IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

A Smooth Velocity Transition Framework Based on Hierarchical Proximity Sensing for Safe Human-Robot Interaction

Ruohan Wang, Chen Li, Honghao Lv, Gaoyang Pang, Haiteng Wu, Geng Yang, *Member, IEEE*

***Abstract*— With the rapid technology development pushing the introduction of the fifth industrial revolution, Industry 5.0, robots are getting rid of fences and sharing the workspace with humans. In such a context, ensuring the safety of humans and robots is a critical demand. One of the effective methods for this is proximity sensing, among which capacitive sensors are widely used to detect the proximity of humans. However, the capacitive sensor cannot get accurate distance information since the capacitance varies with the characteristics of obstacles. This work develops a capacitive robot skin, seamlessly integrated into the proposed smooth velocity transition framework to deal with this challenge. The robot skin is customized to cover a large area on the exterior of a 6-DOF robot arm. A hierarchical proximity perception approach is used to grade the sensing state. Based on this, distance reduction and collision avoidance velocity generation methods are used to reach a smooth and quick decay of the velocity. The control strategy is applied in a pick-and-place scenery for verification. Compared to the traditional threshold trigger method, the proposed smooth velocity transition framework can greatly reduce the absolute value of the local maximum acceleration, which can enable a flexible and natural human-robot interaction while ensuring human safety.**

***Index Terms*— Human-robot interaction, Industry 5.0, robot skin, the safety of human, smooth velocity transition framework.**

I. Introduction

Nowadays, the manufacturing industry has been facing a new paradigm transformation, from intelligent, flexible, scalable Industry 4.0 [1] to resilient, sustainable, human-centric Industry 5.0 [2]. Human-robot interaction (HRI) has been involved in more and more industry applications to pave the way for smart human-centric manufacturing [3]. Unlike conventional industrial robots working in a confined area with physical separation, cooperative robots are allowed to perform operational tasks in a fence-free environment and share a workspace with human counterparts [4]. Hence, a significant concern in human-robot coexistence is to ensure human safety while keeping the robustness and efficiency of the robot [5].

To address the safety challenges in HRI, the technical specification ISO/TS 15066 [6] was given to provide design guidance for four collaborative operations, i.e., safety-rate monitored stop, hand guiding, speed and separation monitoring, and power and force limiting. As shown in **Fig. 1**, these four collaborative operations can be broadly classified into contact type and non-contact type depending on whether physical contact is exerted between humans and robots. Along with two collaborative operations types, three sensing methods, including contact sensing, proximity sensing, and mid-range and long-range sensing, are given based on the sensing range of a robotic system. For contact sensing, one of the key objectives is to minimize the force exerted between robots and humans in an unintentional collision. Various types of sensors, such as electric current sensors [7], joint torque sensors [8], and tactile sensors [9], are employed to mitigate the potential harm caused by collisions. However, contact sensing is initiated following a physical collision, which may cause pain or other psychological issues. In addition, the currently used rigid robots commonly take post-actions such as emergency stop and compliant control [7] after a collision, which may reduce operation efficiency and position accuracy, respectively. Hence, mid-range and long-range sensing, with the general detection distances above 50 cm, is employed to predict the human approach [10]. The most common solution is using external perceptions like vision [11], radar [12], and acoustic sensors [13]. The monitoring allows the robot to adjust the real-time trajectory using some methods like the least square method [14] and quadratic programming [15]. Although mid-range and long-range sensing can cover a wide workspace surveillance range, blind spots still exist due to the occlusion of the robot by the human body, leading to human approach detection failure [10, 16]. One of the effective methods is deploying multiple sensors, but this will cause an increase in costs.

Proximity sensing, with a detection distance below 50 cm, has been proposed to close the perception gap between mid-range and long-range sensing and contact sensing [10]. Various proximity sensors based on different sensing principles can be found in the literature. Some examples are inductance sensors [17] and capacitive sensors [18, 19]. In proximity sensing applications, capacitive sensors are widely used for HRI due to their high reliability, cost-effective solution, and low-latency measurement [10, 20]. Despite these advantages, the nonlinear relationship between approaching distances and induced capacitance values poses a significant challenge to the accurate measurement of the distance, which is an essential part of the robot to assess the proximity of obstacles. Various parameters, such as spatial, electrical, and material properties of conductive objects, will influence the capacitance values.

This work was supported in part by the National Natural Science Foundation of China (No. 51975513), the Natural Science Foundation of Zhejiang Province, China (No. LR20E050003), the Major Research Plan of National Natural Science Foundation of China (No. 51890884), the Major Research Plan of Ningbo Innovation 2025 (No. 2020Z022), and the Bellwethers Research and Development Plan of Zhejiang Province (No. 2023C01045). (Geng Yang is the corresponding author.)

Ruohan Wang, Chen Li, Honghao Lv, and Geng Yang are with the State Key Laboratory of Fluid Power and Mechatronic Systems, the School of Mechanical Engineering, Zhejiang University, Hangzhou, 310000, China (e-mail: ruohanwang@zju.edu.cn; lichen2021@zju.edu.cn; lvhonghao@zju.edu.cn; yanggeng@zju.edu.cn).

Gaoyang Pang is with the School of Electrical and Computer Engineering, The University of Sydney, Sydney, NSW 2006, Australia (e-mail: gaoyang.pang@sydney.edu.au).

Haiteng Wu is with the Hangzhou Shenhao Technology Co., Ltd., and also with Zhejiang Key Laboratory of Intelligent Operation and Maintenance Robot, Hangzhou, 310000, China (e-mail: wuhaiteng@shenhaoinfo.com)

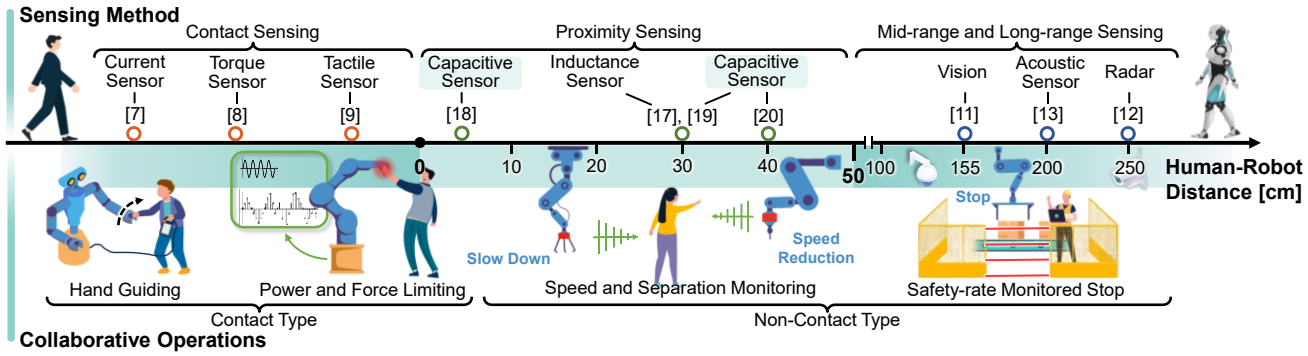

Fig. 1. Three sensing methods supporting the four collaborative operations specified by ISO/TS 15066. In hand guiding, a human operator moves or leads the robot by hand or a handle. In power and force limiting, the robot minimizes the impact force on the human body by tuning operation power. In speed and separation monitoring, the robot monitors the relative speed and distance of a human to maintain a protective separation distance. In safety-rate monitored stop, the robot stops operation while a human operator enters the hazard zone. These operations require different sensing methods, which are classified into three types according to human-robot distances. Proximity sensing, with a detection distance below 50 cm, is used to close the perception gap between mid-range and long-range sensing and contact sensing. Our focus is on proximity sensing using capacitive sensors.

Works have been done to investigate the relationship between these parameters and distance for specified body regions [20], but it requires investigation of all body regions in prior characterization, which is time-consuming. However, only the capacitance variation information can be obtained and used as a trigger condition when implementing capacitive sensors without prior preparation work introduced in [20]. Using capacitance variation to stop the robot's motion often involves a rapid deceleration, causing wear on the actuators and reducing the service life of a robot. Smooth motion is typically deployed in robot control, such as a real-time trajectory planner [21] and a velocity combinator [14]. But the precise distance is commonly required.

In this paper, we developed a robot skin based on the capacitance sensing principle. A novel smooth velocity transition framework is proposed to manage the sensing information from the skin to make robot control decisions for achieving smooth motion and safety purposes, as shown in **Fig. 2**. The main contributions of this letter are: 1) We develop a capacitive robot skin and propose a reliable hierarchical proximity sensing method driven by the relative distance reduction equation, which can act as an alternative solution to obtain a linear and continuous distance reduction between humans and robots in capacitive sensing. 2) We explore an effective collision avoidance velocity (CAV) generator by delicately incorporating the distance reduction equation and robot skin's sensing state, enabling robots to respond quickly, smoothly, and safely when humans approach or leave the robot. 3) The proposed framework is deployed on a dual-arm collaborative robot with our developed capacitive robot skin. The single-joint comparison experiment and task-specific experiment are conducted to evaluate the proposed framework.

The remainder of this article is organized as follows. Section II introduces the capacitive robot skin and the hierarchical perception method based on it. Section III presents the smooth transition method, which can provide dynamic control commands according to the sensing levels. Section IV discusses the results of the single-joint comparison experiment under two conditions. The task-specific experiment is conducted for functional verification. The conclusions are provided in Section V.

[1][Online]. Available: https://www.3m.com/3M/en_US/p/d/b5005474001/

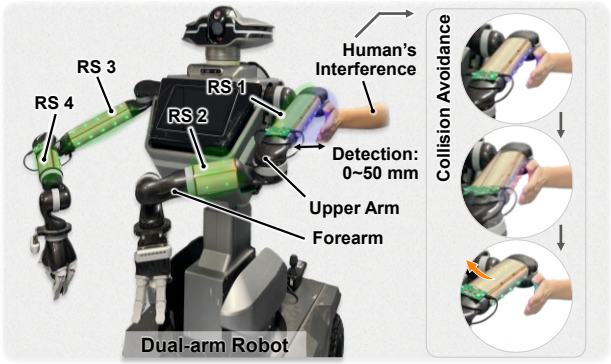

Fig. 2. The robot deploys four pieces of robot skin (RS), which detects the human approach within 50 mm. Active collision avoidance is implemented using the proposed framework according to the sensing states.

II. Hierarchical Proximity Sensing

A. Capacitive Robot Skin

Implementing whole-body proximity sensing can enable a safe physical HRI. To demonstrate this hypothesis, we have developed a robot skin based on the capacitance sensing principle, which uses capacitance changes between the sensor and earth ground to detect the conductive object approach.

As a proximity sensing unit of robot skin, the flexible skin unit (SU) is designed to fit the contours of robot links. It consists of five layers, as illustrated in **Fig. 3**. The insulation layer at the bottom is crafted from polyethylene terephthalate and deployed to safeguard the SU from potential damage during its installation onto the robot. The second layer, made of conductive fabric[1] (5113DFT, 3M, USA) and electroplated copper, is the shielding layer for mitigating electromagnetic interference with a sheet resistance of 1.97 Ω/m. Above the shielding layer, the flexible printed circuit board (FPCB) substrate layer integrated with programmable light-emitting diodes (LEDs, WS2812B, Worldsemi, China) has 0.13 mm thickness and is responsible for electrical connections and real-time visual displays. On the top of the FPCB substrate layer, the sensing layer made of conductive fabric is placed for proximity sensing purposes. The sensing layer is designed to be smaller than the shielding layer, with a 1.5 mm distance difference on each side, to maximize the shielding effect. Another insulation layer is covered on the sensing layer to protect the

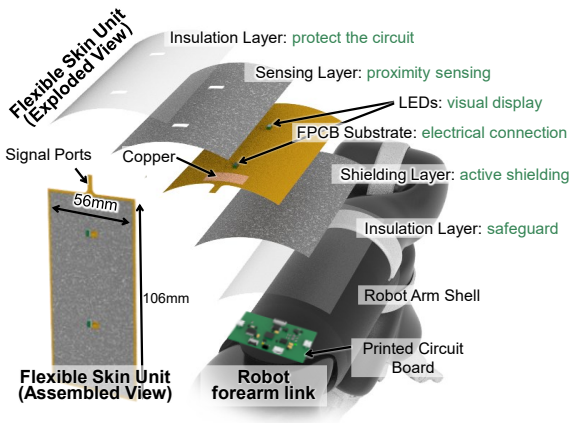

Fig. 3. Diagram of a flexible skin unit (SU) using the five-layer structure and its installation with a printed circuit board on a robot link (exploded view). All the layers are laminated to each other to form an SU (assembled view).

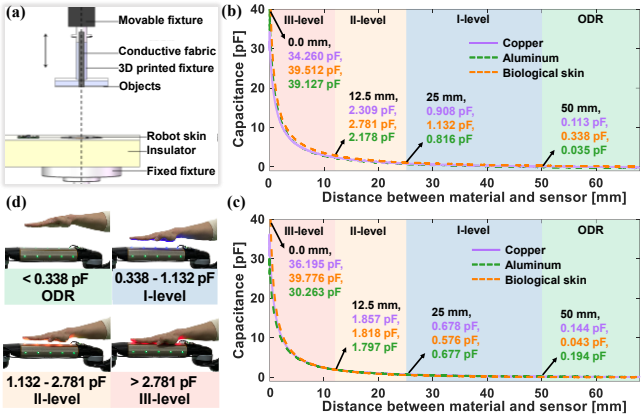

Fig. 4. (a) The calibration experimental setup; The relationship between sensor-material distances and capacitance increments of (b) the SU on the upper arm, and (c) the SU on the forearm; (d) Hierarchy division illustration based on the capacitance increment corresponding to distance ranges.

circuit from the impact of ambient humidity. All the layers are laminated to each other and manufactured by laser cutting and molding. The SU has a total thickness of 0.25 mm.

The four SUs use a proximity capacitive touch sensor controller[2] (MPR121, NXP Semiconductors, Netherlands) to obtain the capacitance values. The signal port at the end of the FPCB substrate is designed for the power supply and signal transmission from the robot skin. The sensing signal is transmitted through the I^2C bus to the microcontroller, which is installed on a 40 mm × 60 mm PCB visible in **Fig. 3**. The system is powered by 5 V, 1.61 W on average. The PCB samples capacitance at a rate of 100 Hz.

On the dual-arm robot, four pieces of robot skin are deployed as illustrated in **Fig. 2**. The size of the robot skin is tailored to be flexible and lightweight to cover a large surface of different robot links. The robot skin for the upper arm and forearm only differs in size, with weights of 23.7 g and 10.6 g, respectively. Each piece of robot skin can be configured by onboard programming via the USB-to-UART bridge. Different pieces of robot skin transmit the sensory data through the controller area network to the master node, which can communicate with the industrial personal computer (IPC).

[1][Online]. Available: https://www.nxp.com/products/no-longer-manufactured/proximity-capacitive-touch-sensor-controller:MPR121

TABLE I. SNR, HYSTERESIS, AND REPEATABILITY ERRORS OF THE SENSOR INDUCED BY DIFFERENT CONDUCTIVE OBJECTS

SU	Material	SNR *at* 50mm (dB)	Hysteresis	Repeatability Errors	
				Loading	Unloading
Upper arm	Aluminum	10.72	3.33%	1.663%	1.775%
	Copper	11.76	2.897%	1.399%	1.703%
	Biological skin	11.19	3.304%	1.843%	1.582%
Fore-arm	Aluminum	10.58	3.597%	2.283%	2.331%
	Copper	11.69	2.994%	1.387%	1.687%
	Biological skin	11.11	3.621%	1.713%	2.328%

B. *Hierarchical Perception*

When a conductive object approaches the capacitive sensor, it interferes with the electric field and changes the capacitance values [21]. Considering the non-linearities posed in capacitive sensing, this study proposes a hierarchical perception method for SU's capacitance values processing.

To evaluate the relationship between the capacitance increment and the distance, a calibration experiment is conducted on a testing machine (34SC-05, Instron, USA) with a process similar to [22], as shown in **Fig. 4(a)**. Conductive materials, including copper, aluminum, biological skin (pig skin), are cut in squares with dimensions of 100 mm × 100 mm × 4 mm and are fixed on the movable side of the system. Two SUs of the upper arm link and forearm link are located on the fixed side and export capacitance values at a rate of 100 Hz. The approaching speed is set to 50 mm/min. **Fig. 4(b)** and **4(c)** illustrate the calibration curves providing the relationship between the sensor-material distances and capacitance. In addition, the hysteresis and repeatability errors of two SUs are evaluated with the three materials using the method presented in [22]. **Table I** illustrates the signal-to-noise ratio (SNR), hysteresis, and repeatability errors of two SUs induced by three conductive objects. The maximum hysteresis and repeatability errors are 3.621% and 2.331% on the forearm when the biological skin and aluminum approach the skin, respectively, indicating a stable reproducibility performance. The maximum detection distance (MDD) of an SU is defined as the largest distance for which the SNR is still greater than 10 dB. Hence, both two SUs can reach a detection distance of 50 mm.

Regarding the safety concern becoming more important as the distance gets smaller, the level division of hierarchical perception is based on the calibration curve shown in **Fig. 4(b)** and **(c)**. When there is no object approaching detected, the sensing state is denoted as being out of detection range (ODR), and the LEDs on the robot skin display a green color. When the distances between conductive objects and SUs are in the three ranges, i.e., 25-50 mm, 12.5-25 mm, and 0-12.5mm, the sensor units will have noticeable capacitance increments (SNR > 10 dB) in these distance ranges as shown in **Fig. 4(b)** and **4(c)**. We use capacitance values induced by biological skin at these distance ranges to define them as perception level I, II, and III, respectively. The LEDs on the sensor will change to blue, yellow, and red, respectively, as shown in **Fig. 4(d)**.

III. SMOOTH TRANSITION SAFE HUMAN-ROBOT INTERACTION

With the robot skin developed in Section II, the robot can perceive the human approach. Using the hierarchical perception strategy, four levels of sensing states can be obtained

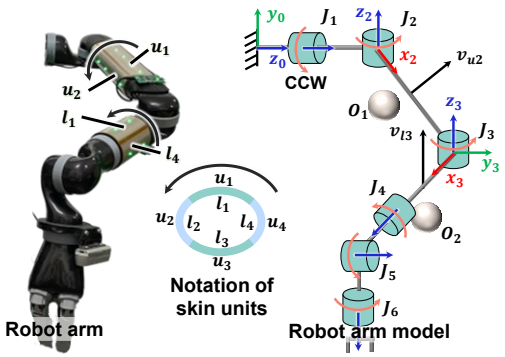

Fig. 5. (left) Sensors' deployment on the robot arm and the notation used for each SU. (right) Mathematic model of the robot arm and the collision avoidance velocity diagrammatic sketch.

according to the change in the induced capacitance. To provide dynamic control commands, this section introduces a multi-level velocity transition strategy by taking detected sensing levels into account.

A. Hierarchical Perception-Based Distance Reduction

To avoid unintended collisions, robots are typically programmed to reduce velocity when objects are perceived at a certain distance. Repulsive action is commonly used to modify the on-line trajectory in collision avoidance [23]. In this work, since accurate distance information cannot be directly gained from the capacitive robot skin, the CAV is generated based on the sensing information from the robot skin. As a conductive object approaches the robot skin, the corresponding SUs will give the current sensing level. Simply assigning each level to a corresponding speed can result in unnatural robot motion due to abrupt changes in distance level. As an example, the signal may change from I-level to II-level, causing a corresponding speed change from 7.5°/s to 15°/s without a smooth transition. Such high-jerk motions have the potential to cause motor wear and decrease the robot's overall operational lifespan. To mitigate the excessive jerk motion, rules of distance reduction are proposed based on the user-defined MDD to give a linear decay of the distance. Inspired by the method to prevent jerky motions [15], a continuous distance reduction D can be obtained from

$$D = D_s\left[\left(d_f - 1\right)\frac{x_f}{x_{max}} + 1\right], \tag{1}$$

where x_{max} is the number of cycles. x_f is a variable that will increase linearly from 0 to x_{max} when the sensing state changes from a low to a high level. x_f will decrease linearly from x_{max} to 0 when the sensing state changes from a high to a low level. At the same cycle frequency, the smaller x_{max} will cause a faster linear decay of the distance. D_s is the initial distance for reduction within the user-defined MDD. $d_f \in [0,1]$ is a factor to adjust the lower value of D. Larger D_s and smaller d_f result in a quicker distance reduction within the same time. Additionally, due to the limited sampling frequency of the robot skin, the sensing state may skip the I and II-level and change directly from ODR to III-level when the object's moving speed exceeds 3 m/s. This occurs when there is a sudden drop in distance. We handle this situation using the same method as it changes from II-level to III-level. All the user-defined parameters are listed in **Table II** with different sensing conditions.

TABLE II. USER-DEFINED PARAMETERS USED IN THE FUNCTION

Parameter	Value at the change of sensing state		
	ODR to I	*I to II and II to III*	*ODR to II and II to III, or ODR to III*
D_s	5 cm	2.5 cm	1.25 cm
d_f	0.5	0.5	0
x_f	80 cycles/ 100 Hz	80 cycles/ 100 Hz	100 cycles/ 100 Hz
x_{max}	80 cycles	80 cycles	100 cycles
α	10	4	6
s	3.3	3.1	1.2
D_0	2.5	1.25	1.25
v_{d_max}	--	20°/s	20°/s

B. Collision Avoidance Velocity Generation

The CAV serves as a critical link between object perception and safe interactive motion control. It takes the distance between SU and the conductive object as input and guides the interactive motion control process. In this work, the CAV for each link is generated based on the distance information from Section III. A. **Fig. 5** illustrates the notation used for each SU deployed on the robot link. Each SU is denoted as u_i and $l_i, \forall i \in \{1,2,3,4\}$ for the upper arm link and forearm link respectively. In **Fig. 5**, the u_2 SU detects the object O_1's approach to the upper arm link, causing the counterclockwise (CCW) motion of joint 2 (J_2) with the velocity v_{u_2}. In the same manner, v_{l_3} is caused by the object O_2's approach to the l_3 SU. The CAV generated by each SU (i.e., v_{u_i} and v_{l_i}) is actuated by the corresponding joints. In **Fig. 5**, the clockwise (CW) and CCW rotation of joint 1 (J_1) represent the CAV v_{u3}, v_{l3} and v_{u1}, v_{l1}, respectively. The CW, CCW rotation of joint 2 (J_2) and joint 3 (J_3) represent the v_{u4}, v_{l4} and v_{u2} and v_{l2}, respectively.

Inspired by [24], defining the maximum joint velocity as v_{max}, v_{rep} is a general notation for the CAV generated by each SU, and can be calculated as

$$v_{rep} = v_{d\_max}\frac{1 + e^{-\alpha}}{1 + e^{\left(2\frac{D}{D_0} - s\right)\alpha}}, \tag{2}$$

where $v_{d\_max} \in [-v_{max}, v_{max}]$ corresponds to the maximum desired magnitude of the CAV. D_0 represents the distance threshold, which is the upper distance limit of the current sensing level. s and α are the parameters of the curve. s represents the translation along the abscissa, and α is the gradient changing from 0 to v_{d_max}.

For changes to different sensing levels, the parameters of D and v_{rep} are different. When the robot is moving according to its predefined tasks, the joint velocity recorded before the sensing level change is denoted as v_{last}. After the change of sensing level, the desired velocity v_d given to the relative link is expressed as

$$v_d = \begin{cases} v_{last}\left(1 - \dfrac{v_{rep}}{v_{d\_max}}\right), \textit{ODR to I}, \\ v_{last}\left(1 - \dfrac{v_{rep}}{v_{d\_max}}\right) - v_{rep}, \textit{others}. \end{cases} \tag{3}$$

When the human moves away from the robot, the variable D can be determined using the same method as equation (2). The velocity of the relative link can be expressed as

$$v_d = \frac{v_{last}}{v_{d\_max}} v_{rep}. \quad (4)$$

When a human approaches the robot arm with a velocity lower than 3 m/s, the SU detects it and gives the signal that shows the sensing state change. From ODR to I-level, the robot reduces its speed to avoid the potential collision. If the human continues to approach, causing the sensing states to change from I to II and from II to III-level, the robot continues to reduce speed and finally moves in the opposite direction with respect to the human's approaching side. When the human's approaching velocity exceeds 3 m/s (i.e., when the sensing state changes from ODR to II/III-level), the robot's velocity will quickly decrease to 0 and increase to v_{d_max} for safe interaction. When the human moves away at any sensing state other than ODR, the robot slows to a stop and then accelerates for the target task to avoid discontinuity in acceleration. The parameters in the previous equations can be found according to different sensing states' changing in **Table II**.

C. Motion Control Strategy

Assuming the robot is performing predefined tasks like picking and putting objects, two goals are supposed to be achieved when it interacts with a human. The first is to control the end-effector (EE) to the target pose $\boldsymbol{x_d} = [\boldsymbol{p_d}, \boldsymbol{q_d}] \in \mathbb{R}^6$, where $\boldsymbol{p_d} = [x_d, y_d, z_d]^T$ and $\boldsymbol{q_d} = [\gamma_d, \beta_d, \psi_d]^T$ represent the desired EE position and orientation, respectively. The second is interacting with humans safely based on the CAV v_{rep}. As mentioned in Section III.B, the CAV acts on each link, hence, the joint velocity control method is deployed in the robot arm motion control. The general relationship between the EE velocity $\dot{\boldsymbol{x}}_{\boldsymbol{ee}}$ and task joint velocity $\dot{\boldsymbol{\theta}}_{\boldsymbol{d}}$ is as

$$\dot{\boldsymbol{\theta}}_{\boldsymbol{d}} = J^{-1}\dot{\boldsymbol{x}}_{\boldsymbol{ee}}, \quad (5)$$

where $\boldsymbol{\theta_d} = [\theta_1, \theta_2, \cdots, \theta_6]^T$ is the matrix of 6-DoF robot joint angles. $\boldsymbol{x_{ee}}$ is the EE pose in the cartesian space. $J \in \mathbb{R}^{6\times6}$ is the full-rank Jacobian matrix of the robot. When the robot interacts with humans, the EE position may not be maintained at the target position.

To get $\dot{\boldsymbol{x}}_{\boldsymbol{ee}}$, a PD controller is used to calculate the desired EE velocity

$$\dot{\boldsymbol{x}}_{\boldsymbol{ee}} = K_p \boldsymbol{e} + K_d \frac{d\boldsymbol{e}}{dt}, \quad (6)$$

where $\boldsymbol{e} = \left[\boldsymbol{e_p}, \boldsymbol{e_q}\right]^T \in \mathbb{R}^6$ represents the error between the current pose and the target pose. $\boldsymbol{e_p} = \boldsymbol{p_d} - \boldsymbol{p_c}$ is the position error of the current EE position $\boldsymbol{p_c} = [x_c, y_c, z_c]^T$ relative to the target position $\boldsymbol{p_d}$. $\boldsymbol{e_q}$ is the orientation error calculated using the Euler angles differential equation [25]

$$\boldsymbol{e_q} = \begin{bmatrix} 1 & 0 & -\sin\beta \\ 0 & \cos\gamma & \sin\gamma\cos\beta \\ 0 & -\sin\gamma & \cos\gamma\cos\beta \end{bmatrix} \begin{bmatrix} \gamma \\ \beta \\ \psi \end{bmatrix}, \quad (7)$$

where $[\gamma \quad \beta \quad \psi]^T$ is the roll-pitch-yaw angle differences of the current EE orientation $\boldsymbol{q_c} = [\gamma_c, \beta_c, \psi_c]^T$ relative to the desired EE orientation $\boldsymbol{q_d}$.

Algorithm 1 shows the detailed procedures of the motion control strategy based on the predefined task and sensing state. As shown in **Algorithm 1(i)**, the current state of the robot's EE, the target pose, and the sensing state are obtained at the beginning of a control cycle. **Algorithm 1(ii)** indicates that the robot is driven by the joint velocity to the target position. s_{ui} and $s_{li}, \forall i \in \{1,2,3,4\}$ are the sensing state of SUs on the upper arm and forearm links, respectively. As described in **Algorithm 1(iii)**, after the change of sensing state, equations (1) and (2) are used to determine the values in equation (3), which is used to get the desired velocity generated by each SU (i.e., $v_{d\_u_i}$ and $v_{d\_l_i}$, $\forall i \in \{1,2,3,4\}$). After the combination of the desired velocity and task joint velocity, the velocity command is sent to control the robot.

Algorithm 1: Motion Control Strategy Based on the Predefined Tasks and Sensing States Using Our Robot Platform

Input: Target pose from predefined tasks and sensing states from robot skin on robot links
Output: Joint velocity for robot joint motion control
while not (termination condition satisfied) **do**
  (i) get current state of the robot's end-effector, the target poses of the tasks, and the sensing state
  $\boldsymbol{p_c} \leftarrow (x_c, y_c, z_c)$, $\boldsymbol{q_c} \leftarrow (\gamma_c, \beta_c, \psi_c)$
  $\boldsymbol{p_d} \leftarrow (x_d, y_d, z_d)$, $\boldsymbol{q_d} \leftarrow (\gamma_d, \beta_d, \psi_d)$
  $s_{ui} \leftarrow$ I or II or III level of upper arm
  $s_{li} \leftarrow$ I or II or III level of forearm
  (ii) generate the joint velocity based on tasks
  if (grasp-put task condition satisfied) **then**
    $\boldsymbol{e_p} \leftarrow \boldsymbol{p_c}, \boldsymbol{p_d}$ and $\boldsymbol{e_q} \leftarrow \boldsymbol{q_c}, \boldsymbol{q_d}$
    $\dot{\boldsymbol{x}}_{\boldsymbol{ee}} \leftarrow K_p \boldsymbol{e} + K_d \frac{d\boldsymbol{e}}{dt}$
    $\dot{\boldsymbol{\theta}}_{\boldsymbol{d}} \leftarrow J^{-1}\dot{\boldsymbol{x}}_{\boldsymbol{ee}}$
  (iii) update the joint velocity based on sensing states
  if (s_{ui} is not ODR **or** s_{li} is not ODR) **then**
    $\dot{\theta}_1 \leftarrow v_{d\_u_1} + v_{d\_u_3} + v_{d\_l_1} + v_{d\_l_3}$
    $\dot{\theta}_2 \leftarrow v_{d\_u_2} + v_{d\_u_4}$
    $\dot{\theta}_3 \leftarrow v_{d\_l_2} + v_{d\_l_4}$
    $\dot{\boldsymbol{\theta}}_{\boldsymbol{d}} \leftarrow [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4, \dot{\theta}_5, \dot{\theta}_6]^T$
  (iv) send the joint velocity command
  joint velocity publisher $\leftarrow \dot{\boldsymbol{\theta}}_{\boldsymbol{d}}$
end

IV. Experiments And Results

A. Experimental Setup

To validate the effectiveness of the proposed method, experiments are performed using a robot with two 6-DoF arms (Jaco 2, Kinova, Canada). Eight SUs are mounted on the upper arm link and forearm link for each arm. The sampling rate for collecting capacitance data from all sensors is set to 100 Hz. A computer with a real-time kernel sends the joint velocity commands at the rate of 100 Hz. The maximum angular velocity of joints is 36°/s for big actuators (J_1, J_2, J_3 in **Fig. 5**) and 48°/s for small actuators (J_4, J_5, J_6 in **Fig. 5**). The user-defined MDD of the SUs is set to 50 mm. The experiment is developed using C++, python, and ROS [26]. Velocity commands are given to individual joints in both the single-joint comparison experiment and the task-specific experiment. For the multiple SUs' activation case, the CAV is the vector sum of the avoidance velocities generated by SUs. In the single-joint comparison experiment, only joint 2 is used with

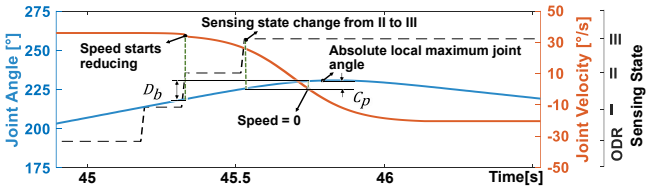

Fig. 6. Illustration of the evaluation metrics in repetitive experiment.

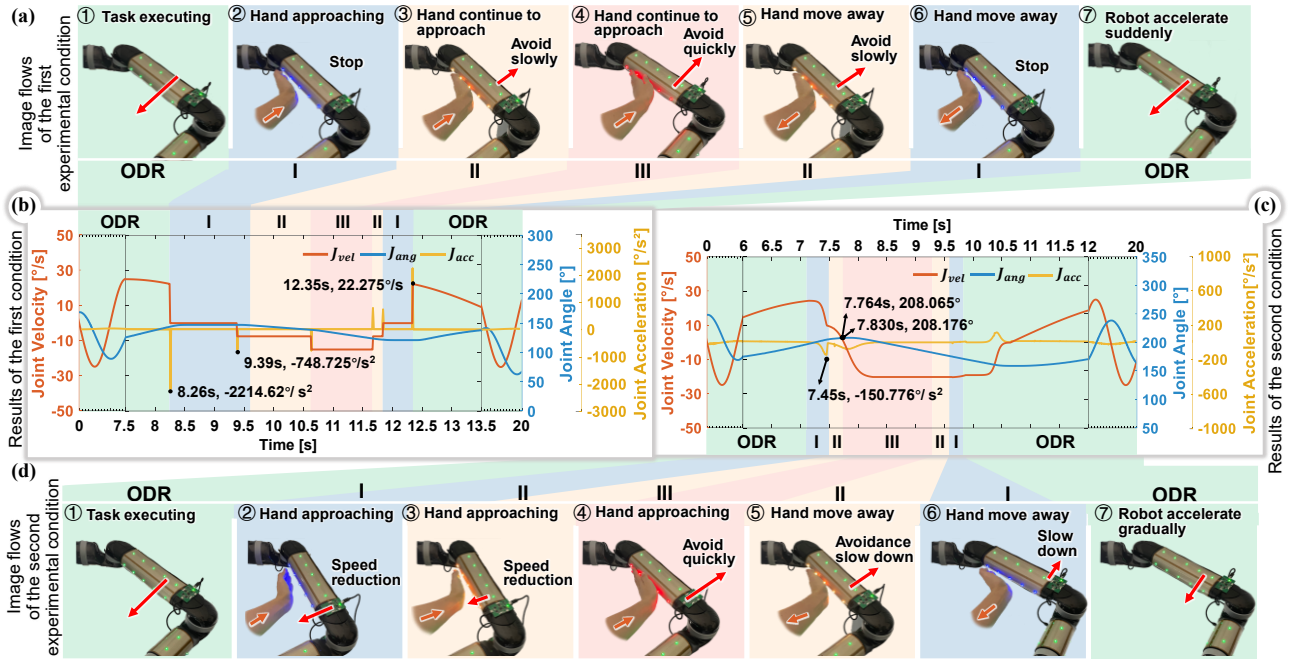

Fig. 7. Single-joint comparison experiment: (a) Image flows of the first experimental condition using traditional threshold trigger method, (b) The curves of joint angle, joint velocity, and joint acceleration with time of the first condition, (c) The curves of join angle, joint velocity, and joint acceleration with time of the second experimental condition using the proposed smooth velocity transition framework, (d) Image flows of the second condition. Variables J_{vel}, J_{ang}, and J_{acc} represent the joint velocity, joint angle, and joint acceleration, respectively.

the task velocity $\dot{\theta}_2$ varying with time t: $\dot{\theta}_2 = 25sin\,(2\pi t)$. For repetitive experiments, joint 2 is given the maximum joint velocity allowed by its actuator. In the task-specific experiment, the linear motion of the robot EE is deployed in dynamic planning to the desired position. The parameters K_p and K_d are increased linearly from 0 to 2 and from 0 to 0.5 in 1000 program loops. Motion control of the robot arm follows **Algorithm 1.** The interactions were performed by the same individual who placed the palm along the path of the robot. The user-defined parameters used in the experiments are listed in **Table II**. Safety checks were conducted before experiments (details refer to the attached video). In particular, following the protective separation distance used in [27], the maximum approaching speed of the human body part is suggested lower than 11.318 cm/s when the robot is operating at its maximum speed. Furthermore, the speed and power of the robot compliant with the power and force limiting in case of contact with the human operator.

B. Evaluation Metrics

To evaluate the performance of the proposed framework, metrics are given for results analysis. Inspired by the stopping distance in ISO/TS 15066 [6], the full braking angle D_b is used for single-joint performance evaluation. It is defined as the angle increment from where the robot starts slowing down towards its 0 speed, as shown in **Fig. 6**. Critical braking angle C_p is another evaluating parameter introduced in this work. It is defined as the angle increment from the transition of the sensing state from II to III-level towards the absolute local maximum joint angle, which is the joint angle that reaches its local maximum or minimum value after the robot skin senses the human approach. The probability of a collision increases as these two angles become greater. For the repetitive experiment, the average values and standard deviation of the absolute local maximum joint angle, full braking angle, and critical braking angle are denoted as $\bar{J}_{amax}$ and SD_{ja}, $\bar{D}_b$ and SD_{Db}, and $\bar{C}_p$ and SD_{Cp}, respectively.

C. Single-Joint Comparison Experiment

The comparative experiment was performed using joint 2, which moved at a given speed $\dot{\theta}_2$. As shown in **Fig. 7(a)** and **7(d)**, two conditions were considered to avoid the human hand using different interaction frameworks. The first condition used the traditional threshold-trigger method, which involved giving a specific joint velocity according to the relative link's sensing level. The selected values for joint velocity were 0, 7.5°/s, and 15°/s when the sensing state was at I, II, and III-level, respectively, and its direction was opposite to the direction of approaching. The second condition used the proposed smooth velocity transition strategy for different levels.

Fig. 7(b) and **7(c)** show the changes in joint angle, joint velocity, and joint acceleration with time for two conditions. The color bars in the figures represent the corresponding sensing states. Note that the human operator can touch the robot link in III-level. The results of the first condition are shown in **Fig. 7(b)**. At around 8.26 s, the human hand enters the detection range, causing a signal change from ORD to I-level. This leads to a sudden change of joint velocity, resulting in a jerk of joint acceleration, which reaches −2214.62°/s² in a short time. It indicates that the use of the threshold-trigger method may lead to unnatural and inflexible HRI because of the high jerk in acceleration caused by sensing signal changes. The results of the second condition are presented in **Fig. 7(c).** The absolute value of joint acceleration starts to increase at the time when the sensing state changes from ODR to I-level, and it reaches the maximum value of −150.776°/s² at 7.45 s due to the exponential decay of the given velocity. It shows a more flexible HRI framework compared to the first condition.

Benefiting from the reduced high jerk, the more natural performance of the robot is shown in the proposed HRI stage

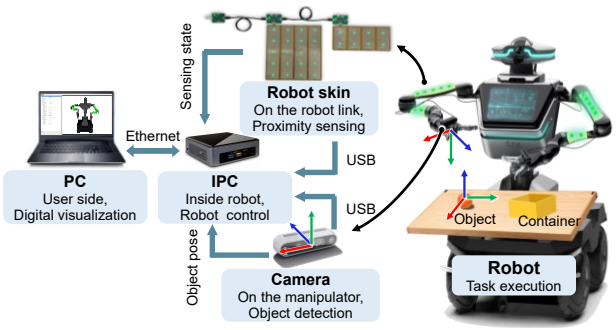

Fig. 8. The communication architecture in the pick-and-place task-based experiment scenario.

when people are getting away from the robot. In **Fig. 7(b)**, when the human moves away from the robot (i.e., from III to ODR), the joint speeds up to 22.275°/s^2 within one second. For people who move away slowly, there is a potential for the robot to catch up to them, resulting in a collision. In **Fig. 7(c)**, under the same situation, the robot keeps moving away till the joint speed is reduced to 0, and then slowly speeds up for the task. Hence, to this extent, the proposed method is safe and provides more natural HRI.

Although the proposed smooth transition method reduces the maximum acceleration, D_b increases compared to the traditional method. To evaluate human safety after increasing the braking angle, experiments were repeated ten times giving the joint 2 at a constant speed of 36°/s. The human was trained in advance to put the palm at a distance further than MDD on the path of the robot arm, which is in front of u_2 SU in this experiment. Table III presents the average values and standard deviations of the evaluation metrics. Note that condition 1 does not require estimation on C_p because the local maximum/ minimum joint angle always occurs before the transition joint angle. In condition 1, the robot will not collide with humans if the human does not actively approach it with $\bar{D}_b$ = 6.14°. As shown in Table III, condition 2 achieved a great reduction on $\bar{J}_{amax}$ with a small increment of $\bar{D}_b$. The $\bar{C}_p$ in condition 2 is 5.22°, which is acceptable in guaranteeing human safety as there is still a certain human-robot distance after reaching the III-level. Furthermore, the predefined value of x_f in equation (1) can be modified in practical application. A smaller x_f value results in faster velocity reduction, leading to a larger $\bar{J}_{amax}$ and a smaller $\bar{C}_p$. Overall, this is a trade-off in braking angle and joint acceleration.

D. Task-Specific Experiment

The task-specific demonstration is given to verify the entire framework, aiming to create a scenario corresponding to an unknown conductive object interface in the workspace of a robot in the industrial setting. Specifically, the robot was supposed to finish the pick-and-place tasks under the random dynamic interference of the human in this demonstration.

TABLE III. EVALUATION METRICS VALUES IN TWO CONDITIONS

Condition	The Average Value of Parameters					
	$\bar{J}_{amax}$ (°/s^2)	SD_{Ja} (°/s^2)	$\bar{D}_b$ (°)	SD_{Db} (°/s^2)	$\bar{C}_p$ (°)	SD_{Cp} (°/s^2)
1	1203.33	3.60	6.14	0.17	---	---
2	127.10	1.98	14.90	1.26	5.22	0.70

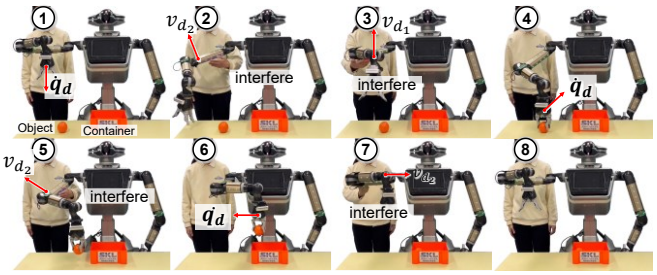

Fig. 9. Image flows of human-robot interaction under the random dynamic interference of human using smooth velocity transition method in a pick-and-place scenario.

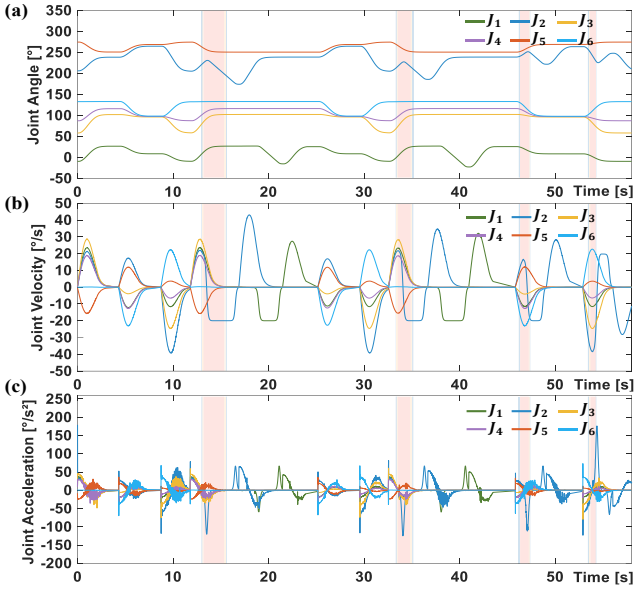

Fig. 10. The robot performance of each joint in the task-specific experiment: (a) The curves of joint angle with time, (b) The curves of joint velocity with time, (c) The curves of joint acceleration with time. $J_1, J_2, \cdots, J_6$ represent joint 1 to joint 6. The red shade indicates the robot skin sensing the human approach.

The communication architecture of the system is shown in **Fig. 8**. The robot skin, robot arm, and camera (Realsense D435, Intel, USA) were connected to the IPC inside the robot through USB, whereas the personal computer (PC) was connected to the IPC via Ethernet. The PC was used on the user side to visualize the robot model in ROS. Considering the reliability and accuracy, the object detection and location were carried out based on the methods [28, 29]. Obstacle sensing and robot control were realized by different nodes' communicating through topics. In addition, object detection and location functions are clients that should operate when a service in the main function poses a request. The personal computer was used to monitor the sensing state and set robot control commands. The control frequency of the final output was 100 Hz.

Fig. 9 shows the robot movements during the task and the HRI. At the initial position, the camera on the robot hand detected the object and sent the object's pose information to the robot. The object placement position was pre-defined. The robot control strategy was introduced in Section II-C, and the CAV was generated in the same way as in the single-joint comparison experiment. Note that x_f was 100 cycles/100 Hz when the sensing states changed from ODR to I-level and I to II/III-level, and it was 120 cycles/100 Hz when the sensing states changed from ODR to II/III level. Additionally, the

value of K_p in equation (6) varies from 0 to 10 when executing specified tasks. The robot grasped the object, reached the placing position, released the gripper, and went back to the initial position. The robot performed the tasks continuously and interacted with humans using the smooth velocity transition framework till all the objects were placed in the container. At that time, there was no object in the camera's view, and therefore, it could not provide the object's pose information. **Fig. 10** shows the corresponding changes in the related angle, velocity, and acceleration of each joint. The red color bars in **Fig. 10** represent the human interferences, that cause sensing state changes on u_3, l_3, u_4, u_2, sequentially. From **Fig. 10(b)**, it can be seen that the curves of velocity are smooth and continuous. In **Fig. 10(c)**, the local maximum acceleration became smaller compared to the single-joint experiment due to the increment of x_f. Thus, the proposed real-time HRI framework can be applied with different user-defined parameters depending on the situation.

V. Conclusion

In this work, a novel smooth velocity transition framework is proposed for safe HRI. The robot skin employs capacitance sensing principles for human approaching detection. In consideration of human safety and the robot arm's smooth motion, the hierarchical sensing method is developed using the capacitance variation to compensate for the non-linearities of the capacitive robot skin. A CAV generator has been designed, incorporating the hierarchical sensing method into the robot control. The proposed framework has been validated through two experiments. The single-joint comparison experiment showcased the framework can greatly reduce the absolute local maximum value of joint acceleration. The pick-and-place experiment presented the robot can perform the tasks under random dynamic interference of humans. Overall, the results show the potential application of the proposed framework in safe HRI.

A drawback of this work is that the developed robot skin can only detect the conductive elements. Hence, working close to conductive elements may cause false recognition of the human being. In addition, the human wearing gloves and suits made of insulators may not be effectively detected.

Concerning future work, experiments will be conducted to further assess the detection reliability of the robot skin. The performance of the robot skin will be further evaluated with the presence of conductive materials in its surroundings.

References

[1] A. G. Frank, L. S. Dalenogare, and N. F. Ayala, "Industry 4.0 technologies: Implementation patterns in manufacturing companies," *Int. J. Prod. Econ.,* vol. 210, pp. 15-26, 2019.

[2] J. Leng *et al.*, "Industry 5.0: Prospect and retrospect," *J. Manuf. Syst.,* vol. 65, pp. 279-295, 2022.

[3] A. Hentout, M. Aouache, A. Maoudj, and I. Akli, "Human–robot interaction in industrial collaborative robotics: A literature review of the decade 2008–2017," *Adv. Robot.,* vol. 33, no. 15-16, pp. 764-799, 2019.

[4] R. Galin and R. Meshcheryakov, "Review on human–robot interaction during collaboration in a shared workspace," in *Int. Conf. Interact. Collab. Robot.*, Cham, 2019.

[5] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics,* vol. 55, pp. 248-266, 2018.

[6] *Robots and robotic devices—Collaborative robots (ISO/TS 15066: 2016)*, International Organization for Standardization, 2016.

[7] L. Han, W. Xu, B. Li, and P. Kang, "Collision detection and coordinated compliance control for a dual-arm robot without force/torque sensing based on momentum observer," *IEEE/ASME Trans. Mechatron.,* vol. 24, no. 5, pp. 2261-2272, 2019.

[8] J. Bimbo, C. Pacchierotti, N. G. Tsagarakis, and D. Prattichizzo, "Collision detection and isolation on a robot using joint torque sensing," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Macau, 2019.

[9] G. Pang *et al.*, "CoboSkin: Soft robot skin with variable stiffness for safer human–robot collaboration," *IEEE Trans. Ind. Electron.,* vol. 68, no. 4, pp. 3303-3314, 2021.

[10] S. E. Navarro *et al.*, "Proximity perception in human-centered robotics: A survey on sensing systems and applications," *IEEE Trans. Robot.,* vol. 38, no. 3, pp. 1599-1620, 2022.

[11] L. M. Amaya-Mejía, N. Duque-Suárez, D. Jaramillo-Ramírez, and C. Martinez, "Vision-based safety system for barrierless human-robot collaboration," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Kyoto, 2022.

[12] M. Geiger and C. Waldschmidt, "160-GHz radar proximity sensor with distributed and flexible antennas for collaborative robots," *IEEE Access,* vol. 7, pp. 14977-14984, 2019.

[13] T. Schlegl, T. Bretterklieber, M. Neumayer, and H. Zangl, "Combined capacitive and ultrasonic distance measurement for automotive applications," *IEEE Sens. J.,* vol. 11, no. 11, pp. 2636-2642, 2011.

[14] H. C. Lin, Y. X. Fan, T. Tang, and M. Tomizuka, "Human Guidance Programming on a 6-DoF Robot with Collision Avoidance," *IEEE/RSJ Int. Conf. Intell. Robot. Syst.,* pp. 2676-2681, 2016.

[15] C. Escobedo, M. Strong, M. West, A. Aramburu, and A. Roncone, "Contact anticipation for physical human–robot interaction with robotic manipulators using onboard proximity sensors," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Prague, 2021.

[16] F. Mohammadi Amin, M. Rezayati, H. W. van de Venn, and H. Karimpour, "A mixed-perception approach for safe human–robot collaboration in industrial automation," *Sensors,* vol. 20, no. 21, pp. 6347-6366, Jul. 2020.

[17] T. D. Nguyen, T. Kim, J. Noh, H. Phung, G. Kang, and H. R. Choi, "Skin-type proximity sensor by using the change of electromagnetic field," *IEEE Trans. Ind. Electron.,* vol. 68, no. 3, pp. 2379-2388, 2021.

[18] D. Göger, H. Alagi, and H. Wörn, "Tactile proximity sensors for robotic applications," in *IEEE Int. Conf. Ind. Technol.*, Cape Town, 2013.

[19] K. E. M'Colo, B. Luong, A. Crosnier, C. Neel, and P. Fraisse, "Obstacle avoidance using a capacitive skin for safe human-robot interaction," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Macau, 2019.

[20] S. Ergun *et al.*, "A unified perception benchmark for capacitive proximity sensing towards safe human-robot collaboration (HRC)," in *IEEE Int. Conf. Robot. Autom.*, Xi'an, 2021.

[21] S. J. Moon, J. Kim, H. Yim, Y. Kim, and H. R. Choi, "Real-time obstacle avoidance using dual-type proximity sensor for safe human-robot interaction," *IEEE Robot. Autom. Lett.,* vol. 6, no. 4, pp. 8021-8028, 2021.

[22] Z. V. Gbouna *et al.*, "User-interactive robot skin with large-area scalability for safer and natural human-robot collaboration in future telehealthcare," *IEEE J. Biomed. Health Inform.,* vol. 25, no. 12, pp. 4276-4288, 2021.

[23] F. Flacco, T. Kroger, A. De Luca, and O. Khatib, "A depth space approach to human-robot collision avoidance," in *IEEE Int. Conf. Robot. Autom.*, Saint Paul, MN, 2012.

[24] C. Escobedo, N. Nechyporenko, S. Kadekodi, and A. Roncone, "A Framework for the systematic evaluation of obstacle avoidance and object-aware controllers," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Kyoto, 2022.

[25] Y. Zhu, C. Yang, Z. Tu, Y. Ling, and Y. Chen, "A haptic shared control architecture for tracking of a moving object," *IEEE Trans. Ind. Electron.,* vol. 70, no. 5, pp. 5034-5043, 2023.

[26] M. Quigley, "ROS: an open-source Robot Operating System," in *IEEE Int. Conf. Robot. Autom.*, Kobe, 2009.

[27] L. Scalera, R. Vidoni, and A. Giusti, "Optimal scaling of dynamic safety zones for collaborative robotics," in *IEEE Int. Conf. Robot. Autom.*, Xi'an, 2021.

[28] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *ArXiv,* vol. abs/2004.10934, 2020.

[29] W. Liu *et al.*, "SSD: Single shot multiBox detector," in *Eur. Conf. Comput. Vis.*, Cham, 2016.