Adaptive technique for physical human–robot interaction handling using proprioceptive sensors

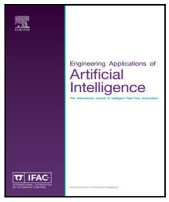
Dmitry Popov, Anatol Pashkevich, Alexandr Klimchik

Engineering Applications of Artificial Intelligence 126 (2023) 107141

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Adaptive technique for physical human–robot interaction handling using proprioceptive sensors

Dmitry Popov [a,b], Anatol Pashkevich [a,b], Alexandr Klimchik [c,*]

[a] IMT-Atlantique, Nantes, France
[b] Laboratoire des Sciences du Numérique de Nantes (LS2N), Nantes, France
[c] University of Lincoln, Lincoln, UK

ARTICLE INFO

Keywords:
Human–robot collaboration
Physical human–robot interaction
Interaction handling
Adaptive robot behavior
Interaction classification

ABSTRACT

The work focuses on the development of an adaptive technique for the physical interaction handling between a human and a robot, as well as its experimental validation. The proposed technique is based on the deep residual neural network and dedicated finite state machine, where the states are the robot behavior modes and transitions are the switchings between the states that depend on the interaction parameters and characteristics. It ensures the human operator safety and improves the human–robot collaboration performance by implementing various scenarios. In the scope of this technique, the parameters of human–robot interaction are used to select an appropriate robot reaction strategy using data from internal robot sensors only, i.e. proprioceptive sensors. These parameters define the interaction force vector and its application point on the robot surface, which allow to classify the interaction within the set of predefined categories. This classification distinguishes interactions applied at the tool or intermediate link (Tool/Link), having soft or hard nature (Soft/Hard), as well as having different intention (Intl/Accd) or duration (Short/Long). Based on identified category and the current robot state, the algorithm chooses an appropriate robot reaction. To confirm the efficiency the developed technique, an experimental study was conducted, which involved the collaboration between the real industrial manipulator KUKA LBR iiwa and the human operator.

1. Introduction

The field of robotics is advancing rapidly and, as robots become more prevalent in various industries, it is becoming increasingly important to ensure safe physical interaction between humans and robots. However, ensuring human safety is still a challenging issue, since most industrial robots can achieve high speeds and forces, which could be dangerous for a human (Falco et al., 2012; Zanchettin et al., 2016). Although currently there exist a number of collision avoidance techniques, sometimes a collision between a human and a robot is unavoidable or even it is a necessary element of the manufacturing process. To ensure the safety in practice, all possible physical interactions should be properly identified and handled by an appropriate robot reaction that can vary from a simple emergency stop to more complicated scenarios, like moving away from the obstacle or lowering robot joint stiffness. It is clear that to implement any of these scenarios, the human–robot interaction force as well as its origin and some properties should be also properly estimated in order to produce relevant robot reaction.

The problem of the human–robot interaction handling was the focus of the robotics community for several years (Fryman and Matthias, 2012). Some early works in this field (Heinzmann and Zelinsky, 1999; Yamada et al., 1997) discussed requirements for the robot mechanics and its control systems in order to provide safe interaction with a human. They highlighted the importance of the robot actuator's backdrivability and the limitation of robot motor torques while contacting with a human and switching the robot controller to the so-called external force compensation mode. Some of the recent works proposed more sophisticated approaches for the interaction handling, which operate with several possible robot reactions depending on the robot state. In particular, Haddadin et al. (2008) and Lippi and Marino (2020) used the interaction handling pipeline, in which some properties of interaction were used to select safe robot behavior on a high level. For such complex robot behavior, a formal way of defining the collaborative robot safety strategies was presented in Guiochet et al. (2008), where the authors used a graph representation of the safety mode concept, which associates a specific set of rules for each functional behavior of the monitored robot system.

To implement desired safety strategies, there are several approaches to control program architectures, which allow a robot and a human to

* Corresponding author.
E-mail address: aklimchik@lincoln.ac.uk (A. Klimchik).

https://doi.org/10.1016/j.engappai.2023.107141
Received 22 May 2023; Received in revised form 6 September 2023; Accepted 10 September 2023
Available online 20 September 2023

execute tasks in the shared workspace. Several research groups applied finite state machines for the interaction handling, where the states are the robot reactions. In particular, Parusel et al. (2011) proposed a practice-oriented approach, which operated with four main robot states: autonomous in the case of a human absence; high compliance mode in the case of a human presence; collaborative mode with a human in the loop; reflex reaction mode in the case of a fault. More advanced techniques were proposed in De Luca and Flacco (2012) and their following work (Magrini and De Luca, 2017), which additionally used external vision sensors for robot collision avoidance and safe human–robot coexistence. In both works, the collision handling strategies operated with the above-mentioned reactions such as switching to the emergency stop, reflex mode, kinematic redundancy mode, etc. Besides, the physical interaction parameters were also taken into account for choosing an appropriate robot reaction. Also, an external RGB-D sensor was used to estimate the interaction force application point. It is clear that the practical application of such sensors is limited because of possible occlusions and low frames-per-second rates.

In this work, the focus is on the development of adaptive robot control technique that in contrast to others ones takes into account both physical interaction parameters and also its type, which are identified and classified using data from internal robot sensors only (proprioceptive sensors). It is worth mentioning that here the *notion of adaptivity* is used for the high-level robot control while it is usually related to the low-level control of robot actuators in the conventional literature. To achieve the principal goal, the remainder of this paper is organized as follows. Section 2 gives a general overview of techniques used in human–robot interaction handling. Section 3 formulates the basic robot behavior modes. Section 4 presents the developed high-level controller based on the finite state machine for the selection of appropriate robot reactions. Section 5 describes the interaction types proposed in this work and shows how to distinguish them using AI-based techniques. Section 6 shows relevant experimental results that confirm the validity and utility of the proposed methods for safe and efficient human–robot physical interaction and collaboration. Finally, Section 7 summarizes the main contributions and gives directions for the future work.

2. Overview of interaction handling strategies for safe human–robot collaboration

Generally, the problem of human–robot physical interaction handling in collaborative robotics is treated in several consecutive steps, which include the interaction *detection*, *isolation*, *identification*, *classification*, and *reaction*. In addition, there are also some post and pre-collision steps that deal with collision avoidance and post-collision recovery. At the first of these steps, the presence of the interaction force acting on the robot is detected. At the second step the force application point is estimated; this can be done with different precision levels: from determining a colliding link number to estimation of the exact point location on the link surface. At the third step, the force amplitude and direction are computed, assuming that the force is applied to the isolated point found at the previous step. Further, at the fourth step, dealing with the interaction classification, some additional properties of the interaction are evaluated, such as accidental or intentional, permanent or repetitive, static or dynamic, etc. And finally, at the reaction step, an appropriate robot behavior is selected based on the interaction force properties in order to provide safe and efficient human–robot collaboration. More details concerning the above-listed stages can be found in Haddadin et al. (2017), De Santis et al. (2008), De Luca et al. (2006), Gribovskaya et al. (2011) and Bolotnikova et al. (2019). A typical example of a such human–robot interaction handling procedure is presented in Fig. 1.

It is clear that, before any physical interaction handling, the interaction should be *detected* by an appropriate algorithm. The principal performance of such algorithms are the response time (i.e. the interval from the time instant when the interaction event really started to the moment when the algorithm detects it), which should be as small as possible. Besides, the detection precision is also important that relate to the number of false-positive classifications. Since the detection is a binary problem, most of the relevant methods are based on the input signals comparison with some thresholds. In practice, the simplest way to detect interaction is monitoring changes in motor current or torque values (Harden et al., 2011; Takakura et al., 1989), which eliminates the need for a robot dynamic model. This simplicity proves advantageous, as slow motions can be treated as quasi-static, enabling the detection of collisions through sudden changes or disturbances in the observed motor signals. A more sophisticated approach uses the dynamic model of the robot and compares the actual joint torques with their estimated values. This technique is also known as the direct torque estimation or inverse dynamics approach (Haddadin et al., 2008). Recent work in this area (Heo et al., 2019) is based on a deep neural network, which demonstrated better response time and a lower false-positive rate compared to classical methods. However, this advantage was achieved only for cyclic operations, where the robot repeated the same trajectory tracking during its operation.

In relevant literature, it is typically assumed that the operator interacts with a general n-dof serial manipulator, which consists of a fixed base, an end-effector, and several links connected by n revolute joints. Besides, the interaction force could be applied to an arbitrary point on the manipulator link surface. In this work, the input data for the considered interaction handling algorithm are provided by one-axis torque sensors whose outputs usually include some additional components caused by robot dynamics. In particular, in case of physical interaction between the robot and external object, the robot dynamics equation is typically written as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}_m + \boldsymbol{\tau}_{ext} \tag{1}$$

where $\mathbf{q},\ \dot{\mathbf{q}},\ \ddot{\mathbf{q}} \in \mathbb{R}^n$ are the vectors of generalized coordinates, velocities, and acceleration, respectively; $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n\times n}$ is an inertia matrix; $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) \in \mathbb{R}^{n\times n}$ is a matrix of Coriolis and centrifugal forces; $\mathbf{F}(\dot{\mathbf{q}}) \in \mathbb{R}^n$ and $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ are the vectors of friction and gravitational torques; $\boldsymbol{\tau}_m$ is the vector of the total motor torques; $\boldsymbol{\tau}_{ext}$ is the vector of the torques at the joints generated by applied interaction force. In this work, it is assumed that the dynamic component is already excluded from the measurement data by means of a dedicated robot state observer (Haddadin et al., 2017; Mamedov and Mikhel, 2020), providing an estimation of the external torque. So further, only pure interaction components $\boldsymbol{\tau}_{ext}$ obtained from relevant state observer will be referred to as the joint torques $\boldsymbol{\tau}$. The use of only external torques values allows to generalize the developed techniques for the arbitrary trajectories and speeds of the manipulator.

The *isolation* step of interaction handling pipeline deals with estimating of the robot link index k where the interaction force is applied to. This index may be found by selecting of k significant components of the joint torque vector $\boldsymbol{\tau} = \left(\tau_1,\dots,\tau_n\right)^T$. In more formal way, the k-index defining the interacting link can be found using the following decision rule as was shown in (Haddadin et al., 2017)

$$\|\tau_k\| \geq \delta_\tau \quad \& \quad \|\tau_i\| < \delta_\tau, \quad \forall i > k \tag{2}$$

where δ_τ is some tolerance value allowing to distinguish significant and non-significant components in the measured torque vector $\boldsymbol{\tau}$.

The interaction *identification* step focuses on finding of the interaction parameters, such as the interaction force magnitude, its direction, and its application point. For the considered interaction, the basic relation (static equilibrium condition) between the force $\mathbf{f} = [f_x, f_y, f_z]^T$ and the joint torques vector $\boldsymbol{\tau} = \left(\tau_1,\dots,\tau_n\right)^T$ can be written using the manipulator Jacobian $\mathbf{J}(\mathbf{q},\mathbf{p})$, where $\mathbf{p}$ is the force application point and $\mathbf{q}$ is the joint coordinate vector. It is worth mentioning that here the interaction force may be applied to the any manipulator kth link, not to the end-effector only. So, the reduced Jacobian $\mathbf{J}_k(\mathbf{q},\mathbf{p})$ should be used, which is obtained from the conventional one $\mathbf{J}(\mathbf{q},\mathbf{p})$ by extraction of

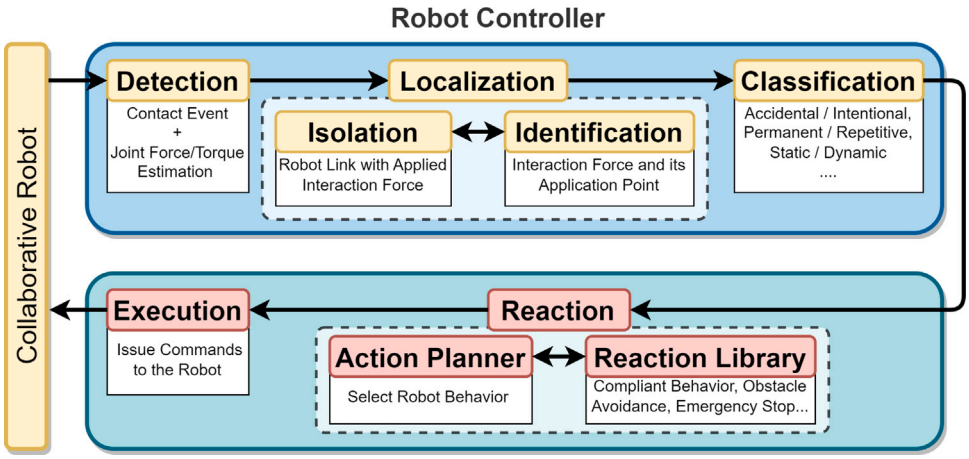


Fig. 1. Basic steps of the interaction handling in human–robot collaboration.

its first k columns. Using these notations, the desired static equilibrium equations can be written as

$$\mathbf{J}_k\left(\mathbf{q}^{(k)},\mathbf{p}\right)^T\cdot\begin{bmatrix}f_x\\f_y\\f_z\end{bmatrix}=\begin{bmatrix}\tau_1\\\dots\\\tau_k\end{bmatrix} \tag{3}$$

where $\mathbf{p}$ is the force application point, $\mathbf{q}^{(k)}=\left(q_1,\dots,q_k\right)^T$ is the reduced joint coordinate vector and $\boldsymbol{\tau}^{(k)}=\left(\tau_1,\dots,\tau_k\right)^T$ is the reduced joint torque vector. It should be noted that here the index k is assumed to be known and estimated at the previous step.

In the above equation, both the joint coordinate vector $\mathbf{q}^{(k)}$ and torque vector $\boldsymbol{\tau}^{(k)}$ are known from measurements, while the interaction force $\mathbf{f}$ and its application point $\mathbf{p}$ should be found. Related reduced Jacobian $\mathbf{J}_k(\mathbf{q},\mathbf{p})$ of size $3\times k$ can be obtained in conventional way, which yields the following expression

$$\mathbf{J}_k\left(\mathbf{q}^{(k)},\mathbf{p}\right)=\left[\mathbf{r}_1\times(\mathbf{p}-\mathbf{p}_1)\quad\dots\quad\mathbf{r}_k\times(\mathbf{p}-\mathbf{p}_k)\right] \tag{4}$$

where the vector $\mathbf{p}_i$ and the unit vector $\mathbf{r}_i$ define location and orientation of the ith joint axis.

For the considered problem, the above static equilibrium equation should be solved in conjunction with constraints describing the manipulator link surfaces. In practice, it is convenient to describe these surfaces using the conventional triangle meshes, which are widely used in 3D CAD modeling. This allows to present the basic geometric constraints as $\mathbf{p}\in\boldsymbol{\Omega}_k$, where the superscript k denotes the link number and the triangle mesh $\boldsymbol{\Omega}_k$ is defined by a set of its vertices $\mathbf{V}^k$, list of connections $\mathbf{E}^k$ forming 3D faces and corresponding face normal vectors $\mathbf{N}^k$, i.e.

$$\mathbf{p}\in\boldsymbol{\Omega}_k=mesh\left(\mathbf{V}^k,\ \mathbf{E}^k,\ \mathbf{N}^k\right). \tag{5}$$

Using this notation, each of the above 3D faces can be described by the equation

$$\begin{gathered}\boldsymbol{\Omega}_k^j=\alpha_1\cdot\mathbf{V}_1^{k,j}+\alpha_2\cdot\mathbf{V}_2^{k,j}+\alpha_3\cdot\mathbf{V}_3^{k,j}\\0\le\alpha_i\le1\quad\sum\alpha_i=1\end{gathered} \tag{6}$$

where k is index of the robot link, j-index denotes the number of triangular face $\mathbf{E}_j^k$ and α_i is some arbitrary variable, which defines the point coordinates at jth triangle of Ω_k with corresponding vertex points $\mathbf{V}_1^{k,j}$, $\mathbf{V}_2^{k,j}$, $\mathbf{V}_3^{k,j}$. It is worth mentioning that in the above equation, the index j is unknown and should be found taking into account the static equilibrium equation (3).

Another essential constraint that must be taken into account is related to the feasible directions of the interaction force $\mathbf{f}\in\boldsymbol{\mathcal{F}}_\mu$. As follows from the Coulomb friction law, the feasible directions can be presented in the form of a so-called 'friction cone', which is built around the link surface inward-pointing normal vector $\mathbf{n}_p$ located at the contact point $\mathbf{p}$. Assuming that the friction coefficient is equal to μ, this constraint can be presented in the following way

$$\left|\angle\left(\mathbf{n}_p,\ \mathbf{f}\right)\right|\le atan\left(\mu\right) \tag{7}$$

Although, that some of the human–robot interactions, like grasping, does not follow Coulomb friction law, it is still possible to use (7) with $\mu\to\infty$, which will correspond to the enlarged friction cone with angle up to 180 deg.

Thus, the desired solution $(\mathbf{f},\mathbf{p})$ computed for the identification step must satisfy the static equilibrium equations (3) as well as constraints on the force direction $\mathbf{f}\in\boldsymbol{\mathcal{F}}_\mu$ and the contact point location $\mathbf{p}\in\boldsymbol{\Omega}_k$. It is worth mentioning that depending on the number of equations k, the relevant linear system can be either over- or under-determined. Besides, in practice, the right hand side of these equations can be corrupted by the measurement noise that may cause the equations inconsistency in the strong sense. For this reason, it is meaningful to present the considered identification task as the following non-linear constrained optimization problem

$$\begin{gathered}\left\|\boldsymbol{\tau}^{(k)}-\mathbf{J}_k\left(\mathbf{q}^{(k)},\mathbf{p}\right)^T\mathbf{f}\right\|\to\min_{\mathbf{f},\mathbf{p}}\\\mathbf{p}\in\boldsymbol{\Omega}_k,\mathbf{f}\in\boldsymbol{\mathcal{F}}_\mu\end{gathered} \tag{8}$$

where the torque measurements $\boldsymbol{\tau}^{(k)}$ are assumed to be corrupted by the measurement noise and the index k is determined using expressions (2).

It should be noted that this paper mainly focuses on the human–robot interaction handling and relevant adaptation of robot behavior, while mathematical and algorithmic details related to the interaction parameters identification are carefully studied in our previous works (Popov et al., 2021a, 2022, 2021b). For this reason, here we concentrate on the *classification* and *reaction* steps, which together with the *identification* step are integrated into the proposed adaptive control strategy allowing to ensure safe robot collaboration with a human operator. As follows from the above overview, there are still a number of open questions in this area. In particular, it is required to improve existing interaction classification techniques and enhance interaction handling methods by means of adaptive control algorithms ensuring human safety during collaboration with a robot.

3. Robot behavior modes in human–robot interaction

In order to provide safe and efficient robot reactions in the human–robot collaboration, it is necessary to prioritize the desired characteristics of the entire workcell consisting of the robot, some auxiliary equipment, and the operator. For classical industrial robots, the main performance indicator is their manufacturing efficiency characterized by accuracy, speed, and load capacity. In contrast, collaborative robots share the working environment with humans, so safety is the most

important issue. Thus, for the collaborative robots, the prioritization of the desired performances should be the following: (i) human safety, (ii) robotic cell safety, (iii) efficient execution of the manufacturing task. It means that for the handling of each physical interaction, the collaborative robot should choose a proper reaction, which is the most riskless for the operator as well as prevent damaging the robot itself while executing the desired task in a more accurate and fast way. In practice, these can be achieved by applying advanced joint-level control, which ensures the required joint angles, velocities and, in addition, desired joint compliance. Relevant low-level controller settings are generated by the high-level controller that ensures the safe interaction between the robot and the operator.

The simplest way of interaction handling in collaborative robotics is a straightforward stop reaction, where the robot suspends all motions if the interaction is detected. Such a reaction is obviously safe but can be hardly efficient in manufacturing workcells. In particular, there are a number of so-called "contact operations", which involve direct physical interaction between a robotic tool and a workpiece. Hence in practice, it is required to employ more complex robot reactions depending on the interaction type and parameters. Besides, it is necessary to take into account that for the contact operations the interaction force can be caused by different sources, such as a technological process, an obstacle, or a human. It is clear that the robot reactions for each case should be different and usually pursue distinct goals.

To define the set of possible robot behavior modes, the existence of physical interaction and its parameters should be taken into account. If the robot does not interact with the operator or any object in its environment, the robot behavior mode will be further referred to as the *autonomous task execution mode*. Here, the robot simply follows the desired reference trajectory, without controlling the joint compliance. In contrast, if the human–robot or environment-robot interaction is detected, then the robot is switched to the so-called *interaction handling mode*. This mode is activated by interaction detection and remains active until it disappears. It should be noted that there is also a special case of the *autonomous task execution mode*, which corresponds to the contact operations and assumes the physical interaction between the robot and a workpiece. In addition, for safe operation, the *emergency fault mode* should be provided that is enabled when some modules of the system are not working as expected. It is clear that generally, the robot's reaction and its working mode depend on the interaction type and parameters.

Let us consider first the *autonomous task execution mode*, which is characterized by the absence of any robot reactions since there is no unexpected physical interaction. As it has been mentioned above, here two sub-cases that are possible:

Non-contact mode. It corresponds to the robot following the desired trajectory without controlling the joint compliance. In this case, no obstacles or a human in the robot environment were detected. The principal goal here is the efficient execution of the task while ignoring the safety issues both for the robot and the operator.

Contact mode. It corresponds to the robot following the desired trajectory while maintaining the desired interaction force amplitude and direction. Typically such interaction force is applied to the workpiece by the robot tool at the desired time. However, any additional robot interaction with a human or environment is not expected, so the safety issues both for the robot and the operator are ignored.

Further, in the *interaction handling mode* four sub-cases are possible that differ in robot reaction to the detected interaction:

Pause mode. In such mode the robot stops its motions along the desired trajectory for some time. The motion recovery process can be initialized either manually, initialized by the operator, or automatically after a period of time. The first option is convenient for the operator and can be implemented as a reaction to the second touch: the first touch stops the robot, and the second one resumes its motion. The automatic option is useful in case of brief interactions, after which the robot can continue its motion if the interaction force is no longer applied.

Compliant mode. In this mode the robot motions are determined by the joint compliance controller, which ignores the desired trajectory. In the literature, this mode can be also referred to as the reflex mode. Here, the robot position and configuration can be changed manually, by the operator who is applying proper force to the end-effector. Such a technique can be used if the robot cannot avoid an obstacle or when it is reaching the joint angle limits.

Redundant mode. In this mode the robot motions are handled by the hybrid position-compliance controllers for all joints, which try to follow the desired trajectory while ensuring certain manipulator stiffness. Here, the robot kinematic redundancy is usually used to simultaneously move the end-effector along the desired path and maintain the interaction force at a safe level. In contrast to the two previous modes, in this case, the task execution is not interrupted. This mode is available for robots with more than six degree-of-freedoms only.

Obstacle avoidance mode. In this mode the robot tries to avoid an obstacle on its way. Since it is assumed that only the internal sensors of the robot are available, the obstacle is detected after interacting with it. Further, the identified interaction parameters are used to compute the obstacle position. Depending on the robot state and the interaction type, the dedicated obstacle avoidance algorithm generates a new collision-free path in the neighborhood of the detected obstacle.

Finally, if some of the robot safety features are violated or the robot encountered some hard-to-solve problems requiring the operator intervention, the *emergency fault mode* is activated:

Emergency stop mode. In such mode the robot stops its motions along the desired trajectory and waits. Activating this mode is similar to pushing the "red button" on the robot controller. Such a mode is enabled if the identified interaction force exceeds some safe limits. In contrast to the pause mode described above, this mode necessarily requires the operator's intervention to proceed.

Hence, to ensure safe human–robot collaboration, three basic behavior modes with seven sub-cases describing the robot reactions should be considered. It is clear that these behavior modes are very basic and may be modified and adapted for any particular practical application. To ensure switching between these modes in real-time, a dedicated high-level controller was developed, which is the focus of the following section.

4. Robot reaction control based on interaction parameters

In order to achieve the desired safe human–robot collaboration, it is necessary to ensure proper switchings between the above-described robot modes. The latter is implemented by a high-level controller that is usually used for trajectory planning and auxiliary equipment control. This additional feature of the human–robot interaction handling may be realized by using a finite state machine, which is widely used in computer science. For the considered application, the machine states are the robot behavior modes while the transitions are the switchings between these modes. In this work, the transitions are executed using some specific conditions, which depend on the identified interaction parameters. In more detail, such finite state machine and relevant transition conditions are presented below.

The proposed *finite state machine* for the human–robot interaction handling is represented in Fig. 2. This machine includes seven states corresponding to different robot behavior modes described in the previous section. The transition between these states is executed when the interaction is detected and highly depends on the identified interaction parameters. In particular, some transitions are invoked if the interaction appears while other ones are activated when the interaction disappears. In more detail, the functioning of this finite state machine is described below.

First, let us consider the robot behavior in the case of the **non-contact operation mode**, which is typically used for robot tool positioning. The simplest and safest reaction to any accidental interaction (**Accd**) is to stop the robot. The latter corresponds to the transition

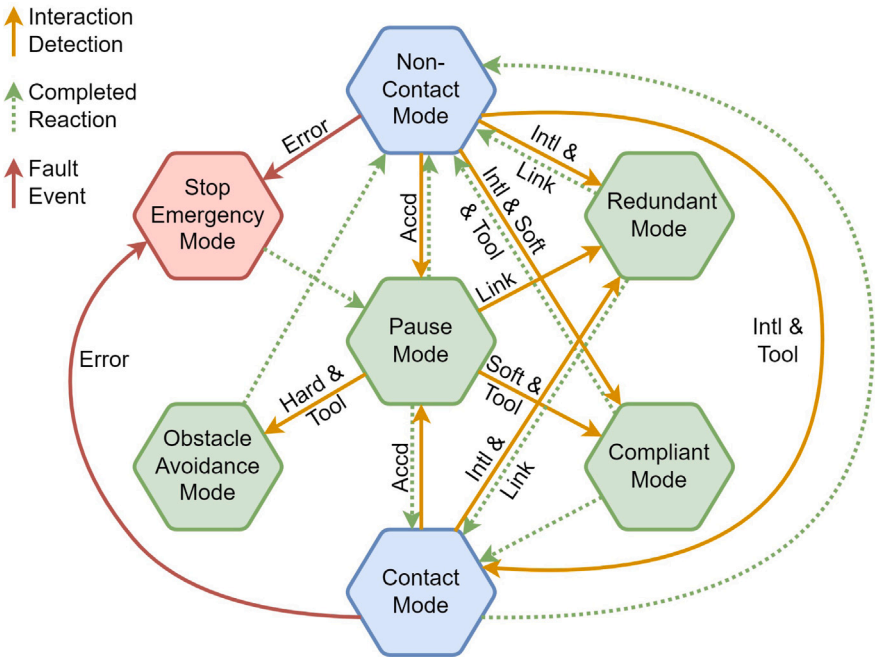


Fig. 2. Proposed finite state machine with its states and transitions. The *states* are the robot behavior modes and *transitions* are switchings between the states depending on the interaction parameters and characteristics: applied at the tool or intermediate link (Tool/Link); having soft or hard nature (Soft/Hard); interaction intention (Intl/Accd). Only the most important states and transitions are shown while duration-based transitions (Short/Long) are hidden.

to the **pause mode**. Here, two options of this mode are used, which depend on the interaction duration. In the case of the brief interaction with the operator (**Soft**), the pause mode is activated and deactivated by touch: the first touch stops the robot motions and the second one resumes the execution of the non-contact mode. But for the long interaction, for instance, if the interaction force is caused by the collision with the robot environment (**Hard**), the pause mode will remain enabled until the force disappears. In the case of intentional interaction (**Intl**), the robot's behavior depends on the interaction parameters. If the interaction force is applied at the robot intermediate link (**Link**), the **redundant mode** is activated that does not stop the robot motion, continues executing the desired end-effector trajectory, but does not allow increasing of interaction force amplitude. In contrast, if the operator interaction with the robot end-effector was detected (**Tool & Soft**), the robot is switched to the **compliant mode**, where it stops the motions and increases the compliance in order to avoid the operator injury. Otherwise, if such interaction force is caused by the robot environment (**Tool & Hard**), then the robot is switched to the **obstacle avoidance mode**. It is worth mentioning that the considered finite state machine also allows to switch the robot from the non-contact mode to the contact mode, which corresponds to the intentional interaction between the robot tool and the workpiece (**Intl & Tool**).

Further, consider the **contact operation mode**, where the robot interacts with a workpiece by means of its tool. In this case, the number of potential reactions is limited since the robot should maintain the desired interaction force depending on the technological process. Here, there are two options for the robot reactions, switching either to the **redundant mode** or **pause mode**. The first one is used when the force is applied to the intermediate robot link (**Intl & Link**) and it is similar to the non-contact case described above. The second reaction is applied in the case of any accidental interaction (**Accd**), it stops all robot motions since it is the safest robot behavior here. Any other reactions are not provided for the contact mode, because of the requirements issued from the technological process specifications e.g. the desired interaction force amplitude.

It should be noted that for the considered finite state machine, some transition conditions require additional characteristics of the interaction, in addition to the previously described force amplitude, direction, and its application point. In particular, it is necessary to distinguish the robot-operator and robot-environment interactions (soft or hard) in order to choose a proper reaction. Besides, the interaction intention (accidental or intentional) is also important since it must influence the robot's behavior. Relevant interaction classification techniques are presented in the following section.

5. Classification of human–robot interactions

To ensure safe human–robot collaboration, the robot controller must take into account not only the interaction parameters **f**, **p** but also distinguish some interaction properties, which may have different nature (Cioffi et al., 2020; Kouris et al., 2018; Briquet-Kerestedjian et al., 2019; Cho et al., 2012; Gamboa-Montero et al., 2020). In particular, the interaction may be caused by accidental contact with the operator or by collision with the auxiliary equipment, as well as by intentional force application to the workpiece. Let us consider this issue in detail, starting from the definition of the possible interaction classes and then proposing techniques allowing to attribute these classes to the interactions.

Interaction classes. The proposed set of the interaction classes is shown in Fig. 3. First of all, it is necessary to distinguish the interaction duration, which is described here by **Short** and **Long** classes. Relevant classification can be done by a simple comparison of the interaction time with a predefined threshold value, which can be found empirically. In our study we used a threshold of about 1 s. The latter allows to separate easily a short accidental operator touch and dangerous collision with an object in the robot environment. Further, the interactions differ in their nature, assumed to be either **Soft** or **Hard**. In practice, the soft interactions are usually associated with a human, while the hard ones normally occur when the robot collides with a workpiece, walls, etc. However, by using the robot's internal sensors only, it is rather hard to separate the human–robot interaction and the robot interaction with some soft objects. Thus, it is explicitly assumed here that any soft interaction is related to the operator. Relevant classification can be done by analyzing the time series describing the robot joint torques evolution over time. Also, the interaction may be applied to either the

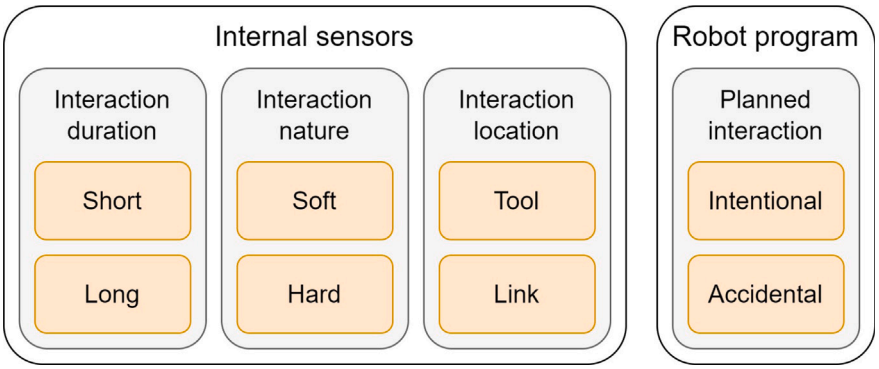


Fig. 3. Diagram of the interaction classes. The interaction duration, nature, and its location are determined by using the robot's internal sensors. The intentional and accidental interactions are predefined in the robot program.

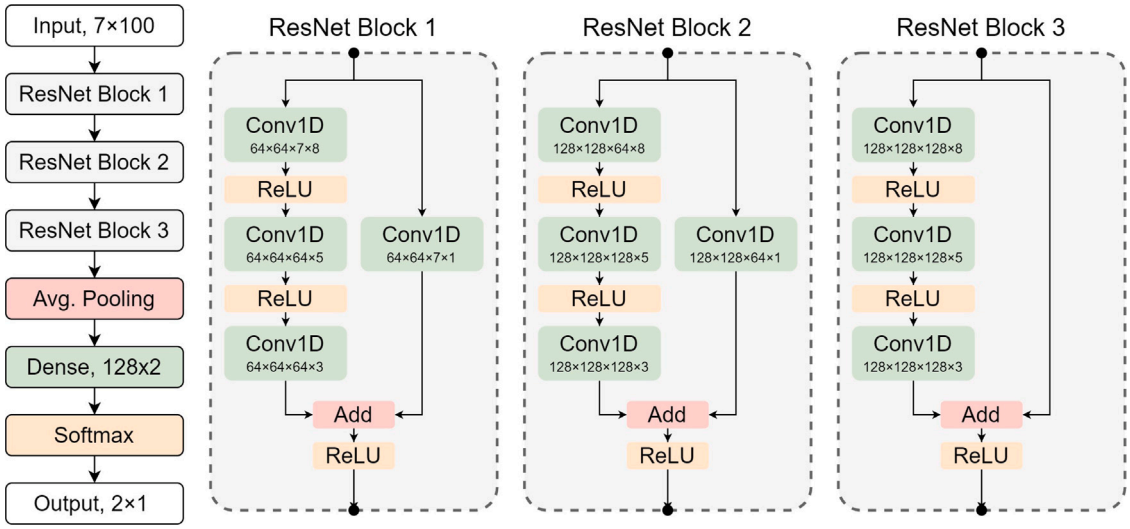


Fig. 4. Structure of the deep neural network for Soft/Hard interaction classification. As input, a fixed time window of the robot joint torque measurements is used, and the output is the interaction class.

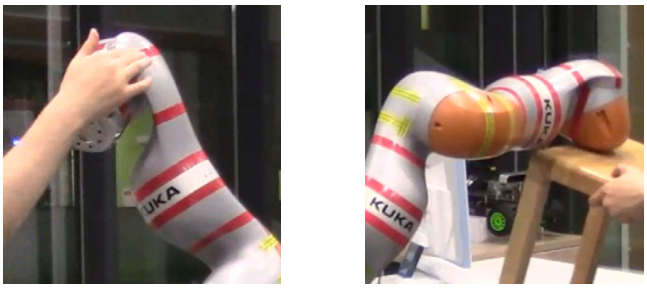
robot end-effector (**Tool**) or its indeterminate **Link**. Relevant classification can be easily done by using the interaction parameter **p** provided by the identification algorithm. Finally, any interaction can be either intentional (**Intl**) or accidental (**Accd**). For non-contact operations, in most cases the interaction is accidental but for the human–robot collaborative work, the interaction is usually intentional. In practice, a relevant decision is made at the robot programming stage and does not depend on robot sensor readings.

Classification techniques. To specify the interaction class, different techniques can be applied. Let us describe those of them that were used for our experimental study. For the Short/Long, Tool/Link, and Intentional/Accidental classes, the classification techniques are rather straightforward in implementation, as was described above. In contrast, distinguishing between Soft and Hard interactions is more complicated and requires some advanced algorithms to analyze the robot joint torques evolution over time. In this study, neural network techniques were used to achieve this classification.

To classify time-series data obtained from the robot joint torque sensors, a deep neural network was used since this technique proved to be very efficient for similar problems (Tchatchoua et al., 2022). In this network, the fixed time window of the torque measurements provided the input and the desired interaction class (Soft/Hard) was obtained at the output. The generalized architecture of such neural network is presented in Fig. 4, which implements the residual-based technique also known as ResNet and widely used in image analysis (He et al., 2016; Cheng et al., 2023; Javidi and Jampour, 2020), robotics applications (Kumra and Kanan, 2017), fault diagnosis (Liang et al., 2022) and speech recognition (Hershey et al., 2017).

Network structure. The input dimensions of the developed ResNet consist of 7 × 100, where 7 corresponds to the number of joints and 100 represents the consecutive external torque measurements. The update frequency of the torque values is 100 Hz, resulting in a fixed time window of exactly 1 s. The developed network consists of three sequential Residual blocks. Similar to the original ResNet, each of these blocks has two paths for the input data, where the first path includes three subsequent convolutions with ReLU activation functions. These convolution layers have a kernel size of 8 × 8, 5 × 5, and 3 × 3 correspondingly. The second path has only one convolution layer, except for the last Residual block, which has a direct connection to the end of the first path. In each of the Residual blocks, the outputs of the first and second paths are combined by the addition operation and following ReLU activation. It should be noted that the size of tensors is 64, 128, and 128 for the first, second, and third Residual blocks correspondingly. The output of the last Residual block is followed by the global average pooling and the dense layer with a softmax activation function. So, after the proposed modifications of the classical ResNet, the developed network is able to attribute the physical interaction to one of the predefined classes, Soft or Hard.

Network training. To train the developed network for the interaction type classification, the proper dataset was captured from the KUKA iiwa robot used in our experiments. To obtain the desired data, the robot followed some random trajectories and repeatedly collided with



(a) Soft interaction (b) Hard interaction

Fig. 5. Dataset acquisition examples for the interaction classification. The robot physically interacts either with a human or some rigid objects from its environment.

the operator and environment. In total, about 430,000 data frames with 100 Hz rate and 500 collision cases were recorded, which corresponded to both hard and soft interactions. The latter of them occurred not only in a collaborative way, where the operator guided the robot, but also in an accidental way, where the robot unintentionally collided with the operator's arm, back, or body. An example of data acquisition for soft or hard interaction cases is presented in Fig. 5. In this study, the captured dataset was partitioned into training, test, and validation sets using a 70-15%-15% proportion. To address class imbalance, the random undersampling technique was employed, which balanced the number of examples in each class. The developed neural network was trained using the backpropagation algorithm with the Adam optimizer, initialized with the following parameters: $Lr = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$. To further improve the optimization process, a learning rate schedule was implemented. If the loss metric showed no improvement for over 10 epochs, the learning rate was reduced by multiplying it with a factor of 0.5, ensuring its minimum value of $1 \cdot 10^{-5}$. As a result, the final learning rate settled at approximately $1 \cdot 10^{-4}$. Since there are only two classes, Soft and Hard, the binary cross-entropy was used as the loss function.

Network evaluation. As follows from a relevant experimental study, the developed neural network provided an accuracy of 98% with score of 0.9788 on test, 0.9697 on validation and 0.9847 on training data. Besides, additional experiments proved the network's ability to separate reliably the interactions with the operator's hand, or body from the interactions with hard objects, such as metal workpieces and walls. It should be also mentioned that the simplest one-layer feedforward neural network was also evaluated for its ability to solve the same task. Here, the Fourier transform coefficients were used as the input and the network was composed of 100 neurons. However, this simplified approach provided essentially lower accuracy of 89%, which is not enough for reliable interaction classification. More details regarding this network and its performance can be found in one of our previous works (Popov et al., 2017).

Thus, to provide safe robot interaction with the operator and the environment, the relevant controller was developed, based on the finite state machine technique. This controller allows to change the robot's behavior depending on the detected interaction type and parameters. In particular, the controller distinguishes the Short and Long, Soft and Hard, Tool and Link, Intentional and Accidental interactions, which cause switching between several predefined robot behavior modes. The efficiency of this controller is confirmed by the experimental study presented in the following section.

6. Implementation of the developed adaptive interaction handling controller

The developed high-level robot controller ensuring safe human–robot collaboration is based on the modular structure. It consists of two independent parts, where the first one was implemented inside of the original robot controller of KUKA iiwa (internal part) and the second one was realized on the external computer. The communication between the robot controller and external computer was executed by means of ROS (Robot Operating System), which allows to receive the sensor measurements from the robot and sends it the motion commands. Here, we used the advantages of KUKA iiwa controller that supports high-level Java programming language, in contrast to the conventional robotic controllers for which highly-specialized languages (KRL, Karel, etc.) are available only. This allowed to connect the robot controller Java Core with the external ROS core by means of the open source ROS package *iiwa_stack* (Hennersperger et al., 2017). It should be also mentioned that some modifications were done to the original *iiwa_stack* in order to obtain required data from the sensors or the state observers as well as provide online adjustment of the joint compliances.

The general structure of the controller is presented in Fig. 6. Its external part consists of five main modules: Communication, Identification, Classification, Finite State Machine and Reaction Library. The *Communication* module creates the ROS core and transforms the robot controller data into a more convenient form. In addition, this module sends the commands to the robot, which define the desired robot joint angles and compliances. The *Identification* module uses the robot joint torques and angles measurements to detect the physical interaction and to identify its parameters ($\mathbf{f}$, $\mathbf{p}$) using the algorithms presented in Popov et al. (2022, 2021a). The *Classification* module attributes the detected interaction to one of the predefined classes listed in Fig. 3 and described in detail above; in case of Soft/Hard classification, the deep neural network is used. Further, the interaction parameters and characteristics are transmitted to the *Finite State Machine* presented in Fig. 2. This machine defines the safe robot behaviors modes, using its states and transition between them as functions of the interaction parameters and properties obtained from the Identification and Classification modules. The set of all possible robot reactions is stored in the *Reaction Library* module, which predefines the safe robot behaviors. It should be mentioned that each part of the high-level controller is implemented as a separate module, so it is quite flexible and can be easily used to enlarge a variety of desired robot behaviors. In particular, by redefining the set of robot behavior modes/reactions and their transitions, the operator can adapt the controller to any particular industrial process that assumes intensive human–robot collaboration.

The external part of the above controller was implemented mainly in *C++* language, while the identification module was compiled as a library from the *MATLAB* functions using MATLAB Coder, and the Soft/Hard classification module was implemented in *Python* language using Keras/TensorFlow libraries. The internal part of the controller was written in *Java* language and included a number of modules from the open source software *iiwa_stack* with relevant modifications.

The developed interaction handling controller proved to be reliable, ensured safe human–robot collaboration in our experimental study and was successfully used in the laboratory experiments validating the proposed techniques.

7. Experimental validation of proposed interaction handling technique

In our experimental study, the robot was programmed to follow the desired hexagon trajectory as shown in Fig. 7. While programming the robot for this task, it was assumed that normally any physical interactions may not occur, which corresponds to the Non-Contact Operation Mode of the developed Finite State Machine. Nevertheless, to ensure safe human–robot collaboration, some additional properties were assigned for each trajectory segment, which admits some unexpected interaction with the operator, either Accidental (Accd) or Intentional (Intl) ones. In particular, the complete trajectory was composed of four closed loops $\mathbf{p}_1 \rightarrow \mathbf{p}_2 \rightarrow \cdots \rightarrow \mathbf{p}_6 \rightarrow \mathbf{p}_1$, where the first two loops treated all detected interactions as Accidental (Accd) ones and the remaining loops treated them as Intentional (Intl) interactions. It should

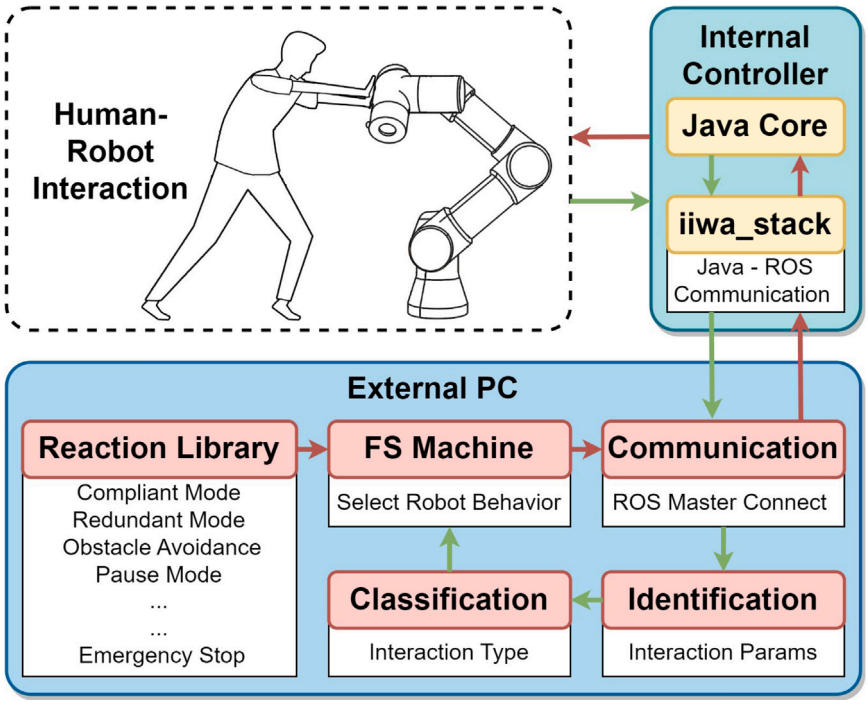


Fig. 6. General structure of developed adaptive interaction handling controller. It consists of two independent parts implemented using the original robot controller and the external computer respectively.

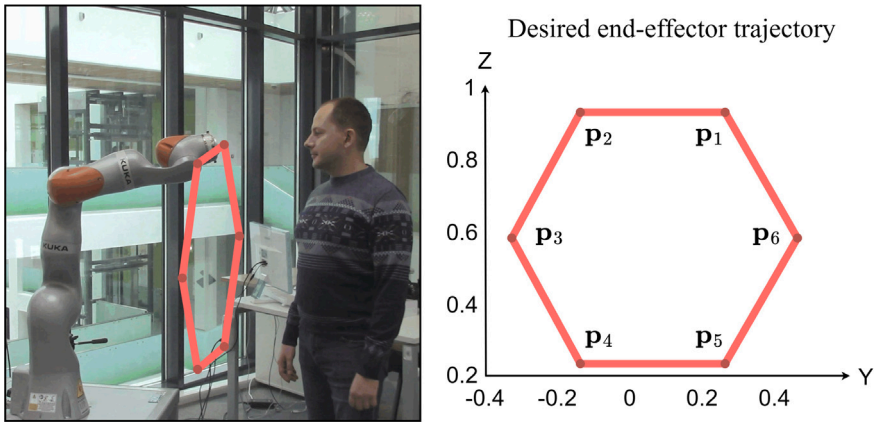


Fig. 7. Experiment setup and benchmark trajectory for validation of proposed interaction handling technique.

be noted that in this experimental study, the general robot behavior was controlled by the Finite State Machine described above.

During the experiment, while the robot followed its desired trajectory, some interactions were created with their force applied either at the robot end-effector (Tool) or its intermediate links (Link). In addition, to verify the developed controller ability to distinguish Soft and Hard interactions, some forces were applied by the human operator (Soft) while other ones were caused by the collision with a rigid object (Hard). Besides, the interaction duration was also different (Short/Long), in order to evaluate the reasonableness of the controller reactions.

The experimental setup included the collaborative robot KUKA LBR iiwa 14, which was connected to the external PC with Intel Core i5 3 GHz CPU, 8 Gb RAM and RTX2080TI. The high-level interaction handling controller was composed of two parts. The external part was based on the PC with a Linux operation system, ensuring fast communication with the internal part. In this controller, the Identification and Classification module implements the main theoretical contributions of this thesis. In general, the developed control system is capable of real-time execution of the desired robot motion program, while ensuring the maximum exchange rate between the internal and external parts of 800 times per second. However, in this application example, the exchange rate was reduced down to 100 times per second. The latter was caused by the limitations of the communication bandwidth and the necessity to obtain some additional service information from the robot. In cases where there is no physical interaction with the robot, the detection module evaluates the input robot torques during every 10 ms loop. However, when the interaction is applied to the robot, it is not immediately detected due to transient processes in the joint torques and observer dynamics. In practice, it takes an average of 28 ms for the detection module to register the interaction. Once the interaction is detected by the detection module, it simultaneously initiates the interaction identification and classification processes. Their runtimes are approximately 5 ms and 9 ms, respectively. Here, the input data for the classification neural network consists of the 100 previous measurements, which corresponds to approximately 1 s of data. This duration includes a few last measurements corresponding to the interaction

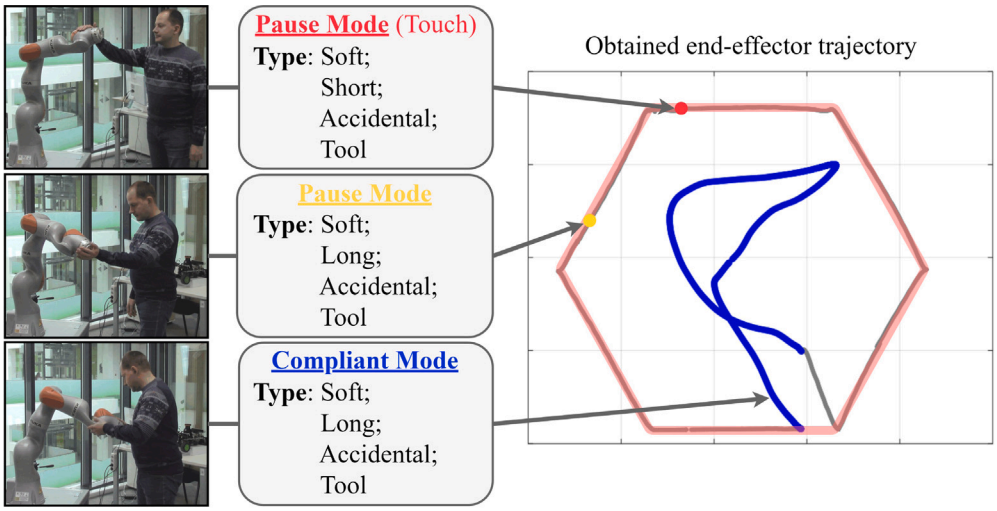


Fig. 8. Stage #1 of the experimental study corresponding to the first loop of the robot end-effector trajectory. Here, all interactions were considered as accidental and the robot interacted with the human operator.

and transient process in joint torques, providing reliable interaction classification. The total response time between applying the interaction force and sending a new control command to the robot controller is about 50 ms. It is also worth mentioning that this experimental study involved the direct physical interaction between the human operator and the robot, which must be definitely safe even in the case of unexpected robot behavior during experimentation. So, to eliminate the potential harm, the maximum stiffness of the robot joints was deliberately set to some low values that caused the increased trajectory tracking errors. It is clear that for real-life industrial applications such stiffness modification is not required and the tracking errors will be essentially low.

Another important issue related to this experimental study is the natural operator behavior, who was allowed to apply the interaction force to any robot link or even to interact with the robot using both his hands. The latter is not in good agreement with the principal assumptions implemented in the Identification and Classification modules, where a single interaction with the links #3...7 was admitted. Nevertheless, the robot behavior observed in the experiments was safe and the interactions were handled in the expected way.

The experimental study was composed of four stages corresponding to four closed loops $\mathbf{p}_1 \rightarrow \mathbf{p}_2 \rightarrow \cdots \rightarrow \mathbf{p}_1$ of the robot end-effector trajectory presented in Fig. 7. At each stage, assuming that Non-Contact Mode was active, the collaborative robot was subject to physical interactions of different types (Soft/Hard, Short/Long, Accd/Intl, Tool/Link). In more detail, these stages covering all possible combinations of the interaction types are described below.

The *Stage #1* is presented in Fig. 8, it corresponds to the first loop of the desired robot end-effector trajectory. Here, three interactions were detected, all of them were considered as accidental (Accd). The first one was generated by the operator who briefly applied the force at the robot end-effector. It was recognized as Soft/Short/Accd/Tool, so the controller reaction was to stop the robot (Pause Mode). The related end-effector position is shown in the figure with a red dot. The subsequent brief human touch on the end-effector resumed the robot's motion. The second interaction was also generated by the operator who grasped the robot end-effector for a certain time. It was recognized as Soft/Long/Accd/Tool, so the controller reaction was to stop the robot as well (Pause Mode). The related end-effector position is shown in the figure with a yellow dot. In this case, the robot motions were suspended until this interaction disappeared. The third interaction was generated by the operator, who did not simply grasp the robot end-effector but tried to move it away. It was recognized as Soft/Long/Accd/Tool, so the controller reaction was to stop the trajectory tracking and essentially increase the joint compliances (Compliant Mode). Related end-effector trajectory is shown in the figure with blue color and was clearly determined by the human operator. In contrast to the previous interaction, here the force was applied for a certain time, its amplitude continuously increased and the robot was switched to the compliant mode when the force exceeded some predefined threshold.

The *Stage #2* is presented in Fig. 9, it corresponds to the second loop of the desired trajectory. Here, three interactions were detected, all of them were considered as accidental (Accd) but in contrast to the previous stage, some of them were caused by a collision with a rigid object. The first interaction was generated by the operator, who grasped the robot's lower link and tried to move it away. It was recognized as Soft/Long/Accd/Link, so the controller reaction was to use the robot kinematic redundancy to continue the desired trajectory (Redundant Mode). Related end-effector trajectory is shown in the figure with a blue line, which is close to the desired path. Here, the obtained trajectory slightly differed from the desired one since the joint compliances were set to rather high values. The second interaction was caused by the collision between the robot end-effector and a rigid obstacle on its path. It was recognized as Hard/Long/Accd/Tool, so the controller reaction was to stop the motions (Pause Mode). Related the end-effector position is shown in the figure with a red dot. In this case, the robot motions were suspended until the interaction force disappeared. The third interaction was also caused by the collision between the robot's $3rd$ link and the rigid obstacle on its path. It was recognized as Hard/Long/Accd/Link, so the controller reaction was to stop the robot (Pause Mode) until the force disappeared.

The *Stage #3* is presented in Fig. 10 and corresponds to the third loop of the desired trajectory. Here, two interactions were detected only, all of them were considered as intentional (Intl). The first interaction was generated by the operator who grasped the robot end-effector and tried to move it away. It was recognized as Soft/Long/Intl/Tool, so the controller reaction was to stop the trajectory tracking and essentially increase the joint compliances (Compliant Mode) in order to follow the human operator hand guiding. Related end-effector trajectory is shown in the figure with a blue color. Here, the force was applied for a certain time, its amplitude continuously increased and the robot was switched to the compliant mode when the force exceeded some predefined threshold. The second interaction was also generated by the operator, who implemented a similar scenario but the interaction force was applied to the robot $4th$ link. It was recognized as Soft/Long/Intl/Link, so the controller reaction was to use the robot kinematic redundancy to continue the desired trajectory (Redundant

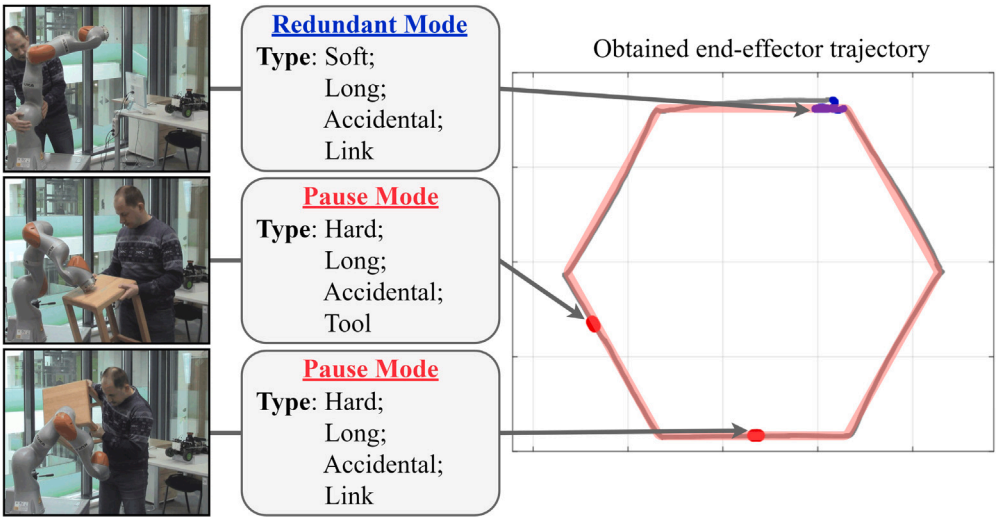


Fig. 9. Stage #2 of the experimental study corresponding to the second loop of the robot end-effector trajectory. Here, all interactions were considered as accidental and the robot interacted with the human or environment.

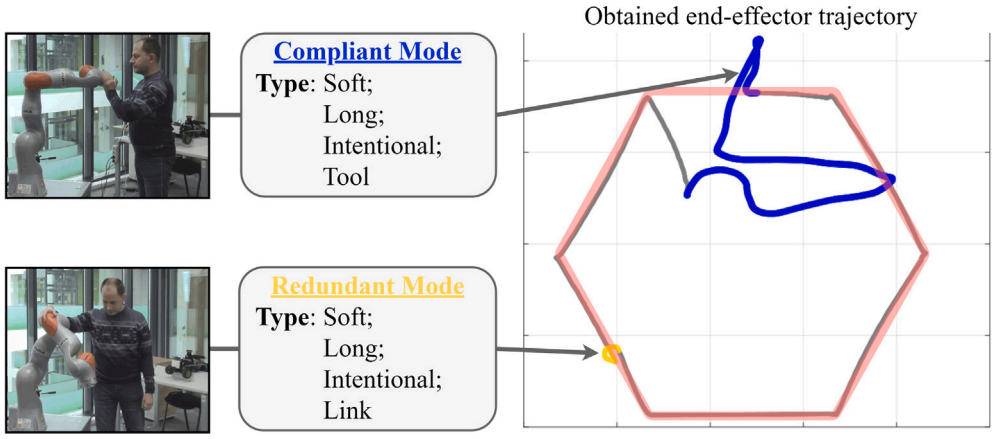


Fig. 10. Stage #3 of the experimental study corresponding to the second loop of the robot end-effector trajectory. Here, all interactions were considered as intentional and the robot interacted with the human operator.

Mode). The related end-effector trajectory is shown in the figure with a yellow line.

The *Stage #4* is presented in Fig. 11 and corresponds to the fourth loop of the desired trajectory. Here, three interactions with the rigid objects were detected, all of them were considered as intentional (Intl). The first and second interactions were caused by the collision between the robot end-effector and rigid obstacles on its path. They both were recognized as Hard/Short/Intl/Tool, so the controller reaction was to avoid the obstacle by changing the robot's path (Obstacle Avoidance Mode). Relevant end-effector trajectories are presented in the figure with red lines, showing how the robot avoids the obstacle. The third interaction was caused by the collision between the robot's 3*rd* link and a rigid obstacle on its path. It was recognized as Hard/Long/Intl/Link, so the controller reaction was to use the robot kinematic redundancy to continue the desired trajectory (Redundant Mode). The relevant trajectory is shown in the figure with a blue line, which is close to the desired path.

It should be noted that the above-described experiments cover all essential combinations of the interaction types, which may be generated from possible options Soft/Hard, Short/Long, Accd/Intl, Tool/Link. In fact, in spite of that, the number of all possible combinations is $2^4 = 16$, only 8 of them are different from the practical point of view. For this reason, the experimental study considered a lower number of sub-cases related to the interaction types.

Thus, the presented experimental study confirms *validity* of the developed interaction handling techniques. The relevant high-level controller implements the proposed identification algorithms as well as the classification methods, it allows adapting the robot behavior to the detected interaction. In particular, depending on the interaction parameters and type, the robot behavior may be switched to one of the predetermined modes ensuring safe human–robot collaboration while executing the manufacturing task.

8. Conclusion

This paper presents a novel adaptive technique for handling physical interactions between humans and robots, which allows to improve human safety and to enhance the performance of human–robot collaboration by means of artificial intelligence methods. The proposed high-level controller integrates the interaction identification algorithms and classification methods, which use data from the proprioceptive sensors only. The developed controller is based on the finite state machine and the deep residual neural network. It implements several robot behavior modes that are predefined and executed depending on the parameters and characteristics of the detected interaction, ensuring the safety of the human operator during the collaboration process while executing complex manufacturing tasks.

In contrast to other works, the extended set of interaction classes is used, which includes interactions applied to the robot end-effector or

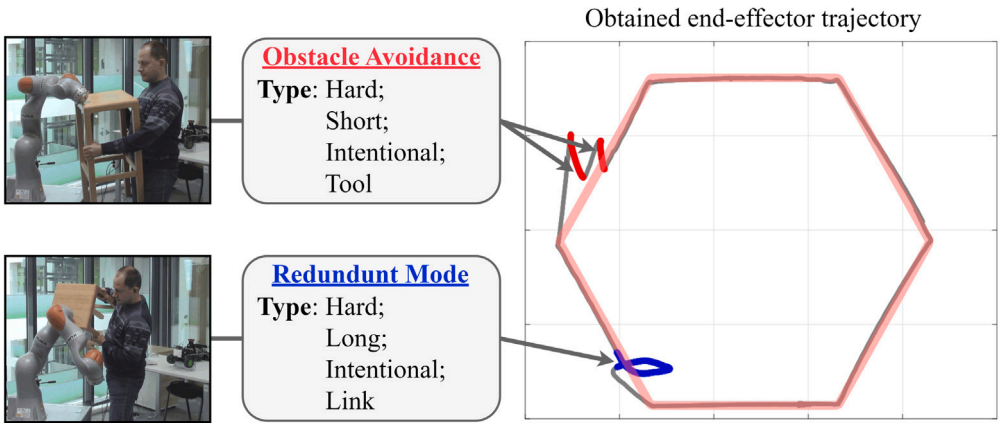


Fig. 11. Stage #4 of the experimental study corresponding to the second loop of the robot end-effector trajectory. Here, all interactions were considered as intentional and the robot interacted with its environment.

intermediate link (Tool/Link), having soft or hard nature (Soft/Hard), as well as having different intention (Intl/Accd) or duration (Short/Long). The classification of the interaction nature was implemented using a ResNet-based deep neural network, achieving an accuracy of 98% in the typical case study.

The validity of the developed technique is confirmed by the experimental results obtained for a collaborative task executed by the industrial robot KUKA LBR iiwa in a shared environment with a human operator. In future, it is planned to adapt the developed technique to a collaborative assembly process. The latter could be done by redefining the set of robot behavior modes and their transitions in the relevant controller, which enables safe and efficient physical human–robot interaction.

Future work may involve the exploration of the neural network structure, focusing on its optimization, and discussing dataset creation and training in more detail. It may include experimenting with different types of neural network architectures for time series analyses and relevant Ablation study to identify the most suitable network configuration for the interaction classification task. Also, it may be interesting to implement a unified neural network that expands its capabilities beyond interaction classification to include interaction detection and identification. Additionally, the applicability and adaptability of the proposed methods can be addressed across diverse robotic systems, varying in the number of degrees of freedom. This evaluation will also consider the problems of transfer learning in the context of different robot types.

CRediT authorship contribution statement

Dmitry Popov: Writing – original draft, Software, Investigation, Methodology. **Anatol Pashkevich:** Conceptualization, Writing – review & editing, Methodology, Formal analysis. **Alexandr Klimchik:** Data curation,Writing – review & editing, Supervision, Validation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Alexandr Klimchik and Anatol Pashkevich are guest editors for the special Issue AI for Industry 4.0

Data availability

The authors do not have permission to share data.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.engappai.2023.107141.

References

Bolotnikova, A., Courtois, S., Kheddar, A., 2019. Multi-contact planning on humans for physical assistance by humanoid. IEEE Robot. Autom. Lett. 5 (1), 135–142.

Briquet-Kerestedjian, N., Wahrburg, A., Grossard, M., Makarov, M., Rodriguez-Ayerbe, P., 2019. Using neural networks for classifying human-robot contact situations. In: IEE European Control Conference (ECC). pp. 3279–3285.

Cheng, Y., Zhao, Z., Wang, Z., Duan, H., 2023. Rethinking vision transformer through human–object interaction detection. Eng. Appl. Artif. Intell. 122, 106123.

Cho, C.-N., Kim, J.-H., Kim, Y.-L., Song, J.-B., Kyung, J.-H., 2012. Collision detection algorithm to distinguish between intended contact and unexpected collision. Adv. Robot. 26 (16), 1825–1840.

Cioffi, G., Klose, S., Wahrburg, A., 2020. Data-efficient online classification of human-robot contact situations. In: IEEE European Control Conference (ECC). pp. 608–614.

De Luca, A., Albu-Schaffer, A., Haddadin, S., Hirzinger, G., 2006. Collision detection and safe reaction with the DLR-iii lightweight manipulator arm. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 1623–1630.

De Luca, A., Flacco, F., 2012. Integrated control for pHRI: Collision avoidance, detection, reaction and collaboration. In: EEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob). pp. 288–295.

De Santis, A., Siciliano, B., De Luca, A., Bicchi, A., 2008. An atlas of physical human–robot interaction. Mech. Mach. Theory 43 (3), 253–270.

Falco, J.A., Marvel, J.A., Norcross, R.J., et al., 2012. Collaborative robotics: Measuring blunt force impacts on humans. Chest 140 (210), 45.

Fryman, J., Matthias, B., 2012. Safety of industrial robots: From conventional to collaborative applications. In: ROBOTIK. VDE, pp. 1–5.

Gamboa-Montero, J.J., Alonso-Martin, F., Castillo, J.C., Malfaz, M., Salichs, M.A., 2020. Detecting, locating and recognising human touches in social robots with contact microphones. Eng. Appl. Artif. Intell. 92, 103670.

Gribovskaya, E., Kheddar, A., Billard, A., 2011. Motion learning and adaptive impedance for robot control during physical interaction with humans. In: IEEE International Conference on Robotics and Automation. pp. 4326–4332.

Guiochet, J., Powell, D., Baudin, É., Blanquart, J.-P., 2008. Online safety monitoring using safety modes. In: Workshop on Technical Challenges for Dependable Robots in Human Environments. pp. 1–13.

Haddadin, S., Albu-Schaffer, A., De Luca, A., Hirzinger, G., 2008. Collision detection and reaction: A contribution to safe physical human-robot interaction. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 3356–3363.

Haddadin, S., De Luca, A., Albu-Schäffer, A., 2017. Robot collisions: A survey on detection, isolation, and identification. IEEE Trans. Robot. 33 (6), 1292–1312.

Harden, T., Schroeder, K.A., Pryor, M.W., 2011. On the Use of Joint Torque Sensors for Collision Detection in a Confined Environment. Technical Report, Los Alamos National Lab. (LANL), Los Alamos, NM (United States).

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.

Heinzmann, J., Zelinsky, A., 1999. The safe control of human-friendly robots. In: IROS. pp. 1020–1025.

Hennersperger, C., Fuerst, B., Virga, S., Zettinig, O., Frisch, B., Neff, T., Navab, N., 2017. Towards MRI-based autonomous robotic US acquisitions: a first feasibility study. IEEE Trans. Med. Imaging 36 (2), 538–548.

Heo, Y.J., Kim, D., Lee, W., Kim, H., Park, J., Chung, W.K., 2019. Collision detection for industrial collaborative robots: A deep learning approach. IEEE Robot. Autom. Lett. 4 (2), 740–746.

Hershey, S., Chaudhuri, S., Ellis, D.P., Gemmeke, J.F., Jansen, A., Moore, R.C., Plakal, M., Platt, D., Saurous, R.A., Seybold, B., et al., 2017. CNN architectures for large-scale audio classification. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 131–135.

Javidi, M., Jampour, M., 2020. A deep learning framework for text-independent writer identification. Eng. Appl. Artif. Intell. 95, 103912.

Kouris, A., Dimeas, F., Aspragathos, N., 2018. A frequency domain approach for contact type distinction in human–robot collaboration. IEEE Robot. Autom. Lett. 3 (2), 720–727.

Kumra, S., Kanan, C., 2017. Robotic grasp detection using deep convolutional neural networks. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 769–776.

Liang, P., Wang, W., Yuan, X., Liu, S., Zhang, L., Cheng, Y., 2022. Intelligent fault diagnosis of rolling bearing based on wavelet transform and improved ResNet under noisy labels and environment. Eng. Appl. Artif. Intell. 115, 105269.

Lippi, M., Marino, A., 2020. Enabling physical human-robot collaboration through contact classification and reaction. In: IEEE International Conference on Robot and Human Interactive Communication (RO-MAN). pp. 1196–1203.

Magrini, E., De Luca, A., 2017. Human-robot coexistence and contact handling with redundant robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4611–4617.

Mamedov, S., Mikhel, S., 2020. Practical aspects of model-based collision detection. Front. Robot. AI 7, 162.

Parusel, S., Haddadin, S., Albu-Schäffer, A., 2011. Modular state-based behavior control for safe human-robot interaction: A lightweight control architecture for a lightweight robot. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 4298–4305.

Popov, D., Klimchik, A., Mavridis, N., 2017. Collision detection, localization & classification for industrial robots with joint torque sensors. In: 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN). pp. 838–843.

Popov, D., Klimchik, A., Pashkevich, A., 2021a. Real-time estimation of multiple potential contact locations and forces. IEEE Robot. Autom. Lett. 6 (4), 7025–7032.

Popov, D., Klimchik, A., Pashkevich, A., 2022. Robustness of interaction parameters identification technique for collaborative robots. IEEE Robot. Autom. Lett. 7 (4), 8582–8589.

Popov, D., Mikhel, S., Yagfarov, R., Klimchik, A., Pashkevich, A., 2021b. Multi-scenario contacts handling for collaborative robots applications. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2985–2992.

Takakura, S., Murakami, T., Ohnishi, K., 1989. An approach to collision detection and recovery motion in industrial robot. In: 15th Annual Conference of IEEE Industrial Electronics Society. pp. 421–426.

Tchatchoua, P., Graton, G., Ouladsine, M., Muller, J., Traoré, A., Juge, M., 2022. 1D ResNet for fault detection and classification on sensor data in semiconductor manufacturing. In: IEEE International Conference on Control, Automation and Diagnosis (ICCAD). pp. 1–6.

Yamada, Y., Hirasawa, Y., Huang, S., Umetani, Y., Suita, K., 1997. Human-robot contact in the safeguarding space. IEEE/ASME Trans. Mech. 2 (4), 230–236.

Zanchettin, A.M., Ceriani, N.M., Rocco, P., Ding, H., Matthias, B., 2016. Safety in human-robot collaborative manufacturing environments: Metrics and control. IEEE Trans. Autom. Sci. Eng. 13 (2), 882–893.