IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

A Learning-Based Approach for Estimating Inertial Properties of Unknown Objects from Encoder Discrepancies

Zizhou Lao[1], Yuanfeng Han[2], Yunshan Ma[3] and Gregory S. Chirikjian[1]

***Abstract*—Many robots utilize commercial force/torque sensors to identify inertial properties of unknown objects. However, such sensors can be difficult to apply to small-sized robots due to their weight, size, and cost. In this paper, we propose a learning-based approach for estimating the mass and center of mass (COM) of unknown objects without using force/torque sensors at the end effector or on the joints. In our method, a robot arm carries an unknown object as it moves through multiple discrete configurations. Measurements are collected when the robot reaches each discrete configuration and stops. A neural network then estimates joint torques from encoder discrepancies. Given multiple samples, we derive the closed-form relation between joint torques and the object's inertial properties. Based on the derivation, the mass and COM of the object are identified by weighted least squares. In order to improve the accuracy of inferred inertial properties, an attention model is designed to generate the weights used in least squares, which indicate the relative importance for each joint. Our framework requires only encoder measurements without using any force/torque sensors, but still maintains accurate estimation capability. The proposed approach has been demonstrated on a 4-degrees-of-freedom (DOF) robot arm.**

***Index Terms*—Calibration and Identification; Representation Learning; Attention Mechanism**

I. Introduction

In order to manipulate previously unseen objects, it is crucial for robots to infer physical properties such as shape, weight, material, and so forth [1], [2], [3], [4]. In this article, we develop a method for estimating mass and center of mass (COM) of a prior unknown objects being carried by robots without using force/torque sensors.

Existing works in the field of robot manipulation have made great progress in estimating the inertial properties of objects [5], [6], [7]. A common approach to these works is by observing changes in force/torque sensors during the manipulation of objects. However, commercial force/torque sensors are heavy and expensive, and are not commonly equipped on small-sized robots. Another relevant topic is estimating interaction force of robots [8], [9], [10]. In many of the force estimation methods, the force at end effector is solved by the joint torque through the corresponding Jacobian matrix. In particular, the cost function to be minimized is often defined as the summation of joint torque errors. Obtaining force information by minimizing this cost function implies that all joints are treated equally, causing the distinctive information to be less concentrated or focused. For instance, if the magnitude of torque error is relatively small, the corresponding joint should be assigned a larger weight. In order to adaptively select important information, it is necessary to develop a mechanism that assigns weights to joint torque errors dynamically.

To address the above issues, we propose a learning-based framework to estimate mass and COM of unknown objects. In the proposed framework, a neural network is designed to estimate joint torques of a robot arm. Without using force/torque sensors, we only use encoders because they are light-weight, small and cheap, and are already built in to most robots [11], [12], [13]. In particular, we find that the discrepancy between the commanded joint angle and that observed by the encoder is useful in assessing load. An attention mechanism is a type of learning technique that adaptively generates weights for input information. The weight assigned to each piece of information is generally high if it is important, and low if it is unimportant. This process leads to a dynamic selection of information. Attention mechanisms have been widely used in the field of deep learning [14], particularly in areas such as computer vision and natural language processing, but less explored in the context of mechanics and robotics. We employ attention mechanisms in the process of solving the optimal mass and COM of unknown objects held at the end effector. In particular, the cost function for inferring inertial properties is defined as a weighted sum of joint torque errors, where the weights are adjusted dynamically by an attention model. The inertial properties of objects are solved by minimizing the cost function. To the best of authors' knowledge, the use of attention mechanisms is introduced here for the first time to infer force/torque information indirectly from encoder discrepancies.

The main contributions of the proposed framework are as follows: (i) A neural network is trained to estimate joint torques accurately with only the measurements from encoders, which saves the trouble of using force/torque sensors. (ii) For a robot carrying an object at steady state, the closed-form relationship between joint toque and the object's inertial properties is derived. Based on the derivation, mass and COM

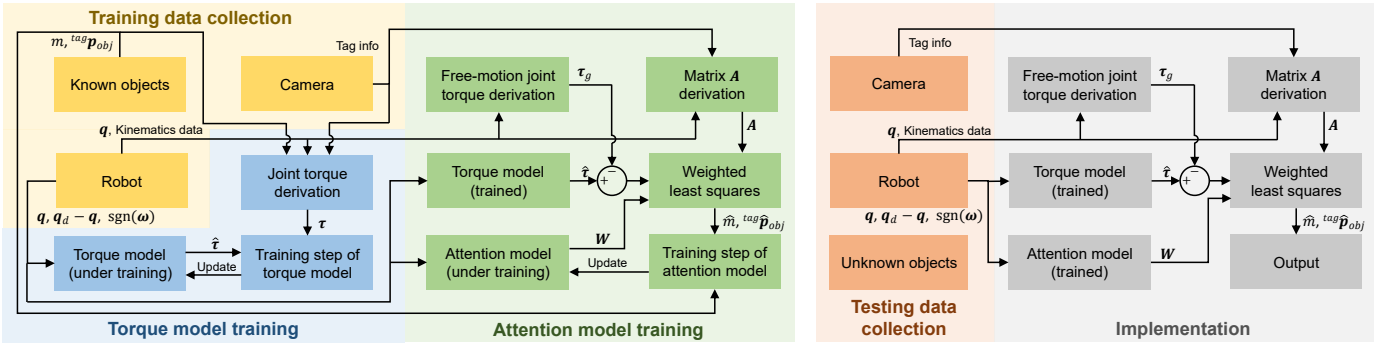
This work was supported by NUS Startup grants A-0009059-02-00 and A-0009059-03-00, CDE Board account E-465-00-0009-01, SMI Grant A-8000081-02-00, and National Research Foundation, Singapore, under its Medium Sized Centre Programme - Centre for Advanced Robotics Technology Innovation (CARTIN), sub award A-0009428-08-00.

[1]Zizhou Lao and Gregory S. Chirikjian are with the Department of Mechanical Engineering, National University of Singapore, Singapore lao.zizhou@u.nus.edu; mpegre@nus.edu.sg

[2]Yuanfeng Han is with the Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD, USA yhan33@jhu.edu

[3]Yunshan Ma is with the Sea-NExT Joint Lab, National University of Singapore, Singapore yunshan.ma@u.nus.edu

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.



(a) (b)

Fig. 1: Block diagram of the proposed approach. (a) Training process. The training data are collected using known objects. The torque model is then trained using the derived ground truth of joint torque. Subsequently, the attention model is trained based on the trained torque model. (b) Testing process. The testing data are collected using unknown objects. The torque model is utilized to estimate the joint torque, while the attention model generates the weight matrix. The mass and COM of the unknown objects are solved by weighted least squares.

can be solved analytically by weighted least squares. (iii) An attention model is designed to generate weights assigned to joints dynamically, which helps to improve the accuracy of least squares estimation of inertial properties.

II. Related Works

We discuss existing works on the identification of inertial properties, force/torque estimation, and attention mechanisms.

Identification of inertial properties. In the field of robot manipulation, many methods are proposed to estimate inertial properties of unknown objects [15], [16], [17]. For example, Atkeson et al. [5] propose a method for estimating inertial parameters of a rigid body load from the measurements of a wrist force/torque sensor and arm kinematics. In [6], the mass and COM of an object is estimated by tipping and leaning operations. Based on a force sensing plate attached to the feet of humanoid robots, another approach for estimating physical properties of unknown boxes is proposed in [7]. The above methods require measurements of force/torque sensors, which are not commonly equipped on small-sized robots.

Force/torque estimation. Another relevant topic is robot interaction force or joint torque estimation, which is applicable to scenarios where force/torque sensors are unavailable. For example, Smith and Hashtrudi-Zaad [8] and Yilmaz [9] propose approaches for robot external force estimation. In these works, joint torque of free motion is estimated by neural networks, but motor torque sensors are still needed. In [10], steady-state joint angle error is utilized to reconstruct interaction force analytically for humanoid robots. There are also many works using neural networks to infer interaction forces from visual data [18] or video [19]. However, most previous works lack the analysis of frictional torque and ignore the differences between joints.

Attention mechanisms. We design an attention model in the proposed framework to improve the performance. As one of the most important concepts in the fields of deep neural networks, attention mechanisms are widely used in various applications [20], [21], [14]. In the past few years, attention mechanisms have also been introduced to problems about robots [22], [23], [24]. However, these works mainly utilize attention mechanisms to solve graphical problems, rather than problems in mechanics and robotics. Essentially, attention mechanisms are good at focusing on the distinctive parts when processing large amounts of information. We observed that this feature is suitable for our scenario, where the errors of joint torque estimation for individual joints are constantly changing. Hence, an attention model is designed to evaluate the weights of joints dynamically during robot motions.

III. Methods

We consider a serial robot manipulator carrying an object moves through multiple configurations. For each configuration, the measurements are collected when all the joints reach steady state. It is assumed that all of the kinematic and inertial parameters of robot links are known, and the joints of the robot are controlled through PD controllers, which have been adopted as the control strategy for many robots [10], [25]. A learning-based framework is proposed to identify the mass and COM of the held object from multiple samples at steady state. Fig. 1(a) illustrates the training process of the proposed framework. By collecting multiple steady-state samples in experiments with several known objects, we train the torque model and attention model sequentially. Our framework is tested with several unknown objects as shown in Fig. 1(b). For each sample, the joint torque is estimated by the torque model, and a weight matrix is generated by the attention model. According to the outputs of networks, the mass and COM of the unknown objects are solved analytically by weighted least squares.

A. Problem Definition

Consider an N-degrees-of-freedom (DOF) robot carrying an unknown object as illustrated in Fig. 2(a). M input samples are received to identify the mass and COM of the object. The ground truth value of mass is assumed to be m. A reference frame is assigned by sticking an April tag [26] to the object. The rigid body transformation from the tag frame to the robot base frame can be obtained by camera. Therefore, the ground

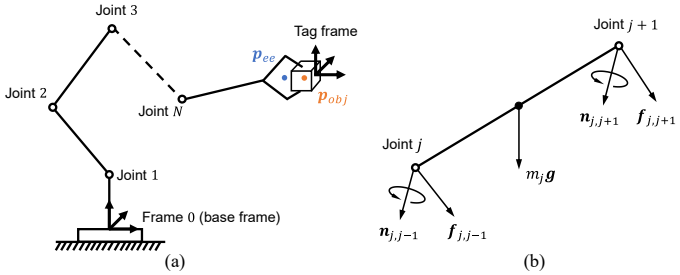


Fig. 2: (a) Schematic of an N-DOF robot carrying an object. (b) Free body diagram of the j-th link of the robot.

truth COM is represented by the 3-dimensional coordinates of COM in the tag frame ${}^{tag}\boldsymbol{p}_{obj}$. Correspondingly, we denote the estimated inertial properties as $\hat{m}$ and ${}^{tag}\hat{\boldsymbol{p}}_{obj}$. For each target configuration, the joints move to the desired position until stopping at the steady state. The measurements of encoders are regarded as actual positions. The desired joint position and actual joint position do not coincide because PD controllers are used. Regarding the i-th sample ($i = 1, 2, \ldots, M$) at steady state, the desired joint position and actual joint position are represented by $\boldsymbol{q}_{d,i}$ and $\boldsymbol{q}_i$, respectively. $\boldsymbol{\omega}_i$ refers to the joint angular velocity during the process of the robot approaching steady state. We define the direction of rotation $\text{sgn}(\boldsymbol{\omega}_i)$ as the sign of $\boldsymbol{\omega}_i$. It should be noted that $\boldsymbol{q}_{d,i}$, $\boldsymbol{q}_i$ and $\text{sgn}(\boldsymbol{\omega}_i)$ are N-dimensional vectors, of which the j-th elements $q^j_{d,i}$, q^j_i and $\text{sgn}(\omega^j_i)$ refer to the corresponding variables of joint j. We represent the desired joint position, actual joint position, and direction of rotation of all the M samples as $\boldsymbol{q}_d$, $\boldsymbol{q}$, and $\text{sgn}(\boldsymbol{\omega})$, respectively.

B. Neural Network for Estimating Joint Torque

We design a neural network to estimate robot joint torque without force/torque sensors. The proposed torque model plays two roles: (i) reconstructing the motor torque from encoder discrepancies, and (ii) eliminating the effects of friction. Therefore, the output of torque model is the estimated joint toque corresponding to external force, including the gravitational force due to the weight of robot itself, as well as the interaction force at the end effector.

We leverage the above insights in designing the torque model, which takes as input the joint position $\boldsymbol{q}$, joint position error $(\boldsymbol{q}_d - \boldsymbol{q})$, and direction of rotation $\text{sgn}(\boldsymbol{\omega})$. For joints controlled through PD controllers, the motor torque is approximately proportional to the joint position error. So we take the joint position error as input for motor torque reconstruction. During the process of a joint approaching its steady state, the friction torque is along the direction opposite to rotation. Moreover, the magnitude of friction torque can be influenced by the joint position. Therefore, the direction of rotation and the joint angle are utilized to eliminate friction torque. Among the three inputs, the steady-state joint position error is necessary in the reconstruction of joint torque. If the joints are controlled through PID controllers, the absence of joint position error would result in a situation in which the same state would be associated with multiple external forces.

As shown in Fig. 3, the torque model consists of N joint representation learning modules[1] and a torque estimator. To learn the specific information of each joint, a separate multi-layer perceptron (MLP) is utilized to embed the state of the corresponding joint. For the i-th sample and the j-th joint, the embedding is generated as:

$$\boldsymbol{h}^j_{\tau,i} = \phi(q^j_i, q^j_{d,i} - q^j_i, \text{sgn}(\omega^j_i); \boldsymbol{\Theta}^j_{\text{rep},\tau}), \tag{1}$$

where $\phi(\cdot)$ refers to the representation learning module and $\boldsymbol{\Theta}^j_{\text{rep},\tau}$ denotes the parameters of joint j's representation learning module in the torque model. Before embedding, q^j_i and $q^j_{d,i}$ are normalized, and the direction of rotation $\text{sgn}(\omega^j_i)$ is converted to a 2-dimensional binary vector, i.e. $\begin{bmatrix}1 & 0\end{bmatrix}$ for positive direction, and $\begin{bmatrix}0 & 1\end{bmatrix}$ for negative direction. Therefore, the input information of each joint state for representation learning is a 4-dimensional vector. After the representation learning process, the embeddings of all the joints are concatenated, which models the interactions between joints. Another MLP is designed as the torque estimator, which takes as input the concatenated embedding and outputs estimated joint torque as:

$$\hat{\boldsymbol{\tau}}_i = \xi(\boldsymbol{h}^1_{\tau,i} \parallel \boldsymbol{h}^2_{\tau,i} \parallel \cdots \parallel \boldsymbol{h}^N_{\tau,i}; \boldsymbol{\Theta}_{\text{est}}), \tag{2}$$

where $\xi(\cdot)$ refers to the torque estimator, $\boldsymbol{\Theta}_{\text{est}}$ denotes the parameters of the torque estimator, and $\parallel$ represents concatenation.

In the training process, the ground truth of joint torque can be calculated analytically. Fig. 2(b) illustrates the free body diagram of the j-th link. $m_j\boldsymbol{g}$ indicates the gravitational force of link. $\boldsymbol{f}_{j,j-1}$ and $\boldsymbol{n}_{j,j-1}$ are the force and moment applied on the j-th link by the $(j-1)$-th link. And $\boldsymbol{f}_{j,j+1}$ and $\boldsymbol{n}_{j,j+1}$ are the force and moment on the j-th link by the $(j+1)$-th link. When the robot is stationary, the summation of force/moment exerted on the link is zero. Therefore, the forces and moments on all the joints from the end effector to the base can be derived recursively by Newton-Euler equations. The torque on each joint can be solved as the component of moment along the rotational axis. The ground truth of joint torque of all the samples is denoted as $\boldsymbol{\tau}$. Compared to the ground truth, we apply mean squared error (L2 loss) on the estimated joint torque to train the torque model as shown in Fig. 1(a).

Free-motion joint torque $\boldsymbol{\tau}_g$ is defined as the joint torque of robot at free motion, which is due to the weight of robot itself and irrelevant to the object at end effector. We can calculate $\boldsymbol{\tau}_g$ recursively in a similar way as $\boldsymbol{\tau}$. The only difference is that the interaction force at end effector is assumed to be zero. It should be noted that both $\boldsymbol{\tau}$ and $\boldsymbol{\tau}_g$ are computed using the actual joint position $\boldsymbol{q}$. For each sample, the difference $(\boldsymbol{\tau}_i - \boldsymbol{\tau}_{g,i})$ is the portion of joint torque related to the interaction force at end effector.

C. Inferring Inertial Properties of Objects by Weighted Least Squares

The closed-form relationship between joint torque and inertial properties are derived in this subsection. Based on the

[1]We use the term "representation learning" rather than "encoder" in the machine learning sense to avoid confusion with joint encoders.

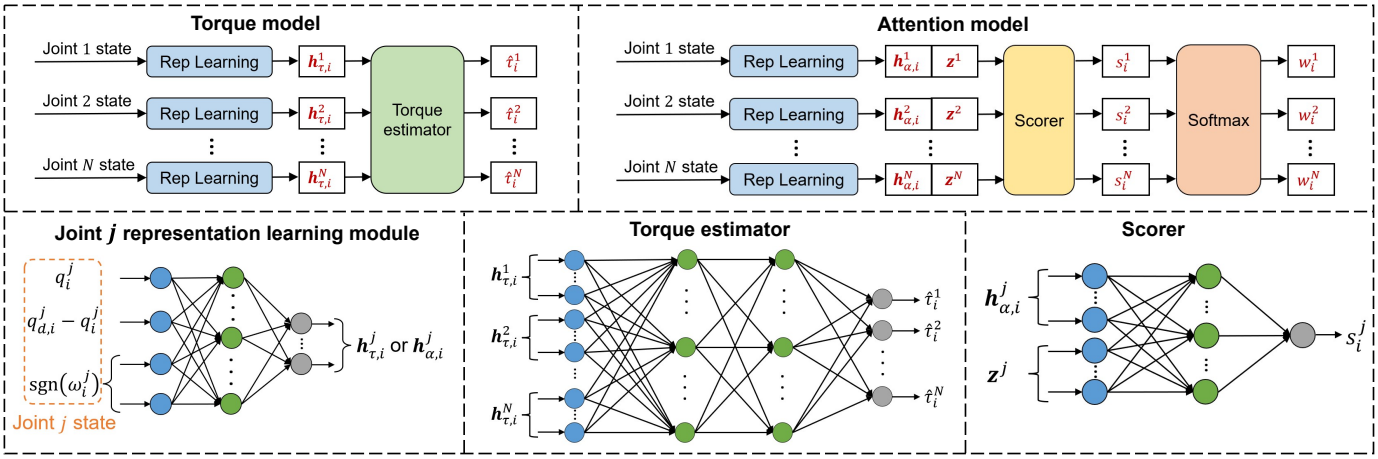


Fig. 3: The architectures of proposed neural networks. The first row illustrates the torque model and the attention model. These are more detailed descriptions of the torque model and attention model blocks in Fig. 1. The second row illustrates the submodules in the torque model and attention model.

derivation, the mass and COM of object can be identified by weighted least squares, taking as input multiple steady-state samples.

For the i-th sample, Jacobian matrix $\boldsymbol{J}_i \in \mathbb{R}^{6\times N}$ provides the relation between joint torque and interaction force at end effector as

$$\boldsymbol{\tau}_i - \boldsymbol{\tau}_{g,i} = \boldsymbol{J}_i^\intercal \boldsymbol{F}_i, \tag{3}$$

where $(\boldsymbol{\tau}_i - \boldsymbol{\tau}_{g,i}) \in \mathbb{R}^N$ is the equivalent torque related to the endpoint force, and $\boldsymbol{F}_i \in \mathbb{R}^6$ denotes the wrench applied to the environment by the end effector. In the case of robot carrying an object, the wrench can be written as

$$\boldsymbol{F}_i = \begin{bmatrix} -\boldsymbol{f}_i^\intercal & -\boldsymbol{n}_i^\intercal \end{bmatrix}^\intercal, \tag{4}$$

where $\boldsymbol{f}_i$ and $\boldsymbol{n}_i$ represent the force and moment exerted on end effector by the object.

Given a single sample, the joint torque can be estimated through trained torque model, and the corresponding Jacobian can be calculated analytically. Therefore, it is possible to solve the wrench from Eq. 3 and use it for inferring object's inertial properties. Considering the scenarios when the DOF of robot is less than 6 or the robot is at singular configurations, as well as to improve the accuracy of inference, it would be better to take multiple samples as input. However, the wrench is not fixed for various robot postures. In order to process multiple samples, we need to directly build the relationship between joint torque and inertial properties.

When a robot carrying an object is at steady state, the force and moment exerted on end effector are

$$\boldsymbol{f}_i = m\boldsymbol{g}, \tag{5}$$

$$\boldsymbol{n}_i = ({}^0\boldsymbol{p}_{obj,i} - {}^0\boldsymbol{p}_{ee,i}) \times \boldsymbol{f}_i, \tag{6}$$

where the 3-dimensional $\boldsymbol{g}$ denotes gravitational acceleration, ${}^0\boldsymbol{p}_{obj,i}$ and ${}^0\boldsymbol{p}_{ee,i}$ denotes the coordinates of object's COM and end effector in base frame, respectively. By sticking an April tag to the object as reference frame, the COM of object is represented by the coordinates in tag frame ${}^{tag}\boldsymbol{p}_{obj}$. Then, the coordinates of object's COM in base frame can be obtained as

$${}^0\boldsymbol{p}_{obj,i} = {}^0_{tag}\boldsymbol{R}_i \cdot {}^{tag}\boldsymbol{p}_{obj} + {}^0\boldsymbol{p}_{tag,i}, \tag{7}$$

where ${}^0_{tag}\boldsymbol{R}_i$ is the rotation matrix from tag frame to base frame and ${}^0\boldsymbol{p}_{tag,i}$ refers to the coordinates of tag in base frame. The above two terms can be obtained from camera. Substituting Eq. 5 and Eq. 7 into Eq. 6, and converting cross product to matrix multiplication form, the moment can be written as

$$\boldsymbol{n}_i = m[\boldsymbol{g}]_\times^\intercal \cdot ({}^0_{tag}\boldsymbol{R}_i \cdot {}^{tag}\boldsymbol{p}_{obj} + {}^0\boldsymbol{p}_{tag,i} - {}^0\boldsymbol{p}_{ee,i}), \tag{8}$$

where the skew-symmetric matrix $[\boldsymbol{g}]_\times$ is generated from the elements of $\boldsymbol{g}$ as

$$[\boldsymbol{g}]_\times = \begin{bmatrix} 0 & -g_z & g_y \\ g_z & 0 & -g_x \\ -g_y & g_x & 0 \end{bmatrix}.$$

Next, substituting Eq. 5 and Eq. 8 into Eq. 4, the wrench at end effector can be represented as

$$\boldsymbol{F}_i = \boldsymbol{B}_i \boldsymbol{x}, \tag{9}$$

where the matrix $\boldsymbol{B}_i \in \mathbb{R}^{6\times 4}$ is a function of joint position $\boldsymbol{q}_i$ and tag information as

$$\boldsymbol{B}_i = \begin{bmatrix} -\boldsymbol{g} & \boldsymbol{O} \\ -[\boldsymbol{g}]_\times^\intercal \cdot ({}^0\boldsymbol{p}_{tag,i} - {}^0\boldsymbol{p}_{ee,i}) & -[\boldsymbol{g}]_\times^\intercal \cdot {}^0_{tag}\boldsymbol{R}_i \end{bmatrix} \tag{10}$$

and $\boldsymbol{x}$ is a 4-dimensional vector determined by the mass and COM of object as

$$\boldsymbol{x} = \begin{bmatrix} m & m\,{}^{tag}\boldsymbol{p}_{obj}^\intercal \end{bmatrix}^\intercal. \tag{11}$$

By substituting Eq. 9 into Eq. 3, we can obtain the following equation:

$$\boldsymbol{\tau}_i - \boldsymbol{\tau}_{g,i} = \boldsymbol{A}_i \boldsymbol{x}, \tag{12}$$

where the matrix $\boldsymbol{A}_i \in \mathbb{R}^{N\times 4}$ is obtained as

$$\boldsymbol{A}_i = \boldsymbol{J}_i^\intercal \boldsymbol{B}_i. \tag{13}$$

So far, we have extended the Jacobian to build the relation between joint torque and inertial properties of object for a

single sample. For multiple samples, the relations in Eq. 12 can be synthetically written as

$$\boldsymbol{\tau} - \boldsymbol{\tau}_g = \boldsymbol{A}\boldsymbol{x}, \tag{14}$$

where vectors $\boldsymbol{\tau} \in \mathbb{R}^{MN}$ and $\boldsymbol{\tau_g} \in \mathbb{R}^{MN}$, and matrix $\boldsymbol{A} \in \mathbb{R}^{MN\times 4}$ are generated by stacking the corresponding variables of M samples as

$$\boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{\tau}_1 \\ \boldsymbol{\tau}_2 \\ \vdots \\ \boldsymbol{\tau}_M \end{bmatrix} \boldsymbol{\tau_g} = \begin{bmatrix} \boldsymbol{\tau}_{g,1} \\ \boldsymbol{\tau}_{g,2} \\ \vdots \\ \boldsymbol{\tau}_{g,M} \end{bmatrix} \boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \\ \vdots \\ \boldsymbol{A}_M \end{bmatrix}. \tag{15}$$

In our framework, the estimated joint torque $\hat{\boldsymbol{\tau}}$ of multiple samples at steady state is obtained by the trained torque model. The corresponding free-motion joint torque $\boldsymbol{\tau}_g$ and matrix $\boldsymbol{A}$ are calculated analytically. Assuming that the amount of samples M is large enough so that Eq. 14 is overconstrained, we can obtain an optimal approximation of vector $\boldsymbol{x}$ by weighted least squares. The cost function is defined as

$$C = (\boldsymbol{A}\boldsymbol{x} - (\hat{\boldsymbol{\tau}} - \boldsymbol{\tau}_g))^\intercal \boldsymbol{W}(\boldsymbol{A}\boldsymbol{x} - (\hat{\boldsymbol{\tau}} - \boldsymbol{\tau}_g)), \tag{16}$$

where $\boldsymbol{W} \in \mathbb{R}^{MN\times MN}$ is a diagonal weight matrix. The optimal estimation of $\boldsymbol{x}$ minimizing the cost function is

$$\hat{\boldsymbol{x}} = (\boldsymbol{A}^\intercal \boldsymbol{W} \boldsymbol{A})^{-1} \boldsymbol{A}^\intercal \boldsymbol{W}(\boldsymbol{\tau} - \boldsymbol{\tau}_g). \tag{17}$$

Finally, the estimated mass and COM of the object can be solved from Eq. 11.

D. Assigning Weights to Joints by Attention Model

In the process of inertial properties estimation, the cost function in Eq. 16 can be written as

$$C = \sum_{i=1}^{M}\sum_{j=1}^{N} w_i^j (\boldsymbol{A}_i^j \boldsymbol{x} - (\hat{\tau}_i^j - \tau_{g,i}^j))^2, \tag{18}$$

where $\boldsymbol{A}_i^j \in \mathbb{R}^{1\times 4}$ is the j-th row of $\boldsymbol{A}_i$, $\hat{\tau}_i^j$ is the j-th element of $\hat{\boldsymbol{\tau}}_i$, and $\tau_{g,i}^j$ is the j-th element of $\boldsymbol{\tau}_{g,i}$. Regarding the i-th sample and the j-th joint, $(\boldsymbol{A}_i^j \boldsymbol{x} - (\hat{\tau}_i^j - \tau_{g,i}^j))$ refers to the error between the estimated torque and the torque derived from inertial properties of object. It can be seen that C is the weighted sum of square error, where the element of the diagonal of weight matrix w_i^j is the corresponding weight.

We could simply set the weight matrix $\boldsymbol{W}$ to an identity matrix, which means that all the joints are treated equally. However, it is better to adjust the weights dynamically as the joint torque errors vary greatly in magnitude, for different joints or for different samples. For example, since the torque of a joint close to the end effector is usually smaller than the torque of a joint close to the base, it is reasonable to increase the weights of the joint close to the end effector appropriately. Moreover, when two joints are parallel, which means they provide similar information about the inertial properties of object, we can appropriately reduce the weight of the joint with larger torque error, so that the joint with smaller torque error contributes more.

In the proposed framework, we design an attention model to generate the weights of joints dynamically. As shown in Fig. 3, the attention model consists of representation learning modules, a scorer and a softmax layer. Taking as input the i-th sample, the model outputs an N-dimensional weight vector $\boldsymbol{w}_i$, corresponding to the N joints. Firstly, the joint states are embedded. Similar to torque model, the state includes joint position, joint position error and direction of rotation. For each joint, a separate MLP is designed as the corresponding representation learning module. The representation learning modules in torque model and attention model have the same architecture but the parameters are not shared. The embedding of the j-th joint and i-th sample can be represented as

$$\boldsymbol{h}_{\alpha,i}^j = \phi(q_i^j, q_{d,i}^j - q_i^j, \mathrm{sgn}(\omega_i^j); \boldsymbol{\Theta}_{\mathrm{rep},\alpha}^j), \tag{19}$$

where $\boldsymbol{\Theta}_{\mathrm{rep},\alpha}^j$ refers to the parameters of representation learning module of joint j in the attention model. The indices of joints are then appended to the latent representations. The index of each joint is represented by an N-dimensional binary vector. For example, $\begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix}$ refers to the third joint of a 4-DOF robot. Next, another MLP is introduced as a scorer to generate scores for all the joints according to the embeddings as

$$s_i^j = \gamma(\boldsymbol{h}_{a,i}^j \parallel \boldsymbol{z}^j; \boldsymbol{\Theta}_{\mathrm{scorer}}), \tag{20}$$

where $\gamma(\cdot)$ is the scorer, $\boldsymbol{z}^j$ denotes the index of joint j, $\boldsymbol{\Theta}_{\mathrm{scorer}}$ denotes the parameters of the scorer, and s_i^j is the score of joint j. Finally, the scores are normalized by a softmax function as

$$w_i^j = \frac{e^{s_i^j}}{\sum_{j=1}^{N} e^{s_i^j}}, \tag{21}$$

where the output w_i^j denotes the weight of joint j in the i-th sample.

To estimate vector $\boldsymbol{x}$ by Eq. 17 from M samples, the attention model generates M weight vectors for the corresponding samples. And the diagonal weight matrix $\boldsymbol{W}$ is generated as $\boldsymbol{W} = \mathrm{diag}(\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_M)$, of which the diagonal is the concatenation of all the weight vectors.

The attention model is trained after the torque model training process. In order to alleviate the influence of mass error on COM estimation, we use the ground truth of mass to solve COM from Eq. 11 in training process. While, in the testing process, COM is solved based on the estimated mass. Regarding the loss function, we apply L2 loss on both the estimated mass and COM. The loss for attention model training $L_{\mathrm{attention}}$ is a weighted sum of mass loss L_m and COM loss L_{com} as $L_{\mathrm{attention}} = w_m L_m + w_{\mathrm{com}} L_{\mathrm{com}}$, where the weights w_m and w_{com} are manually set.

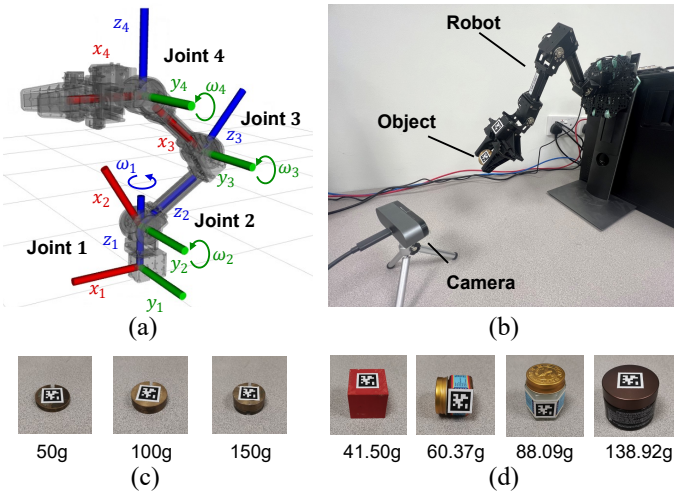
IV. Experiments

The proposed framework is verified on a 4-DOF robot OpenMANIPULATOR-X as shown in Fig. 4(a). The joints from base to end effector are joints 1, 2, 3 and 4. Fig. 4(b) illustrates the experimental setup. The axis of joint 1 is in horizontal direction so that the torque is not constant to zero. The training and testing objects are shown in Fig. 4(c) and (d). We attach April tags to the objects as reference frames.

Training Dataset. The training data consists of planned samples and random samples. The planned samples are generated through the following steps: (i) The joint positions of

TABLE I: Error of estimated inertial properties. Sensor, PE, T model and T-A model refer to the sensor based method, the position based method, proposed torque model without attention, and the proposed torque model with attention. The optimal results are marked in bold.

	Object	Sensor			PE			T model			T-A model		
		MAE	NMAE	NRMSE	MAE	NMAE	NRMSE	MAE	NMAE	NRMSE	MAE	NMAE	NRMSE
Mass error (g / % / %)	Cube	4.66	11.22	13.94	11.93	28.74	32.22	5.57	13.42	15.39	**3.61**	**8.69**	**10.88**
	Red	7.12	11.79	14.40	6.01	9.95	12.42	3.87	6.41	7.86	**3.54**	**5.86**	**7.35**
	White	6.64	7.54	9.36	7.99	9.07	11.29	**3.56**	**4.04**	**5.08**	3.65	4.14	5.18
	Black	15.84	11.40	12.97	9.11	6.56	8.14	15.33	11.04	11.49	**11.67**	**8.40**	**9.21**
	Average	8.56	10.49	12.67	8.76	13.58	16.02	7.08	8.73	9.96	**5.62**	**6.78**	**8.16**
CoM error (mm / % / %)	Cube	275.9	398.29	400.18	128.3	185.15	191.75	22.2	32.06	35.57	**14.3**	**20.61**	**23.14**
	Red	194.0	291.31	292.19	103.8	155.90	157.88	15.6	23.42	25.55	**12.9**	**19.45**	**21.43**
	White	143.4	189.77	190.47	77.0	101.87	103.44	9.9	13.14	14.55	**8.7**	**11.51**	**12.85**
	Black	103.7	121.68	122.27	60.0	70.38	71.57	13.3	15.57	16.30	**11.0**	**12.89**	**13.28**
	Average	179.2	250.26	251.28	92.2	128.33	131.16	15.3	21.05	22.99	**11.7**	**16.12**	**17.68**



(a) (b) (c) (d)

Fig. 4: (a) OpenMANIPULATOR-X robot manipulator. (b) Experimental setup. (c) Training objects. (d) Testing objects.

samples are uniformly distributed with 10° intervals on the joint space. (ii) For each joint position, all the $2^4 = 16$ possible directions of rotation are collected. (iii) The robot carries no object (at free motion), 50g, 100g and 150g to collect the above samples respectively, except for the samples that may collide. In addition, 9000 random samples are collected for each training object (including no object). In summary, we collect 82144 samples for training, including 46144 planned samples and 36000 random samples. Since each step of inertial properties inference requires multiple samples, we construct another training dataset for attention model, in which each data consists of 64 samples. The samples are randomly selected from the above training samples.

Evaluation metrics. In order to assess the performance of the proposed approach for estimating mass, COM and joint torque, we use the mean absolute error (MAE), normalized mean absolute error (NMAE), and normalized root mean square error (NRMSE) defined as

$$MAE = \frac{1}{n}\sum_{k=1}^{n}\left|\hat{y}_k - y_k\right|, \tag{22}$$

$$NMAE = \frac{\frac{1}{n}\sum_{k=1}^{n}\left|\hat{y}_k - y_k\right|}{y_{\text{scale}}} \times 100\%, \tag{23}$$

$$NRMSE = \frac{\sqrt{\frac{1}{n}\sum_{k=1}^{n}\left|\hat{y}_k - y_k\right|^2}}{y_{\text{scale}}} \times 100\%, \tag{24}$$

where $\hat{y}_k$ and y_k $(k = 1, 2, \ldots, n)$ are the estimated value and ground truth, and y_{scale} is a scale value for normalization. The actual mass is used as the scale value when calculating the error of mass. With regard to COM, the difference $|\hat{y}_k - y_k|$ in the above equations refers to the distance between the estimated COM and actual COM, and the scale value is the length of diagonal of the smallest cuboid that can enclose the object. To evaluate the estimated torque, the scale value is set to the maximum joint torque.

Baselines. We compare the proposed approach against the following two baselines:

- *Current sensor based method.* The motors are equipped with built-in electric current sensors. In the official demo code of robot arm, the joint torque is computed by multiplying the current by a coefficient. We make some improvements to this method as the first baseline. The sensor-based estimated torque of the i-th sample and j-th joint is defined as

$$\hat{\tau}^j_{\text{sensor},i} = K_{\text{sensor}} I^j_i - \text{sgn}(\omega^j_i)\tau^j_{f,\text{sensor}} + b^j_{\text{sensor}}, \tag{25}$$

  where I^j_i is the measurement of current sensor, and K_{sensor} is a constant coefficient provided by the demo code. We add a constant $\tau^j_{f,\text{sensor}}$ for each joint so that $-\text{sgn}(\omega^j_i)\tau^j_{f,\text{sensor}}$ represents the friction torque, of which the magnitude is fixed and the direction is opposite to the direction of rotation. The last term b^j_{sensor} is a constant for eliminating bias. Using the same training dataset of the proposed torque model, constants $\tau^j_{f,\text{sensor}}$ and b^j_{sensor} are identified by curve fitting.
- *Position error (PE) based method.* As an extension of [10], the PE-based estimated torque is defined as

$$\hat{\tau}^j_{\text{PE}} = K^j_{\text{PE}}(q^j_{d,i} - q^j_i) - \text{sgn}(\omega^j_i)\tau^j_{f,\text{PE}} + b^j_{\text{PE}}, \tag{26}$$

  where K^j_{PE}, $\tau^j_{f,\text{PE}}$ and b^j_{PE} are constants. The original method assumes that the joint torque is proportional to the joint position error. Similar to the sensor-based baseline, we optimize this method by adding constants $\tau^j_{f,\text{PE}}$ and b^j_{PE} for friction torque and bias. All the constant coefficients in the above equation are obtained by curve fitting using the same training data. For the two baselines, the inertial properties of objects are solved using identity weight matrices.

TABLE II: Error of estimated joint torque. The units of MAE, NMAE and NRMSE are N·mm, % and % respectively. The optimal results are marked in bold.

	Sensor			PE			T model		
	MAE	NMAE	NRMSE	MAE	NMAE	NRMSE	MAE	NMAE	NRMSE
Joint 1	35.12	8.41	11.36	28.66	6.87	8.99	**12.47**	**2.99**	**4.06**
Joint 2	60.83	5.38	6.80	67.07	5.93	7.41	**31.55**	**2.79**	**3.64**
Joint 3	53.94	10.11	12.01	45.58	8.54	10.30	**12.12**	**2.27**	**3.06**
Joint 4	27.93	11.33	13.67	18.81	7.63	9.56	**7.97**	**3.23**	**4.26**
Average	44.45	8.81	10.96	40.03	7.24	9.07	**16.02**	**2.82**	**3.75**

Implementation Details. For each joint, min-max normalization is applied to joint position, so that the normalized joint position is within interval $[0, 1]$. Joint position error and joint torque are scaled down so that the magnitude is less than or equal to 1. In representation learning process, each joint state is embedded using a 2-layer MLP. The 4-dimensional input vectors are embedded as 12-dimensional vectors, and the hidden layer has 12 dimensions. The torque estimator is a 3-layer MLP. The dimension of input layer is 48 and the dimension of output layer is 4. Both the 2 hidden layers have 64 dimensions. The scorer is a 2-layer MLP with 32 neurons in hidden layer. Scalar scores are generated from 16-dimensional embeddings. The above modules are with ReLU non-linearity. All the parameters of models are randomly initialized. The torque model is trained with a batch size of 256 for 300 epochs with an initial learning rate of 0.0003. The attention model is trained with a batch size of 32 for 30 epochs with an initial learning rate of 0.0001. The weights in the loss of attention model are $w_m = 1$ and $w_{\text{com}} = 0.3$. We use Adam optimizer for training.

A. Validation of Inertial Properties Estimation

As shown in Fig. 4(d), we use 4 novel objects to evaluate the proposed framework, which are abbreviated as Cube (41.50g), Red (60.37g), White (88.09g) and Black (138.92g) in the results of experiments. 1000 random samples for each object are collected. Similar to training dataset, 64 samples are randomly selected for a step of inertial properties inference. The testing process is shown in Fig. 1(b). To eliminate the potential bias of random selection, the identification process is repeated 1000 times for each object. Considering the error between networks trained multiple times, we train 10 pairs of torque model and attention model to evaluate the performance. Table I shows the results of mass and COM estimation. It can be seen that all the methods estimate the masses successfully. But the 2 baselines fail to identify the COM. The torque model without attention is able to estimate the COM, and the results are further improved after adding the attention model.

B. Evaluation of Torque Model and Attention Model

Using the random testing samples, we evaluate the accuracy of estimated joint torque. As shown in Table II, all the three methods are capable of estimating joint torque, and the proposed torque model outperforms the two baselines.

In order to evaluate the performance of attention model, we calculate the mean weights of joints for the 4 testing objects, as shown in Table III. Obviously, the weights of joint 4 is much larger than the others. It meets our expectations as the MAE of joint 4 torque is significantly smaller than others. Larger weights prevent the contribution of joint 4 from being ignored. Moreover, as the axes of joints 2, 3 and 4 are parallel to each other, they actually provide similar information for inferring the inertial properties. On the contrary, only the measurements of joint 1 could identify the location of COM along the direction of axes of joints 2, 3 and 4. Therefore, although the error of joint 1 torque is considerable, the weight is still large enough to provide the distinctive information. We can also observe that the weights changes according to the masses of objects. For example, as the object becomes heavier, the weight of joint 4 decreases while the other weights increase. Regarding the torque model without attention, the results of Cube is worse than others. Relatively speaking, the performance of torque model with attention is more even, as the estimation of different objects are of similar accuracy. In summary, the attention model improves the accuracy by adaptively assigning weights to joints, which adjusts the contributions of joints.

TABLE III: Mean weights assigned to joints.

	Joint 1	Joint 2	Joint 3	Joint 4
Cube	0.0399	0.0036	0.0133	0.9432
Red	0.0420	0.0031	0.0133	0.9416
White	0.0456	0.0030	0.0148	0.9365
Black	0.0492	0.0030	0.0176	0.9302

C. Estimating Switching Forces along a Continuous Trajectory

Besides the above testing experiments, we did an experiment in the scenario of continuous trajectories with switching forces. The robot reaches a series of configurations. The measurements are collected at each configuration without stopping. During the motions, the robot carries switching objects, resulting in switching forces at end effector (Fig. 5(a)). We use a single pair of torque model and attention model in this experiment. Due to the proximity of the configurations, the samples provide similar information, which makes it challenging to accurately infer the inertial properties. To address this issue, 128 samples of consecutive configurations are used for each step of inertial properties identification. The vertical force is computed from the estimated mass. And a 128-width mean filter is applied to smooth the estimated force.

Fig. 5(b) shows that the proposed approach is able to estimate the switching forces. The torque model without attention outperforms the baselines, and the attention model further reduces the force error. Parts of the joint torque results are plotted in Fig. 5(c). It can be seen that the proposed torque model is applicable to consecutive postures along a trajectory, and outperforms the baselines.

V. Conclusion

A learning-based approach for estimating inertial properties of unknown objects is proposed in this paper. Without using

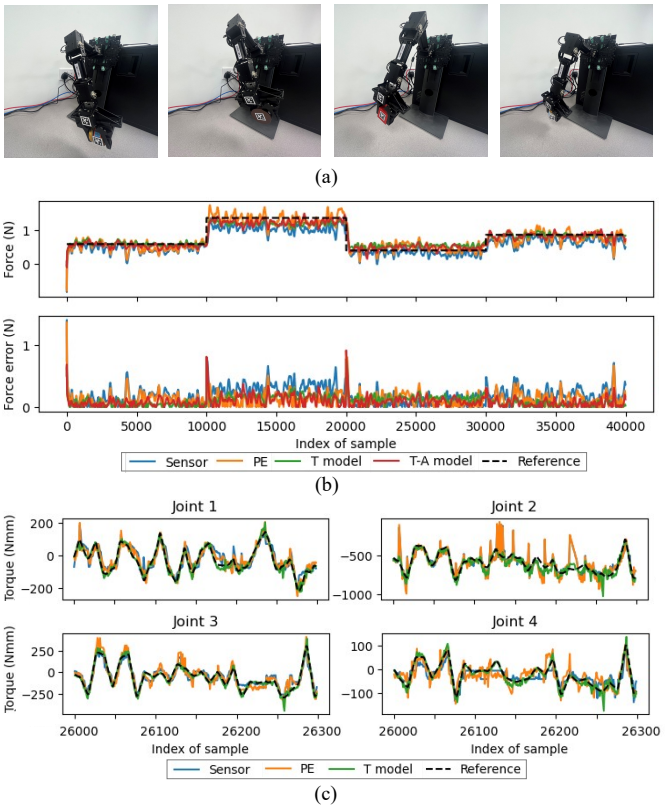


Fig. 5: Results of continuous trajectory experiments. (a) Snapshots of robot carrying objects in experiments. (b) Results and error of force estimation. (c) Results of joint torque estimation.

force/torque sensors, we designed a torque model to reconstruct joint torque from encoder discrepancies. The closed-form relation between joint torque and inertial properties of objects are derived. Given multiple steady-state samples of robot carrying an object, the mass and COM of object can be solved analytically by weighted least squares. To adjust the weight matrix dynamically in the inference process, an attention model is designed to assign weights to joints. The proposed approach is verified in experiments on a 4-DOF robot manipulator. In conclusion, the proposed method achieves relatively accurate estimation of mass and COM without using force/torque sensors.

REFERENCES

[1] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, p. eaat8414, 2019.

[2] M. Suomalainen, Y. Karayiannidis, and V. Kyrki, "A survey of robot manipulation in contact," *Robotics and Autonomous Systems*, vol. 156, p. 104224, 2022.

[3] H. Nguyen and H. La, "Review of deep reinforcement learning for robot manipulation," in *2019 Third IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2019, pp. 590–595.

[4] J. Cui and J. Trinkle, "Toward next-generation learned robot manipulation," *Science robotics*, vol. 6, no. 54, p. eabd9461, 2021.

[5] C. G. Atkeson, C. H. An, and J. M. Hollerbach, "Rigid body load identification for manipulators," in *1985 24th IEEE Conference on Decision and Control*. IEEE, 1985, pp. 996–1002.

[6] Y. Yu, K. Fukuda, and S. Tsujio, "Estimation of mass and center of mass of graspless and shape-unknown object," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 4. IEEE, 1999, pp. 2893–2898.

[7] Y. Han, R. Li, and G. S. Chirikjian, "Can i lift it? humanoid robot reasoning about the feasibility of lifting a heavy box with unknown physical properties," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 3877–3883.

[8] A. C. Smith and K. Hashtrudi-Zaad, "Application of neural networks in inverse dynamics based contact force estimation," in *Proceedings of 2005 IEEE Conference on Control Applications, 2005. CCA 2005.* IEEE, 2005, pp. 1021–1026.

[9] N. Yilmaz, J. Y. Wu, P. Kazanzides, and U. Tumerdem, "Neural network based inverse dynamics identification and external force estimation on the da vinci research kit," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1387–1393.

[10] T. Mattioli and M. Vendittelli, "Interaction force reconstruction for humanoid robots," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 282–289, 2016.

[11] C. Wright, A. Johnson, A. Peck, Z. McCord, A. Naaktgeboren, P. Gianfortoni, M. Gonzalez-Rivero, R. Hatton, and H. Choset, "Design of a modular snake robot," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 2609–2614.

[12] R. W. Hogg, A. L. Rankin, S. I. Roumeliotis, M. C. McHenry, D. M. Helmick, C. F. Bergh, and L. Matthies, "Algorithms and sensors for small robot path following," in *Proceedings 2002 IEEE International Conference on Robotics and Automation*, vol. 4. IEEE, 2002, pp. 3850–3857.

[13] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, "Mechatronic design of nao humanoid," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 769–774.

[14] M.-H. Guo, T.-X. Xu, J.-J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang, R. R. Martin, M.-M. Cheng, and S.-M. Hu, "Attention mechanisms in computer vision: A survey," *Computational Visual Media*, pp. 1–38, 2022.

[15] C. Gaz and A. De Luca, "Payload estimation based on identified coefficients of robot dynamics—with an application to collision detection," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3033–3040.

[16] W. Khalil, M. Gautier, and P. Lemoine, "Identification of the payload inertial parameters of industrial manipulators," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 4943–4948.

[17] M. Capotondi, G. Turrisi, C. Gaz, V. Modugno, G. Oriolo, and A. De Luca, "An online learning procedure for feedback linearization control without torque measurements," in *Conference on Robot Learning*. PMLR, 2020, pp. 1359–1368.

[18] W. Hwang and S.-C. Lim, "Inferring interaction force from visual information without using physical force sensors," *Sensors*, vol. 17, no. 11, p. 2455, 2017.

[19] D. Kim, H. Cho, H. Shin, S.-C. Lim, and W. Hwang, "An efficient three-dimensional convolutional neural network for inferring physical interaction force from video," *Sensors*, vol. 19, no. 16, p. 3579, 2019.

[20] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, 2021.

[21] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[22] Y. Lin, A. S. Wang, E. Undersander, and A. Rai, "Efficient and interpretable robot manipulation with graph neural networks," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2740–2747, 2022.

[23] Z. Wang, C. Liu, and M. Gombolay, "Heterogeneous graph attention networks for scalable multi-robot scheduling with temporospatial constraints," *Autonomous Robots*, vol. 46, no. 1, pp. 249–268, 2022.

[24] Q. Li, W. Lin, Z. Liu, and A. Prorok, "Message-aware graph attention networks for large-scale multi-robot path planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5533–5540, 2021.

[25] J. A. Heredia and W. Yu, "A high-gain observer-based pd control for robot manipulator," in *Proceedings of the 2000 American control conference. ACC*, vol. 4. IEEE, 2000, pp. 2518–2522.

[26] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 3400–3407.