# Nonlinear Model Predictive Control for Human-Robot Handover with Application to the Aerial Case

Gianluca Corsini, Martin Jacquet, Hemjyoti Das, Amr Afifi, Daniel Sidobre, Antonio Franchi

# Nonlinear Model Predictive Control for Human-Robot Handover with Application to the Aerial Case

Gianluca Corsini[1], Martin Jacquet[1], Hemjyoti Das[2], Amr Afifi[2], Daniel Sidobre[1], Antonio Franchi[2,1]

***Abstract*— In this article, we consider the problem of delivering an object to a human coworker by means of an aerial robot. To this aim, we present an ergonomics-aware Nonlinear Model Predictive Control (NMPC) designed to autonomously perform the handover. The method is general enough to be applied to any mobile robot with a minimal adaptation of the robot model. Our formulation lets the NMPC steer the robot toward a handover location optimizing the human coworker ergonomics metrics, which includes the predicted joint torques of the human. The motion task is expressed in a frame relative to the human, whose motion model is included in the equations of the NMPC. This allows the controller to reactively adapt to the human movements by predicting her future poses over the horizon. The control framework also accounts for the problem of maintaining visibility on the human coworker, while respecting both the actuation and state limits. A safety barrier is also embedded in the controller to avoid any risk of collision with the human partner. Realistic simulations are used to validate the feasibility of the approach and the source code of the implementation is released open-source.**

## I. Introduction

Aerial Robots (AR) have attracted a lot of interest in the robotics community within the last decades. The growing research attention is motivated by their remarkable agility and maneuverability, the modularity for the onboard sensing equipment, and the availability of heterogeneous designs. Multi-Rotor Aerial Vehicles (MRAVs) have been deployed in numerous applications, either contactless [1] or requiring physical interaction with the environment [2].

There are multiple examples of real-world cases where the use of an aerial robot is advantageous. A particularly relevant one is in work environments at height, such as, e.g., wind turbines, large construction sites or power transmission lines [3]. These environments usually require specialized and trained personnel employing expensive equipment and special vehicles. Carrying and accessing different tools in these challenging environments would require cumbersome postures and loss of focus from the current activity. The use of aerial vehicles as robotic coworkers, in these cases, can facilitate the tasks done by the operators. In those scenarios, an aerial robot can easily fly to the target location while carrying the additional payload of tools, relieving the human operator from carrying extra equipment. Therefore, there is a clear opportunity for using aerial robots and particularly multi-rotors to reduce the physical and cognitive load experienced by workers at height. To achieve the aforementioned benefits, aerial robots should explicitly take into account the human's ergonomics and safety.

Despite the clear potential, the use of aerial robots in scenarios involving working closely with human operators is still limited. On the contrary, the literature on Human-Robot Interaction (HRI) involving a ground robot and human partners is wide and mature. Recent works have investigated the use of manipulators to assist a human coworker in the manipulation of heavy and bulky objects or during assembly task [4], [5].

The problem of object handover has been studied numerous times in the literature. In general, an object handover is characterized by multiple phases [6], the *approach*, *reach*, and *transfer* phases. Most works treat each phase separately, with some notable exceptions. In [7], the authors propose a control architecture for fluid handovers, that tackles all the phases of the handover in a unified way. They consider the interactions arising during the handover, and in particular, their controller is capable of minimizing the unwanted wrench components that are not used for moving and carrying the object. The proposed control scheme does not explicitly include safety and ergonomics, which are important in a control framework that is designed to enable the collaboration between aerial robots and humans, in particular at height. The idea of including the human's comfort and ergonomics in the robot's control and planning software has been also treated in the literature. One of the earlier works in this area is [8], which develops a manipulation planner that takes into account various human aspects, as the human's ergonomics and field of view, amongst others. The methods of including human ergonomics in robot controllers evolved further, e.g., in [9] the authors develop a method to estimate the joint torques of the human based on the whole body dynamic model, and then control a ground mobile manipulator to minimize the overloading of the human joints.

As identified is [10], an important aspect of human robot collaboration including object handovers is the human perception. Most of the works, that focus on control and planning aspects rely on some sort of human instrumentation for perception of the human subject through sensors like motion capture systems via reflective markers, wearable

[1]LAAS-CNRS, Université de Toulouse, CNRS, INSA, UPS, Toulouse, France, gianluca.corsini@laas.fr, martin.jacquet@laas.fr, daniel.sidobre@laas.fr, antonio.franchi@laas.fr

[2]Robotics and Mechatronics lab, Faculty of Electrical Engineering, Mathematics & Computer Science, University of Twente, Enschede, The Netherlands h.das@utwente.nl, a.n.m.g.afifi@utwente.nl, a.franchi@utwente.nl

This work was partially funded by the ANR, under the Projects ANR-18-CE33-0001 'The flying coworker' and ANR-17-CE33-0007 'MuRoPhen', and by the European Commission project H2020 AERIAL-CORE (EC 871479).

suits, etc. Indeed, a tight coupling between control and perception is critical for the success of such tasks, since the loss of visibility over the coworker would jeopardize the maneuver. The problem of perception constrained control has been tackled previously by some of the authors of this paper in prior work [11], in which a perception constrained motor level nonlinear model predictive controller (NMPC) for generic aerial vehicles is presented to keep specific features in the camera field of view (FoV) while respecting the system actuation limits.

Going beyond the achievements of [11], this work presents an NMPC framework that can also handle multiple aspects of HRI in general, with a special focus on the aerial case. We introduce an NMPC formulation that includes human's ergonomics and comfort as an objective, while enforcing perception and actuation limits, which are of paramount importance to avoid flight instability or unpredictable behavior, thus ensuring the human's safety. Our approach uses simplified models of the collaborating human within the NMPC in a predictive fashion, while remaining real-time.

The paper is organized as follows. In Sec. II, we describe the mathematical models of the different components of our system. In Sec. III the optimal control problem formulation is presented, which is then followed by the validation of the proposed methodology in Sec. IV. We then present our conclusions and perspectives in Sec. V.

## II. Modeling

In this section, we provide the models for the two agents involved in the handover process [7], namely the *giver* (the AR) and the *receiver* (the human coworker).

### A. Aerial Robot Dynamics

The AR is a *Generically-Tilted Multi-Rotor* (GTMR) system [12], [11]. We model a GTMR as a rigid body of mass $m$ and inertia $\mathbf{J} \in \mathbb{R}^{3\times3}$. Besides, it is actuated by $n \geq 4$ motor-propellers, arbitrarily placed and oriented with respect to (w.r.t.) its main body.

As shown in Fig. 1, we define the world inertial frame $\mathcal{F}_W$, with its origin $O_W$ and its axes $\mathbf{x}_W, \mathbf{y}_W, \mathbf{z}_W$. All other frames are denoted using the same convention throughout this manuscript, e.g., $\mathcal{F}_B$ is the body frame of the robot. We denote with ${}^W\mathbf{p}_B \in \mathbb{R}^3$ the position of $O_B$ w.r.t. $\mathcal{F}_W$, and ${}^W\mathbf{q}_B \in \mathbb{R}^4$ the unit quaternion representing the rotation from $\mathcal{F}_B$ to $\mathcal{F}_W$. The same rotation can be expressed as the matrix ${}^W\mathbf{R}_B \in SO(3)$, or the roll-pitch-yaw angles vector ${}^W\boldsymbol{\eta}_B = [\phi\ \theta\ \psi]^\top$. We then indicate with ${}^W\mathbf{v}_B \in \mathbb{R}^3$ the linear speed of $O_B$ expressed in $\mathcal{F}_W$, and with ${}^B\boldsymbol{\omega}_B \in \mathbb{R}^3$ the angular velocity of $\mathcal{F}_B$ w.r.t. $\mathcal{F}_W$ specified in $\mathcal{F}_B$. Similar notations are used for all other frame pairs.

The robot state $\mathbf{x}_r$ is defined as

$$\mathbf{x}_r = \begin{bmatrix} \mathbf{p}_B{}^\top & \mathbf{q}_B{}^\top & \mathbf{v}_B{}^\top & {}^B\boldsymbol{\omega}_B{}^\top & \boldsymbol{\gamma}^\top \end{bmatrix}^\top \in \mathbb{R}^{13+n}. \quad (1)$$

In (1), and identically hereafter, we omit the reference frame notation for any vector expressed in $\mathcal{F}_W$. Similar to [12],

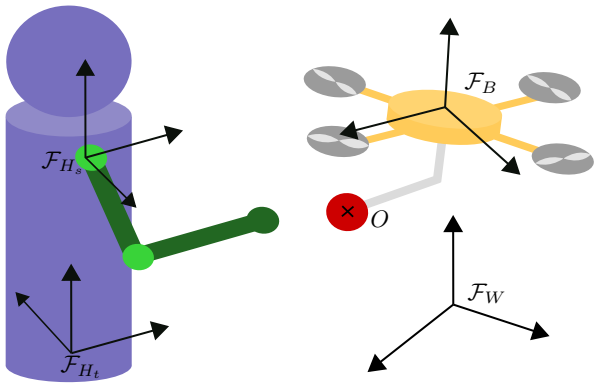


Fig. 1: Schematic depiction of the various frames during the handover of the red ball $O$ to the human coworker.

the vector $\boldsymbol{\gamma}$ collects the $n$ actuator forces. Accordingly, the actuation dynamics is given by

$$\dot{\boldsymbol{\gamma}} = \mathbf{u}_r, \quad (2)$$

where $\mathbf{u}_r \in \mathbb{R}^n$ are the control inputs of the robot, which can be related to the torques applied by the brushless motors on the propellers [12].

In order to account for the force and torque applied on the GTMR body by the weight of the carried object and the physical interaction, we integrate those in the dynamical equation of the GTMR, following the formalism of [13].

Hence, the dynamic model is given by (2) and

$$\dot{\mathbf{p}}_B = \mathbf{v}_B, \quad (3a)$$

$$\dot{\mathbf{q}}_B = \frac{1}{2}\mathbf{q}_B \otimes \begin{bmatrix} 0 \\ {}^B\boldsymbol{\omega}_B \end{bmatrix}, \quad (3b)$$

$$\begin{bmatrix} m\ddot{\mathbf{p}}_B \\ \mathbf{J}\,{}^B\dot{\boldsymbol{\omega}}_B \end{bmatrix} = \begin{bmatrix} -mg\mathbf{z}_W \\ -{}^B\boldsymbol{\omega}_B \times \mathbf{J}\,{}^B\boldsymbol{\omega}_B \end{bmatrix} + \begin{bmatrix} \mathbf{R}_B\mathbf{G}_\mathbf{f}\boldsymbol{\gamma} \\ \mathbf{G}_{\boldsymbol{\tau}}\boldsymbol{\gamma} \end{bmatrix} + \begin{bmatrix} \mathbf{R}_O & \mathbf{O}_3 \\ S({}^B\mathbf{p}_O)\,{}^B\mathbf{R}_O & {}^B\mathbf{R}_O \end{bmatrix} \begin{bmatrix} {}^O\mathbf{f}_O \\ {}^O\boldsymbol{\tau}_O \end{bmatrix}, \quad (3c)$$

where $\otimes$ denotes the Hamilton product of two quaternions, and $g$ is the intensity of the gravity acceleration, the *force* and *moment* allocation matrices, $\mathbf{G}_\mathbf{f}$ and $\mathbf{G}_{\boldsymbol{\tau}} \in \mathbb{R}^{3\times n}$, are mapping the thrusts generated by the robot's actuators to the forces and moments applied to the body, $\mathbf{O}_3$ is the 0 matrix of $\mathbb{R}^3$, $S(\cdot)$ is the skew operator associated with the cross product, and ${}^O\mathbf{f}_O$, ${}^O\boldsymbol{\tau}_O$ are the forces and torques applied on the object, including its weight.

### B. Sensor Model

The GTMR system features a sensor $S$ capable of retrieving the 3D-pose of an observed entity. As in [11], we model this sensor as a punctual device centered in $O_S$, having principal axis $\mathbf{z}_S$. In addition, the pose transformation between its frame $\mathcal{F}_S$ and the one of the multi-rotor body $\mathcal{F}_B$ is constant and known. Finally, the sensor's FoV is described with a pyramidal shape around the principal axis, having vertical and horizontal angles denoted with $\alpha_v \in \mathbb{R}^+$ and $\alpha_h \in \mathbb{R}^+$, respectively.

### C. Human Coworker Model

In this section, we first define the model used to describe the motion of the human coworker. Then, we introduce the formulation for her arm dynamics.

*1) Human Trunk:* Accordingly to Fig. 1, we define a *trunk* reference frame $\mathcal{F}_{H_t}$ at the middle point of the line connecting the human shoulders, with $\mathbf{x}_{H_t}$ being perpendicular to that line and pointing in the forward walking direction, and $\mathbf{z}_{H_t}$ parallel to $\mathbf{z}_W$. For the sake of simplicity it is assumed that the human maintains a standing position when walking, therefore the human roll and pitch angles are assumed constant and null.

The human state $\mathbf{x}_h$ is

$$\mathbf{x}_h = \begin{bmatrix} {\mathbf{p}_{H_t}}^\top & \psi_{H_t} \end{bmatrix}^\top \in \mathbb{R}^4. \tag{4}$$

Taking inspiration from [14], we adopt the following *constant-velocity* model for the human state, i.e.:

$$\dot{\mathbf{x}}_h = \begin{bmatrix} {\mathbf{v}_{H_t}}^\top & \Omega_\psi \end{bmatrix}^\top = \mathbf{u}_h. \tag{5}$$

where $\Omega_\psi$ is the angular speed about $\mathbf{z}_{H_t}$, while $\mathbf{u}_h$ denotes the human inputs. Contrary to the unicycle model in [14], the set of equations (5) allows accounting also for lateral movements of the human, which could naturally occur during a handover.

*2) Human Arm:* We model each human arm as a manipulator attached to one of the edges of the line connecting the shoulders, as depicted in Fig. 1. Therefore, for each upper limb, we define a *shoulder* reference frame denoted with $\mathcal{F}_{H_s}$ centered at the manipulator's base. The relative pose transformation between $\mathcal{F}_{H_s}$ and $\mathcal{F}_{H_t}$ is assumed to be known and constant. In particular, we take $\mathbf{x}_{H_s}$ and $\mathbf{y}_{H_s}$ to be coincident to $\mathbf{x}_{H_t}$ and $\mathbf{z}_{H_t}$, respectively. Each manipulator is composed of a serial chain of rigid links laying on the same plane and in pairs connected by a 1-revolute joint. As a result, the arm is modeled as an $n_q$*-link planar manipulator* with only revolute joints with parallel axes [15]. This assumption is motivated by the intuitive idea that a human would naturally move her arm by keeping it alongside her body during a handover task. Consequently, the human arm's workspace lays on the vertical $\mathbf{y}_{H_s}$-$\mathbf{x}_{H_s}$ (or equivalently, $\mathbf{z}_{H_t}$-$\mathbf{x}_{H_t}$) plane.

The dynamics of each upper limb is given by

$$\mathbf{B}_h(\mathbf{q}_h)\ddot{\mathbf{q}}_h + \mathbf{C}_h(\mathbf{q}_h, \dot{\mathbf{q}}_h)\dot{\mathbf{q}}_h + \mathbf{G}_h(\mathbf{q}_h) = \boldsymbol{\tau}_h + \mathbf{J}_h(\mathbf{q}_h)^\top \mathbf{f}_o, \tag{6}$$

where $\mathbf{B}_h \in \mathbb{R}^{m\times m}$ is the inertia matrix, $\mathbf{C}_h \in \mathbb{R}^{m\times m}$ accounts for the centrifugal and Coriolis terms, and $\mathbf{G}_h \in \mathbb{R}^m$ collects the gravitational effects. The vectors $\mathbf{q}_h$, $\dot{\mathbf{q}}_h$, $\ddot{\mathbf{q}}_h \in \mathbb{R}^m$ are the joint positions, velocities, and accelerations, respectively, while $\boldsymbol{\tau}_h \in \mathbb{R}^m$ gathers the joint torques. The matrix $\mathbf{J}_h(\mathbf{q}_h) \in \mathbb{R}^{m\times 3}$ is the Jacobian mapping the effect of an external force $\mathbf{f}_o \in \mathbb{R}^3$ applied on the human's hand to the arm dynamics.

## III. Optimal Control Problem Formulation

The aim of the proposed control framework is to successfully achieve an object handover to a human coworker by means of an aerial vehicle, while guaranteeing the human safety, and accounting for her ergonomics too. To achieve the goal, the controller has to: 1) execute a human-centered motion task that allows approaching the receiver; 2) guarantee the worker's safety, thus avoiding unwanted collisions while handovering the carried tool; 3) evaluate her articular stress to determine the most ergonomic handover location; 4) constantly observe the human, to prevent losing visibility of her ; and stabilize the robot dynamics by generating torque-level commands, which are compatible with its actuators' limitations. In the following, we detail the objective functions accounting for the individual tasks, the constraints applied on the system, and finally, in Sec. III-E, we formalize the corresponding Optimal Control Problem (OCP).

### A. Human-frame Motion

Aerial robots are usually requested to follow a list of waypoints specified w.r.t. an inertial reference frame. However, in tasks involving human-robot interaction, the robot should maintain a certain relative position and orientation, in conjunction with a precise velocity profile w.r.t. its human partner. Hence, to ensure that the aerial robot tracks a relative trajectory, we introduce a *motion task* expressed in the human-trunk frame $\mathcal{F}_{H_t}$ as part of the cost function.

Thus, we first derive the human-relative coordinates of the robot, expressed in $\mathcal{F}_{H_t}$, as

$${}^{H_t}\mathbf{p}_B = {\mathbf{R}_{H_t}}^\top(\mathbf{p}_B - \mathbf{p}_{H_t}), \quad {}^{H_t}\mathbf{R}_B = {\mathbf{R}_{H_t}}^\top \mathbf{R}_B, \tag{7}$$

In addition, by computing the derivatives of (7), it is possible to obtain the relative robot's linear and angular velocities, expressed in $\mathcal{F}_{H_t}$, as

$${}^{H_t}\mathbf{v}_B = {\mathbf{R}_{H_t}}^\top \left(\mathbf{v}_B - \mathbf{v}_{H_t} - S(\boldsymbol{\omega}_{H_t})\, \mathbf{R}_{H_t}\, {}^{H_t}\mathbf{p}_B\right), \tag{8a}$$

$$S({}^{H_t}\boldsymbol{\omega}_B) = {}^{H_t}{\mathbf{R}_B}^\top \Big[ \left(S({}^{H_t}\boldsymbol{\omega}_W){\mathbf{R}_{H_t}}^\top\right) \mathbf{R}_B + \\ + {\mathbf{R}_{H_t}}^\top \left(S(\boldsymbol{\omega}_B)\, \mathbf{R}_B\right) \Big], \tag{8b}$$

where $\mathbf{v}_B$, $\boldsymbol{\omega}_B$, and $\mathbf{v}_{H_t}$, $\boldsymbol{\omega}_{H_t}$ are the linear and angular velocities of the robot and the human trunk, respectively, and ${}^{H_t}\boldsymbol{\omega}_W$ is given by

$${}^{H_t}\boldsymbol{\omega}_W = {\mathbf{R}_{H_t}}^\top \left(-\boldsymbol{\omega}_{H_t}\right). \tag{9}$$

Finally, to account for the human motion (5) over the predictive horizon, we define the controller state $\mathbf{x}$ as

$$\mathbf{x} = \begin{bmatrix} {\mathbf{x}_r}^\top & {\mathbf{x}_h}^\top \end{bmatrix}^\top. \tag{10}$$

### B. Safety

During the whole operation and while handovering the object, the robot has to ensure the safety of the human coworker. Thus, in the cost function, we introduce a barrier function that precludes the robot from crossing a safety distance, which guarantees to avoid collisions between the two agents. This objective term has to strongly affect the robot behavior only when in the near proximities of the chosen distance and provide an almost null contribution anywhere else, to avoid disturbing other tasks. Therefore, we define the safety function $y_s$ as

$$y_s = \frac{\epsilon_m}{d_{h,r} - d_g}, \tag{11}$$

where $d_g$ is the minimum guard distance, and $d_{h,r}$ is the distance on the $x$, $y$ plan between the robot and $\mathbf{z}_{H_t}$. Finally, $\epsilon_m \in \mathbb{R}^+$ is a scaling factor to shape $y_s$ according to an additional margin.

### C. Ergonomics

The human-robot interaction has to occur in the most natural way possible, as instinctively arising during human-human interactions. In particular, the robot has to encourage and guide the human coworker to perform the handover in a comfortable position. Consequently, the robot has first to bring the object in a position that is both reachable and far enough, to appear natural and not to scare the human, respectively. Secondly, the robotic companion has to wait in the handover location for the human receiver to spontaneously take the object without forcibly requesting the interaction in a pre-defined reference position. Thirdly, the handover location should be such that holding the item does not require excessive effort or causes loss of postural stability, which becomes more and more pivotal as its weight increases [16].

Therefore, we propose to let the controller determine the best handover location based on the trade-off between minimizing the amount of effort required by the human receiver and maximizing the spontaneity of the handover process. In the following, we show how to relate the human joint torques that the human has to apply to hold the exchanged object at a given position. Then, we propose an approach to let the robot request the interaction at a comfortable and natural distance.

*1) Effort:* Similarly to [9], we relate the degree of ergonomics to the torques that the human needs to apply at her joints to hold the received object at a given location. However, we consider the handover task ending right before any physical engagement arises. Thus, we neglect any wrench exchange between the two agents while the object is passed, which constitutes an objective for future investigations. Furthermore, we assume that the task takes place as a quasi-static process such that the human slowly varies her joint variables to bring her hand in the proper position to receive the exchanged object, hence:

$$\ddot{\mathbf{q}}_h = \dot{\mathbf{q}}_h = 0. \tag{12}$$

By substituting (12) into (6), we can compute the torques the human applies on her arm to hold the received object $O$ in a given position as follows

$$\boldsymbol{\tau}_h = \mathbf{G}_h(\mathbf{q}_h) - \mathbf{J}_h(\mathbf{q}_h)^\top \mathbf{f}_o, \tag{13}$$

where $\mathbf{f}_o$ is the weight force induced by the object mass $m_o$.

Since we model the human arm as a manipulator, we rely on *inverse kinematics* to relate the Cartesian hand position to the corresponding joint variables. Consequently, if we denote with ${}^{H_s}\mathbf{p}_{H_h}$ the hand coordinates in the shoulder-frame $\mathcal{F}_{H_s}$, the following equation holds

$$\mathbf{q}_h = \boldsymbol{\Phi}_{\text{IK}}\left({}^{H_s}\mathbf{p}_{H_h}\right), \tag{14}$$

where $\boldsymbol{\Phi}_{\text{IK}}\left({}^{H_s}\mathbf{p}_{H_h}\right) : \mathbb{R}^3 \rightarrow \mathbb{R}^m$ denotes the inverse-kinematics function of the human arm. Since there may exist several solutions, we will later discuss the choice of a meaningful one.

The human hand and the exchanged object must be in the same position to successfully perform the handover. Therefore, substituting (14) into (13), and replacing ${}^{H_s}\mathbf{p}_{H_h}$ with the position of $O$ expressed in $\mathcal{F}_{H_s}$ (${}^{H_s}\mathbf{p}_O$), it results

$$\boldsymbol{\tau}_h = \mathbf{G}_h\left(\boldsymbol{\Phi}_{\text{IK}}\left({}^{H_s}\mathbf{p}_O\right)\right) - \mathbf{J}_h\left(\boldsymbol{\Phi}_{\text{IK}}\left({}^{H_s}\mathbf{p}_O\right)\right)^\top \mathbf{f}_o. \tag{15}$$

Using rototranslations similar to (7), ${}^{H_s}\mathbf{p}_O$ can be related to the pose of the robot pose in $\mathcal{F}_{H_t}$. Using Eq. (15), we can compute the torques the human has to apply at her joints to hold the object $O$ as a function of the robot position and orientation relative to her shoulder. Therefore, the NMPC can compute a handover pose reducing the human joint stress.

To embed (15) in an NMPC controller, we need to provide the solver with an expression for evaluating $\boldsymbol{\Phi}_{\text{IK}}$. In general, the inverse kinematics problem of a manipulator involves the solution of nonlinear equations, and it may have multiple, infinite, or no solution at all [15]. In the case of non-redundant manipulators with a small number of DoFs, it is possible to derive geometrical relationships that allow solving the problem analytically. Therefore, we decide to reduce the human arm to a simple 2-DoF planar manipulator ($n_q = 2$), for which closed-form results are available in textbooks, selecting only elbow-down configuration to comply with the human elbow articulation.

Finally, the existence of solutions for the inverse kinematics problem is guaranteed only if the given object position, ${}^{H_s}\mathbf{p}_O$, belongs to the human arm's workspace [15]. As a result, the NMPC controller cannot evaluate the human ergonomics until the robot gets close enough. To overcome this problem, in (15) it is considered the closest position to the robot that belongs to the human-arm workspace. This workspace is, for a 2-DoF serial manipulator, the space in-between two concentric co-planar circles [15], whose outer radius is given by the sum of the lengths of the links, and inner radius by their difference. Moreover, for the handover to take place in a comfortable and safe configuration, we consider only the front half of such region as human workspace.

Accordingly, we first project the current object position onto the manipulator's plane. Then, if the projected planar point is already part of the human arm's workspace, the inverse kinematics admits a feasible solution. Otherwise, we radially project the point onto the outer border of the workspace.

However, using the approach presented so far may not lead to have the object in the manipulator's plane, thus the controller is tasked to minimize also the normal projection of ${}^{H_s}\mathbf{p}_O$ to the $\mathbf{x}_{H_s}$-$\mathbf{y}_{H_s}$. This is achieved by introducing the following term in the controller

$$y_z = {}^{H_s}\mathbf{p}_{Oz} \in \mathbb{R}, \tag{16}$$

which represents the $z$ coordinate of ${}^{H_s}\mathbf{p}_O$.

Finally, to prevent the object to enter the inner circle, a constraint is imposed on the squared 2-norm of the object position in the manipulator's plane, denoted $c_{ho}$:

$$c_{ho} \geq (a_1 - a_2)^2, \tag{17}$$

where $a_1 \in \mathbb{R}^+$ and $a_2 \in \mathbb{R}^+$ are the link lengths of the human arm.

*2) Handover Distance:* By minimizing only the human joint torques, the robot would prefer to handover the object at the arm's resting configurations, since they constitute the two global minima of function (15). Clearly, these locations are incompatible with the objective of guaranteeing the worker's safety and achieving a natural interaction. Therefore, we introduce a second term to be optimize by the controller, which represents a primitive distance at which the handover could appear comfortable and natural, without neither scaring the human nor jeopardize her safety. Hence, we define an objective function $y_{ho}$ as

$$y_{ho} = \frac{b}{(d_{h,o} - d_{\text{ref}})}, \tag{18}$$

where $d_{h,o}$ is the planar distance between the object and the $\mathbf{y}_{H_s}$, $d_{\text{ref}}$ is a desirable distance, and $b \in \mathbb{R}^+$ is a scaling factor.

In order to let the handover occur at a reachable position and not compromise the operator's safety, we impose $d_g < d_{\text{ref}} < r$, $r$ being the length of the human arm.

### D. Additional Constraints

Based on previous works [12], [11], we impose additionnal constraints on the OCP to ensure the feasibility of the task.

First, maintaining the visibility of the human coworker is of paramount importance, since not knowing her position in the workspace would jeopardize safety of the task.

Thus, as shown in [11], we impose two constraints on the trunk position ${}^{S}\mathbf{p}_{H_t} = \begin{bmatrix} x_{H_t} & y_{H_t} & z_{H_t} \end{bmatrix}^\top$:

$$\left|x_{H_t}/z_{H_t}\right| \leq \tan\frac{\alpha_h}{2}, \quad \left|y_{H_t}/z_{H_t}\right| \leq \tan\frac{\alpha_v}{2}. \tag{19}$$

Moreover, to achieve robust tracking, we introduce a *visibility objective* as an additional term of the cost function of the OCP. As in [11], this quantity consists of the maximization of the cosine of the angular distance between ${}^{S}\mathbf{p}_{H_t}$ and $\mathbf{z}_S$, denoted $\mathrm{c}\beta$. As a result, the controller would maintain the human trunk close to the center of the sensor's FoV, while dealing with the other tasks. This allows a larger reactivity of the system w.r.t. the human motion, while avoiding as much as possible the configurations where the visibility constraints (19) might disturb the realization of other tasks.

Finally, to account for the physical limitations of the motor-propeller pairs (e.g., due to inertia and friction) [12], [11], we impose bounds on $\boldsymbol{\gamma}$ and $\dot{\boldsymbol{\gamma}}$ as

$$\underline{\boldsymbol{\gamma}} \leq \boldsymbol{\gamma} \leq \bar{\boldsymbol{\gamma}}, \tag{20a}$$

$$\underline{\dot{\boldsymbol{\gamma}}}(\boldsymbol{\gamma}) \leq \mathbf{u}_r \leq \bar{\dot{\boldsymbol{\gamma}}}(\boldsymbol{\gamma}), \tag{20b}$$

where the upper and lower bounds $\underline{\boldsymbol{\gamma}}$, $\bar{\boldsymbol{\gamma}}$, $\underline{\dot{\boldsymbol{\gamma}}}(\boldsymbol{\gamma})$, $\bar{\dot{\boldsymbol{\gamma}}}(\boldsymbol{\gamma})$ can be obtained through an identification campaign on the actual hardware, as detailed in [12].

### E. Optimal Control Problem

In this section, we formulate the discrete-time Optimal Control Problem (OCP), sampled in $N$ shooting points, which the controller solves at each sampling instant $t$, over the receding horizon $T$.

First, we define the output map $\mathbf{y}$ as

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_m{}^\top & \mathbf{y}_s{}^\top & \mathbf{y}_e{}^\top & \mathbf{y}_v{}^\top \end{bmatrix}^\top, \tag{21}$$

where $\mathbf{y}_m$, $\mathbf{y}_s$, $\mathbf{y}_e$, and $\mathbf{y}_v$ are the *motion*, *safety*, *ergonomics*, and *visibility* tasks, respectively. In turn, the individual objectives are given by

$$\mathbf{y}_m = \begin{bmatrix} {}^{H_t}\mathbf{p}_B{}^\top & {}^{H_t}\mathbf{q}_B{}^\top & {}^{H_t}\mathbf{v}_B{}^\top & {}^{H_t}\boldsymbol{\omega}_B{}^\top \end{bmatrix}^\top \in \mathbb{R}^{13}, \tag{22a}$$

$$\mathbf{y}_s = y_s \in \mathbb{R}, \tag{22b}$$

$$\mathbf{y}_v = 1 - \mathrm{c}\beta \in \mathbb{R}, \tag{22c}$$

$$\mathbf{y}_e = \begin{bmatrix} \boldsymbol{\tau}_h{}^\top & y_z & y_{ho} \end{bmatrix}^\top \in \mathbb{R}^4, \tag{22d}$$

whose reference values are denoted with $\mathbf{y}^r_\bullet$. The motion reference $\mathbf{y}^r_m$ is provided by an external trajectory planner, and $\mathbf{y}^r_s$, $\mathbf{y}^r_e$, and $\mathbf{y}^r_v$ are set to 0.

Consequently, we compute the cost function of the OCP as the summation of the invidivual task costs. Each term is given by the weighted square Euclidean norm of the difference between $\mathbf{y}_j$ and $\mathbf{y}^r_j$, denoted with $\|\bullet\|^2_{\mathbf{W}_j}$, where $\mathbf{W}_j$ a diagonal weighting matrix for the task $j$.

As a result, we can formulate the OCP as follows

$$\min_{\substack{\mathbf{x}_0 \dots \mathbf{x}_N \\ \mathbf{u}_0 \dots \mathbf{u}_{N-1}}} \sum_{k=0}^{N} \sum_{j}^{m,\dots,v} \|\mathbf{y}_{j,k} - \mathbf{y}^r_{j,k}\|^2_{\mathbf{W}_j} + \sum_{k=0}^{N-1} \|\mathbf{u}_r\|^2_{\mathbf{W}_{\mathbf{u}_r}} \tag{23a}$$

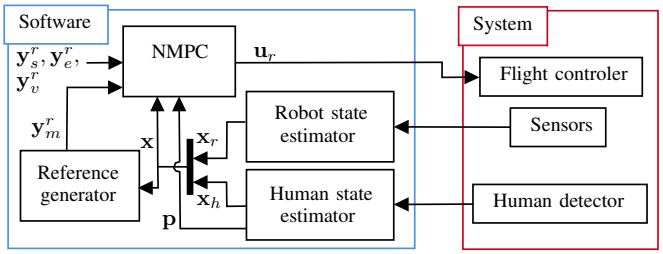


Fig. 2: Block diagram of the framework.

$$
\begin{aligned}
s.t. \quad & \mathbf{x}_0 = \mathbf{x}(t), && && \text{(23b)}\\
& \mathbf{x}_{k+1} = \mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}_k\right), && k\in\{0,\dots,N-1\} && \text{(23c)}\\
& \mathbf{y}_k = \mathbf{h}\left(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}_k\right), && k\in\{0,\dots,N\} && \text{(23d)}\\
& \underline{\boldsymbol{\gamma}}_k \leq \boldsymbol{\gamma}_k \leq \bar{\boldsymbol{\gamma}}_k, && k\in\{0,\dots,N\} && \text{(23e)}\\
& \dot{\underline{\boldsymbol{\gamma}}}_k \leq \mathbf{u}_k \leq \dot{\bar{\boldsymbol{\gamma}}}_k, && k\in\{0,\dots,N-1\} && \text{(23f)}\\
& c_{ho,k} \geq (a_1 - a_2)^2 && k\in\{0,\dots,N\} && \text{(23g)}\\
& \left|x_{H_t}/z_{H_t}\right|_k \leq \tan\frac{\alpha_h}{2}, && k\in\{0,\dots,N\} && \text{(23h)}\\
& \left|y_{H_t}/z_{H_t}\right|_k \leq \tan\frac{\alpha_v}{2}, && k\in\{0,\dots,N\} && \text{(23i)}
\end{aligned}
$$

where $\mathbf{x}(t)$ is the measurement of the current state, and $\mathbf{f}$ synthetically denotes the system dynamics, expressed by (2), (3), and (5), and $\mathbf{p}$ contains the external parameters provided to the controller at each control cycle $t$, namely the human velocities $\mathbf{u}_h$.

## IV. Validation

### A. Motion Reference Generation

The motion reference trajectory $\mathbf{y}_m^r$, needs to be defined in order to drive the AR in front of the human. Such trajectory is generated by a standard motion planner employing a spline interpolation to connect a set of intermediary waypoints. The initial point of this trajectory is the starting position of the AR, while the final destination is to arrive in front of the human. Moreover, the robot has to maintain a suitable distance from the human, which is not too close to scare her and also not too far in order to make clear its intent of approaching. This distance defines the radius of the navigation circle. Once this distance is reached, the robot will move towards the human's trunk, performing a circular path. Once in sight from the human, it will start approaching more closely, until reaching a desired distance in front of the human. At this point the controller switches to the reaching phase.

### B. Simulation Setup

This section depicts the simulation of a human-robot handover, with the AR starting from a random location behind the human coworker. It is achieved using a collinear quadrotor, which performs the handover of a small object, whose weight is set to 250g, picturing a small tool. The object is attached on a stick in front of the UAV, shifted by $45°$ w.r.t. the arms, both to have the object further from the propellers, and to allow easier compensation of the torque induced by the object on the quadrotor body. The simulated

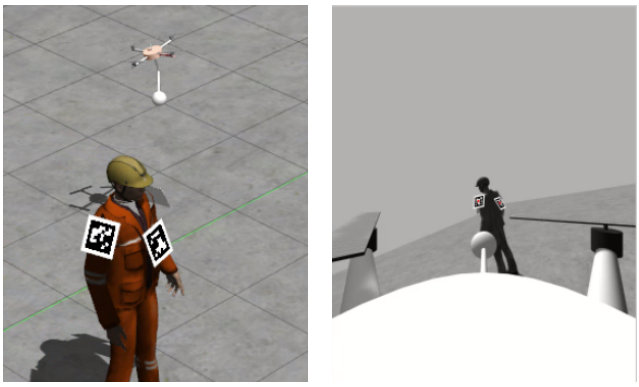

Fig. 3: On the left, a snapshot of the simulated platform performing the approaching phase presented in Sec. IV-C. On the right, a frame of the robot's onboard camera, taken during this motion.

human coworker is controlled by a joystick, both for her planar position and yaw, and for the motion of the arm. A picture of the simulated system in the handover task is shown on the left of Fig. 3.

The framework is implemented in C++, using GenoM [17] which is a middleware-independent component generator, that can be compiled for a given middleware, e.g., ROS. The reference generation is implemented in Matlab. The NMPC implementation is the one introduced in [11], based on [18]. It uses a $4^{\text{th}}$ order explicit Runge-Kutta integrator and implements a RealTime Iteration solving scheme, allowing high frequency control. The block diagram of the framework is drawn in 2. The simulated hardware interface as well as the state estimation and path planning are done using the TeleKyb3 software, available on the OpenRobots platform[1].

The software framework is connected to a Gazebo simulated system that emulates the actual platform interface, whose inputs are the rotor velocities. Details on how to use this software can be found in the provided git repository[2].

The state estimation of the AR is achieved using simulated motion capture (MoCap) and IMU, whose respective frequencies are 50Hz and 500Hz. The rotor positions are retrieved at 100Hz. A Gaussian noise is applied on each of these simulated measurements, with respective standard deviations of 0.003m, 0.02rad/s and 0.1m/s$^2$, and 0.03rad.

The simulated AR are equipped with a front-facing monocular camera, having a sampling frequency of 60Hz. The simulated human is equipped with a set of AruCo fiducial markers [19]. These markers are used to retrieve the position of the human trunk in world frame, $\mathbf{p}_{H_t}$, during the reaching phase. This choice is motivated by the practicality of such markers, and on the low computational time and power required for the detection process. This allows providing $\mathbf{p}_{H_t}$ effectively at 60Hz to the filtering algorithm. Recent developments of machine learning algorithms allow embedding fast, computationally efficient, and reliable object detection solutions on board ARs, e.g. [20]. The use of such algorithm would relieve the human coworker from

[1] https://git.openrobots.org/projects/telekyb3
[2] https://redmine.laas.fr/projects/nmpc-handover

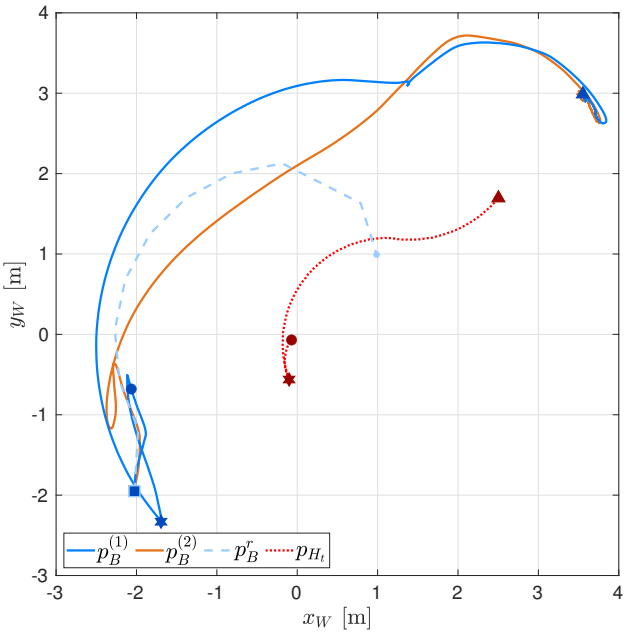


Fig. 4: Top view of the Approaching phase. In light-blue, the initial reference motion task generated by the motion planner. In blue and orange, the motion of the robot in two simulations, (1) and (2), and in red the trajectory of the human.

her markers. However, such algorithm are usually trained on specific datasets, and might not provide the desired pose estimate in a handover configuration where the AR is standing very close to the human. The integration of such tool is promising, but is left out of the scope of this work.

We also neglect the weight of the carried tool, as well as the physical contact with the human. Accounting for those require the use of, e.g., a wrench-observer to estimate the last term in (3c). These aspects are left for future work, as this manuscript focuses on the safety and ergonomics-awareness in the approach and reaching phases.

This simulation is composed of two phases. First, the robot performs an *Approaching* phase where, from the initial position, it moves in front of the human, using the algorithm of Sec. IV-A. Later, in the *Reaching* phase, it narrows the distance to the human coworker to perform the *Handover*. We first present the greater reactivity allowed by considering the motion in $\mathcal{F}_{H_t}$, in particular during the approaching phase. Then, the effects of enabling the ergonomic costs are demonstrated during the reaching phase, which lead the AR to actually perform the handover.

Videos of the reported simulation can be found in the attached multimedia file.

### C. Approaching

The $(x, y)$ motions during the approaching phase are depicted in Fig. 4. The blue and orange curves correspond respectively to simulations with and without the prediction of the human motion in the controller (i.e., $\mathbf{u}_h = 0$). The initial AR position is marked as the blue square, while the initial position of the human as the red circle.

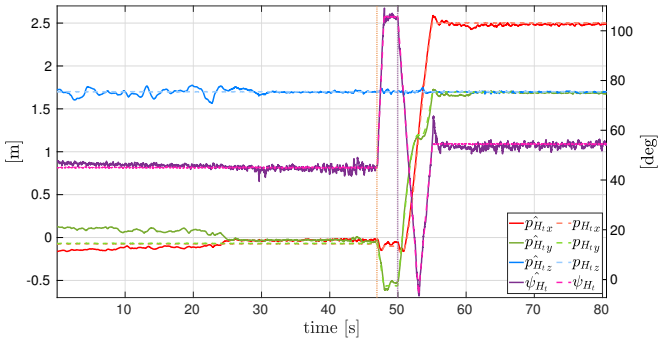


Fig. 5: Estimated and ground-truth values of the human position and yaw.

We first present the simulation depicted by the blue curve. The human moves from the red circle when the AR is at the blue circle, toward the location denoted with a red star. Similarly, when the human moves to the final position marked with a red triangle, the robot is at the blue star. Finally, the robot reaches the final position denoted with a blue triangle.

As a result, since the trajectory is specified w.r.t. $\mathcal{F}_{H_t}$, the controller modifies the motion of the robot accordingly.

As shown, this scheme allows positioning the AR in front of the human for the approaching phase regardless of the unknown human motion, without the need of an online re-planning. In addition to this, the integration of the human state in the NMPC, allows a better reactivity, since the controller can propagate it through the horizon to predict her future poses.

The orange curve shows a replica of the previous simulation with the same human motion. In this case, the motion of the AR reflects with less fidelity the original planned path. Moreover, the distance between the robot and the human is shorter, which could induce safety hazards.

Finally, in Fig. 5, we show the quality of the onboard estimation of the human position and yaw. The dashed lines are the ground-truth values, while the solid ones the output of the onboard estimates. The orange line corresponds to the first displacement of the human shown in the previous figure, while the purple one to her second motion. In general, the human position is well estimated, except for the fist part of the simulation, where the distance toward the human is large, increasing the difficulty of detecting precisely the AruCo markers. During the phase in which the coworker moves, it can be noticed that the estimation is instead less affected by noise. The reason can be appreciated in Fig. 3, where a frame of the robot's onboard camera is shown. That image has been taken while the robot is navigating around the coworker to reach the final position. In that moment, the camera is observing two AruCo markers, which provide more measurements to better estimate the human pose.

### D. Handover

When the AR is in position, i.e., in front of the operator, the handover phase is initiated. The ergonomic objectives are enabled and the motion task is disabled, as presented in III-C.

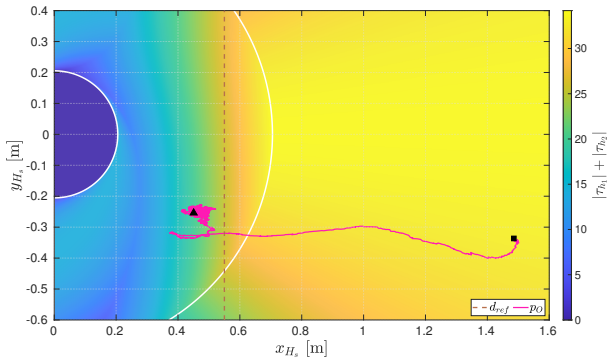


Fig. 6: Side view of the human arm's plane. The color gradient shows the sum of joint torques. $p_O$ is the projection of $O$ over the reaching phase, from the square to the triangle.

Figure 6 shows the path of the AR and the object in the plane of the human shoulder. The color gradient clearly shows how the object is moved toward the hand, while staying in the region that minimizes the sum of the absolute values of the human joint torques. The absolute minimum, corresponding to the rest configuration of the arm, is not reached due to the trade-off with the visibility and the desired handover tasks. Indeed, moving towards that location would jeopardize the detection of the human trunk, and result unnatural for the human coworker. Instead, the controller drive the robot to another location, where it will wait for the human to grab the delivered object. In this way, the control method will prevent the robot from forcibly pushing the object into her hands and, at the same time, it will guarantee a good level of ergonomics.

Lastly, in Fig. 7, we report the visibility task over the approaching phase. As the plot suggests, the controller can maintain the human trunk inside the camera's FoV during the whole simulation, and close to the center ($\mathrm{c}\beta = 1$). Large deviations from the reference value are noticeable when the human moves, and in the last portion of the plot, where the robot has to stop in the final position. Similarly, the actuation constraints are respected during the whole handover (not displayed here for space limitation).

## V. Conclusions

In this work, we propose an ergonomics-aware NMPC designed to autonomously perform the handover of an object with a human coworker. The formulation considers the closed-form equations of the shoulder and elbow torques of the human to determine handover position, in order to minimize the effort stress imposed during the operation. We build upon previous works to ensure that the actuation limitations of the system are strictly observed during the motion, while maintaining the human in the FoV of an onboard camera, used to estimate the human-AR relative pose. Besides, the controller motion is computed relative to the human, in order to increase the reactivity of the framework with any unexpected human motion, without the need of a constant online replanning. The human motion model is included in the NMPC equations, which allows the

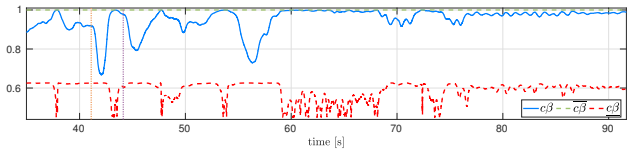


Fig. 7: Visibility constraint over time during the experiment, where $\underline{\mathrm{c}\beta}$ synthetically summarizes the FoV constraints (19) in a 1D representation.

controller to predict her future poses through the horizon. Additionally, the relative formulation allows embedding a safety barrier to avoid collisions with the human, which is of paramount importance during any human-robot interaction. Finally, the framework is tested in a Gazebo simulation. The controller tracks the desired path and bring the robot at the correct position for performing the handover.

Despite its richness, the presented work still leaves some open challenges. Firstly, the physical interaction between the two agents is neglected, while a wrench will arise during the exchange of the object, by pulling or pushing the object before being released. Therefore, the controller should compute motor commands to compensate for the human actions and, at the same time, prevent the risk of losing stability and consequently impacting the partner. Secondly, the model used for the human arm considers only the dynamics of the shoulder and the elbow, and it neglects the transfer of interaction to the torso. Therefore, it is possible to enlarge this model to address both the wrist and the shoulder joints, and take into consideration the trunk posture. Accordingly, this would imply relating the ergonomics not only to the torques of the arm, but also to the wrench transmitted to the trunk, causing a shift in the body configuration of the human. Thirdly, the use of higher DoF models makes unfeasible finding a closed-form solution to the inverse kinematics problem. Consequently, the use of optimization to solve the inverse-kinematics problem could be investigated, e.g., the mapping of the joint torques to the object pose could be directly embedded inside the OCP.

## References


[1] P. Petracek, V. Kratky, and M. Saska, "Dronument: System for reliable deployment of micro aerial vehicles in dark areas of large historical monuments," *IEEE Robotics and Automation Letters*, vol. 5, pp. 2078–2085, 2020.

[2] A. Ollero, M. Tognon, A. Suarez, D. J. Lee, and A. Franchi, "Past, present, and future of aerial robotic manipulators," *IEEE Trans. on Robotics*, 2021.

[3] J. L. Paneque, J. R. M. Dios, A. Ollero, D. Hanover, S. Sun, A. Romero, and D. Scaramuzza, "Perception-aware perching on powerlines with multirotors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3077–3084, 2022.

[4] M. Gienger, D. Ruiken, T. Bates, M. Regaieg, M. MeiBner, J. Kober, P. Seiwald, and A.-C. Hildebrandt, "Human-robot cooperative object manipulation with contact changes," in *2018 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2018, pp. 1354–1360.

[5] S. S. Mirrazavi Salehian, N. Figueroa, and A. Billard, "A unified framework for coordinated multi-arm motion planning," *The International Journal of Robotics Research*, vol. 37, no. 10, pp. 1205–1232, 2018.

[6] K. Strabala, M. K. Lee, A. Dragan, J. Forlizzi, S. S. Srinivasa, M. Cakmak, and V. Micelli, "Toward seamless human-robot handovers," *Journal of Human-Robot Interaction*, vol. 2, no. 1, pp. 112–132, 2013.

[7] J. R. Medina, F. Duvallet, M. Karnam, and A. Billard, "A human-inspired controller for fluid human-robot handovers," in *2016 IEEE Int. Conf. on Humanoid Robots*, Cancun, Mexico, Nov. 2016, pp. 324–331.

[8] E. A. Sisbot and R. Alami, "A human-aware manipulation planner," *IEEE Trans. on Robotics*, vol. 28, no. 5, pp. 1045–1057, 2012.

[9] L. Peternel, W. Kim, J. Babic, and A. Ajoudani, "Towards ergonomic control of human-robot co-manipulation and handover," in *2017 IEEE Int. Conf. on Humanoid Robots*, Birmingham, Nov. 2017, pp. 55–60.

[10] V. Ortenzi, A. Cosgun, T. Pardi, W. P. Chan, E. Croft, and D. Kuli, "Object handovers: A review for robotics," *IEEE Trans. on Robotics*, vol. 37, no. 6, pp. 1855–1873, 2021.

[11] M. Jacquet and A. Franchi, "Motor and perception constrained NMPC for torque-controlled generic aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 518–525, 2021.

[12] D. Bicego, J. Mazzetto, M. Farina, R. Carli, and A. Franchi, "Nonlinear model predictive control with enhanced actuator model for multi-rotor aerial vehicles with generic designs," *Journal of Intelligent & Robotics Systems*, vol. 100, pp. 1213–1247, 2020.

[13] G. Corsini, M. Jacquet, A. E. Jimenez-Cano, A. Afifi, D. Sidobre, and A. Franchi, "A general control architecture for visual servoing and physical interaction tasks for fully-actuated aerial vehicles," in *2021 Aerial Robotic Systems Physically Interacting with the Environment (AIRPHARO)*. IEEE, 2021, pp. 1–8.

[14] G. Arechavaleta, J.-P. Laumond, H. Hicheur, and A. Berthoz, "On the nonholonomic nature of human locomotion," *Autonomous Robots*, vol. 25, no. 1-2, pp. 25–35, 2008.

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

[16] K. Harada, S. Kajita, H. Saito, M. Morisawa, F.Kanehiro, K. Fujiwara, K. Kaneko, and H. Hirukawa, "A humanoid robot carrying a heavy object," in *2005 IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 1712–1717.

[17] A. Mallet, C. Pasteur, M. Herrb, S. Lemaignan, and F. Ingrand, "GenoM3: Building middleware-independent robotic components," in *2010 IEEE Int. Conf. on Robotics and Automation*, 2010, pp. 4627–4632.

[18] Y. Chen, M. Bruschetta, E. Picotti, and A. Beghi, "MATMPC - a MATLAB based toolbox for real-time nonlinear model predictive control," in *2019 European Control Conference*, Jun. 2019, pp. 3365–3370.

[19] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[20] P. Zhang, Y. Zhong, and X. Li, "Slimyolov3: Narrower, faster and better for real-time uav applications," in *2020 IEEE/CVF Int. Conf. on Computer Vision*, 2019.