sensors

Article

Spherical Wrist Manipulator Local Planner for Redundant Tasks in Collaborative Environments

Marcello Chiurazzi [1,2,*], Joan Ortega Alcaide [1,2], Alessandro Diodato [1,2], Arianna Menciassi [1,2] and Gastone Ciuti [1,2]

[1] The BioRobotics Institute, Scuola Superiore Sant'Anna, 56127 Pisa, Italy
[2] Department of Excellence in Robotics and AI, Scuola Superiore Sant'Anna, 56127 Pisa, Italy
[*] Correspondence: marcello.chiurazzi@santannapisa.it; Tel.: +39-3356979421

Abstract: Standard industrial robotic manipulators use well-established high performing technologies. However, such manipulators do not guarantee a safe Human–Robot Interaction (HRI), limiting their usage in industrial and medical applications. This paper proposes a novel local path planner for spherical wrist manipulators to control the execution of tasks where the manipulator number of joints is redundant. Such redundancy is used to optimize robot motion and dexterity. We present an intuitive parametrization of the end-effector (EE) angular motion, which decouples the rotation of the third joint of the wrist from the rest of the angular motions. Manipulator EE motion is controlled through a decentralized linear system with closed-loop architecture. The local planner integrates a novel collision avoidance strategy based on a potential repulsive vector applied to the EE. Contrary to classic potential field approaches, the collision avoidance algorithm considers the entire manipulator surface, enhancing human safety. The local path planner is simulated in three generic scenarios: (i) following a periodic reference, (ii) a random sequence of step signal references, and (iii) avoiding instantly introduced obstacles. Time and frequency domain analysis demonstrated that the developed planner, aside from better parametrizing redundant tasks, is capable of successfully executing the simulated paths (max error = 0.25°) and avoiding obstacles.

Keywords: robotics control; local path planner; task redundancy; collision avoidance strategy; human–robot interaction

check for updates

Citation: Chiurazzi, M.; Alcaide, J.O.; Diodato, A.; Menciassi, A.; Ciuti, G. Spherical Wrist Manipulator Local Planner for Redundant Tasks in Collaborative Environments. *Sensors* **2023**, *23*, 677. https://doi.org/10.3390/s23020677

Academic Editors: Luige Vladareanu, Hongnian Yu, Hongbo Wang and Yongfei Feng

Received: 2 November 2022
Revised: 21 December 2022
Accepted: 31 December 2022
Published: 6 January 2023

1. Introduction

Industrial robotic manipulators use well-established technologies that guarantee high performances in terms of velocity and accuracy. Indeed, typical values of repeatability and joint velocity for industrial manipulators are approximately 0.01 mm and 300°/s, respectively [1]. Despite the clear capabilities of industrial robots, their usage is still very much limited to applications where their workspace is exclusive and humans' presence is forbidden. The standard regulatory ISO 10218-1/2:2011 [2] and ISO/TS 15066:2016 [3] provide requirements, specifications, and guidelines to ensure safety for industrial and collaborative robotic applications and work environments. Nowadays, most industrial robots work inside fences that physically limit the access of humans into the workspace. However, certain tasks such as robotic surgery require the robot to share its workspace with humans. The need for physically close interaction between humans and machines has motivated many groups, both in academia and industry, to study new strategies for safe Human–Robot Interaction (HRI) [4,5]; the followed strategies for a safe HRI can be categorized as post-contact or pre-contact measures.

Post-contact safety measures serve to mitigate the effects of a collision once it has already occurred. Post-contact measures include soft and smooth designs of any potential contact points on the robot structure and the use of integrated sensing capabilities measuring the intensity of such impacts [6,7]; post-contact measures are used to minimize the impact of a collision but also intentional contacts [8,9].

On the other hand, pre-contact strategies use perceptive systems to provide online environment information to the robotic manipulator to generate collision-free trajectories. Path planning algorithms can be classified into: (i) global path planning algorithms, and (ii) local path planning algorithms. Global path planning algorithms solve an optimization problem searching for a free-collision path from an initial configuration to the desired one [10]. Global planners are usually able to find the optimal path in a finite amount of time; however, their computational time requirements can be a limiting factor in dynamic environments. On the other hand, local planners provide a local trajectory to the robotic arm based on the final goal and the local environment information at a certain time. Local planners require much less computational time to be executed and generally cannot guarantee that the generated trajectories will reach the final goal. Local planners are therefore adequate for tasks characterized by having a highly changing objective trajectory (e.g., teleoperation) or highly dynamic environment (e.g., collaboration with humans). Most of the local path planners in literature are based on and/or are inspired by the artificial potential field foundational method introduced by Khatib et al. in [11]. The artificial potential field method assigns a repulsive potential to the obstacles and an attractive potential to the desired goal configuration. Potential field-based methods have proved to effectively avoid collisions in real-time applications [12].

Robotic manipulators executing tasks in non-structured, dynamic environments must guarantee both a safe interaction with the environment and a safe execution of the objective task, especially in critical tasks such as the ones performed by surgical robots. Many such tasks are characterized by having an axis of redundancy aligned with the last joint of the robotic manipulator. Assuming that basic machining operations, such as milling and drilling, only require 5-DoF, the anthropomorphic robot becomes adequate and the task optimizable. The optimization method can be used to exploit the redundancy that certain tasks have. Singularities, joint limits, and collisions were optimized for redundant manipulators [13]. Lukić et al. proposed the optimization of the Cartesian stiffness of a kinematic redundant robot with a null space projection. However, they only considered maintaining position without any specified orientation, which is not the case for machining operations [14]. Several research groups have studied approaches to exploit redundancy in industrial applications. Zanchettin et al. [15] implemented a redundancy resolution criterion that maximizes the manipulator maneuverability to exploit the redundant degree of freedom available on drilling tasks. In [16], Guo et al. presented a novel method based on the Jacobian matrix for computing a performance index based on the stiffness of the robot during machining applications. These studies also paved the way for the use of robotic arms in redundant tasks related to medical scenarios, such as in Focused Ultrasound Surgery [17] or for teleoperation control of a 7-DoF robot manipulator for Minimally Invasive Surgery (MIS) [18].

The manipulation needs, found in the application described in [17], have motivated the development of the novel local planner presented in this work. In this specific robotic scenario, an anthropomorphic manipulator (i.e., an anthropomorphic arm with a spherical wrist) is equipped with a transducer able to stimulate human tissue through ultrasound energy for treating tumors in moving organs. The ultrasound energy is concentrated in a focal spot located along the central axis of the transducer. Hence, the pose of this rotation axis, combined with the manipulator end-effector (EE) linear position, determines the focal spot positioning (i.e., 5-DoF task due to symmetrical tool). The remaining DoF can be used to accomplish secondary tasks such as cable management. Similarly, there are many other scenarios where the position of the joint J_t does not affect the main task, as in welding applications. Hence, it would be useful to describe the EE angular movement decoupling the rotation of the third wrist joint J_t from the rest of the angular motions.

Aim and Organization of the Work

This work aims to develop a local planner that optimizes and simplifies the safe usage of robotic manipulators equipped with a spherical wrist executing redundant tasks in workspaces shared with dynamic obstacles (e.g., humans). The proposed path planner is based on the following features: (i) the rotation of the third wrist joint J_t must be decoupled from the rest of the EE angular motion, (ii) the resulting EE manipulator dynamics should behave as a linear dynamical system, (iii) the collision avoidance strategy must consider the entire surface of the manipulator, and (iv) all the local planner parameters must have a physical meaning.

The present work has four main sections. In Section 2, we first present the theoretical formulation that leads to the local planner and then describe the methods used to validate the proposed local planner making use of an ad hoc simulator. In Section 3, we present and discuss the results of the performed simulations. Finally, we summarize the conclusions in Section 4.

2. Materials and Methods

2.1. Local Planning for Redundant Collaborative Tasks: Theoretical Formulation

In this subsection, we present a new parametrization of the pose of the end-effector in a spherical wrist manipulator that more naturally represents the fundamental degrees of freedom of redundant tasks. The parametrization decouples the rotation of the third wrist joint J_t from the rest of the EE angular motion. We then present the theoretical computation of the disturbance vector D_{EE} based on the collision avoidance strategy. Finally, we present the control law used to implement the local path planner.

2.1.1. Task Parametrization: Separation of the Redundant Axis

The kinematic model of the robotic manipulator depends on the structure of the manipulator [19]. The spherical wrist represents one of the most widely employed joint configurations and its structure is presented in Figure 1. Figure 1 also represents the manipulator base and the manipulator EE reference system following the Denavit–Hartenberg convention [19]. This robotic structure has two important properties: (i) the pose of the $\mathbf{z_{EE}}$ axis depends only on the joints before J_t and (ii) the rotation along $\mathbf{z_{EE}}$ can be independently controlled by means of the J_t. These properties do not depend on the entire robotic structure, but they are intrinsic proprieties of the spherical wrist. We propose a parametrization that separates the EE angular movement into two rotations: (i) a rotation along the axis perpendicular to both the z-axis of the initial orientation, $\mathbf{z_s}$, and the target orientation, $\mathbf{z_t}$, and (ii) a rotation of the joint J_t to the desired rotation along the z-axis.

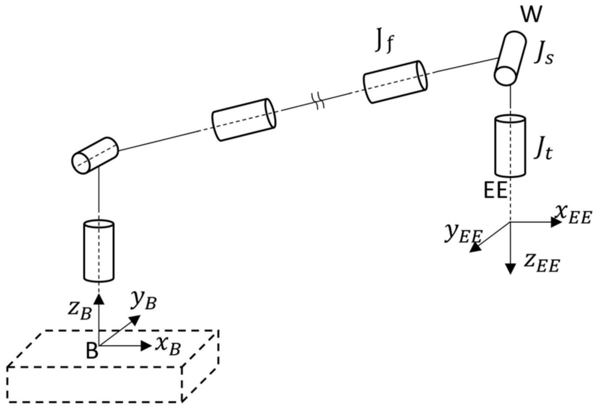

Figure 1. Kinematic model of a 6-DoF manipulator with focus on the spherical wrist, where the center of the spherical wrist (W) and the end-effector (EE) reference system are presented; a possible manipulator base reference system (B) is also represented.

In Figure 2, we present a generic motion of the z-axis together with the described parameters and rotation angles (i.e., θ and γ) along their respective axes of rotation. Figure 2b,c illustrate the parallel and perpendicular views concerning the plane defined by the $\mathbf{z_s}$ and $\mathbf{z_t}$ vectors. Figure 2b presents the θ angle rotation along the $\mathbf{x_\theta}$ axis perpendicular to the $\mathbf{z_s}$ and $\mathbf{z_t}$ plane. On the other hand, the rotation along the $\mathbf{y}_\gamma$ axis performs the out-of-plane rotation. The $\mathbf{y}_\gamma$ axis is defined to be perpendicular to the $\mathbf{x}_\theta$ axis and to the projection of $\mathbf{z}_{\mathbf{EE}}^{\theta}$ into the plane spanned by the θ angle. In the following equations, we formalize the described definitions of the rotation axes $\mathbf{x}_\theta$ and $\mathbf{y}_\gamma$.

$$\mathbf{x}_\theta = \mathbf{z_s} \times \mathbf{z_t} \tag{1}$$

$$\mathbf{y}_\gamma = \mathbf{z}_{\mathbf{EE}}^{\theta} \times \mathbf{x}_\theta \tag{2}$$

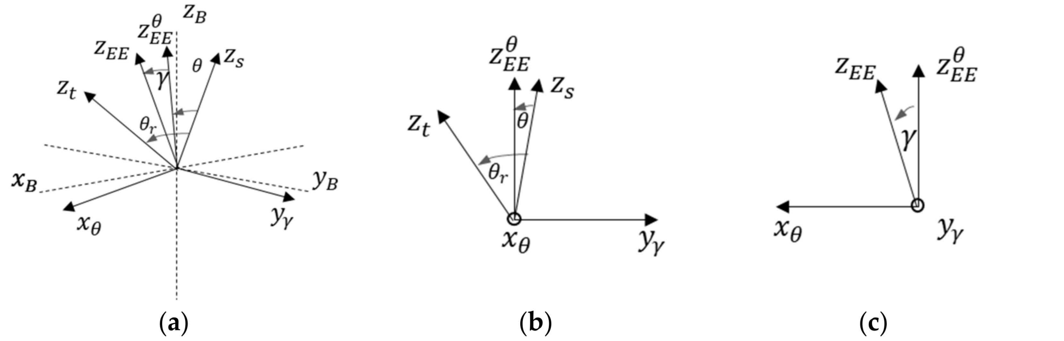

Figure 2. (**a**) Schematic representation of the z-axis overlapping motion for reaching the vector $\mathrm{z_t}$ starting from $\mathrm{z_s}$. The angles θ and γ are defined relative to the rotation axis x_θ and y_γ. (**b**) Section of the plane defined by the vectors $\mathrm{z_t}$ and $\mathrm{z_s}$ spanned by the θ angle. (**c**) Section of the plane spanned by the γ angle perpendicular to the x_θ and $\mathrm{z}_{\mathrm{EE}}^{\theta}$ vector.

We can then compute the $\mathbf{z_{EE}}$ at any instant applying the following ordered rotations to the starting z-axis ($\mathbf{z_s}$).

$$\mathbf{z_{EE}} = \mathrm{R}_{\mathbf{y}_\gamma}(\gamma) R_{\mathbf{x}_\theta}(\theta)\mathbf{z_s} \tag{3}$$

The notation $\mathrm{R}_{\mathbf{a}}(b)$ refers to the rotation along the a-axis by an angle b. Based on the definitions of the rotation axis we will use the equation 3 to express the angular motion of the EE; the following expression can be used to compute the reference angle θ_r (Equation (3)) while, by definition, the reference angle γ_r is always zero.

$$\theta_r = \cos^{-1}\left(\mathbf{z_s^T z_t}\right) \tag{4}$$

More specifically, starting from a random $\mathbf{z_s}$ and using Equations (1)–(3), we can compute the $\mathbf{z_{EE}}$ by multiplying $\mathbf{z_s}$ for two different rotation matrices. A first rotation is performed along $\mathbf{x}_\theta$, which is the axes and the angle needed to align $\mathbf{z_s}$ and $\mathbf{z_t}$ (target orientation). A second rotation is performed along $\mathbf{y}_\gamma$, which is the axes (perpendicular to the plane containing $\mathbf{x}_\theta$ and the projection of $\mathbf{z_{EE}}$) and the angle needed to align $\mathbf{z_t}$ with $\mathbf{z_{EE}}$.

Note that the definition of $\mathbf{x}_\theta$ guarantees that θ_r is always larger than or equal to zero. Given a measured EE orientation, such orientation can be expressed in the described parametrization by applying the following equation where γ_r is the reference target angle (i.e., the angle needed to reach the target orientation), whereas γ is the state variable that evolves (i.e., the real angle).

$$\gamma_r = \sin^{-1}\left(\mathbf{x}_\theta^{\mathbf{T}} \mathbf{z_{EE}}\right) \tag{5}$$

$$\overline{\theta} = \cos^{-1}\left(\mathbf{z_s^T} \mathbf{z}_{\mathbf{EE}}^{\theta}\right) \tag{6}$$

Finally, we compute the last joint J_t directly applying the manipulator inverse kinematics. It is worth noting that the angle position qt of the joint J_t does not influence the z-axis overlapping motion.

Singularity Handling

The described rotation axis is not well defined when $\mathbf{z_s}$ and $\mathbf{z_t}$ are parallel. The $\mathbf{x_\theta}$ axis can be chosen as an arbitrary vector contained in the mutually perpendicular plane. If the scalar product $\mathbf{z_s} \cdot \mathbf{z_t} = -1$, an EE rotation is requested to achieve the desired orientation. This freedom of choice can be used to avoid robotic wrist singularity [19]. Hence, setting the $\mathbf{x_\theta}$ axis as the rotation axis of the joint J_f allows performing the EE rotation through only its joint angle. By doing so, the second joint J_s does not perform any movement, thus allowing it to avoid the robotic wrist singularity.

2.1.2. Disturbance Computation for Collision Avoidance

Inspired by the artificial potential field method [11], we propose to introduce a disturbance vector modifying the planned trajectory based on the distance information between obstacles $\mathrm{O_i}$, i = 1, . . . , N, and each manipulator link. Each obstacle contributes to such a disturbance vector introducing a virtual force $\mathbf{F_{EE_i}}$ and one virtual torque $\mathbf{T_{EE_i}}$. For each obstacle $\mathrm{O_i}$, we define the distance between an obstacle and the manipulator as the minimum distance between the obstacle and each of the M manipulator links. We then compute the virtual force $\mathbf{F_{EE_i}}$ generated by the obstacles using the following piece-wise function.

$$\begin{cases} \mathbf{F_{EE_i}} = F_{MAX} - \frac{(F_{MAX}-F_W)}{d_D}\mathbf{d^m_{O_i}}; \quad \text{if } (\left|\left|d^m_{O_i}\right|\right| < d_D) \\ \mathbf{F_{EE_i}} = F_W - \frac{F_W}{(d_W-d_D)}\left(\mathbf{d^m_{O_i}} - d_D\right); \quad \text{if } (d_D < \left|\left|d^m_{O_i}\right|\right| < d_W) \\ 0; \quad \text{if } (\left|\left|d^m_{O_i}\right|\right| > d_W) \end{cases} \tag{7}$$

In Figure 3, we present the two linear zones: the" warning" and" danger" zones resulting from the proposed Equation (7). In these zones, each obstacle acts as linear springs with different stiffness (i.e., greater stiffness in the" danger" zone). Four parameters characterize the function: (i) the starting distance of the" warning" zone d_W, (ii) the starting distance of the "danger" zone d_D, (iii) the obstacle force $\mathrm{F_W}$ generated at d_W, and (iv) the maximum obstacle force F_{MAX} generated at zero distance.

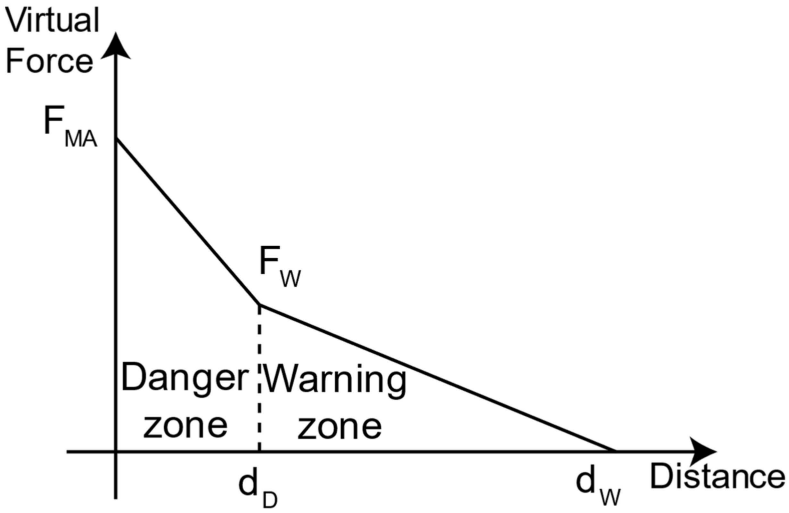

Figure 3. Virtual force generated by an obstacle at a given distance in the environment. The force is proportional to the distance, with different stiffness constants, based on the zone (i.e., "warning" and "danger" zones).

We use the virtual forces computed to generate a virtual torque. We compute the virtual torque as the direct sum of the following two torque components: (i) the torque perpendicular to the $\mathbf{z_{EE}}$ axis ($\mathbf{T^p_{EE_i}}$) and (ii) the torque along the $\mathbf{z_{EE}}$ axis ($\mathbf{T^a_{EE_i}}$). The virtual torque $\mathbf{T^a_{EE_i}}$ is non-zero only when the minimum distance $d^m_{O_i}$ is associated with the last manipulator link. The following equation presents the computation of the virtual torque

$T^{a}_{EE_i}$ based on the force F_{EE_i} vectors components perpendicular to the $\mathbf{z_{EE}}$ axis and the application lever arm, normalized with the maximum lever arm (i.e., $\frac{T_l}{2}$, T_l being the thickness of the last manipulator link).

$$T^{\mathrm{a}}_{EE_{\mathrm{i}}} = \frac{2}{T_l}\left(\left(\mathrm{I}_{3x3} - \mathbf{z_{EE}}\mathbf{z^{T}_{EE}}\right)\left(\mathbf{p_{O_i}} - \mathbf{EE}\right)\right) \times \left(\mathrm{I}_{3x3} - \mathbf{z_{EE}}\mathbf{z^{T}_{EE}}\right)\mathrm{F}_{\mathrm{EE_i}} \tag{8}$$

We compute the virtual torque $T^{p}_{EE_i}$ perpendicular to the $\mathrm{z_{EE}}$ axis as the cross-product between $\mathrm{F_{EEi}}$ and the normalized lever arm along the $\mathbf{z_{EE}}$.

$$T^{p}_{EE_i} = \frac{1}{||\mathbf{z^{T}_{EE}}(\mathbf{EE} - \mathbf{W})||}\left(\mathbf{z_{EE}}\mathbf{z^{T}_{EE}}\left(\mathbf{p_{O_i}} - \mathbf{EE}\right)\right) \times F_{EE_i} \tag{9}$$

Finally, we sum the force and torque contribution of each obstacle obtaining the overall virtual force and torque.

$$\mathbf{F_{EE}} = \sum_i \mathbf{F_{EE_i}} \tag{10}$$

$$\mathbf{T_{EE}} = \sum_i \mathbf{T_{EE^a_i}} + \sum_i \mathbf{T_{EE^p_i}} \tag{11}$$

Once we have the virtual torque $\mathbf{T_{EE}}$ computed in the base manipulator frame, we can express it using the transformation that we present below.

$$\mathbf{T^{s}_{EE}} = \begin{bmatrix} \mathbf{x^{T}_{\theta}} \\ \mathbf{y^{T}_{\gamma}} \\ \mathbf{z^{T}_{EE}} \end{bmatrix} \mathbf{T_{EE}} \tag{12}$$

If we compose the virtual force and torque expressed by the Cartesian and custom axis, respectively, we obtain the disturbance vector $\mathbf{D_{EE}}$.

$$\mathbf{D_{EE}} = \begin{bmatrix} \mathbf{F_{EE}} \\ \mathbf{T^{s}_{EE}} \end{bmatrix} \tag{13}$$

It is worth noting that other virtual forces/torque may be superimposed based on contact forces/torques to implement an impedance/admittance control. The measurement of the actual contact force/torque can be provided by external sensors, such as the sensitive and protective skin presented in [6,20] and/or standard load cells.

2.1.3. Control Law

The state vector $\mathbf{X}$, defined below, represents the manipulator EE pose and it is composed of the Cartesian coordinates of the EE and the angle:

$$\mathbf{X} = [x\ y\ z\ \theta\ \gamma\ q_t]^{\boldsymbol{T}} \tag{14}$$

The decoupled nature of the state variables allows using a decentralized Multiple Input Multiple Output (MIMO) linear dynamical system to control the dynamics of $\mathbf{X}$. Each state variable is controlled through a Single Input Single Output (SISO) system with a closed-loop architecture. Figure 4 depicts the structure of the SISO control system (i.e., equal for the six state variables) and it introduces the state vector reference $\mathbf{X_r}$ and disturbance vector $\mathbf{D_{EE}}$.

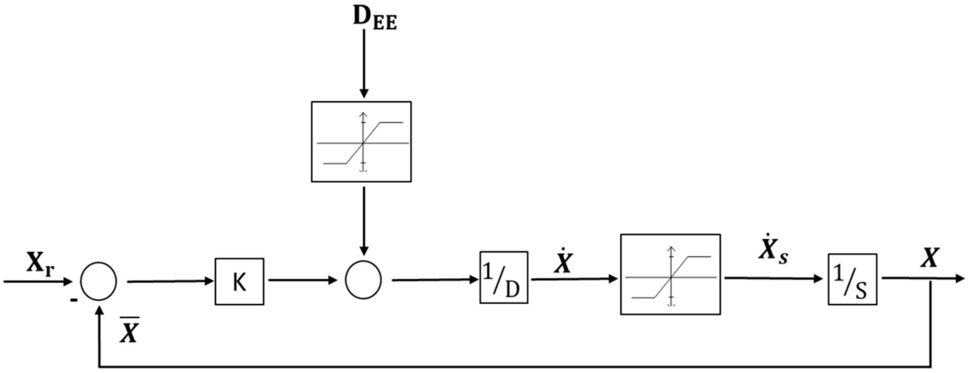

Figure 4. Schematic of the SISO linear system which controls the dynamic evolution of each state variable. The inputs are the reference state vector and the disturbance vector. The saturation for state velocity and disturbance is also reported.

Whenever the disturbance and velocities are not saturated, the system behaves linearly. Under those conditions, we can use the superimposition principle to write the transfer function from the inputs $\mathbf{X_r}$ and $\mathbf{D_{EE}}$ to the output X, as follows:

$$X = \frac{1}{\frac{D}{K}s+1}\mathbf{X_r} + \frac{\frac{1}{K}}{\frac{D}{K}s+1}\mathbf{D_{EE}} = \mathrm{F(s)}\mathbf{X_r} + \frac{1}{K}\mathrm{F(s)}\mathbf{D_{EE}} \tag{15}$$

The transfer function F(s) controls the dynamics of the state vector, parametrized through the damping D and spring K parameters (i.e., the pole of the closed-loop system is $\frac{D}{K}$). Both systems inputs equally influence the $\mathbf{X}$ dynamics with different static gains: 1 for the reference $\mathbf{X_r}$ and $\frac{1}{K}$ for the disturbance $\mathbf{D_{EE}}$. The SISO control system includes two saturations: (i) one saturates the state variable velocity and (ii) the other saturates the maximum amplitude of the disturbance. We define the following linear velocity saturation function.

$$\begin{bmatrix} \dot{\mathbf{x}}_\mathbf{s} \\ \dot{\mathbf{y}}_\mathbf{s} \\ \dot{\mathbf{z}}_\mathbf{s} \end{bmatrix} = \frac{S_l}{\left|\left|\left[\dot{\mathbf{x}}\,\dot{\mathbf{y}}\,\dot{\mathbf{z}}\right]^\mathbf{T}\right|\right|} \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{y}} \\ \dot{\mathbf{z}} \end{bmatrix} \tag{16}$$

Similarly, we define the following saturation function for the EE angular velocity ω.

$$\boldsymbol{\omega} = \begin{bmatrix} \mathbf{x}_\theta & \mathbf{y}_\gamma & \mathbf{z_{EE}} \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\mathbf{y}} \\ \dot{\mathbf{z}} \end{bmatrix} \tag{17}$$

$$\begin{bmatrix} \dot{\theta}_\mathbf{s} \\ \dot{\gamma}_\mathbf{s} \\ \dot{\mathbf{q}}_{\mathbf{t_s}} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_\theta & \mathbf{y}_\gamma & \mathbf{z_{EE}} \end{bmatrix}^{-1} (S_a \frac{\boldsymbol{\omega}}{||\boldsymbol{\omega}||}) \tag{18}$$

We used just two saturation parameters to saturate the EE velocity to maintain the motion direction unchanged, following industry standards [1]. On the other hand, we may use individual saturation constants for each state variable of the disturbance signal.

2.2. *Validation Methodology of the Theoretical Formulation: Simulations*

In this subsection, we present the architecture, the simulation environment, and the tests that we used to validate the proposed local planner. A fundamental element of the simulator is the collision and proximity simulator (CPS). The CPS was developed and used by the authors in [21]. The local planner simulator parameters, used to validate the local planner, are presented in the following subsection, together with the representative simulated scenarios.

2.2.1. Simulator

Simulator Architecture

In Figure 5, we present the general architecture of the proposed local path planner. When a new desired EE pose is provided to the path planner, the reference block computes the state variable's reference $\mathbf{X_r}$ and the axes $\mathbf{x_\theta}$ and $\mathbf{y_\gamma}$. The reference angle θ_r is calculated using Equation (3), while the references for $\mathbf{q_{t_r}}$ are computed using the manipulator inverse kinematics. The linear state variables do not need computation because they directly correspond to EE Cartesian coordinates. The inputs of the MIMO controller are the state variable reference $\mathbf{X_r}$, the $\mathbf{D_{EE}}$ disturbance vector, and the measure of the state variables $\bar{\mathbf{X}}$ provided by the block f_2. The measure of the EE Cartesian coordinate and the joint angle $\mathbf{q_t}$ are directly provided by the manipulator controller. The $\mathbf{D_{EE}}$ disturbance vector is computed based on the CPS information, as presented in the following subsection. The input of the disturbance block is the current manipulator joints angle necessary to bring the virtual manipulator in the simulator. The output of the MIMO block is the state vector $\mathbf{X}$, which is transformed into a manipulator joint trajectory $\mathbf{q_{ref}}$ by means of f_1. Block f_1 uses the manipulator inverse kinematics to compute the references of the joints before J_t is obtained from $\mathbf{X}$. The reference signal to the manipulator $\mathbf{q_{ref}}$ is obtained by adding the state variable q_t to the previously calculated joint reference.

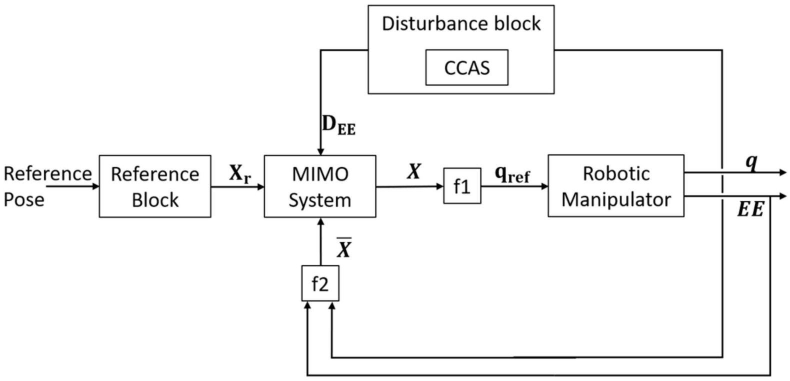

Figure 5. General architecture of the local path planner.

Collision and Proximity Simulator

The collision and proximity simulator is written in C++ language and is based on the Bullet Physics engine [22], following the performance analysis conducted in [23]. The software architecture of the simulator is based on client–server architecture; thus, different applications can interface with the CPS through a simple dedicated interface (e.g., socket applications). To reduce the computational time for collisions and proximity algorithm detection, we simplified the geometries of the manipulator links. The use of a simplified version of the manipulator links is a common practice in the development of collision simulators [24]. The working environment is displayed through an open-source viewer that can be turned off to reduce the CPS computational time. Given a set manipulator pose, the CPS outputs a list of information related to each manipulator link. In particular, the CPS output reports the minimum distance d_{O_i} (with point of application $\mathbf{p_{O_i}}$ and vector $\mathbf{v_{O_i}}$) between each manipulator link and each obstacle in the virtual working environment.

Simulator Parameters

In this work, we use a model of an ABB IRB120 (Zurich, Switzerland) as a representative example of an industrial manipulator with a spherical wrist. The simulator runs in Matlab (MathWorks Inc., Natick, MA, USA) and interfaces with our CPS software [20]. The model of the manipulator is equipped with a tool with maximum thickness, T_l, of 92.8 mm. The dynamics of the manipulator actuator are neglected in the performed simulation. The local planner requires two tuning steps: (i) the tuning of the MIMO system, which controls the dynamics of the manipulator, and (ii) the tuning of the collision avoidance strategy. The MIMO system is defined by the following parameters: (i) the spring parameter K, (ii) the damping parameter D, and (iii) the velocity and disturbance saturation functions. The spring parameter K is set to 1 to not amplify or attenuate the disturbance $\mathbf{D_{EE}}$. The desired linear manipulator dynamics are set to have a settling time (5%) of 3 s (i.e., a pole of the closed loop at 1 rad/s). The pole of the linear system is controlled by the ratio between K and D; therefore, the damping parameter D is set to 1. The saturation thresholds for the linear and angular velocities are set to 100 mm/s and 10°/s, respectively. The disturbance saturation for the linear state variable is set to 100 N, whereas 30 N/mm is set for the angular state variable. These settings lead to 100 mm and 30° of maximum displacement from $\mathbf{X_r}$, since the spring parameter K is 1 when obstacles appear in the workspace. The collision avoidance strategy is only defined by the parameters of Equation (7), which determine the virtual forces generated by obstacles. The parameters set in the simulations are 300 and 100 mm for the d_w and d_D, respectively, while the forces are set to 25 N (F_W) and 100 N (F_{MAX}).

2.2.2. Validation Scenarios

Three generic scenarios have been simulated using the previously described simulator. In the first scenario, the local planner is used to follow a periodic signal as would happen, for example, for a medical robot that needs to adapt its motion to the human breath. Secondly, the local planner is requested to follow a random sequence of step signals that could represent, for example, a set of motions required for welding or teleoperating a robot. Finally, a set of scenarios are simulated to validate the suitability of the local planner to avoid collision with obstacles. It is worth noting that given the linear nature of the local planner, the superimposition principle can be applied to separately investigate the manipulator's EE response to the reference signal $\mathbf{X_{ref}}$ and the disturbance $\mathbf{D_{EE}}$. The robot references generated by the local planner in all cases have been analyzed both in the time and frequency domains. The time domain analysis compares the EE trajectories with the related reference and nominal signals. The nominal signals are computed by exciting the nominal linear system (i.e., F(s)) with the related reference signals. The EE linear motion analysis is performed only by studying the state variable x. This is possible because the Cartesian coordinates are decoupled. On the other hand, angular motion requires a complete EE angular movement investigation (i.e., roll, pitch, and qt angles). The frequency analysis is performed by comparing the spectrum of the joint reference $\mathbf{q_{ref}}$ with the spectrum of nominal dynamics (which uses the RPY parametrization). This serves to investigate how the inverse kinematics affect the joint reference $\mathbf{q_{ref}}$ spectrum, also evaluating its suitability for standard manipulator actuator joints. All the simulations are performed ensuring no velocity saturation occurs.

Periodic Signal

The reference signals used for the periodic signal following the scenario are generated using the roll, pitch, and yaw (RPY) parametrization performed in local axes. The 6-DoF sinusoidal trajectories are composed of three harmonic frequencies (i.e., 0.4, 0.2, and 0.1 rad/s), both for the linear and angular coordinates. We have used the following reference trajectories for the EE Cartesian position and orientation.

$$\begin{bmatrix} EE_r^x \\ EE_r^y \\ EE_r^z \end{bmatrix} = 50 \begin{bmatrix} \sin(0.4t) \\ \sin(0.2t) \\ \sin(0.1t) \end{bmatrix} + \begin{bmatrix} 300 \\ 0 \\ 200 \end{bmatrix} \tag{19}$$

$$\begin{bmatrix} R_r \\ P_r \\ q_{t_r} \end{bmatrix} = 25 \begin{bmatrix} \sin(0.4t) \\ \sin(0.2t) \\ \sin(0.1t) \end{bmatrix} \tag{20}$$

Sequence of Step Signals

A sequence of 100 6-DoF step reference signals has been randomly generated using the roll, pitch, and yaw (RPY) parametrization in local axes. It is worth noting that the roll and pitch angles define the pose of the target axis $\mathbf{z_t}$, whereas the yaw angle is directly related to the angle position qt of the joint J_t [18]. The EE Cartesian position for the 6-DoF step reference is randomly generated in a cube with 200 mm side centered.

Figure 6 reports angular motion from an initial to the desired orientation of the manipulator's EE (depicted in blue). In red, the time evolution of the EE orientation sampled at 0.1 s is reported. The inset represents the θ dynamical response for the depicted angular motion at $[300, 0, 200]^T$ mm, whereas the EE orientations are randomly generated using RPY parametrization with a maximum amplitude of 25°.

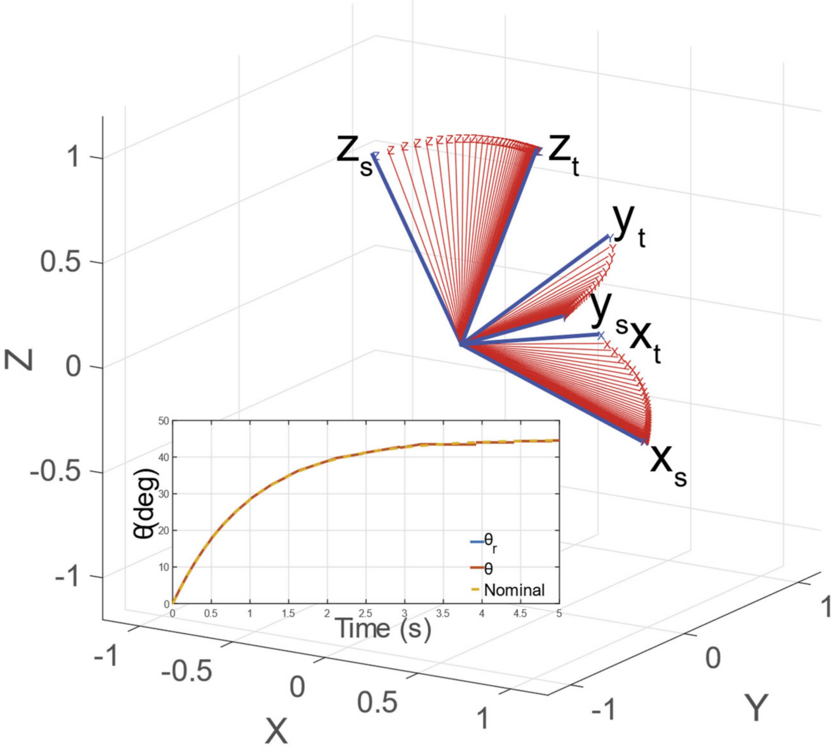

Figure 6. Angular motion from an initial to the desired orientation of the manipulator's EE (depicted in blue). In red, the time evolution of the EE orientation sampled at 0.1 s is reported. The inset represents the θ dynamical response for the depicted angular motion.

Close-Obstacle Collision Avoidance

The collision avoidance strategy is assessed by inserting obstacles (represented as spheres) in the workspace without varying the EE reference position ($[300, 0, 300]^T$ mm with zero RPY angles). Six different configurations of obstacles are chosen as the case studies. The first three simulations include a single obstacle positioned at different $\mathbf{z_b}$ coordinates (i.e., 350, 450, and 550 mm) with x_b = 300 mm and $y_b = -150$ mm. The other simulations include multiple obstacles (i.e., 2 and 4 spheres) to assess the superposition of the proposed collision avoidance strategy. Two simulations are performed with two obstacles: one has the obstacles on the same side of the manipulator, $y_b = -150$ mm) and the other has the obstacles on opposite sides, one in $y_b = -150$ mm and one in y_b = 150 mm). The last simulation is performed in a symmetric configuration with four obstacles placed around the manipulator at different z_b coordinates.

3. Results

3.1. Time Domain Analysis Results

In Figure 7, we report the evolution of x, the nominal dynamics, and the relative step reference signal x_r. As can be observed, the dynamics of x follow the nominal dynamical response with the tuned settling time. Indeed, the maximum error between the real and nominal dynamics is negligible (i.e., ∼6–10 mm). Figure 6 shows the angular motion trajectory (in red), sampled at 10 Hz, from a starting orientation to a target orientation (in blue). In the absence of any disturbances, the motion of the $\mathbf{z_{EE}}$ axis evolves along the plane defined by the $\mathbf{z_s}$ and $\mathbf{z_t}$ vectors, driven by the state variable θ. As can be seen in Figure 6, the θ dynamics are equal to the desired and nominal dynamics tuned with the K and D parameters. The angle position qt belongs to **X;** thus, it evolves identically to the nominal response. Hence, the linear and angular EE motions are linked to a linear dynamical system with the imposed settling time for step references. Figure 8a,b report the dynamic evolution of the coordinates x and qt, their nominal dynamics, and the relative signals reference (see Equations (19) and (20)). The roll and pitch dynamics are reported in Figure 8c,d, respectively. The graphs present an almost perfect match between the nominal and real dynamics; indeed, the errors between them are 0.08 and 0.25° for roll and pitch angles, respectively.

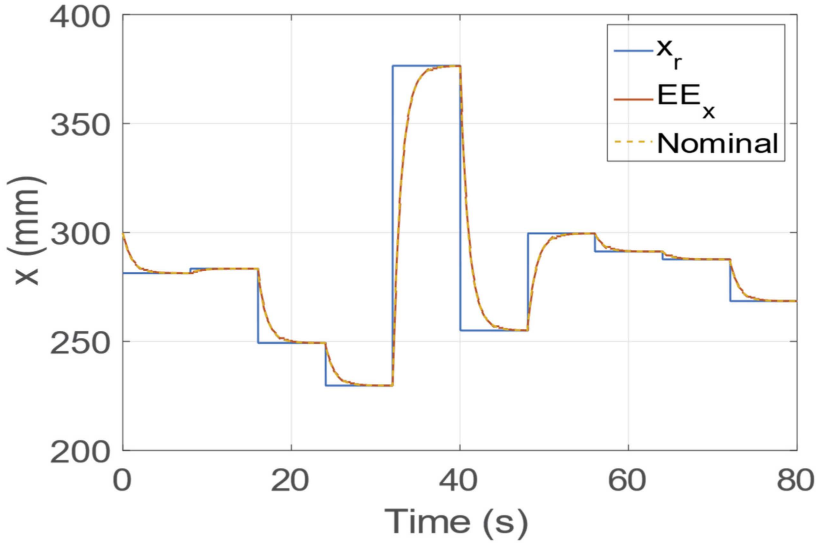

Figure 7. Dynamic evolution of the x coordinate with the relative reference and nominal signals.

3.2. Frequency Domain Analysis Results

The results of the frequency analysis for step and sinusoidal paths are reported in Figure 9a,b, respectively. The graphs report the mean and the maximum spectrum of the joint's references $\mathbf{q_{ref}}$ and the mean spectrum of the nominal dynamics (parametrized with RPY angle). Figure 9a shows that the mean and maximum $\mathbf{q_{ref}}$ spectra are very similar to the spectrum of the nominal dynamics. Therefore, the non-linearity introduced by the manipulator inverse kinematics does not significantly affect the $\mathbf{q_{ref}}$ spectrum. On the other hand, differences between the nominal spectrum and $\mathbf{q_{ref}}$ spectrum can be observed for the sinusoidal path. The manipulator inverse kinematics introduces some components multiple of the exciting input frequencies (i.e., ultra-harmonic frequencies), highlighted in Figure 9b. Nevertheless, the non-desired ultra-harmonic frequencies are attenuated after the linear system band-pass, becoming negligible with the increase in frequency. Indeed, the maximum value of the $\mathbf{q_{ref}}$ spectrum is 19.87 (100%), whereas the maximum values after 1 rad/s and 1 Hz are 0.23 (1.16 %) and 0.02 (0.10 %), respectively. Therefore, the dynamics of the joint reference $\mathbf{q_{ref}}$ are suitable for typical robotic manipulator actuator joints, since its band-pass is larger than 1 Hz [1,25].

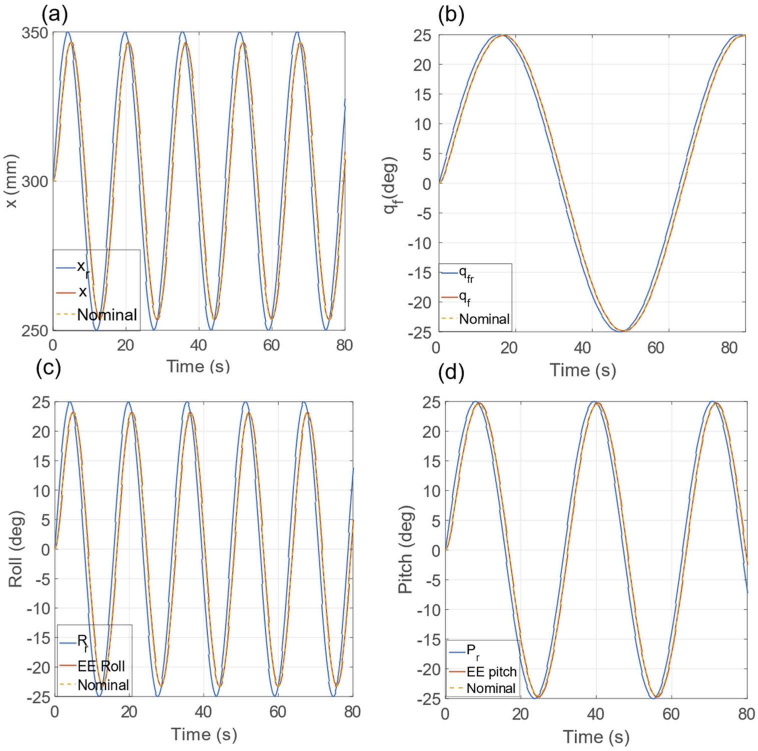

Figure 8. Dynamic evolution of x (**a**), q_t (**b**), roll (**c**), and pitch (**d**) variables compared with the nominal dynamics and the relative reference signal for the sinusoidal input.

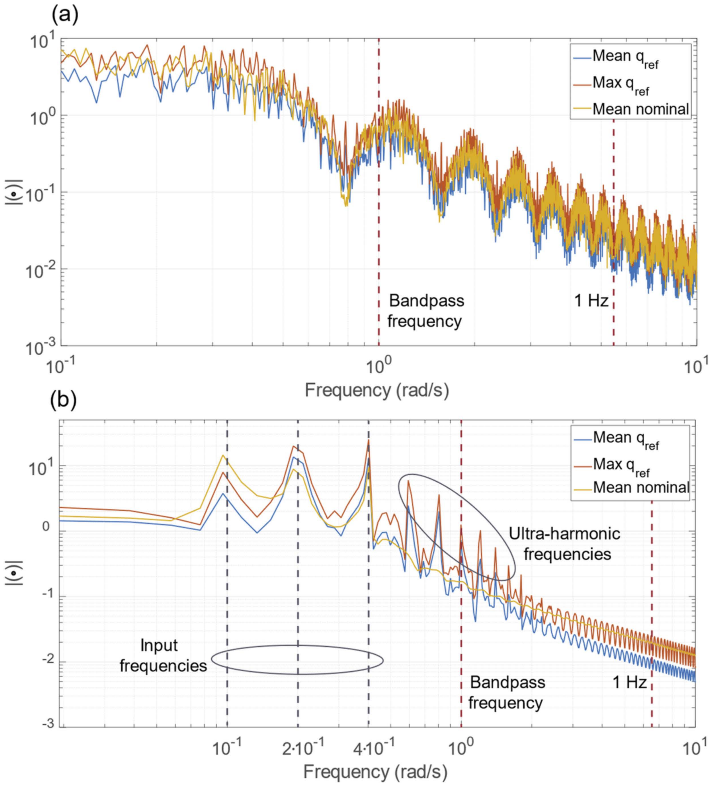

Figure 9. (**a**) Frequency analysis for the step path. The spectra of the mean and the maximum q_{ref} signal are presented, compared with the spectrum of the nominal dynamics highlighting the band-pass frequency of F(s). (**b**) Frequency analysis for the sinusoidal path. The spectra of the mean and the maximum q_{ref} signal are depicted, compared with the spectrum of the nominal dynamics highlighting some notable frequencies.

3.3. Close-Obstacle Collision Avoidance

Figure 10 report the initial poses of the validation scenarios. The x_b coordinate is fixed to 300 mm for all the simulated obstacles, as can be observed in Figure 10a. This choice allows us to describe the EE movements through just two of the state variables (y and γ) without loss of generality. Indeed, the forces produced by the obstacles are principally exerted along $\mathbf{y_b}$. These forces cause a torque along the $\mathbf{x_b}$ axis because the lever arm is mainly along $\mathbf{z_b}$. Subject to this potential field, the resultant EE motion is mainly a translation along $\mathbf{y_b}$ and a rotation along $\mathbf{x_b}$. We have decided to describe this rotation with the state variable γ, responsible for the collision avoidance strategy. The results presented in Figure 10 show how the value of γ increases and the value of y decreases when the obstacle z_B coordinate increases. As could be expected, the local planner responds to the configuration presented in Figure 10d, where the obstacle is closer to the manipulator's wrist than to the manipulator EE, by separating the wrist from the obstacle and keeping the EE close to the Cartesian reference position (i.e., y_r = 0).

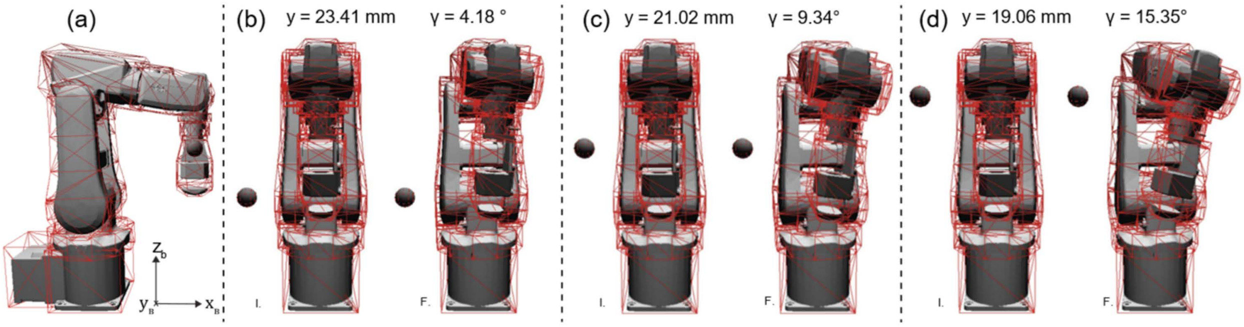

Figure 10. (**a**) Lateral view of the robotic manipulator with one obstacle. The direction of the base reference system axes is also reported. (**b**) The initial (I.) and final (F.) pose of the manipulator when the obstacle is placed at 350 mm on the z_B coordinate. (**c**) The initial and final pose of the manipulator when the obstacle is placed at 450 mm on the z_B coordinate. (**d**) The initial and final pose of the manipulator when the obstacle is placed at 550 mm on the z_B coordinate.

Figure 11 illustrates the resulting dynamics of the state variables y, γ, q_t, and the time response of the minimum distance for the three performed simulations with one obstacle. Figure 11a,b present that the fastest change of the state variables happens when the obstacle is at 550 mm due to the initial obstacle distance (i.e., 58 mm versus 85 mm, as highlighted in Figure 11d). During the initial phase, the distance constantly increases because the angular disturbance saturation is active (the torque is 50.09 N/m at 1 s on state variable γ). The resulting response allows us to conclude that the state variable's responses can be esteemed as a saturated linear dynamical response with different steady-state values. Figure 11c shows that the state variable q_t remains zero for the obstacle at 550 mm given that the obstacle is closer to the penultimate manipulator link than to the last manipulator link. Figure 12 reports the final poses of the validation scenarios. Figure 12a reports the results of the simulations with two obstacles on the same side of the manipulator. The final value of the state variables y and γ is larger than those achieved in the one-obstacle simulations, as expected given the addictive nature of the collision avoidance strategy. The results of the simulation with obstacles on different sides of the manipulator are presented in Figure 12b, resulting in a negative y due to the proximity of the lower obstacle to the EE. Finally, the results of the four obstacles simulation presented in Figure 12c show no significant manipulator motion, as can be expected from the symmetrical obstacle configuration.

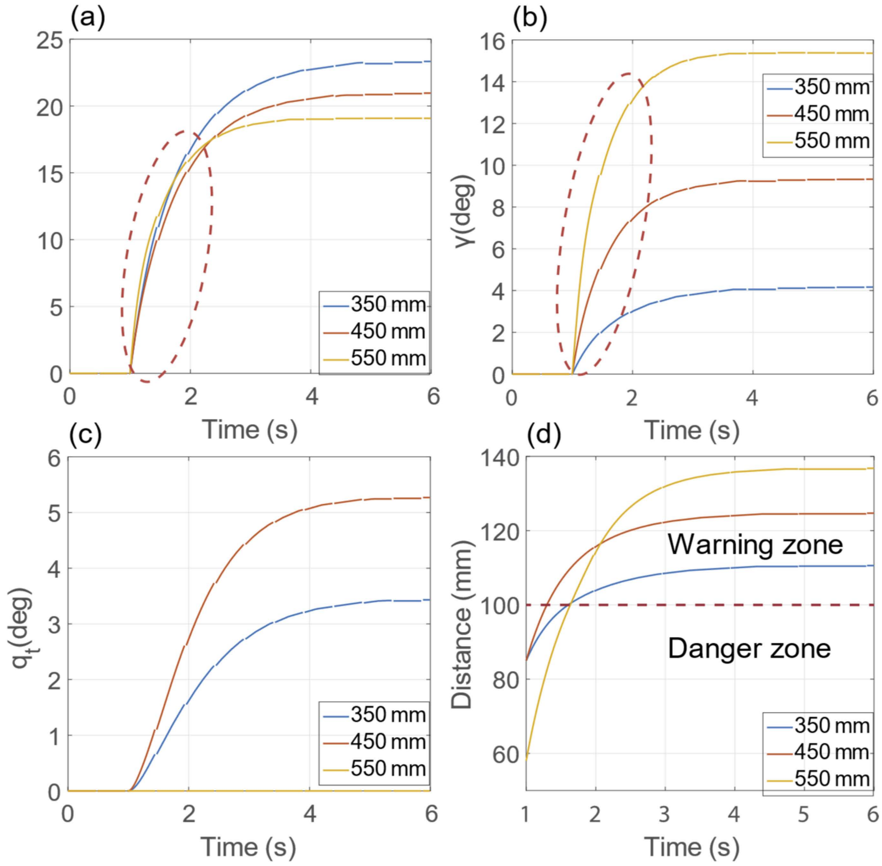

Figure 11. Dynamic evolution of the y (**a**), γ (**b**), q_t (**c**) state variables and time response (**d**) of the minimum distance for the three performed simulations with one obstacle.

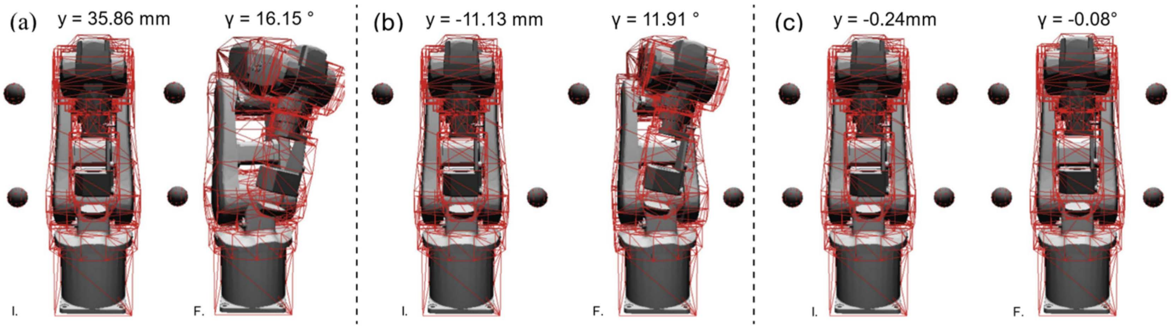

Figure 12. (**a**) Initial and final pose of the manipulator when two obstacles are placed on the same manipulator side. (**b**) The initial (I.) and final (F.) pose of the manipulator when two obstacles are placed on different manipulator sides. (**c**) The initial and final pose of the manipulator with four symmetric obstacles.

4. Conclusions

The described local planner provides a more natural way of describing 5-DoF tasks by using a parametrization of the EE orientation that decouples the rotation of the third wrist joint J_t from the rest of the EE angular motion. The developed parametrization represents the physical behavior of the manipulator in a decoupled manner, facilitating both the interpretation and the tuning of the control parameters. Indeed, θ represents the in-plane rotation along the minimal path, the γ angle represents the out-of-plane rotation, and qt is the position of the third joint of the spherical wrist. The proposed local planner, based on a decentralized MIMO linear system with closed-loop architecture, has been demonstrated to allow the imposition of the EE dynamics to behave as a first-order linear system, facilitating any desired tuning of the EE dynamic response. Approaches using robot redundancy allow us to solve and locally optimize the robot path planning in a dynamic non-structured environment where the manipulator employs potential field approaches. In this regard, the presented approach enables industrial and medical applications where robot stiffness and dexterity can greatly improve task efficiency. It is worth noting that the proposed parametrization can be easily adapted to control 7-DoF manipulators by adding the elbow angle introduced in [26,27] into the state vector **X**. Additionally, the proposed local planner integrates a custom collision avoidance strategy that has proven to successfully deform the reference trajectory to maintain the manipulator separated from surrounding obstacles. The proposed collision avoidance strategy has proven to enhance human safety with a computationally efficient and simple-to-tune disturbance vector that does not require setting control points onto the robotic manipulator.

Author Contributions: Conceptualization, M.C., A.D., J.O.A. and G.C.; methodology, M.C., J.O.A. and A.D; software, A.D.; validation, M.C. and J.O.A.; formal analysis, J.O.A. and A.D.; investigation, M.C., J.O.A. and A.D.; resources, G.C. and A.M.; data curation, M.C. and A.D.; writing—original draft preparation, M.C., J.O.A. and A.D.; writing—review and editing, M.C., A.D., A.M. and G.C.; visualization, M.C., J.O.A. and A.D.; supervision, A.M. and G.C.; project administration, A.M. and G.C.; funding acquisition, A.M. and G.C. All authors have read and agreed to the published version of the manuscript.

Funding: The work described in this paper was partially supported by the European Commission within the frameworks of the FUTURA European Project—FP7/2007-2013 (GA: 611963) and Endoo European Project—H2020-ICT-24-2015 (GA: 688592).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Starting from random initial conditions/constraints (i.e., obstacle position, number of obstacle), as described in Sections 2.2.1 and 2.2.2, data about the relative distances between the robot manipulator and obstacles were computed by our CPS software [20]; the model presented in Section 2.1.3 was then used to compute the robot trajectories in the cartesian space.

Acknowledgments: The authors wish to thank the partners involved in both the FUTURA (www.futuraproject.eu/beneficiaries) and Endoo European projects (https://cordis.europa.eu/project/id/688592/it).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. *IRB 120 ABB's 6 Axis Robot—For Flexible and Compact Production*; ABB Robotics: Zurich, Switzerland, 2021. Available online: https://new.abb.com/products/robotics/industrial-robots/irb-120 (accessed on 31 October 2022).
2. Svenska Institutet för Standarder. *Robots for Industrial Environments-Safety Requirements*; Svenska Institutet för Standarder: Stockholm, Sweden, 2006.
3. Matthias, B. *ISO/TS 15066*; Collaborative Robots Present Status. International Organization for Standardization: Geneva, Switzerland, 2016.
4. Haddadin, S.; Albu-Schäffer, A.; Hirzinger, G. Requirements for Safe Robots: Measurements, Analysis and New Insights. *Int. J. Robot. Res.* **2009**, *28*, 1507–1527. [CrossRef]
5. Hentout, A.; Aouache, M.; Maoudj, A.; Akli, I. Human–robot interaction in industrial collaborative robotics: A literature review of the decade 2008–2017. *Adv. Robot.* **2019**, *33*, 764–799. [CrossRef]
6. Mazzocchi, T.; Diodato, A.; Ciuti, G.; De Micheli, D.M.; Menciassi, A. Smart sensorized polymeric skin for safe robot collision and environmental interaction. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Hamburg, Germany, 28 September–2 October 2015; pp. 837–843. [CrossRef]
7. Zinn, M.; Roth, B.; Khatib, O.; Salisbury, J.K. A New Actuation Approach for Human Friendly Robot Design. *Int. J. Robot. Res.* **2004**, *23*, 379–398. [CrossRef]
8. O'Neill, J.; Lu, J.; Dockter, R.; Kowalewski, T. Practical, stretchable smart skin sensors for contact-aware robots in safe and collaborative interactions. In Proceedings of the IEEE International Conference on Robotics and Automation, Washington, DC, USA, 26–30 May 2015; pp. 624–629. [CrossRef]
9. García, J.G.; Robertsson, A.; Ortega, J.G.; Johansson, R. Sensor Fusion for Compliant Robot Motion Control. *IEEE Trans. Robot.* **2008**, *24*, 430–441. [CrossRef]

10. Lavalle, S.M. PLANNING ALGORITHMS. Available online: http://planning.cs.uiuc.edu/ (accessed on 31 October 2022).
11. Khatib, O. Real-Time Obstacle Avoidance for manipultors and mobile robots. In *Autonomous Robot Vehicles*; Springer: New York, NY, USA, 1986; pp. 396–404.
12. Haddadin, S.; Urbanek, H.; Parusel, S.; Burschka, D.; Rossmann, J.; Albu-Schäffer, A.; Hirzinger, G. Real-time reactive motion generation based on variable attractor dynamics and shaped velocities. In Proceedings of the IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010—Conference Proceedings, Taipei, Taiwan, 18–22 October 2010; pp. 3109–3116. [CrossRef]
13. Xiao, W.; Huan, J. Redundancy and optimization of a 6R robot for five-axis milling applications: Singularity, joint limits and collision. *Prod. Eng.* **2012**, *6*, 287–296. [CrossRef]
14. Lukić, B.; Petrič, T.; Žlajpah, L.; Jovanović, K. KUKA LWR Robot Cartesian Stiffness Control Based on Kinematic Redundancy. In *Advances in Intelligent Systems and Computing*; Springer: Cham, Switzerland, 2020; Volume 980, pp. 310–318. [CrossRef]
15. Zanchettin, A.M.; Rocco, P.; Robertsson, A.; Johansson, R. Exploiting task redundancy in industrial manipulators during drilling operations. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 128–133. [CrossRef]
16. Guo, Y.; Dong, H.; Ke, Y. Stiffness-oriented posture optimization in robotic machining applications. *Robot. Comput. Manuf.* **2015**, *35*, 69–76. [CrossRef]
17. Cafarelli, A.; Mura, M.; Diodato, A.; Schiappacasse, A.; Santoro, M.; Ciuti, G.; Menciassi, A. A computer-assisted robotic platform for Focused Ultrasound Surgery: Assessment of high intensity focused ultrasound delivery. In Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, Milan, Italy, 25–29 August 2015; pp. 1311–1314. [CrossRef]
18. Su, H.; Sandoval, J.; Makhdoomi, M.; Ferrigno, G.; De Momi, E. Safety-Enhanced Human-Robot Interaction Control of Redundant Robot for Teleoperated Minimally Invasive Surgery. In Proceedings of the IEEE International Conference on Robotics and Automation, Brisbane, QLD, Australia, 21–25 May 2018; pp. 6611–6616. [CrossRef]
19. Siciliano, B.; Khatib, O. *Springer Handbook of Robotics*; Springer Science & Business Media: Berlin, Germany, 2008; Available online: https://link.springer.com/content/pdf/bfm:978-3-319-32552-1/1.pdf (accessed on 1 November 2022).
20. Chiurazzi, M.; Garozzo, G.G.; Dario, P.; Ciuti, G. Novel Capacitive-Based Sensor Technology for Augmented Proximity Detection. *IEEE Sens. J.* **2020**, *20*, 6624–6633. [CrossRef]
21. Chiurazzi, M.; Diodato, A.; Vetrò, I.; Alcaide, J.O.; Menciassi, A.; Ciuti, G. Intrinsically Distributed Probabilistic Algorithm for Human–Robot Distance Computation in Collision Avoidance Strategies. *Electronics* **2020**, *9*, 548. [CrossRef]
22. Takaya, K.; Asai, T.; Kroumov, V.; Smarandache, F. Simulation environment for mobile robots testing using ROS and Gazebo. In Proceedings of the 2016 20th International Conference on System Theory, Control and Computing, ICSTCC 2016—Joint Conference of SINTES 20, SACCS 16, SIMSIS 20—Proceedings, Sinaia, Romania, 13–15 October 2016; pp. 96–101. [CrossRef]
23. Balamurugan, B.; Maheswari, K.G.; Skariah, A.; Malathi, V.; Nalinipriya, G. Acceleration of bullet physics physical simulation library using GPU and demonstration on a set-Top box platform. In Proceedings of the ACM International Conference Proceeding Series, Udaipur, India, 4–5 March 2016. [CrossRef]
24. Xia, J.; Jiang, Z.; Liu, H.; Cai, H.; Wu, G. A Novel hybrid safety-control strategy for a manipulator. *Int. J. Adv. Robot Syst.* **2014**, *11*, 2014. [CrossRef]
25. Huber, J.E.; Fleck, N.A.; Ashby, M.F. The selection of mechanical actuators based on performance indices. *Proc. R. Soc. Lond.* **1997**, *453*, 2185–2205. [CrossRef]
26. Zanchettin, A.M.; Bascetta, L.; Rocco, P. Achieving Humanlike Motion: Resolving Redundancy for Anthropomorphic Industrial Manipulators. *IEEE Robot. Autom. Mag.* **2013**, *20*, 131–138. [CrossRef]
27. Zanchettin, A.M.; Rocco, P. Reactive motion planning and control for compliant and constraint-based task execution. In Proceedings of the IEEE International Conference on Robotics and Automation, Seattle, WA, USA, 26–30 May 2015; pp. 2748–2753. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.