Impact-Aware Dual-Arm Manipulation

A Framework for Fast Package Handling

By James Hermus, Michael Bombile, Jari J. van Steen, Elise Jeandupeux, Ahmed Zermane, Alessandro Melone, Mario Troebinger, Abdeldjallil Naceri, Claude Lacoursière, Stijn de Looijer, Sami Haddadin, Abderrahmane Kheddar, Alessandro Saccon, and Aude Billard

This article presents an impact-aware manipulation framework and its application to logistics, where the challenges related to the booming of e-commerce have increased the need for faster and more flexible package handling solutions. Particularly, the proposed impact-aware framework addresses the problem of swiftly grabbing and placing objects in depalletizing tasks with dual-arm robotic systems. Impact-aware robotics leverages intentional collisions to achieve dynamic interactions and thus has the potential to be faster and more energy efficient than the state of the art based on quasi-static interactions with objects or environments. The generation of desired impacts (contacts at a nonzero relative speed), generally avoided in classical robotics, brings multiple challenges encompassing the generation of robust motions, managing the impact with the object, dealing with the physical constraints of the robotic systems, contact state sensing, and simulation of contact behavior. To tackle these challenges, we developed, within the European Union (EU)-funded project Impact-Aware Manipulation (I.AM.), impact-aware technologies that yield an integrated impact-aware solution. The proposed framework exploits nonsmooth mechanics to provide robot–object–environment impact models; it uses dynamical systems (DSs) to generate nominal and contingency motions with intentional impacts; it leverages quadratic programming (QP)-based control to provide motion execution with the ability to enforce hardware and safety constraints; it employs internal state sensing that does not require an external force transducer; and, finally, it develops an impact simulation environment that can handle batch simulations. This article highlights the benefits of the proposed approach in terms of speed (a 29% decrease in average task time) and energy efficiency (a 35% decrease) through a systematic comparison between classical grabbing and impact-aware swift grabbing and tossing. In summary, our article underscores the transformative potential of impact-aware technologies in revolutionizing robotic logistics operations. An accompanying video is available at https://youtu.be/0Tv-MxO0rG0.

Digital Object Identifier 10.1109/MRA.2025.3615262

INTRODUCTION

E-commerce, the process of buying and selling physical goods online, has revolutionized the global marketplace. With the world e-commerce market projected to surpass 7.9 trillion USD by 2027 [1], its influence on warehouse operations, particularly intralogistics—the internal movement of goods within distribution centers [2]—is profound. In most warehouses, human workers are still responsible for picking and placing operations, where items are retrieved from pallets and transferred to conveyor belts, trays, or tote containers, a task commonly known as *depalletization*.

Despite its superior dexterity and adaptability compared to automated systems, the current workforce struggles to meet the increasing demands of the industry [3]. Several challenges contribute to this issue: there is a growing scarcity of labor, and the tasks are physically demanding and repetitive, leading to workplace injuries.

There exist fully automated depalletizing solutions that frequently rely on specialized equipment or single robotic arms[1] with tools adapted to the types of products to be depalletized. Specialized automated systems are convenient for high-volume processes with little variety in product types [2]. However, these systems have a considerably larger footprint compared to manual depalletizing solutions and exhibit limited flexibility in adapting to changing conditions or environments. An alternative approach is to use collaborative robots (cobots) [4]. Nevertheless, cobot systems tend to have lower throughput compared to human operators, primarily due to their quasi-static interaction methods. In addition, cobots commonly use suction grippers, which are increasingly problematic as the industry shifts toward packaging materials that use less plastic and are more environmentally sustainable. Such limitations partly motivate the use of dual-arm solutions.

In the industry, current dual-arm solutions rely mainly on human operators. Not only are humans able to immediately adapt to perturbations or changes in their environment and have a small footprint, but they can swiftly make and break contact to manipulate objects, as illustrated in Figure 1. Such skills have been notoriously hard to replace with robotic systems.

However, in recent decades, robots that can better withstand physical interaction with the environment, including impacts, while providing accurate sensing and actuation capabilities have been developed. This advance in hardware and control, as well as visual perception, presents an opportunity to alleviate this problem. A dual-arm robotic framework is one

[1]Recently, industrial applications have included XYZ Robotics' various depalletizing systems and Boston Dynamics' Stretch robot. These products largely use powerful single-arm systems equipped with various suction cups.

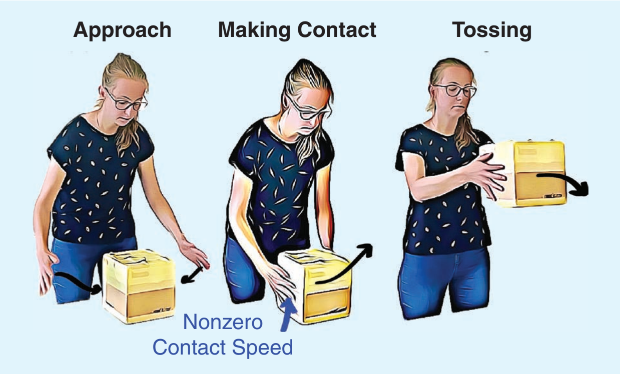

FIGURE 1. Schematic of the problem: Human workers swiftly grab and toss objects with ease, making and breaking contact at a non-zero relative velocity.

such realization. A dual-arm robotic system has some advantages compared to a single arm with suction cup or finger grippers. First, it spares the need to create custom end effectors for different types of products [5]. Second, some objects, such as open-lid crates or parcels, are difficult to handle by means of single or multiple suction cup end effectors.

In robotics, the exploitation of impacts, generally studied in locomotion that deals with hybrid dynamic systems [6], is now drawing increasing interest in the manipulation of objects with nonnegligible masses [7]. Making contact swiftly (impact) and breaking contact dynamically (tossing) together have the potential to substantially speed up depalletization tasks, as they both eliminate the need for complete pauses of the robots when grabbing and releasing objects. In practice, several key challenges must be overcome to generate impact-aware solutions with robots. These challenges include modeling, simulating and predicting impact behavior, producing robust motions, managing the impact while accounting for the physical constraints of the robot, and estimating the contact state.

CHALLENGES IN IMPACT-AWARE MANIPULATION

The current industry standard is to establish contact at a near-zero relative speed to have smooth transitions. However, this comes at the cost of time and energy. In contrast, achieving swift contacts induces impacts. If the object to be manipulated is not fragile, the impact can be exploited. This article is in part inspired by the human worker's ability to safely interact with objects at nonzero relative contact speeds, swiftly grabbing and tossing objects.

Yet, swiftly grabbing and tossing objects poses fundamental challenges.

1) Industry settings vary greatly, requiring solutions robust enough to account for differences in the box position, model, and timing.
2) Existing robotic systems are heavy and feature high gear ratios without modeling impact methods. Failure to account for impacts would result in excessively large forces, potentially damaging the contents of the box.
3) Real systems have physical limitations (joint limits, velocity limits, torque limits, and so on) imposing constraints that must be satisfied in practice.
4) Estimating the state of the system, especially detecting contact and estimating forces, is a notoriously hard challenge without having explicit external force sensors, which are expensive and fragile.
5) There is a need for a versatile simulation environment that interfaces between a physics engine that is suited to deal with impacts and a robot control framework and that allows collecting data for multiple scenarios through batch simulations.

CONTRIBUTION

This article focuses on one industrial use case in logistics that would benefit from the exploitation of impacts. As illustrated in Figure 2, we consider swift pick-and-place operations in depalletizing tasks using dual-arm systems [8], [9], [10]. We summarize the key contributions of the I.AM. European consortium in applying robot grabbing for logistics. Each key component was developed in isolation and has been individually published. In this article, we present an impact-aware framework, developed by integrating these individual components, and we quantify the improvements observed compared to a state-of-the-art classical approach. Furthermore, to assess the contribution of each module, we conduct an ablation study.

We demonstrate that an impact-aware control architecture enables dual-arm depalletizing robots to exhibit humanlike dynamic manipulation abilities, adaptability, and robustness. This approach addresses the increasing need for faster and more flexible package handling solutions. By incorporating these research contributions into a single demonstration, this article exemplifies how leveraging impact can reduce task time up to 29% for dynamic manipulation applications in logistics compared to standard robotic approaches.

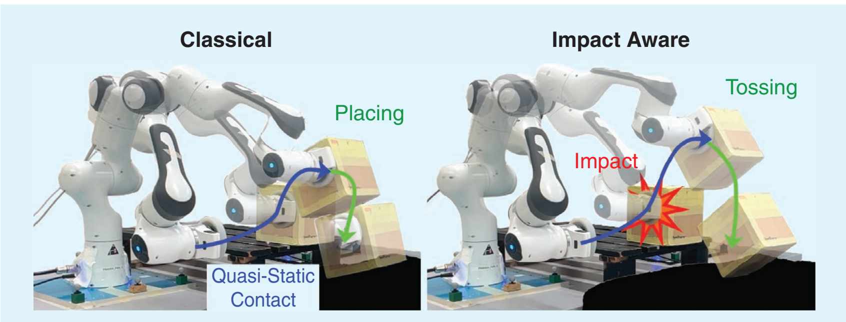

FIGURE 2. In the **classical condition**, the robot makes contact quasi statically (no impact) and breaks contact quasi statically (places). In the **impact-aware condition**, the robot makes contact dynamically (impact) and breaks contact dynamically (tossing). Note that this is a dual-arm task. However, for visual clarity, the right robot has been subtracted from the images.

Motivated by the challenges of depalletization, this article proposes a systematic evaluation of impact-aware manipulation in a task with a dual-arm robotic system (two Franka robots) that was used to dynamically grab boxes from a pallet. We demonstrate the effectiveness of the proposed approach in reducing task execution time and energy expenditure compared to quasi-static grabbing and placing.

In addition, as a result of the integration, resolution of associated challenges, and validation of impact-aware technologies, this article offers some recommendations and perspectives for the use of dual-arm impact-aware manipulation.

OUTLINE

After the introduction, this article is structured as follows. The "Methods" section consists of two parts. First, the integrated framework is introduced, with details on each of the individual integrated components. Second, the experimental method used to assess the performance of the integrated system is introduced. The "Results" section is divided into two subsections: a qualitative summary of observed behavior and a quantitative assessment of performance evaluation metrics. Finally, the "Discussion" and "Conclusions" sections explore the implications of this work, concluding the article with recommendations for future perspectives.

METHODS

We begin with an overview of how each integrated component functions and its interconnection in the overall control architecture. Then, we introduce each component individually: the pre-impact and postimpact motion, the reference adaptation for impacts, the constraint-aware controller, the contact state sensing, and the impact simulation environment. Following this, the experimental methods are presented, and, finally, the performance evaluation metrics are described.

SYSTEM INTEGRATION

This article introduces an integrated architecture to tackle the challenges of impact-aware manipulation, including five key components: 1) a DS-based motion and force generator that addresses robustness challenges of dual-arm coordination, 2) a reference spreading (RS) module that removes unwanted impact-induced peaks in the input signals, 3) a low-level constraint-aware control utilizing a quadratic program to account for the physical constraints of the robotic system, 4) a contact state sensing module used to detect impact and estimate contact forces, and 5) an impact simulation environment employed to numerically model the impact map in a range of pre-impact configurations. A graphical summary of these integrated components is provided in Figure 3.

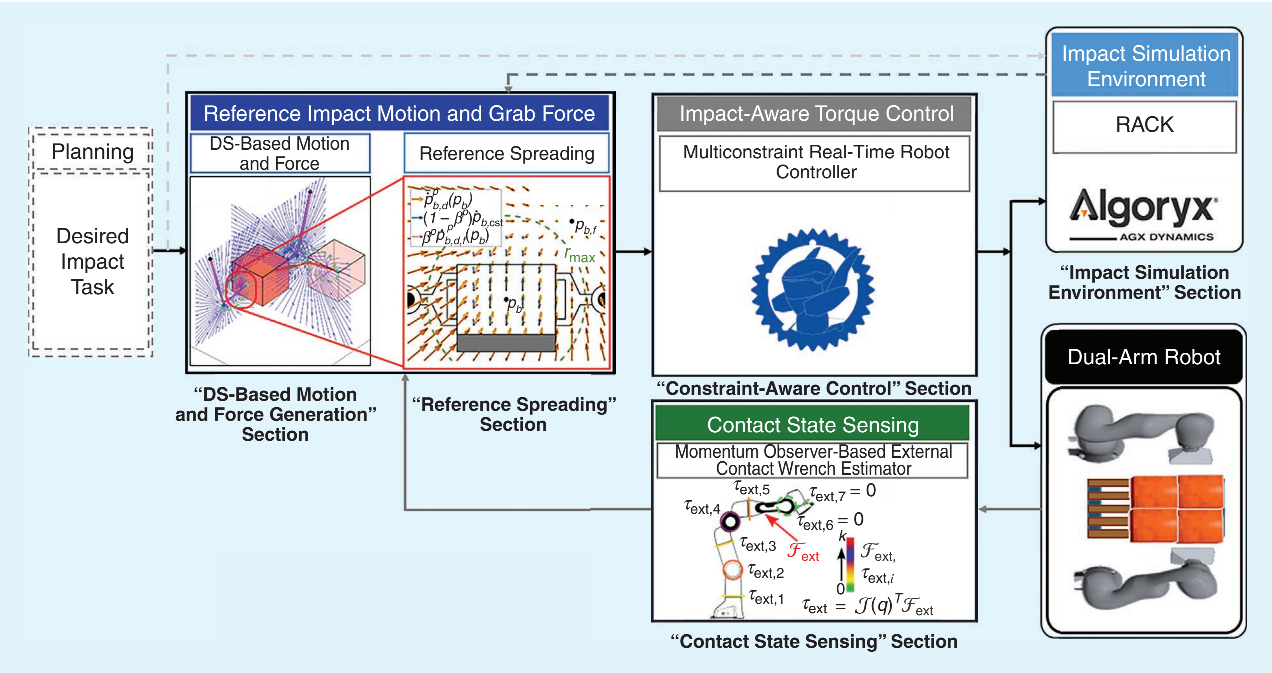

FIGURE 3. Diagram representing the connection between the different components. **Reference motion and force**: the motion flow generated by the DS outside and within the modulated region. The motion of each robot is shaped within the modulation region such that it passes through the desired transitory state (here, an impact state) with the desired position and the direction of the desired velocity (see [8] for more details). **Reference Spreading**: a 2D example of postimpact reference adaptation for impact tasks (figure from [11]). The postimpact reference matches a pre-impact reference where both robots grasp the box with an upward velocity. **Constraint-aware control**: the impact-aware QP regulates the contact velocity in a modified search space to ensure that the postimpact state jumps are hardware affordable. **State sensing**: the robot joint torque sensors and an inertial model are used to estimate the external contact wrenches with a momentum observer. **Impact simulation environment**: RACK is a simulation environment developed for impact simulation, is compatible with mc_rtc, and can run batch simulations.

DS-BASED MOTION AND FORCE GENERATION

Once the desired impact or tossing state has been defined in terms of the desired position and velocity that must be satisfied simultaneously, one needs to generate motions that can drive the robot toward such a desired state. To address this motion generation problem, unlike classical approaches based on motion planning, we propose a solution based on time-invariant DSs, which offers fast and time-independent replanning abilities and robustness to perturbations. More precisely, we adopt a modulated DS approach, where state-dependent modulation functions locally shape the robot's motion so that it passes through the desired impact or release states [8], [12]. This method generates motion toward an attractor located near the desired impact or tossing position, and when in its vicinity (within the modulation region), it reshapes the robot's motion such that it aligns first with the desired velocity while moving toward the desired impact or tossing state.

In this work, the DS-based motion generation initially begins with the grabbing action and, upon impact, automatically transitions to the tossing or placing action. Regarding the tossing task, [12] previously developed a method to determine the minimum release velocities of the object for given relative release positions given a learned inverse throwing map. Using these velocities, a kinematics-based bilevel optimization was employed to determine the associated feasible release states (positions and velocities) of the dual-arm robot for both fixed and moving targets.

In addition to controlling for impact, we also control the coordination of both robotic arms to ensure the success of the dual-arm grabbing task. A poorly coordinated system, where one arm reaches the object before the other, would lead not only to uncontrolled impact but also to failure of the post-grabbing task. To achieve dual-arm coordination with the DS, we compute the cooperative coordinates—the absolute and relative poses between the two end effectors—based on the current and desired end-effector poses for each arm. We then define stable dynamics for these cooperative coordinates to ensure convergence to their desired values. This allows for coordinated control of absolute and relative motion, facilitating synchronized reaching and closing of the dual-hand aperture. We couple the dynamics of relative motion to those of the absolute one to make the hand's aperture closure dependent on the reaching task. Finally, we map the cooperative coordinate dynamics back to the dual-arm end effectors. To ensure stable grasping of the object, a QP-based method was used for the online generation of contact forces that are consistent with the contact constraints. An illustration of the DS motion flow when grabbing with impact and tossing can be seen in Figure 3.

REFERENCE SPREADING

Performing motions with impacts results in instantaneous jumps in the robot velocity signals. If the pre- and postimpact velocity references do not capture this velocity jump correctly, switching from the pre-impact reference to the postimpact reference results in a large velocity tracking error. As a consequence, instant jumps of control input can lead to vibrations or unstable behaviors as well as damage, and they can increase energy consumption. Therefore, the pre- and post-impact velocity references are adapted in order to match the predicted postimpact velocity jump.

This article employs an RS approach, originally introduced in [13] and adapted in [11] to fit with time-invariant velocity references, such as those presented in the "DS-Based Motion and Force Generation" section. In this approach, numerical simulations using the open robotic simulation framework RACK and the commercial physics engine AGX Dynamics (see the "Impact Simulation Environment" section) are performed with the system initialized in a range of possible impact locations, with velocities corresponding to the pre-impact reference. The postimpact velocities resulting from these simulations are then saved and used to locally modify the postimpact velocity reference obtained from the DS-based approach. An example of this local modification for a 2D use case appears in Figure 3. Whenever the manipulated object is within a given distance from its initial position, a convex combination of the predicted postimpact velocity and the post-impact DS is used as the postimpact reference. This results in a more efficient postimpact motion without input spikes at the time of transitioning to the postimpact reference.

In addition to this reference adaptation, three control modes are defined, following earlier work on RS, such as [10] and [11]. These modes are 1) a pre-impact mode, 2) an interim mode, and 3) a postimpact mode. The pre-impact mode is active before any impact is detected. This is to track the pre-impact DS. As soon as the first impact is detected, we switch to the interim mode. The goal of this interim mode is to further prevent control input peaks, especially when impacts are planned to be executed simultaneously. If the impacts instead occur in short succession due to tracking errors or uncertainty in the environment, the contact state is uncertain and rapidly fluctuating. This means that none of the pre-impact and post-impact velocity references can be tracked reliably. The interim mode solves this by initially removing velocity feedback and gradually increasing the velocity feedback control gain over time while also gradually increasing the desired grasping force to promote contact completion. After a fixed time, when the impact event is assumed to be completed, a switch is made to the postimpact mode, where the adapted postimpact reference is tracked.

CONSTRAINT-AWARE CONTROL

To implement motion control and RS control motion, the physical constraints of the robotic system must be taken into account. This can be done using the mc_rtc[2] control framework. This framework uses task space control formulated as a quadratic program to generate desired joint accelerations, which enforces robot limitations as

[2]While this is not an exact acronym, it generally stands for *multiconstraint real-time controller*. More information is available at https://jrl-umi3218.github.io/mc_rtc/.

constraints [14]. These accelerations are used to compute torques sent as input to the robot's motor control. The QP constraints ensure that the robots do not violate joint limits and prevent (self-) collisions, while the cost function is created using tasks that require tracking the reference motion and force. Additionally, a posture task is added to resolve kinematic redundancy if any.

As shown in [15], impact awareness (the ability to handle impact tasks safely) can be achieved in this QP-based framework by reformulating and simply adding the task objectives and constraints associated with impact without introducing new decision variables. Although not integrated here, we refer the reader to [15] for more details.

CONTACT STATE SENSING

Since the RS approach, highlighted in the "Reference Spreading" section, uses impact detection in the switch policy between control modes, having a sensitive and robust impact detector is crucial. Force/torque sensors can be mounted between the robot flange and the end effector to accurately estimate the contact wrench at the end effector. However, incorporating external force/torque sensors into the robot will increase the complexity of the system, reduce the maximum payload, and limit the detection of external interactions with the robot flange. In our current system and scenarios, we are using seven-degree-of-freedom manipulators equipped with torque sensors in each joint. Using torque measurements and an accurate robot model, external contact wrenches are estimated using the momentum observer [16], without any additional hardware or sensors. To achieve accurate robot modeling and consequently improve wrench estimation, the inertial parameters of the arm are identified using a linear matrix inequality approach [17], which employs generalized robot base parameters [18].

The detector utilizes the estimated wrench and the velocity at the end effector. The latter is estimated using position encoders at the joints and is provided by the Franka Control Interface.

Given an abrupt change in end-effector velocity, the core insight of the detector to achieve both sensitivity and robustness is to trigger an impact detection only if a significant external force acts on the end effector in the same direction as the velocity change [10]. This in turn prevents false positive detection of impacts caused by an increased external force when contact is already established.

IMPACT SIMULATION ENVIRONMENT

RACK [19] is a simulation environment developed by Algoryx in collaboration with the Joint Robotics Laboratory at CNRS-AIST and extensively tested by the Eindhoven University of Technology (TU/e). It uses a human-readable robot scene description language and can, among others, be used for open-loop simulations, simulations with external control, parameter identification, and synthetic data generation. Although the only physics engine supported at the time of publication was AGX Dynamics,[3] the environment is designed with compatibility with other physics engines in mind, using dedicated plug-ins. AGX Dynamics is a discrete element multidomain simulation library based on nonsmooth multibody systems dynamics with dry frictional contacts and impacts, designed for real-time performance. It was chosen because it delivered good frictional contact and impact models and provided gravity compensation in simulations, similar to the Franka robots used in this article. A validation of simulations with AGX Dynamics that were performed using the RACK framework against real-world experiments is presented in [20]. This article demonstrates how we can estimate the jump in velocity caused by impacts between robots and the environment with the help of some of the features of the RACK framework. The so-called impact map was then validated against real-life experiments, showing only 3.1% average estimation error.

Through a communication protocol, simulations can be executed with synchronous robot control, and batch simulations can be performed by using the ability to read and write data from and open HDF5 files. Within this article, the control interface feature is used to communicate with the mc_rtc control framework, allowing testing the controller used for the experimental validation directly in simulations. This allows for safe controller development as well as initial tuning of the control parameters. The batch simulation feature is used in the RS approach, highlighted in the "Reference Spreading" section and described in detail in [11], helping to alleviate undesired peaks in the input signals. In particular, it is used to determine the estimated postimpact robot velocity using

[3]Available at http://www.algoryx.com/agx-dynamics.

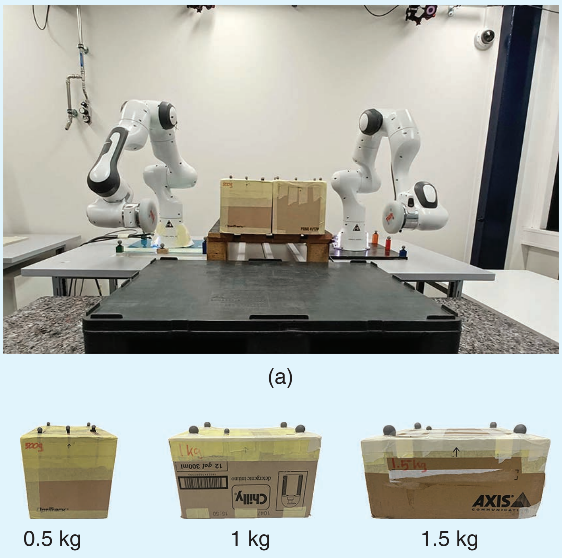

FIGURE 4. The experimental setup used to validate the proposed impact-aware dual-arm manipulation framework. (a) The robotic system. (b) The set of objects used for the systematic assessment of the system's performance: 0.5-, 1-, and 1.5-kg boxes.

the aforementioned validated impact map for a range of possible pre-impact states. Notice that these simulations assume the availability of a rigid model of the objects with which the robots interact, describing their inertia and dimensions. For many industrial applications, such as depalletization, as addressed in the "Introduction" section, this assumption is reasonable. If not available, (online) parameter identification is required before such simulations can be performed.

EXPERIMENTAL METHODS

To assess the benefits of leveraging impact, we performed experiments evaluating the performance of the swift dual-arm pick-and-toss approach. Our "impact-aware" method was compared to a pick-and-place approach using quasistatic contacts, referred to as the "classical" method in this context.

The section begins by describing the experimental setup and conditions. Next, it outlines the key performance indicators (KPIs).

EXPERIMENTAL SETUP

Two Franka robots are placed side by side, as in Figure 4(a). A soft pad, first used in [10], was attached as an end effector for each robot. It consists of a 3D-printed part with a layer of silicone to provide additional grip and impact damping, preventing impact-induced damage to the hardware. A motion capture system (OptiTrack) was used to track the position of the box. Data were recorded at 1 kHz.

To systematically assess the performance of the proposed impact-aware approach, we selected three boxes with different weights and dimensions. The first box, weighing 0.5 kg, contained foam and rice, measuring [19, 18.3, 18.9] cm. The second box, weighing 1 kg, was filled with cardboard and measured [18.7, 28.9, 18.5] cm. Finally, the third box, weighing 1.5 kg, was filled with fabric and measured [26.8, 36, 24.8] cm. These boxes were chosen to represent typical packaging encountered in industrial settings. These objects are displayed in the supplementary video.

In addition, we evaluated the robustness of the method using two boxes with varied contents. These included a box filled with juice containers, simulating a standard grocery store box, and another box containing a loose drill chuck and a water bottle to emphasize potential changes in mass distribution during the task. Furthermore, we conducted a test where the system grasped two adjacent boxes to evaluate its performance under such conditions.

The considered dual-arm grab scenario involved several steps. Following a go command, both robots were steered by the DS toward the box to reach a desired impact speed at a given uncertainty margin prior to impact. This desired impact speed and contact uncertainty margin depend on the robot acceleration limits and the accuracy of the box state estimate. Given the hardware and industry application, a contact uncertainty margin of 6.75 cm was used. After an impact with either robot is detected (using the impact detection scheme highlighted in the "Contact State Sensing" section) the controller transitions from the pre-impact mode to the interim mode, followed by the postimpact mode, as explained in the "Reference Spreading" section. Here, a postimpact DS, locally modified to match the impact dynamics, is followed to lift the box. Here, there are two options. The first is to transit to a DS that "tosses" the box (releases the box with a nonzero velocity), and the second is to transition to a DS that "places" the box (releases the box at the final position with a quasi-static velocity). After the box reaches the goal, the robot returns to the initial position for another cycle.

For each combination of the five task conditions and three boxes (0.5, 1.0, and 1.5 kg), the grabbing was performed for 20 trials. This resulted in 300 trials. Here are the five task conditions considered (summarized in Table 1).

1) In the **impact-aware** task condition, the RS was active; the robot grabbed the box swiftly, resulting in impact; and the robot tossed the box to the goal by releasing the box with a nonzero relative velocity. This condition integrated all key components presented in this work.
2) In the **RS-ablation** task condition, the RS was not active.
3) In the impact ablation task condition, the robot made quasi-static contact with the box (contact speed of 0.1 m/s), and the RS was turned off. Thus, this condition removed the impact when making contact with the box.
4) In the **task-ablation** task condition, the robot placed the box at the desired position quasi statically. Thus, the robot did not dynamically break contact by tossing; instead, it placed the box at the target.
5) In the **classical** task condition, all three ablations were combined. Thus, in this case, the RS was inactive, the robot quasi-statically made contact, and the robot quasi statically placed the box at the target. Thus, all conditions were "inactive" when compared to the impact-aware task condition.

TABLE 1. A summary of the task conditions.

	RS	IMPACT	TOSSING
Impact aware	✓	✓	✓
RS ablation		✓	✓
Impact ablation			✓
Toss ablation	✓	✓	
Classical			

A checkmark (✓) denotes an active component.

PERFORMANCE EVALUATION METRICS

In this article, we evaluate the performance of the proposed impact-aware dual-arm controller in comparison to state-of-the-art dual-arm quasi-static control, referred to as *classical*. Additionally, we assess the system's performance when individual components are ablated. For evaluation, we consider the following KPIs. Methodologically, each of these parameters is computed on the basis of estimates and thresholds. An important measure for detecting these states is the estimate of the time that the robot makes or breaks contact.

The **task time** (in seconds) is defined as the time from the time the contact is detected to the time the release is detected. This measure quantifies the time from pick to place/toss regardless of the robot's reaching and retracting motion phases.

The **cycle time** (in seconds) is defined as the duration from the time the robot starts the task to when it returns to its initial position within a tolerance region of 10 cm. This metric measures the combined benefit of grabbing with impact and tossing.

The **pre-impact time** (in seconds) is computed to study the temporal benefit of making impulsive contact. It is defined as the time the robot end effector is within 10 cm of making contact with the box until contact with the box is detected.

The **mean desired acceleration** (meters per second squared) represents the velocity feedback signal calculated by the controller, which serves as a metric to indicate the presence and severity of unwanted impact-induced peaks in the control input signals. The mean desired acceleration is computed during the 0.05-s time after impact, which coincides with the time frame where the interim mode defined while the RS framework is active in the impact-aware task condition. Excessive spikes in the desired acceleration can excite vibration modes and lead to instability. Such spikes can also result in torque jumps exceeding the robot's limitations, making it challenging for the QP to find solutions within the robot's constraints.

The **robot energy consumption** (in joules) is defined as E and computed by

$$E = \Delta t \sum_{i=n}^{N} \left| \boldsymbol{\tau}_i^T \right| \left| \dot{\boldsymbol{q}}_i \right| \tag{1}$$

where the vertical bars denote the absolute value; n denotes the time sample index that corresponds to the instant when the robot's motion starts; N denotes the time sample index that corresponds to the sample time where the box is released; the joint torque measured by the robot is denoted by $\boldsymbol{\tau}$, where the constant time duration between samples is denoted by Δt; and the joint velocity of the robot is denoted by $\dot{\boldsymbol{q}}$. The absolute values of torque and velocity were computed, this assumes the work done on the robot is not recovered (e.g., through regenerative braking) is consistent with [12].

RESULTS

This section provides results derived from benchmarking experiments conducted at EPFL. The presentation is structured in distinct segments to facilitate clarity and comprehension. First, analysis of the qualitative behavior for the different tasks is presented. These visual representations offer insights into the observed behaviors within these tasks. Subsequently, attention turns to the performance evaluation metrics. By juxtaposing the results of tasks, a comparative analysis among the conditions is facilitated. Finally, the section culminates with a presentation of results highlighting the robustness of the employed methodologies. This serves to underscore the reliability and efficacy of the experimental approaches employed. Supplementary material is available at https://doi.org/10.1109/MRA.2025.3615262, and a video is provided at https://youtu.be/2tk_pJEbDcY.

QUALITATIVE RESULTS

The velocity norm, from representative trials under both the classical and impact-aware conditions, is presented in Figure 5. This figure offers detailed insight into task dynamics. Both conditions exhibit an increase in speed as they approach the box for contact. However, in the impact-aware case, the norm of the velocity reaches approximately 0.4 m/s at the moment of contact with the box, consistent with an impact event. In contrast, the classical condition shows a decrease in speed to 0.1 m/s at contact, with no significant discontinuity in velocity, consistent with its intended use as a control for quasi-static contact conditions. Following contact, velocity rapidly increases during the lifting phase. In the impact-aware condition, this lifting phase is notably shorter, and the box is released with nonzero velocity. Elsewhere, the classical condition takes longer to reach the goal; as it approaches, it again decreases in speed to place the box at the target, evident in the discernible drop in velocity just after 1.0 s. Finally, both conditions exhibit a spike in speed as they return to the initial position. Notably, the impact-aware condition begins its return much earlier than the classical method.

Figure 6 gives the absolute value of the desired acceleration, which is a good indication of unwanted input spikes, as addressed in the "Performance Evaluation Metrics" section. The desired acceleration is shown for the vertical (z) direction around the time of impact for the impact-aware case versus the RS ablation case. Results for this z direction are presented since, given the nonzero pre-impact velocity in this direction, the nonzero postimpact velocity in this direction can cause a sudden velocity tracking error and subsequent control

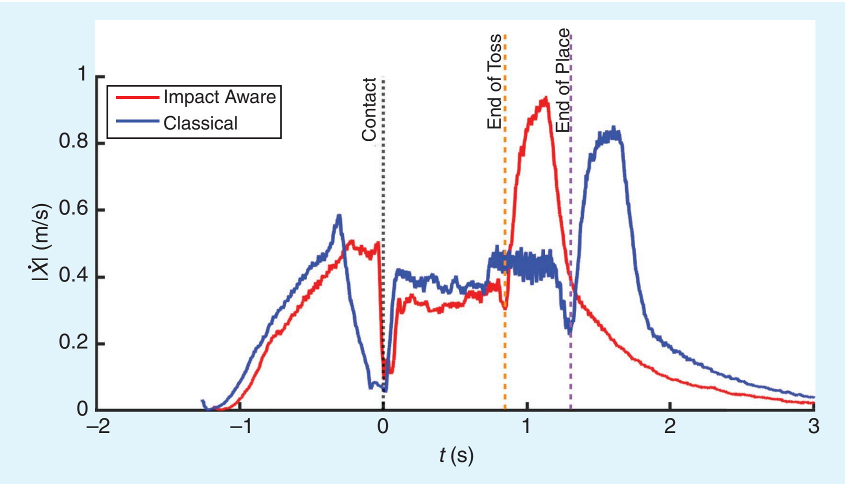

FIGURE 5. The **single-trial comparison**: velocity norm plots. The red line represents the impact-aware condition, while the blue line represents the classical condition.

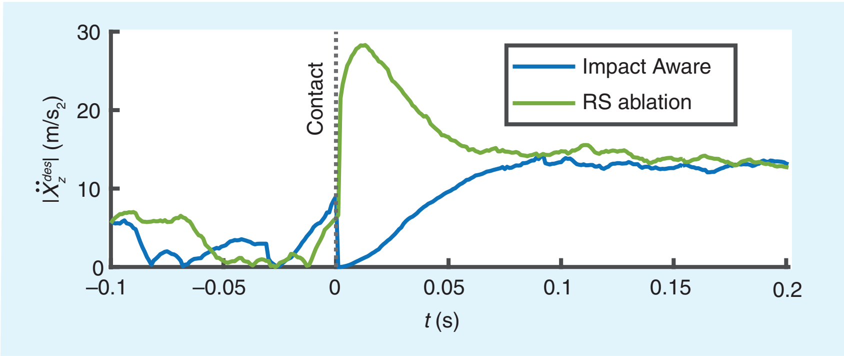

FIGURE 6. The desired acceleration in the vertical (z) direction commanded to robot for the impact-aware condition (blue) and RS ablation condition (green), both for a single experiment with the 1.5-kg box.

feedback error if not considered. The abrupt peak in the desired acceleration for the RS ablation case indicates such a sudden increase in the control effort that can lead to increased vibration, destabilization, and hardware damage. Instead, because of the RS approach taken in the impact-aware condition, this input peak is prevented, and the commanded acceleration gradually increases to the same steady-state level as the RS ablation case, away from the impact.

PERFORMANCE EVALUATION

The goal of performance evaluation metrics is to quantify the performance of the combined system and assess its efficacy. Below, we present the average task time, cycle time, pre-impact time, mean desired acceleration, and robot energy consumption. These results are displayed graphically for the 1.5-kg box in Figure 7 (see "Landing Position") and the results for all the boxes are listed in Table 2.

In Table 3, the percentage differences between each task and the classical method are reported. The impact-aware control demonstrated superior performance across several metrics compared to the classical approach. Specifically, it showed reductions in pre-impact time (8%), mean desired acceleration (76%), task time (29%), and cycle time (14%), indicating faster task completion. A positive percentage value indicates that the impact-aware task outperformed the classical method in terms of speed. In addition, the impact-aware condition showed a notable decrease in energy consumption, with a reduction of 35% compared to the classical approach.

The task time was defined as the duration from the detection of contact with the box to the moment of release of the box. This metric served to quantify only the advantage of tossing, as the benefit of achieving rapid impact occurs before contact. It is evident in Figure 7 that there is essentially a binary change in the time difference between conditions such that tasks with tossing active were more than 0.5 s faster.

Cycle time serves as a metric to quantify the overall time improvement of the combined system. Both the impact-aware and the RS ablation conditions emerged as the fastest. The advantage of making impulsive contact saved approximately 0.1 s, while the benefit of tossing also resulted in nearly 0.5 s of savings.

The pre-impact time metric aimed to quantify the time saved by initiating contact with the box at a nonzero velocity. We observed significant reductions in the percent difference between the classical and impact-aware task conditions (refer to Table 3 and Figure 7). It is crucial to note that this difference is dependent on the duration of movement at a velocity of 0.1 m/s before contact.

The desired mean acceleration served as a metric to quantify the presence and severity of unwanted peaks in the control input signals. In Figure 7, it is clear that the tasks with spreading of the reference (impact awareness and toss ablation) resulted in a substantially lower mean reference acceleration than in the cases without.

In the ablation study, several key aspects emerge from the results. First, in the ablation of RS, there is more than an approximately twofold increase in the observed mean reference acceleration between the RS and ablated RS conditions. This suggests that this control regime significantly benefits from the presence of RS, enabling the use of higher gains and

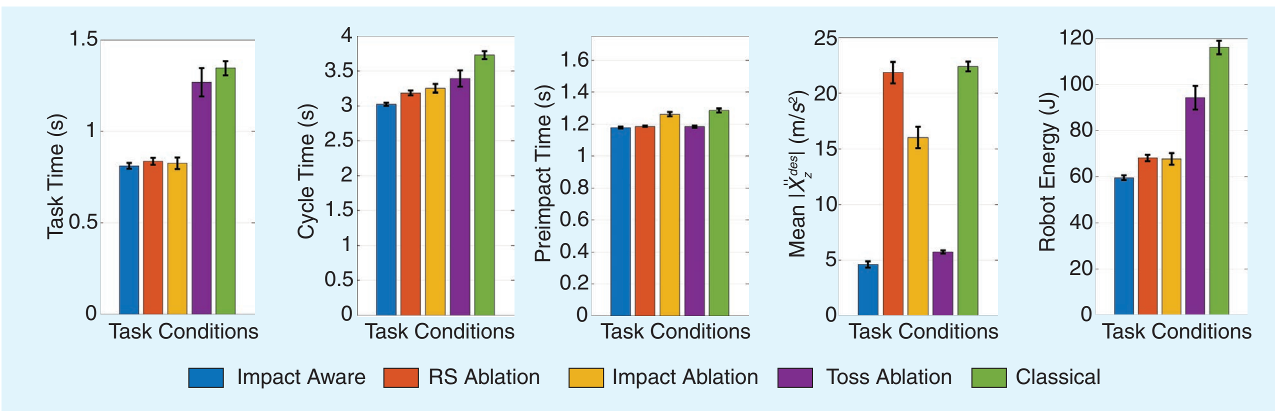

FIGURE 7. The key results. For the 1.5-kg box, each of the performance evaluation metrics means is presented. The black error bars denote plus or minus one standard deviation from the mean. The legend defines which tasks correspond to which colors.

TABLE 2. A summary of the mean and standard deviation of the metrics.

METRIC	BOX MASS (KG)	IMPACT AWARE	RS ABLATION	IMPACT ABLATION	TOSSING ABLATION	CLASSICAL
Task time (s)	0.5	0.71 ± 0.02	0.64 ± 0.04	0.64 ± 0.02	1.29 ± 0.02	1.19 ± 0.03
	1.0	0.83 ± 0.05	0.79 ± 0.02	0.76 ± 0.03	1.24 ± 0.08	1.17 ± 0.03
	1.5	0.81 ± 0.02	0.84 ± 0.02	0.83 ± 0.03	1.27 ± 0.08	1.35 ± 0.04
Cycle time (s)	0.5	3.14 ± 0.03	3.09 ± 0.03	3.22 ± 0.05	3.55 ± 0.03	3.67 ± 0.05
	1.0	3.08 ± 0.09	3.11 ± 0.05	3.21 ± 0.03	3.40 ± 0.14	3.65 ± 0.05
	1.5	3.02 ± 0.02	3.18 ± 0.03	3.25 ± 0.06	3.39 ± 0.12	3.73 ± 0.06
Pre-impact time (s)	0.5	1.26 ± 0.01	1.27 ± 0.05	1.39 ± 0.02	1.27 ± 0.01	1.42 ± 0.02
	1.0	1.17 ± 0.01	1.17 ± 0.00	1.32 ± 0.02	1.19 ± 0.02	1.39 ± 0.03
	1.5	1.18 ± 0.01	1.19 ± 0.00	1.26 ± 0.01	1.18 ± 0.01	1.29 ± 0.01
Mean desired acceleration (m/s^2)	0.5	3.83 ± 0.22	16.19 ± 2.99	11.86 ± 0.55	4.66 ± 0.19	18.66 ± 0.32
	1.0	4.29 ± 0.31	19.08 ± 0.75	14.13 ± 0.85	5.08 ± 0.44	20.80 ± 0.39
	1.5	4.61 ± 0.26	21.85 ± 0.97	16.02 ± 0.95	5.74 ± 0.15	22.40 ± 0.43
Robot energy consumption (J)	0.5	68.46 ± 1.90	68.52 ± 1.21	67.27 ± 1.62	95.59 ± 1.65	106.85 ± 2.12
	1.0	66.92 ± 5.00	68.29 ± 2.46	62.97 ± 2.78	91.56 ± 8.46	102.99 ± 2.24
	1.5	59.57 ± 1.16	68.21 ± 1.44	67.68 ± 2.54	94.29 ± 5.17	116.20 ± 2.95

Landing Position

The landing position of the 1.5-kg box was calculated for the reported results. The standard deviation of the landing position for the impact-aware case (with a toss) and in the classical case (with a placing action) is reported in Table S1.

TABLE S1. The standard deviation of the final box position.

DIRECTION	*X* (CM)	*Y* (CM)
Impact aware	2.6	2.0
Classical	0.5	0.1

achieving better tracking performance without introducing peaks in the control inputs at the time of impact. Second, when impact was ablated, there was an increase in the pre-impact time. Third, when the toss was ablated, there was also a substantial increase in task time. Thus, each component of this design has been validated separately, supporting the reasons for their presence in the control framework.

ROBUSTNESS EVALUATION

To evaluate the robustness of the impact-aware system, experiments were conducted using a broader selection of boxes. These tests included a box containing juice containers, a box housing a loose drill chuck and a water bottle, and simultaneous grasping of two boxes. The system was able to successfully grab and toss each of the three objects quickly and without failure.

TABLE 3. A summary of the percent difference in each KPI between the classical condition and the impact-aware condition.

	BOX TYPE		
Average KPI	0.5 (kg)	1.0 (kg)	1.5 (kg)
Task time	41%	29%	40%
Cycle time	14%	16%	19%
Pre-impact time	11%	16%	8%
Desired acceleration	76%*	78%*	79%*
Robot energy	36%	35%	49%

This table includes results from the different boxes. Note that positive differences correspond to the case where the I.AM. condition outperformed the industry-standard condition. Furthermore, an asterisk (*) indicates that the difference between the mean desired accelerations was computed between the impact-aware and RS-ablation conditions.

DISCUSSION

In this article, we highlight the contributions of the different components used in impact-aware grabbing and tossing developed within the European consortium project I.AM, a collaborative effort involving CNRS, EPFL, TU/e, the Technical University of Munich, and Algoryx. The overarching goal of the I.AM. project is to advance robotics technology, particularly in logistics operations, through the integration of impact-aware technologies. Our work combines several research advancements, including motion generation with a DS, reference adaptation for impacts, constraint-aware control, contact state sensing, and a contact simulation environment. These aspects are pivotal in addressing challenges related to robustness, executing impacts with heavy/highly geared robots, and controlling systems with physical constraints.

KEY RESULTS

Our platform demonstrates notable advancements, particularly in achieving faster task completion and improved energy efficiency compared to conventional industry methods reliant on quasi-static contact.

For example, in the case of the 1-kg box, our pick-and-toss task resulted in a 29% reduction in the average task time compared to the industry-standard place task. This substantial efficiency gain highlights the potential of impulsive contact strategies to significantly minimize the duration of the task and improve the speed of logistics operations. Furthermore, analysis of the average robot energy consumption revealed a 35% decrease in energy expenditure during the pick-and-toss task compared to the pick-and-place task, further emphasizing the benefits of impact-aware technologies in optimizing resource utilization without compromising task performance.

Considering Figure 7, notice that the RS ablation case results in only a relatively small increase in the cycle time. However, as can be seen for the mean desired acceleration, the benefit from RS becomes apparent from the increase in the mean desired acceleration when RS is ablated, indicating a spike in the input signals that can induce vibrations and increase the forces on the robot and objects. This can in turn increase the likelihood of failed task execution and wear on the objects and robots and, thus, is not desired.

Notice that the time savings resulting from tossing are approximately 0.5 s per cycle, as seen in the task time plot in Figure 7, substantially outperforming the impulse pick action, which saved around 0.1 s per cycle, as seen in the pre-impact time plot in Figure 7. Although both strategies contribute to reducing the total cycle time, the magnitude of time saved by tossing is important given scenarios where the considered task permits such action. This disparity in time savings underscores the crucial role of tossing action in expediting logistics operations, particularly in tasks involving repetitive actions and large volumes of objects. For tasks requiring thousands or even millions of cycles, the cumulative time savings achieved by tossing or impulsively picking can be substantial, leading to significant improvements in overall operational efficiency and throughput.

The positive effect from the impulsive pick action could be further improved by ensuring that the impact directions and speeds of both arms are defined such that their net effect propels the object toward its desired postimpact state (the actual postimpact state depends on the properties of the colliding bodies). Inadequate impact directions would unnecessarily stress the object. This highlights the importance of an impact planner. Even if such a module has not been integrated into this article, research is underway to develop a method for optimal impact states given an object's postgrabbing manipulation task.

Regarding tossing, it is worth pointing out that making decisions about when to toss an object or not (based on the predictability of the outcome and whether this is relevant for the task to be executed) is an ongoing research topic.

ASSUMPTIONS AND THRESHOLDS

In this article, we chose an impact velocity of 0.1 m/s and a contact uncertainty margin of 6.75 cm to comprise the quasi-static condition. These parameters directly affect the reported time metrics. We want to clearly articulate that the advantages of impact-based grabbing, as reported here, will decrease as box position certainty improves to allow for a smaller contact uncertainty margin. In contrast, the advantages of impulsive contact grow in scenarios with greater uncertainty, where the RS method is particularly effective. As discussed in the "RS" section, the RS approach, especially its interim mode, is designed to manage positional uncertainty in the environment, as shown in [11], where experiments introduced up to 3 cm of artificial uncertainty. Therefore, under conditions of higher positional uncertainty, the time savings from impulsive contact are expected to exceed the 8% improvement reported here.

In this article, we used an OptiTrack motion capture system to measure the position of the box. It is important to note that this choice does not represent a fundamental limitation; a variety of pose estimation tools or methods could be employed to estimate the box's pose in practice.

FRAGILE OBJECTS

One key insight derived from our work is the nuanced impact of moving into contact with nonzero velocity, contingent upon factors such as the precision of object modeling and the fragility of manipulated objects. Although precise object models enhance the benefits of impulsive contact, the approach was found to work quite well for real-world industrial applications, even when modeling assumptions were violated. This demonstrates the advantages of this approach, particularly for resilient objects.

However, it is essential to acknowledge the inherent limitations and considerations associated with impact-aware strategies. The maximum allowable impact speed plays a crucial role, as excessively high impact velocities could lead to undesirable consequences, especially for fragile objects. Therefore, a careful compromise must be maintained between the predictability of the behavior of objects and the potential impact forces to maximize the effectiveness of impact-aware control methods.

ROBUSTNESS

In this article, the selected boxes were intended to mirror objects commonly encountered in industrial settings. We conducted tests with various masses, sizes, shapes, and inertial distributions, including objects that were not single rigid bodies. However, in our robustness setup, we limited our examination to boxes with a mass no greater than 2 kg. The decision stemmed from the robot's ability to impulsively pick and toss each object 20 times without failure.

The Franka robot's maximum payload restricted the experimental objects. However, there are no theoretical limitations that prevent the application of this technology to any torque-controlled robotic platform with larger torque limits, such as the Kuka IIWA. It is important to note that applying this technology to a system with lower torque limits effectively highlights the advantages of QP control frameworks, such as mc_rtc. Operating close to the hardware limitations of the robot represents a significant real-world challenge.

Notice that while the dual-arm soft-pad design proved effective for a broad range of objects, there are instances where it may not be ideal. For example, bins with handles (designed for lifting rather than grasping under pressure) present challenges. Their deformation and narrowing toward the top indicate a lack of structural integrity for compression-based grasping. In theory, these methods are limited to objects that can withstand compression-based grasping. Thus, the choice to investigate boxed objects was based on its application to industry.

Similarly, small or irregularly shaped soft objects wrapped in plastic may be better suited for single-gripper or suction cup gripper systems. Our experimental setup exhibited remarkable robustness, enabling rapid grabbing and tossing of a wide array of objects.

Notice that the adoption of this technology does not preclude the use of other end effectors or robotic systems employing alternative grasping approaches for objects unsuitable for compression-based manipulation in industrial settings.

PREGRASPING MANIPULATION

In this article, our concern was to measure the advantages of utilizing impact, enabled by integrating scientific knowledge into the dual-arm pick-and-toss task. Thus, we specifically targeted scenarios where the box was prepared for grasping. However, in depalletizing processes, preparatory actions, typically undertaken by human workers, are often necessary to position the box for grasping. Although aspects of pregrasping manipulation are underway, they remain an area of future work.

CONNECTION TO HUMAN RESEARCH

We would like to point out that humans can exhibit extremely low mechanical impedance at the hands or fingertips relative to current robotic systems. Thus, their solutions to handle fast contact are substantially different from those of a torque-controlled robot with large gear reductions. Although motivation comes from human behavior, the solution presented here is distinct from that of a biologically motivated controller. Thus, in this work, we specifically did not make comparisons to quantify the extent to which the robot behavior was humanlike.

SUMMARY

Our article underscores the transformative potential of impact-aware technologies in revolutionizing robotic logistics operations. By addressing challenges associated with conventional quasi-static methods and leveraging controlled impacts, these technologies offer significant improvements in task efficiency and energy utilization, thereby paving the way for enhanced productivity and operational effectiveness in warehouse and distribution center environments.

CONCLUSIONS

This article highlighted the successful integration of impact-aware technologies and their application in logistics scenarios, with a specific focus on grabbing and depalletizing tasks involving dual-arm robotic systems. Through deliberate utilization of intentional collisions, we demonstrated the superior speed and energy efficiency achievable with impact-aware robotics, surpassing state-of-the-art approaches reliant on quasi-static interactions with objects or environments.

The integrated components are pivotal in addressing challenges related to robustness, executing impacts with heavy/highly geared robots, controlling systems with physical constraints, estimating contact force, and numerically modeling impact. Our work combines several research advancements, including motion generation with a DS, reference adaptation for impacts, constraint-aware control, contact state sensing, and a contact simulation environment.

This article emphasized the advantages of our proposed approach through extensive experimentation and systematic comparison between classical grabbing techniques and integrated impact-aware strategies. These findings underscore the transformative potential of impact-aware technologies in revolutionizing robotic logistics operations.

ACKNOWLEDGMENT

This work was funded by the EU's Horizon 2020 research and innovation program, under Grant 871899, through the I.AM. project (https://i-am-project.eu/). James Hermus and Michael Bombile contributed equally to this work. This article has supplementary downloadable material available at https://doi.org/10.1109/MRA.2025.3615262, provided by the authors.

AUTHORS

James Hermus, Learning Algorithm and Systems Laboratory, EPFL, 1015 Lausanne, Switzerland. E-mail: james.hermus@epfl.ch.

Michael Bombile, Learning Algorithm and Systems Laboratory, EPFL, 1015 Lausanne, Switzerland. E-mail: michael.bombile@epfl.ch.

Jari J. van Steen, Faculty of Mechanical Engineering, Eindhoven University of Technology, 5612 Eindhoven, The Netherlands. E-mail: j.j.v.steen@tue.nl.

Elise Jeandupeux, Learning Algorithm and Systems Laboratory, EPFL, 1015 Lausanne, Switzerland. E-mail: elise.jeandupeux@epfl.ch.

Ahmed Zermane, Laboratory of Computer Science, Robotics, and Microelectronics, CNRS–University of Montpellier, 34090 Montpellier, France. E-mail: ahmed.zermane@lirmm.fr.

Alessandro Melone, Munich Institute of Robotics and Machine Intelligence, Technical University of Munich, 80333 Munich, Germany. E-mail: alessandro.melone@tum.de.

Mario Troebinger, Munich Institute of Robotics and Machine Intelligence, Technical University of Munich, 80333 Munich, Germany. E-mail: mario.troebinger@gmail.com.

Abdeldjallil Naceri, Munich Institute of Robotics and Machine Intelligence, Technical University of Munich, 80333 Munich, Germany. E-mail: djallil.naceri@tum.de.

Claude Lacoursière, Algoryx Dynamics, 90736 Umeå, Sweden. E-mail: claude.lacoursiere@algoryx.com.

Stijn de Looijer, Vanderlande, North Brabant, 5466 AC Veghel, The Netherlands. E-mail: stijn.de.looijer@vanderlande.com.

Sami Haddadin, Munich Institute of Robotics and Machine Intelligence, Technical University of Munich, 80333 Munich, Germany. E-mail: sami.haddadin@gmx.de.

Abderrahmane Kheddar, Laboratory of Computer Science, Robotics, and Microelectronics, CNRS–University of Montpellier, 34090 Montpellier, France. E-mail: kheddar@gmail.com.

Alessandro Saccon, Faculty of Mechanical Engineering, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands. E-mail: a.saccon@tue.nl.

Aude Billard, Learning Algorithm and Systems Laboratory, EPFL, 1015 Lausanne, Switzerland. E-mail: aude.billard@epfl.ch.

REFERENCES

[1] K. Snyder, “35 e-commerce statistics of 2024,” *Forbes*, Mar 28, 2024. [Online]. Available: https://www.forbes.com/advisor/business/ecommerce-statistics/

[2] N. Boysen, R. de Koster, and D. Füßler, “The forgotten sons: Warehousing systems for brick-and-mortar retail chains,” *Eur. J. Oper. Res.*, vol. 288, no. 2, pp. 361–381, 2021, doi: 10.1016/j.ejor.2020.04.058.

[3] C. Mims, “As e-commerce booms, robots pick up human slack,” *The Wall Street J.*, Aug. 8, 2020. [Online]. Available: https://www.wsj.com/articles/as-e-commerce-booms-robots-pick-up-human-slack-11596859205

[4] M. Javaid, A. Haleem, R. P. Singh, S. Rab, and R. Suman, “Significant applications of cobots in the field of manufacturing,” *Cogn. Robot.*, vol. 2, pp. 222–233, Oct. 2022, doi: 10.1016/j.cogr.2022.10.001.

[5] K. Benali, J.-F. Brethé, F. Guérin, and M. Gorka, “Dual arm robot manipulator for grasping boxes of different dimensions in a logistics warehouse,” in *Proc. IEEE Int. Conf. Ind. Technol.*, 2018, pp. 147–152, doi: 10.1109/ICIT.2018.8352167.

[6] E. Westervelt, J. Grizzle, and D. Koditschek, “Zero dynamics of underactuated planar biped walkers,” *IFAC Proc. Volumes*, vol. 35, no. 1, pp. 551–556, 2002, doi: 10.3182/20020721-6-ES-1901.00904.

[7] H. Khurana and A. Billard, “Motion planning and inertia-based control for impact aware manipulation,” *IEEE Trans. Robot.*, vol. 40, pp. 2201–2216, 2024, doi: 10.1109/TRO.2023.3319895.

[8] M. Bombile and A. Billard, “Dual-arm control for coordinated fast grabbing and tossing of an object: Proposing a new approach,” *IEEE Robot. Autom. Mag.*, vol. 29, no. 3, pp. 127–138, Sep. 2022, doi: 10.1109/MRA.2022.3177355.

[9] N. Dehio, Y. Wang, and A. Kheddar, “Dual-arm box grabbing with impact-aware MPC utilizing soft deformable end-effector pads,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 5647–5654, Apr. 2022, doi: 10.1109/LRA.2022.3158433.

[10] J. van Steen, G. v. d. Brandt, N. van de Wouw, J. Kober, and A. Saccon, “Quadratic programming-based reference spreading control for dual-arm robotic manipulation with planned simultaneous impacts,” *IEEE Trans. Robot.*, vol. 40, pp. 3341–3355, 2024, doi: 10.1109/TRO.2024.3420800.

[11] J. Van Steen, N. Van de Wouw, and A. Saccon, “Impact-aware control using time-invariant reference spreading,” 2024, *arXiv:2411.09870*.

[12] M. Bombile and A. Billard, “Bimanual dynamic grabbing and tossing of objects onto a moving target,” *Rob. Auton. Syst.*, vol. 167, Sep. 2023, Art. no. 104481, doi: 10.1016/j.robot.2023.104481.

[13] A. Saccon, N. van de Wouw, and H. Nijmeijer, “Sensitivity analysis of hybrid systems with state jumps with application to trajectory tracking,” in *Proc. 53rd IEEE Conf. Decis. Control*, 2014, pp. 3065–3070, doi: 10.1109/CDC.2014.7039861.

[14] K. Bouyarmane, K. Chappellet, J. Vaillant, and A. Kheddar, “Quadratic programming for multirobot and task-space force control,” *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 64–77, Feb. 2019, doi: 10.1109/TRO.2018.2876782.

[15] Y. Wang, N. Dehio, A. Tanguy, and A. Kheddar, “Impact-aware task-space quadratic-programming control,” *Int. J. Robot. Res.*, vol. 42, no. 14, pp. 1265–1282, 2023, doi: 10.1177/02783649231198558.

[16] S. Haddadin, A. De Luca, and A. Albu-Schäffer, “Robot collisions: A survey on detection, isolation, and identification,” *IEEE Trans. Robot.*, vol. 33, no. 6, pp. 1292–1312, Dec. 2017, doi: 10.1109/TRO.2017.2723903.

[17] C. D. Sousa and R. Cortesao, “Physical feasibility of robot base inertial parameter identification: A linear matrix inequality approach,” *Int. J. Robot. Res.*, vol. 33, no. 6, pp. 931–944, 2014, doi: 10.1177/0278364913514870.

[18] M. Tröbinger, A. Naceri, X. Chen, H. Sadeghian, and S. Haddadin, “Identification of a generalized base inertial parameter set of robotic manipulators considering mounting configurations,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 11,502–11,508, doi: 10.1109/ICRA48891.2023.10160248.

[19] F. Nordfeldth, “GLUE application.” about.gitlab.com. 2024 [Online]. Available: https://gitlab.tue.nl/robotics-lab-public/glue-application

[20] J. Van Steen, D. Stokbroekx, N. Van de Wouw, and A. Saccon, “Impact-aware robotic manipulation: Quantifying the sim-to-real gap for velocity jumps,” 2024, *arXiv:2411.06319*.