


H2020-ICT-25-2016-2017

HYbrid FLying rollIng with-snakE-aRm robot for contact inSpection

HYFLIERS

D3.2

Human-friendly remote-control modalities and interfaces of a hyper-redundant robotic arm

Contractual date of delivery	31 Jul 2020, agreed postponed date 31 Dec 2020
Actual date of delivery	31 Dec 2020
Editor(s)	Vincenzo Lippiello (CREATE)
Author(s)	Vincenzo Lippiello (CREATE), Jonathan Cacace (CREATE)
Workpackage	WP3
Estimated person-months	15
Dissemination level	PU
Type	R
Version	1.0
Total number of pages	36

Abstract:

This document reports the progress made on Task 3.3 of the HYFLIERS project. This deliverable focusses on the development of natural and effective human-robot interaction interfaces to allow human operators to remotely control the hyper redundant manipulator needed to perform NonDestructive tests.

Keywords:

2D mapping. 3D mapping. Deliverable. Force control. Human-robot interaction. Hyper-redundant robotic arm. Navigation. Pipe inspection. Snake robot. Task composition. Tube inspection. UT inspection. UT probe. Visual feedback control. Visual inspection.

Executive summary

This document summarizes the activities made in Task 3.3 for Work Package 3 of HYFLIERS project. In this context, effective human-robot interaction techniques are developed to allow human operators to control in a natural way the hyper redundant robotic system developed in Task 3.1 and presented in the deliverable D3.1. In particular, the designed arm consists of 22 degrees of freedom (DOF) and the control of such a system is impractical for a human operator during remote inspection tasks. For this reason, specific control interfaces have been designed to simplify the role of the operator in the control phases. The contributions reported in this document have been investigated by CREATE group and the main topics are listed in the following:

- *Stabilization of the Wheeled Unmanned Aerial Vehicle (WUAV) over the pipe*: the human operator directly interacts with the end effector of the manipulator. Such a manipulator is rigidly attached to the wheeled mobile base developed in WP2. Once the robotic system is landed on the industrial pipe to inspect, it must have the capability to stabilize over it during the manipulation task. For this reason, a nonlinear model predictive controller has been developed to implement a stabilization method for the HYFLIERS WUAV on the industrial pipe supporting the operator and preventing the UAV from slipping down from the pipe.
- *Variable admittance control method for remote Human-Robot Interaction*: a novel mixed-initiative control strategy based on variable admittance control has been deployed to improve the effectiveness in the remote control of the hyper-redundant arm. In this context, the human operator is able to interact with the robotic arm using a haptic interface and an admittance force controller is implemented to move the gripper of such device. In this setup, the operator perceives a force feedback that helps him/her to reach target waypoints modifying the gain of the admittance controller.
- *Human-Robot Collaboration with operator intention estimation*: the remote control of the hyper-redundant system is achieved merging the outcomes of an autonomous controller with the input provided by the human operator by means of a haptic device. In this context, a proactive approach to assess human intentions considering a set of possible target points over the pipe and the state of the system with the interventions of the human operator is exploited by the autonomous system of the robot to assist the operator in the execution of the NonDestructive measure regulating on-line the autonomy level of the robotic system.
- *Multimodal Teleoperation supported by force feedback*: to interact with the surface of the pipe, the human operator is mainly interested in the control of the position of the end effector of the hyper-redundant system. For this reason, a set of different information are fused together to hold the orientation of the probe with respect to the surface of the pipe. In addition, a set of force feedback are designed to help the operator to avoid dangerous configuration of the hyper-redundant arm or unbalance the WUAV. In particular, different task errors have been considered with different priority levels. Considering the state of each task error, a relative force feedback prevents the operator from move the arm in wrong directions.

The approaches discussed in this deliverable have been tested on laboratory mocap with similar robotic platforms. In addition, such techniques have been extensively tested in different simulation environments such as Gazebo ROS and CoppeliaSim.

Abbreviations and symbols

2D	Two-Dimensional
3D	Three-Dimensional
BS	Behaviour-based System
DOF	Degree Of Freedom
EMG	ElectroMyoGraphy
FTS	Force and Torque Sensor
HLC	High-Level Control System
HRI	Human Robot Interaction
LLC	Low-Level Control System
LQR	Linear Quadratic Regulator
LSTM	Long Short-Term Memory
LTM	Long Term Memory
MPC	Model Predictive Control
NDT	Non-Destructive Test
NLP	NonLinear Control Problem
NMPC	Nonlinear MPC
OS	Operating System
PID	Proportional-Integral-Derivative
RNN	Recurrent Neural Network
UAV	Unmanned Aerial Vehicle
WM	Working Memory
WP	Work Package, WayPoint
WUAV	Wheeled Unmanned Aerial Vehicle

Table of Contents

List of Figures

1. Introduction

The goal of the WP3 is to design, develop and support pipe inspection tasks exploiting a hybrid wheeled robotic manipulator able to safely operate in a partially observable environment. This deliverable takes in consideration the work done in Tasks 3.1 and 3.2 in which a mobile platform able to stabilize over an industrial pipe and a hyper-redundant manipulator have been designed and deployed and controlled, respectively. Differently, this deliverable focuses on the development of feasible and interfaces to allow human operators to interact and control the robotic arm in a simple and safe way.

First, a nonlinear model predictive control will be presented to allow the stabilization of the mobile base of the HYFLIERS robot upon the industrial pipe. Then, the use of a haptic device is employed to control the end effector of the mobile base to perform NDT (non-destructive test) measures. In particular, mixed-initiative approaches have been deployed to mix the cognitive capabilities of the human operator with the precise motion of the autonomous system of the robot. In this context, a variable admittance controller has been implemented to properly control the gripper of the haptic device devoted to the control of the robotic arm. In addition, a method to estimate the overall human intention is implemented. Finally, to assist the human operator in the teleoperation of the hyper-redundant arm, a multimodal Human-Robot Interaction framework has been deployed in which the operator receives feedback from the robotic system to avoid reaching dangerous configuration. In particular, different task errors are calculated considering system conditions like how the hyper-redundant arm is moving closer to the mechanical joint limits, or how WUAV is stable upon the pipe. A priority order is imposed on the force feedback generated by each task, preventing the operator from moving the snake arm in faulty conditions. At the same time, a multimodal fusion of the different arm sensors allows the system to autonomously control the orientation of the end effector to be aligned with the pipe to inspect, while the operator is able to control the position.

2. Nonlinear model predictive control for the stabilization of a wheeled unmanned aerial vehicle on a pipe

NonDestructive measurement tests in refineries are currently performed by human operators climbing huge and costly scaffolding. Several robotic commercial solutions are currently (or are ready-to-be) available like the APPELLIX drone (www.apellix.com), the Texo Drone Survey and Inspection platform (www.texodroneservices.co.uk/blog/56), and the Ronik Inspectioneering UT device (www.inspectioneering.eu). The listed devices are drones equipped with a stick, or a telescopic arm, at whose tip is mounted an ultrasonic thickness sensor. As highlighted in several research works, the contact between the stick and the surface may destabilize the robot, because the contact arises a torque at the center of mass of the drone. Several solutions address such a problem. The impacts destabilizing the aerial device have been addressed by combining mechanics, with the use of active and passive joints, and control strategy or, differently, the impacts may also be absorbed through passive vacuum-cup technology. The problem of carrying out large forces in contact with the environment with an underactuated drone is instead investigated in other works through the use of LQR (linear quadratic regulator) -optimized state feedback on the roll and yaw angle.

In case the inspections measurements are performed in a crowded and narrow environment, and the drone is too bulky to stay close to the surface to be inspected, a long reach aerial manipulator can be employed. The solution proposed within the HYFLIERS project for NDT is innovative. The hybrid aerial-ground robot is efficient since it does not require to fly during all the inspection operation, but it can land and navigate on pipes. The stabilization of a hybrid aerial-ground robot on a pipe is indeed a new problem.

As the aim of this work, the crucial element in climbing and perching robots is the stability of robotic systems on the surface of a telegraph pole or a wall. Both the mechanical structure and the employed control methods are critical for the stability of the system. A proportional-integral-derivative (PID) controller can be implemented to govern the motion of a robotic system on a pipe, but the wheels of the robotic system are assumed not to slip or to generate desired pressure between a wheeled robotic system and a non-smooth vertical wall and a horizontal ceiling or finally, to achieve high-speed climbing of a quadrupedal robot on a wooden telegraph pole.

Other approaches solving such a kind of task make use of the Nonlinear Model Predictive Control (NMPC) for an elastic tool interaction of the drone with the ceiling. In general, in the last decades, Model Predictive Control (MPC) and NMPC dominate the process industry for a large variety of applications. Therefore, the use of NMPC over the classical control approaches allow the possibility to explicitly include the slipping conditions and saturation of the actuators within the control design. The NMPC can be categorized into continuous-time and discrete-time settings. The discrete-time setting, employed in this report, is preferred due to the simplicity of notation, the conceptual formulation, and the intrinsic discrete nature of the elaboration systems. Various approaches are then investigated to guarantee the stability of a discrete-time NMPC. The NMPC with a finite time horizon is also employed to control a multiplicative noise stochastic system, where the input delay is considered, and stability is guaranteed by introducing two terminal inequalities.

2.1. Dynamic model

In this section, the dynamic model of the wheeled UAV is presented. Actually, a different setup has been used with respect to our final robotic system. However, this fact does not affect the overall effectiveness of the proposed method. Some simplifications are made to derive the dynamic model of the system briefly described in the introduction.

1. The wheels are in a fixed position, and they can move transversely to the pipe to constrain the motion of the WUAV only around the circumference.
2. The propellers are in a fixed position such as the combined effect of the centre of mass of the WUAV is always parallel to the tangent to the pipe.
3. The wheels and the pipe are rigid, and a Coulomb friction model is assumed, while the rolling friction is negligible.
4. The wheels do not slip on the pipe.
5. The inertia of the wheels and the propellers are negligible compared to the inertia of the WUAV rotation around the centre of the pipe.

With the above assumptions, the resulting system is sketched in Figure 1.

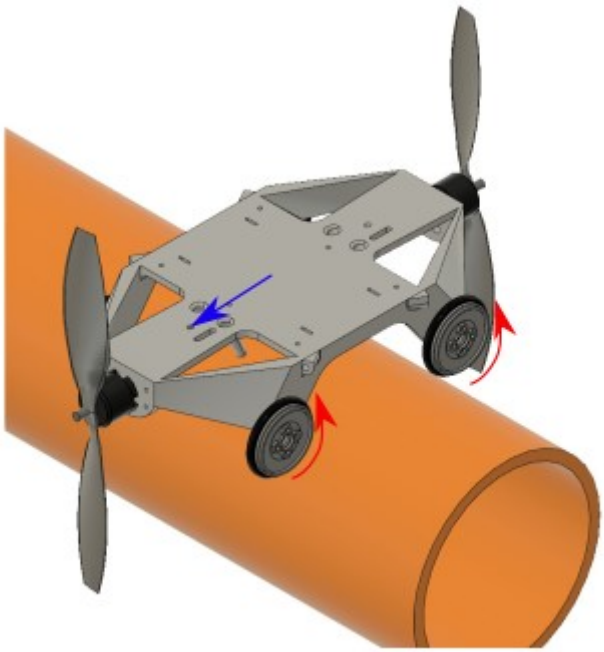


Figure 1. Three-dimensional sketch of the WUAV system on a pipe. In red, the actuation torques on the wheels. In blue, the effect of the propeller effects on the centre of mass of the WUAV.

Since the system is symmetric to the vertical plane cutting the WUAV at its centre of mass, a two-dimensional (2-D) representation can be employed to further simplify the derivation of the dynamic model. Therefore, with reference to Figure 2, the origin O of the inertial reference frame Σ_B is put in the center of the pipe, while the Y-axis is directed along the gravity direction, and the X-axis is directed along the radius of the pipe and parallel to the ground.

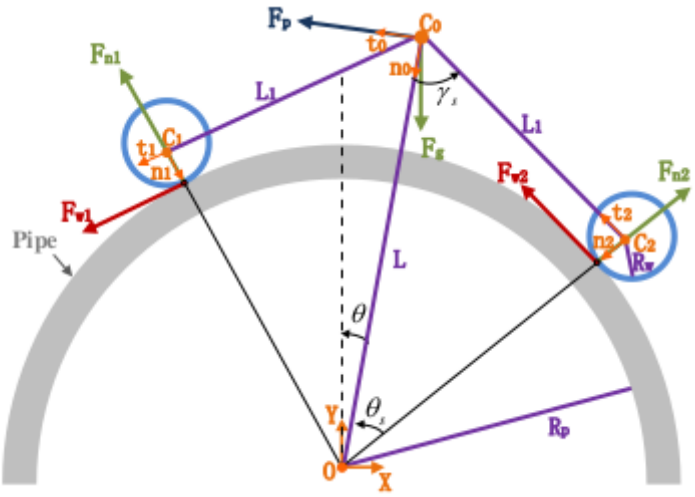


Figure 2: 2-D sketch of the WUAV system on a pipe, with the illustration of the symbols employed to derive the dynamic model of the WUAV and its constraints.

Three coordinate frames, Σ_i with i=0,1,2, are instead attached to the WUAV. The frame Σ_O is placed at the center of mass of the WUAV, with $n_o \in R^2$ the unit vector always directed towards the center

of the pipe, and $t_o \in R^2$ the unit vector perpendicular to n_o and parallel to the tangent to the pipe. The frames Σ_1 and Σ_B are placed at the center of the wheels, with $n_1$and n_2 the unit vectors always directed towards the center of the pipe, and t_1 and t_2 the unit vectors perpendicular to n_1 and n_2, respectively, and parallel to the tangent to the pipe.

The expression of the mentioned vectors in Σ_B:

$$\begin{aligned}
\boldsymbol{t}_0 &= \begin{bmatrix} -\cos(\theta) & -\sin(\theta) \end{bmatrix}^T, \\
\boldsymbol{n}_0 &= \begin{bmatrix} \sin(\theta) & -\cos(\theta) \end{bmatrix}^T, \\
\boldsymbol{t}_1 &= \begin{bmatrix} -\cos(\theta_s + \theta) & -\sin(\theta_s + \theta) \end{bmatrix}^T, \\
\boldsymbol{n}_1 &= \begin{bmatrix} \sin(\theta_s + \theta) & -\cos(\theta_s + \theta) \end{bmatrix}^T, \\
\boldsymbol{t}_2 &= \begin{bmatrix} -\cos(\theta_s - \theta) & \sin(\theta_s - \theta) \end{bmatrix}^T, \\
\boldsymbol{n}_2 &= \begin{bmatrix} -\sin(\theta_s - \theta) & -\cos(\theta_s - \theta) \end{bmatrix}^T.
\end{aligned}$$

The inputs to the system are the lateral thrust $F_p \in R$ created by the propellers, and the torques of the wheels τ_{w1} and τ_{w2}.

The following vectors can be thus defined in Σ_B:

$$\begin{aligned}
\boldsymbol{F}_p &= \begin{bmatrix} -F_p\cos(\theta) & -F_p\sin(\theta) \end{bmatrix}^T, \\
\boldsymbol{F}_{w1} &= \begin{bmatrix} -F_{w1}\cos(\theta_s + \theta) & -F_{w1}\sin(\theta_s + \theta) \end{bmatrix}^T, \\
\boldsymbol{F}_{w2} &= \begin{bmatrix} -F_{w2}\cos(\theta_s - \theta) & F_{w2}\sin(\theta_s - \theta) \end{bmatrix}^T,
\end{aligned}$$

Finally, let $L \in R^+$ be the distance between the center of the pipe and the center of mass of the WUAV, $\theta \in R$ the angle of the WUAV with respect to the Y-axis, $\gamma_s \in R$ and θ_s two angles depending on the geometry of the device and the pipe, and $R_p \in R^+$the radius of the pipe. Based on the Newton-Euler theory, the dynamic model of the system is formulated as:

$$\ddot{\theta} = \frac{1}{mL}\left[mg\sin(\theta) + F_p + \frac{\cos(\theta_s)}{R_w}(\tau_{w1} + \tau_{w2})\right],$$

where $m \in R^+$ is the mass of the WUAV and g is the gravity acceleration.

2.2. Stabilization constraints

From the walking robot's domain, it is useful to import the concept that the resultant of all the forces acting on the robot must remain strictly inside the support polygon to ensure the stability against gravity. Denoting with $F_s = F_g + F_p + F_{w1} + F_{w2}$ the resultant force at the center of mass of the WUAV, expressed in Σ_B with $F_g = [0 \ -mg]^T$, these dynamical conditions imply that F_s must lie within the cone of angle $2\gamma_s$.

Taking inspiration from the Coulomb's friction law, the dynamic conditions can be defined through the following constraints expressed in Σ_B:

$$\begin{cases} \boldsymbol{F_s}^T \boldsymbol{n_0} > 0, \\ |\boldsymbol{F_s}^T \boldsymbol{t_0}| \leq \tan(\gamma_s) \boldsymbol{F_s}^T \boldsymbol{n_0}. \end{cases}$$

The assumption 4 (Section 2.1) related to the non-slipping condition of the wheels must be ensured by the controller. Such a condition also ensures that the WUAV does not detach from the pipe while moving. The procedure to find the non-slipping constraint is here extended for the case of a wheeled robot on a circular tube. The presence of assumption 1 implies that there is not any lateral force acting on the wheels pushing them along the pipe. Therefore, with reference to Figure 2, the total force acting on each wheel can be decomposed into the tangential force F_{wi} with i=1,2, and the normal force $F_{n_i} \in R^2$. From the Coulomb model, the non-slipping condition can be expressed as $|F_{wi_i}| \leq \mu F_{ni}$, where $F_{ni} = ||F_{ni}||$ and $\mu \in R^+$ is the static friction coefficient between each wheel and the pipe, whose value depends on the materials at contact. The problem is how to relate F_{Ni} with the other involved forces. Writing the force and moment balance in Σ_B yields:

$$\begin{cases} F_{n1}\boldsymbol{n}_1 + F_{n2}\boldsymbol{n}_2 = \boldsymbol{F}_p + \boldsymbol{F}_g, \\ F_{n1}L_1 - F_{n2}L_1 = 0, \end{cases}$$

with $L_1 \in R^+$ the distance between the center of each wheel and the center of mass of the WUAV. By manipulating the previous equation with some trigonometric formula, computing F_p in one equation and folding it into the others, it is possible to obtain the following expression for the normal force $F_n = F_{n1} = F_{n2} = mgcos(\theta) \,/\, 2cos(\theta_s)$. Hence, replacing the obtained expression into the Coulomb model yields:

$$|F_{wi}| \leq \mu \frac{mg\cos(\theta)}{2\cos(\theta_s)},$$

and, since such a condition is true to maintain the pure rolling assumption, it is possible to write equivalently

$$\begin{cases} |\tau_{w1}| \leq \frac{\mu m g R_w cos(\theta)}{2cos(\theta_s)}, \\ |\tau_{w2}| \leq \frac{\mu m g R_w cos(\theta)}{2cos(\theta_s)}. \end{cases}$$

2.3. Stabilization constraints

The main idea of the NMPC algorithm is the repetitive solution of an optimal nonlinear control problem (NLP). Given the measured state with the state vector, at each controller time step k=0, $T_s, 2T_s, \cdots$, with $T_s \in R^+$, the discretized version of the dynamic model is employed by the NMPC to predict to future behaviour of the system $\hat{x}(j)$ with j = 0, …, N-1. where N denotes the finite prediction horizon. The prediction sequence is useful to determine the optimal control sequence minimizing the NLP while satisfying the constraints with a given control input vector. In this case, the control and the prediction horizons are coincident. The peculiarity of the NMPC is to apply only the first element of the sequence to the real system.

The NLP is repeatedly solved from each new acquired measure. Then, the NLP at each sampling time k, with initial state x_0 , can be sketched out as:

$$\min_{\overline{\boldsymbol{u}}(0)\ldots\overline{\boldsymbol{u}}(N-1)} J = \sum_{j=0}^{N-1} V\left(\widehat{\boldsymbol{x}}(j), \overline{\boldsymbol{u}}(j)\right) + V_N\left(\widehat{\boldsymbol{x}}(N)\right)$$

subject to

$$\begin{cases} \widehat{\boldsymbol{x}}(j+1) = \boldsymbol{f}\left(\widehat{\boldsymbol{x}}(j), \overline{\boldsymbol{u}}(j)\right), \\ \boldsymbol{x}_\star \in \mathbb{X}_0 \subseteq \mathbb{X}, \\ \overline{\boldsymbol{u}} \in \mathbb{U}, \\ \boldsymbol{F_s}^T \boldsymbol{n_0} > 0, \\ |\boldsymbol{F_s}^T \boldsymbol{t_0}| \leq \mu_0 \boldsymbol{F_s}^T \boldsymbol{n_0}, \\ |u_2| \leq \dfrac{\mu m g R_w cos(\theta)}{2cos(\theta_s)}, \\ |u_3| \leq \dfrac{\mu m g R_w cos(\theta)}{2cos(\theta_s)}, \end{cases}$$

with the initial state $\widehat{x_0}$(j) = $\underline{x_0}$ the discretized version of the dynamic model:

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} x_1 + T_s x_2 \\ x_2 + T_s \left(\frac{g \sin(x_1)}{L} + \frac{u_1}{mL} + \frac{(u_2+u_3)\cos(\theta_s)}{mLR_w} \right) \end{bmatrix}$$

the cost function to minimize:

$$V(\widehat{\boldsymbol{x}}(j), \overline{\boldsymbol{u}}(j)) = (\widehat{\boldsymbol{x}}(j) - \boldsymbol{x}_\star)^T \boldsymbol{Q} (\widehat{\boldsymbol{x}}(j) - \boldsymbol{x}_\star) + (\overline{\boldsymbol{u}}(j) - \boldsymbol{u}_\star)^T \boldsymbol{R} (\overline{\boldsymbol{u}}(j) - \boldsymbol{u}_\star),$$

with Q and R positive definite matrices, the desired equilibrium point of the system *x* and the reference control input *u.* This formulation allows the system to reach the final state of the finite horizon within a neighbourhood of the reference point as demonstrated in the simulation use case defined to demonstrate its effectiveness, as discussed in the next section.

2.4. System testing

Two case studies are developed in this section to validate the proposed NMPC and test its robustness. Within the technical report, the first preliminary experiment on a prototype without propellers is shown. The former case study includes white noise on the measurements, parametric uncertainty on the mass of the vehicle, a one-step delay in the controller. The latter case study considers a different physics engine to simulate the system dynamics. The model parameters of the WUAV are retrieved from the first prototypes under development within HYFLIERS, while the specifications on the parameters of the pipe are given from the oil and gas facilities involved in the project. The numerical tests are performed on a standard PC through the MATLAB/SIMULINK software environment under the R2018a distribution. The real dynamic system is numerically simulated through the ode45 solver with a maximum time step which is ten times lower the Ts employed to run the discrete-time optimal control problem. A first-order filter with a time constant of 0.2 s is implemented for u1 before

applying it to the real system to simulate the slower propellers dynamics of the WUAV compared to the wheels. All the presented simulations have a duration of 8 s. The robustness of the designed NMPC controller is tested by considering a white noise, whose standard deviation is 3.16x10^-4, on the measurements of x from the simulated real system; a one-step time-delay of the controller; a parametric uncertainty about the mass of the WUAV, in particular, the mass considered by the controller is 5.5 kg (10% more of the value employed to simulate the real system). The sampling time is Ts=0.01 s. The objective of this test is twofold. first, the investigation of how N affects the control design; then, through a reasonable value of N, different initial positions of the WUAV are considered. In the first case a set of numerical simulations for different time horizons have been performed and the obtained simulation are reported in Figure 3. In the second case, a set of numerical simulation have been performed considering different initial conditions and, in this case, results are reported in Figure 4.

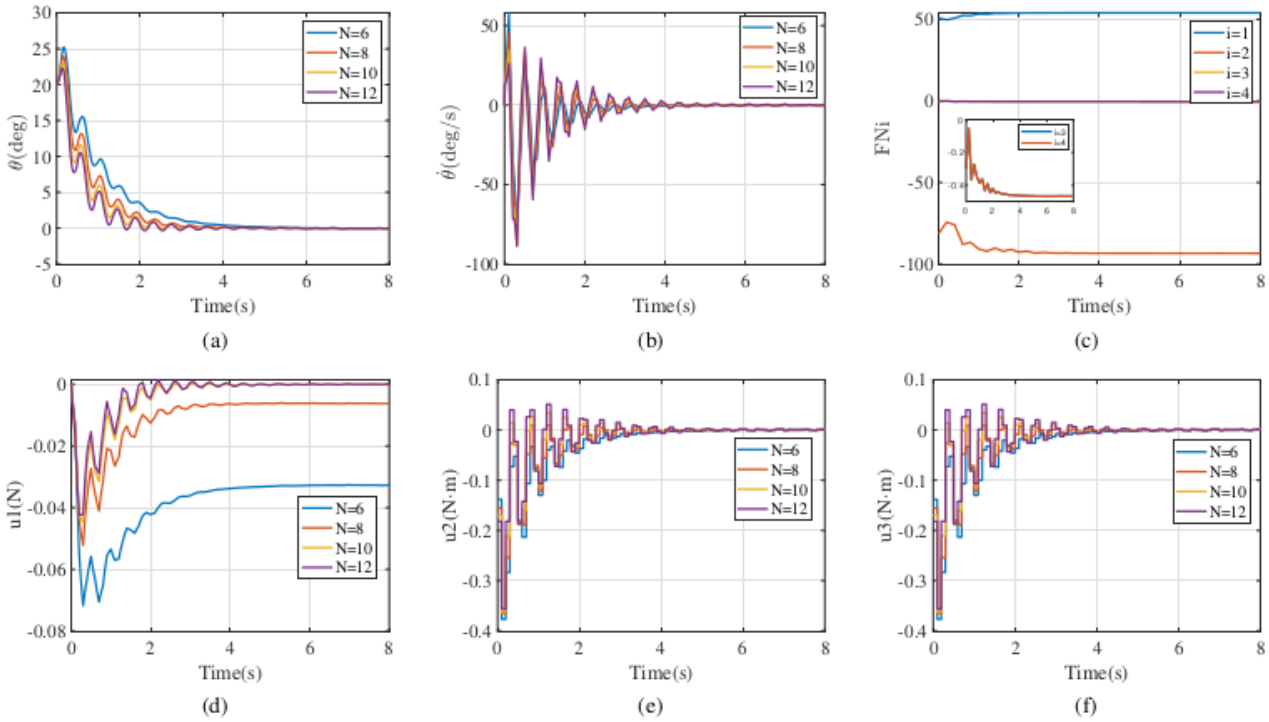


Figure 3: Case study 1: numerical test. Numerical simulations for different time horizons. (a)-Time history of θ. (b)-Time history of θ̇. (c)-Time histories to check the fulfilment of the constraints (4) and (6). (d)-(f) Time histories of the control inputs.

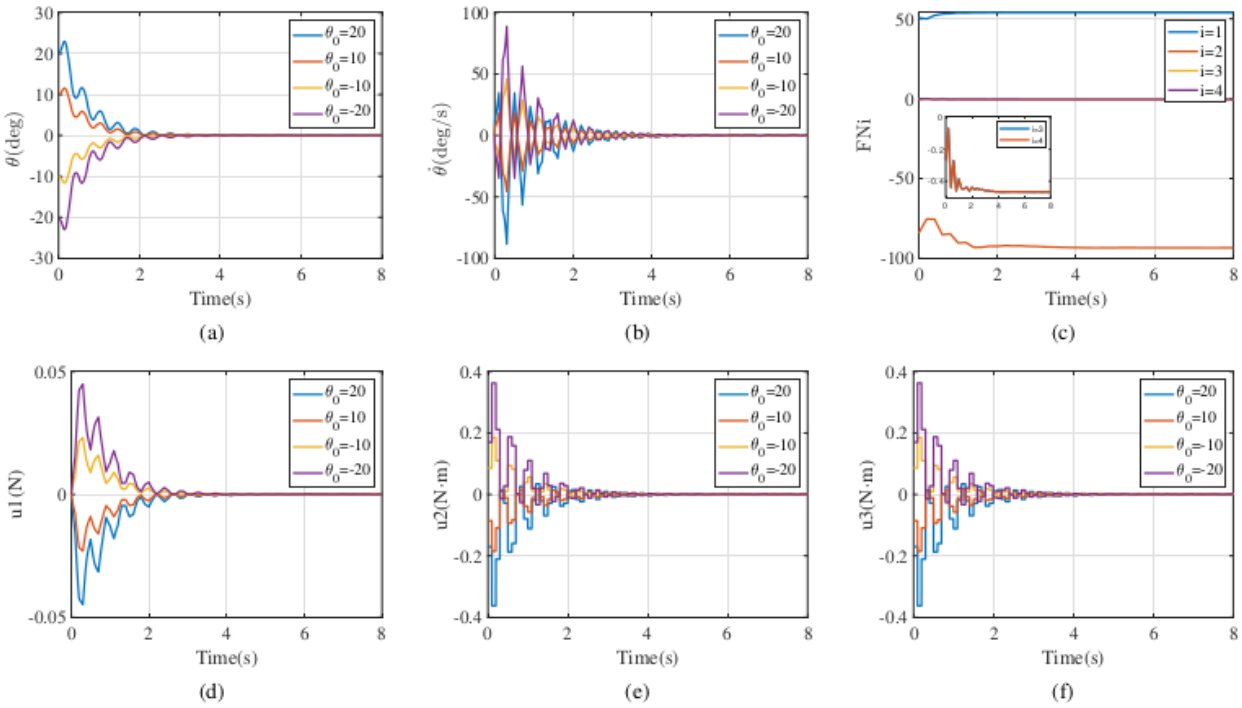


Figure 4: Case study 1: numerical test. Numerical simulations for different initial conditions. (a)-Time history of θ. (b)-Time history of θ̇. (c)-Time histories to check the fulfilment of the constraints (4) and (6). (d)-(f) Time histories of the control inputs.

A second case study has been performed considering a set of dynamic simulation. In particular, a dynamic simulation is carried out by using the Coppelia Sim simulator and MATLAB. The former is not employed as a visualizer of the simulation, but its physics engine simulates the system dynamics. The latter implements the proposed NMPC algorithm, and it is connected with Coppelia Sim through suitable remote application programming interfaces. A proper time synchronization has been ensured using the synchronous modality provided inside Coppelia Sim, with the control loop running at 25 *ms*. In Coppelia Sim the simulated system is dynamically enabled considering physically consistent dynamic parameters for the rover mass and inertia and the wheel-pipe friction. In particular, the following parameters change from the first case study because they are given by the CAD model in Figure 1 that has been imported into Coppelia Sim: m = 6.1 kg, μ = 0.8, L=0.11 m, R_w = 0.02 m, γ_s = 70.6 *deg*, θ_s = pi/4 *rad*. The wheels are actuated with a torque controller applied on the simulated joint motors. The Coppelia Sim model is not planar, but it is symmetric to the vertical plane cutting the WUAV at its centre of mass. Therefore, the torques of the wheels obtained from the NMPC algorithm running in MATLAB are equally split between each of the symmetric pair of wheels in Coppelia Sim. On the other hand, a force has been applied to the rover to simulate the propeller lift force. The gains of the NLP have been practically tuned by a trial and error procedure. The control horizon is set to 7. The presented simulations have a duration of 28 seconds. Other parameters are left unchanged.

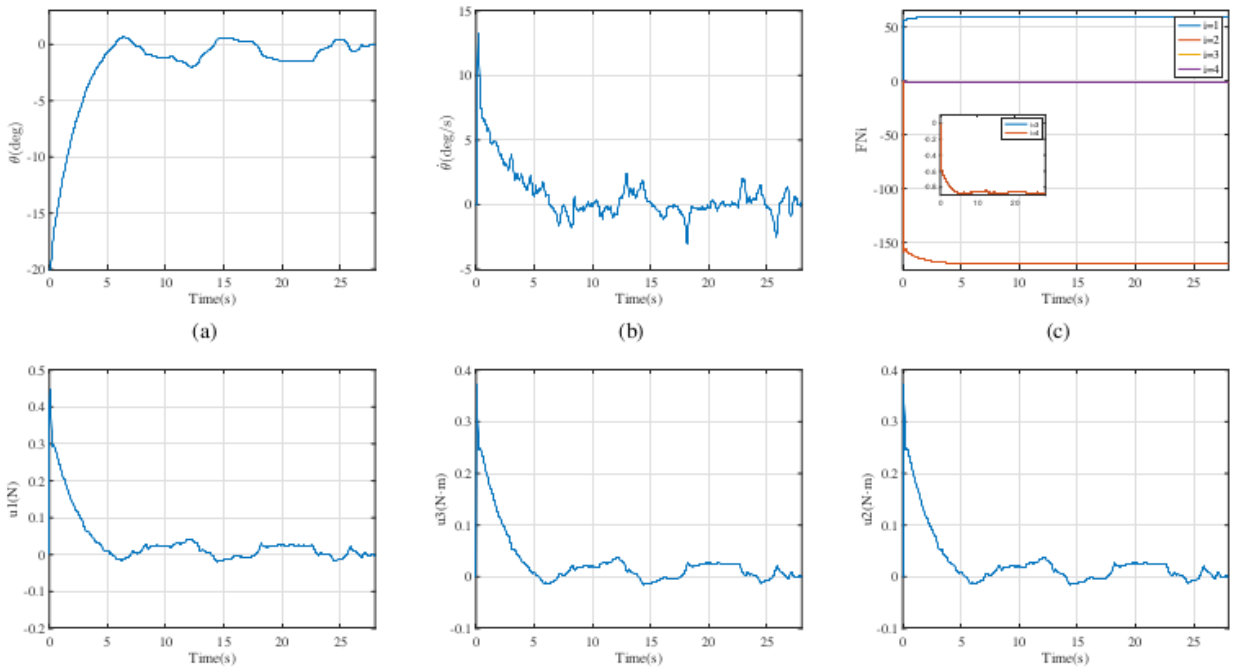


Figure 5: Case study 2: dynamic simulation. (a)-Time history of θ. (b)-Time history of θ̇. (c)-Time histories to check the fulfilment of the constraints (4) and (6). (d)-(f) Time histories of the control inputs.

The obtained simulations are depicted in Figure 5, with the legend mentioned above. The initial condition is set to -20 *deg.* with initial control input equal to zero. For this reason, it is possible to appreciate a peak at the initial time instant in the control inputs depicted in Figure 4 (d)-(f), and in the resulting velocity shown in Figure 4 (b).

The plots show that the controller can stabilize the WUAV on the pipe also during the carried out dynamic simulation. Besides, the control inputs remain within the considered bounds, and the other constraints are fulfilled.

3. Human-friendly remote-control modalities and interface of the hyper-redundant system

Beside the stabilization of the rover upon the industrial pipe, the human operator must be able to easily interact with the end effector of the robot in order to accomplish the NDT measure. In this setup, the human operator interacts with the robotic system using a 3 DOF haptic device as shown in Figure 6. In this context, the work done in Task 3.2 is exploited to enable a safe control of the end effector of the hyper-redundant arm allowing the robot to reconfigure its structure to improve the stability of the system.

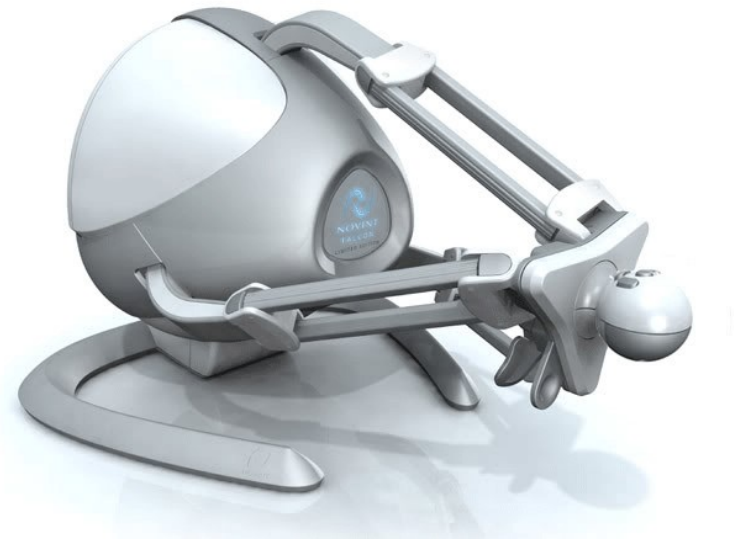


Figure 6: Novint Falcon 3 DOF haptic device used for by the operator to control the hyper-redundant system.

The methodologies developed in this document use the haptic device to provide a feedback to the human operator during the inspection task. In particular, the human operator is able to specify a new position for the end effector of the device while the orientation is properly controlled by the autonomous system to point on the inspection surface. At the same time, considering the state of the system and the goal to achieve, a force feedback is received from the operator through the haptic interface. Mainly, to provide the force feedback a cartesian position controller has been implemented on the end effector of the haptic device, in order to provide information about the better position in which robotic arm should be driven.

The overall system architecture is depicted in Figure 7, where the human operator controls the snake robot with the use of the haptic device, receiving from it a force feedback depending on the executing task.

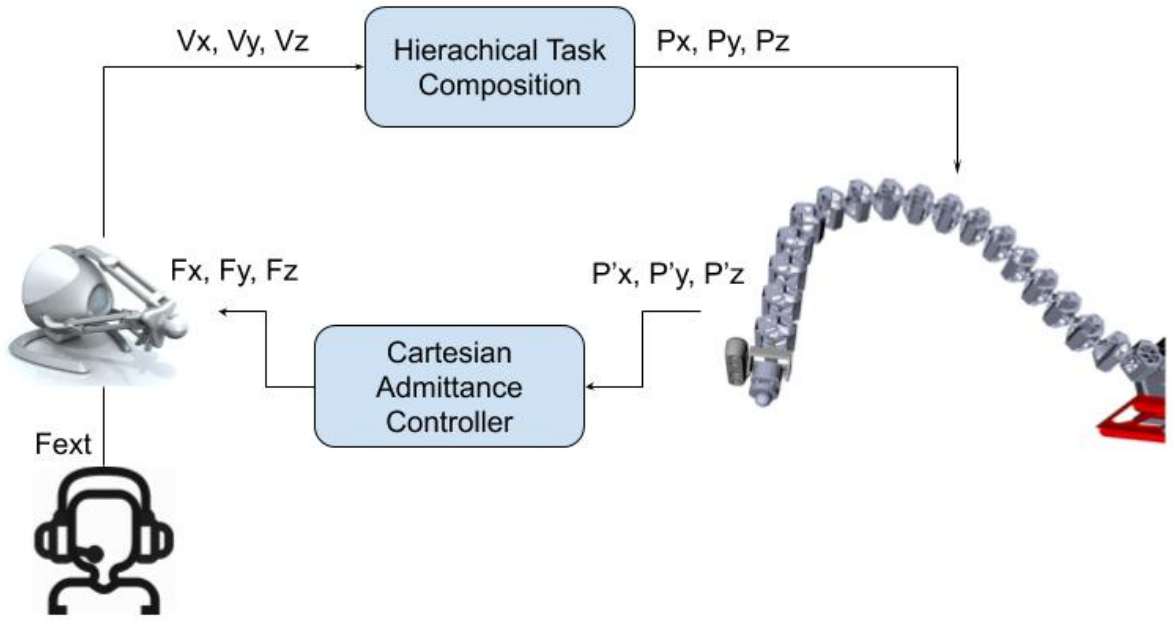


Figure 7: Human-Robot Interaction system architecture.

In this context, two main contributions have been deployed. A variable admittance control method based on virtual fixtures, in which the human operator receives a haptic feedback helping him/her to reach a proper destination point placed over the pipe to inspect, and another method exploiting haptic feedback and position control to guess operator intention estimation, assisting him/her to reach the more plausible waypoint.

3.1. Variable admittance control based on virtual fixtures for HRI

In Cartesian admittance control, human operators can physically interact with a robot by directly applying forces on its end-effector that reacts like a mass-spring-damper system characterized by three parameters: virtual mass, damping and stiffness. In a variable admittance control, these parameters can be on-line regulated in order to increase the effectiveness of the manipulation system. In our setup, the end effector is represented by the gripper of the haptic device.

In the literature, many works proposed different techniques to adjust the admittance control gains. For instance, the damping of the system can be regulated according to the velocity of the manipulator, while the time derivative of the applied external forces can also be taken into account. In some works, the admittance gains are regulated according to the state of the human operator. Following this approach, the virtual damping of the system is regulated by means of electromyography (EMG) information used to track the co-activation of the human muscle. Instead, in other contributions the arm stiffness of the operator is used to regulate the virtual stiffness of the robot during teleoperation tasks. Finally, authors also exploit a Neural Network to implement a variable admittance controller. Differently from these approaches, the current method proposes to regulate damping and stiffness according to distance of the manipulator from a given target point along a virtual reference path.

Target points of a manipulation tasks can be identified considering a set of virtual fixtures. The benefit of admittance regulation with virtual fixtures during cooperative manipulation has been demonstrated in several studies. The proposed control architecture is illustrated in Figure 8. The motion of the robot is managed by the Variable Admittance Control module, whose goal is to actuate the robot considering the velocity data injected via the haptic device by the human operator. As for the hyper-redundant snake like manipulator, we assume to control the position of its end effector, relying on the hierarchical task composition developed in the WP3 of the HYFLIERS project.

We also assume that the (external) forces acting on the robotic body are directly estimated by the haptic device itself by means of integrated sensors. During the execution of a task, a set of virtual fixtures (virtual waypoints and virtual paths) are generated by the Virtual Fixtures Manager module to guide and assist the operator in reaching the desired targets. These fixtures are also exploited to regulate the virtual stiffness and damping of the admittance controller and to select the desired virtual position of the gripper of the end effector. As already stated, we assume that the Human Operator can physically interact with the gripper of the haptic device by moving it along arbitrary trajectories. During the interaction, we assume that the human applies a force F_h on the gripper and perceives a force feedback F_{ext}. The dynamic relationship between the applied forces and the motion of the robot is established by means of the admittance control schema:

$$m\ddot{x} + d\dot{x} + kx = F$$

that in our case can be specialized as follows:

$$M_d(\ddot{X}_c - \ddot{X}_d) + D_d(\dot{X}_c - \dot{X}_d) + K_d(X_c - X_d) = F_t$$

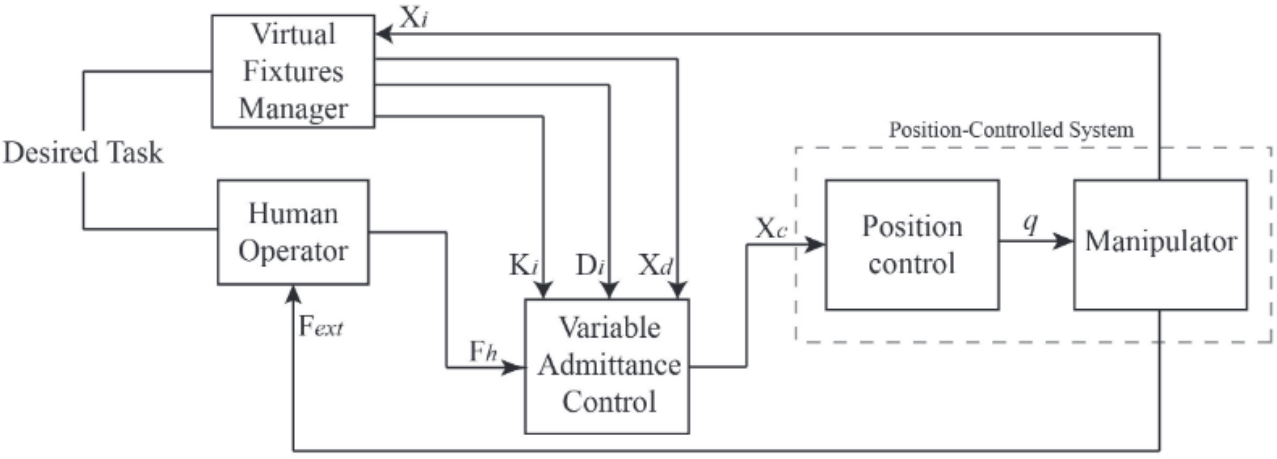


Figure 8: Variable admittance control architecture.

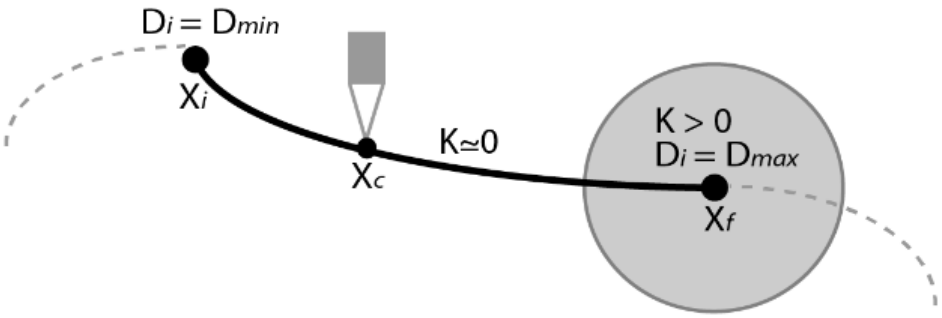
Where M_d, D_d, K_d are positive definite diagonal matrices representing the desired virtual inertia, damping and stiffness, respectively. Following this formula, the output of the Variable Admittance Control module is the compliant position command X_c for the Position-Controlled System given the

destination X_d. Typically, in free space manipulation, the stiffness term K_d is omitted avoiding holding the robot motion to specific workspace locations. In our approach, we assume that the system is provided with a set of target waypoints to be reached in a predefined order to accomplish given tasks. To assist the operator in reaching these points the Virtual Fixture Manager module generates a set of virtual paths, adjusting the damping and stiffness according to the position of the manipulator along the virtual guides. In particular, the higher is the distance of the manipulator from the destination point, the higher are the parameters K_d and D_d of the admittance controller, attracting the robot towards the destination point X_d. In addition, another attraction mechanism becomes active when the manipulator is brought far from the virtual path in order to guide the operator back to the planned route.

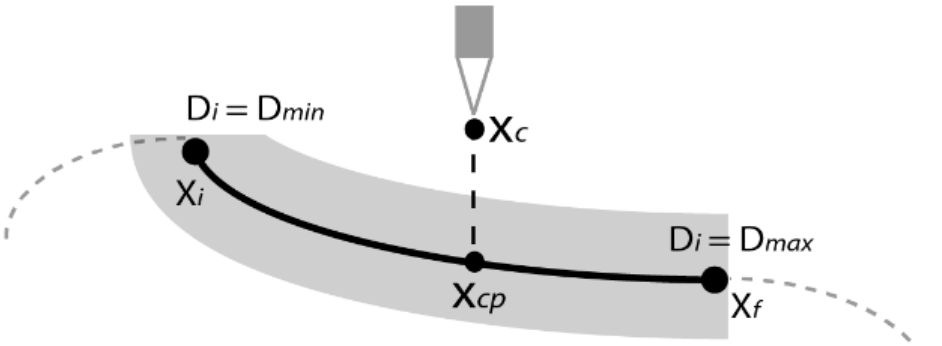
In this setup, we assume that the operator has the task to move the robot end-effector towards the three-dimensional (3D) Cartesian position, $X_f = (x_f, y_f, z_f)$ of the workspace starting from the position $X_i = (x_i, y_i, z_i)$, while the current position of the manipulator is $X_c = (x_c, y_c, z_c)$. A virtual path $VP_i = [X_i; X_f]$, connecting starting and ending points, is generated as shown in Figure 9. This path could be generated by optimizing geometrical proprieties such as the distance from obstacles, the path length, the curvature and so on. Once the virtual path is generated, the operator can move the robot end-effector to reach the desired targets. In this context, the proposed variable admittance control framework is responsible to increase or decrease the resistance of the gripper according to its position along the virtual path. as show in Figure 10. In particular, at the start of the virtual segment the damping of the system is set to its minimum value allowing the operator to interact with the manipulator with fast movements and low physical effort. The value of the damping parameter increases as the robot approaches the target point, following the sigmoid function. Moreover, an elastic force is generated when the manipulator is in the target point proximity (Figure 9(a)) in order to assist the human during the final phase of the end-effector positioning. This attractive force is activated only in the last region of the virtual path, as illustrated in Figure 9(a), where the attractive region is represented by the grey circle. Inside this area, the parameter K_i increases, reaching the maximum value in X_f. As an additional guidance mechanism, when the operator moves the manipulator away for a defined virtual path, he/she will receive a haptic feeling of the deviation between the position of the end-effector and the virtual path, in order to be guided back towards the planned route. For this reason, we implemented a trajectory adherence mechanism to constrain robot motion to the planned path as shown in Figure 9(b). In this context, whenever the operator brings the robot far from the assigned path, the attraction point of the admittance controller will become the X_{cp} point, representing the closest point between the end-effector and the generated virtual path. More specifically, the attraction point $X_d = (x_d, y_d, z_d)$ is calculated as follows:

$$X_d = \begin{cases} X_f & \text{if } \|X_c - X_{cp}\| < \tau \\ X_{cp} & \text{otherwise} \end{cases}$$

Where X_{cp} is the closest point along the virtual path, X_c is the current position of the end effector and τ is a suitable distance threshold.



(a) Manipulator along the virtual path: the Damping parameter increases until X_f is reached from the manipulator. By approaching the target point X_f an attraction force is generated.



(b) Manipulator far from the virtual path: an adherence force attracts the manipulator on the virtual path.

Figure 9: Virtual fixtures when manipulator moves along (up) and far from (down) the path.

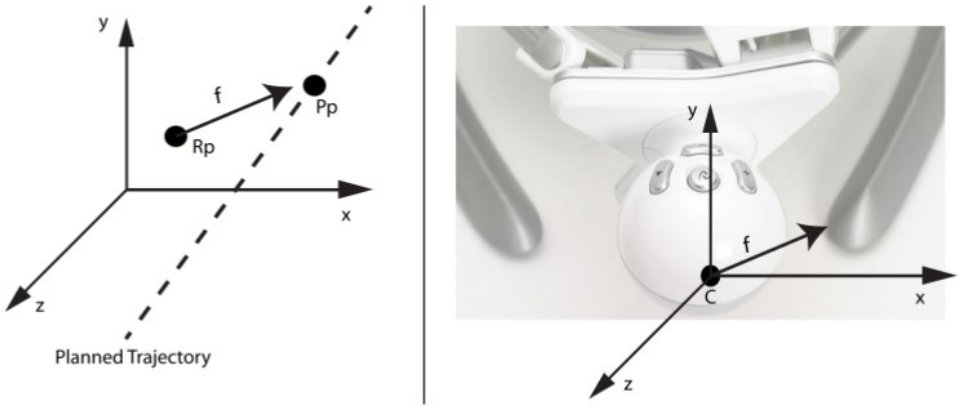


Figure 10: Force feedback and motion control on the haptic device.

Let's discuss how the admittance control parameters are calculated according to the position of the manipulator along the virtual path. For this purpose, we introduce a logistic sigmoid function, whose general form is described by the following equation:

$$\mathscr{L}(t) = \left(\frac{V_{max}}{1 + e^{-w_1 * (t - w_2)}} \right)$$

The logistic function is characterized by three parameters:

- V_{max}, which represents the maximum value of the curve.
- w_1, that regulates the growth rate of the curve.
- w_2, representing the inflection point the curve.

Basically, these parameters enable us to control the offset of the curve along the x axis and its slope. In this work, we deploy the logistic function to regulate the damping and the stiffness of the admittance controller. In particular, for each position of the manipulator along the virtual path we define the following matrices:

$$D_i = \begin{bmatrix} d_x & 0 & 0 \\ 0 & d_y & 0 \\ 0 & 0 & d_z \end{bmatrix}, K_i = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix}$$

where, each element of the diagonal matrices represents the admittance gain relative to each Cartesian axis. In the one-dimensional case of the *x* axis component, the gain value is calculated as follows:

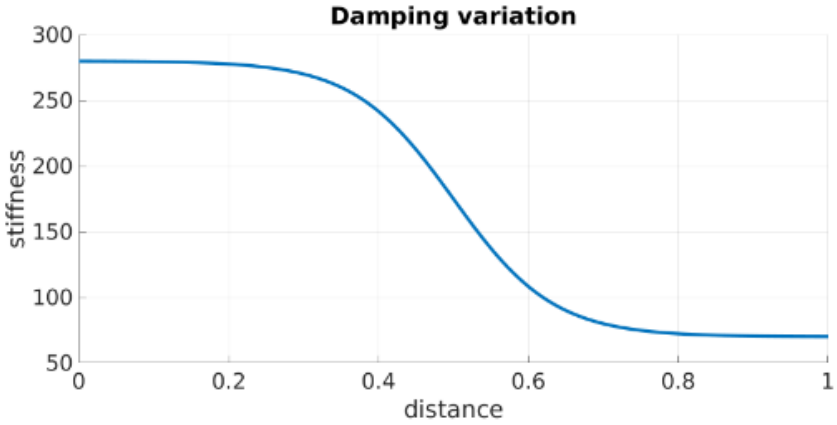
$$\mathscr{L}(e_x) = \left(\frac{v_{max} - v_{min}}{1 + e^{(-w_1/l) * (e_x - l/w_2)}} \right) + v_{min},$$

where $e_x = |x_{cx} - x_{fx}|$ represents the distance between the position of the manipulator and the target position, while $l = |x_{ix} - x_{fx}|$ is for the length of the active virtual path. In this context, the l value is used to normalize the logistic curve parameters with respect to the length of the virtual path. Notice that, in our setting, we need high gain values for small distances to the target and low gain values for high distances, this curve can be obtained by changing the sign of w_1. As for the damping variation, the parameters are selected in order to keep the sigmoid's midpoint in the middle of the virtual path with an associated smooth slope, as shown in Figure 11. This way, damping is smoothly adjusted along the virtual path. In addition, the v_{max} and v_{min} values are suitably selected in order to obtain, respectively, an over-responsive and precise co-manipulation system. In contrast, in the case of stiffness variation, the parameters are set so that the sigmoid's midpoint is closer to the end of the path, with a steep slope (see Figure 11(b)), while the minimum value of the stiffness logistic function is zero. This way, when the manipulator is far from the attraction area of the target point, this gain nullifies the spring propriety of the mechanical system. Instead, the stiffness is rapidly enhanced in

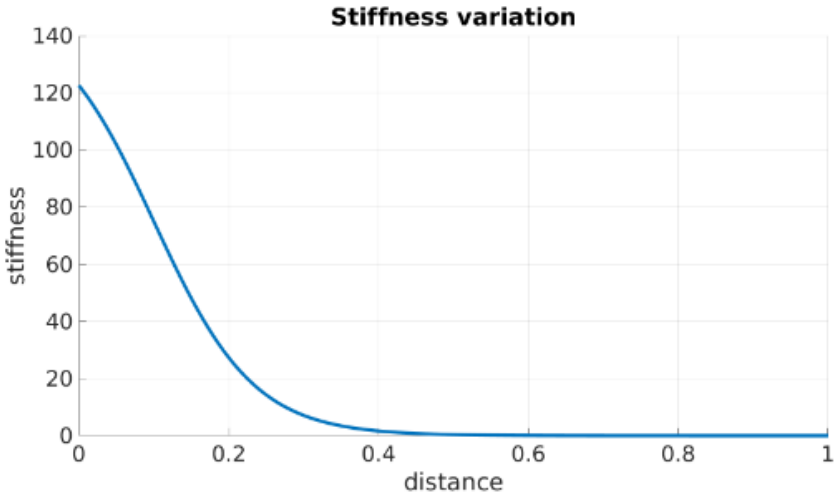
the close proximity of the target point. Finally, in order to address discontinuities that could be generated by the switching of the attraction point when the manipulator is far from the virtual path, a time-vanishing smoothing term has been introduced. Specifically, assuming that the switching of the attraction point starts at time t= 0, the stiffness value is computed as follow:

$$d_x(t) = d_{x_a}(t) + e^{1/\gamma}(d_{x_p}(0) - d_{x_a}(0))$$

where γ is a time constant determining the duration of the transition phase during the switching, while d_{xa} an d_{xp} are the stiffness values related to the parameters of the system acting before and after the target point switching.



(a) Logistic curve with parameters: $v_{min} = 70$, $v_{max} = 280$, $w_1 = 2$, $w_2 = 15$,



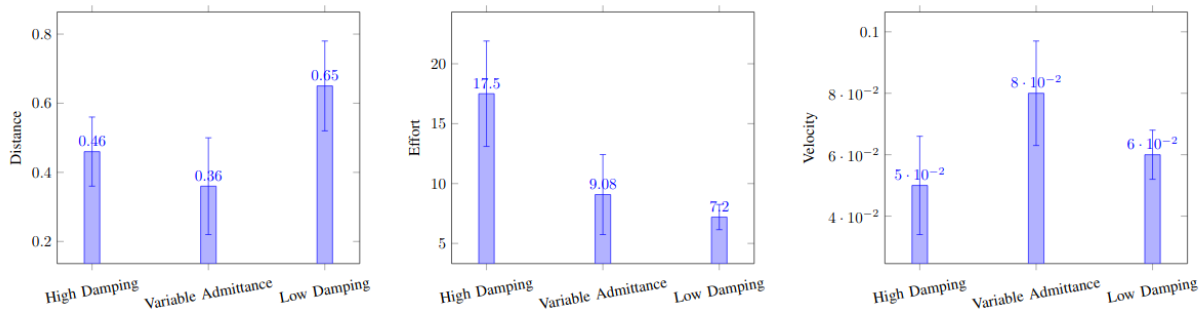
(b) Logistic curve with parameters: $v_{min} = 0$, $v_{max} = 125$, $w_1 = 15$, $w_2 = 10$,

Figure 11: Damping (up) and stiffness (down) variation along a virtual path.

In order to assess the effectiveness of the proposed system, we defined a collaborative manipulation case study where a human worker should physically guide the hyper-redundant arm in order to reach a set of predefined target points in the workspace. In this scenario, our aim is to show that the proposed framework effectively trades-off precision and robot compliance. For this purpose, we compared the system performance considering three different settings:

- High Damping, where a fixed admittance controller is deployed with high damping. In this case, we tuned the damping in order to have a heavy and precise system.
- Low Damping, where a fixed admittance controller is deployed with low damping. In this case, we tuned the damping in order to have a stable, but over-responsive system
- Variable Admittance, which is the framework proposed in this deliverable.

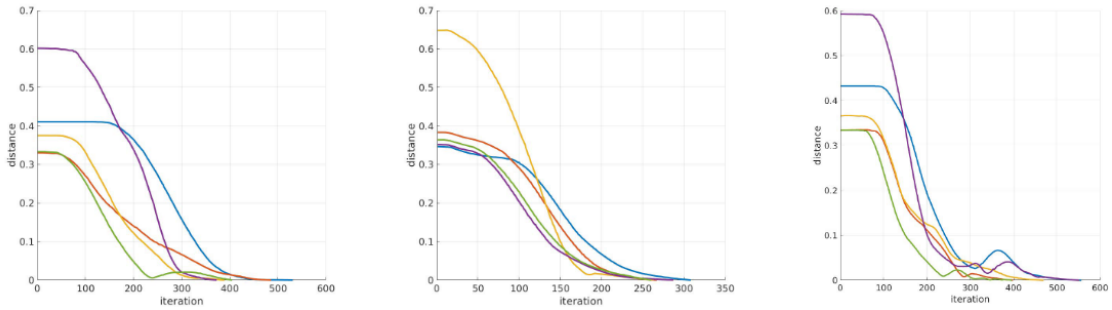
The High Damping and Low Damping systems are here used as baselines to highlight how the proposed framework performs with respect to these two opposite setups. In order to test the system, we involved a group of 10 users, 8 of them male and 2 females, aged between 25 and 35 years old not expert of the system. Each tester was asked to perform repetitive tries of the task in the three settings. Specifically, for each test, the users are to cross a set of 3 waypoints in a pre-defined order. During the experiments, testers are not aware about this order: for each reached waypoint, a new virtual path towards the next location is generated.



(a) Mean of the distance covered by the manipulator

(b) Mean of the effort needed to move the manipulator

(c) Mean of the velocity of the end-effector



(d) Operator trajectory convergence with high-damping

(e) Operator trajectory convergence with variable admittance control

(f) Operator trajectory convergence with low-damping

Figure 12: Evaluation results.

Each subject was asked to perform 3 times the planned route, for a total of 9 virtual path navigation. For each trial, when a new virtual path is generated the operative mode is randomly chosen among the three settings specified above. This way, since users are not aware about the active operative mode, the learning effect is avoided. Tests have been performed using a simulation model of the hyper-redundant robotic system imported in Gazebo simulator, running on GNU/Linux OS (operating system) and exploiting the control framework developed in Task 3.3. In the workspace, the maximum distance between two waypoints is 70cm. During the trials, we monitored different variables. In particular, we considered the effort needed by the operator to complete the manipulation, the velocity of the end-effector, and the distance covered by the manipulator. The operator effort is measured by the norm of the force that he/she exerts on the manipulator. In the bar graphs illustrated in Figure 12, we report the mean of these value (along with the associated variance). In particular, Figure 12(a) shows mean of the distance (in meters), covered by the robot end-effector along all the tests, Figure 12(b) reports mean of the user effort (in Newton), finally, Figure 12(c) shows the comparison between the end-effector mean velocity (in meters per seconds) of the three systems. The collected results show that the proposed variable admittance approach outperforms high and low damping settings in terms of distance and velocity, at the cost of an effort that is slightly higher than the one needed in the low damping configuration. Therefore, the experimental assessment supports the hypothesis that the proposed framework enables the human to easily and effectively manoeuvre the robot in the workspace. We can also assess the operator ability to reach the target point in the three settings. For this purpose, in Figures 12(d), 12(e), 12(f), we plot the trends of the distance to the target during target approach for a representative set of trials, 5 for each modality. In these graphics, we can observe that in the low-damping configuration the operator approach is not very precise, because associated with several corrective oscillations in the proximity of the target point. Instead, we have an oscillation reduction in the variable admittance configuration which is comparable with the one provided by the high damping setting. As expected, the variable admittance system provides even smoother trajectories towards the target points, this effect can be explained by the combined effector of high damping and target attraction in the target proximity.

3.2. Human-Robot Collaboration for the execution of shared NDT measures

The human robot interaction approach discussed in section 3.1 has been empowered with the possibility to predict and estimation the overall intentions of the operator during the inspection task. In particular, a collaborative system has been deployed to assist the operator during the manipulation. Again, the output of the autonomous system is sent to the operator through its haptic device in form of force feedback. In this context, the collaborative system is structured in two main control layers working at different levels of abstraction. The High-Level Control System (HLC) is a deliberative layer responsible for task generation, decomposition, orchestration and interaction. The Low-Level Control System (LLC) is concerned with the actual execution of the primitive operations selected by the HLC, while maintaining the robotic system compliant with respect to the human interventions. The Executive System is an HLC module which manages the orchestration of multiple collaborative tasks taking into account both the environmental changes and the human activities. Tasks are hierarchically structured and collaborative in that they can mention both human and robot operations at different levels of abstraction. During task execution, the human operator can physically interact

with the robot using the haptic device and these interventions are simultaneously interpreted at the different layers of the architecture. Depending on the task, the environmental context, and the human interventions, the Executive System (top-down) proposes a set of primitive operations/processes also called Behaviours that compete for the execution (Contention). Each proposed behaviour is associated with a target position and an activation value, the latter representing an attentional weight, which summarizes the relevance of that activities given the current executive state. The target estimation module generates a trajectory using a local trajectory planner for each proposed target and assesses it with respect to the current human guidance in order to estimate the most aligned with respect to the human interventions. The classification results, along with the associated attention weights, are then exploited to influence behaviour selection (Contention) along with the associated target position for the robot. Finally, the LLC implements a shared controller aimed at mixing the inputs generated by the human operator with the ones needed to perform robot motion (Shared force). As discussed before, the admittance controller is implemented to help the operator in setting the input for the robotic arm. A sketch of the presented architecture is depicted in Figure 13.

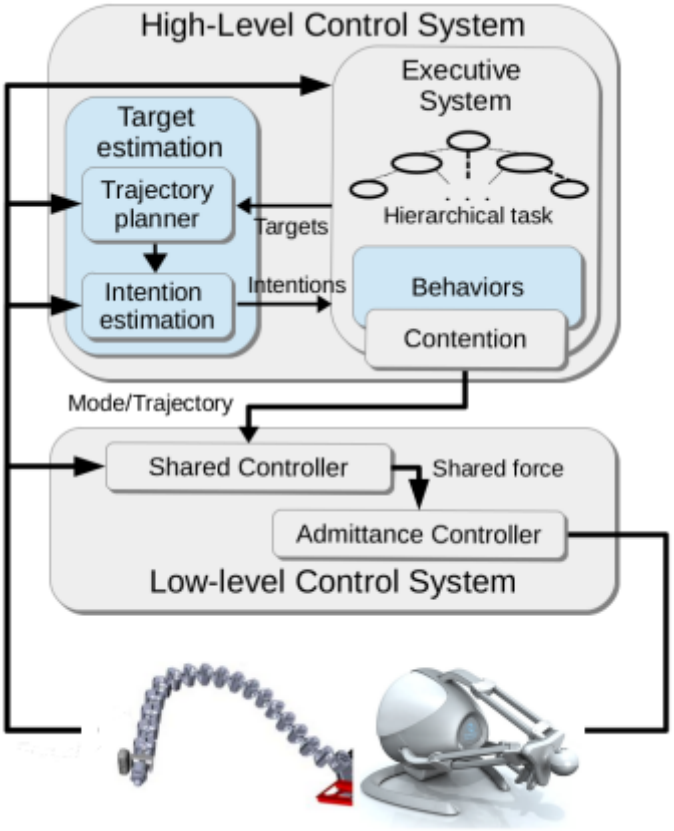


Figure 13: Collaborative system architecture

In the proposed architecture the Executive System is responsible for task retrieving, decomposition, monitoring, orchestration, and regulation. Specifically, we rely on a supervisory attention framework for human-robot collaboration. In this setting, the executive system is decomposed into an Attentional Executive System and an Attentional Behaviour-based System. The first one manages the execution of hierarchically structured tasks along with the associated activations (top-down attentional regulations); the latter collects the active robot processes (behaviours), each associated with an

activation value (obtained as a combination of top-down and bottom-up attentional regulations). A representation of the Executive System is proposed in Figure 14.

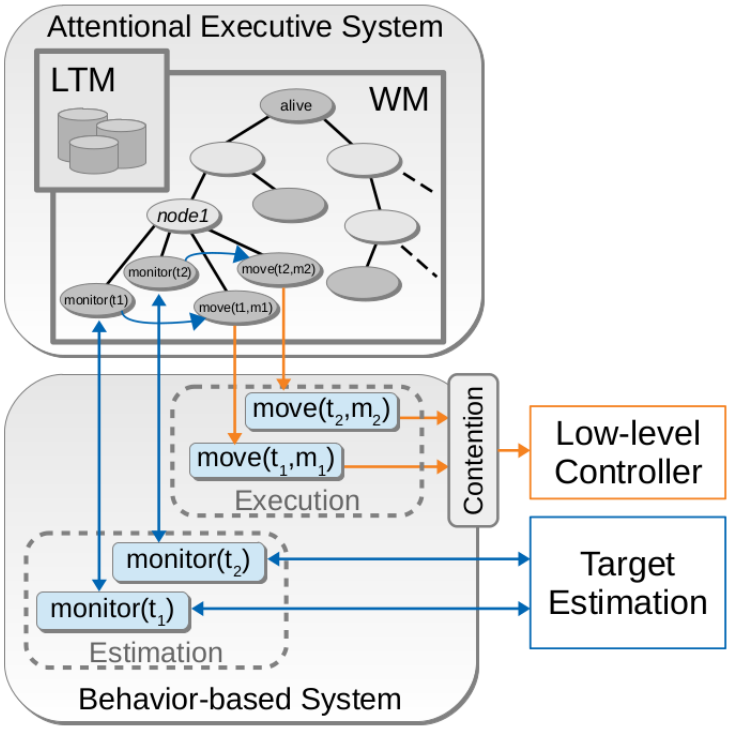


Figure 14: The Executive System manages the execution of multiple hierarchically structured tasks.

In Figure 14 we can highlight three main components: a Long Term Memory (LTM), a Working Memory (WM) and a Behaviour-based System (BS). The LTM collects the system procedural knowledge, i.e., the specification of the tasks available to the robot. A task can be either abstract (to be further decomposed) or concrete (a real sensorimotor process). Each task is defined in the LTM by a predicate (m, l, p), where m is the name of the task, l is a list of m_i subtasks along with associated enabling conditions r_i (releasers), while p is a postcondition used to check task accomplishment. The WM is a data structure which collects hierarchically decomposed tasks instantiated and allocated for the execution. The task set in WM along with the associated state variables characterize the current executive state of the system. The WM is represented by an annotated rooted directed graph (r, B, E), whose nodes in B represent allocated tasks/subtasks, E are parental relations among subtasks, while r is the root process that manages the WM structure. Each node b is represented as a 5-tuple ($m_b, r_b, p_b, x_b, \mu_b$), where m_b is the name of the allocated task, r_b and p_b represent the task precondition and postcondition respectively, x_b is the set of sub-behaviors generated by m_b, while μ_b is an activation value assigned to the task. Leaves in the WM structure correspond to attentional behaviors devoted to the execution of sensorimotor processes. The BS collects all the allocated, active, and concrete behaviors which compete for the execution. An enabled behavior is accomplished if its postcondition is satisfied. Enabled behaviors which are not accomplished can be dispatched for the execution by the executive system once the associated resources are allocated (actuators, input/output devices, control variables, etc.).

Since multiple behaviors can be active at the same time, they may conflict in accessing non-shareable resources. We rely on contention scheduling mechanism to regulate this competition. For this purpose, we exploit the behaviors' activation values. When a conflict arises, following a winner-takes-all approach, the behavior associated with a higher activation value is selected for the exclusive access to a contended resource. In WM, the activation value of a node is given by the weighted sum of all contributions for that node:

$$\mu_b = \sum_i w_{i,b} c_{i,b},$$

where contributions c_i, b can be either inherited from the connected nodes in the WM structure (top-down) or generated by the node itself from external or internal stimuli (bottom-up), while w_i, b are the contribution-specific weights. This way, given a shared resource or variable v and the set of competing behaviors B(v) for that variable, the behavior acquiring v is

$$b_{win} = \arg\max_{b \in B(v)} (\mu_b).$$

Overall, the executive system works as follows: when a new task is allocated in the WM for the execution, the associated schemata are recursively retrieved from the LTM and allocated into the WM till the concrete sensorimotor processes. Preconditions and postconditions associated to allocated tasks are continuously monitored by the executive system in order to establish the set of subtasks that are active and enabled in the current operative context. The behaviors belonging to the enabled subtasks are then associated with specific activation values, which are used to regulate their competition in case of conflicts. This induces a soft scheduling where the most emphasized behaviors (i.e. the ones better fitting the executive context) are prioritized. For instance, let assume a cooperative task where a robotic arm is tasked to test two points over a pipe, as shown in Figure 15. Assuming the two tasks enabled (i.e. both preconditions are satisfied and the tasks are not yet accomplished), in the absence of human interaction, the robot may be attracted by the nearest point to test (i.e., bottom-up stimulated by object proximity). However, during the movement towards the proximal target, the operator can physically *r* use the haptic device moving the arm toward the second point. In this case, the human intervention would elicit an additional activation influence inducing the robot to switch towards the intended target.

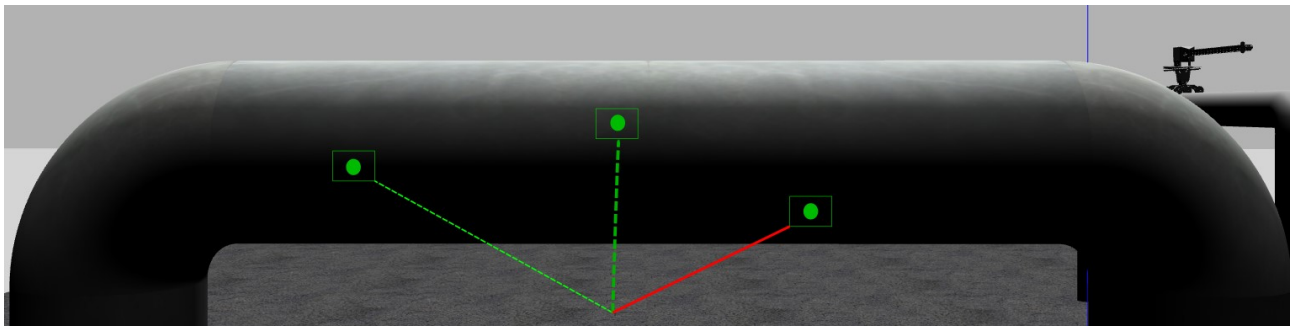


Figure 15: Multiple inspection points over a pipe.

We exploit object accessibility, task-based constraints, and human intention recognition to suitably single out, among the allocated tasks, the ones consistent with respect to the executive context and the user interventions. For this purpose, we distinguish the following types of influences to activations of the nodes in WM:

- Task-based influence: which is top-down provided to node *i* by the allocated tasks/subtasks in WM to be accomplished.
- Human-based influence, which is provided to node *i* by the physical interaction between the human and the robot; it emphasizes enabled nodes, which are also coherent with respect to the human guidance.
- Accessibility-based influence, which is provided by the environment, it emphasizes enabled nodes whose targets (e.g. objects, locations or trajectories) are more accessible (e.g., closer).

The task-based influence is the weighted sum of the contributions inherited from the other nodes in WM. Instead, the human and the accessibility influences are combined together into a unique contribution c_i due to external stimuli. This is obtained by the following convex combination:

$$c_{i,i} = m_h \cdot h_i + m_a \cdot a_i$$

This weighted sum is exploited to mediate between accessibility and human guidance. The accessibility-based influence drives the robot towards the closest location where an operation can be performed (target), as specified by the following equation:

$$a_i = \frac{d_{MAX} - d(i)}{d_{MAX}}$$

Where $d(i)$ is the length of the trajectory calculated to reach the target of the node *i*, and d_{MAX} is the maximum reachable distance in the robot workspace. The human-based influence should induce the autonomous system of the robot to move towards the target pointed by the operator guidance. In our framework, each possible target location is associated with a score $s_h(i)$, obtained from the assessment of the human physical guidance given the target associated to node i. Such score is an

output provided by the LSTM classifier and estimates how likely the user is driving the robot to that location. The human-based influence is then defined as follows:

$$h_i = \frac{max(0, s_h(i) - \lambda)}{1 - \lambda},$$

Depending on the balances of weights we can introduce the following execution setups:

- Human-guided: enabled when $m_h > m_a$; in this case, the robot is more prone to follow the human guidance rather than possible alternatives enabled by the plan and suggested by the environment (i.e. targets accessibility).
- Target-guided: associated with $m_h < m_a$; in this mode the robot tends to act according to the plan guidance and the environmental stimuli, rather than following the operator inputs.
- Balanced: when m_a is similar to m_h; the system is not biased towards accessible targets or human guidance, but the robotic behavior is equally sensitive to both of them.

The combined effect of the human and the accessibility influences can be exemplified considering the scenario depicted in Figure 15, which represents target points in a workspace to be reached by the robot end-effector with the human assistance. The collaborative task is decomposed into 5 behaviors, each associated with a target location (the coloured points depicted in the figure). We assume all behaviors always enabled (satisfied preconditions) with the same task-based influence since the goal is to reach all the target points without a specific ordering. During the execution, the operator can physically interact with the robot to drive its end effector toward the desired location (e.g. from waypoint WP1 to WP4).

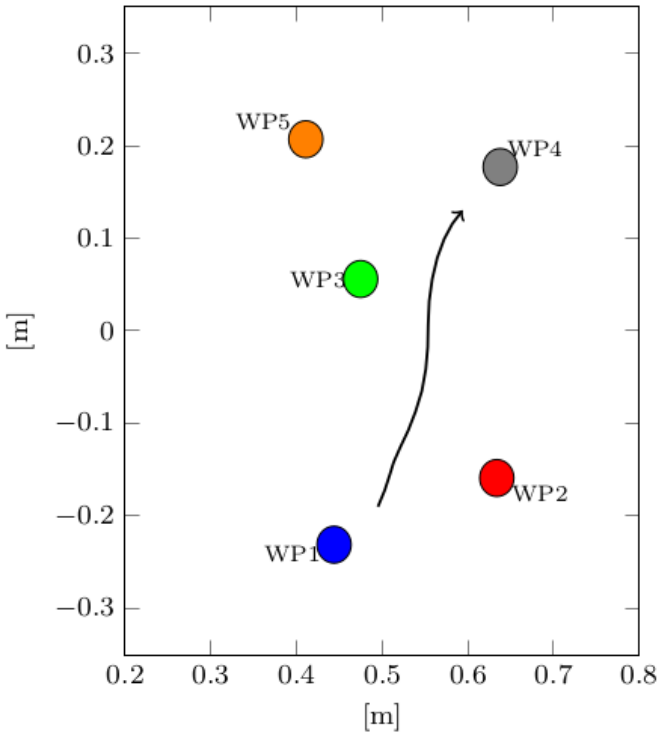


Figure 16: Waypoint map in the workspace.

The development of the activation values in this scenario is illustrated in Figure 17. The first two charts plot the temporal evolution of the distance and the human intention stimuli for each active concrete behavior associated with a target point (WP1, ..., WP5). The other three charts plot the evolution of the combined contributions assuming 85%-15%, 60%-40% and 25%-75% of balance between the accessibility and the human intention stimuli; these three cases are examples of the target-guided, balanced, and human-guided modes, respectively.

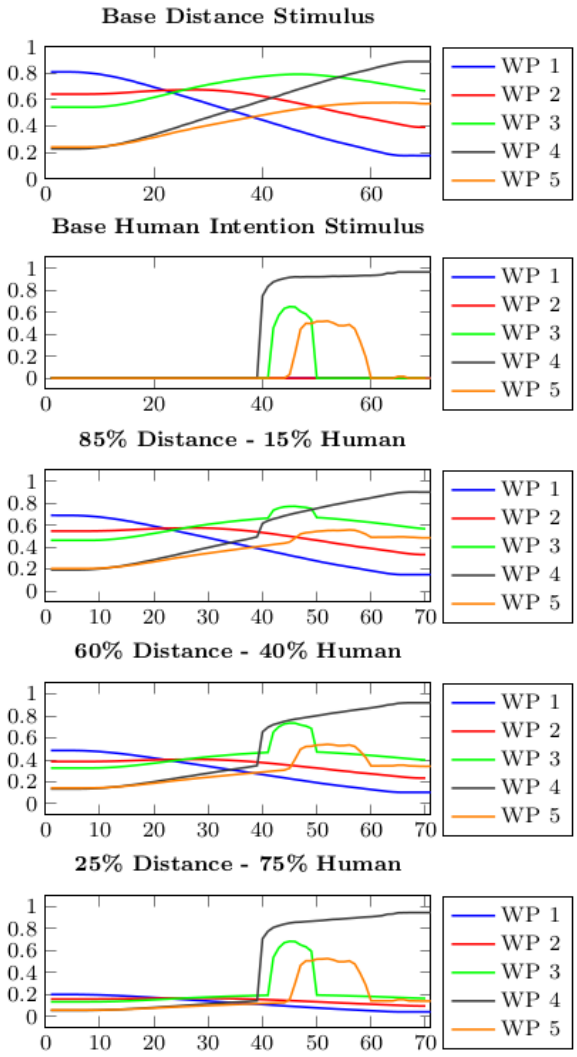


Figure 17: Behaviour regulation.

Coming back to Figure 16, it shows the robot end effector starting from a position close to WP 1 and WP 2 to progressively reach WP 4 while passing near WP 3. The associated activations are plotted in Figure 17. Initially, the human guidance is neglected and the robot behavior is mainly affected by the proximal waypoints, at about half a second (around 40 steps) the human guidance is also considered and WP 3, WP 4 and WP 5 are recognized as possible targets. Notice that WP 1 and WP 2 are opposed to human guidance therefore they do not receive stimuli, while WP 4 is the target which better fits the guidance, hence it receives the higher stimulus. In the e_m target-guided setting (third plot), alternative targets compete and the robot has to reach a certain distance from WP3 before the desired target W4 becomes the one winning. Instead, in the balanced a human-guided setting (last two plots), since the score of the human intention estimation is higher, the behavior associated with WP4 immediately wins the competition among the other targets.

To assess the human intentions, we exploit a Recurrent Neural Network to classify the human interventions through the haptic device. Specifically, the human interventions, are classified by the

network in the following categories depending on the concordance of the operator inputs with respect to targets and trajectories:

- Concorde (C): Human guidance follows the trajectory.
- Deviation Concorde DC: The operator wants to modify the trajectory without changing the active target.
- Opposite (O): The operator wants to go against robot motion.
- Deviation Opposite (DO): The operator wants to switch target.

The input layer takes an interaction snapshot made up by the human force magnitude F_t, the angle between human force direction and planned motion $d_p = \angle\, \vec{d}_d, \overrightarrow{d_p}$ and the distance between the position of the end-effector, and the closest point of the trajectory $d_h = \left|X_p - X\_c\right|$. The middle layer consists of 25 nodes considering the sigmoidal activation function. Finally, 4 nodes corresponding to the possible classes make up the output layer. The proposed network model tries to generalize human intention classification taking into account only one single step of the interaction, i.e. one vector $h = (d_p, d_h, |F_t|)$. The approach is reactive and provides satisfactory results, on the other hand the classification is instantaneous and does not exploit the history of past interactions to disambiguate the human intent. This approach has been improved in order to enhance the intention recognition process exploiting the flow of data collected during the human-robot interaction. Indeed, data about previous interactions may not only support the interpretation of the current intervention, but also reduce possible observational errors, caused either by the sensors or by the way the human touches the robot during collaborative task execution. In this direction, we propose to deploy Recurrent Neural Network (RNN) based on Long Short-Term Memory (LSTM) nodes, which are particularly suited for time series classification. LSTM networks have been introduced to address the vanishing gradient problem in RNNs exploiting gates that selectively retains relevant information, while forgetting irrelevant information. Specifically, each LSTM node is composed of a memory cell and 3 different networks called gates (i.e., input gate, forget gate, output gate) acting as regulators for the manipulation and the utilization of the memory. Our intention classification network consists of an input layer, a hidden layer made up of LSTM cells, and an output layer associated with a *softmax* function. Notice that a new classification network is allocated for each trajectory/target to be assessed, therefore, in order to limit computational effort and memory usage, the desiderata is to deploy simple and small network structures. For this purpose, we designed a method for sequence classification that enables online deployment of such networks.

Given an input sequence $h = (h_1, \dots, h_n)$, where each h_i represents the i-th human interaction snapshot, and given its corresponding classification sequence $s = (y_1, 1, y_1, 2, y_1, 3, y_1, 4), \dots, (y_n, 1, y_n, 2, y_n, 3, y_n, 4)$, where each 4-tupla represents the outputs related to the 4 classes introduced above, the class c assigned to h is the first one for which there exists a subsequence $(y_{t_0}, c, \ \dots, y_{t_0} + d, c)$ such that for all t in $t_0, t_0 + d$ we have that $y_t, c > \lambda$ holds. That is, the sequence h is assigned to the class c, such that the classification result c remains coherent for a fixed time windows d, with confidence always greater than a fixed threshold lambda in that window. In our experimental setting, d was empirically set to 40 steps (about half a second and about one third of the length of all the sequences in the dataset), while lambda was set to 0.5.

In the training phase different hyperparameters, such as hidden layer size and number of training epochs, have been tested in order to select the network with a satisfactory accuracy in the proposed application. The dataset has been generated by physically interacting with the robot during its motion from one point to another following a simple trajectory, while recording data at 100 Hz. Since sequences with different lengths can be collected, these were divided into subsequences of a fixed length to be used in batch for learning. Specifically, we used 120 timesteps (about 1 second); shortest sequences were discarded, while sequences longer than 120 timesteps were divided into subsequences of the fixed length. Notice that in the experimental setup, the fixed length was chosen considering, on the one hand the statistics of the dataset (average length and percentiles), on the other hand the latency given to the system to assess the human intention (about 1 second). The collected dataset was then randomly split into training set and test set, covering the 80% and 20% of the data and an amount of 443 and 111 sequences, respectively. The following table reports the different accuracies reached with different learning windows, different training epochs and different sizes of the hidden layer.

LSTM nodes	Steps	Epochs	Accuracy	Average F1
8	1	150	81.08%	0.83
8	5	300	84.68%	0.85
8	10	850	83.78%	0.84
8	20	850	83.78%	0.84
8	30	400	81.98%	0.83
8	40	850	83.78%	0.85
8	60	850	78.38%	0.79
8	120	800	73.87%	0.75
16	1	400	83.78%	0.85
16	5	700	86.49%	0.87
16	10	650	84.68%	0.85
16	20	200	83.78%	0.84
16	30	300	79.28%	0.82
16	40	950	81.98%	0.84
16	60	1300	81.98%	0.83
16	120	850	81.98%	0.82
24	1	650	83.78%	0.85
24	5	550	84.68%	0.86
24	10	200	82.88%	0.83
24	20	200	84.68%	0.85
24	30	250	82.88%	0.84
24	40	350	81.92%	0.83
24	60	800	85.59%	0.86
24	120	600	82.88%	0.82

For ease of comprehension, not all of the combinations are reported. The obtained results show that, on average, satisfactory accuracies are reached faster when shorter training steps are exploited. On the other hand, with longer training steps more epochs are needed, while we empirically observed that accuracies may also get worse. In particular, learning on shorter subsequences seems a better

choice for on-line classification since the network tends to classify with less information and to better manage its memory during the interactive execution.

4. Multimodal Teleoperation supported by force feedback

In order to perform the inspection task, an additional control strategy has been deployed. The goal of the proposed approach consists in the control of the hyper-redundant system with the use of a haptic device receiving information about the by the two sensors plugged on the end effector of the arm: the depth sensor and the force and torque sensor (FTS). In this context, a visual servoing approach is deployed to hold the orientation of the end effector with respect to the pipe surface, as shown in Figure 18, where different orientation for the depth sensor are defined considering its position with respect to the pipe to inspect.

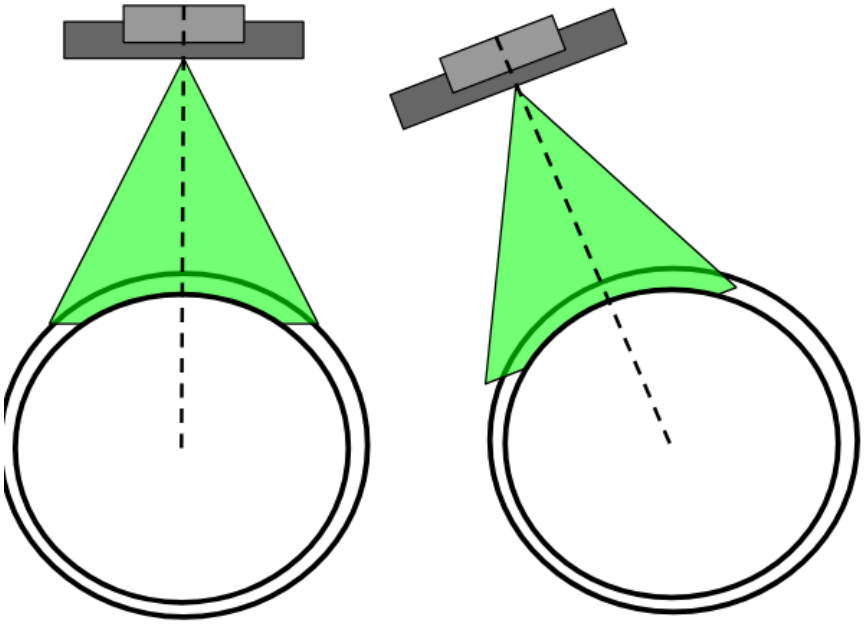


Figure 18: Different orientation for the depth sensor position.

In this setup, the orientation control is left to the autonomous system of the robot that exploit the vision sensor. Differently, the human operator is able to directly control the position of the end effector considering the data get by the vision sensor. In this way, the operator can rely on the visual feedback of the camera sensor placed on the end effector of the robot to perform the inspection task. An example of the vision feedback in the simulation scene used in this work is reported in Figure 19.

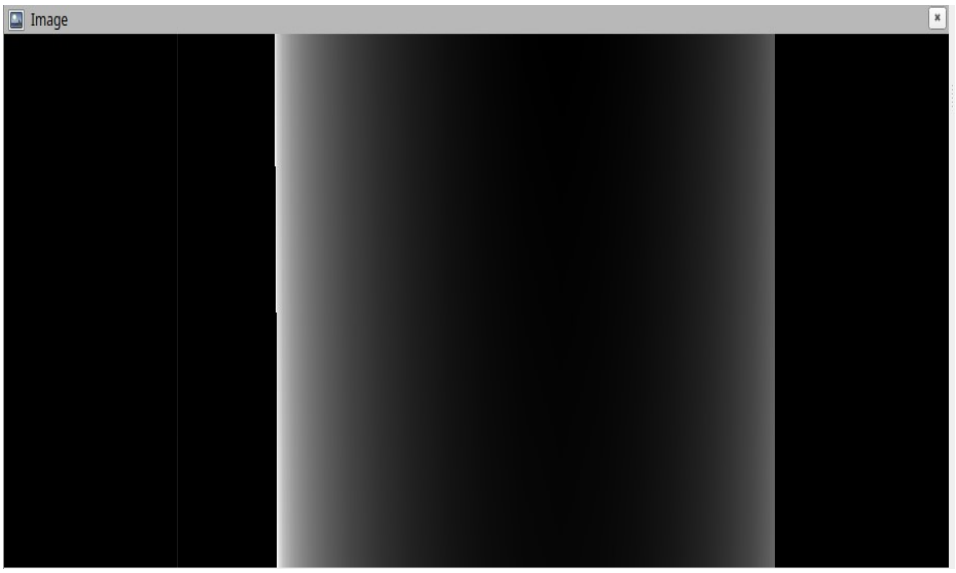


Figure 19: Video output from the simulated camera in the Gazebo simulator during an inspection task.

To assist the human operator in this task, a multimodal Human-Robot Interaction framework supported by force feedback has been implemented. In few words, the autonomous system of the robot merges different control source channels, like the vision data, the human input control and the FTS channel. In particular, to approach the inspection point, the autonomous system of the robot initially calculates the distance between the end effector and the pipe using the depth sensor. Then, when the approaching point appears to be too close to the depth sensor to receive valid range data, a hybrid admittance/force controller assures that the manipulator reaches the contact point. In contrast, during the manipulation task the robot could assume dangerous configurations bringing it to invalid positions or causing the slipping down of the platform from the pipe. In fact, even though a stabilization behaviour is enabled to maintain the robot upon the pipe, some configuration of the robot can affect the effectiveness of such behaviour. In particular, when the centre of mass of the robot is too far from the centre of the pipe or the wheels of the mobile base are not able to exert enough gripping on the pipe surface, the robot could fall down. For this reason, to prevent dangerous configurations a set of force feedback is generated to inform the human operator about the allowed direction in which is possible to move the arm.

In particular, a set of tasks has been formulated to manage possible working conditions, such as:

- *Position Error*: generates force feedback by informing the operator about the distance between the end effector and industrial pipe.
- *Manipulability Error*: informs the operator about the kinematic singularities of the arm.
- *Joint Limit Error*: helps the operator to avoid mechanical joint limit.
- *CoM Error*: Informs the operator about the distance between the centre of mass and the centre of the pipe.
- *Friction Cone Error*: Informs the operator about the position of the wheel inside the friction cone.

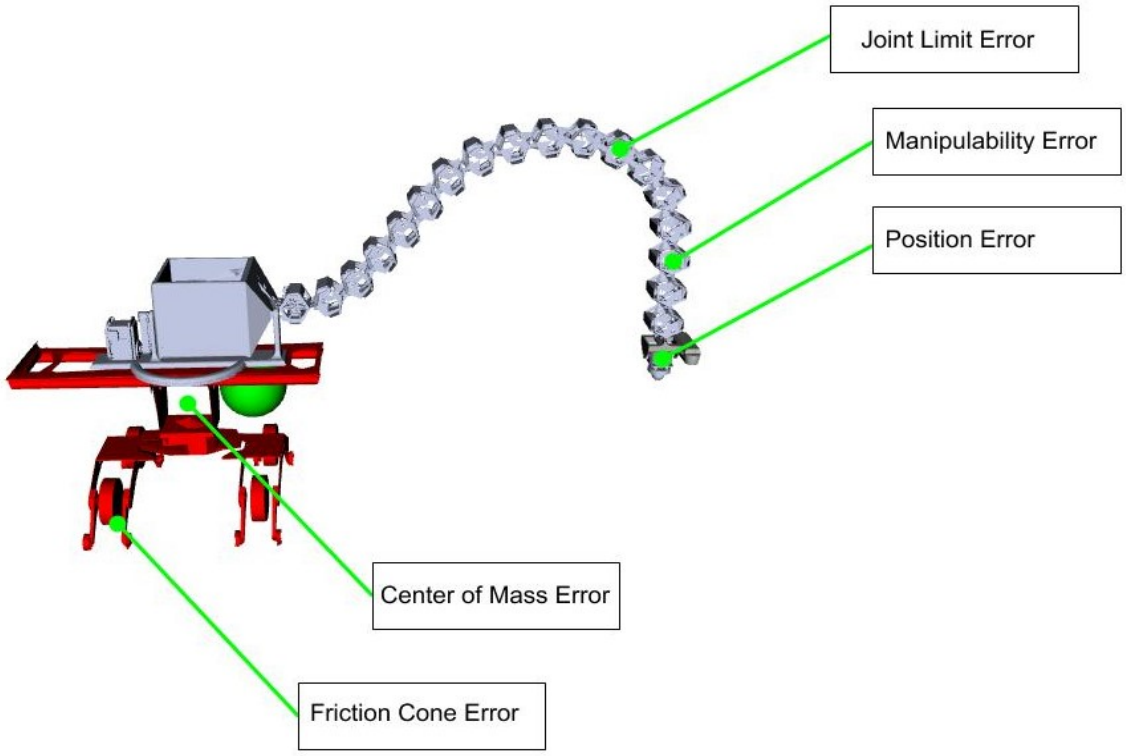


Figure 20: Different task errors considered in the manipulation task to generate the haptic force feedback.

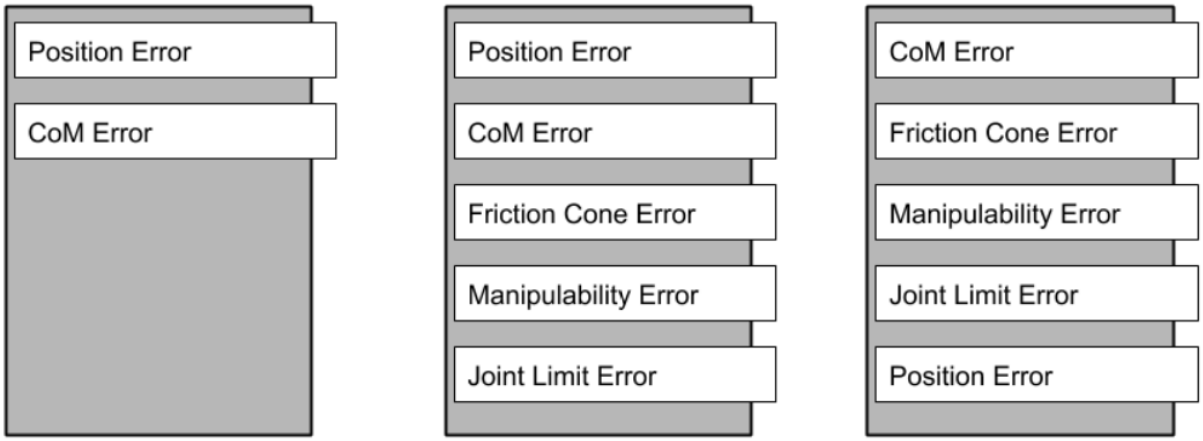


Figure 21: Priority modification for a stack of errors.

To generate a consistent force feedback, predictions about the value of each error considered in Figure 20 is performed. If during the teleoperation a certain error grows reaching its limit, a proper force feedback is generated to prevent the operator to move the arm towards that directions. This process is regulated by a prediction of the value of certain error moving if the operator want to move the end-effector of the manipulator towards a certain direction. A hierarchical priority stack is defined to provide different weight to the forces generated by a given task error. This mechanism works by changing the order in which a certain force vector is generated before to send them to the haptic

device. An example of dynamic priority allocation for such tasks is depicted in Figure 21. In this case, the position of a task inside the stack is univocally defined by the magnitude of the error. For example, the force generated by the joint limit or the manipulability tasks is not taken into account if its magnitude is lower than a certain threshold, allowing the operator to perceive only the force feedback related to the positioning task. At the same time, if the error of the secondary tasks, that are not related to the positioning of the manipulator are low, higher weight will be considered for the position task, attracting the end effector of the snake robot towards the inspecting pipe.