Stabilization and Control on a Pipe-Rack of a Wheeled Mobile Manipulator with a Snake-like Arm*

Simone D'Angelo[a,*], Antonio Corrado[a], Fabio Ruggiero[a], Jonathan Cacace[a], Vincenzo Lippiello[a]

[a] *CREATE Consortium and PRISMA Lab, Department of Engineering and Information Technology, University of Naples Federico II, Via Claudio 21, 80125, Naples, Italy*

Abstract

Autonomous inspection and maintenance tasks with robots in oil and gas refineries require moving along pipelines and manipulation dexterity in cluttered environments. This paper investigates the problem of controlling a wheeled mobile manipulator endowed with a snake-like arm to inspect the structures while stabilizing the supporting pipe. A model predictive control approach stabilizes the wheeled robot on the pipe. When the wheel torques saturate, the stabilization task leverages the resulting propagating force on the wheeled robot given by the snake-like arm's dynamics. The significant number of degrees of freedom given by the snake-like arm allows a prioritized redundancy resolution scheme with hybrid motion/force tracking to inspect the same and adjacent pipes while avoiding self-collisions and environmental impacts. Simulations in the realistic Gazebo environment validate the achieved preliminary results.

Keywords: Mobile robot, Hyper-redundant manipulator, NDT inspections

1. Introduction

Non-destructive test (NDT) measurements represent the standard way to assess the quality status of an industrial pipe. In this scenario, ultrasonic probes are used to retrieve the wall thickness of a surface to prove the integrity of the material without compromising its internal structure. These tests are performed by placing the inspection probe in fixed contact with the surface under examination. Besides, pipelines are usually grouped in steel-framed structures called pipe-racks that are typically set at high locations to enable easy access for inspection and maintenance and preserve the ground space of the industrial plants. Currently, NDT measurements are performed by humans that must climb a high scaffolding to reach the inspection location with the use of

*This work was supported by the AERO-TRAIN project, European Union's Horizon 2020 Research and Innovation Program under the Marie Skłodowska-Curie Grant Agreement 953454, and by the AERIAL-CORE project (Horizon 2020) under Grant 871479.

*Corresponding author

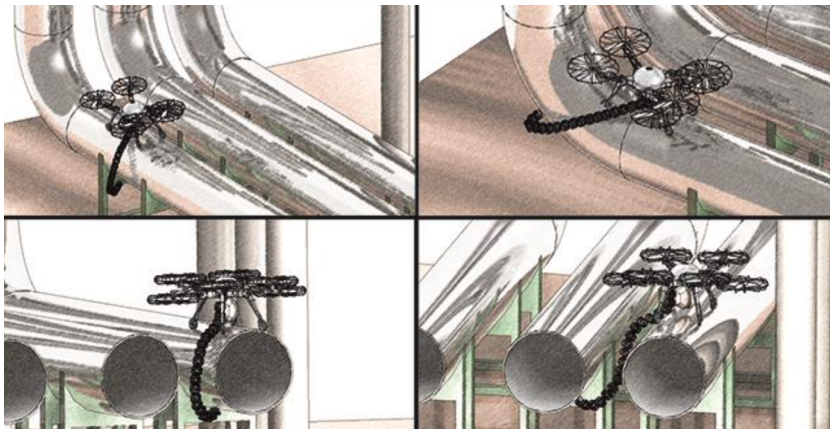

Figure 1: Concept of the device employed within the HYFLIERS project.

tools like man-lifts, cranes, or rope-access systems. Therefore, improving NDT inspection operations plays a fundamental role in raising human safety and decreasing the economic costs related to the inspection procedures (i.e., scaffolding installation, lengthy plant stoppage, and so on).

The solution proposed within the HYFLIERS project (`https://www.oulu.fi/hyfliers/`) is innovative in NDT inspection processes. The project proposes a hybrid aerial-ground robot equipped with a snake-like arm (see Fig. 1). The envisioned device can fly within the oil and gas facility, land over the pipe rack, use the rover's wheels to navigate along the pipe, and use the arm to inspect the pipe and the adjacent ones.

This paper investigates the problem of designing a control framework stabilizing the wheeled manipulator over a pipe while the snake-like arm approaches the pipe-rack structure for inspection purposes. The devised controller uses a model predictive control (MPC) approach to stabilize the rover over the pipe during the arm movement. The need to simultaneously manage and satisfy different dynamic constraints justifies the choice of an optimization technique to control the rover movement on the pipe: the constraints can be modelled by the dynamic of both the system and the interaction with the landing spot. The same result might also be achieved considering several basic control techniques, but an optimal control law allows for adjustment of the control actions based on both these selected constraints and the predictions carried out on the system evolution. Furthermore, the problem cost functions can be chosen to achieve multiple objectives.

Since the wheel torques may saturate, the stabilization task leverages the resulting propagating force on the wheeled robot given by the snake-like arm's dynamics. The significant number of degrees of freedom (DoFs) given by the snake-like arm allows the implementation of a prioritized redundancy resolution scheme with hybrid motion/force tracking to inspect the pipes while avoiding self-collisions and impacts on the environment.

The contribution of the work lies within the novel proposed application. To the best of the authors' knowledge, a wheeled manipulator equipped with a snake-like arm performing NDT measurements over a pipe does not exist so far. The preliminary development of such an idea may improve NDT inspection operations. Besides, the idea of exploiting the snake-like arm's dynamics to stabilize the wheeled base on the tube is a further contribution.

2. Related work

Recently, several robotic systems devoted to executing NDT measurements have been proposed. Some solutions consider using mobile platforms to perform the inspection task [1]. Internal climbing robots that can navigate the interior part of structures such as tanks, pipes, and steam chests have been proposed in [2, 3, 4]. These robots can be equipped with ultrasonic probes. However, this solution is unsuitable for the addressed case study since the internal parts of the industrial pipelines are not accessible. Another class of climbing robot can adhere on the external surface to inspect as in [5, 6, 7]. These works use pneumatic, magnetic, and mechanical actuation to navigate the external section of ducts. However, all these solutions are not helpful when inspecting pipeline locations that are hard to reach.

In this context, the use of aerial platforms has been investigated as well. The forefront technology to remotely perform NDT measurements using flying platforms relies on telescopic tools attached to the robot frame. The APPELLIX [8], Texo Drone Survey Inspection platform [9], Ronik Inspectioneering UT device [10], and Voliro inspection drone [11] represent the most promising business technology in this field. Besides, a research platform, AeroX, composed of an eight-tilted rotor platform with stability and manoeuvrability to make ultrasonic measurements with an active telescopic probe, is presented in [12]. The main problem of these solutions is the stability of the floating base, which can be compromised when the robot is flying in contact with the inspection point and the battery duration. Differently, the HYFLIERS project proposes an innovative solution: the inspection is carried out when the drone lands on a pipe, and the propellers are not considered during the execution of the whole task to preserve the battery duration. A similar hybrid platform equipped with a 4-DoF arm has been presented in [13]. However, adopting a snake-like arm is essential to enlarge the robot's workspace. This type of robotic arm is characterized by many DoFs, allowing the exploration of confined spaces or structured environments. Their use is widespread in literature [14, 15]: it is possible to use them through kinematic control schemes to complete submarine operation combining to an underwater vehicle [16], in amphibious applications [17], for nuclear reactors inspection [18], the inspection of pipes [19] with path planning and kinematic control only, or elastically deformable channel [20] adopting a model-free control scheme. Similarly, the possibility of adding a snake-like arm to an aerial vehicle has already been investigated in [21]. An interesting application of hyper-redundant arms is presented in [22] where an hybrid motion-force

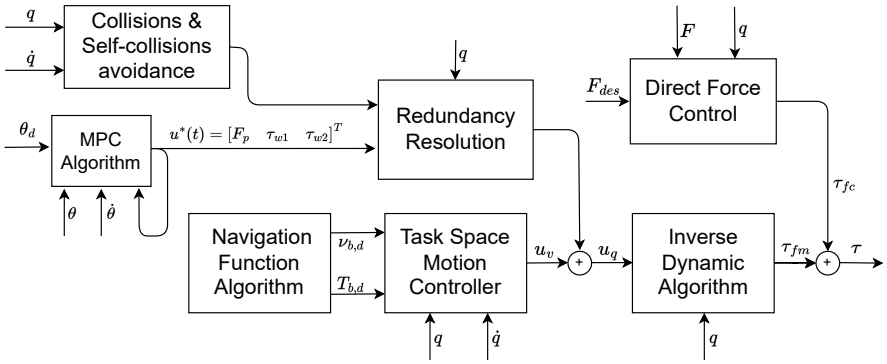

Figure 2: System architecture summarizing the developed controller: all the blocks and their interconnection are showed.

control strategy is proposed for a space manipulators considering the arm as a set of interconnected subsystems.

In this work, the rover platform presented in [23, 24] is considered. There, a robotic arm is addressed for pipe inspection. Here, instead, a snake-like robot is considered mounted on the rover, improving the dexterity of the platform and the range of possible applications in cluttered pipelines. To solve the rover stabilization problem, the solution proposed in [25] is taken as inspiration. The algorithm in [25] has been modified here to obtain a linear version of the optimization problem, while the propellers contribution in [25] has been replaced here by exploiting the snake-like arm's dynamics.

3. Overall architecture

The proposed mobile manipulator is made with a wheeled base (rover) and a snake-like arm. The rover mounts four custom mecanum wheels allowing the omnidirectional robot motion along the pipe. See [23, 24] for more information about the rover. This hyper-redundant manipulator comprises $n \gg 6$ revolute joints, giving a high movement capability in cluttered environments.

The following assumptions are made. A1) The robot is already landed on the pipe-rack and the propellers are turned off for all the task. A2) The rover can move only in its sagittal plane. A3) The wheels and the pipes are rigid and a Coulomb friction model is assumed at the contact. A4) The radius, size, and distances in the pipe-rack are known. A5) The inertia of the wheels is negligible compared to the inertia of the rover's rotation around the pipe's centre.

Assumption A1 is to save battery consumption [24] and has no particular implications towards the paper's aims. Assumption A2 is not limiting as practically demonstrated in [23, 24]. Assumption A3 is commonly used when contacts happen and friction is relevant [26]. Assumption A4 is doable since pipes have standard measures and oil and gas industries have precise knowledge of their plants. Assumption A5 is realistic given the values of the initial prototype in [23, 24].

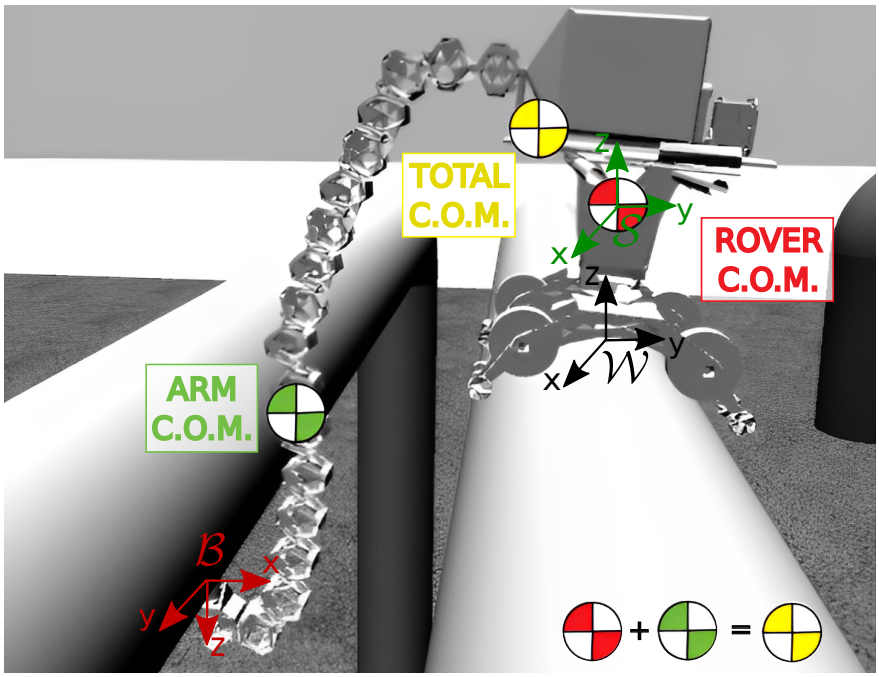

Figure 3: Snapshot from the simulation environment, with the pipe-rack, the rover, and the snake-like arm. The main frames are depicted. The arm's CoM is in green; the rover's CoM is in red; and the whole robot's CoM in yellow.

The devised architecture, as showed in Fig. 2, comprises (*i*) an offline trajectory planner for the end-effector (E-E) of the snake-like arm; (*ii*) an MPC controller to stabilize the rover on the pipe; and (*iii*) a hybrid motion/force control scheme with prioritized redundancy resolution to track a trajectory and a force reference while retaining the rover balance without colliding with the environment and avoiding self-collisions. It is worth highlighting that the proposed architecture has a decentralized structure, in the sense that the snake-like arm and the rover are seen as two separate entities, while the interactions between the two are seen as external effects to deal with [27].

4. Snake-like arm mathematical modeling

With reference to Fig. 3, let $\mathcal{W}$ be the fixed reference frame placed at the top of the pipe. Let $\mathcal{S}$ be the frame attached to the rover's center of mass (CoM). Without loss of generality, at the beginning of the task, $\mathcal{W}$ and $\mathcal{S}$ have the same orientation. Let $\mathcal{B}$ be the frame attached to the snake-like arm's E-E.

Given the assumption A2, it is possible to decouple the kinematics and dynamics of the snake-like arm from the one of the rover. Indeed, the rover has only one DoF given by $\theta \in \mathbb{R}$ which concisely describes the orientation $R_s(\theta) \in SO(3)$ of $\mathcal{S}$ with respect to $\mathcal{W}$. Therefore, the pose of $\mathcal{S}$ in $\mathcal{W}$ is

dictated by $T_s(\theta) = (p_s, R_s) \in SE(3)$, with $p_s \in \mathbb{R}^3$ the position of $\mathcal{S}$ with respect to $\mathcal{W}$. The proposed control law is also decentralized: in the following the manipulator mathematical modeling is retrieved using the screw theory approach [28] to handle with its DoF high value. Moreover in Section 6 the rover dynamic is presented and the optimization problem is introduced.

The pose of $\mathcal{B}$ with respect to $\mathcal{S}$ is given by $T_b^s(q) = (p_b^s, R_b^s) \in SE(3)$, with $p_b^s \in \mathbb{R}^3$ and $R_b^s \in SO(3)$ the position and the orientation of $\mathcal{B}$ with respect to $\mathcal{S}$, respectively, and $q = \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}^T \in \mathbb{R}^n$ the vector collecting the snake-like arm's joint values. The arm's direct kinematic problem can be solved using the product of exponentials formula [28]

$$T_b^s(q) = e^{[s_1]q_1} e^{[s_2]q_2} ... e^{[s_n]q_n} M, \tag{1}$$

where $s_i = \begin{bmatrix} \omega_i^T & -(\hat{\omega}_i b_i)^T \end{bmatrix}^T \in \mathbb{R}^6$, with $i = 1, \ldots, n$, is the screw axis associated to the revolute joint q_i; $[s_i] = \begin{bmatrix} \hat{\omega}_i & -\hat{\omega}_i b_i \\ 0_{1\times 3} & 0 \end{bmatrix} \in se(3)$ is the matrix representation of the screw axis; $\hat{\omega}_i \in so(3)$ is the skew-matrix associated to the vector $\omega_i \in \mathbb{R}^3$ that represents the unit vector in the positive direction of the i-th joint axis expressed in $\mathcal{S}$; $b_i \in \mathbb{R}^3$ is any arbitrary point on the i-th joint axis expressed in $\mathcal{S}$; $0_\times$ is zero vector of proper dimensions; and $M \in SE(3)$ expresses the pose of $\mathcal{B}$ in $\mathcal{S}$ when the arm is at its home position. The single exponential map $e^{[s_i]q_i} \in SE(3)$ is defined in [28]. The E-E pose in $\mathcal{W}$ can be easily obtained as $T_b(q, \theta) = T_s(\theta) T_b^s(q) \in SE(3)$.

The differential kinematic for the snake-like arm is expressed by the relation $v_b^s = J^s(q)\dot{q}$, with $v_b^s \in \mathbb{R}^6$ the E-E twist related to $\mathcal{S}$ and $J^s(q) \in \mathbb{R}^{6\times n}$ the spatial Jacobian expressed in $\mathcal{S}$. The Jacobian is obtained iteratively, and its i-th column can be obtained through

$$J_i^s(q) = \mathrm{Ad}_{e^{[s_1]q_1} e^{[s_2]q_2} ... e^{[s_{i-1}]q_{i-1}}} s_i \in \mathbb{R}^6, \tag{2}$$

for $i = 2, \ldots, n$ and $J_{s,1}(q) = s_1$. The adjoint matrix, $\mathrm{Ad}_T \in \mathbb{R}^{6\times 6}$ [28] is useful to change representation frame, given a generic transformation matrix $T \in SE(3)$. It is thus possible to express the spatial Jacobian $J^s(q)$ in $\mathcal{W}$ as $J(q) = \mathrm{Ad}_{T_s(\theta)} J^s(q)$.

The dynamic model of the manipulator can be obtained through the recursive Newton-Euler approach exploiting again the screw theory approach [28]. The compact form of the dynamic model is expressed by

$$B(q)\ddot{q} + h(q, \dot{q}), = \tau - J_b^T(q) F, \tag{3}$$

where $B(q) \in \mathbb{R}^{n\times n}$ is the inertia matrix; $h(q, \dot{q}) \in \mathbb{R}^n$ accounts for the gravitation, centrifugal, Coriols terms, and friction terms; $J_b = Ad_{T_{eff}^{-1}}[J(q)]$ is the body Jacobian; $F \in \mathbb{R}^6$ is the external force at the manipulator end-effector's tip; $\tau \in \mathbb{R}^n$ is the joint torques vector.

For further details and insights on the presented mathematical modeling, please visit the GitHub page of the project (`https://github.com/prisma-lab/MPC_HyperRedundant`).

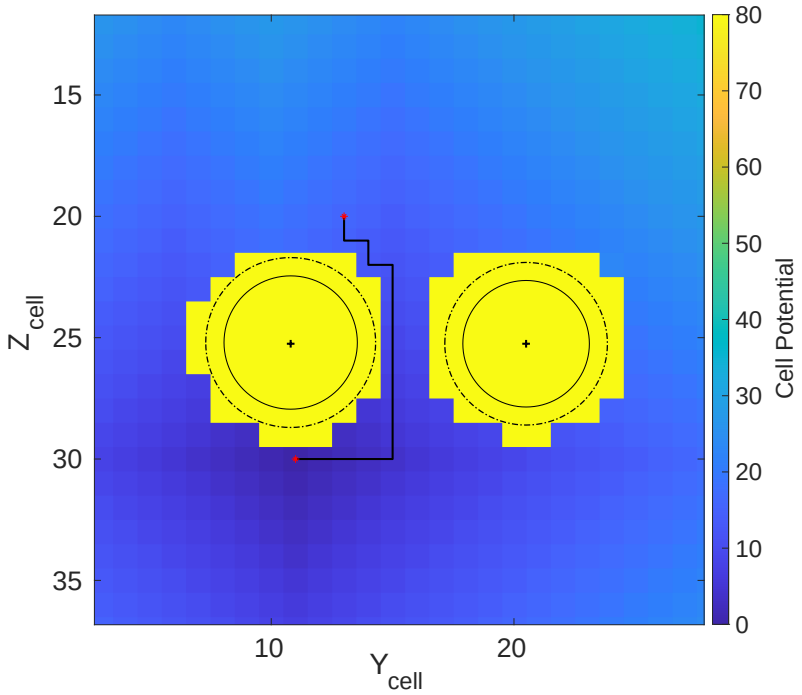

Figure 4: Potential grid-map built with the navigation function algorithm. The change from blue to yellow shows different value of potential in the $Y_{cell}Z_{cell}$-plane. The dotted lines indicate the collision-free region bounds. The cell-like path is shown in black connecting the two red dots representing the initial (top) and final configurations (bottom). The trajectory is computed fixing the E-E's x-position.

5. Offline trajectory planning

Given the assumption A4, the scope of the planner is to offline compute the desired E-E trajectory in $\mathcal{W}$, avoiding the environmental obstacles. Let $T_{b,i} = (p_{b,i}, R_{b,i}) \in SE(3)$ be the initial pose of the robot at time $t = t_i \geq 0$, with $p_{b,i} \in \mathbb{R}^3$ and $R_{b,i} \in SO(3)$ the initial position and orientation of $\mathcal{B}$ in $\mathcal{W}$, respectively. Let $T_{b,f} = (p_{b,f}, R_{b,f}) \in SE(3)$ be the desired pose of the robot at $t = t_f > t_i$, with $p_{b,f} \in \mathbb{R}^3$ and $R_{b,f} \in SO(3)$ the desired position and orientation of $\mathcal{B}$ in $\mathcal{W}$, respectively. The desired E-E trajectory in each instant of time $t \in [t_i, t_f]$ is given by

$$T_{b,d}(t) = \begin{bmatrix} R_{b,d}(t) & p_{b,d}(t) \\ 0_{1x3} & 1 \end{bmatrix}, \quad \dot{T}_{b,d} = \begin{bmatrix} \dot{R}_{b,d}(t) & \dot{p}_{b,d}(t) \\ 0_{1x3} & 0 \end{bmatrix}, \tag{4}$$

Notice that it is possible to compute the desired E-E twist as $\upsilon_{b,d} = \dot{T}_{b,d} T_{b,d}^{-1}$ [28].

The planned position $p_{b,d}(t) \in \mathbb{R}^3$ is obtained using a path primitive of the

desired trajectory [29]: in case of a rectilinear path, the representation

$$p_{b,d}(t) = p_{b,i} + \frac{s(t)}{\left\|p_{b,f} - p_{b,i}\right\|}(p_{b,f} - p_{b,i}) \tag{5}$$

is used, where $\|\cdot\|$ is the Cartesian norm and the evolution of the arc length $s(t) \in \mathbb{R}$ is imposed using a polynomial approach with the expression whose coefficients must be found imposing the initial and final conditions the trajectory. In this paper, a 5-th order polynomial has been employed.

The planned orientation $R_{b,d}(t) \in SO(3)$ is carried out in following way. First, the matrix $R_{b,f}^{b,i} = R_{b,i}^T R_{f,i} \in SO(3)$ is computed. Then, the axis/angle representation is applied to $R_{b,f}^{b,i}$ extracting the axis $r \in \mathbb{R}^3$ and the angle $\alpha_f \in \mathcal{S}^1$ [29]. The matrix $R_{b,d}(t)$ is in turn obtained as $R_{b,d}(t) = R_{b,i}R^i(t)$, where $R^i(t) \in SO(3)$ is constructed through an axis/angle representation at each instant of time, with axis r and angle $\alpha(t)$, such that $\alpha(t_i) = 0$ and $\alpha(t_f) = \alpha_f$. Again, a polynomial representation can be used for $\alpha(t)$. Through this choice it is possible to obtain $R_{b,d}(t_i) = I_3$, with $I_\times$ the identity matrix of proper dimensions, and $R_{b,d}(t_f) = R_{b,f}$.

The entire trajectory is obtained as a concatenation of rectilinear paths computed using the previous approach. For simplicity, the inspection task is planned on the hybrid platform sagittal plane, defined as the plane dividing in half the manipulator in its longitudinal directions. The motion in this plane is defined only along two axes of its reference frame: Y and Z. The end-effector trajectory planned to reach the inspection point under the pipe considers constant the movement along the third component of the reference frame. The sagittal plane can be appreciated in Fig. 4 showing a 2-D discretization of the workspace presented in Fig. 3. Two adjacent pipes of the pipe-rack are considered (see Fig. 4). Each pipe is approximated to a circle whose radius is $R_p > 0$. In order to have a collision-free trajectory, the navigation function technique is used [29].

In Algorithm 1, a pseudo-code for the offline planning technique is presented: the algorithm can be adapted to different environments specifying the discretization resolution and assuming to know all the obstacles and their shapes. One of the improvements of this method with respect to the classical artificial potential algorithm is the avoidance of the so-called local minima.

This method is based on dividing the workspace (the plane ZY) into a grid (the plane $Z_{cell}Y_{cell}$), eliminating those cells associated with the presence of an obstacle. A potential/score value is then associated with each grid cell, and the zero value is associated with the cell containing the goal. Starting from it, a map of potentials is constructed, considering an adjacency law to explore the grid. To come by a collision-free trajectory, the cells related to the obstacles are excluded from the planning, receiving a high score. The final trajectory is then obtained connecting the cells starting from the initial to the final one along a path associated with decreasing potentials. An additional step is considered to minimize the number of segments of this path at the end of the algorithm: the extremities of all the segments are analyzed, removing unnecessary sections, reducing the length of the whole trajectory without neglecting the collision-

Algorithm 1 Navigation Function for E-E trajectory planning

Select $p_0, p_f \in \mathbb{R}^2$ E-E initial and final configurations in the sagittal plane
Build the map modelling the known obstacles
Select a resolution for the discretization
while the map is not entirely explored **do**
    select next cell and check for collision with obstacles
    if no collision are detected **then**
        add the cell to *C-free* region
        if the goal is inside the current cell **then**
            assign to it 0 potential and save its index
        end if
    else
        add the cell to *C-obstacle* region
    end if
end while
Choose an adjacency law to explore the grid-map
while each cell belonging to *C-free* has not a potential value **do**
    select next cell and explore its adjacent cells
    if potential is not already assigned **then**
        give to the visited cell potential equal to the previous one +1
    else
        do not visit it
    end if
end while
Find the cell containing p_0
while cell containing p_f not reached **do**
    add actual cell to the path
    explore the adjacent cell with potential equal to the actual one -1
end while

free hypothesis. Besides, the last segment must be considered to connect the obtained path with the desired position under the pipe to inspect it with the correct orientation. An illustrative example of the navigation function result is given in Fig. 4.

6. MPC controller for stabilization

Along the lines provided in [25], an MPC controller is devised to stabilize the rover on the pipe. Differently from [25], the robot has a snake-like manipulator whose dynamics help in stabilizing the rover. To this purpose, the CoM of the whole robot (see Fig. 3) is employed. Indeed, this quantity is a function of the rover's and the arm's CoM.

With reference to Fig. 3, the rover is approximated as in the schematic representation of Fig. 5 thanks to the symmetry with respect to the sagittal

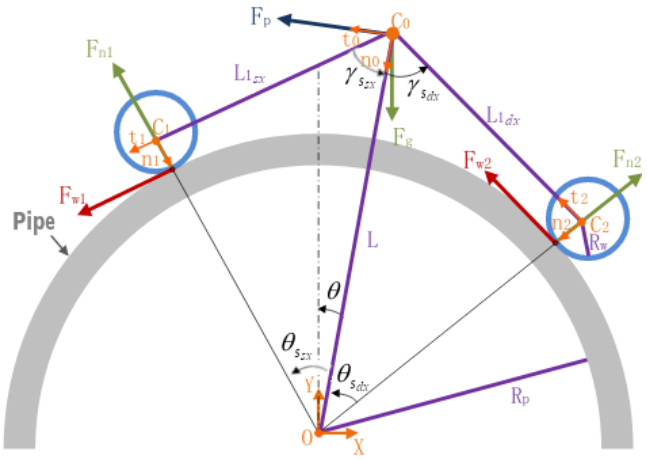

Figure 5: 2-D sketch of the rover on a pipe, with the illustration of the symbols employed to derive the dynamic model of the rover and its constraints.

plane and the A2 assumption. Therefore, the rover is assumed to be controlled by the equivalent of two wheels (each resulting wheel is the pair of two actual wheels) through the torques $\tau_{w1} \in \mathbb{R}$ and $\tau_{w2} \in \mathbb{R}$, respectively. The effect of the snake-arm dynamics is seen as a resulting force $F_p \in \mathbb{R}^2$ in the sagittal plane. The input vector to stabilize the rover on the pipe is thus given by $u = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T = \begin{bmatrix} \|F_p\| & \tau_{w1} & \tau_{w2} \end{bmatrix}^T$. The MPC will compute the optimal u to avoid the rover slipping and falling from the pipe, taking into account many constraints. The wheel torques τ_{w1} and τ_{w2} are directly applied to the rover, while the motion controller explained in the next section takes care of controlling the snake-like arm to track the desired E-E's motion/force and, in parallel, give the resulting F_p at the whole robot's CoM.

With reference to Fig. 5, let C_0 be the whole robot's CoM in the sagittal plane, n_i and t_i are the versors defining the frames attached to the rover's CoM and wheels. Let $L > 0$ be the distance between the center of the pipe and the rover's CoM, γ_s, $\theta_{s,sx}, \theta_{s,dx} \in \mathcal{S}^1$ be the angles depending on the geometry of the device and the pipe, $L_{1,dx}, L_{1,sx} > 0$ the distances from C_0 to the first and second wheel, respectively. Since C_0 varies due to the arm's movement, the parameters $L_{1,dx}$, $L_{1,sx}$, $\theta_{s,dx}$, $\theta_{s,sx}$, $\gamma_{s,dx}$ and $\gamma_{s,dx}$ that appear in Fig. 5 must be computed in each cycle through straightforward geometric relationships, here omitted for brevity. They basically depend on the manipulator's CoM that is computed in each control step using the theory of rigid bodies through $CoM_{manip} = \dfrac{\sum_{i=1}^{n} m_i r_i}{\sum_{i=1}^{n} m_i} \in \mathbb{R}^3$, where $m_i > 0$ is the mass of each link and $r_i \in \mathbb{R}^3$ is the position of the CoM of each link in $\mathcal{S}$.

Let $\theta \in \mathcal{S}^1$ be the angle of C_0 with respect to the vertical axis of $\mathcal{W}$. Define the state vector $x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T = \begin{bmatrix} \theta & \dot{\theta} \end{bmatrix}^T \in \mathbb{R}^2$. Notice that θ can be obtained from the knowledge of the position of C_0 in the sagittal plane through the $Atan2$

function. The dynamics describing the rotation of the structure around the tube can be written in a discrete-time form with time $k \in \mathbb{Z}$ as

$$f(x(k), u(k)) =$$

$$\begin{bmatrix} x_1(k) + T_s x_2(k) \\ x_2(k) + T_s \left[\dfrac{g \sin(x_1(k))}{L} + \dfrac{u_1(k)}{mL} + \dfrac{\cos(\theta_{s,dx})}{mLR_w} u_2(k) + \dfrac{\cos(\theta_{s,sx})}{mLR_w} u_3(k) \right] \end{bmatrix}, \tag{6}$$

where $m > 0$ is the total mass of the mobile manipulator, $g > 0$ is the gravity acceleration, and $T_s > 0$ is the sampling time period.

Define the vectors $F_p = \begin{bmatrix} -\|F_p\| \cos\theta \\ -\|F_p\| \sin\theta \end{bmatrix}$, $F_{w1} = \begin{bmatrix} -\|F_{w1}\| \cos(\theta_{s,sx} + \theta) \\ -\|F_{w1}\| \sin(\theta_{s,sx} + \theta) \end{bmatrix}$, and $F_{w2} = \begin{bmatrix} -\|F_{w2}\| \cos(\theta_{s,dx} - \theta) \\ \|F_{w2}\| \sin(\theta_{s,dx} - \theta) \end{bmatrix}$ where $\|F_{wj}\| = \tau_{wj}/R_w$ with $j = \{1, 2\}$ and $R_w > 0$ the radius of each wheel. They represent the rover's control inputs describing, respectively, the force acting at the rover CoM given by the arm's dynamics to help in the stabilization process and the two force vectors at the wheels remapped then in torques commanding their movement around the pipe. F_p is the term expressing a force acting on the rover CoM to help in the stabilization process. It is expressed as a reaction force given by the internal motion of the hyper-redundant arm. This term is then referred to the fixed frame $\mathcal{W}$ to be used in the next section in the hybrid hierarchical controller using the relative adjoint transformation matrix $Ad_{[T_r^w]}(\cdot)$. T_r^w is the transformation between the rover CoM frame and the $\mathcal{W}$ frame.

The MPC goal is to nullify the rover's angular position error with respect to the pipe. In order to ensure this, it is helpful to consider the concept from the walking robots domain, that the resultant of all the forces acting on the robot must remain strictly inside the support polygon to ensure stability against gravity [30]. The lumped term acting at C_0 must lie within the cone of angle $2\gamma_s$; it is given by $F_s = F_g + F_p + F_{w1} + F_{w2}$, with $F_g = \begin{bmatrix} 0 & -mg \end{bmatrix}^T$. These dynamic conditions can be defined taking inspiration from the Coulumb's friction law [31]:

$$\begin{cases} F_s^T n_0 > 0 \\ |F_s^T t_0| \leq \tan(\gamma_s) F_s^T n_0 \end{cases}, \tag{7}$$

where $n_0 = \begin{bmatrix} \sin(\theta) \\ -\cos(\theta) \end{bmatrix}^T$, $t_0 = \begin{bmatrix} -\cos(\theta) \\ -\sin(\theta) \end{bmatrix}^T$ and $\gamma_s = \dfrac{\gamma_{s,dx} + \gamma_{s,sx}}{2}$.

Following [25], the constraints to be imposed to the MPC to prevent wheels slippage and rover detaching and falling from the pipe are

$$\begin{cases} |\tau_{w1}| \leq \dfrac{\mu m g R_w L_{1,sx} \cos(\theta)}{L_{1,sx} \cos(\theta_{s,dx}) + L_{1,dx} \cos(\theta_{s,sx})} \\ |\tau_{w2}| \leq \dfrac{\mu m g R_w L_{1,dx} \cos(\theta)}{L_{1,sx} \cos(\theta_{s,dx}) + L_{1,dx} \cos(\theta_{s,sx})} \end{cases}, \tag{8}$$

where $\mu > 0$ is the static friction coefficient between each wheel and the pipe. Furthermore, the state variables and the control inputs must lie within the

following bounds representing the saturation limits of the actuators and the physical limits of the joints

$$
\begin{aligned}
x \in \mathbb{X} &:= \left\{ x \in \mathbb{R}^2 : -20^\circ \leq x_1 \leq 20^\circ, x_2 \in \mathbb{R} \right\}, \\
u \in \mathbb{U} &:= \left\{ u \in \mathbb{R}^3 : -10 \text{ N} \leq u_1 \leq 10 \text{ N}, -1 \text{ Nm} \leq u_2, u_3 \leq 1 \text{ Nm} \right\}.
\end{aligned}
\tag{9}
$$

The same cost function expressed as combination of Lagrange $\mathcal{L}(x_i, u_i)$ and Mayer $\mathcal{M}(x_K, u_K)$ objective term and optimization problem as in [25] is imposed, where stability matters are sketched out as well. However, to decrease the computational load and at the same time speed-up the simulations, in this paper the constraints (7, 8) and the dynamics (6) are linearized in each control step around the operative point $(\bar{x}, \bar{u})$ given by the actual state $\bar{x}$ and the previous optimal input $\bar{u}$. The result is a linear MPC, in which the prediction is carried out on the following linearized system

$$
\begin{aligned}
\min_{\{u_i\}_0^{K-1}} \quad & \sum_{i=0}^{K-1} \mathcal{L}(x_i, u_i) + \mathcal{M}(x_K, u_K) \\
\text{s.t.} \quad & \\
& x_{i+1} + \Delta x_{i+1} = f(\bar{x}, \bar{u}) + \frac{\delta f}{\delta x}(\bar{x}, \bar{u}) \Delta x_i + \frac{\delta f}{\delta u}(\bar{x}, \bar{u}) \Delta u_i, \\
& h(\bar{x}, \bar{u}) \leq \bar{h}, \\
& x_i \in \mathbb{X} \quad x_0 = \bar{x}, \\
& u_i \in \mathbb{U},
\end{aligned}
\tag{10}
$$

where $h(\bar{x}, \bar{u})$ is the matrix in compact form of the constraints defined in (7) and (8), $f(\bar{x}, \bar{u})$ is the rover dynamics and $\frac{\delta f}{\delta x}(\bar{x}, \bar{u})$ $\frac{\delta f}{\delta u}(\bar{x}, \bar{u})$ its derivatives with respect to the state and input variables. In each control loop, the optimal control input sequence $\{u_i\}_0^{K-1}$ is computed and only the first element is considered to actuate the robot and to linearize the problem in the next step: $\bar{u} = u_0$. Finally, K is the prediction horizon of the MPC problem.

7. Hybrid control and redundancy resolution

7.1. Hybrid motion/force control algorithm

A hybrid motion/force controller is designed to perform NDT measures for interacting with the environment. The aim is to define the Cartesian-space axes along which controlling either the E-E motion or the interaction force. The controller shown in Fig. 6 is composed of a Cartesian-space inverse dynamic controller with redundancy resolution and a direct force controller. The redundancy of the snake-like arm is later exploited to impose internal motions used to solve tasks with lower priority. Like in [32], the factor $\lambda \in \begin{bmatrix} 0 & 1 \end{bmatrix}$ is introduced to represent the transition between contact-less ($\lambda = 0$) and contact-based motion ($\lambda = 1$). A cosine step function interpolated between 0 and 1 is used to smooth the transition between these two modalities [33]. When this transition

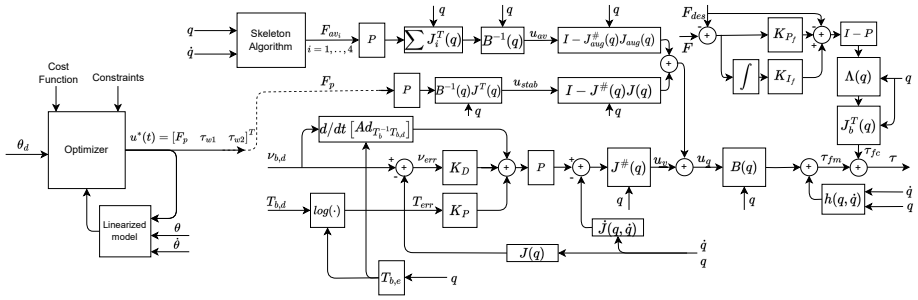

Figure 6: Overall control scheme. The offline planner gives the inputs to an inverse dynamic control scheme in the Cartesian space. At each sample time, the control torques for the manipulator are computed to track the desired trajectory. A prioritized redundancy resolution is devised to impose several subtasks aiming at avoiding the wheels slippage, preventing the rover to fall down (thanks to the MPC control problem results), avoiding collisions with the environment and self-collisions. When a contact with a pipe occurs the matrix P selects the Cartesian-space axis along which perform a motion control or a force control.

is needed, direct force control can track the desired force reference along some directions while the motion controller tracks the E-E position along the other axes. This selection is performed by the matrix

$$P = \begin{bmatrix} I_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1-\lambda \end{bmatrix} \end{bmatrix} \in \mathbb{R}^{6x6}. \tag{11}$$

This matrix is used in the following proportional-integral (PI) control scheme to track the force reference $f_{des}(t) \in \mathbb{R}^3$ nullifying the force error $e_F(t) = \begin{bmatrix} 0_3^T \\ f(t)^T - f_{des}(t)^T \end{bmatrix} \in \mathbb{R}^6$:

$$\tau_{fc} = J_b^T(q)\Lambda(q)(I_6 - P)\Big(- \begin{bmatrix} 0_3 \\ f_{des}(t) \end{bmatrix} + K_{p_f} e_F(t) + K_{i_f} \int e_F(\tilde{t})d\tilde{t}\Big), \tag{12}$$

where $\Lambda(q) = J_b^{\#^T}(q)B(q)J_b^{\#}(q)$ is the inertia matrix defined in the E-E frame coordinates, and $K_{P_f}, K_{I_f} \in \mathbb{R}^{6\times 6}$ are positive definite gains matrices.

The control scheme is completed with the design of a hierarchical motion controller. Taking into account (3), the joint torques are computed choosing a properly virtual control input $\ddot{q} = u_v \in \mathbb{R}^n$. Being $T_{b,d}, \upsilon_{b,d}$ the desired trajectory computed in the above section, it is then possible to impose

$$u_v = J^{\#}\Big(P\Big(\frac{d}{dt}([\mathrm{Ad}_{T_b^{-1}T_{b,d}}]\upsilon_{b,d}) + K_D\upsilon_{err} + K_P T_{err}\Big) - \dot{J}(q,\dot{q})\dot{q}\Big), \tag{13}$$

with $K_P, K_D \in \mathbb{R}^{6\times 6}$ positive definite matrices; $J^{\#} = B^{-1}J^T(JB^{-1}J^T)^{-1}$ the weighted Jacobian pseudo-inverse; $\upsilon_{err} = \upsilon_{b,d} - \upsilon_b$ and $[T_{err}] = \log(T_b^{-1}T_{b,d})$, with $\log(\cdot)$ the matrix logarithm. The selection matrix P partitions the 6D space

into torques that address the motion control task and torques that address the force control task. In the end, the hybrid motion/force controller is

$$\tau = \tau_{fm} + \tau_{fc}, \tag{14}$$

with $\tau_{fm} = B(q)u_v + h(q, \dot{q})$. The two controllers are decoupled by the orthogonal projections P and $I_6 - P$. Obviously, when $\lambda = 0$, controller (14) is a pure Cartesian-space motion controller.

7.2. Null-space projections

Redundancy is treated dynamically and it can be exploited projecting the secondary tasks in the null-space of the main task [34] using

$$\ddot{q} = u_q = u_v + (I_n - J^{\#}J)u^{''}, \tag{15}$$

in which $u^{''} \in \mathbb{R}^n$ is the control input relative to the task with lower priority, and $(I_n - J^{\#}J)$ is the main task null-space projector. When different tasks must be projected in the main task null-space with multiple levels of priority, an approach based on the augmented Jacobian is needed as outlined in [35], where stability matters are addressed.

The addressed tasks in this paper are (*i*) the stabilizing action from the MPC; (*ii*) the collision avoidance with environment obstacles; and (*iii*) the self-collision avoidance. In particular, the stabilizing action belongs to the second priority level, whereas the two collision avoidance actions have the same priority and are placed at the third level. The stabilizing action described by the term $\|F_p\|$, computed using the MPC control algorithm reported above, is relevant when the actuation of the stabilizing wheels fails to counter react the effect of gravity due to saturation. The term $\|F_p\|$ can be considered the reaction force given by the internal motion of the hyper-redundant arm. Starting from this term, it is possible to compute the joint torques using the statics relationship $u_{stab} = B^{-1}(q)J^T(q)PF_p$.

In the third priority level, the collision avoidance terms are summed together and then projected into the null-space of the second task using the relative augmented Jacobian. The avoidance torques $\tau_{av_i} \in \mathbb{R}^n$ are function of the the repulsive forces

$$f_{av_i} = h(d_0, d_{min}, d_{start})\frac{(p_{a,c} - p_{b,c})}{\|p_{a,c} - p_{b,c}\|}, \tag{16}$$

obtained using the *skeleton algorithm* as in [36]. These forces are computed only if one of the links of the manipulator is at a distance from the obstacle, or from another link, lower than a threshold representing a security bound. This threshold is given by the sum of the radius of the circle surrounding the obstacle $d_0 > 0$ and a safety distance $d_{min} > 0$. Besides, in (16) the term $h(d_0, d_{min}, d_{start}) > 0$ is the amplitude of the repulsive forces, which can have different shapes like the exponential or sigmoidal trends. In this paper, the first shape has been used to model the collision avoidance force against the static objects in the environment, whereas the second one is used for the generation

of the self-collision avoidance forces. The terms $p_{a,c} \in \mathbb{R}^3$ and $p_{b,c} \in \mathbb{R}^3$ are the coordinates of the two points which violates the collision constraint returning $d_{min} < d_0 + d_{start}$, respectively. As a result, the collision avoidance torques and the relative accelerations for the third task can be evaluated as $u_{av} = B^{-1}(q) \sum_{i=1}^{n_c} J_i^T(q) P f_{av_i}$, where $n_c > 0$ is the number of the detected collisions and J_i is the Jacobian mapping the generalized forces acting on the j-th joint, which is the one at the minimum distance from the obstacles, into the torques that must be applied on the previous $(j-1)$ joints. Notice that P has the same sense as in (13).

Finally, the total control input for the hyper-redundant manipulator which satisfies all the three tasks is

$$u_q = u_v + (I - J^{\#}(q)J(q))u_{stab} + (I - J_{aug}^{\#}(q)J_{aug}(q))u_{av}. \tag{17}$$

8. Case studies

Some case studies are now presented to show the behaviour of the developed architecture. It will be supposed that the rover is placed with its wheels directly in contact with the pipe, considering an initial angle around the tube different from zero. The parameters that appear in the mobile robot model are retrieved from the HYFLIERS prototype. They are $m = 10.6$ kg, $L = 0.23$ m, $R_w = 0.035$ m, $\mu = 0.85$, $R_p \simeq 1.1367$ m. The snake-like arm is composed of $n = 21$ joints. The mass of the first link is 0.0518 kg, the mass of the second link is 0.2540 kg, whereas the masses of the following 18 modules until the E-E are 0.02 kg.

The tests are performed using a standard PC, with an *i7-8750H* CPU and 8 Gb of RAM. The OS chosen is *Ubuntu*, with its release *20.04 LTS Focal Fossa*. All the code is written both in *C++* and MATLAB programming language and the simulations are carried out in *Gazebo v11.0*. The MPC problem is defined using the Acados [37] library. The communication between the software is managed using *ROS Noetic Ninjemys*. The sampling time is set to $Ts = 0.01$ s, and the gains of the MPC and the hybrid controller have been tuned to avoid overshoots at the beginning of the task and reach the goal smoothly resulting in $Q = 0.001 I_2$, $R = [0.1\ 0.01\ 0.01] I_3$, $K_P = 270 I_6$, $K_D = 35 I_6$, $K_{P_f} = I_6$ and $K_{I_f} = 3 I_6$. Moreover, the parameters for the asymptotic stability of the MPC controller that also appear in [25] are set to $\sigma = 10$ and $\delta = 0.05$.

The simulations are performed in order to show how the stabilizing term F_p is necessary to balance the structure when the wheels torque saturate, and moreover to show the performances of the whole inspection task against parametric uncertainties and noisy measures.

8.1. Case 1: Stabilization task

In this first case, it is shown how the torques applied by the wheels can saturate, requiring the additional stabilizing term F_p to overcome issues that can arise when those torques cannot counteract the gravity effect.

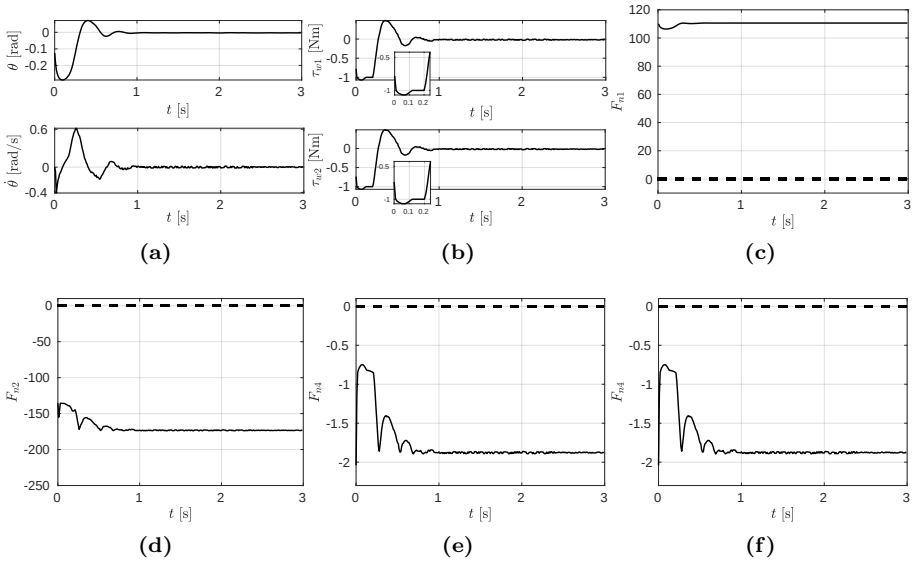

Figure 7: Case study 1: stabilization test. (a)-Time history of the MPC state vector $x = [\theta, \dot{\theta}]$. (b)-Time history of the MPC wheels control inputs $u = [\tau_1, \tau_2]$ with a focus on the saturated input (F_p is ignored in this test since the arm is not used). (c-f)-Time histories to check the fulfilment of the constraints (the dashed lines represent the boundaries). The constraints are re-elaborated during the linearization to appear in the form $f_{n1} > 0$ and $f_{ni} < 0$ with $i = 2, 3, 4$.

The manipulator is placed in a rest-configuration wrapped around the rover, letting the CoM of the entire structure to be fixed in time with respect to the robot base frame. The initial angle of the rover around the pipe is set to $\theta_{rover} \simeq -0.1745$ rad, and both the control and prediction horizons are set equal to $K = 10$.

In Fig. 7.(a) is depicted the evolution of the state in function of the optimal control input shown in Fig. 7.(b). The fulfilment of the constraints is shown in Fig. 7.(c-f), in particular those constraints are expressed through

$$
\begin{aligned}
F_{n1} &:= F_s^T n_0 > 0 && (c),\\
F_{n2} &:= |F_s^T t_0| - \tan(\gamma_s) F_s^T n_0 < 0 && (d),\\
F_{n3} &:= |\tau_{w1}| - \frac{\mu m g R_w L_{1,sx} \cos(\theta)}{L_{1,sx}\cos(\theta_{s,dx}) + L_{1,dx}\cos(\theta_{s,sx})} < 0 && (e),\\
F_{n4} &:= |\tau_{w2}| - \frac{\mu m g R_w L_{1,dx} \cos(\theta)}{L_{1,sx}\cos(\theta_{s,dx}) + L_{1,dx}\cos(\theta_{s,sx})} < 0 && (f).
\end{aligned}
$$

Notice how the control inputs related to the actuation of the wheels saturates at the beginning of the simulation. For this reason, the stabilizing input F_p should

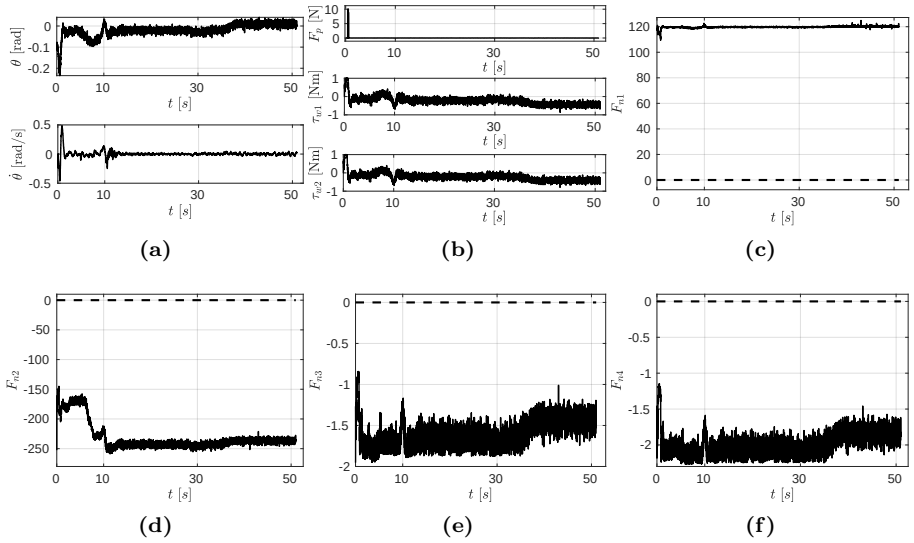

Figure 8: Case study 2: Robustness test. (a)-Time history of the MPC state vector $x = [\theta, \dot{\theta}]$. (b)-Time history of the MPC wheels control inputs $u = [F_p, \tau_1, \tau_2]$. (c-f)-Time histories to check the fulfilment of the constraints (the dashed lines represent the boundaries). The constraints are re-elaborated during the linearization to appear in the form $f_{n1} > 0$ and $f_{ni} < 0$ with $i = 2, 3, 4$.

be used to help stabilizing the structure. The steady-state error is nullified at the end of the control, despite the saturation that afflicts the control inputs.

8.2. Case 2: Inspection task with robustness analysis

In this case study, the desired inspection task is addressed. The goal is to control the manipulator to reach the inspection area under the pipe, avoiding the collisions with the tube and self-collisions of the snake-like arm. During the interaction, the E-E has to track the desired force on the surface maintaining the relative position with the pipe. At the same time, the robot must be balanced on the pipe itself, taking into account the movement of its CoM caused by the new configurations of the manipulator. The initial condition of the MPC algorithm are the same of the previous case. The reference trajectory is composed by two sections: the first one of duration 6 s, necessary to move the E-E towards the first configuration to start the inspection task, the second one of duration 5 s to reach the goal position under the pipe in exam with a collision-free trajectory. The force reference is chosen as a time-variant ramp force profile assuming values equal to 3 N, 6 N, and 9 N as shown in Fig. 9. In an interaction control, it is desirable to guide the desired reference rather than to impose high instantaneous force values to track. Under these last circumstances, problems in both simulated environments and real-scenario could arise.

As explained in section 7.1, a cosine step function interpolated between 0 and

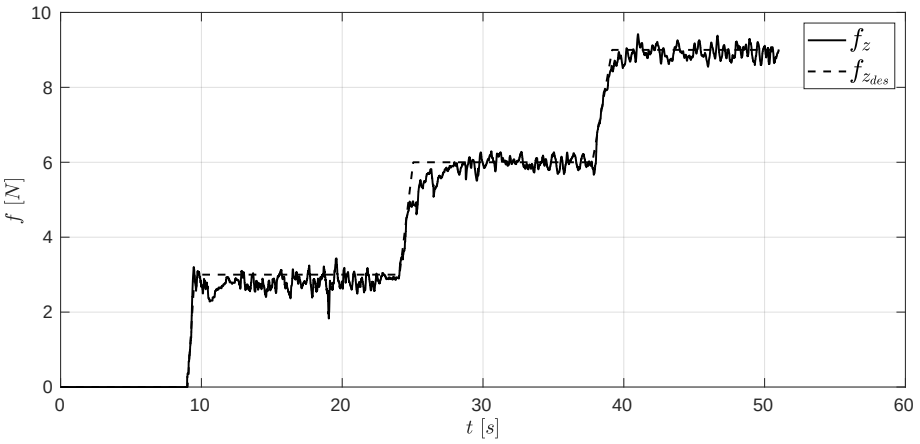

Figure 9: Tracking of 3 different force references. Dashed lines are the desired forces while solid lines are measured forces both expressed in the E-E frame. A ramp force profile is chosen to smoothly control the interaction force

1 is used to smooth the transition between the contact-free and contact-based control modalities. Figure 9 shows how the interaction force is controlled to follow the desired reference without undesired overshoots thanks to the control gains and the switching modality tuning. The collision avoidance torques are computed choosing the following gains: $k_{1_{pipe}} = k_{2_{pipe}} = 0.8$ to compute the avoidance forces to avoid collision with the pipes and $k_{1_{self}} = 1.5$ and $k_{2_{self}} = 35$ to avoid the self-collisions. Finally, $d_{start} = 0.164$ is the starting distance between the object where the force has to act and $d_0 = 0.03$ is a limit distance where a collision may occur.

This test also verifies the robustness of the designed MPC controller by considering a white noise with standard deviation $\sigma = 10^{-2}$ on the state measurements. A parametric uncertainty about the mass of the robot is also considered (50% more of its real value). Figure 8 shows the results of the MPC algorithm during the inspection task. For the instant of time in which τ_{w1} and τ_{w2} saturates, it is shown how the additional control input F_p helps the system to recover this situation. This contribution increases in value as a function of the considered uncertainty. For lower uncertainty values, its magnitude is well below the saturation levels until it reaches saturation for the presented case study.

The control algorithm takes into account also the movement of the CoM of the robot, keeping it in the stable region. The noise in the measure of the angle θ can be seen as a chattering behaviour on the signal plotted in Fig.8.(a-b). The steady-state error in terms of angular position θ, represented in Fig.8.(a), is 0.0269 rad. In Fig.8.(c-f) are represented the fulfilment of the constraints (8). Other plots and case studies can be appreciated in the accompanying video (`https://youtu.be/6YT1jiK7zpY`).

9. Conclusion

The problem of performing NDT measurements with a wheeled mobile manipulator equipped with a snake-like arm is tackled.

The project focuses on developing a controller law that exploits the system's elevated DoF number to complete inspection tasks in hard-to-reach industrial environments. Such operations can be complex and time-consuming, so landing on the inspection spot and exploiting the wheeled platform for the entire execution is an excellent alternative to save the aerial robot battery. Furthermore, using a snake-like manipulator, more than the classic E-E rigid sticks or underactuated arms, has allowed the development of navigation techniques in densely structured environments to reach inaccessible pipe sections without colliding. The redundancy resolution allows both to control the manipulator motion and to design collision avoidance and force propagation algorithms, contributing to the whole structure stabilization on the pipe.

Several preliminary tests have acknowledged that the rover's wheels may saturate during the operations. Therefore, the dynamics of the arm is here exploited by an MPC algorithm to help the rover stabilization over the pipe. This information is, in turn, employed in the null space of a hybrid motion/force control technique with a prioritized redundancy resolution. This framework tracks the desired position and force references at the robot's tip to carry out the sought measurement while avoiding self-collisions and those with the environment.

The discussed results are a good starting point for improving the proposed solution. Future work might include the introduction of various sensors or different control techniques to complete the same assigned tasks: an onboard camera system, for example, might be used to perform image elaboration and detection of particular features on the pipes like welds or cracks, presence of corrosion, or liquid leaks. To this end, different approaches might be used: from the classic OpenCV and Visp libraries to a more promising deep learning approach like YOLO.

An engaging case study could also be the visual and the interaction force feedback merging in a parallel force/vision control law to perform push-&-slide operations. The sliding on a surface with the proposed hybrid platform should be integrated into the optimization problem relaxing one of the proposed hypotheses or directly exploiting the snake-like arm redundancy planning the task in the null space of the Jacobian matrix.

References

[1] D. Schmidt, K. Berns, Climbing robots for maintenance and inspections of vertical structures - A survey of design aspects and technologies, Rob. and Aut. Sys. 61 (12) (2013) 1288-1305.

[2] A. Kakogawa, S. Ma, Design of a multilink-articulated wheeled inspection robot for winding pipelines: AIRo-II, in: 2016 IEEE/RSJ Int. Conf. on Intel. Rob. and Sys., 2016, pp. 2115–2121.

[3] E. Dertien, S. Stramigioli, K. Pulles, Development of an inspection robot for small diameter gas distribution mains, in: 2011 IEEE Int. Conf. on Rob. and Aut., 2011, pp. 5044–5049.

[4] E. Dertien, M. Foumashi, K. Pulles, S. Stramigioli, Design of a robot for in-pipe inspection using omnidirectional wheels and active stabilization, in: 2014 IEEE Int. Conf. on Rob. and Aut., 2014, pp. 5121–5126.

[5] Inspection robotics, [http://inspection-robotics.com/.

[6] B. Ross, J. Bares, C. Fromme, A semi-autonomous robot for stripping paint from large vessels, Int. Jour. of Rob. Res. 22 (2003) 617–626.

[7] J. Resino, A. Jardón, A. Gimenez, C. Balaguer, Analysis of the direct and inverse kinematics of ROMA II robot, in: M. O. Tokhi, G. S. Virk, M. A. Hossain (Eds.), Climbing and Walking Robots, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 869–874.

[8] Appellix, [www.apellix.com].

[9] Texo drone service, [www.texodroneservices.co.uk/blog/56].

[10] Ronik inspectioneering, [http://www.inspectioneering.eu/].

[11] Voliro inspection, [https://www.voliro.com/inspection].

[12] M. A. Trujillo, J. R. Martinez-de Dios, C. Martin, A. Viguria, A. Ollero, Novel aerial manipulator for accurate and robust industrial ndt contact inspection: A new tool for the oil and gas inspection industry, Sensors 19 (6) (2019). doi:10.3390/s19061305.

[13] A. Lopez-Lora, P. Sanchez-Cuevas, A. Suarez, A. Garofano-Soldado, A. Ollero, G. Heredia, MHYRO: Modular HYbrid RObot for contact inspection and maintenance in oil gas plants, in: 2020 IEEE/RSJ Int. Conf. on Intel. Rob. and Sys., 2020, pp. 1268–1275.

[14] C. Wright, A. Johnson, A. Peck, Z. McCord, A. Naaktgeboren, P. Gianfortoni, M. Gonzalez-Rivero, R. Hatton, and H. Choset, Design of a modular snake robot, in: 2007 IEEE/RSJ Int. Conf. on Intel. Rob. and Sys., 2007, pp. 2609–2614.

[15] A.A. Transeth, K.Y. and Pettersen, and P. Liljebäck, A survey on snake robot modeling and locomotion, Robotica 27 (7) (2009) 999–1015.

[16] J. Tang, Y. Zhang, F. Huang, J. Li, Z. Chen, W. Song, S. Zhu, J. Gu, Design and kinematic control of the cable-driven hyper-redundant manipulator for potential underwater applications, Applied Sciences 9 (2019) 1142. doi:10.3390/app9061142.

[17] A. Crespi and A.J. Ijspeert, Online optimization of swimming and crawling in an amphibious snake robot, IEEE Trans. on Rob. 24 (1) (2007) 75–87.

[18] S. Ma, S. Hirose, H. Yoshinada, Development of a hyper-redundant multijoint manipulator for maintenance of nuclear reactors, Adv. Rob. 9 (3) (1994) 281–300.

[19] F. Trebuňa, I. Virgala, M. Pástor, T. Lipták, Ľubica Miková, An inspection of pipe by snake robot, International Journal of Advanced Robotic Systems 13 (5) (2016). doi:10.1177/1729881416663668.

[20] A. M. Andruska, K. S. Peterson, Control of a snake-like robot in an elastically deformable channel, IEEE/ASME Trans. on Mech. 13 (2) (2008) 219–227.

[21] T. W. Danko, P. Y. Oh, A hyper-redundant manipulator for mobile manipulating unmanned aerial vehicles, in: 2013 Int. Conf. on Unm. Airc. Sys.), 2013, pp. 974–981.

[22] Q. Luo, Q. Hu, Y. Zhang, Y. Sun, Segmented hybrid motion-force control for a hyper-redundant space manipulator, Aerospace Science and Technology 131 (2022) 107981. doi:https://doi.org/10.1016/j.ast.2022.107981.

[23] J. Cacace, N. De Silva, G. A. Fontanelli, V. Lippiello, A novel articulated rover for industrial pipes inspection tasks, in: 2021 IEEE/ASME Int. Conf. on Adv. Intel. Mech., 2021.

[24] J. Cacace, G. Fontanelli, V. Lippiello, A novel hybrid aerial-ground manipulator for pipeline inspection tasks, in: 2021 Aer. Rob. Sys. Phys. Interact. with the Envir., 2021.

[25] S. Zhao, F. Ruggiero, G. A. Fontanelli, V. Lippiello, Z. Zhu, B. Siciliano, Nonlinear model predictive control for the stabilization of a wheeled unmanned aerial vehicle on a pipe, IEEE Rob. and Aut. Lett. 4 (4) (2019) 4314–4321.

[26] K.M. Lynch and F.C. Park, Modern Robotics: Mechanics, Planning, and control, Cambridge University Press, 2017.

[27] F. Ruggiero, V. Lippiello, A. Ollero, Aerial manipulation: a literature review, IEEE Robotics and Automation Letters 3 (3) (2018) 1957–1964.

[28] K. M. Lynch, F. C. Park, "Modern Robotics: Mechanics, Planning and Control", Cambridge University Press, 2017.

[29] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, "Robotics: Modelling, Planning and Control", Springer, 2009.

[30] V. Duindam, S. Stramigioli, Modeling and control for efficient bipedal walking robots. A port-based approach, Vol. 53 of Springer Tracts in Advanced Robotics, Springer-Verlag Berlin Heidelberg, 2009.

[31] V. L. Popov, Contact mechanics and friction, Springer Berlin Heidelberg, 2010.

[32] K. Bodie, M. Brunner, M. Pantic, S. Walser, P. Pfändler, U. Angst, R. Siegwart, J. Nieto, Active interaction force control for contact-based inspection with a fully actuated aerial vehicle, IEEE Trans. on Rob. 37 (3) (2021) 709–722.

[33] L. Peric, M. Brunner, K. Bodie, M. Tognon, R. Siegwart, Direct force and pose NMPC with multiple interaction modes for aerial push-and-slide operations, in: 2021 IEEE Int. Conf. on Rob. and Aut,, 2021.

[34] B. Siciliano, J. Slotine, A general framework for managing multiple tasks in highly redundant robotic systems, in: Fifth Int. Conf. on Adv. Rob. 'Rob. in Unstr. Env., 1991, pp. 1211–1216.

[35] G. Antonelli, Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems, IEEE Trans. on Rob. 25 (5) (2009) 985–994.

[36] A. De Santis, A. Albu-Schaffer, C. Ott, B. Siciliano, G. Hirzinger, The skeleton algorithm for self-collision avoidance of a humanoid manipulator, in: 2007 IEEE/ASME Int. Conf. on Adv. Int. Mech., 2007.

[37] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, M. Diehl, ACADOS – A modular open-source framework for fast embedded optimal control, Mathematical Programming Computation (Oct 2021). doi:10.1007/s12532-021-00208-8.
URL https://doi.org/10.1007/s12532-021-00208-8