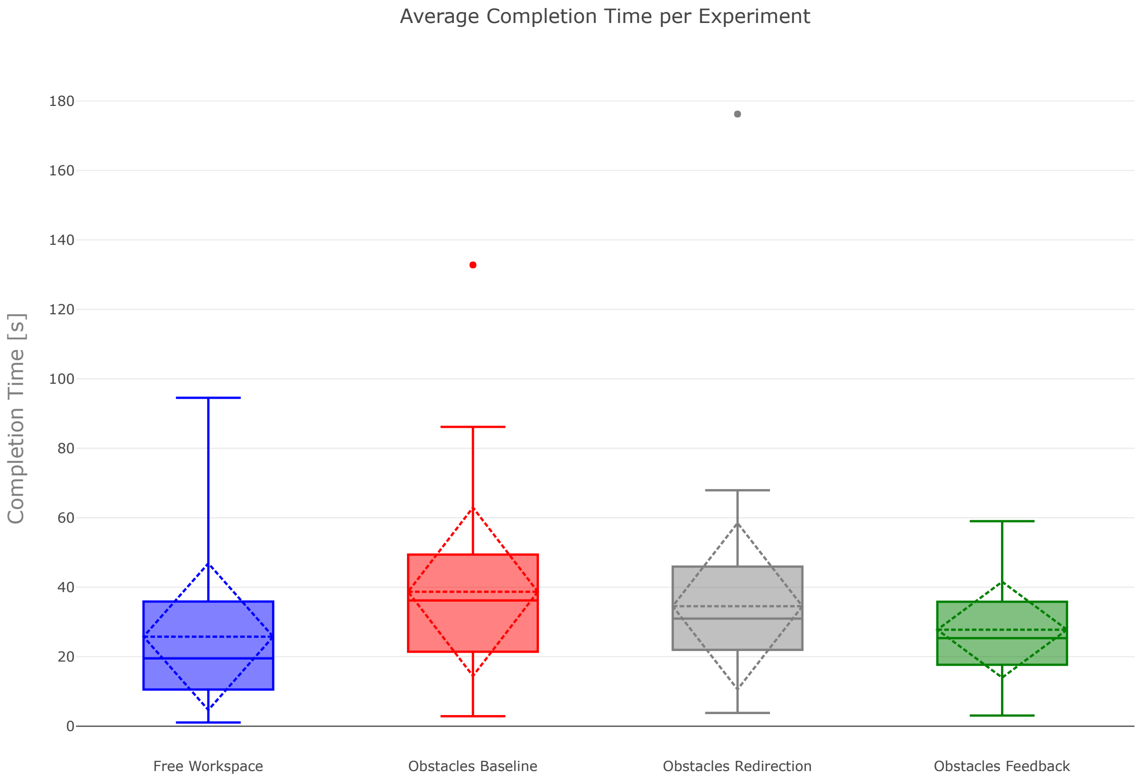


Figure 4.11: Comparison of completion times among the four experiment sets. ©2020 IEEE

the obstacles, the average completion time increases from the scenario with free workspace. However, it is very interesting that the combination of redirection and feedback provide a relative improvement in accomplishing the task in terms of completion time.

To measure the perceived complexity of the task in each scenario, the users had to fill the TLX form: comparing the task load indexes will help understand the effects that the redirection and the additional feedback had on the perceived difficulty of the experiment and also provide insights for the usability evaluation. The first index that is interesting to notice is the average of the perceived effort that the users reported after completing their respective experiment tasks, shown in Figure 4.12.

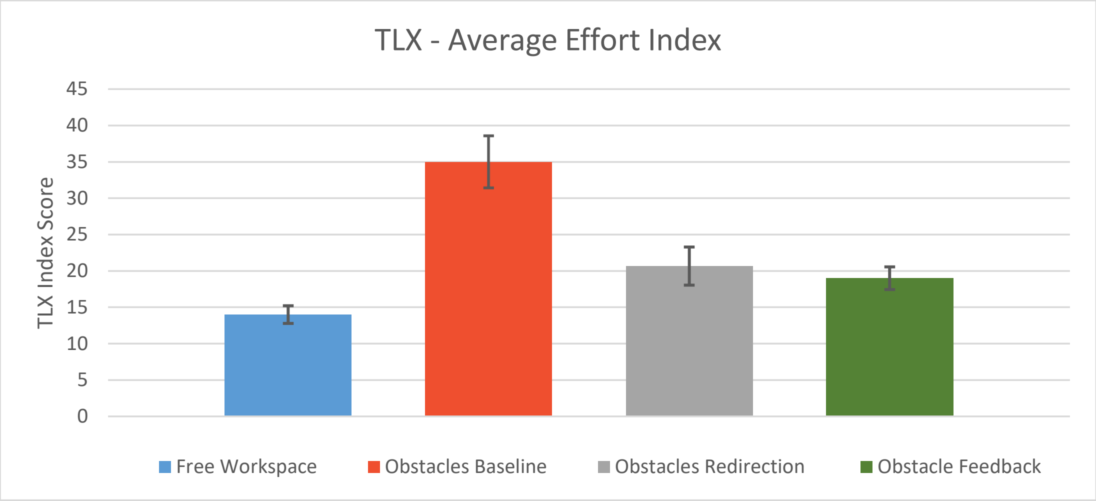


Figure 4.12: Comparison of the effort TLX scores among the four experiment sets. ©2020 IEEE

From a quick glance at Figure 4.12 it can be seen that the average effort decreases from an only obstacles scenario when redirection and additional feedback are introduced, remaining however still greater than the first scenario of the free workspace. This pattern has also characterized the CT analysis, and still positively hinting that although obstacles are increasing the difficulty of the task, redirection and additional feedback contribute in making the task easier for the user. Performing the one-way ANOVA on the users effort scores, it can be stated that there is statistically significant difference between the groups ($F(3,56) = 13.89$, $p = 6.98 \cdot 10^{-6}$). Furthermore, the post hoc tests showed that there is a statistically significant difference between the *Obstacle Baseline* scenario and *Obstacle Redirection* ($P = 0.003$), but not significant enough between *Obstacle Redirection* and *Obstacle Feedback* ($P = 0.59$). This latest result is likely to indicate that the VF redirection plays the bigger role in decreasing the perceived effort to accomplish the task. It is a positive finding in that the most difficult part of the task consisted in avoiding the invisible obstacles, rather than just controlling the monitoring robot to reach a different point of view. The additional feedback did not seem to increase the overall mental load according to the results of the TLX: the mental, physical and temporal scores remained fairly consistent across the different experiment, suggesting that the task was not "rushed", was not demanding in terms of physical abilities, nor was requiring high problem solving capabilities which is still consistent with the intention of the experiment scenarios. However, the additional visual and haptic feedback did have an effect on the performance score. In particular, there is statistically significant difference between the different experiment performance scores ($F(3,56) = 10.89$, $p = 9.75 \cdot 10^{-6}$), with the graph comparison shown in Figure 4.13.

The most interesting detail is that in this case, scenario C (*Obstacle Redirection*) and scenario D (*Obstacle Feedback*) present a statistically significant difference ($P = 0.035$), and at the same time the performance reported in the scenario

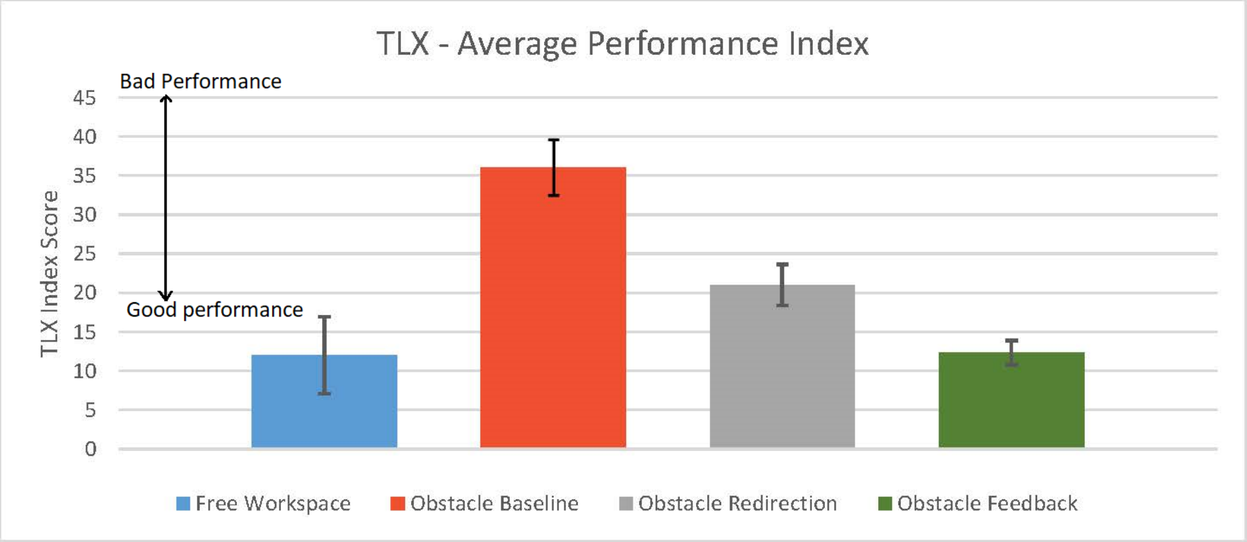


Figure 4.13: Comparison of the performance TLX scores among the four experiment sets. ©2020 IEEE

Table 4.2: Statistically meaningful difference from the analysis of variance (ANOVA)

| | **Statistical Difference** | | |
|---|---|---|---|
| **Scenarios** | CT | Effort | Performance |
| *Obstacle Baseline vs. Obstacle Redirection* | X | X | - |
| *Obstacle Redirection vs. Obstacle Feedback* | - | - | X |

D, where the additional feedback was provided, is very close to the average performance reported in the free workspace of scenario A. Combined with the previous finding, this analysis suggests that the monitoring robot navigation is actually improved by the VF redirection, and the additional feedback has the effect of making the users feel more efficient at accomplishing the task: if this effect cannot be concluded by looking at the completion times alone, it resonates in the decrease of NCA and NPV thanks to the presence of the additional visual clue and haptic feedback of the duration of the collision with an obstacle.

## 4.8 Reflections on an Industrial Application

The usability evaluation aims at answering the questions posed at the beginning of this chapter, with laboratory experiments tailored to have as many controlled variables as possible in order to narrow the search for the key elements in the VF control framework.

However it is important to mention that the development of the monitoring robot and the VF redirection control finally aims at addressing challenges that are faced in the welding industry during robotised welding. The usability evaluation has a key role in understanding what were the effects of the proposed control framework and monitoring solution in a laboratory setup that replicates part of the difficulties encountered during and actual industrial case.

In an actual robotised welding task, the monitoring task is typically made more difficult by the noise introduced by the welding equipment and the variability of the welding process itself, making it hard to control in a laboratory setup when a broader user study needs to be conducted. However, the problem of achieving a different point of view in the workspace monitor the process without interfering with the welding system is the most crucial part when an external monitoring unit is used, as it is proposed in this research, and that has been the main focus of the laboratory setup in this investigation.

The challenges related to the quality of the sensors and their efficacy in different welding application has not been treated in this research, as the proposed system and the findings of this chapter are applicable regardless of the hardware equipment that is provided to the monitoring unit.

## 4.9 Summary

This Chapter examined the effects of different elements in the monitoring framework on users' performance and the relationships with their task evaluation. Four different experiments were carried out, each experiment with 15 users that were instructed to navigate the monitoring robot to find an LED target multiple times. Each scenario introduced an additional element in the system that affected the navigation, either negatively (like the invisible obstacles) or positively (like VF redirection and additional visual and haptic feedback). Following the experiments with a statistical analysis of the user responses and metrics, it was observed that the VF redirection affects positively the navigation of the monitoring robot. The average CT showed a meaningful decrease from the scenario with obstacles only to the scenarios where obstacles were present but redirection was enabled.

Moreover, the extra feedback modality affected the performance score from the TLX form, with the most statistically significant difference. The results indicate that the additional visual feedback, together with haptic information about the duration of the collision with an obstacle, positively affects the user's performance, and in the presented form are suitable candidates for the type of navigation task that the monitoring robot is expected to carry out in an actual industrial setup.

# 5 Conclusion and Future Work

## 5.1 Thesis Conclusions and Contributions

This thesis has presented the investigation and development of a VF control framework of a robotic unit for monitoring an industrial robotic process. The research focused on the selection of the control method for the monitoring robotic unit, keeping in mind the user-in-the-loop design of the final application. By enabling remote monitoring of a robotic process, allowing different viewpoint selection without interfering with the main industrial task, the system helps address the challenges related to navigation and camera placement during teleoperation.

The background presented in Chapter 2 was an aid to the selection of the virtual fixture approach and helped in understanding the challenges related to teleoperation and the camera placement problem; it examined existing methods used in teleoperation and existing systems used in the industry for remote monitoring with their advantages and disadvantages in order to answer the first research question posed in Chapter 1: "What are the challenges related to monitoring a system with an external robotic unit?".

As summarized in Section 2.6, the literature review has played a crucial role in grounding the main features that the monitoring strategy had to possess and the rationale behind each design choice. As a result, in this thesis a VF based approach has been developed with an admittance type of control for the tele-operation of the monitoring robot. The combination of virtual fixtures and admittance control allows for a more precise position control and stable behaviour during free-motion, as well as a smooth transition from unconstrained to constrained motion.

One of the contribution of this thesis has been the development of a VF control framework, presented in Chapter 3, in answer to the research question "Is it possible to provide flexible viewpoint selection independently of the task that is being performed?".

With an external robotic unit being in charge of the monitoring task, the selected control method was able to tackle the additional challenge of navigating the workspace without interfering with the task robot when a user was manually controlling the monitoring unit. The safety of the navigation of the monitoring unit and the task robot is ensured by the combined action of the 3D environment and the VF based control. The virtual fixtures and the varying compliance are able to influence the robot motion by filtering out the velocity components input by the user that are in violation of a forbidden region. By defining forbidden regions, or virtual fixtures, around the workpiece, as well as the task robot and relevant obstacles, the user can navigate the monitoring robot in the workspace without having to consider how to avoid colliding with such objects. The proposed approach has been chosen to obtain a safer system response in case the monitoring robot violates a forbidden region with a part its body or in the case of sudden input from the user side: filtering out components of the monitoring robot velocity does not introduce energy into the system and results in a damped motion, even in the case where connection with the monitoring unit is compromised.

The thesis also posed the question "What are the design choices to have a seamless transition between an autonomous mode of operation and a mode that allows for user input?". In order to answer that question, Chapter 3 detailed how the VF control approach is adopted with the autonomous navigation part and how the admittance type of control has been preferred in a user-in-the-loop design.

Moreover, Chapter 4 presented and discussed the results of the usability evaluation of the VF control for the monitoring robot when solving a laboratory task that required the user to navigate around a test-piece and avoid invisible barriers.

Four experiments have been carried out, where 15 users in each scenario had the task to find the only lit LED in a test piece. Navigating the monitoring robot to find the LED was the main challenge during the experiments. This served to evaluate how demanding the navigation of the monitoring unit can be in the presence of obstacles.

A statistical analysis of the results showed that, compared to an obstacle free workspace, the VF control adds little extra load to the users while improving their performance in terms of completion time and number of collisions. By analysing the TLX scores from each experiment scenario, the virtual fixture redirection decreases the frustration of the users while not impacting their perceived effort or performance in a negative way.

Another contribution of the thesis has been the evaluation of additional navigational feedback and its impact on user performance and cognitive load. The trade-off between multi-modal feedback and cognitive load is a very important aspect in teleoperation, and in this case it has been important not to overload users with information about the navigation of the monitoring robot. As a reminder, the operator's task in an industrial scenario is to be able to reach the different view point to inspect the process or workpiece, and the navigation of the monitoring robot itself should not overload the user with information that would limit his/her capacity to observe the industrial process being carried out by the task robot.

## 5.2 Limitations and Future Work

One of the most important limitations in the current monitoring system is its difficulty to cope with limited degrees of freedom and to avoid reaching configurations where no motion is allowed due to joint limits. This problem is mainly due to the absence of a planner when the user is in control of the monitoring robot, and even if the system can advise the operator to perform a reconfiguration that action could sometime violate forbidden regions. To overcome this limitation the monitoring robot should have 7 or more degrees of freedom, as is typically the case, for example, for equipment used in robotised surgery. With more degrees of freedom available, the VF redirection could be applied without concerns for the available DOF-s without the problem of having the monitoring robot unable to move.

A topic for future work is the development of an intelligent agent for the autonomous tracking part of the control framework. If on one hand the virtual fixture redirection can be left enabled for both the manual and autonomous navigation, on the other the autonomous tracking algorithm cannot influence the user path if a higher level abstraction is made on what view points may be more interesting for the task being monitored. Thanks to some of the techniques developed by researchers to solve the camera placement problem, more knowledge on the most suitable viewpoints for a given workpiece could be "learned" or integrated into the system to achieve higher autonomy.

Moreover, a more intelligent logic for the navigation could improve the transition from manual mode to navigation and restore the path that the monitoring robot would have tracked as if the user hadn’t taken over with manual input. The system can then be tested in a similar laboratory setup for a dynamic task with a moving workpiece. The evaluation will have to consider first the dynamic motion only with no obstacles and then gradually more complex obstacle scenarios would be introduced just as it has been discussed and presented in Chapter 4.

Such improvements will eventually lead to an increasingly more autonomous monitoring unit that can serve the operator better, with less and less need for overriding manual actions, so that the operator can focus on the inspection task. Finally, the prospect of lifting more of the cognitive load from the operator task of controlling the monitoring unit could pave the way for more options in multimodal communication, so as to convey more information about the ongoing process itself.

# Publication Abstracts

## ROS-Unity3D Based System for Monitoring of an Industrial Robotic Process

Planning and monitoring the manufacturing of high quality one-of-a-kind products are challenging tasks. In the implementation of an industrial system, the commissioning phase is typically comprised of a programming phase and an optimization phase. Most of the resources are commonly invested in the optimization of the process. The time and cost of the implementation can be reduced if the monitoring system is not embedded in the industrial process, but kept instead as a decoupled task. In this paper we present a framework to simulate and execute the monitoring task of an industrial process in Unity3D, without interfering with the original system. The monitoring system is made of external additional equipment and is decoupled from the industrial task. The monitoring robot's path is subject to multiple constraints to track the original process without affecting its execution. Moreover, the framework is flexible thanks to the Unity-ROS communication so that the monitoring task can be carried on by any ROS-compatible device. The monitoring system has been applied to a robotic system for heavy, multi-pass TIG welding of voluminous work-pieces. The results of the implementation show that

the constraints for monitoring were satisfactory in the 3D environment and capable for real robot application. (Sita *et al.*, 2017)

## Robot Companion for Industrial Process Monitoring Based on Virtual Fixtures

In this paper, the use of a monitoring companion is proposed to more efficiently collect the process information during the tuning of industrial systems, and therefore to assist the system integrator in the optimization process for complex robotic installations. The monitoring companion consists of an industrial manipulator (monitoring robot) and a Unity-ROS framework for controlling it. We discuss in this paper the features that allow the solution to be re-used in different industrial application thanks to its compatibility with ROS. In particular, we present the approach based on admittance virtual fixtures and how we use such abstraction to track a moving target (it could be the end-effector of another robot for example). Moreover, the concept of varying compliance is introduced as a way to influence the motion of the monitoring robot on the virtual fixtures in the presence of obstacles. The experiments have been conducted in simulation as well as on real hardware to test the accuracy of the system at respecting the virtual fixtures with both a static and a moving monitoring target, although in the current implementation the varying compliance was not included in the experiments. (Sita *et al.*, 2018).

# Usability Study of a Robot Companion for Monitoring Industrial Processes

In this paper we present the findings of a usability study for a monitoring robotic unit tele-operated via a virtual fixtures (VF) based control framework. The study aims at investigating the impact of VF on the robot navigation as well as the impact of multimodal feedback on the user performance in a static inspection task. The findings will help in the design of the monitoring control framework to inspect a robotised welding process, as it has been researched in previous work. The study has been conducted with untrained participants, involved in four (4) different test scenarios. The experiments treated a static case in which users were asked to navigate the monitoring robot in the workspace to find a lit LED of a test-piece. The statistical analysis of the experiment metrics showed a positive impact of the VF control on the navigation of the monitoring robot even for users with no previous experience. Moreover, from the analysis of the task load index forms (TLX) it emerged that the combination of VF control and additional multimodal feedback improved the user performance without negatively impacting the effort required to accomplish the task.

# A Appendix

- 1st document: ©2017 IEEE. Reprinted, with permission, from E. Sita, C. M. Horváth, T. Thomessen, P. Korondi, T. Pipe, ROS-Unity3D Based System for Monitoring of an Industrial Robotic Process, 2017 IEEE/SICE International Symposium on System Integration (SII), December 2017.
- 2nd document: ©2018 IEEE. Reprinted, with permission, from E. Sita, T. Thomessen, A. G. Pipe, F. Dailami, M. Studley, Robot companion for industrial process monitoring based on virtual fixtures, IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society, October 2018.
- 3rd document: paper accepted for publication in ACIRS 2020 and archived in the IEEE Xplore database. E. Sita, T. Thomessen, A. G. Pipe, M. Studley, F. Dailami, (2020). Usability Study of a Robot Companion for Monitoring Industrial Processes.
- 4th document: TLX form for compilation used in the usability study.

# ROS-Unity3D Based System for Monitoring of an Industrial Robotic Process

Enrico Sita[1], Csongor Márk Horváth[2], Trygve Thomessen[3], Péter Korondi[2], Anthony G. Pipe[1]

***Abstract*—Planning and monitoring the manufacturing of high quality one-of-a-kind products are challenging tasks. In the implementation of an industrial system, the commissioning phase is typically comprised of a programming phase and an optimization phase. Most of the resources are commonly invested in the optimization of the process. The time and cost of the implementation can be reduced if the monitoring system is not embedded in the industrial process, but kept instead as a decoupled task. In this paper we present a framework to simulate and execute the monitoring task of an industrial process in Unity3D, without interfering with the original system. The monitoring system is made of external additional equipment and is decoupled from the industrial task. The monitoring robot's path is subject to multiple constraints to track the original process without affecting its execution. Moreover, the framework is flexible thanks to the Unity-ROS communication so that the monitoring task can be carried on by any ROS-compatible device. The monitoring system has been applied to a robotic system for heavy, multi-pass TIG welding of voluminous work-pieces. The results of the implementation show that the constraints for monitoring were satisfactory in the 3D environment and capable for real robot application.**

## I. INTRODUCTION

For complex industrial processes that produce high quality, one-of-a-kind products, planning is one of the most time-consuming phases. However, during the execution of the process constant and accurate monitoring is necessary to ensure that what has been planned, simulated and tested is accurately reproduced on the real workpiece. The methods of classical automation are therefore not suitable for this kind of production. Processes are difficult to adapt, and the complex commissioning phase prevents companies to react to these market demands in time. Automation and industrial robotics however allow to automate such complex processes while maintaining a high level of flexibility. While there are many different industrial applications that reflect this scheme, [1], [2], [3], in this paper we will focus on the particular processes of heavy grinding and welding as they have been used as test cases for our implementation.

In the industry, for both rigid and flexible automation solutions, the commissioning phase remains one of the most expensive parts of the process [4], [5]. During the commissioning phase the optimization is the most time expensive task, where the process is tuned and calibrated with respect to all the controllable parameters. If the optimization phase was too much time consuming, it would be hardly justified for *one-of-a-kind* types of production since the investment wouldn't be supported by a high-volume production. This is one of the reasons why it is not feasible to embed the monitoring system in the industrial process itself. Moreover the monitoring process is mainly used to optimize the industrial system, and with an independent monitoring solution it can be re-used on other applications once it fulfils its purpose.

The framework described in this paper aims at providing the tools to simulate and test the monitoring strategy with an industrial robot for heavy welding and grinding task, in order to shorten the time of the optimization phase and make the commissioning phase more efficient. Figure 1 shows the current setup for the welding task in our lab.

It is also important to notice that the remote monitoring system serves also as a means to provide external assistance on systems which are fully operational. Therefore, after the commissioning phase a system might benefit from external monitoring to evaluate the product quality even though it was not originally designed to include such a system.

Nonetheless, the framework and the results presented in this work are not to be seen as constrained to heavy welding and grinding processes. In fact, the monitoring system described in this paper is independent of the robotic

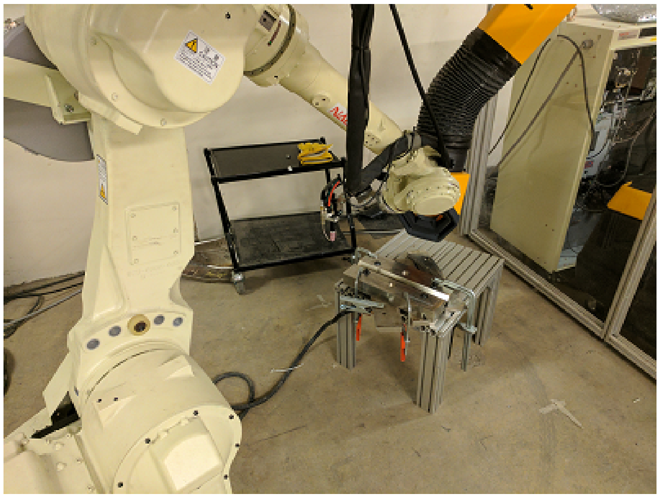


Fig. 1. Lab set-up of the welding task. The welding robot is a NACHI MC70. The actual monitoring robot is not shown in this set-up.

---

This work was supported by PPM AS and The Research Council of Norway.

[1] E. Sita and A. G. Pipe are with Department of Engineering Design and Mathematics at Bristol Robotics Laboratory, University of the West of England, BS16 1 QY, UK `enrico2.sita@live.uwe.ac.uk`, `tony.pipe@brl.ac.uk`

[2] C. M. Horváth and P. Korondi is with Department of Mechatronics, Optics and Mechanical Engineering Informatics, Budapest University of Technology and Economics, 1111 Budapest, Hungary `hcsongorm@mogi.bme.hu`, `korondi@mogi.bme.hu`

[3] T. Thomessen is with PPM AS, Leirfossvegen 27, 7038 Trondheim, Norway `trygve.thomessen@ppm.no`

hardware and also independent of the industrial process being monitored.

Moreover, the recent progress made in the fields of Augmented Reality (AR) and Virtual Reality (VR) is reshaping the future of human robot interactions ([6], [7], [8], [9]).

Multi-modal feedbacks can now be more easily combined to communicate much more information about the real system and also to provide more immersive experiences when we interact with the digital environment [10]. In fact, the gap that exists between the user's actions in the digital world and their effect on the real system is slowly but steadily narrowing. Concretely, the intention to later integrate in our framework multi-modal interactions for man-machine communication is reflected in the choice of Unity [11] as the component in charge of displaying the digital models and interface with the user. Although not the only possible option, the game engine Unity is supported by an active community and a well grounded ecosystem, besides being among the top choices for AR/VR development [11]. In Unity we design the user interface and the monitoring process in a 3D environment before having the task performed by a real robot. In contrast to a simulation software (e.g. Gazebo [12]) Unity3D is not conceived to accurately reproduce real world scenarios in terms of dynamics and physics, but rather to visualize models in a 3D environment and interact with them just as it is the case in games development. The main advantage thus is in the flexibility to design the user interface and the user's interactions with the virtual world.

The second core component is the well known Robot Operating System (ROS), [13] which is used to directly communicate with all the hardware components which are not directly involved in the UI design, such as robots and sensors. ROS allows the framework to interface with any ROS-compatible robot without the need to change the control strategy or the monitoring task.

In this paper we present a framework to simulate the monitoring task of an industrial process in Unity3D. The monitoring process is subject to multiple constraints in order not to interfere with the industrial task. Moreover, we discuss the flexibility of the framework that allows for easy deployment of the monitoring task on a real robot.

## II. RELATED WORK

Due to the flexibility we intend to give to our framework, there are many different research contributions that need to be acknowledged and approaches that have to be mentioned.

As previously stated, the industrial process of welding treated in this paper should not pose a limitation in the conclusions we want to draw. The framework is to be seen as independent of the particular robotic task being treated here, but we shall mention related work for similar processes in order to provide better context to the need of a monitoring system for the entire process.

In contrast to high-volume fully automated production systems where monitoring is mainly used for quality control, in smaller batches production systems there is often the need for continuous monitoring. In [14] Pfeifer et al. describe the advantages of an inspection system along a micro-assembly line. Buschhaus et al. present in [15] monitoring of a robot-based process for the metallization of three dimensional molded interconnected devices. The monitoring is crucial in this processes as it serves as in-line correction method in order ensure high quality results. The implementation of a monitoring task for a robot welding application is a continuation of the concept presented in [16]. In their work, Zimber et al. discuss how an autonomous industrial manipulator (AIMM) can be used for monitoring an industrial process. Ultimately, due to the increasing demand from SMEs of remote support solutions for their industrial robotic systems, a monitoring AIMM can significantly improve remote maintenance and assistance. The separate solution brings significant advantages, since it doesn't interfere with the industrial process. In fact, if additional sensors have to be integrated in the industrial process they have to be included in the design process and their presence can redefine the optimization process of the task itself. If additional equipment or additional support tasks are not included in the design process, it is possible that they cannot be integrated at all without an immense investment of resources. The monitoring robot can be programmed also *after* an industrial process optimization phase and it doesn't require modifications in the original system. It serves as a less expensive, quicker-to-integrate external equipment.

For what regards the ROS-Unity communication and its advantages, there are several papers that explored the potential of such connection and that in general investigated the potential of Unity for designing Human Robot Interface (HRI). The work of Bartneck et al. in [17] is one of the first papers advocating the user friendliness of Unity for the design of human robot interaction. One of the main arguments of the paper is that programming robot behaviours and interactions is easier in Unity due to the presence of a set of tools for animations and visual programming used in game development. However, in their work they decide not to involve any middle-ware solution for robotic hardware and implement all the communication logic and HRI within the Unity environment. Other works ([18] [19]) that followed explored further the possibility of using ROS as middle-ware solution, while still managing the HRI in Unity or similar software. The research works mentioned here are in the field of telepresence and teleoperation where multi-modal user interactions are essential. Nonetheless we mainly cite such contributions to highlight the flexibility and modularity allowed by the connection between Unity and ROS. In [20] Codd-Downey et al. proposed an architecture Unity-ROS to control a mobile robot in virtual reality.

Furthermore, Pan et al. [21] proposed an approach for simulating a robotic welding task in Unity. As previously mentioned, Unity is a game engine and not originally thought as a simulation software and therefore lacking proper tools to include the dynamics of the system and accurate hardware parameters. As the authors point out, such system could be beneficial for educational purposes and training.

From a slightly different perspective, we see the Unity environment as where, besides the user interactions, some of the higher level logic is processed and then communicated to the hardware through ROS or other dedicated channels. Concretely, by receiving real-time information about the state of the system we can animate the 3D environment accordingly and apply constraints on the monitoring robot's motion.

## III. SYSTEM DESCRIPTION

The framework can be divided into two parts:

- The industrial process (i.e. the robotic welding task)
- The monitoring process

The system in charge of performing the welding task is based on the work presented by Horvath et al. in [22] and shown in Fig. 2. Furthermore, the welding process data are communicated to ROS and thus made "accessible" to the monitoring process, which is based in Unity and communicating with hardware through ROS.

The general architecture of the whole framework is shown in Fig.3, where the connection ROS-Unity is the main bridge between the different subparts of the system. The monitoring process is directly linked to Unity as it is designed and implemented with the game engine. It is also worth mentioning that the monitoring task is linked to multi-modal man-machine communication to reflect the HMI design process that takes place in Unity. In fact, it is possible to integrate different types of feedbacks (tactile, audio, visual) into the same digital environment without the need of additional software. The monitoring task should be only partly automated, in the sense that the user should have the freedom to adjust the view to his/her needs without having to worry not to interfere with the welding process currently ongoing. In this context, multi-modal feedbacks can increase the user's comfort when he/she takes control over the monitoring robot. Furthermore, the connection with multi-modal communication also reflects our intent to eventually interface the system with VR/AR equipment to investigate immersive telepresence applications.

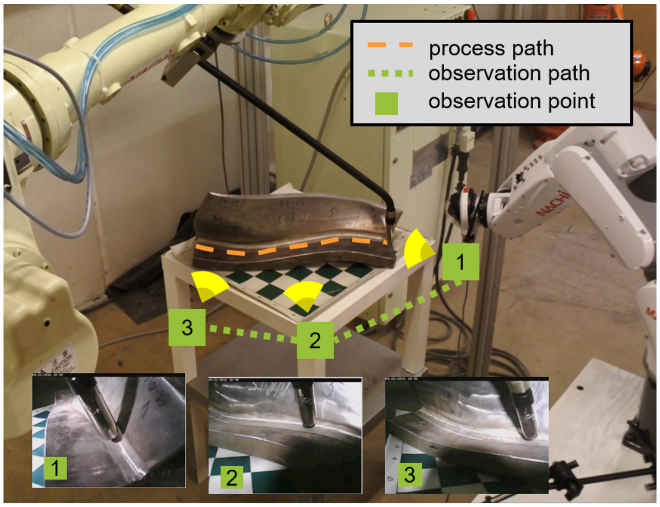


Fig. 2. The set-up of our lab with a welding process (orange dashed line) and the associated monitoring task with way points.

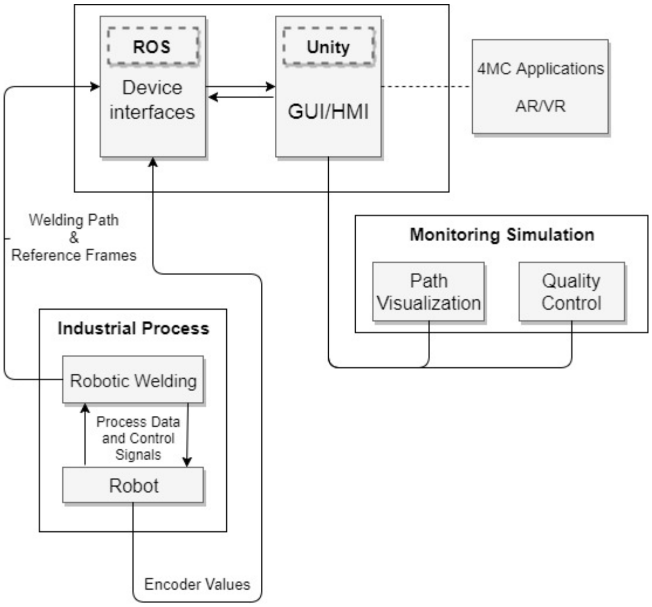


Fig. 3. Layout of the framework presented.

The 3D environment use for the simulation is created by importing the CAD models of the robot and the workpiece involved in the welding process. Such models are then placed in Unity along with the 3D model of the robot used for the monitoring task, as it is shown in Fig.4.

The welding application provides the data about the welding path, which can be displayed into Unity and visualized together with the workpiece. The reference frames of the welding robot, the torch and the workpiece are also communicated by the welding application to Unity through ROS. Once the welding path is available in Unity, the monitoring program calculates the path for the observation taking into account the torch orientation and the welding robot configuration in order to avoid collisions with the system.

The default monitoring strategy consists of simply following the welding process while keeping the welding torch and the part of the workpiece being machined inside the camera's field of view. However, sometimes this strategy is not the most desired one by the user, which should be then allowed to tweak and adjust the camera position if necessary. Therefore the Unity scene allows for user commands that modify the camera position and orientation while tracking the ongoing process.

It is then possible to observe in the 3D environment the welding robot moving according to the real-time joint values provided by the industrial process, furthermore the user can see through the camera view of the simulated monitoring robot and observe how the process while it's performed. The details of the actual implementation of the framework and the monitoring scene in Unity will be discussed in the following section.

## IV. IMPLEMENTATION

This section treats the implementation of some sub-parts of the system, mainly regarding the communication between all elements. The last part of this section describes how the animation/control of the 3D model is carried out in Unity.

### A. ROS

In the implementation of the system, the communication between Unity, ROS and the welding process is based on the following elements:

- C# Rosbridge for Unity-ROS. This script establishes the connection and allows for invoking Ros services
- Rosbbridge script for Welding process and ROS. This script allows the welding system software to publish data onto topics.
- Ros topics of the welding process. Currently the data published are welding path positions, workpiece reference frame and robot reference frame

Regarding this specific welding process the industrial system doesn't allow for external control commands, meaning that Unity can only fetch the real-time information to synchronize the monitoring task but cannot interfere with the ongoing welding operation.

### B. Joint Reading

It is thus important for the monitoring robot to receive at runtime the welding robot configuration, that is its joint values. The welding robot used in our system is a NACHI MC-50, while the robot model used for simulating the monitoring is a NACHI MZ-04 and each robot has 6 DOF. Although the welding robot is ROS-compatible, in our implementation we exploit a different communication channel to receive the robot's encoder values at runtime. In fact, we use a Rasperry-Pi connected to the robot to read the encoder values through UDP communication. The implementation details of such device are not the subject of this paper, but for the sake of clarity Unity receives precise encoder values of the welding robot through UDP communication. Even though the device allows for joint control, as previously mentioned this capability is disabled for Unity since the monitoring process is not allowed to control the welding task. The encoder values received in Unity are then converted into joint angles ([rad]) with double float precision. The conversion is defined by the formula:

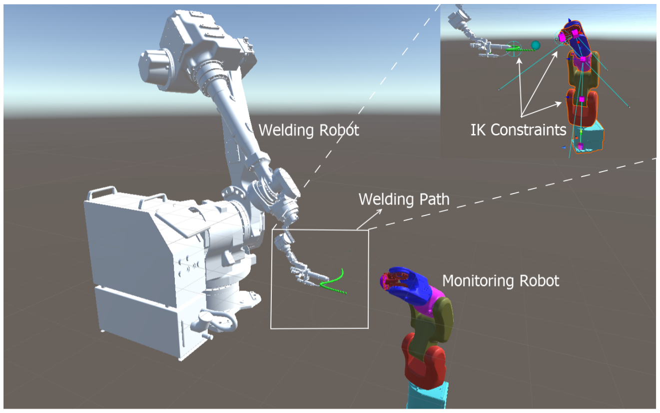


Fig. 4. The Unity environment and the 3D visualization of a welding path. The 3D model of the object has been hidden to better visualize the path. In the detail it is shown the monitoring robot with the visualization of the IK solver constraints.

$$\theta_i = \Theta_i^{offset} + (Enc_i - Enc_i^{offset})/\pi_i \tag{1}$$

where $i$ is the joint index, $Enc_i$ is the encoder value of joint $i$ received through UDP connection and the parameters $\Theta_i^{offset}$, $Enc_i^{offset}$ and $\pi_i$ are constants obtained from the robot controller beforehand. The update frequency of the encoder values has a limit of 5ms (both for writing and reading), which limits the Unity maximum frame rate to 200 fps. However, such a limit is more than enough for real time application and does not constitutes a bottle neck in our system.

In the 3D simulation we display the robotic cell thanks to the CAD models obtained from the industrial process. The digital welding robot is synchronized with the joint values coming from the real robot. We assume that the monitoring robot is equipped with a camera mounted on the end effector and we use the end effector's reference frame for the camera's orientation.

### C. Animation in the 3D Environment

When the welding starts, the monitoring robot starts tracking the welding torch by keeping it in the camera's field of view. The robot is animated via a c# IK solver based on the work presented in [23] and made available as a Unity-plugin.

In addition to the field-of-view constraint, the monitoring robot needs to take into account the following constraints:

- The distance from the torch must not be lower than a certain threshold;
- Avoid collision with the welding robot;
- Keep the welding torch in the centre of the camera;

Each frame, the encoder values received from the Raspberry-Pi are converted into radians and used to update the position of the 3D model. Then, the monitoring algorithm enforces the constraints on the 3D model of the MZ-04 and then compares the newly calculated joint values with the ones of the previous frame. If there is a difference between the two frames it proceeds by performing the inverse conversion to obtain the corresponding encoder values that the actual robot should reach. Finally, the new encoder values are sent through UDP to the Raspberry-Pi connected to the MZ-04. This process is summarized in Algorithm 1.

**Algorithm 1** Joint conversion and update in Unity

```
Input: MZ-04 Encoder Values
Output: Unity-generated Encoder Values
    Encoder Reading and Conversion
 1: for $i = 1$ to $n_{dof}$ do
 2:     Obtain $\theta_i$
 3:     Convert $\theta_i$ from radians to degrees
 4:     Update the 3D model of the $i$-th joint
 5: end for
 6: Enforce the IK-solver constraints
 7: for $i = 1$ to $n_{dof}$ do
 8:     Obtain the new $\theta_i$ in degrees
 9:     Convert $\theta_i$ from degrees to radians
10:     Obtain $ENC_i$ with the inverse conversion
11:     Store $ENC_i$ in the array $ENC_{new}$
12: end for
13: return $ENC_{new}$
```

In Algorithm 1, $n_{dof}$ is the number of joints of the monitoring robot. Moreover, it is worth observing how Step 6 of the algorithm implies that the IK-solver modifies the pose of the 3D model, according to the objectives that are active in that frame.

## V. RESULTS AND DISCUSSION

The system has been evaluated in a simulation conducted entirely in Unity. The model of the welding robot was programmed to move along a test path (see green line in figure 4, and three main objectives were set on the model of the monitoring robot: welding torch had to remain in focus (look-at constraint); maximum distance between the welding torch and the monitoring robot's end effector; collision avoidance with the welding robot. Every run consisted of the welding robot performing the path once (back and forth), while we observed the behaviour of the monitoring robot.

During the monitoring simulation the robot's configuration could occasionally fluctuate due to the multiple objective optimization. In fact, since the optimization algorithm is based on GA, the robot might move from its current configuration to one with a higher fitness. However, a monitoring simulation is considered successful when the main objectives presented in section IV are satisfied. This means that two successful simulations may have slightly different monitoring paths, but they both accomplish collision avoidance while keeping track of the welding torch and the workpiece. Given a specific instance of the objective, we are not interested in the global optimum within the given search space, but rather a sub-optimal solution in a limited time frame (since the search is computed at run-time).

The main reason why we considered different solutions acceptable is due to our intention to include also commands given by the user to control the monitoring view. Therefore, since in the future the monitoring path will be modified at runtime by the user's actions, the system must allow for some flexibility in the robot configurations.

In the welding task considered in this paper we did not incur situations were one or more of the objectives could not be satisfied. However, it is important to consider such cases to prevent unexpected behaviours from the monitoring robot. In fact, when not all objectives can be satisfied the robot might jump between configuration that optimize different objective that however share similar weights. In order to prevent these fluctuations, we decided to implement an agent in charge of supervising the IK solver at runtime. Concretely, in the event of configurations which do not fulfil one or more constraints this agent will add a special constraint to the optimization function of the IK solver.

The additional objective is called ”displacement objective” and its sole purpose is to punish all new configurations found which are ”distant” from the current one in terms of joint space. It is important to observe that the agent is also ensuring that the objective are satisfied with the same priority with which they have been listed in the previous section. This is achieved by changing the weights at runtime in a fashion that consistently reflects the aforementioned order.

Thus, with the assumption that the priorities are kept intact, the displacement objective ensures that robot is not ”jumping” to a new configuration which is significantly different from the current one, even if the overall fitness of the solution would improve.

The experiments in simulation show that the monitoring robot is capable of tracking the welding torch without specific knowledge of the welding path (the trajectory was only known by the welding robot model). In the bigger framework, it helps proving that such a model-based approach is suitable for remote monitoring of an industrial task.

In this work, the monitoring task has been implemented entirely in simulation, checking that the constraints were satisfied in the 3D environment. However, it is possible to implement the very same simulation on a real robot and this will be part of our future works. The intention is to exploit the UDP communication that has been used to synchronize the system with the industrial process, and use it this time for joint control of the monitoring robot. In this context we will conduct tests to assess the capability of the system to decrease the time for troubleshooting compared to a situation where monitoring was absent.

## VI. CONCLUSION AND FUTURE WORK

In this paper we presented a framework for robotic monitoring of an industrial process. The key achievements of this work are the following:

- Remote monitoring system for an industrial robotic process.
- Flexibility of the system due to ROS-Unity communication. The monitoring can be executed with any ROS-compatible hardware.
- Non-invasiveness of the remote monitoring. The parameters of the industrial process remain unmodified and the monitoring equipment can be introduced without compromising the welding task.
- Compact solution to set up a monitoring strategy. The monitoring robot is controlled in the same framework that provides the camera view.

The framework has been used for the planning and evaluation of the monitoring strategy on the welding application. One of the objective is to move toward a shorter set up time thanks to the decoupling from the original process. We are currently running tests in our lab in order to collect more data. The system provides a more flexible compared to an embedded monitoring solution that would have to be designed taking into consideration the welding path and the welding equipment, and that couldn't be re-used on different installations.

Finally, we aim at extending our framework for multi-modal man-machine communication (4MC) and VR/AR devices for remote monitoring.

## VII. ACKNOWLEDGMENT

This research is supported by The Research Council of Norway through the project 245691 Cognitive robot welding system (CoRoWeld), the Industrial PhD project 244972/O30 Virtual presence in remote operation of industrial robot and the Industrial PhD project 264145/O30 Interactive robot operation using multi-modal man-machine communication"

## REFERENCES

[1] D. Meike and L. Ribickis, "Energy efficient use of robotics in the automobile industry," in *2011 15th International Conference on Advanced Robotics (ICAR)*, June 2011, pp. 507–511.

[2] A. K. Sethi and S. P. Sethi, "Flexibility in manufacturing: A survey," *International Journal of Flexible Manufacturing Systems*, vol. 2, no. 4, pp. 289–328, Jul 1990. [Online]. Available: https://doi.org/10.1007/BF00186471

[3] F. Jovane, Y. Koren, and C. Bor, "Present and future of flexible automation: Towards new paradigms," *CIRP Annals - Manufacturing Technology*, vol. 52, no. 2, pp. 543 – 560, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0007850607602030

[4] H. Gattringer, R. Riepl, and M. Neubauer, "Optimizing industrial robots for accurate high-speed applications," *Journal of Industrial Engineering*, vol. 2013, 2013.

[5] Z. Pan, J. Polden, N. Larkin, S. Van Duin, and J. Norrish, "Recent progress on programming methods for industrial robots," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 87–94, 2012.

[6] Digi-Capital. (2017) Augmented/virtual reality report q3 2017. Http://www.digi-capital.com/news/2017/01/after-mixed-year-mobile-ar-to-drive-108-billion-vrar-market-by-2021/#.WXYHFYiGNPb. [Online]. Available: http://www.digi-capital.com

[7] S. Tachi, "Embodied media: Expanding human capacity via virtual reality and telexistence (keynote)," in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ser. ICMI 2016. New York, NY, USA: ACM, 2016, pp. 3–3. [Online]. Available: http://doi.acm.org/10.1145/2993148.3011261

[8] A. Israr, Z. Schwemler, J. Mars, and B. Krainer, "Vr360hd: A vr360&deg; player with enhanced haptic feedback," in *Proceedings of the 22Nd ACM Conference on Virtual Reality Software and Technology*, ser. VRST '16. New York, NY, USA: ACM, 2016, pp. 183–186. [Online]. Available: http://doi.acm.org/10.1145/2993369.2993404

[9] J.-L. Lugrin, D. Obremski, D. Roth, and M. E. Latoschik, "Audio feedback and illusion of virtual body ownership in mixed reality," in *Proceedings of the 22Nd ACM Conference on Virtual Reality Software and Technology*, ser. VRST '16. New York, NY, USA: ACM, 2016, pp. 309–310. [Online]. Available: http://doi.acm.org/10.1145/2993369.2996319

[10] T. Thomessen, M. Niitsuma, K. Suzuki, T. Hatano, and H. Hashimoto, "Towards Virtual Presence Based on Multimodal Man-Machine Communication: A Remote Operation Support System for Industrial Robots," *IFAC-PapersOnLine*, vol. 48, no. 19, pp. 172–177, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896315026531

[11] Unity3d.com. (2017) Unity 3d: Game engine. https://unity3d.com/ and https://unity3d.com/public-relations. [Online]. Available: https://unity3d.com/

[12] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3. IEEE, pp. 2149–2154.

[13] 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 *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

[14] T. Pfeifer and G. Dussler, "Process observation for the assembly of hybrid micro systems," in *Proceedings of 2002 International Symposium on Micromechatronics and Human Science*, 2002, pp. 117–123.

[15] A. Buschhaus and J. Franke, "Industrial robots accuracy optimization in the area of structuring and metallization of three dimensional molded interconnect devices," in *Robotics in Smart Manufacturing: International Workshop, WRSM 2013, Co-located with FAIM 2013, Porto, Portugal, June 26-28, 2013. Proceedings*, 2013, pp. 179–190.

[16] F. Zimber, C. M. Horváth, T. Thomessen, and J. Franke, "Control strategy for an industrial process monitoring robot," in *2016 IEEE/SICE International Symposium on System Integration (SII)*, pp. 706–710.

[17] C. Bartneck, M. Soucy, K. Fleuret, and E. B. Sandoval, "The robot engine making the unity 3d game engine work for hri," in *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2015, pp. 431–437, iD: 1.

[18] D. Krupke, L. Einig, E. Langbehn, J. Zhang, and F. Steinicke, "Immersive remote grasping: Realtime gripper control by a heterogenous robot control system," in *Proceedings of the 22Nd ACM Conference on Virtual Reality Software and Technology*, ser. VRST '16. New York, NY, USA: ACM, 2016, pp. 337–338. [Online]. Available: http://doi.acm.org/10.1145/2993369.2996345

[19] Y. Hu and W. Meng, "Rosunitysim," *Simulation*, vol. 92, no. 10, pp. 931–944, Oct. 2016. [Online]. Available: http://dx.doi.org.ezproxy.uwe.ac.uk/10.1177/0037549716666683

[20] R. Codd-Downey, P. M. Forooshani, A. Speers, H. Wang, and M. Jenkin, "From ros to unity: Leveraging robot and virtual environment middleware for immersive teleoperation," in *2014 IEEE International Conference on Information and Automation (ICIA)*, 2014, pp. 932–936, iD: 1.

[21] J. Pan, Y. Zhuo, L. Hou, and X. Bu, "Research on simulation system of welding robot in unity3d," in *Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry - Volume 1*, ser. VRCAI '16. New York, NY, USA: ACM, 2016, pp. 107–110. [Online]. Available: http://doi.acm.org.ezproxy.uwe.ac.uk/10.1145/3013971.3013982

[22] C. M. Horvath, T. Thomessen, and P. Korondi, "Robotized Multi-Pass Tungsten Inner Gas Welding of Francis Hydro Power Turbines." Edinburgh, Scotland, United Kingdom: IEEE IES, June 2017, pp. 1759–1765.

[23] S. Starke, N. Hendrich, S. Magg, and J. Zhang, "An efficient hybridization of genetic algorithms and particle swarm optimization for inverse kinematics," in *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2016, pp. 1782–1789, iD: 1.

# Robot Companion for Industrial Process Monitoring Based on Virtual Fixtures

Enrico Sita[1], Trygve Thomessen[2], Anthony G. Pipe, Farid Dailami, Matthew Studley

*Abstract*— **In this paper, the use of a monitoring companion is proposed to more efficiently collect the process information during the tuning of industrial systems, and therefore to assist the system integrator in the optimization process for complex robotic installations. The monitoring companion consists of an industrial manipulator (monitoring robot) and a Unity-ROS framework for controlling it. We discuss in this paper the features that allow the solution to be re-used in different industrial application thanks to its compatibility with ROS. In particular, we present the approach based on admittance virtual fixtures and how we use such abstraction to track a moving target (it could be the end-effector of another robot for example). Moreover, the concept of varying compliance is introduced as a way to influence the motion of the monitoring robot on the virtual fixtures in the presence of obstacles. The experiments have been conducted in simulation as well as on real hardware to test the accuracy of the system at respecting the virtual fixtures with both a static and a moving monitoring target, although in the current implementation the varying compliance was not included in the experiments.**

## I. INTRODUCTION

For system integrators, optimizing complex industrial robotic applications (e.g. robotised welding) is a difficult and time-consuming task. This is usually due to discrepancies between the models and the actual behaviour of complex systems, and the system integrator needs to fine tune the final installation by trial and error to obtain the desired quality. This procedure is even more tedious when the operator cannot access the robotic system once in operation and must rely on additional sensors to acquire the necessary process information. However, it is often difficult to find a permanent placement for the sensors to be able to fully monitor the process at any given time during the trials, and this would also be a very expensive and potentially unreliable approach, if applied to all of the robot installations. While it is hard to completely remove this trial and error fashion, it is possible to provide a way to gather process information more effectively that can be used in several robotic installations. It is then proposed to provide the system integrator with a monitoring robot in addition to the robot(s) belonging to the industrial process that needs to be optimized (also referred to as *task robot(s)*). The monitoring robot can be equipped with several different sensors and can be moved into close proximity of any installed robot so that it can be used to collect information from that process during and/or after the operation without interfering. The system operator can control the monitoring robot to change its viewpoint and acquire information from various positions (e.g. inspect a workpiece from different angles). With a more effective way of gathering process data, the system integrator can perform his/her primary task (optimizing the industrial process) more efficiently. Since controlling the monitoring robot is not a primary task, the challenge is to make such interaction as flawless as possible not to overload the operator. The operator will control the monitoring robot with a camera view from its endeffector and via a joystick or similar interface.

The concept and the framework to control the monitoring robot and synchronize it with the task robot has been previously discussed in [1]. This paper instead, focuses on the control strategy to navigate the monitoring robot inside the workspace, which is based on abstract surfaces, called *virtual fixtures* [2], [3] (Conceptual representation in Figure 1).

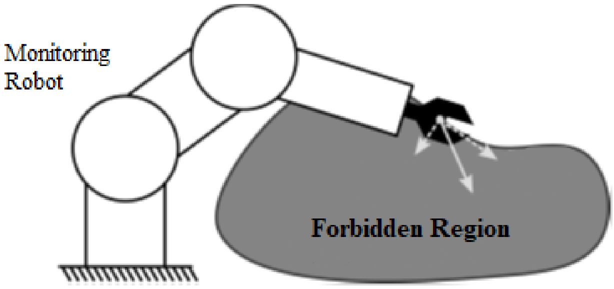


Fig. 1. Conceptual representation of a virtual fixture. The virtual fixture should prevent the robot's end-effector from entering a certain forbidden region.

## II. RELATED WORK

Other researchers have worked on the use of a robot to monitor another industrial process, as it provides flexibility in the choice of viewpoint angle in the workspace as well as allowing any inspection to be performed remotely (so improving EHS conditions when the industrial process is carried on in a harsh environment).

Carvalho *et al.* in [4] discuss virtual reality approaches to be able to inspect an Oil offshore platform, so as to improve understanding during simulation before moving to the real industrial site. Both advanced control techniques and virtual

This work was supported by PPM AS and The Research Council of Norway.

[1] E. Sita, A. G. Pipe, Farid Dailami and Matthew Studley are with the Department of Engineering Design and Mathematics at Bristol Robotics Laboratory, University of the West of England, BS16 1 QY, UK `enrico2.sita@live.uwe.ac.uk`, `tony.pipe@brl.ac.uk`, `Farid.Dailami@uwe.ac.uk`, `Matthew2.Studley@uwe.ac.uk`

[2] T. Thomessen is with PPM AS, Leirfossvegen 27, 7038 Trondheim, Norway `trygve.thomessen@ppm.no`

reality representations can improve the situation awareness of the remote site and facilitate the remote control of industrial manipulators that have to act in such remote environment.

In particular, in [5] the authors have proposed an inspection robot to monitor offshore operations on oil and gas platforms. In their paper, Bjerkeng *et al.* presented a flexible camera view based on the weighted pseudoinverse redundancy resolution method. They could autonomously monitor the operation of a second industrial robot while a user could adjust the zoom according to his/her needs. By appropriately dividing the solution space in task space and null space, the monitoring robot could perform its main objective while handling singularities during its motion.

It is worth noticing that an area of improvement could be in extending the adjustment that the user is allowed to make also to the viewing angle (compared to only relative distance adjustments). However, providing more possibilities to adjust the monitoring viewpoint comes with a cost of increasing the difficulty of the overall monitoring robot control. Therefore, this paper illustrates such an approach that allows for more flexibility in the viewpoint control during operation.

## III. VIRTUAL FIXTURES

Virtual fixtures (also called *active constraints*) are a concept introduced by Rosenberg [2] as a way to anisotropically influence robot movements. Active constraints are a very important concept for many telesurgery applications, and have been thoroughly surveyed in this light by Bowyer *et al.* in [6]. For conciseness, the term virtual fixtures will be used instead of active constraints from now on. This work adopts the geometric approach discussed by Marayong *et al.* in [7]. More specifically, the virtual fixtures are represented by a set of *preferred* and *non-preferred* directions of motion which can be designed to be an abstract surface that the robot's end-effector cannot penetrate. The fact that some directions are identified as non-preferred means that the end-effector motion will be less compliant along such directions, as if the end-effector were experiencing some resistance.

It is important to notice that the use of virtual fixtures is independent of the type of control scheme used on the robot, which could be either admittance or impedance control. With a very brief description, we could say that impedance control imposes a force on the robot (spring-mass-damper behaviour), while admittance control imposes a position. With impedance control, forces can be applied to the robot in response to its interaction with the environment. In admittance control, the robot motion is purely decided by the control software and the robot tends to be stiffer when it gets in contact with external objects. In particular contexts, admittance control can be seen as a "safer" approach compared to impedance control due the absence of motion if the user input drops to zero. However, admittance and impedance control could be interchanged in many applications. This paper focuses solely on admittance control, meaning that the control software will filter the user input and apply the filtered motion to the master reference (which in this case is a 3D model of the monitoring robot).

The way to achieve this behaviour in admittance control, is to filter the user input commands along the directions described by the virtual fixture.

More particularly, let us assume then to have a $6 \times n$ matrix $D = D(t)$, $1 < n < 6$ containing the preferred directions of motion. The dimension of the $D$ matrix is determined by the type of constraint imposed on the end effector. For example, if $n$ is 1, the preferred motion is along a curve in $SE(3)$, or along a surface if $n$ is 2 and so on. As described by Marayong *et al.* in [7] and by Hager in [8] the input vector can then be decomposed along preferred and non-preferred directions with the Kernel and Span operators:

$$v_D \equiv [D]\mathbf{f}_{in} \quad \text{and} \quad \boldsymbol{v}_\tau \equiv \langle D \rangle \mathbf{f}_{in} \tag{1}$$

where $\mathbf{f}_{in}$ is the vector containing the user input motion.

Due to the properties of the Kernel and Span operator (for more details see Hager in [8]), and since it is possible to write $v_D + v_\tau = \boldsymbol{v}$, it is possible to write the following relationship:

$$\boldsymbol{v} = c([D] + c_\tau \langle D \rangle)\mathbf{f}_{in} \tag{2}$$

where $c_\tau \in [0,1]$ is the compliance factor for the non-preferred directions. The smaller the value of $c_\tau$, the smaller the compliance along the non-preferred directions of motion. If $c_\tau$ is chosen to be equal to zero, the virtual constraint is a *hard* virtual fixture, as opposed to any other value which instead would still permit motion along the non preferred directions.

It is worth observing that, with such definition, the monitoring robot's end-effector can move on a path that is *parallel* to the preferred directions of motion specified in $D$ at every given time, and this is typically the case if, at every instant, the robot moves along the tangent plane (or line) of the abstract surface. In fact, the linearisation error accumulates over time and that results in the end-effector moving onto a parallel path. Instead of modifying the $D$ matrix by including an "attraction" term, the linearisation error is automatically corrected after every user input command, in order to eliminate the *drifting* component that tends to move the end-effector away from the virtual fixture. If this correction is performed at every iteration, provided that the user is controlling the robot, the "active" movement that is imposed on the robot at each iteration is small enough not to be a concern in terms of safety.

## IV. LOCAL VIRTUAL FIXTURES AND AUTONOMOUS TRACKING

An additional observation is that such virtual fixtures are not *static* in the 3D environment where we control the monitoring robot. Instead, the virtual fixtures can move, typically together with the end-effector of the task robot. In the literature they are called *dynamic virtual fixtures* or *dynamic active constraints* [6].

However, we believe the use of this term might be inappropriate and misleading here because in this paper we have designed a virtual fixture that in certain situations can be manipulated by the user to influence the navigation capabilities of the monitoring robot.

This section describes the use of a non-compliant virtual fixture that can be manipulated by certain user commands. This virtual fixture can be used together with autonomous tracking motions also when a user performs manual adjustments. Figure 2 shows the conceptual representation of the local virtual fixture (LVF) with the shape of a sphere. The end effector of the monitoring robot is constrained on the sphere's surface and the whole fixture can move in space according to the motion of the task robot's end-effector.

The virtual fixture is initialized with respect to a point or object in the 3D environment, such as the task robot tool tip or for example the workpiece. If the point to which the local virtual fixture is anchored moves inside the workspace, then the whole fixture moves along as well. As a consequence of this motion, the monitoring robot is also moving in order to remain on the virtual sphere's surface. This behaviour is the motion that is generated by the autonomous tracking part. The monitoring robot motion which is instead generated by user input is only affecting the position of the monitoring robot's end-effector relatively to the virtual fixture.

Figure 3 shows schematically how the monitoring robot's reference is influenced by both user motion and autonomous tracking motion. The monitoring robot can be controlled by any combination of these two sources.

Another property of the local virtual fixture is that in certain scenarios the user can manually modify the anchor's position in the workspace. This scenario is contemplated because the user should be able to change the "focus point" during the operation, meaning that the monitoring robot can be set to inspect different areas of the workspace while being constrained to the surface of the local virtual fixture.

$$\boldsymbol{v} = \boldsymbol{v}_u + \boldsymbol{v}_a \quad \text{and} \quad \boldsymbol{v}_u = c([D] + c_\tau \langle D \rangle)\boldsymbol{v}_{in} \tag{3}$$

where $\boldsymbol{v}_a$ is the velocity vector of the autonomous navigation part. In (3) $\boldsymbol{v}_u$ is the user velocity vector that in (2) was simply called $\boldsymbol{v}$.

## V. VARYING COMPLIANCE

Depending on the application scenario, a noncompliant virtual fixture can bring advantages as well as disadvantages. In this particular case, the zero compliance property of the local virtual fixture conveniently allows to fulfil the task implicit objectives. These objectives are the orientation of the monitoring robot end-effector (i.e. the camera view) towards the anchor point (the object that is being inspected) and a minimum distance from the workpiece or task robot for example.

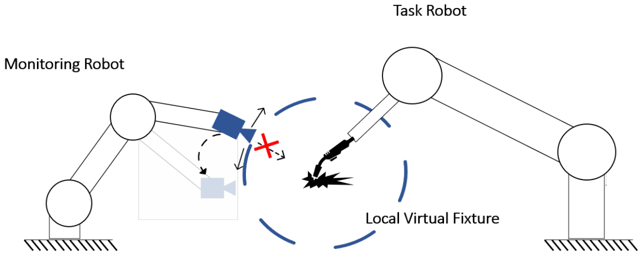


Fig. 2. A local virtual fixture (spherical surface) to monitor an external process.

However, the local virtual fixture does not give any constraint about how the motion should be influenced when approaching other obstacles. Currently obstacles (or forbidden region that can be regarded as obstacles) are expressed in the 3D environment as virtual fixtures so to prevent their contact with the monitoring robot. In the current implementation, however, it possible that the local virtual fixture partially intersects one or more obstacles' forbidden regions if the user decides to change the inspection target by moving the local virtual fixture inside the workspace.

Whenever this happens, it is possible to provide more than just haptic feedback when the monitoring robot comes in contact with the obstacles virtual fixtures (as it moves along the LVF surface). The compliance of the system can be modified to simulate an increase in friction as the robot approaches an obstacle, even though the robot still only moves along the *preferred* directions of motion.

It is important to notice that in the case of a partly compliant VF, a similar result could be achieved by gradually changing $D = D(t)$ to exclude the direction of motion that would move the robot toward the obstacle.

Let us assume to have a spherical LVF, that intersects a certain obstacle represented by its own VF, $S_{obst} \in SE(3)$. Let us continue by assuming that the set of point of the intersection is known, and that for any position of the end effector $\boldsymbol{x}_{tcp}$ it is possible to determine the point $\boldsymbol{P} \in S_{obst}$, and belonging to the intersection, that is closest to the monitoring robot's end effector. Let us introduce the varying compliance $c = c(\boldsymbol{x}_{tcp}(t))$, function of the end-effector position $\boldsymbol{x}(t)$ :

$$c(\boldsymbol{x}(t)) = \begin{cases} 1 & dist(\boldsymbol{x}_{tcp}, \boldsymbol{P}) > h, \quad 0 < h \leq R_{LVF} \\ w_s dist(\boldsymbol{x}_{tcp}, \boldsymbol{P}) & dist(\boldsymbol{x}_{tcp}, \boldsymbol{P}) \leq h, \quad 0 < w_s \leq 1 \\ 0 & dist(\boldsymbol{x}_{tcp}, \boldsymbol{P}) = 0 \end{cases} \tag{4}$$

where $h$ is a threshold that determines at what distance the system starts having less compliance on the preferred directions, while the term $w_s$ is a scaling factor that can be conveniently chosen as $w_s = \frac{1}{h}$ so that $c(\boldsymbol{x}(t)) \in [0, 1]$. If the LVF is spherical, $dist(\boldsymbol{x}_{tcp}, \boldsymbol{P})$ corresponds to the spherical distance. If the LVF is not of canonical shape (e.g. is a sensor generated 3D surface) the distance between two points is computed as the distance between two nodes on a graph, where the graph is obtained by sub-sampling the 3D surface.

## VI. SPHERICAL VIRTUAL FIXTURE IMPLEMENTATION

A local virtual fixture with spherical shape has been implemented (see figure 2), to test the behaviour of the monitoring robot during motion.

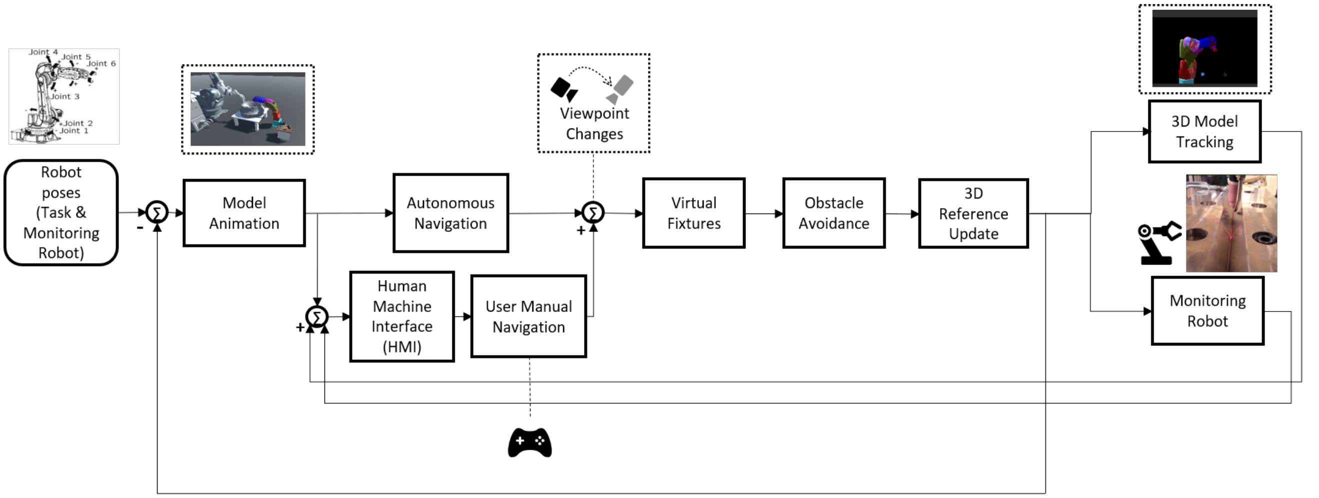


Fig. 3. The simplified control diagram of the monitoring software. Autonomous navigation (or motion) input are in parallel with the user manual navigation. The user can manually adjust the viewpoint of the monitoring robot during the operation, and the changes issued are added to the autonomous tracking motion.

The form of the $D$ matrix then has to describe the preferred motions as the sphere tangent plane that passes through the monitoring robot's end-effector. During the initialization, the control software makes sure that the monitoring robot's end effector is touching the virtual fixture surface, by changing it's radius if necessary. The $D = D(t)$ matrix takes the form:

$$\mathbf{D} = \begin{bmatrix} t_{x_1} & t_{y_1} & t_{z_1} & 1 & 0 & 1 \\ t_{x_2} & t_{y_2} & t_{z_2} & 1 & 0 & 1 \end{bmatrix}^{\mathbf{T}} \qquad (5)$$

where $(t_{x_1}, t_{y_1}, t_{z_1})$ and $(t_{x_2}, t_{y_2}, t_{z_2})$ are the vectors that identify the tangent plane to the sphere's surface at any given time. The tangent vectors will change as the end-effector moves along the sphere's surface, making the $D$ matrix time dependent as expected. The elements of $D$ equal to one indicate the freedom to rotate along the $Z$ and $X$ axis in order to maintain the focus toward the centre of the virtual fixture.

The framework to control the monitoring robot is based on Unity3D [9], where a 3D replica of the monitoring robot serves as the reference for the real hardware (the real robot is shown in figure 4). The pose of both the 3D reference and the real robot were recorded to test the performance of the control software during certain test trajectories. In particular, the monitoring robot motion has been observed when performing *ideal* manual movements, when autonomously tracking a moving point, and in a combination of the two motions.

Manual movements are *ideal* because they are actually simulated via software for reproducibility purposes. The input commands are simulated as coming from the same interface that would be used by the human operator (a Joystick) but without noise in the resultant motion.

### A. Manual Motion

The trajectory used for the manual motion is composed of four simple steps:

1. Move "left" with respect to the monitoring tool frame (or camera view) while zooming *out*
2. Move "right" with respect to the monitoring tool frame (or camera view) while zooming *in*
3. Move "right" with respect to the monitoring tool frame (or camera view) while zooming *out*
4. Move "left" with respect to the monitoring tool frame (or camera view) while zooming *in*

The robot starting position, the zooming speed and movement speed are reported in table I.

The resultant path has been performed ten times, shifting the monitoring robot's end-effector "downward" (w.r.t. the monitoring tool frame) of 0.5 millimetres between each

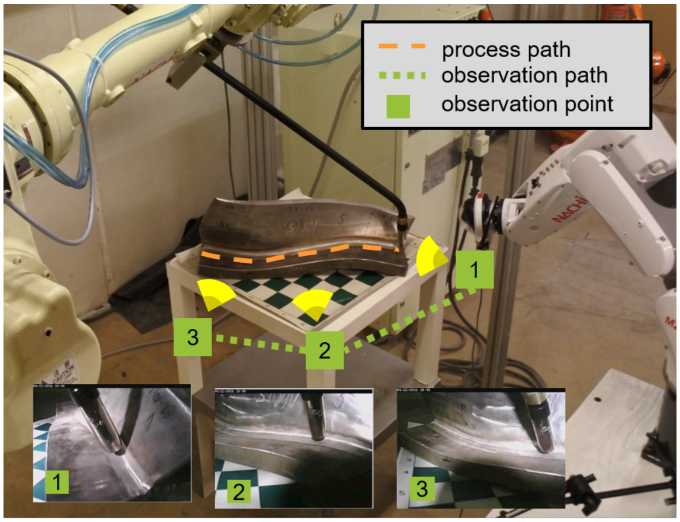


Fig. 4. The industrial setup where a monitoring robot (on the right) can inspect a workpiece. The way-points shown in the picture are selected via user adjustments as the process is being carried out. Note that the relative positioning between the monitoring robot and the task robot changes from one point to the other.

| **Starting Position ($J_1$,$J_2$...$J_6$)** | (0.002, 89.998, -0.004, 0.005, -89.997, -2.324) [deg.] |
|---|---|
| **Starting Position (x,y,z)** | (0.280, 0.553, 0) [m] |
| **Zooming Speed** | 1.0 [cm/s] |
| **Movement Speed** | 1.0 [cm/s] |
| **Initial LVF Radius** | 10 [cm] |
| **Maximum LVF radius** | 15 [cm] |

TABLE I
PARAMETERS FOR THE MANUAL MOTION TEST.

iteration, as can be noticed by the drift along the X-axis in figure 5.

Moreover, in this test the user input not only modifies the position of the monitoring robot on the LVF, but also directly modifies the radius of the virtual sphere (the zooming motion). Although changing the radius of the LVF might seem to contradict the original purpose of a virtual fixture, such degree of freedom is normally not enabled and can only be used by a human operator. Finally, the spherical virtual fixture radius has a minimum value of operation that not even the user in manual motion can override. This radius limit prevents the robot from getting too close to the monitoring target. The tracking error between the 3D model and the real robot while performing the manual motion test is shown in figure 6.

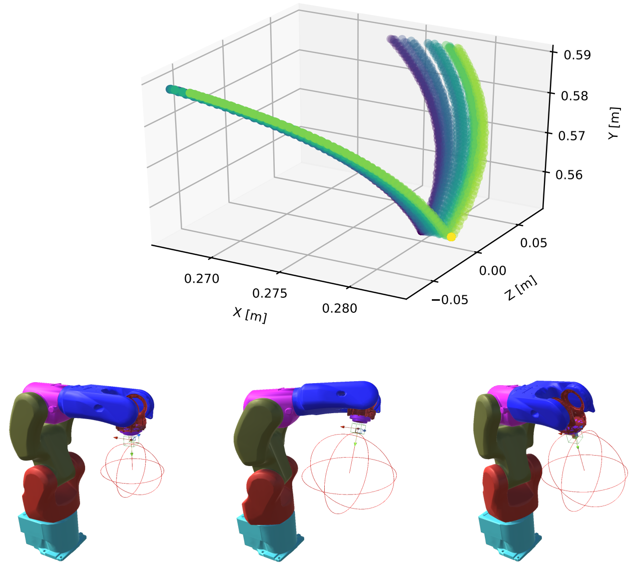


Fig. 5. Trajectory for the test with manual motion. The robot moves along the surface of the local virtual fixture, which has a fixed position.

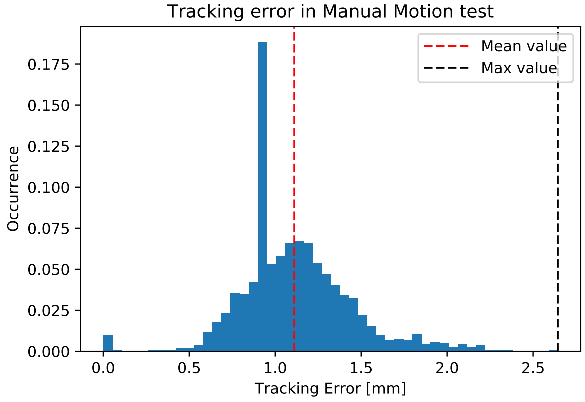


Fig. 6. The histogram of the tracking error during the manual motion test.

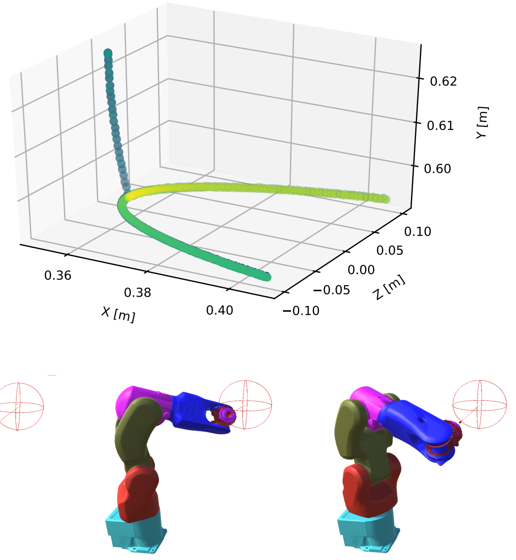


Fig. 7. Trajectory used for the manual & autonomous test. In this picture only the manual motion is displayed.

### B. *Manual & Autonomous Motion*

The other round of tests consists into combining motion commands from the joystick interface with the autonomous tracking motion. More specifically, in the Unity environment the local virtual fixture is initialized anchored to a point in space (referred to as ”tracked point” in table II) that will start moving back and forth along a line parallel to the Z-axis, defined by two predefined positions (see table II).

The trajectory generated by manual input instead consists into a downward movement (vertical axis) followed by a repetitive left/right movement along the X axis of the end-effector frame, always constrained to be on the virtual sphere's surface. The trajectory followed by the robot if there was no input coming from the autonomous tracking software ($\boldsymbol{v}_a = 0$) is shown in figure 7.

The path that results from the combination of the two motion commands is instead shown in figure 8, and the parameters for the manual and autonomous motion test are reported in table II.

| | |
|---|---|
| **Robot Starting Position ($J_1$,$J_2$...$J_6$)** | (0.002, 89.998, -0.004, 0.005, 0.002, -0.004) [deg.] |
| **Robot Starting Position (x,y,z)** | (0.352, 0.625, 0) [m] |
| **Tracked Point: Initial Position** | (3.023, -0.375, 0.15) [m] |
| **Tracked Point: Final Position** | (3.023, -0.375, -0.15) [m] |
| **Robot Zooming Speed** | 1.0 [cm/s] |
| **Robot Movement Speed** | 1.0 [cm/s] |
| **LVF Radius (constant)** | 10 [cm] |
| **Tracked Point Speed** | 2 [cm/s] |

TABLE II

PARAMETERS FOR THE MANUAL AND AUTONOMOUS MOTION TEST. THE LVF RADIUS REMAINS CONSTANT DURING THE TEST. THE TRACKED POINT MOVES LINEARLY BETWEEN THE INITIAL AND THE FINAL POSITION.

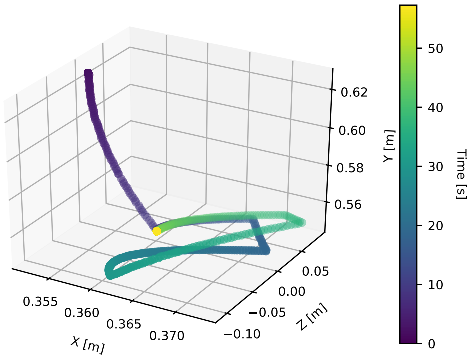


Fig. 8. Path generated by the combination of autonomous tracking movement and manual input commands.

However, it is important to observe that the autonomous tracking motion can generate complications during operation if the monitoring task is not designed appropriately. The monitoring robot can track an abstract point that is only moving in the 3D representation of the scene (as in the current experiment) or, alternatively, it can track the task robot's end-effector. In any case, the autonomous component of the motion can set the monitoring robot after unreachable poses (typically because task robot and monitoring robot have different sizes). Currently, the autonomous tracking is disabled whenever this problem arises, and the LVF anchored to the last reachable position, in order to give priority to the local virtual fixture constraints that still have to be fulfilled by the robot. This possibility is a known problem that will be kept under observation also in future experiments.

## VII. CONCLUSIONS & FUTURE WORK

This paper discussed how our monitoring robot is capable of moving inside the workspace respecting the constraints imposed by the local virtual fixture. It is then possible to inspect a certain workpiece from different angles while respecting constraints like ”look at” orientation and minimum distance from the objective. With this abstraction, the monitoring robot can still perform the inspection on a moving target as shown in the experiments, and manual user adjustments are still permitted during the operation.

Moreover, the concept of varying compliance has been introduced as an approach to regulate the monitoring robot motion on the virtual fixture in the presence of obstacles or other critical forbidden regions.

However additional complications can occur as the monitoring robot moves in the workspace. It might happen that the monitoring robot cannot reach a certain viewpoint and remains stuck due to reachability limitations. Moreover, it is important to evaluate how the use of local virtual fixtures and varying compliance are perceived by the user.

Usability is a very important factor, since the user should be able to operate the monitoring robot for workspace inspection without a sensible increase in workload.

## ACKNOWLEDGMENT

This research is supported by The Research Council of Norway through the Industrial PhD project 264145/O30 Interactive robot operation using multi-modal man-machine communication”

## REFERENCES

[1] E. Sita, C. M. Horváth, T. Thomessen, P. Korondi, and A. G. Pipe, “Rosunity3d based system for monitoring of an industrial robotic process,” in *System Integration (SII), 2017 IEEE/SICE International Symposium on*. IEEE, 2017, pp. 1047–1052.

[2] L. B. Rosenberg, “The use of virtual fixtures as perceptual overlays to enhance operator performance in remote environments.” Stanford Univ Ca Center for Design Research, Tech. Rep., 1992.

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

[4] F. Carvalho, A. Raposo, I. Santos, and M. Galassi, “Virtual reality techniques for planning the offshore robotizing,” in *Industrial Informatics (INDIN), 2014 12th IEEE International Conference on*. IEEE, 2014, pp. 353–358.

[5] M. Bjerkeng, A. A. Transeth, K. Y. Pettersen, E. Kyrkjeb, and S. A. Fjerdingen, “Active camera control with obstacle avoidance for remote operations with industrial manipulators: Implementation and experimental results,” Sept 2011, pp. 247–254.

[6] S. A. Bowyer, B. L. Davies, and F. R. y Baena, “Active constraints/virtual fixtures: A survey,” *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 138–157, 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6634270

[7] P. Marayong, M. Li, A. M. Okamura, and G. D. Hager, “Spatial motion constraints: theory and demonstrations for robot guidance using virtual fixtures,” vol. 2. IEEE, 2003, p. 1959 vol.2. [Online]. Available: http://ieeexplore.ieee.org/document/1241880

[8] G. D. Hager, “Human-machine cooperative manipulation with vision-based motion constraints,” in *WS2: Workshop on Visual Servoing*, 2002, p. 1.

[9] Unity3d.com. (2018) Unity 3d: Game engine. https://unity3d.com/ and https://unity3d.com/public-relations. [Online]. Available: https://unity3d.com/

# Usability Study of a Robot Companion for Monitoring Industrial Processes

1st Enrico Sita
*Engineering Design and Mathematics*
*University of the West of England*
Bristol, UK
enrico2.sita@live.uwe.ac.uk

2nd Trygve Thomessen
*PPM Robotics AS*
Trondheim, Norway
trygve.thomessen@ppm.no

3rd Anthony G. Pipe
*Engineering Design and Mathematics*
*University of the West of England*
Bristol, UK
tony.pipe@brl.ac.uk

4th Matthew Studley
*Engineering Design and Mathematics*
*University of the West of England*
Bristol, UK
Matthew2.Studley@uwe.ac.uk

5th Farid Dailami
*Engineering Design and Mathematics*
*University of the West of England*
Bristol, UK
Farid.Dailami@uwe.ac.uk

***Abstract*—In this paper we present the findings of a usability study for a monitoring robotic unit tele-operated via a virtual fixtures (VF) based control framework. The study aims at investigating the impact of VF on the robot navigation as well as the impact of multimodal feedback on the user performance in a static inspection task. The findings will help in the design of the monitoring control framework to inspect a robotised welding process, as it has been researched in previous work. The study has been conducted with untrained participants, involved in four (4) different test scenarios. The experiments treated a static case in which users were asked to navigate the monitoring robot in the workspace to find a lit LED of a test-piece. The statistical analysis of the experiment metrics showed a positive impact of the VF control on the navigation of the monitoring robot even for users with no previous experience. Moreover, from the analysis of the task load index forms (TLX) it emerged that the combination of VF control and additional multimodal feedback improved the user performance without negatively impacting the effort required to accomplish the task.**

***Index Terms*—virtual fixtures, monitoring robot, user study, usability evaluation, multimodal feedback**

## I. INTRODUCTION

For system integrators, optimizing complex industrial robotic applications (e.g. robotised welding) is a difficult and time-consuming task. This is usually due to discrepancies between the models and the actual behaviour of complex systems, and the system integrator needs to fine tune the final installation by trial and error to obtain the desired quality. This procedure is even more tedious when the operator cannot access the robotic system once in operation and must rely on additional sensors to acquire the necessary process information. However, it is often difficult to find a permanent placement for the sensors to be able to fully monitor the process at any given time during the trials, and this would also be a very expensive and potentially unreliable approach, if applied to all of the robot installations. While it is hard to completely remove this trial and error fashion, it is possible to provide a way to gather process information more effectively that can be used in several robotic installations. It is then proposed to provide the system integrator with a monitoring robot in addition to the robot(s) belonging to the industrial process that needs to be optimized (also referred to as *task robot(s)*). The monitoring robot can be equipped with several different sensors and can be moved into close proximity of any installed robot so that it can be used to collect information from that process during and/or after the operation without interfering. The system operator can control the monitoring robot to change its viewpoint and acquire information from various positions (e.g. inspect a workpiece from different angles). With a more effective way of gathering process data, the system integrator can perform his/her primary task (optimizing the industrial process) more efficiently. Since controlling the monitoring robot is not a primary task, the challenge is to make such interaction as flawless as possible not to overload the operator. The operator will control the monitoring robot with a camera view from its endeffector and via a joystick or similar interface.

The concept and the framework to control the monitoring robot and synchronize it with the task robot has been previously discussed in [1]. The control strategy based on virtual

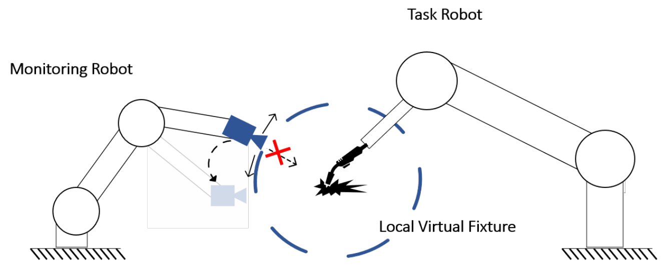


Fig. 1. Conceptual representation of a virtual fixture. The virtual fixture prevents the robot's end-effector from entering a certain forbidden region.

fixtures, its implementation and advantages in a user-centered design have been presented in more detail in the work of [2]. To the best of the authors' knowledge, there hasn't been yet in the literature a user study on the usability of such control and its effect on untrained participants when applied to a monitoring task and it will be the main contribution of this paper. Moreover, the usability study investigates on the effects on performance and cognitive load that multimodal feedback introduce and whether they are suitable for this type of applications. The experiment and its rationale is explained in section V and the data from the experiments are presented in section VI. Conclusions are drawn in section VII with some input for further discussion.

## II. RELATED WORK

The idea of adopting a secondary robot with the function of a monitoring unit has been explored by an increasing number of papers in recent years, as it provides flexibility in the choice of viewpoint angle in the workspace as well as allowing any inspection to be performed remotely (so improving EHS conditions when the industrial process is carried on in a harsh environment).

Carvalho *et al.* in [3] discuss virtual reality approaches to be able to inspect an Oil offshore platform, so as to improve understanding during simulation before moving to the real industrial site.

In particular, the work from [4] discussed a modular robot for autonomous inspection and maintenance of hazardous industrial scenarios, to be deployed in sites such the CERN laboratory where maintenance of the extensive equipment is paramount to the experiments preparation and execution. The focus in their work is mostly on how to make the navigation autonomous and enable manipulation skills in such delicate scenarios and present the information through an adaptive graphical user interface, presented in [5].

Another important work recently published is the paper from [6], where the authors propose a method to assist an operator during a teleoperation task. The approach involves an external monitoring unit that is autonomously following the manipulator robot that is controlled by the user. In order to provide the appropriate viewpoint, the monitoring system has to use motion prediction and concepts from animation and graphics in order to evaluate which pose is the best as the user control the manipulator.

It is also worth mentioning a slightly more dated, but nonetheless pertinent contribution from [7] where they also proposed to used a monitoring robot for an industrial task to be carried offshore.

Although work could be done to integrate subparts of the autonomous behaviours that have been presented in these works, this paper focuses on the challenge of having a user in charge of the *monitoring* task, and not for example of the manipulation part as in the work of [6]. The challenge introduced by a user in charge of the monitoring is in the way the system handles the navigation of the monitoring robot with respect to the task robot and the surroundings. Whenever the user stops the manual operation, the system could resume its motion according to the behaviour specified by [6].

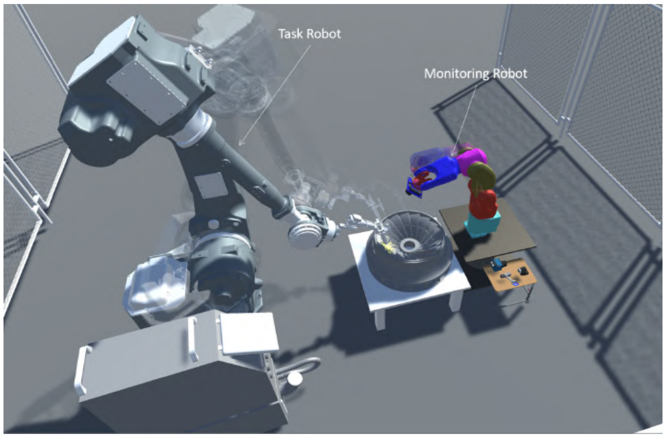


Fig. 2. Example scene in the Unity environment with a task robot (on the left) and the monitoring robot used for the experiments (on the right).

## III. VIRTUAL FIXTURES

Virtual fixtures (VF) (also called active constraints) are a concept introduced by [8] as a way to anisotropically influence robot movements. Active constraints are a very important concept for many telesurgery applications, and have been thoroughly surveyed in this light by [9]. The virtual fixtures are represented by a set of preferred and non-preferred directions of motion which can be designed to be an abstract surface that the robots end-effector cannot penetrate. The fact that some directions are identified as non-preferred means that the end-effector motion will be less compliant along such directions, as if the end-effector were experiencing some resistance. We illustrate in 1 the main equation that dictates how the monitoring robot's end-effector velocity is influenced when in contact with a virtual fixture (see Figure 3 for a schematic representation):

$$\boldsymbol{v} = c([D] + c_\tau \langle D \rangle)\boldsymbol{v}_{in} \tag{1}$$

where $c_\tau \in [0,1]$ is the compliance factor for the non-preferred directions. The smaller the value of $c_\tau$, the smaller the compliance along the non-preferred directions of motion. If $c_\tau$ is chosen equal to 0, it provides a *hard* virtual fixture, as opposed to any other value which instead would still permit motion along the non preferred directions.

The details of such control implementation and the design choices made for its adaptation for using it with a monitoring robot in tele-operation have been discussed in more depth in [2].

## IV. TEST CASES AND MOTIVATION

A usability evaluation has been conducted to provide insights on the efficacy and benefits of the VF control approach and of the monitoring framework in general. It is important to evaluate in particular whether the control of the monitoring robot does not increase too much the mental and cognitive load of the operator. In fact, the goal of the operator is to inspect the workspace and the process carried out in it and not to control the monitoring robot per se.

The main questions that are being investigated in this section are the following:

- *Does the VF control framework allow the operator to control the monitoring robot without overloading him/her?* This question will be answered by comparing TLX indexes among the scenarios where VF redirection was not enabled to the ones where it was active.
- *What are the effects on performance introduced by the VF control framework?* This question will be answered by comparing the average completion time among the different experiment scenarios.
- *What are the effects of an extra feedback modality on the user TLX score and performance?* The final question will investigate if an extra feedback modality can improve performance in this particular task, and what are the effects on the users' TLX scores when this feedback is introduced, with particular focus on the users' cognitive load.

## V. EXPERIMENTS

The purpose of the monitoring robot is to assist the system integrator in the tuning phase of an industrial process, which is the main task for the system integrator. It is therefore important that the monitoring robot can be controlled with as little effort as possible not to increase the workload and the time needed during the optimization phase. It is known that when manual input is allowed for robotic systems, both the control scheme and the user interface greatly contribute to decrease the overall complexity for the human operator.

The modes of controlling the monitoring robot are two: *unconstrained* and *constrained*. In the first mode, the camera view (the robots end effector essentially) can be freely moved inside the workspace with a simple mapping of rotation and translation to the joysticks buttons. In the second mode, the camera view (position and orientation) is influenced by virtual surfaces defined via software.

The experiments aim at investigating whether untrained users can more efficiently accomplish the task of acquiring information from a process with the help of a virtual fixture based control. The VF based control should assist the user in respecting secondary objectives such as minimum distance from specific objects or from the workpiece without requiring additional effort from the the operator. Choosing the most appropriate metrics

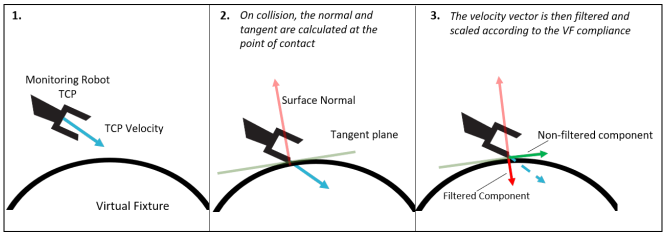


Fig. 3. Representation of the steps of the velocity filtering when colliding with a VF. The computation and control algorithm are performed in Unity.

The metrics that are collected for both the static and dynamic scenarios are the following:

- Completion time (CT): this is the time to complete a single task, from the start event to the end event. The start of a task is identified by a LED lit and followed by a notification to the user. The task ends when the user finds and correctly classifies the lit LED's colour.
- Number of commands (NC): the number of control commands that a user inputted during a single task.
- Number of corrective actions (NCA): the number of times the system with VF actively filters a user movement that would otherwise violate a forbidden region.
- Number of pseudo-violations (NPV): the number of times that the robot collides with a VF and has to be either stopped or redirected.

The static experiment setup consists in a 6-dof manipulator equipped with a RGB camera. The experiment workpiece (see Figure 5, placed in front of the robot, is a Raspberry Pi with eight multicolour LEDs that can be lit remotely from the Unity framework.

The user's task is to identify which LED is lit at a certain time and what is its colour. To accomplish the task, the users must control the camera view via a joystick to be able to look at the correct LED and identify the LEDs colour (input mapping shwown in Figure 6).

Each user is asked to find the coloured LED 4 times in a single session. Once these four (4) attempts have been accomplished the user is asked to fill a NASA-TLX form and the experiment is then over.

Four (4) experiments have been conducted, each with fifteen (15) participants. As it is shown in table I, the number of elements that could influence the users' performance was increased from one experiment to the next. The added obstacles are invisible to the users, and have one additional property in addition to their position and shape: the compliance. As described in more detail in the work of [2], the compliance changes how "hard" an obstacle is in response to the user motion that would penetrate it. For all the experiments, spherical virtual fixtures were used. However, in some cases, overlapping so that to the user the obstacles didn't always have a spherical shape. The Unity environment is shown in Figure 7 as well as the visual feedback element used in one of the experiment scenarios.

- Scenario A - simple free motion without VF redirection, with no obstacles in the workspace. In the first experiment, the only virtual fixtures that have been used were to ensure the safety of the workspace and the robot.
- Scenario B - introduces obstacles in the workspace. The users' goal is unchanged but the robot will have constrained motion every time it will come in contact with a virtual fixture. In the second experiment, whenever the

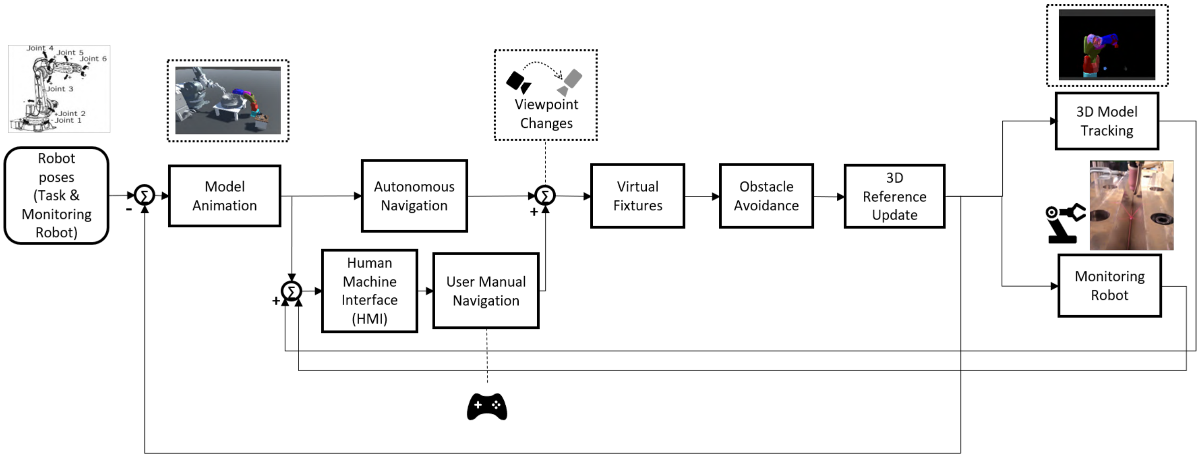


Fig. 4. The simplified control diagram of the monitoring software. Autonomous navigation (or motion) input are in parallel with the user manual navigation. The user can manually adjust the viewpoint of the monitoring robot during the operation, and the changes issued are added to the autonomous tracking motion.

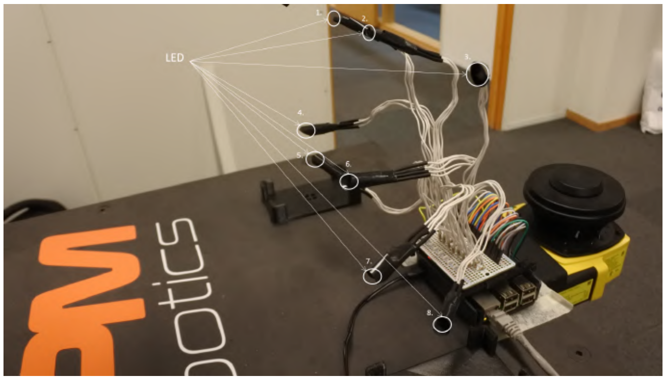


Fig. 5. Experiment piece used for the static experiment. The LEDs are triggered via a Python script on the Raspberry Pi.

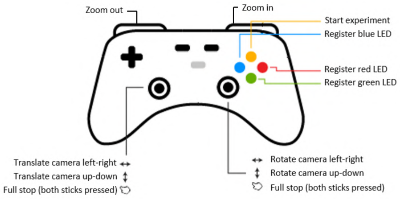


Fig. 6. Input mapping for the joystick used to control the monitoring robot. The commands are from the camera perspective, which corresponds to the robot's TCP.

robot is colliding with a virtual fixture the only motion allowed is in a direction that resolves completely the collision state, generally along the normal of the VF on the point of contact.

- Scenario C - introduces virtual fixture redirection, which facilitate the robot motion when it comes in contact with an obstacle.
- Scenario D - visual and haptic feedback component added to the experiment scenario. The feedback is provided when the user is ”about to” collide with an obstacle, and also during the collision state.

TABLE I
Overview of the difficulty elements in the experiment scenarios

| | Obstacles | VF Redirection | Add. Feedback |
|---|---|---|---|
| **Free Workspace** | - | - | - |
| **Obstacle Baseline** | X | - | - |
| **Obstacle Redirection** | X | X | - |
| **Obstacle Feedback** | X | X | X |

## VI. USER TRIALS RESULTS

We describe the experiment setup, with the rationale behind (we want the monitoring robot to navigate in the work-space, and we want the user to accomplish a task that requires specific positions and orientations). Describe which different experiment have been performed (static and with obstacles) with the hypothesis behind, or at least the expected observations. Then we explain the first static experiment, with how many users and how many trials per users with what have been recorded

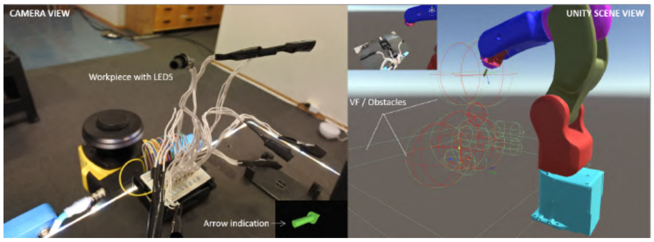


Fig. 7. The 3D arrow used for for the visual feedback in the fourth experiment scenario and the Unity scene view of the obstacles.

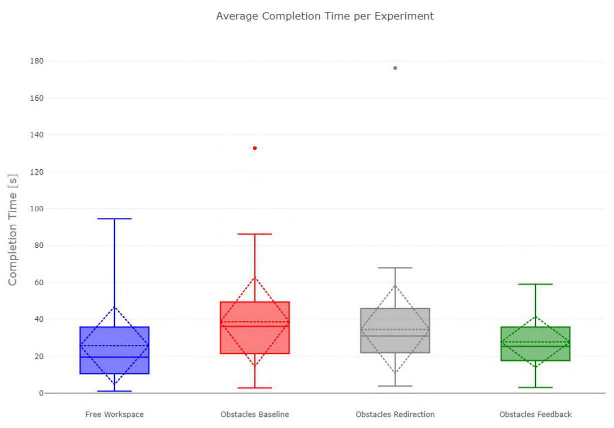


Fig. 8. Average completion times for each experiment scenario

during their trials. We discuss also the TLX questionnaire that each user had to fill after their experiment session. We show the graph of the average completion times and average actions per target.

Each participant is asked to complete a task four (4) times, consisting in locating and classifying one lit LED with the monitoring robot. For each time the metrics are collected, leading to four measurements per metric for a single user trial. Each user performs exactly one trial. We aim at avoiding a ”learning trend” as much as possible since we are not interested in the decrease of completion time and the improvement of the metrics over successive trials but we are rather interested in the differences between the two group of users and their performance with the different control modes.

The participants are mostly students from the last year of BSc and first year of MSc studies (average age is $24.6 \pm 1.87$), with no restrictions on the type of background.

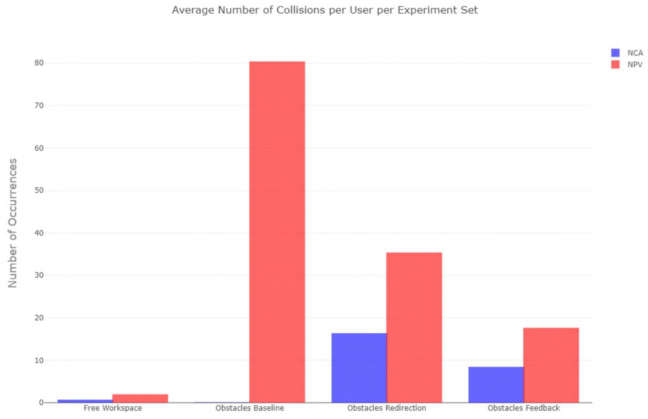


Fig. 9. Average number of NCA and NPV for each experiment set.

TABLE II
COMMON PARAMETERS USED FOR THE EXPERIMENT SCENARIOS. EACH SCENARIO HAS THE SAME MONITORING ROBOT STARTING POSITION.

| | |
|---|---|
| **Robot Starting Position (J$_1$,J$_2$...J$_6$)** | (0.0, 90.0, 0.0, 0.0, 0.0, 0.0) [deg.] |
| **Robot Starting Position (x,y,z)** | (0.0, 0.30, 0.31) [m] |
| **Number of LED Targets** | 8 |
| **Number of Trials per User** | 4 |
| **Robot Zooming Speed** | 1.0 [cm/s] |
| **Robot Movement Speed** | 1.0 [cm/s] |
| **Number of Obstacles in the Space** | 10 |

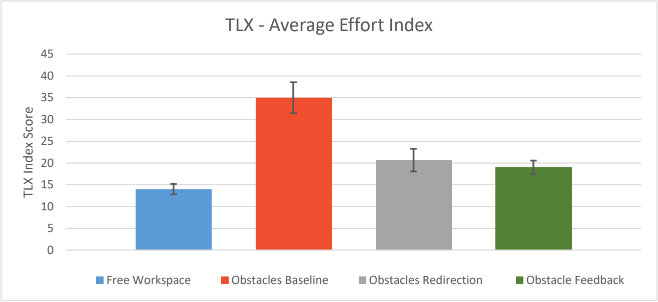


Fig. 10. Average effort index from the TLX forms.

From Figure 10 it can be seen that the average effort decreases from an only obstacles scenario when redirection and additional feedback are introduced, remaining however still greater than the first scenario of the free workspace. This pattern has also characterized the CT analysis, and still positively hinting that although obstacles are increasing the difficulty of the task, redirection and additional feedback contribute in making the task easier for the user.

Performing the one-way analysis of variance (ANOVA) on the users effort scores, it can be stated that there is statistically significant difference between the groups ($F(3,56) = 13.89$, $p = 6.98 \cdot 10^{-6}$). Furthermore, the post hoc tests showed that there is a statistically significant difference between the *Obstalce Baseline* scenario and *Obstacle Redirection* ($P = 0.003$), but not significant enough between *Obstacle Redirection* and *Obstacle Feedback* ($P = 0.59$). This latest result is likely to indicate that the VF redirection plays the bigger role in decreasing the perceived effort to accomplish the task. It is a positive finding in that the most difficult part of the task consisted in avoiding the invisible obstacles, rather than just controlling the monitoring robot to reach a different point of view. The additional feedback did not seem to increase the overall mental load according to the results of the TLX: the mental, physical and temporal scores remained fairly consistent across the different experiment, suggesting that the task was not ”rushed”, was not demanding in terms of physical abilities, nor was requiring high problem solving capabilities which is still consistent with the intention of the experiment scenarios. However, the additional visual and haptic feedback did have an effect on the performance score. In

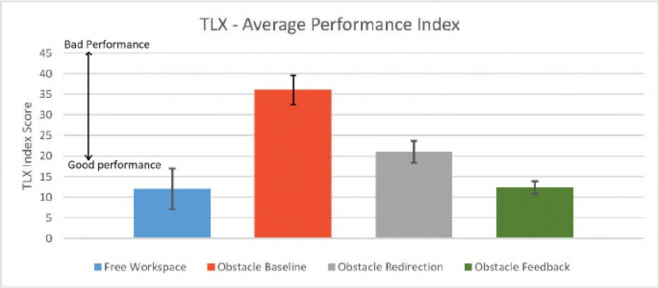


Fig. 11. Average performance index from the TLX forms.

TABLE III

STATISTICALLY MEANINGFUL DIFFERENCE FROM THE ANALYSIS OF VARIANCE (ANOVA)

| | **Statistical Difference** | | |
|---|---|---|---|
| **Scenarios** | CT | Effort | Performance |
| *Obstacle Baseline vs. Obstacle Redirection* | X | X | - |
| *Obstacle Redirection vs. Obstacle Feedback* | - | - | X |

particular, there is statistically significant difference between the different experiment performance scores ($F(3,56) = 10.89$, $p = 9.75 \cdot 10^{-6}$), with the graph comparison shown in Figure 11.

The most interesting detail is that in this case, scenario C (*Obstacle Redirection*) and scenario D (*Obstacle Feedback*) present a statistically significant difference ($P = 0.035$), and at the same time the performance reported in the scenario D, where the additional feedback was provided, is very close to the average performance reported in the free workspace of scenario A. Combined with the previous finding, this analysis suggests that the monitoring robot navigation is actually improved by the VF redirection, and the additional feedback has the effect of making the users feel more efficient at accomplishing the task: if this effect cannot be concluded by looking at the completion times alone, it resonates in the decrease of NCA and NPV thanks to the presence of the additional visual clue and haptic feedback of the duration of the collision with an obstacle.

## VII. CONCLUSIONS & FUTURE WORK

This paper examined the effects of different elements in the monitoring framework on users' performance and the relationships with their task evaluation. Four different experiments were carried out, each experiment with 15 users that were instructed to navigate the monitoring robot to find an LED target multiple times. Each scenario introduced an additional element in the system that affected the navigation, either negatively (like the invisible obstacles) or positively (like VF redirection and additional visual and haptic feedback). Following the experiments with a statistical analysis of the user responses and metrics, it was observed that the VF redirection affects positively the navigation of the monitoring robot. The average CT showed a meaningful decrease from the scenario with obstacles only to the scenarios where obstacles were present but redirection was enabled (see also Table III for the ANOVA summary).

Moreover, the extra feedback modality affected the performance score from the TLX form, with the most statistically significant difference. The results indicate that the additional visual feedback, together with haptic information about the duration of the collision with an obstacle, positively affects the user's performance, and in the presented form are suitable candidates for the type of navigation task that the monitoring robot is expected to carry out in an actual industrial setup.

It is important to mention that the development of the monitoring robot and the VF redirection control finally aims at addressing challenges that are faced in the welding industry during robotized welding. The usability evaluation has a key role in understanding what were the effects of the proposed control framework and monitoring solution in a laboratory setup that replicates part of the difficulties encountered during and actual industrial case.

## ACKNOWLEDGMENT

This research is supported by The Research Council of Norway through the Industrial PhD project 264145/O30 Interactive robot operation using multi-modal man-machine communication"

## REFERENCES

[1] E. Sita, C. M. Horváth, T. Thomessen, P. Korondi, and A. G. Pipe, "Rosunity3d based system for monitoring of an industrial robotic process," in *System Integration (SII), 2017 IEEE/SICE International Symposium on*. IEEE, 2017, pp. 1047–1052.

[2] E. Sita, T. Thomessen, A. G. Pipe, F. Dailami, and M. Studley, "Robot companion for industrial process monitoring based on virtual fixtures," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, Oct 2018, pp. 6051–6056.

[3] F. Carvalho, A. Raposo, I. Santos, and M. Galassi, "Virtual reality techniques for planning the offshore robotizing," in *Industrial Informatics (INDIN), 2014 12th IEEE International Conference on*. IEEE, 2014, pp. 353–358.

[4] M. Di Castro, M. Ferre, and A. Masi, "Cerntauro: A modular architecture for robotic inspection and telemanipulation in harsh and semi-structured environments," *IEEE Access*, vol. 6, pp. 37 506–37 522, 2018.

[5] G. Lunghi., R. M. Prades., and M. D. Castro., "An advanced, adaptive and multimodal graphical user interface for human-robot teleoperation in radioactive scenarios," in *Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics - Volume 2: ICINCO,*, INSTICC. SciTePress, 2016, pp. 224–231.

[6] D. Rakita, B. Mutlu, and M. Gleicher, "An autonomous dynamic camera method for effective remote teleoperation," in *HRI '18*, 2018.

[7] M. Bjerkeng, A. A. Transeth, K. Y. Pettersen, E. Kyrkjeb, and S. A. Fjerdingen, "Active camera control with obstacle avoidance for remote operations with industrial manipulators: Implementation and experimental results," Sept 2011, pp. 247–254.

[8] L. B. Rosenberg, "The use of virtual fixtures as perceptual overlays to enhance operator performance in remote environments." Stanford Univ Ca Center for Design Research, Tech. Rep., 1992.

[9] S. A. Bowyer, B. L. Davies, and F. R. y Baena, "Active constraints/virtual fixtures: A survey," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 138–157, 2014, iD: 1.

**Figure 8.6**

## *NASA Task Load Index*

*Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.*

| | | |
|---|---|---|
| Name | Task | Date |

Mental Demand — How mentally demanding was the task?

Very Low — Very High

Physical Demand — How physically demanding was the task?

Very Low — Very High

Temporal Demand — How hurried or rushed was the pace of the task?

Very Low — Very High

Performance — How successful were you in accomplishing what you were asked to do?

Perfect — Failure

Effort — How hard did you have to work to accomplish your level of performance?

Very Low — Very High

Frustration — How insecure, discouraged, irritated, stressed, and annoyed wereyou?

Very Low — Very High

# References

Abbott, J. J. and Okamura, A. M. (2006) Pseudo-admittance Bilateral Telemanipulation with Guidance Virtual Fixtures. *14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems* [online]. Available from: `http://dx.doi.org/10.1109/HAPTIC.2006.1627057` [Accessed August 8, 2018].

Angelini, L. and Papaspyropoulos, V. (2000) Telesurgery. *Ultrasound in medicine & biology* [online]. 26 (1), pp. 45–47. Available from: `http://dx.doi.org/10.1016/S0301-5629(00)00162-9` [Accessed July 12, 2018].

Arcara, P. and Melchiorri, C. (2002) Control schemes for teleoperation with time delay: A comparative study. *Robotics and Autonomous Systems* [online]. 38 (1), pp. 49–64. Available from: `http://dx.doi.org/10.1016/S0921-8890(01)00164-6` [Accessed October 28, 2018].

Bate, L., Cook, C. D., and Li, Z. (2011) Reducing Wave-Based Teleoperator Reflections for Unknown Environments. *IEEE Transactions on Industrial Electronics* [online]. 58 (2), pp. 392–397. Available from: `http://dx.doi.org/10.1109/TIE.2009.2035994` [Accessed September 23, 2017].

Benz, T. M. and Nitsch, V. (2017) Using multisensory cues for direction information in teleoperation: More is not always better. *IEEE International Conference on Robotics and Automation* [online]. Available from: `http://dx.doi.org/10.1109/ICRA.2017.7989773` [Accessed June 3, 2018].

Bettini, A., Marayong, P., Lang, S., Okamura, A. M., and Hager, G. D. (2004) Vision-assisted control for manipulation using virtual fixtures. *IEEE Transactions on Robotics* [online]. 20 (6), pp. 953–966. Available from: `http://dx.doi.org/10.1109/TRO.2004.829483` [Accessed September 15, 2017].

Birk, A., Schwertfeger, S., and Pathak, K. (2009) A networking framework for teleoperation in safety, security, and rescue robotics. *IEEE Wireless Communications* [online]. 16 (1), pp. 6–13. Available from: `http://dx.doi.org/10.1109/MWC.2009.4804363` [Accessed June 18, 2017].

Bjerkeng, M., Transeth, A. A., Pettersen, K. Y., Kyrkjebø, E., and Fjerdingen, S. A. (2011) Active camera control with obstacle avoidance for remote operations with industrial manipulators: Implementation and experimental results. *IEEE/RSJ International Conference on Intelligent Robots and Systems* [online]. Available from: `http://dx.doi.org/10.1109/IROS.2011.6094823` [Accessed September 8, 2018].

Bowyer, S. A., Davies, B. L., and Rodriguez y Baena, F. (2014) Active Constraints/Virtual Fixtures: A Survey. *IEEE Transactions on Robotics* [online]. Available from: `http://dx.doi.org/10.1109/TRO.2013.2283410` [Accessed December 20, 2018].

Bowyer, S. A. and Rodriguez y Baena, F. (2013) Dynamic frictional constraints for robot assisted surgery. *World Haptics Conference (WHC)* [online]. Available from: `http://dx.doi.org/10.1109/WHC.2013.6548428` [Accessed December 17, 2018].

Bowyer, S. and Baena, F. (2016) A hybrid constraint-penalty proxy method for six degree-of-freedom haptic display of deforming objects. *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology* [online]. Available from: `http://dx.doi.org/10.1145/2993369.2993392` [Accessed October 27, 2018].

Cavenago, F., Voli, L., and Massari, M. (2018) Adaptive Hybrid System Framework for Unified Impedance and Admittance Control. *Journal of Intelligent & Robotic Systems* [online]. 91 (3), pp. 569–581. Available from: `http://dx.doi.org/10.1007/s10846-017-0732-1` [Accessed April 17, 2017].

Chopra, N., Spong, M. W., Hirche, S., and Buss, M. (2003) Bilateral teleoperation over the internet: the time varying delay problem. *Proceedings of the 2003 American Control Conference* [online]. Available from: `http://dx.doi.org/10.1109/ACC.2003.1238930` [Accessed May 27, 2017].

Chung, W. K., Lam, T. L., and Xu, Y. (2014) A natural assisted navigation motion for telepresence robots. *IEEE International Conference on Robotics and Biomimetics (ROBIO)* [online]. Available from: `http://dx.doi.org/10.1109/ROBIO.2014.7090675` [Accessed September 8, 2018].

Di Castro, M., Ferre, M., and Masi, A. (2018) CERNTAURO: A Modular Architecture for Robotic Inspection and Telemanipulation in Harsh and Semi-Structured Environments. *IEEE Access* [online]. 6, pp. 37506–37522. Available from: `http://dx.doi.org/10.1109/ACCESS.2018.2849572` [Accessed September 5, 2018].

Fong, T., Provencher, C., Micire, M., Diftler, M., Berka, R., Bluethmann, B., and Mittman, D. (2012) The Human Exploration Telerobotics project: Objectives, approach, and testing. *IEEE Aerospace Conference* [online]. Available from: `http://dx.doi.org/10.1109/AERO.2012.6187043` [Accessed May 4, 2017].

Gasparetto, A. and Scalera, L. (2019) A Brief History of Industrial Robotics in the 20th Century. *Advances in Historical Studies* [online]. 8 (1), pp. 24–35. Available from: `http://dx.doi.org/10.4236/ahs.2019.81002` [Accessed August 29, 2020].

Hager, G. D. (2002) Human-machine cooperative manipulation with vision-based motion constraints. *IEEE/RSJ International Conference on Intelligent Robots and Systems - Workshop on Visual Servoing*,

Hagn, U., Konietschke, R., Tobergte, A., Nickl, M., Jörg, S., Kübler, B., Passig, G., Gröger, M., Fröhlich, F., Seibold, U., Le-Tien, L., Albu-Schäffer, A., Nothhelfer, A., Hacker, F., Grebenstein, M., and Hirzinger, G. (2010) DLR MiroSurge: a versatile system for research in endoscopic telesurgery. *International Journal of Computer Assisted Radiology and Surgery* [online]. 5 (2), pp. 183–193. Available from: `http://dx.doi.org/10.1007/s11548-009-0372-4` [Accessed November 3, 2017].

Hashtrudi-Zaad, K. and Salcudean, S. E. (2002) Transparency in time-delayed systems and the effect of local force feedback for transparent teleoperation. *IEEE Transactions on Robotics and Automation* [online]. 18 (1), pp. 108–114. Available from: `http://dx.doi.org/10.1109/70.988981` [Accessed November 6, 2018].

Hennekens, D., Constantinescu, D., and Steinbuch, M. (2008) Continuous impulsive force controller for forbidden-region virtual fixtures. *IEEE International Conference on Robotics and Automation* [online]. Available from: `http://dx.doi.org/10.1109/ROBOT.2008.4543648` [Accessed October 26, 2017].

Hokayem, P. F. and Spong, M. W. (2006) Bilateral teleoperation: An historical survey. *Automatica* [online]. 42 (12), pp. 2035–2057. Available from: `http://dx.doi.org/10.1016/j.automatica.2006.06.027` [Accessed August 19, 2017].

Hua, C.-C. and Liu, P. X. (2009) Convergence Analysis of Teleoperation Systems With Unsymmetric Time-Varying Delays. *IEEE Transactions on Circuits and Systems II: Express Briefs* [online]. 56 (3), pp. 240–244. Available from: `http://dx.doi.org/10.1109/TCSII.2009.2015364` [Accessed August 18, 2018].

Hua, C. and Liu, X. P. (2011) Teleoperation Over the Internet With/Without Velocity Signal. *IEEE Transactions on Instrumentation and Measurement* [online]. 60 (1), pp. 4–13. Available from: `http://dx.doi.org/10.1109/TIM.2010.2065410` [Accessed December 9, 2017].

Huang, J., Shi, Y., and Wu, J. (2012) Transparent Virtual Coupler Design for Networked Haptic Systems With a Mixed Virtual Wall. *IEEE/ASME Transactions on Mechatronics* [online]. 17 (3), pp. 480–487. Available from: `http://dx.doi.org/10.1109/TMECH.2012.2188640` [Accessed October 22, 2018].

IFR (2018) *International Federation of Robotics*. Available from: `https://ifr.org/industrial-robots/statistics/` [Accessed August 8, 2018].

Joly, L. D. and Andriot, C. (1995) Imposing motion constraints to a force reflecting telerobot through real-time simulation of a virtual mechanism. *IEEE International Conference on Robotics and Automation* [online]. Available from: `http://dx.doi.org/10.1109/ROBOT.1995.525310` [Accessed May 10, 2019].

Kamezaki, M., Yang, J., Iwata, H., and Sugano, S. (2014) An autonomous multi-camera control system using situation-based role assignment for tele-operated work machines. *IEEE International Conference on Robotics and Automation* [online]. Available from: `http://dx.doi.org/10.1109/ICRA.2014.6907739` [Accessed December 12, 2017].

Kapoor, A. and Taylor, R. H. (2008) A constrained optimization approach to virtual fixtures for multi-handed tasks. *IEEE International Conference on Robotics and Automation* [online]. Available from: `http://dx.doi.org/10.1109/ROBOT.2008.4543730` [Accessed November 4, 2017].

Kikuuwe, R., Takesue, N., and Fujimoto, H. (2008) A control framework to generate nonenergy-storing virtual fixtures: Use of simulated plasticity. *IEEE Transactions on Robotics* [online]. 24 (4), pp. 781–793. Available from: `http://dx.doi.org/10.1109/TRO.2008.924947` [Accessed December 1, 2018].

Kitagawa, M., Dokko, D., Okamura, A. M., and Yuh, D. D. (2005) Effect of sensory substitution on suture-manipulation forces for robotic surgical systems. *The Journal of thoracic and cardiovascular surgery* [online]. 129 (1), pp. 151–158. Available from: `http://dx.doi.org/10.1016/j.jtcvs.2004.05.029` [Accessed April 7, 2017].

Kragic, D., Marayong, P., Li, M., Okamura, A. M., and Hager, G. D. (2005) Human-Machine Collaborative Systems for Microsurgical Applications. *The International Journal of Robotics Research* [online]. 24 (9), pp. 731–741. Available from: `http://dx.doi.org/10.1177/0278364905057059` [Accessed December 3, 2018].

Kristoffersson, A., Coradeschi, S., and Loutfi, A. (2013) A review of mobile robotic telepresence. *Advances in Human-Computer Interaction* [online]. 2013, pp. 1–17. Available from: `http://dx.doi.org/10.1155/2013/902316` [Accessed November 2, 2017].

Kuang, A. B., Payandeh, S., Zheng, B., Henigman, F., and MacKenzie, C. L. (2004) Assembling virtual fixtures for guidance in training environments. *12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems* [online]. Available from: `http://dx.doi.org/10.1109/HAPTIC.2004.1287223` [Accessed January 8, 2018].

Landis, G. A. (2004) Robots and humans: synergy in planetary exploration. *Acta Astronautica* [online]. 55 (12), pp. 985–990. Available from: `http://dx.doi.org/10.1016/j.actaastro.2004.05.072` [Accessed January 23, 2018].

Lawrence, D. (1988) Impedance control stability properties in common implementations. *IEEE International Conference on Robotics and Automation* [online]. Los Alamitos, CA, USA, available from: `http://dx.doi.org/10.1109/ROBOT.1988.12222` [Accessed July 19, 2018].

Li, M., Ishii, M., and Taylor, R. H. (2007) Spatial Motion Constraints Using Virtual Fixtures Generated by Anatomy. *IEEE Transactions on Robotics* [online]. 23 (1), pp. 4–19. Available from: `http://dx.doi.org/10.1109/TRO.2006.886838` [Accessed April 4, 2017].

Li, M. and Okamura, A. M. (2003) Recognition of operator motions for real-time assistance using virtual fixtures. *11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems* [online]. Available from: `http://dx.doi.org/10.1109/HAPTIC.2003.1191253` [Accessed March 8, 2017].

Liu, R. and Wang, Y.-X. (2012) Auditory Feedback and Sensory Substitution During Teleoperated Navigation. *IEEE/ASME Transactions on Mechatronics* [online]. 17 (4), pp. 680–686. Available from: `http://dx.doi.org/10.1109/TMECH.2011.2119490` [Accessed January 18, 2017].

Lunghi., G., Prades., R. M., and Castro., M. D. (2016) An Advanced, Adaptive and Multimodal Graphical User Interface for Human-robot Teleoperation in Radioactive Scenarios. *Proceed-*

*ings of the 13th International Conference on Informatics in Control, Automation and Robotics* [online]. Available from: `http://dx.doi.org/10.5220/0005971402240231` [Accessed September 15, 2017].

Marayong, P., Li, M., Okamura, A. M., and Hager, G. D. (2003) Spatial motion constraints: theory and demonstrations for robot guidance using virtual fixtures. *IEEE International Conference on Robotics and Automation* [online]. Available from: `http://dx.doi.org/10.1109/ROBOT.2003.1241880` [Accessed September 26, 2017].

Marayong, P. and Okamura, A. M. (2004) Speed-accuracy characteristics of human-machine cooperative manipulation using virtual fixtures with variable admittance. *Human factors* [online]. 46 (3), pp. 518–532. Available from: `http://dx.doi.org/10.1518/hfes.46.3.518.1635` [Accessed December 28, 2018].

Massimino, M. J. and Sheridan, T. B. (1994) Teleoperator performance with varying force and visual feedback. *Human factors*. 36 (1), pp. 145–157 [Accessed December 7, 2017].

McKee, G. T., Brooks, B. G., and Schenker, P. S. (2003) Human-robot interaction for intelligent assisted viewing during teleoperation. *36th Annual Hawaii International Conference on System Sciences* [online]. Available from: `http://dx.doi.org/10.1109/HICSS.2003.1174286` [Accessed August 7, 2017].

McMahan, W., Gewirtz, J., Standish, D., Martin, P., Kunkel, J. A., Lilavois, M., Wedmid, A., Lee, D. I., and Kuchenbecker, K. J. (2011) Tool Contact Acceleration Feedback for Telerobotic Surgery. *IEEE Transactions on Haptics* [online]. 4 (3), pp. 210–220. Available from: `http://dx.doi.org/10.1109/TOH.2011.31` [Accessed February 14, 2018].

Meli, L., Pacchierotti, C., and Prattichizzo, D. (2014) Sensory Subtraction in Robot-Assisted Surgery: Fingertip Skin Deformation Feedback to Ensure Safety and Improve Transparency in Bimanual Haptic Interaction. *IEEE Transactions on Biomedical Engineering* [online]. 61 (4), pp. 1318–1327. Available from: `http://dx.doi.org/10.1109/TBME.2014.2303052` [Accessed August 4, 2018].

Mitra, P. and Niemeyer, G. (2008) Model-mediated Telemanipulation. *The International Journal of Robotics Research* [online]. 27 (2), pp. 253–262. Available from: `http://dx.doi.org/10.1177/0278364907084590` [Accessed January 25, 2017].

Moody, L., Baber, C., and Arvanitis, T. N. (2002) Objective surgical performance evaluation based on haptic feedback. *Studies in health technology and informatics* [online]. 85, pp. 304–310 available from: `http://europepmc.org/abstract/MED/15458106` [Accessed January 12, 2017].

Navkar, N. V., Deng, Z., Shah, D. J., Bekris, K. E., and Tsekos, N. V. (2012) Visual and force-feedback guidance for robot-assisted interventions in the beating heart with real-time MRI. *IEEE International Conference on Robotics and Automation* [online]. Available from: `http://dx.doi.org/10.1109/ICRA.2012.6224582` [Accessed April 1, 2017].

Ni, D., Nee, A. Y. C., Ong, S. K., and Song, A. (2016) Local implicit surface based virtual fixtures in point cloud virtual environment for teleoperation. *2016 10th International Conference on Sensing Technology (ICST)* [online]. Available from: `http://dx.doi.org/10.1109/ICSensT.2016.7796325` [Accessed August 10, 2017].

Niemeyer, G. and Slotine, J.-J. E. (2004) Telemanipulation with Time Delays. *The International Journal of Robotics Research* [online]. 23 (9), pp. 873–890. Available from: `http://dx.doi.org/10.1177/0278364904045563` [Accessed October 1, 2018].

Nolin, J. T., Stemniski, P. M., and Okamura, A. M. (2003) Activation cues and force scaling methods for virtual fixtures. *11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems* [online]. Available from: `http://dx.doi.org/10.1109/HAPTIC.2003.1191325` [Accessed January 6, 2017].

Notheis, S., Hein, B., and Worn, H. (2014) Evaluation of a method for intuitive telemanipulation based on view-dependent mapping and inhibition of movements. *2014 IEEE International Conference on Robotics and Automation* [online]. Available from: `http://dx.doi.org/10.1109/ICRA.2014.6907738` [Accessed October 3, 2018].

Ott, C., Mukherjee, R., and Nakamura, Y. (2010) Unified Impedance and Admittance Control. *IEEE International Conference on Robotics and Automation* [online]. Available from: `http://dx.doi.org/10.1109/ROBOT.2010.5509861` [Accessed June 10, 2018].

Ott, C., Mukherjee, R., and Nakamura, Y. (2015) A Hybrid System Framework for Unified Impedance and Admittance Control. *Journal of Intelligent & Robotic Systems* [online]. 78, pp. 359–375. Available from: `http://dx.doi.org/10.1007/s10846-014-0082-1` [Accessed October 7, 2018].

Pan, Y.-J., Canudas-de-Wit, C., and Sename, O. (2006) A New Predictive Approach for Bilateral Teleoperation With Applications to Drive-by-Wire Systems. *IEEE Transactions on Robotics* [online]. 22 (6), pp. 1146–1162. Available from: `http://dx.doi.org/10.1109/TRO.2006.886279` [Accessed December 20, 2018].

Pandya, A., Reisner, L. A., King, B., Lucas, N., Composto, A., Klein, M., and Ellis, R. D. (2014) *A Review of Camera Viewpoint Automation in Robotic and Laparoscopic Surgery*. Available from: `http://dx.doi.org/10.3390/robotics3030310` [Accessed September 3, 2018].

Passenberg, C., Groten, R., Peer, A., and Buss, M. (2011) Towards real-time haptic assistance adaptation optimizing task performance and human effort. *IEEE World Haptics Conference* [Accessed September 2, 2017].

Pezzementi, Z., Okamura, A. M., and Hager, G. D. (2007) Dynamic Guidance with Pseudoadmittance Virtual Fixtures. *Proceedings 2007 IEEE International Conference on Robotics and Automation* [online]. Available from: `http://dx.doi.org/10.1109/ROBOT.2007.363577` [Accessed November 21, 2018].

Polushin, I. G., Liu, P. X., and Lung, C.-H. (2007) A Force-Reflection Algorithm for Improved Transparency in Bilateral Teleoperation With Communication Delay. *IEEE/ASME Transactions on Mechatronics* [online]. 12 (3), pp. 361–374. Available from: `http://dx.doi.org/10.1109/TMECH.2007.897285` [Accessed January 12, 2017].

Prada, R. and Payandeh, S. (2009) On study of design and implementation of virtual fixtures. *Virtual Reality* [online]. 13 (2), pp. 117–129. Available from: `http://dx.doi.org/10.1007/s10055-009-0115-4` [Accessed May 22, 2017].

Prattichizzo, D., Pacchierotti, C., Cenci, S., Minamizawa, K., and Rosati, G. (2010) Using a Fingertip Tactile Device to Substitute Kinesthetic Feedback in Haptic Interaction. Vol. 6191. Berlin, Heidelberg: Springer Berlin Heidelberg. Chap. 1, pp. 125–130. Available from: `http://dx.doi.org/10.1007/978-3-642-14064-8_19` [Accessed August 12, 2017].

Prattichizzo, D., Pacchierotti, C., and Rosati, G. (2012) Cutaneous Force Feedback as a Sensory Subtraction Technique in Haptics. *IEEE Transactions on Haptics* [online]. 5 (4), pp. 289–300. Available from: `http://dx.doi.org/10.1109/TOH.2012.15` [Accessed November 13, 2018].

Prewett, M. S., Johnson, R. C., Saboe, K. N., Elliott, L. R., and Coovert, M. D. (2010) Managing workload in human–robot interaction: A review of empirical studies. *Computers in Human Behavior* [online]. 26 (5), pp. 840–856. Available from: `http://dx.doi.org/10.1016/j.chb.2010.03.010` [Accessed September 15, 2018].

Quek, Z. F., Schorr, S. B., Nisky, I., Provancher, W. R., and Okamura, A. M. (2015) Sensory substitution of force and torque using 6-DoF tangential and normal skin deformation feedback. *IEEE International Conference on Robotics and Automation* [online]. Available from: `http://dx.doi.org/10.1109/ICRA.2015.7139010` [Accessed June 3, 2018].

Rakita, D., Mutlu, B., and Gleicher, M. (2018) An Autonomous Dynamic Camera Method for Effective Remote Teleoperation. *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction* [Accessed December 20, 2017].

Rosenberg, L. B. (1992) *The Use of Virtual Fixtures as Perceptual Overlays to Enhance Operator Performance in Remote Environments*. Tech. rep. available from: `http://www.dtic.mil/docs/citations/ADA292450` [Accessed February 11, 2017].

Ryden, F. and Chizeck, H. J. (2012) Forbidden-region virtual fixtures from streaming point clouds: Remotely touching and protecting a beating heart. *IEEE/RSJ International Conference on Intelligent Robots and Systems* [online]. Available from: `http://dx.doi.org/10.1109/IROS.2012.6386012` [Accessed October 14, 2017].

Salmeròn-Garcìa, J., Íñigo-Blasco, P., Diaz-del-Rio, F., and Cagigas-Muñiz, D. (2015) A Tradeoff Analysis of a Cloud-Based Robot Navigation Assistant Using Stereo Image Processing. *IEEE Transactions on Automation Science and Engineering* [online]. 12 (2), pp. 444–454. Available from: `http://dx.doi.org/10.1109/TASE.2015.2403593` [Accessed July 1, 2018].

Saltaren, R., Aracil, R., Alvarez, C., Yime, E., and Sabater, J. M. (2007) Field and service applications - Exploring deep sea by teleoperated robot - An Underwater Parallel Robot with High Navigation Capabilities. *IEEE Robotics Automation Magazine* [online]. 14 (3), pp. 65–75. Available from: `http://dx.doi.org/10.1109/MRA.2007.905502` [Accessed June 26, 2017].

Schmidt, G. R., Landis, G. A., and Oleson, S. R. (2012) Human exploration using real-time robotic operations (HERRO): A space exploration strategy for the 21st century. *Acta Astronautica* [online]. 80, pp. 105–113. Available from: `http://dx.doi.org/10.1016/j.actaastro.2012.05.036` [Accessed October 28, 2017].

Schoonmaker, R. E. and Cao, C. G. L. (2006) Vibrotactile force feedback system for minimally invasive surgical procedures. *IEEE International Conference on Systems, Man and Cybernetics* [online]. Available from: `http://dx.doi.org/10.1109/ICSMC.2006.385233` [Accessed October 14, 2017].

Siemens (2020) *siemens/ros-sharp*. Available from: `https://github.com/siemens/ros-sharp` [Accessed March 20, 2020].

Sintorn, I.-M. and Borgefors, G. (2004) Weighted distance transforms for volume images digitized in elongated voxel grids. *Pattern Recognition Letters* [online]. 25 (5), pp. 571–580. Available from: `http://dx.doi.org/10.1016/j.patrec.2003.12.006` [Accessed November 8, 2020].

Sita, E., Horváth, C. M., Thomessen, T., Korondi, P., and Pipe, A. G. (2017) ROS-Unity3D based system for monitoring of an industrial robotic process. *IEEE/SICE International Symposium on System Integration (SII)* [online]. Available from: `http://dx.doi.org/10.1109/SII.2017.8279361` [Accessed June 25, 2017].

Sita, E., Thomessen, T., Pipe, A. G., Dailami, F., and Studley, M. (2018) Robot Companion for Industrial Process Monitoring Based on Virtual Fixtures. *IECON 44th Annual Conference of the IEEE Industrial Electronics Society* [online]. Available from: `http://dx.doi.org/10.1109/IECON.2018.8591105` [Accessed April 18, 2017].

Soleimani, M. A., Sharifi, I., Talebi, H. A., Suratgar, A. A., and Koochaki, F. (2014) Forbidden region virtual fixture for nonlinear teleoperation systems with variable time delays. *Second RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)* [online]. Available from: `http://dx.doi.org/10.1109/ICRoM.2014.6990883` [Accessed April 12, 2017].

Srimathveeravalli, G., Gourishankar, V., and Kesavadas, T. (2007) Comparative Study: Virtual Fixtures and Shared Control for Rehabilitation of Fine Motor Skills. *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems* [online]. Available from: `http://dx.doi.org/10.1109/WHC.2007.31` [Accessed June 15, 2018].

Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A., and Goodrich, M. (2006) Common metrics for human-robot interaction. *1st Annual Conference on Human-Robot Interaction* [online]. Available from: `http://dx.doi.org/10.1145/1121241.1121249` [Accessed December 2, 2017].

Stone, R. T., Bisantz, A., Llinas, J., and Paquet, V. (2009) Improving Tele-robotic Navigation through Augmented Reality Devices. *Human Factors and Ergonomics Society Annual Meeting*

*Proceedings* [online]. 53 (18), pp. 1432–1436. Available from: `http://dx.doi.org/10.1518/107118109X12524443347751` [Accessed December 28, 2018].

Technologies, U. (2017) *Unity*. `https://unity3d.com/` and `https://unity3d.com/public-relations`. Available from: `https://unity.com/` [Accessed September 29, 2017].

Tumerdem, U. (2017) Multilateral teleoperation under asymmetric time delays: L 2 stability and robustness. *International Journal of Advanced Robotic Systems* [online]. 14 (3). Available from: `http://dx.doi.org/10.1177/1729881417710148` [Accessed January 15, 2017].

Visell, Y. (2009) Tactile sensory substitution: Models for enaction in HCI. *Interacting with Computers* [online]. 21 (1), pp. 38–53. Available from: `http://dx.doi.org/10.1016/j.intcom.2008.08.004` [Accessed January 7, 2018].

Wagner, C. R., Stylopoulos, N., and Howe, R. D. (2002) The role of force feedback in surgery: analysis of blunt dissection. *Proceedings 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems* [online]. Available from: `http://dx.doi.org/10.1109/HAPTIC.2002.998943` [Accessed August 15, 2018].

Weber, B., Sagardia, M., Hulin, T., and Preusche, C. (2013) Visual, vibrotactile, and force feedback of collisions in virtual environments: Effects on performance, mental workload and spatial orientation. *Virtual Augmented and Mixed Reality. Designing and Developing Augmented and Virtual Environments* [online]. Available from: `http://dx.doi.org/10.1007/978-3-642-39405-8_28` [Accessed April 20, 2017].

Weede, O., Mönnich, H., Müller, B., and Wörn, H. (2011) An intelligent and autonomous endoscopic guidance system for minimally invasive surgery. *IEEE International Conference on Robotics and Automation (ICRA)* [online]. Available from: `http://dx.doi.org/10.1109/ICRA.2011.5980216` [Accessed October 12, 2017].

Westebring van der Putten, E., Goossens, R. H. M., Jakimowicz, J. J., and Dankelman, J. (2008) Haptics in minimally invasive surgery - a review. *Minimally Invasive Therapy & Allied Tech-*

*nologies* [online]. 17 (1), pp. 3–16. Available from: `http://dx.doi.org/10.1080/13645700701820242` [Accessed December 24, 2017].

Wilde, M., Chua, Z. K., and Fleischner, A. (2014) Effects of Multivantage Point Systems on the Teleoperation of Spacecraft Docking. *IEEE Transactions on Human-Machine Systems* [online]. 44 (2), pp. 200–210. Available from: `http://dx.doi.org/10.1109/THMS.2013.2295298` [Accessed September 24, 2018].

Wu, J., Shi, Y., Huang, J., and Constantinescu, D. (2012) Stochastic stabilization for bilateral teleoperation over networks with probabilistic delays. *Mechatronics* [online]. 22 (8), pp. 1050–1059. Available from: `http://dx.doi.org/10.1016/j.mechatronics.2012.08.010` [Accessed June 27, 2018].

Xia, T., Kapoor, A., Kazanzides, P., and Taylor, R. (2011) A constrained optimization approach to virtual fixtures for multi-robot collaborative teleoperation. *IEEE/RSJ International Conference on Intelligent Robots and Systems* [online]. Available from: `http://dx.doi.org/10.1109/IROS.2011.6095056` [Accessed July 12, 2017].

Xia, T., Leonard, S., Deguet, A., Whitcomb, L., and Kazanzides, P. (2012) Augmented reality environment with virtual fixtures for robotic telemanipulation in space. *IEEE/RSJ International Conference on Intelligent Robots and Systems* [online]. Available from: `http://dx.doi.org/10.1109/IROS.2012.6386169` [Accessed November 5, 2017].

Yanco, H. A., Drury, J. L., and Scholtz, J. (2004) Beyond usability evaluation: Analysis of human-robot interaction at a major robotics competition. *Human-Computer Interaction* [online]. 19 (1-2), pp. 117–149. Available from: `http://dx.doi.org/10.1207/s15327051hci1901&2_6` [Accessed March 28, 2017].

Yang, J., Kamezaki, M., Sato, R., Iwata, H., and Sugano, S. (2015) Inducement of visual attention using augmented reality for multi-display systems in advanced tele-operation. *IEEE/RSJ International Conference on Intelligent Robots and Systems* [online]. Available from: `http://dx.doi.org/10.1109/IROS.2015.7354135` [Accessed February 18, 2017].

Yang, Y., Hua, C., Li, J., and Guan, X. (2017) Fixed-time Coordination Control for Bilateral Telerobotics System with Asymmetric Time-varying Delays. *Journal of Intelligent and Robotic Systems: Theory and Applications* [online]. 86 (3-4), pp. 1–20. Available from: `http://dx.doi.org/10.1007/s10846-016-0454-9` [Accessed December 2, 2017].

Ye, Y. and Liu, P. X. (2010) Improving Trajectory Tracking in Wave-Variable-Based Teleoperation. *IEEE/ASME Transactions on Mechatronics* [online]. 15 (2), pp. 321–326. Available from: `http://dx.doi.org/10.1109/TMECH.2009.2020733` [Accessed April 24, 2018].

Yu, W., Alqasemi, R., Dubey, R., and Pernalete, N. (2005) Telemanipulation Assistance Based on Motion Intention Recognition. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* [online]. Available from: `http://dx.doi.org/10.1109/ROBOT.2005.1570266` [Accessed July 8, 2017].

Zhu, D., Gedeon, T., and Taylor, K. (2011) Exploring Camera Viewpoint Control Models for a Multi-Tasking Setting in Teleoperation. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* [online]. Available from: `http://dx.doi.org/10.1145/1978942.1978952` [Accessed July 17, 2017].

Zilles, C. B. and Salisbury, J. K. (1995) A constraint-based god-object method for haptic display. *IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots* [online]. Available from: `http://dx.doi.org/10.1109/IROS.1995.525876` [Accessed March 1, 2017].

Zimber, F., Horváth, C. M., Thomessen, T., and Franke, J. (2016) Control strategy for an industrial process monitoring robot. *IEEE/SICE International Symposium on System Integration (SII)* [online]. Available from: `http://dx.doi.org/10.1109/SII.2016.7844082` [Accessed January 5, 2018].