



HAL
open science

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

► **To cite this version:**

Gianluca Corsini, Martin Jacquet, Hemjyoti Das, Amr Afifi, Daniel Sidobre, et al.. Nonlinear Model Predictive Control for Human-Robot Handover with Application to the Aerial Case. The 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2022), Oct 2022, Kyoto, Japan. 10.1109/IROS47612.2022.9981045 . hal-03716664v2

HAL Id: hal-03716664

<https://hal.science/hal-03716664v2>

Submitted on 20 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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

Gianluca Corsini¹, Martin Jacquet¹, Hemjyoti Das², Amr Afifi², Daniel Sidobre¹, 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 (AR). 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 multi-rotor aerial vehicle (MRAV) with a minimal adaptation of the robot model. The formulation of the optimal control problem steers the AR toward a handover location by optimizing the human coworker ergonomics, which includes the predicted arm 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 promptly 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 of the robot. Additionally, a safety barrier is embedded in the controller to avoid any risk of collision with the human partner. Realistic simulations are performed to validate the feasibility of the approach and the source code of the implementation is released open-source.

I. INTRODUCTION

Aerial robots (ARs) and more specifically multi-rotor aerial vehicles (MRAVs) 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 of the onboard sensing equipment, and the availability of heterogeneous designs. Consequently, they 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 wind turbines, large construction sites, or power transmission lines [3]. These settings usually require specialized and trained personnel employing expensive equipment and special vehicles. Carrying and accessing different tools in these challenging circumstances would require cumbersome postures and a loss of focus from the current activity. The use of aerial vehicles

as robotic coworkers, in these cases, can facilitate the tasks carried out by the operators. In those scenarios, an AR 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 ARs and particularly multi-rotors to reduce the physical and cognitive load experienced by workers at height. Nevertheless, to achieve the aforementioned benefits, aerial robots should explicitly take into account human ergonomics and safety.

Despite the clear potential, the use of MRAVs 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 [4]. Recent works have investigated the use of manipulators to assist a human coworker in the manipulation of heavy and bulky objects or during assembly tasks [5], [6]. Similarly, the problem of object handover has been extensively studied in the literature [7].

In general, an object handover is characterized by several phases [8], namely the *approach*, *reach*, and *transfer* phases. Most works treat each phase separately, with some notable exceptions. In [9], the authors propose a control architecture for fluid handovers, that tackles all the phases 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. However, the proposed control scheme does not explicitly include safety and ergonomics, which are of paramount importance in a control framework that is designed to enable the collaboration between aerial robots and humans, in particular at height.

The idea of including human comfort and ergonomics in the robot control and planning software has been also treated in the literature. One of the earlier works in this area is [10], which develops a manipulation planner that takes into account various human aspects, as ergonomics and field of view, amongst others. The methods of including human ergonomics in robot controllers evolved further. For instance, in [11], the authors propose an approach to compute the joint torques of the human based on a whole-body dynamic model and then control a ground mobile manipulator to minimize the overloading of the human joints.

As identified in [7], an important aspect of human-robot collaboration, including object handovers, is human perception. Most of the works that focus on control and planning aspects usually rely on the possibility of perceiving the human subject through sensors, like motion capture systems,

¹LAAS-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

²Robotics 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).

wearable inertial suits, or cameras. This motivates why a tight coupling between control and perception is critical for the success of such tasks, as the loss of visibility over the coworker would jeopardize the maneuver. Perception-constrained control has been previously explored by some of the authors of this paper. In prior work [12], a perception-constrained motor-level Nonlinear Model Predictive Control (NMPC) for generic aerial vehicles is presented to keep specific features in the field of view (FoV) of the camera while respecting the system actuation limits.

Pushing beyond the achievements of [12], this work presents an NMPC framework that can also handle multiple aspects of HRI, with a special focus on the aerial case. We introduce an NMPC formulation that includes human ergonomics and comfort as an objective, while enforcing perception and actuation limits, which are essential to avoid flight instability or unpredictable behavior, thus ensuring human safety. Moreover, our approach uses a model for the human coworker within the NMPC to predict her future poses, while remaining real-time.

The paper is organized as follows. In Sec. II, we provide the mathematical description of the system. The optimal control problem (OCP) formulation is presented in Sec. III, later followed by the methodology validation 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 [9], 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 [13], [12]. We model a GTMR as a rigid body of mass $m \in \mathbb{R}^+$ and inertia $\mathbf{J} \in \mathbb{R}^{3 \times 3}$. Besides, it is actuated by $n_\gamma \geq 4$ motor-propellers actuators, 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 by ${}^W \mathbf{p}_B \in \mathbb{R}^3$ 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

$$\mathbf{x}_r = \left[\mathbf{p}_B^\top \quad \mathbf{q}_B^\top \quad \mathbf{v}_B^\top \quad {}^B \boldsymbol{\omega}_B^\top \quad \boldsymbol{\gamma}^\top \right]^\top \in \mathbb{R}^{13+n_\gamma}. \quad (1)$$

In (1), and identically hereafter, we omit the reference frame notation for any vector expressed in \mathcal{F}_W . Similar to [13], the vector $\boldsymbol{\gamma}$ collects the n_γ actuator forces. Accordingly,

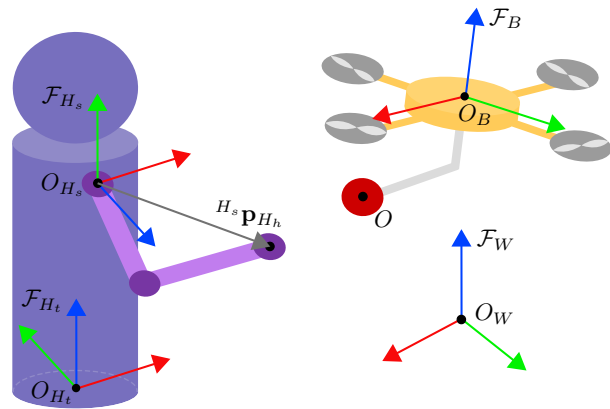


Fig. 1: Schematic depiction of the various frames during the handover of the object O (red sphere) to the human coworker. Frame convention is red, green, and blue for x , y , and z . As an example of AR, a collinear quadrotor is drawn.

the actuation dynamics is given by

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

where $\mathbf{u}_r \in \mathbb{R}^{n_\gamma}$ are the control inputs of the robot, which can be related to the torques applied to the propellers by the brushless motors [13].

Following the formalism of [14], in the dynamical equation of the GTMR, we integrate also the forces and torques applied to the robot body by the weight of the carried object and the physical interaction.

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

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

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

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

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}_f and $\mathbf{G}_\tau \in \mathbb{R}^{3 \times n_\gamma}$, are mapping the thrusts generated by the robot 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 ${}^O \mathbf{f}_O, {}^O \boldsymbol{\tau}_O \in \mathbb{R}^3$ are the forces and torques applied to 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 [12], 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 FoV of the sensor is

described with a pyramidal shape around the principal axis, having vertical and horizontal angles denoted by $\alpha_v \in \mathbb{R}^+$ and $\alpha_h \in \mathbb{R}^+$, respectively.

C. Human Coworker Model

In this section, we show the model used for describing the motion of the human trunk and the arm dynamics.

1) *Human Trunk*: Accordingly to Fig. 1, we define a *trunk* reference frame \mathcal{F}_{H_t} having \mathbf{y}_{H_t} coincident to the line connecting the human shoulders, \mathbf{x}_{H_t} pointing in the forward walking direction, and \mathbf{z}_{H_t} parallel to \mathbf{z}_W . For the sake of simplicity, we assume that the human maintains a standing position while walking; therefore, the human roll and pitch angles are set constant and null.

The human state \mathbf{x}_h is

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

Taking inspiration from [15], we adopt the following *constant-velocity* model for the human state, i.e.

$$\dot{\mathbf{x}}_h = \begin{bmatrix} \mathbf{v}_{H_t}^\top & \Omega_\psi \end{bmatrix}^\top = \mathbf{u}_h, \quad (5)$$

where Ω_ψ is the angular speed about \mathbf{z}_{H_t} , and \mathbf{u}_h denotes the human inputs. Contrary to the unicycle model in [15], the set of equations (5) allows accounting also for lateral movements of the human, which may occur during a handover.

2) *Human Arm*: We model the human arm involved in the handover as a manipulator whose base is connected at the shoulder attaching point, as depicted in Fig. 1. Therefore, we define a *shoulder* reference frame denoted by \mathcal{F}_{H_s} centered at the manipulator 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 parallel to \mathbf{x}_{H_t} and \mathbf{z}_{H_t} , respectively. This 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 [16]. This assumption is motivated by the intuition that a human would naturally move the arm alongside the body during the handover. Consequently, the human arm workspace lays on the vertical plane ($\mathbf{x}_{H_s}, \mathbf{y}_{H_s}$), or equivalently ($\mathbf{x}_{H_t}, \mathbf{z}_{H_t}$).

The dynamics of the upper limb is given by

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

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

III. OPTIMAL CONTROL PROBLEM FORMULATION

The proposed control framework aims to successfully achieve an object handover to a human coworker by means of an aerial vehicle while guaranteeing human safety and accounting for her ergonomics. To achieve the goal, the controller has to: 1) execute a human-centered motion task that allows approaching the receiver; 2) guarantee the safety of the worker, thus avoiding unwanted collisions while handing over the carried tool; 3) evaluate her articular stress to determine the most ergonomic handover location; 4) constantly observe the human, to maintain visibility on her; and 5) stabilize the robot dynamics by generating torque-level commands, which are compatible with its actuation limitations. In the following, we detail the objective functions accounting for the individual tasks, the constraints applied to the system, and finally, in Sec. III-E, we formalize the corresponding OCP.

A. Human-frame Motion

ARs are usually requested to follow a list of waypoints specified w.r.t. an inertial reference frame. However, in HRI scenarios, 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

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

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

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

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

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

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

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

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

B. Safety

During the whole operation and while handing over 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 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

$$y_s = \frac{\epsilon_m}{d_{h,r} - d_g}, \quad (11)$$

where $d_g \in \mathbb{R}^+$ is a minimum guard distance, and $d_{h,r}$ is the relative distance between the robot and the human on the plane $(\mathbf{x}_{H_t}, \mathbf{y}_{H_t})$. 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 and comfortable way. 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. Secondly, the human should not exert excessive physical exertion at the joints to hold the object in the handover location.

Therefore, in our proposed approach, the controller computes the best handover location based on the trade-off between minimizing the human effort and maximizing the spontaneity of the interaction. In the following, we show how to evaluate the human joint torques that the human has to apply to hold the exchanged object in a given position. Then, we propose an approach to make the robot request the interaction at a comfortable and natural distance.

1) *Effort*: Similarly to [11], we relate the degree of ergonomics to the torques the human needs to apply at the joints to hold the received object at a given location. Then, we assume that the task takes place as a quasi-static process, given that the human slowly moves the hand in the proper position to receive the exchanged object, hence:

$$\ddot{\mathbf{q}}_h = \dot{\mathbf{q}}_h = 0. \quad (12)$$

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

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

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

We then rely on *inverse kinematics* to relate the Cartesian hand position to the corresponding joint variables. Consequently, if we denote by ${}^{H_s}\mathbf{p}_{H_h}$ the hand coordinates in the shoulder frame \mathcal{F}_{H_s} , the following equation holds

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

where $\Phi_{\text{IK}}\left({}^{H_s}\mathbf{p}_{H_h}\right) : \mathbb{R}^3 \rightarrow \mathbb{R}^{n_q}$ denotes the inverse-kinematics function of the human arm.

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

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

Through rototranslations similar to (7), ${}^{H_s}\mathbf{p}_O$ can be related to the robot pose in \mathcal{F}_{H_t} . Using (15), we can compute the torques the human has to apply at the joints to hold the object O as a function of the robot position and orientation relative to the 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 Φ_{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 [16]. In the case of non-redundant manipulators with a small number of degrees of freedom (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-DoFs 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.

Moreover, 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 workspace [16]. For a 2-DoFs planar manipulator, the workspace is the space in-between two concentric co-planar circles [16], whose outer radius is equal to the sum of the link lengths, and the inner radius to their difference. As the handover shall take place in a comfortable and safe configuration, we consider only the front half of such a region as the human workspace.

As a result, the NMPC controller cannot evaluate the human ergonomics until the robot gets close enough. To overcome this problem, in (15), we consider the closest position to the robot that belongs to the human arm workspace. Accordingly, we first project the current object position onto the arm plane. Then, if the projected point is already part of the human arm workspace, the inverse kinematics admits a feasible solution. Otherwise, we radially project it onto the outer border of the workspace.

Besides, to assure having the object in the human arm plane, the controller is tasked to minimize the normal projection of ${}^{H_s}\mathbf{p}_O$ to the plane $(\mathbf{x}_{H_s}, \mathbf{y}_{H_s})$. This is achieved by introducing the following term in the control task

$$y_z = {}^{H_s}\mathbf{p}_O z \in \mathbb{R}, \quad (16)$$

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 human arm plane, denoted by c_{ho} :

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

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

2) *Handover Distance*: By minimizing only the human joint torques, the robot would prefer to hand over the object at the arm resting configurations, since they constitute the two global minima of (15). These locations are incompatible with the objective of achieving a safe and natural interaction. Therefore, a second term is optimized by the controller which represents a primitive distance at which the handover appears comfortable and natural, without neither scaring

nor endangering the human. Hence, we define an objective function y_{ho} as

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

where $d_{h,o}$ is the relative distance between the object and the human on the plane $(\mathbf{x}_{H_s}, \mathbf{z}_{H_s})$, $d_{ref} \in \mathbb{R}^+$ is a desirable distance, and $b \in \mathbb{R}^+$ is a scaling factor.

Lastly, we impose $d_g < d_{ref} < (a_1 + a_2)$ to assure the handover at a reachable position and the operator safety.

D. Additional Constraints

Based on previous works [13], [12], we impose additional constraints on the OCP to ensure the task feasibility.

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

Thus, as shown in [12], we impose two constraints on the trunk position, expressed in \mathcal{F}_S , ${}^S\mathbf{p}_{H_t} = [x_{H_t} \ y_{H_t} \ z_{H_t}]^\top$:

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

Moreover, to achieve robust tracking, we introduce a *visibility objective* as an additional term of the cost function in the OCP. As in [12], this quantity consists in maximizing the cosine of the angular distance between ${}^S\mathbf{p}_{H_t}$ and \mathbf{z}_S , denoted by $c\beta$. As a result, the controller would maintain the human trunk close to the center of the FoV of the sensor, while dealing with the other tasks. This enables a larger reactivity of the system w.r.t. the human motion, while avoiding the configurations where the visibility constraints (19) might interfere with the realization of other tasks.

Finally, to account for the physical limitations of the motor-propeller actuators (e.g. due to inertia and friction) [13], [12], we impose bounds on γ and $\dot{\gamma}$ as

$$\underline{\gamma} \leq \gamma \leq \bar{\gamma}, \quad (20a)$$

$$\dot{\underline{\gamma}}(\gamma) \leq \dot{\gamma} \leq \dot{\bar{\gamma}}(\gamma), \quad (20b)$$

where the upper and lower bounds $\underline{\gamma}$, $\bar{\gamma}$, $\dot{\underline{\gamma}}(\gamma)$, $\dot{\bar{\gamma}}(\gamma)$ are obtained through an identification campaign on the actual hardware, as detailed in [13].

E. Optimal Control Problem

In this section, we formulate the discrete-time 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

$$\mathbf{y} = [\mathbf{y}_m^\top \ \mathbf{y}_s^\top \ \mathbf{y}_e^\top \ \mathbf{y}_v^\top]^\top, \quad (21)$$

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 ob-

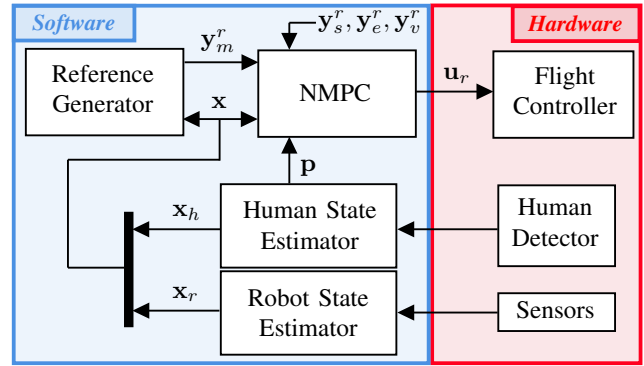


Fig. 2: Block diagram of the control framework.

jectives are given by

$$\mathbf{y}_m = [H_t \mathbf{p}_B^\top \ H_t \mathbf{q}_B^\top \ H_t \mathbf{v}_B^\top \ H_t \boldsymbol{\omega}_B^\top]^\top \in \mathbb{R}^{13}, \quad (22a)$$

$$\mathbf{y}_s = y_s \in \mathbb{R}, \quad (22b)$$

$$\mathbf{y}_v = 1 - c\beta \in \mathbb{R}, \quad (22c)$$

$$\mathbf{y}_e = [\boldsymbol{\tau}_h^\top \ y_z \ y_{ho}]^\top \in \mathbb{R}^4, \quad (22d)$$

whose reference values are denoted by \mathbf{y}_\bullet^r . The motion reference \mathbf{y}_m^r is provided by an external trajectory planner, and \mathbf{y}_s^r , \mathbf{y}_e^r , and \mathbf{y}_v^r are set to 0.

Consequently, we compute the cost function of the OCP as the sum of the individual task costs. Each term is given by the weighted square Euclidean norm of the difference between \mathbf{y}_j and \mathbf{y}_j^r , denoted by $\|\bullet\|_{\mathbf{W}_j}^2$, where \mathbf{W}_j a diagonal weighting matrix for the task $j \in \{m, s, e, v\}$.

As a result, we can formulate the OCP as follows

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

$$s.t. \quad \mathbf{x}_0 = \mathbf{x}(t), \quad (23b)$$

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}_k), \quad k \in \{0, \dots, N-1\} \quad (23c)$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}_k), \quad k \in \{0, \dots, N\} \quad (23d)$$

$$\underline{\gamma}_k \leq \gamma_k \leq \bar{\gamma}_k, \quad k \in \{0, \dots, N\} \quad (23e)$$

$$\dot{\underline{\gamma}}_k \leq \dot{\gamma}_k \leq \dot{\bar{\gamma}}_k, \quad k \in \{0, \dots, N-1\} \quad (23f)$$

$$(a_1 - a_2)^2 \leq c_{ho,k}, \quad k \in \{0, \dots, N\} \quad (23g)$$

$$|x_{H_t}/z_{H_t}|_k \leq \tan \frac{\alpha_h}{2}, \quad k \in \{0, \dots, N\} \quad (23h)$$

$$|y_{H_t}/z_{H_t}|_k \leq \tan \frac{\alpha_v}{2}, \quad k \in \{0, \dots, N\} \quad (23i)$$

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, namely the human velocities \mathbf{u}_h .

IV. VALIDATION

A. Motion Reference Generation

The motion reference trajectory \mathbf{y}_m^r drives the AR in front of the human. This trajectory is generated by a motion

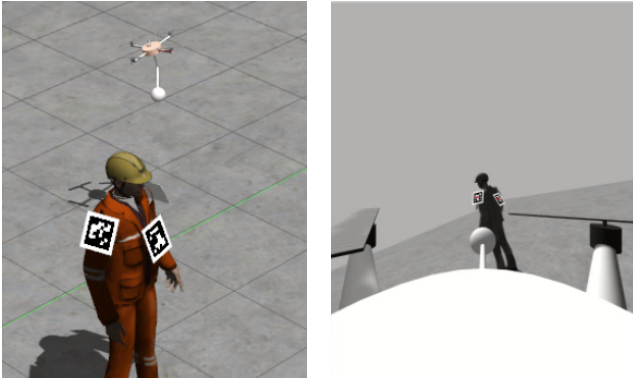


Fig. 3: On the left, a snapshot of the simulated platform performing the Approaching phase presented in Sec. IV-C. On the right, a frame of the robot onboard camera, taken during this motion.

planner employing a spline interpolation to connect a set of intermediary waypoints. The first waypoint of the trajectory is the initial location of the AR, while the final one is a position in front of the human. Moreover, the robot has to maintain a suitable distance from the human, which shall be not too close to result unpleasant and not too far to make evident the intent of the robot to interact. Thus, the robot approaches the human until reaching this distance. Then, the robot moves toward the human trunk performing a circular path. Once in sight of the human, it starts approaching more closely, and later the controller delivers the object.

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. The robot is a collinear quadrotor, which hands over a small object with a weight of 250g, picturing a small tool. The object is attached on a stick in front of the AR shifted by 45° w.r.t. the arms. A picture of the simulated system is provided 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 [12], based on [18]. The simulated hardware interface, the state estimation, and the path planning rely on the TeleKyb3 software, available on the OpenRobots platform¹. The software framework is connected to a Gazebo simulated system that emulates the actual platform interface, whose inputs are the rotor velocities. Furthermore, in the simulator, we can control the planar position, the yaw, and the motion of the arm of the simulated human coworker by means of a joystick. Details on how to use this software can be found in the provided git repository². The block diagram of the framework is drawn in Fig. 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 velocities are

retrieved at 100Hz. Gaussian noise is applied to each of these simulated measurements, with respective standard deviations of 0.003m, 0.003rad/s and 0.02m/s², and 0.03rad/s. The simulated AR is equipped with a front-facing 60Hz monocular camera. The simulated human features a set of Aruco fiducial markers [19], which are used to retrieve the position of the human trunk in the world frame, \mathbf{p}_{H_t} . This choice is motivated by the practicality of such markers, and by the low computational time and little power required for the detection process. This allows providing \mathbf{p}_{H_t} effectively at 60Hz to the filtering algorithm. Recent developments in machine learning algorithms allow embedding fast, computationally efficient, and reliable object detection solutions on board ARs, e.g. [20]. The use of such algorithms would relieve the human coworker from wearing markers. However, they 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 tools is promising, but is left out of the scope of this work.

In the simulator, we neglect the weight of the carried tool and the wrench arising from the physical interaction, accounting for the last term in (3c). The compensation of this effect is left for future work, as this manuscript focuses on safety and ergonomic awareness in the control approach.

The simulation is composed of two phases. First, the robot performs the *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} , specifically during the Approaching phase. Then, the effects of enabling the ergonomic costs are demonstrated during the Reaching phase, where the handover takes place.

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 (1) and orange (2) curves correspond to simulations with and without the prediction of the human motion in the controller, i.e. respectively with $\mathbf{u}_h^{(1)} \neq 0$ and $\mathbf{u}_h^{(2)} = 0$. The initial AR position is marked as a blue square while the starting human location as a red circle.

We first present the simulation depicted by the blue curve. The human moves from the red circle toward the location denoted by a red star when the AR reaches the blue circle. Similarly, when the human starts moving to the final position marked with a red triangle, the robot is at the blue star. Finally, the AR reaches the final position denoted by a blue triangle. As a result, since the trajectory is specified w.r.t. \mathcal{F}_{H_t} , the controller modifies the robot motion according to the human one.

As shown, this scheme allows positioning the AR in front of the human for the Approaching phase regardless of the

¹<https://git.openrobots.org/projects/telekyb3>

²<https://redmine.laas.fr/projects/nmhc-handover>

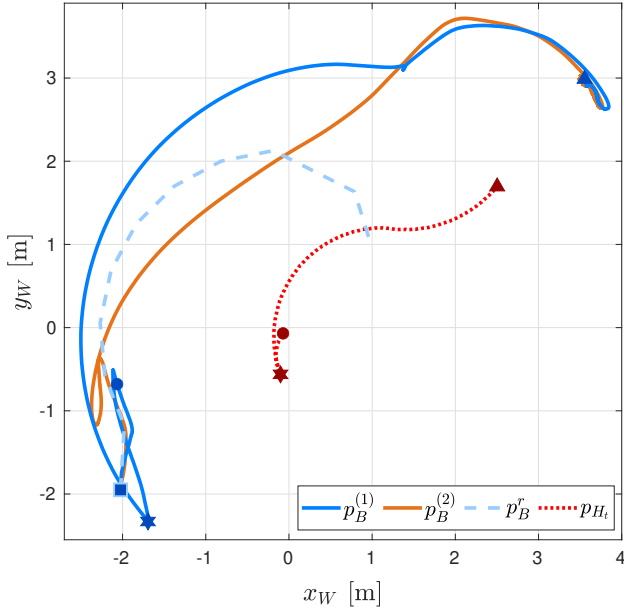


Fig. 4: Top view of the Approaching phase. In light blue, the initial reference motion task generated by the motion planner. In blue and orange, the motion of the robot in two simulations, (1) and (2), and in red the trajectory of the human.

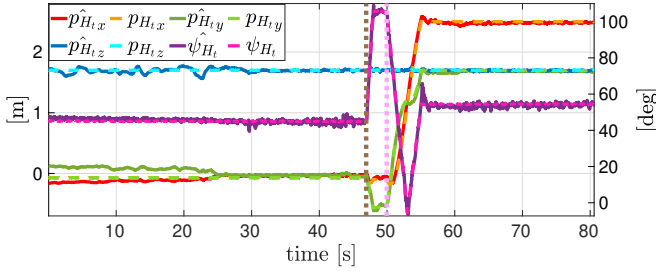


Fig. 5: Estimated and ground-truth values of the human position and yaw during the Approaching phase.

unknown human motion, without the need for 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 the future poses of the coworker.

The orange curve shows a replica of the previous simulation, where the coworker moves in the same way, but the human motion is disabled in the controller (i.e. $\mathbf{u}_h = 0$). 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 are the output onboard estimates. The brown vertical dotted line corresponds to the first displacement of the human shown in the previous figure, while the pink one to her second motion. In general, the human position is well estimated, except for the first part of the simulation, where the distance toward the human is large, increasing the difficulty of detecting precisely

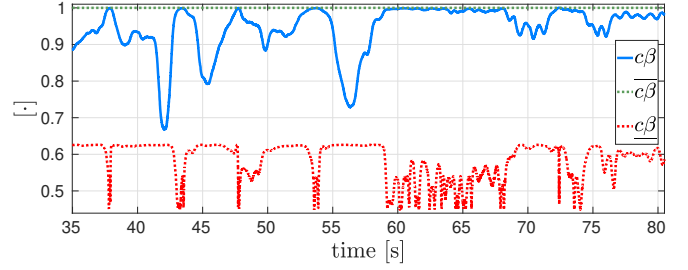


Fig. 6: Visibility constraint over time during the Approaching phase, where c_β synthetically summarizes the FoV constraints (19) in a 1D representation.

the Aruco markers. In the time during which the coworker moves, it can be noticed that the estimation is less affected by noise. The reason can be appreciated in Fig. 3, where a frame of the robot onboard camera is shown. That image has been taken while the robot navigates around the coworker to reach the final position. At that moment, the camera is observing two Aruco markers, which provide more measurements to better estimate the human pose.

Lastly, in Fig. 6, we report the visibility task over the Approaching phase. As the plot suggests, the controller can maintain the human trunk inside the FoV of the camera during the whole simulation, and close to the center ($c_\beta = 1$). 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.

D. Handover

Once the AR is in front of the operator, the handover phase starts: the ergonomic objectives are enabled, and the motion task is disabled, as presented in Sec. III-C.

Fig. 7 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 between the visibility and the desired handover distance (displayed as a brown vertical dotted line). Indeed, moving toward the most ergonomic location would jeopardize the detection of the human trunk and feel unnatural for the human coworker. Instead, the controller drives the robot to another position which prevents forcibly pushing the object into the human hands and, at the same time, provides a good level of ergonomics.

V. CONCLUSIONS

In this work, we propose an ergonomics-aware NMPC designed to autonomously perform the handover of a tool to a human coworker. The formulation considers the closed-form equations of the human shoulder and elbow torques to determine the best handover position which minimizes the human physical effort required to retrieve the object. We build upon previous works to ensure that the actuation limitations of the system are strictly observed during the

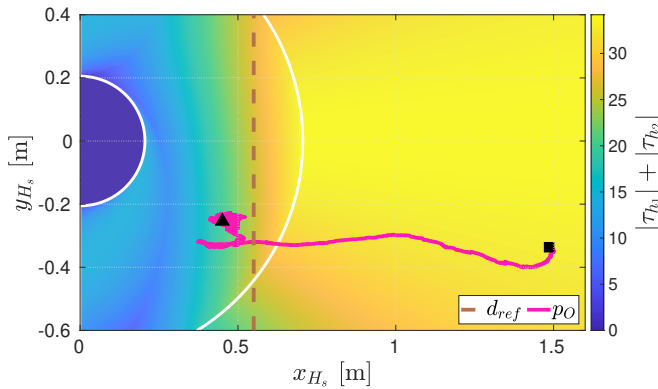


Fig. 7: Side view of the human arm plane. The color gradient shows the sum of the joint torques. p_O is the projection of O over the Reaching phase, from the initial position (square) to the handover location (triangle).

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 to increase the reactivity of the framework with any unexpected human motion, without the need for online re-planning. The human motion model is included in the NMPC equations allowing the controller to predict the future poses of the worker through the horizon. Additionally, the relative formulation allows embedding a safety barrier to avoid collisions with the human, which is of paramount importance in HRI. Finally, the framework is tested in a Gazebo simulation. The controller tracks the desired path and brings the robot to the correct position for performing the handover.

Despite its richness, the presented work still leaves some open challenges which will be addressed in future work. Firstly, the physical interaction between the two agents is neglected, while a wrench may arise during the exchange of the tool (e.g. by the human pulling or pushing the object before being released by the robot). 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 torques at the shoulder and the elbow, and it neglects the transfer of interaction to the torso. Therefore, it is possible to enlarge this model to take into consideration also the trunk torques and relate the ergonomics to the shifts in the body posture. Thirdly, the use of higher DoF models makes finding a closed-form solution to the inverse kinematics problem unfeasible. Consequently, the use of optimization to solve the inverse-kinematics problem could be investigated, and 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] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human-robot collaboration," *Autonomous Robots*, vol. 42, no. 5, pp. 957–975, Jun. 2018.

[5] M. Gienger, D. Ruiken, T. Bates, M. Regaieg, M. Meißner, 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.

[6] S. S. M. 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.

[7] V. Ortenzi, A. Cosgun, T. Pardi, W. P. Chan, E. Croft, and D. Kulic, "Object Handovers: A Review for Robotics," *IEEE Trans. on Robotics*, vol. 37, no. 6, pp. 1855–1873, Dec. 2021.

[8] 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.

[9] J. R. Medina, F. Duvallat, 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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 Work. on Aerial Robotic Systems Physically Interacting with the Environment*, 2021, pp. 1–8.

[15] 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.

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

[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.