Dual-arm Object Transportation via Model Predictive Control and External Disturbance Estimation

Maolin Lei[1], Mario Selvaggio[2], Ting Wang[3,†], Fabio Ruggiero[2], Cheng Zhou[4], Chen Yao[3], Yu Zheng[4]

***Abstract*— This paper addresses the problem of transporting a rigid box filled with unknown objects with a dual-arm robotic system. Enforcing non-sliding contact behavior, which guarantees the transportation of the box despite the unknown load's action, is the main difficulty in this setting. To solve this problem, we propose a high-level model-predictive controller, which uses a nonlinear extended state observer to estimate the external disturbance, and determine the wrench required to the box for tracking a trajectory. A quadratic program transforms the calculated wrench into optimal desired contact forces on the two end-effectors. Finally, a low-level admittance control framework with an inner velocity loop is established to indirectly control the actual contact forces. We verify the effectiveness of the proposed control method with experiments carried out on a real dual-arm robotic system.**

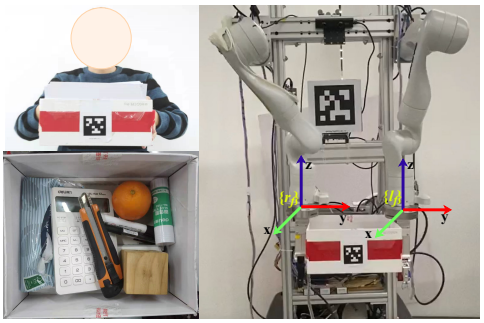


Fig. 1: Transporting a box filled with multiple unknown objects. Top-left: the way a human transports a box. Bottom-left: a box containing multiple objects. Right: the dual-arm robot transporting a box.

I. INTRODUCTION

Dual-arm robots open up possibilities to perform robotic manipulation tasks with human-like dexterity [1], [2]. Carrying a large rigid box employing both arms is usually carried out by humans to simultaneously transport multiple objects, thus improving efficiency when compared to single object transport (see Fig. 1). Although transporting a box represents a relatively simple manipulation task for humans, many requirements need to be satisfied to successfully complete the same task with a robot [3], [4]. Indeed, a dual-arm robot needs to accurately control the contact forces between the end-effectors and the box to avoid slippage. This can be realized by sensing and opportunely compensating for external forces, such as gravity. In this setting, force closure is the property that allows implementing a non-sliding grasp [5]. If the force closure property is realized, the hands can arbitrarily "squeeze" the rigid box, consequently increase the frictional components of the contact forces, thus preventing the object from falling. Besides stably holding it, in some applications the robot may need to transport the box tracking a desired trajectory. The realization of this task requires accounting for the box dynamics when deciding which contact forces to impose. Solving this problem may open new possibilities for the effective deployment of dual-arm robots into several fields, such as human-robot collaboration scenarios, where the robot can assist the human by transporting the box while the human adds or remove objects from it.

In this paper, we propose a method combining the Model Predictive Control (MPC) approach with a Nonlinear Extended State Observer (NESO) to enable the realization of this task. MPC allows to safely transport the box (hereafter, denoted to as 'object') along a desired trajectory when its dynamics is perfectly known. In the case of unknown and time-varying loads applied, we propose to use a NESO to online estimate this disturbance. The combined (MPC+NESO) controller outputs the desired optimal wrench for the object to be produced by end-effectors' contact forces. Optimal contact forces between the object and the end-effectors are computed solving a Quadratic Problem (QP) accounting for the non-sliding friction cone constraints. While MPC, NESO, and QP constitute the high-level control of the devised architecture, a low-level controller, based on Admittance Control (AC), finally regulates the contact forces to the desired value. Compared to only using AC, our high-level architecture can ensure that the robot adjusts its reference to track the desired trajectory while respecting practical robotic and manipulation constraints.

The main contributions of this work is a high-level control architecture that allows: (i) a dual-arm robot to transport multiple objects altogether in a box; (ii) optimally distributing the contact forces among end-effectors with high computational efficiency; (iii) simultaneously tracking the

This work was supported by National Natural Science Foundation of China (U20A20201), and by the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017008 (Harmony).

† denotes corresponding author.

[1]M. Lei is with the Humanoids and Human Centered Mechatronics (HHCM) research line of the Istituto Italiano Di Tecnologia (IIT), 16163 Genova, Italy. e-mail: maolin.lei@iit.it

[2]M. Selvaggio and F. Ruggiero are with the PRISMA Lab, Department of Electrical Engineering and Information Technology, University of Naples Federico II, 80125 Napoli, Italy. e-mail: mario.selvaggio, fabio.ruggiero@unina.it

[3]T. Wang and C. Yao are with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang, China. e-mail: wangting,cyao@sia.cn

[4]C. Zhou and Y. Zheng are with Tencent Robotics X Lab, Shen Zhen, Guang Dong, China . e-mail: chowchzhou, petezheng@tencent.com

desired trajectory and compensating for time-varying external disturbances. The overall architecture is shown in Fig. 2.

II. RELATED WORKS

1) Dual-arm object transportation: the considered problem was faced by considering the dual-arm robot and the object as a closed-chain mechanism in [6], [7]. To increase the dexterity, some researchers combined dual-arm re-grasping methods with object transportation. Hence, the robot changed the grasp points finding more suitable positions for object manipulation [8], [9], [10]. However, all the methods introduced so far consider the object's motion only and not the contact forces exchanged with the end-effectors. The robot needs to embed gripper-like tools at the end-effectors to effectively constrain the object retaining its slipping motion. When this is not realizable, object slipping can be prevented using force closure [11], [12], while the robot needs to appropriately plan and regulate the contact force between its end-effectors and the object [13]. Some researchers proposed a non-sliding manipulation strategy using on dual-arm robot in [12], but this strategy can not track the desired trajectory and copy with external disturbances. In the previous works, force control methods are ultimately exploited to determine the motion of each end-effector. A standard method to control both force and motion is impedance/admittance control [5], which was extensively used on dual-arm robotic systems [14], [15], [16], [17]. However, these papers only aim at controlling the contact force and the object's motion, but they did not consider the friction constraints to hold the object stably. Embedding this constraint inside the controller helps the robot to make the decision on how to adjust its motion and, consequently, the contact forces [18].

2) MPC and external disturbances estimation: MPC is widely used in robotics because it provides consistent inputs under several kinds of constraints, modeling errors, and disturbances from the environment [19]. In [18], researches combine AC and MPC to follow the desired path with different kinds of uncertainties. This method greatly improves the effectiveness and safety for completing the task. Although MPC techniques can be inherently robust, they may still fail in presence of large external disturbances. A framework based on projected inverse dynamics of the robot to estimate the external force without any sensors was proposed in [20]. However, the robot needs to be equipped with torque sensors at each joint, and it cannot track the desired trajectory accurately because it does not consider the mass of the object. MPC and observers are often combined to increase the robustness, as done for legged robots in [21], [22]. The nonlinear dynamic model is often linearized and then combined with a nonlinear observer to estimate the disturbance [22]. The observer needs to gather the robot's state in real-time to estimate the disturbance, but it is a complex task with limited sensors. To overcome these limitations, the Extended State Observer (ESO) proposed in [23] uses minimal information from the model. A Linear Extended State Observer (LESO), dealing with time-varying interaction forces between the human and the robot, was

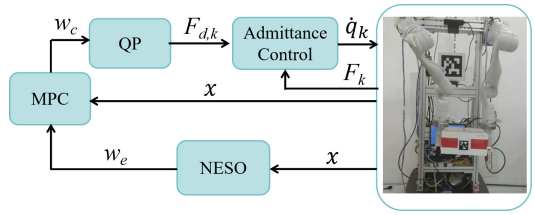


Fig. 2: Schematic representation of the devised control framework for dual-arm object transportation tasks. Symbols and blocks are explained in Sec. III.

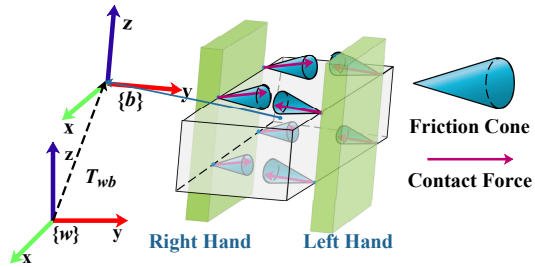


Fig. 3: Representation of the object (black lines) grasped by left and right end-effectors (green). Contact points lie on the vertices of the contact area where the friction cones (blue) and the contact forces (magenta) are shown.

proposed in [24]. Combining the LESO with the MPC allows designing a more robust controller [25], [22]. Although the LESO is superior to the NESO in parameter tuning and theoretical analysis, it may be less effective than the NESO for nonlinear models [26], [27]. For this reason, researchers choose to combine the MPC and NESO to build more robust and effective controllers [28], [26]. However, this solution has never been previously applied to dual-arm robot manipulation tasks such as the one considered here.

III. CONTROL ARCHITECTURE

A. Object Dynamic Model

The transformation between the the world frame $\{w\}$ and the body frame $\{b\}$ is given by $T_{wb} \in SE(3)$ and it is illustrated in Fig. 3. According to the Newton-Euler equations, the general rigid object dynamics can be written in $\{b\}$ as [29]

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{O}_{3\times3} \\ \boldsymbol{O}_{3\times3} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{v}}^b \\ \dot{\boldsymbol{\omega}}^b \end{bmatrix} + \begin{bmatrix} \boldsymbol{\omega}^b \times \boldsymbol{M}\boldsymbol{v}^b \\ \boldsymbol{\omega}^b \times \boldsymbol{I}\boldsymbol{\omega}^b \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^b \\ \boldsymbol{\tau}^b \end{bmatrix}. \quad (1)$$

The matrix $\boldsymbol{M} = m\boldsymbol{I} \in \mathbb{R}^{3\times3}$, where $m > 0$ is the object mass and $\boldsymbol{I} \in \mathbb{R}^{3\times3}$ is the identity matrix, represents the object mass matrix. The matrix $\boldsymbol{I} \in \mathbb{R}^{3\times3}$ is the diagonal and positive-definite object inertia matrix. Let $\bar{\boldsymbol{G}} = \begin{bmatrix} 0 & 0 & mg & 0 & 0 & 0 \end{bmatrix}^T$ be the wrench produced by gravity acceleration g in $\{w\}$, $\boldsymbol{R}_{wb} \in SO(3)$ and $\boldsymbol{p}_{wb} \in \mathbb{R}^3$ be the rotation matrix and the position, respectively, between $\{b\}$ and $\{w\}$. The relation between $\boldsymbol{w}_g$ and $\boldsymbol{G}$ can be written as

$$\boldsymbol{w}_g = \begin{bmatrix} \boldsymbol{R}_{wb}^T & \boldsymbol{O}_3 \\ -\boldsymbol{R}_{wb}^T \boldsymbol{S}(\boldsymbol{p}_{wb}) & \boldsymbol{R}_{wb}^T \end{bmatrix} \bar{\boldsymbol{G}},$$

with $\boldsymbol{S}(\cdot) \in \mathbb{R}^{3\times3}$ the skew-symmetric operator [29] and $\boldsymbol{O}_3 \in \mathbb{R}^{3\times3}$ the zero matrix.

To derive our controller, it is convenient to rewrite the object dynamic model (1) as

$$\begin{bmatrix} \dot{\boldsymbol{v}}^b \\ \dot{\boldsymbol{\omega}}^b \end{bmatrix} = -\bar{\boldsymbol{M}}\boldsymbol{A} \begin{bmatrix} \boldsymbol{v}_t^b \\ \boldsymbol{\omega}^b \end{bmatrix} + \bar{\boldsymbol{M}}\boldsymbol{w}^b, \quad (2)$$

where

$$\bar{\boldsymbol{M}} = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{I} \end{bmatrix}^{-1}, \quad \boldsymbol{A} = \begin{bmatrix} \boldsymbol{S}(\boldsymbol{\omega}^b)\boldsymbol{M} & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{S}(\boldsymbol{\omega}^b)\boldsymbol{I} \end{bmatrix}. \quad (3)$$

Denoting the object state vector as $\boldsymbol{x} = [\boldsymbol{x}^{w^{\mathrm{T}}} \ \boldsymbol{v}_t^{b^{\mathrm{T}}}]^{\mathrm{T}} \in \mathbb{R}^6$, with $\boldsymbol{x}^w$ vector of the parameterized object position in $\{w\}$, and $\boldsymbol{v}_t^b$ object velocity in $\{b\}$. The force $\boldsymbol{f}^b \in \mathbb{R}^3$ applied at the origin of $\{b\}$ can be decomposed as $\boldsymbol{f}^b = \boldsymbol{f}_g + \boldsymbol{f}_e + \boldsymbol{f}_c$, where $\boldsymbol{f}_g = [0 \ \ 0 \ \ mg] \in \mathbb{R}^3$ is the gravity force, $\boldsymbol{f}_e \in \mathbb{R}^3$ is the unknown external disturbance, $\boldsymbol{f}_c \in \mathbb{R}^3$ is the force produced by the contact forces between the object and the end-effector, all expressed in $\{b\}$. Considering only translations, the dynamic model (1) can be equivalently expressed as

$$\dot{\boldsymbol{x}} = f\left(\boldsymbol{x}, \boldsymbol{w}^b\right) = \begin{bmatrix} \boldsymbol{R}_{wb}^T \boldsymbol{v}_t^b \\ \boldsymbol{M}^{-1}\left(-\boldsymbol{S}(\boldsymbol{\omega}^b)\boldsymbol{M}\dot{\boldsymbol{v}}_t^b + \boldsymbol{f}^b\right) \end{bmatrix}, \quad (4)$$

which is the state-space form of the object translational dynamics.

B. External Disturbance Estimation

To improve the robustness of our controller, we use an observer to estimate for the unknown external force applied to the object. To this end, a NESO is introduced in this section. The dynamic model (4) is a coupled multi-input multi-output system that can be rewritten in this form

$$\begin{cases} \dot{\boldsymbol{x}}_1 = f_0\left(\boldsymbol{x}_1\right)\boldsymbol{x}_2 \\ \dot{\boldsymbol{x}}_2 = f_1\left(\boldsymbol{x}_1, \boldsymbol{x}_2\right) + \boldsymbol{B}_1\boldsymbol{u}_e \\ \boldsymbol{y} = \boldsymbol{x}_1 \end{cases}, \quad (5)$$

with $\boldsymbol{x}_1 = \boldsymbol{x}^w$, $\boldsymbol{x}_2 = \boldsymbol{v}_t^b$, $\boldsymbol{B}_1 = \bar{\boldsymbol{M}}$, $\boldsymbol{u}_e = \boldsymbol{f}_e$, $f_0(\boldsymbol{x}_1) = \boldsymbol{R}_{wb}^T$, and $f_1(\boldsymbol{x}_1, \boldsymbol{x}_2) = -\boldsymbol{M}^{-1}\boldsymbol{S}(\boldsymbol{\omega}^b)\boldsymbol{M}\boldsymbol{x}_2$.

To design the disturbances estimator, it is convenient to decouple (5). Following [25], we can define $\hat{\boldsymbol{x}}_2 = f_0\left(\boldsymbol{x}_1\right)\boldsymbol{x}_2$ obtaining the new form of the object dynamic model

$$\begin{cases} \dot{\hat{\boldsymbol{x}}}_1 = \hat{\boldsymbol{x}}_2 \\ \dot{\hat{\boldsymbol{x}}}_2 = \boldsymbol{H}(\boldsymbol{x_1}, \boldsymbol{x_2}) + \hat{\boldsymbol{u}} \\ \boldsymbol{y} = \hat{\boldsymbol{x}}_1 \end{cases}, \quad (6)$$

where $\boldsymbol{H}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \left(f_0^{'}\left(\boldsymbol{x}_1\right)\boldsymbol{x}_2\right)\boldsymbol{x}_2 + f_0\left(\boldsymbol{x}_1\right) f_1\left(\boldsymbol{x}_1, \boldsymbol{x}_2\right)$, $\hat{\boldsymbol{u}} = \hat{\boldsymbol{B}}(\boldsymbol{x_1})\boldsymbol{f}_e$ and $\hat{\boldsymbol{B}}(\boldsymbol{x_1}) = f_0^{-1}\left(\boldsymbol{x}_1\right)\bar{\boldsymbol{M}}$. In the new form (6), the system can be treated as six single-input single-output subsystems.

With this nonlinear dynamic model of the object given by (6), the NESO was used here to estimate the force disturbances caused by the environment/gravity. Foe each component of the state in (6), the observer can be designed as

$$\begin{cases} \dot{z}_1 = z_2 - \beta_1 e \\ \dot{z}_2 = z_3 + \hat{U} - \beta_2\gamma\left(e, \alpha_1, \delta\right) \\ \dot{z}_3 = -\beta_3\gamma\left(e, \alpha_2, \delta\right) \\ e = z_1 - y \end{cases}, \quad (7)$$

with

$$\gamma\left(e, \alpha, \delta\right) = \begin{cases} \dfrac{e}{|\delta|^{1-\alpha}} & |e| \le \delta \\ |e|^{\alpha} sign\left(e\right) & |e| > \delta \end{cases},$$

and where β_1, β_2, β_3, $\delta > 0$, α_1, $\alpha_2 \in [0, 1]$ are gains/thresholds. Suitable choice for β_1, β_2, β_3 can greatly reduce the observation error. Besides, e represents the error between the measured and the observed position, and z_1, z_2 represent the observer state tracking $\hat{x}_1$ and $\hat{x}_2$, respectively. In this setting, $\hat{x}_1$, $\hat{x}_2$ represent the position and the velocity of the object in the world frame, and z_3 represents external disturbances and the external force which can be calculated as $\bar{\boldsymbol{f}}_e = \hat{\boldsymbol{B}}^{-1}\boldsymbol{z}_3$. More information about the design of such observer can be obtained in [23], [30], [31].

C. Model Predictive Control

For the envisioned object transportation task, let $\boldsymbol{x}_{\text{ref}} = \left[\boldsymbol{x}_{\text{ref}}^{w^{\mathrm{T}}} \quad \boldsymbol{v}_{\text{ref}}^{b^{\mathrm{T}}}\right]^{\mathrm{T}} \in \mathbb{R}^6$, $\boldsymbol{x}_{\text{ref}}^w \in \mathbb{R}^3$ be the reference position of $\{b\}$ in $\{w\}$, and $\boldsymbol{v}_{\text{ref}}^{b^T}$ be the reference object velocity in $\{b\}$, calculated from a desired trajectory. Discretizing the state-space equation (4) we obtain

$$\boldsymbol{x}\left(k+i\right) = f_k(\boldsymbol{x}(k), \boldsymbol{u}(k)) = \boldsymbol{x}\left(k\right) + T_s f(\boldsymbol{x}(k), \boldsymbol{u}(k)), \quad (8)$$

with $\boldsymbol{u} = \boldsymbol{f}_c$ the control input and $T_s > 0$ the sampling time, $\boldsymbol{x}\left(k+i\right)$ denotes the state vector at time $k+i$ predicted at time k. Then, we linearize the discretized model (8) around the reference pose, obtaining

$$\boldsymbol{x}\left(k+i\right) = \boldsymbol{A}\boldsymbol{x}\left(k\right) + \boldsymbol{B}\boldsymbol{u}\left(k\right) + T_s\boldsymbol{M}^{-1}\boldsymbol{f}_e(k), \quad (9)$$

with

$$\boldsymbol{A} = \left.\frac{\partial f_k\left(\boldsymbol{x}, \boldsymbol{u}\right)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_{\text{ref},k}, \boldsymbol{u}_{\text{ref},k}}, \boldsymbol{B} = \left.\frac{\partial f_k\left(\boldsymbol{x}, \boldsymbol{u}\right)}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}_{\text{ref},k}, \boldsymbol{u}_{\text{ref},k}}, \quad (10)$$

where $k \in \mathbb{N}$ represents the discrete time variable. In order to reduce the errors created by the model linearization, we assume the object not to have significant angular velocities. The control input is derived from the minimization of the following cost function

$$C_k = \sum_{i=1}^{N} \|\boldsymbol{z}_{k,i}\|_{Q_i}^2 + \sum_{i=1}^{N-1} \|\boldsymbol{u}_{k,i}\|_{R_i}^2 + \sum_{i=2}^{N-1} \|\Delta\boldsymbol{u}_{k,i}\|_{P_i}^2, \quad (11)$$

where $N > 0$ is the prediction horizon, $\boldsymbol{z}_{k,i} = \boldsymbol{x}\left(k+i\right) - \boldsymbol{x}_{\text{ref}}\left(k+i\right)$, $\boldsymbol{u}_{k,i}$ is the controller input, $\Delta\boldsymbol{u}_{k,i} = \boldsymbol{u}_{k,i} - \boldsymbol{u}_{k,i-1}$ is the amplitude of the input change rate and $\boldsymbol{Q}$, $\boldsymbol{R}$, $\boldsymbol{P}$ are diagonal positive-definite weight matrices.

Using the linearized model (9), the MPC formulation calculates the optimal control input within a finite prediction horizon minimizing (11) and supplies the real system with the first sample only. This can be implemented through the following QP problem

$$\min_{\boldsymbol{u}_k} \quad C_k\left(\boldsymbol{u}_k\right) \quad (12)$$

$$\text{s.t.} \quad \boldsymbol{x}\left(k+1\right) = \boldsymbol{A}\boldsymbol{x}\left(k\right) + \boldsymbol{B}\boldsymbol{u}\left(k\right) + T_s\boldsymbol{M}^{-1}\bar{\boldsymbol{f}}_{e,0} \quad (13)$$

$$\boldsymbol{u}\left(0\right) = \boldsymbol{u}_0 \quad (14)$$

$$\boldsymbol{x}\left(0\right) = \boldsymbol{x}_0 \quad (15)$$

$$-\epsilon_v \le v_x^b, v_y^b, v_z^b \le \epsilon_{\dot{v}} \quad (16)$$

$$-\epsilon_{\dot{v}} \le \dot{v}_x^b, \dot{v}_y^b, \dot{v}_z^b \le \epsilon_{\dot{v}} \quad (17)$$

where $\bar{\boldsymbol{f}}_{e,0}$, which is plugged into the object dynamic equation (13), represents the estimated external disturbance at the beginning of the control loop, (14) and (15) represent the initial state and initial input, respectively, while (16) and (17) represent the constraints for the velocity and the acceleration, respectively, imposed on the object's motion due to the actuation and physical limitations of the robot. The optimal inputs $\boldsymbol{u_k}$, calculated in each control loop, allow not only to transport the object by tracking the reference trajectory but also to compensate for external disturbances.

IV. CONTACT FORCES COMPUTATION AND CONTROL

A. Contact Force Computation

At each time step, the inputs $\boldsymbol{u}_k$, calculated solving (12), represent the body wrench that allows following a trajectory while respecting the robot's physical constraints. However, $\boldsymbol{u}_k$ needs to be generated through contact forces realized between the object and the end-effectors. Intuitively, to prevent the object from falling, we need to set some constraints on the contact forces that can ensure they always fall inside the friction cone of the contact points [32]. This assures that the object does not undergo any relative motion with respect to the end-effectors. In our setting, an extended portion of the object's surface is in contact with the end-effectors. To simplify the problem, we consider the ($n = 8$) vertices of the contact area as contact points (see Fig. 3) and assume that only linear forces can be transmitted across them. For each contact the friction cone constraint can be written as

$$\boldsymbol{N}_i^{\mathrm{T}} \boldsymbol{f}_{c,i} \leq \boldsymbol{b}_i, \tag{18}$$

where $\boldsymbol{N}_i \in \mathbb{R}^{3\times 6}$, $\boldsymbol{b}_i \in \mathbb{R}^6$,

$$\boldsymbol{N}_i = -[\mu_i \boldsymbol{n}_i - \boldsymbol{o}_i,\ \ \mu_i \boldsymbol{n}_i + \boldsymbol{o}_i,\ \ \mu_i \boldsymbol{n}_i - \boldsymbol{t}_i,\ \ \mu_i \boldsymbol{n}_i + \boldsymbol{t}_i,\ \ \boldsymbol{n}_i,\ \ -\boldsymbol{n}_i]$$

$$\boldsymbol{b}_i = \begin{bmatrix} 0 & 0 & 0 & 0 & -f_{c,i}^L & f_{c,i}^U \end{bmatrix}^{\mathrm{T}}$$

$\mu_i > 0$ is the Coulomb friction coefficient , $\boldsymbol{n}_i, \boldsymbol{o}_i, \boldsymbol{t}_i \in \mathbb{R}^3$ are the unit normal and two orthonormal tangent vectors, respectively, expressed in $\{w\}$, and $f_{c,i}^L$, $f_{c,i}^U$ are the nonnegative lower and upper bounds on the normal contact force, respectively. The first four linear inequalities given in (18) represent the friction constraint, defining a pyramidal cone limiting the direction of $\boldsymbol{f}_{c,i} \in \mathbb{R}^3$, while the last two linear inequities limit the magnitude of $\boldsymbol{f}_{c,i} \in \mathbb{R}^3$.

The body force $\boldsymbol{f}_c$, generated by the contact forces between the dual-arm robot and the object, is expressed by $\boldsymbol{f}_c = \boldsymbol{G}(\boldsymbol{r})\boldsymbol{f}$, where $\boldsymbol{G}(\boldsymbol{r})$ is the grasp matrix [29], that in this work can be partitioned as $\boldsymbol{G}(\boldsymbol{r}) = \begin{bmatrix} \boldsymbol{G}_r(\boldsymbol{r}_r) & \boldsymbol{G}_l(\boldsymbol{r}_l) \end{bmatrix}$, with $k = r, l$ the right and left end-effector contribution, respectively, $\boldsymbol{r}_{k,i} = \begin{bmatrix} r_{k,i,x} & r_{k,i,y} & r_{k,i,z} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^3$ the position vector from the i-th contact point of the k end-effector to the $\{b\}$'s origin, and $\boldsymbol{f} \in \mathbb{R}^{3n}$ be the stacked vector of contact forces. The optimal quantity $\boldsymbol{f}^*$, minimizing the body wrench error while respecting (18), is obtained by solving

$$\min_{\boldsymbol{f}} \quad \|\boldsymbol{f}_c - \boldsymbol{G}\boldsymbol{f}\|^2 \tag{19}$$

$$s.t. \quad \boldsymbol{N}_i^T \boldsymbol{f}_i \leq \boldsymbol{b}_i, \quad \forall i = 1, \ldots, n. \tag{20}$$

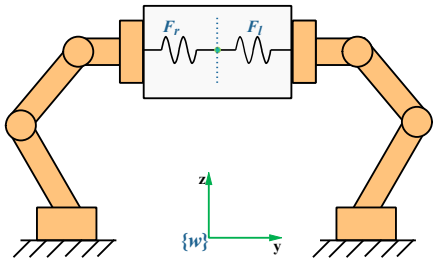


Fig. 4: Realization of contact forces via admittance control for the dual-arm robot. Symbols are explained in Sec. IV-B.

It is worth to note that the contact force calculation can be embedded into the MPC problem solving a single QP (referred to as EMPC). However, the dimension of the EMPC problem can be substantial, since the desired contact force vector $\boldsymbol{f} \in \mathbb{R}^{3n}$ has a higher dimension than the desired wrench $\boldsymbol{u}_k \in \mathbb{R}^3$. Then, we use two much smaller QP problems to compute the required wrench followed by the optimal contact forces calculation.

B. Contact Force Control

This subsection introduces the admittance control framework, defined in Cartesian coordinates, which is used to realize and regulate the contact force between each end-effector and the object. To realize the contact forces $\boldsymbol{f}$ computed solving (19), we need to transform the components belonging to each end-effector, i.e., $\boldsymbol{f}_r$ and $\boldsymbol{f}_l \in \mathbb{R}^{12}$, into the equivalent system of forces, $\boldsymbol{F}_{d,r}$ and $\boldsymbol{F}_{d,l} \in \mathbb{R}^6$, applied into $\{r_f\}$ and $\{l_f\}$ F/T sensor frames, respectively (see Fig. 1). This is necessary because there is only one force sensor at each arm's end-effector, thus it not possible to individually control the contact force at each contact point. The transformation for each arm can be written as

$$\boldsymbol{F}_{d,k} = \boldsymbol{A}_k\left(\boldsymbol{r}_{k,c}\right)\boldsymbol{f}_k, \tag{21}$$

with $k = r, l$ and $\boldsymbol{A}_k \in \mathbb{R}^{6\times 12}$ is the matrix that maps the contact forces into the equivalent force in the corresponding sensor frame. The position vector $\boldsymbol{r}_{k,c} \in \mathbb{R}^{12}$ represents the stacked position vector of contact points on the end-effector k expressed in the $\{k_f\}$ sensor frame. $\boldsymbol{F}_{d,k}$ represents the desired force to be realized by the admittance controller.

With the same arguments, the desired object velocity can be transformed into velocities of the sensor frames. These can be calculated as

$$\begin{bmatrix} \dot{\boldsymbol{x}}_{d,l}^T & \dot{\boldsymbol{x}}_{d,r}^T \end{bmatrix}^T = \boldsymbol{L}^T \boldsymbol{v}_{\mathrm{ref}}^b, \tag{22}$$

where $\boldsymbol{L}^T = \begin{bmatrix} \boldsymbol{L}_l\left(\boldsymbol{r_l}\right)^T & \boldsymbol{L}_r\left(\boldsymbol{r_r}\right)^T \end{bmatrix}^T$ maps the desired velocity of the $\{b\}$ frame into the sensor frames $\{r_f\}$ and $\{l_f\}$, and $\boldsymbol{r}_k \in \mathbb{R}^3$ is the position of the $\{b\}$ frame expressed in the k force sensor frame.

Using force control with inner velocity loop [33], the model of the dual arm robot grasping the object, qualitatively depicted in Fig. 4, can be written as $\tilde{\boldsymbol{F}}_k = \boldsymbol{M}\ddot{\tilde{\boldsymbol{x}}}_k + \boldsymbol{B}\dot{\tilde{\boldsymbol{x}}}_k + \boldsymbol{K}\tilde{\boldsymbol{x}}_k$, which represents a mechanical admittance, with $\tilde{\boldsymbol{F}}_k = \boldsymbol{F}_k - \boldsymbol{F}_{d,k}$ the error of the k-th end-effector between the desired contact force, $\boldsymbol{F}_{d,k}$, and the measured one, $\boldsymbol{F}_k$; $\ddot{\tilde{\boldsymbol{x}}}_k = \ddot{\boldsymbol{x}}_k - \ddot{\boldsymbol{x}}_{des,k}$, $\dot{\tilde{\boldsymbol{x}}}_k = \dot{\boldsymbol{x}}_k - \dot{\boldsymbol{x}}_{d,k}$, $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{x_{des,k}}$ the

errors between the current and desired acceleration, velocity, and position for the sensor frame, respectively. The desired force $\boldsymbol{F}_{d,k}$ can be calculated in (21), the desired velocity $\boldsymbol{x}_{des,k}$ in (22), and the desired acceleration $\ddot{\boldsymbol{x}}_{des,k}$ is set to zero. The matrices $\boldsymbol{M}$, $\boldsymbol{B}$, and $\boldsymbol{K}$ are the Cartesian inertia, damping, and stiffness of each end-effector, respectively. Let $\Delta t > 0$ be the time step for the admittance control loop. At the time $t-1$, the acceleration $\ddot{\boldsymbol{x}}_k^{t-1}$ can be written as

$$\ddot{\boldsymbol{x}}_k^{t-1} = \boldsymbol{M}^{-1}\left(\tilde{\boldsymbol{F}}_{kt} - \boldsymbol{B}\dot{\tilde{\boldsymbol{x}}}_k^{t-1} - \boldsymbol{K}\tilde{\boldsymbol{x}}_k^{t-1}\right) + \ddot{\boldsymbol{x}}_{des,k}^{t-1}. \quad (23)$$

The velocity at the time t can be then calculated as

$$\dot{\boldsymbol{x}}^t = \dot{\boldsymbol{x}}^{t-1} + \int_0^{\Delta t} \ddot{\boldsymbol{x}}_k^{t-1} dt. \quad (24)$$

Accordingly, the robot joints velocity at the time t can be obtained as

$$\dot{\boldsymbol{q}}_k^t = \boldsymbol{J}_{k,t}^{\#}(\boldsymbol{q})\,\dot{\boldsymbol{x}}_k^t, \quad (25)$$

where $\boldsymbol{J}_k(\boldsymbol{q})$ represents the $k = l, r$ robot Jacobian matrix mapping the joints velocity to the velocity of the end-effector, and $\boldsymbol{J}_{k,t}^{\#}(\boldsymbol{q})$ is its pseudo-inverse at a given time t. The joints velocity at the time $t-1$ will constitute the input to the next control step t. Using these desired joints velocity, we can control the contact force between the object and the end-effector.

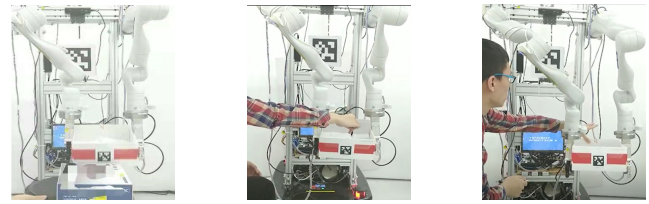
V. EXPERIMENTS

A. Experimental Setup

The experiments are carried out using the dual-arm (Kinova Gen3) platform from Tencent Robotics X Lab, shown in Fig. 1. An RGB camera (Orbbec Pro) is used to provide the controller with the object's position information. Each arm is equipped with a 6-DoF F/T sensor and a gripper at the end-effector. The sensor is used for the admittance controller, while the lateral gripper surface is used to realize the rectangular contact area between the object and the robot. The admittance control loop is implemented on a standard computer, and it runs with $\Delta t = 10^{-3}$ s sampling time. The MPC controller is implemented on a different computer in MATLAB, and it runs with $T_s = 0.1$ s sampling time, with horizon $N = 10$. The ROS framework handles the communication between the two computers. The object size is $210 \times 250 \times 110$ mm, its initial weight is $m = 0.28$ kg and the Coulomb friction coefficient between the object and end-effector (experimentally retrieved) is $\mu = 0.18$. Table I contains the value of each parameter. The parameters used for the NESO were inspired from the work in [23].

B. Experiments and Results

A comparison between the use of AC only (Sec. IV-B), MPC+AC (Sec. III-C), and MPC+NESO+AC (Sec. III-B) has been performed and can be appreciated from the attached video. When only the low-level AC is used, the reference end-effector velocities are derived from the desired object trajectory using (22), while the desired force is set to a constant value, i.e., $\boldsymbol{F}_{d,l} = [0,\, \mu mg/2,\, mg/2]^{\mathrm{T}}$ and $\boldsymbol{F}_{d,r} = [0,\, -\mu mg/2,\, mg/2]^{\mathrm{T}}$ for the left and the right sensor, respectively. It is worth to note that the MPC, introduced in Sec. III-C, requires an accurate knowledge of the model parameters (including the mass and the size of the object). When these are uncertain, good performance can not be guaranteed. In this work we use NESO to estimate and compensate for the external disturbances, improving the overall outcomes.



(a) AC (b) MPC+AC (c) MPC+NESO+AC

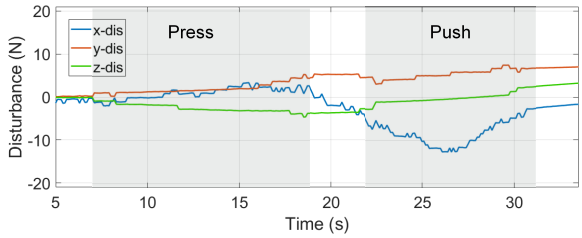
Fig. 5: Frames from the experiment A. The comparison among control modes can be appreciated from the attached video.

Two experiments are proposed to verify the validity of our approach and show the advantages introduced by the use of MPC+NESO+AC: experiment A, in which the robot can estimate and compensate for an unknown external force applied by a human operator pressing or pushing the object; experiment B, in which the robot can transport the object tracking a reference position while unknown weights are inserted/removed from the box.

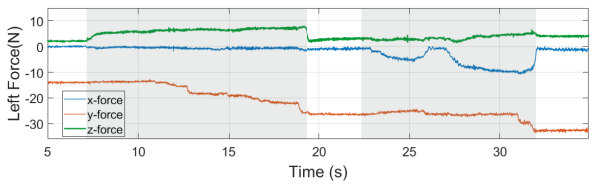
In the experiment A, the robot transports the object along the y-axis of $\{w\}$ from the initial position $\boldsymbol{p}_i = [-1.3,\, 0.3,\, -0.5]^{\mathrm{T}}$ m to the final one $\boldsymbol{p}_e = [-1.3,\, 0.8,\, -0.45]^{\mathrm{T}}$ m, following a simple constant velocity profile given by $\boldsymbol{p}_d(t) = \boldsymbol{p}_i + [0.0,\, 0.4,\, 0.1]^{\mathrm{T}} t$ m. The object is initially pushed along its z-axis and then along its x-axis. Salient frames of the experiment are shown in Fig.5. The estimated object's disturbance and the data from the sensors for each arm in the MPC+NESO+AC modality are shown in Fig. 6. As it can be noticed, the controller can effectively compensate for the external forces estimated by the NESO and, consequently, adjust the contact forces to prevent the object from falling. More specifically, when a human pushes the object in the z-direction, the contact force's magnitude in the z-direction gets bigger to compensate for the external disturbances. Meanwhile, the value of contact force along the y-direction gets bigger as well, to squeeze the object and preventing it from falling. Similarly, when a human pushes the object along the x-direction, the contact force's value along the same direction will become larger. At time

TABLE I: Parameters for the MPC+NESO and AC

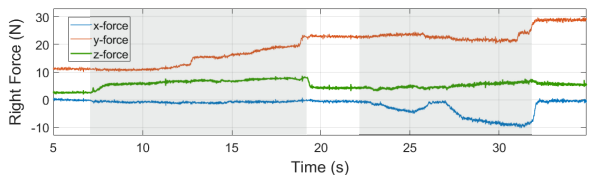
Parameter	Definition
Q	diag[4, 4, 4.5, 2, 2, 2, 0.04, 0.04, 0.04, 0.06, 0.06]
R	diag[0.002, 0.002, 0.002, 0.002, 0.002, 0.002]
P	diag[0.0002, 0.0002, 0.0002, 0.0002, 2, 0.0002]
M	diag[1, 1, 1, 1, 1, 1]
B	diag[200, 200, 200, 200, 200, 200]
K	diag[0, 0, 0, 0, 0, 0]
$[\beta_1, \beta_2 \beta_3]$	[10, 50, 100]
$[\alpha_1, \alpha_2]$	[0.5, 0.25]
δ	0.1



(a) Estimated object's disturbance



(b) Left arm's sensor

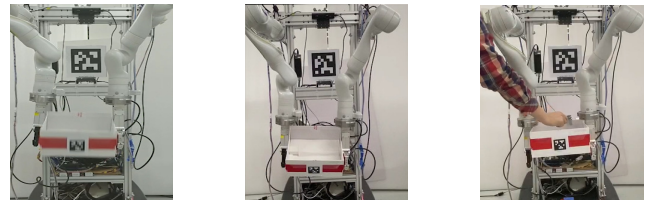


(c) Right arm's sensor

Fig. 6: Results of the experiment A, in which the human pushes the object along the x and z axes. The shaded parts represent the time slots in which the human applies an external force to the object. (a) The estimated external force in the x, y, z directions. (b)-(c) The evolution for the force sensors in the x, y, and z direction of each sensor's frame.

$25-27$ s, the x-force approaches to 0. In contrast, the contact force along the y-direction nearly does not change. When AC only is used, the robot cannot hold the object, when using MPC+AC the object falls as the human applies a force on the z-direction (see the attached video for comparison among the control modes).

In the experiment B, the initial and final object positions are selected as $\boldsymbol{p}_i = [-0.12,\, 0.45,\, -0.40]^{\mathrm{T}}$ m, $\boldsymbol{p}_e = [-0.1,\, 0.2,\, -0.1]^{\mathrm{T}}$ m, respectively. The velocity profile along the desired trajectory is $\boldsymbol{p}_d = \boldsymbol{p}_i + [0.01t,\, -0.2t,\, 0.15t]^{\mathrm{T}}$. Salient frames of experiment B are shown in Fig.7. During this experiment, objects with different weights, shown in Fig. 8 and in the attached video, are added one by one into the box. Numbers in Fig. 8 represent the order in which the objects are added. Following the same order, the weights are 50 g, 100 g, 200 g, 100 g, 200 g, and 100 g, respectively. The evolution of the estimated disturbance, caused by the added weights, and the data of force sensors in each sensor frame are shown in Fig. 8(a) and Fig. 8(b-c), respectively. From the figures, it is possible to note that the first weight is not heavy enough: the estimated disturbance remains nearly equal to zero. Correspondingly, the change for the contact force along the z-direction remains nearly close to zero, while the robot adjusts the contact force along the y-direction to track the reference trajectory. After that, when more weights are added to the box, the estimated disturbance value abruptly changes while being gradually compensated by the NESO+MPC+AC framework. At this moment, the force along the z-direction gradually changes, while the robot not only needs to adjust the contact force



(a) AC (b) MPC+AC (c) MPC+NESO+AC

Fig. 7: Photos from the experiment B. The comparison among control modes can be appreciated from the attached video.

along the y-direction to track the reference trajectory, but it also needs to impose that the contact forces fall inside the friction cone.

These tasks show that the NESO+MPC+AC is more robust than the MPC+AC and AC only control modes. The robot can automatically hold the object more tightly by estimating the external disturbance, and generating appropriate internal forces. Compared to only using AC, combining it to the proposed high-level controller can ensure that robot can adjust the input to track the desired trajectory while respecting practical robotic constraints. As shown in Fig. 5 and Fig. 7, using AC only, the robot may fail to transport the object because the friction constraints can easily be violated. Using MPC+AC, the object falls when the human adds the third weight to it. This situation also occurs when the human applies an external force to the object. The comparison can be appreciated from the attached video.

Finally, we compare our method to EMPC along the same tasks in terms of the average calculation time required in each control loop. In the experiment A, the average computation time of our method is approximately 0.0764 s, against the 0.857 s for the EMPC. In the second experiment, our method takes 0.0771 s against the 0.901 s for the EMPC, showing the advantage of our approach.

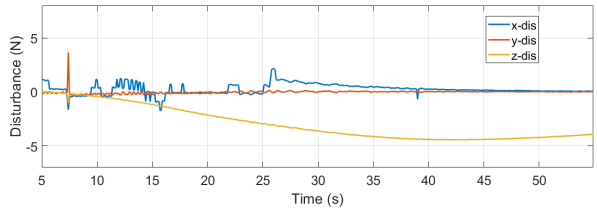
VI. CONCLUSION

In this paper we proposed a high-level MPC-based controller endowed with a NESO for transporting a box using a dual-arm robot in presence of unknown external disturbance. A QP-based controller calculates the optimal contact forces at the end-effectors, which are ultimately realized by means of an AC with inner velocity loop scheme. The proposed architecture allows calculating velocity control inputs, which are consistent with the frictional constraints of the grasp, to track the desired trajectory while simultaneously adapting to external disturbances applied to the system. Experimental validation is carried out considering unknown interaction with a human and time-varying box loading. The performed experiments showed a consistent performance improvement and bolster the devised approach.

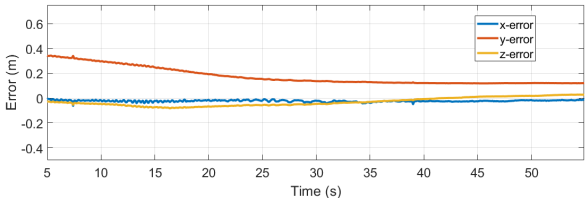
The proposed architecture is missing the estimation of external moments. This will be addressed in future works on this topic.

REFERENCES

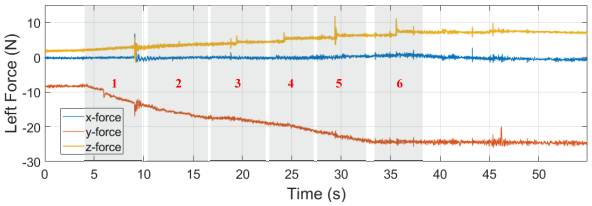
[1] N. Kashiri, L. Baccelliere, L. Muratore, A. Laurenzi, Z. Ren, E. M. Hoffman, M. Kamedula, G. F. Rigano, J. Malzahn, S. Cordasco, *et al.*,



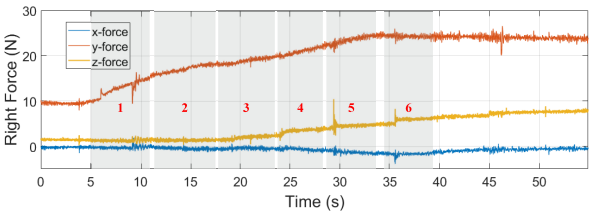
(a) Estimated disturbances



(b) Object's position errors



(c) Left arm's sensor



(d) Right arm's sensor

Fig. 8: Results of the experiment B, in which the human adds a set of weights to the box. The shaded part represent time slots in which the human adds weights to the box. (a) - Estimated external force on x, y, z directions and the position of the box. (b) - Position error on x, y, z directions. (c) and (d) - The time evolution of the force sensors signal on x, y, z directions in each sensor's frame.

"Centauro: A hybrid locomotion and high power resilient manipulation platform," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1595–1602, 2019.

[2] F. Ruggiero, J.-T. Kim, A. Gutierrez-Giles, A. C. Satici, A. Donaire, J. Cacace, L. R. Buonocore, G. A. Fontanelli, V. Lippiello, and B. Siciliano, "Nonprehensile manipulation control and task planning for deformable object manipulation: Results from the rodyman project," in *ICINCO*, pp. 76–100, Springer, 2018.

[3] M. Selvaggio, J. Cacace, C. Pacchierotti, F. Ruggiero, and P. R. Giordano, "A shared-control teleoperation architecture for nonprehensile object transportation," *IEEE Trans Rob*, vol. 38, no. 1, pp. 569–583, 2022.

[4] V. Morlando, M. Selvaggio, and F. Ruggiero, "Nonprehensile object transportation with a legged manipulator," in *2022 IEEE Int. Conf. Robot. Autom.*, 2022.

[5] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2016.

[6] M. Gharbi, J. Cortés, and T. Simeon, "A sampling-based path planner for dual-arm manipulation," in *IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, pp. 383–388, 2008.

[7] B. Cohen, S. Chitta, and M. Likhachev, "Single-and dual-arm motion planning with heuristic search," *Int J Rob Res*, vol. 33, no. 2, pp. 305–320, 2014.

[8] P. Lertkultanon and Q.-C. Pham, "A certified-complete bimanual manipulation planner," *IEEE Trans. Autom. Sci.*, vol. 15, no. 3, pp. 1355–1368, 2018.

[9] L. Chen, L. F. Figueredo, and M. R. Dogar, "Manipulation planning under changing external forces," *Auton. Robots*, vol. 44, no. 7, pp. 1249–1269, 2020.

[10] J.-P. Saut, M. Gharbi, J. Cortés, D. Sidobre, and T. Siméon, "Planning pick-and-place tasks with two-hand regrasping," in *2010 IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, pp. 4528–4533, 2010.

[11] A. Bicchi, "On the closure properties of robotic grasping," *Int J Rob Res*, vol. 14, no. 4, pp. 319–334, 1995.

[12] D. S. Carabis and J. T. Wen, "Slip avoidance in dual-arm manipulation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2872–2879, IEEE, 2018.

[13] A. Bicchi, D. Prattichizzo, and G. Antognetti, "Optimization of internal forces in force-closure grasps," in *Theory and Practice of Control and Systems*, pp. 777–782, World Scientific, 1998.

[14] R. Bonitz and T. C. Hsia, "Internal force-based impedance control for cooperating manipulators," *IEEE Trans. Robot. Autom.*, vol. 12, no. 1, pp. 78–89, 1996.

[15] T. Wimbock, C. Ott, and G. Hirzinger, "Impedance behaviors for two-handed manipulation: Design and experiments," in *Proc. 2007 IEEE Int. Conf. Robot. Autom.*, pp. 4182–4189, 2007.

[16] F. Caccavale, P. Chiacchio, A. Marino, and L. Villani, "Six-dof impedance control of dual-arm cooperative manipulators," *IEEE/ASME Trans. Mechatron.*, vol. 13, no. 5, pp. 576–586, 2008.

[17] D. Heck, D. Kostić, A. Denasi, and H. Nijmeijer, "Internal and external force-based impedance control for cooperative manipulation," in *2013 European Control Conference*, pp. 2299–2304, 2013.

[18] K. J. Kazim, J. Bethge, J. Matschek, and R. Findeisen, "Combined predictive path following and admittance control," in *2018 Annual American Control Conference*, pp. 3153–3158, IEEE, 2018.

[19] B. Calli and A. M. Dollar, "Vision-based model predictive control for within-hand precision manipulation with underactuated grippers," in *2017 IEEE Int. Conf. Robot. Autom.*, pp. 2839–2845, 2017.

[20] H.-C. Lin, J. Smith, K. K. Babarahmati, N. Dehio, and M. Mistry, "A projected inverse dynamics approach for dual-arm cartesian impedance control," *arXiv preprint arXiv:1707.00484*, 2017.

[21] F. M. Smaldone, N. Scianca, V. Modugno, L. Lanari, and G. Oriolo, "Gait generation using intrinsically stable mpc in the presence of persistent disturbances," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots*, pp. 651–656, IEEE, 2019.

[22] J. Huang, H. An, Y. Yang, C. Wu, Q. Wei, and H. Ma, "Model predictive trajectory tracking control of electro-hydraulic actuator in legged robot with multi-scale online estimator," *IEEE Access*, vol. 8, pp. 95918–95933, 2020.

[23] J. Han, "From pid to active disturbance rejection control," *IEEE Trans. Ind. Electron.*, vol. 56, no. 3, pp. 900–906, 2009.

[24] G. Sebastian, Z. Li, V. Crocher, D. Kremers, Y. Tan, and D. Oetomo, "Interaction force estimation using extended state observers: An application to impedance-based assistive and rehabilitation robotics," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1156–1161, 2019.

[25] A. Liu, W.-A. Zhang, L. Yu, H. Yan, and R. Zhang, "Formation control of multiple mobile robots incorporating an extended state observer and distributed model predictive approach," *IEEE Trans. Syst. Man Cybern. Syst.*, 2018.

[26] P. Li, H. Yang, H. Li, and S. Liang, "Nonlinear eso-based tracking control for warehouse mobile robots with detachable loads," *Robotics and Autonomous Systems*, vol. 149, p. 103965, 2022.

[27] H. Liu, J. Xi, and Y. Zhong, "Robust attitude stabilization for nonlinear quadrotor systems with uncertainties and delays," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 7, pp. 5585–5594, 2017.

[28] F. M. Smaldone, N. Scianca, L. Lanari, and G. Oriolo, "Robust mpc-based gait generation in humanoids,"

[29] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[30] H. Jingqing, "The "extended state observer" of a class of uncertain systems," *Control and Decision*, vol. 1, 1995.

[31] D. Sun, "Comments on active disturbance rejection control," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3428–3429, 2007.

[32] M. Erdmann, "On a representation of friction in configuration space," *Int J Rob Res*, vol. 13, no. 3, pp. 240–271, 1994.

[33] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 1st ed., 2008.